



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

**МСЭ-Т**

СЕКТОР СТАНДАРТИЗАЦИИ  
ЭЛЕКТРОСВЯЗИ МСЭ

**J.162**

(11/2005)

СЕРИЯ J: КАБЕЛЬНЫЕ СЕТИ И ПЕРЕДАЧА  
СИГНАЛОВ ТЕЛЕВИЗИОННЫХ И ЗВУКОВЫХ  
ПРОГРАММ И ДРУГИХ МУЛЬТИМЕДИЙНЫХ  
СИГНАЛОВ

Проект IPCablecom

---

**Протокол сигнализации сетевого вызова  
для предоставления критических во времени  
услуг по сетям кабельного телевидения  
с использованием кабельных модемов**

Рекомендация МСЭ-Т J.162

---



## **Рекомендация МСЭ-Т J.162**

### **Протокол сигнализации сетевого вызова для предоставления критических во времени услуг по сетям кабельного телевидения с использованием кабельных модемов**

#### **Резюме**

В настоящей Рекомендации приводится описание профиля приложения программного интерфейса, интерфейса контроллера медиашлюза (MGCI) и соответствующего протокола контроля медиашлюзов (MGCP) для контроля над клиентами, встроенными в систему передачи голоса через IP-протокол (VoIP), из внешних элементов контроля вызовов. Протокол MGCP предполагает наличие архитектуры контроля вызовов, при которой "интеллект" контроля вызова находится вне шлюзов, и управляется внешними элементами контроля вызова. Профиль, описанный в настоящей Рекомендации, будет именоваться сетевым протоколом сигнализации вызова (NCS).

#### **Источник**

Рекомендация МСЭ-Т J.162 утверждена 29 ноября 2005 года 9-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

## ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

## ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или применимости), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений. Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

## ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без письменного разрешения МСЭ.

## СОДЕРЖАНИЕ

	Стр.
1	Сфера применения ..... 1
2	Справочная литература..... 1
2.1	Нормативная справочная литература ..... 1
2.2	Информативная справочная литература..... 2
3	Термины, определения и сокращения..... 3
3.1	Термины и определения..... 3
3.2	Сокращения..... 3
4	Соглашения по терминологии ..... 3
5	Введение ..... 4
5.1	Связь со стандартами H.323 ..... 5
5.2	Связь со стандартами IETF ..... 5
5.3	Связь с RFC 3435 и грамматикой ABNF ..... 6
6	Интерфейс контроллера медийного шлюза (MGCI)..... 6
6.1	Модель и соглашения об именовании ..... 6
6.2	Использование протокола SDP..... 14
6.3	Функции контроля шлюзов ..... 14
6.4	Состояния, восстановление после отказа и состязательные условия..... 40
6.5	Коды завершения и коды ошибок ..... 55
6.6	Коды причины..... 56
6.7	Использование опций локального соединения и дескрипторов соединения... 57
7	Протокол контроля медийных шлюзов..... 60
7.1	Общее описание..... 61
7.2	Заголовок команды ..... 61
7.3	Форматы заголовков ответа..... 75
7.4	Кодирование описания сеанса..... 78
7.5	Передача по протоколу UDP ..... 89
7.6	Совмещение передачи запросов и ответов..... 91
7.7	Идентификаторы транзакций и установление трехсторонней связи..... 92
7.8	Временные ответы..... 93
8	Безопасность..... 94
	Приложение А – Пакеты событий ..... 95
A.1	Линейный пакет ..... 95
A.2	Пакет факсимильной связи ..... 102
	Приложение В – Динамическое качество обслуживания..... 138
	Дополнение I – Пример пакета событий..... 146

	<b>Стр.</b>
Дополнение II – Примеры кодирования команд .....	148
II.1    Команда NotificationRequest .....	148
II.2    Команда Notify .....	148
II.3    Команда CreateConnection.....	149
II.4    Команда ModifyConnection .....	150
II.5    Команда DeleteConnection (от агента вызова).....	151
II.6    Команда DeleteConnection (от встроенного клиента).....	151
II.7    Команда DeleteConnection (от агента вызова для нескольких соединений) ....	152
II.8    Команда AuditEndpoint.....	152
II.9    Команда AuditConnection .....	153
II.10   Команда RestartInProgress .....	154
Дополнение III – Пример последовательности операций для вызова.....	155
Дополнение IV – Режим соединения.....	161
Дополнение V – Информация о совместимости .....	165
Дополнение VI – Дополнительные примеры пакетов событий .....	166
Дополнение VII – Пакеты событий .....	174
Дополнение VIII – Применение протокола NCS к оконечному устройству IPAT сети SCN .....	181
VIII.1  Обзор.....	181
VIII.2  Архитектура оконечного устройства IPAT .....	181
VIII.3  Требования к электрическим и физическим интерфейсам .....	182
VIII.4  Пакет NCS для сообщений протокола сети SCN с интерфейсом V5.....	183
VIII.5  Обеспечение конфигураций .....	191
VIII.6  Поддержка европейского линейного пакета .....	191
VIII.7  Примеры последовательностей операций при вызовах .....	192
Дополнение IX – Поддержка тарификации для сигнализации NCS проекта IPCablecom .....	202
IX.1    Цели .....	202
IX.2    Пакет автоматической тарификации.....	202
IX.3    Случаи использования – примеры последовательностей операций при вызовах .....	205
IX.4    Термины.....	207
Дополнение X – Грамматика ABNF для NCS .....	209
БИБЛИОГРАФИЯ .....	217

# Протокол сигнализации сетевого вызова для предоставления критических во времени услуг по сетям кабельного телевидения с использованием кабельных модемов

## 1 Сфера применения

В настоящей Рекомендации приводится описание профиля приложения программного интерфейса, интерфейса контроллера медиашлюза (MGCI) и соответствующего протокола контроля медиашлюзов (MGCP) для контроля над клиентами, встроенными в систему передачи голоса через IP-протокол (VoIP), из внешних элементов контроля вызовов. Протокол MGCP предполагает наличие архитектуры контроля вызовов, при которой "интеллект" контроля вызова находится вне шлюзов, и управляется внешними элементами контроля вызова. Профиль, описанный в настоящей Рекомендации, будет именоваться сетевым протоколом сигнализации вызова (NCS).

Настоящая Рекомендация основана на протоколе контроля медиашлюзов (MGCP) версии 1.0 (документ RFC705), являющемся результатом объединения простого протокола контроля шлюзов и семейства протоколов контроля IP-устройств (IPDC). Настоящая Рекомендация технически совместима с соответствующей спецификацией CableLabs PacketCable.

## 2 Справочная литература

### 2.1 Нормативная справочная литература

В перечисленных ниже Рекомендациях МСЭ-Т и другой справочной литературе содержатся положения, которые посредством ссылок в этом тексте составляют основные положения данной Рекомендации. На момент опубликования, действовали указанные редакции документов. Все Рекомендации и другая справочная литература, являются предметом корректировки, и стороны пришли к договоренности основываться на этой Рекомендации и стараться изыскивать возможность для использования самых последних изданий Рекомендации и справочной литературы, перечисленной ниже. Регулярно публикуется перечень действующих Рекомендаций МСЭ-Т. Ссылка на документ в рамках этой Рекомендации не даёт ему, как отдельному документу, статуса Рекомендации.

- ITU-T Recommendation G.168 (2004), *Digital network echo cancellers.*
- ITU-T Recommendation J.83 (1997), *Digital multi-programme systems for television, sound and data services for cable distribution.*
- ITU-T Recommendation J.112 Annex A (2001), *Digital Video Broadcasting: DVB interaction channel for Cable TV (CATV) distribution systems.*
- ITU-T Recommendation J.112 Annex B (2004), *Data-over-cable service interface specifications: Radio-frequency interface specification.*
- ITU-T Recommendation J.160 (2005), *Architectural framework for the delivery of time-critical services over cable television networks using cable modems.*
- ITU-T Recommendation J.161 (2001), *Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.*
- ITU-T Recommendation J.163 (2005), *Dynamic quality of service for the provision of real-time services over cable television networks using cable modems.*
- ITU-T Recommendation V.8 (2000), *Procedures for starting sessions of data transmission over the public switched telephone network.*

- ITU-T Recommendation V.25 (1996), *Automatic answering equipment and general procedures for automatic calling equipment on the general switched telephone network including procedures for disabling of echo control devices for both manually and automatically established calls.*
- IETF RFC 821 (1982), *Simple Mail Transfer Protocol.*
- IETF RFC 1034 (1987), *Domain names – Concepts and facilities.*
- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.*
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF.*
- IETF RFC 2327 (1998), *SDP: Session Description Protocol.*
- IETF RFC 2543 (1999), *SIP: Session Initiation Protocol.*
- IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications.*

## **2.2 Информативная справочная литература**

- ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems.*
- CableLabs PKT-SP-EC-MGCP-I08-030728, *PacketCable Network-Based Call Signaling Protocol Specification.*
- IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control.*
- IETF RFC 2705 (1999), *Media Gateway Control Protocol (MGCP) Version 1.0.*
- ETSI ETS 300 001 ed. 4 (1997-01), *Attachments to the Public Switched Telephone Network (PSTN); General technical requirements for equipment connected to an analogue subscriber interface in the PSTN.*
- ETSI EN 300 659-1 V1.3.1 (2001-01), *Access and Terminals (AT); Analogue access to the Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 1: On-hook data transmission.*
- ETSI EN 300 659-3 V1.3.1 (2001-01), *Access and Terminals (AT); Analogue access to the Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 3: Data link message and parameter codings.*
- ETSI ETS 300 324-1 ed. 1 (1994-02), *V interfaces at the digital Local Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification.*
- ETSI ETS 300 347-1 ed. 1 (1994-09), *V interfaces at the digital Local Exchange (LE); V5.2 interface for the support of Access Network (AN); Part 1: V5.2 interface specification.*
- ETSI ETS 300 166 ed. 1 (1993-08), *Transmission and Multiplexing (TM); Physical and electrical characteristics of hierarchical digital interfaces for equipment using the 2048 kbit/s-based plesiochronous or synchronous digital hierarchies.*
- ETSI ETS 300 167 ed. 1 (1993-08), *Transmission and Multiplexing (TM); Functional characteristics of 2048 kbit/s interfaces.*

### 3 Термины, определения и сокращения

#### 3.1 Термины и определения

В настоящей Рекомендации определяются следующие термины:

**3.1.1 кабельный модем:** Кабельный модем – это оконечное устройство второго уровня, которым завершается абонентское окончание соединения согласно Рекомендации МСЭ-Т J.112.

**3.1.2 проект IP**Cablecom**:** Проект МСЭ-Т, охватывающий архитектуру и серии Рекомендаций, которые обеспечивают доставку критических во времени интерактивных услуг по сетям кабельного телевидения.

#### 3.2 Сокращения

В настоящей Рекомендации используются следующие сокращения:

API	Прикладной программный интерфейс
CPE	Оборудование в помещении пользователя
DTMF	Двухтональный многочастотный сигнал
IP	Протокол Интернета (IP-протокол)
MGCI	Интерфейс контроллера медийного шлюза
MGCP	Протокол управления медийными шлюзами
MIB	Информационная база управления
MTA	Оконечный адаптер (абонентский адаптер среды передачи)
MWD	Максимальная задержка на ожидание
NCS	Сигнализация сетевого вызова
КТСОП	Коммутируемая телефонная сеть общего пользования
RTP	Протокол режима реального времени
SDP	Протокол описания сессии
UDP	Протокол дейтаграмм пользователя

### 4 Соглашения по терминологии

Во всем тексте настоящей Рекомендации слова, используемые для определения важности отдельных требований, даются заглавными буквами. К ним относятся:

"ДОЛЖЕН"	Это слово или слово "ТРЕБУЕТСЯ" означает, что рассматриваемое условие является безусловным требованием в настоящей Рекомендации.
"НЕ ДОЛЖЕН"	Это словосочетание означает, что рассматриваемое условие однозначно запрещается в настоящей Рекомендации.
"СЛЕДУЕТ"	Это слово или слово "РЕКОМЕНДУЕТСЯ" означает, что в определенных обстоятельства могут существовать обоснованные причины для того, чтобы не принимать во внимание данное условие, но, прежде чем выбрать другой курс, следует представить все последствия и тщательно взвесить все обстоятельства.
"НЕ СЛЕДУЕТ"	Это словосочетание означает, что могут существовать обоснованные причины в определенных обстоятельствах, когда упомянутое поведение является приемлемым или даже целесообразным, но, прежде чем перейти к реализации какой-либо линии поведения, которая отмечена этим словосочетанием, следует представить все последствия и тщательно взвесить все обстоятельства.
"МОЖЕТ"	Это слово или прилагательное "НЕОБЯЗАТЕЛЬНЫЙ" означает, что данное условие действительно носит необязательный характер и вводится по желанию.

Один поставщик может принять решение включить его, поскольку того требует, например, тот или иной рынок или это улучшает изделие; другой же поставщик может его опустить.

## 5 Введение

В настоящей Рекомендации приводится описание профиля NCS прикладного программного интерфейса (MGCI) и соответствующего протокола (MGCP) для контроля над встроенными клиентами из внешних элементов контроля вызовов. Встроенный клиент – это сетевой элемент, который предоставляет:

- две или несколько традиционных аналоговых линий доступа к сети передачи речи по IP-протоколу (VoIP);
- по желанию, одну или несколько видеолний к сети VoIP.

Использование встроенных клиентов подходит не только для домашнего использования. Например, их можно также использовать в деловых целях. Встроенные клиенты используются для линейного доступа, и поэтому предполагается наличие на них линейного оборудования, например аналоговые линии доступа для сопутствующих им обычных телефонов в противоположность магистральным шлюзам.

Протокол MGCP предполагает наличие архитектуры контроля вызовов, при которой "интеллект" контроля вызовов находится вне шлюзов, и им оперируют внешние элементы контроля вызовов, называемые агентами вызовов. В протоколе MGCP предполагается, что эти элементы контроля вызовов, или агенты вызовов (CA), будут синхронизированы друг с другом для передачи согласованных команд шлюзам, которые находятся под их контролем. В протоколе MGCP, описанном в настоящей Рекомендации, не определяется механизм синхронизации агентов вызовов, хотя такие механизмы могут быть описаны в будущих спецификациях проекта IP-Cablecom.

В протоколе MGCP предполагается модель соединения, где базовыми конструкциями являются конечные точки и соединения. Шлюз содержит набор конечных точек, являющихся источниками, или приемниками, данных и которые могут быть физическими или виртуальными.

Примером физической конечной точки является интерфейс шлюза, который завершает соединение обычной аналоговой телефонной линии (POTS) с телефоном, кнопочной телефонной системой, УАТС и т. д. Шлюз, которым заканчиваются домашние линии POTS (к телефонам), называется *домашним шлюзом, встроенным клиентом* или *адаптером МТА*. Встроенные клиенты могут также необязательно поддерживать видеосвязь.

Примером виртуальной конечной точки является аудиоисточник в сервере аудиоконтента. Создание физических конечных точек требует установки аппаратуры, в то время как создание виртуальных конечных точек может выполняться программными средствами. Однако профиль NCS протокола MGCP обращается только к физическим конечным точкам.

Тип таких соединений – "точка-точка". Тип "точка-точка" – это связь между двумя конечными точками с целью передачи данных между ними. После установления этой связи для обеих конечных точек между этими точками может проводиться передача данных. Эта связь устанавливается путем создания соединения в виде двух половин: одной половины в исходящей конечной точке, а другой – во входящей конечной точке.

Агенты вызовов посылают шлюзам команды на создание соединений между конечными точками и обнаружение определенных событий, например сигнала снятия телефонной трубки, а также генерирование определенных сигналов, например сигнала вызова. Только агент вызова определяет, как и когда устанавливаются соединения, между какими конечными точками, а также какие сигналы и события должны обнаруживаться и генерироваться в конечных точках. При этом шлюз становится простым устройством, не находящимся в каком-либо состоянии вызова, которое принимает от агента вызова общие команды без необходимости знать или даже понимать смысл вызовов, состояния вызовов, параметры или взаимосвязь между параметрами. При вводе новых услуг, изменении профилей пользователя и т. п. эти изменения являются прозрачными для шлюза. Агенты вызовов осуществляют изменения и генерируют соответствующий новый набор команд для шлюзов по введенным изменениям. Каждый раз, когда шлюз перезагружается, он будет находиться в свободном состоянии и просто выполнять команды агента вызова по мере их поступления.

## 5.1 Связь со стандартами H.323

Протокол MGCP разработан как внутренний протокол в распределенной системе, которая является для внешней среды единственным шлюзом VoIP. Эта система состоит из агента вызова, который может быть распределен или не распределен по нескольким компьютерным платформам, и из набора шлюзов. В конфигурации H.323 эта распределенная шлюзовая система может взаимодействовать с одной стороны с одной или несколькими линиями POTS, а с другой стороны – с совместимыми с H.323 системами, как показано ниже:

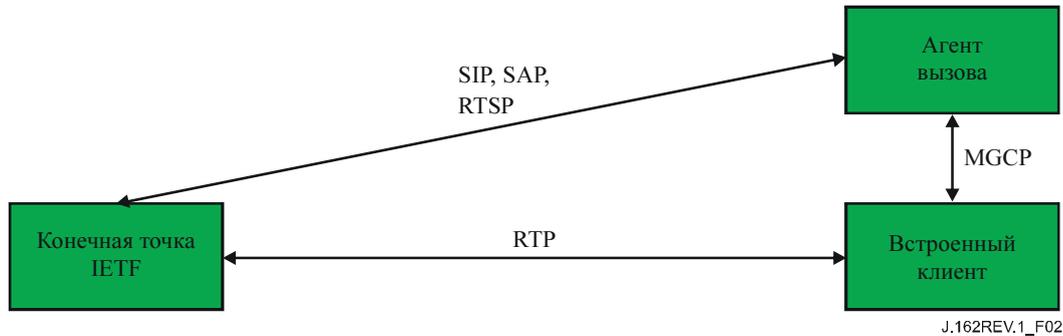


Рисунок 1/J.162 – Связь со стандартами H.323

В модели MGCP шлюзы выполняют главным образом функцию трансляции звуковых сигналов, в то время как агент вызова оперирует функциями обработки сигнализации и вызовов. Как следствие, агент вызова реализует уровни "сигнализации" стандарта H.323 и выступает в качестве "привратника H.323" или одной или нескольких "конечных точек H.323" для систем H.323.

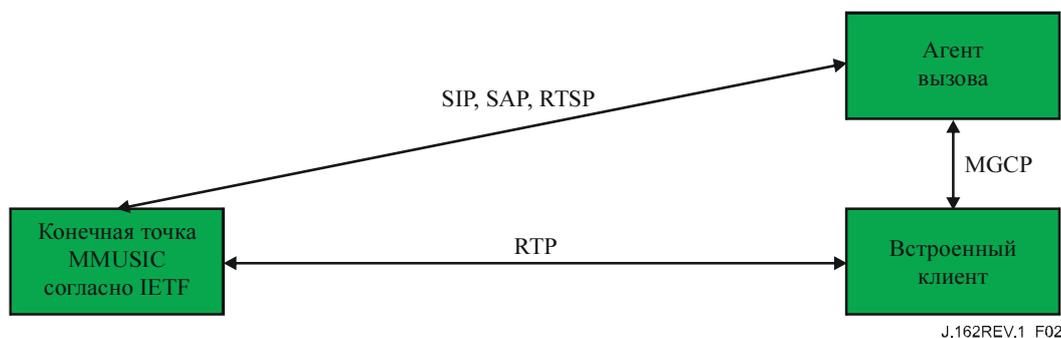
## 5.2 Связь со стандартами IETF

В то время как Рекомендация МСЭ-Т H.323 использовалась в качестве признанного стандарта для оконечных устройств VoIP, IETF разработала также спецификации для других типов мультимедийных приложений. В число этих спецификаций входят:

- протокол описания сеанса (SDP), RFC 2327;
- протокол объявления сеанса (SAP), RFC 2974;
- протокол инициирования сеанса (SIP), RFC 2543;
- протокол поточной передачи в режиме реального времени (RTSP), RFC 2326.

Последние три спецификации фактически являются альтернативными стандартами сигнализации, позволяющими передавать описание сеанса заинтересованной стороне. Протокол SAP используется администраторами многоканальных сеансов для распределения описания многоканального сеанса большой группе получателей. Протокол SIP используется для приглашения отдельного пользователя принять участие в сеансе "точка-точка" или одноканальной связи. Протокол RTSP используется для обеспечения взаимодействия с сервером, предоставляющего данные в режиме реального времени. Во всех трех случаях описание сеанса выполняется согласно протоколу SDP; если передаются аудиосигналы, они передаются посредством транспортного протокола реального времени (RTP и RTCP).

Распределенные шлюзовые системы и протокол MGCP обеспечат пользователям речевой связи сети КТСОП и пользователям встроенных клиентов возможность доступа к сеансам связи, установленным с использованием протоколов SAP, SIP или RTSP. Агент вызова предусматривает преобразование сигнализации, как показано ниже:



**Рисунок 2/J.162 – Связь со стандартами IETF**

В этой архитектуре стандарт протокола SDP является основным. Из дальнейшего описания видно, что он также используется для переноса описаний сеансов в протоколе MGCP.

### 5.3 Связь с RFC 3435 и грамматикой ABNF

RFC 3435 включает в себя формальное описание синтаксиса протокола MGCP, согласно "Расширенной BNF (форма Бэкуса-Науэра) для детальных указаний по синтаксису". На данное формальное описание ссылаются разработчики при создании совместно функционирующих устройств. Копия синтаксиса протокола MGCP, дополненная аннотацией и отредактированная для указания ее применимости к спецификациям IP-Cablecom, содержится в Дополнении Н. Следование данному руководству может улучшить совместимость посредством минимизации сбоев, вызванных различиями синтаксиса и грамматики.

## 6 Интерфейс контроллера медийного шлюза (MGCI)

Функции интерфейса MGCI предусматривают контроль соединения, контроль над конечными точками, сообщения о проверке и состоянии. В каждой из этих функций используются одна и та же модель системы и одни и те же соглашения об именовании.

### 6.1 Модель и соглашения об именовании

В протоколе MGCP предполагается модель соединения, где базовыми конструкциями являются конечные точки и соединения. Соединения группируются по вызовам. Одно или несколько соединений могут принадлежать одному вызову. Соединения и вызовы устанавливаются по инициативе одного или нескольких агентов вызовов.

#### 6.1.1 Имена конечных точек

Имена конечных точек, известные также как идентификаторы конечных точек, имеют две составляющие, которые определены здесь как нечувствительные к регистру:

- доменное имя шлюза, управляющего конечной точкой;
- локальное имя конечной точки в данном шлюзе.

Имена конечных точек будут иметь следующий вид:

```
local-endpoint-name@domain-name
```

где domain-name является абсолютным именем domain-name, как определено в RFC 1034, и содержит базовую часть; таким образом, примером доменного имени может служить конструкция:

```
MyEmbeddedClient.cablelabs.com
```

Кроме того, имя domain-name может быть точечно-десятичным представлением адреса IPv4 в виде текстовой строки, заключенной в левую и правую квадратные скобки ("[" и "]"), как, например, "[128.96.41.1]" – подробная информация содержится в стандарте RFC 821. Однако использование IP-адресов, как правило, нежелательно.

Встроенные клиенты могут иметь одну или несколько конечных точек (например, по одной для каждого гнезда RJ11 для "черных" телефонов), связанных с ними, и каждая из конечных точек

определяется отдельным локальным именем конечной точки. Как и доменное имя, локальное имя конечной точки нечувствительно к регистру. С локальным именем конечной точки связан endpoint-type (тип конечной точки), определяющий ее тип, как, например, аналоговый телефон или видеотелефон. Тип конечной точки может быть получен из локального имени конечной точки. Локальное имя конечной точки – это иерархическое имя, где наименее специфическим компонентом имени является левый крайний член, а наиболее специфическим компонентом – правый крайний член. Более формально, локальное имя конечной точки ДОЛЖНО соответствовать следующим правилам:

- Отдельные члены локального имени конечной точки должны быть разделены одиночной наклонной чертой ("/", шестнадцатеричное 2F в коде ASCII).
- Отдельные члены являются символьными строчками кода ASCII, составленными из букв, цифр или других печатных знаков, за исключением символов, используемых в качестве разграничителей в именах конечных точек ("/", "@"), символов, используемых для подстановки ("\*", "\$"), и знаков пробела.
- Операция подстановки представляется либо звездочкой ("\*"), либо знаком доллара ("\$") для элементов именованного ряда, которые должны быть подставлены. Таким образом, если полное локальное имя конечной точки выглядит как:

term1/term2/term3

и один из элементов этого имени обобщается, тогда локальное имя конечной точки выглядит следующим образом:

term1/term2/\*            если обобщается term3;  
term1/\*/\*                если обобщаются term2 и term3.

В каждом из примеров вместо звездочки может быть использован знак доллара.

- Выполнение подстановки допускается только справа; таким образом, если какой-либо элемент подставляется, тогда все элементы справа от него также должны подвергаться операции подстановки.
- В случаях, когда одновременно используются подстановочные знаки "звездочка" и "доллар", знаки доллара допускаются только справа; таким образом, если элемент имел подстановочный знак "доллар", то все элементы справа от него также должны иметь подстановочные знаки "доллар".
- Элемент, представленный знаком "звездочка", должен интерпретироваться следующим образом: "использовать *все* значения этого элемента, известные в области применения рассматриваемого встроенного клиента". Если не оговорено противоположного, это относится ко всем конечным точкам, сконфигурированным под данную услугу, в независимости от их текущего состояния эксплуатации, т. е. *задействована* или *не задействована*.
- Элемент, представленный знаком "доллар", должен интерпретироваться следующим образом: "использовать *любое одно* значение этого термина, известное в области применения рассматриваемого встроенного клиента".
- Каждый тип конечной точки может определять дополнительную особенность в правилах именования для этого типа конечной точки; однако такие правила не должны противоречить изложенным выше правилам.

Следует отметить, что различные типы конечных точек или даже различные подэлементы, например "линии", в пределах одного и того же типа конечной точки будут приводить к двум различным локальным именам конечных точек. Следовательно, каждая "линия" будет интерпретироваться как отдельная конечная точка. Следует обратить внимание, что поскольку часть доменного имени является частью идентификатора конечной точки, различные формы написания и различные значения, относящиеся к одному и тому же объекту, делают невозможным свободный обмен ими. Всегда НЕОБХОДИМО перезагружать последние полученные форму и значение.

#### 6.1.1.1 Имена конечных точек встроенных клиентов

Конечные точки во встроенных клиентах ДОЛЖНЫ поддерживать дополнительные соглашения об именовании, описанные в данном пункте.

Встроенные клиенты будут поддерживать следующие два типа конечных точек:

- Analogue Telephone (аналоговый телефон) – Аналоговый телефон представляется как аналоговая линия доступа (aaln). Это по существу эквивалент аналоговой телефонной линии, известной в сети КТСОП.

- Video (видео) – Детальная информация, относящаяся к типу видеоустройств, подлежит дальнейшему изучению.
- Basic Access ISDN (ЦСИС базового доступа) – Детальная информация, относящаяся к типу устройств ЦСИС, подлежит дальнейшему изучению.

#### 6.1.1.1.1 Конечные точки аналоговых линий доступа

В дополнение к соглашениям об именовании, описанным выше, локальные имена конечных точек для конечных точек типа "аналоговая линия доступа" (aaln) для встроенных клиентов ДОЛЖНЫ подчиняться следующим правилам:

- Локальные имена конечных точек содержат, как минимум, один элемент и, как максимум, два элемента.
- Элемент 1 (term1) ДОЛЖЕН быть элементом "aaln" или подстановочным знаком. Следует отметить, что использование подстановочного знака для элемента 1 может относиться к любому или ко всем типам конечных точек во встроенном клиенте независимо от их типа. Как правило, этот параметр предполагается использовать для административных целей, например для контроля или перезапуска.
- Элемент 2 (term2) ДОЛЖЕН быть числом от единицы до числа, представляющего собой количество аналоговых линий доступа, поддерживаемых рассматриваемым встроенным клиентом. Таким образом, это число определяет конкретную аналоговую линию доступа у встроенного клиента.
- Если локальное имя конечной точки состоит только из одного элемента, то им будет элемент 1.
- Если элемент 1 *не является* подстановочным знаком, тогда подстановочный знак "доллар" (относящийся к "любому одному значению") считается относящимся к элементу 2 (term2), то есть "aaln" эквивалентно "aaln/\$".
- Если элемент 1 *является* подстановочным знаком, тогда тот же подстановочный знак предполагается использовать и для элемента 2 (term2), то есть знаки "\*" и "\$" эквивалентны, соответственно, знакам "\*/\*" и "\$/\*".

Таким образом, примерами локальных имен конечных точек аналоговых линий доступа могут быть:

- aaln/1 Первая аналоговая линия доступа у рассматриваемого встроенного клиента.
- aaln/2 Вторая аналоговая линия доступа у рассматриваемого встроенного клиента.
- aaln/\$ Любая аналоговая линия доступа у рассматриваемого встроенного клиента.
- aaln/\* Все аналоговые линии доступа у рассматриваемого встроенного клиента.
- \* Все конечные точки (независимо от типа конечной точки) у рассматриваемого встроенного клиента.

Процесс обеспечения/(авто)конфигурации отвечает за получение и предоставление информации о том, сколько конечных точек имеет встроенный клиент и к какому типу относится каждая конечная точка. И хотя логически это разная информация, следует отметить, что *тип конечной точки* (endpoint-type) может быть получен из локальной части имени конечной точки.

#### 6.1.1.1.2 Конечные видеоточки

Подробная информация о конечных видеоточках будет представлена в дальнейшей версии настоящей Рекомендации.

#### 6.1.1.1.3 ЦСИС базового доступа

Подробная информация о ЦСИС базового доступа будет представлена в дальнейшей версии настоящей Рекомендации.

### 6.1.2 Имена вызовов

Вызовы определяются уникальными идентификаторами, не зависящими от базовых платформ или агентов. Идентификаторы вызовов – это шестнадцатеричные строки, которые создаются агентом вызова. ДОЛЖНЫ поддерживаться идентификаторы вызовов длиной максимум 32 символа.

Идентификаторы вызовов ДОЛЖНЫ быть уникальными, как минимум, в рамках группы агентов вызовов, которые управляют одними и теми же шлюзами. Однако вопросы координации этих идентификаторов вызовов между агентами вызовов выходят за рамки настоящей Рекомендации. Когда

агент вызова устанавливает несколько относящихся к одному и тому же вызову соединений либо в одном и том же шлюзе, либо в разных шлюзах, то все эти соединения могут быть связаны с одним и тем же вызовом посредством идентификатора вызова. Этот идентификатор затем можно использовать в процедурах учета или управления, которые выходят за рамки протокола MGCP.

### 6.1.3 Имена соединений

Идентификаторы соединений создаются шлюзом по запросу на установление соединения. Эти идентификаторы определяют соединение в контексте конечной точки. Идентификаторы соединений интерпретируются в протоколе MGCP как шестнадцатеричные строки. Шлюз ДОЛЖЕН гарантировать наличие надлежащего периода ожидания длительностью не менее трех минут между окончанием соединения, где используется этот идентификатор, и началом его использования в новом соединении для той же конечной точки. ДОЛЖНЫ поддерживаться имена соединений с максимальной длиной 32 символа.

### 6.1.4 Имена агентов вызовов и другие объекты

Протокол контроля над медийными шлюзами разработан для повышения надежности сети и позволяет осуществлять реализацию резервных агентов вызовов. Это означает, что между объектами и платформами аппаратных средств или сетевыми интерфейсами нет фиксированной связи.

Имена агентов вызовов, как и имена конечных точек, состоят из двух частей. Локальная часть имени не имеет какой-либо внутренней структуры. Примером имени агента вызова может служить имя:

```
cal@ca.whatever.net
```

Надежность обеспечивается с помощью следующих мер:

- Объекты, такие как встроенные клиенты или агенты вызовов, определяются их доменными именами, а не сетевыми адресами. Несколько адресов может быть связано с доменным именем. Если команда не может быть передана по одному из сетевых адресов, то реализации ДОЛЖНЫ предпринять повторную попытку передачи с использованием другого адреса.
- Объекты могут перемещаться на другую платформу. Информация о связи между логическим именем (доменным именем) и фактической платформой должна храниться в службе доменных имен (DNS). Агенты вызовов и шлюзы ДОЛЖНЫ отслеживать время существования записей, считываемых из DNS. Если время существования истекло, они ДОЛЖНЫ запрашивать в службе DNS обновление информации.

В дополнение к косвенной адресации, обеспечиваемой в результате использования доменных имен и службы DNS, концепция "уведомленного объекта" является основной для обеспечения надежности и восстановления после отказа в протоколе MGCP. "Уведомленным объектом" для конечной точки является агент вызова, который в текущий момент времени контролирует эту конечную точку. В любой момент времени конечная точка имеет один и только один связанный с нею "уведомленный объект", и когда этой конечной точке необходимо послать агенту вызова команду, она ДОЛЖНА послать команду действующему в настоящий момент "уведомленному объекту", конечной точке (точкам) которого предназначена команда. При запуске "уведомленный объект" ДОЛЖЕН быть установлен на значение готовности к работе. Большинство команд, посылаемых агентом вызова, имеют возможность неявным образом именовать "уведомленный объект" путем использования параметра "NotifiedEntity" (уведомленный объект) "Уведомленный объект" ДОЛЖЕН оставаться неизменным до тех пор, пока не будет принят новый параметр "NotifiedEntity" или не произойдет перезагрузка конечной точки. Если "уведомленный объект" для конечной точки не имеет значения (пуст) или не был установлен явным образом<sup>1</sup>, то этот объект будет установлен по умолчанию на адрес источника последней команды обработки соединения или запроса уведомления, полученного для конечной точки. Таким образом, проверка не изменит "уведомленный объект".

В пункте 6.4 содержится более подробное описание надежности и восстановления после отказа.

### 6.1.5 Отображения цифр набора номера

Агент вызова может запросить шлюз накапливать цифры, набираемые пользователем. Эта функция предназначена для использования аналоговых линий доступа с домашними шлюзами для сбора цифр,

---

<sup>1</sup> Это может произойти в результате описания пустого параметра NotifiedEntity.

набираемых пользователем; она также может быть использована для сбора кодов доступа, номеров кредитных карточек и других номеров, запрашиваемых службами контроля вызовов. Конечные точки ДОЛЖНЫ поддерживать отображения цифр набора номера так, как определено в данном пункте.

Альтернативная процедура заключается в использовании шлюза, уведомляющего агента вызова о набранных цифрах, как только они набираются; эта процедура также известна как передача с перекрытием (a.k.a.). Однако при такой процедуре возникает большое число взаимодействий. Предпочтительно накапливать набранные номера в буфере, а затем передавать их в одном сообщении.

Однако при использовании такого метода накопления набранных номеров есть проблема, состоящая в том, что шлюзу трудно спрогнозировать, сколько номеров необходимо накопить до передачи. Например, используя настольный телефон, можно набрать следующие номера:

0	Местный оператор
00	Оператор междугородной связи
xxxx	Местный добавочный номер
8xxxxxxx	Местный номер
#xxxxxxx	Добавление к локальному номеру в других корпоративных объектах
*xx	Услуги, помеченные звездочкой
91xxxxxxxxxx	Междугородный номер
9011 + до 15 цифр	Международный номер

**Рисунок 3/J.162 – Набранные цифры – Пример**

Решение этой проблемы состоит в загрузке шлюза отображением цифр, которое соответствует плану набора номера для той области, где находится шлюз. Таким образом, используемое фактическое отображение цифр может отличаться для разных регионов. Это отображение цифр выражено с использованием синтаксиса, полученного из системной команды egrep операционной системы UNIX. Например, план набора номера, описанный выше, приводит к следующему отображению цифр:

```
(0T| 00T| [1-7]xxx|8xxxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxxxx|9011x.T)
```

Формальный синтаксис отображения цифр описывается с помощью следующей записи в нормальной форме Бэкуса–Наура (BNF):

```
Digit      ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
Timer      ::= "T" | "t" - соответствует распознаванию таймера
Letter     ::= Digit | Timer | "#" | "*" | "A" | "a" | "B" | "b" | "C" | "c" | "D" | "d"
Range     ::= "X" | "x"          -- соответствует любой цифре
           | "[" Letters "]"    -- соответствует любой из заданных букв
Letters   ::= Subrange | Subrange Letters
Subrange  ::= Letter           -- соответствует заданной букве
           | Digit "-" Digit  -- соответствует любой цифре между первой и
                               -- последней
Position  ::= Letter | Range
StringElement ::= Position    -- соответствует появлению позиции
           | Position "."     -- соответствует произвольному числу
                               -- появлений позиции, включая 0
String    ::= StringElement | StringElement String
StringList ::= String | String "|" StringList
DigitMap  ::= String | "(" StringList ")"
```

Отображение цифр, согласно этому синтаксису, определяется либо с помощью (нечувствительной к регистру) "строки", либо с помощью "списка строк", по которым шлюз будет пытаться найти наиболее короткое возможное соответствие. Независимо от приведенного выше синтаксиса использование таймера в текущий момент допустимо, только если он появляется на последней позиции в строке<sup>2</sup>.

<sup>2</sup> Например, в записи "123T" и "123[1-2T5]" это правило выполняется, а в записи "12T3" – не выполняется.

Каждая строка в списке является альтернативной схемой нумерации. Шлюз, распознающий цифры, буквы или таймеры, будет:

- 1) добавлять код параметра события для цифры, буквы или таймера в качестве маркера на конце внутренней переменной состояния "текущая строка набора номера";
- 2) применять "текущую строку набора номера" к таблице отображений цифр, пытаясь найти соответствие всем выражениям в отображении цифр;
- 3) ничего не предпринимать в дальнейшем, если результат является частично определенным (частичное соответствие по меньшей мере одной записи в отображении цифр и неполное соответствие другой записи).

Если результат соответствует элементу отображения или является полностью определенным (то есть никакие другие цифры не могут влиять на соответствие), то текущая строка набора номера посылается агенту вызова<sup>3</sup> и затем стирается параметр „текущая строка набора номера“. Соответствие в настоящей Рекомендации может быть либо "полным" при точном соответствии одной из описанных альтернатив, либо "невозможным", имеющим место, когда строка набора номера не соответствует какой-либо альтернативе. Например, непредвиденные таймеры могут приводить к невозможным соответствиям. Как полные, так и невозможные соответствия инициируют уведомление о накопленных цифрах (которые могут содержать информацию о других событиях).

Таймер T – это таймер ввода цифр, который может быть использован двояко:

- Когда таймер T используется с отображением цифр<sup>4</sup>, он не запускается до тех пор, пока не будет введена первая цифра, и перезапускается после ввода каждой новой цифры, пока не появится либо соответствие отображению цифр, либо несоответствие. В этом случае таймер T функционирует как таймер времени между передачами цифр.
- Когда таймер T используется без отображения цифр, он запускается сразу же и просто аннулируется (но не перезапускается), как только вводится цифра. В этом случае таймер T может быть использован как таймер времени между передачами цифр, когда используется передача с перекрытием.

При использовании с отображением цифр таймер T принимает одно из двух значений –  $T_{par}$  или  $T_{crit}$ . Когда строке цифр требуется еще хотя бы одна цифра, чтобы соответствовать какой-либо из кодовых комбинаций в отображении цифр, таймер T принимает значение  $T_{par}$ , соответствующее частичной синхронизации набора цифр. Если таймер – это все время, которое требуется для получения соответствия, то таймер T принимает значение  $T_{crit}$ , соответствующее критической синхронизации. Когда таймер T используется без отображения цифр, то он принимает значение  $T_{crit}$ . Значение по умолчанию для  $T_{par}$  составляет 16 секунд, а для  $T_{crit}$  – 4 секунды. Подготовка к работе может увеличить оба эти значения.

В Приложении А содержится дополнительная подробная информация и приводится пример использования таймера T.

Конечные точки ДОЛЖНЫ поддерживать отображение цифр длиной не менее 2048 байт во всех телефонных интерфейсах.

Агент вызова может предоставлять шлюзу отображение цифр каждый раз, когда агент вызова выдает шлюзу команды на прием цифр набора номера. Кроме того, следует еще раз отметить, что подробная информация об используемом отображении цифр будет зависеть от области, в которой находится шлюз, таким образом, отображение цифр является программируемым. Отображение цифр, предоставляемое агентом вызова, ДОЛЖНО соответствовать описанному в данном пункте.

### 6.1.6 События и сигналы

Концепция событий и сигналов является базовой для протокола MGCP. Агент вызова может запрашивать уведомление об определенных событиях, происходящих в конечной точке, например события токового состояния абонентского шлейфа (снятие телефонной трубки). Агент вызова также может запрашивать подачу определенных сигналов в конечной точке, например тонального сигнала готовности к набору номера.

<sup>3</sup> Список цифр может также включать другие события – см. п. 6.4.3.1.

<sup>4</sup> Выражаясь техническим языком, с действием "накопление цифр согласно отображению цифр".

События и сигналы группируются в пакеты, в пределах которых они совместно используют одно и то же пространство имен, которое в дальнейшем будет называться именами событий. Пакет – это набор событий и сигналов, поддерживаемых конкретной конечной точкой. Например, один пакет может поддерживать определенную группу событий и сигналов для аналоговых линий доступа, а другой пакет – другую группу событий и сигналов для видеолиний. Для данного типа конечной точки могут существовать один или несколько пакетов, и у каждого типа конечной точки имеется связанный с ним пакет по умолчанию.

Имена событий состоят из имени пакета и кода события; одни и те же коды событий могут быть использованы в разных пакетах, поскольку каждый пакет определяет отдельное пространство имен. Имена пакетов и коды событий – это нечувствительные к регистру строки букв, цифр и дефисов с ограничением, состоящим в том, что дефисы НЕ ДОЛЖНЫ быть первым или последним символом в имени. Может потребоваться параметризация некоторых кодов событий дополнительными данными, что осуществляется внесением параметров между множеством круглых скобок. Имя пакета отделяется от кода события косой чертой ("/"). Имя пакета может быть исключено из имени события, и в этом случае предполагается наличие имени пакета по умолчанию для рассматриваемого типа конечной точки. Например, для аналоговой линии доступа с линейным пакетом (имя пакета "X"), являющимся пакетом по умолчанию, считаются равными два следующих имени событий:

- X/dl тональный сигнал готовности к набору номера в приводимом для примера линейном пакете для аналоговой линии доступа;
- dl тональный сигнал готовности к набору номера в приводимом для примера линейном пакете (по умолчанию) для аналоговой линии доступа.

В настоящей Рекомендации даются определения пакетов для типов встроенных клиентов, перечисленных в таблице 1.

**Таблица 1/J.162 – Пакеты для типов конечных точек встроенных клиентов**

Тип конечной точки	Пакет	Имя пакета	Пакет по умолчанию
Аналоговая линия доступа	Линейный	В	Да
Видео	Для дальнейшего изучения	Для дальнейшего изучения	Для дальнейшего изучения
Интерфейс BRI сети ЦСИС	Для дальнейшего изучения	Для дальнейшего изучения	Для дальнейшего изучения

В Приложении А определяется начальное множество пакетов. Дополнительные имена пакетов и коды событий могут быть определены и/или зарегистрированы в проекте IPcablecom. Любое изменение, вносимое в пакеты, определяемые в настоящей Рекомендации, ДОЛЖНО приводить к изменению имени пакета, или к изменению номера версии профиля NCS, или, возможно, к тому и к другому.

Каждый пакет ДОЛЖЕН иметь определение пакета, которое ДОЛЖНО описывать имя пакета, и определение каждого события, принадлежащего пакету. Определение события ДОЛЖНО включать точное имя события, то есть код события, определение события открытым текстом и, если необходимо, точное определение соответствующих сигналов, например точные частоты звуковых сигналов, таких как тональные сигналы готовности к набору номера или двухтональные многочастотные (DTMF) сигналы. Далее события должны указывать, устойчивы ли они (например, токовое состояние абонентского шлейфа (телефонная трубка снята); см. п. 6.3.1) и содержат ли они контролируемые состояния событий (например, токовое состояние абонентского шлейфа; см. п. 6.3.8.1). ДОЛЖЕН быть определен тип сигналов (включено/выключено, истечение времени ожидания события или короткий сигнал), и сигналы истечения времени ожидания события ДОЛЖНЫ иметь определяемое по умолчанию значение выдержки времени – см. п. 6.3.1.

В дополнение к использованию пакетов проекта IPcablecom реализаторы системы МОГУТ набраться опыта, создавая экспериментальные пакеты. Имя пакета для экспериментальных пакетов ДОЛЖНО начинаться с двух символов – "x-" или "X-"; в проекте IPcablecom НЕ ДОЛЖНЫ регистрироваться имена пакетов, которые начинаются с этих двух символов. Встроенный клиент, принимающий команду, относящуюся к неподдерживаемому пакету, ДОЛЖЕН послать в обратном направлении информацию об ошибке (код ошибки 518 – неподдерживаемый пакет).

Имена пакетов и коды событий поддерживают одно обозначение подстановочного знака каждый. Подстановочный знак "\*" (звездочка) может быть использован как относящийся ко всем пакетам,

поддерживаемым рассматриваемой конечной точкой, а код события "all" (все) – как относящийся ко всем событиям в рассматриваемом пакете. Например:

- X/all относится ко всем событиям в приводимом для примера линейном пакете для аналоговой линии доступа;
- \*/all для аналоговой линии доступа; относится ко всем пакетам и ко всем событиям в тех пакетах, которые поддерживаются рассматриваемой конечной точкой.

Следовательно, имя пакета "\*" НЕ ДОЛЖНО присваиваться пакету, а код события "all" НЕ ДОЛЖЕН использоваться ни в одном пакете.

События и сигналы по умолчанию обнаруживаются и генерируются в конечных точках; однако некоторые события и сигналы могут обнаруживаться и генерироваться в соединениях в дополнение к конечной точке или вместо нее. Например, конечным точкам может быть направлен запрос подать на соединение сигнал контроля посылки вызова. Чтобы сигнал или событие можно было обнаружить или сгенерировать в соединении, в определении события/сигнала ДОЛЖНО явно указываться, что событие/сигнал могут быть обнаружены или сгенерированы в соединении.

Если сигнал должен применяться в соединении, то к имени события добавляется имя соединения с использованием в качестве разграничителя коммерческого знака "собачка" (@) как в конструкции:

```
X/rt@0A3F58
```

Если нужно удалить соединение в тот момент, когда в нем определяются или подаются событие или сигнал, конкретное определение события или образование сигнала просто ДОЛЖНО быть остановлено. В зависимости от сигнала конечной точке СЛЕДУЕТ создавать ситуацию сбоя, т. е. если тип сигнала – ТО, то создается событие сбоя в выполнении, поскольку соединение, связанное с этим сигналом, будет удалено раньше, чем истечет время подачи сигнала. Действие при уведомлении, связанное с сообщением о сбое, должно соответствовать операциям по уведомлению так, как это определено в обработке Запроса на Уведомление (см. 6.3.1).

Подстановочный знак "\*" (звездочка) может быть использован для обозначения "всех соединений" в используемой конечной точке (точках). Когда используется такое условное обозначение, шлюз ДОЛЖЕН генерировать или обнаруживать событие во всех соединениях, которые подключены к конечной точке (точкам). Примером такого обозначения является:

```
X/rt@*
```

Однако если такое событие наблюдается в действительности, шлюз ДОЛЖЕН включать в себя имя конкретного соединения, на котором возникло событие. Подстановочный знак "\$" (знак доллара) может быть использован для обозначения "текущего события". Это обозначение НЕ ДОЛЖНО использоваться, если только запрос уведомления о событии не будет "инкапсулирован" в команде CreateConnection (создать соединение) или ModifyConnection (модифицировать соединение). Когда используется это обозначение, шлюз ДОЛЖЕН генерировать или обнаруживать событие в соединении, которое создается или модифицируется в текущий момент времени. Примером такого обозначения является:

```
X/rt@$
```

При обработке команды с использованием подстановки "текущее соединение" подстановочный символ "\$" ДОЛЖЕН быть восстановлен шлюзом до значения текущего соединения. Если последующая команда явно (например, через проверку соединения) или неявно (например, через устойчивость) ссылается на такое событие, восстановленное значение ДОЛЖНО быть использовано шлюзом. Другими словами, подстановка "текущего соединения" восстанавливается единожды, что происходит при начальной обработке команды, в которую она явно включена. Идентификатор соединения или замена подстановочного знака могут быть использованы в сочетании с обозначениями "все пакеты" и "все события". Например, запись:

```
*/all@*
```

может быть использована для обозначения всех событий во всех соединениях для используемой конечной точки (точек). Однако использование подстановок "все пакеты" или "все соединения" очень нежелательно.

Агенты вызова должны быть способны действовать в окружении, где некоторые конечные точки поддерживают не все пакеты. Конечная точка, получившая команду, ссылающуюся на пакет, который

она не поддерживает, должна ответить сообщением об ошибке 518 (Неподдерживаемый или неизвестный пакет). На получение такого сообщения об ошибке, агент вызова может попытаться послать ту же команду снова без параметра пакета, хотя в случаях, если первоначальная команда содержала параметры для множества пакетов, агента вызова может оказаться не способен определить, какой конкретно пакет(ы) исключить. Агент вызова может также использовать команду AuditEndpoint для определения набора пакетов, поддерживаемых конечной точкой.

## 6.2 Использование протокола SDP

Агент вызова использует протокол MGCP для обеспечения шлюзов описанием параметров соединения, таких как IP-адреса, порт UDP и профили протокола RTP. За исключением случаев, когда в настоящей Рекомендации оговаривается или подразумевается иное, описания протоколов SDP ДОЛЖНЫ следовать соглашениям, определенным в протоколе описания сеанса (SDP), который в настоящее время является стандартом RFC 2327, предложенным IETF. Кроме того, все агенты вызовов и шлюзы ДОЛЖНЫ игнорировать любые параметры SDP, атрибуты или поля, которые непонятны для агента вызова или шлюза.

В протоколе SDP предусматривается описание мультимедийных конференций. Профиль NCS будет поддерживать только установление аудио- и видеосоединений с использованием типов медиа "аудио" и "видео". В настоящее время определены только "аудиосоединения".

SDP допускает для описания факсимильных сообщений в режиме реального времени использование типа медиа "изображение". Профиль NCS будет поддерживать установку факсимильных соединений с использованием типа медиа "изображение".

## 6.3 Функции контроля шлюзов

В данном пункте приводится описание команд протокола MGCP в форме удаленного вызова процедуры (RPC-типа) интерфейса API, который будет именоваться интерфейсом контроллера медийного шлюза (MGCI). Функция MGCI определена для каждой команды протокола MGCP, где функцией MGCI берутся и возвращаются те же параметры, что и соответствующей командой протокола MGCP. Функции, рассмотренные в данном пункте, обеспечивают описание функционирования протокола MGCP на высоком уровне и описывают пример интерфейса API RPC-типа, который МОЖЕТ быть использован для реализации протокола MGCP. И хотя MGCI API является просто примером интерфейса API, семантическое поведение, определяемое интерфейсом MGCI, является неотъемлемой частью настоящей Рекомендации; все реализации ДОЛЖНЫ соответствовать семантике, определенной для интерфейса MGCI. Реальные обмениваемые сообщения MGCP, включая используемые форматы сообщений и кодирование, определяются в пункте о протоколе (п. 7). Встроенные клиенты и агенты вызова ДОЛЖНЫ осуществлять их реализацию так, как указано.

Услуги MGCI состоят из команд обработки соединения и обработки в конечных точках. Ниже приводится обзор этих команд:

- Агент вызова может выдать команду NotificationRequest для шлюза, приказывая шлюзу отслеживать конкретные события, такие как снятие и возврат на место телефонной трубки или сигналы DTMF в определенной конечной точке.
- Затем шлюз будет использовать команду Notify (уведомить) для информирования агента вызова, когда в заданной конечной точке имеют место запрашиваемые события.
- Агент вызова может использовать команду CreateConnection (создать соединение) для установления соединения, которое заканчивается в конечной точке самого шлюза.
- Агент вызова может использовать команду ModifyConnection (модифицировать соединение) для изменения параметров, связанных с ранее установленным соединением.
- Агент вызова может использовать команду DeleteConnection (удалить соединение) для удаления существующего соединения. При некоторых обстоятельствах команда DeleteConnection может быть также использована шлюзом для указания на то, что соединение больше не может поддерживаться.
- Агент вызова может использовать команды AuditEndpoint (проверить конечную точку) и AuditConnection (проверить соединение) для проверки состояния "конечной точки" и всех связанных с ней соединений. Обычно возникает потребность в больших возможностях управления сетью помимо возможностей, предоставляемых этими командами, например,

желательно иметь информацию о состоянии встроенного клиента. Для поддержки таких возможностей предполагается использование простого протокола управления сетью (SNMP) и определение базы MIB, выходящее за рамки настоящей Рекомендации.

- Шлюз может использовать команду RestartInProgress (процесс осуществления перезапуска) для уведомления агента вызова о том, что конечная точка или группа конечных точек, которыми управляет шлюз, выводится из эксплуатации и снова вводится в эксплуатацию.

Эти услуги позволяют контроллеру (обычно агенту вызова) выдавать шлюзу команду на создание соединений, которые заканчиваются в конечной точке, присоединенной к шлюзу, и получать информацию о событиях, имеющих место в конечной точке. В настоящее время конечная точка ограничивается конкретной аналоговой линией доступа через встроенного клиента.

Соединения группируются по "вызовам". Несколько соединений, которые могут принадлежать или не принадлежать одному и тому же вызову, могут завершаться в одной и той же конечной точке. Поток медиапакетов по каждому соединению управляется параметром "режим" (mode), который может быть установлен в состояние "только передача" (sendonly), "только прием" (recvonly), "передача/прием" (sendrecv), "конференция" (confnrc), "неактивный" (inactive), "копирование" (replcate), "сетевой шлейф" (netwloop) или "проверка целостности сети" (netwtest). Параметр "режим" определяет, могут ли мультимедийные пакеты быть переданы и/или получены по соединению. Протокол RTP не зависит от режима соединения; подробная информация содержится в Рекомендации МСЭ-Т J.161.

Обработка мультимедийных пакетов, полученных из конечной точки, определяется параметром режима:

- Медиаданные, исходящие из конечной точки, будут передаваться по всем соединениям для той конечной точки, режим которой соответствует состоянию "только передача", "передача/прием", "конференция" или "копирование".

Обработка медиаданных, полученных по этим соединениям, также определяется параметрами режима:

- Медиаданные, полученные в пакетах данных через соединения в режиме "неактивный", "только передача" и "копирование", отклоняются.
- Медиаданные, полученные в пакетах данных через соединения в режиме "только прием", "конференция" или "передача/прием", собираются вместе, а затем передаются конечной точке.
- В дополнение к передаче конечной точке медиаданные, полученные в пакетах данных через соединение в режиме "конференция", копируются для всех других соединений для конечной точки, режим которой соответствует состоянию "конференция". Подробности такой рассылки, например информация о трансляторе протокола RTP или смесителе (mixer), выходят за рамки настоящей Рекомендации.
- В дополнение к медиаданным, полученным из конечной точки, медиаданные, переданные конечной точке, смешиваются и передаются по всем другим соединениям для конечной точки, режим которой соответствует состоянию "копирование". Сюда СЛЕДУЕТ включить медиаданные, генерируемые сигналами, подаваемыми в конечную точку.
- Медиаданные, полученные в пакетах данных через соединение в режиме "сетевой шлейф" или "проверка целостности сети", будут передаваться в обратном направлении по соединению так, как описано ниже.

Если режим установлен в состояние "сетевой шлейф", звуковые сигналы, принятые из соединения, будут отражены в обратном направлении по тому же соединению. Режим "сетевой шлейф" ДОЛЖЕН функционировать просто как отражатель пакетов по RTP.

Режим "проверка целостности сети" используется для проверки целостности через IP-сеть. По IP-сети к конечным точкам передается сигнал, характерный для типа конечной точки, и тогда предполагается, что конечная точка воспроизводит этот сигнал в IP-сети, после того как он пройдет через внутреннее оборудование шлюза, для проверки правильности функционирования. До передачи в обратном направлении этот сигнал ДОЛЖЕН пройти через внутреннее декодирование и повторное кодирование. Для аналоговых линий доступа этот сигнал будет звуковым сигналом, и этот сигнал НЕ ДОЛЖЕН передаваться дальше к телефону, подключенному к аналоговой линии доступа, независимо от текущего состояния телефонной трубки этого телефона, то есть положена трубка или снята.

На новые и существующие соединения для конечной точки НЕ ДОЛЖНЫ влиять соединения в режиме "сетевой шлейф" или "проверка целостности сети". Однако ограничения на местные ресурсы могут ограничивать число новых соединений, которые могут быть установлены.

В режиме "копирование" ДОЛЖНО, как минимум, поддерживаться копирование потока от конечной точки и некоторого другого соединения в независимости от метода кодирования, используемого для этого другого "соединения". Однако от соединения в режиме "копирование" ТРЕБУЕТСЯ поддержка только результирующего медиапотока при кодировании<sup>5</sup> согласно Рекомендации G.711. Поддержка режима "конференция" является необязательной; все другие режимы соединения ДОЛЖНЫ поддерживаться. Иллюстрация взаимодействия режимов содержится в Дополнении IV.

### 6.3.1 Команда NotificationRequest

Команда NotificationRequest (запрос уведомления) используется для направления шлюзу запроса о передаче уведомления при возникновении указанных событий в конечной точке. Например, уведомление может быть запрошено, когда в конечной точке обнаруживаются тональные сигналы, связанные с факсимильной связью. Объект, получивший это уведомление (обычно это агент вызова), может затем принять решение, что в соединениях, связанных с этой конечной точкой, должен использоваться другой тип кодирования, и дать соответствующую команду шлюзу<sup>6</sup>.

ReturnCode

```
← NotificationRequest (EndpointId
                        [, NotifiedEntity]
                        [, RequestedEvents]
                        , RequestIdentifier
                        [, DigitMap]
                        [, SignalRequests]
                        [, QuarantineHandling]
                        [, DetectEvents])
```

**EndpointId** – это идентификатор конечной точки (точек) в шлюзе, где исполняется команда NotificationRequest. Идентификатор EndpointId ДОЛЖЕН подчиняться правилам для имен конечных точек, описанных в п. 6.1.1. Подстановочный знак "any of" ("любой из") НЕ ДОЛЖЕН использоваться. Встроенный клиент, получающий NotificationRequest с условным подстановочным знаком "любой из", ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена). Подстановочный знак "all of" ("все из") ДОЛЖЕН поддерживаться для команд NotificationRequest, где каждый из параметров RequestedEvents (запрошенное событие), SignalRequests (запрос сигнала), DigitMap (отображение цифр) и DetectEvent (событие распознавания) является либо пустым, либо пропущенным. Для простоты ряд шлюзов может принять решение не поддерживать подстановочный знак "all of" для команд NotificationRequest, где один или несколько из этих параметров не являются ни пустыми, ни пропущенными. Такие шлюзы, если они получают команду NotificationRequest с подстановочным знаком "all of", должны посылать в ответе код ошибки 503, так как они не могут их обрабатывать по вышеуказанной причине.

Параметр **NotifiedEntity** (уведомленный объект) – это необязательный параметр, который определяет новый "уведомленный объект" для конечной точки. При его использовании ДОЛЖНО быть определено полное имя агента вызова, включающее как локальное имя, так и доменное имя, даже если для доменного имени используется заключенный в скобки IP-адрес<sup>7</sup>. Более подробная информация содержится в пп. 6.1.1 и 6.1.4. Однако если представлено только доменное имя, адаптер МТА ДОЛЖЕН использовать доменное имя как ID агента вызова.

Параметр **RequestIdentifier** (идентификатор запроса) используется для корреляции данного запроса с уведомлением, которое этот запрос может инициировать. Он будет повторен в соответствующей команде Notify (уведомить).

Параметр **SignalRequests** (запросы сигналов) – это параметр, содержащий множество сигналов, которые подаются шлюзом по запросу. Если не оговорено другого, сигналы подаются в конечную

<sup>5</sup> Соединение в режиме "копирование" может быть использовано, например, для поддержки "проверки занятости линии" с минимальным влиянием на ресурсы встроенного клиента.

<sup>6</sup> Этой новой командой будет команда ModifyConnection (модифицировать соединение).

<sup>7</sup> Использование IP-адреса в параметре NotifiedEntity не разрешается Рекомендацией по обеспечению безопасности. При реализации этой Рекомендации должно использоваться абсолютное доменное имя (включая имя хоста).

точку; однако некоторые сигналы могут подаваться в соединении. Примерами сигналов могут служить следующие сигналы<sup>8</sup>:

- сигнал вызова;
- тональный сигнал занятости;
- тональный сигнал ожидания вызова;
- тональный предупредительный сигнал снятой трубки;
- тональные сигналы контроля посылки вызова в соединении.

В зависимости от режима использования сигналы делятся на различные типы:

- Включено/выключено (On/off (OO)) – Поданные один раз, они делятся, пока не будут отключены. Это может произойти только в результате появления нового параметра SignalRequests, в котором сигнал отключен (см. далее). Сигналы типа OO определены как идемпотентные; таким образом, множество запросов включить (или выключить) данный сигнал OO являются абсолютно действительными и НЕ ДОЛЖНЫ приводить ни к каким ошибкам. Сигнал "включено/выключено" может быть визуальным индикатором ожидания сообщения (VMWI). Будучи включенным, сигнал OO НЕ ДОЛЖЕН быть выключен, пока от агента вызова не поступит явная команда сделать это; сигналы OO будут отключены после перезапуска конечной точки. Пропущенный или пустой параметр SignalRequests, или же параметр SignalRequests, при котором опускается отдельный сигнал, не является явной командой для сигнала OO; он не будет изменять состояние адаптера МТА для сигнала OO.
- Выдержка времени (Time-out (TO)) – Поданные один раз, эти сигналы делятся, пока либо не будут аннулированы (в результате наступления некоторого события или в результате невключения в последующий [возможно, пустой] список сигналов), либо пока не истечет характерный для сигнала период времени. Сигнал, выдержка времени которого истекла, будет генерировать событие "операция завершена" (дальнейшее описание этого события содержится в Приложении А). Сигналом TO может быть тональный сигнал "контроля посылки вызова" с выдержкой времени 180 секунд. Если событие происходит до истечения 180 секунд, то по умолчанию сигнал прекратится<sup>9</sup>. Если сигнал не прекращается, то для него будет иметь место выдержка времени, сигнал прекращается и генерируется событие "операция завершена", об уведомлении о котором агент вызова мог послать или не послать запрос. Если агент вызова послал запрос об уведомлении о событии "операция завершена", то событие "операция завершена", посланное агенту вызова, будет содержать имя сигнала (имена сигналов) с выдержкой времени<sup>10</sup>. Сигнал (сигналы), генерируемый в соединении, будет содержать имя этого соединения. Сигналы выдержки времени имеют определенное для них значение выдержки времени по умолчанию, которое может быть изменено в процессе инициализации. Кроме того, период выдержки времени может быть предоставлен сигналу в качестве параметра. Нулевое значение указывает на то, что период выдержки времени бесконечен. Сигнал TO, в котором происходит сбой после запуска, но до генерирования события "операция завершена", сгенерирует событие "сбой операции", содержащее имя сигнала (имена сигналов), время подачи которых истекло<sup>10</sup>.
- Короткий сигнал (Brief (BR)) – Длительность этих сигналов так мала, что они прекращаются сами. Если имеет место событие прекращения сигнала или поступает новый набор сигналов согласно параметру SignalRequests, текущий активный сигнал BR прекращаться не будет. Однако все ожидающие сигналы BR, еще не поданные, будут аннулированы. Короткий тональный сигнал может соответствовать цифре двухтонального многочастотного (DTMF) набора номера. Если в текущий момент передается цифра "1" DTMF и появляется событие прекращения сигнала, то передача сигналов цифры "1" закончится.

По умолчанию сигналы относятся к конечным точкам. Если сигнал, поданный в конечную точку, приводит к генерированию медиапотока (аудио, видео и т. д.), то этот поток НЕ ДОЛЖЕН направляться по любому соединению, связанному с этой конечной точкой, независимо от режима соединения. Например, если тональный сигнал ожидания вызова подается в конечную точку, вовлеченную в активный вызов, то только сторона, использующая рассматриваемую конечную точку, будет слышать этот тональный сигнал ожидания вызова. Однако отдельные сигналы могут вести себя по-разному.

<sup>8</sup> Полный список сигналов содержится в Дополнении VII.

<sup>9</sup> Действие "удерживание сигнала(ов) активным(и)" может отменять такое поведение.

<sup>10</sup> Если параметры были посланы сигналу, то о них не будет сообщаться.

Если сигнал подается в соединение, которое получило параметр RemoteConnectionDescriptor (см. п. 6.3.3), то медиапоток, генерируемый этим сигналом, ДОЛЖЕН направляться по соединению, для всех режимов за исключением "сетевого шлейфа". Если параметр RemoteConnectionDescriptor получен не был, то шлюз ДОЛЖЕН вернуть информацию об ошибке (код ошибки 527 – отсутствие параметра RemoteConnectionDescriptor).

Когда предоставляется список (возможно, пустой) сигнала(ов), этот список полностью заменяет текущий список активных сигналов выдержки времени. Сигналы, не представленные в новом списке, ДОЛЖНЫ быть прекращены, а новые представленные сигналы (сигнал) теперь становятся активными. Текущие активные сигналы выдержки времени, которые представлены в новом списке сигналов, ДОЛЖНЫ оставаться активными без прерывания; таким образом, таймер таких сигналов выдержки времени затронут не будет. Следовательно, в настоящее время нет способа перезапуска таймера для активного в текущее время сигнала выдержки времени без первоначального выключения сигнала. Если сигнал выдержки времени параметризован, то исходное множество параметров ДОЛЖНО оставаться в силе, независимо от того, какие значения предоставляются в дальнейшем. Данный сигнал НЕ ДОЛЖЕН появляться более одного раза в параметре SignalRequests. Пропуск параметра SignalRequests интерпретируется как пустой список параметров SignalRequests.

Определенные на настоящее время сигналы представлены в Приложении А.

Параметр **RequestedEvents** – это список событий, которые по запросу должен обнаруживать шлюз в конечной точке. Если не оговорено другого, события обнаруживаются в конечной точке; однако некоторые события могут обнаруживаться в соединении. Примерами событий являются<sup>11</sup>:

- тональные сигналы при факсимильной связи;
- модемные тональные сигналы;
- переход в состояние "телефонная трубка положена" (имеет место в классических телефонных аппаратах, когда пользователь кладет телефонную трубку на аппарат);
- переход в состояние "телефонная трубка снята" (имеет место в классических телефонных аппаратах, когда пользователь поднимает телефонную трубку);
- флэш-сигналы (имеют место в классических телефонных аппаратах, когда пользователь быстро нажимает на рычаг аппарата, который удерживает телефонную трубку);
- цифры сигнала DTMF (или импульсные знаки).

Определенные на настоящее время события представлены в Приложении А.

С каждым событием связано одно или несколько **действий**, определяющих образ действия, которое ДОЛЖЕН предпринять шлюз, когда происходит рассматриваемое событие. Возможными действиями являются:

- немедленное уведомление о событии наряду с передачей накопленного списка наблюдаемых событий;
- накопление событий;
- накопление согласно отображению цифр;
- игнорирование события;
- удержание сигнала(ов) активным(и);
- встроенная команда NotificationRequest;
- встроенная команда ModifyConnection.

Конечная точка будет обнаруживать два множества запрашиваемых событий – устойчивых и неустойчивых.

Устойчивые события всегда обнаруживаются в конечной точке. Если устойчивое событие не включено в список параметра RequestedEvents, а это событие происходит, то оно в любом случае будет обнаружено и обработано подобно всем другим событиям, как если бы это устойчивое событие было запрошено действием "уведомление"<sup>12</sup>. Таким образом, устойчивые события можно неформально рассматривать как всегда включенные неявным образом в список параметра RequestedEvents с действием "уведомление", хотя обнаружение помехи приему вызова звонком

---

<sup>11</sup> Это просто примеры, взятые из примера линейного пакета в Дополнении I.

<sup>12</sup> Таким образом, параметр RequestIdentifier будет параметром RequestIdentifier команды NotificationRequest.

выполняться не будет<sup>13</sup>. Устойчивые события идентифицируются как таковые по определению – см. Приложение А.

Неустойчивые события – это события, которые должны быть явным образом включены в список параметра RequestedEvents. Этот список (возможно, пустой) запрашиваемых событий полностью заменяет предшествующий список запрашиваемых событий. В дополнение к устойчивым событиям конечная точка будет обнаруживать только события, которые описаны в списке запрашиваемых событий. Если устойчивое событие включено в список параметра RequestedEvents, тогда заданное действие заменит действие по умолчанию, связанное с событием, на время существования списка параметра RequestedEvents, после чего восстанавливается действие по умолчанию. Например, если было задано действие "проигнорировать сигнал снятия телефонной трубки" и был получен новый запрос без каких-либо команд снятия телефонной трубки, тогда будет восстановлена операция по умолчанию "уведомить о сигнале снятия телефонной трубки". Данное событие НЕ ДОЛЖНО появляться более одного раза в параметре RequestedEvents. Пропуск параметра RequestedEvents интерпретируется как пустой список параметра RequestedEvents.

Для события может быть определено несколько действий, хотя какое-либо заданное действие для данного события не может иметь место более одного раза. В приведенной ниже таблице определены допустимые сочетания действий:

**Таблица 2/J.162 – Допустимые сочетания действий**

	Уведомление	Накопление	Накопление согласно отображению цифр	Игнорирование	Удерживание сигнала(ов) активным(и)	Встроенный запрос на уведомление	Встроенная команда "модифицировать соединение"
Уведомление	–	–	–	–	√	–	√
Накопление	–	–	–	–	√	√	√
Накопление согласно отображению цифр	–	–	–	–	√	–	√
Игнорирование	–	–	–	–	√	–	√
Удерживание сигнала(ов) активным(и)	√	√	√	√	–	√	√
Встроенный запрос на уведомление	–	√	–	–	√	–	√
Встроенная команда "модифицировать соединение"	√	√	√	√	√	√	–

ПРИМЕЧАНИЕ. – Действие "встроенный запрос на уведомление" может сочетаться только с действием "уведомление", если шлюзу разрешено генерировать несколько команд Notify (уведомить) для каждого запроса на уведомление.

Если клиент получает запрос с неправильным действием или недопустимым сочетанием действий, он ДОЛЖЕН вернуть агенту вызова информацию об ошибке (код ошибки 523 – неизвестное или недопустимое сочетание действий).

Когда указано множество действий, например "удерживание сигнала(ов) активным(и)" или "уведомление", то предполагается, что отдельные действия выполняются одновременно.

<sup>13</sup> Обычно, если сделан, например, запрос на поиск сигнала снятия телефонной трубки, этот запрос будет успешным, только если телефон еще не находится в состоянии со снятой телефонной трубкой.

Агент вызова может послать шлюзу команду NotificationRequest с пустым списком параметра RequestedEvents. Агент вызова может так поступать, например, по отношению к встроенному клиенту, когда он "не желает" накапливать дальше цифры двухтонального многочастотного (DTMF) набора номера. Однако устойчивые события все еще будут обнаруживаться, и о них будут посылаться уведомления.

Когда обнаруживается возбудитель, инициирующий множественные запрошенные события, (например, тональный сигнал факсимильной связи – это возбудитель для обоих FXR/gwfax(start), и L/ft), шлюз ДОЛЖЕН генерировать только одно из событий (а именно наиболее предпочтительное событие из множества запрошенных событий) основываясь на следующих правилах приоритета:

- 1) В событиях, включенных в список RequestedEvents, приоритеты расставляются справа налево, событие с наивысшим приоритетом помещено слева.
- 2) Устойчивые события, не включенные в списки RequestedEvents обладают меньшим приоритетом, чем события (как устойчивые, так и нет), включенные в список RequestedEvents. Жестко установленного регламента расстановки приоритетов для устойчивых событий, не включенных в список RequestedEvents, не существует.

Параметр **DigitMap** (отображение цифр набора номера) – это необязательный параметр, позволяющий агенту вызова предоставлять конечной точке отображение цифр набора номера, согласно которому будут накапливаться цифры, когда агент вызова предоставляет для этой конечной точки параметр RequestedEvents с действием "накопление цифр согласно отображению цифр". Предоставляемое отображение цифр носит устойчивый характер, и поэтому нет необходимости предоставлять его каждый раз, когда делается запрос "накопление цифр согласно отображению цифр", однако агенты вызовов могут предоставить отображение цифр в любое время. Отображение цифр ДОЛЖНО быть предоставлено для конечной точки с первым запросом "накопление цифр согласно отображению цифр". Если шлюз получает запрос об этом и в текущий момент не имеет отображения цифр для рассматриваемой конечной точки, то он ДОЛЖЕН вернуть информацию об ошибке (код ошибки 519 – конечная точка не имеет отображения цифр).

Каждая конечная точка имеет переменную, называемую "текущая строка набора номера", где накапливаются цифры для сравнения с отображением цифр, как описано в п. 6.1.5. Каждый раз, когда посылается команда Notify или должна обрабатываться команда NotificationRequest, переменная "текущая строка набора номера" инициализируется нулевой строкой. Цифры, подлежащие обработке, могут теперь либо распознаваться как входные, либо могут быть считаны из области промежуточного хранения событий на входе, известной как "карантинный буфер": см. более подробную информацию в п. 6.1.3.1.

Сигналы, используемые в параметре SignalRequests (запросы сигналов), синхронизированы с набором событий, описанных или подразумеваемых в параметре RequestedEvents, за исключением случая отмены посредством действия "удерживание сигнала(ов) активным(и)". Например, если команда NotificationRequest управляла "вызывным" сигналом и в запросе события содержалась просьба отыскать событие "снятие телефонной трубки", то вызывной сигнал должен прекратиться по умолчанию, как только шлюз обнаружит такое событие. Если в запросе события не содержалась просьба отыскать событие "снятие телефонной трубки", то вызывной сигнал прекратится в любом случае, поскольку снятие телефонной трубки – это устойчивое событие, и поэтому его наличие подразумевается в параметре RequestedEvents. Формальное определение состоит в том, что генерирование всех сигналов "выдержки времени" ДОЛЖНО прекратиться, как только обнаружится одно из запрашиваемых событий, если только действие "удерживание сигнала(ов) активным(и)" не будет связано с описываемым событием. В случае действия "накопление цифр согласно отображению цифр" поведение по умолчанию будет заключаться в прекращении всех активных сигналов выдержки времени, когда первая цифра<sup>14</sup> заносится в накопитель: это несущественно для данной синхронизации, если накапливаемые цифры приводят к соответствию, несоответствию или частичному соответствию отображению цифр.

Если требуется, чтобы сигнал (сигналы) выдержки времени продолжался при наступлении искомого события, может быть использовано действие "удерживание сигнала(ов) активным(и)". Это действие имеет эффект удерживания всех сигналов выдержки времени активными, отрицая прекращение по умолчанию сигналов выдержки времени при наступлении события.

Если требуется начать передачу сигнала (сигналов), когда наступает искомое событие, может быть использовано действие "Embedded NotificationRequest" ("встроенный запрос на уведомление"). Это действие может включать новый список параметров RequestedEvents, SignalRequests, а также новое

---

<sup>14</sup> Понятие "цифра" соответствует определениям в отображениях цифр, то есть сюда входят "звездочка", таймер и т. д.

отображение цифр набора номера. Семантика встроенной команды NotificationRequest такая, как если бы только что была принята новая команда NotificationRequest с теми же параметрами NotifiedEntity, RequestIdentifier, QuarantineHandling (карантинная обработка) и DetectEvents. Когда действие "Embedded NotificationRequest" активизировано, "текущая строка набора номера" будет освобождена; однако список наблюдаемых событий и карантинный буфер будут оставаться без изменений (в сочетании с командой Notify эта команда очистит весь список параметра ObservedEvents (наблюдаемые события), см. п. 6.4.3.1). Следует отметить, что при встроенном действии NotificationRequest иницирующие события не накапливаются; чтобы достичь этого, оно может быть объединено с действием "накопление". Реализации NCS ДОЛЖНЫ иметь возможность поддержки по крайней мере одного уровня встраивания. Встроенная команда NotificationRequest, которая подчиняется этому ограничению, НЕ ДОЛЖНА содержать другую встроенную команду NotificationRequest.

Встроенное действие NotificationRequest позволяет агенту вызова установить "мини-сценарий", подлежащий исполнению шлюзом сразу же после обнаружения связанного с ним события. Все команды SignalRequests, описанные во встроенном действии NotificationRequest, начинают выполняться немедленно. Значительное внимание должно быть уделено предотвращению несоответствий между агентом вызова и шлюзом. Однако долговременные несоответствия не должны иметь места, поскольку новые параметры SignalRequests полностью заменяют старый список активных сигналов выдержки времени, а сигналы типа BR всегда прекращаются сами. Рекомендуется ограничивать число сигналов типа "включено/выключено". Считается целесообразным, чтобы агент вызова время от времени включал все сигналы типа "включено/выключено", которые должны быть включены, и выключал все сигналы этого типа, которые должны быть выключены.

Если требуется изменить режимы соединения, когда наступают искомые события, то может быть использовано действие "Embedded ModifyConnection" ("встроенная модификация соединения"). Это действие может включать список изменений режимов соединений, каждое из которых содержит изменение режима и идентификатор рассматриваемого соединения. Подстановочный знак "\$" может быть использован для обозначения "текущего соединения"; однако это обозначение НЕ ДОЛЖНО использоваться вне команды на обработку соединения: подстановочный знак относится к рассматриваемому соединению для команды на обработку соединения.

Действие "Embedded ModifyConnection" позволяет агенту вызова дать команду конечной точке изменить режим соединения для одного или нескольких соединений непосредственно за обнаружением связанного с ними события. Каждое из изменений режима соединения действует подобно соответствующей команде ModifyConnection<sup>15</sup>. Когда предоставляется список изменений режимов соединений, изменения режимов соединений ДОЛЖНЫ применяться по одному в порядке слева направо. Когда все изменения режимов соединений заканчиваются, генерируется событие "операция завершена", параметризованное именем завершеного действия (подробная информация содержится в Приложении А). Как только при каком-либо изменении режима соединения происходит ошибка, генерируется событие "ошибка операции", параметризованное именем ошибочного действия и изменением режима соединения (подробная информация содержится в Приложении А). НЕ ДОЛЖНЫ предприниматься попытки выполнить остальные изменения режимов соединений, а предшествующие успешные изменения режимов соединений в списке НЕ ДОЛЖНЫ оставаться действующими.

Наконец, действие Ignore (игнорирование) может быть использовано для игнорирования события, например для предотвращения уведомления об устойчивом событии. Однако синхронизация между событием и активным сигналом будет по-прежнему проводиться по умолчанию.

*Пункт 6.4.3.1 содержит дополнительную подробную информацию по семантике обнаружения событий и сообщениям. Читателю рекомендуется тщательно изучить этот пункт.*

Конкретное определение действий, запрашиваемых через параметры SignalRequests (например, длительность и частота цифры при наборе номера DTMF), выходит за рамки базовой спецификации NCS. Это определение может меняться в зависимости от местоположения и, следовательно, может меняться от шлюза к шлюзу. Значит, определения предоставляются в пакетах событий, которые могут быть обеспечены вне базовой спецификации. Начальный список пакетов событий содержится в Приложении А.

Параметры RequestedEvents и SignalRequests, как правило, относятся к одним и тем же событиям. В одном случае шлюз запрашивает обнаружить появление события, а в другом случае – генерировать

---

<sup>15</sup> Так, если, например, в соединении используется D-QoS, то действие D-QoS по умолчанию все еще будет предприниматься, когда будет выполнено встроенное действие ModifyConnection.

это событие. Из этого правила имеются исключения; например, модемные тональные сигналы или тональные сигналы при факсимильной связи, которые могут быть обнаружены, но о которых не сообщается. Однако нельзя ожидать, что во всех конечных точках обязательно будут обнаружены все события. Конкретные события и сигналы, которые может обнаружить или произвести данная конечная точка, определяются списком пакетов событий, поддерживаемых этой конечной точкой. Каждый пакет описывает список событий и сигналов, которые могут быть обнаружены или применены. Шлюз, который запрашивается об обнаружении или применении события, не поддерживаемого заданной конечной точкой, ДОЛЖЕН возвращать информацию об ошибке (код ошибки 512 или 513 – отсутствие оборудования для обнаружения события или генерирования сигнала). Когда имя события не уточняется именем пакета, принимается определяемое по умолчанию имя пакета для конечной точки. Если имя события не зарегистрировано в этом пакете по умолчанию, то шлюз ДОЛЖЕН вернуть информацию об ошибке (код ошибки 522 – отсутствие такого события или сигнала).

Агент вызова может послать команду NotificationRequest, запрашиваемый список сигналов которой является пустым. Действие этой команды заключается в прекращении всех активных сигналов выдержки времени. Она может сделать это, например, когда должно быть остановлено генерирование тональных сигналов, например сигнала контроля посылки вызова.

Параметр **QuarantineHandling** (карантинная обработка) – это необязательный параметр, который описывает варианты обработки для событий в карантинном буфере (см. п. 6.4.3.1), то есть событий, которые были обнаружены шлюзом до поступления команды NotificationRequest, но уведомление о которых еще не было послано агенту вызова. Этот параметр обеспечивает следующее множество вариантов обработки:

- должны ли карантинные события обрабатываться или отбрасываться (по умолчанию они должны обрабатываться),
- будет ли шлюз в ответ на этот запрос генерировать максимум одно уведомление (шаг блокировки) или множество уведомлений (цикл) (по умолчанию максимум одно уведомление).

Когда этот параметр отсутствует, карантинные события ДОЛЖНЫ обрабатываться. Поддержка режима "шаг блокировки" (по умолчанию) и режима "цикл" обязательна. Конечной точке, принимающей команду NotificationRequest с неподдерживаемым значением параметра QuarantineHandling, СЛЕДУЕТ ответить кодом ошибки 508 (неподдерживаемый параметр QuarantineHandling).

Следует отметить, что параметр карантинной обработки также управляет обработкой событий, которые были обнаружены и обработаны, но о которых еще не было уведомлено, когда принималась команда.

Параметр **DetectEvents** (обнаруживаемые события) – это необязательный параметр, описывающий минимальный список событий, которые должен обнаружить по запросу шлюз в состояниях "уведомление" и "шаг блокировки". Когда этот параметр отсутствует, то событиями, которые ДОЛЖНЫ быть обнаружены в течение карантинного периода, являются события, перечисленные в последнем полученном списке параметра DetectEvents. Кроме того, шлюз ДОЛЖЕН обнаруживать устойчивые события и события, описанные в списке параметра RequestedEvents, включая те события, для которых задано действие "игнорирование". Дальнейшие пояснения относительно данного параметра можно найти в п. 6.4.3.1.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

### 6.3.2 Уведомления

Уведомления посылаются шлюзом с помощью команды Notify, когда необходимо передать уведомление о наблюдаемом событии:

```
ReturnCode  
← Notify(EndpointId  
        [, NotifiedEntity]  
        , RequestIdentifier  
        , ObservedEvents)
```

Параметр **EndpointId** (идентификатор конечной точки) – это имя для конечной точки в шлюзе, которая генерирует команду Notify, как определено в п. 6.1.1. Этот идентификатор ДОЛЖЕН точно определяться именем конечной точки, включая доменное имя шлюза. Локальная часть этого имени НЕ ДОЛЖНА использовать условный подстановочный знак. Агент вызова, получающий уведомление с

условным подстановочным знаком "любой из", ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Параметр **NotifiedEntity** (уведомленный объект) – это необязательный параметр, который идентифицирует объект, которому послано уведомление. Данный параметр равнозначен параметру **NotifiedEntity** команды **NotificationRequest**, инициировавшей это уведомление. Следует отметить, что адаптер МТА МОЖЕТ содержать только абсолютное доменное имя (включая имя хоста) своего параметра **NotifiedEntity**, если в инициирующей команде **NotificationRequest** было получено только абсолютное доменное имя. В этом случае системе CMS СЛЕДУЕТ принять это значение. Параметр отсутствует, если такого параметра не было в инициирующем запросе. Независимо от значения параметра **NotifiedEntity** текущему "уведомленному объекту" ДОЛЖНО быть послано уведомление для конечной точки.

Параметр **RequestIdentifier** (идентификатор запроса) – это параметр, повторяющий параметр **RequestIdentifier** команды **NotificationRequest**, инициировавшей данное уведомление. Он используется для корреляции этого уведомления с инициировавшим его запросом на уведомление. Устойчивые события будут рассматриваться здесь так, как будто они были включены в последнюю команду **NotificationRequest**. Если команда **NotificationRequest** не была получена, то используемый параметр **RequestIdentifier** будет равен нулю ("0").

Параметр **ObservedEvents** (наблюдаемые события) – это список событий, которые обнаружил и накопил шлюз с помощью действий "накопление", "накопление согласно отображению цифр" или "уведомление". В одном уведомлении может сообщаться список событий в порядке их обнаружения. Этот список может содержать только устойчивые события и события, которые были запрошены в параметре **RequestedEvents** инициирующей команды **NotificationRequest**. События, которые были обнаружены в соединении, будут содержать имя этого соединения. Список будет содержать события, которые были либо накоплены (но о которых не было послано уведомление), либо накоплены согласно отображению цифр (но еще без соответствия), и конечное событие, которое инициировало уведомление или обеспечило конечное соответствие в отображении цифр набора номера. Следует отметить, что цифры набора номера добавляются к списку наблюдаемых событий по мере их накопления независимо от того, накапливаются ли они согласно отображению цифр или нет. Например, если пользователь вводит цифры "1234", а некоторое событие E вводится в накопитель между цифрами "3" и "4", то списком наблюдаемых событий будет "1, 2, 3, E, 4".

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое при желании сопровождается комментарием.

### 6.3.3 Команда **CreateConnection**

Команда **CreateConnection** (создание соединения) используется для создания соединения.

```
ReturnCode
, ConnectionId
[, SpecificEndPointId]
, LocalConnectionDescriptor
[, ResourceID]
    ← CreateConnection(CallId
                        , EndpointId
                        [, NotifiedEntity]
                        , LocalConnectionOptions
                        , Mode
                        [, RemoteConnectionDescriptor]
                        [, RequestedEvents]
                        [, RequestIdentifier]
                        [, DigitMap]
                        [, SignalRequests]
                        [, QuarantineHandling]
                        [, DetectEvents])
```

Эта функция используется при установлении соединения между двумя конечными точками. Соединение определяется его атрибутами и конечными точками, которые оно связывает. Входные параметры в команде **CreateConnection** обеспечивают данные, необходимые для создания "видения" одной из двух конечных точек соединения.

Параметр **CallId** (идентификатор вызова) – это параметр, идентифицирующий вызов (или сеанс), которому принадлежит это соединение. Этот параметр является уникальным, как минимум, в наборе агентов вызовов, которые контролируют одни и те же шлюзы; соединения, принадлежащие одному и

тому же вызову, совместно используют один и тот же идентификатор вызова. Идентификатор вызова может быть использован для идентификации вызовов в целях передачи сообщений и проведения учета.

Параметр **EndPointId** (идентификатор конечной точки) – это идентификатор конечной точки в шлюзе, где выполняется команда `CreateConnection`. Параметр `EndPointId` может быть полностью описан путем присвоения этому параметру в вызове функции не подставляемого по знаку подстановки значения или может быть частично описан путем использования условного подстановочного знака "anyone" ("любой"). Если конечная точка описана частично, то идентификатор конечной точки будет присвоен шлюзом, а его полное значение ДОЛЖНО быть возвращено в параметре ответа **SpecificEndPointId** (идентификатор конкретной конечной точки) только в случае успешной команды. В этом случае конечная точка с присвоенным идентификатором ДОЛЖНА быть в рабочем состоянии и НЕ ДОЛЖНА иметь еще какие-либо соединения. Условный подстановочный знак "все" использоваться НЕ ДОЛЖЕН. Встроенный клиент, получающий `NotificationRequest` с условным подстановочным знаком "все", ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Параметр **NotifiedEntity** (уведомленный объект) – это необязательный параметр, который определяет новый "уведомленный объект" для конечной точки.

Структура **LocalConnectionOptions** (варианты локального соединения) – это структура, описывающая характеристики соединения для передачи медиаданных с точки зрения шлюза, выполняющего команду `CreateConnection`. Она передает команды конечной точке по характеристикам передачи и приема медиасоединения. Основными полями структуры `LocalConnectionOptions` являются:

- **Метод кодирования:** Список литеральных имен для алгоритма сжатия (метод кодирования/декодирования), используемого для передачи и приема медиаданных по соединению ДОЛЖЕН быть определен, по меньшей мере, одним значением. Элементы в списке упорядочены по уровню приоритета. Конечная точка ДОЛЖНА выбрать, по меньшей мере, один из кодеков, и выбирать его СЛЕДУЕТ согласно указанному приоритету. Если конечная точка принимает по соединению какие-либо медиаданные, закодированные с использованием другого метода кодирования, она МОЖЕТ отбросить его. Подробная информация, касающаяся процесса выбора кодека, содержится в п. 6.7.

ПРИМЕЧАНИЕ. – "Метод кодирования" включает аудио и видео кодировки и кодировку изображения.

- Конечная точка ДОЛЖНА дополнительно указывать, какие из оставшихся алгоритмов сжатия она хочет поддерживать в качестве альтернативы (см. подробную информацию в п. 7.4).
  - Список допустимых методов кодирования определен в Рекомендации МСЭ-Т J.161. ДОЛЖНЫ использоваться имена литералов, определенные в J.161 в таблице под названием "Параметры отображения кодеков RTP". Неизвестные алгоритмы сжатия в случае их получения СЛЕДУЕТ игнорировать.
- **Период пакетирования:** Одиночный период пакетирования в миллисекундах МОЖЕТ быть задан с одним десятичным значением. Если используется этот спецификатор, то для всех методов кодирования, допускаемых структурой `LocalConnectionOptions`, ДОЛЖЕН использоваться один и тот же период пакетирования. Следует отметить, что если в LCO не определено поле метода кодирования, адаптер МТА НЕ ДОЛЖЕН выбирать метод кодирования с периодом пакетирования, отличным от описанного здесь. Если для разных методов кодирования необходимы разные периоды пакетирования, это поле использоваться НЕ ДОЛЖНО. Значение относится как к переданным, так и принятым медиаданным. Следует отметить, что адаптер МТА должен использовать только действительный период пакетирования в сочетании с соответствующим методом (методами) кодирования. Список допустимых периодов пакетирования определен в Рекомендации МСЭ-Т J.161. Этот спецификатор НЕ ДОЛЖЕН предоставляться в одной структуре LCO с полем "период многократного пакетирования". Адаптер МТА ДОЛЖЕН возвращать информацию об ошибке (код ошибки 524 – несоответствие в структуре `LocalConnectionOptions`), когда он принимает структуру LCO как с полем "период пакетирования", так и с полем "период многократного пакетирования".
- **Период многократного пакетирования:** Список периодов пакетирования в миллисекундах МОЖЕТ быть задан, если и только если включено поле "метод кодирования". Будучи заданным, период многократного пакетирования в миллисекундах ДОЛЖЕН содержать одно десятичное значение или дефис для каждого элемента в поле "метод кодирования", включенном в структуру `LocalConnectionOptions`. Это условие применяется, даже если несколько методов кодирования имеют одно и то же значение. Первый элемент в списке ДОЛЖЕН быть десятичным числом. Если используется дефис, то рассматриваемый метод

кодирования ДОЛЖЕН использовать тот же период пакетирования, что и тот из элементов в списке, который фактически содержит десятичное число, и, кроме того, кодек НЕ ДОЛЖЕН использовать более широкую полосу пропускания, чем та, которую использует этот другой элемент. Этот может, например, быть использовано для неречевых кодеков (например, в случае телефонного события или "комфортного шума"), которые используют тот же период пакетирования, что и речевой кодек, с которым они используются. Последующие элементы в списке периодов пакетирования ДОЛЖНЫ быть упорядочены согласно соответствующим методам кодирования. Значения относятся как к переданным, так и принятым медиаданным. Следует отметить, что адаптер МТА НЕ ДОЛЖЕН выбирать кодек с периодом пакетирования, который отличается от описанного здесь периода пакетирования. Следует отметить также, что адаптер МТА должен использовать только действительный период пакетирования в сочетании с соответствующим методом кодирования. Список допустимых периодов пакетирования определен в Рекомендации МСЭ-Т J.161. Этот спецификатор НЕ ДОЛЖЕН предоставляться в одной структуре LCO с полем "период пакетирования". Адаптер МТА ДОЛЖЕН возвращать информацию об ошибке (код ошибки 524 – несоответствие в структуре LocalConnectionOptions) при следующих условиях:

- когда он принимает структуру LCO как с полем "период пакетирования", так и с полем "период многократного пакетирования";
  - когда он принимает структуру LCO, где число кодеков, заданных в поле "метод кодирования", отличается от числа элементов в поле "период многократного пакетирования".
- **Компенсация эха:** Должна ли первоначально использоваться компенсация эха на линейной стороне или нет<sup>16</sup>. Этот параметр может иметь значение "on" ("включено") (когда запрошена компенсация эха) или "off" ("выключено") (когда компенсация эха выключена). Этот параметр является необязательным. Если этот параметр опущен, встроенный клиент ДОЛЖЕН применять компенсацию эха изначально. Встроенному клиенту СЛЕДУЕТ в дальнейшем включать и выключать компенсацию эха в соответствии с Рекомендациями МСЭ-Т V.8 и V.25 при обнаружении данных в диапазоне звуковых частот. Относительно повторного включения компенсации эха см., например, Рекомендацию МСЭ-Т G.168. При окончании передачи данных в диапазоне звуковых частот обработка сигналов компенсации эха ДОЛЖНА возвращаться к текущему значению параметра компенсации эха. РЕКОМЕНДУЕТСЯ оставлять управление компенсацией эха за встроенным клиентом, а не за агентом вызова.
  - **Тип услуги:** Описывает класс услуги, которая будет использоваться для передачи медиаданных по соединению, путем кодирования 8-битового значения параметра "тип услуги" IP-заголовка в виде двух шестнадцатеричных цифр. Этот параметр является необязательным. Если этот параметр опущен, то ДОЛЖНО использоваться значение по умолчанию 0x00 (если не будет установлено другое значение). Если этот параметр представлен и имеет допустимое значение, полученное в нем значение должно быть использовано встроенным клиентом для заполнения параметра точка кода дифференцированной услуги (DSCP) в IP-заголовке (см. RFC 2474 для более детальной информации о DSCP). Значение параметра ДОЛЖНО быть 0x00 или кратным четырем в диапазоне от 0x01 до 0xFF (биты 6 и 7, биты ECN, зарезервированы и поэтому должны быть установлены на 00). Адаптер МТА ДОЛЖЕН возвращать сообщение об ошибке (код ошибки 532 – Неподдерживаемое значение(я) в LocalConnectionOptions), если он получает недопустимое значение. Крайний слева "бит" параметра соответствует наиболее значимому биту в IP-заголовке.
  - **Подавление пауз:** Встроенный клиент может осуществлять определение активности передачи речи и избегать пересылки пакетов во время периодов паузы. Однако для определенных вызовов (например, модемных) необходимо отключение подавления пауз. Данный параметр может иметь значение "on" ("включено") (когда пауза должна подавляться) или "off" ("выключено") (когда пауза не должна подавляться). Этот параметр является необязательным. Когда этот параметр опущен, по умолчанию устанавливается значение "выключено". Если стоит значение "включено" при обнаружении данных в диапазоне звуковых частот, конечной точке СЛЕДУЕТ отключить подавление пауз. При окончании передачи данных в диапазоне звуковых частот управление подавлением пауз ДОЛЖНО возвращаться к текущему значению параметра подавления пауз.

Для поддержки динамического качества обслуживания (D-QoS) используются следующие поля структуры LocalConnectionOptions (более подробная информация содержится в Приложении В):

- **D-QoS GateID** (идентификатор логического переключателя для динамического качества обслуживания): GateID для логического переключателя, который был установлен в CMTS.

---

<sup>16</sup> Компенсация эха со стороны пакетов не поддерживается.

GateID – это 32-битовый идентификатор, закодированный в виде строки, содержащей до 8 шестнадцатеричных знаков. Обычно этот параметр является необязательным, но когда должны выполняться резервирование и/или предоставление ресурсов для D-QoS, он становится обязательным. Наличие данного параметра означает, что для этой команды ДОЛЖНО обеспечиваться D-QoS. Отсутствие данного параметра означает, что D-QoS НЕ ДОЛЖНО предоставляться.

- **Резервирование ресурсов для D-QoS:** Позволяет явным образом контролировать, должно или не должно выполняться резервирование и/или предоставление ресурсов для D-QoS в направлении передачи и/или приема. Этот параметр является необязательным и может принимать одно или несколько следующих значений:

Значения для резервирования ресурсов:

- "SendReserve" Ресурсы резервируются только в направлении передачи.
- "ReceiveReserve" Ресурсы резервируются только в направлении приема.
- "SendReceiveReserve" Ресурсы резервируются в направлениях передачи и приема.

Значения для предоставления ресурсов:

- "SendCommit" Ресурсы предоставляются только в направлении передачи.
- "ReceiveCommit" Ресурсы предоставляются в направлении приема.
- "SendReceiveCommit" Ресурсы предоставляются в направлениях передачи и приема.

Этот параметр является необязательным, и несколько значений разделяются запятыми. Когда должно обеспечиваться D-QoS, а этот параметр опущен, то резервирование ресурсов ДОЛЖНО выполняться как в направлении передачи, так и в направлении приема. Резервируемые ресурсы определяются параметрами кодирования, примененными к соединению, то есть методом кодирования, периодом пакетирования, подавлением пауз, шифрокомплексом и т. д. Внешние параметры, такие как использование подавления заголовка полезной нагрузки, могут также влиять на объем зарезервированных ресурсов (см. подробную информацию в Рекомендации МСЭ-Т J.163).

Ресурсы в направлении приема могут быть зарезервированы и предоставлены без получения параметра RemoteConnectionDescriptor, в то время как ресурсы в направлении передачи могут быть зарезервированы, но не будут предоставлены до получения параметра RemoteConnectionDescriptor. Следует отметить, что пока не будет получен параметр RemoteConnectionDescriptor, зарезервированные и предоставленные ресурсы должны основываться на кодеке (кодеках), выбранном локально (на месте). По получении параметра RemoteConnectionDescriptor список кодеков, которые действительно могут быть использованы для передачи, может содержать подмножество этих кодеков. Однако список кодеков, которые могут быть использованы для приема, остается неизменным, пока конечная точка не генерирует новый параметр LocalConnectionDescriptor. Если должно выполняться резервирование для D-QoS, а параметр опущен, по умолчанию ресурсы ДОЛЖНЫ предоставляться на основе режима соединения, как указано в таблице 3.

**Таблица 3/J.162 – Значения по умолчанию для резервирования ресурсов**

Режим соединения	D-QoS
"неактивный"	Не предоставлять
"только передача", "копирование"	Предоставить в направлении передачи
"только прием"	Предоставить в направлении приема
"передача/прием", "конференция", "сетевой шлейф", "проверка целостности сети"	Предоставить в направлениях передачи и приема

Если требуется другая операция предоставления ресурсов, то выдается соответствующее значение для такого предоставления ресурсов, которое будет использоваться вместо предыдущего значения. Если должна быть выполнена операция предоставления ресурсов, но резервирование не было осуществлено, или если существующее резервирование не полностью соответствует ресурсам, которые должны быть предоставлены<sup>17</sup>, то резервирование будет выполняться автоматически. Если

<sup>17</sup> Это не возможно для команды CreateConnection, но приводится здесь для полноты изложения. Однако это возможно для команды ModifyConnection (см. п. 6.3.4).

резервное значение задано, но значение для предоставления не определено, то операция предоставления ресурсов не будет выполняться.

- **Параметр ResourceID** (идентификатор ресурса): Существующий ResourceID для уже зарезервированных ресурсов в краевом маршрутизаторе. Использование параметра ResourceID позволяет зарезервировать один и тот же ресурс при отдельных видах резервирования; однако в конкретный момент времени может быть активным только один из видов резервирования. ResourceID – это 32-битовый идентификатор, закодированный в виде строки, содержащей до 8 шестнадцатеричных знаков. Этот параметр является необязательным. Однако он ДОЛЖЕН использоваться встроенным клиентом при резервировании ресурсов, если предоставлен агентом вызова.
- **Параметр ReserveDestination** (адресат резерва): Этот необязательный параметр может описывать адрес IPv4, на необязательной основе сопровождаемый двоеточием и номером порта UDP, то есть адресата для резервирования ресурсов. Когда номер порта UDP не задан, то по умолчанию применяется значение 9. Параметр ReserveDestination обычно используется, когда должно выполняться резервирование ресурсов, а параметр RemoteConnectionDescriptor еще не предоставлен для соединения. Это дает возможность зарезервировать ресурсы и предоставить их в нисходящем направлении передачи для краевого маршрутизатора, когда еще неизвестен источник потока мультимедийных пакетов<sup>18</sup>. Когда параметр RemoteConnectionDescriptor предоставлен, параметр ReserveDestination игнорируется.

Для поддержки услуг обеспечения безопасности в проекте IPCablecom используются следующие поля структуры LocalConnectionOptions:

- **Поле RTP ciphersuite** (шифрокомплект по протоколу RTP): Список допустимых шифрокомплектов для обеспечения безопасности по протоколу RTP в приоритетном порядке. Элементы в этом списке упорядочены по приоритету, где предпочтительным является первый шифрокомплект. Конечная точка ДОЛЖНА выбрать один из шифрокомплектов согласно правилам, описанным в Рекомендации МСЭ-Т J.170. Конечной точке СЛЕДУЕТ дополнительно указать, какие из оставшихся шифрокомплектов она желает поддержать в качестве альтернатив (подробная информация содержится в п. 7.4.1). Каждый шифрокомплект представлен в виде строк в коде ASCII, состоящих из двух подстрок, разделенных наклонной чертой ("/"), где первая подстрока определяет алгоритм аутентификации, а вторая – алгоритм шифрования. Список допустимых шифрокомплектов приведен в Рекомендации МСЭ-Т J.170. Параметр шифрокомплекта по протоколу RTP применяется только к медиапотокам RTP. Если CMS включает в себя LocalConnectionOptions, который требует использования исключительно не-RTP медиа (как, например, T.38 факсимильное реле с использованием UDPTL), параметр шифрокомплекта по протоколу RTP в него включаться НЕ ДОЛЖЕН. Если LCO допускает использование как RTP медиа, так и не-RTP медиа, и параметр шифрокомплекта RTP в него включен, он применяется только для RTP медиа. В любом случае, если итоговый медиапоток соединения не-RTP медиапоток (как например T.38 факсимильное реле с использованием UDPTL), параметр шифрокомплекта по протоколу RTP ДОЛЖЕН быть пропущен, т. е. защита RTP не используется, и параметры защиты RTP в LocalConnectionDescriptor не включаются.
- **Поле RTCP ciphersuite** (шифрокомплект по протоколу RTCP): Список шифрокомплектов для обеспечения безопасности по протоколу RTCP в приоритетном порядке. Элементы в этом списке упорядочены по приоритету, где предпочтительным является первый шифрокомплект. Конечная точка ДОЛЖНА выбрать один из шифрокомплектов согласно правилам, описанным в Рекомендации МСЭ-Т J.170. Конечной точке СЛЕДУЕТ дополнительно указать, какие из оставшихся шифрокомплектов она желает поддержать в качестве альтернатив. Подробная информация содержится в п. 7.4.1. Каждый шифрокомплект представлен в виде строки в коде ASCII, состоящей из двух подстрок, разделенных наклонной чертой ("/"), где первая подстрока определяет алгоритм аутентификации, а вторая – алгоритм шифрования. Список допустимых шифрокомплектов должен быть описан в Рекомендации МСЭ-Т J.170. Параметр шифрокомплекта по протоколу RTCP применяется в RTCP только для медиапотоков RTP. Если CMS включает в себя LocalConnectionOptions, который требует использования исключительно не-RTP медиа (как, например, T.38 факсимильное реле с использованием UDPTL), параметр шифрокомплекта по протоколу RTCP в него включаться НЕ ДОЛЖЕН. Если LCO допускает использование как RTP медиа, так и не-RTP медиа, и параметр

---

<sup>18</sup> Следует отметить, что это будет давать возможность использовать определенные сценарии "похищения услуг". Подробную информацию см. в Рекомендации МСЭ-Т J.163.

шифрокомплекта RTCP в него включен, он применяется в RTCP только для RTP медиа. В любом случае, если итоговый медиапоток соединения не-RTP медиапоток (как например T.38 факсимильное реле с использованием UDPTL), параметр шифрокомплекта по протоколу RTCP ДОЛЖЕН быть пропущен, т. е. защита RTCP не используется, и параметры защиты RTCP в LocalConnectionDescriptor не включаются.

Встроенный клиент ДОЛЖЕН отвечать информацией об ошибке (код ошибки 542 – несоответствие структуры LocalConnectionOptions), если нарушено любое из вышеприведенных правил. Все упомянутые выше значения по умолчанию могут быть изменены во время процесса подготовки к работе.

**Параметр RemoteConnectionDescriptor** (удаленный дескриптор соединения) – это дескриптор соединения для удаленной стороны соединения с другой стороны IP-сети. Он содержит те же поля, что и параметр LocalConnectionDescriptor (не путать со структурой LocalConnectionOptions), то есть поля, которые описывают сеанс в соответствии со стандартом SDP. В п. 7.4 содержится подробная информация по поддерживаемому использованию протокола SDP в профиле NCS. Этот параметр может иметь нулевое значение, если информация для удаленного конца соединения неизвестна. Это происходит, потому что объект, создающий соединение, начинает с передачи команды CreateConnection одному из двух участвующих в этом процессе шлюзов. При генерировании первой команды CreateConnection отсутствует доступная информация о другой стороне соединения. Эта информация может быть предоставлена позднее через вызов команды ModifyConnection.

Когда происходит изменение кодеков во время вызова, могут существовать небольшие периоды времени, когда в конечных точках используются разные коды. Как отмечалось выше, встроенные клиенты МОГУТ отбрасывать любые принимаемые медиаданные, закодированные в кодеке, отличном от описанного в структуре LocalConnectionOptions кодека для соединения.

**Параметр Mode** (режим) указывает на режим операции для данной стороны соединения. Вариантами являются "только передача", "только прием", "передача/прием", "конференция", "неактивный", "копирование", "сетевой шлейф" или "проверка целостности сети". Управление этими режимами описано в начале п. 6.3. В некоторых конечных точках невозможна поддержка всех режимов. Если в команде задан режим, который не поддерживается конечной точкой, то ДОЛЖНО быть возвращено сообщение об ошибке (код ошибки 517 – неподдерживаемый режим). Кроме того, если соединение еще не получило параметр RemoteConnectionDescriptor, то ДОЛЖНО быть возвращено сообщение об ошибке, если предпринимается попытка ввести соединение в любой из режимов "только передача", "передача/прием", "копирование", "конференция", "сетевой шлейф" или "проверка целостности сети" (код ошибки 527 – отсутствие параметра RemoteConnectionDescriptor).

**Параметр ConnectionId** (идентификатор соединения) – это параметр, возвращаемый шлюзом, который однозначно определяет соединение в контексте рассматриваемой конечной точки. ConnectionId ДОЛЖЕН включаться в любой предварительный или успешный ответ на команду CreateConnection. ConnectionId НЕ ДОЛЖЕН включаться, если возвращается какое-либо сообщение об ошибке и соединение не образуется..

**Параметр LocalConnectionDescriptor** (локальный дескриптор соединения) – это параметр, возвращаемый шлюзом, который является описанием сеанса, содержащим информацию, например, об адресах и портах RTP для "интеллектуальных" соединений, как определено в протоколе SDP. Этот параметр подобен параметру RemoteConnectionDescriptor, за исключением того, что он описывает данную сторону соединения. В п. 7.4 содержится подробная информация о поддерживаемом использовании протокола SDP в профиле NCS. LocalConnectionDescriptor ДОЛЖЕН включаться в любой предварительный или успешный ответ на команду CreateConnection. LocalConnectionDescriptor НЕ ДОЛЖЕН включаться, если возвращается какое-либо сообщение об ошибке и соединение не образуется.

После приема команды CreateConnection, не содержащей параметр RemoteConnectionDescriptor, шлюз оказывается в "двусмысленной" ситуации относительно рассматриваемого соединения. Поскольку этот шлюз экспортировал параметр LocalConnectionDescriptor, он может в принципе принимать пакеты по этому соединению. Поскольку он еще не получил параметр RemoteConnectionDescriptor другого шлюза, ему неизвестно, было ли получено разрешение от агента вызова на пакеты, которые он принимает. Таким образом, он должен "лабиринтировать" между двумя опасностями, то есть урезанием ряда важных уведомлений или приемом ненадежных данных. Поведение шлюза определяется значением параметра mode (режим) (подлежащего защите):

- Если режим был установлен в состояние "только прием", то шлюз ДОЛЖЕН принимать речевые сигналы, полученные по соединению, и передавать их дальше конечной точке.

- Если режим был установлен в состояние "неактивный", то шлюз ДОЛЖЕН (как всегда) отбрасывать речевые сигналы, полученные по соединению.
- Следует отметить, что когда конечная точка не имеет параметра RemoteConnectionDescriptor для соединения, то по определению соединение не может находиться ни в одном из следующих режимов: "только передача", "передача/прием", "копирование", "конференция", "сетевой шлейф" или "проверка целостности сети".

Все параметры **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательными. Они могут быть использованы агентом вызова для эффективного включения запроса на уведомление, который выполняется одновременно с созданием соединения. Если присутствуют один или несколько из этих параметров, то одним из них ДОЛЖЕН быть параметр RequestIdentifier. Таким образом, включение запроса на уведомление может быть распознано по наличию параметра RequestIdentifier. Остальные параметры могут присутствовать или нет. Если один из параметров не присутствует, то это ДОЛЖНО интерпретироваться так, как если бы имела место обычная команда NotificationRequest с пропущенным рассматриваемым параметром. Это может иметь эффект аннулирования сигналов и прекращения поиска событий. Следует отметить, что если параметры RequestedEvents и SignalRequests опущены, соответствующие списки считаются пустыми только в том случае, если содержится параметр RequestIdentifier.

В качестве примера использования можно рассмотреть агента вызова, который хочет передать вызов встроенному клиенту. Агент вызова должен:

- запросить встроенного клиента создать соединение, чтобы удостовериться в том, что пользователь может начать говорить, как только будет снята трубка телефона;
- запросить встроенного клиента начать посылать вызывной сигнал;
- запросить встроенного клиента уведомить агента вызова о том, что снята телефонная трубка.

Все упомянутое выше может быть выполнено с помощью одной команды CreateConnection путем включения в нее запроса на уведомление с параметрами RequestedEvents для события "телефонная трубка снята" и с параметром SignalRequests для вызывного сигнала.

Когда эти параметры присутствуют, то создание соединения и запрос на уведомление ДОЛЖНЫ быть синхронизированы, что означает, что они оба либо приняты, либо отклонены. В нашем примере команда CreateConnection должна быть отклонена, если шлюз не имеет достаточных ресурсов или не может получить соответствующие ресурсы от системы локального доступа к сети. Запрос на уведомление о снятии телефонной трубки должен быть отклонен в состоянии помехи приему вызова звонком, если пользователь уже снял телефонную трубку. В этом примере телефон не должен посылать вызывной сигнал, если соединение не может быть установлено, и соединение не должно быть установлено, если пользователь уже снял телефонную трубку. Вместо этого будет возвращена информация об ошибке (код ошибки 401 – снята телефонная трубка), которая информирует агента вызова о состоянии помехи приему вызова звонком.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

Параметр **ResourceID** (идентификатор ресурса) – это параметр D-QoS, который может быть возвращен шлюзом. Следует отметить, что этот параметр ДОЛЖЕН быть возвращен адаптером МТА, когда согласно командам от агента вызова должно быть обеспечено D-QoS. В случае успешного резервирования ресурсов для D-QoS параметр ResourceID обеспечивает оперирование зарезервированными ресурсами. ResourceID НЕ ДОЛЖЕН включаться, если возвращается какое-либо сообщение об ошибке и соединение не образуется.

### 6.3.4 Команда ModifyConnection

Команда ModifyConnection (модификация соединения) используется для модификации характеристик "видения" соединения со стороны шлюза. Это "видение" соединения включает как локальный дескриптор соединения, так и удаленный дескриптор соединения.

```
ReturnCode
  [, LocalConnectionDescriptor]
  [, ResourceID]
    ← ModifyConnection(CallId
```

```

, EndpointId
, ConnectionId
[, NotifiedEntity]
[, LocalConnectionOptions]
[, Mode]
[, RemoteConnectionDescriptor]
[, RequestedEvents]
[, RequestIdentifier]
[, DigitMap]
[, SignalRequests]
[, QuarantineHandling]
[, DetectEvents])

```

Используются те же параметры, что и в команде `CreateConnection`, с добавлением параметра **ConnectionId** (идентификатор соединения), который однозначно определяет соединение в конечной точке. Это параметр возвращается командой `CreateConnection` вместе с локальным дескриптором соединения. Он однозначно определяет соединение в контексте конечной точки.

Параметр **EndpointId** (идентификатор конечной точки) ДОЛЖЕН точно определяться именем конечной точки. В локальном имени НЕ ДОЛЖЕН использоваться условный подстановочный знак. Встроенный клиент, получающий `ModifyConnection` с условным подстановочным знаком, ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Команда `ModifyConnection` может быть использована для того, чтобы влиять на параметры соединения, при соблюдении тех же правил и ограничений, которые определены для команды `CreateConnection`, в частности:

- Предоставлять информацию на другом конце соединения через параметр **RemoteConnectionDescriptor**.
- Активизировать и деактивизировать соединение путем изменения значения параметра **mode** (режим). Это может происходить в любое время в течение соединения при произвольных значениях параметров. Активизация может быть установлена, например, в режим "только прием".
- Изменять параметры соединения через структуру **LocalConnectionOptions**, например, путем переключения на другую систему кодирования, изменения периода пакетирования или модификации управления компенсацией.

Операция по обеспечению динамического качества услуги (D-QoS) была подробно описана в команде `CreateConnection`, и в целом здесь применяются те же правила, за исключением следующего:

- **D-QoS GateID** (идентификатор логического переключателя для динамического качества обслуживания): Идентификатор `GateID` для D-QoS является обязательным, когда требуется операция D-QoS, если только операция D-QoS не была выполнена ранее для рассматриваемого соединения. В последнем случае МТА ДОЛЖЕН использовать ранее предоставленный идентификатор D-QoS `GateID`.
- **D-QoS Resource Reservation** (резервирование ресурсов для D-QoS): Позволяет явным образом контролировать, должно ли выполняться или нет резервирование и/или предоставление ресурсов для D-QoS в направлении передачи и/или приема. Этот параметр является необязательным, и для него может быть определено несколько значений. Если этот параметр пропущен и должно выполняться резервирование для D-QoS, то по умолчанию должно выполняться резервирование как в направлении передачи, так и в направлении приема, если только соответствующее резервирование ресурсов для соединения уже не было осуществлено (см. Приложение В). В этом случае новое резервирование ресурсов выполняться не будет. Ресурсы предоставляются таким же образом, как и при команде `CreateConnection`, за исключением случая изменения режима соединения на "неактивный" режим. В таком случае предоставляемые ресурсы ДОЛЖНЫ быть сведены до нуля. Тем не менее существующее резервирование ресурсов все-таки сохраняется.
- Параметр **ResourceID** (идентификатор ресурса): Этот параметр является необязательным. При наличии он ДОЛЖЕН быть использован встроенным клиентом для резервирования ресурсов; он заменяет параметр `ResourceID`, хранившийся для соединения.
- Параметр **ReserveDestination** (адресат резерва): Этот параметр является необязательным. При наличии он заменяет параметр `ReserveDestination`, сохраняемый встроенным клиентом для соединения. Если для соединения был получен параметр `RemoteConnectionDescriptor`, то параметр `ReserveDestination` игнорируется.

Команда будет возвращать параметр **LocalConnectionDescriptor** только в том случае, если модифицированы параметры локального соединения, такие как порты RTP и т. д. Так, если, например, будет изменен только режим соединения, то параметр LocalConnectionDescriptor возвращен не будет. LocalConnectionDescriptor НЕ ДОЛЖЕН включаться, если было возвращено сообщение об ошибке и соединение не было изменено. Если параметр соединения пропущен, как, например, режим или подавление пауз, то, по возможности, будет сохранено старое значение такого параметра. Если изменение параметра ведет к необходимости изменения одного или нескольких *неопределенных* параметров, то шлюз может свободно выбирать соответствующие значения для неопределенных параметров, которые должны быть изменены<sup>19</sup>.

Адресная информация протокола RTP, предоставляемая в параметре RemoteConnectionDescriptor, указывает удаленный адрес RTP получателя медиаданных для соединения. Эта адресная информация протокола RTP может быть изменена агентом вызова<sup>20</sup>. Когда адресная информация протокола RTP выдается встроенному клиенту для соединения, встроенному клиенту СЛЕДУЕТ также только принимать медиапотоки (и RTCP) от заданного IP-адреса. Все медиапотоки, принимаемые с любого другого адреса, СЛЕДУЕТ отбрасывать. За дополнительной информацией, касающейся требований по обеспечению безопасности, следует обращаться к Рекомендации МСЭ-Т J.170.

Параметры **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательными. Эти параметры могут быть использованы агентом вызова для включения запроса на уведомление, который связан с модификацией соединения и выполняется одновременно с такой модификацией. Если предоставляются один или несколько из этих параметров, то одним из них ДОЛЖЕН быть параметр RequestIdentifier. Например, когда вызов принят, то вызывающий шлюз должен получить команду ввести соединение в режим "передача/прием" и остановить передачу тональных сигналов контроля посылки вызова. Это может быть выполнено с помощью одной команды ModifyConnection путем включения в нее запроса на уведомление с параметрами RequestedEvents для события "телефонная трубка положена" и пустого параметра SignalRequests для прекращения передачи тональных сигналов контроля посылки вызова. Следует отметить, что отсутствие параметров RequestedEvents и SignalRequests интерпретируется как пустой список только в том случае, когда включен параметр RequestIdentifier.

Когда присутствуют эти параметры, то модификация соединения и запрос на уведомление ДОЛЖНЫ быть синхронизированы, что означает, что оба эти параметры либо приняты, либо отклонены.

NotifiedEntity – необязательный параметр, определяют новый "уведомленный объект" для конечной точки.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

Параметр **ResourceID** (идентификатор ресурса) – это параметр D-QoS, который ДОЛЖЕН быть возвращен шлюзом, если он выполняет резервирование ресурсов и получает новый параметр ResourceID от краевого маршрутизатора. В случае успешного резервирования ресурсов для D-QoS параметр ResourceID обеспечивает оперирование зарезервированными ресурсами. ResourceID НЕ ДОЛЖЕН включаться, если было возвращено сообщение об ошибке и соединение не было изменено.

### 6.3.5 Команда DeleteConnection (от агента вызова)

Команда DeleteConnection (исключить соединение) используется для завершения соединения. Кроме того, при этой команде осуществляется сбор статистических данных по выполнению соединения.

```
ReturnCode
, Connection-parameters
  ← DeleteConnection(CallId
                    , EndpointId
                    , ConnectionId
                    [, NotifiedEntity]
                    [, RequestedEvents]
                    [, RequestIdentifier]
                    [, DigitMap]
                    [, SignalRequests]
                    [, QuarantineHandling]
                    [, DetectEvents])
```

<sup>19</sup> Это может произойти, например, если задано изменение кодека и в старом кодеке использовалось подавление пауз, а в новом кодеке это не поддерживается.

<sup>20</sup> Например, если мультимедийному пакету требуется пройти через систему защиты доступа.

Идентификатор конечной точки в данном виде команды DeleteConnection ДОЛЖЕН быть точно определен. Подстановочные знаки использоваться НЕ ДОЛЖНЫ. Встроенному клиенту, получающему DeleteConnection с условным подстановочным знаком, ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

В общем случае, когда соединение имеет два конца, эта команда должна посылаться обоим шлюзам, участвующим в соединении. После исключения соединения медиапотоки в сети с коммутацией пакетов, ранее поддерживаемые соединением, уже не доступны. Все медиапакеты, принятые для старого соединения, просто отбрасываются, и новые медиапакеты для потока не посылаются. Когда для соединения были выполнены одно или несколько резервирований ресурсов и/или предоставлений ресурсов для D-QoS, по команде DeleteConnection зарезервированные ресурсы будут освобождены.

В ответ на команду DeleteConnection шлюз возвращает список параметров, описывающих состояние соединения. Параметры соединения ДОЛЖНЫ возвращаться только, если команда успешно (выполнена) и соединение удалено. Такими параметрами являются:

- **Число переданных пакетов:** Общее число пакетов данных RTP, переданных отправителем с начала передачи по соединению. Показания счетчика не сбрасываются, если отправитель меняет свой идентификатор источника синхронизации (SSRC, как определено в протоколе RTP), например, в результате команды Modify. Значение ДОЛЖНО определяться на основе той же информации, доставляемой по механизму RTCP.
- **Число переданных октетов:** Общее число октетов полезной нагрузки (то есть не включая заголовок или дополнение битами), переданных отправителем в пакетах данных RTP с начала передачи по соединению. Показания счетчика не сбрасываются, если отправитель меняет свой идентификатор SSRC, например, в результате команды ModifyConnection. Значение ДОЛЖНО определяться на основе той же информации, доставляемой по механизму RTCP.
- **Число полученных пакетов:** Общее число пакетов данных RTP, полученных отправителем с начала приема по соединению. В подсчет входят пакеты, полученные от разных SSRC, если отправитель использовал несколько значений. Все полученные пакеты ДОЛЖНЫ быть сосчитаны, в независимости от режима соединения или типов ошибки запуска, например, ошибка при аутентификации.
- **Число полученных октетов:** Общее число октетов полезной нагрузки (то есть не включая заголовок или заполнение), полученных в пакетах данных RTP отправителем с начала приема по соединению. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Все полученные пакеты ДОЛЖНЫ быть сосчитаны, в независимости от режима соединения или типов ошибки запуска, например, ошибка при аутентификации.
- **Число потерянных пакетов:** Общее число пакетов данных RTP, которые были потеряны с начала приема. Это число определяется как число ожидаемых пакетов, из которого вычтено число фактически принятых пакетов, где в число принятых пакетов входят все пакеты, которые приняты с опозданием или являлись дубликатами. В подсчет входят пакеты, принятые от различных SSRC, если отправитель использовал несколько значений. Таким образом, пакеты, поступившие с опозданием, не считаются потерянными, и потери могут быть отрицательными, если имеются дубликаты. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Число ожидаемых пакетов определяется как последний принятый расширенный порядковый номер, из которого вычтен первоначальный принятый порядковый номер. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Значение равно нулю, если, например, ни одного пакета в соединении не было получено.
- **Флуктуация времени между поступлениями пакетов:** Оценка статистической дисперсии времени между поступлениями пакетов данных RTP в миллисекундах, выраженная целым числом без знака. Флуктуация времени между поступлениями пакетов "J" определяется как среднее отклонение (усредненное абсолютное значение) разницы "D" промежутков между пакетами у получателя по сравнению с отправителем для пары пакетов. Детализированные алгоритмы вычисления приведены в стандарте RFC 3550. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Значение равно нулю, если, например, ни одного пакета в соединении не было получено.

- **Средняя задержка при передаче:** Оценка времени задержки в сети, выраженная в миллисекундах. Это среднее значение разности между отметкой времени по протоколу NTP, указанной отправителями сообщений RTSP, и отметкой времени по протоколу NTP получателей, измеренное при приеме сообщений. Среднее получается путем суммирования всех оценок и последующего деления на число принятых сообщений RTSP. Следует отметить, что правильное вычисление этого параметра опирается на синхронизированные такты. Устройства встроенного клиента МОГУТ альтернативно оценивать среднюю задержку при передаче путем деления на два измеренного времени на передачу и подтверждение.

Более детализированное определение этих переменных содержится в стандарте RFC 3550.

В дополнение к параметрам, приведенным выше, конечная точка, получившая от своего однорангового объекта одно или несколько сообщений отправителя или получателя по протоколу RTSP, ДОЛЖНА вернуть следующие параметры:

- Переданные удаленные пакеты: число переданных по соединению пакетов с точки зрения удаленной конечной точки.
- Переданные удаленные октеты: число переданных по соединению октетов с точки зрения удаленной конечной точки.
- Потерянные удаленные пакеты: число пакетов, которые не были получены по соединению (о чем свидетельствуют пропуски в порядковых номерах) с точки зрения удаленной конечной точки.
- Удаленная флуктуация: средний разброс времени между поступлениями пакетов в миллисекундах, выраженный как целое число, с точки зрения удаленной конечной точки.

Параметры **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательными. Они могут быть использованы агентом вызова для передачи запроса на уведомление, который связан с удалением соединения и осуществляется одновременно с последним. Однако при наличии одного или нескольких из этих параметров одним из них ДОЛЖЕН быть параметр **RequestIdentifier**. Например, когда абонент кладет телефонную трубку, шлюз может получить команду исключить соединение и начать поиск события "телефонная трубка снята" (токовое состояние абонентского шлейфа). Это может быть выполнено с помощью одной команды **DeleteConnection** путем передачи также параметра **RequestedEvents** для "телефонная трубка снята" и пустого параметра **SignalRequests**. Следует отметить, что отсутствие параметров **RequestedEvents** и **SignalRequests** интерпретируется как пустой список только в том случае, когда включен параметр **RequestIdentifier**.

Когда присутствуют эти параметры, команда исключения соединения и запрос на уведомление ДОЛЖНЫ быть синхронизированы, что означает, что эти запрос и команда либо приняты, либо отклонены.

**NotifiedEntity** является необязательным параметром, который определяет новый ""уведомленный объект" для конечной точки.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

### 6.3.6 Команда **DeleteConnection** (от встроенного клиента)

При некоторых обстоятельствах шлюзу может потребоваться освободить соединение, например вследствие того, что он утратил ресурс, связанный с этим соединением. Шлюз может завершить соединение, используя вариант команды **DeleteConnection**:

```
ReturnCode
  ← DeleteConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     Reason-code,
                     Connection-parameters)
```

Параметр **EndpointId** (идентификатор конечной точки) в данном виде команды ДОЛЖЕН быть точно определен. Условные подстановочные знаки использоваться НЕ ДОЛЖНЫ. Встроенный клиент, получающий **DeleteConnection** с условным подстановочным знаком, ДОЛЖЕН возвращать в ответе

сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Параметр **Reason-code** (код причины) – это текстовая строка, начинающаяся с числового кода причины и необязательно сопровождаемая описательной текстовой строкой. Список кодов причины содержится в п. 6.6.

В дополнение к параметрам **CallId**, **EndpointId** и **ConnectionId** встроенный клиент будет также посылать параметры соединения, которые будут возвращаться агенту вызова в ответ на поступившую от него команду **DeleteConnection**. Код причины указывает на причину выдачи команды **DeleteConnection**. Если для соединения были выполнены одно или несколько резервирований и/или предоставлений ресурсов для D-QoS, встроенный клиент освободит зарезервированные ресурсы.

**ReturnCode** (код завершения) – это параметр, возвращаемый агентом вызова. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

### 6.3.7 Команда **DeleteConnection** (от агента вызова множественного соединения)

Для одновременного отключения нескольких соединений агент вызова может использовать некоторую разновидность команды **DeleteConnection**. Эта команда может быть использована для отключения всех соединений, которые относятся к вызову для конечной точки:

```
ReturnCode
  ← DeleteConnection(CallId,
                    EndpointId)
```

В параметре **EndpointId** (идентификатор конечной точки) в данном виде команды **DeleteConnection** НЕ ДОЛЖЕН использоваться подстановочный знак "any of" ("любой из"). Все соединения для конечной точки (точек) с заданным идентификатором вызова будут исключены. По этой команде отдельные статистические данные или параметры вызова не возвращаются. Встроенный клиент, получающий **DeleteConnection** (от агента вызова множественного соединения) с "любым из" условных подстановочных знаков, ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Команда **DeleteConnection** может быть также использована агентом вызова для исключения всех соединений, заканчивающихся в данной конечной точке:

```
ReturnCode
  ← DeleteConnection(EndpointId)
```

При таком виде команды **DeleteConnection** агенты вызовов могут использовать имеющуюся в конечных точках иерархическую структуру именования, чтобы исключить все соединения, принадлежащие группе конечных точек. В этом случае часть компонента "локальное имя конечной точки" параметра **EndpointId** может быть определена с использованием условного подстановочного знака "all" ("все"), как описано в п. 6.1.1. Условный подстановочный знак "any of" ("любой из") НЕ ДОЛЖЕН использоваться. По этой команде отдельные статистические данные или параметры вызовов не возвращаются.

После удаления соединения медиапотоки сетей с коммутацией пакетов, ранее поддерживаемые этим соединением, уже не доступны. Все медиапакеты, полученные по старому соединению, просто отбрасываются, а новые медиапакеты для потока не отправляются. Если для соединения были выполнены одно или несколько резервирований и/или предоставлений ресурсов для D-QoS, встроенный клиент освободит зарезервированные ресурсы.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

### 6.3.8 Проверка

Протокол MGCP основан на архитектуре централизованного контроля вызова, где агент вызова действует как удаленный контроллер устройств клиента, которые предоставляют пользователям и сетям речевые интерфейсы. Чтобы достичь таких же или более высоких уровней доступности, как в существующей сети КТСОП (PSTN), в некоторых протоколах реализованы механизмы

периодического контроля абонентов "звуковыми импульсами", чтобы минимизировать время до обнаружения отдельного сбоя. С этой целью между встроенными клиентами и агентами вызовов в системе IP-Cablecom обеспечивается характерный для протокола MGCP механизм проверки, позволяющий агенту вызова проверять состояние конечной точки и соединения и восстанавливать характерные для протокола функциональные возможности конечной точки.

Для встроенных клиентов определены две команды контроля:

- **AuditEndPoint** (проверка конечной точки): Используется агентом вызова для определения состояния конечной точки.
- **AuditConnection** (проверка соединения): Используется агентом вызова для получения информации о соединении.

Обычно желательно управление сетью сверх возможностей, предоставляемых этими командами, например получение информации о состоянии встроенного клиента в отличие от отдельных конечных точек. Предполагается, что такие возможности должны поддерживаться путем использования простого протокола управления сетью (SNMP) и определения базы MIB для встроенного клиента; оба эти вопроса выходят за рамки настоящей Рекомендации.

### 6.3.8.1 Команда AuditEndPoint

Команда AuditEndPoint (проверка конечной точки) может быть использована агентом вызова для выяснения состояния той или иной конечной точки.

```

{ ReturnCode
  [, EndPointIdList]
  [, NumEndPoints] } |
{ ReturnCode
  [, RequestedEvents]
  [, DigitMap]
  [, SignalRequests]
  [, RequestIdentifier]
  [, NotifiedEntity]
  [, ConnectionIdentifiers]
  [, DetectEvents]
  [, ObservedEvents]
  [, EventStates]
  [, VersionSupported]
  [, ReasonCode]
  [, MaxMGCPDatagram]
  [, Capabilities] }
← AuditEndPoint (EndpointId
  [, RequestedInfo] |
  [, SpecificEndPointID]
  [, MaxEndPointIDs] })

```

Параметр **EndpointId** (идентификатор конечной точки) определяет конечную точку, которая проверяется. Условный подстановочный знак "any of" ("любой из") использоваться НЕ ДОЛЖЕН. Встроенный клиент, получающий AuditEndPoint с "любым из" условных подстановочных знаков, ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Условный подстановочный знак "all of" ("все из") может быть использован для проверки группы конечных точек. Если используется этот условный знак, шлюз ДОЛЖЕН возвращать список идентификаторов конечных точек, которые соответствуют подстановочному знаку в параметре **EndPointIdList** (список идентификаторов конечных точек), являющемся просто списком параметров SpecificEndpointIds; параметр RequestedInfo (запрошенная информация) НЕ ДОЛЖЕН быть включен в список в этом случае. Параметр **MaxEndPointIDs** (максимальное число параметров EndpointIds) – это численное значение, которое указывает на максимальное число возвращаемых параметров EndpointId. Если существуют дополнительные конечные точки, то ДОЛЖЕН присутствовать возвращаемый параметр **NumEndPoints** (число конечных точек), указывающий на общее число конечных точек, соответствующих заданному параметру EndpointID. Чтобы восстановить следующий блок параметров EndpointID, параметр **SpecificEndPointID** (идентификатор конкретной конечной точки) устанавливается на значение последней конечной точки, возвращенной в предыдущем параметре EndPointIDList, и генерируется команда.

Когда условный подстановочный знак не используется, тогда (возможно, пустой) параметр **RequestedInfo** описывает информацию, которая запрошена для заданного параметра **EndpointId**; параметры **SpecificEndpointID** и **MaxEndpointID** НЕ ДОЛЖНЫ быть использованы. По данной команде затем может быть проверена следующая характеристика для конечной точки информации:

**RequestedEvents**, **DigitMap**, **SignalRequests**, **RequestIdentifier**, **NotifiedEntity**,  
**ConnectionIdentifiers**, **DetectEvents**, **ObservedEvents**, **EventStates**, **VersionSupported**, **ReasonCode**,  
**MaxMGCPDatagram** и **Capabilities**.

Если конечную точку запрашивают о параметре, который ей неизвестен, то она НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого такой параметр ДОЛЖЕН быть опущен в ответе.

Если конечную точку запрашивают о параметре, который она поддерживает, но значения которого у нее нет, конечная точка НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого в ответ ДОЛЖЕН быть включено пустое значение параметра.

Только при успешном выполнении в ответ, в свою очередь, будет включена информация о каждом из параметров, для которого была запрошена информация для проверки. За исключением параметров, явным образом отмеченных как "необязательные", конечные точки ДОЛЖНЫ поддерживать все следующие параметры:

- **RequestedEvents** (запрашиваемые события) – Текущее значение параметра **RequestedEvents**, которое использует конечная точка, включая действие, связанное с каждым событием. Устойчивые события включаются в список.
- **DigitMap** (отображение цифр набора номера) – Отображение цифр, которое использует конечная точка в настоящее время.
- **SignalRequests** (запросы сигналов) – Список сигналов выдержки времени, которые активны в настоящее время, сигналов "включено/выключено", которые в текущее время "включены" для конечной точки (с параметром или без него), и всех ожидающих передачи коротких сигналов<sup>21</sup>. Сигналы выдержки времени с истечением этого времени и передаваемые в текущее время короткие сигналы в список не входят. О параметризованных сигналах сообщается в параметрах, с которыми они применяются.
- **RequestIdentifier** (идентификатор запроса) – Параметр **RequestIdentifier** для последней команды **NotificationRequest**, полученной конечной точкой (включает запрос на уведомление, встроенный в примитивы обработки соединения). Если запрос на уведомление не был принят, то будет возвращено нулевое значение.
- **NotifiedEntity** (уведомленный объект) – Текущий "уведомленный объект" для конечной точки. Следует отметить, что адаптер МТА может содержать только абсолютное доменное имя (включая имя хоста) своего параметра **NotifiedEntity**, если ему было предоставлено только абсолютное доменное имя (включая имя хоста) через параметр **NotifiedEntity** сообщения или подтверждения сообщения NCS. В этом случае системе CMS СЛЕДУЕТ принять это значение.
- **ConnectionIdentifiers** (идентификаторы соединений) – Это список параметров **ConnectionIdentifiers**, разделенных запятыми, для всех соединений, которые имеются в настоящее время для заданной конечной точки.
- **DetectEvents** (обнаруживаемые события) – Текущее значение параметра **DetectEvents**, используемое конечной точкой. Устойчивые события включены в этот список.
- **ObservedEvents** (наблюдаемые события) – Текущий список наблюдаемых событий для конечной точки.
- **EventStates** (состояние событий) – Этот параметр используется для событий, которые имеют связанные с ними контролируемые состояния; событие соответствует состоянию, в котором находится конечная точка; например, состояние "телефонная трубка снята", если конечная точка находится в состоянии "телефонная трубка снята". В определении отдельных событий будет указываться, имеет ли рассматриваемое событие связанное с ним контролируемое состояние.
- **VersionSupported** (поддерживаемая версия) – Список версий протокола, поддерживаемых конечной точкой.
- **ReasonCode** (код причины) – Значение параметра "код причины" в последней команде **RestartInProgress** или **DeleteConnection**, генерируемой шлюзом для конечной точки, или специальное значение 000, если состояние конечной точки нормальное.

---

<sup>21</sup> В текущее время не должно быть ожидающих обработки коротких сигналов.

- **MaxMGCPDatagram** (максимальная дейтаграмма протокола MGCP) – Максимальный размер дейтаграммы протокола MGCP в байтах, поддерживаемой конечной точкой (см. п. 7.5.3). Из этого значения исключается любая служебная информация нижнего уровня. Поддержка для этого параметра является необязательной. Если значение не возвращается, то по умолчанию предполагается максимальный размер дейтаграммы протокола MGCP.
- **Capabilities** (возможности) – Возможности для конечной точки, аналогичные параметру LocalConnectionOptions и включающие пакеты событий и режимы соединения. Если сообщается о каких-либо неизвестных возможностях, они просто ДОЛЖНЫ быть проигнорированы. Если необходимо указать, что некоторые параметры, такие, например, как подавление пауз, совместимы только с некоторыми кодеками, шлюз будет возвращать несколько наборов возможностей.
- **Compression Algorithm** (алгоритм сжатия) – Список поддерживаемых кодеков. ДОЛЖНЫ быть использованы литеральные имена, определенные в Рекомендации МСЭ-Т J.161. Неизвестные алгоритмы сжатия в случае их получения СЛЕДУЕТ игнорировать. Остальные параметры будут применяться ко всем кодекам, определенным в этом списке.
  - **Период пакетирования** – Может быть задано одно значение или диапазон.
  - **Ширина полосы пропускания** – Могут быть заданы одно значение или диапазон, соответствующие диапазону периодов пакетирования (в предположении отсутствия подавления пауз).
  - **Компенсация эха** – Поддерживается или нет компенсация эха.
  - **Подавление пауз** – Поддерживается или нет подавление пауз.
  - **Тип услуги** – Поддерживается или нет тип услуги.
  - **Пакеты событий** – Список поддерживаемых пакетов событий. Первый пакет событий в списке будет пакетом, определяемым по умолчанию.
  - **Режимы** – Список поддерживаемых режимов соединения.
  - **Динамическое качество обслуживания** – Поддерживается или нет динамическое качество обслуживания.
  - **Обеспечение безопасности** – Поддерживаются или нет услуги обеспечения безопасности в проекте IPcablecom. Если поддерживаются, то могут присутствовать также следующие параметры:
    - **Шифрокомплекты протокола RTP** – Список алгоритмов аутентификации и шифрования, поддерживаемых для протокола RTP.
    - **Шифрокомплекты протокола RTCP** – Список алгоритмов аутентификации и шифрования, поддерживаемых для протокола RTCP.

Затем агент вызова может принять решение об использовании команды AuditConnection для получения дальнейшей информации о соединениях.

Параметр **ReturnCode** (код завершения) является параметром, возвращаемым шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

Если не была запрошена никакая информация, а параметр EndpointId относится к действительному, полностью определенному параметру EndpointId, то шлюз просто возвращает успешный ответ (код завершения 200: нормально выполненная транзакция).

Следует отметить, что вся возвращаемая информация является просто статической. Полученные новые команды, локальные действия и т. д. могут изменить большую часть параметров, приведенных выше. Например, состояние телефонной трубки может измениться до того, как агент вызова получит упомянутую выше информацию.

### 6.3.8.2 Команда AuditConnection

Контроль отдельных соединений в конечной точке может быть обеспечен с использованием команды AuditConnection (проверка соединения).

```
ReturnCode
[, CallId]
[, NotifiedEntity]
[, LocalConnectionOptions]
[, Mode]
[, RemoteConnectionDescriptor]
[, LocalConnectionDescriptor]
[, ConnectionParameters]
```

```

← AuditConnection (EndpointId
, ConnectionId
[, RequestedInfo])

```

Параметр **EndpointId** определяет конечную точку, которая проверяется; подстановочные знаки использоваться НЕ ДОЛЖНЫ. Встроенный клиент, получающий AuditConnection с условным подстановочным знаком, ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена). Параметр (возможно, пустой) **RequestedInfo** (запрашиваемая информация) описывает информацию, которая запрошена для параметра **ConnectionId** в пределах заданного параметра EndpointId. По данной команде может быть проконтролирована следующая информация соединения:

```

CallId, NotifiedEntity, LocalConnectionOptions,
Mode, ConnectionParameters, RemoteConnectionDescriptor,
LocalConnectionDescriptor.

```

Если конечную точку запрашивают о параметре, который ей неизвестен, то она НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого такой параметр ДОЛЖЕН быть опущен в ответе.

Если конечную точку запрашивают о параметре, который она поддерживает, но значения которого у нее нет, конечная точка НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого в ответ ДОЛЖЕН быть включено пустое значение параметра. Только при успешном выполнении, в ответ, в свою очередь, будет включена информация о каждом из параметров, для которого была запрошена информация для проверки. За исключением параметров, явным образом отмеченных как "необязательные", конечные точки ДОЛЖНЫ поддерживать все следующие параметры:

- **CallId** (идентификатор вызова) – Параметр CallId для вызова, которому принадлежит соединение.
- **NotifiedEntity** (уведомленный объект) – Текущий "уведомленный объект" для конечной точки.
- **LocalConnectionOptions** (варианты локального соединения) – Параметр LocalConnectionOptions, предоставленный для соединения.
- **Mode** (режим) – Текущий режим соединения.
- **ConnectionParameters** (параметры соединения) – Параметры текущего соединения для данного соединения.
- **LocalConnectionDescriptor** (местный дескриптор соединения) – Параметр LocalConnectionDescriptor, который предоставлен шлюзом для соединения.
- **RemoteConnectionDescriptor** (удаленный дескриптор соединения) – Последний полученный параметр RemoteConnectionDescriptor, который был предоставлен шлюзу для этого соединения в предшествующей команде CreateConnection и ModifyConnection.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

Если не была запрошена никакая информация, а параметр EndpointId относится к действительной конечной точке, шлюз просто проверяет, что описанное соединение существует, и если это так, то возвращает положительный ответ (код завершения 200 – выполненная транзакция).

### 6.3.9 Команда RestartInProgress

Команда RestartInProgress (перезапуск в процессе работы) используется шлюзом для сигнализации о том, что конечная точка или группа конечных точек выведены из эксплуатации или вводятся обратно в эксплуатацию.

```

ReturnCode
[, NotifiedEntity]
[, VersionSupported]
← RestartInProgress (EndpointId
, RestartMethod
[, RestartDelay]
[ReasonCode])

```

Параметр **EndpointId** определяет конечные точки, которые выводятся из эксплуатации или вводятся в эксплуатацию. Условный подстановочный знак "all of" ("все из") может быть использован для

применения этой команды к группе конечных точек, например ко всем конечным точкам, которые присоединены к заданному интерфейсу, или даже ко всем конечным точкам, которые присоединены к данному шлюзу. Условный подстановочный знак "any of" ("любой из") НЕ ДОЛЖЕН использоваться. Агент вызова, получающий Restart in Progress с условным подстановочным знаком "любой из", ДОЛЖЕН возвращать в ответе сообщение об ошибке (в качестве кода возвращаемой ошибки СЛЕДУЕТ указывать код ошибки 500 – транзакция не может быть произведена, поскольку конечная точка не определена).

Параметр RestartMethod (метод перезапуска) определяет тип перезапуска:

- Метод "постепенного перезапуска" указывает на то, что определенная конечная точка (точки) будет выведена из эксплуатации после заданной "задержки перезапуска". Установленные соединения еще не затронуты этим процессом, но агент вызова должен воздержаться от установления новых соединений и должен попытаться постепенно разъединить все существующие соединения. По истечении времени задержки перезапуска шлюз должен послать новое сообщение RSIP с методом "принудительного" перезапуска. Это явно покажет агенту вызова, что конечная точка теперь выведена из эксплуатации.
- Метод "принудительного" перезапуска указывает на то, что определенные конечные точки внезапно выведены из эксплуатации. Установленные соединения, если таковые имелись, будут потеряны.
- Метод "постепенного, с отменой" перезапуска указывает на то, что шлюз аннулирует ранее полученный метод "постепенного" перезапуска для этих же конечных точек. Конечные точки остаются в эксплуатации. Когда передается эта команда, шлюз сразу же начинает давать разрешение на установку новых соединений в этих конечных точках.
- Метод "чистого перезапуска" указывает на то, что работа в конечных точках будет восстановлена после заданной "задержки перезапуска". Отсутствуют соединения, которые в текущее время устанавливаются в этих конечных точках.
- "Несвязный" метод указывает на то, что конечная точка стала несвязной (разъединенной) и теперь пытается восстановить связность. "Задержка перезапуска" определяет число секунд, в течение которых конечная точка была несвязной. Установленные соединения этим процессом не затрагиваются.

Необязательный параметр "задержка перезапуска" выражается числом секунд. Если это число отсутствует, то значение задержки должно считаться нулевым. В случае использования метода "постепенного" перезапуска нулевая задержка указывает на то, что конечная точка никогда не окажется выведенной из эксплуатации в результате этой операции, и что агент вызова должен просто ждать естественного завершения существующих соединений без установления новых соединений. Задержка перезапуска всегда считается нулевой в случае использования "принудительного" и "постепенного, с отменой" методов. Нулевая задержка перезапуска в случае использования метода "чистого перезапуска" указывает на то, что работа уже восстановлена. Это обычно происходит после запуска/перезапуска шлюза. Для смягчения влияния изменения IP-адреса клиента агент вызова МОЖЕТ устранить доменное имя встроенного клиента путем запроса системы DNS независимо от времени существования (TTL) текущей записи ресурса для перезапускаемого встроенного клиента.

Встроенным клиентам СЛЕДУЕТ в порядке вежливости посылать агенту вызова сообщение RestartInProgress для "постепенного" или "принудительного" перезапуска, когда они выводятся из эксплуатации, например, путем прекращения функционирования. Однако встроенному клиенту СЛЕДУЕТ посылать сообщение RestartInProgress для "принудительного" перезапуска, если они отключаются системой управления сетью, хотя агент вызова не может рассчитывать на постоянный прием таких сообщений.

Встроенные клиенты ДОЛЖНЫ посылать своему агенту вызова сообщение RestartInProgress для "чистого перезапуска" с нулевой задержкой, когда они вводятся обратно в эксплуатацию согласно процедуре перезапуска, описанной в п. 6.4.3.5; агенты вызова могут рассчитывать на прием этого сообщения. Кроме того, встроенные клиенты ДОЛЖНЫ посылать сообщение RestartInProgress для "несвязного" метода перезапуска своему текущему "уведомленному объекту" согласно "несвязной" процедуре, описанной в п. 6.4.3.6. Параметр "задержка перезапуска" НЕ ДОЛЖЕН использоваться при методе "принудительного" и "постепенного, с отменой" перезапуска.

Необязательный параметр ReasonCode (код причины) может быть использован для указания на причину перезапуска. Сообщение RestartInProgress будет послано текущему "уведомленному объекту" для рассматриваемого параметра EndpointId. Ожидается, что агент вызова по умолчанию, то есть "уведомленный объект", был обеспечен для каждой конечной точки, так что после перезагрузки агентом вызова по умолчанию будет "уведомленный объект" для каждой конечной точки. Встроенные клиенты

ДОЛЖНЫ в полной мере использовать подстановочные знаки для минимизации числа сообщений RestartInProgress, которые генерируются, когда в шлюзе перезапускается несколько конечных точек и управление конечными точками осуществляет один и тот же агент вызова.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый агентом вызова. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое необязательно сопровождается комментарием.

Параметр **NotifiedEntity** (уведомленный объект) МОЖЕТ быть дополнительно возвращен с ответом на сообщение RestartInProgress от агента вызова: это обычно должно делаться только в ответ на "чистый перезапуск" или "несвязный" перезапуск (см. также пп. 6.4.3.5 и 6.4.3.6). Если параметр NotifiedEntity был включен в возвращенный ответ, он определяет новый "уведомленный объект" для конечной точки (точек): эта операция ДОЛЖНА быть совершена с кодом ошибки 521 в ответе (конечная точка перенаправлена). Следует обратить внимание на то, что возвращение параметра NotifiedEntity в ответе определено только для ответов на RestartInProgress и его НЕ СЛЕДУЕТ проводить при ответе на другие команды. Любая другая линия поведения не определена.

- Если ответ указывает на успешный перезапуск (код завершения 200 – выполненная транзакция), то рассматриваемый перезапуск прошел успешно, и возвращенный параметр NotifiedEntity – это новый "уведомленный объект" для конечной точки (точек).
- Если ответ от агента вызова содержит код ошибки, то рассматриваемый перезапуск еще не завершен. Если в ответе получен код 521 (переедресация конечной точки), этот ответ ДОЛЖЕН включать параметр NotifiedEntity, который определяет новый "уведомленный объект" для конечной точки (точек) и ДОЛЖЕН быть использован при повторной попытке рассматриваемого перезапуска (в качестве новой транзакции).

В случае "чистого перезапуска" и "несвязного" перезапуска повторная попытка рассматриваемого перезапуска ДОЛЖНА предприниматься каждый раз, когда агент вызова возвращает код нерегулярной ошибки (4xx), в то время как при любом другом методе перезапуска СЛЕДУЕТ предпринимать повторную попытку. РЕКОМЕНДУЕТСЯ, чтобы любой тип перезапуска завершался, при возвращении кода постоянной ошибки (5xx), за исключением описанного выше кода 521.

Наконец, параметр **VersionSupported** (поддерживаемая версия) со списком поддерживаемых версий может быть возвращен, если в ответе указывается на несовместимость версий (код ошибки 528).

## 6.4 Состояния, восстановление после отказа и состязательные условия

Для реализации правильной сигнализации вызова агент вызова должен отслеживать состояние конечной точки, а шлюз должен удостовериться, что агент вызова должным образом уведомляется о событиях. Могут существовать особые условия, когда имеет место перезапуск шлюза или агента вызова: во время процедур "восстановления после отказа" может оказаться необходимым переадресовать шлюз к новому агенту вызова. Аналогичным образом, агенту вызова может потребоваться предпринять специальное действие, когда шлюз автономен или перезапущен.

### 6.4.1 Итоги и выводы

Как упоминалось в п. 6.1.4, агенты вызовов идентифицируются по их доменному имени, а каждая конечная точка имеет один и только один "уведомленный объект", связанный с нею в любой заданный момент времени. В этом пункте приводятся итоги и выводы по тем областям, которые представляют особую важность для обеспечения надежности и восстановлению после отказа согласно протоколу MGCP:

- Агент вызова идентифицируется по его доменному имени, а не по сетевым адресам, и с доменным именем может быть связано несколько сетевых адресов.
- У конечной точки имеется один и только один связанный с нею агент вызова в любой заданный момент времени. Агент вызова, связанный с конечной точкой, – это текущее значение "уведомленного объекта".
- Уведомленный объект первоначально устанавливается на предварительное значение. Когда для конечной точки получены команды с параметром NotifiedEntity, включая имена конечных точек с подстановочными знаками, "уведомленный объект" устанавливается на заданное значение. Если "уведомленный объект" для конечной точки является пустым или не был установлен явным образом<sup>22</sup>, "уведомленный объект" устанавливается по умолчанию на адрес источника

---

<sup>22</sup> Это может произойти, например, в результате определения пустого параметра NotifiedEntity.

последней команды обработки соединения или запроса на уведомление, полученного для конечной точки. В этом случае агент вызова будет, таким образом, идентифицироваться по его сетевому адресу, что СЛЕДУЕТ делать только в исключительном случае.

- Ответы на команды всегда посылаются по адресу источника команды независимо от имеющегося "уведомленного объекта". Когда требуется вложить в ответ сообщение Notify, дейтаграмма по-прежнему посылается по адресу источника новой принятой команды независимо от параметра NotifiedEntity для любой из команд.
- Когда "уведомленный объект" относится к доменному имени, которое разделяется на несколько IP-адресов, конечные точки могут переключаться между каждым из этих адресов; однако они не могут по своему усмотрению отнести "уведомленный объект" к другому доменному имени. Тем не менее агент вызова может выдать им команду произвести коммутацию, предоставив им новый "уведомленный объект".
- Если агент вызова становится недоступным, то конечные точки, управляемые этим агентом вызова, возможно станут "несвязными". Чтобы эти конечные точки вновь стали связными, либо недоступный агент вызова снова должен стать доступным, либо другой (резервный) агент вызова должен взаимодействовать с "затронутыми" конечными точками с новым "уведомленным объектом".
- Если другой (резервный) агент вызова взял на себя контроль над группой конечных точек, предполагается, что недоступный агент вызова будет взаимодействовать и будет синхронизированным с резервным агентом вызова, чтобы передать контроль группой "затронутых" конечных точек обратно исходному агенту вызова, если это необходимо. Как вариант, недоступный агент вызова может просто стать теперь резервным агентом вызова.

Следует отметить, что разрешение конфликтных ситуаций при передаче управления между отдельными агентами вызовов не обеспечивается; следует четко полагаться на то, что агенты вызовов знают, что они делают и как взаимодействуют друг с другом (хотя команда AuditEndpoint может быть использована для получения информации о существующем "уведомленном объекте").

#### **6.4.2 Повторная передача и обнаружение потерянных соединений**

Протокол MGCP организован в виде множества транзакций, каждая из которых состоит из команды и ответа. Сообщения MGCP, которые переносятся по протоколу UDP, могут быть подвержены потерям. При отсутствии своевременного ответа (см. п. 7.5) команды повторяются. Объекты MGCP ДОЛЖНЫ хранить в памяти список ответов, которые они послали последним транзакциям (т. е. список всех ответов, которые они переслали за последние  $T_{t_{hist}}$ , секунд), и список транзакций, выполняемых в текущее время.

По умолчанию значение  $T_{t_{hist}}$  составляет 30 секунд.

Идентификаторы транзакций входящих команд сначала сравниваются с идентификаторами транзакций последних ответов. Если выявлено совпадение, то объект MGCP не выполняет транзакцию, а просто повторяет старый ответ. Если такого совпадения нет, то идентификатор транзакции входящей команды сравнивается со списком транзакций, выполнение которых еще не завершено. Если совпадение выявлено, то объект MGCP не выполняет транзакцию; последующая обработка зависит от рассматриваемой команды. Если это команда CreateConnection или ModifyConnection, то объект MGCP (в данном случае шлюз) ДОЛЖЕН послать предварительный ответ. Если это какая-либо другая команда, она просто игнорируется. В любом случае конечный ответ будет представлен, когда завершится выполнение команды.

Этот механизм повторения используется для защиты от следующих четырех типов возможных ошибок:

- ошибки передачи, когда, например, происходит потеря пакета из-за шума в линии или перегрузки в очереди;
- отказа компонента, когда, например, интерфейс для агента вызова становится недоступным;
- отказа агента вызова, когда, например, все интерфейсы для агента вызова становятся недоступными;
- восстановление после отказа, когда новый агент вызова "принимается" прозрачным образом.

Элементы должны быть способны получать оценку коэффициента потерь пакетов на основе предыстории. В надлежащим образом сконфигурированной системе этот коэффициент потерь должен быть очень

низким, обычно в среднем менее 1%. Если агенту вызова или шлюзу приходится повторять сообщение более, чем несколько раз, законно предположить, что происходит нечто иное, чем ошибка при передаче. Например, при равномерно распределенном коэффициенте потерь 1%, вероятность того, что пять последовательных попыток передачи будут неудачными, составляет 1 на 100 миллиардов; событие, которое должно происходить один раз каждые 10 дней для агента вызова, который обрабатывает 1000 транзакций в секунду. (В действительности, число повторений, которое считается чрезмерным, должно быть функцией преобладающего коэффициента потерь пакетов). Когда ошибки распределены неравномерно, то вероятность последовательных отказов может стать несколько выше. Следует отметить, что "порог подозрения", обозначенный как Max1, обычно ниже "порога разъединения", обозначенного как Max2. Max2 ДОЛЖЕН быть установлен на большее значение, чем Max1.

В классическом алгоритме повторной передачи будет просто подсчитываться число последовательных повторений и заключение, что соединение нарушено, будет делаться после чрезмерно большого количества повторных передач пакета (обычно от 7 до 11 раз). Чтобы учесть возможность необнаруженного или происходящего в текущее время восстановления после отказа, классический алгоритм модифицируется следующим образом (алгоритм повторной передачи, включающий эти модификации, показан на рисунке 4):

- Шлюз ДОЛЖЕН всегда проверять наличие нового агента вызова. Это может быть замечено благодаря:
  - приему команды, где параметр NotifiedEntity указывает на нового агента вызова; или
  - приему ответа переадресации, указывающего на нового агента вызова.
- Если обнаружен новый агент вызова, то шлюз ДОЛЖЕН перенаправлять повторные передачи всех ожидающих выполнения команд для конечной точки (точек) этому новому агенту вызова. Ответы на новые или старые команды по-прежнему передаются по адресу источника команды.
- В первую очередь, до любой повторной передачи проверяется, чтобы время, истекшее с момента передачи начальной дейтаграммы, не превышало  $T_{s_{max}}$ . Если истекшее время больше  $T_{s_{max}}$ , то повторная передача ДОЛЖНА прекратиться. Если истекшее время больше  $2 * T_{s_{max}}$ , то конечная точка становится несвязной (разъединенной).
- Если число повторных передач этому агенту вызова равно Max1, то шлюз МОЖЕТ в активном режиме опросить сервер имен, чтобы обнаружить возможное изменение интерфейсов агента вызова, независимо от времени существования (TTL), связанного с записью системы DNS.
- Шлюз может запомнить несколько IP-адресов для агента вызова. Если число повторных передач по этому IP-адресу больше или равно Max1 и меньше Max2 и есть еще IP-адреса, по которым не предпринимались попытки передач, шлюз ДОЛЖЕН направить повторные передачи по остальным альтернативным адресам в своем локальном списке. Также, получение явных сетевых уведомлений, таких как, например, недостижимая сеть ICMP, хост, протокол или порт, ДОЛЖНЫ указывать шлюзу на необходимость попытки использования альтернативного адреса (с должным вниманием к возможным сообщениям безопасности).
- Если интерфейсов для попыток передачи больше нет, а число повторных передач составляет Max2, тогда шлюзу СЛЕДУЕТ установить связь с системой DNS еще раз, чтобы узнать, не стали ли доступными какие-либо другие интерфейсы. Если других интерфейсов по-прежнему нет, то повторная передача ДОЛЖНА прекратиться. Если истекшее время больше  $2 * T_{s_{max}}$ , то конечная точка (точки) разъединяется (становится несвязной).
- Шлюз ДОЛЖЕН инициировать "несвязную" процедуру, как описано в п 6.4.3.6.
- Агент вызова НЕ ДОЛЖЕН пытаться использовать конечную точку для новых вызовов, до тех пор пока связность будет восстановлена. Более того, агент вызова ДОЛЖЕН реализовать алгоритм для определения момента, когда в дальнейшем связность будет восстановлена (например, при получении ответа на регулярную команду Audit Endpoint). Когда связность с конечной точкой восстановлена, и если не существует никаких других условий, препятствующих конечной точке поддерживать вызов, агент вызова ДОЛЖЕН удостовериться, что конечная точка может использоваться для новых вызовов без затребования какого-либо ручного вмешательства.

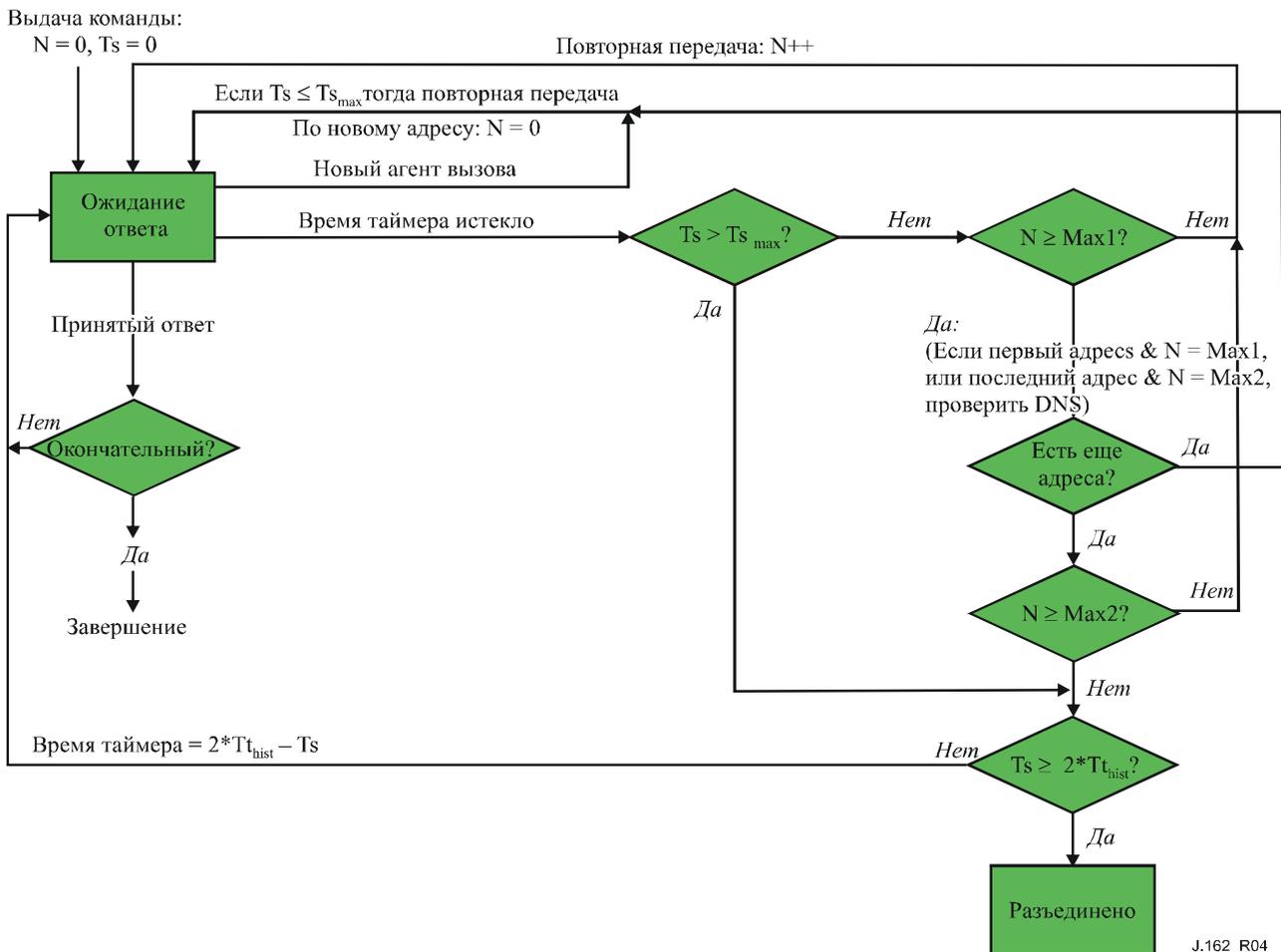


Рисунок 4/J.162 – Алгоритм повторной передачи

Для автоматической адаптации к сетевой нагрузке в протоколе MGCP определены таймеры с экспоненциально возрастающим временем (см. п. 7.5.2). Если начальная выдержка времени установлена на 200 миллисекунд, то потеря пятой повторной передачи обнаружится примерно через 6 секунд. Это, вероятно, приемлемая задержка ожидания для обнаружения восстановления после отказа. Повторная передача должна продолжаться только после этой задержки не только для того, чтобы, вероятно, преодолеть проблему неустойчивой связности, но и для того, чтобы дать некоторое дополнительное время для осуществления восстановления после отказа; полагается, что ожидание общей задержки в 30 секунд является приемлемым.

Однако является важным ограничивать максимальное время задержки повторной передачи. До начала любой повторной передачи проводится проверка не превысило ли время ( $T_s$ ), прошедшее с момента отправления инициирующей дейтаграммы, значения времени  $T_{s_{max}}$ . Если прошло больше времени, чем  $T_{s_{max}}$ , повторная передача ДОЛЖНА прерываться. По истечении времени  $T_{s_{max}}$  или если все повторные передачи на все известные IP-адреса были отосланы, то перед объявлением конечной точки несвязной вводится пауза. Эта пауза представляет собой период времени, единственным действием в течение которого является ожидание ответа от любой из совершенных в последнее время повторных передач. Период молчания длится на протяжении удвоенной продолжительности существования ( $2 * T_{hist}$ ). Такое время стабилизации позволяет завершить или провести выдержку времени всех активных транзакций до того, как конечная точка будет объявлена несвязной. Это помогает удостовериться, что каждая перезагрузка конечной точки осуществляется с чистого и исходного состояния. Если прошло больше времени, чем  $2 * T_{hist}$ , конечная точка становится несвязной. Значение  $T_{s_{max}}$  взаимосвязано со значением  $T_{hist}$ : значение  $T_{hist}$  ДОЛЖНО быть больше или равно, чем  $T_{s_{max}}$  плюс максимальная задержка распространения по сети  $T_{p_{max}}$ . Другими словами, чтобы предотвратить многократное выполнение повторно переданных команд, ДОЛЖНО выполняться следующее соотношение:

$$T_{hist} \geq T_{s_{max}} + T_{p_{max}}$$

Значение для  $T_{s_{max}}$  по умолчанию составляет 20 секунд. Таким образом, если предполагаемая максимальная задержка при распространении составляет 10 секунд, тогда ответы на старые транзакции должны сохраняться в течение по меньшей мере 30 секунд. Важность наличия соглашения по этим значениям между отправителем и получателем нельзя переоценить.

Для Max1 значение по умолчанию составляет 5 повторных передач, а для Max2 – 7 повторных передач. Оба эти значения могут быть изменены в процессе предварительной подготовки.

Более того, процесс предварительной подготовки ДОЛЖЕН иметь возможность нейтрализации запросов системы DNS для одного или обоих Max1 и Max2.

### 6.4.3 Состязательные условия

В данном пункте приводится описание действий протокола MGCP в состязательных условиях.

Прежде всего, протокол MGCP имеет дело с состязательными условиями посредством "карантинного списка", в соответствии с которым события подвергаются карантину, и путем явного обнаружения десинхронизации, например, для несогласованного состояния "телефонная трубка снята/положена" из-за помехи приему вызова звонком для конечной точки.

Во-вторых, протокол MGCP не предполагает, что транспортный механизм будет поддерживать порядок следования команд и ответов. Это может привести к состязательным условиям, которые могут быть устранены посредством должного поведения агента вызова с помощью соответствующего упорядочения команд.

Наконец, в некоторых случаях многие шлюзы могут принять решение выполнить операцию перезапуска в одно и то же время. Это может произойти, например, если в каком-либо районе пропадает энергоснабжение или утрачиваются возможности передачи во время землетрясения или снежной бури. Когда электропитание и возможности передачи вновь восстанавливаются, многие шлюзы могут решить одновременно послать команды RestartInProgress, что может привести к весьма нестабильной работе, если этот процесс оставить без тщательного контроля.

#### 6.4.3.1 Карантинный список

Шлюзы, управляемые по протоколу MGCP, будут получать запросы на уведомление, согласно которым от них требуется следить за списком событий. Протокольными элементами, определяющими обработку этих событий, являются список "запрашиваемых событий", "отображение цифр", "карантинная обработка" и список "обнаруженных событий".

Когда инициализируется конечная точка, список запрашиваемых событий состоит только из устойчивых событий для конечной точки, а отображение цифр предполагается пустым. После получения команды NotificationRequest шлюз начинает наблюдать за конечной точкой на предмет появления событий, упомянутых в этом списке, включая устойчивые события.

События рассматриваются по мере их появления. Последующее действие определяется параметром "действие", связанным с событием из списка запрашиваемых событий, а также отображением цифр набора номера. События с определениями "накопление" или "накопление согласно отображению цифр" накапливаются в списке наблюдаемых событий. События с пометкой "накопление согласно отображению цифр" будут дополнительно накапливаться в "строке текущего набора номера". Это будет продолжаться до тех пор, пока не появится одно из событий, которое инициирует команду Notify, которая будет послана к текущему "уведомленному объекту".

Шлюз в этой точке будет передавать команду Notify и введет конечную точку в "состояние уведомления". Пока конечная точка находится в этом "состоянии уведомления", события, которые обнаруживаются в конечной точке, записываются в "карантинный" буфер для дальнейшей обработки. Эти события, в известном смысле, "подвергаются карантину". Обнаруженные события – это события, определяемые объединением параметра RequestedEvents и самого последнего полученного параметра DetectEvents, или, в случае если параметр DetectEvents не был получен, событий, на которые даются ссылки в параметре RequestedEvents. Также обнаруживаются устойчивые события.

Конечная точка выходит из "состояния уведомления" при получении ответа (успешного или неудачного) на команду Notify<sup>23</sup>. Как описано в п. 6.4.2, команда Notify может быть повторно передана в "состоянии уведомления". Если при этом конечная точка является или становится несвязной (см. п. 6.4.2), то ответ на команду Notify не будет получен никогда. Команда Notify тогда утрачивается и,

---

<sup>23</sup> Следует отметить, что команда Notify не может быть объединена со встроенной командой NotificationRequest.

следовательно, больше не считается ожидающей обработки, хотя конечная точка все еще находится в "состоянии уведомления". Если это случится, завершение выполнения несвязной процедуры, описанной в п. 6.4.3.6, должно тогда привести к выходу конечной точки из "состояния уведомления".

Когда конечная точка выходит из "состояния уведомления", она сбрасывает список наблюдаемых событий и "строку текущего набора номера" до нулевого значения.

После этого поведение шлюза зависит от значения параметра QuarantineHandling (карантинная обработка) при инициировании команды NotificationRequest.

Если агент вызова определил, что он ожидает максимум одно уведомление в ответ на команду запроса на уведомление (режим "шаг блокировки"), тогда шлюз просто ДОЛЖЕН сохранять накапливающиеся события в карантинном буфере, пока не получит следующую команду запроса на уведомление. Пока это не произойдет, конечная точка находится в "состоянии шага блокировки", а события, имеющие место и подлежащие обнаружению, просто заносятся в карантинный буфер. События, подвергаемые карантину, те же самые, что и в "состоянии уведомления". Как только будет получена и успешно выполнена новая команда NotificationRequest, конечная точка выйдет из "состояния шага блокировки".

Однако если шлюзу разрешено посылать несколько последовательных команд Notify (режим "цикл"), он будет действовать следующим образом. Когда шлюз выходит из "состояния уведомления", он сбрасывает список наблюдаемых событий и "строку текущего набора номера" конечной точки до нулевого значения и начинает обработку списка подвергнутых карантину событий, используя уже полученный список запрашиваемых событий и отображение цифр набора номера. При обработке этих событий шлюзу может встретиться событие, которое инициирует команду Notify для передачи. Если это так, шлюз может выбрать один из следующих двух образов действия:

- Он может немедленно передать команду Notify, в которой будет сообщено о всех событиях, накопленных в списке наблюдаемых событий до инициирующего события включительно, оставив необработанные события в карантинном буфере.
- Он может попытаться освободить карантинный буфер и передать одиночную команду Notify, в которой будет сообщено о нескольких наборах событий. Тогда "строка текущего набора номера" ДОЛЖНА сбрасываться до нулевого значения после каждого инициирующего события. События, которые следуют за последним инициирующим событием, ДОЛЖНЫ оставаться в карантинном буфере.

Если шлюз передает команду Notify, конечная точка повторно входит в "состояние уведомления", в котором она остается до тех пор, пока не будет получено подтверждение (как описано выше). Если шлюз не находит подвергаемое карантину событие, инициирующее команду Notify, он вводит конечную точку в нормальное состояние. События затем обрабатываются по мере их поступления точно таким же образом, как если бы только что была принята команда Notification Request.

Шлюз может принять в любое время новую команду NotificationRequest для конечной точки, включая случай, когда конечная точка разъединена, что также вызовет вывод конечной точки из "состояния уведомления", предполагается, что команда NotificationRequest выполнена успешно. Активизация встроенной команды NotificationRequest здесь рассматривается как прием новой команды NotificationRequest, за исключением того, что текущий список параметров ObservedEvents остается немодифицированным, а не обрабатывается вновь.

Когда в "состоянии уведомления" принята новая команда NotificationRequest, шлюзу СЛЕДУЕТ попытаться доставить ждущую обработки команду Notify (следует отметить, что команда Notify, потерянная из-за разъединения, уже не рассматривается как ждущая обработки команда) до успешного ответа на новую команду NotificationRequest. Это возможно при использовании функциональных возможностей "совмещения передачи запросов и ответов" протокола и упорядочении подлежащих передаче сообщений (команд и ответов), причем первым должно быть "самое старое" сообщение. Затем сообщения будут посланы источнику новой команды NotificationRequest в одном пакете независимо от этого источника и "уведомленного объекта" для старой и новой команды. Выполняются следующие пошаговые действия:

- 1) шлюз формирует сообщение, которое переносит в одном пакете повторение старой незавершенной команды Notify и ответ на новую команду NotificationRequest;
- 2) затем конечная точка выводится из "состояния уведомления", не ожидая ответа на команду Notify;

- 3) копия незавершенной команды Notify сохраняется до тех пор, пока не будет получен ответ. Если имеет место выдержка времени, то команда Notify будет повторена в пакете, в котором также повторно переносится ответ на команду NotificationRequest:
- Если пакет, переносящий ответ на команду NotificationRequest, потеряется, то агент вызова повторно передаст команду NotificationRequest. Шлюз ответит на это повторение повторной передачей в одном пакете незавершенной команды Notify и ответа на команду NotificationRequest – эта дейтаграмма будет послана источнику команды NotificationRequest.
  - Команда (команды) Notify для данной конечной точки ДОЛЖНА доставляться по порядку. Если шлюз должен передать новую команду Notify до получения ответа на предыдущую команду Notify, он формирует пакет, совмещающий повторение старой команды Notify, повторение ответа на последнюю команду NotificationRequest и новую команду Notify – эта дейтаграмма будет передана текущему "уведомленному объекту".

После получения команды NotificationRequest параметры "список запрашиваемых событий" и "отображение цифр" (если было предоставлено новое отображение цифр) заменяются вновь полученными параметрами, а "строка текущего набора номера" сбрасывается до нулевого значения. Кроме того, когда в "состоянии уведомления" была получена новая команда NotificationRequest, то список наблюдаемых событий сбрасывается до нулевого значения. Последующее поведение тогда зависит от значения параметра QuarantineHandling. Этот параметр может определять необходимость отбрасывания подвергаемых карантину и наблюдаемых событий (в этом случае список наблюдаемых событий пуст), в таком случае все подвергаемые карантину и наблюдаемые события отбрасываются. Если же этот параметр определяет, что подвергаемые карантину и наблюдаемые события должны быть обработаны, то шлюз начнет обработку подвергаемых карантину и наблюдаемых событий, используя вновь полученный список "запрашиваемых событий" и "отображение цифр", если оно предоставлено. При обработке этих событий шлюз может встретить событие, которое инициирует передачу команды Notify. В этом случае шлюз немедленно передаст команду Notify, в которой будет сообщено о всех событиях, которые были накоплены в списке "наблюдаемых событий" до инициирующего события включительно, оставляя необработанные события в карантинном буфере. После этого конечная точка вновь входит в "состояние уведомления".

Пока шлюз накапливал события согласно предыдущим запросам на уведомление, но еще не обнаружил какие-либо события, инициирующие запрос, может быть получен новый запрос на уведомление. Обработку событий, о которых еще не послано уведомление, как и в случае подвергаемых карантину событий, определяют параметры карантинной обработки:

- Если параметр карантинной обработки определяет, что подвергаемые карантину события должны быть проигнорированы, то список наблюдаемых событий просто сбрасывается.
- Если параметр карантинной обработки определяет, что подвергаемые карантину события должны быть обработаны, то список наблюдаемых событий передается в список подвергаемых карантину событий. Список наблюдаемых событий затем сбрасывается, а список подвергаемых карантину событий обрабатывается. Единственным исключением является активизация встроенного запроса на уведомление. В этом случае список наблюдаемых событий остается немодифицированным, а не обрабатывается вновь.

Приведенная выше процедура применяется ко всем видам запросов на уведомление независимо от того, являются ли они частью команды обработки соединения или представлены в качестве команды NotificationRequest. Команды обработки соединения, не содержащие запроса на уведомление, не влияют на приведенную выше процедуру, а процедура не влияет на них.

На рисунке 5 иллюстрируются определенные выше процедуры в предположении успешного выполнения всех транзакций.



Чтобы избежать таких состязательных условий, шлюз ДОЛЖЕН проверить состояние конечной точки, прежде чем посылать ответ на команду NotificationRequest. В частности, он ДОЛЖЕН возвратить ошибку:

- 1) Если шлюз получил запрос на уведомление о переходе в состояние "телефонная трубка снята", в то время как у телефонного аппарата трубка уже снята (код ошибки 401 – телефонная трубка снята).
- 2) Если шлюз получил запрос на уведомление о состоянии "телефонная трубка положена" или "флэш-сигналы", в то время как трубка телефонного аппарата уже положена (код ошибки 402 – телефонная трубка положена).

Кроме того, в отдельных определениях сигналов может быть указано, что сигнал будет "работать" только в определенных условиях, например, вызывной сигнал возможен, если только телефонная трубка уже снята. Если для данного сигнала существуют такие предварительные условия, то шлюз ДОЛЖЕН возвращать информацию об ошибке, заданную в определении сигнала, если эти предварительные условия не соблюдаются.

Следует отметить, что проверка условия проводится во время приема запроса на уведомление, в то время как о действительном событии, вызвавшем текущее условие, либо могло быть уже сообщено, либо это событие ранее могло быть проигнорировано, либо оно в текущий момент находится "на карантине".

Другие переменные состояний шлюза, такие как список запрашиваемых событий или список запрашиваемых сигналов, полностью заменяются после каждой успешной команды NotificationRequest, что предотвращает любое долговременное несоответствие между агентом вызова и шлюзом.

Когда команда NotificationRequest неуспешна, независимо от того, включена ли она в команду обработки вызова или нет, шлюз будет просто продолжать вести себя так, будто эта команда никогда не принималась, хотя возвращается информация об ошибке. Как и все другие транзакции, команда NotificationRequest ДОЛЖНА выполняться как неделимая транзакция; таким образом, любые изменения, инициированные в результате выполнения команды, ДОЛЖНЫ быть восстановлены.

Если агент вызова получает сообщение, показывающее, что команда NotificationRequest была неуспешной, он ДОЛЖЕН предпринять действия по удостоверению, что все события, помещенные в карантин конечной точкой, либо находятся в обработке, либо отклонены, и что конечная точка возвращена в нормальный режим функционирования, где о вновь запрошенных событиях сообщается по мере их возникновения. Например, скажем, агент вызова получает в ответ на команду NotificationRequest, запрашивающей обнаружение сигнала "телефонная трубка снята", код ошибки "401 – телефонная трубка уже снята". На этот момент агент вызова должен понять, что команда NotificationRequest не имеет никакого влияния на конечную точку, и что конечная точка находится в том же состоянии, что и до получения команды. Если конечная точка помещала события в карантин в состоянии "шаговая блокировка" до получения команды NotificationRequest, то и после отправления ответа с кодом ошибки "401" она будет помещать события в карантин в состоянии "шаговая блокировка". Чтобы удостовериться, что конечная точка не осталась в состоянии постоянного помещения событий в карантин, агенту вызова следует отправить новую команду NotificationRequest с другим (возможно пустым) набором запрашиваемых событий для перевода конечной точки из состояния "шаговая блокировка" в нормальный режим работы, когда она может сообщать о новых событиях.

Другое состязательное условие может возникнуть, когда команда Notify выдается непосредственно перед приемом шлюзом команды NotificationRequest. Параметр RequestIdentifier используется для коррелирования команд Notify с командами NotificationRequest, тем самым позволяя агенту вызова определить, была ли команда Notify генерирована до или после того, как шлюз принял новую команду NotificationRequest.

#### **6.4.3.3 Семантика транзакций**

По мере того как потенциальное время выполнения транзакций возрастает, например, из-за внешнего резервирования ресурсов, становится все более важным тщательное определение семантики транзакций. В частности, проблема состязательных условий, а именно ее связь с состоянием "телефонная трубка положена/снята", должна быть определена тщательным образом.

Важным моментом является необходимость учитывать тот факт, что состояние "телефонная трубка снята/положена" может на самом деле меняться в период между началом транзакции и ее завершением. В более широком смысле можно сказать, что успешное выполнение транзакции зависит от одного или нескольких предварительных условий, которые могут изменяться динамически во время выполнения транзакции.

Простейшей семантикой является лишь выдвижение следующего требования: все предварительные условия ДОЛЖНЫ быть соблюдены с момента инициирования транзакции до ее завершения. Следовательно, если любое из предварительных условий изменяется во время выполнения транзакции, то ДОЛЖЕН иметь место отказ транзакции. Кроме того, как только транзакция инициирована, все новые события подвергаются карантину. Когда результат выполнения транзакции становится известным, то обрабатываются все подвергаемые карантину события.

В качестве примера можно рассмотреть транзакцию, которая включает запрос на событие "телефонная трубка снята". Когда транзакция инициируется, телефон находится в состоянии "телефонная трубка положена", и поэтому данное предварительное условие соблюдается. Если состояние телефонной трубки изменяется на состояние "телефонная трубка снята" до завершения выполнения транзакции, предварительное условие больше не соблюдается, и поэтому транзакция немедленно получает отказ. Событие "телефонная трубка снята" будет теперь записано в память "карантинного" буфера, который затем начинает обрабатываться.

#### 6.4.3.4 Упорядочение команд и обращение с разупорядочением

Протокол MGCP не объявляет, что базовый транспортный протокол гарантирует порядок следования команд, посылаемых шлюзу или конечной точке. Это свойство способствует максимально возможной своевременности действий, но имеет ряд недостатков, например:

- Команды Notify могут запаздывать и поступать к агенту вызова после передачи новой команды Notification Request.
- Если новая команда NotificationRequest передается до ответа на принятую предшествующую аналогичную команду, то нет гарантии, что эта предшествующая команда не будет принята под вторым номером.

Агенты вызовов и шлюзы, которые стремятся гарантировать непротиворечивое функционирование конечных точек, могут использовать приведенные ниже правила:

- 1) Когда шлюз оперирует несколькими конечными точками, команды, относящиеся к различным конечным точкам, могут посылаются параллельно, например согласно модели, где каждая конечная точка контролируется собственным процессом или собственным потоком.
- 2) Когда в одной и той же конечной точке создано несколько соединений, команды, относящиеся к различным соединениям, могут быть переданы параллельно.
- 3) Для заданного соединения обычно должна быть только одна ждущая выполнения команда (создать или модифицировать). Однако в любое время может быть подана команда DeleteConnection. Как следствие, шлюз может иногда принимать команду ModifyConnection, которая относится к ранее отключенному соединению. Такая команда ДОЛЖНА быть проигнорирована, и должна быть возвращена информация об ошибке (код ошибки 515 – неправильный идентификатор соединения).
- 4) Для заданной конечной точки в любой момент времени обычно должна быть только одна ждущая подтверждения команда NotificationRequest. Параметр RequestId используется для коррелирования команд Notify с инициирующей командой NotificationRequest.
- 5) В некоторых случаях команда DeleteConnection с неявным или явным использованием подстановочного знака, применяющаяся к группе конечных точек, может стоять перед ждущей обработки командой CreateConnection. Агент вызова должен самостоятельно исключать все соединения, завершение которых "зависало" во время выполнения глобальной команды DeleteConnection. Кроме того, новые команды CreateConnection для конечных точек, именуемые с помощью подстановочных знаков, не должны посылаются до тех пор, пока не будет получен ответ на команду DeleteConnection с подстановочным знаком.
- 6) Когда команды встроены одна в другую, ДОЛЖНЫ соблюдаться требования к порядку следования для всех команд. Например, команда CreateConnection с содержащимся в ней запросом на уведомление должна одновременно соответствовать требованиям по порядку следования для команд CreateConnection и NotificationRequest.
- 7) Команды AuditEndpoint и AuditConnection не подлежат какому-либо упорядочению.
- 8) Команда RestartInProgress, как определено процедурой перезапуска (см. п. 6.4.3.5), всегда должна быть первой командой, посылаемой конечной точкой. Любая другая команда или ответ должны быть доставлены после этой команды RestartInProgress (допускается совмещение команд и ответов).

- 9) Когда несколько сообщений совмещены в одном пакете, эти сообщения всегда обрабатываются по порядку.

Встроенные клиенты ДОЛЖНЫ придерживаться тех из упомянутых выше правил, которые определяют поведение шлюза; однако встроенный клиент НЕ ДОЛЖЕН делать никаких предположений относительно того, следуют ли агенты вызовов правилам или нет. Следовательно, шлюзы ДОЛЖНЫ всегда отвечать на команды независимо от того, придерживаются ли они упомянутых выше правил или нет.

Чтобы обеспечить согласованность операции, встроенным клиентам СЛЕДУЕТ вести себя как указано ниже, если не выполняется одно или несколько вышеперечисленных правил:

- Там, где предполагается одиночная ожидающая выполнения команда (ModifyConnection, NotificationRequest), а в новой транзакции получена такая же команда до того, как закончено выполнение старой, шлюзу СЛЕДУЕТ сбросить предыдущую команду. Сюда также входит случай, когда одна или более команд были инкапсулированы. РЕКОМЕНДУЕТСЯ использование кода ошибки 407 (транзакция отменена).
- Если команда ModifyConnection получена для ожидающей выполнения команды CreateConnection, команду ModifyConnection СЛЕДУЕТ отклонить. РЕКОМЕНДУЕТСЯ использование кода ошибки 40 (временная ошибка). Следует отметить, что такая ситуация создает программную ошибку агента вызова.

Следует обратить внимание, что если принятие новой команды ведет к отмене старой команды, старую команду СЛЕДУЕТ отменить, несмотря на то успешна новая команда или нет. Например, если команда ModifyConnection отменена командой DeleteConnection, которая сама неуспешна из-за встроенной команды NotificationRequest, команда ModifyConnection все равно отменяется.

#### 6.4.3.5 Противодействие лавине перезапусков

Предположим, что одновременно включается питание для большого числа шлюзов. Если все они должны инициировать транзакцию RestartInProgress, то весьма вероятно, что агент вызова будет сильно перегружен, что приведет к потерям сообщений и перегрузке сети во время критического периода восстановления обслуживания. Чтобы предотвратить такие лавинные явления, ДОЛЖНА выполняться следующая процедура:

- 1) Когда включается питание шлюза или когда конечная точка шлюза включается в эксплуатацию, шлюз инициирует таймер перезапуска, присваивая ему случайное значение, равномерно распределенное между 0 и максимальным обеспечиваемым значением задержки на ожидание (MWD), например 360 секунд (см. ниже). ДОЛЖНА быть проявлена определенная осторожность, чтобы избежать синхронности при генерировании случайных чисел между множеством шлюзов, которые будут использовать один и тот же алгоритм.
- 2) Шлюз затем ждет либо окончания времени таймера, либо приема команды от агента вызова, либо обнаружения действий локального пользователя, таких, например, как переход в состояние "телефонная трубка снята" в домашнем шлюзе. Уже существующее состояние "телефонная трубка снята" приводит к генерированию события "телефонная трубка снята".
- 3) Когда истекает время таймера перезапуска, принимается команда или обнаруживаются активизация либо уже существующее состояние "телефонная трубка поднята", шлюз инициирует процедуру перезапуска.

Процедура перезапуска просто устанавливает, что конечная точка ДОЛЖНА послать агенту вызова команду RestartInProgress, информируя его о перезапуске, и, кроме того, гарантировать, что первое сообщение (команда или ответ), которое видит агент вызова из данной конечной точки, ДОЛЖНО быть этой командой RestartInProgress. Для достижения этого конечная точка ДОЛЖНА в полной мере использовать механизм совмещения передачи запросов и ответов. Например, если действия по снятию телефонной трубки имеют место до истечения времени таймера перезапуска, то будет генерирован пакет, содержащий команду RestartInProgress, вместе с вложенной (совмещенной в этом пакете) командой Notify для события "телефонная трубка снята". В случае когда время таймера перезапуска истекло при отсутствии всех прочих действий, шлюз просто посылает сообщение RestartInProgress.

Следует отметить, что если команда RestartInProgress совмещена с ответом (R) на команду, принятую при перезапуске, то для передачи команды RestartInProgress не требуется совмещения с ответом R. Однако при перезапуске конечной точки повторная передача ответа R требует, чтобы команда

RestartInProgress была совмещена в целях обеспечения доставки надлежащим образом этой команды и ответа R. Процедура перезапуска завершается, когда получен успешный ответ. Если получен ответ об ошибке, то последующее поведение зависит от рассматриваемого кода ошибки:

- Если код ошибки указывает на нерегулярную ошибку (4xx), тогда процедура перезапуска ДОЛЖНА быть инициирована снова (как новая транзакция).
- Если код ошибки равен 521, тогда для конечной точки выполняется переадресация, а процедура перезапуска ДОЛЖНА быть инициирована снова (как новая транзакция). Ответ с кодом 521 должен содержать параметр NotifiedEntity, который будет обозначать "уведомленный объект", по отношению к которому иницируется перезапуск.
- Если ошибка – это любая другая постоянная ошибка (5xx), тогда РЕКОМЕНДУЕТСЯ, чтобы конечная точка больше не инициировала процедуру перезапуска самостоятельно (до перезагрузки), если только не оговорено иное. Если команда получена, то конечная точка ДОЛЖНА вновь инициировать процедуру перезапуска.

Если конечная точка входит в "несвязное" состояние при выполнении процедуры перезапуска, то ДОЛЖНА быть выполнена "несвязная" процедура, описанная в п. 6.4.3.6, а во время этой процедуры посылается сообщение "перезапуск".

Ожидается, что каждая конечная точка в шлюзе будет иметь обеспечиваемого агента вызова, то есть "уведомленный объект", для направления к нему начального сообщения перезапуска. Когда сбор множества конечных точек в шлюзе осуществляется несколькими агентами вызовов, то приведенная выше процедура должна быть выполнена для каждого множества конечных точек, управляемых данным агентом вызова. Шлюз ДОЛЖЕН в полной мере использовать подстановочные знаки для минимизации числа генерируемых сообщений RestartInProgress, когда в шлюзе перезапускается несколько конечных точек, а управление конечными точками осуществляет один и тот же агент вызова.

Значение задержки MWD является параметром конфигурации, который зависит от типа шлюза. Для определения значения этой задержки в домашних шлюзах могут использоваться следующие основания.

Агенты вызовов обычно рассчитаны на обработку трафика в часы наибольшей нагрузки, во время которых в среднем будет занято 10% линий, обслуживающих вызовы, средняя длительность которых составляет обычно 3 минуты. Обработка вызова обычно включает 5–6 транзакций между каждой конечной точкой и агентом вызова. Это простое вычисление показывает, что агент вызова, как ожидается, будет обрабатывать от 5 до 6 транзакций для каждой конечной точки в среднем каждые 30 минут, или, другими словами, примерно по одной транзакции на каждую конечную точку в среднем каждые 5–6 минут. Это позволяет предположить, что приемлемым значением задержки MWD для домашнего шлюза может быть значение от 10 до 12 минут. В отсутствие явной конфигурации встроенные клиенты ДОЛЖНЫ использовать для задержки MWD значение по умолчанию 600 секунд.

#### 6.4.3.6 Несвязные конечные точки

В дополнение к процедуре перезапуска встроенные клиенты также имеют "несвязную" процедуру, которая иницируется, когда конечная точка оказывается "несвязной" ("разъединенной"), как описано в п. 6.4.2. Здесь следует отметить, что конечные точки могут стать несвязными только в том случае, когда они пытаются взаимодействовать с агентом вызова. Конечная точка, которая становится "несвязной", выполняет следующие шаги:

- 1) "Несвязный" таймер устанавливается на случайное значение, равномерно распределенное между нулем и обеспечиваемой начальной задержкой на ожидание для "несвязного" состояния ( $T_{d_{init}}$ ), равной, например, 15 секундам. ДОЛЖНА быть проявлена некоторая осторожность, чтобы избежать синхронности при генерировании случайных чисел между множеством шлюзов и конечными точками, которые будут использовать один и тот же алгоритм.
- 2) Затем шлюз ждет либо окончания времени таймера, либо приема команды от агента вызова, либо обнаружения действий локального пользователя для конечной точки, таких, например, как переход в состояние "телефонная трубка снята".
- 3) Когда время "несвязного" таймера истекает, принимается команда или обнаруживаются действия локального пользователя, шлюз ДОЛЖЕН инициировать "несвязную" процедуру с новым идентификатором транзакции для конечной точки. В случае действий локального пользователя с момента, когда шлюз стал разъединенным ("несвязным"), или с момента, когда он в последний раз закончил "несвязную процедуру", должно, кроме того, истечь время обеспечиваемой минимальной задержки на ожидание для "несвязного" состояния ( $T_{d_{min}}$ ), чтобы ограничить скорость выполнения процедуры.

- 4) Если "несвязная процедура" все еще оставляет конечную точку в "несвязном" состоянии, то выбирается новое значение для "несвязного" таймера. Значение таймера ДОЛЖНО быть выбрано из диапазона, определяемого значением времени последнего таймера, умноженным на 1,5, с одной стороны, и умноженным на 2, с другой стороны; значение таймера из этого диапазона МОЖЕТ выбираться случайным образом. В любом случае новое значение таймера зависит от времени обеспечиваемой максимальной задержки на ожидание для "несвязного" состояния ( $T_{d_{max}}$ ), например 600 секунд, и шлюз заново выполняет действия, начиная с пункта 2.

"Несвязная" процедура аналогична процедуре перезапуска в том, что она просто устанавливает, что конечная точка ДОЛЖНА послать агенту вызова команду RestartInProgress, информируя его о том, что конечная точка была разъединена, и, кроме того, гарантировать, что первое сообщение (команда или ответ), которое теперь видит агент из данной конечной точки, ДОЛЖНО обязательно быть этой командой RestartInProgress. Во время каждого инициирования "несвязной" процедуры при выполнении данной команды ДОЛЖНЫ соблюдаться требования к нормальной повторной передаче и идентификаторам транзакций (см. п. 6.4.2). Для этого в конечной точке ДОЛЖНО оптимальным образом использоваться совмещение передачи запросов и ответов.

По получению сообщения RestartInProgress, содержащего метод перезапуска "несвязной" процедуры, агент вызова ДОЛЖЕН предпринять действия по удостоверению, что все события, помещенные конечной точкой в карантин, обработаны или отклонены, и что конечная точка возвращена в нормальный режим работы, где о вновь запрашиваемых событиях сообщается по мере их возникновения. Агенту вызова СЛЕДУЕТ послать команду NotificationRequest, содержащий параметр карантинной обработки, установленный в этом случае на "отклонить".

Агент вызова может также решить выполнить одно или несколько из следующих действий: проверка конечной точки, сброс всех соединений конечной точки или отправка команды NotificationRequest, запрашивающей конечную точку об обработке сообщений, помещенных в карантин (см. п. 6.4.3.7).

Следует отметить, что если в момент получения команды "несвязная" процедура уже выполняется, существующая процедура разъединения ДОЛЖНА быть завершена и ДОЛЖНА быть запущена новая процедура. Это необходимо для того, чтобы поддержать возможную переадресацию агента вызова.

Следует также отметить, что "несвязная" конечная точка не означает "выведенная из эксплуатации" конечная точка. Несвязная – это не состояние доступности услуг конечной точки, а скорее индикатор того, что конечная точка не в состоянии связаться с агентом вызова.

Следует также отметить, что если команда RestartInProgress совмещена с ответом (R) на полученную при разъединении команду, то повторная передача команды RestartInProgress не требует совмещения передачи ответа R. Однако, пока конечная точка является несвязной (разъединена), при повторной передаче ответа R требуется совмещение передачи команды RestartInProgress с ответом R, а также удостоверение доставки надлежащим образом команды и ответа.

"Несвязная" процедура завершается, как только получен успешный ответ. Ответы об ошибках обрабатываются подобно процедуре перезапуска (см. п. 6.4.3.5). Если после передачи ответа об ошибке должна быть вновь инициирована "несвязная" процедура, то по-прежнему применяются описанные выше положения, касающиеся таймера, ограничивающего скорость перезапуска. Несвязная конечная точка может выразить желание послать команду (помимо команды RestartInProgress), находясь в разъединенном состоянии. Это действие окажется успешным только, когда агент вызова снова станет достижимым; при этом возникает вопрос, что, тем временем, делать с такой командой. Одна крайность заключается в том, что конечная точка могла бы сразу отбросить эту команду; однако это было бы не вполне правильным действием, если агент вызова был действительно доступен, но конечная точка еще не завершила "несвязную" процедуру (можно рассмотреть, например, случай, когда команда NotificationRequest была только что получена, и это сразу же привело к генерированию команды Notify). Для предупреждения таких сценариев разъединенные (несвязные) конечные точки НЕ ДОЛЖНЫ "вслепую" отбрасывать новые команды, подлежащие передаче за период в  $T_{s_{max}}$  секунд, после того как они получают команду "не проверять". Одним из способов удовлетворения этого требования является использование временной буферизации команд, подлежащих передаче, однако при этом конечная точка должна гарантировать, что она:

- не сформирует длинную очередь команд, подлежащих передаче;
- не перегрузит агента вызова вследствие быстрой передачи слишком большого количества команд, как только она вновь станет связной.

Буферизация команд на  $T_{s_{max}}$  секунд и, как только конечная точка снова становится связной, ограничение скорости передачи буферизированных команд на одну ожидающую обработки команду для каждой конечной точки считается надежным. Если в течение  $T_{s_{max}}$  секунд конечная точка не стала связной, но в течение  $T_{s_{max}}$  секунд инициируется "несвязная" процедура, конечная точка МОЖЕТ совместить передачу буферизированной команды (команд) с этой командой RestartInProgress. Следует отметить, что если команда была послана, независимо от того, была ли она сначала помещена в буфер или совмещена при предыдущей передаче, повторная передача этой команды ДОЛЖНА быть прекращена за период в  $T_{s_{max}}$  секунд после начальной передачи, как описано в п. 6.4.2. В настоящей Рекомендации специально не описывается какое-либо дополнительное поведение для несвязной конечной точки. Например, поставщики МОГУТ предпочесть предоставить паузы (молчание), воспроизвести тональный сигнал переупорядочения или даже инициировать воспроизведение загружаемого файла звуковых сигналов в затронутых конечных точках.

Значение по умолчанию для  $T_{d_{init}}$  составляет 15 секунд, для  $T_{d_{min}}$  – 15 секунд, а для  $T_{d_{max}}$  – 600 секунд.

#### 6.4.3.7 Обработка агентом вызова несвязных конечных точек

Когда конечная точка находится в "несвязном" состоянии, она может накопить большое число событий в карантинном буфере. Также "несвязная" конечная точка может автономно отключить установленные соединения (скажем, при перезагрузке шлюза). Следовательно, если связность между "несвязной" конечной точкой и ее агентом вызова в дальнейшем восстанавливается, агент вызова ДОЛЖЕН быть готов к следующим ситуациям:

- Большое число сообщений Notify (уведомить), которые могут быть сгенерированы конечной точкой, если обрабатываются все события из карантинного списка;
- Получение старых/устаревших событий, о которых сообщено конечной точкой, не имеющих больше никакого значения. В карантинном буфере установлен порядок очередности FIFO (первый вошел, первый вышел), при котором наиболее поздние события будут обработаны первыми и о них будет сообщено (если был запрос) агенту вызова. Действие, предпринятое агентом вызова при получении старого события, может не иметь никакого значения, если старое событие было заменено более новыми событиями (например, событие "набора цифр" перестало бы иметь значение, если бы в дальнейшем конечная точка перешла в состояние телефонная трубка положена);
- Несовпадение состояния соединения между агентом вызова и конечной точкой, когда агент вызова думает, что у конечной точки одно или больше соединений, а в действительности у конечной точки нет соединений.

Агент вызова свободен в выборе механизма, поддерживаемого протоколом, по обработке вышеуказанных ситуаций. Одной из возможностей являются следующие действия:

- 1) Определим новую булеву переменную, названную "disconnect-event-sync" (синхронизация несвязных событий), содержащуюся в агенте вызова для каждой его конечной точки. При установке в состояние "истина" эта переменная показывает, что связность с "несвязной" конечной точкой недавно была восстановлена, но синхронизация событий/сигналов еще не достигнута.

ПРИМЕЧАНИЕ. – Данная переменная представлена для описания поведения агента вызова и не предназначена для применения какой-либо конкретной реализации. Извне переменная невидима.

- а) Как только агент вызова получает информацию, что конечная точка несвязна, он устанавливает "disconnect-event-sync" на значение истина. "Несвязная" процедура удостоверяет, что агент вызова получит информацию о том, что конечная точка становится несвязной, через получение сообщения "disconnected" команды RestartInProgress. Если конечная точка получает положительное уведомление для "несвязного" RSIP, она выполняет "несвязную" процедуру. В этот момент конечная точка могла бы немедленно сгенерировать команду Notify (уведомление) по двум причинам: чтобы послать команду Notify, которая была помещена в буфер в то время, когда конечная точка была отсоединена, или если, будучи отсоединена, конечная точка была в состоянии "уведомления" или режиме "шлейфа", то для того, чтобы поместить событие, инициирующее уведомление, в карантинный список:

- Если конечная точка работает в "шаговом" режиме, то ответ на команду Notify не активизирует в ней генерирование каких-либо дальнейших сообщений Notify (для этого понадобится дополнительная команда NotificationRequest).

- Однако если конечная точка работает в режиме "цикла", то ответ на команду Notify вызовет генерирование дальнейших сообщений Notify. Как обсуждалось выше, иногда это нежелательно, поскольку события, о которых сообщается, могут быть устаревшими, а несколько событий может быть помещено в карантин, что в свою очередь, вызовет большое количество сообщений Notify и последующих сообщений NotificationRequest, базирующихся на устаревшей информации.
- b) До тех пор пока "disconnect-event-sync" конечной точки установлен на значение истина, агенту вызова следует действовать, чтобы удостовериться, что потенциально большое число событий в карантинном списке либо отклонено, либо обрабатывается контролируемым и надлежащим образом. Это может быть достигнуто различными методами:
- Агент вызова может послать одиночную команду NotificationRequest, указывающую, что все подлежащие карантину события должны быть отклонены. При получении положительного подтверждения этой команды или при получении команды Notify с тем же самым RequestIdentifier, агенту вызова следует установить индикатор "disconnect-event-sync" на значение ложь, в этот момент возобновляется регулярная обработка событий для конечной точки. Обратной стороной данного подхода является то, что он сбрасывает любые накопленные события, несмотря на число накопленных событий. В некоторых случаях это может привести к необходимости прерывания услуги. Для адресации прерывания будут необходимы расширения протокола.
  - Агент вызова может послать команду NotificationRequest, указывающую, что помещенные в карантин события должны быть обработаны. Если конечная точка работает в "пошаговом" режиме, то она сообщит о каждом единичном событии, инициирующем уведомление, с помощью одной полученной команды NotificationRequest, в то время как режим "цикл", может сообщить о нескольких событиях, с помощью одной команды NotificationRequest.

Поскольку информация, переданная событиями уведомления может больше не иметь никакого значения, агенту вызова не следует "вслепую" обрабатывать такие события (например, будучи уведомленным о событии "телефонная трубка снята", агенту вызова не следует автоматически посылать команду NotificationRequest, чтобы применить тональный сигнал готовности к набору номера и запрашивать уведомление о событии "телефонная трубка снята"). Вместо этого агент вызова должен синхронизировать свои внутреннее состояние данных с актуальным состоянием конечной точки. Поскольку конечная точка предоставляет обработку помехи приему вызова сигналом/событием против текущего состояния линии при переключении телефонной трубки, агент вызова может обнаружить текущее состояние переключения телефонной трубки, базируясь на ответе на команду NotificationRequest. Например, ответ "402 – телефонная трубка уже положена" на команду NotificationRequest, запрашивающей об обнаружении события "телефонная трубка положена", означает, что линия в настоящий момент уже в состоянии "телефонная трубка положена". Агент вызова может выбрать игнорирование событий уведомления, которые становятся незначимыми, базируясь на текущем состоянии телефонной трубки (например, игнорирование события "цифр", если линия в настоящий момент в состоянии "телефонная трубка положена").

Как только все элементы из карантинного списка были обработаны, агенту вызова следует установить индикатор "disconnect-event-sync" на значение ложь. Агент вызова может с уверенностью предположить, что все помещенные в карантин события были выполнены, если истекла задержка продолжительности  $T_{hist}$ , с того момента как агент вызова в последний раз подсказывал конечной точке обрабатывать следующее сообщение (т. е. если задержка продолжительности  $T_{hist}$  истекла с того момента, как агент вызова послал ответ на предыдущую команду Notify в режиме "цикл", или с того момента, как агент вызова получил последний положительный ответ на команду NotificationRequest в "пошаговом" режиме).

- 2) Если конечная точка становится несвязной, соединения, созданные на базе этой конечной точки не должны быть затронуты. Однако всегда есть вероятность, что соединение больше не сможет поддерживаться конечной точкой и, следовательно, отключается; это выразится в том,

что агенту вызова будет послана команда DeleteConnection. Когда конечная точка отсоединена, такая команда может никогда не быть получена агентом вызова, в этом случае он не узнает, что соединение удалено. Следовательно, когда бы агент вызова не выяснил, что конечная точка не соединена, ему следует проверить конечную точку по всему списку соединений, существующих на ней.

## 6.5 Коды завершения и коды ошибок

На все команды по протоколу MGCP принимается ответ. В ответе содержится код завершения, указывающий на состояние команды. Код завершения – это целое число, для которого определены пять диапазонов значений:

- значение 000 указывает на подтверждение ответа<sup>25</sup>;
- значение между 100 и 199 указывает на временный ответ;
- значение между 200 и 299 указывает на успешное завершение;
- значение между 400 и 499 указывает на нерегулярную ошибку;
- значение между 500 и 599 указывает на постоянную ошибку.

В таблице 4 перечислены значения, которые были определены.

**Таблица 4/J.162 – Определения кодов завершения**

Код	Значение
000	Подтверждение ответа.
100	Транзакция выполняется в текущее время. Сообщение о фактическом завершении последует позднее.
200	Запрошенная транзакция была выполнена нормально.
250	Соединение (соединения) было отключено.
400	Транзакция не могла быть выполнена из-за нерегулярной ошибки.
401	Телефон уже находится в состоянии "телефонная трубка снята".
402	Телефон уже находится в состоянии "телефонная трубка положена".
407	Транзакция отменена. Транзакция была отменена каким-то внешним действием, например, командой ModifyConnection, отмененной командой DeleteConnection.
500	Транзакция не могла быть выполнена, так как конечная точка неизвестна.
501	Транзакция не могла быть выполнена, так как конечная точка не готова.
502	Транзакция не могла быть выполнена, так как конечная точка не имеет достаточных ресурсов.
503	Подстановочный знак "all of" ("все из") полностью не поддерживается. Транзакция содержала подстановочный знак; однако шлюз не поддерживает этого полностью. Следует отметить, что в текущий момент это допустимо только для непустых NotificationRequests.
505	Неподдерживаемый параметр RemoteConnectionDescriptor. Его СЛЕДУЕТ использовать, когда в параметре RemoteConnectionDescriptor не поддерживаются один или несколько обязательных параметров или значений.
506	Невозможно одновременно соответствовать параметру LocalConnectionOptions и параметру RemoteConnectionDescriptor. Этот код СЛЕДУЕТ использовать, когда параметры LocalConnectionOptions и RemoteConnectionDescriptor содержат один или несколько обязательных параметров или значений, которые вступают в конфликт друг с другом и/или не могут быть поддержаны в одно и то же время (за исключением отказа при согласовании кодеков – см. код ошибки 534).
508	Неизвестная или не поддерживаемая карантинная обработка.
510	Транзакция не могла быть выполнена, так как была обнаружена ошибка протокола.
511	Транзакция не могла быть выполнена, так как команда содержала нераспознанное расширение.
512	Транзакция не могла быть выполнена, так как шлюз не оборудован для обнаружения одного из запрошенных событий.

<sup>25</sup> Подтверждение ответа используется для временных ответов (см. п. 7.8).

**Таблица 4/J.162 – Определения кодов завершения**

Код	Значение
513	Транзакция не могла быть выполнена, так как шлюз не оборудован для генерирования одного из запрошенных сигналов.
514	Транзакция не могла быть выполнена, так как шлюз не может передать заданное уведомление.
515	Транзакция относится к неправильному идентификатору соединения (возможно уже отключенному).
516	Транзакция относится к неизвестному идентификатору вызова.
517	Неподдерживаемый или недействительный режим.
518	Неподдерживаемый или неизвестный пакет.
519	Конечная точка не имеет отображения цифр.
520	Транзакция не могла быть выполнена, так как конечная точка "перезапускается".
521	Конечная точка переадресована другому агенту вызова.
522	Нет такого события или сигнала.
523	Неизвестное действие или недопустимая комбинация действий.
524	Внутреннее несоответствие в параметре LocalConnectionOptions.
525	Неизвестное расширение в параметре LocalConnectionOptions.
526	Недостаточная ширина полосы пропускания.
527	Пропуск параметра RemoteConnectionDescriptor.
528	Несовместимая версия протокола.
529	Внутренний отказ аппаратных средств.
532	Неподдерживаемое значение (значения) в параметре LocalConnectionOptions.
533	Ответ слишком велик.
534	Отказ при согласовании кодеков.
538	Ошибка параметра события/сигнала (например, отсутствующий, ошибочный, неподдерживаемый, неизвестный и т. п.).

### **6.6 Коды причины**

Коды причины используются шлюзом при отключении соединения для информирования агента вызова о причине отключения соединения. Они также могут быть использованы в команде RestartInProgress для информирования агента вызова о причине перезапуска. Код причины – это целое число. В таблице 5 перечислены определенные значения кода причины.

**Таблица 5/J.162 – Определения кода причины**

Код	Значение
000	Состояние конечной точки нормальное. (Этот код используется только в ответ на запросы проверки).
900	Неправильное функционирование конечной точки.
901	Конечная точка выведена из эксплуатации.
902	Потеря связности нижнего уровня (например, синхронизации в нисходящем направлении передачи).
903	Резервирование ресурсов для QoS потеряно.

## 6.7 Использование опций локального соединения и дескрипторов соединения

Нормальная последовательность установления двустороннего соединения включает по меньшей мере три этапа:

- 1) Агент вызова запрашивает первый шлюз "создать соединение" в конечной точке. Шлюз распределяет ресурсы для этого соединения и отвечает на команду путем предоставления "описания сеанса" (называемого параметром LocalConnectionDescriptor). Описание сеанса содержит информацию, необходимую другой стороне для передачи пакетов в направлении к вновь созданному соединению.
- 2) Затем агент вызова запрашивает второй шлюз "создать соединение" в конечной точке. Команда содержит "описание сеанса", предоставленное первым шлюзом (называемое теперь параметром RemoteConnectionDescriptor). Шлюз распределяет ресурсы для этого соединения и отвечает на команду собственным "описанием сеанса" (параметр LocalConnectionDescriptor).
- 3) Агент вызова использует команду "модифицировать соединение" для предоставления этого второго "описания сеанса" (называемого теперь параметром RemoteConnectionDescriptor) для первой конечной точки. Как только это будет сделано, связь может осуществляться в обоих направлениях.

Когда агент вызова выдает команду CreateConnection (создать соединение) или ModifyConnection (модифицировать соединение), имеются три параметра, которые определяют медиаданные, поддерживаемые этим соединением:

- Параметр LocalConnectionOptions: предоставляется агентом вызова для контроля параметров медиаданных, используемыми шлюзом для соединения. Когда данный параметр предоставлен, шлюз должен соответствовать этим параметрам медиаданных до тех пор, пока либо не будет отключено соединение, либо не будет получена команда ModifyConnection.
- Параметр RemoteConnectionDescriptor: предоставляется агентом соединения для переноса параметров медиаданных, поддерживаемых другой стороной соединения. Когда данный параметр предоставлен, шлюз должен соответствовать этим параметрам медиаданных до тех пор, пока либо не будет исключено соединение, либо не будет получена команда ModifyConnection.
- Параметр LocalConnectionDescriptor: предоставляется шлюзом агенту вызова для переноса параметров медиаданных, которые он поддерживает для соединения. Когда данный параметр предоставлен, шлюз должен соответствовать параметрам медиаданных до тех пор, пока либо соединение не будет исключено, либо шлюз не выдаст новый параметр LocalConnectionDescriptor. В дополнение к параметрам медиаданных, приписанных к соединению, шлюз может также уведомлять о дополнительно поддерживаемых возможностях в LocalConnectionDescriptor. Следует обратить внимание, что такие возможности ДОЛЖНЫ быть обеспечены за пределами линии "m=" в SDP. Шлюз свободно уведомлять о всех своих поддерживаемых возможностях в независимости от полученных от агента вызова параметров LCO или RCD, и в независимости от параметров медиаданных, обусловленных соединением.

Выбор кодека и периода пакетирования должен осуществляться, как описано в этом пункте, только в том случае, когда:

- a) шлюз принимает команду CRCX; или
- b) шлюз принимает команду MDCX и присутствует любой из следующих параметров:
  - метод кодирования (a: в параметре LocalConnectionOptions);
  - период пакетирования (p: в параметре LocalConnectionOptions);
  - множественный период пакетирования (mp: в параметре LocalConnectionOptions);
  - параметр RemoteConnectionDescriptor.

Кроме того, в процессе выбора кодека и периода пакетирования должна использоваться только информация, содержащаяся в запросе на соединение, и не должны сохраняться какие-либо значения, которые могли быть приняты в предыдущих запросах на соединение.

Например, если шлюз принял команду MDCX со всеми необходимыми параметрами LCO, но параметр RemoteConnectionDescriptor был пропущен, шлюз будет осуществлять согласование кодеков, как будто параметр RemoteConnectionDescriptor никогда не принимался для этого соединения. Если все из вышеупомянутых параметров будут опущены в команде MDCX, существующие согласованные кодеки и периоды пакетирования останутся прежними.

При определении, какие кодек (кодеки) и период (периоды) пакетирования должны предоставляться в параметре LocalConnectionDescriptor, шлюзу необходимо рассмотреть три следующих списка кодеков и периодов пакетирования:

- Список кодеков и периодов пакетирования, допустимых параметром LocalConnectionOptions. Кодек допускается параметром LocalConnectionOptions, если он удовлетворяет ограничениям, определенным в полях метода кодирования, периода пакетирования и множественным периодам пакетирования. Если одно или несколько из этих полей опущены, то пропущенные поля не налагают никаких ограничений на допустимые кодеки.
- Список кодеков и периодов пакетирования в параметре RemoteConnectionDescriptor.
- Внутренний список кодеков и периодов пакетирования, которые может поддерживать шлюз для соединения. Шлюз может поддерживать для данного соединения один или несколько кодеков и периодов пакетирования.

Выбор кодеков (включая все соответствующие параметры медиаданных) тогда может быть описан поэтапно следующим образом:

- 1) Утвержденный список кодеков/периодов пакетирования образуется в результате пересечения внутреннего списка кодеков/периодов пакетирования и кодеков/периодов пакетирования, допускаемых параметром LocalConnectionOptions. Если параметр LocalConnectionOptions не был предоставлен, тогда утвержденный список кодеков/периодов пакетирования содержит внутренний список. Если параметр LocalConnectionOptions был предоставлен, но параметр кодеков был опущен, то параметр LocalConnectionOptions косвенно допускает наличие всех кодеков во внутреннем списке, при условии что они не являются несовместимыми с любым заданным периодом (периодами) пакетирования. Аналогичным образом, если параметр LocalConnectionOptions был предоставлен, но период (периоды) пакетирования был опущен, то параметр LocalConnectionOptions неявным образом содержит множество периодов пакетирования, поддерживаемых внутренним списком.
- 2) Если утвержденный список кодеков/периодов пакетирования пуст, то имеет место отказ при согласовании кодеков, и генерируется ответ об ошибке (рекомендуется код ошибки 534 – отказ при согласовании кодеков).
- 3) В противном случае согласованный список кодеков/периодов пакетирования образуется в результате пересечения утвержденного списка кодеков/периодов пакетирования и кодеков/периодов пакетирования, допускаемых параметром RemoteConnectionDescriptor. Если параметр RemoteConnectionDescriptor не был предоставлен, тогда согласованный список кодеков/периодов пакетирования содержит утвержденный список кодеков/периодов пакетирования. Если параметр RemoteConnectionDescriptor не содержит каких-либо линий медиапотоков, то фиксируется отказ при согласовании кодеков и генерируется ответ об ошибке (рекомендуется код ошибки 534, отказ при согласовании кодеков). Если параметр RemoteConnectionDescriptor содержит множество медиапотоков, то адаптеру МТА СЛЕДУЕТ принять только один из этих потоков и отклонить другие потоки, установив значение их порта на "нуль" в параметре LocalConnectionDescriptor. Если параметр RemoteConnectionDescriptor был предоставлен, но период (периоды) пакетирования был опущен, то согласованный список периодов пакетирования будет содержать несколько периодов пакетирования из утвержденного списка. Если период пакетирования явно исключен как из параметра LocalConnectionOptions, так и из параметра RemoteConnectionDescriptor, адаптер МТА ДОЛЖЕН выбрать приемлемые значения по умолчанию в соответствии со стандартом RFC 2327.
- 4) Если согласованный список кодеков/периодов пакетирования пуст, то фиксируется отказ при согласовании кодеков и генерируется ответ об ошибке (рекомендуется код ошибки 534, отказ при согласовании кодеков).
- 5) В противном случае согласование кодеков считается успешным, и согласованный список кодеков/пакетов пакетирования возвращается в параметр LocalConnectionDescriptor.

Следует отметить, что как параметр LocalConnectionOptions, так и параметр RemoteConnectionDescriptor может содержать список кодеков, упорядоченных по уровню приоритета. Когда предоставлены оба этих параметра, шлюзу следует придерживаться порядка приоритетов, указанного в параметре LocalConnectionOptions. Следует также отметить, что по приведенной выше процедуре согласуются как методы кодирования, так и периоды пакетирования в отличие от согласования только методов кодирования. Это сделано для активизирования согласованной операции по обеспечению QoS на локальном и дальнем конце соединения в модели с распределенным по участкам QoS, используемой в проекте IPCablecom.

В случае когда шлюз поддерживает несколько кодеков для одной конечной точки, имеются два варианта, которые может использовать шлюз при решении вопроса о том, сколько кодеков он хочет поддерживать для этого соединения:

- 1) Шлюз поддерживает несколько кодеков и может осуществлять переключение между различными кодеками в режиме реального времени. Шлюз возвращает все согласованные кодеки в линию медиапотока SDP и резервирует наименьшую верхнюю границу (LUB) согласно Рекомендации МСЭ-Т J.163. Граница LUB резервируется для гарантирования того, что переключение на любой из этих кодеков будет выполнено успешно. Несколько кодеков в линии = m означает, что устройство должно быть готово к принятию пакетов медиаданных от любого из согласованных кодеков. Шлюз также может посылать пакеты медиаданных от любого из согласованных кодеков и осуществлять, если требуется, переключение между ними.
- 2) Шлюз поддерживает один или несколько кодеков, но не может осуществлять переключение между различными кодеками в режиме реального времени. Поэтому шлюз согласует и возвращает только один кодек в линию медиапотока SDP (необязательно шлюз также помещает дополнительные поддерживаемые кодеки в атрибуте 'X-rc-codecs' протокола SDP) и резервирует ширину полосы пропускания для одного согласованного кодека в линии медиапотока SDP согласно Рекомендации МСЭ-Т J.163. По этому методу изменение кодека должно инициироваться системой SMS, чтобы выполнить изменение кодеков, во время которого, согласно Рекомендации МСЭ-Т J.163, переустанавливается результирующее изменение ширины полосы пропускания.

### 6.7.1 Согласование RFC 2833

Внутренний список поддерживаемых кодеков ДОЛЖЕН включать в себя кодек телефонных событий с событиями в диапазоне от 0-15. Это будет гарантировать, что реле DTMF RFC 2833 будет использовано для соединения, если оно авторизовано LCO (через включение его в параметр или через пустой параметр a:) и разрешено RCD.

Примером авторизованного через LCO реле DTMF RFC 2833 является:

L: a:PCMU;PCMA;telephone-event, mp:10;20;-

Кодек телефонных событий (telephone-event codec) НЕ ДОЛЖЕН быть единственным кодеком из SMS, поддерживаемым в LCO. Если МТА получает LCO, содержащий только кодек телефонных событий, он ДОЛЖЕН вернуть код ошибки 524 – внутренняя несогласованность в параметре LocalConnectionOptions. Если расширенный, как описано в п. 6.7, список кодеков, как описано в п. 6.7, содержит только кодек телефонных сообщений, то конечная точка ДОЛЖНА вернуть код ошибки 534—отказ согласования кодеков. Подобным же образом, если согласованный, как описано в п. 6.7, список кодеков содержит только кодек телефонных событий, конечная точка ДОЛЖНА возвращать код ошибки 534.

Если для соединения используется период пакетирования параметра LCO, конечная точка ДОЛЖНА использовать эту скорость пакетирования для пакетов реле DTMF. Если в LCO используется множественный параметр периода пакетирования, SMS ДОЛЖЕН использовать дефис для указания скорости пакетирования для кодека телефонных событий. Если конечная точка получает LCO с множественным параметром периода пакетирования со скоростью пакетирования для кодека телефонных событий, не установленной на символ дефис, конечная точка ДОЛЖНА возвращать код ошибки 524 - несогласованный LCO. Если конечная точка возвращает LCD, который включает в себя возможность получать кодек телефонных событий, она ДОЛЖНА использовать в качестве скорости пакетирования дефис в атрибуте mptime SDP.

Примером LCD, который уведомляет о поддержке набора DTMF RFC 2833, является:

v=0

o=- 4723891 7428910 IN IP4 128.96.63.25

s=-

```
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0 8 105
a=mptime:10 20 -
a=rtpmap:105 telephone-event/8000/1
```

За более подробной информацией по использованию набора DTMF RFC 2833 обращаться к МЭС-Т Рек. J.161.

### 6.7.2 Выбор интервала пакетирования кодека для T.38

Интервал пакетирования для T.38 выбирается с использованием той же процедуры, что и для аудиокодеков. См. п. 6.7 выше.

Для T.38 требуется специальный анализ D-QoS. Если T.38 разрешен и используется D-QoS, CMS СЛЕДУЕТ рассмотреть требования к ширине полосы пропускания для T.38 на конкретном периоде пакетирования при его расчете LUB, с целью авторизации достаточной ширины полосы пропускания в случае последующего переключения на T.38. Данная скорость пакетирования известна CMS, либо по умолчанию, либо через конфигурацию. Если CMS позже инструктирует конечную точку перейти на T.38 посредством команды MDCX, используя тот же самый вентиль, период пакетизации, использованный для T.38, либо указанный явно в параметре Local Connection Options, либо принятый по умолчанию, ДОЛЖЕН быть предварительно авторизован через ресурсы D-QoS. Если текущий вентиль не может удовлетворить запрос МТА на ширину полосы пропускания, CMS ДОЛЖЕН установить новый вентиль, способный удовлетворить данный запрос и передать МТА новый GateID в команде MDCX.

Однако конечная точка НЕ ДОЛЖНА включать T.38 в ее расчет LUB до получения команды CRCX или MDCX при запросе D-QoS, если только при процедуре согласования Local Connection Options кодеков не выбрано "изображение/t38" ("image/t38"). Если в качестве метода кодирования в Local Connection Options команды CRCX или MDCX указано только "изображение/t38", то конечная точка ДОЛЖНА рассматривать T.38 для данного периода пакетирования только в своем расчете LUB. Даже если удаленная сторона и может посылать пакеты T.38 с более высоким периодом пакетирования, чем авторизованный локальным агентом вызова, как определено в J.161, конечная точка НЕ ДОЛЖНА рассматривать эти более высокие периоды пакетирования при своем расчете использования ширины полосы пропускания в нисходящем направлении передачи. Данная ширина должна уже быть достаточной для поддержки потенциально более высокого периода пакетирования.

### 6.7.3 Согласование удаленных IP и порта

Удаленные IP и порт предоставляются через параметр Remote Connection Descriptor. Однажды получив их в команде обработки успешного соединения (например, команда Modify Connection), конечная точка ДОЛЖНА продолжать их использование либо до получения нового Remote Connection Descriptor, который укажет новые удаленные IP и порт, либо до отключения соединения. Следует отметить, что получение команды Modify Connection без параметра Remote Connection Descriptor, которое может сбить согласование кодека, не делает недействительными текущие значения удаленных IP и порта, даже если тип медиаданных изменен с аудио на изображение или наоборот.

## 7 Протокол контроля медийных шлюзов

В протоколе MGCP интерфейс контроля медийным шлюзом реализуется в виде множества транзакций. Эти транзакции состоят из команды и обязательного ответа. Имеется восемь типов команд.

- CreateConnection;
- ModifyConnection;
- DeleteConnection;
- NotificationRequest;
- Notify;
- AuditEndpoint;
- AuditConnection;

- RestartInProgress.

Первые четыре команды агент вызова посылает шлюзу. Команда Notify посылается агенту вызова шлюзом. Шлюз может также посылать команду DeleteConnection, как определено в п. 6.3.6. Агент вызова может посылать шлюзу любую из команд Audit и, наконец, шлюз может послать агенту вызова команду RestartInProgress.

## 7.1 Общее описание

Все команды состоят из заголовка команды, за которым, для некоторых команд, может следовать описание сеанса.

Все ответы состоят из заголовка ответа, за которым, для некоторых команд, может следовать описание сеанса.

Заголовки и описания сеансов кодируются в виде множества текстовых строк, разделенных знаком возврата каретки и знаком смещения строки (или, необязательно, одним знаком смещения строки). Заголовки отделяются от описания сеанса пустой строкой.

В протоколе MGCP используется идентификатор транзакции со значением от 1 до 999999999 для корреляции команд и ответов. Идентификатор транзакции кодируется как компонент заголовка команды и повторяется как компонент заголовка ответа.

## 7.2 Заголовок команды

Заголовок команды состоит из:

- строки команды, определяющей запрашиваемое действие или команду, идентификатора транзакции, конечной точки, относительно которой запрашивается действие, и версии протокола MGCP;
- множества строк параметров, составленных из имени параметра, за которым следует значение параметра.

Каждый компонент в заголовке команды нечувствителен к регистру, если в других стандартах, на которые даются ссылки, не указано или не установлено иное. Это относится к командам, а также к параметрам и значениям, и при всех сравнениях интерпретация верхнего и нижнего регистров, а также их комбинаций, ДОЛЖНА быть равнозначной.

### 7.2.1 Строка команды

Строка команды состоит из:

- имени запрашиваемой команды;
- идентификации транзакции;
- имени конечной точки (точек), которая должна выполнять команду (в уведомлениях или перезапусках – имени конечной точки (точек), которая выдает команду);
- версии протокола.

Эти четыре элемента кодируются в виде строк печатных знаков в коде ASCII, разделенных пробелами, то есть знаками пробела (0x20) или знаками табуляции (0x09) в коде ASCII. Для встроенных клиентов СЛЕДУЕТ использовать один разделитель пробелами в коде ASCII; однако встроенные клиенты ДОЛЖНЫ быть способны проводить синтаксический анализ сообщений с дополнительными знаками пробелов.

#### 7.2.1.1 Кодирование запрашиваемых команд

Запрашиваемые команды кодируются в виде четырехбуквенных кодов ASCII верхнего и/или нижнего регистров (сравнения ДОЛЖНЫ быть нечувствительными к регистру), как определено в таблице 6.

**Таблица 6/J.162 – Кодирование запрашиваемых команд**

<b>Команда</b>	<b>Код</b>
CreateConnection	CRCX
ModifyConnection	MDCX
DeleteConnection	DLCX
NotificationRequest	RQNT
Notify	NTFY
AuditEndpoint	AUEP
AuditConnection	AUCX
RestartInProgress	RSIP

В будущих версиях настоящей Рекомендации могут быть определены новые команды. В экспериментальных целях может оказаться необходимым использовать новые команды, прежде чем они будут разрешены в какой-либо версии настоящей Рекомендации. Такие "экспериментальные" команды должны идентифицироваться четырехбуквенным кодом, начинающимся с буквы X (например, XPER).

Встроенный клиент, который принимает команду, содержащую "экспериментальную" команду, которую он не поддерживает, ДОЛЖЕН вернуть информацию об ошибке (код ошибки 511 – неопознанное расширение).

#### **7.2.1.2 Идентификаторы транзакций**

Идентификаторы транзакций используются для корреляции команд и ответов.

Встроенный клиент поддерживает два отдельных пространства имен идентификаторов транзакций:

- пространство имен идентификаторов транзакций для передачи транзакций; и
- пространство имен идентификаторов транзакций для приема транзакций.

Идентификаторы транзакций для команд, посылаемых данному встроенному клиенту, ДОЛЖНЫ быть однозначно определены, как минимум, для максимального времени существования транзакций в пределах множества агентов вызовов, которые управляют этим встроенным клиентом (см. п. 7.5). Таким образом, независимо от передающего агента вызова встроенные клиенты всегда могут обнаружить дублированные транзакции путем простой проверки идентификатора транзакции. Однако, вопросы координации этих идентификаторов транзакций между агентами вызовов выходят за рамки настоящей Рекомендации.

Идентификаторы транзакций для всех команд, посылаемых от данного встроенного клиента, ДОЛЖНЫ быть однозначно определены для максимального времени существования транзакций (см. п. 7.5) независимо от того, какому агенту вызова была послана команда. Таким образом, агент вызова всегда может обнаружить дублированную транзакцию от встроенного клиента по комбинации доменного имени конечной точки и идентификатора транзакции. В свою очередь, встроенный клиент всегда может обнаружить дублированное подтверждение ответа по идентификатору транзакции (идентификаторам транзакций).

Идентификатор транзакции кодируется в виде строки, содержащей до девяти десятичных цифр. В строках команд он следует непосредственно за кодированием команды.

Идентификаторы транзакций имеют значения между 1 и 999999999. В идентификаторах транзакций не следует использовать начальные нули. Равенство базируется на числовом значении, и начальные нули игнорируются. Объект протокола MGCP НЕ ДОЛЖЕН повторно использовать идентификатор транзакции раньше, чем через три минуты после завершения выполнения предыдущей команды, в которой был использован этот идентификатор.

#### **7.2.1.3 Кодирование имен конечных точек, агентов вызовов и уведомленных объектов**

Имена конечных точек и имена агентов вызовов кодируются в виде адресов электронной почты, как определено в стандарте RFC 821. В этих адресах доменное имя определяет систему, где прикреплена конечная точка, а левая сторона – конкретную конечную точку в этой системе. Оба компонента ДОЛЖНЫ быть нечувствительны к регистру.

Примеры таких имен приведены в таблице 7.

**Таблица 7/J.162 – Пример кодирования имени**

aaln/1@ncs2.whatever.net	Аналоговая линия доступа 1 во встроенном клиенте ncs2 в сети "whatever"(любая).
Call-agent@ca.whatever.net	Агент вызова для сети "whatever" (любая).

Имя уведомленных объектов выражается в том же синтаксисе с возможным добавлением номера порта, как, например:

Call-agent@ca.whatever.net:5234

В случае когда номер порта опущен, по умолчанию будет использоваться порт 2727 агента вызова по протоколу MGCP. В п. 6.1.1 содержится дополнительная подробная информация, касающаяся имен конечных точек.

#### 7.2.1.4 Кодирование версии протокола

Версия протокола кодируется в виде ключевого слова "MGCP", за которым следуют пробел и номер версии, который сопровождается именем профиля "NCS" и номером версии профиля. Номера версий состоят из номера главной версии, точки и номера последующей (вторичной) версии. Номера главной и последующей версий кодируются как десятичные числа. Номер версии профиля, определяемый в настоящей Рекомендации, равен 1.0.

Версия протокола настоящей Рекомендации ДОЛЖНА быть закодирована следующим образом:

MGCP 1.0 NCS 1.0

Часть "NCS 1.0" указывает на то, что это профиль NCS 1.0 версии протокола MGCP 1.0.

Объект, принимающий команду с версией протокола, которую он не поддерживает, ДОЛЖЕН отвечать информацией об ошибке (код ошибки 528 – несовместимая версия протокола).

#### 7.2.2 Строки параметров

Строки параметров составлены из имени параметра, которые в большинстве случаев состоят из одного символа верхнего регистра, за которым следуют двоеточие, пробел и значение параметра. Хотя имена параметров и значения по-прежнему нечувствительны к регистру. Параметры, которые могут присутствовать в командах, определены в таблице 8.

**Таблица 8/J.162 – Определения параметров**

Имя параметра	Код	Значение параметра
ResponseAck <sup>26</sup>	K	См. описание.
CallId	C	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака. Идентификаторы вызова сравниваются как строки, а не как численные значения.
ConnectionId	I	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака. Идентификаторы соединения сравниваются как строки, а не как численные значения.
NotifiedEntity	N	Идентификатор в формате RFC 2821, состоящий из произвольной строки и доменного имени запрашивающего объекта, возможно, заканчивающийся именем порта, как, например: Call-agent@ca.whatever.net:5234 .
RequestIdentifier	X	Шестнадцатеричная строка; длина НЕ ДОЛЖНА превышать 32 знака.
LocalConnectionOptions	L	См. описание.

<sup>26</sup> Параметр ResponseAck не был рассмотрен в п. 6.3, поскольку идентификаторы транзакций не видны в приведенном примере интерфейса API. Реализаторы могут выбрать другой подход.

**Таблица 8/J.162 – Определения параметров**

<b>Имя параметра</b>	<b>Код</b>	<b>Значение параметра</b>
Connection Mode	M	См. описание.
RequestedEvents	R	См. описание.
SignalRequests	S	См. описание.
DigitMap	D	Текстовое кодирование отображения цифр.
ObservedEvents	O	См. описание.
ConnectionParameters	P	См. описание.
ReasonCode	E	См. описание.
SpecificEndPointId	Z	Идентификатор в формате кода RFC 2821, состоящий из произвольной строки, необязательно сопровождаемой знаком "@", за которым следует доменное имя встроенного клиента, за которым закреплена эта конечная точка.
MaxEndPointIds	ZM	Десятичная строка; длина НЕ ДОЛЖНА превышать 16 знаков.
NumEndpoints	ZN	Десятичная строка; длина НЕ ДОЛЖНА превышать 16 знаков.
RequestedInfo	F	См. описание.
QuarantineHandling	Q	См. описание.
DetectEvents	T	См. описание.
EventStates	ES	См. описание.
ResourceID	DQ-RI	См. описание.
RestartMethod	RM	См. описание.
RestartDelay	RD	Число секунд, закодированное в виде десятичного числа.
Capabilities	A	См. описание.
VersionSupported	VS	См. описание.
MaxMGCPDatagram	MD	См. описание.

Параметры не обязательно присутствуют во всех командах. В таблице 9 показана связь между параметрами и командами. Буква М означает "обязательный", буква О – "необязательный", а буква F – "запрещенный".

**Таблица 9/J.162 – Связь параметров с командами**

<b>Имя параметра</b>	<b>CRCX</b>	<b>MDCX</b>	<b>DLCX</b>	<b>RQNT</b>	<b>NTFY</b>	<b>AUEP</b>	<b>AUCX</b>	<b>RSIP</b>
ResponseAck	O	O	O	O	O	O	O	O
CallId	M	M	O	F	F	F	F	F
ConnectionId	F	M	O	F	F	F	M	F
RequestIdentifier	O	O	O	M	M	F	F	F
LocalConnectionOptions	O	O	F	F	F	F	F	F
Connection Mode	M	O	F	F	F	F	F	F
RequestedEvents	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	F	F	F	F
SignalRequests	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	F	F	F	F
NotifiedEntity	O	O	O	O	O	F	F	F
ReasonCode	F	F	O	F	F	F	F	O
ObservedEvents	F	F	F	F	M	F	F	F
DigitMap	O	O	O	O	F	F	F	F
Параметры соединения	F	F	O	F	F	F	F	F
SpecificEndpointId	F	F	F	F	F	O	F	F
MaxEndPointIds	F	F	F	F	F	O	F	F

**Таблица 9/J.162 – Связь параметров с командами**

<b>Имя параметра</b>	<b>CRCX</b>	<b>MDCX</b>	<b>DLCX</b>	<b>RQNT</b>	<b>NTFY</b>	<b>AUEP</b>	<b>AUCX</b>	<b>RSIP</b>
NumEndPoints	F	F	F	F	F	F	F	F
RequestedInfo	F	F	F	F	F	O	O	F
QuarantineHandling	O	O	O	O	F	F	F	F
DetectEvents	O	O	O	O	F	F	F	F
EventStates	F	F	F	F	F	F	F	F
ResourceID	F	F	F	F	F	F	F	F
RestartMethod	F	F	F	F	F	F	F	M
RestartDelay	F	F	F	F	F	F	F	O
Capabilities	F	F	F	F	F	F	F	F
VersionSupported	F	F	F	F	F	F	F	F
MaxMGCPDatagram	F	F	F	F	F	F	F	F
RemoteConnectionDescriptor	O	O	F	F	F	F	F	F
а) Параметры RequestedEvents и SignalRequests являются необязательными в команде NotificationRequest. Если эти параметры опущены, то соответствующие списки будут считаться пустыми. Для команд обработки соединения пропуск этих двух параметров, когда команда содержит параметр RequestIdentifier, означает, что соответствующие списки будут считаться пустыми.								

Встроенным клиентам и агентам вызова СЛЕДУЕТ всегда предоставлять обязательные параметры перед необязательными; однако встроенные клиенты НЕ ДОЛЖНЫ попадать в ситуацию сбоя, если данная рекомендация не выполняется.

Если у реализаторов возникает необходимость в экспериментировании с новыми параметрами, например, при разработке нового применения протокола MGCP, они должны идентифицировать эти параметры по именам, которые начинаются со строки "X-" или "X+", как, например:

X-FlowerOfTheDay: Daisy

Имена параметров, начинающиеся с "X+", являются расширениями обязательных параметров. Шлюз, принимающий расширение обязательного параметра, которое ему непонятно, ДОЛЖЕН отвечать информацией об ошибке (код ошибки 511 – неопознанное расширение).

Имена параметров, начинающиеся с "X-", являются расширениями некритических параметров. Шлюз, принимающий расширение некритического параметра, которое ему непонятно, может без всякого риска проигнорировать такой параметр.

Следует отметить, что "экспериментальные" команды имеют форму *XABC*, в то время как "экспериментальные" параметры имеют форму X-ABC.

Если получена строка параметров с запрещенным параметром или с какой-либо другой ошибкой форматирования, то принимающий объект должен ответить кодом наиболее характерной ошибки для рассматриваемой ошибки. Код наименее характерной ошибки 510 – ошибка протокола. Всегда может быть предоставлен текстовый комментарий.

### **7.2.2.1 Подтверждение ответа**

Параметр подтверждения ответа используется для поддержки установления трехсторонней связи, описанного в п. 7.7.. Он содержит разделенный запятыми список "диапазонов подтвержденных идентификаторов транзакций".

Каждый "диапазон подтвержденных идентификаторов транзакций" состоит либо из одного десятичного числа, когда диапазон содержит только одну транзакцию, либо из двух десятичных чисел, разделенных одним знаком дефиса, которые описывают нижнее и верхнее значения идентификаторов транзакций, содержащихся в этом диапазоне.

Примером подтверждения ответа является:

К: 6234-6255, 6257, 19030-19044

### 7.2.2.2 Параметр RequestIdentifier

Параметр RequestIdentifier (идентификатор запроса) коррелирует команду Notify с иницилирующей ее командой NotificationRequest. Параметр RequestIdentifier – это шестнадцатеричная строка, ее длина НЕ ДОЛЖНА превышать 32 знака. Параметры RequestIdentifiers сравниваются как строки, а не как численные значения. Строка "0" зарезервирована для сообщения об устойчивых событиях в случае, когда параметр NotificationRequest еще не получен (см. п. 6.3.2).

### 7.2.2.3 Параметр LocalConnectionOptions

LocalConnectionOptions (опции локальных соединений) описывают необязательные параметры, которые шлюз по команде агентов вызовов использует для соединения. Этими параметрами являются:

- период пакетирования в миллисекундах, закодированный в виде ключевого слова "p", за которым следуют двоеточие и десятичное число;
- период многократного пакетирования в миллисекундах для каждого кодека в параметре LCO метода кодирования, закодированный в виде ключевого слова "mp", за которым следуют двоеточие и список десятичных чисел или дефисов, с одним элементом для каждого элемента в поле "метод кодирования". Каждое значение периода пакетирования отделяется от последующего элемента одной точкой с запятой. Первым элементом в списке ДОЛЖНО быть десятичное число. Последующие элементы в списке ДОЛЖНЫ быть либо десятичным числом, либо знаком дефиса;
- литеральное имя алгоритма сжатия, как определено в Рекомендации МСЭ-Т J.161, кодируется в виде ключевого слова "a", за которым следуют двоеточие и строка символов. Если агент вызова указывает список значений, эти значения должны быть разделены точкой с запятой. Для RTP аудиокодеки ДОЛЖНЫ указываться с использованием имен кодирования, определенных в профиле RTP AV (RFC 1890), имен кодирования, зарегистрированных в IANA, или имен кодирования, на которые ссылается или определяет МСЭ-Т Рек. J.161. Не-аудио медиаданные, зарегистрированные как тип MIME ДОЛЖНЫ использовать форму "<MIME тип>/<MIME подтип>", как в "изображении/t38". РЕКОМЕНДУЕТСЯ, чтобы также поддерживались хорошо известные варианты литеральных имен кодеков;
- параметр "компенсация эха", закодированный в виде ключевого слова "e", за которым следуют двоеточие и значение "on" (включено) или "off" (выключено);
- параметр "тип услуги", закодированный в виде ключевого слова "t", за которым следуют двоеточие и значение, закодированное в виде двух шестнадцатеричных цифр;
- параметр "подавление пауз", закодированный в виде ключевого слова "s", за которым следуют двоеточие и значение "on" (включено) или "off" (выключено).

Параметрами LocalConnectionOptions, используемыми для динамического качества обслуживания, являются:

- Параметр D-QoS GateID, закодированный в виде ключевого слова "dq-gi", за которым следуют двоеточие и строка, содержащая до 8 шестнадцатеричных знаков, соответствующая 32-битовому идентификатору для GateID (идентификатор логического вентиля);
- параметр D-QoS Resource Reservation (резервирование ресурсов для QoS), закодированный в виде ключевого слова "dq-tr", за которым следуют двоеточие и строка знаков. Может быть определен список значений, и в этом случае значения будут разделяться точкой с запятой. Возможные значения приведены в таблице 10;

**Таблица 10/J.162 – Значения параметров резервирования ресурсов для QoS**

Режим	Значение
sendresv	Резервирование ресурсов только в направлении передачи
recvresv	Резервирование ресурсов только в направлении приема
snrcresv	Резервирование ресурсов в направлениях передачи и приема
sendcomt	Предоставление ресурсов только в направлении передачи
recvcomt	Предоставление ресурсов только в направлении приема
snrccomt	Предоставление ресурсов в направлениях передачи и приема

- параметр ResourceID кодируется в виде ключевого слова "dq-ri", за которым следуют двоеточие и строка, содержащая до 8 шестнадцатеричных знаков, соответствующая 32-битному идентификатору для ResourceID;
- параметр ReserveDestination кодируется в виде ключевого слова "dq-rd", за которым следуют двоеточие и IP-адрес, закодированный аналогично IP-адресу для части доменного имени в имени конечной точки. Параметр ReserveDestination может необязательно сопровождаться двоеточием и максимум 5 десятичными знаками для используемого номера порта UDP.

Параметры LocalConnectionOptions, используемые для обеспечения защиты, кодируются следующим образом:

- шифрокомплект протокола RTP кодируется в виде ключевого слова "sc-rtp", за которым следуют двоеточие и строка шифрокомплекта протокола RTP, как определено ниже. Может быть определен список значений, и в этом случае значения ДОЛЖНЫ быть разделены одной точкой с запятой;
- шифрокомплект протокола RTCP кодируется в виде ключевого слова "sc-rtcp", за которым следуют двоеточие и строка шифрокомплекта протокола RTP, как определено ниже. Может быть определен список значений, и в этом случае значения ДОЛЖНЫ быть разделены одной точкой с запятой.

Строки шифрокомплектов протоколов RTP и RTCP подчиняются следующей грамматике:

```

шифрокомплект = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]
AuthenticationAlgorithm = 1*( ALPHA / DIGIT / "-" / "_" )
EncryptionAlgorithm = 1*( ALPHA / DIGIT | "-" / "_" )
    
```

где ALPHA и DIGIT определены в стандарте RFC 2234.

Пробелы НЕ ДОЛЖНЫ посылаться через шифрокомплект или между смежными шифрокомплектами, если предоставляются множественные шифрокомплекты. Следующий пример иллюстрирует форматирование шифрокомплекта и списка шифрокомплектов:

```
sc-rtp:62/51;60/50
```

Фактический список шифрокомплектов, поддерживаемых в проекте IPcablecom, содержится в Рекомендации МСЭ-TJ.170.

Когда присутствуют несколько параметров, то значения разделяются запятой. Включение параметра без значения ДОЛЖНО рассматриваться как ошибка (код ошибки 524 – несоответствие параметру LocalConnectionOptions).

Примерами вариантов локальных соединений могут служить:

```

L: p:10, a:PCMU
L: p:10, a:PCMU, e:off, t:20, s:on
L: p:30, a:G729, e:on, t:A0, s:off
    
```

Шестнадцатеричное значение "20" типа услуги означает приоритет 1 по IP-протоколу, а шестнадцатеричное значение типа услуги "A0" – приоритет 5 по IP-протоколу.

Это множество атрибутов может быть расширено атрибутами расширений. Атрибуты расширений состоят из имени атрибута, за которым следует двоеточие, и списка значений атрибутов, разделенных

точкой с запятой. Имя атрибута ДОЛЖНО начинаться с двух символов "x+" для обязательного расширения или "x-" для необязательного расширения. Если шлюз принимает атрибут обязательного расширения, который он не может распознать, он ДОЛЖЕН отклонить команду с информацией об ошибке (код ошибки 525 – неизвестное расширение в параметре LocalConnectionOptions).

#### 7.2.2.4 Параметр Capabilities

Параметр Capabilities (возможности) информирует агента вызова при проверке о возможностях конечной точки. Кодирование возможностей базируется на кодировании опций локальных соединений для параметров, которые являются общими для обоих. Кроме того, параметр Capabilities может также содержать список поддерживаемых пакетов и список поддерживаемых режимов.

Используются следующие параметры:

- период пакетирования в миллисекундах, закодированный в виде ключевого слова "p", за которым следуют двоеточие и десятичное число. Может быть определен диапазон в виде двух десятичных чисел, разделенных знаком дефиса;
- литеральное имя алгоритма сжатия, закодированное в виде ключевого слова "a", за которым следуют двоеточие и строка знаков. ДОЛЖНЫ использоваться литеральные имена, определенные в Рекомендации МСЭ-Т J.161. Может быть определен список значений; в этом случае значения будут разделяться точкой с запятой;
- ширина полосы пропускания в килобитах в секунду (1000 битов в секунду), закодированная в виде ключевого слова "b", за которым следуют двоеточие и десятичное число. Может быть определен диапазон в виде двух десятичных чисел, разделенных знаком дефиса;
- параметр "компенсация эха", закодированный в виде ключевого слова "e", за которым следуют двоеточие и значение "on" ("включено"), если компенсация эха поддерживается; в противном случае – значение "off" ("выключено");
- параметр "тип услуги", закодированный в виде ключевого слова "t", за которым следуют двоеточие и значение "0", если тип услуги не поддерживается; все другие значения указывают на поддержку типа услуги;
- параметр "подавление пауз", закодированный в виде ключевого слова "s", за которым следуют двоеточие и значение "on" ("включено"), если подавление пауз поддерживается; в противном случае – значение "off" ("выключено");
- пакеты событий, поддерживаемые этой конечной точкой, закодированные в виде ключевого слова "v", за которым следуют двоеточие и список поддерживаемых имен пакетов, разделенных точкой с запятой. Первое определенное значение будет значением пакета по умолчанию для этой конечной точки;
- режимы соединений, поддерживаемые этой конечной точкой, закодированные в виде ключевого слова "m", за которым следуют двоеточие и список поддерживаемых режимов соединения, разделенных точкой с запятой, как определено в п. 7.2.2.7;
- ключевое слово "dq-gi", если поддерживается динамическое качество обслуживания;
- ключевое слово "sc-rtp", за которым следуют двоеточие и список шифрокомплектов протокола RTP, разделенных точкой с запятой, использующих то же кодирование, что и в параметре LocalConnectionOptions;
- ключевое слово "sc-rtcp", за которым следуют двоеточие и список шифрокомплектов протокола RTCP, разделенных точкой с запятой, использующих то же кодирование, что и в параметре LocalConnectionOptions.

Когда присутствует несколько параметров, то значения разделяются запятой.

Примерами возможностей являются:

```
A: a:PCMU; p:10-30, e:on, s:off, v:L;S,  
m:sendonly;recvonly;sendrecv;inactive  
A: a:G729; p:10-20, e:on, s:off, v:L;S,  
m:sendonly;recvonly;sendrecv;inactive  
A: a:G729; p:30-90, e:on, s:on, v:L;S,  
m:sendonly;recvonly;sendrecv;inactive;confrnce,  
dq-gi, sc-rtp: 64/51; 60/51, sc-rtcp: 71/81
```

Следует отметить, что кодеки и алгоритмы обеспечения безопасности являются просто примерами. Действительные кодеки и поддерживаемые алгоритмы, а также используемое кодирование, подробно

рассматриваются в отдельных Рекомендациях проекта IPCalcom. Следует также отметить, что каждый набор возможностей предоставляется в одной строке. В приведенных выше примерах каждый набор возможностей показан на нескольких строках из-за ограничений форматирования документов, налагаемых настоящей Рекомендацией.

### 7.2.2.5 Параметры соединения

Параметры соединения кодируются в виде строки пар "тип и значение", где тип – это один из кодов, приведенных в таблице 11, а значение является десятичным целым числом. Типы отделяются от значений знаком "=". Параметры отделяются друг от друга запятой.

Таблица 11/J.162 – Параметры соединения

Имя параметра соединения	Код	Значение параметра соединения
Переданные пакеты	PS	Число пакетов, которые были переданы по соединению
Переданные октеты	OS	Число октетов, которые были переданы по соединению
Полученные пакеты	PR	Число пакетов, которые были получены по соединению
Полученные октеты	OR	Число октетов, которые были получены по соединению
Потерянные пакеты	PL	Число пакетов, которые не были получены по соединению, определяемое подсчетом пропусков порядковых номеров
Флуктуация	J	Средняя флуктуация времени между поступлениями пакетов в миллисекундах, выраженная целым числом
Задержка	LA	Средняя задержка в миллисекундах, выраженная целым числом
Переданные удаленные пакеты	PC/RPS	Число пакетов, которые были переданы по соединению, с точки зрения удаленной конечной точки
Переданные удаленные октеты	PC/ROS	Число октетов, которые были переданы по соединению, с точки зрения удаленной конечной точки
Потерянные удаленные пакеты	PC/RPL	Число пакетов, которые не были получены по соединению, определяемое подсчетом пропусков порядковых номеров, с точки зрения удаленной конечной точки
Удаленная флуктуация	PC/RJI	Средняя флуктуация времени между поступлениями пакетов в миллисекундах, выраженная целым числом, с точки зрения удаленной конечной точки

Расширенные имена параметров соединения состоят из строки "X-", за которой следует двух- или трехбуквенное расширенное имя параметра. Агенты вызова, которые принимают нераспознанные расширения, ДОЛЖНЫ безоговорочно игнорировать эти расширения. Если конечная точка принимает пакеты RTCP с этими статистическими оценками, она должна в ответ на команды DeleteConnection (исключить соединение) и AuditConnection (проконтролировать соединение) вернуть параметры с определением Remote (удаленный)(коды параметров Rxx, приведенных выше).

Примером кодирования параметра соединения является:

P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48, PC/RPS=0, PC/ROS=0, PC/RPL=0, PC/RJI=0

### 7.2.2.6 Коды причины

Коды причины – это трехзначные численные значения. Код причины необязательно сопровождается пробелом и комментарием, например:

E: 900 Неправильное функционирование конечной точки

Список кодов причины содержится в п. 6.6.

### 7.2.2.7 Режим соединения

Режим соединения описывает режим функционирования соединения. Возможные значения представлены в таблице 12.

Таблица 12/J.162 – Режим соединения

Режим	Значение
M: sendonly	Шлюзу следует только передавать пакеты.
M: recvonly	Шлюзу следует только принимать пакеты.
M: sendrecv	Шлюзу следует передавать и принимать пакеты.
M: confnrc	Шлюзу следует передавать и принимать пакеты согласно режиму конференцсвязи.
M: inactive	Шлюзу не следует ни посылать, ни принимать пакеты.
M: replcate	Шлюзу следует только передавать пакеты согласно режиму копирования.
M: netwloop	Шлюзу следует вводить конечную точку в режим сетевого шлейфа.
M: netwtest	Шлюзу следует вводить конечную точку в режим проверки целостности сети.

### 7.2.2.8 Кодирование имен событий/сигналов

Имена событий/сигналов состоят из необязательного имени пакета, отделенного от имени действительного события косой чертой (/). Имя события может необязательно сопровождаться знаком "собачка" (@) и идентификатором соединения, в котором должно наблюдаться событие. Имена событий используются в параметрах RequestedEvents, SignalRequests, DetectEvents, ObservedEvents и EventStates. Каждое событие идентифицируется кодом события. Эти виды кодирования по коду ASCII являются нечувствительными к регистру. Такие значения, как "hu", "Hu", "HU" или "hU", должны считаться эквивалентными.

В таблице 13 приводятся примеры имен событий:

Таблица 13/J.162 – Примеры имен событий

L/hu	Переход в состояние "телефонная трубка положена" на примере линейного пакета
L/0	Цифра 0 на примере линейного пакета
Hf	Флэш-сигналы в предположении, что приводимый в качестве примера линейный пакет – это пакет по умолчанию для конечной точки
L/rt@0A3F58	Контроль посылки вызова по соединению "0A3F58"

Кроме того, в параметрах RequestedEvents и DetectEvents (но не в параметрах SignalRequests, ObservedEvents или EventStates) вместо индивидуальных имен событий могут быть использованы диапазон событий и обозначение с подстановочными знаками. В таблице 14 приводятся примеры допустимого диапазона и обозначения с подстановочными знаками.

Таблица 14/J.162 – Диапазон событий и обозначение с подстановочными знаками

L/[0-9]	Цифры от 0 до 9 на примере линейного пакета
L/X	Цифры от 0 до 9 на примере линейного пакета
[0-9*#A-D]	Все цифры и буквы на примере линейного пакета (по умолчанию для конечной точки)
L/all	Все события на примере линейного пакета

И наконец, знак "звездочка" может быть использован для обозначения "всех соединений", а знак доллара – "текущего" соединения. В таблице 15 содержатся примеры допустимого использования обозначений со знаками доллара и звездочки:

**Таблица 15/J.162 – Обозначение "всех" соединений и "текущего" соединения**

L/ma@*	Событие начала медиаданных RTP по всем соединениям для конечной точки
L/rt@\$	Контроль отправки вызова по текущему соединению

Начальное множество пакетов событий для встроенных клиентов содержится в Приложении А.

### 7.2.2.9 Параметр RequestedEvents

Параметр RequestedEvents (запрашиваемые события) предоставляет список событий, которые были запрошены. В Приложении А приводится описание кодов, определяемых в настоящее время.

Каждое событие может быть определено запрашиваемым действием или списком действий. Не все действия можно сочетать (допустимые сочетания действий описаны в п. 6.3.1). Действия при описании кодируются в виде списка ключевых слов, заключенных в круглые скобки и разделенных запятыми. В таблице 16 представлены коды для различных действий.

**Таблица 16/J.162 – Запрашиваемые действия для событий**

Действие	Код
Немедленное уведомление	N
Накопление	A
Накопление согласно отображению цифр	D
Игнорирование	I
Удерживание сигнала(ов) активным(и)	K
Встроенное действие NotificationRequest	E
Встроенное действие ModifyConnection	C

Если при описании действия "накопление согласно отображению цифр" отображение цифр номера не представлено, конечная точка просто использует свое текущее отображение цифр. Если в текущее время конечная точка не имеет отображения цифр, ДОЛЖНА быть возвращена информация об ошибке (код ошибки 519 – нет отображения цифр).

Если действие не описывается, то действием по умолчанию является уведомление о событии. Это означает, например, что, обозначения "ft" и "ft(N)" эквивалентны. События, которые не внесены в список, отбрасываются, за исключением устойчивых событий.

Действие с отображением цифр набора номера может быть определено только для цифр, букв и таймеров.

Список запрашиваемых событий кодируется одной строкой с группами событий/действий, разделенных запятыми. Примерами кодировок параметра RequestedEvents являются (при использовании примера линейного пакета):

R: hu (N) , hf (N)                      Уведомление о состоянии "телефонная трубка  
положена", уведомление о флэш-сигналах.  
R: hu (N) , [0-9#T] (D)                Уведомление о состоянии "телефонная трубка  
положена", накопление цифр согласно отображению  
цифр.

Встроенное действие NotificationRequest соответствует формату:

E ( R( <RequestedEvents> ), D( <Digit Map> ), S( <SignalRequests> ) )

где каждый параметр R, D, и S является необязательным и, возможно, предоставляется в другом порядке. Следующий пример иллюстрирует использование встроенного действия NotificationRequest на примере линейного пакета:

R: hd(A, E(S(d1), R( oc(N), [0-9#T] (D) ), D((1xxxxxxxxxxx|9011x.T)) ) )

Это означает: послать сигнал "телефонная трубка снята", приступить к накоплению событий, послать сигнал готовности к набору номера и начать накопление цифр набора номера согласно полученному отображению цифр. Прекратить передачу сигнала готовности к набору

номера при вводе первой цифры или, если цифра не введена до выдержки времени сигнала готовности к набору номера, завершить операцию Notify (уведомить). В противном случае уведомить о состоянии "телефонная трубка снята" и накопленных цифрах, когда имеет место соответствие или несоответствие отображению цифр, или выдержка времени между цифрами. Следует отметить, что поскольку событие "телефонная трубка положена" является устойчивым событием, оно все равно будет обнаруживаться, и о нем будет посылаться уведомление, хотя это здесь и не описано.

Встроенное действие ModifyConnection соответствует формату:

```
C(M(<ConnectionMode1>( <ConnectionID1> )) , ... ,
M(<ConnectionModen>(ConnectionIDn )))
```

Следующий пример иллюстрирует использование встроенного действия ModifyConnection на примере линейного пакета:

```
R: hf(A, C(M(inactive(X43DC)), M(sendrecv($)))) , oc(N) , of(N)
```

Это означает: послать флэш-сигнал, изменить режим соединения для соединения "X43DC" на "неактивный", затем изменить режим соединения для "текущего соединения" на "передача/прием". Уведомить о событиях, указав на состояние "операция завершена" и "ошибка операции".

### 7.2.2.10 Параметр SignalRequests

Параметр SignalRequests (запросы сигналов) предоставляет имя сигналов, которые были запрошены. В Приложении А приводится описание сигналов, определяемых в настоящее время. Данный сигнал может появиться в списке только один раз, а все сигналы по определению будут применяться в одно и то же время. Адаптер МТА ДОЛЖЕН поддерживать, как минимум, один сигнал в каждой конечной точке и одновременно поддерживать генерирование одного сигнала в каждом соединении для данной конечной точки. Конкретные пакеты МОГУТ предоставлять требования сверх этих минимальных возможностей. Для комбинаций сигналов, предоставляющих требования сверх этого минимума, которые не поддерживаются адаптером МТА, СЛЕДУЕТ вернуть код ошибки 502.

Некоторые сигналы могут быть описаны параметрами сигналов. Когда сигнал описывается несколькими сигнальными параметрами, эти параметры разделяются запятыми. Каждый сигнальный параметр ДОЛЖЕН соответствовать формату, описанному ниже (допустимы пробелы):

```
signal-parameter      = signal-parameter-value | signal-parameter-name
                        ="signal-parameter-value | signal-parameter-name
                        "(" signal-parameter-list ")"
signal-parameter-list = signal-parameter-value 0*( "," signal-parameter-
                        value )
```

где signal-parameter-value (значение параметра сигнала) может быть либо строкой, либо строкой в кавычках, то есть строкой, заключенной в две двойные кавычки. Две последовательных двойных кавычки в строке в кавычках, приводят к потере двойных кавычек в этой строке, заключенной в кавычки. Например, строка "ab""c" дает строку ab"c .

Каждый сигнал имеет один из следующих связанных с ним типов сигналов (см. п. 6.3.1):

- включено/выключено (OO);
- выдержка времени (TO);
- короткий сигнал (BR).

Сигналы "включено/выключено" могут быть параметризованы знаком "+" для включения сигнала или знаком "-" для выключения сигнала. Если сигнал "включено/выключено" не параметризован, то по умолчанию сигнал включен. В двух следующих случаях сигнал vmwi из примера линейного пакета будет включен:

```
vmwi(+), vmwi
```

Сигналы выдержки времени могут быть параметризованы сигнальным параметром "TO" и значением выдержки времени, которое заменяет значение выдержки времени по умолчанию. Если сигнал выдержки времени не параметризован значением выдержки времени, будет использоваться значение

выдержки времени по умолчанию. В двух следующих случаях будет применяться в течение 6 секунд вызывной сигнал из примера линейного пакета:

```
rg(to=6000)
rg(to(6000))
```

Отдельные сигналы могут определять дополнительные параметры сигналов.

Параметры сигналов будут заключаться в круглые скобки, как в примере (в предположении, что "Line"(линейный) – это пакет по умолчанию):

S: ci(10/14/17/26, " 555 1212", CableLabs).

Когда запрошено несколько сигналов, их коды разделяются запятой, как, например:

S: rg, rt@FDE234C8.

### 7.2.2.11 Параметр ObservedEvents

Параметры ObservedEvents (наблюдаемые события) предоставляют список событий, которые наблюдались. Коды событий те же, что и коды событий, используемые в команде NotificationRequest. Когда событие обнаруживается в соединении, наблюдаемое событие идентифицирует соединение, в котором было обнаружено событие, используя синтаксис "@<соединение>". Примерами наблюдаемых событий, использующих пример линейного пакета, являются:

```
O: hu
O: ma@A43B81
O: 8,2,9,5,5,5,5,T
O: hf,hf,hu
O: 8,2,9,5,mt,5,5,5,T
```

О событиях, которые были накоплены согласно отображению цифр набора номера, сообщается как об отдельных событиях в том порядке, в каком они были обнаружены. Другие события могут появляться между ними. Следует отметить, что если "строка текущего набора номера" с частичным соответствием отображению цифр не является пустой и имеет место другое событие, которое приводит к генерированию сообщения Notify, "строка текущего набора номера" с частичным соответствием будет включена в список наблюдаемых событий, а затем будет удалена; подробная информация содержится в п. 6.4.3.1.

### 7.2.2.12 Параметр RequestedInfo

Параметр RequestedInfo (запрашиваемая информация) содержит список кодов параметров, разделенных запятой, как определено в п. 7.2.2. В п. 6.3.8 перечисляются параметры, которые могут быть проверены. Также поддерживаются значения, перечисленные в таблице 17:

Таблица 17/J.162 – Значения параметра RequestedInfo

Параметр RequestedInfo	Код
LocalConnectionDescriptor	LC
RemoteConnectionDescriptor	RC

Например, если возникнет желание проверить параметры NotifiedEntity, RequestIdentifier, RequestedEvents, SignalRequests, DigitMap, DetectEvents, EventStates, LocalConnectionDescriptor и RemoteConnectionDescriptor, параметр RequestedInfo будет иметь значение:

```
F: N,X,R,S,D,T,ES,LC,RC
```

Запрос возможностей для команды AuditEndPoint кодируется с помощью кода параметра "A", как в случае:

```
F: A
```

### 7.2.2.13 Параметр QuarantineHandling

Параметр QuarantineHandling (карантинная обработка) содержит список разделенных запятыми ключевых слов:

- Ключевое слово "process" (обработать) или "discard" (отбросить) используется для указания на характер обработки подвергаемых карантину и наблюдаемых событий. Если ключевые слова "process" или "discard" отсутствуют, то предполагается обработка.
- Ключевое слово "step" (шаг) или "loop" (цикл) используется для указания на то, ожидается ли, как максимум, одно уведомление или допустимо несколько уведомлений. Если ключевые слова "step" или "loop" отсутствуют, то предполагается ключевое слово "step". Поддержка этих двух ключевых слов является обязательной.

Следующие значения являются примерами корректных значений:

```
Q: loop
Q: process
Q: loop, discard
```

### 7.2.2.14 Параметр DetectEvents

Параметр DetectEvents (обнаруживаемые события) кодируется в виде списка событий, разделенных запятыми, как, например:

```
T: hu,hd,hf,[0-9#*]
```

Следует отметить, что с этими событиями не могут быть связаны какие-либо действия.

### 7.2.2.15 Параметр EventStates

Параметр EventStates (состояние событий) кодируется в виде списка разделенных запятыми событий, как, например:

```
ES: hu
```

Следует отметить, что с этими событиями не могут быть связаны какие-либо действия.

### 7.2.2.16 Параметр ResourceID

Параметр ResourceID (идентификатор ресурса) – это параметр завершения, используемый при динамическом качестве обслуживания для информирования об идентификаторе ресурса, присвоенном рассматриваемому вентилю. Параметр ResourceID кодируется в виде строки, содержащей до 8 шестнадцатеричных знаков, как, например:

```
DQ-RI: AB345DC
```

### 7.2.2.17 Параметр RestartMethod

Параметр RestartMethod (метод перезапуска) кодируется в виде одного из ключевых слов "graceful" (постепенный), "cancel-graceful" (постепенный с перезапуском), "forced" (принудительный), "restart" (чистый перезапуск) или "disconnected" (несвязный), как, например:

```
RM: restart
```

### 7.2.2.18 Параметр VersionSupported

Параметр VersionSupported (поддерживаемая версия) кодируется в виде списка поддерживаемых версий, разделенных запятыми, как, например:

```
VS: MGCP 1.0, MGCP 1.0 NCS 1.0
```

### 7.2.2.19 Параметр MaxMGCPDatagram

Параметр MaxMGCPDatagram (максимальная дейтаграмма протокола MGCP) кодируется в виде строки, содержащей до девяти десятичных цифр; начальные нули не допускаются. Следующий пример иллюстрирует использование этого параметра:

MD: 8100

### 7.3 Форматы заголовков ответа

Заголовок ответа состоит из строки ответа, которая необязательно сопровождается заголовками, кодирующими параметры ответа.

Строка ответа начинается с кода ответа, являющегося трехзначным числовым значением. Код сопровождается пробелом, идентификатором транзакции и необязательным комментарием, перед которым стоит пробел, например:

200 1201 OK

В таблице 18, ниже, сведены параметры ответов, наличие которых в заголовке ответа является обязательным или необязательным в зависимости от команды, которая инициировала ответ, в предположении, что команда выполнена успешно. Читателю, однако, следует изучить определения отдельных команд, поскольку данная таблица содержит только краткую информацию. Буква М означает "обязательная", буква О – "необязательная", а буква F – "запрещенная".

**Таблица 18/J.162 – Связь параметров заголовков ответов и команд**

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
ResponseAck	O <sup>1)</sup>							
CallId	F	F	F	F	F	F	O	F
ConnectionId	O <sup>2)</sup>	F	F	F	F	O	F	F
RequestIdentifier	F	F	F	F	F	O	F	F
LocalConnectionOptions	F	F	F	F	F	O	O	F
Connection Mode	F	F	F	F	F	F	O	F
RequestedEvents	F	F	F	F	F	O	F	F
SignalRequests	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	O	O	O
ReasonCode	F	F	F	F	F	O	F	F
ObservedEvents	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	O	F	F
ConnectionParameters	F	F	O <sup>3)</sup>	F	F	F	O	F
SpecificEndpointID	O	F	F	F	F	O	F	F
MaxEndPointIds	F	F	F	F	F	F	F	F
NumEndPoints	F	F	F	F	F	O	F	F
RequestedInfo	F	F	F	F	F	F	F	F
QuarantineHandling	F	F	F	F	F	F	F	F
DetectEvents	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	O	F	F
ResourceID	O	O	F	F	F	F	F	F
RestartMethod	F	F	F	F	F	F	F	F
RestartDelay	F	F	F	F	F	F	F	F
Capabilities	F	F	F	F	F	O	F	F

**Таблица 18/J.162 – Связь параметров заголовков ответов и команд**

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
VersionSupported	F	F	F	F	F	O	F	O
MaxMGCPDatagram	F	F	F	F	F	O	F	F
LocalConnection Descriptor	O <sup>4)</sup>	O <sup>4)</sup>	F	F	F	F	O	F
RemoteConnection Descriptor	F	F	F	F	F	F	O	F

ПРИМЕЧАНИЕ 1. - Параметр ResponseAck НЕ ДОЛЖЕН использоваться в каких-либо других ответах, кроме окончательного ответа, выдаваемого после временного ответа для рассматриваемой транзакции. В этом случае наличие параметра ResponseAck ДОЛЖНО инициировать сообщение подтверждения ответа; все предоставляемые значения параметра ResponseAck будут проигнорированы.

ПРИМЕЧАНИЕ 2. – В случае сообщения CreateConnection, строка ответа сопровождается параметром Connection-Id и LocalConnectionDescriptor. Она также может сопровождаться параметром Specific-Endpoint-Id, если создание запроса было послано с подстановкой Endpoint-Id. Параметры Connection-Id и LocalConnectionDescriptor помечены в таблице, как необязательные. В действительности они обязательны, если сопровождаются положительными ответами, когда соединение создается, и запрещены, если ответ отрицательный и соединение не было создано.

ПРИМЕЧАНИЕ 3. – Параметры соединения (Connection-Parameters) действительны только при успешном ответе на команду DeleteConnection без подстановочных знаков, отправленной агентом вызова.

ПРИМЕЧАНИЕ 4. – LocalConnectionDescriptor ДОЛЖЕН быть передан с положительным ответом (код 200) на команду CreateConnection. Он ДОЛЖЕН быть также передан в ответе на команду ModifyConnection, если модификация приводит к изменению Local Connection Descriptor. LocalConnectionDescriptor кодируется как "описание сеанса", как определено в п.7.4. От заголовка ответа он отделяется пустой строкой.

Ниже приводятся описания параметров ответов для каждой из команд.

### 7.3.1 Команда CreatConnection

В случае сообщения CreateConnection (создать соединение) строка ответа сопровождается параметром Connection-Id (идентификатор соединения) с успешным ответом (код 200). Кроме того, передается параметр LocalConnectionDescriptor с положительным ответом. Параметр LocalConnectionDescriptor кодируется в виде "описания сеанса", как описано в п. 7.4. Он отделяется от заголовка ответа пустой строкой, например:

```

200 1204 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96 97 0
a=rtpmap:96 G726-32/8000
a=rtpmap:97 telephone-event/8000
a=mptime: 10 - 10

```

Если ранее был выдан временный ответ, окончательный ответ может также содержать параметр подтверждения ответа, а если используется динамическое качество обслуживания, окончательный ответ может также содержать параметр ResourceID (идентификатор ресурса), как, например:

```

200 1204 OK
K:
I: FDE234C8
DQ-RI: 23DB4A43

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0

```

```
m=audio 3456 RTP/AVP 96 97 0
a=rtpmap:96 G726-32/8000
a=rtpmap:97 telephone-event/8000
a=mptime: 10 - 10
```

Окончательный ответ подтверждается параметром подтверждения ответа:

```
000 1204
```

### 7.3.2 Команда ModifyConnection

В случае успешного сообщения ModifyConnection (модификация соединения) строка ответа сопровождается параметром LocalConnectionDescriptor, если модификация привела к изменению параметров сеанса (например, изменение только режима соединения не изменяет параметры сеанса). Параметр LocalConnectionDescriptor кодируется в виде "описания сеанса", как определено в п. 7.4. Он отделяется от заголовка ответа пустой строкой.

```
200 1207 OK

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime: 20
```

Ответ может также содержать параметр ResourceID, когда используется динамическое качество обслуживания, как, например:

```
200 1207 OK
DQ-RI: 12345
```

Когда ранее был выдан временный ответ, окончательный ответ может также содержать параметр подтверждения ответа, как, например:

```
526 1207 Нет ширины полосы пропускания
К:
```

Окончательный ответ подтверждается параметром подтверждения ответа:

```
000 1207 OK
```

### 7.3.3 Команда DeleteConnection

В зависимости от варианта сообщения DeleteConnection (исключить соединение) строка ответа может сопровождаться строкой параметров соединения, как определено в п. 7.2.2.5.

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48,
  PC/RPS=782, PC/ROS=45238, PC/RPL=5, PC/RJI=26
```

### 7.3.4 Команда NotificationRequest

Ответ на команду NotificationRequest (запрос на уведомление) не содержит каких-либо дополнительных параметров ответа.

### 7.3.5 Команда Notify

Ответ на команду Notify (уведомить) не содержит никаких дополнительных параметров ответа.

### 7.3.6 Команда AuditEndpoint

В случае команды AuditEndPoint (проверка конечной точки) строка ответа может сопровождаться информацией для каждого из запрашиваемых параметров; каждый параметр будет содержаться в отдельной строке. Параметры, для которых в настоящее время значения не существует, например отображение цифр набора номера, все равно будут предоставлены. Каждое локальное имя конечной точки, "расширяемое" подстановочным знаком, будет находиться на отдельной строке при

использовании кода параметра "SpecificEndPointId" (идентификатор конкретной конечной точки), например:

```
200 1200 OK
Z: aaln/1@rgw.whatever.net
Z: aaln/2@rgw.whatever.net
```

Ниже приводится пример ответа на сообщение AuditEndPoint, содержащее имя конечной точки без использования подстановочных знаков. Следует отметить, что в этом случае параметр SpecificEndPointId не предоставляется. Также необходимо отметить, что каждый набор возможностей приводится одной строкой. В приведенном ниже примере каждый набор возможностей представлен на нескольких строках только из-за ограничений форматирования, налагаемых настоящей Рекомендацией.

```
200 1200 OK
A: a:PCMU, p:10, e:on, s:off, t:1, v:X,
  m:sendonly;recvonly;sendrecv;inactive
A: a:G728, p:20, e:on, s:off, t:1, v:L,
  m:sendonly;recvonly;sendrecv;inactive
A: a:G729, p:30, e:on, s:on, t:1, v:X,
  m:sendonly;recvonly;sendrecv;inactive;confrnce
```

### 7.3.7 Команда AuditConnection

В случае команды AuditConnection (проверка соединения) ответ может сопровождаться информацией для каждого из запрашиваемых параметров. Параметры, для которых в настоящее время значения не существует, все равно будут предоставлены. Дескрипторы соединений будут всегда последними, и каждому из них будет предшествовать пустая строка, как, например:

```
200 1203 OK
C: A3C47F21456789F0
N: CA-1@myhost.whatever.net:2345
L: mp:20;10, a:PCMU;G728
M: sendrecv
P: PS=622, OS=31172, PR=390, OR=22561, PL=5, JI=29, LA=50,
  PC/RPS=391, PC/ROS=22619, PC/RPL=5, PC/RJI=26
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 96
a=rtpmap:96 G728/8000
a=mptime: 10
```

В случае предоставления как локального, так и удаленного дескриптора соединения локальный дескриптор соединения будет указан первым из двух. Если запрошен дескриптор соединения, но он не существует для проверяемого соединения, этот дескриптор соединения будет представлен только в поле версии протокола SDP.

### 7.3.8 Команда RestartInProgress

Ответ на команду RestartInProgress (осуществляемый перезапуск) может содержать имя другого агента вызова для связи, например, когда агент вызова переадресует конечную точку другому агенту вызова, как в случае:

```
521 1204 Redirect
N: CA-1@whatever.net
```

## 7.4 Кодирование описания сеанса

Описание сеанса кодируется в соответствии с протоколом описания сеанса (SDP); однако, как определяется ниже, встроенные клиенты могут использовать некоторые упрощающие допущения относительно описания сеанса. Следует отметить, что в соответствии со стандартом RFC 2327 описания сеансов чувствительны к регистру.

Использование протокола SDP, как описано в параметре "media" (среда передачи), зависит от типа сеанса:

- если параметр среды установлен на "аудио", то описание сеанса относится к аудиослуге;
- если параметр среды установлена на "изображение", то описание сеанса относится к услуге изображения, такой как реле FAX Т.38.

#### 7.4.1 Использование протокола SDP

В шлюзе аналоговой линии необходимо описывать только сеансы, которые используют лишь один тип медиаданных MIME в каждый конкретный момент времени; либо "аудио" (для данных голосового диапазона), либо "изображение" (для вызовов факсимильной связи Т.38). Параметры протокола SDP, существенные для типов медиаданных как "аудио", так и "изображение", определены в п. 7.4.2. Параметры, характерные для "изображения", если они используются в Т.38, определяются в п. 7.4.4. Встроенные клиенты должны поддерживать описания сеансов, которые соответствуют этим правилам и представлены в следующем порядке:

- 1) Профиль протокола SDP, представленный ниже.
- 2) SDP: протокол описания сеанса (стандарт RFC 2327).

CMS следует быть осторожным, если он считает необходимым изменить SDP, полученный от конечной точки. SDP обеспечивает средства для связи возможностей одной конечной точки с другой. Если CMS выбирает модифицирование SDP, он НЕ ДОЛЖЕН изменять SDP так, что это приведет к нарушению правил, определенных в данном пункте.

Предоставленный профиль протокола SDP описывает использование протокола описания сеанса в сигнализации NCS. Общее описание и пояснения к общим отдельным параметрам, так же как и к большинству параметров, используемых только для аудиоданных, можно найти в стандарте RFC 2327. Специальные параметры Т.38 для изображения, могут быть найдены в МСЭ-Т Рек. Т.38. Значения конечных точек NCS, которые необходимо предусмотреть для этих полей (передача), и то, что конечные точки NCS должны делать со значениями, предоставленными или не предоставленными для этих полей (прием), указано ниже. Следует отметить, что профиль SDP, используемый здесь не соответствует модели предложение/ответ, определенной в RFC 3264. Таким образом, если CMS нужно взаимодействовать с другим объектом, использующей модель предложение/ответ, CMS может понадобиться отредактировать SDP, который был получен от конечной точки.

#### 7.4.2 Параметры SDP общие для использования как для аудиослуг, так и для услуг изображения

##### 7.4.2.1 Версия (version) (v=) протокола

v= <version>  
v= 0

**Передача:** ДОЛЖНА обеспечиваться в соответствии со стандартом 2327 (то есть v = 0).

**Прием:** ДОЛЖНА обеспечиваться в соответствии со стандартом RFC 2327.

##### 7.4.2.2 Начало (o=)

Поле "начало" состоит (o=) из 6 подполей по стандарту RFC 2327:

o= <username> <session-ID> <version> <network-type> <address-type> <address>  
o= - 2987933615 2987933615 IN IP4 126.16.64.4

Username (имя пользователя):

**Передача:** В качестве имени пользователя ДОЛЖЕН быть использован дефис, если запрошена секретность.

В другом случае СЛЕДУЕТ использовать дефис<sup>27</sup>.

<sup>27</sup> Поскольку конечные точки NCS не знают, когда запрашивается секретность, им СЛЕДУЕТ всегда использовать дефис.

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Session-ID (идентификатор сеанса):

**Передача:** ДОЛЖЕН соответствовать стандарту RFC 2327 для взаимодействия с клиентами, не являющимися клиентами проекта IPCablecom.

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Version (версия):

**Передача:** В соответствии со стандартом RFC 2327.

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Network Type (тип сети):

**Send:** ДОЛЖЕН быть использован тип "IN"(интеллектуальная сеть).

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Address Type (тип адреса):

**Передача:** ДОЛЖЕН быть использован тип "IP4".

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Address (адрес):

**Передача:** ДОЛЖЕН соответствовать стандарту RFC 2327 для взаимодействия с клиентами, не являющимися клиентами проекта IPCablecom.

**Прием:** Это поле ДОЛЖНО быть проигнорировано.

#### 7.4.2.3 Имя сеанса (s=)

*s= <session-name>*

*s= -*

**Передача:** В качестве имени сеанса ДОЛЖЕН быть использован дефис.

**Прием:** Это поле ДОЛЖНО быть проигнорировано.

#### 7.4.2.4 Информация о сеансе и среде передачи (i=)

*i= <session-description>*

**Передача:** Это поле НЕ ДОЛЖНО использоваться для сигнализации NCS.

**Прием:** Это поле ДОЛЖНО быть проигнорировано.

#### 7.4.2.5 Идентификатор URI (u=)

*u= <URI>*

**Передача:** Это поле НЕ ДОЛЖНО использоваться для сигнализации NCS.

**Прием:** Это поле ДОЛЖНО быть проигнорировано.

#### 7.4.2.6 Адрес электронной почты и номер телефона (e=, p=)

*e= <e-mail-address>*

*p= <phone-number>*

**Передача:** Это поле НЕ ДОЛЖНО использоваться для сигнализации NCS.

**Прием:** Это поле ДОЛЖНО быть проигнорировано.

#### 7.4.2.7 Данные соединения (c=)

Данные соединения состоят из 3 подполей:

```
c= <network-type> <address-type> <connection-address>  
c= IN IP4 10.10.111.11
```

Network Type (тип сети):

**Передача:** ДОЛЖЕН быть использован тип "IN" (интеллектуальная сеть).

**Прием:** ДОЛЖЕН присутствовать тип "IN" (интеллектуальная сеть).

Address Type (тип адреса):

**Передача:** ДОЛЖЕН быть использован тип "IP4".

**Прием:** ДОЛЖЕН присутствовать тип "IP4".

Connection Address (адрес соединения):

**Передача:** Это поле ДОЛЖНО быть заполнено IP-адресом для одноадресной передачи, на который прикладная программа будет получать медиапоток. Таким образом, НЕ ДОЛЖНЫ присутствовать значения TTL и значение "числа адресов". Это поле НЕ ДОЛЖНО быть заполнено полностью определенным доменным именем вместо IP-адреса. Ненулевой адрес определяет **как адрес приема, так и адрес передачи для медиапотока (потоков), для которого он предназначен.**

**Прием:** ДОЛЖНЫ присутствовать IP-адрес для одноадресной передачи или полностью определенное доменное имя. Ненулевой адрес определяет как адрес приема, так и адрес передачи для медиапотока (потоков), для которого он предназначен.

#### 7.4.2.8 Ширина полосы пропускания (b=)

```
b= <modifier> : <bandwidth-value>  
b= AS : 64
```

**Передача:** Информация о ширине полосы пропускания является необязательной в протоколе SDP, но ее всегда СЛЕДУЕТ включать<sup>28</sup>. Когда используется отображение rtpmap или не очень известный кодек<sup>29</sup>, информация о ширине полосы пропускания ДОЛЖНА быть использована.

**Прием:** СЛЕДУЕТ включить информацию о ширине полосы пропускания. Если модификатор ширины полосы пропускания не включен, то получатель ДОЛЖЕН принимать допустимые по умолчанию значения ширины полосы пропускания для известных кодеков.

Modifier (модификатор):

**Передача:** ДОЛЖЕН быть использован тип "AS".

**Прием:** ДОЛЖЕН присутствовать тип "AS".

Bandwidth Value (значение ширины полосы пропускания):

**Передача:** Поле ДОЛЖНО быть заполнено значением требуемой максимальной ширины полосы пропускания медиапотока в килобитах в секунду.

**Прием:** ДОЛЖНА присутствовать требуемая максимальная ширина полосы пропускания медиапотока в килобитах в секунду. Подробная информация о вычислении значения ширины полосы пропускания содержится в Рекомендации МСЭ-Т J.161.

<sup>28</sup> Если это поле не используется, то контроллер вентиля может не дать разрешение на использование соответствующей ширины полосы пропускания.

<sup>29</sup> Не очень известный кодек – это кодек, не описанный в Рекомендации МСЭ-Т J.161.

#### 7.4.2.9 Время, количество повторений и временные зоны (t=, r=, z=)

t= <start-time> <stop-time>

t= 36124033 0

r= <repeat-interval> <active-duration> <list-of-offsets-from-start-time>

z= <adjustment-time> <offset>

**Передача:** Время ДОЛЖНО присутствовать; время запуска МОЖЕТ быть нулем, но в качестве такого времени СЛЕДУЕТ использовать текущее время, а в качестве времени остановки СЛЕДУЕТ использовать нуль. НЕ СЛЕДУЕТ использовать количество повторений и временные зоны; если они используются, это должно соответствовать стандарту RFC 2327.

**Прием:** Если присутствует любое из этих полей, их СЛЕДУЕТ проигнорировать.

#### 7.4.2.10 Атрибуты (a=)

a= <attribute> : <value>

a= mptime: <alternative 1> <alternative 2 > ...

a= rcvonly

a= sendrcv

a= sendonly

a= ptime

**Передача:** МОЖЕТ включаться одна или несколько строк атрибутов "a", указанных ниже.

**Прием:** МОЖЕТ включаться одна или несколько строк атрибутов "a", указанных ниже, использоваться они ДОЛЖНЫ соответственно назначению.

Следует отметить, что SDP требует игнорирования неизвестных атрибутов.

mptime:

Этот атрибут является атрибутом уровня среды передачи, определенного проектом IP-Cablecom. Атрибут mptime определяет список значений периода пакетирования, которые может использовать (передавать или принимать) конечная точка для данного соединения.

**Передача:** Атрибут mptime ДОЛЖЕН присутствовать. В списке для каждого элемента <format> (<формат>), содержащегося в строке "m=", ДОЛЖЕН быть один элемент. Номер j элемента в этом списке определяет период пакетирования для элемента с номером j в строке "m=". Первый элемент в этом списке ДОЛЖЕН быть десятичным числом либо дефисом. Для тех форматов среды передачи, где не применяется одна скорость пакетирования (например, в речевых кодеках, связанных с телефонными событиями или комфортным шумом), в соответствующем месте в списке периодов пакетирования ДОЛЖЕН кодироваться дефис ("-").

**Прием:** Передается список периодов пакетирования, которые может использовать удаленная конечная точка для данного соединения, по одному периоду пакетирования для каждого формата медиаданных в строке "m=". Для форматов медиаданных, период пакетирования которых определяется в виде дефиса ("-"), конечная точка ДОЛЖНА использовать один из периодов пакетирования, который был фактически задан в списке. Если атрибут "mptime" отсутствует, ДОЛЖНО быть взято значение атрибута "ptime", если таковой имеется, как указывающее период пакетирования для всех кодеков, присутствующих в строке "m=".

rcvonly:

**Передача:** Данный атрибут встроенному клиенту предоставлять не следует.

**Прием:** Это поле ДОЛЖНО игнорироваться.

sendrcv:

**Передача:** Данный атрибут встроенному клиенту предоставлять не следует.

**Прием:** Это поле ДОЛЖНО игнорироваться.

sendonly:

**Передача:** Данный атрибут встроенному клиенту предоставлять не следует.

**Прием:** Это поле ДОЛЖНО игнорироваться.

ptime:

**Передача:** Атрибут ptime СЛЕДУЕТ передать, если он был получен в удаленном дескрипторе соединения или если в сигнализации CMS использовался период пакетирования ('p:') LocalConnectionOption.

**Прием:** Это поле ДОЛЖНО быть проигнорировано, если протокол SDP содержит атрибут "ptime" (как требуется в устройствах, соответствующих проекту IP-Cablecom). Если атрибут "ptime" отсутствует, тогда это поле используется для определения интервала пакетирования для всех кодеков, содержащихся в описании протокола SDP, и адаптер МТА ДОЛЖЕН использовать атрибут ptime при вычислении резервирования ресурсов для QoS.

### 7.4.3 Использование аудиоуслуг SDP

Следующие параметры SDP применяются на уровне среды передачи и являются характерными для использования аудиоуслуг. Конечные точки PacketCable НЕ ДОЛЖНЫ передавать какой-либо из этих параметров через дескриптор медиаданных изображения (см. 7.4.4). Однако, если конечная точка получает SDP с параметрами атрибутов, характерными только для изображения с дескриптором аудио медиаданных, параметры СЛЕДУЕТ игнорировать. Кроме того, если должны быть предоставлены параметры возможностей медиаданных, каждый дескриптор возможностей медиаданных (который включает в себя строку описания возможностей, a=cdsc, с 0 или более строк атрибутов возможностей, например, a=cpag) ДОЛЖЕН появляться после последнего атрибута, и каждый дескриптор возможностей медиаданных ДОЛЖЕН перечисляться отдельно.

#### 7.4.3.1 Ключи шифрования

*k= <method>*

*k= <method> : <encryption-keys>*

Услуги обеспечения безопасности для проекта IP-Cablecom определены в Рекомендации МСЭ-Т J.170. Услуги обеспечения безопасности, определенные для протоколов RTP и RTCP, не соответствуют аналогичным услугам, описанным в стандартах RFC 3550, RFC 3551 и RFC 2327. В целях взаимодействия с устройствами, не входящими в проект IP-Cablecom, параметр "k", таким образом, не будет использоваться для передачи параметров обеспечения безопасности.

**Передача:** НЕ ДОЛЖНЫ использоваться.

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

#### 7.4.3.2 Атрибуты (a=)

*a= <attribute> : <value>*

*a= rtpmap : <payload type> <encoding name>/<clock rate> [/<encoding parameters>]*

*a= rtpmap : 0 PCMU / 8000*

*a= fmp: <format><format specific parameters>*

*a= X-pc-codecs: <alternative 1> <alternative 2> ...*

*a= X-pc-secret: <method>:<encryption key>[pad]*

*a= X-pc-csuites-rtp: <alternative 1> <alternative 2> ...*

*a= X-pc-csuites-rtcp: <alternative 1> <alternative 2> ...*

*a= X-pc-nrekey: <value>= <attribute>*

**Передача:** МОГУТ быть включены одна или несколько строк атрибутов "a", описанных ниже.

**Прием:** МОГУТ быть включены одна или несколько строк атрибутов "a", описанных ниже, и они ДОЛЖНЫ быть обработаны соответствующим образом.

Следует отметить, что SDP требует игнорирования неизвестных атрибутов.

rtpmap:

**Передача:** Это поле при использовании ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Это поле МОЖЕТ быть использовано для неизвестных,

равно как и для известных кодеков. Используемые имена кодирования содержатся в отдельной Рекомендации по проекту IP-Cablecom. Соответствие кодека типу динамической полезной нагрузки RTP, заданному этим атрибутом, определяет тип полезной нагрузки, которую данный отправитель готов принять по соединению. Оно также служит четким указанием другой стороне, что ей также следует использовать это соответствие полезной нагрузки для ее принимающей стороны, хотя могут быть случаи, когда это невозможно. Если адаптер МТА установил соответствие динамического типа полезной нагрузки данному методу кодирования для принимаемого им медиапотока, то в дальнейшем в данном соединении этот тип полезной нагрузки НЕ ДОЛЖЕН соответствовать другому методу кодирования для принимаемого им медиапотока.

**Прием:** Это поле при использовании ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Этот атрибут определяет соответствие кодека типу полезной нагрузки RTP, которую готовы принять на другом конце соединения. Поэтому адаптеры МТА ДОЛЖНЫ использовать это соответствие типу полезной нагрузки при передаче медиапакетов по данному соединению. Когда эта информация получена в команде CreateConnection, адаптеру МТА СЛЕДУЕТ использовать это соответствие типу полезной нагрузки для его собственной принимающей стороны (то есть он должен вернуть локальный дескриптор соединения, содержащий тот же атрибут rtpmap). Если адаптер МТА принимает атрибут rtpmap в команде ModifyConnection с другим соответствием, данный адаптер МТА ДОЛЖЕН оставить соответствие своему собственному типу полезной нагрузки прежним (так, чтобы использовались типы асимметричной полезной нагрузки).

fmtp:

**Передача:** Это поле МОЖЕТ быть использовано для обеспечения параметров, характерных для определенного формата. Например, это поле могло быть использовано для описания телефонных событий, поддерживаемых для формата стандарта RFC 2833. В случае использования этот формат ДОЛЖЕН быть одним из форматов, заданных для медиаданных. Определенные параметры содержатся в отдельной Рекомендации, в которой подробно описывается использование формата.

**Прием:** В случае использования данное поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2327.

X-rc-codect:

Этот атрибут является атрибутом уровня среды передачи, который определяется в проекте IP-Cablecom.

**Передача:** Это поле содержит список альтернативных кодеков, которые может использовать конечная точка для данного соединения. Список построен в порядке уменьшения уровня приоритета, то есть наиболее предпочтительным альтернативным кодеком является первый кодек в списке. Кодек кодируется аналогично полю "encoding name" (имя кодирования) в атрибуте rtpmap.

**Прием:** Передается список кодеков, которые может использовать удаленная конечная точка для данного соединения. Кодеки НЕ ДОЛЖНЫ быть использованы до тех пор, пока о них не будет сообщено посредством строки медиаданных (m=).

X-rc-secret:

Этот атрибут является атрибутом уровня среды передачи, который определяется в проекте IP-Cablecom.

**Передача:** Это поле содержит закрытую информацию для сквозной передачи и (возможно) заполнение PAD, которые должны использоваться для обеспечения безопасности протоколов RTP и RTCP. Закрытая информация и PAD кодируются аналогично параметру "encryption key" (ключ шифрования) (k=) в стандарте RFC 2327 со следующими ограничениями:

- Ключ шифрования НЕ ДОЛЖЕН содержать шифрокомплект, а должен содержать только фразу-пароль.

- Атрибут `<method>` (`<метод>`), задающий кодирование фразы-пароля, ДОЛЖЕН быть либо "clear" ("освободить"), либо "base64" ("база64"), как определено в стандарте RFC 2045, за исключением максимальной длины строки, которая здесь не задана. Метод "clear" НЕ ДОЛЖЕН быть использован, если закрытая информация или заполнение PAD содержат какие-либо символы, запрещенные в протоколе SDP.

Требования относительно того, когда передавать заполнение, описаны в Рекомендации МСЭ-Т J.170. Если заполнение PAD присутствует, оно ДОЛЖНО быть отделено от закрытой информации по крайней мере одним пробелом. Для заполнения PAD и закрытой информации используется один и тот же метод кодирования.

**Прием:** Передает закрытую информацию для сквозной передачи и заполнение PAD, которые должны использоваться для обеспечения безопасности протоколов RTP и RTCP. Если заполнение PAD присутствует, оно используется согласно Рекомендации МСЭ-Т J.170 и ДОЛЖНО быть отделено от закрытой информации по крайней мере одним пробелом. Для заполнения PAD и закрытой информации используется один и тот же метод кодирования.

X-rc-suites-rtp:

X-rc-suites-rtcp:

Эти атрибуты являются атрибутами уровня среды передачи, которые определяются в проекте IPCom.

**Передача:** Это поле содержит список шифрокомплектов, которые может использовать конечная точка для заданного соединения (соответственно, для протоколов RTP и RTCP). Первый шифрокомплект в списке – это тот, который намерена использовать конечная точка в текущий момент. Все остальные шифрокомплекты списка представляют альтернативы, упорядоченные по степени снижения уровня приоритета, то есть наиболее предпочтительным альтернативным шифрокомплексом является второй шифрокомплект списка. Шифрокомплект кодируется так, как определено ниже:

```
ciphersuite = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]
AuthenticationAlgorithm = 1*( ALPHA / DIGIT / "-" / "_" )
EncryptionAlgorithm = 1*( ALPHA / DIGIT | "-" / "_" )
```

где ALPHA и DIGIT определены в стандарте RFC 2234. В пределах шифрокомплекта пробелы не допускаются. Следующий пример иллюстрирует использование шифрокомплекта:

62/51

Действительный список шифрокомплектов содержится в Рекомендации МСЭ-Т J.170.

**Прием:** Передает список шифрокомплектов, которые может использовать удаленная конечная точка для данного соединения. Любой другой шифрокомплект, кроме первого в списке, не может быть использован до тех пор, пока о нем не будет сообщено посредством новой строки шифрокомплекта, где желательный для использования шифрокомплект будет перечислен первым.

X-rc-nrekey:

Этот атрибут является атрибутом уровня среды передачи, который определяется в проекте IPCom.

**Передача:** Это поле содержит 16-битовый целочисленный счетчик числа событий повторного ввода с клавиатуры. Это поле может потребоваться, когда используется обеспечение безопасности передачи голоса. Требования по использованию этого поля определены в Рекомендации МСЭ-Т J.170.

**Прием:** Передается число событий повторного ввода с клавиатуры. Это поле может присутствовать, когда используется обеспечение безопасности протокола RTP, и использование этого поля определено в Рекомендации МСЭ-Т J.170.

### 7.4.3.3 Уведомления о медиаданных (m=)

Уведомления о медиаданных (m=) состоят из 4 подполей:

*m= <media> <port> <transport> <fmt list>*  
*m= audio 3456 RTP/AVP 0 97*

Media (среда передачи):

- Передача:** ДОЛЖЕН использоваться тип медиаданных "аудио".  
**Прием:** Полученным типом медиаданных ДОЛЖЕН быть "аудио".

Port (порт):

- Передача:** Это поле ДОЛЖНО быть заполнено в соответствии со стандартом RFC 2327. Описанный порт является портом приема независимо от того, является ли поток однонаправленным или двунаправленным. Передающий порт может быть другим.  
**Прием:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Описанный порт является портом приема. Передающий порт может быть другим.

Transport (транспортировка):

- Передача:** ДОЛЖЕН быть использован транспортный протокол "RTP/AVP".  
**Прием:** Транспортным протоколом ДОЛЖЕН быть протокол "RTP/AVP".

Форматы медиаданных:

- Передача:** ДОЛЖЕН быть использован соответствующий тип медиаданных, как определено в стандарте RFC 2327. В частности, это поле содержит список из одного или нескольких типов полезной нагрузки протокола RTP, которые готов принять по соединению данный адаптер МТА и которые он предпочел бы передать. Каждый тип полезной нагрузки однозначно соответствует кодеку статически или динамически. СЛЕДУЕТ использовать статическое соответствие, если таковое имеется (например, 0 для РСМУ, 8 для РСМА). Если используется динамическое соответствие кодека полезной нагрузке, то ДОЛЖЕН также присутствовать атрибут RTPMAP, и ДОЛЖНЫ выполняться указания, содержащиеся в п. 7.4.1.11.  
**Прием:** В соответствии со стандартом RFC 2327. В частности, форматы медиаданных указывают на тип (типы) полезной нагрузки, который готовы принять на другом конце данного соединения.

#### 7.4.4 Использование услуги изображения по протоколу SDP для Т.38

Следующие параметры SDP применяются для уровня среды передачи и являются характерными для использования услуг передачи изображения. Конечные точки PacketCable НЕ ДОЛЖНЫ передавать какой-либо из этих параметров через дескриптор медиаданных аудио (см. 7.4.3). Однако, если конечная точка получает SDP с параметрами атрибутов, характерными только для аудио с дескриптором медиаданных изображения, параметры СЛЕДУЕТ игнорировать. Кроме того, если должны быть предоставлены параметры возможностей медиаданных, каждый дескриптор возможностей медиаданных (который включает в себя строку описания возможностей, a=cdsc, с 0 или более строк атрибутов возможностей, например, a=srag) ДОЛЖЕН появляться после последнего атрибута, и каждый дескриптор возможностей медиаданных ДОЛЖЕН перечисляться отдельно.

##### 7.4.4.1 Ключи шифрования

*k= <method>*  
*k= <method> : <encryption-keys>*

В настоящее время нет услуг обеспечения безопасности для типа медиаданных "изображение/t38".

- Передача:** НЕ ДОЛЖНЫ использоваться.  
**Прием:** Это поле СЛЕДУЕТ проигнорировать.

#### 7.4.4.2 Атрибуты (a=)

a= <attribute> : <value>  
a=T38FaxVersion: <version>  
a=T38MaxBitrate: <bitrate>  
a=T38FaxRateManagement: <faxratemanagement>  
a=T38FaxMaxBuffer: <maxbuffer>  
a=T38FaxMaxDatagram: <maxsize>  
a=T38FaxUdpEC: <ECmethod>  
a=T38FaxFillBitRemoval  
a=T38FaxTranscodingMMR  
a=T38FaxTranscodingJBIG

**Передача:** МОГУТ быть включены одна или несколько строк атрибутов "a", описанных ниже.

**Прием:** МОГУТ быть включены одна или несколько строк атрибутов "a", описанных ниже, и они ДОЛЖНЫ быть обработаны соответствующим образом. Значения атрибутов не чувствительны к регистру. Реализации ДОЛЖНЫ принимать кодирование всех атрибутов в нижнем регистре, верхнем регистре и в смешанном (нижнем и верхнем регистре).

Следует отметить, что SDP требует игнорирования неизвестных атрибутов.

##### T38FaxVersion:

Как определено в МСЭ-Т Рек. Т.38: Получатель предложения ДОЛЖЕН принять данную версию или модифицировать атрибут версии так, чтобы он был меньше или равен версии, которая передает ответ на иницилирующее предложение. Получатель предложения НЕ ДОЛЖЕН отвечать сообщением, содержащим более высокую версию, чем та, что была предложена.

Также, как определено в МСЭ-Т Рек. Т.38: Ранние реализации оборудования Т.38 могут не предоставлять номера версии Т.38. При получении SDP без атрибута версии конечная точка ДОЛЖНА принять номер версии за 0. Это применяется в последующем описании особенностей передачи и приема данного атрибута.

**Передача:** Конечная точка ДОЛЖНА указывать версию, которую она собирается использовать через атрибут T38FaxVersion. Однако она НЕ ДОЛЖНА указывать номер версии выше, чем номер версии, полученный в RemoteConnectionDescriptor.

**Прием:** Если получен RemoteConnectionDescriptor, и атрибут T38FaxVersion в него не включен, то конечная точка ДОЛЖНА использовать версию 0 спецификации Т.38. Если атрибут включен, конечная точка ДОЛЖНА использовать версию спецификации меньше или такую же, как и указанная версия.

##### T38MaxBitRate:

**Передача:** Атрибут T38MaxBitRate включаться НЕ ДОЛЖЕН.

**Прием:** Атрибут T38MaxBitRate СЛЕДУЕТ игнорировать.

##### T38FaxRateManagement:

**Передача:** Атрибут T38FaxRateManagement ДОЛЖЕН включаться и ДОЛЖЕН иметь значение "transferredTCF", если используется UDPTL. Со значением "transferredTCF", TCF передается от одной конечной точки до другой, в отличие от значения атрибута "localTCF", где TCF генерируется локально. Следует отметить, что "localTCF" подходит только для случаев, когда используется надежное средство передачи, такое как TCP.

**Прием:** Когда используется UDPTL, атрибут T38FaxRateManagement ДОЛЖЕН либо присутствовать со значением "transferredTCF", либо отсутствовать, и в этом случае принимается переданный TCF. Все другие значения атрибута ДОЛЖНЫ быть отклонены (код ошибки 505 – Неподдерживаемый RemoteConnectionDescriptor).

#### T38FaxMaxBuffer:

**Передача:** Атрибут T38FaxMaxBuffer включаться НЕ ДОЛЖЕН.

**Прием:** Атрибут T38FaxMaxBuffer СЛЕДУЕТ игнорировать.

#### T38FaxMaxDatagram:

**Передача:** Атрибут T38FaxMaxDatagram ДОЛЖЕН включаться. Показываемое значение НЕ ДОЛЖНО быть меньше, чем 160 байт. Оно базируется на периоде пакетирования 40 мс и скорости передачи данных 14 400 бит/с. Включает в себя дейтаграмму UDPTL без заголовков IP и UDP.

**Прием:** Конечные точки НЕ ДОЛЖНЫ посылать дейтаграмму большую, чем указано в атрибуте T38FaxMaxDatagram. До того, как отправить какую-либо дейтаграмму T.38, конечная точка ДОЛЖНА удостовериться, что она не выходит за границы, определенных для данного атрибута. Если указанное значение T38FaxMaxDatagram слишком мало для поддержки избыточности для данной дейтаграммы, но достаточно для поддержки дейтаграммы T.38 без избыточности, то конечная точка ДОЛЖНА послать эту дейтаграмму без избыточности. Если значение слишком мало, чтобы послать дейтаграмму без избыточности, конечная точка НЕ ДОЛЖНА посылать дейтаграмму T.38, а ДОЛЖНА сгенерировать индикатор отказа.

#### T38FaxUdpEC:

Поддержка избыточности обязательна, в то время как поддержка для упреждающей коррекции ошибок необязательна. Использование одной из этих схем требует согласования.

**Передача:** Атрибут T38FaxUdpEC ДОЛЖЕН включаться. Значение "t38UDPFEC" МОЖЕТ посылаться, если поддерживается FEC, и либо с командой не предоставлен RCD, либо значение атрибута, полученного в RCD для этой команды "t38UDPFEC". В другом случае ДОЛЖНО посылаться "t38UDPRedundancy".

**Прием:** Избыточность ДОЛЖНА использоваться, если значением атрибута T38FaxUdpEC является "t38UDPRedundancy". Если значением атрибута T38FaxUdpEC является "t38UDPFEC", и конечной точкой поддерживается FEC, то СЛЕДУЕТ использовать FEC. Если значением атрибута T38FaxUdpEC является "t38UDPFEC", и FEC не поддерживается, то ДОЛЖНА использоваться избыточность. Если данный атрибут не включен, конечная точка НЕ ДОЛЖНА использовать избыточность или FEC.

#### T38FaxFillBitRemoval:

Поддержка удаления битов дополнения необязательна, и любое ее использование должно быть согласовано.

**Передача:** Если поддерживается вставка и удаление битов дополнения и необходимо их использовать, и команда либо не включает RCD, либо включает в себя RCD при наличии атрибута T38FaxFillBitRemoval, то T38FaxFillBitRemoval ДОЛЖЕН включаться, и ДОЛЖНЫ использоваться вставка и удаление битов дополнения. Во всех других случаях, атрибут T38FaxFillBitRemoval НЕ ДОЛЖЕН включаться, а вставка и удаление битов дополнения использоваться НЕ ДОЛЖНЫ.

**Прием:** Вставка и удаление битов дополнения НЕ ДОЛЖНЫ использоваться, если атрибут T38FaxFillBitRemoval отсутствует.

#### T38FaxTranscodingMMR:

MMR перекодирование не применяется к T.38, базирующемуся на UDPTL.

**Передача:** При использовании UDPTL для T.38, атрибут T38FaxTranscodingMMR включаться НЕ ДОЛЖЕН.

**Прием:** Если атрибут T38FaxTranscodingMMR предоставлен для T.38, базирующемся на UDPTL, команда ДОЛЖНА быть отклонена (код ошибки 505 – неподдерживаемый RemoteConnectionDescriptor).

T38FaxTranscodingJBIG:

JBIG перекодирование не применяется к T.38, базирующемся на UDPTL.

**Передача:** При использовании UDPTL для T.38, атрибут T38FaxTranscodingJBIG включаться НЕ ДОЛЖЕН.

**Прием:** Если атрибут T38FaxTranscodingJBIG предоставлен для T.38, базирующемся на UDPTL, команда ДОЛЖНА быть отклонена (код ошибки 505 – неподдерживаемый RemoteConnectionDescriptor).

#### 7.4.4.3 Уведомления о медиаданных (m=)

Уведомления о медиаданных (m=) состоят из 4 подполей:

```
m= <media> <port> <transport> <fmt list>
"m= image 3456 udpt1 t38"
```

Media (медиаданные):

**Передача:** ДОЛЖЕН использоваться тип медиаданных "изображение" для T.38, базирующегося на UDPTL.

**Прием:** Полученным типом медиаданных ДОЛЖНО быть "изображение" для T.38, базирующегося на UDPTL.

Port (порт):

**Передача:** Это поле ДОЛЖНО быть заполнено в соответствии со стандартом RFC 2327. Описанный порт является портом приема. Передающий порт может быть другим.

**Прием:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Описанный порт является портом приема. Передающий порт может быть другим.

Transport (транспортирование):

**Передача:** ДОЛЖЕН быть использован транспортный протокол "udpt1" для T.38, базирующегося на UDPTL.

**Прием:** Транспортным протоколом ДОЛЖЕН быть протокол "udpt1" для T.38, базирующегося на UDPTL. Реализациям СЛЕДУЕТ принимать "UDPTL" в форме как верхнего регистра, так и в форме строки "udpt1" смешанного (верхнего и нижнего) регистра.

Форматы медиаданных:

**Передача:** ДОЛЖЕН быть тип медиаданных „t38“.

**Прием:** ДОЛЖЕН быть тип медиаданных „t38“.

## 7.5 Передача по протоколу UDP

### 7.5.1 Надежная доставка сообщений

Сообщения MGCP доставляются по протоколу UDP. Команды посылаются по одному из IP-адресов, определяемых в системе доменных адресов для заданной конечной точки или агента вызова. Ответы посылаются в обратном направлении по адресу источника команды. Однако следует отметить, что ответ в действительности может исходить от IP-адреса, отличного от того, по которому была послана команда.

Когда для конечной точки не был обеспечен порт<sup>30</sup>, команды ДОЛЖНЫ посылаться к порту MGCP по умолчанию; портами по умолчанию являются порт 2427 для команд, посылаемых к шлюзам, и порт 2727 для команд, посылаемых к агенту вызова. Чтобы свести к минимуму проблемы совместимости

---

<sup>30</sup> Каждая конечная точка может быть обеспечена адресом и портом отдельного агента вызова.

сверху вниз, РЕКОМЕНДУЕТСЯ, чтобы агент вызова всегда явным образом указывал порт MGCP для использования в сообщениях сигнализации NCS (а не полагался на определение по умолчанию).

Сообщения MGCP, передаваемые по протоколу UDP, могут быть подвержены потерям. В отсутствие своевременного ответа команды повторяются. Ожидается, что объекты MGCP будут хранить в памяти список ответов, посланных на последние транзакции, то есть список всех ответов, посланных за последние  $T_{hist}$  секунд, а также список транзакций, выполняемых в текущее время. Идентификаторы транзакций входящих команд сравниваются с идентификаторами транзакций последних ответов. Если имеет место совпадение, то объект MGCP не выполняет транзакцию, а просто повторяет ответ. Если совпадения нет, то объект MGCP проверяет список транзакций, выполняемых в текущее время. Если имеет место совпадение, то объект MGCP не будет выполнять транзакцию. Если командой является команда CreateConnection или ModifyConnection, то посылается временный ответ; в противном случае команда просто игнорируется.

В обязанности запрашивающего объекта входит предоставление необходимых выдержек времени для всех ожидающих обработки команд и повторение команд, когда были превышены выдержки времени. Стратегия выполнения повторных передач команд описана в п. 7.5.2.

Кроме того, когда на повторные команды не приходит ответ, предполагается, что объект места назначения команды недоступен. Запрашивающий объект, как описано в п. 6.4, отвечает за поиск резервных услуг и/или освобождение существующих или ждущих обработки соединений.

### 7.5.2 Стратегия повторной передачи

Настоящая Рекомендация избегает определения каких-либо статистических значений для таймеров повторной передачи, поскольку эти значения, как правило, зависят от сети. Обычно таймеры повторной передачи должны оценивать время таймера путем измерения времени между передачей команды и возвратом ответа. Встроенные клиенты ДОЛЖНЫ реализовывать стратегии повторной передачи, используя экспоненциальные задержки с конфигурируемыми начальными и максимальными значениями таймеров повторной передачи.

Встроенным клиентам СЛЕДУЕТ использовать алгоритм, реализуемый в протоколах TCP-IP, в котором используются две переменные:

- средняя задержка на подтверждение (AAD), оцениваемая в виде экспоненциально сглаженного среднего наблюдаемых задержек;
- среднее отклонение (ADEV), оцениваемое в виде экспоненциально сглаженного среднего от абсолютного значения разности между наблюдаемой задержкой и текущим средним.

Таймер повторной передачи (RTO) в протоколе TCP устанавливается на сумму средней задержки плюс N-кратное среднее отклонение, где N – константа.

После любой повторной передачи объект MGCP должен выполнить следующее:

- Ему следует удвоить оцениваемое значение средней задержки AAD.
- Ему следует вычислить случайное значение, равномерно распределенное между 0,5 AAD и AAD.
- Ему следует установить таймер повторной передачи (RTO) на минимальное значение:
  - суммы этого случайного значения и N-кратного среднего отклонения.
- $RTO_{max}$ , где значение по умолчанию  $RTO_{max}$  составляет 4 секунды.

Эта процедура имеет два последствия:

- Поскольку она содержит экспоненциально возрастающий компонент, она будет автоматически замедлять поток сообщений в случае перегрузки с учетом потребностей связи в реальном масштабе времени.
- Поскольку процедура содержит случайный компонент, она будет нарушать возможную синхронизацию между уведомлениями, инициируемыми одним и тем же внешним событием.

Начальное значение, используемое для таймера повторной передачи, составляет по умолчанию 200 миллисекунд, а максимальное значение для таймера повторной передачи составляет по умолчанию 4 секунды. Эти значения по умолчанию могут быть изменены за счет процесса подготовки соединения.

### 7.5.3 Максимальный размер дейтаграммы, фрагментация и сборка

В сообщениях MGCP, передаваемых по протоколу UDP, для фрагментации и сборки больших дейтаграмм используется IP-протокол. Максимальный теоретический размер IP-дейтаграммы составляет 65 535 байтов. Если вычесть IP-заголовок в 20 байтов и 8-байтовый заголовок, максимальный теоретический размер сообщения MGCP становится равным 65 507 байтов при использовании протокола UDP.

Однако IP-протокол не требует, чтобы хост принимал IP-дейтаграммы размером более 576 байтов (стандарт RFC 1122), что давало бы неприемлемо малый размер сообщения протокола MGCP. Следовательно, протокол MGCP обязывает реализации поддерживать дейтаграммы MGCP по меньшей мере до 4000 байтов, что требует поддержки соответствующей фрагментации и сборки по IP-протоколу. Следует отметить, что предел в 4000 байтов применяется к уровню протокола MGCP. Служебная нагрузка нижнего уровня потребует поддержки IP-дейтаграмм, которые больше данной: служебная нагрузка для протоколов UDP и IP будет составлять, по меньшей мере, 28 байтов, а, например, для протокола IPsec служебная нагрузка составит еще больше.

Следует отметить, что вышесказанное относится как к агентам вызова, так и к конечным точкам. Агенты вызова могут контролировать конечные точки для определения того, поддерживают ли они дейтаграммы MGCP большего размера, чем задано выше. В настоящее время конечные точки не обладают аналогичной возможностью для определения того, поддерживает ли агент вызова дейтаграммы MGCP больших размеров.

### 7.6 Совмещение передачи запросов и ответов

Бывают случаи, когда агент вызова желает передать в одно и то же время несколько сообщений одной или нескольким конечным точкам в шлюзе, и наоборот. Когда необходимо послать несколько сообщений в одних и тех же пакетах UDP, они разделяются строкой текста, содержащей одну точку, как в следующем примере:

```
200 2005 OK
```

```
DLCX 1244 aaln/2@rgw.whatever.net MGCP 1.0 NCS 1.0  
C: A3C47F21456789F0  
I: FDE234C8
```

Совмещенные сообщения ДОЛЖНЫ обрабатываться так, как будто они были получены по одному в нескольких отдельных дейтаграммах. Каждое сообщение в дейтаграмме должно быть обработано до конца в порядке, начинающемся с первого сообщения, и на каждую команду ДОЛЖЕН быть получен ответ.

Ошибки, встретившиеся в сообщении, которое было совмещено, НЕ ДОЛЖНЫ влиять на какие-либо другие сообщения, принятые в этом пакете; каждое сообщение обрабатывается индивидуально.

Совмещение передачи запросов и ответов может быть использовано для достижения следующих целей:

- для гарантированной упорядоченной доставки и обработки сообщений;
- для совместного использования конечного результата доставки сообщений.

Когда совмещение передачи используется для гарантирования упорядоченной доставки сообщений, объекты ДОЛЖНЫ обеспечить сохранение такой упорядоченной доставки сообщений при повторных передачах отдельных сообщений. Примером этого служит передача нескольких команд Notify с использованием совмещения передачи запросов и ответов (как описано в п. 6.4.3.1).

Совместное использование конечного результата доставки сообщений обеспечивает либо доставку всех сообщений, либо ни одного из них. Когда совмещение передачи запросов и ответов используется для гарантирования этого совместного использования конечного результата, объекты ДОЛЖНЫ также гарантировать сохранение этого свойства при повторной передаче. Например, по получении команды Notify от конечной точки, работающей в режиме "шаг блокировки", агент вызова может изъявить желание послать ответ и новую команду NotificationRequest в одной дейтаграмме, чтобы обеспечить совместное использование конечного результата доставки этих двух сообщений.

## 7.7 Идентификаторы транзакций и установление трехсторонней связи

Идентификаторы транзакций являются целыми числами в диапазоне от 1 до 999 999 999. Агенты вызова могут решить использовать конкретное пространство номеров для каждого из шлюзов, которыми они управляют, или использовать одно и то же пространство номеров для всех шлюзов, которые принадлежат к некоторой произвольной группе. Агенты вызова могут решить разделить нагрузку управления большим шлюзом между несколькими независимыми процессами. Эти процессы будут использовать совместно одно и то же пространство номеров транзакций. Имеются несколько возможных реализаций этого совместного использования, таких как централизованное распределение идентификаторов транзакций или предварительное распределение непересекающихся диапазонов идентификаторов для разных процессов. Реализации ДОЛЖНЫ гарантировать, что однозначно определяемые (уникальные) идентификаторы транзакций будут присвоены всем транзакциям, которые исходят от любого агента вызова и которые переданы отдельному шлюзу в течение периода в  $T_{hist}$  секунд. Шлюзы могут легко обнаружить дублированные транзакции, только "взглянув" на идентификатор транзакции.

Параметр подтверждения ответа может содержаться в любой команде. Он переносит множество "диапазонов подтвержденных идентификаторов транзакций" для полученных окончательных ответов; временные ответы НЕ ДОЛЖНЫ подтверждаться.

Шлюзы MGCP могут предпочесть исключить копии ответов на транзакции, идентификатор которых включен в "диапазоны подтвержденных идентификаторов транзакций", полученных в сообщении; однако тот факт, что транзакция была выполнена, ДОЛЖЕН еще сохраняться в течение  $T_{hist}$  секунд. Кроме того, при получении сообщения подтверждения ответа<sup>31</sup> ответ, который подтверждается этим сообщением, может быть исключен. Шлюзы должны безоговорочно отбрасывать дальнейшие команды от этого агента вызова, когда идентификатор транзакции попадает в эти диапазоны, а ответ был выдан менее чем  $T_{hist}$  секунд назад.

Пусть  $term_{new}$  и  $term_{old}$  будут именем конечной точки, соответственно, в новой команде  $cmd_{new}$  и в некоторой старой команде  $cmd_{old}$ . Тогда идентификаторы транзакций, подлежащие подтверждению в команде  $cmd_{new}$ , СЛЕДУЕТ определять следующим образом:

- 1) Если  $term_{new}$  не содержит каких-либо подстановочных знаков, то это:
  - a) Неподтвержденные ответы на старые команды, где  $term_{old}$  равен  $term_{new}$ .
  - b) Необязательно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "any-of" ("любой из"), а возвращенное в ответе имя конечной точки было  $term_{new}$ .
  - c) Необязательно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "all" ("все"), а  $term_{new}$  охвачен подстановочным знаком в имени  $term_{old}$ .
  - d) Необязательно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "any-of" ("любой из"), имя конечной точки не было возвращено, а  $term_{new}$  охвачен подстановочным знаком в имени  $term_{old}$ .
- 2) Если  $term_{new}$  содержит подстановочный знак "all" ("все"), то это:
  - a) Необязательно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "all" ("все"), а  $term_{new}$  охвачен подстановочным знаком в  $term_{old}$ .
- 3) Если  $term_{new}$  содержит подстановочный знак "any of" ("любой из"), то это:
  - a) Необязательно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "all" ("все"), а  $term_{new}$  охвачен подстановочным знаком в  $term_{old}$ , если подстановочный знак "any of" ("любой из") в  $term_{new}$  был заменен подстановочным знаком "all".

Данный ответ НЕ СЛЕДУЕТ подтверждать в двух отдельных сообщениях.

Следующие примеры иллюстрируют использование этих правил:

- Если  $term_{new}$  есть "aaln/1", а  $term_{old}$  есть "aaln/1", тогда старый ответ может быть подтвержден по правилу 1a.

---

<sup>31</sup> В противоположность команде с параметром "подтверждение ответа".

- Если  $term_{new}$  есть "aaln/1", а  $term_{old}$  есть "\*", тогда старый ответ может быть подтвержден по правилу 1с.
- Если  $term_{new}$  есть "aaln/\*", а  $term_{old}$  есть "\*", тогда старый ответ может быть подтвержден по правилу 2а.
- Если  $term_{new}$  есть "aaln/\$", а  $term_{old}$  есть "aaln/\*", тогда старый ответ может быть подтвержден по правилу 3а.

Значения "диапазонов подтвержденных идентификаторов транзакций" НЕ СЛЕДУЕТ использовать, если прошло более  $T_{hist}$  секунд с того момента, когда шлюз выдал свой последний ответ этому агенту вызова, или когда шлюз возобновляет работу. В этой ситуации команды следует принимать и обрабатывать без какого бы то ни было тестирования в отношении идентификатора транзакции.

Итак, ответ НЕ СЛЕДУЕТ подтверждать, если он был получен более чем  $T_{hist}$  секунд назад.

Сообщения, подтверждающие ответы, могут быть переданы и получены в любом порядке. Шлюз должен сохранять объединение подтвержденных идентификаторов транзакций, принятых в последних командах.

## 7.8 Временные ответы

В некоторых случаях время завершения транзакций может быть значительно больше, чем в других<sup>32</sup>. В сигнализации NCS в качестве транспортного протокола используется протокол UDP, а надежность обеспечивается избирательными повторными передачами с использованием выдержек времени, которые основаны на оценке суммы времени прохождения сигнала в прямом и обратном направлениях в сети и времени завершения транзакции. Поэтому значительный разброс времени завершения транзакции остается нерешенным, когда требуется быстрое обнаружение потери сообщения без большой служебной нагрузки.

Чтобы преодолеть эту проблему, ДОЛЖЕН генерироваться временный ответ, если ожидается, что время завершения транзакции превысит небольшой период времени (РЕКОМЕНДУЕТСЯ период в 200 мс). Временный ответ подтверждает прием команды, хотя результат выполнения команды может быть еще не известен, например, из-за ожидания резервирования ресурсов. В качестве ориентира, транзакция, для которой требуется завершение внешней связи, например резервирование сетевых ресурсов, должна генерировать временный ответ. Кроме того, если получена дублированная команда CreateConnection или ModifyConnection, но еще не закончено выполнение транзакции, в обратном направлении ДОЛЖЕН быть послан временный ответ.

Строгая семантика транзакции будет означать, что временные ответы не должны возвращать никакую другую информацию, кроме факта, что транзакция выполняется в текущее время; однако при оптимистическом подходе допускается возможность возвращения некоторой информации, позволяющей уменьшить задержку, которая будет иметь место в системе в противном случае.

Временные ответы ДОЛЖНЫ передаваться только в ответ на команду CreateConnection или ModifyConnection. Чтобы уменьшить время задержки в системе, во временный ответ на команду CreateConnection ДОЛЖНЫ быть включены идентификатор соединения и описание сеанса. Если описание сеанса будет возвращаться командой ModifyConnection, то описание сеанса ДОЛЖНО быть включено также во временный ответ. Если транзакция завершается успешно, то информация, возвращенная во временном ответе, ДОЛЖНА быть повторена в окончательном ответе. Неповторение этой информации или изменение какой-либо ранее представленной информации в успешном ответе считается ошибкой протокола. Если имеет место ошибка транзакции, то возвращается код ошибки; ранее возвращенная информация больше не является действительной.

Выполняемая в текущее время транзакция CreateConnection или ModifyConnection ДОЛЖНА быть аннулирована, если для конечной точки принята команда DeleteConnection. В таком случае ответ на аннулированную транзакцию СЛЕДУЕТ вернуть автоматически, а если обнаружена повторная передача аннулированной транзакции, ответ на аннулированную транзакцию ДОЛЖЕН быть возвращен (СЛЕДУЕТ использовать код ошибки 407).

<sup>32</sup> Например, когда ресурсы резервируются и предоставляются внешним источником как часть транзакции.

При получении временного ответа период выдержки времени для рассматриваемой транзакции ДОЛЖЕН быть установлен на значительно более высокое значение для этой транзакции ( $T_{longtran}$ ). Назначение этого таймера состоит главным образом в том, чтобы обнаружить отказ конечной точки. Значение  $T_{longtran}$  по умолчанию составляет 5 секунд; однако это значение может быть изменено в процессе подготовки соединения.

Когда заканчивается выполнение транзакции, посылается окончательный ответ, а устаревший к данному моменту временный ответ исключается. Чтобы гарантировать быстрое обнаружение потерянного окончательного ответа, окончательные ответы, выдаваемые после временных ответов для транзакции, ДОЛЖНЫ подтверждаться. Поэтому конечная точка ДОЛЖНА включать пустой параметр "ResponseAck" в такие и только такие окончательные ответы. Наличие параметра "ResponseAck" в окончательном ответе будет инициировать ответ "подтверждение ответа", подлежащий обратной передаче к конечной точке. Таким образом, SMS ДОЛЖНА генерировать ответ "подтверждение ответа" всякий раз, когда она получает окончательный ответ, содержащий пустой параметр "ResponseAck", независимо от приема временного ответа на транзакцию, поскольку временный ответ мог быть потерян. Ответ "подтверждение ответа" будет включать идентификатор транзакции ответа, который он подтверждает, в заголовок ответа. Прием данного ответа "подтверждение ответа" подчиняется тем же стратегиям и процедурам выдержки времени и повторной передачи, что и ответы на команды (см. п. 6.4); другими словами отправитель окончательного ответа повторно передаст его, если "подтверждение ответа" не будет принято вовремя. Ответ "подтверждение ответа" никогда не подтверждается.

## **8 Безопасность**

Если бы несанкционированные объекты могли использовать протокол MGCP, они были бы способны устанавливать несанкционированные вызовы или создавать помехи для санкционированных вызовов. Обеспечение безопасности не входит в протокол MGCP. Вместо этого в протоколе MGCP предполагается наличие нижнего уровня, обеспечивающего фактическую безопасность.

Требования по безопасности и соответствующие решения для сигнализации NCS содержатся в Рекомендации МСЭ-Т J.170, к которой следует обращаться для получения более подробной информации.

## Приложение А

### Пакеты событий

В данном Приложении определяется начальное множество пакетов событий для различных типов конечных точек, которые в настоящее время определены для встроенных клиентов в проекте IPCablecom. Для типов конечных точек встроенных клиентов, перечисленных в таблице, определены следующие пакеты.

Каждый пакет определяет имя пакета для пакета и коды событий, а также описания, для каждого из событий в пакете. В таблицах событий/сигналов для каждого пакета имеется пять столбцов:

<b>Код</b>	Однозначно определяемый код события пакета, используемый для события/сигнала.
<b>Описание События</b>	Краткое описание события/сигнала. В этом столбце проставляется отметка о проверке, если событие может быть запрошено контроллером медийного шлюза. Как вариант, могут быть включены один или несколько следующих символов:
"P"	указывает на то, что событие устойчивое;
"S"	указывает на то, что событие находится в таком состоянии, которое может быть проверено;
"C"	указывает на то, что сигнал/событие могут быть обнаружены/поданы в соединении.
<b>Сигнал</b>	Если в данном столбце для события ничего нет, то сигнал о событии не может быть передан в команде контроллером медийного шлюза. В противном случае тип события определяют следующие символы:
"OO"	Сигнал "включено/выключено". Сигнал будет включенным до тех пор, пока по команде от контроллера медийного шлюза не будет выключен, и наоборот.
"TO"	Сигнал выдержки времени. Этот сигнал длится в течение заданного времени, пока не будет заменен новым сигналом. Обеспечиваются значения выдержки времени по умолчанию. Нулевое значение указывает на то, что период выдержки времени бесконечен. Процесс подготовки может изменить эти значения по умолчанию.
"BR"	Короткий сигнал. У события короткая, известная длительность.
<b>Дополнительная информация</b>	Предоставляется дополнительная информация о событии/сигнале, например длительность сигналов TO по умолчанию.

Если не указано другое, все события/сигналы обнаруживаются/применяются в конечных точках, а звуковая информация, генерируемая ими, не передается по какому-либо соединению, которое может иметь конечная точка. Однако звуковая информация, генерируемая событиями/сигналами, которые обнаруживаются/применяются в соединении, будет передаваться по соответствующему соединению независимо от режима соединения.

В настоящее время определены следующие пакеты для конечных точек аналоговых линий доступа:

- Line (линейный)
- FAX (факсимильной связи)
- Пакет метрик VoIP (передача голоса по IP-протоколу)

#### А.1 Линейный пакет

Имя пакета: L

Коды, перечисленные в таблице А.1 используются для идентификации событий и сигналов для линейных пакетов для "аналоговых линий доступа":

**Таблица А.1/J.162 – Коды линейных пакетов для событий и сигналов**

Код	Описание	Событие	Сигнал	Дополнительная информация
0–9,*,#,A, B,C,D	Сигналы DTMF	√	BR	
bz	Тональный сигнал занятости	–	TO	Выдержка времени = 30 секунд
cf	Тональный сигнал подтверждения	–	BR	
ci(ti, nu, na)	Идентификатор вызывающего абонента	–	BR	"ti" означает время, "nu" означает номер, а "na" означает имя
dl	Тональный сигнал готовности к набору номера	–	TO	Выдержка времени = 16 с
ft	Тональный сигнал факсимильной связи	√	–	
hd	Переход в состояние "телефонная трубка снята"	P, S	–	
hf	Флэш-сигналы	P	–	
hu	Переход в состояние "телефонная трубка положена"	P, S	–	
L	Большая длительность сигнала DTMF	√	–	
ld	Соединение большой продолжительности	C	–	
ma	Запуск медиаданных	C	–	
mt	Модемные тональные сигналы	√	–	
mwi	Индикатор ожидания сообщения	–	TO	Выдержка времени = 16 с
oc	Операция завершена	√	–	
of	Ошибка операции	√	–	
osi	Открытый интервал переключения	–	TO	По умолчанию = 900 мс
ot	Тональный предупредительный сигнал о снятии телефонной трубки	–	TO	Выдержка времени = бесконечная
r0, r1, r2, r3, r4, r5, r6 или r7	Позывные сигналы (0..7)	–	TO	Выдержка времени = 180 с
rg	Вызывной сигнал	–	TO	Выдержка времени = 180 с
ro	Тональный сигнал требования освобождения цепи	–	TO	Выдержка времени = 30 с
rs	Пакет коротких вызывных сигналов	–	BR	
rt	Тональный сигнал контроля посылки вызова	–	C, TO	Выдержка времени = 180 с
sl	Прерывистый сигнал готовности к набору номера	–	TO	Выдержка времени = 16 с
t	Таймер	√	–	
TDD	Тональные сигналы устройств электросвязи для слабослышащих (TDD)	√	–	
vmwi	Визуальный индикатор ожидания сообщения	–	OO	
wt1, wt2, wt3, wt4	Тональные сигналы ожидания вызова	–	TO	Выдержка времени = [(MaxReps + 1) + (MaxReps * Задержка)] в сек., где значение по умолчанию MaxReps = 1, а Задержки = 10, как определено в в сигнализации NCS спецификации MIB.
X	Подстановочный знак для сигналов MFPB (DTMF)	√	–	Соответствует любой из цифр "0–9"

Ниже приводятся определения отдельных событий и сигналов:

**Сигналы DTMF (0–9,\*,#,A,B,C,D):** Обнаружение и генерирование двухтональных многочастотных сигналов DTMF описано в стандарте GR-506-CORE – LSSGR: Сигнализация для аналоговых интерфейсов, раздел 15. Попытка передачи сигналов DTMF к телефону в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефонная трубка положена).

**Тональный сигнал занятости (bz):** Станция занята – это комбинация двух тональных сигналов АС с частотой 480 и 620 Гц и величиной –24 дБм каждый, чтобы получился общий уровень мощности –21 дБм. Модуляция тонального сигнала занятости станции выглядит так: 0,5 секунд включен, 0,5 секунд выключен, повторение. Попытка передачи тонального сигнала занятости к телефону в автономной режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Тональный сигнал подтверждения (cf):** Тональный сигнал подтверждения использует те же частоты и уровни, что и тональный сигнал готовности к набору номера (350 и 440 Гц), но с модуляцией: 0,1 секунды включен, 0,1 секунды выключен, повторение три раза. Попытка передачи тонального сигнала подтверждения к телефону, находящемуся в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Идентификатор вызывающего абонента (ci(time, number, name)):** См. стандарты TR-NWT-001188 и TR-NWT-000031. Каждое из трех полей является необязательным, однако каждая из запятых всегда будет включена.

- Параметр **время (time)** кодируется в виде "MM/DD/HH/MM", где MM – это значение из двух цифр между 01 и 12 для обозначения месяца; DD – это значение из двух цифр между 1 и 31 для обозначения дня, а час (HH) и минута (MM) – это значения из двух цифр, закодированные согласно военному местному времени, например 00 означает полночь, 01 – 1 час ночи, а 13 – 1 час дня.
- Параметр **номер (number)** кодируется в виде символьной строки в коде ASCII из десятичных цифр, которые определяют номер вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы, однако они будут игнорироваться.
- Параметр **имя (name)** кодируется в виде строки символов в коде ASCII, которая определяет имя вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы.

Буква "P" в поле номера или имени используется для указания на частный номер или имя, а буква "O" – на недоступный номер или имя. Следующий пример иллюстрирует использование сигнала "идентификатор вызывающего абонента":

```
S: ci(10/14/17/26, "555 1212", CableLabs)
```

В дополнение к общим требованиям сигнализации, МТА ДОЛЖЕН поддерживать как минимум комбинацию одного сигнала в конечной точке с идентификатором вызывающего абонента (ci) через одну и ту же строку SignalRequest (например, S: rg, ci(time,number,name)) и сигнал в каждой конечной точке, связанный с конечной точкой.

**Тональный сигнал готовности к набору номера (dl):** Тональный сигнал готовности к набору номера – это комбинация двух продолжительных тональных сигналов АС с частотой 350 и 440 Гц и величиной –13 дБм каждый, чтобы получился общий уровень мощности –10 дБм. Попытка передачи тонального сигнала готовности к набору номера к телефону в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Тональный сигнал факсимильной связи (ft):** Событие "тональный сигнал факсимильной связи" генерируется каждый раз, когда обнаруживается факсимильный вызов по наличию преамбулы факсимильного сообщения, согласно Рекомендации МСЭ-Т V.21, или тональных сигналов T.30 CNQ (если предоставляются). См. IPCablecom Аудио/Видео Кодеки МСЭ-Т Рек. J.161, и Рекомендации МСЭ-Т T.30 и V.21.

**Переход в состояние "телефонная трубка снята" (hd):** См. GR-506, LSSGR, раздел 12.

**Флэш-сигналы (hf):** См. GR-506, LSSGR, раздел 12.

**Переход в состояние "телефонная трубка положена" (hu):** Синхронизация сигнала "телефонная трубка положена" активизируется для ответа на флэш-сигналы.

**Большая длительность сигнала MFPB (DTMF) (L):** "Большая длительность сигнала DTMF" наблюдается, когда этот сигнал выдается в течение более двух секунд. В этом случае шлюз обнаружит два последовательных события: первое, когда сигнал был распознан как сигнал DTMF, затем, двумя секундами позднее, сигнал большой продолжительности.

**Соединение большой продолжительности (ld):** "Соединение большой продолжительности" обнаруживается, когда соединение устанавливалось в течение времени, превышающего определенный для его установки период времени. Значение по умолчанию составляет 1 час; однако оно может быть изменено в процессе подготовки соединения.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они созданы.

**Запуск медиаданных (ma):** Событие "запуск медиаданных" имеет место в соединении, когда по соединению принимается первый действительный<sup>33</sup> медиапакет согласно протоколу RTP. Это событие может быть использовано для синхронизации локального сигнала, например, сигнала контроля посылки вызова, с поступлением медиапакета от другой стороны.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они созданы.

**Модемные тональные сигналы (mt):** Модемный тональный сигнал (mt): Событие "модемный тональный сигнал" генерируется каждый раз, когда обнаруживается вызов передачи данных по присутствию тонального сигнала ответа (ANS) согласно Рекомендации МСЭ-Т V.25 с опрокидыванием фазы или без него или по присутствию модифицированного тонального сигнала ответа (ANSam) согласно Рекомендации МСЭ-Т V.8 с опрокидыванием фазы или без него. См. Рекомендации МСЭ-Т V.25 и V.8.

**Индикатор ожидания сообщения (mwi):** Тональный сигнал индикатора ожидания сообщения использует те же частоты и уровни, что и тональный сигнал готовности к набору номера (350 и 440 Гц на -13 дБм каждый), но с модуляцией: 0,1 секунды включен, 0,1 секунды выключен, повторенный 10 раз, сопровождаемый постоянным применением тонального сигнала готовности к набору номера. Попытка передачи сигнала индикатора ожидания сообщения к телефону, находящемуся в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Открытый интервал переключения (osi):** Интерфейс передачи данных голосового диапазона, Раздел 2.2.2.

**Операция завершена (oc):** Событие "операция завершена" генерируется, когда шлюз получает запрос на применение одного или нескольких сигналов типа ТО в конечной точке и когда один или несколько таких сигналов закончились без прерывания при обнаружении запрашиваемого события, такого как переход в состояние "телефонная трубка снята" или "набранная цифра". Сообщение о завершении может содержать в качестве параметра имя сигнала, время существования которого закончилось, как в случае:

O: L/oc(L/d1)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

O: L/oc(B/rt@0A3F58)

Когда запрашивается событие "операция завершена", оно не может быть параметризовано любыми параметрами события. Когда имя пакета опущено, то предполагается имя пакета по умолчанию.

Событие "операция завершена" может быть сгенерировано дополнительно, как определено в базовом протоколе, например, когда успешно завершается встроенная команда ModifyConnection, как в случае:

O: L/oc(B/C)

Следует отметить, что "B" используется здесь в качестве префикса для сообщаемого параметра.

**Ошибка операции (of):** В общем случае событие "ошибка операции" может быть сгенерировано, когда конечная точка получила запрос на применение одного или нескольких сигналов типа ТО в конечной точке и до окончания выдержки времени в одном или нескольких сигналах произошел сбой.

---

<sup>33</sup> Когда используются услуги аутентификации и обеспечения защиты целостности, пакет RTP не считается действительным, пока он не пройдет проверку на обеспечение защиты.

Сообщение о завершении может содержать в качестве параметра имя сигнала, в котором произошел сбой, как в случае:

O: L/of (L/rg)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

O: L/of (L/rt@0A3F58)

Когда запрашивается событие "ошибка операции", параметры события не могут быть определены. Когда имя пакета опущено, предполагается имя пакета по умолчанию.

Событие "ошибка операции" может быть сгенерировано дополнительно, как определено в базовом протоколе, например, когда имеет место неуспешное выполнение встроенной команды ModifyConnection, как в случае:

O: L/of (B/C(M(sendrecv(AB2354))))

Следует отметить, что "B" используется здесь в качестве префикса для сообщаемого параметра.

**Тональный предупредительный сигнал о снятии телефонной трубки (ot):** Тональный предупредительный сигнал о снятии телефонной трубки (тональный сигнал ROH) - это раздражающий шумовой сигнал, который издает телефон, если трубка повешена неверно. Тональный сигнал ROH генерируется при помощи комбинации из четырех тональных сигналов на частотах 1400 Гц, 2060 Гц, 2450 Гц и 2600 Гц с модуляцией: 0,1 с включен, 0,1 с выключен, повторение. GR-506-CORE, раздел 17.2.8 содержит более подробную информацию о требуемых уровнях мощности. Попытка передачи тонального предупредительного сигнала о снятии телефонной трубки к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Позывные сигналы (r0, r1, r2, r3, r4, r5, r6 или r7):** Значения от r1 до r5 определяются шаблоном позывных сигналов от 1 до 5. При процессе подготовки соединения МОЖНО определить модуляцию звонка для каждого из r0-r7 сигналов. МТА ДОЛЖЕН поддерживать обеспечение для r0, r6 и r7. Попытка посылки позывного сигнала к телефону, который находится в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Вызывной сигнал (rg):** Значение для rg определяется шаблоном позывных сигналов 1. При процессе подготовки соединения МОЖНО определить модуляцию звонка. Вызывной сигнал может быть параметризован параметром сигнала "rep", который определяет максимальное число используемых вызывных циклов (повторений). Применение вызывного сигнала до 6 циклов выглядит следующим образом:

S: rg(rep=6)

Попытка передачи вызывного сигнала к телефону, находящемуся в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Тональный сигнал требования освобождения цепи (ro):** Тональный сигнал требования освобождения цепи – это комбинация двух тональных сигналов АС с частотой 480 и 620 Гц и величиной –24 дБм каждый, чтобы получился общий уровень мощности –21 дБм. Модуляция тонального сигнала требования освобождения цепи: 0,25 секунды включен, 0,25 секунды выключен, постоянное повторение. Попытка передачи тонального сигнала требования освобождения цепи к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Пакет коротких вызывных сигналов (rs):** Пакет коротких вызывных сигналов, также известный как "сигнал-напоминание", представляет собой пакет коротких электрических сигналов, который может быть использован в физической линии переадресации (если она свободна) для указания на то, что вызов был переадресован, и для напоминания пользователю, что активизирована функция переадресации вызова. В США определено, что это должен быть второй пакет коротких мощных электрических сигналов длительностью 0,5(–0,+0.1) секунд. См. TR-TSY-000586 – Подствойства передачи вызова. При процессе подготовки соединения МОЖНО определить модуляцию звонка для

пакета коротких вызывных сигналов. Попытка передачи этого сигнала к телефону, который находится в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Тональный сигнал контроля посылки вызова (rt):** Акустический сигнал посылки вызова - это комбинация двух тональных сигналов АС с частотой 440 и 480 Гц и величиной –19 дБм каждый, чтобы получился общий уровень мощности –16 дБм. В США модуляция акустического сигнала посылки вызова должна быть 2 секунды включен, со следующими за ними 4 секундами выключен. Определение тонального сигнала определяется внутренними характеристиками тонального сигнала контроля посылки вызова и МОЖЕТ быть переопределен через процесс подготовки соединения.

Сигнал контроля посылки вызова может быть использован как в конечной точке, так и в соединении.

При применении сигнала контроля посылки вызова в конечной точке попытка передачи тональных сигналов контроля посылки вызова считается ошибкой, если конечная точка находится в автономном режиме (телефонная трубка положена), и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)). Когда сигнал контроля посылки вызова используется в соединении, такая проверка проводится не должна.

**Прерывистый тональный сигнал готовности к набору номера (sl):** Прерывистый тональный сигнал готовности к набору номера (также называемый тональным сигналом готовности к набору номера, требующим ответа) генерируется подачей тонального сигнала подтверждения со следующим за ним тональным сигналом готовности к набору номера. Прерывистый тональный сигнал готовности к набору номера может быть параметризован параметром сигнала "del", который будет определять задержку в миллисекундах между тональным сигналом подтверждения и тональным сигналом готовности к набору номера<sup>34</sup>. Применение прерывистого тонального сигнала готовности к набору номера с задержкой в 1,5 секунды между тональным сигналом подтверждения и тональным сигналом готовности к набору номера выглядит следующим образом:

S: sl(del=1500)

Попытка передачи прерывистого тонального сигнала готовности к набору номера к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Таймер (t):** Как описано в п. 6.1.5, таймер Т – это обеспечиваемый таймер, который может быть аннулирован только путем ввода сигналов DTMF. Когда таймер используется с действием "накопление согласно отображению цифр", он не запускается до тех пор, пока не введена первая цифра, и перезапускается после ввода каждой новой цифры до тех пор, пока не наступит соответствие или несоответствие этому отображению цифр. В этом случае таймер Т функционирует как таймер времени между передачей цифр и принимает одно из двух значений –  $T_{par}$  или  $T_{crit}$ . Когда для строки цифр требуется по крайней мере еще одна цифра, чтобы соответствовать какой-либо из комбинаций отображения цифр, таймер Т принимает значение  $T_{par}$ , соответствующее временным соотношениям при неполном наборе номера. Для получения полного соответствия отображению цифр таймер Т принимает значение  $T_{crit}$ , соответствующее критическим временным соотношениям. Примером использования является:

S: dl  
R: [0-9T] (D)

Если таймер Т используется без действия "накопление согласно отображению цифр", он принимает значение  $T_{crit}$ , запускается немедленно и просто аннулируется (но не перезапускается), как только вводится цифра. В этом случае таймер Т может быть использован как таймер времени между передачей цифр, когда применяется передача с перекрытием, например:

R: [0-9] (N) , T(N)

Следует отметить, что в текущий момент может использоваться только одна из двух форм, поскольку данное событие может быть определено только один раз.

---

<sup>34</sup> Этот параметр необходим, например, для быстрого набора номера.

Значение по умолчанию для  $T_{\text{par}}$  равно 16 секундам, а для  $T_{\text{crit}}$  – 4 секундам. Оба этих значения могут быть изменены в процессе подготовки соединения.

**Тональные сигналы устройств электросвязи для слабослышащих (TDD):** Событие TDD генерируется каждый раз, когда обнаруживается вызов TDD – см., например, Рекомендацию МСЭ-Т V.18.

**Визуальный индикатор ожидания сообщения (vmwi):** Передача сообщений VMWI будет соответствовать требованиям стандарта TR-N-000030, раздел 2.3.2: "Передача данных в состоянии телефонная трубка положена, не связанных с вызывными сигналами", и руководства CPE в SR-TSV-002476. Сообщения VMWI будут посылаться от встроенного клиента к присоединенному оборудованию только тогда, когда линия свободна. Если новое сообщение поступает тогда, когда линия занята, то сообщение "индикатор VMWI" будет задерживаться до тех пор, пока линия снова не станет свободной. При перезагрузке конечной точки ей не следует пытаться выключить визуальный индикатор ожидания сообщения, даже если ей кажется, что его следует выключить. Агент вызова должен периодически обновлять визуальный индикатор оборудования CPE. См. TR-NWT-001401 – Общие требования к визуальному индикатору ожидания сообщения; и интерфейс передачи данных голосового диапазона.

**Тональный сигнал 1 ожидания вызова (wt1, ..., wt4):** Число относится к используемому шаблону тональных сигналов. Длительность шаблона тональных сигналов длится до 1 секунды. GR-571-CORE (FSD 01-02-1201) указывает, что следует по отдельности проиграть два шаблона тональных сигналов за период в 10 секунд. По умолчанию число повторений и задержка между повторениями может быть изменена в процессе подготовки соединения. Тональный сигнал ожидания вызова по умолчанию - это тональный сигнал 440-Гц, подаваемый в течение  $300 \pm 50$  мс. Шлейф разговора следует прерывать максимум через 400 мс после подачи каждого тонального сигнала CW. Если данный сигнал запрошен, встроенный клиент проиграет два шаблона тональных сигналов до того, как кончится выдержка времени для сигнала "Т0". Попытка передачи тональных сигналов ожидания вызова к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

Расчет выдержки времени по умолчанию основывается на числе повторений по умолчанию (или предварительном числе повторений), задержке между повторениями по умолчанию (или предварительной задержке между повторениями) и длительностью времени для генерирования самого тонального сигнала ожидания вызова (1 секунда). Например, если задержка ожидания вызова установлена на 12 секунд, а максимальное число повторений ожидания вызова установлено на 2, то тональный сигнал ожидания вызова будет услышан до трех раз (один раз проигран и дважды повторен). Следовательно, выдержка времени по умолчанию рассчитывается сложением времени, затраченным на то, чтобы проиграть тональный сигнал ожидания вызова три раза (3 секунды), и задержки между повторениями ( $2 \cdot 12$  секунд). Рассчитанная задержка времени по умолчанию в этом примере составляет, следовательно, 27 секунд. По запросу сигнал ожидания вызова может быть параметризован параметром сигнала "к", являющимся вытяжкой из описанного выше расчета выдержки времени по умолчанию.

**Подстановочный знак для сигналов MFPB (DTMF) (X):** Подстановочный знак для сигналов DTMF соответствует любой цифре двухтонального многочастотного набора номера между 0 и 9.

## A.2 Пакет факсимильной связи

Следующий протокол контроля медиашлюза пакета факсимильной связи определяет механизм поддержки вызовов факсимильной связи. NCS налагает следующие дополнительные требования кроме тех, что определены в протоколе контроля медиашлюза пакета факсимильной связи:

- Реализации ДОЛЖНЫ соответствовать кодировке версии протокола NCS.
- Конечная точка ДОЛЖНА быть способна к генерированию события "t38(запуск)", если обнаружен тональный сигнал CNG T.30. Кроме того, конечная точка ДОЛЖНА обеспечивать опции конфигурирования для отключения данной возможности.

Комитет по инженерным проблемам интернета (IETF)

Ф. Андризен

Проект интернет-документа

Системы Cisco

Документ: draft-andreasen-mgcp-fax-04.txt

30 августа 2004 г.

Категория: Информационный

Протокол контроля медиашлюза пакета факсимильной связи

Статус данной заметки

Принимая данный Проект интернет-документа, я удостоверяю, что любой прилагаемый патент или другие IPR права, о которых я знаю, заявлены, а те, о которых я буду узнавать, будут заявлены в соответствии с RFC 3668.

Проект интернет-документа - это рабочая документация комитета по инженерным проблемам интернета (IETF), его областей деятельности и его рабочих групп. Следует отметить, что другие группы могут также распространять свои рабочие документации в качестве Проекта интернет-документа.

Проект интернет-документа - это черновая документация которая действительна в течение максимум 6 месяцев и в любой момент времени может быть обновлена, замещена или сделана недействительной другой документацией. Использовать Проект интернет-документа в качестве справочной литературы нецелесообразно, так же как и цитировать его иначе, чем в качестве "документации на стадии разработки".

Со списком текущих Проектов интернет-документов можно ознакомиться на: <http://www.ietf.org/ietf/lid-abstracts.txt>.

Со списком каталогов Проектов интернет-документов можно ознакомиться на <http://www.ietf.org/shadow.html>.

Срок действия данного Проекта интернет-документа истекает 1 марта 2005 г.

### Реферат

Данный документ определяет пакет протокола контроля медиашлюза (MGCP) для поддержки вызовов факсимильной связи. Пакет позволяет поддерживать факсимильную связь двумя различными способами. Первый использует МСЭ-Т Рекомендацию T.38 для реле факсимильной связи под контролем агента вызова. Второй позволяет шлюзу самостоятельно выбирать метод для передачи факсимильного сообщения, также как и разбираться с деталями вызова факсимильной связи, без привлечения агента вызова.

### Соглашения, используемые в данном документе

Ключевые слова "ДОЛЖЕН", "НЕ ДОЛЖЕН", "ТРЕБУЕТСЯ", "БУДУТ", "НЕ БУДУТ", "СЛЕДУЕТ", "НЕ СЛЕДУЕТ", "РЕКОМЕНДУЕТСЯ", "МОЖЕТ", и "НЕОБЯЗАТЕЛЬНО" в данном документе должны интерпретироваться как описано в ВСП 14, RFC-2119 [RFC2119].

## Содержание

1. Вступление
2. Определение пакета факсимильной связи
  - 2.1 LocalConnectionOptions
    - 2.1.1 Процедура T.38 (ограниченный или неограниченный)
    - 2.1.2 Процедура шлюза
    - 2.1.3 Процедура отключения
    - 2.1.4 Режим работы
    - 2.1.5 Обнаружение вызова факсимильной связи
    - 2.1.6 Определение того, какие процедуры запрашивать
  - 2.2 События и сигналы
    - 2.2.1 Факсимильная связь, контролируемая шлюзом (gwfax)
    - 2.2.2 Отсутствие специальной обработки факсимильной связи (nopfax)
    - 2.2.3 Реле факсимильной связи T.38 (t38):
  - 2.3 Параметры соединения
  - 2.4 Согласование параметров T.38
  - 2.5 Анализ реализации
    - 2.5.1 Адрес IP и порт медиаданных для T.38
    - 2.5.2 Чувствительность к регистру
    - 2.5.3 Булев индикатор после параметров T.38
3. Примеры протекания вызова
  - 3.1 Контролируемый агентом вызова ограниченный T.38
  - 3.2 Множественные и различные опции
  - 3.3 Взаимодействие с конечными точками SIP
4. Анализ безопасности
5. Анализ IANA
6. Нормативная литература
7. Справочная литература
8. Благодарности
9. Адрес автора

## 1. Вступление

Данный документ определяет пакет протокола контроля медиашлюза (MGCP) [RFC3435], который дает шлюзам, контролируемым MGCP, возможность поддерживать вызовы факсимильной связи. Пакет позволяет поддерживать факсимильную связь двумя различными способами. Первый использует МСЭ-Т Рекомендацию T.38, при использовании либо UDP/TLS, либо TCP (см. [T38]) для реле факсимильной связи под контролем агента вызова. Второй позволяет шлюзу самостоятельно выбирать метод для передачи факсимильного сообщения, также как и разбираться с деталями вызова факсимильной связи, без привлечения агента вызова.

Определение пакета факсимильной связи предоставлено в разделе 2, а в разделе 3 представлены два примера протекания вызова, показывающие, как его использовать. Анализ безопасности можно найти в разделе 4, за которым следуют анализ IANA и ссылки.

## 2. Определение пакета факсимильной связи

Пакет определяется для факсимильной связи. Пакет определяет новые опции местного соединения (`localconnectionoptions`), события и параметры соединения так, как указано ниже.

Имя пакета: FXR

Версия пакета: 0

### 2.1 LocalConnectionOptions

Новый параметр `LocalConnectionOptions` (LCO) факсимильной связи определяется для обработки факсимильных сообщений. Агент вызова предоставляет данные параметры (LCO) факсимильной связи для указания медиашлюзу желаемой процедуры обработки факсимильных сообщений. Параметры (LCO) факсимильной связи содержат список желаемых процедур обработки факсимильных сообщений, расположенных по приоритету, где наиболее предпочтительная процедура указана первой. Если параметр явно включен в команду, шлюз ДОЛЖЕН иметь возможность использовать по крайней мере одну из перечисленных процедур, чтобы команда была успешной. Список может в настоящее время указывать на одну или более следующих процедур (см. Раздел 2.1.1 до 2.1.4 для дальнейших деталей по ним):

- \* Ограниченный T.38: Использовать T.38 [T38] либо с UDP/TLS, либо с TCP для реле факсимильной связи и контролировать его через агента вызова. Предполагая использование допустимой процедуры (см. раздел 2.1.1), переключение на процедуры T.38 будет инициироваться обнаружением факсимильной связи и генерированием события "t38(запуск)" (см. Раздел 2.2). Чтобы использовать данный режим требуется указание поддержки T.38 удаленной стороной, как описано далее в разделе 2.1.1.
- \* Неограниченный T.38: Идентичен процедуре ограниченного T.38, за исключением того, что для использования процедуры не требуется указание поддержки T.38 удаленной стороной.
- \* Выключен: Не активизирует какой-либо специальной процедуры для факсимильной связи, за исключением регулирования компенсации и возможного переключения на другой кодек.
- \* Шлюз: Позволяет шлюзу контролировать и решать каким образом обрабатывать вызовы факсимильной связи без привлечения агента вызова. Сюда включен случай, когда шлюз не делает ничего специального для факсимильной связи, таким образом, такая процедура всегда может быть поддерживаемой. Если шлюз не активизирует специальной процедуры при обнаружении факсимильной связи, он генерирует событие "gwfax(запуск)", так что агент вызова будет о ней уведомлен (см. раздел 2.2). Агенту вызова СЛЕДУЕТ тогда отозваться выдачей потенциально конфликтующих команд шлюзу, до тех пор, пока шлюз не завершит свою специальную процедуру обработки факсимильной связи.

Шлюз, который заканчивает не будучи способен активизировать какой-либо специальной процедуры факсимильной связи, при обнаружении факсимильной связи генерирует событие "norfax(запуск)" (см раздел 2.2).

Набор возможных значений (т. е. процедур) для LCO факсимильной связи может расширяться. Префикс "x-", который указывает на необязательное расширение, и префикс "x+", который указывает на обязательное расширение, резервируются для целей, определяемых разработчиком.

В командах `CreateConnection` значение LCO факсимильной связи по умолчанию "шлюз". В командах `ModifyConnection` значение LCO факсимильной связи по умолчанию устанавливается на его текущее значение в соединении. Таким образом, если параметры `LocalConnectionOptions` либо опущены, либо не включены в команду `ModifyConnection`, для соединения остается предыдущее значение LCO факсимильной связи, но без оказания какого-либо воздействия на команду; следовательно, шлюз может не применять какой-либо специальной процедуры для факсимильной связи. Если агент вызова хочет удостовериться, только что команда выполнена успешно, когда применяется процедура факсимильной связи, то команда должна явно включать LCO факсимильной связи.

В качестве примера для сказанного, предположим, что команда `CreateConnection` успешно определяет использование "ограниченного Т.38", а команда `ModifyConnection` теперь получена без LCO факсимильной связи, но в `RemoteConnectionDescriptor` указывается отсутствие поддержки Т.38; в этом случае команда `ModifyConnection` будет успешной, однако процедуры Т.38 при обнаружении факсимильной связи больше активизированы не будут. Если вместо этого агент вызова включит LCO факсимильной связи, установленные на "ограниченный Т.38", команда будет неуспешной.

Если предоставлены значения множества параметров факсимильной связи, шлюз ДОЛЖЕН выбрать одну из этих процедур, указанных в порядке их предоставления, за исключением следующего:

1. Если "шлюз" был бы выбран, и это выразилось бы в отсутствии применения специальной процедуры, и
2. есть другие процедуры, кроме "выключен", которые указаны после "шлюза" (например, "t38")

тогда шлюз ДОЛЖЕН использовать наиболее предпочтительную из тех последующих процедур, которые поддерживаются. Если не одна из этих последующих процедур не поддерживается, то шлюз возвращает решение не активизировать какой-либо специальной процедуры для факсимильной связи. Для дальнейших деталей по определению поддерживаемых процедур обращаться к разделу 2.1.4.

Параметр LCO факсимильной связи кодируется в виде ключевого слова "fx" (с префиксом имени пакета по [RFC3435]), сопровождаемым разделенными двоеточием и точкой с запятой списком значений, где ограниченный Т.38 кодируется как "t38", неограниченный Т.38 кодируется как "t38-loose", шлюз кодируется как "gw", а выключен кодируется как "off".

Следующий пример иллюстрирует использование PCMU или G.729 для аудио кодирования и реле факсимильной связи ограниченного Т.38 (предпочтительнее) или контроля шлюза для факсимильной связи:

```
L: a:PCMU;G729, fxr/fx:t38;gw
```

Следует отметить, что MGCP позволяет команде `CreateConnection` опускать оба параметра `LocalConnectionOptions` и `RemoteConnectionDescriptor`, тем самым давая шлюзу возможность самому решать какие параметры медиаданных использовать. Если поддерживается пакет факсимильной связи Т.38, шлюз, таким образом, может выбирать использовать аудио или Т.38 реле факсимильной связи. Скорее всего, агент вызова требует использования или одного, или другого, и, следовательно, ему НЕ СЛЕДУЕТ пропускать оба параметра `LocalConnectionOptions` и `RemoteConnectionDescriptor` в команде `CreateConnection`.

При проверке возможностей LCO факсимильной связи могут быть возвращены со списком поддерживаемых параметров обработки факсимильной связи, разделенных точкой с запятой. Значения "t38", "off" и "gw" МОГУТ быть опущены в таком списке, так как они всегда внедрены. Шлюзам, которые реализуют дополнительные параметры, СЛЕДУЕТ возвращать эти дополнительные параметры, если возможности проверяются, иллюстрируются на следующем примере:

```
A: a:image/t38, fxr/fx:mypar, ...
```

В следующем подразделе предоставляются дополнительные подробности вышеописанных процедур факсимильной связи.

### 2.1.1 Процедура Т.38 (ограниченного или неограниченного)

Если шлюз получает инструкцию использовать одну из процедур Т.38 (ограниченного или неограниченного), также известную как режим Т.38, контролируемого агентом вызова, строка "m=", возвращенная в SDP, не будет указывать на использование основанного на UDPTL или основанного на TCP Т.38 (если только шлюз не был также проинструктирован об использовании "изображение/t38" для медиапотоков). Любой другой объект, видя данный SDP, не будет знать, поддерживается ли Т.38 или нет, и соответственно, безопасно ли попытаться включить Т.38 при обнаружении факсимильной связи. Для разрешения этой дилеммы информация о возможности использования Т.38 (если он поддерживается) ДОЛЖНА включаться при использовании расширений простого объявления возможностей SDP [RFC3407]. Другая информация о возможностях также включается в независимости от того, авторизовал ли агент вызова их использование в команде обработки соединения или нет. Дальнейшие попытки использовать их в действительности могут, конечно, не увенчаться успехом, например, потому что агент вызова LCOs не разрешает их использовать. Следующий пример иллюстрирует дескриптор возможностей RFC 3407 – следует обратить внимание на включение обеих текущих возможностей (аудио) и латентный (Т.38), как определено в RFC 3407:

```
m=audio 3456 RTP/AVP 18
```

```
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 18
a=cdsc: 2 image udptl t38
```

Для уточнения списка параметров, относящихся к T.38, который должен быть включен в SDP, обращайтесь к T.38 дополнение D [T38].

При обнаружении факсимильной связи шлюз, который был успешно проинструктирован об использовании одной из процедур T.38 будет:

1. Инициирование процедуры реле факсимильной связи T.38 и перекрывает медиаканал как в направлении приема, так и в направлении отправления (если только медиаканал уже не использует T.38).
2. Генерирует событие "t38(запуск)".
3. Ожидает дальнейших инструкций от агента вызова, чтобы инициировать действительный обмен медиаданными (если только медиаканал уже не использует T.38).

Агент вызова инструктирует шлюз о выполнении обмена медиаданными, посредством отправления ему команды `ModifyConnection` с "изображение/t38", указанным в качестве метода кодирования в параметре `LocalConnectionOptions` (получение команды `ModifyConnection` без параметра `LocalConnectionOptions`, но с `RemoteConnectionDescriptor`, содержащим строку "m=" с типом MIME "изображение/t38" означало бы то же самое). При нормальной процедуре согласования кодека MGCP (см. [RFC3435] раздел 2.6), если параметр `RemoteConnectionDescriptor` был также включен, то ему необходимо включить в себя строку "m=" с "изображением/t38" в качестве допустимого формата медиаданных, для того чтобы команда была успешной. Шлюз может выбирать между транспортными протоколами UDPTL и TCP на свое собственное усмотрение при нормальной процедуре согласования кодеков (на практике реализации на базе TCP в настоящее время редки).

Если `RemoteConnectionDescriptor` не был включен в команду `ModifyConnection`, посланную шлюзу, который иницирует процедуру T.38, возможно (а в действительности наиболее вероятно), что последний полученный `RemoteConnectionDescriptor` не включает в себя строку "m=", содержащую "изображение/t38" в качестве допустимого формата медиаданных. В этом случае конечная точка не может посылать медиаданные T.38 другой стороне. Вместо этого конечная точка ДОЛЖНА ожидать обновленный `RemoteConnectionDescriptor` с "изображением/t38" в качестве допустимого формата медиаданных и поддерживаемым транспортным протоколом (UDPTL или TCP). Процедура факсимильной связи T.38 продолжает работу, когда получен допустимый `RemoteConnectionDescriptor`. Допустимый `RemoteConnectionDescriptor` содержит строку "m=" с типом MIME "изображение/t38" (использование нормального синтаксиса SDP) и поддерживаемым транспортным протоколом (UDPTL или TCP). Если вызов факсимильной связи неуспешен, например, из-за выдержки времени факсимильной связи во время либо ожидания команды от агента вызова переключиться на "изображение/t38", либо при ожидании допустимого `RemoteConnectionDescriptor`, ДОЛЖНО быть сгенерировано событие "t38(остановка)" или "t38(отказ)". Когда процедура T.38 заканчивается, ДОЛЖНО быть сгенерировано событие "t38(остановка)" или "t38(отказ)".

### 2.1.2 Процедура шлюза

Шлюз, используя процедуру шлюза, также известную как режим, контролируемый шлюзом, может инициировать специальную обработку факсимильной связи при обнаружении вызова факсимильной связи. Детали такой специальной обработки факсимильной связи выходят за рамки данного документа. Однако для того, чтобы использовать специальную обработку факсимильной связи, ее поддержка ДОЛЖНА быть согласована с другой стороной посредством передачи и распознавания соответствующих параметров через `LocalConnectionDescriptor` и `RemoteConnectionDescriptor`. Если с другой стороны не указывается поддержка желаемой специальной обработки факсимильной связи, шлюз НЕ ДОЛЖЕН пытаться ее инициировать. Если специальная обработка факсимильной связи инициирована, то ДОЛЖНО быть сгенерировано событие "gwfax(запуск)", давая, тем самым, агенту вызова возможность различать режим, контролируемый агентом вызова или шлюзом, во время продолжающегося информирования об актуальном переходе на факсимильную связь. Когда специальная обработка факсимильной связи шлюзом заканчивается, ДОЛЖНО быть сгенерировано событие "gwfax(остановка)" или "gwfax(отказ)".

### 2.1.3 Процедура отключения

Шлюз, используя процедуру "отключения" не будет активизировать какую-либо специальную процедуру факсимильной связи, например, T.38 при обнаружении факсимильной связи. Однако шлюз может по-прежнему регулировать локальную компенсацию и/или переключиться на альтернативный кодек при необходимости (в частности, это не препятствует использованию T.38 на базе RTP). Также ДОЛЖНО быть сгенерировано событие "nopfax(запуск)"; однако соответствующее событие "остановка" сгенерировано не будет.

Генерирование события "остановки" привело бы к тому, что шлюз должен был бы сделать вывод, что заканчивается вызов факсимильной связи, что влияет на обработку медиапотока. Однако при использовании режима "отключения" такая обработка возникать не должна.

#### 2.1.4 Режим работы

Для каждого из вышеуказанных режимов `RemoteConnectionDescriptor` предоставляет информацию о процедуре(рах), поддерживаемых другой стороной. Следующие правила используются для определения того, какую процедуру применить:

1. Чтобы не указал агент вызова в `LocalConnectionOptions` факсимильной связи для текущей команды, это ДОЛЖНО быть выполнено. Если шлюз не может выполнить какую-либо из указанных опций, команда выдает отказ (РЕКОМЕНДУЕТСЯ код ошибки 532 – неподдерживаемое значение(я) в `LocalConnectionOptions`).
2. Если предоставлены как `LocalConnectionOptions` факсимильной связи, так и `RemoteConnectionDescriptor`, выбранная процедура ДОЛЖНА поддерживаться обеими сторонами – в настоящее время она выдается только для "ограниченного Т.38". Процедура может быть выполнена удаленной стороной, если:
  - \* соответствующий тип медиаданных MIME, например, "изображение/t38", включен в строку "m=" в `RemoteConnectionDescriptor`, или
  - \* соответствующий тип медиаданных MIME включен в качестве возможности (см. [RFC3407]) в `RemoteConnectionDescriptor`.Если шлюз не может выбрать какую-либо из процедур в `LocalConnectionOptions` факсимильной связи, команда выдает отказ (РЕКОМЕНДУЕТСЯ код ошибки 532). Следует отметить, что "Т.38 Loose" (неограниченный Т.38), "шлюз", и "отключение" по определению могут всегда быть поддержаны реализацией, поддерживающей данный пакет, в независимости от того, что указывает `RemoteConnectionDescriptor`.
3. Если агент вызова не включает `LocalConnectionOptions` факсимильной связи или `RemoteConnectionDescriptor` в команду, шлюз ДОЛЖЕН продолжать использование той процедуры, которую он использует в настоящий момент.
4. Если агент вызова не включает каких-либо `LocalConnectionOptions` факсимильной связи, но `RemoteConnectionDescriptor` включен, шлюз ДОЛЖЕН при выборе процедуры следовать правилу 2. При этом ДОЛЖНО использоваться значение `LocalConnectionOptions` факсимильной связи по умолчанию, т. е. "шлюз" в `CreateConnection`, или текущее значение в `ModifyConnection`. В случае `ModifyConnection` результат команды не зависит от способности шлюза выбирать одну из этих процедур (как описано в разделе 2.1). Следует отметить, что это не относится к команде `CreateConnection`, поскольку там значение по умолчанию всегда поддерживается по определению.
5. Полученный в предыдущий раз `RemoteConnectionDescriptor` не влияет на то, какая процедура будет выбрана. Только `RemoteConnectionDescriptor`, полученный с текущей командой, влияет на выбор процедуры. Однако для того, чтобы послать медиаданные определенного типа (например, "изображение/t38"), `RemoteConnectionDescriptor`, полученный в последнюю очередь ДОЛЖЕН включать в себя соответствующую строку медиаданных.

Следующий пример иллюстрирует использование вышеперечисленных правил:

По правилу 1 шлюз, поддерживающий только реле факсимильной связи стандарта Т.38, выдаст отказ команде, которая содержит только опцию "m=par", в то время как команда, содержащая "t38-loose", "gw", "off" или отсутствие LCO факсимильной связи, будет успешной. Команда, которая содержит только "t38", т. е. использование только Т.38 в "ограниченном" режиме, может быть или не быть успешной (в зависимости от `RemoteConnectionDescriptor`).

Шлюз, поддерживающий Т.38, при получении команды `CreateConnection` с LCO обработки факсимильной связи, установленными на "t38", и `RemoteConnectionDescriptor` без возможности использования Т.38 и без медиапотока Т.38, выдаст отказ по правилу 2. Если LCO обработки факсимильной связи включала бы либо "t38-loose", "gw" или "off", то команда была бы успешной, и любая включенная процедура могла бы быть выбрана.

Предположим, что шлюз, поддерживающий T.38, успешно выполнил команду CreateConnection с обработкой факсимильной связи, установленной на "t38" (т. е. ограниченный). Если шлюз теперь получит команду ModifyConnection без LCO обработки факсимильной связи, но с RemoteConnection Descriptor, у которого нет ни возможности использования T.38, ни медиапотока с "изображением /t38", команда будет успешной (поскольку правило 1 в этом случае влияния не имеет). Однако по правилу 2 и 4, процедура T.38 иметь место не будет. Если бы вместо этого CA снова включил LCO обработки факсимильной связи, установленные на "t38", команда получила бы отказ по правилу 2.

Наконец, следует отметить, что переключение T.38 может быть инициирован либо через один, либо через оба исходящий и оконечный шлюзы и, следовательно, реализации ДОЛЖНЫ быть готовы к иметь с этим дело. Сюда включен случай, когда обе стороны инициируют переключение, который может, например, возникнуть, если исходящее устройство факсимильной связи генерирует тональный сигнал вызова (CNG), а оконечное устройство факсимильной связи обнаруживает преамбулу факсимильной связи (см. [T30]) до того, как на оконечной стороне было выполнено переключение на T.38.

#### **2.1.5 Определение вызова факсимильной связи**

Вызов факсимильной связи может быть обнаружен несколькими различными средствами, например, преамбулой факсимильной связи V.21, тональным сигналом CNG T.30, или сигналами V.8, в зависимости от используемого метода передачи факсимильных сообщений. Реализации данного пакета ДОЛЖНЫ как минимум определять вызов факсимильной связи, базирующийся на преамбуле факсимильной связи V.21.

МОЖЕТ быть сделано инициирование базирующегося на тональном сигнале CNG T.30; это считается в общем допустимым для G3 и низкоскоростной факсимильной связи. Однако при использовании его с T.38 версией 2 или раньше, он окажет влияние на высокоскоростную факсимильную связь V.34. Причина этого заключается в том, что T.38 версии 2 (и раньше) не поддерживает ANSam и сигналы CM V.8, используемые с факсимильной связью V.34, и следовательно, факсимильной связи V.34 снизят скорость G3 (14.400 бит/с) или ниже при использовании T.38 версии 2 (или ранее). Таким образом, будет сообщено о нескольких редких случаях генерирования модемами тонального сигнала CNG T.30 для вызова не факсимильной связи; такие модемы генерировали бы неправильный инициатор факсимильной связи. Как следствие вышесказанного, РЕКОМЕНДУЕТСЯ, чтобы реализации данного пакета, поддерживающие обнаружение факсимильной связи на базе CNG T.30, предоставляли конфигурацию опций для его отключения для T.38 версии 2 (или ранее).

#### **2.1.6 Определение, какую процедуру запрашивать**

Важно понимать последствия использования какой-либо из определенных выше процедур. Кроме того, может быть запрошено множество альтернативных процедур, однако не все комбинации имеют смысл. В данном разделе изучены оба этих вопроса.

Использование режима ограниченного T.38 идеально в среде, где известно, что другие конечные точки генерируют описания возможности использования RFC 3407 с информацией о реле факсимильной связи T.38. Если в такой среде получен RemoteConnectionDescriptor без возможностей реле факсимильной связи T.38, становится понятно, что другая сторона не поддерживает T.38, и следовательно, нужно избегать безуспешной попытки переключиться на T.38 (которая, в свою очередь, может привести к отказу вызова факсимильной связи). Если неизвестно поддерживают ли другие конечные точки дескрипторы возможности RFC 3407 или нет, выполнение обмена менее прозрачно. Преимущество заключается в том, что переключение на T.38 можно пытаться осуществить только в том случае, если известно, что другая сторона его поддерживает, однако конечные точки, которые не отображают поддержку T.38, могут все равно его поддерживать; однако T.38 не будет использоваться для них, что, в свою очередь, может привести к необоснованному отказу факсимильной связи для кодеков с низкой полосой пропускания и сетей с потерями.

Использование режима неограниченного T.38 влечет за собой те же рассуждения, как и для ограниченного T.38, однако за и против поменялись местами. Если одноранговая конечная точка не поддерживает T.38, режим неограниченного T.38 будет все равно пытаться переключиться на T.38 (безуспешно), что, в свою очередь, может привести к отказу вызова факсимильной связи. С другой стороны, если одноранговая\конечная точка не поддерживает дескрипторы возможности использования RFC 3407, но одноранговая конечная точка в действительности поддерживает T.38, T.38 все равно использовался бы для этого режима.

Подводя итоги, нет единого правильного ответа на вопрос: использовать ли режим ограниченного или неограниченного T.38; это зависит от возможностей конечной точки, заключенных в компромисс между потенциально допускающими отказ вызовами факсимильной связи из-за нехватки индикаторов возможностей (где, напротив, поддерживается T.38) по сравнению с потенциально допускающими отказ вызовами факсимильной связи из-за безуспешного переключения на T.38 (потому что T.38, в действительности, не поддерживался). Следует отметить, что агент вызова может иметь другие средства помимо дескрипторов возможностей RFC 3407, для определения поддерживает ли одноранговая точка T.38 или нет. Например, когда

в качестве протокола сигнализации с другими одноранговыми устройствами используется SIP (например, агент вызова или другие устройства SIP), метод ОПЦИЙ SIP (SIP OPTIONS) может быть использована для выяснения поддерживается ли Т.38. Также, если агент вызова позволяет использование кодеков с большей шириной полосы с избыточностью там, где поддержка Т.38 не отображена, вызовы факсимильной связи могут все равно быть успешными без использования Т.38, даже в сетях с неигнорируемой потерей пакетов.

Если выбирается режим, контролируемый шлюзом, специальная обработка факсимильной связи будет иметь место только, если две одноранговых конечных точки поддерживают один и тот же метод обработки факсимильной связи; обратите внимание, что детали действующего метода целиком и полностью зависят от производителя. Также следует отметить, что если две одноранговых конечных точки не поддерживают один и тот же метод обработки факсимильной связи, или метод не отображен в обмениваемом SDP, специальная обработка факсимильной связи иметь место не будет. Кроме того, агент вызова не будет знать, что это именно этот случай до тех пор, пока не начнется передача факсимильного сообщения и не будет сгенерировано сообщение "nopfax(запуск)".

Режим отключения прямолинеен; Специальная процедура для обработки факсимильной связи иметь место не будет, за исключением обычной обработки компенсации и возможно переключения на кодек с большей шириной полосы.

Рассмотрев отдельные процедуры более подробно, мы в настоящее время разработали некоторые комбинации процедур, которые могут быть запрещены:

- \* Ограниченный Т.38: Если процедура ограниченного Т.38 помещена за процедурой неограниченного Т.38 или процедурой отключения (обе из которых всегда поддерживаются), она не будет выбрана. Кроме того, не имеет большого смысла запрашивать одновременно ограниченный и неограниченный Т.38.
- \* Неограниченный Т.38: Процедура неограниченного Т.38 всегда поддерживается, поэтому любая процедура, указанная после неограниченного Т.38, выбрана не будет.
- \* Шлюз: Процедура, контролируемая шлюзом, всегда поддерживается. Если процедура, контролируемая шлюзом привела бы к отсутствию специальной процедуры факсимильной связи и были представлены дальнейшие опции (за исключением "отключения"), эти процедуры будут опробованы. Если ни одна из этих процедур не предоставлена, специальная процедура факсимильной связи иметь место не будет.
- \* Отключение: Процедура отключения поддерживается всегда. Любая процедура, указанная за ней, не будет выбрана.

## 2.2 События и сигналы

Следующие события определены для поддержки вышесказанного:

Обозначение	Определение	R	Длительность, S
Gwfax	Факсимильная связь, контролируемая шлюзом	X	
nopfax	Отсутствие специальной обработки факсимильной связи	x	
t38	Реле факсимильной связи Т.38	x	

Определения отдельных событий представлены в следующих подразделах.

### 2.2.1 факсимильная связь, контролируемая шлюзом (gwfax)

Событие "факсимильная связь, контролируемая шлюзом" возникает тогда, когда шлюз управляет процедурой факсимильной связи либо при запуске, либо при остановке, либо в случае отказа. Событие кодируется как "gwfax", и определяются следующие параметры события, которые применяются только к ObservedEvents:

- \* **запуск:** процедура факсимильной связи, контролируемая шлюзом, была инициирована. Агенту вызова СЛЕДУЕТ отозваться выдачей шлюзу инструкций по обработке медиаданных до тех пор, пока не будет сгенерировано событие "gwfax(stop)" или "gwfax(failure)".
- \* **остановка:** процедура контролируемая шлюзом, закончена, и шлюз не обнаружил никаких ошибок. Следует отметить, что это необязательно означает успешно переданный факс. Скорее всего, это показывает, что процедура факсимильной связи, контролируемая шлюзом, была закончена и сама процедура не сталкивается с какими-либо ошибками. Параметры медиаданных для соединения остаются такими же, как до начала процедуры факсимильной связи, управляемой шлюзом.

- \* **отказ:** процедура факсимильной связи, контролируемая шлюзом, закончена аномальным образом. Какой-то тип проблемы был зафиксирован в процедуре факсимильной связи, контролируемой шлюзом, и она была завершена. Параметры медиаданных для соединения остаются такими же, как до начала процедуры факсимильной связи, управляемой шлюзом.

Один из вышеуказанных параметров будет представлен при получении сообщения о событии. Событие "gwfax" МОЖЕТ быть параметризовано дополнительными параметрами в ObservedEvents, однако РЕКОМЕНДУЕТСЯ, чтобы один из вышеуказанных параметров был первым предоставленным параметром. Неизвестные параметры ДОЛЖНЫ игнорироваться.

Следующий пример иллюстрирует кодирование события "gwfax":

```
O: fxr/gwfax(start)
O: fxr/gwfax(stop, foobar)
```

### 2.2.2 Отсутствие специальной обработки факсимильной связи (nopfax)

Событие "Отсутствие специальной обработки факсимильной связи" возникает тогда, когда отсутствует процедура специальной обработки факсимильной связи, и обнаруживается вызов факсимильной связи. Это может произойти либо из-за отсутствия запроса процедуры специальной обработки факсимильной связи (включая "отключение"), либо из-за согласования, ведущего к тому, что процедура специальной обработки факсимильной связи невозможна. Событие кодируется как "nopfax", и определяются следующие параметры события, которые применяются только к ObservedEvents:

- \* **Запуск:** Процедура специальной обработки факсимильной связи отсутствует, однако в настоящее время обнаружен вызов факсимильной связи. Агент вызова может быть должен отдать дальнейшие команды, чтобы гарантировать успешный вызов факсимильной связи факсимильной связи (например, переключение на другой кодек).

Вышеуказанный параметр будет представлен при получении сообщения о событии. Событие "nopfax" МОЖЕТ быть параметризовано дополнительными параметрами в ObservedEvents, однако РЕКОМЕНДУЕТСЯ, чтобы один из вышеуказанных параметров был первым предоставленным параметром. Неизвестные параметры ДОЛЖНЫ игнорироваться. Следует отметить, что данное событие в текущий момент не может быть параметризовано параметрами "остановка" или "отказ", поскольку это определяется только в начале вызова факсимильной связи.

Следующий пример иллюстрирует кодирование события "nopfax":

```
O: fxr/nopfax(start)
```

### 2.2.3 Реле факсимильной связи Т.38 (t38)

Событие "Реле факсимильной связи Т.38" возникает тогда, когда одна из процедур реле факсимильной связи Т.38 (ограниченного или неограниченного) находится либо при запуске, либо при остановке, либо в случае отказа. Событие кодируется как "t38", и определяются следующие параметры события, которые применяются только к ObservedEvents:

- \* **запуск:** В конечной точке был обнаружен вызов факсимильной связи и процедура реле факсимильной связи Т.38, контролируемая агентом вызова, была инициирована. Агенту вызова СЛЕДУЕТ модифицировать каждую сторону соединения на использование формата медиаданных "изображение/t38", если только они уже его не используют. Следует отметить, что до тех пор пока имеет влияние использование процедуры реле факсимильной связи Т.38, контролируемой агентом вызова, при обнаружении вызова факсимильной связи будет сгенерировано событие в независимости от текущего метода кодирования для любого соединения конечной точки (включая "изображение/t38"). Событие Т38(запуск) ДОЛЖНО быть сгенерировано конечной точкой при вызове факсимильной связи максимум один раз, в независимости от того, запрашивалось ли оно снова в последующем списке запрашиваемых событий или нет.
- \* **остановка:** закончена, и шлюз не обнаружил никаких ошибок. Следует отметить, что это необязательно означает успешно переданный факс. Скорее всего, это показывает, что процедура реле факсимильной связи Т.38, контролируемая агентом вызова, была закончена и сама процедура не столкнулась ни с какими ошибками. Следует отметить, что в отличие от случая процедуры факсимильной связи, контролируемой шлюзом, параметры медиаданных, такие как кодеки, не возвращают автоматически значения, бывшие до начала процедуры факсимильной связи, управляемой шлюзом; однако компенсация эха и подавление пауз делают это посредством процедур, описанных в [RFC3435] раздел 2.3.5. Событие "t38(отказ)" НЕ ДОЛЖНО генерироваться, если только соответствующее событие "t38(запуск)" для рассматриваемой факсимильной связи не было сгенерировано ранее.

\* **отказ:** процедура реле факсимильной связи T.38, контролируемая агентом вызова, закончена аномальным образом. Какой-то тип проблемы был зафиксирован в процедуре факсимильной связи, контролируемой шлюзом, и она была завершена. Агент вызова может изъявить желание изменить параметры медиаданных для каждой из сторон соединения. Следует отметить, что в отличие от случая процедуры факсимильной связи, контролируемой шлюзом, параметры медиаданных, такие как кодеки, не возвращают автоматически значения, бывшие до начала процедуры факсимильной связи, управляемой шлюзом; однако компенсация эха и подавление пауз делают это посредством процедур, описанных в [RFC3435] раздел 2.3.5. Событие "t38(отказ)" НЕ ДОЛЖНО генерироваться, если только соответствующее событие "t38(запуск)" для рассматриваемой факсимильной связи не было сгенерировано ранее.

Один из вышеуказанных параметров будет представлен при получении сообщения о событии. Событие "t38" МОЖЕТ быть параметризовано дополнительными параметрами, однако РЕКОМЕНДУЕТСЯ, чтобы один из вышеуказанных параметров был первым предоставленным параметром. Неизвестные параметры ДОЛЖНЫ игнорироваться.

Следующий пример иллюстрирует кодирование события "t38":

O: fxr/t38(start)

O: fxr/t38(foobar, stop)

### 2.3 Параметры соединения

Параметры соединения для соединения, которые измеряют отправленные и полученные пакеты и октеты, ДОЛЖНЫ также включать пакеты с октеты для обработки факсимильной связи. Расчет интервала между прибытиями пакетов и средней задержки передачи однако МОЖЕТ быть не предоставлен, во время обработки соединения, например, если используется T.38. В таких случаях расчеты интервала между прибытиями пакетов и средней задержки передачи просто приостанавливаются, до того как расчеты могут продолжаться, например, при обратном переключении на медиапоток на базе RTP.

В Дополнение к этим параметрам, пакет факсимильной связи определяет следующие параметры соединения, которые МОЖЕТ поддерживать шлюз:

Число посланных страниц факсимильного сообщения (PGS):

Кумулятивное число страниц факсимильного сообщения, посланных конечной точкой за время жизни соединения. Параметр кодируется как "PGS", а значение предоставляется в виде строки размером до 9 десятичных цифр.

Число полученных страниц факсимильного сообщения (PGR):

Кумулятивное число страниц факсимильного сообщения, полученного конечной точкой за время жизни соединения. Параметр кодируется как "PGR", а значение предоставляется в виде строки размером до 9 десятичных цифр.

Следующий пример иллюстрирует использование данных параметров:

P: FXR/PGS=3, FXR/PGR=0, PS=1245, OS=62345, ...

### 2.4 Согласование параметров T.38

T.38 Дополнение D определяет число параметров T.38, которые могут быть согласованы в SDP. В настоящее время T.38 не определяет процедуры по согласованию данных параметров, а в частности, должна ли каждая сторона использовать одно и то же значение или нет. Из-за отсутствия данного определения было решено добавить такие определения и процедуры сюда. Однако предполагается, что T.38 будет rectify вышесказанное, что могло бы привести к конфликту между определениями и процедурами. Чтобы избежать такой ситуации вместо этого предполагается существование раздела T.38 предложение/ответ, где параметры дополнения D T.38 классифицируются либо как объявляемые, либо как согласуемые, а затем даются указания, как отображать такие определения и процедуры в пакете факсимильной связи MGCP, определенном здесь.

MGCP не определяет использование модели предложение/ответ, а вместо этого работает с концепцией команд обработки соединения (например, CreateConnection и ModifyConnection) которые могут включать RemoteConnectionDescriptor (SDP) и, в свою очередь, могут генерировать в своих ответах LocalConnectionDescriptor (SDP).

Если конечная точка MGCP получает команду CreateConnection без RemoteConnectionDescriptor, ей следует следовать соответствующим процедурам T.38 для генерирования начального предложения и вернуть результирующий SDP в своем LocalConnectionDescriptor.

Если конечная точка MGCP получает команду CreateConnection с RemoteConnectionDescriptor, ей следует следовать соответствующим процедурам T.38 для получения начального предложения и генерирования ответа на нее. Результирующий SDP возвращается в LocalConnectionDescriptor.

Если конечная точка MGCP получает команду ModifyConnection с RemoteConnectionDescriptor, она не может определить относится ли она к ответу на начальное предложение или на новое предложение. Это не касается объявляемых параметров, поскольку они могут быть указаны независимо в обоих направлениях. Однако согласованные параметры требуют некоторого анализа:

Если предлагающая сторона получает ответ на предыдущее предложение, согласование является законченным, а согласованные параметры больше не могут изменяться во время данного обмена предложение/ответ. Согласованные параметры могут быть предметом для определенной проверки действительности. И напротив, когда отвечающая сторона получает предложение, согласование открыто, и отвечающая сторона может изменить некоторые предлагаемые согласовываемые параметры. Поскольку конечная точка MGCP не знает в какой ситуации она находится, она не может провести проверку действительности "предлагающей стороны". Подобным же образом для того, чтобы быть уверенным, что любое требуемое согласование имеет место в действительности, ей необходимо обработать входящий SDP как предложение. Если SDP в действительности соответствует предложению, тогда это очевидно верное поведение. Однако, если SDP соответствует ответу, и один или более согласовываемых параметров изменен, то это приведет к появлению нового SDP. Агент вызова может обладать или не обладать достаточным "разумом", чтобы определить необходимо ли новому SDP другой обмен предложение/ответ.

Например, если начальное предложение (от команды CreateConnection без SDP), содержащее версию 2 факсимильной связи, а ответ (от команды CreateConnection с SDP), содержащее версию 0 факсимильной связи, то соответствующая команда ModifyConnection (с SDP) приведет к тому, что обновленный SDP с версией факсимильной связи будет также установлен на нуль. Если это единственная переменная в обновленном SDP, новый обмен предложение/ответ не понадобится. Следует отметить, что этот пример не сообщает, что в общем считается хорошей идеей для агента вызова проанализировать SDP, для того чтобы определить нужен ли новый обмен предложение/ответ или нет.

Наконец, команда ModifyConnection без SDP, которая генерирует SDP, должна приниматься во внимание. Сгенерированный SDP может либо соответствовать начальному обмену предложение/ответ, либо последующему обмену предложение/ответ. Конечной точке следует генерировать SDP, как если бы это была часть последующего обмена предложение/ответ. Если агент вызова не хочет такой семантики, он может вместо этого просто создать новое соединение.

## **2.5 Анализ внедрения**

### **2.5.1 IP-адрес и порт медиаданных для T.38**

Когда конечная точка получает инструкцию перейти на или с T.38 для медиапотока, ей СЛЕДУЕТ продолжать использование тех же самых IP-адреса и порта, что и используемый в настоящее время медиаток, поскольку это сведет к минимуму изменения при взаимодействии качества услуги, транслятора сетевых адресов (NAT) и Firewall (средств межсетевой защиты). Однако, если конечная точка обладает достаточной причиной, она МОЖЕТ решить не следовать данной рекомендации.

Когда конечная точка использует один и тот же порт для аудио RTP и T.38 либо с UDPTL, либо с TCP, пакеты одного типа (например, T.38) могут быть получены при ожидании пакетов другого типа (аудио RTP). Поскольку существует явная сигнализация, показывающая, какой тип ожидается в какой-либо конкретный момент времени, это не представляет собой дополнительных проблем. Другими словами, получатель не работает в качестве демультиплексера, чтобы определить, является ли полученный пакет аудио RTP или пакет T.38 UDPTL/TCP. Получатель просто обрабатывает входящие пакеты как обычно. Если ожидаются пакеты T.38, то входящие пакеты проверяются на соответствие T.38, если ожидаются пакеты аудио RTP, то входящие пакеты проверяются на соответствие аудио RTP.

### **2.5.2 Чувствительность к регистру**

IANA зарегистрировал строку верхнего регистра "UDPTL" в качестве идентификатора транспортного протокола для использования в T.38 на базе UDP. Однако примеры, приведенные в МСЭ-Т Рек. T.38, также как большинство (если не все) текущих реализаций, используют вместо этого строку нижнего регистра "udptl". Реализациям, соответствующим этому пакету, СЛЕДУЕТ сгенерировать строку нижнего регистра "udptl" и принять равнозначность строк нижнего, верхнего и смешанного (верхнего/нижнего) регистров.

Атрибут "T38MaxBitRate" неверно зарегистрирован в IANA в качестве "T38maxBitRate" ("m" нижнего регистра). В соответствии с примерами и практикой общих реализаций T.38 форму "T38MaxBitRate" СЛЕДУЕТ генерировать при реализациях, соответствующих данному пакету.

В общем случае РЕКОМЕНДУЕТСЯ, чтобы реализации данного пакета принимали кодирования всех атрибутов T.38 нижнего, верхнего и смешанного (верхнего/нижнего) регистров.

### 2.5.3 Булев индикатор после параметров T.38

Некоторые реализации неверно используют двоеточие (':'), сопровождаемое числом (ноль или единица) после атрибутов T38FaxFillBitRemoval, T38FaxTranscodingMMR и T38FaxTranscodingJBIG. Реализации, которые получают такие ошибочные кодировки МОГУТ интерпретировать значение ":0" как отсутствие поддержки опции и все другие значения как отображение поддержки рассматриваемой опции.

### 3. Пример протекания вызова

В этом разделе предоставлены два примера протекания вызова. Первый иллюстрирует вызов факсимильной связи T.38 под контролем агента вызова для обеих сторон: исходящей и оконечной. Второй иллюстрирует использование множественных и различных опций для двух сторон.

#### 3.1 Контролируемый агентом вызова ограниченный T.38

В этом примере обе стороны находятся под контролем агента вызова ограниченного T.38. Предполагается, что инициирование и отключение связи агентом вызова осуществляется через протокол инициирования сеанса (SIP) [RFC3261] (см. также [SIPfax]):

#	GW-o	CA-o	CA-t	GW-t
1	←	CRCX		
2	200 (sdp-o)	→		
3		INVITE (sdp-o)	→	
4			CRCX (sdp-o)	→
5				← 200 (sdp-t)
6		←	200 (sdp-t)	
7	←	MDCX (sdp-t)		
8	200	→		
9				←ANS/ T.30CED
10				←T.30 fax preamble
11			←	NTFY(t38start)
12			200	→
13			MDCX (t38)	→
14			←	200 (sdp-t2)
15		←	INVITE (sdp-t2)	
16	←	MDCX (sdp-t2)		
17	200 (sdp-o2)	→		
18		200 (sdp-o2)	→	
19			MDCX (sdp-o2)	→
20			←	200
21				(fax ends)
22			←	NTFY(t38stop)
24			200	→

Шаг 1:

Агент вызова выдает шлюзу команду `CreateConnection` с инструкцией использовать кодирование медиаданных PCMU и процедуру ограниченного контролируемого агентом вызова T.38. Следовательно, агент вызова просит шлюз уведомить его о событии t38:

```
CRCX 1000 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
L: a:PCMU, fxr/fx:t38
M: recvonly
R: fxr/t38
X: 1
```

Шаг 2:

Шлюз подтверждает команду и включает туда SDP с информацией о кодеке, а также о возможности использования RFC 3407:

```
200 1000 OK
I:1

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 3:

Иницирующий соединение агент вызова посылает сообщение SIP INVITE (приглашение), содержащее SDP, окончному агенту вызова.

Шаг 4:

Оконечный агент вызова выдает команду `CreateConnection` окончному шлюзу с инструкцией использовать кодирование медиаданных PCMU и процедуру ограниченного контролируемого агентом вызова T.38:

```
CRCX 2000 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
L: a:PCMU, fxr/fx:t38
M: sendrecv
R: fxr/t38
X: 20
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 5:

Оконечный шлюз поддерживает T.38, а включенный в него `RemoteConnectionDescriptor` показывает, что другая сторона также поддерживает T.38, так что запрошенная процедура ограниченного контролируемого агентом вызова T.38 может использоваться. Оконечный шлюз посылает обратно ответ об успехе со своим SDP, который также включает в себя информацию о возможностях:

```
200 2000 OK
I:2
v=0
```

```
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 6:

Оконечный агент вызова посылает в обратном направлении, к инициирующему агенту вызова, ответ SIP 200 OK, который в свою очередь посылает SIP ACK (не отображен).

Шаг 7:

Иницирующий агент вызова в свою очередь посылает инициирующему шлюзу команду ModifyConnection:

```
MDCX 1001 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
M: sendrecv
v=0
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Команда ModifyConnection не повторяет LocalConnectionOptions, посланную до этого. Так как рассматривается обработка факсимильной связи, то, следовательно, шлюз пытается продолжать использование текущей процедуры обработки факсимильных сообщений, т. е. ограниченного контролируемого агентом вызова T.38. Поскольку информация о возможностях показывает, что другая сторона поддерживает T.38, шлюз фактически будет способен использовать процедуру ограниченного контролируемого агентом вызова T.38. Если бы поддержки T.38 в RemoteConnectionDescriptor не было, то данная команда все равно была бы успешной, однако там не было бы специальной процедуры обработки факсимильных сообщений (поскольку ограниченный режим не мог бы поддерживаться).

Шаг 8

Шлюз подтверждает команду. В данный момент устанавливается вызов при использовании кодирования PCMU, и если вызов факсимильной связи обнаружен, будет инициирована процедура ограниченного контролируемого агентом вызова T.38.

Шаг 9-11:

Теперь возникает вызов факсимильной связи. В первую очередь, посылается тональный сигнал CED T.30 (так же известный как V.25 ANS), который в данном случае просто проходит через текущее кодирование PCMU. Поскольку согласно этой последовательности могут запускаться вызовы факсимильной и модемной связи, невозможно определить, что это вызов факсимильной связи до шага 10, где обнаруживается преамбула факсимильной связи V.21.

Шлюз получает указания применить процедуру ограниченного контролируемого агентом вызова T.38 для вызовов факсимильной связи, таким образом возникает событие "t38(start)" и об этом сообщается агенту вызова:

```
NTFY 2500 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
O: fxr/t38(start)
X: 20
```

Шаг 12:

Агент вызова подтверждает команду Notify:

```
200 2500 OK
```

Шаг 13:

Агент вызова инструктирует оконечный шлюз перейти вместо этого на использование MIME типа "изображение/t38":

```
MDCX 2002 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
I: 2
L: a:image/t38
R: fxr/t38
X: 21
```

Шаг 14:

Шлюз переходит на T.38, и отправляет в обратном направлении ответ об успехе с обновленным SDP:

```
200 2002 OK
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Следует отметить, что поскольку текущий RemoteConnectionDescriptor шлюза (в отличие от возвращенного здесь LocalConnectionDescriptor) не причисляет "изображение/t38" к допустимым методам кодирования, оконечный шлюз все равно приостанавливает медиаданные и теперь ожидает получения обновленного RemoteConnectionDescriptor с "изображение/t38".

Шаг 15:

Оконечный агент вызова посылает повторное приглашение (re-INVITE) иницирующему агенту вызова с обновленным SDP.

Шаг 16:

Иницирующий агент вызова затем посылает команду ModifyConnection иницирующему шлюзу:

```
MDCX 1003 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 17:

Иницирующий шлюз переходит на T.38 и посылает в обратном направлении ответ об успешном выполнении с обновленным SDP:

```
200 1003 OK
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=image 3456 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 18:

Иницирующий агент вызова посылает OK ответ SIP 200 с обновленным SDP окончному агенту вызова, который, в свою очередь, посылает SIP ACK (не отображен).

Шаг 19:

Оконечный агент вызова посылает команду `ModifyConnection` с обновленным SDP окончному шлюзу:

```
MDCX 2003 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
I: 2
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=image 3456 udpt1 t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udpt1 t38
```

Шаг 20:

Оконечный шлюз посылает обратно ответ об успешном выполнении:

```
200 2003 OK
```

Поскольку окончный шлюз теперь обладает `RemoteConnectionDescriptor` с "изображением/t38" в качестве допустимых медиаданных, он может начинать обмен T.38 с иницирующим шлюзом.

Шаг 21, 22:

Если факсимильная связь заканчивается, генерируется событие "t38(остановка)", о котором уведомляется агент вызова:

```
NTFY 2501 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
O: t38(stop)
X: 3
```

Шаг 23:

Агент вызова подтверждает команду `Notify`:

```
200 2501 OK
```

Вызов факсимильной связи обработан. Агент вызова может теперь принять решение перейти обратно на голосовой кодек, удалить соединение, или что-то еще.

### 3.2 Множественные и различные опции

В этом примере иницирующий шлюз получает команду использовать процедуру шлюза, в то время как окончному шлюзу дается выбор между процедурой шлюза и процедурой ограниченного t38. Кроме того, иницирующее устройство факсимильной связи генерирует тональный сигнал CNG.

#	GW-o	CA-o	CA-t	GW-t
1	←	CRCX		
2	200 (sdp-o)	→		
3		INVITE (sdp-o)	→	
4			CRCX (sdp-o)	→
5				← 200 (sdp-t)
6		←	200 (sdp-t)	
7	←	MDCX (sdp-t)		
8	200	→		
9	CNG→			
10				←ANS/T.30 CED
11				←T.30 fax p.
12			←	NTFY(t38 start)
13			200	→
14			MDCX (t38)	→
15			←	200 (sdp-t2)
16		←	INVITE (sdp-t2)	
17	←	MDCX (sdp-t2)		
18	200 (sdp-o2)	→		
19		200 (sdp-o2)	→	
20			MDCX (sdp-o2)	→
21			←	200
22				(fax ends)
23			←	NTFY(t38 stop)
24			200	→

Шаг 1:

Агент вызова выдает шлюзу команду CreateConnection, инструктируя его об использовании кодирования медиаданных PCMU и использовании процедуры шлюза. Следовательно, агент вызова запрашивает шлюз об уведомлении о событии gwfax:

```
CRCX 1000 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
L: a:PCMU, fxr/fx:gw
M: recvonly
R: fxr/gwfax
X: 1
```

Шаг 2:

Шлюз подтверждает получение команды, а также включает SDP с информацией о кодеке в информацию о возможностях:

```
200 1000 OK
I:1
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
a=X-FaxScheme123
```

Предполагается, что шлюз поддерживает некоторые другие схемы факсимильной связи и указывает это посредством включения атрибута "FaxScheme123" (схема факсимильной связи).

Шаг 3:

Иницирующий соединение агент вызова посылает сообщение SIP INVITE (приглашение), содержащее SDP, окончному агенту вызова.

Шаг 4:

Оконечный агент вызова выдает команду `CreateConnection` оконечному шлюзу с инструкцией использовать кодирование медиаданных PCMU и либо процедуру шлюза, либо процедуру ограниченного контролируемого агентом вызова T.38. Следовательно, агент вызова запрашивает шлюз об уведомлении о появлении обоих событий `gwfax` и `t38`:

```
CRCX 2000 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
L: a:PCMU, fxr/fx:gw;t38
M: sendrecv
R: fxr/t38, fxr/gwfax
X: 20
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
a=X-FaxScheme123
```

Шаг 5:

Оконечный шлюз не поддерживает никакой специальной обработки факсимильной связи шлюзом, однако он поддерживает T.38, а включенный `RemoteConnectionDescriptor` показывает, что другая сторона также поддерживает T.38, так что может быть использована согласованная процедура контролируемого агентом вызова ограниченного T.38. Оконечный шлюз посылает обратно успешный ответ со своим SDP, который также включает информацию о возможностях:

```
200 2000 OK
I:2
v=0
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 6:

Оконечный агент вызова посылает обратно исходящему агенту вызова ответ OK SIP 200, который, в свою очередь, посылает SIP ACK (не отображен).

Шаг 7:

Исходящий агент вызова, в свою очередь, посылает исходящему шлюзу команду `ModifyConnection`:

```
MDCX 1001 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
M: sendrecv
v=0
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Команда `ModifyConnection` не повторяет параметр `LocalConnectionOptions`, посланный до этого. Постольку, поскольку это касается обработки факсимильной связи, шлюз, следовательно, пытается продолжать использование текущей обработки факсимильной связи, т. е. процедуры шлюза. Однако возвращенная информация SDP не отображает поддержки для "FaxScheme123", и, таким образом, исходящий шлюз не активизирует какую-либо специальную процедуру обработки факсимильной связи для этого вызова.

Шаг 8:

Шлюз подтверждает получение команды. В этот момент устанавливается вызов при использовании кодирования PCMU и, если обнаружен вызов факсимильной связи, не возникнет никакой специальной процедуры обработки факсимильной связи.

Шаг 9-12:

Сначала исходящим устройством факсимильной связи генерируется тональный сигнал CNG, отображая тем самым, вызов факсимильной связи. Если шлюз использовал либо режимы T.38, либо согласовал с другой стороной поддержку специальной процедуры обработки шлюза, было бы сгенерировано событие "t38(запуск)" или "gwfax(запуск)" и переключение на T.38 (или специальную обработку шлюза) могло начинаться. Однако поскольку согласование с окончательным шлюзом не приводит к тому, что шлюз делает что-либо особое для факсимильной связи, такое событие не генерируется. Вместо этого теперь генерируется событие "nopfax(запуск)", но поскольку агент вызова не запрашивал данное событие, оно не обнаруживается, и следовательно, о нем не сообщается агенту вызова. Как следствие, тональный сигнал CNG просто пропускается через текущее кодирование PCMU без сообщения (исходящему) агенту вызова о вызове факсимильной связи.

Далее возникает тональный сигнал T.30 CED (также известный как V.25 ANS), который в данном случае также просто пропускается через текущее кодирование PCMU. Поскольку согласно этой последовательности могут запускаться вызовы и факсимильной, и модемной связи, невозможно определить, что это вызов факсимильной связи до шага 11, где обнаруживается преамбула факсимильной связи V.21.

Оконечный шлюз применяет процедуру контролируемого агентом вызова T.38 для вызовов факсимильной связи, таким образом возникает событие "t38(start)" и об этом сообщается агенту вызова:

```
NTFY 2500 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
O: fxr/t38(start)
X: 20
```

Шаг 13:

Агент вызова подтверждает команду `Notify`:

```
200 2500 OK
```

Шаг 14:

Агент вызова инструктирует окончательный шлюз перейти вместо этого на использование MIME типа "изображение/t38":

```
MDCX 2002 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
I: 2
L: a:image/t38
R: fxr/t38
X: 21
```

Шаг 15:

Шлюз переходит на T.38, и отправляет в обратном направлении ответ об успехе с обновленным SDP:

```
200 2002 OK
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udpt1 t38
a=sqn: 0
a=csrc: 1 audio RTP/AVP 0 18
a=csrc: 3 image udpt1 t38
```

Следует отметить, что поскольку последний полученный окончательным шлюзом RemoteConnectionDescriptor (в отличие от возвращенного здесь LocalConnectionDescriptor) не причисляет "изображение/t38" к допустимым методам кодирования, окончательный шлюз все равно приостанавливает медиаданные и теперь ожидает получения обновленного RemoteConnectionDescriptor с "изображение/t38".

Шаг 16:

Оконечный агент вызова посылает повторное приглашение (re-INVITE) иницирующему агенту вызова с обновленным SDP.

Шаг 17:

Иницирующийся агент вызова затем посылает команду ModifyConnection иницирующему шлюзу:

```
MDCX 1003 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 18:

Иницирующийся шлюз переходит на T.38 и посылает в обратном направлении ответ об успешном выполнении с обновленным SDP:

```
200 1003 OK
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=image 3456 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 19:

Иницирующийся агент вызова посылает OK ответ SIP 200 с обновленным SDP окончательному агенту вызова, который, в свою очередь, посылает SIP ACK (не отображен).

Шаг 20:

Оконечный агент вызова посылает команду ModifyConnection с обновленным SDP окончательному шлюзу:

```
MDCX 2003 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
C: 2
I: 2
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=image 3456 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 21:

Оконечный шлюз посылает обратно ответ об успешном выполнении:

200 2003 OK

Поскольку окончательный шлюз теперь обладает RemoteConnectionDescriptor с "изображением/t38" в качестве допустимых медиаданных, он может начинать обмен T.38 с иницилирующим шлюзом.

Шаг 22, 23:

Если факсимильная связь заканчивается, генерируется событие "t38(остановка)", о котором уведомляется агент вызова:

```
NTFY 2501 ds/ds1-1/2@gw-t.whatever.net MGCP 1.0
O: t38(stop)
X: 3
```

Шаг 24:

Агент вызова подтверждает команду Notify:

200 2501 OK

Вызов факсимильной связи обработан. Агент вызова может теперь принять решение перейти обратно на голосовой кодек, удалить соединение, или что-то еще.

### 3.3 Взаимодействие с конечными точками SIP

В этом примере рассмотрено взаимодействие с конечной точкой, неподдерживающей дескрипторы возможностей RFC 3407. Для обслуживания таких конечных точек используется режим неограниченного T.38 (с риском инициирования T.38 в конечной точке его не поддерживающей).

#	GW-o	CA-o	SIP-UA-t	fax
1	←	CRCX		
2	200 (sdp-o)	→		
3		INVITE (sdp-o)	→	
4		←	200 (sdp-t)	
5		ACK	→	
6	←	MDCX (sdp-t)		
7	200	→		
8				← ANS/ T.30 CED
9				← T.30 fax preamble
10		←	INVITE (sdp-t2)	
11	←	MDCX (sdp-t2)		
12	200 (sdp-o2)	→		
13		200 (sdp-o2)	→	
14		←	ACK	
15				(fax ends)
16		←	BYE	
17		200	→	

Шаг 1:

Агент вызова выдает шлюзу команду CreateConnection с инструкцией использовать кодирование медиаданных PCMU и процедуру неограниченного контролируемого агентом вызова T.38. Следовательно, агент вызова просит шлюз уведомить его о событии t38:

```
CRCX 1000 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
L: a:PCMU, fxr/fx:t38-loose
M: recvonly
R: fxr/t38
X: 1
```

Шаг 2:

Шлюз подтверждает команду и включает туда SDP с информацией о кодеке, а также о возможности использования RFC 3407:

```
200 1000 OK
I:1
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 3:

Иницирующий соединение агент вызова посылает сообщение SIP INVITE (приглашение), содержащее SDP, оконечному агенту вызова (не все подробности SIP, показанные здесь).

```
INVITE sip:bob@biloxi.example.com SIP/2.0
...
Content-Type: application/sdp
Content-Length: 167
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 4:

Оконечный агент вызова посылает в обратном направлении, к иницирующему агенту вызова, ответ SIP 200 OK (не все подробности SIP, показанные здесь).

```
SIP/2.0 200 OK
...
Content-Type: application/sdp
Content-Length: 100
v=0
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
```

Следует отметить, что оконечный агент вызова пользователя SIP не использует дескриптор возможностей RFC 3407 для отображения поддержки (или отсутствия поддержки) для T.38.

Шаг 5:

Иницирующий агент вызова получает ответ SIP 200 и посылает сообщение SIP ACK оконечному SIP UA.

Следует отметить, что агент вызова не знает, поддерживает ли одноранговый объект T.38 или нет. Для того, что бы выяснить это, агент вызова мог бы послать оконечному SIP UA запрос SIP OPTIONS, запрашивая его о возвращении его возможностей (здесь не отображено). Обратите внимание, что это может, конечно быть сделано по отношению к любому одноранговому SIP, например, если другая сторона была агентом вызова, разговаривающем на SIP, это может быть сделано также по отношению к ней.

Шаг 6:

Иницирующий агент вызова в свою очередь посылает иницилирующему шлюзу команду ModifyConnection:

```
MDCX 1001 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
M: sendrecv
v=0
o=- 25678 753849 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=audio 1296 RTP/AVP 0
```

Команда ModifyConnection не повторяет параметр LocalConnectionOptions, посланный до этого. Постольку, поскольку это касается обработки факсимильной связи, шлюз, следовательно, пытается продолжать использование текущей обработки факсимильной связи, т. е. Процедура неограниченного T.38 поддерживается всегда, следовательно, если исходящий шлюз обнаружит вызов факсимильной связи, он попытается переключиться на T.38.

Шаг 7:

Шлюз подтверждает получение команды. В этот момент устанавливается вызов при использовании кодирования PCMU и, если обнаружен вызов факсимильной связи, будет инициирована процедура контролируемого агентом вызова T.38.

Шаг 8-9:

Теперь появляется вызов факсимильной связи. Сначала посылается тональный сигнал T.30 CED (так же известный как V.25 ANS), который в данном случае просто пропускается через текущее кодирование PCMU. Поскольку согласно этой последовательности могут запускаться вызовы и факсимильной, и модемной связи, невозможно определить, что это вызов факсимильной связи до шага 11, где обнаруживается преамбула факсимильной связи V.21.

Шаг 10:

Оконечный SIP UA в действительности не поддерживает T.38, и при обнаружении вызова факсимильной связи он пытается переключиться на T.38. Как следствие, он посылает повторное приглашение (re-INVITE) иницилирующему агенту вызова с обновленным SDP, показывающее переключение на T.38.

```
INVITE sip:ca@ca-o.whatever.net SIP/2.0
...
Content-Type: application/sdp
Content-Length: 100
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udpt1 t38
```

Шаг 11:

Иницирующий агент вызова затем посылает команду ModifyConnection иницилирующему шлюзу:

```
MDCX 1003 ds/ds1-1/1@gw-o.whatever.net MGCP 1.0
C: 1
I: 1
v=0
o=- 25678 753850 IN IP4 128.96.41.2
s=-
c=IN IP4 128.96.41.2
t=0 0
m=image 1296 udpt1 t38
```

Шаг 12:

Иницирующий шлюз переходит на T.38 и посылает в обратном направлении ответ об успешном выполнении с обновленным SDP:

```
200 1003 OK
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
```

```
t=0 0
m=image 3456 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 13:

Иницирующий агент вызова посылает OK ответ SIP 200 с обновленным SDP окончному агенту пользователя SIP:

```
SIP/2.0 200 OK
...
Content-Type: application/sdp
Content-Length: 167
v=0
o=- 25678 753850 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=image 3456 udptl t38
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 image udptl t38
```

Шаг 14:

Оконечный агент пользователя SIP получает SIP 200 и отправляет SIP ACK.

Поскольку окончный агент пользователя SIP теперь обладает RemoteConnectionDescriptor с "изображением/t38" в качестве допустимых медиаданных, он может начинать обмен T.38 с иницирующим шлюзом (и наоборот).

Шаг 15-17:

Когда факсимильная связь заканчивается, окончный агент пользователя SIP принимает решение о разрыве вызова и, следовательно, посылает сообщение SIP BYE, на которое агент вызова отвечает SIP 200.

Агент вызова может теперь принять решение перейти обратно на голосовой кодек, удалить соединение, или что-то еще.

#### 4. Анализ безопасности

Пакет факсимильной связи MGCP сам по себе не представляет ничего нового касательно безопасности. Однако реализаторам следует обратить внимание на то, что медиаданные T.38 в настоящее время передаются через UDP (UDPTL) или TCP открытым текстом и без всякой защиты целостности. Если, например, имеют место услуги по безопасности для защиты медиапоточков RTP, они, следовательно, ни предоставят никакой защиты для медиапоточков T.38. Если такое отсутствие безопасности является причиной для беспокойства, параметр факсимильной связи LocalConnectionOptions, позволяющий использование T.38, в данном пакете использовать НЕ СЛЕДУЕТ, т. е. следует использовать LocalConnectionOption факсимильной связи "отключение" (или новое расширение безопасности).

#### 5. Анализ IANA

Данным образом у IANA запрашивается регистрация следующего пакета MGCP:

```
Название пакета  Имя  Версия
-----
Fax    FXR    0
```

#### 6. Нормативная литература

[RFC2119] Брэднер, С., "Ключевые слова для использования в RFCs для отображения уровней требования", BCP 14, RFC 2119, март 1997 г.

[RFC3435] Ф. Андризен, Б. Форстер, "Протокол контроля медиашлюза (MGCP) Версия 1.0", RFC 3435, январь 2003 г.

[T38] МЭС-Т Рекомендация T.38, "Процедуры для группы 3 факсимильной связи в режиме реального времени через сети IP", 03/2002.

[RFC3407] Ф. Андризен, "Протокол описания сессии (SDP) простое объявление возможностей", RFC 3407, октябрь 2002 г.

#### 7. Справочная литература

[T30] МЭС-Т Рекомендация T.30, "Процедуры для факсимильной передачи документов в телефонной сети общего пользования", 07/03.

[RFC3261] Й. Розенберг, Х. Шульцринне, Дж. Камарильо, А. Йонстон, Й. Петерсон, Р. Спаркс, М. Хэндли, Е. Шулер, "SIP: Протокол инициирования сеанса", RFC 3261, июнь 2002 г.

[SIPfax] Муль, Дж., и Дж. Ли, "Поддержка SIP для факсимильной связи в режиме реального времени: Примеры протекания вызова и лучшее практическое использование в настоящее время", текущие разработки.

## 8. Благодарности

Несколько людей внесли вклад в развитие пакета факсимильной связи MGCP. В частности, автор хотел бы поблагодарить Билла Форстера, Пауля Джонса, Гари Келли, Раджешь Кумара, Дейва Хорвица, Хироши Тамура, Роба Томпсона и комиссию CableLabs PacketCable NCS за их вклад.

## 9. Адрес автора

Flemming Andreassen  
Cisco Systems  
499 Thornall Street, 8th Floor  
Edison, NJ 08837  
Email: fandreas@cisco.com

Положение об интеллектуальной собственности

IETF не занимает какой бы то ни было позиции относительно обоснованности или применимости прав интеллектуальной собственности или других прав, которые могут быть заявлены, относящихся к реализации или использованию технологии, описанной в данном документе или степень, в которой лицензия, попадающая под эти права, может быть доступной или недоступной; также это не означает, что она делала независимые попытки идентифицировать такие права. Информация о процедурах IETF с учетом прав, указанных в документах IETF может быть найдена в BCP 78 и BCP 79.

Копии объявлений IPR, сделанных для секретариата IETF, и любые удостоверения о наличии лицензий, или результат попытки, сделанной реализаторами или пользователями данных указаний, для того, чтобы получить генеральную лицензию или разрешение на использование таких прав собственности, может быть получено в работающем в режиме on-line IPR репозитории IETF по электронному адресу: <http://www.ietf.org/ipr>.

IETF приглашает любую заинтересованную сторону довести до его сведения информацию о любых авторских правах, патентах, заявках на патенты или других правах собственности, включающих в себя технологию, которая может потребоваться при реализации данного стандарта. За информацией обращайтесь в IETF по электронному адресу: [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

Отсутствие ответственности, связанной с нарушением авторских прав

Данный документ и включенная в него информация, предоставлена на основе "как есть" и вкладчик, ОРГАНИЗАЦИЯ, КОТОРУЮ ОН/ОНА ПРЕДСТАВЛЯЮТ ИЛИ КОТОРОЙ СПОНСИРУЮТСЯ (ЕСЛИ ИМЕЕТСЯ), ИНТЕРНЕТ СООБЩЕСТВО И КОМИТЕТ ПО ИНЖЕНЕРНЫМ ПРОБЛЕМАМ ИНТЕРНЕТА, ОБЪЯВЛЯЮТ ВСЕ ГАРАНТИИ, ВЫРАЖЕННЫЕ ИЛИ ПРЕДПОЛАГАЕМЫЕ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯ КАКУЮ-ЛИБО ГАРАНТИЮ, ЧТО ИСПОЛЬЗОВАНИЕ ЗДЕСЬ ИНФОРМАЦИИ НЕ НАРУШИТ НИКАКИХ ПРАВ ИЛИ ЗНАЧАЩИХ ГАРАНТИЙ ТОВАРНОЙ ПРИГОДНОСТИ ИЛИ ГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ.

Положение об авторских правах

Авторские права (С) интернет сообщества (2004). Данный документ является субъектом прав, лицензий и ограничений, содержащихся в BCP 78 и, за исключением того, что установлено здесь далее, авторы сохраняют все свои права.

Благодарности

Основания для функции редактора RFC в настоящее время предоставлена интернет сообществом.

### A.3 Пакет метрик передачи голоса через IP (VoIP)

Пакет метрик VoIP определяет механизм контроля сбора и сообщения метрик VoIP. МТА ДОЛЖЕН поддерживать сообщение о параметрах метрик VoIP, как указано в данном пакете для всех обязательных параметров расширенного отчета в МСЭ-Т Рек. J.161. Реализации ДОЛЖНЫ соответствовать кодировке версии протокола NCS, как указано в 7.2.1.4.

Комитет по инженерным проблемам интернета (IETF)

Д. Ауэрбах  
Cisco Systems  
Д. Хэнкок  
CableLabs  
Б. Хаар  
Arris  
Р. Кумар

Проект интернет-документа

Системы Cisco

Документ: .txt

1 октября 2004 г.

Категория: Информационный

#### **Пакет метрик VoIP RTSP-XR протокола контроля медиашлюза**

Статус данной заметки

Принимая данный Проект интернет-документа, мы удостоверяем, что о любом прилагаемом патенте или других правах интеллектуальной собственности (IPR), о которых мы знаем, здесь объявлено, а о тех, о которых мы будем узнавать, будет объявлено в соответствии с RFC 3668.

Интернет-проект - это рабочая документация комитета по инженерным проблемам интернета (IETF), его областей деятельности и его рабочих групп. Следует отметить, что другие группы могут также распространять свои рабочие документации в качестве Интернет-проектов.

Интернет-проект - это черновая документация которая действительна в течение максимум 6 месяцев и в любой момент времени может быть обновлена, замещена или сделана недействительной другой документацией. Использовать Интернет-проект в качестве справочной литературы нецелесообразно, так же как и цитировать его иначе, чем в качестве "документации на стадии разработки".

Со списком текущих Интернет-проектов можно ознакомиться на: <http://www.ietf.org/ietf/lid-abstracts.txt>.

Со списком теневого директорий Интернет-проектов можно ознакомиться на <http://www.ietf.org/shadow.html>.

Срок действия данного Проект интернета истекает 1 марта 2005 г.

#### **Реферат**

Данный документ определяет пакет протокола контроля медиашлюза для контроля сбора метрик VoIP, поддерживаемых расширенным отчетом RTSP, как указано в RFC 3611. Пакет позволяет агенту вызова активировать или отключать сбор данных метрик. Он также позволяет агенту вызова контролировать будет ли шлюз отвечать на удаленные запросы от одностороннего шлюза о сборе метрики.

#### **Соглашения, используемые в данном документе**

Ключевые слова "ДОЛЖЕН", "НЕ ДОЛЖЕН", "ТРЕБУЕТСЯ", "БУДУТ", "НЕ БУДУТ", "СЛЕДУЕТ", "НЕ СЛЕДУЕТ", "РЕКОМЕНДУЕТСЯ", "МОЖЕТ", и "ПРИ НЕОБХОДИМОСТИ" в данном документе должны интерпретироваться как описано в ВСП 14, RFC-2119 [RFC2119].

1. Введение
2. Определение пакета метрик VoIP
  - 2.1 LocalConnectionOptions
    - 2.1.1 Процедуры конечной точки
      - 2.1.1.1 Процедура отключения
      - 2.1.1.2 Процедура включения
      - 2.1.1.3 Процедура согласования
    - 2.2 Процедуры AuditConnection
    - 2.3 Новые параметры расширения MGCP
      - 2.3.1 Значения параметра расширения
      - 2.3.2 Грамматика параметра расширения
    - 2.4 Согласование метрик VoIP
    - 2.5 Анализ реализации
      - 2.5.1 Анализ совместной работы
  3. Примеры протекания вызова
    - 3.1 DLCX, инициированный агентом вызова
    - 3.2 DLCX, инициированный шлюзом
  4. Анализ безопасности
  5. Анализ IANA
  6. Нормативная литература
  7. Благодарности

## 1. Введение

Данный документ определяет пакет протокола контроля медиашлюза (MGCP) [RFC3435], который делает возможным сбор и сообщение метрик, измеряющих качество голосового трафика в соединениях VoIP. Эта возможность базируется на блоке метрик VoIP расширенного отчета RTCP и использует его, как указано в RFC 3611.

В дополнение к параметрам в блоке метрик VoIP RTCP-XR, данный документ включает в себя интервал между прибытиями пакетов, как определено в RFC 3550. Целью такого включения является согласование сообщения метрик VoIP на базе MGCP с SIP. Следует отметить, что структура ConnectionParameters MGCP [RFC3435] позволяет осуществлять отчет локальной, но не удаленной, версии данного параметрас.

Данный пакет делает различие между сбором, ответом и отчетом. Термин "сбор" относится к процессу расчета и накопления локальных данных метрик VoIP и накопления удаленных данных метрик VoIP, полученных в пакетах RTCP XR от удаленного устройства. Термин "ответ" означает отправление накопленных локальных данных метрик в пакетах RTCP XR удаленному соединенному устройству, как указано в RFC 3611. Термин "отчет" относится к процессу сообщения накопленных локальных и удаленных данных метрик VoIP по соединению агенту вызова в последовательности либо команды DeleteConnection, либо команды AuditConnection.

Агент вызова может попросить локальную конечную точку активизировать или отключить отчет о метриках VoIP для соединения в команде CreateConnection или ModifyConnection. Когда отчет о метриках VoIP активизирован, локальная конечная точка будет использовать сигнализацию SDP, как указано в RFC 3611, чтобы попросить удаленную конечную точку начать передачу данных метрик VoIP RTCP XR для удаленного конца соединения. Локальная конечная точка накопит локальную и удаленную копию данных метрик VoIP, о которых должно быть сообщено агенту вызова, когда соединение удаляется или подвергается проверке (и запрашиваются данные метрик).

Конечная точка будет собирать местные данные метрик VoIP по двум отдельным причинам; если отчет о метриках VoIP активизируется агентом вызова через команду соединения, или если сбор метрик VoIP запрашивается удаленной конечной точкой через сигнализацию. Агент вызова может препятствовать сбору и ответу удаленной конечной точки посредством специального отключения функции сбора/ответа. Данные метрик VoIP, о которых сообщается агенту вызова, когда соединение отключается, представляют собой данные, накопленные с того момента, как конечная точка начала сбор непосредственно на соединении, в независимости от того, был ли запрос инициирован запросом агента вызова или сигнализацией SDP.

Данный пакет разработан для обеспечения точных измерений качества голоса для базовых двухсторонних соединений вызова, где метрики накапливаются в течение времени существования соединения, и агенту вызова сообщается о них, когда соединение удаляется. Пакет не делает никаких специальных подготовительных мероприятий для событий реконфигурации пути передачи во время вызова, которые могут оказать влияние на качество голоса. Например, если команда ModifyConnection изменяет удаленную конечную точку соединения, единственный отчет о метриках, сгенерированный когда в дальнейшем соединение будет удалено, не предоставило бы каких-либо средств для сравнения метрик до и после изменения удаленной конечной точки.

Следует отметить, что данный пакет не влияет на предоставление отчетов о других статистических данных MGCP, как определено в базовых протоколах.

Определение пакета метрик VoIP предоставлено в разделе 2, а разделе 3 отображены два примера протекания вызова, показывающие как его использовать. Анализ безопасности можно находится в разделе 4, за которым следует анализ IANA и ссылки.

## 2. Определение пакета метрик VoIP

Пакет определяется для метрик VoIP. Пакет определяет новые опции местного соединения (localconnectionoptions) и параметры соединения так, как указано ниже.

Имя пакета: XRM

Версия пакета: 0

### 2.1 LocalConnectionOptions

Данный пакет определяет новый параметр LocalConnectionOptions (LCO) для активизации и отключения в шлюзе предоставления отчета о метриках VoIP. Данные параметры являются необязательными со значением по умолчанию "согласовать". Агент вызова предоставляет данные параметры LCO для инструктирования конечной точки активизировать или отключить сбор и предоставление отчета о метриках RTCP-XR для целевого соединения. Новым параметром является параметр сбора метрик (и) предоставления отчета и кодируется он как "mcr".

Параметр может иметь одно из следующих трех значений:

- выключено – отключение сбора, предоставления отчета и ответа метрик VoIP (кодируется как "off").
- включено – активизация сбора, предоставления отчета и ответа метрик VoIP (кодируется как "on").
- согласование – позволяет сбор и ответ метрик VoIP, если запрошено, но отключает предоставление отчета VoIP (кодируется как "negotiate").

Следующий пример показывает как использовать данный параметр для включения сбора, предоставления отчета и ответа метрик VoIP:

```
L: xrm/mcr:on
```

В командах `CreateConnection` значение `xrm/mcr LCO` по умолчанию "согласовать". В командах `ModifyConnection` значение `xrm/mcr LCO` по умолчанию установлен на его текущем значении для соединения. Таким образом, если `LocalConnectionOptions` либо пропущен, либо `xrm/mcr LCO` не включен в команду `ModifyConnection`, сохраняется предыдущее значение `xrm/mcr LCO` для соединения.

Следует обратить внимание, что данный LCO прямо контролирует предоставление отчета метрик XR агенту вызова и может контролировать сбор метрик. Сбор метрик также зависит от возможного запроса от однорангового шлюза, как отображено в `RemoteConnectionDescriptor`.

#### 2.1.1.1 Процедуры конечной точки

Данный раздел описывает процедуры конечной точки для каждого из значений параметра `xrm/mcr`.

##### 2.1.1.1.1 Процедура отключения

При получении данного значения конечная точка ДОЛЖНА стирать любые ранее собранные для соединения локальные или удаленные данные метрик VoIP, а затем ДОЛЖНА инструктировать удаленное устройство об остановке отправления блоков отчета метрик VoIP RTCP XR. Конечная точка ДОЛЖНА игнорировать запрос SDP для метрик и НЕ ДОЛЖНА передавать локальных метрик одноранговой конечной точке. Локальная конечная точка НЕ ДОЛЖНА передавать отчет о метриках XR при получении или инициировании последовательности `DeleteConnection` и НЕ ДОЛЖНА передавать отчет о метриках XR при ответе на команду `AuditConnection`.

##### 2.1.1.1.2 Процедура включения

При получении данного значения, если она еще не делает этого, то конечная точка ДОЛЖНА начать подсчет локальных данных метрик VoIP для соединения, и ДОЛЖНА проинструктировать удаленное устройство о начале отправления данных метрик VoIP для соединения посредством включения атрибута `"rtcp-xr"` со значением `"voip-metrics"` в SDP.

Конечная точка ДОЛЖНА предоставлять отчет о локальных и удаленных метриках VoIP в ответе на команду `DeleteConnection`, в ответе на команду `AuditConnection`, когда проверяются параметры измерения, и если посылается команда `DeleteConnection`. Конечная точка ДОЛЖНА опускать удаленные метрики в отчете, если от удаленной конечной точки не было получено никаких удаленных метрик.

##### 2.1.1.1.3 Процедура согласования

При получении данного значения конечная точка не делает ничего, за исключением установки внешнего флага для отображения того, что запрос SDP для отправления локальных метрик удаленной одноранговой конечной точке разрешен. Локальная конечная точка не посылает данной информации до тех пор, пока ее не запросит об этом удаленная конечная точка.

## 2.2 Процедуры `AuditConnection`

Данный пакет определяет использование двух новых значений для команды `AuditConnection` (AUCX). Это позволяет агенту вызова явно запрашивать, чтобы конечная точка возвратила текущие значения либо локально, либо удаленной метрик. Запрос не сбрасывает значений. Если один или оба этих значений информации запроса включены в AUCX, конечная точка ДОЛЖНА возвращать соответствующие значения метрик, если они имеются в наличии. Двумя новыми значениями информации запроса:

XRM/LVM	Возвращает локальные значения голосовых метрик.
XRM/RVM	Возвращает удаленные значения голосовых метрик.

Если конечная точка спрашивается о параметре, который она не понимает, конечная точка НЕ ДОЛЖНА генерировать ошибку; вместо этого параметр ДОЛЖЕН быть опущен в ответе на команду проверки.

Если конечная точка спрашивается о параметре, который она поддерживает, но значения которого у нее нет, то конечная точка НЕ ДОЛЖНА генерировать ошибку; вместо этого параметр ДОЛЖЕН быть включен в ответ на команду проверки с пустым значением параметра.

### 2.3 Новые параметры расширения MGCP

Два новых параметра MGCP определены для поддержки предоставления отчетов о метриках VoIP; один для предоставления отчетов о локальных данных метрик и один для предоставления отчетов об удаленных данных метрик. Они возвращаются в последовательности DeleteConnection и могут быть возвращены ответом на AuditConnection (если запрашивалась проверка). Локальные метрики включаются в строку "XRM/LVM". Удаленные метрики включаются в строку XRM/RVM.

В качестве примера для шлюза, записывающего локальные метрики и получившего удаленные метрики, DLX приведет к следующим дополнительным строкам, возвращенным в ответе:

```
XRM/LVM: PLC=1, JBA=2, JBR=7 ...
XRM/RVM: PLC=0, JBA=0, JBR=3 ...
```

Для подробной семантики данных параметров, смотрите описание блоков отчета метрик VoIP в RFC 3611. Например, значение "standard" (стандартная) в отличие от "enhanced" (улучшенная) защита от потери пакетов должно быть получено из RFC 3611.

#### 2.3.1 Значения параметра расширения

В отличие от RTSP, поля параметров в MGCP выражены чаще в текстовом, чем в фиксированном двоичном формате. Следовательно, не является необходимым оговаривать верхние границы для числовых параметров в MGCP. Однако верхние границы указанные в RFC 3611 все равно применяются. Обратите внимание, что верхние границы в RFC 3611 иногда диктуются чаще закономерностями в размере поля (например, инкремент 8), чем действительным диапазоном значений параметра. Например, хотя в 16-битном поле может быть выражено числовое значение 65,535 мс, жизнеспособное соединение RTP никогда не получит время задержки обхода с данным значением.

Имя параметра расширения	Код	Значение
Процент потери пакетов в сети	NLR	Доля пакетов, потерянных с момента начала передачи, выраженная в виде 8-битной двоичной дроби, полученной делением числа потерянных в процессе передачи пакетов к общему числу пакетов, ожидаемых получателем, умножая полученное значение на 256 и беря только целую часть результата.
Процент отклонения буфером устранения дрожаний	JDR	Доля пакетов, отклоненных получающим буфером устранения дрожаний с начала передачи, выраженная в виде 8-битной двоичной дроби, полученной делением числа отклоненных буфером устранения дрожаний к общему числу пакетов, ожидаемых получателем, умножая полученное значение на 256 и беря только целую часть результата.
Плотность потерь пачки	BLD	Средняя доля потерянных и отклоненных пакетов, возникающих за время периодов пачки потерянных импульсов, выраженная в виде 8-битной двоичной дроби, полученной делением суммы числа пакетов, считающихся потерянными за время периодов пачки потерянных импульсов, и числа пакетов, отклоненных буфером устранения дрожаний за время периодов пачки потерянных импульсов, к общему числу пакетов, ожидаемых получателем, умножая полученное значение на 256 и беря только целую часть результата.
Плотность потерь зазора	GLD	Средняя доля потерянных и отклоненных пакетов, возникающих за время периодов зазора потери импульсов, выраженная в виде 8-битной двоичной дроби, полученной делением суммы числа пакетов, считающихся потерянными за время периодов зазора потери импульсов, и числа пакетов, отклоненных буфером устранения дрожаний за время периодов зазора потери импульсов, к общему числу пакетов, ожидаемых получателем, умножая полученное значение на 256 и беря только целую часть результата.
Длительность пачки потерянных импульсов	BD	Средняя длительность периодов пачки потерянных импульсов, в миллисекундах.
Длительность зазора потери импульсов	GD	Средняя длительность периодов зазора потери импульсов, в миллисекундах.
Задержка полного обхода сети	RTD	Задержка полного обхода, в миллисекундах, между интерфейсами RTP локальной и удаленной конечных точек вызова. Корректные значения ДОЛЖНЫ быть больше или равны 0 миллисекундам.

Имя параметра расширения	Код	Значение
Задержка конечной системы	ESD	Задержка конечной системы (конечной точки), в миллисекундах, включающая задержку кодировки, декодировки и буфера устранения дрожания. Корректные значения ДОЛЖНЫ быть больше или равны 0 миллисекундам.
Уровень сигнала	SL	Отношение среднего уровня сигнала к началу отсчета в 0 дБм, выраженное в дБ. Данное число выражено в виде целого числа с необязательным знаком отрицания ("-"). Если знак опущен, то имеется в виду положительное значение. В общем случае типовым значениям следует находиться между -15 и -20.
Уровень шумовых помех	NL	Отношение периода тишины средних фоновых шумов к началу отсчета в 0 дБм в дБ.
Остаточные потери эховозвращения	ERL	Потери возвращения эха после эффекта компенсации эха в дБ. Необходимости в знаке нет, так как корректные значения МОГУТ быть меньше или равны 0 дБ.
Минимальный порог зазора потери импульсов	GMN	Порог передачи зазора/пачки потери импульсов; рекомендуемое значение 16. Разрешены значения в диапазоне 1 - 255.
Фактор R	NSR	Значение, представляющее собой качество получаемого конечного вызова для потока пакетов RTP, исходящего из докладывающей конечной точки; рассчитывается, как указано в МСЭ-Т Рек. G.107. Корректные значения больше 0 и меньше или равны 100.
Внешний фактор R	XSR	Значение, представляющее собой влияние любого сегмента вызова, перенесенного через внешний сегмент сети к тому сегменту, куда подключена конечная точка; рассчитывается, как указано в МСЭ-Т Рек. G.107. Корректные значения больше 0 и меньше или равны 100.
Предполагаемое MOS-LQ	MLQ	Предполагаемое качество слышания MOS для принимающей стороны. Номинальный диапазон оценки MOS составляет 0-5. Перед его выражением через MGCP, оценки MOS умножается на 10, а дробная часть отсекается. Результатом является целое число с корректными значениями в диапазоне 10 -50.
Предполагаемое MOS-CQ	MCQ	Предполагаемое качество речи MOS для принимающей стороны. Номинальный диапазон оценки MOS составляет 0-5. Перед его выражением через MGCP, оценки MOS умножается на 10, а дробная часть отсекается. Результатом является целое число с корректными значениями в диапазоне 10 -50.
Тип защиты от потери пакетов	PLC	Тип используемого алгоритма защиты от потери пакетов; должно принимать одно из следующих перечисленных значений: 0 - неуказанный 1 - отключенный 2 - улучшенный 3 - стандартный
Адаптивный буфер устранения дрожаний	JVA	Адаптируемость буфера устранения дрожаний. Должно принимать одно из следующих перечисленных значений: 0 - неизвестный 1 - зарезервированный 2 - неадаптивный 3 - адаптивный
Скорость буфера устранения дрожаний	JBR	Скорость настройки буфера устранения дрожаний. Значение между 0 и 15.
Номинальная задержка буфера устранения дрожаний	JBN	Текущая номинальная задержка в миллисекундах, соответствующая номинальной задержке буфера устранения дрожаний для прибывших точно во время пакетов.
Максимальная задержка буфера устранения дрожаний	JBM	Текущая максимальная задержка в миллисекундах, соответствующая прибывшему раньше всех пакету, который не был отклонен. В простых очередных реализациях, она может соответствовать номинальной задержке буфера устранения дрожаний. В реализациях адаптивного буфера устранения дрожаний, может динамически варьироваться до величины абсолютного максимума задержки буфера устранения дрожаний (см. ниже).

Имя параметра расширения	Код	Значение
Абсолютный максимум задержки буфера устранения дрожаний	JBS	Абсолютный максимум задержки в миллисекундах, которого может достичь адаптивный буфера устранения дрожаний в наиболее неблагоприятных условиях. Для фиксированных буфера устранения дрожаний, оно может быть установлено на величину максимальной задержки буфера устранения дрожаний.
Дрожание прибытия пакетов	IAJ	Используя определение из RFC 3550, дрожание прибытия пакетов определено для создания средства измерения отклонения (сглаженное абсолютное значение) различий в пакетах, размещенных у получателя, по сравнению с отправителем для пары пакетов. Данная метрика предоставляется в миллисекундах.

### 2.3.2 Грамматика параметров расширения

Далее описано ABNF кодирование для локальных и удаленных параметров метрик.

```
LocalVoIPMetrics = "XRM/LVM:" 0*WSP [VoIPMetric 0*("," 0*WSP VoIPMetric)]
```

```
RemoteVoIPMetrics = "XRM/RVM:" 0*WSP [VoIPMetric 0*("," 0*WSP VoIPMetric)]
```

```
VoIPMetric = NetworkPacketLossRate
            / JitterBufferDiscardRate
            / BurstLossDensity
            / GapLossDensity
            / BurstDuration
            / GapDuration
            / RoundTripNetworkDelay
            / EndSystemDelay
            / SignalLevel
            / NoiseLevel
            / ResidualEchoReturnLoss
            / MinimumGapThreshold
            / RFactor
            / ExternalRFactor
            / EstimatedMOS-LQ
            / EstimatedMOS-CQ
            / PacketLossConcealmentType
            / JitterBufferAdaptive
            / JitterBufferRate
            / JitterBufferNominal
            / JitterBufferMax
            / JitterBufferAbsMax
            / InterArrivalJitter
```

```
NetworkPacketLossRate = "NLR=" 1*3 (DIGIT) ;0-255
```

```
JitterBufferDiscardRate = "JDR=" 1*3 (DIGIT) ;0-255
```

```
BurstLossDensity = "BLD=" 1*3 (DIGIT) ; 0-255
```

```
GapLossDensity = "GLD=" 1*3 (DIGIT) ; 0-255
```

```
BurstDuration = "BD=" 1*5 (DIGIT) ; 0-65535
```

```
GapDuration = "GD=" 1*5 (DIGIT) ; 0-65535
```

```
MinimumGapThreshold = "GMN=" 1*3 (DIGIT) ; 1-255
```

```
RoundTripNetworkDelay = "RTD=" 1*5 (DIGIT) ;0-65535
```

```
EndSystemDelay = "ESD=" 1*5 (DIGIT) ;0-65535
```

```
SignalLevel = "SL=" ["-"] 1*3 (DIGIT) ; -128 to 127
```

```
NoiseLevel = "NL=" ["-"] 1*3 (DIGIT) ; -128 to 127
```

```
ResidualEchoReturnLoss = "RERL=" 1*3 (DIGIT) ;0-127
```

```
RFactor = "RF=" 1*3 (DIGIT) ;0-100, or 127
```

ExternalRFactor	= "XRF=" 1*3 (DIGIT) ;0-100, or 127
EstimatedMOS-LQ	= "MLQ=" 1*3 (DIGIT) ; 10-50, or 127
EstimatedMOS-CQ	= "MCQ=" 1*3 (DIGIT) ; 10-50, or 127
PacketLossConcealmentType	= "PLC=" ("0" / "1" / "2" / "3")
JitterBufferAdaptive	= "JBA=" ("0" / "1" / "2" / "3")
JitterBufferRate	= "JBR=" 1*2 (DIGIT) ;0-15
JitterBufferNominal	= "JBN=" 1*5 (DIGIT) ;0-65535
JitterBufferMax	= "JBM=" 1*5 (DIGIT) ;0-65535
JitterBufferAbsMax	= "JBS=" 1*5 (DIGIT) ;0-65535
InterArrivalJitter	= "IAJ=" 1*5 (DIGIT) ;0-65535

#### 2.4 Согласование метрик VoIP

RFC 3611 добавляет новый атрибут, называемый "rtcp-xr", который может использоваться локальной конечной точкой соединения для запроса удаленной конечной точки об отправке отчетов RTCP XR для ее локально собранных метрических данных. Данный атрибут кодируется следующим образом:

```
rtcp-xr:<xr-format>
```

Данный атрибут может использоваться как атрибут либо сеансового уровня, либо уровня медиаданных. В PacketCable обязательна только опция xr-формата метрик VoIP. Другие опции МОГУТ предоставляться. Если предоставляются несколько опций, они ДОЛЖНЫ быть разделены пробелом. За более подробной информацией об использовании данного параметра атрибута обращаться к RFC 3611.

Отправка: Параметр rtcp-xr:voip-metrics ДОЛЖЕН быть послан в соответствии с RFC 3611, если "on" с помощью Local Connection Options. При наличии параметр МОЖЕТ выступать как атрибут либо сеансового уровня, либо уровня медиаданных. Данный параметр атрибута НЕ ДОЛЖЕН посылаться, если voip-metrics сконфигурированы на "off" или "negotiate". Конечная точка МОЖЕТ послать другие опции xr-формата, независимые от метрик VoIP, которые определены в RFC 3611.

Получение: Когда через данный параметр атрибута получены метрики VoIP xr-формата, конечная точка ДОЛЖНА включать в себя расширенные отчеты о метриках VoIP для удаленной стороны через RTCP, как определено в RFC 3611, если только агент вызова не проинструктировал явно конечную точку не делать этого с помощью конфигурирования Local Connection Option метрик VoIP на "off". Если параметр атрибута не получен, или получен, но без опций метрик VoIP или вообще пустой, т. е. xr-формата не предоставлено, то конечная точка НЕ ДОЛЖНА включать расширенные отчеты о метриках VoIP для удаленной стороны через RTCP. Другие полученные опции xr-формата, определенные в RFC3611, МОЖНО игнорировать.

#### 2.5 Анализ реализации

Как объяснялось во введении, конечная точка или шлюз должны определять различия между сбором метрик качества голоса RTCP-XR при ответе на запросы однорангового объекта о метриках и сообщении отчетов о метриках локальному агенту вызова. Сбор метрик может быть инициирован с помощью одной или обеих следующих возможностей:

Удаленный запрос от однорангового шлюза (через SDP)

Удаленный запрос от контролирующего агента вызова (через команду CreateConnection или ModifyConnection)

Шлюз ДОЛЖЕН сообщать о метриках при выполнении последовательности DeleteConnection или AuditConnection (где проверяются параметры метрик VoIP), если полученные последними команды CreateConnection или ModifyConnection активизировали предоставление отчета о метриках (xrm/mcr: on).

Если метрика VoIP не была рассчитана локально, она ДОЛЖНА быть пропущена в строке XRM/LVM: . Однако конечная точка, поддерживающая пакет XRM ДОЛЖНА рассчитывать и сообщать, по крайней мере, одну метрику VoIP из вышеприведенной таблицы.

Если удаленная сторона показывает, отправляя пустое значение, что конкретный параметр RTCP-XR недоступен, то о пустом значении ДОЛЖНО быть сообщено, или данный параметр ДОЛЖЕН быть опущен в строке XRM/RVM: . Если удаленная сторона не предоставляет никаких удаленных данных метрик VoIP, то ответ шлюза на запрос о данных метрик основывается на следующем инициировании команды:

- DeleteConnection: параметр XRM/RVM: ДОЛЖЕН БЫТЬ опущен,
- AuditConnection: параметр XRM/RVM: ДОЛЖЕН БЫТЬ включен в виде пустого параметра.

### 3. Примеры протекания вызова

Приведенные примеры являются частями протекания вызова для локальной части вызова, которая задействует свойство RTCP-XR.

#### 3.1 DLCX, инициированный агентом вызова

Шаг 1:

Агент вызова выдает шлюзу команду CreateConnection с инструкциями использовать кодирование медиаданных PCMU и активизировать сбор, предоставление отчета и ответа о метриках RTCP-XR:

```
CRCX 1000 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
L: a:PCMU, xrm/mcr: on
M: recvonly
```

Шаг 2:

Шлюз подтверждает получение команды и включает туда атрибут rtxp-xr, установленный на значение "voip-metrics":

```
200 1000 OK
I:1
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=rtcp-xr:voip-metrics
```

Шаг 3:

Шлюз получает команду ModifyConnection, содержащую LCO без параметра xrm/mcr, что означает, что конечной точке следует продолжать использовать полученное до этого значение "on". Также, полученный SDP не содержит параметра rtxp-xr, что показывает, что удаленная конечная точка не хочет получать отчеты RTCP XR от данной конечной точки. Обратите внимание, что отсутствие атрибута rtxp-xr в полученном SDP не сигнализирует о желании или способности конечной точки соответствовать предыдущему запросу rtxp-xr "voip-metrics", который он получил от данной конечной точки.

```
MDCX 1001 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
L: a:PCMU,G728
M: recvonly
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 8
```

Шаг 4:

Шлюз подтверждает команду и, поскольку сбор/предоставление отчета о метриках VoIP по-прежнему на значении "on", включает атрибут rtxp-xr, установленный на значение "voip-metrics":

```
200 1001 OK
I:1
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 8
a=rtcp-xr:voip-metrics
```

Шаг 5:

Агент вызова, возможно, выдает команду DeleteConnection:

```
DLCX 1100 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
I: 1
```

Шаг 6:

Шлюз подтверждает команду и возвращает стандартные параметры соединения MGCP вместе с параметрами RTCP-XR (обратите внимание, что новые строки показаны здесь только по причинам форматирования документа):

```
250 1100 OK
P: PS=1000, OS=60000, PR=800, OR=50000, PL=10, JI=27, LA=48,
XRM/LVM: NLR=128, JDR=40, BLD=33, GLD=10, BD=55, GD=45, RTD=1000,
ESD=1000, SL=-10, NL=2, RERL=23, GMN=80, RF=19, XRF=19, MLQ=19,
MCQ=22, PLC=1, JBA=2, JBR=8, JBN=40, JBM=80, JBS=120, IAJ=10
XRM/RVM: NLR=128, JDR=40, BLD=33, GLD=10, BD=55, GD=45, RTD=1000,
ESD=1000, SL=-10, NL=2, RERL=23, GMN=80, RF=19, XRF=19, MLQ=19,
MCQ=22, PLC=1, JBA=0, JBR=8, JBN=30, JBM=60, JBS=100, IAJ=15
```

### 3.2 DLCX, инициированный шлюзом

Шаг 1:

Агент вызова выдает шлюзу команду CreateConnection с инструкциями использовать кодирование медиаданных PCMU и активизировать сбор и предоставление отчета о метриках RTCP-XR:

```
CRCX 1000 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
L: a:PCMU, xrm/mcr: on
M: recvonly

CRCX 1000 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
L: a:PCMU, xrm/mcr: on
M: recvonly
```

Шаг 2:

Шлюз подтверждает получение команды и включает туда атрибут `rtcp-xr`, установленный на значение `"voip-metrics"`:

```
200 1000 OK
I:1
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=rtcp-xr:voip-metrics
```

Шаг 3:

Шлюз выдает команду DeleteConnection, включающую параметры соединения и метрики RTCP:

```
DLCX 1100 ds/ds1-1/1@gw-o.example.net MGCP 1.0
C: 1
I: 1
E: 900 - Hardware error
P: PS=1000, OS=60000, PR=800, OR=50000, PL=10, JI=27, LA=48,
XRM/LVM: NLR=128, JDR=40, BLD=33, GLD=10, BD=55, GD=45, RTD=1000,
ESD=1000, SL=-10, NL=2, RERL=23, GMN=80, RF=19, XRF=19, MLQ=19,
MCQ=22, PLC=1, JBA=2, JBR=8, JBN=40, JBM=80, JBS=120
XRM/RVM: NLR=128, JDR=40, BLD=33, GLD=10, BD=55, GD=45, RTD=1000,
ESD=1000, SL=-10, NL=2, RERL=23, GMN=80, RF=19, XRF=19, MLQ=19,
MCQ=22, PLC=1, JBA=0, JBR=8, JBN=30, JBM=60, JBS=100
```

Обратите внимание, что параметр IAJ не включен в строки XRM/LVM и XRM/RVM. Это всего лишь пример пропуска разрешенных параметров. Данный пример может быть прекрасно выполнен с помощью параметра IAJ.

Шаг 4:

Агент вызова подтверждает команду:

```
200 1100 OK
```

#### 4. Анализ безопасности

Данный пакет наследует анализ безопасности на базе протокола MGCP. Возможные новые проблемы безопасности затрагивают локальную конечную точку, на которую удаленная точка взваливает дополнительную задачу по запросу сбора метрик на этой локальной конечной точке. Эта угроза может быть сдержана агентом вызова целенаправленным запрещением генерирования локальных метрик с использованием опции локального соединения: `xrm/mcr: off`.

#### 5. Анализ IANA

Данным образом у IANA запрашивается регистрация следующего пакета MGCP:

Название пакета	Имя	Версия
RTCP-XR metrics	XRM	0

#### 6. Нормативная литература

[RFC2119] Брэднер, С., "Ключевые слова для использования в RFCs для отображения уровней требования", BCP 14, RFC 2119, март 1997 г.

[RFC3435] Ф. Андризен, Б. Форстер, "Протокол контроля медиашлюза (MGCP) Версия 1.0", RFC 3435, январь 2003 г.

[RFC3611] Т. Фридман, Р. Касерес, А. Кларк, "Расширенные отчеты протокола контроля RTP (RTCP XR)", RFC 3611, ноябрь 2003 г.

#### 7. Благодарности

Вклад в разработку этого документа внесли Флемминг Андризен, Дэвид Энсайн и группа по разработке указаний PacketCable NCS.

## Приложение В

### Динамическое качество обслуживания

В настоящем Приложении содержится дополнительная подробная информация об использовании динамического качества обслуживания (D-QoS) в сигнализации NCS. Более подробно описывается ожидаемое поведение адаптера МТА, включая конечный автомат и псевдокод, который ДОЛЖЕН использовать адаптер МТА для поддержки описываемого режима D-QoS. Для получения дополнительной информации следует обращаться к Рекомендации МСЭ-Т J.163.

#### Введение

Адаптеры МТА, осуществляющие поддержку динамического качества обслуживания, должны хранить и поддерживать состояние D-QoS для каждого соединения. Каждый раз, когда D-QoS используется для соединения, конечная точка будет хранить следующую информацию D-QoS, связанную с соединением, до тех пор, пока оно не будет исключено:

- **GateID** (идентификатор логического вентиля) – Текущий GateID, используемый для соединения.
- **ResourceID** (идентификатор ресурса) – Текущий ResourceID, используемый для соединения.
- **Last reservation** (последнее резервирование ресурсов) – Параметры для самого последнего резервирования ресурсов для соединения. Сюда входят классификаторы, а также параметры среды передачи, как в направлении передачи, так и в направлении приема.
- **Last commit** (последнее предоставление ресурсов) – Параметры для самого последнего предоставления ресурсов для соединения. Сюда входят классификаторы, а также параметры среды передачи, как в направлении передачи, так и в направлении приема.
- **Reserve Destination** (адресат резерва) – IP-адрес и порт, которые могут быть использованы для обеспечения возможности резервирования ресурсов, когда, как поясняется ниже, еще не известна удаленная адресная информация.

Идентификатор логического вентиля (GateID) является ключом к резервированию ресурсов. Если GateID был обеспечен для соединения, то для этого соединения создается конечный автомат D-QoS, а вся упомянутая выше информация будет поддерживаться для этого соединения до тех пор, пока оно не будет исключено.

Ресурсы могут быть зарезервированы и предоставлены адаптером МТА независимо как в направлении передачи, так и в направлении приема. IP-адрес адресата и порт-адресат, а также IP-адрес источника берутся из параметра RemoteConnectionDescriptor, если этот параметр был представлен. В таком случае адаптер МТА ДОЛЖЕН использовать для резервирования и предоставления ресурсов следующие классификаторы:

	МТА-о (DOCSIS)
<b>Нисходящее направление/прием</b>	
IP-адрес источника	IP(SDP-t)
Порт источника	*
IP-адрес адресата	IP(SDP-o)
Порт-адресат	Порт (SDP-o)
<b>Восходящее направление/передача</b>	
IP-адрес источника	IP(SDP-o)
Порт источника	Порт(o)
IP-адрес адресата	IP(SDP-t)
Порт-адресат	Порт(SDP-t)

где:

- **IP(SDP-o)** относится к IP-адресу для мультимедийных пакетов в параметре LocalConnectionDescriptor адаптера МТА (исходящего);
- **IP(SDP-t)** относится к IP-адресу для мультимедийных пакетов в параметре RemoteConnectionDescriptor адаптера МТА (исходящего);

- **Порт(SDP-o)** относится к порту для мультимедийных пакетов в параметре LocalConnectionDescriptor адаптера МТА (исходящего);
- **Порт(SDP-t)** относится к порту для мультимедийных пакетов в параметре LocalConnectionDescriptor адаптера МТА (входящего);
- **Порт(o)** относится к порту источника, который будет использовать адаптер МТА (исходящий) при передаче мультимедийных пакетов по этому соединению. Следует отметить, что этот порт может быть или не быть тем же, что и порт(SDP-o).

Когда параметр RemoteConnectionDescriptor еще не предоставлен, фактический IP-адрес адресата и порт-адресат неизвестны, поэтому вместо них используется адрес Reserve Destination (адресата резерва). Для направления приема IP-адрес и порт источника определяются по подстановочному знаку. Это позволяет осуществлять резервирование и предоставление ресурса в направлении приема по линии доступа. ДОЛЖНЫ использоваться следующие классификаторы:

	MTA-o (DOCSIS)
<b>Нисходящее направление/прием</b>	
IP-адрес источника	*
Порт источника	*
IP-адрес адресата	IP(SDP-o)
Порт-адресат	Порт(SDP-o)
<b>Восходящее направление/передача</b>	
IP-адрес источника	IP(SDP-o)
Порт источника	Порт(o)
IP-адрес адресата	IP(RD-o)
Порт-адресат	Порт(RD-o)

где:

- **IP(RD-o)** относится к IP-адресу в предоставляемом ReserveDestination (адресат ресурса);
- **IP(Порт-o)** относится к номеру порта в предоставляемом ReserveDestination. Если номер порта не задан, то используется значение по умолчанию 9;
- если известны фактические адреса адресата и источника для передачи и приема мультимедийных пакетов, а также порт, то резервирования обновляются с помощью соответствующих классификаторов;

### Конечный автомат NCS/D-QoS

Как пояснялось выше, адаптер МТА поддерживает состояние для динамического качества обслуживания, используемого в соединении. Состояние порождается конечным автоматом, на управление которым воздействуют следующие факторы:

- **Текущее состояние**, состоящее из пары (SendQoSState, ReceiveQoSState – состояние качества обслуживания при передаче, состояние качества обслуживания при приеме), где каждое состояние QoS может быть одним из следующих состояний:
  - **N** – резервирование ресурсов для данного направления отсутствует;
  - **R** – резервирование ресурсов существует для данного направления, но в текущее время ресурсы не предоставлены;
  - **C** – резервирование ресурсов существует для данного направления; некоторые ресурсы предоставлены в текущее время;
  - **Режим соединения**, являющийся режимом соединения сигнализации NCS. В конечном автомате не присутствуют в явном виде режимы соединения "конференция", "сетевой шлюз" и "проверка целостности сети", поскольку они подобны состоянию "передача/прием" (SendReceive). Режим соединения "копирование" также не представлен, поскольку он подобен состоянию "только передача" (SendOnly).
- **Изменение ресурсов**, представляющее одно или несколько из следующих действий:
  - изменения IP-адреса параметра RemoteConnectionDescriptor или порта (необходимо обновление классификатора). Сюда входит случай изменений с самого начала;
  - изменения кодеков (включая изменение типа медиаданных MIME);

- изменения параметра ptime;
- прочее.

• **Правила для D-QoS** представлены в п. 6.3.3.

В качестве результата команды CreateConnection или ModifyConnection, если параметры QoS DOCSIS, блок классификатора и авторизации изменили свое предыдущее состояние, то новое сообщение DSA/DSC ДОЛЖНО быть послано от E-MTA к CMTS, содержащее обновленные параметры. E-MTA МОЖЕТ послать сообщение DSC к CMTS, если параметры QoS DOCSIS, блок классификатора и авторизации не изменились в результате команды CreateConnection или ModifyConnection. Если будет также предоставлен ResourceID и он будет таким же, как старый ResourceID, то резервирование (резервирования) для нового конечного автомата ДОЛЖНО быть выполнено до отмены (освобождения) резервирования (резервирований) для старого конечного автомата.

Множество возможных *состояний* таково:

- (N, N) ресурсы в направлении передачи не зарезервированы, ресурсы в направлении приема не зарезервированы;
- (R, R) ресурсы в направлении передачи зарезервированы, ресурсы в направлении приема зарезервированы;
- (C, R) ресурсы в направлении передачи зарезервированы и предоставлены, ресурсы в направлении приема зарезервированы;
- (R, C) ресурсы в направлении передачи зарезервированы, ресурсы в направлении приема зарезервированы и предоставлены;
- (C, C) ресурсы в направлении передачи зарезервированы и предоставлены, ресурсы в направлении приема зарезервированы и предоставлены;
- (R, N) ресурсы в направлении передачи зарезервированы, ресурсы в направлении приема не зарезервированы;
- (C, N) ресурсы в направлении передачи зарезервированы и предоставлены, ресурсы в направлении приема не зарезервированы;
- (N, R) ресурсы в направлении передачи не зарезервированы, ресурсы в направлении приема зарезервированы;
- (N, C) ресурсы в направлении передачи не зарезервированы, ресурсы в направлении приема зарезервированы и предоставлены.

Раз ресурсы были зарезервированы и/или предоставлены для какого-либо из направлений, то резервирование для этого направления будет оставаться в силе в течение времени существования соединения. В таблице, ниже, показана связь между состояниями и режимом соединения или параметрами резервирования для D-QoS:

	Состояние при передаче	Состояние при приеме
<b>Параметр "резервирование/предоставление" не задан – режим соединения:</b>		
Inactive	R	R
sendonly, replcate	C	R
recvonly	R	C
sendrecv, confrnce, netwloop, netwtst	C	C
<b>Параметр "резервирование/предоставление" задан</b>		
sendresv	R	N, R (Прим.)
recvresv	N, R(Прим.)	R
snrcresv	R	R
sendcomt	C	N, R (Прим.)
recvcomt	N, R (Прим.)	C
snrccomt	C	C
ПРИМЕЧАНИЕ. – Если для данного направления ресурсы были зарезервированы или предоставлены ранее, то состоянием будет R, в противном случае состоянием будет N.		

На рисунке В.1 представлена диаграмма действительных переходов состояний:

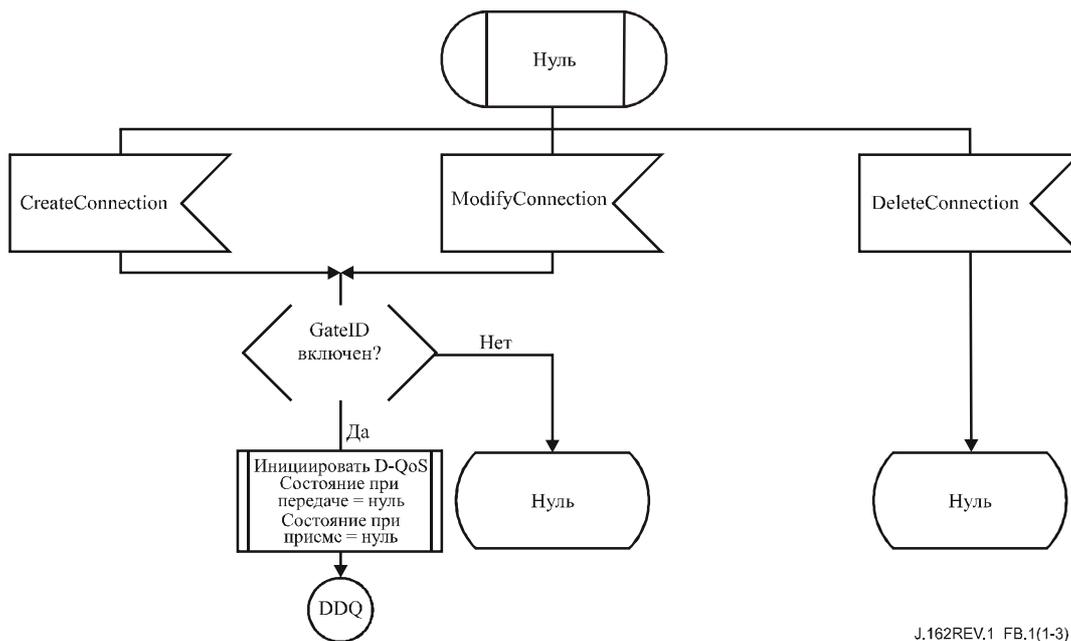
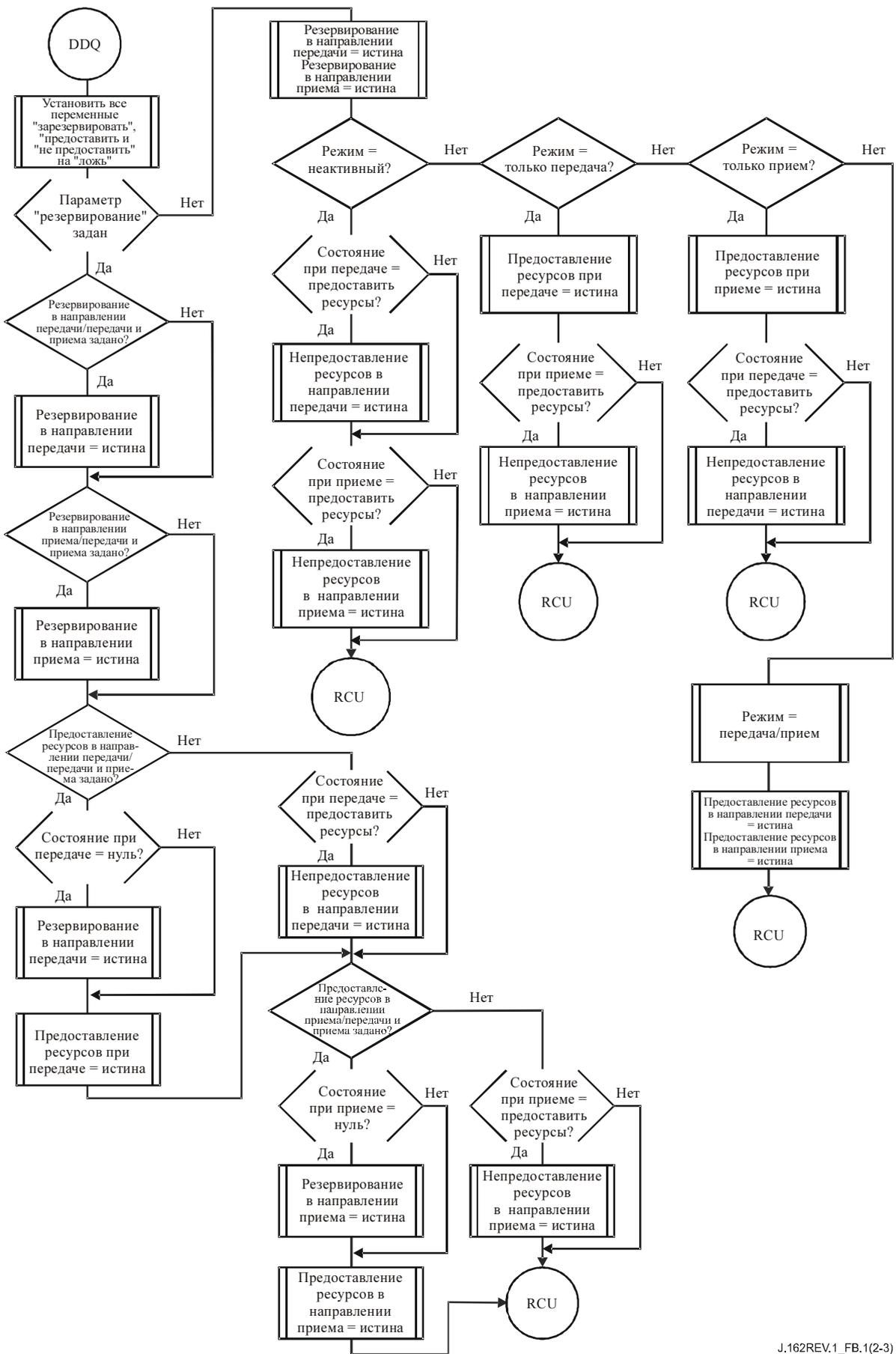


Рисунок В.1/J.162 – Диаграмма состояний NCS/D-QoS (лист 1 из 3)



J.162REV.1\_FB.1(2-3)

Рисунок В.1/J.162 – Диаграмма состояний NCS/D-QoS (лист 2 из 3)

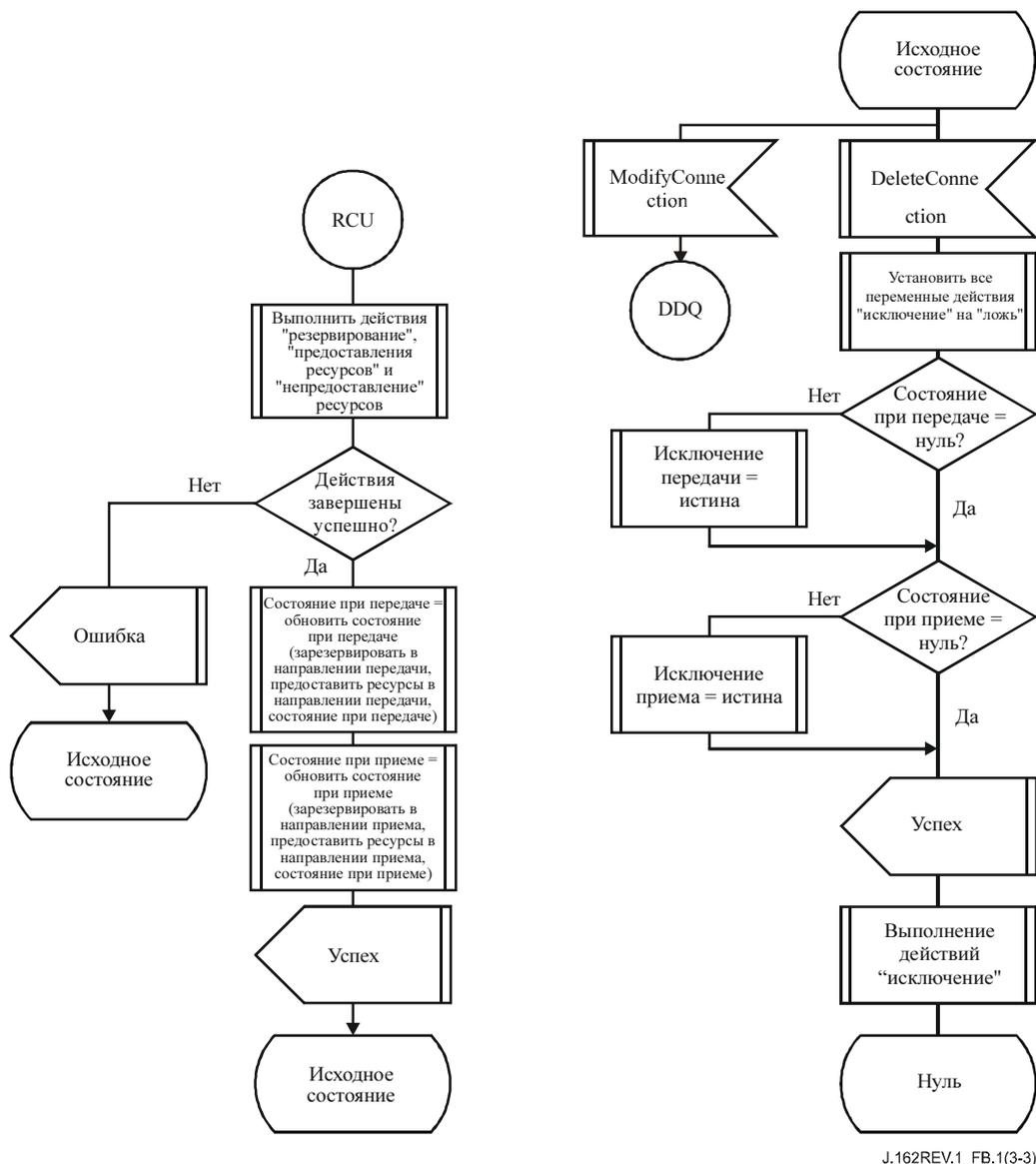


Рисунок В.1/J.162 – Диаграмма состояний NCS/ D-QoS (лист 3 из 3)

При использовании конечного автомата булевы переменные будут устанавливаться так, чтобы указывать на то, должны ли выполняться операции "зарезервировать ресурсы", "не резервировать ресурсы", "предоставить ресурсы" и "не предоставлять ресурсы". Псевдокод, приведенный ниже, обеспечивает детализацию по отдельным процедурам D-QoS, которые должны выполняться, как указано этими булевыми переменными. Следующие *действия* определяют действия D-QoS, которые должны быть предприняты в каждой из этих процедур:

- **SR** означает, что будет выполнено резервирование ресурсов в направлении передачи для D-QoS;
- **RR** означает, что будет выполнено резервирование ресурсов в направлении приема для D-QoS;
- **SC** означает, что будет выполнено предоставление ресурсов в направлении передачи для D-QoS;
- **RC** означает, что будет выполнено предоставление ресурсов в направлении приема для D-QoS;
- **SD** означает, что будет выполнено исключение резервирования ресурсов в направлении передачи для D-QoS;
- **RD** означает, что будет выполнено исключение резервирования ресурсов в направлении приема для D-QoS;
- **SU** означает, что будет выполнено непредоставление ресурсов в направлении передачи для D-QoS, то есть сведение предоставленных ресурсов в направлении передачи до нуля;
- **RU** означает, что будет выполнено непредоставление ресурсов в направлении приема для D-QoS, то есть сведение предоставленных ресурсов в направлении приема к нулю.

**SendReserve()**

```
If <current resources reserved ≠ resources to reserve> then {
    -- опустить резервирование, если с существующим
    -- резервированием все в порядке
    If <RemoteConnectionDescriptor provided> then
        SR(RemoteConnectionDescriptor)
        -- Использовать классификатор --RemoteConnectionDescriptor
    else if <ReserveDestination provided> then
        SR(ReserveDestination)
        -- Использовать классификатор параметров ReserveDestination
    else ERROR
}
```

**ReceiveReserve()**

```
If <current resources reserved ≠ resources to reserve> then {
    -- опустить резервирование, если с существующим
    -- резервированием все в порядке
    If <RemoteConnectionDescriptor provided> then
        RR(RemoteConnectionDescriptor)
        -- Использовать классификатор RemoteConnectionDescriptor
    else RR(*)
        -- Использовать классификатор подстановочных знаков,
    }
}
```

**SendCommit()**

```
If <current resources committed ≠ resources to commit> then {
    -- опустить предоставление ресурсов, если с существующим все в
    -- порядке
    If <RemoteConnectionDescriptor provided> then {
        If not <resources to commit ≤ resources reserved > then {
            -- старое резервирование не соответствует тому, что должно
            -- быть предоставлено, поэтому обновить резервирование
            SR(RemoteConnectionDescriptor)
        }
        {
            SC(RemoteConnectionDescriptor)
        }
    } else ERROR. -- Не может предоставить ресурсы в направлении передачи без
    -- параметра RemoteConnectionDescriptor
}
```

**ReceiveCommit()**

```
If <current resources committed ≠ resources to commit> then {
    -- опустить предоставление ресурсов, если с существующим все в
    -- порядке
    If not <resources to commit ≤ resources reserved> then {
        If <RemoteConnectionDescriptor provided> then
            RR(RemoteConnectionDescriptor)
        else
            RR(*) -- Использовать классификатор подстановочных знаков,
    }
    If <RemoteConnectionDescriptor provided> then
        RC(RemoteConnectionDescriptor)
    else
        RC(*) -- Использовать классификатор подстановочных знаков,
}
```

**SendReserveDelete()**

```
If <send resources reserved> then
    SD() -- исключить резервирование
```

**ReceiveReserveDelete()**

```
If <receive resources reserved> then
    RD()      -- исключить резервирование
```

**SendUnCommit()**

```
If <send resources committed> then
    SU()      -- не предоставлять предоставленные ресурсы
```

**ReceiveUnCommit()**

```
If <receive resources committed> then
    RU()      -- не предоставлять предоставленные ресурсы
```

**State UpdateState(DoCommit, DoReserve, OldState)**

```
If <DoCommit = true> then
    return Commit
else if <DoReserve = true> then
    return Reserve
else
    return OldState
```

## Дополнение I

### Пример пакета событий

В настоящем Дополнении приводится пример пакета событий для аналоговых линий доступа. Этот пакет используется здесь только в иллюстративных целях и для содействия включению информативных примеров в основную часть настоящей Рекомендации. Он никоим образом не служит законченным определением пакета, а имя этого пакета не следует рассматривать как присвоенное ему имя. Поскольку этот пакет является просто примером, то детали отдельных событий и сигналов здесь также опущены и представлены только как описания высокого уровня, служащие иллюстративным целям.

#### Пример линейного пакета

Имя пакета: X

Для идентификации событий и сигналов в примере "линейного" пакета для "аналоговых линий доступа" используются следующие коды:

Код	Описание	Событие	Сигнал	Дополнительная информация
0-9,*,#,A,B,C,D	Двухтональные многочастотные (DTMF) сигналы	√	BR	
bz	Тональный сигнал занятости	–	TO	
dl	Тональный сигнал готовности к набору номера	–	TO	
hd	Переход в состояние "телефонная трубка снята"	P, S	–	
hf	Флэш-сигналы	P	–	
hu	Переход в состояние "телефонная трубка положена"	P, S	–	
rg	Вызывной сигнал	–	TO	
rt	Тональный сигнал контроля посылки вызова	–	C, TO	
t	Таймер	√	–	
vmwi	Визуальный индикатор ожидания сообщения	–	OO	
X	Подстановочный знак для сигналов DTMF	√	–	Совпадает с любой из цифр "0–9"

Поскольку приведенный выше пакет является просто примером, то приведенное ниже определение отдельных событий и сигналов представлено только как описание высокого уровня. Действительный и реализуемый пакеты должны уточнять детали каждого события и сигнала. Эти детали могут быть разными у поставщиков услуг аналоговой КТСОП:

**Двухтональные многочастотные (DTMF) сигналы (0–9,\*,#,A, B,C,D):** Определяют все сигналы DTMF.

**Тональный сигнал занятости (bz):** Тональный сигнал занятости указывает вызывающей стороне, что вызываемая сторона уже участвует в некотором вызове.

**Тональный сигнал готовности к набору номера (dl):** Тональный сигнал готовности к набору номера указывает вызывающей стороне, что можно посылать вызов.

**Переход в состояние "телефонная трубка снята" (hd):** Событие "телефонная трубка снята" указывает на то, что у телефонного аппарата, связанного с конечной точкой, поднята телефонная трубка (токовое состояние абонентского шлейфа).

**Флэш-сигналы (hf):** Событие "флэш-сигналы" указывает на то, что телефонный аппарат, связанный с конечной точкой, посылает флэш-сигналы.

**Переход в состояние "телефонная трубка положена" (hu):** Событие "телефонная трубка положена" указывает на то, что у телефонного аппарата, связанного с конечной точкой, телефонная трубка положена (бестоковое состояние абонентского шлейфа).

**Вызывной сигнал (rg):** Вызывной сигнал указывает на то, что телефонный аппарат вызываемой стороны должен "звонить".

**Тональный сигнал контроля посылки вызова (rt):** Тональный сигнал контроля посылки вызова информирует вызывающую сторону о том, что вызываемая сторона оповещается о посылке к ней вызова.

**Таймер (t):** Как описано в п. 6.1.5, таймер Т – это обеспечиваемый таймер, который может быть аннулирован только вводом сигнала DTMF.

**Визуальный индикатор ожидания сообщения (vmwi):** Сигнал визуального индикатора ожидания сообщения либо активизирует, либо нейтрализует визуальную индикацию ожидания сообщения голосовой почты.

**Подстановочный знак для сигналов DTMF (X):** Подстановочный знак сигналов DTMF соответствует любой цифре двухтонального многочастотного (DTMF) набора номера от 0 до 9.

## Дополнение II

### Примеры кодирования команд

В настоящем Дополнении представлены примеры команд и ответов при действительном кодировании в случае использования примера линейного пакета. Примеры приведены для каждой команды. Все приведенные в командах и ответах комментарии носят необязательный характер.

#### II.1 Команда NotificationRequest

Первый пример иллюстрирует использование команды NotificationRequest для вызывного сигнала и поиска события "телефонная трубка снята":

```
RQNT 1201 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hd(N)
S: rg
```

Ответ указывает на то, что транзакция была успешной:

```
200 1201 OK
```

Второй пример иллюстрирует использование команды NotificationRequest для поиска и занесения в накопитель события "телефонная трубка снята", а затем для передачи сигнала готовности к набору номера и накопления цифр набора номера согласно имеющемуся отображению цифр. "Уведомленный объект" устанавливается на "ca@ca1.whatever.net:5678", и поскольку параметр SignalRequests является пустым<sup>35</sup>, все активные в текущее время сигналы ТО останавливаются. Все события в карантинном буфере будут обработаны, а список событий, обнаруживаемых в состоянии "уведомление" и "шаг блокировки", будет включать тональные сигналы факсимильной связи в дополнение к "запрашиваемым событиям" и устойчивым событиям:

```
RQNT 1202 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hd(A, E(S(d1), R(B/oc, hu, [0-9#*T](D))))
D: (0T|00T|#xxxxxxx|*xx|91xxxxxxxxxxxx|9011x.T)
S:
Q: process
T: ft
```

Ответ указывает на то, что транзакция была успешной:

```
200 1202 OK
```

#### II.2 Команда Notify

Приведенный ниже пример иллюстрирует применение сообщения Notify для уведомления о событии "телефонная трубка снята", за которым следует 12-значный номер, начинающийся на "91". Включен идентификатор запроса, коррелирующий команду Notify с командой NotificationRequest, из которой происходит команда Notify. Команда посылается текущему "уведомленному объекту", который, как правило, будет фактическим значением, передаваемым в параметре NotifiedEntity, то есть ca@ca1.whatever.net:5678, – ситуация восстановления после отказа могла бы это изменить:

```
NTFY 2002 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: hd,9,1,2,0,1,8,2,9,4,2,6,6
```

---

<sup>35</sup> Он может быть также пропущенным.

Ответ на команду Notify указывает на то, что транзакция была успешной:

```
200 2002 OK
```

### II.3 Команда CreateConnection

В первом примере показано использование команды CreateConnection для создания соединения в заданной конечной точке. Соединение будет частью заданного идентификатора вызова (CallId). Параметр LocalConnectionOptions определяет, что  $\mu$ -закон согласно Рекомендации МСЭ-Т G.711 будет использоваться кодеком, а период пакетирования составит 10 мс. Режимом соединения будет режим "только прием":

```
CRCX 1204 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
```

Ответ указывает на то, что транзакция была успешной, и поэтому включен идентификатор соединения для вновь созданного соединения. Также включено описание сеанса для нового соединения; следует отметить, что это описание сеанса предваряется пустой строкой.

```
200 1204 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

Во втором примере показана команда CreateConnection, содержащая запрос на уведомление и параметр RemoteConnectionDescriptor:

```
CRCX 1205 aaln/1@rgw-2569.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: sendrecv
X: 0123456789AD
R: hd
S: rg
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

Ответ указывает на то, что транзакция была неуспешной, поскольку телефонный аппарат уже был в состоянии "телефонная трубка снята". Следовательно, не возвращаются ни идентификатор соединения, ни описание сеанса:

```
401 1205 Phone off-hook
```

В третьем примере показано использование временного ответа и установления трехсторонней связи. В этом случае создается другое соединение с использованием динамического качества обслуживания и подтверждения полученного предыдущего ответа:

```
CRCX 1206 aaln/1@rgw-2569.whatever.net MGCP 1.0 NCS 1.0
K: 1205
C: A3C47F21456789F0
L: p:10, a:PCMU, dq-gi:A735C2
M: inactive
```

```
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 18
a=mptime:10 10
```

Сначала возвращается временный ответ:

```
100 1206 Pending
I: DFE233D1
```

```
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

Следует отметить, что конечная точка была выбрана только для поддержки кодека PCMU, то есть номер полезной нагрузки равен 0.

Немного позднее принимается окончательный ответ:

```
200 1206 OK
K:
DQ-RI: A12D5F1
I: DFE233D1
```

```
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

Агент вызова подтверждает окончательный ответ в соответствии с запросом:

```
000 1206
```

и транзакция завершается.

#### II.4 Команда ModifyConnection

В первом примере показано использование команды ModifyConnection, которая просто устанавливает соединение в режим "передача/прием"; также устанавливается "уведомленный объект":

```
MDCX 1209 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
N: ca@ca1.whatever.net
M: sendrecv
```

Ответ указывает на то, что транзакция была успешной:

```
200 1209 OK
```

Во втором примере описание сеанса пропускается и включается запрос на уведомление по команде `ModifyConnection`. Конечная точка будет передавать пользователю тональные сигналы контроля посылки вызова, пока она не обнаружит в соединении аудиоинформацию, определенную для события начала передачи мультимедийных пакетов:

```
MDCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: hu, ma@FDE234C8
S: rt

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

Ответ указывает на то, что транзакция была успешной:

```
200 1206 OK
```

## II.5 Команда `DeleteConnection` (от агента вызова)

В этом примере агент вызова просто передает встроенному клиенту команду исключить соединение `FDE234C8` в заданной конечной точке:

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
```

Ответ указывает на успешное выполнение команды и на то, что соединение было исключено. Поэтому также включены параметры соединения для данного соединения:

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48,
    PC/RPS=782, PC/ROS=45238, PC/RPL=5, PC/RJI=26
```

## II.6 Команда `DeleteConnection` (от встроенного клиента)

В этом примере встроенный клиент посылает агенту вызова команду `DeleteConnection`, чтобы было исключено соединение в заданной конечной точке. Параметр `ReasonCode` определяет причину исключения; кроме того, предоставляются параметры соединения:

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
E: 900 - Hardware error
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48,
    PC/RPS=782, PC/ROS=45238, PC/RPL=5, PC/RJI=26
```

Агент вызова передает шлюзу успешный ответ:

```
200 1210 OK
```

## II.7 Команда DeleteConnection (от агента вызова для нескольких соединений)

В первом примере агент вызова передает встроенному клиенту команду исключить все соединения, связанные с вызовом "A3C47F21456789F0" в заданной конечной точке:

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
```

Ответ указывает на успешное выполнение команды и на исключение соединения (соединений):

```
250 1210 OK
```

Во втором примере агент вызова передает встроенному клиенту команду исключить все соединения, относящиеся ко всем заданным конечным точкам:

```
DLCX 1210 aaln/*@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
```

Ответ указывает на успешное выполнение команды:

```
250 1210 OK
```

## II.8 Команда AuditEndpoint

В первом примере агент вызова хочет узнать, какие конечные точки имеются у определенного встроенного клиента, и, следовательно, используется подстановочный знак "all of" ("все из") для локальной части имени конечной точки:

```
AUEP 1200 *@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
```

Встроенный клиент указывает на успешное выполнение команды и предоставляет список имен конечных точек:

```
200 1200 OK
Z: aaln/1@rgw-2567.whatever.net
Z: aaln/2@rgw-2567.whatever.net
```

Во втором примере запрашиваются возможности одной из конечных точек:

```
AUEP 1201 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0 F: A
```

Ответ указывает на успешный ответ на запрос, а также на запрашиваемые возможности. Поддерживаются два кодека, но с разными возможностями. Как следствие, возвращаются два отдельных набора возможностей. И вновь каждый набор возможностей должен быть возвращен на одной строке. Из-за ограничений, связанных с форматированием, в приведенном ниже примере представлено несколько строк:

```
200 1201 OK

A: a:PCMU, p:10-100, e:on, s:off, v:X;B, m:sendonly;
    recvonly;sendrecv;inactive;netwloop;netwtest
A: a:G729, p:30-90, e:on, s:on, v:X;B, m:sendonly;
    recvonly;sendrecv;inactive;confrnce;netwloop
```

В третьем примере агент вызова контролирует всю возможную информацию для конечной точки:

```
AUEP 2002 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
F: R,D,S,X,N,I,T,O,ES,VS,E,MD
```

Ответ указывает на успешное выполнение команды:

```
200 2002 OK
R: X/lu,oc(N),[0-9](N)
D:
S: vmwi(+)
X: 0123456789B1
N: Call-agent@ca.whatever.net
I: 32F345E2
T: L/hd,L/lu,L/ft
O: hd,9,1,2
ES: hd
VS: MGCP 1.0, MGCP 1.0 NCS 1.0
E: 000
MD: 4000
```

В списке запрашиваемых событий содержатся три события. Там, где имя пакета не определено, предполагается имя пакета по умолчанию. Это справедливо и в отношении действий, так что для события "X/lu" должно, таким образом, предполагаться по умолчанию действие Notify. Пропуск значения для "отображения цифр" означает, что у конечной точки в текущее время нет отображения цифр набора номера. Активные сигналы выдержки времени в данный момент отсутствуют, тем не менее, сигнал OO для индикатора "vmvi" в текущее время включен и, следовательно, имеется – в этом случае он параметризован; однако этот параметр мог быть исключен. Текущий "уведомленный объект" относится к IP-адресу, и для конечной точки существует только одно соединение. Текущим значением параметра DetectEvents является "ft", а список параметров ObservedEvents содержит четыре определенных события. И, наконец, контроль состояний событий показал, что телефонный аппарат во время обработки транзакции был в состоянии "телефонная трубка снята".

## II.9 Команда AuditConnection

В первом примере с помощью команды AuditConnection проводится контроль параметров CallId, NotifiedEntity, LocalConnectionOptions, Connection Mode, LocalConnectionDescriptor и параметров соединения:

```
AUCX 2003 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
I: 32F345E2
F: C,N,L,M,LC,P
```

Ответ указывает на успешное выполнение команды и включает информацию для параметра RequestedInfo:

```
200 2003 OK
C: A3C47F21456789F0
N: ca@ca1.whatever.net
L: p:10, a:PCMU
M: sendrecv
P: PS=395, OS=22850, PR=615, OR=30937, PL=7, JI=26, LA=47,
    PC/RPS=615, PC/ROS=30937, PC/RPL=5, PC/RJI=26
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0
a=mptime:10
```

Во втором примере запрашивается контроль параметров RemoteConnectionDescriptor и LocalConnectionDescriptor:

```
AUCX 1203 aaln/2@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
I: FDE234C8
F: RC,LC
```

Ответ указывает на успешное выполнение контроля и содержит информацию для параметра RequestedInfo. В этом случае параметр RemoteConnectionDescriptor не существует; следовательно, для параметра RemoteConnectionDescriptor включено только поле "версия протокола":

```
200 1203 OK

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0
a=mptime:10

v=0
```

## II.10 Команда RestartInProgress

В первом примере показана передача встроенным клиентом сообщения RestartInProgress для информирования агента вызова о том, что определенная конечная точка будет выведена из эксплуатации через 300 секунд:

```
RSIP 1200 aaln/1@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
RM: graceful
RD: 300
```

Ответ агента вызова указывает на то, что транзакция была выполнена успешно:

```
200 1200 OK
```

Во втором примере сообщение RestartInProgress, переданное встроенным клиентом, информирует агента вызова о том, что все конечные точки встроенного клиента вводятся в эксплуатацию через 0 секунд, то есть они вновь находятся в эксплуатации. Задержка также могла быть опущена:

```
RSIP 1204 *@rgw-2567.whatever.net MGCP 1.0 NCS 1.0
RM: restart
RD: 0
```

Ответ агента вызова указывает на успешное выполнение команды и, кроме того, предоставляет рассматриваемым конечным точкам "новый уведомленный объект":

```
200 1204 OK
N: CA-1@ca.whatever.net
```

И наоборот, команда может быть неуспешной из-за нового "уведомленного объекта", как в случае:

```
521 1204 OK
N: CA-1@ca.whatever.net
```

В таком случае команда должна быть повторена (как новая транзакция), чтобы соответствовать "процедуре перезапуска" (см. п. 6.4.3.5), применяемой в это время к агенту вызова "CA-1@whatever.net".

## Дополнение III

### Пример последовательности операций для вызова

В настоящем Дополнении приводится пример последовательности операций для вызова между двумя встроенными клиентами ЕС-1 и ЕС-2. Следует отметить, что эта последовательность операций для вызова хотя и является действительной, служит просто примером, который может быть использован на практике или нет. В этом примере последовательности операций для вызова также используется пример линейного пакета.

В приведенной ниже последовательности операций для вызова СА означает агента вызова, CDB – базу данных конфигурации, а ACC – базу данных учета.

Usr-1	ЕС-1	СА	CDB	ACC	ЕС-2	Usr-2
	←	Запрос на уведомление				
	Подтверждение	→				
Снятие телефонной трубки	Уведомить	→				
	←	Подтверждение				
(тональный сигнал готовности к набору номера)	←	Создать соединение + запрос на уведомление				
	Подтверждение (SDP1)	→				
Цифры	Уведомить	→				
	←	Подтверждение				
(ход выполнения)	←	Запрос на уведомление				
	Подтверждение	→				
		Запрос(Е.164)	→			
		←	IP			
		Создать соединение (SDP1) + запрос на уведомление	----	----	→	
		←	----	----	P-подтверждение (SDP2)	
		←	----	----	Подтверждение (SDP2)	(вызывной сигнал)
		Подтверждение	----	----	→	
(контроль посылки вызова)	←	Модифицировать соединение (SDP2) + запрос на уведомление				
	Подтверждение	→				
		←	----	----	Уведомить	Снятие телефонной трубки
		Подтверждение	----	----	→	
	←	Модифицировать соединение + запрос на уведомление				
	Подтверждение (включение)	→				
		Начало вызова	----	→		
		Запрос на уведомление	----	----	→	
		←	----	----	Подтверждение	
		(Вызов установлен)				
		←	----	----	Уведомить	Телефонная трубка положена

Usr-1	EC-1	CA	CDB	ACC	EC-2	Usr-2
		Подтверждение	---	---	→	
	←	Исключить соединение				
		Исключить соединение	---	---	→	
	Подтверждение (Perf Data)	→				
		←	---	---	Подтверждение (Perf data)	
		Конец вызова	---	→		
		Запрос на уведомление	---	---	→	
		←	---	---	Подтверждение	
Телефонная трубка положена	Уведомить	→				
	←	Подтверждение				
	←	Запрос на уведомление				
	Подтверждение	→				

Во время этих обменов информационный агент вызова использует профиль сигнализации NCS протокола MGCP для управления обоими встроенными клиентами. Обмены информацией имеют место на двух сторонах.

Первой командой является команда NotificationRequest, посылаемая агентом вызова входному встроенному клиенту. Запрос будет состоять из следующих строк:

```
RQNT 1201 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AB
R: hd
```

Встроенный клиент в этот момент получает команду на поиск события "телефонная трубка снята" и уведомление о нем. Сначала встроенный клиент передаст ответ на команду, повторяя в ответе идентификатор транзакции, присоединенный агентом вызова к запросу, и предоставляя код завершения, указывающий на успешное выполнение команды:

```
200 1201 OK
```

Когда событие "телефонная трубка снята" замечено, встроенный клиент посылает агенту вызова сообщение Notify:

```
NTFY 2001 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AB
O: hd
```

Агент вызова немедленно подтверждает получение уведомления:

```
200 2001 OK
```

Агент вызова проверяет услуги, связанные с событием "телефонная трубка снята" для данной конечной точки (он может предпринять особые действия в случае прямой линии, отсутствия абонирования в текущее время и т.д.). В большинстве случаев агент вызова будет посылать комбинацию команд CreateConnection и NotificationRequest для создания соединения, обеспечения тонального сигнала готовности к набору номера и сбора цифр двухтонального многочастотного (DTMF) набора номера<sup>36</sup>:

```
CRCX 1202 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
```

<sup>36</sup> Фактическое отображение цифр зависит от плана набора номеров в локальной области, а также от абонированных услуг. Имеющееся отображение цифр следует рассматривать только как пример такого отображения.

```
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hu, [0-9#*T] (D)
D: (0T | 00T | [2-9]xxxxxx | 1[2-9]xxxxxxxxxx | 011xx.T)
S: dl
```

Встроенный клиент подтверждает транзакцию, посылая назад идентификацию вновь созданного соединения и описание сеанса, используемого для приема аудиоданных:

```
200 1202 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

В данном примере в спецификации протокола SDP определяется адрес, при котором встроенный клиент готов принимать аудиоданные (128.96.41.1), транспортный протокол (RTP), порт RTP (3456) и профиль аудиоданных (AVP). Профиль аудиоданных относится к стандарту RFC 3551, который определяет, что тип полезной нагрузки 0 был присвоен для передачи по  $\mu$ -закону согласно Рекомендации МСЭ-Т G.711.

Встроенный клиент начнет накапливать цифры набора номера согласно отображению цифр. Когда в дальнейшем будет иметь место соответствие этому отображению цифр, встроенный клиент уведомит агента вызова о наблюдаемых событиях:

```
NTFY 2002 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: 1,2,0,1,8,2,9,4,2,6,6
```

Агент вызова немедленно подтверждает это уведомление:

```
200 2002 OK
```

На этом этапе агент вызова пошлет команду NotificationRequest для прекращения накопления цифр набора номера, все еще продолжая следить за переходом в состояние "телефонная трубка положена". Кроме того, агент вызова принимает решение подтвердить получение ответов для транзакции 1202:

```
RQNT 1203 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
K: 1202
X: 0123456789AD
R: hu
```

Встроенный клиент немедленно подтверждает эту команду:

```
200 1203 OK
```

Агент вызова должен теперь создать соединение у выходного встроенного клиента EC-2, а также послать вызывной сигнал к телефонному аппарату, присоединенному к встроенному клиенту. Он делает это путем передачи команды CreateConnection, объединенной с командой NotificationRequest, встроенному клиенту:

```
CRCX 2001 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: sendrecv
X: 0123456789B0
R: hd
S: rg
```

```
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
a=mptime:10
```

В этот момент выходной встроенный клиент получает команду послать вызывной сигнал, отыскать событие "телефонная трубка снята" и сообщить об этом. Это событие и вызывной сигнал синхронизированы, поэтому когда имеет место событие "телефонная трубка снята", вызывной сигнал прекращается. Часть команды по созданию соединения имеет те же параметры, что и команда, посланная входному встроенному клиенту, при двух различиях:

- идентификатор конечной точки указывает на исходящий канал;
- сообщение переносит описание сеанса, возвращенное входным встроенным клиентом;
- поскольку имеется описание сеанса, параметр "режим" устанавливается в позицию "передача/прием".

Очевидно, что для этих двух соединений идентификатор вызова одинаков. Это нормально, поскольку эти два соединения относятся к одному и тому же вызову.

Предполагается, что выполнение команды не заканчивается немедленно<sup>37</sup>, и поэтому выходной встроенный клиент возвращает временный ответ, подтверждая команду и передавая в описании сеанса свои собственные параметры, такие как адрес, порты и профиль протокола RTP, а также идентификатор соединения для нового соединения:

```
100 2001 Pending
I: 32F345E2

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
a=mptime:10
```

Как только заканчивается выполнение транзакции, встроенный клиент посылает агенту вызова окончательный ответ, повторяя информацию, которую он передал во временном ответе:

```
200 2001 OK
K:
I: 32F345E2

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
a=mptime:10
```

Когда агент вызова принимает окончательный ответ, он отмечает наличие пустого атрибута подтверждения ответа и поэтому генерирует для транзакции подтверждение ответа:

```
000 2001
```

Агент вызова передает эту информацию входному встроенному клиенту и выдает ему команду генерировать локальные тональные сигналы контроля посылки вызова, используя сочетание команд ModifyConnection и NotificationRequest:

---

<sup>37</sup> Это может происходить, например из-за внешнего резервирования ресурсов, хотя в приведенном здесь примере это не рассматривается.

```
MDCX 1204 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: hu
S: rt
```

```
v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
a=mptime:10
```

Встроенный клиент немедленно подтверждает модификацию:

```
200 1204 OK
```

На этом этапе агент вызова устанавливает полудуплексный тракт передачи. Телефонный аппарат, присоединенный к входному встроенному клиенту, будет способен принимать сигналы, такие как тональные сигналы или уведомления, которые могут генерироваться в случае любых ошибок, а также первоначальную речь, которая, наиболее вероятно, будет иметь место, когда выходной пользователь ответит по телефону.

Когда наблюдается событие "телефонная трубка снята", выходной встроенный клиент посылает агенту вызова сообщение Notify:

```
NTFY 3001 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789B0
O: hd
```

Агент вызова немедленно подтверждает это уведомление:

```
200 3001 OK
```

Теперь агент вызова посылает входному встроенному клиенту комбинацию команд ModifyConnection и NotificationRequest, чтобы ввести соединение в режим "передача/прием" и остановить передачу тональных сигналов контроля посылки вызова:

```
MDCX 1206 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: sendrecv
X: 0123456789AF
R: hu
```

Встроенный клиент немедленно отвечает на команду:

```
200 1206 OK
```

Параллельно с этим агент вызова просит выходного встроенного клиента уведомить его о появлении события "телефонная трубка положена". Он делает это путем передачи встроенному клиенту команды NotificationRequest<sup>38</sup>:

```
RQNT 2002 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789B1
R: hu
```

Встроенный клиент немедленно отвечает на команду:

```
200 2002 OK
```

---

<sup>38</sup> Следует отметить, что хотя событие "телефонная трубка положена" является устойчивым событием, режим "шаг блокировки" требует, чтобы агент вызова передал встроенному клиенту новую команду NotificationRequest.

В этот момент соединение для вызова полностью установлено.

В какой-то более поздний момент времени, согласно принятому сценарию, телефонный аппарат, присоединенный к выходному встроенному клиенту, переходит в состояние "телефонная трубка положена". В соответствии с принятой в последней команде NotificationRequest стратегией Агент вызова уведомляется об этом событии путем передачи команды Notify:

```
NTFY 2003 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789B1
O: hu
```

Агент вызова немедленно отвечает на эту команду:

```
200 2003 OK
```

Теперь агент вызова определяет, что вызов завершается, и поэтому посылает обоим встроенным клиентам команду DeleteConnection:

```
DLCX 1207 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8

DLCX 2004 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: 32F345E2
```

Встроенные клиенты ответят на эту команду подтверждениями, которые включают параметры соединения для данного соединения:

```
250 1207 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48,
  PC/RPS=790, PC/ROS=45700, PC/RPL=15, PC/RJI=26
250 2004 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27, LA=48,
  PC/RPS=1245, PC/ROS=62345, PC/RPL=10, PC/RJI=27
```

Агент вызова также выдаст выходному встроенному клиенту новую команду NotificationRequest, чтобы быть готовым принять следующее событие "телефонная трубка снята", обнаруженное встроенным клиентом:

```
RQNT 2005 aaln/1@ec-2.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789B2
R: hd
```

Встроенный клиент подтвердит это сообщение:

```
200 2005 OK
```

И, наконец, входной встроенный клиент посылает сигнал отбоя, генерируя для агента вызова сообщение Notify:

```
NTFY 1208 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789AF
O: hu
```

Агент вызова немедленно отвечает на команду:

```
200 1208 OK
```

Затем агент вызова выдаст входному встроенному клиенту новую команду NotificationRequest, чтобы быть готовым принять следующее событие "телефонная трубка снята", обнаруженное встроенным клиентом:

```
RQNT 1209 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
X: 0123456789B3
R: hd
```

Встроенный клиент подтвердит это сообщение:

```
200 1209 OK
```

В это время оба встроенных клиента готовы для следующего вызова.

## Дополнение IV

### Режим соединения

Соединение по протоколу MGCP может пропускать один или несколько потоков мультимедийных пакетов. Эти потоки являются либо входящими, либо исходящими. Параметр "режим соединения" управляет последовательностью операций над мультимедийными пакетами в потоке таких пакетов. Когда имеется только одно соединение с конечной точкой, то отображение этих потоков является простым. Однако если с конечной точкой установлено несколько соединений, то может иметься много входящих и исходящих потоков. В зависимости от используемого режима соединения эти потоки могут взаимодействовать по-разному друг с другом и с потоками, идущими к телефонной трубке и от нее. В приведенной ниже таблице описывается, как должны смешиваться мультимедийные пакеты от разных соединений, когда существует одно или несколько соединений. В этой таблице предполагается, что нет сигналов, подаваемых в соединение. В таблице используются следующие условные обозначения:

- $A_{in}$  – входящий поток мультимедийных пакетов от соединения А;
- $B_{in}$  – входящий поток мультимедийных пакетов от соединения В;
- $H_{in}$  – входящий поток мультимедийных пакетов от микрофона телефонной трубки;
- $A_{out}$  – исходящий поток мультимедийных пакетов к соединению А;
- $B_{out}$  – исходящий поток мультимедийных пакетов к соединению В;
- $H_{out}$  – исходящий поток мультимедийных пакетов к раковине телефонной трубки.
- NA указывает на абсолютное отсутствие потока.

		Режим соединения А						
		sendonly	recvonly	sendrecv	confrnce	inactive	netwloop/ netwtest	replicate
Режим соединения В	sendonly	$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = NA$	$A_{out} = NA$ $B_{out} = H_{in}$ $H_{out} = A_{in}$	$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = A_{in}$	$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = A_{in}$	$A_{out} = NA$ $B_{out} = H_{in}$ $H_{out} = NA$	$A_{out} = A_{in}$ $B_{out} = H_{in}$ $H_{out} = NA$	$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = NA$
	recvonly		$A_{out} = NA$ $B_{out} = NA$ $H_{out} = A_{in} + B_{in}$	$A_{out} = H_{in}$ $B_{out} = NA$ $H_{out} = A_{in} + B_{in}$	$A_{out} = H_{in}$ $B_{out} = NA$ $H_{out} = A_{in} + B_{in}$	$A_{out} = NA$ $B_{out} = NA$ $H_{out} = B_{in}$	$A_{out} = A_{in}$ $B_{out} = NA$ $H_{out} = B_{in}$	$A_{out} = H_{in} + B_{in}$ $B_{out} = NA$ $H_{out} = B_{in}$
	sendrecv			$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = A_{in} + B_{in}$	$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = A_{in} + B_{in}$	$A_{out} = NA$ $B_{out} = H_{in}$ $H_{out} = B_{in}$	$A_{out} = A_{in}$ $B_{out} = H_{in}$ $H_{out} = B_{in}$	$A_{out} = H_{in} + B_{in}$ $B_{out} = H_{in}$ $H_{out} = B_{in}$
	confrnce				$A_{out} = H_{in} + B_{in}$ $B_{out} = H_{in} + A_{in}$ $H_{out} = A_{in} + B_{in}$	$A_{out} = NA$ $B_{out} = H_{in}$ $H_{out} = B_{in}$	$A_{out} = A_{in}$ $B_{out} = H_{in}$ $H_{out} = B_{in}$	$A_{out} = H_{in} + B_{in}$ $B_{out} = H_{in}$ $H_{out} = B_{in}$
	inactive					$A_{out} = NA$ $B_{out} = NA$ $H_{out} = NA$	$A_{out} = A_{in}$ $B_{out} = NA$ $H_{out} = NA$	$A_{out} = H_{in}$ $B_{out} = NA$ $H_{out} = NA$
	netwloop/ netwtest						$A_{out} = A_{in}$ $B_{out} = B_{in}$ $H_{out} = NA$	$A_{out} = H_{in}$ $B_{out} = B_{in}$ $H_{out} = NA$
	replicate							$A_{out} = H_{in}$ $B_{out} = H_{in}$ $H_{out} = NA$

Если имеются три или более соединений, то их мультимедийные пакеты будут смешиваться, как определено в таблице, ниже. Если внутренние ресурсы недоступны, так что нельзя смешать мультимедийные пакеты, то шлюз должен возвращать код ошибки 502 (недостаточные ресурсы).

Эти соединения графически можно представить следующим образом:

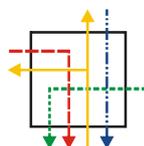


Например, если соединение А находится в режиме "sendrecv", соединение В – в режиме "confncse", а соединение С – в режиме "recvonly", то исходя из приведенной выше таблицы выходами в каждом режиме будут:

Взаимодействие от А к В	$B_{out}=H_{in}$	$A_{out}=H_{in}$	$H_{out}=A_{in}+B_{in}$	
Взаимодействие от А к С	$A_{out}=H_{in}$	$C_{out}=NA$	$H_{out}=A_{in}+C_{in}$	
Взаимодействие от В к С	$B_{out}=H_{in}$	$C_{out}=NA$	$H_{out}=B_{in}+C_{in}$	

Взяв объединение всех потоков на каждом выходе, получим:

$$\begin{aligned}
 A_{out} &= H_{in} \\
 B_{out} &= H_{in} \\
 C_{out} &= NA \\
 H_{out} &= B_{in} + A_{in} + C_{in}
 \end{aligned}$$

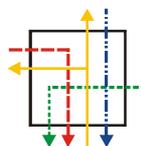


J.162REV.1App.IV\_F01

Для ясности описанная выше таблица повторяется ниже в графической форме:

Взяв объединение всех потоков на каждом выходе, получим:

$$\begin{aligned}
 A_{out} &= H_{in} \\
 B_{out} &= H_{in} \\
 C_{out} &= NA \\
 H_{out} &= B_{in} + A_{in} + C_{in}
 \end{aligned}$$



J.162REV.1App.IV\_F01b

Для ясности описанная выше таблица повторена ниже в графической форме:

		Режим соединения А (вверху)						
		sendonly	recvonly	sendrecv	confrnce	inactive	newloop/ newtest	replcate
Режим соединения В (слева)	sendonly							
	recvonly							
	sendrecv							
	confrnce							
	inactive							
	newloop/ newtest							
	replcate							

J.162REV.1.App.IV\_F02

## Дополнение V

### Информация о совместимости

В настоящем Дополнении приводится информация о совместимости протоколов сигнализации NCS.

#### Совместимость протокола MGCP

NCS является профилем протокола MGCP 1.0; однако сигнализация NCS ввела также несколько дополнений. Ниже перечисляются дополнения сигнализации NCS, которые в настоящее время не включены в протокол MGCP:

- **Схема именования конечных точек** – Правила использования подстановочных знаков носят более ограничительный характер, чем в протоколе MGCP.
- **Встроенная модификация соединения** – Было введено новое действие "встроенная команда ModifyConnection".
- **Динамическое качество обслуживания** – В NCS поддерживаются услуги обеспечения безопасности в проекте IPCablecom. Это влияет на параметр LocalConnectionOptions, возможности и протокол SDP. Кроме того, для команд CreateConnection и ModifyConnection добавляется новый параметр возврата ResourceID (идентификатор ресурса).
- **Безопасность** – В NCS поддерживаются услуги обеспечения безопасности в проекте IPCablecom. Это влияет на параметр LocalConnectionOptions, возможности и протокол SDP.
- **Восстановление имени конечной точки** – Команда AuditEndpoint была расширена путем добавления возможности возврата некоторого числа конечных точек, которые соответствуют подстановочному знаку, а также механизма блочного восстановления имен этих конечных точек. Помимо расширения команды AuditEndpoint это подразумевает ввод двух новых имен параметров: MaxEndPointIds и NumEndpoints.
- **Поддерживаемые версии** – Ответ на команду RestartInProgress и команда AuditEndpoint были расширены параметром VersionSupported, чтобы дать агентам вызовов и шлюзам возможность определять, какие версии протоколов поддерживает каждый из них.
- **Коды ошибок** – Были введены два новых кода ошибок: 532 и 533.
- **Использование протокола SDP** – В NCS включен новый профиль использования протокола SDP. Наиболее примечательно то, что профиль и все примеры использования, в частности, требуют строгого соответствия протоколу SDP, независимо от пригодности включенных полей. Кроме того, к протоколу SDP добавлены характерные для проекта IPCablecom расширения.
- **Временный ответ** – В NCS включены дополнительный элемент и спецификация механизма временного ответа. Введен ответ "подтверждение ответа" (000); в окончательных ответах, которые следуют за временными ответами, допускается пустой параметр ResponseAck (подтверждение ответа); и определена процедура для механизма временного ответа.
- **Сигнальные параметры** – Был расширен синтаксис сигнальных параметров, чтобы учесть использование симметричных круглых скобок в этих параметрах. У всех сигналов выдержки времени их значение выдержки времени может быть изменено сигнальным параметром.
- **Пакеты событий** – В NCS вводится множество новых пакетов событий.
- **Период пакетирования** – Новый множественный период пакетирования LocalConnectionOption был определен, и МТА не разрешено выбирать кодеки с размером кадра, несогласованном с периодом(ами) пакетирования, указанные агентом вызова. Таким образом, согласование кодека различается зависимостью от согласования периодов пакетирования так же, как и кодеков.

Наконец, следует отметить, что в NCS предоставляются интерпретация и, в некоторых случаях, дополнительная спецификация или пояснения базового режима работы протокола MGCP, которые могут отражать или не отражать предполагаемый режим работы протокола MGCP.

## Дополнение VI

### Дополнительные примеры пакетов событий

В настоящем Дополнении приводятся дополнительные примеры пакетов событий для различных типов конечных точек, которые определены в данное время для встроенных клиентов.

#### Аналоговые линии доступа

В настоящее время для конечных точек аналоговых линий доступа определены следующие пакеты:

- японский линейный;
- ADSL.

#### Японский линейный пакет

Имя пакета: J

В целях идентификации событий и сигналов для "японского линейного" пакета для "аналоговых линий доступа" используются следующие коды:

1) *Тип абонентской линейной сигнализации*

Абонентские линейные сигналы (сигналы) могут быть разделены на сигналы, относящиеся к управлению соединением (контрольные сигналы); сигналы, относящиеся к управлению выбором (сигналы выбора); и тональные сигналы (акустические сигналы).

2) *Контрольные сигналы*

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
cs	Сигнал исходящего вызова	P, S	–	Уведомление об исходящем вызове (= переход в состояние "телефонная трубка снята")
ir	Вызывной сигнал	–	TO	Уведомление о входящем вызове Выдержка времени = бесконечна См. статью 31, пункт 2 в документе "Carriers Telecommunication Facilities Regulations".
as1	Сигнал ответа 1	P, S	–	Уведомление, на которое ответил вызываемое оконечное устройство (от оконечного устройства к сети) (= переход в состояние "телефонная трубка снята")
as2	Сигнал ответа 2	–	TO	Уведомление о том, что вызываемый оконечное устройство ответило (от сети к оконечному устройству) Выдержка времени = бесконечная
ds1	Сигнал разъединения 1	P, S	–	Уведомление о том, что связь завершена (от оконечного устройства к сети) (= переход в состояние "телефонная трубка положена")
ds2	Сигнал разъединения 2	–	TO	Уведомление о том, что исходящее оконечное устройство завершило связь (от сети к оконечному устройству). Выдержка времени = бесконечная
cbs	Сигнал отбоя	P, S	–	Уведомление о том, что вызываемое оконечное устройство завершило связь (= переход в состояние "телефонная трубка положена")
hs	Сигнал подключения	P	–	Для "ожидания вызова" и "трехсторонней связи"
sir	Сигнал расширения вызова	–	TO	Выдается централизованной системой расширений (CES) Выдержка времени = бесконечная

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
tir	Предупредительный сигнал переадресации вызова	–	ТО	Для услуги "речевое отклонение" Выдержка времени = 2–3 с
car	Сигнал активизации оконечного устройства, принимающего данные	–	ТО	Уведомление модемным сигналом Выдержка времени = бесконечная
pas	Сигнал первичного ответа	P, S	–	Для отображения номера на экране (= переход в состояние "телефонная трубка снята")
iss	Входящий успешный сигнал	P, S	–	Для отображения номера на экране (= переход в состояние "телефонная трубка положена")
cei1(nu)	Идентификатор вызываемого абонента (тональный сигнал тастатурного набора)	–	BR	"nu" обозначает номер
cei2(nu)	Идентификатор вызываемого абонента (модемный тональный сигнал)	–	BR	"nu" обозначает номер
ci	Идентификатор вызывающего абонента	–	BR	"nu" обозначает номер
aw	Тональный сигнал ответа	✓	–	
ft	Тональный сигнал факсимильной связи	✓	–	
mt	Модемный тональный сигнал	✓	–	
ma	Мультимедийный запуск	C	–	
oc	Операция завершена	✓	–	
of	Ошибка операции	✓	–	
t	Таймер	✓	–	
l	Большая длительность сигнала DTMF	✓	–	
ld	Соединение большой продолжительности	C	–	

### 3) Сигнал выбора

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
ssn	Сигнал выбора (0–9,*,#)	✓	BR	Выдержка времени неполного набора номера = 20–30 с Выдержка времени между передачей цифр = 4–6 с
ssw	Подстановочный знак для тастатурных (PB) тональных сигналов	✓	–	Соответствует любой из цифр "0–9"

### 4) Акустические сигналы

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
dt	Тональный сигнал готовности к набору номера	–	ТО	Готовность к приему сигнала набора номера Выдержка времени = 20–30 с
sdt	Второй тональный сигнал готовности к набору номера	–	ТО	Для услуг регистрового типа, таких как "переадресация вызова", "услуга автоответчика" Выдержка времени = 20–30 с
rbt	Тональный сигнал контроля посылки вызова	–	C, TO	Выдержка времени = бесконечная
bt	Тональный сигнал занятости	–	ТО	Выдержка времени = 60–70 с

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
cpt	Тональный сигнал приема	–	BR	Для услуг регистрового типа, таких как "переадресация вызова", "услуга телефонного автоответчика"
hst	Тональный сигнал услуги удержания	–	TO	Выдержка времени = бесконечная
iit	Входящий идентификационный тональный сигнал	–	C, BR	Для "услуги телефонного автоответчика"
siit	Специфический входящий идентификационный тональный сигнал	–	C, BR	В случае двойной связи с "услугой телефонного автоответчика" и "услугой NARIWAKE"
nft	Тональный сигнал уведомления	–	TO	Только для услуги приема идентификации сообщения Выдержка времени = 3–4 с
how1	Мощный зуммер 1	–	TO	Выдержка времени = 10–22 с
how2	Мощный зуммер 2	–	TO	Выдержка времени = бесконечная

Ниже приводятся определения отдельных событий и сигналов:

**Сигнал исходящего вызова (cs):** Уведомляет сеть об исходящем вызове.

**Вызывной сигнал (ir):** См. статью 31, пункт 2 в документе "Carriers Telecommunication Facilities Regulations". В процессе обеспечения могут определяться вызывные тактовые сигналы. Вызывной сигнал может быть параметризован параметром сигнала "rep", который определяет максимальное число используемых вызывных циклов (повторений). Применение вызывного сигнала с 6 циклами выглядит следующим образом:

S: ir(rep=6)

Попытка передачи вызывного сигнала к телефону, находящемуся в состоянии "телефонная трубка снята", считается ошибкой, и, как следствие, должна быть возвращена информация об ошибке, когда предпринимаются такие попытки.

**Сигнал ответа (as):** Уведомляет сеть о том, что вызываемое оконечное устройство ответило (as1). В обратном направлении сеть посылает уведомление исходному оконечному устройству о том, что вызываемое оконечное устройство ответило (as2).

**Сигнал разъединения (ds):** Исходящее оконечное устройство уведомляет сеть о том, что связь завершена (ds1). В обратном направлении сеть посылает уведомление вызываемому оконечному устройству о том, что исходящее оконечное устройство завершило связь (ds2).

**Сигнал отбоя (cbs):** Уведомляет сеть о том, что вызываемое оконечное устройство завершило связь.

**Сигнал подключения (hs):** Оконечное устройство уведомляет сеть о выделении канала для подключения или о том, что услуга была изменена во время сеанса связи. Этот сигнал используется для "ожидания вызова" и для "трехсторонней связи".

**Сигнал расширения вызова (sir):** При использовании телефонов с централизованной системой расширений (CES) сеть уведомляет оконечное устройство о том, что входящий вызов подвергается переадресации. Кроме того, в случае "услуги NARIWAKE" сеть информирует оконечное устройство о том, что имеется входящий вызов от стороны, которая желает быть идентифицированной.

**Предупредительный сигнал переадресации вызова (tir):** Во время введения в действие услуги "переадресация по номеру телефона" или в режиме безусловной передачи в случае услуги "речевое отклонение" сеть уведомляет оконечное устройство о том, что к подписавшемуся на услугу пользователю имеется входящий вызов и переадресация активизирована.

**Сигнал активизации оконечного устройства, принимающего данные (car):** Сеть извещает оконечное устройство, принимающий данные, о том, что имеется входящий вызов с информацией, о которой уведомил модемный сигнал.

**Сигнал первичного ответа (pas):** Вызываемое оконечное устройство уведомляет сеть о том, что у телефонного аппарата сняли трубку. Эта функция используется для отображения номера на экране.

**Входящий успешный сигнал (iss):** Сеть уведомляет исходящее оконечное устройство о том, что входящий сигнал принят успешно. Эта функция используется для отображения номера на экране.

**Сигнал выбора (ss):** Исходное оконечное устройство уведомляет сеть о типе услуги и номере другой стороны. Кодом, присваиваемым сигналу выбора (0–9, \*, #), является ssn, а кодом для подстановочного знака тастатурных (PB) тональных сигналов является ssw. В приведенных далее таблицах и на рисунках представлены частоты и уровни сигналов тастатурного набора номера (PB) на приеме.

1) Частота

Группа низких частот \ Группа высоких частот	Группа высоких частот	1209 Гц	1336 Гц	1477 Гц
	Группа низких частот	697 Гц	770 Гц	852 Гц
		1	2	3
		4	5	6
		7	8	9
		*	0	#

2) Стандартные значения на приеме

Элемент		Стандартное значение
Отклонение частоты сигнала		В пределах $\pm 1,5\%$
Допустимый диапазон мощности принятого сигнала	Группа низких частот	Показано на рисунке VI.1
	Группа высоких частот	Показано на рисунке VI.2
	Отклонение электрической мощности между двумя частотами	В пределах 5 дБ, однако электрическая мощность для группы низких частот должна быть ниже такой мощности для группы высоких частот.
Время выхода сигнала		50 мс или больше
Минимальная пауза		30 мс или больше
Цикл		120 мс или больше
ПРИМЕЧАНИЕ 1. – Минимальная пауза – это наименьшее время запаздывания между смежными сигналами.		
ПРИМЕЧАНИЕ 2. – Один цикл – это сумма времени передачи сигнала и минимальной паузы.		



ПРИМЕЧАНИЕ. – Мощность принятого сигнала, когда поступающий ток меньше 20 мА, должна находиться в диапазоне от -15,4 дБм до -3,5 дБм. Когда поступающий ток больше 120 мА, тогда мощность принятого сигнала должна находиться в диапазоне от -20,3 дБм до -5,8 дБм.

**Рисунок VI.1/J.162 – Допустимый диапазон мощности принятого сигнала (группа низких частот)**



ПРИМЕЧАНИЕ. – Мощность принятого сигнала, когда поступающий ток меньше 20 мА, должна находиться в диапазоне от -14 дБм до -2,5 дБм. Когда поступающий ток больше 120 мА, тогда мощность принятого сигнала должна находиться в диапазоне от -20,3 дБм до -5,7 дБм.

**Рисунок VI.2/J.162 – Допустимый диапазон мощности принятого сигнала (группа высоких частот)**

Другие условия оговорены в Постановлении 13 Министерства почт и электросвязи 1998 года.

**Тональный сигнал готовности к набору номера (dt):** Сеть уведомляет исходящее оконечное устройство о том, что она готова принять сигнал набора номера. В случае внесетевого вызова от компонентной телефонной сети сеть уведомляет исходящее оконечное устройство о том, что она готова принять сигнал набора номера. Тональный сигнал готовности к набору номера – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями между  $(-22 - L)$  и  $-19$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц.

**Второй тональный сигнал готовности к набору номера (sdt):** Сеть уведомляет исходящее оконечное устройство о том, что она готова принять второй сигнал набора номера. В случае внесетевого вызова от компонентной телефонной сети сеть уведомляет исходящее оконечное устройство о том, что она готова принять сигнал набора номера. Второй тональный сигнал готовности к набору номера – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями между  $(-22 - L)$  дБм и  $-19$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц. Импульсный коэффициент и коэффициент замыкания находятся в пределах 240 импульсов в минуту (IPM) и 50%, соответственно.

**Тональный сигнал контроля посылки вызова (rbt):** Сеть уведомляет исходящее оконечное устройство, что она вызывает приемное оконечное устройство. Передача тонального сигнала заканчивается при получении ответа от вызываемого оконечного устройства. Акустический сигнал контроля посылки вызова является комбинацией двух тональных сигналов от переменного тока с частотами 400 Гц и 15–20 Гц и уровнями между  $-4$  дБм и  $(-29 - L)$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц. Импульсный коэффициент и коэффициент замыкания находятся в пределах  $20 \text{ IPM} \pm 20\%$  и  $33 \pm 10\%$ , соответственно (составляющая модуляции: в пределах  $85 \pm 15\%$ ).

**Тональный сигнал занятости (bt):** Сеть уведомляет исходящее оконечное устройство, что приемное оконечное устройство находится в состоянии связи; таким образом, она не может предоставить услугу или установить соединение, запрашиваемые исходящим оконечным устройством. Тональный сигнал занятости – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями между  $(-29 - L)$  дБм и  $-4$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц. Импульсный коэффициент и коэффициент замыкания находятся в пределах  $60 \text{ IPM} \pm 20\%$  и  $50 \pm 10\%$ , соответственно.

**Тональный сигнал приема (cpt):** Сеть уведомляет исходящее оконечное устройство о том, что она получила запрос на услугу. Тональный сигнал приема – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями между  $(-26 - L)$  дБм и  $-16$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц.

**Тональный сигнал услуги удержания (hst):** Сеть уведомляет ожидающее оконечное устройство о том, что состояние ожидания продолжается. Акустический сигнал услуги удержания – это комбинация двух тональных сигналов от переменного тока с частотами 400 Гц и 16 Гц и уровнями между  $-14$  дБм и  $(-22 - L)$  дБм, где  $L$  – потери при передаче в абонентском шлейфе на 400 Гц (составляющая модуляции: в пределах 85%).

**Входящий идентификационный тональный сигнал (iit):** Сеть уведомляет соответствующее вызываемое оконечное устройство, что она получила входящий вызов от третьей стороны во время разговора со второй стороной. Акустический входящий идентификационный сигнал – это комбинация двух тональных сигналов от переменного тока с частотами 400 Гц и 16 Гц и уровнями между –14 дБм и (–25 – L) дБм, где L – потери при передаче в абонентском шлейфе на 400 Гц (составляющая модуляции: в пределах 85%).

**Специфический входящий идентификационный тональный сигнал (siit):** Сеть уведомляет соответствующее вызываемое оконечное устройство, что она получила входящий вызов от третьей стороны, которая была идентифицирована. Акустический специфический входящий идентификационный сигнал – это комбинация двух тональных сигналов от переменного тока с частотами 400 Гц и 16 Гц и уровнями между –14 дБм и (–25 – L) дБм, где L – потери при передаче в абонентском шлейфе на 400 Гц (составляющая модуляции: в пределах 85%).

**Тональный сигнал уведомления (nft):** Сеть уведомляет оконечное устройство пользователя, подписавшегося на "услугу приема идентификации сообщения", о том, что она получила идентификацию сообщения. Тональный сигнал уведомления – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями между (–26 – L) дБм и –16 дБм, где L – потери при передаче в абонентском шлейфе на 400 Гц.

**Мощный зуммер (how):** Сеть уведомляет оконечное устройство о том, что у неиспользуемого телефонного приемника поднята телефонная трубка в течение определенного времени, побуждая этим сигналом опустить телефонную трубку. Имеются два мощных зуммерных сигнала. Мощный зуммер (how1) – это тональный сигнал от переменного тока с частотой 400 Гц и уровнями в +35 дБм или ниже. Мощный зуммер 1 – это звук, постепенно нарастающий в течение 3–15 секунд, и сигнал выдержки времени в течение 10–22 секунд. Мощный зуммерный сигнал 2 (how2) генерируется комбинацией из трех тональных сигналов на частотах 1600 Гц, 1000 Гц и 2000 Гц в ритме частоты в 1600 Гц в течение 0,5 секунды и двойного повторения частот 1000 Гц и 2000 Гц в течение 0,125 секунды. Уровень комбинированного тонального сигнала составляет –1 дБм или ниже. Между этими акустическими сигналами вставляется речевое указание, такое как "У приемника снята трубка". Попытка передачи мощного зуммерного сигнала к телефону с положенной трубкой считается, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке. У мощного зуммера 2 сигнал выдержки времени "бесконечен".

**Идентификатор вызываемого абонента (cei1(nu)):** Идентификатор вызываемого абонента требуется для системы сигнализации с тастатурным набором номера (PB) при прямом наборе дополнительных номеров.

**Идентификатор вызываемого абонента (cei2(nu)):** Идентификатор вызываемого абонента требуется для системы сигнализации модема при прямом наборе дополнительных номеров.

**Идентификатор вызывающего абонента (ci(time, number, name)):** Каждое из этих трех полей является необязательным; однако всегда будет включена каждая из запятых.

- Параметр **время (time)** кодируется в виде "MM/DD/HH/MM", где MM – это значение из двух цифр между 01 и 12 для обозначения месяца; DD – это значение из двух цифр между 1 и 31 для обозначения дня, а час (HH) и минута (MM) – это значения из двух цифр, закодированных согласно военному местному времени, например 00 означает полночь, 01 – 1 час ночи, а 13 – 1 час дня.
- Параметр **номер (number)** кодируется в виде символьной строки в коде ASCII из десятичных цифр, которые определяют номер вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы; однако они будут игнорироваться.
- Параметр **имя (name)** кодируется в виде строки символов в коде ASCII, которая определяет имя вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы.

Буква "P" в поле номера или имени используется для указания на частный номер или имя, а буква "O" – на недоступный номер или имя. Следующий пример иллюстрирует использование сигнала "идентификатор вызывающего абонента":

```
S: ci(02/20/19/47, "5273 4671", JCTEA)
```

**Тональный сигнал ответа (aw):** Тональный сигнал ответа – это тональный сигнал, который может быть выдан модемом или аппаратом факсимильной связи, отвечающим на входящий вызов. Этот тональный сигнал является синусоидальным сигналом на частоте 2100 Гц – см. Рекомендацию МСЭ-Т V.8.

**Тональный сигнал факсимильной связи (ft):** Событие "тональный сигнал факсимильной связи" генерируется каждый раз, когда обнаруживается факсимильный вызов – см., например, Рекомендацию МСЭ-Т T.30 или Рекомендацию МСЭ-Т V.21.

**Мультимедийный запуск (ma):** Событие "мультимедийный запуск" имеет место в соединении, когда по соединению принимается первый действительный<sup>39</sup> мультимедийный пакет согласно протоколу RTP. Это событие может быть использовано для синхронизации локального сигнала, например сигнала контроля посылки вызова, с поступлением мультимедийного пакета от другой стороны.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они создаются.

**Модемные тональные сигналы (mt):** Событие "модемный тональный сигнал" генерируется каждый раз, когда обнаруживается модемный вызов – см., например, Рекомендацию МСЭ-Т V.8.

**Операция завершена (oc):** Событие "операция завершена" генерируется, когда шлюз получает запрос на применение одного или нескольких сигналов типа ТО в конечной точке и когда один или несколько таких сигналов закончились без прерывания при обнаружении запрашиваемого события, такого как переход в состояние "телефонная трубка снята" или "набранная цифра". Сообщение о завершении может содержать в качестве параметра имя сигнала, время существования которого закончилось, как в случае:

O: L/oc (L/dt)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

O: L/oc (L/rbt@0A3F58)

Когда запрашивается событие "операция завершена", оно не может быть параметризовано любыми параметрами события. Когда имя пакета опущено, то предполагается имя пакета по умолчанию.

Событие "операция завершена" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда успешно завершается встроенная команда модификации соединения (Modify Connection), как в случае<sup>40</sup>:

O: L/oc (B/C)

**Ошибка операции (of):** В общем случае событие "ошибка операции" может быть генерировано, когда конечная точка получила запрос на применение одного или нескольких сигналов типа ТО в конечной точке и до окончания выдержки времени в одном или нескольких сигналах произошел сбой. Сообщение о завершении может содержать в качестве параметра имя сигнала, в котором произошел сбой, как в случае:

O: L/of (L/ir)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

O: L/of (L/rbt@0A3F58)

Когда запрашивается событие "ошибка операции", параметры события не могут быть определены. Когда опущено имя пакета, предполагается имя пакета по умолчанию.

Событие "ошибка операции" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда имеет место неуспешное выполнение встроенной команды модификации соединения, как в случае<sup>40</sup>:

O: L/of (B/C (M (sendrecv (AB2354) ) ) ) )

**Таймер (t):** Таймер Т – это обеспечиваемый таймер, который может быть аннулирован только путем ввода сигналов DTMF. Когда таймер Т используется с действием "накопление согласно отображению цифр", он не запускается до тех пор, пока не поступит первая цифра, и перезапускается после ввода каждой новой цифры до тех пор, пока не наступит соответствие или несоответствие этому отображению цифр. В этом случае таймер Т функционирует как таймер времени между передачей цифр и принимает одно из двух значений –  $T_{\text{par}}$  или  $T_{\text{crit}}$ . Когда для строки цифр требуется, по крайней мере, еще одна цифра, чтобы соответствовать какой-либо из комбинаций отображения цифр, таймер

<sup>39</sup> Когда используются услуги аутентификации и обеспечения защиты целостности, пакет RTP не считается действительным, пока он не пройдет проверку на обеспечение защиты.

<sup>40</sup> Следует отметить, что "B" используется здесь в качестве префикса для сообщаемого параметра.

T принимает значение  $T_{par}$ , соответствующее временным соотношениям при неполном наборе номера. Для получения полного соответствия отображению цифр таймер T принимает значение  $T_{crit}$ , соответствующее критическим временным соотношениям. Примером использования является:

S: dt  
R: [0-9T] (D)

Если таймер T используется без действия "накопление согласно отображению цифр, он принимает значение  $T_{crit}$ , запускается немедленно и просто аннулируется (но не перезапускается), как только вводится цифра. В этом случае таймер T может использоваться как таймер времени между передачей цифр, когда применяется передача с перекрытием, например:

R: [0-9] (N) , T (N)

Следует отметить, что в текущий момент может использоваться только одна из двух форм, поскольку данное событие может быть определено только один раз.

Значение по умолчанию для  $T_{par}$  составляет 16 секунд, а для  $T_{crit}$  – 4 секунды. Оба эти значения могут быть изменены в процессе обеспечения.

**Большая длительность сигнала DTMF (l):** "Большая длительность сигнала DTMF" наблюдается, когда этот сигнал выдается в течение более 2 секунд. В этом случае шлюз обнаружит два последовательных события: первое, когда сигнал был распознан как сигнал DTMF, и затем, 2 секундами позднее, сигнал большой продолжительности.

**Соединение большой продолжительности (ld):** "Соединение большой продолжительности" обнаруживается, когда соединение устанавливалось в течение времени, превышающем определенный для его установления период времени. Значение по умолчанию составляет 1 час; однако это значение может быть изменено в процессе обеспечения.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они создаются.

**Подстановочный знак для тастатурных (PB) тональных сигналов (x):** Подстановочный знак для тастатурных (PB) тональных сигналов соответствует любой цифре тастатурного набора номера от 0 до 9.

## Пакет ADSI

Имя пакета: JS

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
adsi(string)	Отображение на экране для ADSI	–	BR	

**Отображение на экране для ADSI (adsi(string)):** Интерфейс услуг для аналогового дисплея (ADSI) используется главным образом для отображения на экране дисплея телефонного номера отправителя сигналов.

## Видеоинформация

Пакеты событий для видеоинформации будут представлены в будущей версии настоящей Рекомендации.

## Дополнение VII

### Пакеты событий

В данном дополнении определяется начальное множество пакетов событий для различных типов конечных точек, определяемых в настоящее время в проекте IP-Cablecom для встроенных клиентов. Определены следующие пакеты для перечисленных типов конечных точек для встроенных клиентов:

Тип конечной точки	Пакет	Имя пакета	Пакет по умолчанию
Аналоговая линия доступа	Линейный	L	Да
Сетевой интерфейс для V5LE	Европейский	E	Нет
Видео	Для дальнейшего изучения	Для дальнейшего изучения	Для дальнейшего изучения
Интерфейс BRI для ЦСИС	Для дальнейшего изучения	Для дальнейшего изучения	Для дальнейшего изучения

Каждый пакет определяет имя пакета и коды событий и определения для каждого из событий в пакете. В таблицах событий/сигналов для каждого пакета имеется пять столбцов:

**Код:** Однозначно определяемый (уникальный) код события пакета, используемый для события/сигнала.

**Описание:** Краткое описание события/сигнала.

**Событие:** В этом столбце проставляется "галочка", если событие может быть запрошено контроллером медиашлюза. Как вариант, могут быть приведены один или несколько следующих символов:

"P" указывает на то, что событие устойчивое;

"S" указывает на то, что событие находится в таком состоянии, которое может быть проконтролировано;

"C" указывает на то, что сигнал/событие могут быть обнаружены/применены в соединении.

**Сигнал:** Если в данном столбце для события ничего не указано, тогда сигнал о событии не может быть передан в команде контроллером медиашлюза. В противном случае следующие символы определяют тип события:

"OO": Сигнал "включено/выключено". Сигнал будет включенным, пока он не будет выключен по команде от контроллера медиашлюза, и наоборот.

"TO": Сигнал выдержки времени. Этот сигнал длится в течение заданного времени, если его не заменяет новый сигнал. Предоставляются значения выдержки времени по умолчанию. Нулевое значение указывает на то, что период выдержки времени бесконечен. Процесс обеспечения может изменить эти значения по умолчанию.

"BR": Короткий сигнал. Событие имеет короткую известную длительность.

**Дополнительная информация:** Предоставляется дополнительная информация о событии/сигнале, например длительность сигналов TO по умолчанию.

Если не указано иное, то все события/сигналы обнаруживаются/применяются в конечных точках, а аудиоинформация, генерируемая ими, не передается по какому-либо из соединений, которые может иметь конечная точка. Однако аудиоинформация, генерируемая событиями/сигналами, которые обнаруживаются/применяются в соединении, будет передаваться по соответствующему соединению независимо от режима соединения.

#### Аналоговые линии доступа

Для конечных точек аналоговых линий доступа в настоящее время определен следующий пакет. Этот пакет применяется ко всем конечным точкам:

- Линейный

Имя пакета: L.

В целях идентификации событий и сигналов для "линейного" пакета для "аналоговых линий доступа" используются следующие коды:

Код	Описание	Событие	Сигнал	Дополнительная информация
0-9,*,#,A, B,C,D	Сигналы MFPB (DTMF)	√	BR	
bz	Тональный сигнал занятости	–	TO	Выдержка времени = 30 секунд
cf	Тональный сигнал подтверждения	–	BR	
ci(ti, nu, na)	Идентификатор вызывающего абонента	–	BR	"ti" означает время, "nu" означает номер, а "na" означает имя
dl	Тональный сигнал готовности к набору номера	–	TO	Выдержка времени = 16 с
ft	Тональный сигнал факсимильной связи	√	–	
hd	Переход в состояние "телефонная трубка снята"	P, S	–	
hf	Флэш-сигналы	P	–	
hu	Переход в состояние "телефонная трубка положена"	P, S	–	
L	Большая длительность сигнала MFPB (DTMF)	√	–	
ld	Соединение большой продолжительности	C	–	
ma	Мультимедийный запуск	C	–	
mt	Модемные тональные сигналы	√	–	
mwi	Индикатор ожидания сообщения	–	TO	Выдержка времени = 16 с
oc	Операция завершена	√	–	
of	Ошибка операции	√	–	
ot	Тональный предупредительный сигнал о снятии телефонной трубки	–	TO	Выдержка времени = бесконечная
r0, r1, r2, r3, r4, r5, r6 or r7	Позывные сигналы (0..7)	–	TO	Выдержка времени = 180 с
rg	Вызывной сигнал	–	TO	Выдержка времени = 180 с
ro	Тональный сигнал требования освобождения цепи	–	TO	Выдержка времени = 30 с
rs	Пакет коротких вызывных сигналов	–	BR	
rt	Тональный сигнал контроля посылки вызова	–	C, TO	Выдержка времени = 180 с
sl	Прерывистый сигнал готовности к набору номера	–	TO	Выдержка времени = 16 с
t	Таймер	√	–	
TDD	Тональные сигналы устройств электросвязи для слабослышащих (TDD)	√	–	
vmwi	Визуальный индикатор ожидания сообщения	–	OO	
wt1, wt2, wt3, wt4	Тональные сигналы ожидания вызова	–	TO	Выдержка времени = 12 с
X	Подстановочный знак для сигналов MFPB (DTMF)	√	–	Соответствует любой из цифр "0–9"

Ниже приводятся определения отдельных событий и сигналов:

**Сигналы MFPB (DTMF) (0–9,\*,#,A,B,C,D):** Обнаружение и генерирование многочастотных тастатурных (MFPB (DTMF)) сигналов описано в стандарте ETS 300 001, Глава 5: Функция вызова. Попытка передачи сигналов MFPB (DTMF) к телефону в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефонная трубка положена).

**Тональный сигнал занятости (bz):** Занятость станции определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тонального сигнала занятости к телефону в автономной режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Тональный сигнал подтверждения (cf):** Тональный сигнал подтверждения определяется местной администрацией и МОЖЕТ переопределяться через обеспечение. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тонального сигнала подтверждения к телефону, находящемуся в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Идентификатор вызывающего абонента (ci(time, number, name)):** См. стандарты EN 300 659-1 и EN 300 659-3. Каждое из трех полей является необязательным, однако всегда будет включена каждая из запятых.

- Параметр **время (time)** кодируется в виде "MM/DD/HH/MM", где MM – это значение из двух цифр между 01 и 12 для обозначения месяца; DD – это значение из двух цифр между 1 и 31 для обозначения дня, а час (HH) и минута (MM) – это значения из двух цифр, закодированные согласно военному местному времени, например 00 означает полночь, 01 – 1 час ночи, а 13 – 1 час дня.
- Параметр **номер (number)** кодируется в виде символьной строки в коде ASCII из десятичных цифр, которые определяют номер вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы, однако они будут игнорироваться.
- Параметр **имя (name)** кодируется в виде строки символов в коде ASCII, которая определяет имя вызывающего абонента. Если строка заключается в кавычки, то допускаются пробелы.

Буква "P" в поле номера или имени используется для указания на частный номер или имя, а буква "O" – на недоступный номер или имя. Следующий пример иллюстрирует использование сигнала "идентификатор вызывающего абонента":

```
S: ci(08/14/17/26, "33 4 92 94 42 00", European)
```

**Тональный сигнал готовности к набору номера (dl):** Тональный сигнал готовности к набору номера определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тонального сигнала готовности к набору номера к телефону в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Тональный сигнал факсимильной связи (ft):** Событие "тональный сигнал факсимильной связи" генерируется каждый раз, когда обнаруживается факсимильный вызов по *наличию преамбулы факсимильного сообщения согласно Рекомендации МСЭ-Т V.21*. Событие "тональный сигнал факсимильной связи" СЛЕДУЕТ также генерировать при обнаружении тонального сигнала CNG согласно Рекомендации МСЭ-Т T.30. См. Рекомендации МСЭ-Т T.30 и V.21.

**Переход в состояние "телефонная трубка снята" (hd):** См. стандарт ETSI EG 201 188, раздел 7: Сигнал занятия.

**Флэш-сигналы (hf):** См. стандарт ETSI EG 201 188, раздел 14.2: Выбор данных регистра.

**Переход в состояние "телефонная трубка положена" (hu):** См. стандарт ETSI EG 201 188, Раздел 8: Сигнал отбоя. Синхронизация сигнала "телефонная трубка положена" активизируется для ответа на флэш-сигналы.

**Большая длительность сигнала MFPB (DTMF) (L):** "Большая длительность сигнала MFPB (DTMF)" наблюдается, когда этот сигнал выдается в течение более двух секунд. В этом случае шлюз

обнаружит два последовательных события: первое, когда сигнал был распознан как сигнал МРРВ (DTMF), затем, двумя секундами позднее, сигнал большой продолжительности.

**Соединение большой продолжительности (ld):** "Соединение большой продолжительности" обнаруживается, когда соединение устанавливалось в течение времени, превышающего определенный для его установки период времени. Значение по умолчанию составляет 1 час; однако оно может быть изменено в процессе обеспечения.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они создаются.

**Мультимедийный запуск (ma):** Событие "мультимедийный запуск" имеет место в соединении, когда по соединению принимается первый действительный<sup>41</sup> мультимедийный пакет согласно протоколу RTP. Это событие может быть использовано для синхронизации локального сигнала, например сигнала контроля посылки вызова, с поступлением мультимедийного пакета от другой стороны.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они создаются.

**Модемные тональные сигналы (mt):** Модемные тональные сигналы (mt): Событие "модемный тональный сигнал" генерируется каждый раз, когда обнаруживается вызов передачи данных по присутствию тонального сигнала ответа (ANS) согласно Рекомендации МСЭ-Т V.25 с опрокидыванием фазы или без него или по присутствию модифицированного тонального сигнала ответа (ANSam) согласно Рекомендации МСЭ-Т V.8 с опрокидыванием фазы или без него. См. Рекомендации МСЭ-Т V.25 и V.8.

**Индикатор ожидания сообщения (mwi):** Тональный сигнал индикатора ожидания сообщения определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи сигнала индикатора ожидания сообщения к телефону, находящемуся в автономном режиме (телефонная трубка положена) считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Операция завершена (oc):** Событие "операция завершена" генерируется, когда шлюз получает запрос на применение одного или нескольких сигналов типа ТО в конечной точке и когда один или несколько таких сигналов закончились без прерывания при обнаружении запрашиваемого события, такого как переход в состояние "телефонная трубка снята" или "набранная цифра". Сообщение о завершении может содержать в качестве параметра имя сигнала, время существования которого закончилось, как в случае:

О: L/oc(L/d1)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

О: L/oc(L/rt@0A3F58)

Когда запрашивается событие "операция завершена", оно не может быть параметризовано любыми параметрами события. Когда имя пакета опущено, то предполагается имя пакета по умолчанию.

Событие "операция завершена" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда успешно завершается встроенная команда ModifyConnection, как в случае<sup>42</sup>:

О: L/oc(B/C)

**Ошибка операции (of):** В общем случае событие "ошибка операции" может быть генерировано, когда конечная точка получила запрос на применение одного или нескольких сигналов типа ТО в конечной точке и до окончания выдержки времени в одном или нескольких сигналах произошел

<sup>41</sup> Когда используются услуги аутентификации и обеспечения защиты целостности, пакет RTP не считается действительным, пока он не пройдет проверку на обеспечение защиты.

<sup>42</sup> Следует отметить, что "B" используется здесь в качестве префикса для сообщаемого параметра.

сбой. Сообщение о завершении может содержать в качестве параметра имя сигнала, в котором произошел сбой, как в случае:

O: L/of (L/rg)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

O: L/of (L/rt@0A3F58)

Когда запрашивается событие "ошибка операции", параметры события не могут быть определены. Когда имя пакета опущено, предполагается имя пакета по умолчанию.

Событие "ошибка операции" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда имеет место неуспешное выполнение встроеной команды ModifyConnection, как в случае:

O: L/of (B/C (M (sendrecv (AB2354) ) ) ) )

**Тональный предупредительный сигнал о снятии телефонной трубки (ot):** Тональный сигнал о состоянии ресивера "телефонная трубка снята" (РОН-тон) или "мощный зуммерный" сигнал определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тонального предупредительного сигнала о снятии телефонной трубки к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие, должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Позывные сигналы (r0, r1, r2, r3, r4, r5, r6 или r7):** Эти тактовые позывные сигналы определяются местной администрацией и МОГУТ переопределяться через процесс обеспечения.

См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 3. Попытка посылки позывного сигнала к телефону, который находится в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Вызывной сигнал (rg):** Электрический вызывной сигнал определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 3. Вызывной сигнал может быть параметризован параметром сигнала "rep", который определяет максимальное число используемых вызывных циклов (повторений). Применение вызывного сигнала с 6 циклами выглядит следующим образом:

S: rg (rep=6)

Попытка передачи вызывного сигнала к телефону, находящемуся в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Тональный сигнал требования освобождения цепи (ro):** Тональный сигнал требования освобождения цепи определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тонального сигнала требования освобождения цепи к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Пакет коротких вызывных сигналов (rs):** Пакет коротких вызывных сигналов, также известный как "сигнал-напоминание", представляет собой пакет коротких электрических сигналов, который может быть использован в физической линии переадресации (если она свободна) для указания на то, что вызов был переадресован, и для напоминания пользователю, что активизирована функция переадресации вызова. Этот сигнал определяется местной администрацией и МОЖЕТ быть переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 3. Попытка передачи этого сигнала к телефону, который находится в диалоговом режиме (телефонная трубка снята), считается ошибкой, и когда такие попытки предпринимаются, как

следствие должна быть возвращена информация об ошибке (код ошибки 401 – телефон в диалоговом режиме (телефонная трубка снята)).

**Тональный сигнал контроля посылки вызова (rt):** Это акустический сигнал, определяемый местной администрацией, который может быть переопределен через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Сигнал контроля посылки вызова может быть использован как в конечной точке, так и в соединении.

При применении сигнала контроля посылки вызова в конечной точке попытка передачи тональных сигналов контроля посылки вызова считается ошибкой, если конечная точка находится в автономном режиме (телефонная трубка положена), и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)). Когда сигнал контроля посылки вызова используется в соединении, такая проверка проводится не должна.

**Прерывистый тональный сигнал готовности к набору номера (sl):** Прерывистый тональный сигнал готовности к набору номера (также называемый тональным сигналом готовности к набору номера, требующим ответа) определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Прерывистый тональный сигнал готовности к набору номера может быть параметризован параметром сигнала "del", который будет определять задержку в миллисекундах между тональным сигналом подтверждения и тональным сигналом готовности к набору номера<sup>43</sup>. Применение прерывистого тонального сигнала готовности к набору номера с задержкой в 1,5 секунды между тональным сигналом подтверждения и тональным сигналом готовности к набору номера выглядит следующим образом:

S: sl(del=1500)

Попытка передачи прерывистого тонального сигнала готовности к набору номера к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Таймер (t):** Как описано в п. 6.1.5, таймер T – это обеспечиваемый таймер, который может быть аннулирован только путем ввода сигналов MFPB (DTMF). Когда таймер используется с действием "накопление согласно отображению цифр", он не запускается до тех пор, пока не поступит новая цифра, и перезапускается после ввода каждой новой цифры до тех пор, пока не наступит соответствие или несоответствие этому отображению цифр. В этом случае таймер T функционирует как таймер времени между передачей цифр и принимает одно из двух значений –  $T_{par}$  или  $T_{crit}$ . Когда для строки цифр требуется, по крайней мере, еще одна цифра, чтобы соответствовать какой-либо из комбинаций отображения цифр, таймер T принимает значение  $T_{par}$ , соответствующее временным соотношениям при неполном наборе номера. Для получения полного соответствия отображению цифр таймер T принимает значение  $T_{crit}$ , соответствующее критическим временным соотношениям. Примером использования является:

S: dl  
R: [0-9T] (D)

Если таймер T используется без действия "накопление согласно отображению цифр", он принимает значение  $T_{crit}$ , запускается немедленно и просто аннулируется (но не перезапускается), как только вводится цифра. В этом случае таймер T может быть использован как таймер времени между передачей цифр, когда применяется передача с перекрытием, например:

R: [0-9] (N) , T (N)

Следует отметить, что в текущий момент может использоваться только одна из двух форм, поскольку данное событие может быть определено только один раз.

Значение по умолчанию для  $T_{par}$  равно 16 секундам, а для  $T_{crit}$  – 4 секундам. Оба этих значения могут быть изменены в процессе обеспечения.

---

<sup>43</sup> Этот параметр необходим, например, для быстрого набора номера.

**Тональные сигналы устройств электросвязи для слабослышащих (TDD):** Событие TDD генерируется каждый раз, когда обнаруживается вызов TDD – см., например, Рекомендацию МСЭ-Т V.18.

**Визуальный индикатор ожидания сообщения (vmwi):** Передача сообщений VMWI будет соответствовать требованиям стандарта EN 300 659-1, раздел 6.2: Передача данных, не связанных с вызывными сигналами, и стандарта EN 300 659-3, раздел 5.2.2: Сообщение "индикатор ожидания сообщения". Сообщения VMWI будут посылаться от встроенного клиента к присоединенному оборудованию только тогда, когда линия свободна. Если новое сообщение поступает тогда, когда линия занята, то сообщение "индикатор VMWI" будет задерживаться до тех пор, пока линия снова не станет свободной. Агент вызова должен периодически обновлять визуальный индикатор оборудования СРЕ.

**Тональный сигнал 1 ожидания вызова (wt1, .., wt4):** Тональные сигналы ожидания вызова определяются местной администрацией и МОГУТ переопределяться через процесс обеспечения. См. стандарты ETSI EG 201 188 и ETS 300 001, Глава 1. Попытка передачи тональных сигналов ожидания вызова к телефону, находящемуся в автономном режиме (телефонная трубка положена), считается ошибкой, и когда такие попытки предпринимаются, как следствие должна быть возвращена информация об ошибке (код ошибки 402 – телефон в автономном режиме (телефонная трубка положена)).

**Подстановочный знак для сигналов MFPB (DTMF) (X):** Подстановочный знак для сигналов MFPB (DTMF) соответствует любой цифре многочастотного тастатурного (MFPB (DTMF)) набора номера между 0 и 9.

#### **Видеоинформация**

Пакеты событий для видеоинформации подлежат дальнейшему изучению.

#### **ЦСИС**

Пакеты событий для ЦСИС базового доступа подлежат дальнейшему изучению.

## Дополнение VIII

### Применение протокола NCS к оконечному устройству IPAT сети SCN

#### VIII.1 Обзор

В данном Дополнении определяется применение протокола NCS, описанного в основной части настоящей Рекомендации, к устройству IPAT, который может эмулировать сеть доступа к местной станции (LE) европейского стандарта, являющегося частью сети SCN. В настоящем Дополнении описывается соответствие между протоколом NCS и подмножеством протокола V5.2 (см. стандарт ETS 300 324), применяемое для поддержки услуг SCN для аналоговых телефонов. Следует отметить, что настоящее Дополнение было подготовлено в ответ на запросы операторов современных европейских кабельных сетей, связанные с предоставлением телефонных услуг по их гибридным волоконно-коаксиальным системам (HFC) при использовании существующей пропускной способности коммутационной системы V5 для доступа к сети SCN, как описано в документе, касающемся требований Рабочей группы EuroPacket Cable ассоциации ЕССА (ЕРС-ReqDoc-V10-0501, май 2001 года: Европейские требования по предоставлению критических во времени услуг по сетям кабельного телевидения при использовании протокола IP-Cablecom).

Настоящее Дополнение применяется к подмножеству протокола сигнализации V5, который относится к услугам, предоставляемым по 2-проводной (выводы a-b), включенной шлейфом аналоговой линии POTS.

**ПРИМЕЧАНИЕ 1.** – Вопрос поддержки дополнительных типов линий подлежит дальнейшему изучению. Следует иметь в виду, что хотя предлагаемый протокол обеспечивает поддержку пакета услуг POTS в сети SCN с интерфейсом V5, в связи с развивающимися требованиями рынка некоторые такие услуги могут уже не требоваться или больше не использоваться в границах некоторых администраций. Поэтому рекомендуется, чтобы соответствие изделия протоколу, поддерживающему эти услуги, базировалось на заявлении изготовителя, подобно практике использования заявлений о соответствии реализации протоколу (PICS) V5, а не на "доверительном" соответствии услугам. В тех случаях, когда изделие может не поддерживать конкретную услугу, соответствие протоколу должно интерпретироваться как возможность использовать интерфейс протокола и сгладить несоответствия в запросах на услуги и возможностей изделия. Таким способом могут быть оптимизированы сложность и стоимость изделия согласно требованиям рынка и потребностям администраций при сохранении взаимодействия протоколов.

**ПРИМЕЧАНИЕ 2.** – Описание сигналов для автоматической тарификации, приведенное в настоящем Дополнении, и описание сигналов для автономного пакета тарификации в Дополнении IX намеренно являются одинаковыми и должны оставаться такими же. Эквивалентность сигналов тарифных импульсов, описанных в настоящем Дополнении, и аналогичных сигналов, описанных в Дополнении IX, носит характер прямого соответствия; E/ps(lt=em) отображается непосредственно в am/em, а E/ps(mpb) – в am/mpb, соответственно. В этих сигналах в обоих пакетах используются одни и те же параметры.

**ПРИМЕЧАНИЕ 3.** – В этой Рекомендации предполагается использование только кодеков G.711, все другие кодеки должны рассматриваться как подлежащие изучению в будущем.

**ПРИМЕЧАНИЕ 4.** – Вопросы, связанные с линиями ЦСИС/BRI, подлежат дальнейшему изучению.

#### VIII.2 Архитектура оконечного устройства IPAT

На рисунке VIII.1 представлена эталонная архитектура, используемая в настоящем Дополнении. Оконечное устройство IPAT обеспечивает взаимодействие между сетью проекта IP-Cablecom и местной станцией, являющейся частью сети SCN. В интерфейсе между оконечным устройством IPAT и станцией LE используется подмножество стандарта ETS 300 324, который применим для поддержки услуг SCN для аналоговых телефонов.

В отображении, описанном в настоящем Дополнении, не делается каких-либо предположений относительно внутренней структуры оконечного устройства IPAT; однако предполагается, что он обеспечивает как функции сигнализации, так и функции мультимедийного взаимодействия.

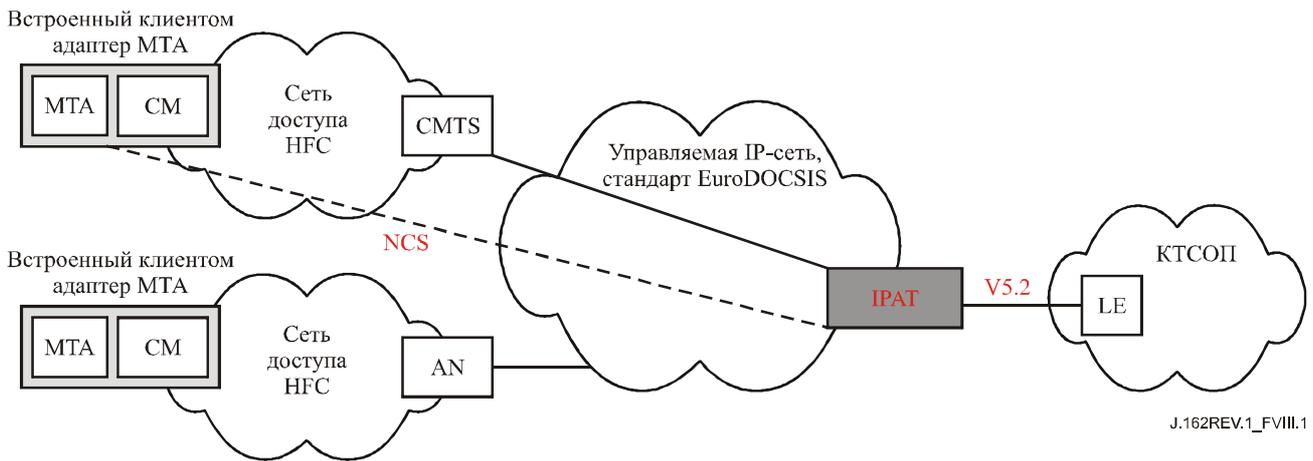


Рисунок VIII.1/J.162 – Эталонная модель, используемая в настоящем Дополнении

### VIII.3 Требования к электрическим и физическим интерфейсам

Здесь предполагается архитектура системы согласно стандарту ETS 300 324, состоящая из местной станции (LE) и оконечного устройства IP-доступа (IPAT), подключенного через интерфейс V5.

Интерфейс V5 может быть интерфейсом со скоростью передачи от 2048 кбит/с до  $16 \times 2048$  кбит/с, как определено в стандартах ETS 300 347-1, ETS 300 166 и ETS 300 167.

Электрические и физические характеристики этого интерфейса должны соответствовать стандарту ETS 300 166 при скорости передачи 2048 кбит/с.

В стандарте ETS 300 166 определены два альтернативных представления интерфейса: симметричная двухпроводная линия и коаксиальная пара. Согласно этим двум альтернативным приложениям интерфейса, представленным на рисунке VIII.1, оператор сети может запросить требуемое представление интерфейса.

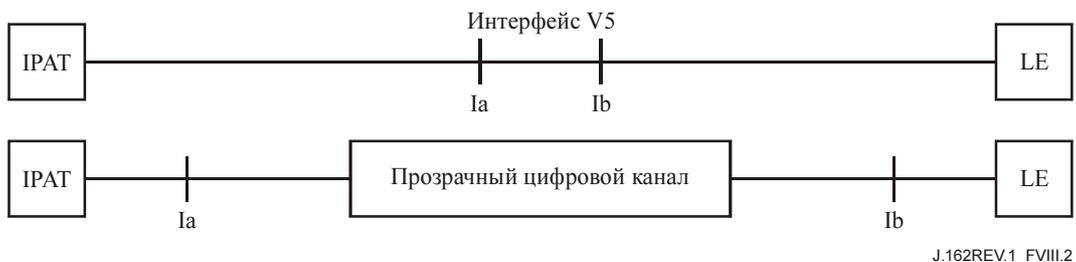


Рисунок VIII.2/J.162 – Альтернативные представления интерфейса V5

В настоящем Дополнении система CMTS расширена, чтобы описать сеть проекта IPCablecom, состоящую из оконечного устройства IP-доступа (IPAT), головного узла кабельных модемов (CMTS), кабельного модема (CM) и адаптера мультимедийного оконечного устройства (MTA) или встроенного адаптера мультимедийного оконечного устройства (E-MTA).

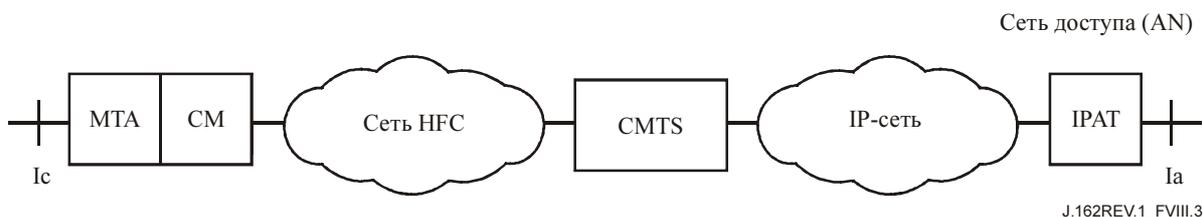


Рисунок VIII.3/J.162 – Сеть доступа

ПРИМЕЧАНИЕ. – Ia=точка интерфейса на стороне сети доступа; Ib=точка интерфейса на стороне станции LE; Ic=точка интерфейса на стороне помещения пользователя.

Данная сеть доступа является синонимичной сети доступа, использующей удаленное цифровое оконечное устройство (RDT) в традиционной архитектуре коммутатора каналов.

Электрические и логические описания IP-сети и сети HFC являются предметом других процессов стандартизации.

В настоящем Дополнении предполагается, что эти сети просто обеспечивают прозрачный цифровой канал, как описано в стандарте ETS 300 324. Это позволяет сосредоточить внимание в настоящем Дополнении на методе обеспечения сигнализации, необходимой, как определено в стандарте ETS 300 324, между станцией LE интерфейса V5 и точкой интерфейса в помещениях пользователя в поддержку требуемых услуг в оконечной точке в помещениях пользователя.

Для запросов тактовых вызывных сигналов в настоящем Дополнении определен расширенный диапазон вызывных тактовых сигналов, где используется синтаксис, подобный синтаксису вызывных тактовых сигналов сигнализации NCS.

Для импульсных и постоянных сигналов в настоящем Дополнении допускается, чтобы оконечное устройство IPAT проекта IPCablecom преобразовывал сообщение протокола для интерфейса V5, полученное от коммутатора для интерфейса V5, в соответствующий запрос сигнала от оконечного устройства IPAT к адаптеру E-MTA, определяющему необходимый сигнал, который должен поступать в оконечную точку в помещениях пользователя (линейная обработка; длительность импульса, период импульса и число повторений, и т. д.). В настоящее Дополнение также включен способ поддержки оконечным устройством IPAT запросов коммутатора в интерфейсе V5 для подтверждений.

#### **VIII.4 Пакет NCS для сообщений протокола сети SCN с интерфейсом V5**

В данном пункте описывается добавление запроса сигнала и запроса события в проекте IPCablecom к предопределенному европейскому линейному пакету, разрабатываемому для сигнализации NCS в европейском проекте IPCablecom.

Эти запросы сигналов и запросы событий отображают соответствующие информационные элементы, содержащиеся в двоичном формате в типе сообщения протокола сети SCN с интерфейсом V5, в формат сигнализации NCS.

ПРИМЕЧАНИЕ. – Значения по умолчанию, приведенные в настоящем Дополнении, предназначены для предоставления поставщикам оборудования значений для начальной партии изделий.

Должны быть предусмотрены положения, разрешающие замену этих значений как часть конфигурации блока или же обеспечение другими значениями согласно требованиям местной администрации.

##### **VIII.4.1 Запрос вызывных тактовых сигналов**

Типы сообщений "Establish" ("установить") или "Signal" ("сигнал") в интерфейсе V5 для "вызывных тактовых сигналов" отображаются в "SignalRequest" ("запрос сигнала") сигнализации NCS:

S: <request code>

Кодом запроса сигнала для европейского вызывного тактового сигнала является **cr(x)**.

ПРИМЕЧАНИЕ. – Определяемый в настоящее время вызывной сигнал NCS для линейного пакета в проекте IPCablecom "rx" описывается равенством  $x = g, s$  или числами 0–7 (десятичными). Некоторые из этих тактовых сигналов являются фиксированными и не могут быть обеспечены согласно руководящим указаниям проекта IPCablecom.

В интерфейсе V5 вызывные тактовые сигналы могут занимать диапазон от 0 до 127; таким образом, код запроса сигнала  $cr(x)$  определяется при  $x = 0, 127$ . В системах с интерфейсом V5 вызывным тактовым сигналом по умолчанию является  $cr(0)$ , и любой из тактовых сигналов может быть однозначным образом обеспечен согласно национальным нормам или требованиям администраций.

##### **VIII.4.1.1 Значения по умолчанию и диапазоны значений для вызывных тактовых сигналов**

Адаптер MTA допускает обеспечение значений вызывных тактовых сигналов (от 0 до 127), соответствующих отображению вызывных тактовых сигналов станции LE согласно национальным нормам или требованиям местных администраций.

В таблице VIII.1 приведены значения по умолчанию вызывных тактовых сигналов. Все временные параметры выражены в миллисекундах.

Обеспечение в диапазоне от 0 до 5000 мс должно выполняться с шагом в 50 мс:

**Таблица VIII.1/J.162 – Значения по умолчанию вызывных тактовых сигналов**

cr(x)	t1 – сигнал	t2 – свободно	t3 – сигнал	t4 – свободно	t5 – сигнал	t6 – свободно
0	1000	4000	1000	4000	1000	4000
1	1000	500	1000	3500	1000	3500
2	500	500	500	500	1000	3000
3	500	500	1000	500	500	3000
4	1000	500	500	4000		
5						
6						
7						
8						
...						
127						

#### VIII.4.2 Запрос импульсного сигнала

Запрос "импульсный сигнал" в типе сообщения "установить" или "сигнал" интерфейса V5 отображает запрос импульсного сигнала в запрос сигнала NCS.

Кодом запроса сигнала для импульсного сигнала является **ps**.

Параметрами для этого запроса сигнала являются:

- **lt** обозначает применяемую линейную обработку (соответствует кодированию типа импульса в интерфейсе V5);
- **pd** обозначает длительность импульса (длина одного импульса);
- **pr** обозначает интервал повторения импульса для всех импульсов.

Значения **pd** и **pr** являются необязательными. Если значения не заданы, то адаптер МТА должен использовать ранее обеспеченные значения в блоке MIB адаптера МТА согласно коду типа для линейной обработки/типа импульса (**lt**).

В дополнение к этим параметрам запрос сигнала может быть применен со следующими параметрами:

- **rep** обозначает число импульсов (повторений);
- **rpe** обозначает число импульсов между сообщениями о тарифных импульсах (необязательный параметр, только для сигнала **em**).

Большинство запросов импульсных сигналов являются по существу сигналами выдержки времени (TO), значение выдержки времени для которых может определяться следующим образом:

$$t_o = pr \times rep$$

Для окончного устройства IPAT не требуется иметь параметр выдержки времени в запросе сигнала, если для запроса сигнала достаточно значение выдержки времени по умолчанию. Это значение по умолчанию должно обеспечиваться как в адаптере МТА, так и в окончном устройстве IPAT.

Если произведение  $pr \times rep$  существенно меньше 180 секунд, то окончному устройству IPAT СЛЕДУЕТ включить значение выдержки времени; если произведение  $pr \times rep$  больше 180 секунд, то окончное устройство IPAT ДОЛЖЕН включить значение выдержки времени.

Сигналы "активизировать генерирование тарифного импульса" (**em**) и "генерирование пачки тарифных импульсов" (**mpb**) определяются как сигналы "включено/выключено" (OO) или короткие (BR) сигналы, соответственно. Число импульсов (**rep**) неприменимо к запросу сигнала **em**. Точнее, сигнал **em** может содержать только параметр "счетчик импульсов сообщения" (**rpe**). Параметр "число импульсов" требуется для запроса сигнала **mpb**.

### VIII.4.2.1 Кодирование линейной обработки

В таблице VIII.2 приводится описание кодирования для линейных обработок, которые могут применяться, наряду с применимостью параметров и типов сигналов. Параметры могут быть необязательными (O), обязательными (M) или запрещенными (F).

Таблица VIII.2/J.162 – Кодирование линейной обработки

Код It	Описание	Тип сигнала	pd	pr	гер (примечание)	грс
ir	Начальный вызывной сигнал	TO	O	O	O	F
lc	Импульсный цикл замкнут	TO	O	O	O	F
lo	Импульсный цикл разомкнут	TO	O	O	O	F
em	(Активизировать) генерирование тарифных импульсов	OO	F	O	F	O
mpb	Генерирование пачки тарифных импульсов	BR	O	O	O	F
nb	Нет батарейного импульса	TO	O	O	O	F
np	Импульс нормальной полярности	TO	O	O	O	F
rb	Подавленный батарейный импульс	TO	O	O	O	F
rp	Импульс обратной полярности	TO	O	O	O	F

ПРИМЕЧАНИЕ. – Параметр "гер" является ОБЯЗАТЕЛЬНЫМ, если значение предоставлено интерфейсом V5 станции LE. Значение НЕОБЯЗАТЕЛЬНЫЙ в этом поле означает использование значений по умолчанию (см. таблицу VIII.3) при поддержке архитектур агента вызова или коммутатора с распределенным управлением (Softswitch).

### VIII.4.2.2 Значения по умолчанию и диапазоны параметров для линейных обработок

В таблице VIII.3 приводится описание значений по умолчанию и диапазонов параметров для линейных обработок, представленных в таблице VIII.2. Значения для параметров времени заданы в миллисекундах.

Таблица VIII.3/J.162 – Значения по умолчанию и диапазоны параметров для линейных обработок

Код It	Описание	Частота (допуск)	Амплитуда (минимум–максимум, шаги)	pd (минимум–максимум, шаги)	pr (минимум–максимум, шаги)	гер (минимум–максимум, шаги)
ir	Начальный вызывной сигнал	25 Гц (+1 Гц)	Полная	200 (0–5000, 50)	200 (0–5000, 50)	1 (1–5, 1)
lc	Импульсный цикл замкнут	Нулевая	Нулевая	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
lo	Импульсный цикл разомкнут	Нулевая	Нулевая	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
em	(Активизировать) генерирование тарифных импульсов	16 кГц	–13,5 дБм <sup>a)</sup> (от –25 до +15,2 дБ)	150 (0–5000, 10)	1000 (0–5000, 10)	Нулевое
mpb	Генерирование пачки тарифных импульсов	16 кГц	–13,5 дБм <sup>a)</sup> (от –25 до +15,2 дБ)	150 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
nb	Нет батарейного импульса	Нулевая	0	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
np	Импульс нормальной полярности	Нулевая	1	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)

**Таблица VIII.3/J.162 – Значения по умолчанию и диапазоны параметров для линейных обработок**

<b>Код lt</b>	<b>Описание</b>	<b>Частота (допуск)</b>	<b>Амплитуда (минимум–максимум, шаги)</b>	<b>pd (минимум–максимум, шаги)</b>	<b>pr (минимум–максимум, шаги)</b>	<b>per (минимум–максимум, шаги)</b>
<b>rb</b>	Подавленный батарейный импульс	Нулевая	1	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
<b>rp</b>	Импульс обратной полярности	Нулевая	0	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
a) Амплитуда тарифного импульса определяется в дБм через нагруженные выводы a–b при полном сопротивлении эталонной нагрузки согласно национальным нормам.						

#### **VIII.4.2.3 Запрашиваемые события**

Приведенные ниже события могут быть запрошены для импульсных сигналов путем их включения в запрашиваемые события (R: список параметров в запросе на уведомление):

- **oc** означает, что должно быть уведомление об окончании операции;
- **of** означает, что должно быть уведомление об ошибке операции;
- **pc** означает, что должно быть уведомление об окончании импульса.

#### **VIII.4.2.4 Кодирование импульсов**

Оконечное устройство IPAT должен преобразовывать кодирование перечислимых типов импульсов и продолжительности импульсов интерфейса V5 в типы линейной обработки и продолжительности импульсов NCS в миллисекундах согласно таблицам обеспечения, как определено станцией LE или местной администрацией.

##### **VIII.4.2.4.1 Кодирование продолжительности импульсов**

Длительность импульса определяется в миллисекундах с использованием параметра **pd**. Например, 200-миллисекундный импульс определяется как:

`pd=200`

Длительность импульса является *необязательной*. Если длительность импульса не предоставляется запрашивающим объектом, то адаптеру МТА СЛЕДУЕТ применить обеспеченное или внутренним образом заданное по умолчанию значение на основе параметра линейной обработки (lt) (см. таблицу VIII.3).

##### **VIII.4.2.4.2 Кодирование импульсного периода**

Импульсный период определяется в миллисекундах с использованием параметра **pr**. Например, 1-секундный период задается как:

`pr=1000`

Так, например, 50%-ный рабочий цикл и 1-секундный периодический импульс задаются следующим образом:

`pd=500, pr=1000`

Импульсный период является *необязательным*. Если он не предоставляется запрашивающим объектом, то адаптеру МТА СЛЕДУЕТ применять обеспеченное или внутренним образом заданное по умолчанию значение на основе параметра линейной обработки (lt) (см. таблицу VIII.3).

##### **VIII.4.2.5 Кодирование события "окончание импульса"**

Адаптер МТА передает сообщение о событии "окончание импульса" при получении запроса об этом в первом запросе сигнала от окончательного устройства IPAT, когда каждый запрошенный импульс

заканчивается. Об этом событии для каждого законченного импульса уведомляют в течение запроса сигнала без дополнительных запросов на уведомление от оконечного устройства IPAT. Обнаружение этого события не влияет на продолжающееся использование импульсов со стороны адаптера МТА.

Кодом запроса события "окончание импульса" является **рс**, который включается в запрос сигнала подобно коду события "операция завершена (**ос**)".

#### **VIII.4.2.6 Кодирование отчета о тарифных импульсах**

Адаптер МТА сообщает о событии "отчет о тарифных импульсах", когда оно запрошено в запросе сигнала "генерирование тарифных импульсов" с ненулевым параметром "счетчик импульсов отчета" (**гpc**). Уведомление об этом событии происходит каждый раз, когда значение счетчика тарифных импульсов достигает значения счетчика импульсов отчета. В результате генерирования события счетчик тарифных импульсов адаптера МТА сбрасывается до нуля. При подсчете не учитываются импульсы, генерируемые по запросам сигнала "генерирование пачки тарифных импульсов" (**mpb**). Генерирование события не влияет на продолжающееся генерирование тарифных импульсов и на последующее уведомление о событии "отчет о тарифных импульсах". Оконечному устройству IPAT не нужно посылать новый запрос на уведомление.

Кодом события для сообщения о тарифных импульсах является **mpg**. Это уведомление включает счетчик. Пример:

O: mpg(10)

#### **VIII.4.2.7 Индикатор подавления в интерфейсе V5**

Индикатор подавления в интерфейсе V5 используется как в информационном элементе (IE) "импульсный сигнал", так и в информационном элементе "активизация тарификации". Он позволяет станции LE указать сети доступа на то, должен ли подавляться текущий импульсный сигнал.

Индикатор подавления используется для указания на то, должно ли быть остановлено генерирование импульсов в сети, если изменяются линейные условия, если от станции LE принято новое сообщение СИГНАЛ или если происходит и то, и другое. Это особенно важно для тарифных импульсов в некоторых сетях, где после отбоя вызова тарифные импульсы не передаются; индикатор подавления можно было бы использовать для подавления тарифных импульсов после отбоя вызова.

В других сетях важно, чтобы тарифные импульсы выдавались независимо от изменения состояния линии либо из-за сообщений от станции LE, либо из-за изменений в оборудовании ТЕ.

Индикатор подавления кодируется следующим образом:

- 00 нет подавления;
- 01 подавление допускается заранее определенным сообщением СИГНАЛ согласно V5.1 от станции LE;
- 10 подавление допускается заранее определенным линейным сигналом от оборудования ТЕ;
- 11 подавление допускается заранее определенным сообщением СИГНАЛ согласно V5.1 от станции LE или заранее определенным линейным сигналом от оборудования ТЕ.

Индикация подавления сигнала не отображается эффективным образом на протокол сигнализации NCS. Например, для применения запроса сигнала с индикацией "нет подавления" этот сигнал должен быть определен как "короткий" сигнал; для применения сигнала с индикацией "подавление допускается заранее определенным сигналом от оборудования ТЕ" требуется, чтобы сигнал был определен как сигнал "выдержки времени". В целях обеспечения взаимодействия интерфейс V5 – сигнализация NCS принимаются правила протокола сигнализации NCS, и сигналы определяются на основе предположений о нормальном использовании.

Чтобы разрешить эти противоречия с сигнализацией NCS, оконечное устройство IPAT должен "установить сопряжение" протокола V5 с сигнализацией NCS путем принятия индикации подавления в интерфейсе V5 и выполнения последующих действий согласно соответствующему множеству сообщений сигнализации NCS для достижения желаемого эффекта.

#### **VIII.4.2.7.1 Индикация "нет подавления"**

По получении кода "00" V5 оконечное устройство IPAT должен генерировать для адаптера МТА соответствующее сообщение NCS о линейной обработке. Адаптер МТА должен выполнить соответствующую линейную обработку, как определено в настоящем Дополнении, независимо от изменений в состоянии линии или дополнительных сигнальных сообщений от LE–IPAT.

#### **VIII.4.2.7.2 Индикация "подавление заранее определенным сообщением СИГНАЛ в интерфейсе V5"**

В этом случае оконечное устройство IPAT должен быть заранее обеспечен соответствующим сообщением СИГНАЛ интерфейса V5 (например, состояние "телефонная трубка положена" на дальнем конце).

По получении кода "01" V5 оконечное устройство IPAT должен приступить к непрерывному контролю заранее обеспеченного сообщения СИГНАЛ интерфейса V5 .

Адаптер МТА должен выполнить соответствующую линейную обработку, как определено в настоящем Дополнении.

По получении заранее обеспеченного сообщения СИГНАЛ оконечное устройство IPAT должен передать адаптеру МТА соответствующее сообщение аннулирования импульсного сигнала (см. п. VIII.4.5).

Адаптер МТА должен ответить на это сообщение аннулирования импульсного сигнала так, как определено в настоящем Дополнении.

#### **VIII.4.2.7.3 Индикация "подавление заранее определенным линейным сигналом от оборудования ТЕ"**

В этом случае оконечное устройство IPAT должен быть заранее обеспечен соответствующим сигнальным сообщением линейной обработки сигнализации (например, "телефонная трубка положена").

По получении кода "10" V5 оконечное устройство IPAT должен приступить к непрерывному контролю заранее обеспеченного сигнального сообщения линейной обработки NCS от адаптера МТА.

Адаптер МТА должен выполнить согласно этому сообщению соответствующую линейную обработку, как определено протоколами NCS (например, сигнал "телефонная трубка положена").

По получении заранее обеспеченного сообщения линейной обработки NCS оконечное устройство IPAT должен выдать адаптеру МТА соответствующее сообщение аннулирования импульсного сигнала (см. п. VIII.4.5).

Адаптер МТА должен ответить на это сообщение аннулирования импульсного сигнала так, как определено в настоящем Дополнении.

#### **VIII.4.2.7.4 Индикация "подавление заранее определенным сообщением СИГНАЛ в интерфейсе V5 от станции LE или при заранее определенным линейным сигналом от оборудования ТЕ"**

В этом случае оконечное устройство IPAT должен быть заранее обеспечен соответствующим сообщением СИГНАЛ в интерфейсе V5 **И** соответствующим сигнальным сообщением линейной обработки NCS (например, сигнал "телефонная трубка положена" на дальнем конце" **И** "телефонная трубка положена" на станции ТЕ).

По получении кода "11" V5 оконечное устройство IPAT должен приступить к непрерывному контролю заранее обеспеченного сообщения СИГНАЛ интерфейса V5 и заранее обеспеченного сигнального сообщения линейной обработки NCS от адаптера МТА.

При наличии соответствующего сообщения линейной обработки адаптер МТА должен выполнить линейную обработку согласно этому сообщению, как определено протоколами NCS (например, "телефонная трубка снята").

По получении от адаптера МТА заранее обеспеченного сообщения СИГНАЛ интерфейса V5 **ИЛИ** сообщения линейной обработки NCS оконечное устройство IPAT должен выдать адаптеру МТА соответствующее сообщение аннулирования импульсного сигнала (см. п. VIII.4.5).

Адаптер МТА должен ответить на это сообщение аннулирования импульсного сигнала так, как определено в настоящем Дополнении.

#### **VIII.4.2.8 Индикатор повторений**

Индикатор повторений используется только в информационном элементе "активизация тарификации" согласно V5. Он посылается в направлении от станции LE к сети доступа со счетчиком импульсов отчета, чтобы выдать сети доступа команду на продолжение или прекращение применения автоматических тарифных импульсов, когда было использовано число импульсов, определяемое **счетчиком импульсов отчета**.

Кодирование индикатора повторений:

- 00 прекращение применения импульсов после того, как было использовано число импульсов, определяемое счетчиком импульсов отчета;
- 11 продолжение применения импульсов с той же частотой, пока вызов не будет разъединен или пока не будут получены новые команды от станции LE;
- 01 зарезервирован для использования в Европе;
- 10 зарезервирован для использования в Европе.

В режиме линейной обработки сигнала **em** по умолчанию предусматривается сигнал, применяемый как сигнал "включено/выключено", пока не будет разъединен оконечным устройством IPAT. Оконечное устройство IPAT может отключить импульсы, как только будет достигнуто число импульсов в счетчике импульсов отчета, путем передачи встроенного запроса на уведомление, чтобы выключить сигнал **em** (см. п. VIII.4.5).

#### **VIII.4.3 Кодирование повторений импульсов**

Оконечное устройство IPAT отображает счетчик повторений импульсов интерфейса V5 непосредственно в существующий параметр повторений NCS (**rep**).

Этот параметр должен быть предоставлен согласно таблице VIII.2. Для повторений импульсов не существует значения по умолчанию.

**ПРИМЕЧАНИЕ.** – Согласно руководству по интерфейсу V5 значение "0" для параметра **rep** является недопустимым. Если оконечное устройство IPAT принимает от станции LE интерфейса V5 запрос со значением "**rep** = 0" или с пропущенным значением **rep**, оконечное устройство должен подставить значение параметра "**rep** = 1".

В информационном элементе "импульсный сигнал интерфейса V5" поле "число импульсов" – это 5-битовое поле. Диапазон допустимых значений составляет от 1 до 31. В информационном элементе "активизация тарификации" сочетание полей "индикатор повторений = 00" и "счетчик импульсов отчета" также учитывает описание ограниченного "числа импульсов". Счетчик импульсов отчета занимает 12-битовое поле, что соответствует допустимому диапазону от 1 до 4095. Хотя для интерфейса V5 значение повторений импульсов может быть только в диапазоне от 1 до 31, повторения импульсов могут быть заданы в полном диапазоне от 1 до 4095.

#### **VIII.4.4 Использование параметров**

Все параметры, описанные для запроса импульсных сигналов, относятся ко всем описанным линейным обработкам.

Оконечное устройство IPAT должен предоставлять значения для продолжительности импульсов, интервала повторений импульсов и числа повторений.

Для учета национальных различий в тарифных импульсах адаптер МТА обеспечивается значением частоты и амплитуды, поскольку этой информации нет в сообщении от интерфейса V5. Оконечное устройство IPAT должен по типу тарифа в сообщении от интерфейса V5 определить интервал повторения импульса и послать адаптеру МТА в запросе сигнала время интервала (в миллисекундах).

Информационный элемент "активизация тарификации" интерфейса V5-2000 имеет поле "тип тарифа". Это перечислимый тип. Оконечное устройство IPAT должен преобразовывать различные перечислимые значения в соответствующие значения в миллисекундах на основе их обеспечения, зависящего от местной администрации.

Для выдачи фиксированного числа импульсов в абонентскую линию оконечное устройство IPAT может использовать интервал повторения импульса и параметр "повторение сигнала".

#### VIII.4.5 Аннулирование импульсного сигнала

Большинство импульсных сигналов, будучи сигналами выдержки времени, завершаются при обнаружении любого запрашиваемого события, *за исключением окончания импульса (ps)*.

Кроме того, станция LE путем передачи пустого запроса сигнала может в любое время приводить все активные импульсные сигналы к завершению.

Поскольку станция LE может одновременно посылать несколько импульсных сигналов в абонентскую линию (например, генерируется тарифный импульс и применяется линейная обработка другого импульса), то оконечное устройство IPAT может завершать обработку в диалоговом/автономном режиме характерным для обработки синтаксисом команды. Примером окончания обработки поступившего тарифного импульса может служить:

S: E/ps(em(-))

#### VIII.4.6 Событие завершения импульса

Адаптер МТА сообщает оконечному устройству IPAT о событии завершения импульса, когда завершается каждый запрашиваемый импульс.

Кодом запроса события завершения импульса является код **ps**.

#### VIII.4.7 Событие "сбой импульсного сигнала"

Адаптер МТА сообщает оконечному устройству IPAT о событии "сбой импульсного сигнала" при неудачном завершении любого запроса импульсного сигнала, если в список запрашиваемых событий включено событие ошибки операции "**of**". Запрос импульсного сигнала может оказаться неудачным по любой причине, по которой может оказаться неудачным всякий другой запрос сигнала.

#### VIII.4.8 Запрос постоянного сигнала

Запрос постоянного сигнала "установить" интерфейса V5 отображает запрос постоянного сигнала в запрос сигнала NCS.

Кодом запроса сигнала для постоянного сигнала является **ss**.

Параметром для этого запроса сигнала является:

- **It** обозначает применяемую линейную обработку (соответствует кодированию типа постоянного сигнала в интерфейсе V5).

Эта обработка сохраняется до тех пор, пока станция LE интерфейса V5 не укажет на новую обработку.

##### VIII.4.8.1 Кодирование линейной обработки

Виды линейной обработки кодируются с использованием кодовых слов, как указано в таблице VIII.4.

Таблица VIII.4/J.162 – Кодирование запроса постоянного сигнала

Код It	Описание
<b>fb</b>	нормальный (полный) батарейный
<b>lc</b>	цикл замкнут
<b>lo</b>	цикл разомкнут
<b>nb</b>	Небатарейный
<b>np</b>	нормальная полярность
<b>rb</b>	подавленный батарейный
<b>rp</b>	обратная полярность

##### VIII.4.8.2 Обеспечение линейной обработки

Обеспечение не требуется в том смысле, что эти состояния линии не имеют количественных значений (время, частота или амплитуда).

#### **VIII.4.9 Генерирование тарифных импульсов**

По получении запроса сигнала "активизировать генерирование тарифных импульсов" **ps(lt=em(+))** адаптер МТА должен подать первый тарифный импульс непосредственно на окончание, а затем применять последующие тарифные импульсы через интервалы, определяемые значением параметра "интервал повторения импульса" **pr**, если он предоставлен в запросе сигнала, или обеспеченным значением.

Адаптер МТА должен продолжать генерировать тарифные импульсы до тех пор, пока не получит запрос сигнала "прекратить генерирование тарифных импульсов" **ps(lt=em(-))** или пустой список запросов сигналов.

Запрос сигнала "пачка тарифных импульсов" **ps(lt=mpb)** может быть включен в запрос сигнала, который также активизирует генерирование тарифных импульсов, например, для того чтобы произвести начальное начисление платы за вызов. В этом случае адаптер МТА должен полностью использовать пачку тарифных импульсов для конечной точки и затем начать генерирование нормальных тарифных импульсов.

Поскольку сигнал "пачка тарифных импульсов" является типом короткого сигнала, то применяются все импульсы, определенные для запроса (**rep=n**), даже если во время передачи этой пачки тарифных импульсов абонент положит трубку.

Запрос сигнала "пачка тарифных импульсов" может иметь место в процессе вызова, например, чтобы учесть действия абонента, подлежащие оплате. Когда это происходит, адаптер МТА должен приостановить нормальное генерирование тарифных импульсов и применить запрос сигнала "пачка тарифных импульсов". Затем адаптер МТА должен возобновить нормальное генерирование тарифных импульсов, не требуя от оконечного устройства IPAT нового запроса "активизировать генерирование тарифных импульсов". Оконечное устройство IPAT должен учитывать все нормальные тарифные импульсы, пропущенные при генерировании пачки импульсов, путем включения пропущенных импульсов в счетчик пачек импульсов.

Оконечное устройство IPAT может необязательно использовать параметр "счетчик импульсов отчета" (**grc**) с запросом сигнала "активизировать генерирование тарифных импульсов" (**em**). Если этот параметр ненулевой (**grc=n**, где  $n=1-x$ ), адаптер МТА генерирует отчет о тарифных импульсах в виде уведомлений каждый раз, когда его счетчик импульсов достигает значения **grc**. При генерировании уведомления о событии счетчик **grc** сбрасывается в исходное состояние, так что отчет будет генерироваться каждый раз, когда достигается значение "n" для **grc**. В этот счетчик не входят какие-либо тарифные импульсы, генерируемые по запросам сигнала "пачка тарифных импульсов" (**mpb**).

#### **VIII.5 Обеспечение конфигураций**

##### **VIII.5.1 Адаптер МТА**

Адаптер МТА должен обеспечиваться электрическими параметрами для каждой линейной обработки. Когда это целесообразно, эти параметры включают амплитуду, частоту, минимальные значения ширины импульсов и максимальную частоту повторений (минимальное время между импульсами). Подробная информация содержится в таблицах VIII.1–VIII.3. Эти параметры должны использоваться, если только в сообщениях интерфейса V5 не будут предоставлены значения, характерные для линейной обработки.

##### **VIII.5.2 Оконечное устройство IPAT**

Оконечное устройство IPAT должен быть обеспечен отображением кодирований продолжительности импульсов и типа импульсов интерфейса V5 в тип импульсов NCS и время продолжительности импульсов в миллисекундах. Это обеспечение должно соответствовать обеспечению станции LE и руководящим указаниям местной администрации.

#### **VIII.6 Поддержка европейского линейного пакета**

##### **VIII.6.1 Контроль сигнализации NCS**

Команда "контроль конечной точки" (AUEP) сигнализации NCS позволяет адаптеру МТА передавать сообщения о сигналах, которые он поддерживает.

В ответ на команду AUEP адаптер МТА, поддерживающий любой из запросов сигнализации, перечисленных в настоящем Дополнении, должен сообщить о поддержке данного "европейского" пакета (обозначенного кодом "E").

Примером обмена информацией при контроле служит:

```
AUEP 1232 aaln/1@rgw.mso.net
F: A
```

Адаптер МТА отвечает:

```
200 1232 OK
A: a:PCMU,
p:30-90,
v:L;E,
m:sendonly;recvonly;sendrecv;inactive,
DQ-GI,SC-ST, SC-RTP: 00/51;03
```

Важной строкой для пакетов является "v:L;E", которая указывает на поддержку линейного пакета NCS (L) и европейского линейного пакета (E).

### **VIII.6.2 Неподдерживаемые сигналы – заявление о соответствии реализации протоколу (PICS)**

Это индикация ограничений платформы устройства (аппаратуры или программного обеспечения), а не состояния ошибки.

Поставщики изделий должны отражать все неподдерживаемые сигналы, перечисленные в настоящем Дополнении, в заявлении PICS на изделие.

В сигнализации NCS предоставляется средство обмена сообщениями, при котором устройство должно возвращать ответ "неподдерживаемый сигнал" (код 513), если оно не может поддержать запрашиваемый тип сигнала.

Пример 1:

Система CMS → адаптер МТА (запрос пачки тарифных импульсов):

```
RQNT 9915 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 2255
S: E/ps(lt=mpb, pd=500, pr=1000, rep=5)
R: oc, hu, hf
```

Адаптер МТА → система CMS (отклонение запроса):

```
513 9915 Unsupported Signal in Signal Request
```

Пример 2:

Система CMS → адаптер МТА (запрос активизации тарификации, использование обеспечиваемых по умолчанию значений):

```
RQNT 9915 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 2255
S: E/ps(lt=em(+))
R: E/ps, hu, hf
```

Адаптер МТА → система CMS (отклонение запроса):

```
513 9915 Unsupported Signal in Signal Request
```

## **VIII.7 Примеры последовательностей операций при вызовах**

### **VIII.7.1 Вызывные тактовые сигналы**

#### **VIII.7.1.1 Последовательность операций при вызывных тактовых сигналах для базового вызывного тактового сигнала**

Эта последовательность операций иллюстрирует запрос на применение простого вызывного тактового сигнала.

- 1) Станция LE интерфейса V5 включает в сообщение к оконечному устройству IPAT запрос импульсного вызывного тактового сигнала.
- 2) Оконечное устройство IPAT преобразует двоичный закодированный вызывной тактовый сигнал в десятичное значение в диапазоне от 0 до 127.
- 3) Предполагая, что значение вызывного тактового сигнала преобразовано в десятичное число "0":

```
RQNT 500 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
S: E/cr(0)
```

- 4) Адаптер МТА подтверждает запрос сигнала.  
200 500 ОК
- 5) Адаптер МТА отыскивает в своей таблице обеспечения вызова определение  $cr(0)$  вызывной частоты и вызывной тактовой сигнал и посылает его к выводам a–b для имеющейся в адаптере МТА линии aaln/1.  
Этот тактовый сигнал продолжает передаваться до тех пор, пока адаптер МТА не обнаружит сигнал "телефонная трубка снята", когда он начинает выполнять нормальную последовательность действий по установлению соединения в системе сигнализации NCS, либо пока оконечное устройство IPAT не передаст сообщение разъединения.

### **VIII.7.1.2 Передача вызывных тактовых сигналов – пакет коротких вызывных сигналов, сопровождаемый вызывным тактовым сигналом**

Эта последовательность операций при вызове показывает использование типа импульсного сигнала "начальный вызывной сигнал", сопровождаемого вызывным тактовым сигналом, для обеспечения "пакета коротких вызывных сигналов", сопровождаемых вызывным тактовым сигналом.

- 1) Станция LE интерфейса V5 посылает в сообщении к оконечному устройству IPAT запрос типа импульсного сигнала "начальный вызывной сигнал" вместе с типом продолжительности импульса.
- 2) Оконечное устройство IPAT преобразует тип "начальный вызывной сигнал" в тип  $ir$  параметра  $lt$  сигнализации NCS со значением продолжительности импульса и запрашивает уведомление о завершении операции.

```
RQNT 510 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 000691
S: E/ps (lt=ir, pd=200, rep=1)
R: oc
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 691 ОК
- 4) Адаптер МТА отыскивает в своей таблице обеспечения вызова определение  $ir$  начальной вызывной частоты и длительность начального вызывного сигнала ( $pd=200$  приводит к пачке вызывных сигналов в 200 мс) и посылает эту информацию на выводы a–b имеющейся в адаптере МТА линии aaln/1.
- 5) По завершении начального вызывного сигнала адаптер МТА отвечает сообщением "операция завершена".

```
NTFY 1298 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 691
O: oc(E/ps(ir))
```

Следует отметить, что здесь предполагается наличие "европейского линейного пакета", обозначенного именем "E". Имя пакета может быть опущено, если этот пакет является пакетом по умолчанию.

- 6) Оконечное устройство IPAT сигнализирует станции LE интерфейса V5 о том, что импульс завершен.
- 7) Станция LE интерфейса V5 включает в сообщение к оконечному устройству IPAT запрос импульсного вызывного тактового сигнала.
- 8) Оконечное устройство IPAT преобразует двоичный закодированный вызывной тактовый сигнал в десятичное значение в диапазоне от 0 до 127.
- 9) Предполагая, что значение вызывного тактового сигнала преобразовано в десятичное число "0":

```
RQNT 520 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 699
S: E/cr(0)
```

- 10) Адаптер МТА подтверждает запрос сигнала.  
200 520 ОК
- 11) Адаптер МТА отыскивает в своей таблице обеспечения вызова определение  $cr(0)$  вызывной частоты и вызывной тактовой сигнал и посылает его к выводам a–b для имеющейся в адаптере МТА линии aalt/1.  
Этот тактовый сигнал продолжает передаваться до тех пор, пока адаптер МТА не обнаружит сигнал "телефонная трубка снята", когда он начинает выполнять нормальную последовательность действий по установлению соединения в системе сигнализации NCS, либо пока оконечное устройство IPAT не передаст сообщение разъединения.

### VIII.7.1.3 Передача вызывных тактовых сигналов – пакет коротких вызывных сигналов, сопровождаемый данными о состоянии "телефонная трубка положена", а затем вызывным тактовым сигналом

Эта последовательность операций иллюстрирует передачу данных о состоянии "телефонная трубка положена", связанную с вызывными сигналами (идентификатор CLID).

Пакет вызывных сигналов предшествует тональным сигналам ФМн, генерируемым станцией LE интерфейса V5, за которыми следует использование вызывного тактового сигнала.

1) Станция LE интерфейса V5 посылает в сообщении к оконечному устройству IPAT запрос типа импульсного сигнала "начальный вызывной сигнал" вместе с типом продолжительности импульсов.

2) Оконечное устройство IPAT преобразует тип "начальный вызывной сигнал" в тип *ir* параметра *lt* сигнализации NCS со значением продолжительности импульса и запрашивает уведомление о завершении операции.

```
RQNT 530 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 777
S: E/ps (lt=ir, pd=200, rep=1)
R: oc
```

3) Адаптер МТА подтверждает запрос сигнала.

```
200 530 OK
```

4) Адаптер МТА отыскивает в своей таблице обеспечения вызова определение *ir* начальной вызывной частоты и длительность начального вызывного сигнала (*pd=200* приводит к пачке вызывных импульсов в 200 мс) и посылает эту информацию на выходы *a-b* имеющейся в адаптере МТА линии *aaIn/1*.

5) По завершении начального вызывного сигнала адаптер МТА отвечает сообщением "операция завершена".

```
NTFY 1298 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 777
O: oc(E/ps(ir))
```

Следует отметить, что здесь предполагается наличие "европейского линейного пакета", обозначенного именем "E". Имя пакета может быть опущено, если этот пакет является пакетом по умолчанию.

6) Оконечное устройство IPAT сигнализирует станции LE интерфейса V5, что импульс завершен.

7) Станция LE интерфейса V5 затем генерирует для окончания линии *aaIn/1* внутриполосные тональные сигналы ФМн.

8) Адаптер МТА передает внутриполосные тональные сигналы ФМн аналоговой линии POTS *aaIn/1*.

9) Станция LE интерфейса V5 осуществляет задержку в 200 мс от конца тонального сигнала ФМн (чтобы удовлетворять минимальным требованиям стандарта ETSI EN 300 659-1) и затем генерирует запрос импульсного тактового вызывного сигнала в сообщении к оконечному устройству IPAT.

10) Оконечное устройство IPAT преобразует двоичный закодированный вызывной тактовый сигнал в десятичное значение между 0 и 127.

11) Предполагая, что значение вызывного тактового сигнала преобразовано в десятичное число "0":

```
RQNT 540 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 778
S: E/cr(0)
```

12) Адаптер МТА подтверждает запрос сигнала.

```
200 540 OK
```

13) Адаптер МТА отыскивает в своей таблице обеспечения вызова определение *cr(0)* вызывной частоты и вызывной тактовый сигнал и посылает его на выходы *a-b* имеющейся в адаптере МТА линии *aaIn/1.α*

Этот тактовый сигнал продолжает передаваться до тех пор, пока адаптер МТА не обнаружит сигнал "телефонная трубка снята", когда он начинает выполнять нормальную

последовательность действий по установлению соединения сигнализации NCS, либо пока окончное устройство IPAT не передаст сообщение разъединения.

### VIII.7.2 Запрос импульсного сигнала

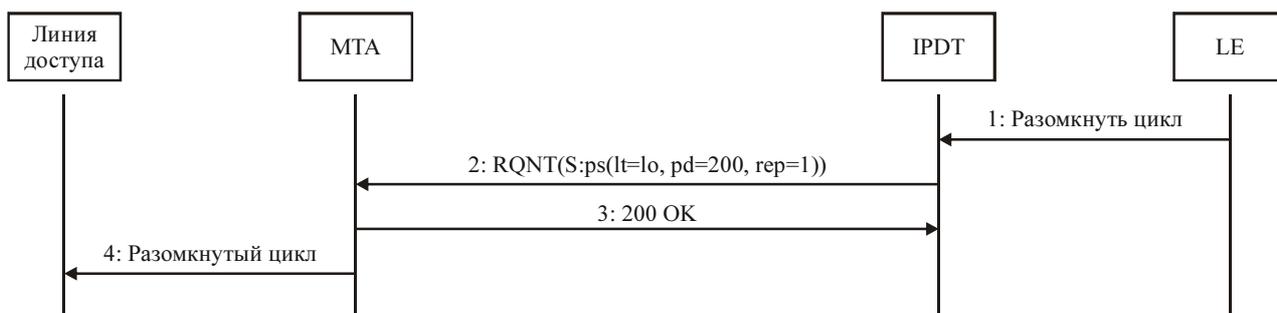
#### VIII.7.2.1 Запрос импульсного сигнала для одного импульса размыкания цикла

- 1) Станция LE интерфейса V5 включает в сообщении к окончному устройству IPAT запрос импульсного сигнала размыкания цикла.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и длительность импульса из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS.

```
RQNT 525 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0  
X: 795  
S: E/ps(lt=1o, pd=200, rep=1)
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 525 ОК
- 4) Адаптер МТА применяет 200-миллисекундный разомкнутый цикл к абонентской линии доступа.

Рисунок VIII.4/J.162 – Запрос импульсного сигнала



J.162REV.1\_FVIII.4

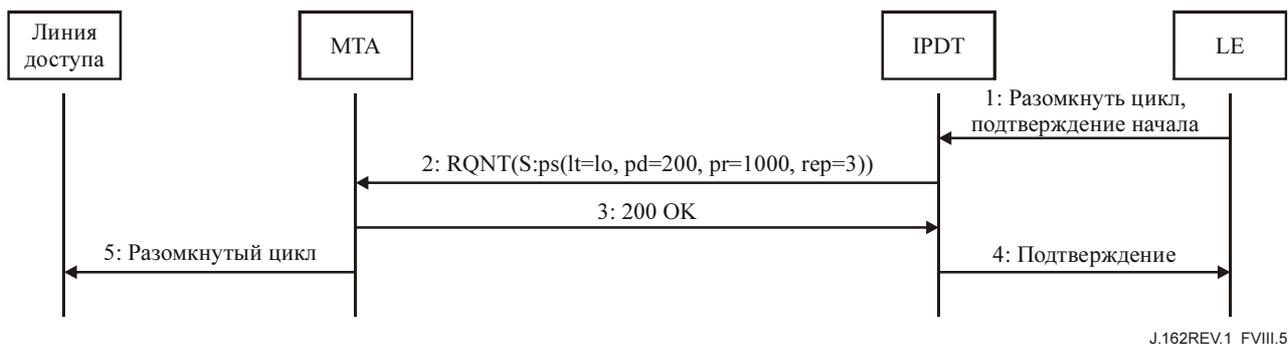
#### VIII.7.2.2 Импульсный сигнал с подтверждением начала

Эта последовательность операций для вызова иллюстрирует запрос импульсного сигнала с несколькими импульсами, при котором коммутатор запросил подтверждение при начале применения сигнала к абонентской линии доступа.

- 1) Станция LE интерфейса V5 запрашивает разомкнутый цикл с несколькими импульсами и подтверждением начала их применения.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку, длительность импульса и импульсный период из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульса, предоставляемое станцией LE интерфейса V5. Оконечное устройство IPAT должен "помнить", что коммутатор запросил подтверждение начала применения сигнала.

```
RQNT 525 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0  
X: 919  
S: E/ps(lt=1o, pd=200, pr=1000, rep=3)
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 525 ОК
- 4) Оконечное устройство IPAT передает станции LE интерфейса V5 подтверждение.
- 5) Адаптер МТА начинает использовать импульсы разомкнутого цикла абонентской линии доступа.



J.162REV.1\_FVIII.5

**Рисунок VIII.5/J.162 – Импульсный сигнал с подтверждением начала**

### VIII.7.2.3 Импульсный сигнал с подтверждением завершения

Эта последовательность операций для вызова иллюстрирует запрос импульсного сигнала, при котором станция LE интерфейса V5 запросила подтверждение после использования всех импульсов.

- 1) Станция LE интерфейса V5 запрашивает разомкнутый цикл с несколькими импульсами и подтверждением завершения их использования.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и длительность импульса из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульса, предоставляемое станцией LE интерфейса V5. Поскольку станция LE интерфейса V5 также запросила подтверждение завершения, то оконечное устройство IPAT включает в запрос сигнала параметр "операция завершена". Для данного примера можно предположить также, что станция LE интерфейса V5 запросила подтверждение начала.

```
RQNT 525 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 942
S: E/ps(lt=lo, pd=200, pr=1000, rep=3)
R: oc
```

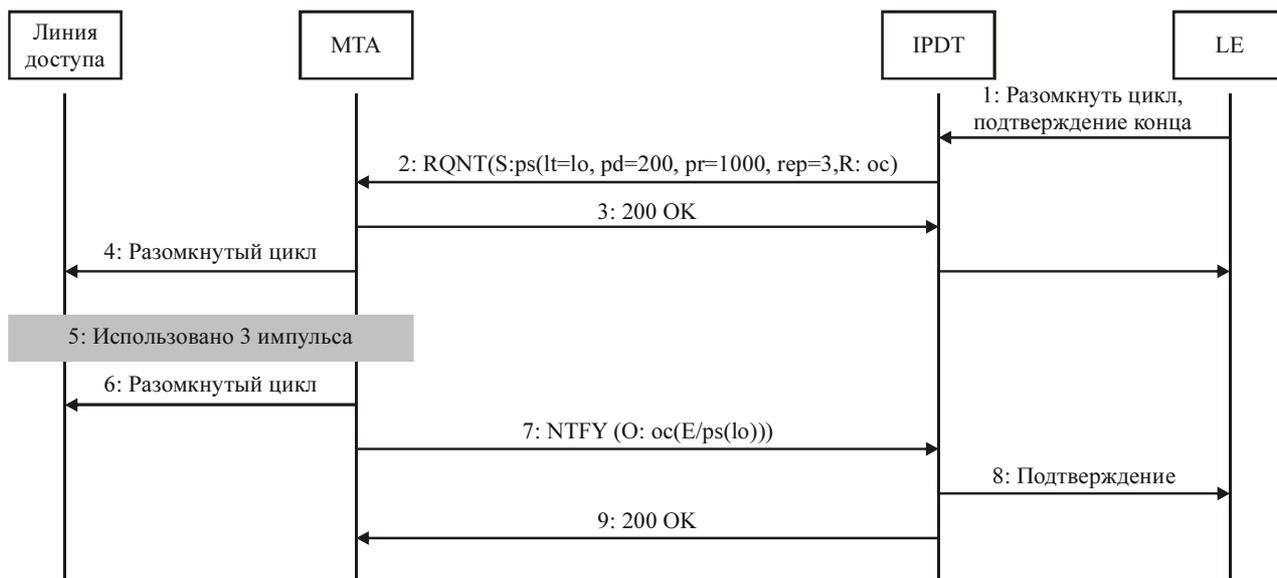
- 3) Адаптер MTA подтверждает запрос сигнала.
 

```
200 525 OK
```
- 4) Адаптер MTA начинает использовать запрошенные импульсы в линии.
- 5) 2-й импульс
- 6) 3-й импульс
- 7) По завершении последнего импульса адаптер MTA уведомляет оконечное устройство IPAT о том, что операция завершена.

```
NTFY 1298 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 942
O: oc(E/ps(lo))
```

Следует отметить, что здесь предполагается наличие "европейского линейного пакета", обозначенного именем "E". Имя пакета может быть опущено, если этот пакет является пакетом по умолчанию.

- 8) Оконечное устройство IPAT посылает станции LE интерфейса V5 запрошенное подтверждение.
- 9) Оконечное устройство IPAT подтверждает уведомление о событии для адаптера MTA.



J.162REV.1\_FVIII.6

**Рисунок VIII.6/J.162 – Импульсный сигнал с подтверждением завершения**

#### VIII.7.2.4 Импульсный сигнал с подтверждением импульса

Эта последовательность операций для вызова иллюстрирует запрос импульсного сигнала, при котором станция LE интерфейса запросила подтверждение после использования каждого импульса.

- 1) Станция LE интерфейса V5 запрашивает разомкнутый цикл с несколькими импульсами и подтверждением импульса.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и длительность импульса из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульса, предоставляемое станцией LE интерфейса V5. Поскольку станция LE интерфейса V5 также запросила подтверждение импульса, окончное устройство IPAT включает встроенный запрос сигнала для сигнала ps.

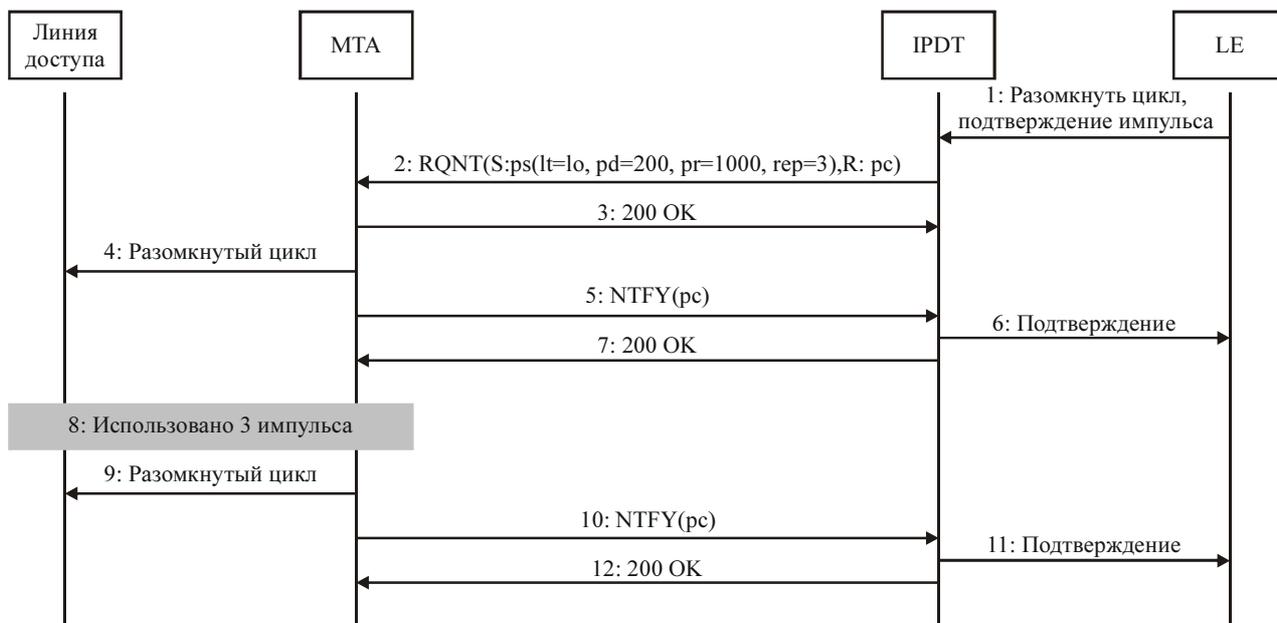
```
RQNT 525 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 1111
S: E/ps(lt=lo, pd=200, pr=1000, rep=3)
R: E/ps
```

- 3) Адаптер МТА подтверждает запрос сигнала.
 

```
200 525 OK
```
- 4) Адаптер МТА подает первый импульс в абонентскую линию доступа.
- 5) По завершении импульса адаптер МТА посылает окончному устройству IPAT уведомление о событии.

```
NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 1111
O: E/ps(lt)
```

- 6) Оконечное устройство IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 7) Оконечное устройство IPAT подтверждает уведомление о событии. Оконечное устройство IPAT не должен посылать новый запрос на уведомление о завершении импульса. Этот запрос остается в силе до тех пор, пока не завершится генерирование тарифных импульсов.
- 8) Адаптер МТА продолжает передавать импульсы и уведомлять об их завершении.



J.162REV.1\_FVIII.7

**Рисунок VIII.7/J.162 – Импульсный сигнал с подтверждением импульса**

### VIII.7.2.5 Импульсный сигнал – тарифный импульс с подтверждением импульса

Эта последовательность операций для вызова иллюстрирует запрос импульсного сигнала, при котором станция LE интерфейса V5 запросила применение тарифного импульса с подтверждением после каждого использованного импульса. Адаптер МТА был обеспечен частотой тарифных импульсов.

- 1) Станция LE интерфейса V5 запрашивает активизацию генерирования тарифных импульсов и подтверждение импульсов.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и генерирует соответствующий запрос сигнала NCS. Поскольку станция LE интерфейса V5 также запросила подтверждение импульса, окончное устройство IPAT включает параметр pc вместе с запросом сигнала.

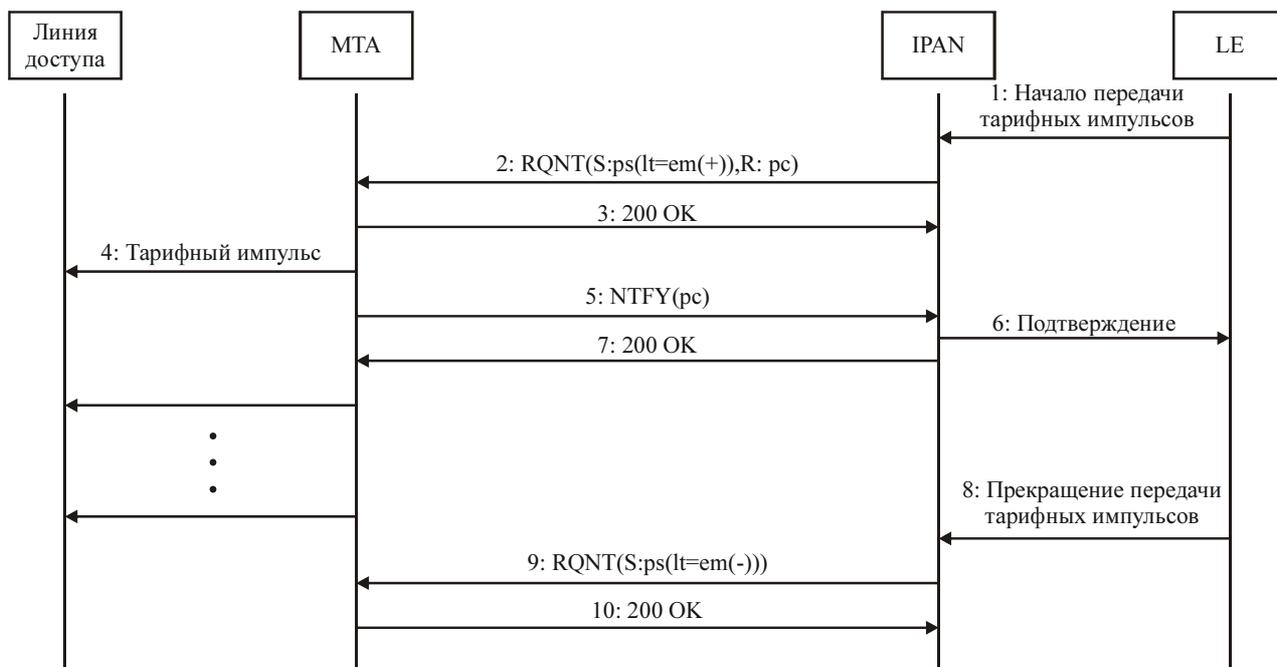
```
RQNT 535 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 2345
S: E/ps(lt=em(+))
R: E/pc
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 535 OK
- 4) Адаптер МТА обращается к своей таблице обеспечения для определения частоты тарифных импульсов, амплитуды и временных параметров по умолчанию и подает первый тарифный импульс в абонентскую линию доступа.
- 5) По завершении импульса адаптер МТА посылает окончному устройству IPAT уведомление о событии:

```
NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 535
O: pc(em)
```

- 6) Оконечное устройство IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 7) Оконечное устройство IPAT подтверждает уведомление о событии.
- 8) Адаптер МТА продолжает передавать импульсы и уведомлять о завершении импульсов до тех пор, пока станция LE интерфейса V5 не прекратит генерирование тарифных импульсов:

```
RQNT 599 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
S: E/ps(lt=em(-))
```



J.162REV.1\_FVIII.8

Рисунок VIII.8/J.162 – Тарификация с подтверждением импульсов

### VIII.7.2.6 Импульсный сигнал – тарифный импульс с подтверждением импульса и изменением тарифа

Эта последовательность операций для вызова иллюстрирует запрос импульсного сигнала, при котором станция LE интерфейса V5 запросила применение тарифного импульса с подтверждением. После того как в первой строке будет использовано несколько импульсов, инициируется изменение тарифа. Частота тарифных импульсов была обеспечена для адаптера МТА.

- 1) Станция LE интерфейса V5 запрашивает применение тарифного импульса с несколькими импульсами и с подтверждением импульса.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и длительность импульса из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульса, предоставляемое станцией LE интерфейса V5. Поскольку станция LE интерфейса V5 также запросила подтверждение импульса, окончное устройство IPAT включает встроенный запрос сигнала для сигнала pc.

```

RQNT 545 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3579
S: E/ps(lt=em(+), pd=150, pr=1000)
R: E/pc
  
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 545 ОК
- 4) Адаптер МТА обращается к своей таблице обеспечения, чтобы определить частоту тарифных импульсов, амплитуду и значения временных параметров по умолчанию (минимальные допустимые значения).
- 5) Оконечное устройство IPAT передает подтверждение начала передачи тарифных импульсов станции LE интерфейса V5.  
Может не быть подтверждения как начала передачи импульсов, так и каждого импульса.
- 6) Адаптер МТА подает первый тарифный импульс в абонентскую линию доступа.
- 7) По завершении импульса адаптер МТА передает окончному устройству IPAT уведомление о событии.

```

NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3579
O: pc(em)
  
```

- 8) Оконечное устройство IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 9) Оконечное устройство IPAT подтверждает уведомление о событии. Оконечное устройство IPAT не должен посылать новый запрос на уведомление о завершении импульса. Этот запрос остается в силе до тех пор, пока не завершится генерирование тарифных импульсов.
- 10) Адаптер МТА продолжает передавать импульсы и уведомлять о завершении импульсов.

В результате изменения состояния вызова (например, начало связи при трехстороннем вызове) станция LE определяет, что должен применяться новый тариф. На основе нового тарифа станция LE определяет новую частоту тарифных импульсов.

- 11) Станция LE интерфейса V5 запрашивает применение тарифного импульса с новым счетчиком тарифных импульсов и подтверждение начала передачи тарифных импульсов.
- 12) Оконечное устройство IPAT преобразует новое закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и длительность импульсов из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульсов, предоставляемое станцией LE интерфейса V5. Поскольку станция LE интерфейса V5 также запросила подтверждение импульсов, окончное устройство IPAT включает встроенный запрос сигнала для сигнала ps. Для данного примера предполагается также, что станция LE интерфейса V5 запросила подтверждение начала передачи тарифных импульсов.

```
RQNT 547 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3581
S: E/ps(lt=em(+), pd=150, pr=500)
R: E/ps
```

- 13) Адаптер МТА подтверждает запрос сигнала.  
200 547 ОК
- 14) Адаптер МТА обращается к своей таблице обеспечения для определения частоты тарифных импульсов, амплитуды и значений временных параметров по умолчанию (минимальные допустимые значения).
- 15) Оконечное устройство IPAT передает станции LE интерфейса V5 подтверждение начала передачи тарифных импульсов.
- 16) Адаптер МТА подает первый новый тарифный импульс в абонентскую линию доступа с новой частотой следования тарифных импульсов.
- 17) По завершении импульса адаптер МТА посылает окончному устройству IPAT уведомление о событии.

```
NTRY 791 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3581
O: ps(em)
```

- 18) Оконечное устройство IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 19) Оконечное устройство IPAT подтверждает уведомление о событии.
- 20) Адаптер МТА продолжает передавать импульсы и уведомлять о завершении импульсов.

### **VIII.7.3 Применение фиксированного числа тарифных импульсов – завершение операции**

Эта последовательность операций для вызова иллюстрирует применение тарифных импульсов с уведомлением о завершении операции.

- 1) Станция LE запрашивает применение двадцати пяти (25) тарифных импульсов в абонентской линии доступа при продолжительности импульса 150 миллисекунд и с интервалом повторений 2000 миллисекунд. Частота тарифных импульсов была обеспечена для адаптера МТА.
- 2) Оконечное устройство IPAT запрашивает применение сигнала тарифных импульсов адаптером МТА.

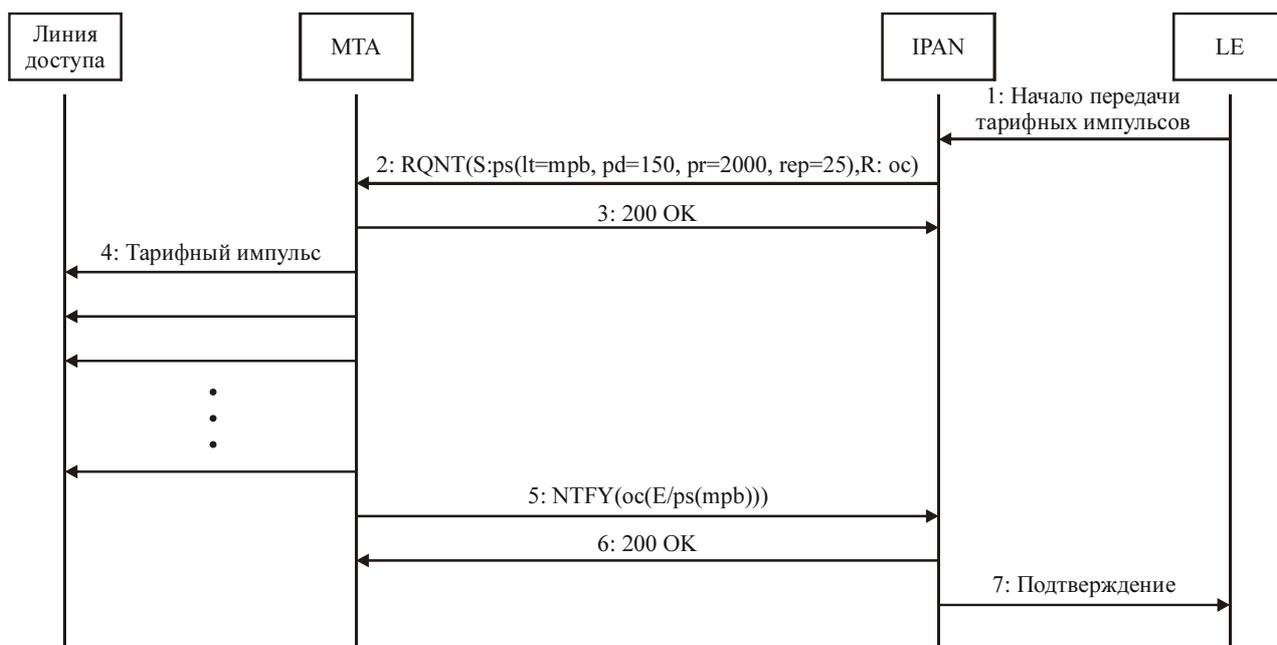
```
RQNT 2367 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 7632
S: E/ps(lt=mpb, pd=150, pr= 2000, rep=25)
R: oc, hu, hf
```

- 3) Адаптер МТА подтверждает запрос.
- 4) Адаптер МТА начинает подавать тарифные импульсы в абонентскую линию доступа.

- 5) В этом примере станция LE запросила уведомление о завершении операции в начальном запросе на генерирование фиксированного числа тарифных импульсов. Теперь адаптер МТА уведомляет оконечное устройство IPAT о том, что операция завершена.

```
NTFY 12876 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 7632
O: oc (E/ps (mpb))
```

- 6) Оконечное устройство IPAT подтверждает уведомление о событии.
- 7) Оконечное устройство IPAT передает станции LE подтверждение завершения импульсного сигнала.



J.162REV.1\_FVIII.9

**Рисунок VIII.9/J.162 – Применение фиксированного числа тарифных импульсов – завершение операции**

#### VIII.7.4 Линейная обработка постоянного сигнала

##### Линейная обработка постоянного сигнала – обратная полярность

Эта последовательность операций для вызова иллюстрирует запрос постоянного сигнала, при котором станция LE интерфейса V5 запросила передать постоянный сигнал обратной полярности на выходы a-b линии POTS.

- 1) Станция LE интерфейса V5 включает в сообщение к оконечному устройству IPAT запрос постоянного сигнала обратной полярности.
- 2) Оконечное устройство IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5, отображает закодированное двоичным кодом сообщение обработки постоянного сигнала обратной полярности в сообщение линейной обработки (lt) NCS и передает это сообщение адаптеру МТА.

```
RQNT 550 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
S: E/ss(lt=rp)
```

- 3) Адаптер МТА подтверждает запрос сигнала.

```
200 550 OK
```

- 4) Адаптер МТА посылает постоянный сигнал обратной полярности на выходы a-b для линии aaln/1, имеющейся в адаптере МТА.

## Дополнение IX

### Поддержка тарификации для сигнализации NCS проекта IP-Cablecom

#### IX.1 Цели

Как описано в документе EPC-ReqDoc-V10-0501 (май 2001 года) "Европейские требования по предоставлению критических во времени услуг по сетям кабельного телевидения при использовании проекта IP-Cablecom", аппаратная тарификация является требованием в поддержку аналоговых линий в кабельной среде IP-протокола. В настоящем Дополнении приводится описание пакета для автоматической передачи импульсов аппаратной тарификации по аналоговым линиям. Это Дополнение также содержит характерные для тарификации последовательности операций при вызове.

ПРИМЕЧАНИЕ. – Описание автономного пакета автоматической тарификации, приведенное в настоящем Дополнении, и описание этого же пакета в Дополнении VIII намеренно являются одинаковыми и должны оставаться такими же. Эквивалентность сигналов тарифных импульсов, описанных в Добавлении VIII, и аналогичных сигналов, описанных в настоящем Дополнении, носит характер прямого соответствия; E/ps(lt=em) отображается в am/em, а E/ps(mpb) – в am/mpb, соответственно. В этих сигналах в обоих пакетах используются одни и те же параметры.

При генерировании данного пакета внимание было уделено вопросу отделения медиашлюза от информации о денежных суммах. Тарифные единицы меняются в зависимости от рынка. Медиашлюз не должен обладать информацией о стоимости импульса (в денежных единицах).

#### IX.2 Пакет автоматической тарификации

Пакет автоматической тарификации разработан так, чтобы удовлетворять требованиям медиашлюзов с аналоговыми линиями, сконфигурированными для телефонии общего назначения, с добавлением возможности автоматической передачи тарифных импульсов.

Импульсные характеристики (тип импульса, длительность импульса, минимальная длительность паузы) зависят от рынка [EN 300 001] и не изменяются во время вызова. Если в сообщении MGCP не включать импульсные характеристики, пакет сохраняет возможность поддерживать любой тип тарифного импульса для любого рынка. Предполагается, что импульсные характеристики для данного пакета обеспечиваются (MIB) в медиашлюзе.

Для данного пакета предполагается, что накопление является задачей устройства CPE. В данном пакете не требуется, чтобы медиашлюз отслеживал число генерируемых импульсов.

Для данного пакета предполагается, что шлюзы обладают надежностью в отношении генерирования импульсов. Пакет не имеет обратной связи (события, свойства, статистические данные) с числом импульсов, фактически генерируемых во время выполнения вызова.

##### IX.2.1 Имя пакета

Имя пакета: am

Версия: 1

Перед тарифными сигналами и событиями всегда ДОЛЖЕН стоять префикс в виде имени пакета "am".

##### IX.2.2 Варианты локального соединения

Отсутствуют.

##### IX.2.3 События и сигналы

В данном пакете вводятся два сигнала.

**Таблица IX.1/J.162 – Сигналы в пакете тарификации**

Символ	Определение	R	Тип	Длительность
em	Активизация тарификации		OO	нет данных
mpb	Пачка тарифных импульсов		BR	нет данных
R	В этом столбце проставляется "x", если агент вызова может запросить событие. Как вариант может быть включено "S", если состояние события может быть проконтролировано. "C" означает, что событие может быть обнаружено в соединении.			
Тип	Если для события в данном столбце ничего не указано, агент вызова не может передать сигнал о событии. В противном случае тип события идентифицируют следующие символы: OO сигнал "включено/выключено"; TO сигнал выдержки времени; BR короткий сигнал.			
Длительность	Определяет длительность сигналов TO. Если длительность остается неопределенной, тогда предполагается, что выдержка времени по умолчанию будет бесконечной.			

### IX.2.3.1 Сигнал "пачка тарифных импульсов"

Имя сигнала: am/mpb

Тип сигнала: короткий

Сигнал тарифных импульсов используется для передачи сигнала о начислении платы за попытку вызова, за установление соединения и о дополнительном начислении платы. При этом сигнале запрашивается генерирование фиксированного числа тарифных импульсов в аналоговой линии. Следует отметить, что сигнал тарифных импульсов может быть также использован для запроса генерирования одного тарифного импульса.

Дополнительные параметры:

- *Счетчик импульсов*  
Идентификатор параметра: гер  
Тип: целочисленный, гер > 0  
Значение по умолчанию: 1  
Этот параметр определяет число тарифных импульсов, которое должно быть подано в линию. Адаптер МТА ДОЛЖЕН генерировать импульсы, пока их число не достигнет значения, установленного в счетчике.  
Значение по умолчанию данного параметра, которое ДОЛЖНО использоваться, если этот параметр опущен, равно 1.
- *Интервал повторения импульса*  
Идентификатор параметра: пр  
Тип: целочисленный, пр > 0  
Значение по умолчанию: 1000  
Этот параметр определяет интервал в миллисекундах между повторениями тарифных импульсов в линии. Он представляет время, которое ДОЛЖНО истечь между передним фронтом импульса и передним фронтом последующего импульса.  
Значение по умолчанию данного параметра, которое ДОЛЖНО использоваться, если этот параметр опущен, составляет 1000 мс.

Запрос сигнала "пачка тарифных импульсов" может быть включен в запрос сигнала, который активизирует генерирование тарифных импульсов, например, для того чтобы произвести начальное начисление платы для вызова. Когда это происходит, адаптер МТА ДОЛЖЕН полностью использовать пачку тарифных импульсов для конечной точки и затем начать генерирование нормальных тарифных импульсов.

Поскольку сигнал "пачка тарифных импульсов" является типом короткого сигнала, то ДОЛЖНЫ использоваться все сигналы, определяемые для запроса (**rep=n**), даже если во время передачи этой пачки импульсов абонент положит телефонную трубку.

Считается ошибкой, когда адаптер МТА принимает сигнал "пачка тарифных импульсов", а телефон находится в состоянии "телефонная трубка положена". Когда предпринимаются такие попытки, то ДОЛЖНА быть возвращена информация об ошибке (код ошибки 402 – телефонная трубка положена).

Сигнал am/mpb ДОЛЖЕН использоваться в конечных точках, а НЕ в соединениях.

### IX.2.3.2 Сигнал активизации тарификации

Имя сигнала: am/em

Тип сигнала: включено/выключено

По этому сигналу начинается автоматическое генерирование тарифных импульсов в аналоговой линии. Этот сигнал используется для информирования о регулярном, повременном начислении платы за вызов. Первый тарифный импульс по начислению платы за вызов ДОЛЖЕН быть выдан немедленно после получения сигнала em.

Дополнительные параметры:

- *Интервал повторения импульса*

Идентификатор параметра: pr

Тип: целочисленный,  $pr > 0$

Значение по умолчанию: 1000

Этот параметр задает интервал в миллисекундах между повторениями тарифных импульсов в линии. Он представляет время, которое ДОЛЖНО истечь между передним фронтом импульса и передним фронтом последующего импульса. Адаптер МТА ДОЛЖЕН продолжать генерировать импульсы до тех пор, пока он не получит новый сигнал am/em или пока сигнал em не будет выключен явным образом. Если линия входит в бестоковое состояние (телефонная трубка положена), то адаптеру МТА СЛЕДУЕТ отключить тарифные импульсы в ожидании установления нового соединения для вызова (при новом вызове линия СРЕ снова входит в токовое состояние (телефонная трубка снята)).

Значение этого параметра по умолчанию, которое ДОЛЖНО использоваться, если параметр опущен, равно 1000 мс.

Сигналы "активизация тарификации" являются взаимоисключающими; в любой момент времени ДОЛЖЕН быть активным только один сигнал "активизация тарификации". Если поступает новый сигнал am/em, он ДОЛЖЕН заменить любой предшествующий сигнал am/em.

Запрос сигнала "пачка тарифных импульсов" может иметь место в процессе вызова, например, чтобы учесть действия абонента, подлежащие оплате. Когда это происходит, адаптер МТА ДОЛЖЕН приостановить нормальное генерирование тарифных импульсов и применить сигналы "пачка тарифных импульсов". Затем адаптер МТА должен возобновить нормальное генерирование тарифных импульсов, не требуя от агента вызова нового запроса сигнала "активизация тарификации". Агент вызова ДОЛЖЕН учитывать все нормальные тарифные импульсы, пропущенные при передаче пачки импульсов, путем включения пропущенных импульсов в счетчик пачек тарифных импульсов.

Считается ошибкой, когда адаптер МТА принимает сигнал "активизация тарификации", а телефон находится в состоянии "телефонная трубка положена". Когда предпринимаются такие попытки, то ДОЛЖНА быть возвращена информация об ошибке (код ошибки 402 – телефонная трубка положена).

Синтаксическим обозначением выключения сигнала "активизация тарификации" является am/em(-). Когда сигнал выключения активизации тарификации получен в состоянии "телефонная трубка снята", информация об ошибке возвращается НЕ ДОЛЖНА.

Сигнал am/em ДОЛЖЕН использоваться в конечных точках, а НЕ в соединениях.

### IX.2.4 Свойства

Информация о свойствах отсутствует.

### IX.2.5 Статистические данные

Статистические данные отсутствуют.

### IX.2.6 Процедуры

Информация о процедурах отсутствует.

### **IX.3 Случаи использования – примеры последовательностей операций при вызовах**

#### **IX.3.1 Тарифный импульс при снятой телефонной трубке**

Агент вызова посылает адаптеру МТА команду использовать один импульс. Если параметр "rep" опущен, то его значение устанавливается по умолчанию на "1". Телефонный аппарат находится в состоянии "телефонная трубка снята".

```
RQNT 309 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 860
S: am/mpb
```

Адаптер МТА передает подтверждение.

```
200 309 OK
```

#### **IX.3.2 Телефонный импульс при положенной телефонной трубке**

Агент вызова посылает адаптеру МТА команду использовать один вызов, пока телефонный аппарат находится в состоянии "телефонная трубка положена".

```
RQNT 310 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 870
S: am/mpb
```

Адаптер МТА отклоняет запрос.

```
402 310 phone on-hook
```

#### **IX.3.3 Регулярное начисление платы за вызов**

Агент вызова посылает адаптеру МТА команду на применение регулярного начисления платы за вызов согласно одному тарифному импульсу каждые 12 секунд.

```
RQNT 311 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 880
S: am/em(pr=12000)
```

Адаптер МТА передает подтверждение.

```
200 311 OK
```

#### **IX.3.4 Начисление платы за установление соединения**

Агент вызова посылает адаптеру МТА команду использовать пачку тарифных импульсов из 33 импульсов.

```
RQNT 321 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 881
S: am/mpb(rep=33)
```

Адаптер МТА передает подтверждение.

```
200 321 OK
```

Потом агент вызова посылает адаптеру МТА команду использовать регулярное начисление платы за вызов согласно одному тарифному импульсу каждые 5 секунд.

```
RQNT 322 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 882
S: am/em(pr=5000)
```

Адаптер МТА передает подтверждение.

```
200 322 OK
```

Следует отметить, что агент вызова имеет возможность использовать оба сигнала в одном запросе:

```
RQNT 323 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
X: 883
```

S: am/mpb(rep=33), am/em(pr=5000)

Адаптер МТА передает подтверждение.

200 323 ОК

### **IX.3.5 Изменение тарифа при промежуточном состоянии вызова**

Агент вызова посылает адаптеру МТА команду использовать регулярное начисление платы за вызов согласно одному тарифному импульсу каждые 8 секунд.

RQNT 331 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0  
X: 884  
S: am/em(pr=8000)

Адаптер МТА передает подтверждение.

200 331 ОК

Позднее, когда в течение вызова наступает другое время суток, происходит изменение тарифа. Агент вызова посылает адаптеру МТА команду использовать один тарифный импульс каждые 12 секунд.

RQNT 332 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0  
X: 885  
S: am/em(pr=12000)

Адаптер МТА передает подтверждение.

200 332 ОК

### **IX.3.6 Дополнительное начисление платы при промежуточном состоянии вызова**

Пусть первоначально маршрутизация вызова осуществляется к устройству оповещения. Агент вызова посылает адаптеру МТА команду использовать тарифный импульс каждые 10 секунд.

RQNT 341 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0  
X: 886  
S: am/em(pr=10000)

Адаптер МТА передает подтверждение.

200 341 ОК

Позднее, когда вызов маршрутизируется к оператору, имеет место дополнительное начисление платы. Агент вызова посылает адаптеру МТА команду использовать разовую пачку из 20 тарифных импульсов, не влияя на регулярное начисление платы за вызов.

RQNT 342 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0  
X: 887  
S: am/mpb(rep=20)

Адаптер МТА передает подтверждение.

200 342 ОК

### **IX.3.7 Окончание вызова**

В конце вызова агент вызова посылает адаптеру МТА команду исключить соединение и прекратить регулярное начисление платы за вызов.

DLCX 351 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0  
C: abcd  
S: am/em(-)

Адаптер МТА передает подтверждение.

250 351 ОК

### IX.3.8 Контроль конечной точки

Короткие сигналы не обладают контролируемым состоянием. В ответ на запрос сигнала контроля не включаются короткие сигналы, в текущий момент передаваемые согласно спецификации протокола MGCP.

Состояние сигналов "включено/выключено" является контролируемым. Если в команде "контроль конечной точки" запрашивается проверка равенства параметров (RequestedInfo=SignalRequests), тогда адаптер МТА ДОЛЖЕН вернуть список сигналов "включено/выключено", которые в текущий момент "включены" для конечной точки (с параметрами или без них).

Агент вызова контролирует конечную точку.

```
AUEP 361 aaln/1@mg23.whatever.net MGCP 1.0 NCS 1.0
F: S
```

Ответ указывает на то, что сигнал регулярного начисления платы за вызов включен.

```
200 361 OK
S: am/em(pr=10000)
```

### IX.4 Термины

**IX.4.1 плата:** Число тарифных единиц (за использование оплачиваемого события (услуги электросвязи)).

**IX.4.2 тарифная единица:** Базовый элемент процесса начисления платы, выраженный в виде единиц тарифных импульсов или в виде денежной величины.

**IX.4.3 дополнительное начисление платы:** Одно дополнительное начисление платы, не меняющее текущий тариф.

**IX.4.4 тариф:** Набор параметров, используемых в целях начисления платы путем вычисления численных тарифных единиц для используемой услуги электросвязи или группы используемых услуг электросвязи. Тариф состоит из тарифной последовательности.

**IX.4.5 тарифная последовательность:** Перечень из 4 последовательных субтарифов, который должен использоваться для тарификации события связи. Субтарифы применяются от начала события связи и применяются последовательно согласно перечню субтарифов. Последний субтариф может иметь неограниченную длительность.

**IX.4.6 субтариф:** Тарифная единица в единицу времени в рамках тарифной последовательности. Каждый субтариф имеет свою длительность и свою тарифную единицу.

**IX.4.7 тарифный импульс:** Периодический тактовый сигнал с одним периодом включения и одним периодом выключения. Тремя наиболее общими типами тарифных импульсов являются: импульс с частотой 12 кГц, импульс с частотой 16 кГц и импульс с обратной полярностью.



**IX.4.8 МІВ:** Информационная база управления.

**IX.4.9 интервал повторения импульса:** Изменяется с изменением платы; чем выше плата, тем короче интервал повторения импульса.

**IX.4.10 период включенного импульса:** Имеет фиксированную длину; однако его длительность зависит от национальных спецификаций. См. EN 300 001 V1.5.1 (1998-10), раздел 1.7.8.

**IX.4.11 период выключенного импульса (пауза):** Изменяется вместе с изменением интервала повторения импульса; его минимальная длительность зависит от национальных спецификаций. См. EN 300 001 V1.5.1 (1998-10), раздел 1.7.8.

## Дополнение X

### Грамматика ABNF для NCS

RFC 3435 включает в себя формальное описание синтаксиса протокола MGCP, согласно "Расширенной BNF (форма Бэкуса-Науэра) для детальных указаний по синтаксису". Данное формальное описание ссылается на разработчиков совместно функционирующих устройств. Копия синтаксиса протокола MGCP, дополненная аннотацией и отредактированная для наглядности применения к указаниям по PacketCable, содержится в данном дополнении.

Реализации ДОЛЖНЫ отвечать частям грамматики ABNF, относящимся к соответствующим им детальным указаниям, т. е. NCS, TGCP. Также следует обратить внимание на то, что существуют несколько кодировок параметров (например, встроенный запрос, отображение цифр, имена расширения разработчиков), где грамматика NCS и/или TGCP отличается от грамматики MGCP.

Для определения различий между четырьмя разными случаями используются пять аннотаций:

- 1) Язык RFC изменен для соответствия требованиям NCS и TGCP.
- 2) Язык RFC применим только к NCS (и возможно MGCP).
- 3) Язык RFC применим только к MGCP (и возможно NCS).
- 4) Язык RFC применим только к NCS и MGCP.
- 5) Язык RFC применим только к MGCP.

Язык в каждом случае выделен посредством различного семейства шрифтов так, как это определено в описании изменений ниже.

```
;Грамматика RFC 3435 изменена для соответствия требованиям NCS и TGCP.  
;Bold показывает только NCS (и возможно MGCP)  
;Italics показывает только TGCP (и возможно MGCP)  
;Bold italics показывает только NCS и TGCP  
;Текст серого цвета - только MGCP
```

```
MGCPMessage = MGCPCommand / MGCPResponse  
MGCPCommand = MGCPCommandLine 0*(MGCPParameter) [EOL *SDPinformation]  
MGCPCommandLine = MGCPVerb 1*(WSP) transaction-id 1*(WSP)  
                    endpointName 1*(WSP) MGCPversion EOL  
MGCPVerb = "EPCF" / "CRCX" / "MDCX" / "DLCX" / "RQNT"  
           / "NTFY" / "AUER" / "AUCX" / "RSIP" / extensionVerb  
extensionVerb = ALPHA 3(ALPHA / DIGIT) ; Экспериментальный пакет начинается с X  
transaction-id = 1*9(DIGIT)
```

```
endpointName      = LocalEndpointName "@" DomainName  
LocalEndpointName = LocalNamePart 0*("/" LocalNamePart)  
LocalNamePart     = AnyName / AllName / NameString  
AnyName           = "$"  
AllName           = "*"  
NameString        = 1*(range-of-allowed-characters)  
; VCHAR за исключением "$", "*", "/", "@"  
range-of-allowed-characters = %x21-23 / %x25-29 / %x2B-2E  
                             / %x30-3F / %x41-7E
```

```
DomainName = 1*255(ALPHA / DIGIT / "." / "-") ; как указано  
           / "#" number ; в RFC 821  
           / "[" IPv4address / IPv6address "]" ; см. RFC 2373
```

```
; Переписано в ABNF из RFC 821  
number = 1*DIGIT
```

```

;из RFC 2373
IPv6address = hexpart [ ":" IPv4address ]
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
; на эту конструкцию при появлении ее в RFC 2373 ссылаться нельзя
; IPv6prefix = hexpart "/" 1*2DIGIT
hexpart = hexseq / hexseq ":" [ hexseq ] / ":" [ hexseq ]
hexseq = hex4 *( ":" hex4)
hex4 = 1*4HEXDIG

MGCPversion = "MGCP" 1*(WSP) 1*(DIGIT) "." 1*(DIGIT)
              [1*(WSP) ProfileName]
ProfileName = "NCS 1.0" ; Для NCS
              / "TGCP 1.0" ; Для TGCP
              / VCHAR *( WSP / VCHAR)

MGCPPparameter = ParameterValue EOL

; Если определены еще значения параметра, проверьте infoCode
; Наиболее необязательные значения могут быть опущены только при проверке
ParameterValue = ("K" ":" 0*(WSP) [ResponseAck])
                 / ("B" ":" 0*(WSP) [BearerInformation])
                 / ("C" ":" 0*(WSP) CallId)
                 / ("I" ":" 0*(WSP) [ConnectionId])
                 / ("N" ":" 0*(WSP) [NotifiedEntity])
                 / ("X" ":" 0*(WSP) [RequestIdentifier])
                 / ("L" ":" 0*(WSP) [LocalConnectionOptions])
                 / ("M" ":" 0*(WSP) ConnectionMode)
                 / ("R" ":" 0*(WSP) [RequestedEvents])
                 / ("S" ":" 0*(WSP) [SignalRequests])
                 / ("D" ":" 0*(WSP) [DigitMap]) ; Для NCS (и MGCP)
                 / ("O" ":" 0*(WSP) [ObservedEvents])
                 / ("P" ":" 0*(WSP) [ConnectionParameters])
                 / ("E" ":" 0*(WSP) ReasonCode)
                 / ("Z" ":" 0*(WSP) [SpecificEndpointID])
                 / ("ZZ" ":" 0*(WSP) SecondEndpointID)
                 / ("I2" ":" 0*(WSP) SecondConnectionID)
                 / ("F" ":" 0*(WSP) [RequestedInfo])
                 / ("Q" ":" 0*(WSP) QuarantineHandling)
                 / ("T" ":" 0*(WSP) [DetectEvents])
                 / ("RM" ":" 0*(WSP) RestartMethod)
                 / ("RD" ":" 0*(WSP) RestartDelay)
                 / ("A" ":" 0*(WSP) [Capabilities])
                 / ("ES" ":" 0*(WSP) [EventStates])
                 / ("PL" ":" 0*(WSP) [PackageList]) ; Только проверка
                 / ("MD" ":" 0*(WSP) MaxMGCPDatagram) ; Только проверка /
(extensionParameter ":" 0*(WSP) [parameterString])
  / VersionSupported ; NCS и TGCP - только в направлении передачи
  / MaxEndpointIds ; NCS и TGCP
  / NumEndpoints ; NCS и TGCP - только в направлении передачи

; <extensionParameter> ":" parameterString, определенный NCS и TGCP
VersionSupported = "VS" ":" MGCPversion *( "," 0*(WSP) MGCPversion)
MaxEndpointIds = "ZM" ":" 0*(WSP) 1*16(DIGIT)
NumEndpoints = "ZN" ":" 0*(WSP) 1*16(DIGIT) ; Только ответы
; Конечный ответ может включать пустой параметр ResponseAck
ResponseAck = confirmedTransactionIdRange
              *( "," 0*(WSP) confirmedTransactionIdRange )

confirmedTransactionIdRange = transaction-id ["-" transaction-id]

```

```

BearerInformation = BearerAttribute 0*("," 0*(WSP) BearerAttribute)
BearerAttribute   = ("e" ":" BearerEncoding)
                  / (BearerExtensionName [":" BearerExtensionValue])
BearerExtensionName = PackageLCOExtensionName
BearerExtensionValue = LocalOptionExtensionValue
BearerEncoding = "A" / "mu"
CallId = 1*32(HEXDIG)

; Ответ на запрос проверки может включать в себя список идентификаторов
ConnectionId = 1*32(HEXDIG) 0*("," 0*(WSP) 1*32(HEXDIG))
SecondConnectionID = ConnectionId

NotifiedEntity = [LocalName "@"] DomainName [":" portNumber]
LocalName = 1*(LocalCallAgentNameCharacters)
LocalCallAgentNameCharacters = %x21-3F / %x41-7E ; VCHAR за исключением "@"

portNumber = 1*5(DIGIT)

RequestIdentifier = 1*32(HEXDIG)

LocalConnectionOptions = LocalOptionValue 0*(WSP)
                        0*("," 0*(WSP) LocalOptionValue 0*(WSP))
LocalOptionValue = ("p" ":" packetizationPeriod)
                  / ("a" ":" compressionAlgorithm)
                  / ("b" ":" bandwidth) ; Только для возможностей
                  ; NCS и TGCP
                  / ("e" ":" echoCancellation)
                  / ("gc" ":" gainControl)
                  / ("s" ":" silenceSuppression)
                  / ("t" ":" typeOfService)
                  / ("r" ":" resourceReservation)
                  / ("k" ":" encryptiondata)
                  / ("nt" ":" ( typeOfNetwork /
                               supportedTypeOfNetwork))
                  / (LocalOptionExtensionName
                     [":" LocalOptionExtensionValue])
                  / MPacketizationPeriod ; Только NCS и TGCP
                  / RTPciphersuite ; Только NCS и TGCP
                  / RTCPciphersuite ; Только NCS и TGCP
                  / DQoSGateID ; Только NCS и TGCP
                  / DQoSReservation ; Только NCS и TGCP
                  / DQoSResourceID ; Только NCS и TGCP
                  / DQoSReserveDestination ; Только NCS и TGCP
                  / CallContentId ; Только NCS и TGCP
                  / CallContentDestination ; Только NCS и TGCP

Capabilities = CapabilityValue 0*(WSP)
              0*("," 0*(WSP) CapabilityValue 0*(WSP))
CapabilityValue = LocalOptionValue
                 / ("v" ":" supportedPackages)
                 / ("m" ":" supportedModes)

PackageList = pkgNameAndVers 0*("," pkgNameAndVers)
pkgNameAndVers = packageName ":" packageVersion
packageVersion = 1*(DIGIT)

; Для NCS и TGCP, диапазон форматов разрешен только для возможностей
; а не для LocalConnectionOptions.
packetizationPeriod = 1*4(DIGIT) ["-" 1*4(DIGIT)]
compressionAlgorithm = algorithmName 0*("; algorithmName)

```

```

algorithmName      = 1*(SuitableLCOCharacter)
bandwidth          = 1*4(DIGIT) ["-" 1*4(DIGIT)]
echoCancellation  = "on" / "off"
gainControl       = "auto" / ["-"] 1*4(DIGIT)
silenceSuppression = "on" / "off"
typeOfService     = 1*2(HEXDIG) ; шестнадцатер. 1 только для возможностей
resourceReservation = "g" / "cl" / "be"

;параметры шифрования кодируются как в SDP (RFC 2327)
;ПРИМЕЧАНИЕ: ключ шифрования может содержать алгоритм, как определено в RFC 1890
encryptiondata = ( "clear" ":" encryptionKey )
                / ( "base64" ":" encodedEncryptionKey )
                / ( "uri" ":" URIToObtainKey )
                / ( "prompt" ) ; определено в SDP, не подлежит использованию в MGCP!
encryptionKey = 1*(SuitableLCOCharacter) / quotedString
; См. RFC 2045
encodedEncryptionKey = 1*(ALPHA / DIGIT / "+" / "/" / "=")
URIToObtainKey = 1*(SuitableLCOCharacter) / quotedString

typeOfNetwork = "IN" / "ATM" / "LOCAL" / OtherTypeOfNetwork
; Зарегистрировано с IANA - см. RFC 2327
OtherTypeOfNetwork = 1*(SuitableLCOCharacter)
supportedTypeOfNetwork = typeOfNetwork *("; " typeOfNetwork)
supportedModes = ConnectionMode 0*("; " ConnectionMode)

supportedPackages = packageName 0*("; " packageName)

packageName = 1*(ALPHA / DIGIT / HYPHEN) ; Дефис не может стоять первым или
последним

LocalOptionExtensionName = VendorLCOExtensionName
                        / PackageLCOExtensionName
                        / OtherLCOExtensionName
VendorLCOExtensionName = "x" ("+" / "-") 1*32(SuitableExtLCOCharacter)
PackageLCOExtensionName = packageName "/"
                        1*32(SuitablePkgExtLCOCharacter)
; must not start with "x-" or "x+"
OtherLCOExtensionName = 1*32(SuitableExtLCOCharacter)

; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; определено в NCS/TGCP
MPacketizationPeriod = "mp" ":" multiplepacketizationPeriod
multiplepacketizationPeriod = mpPeriod 0*("; " mpPeriod)
mpPeriod              = 1*4(DIGIT) / HYPHEN

RTPciphersuite = "sc-rtp" ":" ciphersuite
RTCPciphersuite = "sc-rtcp" ":" ciphersuite
ciphersuite     = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]
AuthenticationAlgorithm = 1*( ALPHA / DIGIT / "-" / "_" )
EncryptionAlgorithm     = 1*( ALPHA / DIGIT / "-" / "_" )

; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; определено только в NCS
DQoSGateID = "dq-gi" [":" 1*8(HEXDIG)] ; Пустое только для
; возможностей
DQoSReservation = "dq-rr" ":" DQoSResMode *("; " DQoSResMode)
DQoSResMode     = "sendresv" / "recvresv" / "snrcresv" /
                 "sendcomt" / "recvcomt" / "snrccomt"
DQoSResourceID = "dq-ri" ":" 1*8(HEXDIG)
DQoSReserveDestination = "dq-rd" ":" IPv4address [":" portNumber]

```

```

; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; определено только в TGCP
CallContentId = "es-cci" ":" 1*8(HEXDIG)
CallContentDestination = "es-ccd" ":" IPv4address ":" portNumber

LocalOptionExtensionValue = (1*(SuitableExtLCOValChar
                               / quotedString)
                              *("; " (1*(SuitableExtLCOValChar
                                       / quotedString)))

; Примечание: Режимы "данные" нет.
ConnectionMode = "sendonly" / "recvonly" / "sendrecv"
                / "confrnce" / "inactive"
                / "loopback" / "contttest" ; только TGCP (и MGCP)
                / "replcate" ; только NCS
                / "netwloop" / "netwttest"
                / ExtensionConnectionMode
ExtensionConnectionMode = PkgExtConnectionMode
PkgExtConnectionMode = packageName "/" 1*(ALPHA / DIGIT)

RequestedEvents = requestedEvent 0*("," 0*(WSP) requestedEvent)
requestedEvent = (eventName ["(" requestedActions ")"])
                / (eventName "(" requestedActions ")"
                   (" eventParameters ")")

eventName = [(packageName / "*") "/"
             (eventId / "all" / eventRange
              / "*" / "#") ; для DTMF
             ["@" (ConnectionId / "$" / "*")]]

eventId = 1*(ALPHA / DIGIT / HYPHEN) ; Дефис нельзя ни первым ни последним
eventRange = "[" 1*(DigitMapLetter / (DIGIT "-" DIGIT) /
                (DTMFLetter "-" DTMFLetter)) "]"
DTMFLetter = "A" / "B" / "C" / "D"

requestedActions = requestedAction 0*("," 0*(WSP) requestedAction)
requestedAction = "N" / "A"
                / "D" ; For NCS (and MGCP)
                / "S" / "I" / "K"
                / "E" "(" EmbeddedRequest ")"
                / ExtensionAction
                / "C" "(" EmbeddedModeChange ; Только для NCS
                    0*("," 0*WSP EmbeddedModeChange) ")" ; и TGCP
; NCS и TGCP определяют действие Embedded ModifyConnection.
; Грамматика MGCP не допускает форматы, используемые в NCS и TGCP:
EmbeddedModeChange = "M" "(" ConnectionMode "(" EmConnectionId ")" ")"
EmConnectionId = ConnectionId / "$"

ExtensionAction = PackageExtAction
PackageExtAction = packageName "/" Action ["(" ActionParameters ")"]
Action = 1*ALPHA

ActionParameters = eventParameters ; Может содержать действия

; ПРИМЕЧАНИЕ. - Различные порядки следует согласовывать при получении, например,
для NCS
EmbeddedRequest = ( "R" "(" EmbeddedRequestList ")"
                  [", " 0*(WSP) "S" "(" EmbeddedSignalRequest ")"]
                  [", " 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )
                / ( "S" "(" EmbeddedSignalRequest ")"
                  [", " 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )

```

```

        / ( "D" "(" EmbeddedDigitMap ")" )
    / NCSTGCPEEmbeddedRequest
; Текст, приведенный ниже, относится только к NCS и TGCP. Отличие от MGCP
; заключается только в том, что порядок элементов не фиксирован. Также
; для TGCP не используется отображение цифр
NCSTGCPEEmbeddedRequest = NCSTGCPEEmbeddedRequestItem
    *2( " ," 0*(WSP) NCSTGCPEEmbeddedRequestItem )
NCSTGCPEEmbeddedRequestItem = ( "R" "(" EmbeddedRequestList ")" )
    / ( "S" "(" EmbeddedSignalRequest ")" )
    / ( "D" "(" EmbeddedDigitMap ")" )

EmbeddedRequestList = RequestedEvents
EmbeddedSignalRequest = SignalRequests
EmbeddedDigitMap = DigitMap

SignalRequests = SignalRequest 0*( " ," 0*(WSP) SignalRequest )
SignalRequest = eventName [ "(" eventParameters ")" ]

eventParameters = eventParameter 0*( " ," 0*(WSP) eventParameter )
eventParameter = eventParameterValue
    / eventParameterName "=" eventParameter
    / eventParameterName "(" eventParameters ")"
eventParameterString = 1*(SuitableEventParamCharacter)
eventParameterName = eventParameterString
eventParameterValue = eventParameterString / quotedString

; Для NCS (и MGCP)
DigitMap = DigitString / "(" DigitStringList ")"
DigitStringList = DigitString 0*( "|" DigitString )
DigitString = 1*(DigitStringElement)
DigitStringElement = DigitPosition [ "." ]
DigitPosition = DigitMapLetter / DigitMapRange
; теперь включено
DigitMapLetter = DIGIT / "#" / "*" / "A" / "B" / "C" / "D" / "T"
    / "X" / ExtensionDigitMapLetter
ExtensionDigitMapLetter = "E" / "F" / "G" / "H" / "I" / "J" / "K"
    / "L" / "M" / "N" / "O" / "P" / "Q" / "R"
    / "S" / "U" / "V" / "W" / "Y" / "Z"
; ПРИМЕЧАНИЕ: "[x]" теперь разрешен в MGCP.
; В NCS, разрешена только форма "x"
DigitMapRange = "[" 1*DigitLetter "]"
    / "X" ; Добавлена только для NCS
DigitLetter = *( (DIGIT "-" DIGIT) / DigitMapLetter )

ObservedEvents = SignalRequests
EventStates = SignalRequests

ConnectionParameters = ConnectionParameter
    0*( " ," 0*(WSP) ConnectionParameter )
ConnectionParameter = ( "PS" "=" packetsSent )
    / ( "OS" "=" octetsSent )
    / ( "PR" "=" packetsReceived )
    / ( "OR" "=" octetsReceived )
    / ( "PL" "=" packetsLost )
    / ( "JI" "=" jitter )
    / ( "LA" "=" averageLatency )
    / ( ConnectionParameterExtensionName
        "=" ConnectionParameterExtensionValue )
    / RemotePacketsSent
    / RemoteOctetsSent

```

```

        / RemotePacketsLost
        / RemoteJitter
; NCS и TGCP определяют следующие четыре имени расширения параметров
; соединения:
RemotePacketsSent = "PC/RPS" "=" packetsSent
RemoteOctetsSent = "PC/ROS" "=" octetsSent
RemotePacketsLost = "PC/RPL" "=" packetsLost
RemoteJitter = "PC/JI" "=" jitter

packetsSent      = 1*9(DIGIT)
octetsSent       = 1*9(DIGIT)
packetsReceived  = 1*9(DIGIT)
octetsReceived   = 1*9(DIGIT)
packetsLost      = 1*9(DIGIT)
jitter           = 1*9(DIGIT)
averageLatency   = 1*9(DIGIT)

ConnectionParameterExtensionName = VendorCPEExtensionName
                                   / PackageCPEExtensionName
VendorCPEExtensionName = "X" "-" 2*ALPHA
                                   / NCSTGCPVendorCPEExtensionName
; Текст, приведенный ниже, относится только к NCS и TGCP. Отличие от MGCP
; заключается только в том, что MGCP требует 2 alpha символа, в то время как NCS
; и TGCP разрешают 2 или 3 alpha символа для VendorCPEExtensionName
NCSTGCPVendorCPEExtensionName = "X" "-" 2*3ALPHA

PackageCPEExtensionName = packageName "/" CPName
CPName = 1*(ALPHA / DIGIT / HYPHEN)
ConnectionParameterExtensionValue = 1*9(DIGIT)

MaxMGCPDatagram = 1*9(DIGIT)

ReasonCode = 3DIGIT
             [1*(WSP) "/" packageName] ; Только для 8xx
             [WSP 1*(%x20-7E)]

SpecificEndpointID = endpointName
SecondEndpointID   = endpointName

RequestedInfo = infoCode 0*("," 0*(WSP) infoCode)

infoCode = "B" / "C" / "I" / "N" / "X" / "L" / "M" / "R" / "S"
           / "D" ; Только для NCS (и MGCP)
           / "O" / "P" / "E" / "Z" / "Q" / "T" / "RC" / "LC"
           / "A" / "ES" / "RM" / "RD" / "PL" / "MD" / extensionParameter
           / "VS" / "ZM" / "ZN" ; NCS и TGCP определяют эти
                               ; три extensionParameters (параметры расширения)

; NCS и TGCP позволяют контроль процесса и шлейфа в любом порядке
QuarantineHandling = loopControl / processControl
                    / (loopControl "," 0*(WSP) processControl )
                    / (processControl "," 0*(WSP) loopControl)
loopControl        = "step" / "loop"
processControl     = "process" / "discard"

DetectEvents = SignalRequests

RestartMethod = "graceful" / "forced" / "restart" / "disconnected"
               / "cancel-graceful" / extensionRestartMethod
extensionRestartMethod = PackageExtensionRM

```

```

PackageExtensionRM = packageName "/" 1*32(ALPHA / DIGIT / HYPHEN)
RestartDelay = 1*6(DIGIT)

extensionParameter = VendorExtensionParameter
                    / PackageExtensionParameter
                    / OtherExtensionParameter
VendorExtensionParameter = "X" ("-" / "+") 1*6(ALPHA / DIGIT)
PackageExtensionParameter = packageName "/"
                            1*32(ALPHA / DIGIT / HYPHEN)
; не может начинаться с "x-" или x+"
OtherExtensionParameter = 1*32(ALPHA / DIGIT / HYPHEN)

; Если первый символ - двойная кавычка, то это строка-цитата
parameterString = (%x21 / %x23-7F) *(%x20-7F) ; первым и последним не может
                                                    ; быть пробел
                    / quotedString

MGCPResponse = MGCPResponseLine 0*(MGCPPParameter)
               *2(EOL *SDPinformation)

MGCPResponseLine = responseCode 1*(WSP) transaction-id
                  [1*(WSP) "/" packageName] ; Только для 8xx
                  [WSP responseString] EOL

responseCode = 3DIGIT
responseString = *(%x20-7E)

SuitablePkgExtLCOCharacter = SuitableLCOCharacter

SuitableExtLCOCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
                        / "!" / "'" / "|" / "=" / "#" / "?"
                        / "." / "$" / "*" / "@" / "[" / "]"
                        / "^" / "`" / "{" / "}" / "~"

SuitableLCOCharacter = SuitableExtLCOCharacter / "/"
SuitableExtLCOValChar = SuitableLCOCharacter / ":"

; VCHAR за исключением ", "(", ")", ",", и "="
SuitableEventParamCharacter = %x21 / %x23-27 / %x2A-2B
                             / %x2D-3C / %x3E-7E

; ПРИМЕЧАНИЕ: закодировано с помощью UTF8
quotedString = DQUOTE 0*(quoteEscape / quoteChar) DQUOTE
quoteEscape = DQUOTE DQUOTE
quoteChar = (%x00-21 / %x23-FF)

EOL = CRLF / LF

HYPHEN = "-"

; Вместо этого приложения, см. RFC 2327 для надлежащей грамматики SDP.
SDPinformation = SDPLine CRLF *(SDPLine CRLF) ; см. RFC 2327
SDPLine = 1*(%x01-09 / %x0B / %x0C / %x0E-FF) ; для надлежащего опр.

```

## БИБЛИОГРАФИЯ

- ECCA EuroPacketCable working group requirements document EPC-RequDoc-V10-0501, May 2001: *European Requirements for the Delivery of Time-critical Services over Cable Television Networks*.
- ETSI EG 201 188 V1.2.1 (2000-01): Public Switched Telephone Network (PSTN); Network Termination Point (NTP) analogue interface; Specification of physical and electrical characteristics at a 2-wire analogue presented NTP for short to medium length loop applications.





## СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
<b>Серия J</b>	<b>Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов</b>
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевых протоколов и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи