



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

**МСЭ-Т**

СЕКТОР СТАНДАРТИЗАЦИИ  
ЭЛЕКТРОСВЯЗИ МСЭ

**J.162**

(03/2004)

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

---

## РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ J

**КАБЕЛЬНЫЕ СЕТИ И ПЕРЕДАЧА СИГНАЛОВ ТЕЛЕВИЗИОННЫХ И ЗВУКОВЫХ ПРОГРАММ  
И ДРУГИХ МУЛЬТИМЕДИЙНЫХ СИГНАЛОВ**

Общие рекомендации	J.1–J.9
Общие спецификации для аналоговой передачи звуковых программ	J.10–J.19
Характеристики показателей качества аналоговых каналов для звуковых программ	J.20–J.29
Оборудование и линии, используемые для аналоговых каналов для звуковых программ	J.30–J.39
Цифровые кодеры для аналоговых сигналов звуковых программ	J.40–J.49
Цифровая передача сигналов звуковых программ	J.50–J.59
Каналы для аналоговой телевизионной передачи	J.60–J.69
Аналоговая телевизионная передача по металлическим линиям и соединение с радиорелейными звеньями	J.70–J.79
Цифровая передача телевизионных сигналов	J.80–J.89
Вспомогательные цифровые услуги для телевизионной передачи	J.90–J.99
Эксплуатационные требования и методы для телевизионной передачи	J.100–J.109
Интерактивные системы для распределения цифрового телевидения	J.110–J.129
Транспортирование сигналов MPEG-2 по сетям с пакетной обработкой	J.130–J.139
Измерение качества обслуживания	J.140–J.149
Распределение цифрового телевидения по местным абонентским сетям	J.150–J.159
<b>Проект IP-Cablecom</b>	<b>J.160–J.179</b>
Некоторые аспекты	J.180–J.199
Приложения для интерактивного кабельного телевидения	J.200–J.209

*Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.*

## **Рекомендация МСЭ-Т J.162**

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

#### **Резюме**

Многие операторы кабельного телевидения совершенствуют свои технические средства для обеспечения возможности двусторонней связи и используют эту возможность для предоставления услуг высокоскоростной передачи данных по IP-протоколу согласно Рекомендациям МСЭ-Т J.83 и J.112. Эти операторы в настоящее время хотят расширить такую возможность данной платформы доставки для включения множества критических во времени услуг. Настоящая Рекомендация является одной из серии Рекомендаций, необходимых для достижения этой цели. В ней содержится протокол сигнализации сетевого вызова, необходимый для установления соединений.

Настоящая пересмотренная Рекомендация включает самые последние данные по всем актуальным разработкам с момента первоначального утверждения Рекомендации (март 2001 г.) и изменения, приведенные в Изменении 1 к Рекомендации МСЭ-Т J.162 (февраль 2002 г.).

#### **Источник**

Рекомендация МСЭ-Т J.162 утверждена 15 марта 2004 года 9-й Исследовательской комиссией МСЭ-Т (2001–2004 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 извещения 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.

© ITU 2005

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

## СОДЕРЖАНИЕ

	Стр.
1 Область применения .....	1
2 Ссылки .....	1
2.1 Нормативные ссылки .....	1
2.2 Информативные ссылки.....	2
3 Термины и определения .....	2
4 Сокращения и соглашения .....	3
4.1 Сокращения.....	3
4.2 Соглашения .....	3
5 Введение.....	4
5.1 Связь со стандартами H.323 .....	5
5.2 Связь со стандартами IETF.....	5
6 Интерфейс контроллера медийного шлюза (MGCI).....	6
6.1 Модель и соглашения об именовании .....	6
6.2 Использование протокола SDP.....	13
6.3 Функции управления шлюзами .....	13
6.4 Состояния, восстановление после отказа и состязательные условия .....	37
6.5 Коды завершения и коды ошибок .....	49
6.6 Коды причины.....	51
6.7 Использование параметров LocalConnectionOptions и ConnectionDescriptors..	51
7 Протокол управления медийными шлюзами .....	53
7.1 Общее описание.....	54
7.2 Заголовок команды .....	54
7.3 Формат заголовков ответа.....	68
7.4 Кодирование описания сеанса.....	72
7.5 Передача по протоколу UDP .....	79
7.6 Совмещение передачи запросов и ответов.....	81
7.7 Идентификаторы транзакций и установление трехсторонней связи.....	81
7.8 Временные ответы .....	82
8 Безопасность.....	84
Приложение А – Пакеты событий .....	84
Приложение В – Динамическое качество обслуживания.....	86
Дополнение I – Пример пакета событий .....	94
Дополнение II – Примеры кодирования команд .....	95
II.1 Команда NotificationRequest .....	95
II.2 Команда Notify .....	96
II.3 Команда CreateConnection.....	96
II.4 Команда ModifyConnection .....	98

	<b>Стр.</b>
II.5 Команда DeleteConnection (от агента вызова).....	98
II.6 Команда DeleteConnection (от встроенного клиента).....	99
II.7 Команда DeleteConnection (от агента вызова для нескольких соединений) .....	99
II.8 Команда AuditEndpoint.....	99
II.9 Команда AuditConnection.....	100
II.10 Команда RestartInProgress.....	101
Дополнение III – Пример последовательности операций для вызова.....	102
Дополнение IV – Режим соединения.....	108
Дополнение V – Информация о совместимости.....	111
Дополнение VI – Дополнительные примеры пакетов событий.....	112
Дополнение VII – Пакеты событий.....	121
Дополнение VIII – Применение протокола NCS к терминалу IPAT сети SCN.....	127
VIII.1 Обзор.....	127
VIII.2 Архитектура терминала IPAT.....	127
VIII.3 Требования к электрическим и физическим интерфейсам.....	128
VIII.4 Пакет NCS для сообщений протокола сети SCN с интерфейсом V5.....	129
VIII.5 Обеспечение конфигураций.....	137
VIII.6 Поддержка европейского линейного пакета.....	137
VIII.7 Примеры последовательностей операций при вызовах.....	138
Дополнение IX – Поддержка тарификации для сигнализации NCS проекта IPCablecom.....	148
IX.1 Цели.....	148
IX.2 Пакет автоматической тарификации.....	148
IX.3 Случаи использования – примеры последовательности операций при вызовах.....	151
IX.4 Термины.....	153
Библиография.....	154

# Рекомендация МСЭ-Т J.162

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

### 1 Область применения

В настоящей Рекомендации приводится описание профиля прикладного программного интерфейса, интерфейса контроллера медийного шлюза (MGCI) и соответствующего протокола управления медийными шлюзами (MGCP) для управления клиентами, встроенными в систему передачу речи по IP-протоколу (VoIP), из внешних элементов управления вызовами. Протокол MGCP предполагает наличие архитектуры управления вызовами, при которой "интеллект" управления вызовами находится вне шлюзов, и им оперируют внешние элементы управления вызовами. Профиль, описанный в настоящей Рекомендации, будет именоваться протоколом сигнализации сетевого вызова (NCS).

Настоящая Рекомендация основана на протоколе управления медийными шлюзами (MGCP) версии 1.0 (документ RFC 2705), являющемся результатом объединения простого протокола управления шлюзами и семейства протоколов управления IP-устройствами (IPDC). Настоящая Рекомендация технически совместима с соответствующей спецификацией CableLabs PacketCable.

### 2 Ссылки

Указанные ниже Рекомендации МСЭ-Т и другие источники содержат положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники могут подвергаться пересмотру; поэтому всем пользователям данной Рекомендации предлагается изучить возможность применения последнего издания Рекомендаций и других источников, перечисленных ниже. Список действующих в настоящее время Рекомендаций МСЭ-Т регулярно публикуется. Ссылка на документ в данной Рекомендации не придает ему как отдельному документу статус Рекомендации.

#### 2.1 Нормативные ссылки

- ITU-T Recommendation G.168 (2002), *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, *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 (2002), *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 (2004), *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 Информативные ссылки

- 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*.
- ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems*.

## 3 Термины и определения

В настоящей Рекомендации определяются следующие термины:

**3.1 кабельный модем:** Кабельный модем – это оконечное устройство уровня 2, которым завершается абонентское окончание соединения согласно Рекомендации МСЭ-Т J.112.

**3.2 проект IPcablecom:** Проект МСЭ-Т, охватывающий архитектуру и серию Рекомендаций, которые обеспечивают доставку критических во времени интерактивных услуг по сетям кабельного телевидения.



## 4 Сокращения и соглашения

### 4.1 Сокращения

В настоящей Рекомендации используются следующие сокращения:

API	Прикладной программный интерфейс
CPE	Оборудование в помещении пользователя
DTMF	Двухтональный многочастотный
IP	Межсетевой протокол (IP-протокол)
MGCI	Интерфейс контроллера медийного шлюза
MGCP	Протокол управления медийными шлюзами
MIB	Информационная база управления
MTA	Адаптер мультимедийного терминала
MWD	Максимальная задержка на ожидание
NCS	Сигнализация сетевого вызова
КТСОП	Коммутируемая телефонная сеть общего пользования
RTP	Протокол реального времени
SDP	Протокол описания сеанса
UDP	Протокол дейтаграмм пользователя

### 4.2 Соглашения

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 рассматриваемый элемент (вопрос) является безусловным требованием в настоящей Рекомендации.
"НЕ ДОЛЖЕН"	Это словосочетание означает, что рассматриваемый элемент (вопрос) безусловно 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 данный элемент (вопрос), но, прежде чем выбрать другой курс, следует понять 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 событие – целесообразным, но, прежде чем перейти к реализации какой-либо линии поведения, которая отмечена этим словосочетанием, следует понять все последствия и тщательно взвесить все обстоятельства.
"МОЖЕТ"	Это слово или прилагательное "ФАКУЛЬТАТИВНЫЙ" означает, что данный элемент (вопрос) действительно носит факультативный (необязательный) характер. Один поставщик может принять решение включить этот элемент (вопрос), поскольку того требует, например, тот или иной рынок или этот элемент улучшает изделие; другой же поставщик может тот же элемент (вопрос) опустить.

## 5 Введение

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

- две или несколько традиционных аналоговых линий доступа к сети передачи речи по IP-протоколу (VoIP);
- одну или несколько видеолиний к сети VoIP (подлежит дальнейшему изучению).

Использование встроенных клиентов может не ограничиваться только жилыми помещениями. Например, их можно использовать также в учреждениях. Встроенные клиенты используются для линейного доступа и как таковые предположительно имеют линейное оборудование, например аналоговые линии доступа для связанных с ними обычных телефонов в противоположность магистральным шлюзам.

Протокол MGCP предполагает наличие архитектуры управления вызовами, при которой "интеллект" управления вызовами находится вне шлюзов, и им оперируют внешние элементы управления вызовами, называемые агентами вызовов. В протоколе MGCP предполагается, что эти элементы управления вызовами, или агенты вызовов (CA), будут синхронизированы друг с другом для передачи согласованных команд шлюзам, которые находятся под их управлением. 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 в шлюзе, 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 факультативно 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 в профиле NCS протокола MGCP рассматриваются только физические конечные точки.

Соединения являются двухточечными. Двухточечное соединение – это связь между двумя конечными точками с целью передачи данных между ними. После установления этой связи для обеих конечных точек между этими точками может иметь место 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, обнаружение определенных событий, например сигнала снятия телефонной трубки (токовые состояния абонентского шлейфа), и генерирование определенных сигналов, например сигнала контроля отправки 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 пользователей и т. д. 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 этой системы одним шлюзом VoIP. Эта система состоит из агента вызова, который может быть распределен или не распределен по нескольким компьютерным платформам, и из множества шлюзов. В конфигурации по Рекомендации МСЭ-Т H.323 эта распределенная шлюзовая система может сопрягаться с одной стороны с одной или несколькими линиями POTS, а с другой стороны – с совместимыми по Рекомендации H.323 системами, как показано ниже:

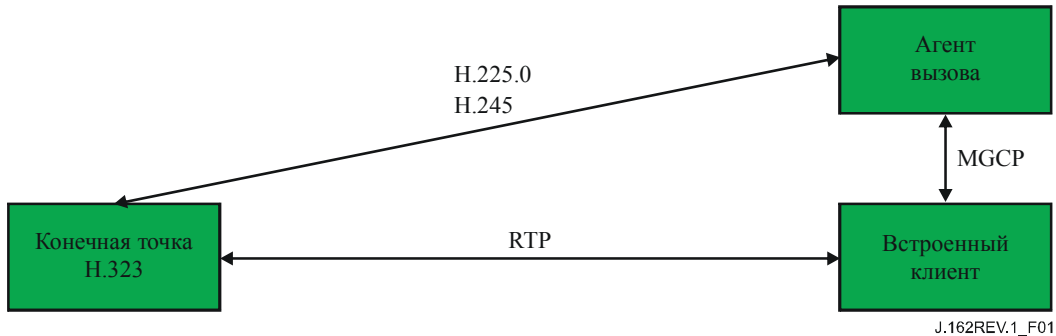


Рисунок 1/J.162 – Связь со стандартами H.323

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 согласно Рекомендации МСЭ-Т H.323. Поэтому сигналы для вызовов по Рекомендации МСЭ-Т H.225 и мультимедийные сигналы по Рекомендации МСЭ-Т H.245 маршрутизируются к агенту вызова.

## 5.2 Связь со стандартами IETF

В то время как Рекомендация МСЭ-Т H.323 служит в качестве признанного стандарта для терминалов 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: работа продолжается;
- протокол инициирования сеанса (SIP), RFC 2543;
- протокол поточной передачи в режиме реального времени (RTSP), RFC 2326.

Последние три спецификации фактически являются альтернативными стандартами сигнализации, которые предусматривают передачу описания сеанса заинтересованной стороне. Протокол SAP используется администраторами многоадресных сеансов для распределения описания многоадресного сеанса большой группе получателей. Протокол SIP используется для приглашения отдельного пользователя принять участие в сеансе двухточечной или многоадресной связи. Протокол RTSP используется для обеспечения взаимодействия сервера, предоставляющего данные в режиме реального времени. Во всех трех случаях описание сеанса выполняется согласно протоколу SDP; когда передаются звуковые сигналы, 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:

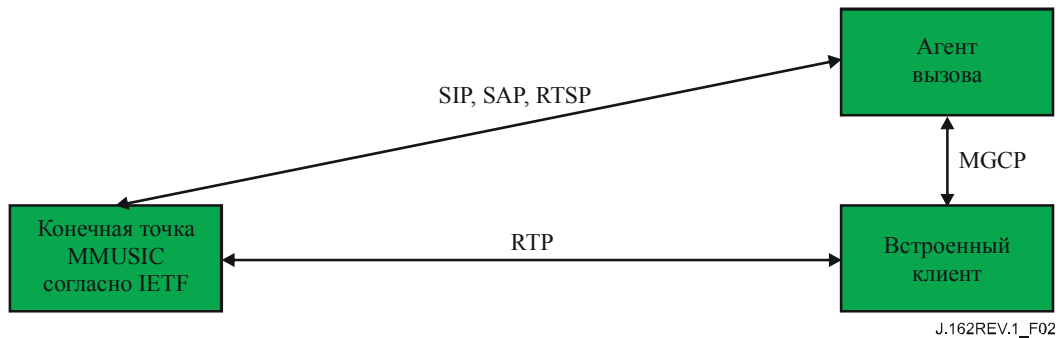


Рисунок 2/J.162 – Связь со стандартами IETF

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

## 6 Интерфейс контроллера медийного шлюза (MGCI)

Функции интерфейса MGCI предусматривают управление соединением, управление конечными точками, сообщения о контроле 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 два компонента, 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 стандарте RFC 1034, которое содержит базовую часть; таким образом, примером доменного имени может служить конструкция:

```
MyEmbeddedClient.cablelabs.com
```

Кроме того, имя domain-name может быть точно-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, одну конечную точку 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 в коде ASCII, составленными из букв, цифр или других печатных знаков, за исключением знаков, используемых в качестве разграничителей в именах конечных точек ("/", "@"), знаков, используемых для подстановки ("\*", "\$"), и знаков пробела.
- Операция подстановки представляется либо звездочкой ("\*"), либо знаком доллара ("\$") для членов ряда имен, которые должны быть подставлены. Таким образом, если полное локальное имя конечной точки выглядит как:

term1/term2/term3

и один из членов этого имени подставляется, тогда локальное имя конечной точки выглядит следующим образом:

term1/term2/\*                   если подставляется term3;  
term1/\*/\*                       если подставляются term2 и term3.

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

- Выполнение подстановки допускается только справа; таким образом, если какой-либо член подставляется, тогда все члены 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 член имел подстановочный знак "доллар", то все члены справа от этого члена также должны иметь подстановочные знаки "доллар".
- Член, представленный знаком "звездочка", должен интерпретироваться следующим образом: "использовать *все* значения этого члена, известные в области применения рассматриваемого встроенного клиента".
- Член, 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 деталь 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 субчлены, 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.

#### 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 член и, как максимум, два члена.

- Член 1 (term1) ДОЛЖЕН быть членом "aaln" или подстановочным знаком. Следует отметить, что использование подстановочного знака для члена 1 может относиться к любому или 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.
- Член 2 (term2) ДОЛЖЕН быть числом от единицы до числа, представляющего 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 члена, то им будет член 1.
- Если член 1 *не является* подстановочным знаком, тогда подстановочный знак "доллар" (относящийся к "любому одному значению") считается относящимся к члену 2 (term2), то есть "aaln" эквивалентно "aaln/\$".
- Если член 1 *является* подстановочным знаком, тогда тот же подстановочный знак предполагается использовать и для члена 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.

Как минимум, идентификаторы вызовов ДОЛЖНЫ быть уникальными 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 управления медийными шлюзами предназначен 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, подобно именам 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 у DNS. Они ДОЛЖНЫ запрашивать службу DNS об обновлении информации, если время существования истекло.

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 управляет этой конечной точкой. 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 текущему "уведомленному объекту", к конечной точке (точкам) которого относится команда. При запуске "уведомленный объект" ДОЛЖЕН быть установлен на обеспечиваемое значение. Большинство команд, посылаемых агентом вызова, имеют возможность неявным образом именовать "уведомленный объект" путем использования параметра "NotifiedEntity" (уведомленный объект) "Уведомленный объект" ДОЛЖЕН оставаться в том же состоянии, пока либо не будет принят новый параметр "NotifiedEntity", либо не произойдет перезагрузка конечной точки. Если "уведомленный объект" для конечной точки является пустым или не был установлен неявным образом<sup>1</sup>, тогда этот объект будет установлен по умолчанию на исходный адрес последней команды обработки соединения или запроса уведомления, принимаемого для конечной точки. Таким образом, контроль не изменит "уведомленный объект".

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

#### 6.1.5 Отображения цифр набора номера

Агент вызова может запросить шлюз осуществлять сбор цифр, набираемых пользователем. Эта функция предназначена для использования аналоговых линий доступа домашними шлюзами для сбора цифр, набираемых пользователем; она также может быть использована для сбора кодов доступа, номеров кредитных карточек и других номеров, запрашиваемых службами управления вызовами. Конечные точки ДОЛЖНЫ поддерживать отображения цифр набора номера, как определено в данном пункте.

Альтернативная процедура состоит в использовании шлюза, уведомляющего агента вызова о набранных цифрах, как только они будут набраны; 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.

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

Однако проблема при использовании такого метода накопления набранных номеров состоит в том, что для шлюза трудно прогнозировать, сколько номеров необходимо накопить перед передачей. Например, используя телефон на столе, 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 дается 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| 8xxxxxxx| #xxxxxxx| *xx| 91xxxxxxxxxxx| 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 вхождению позиции
              | Position "." -- соответствует произвольному числу
              -- вхождений позиции, включая 0
String    ::= StringElement | StringElement String
StringList ::= String | String "|" StringList
DigitMap  ::= String | "(" StringList ")"
```

Согласно этому синтаксису, отображение цифр определяется либо с помощью (нечувствительной к регистру) "строки", либо с помощью "списка строк", по которым шлюз будет пытаться найти наиболее близкое возможное соответствие. Независимо от приведенного выше синтаксиса использование таймера в текущий момент допустимо, если только он появляется в последней позиции в строке<sup>2</sup>. 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 к 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 одному элементу в отображении цифр и неполное соответствие другому элементу).

<sup>2</sup> Например, запись "123T" и "123[1-2T5]" удовлетворяют этому правилу, а запись "12T3" – не удовлетворяет.



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 освобождается. 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 Т – это таймер ввода цифр, который может быть использован двояким образом:

- Когда таймер Т 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 будет иметь место либо соответствие отображению цифр, либо несоответствие. В этом случае таймер Т функционирует как таймер времени между передачами цифр.
- Когда таймер Т используется без отображения цифр, он запускается сразу же и просто аннулируется (но не перезапускается), как только вводится цифра. В этом случае таймер Т может быть использован как таймер времени между передачами цифр, когда используется передача с перекрытием.

При использовании с отображением цифр таймер Т принимает одно из двух значений –  $T_{par}$  или  $T_{crit}$ . Когда для строки цифр требуется хотя бы еще одна цифра, чтобы соответствовать какой-либо из кодовых комбинаций в отображении цифр, таймер Т принимает значение  $T_{par}$ , соответствующее частичной синхронизации набора цифр. Если время таймера истекло, что требуется для получения соответствия, то таймер Т принимает значение  $T_{crit}$ , соответствующее критической синхронизации. Когда таймер Т 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. Процесс обеспечения может изменить эти оба значения.

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

Конечные точки ДОЛЖНЫ поддерживать отображение цифр длиной не менее 2048 байтов 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 отметить, что подробная информация об используемом отображении цифр будет зависеть от области, в которой находится шлюз, и, таким образом, отображение цифр является программируемым. Отображение цифр, когда оно предоставляется агентом вызова, ДОЛЖНО соответствовать описанию 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 применение определенных сигналов в конечной точке, например тонального сигнала готовности к набору номера.

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 определенной 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 иметься один или несколько пакетов, и у каждого типа конечной точки имеется пакет по умолчанию, с которым он связан.

Имена событий состоят из имени пакета и кода события; одни и те же коды событий могут быть использованы в разных пакетах, поскольку каждый пакет определит 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

<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".

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 IP-Cablecom. 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 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 точных частот звуковых сигналов, таких как тональные сигналы готовности к набору номера или двухтональные многочастотные (DTMF) сигналы. События должны далее 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). У сигналов ДОЛЖЕН быть определен их тип (включено/выключено, выдержка времени или короткий сигнал), и сигналы выдержки времени ДОЛЖНЫ иметь определяемое по умолчанию значение выдержки времени – см. п. 6.3.1.

В дополнение к использованию пакетов проекта IP-Cablecom те, кто реализует систему, МОГУТ приобрести опыт путем определения экспериментальных пакетов. Имя пакета для экспериментальных пакетов ДОЛЖНО начинаться с двух символов – "x-" или "X-"; в проекте IP-Cablecom 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 каком пакете.

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

Когда сигнал должен подаваться в соединение, то к имени события добавляется имя соединения при использовании в качестве разграничителя коммерческого знака "собачка" (@) как в конструкции:

X/rt@0A3F58

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 обозначение:

```
x/rt@*
```

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, то шлюз ДОЛЖЕН генерировать или обнаруживать событие в соединении, которое создается или модифицируется в текущий момент времени. Примером такого условного обозначения является обозначение:

```
x/rt@$
```

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).

## 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 тех случаев в настоящей Рекомендации, когда 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 мультимедийной конференцсвязи. Профиль NCS будет только поддерживать установление аудио- и видеосоединений с использованием типов среды "аудио" и "видео". В настоящее время специфицированы только "аудиосоединения".

## 6.3 Функции управления шлюзами

В данном пункте приводится описание команд протокола MGCP в форме интерфейса API типа удаленного вызова процедуры (RPC), 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, показанные в данном пункте, обеспечивают описание высокого уровня функционирования протокола MGCP, и с их помощью описывается пример интерфейса API типа RPC, 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 сообщения MGCP, которыми обмениваются, включая форматы сообщений и используемое кодирование, определяются в пункте о протоколе (п. 7). Встроенные клиенты ДОЛЖНЫ осуществлять их реализацию так, как это определено.

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 DTMF-сигналы в определенной конечной точке.
- Затем шлюз будет использовать команду Notify (уведомить) для информирования агента вызова о том, когда имеют место запрашиваемые события в заданной конечной точке.
- Агент вызова может использовать команду CreateConnection (создать соединение) для установления соединения, которое заканчивается в конечной точке в самом шлюзе.

- Агент вызова может использовать команду ModifyConnection (модифицировать соединение) для изменения параметров, связанных с ранее установленным соединением.
- Агент вызова может использовать команду DeleteConnection (исключить соединение) для исключения 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 (проконтролировать конечную точку) и AuditConnection (проконтролировать соединение) для проверки состояния "конечной точки" и всех связанных с нею соединений. Обычно желательно управление сетью сверх возможностей, предоставляемых этими командами, например желательно иметь информацию о состоянии встроенного клиента. Ожидается, что такие возможности поддерживаются путем использования простого протокола управления сетью (SNMP) и определения базы MIB, что выходит за рамки настоящей Рекомендации.
- Шлюз может использовать команду RestartInProgress (перезапуск в процессе осуществления) 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 у 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 мультимедийных пакетов по каждому соединению управляется параметром "режим" (mode), который может быть установлен в состояние "только передача" (sendonly), "только прием" (recvonly), "передача/прием" (sendrecv), "конференция" (confncc), "неактивный" (inactive), "копирование" (replicate), "сетевой шлейф" (netwloop) или "проверка целостности сети" (netwtest). Параметр "режим" определяет, могут ли быть переданы и/или получены мультимедийные пакеты по соединению. Протокол RTCP не зависит от режима соединения; подробная информация содержится в Рекомендации МСЭ-Т J.161.

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

- Мультимедийные пакеты, 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 мультимедийных пакетов, полученных по этим соединениям, также определяется параметрами режима:

- Мультимедийные пакеты, полученные в пакетах данных через соединения в режиме "неактивный", "только передача" и "копирование", отбрасываются.
- Мультимедийные пакеты, 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 мультимедийные пакеты, 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 микшере, выходят за рамки настоящей Рекомендации.
- В дополнение к мультимедийным пакетам, полученным из конечной точки, мультимедийные пакеты, переданные конечной точке, перемешиваются и передаются по всем другим соединениям для конечной точки, режим которой соответствует состоянию "копирование". Сюда СЛЕДУЕТ включить мультимедийные пакеты, генерируемые сигналами, подаваемыми в конечную точку.
- Мультимедийные пакеты, полученные в пакетах данных через соединение в режиме "сетевой шлейф" или "проверка целостности сети", будут передаваться в обратном направлении по соединению, как описано ниже.

Если режим установлен в состояние "сетевой шлейф", звуковые сигналы, принятые из соединения, будут воспроизведены в обратном направлении по тому же соединению. Режим "сетевой шлейф" ДОЛЖЕН просто функционировать 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 правильного функционирования. До передачи в обратном направлении этот сигнал ДОЛЖЕН пройти через внутреннее декодирование и повторное кодирование. Что касается аналоговых линий доступа, то этот сигнал будет звуковым сигналом, который НЕ ДОЛЖЕН передаваться дальше к телефону, подключенному к аналоговой линии доступа, независимо от текущего состояния телефонной трубки этого телефона, то есть положена эта 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 независимо от метода кодирования, используемого для этого другого "соединения". Однако от соединения в режиме "копирование" ТРЕБУЕТСЯ только поддержка результирующего мультимедийного потока при кодировании<sup>5</sup> согласно Рекомендации G.711. Поддержка режима "конференция" является факультативной; 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 передать уведомление при возникновении определенных 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 тогда решить, 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 выполняется 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. 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 ни пусты, ни пропущены. Такие шлюзы должны посылать в ответ код ошибки 503, если они получают команду NotificationRequest с подстановочным знаком "all of", которую они не могут обработать по причине наличия ошибки.

Параметр **NotifiedEntity** (уведомленный объект) – это факультативный параметр, который определяет новый "уведомленный объект" для конечной точки. При его использовании ДОЛЖНО быть определено все 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 идентификатор агента вызова.

<sup>5</sup> Соединение в режиме "копирование" может быть использовано, например, для поддержки "проверки занятой линии" с минимальным влиянием ресурсов на 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 хост-узла).

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 иначе, сигналы подаются в конечную точку; однако некоторые сигналы могут подаваться в соединение. Примерами сигналов могут служить следующие сигналы<sup>8</sup>:

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

В зависимости от режима использования сигналы делятся на различные типы:

- Включено/выключено (On/off (OO)) – Когда эти сигналы подаются, они делятся, пока не будут выключены. Это может происходить только в результате наличия нового параметра SignalRequests, при котором сигнал выключается (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 выключаться, пока от агента вызова не поступит явная команда сделать это; сигналы OO будут отключаться 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)) – Когда эти сигналы подаются, они делятся, пока либо не будут аннулированы (в результате поступления некоторого события или в результате невключения в последующий [возможно, пустой] список сигналов), либо 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", о котором агент вызова мог послать или не послать запрос об уведомлении. 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 определяемое для них значение выдержки времени по умолчанию, которое может быть изменено процессом обеспечения. Кроме того, период выдержки времени может быть предоставлен сигналу в качестве параметра. Нулевое значение указывает на то, что период выдержки времени бесконечен. Сигналу TO, в котором происходит сбой после его запуска, но до генерирования по событию "операция завершена", будет генерироваться событие "сбой операции", которое будет содержать имя сигнала (имена сигналов) с выдержкой времени<sup>10</sup>.
- Короткий сигнал (Brief (BR)) – Продолжительность 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 для цифры "1" закончится.

По умолчанию сигналы относятся к конечным точкам. Если сигнал, поданный в конечную точку, приводит к генерированию мультимедийного потока (аудио, видео и т. д.), то этот поток НЕ ДОЛЖЕН направляться по любому соединению, связанному с этой конечной точкой, независимо от режима

<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.

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 определять разное поведение.

Если сигнал подается в соединение, которое получило параметр `RemoteConnectionDescriptor` (см. п. 6.3.3), тогда мультимедийный поток, генерируемый этим сигналом, ДОЛЖЕН направляться по соединению *независимо* от текущего режима соединения. Если параметр `RemoteConnectionDescriptor` получен не был, то шлюз ДОЛЖЕН вернуть информацию об ошибке (код ошибки 527 – отсутствие параметра `RemoteConnectionDescriptor`).

Когда предоставляется список (возможно, пустой) сигналов, этот список полностью заменяет текущий список активных сигналов выдержки времени. Такие 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 для таких сигналов выдержки времени не будет затронут. 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`.

Определяемые в текущее время сигналы представлены в Приложении А.

Параметр **RequestedEvents** – это список событий, которые по запросу должен обнаруживать шлюз в конечной точке. Если не оговорено иначе, 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 DTMF-сигнала (или импульсные знаки).

События, определяемые в настоящее время, содержатся в Приложении А.

С каждым событием связано одно или несколько **действий**, определяющих действие, 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

<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`.

неформально рассматривать как всегда включенные неявным образом в список параметра RequestedEvents с действием "уведомление", хотя будет выполняться самое обычное обнаружение событий<sup>13</sup>. Устойчивые события идентифицируются как таковые по их 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 может иметь место только один раз. В приведенной ниже таблице определены допустимые сочетания действий:

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

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

Параметр **DigitMap** (отображение цифр набора номера) – это факультативный 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, то переменная "текущая строка набора номера" инициализируется по нулевой строке. 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 действием "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 состоять 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 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, тогда "текущая строка набора номера" будет освобождаться; 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 иницилирующие события не накапливаются; чтобы это сделать, это действие 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

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

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, тогда может быть использовано действие "Embedded ModifyConnection" ("встроенная модификация соединения"). Это действие может включать список изменений режимов соединений, каждое из которых содержит для соединения 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 завершенного 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 в силе.

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 все еще иметь место 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 эти параметры SignalRequests (например, продолжительность и частота цифры при DTMF-наборе номера), 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 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 определяется именем пакета, тогда 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).

---

<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.

Агент вызова может послать команду `NotificationRequest`, запрашиваемый список сигналов которой является пустым. Эта команда имеет эффект прекращения 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 факультативный 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 этой 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 факультативный 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 факультативно 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 полностью определяться именем конечной точки, включая доменное имя шлюза. Локальная часть этого имени НЕ ДОЛЖНА использовать условный подстановочный знак.

Параметр **NotifiedEntity** (уведомленный объект) – это факультативный 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 хост-узла) своего параметра `NotifiedEntity`, если только абсолютное доменное имя было получено в инициирующей команде `NotificationRequest`. В этом случае системе CMS СЛЕДУЕТ принять это значение. Этот 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`, инициирующей 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 иницирующей 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 факультативно 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 "вида" 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 управляют одними и теми же шлюзами; 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 проведения учета и передачи сообщений.

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 будет возвращено в параметре **SpecificEndPointId** (идентификатор конкретной конечной точки) ответа. В этом случае конечная точка с присвоенным идентификатором ДОЛЖНА быть в рабочем состоянии и НЕ ДОЛЖНА иметь еще какие-либо соединения.

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

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

- **Метод кодирования:** ДОЛЖЕН быть определен по меньшей мере с одним значением список литеральных имен для алгоритма сжатия (метод кодирования/декодирования), используемого для передачи и приема мультимедийных пакетов по соединению. Элементы в списке упорядочены по уровню приоритета. Конечная точка ДОЛЖНА выбрать один из кодеков, и его СЛЕДУЕТ выбирать согласно указанному приоритету. Если конечная точка принимает по соединению какой-либо мультимедийный пакет, закодированный с использованием другого метода кодирования, она может отбросить его. Подробная информация, касающаяся процесса выбора кодека, содержится в п. 6.7.
  - Конечная точка ДОЛЖНА дополнительно указывать, какие из оставшихся алгоритмов сжатия она хочет поддерживать как альтернативы; см. подробную информацию в п. 7.4.
  - Список допустимых методов кодирования определен в Рекомендации МСЭ-Т J.161. ДОЛЖНЫ использоваться имена литералов, которые определены в таблице 3. Неизвестные алгоритмы сжатия в случае их получения СЛЕДУЕТ игнорировать.
- **Период пакетирования:** Период пакетирования в миллисекундах, как определено в стандарте SDP (RFC 2327), МОЖЕТ быть задан с одним десятичным значением. Если используется этот спецификатор, тогда для всех кодеков, допускаемых структурой `LocalConnectionOptions`, ДОЛЖЕН использоваться один и тот же период пакетирования. Следует отметить, что адаптер МТА НЕ ДОЛЖЕН выбирать кодек с периодом пакетирования, отличающимся от описываемого здесь периода. Если необходимы разные периоды пакетирования для разных кодеков, это поле использоваться НЕ ДОЛЖНО. Значение относится как к переданному, так и принятому мультимедийному пакету. Следует отметить, 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 и той же структуре LCO как поле "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, как определено в стандарте SDP (RFC 2327), 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`. Этот список применяется, даже если несколько методов кодирования имеют одно и то же значение. Первый элемент в списке ДОЛЖЕН быть десятичным числом. Когда используется дефис, то рассматриваемый кодек ДОЛЖЕН использовать тот же период пакетирования, что и один из других 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, при котором они используются. Последовательные элементы в списке периодов пакетирования ДОЛЖНЫ быть упорядочены согласно соответствующим методам кодирования. Значения относятся как к переданному, так и принятому мультимедийному пакету. 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 и той же структуре LCO как поле "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 структур LCO, где число кодеков, заданных в поле "метод кодирования", отличается от числа элементов в поле "период многократного пакетирования".

- **Эхокомпенсация:** Должна ли первоначально использоваться эхокомпенсация на линейной стороне или нет<sup>16</sup>. Этот параметр может иметь значение "on" ("включено") (когда запрошена эхокомпенсация) или "off" ("выключено") (когда эхокомпенсация выключена). Этот параметр является факультативным. Когда этот параметр опущен, встроенный клиент ДОЛЖЕН применять эхокомпенсацию. Встроенному клиенту СЛЕДУЕТ в дальнейшем включать и выключать эхокомпенсацию в соответствии с Рекомендациями МСЭ-Т V.8 и V.25 при обнаружении данных в диапазоне звуковых частот. Относительно повторного включения эхокомпенсации см., например, Рекомендацию МСЭ-Т G.168. После окончания передачи данных в диапазоне звуковых частот обработка сигналов эхокомпенсации ДОЛЖНА возвращаться к текущему значению параметра эхокомпенсации.
- **Тип услуги:** Описывает класс услуги, которая будет использоваться для передачи мультимедийных пакетов по соединению, путем кодирования 8-битового значения параметра "тип услуги" IP-заголовка в виде двух шестнадцатеричных цифр. Этот параметр является факультативным. Когда этот параметр опущен, то применяется значение по умолчанию 0xA0 (если не будет обеспечено другое значение) согласно пяти установленным битам приоритетов в IP-сети.
- **Подавление пауз:** Должно ли использоваться подавление пауз (молчания) в направлении передачи или нет. Этот параметр может иметь значение "on" ("включено") (когда пауза должна подавляться) или "off" ("выключено") (когда пауза не должна подавляться). Этот параметр является факультативным. Когда этот параметр опущен, то по умолчанию подавление пауз не используется.

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

- **D-QoS GateID** (идентификатор логического затвора для динамического качества обслуживания): GateID для логического затвора, который был установлен в краевом маршрутизаторе. GateID – это 32-битовый идентификатор, закодированный в виде строки, содержащей до 8 шестнадцатеричных знаков. Обычно этот параметр является факультативным, но когда должны выполняться резервирование и/или предоставление ресурсов для D-QoS, он становится обязательным. Наличие этого параметра означает, что для этой команды должно обеспечиваться D-QoS, а его отсутствие означает, что D-QoS не должно предоставляться.
- **Резервирование ресурсов для D-QoS:** Позволяет явным образом контролировать, должно ли выполняться или не выполняться резервирование и/или предоставление ресурсов для D-QoS в направлении передачи и/или приема. Этот параметр является факультативным и может иметь 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 факультативным, 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, применяемыми 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, пока не будет получен параметр RemoteConnectionDescriptor. Следует отметить, что пока не будет получен параметр RemoteConnectionDescriptor, зарезервированные и предоставленные ресурсы должны базироваться на

<sup>16</sup> Эхокомпенсация на стороне пакетов не поддерживается.

кодеке (кодеках), выбранном локальным образом. По получении параметра RemoteConnectionDescriptor список кодеков, которые фактически 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. Когда 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 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 разных видах резервирования; однако в данный 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 факультативным. Однако он ДОЛЖЕН использоваться встроенным клиентом при резервировании ресурсов, если предоставлен агентом вызова.
- **Параметр ReserveDestination** (адресат резерва): Этот факультативный 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.

<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).

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

- **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.

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 за счет процесса обеспечения.

**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, когда 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 ходы. Как отмечалось выше, встроенные клиенты МОГУТ отбрасывать любой принимаемый мультимедийный пакет, который закодирован 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 ДОЛЖНА быть возвращена информация об ошибке (код ошибки 517 – неподдерживаемый режим). Кроме того, если соединение еще не получило параметр RemoteConnectionDescriptor, то ДОЛЖНА быть возвращена информация об ошибке, когда 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.

**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.

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 непонятных 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 факультативными. 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, когда 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 быть уверенным 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, когда 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 получили отказ. В нашем примере команда CreateConnection должна получить отказ, 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 получить отказ 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 факультативно 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.

### 6.3.4 Команда ModifyConnection

Команда ModifyConnection (модификация соединения) используется для модификации характеристик "представления" соединения со стороны шлюза. Это "представление" 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 быть полностью классифицированным именем конечной точки. В локальном имени НЕ ДОЛЖЕН использоваться условный подстановочный знак.

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 эхокомпенсаций.

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 затвора 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 будет использован 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 факультативным, и для него может быть определено несколько значений. Когда 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 факультативным. 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 факультативным. 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.

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 свободен в выборе соответствующих значений для неопределенных параметров, которые должны быть изменены<sup>19</sup>.

Адресная информация протокола RTP, предоставляемая в параметре RemoteConnectionDescriptor, точно определяет удаленный адрес RTP получателя мультимедийных пакетов 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, то встроенному клиенту СЛЕДУЕТ также только принимать потоки мультимедийных пакетов (и RTCP) от заданного IP-адреса. Все потоки мультимедийных пакетов, принимаемые от 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 факультативными. 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 получили отказ.

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

### 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])
```

Идентификатор конечной точки в данном виде команды DeleteConnection ДОЛЖЕН быть полностью классифицирован. Подстановочные знаки использоваться НЕ ДОЛЖНЫ.

В общем случае, когда соединение имеет два конца, эта команда должна посылаться обоим шлюзам, участвующим в соединении. После исключения соединения потоки мультимедийных пакетов в сети с

<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.

коммутацией пакетов, ранее поддерживаемые соединением, уже не доступны. Все мультимедийные пакеты, принятые для старого соединения, просто отбрасываются, и новые мультимедийные пакеты 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. Этими 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.
- **Число переданных октетов:** Общее число октетов полезной нагрузки (то есть не включая заголовок или заполнение), 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.
- **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. Значение равно нулю, если, например, соединение всегда было установлено в режим "только передача".
- **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. Значение равно нулю, если, например, соединение всегда было установлено в режим "только передача".
- **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, соединение всегда было установлено в режим "только передача".
- **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, соединение всегда было установлено в режим "только передача".
- **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 RTCP, и отметкой времени по протоколу NTP получателей, измеренное при приеме сообщений. Среднее получается путем суммирования всех оценок и последующего деления на число принятых сообщений RTCP. 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 RTPSP, 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 **NotifiedEntity**, **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются факультативными. Они могут быть использованы агентом вызова для передачи запроса на уведомление, который связан с исключением 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 получили отказ.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 6.5), которое факультативно 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 полностью классифицирован. Условные подстановочные знаки использоваться НЕ ДОЛЖНЫ.

Параметр **Reason-code** (код причины) – это текстовая строка, начинающаяся с числового кода причины и факультативно 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**. Когда для соединения были выполнены одно или несколько резервирований ресурсов и/или предоставлений ресурсов для D-QoS, встроенный клиент будет освобождать 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 факультативно сопровождается комментарием.

### 6.3.7 Команда DeleteConnection (от агента вызова для нескольких соединений)

Для одновременного исключения нескольких соединений агент вызова может использовать некоторую разновидность команды DeleteConnection. Эта команда может быть использована для исключения 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.

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

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

При этом 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 исключения соединения мультимедийные потоки сети с коммутацией пакетов, ранее поддерживаемые этим соединением, уже не доступны. Все мультимедийные пакеты, полученные по старому соединению, просто отбрасываются, и новые мультимедийные пакеты для потока не посылаются. Когда для этого соединения были выполнены одно или несколько резервирований ресурсов и/или предоставлений ресурсов для D-QoS, встроенный клиент будет освобождать 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 факультативно сопровождается комментарием.

### 6.3.8 Контроль

Протокол MGCP основан на архитектуре централизованного управления вызовом, 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, 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 контроля, позволяющий агенту вызова контролировать состояние конечной точки и соединения и восстанавливать характерные для протокола функциональные возможности конечной точки.

Для встроенных клиентов определены две команды контроля:

- **AuditEndPoint** (контроль конечной точки): Используется агентом вызова для определения состояния конечной точки.
- **AuditConnection** (контроль 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. Ожидается, 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 (контроль 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 находится под контролем. Условный подстановочный знак "any of" ("любой из") использоваться НЕ ДОЛЖЕН.

Условный подстановочный знак "all of" ("все из") может быть использован для контроля 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 параметр возврата **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 проконтролирована следующая характерная для конечной точки информация:

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.

В ответ, в свою очередь, будет включена информация о каждом из параметров, для которого была запрошена информация для контроля:

- **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 текущее время.

- **SignalRequests** (запросы сигналов) – Список сигналов выдержки времени, которые активны в текущее 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 исчерпанием 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 хост-узла) своего параметра NotifiedEntity, если ему было предоставлено только абсолютное доменное имя (включая имя хост-узла) 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, событие находится в состоянии "телефонная трубка снята" в примере с линейным пакетом, если конечная точка находится в таком состоянии. 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.
- **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 факультативной. 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 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 согласно 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).
  - **Эхокомпенсация** – Поддерживается или нет эхокомпенсация.
  - **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 определяемым по умолчанию пакетом.

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



- **Режимы** – Список поддерживаемых режимов соединения.
- **Динамическое качество обслуживания** – Поддерживается или нет динамическое качество обслуживания.
- **Обеспечение безопасности** – Поддерживаются или нет услуги обеспечения безопасности в проекте IP-Cablecom. 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 факультативно 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` (контроль соединения).

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

← AuditConnection(EndpointId
                  , ConnectionId
                  [, RequestedInfo])
```

Параметр `EndpointId` определяет конечную точку, которая контролируется; подстановочные знаки использоваться НЕ ДОЛЖНЫ. 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.
```

В ответ, в свою очередь, будет включена информация о каждом из параметров, для которого была запрошена информация для контроля:

- **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) – Режим текущего 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) – Параметр RemoteConnectionDescriptor, который был предоставлен шлюзу для соединения.

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

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.
- 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 одних и тех же конечных точек. Когда передается эта команда, шлюз сразу же начинает пытаться установить новые соединения 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.

Факультативный 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 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 либо 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" перезапуска.

Факультативный 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 факультативно 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 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 указывает на ошибку, то рассматриваемый перезапуск еще не завершен. Если в ответе получен код 521 (переадресация 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 завершился, 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 удостовериваться, 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 перезапушен.

### 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 ней 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 обеспечиваемое значение. Когда для конечной точки получена команда 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 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 сообщение Notify в ответ, дейтаграмма однако посылается по адресу источника новой принятой команды независимо от параметра NotifiedEntity для любой из команд.
- Когда "уведомленный объект" относится к доменному имени, которое рассылается по нескольким IP-адресам, конечные точки могут коммутироваться по каждому 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, станут в конце концов "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".
- Когда другой (резервный) агент вызова взял на себя управление группой конечных точек, предполагается, что недоступный агент вызова будет взаимодействовать и будет синхронизированным с резервным агентом вызова, чтобы передать управление 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 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. Шлюзы ДОЛЖНЫ хранить в памяти список ответов, которые они послали последним транзакциям, и список транзакций, выполняемых в текущее время. Понятие "последняя транзакция" определяется здесь значением  $T_{hist}$ , которое задает число секунд, в течение которых должны сохраняться ответы на старые транзакции. По умолчанию значение  $T_{hist}$  составляет 30 секунд.

Идентификаторы транзакций входящих команд сначала сравниваются с идентификаторами транзакций последних ответов. Если выявлено совпадение, то шлюз 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 шлюз не выполняет транзакцию; последующая обработка зависит от рассматриваемой команды. Если это команда CreateConnection или ModifyConnection, то шлюз посылает временный ответ. Если это какая-либо другая команда, она просто игнорируется. В любом случае конечный ответ будет представлен, когда завершится выполнение команды.

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

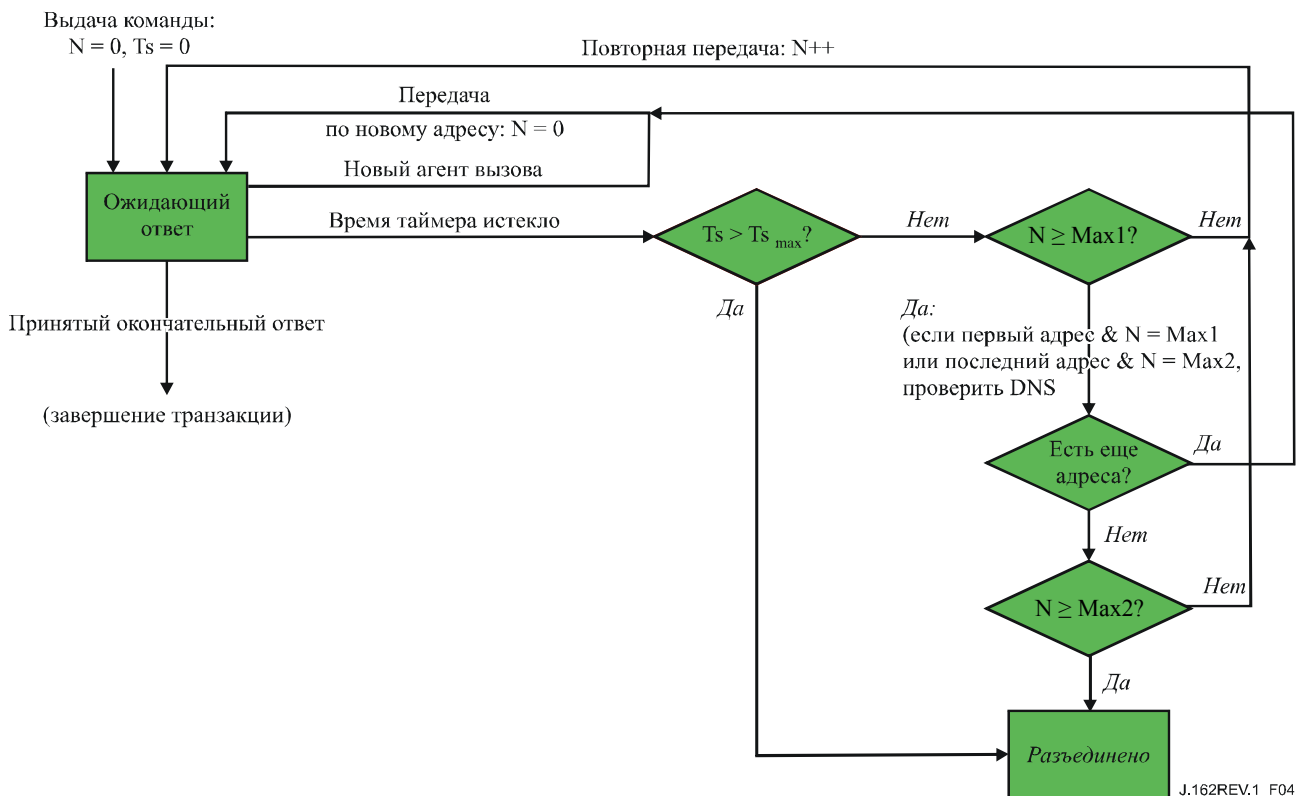
Этот механизм повторения используется для защиты от следующих возможных ошибок четырех типов:

- 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 много 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  $Max2$ , который должен быть установлен на большее значение.

В классическом алгоритме повторной передачи будет просто подсчитываться число последовательных повторений и делаться заключение, что соединение нарушено, 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, ниже):

- 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 неподтвержденных команд для конечной точки (точек) этому новому агенту вызова. Ответы на новые или старые команды все еще передаются по адресу источника команды.
- До любой повторной передачи проверяется, чтобы время, истекшее с момента передачи начальной дэйтаграммы, не превышало  $T_{s_{max}}$ . Если истекшее время больше  $T_{s_{max}}$ , 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 этом локальном списке.
- Если для попыток передачи интерфейсов 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 нет, то конечная точка (точки), управляемая этим агентом вызова, разъединяется (становится несвязной). Когда конечная точка становится несвязной, она ДОЛЖНА затем инициировать "несвязную" процедуру, как описано в п 6.4.3.6.



**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 на ожидание для обнаружения восстановления после отказа. Повторная передача должна продолжаться 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; вероятно, общая задержка на ожидание в 30 секунд является приемлемой.

Следует отметить, что между  $T_{s_{max}}$ ,  $T_{t_{hist}}$  и максимальным транзитным временем  $T_{p_{max}}$  существует тесная связь. В частности, чтобы предотвратить многократное выполнение повторно переданных команд, ДОЛЖНО выполняться следующее соотношение:

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

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 между отправителем и получателем соглашения по этим значениям 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 обеспечения.

Кроме того, процесс обеспечения ДОЛЖЕН иметь возможность нейтрализации запросов системы DNS для одного или обоих порогов  $Max1$  и  $Max2$ .

### 6.4.3 Состязательные условия

В данном пункте приводится описание действий по протоколу MGCP в состязательных условиях.

Прежде всего, протокол MGCP имеет дело с состязательными условиями посредством понятия "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, в протоколе MGCP не предполагается, 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 управления.

#### 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 "уведомленному объекту".

В этот момент шлюз 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 когда параметр DetectEvents не был получен, это события, 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, 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.

<sup>23</sup> Следует отметить, что команда Notify не может быть объединена со встроенной командой 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 видов поведения:

- 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 одну 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 подвергнутое 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" в предположении, 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 должным образом. Если шлюз должен передать новую команду Notify до ответа на ранее полученную 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. Если это так, то шлюз немедленно передаст команду Notify, в которой будет сообщено о всех событиях, которые были накоплены в списке "наблюдаемых событий" до инициирующего события включительно, оставив 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 встроеного 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 как команда NotificationRequest. Команды обработки соединения, не содержащие запроса на уведомление, не влияют на приведенную выше процедуру, как и процедура не влияет на них.

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

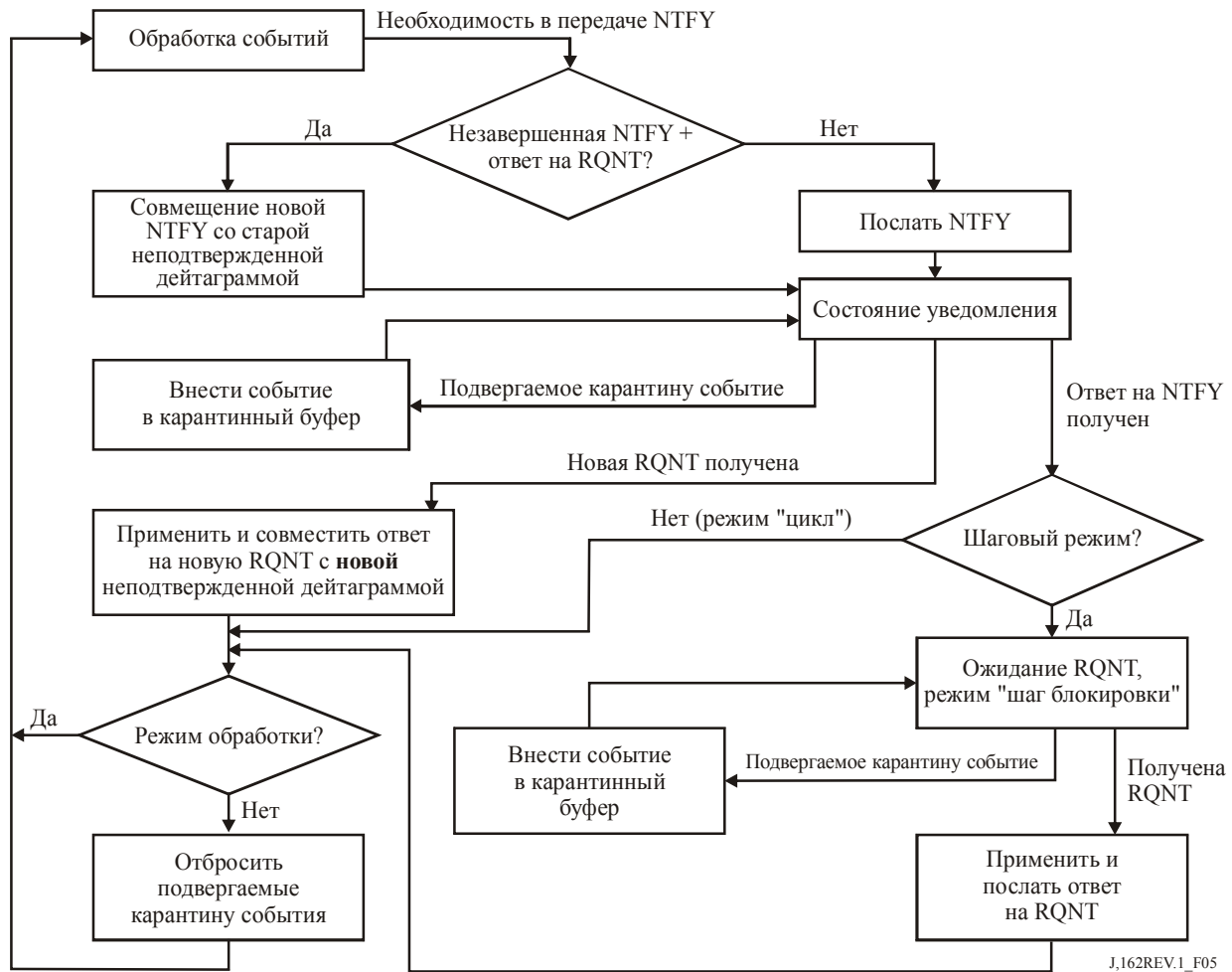


Рисунок 5/J.162 – Процедуры для карантинного списка

Агентам вызова СЛЕДУЕТ представить ответ на успешное сообщение Notify и новую команду NotificationRequest в одной и той же дейтаграмме, используя механизм совмещения запросов и ответов<sup>24</sup>.

#### 6.4.3.2 Явное обнаружение

Ключевым элементом состояния некоторых конечных точек является положение телефонной трубки (трубка снята/положена). И хотя в состязательных условиях сигнализации NCS события изменения состояния "телефонная трубка положена/снята" являются устойчивыми и все еще может иметь место несовпадение состояний, например, когда пользователь решает позвонить (снять телефонную трубку), в то время как агент вызова находится в процессе запроса шлюза относительно поиска событий "телефонная трубка снята" и, возможно, подачи вызывного сигнала (состояние "помехи приему вызова звонком", хорошо известное в телефонной связи).

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

<sup>24</sup> Поставщики, которые предпочитают не следовать положениям настоящей Рекомендации, должны тщательно изучить сценарии отказов агента вызова.

Кроме того, в отдельных определениях сигналов может быть определено, 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 наличие условия проверяется во время приема запроса на уведомление, а о действительном событии, вызвавшем текущее условие, либо могло быть уже сообщено, либо это событие могло быть проигнорировано раньше, 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 информация об ошибке возвращается. Как и все другие транзакции, команда NotificationRequest ДОЛЖНА выполняться как неделимая транзакция; таким образом, любые изменения, инициированные в результате выполнения команды, ДОЛЖНЫ быть восстановлены.

Другое состязательное условие может возникнуть, когда команда Notify выдается непосредственно перед приемом команды NotificationRequest шлюзом. Параметр RequestIdentifier используется для коррелирования команд Notify с командами NotificationRequest, тем самым позволяя агенту вызова определять, 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, после этого 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 интерпретация разупорядочения**

В протоколе MGCP базовому транспортному протоколу не передаются функции гарантирования порядка следования команд, посылаемых шлюзу или конечной точке. Это свойство способствует максимально возможной своевременности действий, но имеет ряд недостатков:

- 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 во второй раз.

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 посылаться параллельно, например согласно модели, где каждой конечной точкой управляет собственный процесс или собственный поток.
- 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 подтверждения 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 исключенному 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 иницилирующей командой NotificationRequest.
- 5) В некоторых случаях команда DeleteConnection с неявным или явным использованием подстановочного знака, которая применяется 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 посылаться до тех пор, пока не будет получен ответ на команду DeleteConnection с подстановочным знаком.
- 6) Когда команды встроены одна в другую, ДОЛЖНЫ соблюдаться требования по 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.

#### **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, 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 действия пользователя 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, тогда 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 инициируется 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 шлюз входит в "несвязное" состояние при выполнении процедуры перезапуска, то ДОЛЖНА быть выполнена "несвязная" процедура, описанная в п. 4.4.3.6, за исключением того, что во время этой процедуры посылается сообщение "перезапуск", а не сообщение "разъединено".

Ожидается, что каждая конечная точка в шлюзе будет иметь обеспечиваемого агента вызова, то есть "уведомленный объект", для направления к нему начального сообщения перезапуска. Когда управление множеством 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 перезапускается несколько конечных точек в шлюзе, 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 продолжительность 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 инициируется, 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 ( $Td_{mit}$ ), 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) Шлюз затем 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 ( $Td_{min}$ ), чтобы ограничить скорость выполнения процедуры.
- 4) Если "несвязная процедура" все еще составляет 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 ( $Td_{max}$ ), например 600 секунд, и шлюз снова 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 полностью использоваться совмещение передачи запросов и ответов. Затем агент вызова может, например, решить проконтролировать конечную точку или просто освободить все соединения для конечной точки.

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 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 обеспечение 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 применяются описанные выше положения, касающиеся таймера, ограничивающего скорость перезапуска. Несвязная конечная точка может пожелать послать команду (помимо команды RestartInProgress), находясь в разъединенном состоянии. Это действие окажется успешным, как только агент вызова снова станет достижимым; при этом возникает вопрос, что делать с такой командой. Одна крайность, когда конечная точка могла бы сразу отбросить эту команду; однако это было бы не вполне правильным действием, когда агент вызова был действительно доступен, но конечная точка еще не завершила "несвязную" процедуру (можно рассмотреть, например, случай, когда команда NotificationRequest была только 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 контролировать". 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, если конечная точка снова связная, 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 иницируется "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 аннулировать 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), воспроизводить тональный сигнал переупорядочения или даже инициировать воспроизведение загружаемого файла звуковых сигналов в несвязных 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.

## 6.5 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 исключено.
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".
500	Транзакция не могла быть выполнена, так как конечная точка неизвестна.
501	Транзакция не могла быть выполнена, так как конечная точка к этому не готова.
502	Транзакция не могла быть выполнена, так как конечная точка не имеет достаточных ресурсов.

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

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

Код	Значение
503	Подстановочный знак "all of" ("все из") полностью не поддерживается. Транзакция содержалась в этом подстановочном знаке; однако шлюз не поддерживает это полностью. Следует отметить, что в текущий момент это допустимо только для непустого параметра NotificationRequests.
505	Неподдерживаемый параметр RemoteConnectionDescriptor. Его СЛЕДУЕТ использовать, когда в параметре RemoteConnectionDescriptor не поддерживаются один или несколько параметров или одно или несколько значений.
506	Невозможно удовлетворить как параметр LocalConnectionOptions, так и параметр RemoteConnectionDescriptor. 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 из-за команды, содержащейся в нераспознанном расширении.
512	Транзакция не могла быть выполнена, так как шлюз не оборудован для обнаружения одного из запрошенных событий.
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 исключенному).
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.



## 6.6 Коды причины

Коды причины используются шлюзом при исключении соединения для информирования агента вызова о причине исключения 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 контроля.)
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 параметров LocalConnectionOptions и ConnectionDescriptors

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) Агент вызова затем 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 мультимедийные пакеты, поддерживаемые этим соединением:

- Параметр LocalConnectionOptions: предоставляется агентом вызова для управления параметрами мультимедийных пакетов, используемыми шлюзом для соединения. Когда данный параметр предоставлен, шлюз должен соответствовать этим параметрам мультимедийных пакетов до тех пор, пока либо не будет исключено соединение, либо не будет получена команда ModifyConnection.
- Параметр RemoteConnectionDescriptor: предоставляется агентом соединения для переноса параметров мультимедийных пакетов, поддерживаемых другой стороной соединения. Когда данный параметр предоставлен, шлюз должен соответствовать этим параметрам мультимедийных пакетов до тех пор, пока либо не будет исключено соединение, либо не будет получена команда ModifyConnection.
- Параметр LocalConnectionDescriptor: предоставляется шлюзом агенту вызова для переноса параметров мультимедийных пакетов, которые он поддерживает для соединения. Когда данный параметр предоставлен, шлюз должен соответствовать параметрам мультимедийных пакетов до тех пор, пока либо соединение не будет исключено, либо шлюз не выдаст новый параметр LocalConnectionDescriptor.

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);
  - период многократного 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 периодов многократного 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 мультимедийных пакетов) затем 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 пустой, 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 строк потоков мультимедийных пакетов, то фиксируется отказ при согласовании кодеков и генерируется ответ об ошибке (рекомендуется код ошибки 534 – отказ при согласовании кодеков). Если параметр RemoteConnectionDescriptor содержит информацию о множестве потоков мультимедийных пакетов, 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 множество 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 пустой, 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, шлюз должен 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 создания возможности для 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 реальном масштабе времени. Шлюз возвращает все несогласованные кодеки в строке потока мультимедийных пакетов 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 строке = m означает, что устройство должно быть готово принять мультимедийные пакеты от любого из согласованных кодеков. Шлюз также может посылать мультимедийные пакеты 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 реальном масштабе времени. Поэтому шлюз согласует и возвращает только один кодек в строке потока мультимедийных пакетов SDP (факультативно шлюз также помещает дополнительные поддерживаемые кодеки в атрибуте 'X-rc-codex' протокола SDP) и резервирует ширину полосы пропускания для одного согласованного кодека в строке потока мультимедийных пакетов согласно Рекомендации МСЭ-Т J.163. Согласно этому методу изменение кодека должно инициироваться системой CMS, чтобы выполнить изменение кодеков, во время которого, согласно Рекомендации МСЭ-Т J.163, переустанавливается результирующее изменение в ширине полосы пропускания.

## 7 Протокол управления медийными шлюзами

В протоколе MGCP интерфейс управления 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, факультативно, 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 установлено иное. Это относится к командам, а также к параметрам и значениям, и при всех сравнениях представлений в верхнем и нижнем регистрах, а также их комбинаций интерпретация этих представлений ДОЛЖНА рассматриваться как одна и та же.

### 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 символами 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

Команда	Код
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 – нераспознанное 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.

Как минимум, идентификаторы транзакций для команд, посылаемых данному встроенному клиенту, ДОЛЖНЫ быть однозначно определены 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. 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 жизни 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 должны использоваться 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 присоединена 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 агента вызова по протоколу MGCP (2727, если только не обеспечено иное). 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 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>	К	См. описание.
CallId	С	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака. Идентификаторы вызовов сравниваются как строки, а не как численные значения.
ConnectionId	I	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака. Идентификаторы вызовов сравниваются как строки, а не как численные значения.
NotifiedEntity	N	Идентификатор в формате кода 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.
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, факультативно сопровождаемой знаком "@", за которым следует доменное имя встроенного клиента, к которому присоединена 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.

<sup>26</sup> Параметр ResponseAsk не был рассмотрен в п. 6.3, поскольку идентификаторы транзакций не видны в приведенном примере интерфейса API. Те, кто реализует систему, могут выбрать другой подход.

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

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

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
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
RequestIdIdentifier	O	O	O	M	M	F	F	F
LocalConnectionOptions	M	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	F
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
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
<sup>a)</sup> Параметры RequestedEvents и SignalRequests являются факультативными 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 вызовов СЛЕДУЕТ всегда представлять обязательные параметры перед факультативными; однако встроенные клиенты НЕ ДОЛЖНЫ иметь сбой, если данная рекомендация не выполняется.

Если у тех, кто реализует систему, 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 – нераспознанное 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 `ResponseAck` (подтверждение ответа) используется для поддержки установления трехсторонней связи, описанного в п. 7.7.1. 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` (варианты локальных соединений) описывают факультативные 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 знаков. Эти литеральные имена ДОЛЖНЫ использоваться и быть эквивалентны определениям кодеков в параметрах RTP (Рекомендация МСЭ-Т J.161). РЕКОМЕНДУЕТСЯ, чтобы также поддерживались хорошо известные варианты литеральных имен кодеков;
- параметр "экокомпенсация", 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 затвора);
- параметр D-QoS Resource Reservation (резервирование ресурсов для QoS), закодированный в виде ключевого слова "dq-rr", 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-битовому 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 факультативно 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 будут разделены 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 будут разделены 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. В шифрокомплекте пробелы не допускаются. Следующий пример иллюстрирует использование шифрокомплекта:

62/51

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 о возможностях конечной точки при контроле. Кодирование возможностей базируется на кодировании вариантов 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 "эхокомпенсация", закодированный в виде ключевого слова "e", за которым следуют двоеточие и значение "on" ("включено"), если эхокомпенсация 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-rtsp", 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;confnrc,
dq-gi, sc-rtp: 64/51;0360/51, 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 IPCalecom. 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, 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	JI	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 вызовов, 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 факультативно 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	Шлюз должен только передавать пакеты.
M: recvonly	Шлюз должен только принимать пакеты.
M: sendrecv	Шлюз должен передавать и принимать пакеты.
M: confnrc	Шлюз должен передавать и принимать пакеты согласно режиму конференцсвязи.
M: inactive	Шлюз не должен ни посылать, ни принимать пакеты.
M: replcate	Шлюз должен только передавать пакеты согласно режиму копирования.
M: netwloop	Шлюз должен вводить конечную точку в режим сетевого шлейфа.
M: netwtst	Шлюз должен вводить конечную точку в режим проверки целостности сети.

### 7.2.2.8 Кодирование имен событий/сигналов

Имена событий/сигналов состоят из факультативного имени пакета, отделенного от имени действительного события косой чертой (/). Имя события может факультативно 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

X/hu	Переход в состояние "телефонная трубка положена" в примере линейного пакета
X/0	Цифра 0 в примере линейного пакета
hf	Флэш-сигналы в предположении, что приводимый в качестве примера линейный пакет – это пакет по умолчанию для конечной точки
X/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**

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

X/rt@*	Контроль посылки вызова по всем соединениям для конечной точки
X/rt@\$	Контроль посылки 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. 6.1. 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, что, например, 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 кодирования 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 факультативным и, возможно, предоставляется в другом порядке. Следующий пример иллюстрирует использование встроенного действия NotificationRequest с примером 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 отображению цифр, несоответствие или выдержка времени между цифрами. Следует отметить, что поскольку событие "телефонная трубка положена" является устойчивым событием, оно все еще будет обнаруживаться, и о нем будет посылаться уведомление, хотя здесь это не описано.

Встроенное действие ModifyConnection соответствует формату:

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

Следующий пример иллюстрирует использование встроенного действия ModifyConnection с примером линейного пакета:

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

Это означает: послать флэш-сигналы, 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 определять требования помимо этих минимальных возможностей. Для комбинаций сигналов (помимо этого минимального требования), 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. Двое последовательных двойных кавычек 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 сигнальные параметры.

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, то их коды разделяются запятой, как, например:

```
S: rg, rt@FDE234C8.
```

#### 7.2.2.11 Параметр `ObservedEvents`

Параметры `ObservedEvents` (наблюдаемые события) предоставляют список событий, которые были запрошены. 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 установлена в исходное состояние; 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 проконтролированы. Также поддерживаются значения, перечисленные в таблице 17:

Таблица 17/J.162 – Значения параметра `RequestedInfo`

Параметр <code>RequestedInfo</code>	Код
<code>LocalConnectionDescriptor</code>	LC
<code>RemoteConnectionDescriptor</code>	RC



Например, если возникнет желание проконтролировать 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 действительных значений:

Q: loop  
Q: process  
Q: discard, loop

### 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 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 действия.

### 7.2.2.16 Параметр `ResourceID`

Параметр `ResourceID` (идентификатор ресурса) – это параметр возврата, используемый для динамического качества обслуживания для информирования об идентификаторе ресурса, присвоенном рассматриваемому шлюзу. 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), "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 факультативно сопровождается заголовками, кодирующими параметры ответа.

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

200 1201 ОК

В таблице 18, ниже, сведены параметры ответов, наличие которых в заголовке ответа является обязательным или факультативным 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 – "факультативная", 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>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>	O <sup>a)</sup>
CallId	F	F	F	F	F	F	O	F
ConnectionId	M	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	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

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

<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>
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
VersionSupported	F	F	F	F	F	O	F	O
MaxMGCPDatagram	F	F	F	F	F	O	F	F
LocalConnection Descriptor	M	O	F	F	F	F	O	F
RemoteConnection Descriptor	F	F	F	F	F	F	O	F
а) Параметр ResponseAck НЕ ДОЛЖЕН использоваться во всех других ответах, кроме окончательного ответа, выдаваемого после временного ответа для рассматриваемой транзакции. В этом случае присутствие параметра ResponseAck ДОЛЖНО инициировать сообщение подтверждения ответа; все предоставляемые значения параметра ResponseAck будут проигнорированы.								

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 ОК

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 ОК
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 ОК
```

### 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 ОК
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 AuditEndpointy

В случае команды AuditEndPoint (контроль 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 не существует значение, например отображение цифр набора номера, все еще будут обеспечиваться. 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 содержится в одной строке. 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 (контроль соединения) ответ может сопровождаться информацией для каждого из запрашиваемых параметров. Параметры, для которых в настоящее время не существует значение, все еще будут предоставляться. 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 контролируемого соединения, этот дескриптор соединения будет представлен только с полем 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 аудиоуслуге;
- если параметр среды установлена на "видео", то описание сеанса относится к видеоуслуге.

Для аудиоуслуги шлюз будет рассматривать информацию, предоставляемую в протоколе SDP для среды "аудио", а для видеоуслуги шлюз будет рассматривать информацию, предоставляемую в протоколе SDP для среды "видео".

### 7.4.1 Использование аудиоуслуги по протоколу SDP

В только речевом шлюзе необходимо описывать сеансы, которые используют лишь только одну среду передачи – аудио. Ниже описываются параметры протокола SDP, существенные для речевого применения. 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).

Предоставленный профиль протокола SDP описывает использование протокола описания сеанса в сигнализации NCS. Общее описание и пояснения к отдельным параметрам можно найти в стандарте RFC 2327; однако ниже уточняется, какие значения конечных точек NCS необходимо предусмотреть для этих полей (передача) и что конечные точки NCS должны делать со значениями, предоставленными или не предоставленными для этих полей (прием).

#### 7.4.1.1 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.

#### 7.4.1.2 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, когда запрошена конфиденциальность.

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

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Session-ID (идентификатор сеанса):

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

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

---

<sup>27</sup> Поскольку конечные точки NCS не знают, когда запрашивается конфиденциальность, они ДОЛЖНЫ всегда использовать дефис.

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:** ДОЛЖНА быть использован тип "IP4".

**Прием:** Это поле СЛЕДУЕТ проигнорировать.

Address (адрес):

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

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

#### 7.4.1.3 Имя сеанса (s=)

*s= <session-name>*

*s= -*

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

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

#### 7.4.1.4 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.

#### 7.4.1.5 Идентификатор URI (u=)

*u= <URI>*

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

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

#### 7.4.1.6 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.

#### 7.4.1.7 Дата соединения (c=)

Дата соединения состоит 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, когда прикладная программа будет получать мультимедийный поток. 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 мультимедийного потока (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 мультимедийного потока (потоков), для которого он предназначен.

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

*b= <modifier> : <bandwidth-value>*

*b= AS : 64*

**Передача:** Информация о полосе пропускания является факультативной в протоколе SDP, но ее всегда СЛЕДУЕТ включать<sup>28</sup>. Когда используется отображение rtpmap или неизвестный кодек<sup>29</sup>, информация о полосе пропускания ДОЛЖНА быть использована.

**Прием:** СЛЕДУЕТ включить информацию о полосе пропускания. Если модификатор 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 требуемой максимальной шириной полосы пропускания мультимедийного потока в килобитах в секунду.

**Прием:** ДОЛЖНА присутствовать требуемая максимальная ширина полосы пропускания мультимедийного потока 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 шлюза может не дать разрешение на использование соответствующей ширины полосы пропускания.

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



#### 7.4.1.9 Время, время 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 останова СЛЕДУЕТ использовать нуль. НЕ СЛЕДУЕТ использовать время повторений и временные зоны; если они используются, это должно соответствовать стандарту RFC 2327.

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

#### 7.4.1.10 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.

#### 7.4.1.11 Атрибуты (a=)

a= <attribute> : <value>

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

a= rtpmap : 0 PCMU / 8000

a= fmtp:<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>

a= recvonly

a= sendrecv

a= sendonly

a=ptime

**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.

rtpmap:

**Передача:** Это поле, если используется, 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 нагрузке для ее приемного конца, хотя могут быть случаи, когда это невозможно. Если адаптер МТА установил соответствие динамического типа полезной нагрузки данному методу кодирования для его мультимедийного потока на приеме, то в данном соединении этот тип полезной нагрузки НЕ ДОЛЖЕН в дальнейшем соответствовать другому методу кодирования для его принятого мультимедийного потока.

**Прием:** Это поле, если используется, ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Этот атрибут определяет соответствие кодека типу полезной нагрузки RTP, которую готовы принять на другом конце соединения. Адаптеры МТА поэтому ДОЛЖНЫ использовать это соответствие типу полезной нагрузки при передаче мультимедийных пакетов по данному соединению. Когда эта информация получена в команде CreateConnection, адаптеру МТА СЛЕДУЕТ использовать это соответствие типу полезной нагрузки для своего собственного приемного конца (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, этот адаптер МТА ДОЛЖЕН оставить соответствие своему собственному типу полезной нагрузки, как оно есть (так, чтобы использовались типы асимметричной полезной нагрузки).

fntp:

**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 среды передачи. 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-codecs:

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 (приоритетным) 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 среды передачи (m=).

mptime:

Этот атрибут является атрибутом уровня среды передачи, который определяется в проекте 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 применяется не одна скорость пакетирования (например, в неречевых 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 среды передачи в строке "m=". Для форматов среды передачи, 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=".

X-rc-secret:

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

**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:** Передаются закрытая информация 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-csuites-rtsp:

X-rc-csuites-rtcp:

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

**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:** Передается 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.

recvonly:

**Передача:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2543. В настоящее время этот атрибут не должен предоставляться встроенным клиентом.

**Прием:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2543.

sendrecv:

**Передача:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2543.

**Прием:** Это поле должно использоваться в соответствии со стандартом RFC 2543.

sendonly:

**Передача:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2543, за исключением того, что IP-адрес и номер порта НЕ ДОЛЖНЫ быть нулевыми. В настоящее время этот атрибут не должен предоставляться встроенным клиентом.

**Прием:** Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2543.

ptime:

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

**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.

X-rc-nrkey:

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

**Передача:** Это поле содержит 16-битовый целочисленный счетчик числа событий повторного ввода с клавиатуры. Это поле может потребоваться, когда используется обеспечение безопасности речевой передачи. 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.

#### 7.4.1.12 Уведомления о среде передачи (m=)

Уведомления о среде передачи (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 среды передачи "аудио".

**Прием:** Принятым типом среды передачи 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 среды передачи:

**Передача:** ДОЛЖЕН быть использован соответствующий тип среды передачи, 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 предпочтет передать. Каждый тип полезной нагрузки однозначно соответствует (статическое или динамическое соответствие) кодеку. 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 `rtmap`, и ДОЛЖНЫ выполняться руководящие указания, содержащиеся в п. 7.4.1.11.

**Прием:** В соответствии со стандартом RFC 2327. В частности, форматы среды передачи указывают на тип (типы) полезной нагрузки, который готовы принять на другом конце данного соединения.

## 7.4.2 Использование видеослужбы по протоколу SDP

Детали использования видеослужбы по протоколу SDP подлежат дальнейшему изучению.

## 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 сверху вниз, РЕКОМЕНДУЕТСЯ, чтобы агент вызова всегда явным образом указывал порт MGCP для использования в сообщении сигнализации NCS (и не полагался на определяемый по умолчанию порт).

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.

---

<sup>30</sup> Каждая конечная точка может быть обеспечена адресом и портом отдельного агента вызова.

### 7.5.2 Стратегия повторных передач

В настоящей Рекомендации отсутствуют спецификации 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 повторных передач, 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:

- Он должен удвоить оцениваемое значение средней задержки AAD.
- Он должен вычислить случайное значение, равномерно распределенное между 0,5 AAD и AAD.
- Он должен 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 обеспечения.

### 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 в IP-протоколе не требуется, чтобы хост-узел 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 вызовов, так и к конечным точкам. Агенты вызовов 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 недоставку 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 пожелать 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 вызовов 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 вызовов 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 распределены всем транзакциям, которые исходят от любого агента вызова и которые переданы отдельному шлюзу в течение периода в  $T_{hist}$  секунд. Шлюзы могут 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", полученные 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

<sup>31</sup> В противоположность команде с параметром "подтверждение ответа".

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) Факультативно, 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) Факультативно, один или несколько неподтвержденных ответов, где  $term_{old}$  содержал подстановочный знак "all" ("все"), а  $term_{new}$  охвачен подстановочным знаком в имени  $term_{old}$ .
  - d) Факультативно, 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) Факультативно, 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) Факультативно, 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.
- Если  $term_{new}$  есть "aaln/1", а  $term_{old}$  есть "\*", тогда старый ответ может быть подтвержден по правилу 1c.
- Если  $term_{new}$  есть "aaln/\*", а  $term_{old}$  есть "\*", тогда старый ответ может быть подтвержден по правилу 2a.
- Если  $term_{new}$  есть "aaln/\$", а  $term_{old}$  есть "aaln/\*", тогда старый ответ может быть подтвержден по правилу 3a.

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.

Кроме того, 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 случаях<sup>32</sup>. В сигнализации NCS в качестве транспортного протокола используется протокол UDP, а надежность обеспечивается избирательными повторными передачами с использованием выдержек

---

<sup>32</sup> Например, когда ресурсы резервируются и предоставляются внешним источником как часть транзакции.



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 вопросом, когда требуется быстрое обнаружение потери сообщения без большой служебной нагрузки.

Чтобы преодолеть эту трудность, ДОЛЖЕН генерироваться временный ответ, если ожидается, что время завершения транзакции превысит малый 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 руководящего указания 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 в противном случае будет иметь место в системе.

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.

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 обеспечения.

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 передаче обратно к конечной точке. Таким образом, SMS ДОЛЖНА генерировать ответ "подтверждение ответа" всякий раз, когда она принимает 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 этого ответа "подтверждение ответа" подчиняется тем же стратегиям и процедурам выдержки времени и повторных передач, что и ответы на команды (см. п. 6.4); другими словами отправитель окончательного ответа повторно передает 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 разных типов конечных точек, которые в настоящее время определены для встроенных клиентов в проекте IP-Cablecom. Для типов конечных точек для встроенных клиентов, перечисленных в таблице 1, 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 проконтролировано;
"C"	указывает на то, что сигнал/событие могут быть обнаружены/поданы в соединении.
<b>Сигнал</b>	Если в данном столбце для события ничего нет, тогда 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 обеспечения может изменить эти значения по умолчанию.
"BR"	Короткий сигнал. У события короткая известная продолжительность.
<b>Дополнительная информация</b>	Предоставляется дополнительная информация о событии/сигнале, например продолжительность 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.

## Пакеты базового протокола

В базовом протоколе в настоящее время определены следующие пакеты. Эти пакеты применяются ко всем конечным точкам:

- Base (базовый)

### Базовый пакет

Имя пакета: В

Для идентификации событий и сигналов для "базового" пакета для всех типов конечных точек используются следующие коды:

Код	Описание	Событие	Сигнал	Дополнительная информация
oc	Операция завершена	√	–	
of	Ошибка операции	√	–	

**Операция завершена (oc):** Событие "операция завершена" генерируется, когда шлюзу был послан запрос на применение одного или нескольких сигналов типа ТО в конечной точке и один или несколько из этих сигналов были завершены без останова в результате обнаружения запрашиваемого события, такого как переход в состояние "телефонная трубка снята" или набранная цифра. Сообщение о завершении операции может переносить в качестве параметра имя сигнала, время существования которого подошло к концу, как, например:

О: В/oc (mypackage/mysignal)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет содержать также имя соединения, как, например::

О: В/oc (mypackage/mysignal@0A3F58)

Когда запрошено событие завершения операции, оно не может быть параметризовано какими-либо параметрами события. Когда имя пакета опущено, то предполагается имя пакета по умолчанию.

Событие завершения операции может быть генерировано дополнительно, как определено в базовом протоколе, например, когда успешно завершается выполнение встроенной команды ModifyConnection, как в случае:

О: В/oc (В/С)

**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 когда до истечения выдержки времени в одном или нескольких из этих сигналов произошел сбой. Сообщение о завершении операции может переносить в качестве параметра имя сигнала, в котором произошел сбой, как в случае:

О: В/of (mypackage/mysignal)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет содержать также имя соединения, как, например:

О: В/of (mypackage/mysignal@0A3F58)

Когда запрошено событие "ошибка операции", параметры события не могут быть заданы. Когда имя пакета опущено, предполагается имя пакета по умолчанию.

Событие "ошибка операции" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда произошел сбой при выполнении встроенной команды ModifyConnection, как в случае:

О: В/of (В/С (М (sendrecv (AB2354) )))

### Аудиоинформация

Пакеты событий для аудиоинформации подлежат дальнейшему изучению.

## Видеоинформация

Пакеты событий для видеоинформации подлежат дальнейшему изучению.

## ЦСИС

Пакеты событий для сети ЦСИС базового доступа подлежат дальнейшему изучению.

# 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 затвора) – 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.
- **Gate Location** (местоположение логического затвора) – IP-адрес и порт, куда следует послать сообщение о предоставлении ресурсов D-QoS с использованием протокола RSVP. Адаптер МТА узнает об этом адресе через сообщения D-QoS RSVP.

Идентификатор логического затвора (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 (J.112/RSVP)
<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 (входящего);
- **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:

	МТА-о (J.112/RSVP)
<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;
- когда протокол RSVP используется как протокол резервирования ресурсов, то адресом места назначения (адрес адресата), используемым для сообщения RSVP PATH, будет предоставляемый IP-адрес адресата резерва, до тех пор пока не будет предоставлен параметр RemoteConnectionDescriptor.

### 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;
  - изменения параметраptime;
  - прочее.
- **Правила для D-QoS** представлены в п. 6.3.3.

Как пояснялось выше, конечный автомат будет повторно инициализирован, когда будет получен новый GateID. 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, confnce, netwloop, netwtst	C	C
<b>Параметр "резервирование/предоставление" задан</b>		
sendresv	R	N, R <sup>a)</sup>
recvresv	N, R <sup>a)</sup>	R
snrcresv	R	R
sendcomt	C	N, R <sup>a)</sup>
recvcomt	N, R <sup>a)</sup>	C
snrccomt	C	C
a) 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:

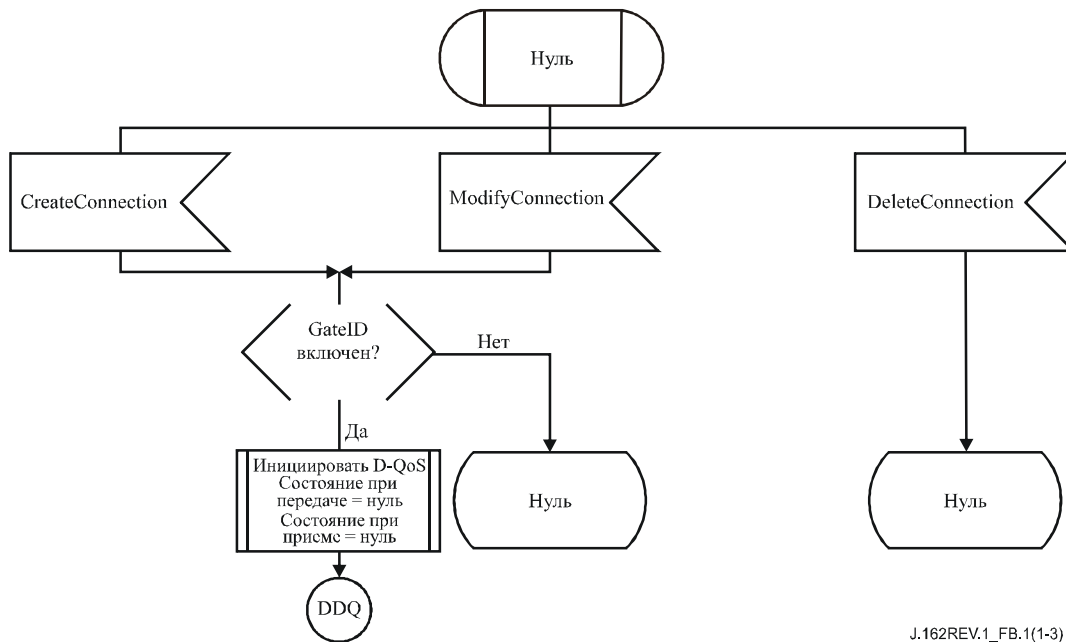
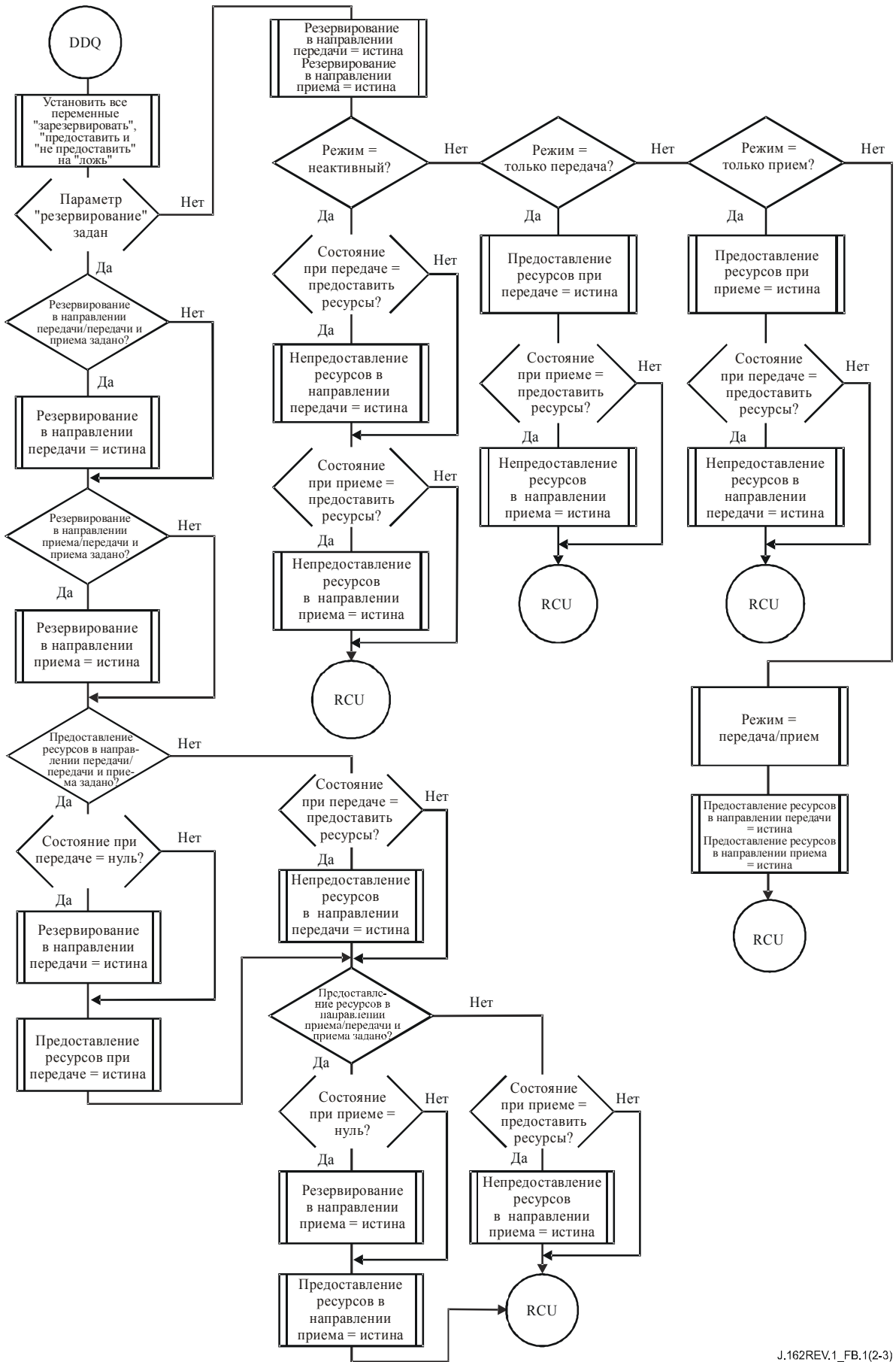


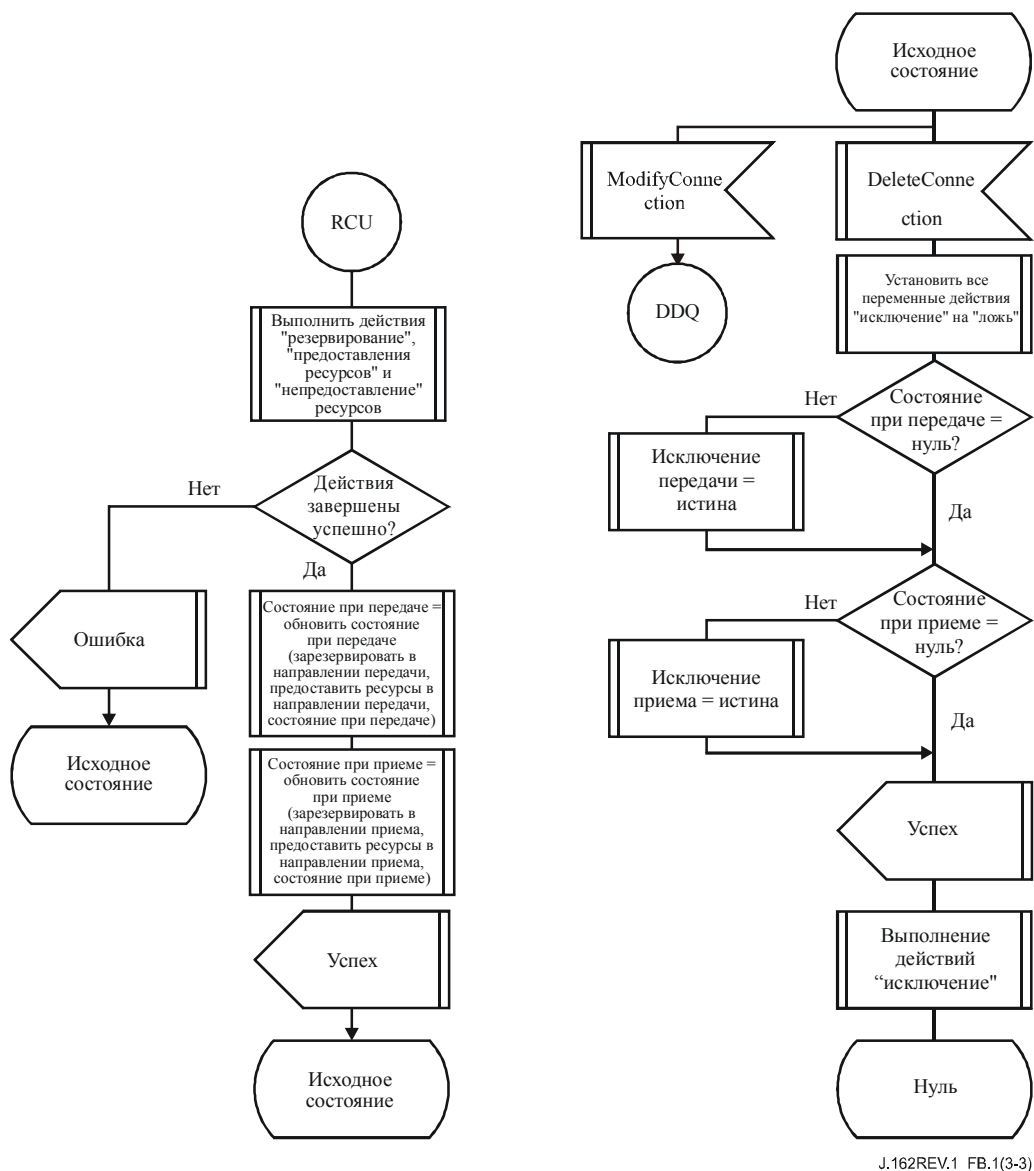
Рисунок В.1/J.162 – Диаграмма состояний NCS/D-QoS  
(лист 1 из 3)



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,
        -- послать к параметру ReserveDestination, если RSVP
    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 if <(J.112 QoS) or (RSVP and ReserveDestination provided )>
    then RR(*)
        -- Использовать классификатор подстановочных знаков,
        -- послать к параметру ReserveDestination, если RSVP
    else ERROR
}

```

#### 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)
        }
        if <(J.112 QoS) or (RSVP and ReserveDestination provided )> then {
            SC(RemoteConnectionDescriptor)
            -- послать к параметру ReserveDestination, если RSVP
        } else ERROR
    } 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 if <(J.112 QoS) or (RSVP and ReserveDestination provided )> then
            RR(*) -- Использовать классификатор подстановочных знаков,
            -- послать к параметру ReserveDestination, если RSVP
        else ERROR
    }
}

```

```

    If <RemoteConnectionDescriptor provided> then
        RC(RemoteConnectionDescriptor)
    else if <(J.112 QoS) or (RSVP and ReserveDestination provided )> then
        RC(*)      -- Использовать классификатор подстановочных знаков,
                  -- послать к параметру ReserveDestination, если RSVP
    else ERROR
}
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 DTMF-сигналов	√	–	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 DTMF-сигналы.

**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 T – это обеспечиваемый таймер, который может быть аннулирован только вводом DTMF-сигнала.

**Визуальный индикатор ожидания сообщения (vmwi):** Сигнал визуального индикатора ожидания сообщения либо активизирует, либо нейтрализует визуальную индикацию ожидания сообщения голосовой почты.

**Подстановочный знак для DTMF-сигналов (X):** Подстановочный знак DTMF-сигналов 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 факультативный 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>33</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|91xxxxxxxxxxx|9011x.T)
S:
Q: process
T: ft
```

Ответ указывает на то, что транзакция была успешной:

```
200 1202 OK
```

<sup>33</sup> Он может быть также пропущенным.

## 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@cal.whatever.net:5678](mailto:ca@cal.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@cal.whatever.net:5678
X: 0123456789AC
O: hd,9,1,2,0,1,8,2,9,4,2,6,6
```

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 используемым 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/hu,oc(N),[0-9](N)
D:
S: vmwi(+)
X: 0123456789B1
N: Call-agent@ca.whatever.net
I: 32F345E2
T: L/hd,L/hu,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/hu" 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(E.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	---	---	→	

Usr-1	EC-1	CA	CDB	ACC	EC-2	Usr-2
		←	---	---	Подтверждение	
		(Вызов установлен)				
		←	---	---	Уведомить	Телефонная трубка положена
		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>34</sup>:

```
CRCX 1202 aaln/1@ec-1.whatever.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: hu, [0-9#*T] (D)
D: (0T | 00T | [2-9]xxxxxxx | 1[2-9]xxxxxxxxxxx | 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
```

---

<sup>34</sup> Фактическое отображение цифр зависит от плана набора номеров в локальной области, а также от абонируемых услуг. Имеющееся отображение цифр следует рассматривать только как пример такого отображения.

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>35</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
```

---

<sup>35</sup> Это может происходить, например из-за внешнего резервирования ресурсов, хотя в приведенном здесь примере это не рассматривается.

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:

```
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>36</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
```

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
```

---

<sup>36</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.

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 "confnrcse", 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:

$A_{out} = H_{in}$	
$B_{out} = H_{in}$	
$C_{out} = NA$	
$H_{out} = B_{in} + A_{in} + C_{in}$	

J.162REV.1App.IV\_F01

Для ясности описанная выше таблица повторяется ниже в графической форме:

Взяв объединение всех потоков на каждом выходе, получим:

$A_{out} = H_{in}$	
$B_{out} = H_{in}$	
$C_{out} = NA$	
$H_{out} = B_{in} + A_{in} + C_{in}$	

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	replicate
Режим соединения В (слева)	sendonly							
	recvonly							
	sendrecv							
	confrnce							
	inactive							
	newloop/ newtest							
	replicate							

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 IPCom. 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 IPCom. 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 and 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 IPCom 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.

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;
- ADSI.

#### 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 вызываемый терминал (от терминала к сети) (= переход в состояние "телефонная трубка снята")
as2	Сигнал ответа 2	–	TO	Уведомление о том, что вызываемый терминал ответил (от сети к терминалу) Выдержка времени = бесконечная
ds1	Сигнал разъединения 1	P, S	–	Уведомление о том, что связь завершена (от терминала к сети) (= переход в состояние "телефонная трубка положена")
ds2	Сигнал разъединения 2	–	TO	Уведомление о том, что исходящий терминал завершил связь (от сети к терминалу). Выдержка времени = бесконечная
cbs	Сигнал отбоя	P, S	–	Уведомление о том, что вызываемый терминал завершил 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
tir	Предупредительный сигнал переадресации вызова	–	TO	Для услуги "речевое отклонение" Выдержка времени = 2–3 с
car	Сигнал активизации терминала, принимающего данные	–	TO	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

Код	Имя сигнала	Событие	Сигнал	Дополнительная информация
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 продолжительность DTMF-сигнала	✓	–	
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 тактовых (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	–	TO	Готовность к приему сигнала набора номера Выдержка времени = 20–30 с
sdt	Второй тональный сигнал готовности к набору номера	–	TO	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	–	TO	Выдержка времени = 60–70 с
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 вызываемый терминал ответил (as1). В обратном направлении сеть посылает уведомление исходному терминалу о том, что вызываемый терминал ответил (as2).

**Сигнал разъединения (ds):** Исходящий терминал уведомляет сеть о том, что связь завершена (ds1). В обратном направлении сеть посылает уведомление вызываемому терминалу о том, что исходящий терминал завершил связь (ds2).

**Сигнал отбоя (cbs):** Уведомляет сеть о том, что вызываемый терминал завершил связь.

**Сигнал подключения (hs):** Терминал 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 терминал о том, что входящий вызов подвергается переадресации. Кроме того, в случае "услуги NARIWAKE" сеть информирует терминал 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 терминал о том, что к подписавшемуся на услугу пользователю имеется входящий вызов и переадресация активизирована.

**Сигнал активизации терминала, принимающего данные (car):** Сеть извещает терминал, принимающий данные, о том, что имеется входящий вызов с информацией, о которой уведомил модемный сигнал.

**Сигнал первичного ответа (pas):** Вызываемый терминал уведомляет сеть о том, что у телефонного аппарата сняли трубку. Эта функция используется для отображения номера на экране.

**Входящий успешный сигнал (iss):** Сеть уведомляет исходящий терминал о том, что входящий сигнал принят успешно. Эта функция используется для отображения номера на экране.

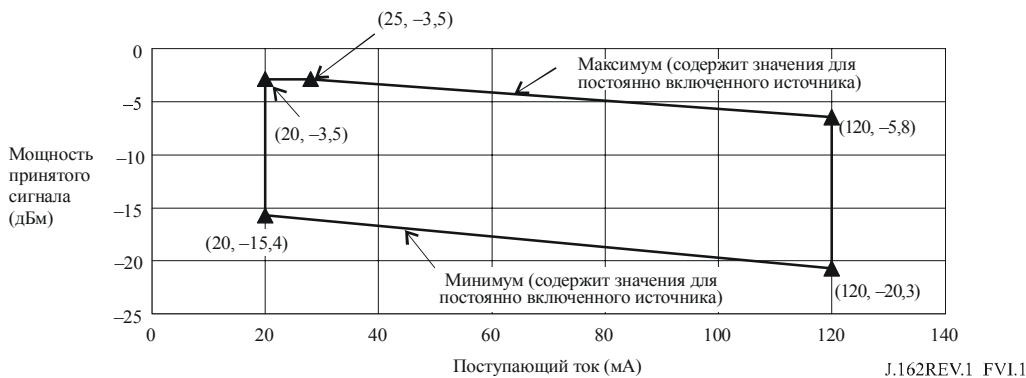
**Сигнал выбора (ss):** Исходный терминал уведомляет сеть о типе услуги и номере другой стороны. Кодом, присваиваемым сигналу выбора (0-9, \*, #), является ssn, а кодом для подстановочного знака тастатурных (PB) тональных сигналов является ssw. В приведенных далее таблицах и на рисунках представлены частоты и уровни сигналов тастатурного набора номера (PB) на приеме.

1) Частота

Группа высоких частот			
Группа низких частот	1209 Гц	1336 Гц	1477 Гц
697 Гц	1	2	3
770 Гц	4	5	6
852 Гц	7	8	9
941 Гц	*	0	#

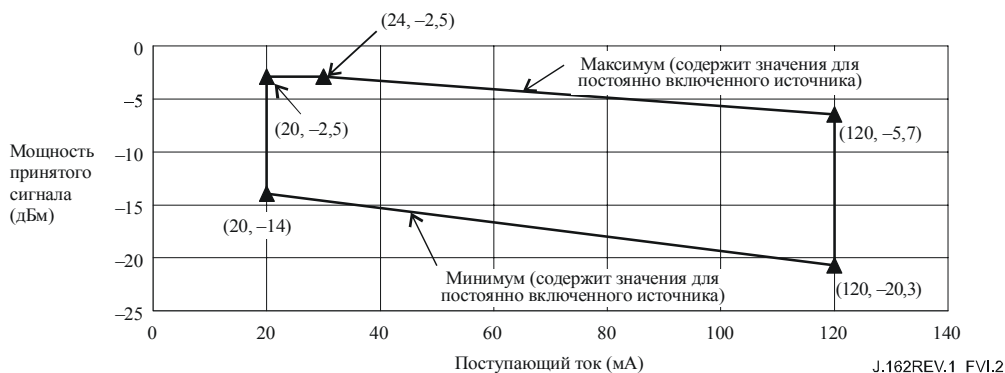
2) Стандартные значения на приеме

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 исходящий терминал о том, что она готова принять сигнал набора номера. В случае внесетевого вызова от компонентной телефонной сети сеть уведомляет исходящий терминал 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 исходящий терминал о том, что она готова принять второй сигнал набора номера. В случае внесетевого вызова от компонентной телефонной сети сеть уведомляет исходящий терминал 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 исходящий терминал, что она вызывает приемный терминал. Передача тонального сигнала заканчивается при получении ответа от вызываемого терминала. 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 исходящий терминал, что приемный терминал находится в состоянии связи; таким образом, она не может предоставить услугу или установить соединение, запрашиваемые исходящим терминалом. 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 исходящий терминал 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 ожидающий терминал 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 соответствующий вызываемый терминал, 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 соответствующий вызываемый терминал, 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 терминал 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 терминал 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 факультативным; 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 Т.30 или Рекомендацию МСЭ-Т V.21.

**Мультимедийный запуск (ma):** Событие "мультимедийный запуск" имеет место в соединении, когда по соединению принимается первый действительный<sup>37</sup> мультимедийный пакет согласно протоколу RTP. Это событие может быть использовано для синхронизации локального сигнала, например сигнала контроля послышки 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 "набранный цифра". Сообщение о завершении может содержать в качестве параметра имя сигнала, время существования которого закончилось, как в случае:

О: L/oc (L/dt)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

О: 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>38</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 сбой. Сообщение о завершении может содержать в качестве параметра имя сигнала, в котором произошел сбой, как в случае:

О: L/of (L/ir)

Когда сигнал, о котором передано сообщение, был использован в соединении, предоставленный параметр будет также содержать имя соединения, как в случае:

О: 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>38</sup>:

О: L/of (B/C (M (sendrecv (AB2354) ) ) ) )

**Таймер (t):** Таймер Т – это обеспечиваемый таймер, который может быть аннулирован только путем ввода DTMF-сигналов. Когда таймер Т используется с действием "накопление согласно отображению цифр", он не запускается до тех пор, пока не поступит первая цифра, и перезапускается

<sup>37</sup> Когда используются услуги аутентификации и обеспечения защиты целостности, пакет RTP не считается действительным, пока он не пройдет проверку на обеспечение защиты.

<sup>38</sup> Следует отметить, что "B" используется здесь в качестве префикса для сообщаемого параметра.

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 –  $T_{\text{par}}$  или  $T_{\text{crit}}$ . Когда для строки цифр требуется по крайней мере еще одна цифра, чтобы соответствовать какой-либо из комбинаций отображения цифр, таймер  $T$  принимает значение  $T_{\text{par}}$ , соответствующее временным соотношениям при неполном наборе номера. Для получения полного соответствия отображению цифр таймер  $T$  принимает значение  $T_{\text{crit}}$ , 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  $T_{\text{crit}}$ , 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  $T_{\text{par}}$  составляет 16 секунд, а для  $T_{\text{crit}}$  – 4 секунды. Оба эти значения могут быть изменены в процессе обеспечения.

**Большая продолжительность DTMF-сигнала (l):** "Большая продолжительность DTMF-сигнала" наблюдается, когда этот сигнал выдается в течение более 2 секунд. В этом случае шлюз обнаружит два последовательных события: первое, когда сигнал был распознан как DTMF-сигнал, 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. См. п. 4.2, функции приема для телефонного номера отправителя (отображение номера на экране дисплея), технический эталон интерфейсов телефонных услуг.

## Видеоинформация

Пакеты событий для видеоинформации будут представлены в будущей версии настоящей Рекомендации.

## Дополнение VII

### Пакеты событий

В данном разделе 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 медийного шлюза. 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 медийного шлюза. В противном случае следующие символы определяют тип события:

"OO": Сигнал "включено/выключено". Сигнал будет включенным, пока он не будет выключен по команде от контроллера медийного шлюза, 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 продолжительность.

**Дополнительная информация:** Предоставляется дополнительная информация о событии/сигнале, например продолжительность 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	MFPB (DTMF)-сигналы	√	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 продолжительность MFPB (DTMF)-сигнала	√	–	
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 MFPB (DTMF)-сигналов	√	–	Соответствует любой из цифр "0–9"



Ниже приводятся определения отдельных событий и сигналов:

**MFRB (DTMF)-сигналы (0–9, \*, #, A, B, C, D):** Обнаружение и генерирование многочастотных тактовых (MFRB (DTMF)) сигналов описано в стандарте ETS 300 001, Глава 5: Функция вызова. Попытка передачи MFRB (DTMF)-сигналов 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 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 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 факультативным, 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 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 CNQ согласно Рекомендации МСЭ-Т T.30. См. Рекомендации МСЭ-Т T.30 и V.21.

**Переход в состояние "телефонная трубка снята" (hd):** См. стандарт EG 201 188, раздел 7: Сигнал занятия.

**Флэш-сигналы (hf):** См. стандарт EG 201 188, раздел 14.2: Выбор данных регистра.

**Переход в состояние "телефонная трубка положена" (hu):** См. стандарт EG 201 188, Раздел 8: Сигнал отбоя. Синхронизация сигнала "телефонная трубка положена" активизируется для ответа на флэш-сигналы.

**Большая продолжительность MFRB (DTMF)-сигнала (L):** "Большая продолжительность MFRB (DTMF)-сигнала" наблюдается, когда этот сигнал выдается в течение более двух секунд. В этом случае шлюз обнаружит два последовательных события: первое, когда сигнал был распознан как MFRB (DTMF)-сигнал, 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>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):** 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 определяется местной администрацией и МОЖЕТ переопределяться через процесс обеспечения. См. стандарты 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:

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 (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>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/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.

<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.

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 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 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 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 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 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 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 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>41</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 MFPB (DTMF)-сигналов. 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 –  $T_{\text{par}}$  или  $T_{\text{crit}}$ . Когда для строки цифр требуется по крайней мере еще одна цифра, чтобы соответствовать какой-либо из комбинаций отображения цифр, таймер T принимает значение  $T_{\text{par}}$ , соответствующее временным соотношениям при неполном наборе номера. Для получения полного соответствия отображению цифр таймер T принимает значение  $T_{\text{crit}}$ , 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  $T_{\text{crit}}$ , 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  $T_{\text{par}}$  равно 16 секундам, а для  $T_{\text{crit}}$  – 4 секундам. Оба этих значения могут быть изменены в процессе обеспечения.

**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 CPE.

**Тональный сигнал 1 ожидания вызова (wt1, .., wt4):** Тональные сигналы ожидания вызова определяются местной администрацией и МОГУТ переопределяться через процесс обеспечения. См. стандарты 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 MFPB (DTMF)-сигналов (X):** Подстановочный знак для MFPB (DTMF)-сигналов соответствует любой цифре многочастотного тастатурного (MFPB (DTMF)) набора номера между 0 и 9.

<sup>41</sup> Этот параметр необходим, например, для быстрого набора номера.

## **Видеоинформация**

Пакеты событий для видеоинформации подлежат дальнейшему изучению.

## **ЦСИС**

Пакеты событий для ЦСИС базового доступа подлежат дальнейшему изучению.

# **Дополнение VIII**

## **Применение протокола NCS к терминалу IPAT сети SCN**

### **VIII.1 Обзор**

В данном Дополнении определяется применение протокола NCS, описанного в основной части настоящей Рекомендации, к терминалу 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 ECCA (EPC-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.

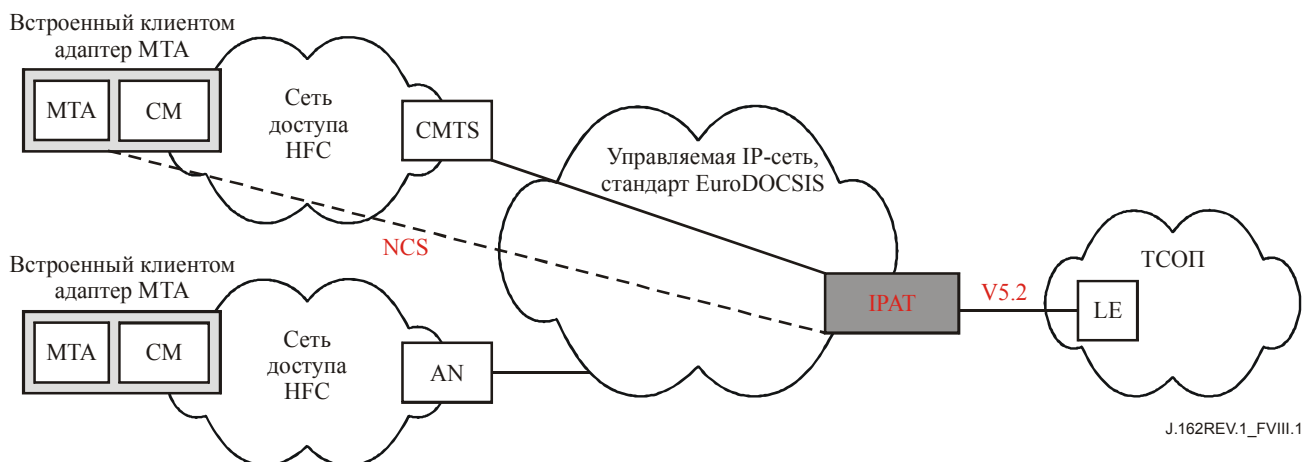
**ПРИМЕЧАНИЕ 3.** – В этой Рекомендации предполагается использование только кодеков G.711, все другие кодеки должны рассматриваться как подлежащие изучению в будущем.

**ПРИМЕЧАНИЕ 4.** – Вопросы, связанные с линиями ЦСИС/BRI, подлежат дальнейшему изучению.

### **VIII.2 Архитектура терминала IPAT**

На рисунке VIII.1 представлена эталонная архитектура, используемая в настоящем Дополнении. Терминал IPAT обеспечивает взаимодействие между сетью проекта IP-Cablecom и местной станцией, являющейся частью сети SCN. В интерфейсе между терминалом IPAT и станцией LE используется подмножество стандарта ETS 300 324, который применим для поддержки услуг SCN для аналоговых телефонов.

В этом отображении, описанном в настоящем Дополнении, не делается каких-либо предположений относительно внутренней структуры терминала 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**

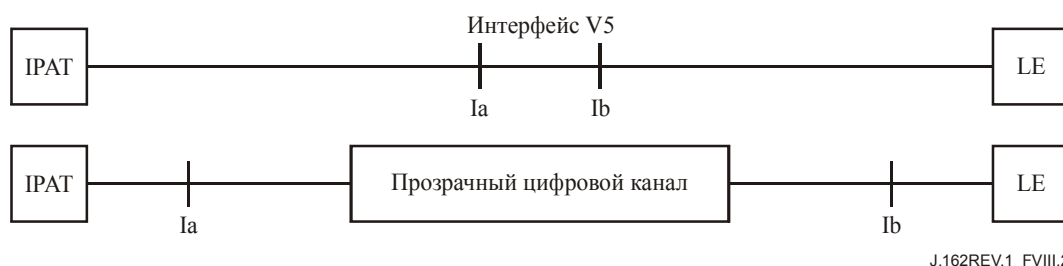
### VIII.3 Требования к электрическим и физическим интерфейсам

Здесь предполагается архитектура системы согласно стандарту ETS 300 324, состоящая из местной станции (LE) и терминала 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.

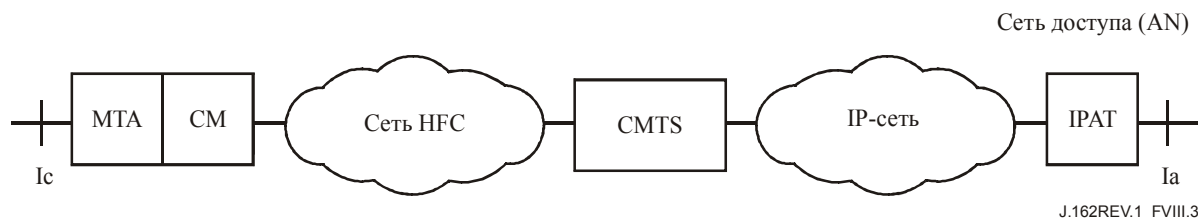
Электрические и физические характеристики этого интерфейса должны соответствовать стандарту ETS 300 166 при скорости передачи 2048 кбит/с.

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 терминала IP-доступа (IPAT), головного узла кабельных модемов (CMTS), кабельного модема (СМ) и адаптера мультимедийного терминала (МТА) или встроенного адаптера мультимедийного терминала (Е-МТА).



**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 удаленный цифровой терминал (RDT) в традиционной архитектуре коммутатора каналов.

Электрические и логические описания IP-сети и сети НФС 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 терминал IPAT проекта IPCablecom преобразовывал сообщение протокола для интерфейса V5, полученное от коммутатора для интерфейса V5, в соответствующий запрос сигнала от терминала 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 терминалом 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 МТА 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 продолжительность импульса (длина одного импульса);
- **pr** обозначает интервал повторения импульса для всех импульсов.

Значения **pd** и **pr** являются факультативными. 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);
- **rpc** обозначает число импульсов между сообщениями о тарифных импульсах (факультативный параметр, только для сигнала em).

Большинство запросов импульсных сигналов являются по существу сигналами выдержки времени (ТО), значение выдержки времени для которых может определяться следующим образом:

$$t_o = pr \times rep$$

Для терминала 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 терминале IPAT.



Если произведение  $pr \times \text{гер}$  существенно меньше 180 секунд, то терминалу IPAT СЛЕДУЕТ включить значение выдержки времени; если произведение  $pr \times \text{гер}$  больше 180 секунд, то терминал 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" (**rpc**). 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 факультативными (O), обязательными (M) или запрещенными (F).

Таблица VIII.2/J.162 – Кодирование линейной обработки

Код It	Описание	Тип сигнала	pd	pr	гер (примечание)	rpc
<b>ir</b>	Начальный вызывной сигнал	TO	O	O	O	F
<b>lc</b>	Импульсный цикл замкнут	TO	O	O	O	F
<b>lo</b>	Импульсный цикл разомкнут	TO	O	O	O	F
<b>em</b>	(Активизировать) генерирование тарифных импульсов	OO	F	O	F	O
<b>mpb</b>	Генерирование пачки тарифных импульсов	BR	O	O	O	F
<b>nb</b>	Нет батарейного импульса	TO	O	O	O	F
<b>np</b>	Импульс нормальной полярности	TO	O	O	O	F
<b>rb</b>	Подавленный батарейный импульс	TO	O	O	O	F
<b>rp</b>	Импульс обратной полярности	TO	O	O	O	F

ПРИМЕЧАНИЕ. – Параметр "гер" является ОБЯЗАТЕЛЬНЫМ, если значение предоставлено интерфейсом V5 станции LE. Значение ФАКУЛЬТАТИВНЫЙ 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)
<b>ir</b>	Начальный вызывной сигнал	25 Гц ( $\pm 1$ Гц)	Полная	200 (0–5000, 50)	200 (0–5000, 50)	1 (1–5, 1)
<b>lc</b>	Импульсный цикл замкнут	Нулевая	Нулевая	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
<b>lo</b>	Импульсный цикл разомкнут	Нулевая	Нулевая	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
<b>em</b>	(Активизировать) генерирование тарифных импульсов	16 кГц	$-13,5 \text{ дБм}^{\text{a}}$ (от $-25$ до $+15,2$ дБ)	150 (0–5000, 10)	1000 (0–5000, 10)	Нулевое

**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>mpb</b>	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)
<b>nb</b>	Нет батарейного импульса	Нулевая	0	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
<b>np</b>	Импульс нормальной полярности	Нулевая	1	200 (0–5000, 10)	1000 (0–5000, 10)	1 (1–50, 1)
<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)

<sup>a)</sup> Амплитуда тарифного импульса определяется в дБм через нагруженные выходы 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 список параметров запрашиваемых событий (R) 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**

Терминал 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**

Продолжительность импульса определяется в миллисекундах с использованием параметра **pd**. Например, 200-миллисекундный импульс определяется как:

`pd=200`

Продолжительность импульса является *факультативной*. Если продолжительность 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 *факультативным*. 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 терминала IPAT, когда каждый запрошенный импульс заканчивается. Об этом событии для каждого законченного импульса уведомляют в течение запроса сигнала без дополнительных запросов на уведомление от терминала 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" (**грс**). 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". Терминалу IPAT не нужно посылать новый запрос на уведомление.

Кодом события для сообщения о тарифных импульсах является **mpg**. Это уведомление включает счетчик. Пример:

О: 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, терминал 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 терминал 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 терминал IPAT должен быть заранее обеспечен соответствующим сообщением СИГНАЛ интерфейса V5 (например, состояние "телефонная трубка положена" на дальнем конце).

По получении кода "01" V5 терминал 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 терминал 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 терминал IPAT должен быть заранее обеспечен соответствующим сигнальным сообщением линейной обработки сигнализации (например, "телефонная трубка положена").

По получении кода "10" V5 терминал 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 терминал 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 терминал 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 терминал 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 терминал 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 терминалом IPAT. Терминал 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**

Терминал 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 терминал IPAT принимает от станции LE интерфейса V5 запрос со значением "*rep* = 0" или с пропущенным значением *rep*, терминал 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.

Терминал 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. Терминал 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. Терминал 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 терминал 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 терминал 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 MTA сообщает терминалу 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 MTA сообщает терминалу 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 терминала IPAT нового запроса "активизировать генерирование тарифных импульсов". Терминал IPAT должен учитывать все нормальные тарифные импульсы, пропущенные при генерировании пачки импульсов, путем включения пропущенных импульсов в счетчик пачек импульсов.

Терминал IPAT может факультативно 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 Терминал IPAT**

Терминал 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 терминалу IPAT запрос импульсного вызывного тактового сигнала.
- 2) Терминал 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 терминал 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 терминалу IPAT запрос типа импульсного сигнала "начальный вызывной сигнал" вместе с типом продолжительности импульса.
- 2) Терминал 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 продолжительность 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".
- 6) 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.
- 7) Терминал IPAT сигнализирует станции LE интерфейса V5 о том, что импульс завершен.
- 8) Станция LE интерфейса V5 включает в сообщение к терминалу IPAT запрос импульсного вызывного тактового сигнала.
- 9) Терминал IPAT преобразует двоичный закодированный вызывной тактовый сигнал в десятичное значение в диапазоне от 0 до 127.
- 10) 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)
```

- 11) Адаптер МТА подтверждает запрос сигнала.  
200 520 ОК
- 12) 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 терминал 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 терминалу IPAT запрос типа импульсного сигнала "начальный вызывной сигнал" вместе с типом продолжительности импульсов.

2) Терминал 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 продолжительность начального вызывного сигнала (*pd=200* приводит к пачке вызывных импульсов в 200 мс) и посылает эту информацию на выводы *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) Терминал IPAT сигнализирует станции LE интерфейса V5, что импульс завершен.

7) Станция LE интерфейса V5 затем генерирует для окончания линии *aaIn/1* внутрисполосные тональные сигналы ФМн.

8) Адаптер МТА передает внутрисполосные 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 терминалу IPAT.

10) Терминал 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 выводы *a-b* имеющейся в адаптере МТА линии *aaIn/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 сигнализации NCS, либо пока терминал 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 сообщение к терминалу IPAT запрос импульсного сигнала размыкания цикла.
- 2) Терминал IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и продолжительность 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=lo, 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.

### VIII.7.2.2 Импульсный сигнал с подтверждением начала

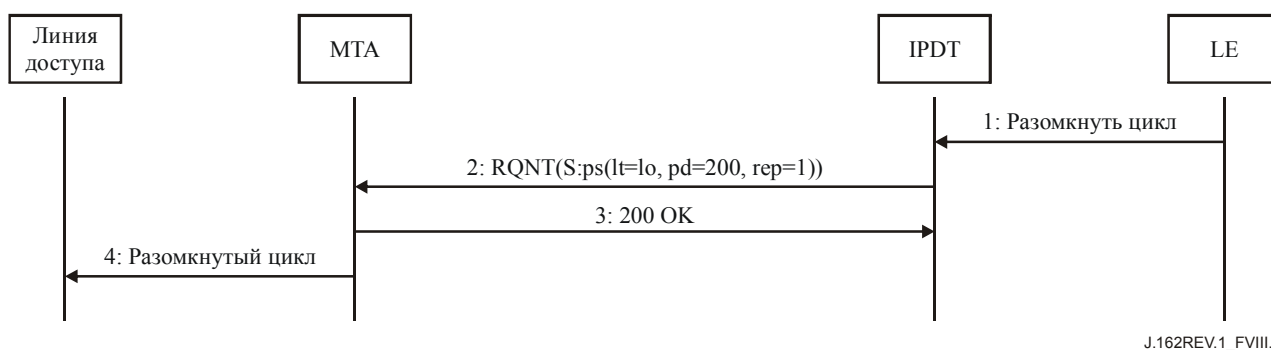


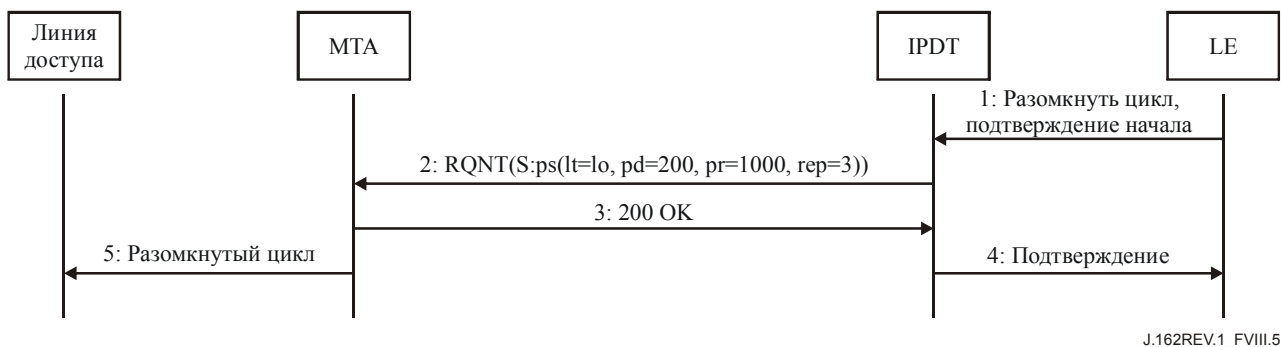
Рисунок VIII.4/J.162 – Запрос импульсного сигнала

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) Терминал IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку, продолжительность импульса и импульсный период из параметров, предоставляемых коммутатором, и генерирует соответствующий запрос сигнала NCS, включая число повторений импульса, предоставляемое станцией LE интерфейса V5. Терминал 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=lo, pd=200, pr=1000, rep=3)
```

- 3) Адаптер МТА подтверждает запрос сигнала.  
200 525 ОК
- 4) Терминал 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) Терминал IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и продолжительность 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 терминал 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 МТА подтверждает запрос сигнала.
 

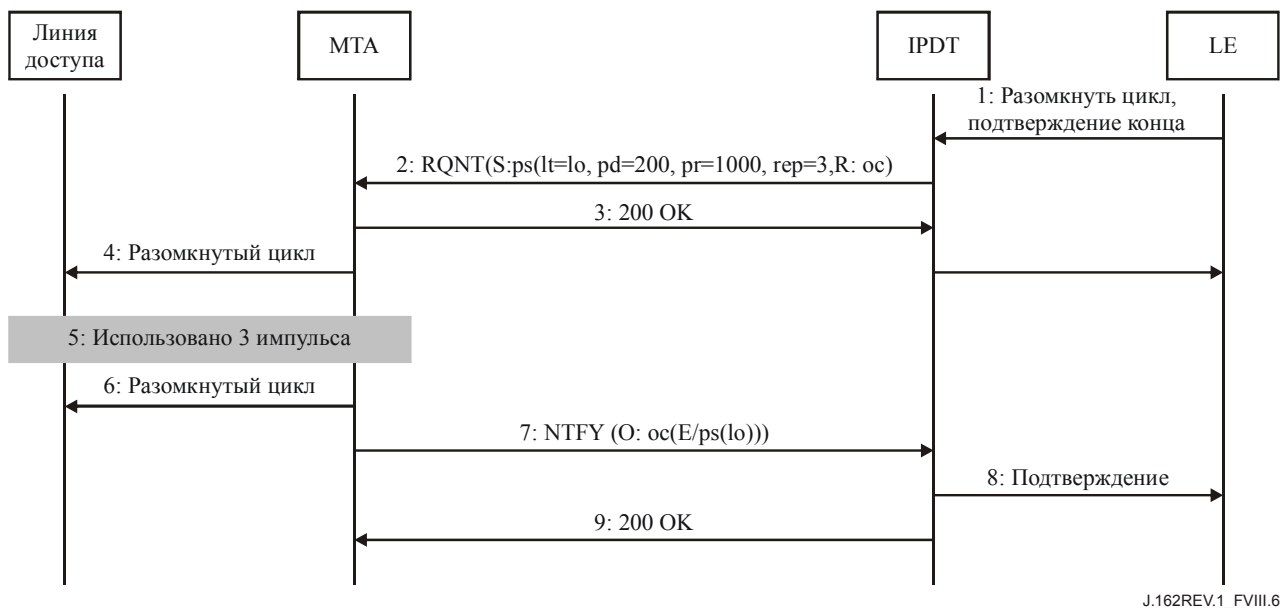
```
200 525 OK
```
- 4) Адаптер МТА начинает использовать запрошенные импульсы в линии.
- 5) 2-й импульс
- 6) 3-й импульс
- 7) По завершении последнего импульса адаптер МТА уведомляет терминал IPAT о том, что операция завершена.

```

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) Терминал IPAT посылает станции LE интерфейса V5 запрошенное подтверждение.
- 9) Терминал IPAT подтверждает уведомление о событии для адаптера МТА.



**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) Терминал 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, терминал IPAT включает встроенный запрос сигнала для сигнала ps.

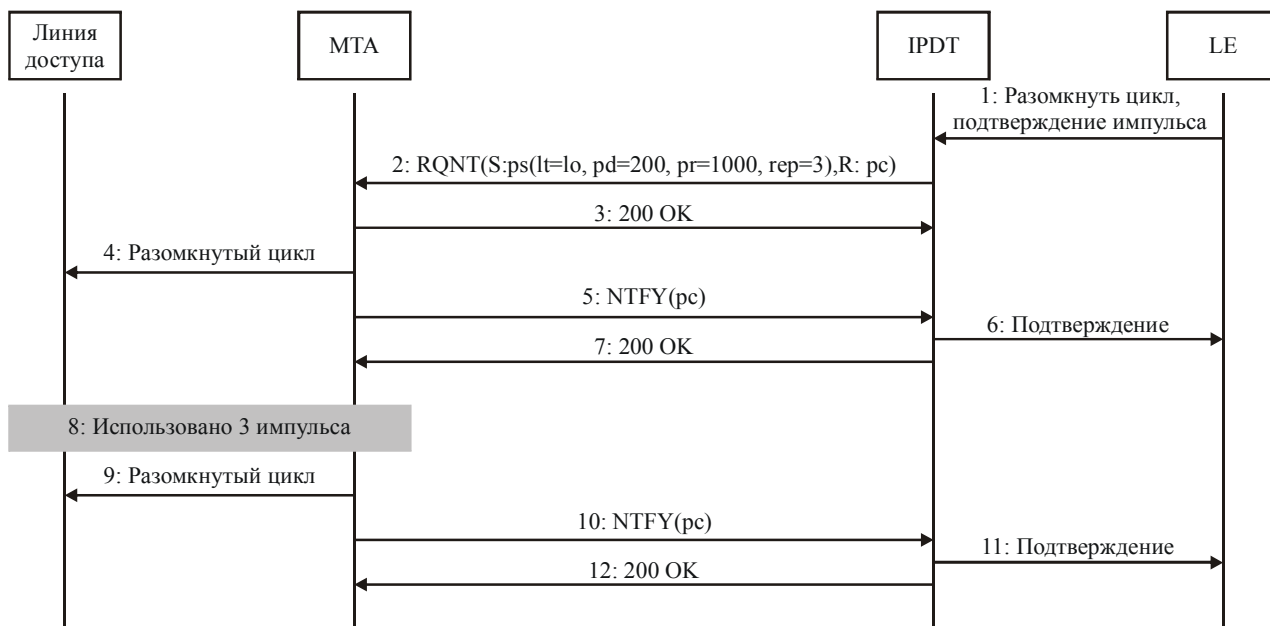
```
RQNT 525 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 1111
S: E/ps(lt=lo, pd=200, pr=1000, rep=3)
R: E/ps
```

- 3) Адаптер MTA подтверждает запрос сигнала.
 

```
200 525 OK
```
- 4) Адаптер MTA подает первый импульс в абонентскую линию доступа.
- 5) По завершении импульса адаптер MTA посылает терминалу IPAT уведомление о событии.

```
NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 1111
O: E/ps(lt)
```

- 6) Терминал IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 7) Терминал IPAT подтверждает уведомление о событии. Терминал 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 MTA 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) Терминал IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и генерирует соответствующий запрос сигнала NCS. Поскольку станция LE интерфейса V5 также запросила подтверждение импульса, терминал IPAT включает параметр pc вместе с запросом сигнала.

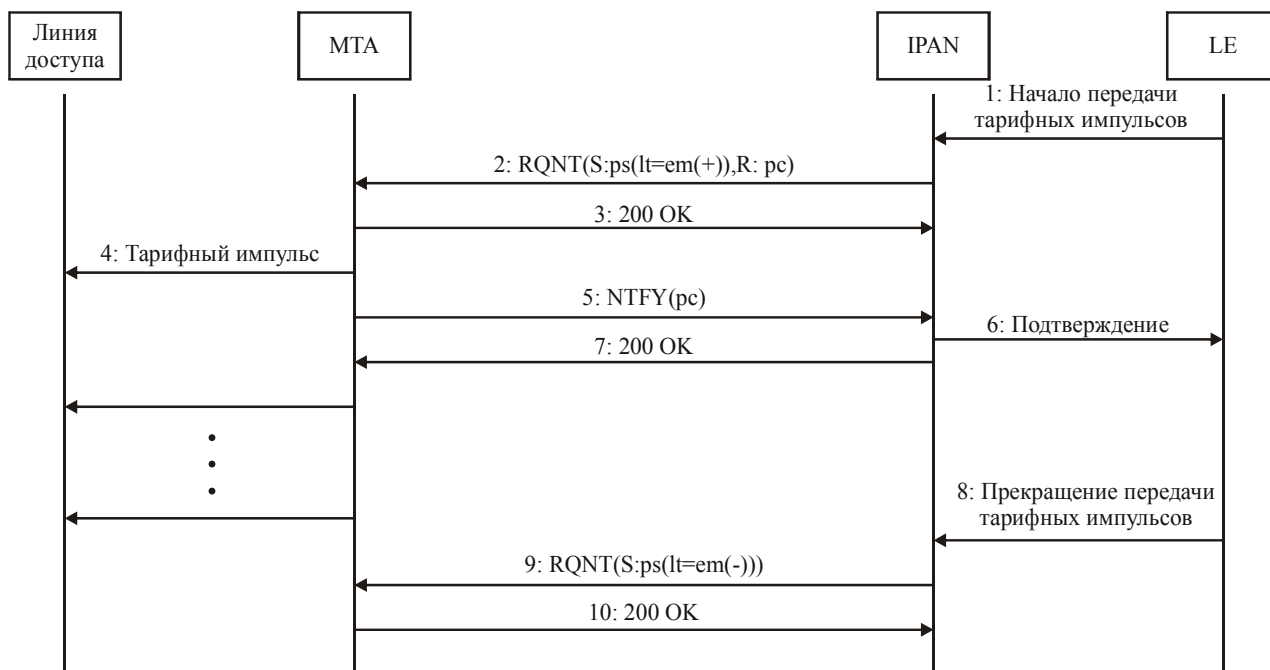
```
RQNT 535 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 2345
S: E/ps(lt=em(+))
R: E/pc
```

- 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 терминалу IPAT уведомление о событии:
 

```
NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 535
O: pc(em)
```
- 6) Терминал IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 7) Терминал 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) Терминал IPAT преобразует закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и продолжительность 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, терминал 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) Терминал 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 терминалу IPAT уведомление о событии.

```

NTFY 3981 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3579
O: pc(em)
  
```

- 8) Терминал IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 9) Терминал IPAT подтверждает уведомление о событии. Терминал 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) Терминал IPAT преобразует новое закодированное двоичным кодом сообщение интерфейса V5 и определяет линейную обработку и продолжительность 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, терминал 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 OK
```
- 14) Адаптер МТА обращается к своей таблице обеспечения для определения частоты тарифных импульсов, амплитуды и значений временных параметров по умолчанию (минимальные допустимые значения).
- 15) Терминал 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 терминалу IPAT уведомление о событии.

```
NTFY 791 aaln/1@rgw.mso.net MGCP 1.0 NCS 1.0
X: 3581
O: ps(em)
```

- 18) Терминал IPAT посылает станции LE интерфейса V5 подтверждение импульса.
- 19) Терминал 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) Терминал 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
```

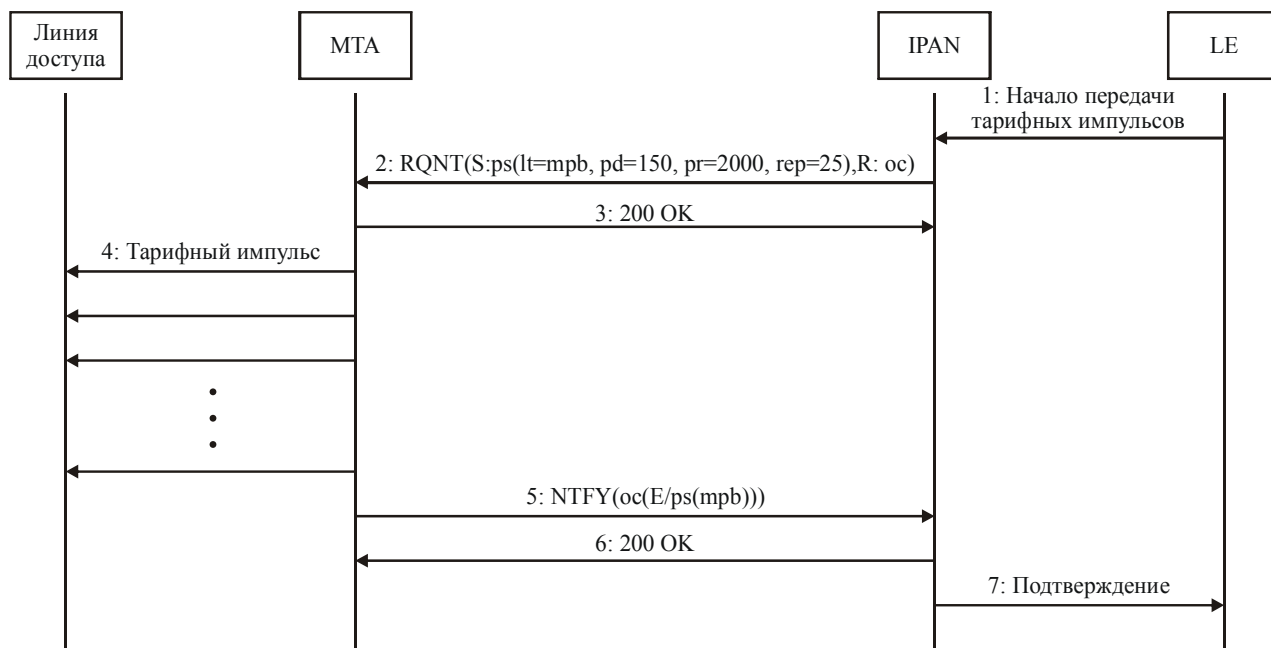
- 3) Адаптер МТА подтверждает запрос.
- 4) Адаптер МТА начинает подавать тарифные импульсы в абонентскую линию доступа.



- 5) В этом примере станция LE запросила уведомление о завершении операции в начальном запросе на генерирование фиксированного числа тарифных импульсов. Теперь адаптер МТА уведомляет терминал 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) Терминал IPAT подтверждает уведомление о событии.  
7) Терминал 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 выводы a-b линии POTS.

- 1) Станция LE интерфейса V5 включает в сообщение к терминалу IPAT запрос постоянного сигнала обратной полярности.
- 2) Терминал 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 выводы 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  $E/ps(mrb)$  – 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 медийного шлюза от информации о денежных суммах. Тарифные единицы меняются в зависимости от рынка. Медийный шлюз не должен обладать информацией о стоимости импульса (в денежных единицах).

#### IX.2 Пакет автоматической тарификации

Пакет автоматической тарификации разработан так, чтобы удовлетворять требованиям медийных шлюзов с аналоговыми линиями, сконфигурированными для телефонии общего назначения, с добавлением возможности автоматической передачи тарифных импульсов.

Импульсные характеристики (тип импульса, продолжительность 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 медийном шлюзе.

Для данного пакета предполагается, что накопление является задачей устройства CPE. В данном пакете не требуется, чтобы медийный шлюз 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	Продолжительность
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.			
Продолжительность	Определяет продолжительность сигналов TO. Если продолжительность 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/mprb 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 Случай 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 продолжительность.

**IX.4.6 субтариф:** Тарифная единица в единицу времени в рамках тарифной последовательности. Каждый субтариф имеет свою продолжительность 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 продолжительность 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 продолжительность зависит от национальных спецификаций. См. EN 300 001 V1.5.1 (1998-10), раздел 1.7.8.

### **Библиография**

- ECCA EuroPacketCable working group requirements document EPC-ReqDoc-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 В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
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	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
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 и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевых протоколов и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи