



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

J.171.1

(11/2005)

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

Проект IP-Cablecom

**Протокол управления шлюзом транкинговой
связи IP-Cablecom (TGCP): Профиль 1**

Рекомендация МСЭ-Т J.171.1

Рекомендация МСЭ-Т J.171.1

Протокол управления шлюзом транкинговой связи IPСablecom (TGСР): Профиль 1

Резюме

В настоящей Рекомендации описывается профиль IPСablecom интерфейса прикладного программирования (API), называемый интерфейсом управления медиашлюзом (MGCI), и соответствующий протокол (MGСР – протокол управления медиашлюзом) для управления шлюзами КТСОП передачи голоса по протоколу IP (VoIP) из внешних элементов управления вызовами. Это один из двух профилей, указанных в Рек. МСЭ-Т J.171.0. Второй профиль определен в Рек. МСЭ-Т J.171.2.

Протокол MGСР предполагает наличие архитектуры управления вызовом, в которой "интеллект" управления вызовами находится за пределами шлюзов, и им оперируют внешние элементы управления вызова. Профиль IPСablecom, описанный в этой Рекомендации, будет именоваться как протокол управления шлюзом транкинговой связи IPСablecom (TGСР).

Источник

Рекомендация МСЭ-Т J.171.0 утверждена 29 ноября 2005 года 9-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

ПРЕДИСЛОВИЕ

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

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

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

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

ПРИМЕЧАНИЕ

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

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

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

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

© ITU 2007

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

СОДЕРЖАНИЕ

	Стр.
1 Сфера применения	1
2 Справочные документы	1
2.1 Нормативные справочные документы	1
2.2 Информативные справочные документы	1
3 Определения	2
4 Сокращения и акронимы	2
5 Соглашения	2
5.1 Исходные данные	3
6 Введение	4
6.1 Взаимосвязь с другими стандартами IPCablecom	5
6.2 Взаимосвязь с RFC 3435 и грамматикой ABNF	6
7 Интерфейс управления медиашлюзом (MGCI)	6
7.1 Модель и соглашения об именовании	6
7.2 Использование протокола SDP	12
7.3 Функции управления шлюзом	13
7.4 Состояния, восстановление после отказа и состязательные условия	36
7.5 Коды завершения и коды ошибок	51
7.6 Коды причины	52
7.7 Использование параметров LocalConnectionOptions и ConnectionDescriptors ..	53
8 Протокол управления медиашлюза	56
8.1 Общее описание	56
8.2 Заголовок команды	57
8.3 Форматы заголовков ответа	69
8.4 Кодирование описания сеанса	72
8.5 Передача по протоколу UDP	83
8.6 Совмещение передачи запросов и ответов	85
8.7 Идентификаторы транзакций и установление трехсторонней связи	85
8.8 Временные ответы	86
9 Безопасность	87
Приложение А – Пакеты событий	88
А.1 Пакет линии связи ISUP	88
Дополнение I – Взаимодействия режимов	92
Дополнение II – Примеры кодирования команд	94
II.1 Команда NotificationRequest	94
II.2 Команда Notify	94
II.3 Команда CreateConnection	94

	Стр.
II.4 Команда ModifyConnection	96
II.5 Команда DeleteConnection (от контроллера медиашлюза)	97
II.6 Команда DeleteConnection (от транкингового шлюза).....	97
II.7 Команда DeleteConnection (от контроллера медиашлюза для нескольких соединений).....	97
II.8 Команда AuditEndpoint.....	98
II.9 Команда AuditConnection.....	99
II.10 Команда RestartInProgress	99
Дополнение III – Пример последовательности операций для вызова.....	101
Дополнение IV – Требования конечной точки	104
IV.1 Поддерживаемые режимы соединения.....	104
Дополнение V – Информация о совместимости.....	105
V.1 Совместимость протокола NCS.....	105
V.2 Совместимость протокола MGCP	105
Дополнение VI – Грамматика ABNF для TGCP	107
Дополнение VII – Электронное наблюдение	114
VII.1 MGC	114
VII.2 MG.....	114
Дополнение VIII – Примеры пакетов событий.....	116
VIII.1 Пакет услуг оператора FGD MF	116
VIII.2 Завершающий пакет протокола MF	119
БИБЛИОГРАФИЯ	122

Рекомендация МСЭ-Т J.171.1

Протокол управления шлюзом транкинговой связи IP-Cablecom (TGCP): Профиль 1

1 Сфера применения

В настоящей Рекомендации описывается профиль IP-Cablecom интерфейса прикладного программирования (API), называемый интерфейс управления медиашлюзом (MGCI), и соответствующий протокол (MGCP – протокол управления медиашлюзом) для управления шлюзами КТСОП передачи голоса по протоколу IP (VoIP) из внешних элементов управления вызовами. Это один из двух профилей, указанных в Рек. МСЭ-Т J.171.0. Второй профиль определен в Рек. МСЭ-Т J.171.2.

Протокол MGCP предполагает наличие архитектуры управления вызовом, в которой "интеллект" управления вызовами находится за пределами шлюзов, и им оперируют внешние элементы управления вызовом. Профиль IP-Cablecom, описанный в этой Рекомендации, будет именоваться как протокол управления шлюзом транкинговой связи IP-Cablecom (TGCP).

Данная Рекомендация основана на Рекомендации, касающейся сигнализации сетевого вызова IP-Cablecom (Рек. МСЭ-Т J.162), и стандарте IETF RFC 2705, *Протокол управления медиашлюзом (MGCP)*. В данной Рекомендации, определяющей протокол TGCP IP-Cablecom, устанавливаются технические требования, которые не зависят от MGCP. Профиль TGCP протокола MGCP определяется точно и исключительно положениями этой Рекомендации.

2 Справочные документы

2.1 Нормативные справочные документы

Нижеследующие Рекомендации МСЭ-Т и другие источники содержат положения, которые путем ссылок на них в данном тексте, составляют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники являются предметом пересмотра; поэтому всем пользователям данной Рекомендации предлагается рассмотреть возможность применения последнего издания Рекомендаций и других ссылок, перечисленных ниже. Перечень действующих на настоящий момент Рекомендаций МСЭ-Т публикуется регулярно. Ссылка на документ, приведенный в настоящей Рекомендации, не придает ему, как отдельному документу, статус Рекомендации.

- ITU-T Recommendation J.161 (Draft, version 2), *Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.*
- ITU-T Recommendation J.162 (2005), *Network call signalling protocol for the delivery of time-critical services over cable television networks using cable modems.*
- ITU-T Recommendation J.170 (2005), *IP-Cablecom security specification.*
- IETF RFC 2327 (1998), *SDP: Session Description Protocol.*

2.2 Информативные справочные документы

- ITU-T Recommendation E.180/Q.35 (1998), *Technical characteristics of tones for the telephone service.*
- ITU-T Recommendation J.163 (2005), *Dynamic quality of service for the provision of real-time services over cable television networks using cable modems.*
- Рекомендация МСЭ-Т J.171.0 (2005 г.), *Протокол управления транкинговым шлюзом IP-Cablecom (TGCP): Обзор профилей.*
- IETF RFC 1889 (1996) *RTP: A Transport Protocol for Real-Time Applications.*

- IETF RFC 1890 (1996), *RTP Profile for Audio and Video Conferences with Minimal Control*.
- IETF RFC 2543 (1999), *SIP: Session Initiation Protocol*.
- IETF RFC 2705 (1999), *Media Gateway Control Protocol (MGCP) Version 1.0*.
- TCP/IP Illustrated, Volume 1 (2001), *The Protocols*, Addison-Wesley, 1994.

3 Определения

В данной Рекомендации определены следующие термины:

3.1 кабельный модем: Поставка высокоскоростного доступа данных к местоположению клиента, с использованием оборудования, созданного в соответствии с Рек. МСЭ-Т J.83 и J.112.

3.2 IPcablecom: Проект МСЭ-Т, в который включены архитектура и серия Рекомендаций, делающие возможной поставку услуг в режиме реального времени по сети кабельного телевидения с использованием кабельных модемов.

4 Сокращения и акронимы

В данной Рекомендации используются следующие сокращения и акронимы:

DNS	Domain Name System	Система доменных имен
IP	Internet Protocol	Протокол Интернет
IPSec	Internet Protocol Security	Безопасность протоколов Интернет
ISUP	ISDN User Part	Подписка пользователя ISDN
MGC	Media Gateway Controller	Контроллер медиашлюза
MGCP	Media Gateway Control Protocol	Протокол управления медиашлюзом
MIB	Management Information Base	Информационная база управления
MTA	Media Terminal Adapter	Оконечный адаптер среды передачи
MWD	Maximum Waiting Delay	Максимальная задержка на ожидание
NCS	Network-based Call Signalling	Сигнализация сетевого вызова
NTP	Network Time Protocol	Сетевой протокол службы времени
QoS	Quality of Service	Качество обслуживания
RTCP	Real-Time Control Protocol	Протокол управления в режиме реального времени
RTO	Retransmission Timeout	Интервал для повторной передачи
RTP	Real-Time Protocol	Протокол режима реального времени
SDP	Session Description Protocol	Протокол описания сессии
SG	Signalling Gateway	Шлюз сигнализации
SPI	Security Parameters Index	Индексы параметров безопасности

5 Соглашения

При выполнении данной Рекомендации ключевые слова "ДОЛЖЕН" и "СЛЕДУЕТ", так же как и "НЕОБХОДИМО", должны интерпретироваться как указывающие на обязательный аспект данной Рекомендации.

Перечень ключевых слов, используемых в данной Рекомендации для указания на определенный уровень значимости отдельного требования, приведен ниже:

"ДОЛЖЕН"	Данное слово, наречие "НЕОБХОДИМО" или глагол "ТРЕБУЕТСЯ" означает, что данное условие является абсолютным требованием этой Рекомендации.
"НЕ ДОЛЖЕН"	Данное словосочетание означает, что на данное условие этой Рекомендацией налагается абсолютный запрет.
"СЛЕДУЕТ"	Данное слово или глагол "РЕКОМЕНДУЕТСЯ" означает, что могут существовать веские условия при определенных обстоятельствах, в которых данное условие можно игнорировать, но перед тем, как выбрать другой вариант, необходимо получить полное понимание последствий и тщательно взвесить ситуацию.
"НЕ СЛЕДУЕТ"	Данное словосочетание означает, что могут существовать веские условия при определенных обстоятельствах, в которых описанный образ действий приемлем или даже полезен, но перед тем, как выполнить действия, отмеченные этим обозначением, необходимо получить полное понимание последствий и тщательно взвесить ситуацию.
"МОЖЕТ"	Данное слово или наречия "МОЖНО", "НЕОБЯЗАТЕЛЬНО" означает, что данное условие является необязательным. Один поставщик вправе использовать его, потому что этого будет требовать рыночная ситуация или, например, это приведет к улучшению продукта, а другой поставщик может опустить это условие.

5.1 Исходные данные

В настоящей Рекомендации описывается профиль IP-Cablecom интерфейса прикладного программирования, называемый интерфейс управления медиашлюзом (MGCI), и соответствующий протокол (MGCP) для управления шлюзами КТСОП передачи голоса по протоколу IP (VoIP) из внешних элементов управления вызовами. Протокол MGCP предполагает наличие архитектуры управления вызовом, в которой "интеллект" управления вызовами находится за пределами шлюзов, и им оперируют внешние элементы контроля вызова. Профиль IP-Cablecom, описанный в этой Рекомендации, будет именоваться как протокол управления шлюзом транкинговой связи IP-Cablecom (TGCP): Профиль 1.

Данная Рекомендация основана на Рек. МСЭ-Т J.162 Сигнализация сетевого вызова IP-Cablecom (NCS), IETF RFC 3435 Протокол управления медиашлюзом (MGCP) 1.0. (который был получен в результате объединения семейства протоколов: проекта IETF Протокол управления простым шлюзом и проекта семейства протоколов IETF Управление устройством IP (IPDC)), и вклада, созданного целевой группой шлюз КТСОП – IP-Cablecom.

В данной Рекомендации определяется протокол TGCP. Рекомендация является документом, не зависящим от протокола MGCP, для того чтобы предоставить стабильный справочный документ, соответствующий текущим потребностям для таких документов во время вывода новых изделий на рынок.

Цель данной Рекомендации заключается в том, чтобы как можно ближе согласовать NCS и MGCP 1.0 в сетевом окружении IP-Cablecom, для того чтобы избежать разработки множества протоколов для решения одной и той же проблемы. Эту цель выполняли и продолжают выполнять на основе сотрудничества с авторами Рекомендаций NCS и MGCP. Тем не менее, протокол TGCP профиля MGCP точно и исключительно определен положениями данной Рекомендации.

Данный протокол TGCP профиля MGCP, который в дальнейшем будет именоваться как протокол сигнализации вызова шлюза транкинговой связи КТСОП 1.0, TGCP 1.0, профиль TGCP или в этой Рекомендации просто TGCP, был модифицирован из IETF RFC 2435 MGCP 1.0 следующим образом:

- *Протокол TGCP направлен только на поддержку шлюзов КТСОП передачи голоса по протоколу IP.* Протокол TGCP поддерживает шлюзы КТСОП передачи голоса по протоколу IP, как описано в IP-Cablecom. Функционально присутствующий в MGCP 1.0 протокол являлся ненужным для TGCP и был удален.

- *Протокол TGCP содержит расширения и модификации для MGCP.* Требования, характерные для проекта IP-Cablecom, учитываются в TGCP. Тем не менее, архитектура MGCP и все структурные компоненты MGCP, важные для шлюзов КТСОП, были сохранены в TGCP.
- *Протокол TGCP содержит незначительные упрощения из MGCP 1.0.* Когда имелось несколько вариантов и не было необходимости в шлюзе КТСОП в сетевом окружении IP-Cablecom, были сделаны некоторые упрощения для введения в эксплуатацию шлюза транкинговой связи.

Несмотря на то, что MGCP не является TGCP и TGCP это не MGCP, обозначения MGCP и TGCP будут использоваться взаимозаменяемо в данной Рекомендации, т. к. данная Рекомендация основана на MGCP. Если иное не установлено и не зафиксировано контекстом, то в данной Рекомендации слово MGCP будет использоваться для обозначения профиля TGCP протокола MGCP.

TGCP разработан для того, что отвечать требованиям протокола для контроллера медиашлюза и интерфейса медиашлюза, определяемых в архитектуре IP-Cablecom.

6 Введение

В настоящей Рекомендации описывается профиль TGCP интерфейса прикладного программирования (MGCI) и соответствующий протокол (MGCP) для управления шлюзами транкинговой связи из внешних элементов управления вызовами. Шлюз транкинговой связи это сетевой элемент, который предоставляет аналоговое устройство, имитирует аналоговое устройство или цифровой широкополосный канал связи и доступ сигнализации транкинговой линии по связанному каналу к сети передачи голоса по протоколу IP.

Шлюзы транкинговой связи используются для интерфейса КТСОП и предполагается, что как таковые, придерживаются важных электрических, эксплуатационных стандартов, стандартов сигнализации для типа транкинговой связи, выполнение которых ими обеспечивается.

Рекомендация TGCP разработана для конкурентоспособного оператора местной сети (CLEC), и в первой версии данной Рекомендации типы транкинговых связей, работа которых будет обеспечиваться, ограничены следующим:

- ISUP (подсистема пользователя ISDN) SS7:
 - стандартный широкополосный транкинговый канал связи.
- Транкинговые линии СЧ на комбинации конечная телефонная станция равного доступа/последовательный доступ (ЕАЕО/АТ):
 - транкинговые связи услуг оператора¹;
 - доступ абонента CLEC через последовательный доступ LEC;
 - доступ оператора к ЕАЕО/АТ для проверки занятости линии и вторжения;
 - линии экстренного обслуживания для доступа к последовательному доступу экстренного обслуживания.

Протокол MGCP предполагает наличие структуры управления вызовом, в которой "интеллект" управления вызовами находится за пределами шлюзов, и им оперируют внешние элементы контроля вызова, которые именуются контроллерами медиашлюза. Протокол MGCP предполагает, что эти элементы контроля вызова, или контроллеры медиашлюза (MGC), будут синхронизированы друг с другом для передачи согласованных команд шлюзам под их контролем. В протоколе MGCP, описанном в настоящей Рекомендации, не определяется механизм синхронизации для контроллеров медиашлюза, хотя в будущих Рекомендациях IP-Cablecom, возможно, будут определены такие механизмы.

В протоколе MGCP предполагается модель соединения, в которой основными структурными компонентами являются конечные точки и соединения. Шлюз содержит совокупность конечных точек, являющихся источниками или приемниками данных, и которые могут быть физическими или виртуальными.

Примером физической конечной точки является транкинговая линия связи в транкинговом шлюзе, которая завершает широкополосный транкинговый канал связи ISUP в коммутаторе. Другой пример – это встроенный клиент или резидентский шлюз, который завершает резидентские линии POTS (обычная аналоговая телефонная связь) (в телефоны), хотя такие устройства не исследуются в этой Рекомендации.

¹ Предполагается, что услуги оператора обеспечиваются или являются доступными через местную телефонную компанию (LEC).

Примером виртуальной конечной точки является аудиисточник в сервере аудиоконтента. Создание физических конечных точек требует установки аппаратного обеспечения, в то время как создание виртуальных конечных точек может выполняться с помощью программного обеспечения. Тем не менее, в профиле TGCP протокола MGCP рассматриваются только физические конечные точки.

Соединения являются соединениями пункта с пунктом. Соединение пункта с пунктом – это связь между двумя конечными точками с целью передачи данных между этими конечными точками. После установления такой связи для обоих конечных точек между данными точками может иметь место передача данных. Такая связь устанавливается путем создания соединения в виде двух половин: одной половины в исходящей конечной точке, а другой – во входящей конечной точке.

Контроллеры медиашлюза посылают шлюзам команды на создание соединений между конечными точками и обнаружение определенных событий, например, проверка целостности, и генерирование определенных сигналов, например, посылка вызова. Только контроллер медиашлюза абсолютно точно определяет, где и когда устанавливаются соединения, между какими именно конечными точками, а также какие сигналы и события должны обнаруживаться и генерироваться в конечных точках. При этом шлюз становится простым устройством, не находящемся 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. Контроллеры медиашлюза 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.

6.1 Взаимосвязь с другими стандартами IP-Cablecom

Шлюз КТСОП IP-Cablecom состоит из трех функциональных элементов:

- контроллер медиашлюза (MGC), который содержит информацию вызова и завершает сигнализацию вызова. Данный элемент также обозначается как агент вызова;
- медиашлюз (MG), который завершает широкополосные каналы, согласно командам и под контролем MGC. Эта функция также известна под названием транкинговый шлюз (TGW);
- шлюз сигнализации (SG), который соединяет сигнализацию вызова со шлюзом КТСОП и обеспечивает функцию преобразования сигнализации.

В дополнение к протоколам шлюза КТСОП, проект IP-Cablecom также включает в себя протокол сигнализации сетевого вызова (NCS), который является профилем MGCP. В этой архитектуре стандарт протокола SDP является основным. Оба протокола и NCS, и TGCP используют протокол описания сеанса (SDP) для переноса описаний сеанса.

Системы NCS и TGCP IP-Cablecom фиксируют все состояния вызовов в CMS и агентах вызова MGC.

На приведенном ниже рисунке 1 иллюстрируется взаимосвязь между этими различными элементами:

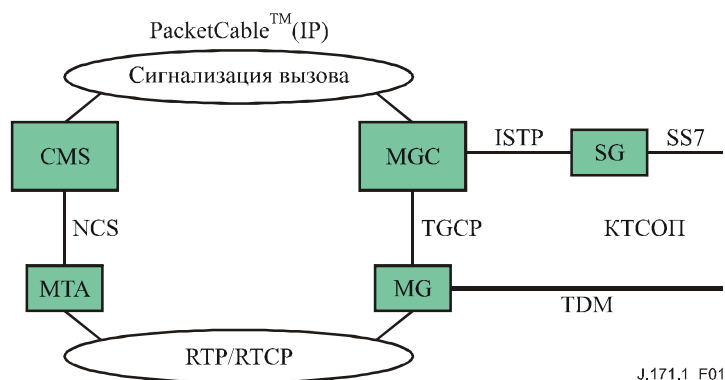


Рисунок 1/J.171.1 – Взаимосвязь между элементами NCS и TGCP

6.2 Взаимосвязь с RFC 3435 и грамматикой ABNF

RFC 3435 включает в себя формальное описание синтаксиса протокола MGCP, согласно "Расширенной BNF (форма Бэкуса-Науэра) для детальных указаний по синтаксису". На данное формальное описание ссылаются разработчики совместно функционирующих устройств. Копия синтаксиса протокола MGCP, дополненная аннотацией и отредактированная для наглядности применения указаний в Рекомендациях проекта IPCablecom, содержится в Дополнении VI. Следование данным ориентирам улучшает качество взаимодействия посредством минимизации сбоев, вызванных разночтением синтаксиса и грамматики.

7 Интерфейс управления медиашлюзом (MGCI)

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

7.1 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 контроллеров медиашлюза.

Несмотря на это, следует признать, что ни в одном из этих случаев нет "соединения", установленного внутри сети IPCablecom, т.к. под термином "соединение" понимается соединение внутри переключаемой линии связи КТСОП. Термины "вызов" и "соединение" в этом контексте (и везде в данной Рекомендации) используются для удобства ссылок, и не указывают на любое фактическое техническое или другое подобие между сетью IPCablecom и КТСОП.

7.1.1 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 имя домена (domain-name) является абсолютным именем domain-name, как определено в стандарте IETF RFC 1034, которое содержит базовую часть; таким образом, примером доменного имени может служить конструкция:

```
MyTrunkingGateway.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.

Транкинговые шлюзы имеют одну или несколько конечных точек (например, одну конечную точку для каждой транкинговой линии связи) 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, линия DS-0 (уровень цифрового сигнала 0) или аналоговая линия доступа. 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 элемент, а наиболее специфическим компонентом – правый крайний элемент. Более формально, локальное имя конечной точки ДОЛЖНО соответствовать следующим правилам:

- Отдельные элементы локального имени конечной точки должны быть разделены одиночной наклонной чертой ("/", шестнадцатеричное 2F в коде ASCII).
- Отдельные элементы 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 элементов ряда имен, которые должны быть подставлены. Таким образом, если полное локальное имя конечной точки выглядит как:

term1/term2/term3

и один из элементов 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 элемент подставляется, тогда все элементы справа от него также должны подвергаться операции подстановки.
- В случаях, когда одновременно используются подстановочные знаки "звездочка" и "доллар", знаки доллара допускаются только справа; таким образом, если элемент имел подстановочный знак "доллар", то все элементы справа от него также должны иметь подстановочные знаки "доллар".
- Элемент, представленный знаком "звездочка", должен интерпретироваться следующим образом: "использовать *все* значения этого элемента, известные в области применения рассматриваемого транкингового шлюза".
Если не определено иначе, это относится ко всем конечным точкам, сконфигурированным для услуги, независимо от действительного состояния услуги, т.е. находится она в эксплуатации или выведены из эксплуатации.
- Элемент, представленный знаком "доллар", должен интерпретироваться следующим образом: "использовать *любое одно* значение этого элемента, известное в области применения рассматриваемого транкингового шлюза". Если не определено иначе, это относится только к тем конечным точкам, которые находятся в эксплуатации.
- Каждый тип конечной точки может определять дополнительную деталь в правилах именовании для этого типа конечной точки; однако такие правила не должны противоречить изложенным выше правилам.

Следует отметить, что различные типы конечных точек или даже различные субэлементы, например "линии", в пределах одного и того же типа конечной точки будут приводить к двум различным локальным именам конечных точек. Следовательно, каждая "линия" будет интерпретироваться, как отдельная конечная точка.

Следует отметить, что поскольку часть доменного имени является частью идентификатора конечной точки, различные формы написания и различные значения, относящиеся к одному и тому же объекту, делают невозможным свободный обмен ими. Всегда НЕОБХОДИМО перезагружать последние полученные форму и значение.

7.1.1.1 Имена конечных точек транкингового шлюза

Конечные точки транкинговых шлюзов ДОЛЖНЫ поддерживать дополнительные соглашения об именовании, описанные в данном пункте.

Транкинговые шлюзы будут поддерживать следующий основной тип конечной точки:

- ds Линия связи DS-0.

Основной тип конечной точки обеспечивается дополнительной информацией о типе сигнализации, поддерживаемой транкинговой линией связи, а также данный тип конечной точки выполняет роль системной коммутации.

7.1.1.1.1 Конечные точки транкинговой линии связи

В дополнение к соглашениям об именовании, описанным выше, локальные имена конечных точек для конечных точек типа "ds" для транкингового шлюза КТСОП, ДОЛЖНЫ подчиняться следующим правилам:

- Локальные имена конечных точек будут состоять из набора элементов, отделенных друг от друга наклонной чертой ("/"), которые описывают физическую иерархию внутри шлюза:

ds/<unit-type1>-<unit #>/<unit-type2>-<unit #>/.../<channel #>

- Первый элемент (ds) обозначает используемое именование схемы конечной точки и основной тип конечной точки.

- Последний элемент – это двоичное число, которое обозначает номер канала² на самом низком уровне иерархии.
- Промежуточные элементы между первым элементом (ds) и последним элементом (номер канала) представляют собой промежуточные уровни иерархии и состоят из <unit-type> и <unit #>, отделенных друг от друга тире ("-"), где:
 - <unit-type> определяет отдельный уровень иерархии. Значения <unit-type>, здесь определены s", "su", "oc3", "ds3", "e3", "ds2", "e2", "ds1", "e1", где "s" указывает номер отсека, а "su" указывает подмодуль в отсеке. Другие значения, определяющие физические уровни иерархии, которые не включены в приведенный список, но, удовлетворяющие тем же основным правилам наименования, тоже будут являться допустимыми.
 - <unit #> – это десятичное число, которое используется для связи с конкретным значением объекта <unit-type> в данном уровне иерархии.
- Число уровней и именований этих уровней основано на физической иерархии внутри медиашлюза, что иллюстрируется следующими примерами:
 - медиашлюз, у которого имеется несколько номеров интерфейсов сигнала DS1:

ds/ds1-#/ #
 - медиашлюз, у которого имеется несколько номеров интерфейсов OC3, который содержит сформированные по каналам иерархии сигналы DS3 и DS1:

ds/oc3-#/ds3-#/ds1-#/ #
 - медиашлюз, содержащий несколько номеров отсеков, в каждом отсеке имеется несколько номеров интерфейсов сигнала DS3:

ds/s-#/ds3-#/ds1-#/ #
- Некоторые конечные точки могут не содержать все возможные уровни иерархии; однако все уровни, поддерживаемые данной конечной точкой, содержатся в схеме именования конечной точки. Например, DS3 без синхронизации DS1 может представляться в виде следующей схемы именования:

ds/s-#/ds3-#/ #

Однако DS3 с синхронизацией DS1 не может быть представлен в виде этой схемы именования:
- Именованное подстановочных знаков соответствует соглашениям, установленным в п. 7.1.1, где знак звездочка ("*") относится к "все"("all"), и знак доллар ("\$") относится к "любой" ("any"). Также поддерживается диапазон условных подстановочных знаков "[N–M]", представленный "диапазоном" каналов от канала N до канала M включительно:
 - следует отметить, что использование подстановочного знака "все" для первого элемента (ds) относится ко всем типам конечных точек в медиашлюзе, независимо от их типа. Как правило, такая возможность используется в административных целях, например, контроль или перезагрузка;
 - локальное имя конечной точки может быть предопределено путем предоставления несколько меньшего, чем обычно числа элементов, начинающихся с левой стороны имени конечной точки. В таком случае, определенным пропущенным элементам справа от последнего элемента присваивается подстановочный знак "*", относящийся к "все из", исключая определенные элементы, содержащие подстановочный знак "любой из", в этом случае оставшимся определенными элементами справа от последнего элемента присваивается подстановочный знак "любой из".
 - Если только разрешено использование подстановочного знака "все", диапазон каналов подстановочных знаков "[N–M]" может быть использован в последнем элементе (т. е. <channel-#>) вместо локального имени конечной точки. Далее, подстановочный знак "диапазон" будет относиться ко всем каналам от N до M. Правила и ограничения,

² Следует отметить, что использование термина "канал" противоположно использованию термина "временной слот".

применяемые в использовании подстановочного знака "все" применяются также в использовании подстановочного знака "range".

В следующих примерах иллюстрируется использование подстановочных знаков:

<code>ds/ds1-3/*</code>	Все каналы в ds1 номер 3 в рассматриваемом медиашлюзе.
<code>ds/ds1-3/\$</code>	Любой канал в ds1 номер 3 в рассматриваемом медиашлюзе.
<code>ds/*</code>	Все конечные точки транкинговой линии связи в рассматриваемом медиашлюзе.
<code>*</code>	Все конечные точки (независимо от типа конечной точки) в рассматриваемом медиашлюзе.
<code>ds/ds1-3/[1-24]</code>	Каналы с 1 по 24 в ds1 номер 3 в рассматриваемом медиашлюзе.

Выше определяются канонические имена для конечных точек в транкинговом шлюзе. Ожидается, что совмещение имен может быть поддержано в будущих версиях этой Рекомендации, например, поддержка связывания многочисленных линий DS-0 для видеовызовов, например, в форме "ds/ds1-1/NO-1".

7.1.2 Имена вызовов

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

Как минимум, идентификаторы вызовов ДОЛЖНЫ быть уникальными в рамках группы контроллеров вызова, которые управляют одними и теми же шлюзами. Однако вопросы координации этих идентификаторов вызовов между контроллерами медиашлюзов выходят за рамки настоящей Рекомендации. Когда контроллер медиашлюза 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.

7.1.3 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.

7.1.4 Имена контроллеров медиашлюза и другие объекты

Протокол управления медиашлюза предназначен для повышения надежности сети и позволяет осуществлять реализацию резервных контроллеров шлюзов медиа. Это означает, что между объектами и платформами аппаратных средств или сетевыми интерфейсами нет фиксированной связи.

Имена MGC, подобно именам конечных точек, состоят из двух частей. Локальная часть имени не имеет какой-либо внутренней структуры. Примером имени MGC может служить имя:

```
mgc1@mgc.whatever.net
```

Надежность обеспечивается с помощью следующих мер:

- Объекты, такие как транкинговые шлюзы или контроллеры шлюзов медиа, 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 имен доменов (DNS – Domain Name System). Контроллеры медиашлюзов 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 имен доменов и службы DNS, концепция "уведомленного объекта" является основной для обеспечения надежности и восстановления после отказа в протоколе MGCP. "Уведомленным объектом" для конечной точки является контроллер медиашлюза, 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 контроллеру медиашлюза команду, она ДОЛЖНА послать команду текущему "уведомленному объекту", к конечной(ым) точке(ам) которого относится команда. При запуске "уведомленный объект" ДОЛЖЕН быть установлен на обеспечиваемое значение. Большинство команд, посылаемых контроллером медиашлюза, имеют возможность неявным образом именовать "уведомленный объект" путем использования параметра "NotifiedEntity". "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 образом³, тогда этот объект будет установлен по умолчанию на исходный адрес последней команды обработки соединения или запроса уведомления, принимаемого для конечной точки. Таким образом, контрольная проверка не изменит "уведомленный объект".

В п. 7.4 содержится более подробное описание надежности и восстановления после отказа.

7.1.5 Отображение цифр набора номера

В протоколе MGCP, MGC может запросить шлюз накапливать цифры, набираемые пользователем. Эта функция обычно предназначена для использования аналоговых линий доступа с резидентскими шлюзами для сбора цифр, набираемых пользователем; она также может быть использована для интерфейсов PBX CAS. Вместо того чтобы передавать каждую цифру контроллеру медиашлюза для обнаружения, лучше обеспечить контроллер медиашлюза грамматикой, описывающей какое количество цифр следует накопить до момента уведомления MGC. Такая грамматика известна под названием *отображения цифр набора номера*.

В настоящей версии Рекомендации TGCP ни один из типов транкинговых линий связи не нуждается в отображениях цифр набора номера, поэтому отображения цифр набора номера не являются частью этой Рекомендации.

7.1.6 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 off-hook (состояние занятости абонентского шлейфа). Агент вызова также может запрашивать применение определенных сигналов в конечной точке, например, состояния контроля посылки вызова.

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 транкинговых линий связи ISUP, а другой пакет – другую группу событий и сигналов для транкинговых линий связи СЧ. 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 определяет отдельное пространство имен. Имена пакетов и коды событий – это нечувствительные к регистру строки букв, цифр и дефисов с ограничением, состоящим в том, что дефисы НЕ ДОЛЖНЫ быть первым или последним символом в имени. Может потребоваться параметризация некоторых кодов событий дополнительными данными, что осуществляется внесением параметров между множеством круглых скобок. Имя пакета отделяется от кода события косой чертой ("/"). Имя пакета может быть исключено из имени события, и в этом случае предполагается наличие имени пакета по умолчанию для рассматриваемого типа конечной точки. Например, для транкинговой линии связи ISUP с пакетом ISUP (имя пакета "IT"), являющимся пакетом по умолчанию, считаются равными два следующих имени событий:

- IT/oc Операция завершена в пакете ISUP для транкинговой линии связи ISUP.
- oc Операция завершена в пакете ISUP (по умолчанию) для транкинговой линии связи ISUP.

³ Это может произойти в результате описания пустого параметра NotifiedEntity.

В приведенной ниже таблице 1 перечислены типы конечных точек транкинговых шлюзов и, определенные в данной Рекомендации, пакеты для них. Контроллер медиашлюза ДОЛЖЕН поддерживать все пакеты, перечисленные в таблице 1. Медиашлюз ДОЛЖЕН поддерживать пакеты "IT", "FXR" и "XRM". Медиашлюз МОЖЕТ поддерживать пакеты "MO" и "MT".

Таблица 1/J.171.1 – Пакеты, связанные с типами конечных точек

Тип конечной точки	Пакет	Имя пакета	Пакет по умолчанию	Первая версия, в которой введено требование поддержки пакета
DS-0	Транкинговая линия ISUP	IT	Да	1.0
DS-0	MF OSS	MO	Нет	1.5
DS-0	Завершение MF	MT	Нет	1.5
DS-0	FAX	FXR	Нет	1.5
DS-0	Метрика VoIP	XRM	Нет	1.5

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 TGCP, или, возможно, и к тому и к другому.

Каждый пакет ДОЛЖЕН иметь определение пакета, которое ДОЛЖНО описывать имя пакета, и определение каждого события, принадлежащего пакету. Определение события ДОЛЖНО включать точное имя события, т. е. код события, определение события открытым текстом и, если необходимо, точное определение соответствующих сигналов, например, точных частот звуковых сигналов, таких как ringback (посылка вызова) и fax tones (факсимильные тональные сигналы). События должны далее указывать, устойчивы ли они (см. п. 7.3.1) и содержат ли они контролируемые состояния событий (см. п. 7.3.8.1). 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. 7.3.1.

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. Шлюз, 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:

IT/all относится ко всем событиям в пакете транкинговой связи ISUP для транкинговой линии связи ISUP;

*/all для транкинговой линии связи ISUP; 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:

IT/rt@0A3F58

Если нужно удалить соединение в тот момент, когда в нем определяются или на него подаются событие или сигнал, конкретное определение события или образование сигнала просто ДОЛЖНО быть остановлено. В зависимости от сигнала конечной точке СЛЕДУЕТ создавать ситуацию сбоя, т. е. если тип сигнала – ТО, то будет генерироваться событие "operation failure" (ошибка операции), поскольку соединение, связанное с этим сигналом, будет удалено раньше, чем истечет время подачи сигнала. Действие при уведомлении, связанное с сообщением о сбое, должно соответствовать операциям по уведомлению так, как это определено для обработки параметра NotificationRequest (запрос на уведомление) (п. 7.3.1).

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

IT/ma@*

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

IT/rt@\$

При обработке команды с использованием подстановки "текущее соединение" подстановочный символ "\$" ДОЛЖЕН быть восстановлен шлюзом до значения текущего соединения. Если подчиненная команда явно (например, через проверку соединения) или неявно (например, через устойчивость) ссылается на такое событие, восстановленное значение ДОЛЖНО быть использовано шлюзом. Другими словами, подстановка "текущего соединения" восстанавливается единожды, что происходит при начальной обработке команды, в которую она включена явно.

Идентификатор соединения или замена подстановочного знака могут быть использованы в сочетании с обозначениями "all packages" (все пакеты) и "all events" (все события). Например, запись:

/all@

может быть использована для обозначения всех событий во всех соединениях для используемой(ых) конечной(ых) точки (точек).

Однако использование подстановочных знаков "все пакеты" и "все события" настоятельно не рекомендуется.

Агенты вызова должны быть способны действовать в окружении, где некоторые конечные точки поддерживают не все пакеты. Конечная точка, получившая команду, ссылающуюся на пакет, который она не поддерживает, должна ответить сообщением об ошибке 518 (неподдерживаемый или неизвестный пакет). На получение такого сообщения об ошибке, агент вызова может попытаться послать ту же команду снова без параметра пакета, хотя в случаях, если первоначальная команда содержала параметры для множества пакетов, агент вызова может оказаться не способен определить, какой(ие) конкретно пакет(ы) исключить. Агент вызова может также использовать команду AuditEndpoint (проверить конечную точку) для определения набора пакетов, поддерживаемых конечной точкой.

7.2 Использование протокола SDP

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

В протоколе SDP предусматривается описание конференц-связи мультимедиа. Профиль TGCP будет поддерживать только установление аудиосоединений с использованием типа медиа "audio" (аудио).

В протоколе SDP предусматривается описание факсимильного аппарата в режиме реального времени, использующего тип медиа (сигнала) "image" (изображение). Профиль TGCP будет поддерживать установление факсимильных соединений с использованием типа медиа "изображение".

7.3 Функции управления шлюзом

В данном пункте приводится описание команд протокола MGCP в форме интерфейса API типа удаленного вызова процедуры (RPC), который будет именоваться интерфейсом управления медиашлюзом (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, представленные 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, определены в п. 8. Транкинговые медиашлюзы и агенты вызовов **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:

- Контроллер медиашлюза может выдавать команду NotificationRequest для шлюза, приказывая шлюзу отслеживать конкретные события, такие как занятие факсимильного тонального сигнала в определенной конечной точке
- Далее, шлюз будет использовать команду Notify (уведомить) для информирования контроллера медиашлюза о том, когда имеют место запрашиваемые события в заданной конечной точке
- Контроллер медиашлюза может использовать команду CreateConnection (создать соединение) для установления соединения, которое завершается в конечной точке внутри шлюза.
- Контроллер медиашлюза может использовать команду ModifyConnection (модифицировать соединение) для изменения параметров, связанных с ранее установленным соединением.
- Контроллер медиашлюза 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.
- Контроллер медиашлюза может использовать команды AuditEndpoint (проконтролировать конечную точку) и AuditConnection (проконтролировать соединение) для проверки состояния "конечной точки" и всех связанных с ней соединений. Обычно, желательно, управление сетью сверх возможностей, предоставляемых этими командами, например, желательно иметь информацию о состоянии транкингового шлюза и каждой из транкинговых линий связи. Ожидается, что такие возможности поддерживаются путем использования Протокола управления простой сетью (SNMP) и определения базы MIB (Management Information Base – информационная база управления), что выходит за рамки данной Рекомендации.
- Шлюз может использовать команду RestartInProgress (перезапуск в процессе осуществления) для уведомления контроллера медиашлюза о том, что конечная точка или группа конечных точек, управляемых шлюзом, выводятся из эксплуатации или снова вводятся в эксплуатацию

Эти услуги позволяют контроллеру (обычно контроллеру медиашлюза) давать шлюзу команду на создание соединений, которые завершаются в конечной точке, присоединенной к шлюзу, и получать информацию о событиях, происходящих в конечной точке. В настоящее время конечная точка транкингового шлюза ограничивается конкретной транкинговой линией связи в транкинговом шлюзе.

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. Каждое соединение квалифицируется параметром "режим" (mode), который может быть установлен в состояние "только передача" (sendonly), "только прием" (recvonly), "передача/прием" (sendrecv), "неактивный" (inactive), "кольцевая проверка линий" (loopback), "проверка целостности" (conttest), "кольцевая проверка сети" (netwloop) или "проверка целостности сети" (netwttest). Параметр "режим" определяет, могут ли быть отосланы и/или получены пакеты мультимедиа по соединению; однако это не влияет на протокол RTP.

Аудиосигналы, полученные от конечной точки **ДОЛЖНЫ** посылаться любому соединению, у которого установлен режим либо "только передача", либо "передача/прием", исключая конечную точку, имеющую соединение в режиме "кольцевая проверка линий" или "проверка целостности". Тем не менее, звуковой сигнал, сгенерированный путем применения сигнала в соединении, **ДОЛЖЕН** передаваться по соединению для всех режимов, исключая режим "кольцевая проверка сети".

Обработка аудиосигналов, полученных по таким соединениям, определяется также параметрами режима:

- Аудиосигналы, полученные в пакетах данных через соединения в режиме "неактивный", "кольцевая проверка линий" ДОЛЖНЫ быть отброшены.
- Аудиосигналы, полученные в пакетах данных через соединения в режиме "только прием" или "передача/прием" ДОЛЖНЫ быть собраны вместе и отосланы конечной точке⁴, исключая конечную точку, имеющую соединение в режиме "кольцевая проверка линий" или "проверка целостности".
- Аудиосигналы, порожденные конечной точкой ДОЛЖНЫ передаваться через все соединения, в которых установлен режим "только прием" или "передача/прием", исключая конечную точку, имеющую другое соединение в режиме "кольцевая проверка линий" или "тест целостности".

Следует отметить, что для того, чтобы обнаружить события в соединении, соединение, по умолчанию, должно находиться в одном из следующих режимов: "только прием", "передача/прием", "кольцевая проверка линий" или "проверка целостности сети". Обнаружение события применяется только для входящего звукового сигнала. Соединения в режиме "только прием", "неактивный", "кольцевая проверка линий" или "проверка целостности" не смогут нормально обнаруживать события, несмотря на то, что запрос на выполнение такого действия не считается ошибкой.

Режимы "кольцевая проверка линий" или "проверка целостности" используются во время проведения проверки целостности и сохранения. Конечная точка может иметь более чем одно соединение в одном из двух режимов: "кольцевая проверка линий" либо "проверка целостности". До тех пор, пока соединение находится в определенном режиме, и на конечной точке не размещено другого соединения находящегося в другом текущем обслуживании или режиме проверки, текущее обслуживание или операция проверки ДОЛЖНА продолжаться без прерываний. Существует два вида проверок целостности (COT), одна определена МСЭ, а другая – используется в нескольких национальных сетях. В первом случае, проверка является кольцевой проверкой линий. Создаваемый ключ пошлет тональный сигнал (исходящий тональный сигнал) в широкополосный канал связи и будет ожидать ключ завершения по цепи обратной связи. Если создаваемый ключ видит, что возвращается тот же самый тональный сигнал (возвращаемый тональный сигнал), COT считается пройденной. Если нет, COT считается непройденной. Во втором случае, исходящий и возвращаемый тональные сигналы различаются. Создаваемый ключ посылает определенный исходящий тональный сигнал. Ключ завершения определяет исходящий тональный сигнал, и утверждает отличающийся возвращаемый тональный сигнал в обратном направлении. Когда создаваемый ключ обнаружит возвращаемый тональный сигнал, COT считается пройденной. Если же создаваемый ключ не обнаружит возвращаемый тональный сигнал в течение определенного промежутка времени, то COT считается непройденной.

Если режим установлен в состояние "кольцевая проверка линий", шлюз ДОЛЖЕН вернуть входящий сигнал из конечной точки назад в ту же самую конечную точку. Данная процедура будет использоваться, согласно спецификациям МСЭ, для проверки целостности в транкинговых линиях связи. Если режим установлен в состояние "проверка целостности", шлюз уведомляется, что на другом конце линии связи инициирована процедура проверки целостности, согласно процедурам, определенным для нескольких национальных сетей. Шлюз поместит линию связи в режим ретрансляции, необходимый для двухтональных проверок целостности.

Более того, когда соединение для конечной точки находится в режиме "кольцевая проверка линий" или "проверка целостности":

- аудиосигналы, принятые из любого соединения для конечной точки, НЕ ДОЛЖНЫ посылаться конечной точке;
- аудиосигналы, принятые от конечной точки НЕ ДОЛЖНЫ посылаться никакому соединению для конечной точки.

Если режим установлен на "кольцевая проверка сети", аудиосигналы, принятые из соединения, ДОЛЖНЫ воспроизводиться в обратном направлении по тому же соединению. Режим "кольцевая проверка сети" ДОЛЖЕН просто функционировать, как отражатель пакетов по RTP. Среда медиа ДОЛЖНА пересылаться конечной точке.

Режим "проверка целостности сети" используется для проверки целостности через сеть IP. По сети IP к конечным точкам передается сигнал, характерный для типа конечной точки, и тогда предполагается, что конечная точка воспроизводит этот сигнал в сети IP. До передачи в обратном направлении этот сигнал ДОЛЖЕН пройти через внутреннее декодирование и повторное кодирование. Для конечных

⁴ Тем не менее, конечные точки TGSP к настоящему времени не требуют поддержки смещения.

точек DS-0 этот сигнал будет аудиосигналом, который НЕ ДОЛЖЕН передаваться дальше к линии связи, подключенной к конечной точке, независимо от текущего состояния занятости этой линии связи.

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

Пожалуйста, обращайтесь к Дополнению I, в котором иллюстрируется взаимодействие режимов.

7.3.1 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 контроллер медиашлюза, может тогда решить, что в соединениях, связанных с этой конечной точкой, должен использоваться другой тип кодирования, и дать соответствующую команду шлюзу⁵.

ReturnCode

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

EndpointId – это идентификатор конечной(ых) точки(точек) в шлюзе, где выполняется команда NotificationRequest. Идентификатор EndpointId ДОЛЖЕН подчиняться правилам для имен конечных точек, определенных в п. 7.1.1. Подстановочный знак "any of" ("любой из") НЕ ДОЛЖЕН использоваться. Шлюз, получивший команду NotificationRequest с соглашением подстановочного знака "любой из", ДОЛЖЕН послать в ответ ошибку (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть осуществлена, потому что конечная точка неизвестна).

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 "все из" 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 "все из", которую они не могут обработать по причине наличия ошибки.

Параметр **NotifiedEntity** (уведомленный объект) – это необязательный параметр, который определяет новый "уведомленный объект" для конечной точки. При его использовании ДОЛЖНО быть определено все имя контроллера медиашлюза, включающее как локальное имя, так и доменное имя – даже если для доменного имени используется заключенный в скобки адрес IP. Более подробная информация содержится в пп. 7.1.1 и 7.1.4. Однако если представлено только доменное имя, медиашлюз ДОЛЖЕН использовать доменное имя в качестве идентификатора агента вызова.

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

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 сигналы⁶:

- проверка целостности;
- установка вызова MF OSS.

⁵ Этой новой командой будет команда ModifyConnection.

⁶ Полный список сигналов содержится в Приложении А.

В зависимости от режима использования сигналы подразделяются на различные типы:

- Включено/выключено (On/off (OO)) – Поданные один раз, 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. Будучи включенным, сигнал OO НЕ ДОЛЖЕН выключаться, пока от контроллера медиашлюза не поступит явная команда сделать это; или произойдет перезапуск конечной точки.
- Выдержка времени (Time-out (TO)) – Поданные один раз, 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 А.1). Сигналом TO может быть тональный сигнал "размещение вызова MF" с выдержкой времени 16 секунд. Если событие происходит до истечения 16 секунд, то по умолчанию сигнал прекратится⁷. Если сигнал не прекращается, то для него будет иметь место выдержка времени, сигнал прекращается и генерируется событие "операция завершена", о котором контроллер медиашлюза мог послать или не послать запрос об уведомлении. Если контроллер медиашлюза послал запрос об уведомлении о событии "операция завершена", то событие "операция завершена", посланное контроллеру медиашлюза, будет содержать имя сигнала (имена сигналов) с выдержкой времени⁸. Сигнал(ы), сгенерированный(ые) в соединении, будет(ут) 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 (имена) сигнала(ов) с выдержкой времени⁸.
- 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.

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

Если сигнал подается в соединение, которое получило параметр RemoteConnectionDescriptor (см. п. 7.3.3), тогда поток медиа, 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. Текущие активные сигналы выдержки времени, не представленные в новом списке, ДОЛЖНЫ быть прекращены, а новый(ые) представленный(ые) сигнал(ы) теперь станет(ут) активным(и). 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 текущего активного 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.

⁷ Действие "удерживание сигнала(ов) активным(и)" может отменять такое поведение.

⁸ Если параметры были посланы сигналу, то о них не будет сообщаться.

Сигналы, определяемые в настоящее время, 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:

- занятость (линии);
- тональные сигналы факсимильной связи;
- операция завершена;
- входящий вызов СЧ.

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;
- накоплений событий;
- игнорирование события;
- удерживание 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"⁹. Таким образом, устойчивые события можно неформально рассматривать как всегда включенные неявным образом в список параметра RequestedEvents с действием "уведомление", хотя будет выполняться самое обычное обнаружение событий¹⁰. 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. 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 2 определены допустимые сочетания действий:

⁹ Таким образом, параметр RequestIdentifier будет параметром RequestIdentifier команды NotificationRequest.

¹⁰ 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.

Таблица 2/J.171.1 – Действия, связанные с событиями

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

Если клиент получает запрос с неправильным действием или недопустимым сочетанием действий, он ДОЛЖЕН вернуть контроллеру медиашлюза 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.

Контроллер медиашлюза может послать шлюзу команду NotificationRequest с пустым списком параметра RequestedEvents. Однако устойчивые события все еще будут обнаруживаться, и о них будут посылаться уведомления.

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

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

Сигналы, используемые в параметре SignalRequests, синхронизированы с набором событий, описанных или подразумеваемых в параметре RequestedEvents, за исключением случая отмены действием "удерживание сигнала(ов) активным(и)". 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.

Если требуется, чтобы сигнал(ы) выдержки времени продолжался(ись) 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(ов), когда наступает искомое событие, может быть использовано действие "Embedded NotificationRequest" ("встроенный запрос на уведомление"). Это действие может включать новый список параметров RequestedEvents, SignalRequests, и новые параметры RequestedEvents. Тем не менее, действие "Embedded NotificationRequest" не может содержать другое такое же действие "Embedded NotificationRequest". Когда действие "Embedded NotificationRequest" активизировано, список наблюдаемых событий и буфер карантина будут оставаться без изменений (см. п. 7.4.3.1).

Встроенное действие NotificationRequest позволяет агенту вызова установить "мини-сценарий", подлежащий исполнению шлюзом сразу же после обнаружения связанного с ним события. Все команды SignalRequests, описанные во встроенном действии NotificationRequest, начинают выполняться немедленно. Значительное внимание должно быть уделено предотвращению

несоответствий между контроллером медиашлюза 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 контроллеру медиашлюза 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. Когда предоставляется список изменений режимов соединений, изменения режимов соединений ДОЛЖНЫ применяться по одному в порядке слева направо. Когда все изменения режимов соединений заканчиваются, генерируется событие "операция завершена", параметризованное именем завершеного действия (подробная информация содержится в Приложении А). Как только при каком-либо изменении режима соединения происходит ошибка, генерируется событие "ошибка операции", параметризованное именем ошибочного действия и изменением режима соединения (подробная информация содержится в Приложении А) – НЕ ДОЛЖНЫ предприниматься попытки выполнить остальные изменения режимов соединений, а предшествующие успешные изменения режимов соединений в списке НЕ ДОЛЖНЫ оставаться 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.

ПРИМЕЧАНИЕ. – В п. 7.4.3.1 содержится дополнительная подробная информация по семантике обнаружения событий и сообщениям. Читателю рекомендуется тщательно изучить указанный параграф.

Конкретное определение действий, запрашиваемых через эти параметры SignalRequests, выходит за рамки базовой Рекомендации TGCP. 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 Рекомендации. 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 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).

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 необязательный параметр, который описывает варианты обработки для событий в буфере карантина (см. п. 7.4.3.1), т. е. событий, которые были обнаружены шлюзом до поступления этой команды NotificationRequest, но уведомление о которых еще не было послано контроллеру медиашлюза. 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 точкой, принимающей команду NotificationRequest с неподдерживаемым значением параметра QuarantineHandling, СЛЕДУЕТ ответить кодом ошибки 508 (неподдерживаемый параметр QuarantineHandling).

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

Параметр **DetectEvents** (обнаруживаемые события) – это необязательный параметр, описывающий минимальный список событий, которые должен обнаружить по запросу шлюз в состояниях "уведомление" и "шаг блокировки". Такой список является устойчивым, до тех пор, пока не будут определены новые значения. Дальнейшие пояснения относительно данного параметра можно найти в п. 7.4.3.1.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

7.3.2 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. 7.1.1. Этот идентификатор ДОЛЖЕН полностью определяться именем конечной точки, включая доменное имя шлюза. Локальная часть этого имени НЕ ДОЛЖНА использовать условный подстановочный знак. Контроллер медиашлюза, получающий команду Notification с условным подстановочным знаком ДОЛЖНА вернуть ответ с ошибкой (возвращаемой ошибке следует присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

Параметр **NotifiedEntity** (уведомленный объект) – это необязательный параметр, который идентифицирует объект, которому послано уведомление. Данный параметр равнозначен параметру NotifiedEntity команды NotificationRequest, инициировавшей это уведомление. Следует отметить, что медиашлюз может содержать только абсолютное доменное имя своего параметра NotifiedEntity, если только абсолютное доменное имя было получено в инициирующей команде NotificationRequest. В этом случае, контроллеру медиашлюза 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 (включает запрос на уведомление, встроенный в примитивы обработки соединения). 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" 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 иницилирующей команды NotificationRequest. События, которые были обнаружены в соединении, будут содержать имя этого соединения. Список будет содержать события, которые были также накоплены, (но о которых не было послано уведомление), и конечное событие, которое инициировало уведомление.

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

7.3.3 Команда CreateConnection

Команда CreateConnection (создание соединения) используется для создания соединения.

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

Эта функция используется при установлении соединения между двумя конечными точками. Соединение определяется его атрибутами и конечными точками, которые оно связывает. Транкинговый шлюз ДОЛЖЕН поддерживать, как минимум, одно соединение через конечную точку. Входные параметры в команде CreateConnection обеспечивают данные, необходимые для создания "вида" одной из двух конечных точек соединения.

Параметр **CallId** – это параметр, идентифицирующий вызов (или сеанс), которому принадлежит это соединение. Этот параметр является уникальным, как минимум, в наборе контроллеров медиашлюза, 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** – 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** ответа. В этом случае конечная точка с присвоенным идентификатором ДОЛЖНА быть в рабочем состоянии и НЕ ДОЛЖНА иметь еще какие-либо соединения. Условный подстановочный знак "все" НЕ ДОЛЖЕН использоваться. Шлюз, получающий команду CreateConnection с условным подстановочным знаком "все" ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке следует присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

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

Структура **LocalConnectionOptions** – 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:** Список буквенных наименований для алгоритма сжатия (метод кодирования/декодирования), используемого для передачи и приема мультимедийных данных по соединению ДОЛЖЕН быть определен, по меньшей мере, с одним значением. 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, кодированный с использованием другого метода кодирования, она МОЖЕТ отбросить его. Конечная точка ДОЛЖНА дополнительно указывать, какие из оставшихся алгоритмов сжатия будут поддерживаться в качестве альтернативы – для подробной информации см. п. 8.4.1. Список допустимых методов кодирования определен в Рек. МСЭ-Т J.161. ДОЛЖНЫ использоваться буквенные наименования, которые определены в п. 7.5 (таблица 3)/J.161. Неизвестные алгоритмы сжатия в случае их получения СЛЕДУЕТ игнорировать. Подробная информация, касающаяся процесса выбора кодека, содержится в п. 7.7.

ПРИМЕЧАНИЕ. – "Метод кодирования" включает кодирования аудиосигнала, изображения и видеосигнала.

- **Период пакетирования:** Период пакетирования в миллисекундах 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 если нет поля метода кодирования, описанного в структуре **LCO** (**LocalConnectionOptions** – варианты локального соединения), медиашлюз 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 пакету медиа. Следует отметить, что медиашлюз должен использовать только действительный период пакетирования в сочетании с соответствующим методом кодирования. Список допустимых периодов пакетирования описан в Спецификации J.161 – Аудио/Видео кодеки **IPCom**. Этот спецификатор НЕ ДОЛЖЕН предоставляться в одной и той же структуре **LCO** как поле "период многократного пакетирования". Медиашлюз ДОЛЖЕН возвращать информацию об ошибке (код ошибки 524 – несоответствие в структуре **LocalConnectionOptions**), когда он принимает структуру **LCO** как с полем "период пакетирования", так и с полем "период многократного пакетирования".

- **Период многократного пакетирования:** Список периодов пакетирования в миллисекундах, МОЖЕТ быть задан, тогда и только тогда, когда 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", включенного 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. Это 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 медиашлюз НЕ ДОЛЖЕН выбирать кодек с периодом пакетирования, который отличается от описанного здесь периода пакетирования. Следует отметить также, что медиашлюз должен использовать только действительный период пакетирования в сочетании с соответствующим методом кодирования. Список допустимых периодов пакетирования определен в Спецификации Аудио/Видео кодеки **IPCom** (J.161). Этот спецификатор НЕ ДОЛЖЕН предоставляться в одной и той же структуре **LCO**, как поле "период пакетирования". Медиашлюз 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 нет¹¹. Этот параметр может иметь значение "on" ("включено") (когда запрошена эхокомпенсация) или "off" ("выключено") (когда эхокомпенсация выключена). Этот параметр является необязательным. Когда этот параметр опущен, медиашлюз ДОЛЖЕН применять эхокомпенсацию. Медиашлюзу СЛЕДУЕТ в дальнейшем включать и выключать эхокомпенсацию в соответствии с Рек. МСЭ-Т V.8 при обнаружении данных в диапазоне звуковых частот. Относительно повторного включения эхокомпенсации, смотри Рек. МСЭ-Т G.168. После окончания передачи данных в диапазоне звуковых частот обработка сигналов ЭХОКОМПЕНСАЦИИ ДОЛЖНА возвращаться к текущему значению параметра эхокомпенсации. РЕКОМЕНДУЕТСЯ, чтобы обработка эхокомпенсации разрешалась шлюзу, а не оставлялось определение этого параметра контроллеру медиашлюза.
- **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 необязательным. Когда этот параметр опущен, то ДОЛЖНО использоваться значение по умолчанию 0x00 (если не будет обеспечено другое значение).
Если этот параметр представлен и имеет допустимое значение, полученное в нем значение должно быть использовано конечной точкой для заполнения параметра точка кода дифференцированной услуги (DSCP) в IP-заголовке (см. RFC 2474 для более детальной информации о DSCP). Значение параметра ДОЛЖНО быть 0x00 или кратным четырем в диапазоне от 0x01 до 0xFF (биты 6 и 7, биты ECN, зарезервированы и поэтому должны быть установлены на 00). Конечная точка ДОЛЖНА возвращать сообщение об ошибке (код ошибки 532 – Неподдерживаемое(ые) значение(я) в LocalConnectionOptions), если она получает недопустимое значение. Крайний слева "бит" параметра соответствует наиболее значимому биту в IP-заголовке.
- **Подавление пауз:** Телефонные шлюзы могут осуществлять определение активности передачи речи и избегать пересылки пакетов во время периодов паузы. Однако для определенных вызовов (например, модемных вызовов) необходимо отключение подавления пауз. Этот параметр может иметь значение "включено" (когда пауза должна подавляться) или "выключено" (когда пауза не должна подавляться). Этот параметр является необязательным. Когда этот параметр опущен, то по умолчанию используется значение "выключено". Если установлено значение "включено" при обнаружении данных в диапазоне звуковых частот, конечной точке СЛЕДУЕТ отключить подавление пауз. При окончании передачи данных в диапазоне звуковых частот управление подавлением пауз ДОЛЖНО возвращаться к текущему значению параметра подавления пауз.

В дополнении, для поддержки услуг безопасности IPCablecom используются следующие поля структуры LocalConnectionOptions (МОЖЕТ присутствовать либо один из двух, либо оба шифрокомплекта):

- **Поле RTP ciphersuite** (шифрокомплект по протоколу RTP): Список допустимых шифрокомплектов для обеспечения безопасности по протоколу RTP (Real-Time Protocol – протокол режима реального времени) в приоритетном порядке. Элементы в этом списке упорядочены по приоритету, где предпочтительным является первый шифрокомплект. Конечная точка ДОЛЖНА выбрать только один из шифрокомплектов согласно правилам, описанным в Спецификации безопасности IPCablecom (J.170). Конечной точке СЛЕДУЕТ дополнительно указать, какие из оставшихся шифрокомплектов она желает поддержать в качестве альтернатив (подробная информация содержится в п. 8.4.1). Каждый шифрокомплект представлен в виде строк в коде ASCII (Американский стандартный код обмена информацией), состоящих из двух подстрок (возможно пустых), разделенных наклонной чертой ("/"), где первая подстрока определяет алгоритм аутентификации, а вторая – алгоритм шифрования. Список допустимых шифрокомплектов описан в Рек. МСЭ-Т J.170. Параметр шифрования RTP применяется только для мультимедийных потоков RTP. Если контроллер медиашлюза включает структуру LocalConnectionOptions, в которой требуется использование только среды не-RTP (как, например, передача факса T.38 с использованием UDPTL (транспортный уровень протокола дейтаграмм пользователя), то параметр шифрокомплекта RTP НЕ ДОЛЖЕН включаться. Если в структуре LCO

¹¹ Эхокомпенсация на стороне пакетов не поддерживается.

содержатся обе среды: и среда RTP, и среда не-RTP, а при этом включен параметр шифрокомплекта, то этот параметр применяется только к среде RTP. Во всех остальных случаях, если результирующий поток медиа в соединении не является потоком медиа RTP (как, например, передача факса T.38 с использованием UDPTL), то параметр шифрокомплекта RTP ДОЛЖЕН быть игнорирован, т. е. безопасность RTP не может применяться, а параметры шифрокомплекта RTP не включаются в структуру LocalConnectionDescriptor.

- **Поле RTCP ciphersuite** (шифрокомплект по протоколу RTCP): Список шифрокомплектов для обеспечения безопасности по протоколу RTCP (Real-Time Control Protocol – протокол управления режима реального времени) 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 Спецификации безопасности IPsec (J.170). Конечной точке СЛЕДУЕТ дополнительно указать, какие из оставшихся шифрокомплектов она желает поддерживать в качестве альтернатив. См. п. 8.4.1 для подробной информации. Каждый шифрокомплект представлен в виде строки в коде ASCII, состоящей из двух (возможно, пустых) подстрок, разделенных наклонной чертой ("/"), где первая подстрока определяет алгоритм аутентификации, а вторая – алгоритм шифрования. Список допустимых шифрокомплектов описан в Рек. МСЭ-Т J.170. Параметр шифрокомплекта RTCP применяется только в протоколе RTCP для потоков медиа RTP. Если контроллер медиашлюза включает структуру LocalConnectionOptions, в которой требуется использование только среды не-RTP (как, например, передача факса T.38 с использованием UDPTL), то параметр шифрокомплекта RTCP НЕ ДОЛЖЕН включаться. Если в структуре LCO содержатся обе среды: и среда RTP, и среда не-RTP, а при этом включен параметр шифрокомплекта, то этот параметр применяется только к протоколу RTCP для среды RTP. Во всех остальных случаях, если результирующий поток медиа в соединении не является потоком медиа RTP (как, например, передача факса T.38 с использованием UDPTL), то параметр шифрокомплекта RTP ДОЛЖЕН быть игнорирован т. е. безопасность RTCP не может применяться, а параметры шифрокомплекта RTCP не включаются в структуру LocalConnectionDescriptor.

Транкинговый шлюз ДОЛЖЕН отвечать информацией об ошибке (код ошибки 524 – несоответствие структуры LocalConnectionOptions), если нарушено любое из вышеприведенных правил. Все, упомянутые выше значения по умолчанию, могут изменяться за счет процесса обеспечения.

Более того, протокол TGCP поддерживает электронное наблюдение IPsec. Если соединение является объектом для электронного наблюдения, все действительные мультимедийные пакеты, полученные по соединению, и все мультимедийные пакеты, переданные по соединению, будут тиражироваться и перенаправляться в функцию доставки электронного наблюдения¹² после включения идентификатора соединения содержания вызова. Тиражирование будет следовать режиму соединения для соединения, за исключением среды, генерированной путем применения сигналов в соединении, которое будет тиражироваться независимо от режима соединения. Например, соединение в режиме "неактивный" не будет генерировать какую-либо перехватываемую среду¹³, тогда как соединения в режиме "только передача" будет генерировать только перехватываемую среду в передаваемом направлении. Тиражированные пакеты не будут включаться в статистику для соединения. Для поддержки электронного наблюдения IPsec используются следующие поля LocalConnectionOptions:

- **Идентификатор соединения содержимого вызова (ССС)** – это 32-битовое значение, которое определяет идентификатор соединения содержимого вызова для использования в соединениях, которые являются объектами электронного наблюдения. Идентификатор добавляется к заголовкам перехватываемых речевых пакетов.
- **Назначение содержимого вызова:** Назначение содержимого вызова определяет адрес IPv4, следующий за двоеточием, и номер порта UDP. Назначение содержимого вызова определяет адрес назначения IP и порт для перехватываемого содержимого вызова.

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), т. е. поля, которые описывают сеанс в соответствии со стандартом SDP. В п. 8.4 содержится подробная информация по поддерживаемому использованию протокола SDP в профиле TGCP. Этот параметр

¹² Следует отметить, что тиражирование появляется в сетевом уровне – для более подробной информации см. PKT-SP-ESP-I01-991229.

¹³ Предполагается, что в соединении не будут применяться генерированные сигналы медиа.

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 коды. Как отмечалось выше, транкинговые шлюзы 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", "неактивный", "кольцевая проверка линий", "проверка целостности", "кольцевая проверка сети" или "проверка целостности сети". Управление этими режимами описано в начале п. 7.3.

Следует отметить, что сигналы, применяемые в соединении, не придерживаются режима соединения. В некоторых конечных точках невозможна поддержка всех режимов – см. V.1. 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", "кольцевая проверка сети", "проверка целостности сети", или если сигнал (в противоположность обнаружению события) применяется в соединении (РЕКОМЕНДОВАН код ошибки 527 – отсутствие параметра RemoteConnectionDescriptor).

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

Параметр LocalConnectionDescriptor (локальный дескриптор соединения) – это параметр, возвращаемый шлюзом, который является описанием сеанса, содержащим информацию, например, об адресах и портах RTP для "интеллектуальных" соединений, как определено в протоколе SDP. Этот параметр подобен параметру RemoteConnectionDescriptor, за исключением того, что он описывает данную сторону соединения. В п. 8.4 содержится подробная информация о поддерживаемом использовании протокола SDP в профиле TGCP. Параметр LocalConnectionDescriptor ДОЛЖЕН включаться в любой предварительный или успешный ответ на команду CreateConnection. LocalConnectionDescriptor НЕ ДОЛЖЕН включаться, если возвращается какое-либо сообщение об ошибке и соединение не образуется.

После приема команды CreateConnection, не содержащей параметр RemoteConnectionDescriptor, шлюз оказывается в "двусмысленной" ситуации относительно рассматриваемого соединения. Поскольку этот шлюз экспортировал параметр LocalConnectionDescriptor, он может, в принципе, принимать пакеты по этому соединению. Поскольку он еще не получил параметр RemoteConnectionDescriptor другого шлюза, ему неизвестно, было ли получено разрешение от контроллера медиашлюза на пакеты, которые он принимает. Таким образом, он должен лавировать между двумя опасностями, т. е. урезанием ряда важных уведомлений или приемом непонятных данных. Поведение шлюза определяется значением параметра режима (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", "кольцевая проверка линий" или "проверка целостности", то шлюз ДОЛЖЕН (как всегда) отбрасывать речевые сигналы, полученные по соединению;
- если режим был установлен в состояние "проверка целостности сети", то шлюз ДОЛЖЕН представлять предполагаемое эхо или ответ. Дальше, отражаемая или генерированная среда ДОЛЖНА посылаться источнику полученной среды.

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".

Все параметры **RequestedEvents**, **RequestIdentifier**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательным и. Они могут быть использованы контроллером медиашлюза

для эффективного включения встроенного 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 контроллер медиашлюза, который хочет передать вызов системе услуг оператора через транкинговый шлюз СЧ. Контроллер медиашлюза может:

- запросить транкинговый шлюз создать соединение, чтобы быть уверенным в том, что у медиашлюза имеются ресурсы для вызова;
- запросить транкинговый шлюз захватить линию услуг оператора и инициировать вызов;
- запросить транкинговый шлюз уведомить контроллер медиашлюза о том, когда начнется вызов.

Все, упомянутое выше, может быть выполнено с помощью одной команды CreateConnection, путем включения в нее запроса на уведомление с параметрами RequestedEvents для события ответа и с параметром SignalRequests для установления 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 начале вызова должен получить отказ в состоянии помехи приему вызова звонком, если линия связи уже занята. В этом примере, вызов не должен начинаться, если соединение не может быть установлено, и соединение не должно быть установлено, если линия связи уже занята. Вместо этого будет возвращена информация об ошибке (код ошибки 401 – линия связи уже занята), которая информирует контроллера медиашлюза 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. 7.5), которое необязательно сопровождается комментарием.

7.3.4 Команда ModifyConnection

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

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

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

Параметр **EndpointId** (идентификатор конечной точки) ДОЛЖЕН точно определяться именем конечной точки. В локальном имени НЕ ДОЛЖЕН использоваться условный подстановочный знак. Шлюз, который получает команду **ModifyConnection** с условным подстановочным знаком ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

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**. 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.

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. Параметр **LocalConnectionDescriptor** НЕ ДОЛЖЕН включаться, если возвращается какой-либо ответ с ошибкой, и соединение не было модифицировано.

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 изменены¹⁴.

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 вызова¹⁵. Когда адресная информация протокола RTP выдается транкинговому шлюзу для соединения, то транкинговому шлюзу СЛЕДУЕТ также только принимать потоки мультимедийных пакетов (и RTCP) от заданного адреса RTP. Все потоки мультимедийных пакетов, принимаемые от любого другого адреса, СЛЕДУЕТ отбрасывать. За дополнительной информацией, касающейся требований по обеспечению безопасности, следует обращаться к Рек. МСЭ-Т J.170.

Параметры **RequestedEvents**, **RequestIdentifier**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательными. Эти параметры могут быть использованы контроллером медиашлюза для включения встроенного 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**. 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 параметра либо приняты, либо получили отказ.

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

Параметр **ReturnCode** (код завершения) – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

¹⁴ Это может произойти, например, если задано изменение кодека и в старом кодеке использовалось подавление пауз, а в новом кодеке это не поддерживается. Если, например, период пакетирования не был определен далее, а новый кодек поддерживал старый период пакетирования, то значение этого параметра не изменится, в случае, если изменения не будут являться необходимыми.

¹⁵ Например, если мультимедийному пакету требуется пройти через систему защиты доступа.

7.3.5 Команда DeleteConnection (от контроллера медиашлюза)

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]
    [, SignalRequests]
    [, QuarantineHandling]
    [, DetectEvents])
```

Идентификатор конечной точки в данном виде команды DeleteConnection ДОЛЖЕН быть полностью классифицирован. Подстановочные знаки использоваться НЕ ДОЛЖНЫ. Шлюз, который получает команду DeleteConnection в такой форме с условным подстановочным знаком, ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция операция не может быть произведена, потому что конечная точка неизвестна).

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 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. Также после удаления соединения, любая кольцевая проверка, которая запрашивалась для соединения ДОЛЖНА быть отменена (до тех пор, пока у конечной точки не появится другой запрос кольцевой проверки линий соединения).

В ответ на команду DeleteConnection шлюз возвращает список параметров, описывающих состояние соединения¹⁶. Параметры соединения ДОЛЖНЫ возвращаться только в том случае, если команда прошла успешно и соединение удалено. 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. Значение ДОЛЖНО основываться на той же информации, предоставленной с помощью механизма протокола RTCP.
- **Число переданных октетов:** Общее число октетов полезной нагрузки (т. е. 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. Значение ДОЛЖНО основываться на той же информации, предоставленной с помощью механизма протокола RTCP.
- **Число полученных пакетов:** Общее число пакетов данных RTP, полученных отправителем с начала приема по соединению. В подсчет входят пакеты, полученные от разных SSRC, если отправитель использовал несколько значений. Все полученные пакеты ДОЛЖНЫ подсчитываться независимо от режима соединения или типа ошибки обработки, например аутентификация неисправна.
- **Число полученных октетов:** Общее число октетов полезной нагрузки (т. е. не включая заголовок или заполнение), полученных в пакетах данных RTP отправителем с начала приема по соединению. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Все полученные пакеты ДОЛЖНЫ подсчитываться независимо от режима соединения или типа ошибки обработки, например аутентификация неисправна.

¹⁶ Вычисленные значения не будут содержать пакеты, которое являются результатом электронного наблюдения.

- **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, в соединении не было получено ни одного пакета.
- **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 IETF RFC 1889. В подсчет входят пакеты, принятые от разных SSRC, если отправитель использовал несколько значений. Значение равно нулю, если, например, в соединении не было получено ни одного пакета.
- **Средняя задержка при передаче:** Оценка времени задержки в сети, выраженная в миллисекундах. Это среднее значение разности между отметкой времени по протоколу NTP (Network Time Protocol – сетевой протокол службы времени), 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 транкинговых шлюзов МОГУТ альтернативно оценивать среднюю задержку при передаче путем деления на два измеренного времени на передачу и подтверждение

Более детализированное определение этих переменных содержится в IETF RFC 1889.

Параметры **NotifiedEntity**, **RequestedEvents**, **SignalRequests**, **QuarantineHandling** и **DetectEvents** являются необязательными. Они могут быть использованы контроллером медиашлюза для передачи встроенного 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 линия связи разъединяется, шлюз получит команду исключить соединение и начать поиск события занятости.

Это может быть выполнено с помощью одной команды **DeleteConnection**, путем передачи также параметра **RequestedEvents** для события занятости 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** – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

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

7.3.6 Команда DeleteConnection (от транкингового шлюза)

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** в данном виде команды DeleteConnection ДОЛЖЕН быть полностью классифицирован. Условные подстановочные знаки использоваться НЕ ДОЛЖНЫ. Шлюз, который получает команду DeleteConnection с условным подстановочным знаком, ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

Параметр **Reason-code** – это текстовая строка, начинающаяся с числового кода причины и необязательно сопровождаемая описательной текстовой строкой. Список кодов причины содержится в п. 7.6.

В дополнение к параметрам **CallId**, **EndpointId** и **ConnectionId** транкинговый шлюз будет также посылать параметры соединения, которые будут возвращаться контроллеру медиашлюза в ответ на поступившую от него команду DeleteConnection. Код причины указывает на причину выдачи команды DeleteConnection.

ReturnCode – это параметр, возвращаемый контроллером медиашлюза. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

7.3.7 Команда DeleteConnection (от контроллера медиашлюза для нескольких соединений)

Для одновременного исключения нескольких соединений контроллер медиашлюза 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** в данном виде команды DeleteConnection НЕ ДОЛЖЕН использоваться подстановочный знак "любой из". Все соединения для конечной(ых) точки (точек) с заданным идентификатором вызова CallId будут исключены. По этой команде отдельные статистические данные или параметры вызова не возвращаются. Шлюз, который получает команду DeleteConnection (от контроллера медиашлюза для нескольких соединений) с условным подстановочным знаком "любой из", ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

Команда DeleteConnection может быть также использована контроллером медиашлюза для исключения всех соединений, завершающихся в данной конечной точке:

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

При этом виде команды DeleteConnection контроллеры медиашлюза 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 "все", как описано в п. 7.1.1. Условный подстановочный знак "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. Также после удаления соединения, любая кольцевая проверка линий, которая запрашивалась для соединения ДОЛЖНА быть отменена (до тех пор, пока у конечной точки не появится другой запрос кольцевой проверки линий соединения).

Параметр **ReturnCode** – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

7.3.8 Контроль

Протокол MGCP основан на архитектуре централизованного управления вызовом, где контроллер медиашлюза 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 транкинговыми шлюзами и контроллерами медиашлюза в системе IP-Cablecom обеспечивается характерный для протокола MGCP механизм контроля, позволяющий контроллеру медиашлюза контролировать состояние конечной точки и соединения и восстанавливать характерные для протокола функциональные возможности конечной точки.

Для транкинговых шлюзов определены две команды контроля:

AuditEndPoint (контроль конечной точки): Используется контроллером медиашлюза 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 транкингового шлюза в отличие от отдельных конечных точек. Ожидается, что такие возможности должны поддерживаться путем использования простого протокола управления сетью (SNMP) и определения базы MIB (Management Information Base – информационная база управления) для транкингового шлюза; оба эти вопроса выходят за рамки настоящей Рекомендации.

7.3.8.1 Команда AuditEndPoint

Команда AuditEndPoint (контроль конечной точки) может быть использована контроллером медиашлюза для выяснения состояния той или иной конечной точки.

```
{ ReturnCode
  [, EndPointIdList]
  [, NumEndPoints] } |
{ ReturnCode
  [, RequestedEvents]
  [, SignalRequests]
  [, RequestIdentifier]
  [, NotifiedEntity]
  [, ConnectionIdentifiers]
  [, DetectEvents]
  [, ObservedEvents]
  [, EventStates]
  [, MaxMGCPDatagram]          [, Capabilities] }
  ← AuditEndPoint(EndpointId
    [, RequestedInfo] |
    { [, SpecificEndpointID]
      [, MaxEndpointIDs] } )
```

Параметр **EndpointId** (идентификатор конечной точки) определяет конечную точку, которая находится под контролем. Условный подстановочный знак "любой из" использоваться НЕ ДОЛЖЕН. Шлюз, который получает команду AuditEndPoint с условным подстановочным знаком ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

Условный подстановочный знак "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**, являющемся просто списком параметров **SpecificEndpointIds** – параметр **RequestedInfo** НЕ ДОЛЖЕН быть включен в список в этом случае. Параметр **MaxEndPointIDs** – это численное значение, которое указывает на максимальное число возвращаемых параметров **EndPointId**. Если существуют дополнительные конечные точки, то ДОЛЖЕН присутствовать параметр возврата **NumEndPoints**, указывающий на общее число конечных точек, соответствующих заданному параметру **EndPointID**. Чтобы восстановить следующий блок параметров **EndPointID**, параметр **SpecificEndPointID** 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, **SignalRequests**, **RequestIdentifier**, **NotifiedEntity**, **ConnectionIdentifiers**, **DetectEvents**, **ObservedEvents**, **EventStates**, **VersionSupported**, **MaxMGCPDatagram** и **Capabilities**.

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

Если конечную точку запрашивают о параметре, который ей известен, но у нее не имеется для него значения, то конечная точка НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого такой параметр ДОЛЖЕН быть включен в ответе с пустым значением параметра.

Только в успешном случае, в ответ команды **AuditEndPoint** ДОЛЖНА, в свою очередь, включаться информация о каждом из параметров, для которого была запрошена информация для контроля. Следует отметить, что параметры, которые ниже явным образом отмечены как "необязательные", и которые не поддерживаются конечной точкой, в ответе опускаются:

- **RequestedEvents** (запрашиваемые события): Текущее значение параметра **RequestedEvents**, которое использует конечная точка, включая действие, связанное с каждым событием. Устойчивые события включаются в список.
- **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 сигналов¹⁷. Сигналы выдержки времени с исчерпанием этого времени и передаваемые в текущее время короткие сигналы в список не включаются. 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**: Текущий "уведомленный объект" для конечной точки. Следует отметить, что медиашлюз может содержать только абсолютное доменное имя своего параметра **NotifiedEntity**, если ему было предоставлено только абсолютное доменное имя через параметр **NotifiedEntity** сообщения или подтверждения сообщения TGCP. В этом случае, контроллеру медиашлюза 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, занято, если транкинговая линия СЧ для конечной точки сейчас занята. В определении отдельных событий будет указываться, имеет ли рассматриваемое событие связанное с ним контролируемое состояние.

¹⁷ В текущее время не должно быть ожидающих обработки коротких сигналов.

- **VersionSupported:** Список версий протокола, поддерживаемых конечной точкой.
- **MaxMGCPDatagram** (максимальная дейтаграмма протокола MGCP): Максимальный размер дейтаграммы протокола MGCP в байтах, поддерживаемой конечной точкой (см. п. 8.5.3). Из этого значения исключается любая служебная информация нижнего уровня. Поддержка для этого параметра является необязательной. Если значение не возвращается, то по умолчанию предполагается максимальный размер дейтаграммы протокола MGCP.
- **Capabilities** (возможности): Возможности для конечной точки, аналогичные параметру LocalConnectionOptions и включающие пакеты событий и режимы соединения. Если необходимо указать, что некоторые параметры, такие, например, как подавление пауз, совместимы только с некоторыми кодеками, шлюз будет возвращать несколько наборов возможностей.
 - **Алгоритм сжатия** Список поддерживаемых кодеков. ДОЛЖНЫ использоваться буквенные наименования, определенные в п. 7.5 (таблица 3) Спецификации Аудио/Видео кодеки IP-Cablecom (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 эхокомпенсация¹⁸.
 - **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.
 - **Режимы** Список поддерживаемых режимов соединения.
 - **Шифрокомплекты протокола RTP** Список алгоритмов аутентификации и шифрования, поддерживаемых для протокола RTP.
 - **Шифрокомплекты протокола RTCP** Список алгоритмов аутентификации и шифрования, поддерживаемых для протокола RTCP.
 - **Электронное наблюдение** Поддерживается или нет электронное наблюдение IP-Cablecom.

Затем контроллер медиашлюза может принять решение об использовании команды AuditConnection для получения дальнейшей информации о соединениях.

Параметр **ReturnCode** является параметром, возвращаемым шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

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

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

¹⁸ В данном случае, все конечные точки TGCP должны поддерживать эхокомпенсацию.

7.3.8.2 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. Шлюз, который получает команду AuditConnection с условным подстановочным знаком ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна). Параметр (возможно, пустой) **RequestedInfo** (запрашиваемая информация) описывает информацию, которая запрошена для параметра **ConnectionId** в пределах заданного параметра EndpointId. По данной команде может быть проконтролирована следующая информация соединения:

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

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

Если конечную точку запрашивают о параметре, который ей известен, но у нее не имеется для него значения, то конечная точка НЕ ДОЛЖНА генерировать сообщение об ошибке; вместо этого такой параметр ДОЛЖЕН быть включен в ответе с пустым значением параметра.

Только в успешном случае, в ответ команды AuditConnection ДОЛЖНА, в свою очередь, включаться информация о каждом из параметров, для которого была запрошена информация для контроля. Следует отметить, что параметры, которые ниже явным образом отмечены как "необязательные", и которые не поддерживаются конечной точкой, в ответе опускаются:

- **CallId** Параметр CallId для вызова, которому принадлежит соединение.
- **NotifiedEntity** Текущий "уведомленный объект" для конечной точки.
- **LocalConnectionOptions** Параметр LocalConnectionOptions, предоставленный для соединения.
- **Mode** Режим текущего соединения.
- **ConnectionParameters** Параметры текущего соединения для данного соединения.
- **LocalConnectionDescriptor** Параметр LocalConnectionDescriptor, который предоставлен шлюзом для соединения.
- **RemoteConnectionDescriptor** Самый последний параметр RemoteConnectionDescriptor, который был предоставлен предыдущей командой CreateConnection или ModifyConnection шлюзу для этого соединения.

Параметр **ReturnCode** – это параметр, возвращаемый шлюзом. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

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

7.3.9 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])
```

Параметр **EndpointId** определяет конечные точки, которые выводятся из эксплуатации или вводятся в эксплуатацию. Условный подстановочный знак "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 "любой из" НЕ ДОЛЖЕН использоваться. Контроллер медиашлюза, который получает команду RestartInProgress с условным подстановочным знаком "любой из" ДОЛЖЕН вернуть ответ с ошибкой (возвратной ошибке СЛЕДУЕТ присвоить код ошибки 500 – транзакция не может быть произведена, потому что конечная точка неизвестна).

Параметр RestartMethod определяет тип перезапуска:

- Метод "постепенного перезапуска" указывает на то, что определенная(ые) конечная(ые) точка(и) будет(ут) выведена(ы) из эксплуатации после заданной "задержки перезапуска". Установленные соединения еще не затронуты этим процессом, но контроллер медиашлюза должен воздержаться от установления новых соединений и должен попытаться постепенно разъединить все существующие соединения. По окончании задержки перезапуска, медиашлюзу следует послать новое сообщение RSIP, используя "принудительный" метод перезапуска. Для контроллера медиашлюза это будет однозначно указывать на то, что конечные точки выведены из эксплуатации.
- Метод "постепенного, с отменой" перезапуска указывает на то, что шлюз аннулирует ранее полученный метод "постепенного" перезапуска для одних и тех же конечных точек. Оставшиеся конечные точки находятся в эксплуатации. Когда передается эта команда, шлюз сразу же начинает пытаться установить новые соединения в этих конечных точках.
- Метод "принудительного" перезапуска указывает на то, что определенные конечные точки внезапно выведены из эксплуатации. Установленные соединения, если таковые имелись, будут потеряны.
- Метод "чистого перезапуска" указывает на то, что работа в конечных точках будет восстановлена после заданной "задержки перезапуска". Отсутствуют соединения, которые в текущее время устанавливаются в этих конечных точках.
- "Несвязный" метод указывает на то, что конечная точка стала несвязной (разъединенной) и теперь пытается восстановить связность. "Задержка перезапуска" определяет число секунд, в течение которых конечная точка была несвязной. Установленные соединения этим процессом не затрагиваются.

Необязательный параметр "задержка перезапуска" выражается числом секунд. Если это число отсутствует, то значение задержки должно считаться нулевым. В случае использования метода "постепенного" перезапуска нулевая задержка указывает на то, что в результате такого действия конечная точка никогда не выйдет из эксплуатации, а также на то, что контроллер медиашлюза 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 и поэтому параметр "задержка перезапуска" НЕ ДОЛЖЕН использоваться шлюзом в данном методе перезапуска. 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 шлюза, контроллер медиашлюза МОЖЕТ устранить доменное имя шлюза путем запроса системы DNS независимо от времени существования (Time To Live – TTL) текущей записи ресурса для перезапускаемого шлюза.

Транкинговым шлюзам СЛЕДУЕТ, в порядке вежливости, посылать контроллеру медиашлюза сообщение RestartInProgress для "постепенного" или "принудительного" перезапуска, когда они выводятся из эксплуатации, например, путем прекращения функционирования. Агент вызова не может рассчитывать на постоянный прием таких сообщений. Тем не менее, транкинговый шлюз ДОЛЖЕН посылать сообщение RestartInProgress для "принудительного" или "постепенного"

перезапуска, когда конечная точка выводится из эксплуатации через процесс инициализации, или путем обнаружения неисправности в устройстве переноса конечной точки (например, потеря сигнала, получение желтого сигнала тревоги и т. д.). Транкинговые шлюзы ДОЛЖНЫ посылать своему контроллеру медиашлюза сообщение RestartInProgress для "чистого перезапуска" с нулевой задержкой, когда они вводятся обратно в эксплуатацию согласно процедуре перезапуска, описанной в п. 7.4.3.5. – контроллеры медиашлюза могут рассчитывать на прием этого сообщения. Кроме того, транкинговые шлюзы ДОЛЖНЫ посылать сообщение RestartInProgress для "несвязного" метода перезапуска своему текущему "уведомленному объекту", согласно "несвязной" процедуре, описанной в п. 7.4.3.6. Параметр "задержка перезапуска" НЕ ДОЛЖЕН использоваться при методе "принудительного" перезапуска.

Сообщение RestartInProgress будет послано текущему "уведомленному объекту" для рассматриваемого параметра EndpointId. Ожидается, что контроллер медиашлюза по умолчанию, т. е. "уведомленный объект", был обеспечен для каждой конечной точки, так что после перезагрузки контроллером медиашлюза по умолчанию будет "уведомленный объект" для каждой конечной точки. Транкинговые шлюзы 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 контроллер медиашлюза.

Параметр **ReturnCode** – это параметр, возвращаемый агентом вызова. Он указывает на результат выполнения команды и состоит из целого числа (см. п. 7.5), которое необязательно сопровождается комментарием.

Параметр **NotifiedEntity** МОЖЕТ быть дополнительно возвращен с ответом на сообщение RestartInProgress от контроллера медиашлюза – это обычно должно делаться только в ответ на "чистый перезапуск" или "несвязный" перезапуск (см. также пп. 7.4.3.5 и 7.4.3.6). Если параметр NotifiedEntity был включен в возвратный ответ, то определяется новый "уведомленный объект" для конечной(ых) точки (точек) – такая операция делается только с кодом ответа ошибки 521 (переадресация конечной точки). Следует отметить, что указанное выше ситуация, когда в ответе возвращается параметр NotifiedEntity, определяется только для ответов команды RestartInProgress и НЕ МОЖЕТ иметь место для ответов других команд. Любые другие ситуация являются неопределенными:

- 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(ых) точки (точек).
- Если ответ от контроллера медиашлюза 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(ых) 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 контроллер медиашлюза возвращает код нерегулярной ошибки (4xx), в то время как при любом другом методе перезапуска СЛЕДУЕТ предпринимать повторную попытку. РЕКОМЕНДУЕТСЯ, чтобы любой тип перезапуска завершался, при возвращении кода постоянной ошибки (5xx), за исключением описанного выше кода 521.

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

7.4 Состояния, восстановление после отказа и состоятельные условия

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

7.4.1 Итоги и выводы

Как упоминалось в п. 7.1.4, контроллеры медиашлюза идентифицируются по их доменному имени, а каждая конечная точка имеет один и только один "уведомленный объект", связанный с нею в любой заданный момент времени. В этом пункте приводятся итоги и делаются выводы по тем областям, которые представляют особую важность для обеспечения надежности и восстановлению после отказа согласно протоколу MGCP:

- Контроллер медиашлюза 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 контроллер медиашлюза в любой заданный момент времени. Контроллер медиашлюза, 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 образом¹⁹, "уведомленный объект" устанавливается по умолчанию на адрес источника последней команды обработки соединения или запроса на уведомление, полученного для конечной точки. В этом случае, контроллер медиашлюза 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, контроллер медиашлюза может выдать им команду произвести коммутацию, предоставив им новый "уведомленный объект".
- Если контроллер медиашлюза становится недоступным, то конечные точки, управляемые этим контроллером медиашлюза, станут в конце концов "несвязными". Чтобы эти конечные точки вновь стали связными, либо недоступный контроллер медиашлюза снова должен стать доступным, либо другой (резервный) контроллер медиашлюза должен взаимодействовать с "затронутыми" конечными точками с новым "уведомленным объектом".
- Когда другой (резервный) контроллер медиашлюза взял на себя управление группой конечных точек, предполагается, что недоступный контроллер медиашлюза будет взаимодействовать и будет синхронизированным с резервным контроллером медиашлюза, чтобы передать управление группой "затронутых" конечных точек обратно исходному агенту вызова, если это необходимо. Как вариант, недоступный контроллер медиашлюза может просто стать теперь резервным контроллером медиашлюза.

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

7.4.2 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. 8.5) команды повторяются. Шлюзы ДОЛЖНЫ хранить в памяти список ответов, которые они послали последним транзакциям (список всех ответов, которые они послали за последние T_{hist} секунд), и список транзакций, выполняемых в текущее время. 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 ДОЛЖЕН послать временный ответ. Если это какая-либо другая команда, она просто игнорируется. В любом случае конечный ответ будет представлен, когда завершится выполнение команды.

¹⁹ 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 контроллера медиашлюза становится недоступным;
- отказа контроллера медиашлюза, когда, например, все интерфейсы для контроллера медиашлюза становятся недоступными;
- восстановление после отказа, когда новый контроллер медиашлюза "принимается" прозрачным образом.

Элементы должны быть способны получать оценку коэффициента потерь пакетов на основе предыстории. В правильно сконфигурированной системе, этот коэффициент потерь должен быть очень низким, обычно в среднем менее 1%. Если контроллеру медиашлюза 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 контроллера медиашлюза, 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 устанавливаться на значение, большее, чем Max 1.

На рисунке 2 иллюстрируется алгоритм повторной передачи протокола MGCP, а более подробные объяснения приведены ниже:

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:

- Шлюз ДОЛЖЕН всегда проверять наличие нового контроллера медиашлюза. Это может быть замечено благодаря:
 - приему команды, где параметр `NotifiedEntity` указывает на нового контроллера медиашлюза; или
 - приему ответа переадресации, указывающего на нового контроллера медиашлюза.
- Если обнаружен новый контроллер медиашлюза, то шлюз ДОЛЖЕН перенаправлять повторные передачи всех неподтвержденных команд для конечной(ых) точки (точек) этому новому контроллеру медиашлюза. 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 дейтаграммы, не превышало $T_{s_{max}}$. Если истекшее время больше $T_{s_{max}}$, то далее повторные передачи ДОЛЖНА прекратиться. Если истекшее время больше $2 * T_{t_{hist}}$, то конечная точка становится несвязной (разъединенной).
- Если число повторных передач этому контроллеру медиашлюза равно "Max1", то шлюз МОЖЕТ в активном режиме опросить сервер имен, чтобы обнаружить возможное изменение интерфейсов контроллера медиашлюза, независимо от времени существования (TTL), связанного с записью системы DNS.
- Шлюз может запомнить несколько IP-адресов для контроллера медиашлюза. 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. Также, получение явных сетевых уведомлений таких как, например, недостижимая сеть ICMP (протокол управляющих сообщений Интернет), хост, протокол или порт, шлюзу ДОЛЖНЫ указывать шлюзу на необходимость попытки использования альтернативного адреса (с должным вниманием к возможным сообщениям безопасности).
- Если для попыток передачи интерфейсов больше нет, а число повторных передач для данного адреса составляет "Max2", тогда шлюзу СЛЕДУЕТ установить связь с системой DNS

еще раз, чтобы узнать, не стали ли доступными какие-либо другие интерфейсы. Если все еще нет попыток передачи интерфейсов, тогда повторные передачи ДОЛЖНЫ быть прекращены. Если истекшее время больше $2 * T_{hist}$, то конечная точка становится несвязной (разъединенной).

- Когда конечная точка становится несвязной, последующая обработка зависит от того, были ли обнаружены шлюзом или контроллером медиашлюза потери в соединении:
 - Шлюз ДОЛЖЕН инициировать "несвязную" процедуру, как описано в п. 7.4.3.6.
 - Контроллер медиашлюза НЕ ДОЛЖЕН предпринимать попытку использовать конечную точку для любого нового вызова, пока не будет восстановлена возможность подключения. Более того, контроллер медиашлюза ДОЛЖЕН применить алгоритм для обнаружения момента, когда возможность подключения к конечной точке полностью восстановится (например, в момент получения ответа для повторяющейся команды Audit Endpoint). Когда восстановится возможность передачи к конечной точке, а также не будет существовать никаких ограничительных условий, ограждающих конечную точку от поддерживающего вызова, контроллер медиашлюза ДОЛЖЕН убедиться, что конечная точка может использоваться для новых вызовов и не нуждается во вмешательстве с указаниями.

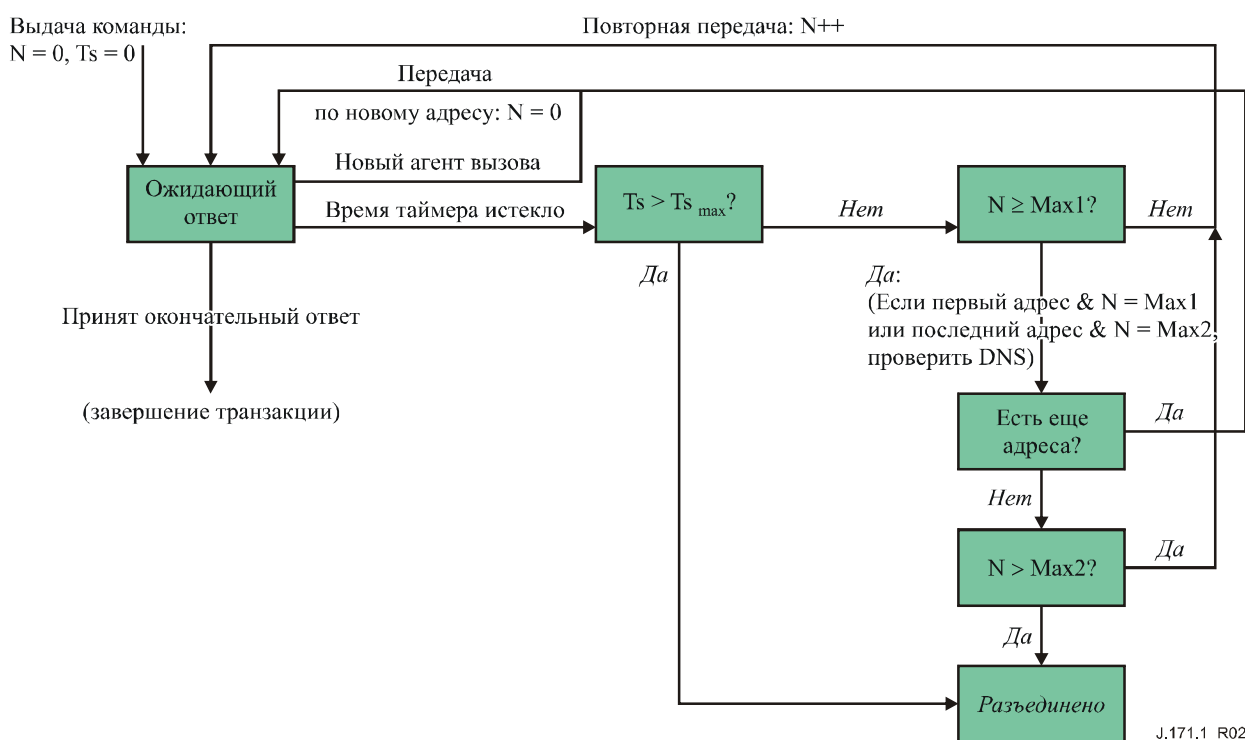


Рисунок 2/J.171.1 – Алгоритм повторной передачи

Для автоматической адаптации к сетевой нагрузке в протоколе MGCP определены таймеры с экспоненциально возрастающим временем (см. п. 8.5.2). 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.

Однако является важным ограничивать максимальное время задержки повторной передачи. В первую очередь для любой повторной передачи проводится проверка не превысило ли время (T_s), прошедшее с момента отправления инициирующей дейтаграммы, значения времени $T_{s_{max}}$. Если прошло больше времени, чем $T_{s_{max}}$, повторная передача ДОЛЖНА прерываться. По истечении времени $T_{s_{max}}$ или если все повторные передачи на все известные IP-адреса были отосланы, то перед объявлением конечной точки несвязной вводится пауза. Эта пауза представляет собой период времени, единственным действием в течение которого является ожидание ответа от любой из совершенных в последнее время повторных передач. Период молчания длится на протяжении удвоенной продолжительности

существования ($2 \cdot T_{t_{hist}}$). Такое время стабилизации позволяет завершить или провести выдержку времени всех активных транзакций до того, как конечная точка будет объявлена несвязной. Это помогает удостовериться, что каждая перезагрузка конечной точки осуществляется с чистого и исходного состояния. Если прошло больше времени, чем $2 \cdot T_{t_{hist}}$, конечная точка становится несвязной. Значение $T_{s_{max}}$ взаимосвязано со значением $T_{t_{hist}}$: значение $T_{t_{hist}}$ ДОЛЖНО быть больше или равно, чем $T_{s_{max}}$ плюс максимальная задержка распространения по сети $T_{p_{max}}$.

Следует отметить, что между $T_{s_{max}}$, $T_{t_{hist}}$ и максимальным транзитным временем $T_{p_{max}}$ существует тесная связь.

Другими словами, 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".

7.4.3 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 занятости 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 контроллера медиашлюза 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.

7.4.3.1 Карантинный список

Шлюзы, управляемые по протоколу MGCP, будут получать запросы на уведомление, согласно которым от них требуется следить за списком событий. Элементами протокола, определяющими обработку этих событий, являются список "запрашиваемых событий" и список "обнаруженных событий".

Когда инициализируется конечная точка, список запрашиваемых событий состоит только из устойчивых событий для конечной точки. После получения команды, 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. События с определениями "накопление" накапливаются в списке наблюдаемых событий. 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 на команду Notify²⁰. Как описано в п. 7.4.2, команда Notify может быть повторно передана в "состоянии уведомления".

Если при этом конечная точка является или становится несвязной (см. п. 7.4.2), 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. 7.4.3.6, 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 при инициировании команды NotificationRequest:

Если параметр QuarantineHandling режима "шаг" был задан агентом вызова/контроллером медиашлюза или если вообще не было задано никакого значения, тогда шлюз будет использовать режим "шаг блокировки", который означает, что шлюз ДОЛЖЕН получить новую команду NotificationRequest, после того, как отправит команду Notify. 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".

Если параметр QuarantineHandling режима "цикл" был задан агентом вызова/контроллером медиашлюза (т. е. режим "цикл"), он будет действовать следующим образом. Когда шлюз выходит из "состояния уведомления", он сбрасывает список наблюдаемых событий конечной точки до нулевого значения и начинает обработку списка подвергнутых карантину событий, используя уже полученный список запрашиваемых событий. 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. События, которые следуют за последним инициирующим событием, ДОЛЖНЫ оставаться в карантинном буфере.

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

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

²⁰ Следует отметить, что команда Notify не может быть объединена со встроенной командой NotificationRequest.

"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/контроллер медиашлюза 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(ы) Notify для данной конечной точки ДОЛЖНА(Ы) 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".

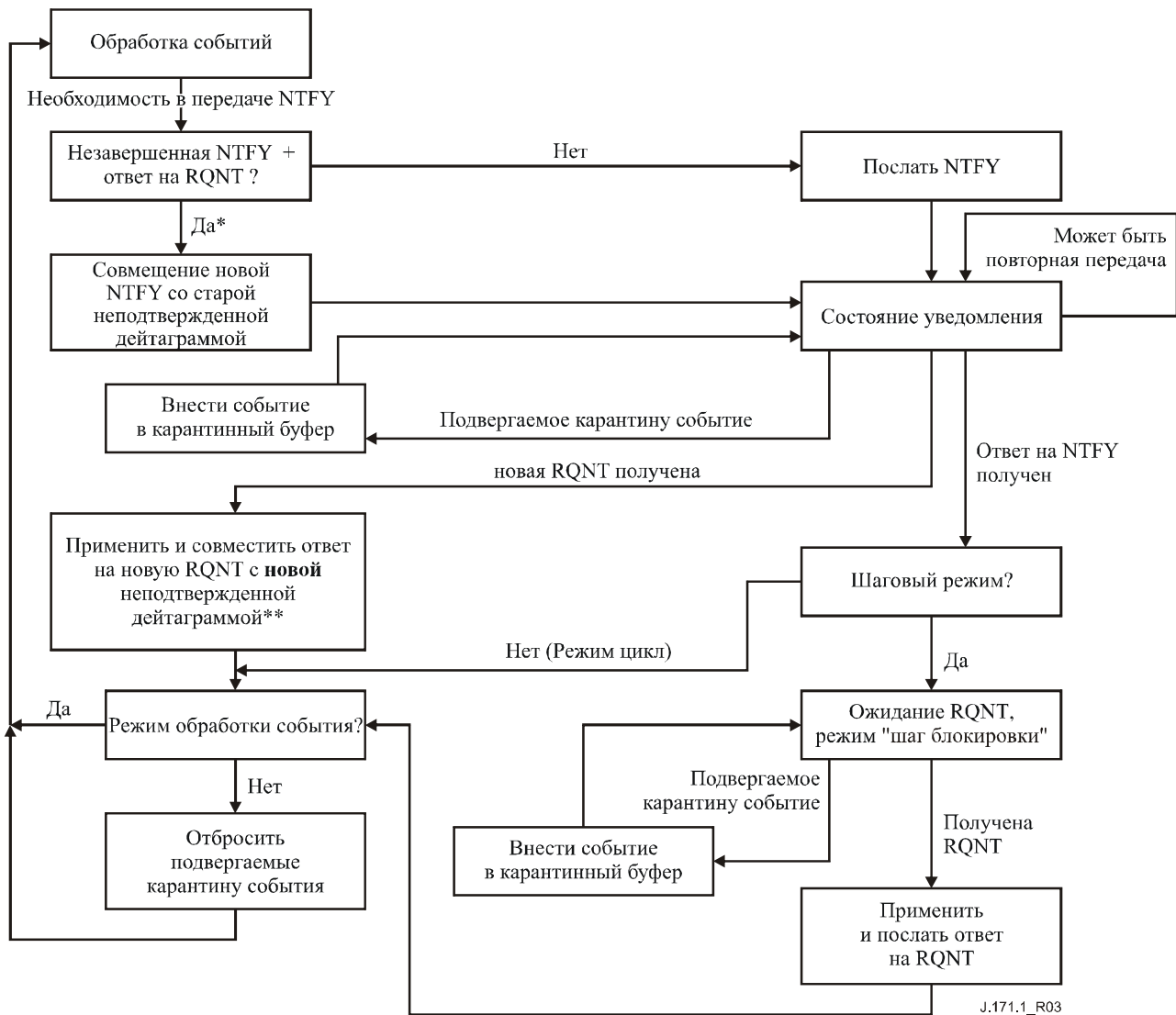
После получения команды NotificationRequest параметры "список запрашиваемых событий" заменяются вновь полученными параметрами. Кроме того, когда в "состоянии уведомления" была получена новая команда NotificationRequest, то список "наблюдаемых событий" сбрасывается до нулевого значения. Последующее поведение 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". 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 встроенного запроса на уведомление. В этом случае, список наблюдаемых событий остается немодифицированным, а не обрабатывается вновь.

Приведенная выше процедура применяется ко всем видам запросов на уведомление независимо от того, являются ли они частью команды обработки соединения или представлены 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 3 иллюстрируются определенные выше процедуры в предположении успешного выполнения всех транзакций:



- * Данная ветвь принятия решения занимается шлюзом, если ему нужно отправить новое уведомление, в то время, когда он ожидает ответа на предыдущее уведомление, в той же самой конечной точке. Это может возникнуть как результат получения нового RQNT в "состоянии уведомления", как описано в тексте, сопровождающем диаграмму.
- ** "Новая невыполненная дейтаграмма" относится к дейтаграмме, содержащей ожидающее обработки уведомление, возможно совмещенное с дополнительным(и) уведомлением(ями) и ответом(ами) RQNT, переданными в состоянии уведомления, если был получен новый RQNT. Должная доставка ответа RQNT с ожидающим обработкой уведомлением(ями) выполняется при желании. Вместо этого, шлюз может выбрать отправку ответа RQNT в отдельной дейтаграмме. С другой стороны, требование об удостоверении должной доставки уведомления является обязательным.

Рисунок 3/J.171.1 – Алгоритм для карантина

Контроллерам медиашлюза СЛЕДУЕТ представить ответ на успешное сообщение Notify и новую команду NotificationRequest в одной и той же дейтаграмме, используя механизм совмещения запросов и ответов²¹.

7.4.3.2 Явное обнаружение

Ключевым элементом состояния некоторых конечных точек является состояние занятости линии связи. Хотя события изменения состояния занятости являются устойчивыми в протоколе TGCP, состязательные условия и несовпадение состояний все же могут иметь место, например, когда линия связи занята, в то время как контроллер медиашлюза находится в процессе запроса шлюза относительно поиска событий "seizure" (состояние "помехи приему вызова звонком", хорошо известное в телефонной связи – однако это является в основном проблемой для двусторонних линий связи CAS, которые не поддерживаются в этой версии данной Рекомендации).

²¹ Поставщики, которые предпочитают не следовать положениям настоящей Рекомендации, должны тщательно изучить сценарии отказов контроллера медиашлюза.

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 "занято"²², в то время как линия связи уже занята (код ошибки 401 – линия связи занята);
- 2) если шлюз получил запрос на уведомление о состоянии "незанято"²³, в то время как линия связи является не занятой (код ошибки 402 – линия связи не занята).

Кроме того, в отдельных определениях сигналов может быть определено, что сигнал будет работать только в определенных условиях, например, оператор посылки вызова СЧ возможен, если только линия связи уже занята. 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 контроллером медиашлюза 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.

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

Другое состязательное условие может возникнуть, когда команда Notify выдается непосредственно перед приемом команды NotificationRequest шлюзом. Параметр RequestIdentifier используется для коррелирования команд Notify с командами NotificationRequest (включает запрос на уведомление, встроенный в примитивы обработки соединения), тем самым позволяя контроллеру медиашлюза определять, была ли команда Notify генерирована до или после того, как шлюз принял новую команду NotificationRequest.

7.4.3.3 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 "занятости", должна быть определена тщательным образом.

²² Для примера, запрос события "sup" (к) завершающейся транкинговой линии BLV/OI MF без какого-либо имеющего места.

²³ Для примера, запрос события "rel" (к) оператору транкинговой службы MF без какого-либо имеющего места.

Важным моментом является необходимость учитывать тот факт, что состояние "seizure-state" 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 подверженные карантину события.

В качестве примера можно рассмотреть транзакцию, которая включает запрос на событие "занято". Когда транзакция иницируется, линия связи находится в состоянии "не занято", и поэтому данное предварительное условие соблюдается. Если состояние занятости изменяется на состояние "занято" до завершения выполнения транзакции, предварительное условие больше не соблюдается, и поэтому транзакция немедленно получает отказ. Событие "занятости" будет теперь записано в память "карантинного" буфера, который затем начинает обрабатываться.

7.4.3.4 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. Например:

- команды Notify могут запаздывать и поступать к контроллеру медиашлюза 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.

Контроллер медиашлюза 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 иницирующей 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. Контроллер медиашлюза 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. 7.4.3.5), 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.

Транкинговые шлюзы ДОЛЖНЫ придерживаться тех из упомянутых выше правил, которые определяют поведение шлюза; однако транкинговый шлюз НЕ ДОЛЖЕН делать никаких предположений относительно того, следуют ли контроллеры медиашлюза правилам или нет. Следовательно, шлюзы ДОЛЖНЫ всегда отвечать на команды независимо от того, придерживаются ли они упомянутых выше правил или нет.

Чтобы обеспечить согласованность операции, транкинговые шлюзы ДОЛЖНЫ вести себя как указано ниже, если не выполняется одно или несколько вышеперечисленных правил:

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

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

7.4.3.5 Противодействие лавине перезапусков

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

- 1) Когда включается питание шлюза, или когда все конечные точки шлюза или подмножество конечных точек шлюза вводятся в эксплуатацию, шлюз инициирует таймер перезапуска, присваивая ему случайное значение, равномерно распределенное между 0 и максимальным обеспечиваемым значением задержки на ожидание (maximum waiting delay – 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 контроллера медиашлюза, либо обнаружения действий локальной линии связи, таких, например, как переход в состояние "занятости" в транкинговом шлюзе. Уже существующее состояние "занятости" приводит к генерированию события "занятости".
- 3) Когда истекает время таймера перезапуска, принимается команда или когда обнаруживаются уже существующее состояние занятости, шлюз инициирует процедуру перезапуска.

Процедура перезапуска просто устанавливает, что конечная точка ДОЛЖНА послать контроллеру медиашлюза команду RestartInProgress, информируя его о перезапуске, и, кроме того, гарантировать, что первое сообщение (команда или ответ), которое видит контроллер медиашлюза 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. Во время каждой инициации "несвязной" процедуры, в команде ДОЛЖНЫ наблюдаться обычная повторная передача и требования идентификаторов транзакции (см. п. 7.4.2).

Для достижения этого конечная точка ДОЛЖНА в полной мере использовать механизм совмещения передачи запросов и ответов. Например, если действие занятия линии связи имеет место до истечения времени таймера перезапуска, то будет генерирован пакет, содержащий команду RestartInProgress, вместе с вложенной (совмещенной в этом пакете) командой Notify для события "занятости". В случае, когда время таймера перезапуска истекло при отсутствии всех прочих действий, шлюз просто посылает сообщение RestartInProgress.

Процедура перезапуска завершается, когда получен успешный ответ. Если получен ответ об ошибке, то последующее поведение зависит от рассматриваемого кода ошибки:

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

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.

Если шлюз входит в "несвязное" состояние при выполнении процедуры перезапуска, то ДОЛЖНА быть выполнена несвязная процедура, описанная в п. 7.4.3.6, и во время этой процедуры посылается сообщение".

Ожидается, что каждая конечная точка в шлюзе будет иметь обеспечиваемого контроллера медиашлюза, т. е. "уведомленный объект", для направления к нему начального сообщения перезапуска. Когда управление множеством конечных точек в шлюзе осуществляется несколькими контроллерами медиашлюза, то приведенная выше процедура должна быть выполнена для каждого множества конечных точек, управляемых данным контроллером медиашлюза. 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 контроллер медиашлюза.

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

Контроллеры медиашлюза обычно рассчитаны на обработку трафика в часы наибольшей нагрузки, во время которых, в среднем, будет занято 60% линий, обслуживающих вызовы, средняя продолжительность которых составляет обычно 3 минуты. Обработка вызова обычно включает 5–6 транзакций между каждой конечной точкой и контроллером медиашлюза. Это простое вычисление показывает, что контроллер медиашлюза, 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 шлюза может быть значение 2 минуты на конечную точку. Когда значение задержки MWD устанавливается для шлюза, то такое значение должно быть обратно пропорционально числу конечных точек, участвующих в процессе перезапуска. Например, значение задержки MWD следует установить равным 5 секундам для шлюза, который обрабатывает линию T1, или равным 180 миллисекундам для шлюза, который обрабатывает линию T3.

7.4.3.6 Несвязные конечные точки

В дополнение к процедуре перезапуска, Гранкинговые шлюзы также имеют "несвязную" процедуру, которая иницируется, когда конечная точка оказывается "несвязной" ("разъединенной"), как описано в п. 7.4.2. Здесь следует отметить, что конечные точки могут стать несвязными только в том случае, когда они пытаются взаимодействовать с контроллером медиашлюза. Конечная точка, которая становится "несвязной", выполняет следующие шаги:

- 1) "Несвязный" таймер устанавливается на случайное значение, равномерно распределенное между нулем и обеспечиваемой начальной задержкой на ожидание для "несвязного" состояния ($T_{d_{init}}$),

равной, например, 15 секундам. ДОЛЖНА быть проявлена некоторая осторожность, чтобы избежать синхронности при генерировании случайных чисел между множеством шлюзов и конечными точками, которые будут использовать один и тот же алгоритм.

- 2) Шлюз, затем, ждет, либо окончания времени таймера, либо приема команды от контроллера медиашлюза, либо обнаружения действий локальной линии связи для конечной точки, таких, например, как переход в состояние "занятости".
- 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 локальной линии связи с момента, когда шлюз стал разъединенным ("несвязным"), или с момента, когда он в последний раз закончил "несвязную процедуру", должно, кроме того, истечь время обеспечиваемой минимальной задержки на ожидание для "несвязного" состояния ($T_{d_{min}}$), чтобы ограничить скорость выполнения процедуры.
- 4) Если "несвязная процедура" все еще оставляет конечную точку в "несвязном" состоянии, тогда значение для "несвязного" таймера удваивается, по отношению к обеспечиваемой максимальной задержки на ожидание ($T_{d_{max}}$), например, 600 секунд, и снова выполняет действия, начиная с п. 2).

"Несвязная" процедура аналогична процедуре перезапуска в том, что она просто устанавливает, что конечная точка ДОЛЖНА послать контроллеру медиашлюза команду RestartInProgress, информируя его о том, что конечная точка была разъединена, и, кроме того, гарантировать, что первое сообщение (команда или ответ), которое теперь видит контроллер медиашлюза 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. 7.4.2). Для этого в конечной точке ДОЛЖНО полностью использоваться совмещение передачи запросов и ответов.

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

Контроллер медиашлюза может также решить выполнить одно или несколько из следующих действий: контроль конечной точки, сброс всех соединений конечной точки или передача команды NotificationRequest, запрашивающей конечную точку об обработке сообщений, помещенных в карантин (см. п. 7.4.3.7).

Несвязная конечная точка может пожелать послать команду (помимо команды RestartInProgress), находясь в разъединенном состоянии. Это действие окажется успешным, как только контроллер медиашлюза 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 контроллер медиашлюза был действительно доступен, но конечная точка еще не завершила "несвязную" процедуру (можно рассмотреть, например, случай, когда команда NotificationRequest была только что получена, и это сразу же привело к генерированию команды Notify). Для предупреждения таких сценариев разъединенные (несвязные) конечные точки НЕ ДОЛЖНЫ "вслепую" отбрасывать новые команды, подлежащие передаче за период в $T_{s_{max}}$ секунд, после того как они получают команду "не 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 контроллер медиашлюза, 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 инициируется "несвязная" процедура, конечная точка МОЖЕТ совместить передачу буферизированной(ых) 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. 7.4.2.

"Несвязная" процедура завершается, как только получен успешный ответ. Ответы об ошибках обрабатываются подобно процедуре перезапуска (см. п. 7.4.3.5). Если после передачи ответа об ошибке должна быть вновь инициирована "несвязная" процедура, то применяются описанные выше положения, касающиеся таймера, ограничивающего скорость перезапуска.

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.

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 контроллера медиашлюза.

Следует отметить, что "несвязное" состояние конечной точки не означает, что конечная точка находится в состоянии "выведена из эксплуатации". "Несвязное" состояние не является состоянием готовности услуг конечной точки, а скорее – указанием на неспособность шлюза связываться с его MGC.

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.

7.4.3.7 Обработка агентом вызова несвязных конечных точек

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

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

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

- 1) Определим новую булеву переменную, названную "disconnect-event-sync" (синхронизация несвязных событий), содержащуюся в агенте вызова для каждой его конечной точки. При установке в состояние "истина" эта переменная показывает, что связность с "несвязной" конечной точкой недавно была восстановлена, но синхронизация событий/сигналов еще не достигнута. (Следует отметить, данная переменная представлена для описания поведения агента вызова и не предназначена для применения какой-либо конкретной реализации. Изначально переменная невидима.)

- a) Как только агент вызова получает информацию, что конечная точка несвязна, он устанавливает "disconnect-event-sync" на значение истина. "Несвязная" процедура удостоверяется, что агент вызова получит информацию о том, что конечная точка становится несвязной, через получение сообщения "disconnected" команды RestartInProgress. Если конечная точка получает положительное уведомление к "несвязному" RSIP, она выполняет "несвязную" процедуру. В этот момент конечная точка могла бы немедленно сгенерировать команду Notify по двум причинам: чтобы послать команду Notify, которая была помещена в буфер в то время, когда конечная точка была отсоединена, или если, будучи отсоединена, конечная точка была в состоянии "уведомления" или режиме "цикл", то для того, чтобы сообщить событие, инициирующее уведомление, в карантинный список:
- Если конечная точка работает в "шаговом" режиме, то ответ на команду Notify не вызовет в ней генерирование каких-либо дальнейших сообщений Notify (для этого понадобится дополнительная команда NotificationRequest).
 - Однако если конечная точка работает в режиме "цикла", то ответ на команду Notify вызовет генерирование дальнейших сообщений Notify. Как обсуждалось выше, иногда это нежелательно, поскольку события, о которых сообщается, могут быть устаревшими, а множество событий может быть помещено в карантин, что в свою очередь, вызовет большое количество сообщений Notify и последующих сообщений NotificationRequest, базирующихся на устаревшей информации.
- b) До тех пор, пока "disconnect-event-sync" конечной точки установлен на значение истина, агенту вызова следует предпринять действие, чтобы удостовериться, что потенциально большое число событий в карантинном списке либо отклонено, либо обрабатывается контролируемым и надлежащим образом. Это может быть достигнуто различными методами:
- Агент вызова может послать одиночную команду NotificationRequest, указывающую, что все подлежащие карантину события должны быть отклонены. При получении положительного уведомления на эту команду или при получении команды Notify с тем же самым RequestIdentifier, агенту вызова следует установить индикатор "disconnect-event-sync" на значение ложь, в этот момент проводится регулярная обработка событий для конечной точки. Обратной стороной данного подхода является то, что он сбрасывает любые накопленные события, несмотря на число накопленных событий. В некоторых случаях это может привести к необходимости прерывания услуги. Для адресации прерывания будут необходимы расширения протокола.
 - Агент вызова может послать команду NotificationRequest, указывающую, что помещенные в карантин события должны быть обработаны. Если конечная точка работает в режиме "шаг", то она сообщит о каждом единичном событии, инициирующем уведомление, с помощью одной полученной команды NotificationRequest, в то время как режим "цикл", может сообщить о нескольких событиях, с помощью одной команды NotificationRequest.
- Поскольку информация, переданная событиями уведомления может больше не иметь никакого значения, агенту вызова не следует "вслепую" обрабатывать такие события (например, будучи уведомленным о начинающемся событии "MT/sup", агенту вызова не следует автоматически посылать команду NotificationRequest, чтобы запросить уведомление о событии цифр "MT/inf"). Вместо этого агент вызова должен синхронизировать свои внутреннее состояние данных с актуальным состоянием конечной точки. Поскольку конечная точка предоставляет обработку помехи приему вызова сигналом/событием против текущего состояния линии при переключении телефонной трубки, агент вызова может обнаружить текущее состояние занятости, базируясь на ответе на команду NotificationRequest. Например, ответ "402 – линия связи не занята" на команду NotificationRequest, запрашивающей об обнаружении события "MT/inf", означает, что линия MT в настоящий момент не занята. Агент вызова может сделать выбор игнорировать уведомленные события, которые становятся незначимыми, базируясь на текущем состоянии телефонной трубки (т. е. игнорирование события цифр "MT/inf", если линия, например, не занята).
- Как только все элементы из карантинного списка были обработаны, агенту вызова следует установить индикатор "disconnect-event-sync" на значение ложь. Агент вызова может с уверенностью предположить, что все, помещенные в карантин

события, были выполнены, если истекла задержка длительности T_{hist} , с того момента, как агент вызова в последний раз указывал конечной точке обрабатывать следующее сообщение (т. е. если задержка длительности T_{hist} истекла с того момента, как агент вызова послал ответ на предыдущую команду Notify в режиме "цикл", или с того момента, как агент вызова получил последний положительный ответ на команду NotificationRequest в режиме "шаг").

- 2) Если конечная точка становится несвязной, соединения, созданные на базе этой конечной точки не должны быть затронуты. Однако всегда есть возможность, что соединение больше не может поддерживаться конечной точкой и, следовательно, отключается; это выразится в том, что агенту вызова будет послана команда DeleteConnection. Когда конечная точка отсоединена, такая команда может никогда не получиться агентом вызова, в этом случае он не узнает, что соединение удалено. Следовательно, когда бы агент вызова не выяснил, что конечная точка не соединена, ему следует проверить конечную точку по всему списку соединений, существующих на ней.

7.5 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 ответа²⁴;
- 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 3 перечислены значения, которые были определены.

Таблица 3/J.171.1 – Коды 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(я) было(и) исключено(ы).
400	Транзакция не могла быть выполнена из-за нерегулярной ошибки.
401	Телефон уже находится в состоянии "телефонная трубка снята" или линия связи занята.
402	Телефон уже находится в состоянии "телефонная трубка положена" или линия связи не занята.
407	Транзакция отменена. Транзакция была отменена каким-то внешним действием, например, командой ModifyConnection, отмененной командой DeleteConnection.
500	Транзакция не могла быть выполнена, т. к. конечная точка неизвестна.
501	Транзакция не могла быть выполнена, т. к. конечная точка к этому не готова.
502	Транзакция не могла быть выполнена, т. к. конечная точка не имеет достаточных ресурсов.
503	Подстановочный знак "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.

²⁴ Подтверждение ответа используется для временных ответов (см. п. 8.8).

Таблица 3/J.171.1 – Коды завершения

Код	Значение
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).
510	Транзакция не могла быть выполнена, т. к. была обнаружена ошибка протокола.
511	Транзакция не могла быть выполнена из-за команды, содержащейся в нераспознанном расширении.
512	Транзакция не могла быть выполнена, т. к. шлюз не оборудован для обнаружения одного из запрошенных событий.
513	Транзакция не могла быть выполнена, т. к. шлюз не оборудован для генерирования одного из запрошенных сигналов.
514	Транзакция не могла быть выполнена, т. к. 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, т. к. конечная точка "перезапускается".
521	Конечная точка переадресована другому контроллеру медиашлюза.
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(ые) значение(я) в параметре LocalConnectionOptions.
533	Ответ слишком велик.
534	Отказ при согласовании кодеков.
538	Ошибка параметра событие/сигнал (например, пропущенный, ошибочный, неподдерживаемый, неизвестный параметр и т. д.).

7.6 Коды причины

Коды причины используются шлюзом при исключении соединения для информирования контроллера медиашлюзам о причине исключения соединения. Код причины – это целое число. В таблице 4 перечислены определенные значения кода причины:

Таблица 4/J.171.1 – Коды причины

Код	Значение
900	Неправильное функционирование конечной точки.
901	Конечная точка выведена из эксплуатации.
902	Потеря связности нижнего уровня (например, синхронизации в нисходящем направлении передачи).

7.7 Использование параметров LocalConnectionOptions и ConnectionDescriptors

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

- 1) Контроллер медиашлюза/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) Контроллер медиашлюза/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) Контроллер медиашлюза/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 контроллер медиашлюза/агент вызова выдает команду CreateConnection или ModifyConnection, имеются три параметра, которые определяют мультимедийные пакеты, поддерживаемые этим соединением:

- Параметр LocalConnectionOptions: предоставляется контроллером медиашлюза/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 контроллером медиашлюза/агентом вызова 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 контроллеру медиашлюза/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. В дополнение к мультимедийным параметрам пакетов, приписанных к соединению, шлюз может также уведомлять о дополнительных поддерживаемых возможностях в LocalConnectionDescriptor. Следует обратить внимание, что такие возможности ДОЛЖНЫ быть обеспечены за пределами линии "m=" в SDP. Шлюз свободен для уведомления о всех своих поддерживаемых возможностях в независимости от полученных от агента вызова параметров LCO или RCD, и в независимости от мультимедийных параметров пакетов, обусловленных соединением.

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

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

- 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, какой(ие) кодек(и) и период(ы) 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(и) заданным(и) периодом(ами) пакетирования. Аналогичным образом, если параметр LocalConnectionOptions был предоставлен, но период(ы) пакетирования был(и) опущен(ы), 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 медиашлюзу 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(ы) пакетирования был(и) опущен(ы), 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, медиашлюз 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 период пакетирования для T.38 выбирается так же, как описано выше с использованием тех же процедур, что и для аудио кодека.

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

- 1) Шлюз поддерживает несколько кодеков и может осуществлять переключение между различными кодеками в режиме реального времени. Шлюз возвращает все несогласованные кодеки в строке потока мультимедийных пакетов SDP. Несколько кодеков в строке = m означает, что устройство должно быть готово принять мультимедийные пакеты от любого из согласованных кодеков. В дополнении к этому, 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 режиме реального времени. Поэтому шлюз согласует и возвращает только один кодек в строке потока мультимедийных пакетов SDP (необязательно шлюз также помещает дополнительные поддерживаемые кодеки в атрибуте 'X-rc-codecs' протокола SDP). Согласно этому методу, изменение кодека должно инициироваться контроллером медиашлюза, чтобы выполнить изменение кодеков.

7.7.1 Согласование RFC 2833

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

Примером авторизованной через LCO передачи DTMF RFC 2833 является:

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

Кодек телефонных событий НЕ ДОЛЖЕН быть единственным кодеком из MGC, применяемых в LCO. Если адаптер МТА получает LCO, содержащий только кодек телефонных событий, он ДОЛЖЕН вернуть код ошибки 524 – внутренняя несогласованность в параметре LocalConnectionOptions. Если расширенный как описано в п. 7.7, список кодеков, содержит только кодек телефонных сообщений, то конечная точка ДОЛЖНА вернуть код ошибки 534 – отказ согласования кодеков. Подобным образом, если согласованный, как описано в п. 6.7, список кодеков содержит только кодек телефонных событий, конечная точка ДОЛЖНА возвращать код ошибки 534.

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

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

```
v=0  
o=- 4723891 7428910 IN IP4 128.96.63.25  
s=-  
c=IN IP4 128.96.63.25  
t=0 0
```

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

За более подробной информацией по использованию DTMF набора RFC 2833 обращаться к Спецификации J.161 Аудио/Видео кодеки проекта IP-Cablecom.

7.7.2 Согласование удаленных IP и порта

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

8 Протокол управления медиашлюза

В протоколе MGCP интерфейс управления шлюзом медиа 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 контроллер медиашлюза посылает шлюзу. Команда Notify посылается шлюзом контроллеру медиашлюза. Шлюз может также посылать команду DeleteConnection, как определено в п. 7.3.6. Контроллер медиашлюза может посылать шлюзу любую из команд Audit и, наконец, шлюз может послать контроллеру медиашлюза команду RestartInProgress.

8.1 Общее описание

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

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

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

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

8.2 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.

8.2.1 Строка команды

Строка команды состоит из:

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

Эти четыре элемента кодируются в виде строк печатных знаков в коде ASCII, разделенных пробелами, т. е. знаками пробела (0x20) или знаками табуляции (0x09) в коде ASCII. Для транкинговых шлюзов СЛЕДУЕТ использовать один разделитель пробелами в коде ASCII; однако транкинговые шлюзы ДОЛЖНЫ быть способны проводить синтаксический анализ сообщений с дополнительными символами пробелов.

8.2.1.1 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 5:

Таблица 5/J.171.1 – Кодирование запрашиваемых команд

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

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

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

8.2.1.2 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 транкинговому шлюзу, ДОЛЖНЫ быть однозначно определены для максимального времени существования транзакций в пределах множества контроллеров медиашлюза, которые управляют этим транкинговым шлюзом (см. п. 8.5). Таким образом, независимо от передающего контроллера медиашлюза, транкинговые шлюзы всегда могут обнаружить дублированные транзакции путем простой проверки идентификатора транзакции. Вопросы координации этих идентификаторов транзакций между контроллерами медиашлюза выходят за рамки настоящей Рекомендации.

Идентификаторы транзакций для всех команд, посылаемых от данного транкингового шлюза, ДОЛЖНЫ быть однозначно определены для максимального времени жизни транзакций (см. п. 8.5) независимо от того, какому контроллеру медиашлюза была послана команда. Таким образом, контроллер медиашлюза всегда может обнаружить дублированную транзакцию от транкингового шлюза по комбинации доменного имени конечной точки и идентификатора транзакции. В свою очередь, шлюз 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.

8.2.1.3 Кодирование имен конечных точек, контроллеров медиашлюза и уведомленных объектов

Имена конечных точек и имена контроллера медиашлюза кодируются в виде адресов электронной почты, как определено в стандарте IETF RFC 2821. 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.

Примерами таких имен являются:

ds/ds1-3/2@TGCP2.whatever.net	Вторая линия связи на третьем сигнале DS1 в транкинговом шлюзе протокола TGCP2 в сети "whatever" (любая).
MGC@mgc.whatever.net	Котроллер медиашлюза для сети "любая".

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

MGC@mgc.whatever.net:5234

В случае, когда номер порта опущен, по умолчанию будет использоваться порт по протоколу MGCP (2727, если только не обеспечено иное). В п. 7.1.1 содержится дополнительная подробная информация, касающаяся имен конечных точек.

8.2.1.4 Кодирование версии протокола

Версия протокола кодируется в виде ключевого слова "MGCP", за которым следуют пробел и номер версии, который сопровождается именем профиля "TGCP" 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 TGCP 1.0

Часть "TGCP 1.0" указывает на то, что это профиль TGCP 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).

8.2.2 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 6:

Таблица 6/J.171.1 – Параметры команды

Имя параметра	Код	Значения параметра
ResponseAck (Note)	K	См. описание.
CallId	C	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака.
ConnectionId	I	Шестнадцатеричная строка; НЕ ДОЛЖНА превышать 32 знака.
NotifiedEntity	N	Идентификатор в формате кода IETF RFC 821, 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.
ConnectionMode	M	См. описание.
RequestedEvents	R	См. описание.
SignalRequests	S	См. описание.
ObservedEvents	O	См. описание.
ConnectionParameters	P	См. описание.
ReasonCode	E	См. описание.
SpecificEndPointId	Z	Идентификатор в формате кода IETF RFC 821, состоящий из произвольной строки, необязательно сопровождаемой знаком "@", за которым следует доменное имя транкингового шлюза, 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.
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.
ПРИМЕЧАНИЕ. – Параметр ResponseAsk не был рассмотрен в п. 7.3, 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 7 показана связь между параметрами и командами. Буква М означает "обязательный", буква О – "необязательный", а буква F – "запрещенный".

Таблица 7/J.171.1 – Связь параметра с запросом команды

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
ResponseAck (Note)	O	O	O	O	O	O	O	O
CallId	M	M	O	F	F	F	F	F
ConnectionId	F	M	O	F	F	F	M	F
RequestIdentifier	O	O	O	M	M	F	F	F
LocalConnectionOptions	O	O	F	F	F	F	F	F
ConnectionMode	M	O	F	F	F	F	F	F
RequestedEvents	O ^{a)}	O ^{a)}	O ^{a)}	O ^{a)}	F	F	F	F
SignalRequests	O ^{a)}	O ^{a)}	O ^{a)}	O ^{a)}	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
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
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
<p>^{a)} Параметры RequestedEvents и SignalRequests являются необязательным и 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.</p> <p>ПРИМЕЧАНИЕ. – Параметр ResponseAsk не был рассмотрен в п. 7.3, поскольку идентификаторы транзакций не видны в приведенном примере интерфейса API. Те, кто реализует систему, могут выбрать другой подход.</p>								

Транкинговым шлюзам и контроллерам медиашлюза СЛЕДУЕТ всегда представлять обязательные параметры перед необязательным и; однако транкинговые шлюзы 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.

8.2.2.1 Подтверждение ответа

Параметр `ResponseAck` используется для поддержки установления трехсторонней связи, описанного в п. 8.7. Он содержит разделенный запятыми список "диапазонов подтвержденных идентификаторов транзакций".

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

Примером подтверждения ответа является:

K: 6234-6255, 6257, 19030-19044

8.2.2.2 Параметр `RequestIdentifier`

Параметр `RequestIdentifier` коррелирует команду `Notify` с иницирующей ее командой `NotificationRequest` (включает встроенный запрос на уведомление в примитиве управления соединением). 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. 7.3.2).

8.2.2.3 Параметр `LocalConnectionOptions`

`LocalConnectionOptions` (варианты локальных соединений) описывают необязательные параметры, которые шлюз по команде контроллеров медиашлюза 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.
- Буквенное наименование алгоритма сжатия, как определено в Рек. МСЭ-Т J.161, кодируется в виде ключевого слова "a", за которым следуют двоеточие и строка знаков. Если контроллер медиашлюза (MGC) указывает список значений, эти значения должны быть разделены точкой с запятой. Для RTP аудиокодеки ДОЛЖНЫ указываться с использованием имен кодирования, определенных в профиле RTP AV RFC 1890, имен кодирования, зарегистрированных в IANA, или имен кодирования, на которые ссылается или которые определяется в Спецификации Аудио/Видео кодеки проекта IP-Cablecom. В среде не-аудио, зарегистрированной как тип MIME (многоцелевые расширения почтовой службы в интернете/стандарт MIME), ДОЛЖНЫ использоваться формы "<тип MIME>/<подтип MIME>", как в "изображении/t38". РЕКОМЕНДУЕТСЯ, чтобы также поддерживались хорошо известные варианты буквенных наименований кодеков.
- Параметр "эхокомпенсация", 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).

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

- шифрокомплект протокола RTP кодируется в виде ключевого слова "sc-rtп", 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 IETF RFC 2234. В шифрокомплекте или между соседними шифрокомплектами, когда применяется несколько шифрокомплектов, НЕ ДОЛЖНЫ допускаться пробелы. Следующий пример иллюстрирует использование шифрокомплекта и списка шифрокомплекта:

```
sc-rtп 62/51;64/51;60/50
```

Фактический список шифрокомплектов, поддерживаемых в проекте IPСablecom, содержится в Рек. МСЭ-Т J.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 являются:

```

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).

8.2.2.4 Параметр Capabilities

Параметр Capabilities информирует контроллера медиашлюза о его возможностях 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 таблице 3 Спецификации Аудио/Видео кодеки (J.161) проекта IP-Cablecom. 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. 8.2.2.7;
- ключевое слово "sc-rtp", за которым следуют двоеточие и список шифрокомплектов протокола RTP, разделенных точкой с запятой, использующих то же кодирование, что и в параметре LocalConnectionOptions;
- ключевое слово "sc-rtcp", за которым следуют двоеточие и список шифрокомплектов протокола RTCP, разделенных точкой с запятой, использующих то же кодирование, что и в параметре LocalConnectionOptions.

Когда присутствует несколько параметров, то значения разделяются запятой.

Примерами возможностей являются:

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

Следует отметить, что кодеки и алгоритмы обеспечения безопасности являются просто примерами – действительные кодеки и поддерживаемые алгоритмы, а также используемое кодирование подробно рассматриваются в отдельных Рекомендациях проекта IP-Cablecom (см. Рек. МСЭ-Т J.170, J.162 и J.161). Следует отметить, что кодеки и алгоритмы обеспечения безопасности являются просто примерами – действительные кодеки и поддерживаемые алгоритмы, а также используемое кодирование подробно рассматриваются в отдельных Спецификациях проекта IP-Cablecom (см. стандарты RFC 1827 и RFC 1034). 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.

8.2.2.5 Параметры соединения

Параметры соединения кодируются в виде строки пар "тип и значение", где тип – это один из кодов, приведенных в таблице 8, а значение является десятичным целым числом. Типы отделяются от значений знаком "=". Параметры отделяются друг от друга запятой.

В таблице 8 описаны типы параметров соединения:

Таблица 8/J.171.1 – Типы параметров 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. Контроллеры медиашлюза, которые принимают нераспознанные расширения, ДОЛЖНЫ безоговорочно игнорировать эти расширения. Если конечная точка принимает пакеты RTSP с этими статистическими оценками, она ДОЛЖНА в ответ на команды DeleteConnection и AuditConnection возвратить параметры с определением Remote (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

8.2.2.6 Коды причины

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

E: 900 Неправильное функционирование конечной точки

Список кодов причины содержится в п. 7.6.

8.2.2.7 Режим соединения

Режим соединения описывает режим функционирования соединения. Возможные значения представлены в таблице 9.

Таблица 9/J.171.1 – Значения режима 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: inactive	Шлюз не должен ни посылать, ни принимать пакеты.
M: loopback	Шлюз должен вводить конечную точку в режим "кольцевая проверка линий".
M: conttest	Шлюз должен вводить конечную точку в режим проверки целостности.
M: netwloop	Шлюз должен вводить конечную точку в режим сетевого шлейфа.
M: netwtest	Шлюз должен вводить конечную точку в режим проверки целостности сети.

8.2.2.8 Кодирование имен событий/сигналов

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

Ниже приводятся допустимые примеры имен событий:

IT/col	Начальная проверка целостности в пакете линии связи ISUP.
MT/oc	Операция завершена в завершающем пакете протокола MF.
col	Начальная проверка целостности в пакете линии связи ISUP, предполагается, что пакет линии связи ISUP, по умолчанию, является пакетом для конечной точки.
IT/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 события с обозначением с подстановочными знаками.

IT/all	Все события в пакете линии ISUP.
--------	----------------------------------

И наконец, знак "звездочка" может быть использован для обозначения "всех соединений", а знак доллара – "текущего" соединения. Ниже приводятся допустимые примеры таких обозначений:

IT/ma@*	Начальное событие медиа RTP по всем соединениям для конечной точки.
IT/rt@\$	Контроль посылки вызова по текущему соединению.

Начальное множество пакетов событий для транкинговых шлюзов содержится в Приложении А.

8.2.2.9 Параметр RequestedEvents

Параметр RequestedEvents 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. 7.3.1. Действия при описании кодируются в виде списка ключевых слов, заключенных в круглые скобки и разделенных запятыми. В таблице 10 представлены коды для различных действий.

Таблица 10/J.171.1 – Коды действий

Действие	Код
Немедленное уведомление	N
Накопление	A
Игнорирование	I
Удержание сигнала(ов) активным(и)	K
Встроенное действие NotificationRequest	E
Встроенное действие ModifyConnection	C

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.

Список запрашиваемых событий кодируется одной строкой с группами событий/действий, разделенных запятыми. Примерами кодирования параметра RequestedEvents являются:

R: oc(N), of(N) Завершение операции "уведомление", сбой операции "уведомление".

Встроенное действие NotificationRequest соответствует формату:

E (R (<RequestedEvents>), S (<SignalRequests>)),

где каждый параметр R, D и S является необязательным и, возможно, предоставляется в другом порядке.

Встроенное действие ModifyConnection соответствует формату:

C (M (<ConnectionMode₁> (<ConnectionID₁>)), . . . ,
M (<ConnectionMode_n> (ConnectionID_n)))

Следующий пример иллюстрирует использование встроенного действия ModifyConnection:

R: ma@23B34D(A, C(M(sendrecv(\$))), oc(N), of(N))

Это означает: послать сигнал медиа по соединению "23B34D", изменить режим соединения для соединения "текущего соединения" на "передача/прием". Уведомить о событиях, указав на состояние "операция завершена" и "ошибка операции".

8.2.2.10 Параметр SignalRequests

Параметр SignalRequests 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. Медиашлюз 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 шлюзом медиа, 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, т. е. 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. 7.3.1):

- 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 "mysignal" будет включен: `mysignal(+)`, `mysignal`

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 тональный сигнал послышки вызова:

```
rt(to=6000)
rt(to(6000))
```

Отдельные сигналы могут определять дополнительные сигнальные параметры.

Параметры сигналов будут заключаться в круглые скобки, как в следующем гипотетическом примере:

```
S: display(10/14/17/26, "555 1212", CableLabs)
```

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

```
S: signal1, signal2
```

8.2.2.11 Параметр ObservedEvents

Параметры ObservedEvents 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 являются:

```
O: ma@A43B81
O: ft
O: IT/ft
O: IT/ft, IT/mt
```

8.2.2.12 Параметр RequestedInfo

Параметр RequestedInfo содержит список кодов параметров, разделенных запятой, как определено в п. 8.2.2. "Линии параметров"; в п. 7.3.8 перечисляются параметры, которые могут быть проконтролированы. Также поддерживаются значения, перечисленные в таблице 11:

Таблица 11/J.171.1 – Значения параметра RequestedInfo

Параметр RequestedInfo	Код
LocalConnectionDescriptor	LC
RemoteConnectionDescriptor	RC

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

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

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

F: A

8.2.2.13 Параметр QuarantineHandling

Параметр QuarantineHandling 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: loop discard

8.2.2.14 Параметр DetectEvents

Параметр DetectEvents кодируется в виде списка событий, разделенных запятыми, как, например:

T: ft, mt

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

8.2.2.15 Параметр EventStates

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

ES: MO/rlc

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

8.2.2.16 Параметр RestartMethod

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

RM: restart

8.2.2.17 Параметр VersionSupported

Параметр VersionSupported кодируется в виде списка поддерживаемых версий, разделенных запятыми, как, например:

VS: MGCP 1.0, MGCP 1.0 TGCP 1.0

8.2.2.18 Идентификатор вызова

Идентификатор вызова кодируется в виде шестнадцатеричной строки, не превышающей 32 знака. Идентификатор вызова воспринимается в виде строк, а не как численные значения.

8.2.2.19 Идентификатор соединения

Идентификатор соединения кодируется в виде шестнадцатеричной строки, не превышающей 32 знака. Идентификатор соединения воспринимается в виде строк, а не как численные значения.

8.2.2.20 Параметр MaxMGCPDatagram

Параметр MaxMGCPDatagram кодируется в виде строки, содержащей до девяти десятичных цифр – начальные нули не допускаются. Следующий пример иллюстрирует использование этого параметра:

MD: 8100

8.3 Форматы заголовков ответа

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

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

200 1201 OK

В таблице 12, ниже, сведены параметры ответов, наличие которых в заголовке ответа является обязательным или необязательным в зависимости от команды, которая инициировала ответ. Читателю, однако, следует изучить определения отдельных команд, поскольку данная таблица содержит только краткую информацию. Буква М означает "обязательная", буква О – "необязательная", а буква F – "запрещенная".

Таблица 12/J.171.1 – Связь параметров с ответом команды

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
ResponseAck (Прим. 5)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)	О (Прим. 1)
CallId	F	F	F	F	F	F	О	F
ConnectionId	О (Прим. 2)	F	F	F	F	О	F	F
RequestIdentifier	F	F	F	F	F	О	F	F
LocalConnectionOptions	F	F	F	F	F	О	О	F
ConnectionMode	F	F	F	F	F	F	О	F
RequestedEvents	F	F	F	F	F	О	F	F
SignalRequests	F	F	F	F	F	О	F	F
NotifiedEntity	F	F	F	F	F	О	О	О
ReasonCode	F	F	F	F	F	F	F	F
ObservedEvents	F	F	F	F	F	О	F	F
ConnectionParameters	F	F	О (Прим. 3)	F	F	F	О	F
Specific Endpoint ID	О	F	F	F	F	О	F	F
MaxEndPointIds	F	F	F	F	F	F	F	F
NumEndPoints	F	F	F	F	F	О	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	О	F	F
EventStates	F	F	F	F	F	О	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	О	F	F
VersionSupported	F	F	F	F	F	О	F	О

Таблица 12/J.171.1 – Связь параметров с ответом команды

Имя параметра	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
LocalConnection Descriptor	O (Прим. 4)	O (Прим. 4)	F	F	F	F	O	F
MaxMGCPDatagram	F	F	F	F	F	O	F	F
RemoteConnection Descriptor	F	F	F	F	F	F	O	F

ПРИМЕЧАНИЕ 1. – 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

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

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

ПРИМЕЧАНИЕ 4. – LocalConnectionDescriptor ДОЛЖЕН быть передан с положительным ответом (код 200) на команду CreateConnection. Он ДОЛЖЕН быть также передан в ответе на команду ModifyConnection, если модификация приводит к изменению Local Connection Descriptor. LocalConnectionDescriptor кодируется как "описание сеанса", как определено в п. 8.4. От заголовка ответа он отделяется пустой строкой.

ПРИМЕЧАНИЕ 5. – Параметр ResponseAck, как идентификатор транзакции, не указан в п. 7.3 и не видим в нашем примере API. Те, кто реализуют систему, могут выбрать другой подход.

Ниже приводятся описания параметров ответов для каждой из команд.

8.3.1 Команда CreateConnection

В случае сообщения CreateConnection, строка ответа сопровождается параметром Connection-Id с успешным ответом (код 200). Кроме того, передается параметр LocalConnectionDescriptor с положительным ответом. Параметр LocalConnectionDescriptor кодируется в виде "описания сеанса", как описано в п. 8.4. 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 18 96 97 0
a=rtpmap:96 G726-32/8000
a=rtpmap:97 telephone-event/8000
a=mptime:20 10 - 10
```

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

```
200 1204 OK
K:
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 18 96 97 0
a=rtpmap:96 G726-32/8000
a=rtpmap:97 telephone-event/8000
a=mptime:20 10 - 10
```

Окончательный ответ подтверждается параметром подтверждения ответа:

```
000 1204
```

8.3.2 Команда ModifyConnection

В случае успешного сообщения ModifyConnection, 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. 8.4. Он отделяется от заголовка ответа пустой строкой.

```
200 1207 OK

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

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

```
526 1207 No bandwidth
K:
```

Окончательный ответ подтверждается параметром подтверждения ответа:

```
000 1207 OK
```

8.3.3 Команда DeleteConnection

В зависимости от варианта сообщения DeleteConnection строка ответа может сопровождаться строкой параметров соединения, как определено в п. 8.2.2.5.

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

8.3.4 Команда NotificationRequest

Ответ на команду NotificationRequest не содержит каких-либо дополнительных параметров ответа.

8.3.5 Команда Notify

Ответ на команду Notify не содержит никаких дополнительных параметров ответа.

8.3.6 Команда AuditEndPoint

В случае команды AuditEndPoint, 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, все еще будут обеспечиваться. Каждое локальное имя конечной точки, "расширяемое" подстановочным знаком, будет находиться на отдельной строке при использовании кода параметра "SpecificEndPointId", например:

```
200 1200 OK
Z: ds/ds1-1/1@tgw.whatever.net
Z: ds/ds1-1/2@tgw.whatever.net
ZN: 24
```

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:IT,
  m:sendonly;recvonly;sendrecv;inactive
A: a:G728, p:20, e:on, s:off, t:1, v:IT,
  m:sendonly;recvonly;sendrecv;inactive
A: a:G729A; p:30-90, e:on, s:on, t:1, v:L,
  m:sendonly;recvonly;sendrecv;inactive;confrnce
```

8.3.7 Команда AuditConnection

В случае команды AuditConnection, 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: [128.96.41.12]
L: mp:20;10, a:PCMU;G728
M: sendrecv
P: PS=622, OS=31172, PR=390, OR=22561, PL=5, JI=29, LA=50
```

```
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.

8.3.8 Команда RestartInProgress

Ответ на команду RestartInProgress может содержать имя другого контроллера медиашлюза для связи, например, когда контроллер медиашлюза переадресует конечную точку другому контроллеру медиашлюза агенту, как в случае:

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

8.4 Кодирование описания сеанса

Описание сеанса кодируется в соответствии с протоколом описания сеанса (SDP); однако, как определяется ниже, транкинговые шлюзы могут использовать некоторые упрощающие допущения

относительно описания сеанса. Следует отметить, что в соответствии со стандартом IETF RFC 2327 описания сеансов чувствительны к регистру.

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

- если параметр среды установлен на "аудио", то описание сеанса относится к аудиослуге;
- если параметр среды установлена на "изображение", то описание сеанса относится к услуге изображения, такой как факсимильное сообщение.

8.4.1 Использование аудиослужбы по протоколу SDP

В транкинговом шлюзе необходимо описывать только сеансы, которые используют лишь один тип среды передачи MIME в каждый конкретный момент времени; либо "аудио" (для речевых данных или для данных речевого диапазона), либо "изображение" (для вызовов факсимильной связи Т.38). Параметры протокола SDP, существенные для типов среды передачи как "аудио", так и "изображение", определены в п. 8.4.2. Параметры, характерные для "аудио" определяются в п. 8.4.3. Параметры, характерные для "изображения", если они используются в Т.38, определяются в п. 8.4.4. Транкинговые шлюзы ДОЛЖНЫ поддерживать описания сеансов, которые соответствуют этим правилам и представлены в следующем порядке:

- 1) профиль протокола SDP, представленный ниже;
- 2) стандарт RFC 2327 (*SDP: Протокол описания сеанса*).

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

Предоставленный профиль протокола SDP описывает использование протокола описания сеанса в TGCP. Общее описание и пояснения к отдельным параметрам, также как и к большинству параметров, используемых для аудио можно найти только в стандарте IETF RFC 2327; параметры, характерные для изображения для Т.38, можно найти в Т.38.

Ниже уточняется, какие значения конечных точек TGCP необходимо предусмотреть для этих полей (передача) и что конечные точки TGCP должны делать со значениями, предоставленными или не предоставленными для этих полей (прием).

Следует отметить, что профиль SDP, используемый здесь не соответствует модели предложение/ответ, определенной в RFC 3264. Таким образом, если MGC нужно взаимодействовать с другим объектом, использующей модель предложение/ответ, MGC может понадобиться отредактировать SDP, который был получен от конечной точки.

8.4.2 Общие параметры SDP для аудио услуги и услуги изображения

8.4.2.1 Версия протокола (v=)

```
v=<version>
v=0
```

Передача: ДОЛЖНА обеспечиваться в соответствии со стандартом IETF RFC 2327 (т. е. v = 0).

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

8.4.2.2 Начало (o=)

Поле "начало" (o=) состоит из 6 подполей по стандарту IETF 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 дефис²⁵. Поскольку конечные точки TGCP не знают, когда запрашивается конфиденциальность, им ДОЛЖНЫ всегда использовать дефис.

²⁵ Поскольку конечные точки TGCP не знают, когда запрашивается конфиденциальность, им ДОЛЖНЫ всегда использовать дефис.

Прием: Это поле СЛЕДУЕТ проигнорировать.

Session-ID (идентификатор сеанса):

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

Прием: Это поле СЛЕДУЕТ проигнорировать.

Version (версия):

Передача: В соответствии со стандартом IETF RFC 2327.

Прием: Это поле СЛЕДУЕТ проигнорировать.

Network Type (тип сети):

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

Прием: Это поле СЛЕДУЕТ проигнорировать.

Address Type (тип адреса):

Передача: ДОЛЖЕН быть использован тип "IP4".

Прием: Это поле СЛЕДУЕТ проигнорировать.

Address (адрес):

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

Прием: Это поле ДОЛЖНО быть проигнорировано.

8.4.2.3 Имя сеанса (s=)

s=<session-name>

s=-

Передача: В качестве имени сеанса ДОЛЖЕН быть использован дефис.

Прием: Это поле ДОЛЖНО быть проигнорировано.

8.4.2.4 Информация о сеансе и среде передаче (i=)

i=<session-description>

Передача: Это поле НЕ ДОЛЖНО использоваться для протокола TGSP.

Прием: Это поле ДОЛЖНО быть проигнорировано.

8.4.2.5 Идентификатор URI (u=)

u= <URI>

Передача: Это поле НЕ ДОЛЖНО использоваться для протокола TGSP.

Прием: Это поле ДОЛЖНО быть проигнорировано.

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

e=<e-mail-address>

p=<phone-number>

Передача: Это поле НЕ ДОЛЖНО использоваться для протокола TGSP.

Прием: Это поле ДОЛЖНО быть проигнорировано.

8.4.2.7 Данные соединения (c=)

Данные 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(ых) потока(ов), для которого(ых) 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(ых) потока(ов), для которого(ых) он предназначен.

8.4.2.8 Ширина полосы пропускания (b=)

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

Передача: Информация о полосе пропускания является необязательной в протоколе SDP, но ее всегда СЛЕДУЕТ включать²⁶. Когда используется отображение rtpmap или неизвестный кодек²⁷, 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. Подробная информация о вычислении значения ширины полосы пропускания содержится в п. 7.5 Спецификации кодека проекта IPCablecom (J.161).

Прием: ДОЛЖНА присутствовать требуемая максимальная ширина полосы пропускания мультимедийного потока в килобитах в секунду.

²⁶ Если это поле не используется, то контроллер шлюза может не дать разрешение на использование соответствующей ширины полосы пропускания.

²⁷ Неизвестный кодек – это кодек, не описанный в кодеке проекта IPCablecom (Рек. МСЭ-Т J.161).

8.4.2.9 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 остановки СЛЕДУЕТ использовать нуль. НЕ СЛЕДУЕТ использовать время повторений и временные зоны; если они используются, это должно соответствовать стандарту IETF RFC 2327.

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

8.4.2.10 Атрибуты (a=)

a= <attribute> : <value>

a= mptime: <alternative 1> <alternative 2> ...a = X-pc-bridge: <number-ports>

a= <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.

Следует отметить, что в протоколе SDP требуется, чтобы неизвестные атрибуты игнорировались.

mptime: 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, тогда, как последующие элементы в этом списке ДОЛЖНЫ быть либо десятичными числами, либо дефисами. Для тех форматов среды передачи, где применяется не одна скорость пакетирования (например, в речевых кодеках, связанных с телефонными событиями или комфортным шумом), в соответствующем месте в списке периодов пакетирования ДОЛЖЕН кодироваться дефис ("-").

Прием: Передается список периодов пакетирования, которые может использовать удаленная конечная точка для данного соединения, по одному периоду пакетирования для каждого формата 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=". Если атрибуты "mptime" или "ptime" не присутствуют, тогда медиашлюз по умолчанию должен принять значение для известных кодеков (как определено в RFC 1890).

X-pc-bridge:

Передача: Конечные точки протокола TGCP НЕ ДОЛЖНЫ использовать этот атрибут.

Прием: Конечные точки протокола TGCP, если получают такой атрибут, ДОЛЖНЫ его проигнорировать.

recvonly:

Передача: Это поле не обслуживается конечной точкой протокола TGCP.

Прием: Это поле ДОЛЖНО быть игнорировано.

sendrecv:

Передача: Это поле не обслуживается конечной точкой протокола TGCP.

Прием: Это поле ДОЛЖНО быть игнорировано.

sendonly:

Передача: Это поле не обслуживается конечной точкой протокола TGCP.

Прием: Это поле ДОЛЖНО быть игнорировано.

ptime:

Передача: Атрибут ptime СЛЕДУЕТ передать, если он был получен в параметре RemoteConnectionDescriptor или, если в контроллере медиашлюза использовался параметр LocalConnectionOption периода пакетирования ('p:').

Прием: Это поле ДОЛЖНО быть проигнорировано, если протокол SDP содержит атрибут "ptime" (как требуется в устройствах, соответствующих проекту PacketCable). Если атрибут "ptime" отсутствует, тогда это поле используется для определения интервала пакетирования для всех кодеков, содержащихся в описании протокола SDP. Если атрибуты "ptime" и "ptime" не присутствуют, тогда контроллер медиашлюза по умолчанию должен принять подходящее значение для известных кодеков, как определено в Рек. МСЭ-Т J.161.

8.4.3 Использование аудиоуслуг SDP

Следующие параметры SDP применяются в уровне среды передачи и являются характерными для использования аудиоуслуг. Конечные точки PacketCable НЕ ДОЛЖНЫ передавать какой-либо из этих параметров через дескриптор среды изображения (см. 8.4.4). Однако если конечная точка получает SDP с параметрами атрибутов, характерными только для изображения с дескриптором среды аудио, параметры СЛЕДУЕТ игнорировать. Кроме того, если должны быть предоставлены параметры возможностей медиаданных, каждый дескриптор возможностей среды передачи (который включает в себя строку описания возможностей, a=cpsc, с 0 или более строк атрибутов возможностей, например, a=spra) ДОЛЖЕН появляться после последнего атрибута, и каждый дескриптор возможностей среды передачи ДОЛЖЕН перечисляться отдельно.

8.4.3.1 Ключи шифрования

k=<method>

k=<method> : <encryption-keys>

Услуги обеспечения безопасности для проекта IPcablecom определены в Рек. МСЭ-Т J.170. Услуги обеспечения безопасности, определенные для протоколов RTP и RTCP, не соответствуют аналогичным услугам, описанным в стандартах IETF RFC 1889, IETF RFC 1890 и IETF RFC 2327. В целях взаимодействия с устройствами, не входящими в проект IPcablecom, параметр "k", таким образом, не будет использоваться для передачи параметров обеспечения безопасности.

Передача: НЕ ДОЛЖНО использоваться.

Прием: Это поле СЛЕДУЕТ проигнорировать.

8.4.3.2 Attributes (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>...

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.

rtptime:

Передача: Это поле, если используется, ДОЛЖНО использоваться в соответствии со стандартом IETF RFC 2327. Это поле МОЖЕТ быть использовано для неизвестных, равно как и для известных кодеков. Используемые имена кодирования содержатся в отдельной Рекомендации проекта IECablecom (см. Рек. МСЭ-Т J.161 и J.170). В данном соединении динамического типа полезной нагрузки для данного метода кодирования ДОЛЖЕН быть одинаковым для указаний приема и передачи. Более того, если установлено соответствие динамического типа полезной нагрузки данному методу кодирования, то в данном соединении этот тип полезной нагрузки НЕ ДОЛЖЕН в дальнейшем соответствовать другому методу кодирования.

Прием: Это поле, если используется, ДОЛЖНО использоваться в соответствии со стандартом IETF RFC 2327. Для заданного соединения, реализации НЕ СЛЕДУЕТ считать ошибочными, если заданный метод кодирования отображает различные типы полезной нагрузки в направлениях передача и прием, или, если заданный тип полезной нагрузки переназначается.

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 спецификации, в которой подробно описывается использование формата.

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

X-rc-codecs:

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

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 сигнализировано посредством строки среды передачи (m=).

X-rc-secret:

Передача: Это поле содержит закрытую информацию для сквозной передачи и (возможно) заполнение PAD (сборщик/разборщик пакетов), которые должны использоваться для обеспечения безопасности протоколов RTP и RTCP. Закрытая информация и PAD кодируются аналогично параметру "ключ шифрования" (k=) в стандарте IETF 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 IETF RFC 2045, за исключением максимальной длины строки, которая здесь не задана. Метод "clear" НЕ ДОЛЖЕН быть использован, если закрытая информация содержит какие-либо символы, запрещенные в протоколе SDP.

Требования относительно того, когда передавать заполнение PAD, описаны в Рекомендации безопасности 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: Передается закрытая информация для сквозной передачи, которая должна использоваться для обеспечения безопасности протоколов RTP и RTCP. Если закрытая информация для сквозной передачи присутствует, она используется согласно Рекомендации безопасности МСЭ-Т J.170 и ДОЛЖНА быть отделена от закрытой информации, по крайней мере, одним пробелом. Для заполнения PAD и закрытой информации используется один и тот же метод кодирования

X-rc-suites-rtp
X-rc-suites-rtcp

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, т.е. 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 IETF RFC 2234. В пределах шифрокомплекта пробелы не допускаются. Следующий пример иллюстрирует использование шифрокомплекта:

62/51

Действительный список шифрокомплектов содержится в Рек. 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 сигнализировано посредством новой строки шифрокомплекта, где желательный для использования шифрокомплект будет перечислен первым.

8.4.3.3 Уведомления о среде передачи (m=)

Уведомления о среде передачи (m=) состоят из 4 подполей:

```
M=<media> <port> <transport> <format>[<format>]
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 IETF 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 IETF 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".

Media Formats (форматы среды передачи):

Передача: ДОЛЖЕН быть использован соответствующий тип среды передачи, как определено в стандарте IETF RFC 2327. В частности, это поле содержит список из одного или нескольких типов полезной нагрузки протокола RTP, которые готова принять по соединению данная конечная точка и которые она предпочтет передать. Каждый тип полезной нагрузки однозначно соответствует кодеку, статическое или динамическое соответствие. СЛЕДУЕТ использовать статическое соответствие, если таковое имеется (например, 0 для PCMU, 8 для PCMA). Если используется динамическое соответствие кодека полезной нагрузке, то ДОЛЖЕН также присутствовать атрибут rtpmap и ДОЛЖНЫ выполняться руководящие указания, содержащиеся в п. 8.4.3.2.

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

8.4.4 Использование услуги изображения по протоколу SDP для T.38

Следующие параметры SDP применяются для уровня среды передачи и являются характерными для использования услуг передачи изображения для T.38. Конечные точки PacketCable NE ДОЛЖНЫ передавать какой-либо из этих параметров через дескриптор среды аудио (см. п. 8.4.3). Однако если конечная точка получает SDP с параметрами атрибутов, характерными только для аудио с дескриптором среды изображения, параметры СЛЕДУЕТ игнорировать. Кроме того, если должны быть предоставлены параметры возможностей медиаданных, каждый дескриптор возможностей медиаданных (который включает в себя строку описания возможностей, a=cpsc, с 0 или более строк атрибутов возможностей, например, a=cpag) ДОЛЖЕН появляться после последнего атрибута, и каждый дескриптор возможностей медиа ДОЛЖЕН перечисляться отдельно.

8.4.4.1 Ключи шифрования

k= <method>

k= <method> : <encryption-keys>

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

Передача: НЕ ДОЛЖНЫ использоваться.

Прием: Это поле СЛЕДУЕТ проигнорировать.

8.4.4.2 Атрибуты (a=)

a= <attribute> : <value>

a=T38FaxVersion: <version>

a=T38MaxBitrate: <bitrate>

a=T38FaxRateManagement: <faxratemanagement>

a=T38FaxMaxBuffer: <maxbuffer>

a=T38FaxMaxDatagram: <maxsize>

a=T38FaxUdpEC: <ECmethod>

a=T38FaxFillBitRemoval

a=T38FaxTranscodingMMR

a=T38FaxTranscodingJBIG

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

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

Следует отметить, что SDP требует игнорирования неизвестных атрибутов.

T38FaxVersion:

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

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

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

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

T38MaxBitRate:

Передача: Атрибут T38MaxBitRate включаться НЕ ДОЛЖЕН.

Прием: Атрибут T38MaxBitRate СЛЕДУЕТ игнорировать.

T38FaxRateManagement:

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

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

T38FaxMaxBuffer:

Передача: Атрибут T38FaxMaxBuffer включаться НЕ ДОЛЖЕН.

Прием: Атрибут T38FaxMaxBuffer СЛЕДУЕТ игнорировать.

T38FaxMaxDatagram:

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

Прием: Конечные точки НЕ ДОЛЖНЫ посылать дейтаграмму большую, чем указано в атрибуте T38FaxMaxDatagram. До того, как отправить какую-либо дейтаграмму T.38, конечная точка ДОЛЖНА удостовериться, что она не выходит за границы, определенных для данного атрибута. Если указанное значение T38FaxMaxDatagram слишком мало для поддержки избыточности для данной дейтаграммы, но достаточно для поддержки дейтаграммы T.38 без избыточности, то конечная точка ДОЛЖНА послать эту дейтаграмму без избыточности. Если значение слишком мало, чтобы послать дейтаграмму без избыточности, конечная точка НЕ ДОЛЖНА посылать дейтаграмму T.38, а ДОЛЖНА сгенерировать индикатор отказа.

T38FaxUdpEC:

Поддержка избыточности (добавление резервных битов) обязательна, в то время как поддержка для прямой коррекции ошибок необязательна. Использование одной из этих схем требует согласования.

Передача: Атрибут T38FaxUdpEC ДОЛЖЕН включаться. Значение "t38UDPFEC" МОЖЕТ посылаться, если поддерживается FEC, и либо с командой не предоставлен RCD, либо значение атрибута, полученного в RCD для этой команды "t38UDPFEC". В другом случае ДОЛЖНО посылаться "t38UDPRedundancy".

Прием: Избыточность ДОЛЖНА использоваться, если значением атрибута T38FaxUdpEC является "t38UDPRedundancy". Если значением атрибута T38FaxUdpEC является "t38UDPFEC", и конечной точкой поддерживается FEC, то СЛЕДУЕТ использовать FEC. Если значением атрибута T38FaxUdpEC является "t38UDPFEC", и FEC не поддерживается, то ДОЛЖНА использоваться избыточность. Если данный атрибут не включен, конечная точка НЕ ДОЛЖНА использовать избыточность или FEC.

T38FaxFillBitRemoval:

Поддержка удаления битов дополнения необязательна, и любое ее использование должно быть согласовано.

Передача: Если поддерживается вставка и удаление битов дополнения и необходимо их использовать, и команда либо не включает RCD, либо включает в себя RCD при наличии атрибута T38FaxFillBitRemoval, то T38FaxFillBitRemoval ДОЛЖЕН включаться, и ДОЛЖНЫ использоваться вставка и удаление битов дополнения. Во всех других случаях, атрибут T38FaxFillBitRemoval НЕ ДОЛЖЕН включаться, а вставка и удаление битов дополнения использоваться НЕ ДОЛЖНЫ.

Прием: Вставка и удаление битов дополнения НЕ ДОЛЖНЫ использоваться, если атрибут T38FaxFillBitRemoval отсутствует.

MMR перекодирование не применяется к T.38, базирующемуся на UDPTL.

Передача: При использовании UDPTL для T.38, атрибут T38FaxTranscodingMMR включаться НЕ ДОЛЖЕН.

Прием: Если атрибут T38FaxTranscodingMMR предоставлен для T.38, базирующемуся на UDPTL, команда ДОЛЖНА быть отклонена (код ошибки 505 – неподдерживаемый RemoteConnectionDescriptor).

T38FaxTranscodingJBIG:

JBIG перекодирование не применяется к T.38, базирующемуся на UDPTL.

Прием: При использовании UDPTL для T.38, атрибут T38FaxTranscodingJBIG включаться НЕ ДОЛЖЕН.

Передача: Если атрибут T38FaxTranscodingJBIG предоставлен для T.38, базирующемуся на UDPTL, команда ДОЛЖНА быть отклонена (код ошибки 505 – неподдерживаемый RemoteConnectionDescriptor).

8.4.4.3 Уведомления о медиаданных (m=)

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

```
m= <media> <port> <transport> <fmt list>  
"m= image 3456 udpt1 t38"
```

Media (медиаданные):

Передача: ДОЛЖЕН использоваться тип медиаданных "изображение" для T.38, базирующегося на UDPTL.

Прием: Полученным типом медиаданных ДОЛЖНО быть "изображение" для T.38, базирующегося на UDPTL.

Port (порт):

Передача: Это поле ДОЛЖНО быть заполнено в соответствии со стандартом RFC 2327. Описанный порт является портом приема. Передающий порт может быть другим.

Прием: Это поле ДОЛЖНО использоваться в соответствии со стандартом RFC 2327. Описанный порт является портом приема. Передающий порт может быть другим.

Transport (транспортирование):

Передача: ДОЛЖЕН быть использован транспортный протокол "udptl" для T.38, базирующегося на UDPTL.

Прием: Транспортным протоколом ДОЛЖЕН быть протокол "udptl" для T.38, базирующегося на UDPTL. Реализациям СЛЕДУЕТ принимать "UDPTL" в форме как верхнего регистра, так и в форме строки "udptl" смешанного (верхнего и нижнего) регистра.

Форматы медиаданных:

Передача: ДОЛЖЕН быть тип медиаданных "t38".

Прием: ДОЛЖЕН быть тип медиаданных "t38".

8.5 Передача по протоколу UDP

8.5.1 Надежная доставка сообщений

Сообщения MGCP доставляются по протоколу UDP. Команды посылаются по одному из IP-адресов, определяемых в Базе данных имен доменов (DNS – Domain Name System) для заданной конечной точки или MGC. 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 порт²⁸, 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 контроллерам медиашлюза. Чтобы свести к минимуму проблемы совместимости сверху вниз, РЕКОМЕНДУЕТСЯ, чтобы контроллер медиашлюза всегда явным образом указывал порт MGCP для использования в сообщениях TGCP (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, т. е. 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.

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. 8.5.2.

Кроме того, когда на повторные команды не приходит ответ, предполагается, что объект места назначения команды недоступен. Запрашивающий объект, как описано в п. 7.4, отвечает за поиск резервных услуг и/или освобождение существующих или ждущих обработки соединений.

²⁸ Каждая конечная точка может быть обеспечена адресом и портом отдельного MGC.

8.5.2 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. Транкинговые шлюзы ДОЛЖНЫ реализовывать стратегии повторных передач, используя экспоненциальные задержки с конфигурируемыми начальными и максимальными значениями таймеров повторной передачи.

Транкинговым шлюзам 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 режиме реального 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.

8.5.3 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 UDP, 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.

8.6 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 1210 ds/ds1-1/1@tgw.whatever.net MGCP 1.0 TGCP 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. 7.4.3.1).

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.

8.7 Идентификаторы транзакций и установление трехсторонней связи

Идентификаторы транзакций являются целыми числами в диапазоне от 1 до 999 999 999. Контроллеры медиашлюза 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. Контроллеры медиашлюза 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 MGC, 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 ответа²⁹ ответ, который подтверждается этим

²⁹ В противоположность команде с параметром "подтверждение ответа".

сообщением, может быть исключен. Шлюзы должны безоговорочно отбрасывать дальнейшие команды от этого MGC, когда идентификатор транзакции попадает в эти диапазоны, а ответ был выдан менее чем T_{hist} секунд назад.

Пусть $term_{new}$ и $term_{old}$ будут именем конечной точки, соответственно, в новой команде cmd_{new} и в некоторой старой команде cmd_{old} . Тогда идентификаторы транзакций, подлежащие подтверждению в команде cmd_{new} , СЛЕДУЕТ определять следующим образом:

- 1) Если $term_{new}$ не содержит каких-либо подстановочных знаков, то это:
 - a) Неподтвержденные ответы на старые команды, где $term_{old}$ равен $term_{new}$.
 - b) Необязательно, один или несколько неподтвержденных ответов, где $term_{old}$ содержал подстановочный знак "any-of" ("любой из"), а возвращенное в ответе имя конечной точки было $term_{new}$.
 - c) Необязательно, один или несколько неподтвержденных ответов, где $term_{old}$ содержал подстановочный знак "all" ("все"), а $term_{new}$ охвачен подстановочным знаком в имени $term_{old}$.
 - d) Необязательно, один или несколько неподтвержденных ответов, где $term_{old}$ содержал подстановочный знак "any-of" ("любой из"), имя конечной точки не было возвращено, а $term_{new}$ охвачен подстановочным знаком в имени $term_{old}$.
- 2) Если $term_{new}$ содержит подстановочный знак "all" ("все"), то это:
 - a) Необязательно, один или несколько неподтвержденных ответов, где $term_{old}$ содержал подстановочный знак "all" ("все"), а $term_{new}$ охвачен подстановочным знаком в $term_{old}$
- 3) Если $term_{new}$ содержит подстановочный знак "any of" ("любой из"), то это:
 - a) Необязательно, один или несколько неподтвержденных ответов, где $term_{old}$ содержал подстановочный знак "all" ("все"), а $term_{new}$ охвачен подстановочным знаком в $term_{old}$, если подстановочный знак "any of" ("любой из") в $term_{new}$ был заменен подстановочным знаком "all".

Данный ответ НЕ СЛЕДУЕТ подтверждать в двух отдельных сообщениях.

Следующие примеры иллюстрируют использование этих правил:

- Если $term_{new}$ есть "ds/ds1-2/1", а $term_{old}$ есть "ds/ds1-2/1", тогда старый ответ может быть подтвержден по правилу 1a.
- Если $term_{new}$ есть "ds/ds1-1/3", а $term_{old}$ есть "*", тогда старый ответ может быть подтвержден по правилу 1c.
- Если $term_{new}$ есть "ds/ds1-2/*", а $term_{old}$ есть "*", тогда старый ответ может быть подтвержден по правилу 2a.
- Если $term_{new}$ есть "ds/ds1-2/\$", а $term_{old}$ есть "ds/ds1-2/*", 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 MGC, 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.

8.8 Временные ответы

В некоторых случаях, время завершения транзакций может быть значительно больше, чем в других случаях. В протоколе TGCP, в качестве транспортного протокола, используется протокол UDP, а надежность обеспечивается избирательными повторными передачами с использованием выдержек времени, которые основаны на оценке суммы времени прохождения сигнала в прямом и обратном направлениях в сети и времени завершения транзакции. Поэтому значительный разброс времени завершения транзакции остается нерешенным 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 (СЛЕДУЕТ использовать код ошибки 407).

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. 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. 7.4), другими словами, отправитель окончательного ответа повторно передает его, если "подтверждение ответа" не будет принято вовремя. Ответ "подтверждение ответа" никогда не подтверждается.

9 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 протокола TGCP содержатся в Рек. 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 транкинговых шлюзов в проекте IPCablecom.

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:

- **Код** Однозначно определяемый код события пакета, используемый для события/сигнала.
- **Описание** Краткое описание события/сигнала.
- **Событие** В этом столбце проставляется отметка о проверке, если событие может быть запрошено MGC. 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.
- **Сигнал** Если в данном столбце для события ничего нет, тогда сигнал о событии не может быть сигнализирован в команде контроллером медиашлюза. В противном случае тип события определяют следующие символы:
 - "OO" Сигнал "включено/выключено". Сигнал будет включенным до тех пор, пока по команде от контроллера медиашлюза 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.
- **Дополнительная информация** 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.

A.1 Пакет линии связи ISUP

Package name: IT.

В целях идентификации событий и сигналов для пакета "IT" используются следующие коды. Если медиашлюз поддерживает данный пакет "IT", то он ДОЛЖЕН поддерживать все события и сигналы перечисленные здесь.

Таблица А.1/J.171.1 – События и сигналы пакета линии ISUP

Код	Описание	Событие	Сигнал	Дополнительная информация
co1	Тональный сигнал целостности 1	√	ТО	Выдержка времени = 3 секунд
co2	Тональный сигнал целостности 2	√	ТО	Выдержка времени = 3 секунд
ft	Тональный сигнал факсимильной связи	√	–	
ld	Соединение большой продолжительности	С	–	
ma	Мультимедийный запуск	С	–	
mt	Модемные тональные сигналы	√	–	
oc	Операция завершена	√	–	
of	Ошибка операции	√	–	
ro	Тональный сигнал требования освобождения цепи	–	ТО	Выдержка времени = 30 секунд
rt	Тональный сигнал контроля посылки вызова	–	С, ТО	Выдержка времени = 180 секунд
TDD	Тональные сигналы устройств электросвязи для слабослышащих (TDD)	√		

Ниже приводятся определения отдельных событий и сигналов:

Тональный сигнал целостности 1 (co1): Согласно Рек. МСЭ-Т Q.724 – тональный сигнал с частотой 2010 Гц. Для согласования с текущим осуществлением проверки целостности, событие НЕ СЛЕДУЕТ генерировать, пока тональный сигнал не удален. Этот тональный сигнал относится к типу сигнала ТО – проверка целостности будет применяться только для определенного промежутка времени. Обеспечиваемый процесс может изменять значение по умолчанию.

Тональный сигнал целостности 2 (co2): Согласно Рек. МСЭ-Т Q.724 – тональный сигнал с частотой 2010 Гц. Для согласования с текущим осуществлением проверки целостности, событие НЕ СЛЕДУЕТ генерировать, пока тональный сигнал не удален. Этот тональный сигнал относится к типу сигнала ТО – проверка целостности будет применяться только для определенного промежутка времени. Обеспечиваемый процесс может изменять значение по умолчанию.

Тональный сигнал целостности используются, когда MGC хочет инициировать проверку целостности. Существует два типа проверки: одинарный тональный сигнал и двойной тональный сигнал. Абонент, порождающий сигналы проверки целостности, обнаруживает подходящий тональный сигнал для задействованной линии связи. Например, в проверке целостности из четырехпроводной в двухпроводную линию связи будут использоваться следующие сообщения:

Исходящий шлюз

```
RQNT 1234 ds/ds3-1/ds1-6/17@tgw1.example.net
X: AB123FE0
S: co2
R: co1
```

Завершающий шлюз

```
CRCX 1234 ds/ds1-4/7@tgw2.example.net
C: A3C47F21456789F0
L: p:10, a:PCMU
M: conttest
```

Исходящий шлюз посылает запрашиваемый сигнал, и ожидает возврата подходящего тонального сигнала для рассматриваемой линии. Когда исходящий шлюз обнаруживает этот тональный сигнал, он полагает, что контроль целостности прошел успешно, затем генерирует событие "co1", о котором, в указанном примере, уведомляется MGC. Если проверка не проходит (не успешна) до истечения выдержки времени, будет генерироваться событие "операция завершена", и, в данном случае, посылаться MGC. Аналогичным образом, если ошибка предшествует появлению выдержки времени, будет генерироваться событие "ошибка операции". События "oc" и "of" будут параметризованы с сообщаемым именем события/сигнала, т. е., в данном случае, "co1".

Тональный сигнал факсимильной связи (ft): Событие "тональный сигнал факсимильной связи" генерируется каждый раз, когда обнаруживается факсимильный вызов – см., например, Рек. МСЭ-Т T.30 или V.21.

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 действительный³⁰ мультимедийный пакет согласно протоколу RTP. Это событие может быть использовано для синхронизации локального сигнала, например сигнала контроля посылки вызова, с поступлением мультимедийного пакета от другой стороны.

Это событие может быть обнаружено в соединении. Когда соединение не определено, это событие относится ко всем соединениям для конечной точки независимо от того, когда они создаются.

Модемные тональные сигналы (mt): Событие "модемный тональный сигнал" генерируется каждый раз, когда обнаруживается вызов модема – см., например, Рек. 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 "тональный сигнал целостности 1". Сообщение о завершении может содержать в качестве параметра имя сигнала, время существования которого закончилось, как в случае:

О: IT/oc (IT/co1)

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

О: IT/oc (IT/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:

О: IT/oc (B/C)

³⁰ Когда используются услуги аутентификации и обеспечения защиты целостности, пакет RTP не считается действительным, пока он не пройдет проверку на обеспечение защиты.

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: IT/of(IT/co2)

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

O: IT/of(IT/rt@0A3F58)

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

Событие "ошибка операции" может быть генерировано дополнительно, как определено в базовом протоколе, например, когда имеет место неуспешное выполнение встроенной команды ModifyConnection, как в случае:

O: IT/of(B/C(M(sendrecv(AB2354))))

Тональный сигнал требования освобождения цепи (ro): Тональный сигнал требования освобождения цепи, так же известный как, тональный сигнал перезагрузки, определяется в Рек. МСЭ-Т E.180/Q.35.

Тональный сигнал контроля посылки вызова (rt): Тональный сигнал звукового вызова определяется в Рек. МСЭ-Т E.180/Q.35.Т. Определение тонального сигнала определяется национальными характеристиками тонального сигнала контроля посылки вызова, и МОЖЕТ устанавливаться через предоставление услуг. Сигнал контроля посылки вызова может применяться как в конечной точке, так и в соединении.

Тональные сигналы устройств электросвязи для слабослышащих (TDD): Событие TDD генерируется каждый раз, когда обнаруживается вызов TDD – см., например, Рек. МСЭ-Т V.18.

Дополнение I

Взаимодействия режимов

Соединение по протоколу MGCP может пропускать один или несколько потоков мультимедийных пакетов. Эти потоки являются либо входящими (от удаленной конечной точки), либо исходящими (сгенерированными в конечной точке линии связи). Параметр "режим соединения" устанавливает направление потоков и формирует такие потоки. Когда имеется только одно соединение с конечной точкой, то отображение этих потоков является простым. Конечная точка линии связи воспроизводит входящий поток на другом конце линии связи и генерирует исходящий поток от сигнала линии связи, в зависимости от режима параметра.

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 конечной точке и от нее. В приведенной ниже таблице описывается, как должны смешиваться мультимедийные пакеты от разных соединений, когда существует одно или несколько соединений.

В таблице I.1 описывается, как следует смешивать различные соединения, когда одно или несколько "текущих" соединений используются одновременно. В данном случае, под текущим соединением понимается соединение, которое находится в одном из следующих режимов:

- "передача/прием";
- "только передача";
- "только прием".

Таблица I.1/J.171.1 – Смешивание различных соединений, когда одновременно используется одно или несколько текущих соединений

		Режим соединения А					
		sendonly	recvonly	sendrecv	loopback/ conttest	inactive	netwloop/ netwttest
Режим соединения В	sendonly	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=A_{in}$	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=NA$	$A_{out}=A_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$
	recvonly		$A_{out}=NA$ $B_{out}=NA$ $H_{out}=A_{in}+B_{in}$	$A_{out}=H_{in}$ $B_{out}=NA$ $H_{out}=A_{in}+B_{in}$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=B_{in}$	$A_{out}=A_{in}$ $B_{out}=NA$ $H_{out}=B_{in}$
	sendrecv			$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}+B_{in}$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=B_{in}$	$A_{out}=A_{in}$ $B_{out}=H_{in}$ $H_{out}=B_{in}$
	loopback/ conttest				$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=cot$
	inactive					$A_{out}=NA$ $B_{out}=NA$ $H_{out}=NA$	$A_{out}=A_{in}$ $B_{out}=NA$ $H_{out}=NA$
	netwloop/ netwttest						$A_{out}=A_{in}$ $B_{out}=B_{in}$ $H_{out}=NA$

Соединения в режимах "кольцевая проверка сети", "проверка целостности сети" или "неактивный" не влияют на соединения. В таблице I.1 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 линии;
- 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 конечной точке, где "cot" обозначает проверку целостности либо в режиме "проверка целостности", либо в режиме "кольцевая проверка".
- NA указывает на абсолютное отсутствие потока.

Дополнение II

Примеры кодирования команд

В настоящем Дополнении представлены примеры команд и ответов с использованием текущего кодирования. Примеры приведены для каждой команды. Все приведенные в командах и ответах комментарии носят необязательный характер.

II.1 Команда NotificationRequest

Первый пример иллюстрирует использование команды NotificationRequest, которая будет инициировать проверку целостности и поиск контроля проверки. "Уведомленный объект" для конечной точки устанавливается на ca@ca1.whatever.net:5678, а параметр RequestIdentifier будет повторяться в соответствующей команде Notify.

```
RQNT 1201 ds/ds1-1/2@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0
N: mgc@mgc1.whatever.net:5678
X: 0123456789AC
R: col, oc(N), of(N)
S: col
```

Ответ указывает на то, что транзакция была успешной:

```
200 1201 OK
```

II.2 Команда Notify

Приведенный ниже пример иллюстрирует применение сообщения Notify для уведомления об успешной проверке целостности, как обозначается в наблюдаемых событиях.

С момента описания "уведомленного объекта" в инициирующей команде NotificationRequest, начинается повторение. Также включается параметр RequestIdentifier для установления связи команды Notify с командой NotificationRequest (включает встроенный запрос на уведомление в примитивах управления соединением), которая генерирует:

```
NTFY 2002 ds/ds1-1/2@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0
N: mgc@mgc1.whatever.net:5678
X: 0123456789AC
O: col
```

Ответ на команду Notify указывает на то, что транзакция была успешной:

```
200 2002 OK
```

II.3 Команда CreateConnection

В первом примере иллюстрируется использование команды CreateConnection для создания соединения в заданной конечной точке. Соединение будет частью заданного идентификатора вызова (CallId). В параметре LocalConnectionOptions описывается, что μ -закон согласно Рек. МСЭ-Т G.711 будет использоваться кодеком, а период пакетирования составит 10 мс. Режимом соединения будет режим "только прием":

```
CRCX 1204 ds/ds1-1/17@tgw2.whatever.net MGCP 1.0 TGCP 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 иллюстрируется команда CreateConnection, содержащая запрос на уведомление и параметр RemoteConnectionDescriptor:

```
CRCX 1205 ds/ds1-1/1@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
X: 0123456789AD
R: MO/sup(addr(K0, 4,1,1, s2), id(K0,0,0,7,3,2,5,5,5,1,2,3,4,s0))
S: MO/ans
```

```
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 линия уже была занята. Следовательно, не возвращаются ни идентификатор соединения, ни описание сеанса:

```
401 2005 Circuit already seized
```

В третьем примере иллюстрируется использование временного ответа и установления трехсторонней связи:

```
CRCX 1206 ds/ds1-1/1@tgw.whatever.net MGCP 1.0 TGCP 1.0
K: 1205
C: A3C47F21456789F0
L: p:10, a:PCMU
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
```

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, т. е. номер полезной нагрузки равен 0.

Немного позднее принимается окончательный ответ:

```
200 1206 OK
K:
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
```

MGC 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 ds/ds1-1/21@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
N: mgc@mgc1.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 КТСОП тональные сигналы контроля посылки вызова, пока она не обнаружит в соединении аудиоинформацию, определенную для сигнала контроля посылки вызова:

```
MDCX 1210 ds/ds1-1/3@abc5.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: 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 контроллера медиашлюза)

В этом примере MGC просто передает транкинговому шлюзу команду исключить соединение `FDE234C8` в заданной конечной точке:

```
DLCX 1210 ds/ds1-1/1@tgw.whatever.net MGCP 1.0 TGCP 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
```

II.6 Команда `DeleteConnection` (от транкингового шлюза)

В этом примере транкинговый шлюз посылает MGC команду `DeleteConnection`, чтобы было исключено соединение в заданной конечной точке. Параметр `ReasonCode` определяет причину исключения; кроме того, предоставляются параметры соединения:

```
DLCX 1210 ds/ds1-1/1@tgw-2567.whatever.net MGCP 1.0 TGCP 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
```

MGC передает шлюзу успешный ответ:

```
200 1210 OK
```

II.7 Команда `DeleteConnection` (от контроллера медиашлюза для нескольких соединений)

В первом примере MGC передает транкинговому шлюзу команду исключить все соединения, связанные с вызовом "A3C47F21456789F0" в заданной конечной точке:

```
DLCX 1210 ds/ds1-1/6@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
```

Ответ указывает на успешное выполнение команды и на исключение соединения(й):

```
250 1210 OK
```

Во втором примере MGC передает транкинговому шлюзу команду исключить все соединения, относящиеся ко всем заданным конечным точкам:

```
DLCX 1210 ds/ds1-1/*@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0
```

Ответ указывает на успешное выполнение команды:

```
250 1210 OK
```

II.8 Команда AuditEndpoint

В первом примере MGC хочет узнать, какие конечные точки имеются у определенного транкингового шлюза, и, следовательно, используется подстановочный знак "все из" для локальной части имени конечной точки. MGC хочет узнать только два имени конечной точки:

```
AUEP 1200 *@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0  
ZM: 2
```

Транкинговый шлюз указывает на успешное выполнение команды и предоставляет список имен конечных точек. Конечное число имен 24 конечных точек, сопоставленных с определенными подстановочными знаками:

```
200 1200 OK  
Z: ds/ds1-1/1@tgw-2567.whatever.net  
Z: ds/ds1-1/2@tgw-2567.whatever.net  
ZN: 24
```

Во втором примере запрашиваются возможности одной из конечных точек:

```
AUEP 1201 ds/ds1-1/1@tgw-2567.whatever.net MGCP 1.0 TGCP 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.

```
200 1201 OK  
A: a:PCMU, p:10-100, e:on, s:off, v:IT, m:sendonly;recvonly;sendrecv;  
inactive;loopback;conttest;netwloop;netwtest  
A: a:G728, p:30-90, e:on, s:on, v:IT, m: sendonly;recvonly;sendrecv;  
inactive;loopback;conttest;netwloop
```

В третьем примере MGC контролирует всю возможную информацию для конечной точки:

```
AUEP 2002 ds/ds1-1/1@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0  
F: R, S,X,N,I,T,O,ES
```

Ответ указывает на успешное выполнение команды:

```
200 2002 OK  
R: IT/ft,mt (N)  
S:  
X: 0123456789B1  
N: [128.96.41.12]  
I: 32F345E2  
T: ft  
O:  
ES:
```

В списке запрашиваемых событий содержатся два события. Там, где имя пакета не определено, предполагается имя пакета по умолчанию. Это справедливо и в отношении действий, так что для события "IT/ft" должно, таким образом, предполагаться по умолчанию действие Notify. Пропуск

значения для параметра "SignalRequests" означает, что в текущее время нет активных сигналов. 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 и EventStates – пустой.

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 ds/ds1-1/18@tgw-2567.whatever.net MGCP 1.0 TGCP 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: mgc@mgc1.whatever.net
L: p:10, a:PCMU
M: sendrecv
P: PS=395, OS=22850, PR=615, OR=30937, PL=7, JI=26, LA=47
```

```
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 ds/ds1-1/2@tgw.whatever.net MGCP 1.0 TGCP 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 транкинговым шлюзом сообщения RestartInProgress для информирования MGC о том, что определенная конечная точка будет выведена из эксплуатации через 300 секунд:

```
RSIP 1200 ds/ds1-1/1@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0
RM: graceful
RD: 300
```

Ответ MGC указывают на то, что транзакция была выполнена успешно:

```
200 1200 OK
```

Во втором примере сообщение RestartInProgress, переданное транкинговым шлюзом, информирует MGC о том, что все конечные точки транкингового шлюза вводятся в эксплуатацию через 0 секунд, т. е. они вновь находятся в эксплуатации. Задержка также могла быть опущена:

```
RSIP 1204 *@tgw-2567.whatever.net MGCP 1.0 TGCP 1.0  
RM: restart  
RD: 0
```

Ответ MGC указывает на успешное выполнение команды и, кроме того, предоставляет рассматриваемым конечным точкам новый "уведомленный объект":

```
200 1204 OK  
N: MGC-1@whatever.net
```

И наоборот, команда может быть неуспешной из-за нового "уведомленного объекта", как в случае:

```
521 1204 OK  
N: MGC-1@whatever.net
```

В таком случае команда должна быть повторена, чтобы соответствовать "процедуре перезапуска" (см. п. 7.4.3.5), применяемой в это время к MGC "MGC-1@whatever.net".

Дополнение III

Пример последовательности операций для вызова

В настоящем Дополнении приводится пример последовательности операций для вызова между внутрисетевым пользователем, использующего неопределенный адаптер MTA и протокол сигнализации³¹, и внесетевым пользователем, имеющего доступ через транкинговый шлюз, с использованием протокола TGCP и шлюз сигнализации, поддерживающий сигнализацию SS7 ISUP. Следует отметить, что несмотря на то, что данная последовательность операции для вызова – действующая, она просто является примером, который либо может, либо не может использоваться на практике.

В приведенной ниже в рисунке III.1 последовательности операций для вызова, CMS означает сервер управления вызовом, MGC означает контроллер медиашлюза, TGW означает транкинговый шлюз, SG означает шлюз сигнализации:

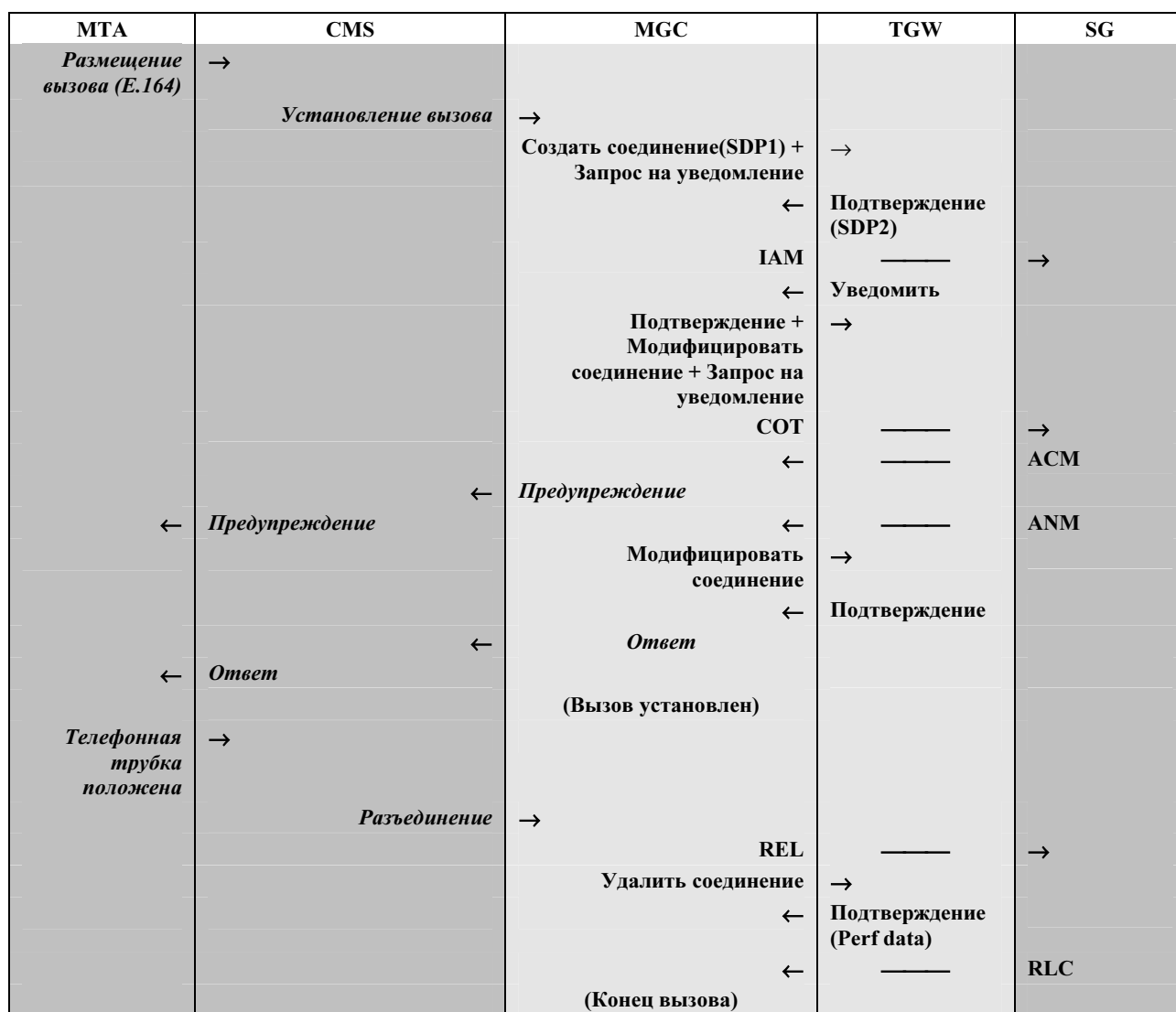


Рисунок III.1/J.171.1 – Пример последовательности операций для вызова

Во время этих обменов информацией, MGC использует профиль TGCP протокола MGCP для управления транкинговым шлюзом. Обмены информацией имеют место на двух сторонах. Неопределенные протоколы предполагаются между адаптером MTA, CMS и MGC.

³¹ Это может быть либо сигнализация NCS, либо сигнализация DCS.

Предполагается, что адаптер МТА указывает на желание обозначает (прямо или косвенно) контроллеру медиашлюза на желание установить речевую передачу на номер телефона E.164, и включить описание сеанса с таким запросом. CMS ищет запрошенный номер E.164 и устанавливает, что его необходимо поместить вне сети, и затем связать с соответствующим MGC. MGC решает, что необходимо установить вызов через транкинговый шлюз `tgw.whatever.net`. Более того, MGC решает, что для данного вызова следует осуществить проверку целостности.

Первой командой является комбинация команд `CreateConnection` и `NotificationRequest` `NotificationRequest`, посылаемая транкинговому шлюзу.

```
CRCX 2001 ds/ds1-1/6@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: inactive
X: 0123456789B0
R: co2, oc, of
S: co1
```

```
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 транкинговый шлюз получает команду начать проверку целостности, отыскать результат проверки и сообщить об этом. Генерирование сигнала проверки целостности и определение об успешности выполнения команды (или сбое) через механизм события синхронизированы, поэтому когда имеет место событие "co2", проверка "co1" будет прекращена. Следует отметить, что конечная точка отбирается только для поддержки кодека PCMU, т. е. число полезной нагрузки равняется 0. Часть команды по созданию соединения получает команду создать соединение в режиме "inactive" в заданной конечной точке с периодом пакетирования 10 мс, согласно G.711. Также эта команда включает описание сеанса, полученное от начального адаптера МТА.

Выходящий транкинговый шлюз подтверждает команду, передавая в описании сеанса свои собственные параметры, такие как адрес, порты и профиль протокола RTP, а также идентификатор соединения для нового соединения.

```
200 2001 OK
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
```

MGC посылает через транкинговый шлюз сообщение SS7 IAM коммутатору, соединенному с линией, где находится вызов. Содержимое данного сообщения обозначает, что должна выполняться проверка целостности.

Предполагается, что проверка целостности, впоследствии, сменяется результатом, сгенерированным в событии "co2", MGC уведомляется о полученном результате:

```
NTFY 3001 ds/ds1-1/6@tgw.whatever.net MGCP 1.0 TGCP 1.0
X: 0123456789B0
O: co2
```

MGC передает SS7 COT удаленному коммутатору, обозначающее "контроль целостности выполнен успешно" и запрашивает передачу команды Notify. MGC также совмещает комбинацию команд ModifyConnection и NotificationRequest, при этом, указывая шлюзу поместить соединение в режим "только передача" и начать поиск факсимильных и модемных тональных сигналов.

200 3001 OK

.
MDCX 2006 ds/ds1-1/6@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
I: 32F345E2
M: recvonly
X: 0123456789B0
R: ft,mt

На этом этапе MGC устанавливает полудуплексный тракт передачи. Телефонный аппарат, присоединенный к входному адаптеру МТА, 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.

Затем MGC получает сообщение SS7 ACM, обозначающее, что в вызываемой части появился сигнал тревоги, и позже сообщение SS7 ACM, обозначающее, что на эту вызываемую часть получен ответ. MGC помещает соединение в полнодуплексный режим, путем передачи транкинговому шлюзу команды ModifyConnection.

MDCX 2007 ds/ds1-1/6@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
I: 32F345E2
M: sendrecv

Транкинговый шлюз немедленно отвечает на команду:

200 2007 OK

Параллельно с этим MGC информирует начальный адаптер МТА о появлении события "ответ вызову" и записывает время ответа вызова.

В этот момент соединение для вызова полностью установлено.

В какой-то более поздний момент времени, согласно принятому сценарию, телефонный аппарат, присоединенный к начальному адаптеру МТА, переходит в состояние "телефонная трубка положена", а событие "телефонная трубка снята" передается MGC (либо напрямую либо косвенно через протокол CMS, как показано здесь), посылая команду MGC, что следует завершить вызов.

MGC проверяет, чтобы вызов в действительности был завершен, например, нет сдерживающего устройства, поэтому сообщения SS7 REL посылаются отдаленному коммутатору, а команда DeleteConnection посылается транкинговому шлюзу.

DLCX 2009 ds/ds1-1/6@tgw.whatever.net MGCP 1.0 TGCP 1.0
C: A3C47F21456789F0
I: 32F345E2

Транкинговые шлюзы ответят на эту команду подтверждениями, которые включают параметры соединения для данного соединения:

250 2009 OK

P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48

Подтверждение для вызова освобождение канала в форме сообщения RLC SS7, которое также принимается MGC, в котором окончательно записывается конец вызова.

Дополнение IV

Требования конечной точки

В данном Дополнении определяются определенные требования множества конечных точек протокола TGCP.

IV.1 Поддерживаемые режимы соединения

В таблице IV.1 перечислены режимы соединения, которые ДОЛЖНА поддерживать данная конечная точка протокола TGCP.

Таблица IV.1/J.171.1 – Список режимов соединения, которые должны поддерживаться конечной точкой протокола TGCP

Тип конечной точки	Дополнительная информация конечной точки	sendonly	recvonly	sendrecv	inactive	loopback	confest	netwloop	netwfest
DS-0	Линия ISUP	√	√	√	√	√	√	√	√
DS-0	Линия СЧ	√	√	√	√	–	–	√	√

Дополнение V

Информация о совместимости

В настоящем Дополнении приводится информация о совместимости протоколов TGCP.

V.1 Совместимость протокола NCS

Даная версия протокола TGCP ориентирована и основана на Рек. J.162, насколько это возможно. Поскольку у протоколов TGCP и NCS различные типы адресов шлюзов, существует некоторые отличия. Далее рассматриваются эти отличия:

- **Режимы соединения:** NCS и TGCP используют совместно общие множества режимов соединения, но у каждого, также, есть множество режимов, которые не поддерживаются другим:
 - NCS поддерживает режим соединения "конференция" и "копирование", а TGCP – нет;
 - TGCP поддерживает режимы соединения "проверка целостности" и "кольцевая проверка линий", а NCS не поддерживает.
- **Отображения цифр набора номера:** TGCP не поддерживает параметр "отображения цифр"; тем не менее, NCS поддерживает. У этого имеется несколько последствий, например:
 - в протоколе TGCP не такой команды TGCP, которая может принимать параметр "отображения цифр";
 - действие "накапливать согласно отображению цифр набора номера" не поддерживается в протоколе TGCP;
 - параметр "отображения цифр набора номера" не может контролироваться.
- **Динамическое качество обслуживания:** В NCS поддерживается сигнализация услуг динамического качества проекта IPCablecom, однако в протоколе however TGCP – нет.

В дополнение к вышесказанному, существуют следующие, не относящиеся к протоколам, отличия между NCS и TGCP:

- **Пакеты событий:** В TGCP и NCS отличаются исходные пакеты событий.
- **Схема именования конечной точки:** Схемы именования конечной точки для TGCP и NCS имеют незначительные отличия.

V.2 Совместимость протокола MGCP

Протокол TGCP (и NCS), кроме того, является профилем протокола MGCP стандарта IETF RFC 2705 1.0; однако протокол TGCP ввел также несколько дополнений. Ниже перечисляются дополнения протокола TGCP, которые в настоящее время не включены в протокол MGCP:

- **Схема именования конечных точек:** Определенная схема именования конечных точек была введена для конечных точек DS-0. Правила использования подстановочных знаков носят более ограничительный характер, чем в протоколе MGCP, а также вводят понятие "диапазон" для конечных точек DS-0.
- **Встроенная модификация соединения:** Было введено новое действие: встроенная команда "ModifyConnection".
- **Безопасность:** В протоколе TGCP поддерживаются услуги обеспечения безопасности в проекте IPCablecom. Это влияет на параметр LocalConnectionOptions, возможности и протокол SDP.
- **Восстановление имени конечной точки:** Команда AuditEndpoint была расширена путем добавления возможности возврата некоторого числа конечных точек, которые соответствуют подстановочному знаку, а также механизма блочного восстановления имен этих конечных точек. Помимо расширения команды AuditEndpoint это подразумевает ввод двух новых имен параметров: MaxEndPointIds и NumEndpoints.

- **Поддерживаемые версии:** Ответ на команду RestartInProgress и команда AuditEndpoint были расширены параметром VersionSupported, чтобы дать контроллерам медиашлюза и шлюзам возможность определять, какие версии протоколов поддерживает каждый из них.
- **Коды ошибок:** Были введены два новых кода ошибок: 532 и 533.
- **Использование протокола SDP:** В протокол TGCP включен новый профиль использования протокола SDP. Наиболее примечательно то, что профиль и все примеры использования, в частности, требуют строгого соответствия протоколу SDP, независимо от пригодности включенных полей. Кроме того, к протоколу SDP добавлены характерные для проекта IPCablecom расширения.
- **Временный ответ:** В протокол TGCP включены дополнительный элемент и рекомендация 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 протоколе TGCP вводится множество новых пакетов событий.

Наконец, следует отметить, что в протоколе TGCP предоставляются интерпретация и, в некоторых случаях, дополнительная рекомендация или пояснения базового режима работы протокола MGCP, которые могут отражать или не отражать предполагаемый режим работы протокола MGCP.

Дополнение VI

Грамматика ABNF для TGCP

RFC 3435 включает в себя формальное описание синтакса протокола MGCP, согласно "Расширенной BNF (форма Бэкуса-Науэра) для детальных указаний по синтаксису". Данное формальное описание ссылается на разработчиков совместно функционирующих устройств. Копия синтакса протокола MGCP, дополненная аннотацией и отредактированная для наглядности применения к указаниям по Рекомендациям PacketCable, содержится в данном Дополнении.

Реализации ДОЛЖНЫ отвечать частям грамматики ABNF, относящимся к их соответственным детальным указаниям, т. е. NCS, TGCP. Также следует обратить внимание на то, что существуют несколько кодировок параметров (например, встроенный запрос, отображение цифр, имена расширения разработчиков), где грамматика NCS и/или TGCP отличается от грамматики MGCP.

Для определения различий между четырьмя разными случаями используются пять аннотаций:

- 1) Язык RFC изменен для соответствия требованиям NCS и TGCP.
- 2) Язык RFC применим к NCS и возможно MGCP, но не TGCP.
- 3) Язык RFC применим к TGCP и возможно к MGCP, но не NCS.
- 4) Язык RFC применим только к NCS и TGCP.
- 5) Язык RFC применим только к MGCP.

Язык в каждом случае выделен посредством различного семейства шрифтов так, как это определено в описании изменений ниже.

;Грамматика RFC 3435 изменена для соответствия требованиям NCS и TGCP

;Шрифт Bold показывает только NCS (и возможно MGCP)

;Шрифт Italics показывает только TGCP (и возможно MGCP)

;Шрифт Bold italics показывает только NCS и TGCP

;Текст серого цвета - только MGCP

```

MGCPMessage = MGCPCommand / MGCPResponse
MGCPCommand = MGCPCommandLine 0*(MGCPPParameter) [EOL *SDPInformation]
MGCPCommandLine = MGCPVerb 1*(WSP) transaction-id 1*(WSP)
                    endpointName 1*(WSP) MGCPversion EOL
MGCPVerb = "EPCF" / "CRCX" / "MDCX" / "DLCX" / "RQNT"
           / "NTFY" / "AUEP" / "AUCX" / "RSIP" / extensionVerb
extensionVerb = ALPHA 3(ALPHA / DIGIT) ; experimental starts with X
                    (экспериментальный пакет начинается с X)

transaction-id = 1*9(DIGIT)
endpointName = LocalEndpointName "@" DomainName
LocalEndpointName = LocalNamePart 0*("/" LocalNamePart)
LocalNamePart = AnyName / AllName / NameString
AnyName = "$"
AllName = "*"
NameString = 1*(range-of-allowed-characters)
; VCHAR except "$", "*", "/", "@"
range-of-allowed-characters = %x21-23 / %x25-29 / %x2B-2E
                               / %x30-3F / %x41-7E

DomainName = 1*255(ALPHA / DIGIT / "." / "-") ; as defined (как определено)
            / "#" number ; in RFC 821 (в RFC 821)
            / "[" IPv4address / IPv6address "]" ; see RFC 2373 (см. RFC 2373)
; Rewritten to ABNF from RFC 821 (Переписано в ABNF из RFC 821)
number = 1*DIGIT
;From RFC 2373
IPv6address = hexpart [ ":" IPv4address ]
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
; this production, while occurring in RFC 2373, is not referenced (на эту
конструкцию при появлении ее в RFC 2373 ссылаться нельзя)
; IPv6prefix = hexpart "/" 1*2DIGIT
hexpart = hexseq / hexseq ":" [ hexseq ] / ":" [ hexseq ]
hexseq = hex4 *( ":" hex4)
hex4 = 1*4HEXDIG
MGCPversion = "MGCP" 1*(WSP) 1*(DIGIT) "." 1*(DIGIT)
                    [1*(WSP) ProfileName]

ProfileName = "NCS 1.0" ; For NCS
            / "TGCP 1.0" ; For TGCP
            / VCHAR *( WSP / VCHAR)

MGCPPParameter = ParameterValue EOL
; Check infoCode if more parameter values defined (если определены еще значения
параметра, проверьте infoCode)
; Most optional values can only be omitted when auditing (наиболее необязательные
значения могут быть опущены только при проверке)
ParameterValue = ("K" ":" 0*(WSP) [ResponseAck])
                / ("B" ":" 0*(WSP) [BearerInformation])
                / ("C" ":" 0*(WSP) CallId)
                / ("I" ":" 0*(WSP) [ConnectionId])
                / ("N" ":" 0*(WSP) [NotifiedEntity])
                / ("X" ":" 0*(WSP) [RequestIdentifier])
                / ("L" ":" 0*(WSP) [LocalConnectionOptions])
                / ("M" ":" 0*(WSP) ConnectionMode)
                / ("R" ":" 0*(WSP) [RequestedEvents])
                / ("S" ":" 0*(WSP) [SignalRequests])
                / ("D" ":" 0*(WSP) [DigitMap]) ; For NCS (and MGCP) (Для NCS (и MGCP))
                / ("O" ":" 0*(WSP) [ObservedEvents])
                / ("P" ":" 0*(WSP) [ConnectionParameters])
                / ("E" ":" 0*(WSP) ReasonCode)
                / ("Z" ":" 0*(WSP) [SpecificEndpointID])
                / ("Z2" ":" 0*(WSP) SecondEndpointID)
                / ("I2" ":" 0*(WSP) SecondConnectionID)
                / ("F" ":" 0*(WSP) [RequestedInfo])
                / ("Q" ":" 0*(WSP) QuarantineHandling)
                / ("T" ":" 0*(WSP) [DetectEvents])
                / ("RM" ":" 0*(WSP) RestartMethod)
                / ("RD" ":" 0*(WSP) RestartDelay)
                / ("A" ":" 0*(WSP) [Capabilities])
                / ("ES" ":" 0*(WSP) [EventStates])
                / ("PL" ":" 0*(WSP) [PackageList]) ; Auditing only (только
проверка)

```

```

        / ("MD" ":" 0*(WSP) MaxMGCPDatagram) ; Auditing only (только
проверка)
        / (extensionParameter ":" 0*(WSP) [parameterString])
        / VersionSupported ; NCS and TGCP - response only (NCS и TGCP -
только в направлении передачи)
        / MaxEndpointIds ; NCS and TGCP
        / NumEndpoints ; NCS and TGCP - response only (NCS и TGCP -
только в направлении передачи)
; <extensionParameter> ":" parameterString defined by (определенный) NCS and (и)
TGCP
VersionSupported = "VS" ":" MGCPversion *( "," 0*(WSP) MGCPversion)
MaxEndpointIds = "ZM" ":" 0*(WSP) 1*16 (DIGIT)
NumEndpoints = "ZN" ":" 0*(WSP) 1*16 (DIGIT) ; Responses only (Только ответы)
; A final response may include an empty ResponseAck (конечный ответ может
включать пустой параметр ResponseAck)
ResponseAck = confirmedTransactionIdRange
                *( "," 0*(WSP) confirmedTransactionIdRange )
confirmedTransactionIdRange = transaction-id ["-" transaction-id]
BearerInformation = BearerAttribute 0*( "," 0*(WSP) BearerAttribute)
BearerAttribute = ("e" ":" BearerEncoding)
                / (BearerExtensionName [":" BearerExtensionValue])
BearerExtensionName = PackageLCOExtensionName
BearerExtensionValue = LocalOptionExtensionValue
BearerEncoding = "A" / "mu"
CallId = 1*32(HEXDIG)
; The audit request response may include a list of identifiers (ответ на запрос
контроля может включать в себя список идентификаторов)
ConnectionId = 1*32(HEXDIG) 0*( "," 0*(WSP) 1*32(HEXDIG) )
SecondConnectionID = ConnectionId
NotifiedEntity = [LocalName "@"] DomainName [ ":" portNumber]
LocalName = LocalEndpointName ; No internal structure (нет
внутренней структуры)
portNumber = 1*5(DIGIT)
RequestIdentifier = 1*32(HEXDIG)
LocalConnectionOptions = LocalOptionValue 0*(WSP)
                        0*( "," 0*(WSP) LocalOptionValue 0*(WSP) )
LocalOptionValue = ("p" ":" packetizationPeriod)
                  / ("a" ":" compressionAlgorithm)
                  / ("b" ":" bandwidth) ; Only for capabilities in
; NCS and TGCP (Только для возможностей
NCS и TGCP)
                  / ("e" ":" echoCancellation)
                  / ("gc" ":" gainControl)
                  / ("s" ":" silenceSuppression)
                  / ("t" ":" typeOfService)
                  / ("r" ":" resourceReservation)
                  / ("k" ":" encryptiondata)
                  / ("nt" ":" ( typeOfNetwork /
                                supportedTypeOfNetwork) )
                  / (LocalOptionExtensionName
                    [ ":" LocalOptionExtensionValue] )
                  / MPacketizationPeriod ; NCS and TGCP only (только NCS и TGCP)
                  / RTPCiphersuite ; NCS and TGCP only (только NCS и TGCP)
                  / RTCPciphersuite ; NCS and TGCP only (только NCS и TGCP)
                  / DQoSGateID ; NCS only (только NCS)
                  / DQoSReservation ; NCS only (только NCS)
                  / DQoSResourceID ; NCS only (только NCS)
                  / DQoSReserveDestination ; NCS only (только NCS)
                  / CallContentId ; TGCP only (только TGCP)
                  / CallContentDestination ; TGCP only (только TGCP)

Capabilities = CapabilityValue 0*(WSP)
              0*( "," 0*(WSP) CapabilityValue 0*(WSP) )
CapabilityValue = LocalOptionValue
                / ("v" ":" supportedPackages)
                / ("m" ":" supportedModes)

PackageList = pkgNameAndVers 0*( "," pkgNameAndVers)
pkgNameAndVers = packageName ":" packageVersion
packageVersion = 1*(DIGIT)
; For NCS and TGCP, range format is only allowed for capabilities (Для NCS и
TGCP, диапазон форматов разрешен только для возможностей)
; and not for LocalConnectionOptions. (а не для LocalConnectionOptions)

```

```

packetizationPeriod = 1*4(DIGIT) ["-" 1*4(DIGIT)]
compressionAlgorithm = algorithmName 0*("; " algorithmName)
algorithmName = 1*(SuitableLCOCharacter)
bandwidth = 1*4(DIGIT) ["-" 1*4(DIGIT)]
echoCancellation = "on" / "off"
gainControl = "auto" / ["-"] 1*4(DIGIT)
silenceSuppression = "on" / "off"
typeOfService = 1*2(HEXDIG) ; 1 hex only for capabilities (одна
шестнадцатиричная цифра только для возможностей)
resourceReservation = "g" / "cl" / "be"
; encryption parameters are coded as in SDP (RFC 2327) (параметры шифрования
кодируются как в SDP (RFC 2327))
; NOTE: encryption key may contain an algorithm as specified in RFC 1890
(ПРИМЕЧАНИЕ: Ключ шифрования может содержать алгоритм, как определено в RFC 1890.)
encryptiondata = ( "clear" ":" encryptionKey )
/ ( "base64" ":" encodedEncryptionKey )
/ ( "uri" ":" URIToObtainKey )
/ ( "prompt" ) ; defined in SDP, not usable in MGCP! (определено
в SDP, не подлежит использованию в MGCP!)
encryptionKey = 1*(SuitableLCOCharacter) / quotedString
; See RFC 2045
encodedEncryptionKey = 1*(ALPHA / DIGIT / "+" / "/" / "=")
URIToObtainKey = 1*(SuitableLCOCharacter) / quotedString
typeOfNetwork = "IN" / "ATM" / "LOCAL" / OtherTypeOfNetwork
; Registered with IANA - see RFC 2327 (Зарегистрировано с IANA - см. RFC 2327)
OtherTypeOfNetwork = 1*(SuitableLCOCharacter)
supportedTypeOfNetwork = typeOfNetwork *("; " typeOfNetwork)
supportedModes = ConnectionMode 0*("; " ConnectionMode)
supportedPackages = packageName 0*("; " packageName)
packageName = 1*(ALPHA / DIGIT / HYPHEN) ; Hyphen neither first or last (Дефис
не может стоять первым или последним)
LocalOptionExtensionName = VendorLCOExtensionName
/ PackageLCOExtensionName
/ OtherLCOExtensionName
VendorLCOExtensionName = "x" ("+" / "-" ) 1*32(SuitableExtLCOCharacter)
PackageLCOExtensionName = packageName "/"
1*32(SuitablePkgExtLCOCharacter)
; must not start with "x-" or "x+" (не должно начинаться с "x-" или "x+")
OtherLCOExtensionName = 1*32(SuitableExtLCOCharacter)
; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; defined by NCS/TGCP
MPacketizationPeriod = "mp" ":" multiplepacketizationPeriod
multiplepacketizationPeriod = mpPeriod 0*("; " mpPeriod)
mpPeriod = 1*4(DIGIT) / HYPHEN
RTPciphersuite = "sc-rtp" ":" ciphersuite
RTCPciphersuite = "sc-rtcp" ":" ciphersuite
ciphersuite = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]
AuthenticationAlgorithm = 1*( ALPHA / DIGIT / "-" / "_" )
EncryptionAlgorithm = 1*( ALPHA / DIGIT / "-" / "_" )
; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; defined by NCS only (определено только в NCS)
DQoSGateID = "dq-gi" [":" 1*8(HEXDIG)] ; Only empty for (Пустое
только для)
; capabilities (возможностей)
DQoSReservation = "dq-rr" ":" DQoSResMode *("; " DQoSResMode)
DQoSResMode = "sendresv" / "recvresv" / "snrcresv" /
"sendcomt" / "recvcomt" / "snrccomt"
DQoSResourceID = "dq-ri" ":" 1*8(HEXDIG)
DQoSReserveDestination = "dq-rd" ":" IPv4address [":" portNumber]
; <LocalOptionExtensionName> ":" <LocalOptionExtensionvalue>
; defined by TGCP only (определено только в TGCP)
CallContentId = "es-cci" ":" 1*8(HEXDIG)
CallContentDestination = "es-ccd" ":" IPv4address ":" portNumber

LocalOptionExtensionValue = (1*(SuitableExtLCOValChar)
/ quotedString)
*("; " (1*(SuitableExtLCOValChar)
/ quotedString))

```

```

;Note: No "data" mode. (ПРИМЕЧАНИЕ: Режим "данные" нет.)
ConnectionMode = "sendonly" / "recvonly" / "sendrecv"
                / "confrnce" / "inactive"
                / "loopback" / "contttest" ; TGCP (and MGCP) only (только TGCP (и
MGCP))
                / "replcate" ; NCS only (только NCS)
                / "netwloop" / "netwttest"
                / ExtensionConnectionMode
ExtensionConnectionMode = PkgExtConnectionMode
PkgExtConnectionMode   = packageName "/" 1*(ALPHA / DIGIT)
RequestedEvents = requestedEvent 0*("," 0*(WSP) requestedEvent)
requestedEvent = (eventName ["(" requestedActions ")"])
                / (eventName ["(" requestedActions ")"]
                  ["(" eventParameters ")"] )
eventName = [(packageName / "**") "/" ]
            (eventId / "all" / eventRange
              / "*" / "#") ; for DTMF (для DTMF)
            ["@" (ConnectionId / "$" / "**")]
eventId = 1*(ALPHA / DIGIT / HYPHEN) ; Hyphen neither first nor last (Дефис не
может стоять первым или последним)
eventRange = "[" 1*(DigitMapLetter / (DIGIT "-" DIGIT) /
              (DTMFLetter "-" DTMFLetter)) "]"
DTMFLetter = "A" / "B" / "C" / "D"
requestedActions = requestedAction 0*("," 0*(WSP) requestedAction)
requestedAction = "N" / "A"
                / "D" ; For NCS (and MGCP) (Для NCS (и MGCP))
                / "S" / "I" / "K"
                / "E" ["(" EmbeddedRequest ")"]
                / ExtensionAction
                / "C" ["(" EmbeddedModeChange ; For NCS and TGCP (Для NCS и
TGCP)
                0*("," 0*WSP EmbeddedModeChange) ")"] ; only (только)
; NCS and TGCP define the Embedded ModifyConnection action. (NCS и TGCP
определяют действие Embedded ModifyConnection.)
; MGCP grammar does not allow for the format used in NCS and TGCP: (Грамматика
MGCP не допускает форматы, используемые в NCS и TGCP:)
EmbeddedModeChange = "M" ["(" ConnectionMode ["(" EmConnectionId ")"] ")"]
EmConnectionId = ConnectionId / "$"
ExtensionAction = PackageExtAction
PackageExtAction = packageName "/" Action ["(" ActionParameters ")"]
Action = 1*ALPHA
ActionParameters = eventParameters ; May contain actions (Может содержать
действия)
;NOTE: Should tolerate different order when receiving, e.g. for NCS (ПРИМЕЧАНИЕ:
Различные порядки следует согласовывать при получении, например, для NCS.)
EmbeddedRequest = (
    "R" ["(" EmbeddedRequestList ")"]
    ["(", 0*(WSP) "S" ["(" EmbeddedSignalRequest ")"]
    ["(", 0*(WSP) "D" ["(" EmbeddedDigitMap ")"] )
    / (
    "S" ["(" EmbeddedSignalRequest ")"]
    ["(", 0*(WSP) "D" ["(" EmbeddedDigitMap ")"] )
    / (
    "D" ["(" EmbeddedDigitMap ")"]
    / NCSTGCPEmbeddedRequest
;Text below is for NCS and TGCP only. The difference compared to MGCP (Текст,
приведенный ниже, относится только к NCS и TGCP. Отличие от MGCP)
;is simply that the order of the items is not fixed. Also for TGCP Digit Maps
(заключается в том, что порядок элементов не фиксирован. Также для TGCP)
; are not used (не используется отображение цифр)
NCSTGCPEmbeddedRequest = NCSTGCPEmbeddedRequestItem
                        *2(["(", 0*(WSP) NCSTGCPEmbeddedRequestItem)
NCSTGCPEmbeddedRequestItem = ("R" ["(" EmbeddedRequestList ")"] )
                            / ("S" ["(" EmbeddedSignalRequest ")"] )
                            / ("D" ["(" EmbeddedDigitMap ")"] )
EmbeddedRequestList = RequestedEvents
EmbeddedSignalRequest = SignalRequests
EmbeddedDigitMap = DigitMap
SignalRequests = SignalRequest 0*("," 0*(WSP) SignalRequest )
SignalRequest = eventName [ ["(" eventParameters ")"] ]
eventParameters = eventParameter 0*("," 0*(WSP) eventParameter)
eventParameter = eventParameterValue
                / eventParameterName "=" eventParameter
                / eventParameterName ["(" eventParameters ")"]
eventParameterString = 1*(SuitableEventParamCharacter)

```

```

eventParameterName = eventParameterString
eventParameterValue = eventParameterString / quotedString
; For NCS (and MGCP) (Для NCS (и MGCP))
DigitMap = DigitString / "(" DigitStringList ")"
DigitStringList = DigitString 0*( "|" DigitString )
DigitString = 1*(DigitStringElement)
DigitStringElement = DigitPosition ["."]
DigitPosition = DigitMapLetter / DigitMapRange
; NOTE "X" is now included (ПРИМЕЧАНИЕ: "X" теперь включено.)
DigitMapLetter = DIGIT / "#" / "*" / "A" / "B" / "C" / "D" / "T"
/ "X" / ExtensionDigitMapLetter
ExtensionDigitMapLetter = "E" / "F" / "G" / "H" / "I" / "J" / "K"
/ "L" / "M" / "N" / "O" / "P" / "Q" / "R"
/ "S" / "U" / "V" / "W" / "Y" / "Z"
; NOTE "[x]" is now allowed in MGCP. (ПРИМЕЧАНИЕ: "[x]" теперь разрешен.)
; In NCS, only the "x" form is allowed (В NCS разрешена только форма "x")
DigitMapRange = "[" 1*DigitLetter "]"
/ "X" ; Added for NCS only (Добавлена только для NCS)
DigitLetter = *(DIGIT "-" DIGIT) / DigitMapLetter
ObservedEvents = SignalRequests
EventStates = SignalRequests
ConnectionParameters = ConnectionParameter
0*( "," 0*(WSP) ConnectionParameter )
ConnectionParameter = ( "PS" "=" packetsSent )
/ ( "OS" "=" octetsSent )
/ ( "PR" "=" packetsReceived )
/ ( "OR" "=" octetsReceived )
/ ( "PL" "=" packetsLost )
/ ( "JI" "=" jitter )
/ ( "LA" "=" averageLatency )
/ ( ConnectionParameterExtensionName
"=" ConnectionParameterExtensionValue )
/ RemotePacketsSent
/ RemoteOctetsSent
/ RemotePacketsLost
/ RemoteJitter
; NCS and TGCP define the following four connection parameter extension (NCS и
TGCP определяют следующие четыре имени расширения параметров)
; names: (соединения:)
RemotePacketsSent = "PC/RPS" "=" packetsSent
RemoteOctetsSent = "PC/ROS" "=" octetsSent
RemotePacketsLost = "PC/RPL" "=" packetsLost
RemoteJitter = "PC/JI" "=" jitter
packetsSent = 1*9(DIGIT)
octetsSent = 1*9(DIGIT)
packetsReceived = 1*9(DIGIT)
octetsReceived = 1*9(DIGIT)
packetsLost = 1*9(DIGIT)
jitter = 1*9(DIGIT)
averageLatency = 1*9(DIGIT)
ConnectionParameterExtensionName = VendorCPEExtensionName
/ PackageCPEExtensionName
VendorCPEExtensionName = "X" "-" 2*ALPHA
/ NCSTGCPVendorCPEExtensionName
;Text below is for NCS and TGCP only. The difference compared to MGCP (Текст,
приведенный ниже, относится только к NCS и TGCP. Отличие от MGCP)
;is simply that MGCP requires 2 alpha characters whereas NCS and TGCP
(заключается только в том, что MGCP требует 2 alpha символа, в то время как NCS)
;allow 2 or 3 alpha characters for VendorCPEExtensionName (и TGCP разрешают 2 или
3 alpha символа для VendorCPEExtensionName)
NCSTGCPVendorCPEExtensionName = "X" "-" 2*3ALPHA
PackageCPEExtensionName = packageName "/" CPName
CPName = 1*(ALPHA / DIGIT / HYPHEN)
ConnectionParameterExtensionValue = 1*9(DIGIT)
MaxMGCPDatagram = 1*9(DIGIT)
ReasonCode = 3DIGIT
[1*(WSP) "/" packageName] ; Only for 8xx (Только для 8xx)
[WSP 1*(%x20-7E)]

```

```

SpecificEndpointID = endpointName
SecondEndpointID = endpointName
RequestedInfo = infoCode 0*(", " 0*(WSP) infoCode)
infoCode = "B" / "C" / "I" / "N" / "X" / "L" / "M" / "R" / "S"
           / "D" ; For NCS (and MGCP) only (Только для NCS (и MGCP))
           / "O" / "P" / "E" / "Z" / "Q" / "T" / "RC" / "LC"
           / "A" / "ES" / "RM" / "RD" / "PL" / "MD" / extensionParameter
           / "VS" / "ZM" / "ZN" ; NCS and TGCP define these (NCS и TGCP определяют
эти)
           ; three extensionParameters (три
extensionParameters)
;NCS and TGCP allows for process and loop control in either order (NCS и TGCP
позволяют контроль процесса и шлейфа в любом порядке)
QuarantineHandling = loopControl / processControl
                   / (loopControl ", " 0*(WSP) processControl )
                   / (processControl ", " 0*(WSP) loopControl)
loopControl = "step" / "loop"
processControl = "process" / "discard"
DetectEvents = SignalRequests
RestartMethod = "graceful" / "forced" / "restart" / "disconnected"
               / "cancel-graceful" / extensionRestartMethod
extensionRestartMethod = PackageExtensionRM
PackageExtensionRM = packageName "/" 1*32(ALPHA / DIGIT / HYPHEN)
RestartDelay = 1*6(DIGIT)
extensionParameter = VendorExtensionParameter
                   / PackageExtensionParameter
                   / OtherExtensionParameter
VendorExtensionParameter = "X" ("-" / "+") 1*6(ALPHA / DIGIT)
PackageExtensionParameter = packageName "/"
                           1*32(ALPHA / DIGIT / HYPHEN)
; must not start with "x-" or "x+" (не должен начинаться с "x-" или "x+")
OtherExtensionParameter = 1*32(ALPHA / DIGIT / HYPHEN)

;If first character is a double-quote, then it is a quoted-string (Если первый
символ - двойная кавычка, то это строка-цитата)
parameterString = (%x21 / %x23-7F) *(%x20-7F) ; first and last must not (первым
и последним не может)
                                   ; be white space (быть пробел)
                                   / quotedString
MGCPResponse = MGCPResponseLine 0*(MGCPPParameter)
              *2(EOL *SDPinformation)
MGCPResponseLine = responseCode 1*(WSP) transaction-id
                  [1*(WSP) "/" packageName] ; Only for 8xx (Только для
8xx)
                  [WSP responseString] EOL
responseCode = 3DIGIT
responseString = *(%x20-7E)
SuitablePkgExtLCOCharacter = SuitableLCOCharacter
SuitableExtLCOCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
                        / "!" / "'" / "|" / "=" / "#" / "?"
                        / "." / "$" / "*" / "@" / "[" / "]"
                        / "^" / "`" / "{" / "}" / "~"
SuitableLCOCharacter = SuitableExtLCOCharacter / "/"
SuitableExtLCOValChar = SuitableLCOCharacter / ":"
; VCHAR except (за исключением) "(", ")", ",", and (и) "="
SuitableEventParamCharacter = %x21 / %x23-27 / %x2A-2B
                             / %x2D-3C / %x3E-7E
; NOTE: UTF8 encoded (ПРИМЕЧАНИЕ: Закодировано с помощью UTF8.)
quotedString = DQUOTE 0*(quoteEscape / quoteChar) DQUOTE
quoteEscape = DQUOTE DQUOTE
quoteChar = (%x00-21 / %x23-FF)
EOL = CRLF / LF
HYPHEN = "-"
; See RFC 2327 for proper SDP grammar instead. (Вместо этого приложения
см. RFC 2327 для надлежащей грамматики SDP.)
SDPinformation = SDPLine CRLF *(SDPLine CRLF) ; see RFC 2327
SDPLine = 1*(%x01-09 / %x0B / %x0C / %x0E-FF) ; for proper def.

```

Дополнение VII

Электронное наблюдение

VII.1 MGC

Форматом параметров электронного наблюдения в структуре LocalConnectionOptions с командой CRCX или MDCX, посылаемых MG, является:

- Идентификатор соединения содержимого вызова, закодированный в виде ключевого слова "es-cci", следующего за двоеточием и строкой из 8 шестнадцатеричных знаков, соответствующих 32-битовому идентификатору для идентификатора соединения содержимого вызова.
- Назначение содержимого вызова кодируется ключевым словом "es-ccd", следует за двоеточием и закодированным IP-адресом, схожим с IP-адресом части доменного имени конечной точки. IP-адрес следует за двоеточием и может занимать до 5 десятичных знаков для используемого номера порта UDP.

MGC должен включать оба параметра "es-cci" и "es-ccd" в структуру LocalConnectionOptions CRCX или MDCX, когда уведомляет MG о его параметрах электронного наблюдения.

Ниже представлен пример команды CRCX с параметрами электронного наблюдения:

```
CRCX 1204 ds/ds1-1/1@mg.cablelabs.com MGCP 1.0 TGCP 1.0
C: 5678ABCD
L: p:10, a:PCMU, es-cci:123456, es-ccd:[128.96.41.1]:3456
M: sendrecv
X: 1237
```

Ниже представлен пример команды MDCX с параметрами электронного наблюдения:

```
MDCX 1206 ds/s-1/ds1-1/1@mg.cablelabs.com MGCP 1.0 TGCP 1.0
C: 5678ABCD
I: 32F345E2
L: p:10, a:PCMU, es-cci:123456, es-ccd:[128.96.41.1]:3456
M: sendrecv
X: 1238
```

VII.2 MG

Когда MG в LCO получает CRCX с не пустым параметром "es-cci" и не пустым параметром "es-ccd", MG ДОЛЖЕН начать тиражирование, пересылка и инкапсулирование всех пакетов, которые принимаются и передаются по соединению. Процесс тиражирования, пересылки и инкапсулирования всех пакетов в соединении именуется "наблюдение содержимого вызова". MG, выполняющий наблюдение содержимого вызова, ДОЛЖЕН тиражировать, пересылать и инкапсулировать все пакеты, которые генерируются для соединения. Пакеты, которые инкапсулируются ДОЛЖНЫ быть идентичны пакетам в соединении. MG ДОЛЖЕН инкапсулировать и перенаправлять пакеты с идентификатором соединения содержимого вызова, содержащие параметр "es-cci" в поле LCO. MG ДОЛЖЕН пересылать все тиражированные и инкапсулированные пакеты IP-адресам и портам UDP, у которых в поле LCO указан параметр "es-ccd".

MG, выполняющий наблюдение содержимого вызова ДОЛЖЕН завершать наблюдение содержимого вызова, когда:

- 1) соединение завершается вследствие DLCX от MGC;
- 2) соединение завершается вследствие DLCX от MG;

- 3) соединение завершается вследствие внутренних условий ошибки, таких как:
 - MG "страдает" от неисправного компонента;
 - DS1, который отправляет конечная точка, вышел из эксплуатации;
- 4) MG получает MDCX с пустым параметром "es-cci" поля ("es-cci:") или пустым параметром "es-ccd" поля ("es-ccd:").

Если MG, выполняющий наблюдение содержимого вызова, получает MDCX с LCO, в которой содержатся действующие и новые параметры "es-cci" или "es-ccd", MG ДОЛЖЕН использовать новые параметры "es-cci" или "es-ccd", когда будет выполнять наблюдение содержимого вызова.

Если MG получает CRCX или MDCX с полями "es-cci" и "es-ccd" и не поддерживает электронное наблюдение, но может выполнять команды, исключая команду для части электронного наблюдения, MG ДОЛЖЕН вернуть ответ с кодом 210 – запрашиваемая транзакция была выполнена нормально, но шлюз не поддерживает электронное наблюдение.

Если MG получает CRCX или MDCX с полями "es-cci" и "es-ccd" и не поддерживает электронное наблюдение, но может выполнять команды, исключая команду для электронного наблюдения, вследствие ограничения ресурсов, MG ДОЛЖЕН вернуть ответ с кодом 211 – запрашиваемая транзакция была выполнена нормально, но шлюз не смог выполнить электронное наблюдение, т. к. не имеется достаточно ресурсов.

Если MG получает CRCX или MDCX с LCO, содержащей только параметры "es-cci" или "es-ccd", но не оба, и может выполнять команды, исключая команду для части электронного наблюдения, MG ДОЛЖЕН вернуть ответ с кодом 212 – запрашиваемая транзакция была выполнена нормально, но не все необходимые параметры электронного наблюдения содержатся в LCO.

Если MG, который не выполняет наблюдение содержимого вызова в соединении, получает MDCX с полями "es-cci" или "es-ccd" для этого соединения, MG ДОЛЖЕН вернуть ответ с кодом 213 – запрашиваемая транзакция была выполнена нормально, но электронное наблюдение не может быть начато в середине вызова.

Если MG получает CRCX или MDCX с неподходящими полями "es-cci" или "es-ccd", и MG может выполнять команду, исключая команду для части электронного наблюдения, MG ДОЛЖЕН вернуть ответ с кодом 214 – запрашиваемая транзакция была выполнена нормально, но электронное наблюдение было неопознано. Если MG, получающий MDCX с неподходящими параметрами "es-cci" или "es-ccd" уже был готов для выполнения наблюдения содержимого вызова, то тогда MG ДОЛЖЕН продолжить выполнение наблюдения содержимого вызова и игнорировать параметры "es-cci" и "es-ccd" в MDCX.

Когда MG получает команду AuditEndpoint с параметром возможностей, MG ДОЛЖЕН вернуть ключевое слово "es", если электронное наблюдение поддерживается.

Когда MG, выполняющий наблюдение содержимого вызова получает команду AuditConnection для этого соединения с LocalConnectionsOptions, являющейся одним из контролируемых параметров, MG ДОЛЖЕН вернуть в LCO параметры "es-cci" и "es-ccd", которые MG используют в настоящий момент для выполнения наблюдения содержимого вызова.

Дополнение VIII

Примеры пакетов событий

VIII.1 Пакет услуг оператора FGD MF

Имя пакета: MO

В таблице VIII.1 использованы коды для определения событий и сигналов для одностороннего исходящего оператора транкинговых служб СЧ "Оператора сигнализации услуг". Также поддерживается оператор сигнализации услуг MF FGC. Данный пакет будет использоваться как для обычных операторов транкинговых служб, так и для специальных запасных (аварийных) транкинговых служб.

Таблица VIII.1/J.171.1 – Коды, используемые для определения не пакетных событий и сигналов

Код	Описание	Событие	Сигнал	Дополнительная информация
ans	Ответ вызова	P	–	
ft	Тональный сигнал факсимильной связи	√	–	
ld	Соединение большой продолжительности	C	–	
mt	Модемный тональный сигнал	√	–	
orbk	Оператор контроля посылки вызова	√	–	
rbz	Отмена делает занято	P	–	
rcl	Оператор повторного вызова	–	BR	
rel	Разъединение вызова	P	BR	
res	Возобновление вызова	–	BR	
rlc	Разъединение завершено	P, S	BR	
sup(<addr>, <id>)	Установка вызова	–	TO	Переменная выдержка времени
sus	Приостановление вызова	–	BR	
swk	Момент начала	√	–	
TDD	Тональные сигналы устройств электросвязи для слабослышащих (TDD)	√		
oc	Операция завершена	√		
of	Ошибка операции	√		

Ниже приводятся определения отдельных событий и сигналов:

Ответ вызова (ans): Ответ вызова появляется во время запроса ANI OSS, т. е. вызов не обязательно должен быть на лету к оператору. После появления ответа вызова, будет установлено сдерживающее устройство, т. е. только OSS теперь может разъединить вызов.

Тональный сигнал факсимильной связи (ft): Событие "тональный сигнал факсимильной связи" генерируется каждый раз, когда обнаруживается факсимильный вызов – см., например, Рек. МСЭ-Т T.30 и V.21.

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.

Модемные тональные сигналы (mt): Событие "модемный тональный сигнал" генерируется каждый раз, когда обнаруживается вызов модема – см., например, Рек. МСЭ-Т V.8.

Оператор контроля посылки вызова (orbk): Данное событие будет генерироваться, когда поступает запрос OSS о том, что вызывающий абонент находится в опасности³².

Отмена делает занято (rbz): Это событие появляется, когда OSS помечает линию. Событие разъединения будет генерироваться, когда линия больше не занята.

Оператор повторного вызова (rc1): Данный сигнал может быть применен для вовлечения оператора повторного вызова, например, по причине сигнала отбоя клиента оператор возвращается.

Разъединение вызова (rel): Разъединение вызова может быть сигнализировано медиашлюзу, однако если установлено сдерживающее устройство, то вызов не может быть отключен, пока OSS не разъединит его. Медиашлюз генерирует событие "разъединение вызова", когда OSS собирается разъединить линию. В этом случае, событие может быть параметризовано одним из кодов причин из таблицы VIII.2, указывающего на причины для разъединения:

Таблица VIII.2/J.171.1 – Коды причин разъединения вызова

Код причины	Причина
0	Нормальное разъединение
3	Нет маршрута к адресу получателя
8	Внеочередное занятие линии
19	Нет ответа
21	Вызов отклонен
27	Адрес получателя неисправен
28	Недопустимый формат номера (например, адрес неполный)
38	Сеть неисправна
111	Ошибка протокола/сигнализации, неопределенный протокол/сигнализация (например, выдержка времени)

Возобновление вызова (res): Данный сигнал указывает на то, что другой абонент возобновил вызов, т. е. абонент перешел в состоянии "телефонная трубка снята".

Разъединение завершено (rlc): Конечная точка и MGC используют событие/сигнал "разъединение завершено" для подтверждения о том, что вызов был разъединен и линия доступна для другого вызова.

Установка вызова (sup(<addr>, <id>)): Установка вызова с системой услуг оператора обеспечивается с использованием адреса и информации об идентификации. Адресная информация будет в виде:

addr (MF₁, MF₂, ..., MF_n),

а информация об идентификации будет в виде:

id (MF₁, MF₂, ..., MF_n),

где каждый из MF_i будет одним из следующих символов цифр MF из таблицы VIII.3:

³² Если вызывающий абонент находится в состоянии on-hook (телефонная трубка снята), обычно применяется вызов звонка, тогда как в случае, если вызывающий абонент находится в состоянии off-hook (телефонная трубка положена), то будет применяться тональный сигнал освобождения линии.

Таблица VIII.3/J.171.1 – Символы цифр MF

Символ	Цифра MF	Символ	Цифра MF
0	MF 0	K0	MF K0 или KP
1	MF 1	K1	MF K1
2	MF 2	K2	MF K2
3	MF 3	S0	MF S0 или ST
4	MF 4	S1	MF S1
5	MF 5	S2	MF S2
6	MF 6	S3	MF S3
7	MF 7	K0	MF K0 или KP
8	MF 8		
9	MF 9		

Таким образом, примером сигнала установки вызова может быть:

`sup(addr(K0, 5,5,5,1,2,1,2, SO), id(K0, 5,5,5,1,2,3,4, SO))`

Приостановление вызова (sus): Данный сигнал указывает на то, что другой абонент приостановил вызов, т. е. абонент перешел в состояние "телефонная трубка положена".

Момент начала (swk): Контроллер медиашлюза может запросить медиашлюз об уведомлении о появлении сигнала момента начала.

Тональные сигналы устройств электросвязи для слабослышащих (TDD): Событие TDD генерируется, когда обнаруживается вызов TDD – см., например, Рек. МСЭ-Т V.18.

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 остановки 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:

O: MO/oc (MO/sup)

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

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:

O: MO/of (MO/sup)

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

VIII.2 Завершающий пакет протокола MF

Имя пакета: MT

В данной версии Рекомендации TGCP, этот пакет может быть использован только для проверки занятости линии (BLV) и оператора прерывания (OI) во входящей односторонней завершающей линии связи СЧ, предназначенной для BLV и OI³³.

Коды в таблице VIII.4 используются для определения событий и сигналов для пакета "MT" для входящих односторонних "завершающих транкинговых линий связи СЧ", используемых для BLV и OI.

Таблица VIII.4/J.171.1 – Коды, используемые для определения пакетов событий и сигналов MT

Код	Описание	Событие	Сигнал	Дополнительная информация
ans	Ответ вызова	–	BR	
bz	Тональный сигнал занятости	–	TO	Выдержка времени = 30 секунд
hf	Сигнал отбоя	–	BR	
inf	Информация о цифрах	√		
oc	Операция завершена	√	–	
of	Ошибка операции	√	–	
oi	Оператор прерывания	√	–	
pst	Постоянный тональный сигнал	–	TO	Выдержка времени бесконечная
rel	Разъединение вызова	P	BR	
res	Возобновление вызова	–	BR	
rlc	Разъединение завершено	P, S	BR	
ro	Тональный сигнал освобождения линии	–	TO	Выдержка времени = 30 секунд
sup	Установление вызова	P	–	
sus	Приостановление вызова	–	BR	

Определения индивидуальных событий и сигналов следующие:

ПРИМЕЧАНИЕ. – Для более подробного рассмотрения технических деталей используемых тональных сигналов см. Рек. МСЭ-Т E.180/Q.35.

Ответ вызова (ans): Сигнал ответа вызова информирует конечную точку о том, что проверяемый абонент уже находится в состоянии "телефонная трубка положена". Конечная точка ожидает прохождения наблюдаемого ответа в OSS.

Тональный сигнал занятости (bz): Станция занята.

Сигнал отбоя (hf): Данный сигнал указывает на то, что проверяемый абонент дал сигнал отбоя.

Информация о цифрах (inf (<inf-digits>)): Используются во входящей транкинговой линии MF для обозначения полученных цифр. Значение параметра <inf-digits> накапливает все цифры и включает цифровой ограничитель, т. е. ST, ST', ST" или ST'''.

Значение параметра <inf-digits> – это список цифр MF, отделенных запятыми:

MF₁, MF₂, ..., MF_n,

³³ Следует отметить, что когда услуги оператора обеспечиваются внесетевым провайдером, OSS не может иметь доступ к базам данных абонентов, для определения разрешены или нет BLV and OI.

где каждая из цифр MF_i будет одной из следующих символов цифр MF из таблицы VIII.5:

Таблица VIII.5/J.171.1 – Цифровые символы MF

Символ	Цифра MF	Символ	Цифра MF
0	MF 0	K0	MF K0 или KP
1	MF 1	K1	MF K1
2	MF 2	K2	MF K2
3	MF 3	S0	MF S0 или ST
4	MF 4	S1	MF S1
5	MF 5	S2	MF S2
6	MF 6	S3	MF S3
7	MF 7	K0	MF K0 или KP
8	MF 8		
9	MF 9		

Таким образом, пример сигнала или события может выглядеть так:

`inf(k0, 5,5,5,1,2,3,4, s0)`

Пример, где встроенный цифровой таймер истекает после 5,5,5, может выглядеть следующим образом:

`inf(k0, 5,5,5)`

Операция завершена (oc): См. определение "операция завершена" в пакете линии ISUP.

Ошибка операции (of): См. определение "ошибка операции" в пакете линии ISUP.

Оператор прерывания (oi): Событие оператора прерывания появляется, когда оператор предпринимает попытку прервать вызов и сгенерировать тональный сигнал "оператор прерван". Поскольку нет стандартных тональных сигналов, определенных для этого, то событие, определяемое здесь, появляется, когда имеющийся уровень энергии обнаруживается в линии, соответствующей переходу из шума в линии к речи или тональным сигналам. Следует отметить, что таким образом не возможно обнаружить переход назад к шуму в линии из речи/тональных сигналов.

Постоянный тональный сигнал (pst): Разъединение вызова (rel): MGC может использовать разъединение сигнала для того, чтобы разъединить вызов³⁴. В этом случае, сигнал разъединения не может быть параметризован.

В ответ конечная точка может использовать событие для информирования MGC о том, что вызов разъединился – в этом случае, событие может быть параметризовано одной из кодов причин из таблицы VIII.6, указывающего на причину разъединения.

³⁴ Следует отметить, что проверяемый оператор обычно контролирует разъединение завершенных непроверяемых соединений и, таким образом, обычно следует использовать сигнал приостановления.

Таблица VIII.6/J.171.1 – Коды причин разъединения вызова

Код причины	Причина
0	Нормальное разъединение
3	Нет маршрута к адресу получателя
8	Внеочередное занятие линии
19	Нет ответа
21	Вызов отклонен
27	Адрес получателя неисправен
28	Недопустимый формат номера (например, адрес неполный)
38	Сеть неисправна
111	Ошибка протокола/сигнализации, неопределенный протокол/сигнализация (например, выдержка времени)

Возобновление вызова (res): Данный сигнал указывает на то, что проверяемый абонент возобновил вызов, т. е. абонент перешел в состоянии "телефонная трубка снята".

Разъединение завершено (rlc): Конечная точка и MGC используют событие/сигнал "разъединение завершено" для подтверждения о том, что вызов был разъединен и линия доступна для другого вызова.

Установка вызова (sup): Событие "sup" используется для обозначения, что входящий вызов прибывает (соответствующее событие – входящее событие "телефонная трубка снята"). Событие не обеспечивается параметрами.

Приостановление вызова (sus): Данный сигнал указывает на то, что проверяемый абонент приостановил вызов, т. е. абонент перешел в состоянии "телефонная трубка положена".

БИБЛИОГРАФИЯ

- *Bellcore Notes on the Networks*, Bellcore, SR-2275.
- *Compatibility Information for Feature Group D Switched Access Service*, Bellcore, TR-NPL-000258, Issue 1, October 1985.
- *Interoffice LATA Switching Systems Generic Requirements (LSSGR): Verification Connections (25-05-0903)*, Bellcore, TR-TSY-000531, Issue 2, July 1987.
- *Signalling for Analog Interfaces*, Bellcore, LSSGR GR-506-CORE, Issue 1, June 1996.
- *Switching System Generic Requirements for Call Control Using the Integrated Services Digital Network User Part (ISDNUP)*, Bellcore, LSSGR GR-317-CORE, Issue 2, December 1997.
- *Custom Call-Handling Features (FSD 80 Series)*, Bellcore, OSSGR GR-1176-CORE, Issue 1, March 1999.
- IETF RFC 1827 (1995), *IP Encapsulating Security Payload (ESP)*.
- IETF RFC 2974 (Experimental, 2000), *Session Announcement Protocol*.
- *RTP Parameters*, <http://www.iana.org/assignments/rtp-parameters>.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевого протокола и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи