



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

H.248.1

(09/2005)

СЕРИЯ H: АУДИОВИЗУАЛЬНЫЕ И
МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб – Процедуры
связи

Протокол управления шлюзом: Версия 3

Рекомендация МСЭ-Т H.248.1

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Н
АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	Н.100–Н.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ УСЛУГ	
Общие положения	Н.200–Н.219
Мультиплексирование и синхронизация при передаче	Н.220–Н.229
Системные аспекты	Н.230–Н.239
Процедуры связи	Н.240–Н.259
Кодирование движущихся видеоизображений	Н.260–Н.279
Сопутствующие системные аспекты	Н.280–Н.299
Системы и окончное оборудование для аудиовизуальных услуг	Н.300–Н.349
Архитектура услуг справочника для аудиовизуальных и мультимедийных услуг	Н.350–Н.359
Качество архитектуры обслуживания для аудиовизуальных и мультимедийных услуг	Н.360–Н.369
Дополнительные услуги для мультимедиа	Н.450–Н.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	Н.500–Н.509
Мобильность для мультимедийных систем и услуг серии Н	Н.510–Н.519
Приложения и услуги мобильной мультимедийной совместной работы	Н.520–Н.529
Защита мобильных мультимедийных систем и услуг	Н.530–Н.539
Защита приложений и услуг мобильной мультимедийной совместной работы	Н.540–Н.549
Процедуры мобильного взаимодействия	Н.550–Н.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	Н.560–Н.569
ШИРОКОПОЛОСНЫЕ И МУЛЬТИМЕДИЙНЫЕ TRIPLE-PLAY УСЛУГИ	
Предоставление широкополосных мультимедийных услуг по VDSL	Н.610–Н.619

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

Рекомендация МСЭ-Т Н.248.1

Протокол управления шлюзом: Версия 3

Резюме

В настоящей Рекомендации, с целью обеспечения лучшей масштабируемости, функция шлюза Н.323, определенная в Рекомендации Н.246, разбивается на функциональные компоненты, а также определяются протоколы, которые используются этими компонентами для связи. Это позволяет обеспечить высокую степень масштабируемости при реализации шлюзов, а также способствует использованию таких возможностей широко распространенных сетей с коммутацией каналов (SCN), как коммутаторы SS7. Кроме того, открывается возможность создания шлюзов Н.323 из компонентов нескольких производителей, распределенных среди нескольких физических платформ. Целью данной Рекомендации является дополнение возможностей, определяемых в настоящее время для систем Н.323, а также она предназначена для обеспечения новых путей выполнения операций, уже поддержанных в Рекомендации МСЭ-Т Н.323.

В данной Рекомендации содержатся некоторые дополнения к Рекомендации МСЭ-Т Н.248.1 версия 2:

- возможность определять свойства контекста через комплекты;
- свойство контекста IEPS;
- флаг для индикации того, что MG имеет завершения OutOfService для отчета во время регистрации;
- новый комплект сегментации сообщения и процедуры для несегментированного транспорта;
- усовершенствованные требования определения комплекта и новый шаблон комплекта;
- усовершенствованные требования определения профиля и новый шаблон профиля;
- добавление статистики на уровне потока;
- добавление идентификатора запроса сигнала для дифференциации похожих сигналов внутри SignalList;
- добавление параметра базового сигнала для индикации того, в каком направлении воспроизводить сигнал;
- добавление двух новых типов Topology;
- добавление таймера межсигнальной задержки для сигналов в SignalList;
- добавление нового структурного элемента ContextIDList для командных откликов;
- добавление структурного элемента TerminationIDList для команд и откликов;
- усовершенствованные процедуры ServiceChange;
- добавление для MGC возможности регулирования скорости, с которой он получает уведомления;
- добавление возможности добавлять условия фильтра для проверки запросов.

Источник

Рекомендация МСЭ-Т Н.248.1 утверждена 13 сентября 2005 года 16-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации 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 получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

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

СОДЕРЖАНИЕ

	Стр.
1	Сфера применения 1
2	Справочные документы 1
2.1	Нормативные справочные документы 1
2.2	Информативные справочные документы 3
3	Определения 4
4	Сокращения 5
5	Соглашения по терминам 6
6	Модель соединения 6
6.1	Контексты 7
6.2	Завершения 8
6.3	Принципы обобщения 14
7	Команды 16
7.1	Дескрипторы 16
7.2	Программный интерфейс приложения команд 37
8	Транзакции 52
8.1	Общие параметры 53
8.2	Программный интерфейс приложения транзакций 54
8.3	Сообщения 56
9	Передача 57
9.1	Порядок выполнения команд 57
9.2	Защита от перегрузки при перезапуске 58
9.3	Защита от перегрузки Notify 59
10	Вопросы безопасности 59
10.1	Защита соединений протокола 59
10.2	Промежуточная схема АН 59
10.3	Защита медиасоединений 60
11	Интерфейс управления MG-MGC 60
11.1	Множество Виртуальных MG 60
11.2	Начальный запуск 61
11.3	Согласование версии протокола 61
11.4	Выход MG из строя 62
11.5	Выход MGC из строя 62
11.6	Контроль связи управления MGC-MG 63
12	Определение комплекта 63
12.1	Указания по определению комплектов 64
12.2	Указания по определению параметров событий и сигналов 67
12.3	Идентификаторы 68
12.4	Регистрация комплекта 68
13	Определение профиля 68

	Стр.
14	68
14.1	68
14.2	69
14.3	69
14.4	69
Приложение А – Двоичное кодирование протокола.....	70
А.1	70
А.2	71
А.3	88
Приложение В – Текстовое кодирование протокола	89
В.1	89
В.2	89
В.3	104
В.4	104
Приложение С – Тэги для свойств потока медиа	105
С.1	105
С.2	106
С.3	106
С.4	107
С.5	109
С.6	109
С.7	110
С.8	111
С.9	112
С.10	120
С.11	120
С.12	121
Приложение D – Передача по IP	121
D.1	121
D.2	125
Приложение E – Основные комплекты	126
E.1	126
E.2	128
E.3	131
E.4	132
E.5	135
E.6	137
E.7	139
E.8	140
E.9	141
E.10	144

	Стр.
E.11 Комплект Network.....	146
E.12 Комплект RTP	149
E.13 Комплект TDM Circuit.....	151
E.14 Комплект Segmentation.....	152
E.15 Комплект Notification Behaviour.....	155
Приложение F – Процедуры ServiceChange.....	159
F.1 Введение	159
F.2 Определение связи управления	161
F.3 События, приводящие к процедурам ServiceChange.....	161
F.4 Описание элемента ServiceChange	165
F.5 Использование параметров ServiceChange.....	168
F.6 ServiceChange в сравнении с TerminationState	170
Дополнение I – Примеры потоков соединений	171
I.1 Соединение частный шлюз – частный шлюз	171
Дополнение II – Шаблон комплекта H.248	180
Дополнение III – Шаблон определения профиля H.248	183

Рекомендация МСЭ-Т Н.248.1

Протокол управления шлюзом: Версия 3

1 Сфера применения

В данной Рекомендации определяются протоколы, используемые между элементами физически распределенного мультимедийного шлюза. С системной точки зрения нет функциональных различий между шлюзом, разделенным на компоненты, потенциально распределенными в более чем одном физическом устройстве, и единым шлюзом, описанным в Рекомендации МСЭ-Т Н.246. Эта Рекомендация не определяет, как работают шлюзы, многоточечные управляющие устройства или устройства интерактивного голосового ответа (IVR). Вместо этого в ней создается общая инфраструктура, подходящая для этих приложений.

Интерфейсы пакетной сети могут включать IP, ATM или, возможно, и другие. Данные интерфейсы будут поддерживать разнообразные системы сигнализации сети с коммутацией каналов (SCN), включающие тоновую сигнализацию, ЦСИС, ISUP, QSIG и GSM. Там, где это применимо, будет обеспечена поддержка национальных разновидностей этих сигнальных систем.

Продукты, заявляющие соответствие версии 1 Рек. МСЭ-Т Н.248.1, должны соответствовать всем обязательным требованиям Рек. МСЭ-Т Н.248.1, первоначально утвержденной в 06/2000 и обновленной в 03/2002.

Продукты, заявляющие соответствие версии 2 Рек. МСЭ-Т Н.248.1, должны соответствовать всем обязательным требованиям Рек. МСЭ-Т Н.248.1, первоначально утвержденной в 05/2002 и обновленной в 03/2004.

Продукты, заявляющие соответствие данной Рекомендации, должны соответствовать всем обязательным требованиям Рек. МСЭ-Т Н.248.1, утвержденной в 09/2005.

Продукты должны обозначать версию используемого протокола, используя ServiceChangeVersion со значением '1' для ссылки на Рек. МСЭ-Т Н.248.1 (03/2002), '2' для ссылки на Рек. МСЭ-Т Н.248.1 Исправление 1 (03/2004) и '3' для ссылки на Рек. МСЭ-Т Н.248.1 (09/2005).

2 Справочные документы

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

2.1 Нормативные справочные документы

- ITU-T Recommendation E.106 (2003), *International Emergency Preference Scheme (IEPS) for disaster relief operations*.
- ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.
- ITU-T Recommendation H.235.0 (2005), *H.323 Security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems*.
- ITU-T Recommendation H.245 (2005), *Control protocol for multimedia communication*.
- ITU-T Recommendation H.246 (1998), *Interworking of H-series multimedia terminals with H-series multimedia terminals and voice/voiceband terminals on GSTN and ISDN*.

- ITU-T Recommendation H.248.4 (2000), *Gateway control protocol: Transport over Stream Control Transmission Protocol (SCTP)*, plus Corrigendum 1 (2004).
- ITU-T Recommendation H.248.5 (2000), *Gateway control protocol: Transport over ATM*.
- ITU-T Recommendation H.248.8 (2005), *Gateway control protocol: Error code and service change reason description*.
- ITU-T Recommendation H.248.14 (2002), *Gateway control protocol: Inactivity timer package*.
- ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems*.
- ITU-T Recommendation I.363.1 (1996), *B-ISDN ATM Adaptation Layer specification: Type 1 AAL*.
- ITU-T Recommendation I.363.2 (2000), *B-ISDN ATM Adaptation Layer specification: Type 2 AAL*.
- ITU-T Recommendation I.363.5 (1996), *B-ISDN ATM Adaptation Layer specification: Type 5 AAL*.
- ITU-T Recommendation I.366.1 (1998), *Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2*.
- ITU-T Recommendation I.366.2 (2000), *AAL type 2 service specific convergence sublayer for narrow-band services*, plus Corrigendum 1 (2002).
- ITU-T Recommendation I.371 (2004), *Traffic control and congestion control in B-ISDN*.
- ITU-T Recommendation Q.763 (1999), *Signalling System No. 7 – ISDN user part formats and codes*, plus Amendment 3 (2004).
- ITU-T Recommendation Q.765.5 (2004), *Signalling System No. 7 – Application transport mechanism: Bearer Independent Call Control (BICC)*.
- ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*, plus Amendment 1 (2002): *Extensions for the support of digital multiplexing equipment*.
- ITU-T Recommendation Q.2630.1 (1999), *AAL type 2 signalling protocol – Capability Set 1*.
- ITU-T Recommendation Q.2931 (1995), *Digital subscriber signalling system No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control*, plus Amendment 4 (1999).
- ITU-T Recommendation Q.2941.1 (1997), *Digital subscriber signalling system No. 2 – Generic identifier transport*.
- ITU-T Recommendation Q.2961.1 (1995), *Digital subscriber signalling system No. 2 – Additional traffic parameters: Additional signalling capabilities to support traffic parameters for the tagging option and the sustainable call rate parameter set*.
- ITU-T Recommendation Q.2961.2 (1997), *Digital subscriber signalling system No. 2 – Additional traffic parameters: Support of ATM transfer capability in the broadband bearer capability information element*, plus Corrigendum 1 (1999).
- ITU-T Recommendation Q.2965.1 (1999), *Digital subscriber signalling system No. 2 – Support of Quality of Service classes*, plus Amendment 1 (2000).
- ITU-T Recommendation Q.2965.2 (1999), *Digital subscriber signalling system No. 2 – Signalling of individual Quality of Service parameters*.
- ITU-T Recommendation V.76 (1996), *Generic multiplexer using V.42 LAPM-based procedures*, plus Corrigendum 1 (2005).
- ITU-T Recommendation X.213 (2001) | ISO/IEC 8348:2002, *Information technology – Open Systems Interconnection – Network service definition*.

- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*, plus Amendment 2 (2004): *Alignment with changes made to Rec. X.660 | ISO/IEC 9834-1 for identifiers in object identifier value notation.*
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, plus Amendment 1 (2003): *Support for EXTENDED-XER.*
- ISO/IEC 10646 (2003), *Information technology – Universal Multiple-Octet Coded character Set (UCS).*
- ATM Forum (1996), *ATM User-Network Interface (UNI) Signalling Specification – Version 4.0.*
- IETF RFC 1006 (1987), *ISO Transport Service on top of the TCP, Version 3.*
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF.*
- IETF RFC 2327 (1998), *SDP: Session Description Protocol.*
- IETF RFC 2402 (1998), *IP Authentication Header.*
- IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP).*

2.2 Информативные справочные документы

- ITU-T Recommendation E.180/Q.35 (1998), *Technical characteristics of tones for the telephone service.*
- ITU-T Recommendation G.711 (1988), *Pulse code modulation (PCM) of voice frequencies.*
- ITU-T Recommendation H.221 (2004), *Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices.*
- ITU-T Recommendation H.223 (2001), *Multiplexing protocol for low bit rate multimedia communication.*
- ITU-T Recommendation H.226 (1998), *Channel aggregation protocol for multilink operation on circuit-switched networks.*
- ITU-T Recommendation Q.724 (1998), *Telephone user part signalling procedures*, plus Amendment 1 (1993).
- ITU-T Recommendation Q.764 (1999), *Signalling System No. 7 – ISDN user part signalling procedures*, plus Amendment 3 (2004).
- ITU-T Recommendation Q.1902.4 (2001), *Bearer Independent Call Control protocol – (Capability Set 2): Basic call procedures*, plus Amendment 2 (2004).
- IETF RFC 768 (1980), *User Datagram Protocol.*
- IETF RFC 791 (1981), *Internet protocol.*
- IETF RFC 793 (1981), *Transmission control protocol.*
- IETF RFC 1661 (1994), *The Point-to-Point Protocol (PPP).*
- IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol.*
- IETF RFC 2460 (1998), *Internet Protocol, Version 6 (IPv6) Specification.*
- IETF RFC 2805 (2000), *Media Gateway Control Protocol Architecture and Requirements.*

- IETF RFC 3261 (2002), *SIP: Session Initiation Protocol*.
- IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.
- IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control*.

3 Определения

В данной Рекомендации используются следующие термины:

3.1 шлюз доступа: Тип шлюза, обеспечивающий интерфейс "пользователь-сеть" (UNI), такой, как ISDN.

3.2 дескриптор: синтаксический элемент протокола, группирующий связанные свойства. Например, свойства потока медиа на MG можно установить при помощи MGC путем включения соответствующего дескриптора в команду.

3.3 медиашлюз (MG): медиашлюз преобразует медиаданные, предоставляемые в одном типе сети, в формат, требуемый в другом типе сети. Например, MG может служить завершением каналов – носителей в сетях с коммутацией каналов (например, DS0) и потоков медиа в сетях с коммутацией пакетов (например, потоков RTP в IP сети). Такой шлюз может обрабатывать аудио, видео и T.120 по отдельности или в любой комбинации, и будет способен выполнить полнодуплексное преобразование медиа. Шлюз MG может также проигрывать аудио/видео сообщения и выполнять другие функции IVR, или может осуществлять медиаконференции.

3.4 контроллер медиашлюза (MGC): контролирует те части состояния соединения, которые относятся к управлению соединением для каналов медиа в MG.

3.5 устройство управления многосторонней связью (MCU): объект, контролирующий настройку и координацию многопользовательской конференции, что обычно включает в себя обработку аудио, видео и данных.

3.6 частный шлюз: шлюз, обеспечивающий межсетевой обмен между аналоговой линией и сетью с коммутацией пакетов. Частный шлюз обычно содержит одну или две аналоговых линии и располагается в помещении клиента.

3.7 шлюз сигнализации FAS SCN: данная функция содержит интерфейс сигнализации SCN, завершающий SS7, ISDN или другие звенья сигнализации, причем канал управления соединением и каналы – носители располагаются в одном и том же физическом пространстве.

3.8 шлюз сигнализации NFAS SCN: данная функция содержит интерфейс сигнализации SCN, завершающий SS7 или другие звенья сигнализации, причем каналы управления соединением отделены от каналов – носителей.

3.9 поток: двунаправленный поток медиа или команд управления получаемый/ отправляемый медиашлюзом как часть соединения или конференции.

3.10 канал: соединительный канал между двумя коммутационными узлами такими, как DS0 на линии T1 или E1.

3.11 канальный шлюз: шлюз между сетью с коммутацией каналов и сетью с коммутацией пакетов, обычно являющийся завершением большого числа цифровых каналов.

4 Сокращения

В данной Рекомендации используются следующие сокращения:

AAD	Средняя задержка подтверждения
AAL	Уровень адаптации ATM
ADEV	Среднее отклонение
ALF	Кадрирование прикладного уровня
ATM	Асинхронный режим передачи
C	Контекст
CAS	Сигнализация по выделенному каналу
DNS	Система доменных имен
DTMF	Двухтональный многочастотный набор телефонного номера
FAS	Сигнализация, связанная с устройством
GSM	Глобальная система связи с подвижными объектами, система GSM
GW	Шлюз
IANA	Центр по присвоению номеров Internet, организация IANA, замещенная ICANN
ICANN	Интернет-корпорация по присвоению имен и адресов
IEPS	Международная приоритетная схема для оказания помощи в чрезвычайных ситуациях
IP	Протокол Интернет
IS	В рабочем состоянии
ISUP	Пользовательская часть сети ISDN
IVR	Интерактивное речевое взаимодействие
MG	Медиашлюз
MGC	Контроллер медиашлюза
MWD	Максимальная задержка ожидания
NFAS	Сигнализация, несвязанная с устройством
OoS	Неисправный
PRI	Интерфейс основного уровня
КТСОП	Телефонная коммутируемая сеть общего пользования
QoS	Качество обслуживания
RTP	Транспортный протокол режима реального времени
SC	Смена обслуживания
SCN	Сеть с коммутацией каналов
SG	Шлюз сигнализации
SS7	Система сигнализации № 7
T, Term	Завершение

5 Соглашения по терминам

В данной Рекомендации слово "должен" обозначает обязательное требование, а слово "следует" обозначает рекомендуемую, но необязательную характеристику или процедуру. Слово "может" обозначает необязательное действие без выражения предпочтительности.

6 Модель соединения

Модель соединения для данного протокола описывает логические элементы, или объекты, внутри медиашлюза, которые может контролировать контроллер медиашлюза. Основными абстракциями, используемыми в модели соединения, являются завершения и контексты.

Завершение является источником и / или приемником одного или более потоков. В мультимедийной конференции завершение может быть мультимедийным и служить источником или приемником многочисленных потоков медиа. Параметры потока медиа, а также параметры носителя инкапсулированы в завершение.

Контекст является взаимосвязью между совокупностью завершений. Существует особый тип контекста, Нулевой (NULL) контекст, который содержит все завершения, не присутствующие в других контекстах и поэтому не соединенные ни с одним завершением. Например, в распределенном шлюзе доступа все свободные линии представлены завершениями в Нулевом (NULL) контексте.

Нижеследующие рисунки являются графическим отображением данных понятий. В схеме на рисунке 1 даны несколько примеров, однако она не является исчерпывающей иллюстрацией. Прямоугольник со звездочкой в каждом из контекстов отображает логическое соединение завершений, подразумеваемых контекстом.

Медиашлюз

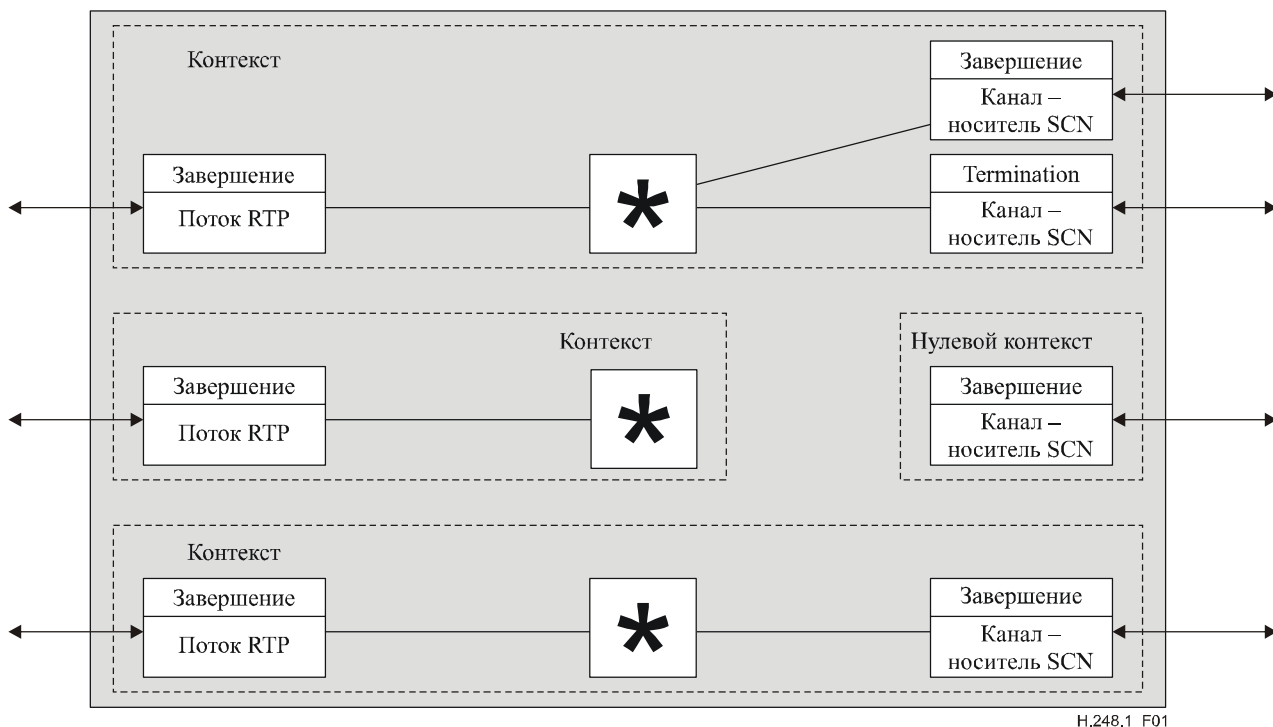


Рисунок 1/Н.248.1 – Пример модели соединения Н.248.1

На рисунке 2 представлен пример реализации сценария ожидания соединения в распределенном шлюзе доступа, иллюстрирующий перемещение завершения между контекстами. Завершения T1 и T2 принадлежат Контексту C1 в двухстороннем аудиосоединении. Второе аудиосоединение ожидает T1 от Завершения T3. Само T3 находится в Контексте C2. T1 принимает вызов от T3, переводя T2 в режим ожидания. В результате T1 перемещается в Контекст C2, как показано на рисунке 3.

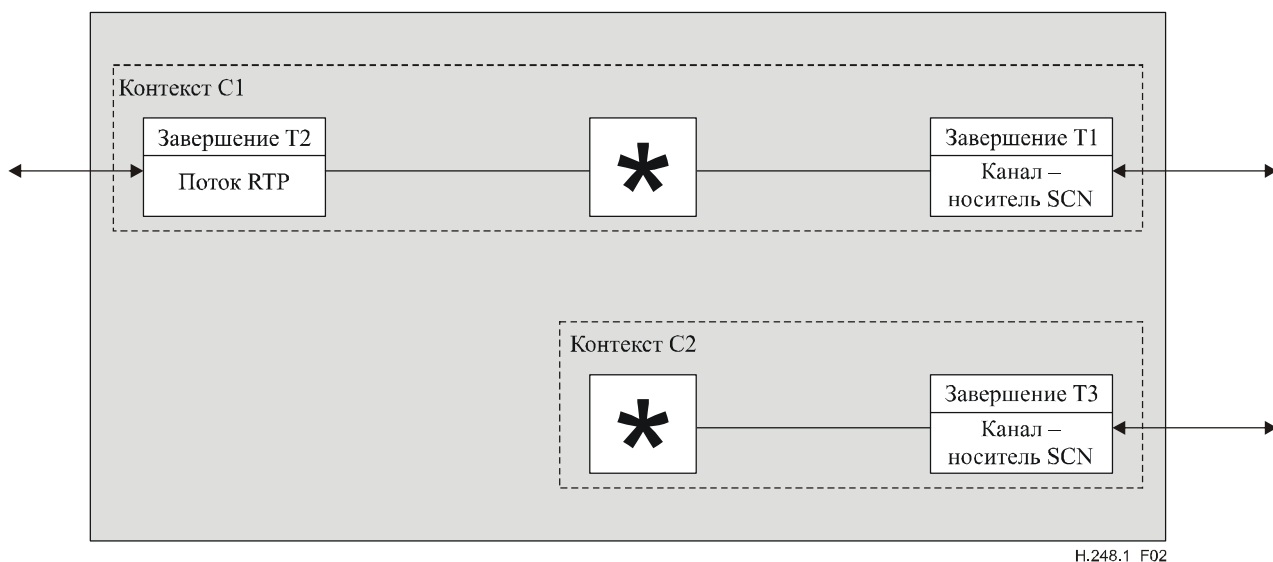


Рисунок 2/Н.248.1 – Пример сценария ожидания соединения/перемещения, применяемого к T1

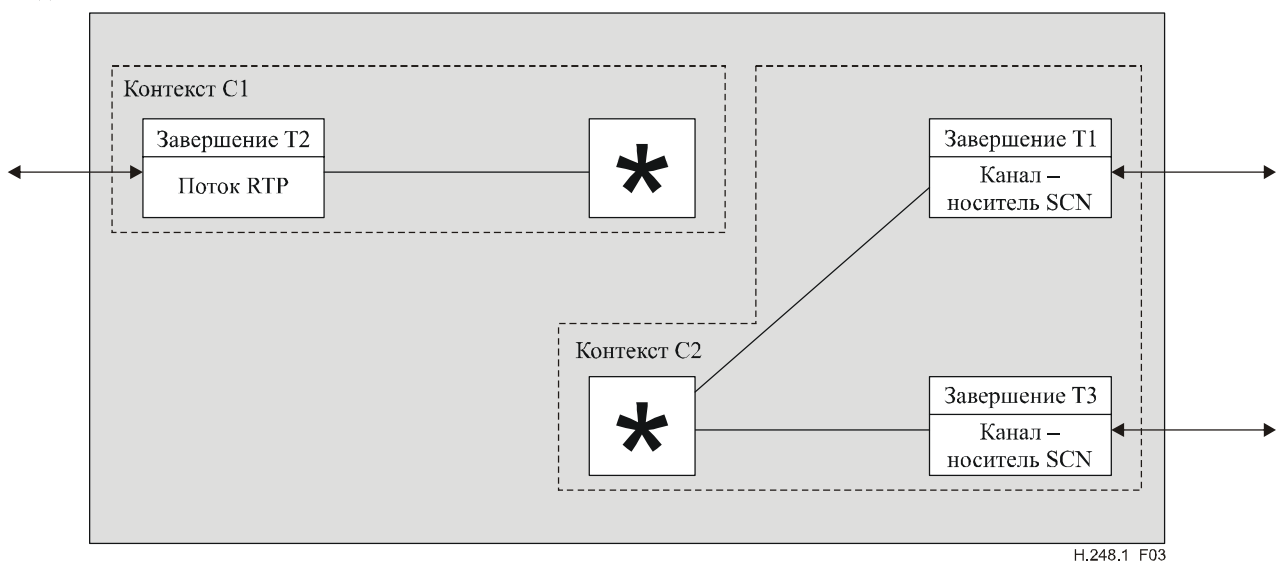


Рисунок 3/Н.248.1 – Пример сценария ожидания соединения/ответ от T1

6.1 Контексты

Контекст представляет собой соединение между несколькими завершениями. Контекст описывает топологию (кто кого слышит/видит) и микширование носителей и/или параметры коммутации, если в соединении участвуют более двух завершений.

Существует особый вид контекста, называемый *нулевым (NULL)* контекстом. Он содержит завершения, которые не присутствуют в других контекстах и поэтому не связаны ни с одним завершением. Завершения в нулевом контексте могут иметь свои проверенные или измененные параметры и могут иметь обнаруженные на них события.

В общих чертах для добавления завершений в контексты используется команда Add. Если MGC не определяет существующий контекст, к которому следует прибавить завершение, MG создает новый контекст. Завершение можно вычесть из контекста при помощи команды Subtract, завершение можно переместить из одного контекста в другой при помощи команды Move. Завершение может существовать только в одном контексте в один и тот же промежуток времени.

Максимальное число завершений в контексте является свойством MG. Медиашлюзы, предлагающие только непосредственное соединение, могут разрешить не более двух завершений на один контекст. Медиашлюзы, поддерживающие многоточечные конференции, могут разрешить три или более завершений на контекст.

6.1.1 Качественные признаки и дескрипторы контекста

Качественными признаками контекстов являются:

- Идентификатор ContextID.
- Дескриптор топологии (кто кого слышит/видит)
Топология контекста описывает поток медиа между завершениями внутри контекста. В противоположность этому, свойство Mode завершения ("SendOnly"/"RecvOnly"/...) описывает поток медиа на входе/выходе медиашлюза.
- Приоритет используется в контексте для того, чтобы обеспечить шлюз MG информацией об определенном порядке обработке контекста. В некоторых ситуациях (например, перезапуск) контроллер MGC может также использовать приоритет для плавного автоматического управления порядком трафика, когда необходимо одновременно обработать большое число контекстов. Приоритет 0 является низшим приоритетом, а приоритет 15 соответствует самому высокому приоритету
- Также предоставляется индикатор для аварийного вызова для осуществления обработки в порядке очереди в MG.
- Для достижения характеристик и методов E.106 предоставляется индикатор для вызова IEPS.
- Дескриптор ContextAttribute, позволяющий определять дополнительные качественные признаки контекста, используя алгоритм расширения пакетов. (см. п. 7.1.19).

6.1.2 Создание, удаление и изменение контекстов

Данный протокол можно (в неявном виде) использовать для создания контекстов и изменения параметров существующих контекстов. Протокол содержит команды для добавления завершений к контекстам, вычитания их из контекстов и перемещения завершений между контекстами. Контексты неявно удаляются, когда вычитается или перемещается последнее оставшееся завершение.

6.2 Завершения

Завершение является логическим элементом шлюза MG, представляющим собой источник и/или приемник медиапоточков и/или потоков управления. Завершение описывается рядом характерных свойств, группирующихся в набор дескрипторов, которые включаются в команды. Завершения имеют уникальные идентификаторы (TerminationIDs), присвоенные шлюзом MG во время их создания.

Завершения, представляющие физические объекты, имеют полупостоянное существование. Например, завершение, представляющее канал TDM, может существовать так долго, как это обеспечивается шлюзом. Завершения, представляющие непродолжительные информационные потоки, такие как потоки RTP, обычно будут существовать только в течение их использования.

Непродолжительные завершения создаются при помощи команды Add. Они разрушаются посредством команды Subtract. В противоположность этому, когда физическое завершение добавляется или вычитается из контекста, оно поступает из нулевого контекста или в нулевой контекст, соответственно.

К завершениям могут применяться сигналы (см. п. 7.1.11). Можно запрограммировать завершения на распознавание событий, возникновение которых может вызвать отсылку MGC уведомляющих сообщений или определенные действия MG. На завершении может проводиться сбор статистической информации. Статистические данные сообщаются MGC по запросу (посредством команды AuditValue, см. п. 7.2.5) и тогда, когда завершение прекращает существовать или возвращается в нулевой контекст в результате выполнения команды Subtract.

Мультимедийные шлюзы могут обрабатывать мультиплексированные потоки медиа. Например, в Рек. МСЭ-Т Н.221 описывается структура кадра для многочисленных потоков медиа, мультиплексированных на нескольких цифровых каналах 64 кбит/с. В модели соединения подобный

случай обрабатывается следующим образом. Для каждого канала – носителя, несущего часть мультимплексированных потоков, существует физическое или непродолжительное "основное завершение". Основные завершения, служащие источниками/ приемниками цифровых каналов, соединены в отдельное завершение, называемое "мультимплексное завершение". Мультимплексное завершение является непродолжительным завершением, представляющим сеанс, связанный с кадром. Дескриптор Multiplex для этого завершения описывает использованное объединение каналов (мультимплексирование) (например, H.221 для сеанса H.320) и показывает, в каком порядке содержащиеся в завершении цифровые каналы объединяются в кадр.

Мультимплексные завершения могут быть каскадными (например, объединение цифровых каналов (мультимплексирование) H.226, передаваемое в объединение H.223, поддерживающее сеанс H.324).

Отдельные потоки медиа, передаваемые во время сеанса, описываются дескрипторами Stream на мультимплексном завершении. Данные потоки медиа могут быть связаны с потоками, исходящими/ принимаемыми завершениями в контексте, отличном от основных завершений, поддерживающих мультимплексное завершение. Каждое основное завершение поддерживает только один поток данных. Эти потоки данных не появляются в явном виде, как потоки на мультимплексном завершении, они скрыты от остальной части контекста.

На рисунках 4, 5 и 6 иллюстрируются типичные приложения мультимплексного завершения и дескриптора Multiplex.

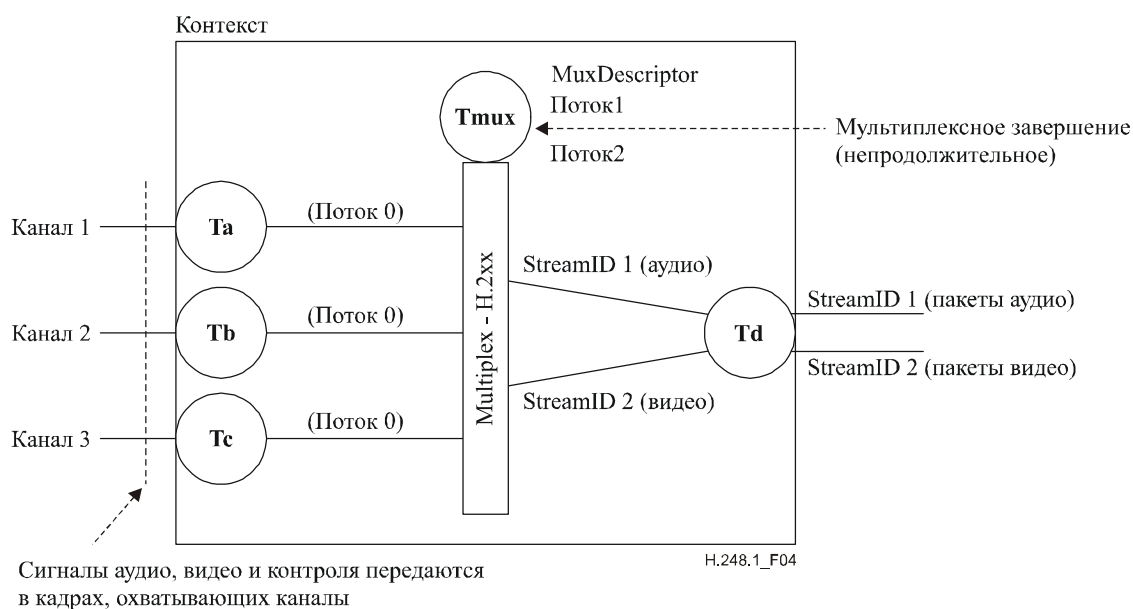


Рисунок 4/Н.248.1 – Сценарий мультимплексного завершения – Канал – пакет

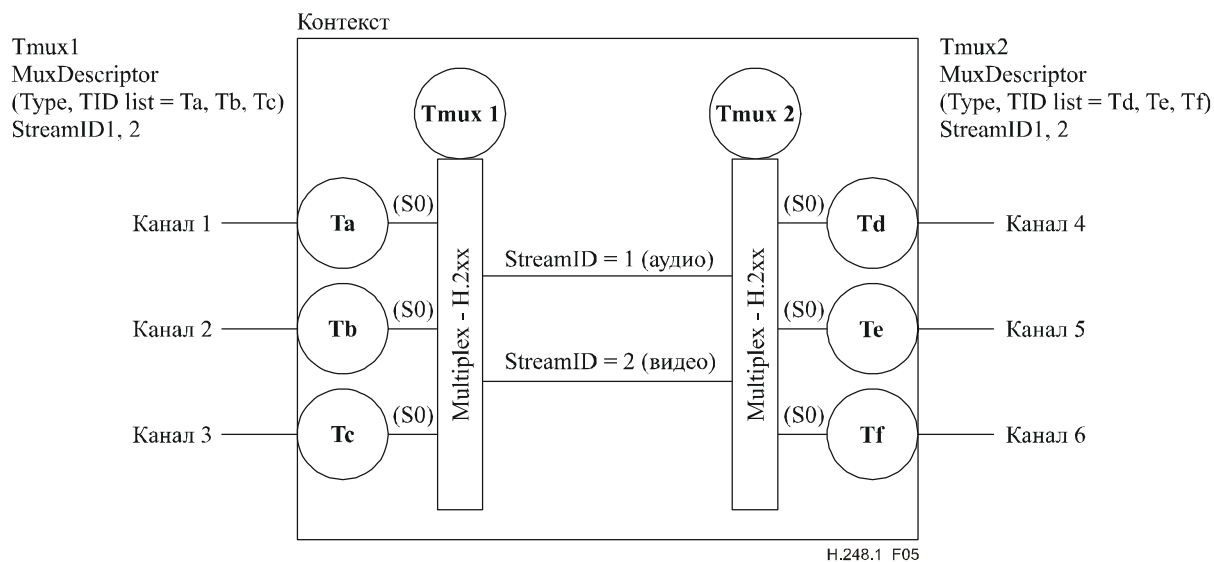


Рисунок 5/Н.248.1 – Сценарий мультиплексного завершения – Канал – канал

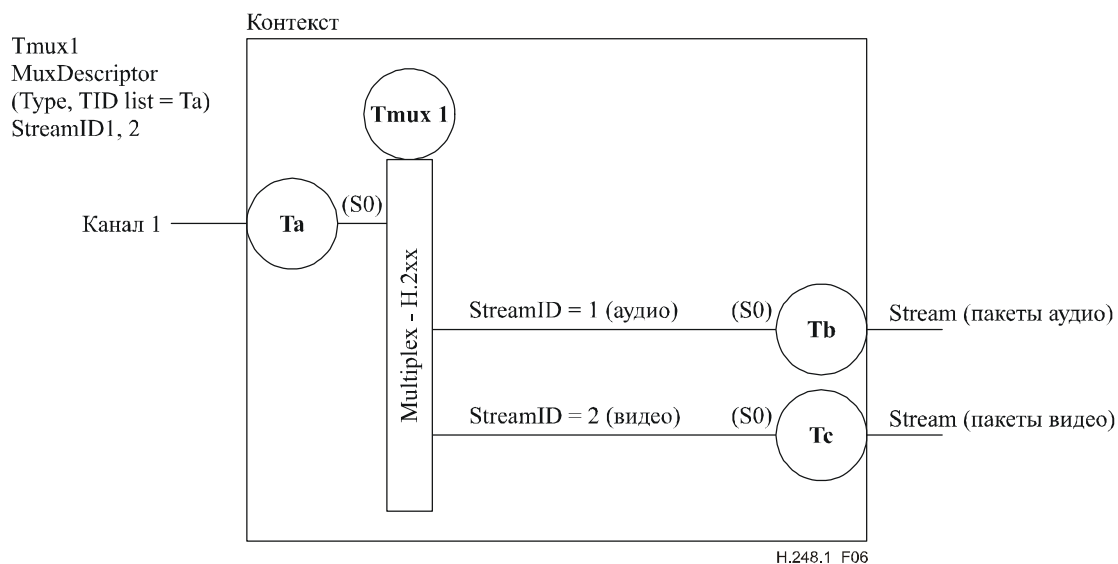


Рисунок 6/Н.248.1 – Сценарий мультиплексного завершения – Одно завершение – множество завершений

В отличие от мультиплексных завершений, описанных в предыдущем абзаце, мультиплексные основные завершения, которые представляют мультиплексированные носители такие, как носители АТМ ААЛ типа 2, не несут потоков медиа. Они присутствуют только для моделирования создания и разрушения действительного носителя. Когда необходимо создать новый мультиплексированный носитель, в контексте, созданном для этой цели, создается непродолжительное завершение. Когда завершение вычитается, мультиплексированный носитель разрушается.

6.2.1 Динамика завершения

Данный протокол может быть использован для создания новых завершений и изменения значений свойств существующих завершений. Эти изменения включают возможность добавления или удаления событий и/или сигналов. Свойства завершений, события и сигналы описываются в последующих подпунктах. МГС может лишь освобождать/изменять завершения и ресурсы, представляемые завершением, и находящиеся в нулевом контексте, или которые ранее были получены, например, при помощи команды Add.

6.2.2 Идентификаторы TerminationID

Ссылки на завершения осуществляются посредством идентификатора TerminationID, являющегося произвольной схемой, выбранной шлюзом MG.

Идентификаторы TerminationID физических завершений предоставляются в медиашлюзе. Можно выбрать, чтобы идентификаторы TerminationID имели структуру. Например, идентификатор TerminationID может состоять из канальной группы и канала в пределах группы.

С идентификаторами TerminationIDs используются два типа групповых символов замены. Этими символами являются ALL и CHOOSE. Первый символ применяется для одновременного обозначения нескольких завершений, а второй используется для указания медиашлюзу, что он должен выбрать завершение, удовлетворяющее частично заданному идентификатору завершения. Это позволяет контроллеру MGC, например, дать команду шлюзу MG относительно выбора канала в пределах канальной группы.

Идентификаторы TerminationID можно также задать списком. Рекомендуется использовать список TerminationIDList в случаях, когда иерархическая организация идентификатора TerminationID невозможна и не требуется для отправки отдельных команд для каждого TerminationID.

6.2.3 Комплекты

Различные типы шлюзов могут реализовывать завершения со значительно различающимися характеристиками. Различия в завершениях сглаживаются в протоколе посредством разрешения завершениям иметь дополнительные свойства, события, сигналы и статистику, реализуемые шлюзами MG.

Для достижения взаимодействия между MG и MGC, такие дополнительные свойства группируются в комплекты, и обычно завершение реализует набор таких комплектов. Более подробную информацию по определению комплектов можно найти в разделе 12. MGC может осуществлять проверку завершения для того, чтобы определить, какие комплекты оно реализует.

Ссылки на свойства, события, сигналы и статистику, определенные в комплектах, а также их параметры, осуществляются посредством идентификаторов (IDs). Идентификаторы масштабируются. Для каждого комплекта, идентификаторы PropertyIDs, EventIDs, SignalIDs, StatisticsIDs и ParameterIDs обладают уникальным пространством имен, и в каждом из них может быть использован один и тот же идентификатор. Два идентификатора PropertyIDs в различных комплектах также могут иметь один и тот же идентификатор, и т. д.

Для поддержки определенного комплекта шлюз MG должен распознавать все свойства, сигналы, события и статистику, определенные в комплекте. Он также должен поддерживать все параметры сигнала и события. Если набор функциональных возможностей по данным свойствам, сигналам, событиям и статистике не реализуется, MG должен отослать сообщение об ошибке 501 "Не реализуется", а не сообщение о синтаксической ошибке или ошибке unknown ID (неизвестный идентификатор) для любого из этих элементов.

MG может поддерживать подмножество значений определенного свойства или параметра, перечисленных в комплекте. Если MGC задает неподдерживаемое значение, MG должен отправить сообщение об ошибке 501 "Не реализуется".

Когда происходит расширение комплектов, ссылки на свойства, события, сигналы и статистику, определенные в основном комплекте, будут осуществляться при помощи использования либо расширенного имени комплекта, либо основного имени комплекта. Например, если Комплект А определяет Событие e1, а Комплект В расширяет Комплект А, событие В/e1 является событием завершения, реализующего Комплект В. По определению MG должен также реализовывать основной комплект, однако публикация основного комплекта в качестве разрешенного интерфейса является необязательной. Если медиашлюз публикует А, тогда об А, равно как и о В, будет сообщено в составе дескриптора Packages в AuditValue, а Событие А/e1 будет доступно на завершении. Если MG не публикует А, доступно будет лишь событие В/e1. При публикации посредством AuditValue А/e1 и В/e1 являются одним и тем же событием.

Для улучшенного взаимодействия и обратной совместимости шлюз MG может опубликовать все комплекты, поддерживаемые его завершениями, включая основные комплекты, из которых

образовываются расширенные комплекты. Исключением этого являются случаи, когда основные комплекты в явном виде определяются как "Разработанные только для расширения".

6.2.4 Свойства и дескрипторы завершения

Завершения имеют свойства. Свойства имеют уникальные идентификаторы PropertyID. Большинство свойств имеют значения, заданные по умолчанию, которые в явном виде определяются в спецификации данного протокола, либо в комплекте (см. пункт 12), либо устанавливаются посредством инициализации. Если не определено иначе, при создании нового завершения или при возвращении его в нулевой контекст свойства во всех дескрипторах кроме TerminationState и LocalControl по умолчанию выставлены на empty/"no value". При создании нового завершения или при возвращении его в нулевой контекст, данное состояние представляет "свободную" линию, канал или другой объект. Заданное по умолчанию содержание двух дескрипторов – исключений описано в пунктах 7.1.5 и 7.1.7.

Инициализация значения свойства в MG заменит любое значение, заданное по умолчанию, если оно поддерживается в спецификации данного протокола или в комплекте. Поэтому если необходимо, чтобы MGC имел полный контроль над значениями свойств завершения, при добавлении завершения в контекст контроллер должен поддерживать явно заданные значения. В качестве альтернативы для физического завершения MGC может определить любые инициализируемые значения свойства путем проверки завершения во время нахождения его в нулевом контексте.

Для завершений и свойств, характерных для потоков медиа, существует несколько общих свойств. Общие свойства также называются свойствами TerminationState. Для каждого потока медиа существуют локальные свойства и свойства получаемых и передаваемых потоков.

Свойства, не включенные в основной протокол, определяются в комплектах. Ссылки на данные свойства осуществляются при помощи имени, содержащего имя комплекта и идентификатор PropertyID. Большинство свойств имеют значения, заданные по умолчанию, содержащиеся в описании комплекта. Свойства могут быть только для чтения (read-only) или для чтения/ записи (read/write). Возможные значения свойств можно проверить, как и их текущие значения. Для свойств чтение/ запись (read/write), MGC может задавать собственные значения. Свойство может быть заявлено как "глобальное", имеющее одно значение для всех завершений, реализующих комплект. Для удобства взаимосвязанные свойства группируются в дескрипторы.

При добавлении завершения в контекст можно установить значение его свойств для чтения/ записи путем включения соответствующих дескрипторов в команду Add в качестве параметров. Таким же образом можно изменить значение свойства завершения в контексте при помощи команды Modify. Также можно изменить значение свойств, когда происходит перемещение завершения из одного контекста в другой в результате применения команды Move. В некоторых случаях дескрипторы возвращаются от команды в качестве результата.

Установка свойств на различных завершениях в одном и том же контексте в неявном виде отдает MG приказ выполнять определенные функции. Например: если на Завершении А устанавливается кодек G.711, а на Завершении В устанавливается кодек G.729, MG активирует функцию перекодирования, как только станет возможным поток медиа между двумя завершениями (т. е. путем установки на каждом завершении свойства Mode Property в состояние, отличное от "Неактивен" ("Inactive")).

ПРИМЕЧАНИЕ. – Во избежание ненужной активации ресурсов MG, MGC должен установить свойство Mode Property на "Неактивен" ("Inactive") для данного завершения и потока до определения свойств, используемых для этого потока.

Вообще если один из дескрипторов полностью отсутствует в вышеупомянутых командах, свойства в таком дескрипторе сохраняют свои прежние значения для завершения(ий), для которых предназначена команда. С другой стороны, если некоторые из свойств для чтения/записи отсутствуют в дескрипторе в команде (т. е. дескриптор описан лишь частично), данные свойства будут переустановлены на значения, заданные по умолчанию для завершения (ий), для которых предназначена команда, если комплект не предпишет другой ход действий. Для получения более подробной информации см. п. 7.1, в котором рассматриваются отдельные дескрипторы.

Вышеописанный режим работы равным образом применим к сигналам и событиям и их соответствующим параметрам. Инициализация дескрипторов Events включает в себя идентификатор RequestID, который следует использовать, и любые параметры событий, включенных в дескриптор.

Любую инициализацию, касающуюся дескрипторов Events, в MG следует продублировать в MGC во избежание ошибочных ответов на команды Notify от MG.

В нижеследующей таблице перечислены все возможные дескрипторы и их применение. Не все дескрипторы являются допустимыми в качестве входных или выходных параметров к каждой команде.

Название дескриптора	Описание
Modem	Идентифицирует тип модема и его свойства, если он применяется. (Примечание)
Mux	Описывает тип мультиплексирования для мультимедийных завершений (например, H.221, H.223, H.225.0) и завершения, образующие входной мультиплексор.
Media	Список спецификаций потока медиа (см. п. 7.1.4).
TerminationState	Свойства завершения (которые могут быть определены в комплектах), не характерные для потока.
Stream	Список дескрипторов Remote/Local/LocalControl для одного потока.
Local	Содержит свойства, определяющие потоки медиа, получаемые MG от удаленного объекта.
Remote	Содержит свойства, определяющие потоки медиа, передаваемые MG удаленному объекту.
LocalControl	Содержит свойства (которые могут быть определены в комплектах), представляющие интерес для MG и MGC.
Events	Описывает события, которые необходимо обнаружить MG, и то, что необходимо делать при обнаружении события.
EventBuffer	Описывает события, которые необходимо обнаружить MG, когда активна буферизация событий.
Signals	Описывает сигналы (см. п. 7.1.11), применяемые к завершениям.
Audit	В командах Audit, идентификаторы, информация о которых необходима.
Packages	В AuditValue, возвращает список комплектов, реализуемых завершением.
DigitMap	Определяет шаблоны, с которыми должны сопоставляться последовательности определяемого набора событий так, чтобы они могли быть зарегистрированы как группа, а не по отдельности.
ServiceChange	В ServiceChange, что, почему произошло ServiceChange, и т. д.
ObservedEvents	В Notify или AuditValue, отчет о наблюдаемых событиях.
Statistics	В Subtract и Audit, отчет о статистике, ведущейся на завершении или потоке.
Topology	Описывает направления потоков между завершениями и контекстом.
ContextAttribute	Содержит свойства (которые могут быть определены в комплектах), влияющих на контекст в целом.
Error	Содержит код Error Code и дополнительно текст ошибки; это может произойти в ответах на команды и запросах Notify.
ПРИМЕЧАНИЕ. – Дескриптор Modem в Рек. МСЭ-Т Н.248.1 версия 2 (05/2002) не рассматривается.	

6.2.5 Завершение Root

Иногда команда должна относиться к шлюзу как к объекту самому по себе, а не к завершению внутри него. Для этой цели зарезервирован особый идентификатор TerminationID – "Root". На Root могут быть определены комплекты. Таким образом, Root может обладать свойствами, событиями, сигналами и статистикой. Соответственно, идентификатор Root TerminationID может появиться в:

- команде Modify – для изменения свойства, передачи сигнала или установления события;
- команде Notify – для отчета о событии;
- команде AuditValue – для проверки значений свойств и статистики, ведущейся на Root;
- команде AuditCapabilities – для определения, какие свойства Root реализуются;

- команде ServiceChange – для объявления о том, что весь шлюз находится в работе или вышел из строя.

Любое другое использование идентификатора Root TerminationID является ошибкой. В подобных случаях будет послан код ошибки 410 ("Неверный идентификатор").

6.3 Принципы обобщения

В данном пункте описывается режим работы с групповыми символами для идентификаторов ContextID и TerminationID, которые должны применяться ко всем командам. При обработке данных команд следует рассматривать две формы обобщения:

- 1) Обобщение контекста (Context wildcarding);
- 2) Обобщение завершения (Termination wildcarding).

Для целей процедур обобщения, список TerminationIDList, содержащий более одного TerminationID считается обобщенным TerminationID. При выполнении транзакции, содержащей обобщенные контексты и дополнительно обобщенные завершения, в транзакции все команды для определенной ступени ContextID выполняются до того, как перейти к последующей ступени ContextID. В случае если в транзакции множество команд, последующие команды в транзакции выполняются только тогда, когда TerminationID (обобщенный или заданный), заданный в первой команде, совпадает с отдельной ступенью ContextID. Если TerminationID последующих(ей) команд(ы) в транзакции (обобщенный или заданный) не совпадает с отдельной ступенью ContextID, отсылается код ошибки 431 ("Ни один TerminationID не совпал с обобщением"), и обработка последующих ступеней обобщенного ContextID прекращается, если команда, генерирующая ошибку, не является дополнительной.

Выполнение определенных комбинаций обобщения описывается ниже:

6.3.1 Заданный ContextID с обобщенным TerminationID

В случае, когда ContextID является заданным, когда в TerminationID команды используется ALL, результат аналогичен повторению команды для каждого из совпадающих TerminationID. Использование ALL не относится к завершению Root. Поскольку каждая из этих команд может генерировать ответ, совокупный размер ответа может быть большим. Таким образом, если обобщение совпадает более чем с одним TerminationID в контексте, исследуются все возможные совпадения, причем отчет делается по каждому результату. Если в заданном контексте нет ни одного из завершений, на которые ссылается обобщенный TerminationID, отсылается код ошибки 431 ("Ни один No TerminationID не совпал с обобщением"). Для заданных завершений, заданных обобщенным TerminationID, и не обнаруженных в заданном контексте, сообщений об ошибке не отсылается.

Например: Предположим, что на шлюзе имеется четыре завершения: t1/1, t1/2, t2/1 и t2/2. Предположим, что Контекст 1 содержит t1/1 и t2/1, а Контекст 2 содержит t1/2 и t2/2.

Команда:

```
Context=1 {Command=t1/* {Descriptor/s}}
```

ответ:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

6.3.2 Обобщенный ContextID (ALL) с заданным TerminationID

В случае, когда ContextID обобщен (т. е. ContextID = ALL), а TerminationID – полностью задан, результат аналогичен выполнению команды, задающей ненулевой контекст, содержащий заданное завершение. Таким образом, необходимо провести поиск для обнаружения контекста, выполняется только одна ступень команды. Для контекстов, не содержащих заданного завершения, сообщений об ошибке не отсылается. Если завершение не содержится ни в одном (ненулевом) контексте, отсылается сообщение об ошибке 411 ("Транзакция ссылается на неизвестный ContextID"). Использование данной модели вместо задания ContextID не рекомендуется, однако может быть полезным, например, при корректировании противоречивого состояния между MG и MGC.

Например: При использовании конфигурации, описанной выше, команда:

```
Context=* {Command=t1/1 {Descriptor/s}}
```

ответ:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

6.3.3 Обобщенный ContextID (ALL) с обобщенным TerminationID

В случае, когда ContextID обобщен (т. е. ContextID = ALL), и TerminationID обобщен, результат аналогичен повторению команды для каждого TerminationID, совпадающего с групповым символом в каждом ненулевом контексте, содержащем один или более совпадающих TerminationID. Таким образом, если групповой символ совпадает более чем с одним TerminationID на заданной ступени обобщенного ContextID, исследуются все возможные совпадения, причем отчет делается для каждого совпадения. Для контекстов, не содержащих завершения, совпадающего с обобщенным TerminationID, сообщений об ошибке не выводится. Для отдельных завершений, заданных в обобщенном TerminationID, но не находящихся на заданной ступени обобщенного ContextID, сообщений об ошибке не выводится. Если совпадений с обобщенными ContextID и TerminationID нет, выводится сообщение об ошибке 431 ("Ни один TerminationID не совпал с обобщением").

Например: Используя конфигурацию, описанную выше, команда:

```
Context=* {Command=t1/* {Descriptor/s}}
```

ответ:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

```
Context=2 {Command=t1/2 {Descriptor/s}}
```

В случае, когда в запросе по обобщенному TerminationID и/ или обобщенному ContextID содержится множество команд, если первая команда не совпадает с первой ступенью ContextID и TerminationID, последующая команда запроса для этой ступени выполнена не будет.

6.3.4 Обобщенные ответы

Если не требуются отдельные ответы, можно запросить обобщенный ответ. В таком случае генерируется единый запрос, содержащий (команду) UNION всех отдельных ответов, которые бы генерировались в противном случае, причем дублирующие значения исключаются. Например, дано Завершение Ta со Свойствами p1=a, p2=b и Завершение Tb со Свойствами p2=c, p3=d, ответ UNION будет состоять из обобщенного TerminationID и последовательности Свойств p1=a, p2=b,c и p3=d. Обобщенные ответы могут быть особенно полезны при использовании команд Audit. Если обобщенный ответ UNION используется совместно с обобщенным контекстом, посылается единый ответ с (командой?) UNION всех отдельных завершений, на которые ссылается TerminationID. Ответ будет содержать Context=ALL, обобщенный TerminationID и последовательность свойств.

Если при выполнении обобщенного запроса, требующего обобщенного ответа, возникает ошибка, требуется особая обработка для получения полезной информации об ошибке(ах) при сохранении умеренного (ограниченного) размера ответа. Когда запрашивается обобщенный ответ, необходимо выполнить все ступени команды (как описано выше), даже если результатом одной или более из них является ошибка, однако позднее команды транзакции не будут выполняться (если это не задано дополнительно). Для команды, результатом выполнения которой стала ошибка, необходимо вывести множество ответов. Первым ответом команды должен быть обычный обобщенный ответ, содержащий UNION ответов для тех команд, выполнение которых удалось. Если выполнение ни одной из команд не удалось, UNION не должен содержать ничего. Дополнительные ответы для каждого TransactionID для команд, выполнение которых не удалось, должны быть выведены с соответствующим дескриптором Error.

Например, команда:

```
Context=* {Command=t1/* {Descriptor/s}}
```

Ответ на ошибку:

```
Context=* {Command=t1/* {Union response descriptors},
```

```
Command=t1/3 {Error=errorcode}}
```

Кодирование алгоритма обобщения подробно описано в Приложениях А и В.

7 Команды

Данный протокол обеспечивает команды для управления логическими объектами модели соединения протокола, контекстами и завершениями. Команды обеспечивают управление на самом мелком уровне модульности, поддерживаемом протоколом. Например, существуют команды для добавления завершений к контексту, изменения завершений, вычитания завершений из контекста, проверки свойств контекстов и завершений. Команды предусматривают полное управление свойствами контекстов и завершений. Это включает в себя указание, по каким событиям завершение должно представлять отчет, какие сигналы/действия должны применяться к завершению, а также описание топологии контекста (кто кого слышит/видит).

Большинство команд предназначены для особого использования контроллера медиашлюза в качестве инициатора команд в управлении медиашлюзами, принимающими команду. Исключение составляют команды **Notify** и **ServiceChange**: команда **Notify** посылается от медиашлюза контроллеру медиашлюза, а команда **ServiceChange** может быть послана любым из этих объектов. Ниже приводится обзор команд, более подробно они описаны в п. 7.2.

- 1) **Add**: Команда **Add** добавляет завершение в контекст. Команда **Add** по первому завершению в контексте служит для создания нового контекста.
- 2) **Modify**: Команда **Modify** изменяет свойства, события и сигналы завершения.
- 3) **Subtract**: Команда **Subtract** отсоединяет завершение от контекста и запрашивает статистику по участию завершения в контексте. Команда **Subtract** по последнему завершению в контексте удаляет контекст.
- 4) **Move**: Команда **Move** атомарно переносит завершение в другой контекст.
- 5) **AuditValue**: Команда **AuditValue** делает запрос о текущем состоянии свойств, событий, сигналов и статистики завершения.
- 6) **AuditCapabilities**: Команда **AuditCapabilities** запрашивает все возможные значения для свойств, событий и сигналов завершений, разрешенные медиашлюзом.
- 7) **Notify**: Команда **Notify** позволяет медиашлюзу информировать контроллер медиашлюза о наступлении событий на медиашлюзе.
- 8) **ServiceChange**: Команда **ServiceChange** позволяет медиашлюзу уведомлять контроллер медиашлюза о том, что завершение или группа завершений вскоре не будет обслуживаться или только что было(а) только что введено в эксплуатацию. Команда **ServiceChange** также используется **MG** для извещения **MGC** о своей доступности (регистрация), и для уведомления **MGC** о предстоящем или выполненном перезапуске **MG**. **MGC** может объявить о передаче **MG** путем отсылки команды **ServiceChange**. **MGC** также может использовать команду **ServiceChange** для инструктирования **MG** о включении завершения или группы завершений в работу или отключении их.

Данные команды подробно описаны в пунктах с 7.2.1 по 7.2.8.

7.1 Дескрипторы

Параметры команды называются дескрипторами. Дескриптор состоит из имени и списка элементов. Некоторые элементы могут иметь значения. Многие команды имеют общие дескрипторы. В данном подпункте перечислены данные дескрипторы. Дескрипторы могут быть возвращены как результат команды. При любом таком возврате содержимого дескриптора, пустой дескриптор представлен лишь своим именем без какого-либо списка. Параметры и использование параметров, характерные для определенного типа команд описаны в подпунктах, где описываются команды.

7.1.1 Определяющие параметры

Параметры команды структурированы в несколько дескрипторов. Вообще текстовый формат дескрипторов представляет собой: DescriptorName=<someID> {parm=value, parm=value...}.

Параметры могут быть определены полностью, переопределены или недоопределены:

- 1) Параметры, определенные полностью, имеют одно точно выраженное значение, которое предписано инициатором команды принимающему команду для использования с заданным параметром.
- 2) Недоопределенные параметры, используя значение CHOOSE, позволяют принимающему команду выбрать любое значение, которое он может поддерживать.
- 3) Переопределенные параметры имеют список возможных значений. Порядок следования значений в списке определяет очередность выбора для инициатора команды. Принимающий команду выбирает одно значение из предложенного списка и возвращает это значение команде инициатору команды.

Если требуемый дескриптор, кроме дескриптора Audit не задан (т. е. полностью отсутствует) в команде, сохраняются прежние значения, установленные в данном дескрипторе для данного завершения. В командах, кроме Subtract, пропущенный дескриптор Audit аналогичен пустому дескриптору Audit. Режим работы MG при незадаваемых параметрах в дескрипторе зависит от отдельного дескриптора, что будет показано в последующих подпунктах. Если параметр недоопределен или переопределен, дескриптор, содержащий значение, выбранное принимающим, включается в результат команды.

Каждая команда задает TerminationID, с которым оперирует команда. Данный TerminationID может быть "обобщен". Когда TerminationID команды обобщен, результат должен быть таким, как если бы команда повторялась для каждого совпавшего TerminationID.

7.1.2 Дескриптор Modem

Дескриптор Modem определяет тип модема и параметры, если таковые имеются, для использования, например, в H.324 и текстовом разговоре. Данный дескриптор включает следующие типы модемов: V.18, V.22, V.22 bis, V.32, V.32 bis, V.34, V.90, V.91, Синхронный ISDN, и допускает добавления. Дескриптор Modem не присутствует в завершении по умолчанию.

Использование дескриптора Modem в Рек. МСЭ-Т Н.248.1 версия 2 (05/2002) и последующих версиях не рассматривается. Это означает, что не следует включать дескриптор Modem в качестве части передаваемого содержания и, при получении (данного дескриптора) его следует либо пропустить, либо обрабатывать как возможность реализации. Тип модема необходимо указывать в качестве атрибута потоков данных в дескрипторах Local и Remote.

7.1.3 Дескриптор Multiplex

В мультимедийных соединениях, несколько потоков медиа передаются по (возможно иному) числу транспортных каналов. Дескриптор Multiplex связывает медиа и транспортные каналы. Данный дескриптор включает следующие типы мультиплексирования:

- H.221;
- H.223;
- H.226;
- V.76;
- N × 64К;
- возможные дополнения,

и набор TerminationID, представляющих мультиплексированные транспортные каналы, в определенном порядке. Например:

$$\text{Mux} = \text{H.221}\{\text{M}_{\text{yT3}}/1/2, \text{M}_{\text{yT3}}/2/13, \text{M}_{\text{yT3}}/3/6, \text{M}_{\text{yT3}}/21/22\}$$

Тип мультиплексирования $N \times 64K$ подразумевает службу $N \times 64K$ (например, как определено Q.931 Скоростью Передачи Информации или Q.763 Требование Среды Передачи). Со стороны контекста, он поддерживает один поток широкополосных данных. Когда к контексту в неявном виде добавляется основное завершение в результате создания мультиплексного завершения $N \times 64K$, дескриптор Stream для основного завершения должен принять те же значения, как и дескриптор Stream, определенный для мультиплексного завершения, за исключением того, что ширина полосы для основного завершения должна составить 64 кбит/с.

7.1.4 Дескриптор Media

Дескриптор Media описывает параметры для всех потоков медиа. Эти параметры систематизированы в два дескриптора: дескриптор TerminationState, определяющий свойства завершений, не зависящих от потока, и один или более дескрипторов Stream, каждый из которых описывает один поток медиа.

Поток идентифицируется StreamID. Значение StreamID должно быть в диапазоне от 1 до 65535. StreamID используется для связывания в контексте потоков, подходящих друг другу. Множество потоков, исходящих из завершения, должны быть синхронизированы друг с другом. Внутри дескриптора Stream существует до четырех вспомогательных дескрипторов: LocalControl, Local, Remote и Statistics. Таким образом, связь между данными дескрипторами представляет собой:

Дескриптор Media

```
TerminationState Descriptor
Stream
Descriptor
  LocalControl Descriptor
  Local Descriptor
  Remote Descriptor
  Statistics Descriptor
```

Для удобства дескрипторы LocalControl, Local, Remote или Statistics могут быть включены в дескриптор Media без заключительного дескриптора Stream. В таком случае предполагается, что StreamID – один.

7.1.5 Дескриптор TerminationState

Дескриптор TerminationState содержит свойства ServiceStates, EventBufferControl и свойства завершения (определенные в комплектах), не свойственные определенному потоку (not stream specific).

Свойство ServiceStates описывает полное состояние завершения (не свойственное определенному потоку). Завершение может находиться в одном из следующих состояний: "Test", "OutOfService", или "InService". Состояние "Test" обозначает, что завершение проходит проверку. Состояние "OutOfService" показывает, что завершение не может быть использовано для перемещения данных. Состояние "InService" показывает, что завершение может быть использовано или используется для нормального перемещения данных. Состояние "InService" задано по умолчанию.

Значения, назначенные свойствам, могут быть простыми значениями (integer/string/enumeration) или могут быть недоопределенными, когда предоставляется более одного значения, и MG может сделать выбор:

- Альтернативные значения: в списке множество значений, одно из которых нужно выбрать;
- Диапазоны: минимальное и максимальное значения, нужно выбрать значение между минимальным и максимальным, включая граничные значения;
- Больше чем/ меньше чем: Значение должно быть больше/ меньше заданного значения;
- Групповой символ CHOOSE: MG выбирает из разрешенных для данного свойства значений.

Свойство EventBufferControl задает, заносятся ли события после обнаружения их в дескрипторе Events в буфер, или обрабатываются незамедлительно. Для получения более подробной информации см. п. 7.1.9.

7.1.5.1 Свойства TerminationState

7.1.5.1.1 ServiceStates

Название свойства: ServiceStates

Описание: Значение данного свойства показывает текущее состояние работы завершения.

Тип: Перечисление

Возможные значения: InService: Завершение находится в рабочем состоянии и функционирует нормально.

OutOfService: Завершение находится в нерабочем состоянии и не доступно для передачи данных.

Test: Завершение проходит проверку.

Задано по умолчанию: InService

Определяется в: TerminationState

Характеристики: Чтение/ Запись

7.1.5.1.2 EventBufferControl

Название свойства: EventBufferControl

Описание: Задаёт, помещаются ли события после их обнаружения в дескрипторе Events в буфер, или обрабатываются незамедлительно. См. п. 7.1.9.

Тип: Перечисление

Возможные значения: LockStep: События помещаются в буфер и обрабатываются согласно п. 7.1.9.

OFF: События обрабатываются незамедлительно.

Задано по умолчанию: OFF

Определено в: TerminationState

Характеристики: Чтение/ Запись

7.1.6 Дескриптор Stream

Дескриптор Stream задает параметры одного двунаправленного потока. Данные параметры систематизированы в три дескриптора: один, содержащий свойства завершения, характерные для определенного потока, и по одному для локального и удаленного потоков. Дескриптор Stream включает StreamID, который идентифицирует поток. Потоки создаются путем назначения нового StreamID на одном из завершений контекста. Удаление потока происходит путем установления для потока пустых дескрипторов Local и Remote, причем значения ReserveGroup и ReserveValue в LocalControl установлены на "False" на всех завершениях в контексте, предварительно поддерживавших данный поток.

StreamID представляют локальную значимость между MGC и MG, они назначаются MGC. Внутри контекста StreamID является средством, при помощи которого показывается, какие потоки медиа взаимосвязаны: соединяются потоки, имеющие одинаковый StreamID.

Если завершение переносится из одного контекста в другой, эффект на контекст в который переносится завершение такой же, как в случае, если бы добавлялись новые завершения с такими же StreamID, как у перемещенного завершения.

7.1.7 Дескриптор LocalControl

Дескриптор LocalControl содержит свойства Mode, ReserveGroup и ReserveValue, а также свойства завершения (определенные в комплектах), характерные для определенного потока и представляющие интерес для MG и MGC. Значения свойств могут быть заданы как в п. 7.1.1.

Разрешенными значениями для свойства Mode являются "SendOnly", "RecvOnly", "SendRecv", "Inactive" и "LoopBack". "SendOnly", "RecvOnly" и "LoopBack" работают с внешней областью контекста, так, например, поток, режим которого mode = "SendOnly", не пропускает получаемые медиа в контекст. Когда поток находится в режиме "LoopBack" на завершении, медиа, получаемые на завершении (дескриптор Local), будут возвращены обратно завершению на отсылающей стороне (дескриптор Remote), и между этим завершением и другими завершениями в данном контексте медиа передаваться не будут. Возвращаемые медиа должны быть переданы согласно дескриптору Remote. По умолчанию для свойства Mode задано значение "Inactive" (неактивен). Свойство Mode не оказывает влияния на сигналы и события. Свойство LocalControl Mode имеет приоритет перед любым режимом, задаваемым дескрипторами Local и Remote.

Свойства с логическими значениями на завершениях Reserve, ReserveValue и ReserveGroup, показывают, какие действия ожидаются от MG при получении дескриптора Local и/или Remote.

Если значение свойства Reserve – "True", MG должен зарезервировать ресурсы для всех альтернатив, определенных в дескрипторах Local и/или Remote для которых у него в данный момент имеются доступные ресурсы. Он должен ответить, для каких альтернатив он резервирует ресурсы. Если MG не может поддерживать ни одну из альтернатив, он ответ MGC должен содержать пустые дескрипторы Local и/или Remote. Если передача медиа начинается, в то время как зарезервировано более одной альтернативы, пакеты медиа могут быть переданы или получены на любую из альтернатив и должны быть обработаны, хотя в любой отдельно взятый момент времени может быть активна только одна альтернатива.

Если значение свойства Reserve – False, MG должен выбрать одну из альтернатив, определенных в дескрипторе Local Descriptor (если он присутствует), и одну из альтернатив, определенных в дескрипторе Remote (если он присутствует). Если MG еще не зарезервировал ресурсы для поддержки выбранной альтернативы, он должен зарезервировать ресурсы. С другой стороны, если он уже зарезервировал ресурсы для адресуемого завершения (в результате предыдущего обмена с ReserveValue и/или ReserveGroup со значением – "True"), он должен освободить любые избыточные ресурсы, зарезервированные ранее. В заключение MG должен послать MGC ответ, содержащий выбранные альтернативы для дескрипторов Local и/или Remote. Если у MG нет достаточно ресурсов для поддержки каких-либо из заданных альтернатив, он должен отослать в ответ код ошибки – Error Code 510 ("Недостаточно ресурсов").

Для ReserveValue и ReserveGroup по умолчанию задано значение "False". Более подробно использование данных двух свойств Reserve описано в п. 7.1.8.

В MG новые установочные параметры дескриптора LocalControl полностью замещают предыдущие установочные параметры данного дескриптора. Таким образом, чтобы сохранить информацию от предыдущих установочных параметров, MGC должен включить эту информацию в новые установочные параметры. Если MGC желает удалить некоторую информацию из существующего дескриптора, он просто заново посылает дескриптор (в команде Modify), извлекая ненужную информацию.

ПРИМЕЧАНИЕ. – В определениях по шифрованию в Приложениях А и В свойство Mode также называется "StreamMode". В рамках H.248 данные термины являются взаимозаменяемыми.

7.1.7.1 Свойства LocalControl

7.1.7.1.1 Mode

Название свойства: StreamMode

Описание: Значение данного свойства показывает текущее состояние работы завершения.

Тип: Перечисление

Возможные значения: Inactive: Завершение не пропускает медиа в поток.

SendOnly: Завершение пропускает медиа в поток из внутренней во внешнюю область контекста.

- RecvOnly: Завершение пропускает медиа в поток из внешней во внутреннюю область контекста.
- SendRecv: Завершение пропускает медиа в поток как в, так и из контекста.
- LoopBack: Завершение возвращает полученные медиа для потока обратно отправителю.

Задано по умолчанию: Inactive

Определено в: LocalControl

Характеристики: Чтение/ Запись

7.1.7.1.2 ReserveGroup

Название свойства: ReserveGroup

Описание: Определяет, нужно ли MG резервировать ресурсы для поддержки одной медиагруппы, или как можно большего числа медиагрупп так, как они определены в дескрипторах Local и Remote. См. 7.1.8.

ПРИМЕЧАНИЕ. – Термин "медиагруппа" обозначает содержимое продукта ASN.1 "PropertyGroup" (см. Приложение А) в двоичном коде, или индивидуальное описание сеанса SDP в текстовом коде.

Тип: Логическое выражение

- Возможные значения:** True: MG должен зарезервировать все возможные медиагруппы, обозначенные в дескрипторах Local и Remote.
- False: MG должен зарезервировать по одной медиагруппе для каждого из дескрипторов Local и Remote согласно системе, описанной в п. 7.1.8.

Задано по умолчанию: False

Определено в: LocalControl

Характеристики: Чтение/ Запись

7.1.7.1.3 ReserveValue

Название свойства: ReserveValue

Описание: Определяет, нужно ли MG резервировать ресурсы для поддержки одного набора значений свойства (например, один кодек и связанные с ним атрибуты) или как можно большего числа наборов так, как они определены в дескрипторах Local и Remote. См. 7.1.8.

Тип: Логическое выражение

- Возможные значения:** True: MG должен зарезервировать ресурсы для сохранения как можно большего числа наборов значений свойства, обозначенных в выбранной медиагруппе (если значение ReserveGroup – False) или в каждой медиагруппе (если значение ReserveGroup – True) в дескрипторах Local и Remote
- False: MG должен зарезервировать один набор значений свойства из обозначенных в выбранной медиагруппе (если значение ReserveGroup – False) или в каждой медиагруппе (если ReserveGroup – True) в дескрипторах Local и Remote.

Задано по умолчанию: False

Определено в: LocalControl

Характеристики: Чтение/ Запись

7.1.8 Дескрипторы Local и Remote

MGC использует дескрипторы Local и Remote для резервирования и высвобождения (commit) ресурсов MG для декодирования и кодирования данного(ых) потока(ов) и завершения, к которым они

применяются. MG включает данные дескрипторы в свой ответ для того, чтобы показать, что он действительно готов поддерживать. MG должен включить в свой ответ дополнительные свойства и их значения, если данные свойства являются обязательными, но не присутствуют в запросе, сделанном MGC (например, путем детального указания параметров кодирования видео, в то время как MGC указал лишь тип полезной нагрузки).

Во избежание неоднозначности при запросе MG о резервировании и предоставлении ресурсов, при использовании недоопределения (например, CHOOSE) MGC должен предоставить необходимое количество информации для того, чтобы MG мог сделать однозначный выбор. Например, при использовании CHOOSE без указания требуемого типа приложения (например, "название медиа" в случае кодирования SDP), может потребоваться дополнительная информация (например, строки атрибутов в случае кодирования SDP).

Дескриптор Local относится к медиа, получаемым MG, а дескриптор Remote относится к медиа, передаваемым MG.

При текстовом кодировании данного протокола, дескрипторы состоят из описаний сессии, как определено в SDP (RFC 2327). В описаниях сессии, передаваемых MG от MGC, допускаются следующие исключения из правил синтаксиса RFC 2327:

- строки "s = ", "t = " и "o = " являются дополнительными;
- допускается использование CHOOSE вместо единственного значения параметра; и
- допускает использование альтернатив вместо единственного значения параметра.

Дескриптор Stream определяет один двунаправленный поток медиа и, таким образом, одно описание сессии не должно содержать в себе более одного описания медиа (строка "m = "). Дескриптор Stream может содержать дополнительные описания сессии в качестве альтернатив. Каждый поток медиа должен проявиться/ возникнуть в отдельном дескрипторе Stream. Когда в одном дескрипторе предоставляется множество описаний сессии, в качестве разделителей

строки "v = " – разделители; в противном случае в описании сессии, посылаемом MG, они будут являться дополнительными. Реализации должны принимать описания сессии, которые полностью удовлетворяют требованиям RFC 2327 с учетом вышеупомянутых ограничений. При двоичном кодировании данного протокола, дескриптор состоит из групп свойств (пар значений тэгов), как описано в Приложении С. Каждая такая группа может содержать параметры описания сессии.

Ниже подробно описывается семантика дескрипторов Local и Remote. Описание состоит из двух частей. В первой части интерпретируется содержание дескриптора. Во второй части описываются действия, которые должен предпринять MG после получения дескрипторов Local и Remote. Действия, которые будет предпринимать MG, зависят от значений свойств ReserveValue и ReserveGroup дескриптора LocalControl.

Дескриптор Local или Remote, или они оба могут быть:

- заданными (т. е. отсутствовать);
- пустыми;
- недоопределенными посредством использования CHOOSE в значении свойства;
- полностью определенными; или
- переопределенными посредством представления множества групп свойств и по возможности множества значений свойств в одной или большем числе данных групп.

Когда происходит передача дескрипторов от MGC к MG, их интерпретация происходит согласно правилам, данным в п. 7.1.1, для разъяснения приводятся следующие дополнительные комментарии:

- а) Незаданный дескриптор Local или Remote считается пропущенным обязательным параметром. При этом требуется, чтобы MG использовал последнее значение, заданное для данного дескриптора. Возможно, что предварительно значение задано не было, в таком случае рассматриваемый дескриптор при дальнейшем выполнении команды игнорируется.
- б) Пустой дескриптор Local (Remote) в сообщении от MGC обозначает запрос о высвобождении любых ресурсов, зарезервированных для получаемого (передаваемого) потока медиа.

- c) Если в дескрипторе Local или Remote присутствует множество групп свойств или множество значений внутри группы, очередность – в порядке убывания.
- d) Недоопределенные или переопределенные свойства внутри группы свойств, посылаемой MGC, являются запросами для MG о выборе одного или более значений, которые он может поддерживать для каждого из свойств. В случае переопределенного свойства, список значений приводится в порядке убывания.

С учетом вышеописанных правил, последующие действия зависят от значений свойств ReserveValue и ReserveGroup в LocalControl.

Если значение ReserveGroup – "True", MG резервирует ресурсы, требующиеся для поддержки как можно большего числа запрашиваемых альтернатив групп свойств, которые он может поддерживать в настоящий момент. Если значение ReserveValue – "True", MG резервирует ресурсы, требующиеся для поддержки как можно большего числа запрашиваемых альтернатив значения свойств, которые он может поддерживать в настоящий момент.

ПРИМЕЧАНИЕ. – Если дескриптор Local или Remote содержит множество групп свойств, а значение ReserveGroup – "True", тогда на MG делается запрос о резервировании ресурсов так, чтобы он мог декодировать и кодировать поток медиа в соответствии с каждой из альтернатив. Например, Если дескриптор Local содержит две группы свойств, одна из которых определяет пакетированные аудиоданные А-типа G.711, а другая аудиоданные G.723.1, MG резервирует ресурсы таким образом, чтобы он мог декодировать один поток аудио, закодированный либо в формате А-тип G.711 или формате G.723.1. MG не нужно резервировать ресурсы для декодирования двух потоков аудио одновременно, одного, закодированного в формате А-тип G.711, а другого, закодированного в формате G.723.1. Принцип использования ReserveValue – аналогичный.

Если значение ReserveGroup – "True" или ReserveValue – "True", применяются следующие правила:

- Если у MG недостаточно ресурсов для поддержки всех альтернатив, запрашиваемых MGC, а MGC запросил ресурсы как в Local, так и в Remote, MG следует зарезервировать ресурсы для поддержки по крайней мере одной альтернативы как в Local, так и в Remote.
- Если у MG недостаточно ресурсов для поддержки по крайней мере одной альтернативы в дескрипторе Local (Remote), полученном от MGC, он должен выслать в качестве ответа пустой Local (Remote).
- В свой ответ MGC, если MGC включил дескрипторы Local и Remote, MG должен включить дескрипторы Local и Remote для всех групп свойств и значений свойств, для которых он зарезервировал ресурсы. Если MG не может поддерживать по крайней мере одну из альтернатив в дескрипторе Local (Remote), полученном от MGC, он должен выслать в качестве ответа пустой дескриптор Local (Remote).
- Если свойство Mode дескриптора LocalControl – "RecvOnly", "SendRecv", или "LoopBack", MG должен приготовиться принимать медиа, закодированные в соответствии с любой из альтернатив, включенных в его ответ MGC.

Если значение ReserveGroup – "False" и значение ReserveValue – "False", MG должен выполнять следующие правила для приведения Local и Remote к одной альтернативе:

- MG выбирает первую альтернативу в Local, для которой он может поддерживать по меньшей мере одну альтернативу в Remote.
- Если MG не может поддерживать по меньшей мере одну альтернативу Local и одну альтернативу Remote, он возвращает Код ошибки Error Code 510 ("Недостаточно ресурсов").
- MG возвращает альтернативу, выбранную им в каждом Local и Remote.

Новые установочные параметры дескриптора Local или Remote полностью замещают предыдущие установочные параметры данного дескриптора в MG. Таким образом, для того чтобы сохранить информацию из предыдущих установочных параметров, MGC должен включить данную информацию в новые установочные параметры. Если MGC желает удалить некоторую информацию из существующего дескриптора, он просто заново посылает дескриптор (в команде Modify), исключая ненужную информацию.

7.1.9 Дескриптор Events

Параметр дескриптора Events содержит RequestID и список событий, которые должен обнаруживать и по которым должен делать отчет медиашлюз. RequestID используется для установления

соотношения между запросом и уведомлениями, которые может вызвать этот запрос. Запрашиваемые события включают в себя, например, тональные сигналы факса, результаты проверки целостности тракта связи, а также переход между состояниями "трубка положена" и "трубка снята". Если дескриптор Events пустой, RequestID пропускается (т. е. события не определяются).

Каждое событие в дескрипторе содержит название события, дополнительный StreamID, дополнительный флаг KeepActive, дополнительный флаг NotifyBehaviour, дополнительный флаг ResetEventsDescriptor, а также дополнительные параметры. Название события состоит из PackageID (где определяется событие) и EventID. Для EventID может использоваться групповой символ ALL, обозначающий, что необходимо обнаруживать все события из определенного комплекта. По умолчанию значение StreamID составляет 0. Это означает, что искомое событие не относится к отдельному потоку медиа. События могут обладать параметрами. Это позволяет одному описанию события несколько варьироваться в значении, не создавая при этом большого количества отдельных событий. Дополнительные параметры события определяются в комплекте.

Если в дескрипторе Events присутствует или подразумевается событие завершения DigitMap, для передачи названия либо связанного значения DigitMap используется параметр EventDM. Для получения более подробной информации, см. п. 7.1.14.

Когда обработка события происходит в отношении к содержимому активного дескриптора Events, и оно было найдено в данном дескрипторе ("распознано"), МГ по умолчанию посылает MGC команду Notify. Уведомление можно отложить, если событие поглощается текущей строкой набора активного DigitMap (см. п. 7.1.14). На отправку команды Notify может повлиять флаг NotifyBehaviour. Кроме того, распознавание событий может вызвать прекращение активных в данный момент сигналов, или замену текущих дескрипторов Events и Signals, как описано в конце данного подпункта. Если замены дескриптора Events другим дескриптором Events не происходит, после распознавания события он остается активным.

Если значение свойства EventBufferControl равняется "LockStep", после обнаружения такого события, обычная обработка событий приостанавливается. Любое событие, обнаруживаемое впоследствии и возникающее в дескрипторе EventBuffer, добавляется в конец EventBuffer (простая очередь), вместе с указанием на время обнаружения. МГ должен ждать загрузки нового дескриптора Events. Загрузка нового дескриптора Events может произойти либо в результате получения команды с новым дескриптором Events, или путем активации вложенного дескриптора Events.

Если значение EventBufferControl равняется "Off", МГ продолжает обработку на основе активного дескриптора Events.

В случае активации вложенного дескриптора Events МГ продолжает обработку на основе вновь активированного дескриптора Events.

ПРИМЕЧАНИЕ 1. – Для целей управления EventBuffer, активация вложенного дескриптора Events аналогична получению нового дескриптора Events.

Когда МГ получает команду с новым дескриптором Events, одно или более событий можно занести в буфер в EventBuffer в МГ. Затем значение EventBufferControl определяет работу МГ с такими событиями, помещенными в буфер.

Случай 1

Если значение EventBufferControl равняется "LockStep", а МГ получает новый дескриптор Events, он проверит буфер обратного магазинного типа EventBuffer и предпримет следующие действия:

- 1) Если буфер EventBuffer пуст, МГ ожидает обнаружения событий, основываясь на новом дескрипторе Events.
- 2) Если буфер EventBuffer не пуст, МГ обрабатывает простую очередь, начиная с первого события:
 - a) Если событие в очереди является одним из событий, перечисленных в новом дескрипторе Events, МГ реагирует на событие и удаляет его из буфера EventBuffer. Штампом времени в Notify должно быть время действительного обнаружения события. Затем МГ ожидает нового дескриптора Events. Во время ожидания нового дескриптора Events, любые обнаруживаемые события, возникающие в дескрипторе EventBuffer, будут

помещаться в буфер EventBuffer. Когда происходит получение нового дескриптора Events, обработка событий будет повторена, начиная с шага 1).

- b) Если событие не присутствует в новом дескрипторе Events, MG должен отбросить данное событие и повторить, начиная с шага 1).

Случай 2

Если значение EventBufferControl равняется "Off", а MG получает новый дескриптор Events, он обрабатывает новые события при помощи нового дескриптора Events.

Если MG получает команду, предписывающую ему установить значение EventBufferControl на "Off", все события в буфере EventBuffer должны быть отброшены.

MG может предоставлять отчет о нескольких событиях в одной транзакции, если только это не создает ненужной задержки в предоставлении отчета по отдельным событиям.

Для получения отдельной информации по процедурам, касающимся передачи команды Notify, обратитесь к соответствующему приложению или Рекомендации H.248.x.

Заданное по умолчанию значение EventBufferControl – "Off".

ПРИМЕЧАНИЕ 2. – Поскольку свойство EventBufferControl находится в дескрипторе TerminationState, MG может получать команду, изменяющую свойство EventBufferControl и не содержащую дескриптора Events.

Обычно распознавание какого-либо события должно вызывать прекращение всех активных сигналов. Если в событии назначается KeepActive, MG не должен прерывать никакие активные сигналы на завершении, на котором было обнаружено событие.

Флаг NotifyBehaviour может использоваться для обозначения того, что команда Notify:

- посылается незамедлительно (NotifyImmediate: данное значение задано по умолчанию);
- не посылается совсем (NeverNotify);
- или регулируется (т. е. посылается или не посылается) в соответствии с загрузкой MGC (RegulatedNotify).

Более подробную информацию по использованию NotifyBehaviour см. в пункте E.15 Приложения. При использовании с DigitMaps уведомительный режим работы имеет место, когда завершается активный DigitMap. Regulated Notify может иметь дополнительный регулируемый вложенный дескриптор Events или Signals, связанный с ним. Если уведомление регулируется (т. е. подавляется), нужно активировать дополнительный регулируемый вложенный дескриптор. Если Notify не регулируется, запускается исходный вложенный дескриптор. Если устанавливается NotifyImmediate или NeverNotify, исходный вложенный дескриптор запускается при обнаружении события.

Результат появления флага ResetEventDescriptor напротив события зависит от того, является ли дескриптор Events, содержащий событие, вложенным дескриптором Events. В случае с вложенным дескриптором Events флаг ResetEventsDescriptor вызывает повторную установку дескриптора Events для данного завершения в состояние, предшествовавшее активации вложенного дескриптора (т. е. последний дескриптор Events, установленный в явном виде при помощи команды Modify или, если команды Modify не было с тех пор, как завершение было запущено повторно путем помещения в нулевой контекст, дескриптор Events, предоставленный MG). В случае невложенного дескриптора Events флаг ResetEventsDescriptor вызывает сброс активного дескриптора Events для данного завершения путем повторной активации любого(ых) события(ий) завершения DigitMap, которые были согласованы и деактивированы.

Флаг ResetEventsDescriptor должен быть установлен только на физических завершениях. Когда завершение находится в нулевом контексте, нужно действовать в соответствии с флагом ResetEventsDescriptor. Если флаг ResetEventsDescriptor встретился, когда завершение находится вне нулевого контекста, он не будет действовать.

Событие может включать в себя вложенный дескриптор Signals и/ или вложенный дескриптор Events, который, если присутствует, заменяет текущий дескриптор Signals/Events при распознавании события. Например, возможно задать, чтобы при распознавании события состояния занятости абонентского шлейфа генерировался сигнал тонального вызова, или чтобы при распознавании цифры

сигнал тонального вызова прекращался. Контроллер медиашлюза не должен посылать дескриптор Events, содержащий как вложенный дескриптор Signals так и отметку KeepActive.

Разрешается только один уровень вложения. Вложенный дескриптор Events не должен содержать другого вложенного дескриптора Events; вложенный дескриптор Events может содержать вложенный дескриптор Signals.

Дескриптор Events, полученный медиашлюзом заменяет все предыдущие дескрипторы Events. Выполняемое уведомление о событиях должно завершиться, и обработка событий, обнаруженных после выполнения команды, содержащей новый дескриптор Events, должна происходить согласно новому дескриптору Events.

Пустой дескриптор Events блокирует распознавание событий и предоставление отчетов по ним. Пустой дескриптор EventBuffer очищает EventBuffer и полностью блокирует накопление событий в режиме "LockStep": отчет будет предоставляться только о событиях, произошедших, пока активен дескриптор Events. Если происходит активация дескриптора Events, пока завершение работает в режиме "LockStep", EventBuffer очищается незамедлительно.

7.1.10 Дескриптор EventBuffer

Дескриптор EventBuffer содержит список событий и их параметров (при наличии), которые MG должен обнаружить и занести в буфер, если значение EventBufferControl равно LockStep (см. п. 7.1.9).

7.1.11 Дескриптор Signals

Сигналы представляют собой средства связи, генерируемые MG, такие, как тональные сигналы и сигналы оповещения, а также сигналы, относящиеся к каналу – носителю, такие, как переключение рычага телефонного аппарата. Более сложные сигналы могут включать в себя последовательность таких простых сигналов, перемежающуюся и обусловленную получением и анализом сигналов, относящихся к среде передачи и каналу – носителю. В качестве примера можно привести отражение полученных данных как, например, в комплекте Continuity Test. Сигналы могут также запросить подготовку содержания медиа для будущих сигналов.

Дескриптор Signals является параметром, содержащим набор сигналов, которые медиашлюз должен применять к завершению. Дескриптор Signals содержит несколько сигналов и/ или последовательные списки сигналов. Дескриптор Signals может содержать нулевые сигналы и последовательные списки сигналов. Поддержка последовательных списков сигналов является дополнительной.

Сигналы определяются в комплектах. Название сигнала должно состоять из PackageID (в котором определяется сигнал) и SignalID. В SignalID не должны использоваться групповые символы (замены). Сигналы, возникающие в дескрипторе Signals, имеют дополнительный параметр StreamID (по умолчанию задано значение 0 для обозначения того, что сигнал не относится к какому-либо определенному потоку медиа), и дополнительный тип сигнала (см. ниже), дополнительную длительность и, возможно, параметры, определенные в комплекте, определяющим сигнал. Это допускает некоторое варьирование значения одного сигнала, избавляя от необходимости создавать большое число отдельных каналов.

Дополнительный параметр NotifyCompletion позволяет MGC обозначить свое желание получить уведомление, когда сигнал закончит существовать. Это возможно в случаях, когда время существования сигнала закончилось (или иными словами сигнал завершился сам по себе), сигнал был прерван событием, цикл/ повтор сигнала завершился, сигнал был прерван при замене дескриптора Signals, или сигнал прекратился или даже не начинался по другим причинам.

Если параметр NotifyCompletion не включается в дескриптор Signals, уведомление генерируется, только если сигнал прекратился или даже не начинался по другим причинам. Для получения отчета, в дескрипторе Events, активном в данный момент, должен быть активировано событие Signal Completion (см. E.1.2). Дополнительный параметр "RequestID" можно связать с определенной ступенью SignalID, где запрашивается множество одного и того же SignalID. Это позволяет MGC распознавать различные Signal Completion ObservedEvents для этой определенной ступени сигнала. Если параметр NotifyCompletion отсутствует, параметр RequestID включать не нужно.

Продолжительность представляет собой целое значение, выражаемое в сотых долях секунды.

Существует три типа сигналов:

- OnOff (OO): сигнал длится, пока его не выключат;
- TimeOut (TO): сигнал длится до тех пор, пока его не выключат, или пока не пройдет определенный период времени;
- Brief (BR): сигнал прекратится сам по себе, если только новый применяемый дескриптор Signals не вызовет его прекращение, значение лимита времени не нужно.

Если в дескрипторе Signals происходит замещение заданного по умолчанию типа сигнала, отличного от TO, на тип TO, должен присутствовать параметр продолжительности.

Если в дескрипторе Signals определяется тип сигнала, происходит замещение заданного по умолчанию типа сигнала (см. п. 12.1.4). Если назначается продолжительность для сигнала on/off, ее следует пропускать.

Последовательный список сигналов состоит из SignalListID и последовательности сигналов, которые следует воспроизвести один за другим. Сигнал on/off может быть лишь завершающим элементом в последовательности сигналов в последовательном списке сигналов. Продолжительностью последовательного списка сигналов является суммарная продолжительность сигналов, содержащихся в нем, плюс сумма расчетной межсигнальной задержки, определяемая в качестве параметров сигналов.

Если межсигнальная задержка определяется для сигнала, отсутствующего в последовательном списке сигналов или являющегося последним в последовательном списке сигналов, ее следует игнорировать. И сигнал должен завершиться на завершении сигнала до применения хронометраж межсигнальной задержки. Продолжительность сигнала в SignalList с межсигнальной задержкой включает в себя хронометраж межсигнальной задержки.

Множественные сигналы и последовательные списки сигналов в одном и том же дескрипторе Signals нужно воспроизводить одновременно.

По определению сигналы направляются от завершения во внешнюю область контекста, если не определено иначе. Если направление сигнала задается в дескрипторе Signals, происходит замещение заданного по умолчанию направления движения сигнала. Для сигналов, имеющих параметр направления, базовый параметр направления протокола является приоритетным перед любым параметром направления, определенным в комплекте, в случае, когда заданы оба эти параметра. Если MGC определяет направление для сигнала, которое не поддерживается MG, MG должен ответить кодом ошибки Error Code 501 ("Не реализуется"). Когда один и тот же сигнал применяется к множеству завершений в рамках одной транзакции, для генерирования данных сигналов MG следует рассматривать использование одного и того же источника.

Порождение сигнала на завершении прекращается при применении нового дескриптора Signals, или при обнаружении события на завершении (см. п. 7.1.9).

Новый дескриптор Signals заменяет любой существующий дескриптор Signals. Необходимо приостановить любые сигналы, применяемые к завершению, и не находящиеся в заменяющем дескрипторе. Применяются новые сигналы, кроме следующих случаев. Порождение сигналов, присутствующих в заменяющем дескрипторе и содержащих флаг KeepActive, должно быть продолжено, если в данный момент их воспроизведение продолжается и еще не закончилось. Если замещающий дескриптор Signals содержит сигнал, который в данный момент не воспроизводится и содержит флаг KeepActive, данный сигнал нужно пропустить. Если замещающий дескриптор содержит последовательный список сигналов с таким же идентификатором, как и существующий дескриптор, тогда:

- нужно игнорировать тип сигнала и последовательность сигналов в последовательном списке сигналов замещающего дескриптора;
- воспроизведение сигналов в последовательном списке сигналов в существующем дескрипторе не должно прерываться.

7.1.12 Дескриптор Audit

Дескриптор Audit определяет, проверка какой информации будет проводиться. Дескриптор Audit назначает список дескрипторов и/ или отдельных свойств, по которым хочет получить ответ. Можно использовать проверку в любой команде для получения ответа от любого дескриптора, содержащего

текущие значения свойств, событий, сигналов и статистики, даже если данный дескриптор не присутствовал в команде или не имел недоопределенных параметров. Возможными элементами в дескрипторе Audit являются:

Modem (не рассматривается, см. п. 7.1.2)	
Mux	
Events	
Media	
Signals	
ObservedEvents	
DigitMap	
Statistics	
Packages	
EventBuffer	
Individual Audit Items:	
Media Properties	DigitMaps
Events	Statistics
EventBuffer	Packages
Signals, SignalLists	ContextAttribute

Дескриптор Audit может быть пустым, в таком случае, ответы по дескрипторам не передаются. Это полезно в команде Subtract для запрещения ответов по статистике, особенно при использовании групповых символов замены.

7.1.13 Дескриптор ServiceChange

Дескриптор ServiceChange содержит следующие параметры:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeAddress;
- ServiceChangeDelay;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMGCID;
- TimeStamp;
- Extension;
- ServiceChangeInfo;
- ServiceChangeIncompleteFlag.

См. п. 7.2.8.

7.1.14 Дескриптор DigitMap

7.1.14.1 Определение, создание, изменение и удаление DigitMap

DigitMap представляет собой план нумерации, резидентно находящийся на медиашлюзе, и используемый для обнаружения и предоставления отчета о цифровых событиях, полученных на завершении. Дескриптор DigitMap содержит имя DigitMap, и присвоенный DigitMap. DigitMap может быть предварительно загружен на MG системным администратором, ссылка на него может осуществляться по имени в дескрипторе Events, он может быть определен динамически, и впоследствии ссылка на него может осуществляться по имени, или сам действующий план нумерации может быть определен в дескрипторе Events. Для события завершения DigitMap в дескрипторе Events

разрешено делать ссылку по имени на DigitMap, определенный дескриптором DigitMap в рамках одной и той же команды, вне зависимости от порядка передачи соответствующих дескрипторов.

DigitMaps, определенные в дескрипторе DigitMap могут появиться в любой стандартной команде управления завершением протокола. Однажды определенный DigitMap может быть использован на всех завершениях, заданных TerminationID (возможно обобщенным) в такой команде. DigitMaps, определенные на завершении Root, являются глобальными, и могут быть использованы на каждом завершении в MG, при условии, что DigitMap с таким же именем не был определен на данном завершении. Когда DigitMap в дескрипторе DigitMap определяется динамически:

- Новый DigitMap создается путем указания имени, которое еще не было определено. Значение должно присутствовать.
- Обновление значения DigitMap происходит путем предоставления нового значения для уже определенного имени. Завершения, использующие DigitMap в данный момент должны продолжить использовать старое определение; последующие дескрипторы Events, определяющие имя, включая любой дескриптор Events в команде, содержащей дескриптор DigitMap, должны использовать новое (значение).
- Удаление DigitMap происходит путем предоставления пустого значения для уже определенного имени. Завершения, использующие DigitMap в настоящий момент должны продолжить использовать старое определение.

7.1.14.2 Таймеры DigitMap

Совокупность цифр согласно DigitMap может быть защищена тремя таймерами, а именно стартовым таймером (start timer) (T), коротким таймером (short timer) (S), и длинным таймером (long timer) (L).

- 1) Стартовый таймер (T) используется до того, как любые цифры (digits) станут доступными для обработки по DigitMap. Если стартовый таймер замещается значением, установленным на ноль ($T = 0$), тогда стартовый таймер должен быть отключен. Это предполагает то, что MG будет ждать цифр в течение неопределенного времени.
- 2) Если медиашлюз может определить, чтобы для соответствия цифровой строки любому из разрешенных шаблонов DigitMap была необходима по крайней мере еще одна цифра, то значение межцифрового таймера должно быть установлено равным большой (L) продолжительности (например, 16 секунд)
- 3) Если цифровая строка совпала с одним из шаблонов в DigitMap, но возможно, что могут быть дополнительно получены цифры, которые вызвали бы совпадение с другим шаблоном, тогда, вместо того, чтобы делать отчет о совпадении незамедлительно, MG должен применить короткий таймер (S) и ждать дополнительных цифр.

Таймеры являются реконфигурируемыми параметрами DigitMap. Заданные по умолчанию значения данных таймеров следует предоставить на MG, но они могут быть замещены значениями, заданными в рамках DigitMap.

7.1.14.3 Синтаксис DigitMap

Формальный синтаксис DigitMap описывается посредством правила DigitMap в описании формального синтаксиса протокола (см. Приложения А и В). Согласно этому синтаксису DigitMap определяется либо строкой, либо списком строк. Каждая строка в списке представляет собой альтернативную последовательность событий, определенной либо как последовательность символов DigitMap, либо как регулярное выражение символов DigitMap. Данные символы DigitMap, представляющие собой цифры от "0" по "9" и буквы от "A" до максимального значения в зависимости от задействованной системы сигнализации, не превышающего, однако, буквы "K", соответствуют событиям, определяемым в комплекте, определяемым в дескрипторе Events на завершении, к которому применяется данный DigitMap. (Соответствие между событиями и символами DigitMap определяется в документации к комплектам, относящимся к системам сигнализации по каналам, таким, как DTMF, MF, или R2. Цифры с "0" по "9" должны быть приведены в соответствие с соответствующими цифровыми событиями в рамках рассматриваемой системы сигнализации. Следует назначать буквы логичным образом, облегчая использование диапазонной нотации для альтернативных событий.)

Буква "x" используется в качестве группового символа замены, обозначая любое событие, соответствующее символам в диапазоне "0"–"9". Строка может также содержать явно выраженные диапазоны и, в общем смысле, явно выраженные наборы символов, обозначающих альтернативные события, каждое из которых удовлетворяет данной позиции DigitMap. Символ "точка" "." заменяет ноль или более повторений селектора событий (событие, диапазон событий, набор альтернативных событий, или групповой символ замены), который предшествует ему. В качестве результата третьего правила расчета времени, описанного выше, меж-событийный отсчет времени, соответствующий символу завершения "точка", использует по умолчанию короткий таймер.

В дополнение к данным символам событий, строка может содержать спецификаторы межсобытийного отсчета времени "S" и "L", а также модификатор длительности "Z". Буквы "S" и "L" означают соответственно, что MG следует использовать короткий (S) и длинный (L) таймеры для последующих событий, замещая правила отсчета времени, описанные выше. Если выраженный в явном виде спецификатор отсчета времени в действительности находится в одной альтернативной последовательности событий, но в любой другой возможной альтернативе не дается такого спецификатора, необходимо использовать значение таймера, установленное спецификатором отсчета времени, выраженным в явном виде. Если все последовательности с явно выраженным управлением отсчетом времени удаляются из набора возможностей, отсчет времени возвращается к заданным по умолчанию правилам, приведенным выше. При использовании в рамках диапазонной нотации, спецификаторы S и L нужно пропускать/игнорировать. В заключение, если конфликтующие спецификаторы отсчета времени находятся в разных альтернативных последовательностях, необходимо использовать длинный таймер.

Буква "Z" обозначает продолжительное событие. Эта буква, расположенная перед символом(ами), обозначающим(и) событие(я), удовлетворяющее(ие) данной позиции цифры, показывает, что эта позиция удовлетворяется, только если продолжительность события превышает пороговое значение длиной продолжительности. Предполагается, что данное значение задается MG, однако, как и таймеры T, L, и S, его можно заменить спецификацией в DigitMap. Если после спецификатора Z не стоит цифра (0-9 или A-K), MG должен отклонить данный план нумерации как неверную процедуру. При использовании в диапазонной нотации спецификатор Z применяется только к цифре, непосредственно следующей за ним. При использовании непосредственно перед диапазоном модификатор Z применяется ко всем цифрам в диапазоне (таким образом, требуя, чтобы совпадение в диапазоне было длиной продолжительности).

7.1.14.4 Событие завершения DigitMap

DigitMap активен, пока запускающий его дескриптор Events активен и не завершен. DigitMap завершается, когда:

- закончился таймер; или
- было найдено соответствие альтернативной последовательности событий, и ни одна другая альтернативная последовательность событий в DigitMap не может быть подобрана путем обнаружения дополнительного события (однозначное совпадение); или
- было обнаружено событие так, что совпадение с полной альтернативной последовательностью событий DigitMap стало невозможным, вне зависимости от получения дополнительных событий.

По завершении должно быть сгенерировано событие завершения DigitMap, как определено в комплекте, обеспечивающем отображение данных событий в DigitMap. В этот момент происходит деактивация DigitMap. Последующие события в комплекте обрабатываются следующим образом:

- Если значение EventBufferControl – "LockStep", последующие цифровые события обрабатываются как все остальные события;
- Если значение EventBufferControl – "OFF", если активный дескриптор Events не изменялся, и если в данном дескрипторе не активировались отдельные цифровые события, тогда будет инициировано занесение цифр в буфер. Буферизация будет продолжаться до тех пор, пока не истечет время буферизации, определенное в исходном событии завершения DigitMap, или до тех пор, пока не будет заменен активный дескриптор Events.

Цифровой буфер должен принять логическую форму строки набора, включающей в себя цифровые символы, как представлено в DigitMap, перед которыми, возможно, стоит 'Z'.

Пороговое значение продолжительности тонального сигнала, используемого для идентификации продолжительных событий, должно быть таким же, как и пороговое значение, используемое в недавно завершённом DigitMap.

По умолчанию время буферизации составляет ноль (буферизации не происходит), если иное не определено в явном виде в событии завершения DigitMap. Если буферизация прекращается из-за истечения таймера буферизации, содержимое цифрового буфера удаляется.

Если буферизация была прекращена новым дескриптором Events, и этот новый дескриптор Events содержит новое событие завершения DigitMap из того же комплекта, что и предыдущее, обработка любых буферизируемых цифр происходит по DigitMap как описывается ниже. Буферизованные цифры, не поглощенные новым DigitMap обрабатываются так, как если бы они были замещены после завершения схемы (map).

Если вместо этого новый дескриптор Events активирует предоставление отчета по отдельным цифровым событиям, необходимо произвести незамедлительную обработку всего набора буферизованных событий, предоставить отчет по применимым событиям и очистить буфер.

В заключение, если новый дескриптор Events не активирует ни события завершения DigitMap, ни отчета по отдельным цифровым событиям из рассматриваемого комплекта, содержимое буфера удаляется, и буфер перестает существовать.

7.1.14.5 Процедуры DigitMap

В ожидании завершения обработка последующих событий должна проходить согласно следующим правилам:

- 1) "Текущая строка набора", внутренняя переменная, первоначально пуста. Набор возможных альтернативных последовательностей событий включает в себя все альтернативы, определенные в DigitMap.
- 2) На каждом этапе, если буферизованные цифры доступны, самая старшая из них (возможно, вместе со спецификатором длинной продолжительности (Z)) удаляется из буфера, и обработка переходит к следующему этапу, как если бы цифровое событие было замечено только что. В противном случае, создается таймер, ожидающий следующего события. Он создается либо на основе заданных по умолчанию правил отсчета времени, приведенных выше, либо на основе явно выраженного отсчета времени, определенного в одной или более альтернативных последовательностей событий. Если таймер истекает, и представитель возможного набора альтернатив полностью удовлетворен, отсылается отчет о завершении по истечении времени с полным совпадением. Если таймер истекает, а удовлетворяется лишь часть альтернатив, или не удовлетворяется ни одной альтернативы, отсылается отчет о завершении по истечении времени с частичным совпадением. В обоих случаях, если событие завершения DigitMap разрешает предоставление подробного отчета об истечении времени, строка набора, по которой предоставляется отчет, будет заканчиваться на 'L', 'S', или 'T' соответствующим образом.
- 3) Если обнаружение события происходит до истечения таймера, устанавливается соответствие данного события с каким-либо символом в цифровой строке, и оно временно присоединяется в конец текущей строки набора. Продолжительность события (длительное или не длительное) указывается тогда и только тогда, когда это важно в текущей позиции символа (так как по меньшей мере одна из возможных альтернативных последовательностей событий включает в себя модификатор "Z" в данной позиции последовательности).
- 4) Текущая строка набора сравнивается с возможными альтернативными последовательностями событий. Тогда и только тогда, когда происходит совпадение последовательности, ожидающей события большой продолжительности, в данной позиции (т. е. событие имеет большую продолжительность и удовлетворяет требованиям спецификации для данной позиции), происходит удаление всех альтернативных последовательностей событий, не определяющих события большой продолжительности в данной позиции, и происходит изменение текущей строки набора путем помещения буквы "Z" перед символом, представляющим последнее событие. Любая последовательность, ожидающая события большой продолжительности в данной позиции, но не совпадающая с наблюдаемым/

замеченным событием, удаляется из набора возможных альтернатив. Если альтернативные последовательности событий, не определяющие события большой длительности в данной позиции, остаются в наборе альтернатив после применения вышеописанных правил, наблюдаемая продолжительность события не расценивается как важная в определении соответствий между ними.

- 5) Если остается ровно одна альтернатива, и происходит полное совпадение, генерируется событие завершения, обозначающее однозначное соответствие. Если не остается ни одной альтернативы, последнее событие удаляется из текущей строки набора, и генерируется событие завершения, обозначающее полное совпадение, если одна из альтернатив предыдущего этапа была полностью удовлетворена до обнаружения последнего события, или, в противном случае, частичное совпадение. В отчете событие, удаленное из текущей строки набора, будет представлено как отдельное событие, занесенное в буфер или удаленное в соответствии с правилами, описанными в предыдущем пункте. Данное утверждение определяется следующим образом:
 - а) Событие завершения DigitMap может задать, чтобы отчет по удаленному событию был предоставлен как параметр события завершения. Это происходит независимо от последующей обработки цифрового события.
 - б) Событие завершения DigitMap может задать, чтобы лишняя цифра была удалена. В таком случае, она удаляется незамедлительно. Любое занесение в буфер или другая обработка применяется только к последующим событиям.
- 6) Если в результате этапа 5) отчет о событии завершения не приходит, обработка снова переходит к этапу 2).

7.1.14.6 Активация DigitMap

Активация DigitMap происходит всякий раз, когда к завершению применяется новый дескриптор Events, или активируется вложенный дескриптор Events, и этот дескриптор Events содержит событие завершения DigitMap. Событие завершения DigitMap содержит поле EventDM в поле запрашиваемых действий. Каждая новая активация DigitMap начинается с этапа 1 вышеописанной процедуры с чистой текущей строкой набора. Все предыдущее содержимое текущей строки набора от предыдущей активации теряется.

Событие завершения DigitMap, не содержащее поля EventDM в поле запрашиваемых действий, считается ошибочным. После получения такого события в дескрипторе Events, MG должен ответить сообщением об ошибке, включая код ошибки Error Code 457 ("В сигнале или событии пропущен параметр").

7.1.14.7 Взаимодействие DigitMap и обработки событий

Во время активации DigitMap, запускается процесс обнаружения для всех событий, определенных в комплекте, содержащем заданное событие завершения DigitMap. Для каждого из таких обнаруженных событий продолжает применяться обычный режим работы с событиями (например, прекращение сигнала в случае, если флаг KeepActive в цифровом событии завершения не активирован) кроме случаев, когда:

- события в комплекте, содержащем заданное событие завершения DigitMap, кроме самого события завершения, не объявлены отдельно и не имеют побочных эффектов, если не активируются по отдельности; и
- событие, которое запускает событие завершения частичного совпадения, не распознается, и поэтому не имеет побочных эффектов до новой обработки после распознавания события завершения DigitMap. Таким же образом цифровые события, занесенные в буфер, не распознаются и не имеют побочных эффектов до обработки.

7.1.14.8 Групповые символы замены

Заметьте, что если комплект содержит событие завершения DigitMap, спецификация события, состоящая из PackageID с обобщенным EventID, активирует DigitMap; с этой целью, спецификация события должна включать в себя поле EventDM согласно п. 7.1.14.6. Если комплект также содержит

сами цифровые события, данная форма спецификации события вызовет предоставление отчета MGC по отдельным событиям по мере их обнаружения.

7.1.14.9 Пример

В качестве примера рассмотрим следующий план набора:

0	Местный оператор
00	Междугородний оператор
xxxx	Местный добавочный номер (начинается на 1-7)
8xxxxxxx	Местный номер
#xxxxxxx	Сторонний добавочный номер
*xx	Услуги по звездочке
91xxxxxxxxxxx	Междугородний номер
9011 + до 15 цифр	Международный номер

Если для накопления набранных цифр используется комплект обнаружения DTMF, описанный в Е.6, в результате плана набора, изображенного выше, получится следующий DigitMap:

(0 | 00 | [1-7] xxx | 8xxxxxxx | Fxxxxxxx | Exx | 91xxxxxxxxxxx | 9011x.)

7.1.15 Дескриптор Statistics

Дескриптор Statistics предоставляет информацию, описывающую статус и использование завершения во время его существования (непродолжительное завершение) или в то время, когда оно находится вне нулевого контекста (физическое завершение). Статистика может предоставляться на уровне завершения или быть связана с определенным дескриптором Stream. По умолчанию статистика предоставляется на уровне завершения. Если поток не в состоянии поддерживать какую-либо статистику, в качестве ответа должен быть возвращен код ошибки Error Code 460 ("Невозможно наладить статистику на потоке"). Для каждого завершения ведется набор стандартной статистики по соответствующим параметрам (число переданных и полученных байтов, например). По умолчанию конкретная статистика, по которой производится отчет для данного завершения, определяется Комплектами, реализуемыми данным завершением. Используя данный дескриптор, можно также обозначить какие статистические данные следует собирать. Установочные параметры дескриптора Statistics замещают ранее установленный дескриптор Statistics. Таким образом, чтобы поддерживать уже существующую статистику, нужно включить ее в новый дескриптор Statistics, и сброса статистических значений не произойдет. Статистические данные, удаленные из дескриптора Statistics, должны сохранять свои значения до того, как завершение будет вычтено. Однако если происходит реактивация определенных статистических данных при помощи последующего дескриптора Statistics, произойдет сброс значения. MGC может реактивировать всю статистику на завершении/ потоке путем запуска дескриптора Statistics с единой статистикой с PackageID и StatisticID, обобщенным символом "ALL". Для реактивации всей статистики в определенном комплекте на завершении/ потоке, запускает дескриптор Statistics с единой статистикой с определенным PackageID и StatisticID, обобщенным при помощи символа "ALL". Получение пустого дескриптора означает, что не нужно собирать статистику для определенного завершения.

Если предварительно не был использован пустой дескриптор Statistics для обозначения того, что собирать статистику не нужно, по умолчанию отчет по статистике на уровне завершения и потока предоставляется, когда завершение перестает существовать или возвращается в нулевой контекст в результате команды Subtract. Данный режим работы по умолчанию может также быть замещен путем включения пустого дескриптора Audit в команду Subtract. Если поток удаляется согласно п. 7.1.6, по умолчанию отчет по статистике на уровне потока не предоставляется. Проверка дескриптора Statistics в потоке должна осуществляться до удаления потока для сбора статистики. Статистические данные могут быть также получены в результате выполнения команды AuditValue, или любой команды Add/Move/Modify, использующей дескриптор Audit.

Статистика является кумулятивной; предоставление отчета по статистике не сбрасывает статистических значений. Значение Statistic на уровне завершения является результатом многозначительной функции более высокого уровня (как, например, суммой или средним значением) значений так, как если бы статистику применяли на всех потоках в завершении. Такая функция более

высокого уровня зависит от определенного типа статистики. Если в комплекте, определяющем отдельный вид статистики, не определено иначе, по умолчанию производится суммирование значений. Статистические данные сбрасываются, когда завершение перестает существовать или возвращается в нулевой контекст в результате команды `Subtract`.

7.1.16 **Дескриптор Packages**

Дескриптор `Packages`, используемый только с командой `AuditValue`, возвращает список комплектов, реализуемых завершением.

7.1.17 **Дескриптор ObservedEvents**

Дескриптор `ObservedEvents` предоставляется вместе с командой `Notify` для информирования MGC о том, какое(ие) событие(я) было(и) обнаружено(ы). При использовании совместно с командой `AuditValue`, дескриптор `ObservedEvents` возвращает информацию о событиях в буфере событий, о которых не было уведомления. `ObservedEvents` содержит `RequestID` дескриптора `Events`, запускающего уведомление, сообщение об обнаруженном(ых) событии(ях), дополнительно время обнаружения и любые параметры наблюдаемого события. Отчет о времени обнаружения производится с точностью до сотых долей секунды.

7.1.18 **Дескриптор Topology**

Дескриптор `Topology` используется для определения направлений потока между завершениями в контексте. В отличие от дескрипторов, описанных в предыдущих пунктах, дескриптор `Topology` применяется к контексту, а не к завершению. В заданной по умолчанию топологии контекста каждую передачу завершения получают все остальные завершения. Дескриптор `Topology` является дополнительным. При получении команды, содержащей дескриптор `Topology`, MG, не поддерживающий дескрипторы `Topology`, в качестве ответа возвращает код ошибки `Error Code 444` ("Неподдерживаемый или неизвестный дескриптор"), и дополнительно включает строку, содержащую имя неподдерживаемого дескриптора ("`Topology`") в тексте ошибки в дескрипторе `Error`.

Дескриптор `Topology` реализуется до выполнения команд. Возможно дать команду, содержащую только один дескриптор `Topology`, при условии, что контекст, к которому применяется команда, уже существует.

Дескриптор `Topology` состоит из последовательности связанных завершений данной формы ($T1$, $T2$, *соединение*[*StreamID*]). $T1$ и $T2$ обозначают завершения внутри контекста, возможно использующими групповые символы ALL или CHOOSE. Если используется дополнительное поле `StreamID`, соединение применяется только к отдельному потоку между $T1$ и $T2$, маркированному `StreamID`. Если поле `StreamID` пропускается, топология применяется ко всем потокам на завершении. *Соединение* определяет, каким образом медиа проходят между данными двумя завершениями следующим образом:

- ($T1$, $T2$, изоляция) означает, что завершения, подходящие $T2$, не получают медиа от завершений, подходящих $T1$, и наоборот.
- ($T1$, $T2$, односторонняя) означает, что завершения, подходящие $T2$, получают медиа от завершений, подходящих $T1$, но не наоборот. В данном случае следует использовать групповой символ ALL так, чтобы были разрешены завершения, подходящие либо $T1$, либо $T2$, но не $T1$, $T2$ вместе.
- ($T1$, $T2$, односторонняя внешняя) означает, что завершения, подходящие $T2$, получают медиа, отправленные извне завершениями, подходящими $T1$, но не наоборот. В данном случае групповой символ ALL к $T1$ не применяется.
- ($T1$, $T2$, односторонняя для обоих) означает, что завершения, подходящие $T2$, получают медиа, отправленные и полученные извне завершениями, подходящими $T1$, но не наоборот. В данном случае групповой символ ALL к $T1$ и/или $T2$ не применяется.
- ($T1$, $T2$, двусторонняя) означает, что завершения, подходящие $T2$, получают медиа от завершений, подходящих $T1$, и наоборот. В данном случае использовать групповые символы разрешается, так как имеются завершения, подходящие как $T1$, так и $T2$. Однако если существует завершение, подходящее им обоим, кольцевой проверки линии связи не проводится.

Групповые символы CHOOSE также могут использоваться в *T1* и *T2* с некоторыми ограничениями:

- действие (см. пункт 8), частью которого является дескриптор Topology, содержит команду Add, в которой используется групповой символ CHOOSE;
- если групповой символ CHOOSE возникает в *T1* или *T2*, неполное имя не должно задаваться.

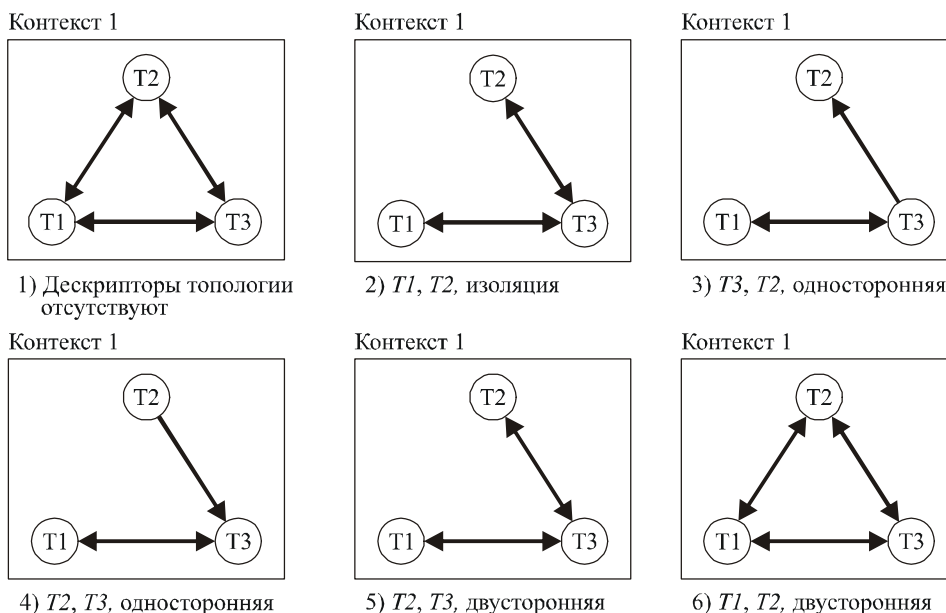
Групповой символ CHOOSE в дескрипторе Topology совпадает с TerminationID, назначаемым MG в первой команде Add, использующим групповой символ CHOOSE в том же действии. Существующее завершение, подходящее *T1* или *T2*, в контексте, к которому добавляется завершение, соединяется в вновь добавленным завершением так, как определено в дескрипторе Topology. Если завершение не упоминается в дескрипторе Topology, любая топология, связанная с ним, остается неизменной. Однако если в контекст добавляется новое завершение, по умолчанию его связь с другими завершениями в данном контексте является двусторонней, если для изменения данной ситуации не предоставляется дескриптор Topology (например, если в контекст с *T1* и *T2* с топологией (*T3*, *T1*, односторонняя) добавляется завершение *T3*, с завершением *T2* оно будет соединено двусторонней связью).

Если данная топология применяется к одному отдельному потоку (*T1*, *T2*, соединение, StreamID), топология других потоков между завершениями не меняется.

Дескриптор Topology не должен включать в себя комбинацию соединений между двумя завершениями (*T_i*, *T_j*) с и без дополнительного поля StreamID во избежание неопределенного режима работы. Например, (*T1*, *T2*, двусторонняя) и (*T1*, *T2*, изолированная, S1) не должны появляться в одном дескрипторе. После получения такого дескриптора Topology, MG должен ответить сообщением об ошибке, включая код ошибки Error Code 421 ("Неизвестное действие или недопустимая комбинация действий").

Одностороннее соединение должно реализовываться таким образом, чтобы другим завершениям в контексте не было известно об изменении в топологии.

На рисунке 7, таблице после него и рисунке 8 показаны несколько примеров результата включения дескрипторов Topology в действия. В данных примерах подразумевается, что дескрипторы Topology применяются последовательно. На рисунках 9 и 10 показаны отдельные примеры, показывающие особые эффекты установки односторонней внешней и односторонней для обеих топологий.



ПРИМЕЧАНИЕ. – Направление стрелки показывает направление потока.

H.248.1_F07

Рисунок 7/Н.248.1 – Примеры топологий

Топология	Описание
1	Дескрипторов Topology нет Когда дескрипторы Topology не включаются, все завершения имеют двустороннее соединение со всеми остальными завершениями.
2	<i>T1, T2</i> Изоляция Соединение между <i>T1</i> и <i>T2</i> устраняется. <i>T3</i> имеет двустороннее соединение как с <i>T1</i> , так и с <i>T2</i> . <i>T1</i> и <i>T2</i> имеют двустороннее соединение с <i>T3</i> .
3	<i>T3, T2</i> односторонняя Одностороннее соединение <i>T3</i> и <i>T2</i> (т. е. <i>T2</i> получает поток медиа от <i>T3</i>). Двустороннее соединение между <i>T1</i> и <i>T3</i> .
4	<i>T2, T3</i> односторонняя Одностороннее соединение между <i>T2</i> и <i>T3</i> . <i>T1</i> и <i>T3</i> соединены двусторонней связью
5	<i>T2, T3</i> двусторонняя <i>T2</i> соединен с <i>T3</i> двусторонней связью. Результат этого – такой же, как в Топологии 2.
6	<i>T1, T2</i> двусторонняя (<i>T2, T3</i> двусторонняя и <i>T1, T3</i> двусторонняя могут подразумеваться или выражаться в явном виде). Все завершения имеют двустороннее соединение со всеми остальными завершениями.

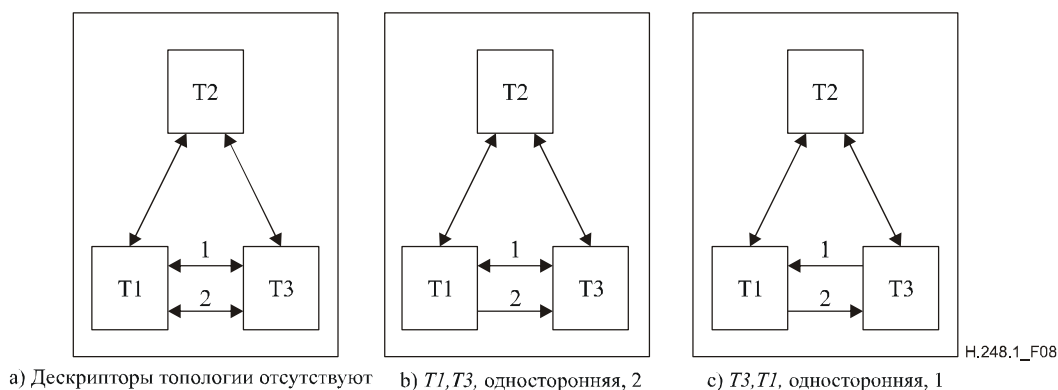


Рисунок 8/Н.248.1 – Пример топологии на уровне потока

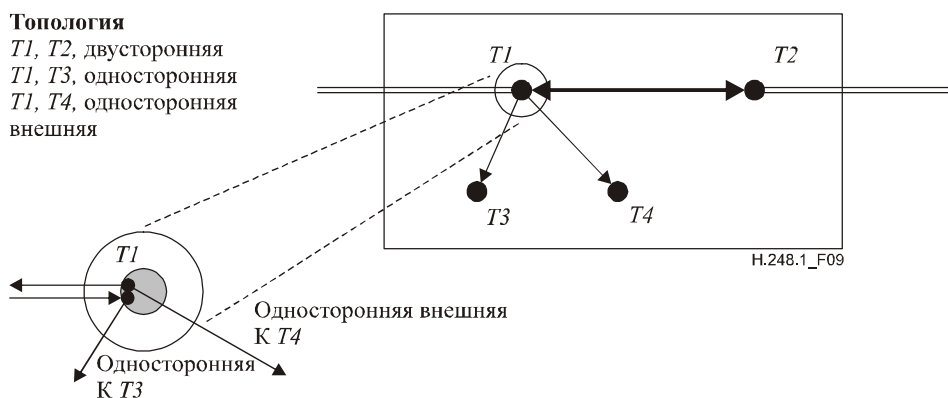


Рисунок 9/Н.248.1 – Односторонняя внешняя в сравнении с односторонней топологией

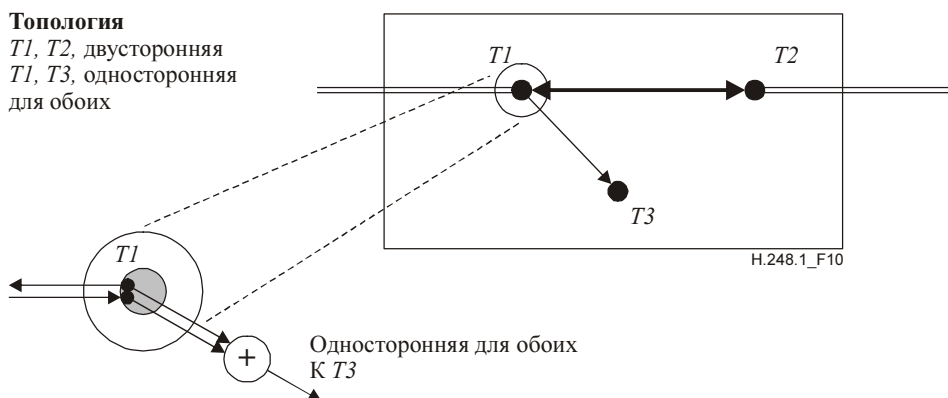


Рисунок 10/Н.248.1 – Функционирование топологии, односторонней для обоих

7.1.19 Дескриптор ContextAttribute

Дескриптор ContextAttribute используется для определения свойств (определенных в комплексах), которые относятся к контексту в целом и применяются к контексту, а не к завершению. Свойства завершений в дескрипторе ContextAttribute не действительны. Дескриптор ContextAttribute находится до команд в действии. Возможно, что действие будет содержать только дескриптор ContextAttribute при условии, что указанный контекст уже существует. Значения свойств контекста могут быть недоопределенными как описано в п. 7.1.1.

Новые установочные параметры дескриптора ContextAttribute полностью заменяют предыдущие установочные параметры данного дескриптора в MG. Таким образом, чтобы сохранить информацию от предыдущих установочных параметров, MGC должен включить эту информацию в новые установочные параметры. Если MGC желает удалить некоторую информацию из существующего дескриптора, он просто заново посылает данный дескриптор, исключая ненужную информацию. Включение дескриптора ContextAudit или ContextAttribute в действие, содержащее только команду AuditValue или AuditCapabilities, не создает новых установочных параметров.

Параметр ContextIDList используется для создания компактного списка идентификаторов ContextID. Для того чтобы запросить использование параметра ContextIDList в ответе, следует включить параметр ContextIDList в соответствующий запрос.

7.1.20 Дескриптор Error

Если при обработке запроса транзакции отвечающий получает ошибку, он должен включить в свой ответ дескриптор Error. Запрос Notify может также содержать дескриптор Error.

Дескриптор Error содержит код ошибки, зарегистрированный в организации IANA, дополнительно может присутствовать текст ошибки. Действительные коды ошибок и описания ошибок содержатся в Рек. МСЭ-Т Н.248.8.

Дескриптор Error должен быть определен на "самом глубоком уровне", семантически приемлемым для описываемой ошибки, и по возможности должен быть дан разбор любых проблем с исходным запросом. Дескриптор Error может ссылаться на синтаксический структурный компонент, в котором он не находится. Например, описание ошибки 422 ("Синтаксическая ошибка в Действии"), может появиться в команде, даже если ошибка ссылается на более крупный структурный компонент – действие, а не на определенную команду, внутри которой появляется ошибка.

7.2 Программный интерфейс приложения команд

Нижеследующий текст является программным интерфейсом приложения (Application Programming Interface (API)), где описываются команды протокола. Данный API приводится для иллюстрации команд и их параметров и не предназначен для уточнения их выполнения (например, путем использования блокирующих функциональных соединений). После имени команды в круглых скобках описываются входные параметры, а перед командой – их возвращаемые значения. Это делается только с целью описания; действительный синтаксис команд и кодирование детально излагаются в последующих подпунктах. Порядок следования параметров команд не фиксирован. В

качестве параметров команд дескрипторы могут появляться в любом порядке. Дескрипторы должны обрабатываться в том порядке, в котором они появляются.

Любой ответ на команду может содержать дескриптор Error; в API специально этого не показано.

Все параметры, заключенные в квадратные скобки ([...]), считаются дополнительными.

7.2.1 Add

Команда Add добавляет завершение к контексту.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

 [, StatisticsDescriptor]

)

(*) В версии 2 Рек. Н.248.1 (05/2002) дескриптор Modem не рассматривается.

TerminationIDList задает одно или более завершений, которые следует добавить к контексту. Это(и) завершение(я) либо создаются, либо извлекаются из нулевого контекста. Если в TerminationID используется групповой символ CHOOSE, будет возвращен выбранный TerminationID. В команде Add могут использоваться групповые символы ALL, однако такое использование будет несвойственным. Если групповой символ дает совпадение более чем с одним TerminationID, попытки предпринимаются по всем совпадениям, отчет предоставляется по каждой попытке. Очередность попыток при многочисленных TerminationID не определена.

Дополнительный дескриптор Media описывает все потоки медиа.

Дополнительный дескриптор Mux определяет мультиплексор, если он используется. Для удобства если в команде Add присутствует дескриптор Multiplex, в котором перечисляются все завершения, в данный момент не находящиеся в контексте, такие завершения добавляются в контекст так, как если бы к ним применялись отдельные команды Add, перечисляющие данные завершения. Если при выполнении такой команды Add возникает ошибка, будет возвращена ошибка 471 ("Сбой команды Implied Add для Multiplex"), и дальнейшая обработка команды прекратится.

Параметр дескриптора Events является необязательным. Если он присутствует, предоставляется список событий, которые необходимо распознавать на завершении.

Параметр дескриптора EventBuffer является необязательным. Если он присутствует, предоставляется список событий, которые MG должен обнаруживать и заносить в буфер, когда значение EventBufferControl равняется LockStep.

Параметр дескриптора Signals является необязательным. Если он присутствует, предоставляется список сигналов, которые следует применить к завершению.

Параметр дескриптора DigitMap является необязательным. Если он присутствует, он задает определение DigitMap, которое можно использовать в дескрипторе Events.

Дескриптор Audit является необязательным. Если он присутствует, команда возвратит значения любых дескрипторов/ свойств/ сигналов/ событий/ статистики, определенных в дескрипторе Audit.

Дескриптор Statistics является необязательным. Если он присутствует, он обеспечивает статистику по завершению или потоку, которую желает получить MGC. Каждый раз, когда устанавливается статистика, статистические значения сбрасываются.

Все дескрипторы, которые могут быть модифицированы, могут быть возвращены MG, если параметр недоопределен или переопределен. Дескрипторы ObservedEvents, Statistics, Packages и EventBuffer возвращаются, только если дескриптором Audit производится запрос.

Команда Add не должна использоваться на завершении, serviceState которого – "OutOfService".

7.2.2 Modify

Команда Modify изменяет свойства завершения.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Modify(TerminationIDList

[, MediaDescriptor]

[, ModemDescriptor] (*)

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

[, StatisticsDescriptor]

)

(*) Дескриптор Modem в версии 2 Рек. Н.248.1 (05/2002) не рассматривается.

В TerminationIDList определяются завершения, которые следует изменить. TerminationID может быть заданным, если требуется изменить одно завершение к контексте. Для некоторых операций может применяться использование групповых символов в TerminationID. Если групповой символ дает совпадение более чем с одним завершением, делаются попытки по всем совпадениям, отчет

предоставляется по каждой из них. Порядок следования попыток при совпадении с несколькими TerminationID не задается. Применение CHOOSE TerminationID является ошибкой, поскольку команда Modify может использоваться только для существующих завершений.

Для удобства если в команде Modify используется дескриптор Multiplex, тогда:

- если новый дескриптор Multiplex задает список любых завершений, в данный момент отсутствующих в контексте, такие завершения добавляются в контекст так, как если бы применялись отдельные команды Add, перечисляющие данные завершения;
- если любые завершения, ранее перечисленные в дескрипторе Multiplex Descriptor, больше не присутствуют в новом дескрипторе Multiplex, они вычитаются из контекста так, как если бы применялись отдельные команды Subtract, перечисляющие данные завершения.

Остальные параметры для команды Modify – такие же, как параметры для команды Add. Возможные возвращаемые значения – такие же, как для Add.

7.2.3 Subtract

Команда Subtract отсоединяет завершение от контекста и высылает статистику по участию завершения в контексте.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Subtract(TerminationIDList

[, AuditDescriptor]

)

(*) Дескриптор Modem в версии 2 Рек. Н.248.1 (05/2002) не рассматривается.

TerminationIDList в входных параметрах отражает вычитаемое(ые) завершение(я). TerminationID может быть задан или может содержать групповой символ замены, обозначающий, что следует вычесть все (или набор связанных) завершения в контексте команды Subtract. Если групповой символ дает совпадение более чем с одним TerminationID, делаются попытки по всем совпадениям, отчет предоставляется по каждой попытке. Очередность попыток при совпадении с несколькими TerminationID не задается.

Использование CHOOSE в TerminationID является ошибкой, поскольку команда Subtract может использоваться только для существующих завершений.

В команде Subtract групповой символ ALL может использоваться в качестве ContextID также как TerminationID, что будет иметь эффект удаления всех контекстов, удаления всех непродолжительных завершений и возвращения всех физических завершений в нулевой контекст. Вычитание завершения из нулевого контекста не разрешается.

Для удобства если объектом команды Subtract является мультиплексное завершение, все основные завершения, перечисленные в дескрипторе Multiplex, вычитаются из контекста так, как если бы применялись отдельные команды Subtract, перечисляющие данные завершения.

По умолчанию, если не происходит замещения, происходит возвращение дескриптора Statistics как на уровне завершений, так и на уровне потока для передачи информации, собранной на завершении или

завершениях, заданных в команде. Информация, по которой производится отчет, касается существования завершения или завершений в контексте, из которого оно или они вычитаются.

Дескриптор Audit является необязательным. Если он присутствует, команда возвратит только элементы, заданные в дескрипторе Audit, который может быть пустым. Если дескриптор Audit пропускается, по умолчанию возвращается дескриптор Statistics. Возможные возвращаемые значения – такие же, как и для Add.

Когда из контекста вычитается инициализированное завершение, его значения должны быть возвращены к:

- значению, заданному по умолчанию, если это задано для данного дескриптора и не было замещено в ходе инициализации;
- в противном случае – к инициализированному значению.

7.2.4 Move

Команда Move перемещает завершение в другой контекст из текущего контекста за одну атомарную операцию. Команда Move является единственной командой, которая ссылается на завершение в контексте, отличное от того, к которому применяется команда. Команду Move не следует использовать для перемещения завершений в или из нулевого контекста.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Move(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

 [, StatisticsDescriptor]

)

(*) Дескриптор Modem в версии 2 Рек. Н.248.1 (05/2002) не рассматривается.

Завершение(я), которое(ые) следует переместить, задаются в TerminationIDList. Идентификаторы TerminationID могут быть обобщенными, однако использовать групповой символ CHOOSE нельзя. Если групповой символ замены дает совпадение более чем с одним TerminationID, выполняются попытки по всем совпадениям, отчет предоставляется по всем попыткам. При многочисленных совпадениях TerminationID порядок осуществления попыток не задается. Контекст, в который перемещается завершение, в действии обозначается указателем цели ContextID. Если из контекста перемещается единственное оставшееся завершение, контекст удаляется.

Команда Move не влияет на свойства завершения, на которое она направлена, кроме свойств, открыто изменяемых дескрипторами, включенными в команду Move. Например, перед выполнением команды Move дескрипторы Audit и Statistics возвратят статистику по завершению. Возможные дескрипторы, возвращаемые от команды Move – такие же, как для Add.

Для удобства если объектом команды Move является мультиплексное завершение, тогда также происходит перемещение всех основных завершений, перечисленных в дескрипторе Multiplex, как если бы применялись отдельные команды Move, перечисляющие завершения.

Не следует использовать команду Move для завершений, serviceState которых – "OutOfService".

7.2.5 AuditValue

Команда AuditValue возвращает текущие значения свойств, событий, сигналов и статистики, связанных с завершениями. Команда AuditValue может осуществлять запрос по содержимому дескриптора или по отдельному свойству, событию, сигналу или статистике. Команда AuditValue может быть полезной для поддержания синхронизации завершений между MGC и MG. AuditValue может сделать запрос, чтобы производилась сортировка возвращаемых значений согласно определенным критериям.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
AuditValue(TerminationIDList,  
           AuditDescriptor  
           )
```

(*) Дескриптор Modem в версии 2 Рек. Н.248.1 (05/2002) не рассматривается.

TerminationIDList задает последовательность идентификаторов TerminationID, которые могут быть заданными или обобщенными. Если групповой символ дает совпадение более чем с одним TerminationID, выполняются попытки по всем совпадениям, отчет производится по всем попыткам. При множественных совпадениях идентификаторов TerminationID последовательность осуществления попыток не задается. Если производится запрос обобщенного ответа, генерируется только один ответ на команду, причем он содержит объединение значений всех данных завершений, имеющих совпадение с групповым символом или списком. Данное соглашение может сократить объем данных, требующихся для проверки группы завершений. Использование группового символа CHOOSE является ошибкой.

Может осуществляться проверка отдельных свойств, сигналов, событий и статистики. Возвращаемые значения могут быть рассортированы согласно определенным критериям.

- Запрос о проверке дескриптора может быть сделан при помощи идентификации требуемого дескриптора в дескрипторе Audit без указания дополнительной информации.
- Для проверки отдельного свойства в дескриптор Media включаются релевантные StreamID (дополнительный), GroupID (дополнительный) и PropertyID. Возвращается текущее значение свойства. В случае, когда используется флаг ReserveGroup дескриптора LocalControl, используется идентификатор GroupID. Идентификатор GroupID 1 соответствует первой зарезервированной группе (описание сессии), GroupID 2 – следующей группе, и т. д.

- Для проверки сигнала предоставляются релевантные SignalListID и/ или SignalID. StreamID и/ или RequestID являются необязательными. Только если сигнал является активным, возвращаются значения всех параметров сигнала, включая: индикацию KeepActive, тип сигнала, продолжительность, индикацию завершения сигнала и свойства, определяемые комплектом.
- Для проверки события предоставляются релевантные StreamID (дополнительный), EventID и RequestID (дополнительный). Возвращаются значения всех параметров события, включая: действия события и параметры, определяемые комплектом.
- Для проверки статистики на уровне завершения предоставляются отличительные черты статистики. Для проверки статистики на уровне потока в дескриптор Media включаются релевантные StreamID (дополнительный), GroupID (дополнительный) и StatisticID. Возвращается текущее значение статистики. Статистические данные не сбрасываются.
- Для проверки комплекта предоставляются отличительные черты и версия комплекта. Все свойства, сигналы, события и статистика, определенные в данном конкретном комплекте возвращаются со своими текущими значениями.
- Для сортировки возвращаемых значений проверки в запрос по проверке включаются критерии отбора. Можно включить несколько критериев выбора. Когда включается несколько критериев отбора, можно также включить логические операции AND (и) или OR (или) для обозначения того, как следует интерпретировать данные критерии отбора. Если не включается никакой логической операции, подразумевается использование логической операции AND (и).

Можно проверять многочисленные отдельные элементы в одном запросе.

Если запрашивается проверка дескриптора, в результате команды AuditValue возвращаются соответствующие дескрипторы с текущими значениями для данного завершения. Значения, возникающие на нескольких ступенях дескриптора, по определению являются чередующимися поддерживаемыми значениями, причем каждый параметр в дескрипторе считается независимым.

ObservedEvents возвращает список событий в EventBuffer. Если проверка дескриптора ObservedEvents осуществляется, в то время как активен DigitMap, возвращаемый дескриптор ObservedEvents также включает в себя событие завершения DigitMap, показывающее текущую строку набора, но не показывающее метода завершения.

EventBuffer возвращает набор событий и связанных значений параметра, в данный момент активированных в дескрипторе EventBuffer. Дескриптор Packages возвращает список комплектов, реализуемых завершением. Дескриптор DigitMap возвращает имя или значение текущего DigitMap для завершения. DigitMap, запрашиваемый в команде AuditValue с TerminationID ALL возвращает все DigitMaps на шлюзе. Statistics возвращает текущие значения всей статистики, ведущейся на уровне завершения. Проверка дескриптора Media возвратит любую статистику на уровне потока. Задание пустого дескриптора Audit приводит к тому, что возвращается лишь TerminationID. При использовании групповых символов замены, это может быть полезным для получения списка идентификаторов TerminationID. В Приложениях А и В описывается специальный синтаксис для представления такого списка в сжатом виде так, что тэг команды AuditValue не нужно повторять для каждого TerminationID.

Результаты команды AuditValue зависят от контекста, а именно определенного, нулевого или обобщенного. (Заметьте, что ContextID ALL не включает в себя нулевой контекст.) TerminationID может быть задан или обобщен.

Далее приводятся примеры возвращаемых значений в случае, когда контекст и/ или завершение обобщены или был задан обобщенный ответ.

Предположим, что на шлюзе имеются 4 завершения: t1/1, t1/2, t2/1 и t2/2. Предположим, что Завершения t1/* реализуют Комплекты aaa и bbb, а Завершения t2/* реализуют Комплекты ccc и ddd. Предположим, что Контекст 1 содержит t1/1 и t2/1, а Контекст 2 содержит t1/2 и t2/2.

Команда:

```
Context=1 {AuditValue=t1/1 {Audit {Packages}}}
```

ответ:

```
Context=1 {AuditValue=t1/1 {Packages {aaa,bbb}}}
```

Команда:

```
Context=* {AuditValue=t2/* {Audit {Packages}}}
```

ответ:

```
Context=1 {AuditValue=t2/1 {Packages {ccc,ddd}}},  
Context=2 {AuditValue=t2/2 {Packages {ccc,ddd}}}
```

Команда:

```
Context=* {W-AuditValue=t1/* {Audit {Packages}}}
```

ответ:

```
Context=* {AuditValue=t1/* {Packages {aaa,bbb}}}
```

ПРИМЕЧАНИЕ. – Обобщенный ответ может также использоваться для других команд таких, как Subtract.

В случае обобщенного контекста с Root, MGC может показывать, что он хочет получить компактный список идентификаторов ContextID, а не разбивать каждый контекст на отдельные ответы по действиям.

Предположим, что шлюз имеет 4 контекста: 1, 2, 3, 4, каждый из которых содержит по два завершения (t1-t8). В нижеследующих примерах показывается, как MG будет отвечать на команду проверки:

Команда:

```
Context=* {AuditValue=Root {Audit {}}}
```

ответ:

```
Context=1 {AuditValue=Context {*}}, Context=2 {AuditValue=Context {*}},  
Context=3 {AuditValue=Context {*}}, Context=4 {AuditValue=Context {*}}
```

Или

```
Context=1 {AuditValue=t1 {},AuditValue=t2 {}},Context=2 {AuditValue=t3 {},AuditValue=t4 {}},C  
ontext=3 {AuditValue=t5 {},AuditValue=t6 {}},Context=4 {AuditValue=t7 {},AuditValue=t8 {}}
```

Команда:

```
Context=* {ContextAttr {ContextList={*}},AuditValue=Root {Audit {}}}
```

ответ:

```
Context=* {ContextAttr {ContextList={1,2,3,4}},AuditValue=Root {}}
```

Далее показано, как можно получить другую информацию при помощи команды AuditValue:

ContextID	TerminationID	Получаемая информация
Задан	Обобщен	Проверка совпадающих завершений в контексте
Задан	Задан	Проверка отдельного завершения в контексте
NULL	Root	Проверка состояния и событий медиашлюза
NULL	Обобщен	Проверка всех совпадающих завершений в нулевом контексте
NULL	Задан	Проверка отдельного завершения вне любого контекста
ALL	Обобщен	Проверка всех совпадающих завершений, не находящихся в нулевом контексте и в контексте, с которым они связаны
ALL	Root	Список всех идентификаторов ContextID (список ContextID может быть возвращен либо путем использования многочисленных ответов действия, каждый из которых содержит ContextID из списка, либо используя параметр ContextIDList). Метод ответа определяется наличием в запросе параметра ContextIDList.)
ALL	Задан	(Ненулевой) ContextID, в котором в данный момент существует данное завершение

7.2.6 AuditCapabilities

Команда AuditCapabilities возвращает возможные значения свойств, событий, сигналов и статистики, связанных с завершениями. Запрос AuditCapabilities может осуществляться по содержимому дескриптора или по отдельному свойству, событию, сигналу или статистике.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor](*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

AuditCapabilities(TerminationIDList,
AuditDescriptor)

(*) Дескриптор Modem в версии 2 Рек. Н.248.1 (05/2002) не рассматривается.

Может осуществляться проверка дескрипторов или отдельных свойств, сигналов, событий и статистики.

- Запрос проверки всего дескриптора может осуществляться путем идентификации требуемого дескриптора в дескрипторе Audit без указания дополнительной информации.
- Для проверки отдельных свойств в дескриптор Media включаются релевантный StreamID (дополнительный) и PropertyID. Возвращается список возможных значений свойства.
- Для проверки сигнала предоставляются релевантные SignalListID и/ или SignalID. Идентификаторы StreamID и/ или RequestID являются дополнительными. Возвращается список возможных значений, связанных с каждым параметром сигнала (включая свойства, определенные в комплекте). Индикация KeepActive, тип сигнала, продолжительность и индикация завершенности сигнала не возвращаются.
- Для проверки события предоставляются релевантные StreamID (дополнительный), EventID и RequestID (дополнительный). Возвращается список возможных значений, связанный с каждым параметром события (включая действия события и параметры, определенные в комплекте).

- Для проверки статистики на уровне завершения предоставляется идентичность статистики. Для проверки статистики на уровне потока в дескриптор Media включаются релевантные StreamID (дополнительный), GroupID (дополнительный) и StatisticID. Возвращаются возможные значения статистики. Статистические данные не сбрасываются.

Если запрашивается проверка дескриптора, в результате команды AuditCapabilities возвращаются соответствующие дескрипторы с возможными значениями для завершения. Дескрипторы могут повторяться там, где есть множество возможных значений.

Если запрашивается обобщенный ответ, генерируется только один ответ команды, причем в нем содержится объединение значений всех завершений списка, совпадающих с групповым символом замены. Это соглашение может сократить объем данных, требующихся для проверки группы завершений.

Если осуществляется проверка свойства, сигнала, события или статистики, в результате выполнения команды AuditCapabilities возвращаются соответствующие свойства, сигналы, события и статистика с возможностями завершения.

Интерпретация возможностей, запрашиваемых для различных значений идентификаторов ContextID и TerminationID, происходит так же, как в AuditValue.

Для значений свойства и параметра строки типа, строки символов или строки битов, MG должен вернуть пустое значение.

Для текстового кодирования, строки и символы возвращают пустой структурный элемент quotedString, в то время как байтовые строки возвращают NULL (0x00). Данный режим работы может быть замещен определением комплекта.

Дескриптор Events возвращает список возможных событий на завершении вместе со списком всех возможных значений для параметров дескриптора Events. Дескриптор EventBuffer возвращает ту же информацию, что и дескриптор Events. Дескриптор Signals возвращает список возможных сигналов, которые могут быть применены к завершению, вместе со списком всех возможных значений для параметров сигналов. Дескриптор Statistics возвращает имена статистики, ведущейся на завершении. Дескриптор ObservedEvents возвращает имена событий, активных на завершении. Дескрипторы DigitMap и Packages в AuditCapability не разрешены.

Далее приводится другая информация, которую можно получить при помощи команды AuditCapabilities:

ContextID	TerminationID	Получаемая информация
Задан	Обобщен	Проверка совпадающих завершений в контексте
Задан	Задан	Проверка одного завершения в контексте
NULL	Root	Проверка состояния и событий MG
NULL	Обобщен	Проверка всех совпадающих завершений в нулевом контексте
NULL	Задан	Проверка одного завершения вне всех контекстов
ALL	Обобщен	Проверка всех совпадающих завершений вне нулевого контекста и контекста, с которым они связаны
ALL	Root	Та же, что и для AuditValue
ALL	Задан	Та же, что и для AuditValue

7.2.7 Notify

Команда Notify позволяет медиашлюзу уведомить контроллер медиашлюза о событиях, возникающих на медиашлюзе.

TerminationID

```
Notify(TerminationID,  
      ObservedEventsDescriptor,  
      [ErrorDescriptor])
```

TerminationID задает завершение, посылающее команду Notify. Идентификатор TerminationID должен быть полностью заданным именем.

Дескриптор ObservedEvents содержит RequestID и список событий, обнаруженных медиашлюзом в том порядке, в котором они были обнаружены. Каждое событие в списке сопровождается параметрами, связанными с данным событием, и дополнительно индикацией времени обнаружения события. Процедуры для передачи команд Notify с RequestID равным нулю являются предметом дальнейших исследований.

Команды Notify с RequestID, неравным нулю, должны возникать только в результате обнаружения события, определенного в дескрипторе Events, который активен на рассматриваемом завершении.

RequestID возвращает параметр RequestID дескриптора Events, запустившего команду Notify. Он используется для установления соотношения между уведомлением и запросом, который активировал его. Запрос по событиям в списке должен быть сделан при помощи активации дескриптора Events или вложенного дескриптора Events, если RequestID не равен нулю (что является предметом дальнейшего изучения).

В команде Notify может быть послан дескриптор Error в результате Ошибки 518 ("Буфер событий полон").

7.2.8 ServiceChange

Команда ServiceChange позволяет медиашлюзу уведомлять контроллер медиашлюза о том, что завершение или группа завершений вскоре прекратят работу или только что возвратились к работе. Контроллер медиашлюза может указать, что завершения должны прекратить работу или вернуться к работе. Медиашлюз может уведомить MGC о том, что возможности завершения изменились. Это также позволяет MGC передать управление MG другому MGC.

TerminationIDList,

[ServiceChangeDescriptor]

```
ServiceChange(TerminationIDList,  
              ServiceChangeDescriptor  
              )
```

TerminationIDList определяет завершение(я), прекращающие работу или возвращающиеся к работе. Обобщение названий завершений разрешено, за исключением того, что алгоритм CHOOSE не должен использоваться. Использование "Root" TerminationID показывает, что ServiceChange оказывает влияние на весь медиашлюз.

ПРИМЕЧАНИЕ. – В первоначальной команде ServiceChange версии 1 Н.248.1 использование TerminationIDList не допустимо/ не верно/ не имеет силы.

7.2.8.1 Содержимое дескриптора ServiceChange

Дескриптор ServiceChange содержит следующие требуемые параметры:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeDelay;
- ServiceChangeAddress;

- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMgcID;
- TimeStamp;
- ExtensionParameter;
- ServiceChangeInfo;
- ServiceChangeIncompleteFlag.

7.2.8.1.1 ServiceChangeMethod

Параметр ServiceChangeMethod определяет тип ServiceChange, который возник или возникнет:

- 1) Graceful: показывает, что указанные завершения прекратят работу после определенной ServiceChangeDelay; на установленные соединения это не повлияет, однако контроллер медиашлюза должен воздержаться от установления новых соединений и ему следует попытаться деликатно удалить существующие соединения на завершении(ях), на которые влияет команда ServiceChange. По истечении задержки ServiceChangeDelay или удалении завершения из активного контекста (какой бы из них ни был первым) MG следует установить serviceState завершения на "out of service".
- 2) Forced: показывает, что указанные завершения резко прекратили работу, и все установленные соединения, связанные с ними, были утеряны. В случае с завершениями не – Root, за очистку контекста (если таковой присутствует), с которым связано вышедшее из строя завершение, отвечает MGC. Как минимум, завершение должно быть вычтено из контекста. serviceState данного завершения должно быть "out of service". В случае с завершениями Root, MGC может предположить, что были потеряны все соединения на MG, таким образом, он может счесть, что были вычтены все завершения.
- 3) Restart: показывает, что после истечения ServiceChangeDelay, работа на указанных завершениях будет восстановлена. ServiceState следует выставить на "inService" по истечении ServiceChangeDelay.
- 4) Disconnected: всегда применяется для Root TerminationID, показывает, что MG потерял связь с MGC, однако позднее был восстановлен на том же самом MGC (возможно после проверки других MGC из предварительно составленного списка). Так как состояние MG могло измениться, MGC может пожелать использовать команду Audit для повторной синхронизации своего состояния со шлюзами MG.
- 5) Handoff: посылается MG от MGC, данная причина показывает, что MGC прекращает работу, и должна быть установлена новая связь MGC. При передаче от MG – MGC, показывает, что MG пытается установить новую связь в соответствии с Handoff (переадресацией вызова), полученной от MGC, с которым он был соединен до этого.
- 6) Failover: посылается MGC от MG для указания на то, что исходный MG прекратил работу, а вторичный MG начинает работу. Данный метод ServiceChange также посылается MGC от MG, когда MG обнаруживает, что MGC вышел из строя.
- 7) Другое значение, смысл которого понимают MG и MGC.

7.2.8.1.2 ServiceChangeReason

Параметр ServiceChangeReason определяет причину, по которой возник или возникнет ServiceChange. Он состоит из буквенно – цифровой метки (зарегистрированной в IANA) и дополнительно пояснительной статьи.

7.2.8.1.3 ServiceChangeAddress и ServiceChangeMgcID

Дополнительный параметр ServiceChangeAddress определяет адрес (например, номер IP порта для IP сетей), который будет использоваться для последующих соединений. Он может быть задан в дескрипторе входного параметра или дескрипторе возвращенного результата. Параметры ServiceChangeAddress и ServiceChangeMgcID не должны присутствовать в дескрипторе ServiceChange или дескрипторе ServiceChangeResult одновременно. Параметр ServiceChangeAddress предоставляет адрес, который будет использован внутри контекста соединения, условия которого обсуждаются в

данный момент, в то время как ServiceChangeMgcID предоставляет дополнительный адрес, на котором MG должен пытаться установить другое соединение. Заметьте, что использование ServiceChangeAddress не поощряется. Контроллеры MGC и шлюзы MG должны уметь работать с ServiceChangeAddress, который может представлять собой как полный адрес, так и лишь номер порта в случае передачи TCP.

7.2.8.1.4 ServiceChangeDelay

Дополнительный параметр ServiceChangeDelay выражается в секундах. Если задержка отсутствует, или установлена на ноль, значение задержки считается равным нулю. В случае "Graceful" ServiceChangeMethod, нулевая задержка показывает, что контроллер медиашлюза должен ждать естественного удаления существующих соединений и не должен устанавливать новые соединения. Только для "Graceful" нулевая задержка означает, что MG не должен устанавливать ServiceState "OutOfService" пока завершение не окажется в нулевом контексте.

7.2.8.1.5 ServiceChangeProfile

Дополнительный параметр ServiceChangeProfile определяет Профиль (при наличии) поддерживаемого протокола. ServiceChangeProfile включает в себя версию поддерживаемого протокола. При отсутствии данного параметра подразумевается значение "NoProfile".

7.2.8.1.6 ServiceChangeVersion

Дополнительный параметр ServiceChangeVersion содержит версию протокола и используется, если происходит согласование версии протокола.

7.2.8.1.7 TimeStamp

Дополнительный параметр TimeStamp задает действительное время, по которому работает отправитель. Как таковое это время не обязательно является абсолютным временем, соответствующим, например, местному часовому поясу: оно лишь определяет произвольное время запуска, с которым должны сравниваться все будущие штампы времени, передаваемые отправителем во время данного соединения. Оно может использоваться отвечающим для определения того, насколько его представление о времени отличается от времени отсылающего. TimeStamp посылается с точностью до сотых долей секунды.

7.2.8.1.8 ExtensionParameter

Дополнительный параметр Extension может содержать любое значение, значение которого понимают MG и MGC. Значение "X-SC" резервируется для использования параметров сигнализации ServiceChange, которые были добавлены в версии 3 H.248.1 (и последующих версиях). Оно используется только в команде ServiceChange, закодированной первоначальной версией H.248.1, от MG к MGC, когда версия ServiceChangeVersion больше или равна 3. Структура значения определяется следующим ABNF:

$$X-SC = 1 * (NAME EQUAL paramValue [COMMA])$$

При добавлении новых параметров ServiceChange рекомендуется, чтобы имя/ значение было назначено, если необходимо передать данный параметр в первоначальной ServiceChange. Например:

Название: Строки до 64 символов без пробелов, начинающиеся с буквенного знака и состоящие из буквенно – цифровых знаков и/ или цифр, и, возможно, включающие в себя особый символ подчеркивания ("_").

Тип: Как в 12.1.2 "Свойства"

Возможные значения: Как в 12.1.2 "Свойства"

7.2.8.1.9 ServiceChangeInfo

Дополнительный параметр ServiceChangeInfo может содержать комплект/ свойство/ сигнал/ событие/ статистику причины, вызвавшей ServiceChange.

7.2.8.1.10 ServiceChangeIncompleteFlag

Данный дополнительный флаг показывает, что передача последующих команд ServiceChange от MG к MGC будет осуществляться с указанием на состояние завершений. Он используется только во время регистрации или перезапуска MG (ServiceChange на Root с ServiceChangeMethod Restart), когда MG желает сообщить MGC о состоянии всего MG и завершения. После получения данного флага MGC должен воздержаться от генерирования для MG команд, оперирующих с завершениями, отличными от Root. После передачи ServiceChangeIncompleteFlag в начальной команде регистрации/перезапуска, MG должен передавать его (флаг) во всех последующих командах ServiceChange до тех пор, пока MG не определит, что он предоставил отчет о текущем состоянии MG и его завершениях. Затем данный флаг должен быть удален из последней команды ServiceChange, показывающей состояние MG или его завершениях. После получения данной ServiceChange без ServiceChangeIncompleteFlag, MGC может снова генерировать команды для MG. Когда данный флаг передается в первоначальной команде ServiceChange (H.248.1 версия 1) с использованием параметра расширения ServiceChange "X-SC", используется следующее:

Название: SIC

Тип: логическое выражение

Возможные значения: ON: Флаг включается

ПРИМЕЧАНИЕ. – Значение OFF не требуется, поскольку последующие команды будут закодированы в версии 3 H.248.1.

7.2.8.1.11 ServiceChange Command and Response

Команда ServiceChange, задающая "Root" для TerminationID, и ServiceChangeMethod со значением Restart является командой регистрации, при помощи которой медиашлюз заявляет о своем существовании контроллеру медиашлюза. Медиашлюз может зарегистрироваться путем задания Root TerminationID и ServiceChangeMethod со значением Failover, если MG обнаруживает сбой в работе MGC. Сообщение, содержащее команду ServiceChange, задающую Root TerminationID и ServiceChangeMethod со значением Restart или Failover, не должны содержать других команд, так как данные команды должны использовать новый ServiceChangeAddress и согласованную версию протокола.

Предполагается, сто медиашлюз располагает названием одного основного и дополнительно нескольких дополнительных контроллеров медиашлюза.

Подтверждение приема команды ServiceChange завершает процесс регистрации, кроме случаев, когда MGC возвратил альтернативный ServiceChangeMgcID как описано ниже.

MG может задать транспортный ServiceChangeAddress, который следует использовать MGC для передачи сообщений, в параметре ServiceChangeAddress в входящем дескрипторе ServiceChange. MG может указать адрес в параметре ServiceChangeAddress запроса ServiceChange, а MGC может сделать то же самое в ответе ServiceChange. В любом случае, получатель должен использовать предоставленный адрес в качестве адреса назначения для всех последующих запросов для операций связи внутри соединения. В то же время, как показано в пункте 9, ответы для операций связи и индикации ожидания должны передаваться на адрес, с которого были посланы соответствующие запросы. Это должно выполняться, даже если предполагает передачу дополнительных сообщений из-за того, что команды и ответы не могут быть упакованы вместе.

Параметр TimeStamp должен передаваться вместе с командой регистрации и ответом на нее.

При регистрации MG может передать флаг ServiceChangeIncompleteFlag, если он определил, что MG находится в состоянии "InService", однако отдельные завершения могут находиться в состоянии "OutOfService". Это необходимо для того, чтобы предотвратить попытку захвата неработающих завершений. Последующие команды ServiceChange используются для предоставления отчета по завершениям "OutOfService".

Контроллер медиашлюза может вернуть параметр ServiceChangeMgcID, описывающий контроллер медиашлюза, к которым предпочтительно связываться для дальнейшего обслуживания медиашлюза. В таком случае медиашлюз должен заново активировать команду ServiceChange Command для нового контроллера медиашлюза. С MGC, определенным в ServiceChangeMgcID, если

он присутствует, необходимо связаться до всех дополнительных MGC. В сообщении переадресации вызова, посылаемого MG от MGC, ServiceChangeMgcID представляет новый MGC, к которому перейдут полномочия от текущего MGC.

Ответ на команду ServiceChange является пустым кроме случаев, когда используется Root TerminationID. В таком случае она включает в себя следующие параметры, как требуется:

- ServiceChangeAddress, если отвечающий MGC желает определить новое место назначения для сообщений от MG для оставшейся части соединения;
- ServiceChangeMgcID, если отвечающий MGC не желает поддерживать связь с MG;
- ServiceChangeProfile, если отвечающий желает согласовать профиль для использования для связи. Возврат профиля (название и версия) производится в ответе только в случае, когда MGC не может поддерживать профили, определенные в ServiceChangeRequest. В возвращаемом ответе должны быть указаны поддерживаемые профиль и версия или "NoProfile", если не поддерживается ни один из профилей. После получения профиля в ответе MG может продолжить связь с текущим MGC или связаться с дополнительными MGC и установить связь с ними. Если MGC в ответе возвратил профиль, отличающийся от профиля, предоставленного в запросе, MG должен:
 - a) продолжить контрольную связь путем активации новой команды ServiceChange Command с согласованным профилем для подтверждения MGC того, что MG согласен с данным профилем; или
 - b) поддерживать контрольную связь активной так, чтобы MGC использовал профиль, передаваемый в ServiceChange Reply; или
 - c) инициировать контрольную связь с другим MGC, используя исходный профиль.
- ServiceChangeVersion, если отвечающий желает согласовать версию протокола, используемую для связи.

ServiceChangeReasons определяются в Рек. МСЭ-Т Н.248.8. Данный список может быть расширен при помощи регистрации IANA, как описано в 14.3.

7.2.9 Изменение и проверка атрибутов контекста

Как отмечалось в предыдущих подпунктах, команды протокола применяются к завершениям. В данном пункте описывается, как производится изменение и проверка контекстов.

Действие может содержать инструкции по изменению и проверке контекста (см. пункт 8).

Запрос действия, посланный на MG, может включать в себя запрос по проверке атрибутов контекста. Существует два типа проверки:

Проверка значений (Value Audit): MGC может проверить определенный контекст для определения текущего значения отдельных свойств контекста. MGC может определять текущие значения для всех существующих (ненулевых) контекстов путем указания ContextID ALL в запросе Audit. Если атрибуты контекста добавляются или были изменены тем же действием, что и запрос Audit, значение(я) должны возвращаться после применения действия. Атрибуты контекста можно использовать в качестве критериев отбора для сортировки возвращаемых значений проверки. Можно включать несколько критериев отбора. При включении нескольких критериев отбора можно также включить логическую операцию AND (и) или OR (или) для указания на то, как следует интерпретировать критерии отбора. Если логические операции не включаются, подразумевается использование логической операции AND.

Проверка возможностей (Capabilities Audit): MGC может проверить определенный контекст для определения возможных значений, которые могут допустить отдельные свойства данного контекста. MGC может определить возможные значения для всего MG путем указания ContextID ALL в запросе проверки.

Далее приводится информация, которую можно получить при помощи Context Audit:

ContextID	TerminationID	AuditValue	AuditCapabilities
Задан	Не применяется	Значение свойства контекста в указанном контексте.	Возможные значения свойства контекста в указанном контексте.
NULL	Не применяется	Не разрешено	Не разрешено
ALL	Не применяется	Текущие значения для всех (ненулевых) контекстов путем указания ContextID ALL в Audit request. Ответ на ContextID ALL предоставляется посредством actionReply для контекста.	Ответ на ContextID ALL предоставляется с возможными значениями свойства контекста по всему MG.

Действие может также включать в себя запрос по изменению атрибутов контекста.

Свойства контекста, которые могут быть включены в ответ на действие, используются для возвращения информации MGC. Это может быть информация, запрашиваемая в ходе проверки атрибутов контекста, или детали эффекта изменения контекста.

Если MG получает действие, содержащее как запрос по проверке атрибутов контекста, так и запрос по изменению данных атрибутов, ответ должен включать в себя значения атрибутов после обработки запроса по изменению.

7.2.10 Общий синтаксис команд

Данный протокол может быть закодирован в двоичном формате или в текстовом формате. Контроллерам MGC следует поддерживать оба формата кодирования. Медиашлюзы могут поддерживать оба формата.

Синтаксис протокола для двоичного формата определяется в Приложении А. В Приложении С определяется кодирование дескрипторов Local и Remote для использования с двоичным форматом.

Полный ABNF текстового кодирования протокола по RFC 2234 дается в Приложении В. SDP используется в качестве кодирования дескрипторов Local и Remote для использования с текстовым кодированием как модифицированный в 7.1.8.

8 Транзакции

Команды, передаваемые между контроллером медиашлюза и медиашлюзом, группируются в транзакции, каждая из которых обозначается TransactionID. Транзакции состоят из одного или более действий. Действие состоит из непустой последовательности команд, изменений свойств контекста и проверок свойств контекста, ограниченных до оперирования в пределах одного контекста. Поэтому обычно каждая транзакция обычно задает ContextID. Однако есть два условия, при которых для транзакции отдельный ContextID не предоставляется. Одним из них является случай, когда происходит изменение завершения вне контекста. Другим условием является случай, когда контроллер запрашивает у шлюза создание нового контекста. На рисунке 11 иллюстрируется взаимосвязь между транзакцией, действием и командой.

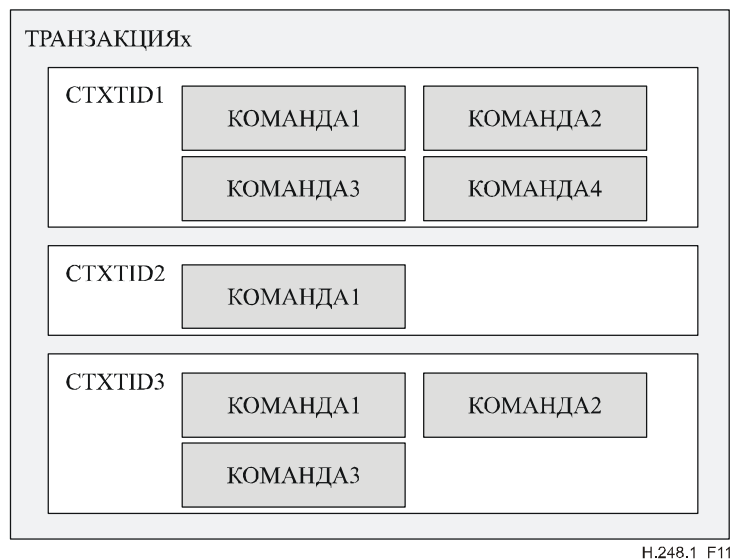


Рисунок 11/Н.248.1 – Транзакции, действия и команды

Транзакции предоставляются в виде TransactionRequests. Ответы на соответствующие TransactionRequest предоставляются в одном ответном сообщении, которому, возможно, предшествовали несколько сообщений TransactionPending (см. п. 8.2.3).

Транзакции гарантируют упорядоченную обработку команд. Это означает, что команды внутри транзакции выполняются последовательно. Упорядочивание транзакций не гарантировано; транзакции могут выполняться в любом порядке или одновременно; однако TransactionReplies должен быть выполнен до TransactionRequests, если оба они содержатся в одном сообщении.

При первом сбое в выполнении команды в транзакции, обработка оставшихся команд в данной транзакции прекращается. Если команда содержит обобщенный TerminationID, осуществляются попытки для каждого из действительных TerminationID, имеющих совпадение с групповым символом. В TransactionReply включается ответ по каждому совпадающему TerminationID, даже если одна или более ступеней генерировали ошибку. Если выполнение всех TerminationID, имеющих совпадение с групповым символом, закончилось ошибкой, попытки по всем командам, следующим за обобщенной командой, не осуществляются.

Команды могут быть помечены как "Optional" ("Дополнительная"), что может данный режим работы – если выполнение команды, помеченной как дополнительная, заканчивается ошибкой, последующие команды в транзакции выполняться будут. Если команда дает сбой, перед тем, как продолжить обработку команды, MG должен восстановить состояние, существовавшее до попытки выполнения команды, насколько это возможно.

TransactionReply включает в себя результаты для всех команд соответствующего TransactionRequest. TransactionReply включает в себя ответные значения для команд, выполненных успешно, и команду и дескриптор Error для любых команд, давших сбой. Ответы на команды приходят в той же последовательности, в какой они были в соответствующем TransactionRequest. TransactionPending используется для того, чтобы периодически уведомлять получателя о том, что транзакция еще не завершилась, но активно обрабатывается.

Приложениям следует приводить в исполнение таймер прикладного уровня для каждой транзакции. Истечение времени работы таймера должно вызывать повторную передачу запроса. При получении Reply таймер должен быть остановлен. При получении Pending таймер должен быть перезапушен.

8.1 Общие параметры

8.1.1 Идентификаторы транзакций

Идентификация транзакций происходит при помощи TransactionID, назначаемого отправителем, который является уникальным в области действия отправителя. Для того чтобы отправить ответное

сообщение, содержащее дескриптор Error Descript, или указать, что в запросе пропущен TransactionID, необходимо использовать TransactionID со значением ноль в соответствующем TransactionReply.

8.1.2 Идентификаторы контекстов

Идентификация контекстов происходит при помощи ContextID, назначаемого медиашлюзом, который является уникальным в области действия медиашлюза. Контроллер медиашлюза должен использовать ContextID, предоставленный медиашлюзом, во всех последующих транзакциях, относящихся к данному контексту. Протокол обращается к отмеченному значению, которое может быть использовано контролером медиашлюза при ссылке на завершение, в данный момент не связанное с контекстом, а именно *NULL* ContextID.

Для выполнения запроса о том, чтобы медиашлюз создал новый контекст, используется групповой символ CHOOSE.

Для обращения ко всем контекстам на MG MGC может использоваться групповой символ ALL. При использовании группового символа ALL *NULL* Context не включается.

MGC не должен использовать частично определенные ContextID, содержащие групповые символы CHOOSE или ALL.

8.2 Программный интерфейс приложения транзакций

Далее приводится программный интерфейс приложения (Application Programming Interface (API)), где описываются транзакции данного протокола. Данный API иллюстрирует транзакции и их параметры и не предназначен для определения их выполнения (например, путем использования блокирующих соединений функции). В нем описываются входные параметры и возвращаемые значения, которые будут использоваться различными транзакциями данного протокола с очень высокого уровня. Синтаксис транзакций и кодировки описываются в последующих подпунктах.

8.2.1 TransactionRequest

TransactionRequest запускается отправителем. За один запрос может быть запущена только одна транзакция. Запрос содержит одно или более действий, каждое из которых определяет свой целевой контекст.

```
TransactionRequest(TransactionID {  
    ContextID {Command ... Command},  
    ...  
    ContextID {Command ... Command} })
```

Параметр TransactionID должен определять значение для последующего согласования с ответным сообщением TransactionReply или TransactionPending от получателя.

Параметр ContextID должен определять значение, которое будет принадлежать всем последующим командам до следующего назначения параметра ContextID или до конца TransactionRequest, в зависимости от того, что произойдет раньше.

Параметр команды представляет одну из команд, упомянутых в пункте 7.2 ("Программный интерфейс приложения команд").

8.2.2 TransactionReply

TransactionReply запускается получателем. За одну транзакцию запускается только один ответ. Ответ содержит одно или более действий, в каждом из которых должен быть определен целевой контекст и один или более Responses на контекст. Ответ может быть разделен на несколько сообщений. TransactionReply запускается отвечающим после проведения обработки TransactionRequest.

TransactionRequest считается обработанным:

- когда были обработаны все действия в данном TransactionRequest; или
- когда при обработке TransactionRequest произошла ошибка, кроме случаев, когда ошибочной является дополнительная команда.

Если сообщение содержит больше TransactionRequest, чем получатель может обработать, в качестве ответа получатель возвратит дескриптор Error 413 ("Число транзакций в сообщении превышает максимальное значение").

Команда считается обработанной, когда завершилась обработка всех дескрипторов в данной команде.

Дескриптор Signals считается обработанным, когда было установлено, что дескриптор правилен с точки зрения синтаксиса, запрашиваемые сигналы поддерживаются, и они были выстроены в очередь для выполнения.

Дескрипторы Events или EventBuffer считаются обработанными, когда было установлено, что дескриптор правилен с синтаксической точки зрения, запрашиваемые события можно наблюдать, любые вложенные сигналы могут быть сгенерированы, любые вложенные события могут быть обнаружены, и MG был приведен в состояние, в котором будет происходить обнаружение данных событий.

```
TransactionReply(TransactionID {
    ContextID {Response ... Response},
    ...
    ContextID {Response ... Response}})
```

Параметр TransactionID должен быть таким же, как и в соответствующем TransactionRequest.

Параметр ContextID должен задавать значение, которое будет принадлежать всем Response для данного действия. ContextID может быть задан, ALL или NULL.

Каждый из параметров Response представляет собой возвращаемое значение как описано в 7.2 или дескриптор Error, если при выполнении команды произошла ошибка. Команды, находящиеся после точки сбоя, не обрабатываются и, поэтому Response для них не отправляются.

Исключением этого являются случаи, когда в TransactionRequest команда была отмечена как дополнительная. Если ошибка случается при выполнении дополнительной команды, выполнение транзакции продолжается, поэтому в данном случае ответ будет содержать Responses после Error.

Генерирование дескрипторов Error описывается пункте 7.1.20 ("Дескриптор Error"). Ниже описываются некоторые отдельные случаи.

Если при обработке ContextID получателем происходит ошибка, однако получатель может проанализировать ContextID, запрашиваемый ответ на действие будет состоять из ContextID и одного дескриптора Error, 422 ("Синтаксическая ошибка в Действии"). Если получатель не может проанализировать ContextID, он возвратит TransactionReply, состоящий из TransactionID и одного дескриптора Error, 422 (Синтаксическая ошибка в Действии).

Если получатель сталкивается с ошибкой так, что он не может определить допустимое действие, он возвратит TransactionReply, состоящий из TransactionID и одного дескриптора Error, 422 ("Синтаксическая ошибка в Действии"). Если конец действия не может быть определен с большой вероятностью, однако можно проанализировать одну или более команд, получатель обработает данные команды и затем отошлет 422 ("Синтаксическая ошибка в Действии") в качестве последнего действия в транзакции. Если получатель сталкивается с ошибкой так, что он не может определить допустимую транзакцию, он возвратит TransactionReply с NULL TransactionID и одним дескриптором Error 403 ("Синтаксическая ошибка в TransactionRequest").

Если конец транзакции не может быть определен с большой степенью вероятности, и можно проанализировать одно или более действий, получатель обработает данные действия и затем возвратит 403 ("Синтаксическая ошибка в TransactionRequest") в качестве последнего действия в транзакции. Если невозможно проанализировать ни одного действия, в качестве единственного ответа будет послана 403 ("Синтаксическая ошибка в TransactionRequest").

Если определить TerminationID с большой степенью вероятности невозможно, в качестве ответа на действие будет отправлена 442 ("Синтаксическая ошибка в Команде").

Если определить конец команды с большой степенью вероятности невозможно, в качестве ответа на последнее действие, которое можно проанализировать, будет отправлена 442 ("Синтаксическая ошибка в Команде").

8.2.3 TransactionPending

TransactionPending запускается получателем. TransactionPending показывает, что происходит активная обработка транзакции, и что она еще не закончена. TransactionPending используется для того, чтобы отправитель не мог предположить, что TransactionRequest был потерян, в случаях, когда для завершения транзакции требуется некоторое время.

TransactionPending(TransactionID { })

Параметр TransactionID должен быть таким же, как и в соответствующем TransactionRequest. Свойство Root Termination (normalMGExecutionTime) устанавливается MGC для обозначения интервала, в котором MGC ожидает ответа на любую транзакцию от MG (за исключением задержки сети). Другое свойство (normalMGCEExecutionTime) устанавливается MGC для обозначения интервала, в котором MG следует ожидать ответа на любую транзакцию от MGC (за исключением задержки сети). MGProvisionalResponseTimerValue обозначает время, в рамках которого MGC следует ожидать Pending Response от MG в случае, если транзакция не может быть завершена (первоначально установлено на normalMGExecutionTime плюс задержка сети, но может быть снижено). MGCProvisionalResponseTimerValue имеет аналогичное значение для MG. Отправители могут получать более одного TransactionPending за команду. Если во время ожидания получен двойной запрос, отвечающий может отправить двойное ожидание незамедлительно или продолжить ожидание момента, когда его таймер запустит другой TransactionPending.

Свойство Root Termination (MGOriginatedPendingLimit) устанавливается MGC для обозначения числа TransactionPending, которые могут быть получены от MG. Когда происходит превышение значения, выраженного данным свойством, MG должен прекратить обработку транзакции и отослать TransactionReply, в противном случае MGC может предположить, что транзакция была неверной.

Другое свойство Root Termination (MGCOrganatedPendingLimit) устанавливается MGC для обозначения числа TransactionPending, которые могут быть получены от MGC. Когда происходит превышение значения, выраженного данным свойством, MGC должен прекратить обработку транзакции и отослать TransactionReply, в противном случае MG может предположить, что транзакция была неверной.

Превышение xxxOriginatedPendingLimit (MGOriginatedPendingLimit или MGCOrganatedPendingLimit) может произойти либо из-за долгой обработки команды, либо из-за ошибки (например, команда вызвала цикл). В обоих случаях получатель исходного TransactionRequest отправит TransactionReply с дескриптором Error в качестве параметра ответа, указав причиной либо продолжительное выполнение команды, либо команду, вызвавшую ошибку соответственно. Последующие команды в транзакции обрабатываться не будут. Необходимо использовать Ошибку Error 506 (Произошло превышение числа TransactionPendings).

ПРИМЕЧАНИЕ. – Для предотвращения ситуации, когда происходит превышение xxxOriginatedPendingLimit (MGOriginatedPendingLimit или MGCOrganatedPendingLimit) из-за ошибки, а получатель исходного TransactionRequest продолжает посылать TransactionPending, для того, чтобы запустить соответствующие действия по восстановлению, получателю исходного TransactionRequest следует применять алгоритм защиты управления. Отправитель исходного TransactionRequest может вести счет полученных Pending и запускать корректирующие действия.

8.3 Сообщения

Множество транзакций могут быть соединены в Сообщение (Message). Сообщения имеют заголовок, который включает в себя идентификационные данные отправителя. Значение Message Identifier (MID) сообщения устанавливается на заданное имя (например, адрес домена/ доменное имя/ название устройства) объекта, передающего сообщение. По умолчанию предлагается доменное имя. Объект H.248.1 (MG/MGC) должен последовательно использовать один и тот же MID во всех сообщениях, которые он отправляет, для продолжения связи управления с устройством – партнером (MGC/MG).

Каждое сообщение содержит номер версии, обозначающий версию протокола, которой соответствует сообщение. Номер версии состоит из одной или двух цифр, начиная с версии 1. Текущий номер версии протокола – версия 3.

Транзакции в сообщении обрабатываются независимо друг от друга. Какого-либо определенного порядка не подразумевается; не существует приложения или подтверждения сообщения протоколом.

Сообщение является по существу алгоритмом/ механизмом передачи. Например, в ответ на сообщение X, содержащее запросы транзакций A, B, и C, может быть получено сообщение Y, содержащее ответы на A и C, и сообщение Z, содержащее ответ на B. Подобным образом, в ответ на сообщение L, содержащее запрос D, и сообщение M, содержащее запрос E, может быть получено сообщение N, содержащее ответы как на D, так и на E.

9 Передача

Алгоритм/ механизм передачи для протокола должен обеспечивать надежную передачу транзакций между MGC и MG. Передача должна оставаться независимой от того, какие конкретно команды отсылаются, и должна быть применима ко всем состояниям приложения. Для данного протокола существует несколько определенных (механизмов) передачи, описываемых в приложениях к данной Рекомендации и других Рекомендациях подсерии H.248 (например, H.248.4 и H.248.5). Дополнительные (механизмы) передачи могут быть определены как дополнительные Рекомендации в Рекомендациях подсерии H.248. Для передачи данного протокола по IP, контроллеры MGC должны применять TCP или UDP/ALF, или и то и другое.

Шлюзу MG предоставляется имя или адрес (такой как имя DNS или IP адрес) основного и нулевого или более второстепенных MGC (см. п. 7.2.8), который MG использует для передачи сообщений MGC. Если в качестве (механизма) передачи протокола используется TCP или UDP, а порт, на который следует отправить первоначальный запрос ServiceChange, иначе не известен, данный запрос должен быть отправлен на заданный по умолчанию для данного протокола номер порта. Для операции в текстовом коде номер порта – 2944, а для операции в двоичном коде – 2945, для UDP или TCP. MGC получает от MG сообщение, содержащее запрос ServiceChange, и может узнать адрес MG из него. Как описано в 7.2.8, MG либо MGC могут предоставить адрес, на который должны быть адресованы последующие запросы транзакций, в параметре ServiceChangeAddress, однако ответы (включая ответ на первоначальный запрос ServiceChange) должны всегда отправляться на адрес, с которого был послан соответствующий запрос. Например, в IP сетях таким адресом является адрес источника в IP заголовке и номер порта источника в заголовке TCP/UDP/SCTP.

9.1 Порядок выполнения команд

Данная Рекомендация не mandate, что базовый транспортный протокол гарантирует упорядочение транзакций, передаваемых объекту. Данное свойство стремится максимально увеличить своевременность действий, однако оно имеет несколько недостатком. Например:

- Может произойти задержка команд Notify, и они придут на MGC после передачи новой команды, изменяющей дескриптор Events.
- Если передача новой команды осуществляется до подтверждения предыдущей, нет гарантии того, что предыдущая команда будет выполнена до новой команды.

Контроллеры медиашлюза, желающие гарантировать согласованную работу медиашлюза, могут использовать следующие правила. Данные правила относятся к командам, находящимся в разных транзакциях. Команды, находящиеся в одной транзакции, выполняются по порядку (см. пункт 8).

- 1) Когда медиашлюз работает с несколькими завершениями, передача команд, относящихся к разным завершениям, может происходить параллельно, например, по модели, в которой каждое завершение (или группа завершений) контролируется своим собственным процессом или своей собственной цепочкой взаимосвязанных программ (thread).
- 2) На завершении обычно должна быть по крайней мере одна команда, ожидающая выполнения (Add или Modify или Move), кроме случаев, когда команды, ожидающие выполнения, находятся в одной и той же транзакции. Однако команда Subtract может быть запущена в любое время. Вследствие этого, иногда медиашлюз может получать команду Modify, относящуюся к ранее вычтенному завершению. Такие команды следует игнорировать, в качестве ответа следует вернуть код ошибки.
- 3) Для алгоритмов передачи, которые не гарантируют последовательную доставку сообщений (т. е. UDP), на данном завершении в любой момент времени обычно должна быть по крайней мере одна команда Notify, ожидающая выполнения.

- 4) В некоторых случаях, явно или неявно, обобщенная команда Subtract, относящаяся к группе завершений, может выступить вперед ожидающей команды Add. Контроллеру медиашлюза следует в индивидуальном порядке удалить все завершения, которых ожидала команда Add во время выполнения глобальной команды Subtract. Таким образом нельзя отсылать новые команды Add для завершений, поименованных с использованием групповых символов замены (или содержащихся в дескрипторе Multiplex), до тех пор, пока не получено подтверждение обобщенной команды Subtract.
- 5) AuditValue и AuditCapability не подлежат упорядочиванию.
- 6) ServiceChange всегда должна быть первой командой, передаваемой MG, как определено процедурой перезапуска. Все другие команды или ответы должны доставляться после данной команды ServiceChange.

Данные правила не влияют на отвечающего на команды, который всегда должен отвечать на команды.

9.2 Защита от перегрузки при перезапуске

В случае одновременного включения большого числа медиашлюзов, все они запускают транзакцию ServiceChange, и весьма вероятно, что контроллер медиашлюза будет перегружен, что приведет к потерям сообщений и перегрузке сети во время критического периода восстановления обслуживания. Во избежание подобных перегрузок предлагается следующий режим работы:

- 1) При включении медиашлюза он должен выставить таймер перезапуска на случайное значение, равномерно распределенное между нулем и значением максимальной задержки ожидания (maximum waiting delay (MWD)). Следует позаботиться о том, чтобы при генерировании случайного числа на нескольких медиашлюзах, использующих один и тот же алгоритм, не возникло синхронности.
- 2) Затем медиашлюзу следует либо дождаться истечения данного таймера, либо обнаружения активности локального пользователя такой, например, как переход в состояние "трубка снята" на частном шлюзе.
- 3) При истечении таймера или при обнаружении активности, медиашлюз должен активировать процедуру перезапуска.

Процедура перезапуска просто требует от MG гарантии того, что первое сообщение, которое приходит контроллеру медиашлюза от MG, представляет собой сообщение ServiceChange, информирующее контроллер медиашлюза о перезапуске.

ПРИМЕЧАНИЕ. – Значение MWD является параметром конфигурации и зависит от типа медиашлюза. Для определения значения данной задержки на частных шлюзах можно использовать дальнейшие объяснения.

Обычно контроллеры медиашлюзов распределяются для обработки информационной нагрузки в пиковые часы, во время которых в среднем загружены будут 10% линий, размещая соединения со средней продолжительностью около 3 минут. Обработка соединения обычно включает от пяти до шести транзакций контроллера медиашлюза между каждым медиашлюзом и контроллером медиашлюза. Простой расчет показывает, что ожидаемое время обработки контроллером медиашлюза пяти-шести транзакций для каждого завершения составит в среднем 30 минут, или, другими словами, в среднем около одной транзакции на завершение каждые пять-шесть минут. Это наводит на мысль, что разумным значением MWD для частного шлюза будет от 10 до 12 минут. При отсутствии явной конфигурации, частным шлюзам следует принять значение MWD, равное 600 секундам.

То же самое рассуждение наталкивает на мысль, что значение MWD для канальных шлюзов или коммерческих шлюзов (business gateways) должно быть гораздо меньше, поскольку они работают с большим количеством завершений, а также скорость использования данных завершений во время часов наибольшей нагрузки гораздо больше 10%, обычно это значение составляет 60%. Таким образом, ожидается, что во время часов наибольшей нагрузки данные завершения будут добавлять одну транзакцию в минуту к нагрузке контроллера медиашлюза. Рациональным будет сделать значение MWD для "канального" завершения в шесть раз короче, чем MWD для частного шлюза, а также обратно пропорциональным числу перезапускаемых завершений. Например, Для шлюза,

обрабатывающего линию T1, следует установить значение MWD на 2,5 секунд, а для шлюза, обрабатывающего линию T3, – 60 миллисекунд.

9.3 Защита от перегрузки Notify

Если возникает ситуация, когда на MG скапливается большое количество уведомлений либо из-за трудностей при передаче, либо из-за того, что MG распознал несколько событий за короткий промежуток времени, шлюзу MG следует ограничить передачу уведомлений, пока очередь сообщений не очистится.

10 Вопросы безопасности

В данном пункте рассматриваются вопросы безопасности при использовании данного протокола в IP окружении.

10.1 Защита соединений протокола

Для предотвращения использования протокола, определенного в данной Рекомендации, несанкционированными объектами в целях установления несанкционированных соединений или вмешательства в санкционированные соединения необходим алгоритм/ механизм безопасности. Алгоритмом безопасности для данного протокола при передаче по IP сетям является IPSec (RFC 2401 – RFC 2411).

Заголовок AH (RFC 2402) предоставляет информацию об аутентификации происхождения данных, целостности без установления соединения и дополнительно защиту от повторного воспроизведения (anti-replay protection) сообщений, которыми обмениваются MG и MGC.

Заголовок ESP (RFC 2406) при необходимости обеспечивает конфиденциальность сообщений. Например, следует запросить службу шифрования ESP, если для передачи ключей сессии используются описания сессии, как определено в SDP.

Реализации протокола, определенного в данной Рекомендации, использующие заголовок ESP должны удовлетворять требованиям раздела 5 RFC 2406, в котором определяется минимальный набор алгоритмов для проверки целостности и шифрования. Аналогичным образом, реализации, использующие заголовок AH должны удовлетворять требованиям раздела 5 RFC 2402, в котором определяется минимальный набор алгоритмов для проверки целостности с использованием неавтоматических ключей.

Реализации должны использовать IKE (RFC 2409) для улучшения опций манипуляции (keying options). Реализации, использующие IKE должны поддерживать аутентификацию с подписями RSA и шифрованием открытым ключом RSA.

10.2 Промежуточная схема AH

При реализации IPSec требуется, чтобы сразу после IP заголовка вставлялись заголовки AH или ESP. Выполнить данное условие на прикладном уровне нелегко. Поэтому для протокола, определенного в данной Рекомендации, встает проблема развертывания, т.к. основная реализация сети не поддерживает IPSec.

В качестве временного решения дополнительный заголовок AH определяется в заголовке протокола H.248.1. Поля заголовка являются как раз теми полями SPI, SEQUENCE NUMBER и DATA, определенными в RFC 2402. Значение полей заголовка является таким же, как и в "режиме передачи" RFC 2402, за исключением вычисления значения проверки целостности (Integrity Check Value (ICV)). В IPSec, ICV вычисляется по всему IP пакету, включая IP заголовок. Это предотвращает получения IP адресов обманном путем (spoofing of the IP addresses). Для сохранения того же набора функциональных возможностей вычисление ICV следует проводить по всем транзакциям (объединенным) в сообщении, которому предшествует синтезированный IP заголовок, состоящий из 32-битного IP адреса источника, 32-битного адреса места назначения и 16-битного UDP порта назначения, закодированного в виде 20 шестнадцатеричных символов. Если промежуточный алгоритм AH используется, когда TCP находится на транспортном уровне Layer, вышеупомянутый UDP Порт становится TCP портом, и все остальные операции проводятся аналогично.

Реализации протокола H.248.1 должны применять IPSec там, где базовая оперативная система и транспортная сеть поддерживают IPSec. Реализации данного протокола, использующие IPv4, должны применять промежуточную схему АН. Однако данную временную схему нельзя использовать, когда основной уровень сети поддерживает IPSec. Предполагается, что реализации IPv6 поддерживают IPSec и не должны использовать промежуточную схему АН.

Все реализации промежуточного алгоритма АН должны удовлетворять требованиям раздела 5 RFC 2402, где определяется минимальный набор алгоритмов для проверки целостности с использованием неавтоматизированных ключей.

Промежуточная схема АН не защищает от перехвата/ подслушивания (eavesdropping), таким образом, запрещая третьим сторонам контролировать соединения, установленные данным завершением. Также она не защищает от атак воспроизведения (replay attacks). Данные процедуры не обязательно защищают от атак типа "отказ в обслуживании" (denial of service) со стороны враждебных MG или MGC. Однако они обеспечат идентификацию данных враждебных объектов, которые затем должны быть лишены авторизации посредством технологических процедур.

10.3 Защита медиасоединений

Данный протокол разрешает MGC обеспечивать шлюзы MG "ключами сессии", которые можно использовать при шифровании аудиосообщений для защиты от перехвата.

Характерной проблемой сетей с коммутацией пакетов является "неконтролируемое вмешательство". Данный вид атаки можно осуществить путем направления пакетов медиа на IP адрес и UDP порт, используемые в соединении. Если защиты нет, данные пакеты будут распакованы и воспроизведены на разъемах шин на стороне линии связи ("line side").

Основной вид защиты против таких атак состоит в приеме пакетов только из известных источников, и осуществления проверки того, например, что IP адрес источника и UDP порт источника совпадают со значениями, обозначенными в дескрипторе Remote. У данного метода существует два недостатка: замедляется установление соединения и можно ввести в заблуждение путем перехвата источника (source spoofing):

- Для обеспечения защиты на основе адресов MGC должен получить удаленное описание сеанса исходящего MG и передать его входящему MG. Для этого потребуется как минимум один двойной проход сети, что ставит перед нами дилемму: разрешить дальнейшее соединение, не ожидая завершения двойного прохода и рисковать, например, "ограничивая" удаленную сигнализацию, или ждать полного двойного прохода и согласиться на более медленное выполнение процедур установления вызова.
- Перехват источника эффективен, только если злоумышленник может получить верные пары адресов и портов источника и места назначения, например путем прослушивания части трафика. Для борьбы с перехватом источника можно попробовать контролировать все точки доступа в сеть. Однако на практике это очень трудно реализовать.

Альтернативой проверке адреса источника является шифрование и аутентификация пакетов с использованием секретного ключа, сообщаемого во время процедуры установления соединения. Это не вызовет замедления установления соединения, а также обеспечивает надежную защиту от перехвата адресов.

11 Интерфейс управления MG-MGC

Соединение управления/ контроля между MG и MGC запускается при начальном запуске/ холодном пуске MG, и объявляется при помощи сообщения ServiceChange, однако его могут изменить последующие события такие, как сбой в работе или события обслуживания вручную.

ПРИМЕЧАНИЕ. – Поскольку у данного протокола нет явного алгоритма для поддержки множества MGC, контролирующих физический MG, он был разработан для поддержки множества логических MG (внутри одного физического MG), который может быть связан с различными MGC.

11.1 Множество Виртуальных MG

Физический медиашлюз может быть разделен на один или более Виртуальных MG (VMG). Виртуальный MG состоит из набора статистически общих физических завершений и/ или наборов

непродолжительных завершений. Управление физическим завершением осуществляется один MGC. Для данной модели не требуется, чтобы были статистически распределены другие ресурсы – только завершения. Алгоритм распределения завершений по Виртуальным MG является методом управления, не входящим в сферу применения данной Рекомендации. Для MGC каждый Виртуальный MG является законченным/ полным клиентом MG.

У физического MG может быть только один сетевой интерфейс, который может быть разделен между Виртуальными MG. В таком случае происходит разделение завершения со стороны пакета ячейки. Следует заметить, однако, что при использовании такие интерфейсы требуют создания непродолжительной ступени завершения для каждого потока, разделение завершения, таким образом, происходит напрямую. Данный алгоритм действительно приводит к усложнению, а именно к тому, что MG всегда должен знать, какого из управляющих им контроллеров MGC следует уведомить, если на интерфейсе произойдет какое-либо событие.

При нормальной работе MGC отдаст распоряжение Виртуальному MG создать сетевые потоки (если он является исходящей стороной), или ожидать запросов потоков (если он является завершающей стороной), и замешательства не произойдет. Однако, если произойдет непредвиденное событие, Виртуальный MG должен знать, что делать с физическими ресурсами, которые он контролирует.

Если для восстановления после события требуется изменение состояния физического интерфейса, изменить свое состояние должен только один MGC. Данные вопросы разрешаются путем разрешения любому из контроллеров MGC создать дескрипторы Events для уведомления о таких событиях, однако доступ чтение/ запись к свойствам физического интерфейса может иметь только один MGC; все остальные MGC имеют доступ только – чтение. Данный алгоритм управления используется для определения того, какой из MGC имеет возможность чтения/ записи, и назначается Главным (Master) MGC.

У каждого Виртуального MG есть свое собственное Завершение Root. В большинстве случаев, значения для свойств Завершения Root Termination устанавливаются каждым MGC независимо друг от друга. Там, где возможно лишь одно значение, параметром является только – чтение для всех MGC, кроме Главного.

Команда ServiceChange может быть применена к завершению или группе завершений, общих для Виртуальных MG или созданных (в случае непродолжительных завершений) данным Виртуальным MG.

11.2 Начальный запуск

Алгоритмом управления, находящимся вне сферы применения данной Рекомендации, шлюзу MG предварительно предоставлены основной MGC и (дополнительно) упорядоченный список второстепенных MGC. После начального запуска MG, он пошлет команду ServiceChange с методом "Restart" на завершении Root Termination основному MGC. Если MGC принимает данный MG, он посылает TransactionReply, не включая в него параметр ServiceChangeMgcID. Если MGC не принимает регистрацию MG, он посылает TransactionReply, предоставляя адрес дополнительного MGC, с которым следует связаться, путем включения параметра ServiceChangeMgcID.

Если MG получает TransactionReply, включающий в себя параметр ServiceChangeMgcID, он посылает команду ServiceChange контроллеру MGC, определенному в ServiceChangeMgcID. Шлюз продолжает действовать таким образом до тех пор, пока контролирующей/ управляющей MGC не примет его регистрацию, или пока он не потерпит неудачу в получении ответа. После неудавшейся попытки получить ответ либо от основного MGC, либо от заданного последующего, MG осуществляет попытки с предварительно предоставленными дополнительными MGC по порядку. Если MG не удастся установить отношение управления ни с одним MGC, шлюз должен выждать случайное количество времени, как описано в п. 9.2, а затем снова начать устанавливать контакт со своим основным и, при необходимости, второстепенными MGC.

Возможно, что ответ на ServiceChange с Restart будет потерян, и команда будет получена MG до получения ответа ServiceChange. MG должен отослать Код Ошибки 505 ("TransactionRequest получен до получения ServiceChange Reply").

11.3 Согласование версии протокола

Команда ServiceChange от MG, регистрирующегося на MGC, должна содержать в параметре ServiceChangeVersion номер версии данного протокола, поддерживаемого данным MG. Вне

зависимости от номера версии, указываемого в параметре ServiceChangeVersion, сообщение, содержащее команду, должно быть закодировано как сообщение версии 1. После получения такого сообщения, если MGC поддерживает только более низкую версию данного протокола, MGC должен послать ServiceChange Reply с более низкой версией, и после этого все сообщения, которыми обмениваются MG и MGC, должны соответствовать более низкой версии протокола. Если MG не может соответствовать, но уже установил транспортное соединение с MGC, ему следует закрыть это соединение. Так или иначе, шлюзу следует отклонять все последующие запросы от MGC при помощи Кода Ошибки 406 ("Версия не поддерживается").

Если MGC поддерживает только более высокую(ие) версию(и), чем MG, контроллер должен отказать в соединении при помощи Кода Ошибки 406 ("Версия не поддерживается").

Если MGC поддерживает версию, обозначенную MG, он должен придерживаться данной версии во всех последующих сообщениях. В таком случае возврат версии в ServiceChange Reply является для MGC необязательным.

Согласование версии протокола может также происходить на "Handoff" и "Failover" ServiceChanges.

При добавлении новых версий к данному протоколу следует соблюдать следующие правила:

- 1) Не следует изменять существующие элементы протокола, т. е. процедуры, параметры, дескриптор, свойство, значения, кроме случаев, когда необходимо исправить ошибку в протоколе, или возникает необходимость в изменении функционирования службы, поддерживаемой данным протоколом.
- 2) Не следует изменять семантику команды, параметра, дескриптора, свойства или значения.
- 3) Не следует изменять установленные требования для форматирования и кодирования сообщений и параметров.
- 4) Когда обнаруживается, что элементы информации устарели, они могут быть помечены как неиспользуемые. Однако идентификатор для данного элемента информации должен быть помечен как зарезервированный. В таком случае его нельзя будет использовать в следующих версиях.

11.4 Выход MG из строя

Если в работе MG происходит сбой, но он способен послать сообщение MGC, шлюз посылает ServiceChange подходящим способом (Graceful или Forced) и указывает Root TerminationID. Когда шлюз возвращается к работе, он посылает ServiceChange при помощи "Restart" ServiceChangeMethod.

Разрешение MGC отсылать двойные сообщения обоим MG позволяет создавать пары MG, способных к дублирующему замещению при отказе одного из MG. Принимать или отклонять транзакции должен только Работающий (Working) MG. После преодоления отказа основной MG посылает команду ServiceChange методом "Failover", указывая в качестве причины "MG Impending Failure" (приближающийся сбой MG). Тогда MGC использует в качестве активного MG дополнительный MG. Когда условие возникновения ошибки устранено, работающий MG может послать "ServiceChange" методом "Restart" ServiceChangeMethod.

ПРИМЕЧАНИЕ. – Для MG, дублирующих при отказе, требуется надежная передача, так как протокол не предоставляет дополнительному MG, запускающего ALF, средств для подтверждения получения сообщений, отправленных MGC.

11.5 Выход MGC из строя

Если MG обнаруживает сбой на управляющем им MGC, он пытается связаться со следующим MGC в своем предварительно предоставленном списке. Он начинает осуществлять попытки с начала списка (основной MGC), если только сбой произошел не на основном MGC. В таком случае шлюз начинает с первого дополнительного MGC. Он посылает сообщение ServiceChange методом "Failover", указывая в качестве причины "MGC Impending Failure". Если MG не удается установить отношения управления ни с одним MGC, он должен выждать случайное количество времени, как описано в п. 9.2, а затем снова начать связываться со своим основным и (при необходимости) с дополнительными MGC. При соединении с ранее управлявшим MGC, MG посылает сообщение ServiceChange методом "Disconnected".

При частичном сбое или по соображениям эксплуатации вручную MGC может пожелать направить управляемые им MG на другой MGC. Для того, чтобы сделать это, он посылает MG ServiceChange с методом "Handoff", и назначенной им замену в ServiceChangeMgcID. Если "Handoff" поддерживается, MG должен послать назначенному MGC сообщение ServiceChange методом "Handoff", указав в качестве причины "MGC directed change". Если шлюзу не удастся получить ответ от назначенного MGC, MG должен действовать так, как будто его MGC дал сбой, и начать связываться в дополнительными MGC, как описывается в предыдущем абзаце. Если MG не удастся установить отношения управления ни с одним из MGC, он должен выждать случайное количество времени, как описано в п. 9.2, а затем начать снова связываться со своим основным и, при необходимости, дополнительными MGC.

Рекомендации о том, как контроллеры MGC, участвующие в переадресации вызова, сохраняют информацию о состоянии оборудования; считается, что это не входит в сферу применения данной Рекомендации. При возникновении переадресации вызова MGC и MG могут предпринять следующие шаги. Когда MGC инициирует переадресацию вызова, данная переадресация должна быть прозрачной для процессов на медиашлюзе. Выполнение транзакций может осуществляться в любом порядке и может исполняться при выполнении ServiceChange. Соответственно, выполнение текущих команд продолжается, и ответы на все команды от исходного MGC должны быть отправлены на тот транспортный адрес, с которого они были посланы. Если обслуживание отправляющего MGC закончилось, следует удалить ответы. MG может получить невыполненные ответы транзакций от нового MGC. Нельзя посылать новые сообщения новому MGC, пока не будет установлена связь управления. Повторяющиеся запросы транзакций должны быть направлены новому MGC. MG должен сохранять режим работы всех завершений и контекстов.

Возможно применить MGC таким образом, что неисправный MGC заменяется рабочим MGC в то время как идентификационная информация нового MGC такая же, как и у неисправного. В таком случае ServiceChangeMgcID будет определен с предыдущим значением, а MG должен действовать так, как будто значение было изменено, и отправить сообщение ServiceChange, как описано выше.

Пары MGC, способные к дублирующему замещению при отказе могут уведомлять управляемые ими MG о сбое в работе, следуя вышеописанному алгоритму.

11.6 Контроль связи управления MGC-MG

Контроль связи управления MGC-MG необходим для сетей с высокой готовностью. Контроль может осуществляться путем непрерывной проверки соединения MGC-MG (состояние канала). Многие виды передачи предоставляют данную функциональную возможность, поэтому обязательный для выполнения алгоритм протокола необязателен.

Для видов передачи, не обеспечивающих контроль состояния канала, выполнить данную функциональную возможность можно в протоколе H.248 путем использования существующего обмена сообщениями. При отсутствии какого-либо обмена сообщениями H.248 для обнаружения потери соединения с MG, MGC может использовать команду AuditValue на Root с пустым дескриптором Audit. MG может обнаружить потерю соединения самостоятельно путем использования Inactivity Timer Package (Рек. МСЭ-Т Н.248.14).

12 Определение комплекта

Основным алгоритмом для расширения являются средства Комплектов. В комплектах определяются дополнительные свойства, которые могут возникнуть на завершениях, и контексты, события, сигналы и статистика, которые могут возникнуть на завершениях.

Комплекты, определенные IETF, появятся в отдельных RFC.

Комплекты, определенные МСЭ-Т, могут появиться в соответствующих Рекомендациях (например, как в подсерии Рекомендаций H.248).

- 1) Должен быть определен официальный документ или стандартный документ форума, на который можно осуществлять ссылку, как на документ, описывающий комплект, следуя вышеописанным указаниям.
- 2) В данном документе должна определяться описываемая версия Комплекта.

- 3) Данный документ должен быть доступен на официальном сайте и должен иметь неизменный URL. На сайте должен обеспечиваться механизм для предоставления комментариев, и должны даваться соответствующие ответы.

12.1 Указания по определению комплектов

В комплектах определяются свойства, события, сигналы и статистику.

В комплектах могут также определяться новые коды ошибок в соответствии с рекомендациями, данными в 14.2. В целях удобства использования документации документация комплектов при поддержке регистрации кодов ошибок подчиняется организации IANA. Если происходит изменение комплекта, предоставлять организации IANA новую ссылку на документ при поддержке кода ошибки не нужно, кроме случаев, когда изменяется само описание кода ошибки.

Названия всех подобных структурных компонентов должны состоять из PackageID (который уникально идентифицирует комплект) и ID элемента (который уникально идентифицирует элемент в данном комплекте). При текстовом кодировании эти два должны быть отделены друг от друга символом косой черты ("/"). Например: tonegen/playtone является текстовым кодированием, обозначающим тональный сигнал воспроизведения в комплекте генерирования звука.

Комплект должен содержать следующие разделы и ключевые слова (обозначенные жирным шрифтом). Шаблон для определения комплектов содержится в Приложении II.

12.1.1 Package

Общее описание комплекта, с указанием:

Название комплекта: только описательное

PackageID: является идентификатором

Описание: является описанием комплекта

Версия:

В новой версии комплекта возможно только добавление дополнительных свойств, событий, сигналов, статистики и новых возможных значений для существующего параметра, описанного в исходном комплекте. Какие-либо удаления или изменения не допускаются. Номер версии представляет собой целое число от 1 до 99.

Разработан только для расширения (Дополнительно): Да

Данный пункт указывает на то, что данный комплект был специально разработан для того, чтобы его расширяли другие, а не ссылались напрямую. Например, данный комплект может не иметь никаких собственных функций или может сам по себе быть бессмысленным. При предоставлении отчета по комплектам MG не должен оглашать/публиковать данный PackageID.

Расширения/ добавления (Extends): существующий идентификатор комплекта и версия

Комплект может расширять существующий комплект. Должна быть указана версия исходного комплекта. При расширении комплекта другим комплектом должно происходить только добавление дополнительных свойств, событий, сигналов, статистики и новых возможных значений для существующего параметра, описанного в исходном комплекте. Расширенный комплект не должен переопределять или перегружать идентификатор, определенный в исходном комплекте и комплектах, которые он мог расширить ранее (несколько уровней расширения). Поэтому, если комплект В версия 1 расширяет комплект А версия 1, версия 2 комплекта В не сможет расширить комплект А версия 2, если комплект А версия 2 определяет имя уже в комплекте В версия 1. Если комплект не расширяет другой комплект, должно указываться – "нет" ("none")

12.1.2 Свойства

Свойства, определяемые комплектом, с указанием:

Название свойства: только описательное

PropertyID: идентификатором.

Описание: является описанием функции свойства

Тип: Один из:

Логическое выражение

Строка: строка UTF-8

Байтовая строка: Несколько байтов. См. Приложения А и В.3 для кодирования

Целое число: 4-байтовое целое число со знаком

Двойное число: 8-байтовое целое число со знаком

Буква: Уникод UTF-8 кодирование одной буквы. Может быть больше, чем один байт.

Перечисление: одно из списка возможных уникальных значений.

Подсписок: список из нескольких значений списка. Необходимо указать тип подсписка. Тип следует выбрать из типов, описанных в данном разделе (за исключением подсписка). Например, Тип: подсписок перечисления. Кодирование подсписков рассматривается в Приложениях А и В.2.

Возможные значения:

В комплекте должен быть указан либо определенный набор значений, либо описание того, как определяются значения. В комплекте также должно указываться заданное по умолчанию значение или заданный по умолчанию режим работы, когда значение отсутствует в своем дескрипторе. Например, в комплекте может указываться, что, когда его значение пропускается, процедуры, относящиеся к данному свойству, временно приостанавливаются.

Задано по умолчанию:

Значение, заданное по умолчанию (но не процедуры), может быть определено как предоставляемое.

Определено в:

Дескриптор Н.248.1, в котором определено свойство. LocalControl – для свойств, зависящих от потока. TerminationState – для свойств, не зависящих от потока. ContextAttribute – для свойств, влияющих на контекст в целом, т.е. свойства перемешивания (mixing properties). Данные дескрипторы являются наиболее общими случаями, но возможно определение свойств в других дескрипторах. Свойства контекста должны определяться в дескрипторе ContextAttribute.

Характеристики: Только чтение или Чтение/ Запись, и (дополнительно), глобальный:

Показывает, предназначено ли свойство только для чтения или для чтения/ записи, а также указывается, если оно является глобальным. Если характеристика "глобальный" пропускается, свойство не является глобальным. Если свойство объявляется глобальным, значение данного свойства используется всеми завершениями, реализующими данный комплект, совместно. Если глобальным объявляется свойство контекста, данное свойство используется совместно всеми контекстами, реализующими данный комплект.

12.1.3 События

События, определяемые комплектом, с указанием:

Название события: только описательное

EventID: является идентификатором

Описание: описание функции события

Параметры EventsDescriptor:

Параметры, используемые MGC для конфигурирования события, находятся в дескрипторе Events. См. 12.2. Если параметров для дескриптора Events нет, следует указать "нет" ("none").

Параметры ObservedEventsDescriptor:

Параметры возвращаются MGC в запросах Notify и в ответах на запросы команд от MGC, проверяющего дескриптор ObservedEvents, параметры находятся в дескрипторе ObservedEvents. См. 12.2. Если параметров для дескриптора ObservedEvents нет, следует указать "нет" ("none").

12.1.4 Сигналы

Сигналы, определяемые комплектом, с указанием:

Название сигнала: только описательное

SignalID: является идентификатором. SignalID используется в дескрипторе Signals

Описание: описание функции сигнала

SignalType: один из:

OO (OnOff)

TO (TimeOut)

BR (Brief)

ПРИМЕЧАНИЕ. – SignalType может быть определен так, что он будет зависимым от значения одного или более параметров. В комплекте должен указываться заданный по умолчанию тип сигнала. Если по умолчанию задан тип сигнала TO, в комплекте должна указываться заданная по умолчанию продолжительность, значение которой может быть предоставлено. Заданная по умолчанию продолжительность для BR не имеет смысла.

Продолжительность: в сотых долях секунды

Дополнительные параметры: см. п. 12.2

12.1.5 Статистика

Статистика, определяемая комплектом, с указанием:

Название статистики: только описательное

StatisticID: является идентификатором

StatisticID используется в дескрипторе Statistics

Описание: описание статистики

Тип: один из:

Логическое выражение

Строка: строка UTF-8

Байтовая строка: несколько байтов. См. Приложения А и В.3 для кодирования

Целое число: 4- байтовое целое число со знаком

Двойное число: 8-байтовое целое число со знаком

Буква: Уникод UTF-8 кодирование одной буквы. Может быть больше, чем один байт.

Перечисление: Одно из списка возможных уникальных значений.

Подсписок: Список из нескольких значений списка. Необходимо также указать тип подсписка. Тип следует выбрать из типов, определяемых в данном разделе (за исключением подсписка). Например, Тип: подсписок перечисления. Кодирование подсписков рассматривается в Приложениях А и В.2.

Возможные значения:

В комплекте должна указываться единица измерения, например, миллисекунды, пакеты либо здесь, либо вместе с типом, указанным выше, а также индикация любых ограничений диапазона.

Уровень: Указывается, можно ли вести статистику на уровне завершения, потока или на обоих уровнях.

12.1.6 Коды ошибок

Если в комплекте коды ошибок не определяются, этот раздел можно пропустить. В противном случае в данной разделе описываются коды ошибок, определяемые данным комплектом, с указанием:

Код ошибки #: Номер кода ошибки.

Название: Название ошибки

Определение: описание кода ошибки.

Текст ошибки в дескрипторе Error:

Описание того, какой текст следует возвращать в дескрипторе Error.

Комментарии: Любые дальнейшие комментарии по использованию кода ошибки.

12.1.7 Процедуры

Дальнейшие рекомендации по использованию данного комплекта.

12.2 Указания по определению параметров событий и сигналов

Название параметра: только описательный

ParameterID: является идентификатором. Текстовый ParameterID параметров событий и сигналов не должен начинаться с "EPA" и "SPA", соответственно. Также текстовый ParameterID не должен быть "ST", "Stream", "SY", "SignalType", "DR", "Duration", "NC", "NotifyCompletion", "KA", "KeepActive", "EB", "Embed", "DM", "DigitMap", "SPADI", "SPADirection", "SPARQ" или "SPARequestID".

Описание: описание функции параметра.

Тип: один из:

Логическое выражение

Строка: байтовая строка UTF-8

Байтовая строка: Несколько байтов. См. Приложения А и В.3 для кодирования

Целое число: 4-байтовое целое число со знаком

Двойное число: 8- байтовое целое число со знаком

Буква: Уникод UTF-8 кодирование одной буквы. Может быть больше, чем один байт.

Перечисление: одно из списка возможных уникальных значений.

Подпись: список нескольких значений из списка (не поддерживается для статистики). Также должен быть указан тип подписки. Тип следует выбирать из типов, определенных в данном разделе (за исключением подписки). Например, Тип: подпись перечисления. Кодирование подписок приводится в Приложениях А и В.2.

Дополнительно: Да/ Нет (Yes/No)

Описывает, можно ли пропустить данный параметр в сигнале или событии.

Возможные значения:

В комплекте должен определяться либо определенный набор значений или описание того, как определяются данные значения. В комплекте также должно указываться заданное по умолчанию значение или заданный по умолчанию режим работы, когда значение в дескрипторе пропущено. Например, в комплекте может указываться, что процедуры, относящиеся к данному параметру, приостанавливаются, когда пропускается его значение.

Задано по умолчанию:

Значение (не не процедуры), заданное по умолчанию, может быть определено как предоставляемое.

12.3 Идентификаторы

Идентификаторы в текстовом кодировании должны представлять собой строки до 64 символов, без пробелов, начинающиеся с буквенного символа и состоящие из буквенно-цифровых символов и/ или цифр, и, возможно, включающие в себя специальный символ подчеркивания ("_").

Длина идентификаторов в двоичном кодировании – 2 байта.

Для каждого идентификатора, включая идентификаторы, использующиеся как значения в типах перечисления, должно быть указаны как текстовые, так и двоичные значения.

12.4 Регистрация комплекта

Комплект может быть зарегистрирован в организации IANA для целей функциональной совместимости. См. пункт 14 для вопросов IANA.

13 Определение профиля

Для дальнейшего определения того, как используется протокол H.248.1, и какие функциональные характеристики поддерживаются MG, могут определяться профили. Профиль описывает только возможности интерфейса MGC/MG H.248. Сам профиль определяет, какие опции, связанные с H.248.1, были использованы. Например, передача (transport) и комплекты, использованные для приложения.

Профили маркируются при помощи названия (зарегистрированного в IANA) и версии. Название должно представлять собой строку, независимую от регистра, длиной до 64 символов. Номер версии должен быть от 1 до 99.

Сам профиль представляет собой документ, показывающий опции для определенного приложения. Установленного формата для данного документа не существует. Единственным обязательным элементом является раздел, в котором указывается название профиля (Profile Name) и версия (Version) и резюме профиля.

Следующие два пункта являются единственными обязательными разделами профиля:

- Идентификация профиля (Profile Identification): Название и версия профиля, посылаемого в команде ServiceChange.
- Резюме (Summary): Описание того, что представляет собой данный профиль.

В Дополнении III содержится шаблон для определения профилей. Его следует использовать в качестве основы определения профиля.

14 Вопросы взаимодействия с IANA

14.1 Комплекты

Для регистрации комплекта в IANA необходимо удовлетворить следующие требования:

- 1) Для каждого комплекта регистрируется уникальное название строки, уникальный серийный номер и номер версии. Название строки используется с текстовым кодированием. Серийный номер необходимо использовать с двоичным кодированием. Серийные номера от 0x8000 до 0xFFFF резервируются для частного использования. Серийный номер 0 зарезервирован.
- 2) Для данного контакта должны быть указаны контактное имя, адрес электронной почты и почтовый адрес. Контактную информацию необходимо обновлять путем должного определения организации.
- 3) Ссылка на документ, описывающий комплект, который должен являться открытым:
В документе должна указываться версия комплекта, которую он описывает.
Если документ является открытым, он должен быть размещен на официальном сервере в сети Интернет и должен иметь неизменный URL. На сайте должен присутствовать механизм для предоставления комментариев и возвращения соответствующих ответов.

- 4) Минимальная длина комплектов, зарегистрированных не в общепризнанных организациях стандартизации, должна составлять 8 символов.
- 5) Все названия комплектов обслуживаются в порядке поступления, если все остальные условия выполнены.

14.2 Коды ошибок

Для регистрации кода ошибки в IANA следует удовлетворить следующие условия:

- 1) Для каждой ошибки регистрируются номер ошибки и одна строка (максимальная длина 80 символов).
- 2) Полное описание условий, при которых происходит обнаружение ошибки, должно быть включено в открытый для доступа документ. Описание должно быть достаточно понятным для дифференциации данной ошибки от всех остальных существующих кодов ошибок.
- 3) Документ должен быть доступен на официальном сайте в Интернете, и должен иметь неизменный URL.
- 4) Номера ошибок, регистрируемых в общепризнанных организациях стандартизации, должны состоять из 3-4 символов.
- 5) Номера ошибок, регистрируемых всеми остальными организациями или частными лицами должны состоять из 4 символов.
- 6) Номер ошибки не может быть ни переопределен, ни изменен, кроме как организацией или частным лицом, первоначально определившим его, или их последователями или правопреемниками.

14.3 Причины ServiceChange

Для регистрации причины ServiceChange в IANA необходимо выполнить следующие условия:

- 1) Для каждой причины регистрируется уникальный код причины, представляющий собой одну фразу максимальной длиной 80 символов.
- 2) Полное описание условий, при которых используется причина, должно быть включено в документ, находящийся в открытом доступе. Описание должно быть достаточно ясным для различения данной причины и всех остальных существующих причин.
- 3) Документ должен быть доступен на официальном сайте в Интернете, и должен иметь неизменный URL.

14.4 Профили

Для регистрации профиля в IANA необходимо выполнить следующие требования:

- 1) Для каждого профиля регистрируется уникальное название строки и номер версии (номер версии можно пропустить, если название профиля содержит групповой символ замены).
- 2) Необходимо указать контактное имя, адрес электронной почты и почтовый адрес для данного контакта. Необходимо обновлять контактную информацию путем определения организации необходимым образом.
- 3) Минимальная длина названия профилей, регистрируемых не в общепризнанных организациях стандартизации, должна составлять 6 символов.
- 4) Названия профилей, содержащие групповой символ "*" в конце названий должны приниматься, если первые 6 символов полностью заданы. Предполагается, что организация, выпустившая название профиля, будет распоряжаться пространством имен, связанным с групповым символом.. IANA не должна выпускать другие названия профилей в диапазоне "название*".

Все остальные названия профилей обрабатываются в порядке поступления, если выполняются все остальные условия.

Приложение А

Двоичное кодирование протокола

В данном приложении описывается синтаксис сообщений, использующих нотацию, определенную в Рек. МСЭ-Т X.680, *Информационные технологии – Абстрактно-синтаксическая нотация один (ASN.1): спецификация основной нотации*. Сообщения должны быть закодированы для передачи путем применения основных правил кодирования, определенных в Рек. МСЭ-Т X.690, *Информационные технологии – ASN.1 Правила кодирования: Спецификация основных правил кодирования (Basic Encoding Rules (BER)), канонических правил кодирования (Canonical Encoding Rules (CER)) и известных правил кодирования (Distinguished Encoding Rules (DER))*.

А.1 Кодирование групповых символов замены

В протоколе разрешается использование групповых символов замены ALL и CHOOSE. Это позволяет MGC частично определять идентификаторы TerminationID и дает возможность MG выбирать из значений, соответствующих частичной спецификации. При помощи идентификаторов TerminationID может происходить кодирование иерархии названий. Эта иерархия предоставляется. Например, TerminationID может состоять из магистральной группы, магистральной линии связи внутри группы и линии связи. Обобщение должно быть возможно на всех уровнях. В следующих абзацах объясняется, как можно этого достичь.

Описание ASN.1 использует для идентификаторов TerminationID байтовые строки длиной до 8 байтов. Это означает, что идентификаторы TerminationID состоят не более чем из 64 битов. Полностью определенному TerminationID может предшествовать последовательность полей обобщения. Длина поля обобщения составляет один байт. В бите 7 (наиболее важный бит) данного байта определяется, какой тип обобщения используется: Если значение бита равняется 1, используется групповой символ замены ALL; если значение бита равно 0, используется групповой символ CHOOSE. В бите 6 поля обобщения определяется, относится ли обобщение к одному уровню в иерархической системе наименования (значение бита 0), или к уровню иерархии, определенному в поле обобщения плюс все низлежащие уровни (значение бита 1). В битах 0-5 поля обобщения определяется позиция бита в TerminationID, с которой начинается обобщение.

Проиллюстрируем данную схему несколькими примерами. В данных примерах наиболее важный бит в строке битов появляется на левой стороне.

Предположим, что длина идентификаторов TerminationID составляет 3 байта, и каждый байт представляет уровень в иерархической системе наименования. Действительным TerminationID является:

00000001 00011110 01010101.

Адресация названий ALL при помощи префикса 00000001 00011110 выполняется следующим образом:

Поле обобщения: 10000111

TerminationID: 00000001 00011110 xxxxxxxx.

Значения битов, помеченных "x", не важны, и получатель может их проигнорировать.

Указание получателю о том, что он должен выбрать название с 00011110 в качестве второго байта, выполняется следующим образом:

поля обобщения: 00010111, а затем 00000111

TerminationID: xxxxxxxx 00011110 xxxxxxxx.

Первое поле обобщения обозначает групповой символ замены CHOOSE для уровня в иерархии наименования, начинающегося с бита 23, высшего уровня в нашей вымышленной схеме наименования. Второе поле обобщения обозначает групповой символ замены CHOOSE для уровня в иерархии наименования, начинающегося с бита 7, низшего уровня в нашей вымышленной схеме наименования.

В результате, название с обобщением CHOOSE с высшим уровнем имени, равным 00000001, определяется следующим образом:

поле обобщения: 01001111

TerminationID: 0000001 xxxxxxxx xxxxxxxx.

Значение бита 1 в позиции бита 6 первого байта обозначает, что обобщение относится к определенному уровню в иерархии наименования и всем низлежащим уровням.

Обобщаться могут также и идентификаторы ContextID. Однако в случае с идентификаторами ContextID не разрешается определение частичных названий. Для обозначения нулевого контекста нужно использовать ContextID 0x0, для обозначения группового символа замены CHOOSE нужно использовать ContextID 0xFFFFFFFFE, и для обозначения группового символа замены ALL нужно использовать ContextID 0xFFFFFFFF.

Для обозначения завершения Root нужно использовать TerminationID 0xFFFFFFFFFFFFFFFF.

A.2 Спецификация синтаксиса ASN.1

В данном пункте содержится спецификация ASN.1 синтаксиса протокола H.248.1.

ПРИМЕЧАНИЕ 1. – В случае, если используется механизм передачи, применяющий кадрирование на прикладном уровне, определение **транзакции (Transaction)**, употребляемое ниже, меняется. Определение, применяемое в данном случае, см. в приложении или Рекомендации подсерии H.248.x, определяющих механизм передачи.

ПРИМЕЧАНИЕ 2. – В данной спецификации синтаксиса не вводятся в действие все ограничения по включению элементов и значений. Некоторые дополнительные ограничения приводятся в комментариях, другие ограничения находятся в тексте данной Рекомендации. Данные дополнительные ограничения являются частью протокола, даже если они не вводятся в действие данной Рекомендацией.

ПРИМЕЧАНИЕ 3. – В модуле ASN.1 в данном приложении типы байтовых строк используются для кодирования значений для значений параметра свойства, параметра сигнала и параметра события и статистики. Действительные типы данных значений отличаются, они определены в Приложении С или определении соответствующего комплекта.

Сначала значение кодируется при помощи BER по типу, руководствуясь таблицей внизу. Затем результат кодирования BER кодируется как байтовая строка ASN.1, делая "двойное упаковывание" данного значения. Формат, определенный в Приложении С, или комплект соотносится с кодированием BER в соответствии со следующей таблицей:

Тип, определенный в комплекте	Тип ASN.1 BER
Строка	IA5String or UTF8String (Примечание 4)
Целое число (4-байт)	INTEGER
Двойное число (8-байтовое целое число со знаком)	INTEGER (Примечание 3)
Символ (UTF-8) (Примечание 1)	IA5String
Перечисление	ENUMERATED
Логическое выражение	BOOLEAN
Целое число без знака (Примечание 2)	INTEGER (Примечание 3)
Байт (Строка)	OCTET STRING
ПРИМЕЧАНИЕ 1. – Может быть больше одного байта.	
ПРИМЕЧАНИЕ 2. – Целое число без знака рассматривается в Приложении С.	
ПРИМЕЧАНИЕ 3. – Кодирование BER INTEGER не подразумевает использования 4 байтов.	
ПРИМЕЧАНИЕ 4. – Когда содержимое строки представляет собой символы ASCII, строка должна быть закодирована как IA5String, но как UTF8String, если она содержит какие-либо символы не-ASCII.	

Определение кодирования значения байтовой строки см. в 8.7/X.690.

```
MEDIA-GATEWAY-CONTROL {itu-t(0) recommendation(0) h(8) h248(248) modules(0)
media-gateway-control(0) version3(3)}
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
MegacoMessage ::= SEQUENCE
```

```

{
    authHeader      AuthenticationHeader OPTIONAL,
    mess            Message
}

AuthenticationHeader ::= SEQUENCE
{
    secParmIndex    SecurityParmIndex,
    seqNum          SequenceNum,
    ad              AuthData
}

SecurityParmIndex ::= OCTET STRING(SIZE(4))
SequenceNum       ::= OCTET STRING(SIZE(4))
AuthData          ::= OCTET STRING(SIZE(12..32))
Message           ::= SEQUENCE
{
    version         INTEGER(0..99),
    -- Версия протокола, определенного здесь, равняется 3.
    mId            Mid, -- Имя/ адрес сообщения отправителя
    messageBody    CHOICE
    {
        messageError      ErrorDescriptor,
        transactions      SEQUENCE OF Transaction
    },
    ...
}

Mid ::= CHOICE
{
    ip4Address      IP4Address,
    ip6Address      IP6Address,
    domainName      DomainName,
    deviceName      PathName,
    mtpAddress      OCTET STRING(SIZE(2..4)),
    -- Структура адресации mtpAddress:
    -- 25 - 15 0
    -- | PC | NI |
    -- 24 - 14 битов 2 бита
    -- ПРИМЕЧАНИЕ - 14 битов определены для международного использования.
    -- Существуют две национальные опции, где код точки - 16 или 24 бита.
    -- Для байтового выравнивания mtpAddress, MSBs должен быть закодирован как
    0s.
    ...
}

DomainName ::= SEQUENCE
{
    name            IA5String,
    -- Название начинается с буквенно-цифрового символа, за которым следует
    -- последовательность
    -- буквенно-цифровых символов, дефисов и точек. Две точки не должны
    -- следовать одна за другой.
    portNumber      INTEGER(0..65535) OPTIONAL
}

IP4Address ::= SEQUENCE
{
    address         OCTET STRING(SIZE(4)),
    portNumber      INTEGER(0..65535) OPTIONAL
}

IP6Address ::= SEQUENCE
{

```



```

    address                OCTET STRING (SIZE(16)),
    portNumber             INTEGER(0..65535) OPTIONAL
}

PathName                 ::= IA5String(SIZE (1..64))
-- См. А.3

Transaction ::= CHOICE
{
    transactionRequest     TransactionRequest,
    transactionPending     TransactionPending,
    transactionReply       TransactionReply,
    transactionResponseAck TransactionResponseAck,
    -- использование подтверждения приема зависит от основного вида передачи
    ...,
    segmentReply           SegmentReply
}

TransactionId            ::= INTEGER(0..4294967295) -- 32-битовое целое число без
знака

TransactionRequest ::= SEQUENCE
{
    transactionId          TransactionId,
    actions                SEQUENCE OF ActionRequest,
    ...
}

TransactionPending ::= SEQUENCE
{
    transactionId          TransactionId,
    ...
}

TransactionReply        ::= SEQUENCE
{
    transactionId          TransactionId,
    immAckRequired         NULL OPTIONAL,
    transactionResult      CHOICE
    {
        transactionError   ErrorDescriptor,
        actionReplies      SEQUENCE OF ActionReply
    },
    ...,
    segmentNumber          SegmentNumber OPTIONAL,
    segmentationComplete   NULL OPTIONAL
}

SegmentReply           ::= SEQUENCE
{
    transactionId          TransactionId,
    segmentNumber          SegmentNumber,
    segmentationComplete   NULL OPTIONAL,
    ...
}

SegmentNumber          ::= INTEGER(0..65535)

TransactionResponseAck ::= SEQUENCE OF TransactionAck
TransactionAck         ::= SEQUENCE
{
    firstAck              TransactionId,
    lastAck               TransactionId OPTIONAL
}

```

```

ErrorDescriptor      ::= SEQUENCE
{
    errorCode         ErrorCode,
    errorText         ErrorText OPTIONAL
}

ErrorCode           ::= INTEGER(0..65535)
-- Вопросы взаимодействия с IANA в отношении кодов ошибок см. в пункте 14

ErrorText           ::= IA5String

ContextID           ::= INTEGER(0..4294967295)
-- Значение Context NULL: 0
-- Значение Context CHOOSE: 4294967294 (0xFFFFFFFFE)
-- Значение Context ALL: 4294967295 (0xFFFFFFFFF)

ActionRequest       ::= SEQUENCE
{
    contextId         ContextID,
    contextRequest    ContextRequest OPTIONAL,
    contextAttrAuditReq ContextAttrAuditRequest OPTIONAL,
    commandRequests   SEQUENCE OF CommandRequest
}

ActionReply         ::= SEQUENCE
{
    contextId         ContextID,
    errorDescriptor   ErrorDescriptor OPTIONAL,
    contextReply      ContextRequest OPTIONAL,
    commandReply      SEQUENCE OF CommandReply
}

ContextRequest      ::= SEQUENCE
{
    priority          INTEGER(0..15) OPTIONAL,
    emergency         BOOLEAN OPTIONAL,
    topologyReq       SEQUENCE OF TopologyRequest OPTIONAL,
    ...,
    iepscallind       BOOLEAN OPTIONAL,
    contextProp       SEQUENCE OF PropertyParm OPTIONAL,
    contextList       SEQUENCE OF ContextID OPTIONAL
}
-- При возвращении contextList, contextId в структурном элементе ActionReply
будет возвращать contextId от связанного ActionRequest.

ContextAttrAuditRequest ::= SEQUENCE
{
    topology          NULL OPTIONAL,
    emergency         NULL OPTIONAL,
    priority          NULL OPTIONAL,
    ...,
    iepscallind       NULL OPTIONAL,
    contextPropAud    SEQUENCE OF IndAudPropertyParm OPTIONAL,
    selectpriority    INTEGER(0..15) OPTIONAL,
    -- выбрать заданный приоритет
    selectemergency   BOOLEAN OPTIONAL,
    -- выбрать, если аварийный вызов установлен/ не установлен (T/F)
    selectiepscalleind BOOLEAN OPTIONAL,
    -- выбрать, если IEPS установлен/не установлен (T/F)
    selectLogic       SelectLogic OPTIONAL -- по умолчанию задано AND
}

SelectLogic ::= CHOICE

```

```

{
    andAUDITSelect      NULL, -- удовлетворены все условия фильтра
    orAUDITSelect       NULL, -- удовлетворено по крайней мере одно условие
    фильтра
    ...
}

CommandRequest ::= SEQUENCE
{
    command              Command,
    optional              NULL OPTIONAL,
    wildcardReturn       NULL OPTIONAL,
    ...
}

Command ::= CHOICE
{
    addReq               AmmRequest,
    moveReq              AmmRequest,
    modReq               AmmRequest,
    -- запросы Add, Move, Modify имеют те же параметры
    subtractReq          SubtractRequest,
    auditCapRequest      AuditRequest,
    auditValueRequest    AuditRequest,
    notifyReq            NotifyRequest,
    serviceChangeReq     ServiceChangeRequest,
    ...
}

CommandReply ::= CHOICE
{
    addReply              AmmsReply,
    moveReply             AmmsReply,
    modReply              AmmsReply,
    subtractReply         AmmsReply,
    -- ответы Add, Move, Modify, Subtract имеют те же параметры
    auditCapReply         AuditReply,
    auditValueReply       AuditReply,
    notifyReply           NotifyReply,
    serviceChangeReply    ServiceChangeReply,
    ...
}

TopologyRequest ::= SEQUENCE
{
    terminationFrom      TerminationID,
    terminationTo        TerminationID,
    topologyDirection    ENUMERATED
    {
        bothway(0),
        isolate(1),
        oneway(2)
    },
    ...,
    streamID              StreamID OPTIONAL,
    topologyDirectionExtension    ENUMERATED
    {
        onewayexternal(0),
        onewayboth(1),
        ...
    }
}

AmmRequest ::= SEQUENCE
{

```

```

    terminationID      TerminationIDList,
    descriptors        SEQUENCE OF AmmDescriptor,
    -- В последовательности разрешено не более одного дескриптора каждого типа
    (см. AmmDescriptor).
    ...
}

AmmDescriptor ::= CHOICE
{
    mediaDescriptor      MediaDescriptor,
    modemDescriptor      ModemDescriptor,
    muxDescriptor        MuxDescriptor,
    eventsDescriptor     EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor    SignalsDescriptor,
    digitMapDescriptor   DigitMapDescriptor,
    auditDescriptor      AuditDescriptor,
    ...,
    statisticsDescriptor StatisticsDescriptor
}

AmmsReply ::= SEQUENCE
{
    terminationID      TerminationIDList,
    terminationAudit   TerminationAudit OPTIONAL,
    ...
}

SubtractRequest ::= SEQUENCE
{
    terminationID      TerminationIDList,
    auditDescriptor    AuditDescriptor OPTIONAL,
    ...
}

AuditRequest ::= SEQUENCE
{
    terminationID      TerminationID,
    auditDescriptor    AuditDescriptor,
    ...,
    terminationIDList  TerminationIDList OPTIONAL
}
-- при использовании структурного элемента terminationIDList в AuditRequest,
terminationID должен содержать первое завершение в списке.

AuditReply ::= CHOICE
{
    contextAuditResult  TerminationIDList,
    error                ErrorDescriptor,
    auditResult          AuditResult,
    ...,
    auditResultTermList TermListAuditResult
}

AuditResult ::= SEQUENCE
{
    terminationID      TerminationID,
    terminationAuditResult TerminationAudit
}

TermListAuditResult ::= SEQUENCE
{
    terminationIDList  TerminationIDList,
    terminationAuditResult TerminationAudit,
    ...
}

```

```

}

TerminationAudit      ::= SEQUENCE OF AuditReturnParameter

AuditReturnParameter ::= CHOICE
{
    errorDescriptor      ErrorDescriptor,
    mediaDescriptor      MediaDescriptor,
    modemDescriptor      ModemDescriptor,
    muxDescriptor        MuxDescriptor,
    eventsDescriptor     EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor    SignalsDescriptor,
    digitMapDescriptor   DigitMapDescriptor,
    observedEventsDescriptor ObservedEventsDescriptor,
    statisticsDescriptor StatisticsDescriptor,
    packagesDescriptor   PackagesDescriptor,
    emptyDescriptors     AuditDescriptor,
    ...
}

AuditDescriptor      ::= SEQUENCE
{
    auditToken          BIT STRING
    {
        muxToken(0),
        modemToken(1),
        mediaToken(2),
        eventsToken(3),
        signalsToken(4),
        digitMapToken(5),
        statsToken(6),
        observedEventsToken(7),
        packagesToken(8),
        eventBufferToken(9)
    } OPTIONAL,
    ...,
    auditPropertyToken SEQUENCE OF IndAuditParameter OPTIONAL
}

IndAuditParameter ::= CHOICE
{
    indaudmediaDescriptor      IndAudMediaDescriptor,
    indaudeventsDescriptor     IndAudEventsDescriptor,
    indaudeventBufferDescriptor IndAudEventBufferDescriptor,
    indaudsignalsDescriptor    IndAudSignalsDescriptor,
    indauidigitMapDescriptor    IndAudDigitMapDescriptor,
    indaudstatisticsDescriptor IndAudStatisticsDescriptor,
    indaudpackagesDescriptor   IndAudPackagesDescriptor,
    ...
}

IndAudMediaDescriptor ::= SEQUENCE
{
    termStateDescr      IndAudTerminationStateDescriptor OPTIONAL,
    streams              CHOICE
    {
        oneStream        IndAudStreamParms,
        multiStream      SEQUENCE OF IndAudStreamDescriptor
    } OPTIONAL,
    ...
}

IndAudStreamDescriptor ::= SEQUENCE

```

```

{
    streamID                StreamID,
    streamParms             IndAudStreamParms
}

IndAudStreamParms ::= SEQUENCE
{
    localControlDescriptor  IndAudLocalControlDescriptor OPTIONAL,
    localDescriptor         IndAudLocalRemoteDescriptor OPTIONAL,
    remoteDescriptor        IndAudLocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor    IndAudStatisticsDescriptor OPTIONAL
}

IndAudLocalControlDescriptor ::= SEQUENCE
{
    streamMode              NULL OPTIONAL,
    reserveValue            NULL OPTIONAL,
    reserveGroup            NULL OPTIONAL,
    propertyParms           SEQUENCE OF IndAudPropertyParm OPTIONAL,
    ...,
    streamModeSel          StreamMode OPTIONAL
}
-- не должен иметь одновременно streamMode и streamModeSel
-- если присутствуют оба, нужно использовать/ учитывать только streamModeSel

IndAudPropertyParm ::= SEQUENCE
{
    name                    PkgdName,
    ...,
    propertyParms           PropertyParm OPTIONAL
}

-- выбрать на основе значений свойства
-- логика выбора AND/OR определяется на уровне контекста

IndAudLocalRemoteDescriptor ::= SEQUENCE
{
    propGroupID             INTEGER(0..65535) OPTIONAL,
    propGrps                IndAudPropertyGroup,
    ...
}

IndAudPropertyGroup      ::= SEQUENCE OF IndAudPropertyParm

IndAudTerminationStateDescriptor ::= SEQUENCE
{
    propertyParms           SEQUENCE OF IndAudPropertyParm,
    eventBufferControl      NULL OPTIONAL,
    serviceState            NULL OPTIONAL,
    ...,
    serviceStateSel        ServiceState OPTIONAL
}
-- не должен иметь serviceState и serviceStateSel одновременно
-- если присутствуют оба, использовать нужно только serviceStateSel

IndAudEventsDescriptor ::= SEQUENCE
{
    requestID               RequestID OPTIONAL,
    pkgdName                PkgdName,
    streamID                StreamID OPTIONAL,
    ...
}

```

```

IndAudEventBufferDescriptor ::= SEQUENCE
{
    eventName                PkgdName,
    streamID                  StreamID OPTIONAL,
    ...
}

IndAudSignalsDescriptor ::= CHOICE
{
    signal                    IndAudSignal,
    seqSigList                IndAudSeqSigList,
    ...
}

IndAudSeqSigList           ::= SEQUENCE
{
    id                        INTEGER(0..65535),
    signalList                IndAudSignal OPTIONAL
}

IndAudSignal               ::= SEQUENCE
{
    signalName                PkgdName,
    streamID                  StreamID OPTIONAL,
    ...,
    signalRequestID          RequestID OPTIONAL
}

IndAudDigitMapDescriptor ::= SEQUENCE
{
    digitMapName              DigitMapName OPTIONAL
}

IndAudStatisticsDescriptor ::= SEQUENCE
{
    statName                  PkgdName
}

IndAudPackagesDescriptor ::= SEQUENCE
{
    packageName                Name,
    packageVersion            INTEGER(0..99),
    ...
}

NotifyRequest              ::= SEQUENCE
{
    terminationID             TerminationIDList,
    observedEventsDescriptor  ObservedEventsDescriptor,
    errorDescriptor           ErrorDescriptor OPTIONAL,
    ...
}

NotifyReply                ::= SEQUENCE
{
    terminationID             TerminationIDList,
    errorDescriptor           ErrorDescriptor OPTIONAL,
    ...
}

ObservedEventsDescriptor ::= SEQUENCE
{
    requestId                 RequestID,
    observedEventList         SEQUENCE OF ObservedEvent
}

```

```

ObservedEvent ::= SEQUENCE
{
    eventName          EventName,
    streamID           StreamID OPTIONAL,
    eventParList       SEQUENCE OF EventParameter,
    timeNotation       TimeNotation OPTIONAL,
    ...
}

EventName ::= PkgdName

EventParameter ::= SEQUENCE
{
    eventParameterName Name,
    value               Value,
    -- По использованию extraInfos см. комментарий, относящийся к PropertyParm
    extraInfo           CHOICE
    {
        relation        Relation,
        range            BOOLEAN,
        sublist         BOOLEAN
    } OPTIONAL,
    ...
}

ServiceChangeRequest ::= SEQUENCE
{
    terminationID      TerminationIDList,
    serviceChangeParms ServiceChangeParm,
    ...
}

ServiceChangeReply ::= SEQUENCE
{
    terminationID      TerminationIDList,
    serviceChangeResult ServiceChangeResult,
    ...
}

-- Для ServiceChangeResult нет обязательных параметров. Отсюда отличие
ServiceChangeParm от ServiceChangeResParm.
ServiceChangeResult ::= CHOICE
{
    errorDescriptor    ErrorDescriptor,
    serviceChangeResParms ServiceChangeResParm
}

WildcardField ::= OCTET STRING(SIZE(1))

TerminationID ::= SEQUENCE
{
    wildcard           SEQUENCE OF WildcardField,
    id                 OCTET STRING(SIZE(1..8)),
    ...
}

-- Объяснение алгоритма обобщения см. в А.1.
-- TerminationID 0xFFFFFFFFFFFFFFFF обозначает завершение Root.

TerminationIDList ::= SEQUENCE OF TerminationID

```



```

MediaDescriptor      ::= SEQUENCE
{
    termStateDescr   TerminationStateDescriptor OPTIONAL,
    streams          CHOICE
    {
        oneStream     StreamParms,
        multiStream   SEQUENCE OF StreamDescriptor
    } OPTIONAL,
    ...
}

StreamDescriptor    ::= SEQUENCE
{
    streamID         StreamID,
    streamParms     StreamParms
}

StreamParms        ::= SEQUENCE
{
    localControlDescriptor LocalControlDescriptor OPTIONAL,
    localDescriptor   LocalRemoteDescriptor OPTIONAL,
    remoteDescriptor  LocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor StatisticsDescriptor OPTIONAL
}

LocalControlDescriptor ::= SEQUENCE
{
    streamMode       StreamMode OPTIONAL,
    reserveValue     BOOLEAN OPTIONAL,
    reserveGroup     BOOLEAN OPTIONAL,
    propertyParms    SEQUENCE OF PropertyParm,
    ...
}

StreamMode         ::= ENUMERATED
{
    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
    ...
}

```

-- В PropertyParm, значение – ПОСЛЕДОВАТЕЛЬНОСТЬ (SEQUENCE OF) байтовой строки.
-- Если передачу осуществляет MGC, интерпретировать нужно следующим образом:
-- пустая последовательность означает CHOOSE один элемент последовательности
-- определяет значение.
-- Если поле подписка не выбрано, более длинная последовательность обозначает
-- "выберите одно из значений" (т. е. value1 OR value2 OR...)
-- Если поле подписка выбрано, последовательность более, чем с одним элементом,
-- кодирует значение свойства со значением из списка
-- (т. е., value1 AND value2 AND...)
-- Поле relation может быть выбрано только если длина последовательности
-- значений равна 1. Оно показывает, что MG нужно выбрать значение для свойства.
-- Например, x > 3 (используя значение greaterThan
-- для relation) дает указание MG выбрать любое значение больше 3 для свойства
x.
-- Поле range может быть выбрано, только длина
-- последовательности значений равна 2.
-- Она показывает, что MG нужно выбрать значение в диапазоне от первого байта
-- в последовательности значений до заключительного байта

-- в последовательности значений, включая граничные значения
 -- Если передачу осуществляет MG, множество значений, диапазон или поле relation
 -- могут содержать только ответы на запрос AuditCapability.

PropertyParm ::= SEQUENCE

```
{
  name          PkgdName,
  value         SEQUENCE OF OCTET STRING,
  extraInfo     CHOICE
  {
    relation    Relation,
    range       BOOLEAN,
    sublist     BOOLEAN
  } OPTIONAL,
  ...
}
```

Name ::= OCTET STRING(SIZE(2))

PkgdName ::= OCTET STRING(SIZE(4))

-- представляет Название комплекта (2 байта) плюс названия свойства, события,
 -- сигнала или StatisticsID. (2 байта)
 -- Для обобщения комплекта используйте 0xFFFF для первых двух байтов,
 -- использование CHOOSE не разрешено. Для ссылки на собственный тэг свойства,
 -- рассматриваемый в Приложении C, используйте 0x0000 в качестве первых двух
 -- байтов.
 -- Для обобщения PropertyID, EventID, SignalID, или StatisticsID, используйте
 -- 0xFFFF для последних двух байтов, использование CHOOSE не разрешено.
 -- Обобщение Названия комплекта разрешается только если PropertyID,
 -- EventID, SignalID, или StatisticsID также обобщены.

Relation ::= ENUMERATED

```
{
  greaterThan(0),
  smallerThan(1),
  unequalTo(2),
  ...}
```

LocalRemoteDescriptor ::= SEQUENCE

```
{
  propGrps     SEQUENCE OF PropertyGroup,
  ...
}
```

PropertyGroup ::= SEQUENCE OF PropertyParm

TerminationStateDescriptor ::= SEQUENCE

```
{
  propertyParms SEQUENCE OF PropertyParm,
  eventBufferControl EventBufferControl OPTIONAL,
  serviceState  ServiceState OPTIONAL,
  ...
}
```

EventBufferControl ::= ENUMERATED

```
{
  off(0),
  lockStep(1),
  ...
}
```

ServiceState ::= ENUMERATED

```
{
  test(0),
```

```

    outOfSvc (1),
    inSvc (2),
    ...
}

MuxDescriptor ::= SEQUENCE
{
    muxType          MuxType,
    termList         SEQUENCE OF TerminationID,
    nonStandardData NonStandardData OPTIONAL,
    ...
}

MuxType ::= ENUMERATED
{
    h221 (0),
    h223 (1),
    h226 (2),
    v76 (3),
    ...,
    nx64k (4)
}

StreamID ::= INTEGER (0..65535) -- 16-битное целое число без знака

EventsDescriptor ::= SEQUENCE
{
    requestID      RequestID OPTIONAL,
    -- RequestID должен присутствовать, если eventList
    -- не ПУСТ
    eventList      SEQUENCE OF RequestedEvent,
    ...
}

RequestedEvent ::= SEQUENCE
{
    pkgdName       PkgdName,
    streamID       StreamID OPTIONAL,
    eventAction    RequestedActions OPTIONAL,
    evParList      SEQUENCE OF EventParameter,
    ...
}

RegulatedEmbeddedDescriptor ::= SEQUENCE
{
    secondEvent    SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...
}

NotifyBehaviour ::= CHOICE
{
    notifyImmediate      NULL,
    notifyRegulated     RegulatedEmbeddedDescriptor,
    neverNotify          NULL,
    ...
}

RequestedActions ::= SEQUENCE
{
    keepActive        BOOLEAN OPTIONAL,
    eventDM           EventDM OPTIONAL,
    secondEvent       SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...,
}

```

```

    notifyBehaviour      NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventDM                ::= CHOICE
{
    digitMapName        DigitMapName,
    digitMapValue       DigitMapValue
}

SecondEventsDescriptor ::= SEQUENCE
{
    requestID           RequestID OPTIONAL,
    eventList           SEQUENCE OF SecondRequestedEvent,
    ...
}

SecondRequestedEvent   ::= SEQUENCE
{
    pkgdName            PkgdName,
    streamID            StreamID OPTIONAL,
    eventAction         SecondRequestedActions OPTIONAL,
    evParList           SEQUENCE OF EventParameter,
    ...
}

SecondRequestedActions ::= SEQUENCE
{
    keepActive          BOOLEAN OPTIONAL,
    eventDM             EventDM OPTIONAL,
    signalsDescriptor   SignalsDescriptor OPTIONAL,
    ...,
    notifyBehaviour     NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventBufferDescriptor  ::= SEQUENCE OF EventSpec

EventSpec              ::= SEQUENCE
{
    eventName           EventName,
    streamID            StreamID OPTIONAL,
    eventParList        SEQUENCE OF EventParameter,
    ...
}

SignalsDescriptor      ::= SEQUENCE OF SignalRequest

SignalRequest          ::= CHOICE
{
    signal              Signal,
    seqSigList          SeqSigList,
    ...
}

SeqSigList             ::= SEQUENCE
{
    id                  INTEGER(0..65535),
    signalList          SEQUENCE OF Signal
}

Signal                 ::= SEQUENCE
{
    signalName          SignalName,

```

```

    streamID          StreamID OPTIONAL,
    sigType           SignalType OPTIONAL,
    duration          INTEGER (0..65535) OPTIONAL,
    notifyCompletion  NotifyCompletion OPTIONAL,
    keepActive        BOOLEAN OPTIONAL,
    sigParList        SEQUENCE OF SigParameter,
    ...,
    direction         SignalDirection OPTIONAL,
    requestID         RequestID OPTIONAL,
    intersigDelay     INTEGER (0..65535) OPTIONAL
}

SignalType          ::= ENUMERATED
{
    brief(0),
    onOff(1),
    timeOut(2),
    ...
}

SignalDirection     ::= ENUMERATED
{
    internal(0),
    external(1),
    both(2),
    ...
}

SignalName          ::= PkgdName

NotifyCompletion     ::= BIT STRING
{
    onTimeOut(0), onInterruptByEvent(1),
    onInterruptByNewSignalDescr(2), otherReason(3), onIteration(4)
}

SigParameter        ::= SEQUENCE
{
    sigParameterName  Name,
    value             Value,
    -- По использованию extraInfo см. комментарий, касающийся PropertyParm
    extraInfo         CHOICE
    {
        relation      Relation,
        range          BOOLEAN,
        sublist        BOOLEAN
    } OPTIONAL,
    ...
}

-- Для AuditCapReply со всеми событиями, RequestID должен быть ALL.
-- ALL представлен 0xffffffff.
RequestID            ::= INTEGER(0..4294967295) -- 32-битное целое число без
знака

ModemDescriptor      ::= SEQUENCE
{
    mt1               SEQUENCE OF ModemType,
    mpl               SEQUENCE OF PropertyParm,
    nonStandardData   NonStandardData OPTIONAL
}

ModemType            ::= ENUMERATED
{

```

```

    v18(0),
    v22(1),
    v22bis(2),
    v32(3),
    v32bis(4),
    v34(5),
    v90(6),
    v91(7),
    synchISDN(8),
    ...
}

DigitMapDescriptor ::= SEQUENCE
{
    digitMapName          DigitMapName OPTIONAL,
    digitMapValue         DigitMapValue OPTIONAL
}

DigitMapName          ::= Name

DigitMapValue         ::= SEQUENCE
{
    startTimer           INTEGER(0..99) OPTIONAL,
    shortTimer           INTEGER(0..99) OPTIONAL,
    longTimer            INTEGER(0..99) OPTIONAL,
    digitMapBody         IA5String,
    -- Единицами измерения являются секунды для начального,
    -- короткого и длинного таймеров, и сотые доли миллисекунд
    -- для таймера продолжительности. Таким образом, диапазон для начального,
    -- короткого и длинного таймеров составляет от 1 до 99 секунд, а таймера
    -- продолжительности от 100 мс до 9.9 с
    -- Объяснение синтаксиса DigitMap см. в А.3
    ...,
    durationTimer        INTEGER (0..99) OPTIONAL
}

ServiceChangeParm    ::= SEQUENCE
{
    serviceChangeMethod  ServiceChangeMethod,
    serviceChangeAddress ServiceChangeAddress OPTIONAL,
    serviceChangeVersion INTEGER(0..99) OPTIONAL,
    serviceChangeProfile ServiceChangeProfile OPTIONAL,
    serviceChangeReason  Value,
    -- serviceChangeReason состоит из цифрового кода причины
    -- и дополнительного текстового описания.
    -- serviceChangeReason должен представлять собой строку, состоящую из
    -- десятичного кода причины, за ним дополнительно следует один
    -- символ пробела и текстовая строка описания.
    -- Сначала эту строку кодируют по BER как IA5String.
    -- Затем результат этого кодирования по BER кодируется как
    -- тип БАЙТОВАЯ СТРОКА ASN.1, выполняя "двойное свертывание"
    -- значения также, как это было сделано для элементов комплекта.
    serviceChangeDelay   INTEGER(0..4294967295) OPTIONAL,
    -- 32-битное целое число без знака
    serviceChangeMgcId   Mid OPTIONAL,
    timeStamp            TimeNotation OPTIONAL,
    nonStandardData      NonStandardData OPTIONAL,
    ...,
    serviceChangeInfo    AuditDescriptor OPTIONAL,
    serviceChangeIncompleteFlag NULL OPTIONAL
}

ServiceChangeAddress ::= CHOICE
{

```

```

    portNumber          INTEGER(0..65535),    -- TCP/UDP номер порта
    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName          DomainName,
    deviceName          PathName,
    mtpAddress          OCTET STRING(SIZE(2..4)),
    ...
}

ServiceChangeResParm ::= SEQUENCE
{
    serviceChangeMgcId      MId OPTIONAL,
    serviceChangeAddress    ServiceChangeAddress OPTIONAL,
    serviceChangeVersion    INTEGER(0..99) OPTIONAL,
    serviceChangeProfile    ServiceChangeProfile OPTIONAL,
    timestamp               TimeNotation OPTIONAL,
    ...
}

ServiceChangeMethod ::= ENUMERATED
{
    failover(0),
    forced(1),
    graceful(2),
    restart(3),
    disconnected(4),
    handOff(5),
    ...
}

ServiceChangeProfile ::= SEQUENCE
{
    profileName             IA5String(SIZE (1..67))
    -- 64 СИМВОЛОВ ДЛЯ НАЗВАНИЯ, 1 ДЛЯ "/", 2 ДЛЯ ТОГО, ЧТОБЫ ВЕРСИЯ СОВПАЛА С
ABNF
}

PackagesDescriptor ::= SEQUENCE OF PackagesItem

PackagesItem ::= SEQUENCE
{
    packageName            Name,
    packageVersion         INTEGER(0..99),
    ...
}

StatisticsDescriptor ::= SEQUENCE OF StatisticsParameter

StatisticsParameter ::= SEQUENCE
{
    statName               PkgdName,
    statValue              Value OPTIONAL
}
-- Если статистика состоит из подписков, в statValue будет более одной
-- байтовой строки.

NonStandardData ::= SEQUENCE
{
    nonStandardIdentifier  NonStandardIdentifier,
    data                  OCTET STRING
}

```

```

NonStandardIdentifier ::= CHOICE
{
    object                OBJECT IDENTIFIER,
    h221NonStandard      H221NonStandard,
    experimental         IA5String(SIZE(8)),
    -- первые два символа должны быть "X-" или "X+"
    ...
}

H221NonStandard ::= SEQUENCE
{
    t35CountryCode1      INTEGER(0..255),
    t35CountryCode2      INTEGER(0..255),      -- страна, согласно Т.35
    t35Extension         INTEGER(0..255),      -- назначается в масштабе страны
    manufacturerCode     INTEGER(0..65535),    -- назначается в масштабе страны
    ...
}

TimeNotation ::= SEQUENCE
{
    date                 IA5String(SIZE(8)),    -- формат yyyymmdd
    time                 IA5String(SIZE(8))     -- формат hhmmssss
    -- согласно ISO 8601:2004
}

Value ::= SEQUENCE OF OCTET STRING

END

```

A.3 DigitMaps и имя пути (файла)

С точки зрения синтаксиса DigitMap представляют собой строки с налагаемыми на них синтаксическими ограничениями. Синтаксис действительных DigitMap определяется в ABNF (RFC 2234). Синтаксис для DigitMaps, представленный в данном пункте, предназначен только для наглядности. Определение digitMap в Приложении В является приоритетным в случае различий ними.

```

digitMap                = (digitString / LWSP "(" LWSP digitStringList LWSP
                           ")" LWSP)

digitStringList         = digitString *(LWSP "|" LWSP digitString)

digitString             = 1*(digitStringElement)

digitStringElement     = digitPosition [DOT]

digitPosition           = digitMapLetter / digitMapRange

digitMapRange           = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter             = *((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter          = DIGIT                ;Основные символы события
                        / %x41-4B / %x61-6B ; a-k, A-K
                        / "L" / "S" / "T"     ;Межсобытийные таймеры
                                                ; (длинный, короткий, начальный)
                        / "Z"                ;Модификатор большой продолжительности

DOT                     = %x2E                ; "."

SP                      = %x20                ; пробел

HTAB                    = %x09                ; символ горизонтальной табуляции

```


CR	= %x0D	; возврат каретки
LF	= %x0A	; символ перехода на новую строку
LWSP	= *(WSP / COMMENT / EOL)	
EOL	= (CR [LF] / LF)	
WSP	= SP / HTAB	; пробельный символ
SafeChar	= DIGIT / ALPHA / "+" / "-" / "&" / "! " / "_" / "/" / "'" / "?" / "@" / "^" / "`" / "~" / "*" / "\$" / "\" / " (" / ")" / "%" / " " / "."	
RestChar	= ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" / "<" / ">" / "="	
DIGIT	= %x30-39	; 0-9
ALPHA	= %x41-5A / %x61-7A	; A-Z / a-z

Имя пути файла также представляет собой строку с налагаемыми на нее синтаксическими ограничениями. Правило вывода ABNF, определяющее его (имя пути) скопировано из Приложения В.

; Суммарная длина pathNAME не должна превышать 64 символа.

```
pathname = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$" )
["@" pathDomainName ]
```

; ABNF разрешает появление двух и более последовательных ".", хотя это является бессмысленным в доменном имени пути.

```
pathDomainName = (ALPHA / DIGIT / "*" ) *63(ALPHA / DIGIT / "-" / "*" / ".")
```

```
NAME = ALPHA *63(ALPHA / DIGIT / "_")
```

Приложение В

Текстовое кодирование протокола

В.1 Кодирование групповых символов замены

При текстовом кодировании данного протокола, поскольку идентификаторы TerminationID являются произвольными, при разумном выборе названий групповой символ замены "*" может быть более полезен. При появлении группового символа замены, он будет "совпадать" со всеми TerminationID, имеющими такие же предыдущие и последующие символы (если соответствуют). Например, если бы существовали идентификаторы TerminationID R13/3/1, R13/3/2 и R13/3/3, TerminationID R13/3/* совпал бы со всеми из них. Существуют некоторые обстоятельства, при которых необходимо сделать ссылку на все завершения. Идентификатора TerminationID "*" достаточно, и ссылка осуществляется на ALL. CHOOSSE TerminationID "\$" может быть использован для сигнализации MG о том, что ему нужно создать непродолжительное завершение или выбрать из неактивных физических завершений.

В.2 Спецификация ABNF

Синтаксис протокола представлен в ABNF согласно RFC 2234.

ПРИМЕЧАНИЕ 1. – Данная спецификация синтаксиса не вводит в действие все ограничения по включению элементов и значений. Некоторые дополнительные ограничения приводятся в комментариях, а другие ограничения появляются в тексте данной Рекомендации. Данные дополнительные ограничения являются частью протокола, даже хотя они не вводятся в действие данной Рекомендацией.

ПРИМЕЧАНИЕ 2. – Синтаксис зависит от содержания. Например, "Add" может быть AddToken или NAME в зависимости от контекста, в котором он появляется.

Все в ABNF и текстовом кодимым от регистра. Сюда также относятся идентификаторы TerminationID, DigitMapID, и т. д. SDP является чувствительным к регистру, согласно RFC 2327.

; ПРИМЕЧАНИЕ. – ABNF в данном разделе использует структурный элемент VALUE (или
; списки структурных элементов VALUE) для кодирования различных значений
; элементов комплекта (свойства, параметры сигнала, и т. д.).
; Типы данных значений варьируются и определяются в соответствующем
; определении комплекта. Некоторые такие типы описаны в п. 12.2.
;
; Спецификация ABNF для VALUE разрешает форму quotedString или совокупность
SafeChars. Кодирование значений элементов комплекта в
; ABNF VALUES определяется ниже. Если в кодировании типа разрешаются символы,
отличные от
; SafeChars, для всех значений этого типа нужно использовать форму quotedString
; даже для специальных значений, состоящих только из SafeChars.
;
; Строка: В строке должна использоваться форма VALUE quotedString, строка может
содержать все, что разрешено в форме quotedString.
;
; Целое число, Двойное число, Целое число без знака: Десятичные значения могут
быть закодированы при помощи символов 0-9.
; Перед шестнадцатеричными значениями должно стоять '0x', в них могут
использоваться символы 0-9, a-f, A-F. Восьмеричный формат не поддерживается.
; Отрицательные целые числа начинаются с '-' и должны быть десятичными. Должны
использоваться форма VALUE SafeChar
;
; Буква: Кодирование типа UTF-8 отдельной буквы, окруженное двойными кавычками.
;
; Перечисление: В перечислении должна использоваться форма SafeChar VALUE,
; перечисление может содержать все, что разрешено в форме SafeChar.
;
; Логическое выражение: Логические значения кодируются в виде "on" и "off", и не
зависят от регистра. Нужно использовать форму SafeChar VALUE.
;
; Будущие типы: Все определенные типы должны соответствовать
; спецификации ABNF VALUE. В особенности, если в кодировании типа
; разрешаются символы, отличные от SafeChars, для всех значений данного типа
; нужно использовать форму quotedString, даже для специальных значений,
состоящих только из SafeChars.
;
; Заметьте, что символ двойных кавычек в значении не используется.
;
; Заметьте, что SDP не разрешает пробельный символ в начале строки, Megaco
; ABNF разрешает пробельный символ до начала SDP в дескрипторе
; Local/Remote. Синтаксические анализаторы должны принимать пробельный символ
между
; LBRKT, следующим за меткой Local/Remote, и началом SDP.

megacoMessage = LWSP [authenticationHeader SEP] message

authenticationHeader = AuthToken EQUAL SecurityParmIndex COLON
SequenceNum COLON AuthData

SecurityParmIndex = "0x" 8 (HEXDIG)

SequenceNum = "0x" 8 (HEXDIG)

AuthData = "0x" 24*64 (HEXDIG)

Message = MegacoToken SLASH Version SEP mId SEP messageBody

; Версия протокола, определенного здесь, равняется 3

```

messageBody          = (errorDescriptor / transactionList)

transactionList      = 1*(transactionRequest / transactionReply /
                       transactionPending / transactionResponseAck /
                       segmentReply)
;Использование подтверждений приема ответа зависит от основного типа передачи

transactionPending   = PendingToken EQUAL TransactionID LBRKT RBRKT

transactionResponseAck = ResponseAckToken LBRKT transactionAck
                       *(COMMA transactionAck) RBRKT
transactionAck       = TransactionID / (TransactionID "-" TransactionID)

transactionRequest   = TransToken EQUAL TransactionID LBRKT
                       actionRequest *(COMMA actionRequest) RBRKT

actionRequest        = CtxToken EQUAL ContextID LBRKT ((contextRequest
[COMMA commandRequestList]) /
commandRequestList) RBRKT

contextRequest       = ((contextProperties [COMMA contextAudit])
/ contextAudit)

contextProperties     = contextProperty *(COMMA contextProperty)

; не более одного раза
; EmergencyOff предназначен для использования в направлении от MG к MGC только в
H.248.1 V1 и V2
; EmergencyToken или EmergencyOffToken, но не оба одновременно
contextProperty      = (topologyDescriptor / priority / EmergencyToken /
                       EmergencyOffToken / iepsValue /
                       contextAttrDescriptor)

contextAudit         = ContextAuditToken LBRKT (contextAuditProperties
*(COMMA contextAuditProperties)) /
indAudcontextAttrDescriptor RBRKT

; не более одного раза, кроме contextAuditSelector
contextAuditProperties = (TopologyToken / EmergencyToken /
PriorityToken / IEPSToken/ pkgdName /
contextAuditSelector)

; не более одного раза
contextAuditSelector = priority / emergencyValue / iepsValue /
contextAttrDescriptor / auditSelectLogic

auditSelectLogic = [ AndAUDITselectToken / OrAUDITselectToken ]
; Если пустой, подразумевается условия выбора AND.

indAudcontextAttrDescriptor
= ContextAttrToken LBRKT contextAuditProperties
*(COMMA contextAuditProperties) RBRKT

; "O-" обозначает дополнительную команду
; "W-" обозначает обобщенный ответ на команду
commandRequestList   = ["O-"] ["W-"] commandRequest *(COMMA ["O-"]
["W-"] commandRequest)

commandRequest       = (ammRequest / subtractRequest / auditRequest /
notifyRequest / serviceChangeRequest)

transactionReply     = ReplyToken EQUAL TransactionID [SLASH segmentNumber
[SLASH SegmentationCompleteToken]] LBRKT

```

```

[ImmAckRequiredToken COMMA]
(errorDescriptor / actionReplyList) RBRKT

segmentReply      = MessageSegmentToken EQUAL TransactionID SLASH
                  segmentNumber [SLASH SegmentationCompleteToken]

segmentNumber     = UINT16

actionReplyList   = actionReply *(COMMA actionReply)

actionReply       = CtxToken EQUAL ContextID [LBRKT (errorDescriptor /
                  commandReply /(commandReply COMMA
                  errorDescriptor)) RBRKT]

commandReply      = ((contextProperties [COMMA commandReplyList]) /
                  commandReplyList)

commandReplyList  = commandReplies *(COMMA commandReplies)

commandReplies    = (serviceChangeReply / auditReply / ammsReply /
                  notifyReply)

;Add, Move и Modify имеют те же параметры запроса
ammRequest        = (AddToken / MoveToken / ModifyToken) EQUAL
                  termIDList [LBRKT ammParameter *(COMMA
                  ammParameter) RBRKT]

; не более одного раза
ammParameter      = (mediaDescriptor / modemDescriptor / muxDescriptor /
                  eventsDescriptor / signalsDescriptor /
                  digitMapDescriptor / eventBufferDescriptor /
                  auditDescriptor / statisticsDescriptor)

ammsReply         = (AddToken / MoveToken / ModifyToken / SubtractToken)
                  EQUAL termIDList [LBRKT terminationAudit RBRKT]

subtractRequest   = SubtractToken EQUAL termIDList [LBRKT
                  auditDescriptor RBRKT]

auditRequest      = (AuditValueToken / AuditCapToken) EQUAL
                  termIDList LBRKT auditDescriptor RBRKT

auditReply        = (AuditValueToken / AuditCapToken)
                  (contextTerminationAudit / auditOther)

auditOther        = EQUAL termIDList [LBRKT terminationAudit RBRKT]

terminationAudit = auditReturnParameter *(COMMA auditReturnParameter)

contextTerminationAudit = EQUAL CtxToken (terminationIDList / LBRKT
                  errorDescriptor RBRKT)

auditReturnParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /
                  eventsDescriptor / signalsDescriptor /
                  digitMapDescriptor / observedEventsDescriptor /
                  eventBufferDescriptor / statisticsDescriptor /
                  packagesDescriptor / errorDescriptor /
                  auditReturnItem)

auditReturnItem   = (MuxToken / ModemToken / MediaToken /
                  DigitMapToken / StatsToken / ObservedEventsToken /
                  PackagesToken)

auditDescriptor   = AuditToken LBRKT [auditItem *(COMMA auditItem)]
                  RBRKT

```

```

notifyRequest      = NotifyToken EQUAL termIDList LBRKT
                    (observedEventsDescriptor [COMMA errorDescriptor])
                    RBRKT

notifyReply        = NotifyToken EQUAL termIDList [LBRKT errorDescriptor
                    RBRKT]

serviceChangeRequest = ServiceChangeToken EQUAL termIDList LBRKT
                    serviceChangeDescriptor RBRKT

serviceChangeReply  = ServiceChangeToken EQUAL termIDList [LBRKT
                    (errorDescriptor / serviceChangeReplyDescriptor)
                    RBRKT]

errorDescriptor    = ErrorToken EQUAL ErrorCode LBRKT [quotedString]
                    RBRKT

ErrorCode          = 1*4(DIGIT); could be extended

TransactionID      = UINT32

mId                = ((domainAddress / domainName) [":" portNumber]) /
                    mtpAddress / deviceName

; ABNF разрешает использование двух или более последовательных "." хотя в
; доменном имени это не имеет смысла.
domainName         = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" / ".")
                    ">"

deviceName         = pathNAME

;Значения 0x0, 0xFFFFFFFFE и 0xFFFFFFFF зарезервированы.
; '-' используется для нулевого контекста. '*' обозначает ALL. '$' обозначает
; CHOOSE.
ContextID         = (UINT32 / "*" / "-" / "$")

domainAddress      = "[" (IPv4address / IPv6address) "]"

;Определение адресов IPv6 содержится в RFC 2373.
IPv6address       = hexpart [":" IPv4address]

IPv4address       = V4hex DOT V4hex DOT V4hex DOT V4hex

V4hex             = 1*3(DIGIT); "0".."255"

; на данную продукцию ссылка не делается, хотя она встречается в RFC 2373
; IPv6prefix      = hexpart SLASH 1*2DIGIT

hexpart           = hexseq ":@" [hexseq] / ":@" [hexseq] / hexseq

hexseq           = hex4 *(":" hex4)

hex4              = 1*4HEXDIG

portNumber        = UINT16

; Структура адресации mtpAddress:
; 25 - 15 0
; | PC | NI |
; 24 - 14 битов 2 бита
; ПРИМЕЧАНИЕ. - 14 битов определены для международного использования.
; Там, где код точки (point code) 16 или 24 бита, существуют два государственных
; варианта.
; Для байтового выравнивания mtpAddress, MSBs должны быть закодированы как 0s.
; Байт должен быть представлен двумя шестнадцатеричными числами.
mtpAddress        = MTPToken LBRKT 4*8 (HEXDIG) RBRKT

```

```

termIDList          = (TerminationID / LSBRKT TerminationID 1*(COMMA
TerminationID) RSBRKT)

terminationIDList  = LBRKT TerminationID *(COMMA TerminationID) RBRKT

; Суммарная длина pathNAME не должна превышать 64 символа.
pathNAME           = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$")
["@" pathDomainName]

; ABNF разрешает использование двух или более последовательных ".", хотя в
доменном имени это не имеет смысла.
pathDomainName     = (ALPHA / DIGIT / "*") *63(ALPHA / DIGIT / "-" /
"*" / ".")

; '*' обозначает ALL. '$' обозначает CHOOSE.
TerminationID     = "ROOT" / pathNAME / "$" / "*"

mediaDescriptor    = MediaToken LBRKT mediaParm *(COMMA mediaParm) RBRKT

; не более одного terminationStateDescriptor
; и либо streamParm(s), либо streamDescriptor(s), но не оба одновременно
mediaParm          = (streamParm / streamDescriptor /
terminationStateDescriptor)

; не более одного на каждый элемент
streamParm         = (localDescriptor / remoteDescriptor /
localControlDescriptor / statisticsDescriptor)

streamDescriptor   = StreamToken EQUAL StreamID LBRKT streamParm *(COMMA
streamParm) RBRKT

localControlDescriptor = LocalControlToken LBRKT localParm *(COMMA localParm)
RBRKT

; не более одного на каждый элемент, кроме propertyParm
localParm          = (streamMode / propertyParm / reservedValueMode /
reservedGroupMode)

reservedValueMode  = ReservedValueToken EQUAL ("ON" / "OFF")

reservedGroupMode  = ReservedGroupToken EQUAL ("ON" / "OFF")

streamMode         = ModeToken EQUAL streamModes

streamModes        = (SendonlyToken / RecvonlyToken / SendrecvToken /
InactiveToken / LoopbackToken)

propertyParm       = pkgdName parmValue

; (Safe)Char '$' обозначает CHOOSE
; (Safe)Char '*' обозначает ALL
parmValue          = (EQUAL alternativeValue / INEQUAL VALUE)

alternativeValue    = (VALUE
/ LSBRKT VALUE *(COMMA VALUE) RSBRKT
; sublist (i.e., A AND B AND...)
/ LBRKT VALUE *(COMMA VALUE) RBRKT
; alternatives (i.e., A OR B OR...)
/ LSBRKT VALUE COLON VALUE RSBRKT)
; range

INEQUAL           = LWSP(">" / "<" / "#") LWSP; '#' means "not equal"

```

```

LSBRKT          = LWSP "[" LWSP
RSBRKT          = LWSP "]" LWSP

; ПРИМЕЧАНИЕ - Нулевой байт не является разрешенным символом в байтовой строке
; Поскольку текущее определение ограничивается SDP, а нулевой байт не будет
; разрешенным символом в SDP, об этом не следует беспокоиться.
localDescriptor = LocalToken LBRKT octetString RBRKT

remoteDescriptor = RemoteToken LBRKT octetString RBRKT

eventBufferDescriptor = EventBufferToken [LBRKT eventSpec*(COMMA eventSpec)
RBRKT]

eventSpec       = pkgdName [LBRKT eventSpecParameter *(COMMA
eventSpecParameter) RBRKT]

eventSpecParameter = (eventStream / eventOther)

eventBufferControl = BufferToken EQUAL eventBufferControlValue
eventBufferControlValue = ("OFF" / LockStepToken)

terminationStateDescriptor
= TerminationStateToken LBRKT terminationStateParm
*(COMMA terminationStateParm) RBRKT

; не более одного на каждый элемент, кроме propertyParm
terminationStateParm = (propertyParm / serviceStates / eventBufferControl)

serviceStates = ServiceStatesToken EQUAL serviceStatesValue
serviceStatesValue = (TestToken / OutOfSvcToken / InSvcToken)

muxDescriptor = MuxToken EQUAL MuxType terminationIDList

MuxType = (H221Token / H223Token / H226Token / V76Token /
extensionParameter / Nx64kToken)

StreamID = UINT16

pkgdName = (PackageName SLASH ItemID);specific item
/ (PackageName SLASH "*");all items in package
/ ("*" SLASH "*"); all items supported by the MG

PackageName = NAME

ItemID = NAME

eventsDescriptor = EventsToken [EQUAL RequestID LBRKT requestedEvent
*(COMMA requestedEvent) RBRKT]

requestedEvent = pkgdName [LBRKT eventParameter *(COMMA
eventParameter) RBRKT]

notifyRegulated = NotifyRegulatedToken [LBRKT (embedWithSig/
embedNoSig) RBRKT]

notifyBehaviour = NotifyImmediateToken / notifyRegulated /
NeverNotifyToken

; не более одного KeepActiveToken, notifyBehaviour, eventDM,
; ResetEventsDescriptor и eventStream
; самое большее один embedWithSig либо embedNoSig, но не оба одновременно
; KeepActiveToken и embedWithSig не должны присутствовать одновременно
eventParameter = (embedWithSig / embedNoSig / KeepActiveToken /

```

```

eventDM / eventStream / eventOther /
notifyBehaviour / ResetEventsDescriptorToken)

embedWithSig          = EmbedToken LBRKT signalsDescriptor [COMMA
embedFirst] RBRKT

embedNoSig            = EmbedToken LBRKT embedFirst RBRKT

; не более одного каждого
embedFirst           = EventsToken [EQUAL RequestID LBRKT
secondRequestedEvent *(COMMA secondRequestedEvent)
RBRKT]

secondRequestedEvent  = pkgdName [LBRKT secondEventParameter *(COMMA
secondEventParameter) RBRKT]

; не более одного embedSig, KeepActiveToken, notifyBehaviour, eventDM
; ResetEventsDescriptor или eventStream
; KeepActiveToken и embedSig не должны присутствовать одновременно
secondEventParameter = (embedSig / KeepActiveToken / eventDM /
eventStream / eventOther / notifyBehaviour /
ResetEventsDescriptorToken)

embedSig              = EmbedToken LBRKT signalsDescriptor RBRKT

eventStream           = StreamToken EQUAL StreamID

eventOther            = eventParameterName parmValue

eventParameterName   = NAME

eventDM               = DigitMapToken EQUAL((digitMapName) / (LBRKT
digitMapValue RBRKT))

signalsDescriptor     = SignalsToken [LBRKT signalParm *(COMMA signalParm)
RBRKT]

signalParm            = signalList / signalRequest

signalRequest         = signalName [LBRKT sigParameter *(COMMA sigParameter)
RBRKT]

signalList            = SignalListToken EQUAL signalListId LBRKT
signalListParm *(COMMA signalListParm) RBRKT

signalListId         = UINT16

; точно одно появление signalType, не более одной продолжительности и каждого
параметра сигнала
signalListParm       = signalRequest

signalName            = pkgdName

; не более одного появления sigStream, не более одного появления sigSignalType,
; не более одного появления sigDuration, не более одного появления sigDirection,
; не более одного появления sigRequestID, не более одного появления
sigIntsigDelay
; не более одного появления каждого signalParameterName
sigParameter         = sigStream / sigSignalType / sigDuration / sigOther /
notifyCompletion / KeepActiveToken /
sigDirection / sigRequestID / sigIntsigDelay

sigStream             = StreamToken EQUAL StreamID

sigOther              = sigParameterName parmValue

```



```

sigParameterName      = NAME

sigSignalType         = SignalTypeToken EQUAL signalType

signalType            = (OnOffToken / TimeOutToken / BriefToken)

sigDuration           = DurationToken EQUAL UINT16

sigDirection          = DirectionToken EQUAL direction

direction             = ExternalToken / InternalToken / BothToken

sigRequestID          = RequestIDToken EQUAL RequestID

sigIntsigDelay        = IntsigDelayToken EQUAL UINT16

notifyCompletion      = NotifyCompletionToken EQUAL (LBRKT
notificationReason *(COMMA notificationReason)
RBRKT)

notificationReason    = TimeOutToken / InterruptByEventToken /
InterruptByNewSignalsDescrToken / OtherReasonToken /
IterationToken

observedEventsDescriptor
= ObservedEventsToken EQUAL RequestID LBRKT
observedEvent *(COMMA observedEvent) RBRKT

; время за событие, поскольку события можно заносить в буфер
observedEvent         = [TimeStamp LWSP COLON] LWSP pkgdName [LBRKT
observedEventParameter *(COMMA
observedEventParameter) RBRKT]

; не более одного появления eventStream, не более одного появления каждого
eventParameterName
observedEventParameter = eventStream / eventOther

; Для AuditCapReply со всеми событиями, RequestID должен быть ALL.
RequestID             = UINT32 / "*"

modemDescriptor       = ModemToken ((EQUAL modemType) / (LSBRKT modemType
*(COMMA modemType) RSBKRT)) [LBRKT propertyParm
*(COMMA propertyParm) RBRKT]

; не более одного появления, кроме, как для extensionParameter
modemType             = V32bisToken / V22bisToken / V18Token / V22Token /
V32Token / V34Token / V90Token / V91Token /
SynchISDNToken / extensionParameter

digitMapDescriptor    = DigitMapToken EQUAL ((LBRKT digitMapValue RBRKT) /
(digitMapName [LBRKT digitMapValue RBRKT]))

digitMapName          = NAME

digitMapValue         = ["T" COLON Timer COMMA] ["S" COLON Timer COMMA] ["L"
COLON Timer COMMA] ["Z" COLON Timer COMMA] digitMap

Timer                 = 1*2DIGIT
; Для таймеров T, S, и L единицей измерения являются секунды, для таймера
; Z - сотые доли миллисекунд. Таким образом диапазон для T, S, и L - от 1 до 99
; секунд, а для Z - от 100 мс до 9.9 с

digitMap              = (digitString / LWSP "(" LWSP digitStringList LWSP
")" LWSP)

```

```

digitStringList      = digitString *(LWSP "|" LWSP digitString)
digitString          = 1*(digitStringElement)
digitStringElement  = digitPosition [DOT]
digitPosition        = digitMapLetter / digitMapRange
digitMapRange        = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))
digitLetter          = *((DIGIT "-" DIGIT) / digitMapLetter)
digitMapLetter       = DIGIT                ;Basic event symbols
                    / %x41-4B / %x61-6B ; a-k, A-K
                    / "L" / "S" / "T"      ;Inter-event timers
                    ; (long, short, start)
                    / "Z"                  ;Long duration modifier

; не более одного появления, DigitMapToken и PackagesToken
; в команде AuditCapabilities не разрешены
auditItem            = auditReturnItem / SignalsToken / EventBufferToken /
                    EventsToken / indAudterminationAudit

indAudterminationAudit = indAudauditReturnParameter *(COMMA
                    indAudauditReturnParameter)

indAudauditReturnParameter
                    = indAudmediaDescriptor / indAudeventsDescriptor /
                    indAudsignalsDescriptor /
                    indAuddigitMapDescriptor /
                    indAudeventBufferDescriptor /
                    indAudstatisticsDescriptor /
                    indAudpackagesDescriptor

indAudmediaDescriptor = MediaToken LBRKT indAudmediaParm *(COMMA
                    indAudmediaParm) RBRKT

; либо streamParm, либо streamDescriptor, но не оба одновременно
indAudmediaParm      = indAudstreamParm / indAudstreamDescriptor /
                    indAudterminationStateDescriptor

; не более одного появления
indAudstreamParm     = (indAudlocalControlDescriptor /
                    indAudstatisticsDescriptor /
                    indAudremoteDescriptor / indAudlocalDescriptor )

indAudremoteDescriptor = RemoteToken LBRKT octetString RBRKT
indAudlocalDescriptor  = LocalToken LBRKT octetString RBRKT

indAudstreamDescriptor = StreamToken EQUAL StreamID LBRKT indAudstreamParm
                    RBRKT

indAudlocalControlDescriptor
                    = LocalControlToken LBRKT indAudlocalParm
                    *(COMMA indAudlocalParm) RBRKT

; не более одного появления для каждого элемента
indAudlocalParm      = ModeToken [(EQUAL/INEQUAL) streamModes]/ pkgdName /
                    propertyParm / ReservedValueToken /
                    ReservedGroupToken

; propertyParm и streamModes используются только для определения критериев
; выбора при проверке
; логика выбора AND/OR определяется на уровне контекста.

```

```

indAudterminationStateDescriptor
    = TerminationStateToken LBRKT
      indAudterminationStateParm RBRKT

; не более одного появления для каждого элемента
indAudterminationStateParm
    = pkgdName / propertyParm / ServiceStatesToken
      [(EQUAL/INEQUAL) serviceStatesValue ] / BufferToken

; При включении значений применяется операция Select.
; логика AND/OR определяется на уровне контекста.

indAudeventBufferDescriptor
    = EventBufferToken LBRKT indAudeventSpec RBRKT

indAudeventSpec
    = pkgdName [LBRKT indAudeventSpecParameter RBRKT]

indAudeventSpecParameter= eventStream / eventParameterName

indAudeventsDescriptor   = EventsToken [EQUAL RequestID] LBRKT
                          indAudrequestedEvent RBRKT

indAudrequestedEvent     = pkgdName

indAudsignalsDescriptor  = SignalsToken LBRKT [indAudsignalParm] RBRKT

indAudsignalParm         = indAudsignalList / indAudsignalRequest

indAudsignalRequest      = signalName [LBRKT indAudsignalRequestParm
    *(COMMA indAudsignalRequestParm) RBRKT]

indAudsignalRequestParm = sigStream / sigRequestID

indAudsignalList         = SignalListToken EQUAL signalListId [LBRKT
    indAudsignalListParm RBRKT]

indAudsignalListParm     = indAudsignalRequest

indAuddigitMapDescriptor= DigitMapToken EQUAL (digitMapName)

indAudstatisticsDescriptor
    = StatsToken LBRKT pkgdName RBRKT

indAudpackagesDescriptor= PackagesToken LBRKT packagesItem RBRKT

serviceChangeDescriptor  = ServicesToken LBRKT serviceChangeParm *(COMMA
    serviceChangeParm) RBRKT

; не более одного появления для каждого параметра, кроме auditItem
; самое большее один serviceChangeAddress, либо serviceChangeMgcId,
; но не оба одновременно
; serviceChangeMethod и serviceChangeReason являются REQUIRED
serviceChangeParm        = serviceChangeMethod / serviceChangeReason /
    serviceChangeDelay / serviceChangeAddress /
    serviceChangeProfile / extension / TimeStamp /
    serviceChangeMgcId / serviceChangeVersion /
    ServiceChangeIncompleteToken / auditItem

serviceChangeReplyDescriptor
    = ServicesToken LBRKT servChgReplyParm *(COMMA
    servChgReplyParm) RBRKT

; не более одного появления. Версия является REQUIRED на первый ответ
ServiceChange
; самое большее один serviceChangeAddress, либо serviceChangeMgcId,
; но не оба одновременно

```

```

servChgReplyParm      = serviceChangeAddress / serviceChangeMgcId /
                       serviceChangeProfile / serviceChangeVersion /
                       TimeStamp

serviceChangeMethod   = MethodToken EQUAL (FailoverToken / ForcedToken /
                       GracefulToken / RestartToken / DisconnectedToken /
                       HandOffToken / extensionParameter)

; serviceChangeReason состоит из цифрового кода причины
; и дополнительного текстового описания.
; serviceChangeReason должен быть закодирован с использованием формы
; quotedString VALUE.
; quotedString должен содержать десятичный код причины,
; за ним дополнительно следует один символ пробела
; и строка текстового описания.
serviceChangeReason   = ReasonToken EQUAL VALUE

serviceChangeDelay    = DelayToken EQUAL UINT32

serviceChangeAddress  = ServiceChangeAddressToken EQUAL (mId / portNumber)

serviceChangeMgcId    = MgcIdToken EQUAL mId

serviceChangeProfile  = ProfileToken EQUAL NAME SLASH Version

serviceChangeVersion  = VersionToken EQUAL Version

extension              = extensionParameter parmValue

packagesDescriptor    = PackagesToken LBRKT packagesItem *(COMMA
                       packagesItem) RBRKT

Version                = 1*2(DIGIT)

packagesItem           = NAME "-" UINT16

TimeStamp              = Date "T" Time; per ISO 8601:2004

; Дата = yyyymmdd
Date                   = 8(DIGIT)

; Время = hhmmssss
Time                   = 8(DIGIT)

statisticsDescriptor  = StatsToken LBRKT statisticsParameter *(COMMA
                       statisticsParameter) RBRKT

;не более одного появления на каждый элемент
statisticsParameter   = pkgdName [EQUAL VALUE /
                       (LSBRKT VALUE *(COMMA VALUE) RSBKRT)]

topologyDescriptor    = TopologyToken LBRKT topologyTriple *(COMMA
                       topologyTriple) RBRKT

topologyTriple        = terminationA COMMA terminationB COMMA
                       topologyDirection [COMMA eventStream]

terminationA          = TerminationID

terminationB          = TerminationID

topologyDirection     = BothwayToken / IsolateToken / OnewayToken /
                       OnewayExternalToken / OnewayBothToken

```

```

priority                = PriorityToken EQUAL UINT16

iepsValue               = IEPSToken EQUAL ("ON" / "OFF")

emergencyValue          = EmergencyValueToken EQUAL (EmergencyToken /
EmergencyOffToken)

contextAttrDescriptor   = ContextAttrToken LBRKT (contextIdList /
propertyParm *(COMMA propertyParm)) RBRKT

; При использовании структурного элемента contextIdList, the ContextID в
; структурном элементе actionReply должен быть таким же, как и ContextID в
; соответствующем actionRequest
contextIdList           = ContextListToken EQUAL LBRKT ContextID
                        *(COMMA ContextID) RBRKT

extensionParameter      = "X" ("- " / "+") 1*6(ALPHA / DIGIT)

; octetString используется для описания SDP, определенного в RFC 2327.
; Если используется CRLF в RFC 2327, следует проявлять осторожность.
; Для надежности в данном ABNF используйте EOL.
; Везде, где в SDP появляется символ "}", перед ним ставится "\", например, "\\}"
octetString              = *(nonEscapeChar)

nonEscapeChar           = ("\"}" / %x01-7C / %x7E-FF)

; ПРИМЕЧАНИЕ - Символ двойных кавычек в quotedString не разрешается.
quotedString             = DQUOTE *(SafeChar / EOL / %x80-FF / RestChar / WSP)
                        DQUOTE

UINT16                  = 1*5(DIGIT); %x0-FFFF

UINT32                  = 1*10(DIGIT); %x0-FFFFFFFF

NAME                    = ALPHA *63(ALPHA / DIGIT / "_")

VALUE                   = quotedString / 1*(SafeChar / %x80-FF)

SafeChar                = DIGIT / ALPHA / "+" / "-" / "&" /
"! " / " " / "/" / " " / "?" / "@" /
"^" / "~" / "~" / "*" / "$" / "\" /
"(" / ")" / "%" / "|" / "."

EQUAL                   = LWSP %x3D LWSP          ; "="

COLON                   = %x3A                    ; ":"

LBRKT                   = LWSP %x7B LWSP          ; "{"

RBRKT                   = LWSP %x7D LWSP          ; "}"

COMMA                   = LWSP %x2C LWSP          ; ","

DOT                     = %x2E                    ; "."

SLASH                   = %x2F                    ; "/"

ALPHA                   = %x41-5A / %x61-7A        ; A-Z / a-z

DIGIT                   = %x30-39                 ; 0-9

DQUOTE                  = %x22                    ; " (Двойные кавычки)

HEXDIG                  = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"

```

SP = %x20 ; пробел
 HTAB = %x09 ; символ горизонтальной табуляции
 CR = %x0D ; символ возврата каретки
 LF = %x0A ; символ перехода на новую строку
 LWSP = *(WSP / COMMENT / EOL)
 EOL = (CR [LF] / LF)
 WSP = SP / HTAB ; пробельный символ
 SEP = (WSP / EOL / COMMENT) LWSP
 COMMENT = ";" *(SafeChar/ RestChar / WSP / %x22) EOL
 RestChar = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" / "<" / ">" / "="

; Новые метки, добавляемые к sigParameter должны принять форму SPA*
 ; * может быть любой формы, т. е. SPAM
 ; Новые метки, добавляемые к eventParameter должны принять форму EPA*
 ; * может быть любой формы, т. е. EPAD

AddToken = ("Add" / "A")
 AndAUDITSelectToken = ("ANDLgc")
 AuditToken = ("Audit" / "AT")
 AuditCapToken = ("AuditCapability" / "AC")
 AuditValueToken = ("AuditValue" / "AV")
 AuthToken = ("Authentication" / "AU")
 BothToken = ("Both" / "B")
 BothwayToken = ("Bothway" / "BW")
 BriefToken = ("Brief" / "BR")
 BufferToken = ("Buffer" / "BF")
 CtxToken = ("Context" / "C")
 ContextAuditToken = ("ContextAudit" / "CA")
 ContextAttrToken = ("ContextAttr" / "CT")
 ContextListToken = ("ContextList" / "CLT")
 DigitMapToken = ("DigitMap" / "DM")
 DirectionToken = ("SPADirection" / "SPADI")
 DisconnectedToken = ("Disconnected" / "DC")
 DelayToken = ("Delay" / "DL")
 DurationToken = ("Duration" / "DR")
 EmbedToken = ("Embed" / "EM")
 EmergencyToken = ("Emergency" / "EG")
 EmergencyOffToken = ("EmergencyOff" / "EGO")
 EmergencyValueToken = ("EmergencyValue" / "EGV")
 ErrorToken = ("Error" / "ER")
 EventBufferToken = ("EventBuffer" / "EB")
 EventsToken = ("Events" / "E")
 ExternalToken = ("External" / "EX")
 FailoverToken = ("Failover" / "FL")
 ForcedToken = ("Forced" / "FO")
 GracefulToken = ("Graceful" / "GR")
 H221Token = ("H221")
 H223Token = ("H223")
 H226Token = ("H226")
 HandOffToken = ("HandOff" / "HO")
 IEPSToken = ("IEPSCall" / "IEPS")
 ImmAckRequiredToken = ("ImmAckRequired" / "IA")
 InactiveToken = ("Inactive" / "IN")

```

InternalToken           = ("Internal"           / "IT")
IntsigDelayToken       = ("Intersignal"       / "SPAIS")
IsolateToken           = ("Isolate"           / "IS")
InSvcToken             = ("InService"         / "IV")
InterruptByEventToken  = ("IntByEvent"        / "IBE")
InterruptByNewSignalsDescrToken
= ("IntBySigDescr"     / "IBS")
IterationToken         = ("Iteration"         / "IR")
KeepActiveToken       = ("KeepActive"       / "KA")
LocalToken            = ("Local"            / "L")
LocalControlToken     = ("LocalControl"     / "O")
LockStepToken         = ("LockStep"         / "SP")
LoopbackToken         = ("Loopback"         / "LB")
MediaToken            = ("Media"            / "M")
MegacopToken          = ("MEGACO"          / "!")
MessageSegmentToken   = ("Segment"         / "SM")
MethodToken           = ("Method"           / "MT")
MgcIdToken            = ("MgcIdToTry"      / "MG")
ModeToken             = ("Mode"            / "MO")
ModifyToken           = ("Modify"          / "MF")
ModemToken            = ("Modem"           / "MD")
MoveToken             = ("Move"            / "MV")
MTPToken              = ("MTP")
MuxToken              = ("Mux"             / "MX")
NeverNotifyToken      = ("NeverNotify"     / "NBNN")
NotifyToken           = ("Notify"          / "N")
NotifyCompletionToken = ("NotifyCompletion" / "NC")
NotifyImmediateToken  = ("ImmediateNotify" / "NBIN")
NotifyRegulatedToken  = ("RegulatedNotify" / "NBRN")
Nx64kToken           = ("Nx64Kservice"    / "N64")
ObservedEventsToken   = ("ObservedEvents"   / "OE")
OnewayToken           = ("Oneway"          / "OW")
OnewayBothToken       = ("OnewayBoth"      / "OWB")
OnewayExternalToken   = ("OnewayExternal"  / "OWE")
OnOffToken            = ("OnOff"           / "OO")
OrAUDITselectToken   = ("ORLgc")
OtherReasonToken      = ("OtherReason"     / "OR")
OutOfSvcToken         = ("OutOfService"    / "OS")
PackagesToken         = ("Packages"        / "PG")
PendingToken          = ("Pending"         / "PN")
PriorityToken         = ("Priority"         / "PR")
ProfileToken          = ("Profile"         / "PF")
ReasonToken           = ("Reason"          / "RE")
RecvonlyToken         = ("ReceiveOnly"     / "RC")
ReplyToken            = ("Reply"           / "P")
ResetEventsDescriptorToken
= ("ResetEventsDescriptor" / "RSE")
RestartToken          = ("Restart"         / "RS")
RemoteToken           = ("Remote"          / "R")
RequestIDToken        = ("SPARrequestID"   / "SPARQ")
ReservedGroupToken    = ("ReservedGroup"   / "RG")
ReservedValueToken    = ("ReservedValue"   / "RV")
SegmentationCompleteToken
= ("END"              / "&")
SendonlyToken         = ("SendOnly"        / "SO")
SendrecvToken         = ("SendReceive"     / "SR")
ServicesToken         = ("Services"        / "SV")
ServiceStatesToken    = ("ServiceStates"   / "SI")
ServiceChangeIncompleteToken
= ("ServiceChangeInc"  / "SIC")
ServiceChangeToken    = ("ServiceChange"   / "SC")
ServiceChangeAddressToken
= ("ServiceChangeAddress" / "AD")
SignalListToken       = ("SignalList"      / "SL")

```

```

SignalsToken      = ("Signals"           / "SG")
SignalTypeToken   = ("SignalType"        / "SY")
StatsToken        = ("Statistics"       / "SA")
StreamToken       = ("Stream"           / "ST")
SubtractToken     = ("Subtract"         / "S")
SynchISDNToken   = ("SynchISDN"        / "SN")
TerminationStateToken = ("TerminationState" / "TS")
TestToken         = ("Test"             / "TE")
TimeOutToken      = ("TimeOut"          / "TO")
TopologyToken     = ("Topology"         / "TP")
TransToken        = ("Transaction"      / "T")
ResponseAckToken  = ("TransactionResponseAck" / "K")
V18Token          = ("V18")
V22Token          = ("V22")
V22bisToken       = ("V22b")
V32Token          = ("V32")
V32bisToken       = ("V32b")
V34Token          = ("V34")
V76Token          = ("V76")
V90Token          = ("V90")
V91Token          = ("V91")
VersionToken      = ("Version"          / "V")

```

В.3 Шестнадцатеричное кодирование байтов

Шестнадцатеричное кодирование байтов является средством для представления строки байтов в виде строки шестнадцатеричных чисел, причем каждый байт представлен двумя цифрами. Данное кодирование байтов следует использовать при кодировании байтовых строк в текстовой версии данного протокола.

Для каждого байта 8-битовая последовательность кодируется как два шестнадцатеричных числа. Первым передается бит 0; последним – бит 7.

Биты 7-4 кодируются как первое шестнадцатеричное число, причем Бит 7 кодируется как MSB, а Бит 4 – как LSB. Биты 3-0 кодируются как второе шестнадцатеричное число, причем Бит 3 кодируется как MSB, а Бит 0 – как LSB.

Примеры:

Байтовый образец бита	Шестнадцатеричное кодирование
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

В.4 Шестнадцатеричная последовательность байтов

Шестнадцатеричная последовательность байтов представляет собой четное число шестнадцатеричных чисел, завершающееся символом <CR>.

Приложение С

Тэги для свойств потока медиа

Если для данного протокола используется двоичное кодирование, параметры для дескрипторов Local, Remote и LocalControl задаются как пары тэг – значений. В данном Приложении содержатся названия свойств (PropertyID), тэги (Property tag), тип свойства (Type) и значения (Value). Если поле Value содержит ссылки, значения, представленные в нем, нужно считать "информацией". В ссылке содержатся нормативные значения. Если поле Value не содержит ссылки, тогда значения в данном поле можно считать "нормативными".

Структура ссылок на свойства Приложения С соответствует структуре PackageID/PropertyID; однако, Приложение С само по себе не является комплектом. Считается, что в Приложении С имеется PackageID 0x0000 для двоичного кодирования и "анхс" для текстового кодирования. Для текстового кодирования H.248.1, нужно использовать Приложение С, только если требуемое свойство либо не определяется комплектом, либо не представлено SDP. Вложение одного свойства Приложения С внутрь другого запрещено.

В данном приложении тэги даются в виде шестнадцатеричных чисел. При установке значения свойства, MGC может недоопределить данное значение в соответствии с одним из алгоритмов, описанных в п. 7.1.1.

Поддержка свойств в данном приложении или в любом из его подпунктов является дополнительной. Например, могут быть применены только три свойства из С.3 и только пять свойств из С.8.

Для типа "перечисление" значение представлено значением в скобках, например, Send (0), Receive (1). При кодировании протокола свойства Приложения С с типами "N bits" или "M Octets" следует обрабатывать как строки байтов. Свойства с "N bit integer" нужно обрабатывать как целые числа. При кодировании данного протокола "String" нужно обрабатывать как IA5String.

Когда тип меньше одного байта, данное значение следует сохранять в битах низшего порядка строки байта размером в один байт.

С.1 Общие атрибуты медиа

PropertyID	Property tag	Тип	Значение
Media	1001	Перечисление	Audio(0), Video(1), Data(2)
Transmission mode	1002	Перечисление	Send(0), Receive(1), Send&Receive(2)
Number of Channels	1003	Целое число без знака	0–255
Sampling rate	1004	Целое число без знака	$0-2^{32}$
Bitrate	1005	Целое число	(0..4294967295) ПРИМЕЧАНИЕ. – Единица измерения 100 бит/с.
ACodec	1006	Байтовая строка	Тип аудио кодека: См. Рек. МСЭ-Т Q.765.5 Кодеки, не относящиеся к МСЭ-Т, определяются соответствующими организациями по стандартизации при помощи определенных организационных идентификаторов.
Samplepp	1007	Целое число без знака	Максимальное число замеров или кадров на пакет: 0..65535
Silencesupp	1008	Логическое выражение	Подавление пауз: Верно (True)/Неверно (False)

PropertyID	Property tag	Тип	Значение
Encrypttype	1009	Байтовая строка	См. Рек. МСЭ-Т Н.245
Encryptkey	100A	Размер байтовой строки (0..65535)	Ключ шифрования См. Рек. МСЭ-Т Н.235.0
Echocanc	100B		Не используется. Пример возможных свойств Echo Control см. в Е.13.
Gain	100C		Не используется. Доступное свойство усиления см. в Е.13.
Jitterbuff	100D	Целое число без знака	Размер буфера дрожания в мс: 0..65535
PropDelay	100E	Целое число без знака	Задержка распространения: 0..65535 Максимальная задержка распространения в миллисекундах для основного соединения между двумя медиашлюзами. Величина максимальной задержки будет зависеть от технологии транспортного канала.
RTPpayload	100F	Целое число	Тип полезной нагрузки в профиле RTP Profile для аудио- и видеоконференций с минимальным управлением См.: RFC 1890
Ptime	1010	Целое число	Время пакетизации Дает количество времени в миллисекундах, представленное медиа в пакете. См.: IETF RFC 2327

С.2 Свойства мультиплексора

PropertyID	Property tag	Тип	Значение
H222	2001	Байтовая строка	H222LogicalChannelParameters См.: Рек. МСЭ-Т Н.245
H223	2002	Байтовая строка	H223LogicalChannelParameters См.: Рек. МСЭ-Т Н.245
V76	2003	Байтовая строка	V76LogicalChannelParameters См.: Рек. МСЭ-Т Н.245
H2250	2004	Байтовая строка	H2250LogicalChannelParameters См.: Рек. МСЭ-Т Н.245

С.3 Общие свойства транспортного канала

PropertyID	Property tag	Тип	Значение
Mediatx	3001	Перечисление	Тип передачи медиа TDM Circuit(0), ATM(1), FR(2), Ipv4(3), Ipv6(4), ...
BIR	3002	4 байта	Значение зависит от технологии передачи
NSAP	3003	1-20 байтов	См. NSAP. См.: Приложение А/Х.213

С.4 Общие свойства АТМ

PropertyID	Property tag	Тип	Значение
AESA	4001	20 байтов	Адрес конечной системы АТМ
VPVC	4002	4 байта: VPCI в первых двух самых младших байтах, VCI во вторых двух байтах	VPCI/VCI См.: Рек. МСЭ-Т Q.2931
SC	4003	Перечисление	Категория обслуживания: CBR(0), nrt-VBR1(1), nrt-VBR2(2), nrt-VBR3(3), rt-VBR1(4), rt-VBR2(5), rt-VBR3(6), UBR1(7), UBR2(8), ABR(9). См.: Форум АТМ UNI 4.0
BCOB	4004	5-битное целое число	Класс широкополосного транспортного канала (Broadband Bearer Class) См.: Рек. МСЭ-Т Q.2961.2
BBTC	4005	7-битное целое число	Возможность широкополосной передачи (Broadband Transfer Capability) См.: Рек. МСЭ-Т Q.2961.1
ATC	4006	Перечисление	Возможность трафика I.371 АТМ (ATM Traffic Capability) DBR(0), SBR1(1), SBR2(2), SBR3(3), АВТ/IT(4), АВТ/DT(5), ABR(6) См.: Рек. МСЭ-Т I.371
STC	4007	2 бита	Чувствительность к ограничению (Susceptibility to clipping): Биты <u>2 1</u> 0 0 не чувствителен к ограничению 0 1 чувствителен к ограничению См.: Рек. МСЭ-Т Q.2931
UPCC	4008	2 бита	Конфигурация соединения плоскости пользователя (User Plane Connection configuration): Биты <u>2 1</u> 0 0 точка – точка 0 1 точка – много точек См.: Рек. МСЭ-Т Q.2931
PCR0	4009	24-битное целое число	Пиковая скорость передачи (Peak Cell Rate) (Для CLP = 0) См.: Рек. МСЭ-Т Q.2931
SCR0	400A	24-битное целое число	Нормальная скорость передачи (Sustainable Cell Rate) (Для CLP = 0) См.: Рек. МСЭ-Т Q.2961.1
MBS0	400B	24-битное целое число	Максимальный размер пакета (Maximum Burst Size) (Для CLP = 0) См.: Рек. МСЭ-Т Q.2961.1
PCR1	400C	24-битное целое число	Пиковая скорость передачи (Peak Cell Rate) (Для CLP = 0 + 1) См.: Рек. МСЭ-Т Q.2931
SCR1	400D	24-битное целое число	Нормальная скорость передачи (Sustainable Cell Rate) (Для CLP = 0 + 1)

PropertyID	Property tag	Тип	Значение	
			См.: Рек. МСЭ-Т Q.2961.1	
MBS1	400E	24-битное целое число	Максимальный размер пакета (Maximum Burst Size) (Для CLP = 0 + 1) См.: Рек. МСЭ-Т Q.2961.1	
BEI	400F	Логическое выражение	Индикатор негарантированной доставки (Best Effort Indicator) Значение 1 показывает, что BEI нужно включить в сигнализацию ATM; значение 0 показывает, что BEI не нужно включать в сигнализацию ATM. См.: Форум ATM UNI 4.0	
TI	4010	Логическое выражение	Индикатор тэгирования (Tagging Indicator) Значение 0 показывает, что тэгирование не разрешается; значение 1 показывает, что тэгирование запрашивается. См.: Рек. МСЭ-Т Q.2961.1	
FD	4011	Логическое выражение	Сброс кадра (Frame Discard) Значение 0 показывает, что сброс кадра не разрешается; значение 1 показывает, что сброс кадра разрешается. См.: Форум ATM UNI 4.0	
A2PCDV	4012	24-битное целое число	Допустимая 2-точечная CDV См.: Рек. МСЭ-Т Q.2965.2	
C2PCDV	4013	24-битное целое число	Совокупная 2-точечная CDV См.: Рек. МСЭ-Т Q.2965.2	
APPCDV	4014	24-битное целое число	Допустимая P-P CDV См.: Форум ATM UNI 4.0	
CPPCDV	4015	24-битное целое число	Совокупная P-P CDV См.: Форум ATM UNI 4.0	
ACLR	4016	8-битное целое число	Допустимый процент потери ячеек (Acceptable Cell Loss Ratio) См.: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
MEETD	4017	16-битное целое число	Максимальная задержка при сквозной передаче (Maximum End-to-end transit delay) См.: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
CEETD	4018	16-битное целое число	Совокупная задержка при сквозной передаче (Cumulative End-to-end transit delay) См.: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
QoSClass	4019	Целое число 0-5	Класс QoS	
			Класс QoS	Значение
			0	Заданное по умолчанию QoS, связанное с АТС, как определено в Рек. МСЭ-Т Q.2961.2
			1	Строгий
			2	Терпимый
			3	Двухуровневый
			4	Неограниченный
			5	Строгий двухуровневый
			См.: Рек. МСЭ-Т Q.2965.1	

PropertyID	Property tag	Тип	Значение
AALtype	401A	1 байт	Тип AAL Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 AAL для голоса 0 0 0 0 0 0 0 1 AAL тип 1 0 0 0 0 0 0 1 0 AAL тип 2 0 0 0 0 0 0 1 1 AAL тип 3/4 0 0 0 0 0 1 0 1 AAL тип 5 0 0 0 1 0 0 0 0 AAL, определяемый пользователем См.: Рек. МСЭ-Т Q.2931

C.5 Ретрансляция кадров

PropertyID	Property tag	Тип	Значение
DLCI	5001	Целое число без знака	ID соединения канала передачи (Data link connection ID)
CID	5002	Целое число без знака	ID подканала (sub-channel ID)
SID/Noiselevel	5003	Целое число без знака	Дескриптор внесения пауз (silence insertion descriptor)
Primary Payload type	5004	Целое число без знака	Основной тип полезной нагрузки (Primary Payload Type) Охватывает FAX и кодеки

C.6 Протокол IP

PropertyID	Property tag	Тип	Значение
IPv4	6001	32 бита Ipv4Address	IPv4Address См.: IETF RFC 791
IPv6	6002	128 битов	IPv6 Address См.: IETF RFC 2460
Port	6003	Целое число без знака	0..65535
Porttype	6004	Перечисление	TCP(0), UDP(1), SCTP(2)
RtcpbwRS	6005	Целое число	Модификатор пропускной способности RS RTCP показывает пропускная способность RTCP размещена по активным отправителям данных (как определено в спецификации RTP) См.: IETF RFC 3556
RtcpbwRR	6006	Целое число	Модификатор пропускной способности RR RTCP показывает пропускную способность TCP, размещенную по остальным участникам в сеансе RTP (т. е. получателями) См.: IETF RFC 3556

C.7 AAL 2 ATM

PropertyID	Property tag	Тип	Значение
AESA	7001	20 байтов	Адрес конечной точки службы AAL 2 как определено в Рекомендации, на которую делается ссылка. ESEA NSEA См.: Рек. МСЭ-Т Q.2630.1
BIR	См. С.3	4 байта	Ссылка, генерируемая обслуживаемым пользователем, как определено в Рекомендации, на которую делается ссылка. SUGR См.: Рек. МСЭ-Т Q.2630.1
ALC	7002	12 байтов	Характеристики звена AAL 2, как определено в Рекомендации, на которую делается ссылка. Максимальная/Средняя CPS-SDU битовая скорость; Максимальный/ Средний CPS-SDU размер См.: Рек. МСЭ-Т Q.2630.1
SSCS	7003	I.366.2: Аудио (8 байтов); Многоскоростной (3 байта), или I.366.1: Обеспеченный SAR Необеспеченный SAR (7 байтов).	Информация служебно-ориентированного подуровня, определенная в: – Рек. МСЭ-Т. Q.2630.1, и используемая в: – Рек. МСЭ-Т I.366.2: Аудио/ Многоскоростной; – Рек. МСЭ-Т I.366.1: Обеспеченный SAR/- необеспеченный. См.: Рек. МСЭ-Т Q.2630.1, I.366.1 и I.366.2
SUT	7004	1..254 байтов	Параметр передачи обслуживаемого пользователя, как определено в Рекомендации, на которую делается ссылка. См.: Рек. МСЭ-Т Q.2630.1
TCI	7005	Логическое выражение	Индикатор тестового соединения, как определено в Рекомендации, на которую делается ссылка. См.: Рек. МСЭ-Т Q.2630.1
Timer_CU	7006	32-битное целое число	Таймер CU (Timer-CU) Количество миллисекунд для удержания частично заполненной ячейки перед отправкой.
MaxCPSSDU	7007	8-битное целое число	Максимальная единица данных службы подуровня общей части (Maximum Common Part Sublayer Service Data Unit) См.: Рек. МСЭ-Т Q.2630.1
CID	7008	8 битов	ID подканала: 0–255 См. Рек. МСЭ-Т I.363.2

C.8 AAL 1 ATM

PropertyID	Property tag	Тип	Значение
BIR	См. таблицу в С.3	4–29 байтов	GIT (Generic Identifier Transport) – Передача общего идентификатора См.: Рек. МСЭ-Т Q.2941.1
AAL1ST	8001	1 байт	Подтип AAL 1 Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 ноль 0 0 0 0 0 0 0 1 передача сигнала частот телефонной связи на 64 кбит/с 0 0 0 0 0 0 1 0 передача канала 0 0 0 0 0 1 0 0 передача высококачественного аудиосигнала 0 0 0 0 0 1 0 1 передача видеосигнала См.: Рек. МСЭ-Т Q.2931
CBRR	8002	1 байт	Скорость CBR Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 1 64 кбит/с 0 0 0 0 0 1 0 0 1544 кбит/с 0 0 0 0 0 1 0 1 6312 кбит/с 0 0 0 0 0 1 1 0 32 064 кбит/с 0 0 0 0 0 1 1 1 44 736 кбит/с 0 0 0 0 1 0 0 0 97 728 кбит/с 0 0 0 1 0 0 0 0 2048 кбит/с 0 0 0 1 0 0 0 1 8448 кбит/с 0 0 0 1 0 0 1 0 34 368 кбит/с 0 0 0 1 0 0 1 1 139 264 кбит/с 0 1 0 0 0 0 0 0 $n \times 64$ кбит/с 0 1 0 0 0 0 0 1 $n \times 8$ кбит/с См.: Рек. МСЭ-Т Q.2931
MULT	См. таблицу в С.9		Множитель, или $n \times 64k/8k/300$ См.: Рек. МСЭ-Т Q.2931
SCRI	8003	1 байт	Метод восстановления тактовой частоты источника (Source Clock Frequency Recovery Method) Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 ноль 0 0 0 0 0 0 0 1 SRTS 0 0 0 0 0 0 1 0 ACM См.: Рек. МСЭ-Т Q.2931
ECM	8004	1 байт	Метод исправления ошибок (Error Correction Method) Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 ноль 0 0 0 0 0 0 0 1 FEC – Потеря (Loss) 0 0 0 0 0 0 1 0 FEC – Задержка (Delay) См.: Рек. МСЭ-Т Q.2931
SDTB	8005	16-битное целое число	Размер блока передачи структурированных данных (Structured Data Transfer Blocksize) Размер блока службы SDT CBR

PropertyID	Property tag	Тип	Значение
			См.: Рек. МСЭ-Т I.363.1
PFCI	8006	8-битное целое число	Идентификатор частично заполненных ячеек (Partially filled cells identifier) 1–47 См.: Рек. МСЭ-Т I.363.1

С.9 Возможности транспортного канала

Записи таблицы, ссылающиеся на Рек. МСЭ-Т Q.931, относятся к кодированию элемента информации о возможностях транспортного канала Q.931, а не элемента информации более низкого уровня.

PropertyID	Property tag	Тип	Значение
TMR	9001	1 байт	Требование среды передачи (Transmission Medium Requirement) (Рек. МСЭ-Т Q.763) Биты <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 речь 0 0 0 0 0 0 0 1 свободный 0 0 0 0 0 0 1 0 64 кбит/с неограниченный 0 0 0 0 0 0 1 1 3.1 кГц аудио 0 0 0 0 0 1 0 0 зарезервировано для чередующейся речи (служба 2)/64 кбит/с неограниченный (служба 1) 0 0 0 0 0 1 0 1 зарезервировано для чередующейся 64 кбит/с неограниченный (служба 1)/речь (служба 2) 0 0 0 0 0 1 1 0 64 кбит/с предпочтительно 0 0 0 0 0 1 1 1 2 × 64 кбит/с неограниченный 0 0 0 0 1 0 0 0 384 кбит/с неограниченный 0 0 0 0 1 0 0 1 1536 кбит/с неограниченный 0 0 0 0 1 0 1 0 1920 кбит/с неограниченный 0 0 0 0 1 0 1 1 через свободный 0 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 3 × 64 кбит/с неограниченный 0 0 0 1 0 0 0 1 4 × 64 кбит/с неограниченный 0 0 0 1 0 0 1 0 5 × 64 кбит/с неограниченный 0 0 0 1 0 0 1 1 свободный 0 0 0 1 0 1 0 0 7 × 64 кбит/с неограниченный 0 0 0 1 0 1 0 1 8 × 64 кбит/с неограниченный 0 0 0 1 0 1 1 0 9 × 64 кбит/с неограниченный 0 0 0 1 0 1 1 1 10 × 64 кбит/с неограниченный 0 0 0 1 1 0 0 0 11 × 64 кбит/с неограниченный 0 0 0 1 1 0 0 1 12 × 64 кбит/с неограниченный 0 0 0 1 1 0 1 0 13 × 64 кбит/с неограниченный 0 0 0 1 1 0 1 1 14 × 64 кбит/с неограниченный 0 0 0 1 1 1 0 0 15 × 64 кбит/с неограниченный 0 0 0 1 1 1 0 1 16 × 64 кбит/с неограниченный 0 0 0 1 1 1 1 0 17 × 64 кбит/с неограниченный 0 0 0 1 1 1 1 1 18 × 64 кбит/с неограниченный 0 0 1 0 0 0 0 0 19 × 64 кбит/с неограниченный 0 0 1 0 0 0 0 1 20 × 64 кбит/с неограниченный 0 0 1 0 0 0 1 0 21 × 64 кбит/с неограниченный 0 0 1 0 0 0 1 1 22 × 64 кбит/с неограниченный 0 0 1 0 0 1 0 0 23 × 64 кбит/с неограниченный 0 0 1 0 0 1 0 1 свободный

PropertyID	Property tag	Тип	Значение
			0 0 1 0 0 1 1 0 25 × 64 кбит/с неограниченный 0 0 1 0 0 1 1 1 26 × 64 кбит/с неограниченный 0 0 1 0 1 0 0 0 27 × 64 кбит/с неограниченный 0 0 1 0 1 0 0 1 28 × 64 кбит/с неограниченный 0 0 1 0 1 0 1 0 29 × 64 кбит/с неограниченный 0 0 1 0 1 0 1 1 через свободный 1 1 1 1 1 1 1 1 См.: Рек. МСЭ-Т Q.763
TMRSR	9002	1 байт	Пониженная скорость требования среды передачи (Transmission Medium Requirement Subrate) 0 неопределен 1 8 кбит/с 2 16 кбит/с 3 32 кбит/с
Contcheck	9003	Логическое выражение	Контроль непрерывности передачи данных (Continuity Check) 0 контроль непрерывности на данном канале не требуется 1 контроль непрерывности на данном канале требуется См.: Рек. МСЭ-Т Q.763
ITC	9004	5 битов	Возможность передачи информации (Information Transfer Capability) Биты <u>5 4 3 2 1</u> 0 0 0 0 0 Речь 0 1 0 0 0 Неограниченная цифровая информация 0 1 0 0 1 Ограниченная цифровая информация 1 0 0 0 0 3.1 кГц аудио 1 0 0 0 1 Неограниченная цифровая информация с тональными сигналами/ извещениями 1 1 0 0 0 Видео Все остальные значения зарезервированы. См.: Рек. МСЭ-Т Q.763
TransMode	9005	2 бита	Режим передачи (Transfer Mode) Биты <u>2 1</u> 0 0 Режим передачи каналов 1 0 Режим передачи пакетов См.: Рек. МСЭ-Т Q.931
TransRate	9006	5 битов	Скорость передачи (Transfer Rate) Биты <u>5 4 3 2 1</u> 0 0 0 0 0 Данный код должен использоваться для соединений в режиме передачи пакетов 1 0 0 0 0 64 кбит/с 1 0 0 0 1 2 × 64 кбит/с 1 0 0 1 1 384 кбит/с 1 0 1 0 1 1536 кбит/с 1 0 1 1 1 1920 кбит/с 1 1 0 0 0 Многоскоростной (основная скорость 64 кбит/с) См.: Рек. МСЭ-Т Q.931

PropertyID	Property tag	Тип	Значение
MULT	9007	7 битов	Множитель скорости (Rate Multiplier) Любое значение от 2 до n (максимальное число В-каналов) См.: Рек. МСЭ-Т Q.931
layer1prot	9008	5 битов	Протокол уровня 1 информации пользователя (User Information Layer 1 Protocol) Биты <u>5 4 3 2 1</u> 0 0 0 0 1 Адаптация по скорости связи МСЭ-Т V.110 и X.30. 0 0 0 1 0 Рек.МСЭ-Т G.711 функция вида μ 0 0 0 1 1 Рек.МСЭ-Т G.711 функция вида А 0 0 1 0 0 Рек.МСЭ-Т G.726 32 кбит/с ADPCM и Рек.МСЭ-Т I.460 0 0 1 0 1 Рек.МСЭ-Т H.221 и H.242 0 0 1 1 0 Рек.МСЭ-Т H.223 и H.245 0 0 1 1 1 Адаптация по скорости стандарта не МСЭ-Т. 0 1 0 0 0 Адаптация по скорости стандарта МСЭ-Т V.120. 0 1 0 0 1 Адаптация по скорости стандарта МСЭ-Т X.31 HDLC заполнение флага Все остальные значения зарезервированы. См.: Рек. МСЭ-Т Q.931
syncasync	9009	Логическое выражение	Сигхронный/ Асинхронный 0 Синхронные данные 1 Асинхронные данные См.: Рек. МСЭ-Т Q.931
negotiation	900A	Логическое выражение	Согласование 0 согласование внутри канала возможно 1 согласование внутри канала не возможно См.: Рек. МСЭ-Т Q.931

PropertyID	Property tag	Тип	Значение
Userrate	900B	5 битов	<p>Пользовательская скорость (User Rate)</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 0 Скорость указывается в Е-битах, определенных в Рек. МСЭ-Т I.460 или может согласовываться внутри канала</p> <p>0 0 0 1 0,6 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>0 0 0 1 0 1,2 кбит/с Рек. МСЭ-Т V.6</p> <p>0 0 0 1 1 2,4 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>0 0 1 0 0 3,6 кбит/с Рек. МСЭ-Т V.6</p> <p>0 0 1 0 1 4,8 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>0 0 1 1 0 7,2 кбит/с Рек. МСЭ-Т V.6</p> <p>0 0 1 1 1 8 кбит/с Рек. МСЭ-Т I.460</p> <p>0 1 0 0 0 9,6 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>0 1 0 0 1 14,4 кбит/с Рек. МСЭ-Т V.6</p> <p>0 1 0 1 0 16 кбит/с Рек. МСЭ-Т I.460</p> <p>0 1 0 1 1 19,2 кбит/с Рек. МСЭ-Т V.6</p> <p>0 1 1 0 0 32 кбит/с Рек. МСЭ-Т I.460</p> <p>0 1 1 0 1 38,4 кбит/с Рек. МСЭ-Т V.110</p> <p>0 1 1 1 0 48 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>0 1 1 1 1 56 кбит/с Рек. МСЭ-Т V.6</p> <p>1 0 0 1 0 57,6 кбит/с Рек. МСЭ-Т V.14 расширенная</p> <p>1 0 0 1 1 28,8 кбит/с Рек. МСЭ-Т V.110</p> <p>1 0 1 0 0 24 кбит/с Рек. МСЭ-Т V.110</p> <p>1 0 1 0 1 0,1345 кбит/с Рек. МСЭ-Т X.1</p> <p>1 0 1 1 0 0,100 кбит/с Рек. МСЭ-Т X.1</p> <p>1 0 1 1 1 0,075/1,2 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 0 0 0 1,2/0,075 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 0 0 1 0,050 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 0 1 0 0,075 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 0 1 1 0,110 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 1 0 0 0,150 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 1 0 1 0,200 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 1 1 0 0,300 кбит/с Рек. МСЭ-Т V.6 и X.1</p> <p>1 1 1 1 1 12 кбит/с Рек. МСЭ-Т V.6</p> <p>Все остальные значения зарезервированы.</p> <p>См.: Рек. МСЭ-Т Q.931</p>
INTRATE	900C	2 бита	<p>Промежуточная скорость (Intermediate Rate)</p> <p>Биты</p> <p><u>2 1</u></p> <p>0 0 Не используется</p> <p>0 1 8 кбит/с</p> <p>1 0 16 кбит/с</p> <p>1 1 32 кбит/с</p> <p>См.: Рек. МСЭ-Т Q.931</p>
nictx	900D	Логическое выражение	<p>Задающий генератор, независимый от сети (Network Independent Clock (NIC) при передаче</p> <p>0 Передавать данные с задающим генератором, независимым от сети не требуется</p> <p>1 Передавать данные с задающим генератором, независимым от сети требуется</p> <p>См.: Рек. МСЭ-Т Q.931</p>

PropertyID	Property tag	Тип	Значение
nicrx	900E	Логическое выражение	<p>Задающий генератор, независимый от сети, при приеме (Network independent clock (NIC))</p> <p>0 Не может принимать данные с задающим генератором, независимым от сети (т. е. отправитель не поддерживает данную дополнительную процедуру)</p> <p>1 Может принимать данные с задающим генератором, независимым от сети (т. е. отправитель поддерживает данную дополнительную процедуру)</p> <p>См.: Рек. МСЭ-Т Q.931</p>
flowconttx	900F	Логическое выражение	<p>Управление потоком данных при передаче (Flow Control on transmission) (Tx)</p> <p>0 Передавать данные с алгоритмом управления потоком не требуется</p> <p>1 Передавать данные с алгоритмом управления потоком требуется</p> <p>См.: Рек. МСЭ-Т Q.931</p>
flowcontrx	9010	Логическое выражение	<p>Управление потоком данных при получении (Flow control on reception) (Rx)</p> <p>0 Не может принимать данные с алгоритмом управления потоком (т. е. отправитель не поддерживает данную дополнительную процедуру)</p> <p>1 Может принимать данные с алгоритмом управления потоком (т. е. отправитель не поддерживает данную дополнительную процедуру)</p> <p>См.: Рек. МСЭ-Т Q.931</p>
rateadapthdr	9011	Логическое выражение	<p>Заголовок адаптации по скорости/ нет заголовка</p> <p>0 Заголовок адаптации по скорости связи не включается</p> <p>1 Заголовок по скорости связи включается</p> <p>См.: Рек. МСЭ-Т Q.931</p>
multiframe	9012	Логическое выражение	<p>Поддержка создания множества кадров в канале передачи данных (Multiple frame establishment support in data link)</p> <p>0 Создание множества кадров не поддерживается. Разрешены только кадры UI</p> <p>1 Создание множества кадров поддерживается.</p> <p>См.: Рек. МСЭ-Т Q.931</p>
OPMODE	9013	Логическое выражение	<p>Режим работы</p> <p>0 Двоичный прозрачный режим работы</p> <p>1 Режим работы, чувствительный к протоколу</p> <p>См.: Рек. МСЭ-Т Q.931</p>
llidnegot	9014	Логическое выражение	<p>Согласование идентификатора логического канала передачи данных (Logical link identifier negotiation)</p> <p>0 Задано по умолчанию, LLI = только 256</p> <p>1 Согласование полного протокола</p> <p>См.: Рек. МСЭ-Т Q.931</p>
assign	9015	Логическое выражение	<p>Поручитель/ правопреемник</p> <p>0 Источник сообщения является "заданным по умолчанию правопреемником"</p> <p>1 Источник сообщения является "только поручителем"</p> <p>См.: Рек. МСЭ-Т Q.931</p>

PropertyID	Property tag	Тип	Значение
inbandneg	9016	Логическое выражение	Согласование внутри канала/ вне канала 0 Согласование выполняется посредством сообщений USER INFORMATION во временном соединении сигнализации 1 Согласование выполняется внутри канала с использованием логического канала ноль См.: Рек. МСЭ-Т Q.931
stopbits	9017	2 бита	Число стоповых битов Биты <u>2 1</u> 0 0 Не используется 0 1 1 бит 1 0 1.5 бита 1 1 2 бита См.: Рек. МСЭ-Т Q.931
databits	9018	2 бита	Число битов данных за исключением битов четности, если они присутствуют Биты <u>2 1</u> 0 0 Не используется 0 1 5 битов 1 0 7 битов 1 1 8 битов См.: Рек. МСЭ-Т Q.931
parity	9019	3 бита	Информация по четности (Parity information) Биты <u>3 2 1</u> 0 0 0 Нечетное 0 1 0 Четное 0 1 1 Нет 1 0 0 Установлено на 0 1 0 1 Установлено на 1 Все остальные значения зарезервированы. См.: Рек. МСЭ-Т Q.931
duplexmode	901A	Логическое выражение	Дуплексный режим (Mode duplex) 0 Полудуплекс 1 Полный дуплекс См.: Рек. МСЭ-Т Q.931

PropertyID	Property tag	Тип	Значение
modem	901B	6 битов	<p>Тип модема</p> <p>Биты</p> <p><u>6 5 4 3 2 1</u></p> <p>0 0 0 0 0 0 через Использование в стране</p> <p>0 0 0 1 0 1 0 1 0 0 0 1 Рек. МСЭ-Т V.21 0 1 0 0 1 0 Рек. МСЭ-Т V.22 0 1 0 0 1 1 Рек. МСЭ-Т V.22 <i>bis</i> 0 1 0 1 0 0 Рек. МСЭ-Т V.23 0 1 0 1 0 1 Рек. МСЭ-Т V.26 0 1 1 0 0 1 Рек. МСЭ-Т V.26 <i>bis</i> 0 1 0 1 1 1 Рек. МСЭ-Т V.26 <i>ter</i> 0 1 1 0 0 0 Рек. МСЭ-Т V.27 0 1 1 0 0 1 Рек. МСЭ-Т V.27 <i>bis</i> 0 1 1 0 1 0 Рек. МСЭ-Т V.27 <i>ter</i> 0 1 1 0 1 1 Рек. МСЭ-Т V.29 0 1 1 1 0 1 Рек. МСЭ-Т V.32 0 1 1 1 1 0 Рек. МСЭ-Т V.34</p> <p>1 0 0 0 0 0 через Использование в стране</p> <p>1 0 1 1 1 1 1 1 0 0 0 0 через Заданное пользователем</p> <p>1 1 1 1 1 1</p> <p>См.: Рек. МСЭ-Т Q.931</p>
layer2prot	901C	5 битов	<p>Протокол уровня 2 информации пользователя (User information layer 2 protocol)</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Рек. МСЭ-Т Q.921/I.441 0 0 1 1 0 Рек. МСЭ-Т X.25, канальный уровень 0 1 1 0 0 Управление логическим каналом LAN (ISO/IEC 8802-2)</p> <p>Все остальные значения зарезервированы.</p> <p>См.: Рек. МСЭ-Т Q.931</p>
layer3prot	901D	5 битов	<p>Протокол уровня 3 информации пользователя</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Рек. МСЭ-Т Q.931 0 0 1 1 0 Рек. МСЭ-Т X.25, пакетный уровень 0 1 0 1 1 ISO/IEC TR 9577 (Идентификация протокола на сетевом уровне)</p> <p>Все остальные значения зарезервированы.</p> <p>См.: Рек. МСЭ-Т Q.931</p>

PropertyID	Property tag	Тип	Значение
addlayer3prot	901E	Байт	<p>Протокол 3 уровня дополнительной информации пользователя (Additional User Information layer 3 protocol)</p> <p>Биты Биты</p> <p><u>4 3 2 1</u> <u>4 3 2 1</u></p> <p>1 1 0 0 1 1 0 0 Протокол Интернета (RFC 791) (ISO/IEC TR 9577)</p> <p>1 1 0 0 1 1 1 1 Протокол двухточечного соединения (RFC 1661)</p> <p>См.: Рек. МСЭ-Т Q.931</p>
DialledN	901F	30 байтов	Набранный номер (Dialled Number)
DiallingN	9020	30 байтов	Набирающий номер (Dialling Number)
ECHOICI	9021		Не используется. Пример возможных свойств управления эхо (Echo Control Properties) см. в E.13.
NCI	9022	1 байт	<p>Природа индикаторов соединения (Nature of Connection Indicators)</p> <p>Биты</p> <p><u>2 1</u> <i>Спутниковый индикатор</i></p> <p>0 0 спутниковых каналов в соединении нет</p> <p>0 1 один спутниковый канал в соединении</p> <p>1 0 два спутниковых канала в соединении</p> <p>1 1 свободен</p> <p>Биты</p> <p><u>4 3</u> <i>Индикатор контроля непрерывности передачи</i></p> <p>0 0 контроль непрерывности не требуется</p> <p>0 1 контроль непрерывности требуется на данном канале</p> <p>1 0 контроль непрерывности выполнялся на предыдущем канале</p> <p>Бит</p> <p><u>5</u> <i>Индикатор устройства контроля эхо</i></p> <p>0 исходящее устройство контроля эхо не включено</p> <p>1 исходящее устройство контроля эхо включено</p> <p>Биты</p> <p><u>8 7 6</u> свободен</p> <p>См.: Рек. МСЭ-Т Q.763</p>
USI	9023	Байтовая строка	<p>Информация службы пользователя (User Service Information)</p> <p>См.: 3.57/Q.763</p>

С.10 Свойства AAL 5

PropertyID	Property tag	Тип	Значение
FMSDU	A001	32-битное целое число	Максимальный размер CPCS-SDU в прямом направлении (Forward Maximum CPCS-SDU Size): Максимальный размер CPCS-SDU, передаваемый в направлении от вызывающего пользователя к вызываемому пользователю. См.: Рек. МСЭ-Т Q.2931
BMSDU	A002	32-битное целое число	Максимальный размер CPCS-SDU в обратном направлении (Backwards Maximum CPCS-SDU Size): Максимальный размер CPCS-SDU, передаваемый в направлении от вызываемого пользователя к вызывающему пользователю. См.: Рек. МСЭ-Т Q.2931
SSCS	См. таблицу С.7	См. таблицу в С.7	См. таблицу в С.7 Дополнительные значения: VPI/VCI

С.11 Эквиваленты SDP

На эквиваленты SDP действуют исключения, обозначенные в п. 7.1.8 для текстового кодирования данного протокола. Например, групповой символ замены CHOOSE разрешено использовать в направлении от MGC к MG вне зависимости от кодирования (двоичного или текстового) данного протокола.

PropertyID	Property tag	Тип	Значение
SDP_V	B001	Строка	Версия протокола См.: RFC 2327
SDP_O	B002	Строка	Владелец/ создатель и ID сеанса См.: RFC 2327
SDP_S	B003	Строка	Название сеанса См.: RFC 2327
SDP_I	B004	Строка	Идентификатор сеанса См.: RFC 2327
SDP_U	B005	Строка	URI дескриптора См.: RFC 2327
SDC_E	B006	Строка	Адрес электронной почты См.: RFC 2327
SDP_P	B007	Строка	Номер телефона См.: RFC 2327
SDP_C	B008	Строка	Информация о соединении См.: RFC 2327
SDP_B	B009	Строка	Информация о пропускной способности См.: RFC 2327
SDP_Z	B00A	Строка	Настройка часового пояса См.: RFC 2327
SDP_K	B00B	Строка	Ключ шифрования См.: RFC 2327

PropertyID	Property tag	Тип	Значение
SDP_A	B00C	Строка	Ноль или более атрибутов сеанса См.: RFC 2327
SDP_T	B00D	Строка	Активное время сеанса (Active Session Time) См.: RFC 2327
SDP_R	B00E	Строка	Ноль или более раз повторения См.: RFC 2327
SDP_M	B00F	Строка	Тип медиа, порт, вид передачи и формат См.: RFC 2327

C.12 H.245

PropertyID	Property tag	Тип	Значение
OLC	C001	Байтовая строка	Значение структуры H.245 OpenLogicalChannel. См.: Рек.МСЭ-Т Н.245
OLCack	C002	Байтовая строка	Значение структуры H.245 OpenLogicalChannelAck. См.: Рек.МСЭ-Т Н.245
OLCcnf	C003	Байтовая строка	Значение структуры H.245 OpenLogicalChannelConfirm. См.: Рек.МСЭ-Т Н.245
OLCrej	C004	Байтовая строка	Значение структуры H.245 OpenLogicalChannelReject. См.: Рек.МСЭ-Т Н.245
CLC	C005	Байтовая строка	Значение структуры H.245 CloseLogicalChannel. См.: Рек.МСЭ-Т Н.245
CLCack	C006	Байтовая строка	Значение структуры H.245 CloseLogicalChannelAck. См.: Рек.МСЭ-Т Н.245
LCN	C007	Целое число	Значение H.245 Local Channel Number 0-65535. См.: Рек.МСЭ-Т Н.245

Приложение D

Передача по IP

D.1 Передача по IP/UDP с использованием кадрирования на прикладном уровне (ALF)

Сообщения протокола, определенные в данной Рекомендации, можно передавать по UDP. Если равноправный участник сети не предоставляет порта (см. п. 7.2.8), команды следует посылать на заданный по умолчанию порт: 2944 для работы в текстовом кодировании, или 2945 для работы в двоичном кодировании. Ответы следует посылать на адрес и порт, с которого были посланы соответствующие команды.

ALF представляет собой набор методов, позволяющих приложению, в противоположность стеку, влиять на то, как сообщения передаются другой стороне. Типичным методом ALF является предоставление приложению возможности изменять порядок отсылаемых сообщений в очереди уже после того, как данная очередь сформирована. Формальной спецификации для ALF нет. Процедуры в Приложении D.1 содержат минимальный рекомендуемый набор режимов работы ALF.

Реализаторы, использующие IP/UDP с ALF должны знать об ограничениях MTU по максимальному размеру сообщения.

D.1.1 Обеспечение функциональной возможности "Не более одного раза"

Сообщения, передаваемые по UDP могут нести потери. При отсутствии своевременного ответа, команды повторяются. Большая часть команд не являются идемпотентными. Например, если команды Add будут выполнены несколько раз, состояние MG станет непредсказуемым. Таким образом, процедуры передачи должны обеспечивать функциональную возможность "Не более одного раза" ("At-Most-Once")

Предполагается, что равноправные объекты протокола содержат в памяти список ответов, которые они отправили по недавним транзакциям и список транзакций, находящихся в состоянии ожидания в настоящий момент. TransactionID каждого входящего сообщения сравнивается с идентификаторами TransactionID недавних ответов, направленных тому же MID. Если обнаруживается совпадение, объект не выполняет транзакцию, а просто повторяет ответ. Если совпадения не обнаруживается, сообщение будет сравниваться со списком транзакций, находящихся в состоянии ожидания в настоящий момент. Если совпадение обнаруживается в данном списке, что означает дублирующую транзакцию, объект не выполняет транзакцию (процедуры по передаче TransactionPending см. в D.1.4).

Процедура использует значение длинного таймера, обозначаемого LONG-TIMER в следующих случаях. Значение таймера должно быть больше максимальной продолжительности транзакции, причем следует учесть максимальное число повторений, максимальное значение таймера повторений и максимальную задержку распространения пакета в сети. Рекомендуемое значение – 30 секунд.

Копии ответов могут быть удалены по истечению секунд LONG-TIMER после отправки ответа, либо когда объект получает подтверждение получения ответа посредством "Response Acknowledgement parameter (Параметр подтверждения ответа)". Для транзакций, подтверждаемых посредством данного параметра, объект должен сохранять копию TransactionID в течение секунд LONG-TIMER после отправки ответа, для обнаружения пропуска дублирующих копий запроса транзакций, которые могут возникнуть в сети.

D.1.2 Идентификаторы транзакций и трехстороннее квитирование

D.1.2.1 Идентификаторы транзакций

Идентификаторы TransactionID представляют собой 32-битные целые числа. Контроллер медиашлюза может принять решение об использовании определенного числового пространства для каждого из шлюзов MG, которыми они управляют, или об использовании одного и того же числового пространства для всех MG, принадлежащих какой-либо произвольной группе. MGC могут принять решение разделить нагрузку по управлению большим MG на несколько независимых процессов. Данные транзакции будут иметь общее числовое пространство транзакции. Существует множество возможных реализаций такого разделения таких, как централизованное размещение идентификаторов TransactionID, или предварительное размещение неперекрывающихся диапазонов идентификаторов различных процессов. Данные реализации должны гарантировать, что всем транзакциям, происходящим из логического MGC (идентичного MID), назначены уникальные TransactionID. Шлюзы MG могут просто обнаружить дублирующие транзакции, рассмотрев лишь TransactionID и MID.

D.1.2.2 Трехстороннее квитирование

Параметр TransactionResponseAcknowledgement может быть обнаружен в любом сообщении. Он несет набор "подтвержденных диапазонов TransactionID". Объекты могут принять решение об удалении копий ответов на транзакции, ID которых включены в "подтвержденные диапазоны TransactionID", полученные в ответных сообщениях на транзакции. Если TransactionID попадает в эти диапазоны, им следует просто удалить дальнейшие команды.

Значения "подтвержденных диапазонов TransactionID" нельзя использовать, если со времени, когда MG отправил последний ответ MGC, или когда MG возобновил работу, прошло больше секунд, чем LONG-TIMER. В данной ситуации, следует принимать и обрабатывать транзакции, не проверяя TransactionID.

Сообщения, содержащие параметр "TransactionResponseAcknowledgement", можно передавать в любом порядке. Объект должен сохранять полученные "подтвержденные диапазоны TransactionID" в течение секунд LONG-TIMER.

При двоичном кодировании, если в подтверждении ответа присутствует только firstAck (см. A.2), подтверждается только одна транзакция. Если присутствуют firstAck и lastAck одновременно, происходит подтверждение транзакций от firstAck до lastAck. При текстовом кодировании, для обозначения диапазона подтверждаемых транзакций используется горизонтальное длинное тире (см. B.2).

D.1.3 Вычисление таймеров повторной передачи

Обеспечение соответствующих промежутков ожидания события для всех транзакций, ожидающих выполнения, и осуществление повторных попыток по транзакциям по истечении данных промежутков, является обязанностью запрашивающего объекта. Кроме того, при отказе в подтверждении повторных транзакций, поиск резервных служб и/или очистка существующих или ожидающих соединений является обязанностью запрашивающего объекта.

В данной спецификации намеренно избегается определение какого-либо значения для таймеров повторной передачи. Обычно данные значения зависят от сети. Обычно таймеры повторной передачи должны установить значение таймера путем измерения времени от отправки команды до получения ответа. Реализации должны обеспечивать, чтобы алгоритм, используемый для вычисления интервала повторной передачи, обеспечивал экспоненциально возрастающую выдержку времени ожидания до повторной передачи для каждой повторной передачи или повторения после первой повторной передачи.

ПРИМЕЧАНИЕ. – Одной из таких возможностей является использование алгоритма, применяемого в TCP-IP, который использует две переменные:

- Средняя задержка подтверждения (average acknowledgement delay (AAD)), измеряемая посредством экспоненциально сглаженного среднего наблюдаемых задержек.
- Среднее отклонение (average deviation (ADEV)), измеряемое посредством экспоненциально сглаженного среднего разницы между наблюдаемой задержкой и текущим средним значением. Значение таймера повторной передачи в TCP устанавливается на сумму средней задержки и N раз среднего отклонения. Однако, для протокола, описанного в данной Рекомендации, максимальное значение данного таймера должно быть ограничено для того, чтобы гарантировать, что по истечении секунд LONG-TIMER шлюзы не будут повторно получать пакеты. Рекомендуемое максимальное значение – 4 секунды.

После каждой повторной передачи объект должен выполнить следующее:

- Удвоить предполагаемое значение средней задержки, AAD.
- Вычислить случайное значение, равномерно распределенное между 0.5 AAD и AAD.
- Выставить таймер повторной передачи на сумму данного случайного значения и N раз среднего отклонения.

У данной процедуры два следствия. Поскольку она включает в себя экспоненциально возрастающий компонент, в случае перегрузки поток сообщений автоматически замедлится. Поскольку она включает в себя случайный компонент, потенциальной синхронизации между уведомлениями, вызываемыми одним и тем же внешним событием, не произойдет.

D.1.4 Предварительные ответы

Выполнение некоторых транзакций может потребовать много времени. Продолжительные промежутки времени выполнения могут взаимодействовать с процедурой повторной передачи, основанной на таймере. Это может привести либо к чрезмерному числу повторных передач, или к увеличению значений таймера так, что он становится неэффективным. Объекты, обладающие способностью прогнозировать, что выполнение транзакции займет много времени, могут отправить предварительный ответ, "TransactionPending". Им следует отправить данный ответ, если они получают повтор транзакции, выполняемой в данный момент.

Объекты, получившие TransactionPending должны переключиться на другой таймер повторной передачи для повторения запросов. После получения окончательного ответа, следующего за получением предварительных ответов, нужно отправить промежуточное подтверждение, а затем

следует использовать обычные таймеры повторной передачи. Объект, посылающий предварительный ответ, должен включить в последующий окончательный ответ поле `immAckRequired`, показывающее, что ожидается незамедлительное подтверждение. Получение `TransactionPending` после получения ответа нужно проигнорировать.

D.1.5 Повторение запросов, ответов и подтверждений

Данный протокол организован как набор транзакций, каждая из которых состоит из запроса и ответа, который обычно называют подтверждением. Сообщения протокола, передаваемые по UDP, могут теряться. При отсутствии своевременного ответа, происходит повторение транзакций. Предполагается, что объекты сохраняют в памяти список ответов, которые они отправили по недавним транзакциям, т.е список всех ответов, которые они отправили за последние секунды `LONG-TIMER`, и список транзакций, находящихся в состоянии ожидания в настоящий момент.

Алгоритм повторения используется для предотвращения трех типов возможных ошибок:

- ошибки при передаче, когда, например, потеря пакета происходит из-за шума на линии или перегрузки в очереди;
- сбой компонента, когда, например, интерфейс объекта становится недоступным;
- сбой объекта, когда, например, недоступным становится весь объект.

Объекты должны быть в состоянии извлечь из прошлой истории оценку коэффициента потерь пакетов из-за ошибок передачи. В правильно сконфигурированной системе этот коэффициент потерь должен поддерживаться очень низким, обычно менее 1%. Если контроллеру медиашлюза или медиашлюзу приходится повторять сообщение более нескольких раз, очень разумно предположить, что происходит что-то еще, кроме ошибок передачи. Например, принимая коэффициент потерь равным 1%, вероятность того, что пять попыток передачи подряд окажутся неудачными, равна 1 из 100 миллиардов. Такой случай должен происходить меньше, чем один раз в каждые 10 дней для контроллера медиашлюза, обрабатывающего 1000 транзакций в секунду. (В самом деле, число повторений, которое полагается слишком большим, должно быть функцией преобладающего коэффициента потерь пакетов). Мы должны отметить, что "опасный пороговый уровень", который мы будем называть "Max1", обычно ниже, чем "пороговый уровень разъединения", который должен быть установлен равным большей величине.

При классическом алгоритме повторной передачи произойдет простой подсчет числа последовательных повторений, и при чрезмерном количестве повторных передач будет сделано заключение, что взаимосвязь разорвана. Для того, чтобы подсчитать вероятность необнаруженного или текущего "преодоление отказа", мы изменяем классический алгоритм так, что, если медиашлюз получает правильное сообщение `ServiceChange`, объявляющее преодоление отказа, он начнет передачу команд, ожидающих выполнения, этому новому контроллеру MGC. Ответы на команды все еще передаются по исходному адресу команды.

Для автоматической адаптации к нагрузке сети в данной Рекомендации определяются экспоненциально возрастающие таймеры. Если начальный таймер установлен на 200 миллисекунд, потеря пятой повторной передачи будет обнаруживаться примерно спустя 6 секунд. Возможно, что это допустимая задержка ожидания для обнаружения преодоления отказа. После такой задержки повторы передачи должны происходить не только для того, чтобы, возможно, преодолеть временную проблему возможности соединения, но также для того, чтобы предоставить несколько больше времени для осуществления преодоления отказа (возможно, приемлемой является ожидание общей задержки, равной 30 секундам).

Однако важно, чтобы максимальная задержка повторных передач была бы ограниченной. Перед любой повторной передачей проверяется, что время, прошедшее с момента отправки начальной дейтаграммы, не превышает `T-MAX`. Если прошло время большее, чем `T-MAX`, MG приходит к заключению, что контроллер медиашлюза вышел из строя и начинает свой восстановительный процесс, как описано в 11.5. Если MG пытается повторно соединиться с текущим MGC, он должен использовать `ServiceChange` с `ServiceChangeMethod`, установленным в состояние `Disconnected` так, чтобы новый MGC был бы уверен, что шлюз MG потерял одну или более транзакций. Значение `T-MAX` относится к значению `LONG-TIMER`: значение `LONG-TIMER` получается путем прибавления к `T-MAX` максимальной задержки распространения в сети.

D.2 Использование TCP

Сообщения протокола, как определено в данной Рекомендации, могут передаваться по TCP. Если другой стороной не определено какого-либо порта (см. п. 7.2.8), команды следует посылать на порт, заданный по умолчанию. В определенном протоколе единицами передачи являются сообщения, в то время, как TCP является поточным протоколом. Согласно RFC 1006, для выражения сообщений в потоке TCP нужно использовать TPKT.

В протоколе, ориентированном на транзакции все

Рекомендуется, чтобы объекты, использующие передачу TCP, применяли таймеры прикладного уровня для каждого запроса и каждого ответа, также как это описывается для кадрирования прикладного уровня по UDP.

D.2.1 Обеспечение функциональной возможности "не более одного раза"

Сообщения, передаваемые по TCP, не подвержены потерям при передаче, однако тем не менее в реальных реализациях можно заметить потерю запроса или ответа транзакции. При отсутствии своевременного ответа, происходит повторение команд. Большинство команд не идемпотентны. Например, если команды Add будут выполнены несколько раз, состояние MG станет непредсказуемым.

Для защиты от таких потерь рекомендуется, чтобы объекты следовали процедурам в D.1.1.

D.2.2 Идентификаторы транзакций и трехстороннее квитирование

По тем же причинам возможна потеря ответов на транзакции, даже при надежном протоколе доставки таком, как TCP. Рекомендуется, чтобы объекты следовали процедурам в D.1.2.2.

D.2.3 Вычисление таймеров повторной передачи

Предполагается, что при надежной доставке вероятность потери запроса или ответа транзакции очень низка. Поэтому, требуются только простые алгоритмы таймеров. Экспоненциальные алгоритмы отсрочки не нужны, хотя их можно использовать там, где, как и в MGC, данные алгоритмы уже требуются, поскольку контроллеры MGC должны применять ALF/UDP, а также TCP.

D.2.4 Предварительные ответы

Как и в случае с UDP, выполнение некоторых транзакций может потребовать большого количества времени. Объекты, которые могут прогнозировать, что выполнение транзакции потребует большого количества времени, могут отправить предварительный ответ, "TransactionPending". Им следует отправить данный ответ, если они получают повторение транзакции, которая все еще выполняется.

Объекты, получающие TransactionPending, для данной транзакции должны переключиться на таймер повторной отсылки большей продолжительности.

Объекты должны сохранять транзакции и ответы до тех пор, пока они не будут подтверждены. Следует придерживаться основной процедуры D.1.4, однако достаточными являются значения простого таймера. После получения окончательного ответа посылать подтверждение немедленно не нужно.

D.2.5 Расположение команд

TCP обеспечивает упорядоченную доставку транзакций. Специальных процедур не требуется. Следует заметить, что ALF/UDP разрешает отсылающему объекту изменять режим работы при перегрузке, в особенности, изменять порядок следования транзакций при перегрузке. TCP не может достичь тех же результатов.

Приложение Е

Основные комплекты

В данном приложении содержатся определения некоторых комплектов для использования в рамках данной Рекомендации.

Е.1 Generic

Название комплекта: Generic

PackageID: g (0x0001)

Описание: Общий комплект для обычно встречающихся элементов

Версия: 2

Расширения: Нет

Е.1.1 Свойства

Нет

Е.1.2 События

Е.1.2.1 Cause

Название события: Cause

EventID: cause (0x0001)

Описание: Общее событие ошибки

Параметры EventsDescriptor: Нет

Параметры дескриптора ObservedEvents:

General Cause

Название параметра: General Cause

ParameterID: Generalcause (0x0001)

Описание: Данный параметр разделяет ошибки на шесть групп, в соответствии с которыми может действовать MGC.

Тип: перечисление

Дополнительный: Нет

Возможные значения:	"NR" (0x0001)	Нормальный	Освободить
	"UR" (0x0002)	Недоступен	Ресурсы
	"FT" (0x0003)	Сбой,	Временный
	"FP" (0x0004)	Сбой,	Постоянный
	"IW" (0x0005)	Взаимодействие	Ошибка
	"UN" (0x0006)	Не поддерживается	

Задано по умолчанию: Нет

Failure Cause

Название параметра: Failure Cause

ParameterID: Failurecause (0x0002)

Описание: Failure Cause представляет собой значение, генерируемое освобожденным оборудованием, т. е. освобожденным соединением сети. Рассматриваемое значение определяется в соответствующем протоколе управления транспортным каналом.

Тип: БАЙТОВАЯ СТРОКА

Дополнительный: Да

Возможные значения: БАЙТОВАЯ СТРОКА

Задано по умолчанию: Нет

E.1.2.2 Signal Completion

Название события: Signal Completion

EventID: sc (0x0002)

Описание: Показывает завершение сигнала, для которого был установлен параметр NotifyCompletion для того, чтобы предоставлять отчет о событии завершения. Более подробное описание данной процедуры, см. в пп. 7.1.1, 7.1.11 и 7.2.7.

Параметры EventsDescriptor: Нет

Параметры дескриптора ObservedEvents:

Signal Identity

Название параметра: Signal Identity

ParameterID: SigID (0x0001)

Описание: Данный параметр идентифицирует завершившийся сигнал. Для сигнала, содержащегося в SignalList, также должно происходить возвращение параметра SignalListID, обозначающего соответствующий список.

Тип: Двоичное кодирование: байт (строка), Текстовое кодирование: строка

Дополнительный: Нет

Возможные значения: завершившийся сигнал. Идентификация сигнала должно происходить путем использования синтаксиса pkgdName без групповых символов замены.

Задано по умолчанию: Нет

Termination Method

Название параметра: Termination Method

ParameterID: Meth (0x0002)

Описание: Показывает средства, при помощи которых завершается сигнал.

Тип: Перечисление

Дополнительный: Нет

Возможные значения:

"TO" (0x0001) Время действия сигнала истекло, либо он завершился сам по себе

"EV" (0x0002) Прерван событием

"SD" (0x0003) Остановлен новым дескриптором Signals

"NC" (0x0004) Не завершен, другая причина

"PI" (0x0005) Сначала до предпоследней итерации. Для последнего повторения используйте TO.

Задано по умолчанию: Нет

Signal List ID

Название параметра: Signal List ID

ParameterID: SLID (0x0003)

Описание: Показывает, к какому SignalList принадлежит сигнал. Возвращение SignalListID происходит только в случаях, когда сигнал находится в SignalList.

Тип: Целое число

Дополнительный: Да (поддерживается, только если используются списки сигналов)

Возможные значения: 1-65535

Задано по умолчанию: Нет

Request ID

Название параметра: Request ID

ParameterID: RID (0x0004)

Описание: Показывает, с каким запросом NotifyCompletion связан данный SignalID.

Тип: Целое число

Дополнительный: Да (поддерживается, только если используются списки сигналов)

Возможные значения: 1-4294967295

Задано по умолчанию: Нет

Е.1.3 Сигналы

Нет

Е.1.4 Статистика

Нет

Е.2 Комплект Base Root

Название комплекта: Base Root Package

PackageID: root (0x0002)

Описание: Данный комплект определяет широкие (общие) свойства шлюза.

Версия: 2

Расширения: Нет

Е.2.1 Свойства

Е.2.1.1 Maximum Number of Contexts

Название свойства: MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

Описание: Значение данного свойства отражает максимальное число контекстов, которые могут существовать в любой момент времени. Нулевой контекст (NULL) в данное число не включается.

Тип: Двойное число

Возможные значения: 1 и более

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: ReadOnly (только для чтения)

E.2.1.2 Maximum Terminations Per Context

Название свойства: MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

Описание: Максимальное разрешенное число завершений в контексте, см. п. 6.1.

Тип: Целое число

Возможные значения: любое положительное целое число

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: ReadOnly

E.2.1.3 Normal MG Execution Time

Название свойства: normalMGExecutionTime

PropertyID: normalMGExecutionTime (0x0003)

Описание: Устанавливается MGC для обозначения интервала, в рамках которого MGC ожидает ответа на любую транзакцию от MG (не включая задержку сети).

Тип: Целое число

Возможные значения: любое положительное целое число; обозначает миллисекунды

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write (Чтение/ Запись)

E.2.1.4 Normal MGC Execution Time

Название свойства: normalMGCExecutionTime

PropertyID: normalMGCExecutionTime (0x0004)

Описание: Устанавливается MGC для обозначения интервала времени, в рамках которого шлюзу MG следует ожидать ответа на любую транзакцию от MGC (не включая задержку сети).

Тип: Целое число

Возможные значения: любое положительное целое число; обозначает миллисекунды

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write (Чтение/ Запись)

E.2.1.5 MG Provisional Response Timer Value

Название свойства: MGProvisionalResponseTimerValue

PropertyID: MGProvisionalResponseTimerValue (0x0005)

Описание: Показывает время, в течение которого контроллеру MGC следует ожидать Pending Response от MG в случае, если транзакция не может быть завершена. Первоначальное значение устанавливается на normalMGExecutionTime плюс задержка сети, однако его можно снизить.

Тип: Целое число

Возможные значения: любое положительное целое число; обозначает миллисекунды

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.2.1.6 MGC Provisional Response Timer Value

Название свойства: MGCProvisionalResponseTimerValue

PropertyID: MGCProvisionalResponseTimerValue (0x0006)

Описание: Показывает время, в течение которого шлюзу MG следует ожидать Pending Response от MGC в случае, если транзакция не может быть завершена. Первоначальное значение устанавливается на normalMGCExecutionTime плюс задержка сети, однако его можно снизить.

Тип: Целое число

Возможные значения: любое положительное целое число; обозначает миллисекунды

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.2.1.7 MGC Originated Pending Limit

Название свойства: MGCOriginatedPendingLimit

PropertyID: MGCOriginatedPendingLimit (0x0007)

Описание: Обозначает число TransactionPending, которое может быть получено от MGC. При превышении данного предела MGC должен послать TransactionReply с Ошибкой 506 ("Превышение числа TransactionPendings"); иначе MG может предположить, что в транзакции произошла ошибка.

Тип: Целое число

Возможные значения: любое положительное целое число

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.2.1.8 MG Originated Pending Limit

Название свойства: MGOriantedPendingLimit

PropertyID: MGOriantedPendingLimit (0x0008)

Описание: Обозначает число TransactionPending, которое может быть получено от MG. При превышении данного предела MG должен послать TransactionReply с Ошибкой 506 ("Превышение числа TransactionPendings"); иначе MGC может предположить, что в транзакции произошла ошибка.

Тип: Целое число

Возможные значения: любое положительное целое число

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.2.2 События

Нет

E.2.3 Сигналы

Нет

E.2.4 Статистика

Нет

E.2.5 Процедуры

Нет

E.3 Комплект Tone Generator

Название комплекта: Tone Generator Package

PackageID: tonegen (0x0003)

Описание: В данном комплекте определяются сигналы для генерирования тональных сигналов аудио. В данном комплекте не определяются значения параметров. Он предназначен для расширения. В общем тональные сигналы определяются как отдельные сигналы с параметром, индикатором, представляющим "межцифровую" временную задержку, и ID тонального сигнала, используемого с тональным сигналом воспроизведения. ID тонального сигнала должен оставаться неизменным для одного и того же тонального сигнала при любом генерировании тонального сигнала. Предполагается, что шлюзам MG предоставлены характеристики соответствующих тональных сигналов для страны, в которой расположен данный MG.

Версия: 2

Разработано только для расширения: Да

Расширения: Нет

E.3.1 Свойства

Нет

E.3.2 События

Нет

E.3.3 Сигналы

E.3.3.1 Play Tone

Название сигнала: Play Tone

SignalID: pt (0x0001)

Описание: Воспроизводит тональный сигнал аудио в аудиоканале

Тип сигнала: Короткий

Продолжительность: Предоставляется

Дополнительные параметры:

Tone ID list

Название параметра: Tone ID List

ParameterID: tl (0x0001)

Описание: Список тональных сигналов, предназначенных для последовательного воспроизведения.

Тип: Подсписок перечисления

Дополнительный: Нет

Возможные значения: Подсписок должен содержать один или более идентификаторов тональных сигналов.

Задано по умолчанию: Нет

Inter-signal duration

Название параметра: Inter-signal duration

ParameterID: ind (0x0002)

Описание: Перерыв между двумя последовательными тональными сигналами в миллисекундах

Тип: Целое число

Дополнительный: Да

Возможные значения: Любое целое число большее или равное 0

Задано по умолчанию: Предоставляется

Tone Direction

Название параметра: Tone Direction

ParameterID: btd (0x0003)

Описание: Направление, в котором тональный сигнал движется от завершения.

Тип: Перечисление

Дополнительный: Да

Возможные значения: "EXT" (0x0001) Наружу
"INT" (0x0002) Внутрь
"BOTH" (0x0003) Оба

Задано по умолчанию: Наружу

E.3.4 Статистика

Нет

E.3.5 Процедуры

В данном комплекте идентификаторы тональных сигналов не определяются. Комплекты, расширяющие данный комплект, могут добавлять возможные значения идентификаторов тональных сигналов, а также добавлять отдельные тональные сигналы.

E.4 Комплект Tone Detection

Название комплекта: Tone Detection Package

PackageID: tonedet (0x0004)

Описание: В данном комплекте определяются события для обнаружения тональных сигналов аудио. Тональные сигналы выбираются по названию (ID тонального сигнала). Предполагается, что шлюзам MG предоставлены характеристики соответствующих тональных сигналов для страны, в которой расположен данный MG.

В данном комплекте не определяются значения параметров. Он предназначен для расширения.

Версия: 1

Разработан только для расширения: Да

Расширения: Нет

E.4.1 Свойства

Нет

E.4.2 События

E.4.2.1 Start Tone Detected

Название события: Start tone detected

EventID: std (0x0001)

Описание: Обнаруживает начало тонального сигнала. Характеристики положительного обнаружения тонального сигнала зависят от реализации.

Параметры EventsDescriptor:

Tone ID List

Название параметра: Tone ID List

ParameterID: tl (0x0001)

Тип: Подсписок перечисления

Дополнительный: Нет

Возможные значения: Единственным ID тонального сигнала, определенным в данном комплекте, является "групповой символ замены" "*" в текстовом кодировании и 0x0000 в двоичном кодировании. Расширения для данного комплекта могут добавлять возможные значения ID тонального сигнала. Если список тонов "обобщен", обнаруживаются все ID тонального сигнала.

Задано по умолчанию: Нет

Параметры ObservedEventsDescriptor:

Tone ID

Название параметра: Tone ID

ParameterID: tid (0x0003)

Описание: ID обнаруживаемого тонального сигнала.

Тип: Перечисление

Дополнительный: Нет

Возможные значения: "групповой символ замены", определенный выше, является единственным значением, определяемым в данном комплекте. Расширения к данному комплекту могут добавлять дополнительные возможные значения ID тональных сигналов.

Задано по умолчанию: Нет

E.4.2.2 End Tone Detected

Название события: End tone detected

EventID: etd (0x0002)

Описание: Обнаруживает завершение тонального сигнала.

Параметры EventsDescriptor:

Tone ID List

Название параметра: Tone ID List

ParameterID: tl (0x0001)

Описание: Список идентификаторов тональных сигналов для обнаружения.

Тип: Подсписок перечисления

Дополнительный: Нет

Возможные значения: В данном комплекте не определяется возможных значений. Расширения к данному комплекту могут добавлять возможные значения ID тональных сигналов.

Задано по умолчанию: Нет

Параметры ObservedEventsDescriptor:

Tone ID

Название параметра: Tone ID

ParameterID: tid (0x0003)

Описание: ID обнаруженного тонального сигнала.

Тип: Перечисление

Дополнительный: Нет

Возможные значения: Единственным значением, определяемым в данном комплекте, является "групповой символ замены", определенный выше. Расширения к данному комплекту могут добавлять возможные значения ID тональных сигналов.

Задано по умолчанию: Нет

Duration

Название параметра: Duration

ParameterID: dur (0x0002)

Описание: В данном параметре содержатся данные по продолжительности тонального сигнала с момента первого обнаружения до прекращения сигнала.

Тип: Целое число

Дополнительный: Да

Возможные значения: Любое положительное целое число в миллисекундах

Задано по умолчанию: Нет

E.4.2.3 Long Tone Detected

Название события: Long Tone Detected

EventID: ltd (0x0003)

Описание: Обнаруживает, что воспроизведение тонального сигнала происходит по крайней мере в течение определенного количества времени

Параметры EventsDescriptor:

Tone ID List

Название параметра: Tone ID List

ParameterID: tl (0x0001)

Описание: Список идентификаторов тональных сигналов для обнаружения.

Тип: Подсписок перечисления

Дополнительный: Нет

Возможные значения: Единственным значением, определяемым в данном комплекте является "групповой символ замены", определенный выше. Расширения к данному комплекту могут добавлять возможные значения ID тонального сигнала.

Задано по умолчанию: Нет

Duration

Название параметра: Duration

ParameterID: dur (0x0002)

Описание: Продолжительность, на основе которой нужно проводить испытания.

Тип: Целое число

Дополнительный: Да

Возможные значения: любое положительное целое число, выраженное в миллисекундах

Задано по умолчанию: Предоставляется

Параметры ObservedEventsDescriptor:

Tone ID

Название параметра: Tone ID

ParameterID: tid (0x0003)

Описание: ID обнаруженного тонального сигнала.

Тип: Перечисление

Дополнительный: Нет

Возможные значения: В данном комплекте возможные значения не определяются. Расширения к данному комплекту могут добавлять возможные значения ID тонального сигнала.

Задано по умолчанию: Нет

Е.4.3 Сигналы

Нет

Е.4.4 Статистика

Нет

Е.4.5 Процедуры

Нет

Е.5 Комплект Basic DTMF Generator

Название комплекта: Basic DTMF Generator Package

PackageID: dg (0x0005)

Описание: В данном комплекте основные тональные сигналы DTMF определяются как сигналы, и расширяются разрешенные значения для параметра списка тональных сигналов воспроизведения тонального сигнала в генераторе тональных сигналов.

Версия: 2

Расширения: генератор тональных сигналов версия 2

Е.5.1 Свойства

Нет

Е.5.2 События

Нет

Е.5.3 Сигналы

Е.5.3.1 DTMF Character 0

Название сигнала: DTMF Character 0

SignalID: d0 (0x0010)

Описание: Генерирует тональный сигнал DTMF 0. Физические характеристики DTMF 0 шлюзе.

Тип сигнала: Короткий

Продолжительность: Предоставляется

Дополнительные параметры:

Tone Direction

Название параметра: Tone Direction

ParameterID: btd (0x0001)

Описание: Направление, в котором тональный сигнал должен двигаться от завершения.

Тип: Перечисление

Дополнительный: Да

Возможные значения: "EXT" (0x0001) Наружу
"INT" (0x0002) Внутрь
"BOTH" (0x0003) Оба

Задано по умолчанию: Наружу

Дополнительные значения: значение d0 (0x0010) определено как ID тонального сигнала для тонального сигнала воспроизведения.

Определение остальных символов DTMF происходит точно таким же образом. Далее приводится таблица со всеми названиями сигналов и идентификаторами SignalID. Заметьте, что каждый символ DTMF определяется как сигнал и как ID тонального сигнала, таким образом расширяя основной комплект генерирования тонального сигнала. Также заметьте, что идентификаторы SignalID DTMF отличаются от названий, используемых в DigitMap.

Название сигнала	SignalID/Tone ID
DTMF character 0	d0 (0x0010)
DTMF character 1	d1 (0x0011)
DTMF character 2	d2 (0x0012)
DTMF character 3	d3 (0x0013)
DTMF character 4	d4 (0x0014)
DTMF character 5	d5 (0x0015)
DTMF character 6	d6 (0x0016)
DTMF character 7	d7 (0x0017)
DTMF character 8	d8 (0x0018)
DTMF character 9	d9 (0x0019)
DTMF character *	ds (0x0020)
DTMF character #	do (0x0021)
DTMF character A	da (0x001a)
DTMF character B	db (0x001b)
DTMF character C	dc (0x001c)
DTMF character D	dd (0x001d)

Е.5.4 Статистика

Нет

Е.5.5 Процедуры

Нет

Е.6 Комплект DTMF Detection

Название комплекта: DTMF Detection Package

PackageID: dd (0x0006)

Описание: В данном комплекте определяется обнаружение основных тональных сигналов DTMF. Данный комплект расширяет возможные значения ID тонального сигнала в событиях Start Tone Detected, End Tone Detected и Long Tone Detected.

Все дополнительные значения ID тональных сигналов являются идентификаторами ID тональных сигналов, описанных в комплекте dg (Basic DTMF Generator Package).

В нижеследующей таблице проводится соответствие между событиями DTMF и символами DigitMap, описанными в 7.1.14.

DTMF	Символ события
d0	"0"
d1	"1"
d2	"2"
d3	"3"
d4	"4"
d5	"5"
d6	"6"
d7	"7"
d8	"8"
d9	"9"
da	"A" или "a"
db	"B" или "b"
dc	"C" или "c"
dd	"D" или "d"
ds	"E" или "e"
do	"F" или "f"

Версия: 1

Расширения: tonedet версия 1

Е.6.1 Свойства

Нет

Е.6.2 События

Е.6.2.1 DTMF Digits

Название события: DTMF Digits

EventID: Идентификаторы EventID задают тех же названий, что и идентификаторы SignalID, определенные в таблице в Е.5.3.

Описание: Генерируется, когда MG обнаруживает цифру.

Параметры EventDescriptor: Нет

Параметры дескриптора ObservedEvents: Нет

Е.6.2.2 DigitMap Completion Event

Название события: DigitMap Completion Event

EventID: ce (0x0004)

Описание: Генерируется, когда DigitMap завершается, как описано в 7.1.14.

Параметры EventsDescriptor: Нет

Параметры ObservedEventsDescriptor:

DigitString

Название параметра: DigitString

ParameterID: ds (0x0001)

Описание: часть текущей строки набора, описанной в 7.1.14, совпадающая с частью или всей последовательностью альтернативных событий, определенной в DigitMap.

Тип: Строка символов DigitMap (возможно, пустая), возвращаемая как quotedString

Дополнительный: Нет

Возможные значения: последовательность символов от "0" до "9", от "A" до "F", и модификатор большой продолжительности "Z".

Задано по умолчанию: Нет

Termination Method

Название параметра: Termination Method

ParameterID: Meth (0x0003)

Описание: показывает причину генерирования события. Процедуры см. в п. 7.1.14.

Тип: Перечисление

Дополнительный: Нет

Возможные значения:

"UM" (0x0001)	Однозначное соответствие
"PM" (0x0002)	Частичное совпадение, завершение при истечении таймера или несовпадающем событии
"FM" (0x0003)	Полное совпадение, завершение при истечении таймера или несовпадающем событии

Задано по умолчанию: Нет

Е.6.3 Сигналы

Нет

Е.6.4 Статистика

Нет

Е.6.5 Процедуры

Обработка DigitMap активируется, только если происходит активация дескриптора Events, содержащего событие завершения DigitMap, определенного в Е.6.2, и если данное событие завершения DigitMap содержит в запрашиваемых действиях поле EventDM, определенное в 7.1.9. Другие параметры такие, как KeepActive или вложенные дескрипторы Events или Signals, также могут присутствовать в дескрипторе Events и не влияют на активацию обработки DigitMap.

Е.7 Комплект Call Progress Tones Generator

Название комплекта: Call Progress Tones Generator Package

PackageID: cg 0x0007

Описание: В данном комплекте основные тональные сигналы прохождения вызова определяются как сигналы, и расширяются разрешенные значения параметра tl тонального сигнала воспроизведения в tonegen.

Версия: 2

Расширения: tonegen версия 2

Е.7.1 Свойства

Нет

Е.7.2 События

Нет

Е.7.3 Сигналы

Е.7.3.1 Dial Tone

Название сигнала: Dial Tone

SignalID: dt (0x0030)

Описание: Генерирует тональный сигнал готовности. Физические характеристики тонального сигнала готовности доступны на шлюзе.

Тип сигнала: TimeOut

Продолжительность: Предоставляется

Дополнительные параметры: Нет

Дополнительные значения: dt (0x0030) определяется как ID тонального сигнала для тонального сигнала воспроизведения

Определение остальных тональных сигналов в данном комплекте происходит точно таким же образом. Далее приводится таблица со всеми названиями сигналов и идентификаторами SignalID. Заметьте, что каждый тональный сигнал определяется как сигнал и как ID тонального сигнала, таким образом расширяя основной комплект генерирования тонального сигнала.

Название сигнала	SignalID/tone ID
Dial Tone	dt (0x0030)
Ringing Tone	rt (0x0031)
Busy Tone	bt (0x0032)
Congestion Tone	ct (0x0033)
Special Information Tone	sit (0x0034)
(Recording) Warning Tone	wt (0x0035)
Payphone Recognition Tone	prt (0x0036)
Call Waiting Tone	cw (0x0037)
Caller Waiting Tone	cr (0x0038)

E.7.4 Статистика

Нет

E.7.5 Процедуры

ПРИМЕЧАНИЕ – Требуемый набор идентификаторов тонального сигнала соответствует идентификаторам тонального сигнала, определенным в Рек. МСЭ-ТЕ.180/Q.35. Определение значений данных тональных сигналов см. в Рек. МСЭ-Т E.180/Q.35.

E.8 Комплект Call progress tones detection

Название комплекта: Call Progress Tones Detection Package

PackageID: cd (0x0008)

Описание: В данном комплекте определяются основные тональные сигналы обнаружения прохождения соединения. Данный комплект расширяет возможные значения ID тонального сигнала в событиях Start Tone Detected, End Tone Detected и Long Tone Detected.

Дополнительные значения

Значения ID тональных сигналов для Start Tone Detected, End Tone Detected и Long Tone Detected определяются при помощи тех же значений, что и значения ID тональных сигналов в комплекте cg (Комплект Call Progress Tones Generation).

Требуемый набор идентификаторов ID тональных сигналов соответствует Рек. МСЭ-Т E.180/Q.35. Определение значений данных тональных сигналов см. в Рек. МСЭ-Т.

Версия: 1

Расширения: tonedet версия 1

E.8.1 Свойства

Нет

E.8.2 События

События определяются как в комплекте Call Progress Tones Generator Package (cg) для тональных сигналов, перечисленных в таблице в E.7.3.

E.8.3 Сигналы

Нет

E.8.4 Статистика

Нет

Е.8.5 Процедуры

Нет

Е.9 Комплект Analog Line Supervision

Название комплекта: Analog Line Supervision Package

PackageID: al, 0x0009

Описание: В данном комплекте определяются события и сигналы для аналоговой линии.

Версия: 1

Расширения: Нет

Е.9.1 Свойства

Нет

Е.9.2 События

Е.9.2.1 On-hook

Название события: On-hook

EventID: on (0x0004)

Описание: Обнаруживает нахождение телефонной трубки в состоянии "трубка положена". При каждой активации дескриптора Events это запрашивает мониторинг события "трубка положена" и линия уже отключена, тогда MG должен работать согласно установке параметра "strict".

Параметры. EventsDescriptor:

Strict Transition

Назваени параметра: Strict Transition

ParameterID: strict (0x0001)

Описание: Показывает как происходит обнаружение события "трубка положена".

Тип: Перечисление

Дополнительный: Да

Возможные значения: "exact" (0x0000) Распознаваться будет только действительный переход телефонной трубки в состояние "трубка положена"

"state" (0x0001) Событие будет распознаваться, либо если обнаруживается переход состояния телефонной трубки, либо если телефонная трубка уже положена

"failWrong" (0x0002) Если телефонная трубка уже положена, команда завершается неудачно и передается сообщение об ошибке.

Задано по умолчанию: Exact

Параметры ObservedEventsDescriptor:

Initial State

Название параметра: Initial State

ParameterID: init (0x0002)

Описание: Причина сообщения о переходе в состояние "трубка положена". Возвращается, только если параметр Strict Transition установлен на "state".

Тип: Логическое выражение

Возможные значения: "True" Линия уже была в отключена, когда был активирован дескриптор Events, содержащий данное событие
"False" Был обнаружен действительный переход в состояние "трубка положена".

Задано по умолчанию: Нет

E.9.2.2 Off-hook

Название события: Off-hook

EventID: of (0x0005)

Описание: Обнаруживает, когда телефонная трубка снята. При каждой активации дескриптора Events это запрашивает мониторинг события "трубка снята", и линии уже подключена, затем MG должен работать согласно установке параметра "strict".

Параметры EventsDescriptor:

Strict Transition

Название параметра: Strict Transition

ParameterID: strict (0x0001)

Описание: Показывает как происходит обнаружение события "трубка снята".

Тип: Перечисление

Возможные значения: "exact" (0x0000) Нужно распознавать только действительный переход состояния телефонной трубки к подключению к линии

"state" (0x0001) Нужно распознавать событие либо, если обнаруживается переход состояния телефонной трубки, либо если телефонная трубка уже снята,

"failWrong" (0x0002) Если телефонная трубка уже снята, команда завершается неудачей и передается сообщение об ошибке.

Задано по умолчанию: Exact

Параметры ObservedEventsDescriptor:

Initial State

Название параметра: Initial State

ParameterID: init (0x0002)

Описание: Причина предоставления отчета по переходу в состояние "трубка снята". Возвращается, только если параметр Strict Transition установлен на "state".

Тип: Логическое выражение

Дополнительный: Да

Возможные значения: "True" Когда произошла активация дескриптора Events, содержащего данное событие, линия уже была подключена.
"False" Был обнаружен действительный переход в состояние подключения к линии.

Задано по умолчанию: Нет

E.9.2.3 Flashhook

Название события: Flashhook

EventID: fl (0x0006)

Описание: Обнаруживает мигание светового индикатора трубки. Мигание возникает, когда за положением "трубка положена" следует положение "трубка снята" в интервале между минимальной и максимальной длительностями.

Парметры EventsDescriptor:

Minimum Duration

Название параметра: Minimum Duration

ParameterID: mindur (0x0004)

Описание: Минимальное время между положениями "трубка положена" и "трубка снята", которое следует определять как мигание.

Тип: Целое число

Дополнительный: Да

Возможные значения: Любое положительное целое число в миллисекундах

Задано по умолчанию: Предоставляется.

Maximum Duration

Название параметра: Maximum Duration

ParameterID: maxdur (0x0005)

Описание: Максимальное время между состояниями "трубка положена" и "трубка снята", которое следует определять как мигание.

Тип: Целое число

Дополнительный: Да

Возможные значения: Любое положительное целое число в миллисекундах

Задано по умолчанию: Предоставляется.

Парметры ObservedEventsDescriptor: Нет

Е.9.3 Сигналы

Е.9.3.1 Ring

Название сигнала: Ring

SignalID: ri (0x0002)

Описание: Применяет звонок на линии

Тип сигнала: TimeOut

Продолжительность: Предоставляется

Дополнительные параметры:

Cadence

Название параметра: Cadence

ParameterID: cad (0x0006)

Описание: Отражает продолжительности чередующихся сегментов on и off, составляющих полный цикл звонка, начинающийся с on. Шлюзы MG с ограниченной функцией могут игнорировать значения cadence, которые они не могут генерировать.

Тип: Подсписок целых чисел

Дополнительный: Да

Возможные значения: Любое положительное целое число в миллисекундах

Задано по умолчанию: Предоставляется

Frequency

Название параметра: Frequency

ParameterID: freq (0x0007)

Описание: Частота звонка. Шлюзы MG с ограниченной функцией могут игнорировать частотные значения, которые они не могут генерировать.

Тип: Целое число

Дополнительный: Да

Возможные значения: Любое положительное целое число в Hz

Задано по умолчанию: Предоставляется.

Е.9.4 Статистика

Нет

Е.9.5 Коды ошибок

Код ошибки #: 540

Название: Unexpected initial hook state

Описание: Данная ошибка генерируется, когда MGC пытался запросить событие перехода состояния телефонной трубки с параметром Strict, установленным на "failWrong", а телефонная трубка уже находилась в состоянии, которое подразумевал переход.

Текст ошибки в дескрипторе Error: –

Комментарии: –

Е.9.6 Процедуры

Если MGC устанавливает дескриптор Events, содержащий событие перехода состояния телефонной трубки ("трубка положена" или "трубка снята" с параметром Strict (0x0001), установленным на "failWrong", а телефонная трубка уже находится в состоянии, которое подразумевает переход, выполнение команды, содержащей дескриптор Events, не удастся. В свой ответ MG должен включить Код ошибки 540 ("Неожиданное первоначальное состояние телефонной трубки").

Е.10 Комплект Basic Continuity

Название комплекта: Basic Continuity Package

PackageID: ct (0x000a)

Описание: В данном комплекте определяются события и сигналы для проверки целостности тракта связи. Проверка целостности тракта связи включает в себя предоставление функциональных возможностей либо кольцевой проверки линии связи, либо приемопередатчика.

Версия: 1

Расширения: Нет

Е.10.1 Свойства

Нет

Е.10.2 События

Е.10.2.1 Continuity Test Completion

Название события: Completion

EventID: cmp (0x0005)

Описание: Данное событие обнаруживает завершение проверки целостности тракта связи.

Параметры EventsDescriptor: Нет

Параметры ObservedEventsDescriptor

Continuity Test Result

Название параметра: Result

ParameterID: res (0x0008)

Описание: Показывает результат проверки целостности тракта связи.

Тип: Перечисление

Дополнительный: Нет

Возможные значения: "SUCCESS" (0x0001) Успешно
"FAILURE" (0x0000) Сбой

Задано по умолчанию: Нет

Е.10.3 Сигналы

Е.10.3.1 Continuity Test

Название сигнала: Continuity Test

SignalID: ct (0x0003)

Описание: Иницирует отправку тонального сигнала проверки целостности тракта связи на том завершении, к которому применяется.

Тип сигнала: TimeOut

Задано по умолчанию: Предоставляется

Дополнительные параметры: Нет

Е.10.3.2 Respond to Continuity Test

Название сигнала: Respond

SignalID: rsp (0x0004)

Описание: Данный сигнал используется для ответа на проверку целостности тракта связи. Более подробные объяснения см. в Е.10.5.

Тип сигнала: On/Off

Задано по умолчанию: Предоставляется

Дополнительные параметры: Нет

Е.10.4 Статистика

Нет

Е.10.5 Процедуры

Когда MGC желает иницировать проверку целостности тракта связи, он посылает MG команду, содержащую:

- Дескриптор Signals с сигналом ct; и
- Дескриптор Events, содержащий событие csp

После получения команды, содержащей сигнал ct и событие csp, MG иницирует тональный сигнал проверки целостности тракта связи для заданного завершения. Если до завершения сигнала обнаруживается ответный тональный сигнал, и удовлетворяются все остальные условия, должно быть сгенерировано событие csp со значением параметра результата – успешно. В остальных случаях должно быть сгенерировано событие csp со значением параметра результата – неудачно.

Если MGC желает, чтобы MG ответил на проверку целостности тракта связи, он посылает MG команду, содержащую дескриптор Signals с сигналом rsp. После получения команды с сигналом rsp, MG либо применяет кольцевую проверку линии связи, либо (для двухпроводных линий) ожидает получения тонального сигнала проверки целостности тракта связи. В случае кольцевой проверки вся входящая информация должна быть отослана обратно как исходящая информация. В случае двухпроводной линии в любое время получения соответствующего тонального сигнала проверки следует послать соответствующий тональный сигнал ответа. MGC определяет, когда удалять сигнал rsp.

Когда на завершении выполняется проверка целостности тракта связи, на данном завершении не должны быть активными устройства эхо или кодеки.

Выполнение обеспечения голосового тракта как часть проверки целостности тракта связи обеспечивается двусторонним соглашением между операторами сети.

ПРИМЕЧАНИЕ. – Примеры тональных сигналов и подробности процедуры проверки даются в пунктах 7 и 8/Q.724, 2.1.8/Q.764 и Рек. МСЭ-Т Q.1902.4.

E.11 Комплект Network

Название комплекта: Network Package

PackageID: nt (0x000b)

Описание: В данном комплекте определяются свойства завершений сети вне зависимости от типа сети. Типы сети включают в себя, не ограничиваются TDM, IP и ATM.

Версия: 1

Расширения: Нет

E.11.1 Свойства

E.11.1.1 Maximum Jitter Buffer

Название свойства: Maximum Jitter Buffer

PropertyID: jit (0x0007)

Описание: Данное свойство определяет максимальный размер буфера дрожания.

Тип: Целое число

Возможные значения: Любое положительное целое число в миллисекундах.

Задано по умолчанию: Предоставляется

Определено в: LocalControl Descriptor

Характеристики: Read/Write

E.11.2 События

E.11.2.1 Network Failure

Название события: Network Failure

EventID: netfail (0x0005)

Описание: Завершение генерирует данное событие после обнаружения сбоя, произошедшего из-за внешних или внутренних причин сети.

Параметры EventsDescriptor: Нет

Параметры ObservedEventsDescriptor:

Cause

Название параметра: Cause

ParameterID: cs (0x0001)

Описание: Данный параметр может быть включен вместе с событием сбоя для обеспечения диагностической информации по причине сбоя.

Тип: Строка

Дополнительный: Да

Возможные значения: любая текстовая строка

Задано по умолчанию: Нет

E.11.2.2 Quality Alert

Название события: Quality Alert

EventID: qualert (0x0006)

Описание: Данное событие позволяет МГ указать на потерю качества соединения сети. МГ может сделать это путем измерения потери пакетов, дрожания между двумя последовательными прибытиями пакетов, задержки распространения в сети, и указания этого при помощи процентного выражения потери качества.

Параметры EventsDescriptor:

Threshold

Название параметра: Threshold

ParameterID: th (0x0001)

Описание: Пороговая величина процента измеренной потери качества, вычисленная на основе предоставленного метода, учитывающего, например, потерю пакетов, случайные искажения и задержку. Событие запускается, когда вычисление превышает пороговую величину.

Тип: Целое число

Дополнительный: Да

Возможные значения: от 0 до 99

Задано по умолчанию: Предоставляется

Параметры ObservedEventsDescriptor:

Threshold

Название параметра: Threshold

ParameterID: th (0x0001)

Описание: Процент измеренной потери качества, вычисленный на основе предоставленного метода, учитывающего, например, потерю пакетов, случайные искажения и задержку.

Тип: Целое число

Дополнительный: Да

Возможные значения: от 0 до 99

Задано по умолчанию: Нет

E.11.3 Сигналы

Нет

Е.11.4 Статистика

Е.11.4.1 Duration

Название статистики: Duration

StatisticsID: dur (0x0001)

Описание: Предоставляет продолжительность времени существования завершения или его нахождения вне нулевого контекста.

Тип: Двойное число

Возможные значения: Любое положительное целое число в миллисекундах

Уровень: Любой

Е.11.4.2 Octets Sent

Название статистики: Octets Sent

StatisticID: os (0x0002)

Описание: Предоставляет число байтов, отправленное с завершения за время его существования или нахождения вне нулевого контекста. Данные байты представляют исходящий поток медиа за исключением всех служебных данных передачи. На уровне завершения данное значение равняется сумме исходящих потоков по всем потокам.

Для аналоговой среды передачи число байтов должно быть выставлено на ноль.

Тип: Двойное число

Возможные значения: любое 64-битное целое число от 0 и выше

Уровень: Любой

Е.11.4.3 Octets Received

Название статистики: Octets Received

StatisticID: or (0x0003)

Тип: Двойное число

Описание: Предоставляет число байтов, полученных по завершении или потоке за время существования завершения или его нахождения вне нулевого контекста. Данные байты представляют входящий поток медиа за исключением всех служебных данных передачи. На уровне завершения, данное значение равняется сумме входящих потоков по всем потокам.

Для аналоговой среды передачи, число байтов должно быть выставлено на ноль.

Возможные значения: любое 64-битное целое число от 0 и выше

Уровень: Любой

Е.11.5 Процедуры

Е.11.5.1 Процедуры для применения событий Network Failure и Quality Alert

Применение двух событий, определяемых данным комплектом зависит от:

- Определенного типа транспортного канала (например, TDM, Аналоговая линия, RTP, IP, AAL 2, AAL 1, и т. п.);
- Основной технологии уровня передачи (например, аналоговая, PDH, SDH, SONET, xDSL, 802.3, 802.11, и т. п.);
- Того, связано ли событие с ресурсами внешними или внутренними по отношению к MG.

Е.11.5.1.1 Общие для двух событий

Применение любого из данных событий требует понимания между MGC и MG. Если шлюзу MG и контроллеру MGC не достает последовательной точки зрения на свое значение, MGC не сможет эффективно действовать по ним.

Подробные рекомендации находятся вне сферы применения основного протокола H.248.1. Особое использование данных событий скорее рассматривается в спецификациях Профиля H.248.

Е.11.5.1.2 Характерные для события "Network Failure"

Для разграничения различных режимов сбоя можно использовать параметр Cause. При интерпретации события также помогает тип завершения.

Е.11.5.1.3 Характерные для события "Quality Alert"

Логика обнаружения должна основываться на одной или более метриках работы, относящихся к качеству обслуживания, из которых может быть получено значение, сравнимое с параметром Threshold.

Подробности основного измерения и процесса оценки (переменные наблюдения, относящиеся к QoS, алгоритмы фильтрования, длины периодов модернизации, и т. п.) могут быть определены в Профиле H.248.

Е.11.5.1.4 Связь с процедурами ServiceChange

События Network Failure или Quality Alert могут запустить процедуру ServiceChange на завершении(ях) – участнике(ах). Переход к OutOfService может быть инициирован MG или MGC, в зависимости от обстоятельств.

Е.12 Комплект RTP

Название комплекта: RTP Package

PackageID: rtp (0x000c)

Описание: Данный комплект используется для поддержки передачи данных мультимедиа на основе коммутации пакетов средствами протокола Real-time Transport Protocol (RTP) (RFC 3550).

Версия: 1

Расширения: nt версия 1

Е.12.1 Свойства

Нет

Е.12.2 События

Е.12.2.1 Payload Transition

Название события: Payload Transition

EventID: pltrans (0x0001)

Описание: Данное событие обнаруживает и уведомляет, когда происходит переход из одного формата полезной нагрузки RTP в другой формат.

Параметры EventsDescriptor: Нет

Параметры ObservedEventsDescriptor:

RTP Payload Type

Название параметра: rtppayload

ParameterID: rtppltype (0x01)

Описание: Формат полезной нагрузки в который произошел переход.

Тип: Подсписок перечисления

Дополнительный: Нет

Возможные значения: Необходимо определить метод кодирования путем использования одного или более действительных имен кодирования, определенных в Профиле RTP AV или зарегистрированных в IANA.

Задано по умолчанию: Нет

Е.12.3 Сигналы

Нет

Е.12.4 Статистика

Е.12.4.1 Packets Sent

Название статистики: Packets Sent

StatisticID: ps (0x0004)

Описание: Предоставляет число пакетов, отправленных с завершения или потока за время существования завершения или его нахождения вне нулевого контекста.

Тип: Двойное число

Возможные значения: любое 64-битное число от 0 и выше

Уровень: Любой

Е.12.4.2 Packets Received

Название статистики: Packets Received

StatisticID: pr (0x0005)

Описание: Предоставляет число пакетов, полученных на завершении или потоке за время существования завершения или его нахождения вне нулевого контекста.

Тип: Двойное число

Возможные значения: любое 64-битное целое число от 0 и выше

Уровень: Любой

Е.12.4.3 Packet Loss

Название статистики: Packet Loss

StatisticID: pl (0x0006)

Описание: Описывает текущую скорость потери пакетов в потоке RTP, определенную в RFC 3550. Потеря пакетов выражается в процентах: число пакетов, потерянных в интервале между двумя отчетами о получении, деленное на число пакетов, ожидаемых в данный интервал времени.

Тип: Двойное число

Возможные значения: 32-битное целое число и 32-битная дробь.

Уровень: Любой

Е.12.4.4 Jitter

Название статистики: Jitter

StatisticID: jit (0x0007)

Описание: Запрашивает текущее значение дрожания между двумя получениями пакетов в потоке RTP, определенное в RFC 3550. Дрожание измеряет изменение времени между двумя получениями для пакетов данных RTP.

Тип: Двойное число

Возможные значения: любое 64-битное целое число от 0 и выше

Уровень: Любой

E.12.4.5 Delay

Название статистики: Delay

StatisticID: delay (0x0008)

Описание: Запрашивает текущее значение задержки распространения пакетов в сети, выраженное в единицах измерения штампов времени. То же самое, что и среднее время задержки.

Тип: Двойное число

Возможные значения: любое 64-битное целое число от 0 и выше

Уровень: Любой

E.12.5 Процедуры

Когда RTCP связывается с потоком RTP, на RTCP не должно влиять свойство Mode H.248.1 в дескрипторе LocalControl.

Когда RTCP связывается с потоком RTP, а MG получает для этого потока пустой дескриптор Remote, MG должен прекратить поток RTCP и соответствующий поток RTP.

E.13 Комплект TDM Circuit

Название комплекта: TDM Circuit Package

PackageID: tdmc (0x000d)

Описание: Данный комплект может быть использован любым завершением, поддерживающим регулировку усиления и эхо. Первоначально данный комплект предназначался для использования на каналах TDM, однако его можно использовать более широко.

Новые версии или расширения данного комплекта должны учитывать использование не-TDM.

Версия: 1

Расширения: Комплект Network версия 1

E.13.1 Свойства

E.13.1.1 Echo Cancellation

Название свойства: Echo Cancellation

PropertyID: ec (0x0008)

Тип: Логическое выражение

Возможные значения: "True" (когда эхоподавление запрашивается)
"False" (когда эхоподавление выключено)

Задано по умолчанию: Предоставляется

Определено в: дескрипторе LocalControl

Характеристики: Read/Write

E.13.1.2 Gain Control

Название свойства: Gain Control

PropertyID: gain (0x000a)

Описание: Регулировка усиления или использование адаптации уровня сигнала и сокращения уровня шума используются для настройки уровня исходящего сигнала. Однако, например, для модемных соединений необходимо отключать данную функцию. Когда значение установлено на "automatic", завершение работает с автоматическая регулировка уровня (automatic level control (ALC)) в заданным на MG уровнем и направлением наружу.

Тип: Целое число

Возможные значения: Регулятор громкости задает усиление в децибелах (положительных или отрицательных), причем максимальное положительное целое число, 214748647 (0x7fffffff), зарезервировано для значения "automatic".

Задано по умолчанию: Предоставляется

Определено в: Дескрипторе LocalControl

Характеристики: Read/Write

E.13.2 События

Нет

E.13.3 Сигналы

Нет

E.13.4 Статистика

Нет

E.13.5 Процедуры

Нет

E.14 Комплект Segmentation

Название комплекта: Segmentation Package

PackageID: seg (0x00A3)

Описание: В данном комплекте определяются свойства для использования при выполнении сегментации на основе H.248 и несегментированной передачи.

Версия: 1

Расширения: root версия 2

E.14.1 Свойства

E.14.1.1 MG Segmentation Timer Value

Название свойства: MGSegmentationTimerValue

PropertyID: MGSegmentationTimerValue (0x0009)

Описание: Показывает время, в течение которого контроллеру MGC следует ожидать от MG получения любых ожидающих выполнения сегментов сообщения после получения SegmentationCompleteToken. Первоначальное значение – MGProvisionalResponseTimerValue, однако его можно изменить.

Тип: Целое число

Возможные значения: любое положительное целое число в миллисекундах

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.14.1.2 MGC Segmentation Timer Value

Название свойства: MGCSegmentationTimerValue

PropertyID: MGCSegmentationTimerValue (0x000A)

Описание: Показывает время, в течение которого шлюзу MG следует ожидать от MGC получения любых ожидающих выполнения сегментов сообщения после получения SegmentationCompleteToken. Первоначальное значение – MGCProvisionalResponseTimerValue, однако его можно изменить.

Тип: Целое число

Возможные значения: любое положительное целое число в миллисекундах

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.14.1.3 MG Maximum PDU Size

Название свойства: MGMaxPDUSize

PropertyID: MGMaxPDUSize (0x000B)

Описание: Показывает максимальный размер входящей единицы данных MG для транспортного протокола связи управления. Контроллеру MGC следует избегать создания сообщений, превышающих данный размер.

Тип: Целое число

Возможные значения: любое положительное целое число в байтах

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: ReadOnly

E.14.1.4 MGC Maximum PDU Size

Название свойства: MGCMaxPDUSize

PropertyID: MGCMaxPDUSize (0x000C)

Описание: Показывает максимальный размер входящей единицы данных MGC для транспортного протокола связи управления. Шлюзу MGC следует избегать создания сообщений, превышающих данный размер.

Тип: Целое число

Возможные значения: любое положительное целое число в байтах

Задано по умолчанию: Предоставляется

Определено в: TerminationState

Характеристики: Read/Write

E.14.2 События

Нет

E.14.3 Сигналы

Нет

Е.14.4 Статистика

Нет

Е.14.5 Коды ошибок

Код ошибки #: 459

Название: Segments not received

Описание: Данная ошибка показывает, что получатель сегментированного TransactionReply прекратил ожидание доставки всех сегментов по превышению лимита времени.

Текст ошибки в дескрипторе Error: Включаются номера недостающих сегментов.

Комментарии: –

Е.14.6 Процедуры

Поскольку некоторые виды передачи автоматически не сегментируют сообщения, превышающие максимальную величину единицы передачи, сообщения могут оказаться усеченными/ укороченными TransactionReply, отправителю следует посылать каждый TransactionReply в отдельном сообщении. Если же все еще остаются сообщения, превышающие длину MTU, отправитель может применить для TransactionReply следующие процедуры сегментации. Запросы TransactionRequest сегментировать нельзя.

При сегментировании ответа на транзакцию, отправитель должен проследить, что данное сообщение содержит полные команды и/ или действия. Каждый сегмент обозначается при помощи дополнительной информации о сегменте на уровне транзакций. Для каждого сегмента должен использоваться один и тот же transactionID, команды или завершенные действия не должны повторяться в последующих сегментах. Каждый сегмент должен иметь последовательный номер, начиная с 1 и до последнего сегмента, который обозначается включением SegmentationCompleteToken. В любом случае, сегментированные сообщения должны представлять собой синтаксически правильные структурные элементы. Получатель отвечает на каждый сегмент в порядке очередности, используя ответ сегментации. Поскольку сегменты представляют собой завершенные сообщения сами по себе, для обработки любого отдельного сегмента получающему объекту не нужно ожидать остальных сегментов.

После получения последнего сегмента, получатель должен определить, получил ли он все сегменты сообщения. Если да, то получатель должен ответить TransactionResponseAcknowledgement.

Если получатель не получил все сегменты, он запустит таймер и будет ждать любых недостающих сегментов. Длина таймера должна предоставляться получателю, однако она должна быть такой же как и длина ProvisionalResponseTimerValue (см. в Е.2.1) другого объекта. Если недостающие сегменты не прибывают до истечения таймера, получатель должен ответить Кодом Ошибки 459 ("Сегменты не получены").

Пример 1:

```
Отправитель:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}
Получатель:      !/3 [12.34.56.79]:2944 SM=1/1
Отправитель:      !/3 [12.34.56.78]:2944
P=1/2{C=1{AV=term3{...}}, C=2{AV=term4{...}}}
Получатель:      !/3 [12.34.56.79]:2944 SM=1/2
Отправитель:      !/3 [12.34.56.78]:2944 P=1/3/{C=3{AV=term5{...}}}
Получатель:      !/3 [12.34.56.79]:2944 SM=1/3/#
Получатель:      !/3 [12.34.56.79]:2944 K=1
```

Пример 2:

```
Отправитель:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}
Получатель:      !/3 [12.34.56.79]:2944 SM=1/1
Отправитель:      !/3 [12.34.56.78]:2944 P=1/4/{C=3{AV=term5{...}}}
Получатель:      !/3 [12.34.56.79]:2944 SM=1/4/#
/* Таймер сегментации истекает */
Получатель:      !/3 [12.34.56.79]:2944 ER=459{"2,3"}
```

E.15 Комплект Notification Behaviour

Название комплекта: Notification Behaviour Package

PackageID: nb (0x009a)

Описание: Комплект имеет функциональную возможность, которая позволяет MG по запросу MGC регулировать отправку команд Notify. Данный комплект взаимодействует с флагом NotifyBehaviour, описанным в 7.1.9. В данной версии комплекта описывается режим регулирования, основанный на регулировании процентного отношения. Другие типы регулирования являются предметом дальнейшего изучения.

Версия: 1

Расширения: Нет

E.15.1 Свойства

E.15.1.1 Notification Regulation

Название свойства: Notification Regulation

PropertyId: notreg (0x0001)

Описание: Показывает процентное отношение уведомлений на MG, которые следует сдерживать. Значение 0% обозначает отсутствие сдерживания. Значение 100% обозначает, что сдерживаться должны все уведомления. Данное свойство должно определяться только на завершении Root. Когда на определенном событии отправляется "Regulated Notify", MG использует свойство Notification Regulation для определения верного процентного отношения Notifies для регулирования (т. е. сдерживания).

Тип: Целое число

Возможные значения: 0-100

Задано по умолчанию: Нет

Определено в: TerminationState

Характеристики: Read/Write

E.15.2 События

Нет

E.15.3 Сигналы

Нет

E.15.4 Статистика

Нет

E.15.5 Процедуры

E.15.5.1 NotifyBehaviour

Алгоритм NotifyBehaviour позволяет MGC управлять тем, как он будет принимать команды Notify.request от MG.

E.15.5.1.1 NeverNotify

Могут быть случаи, когда MGC не заинтересован в получении команды Notify.request. Например: когда обнаружение данного события является важным для запуска вложенного сигнала или события, но MGC не нужно, чтобы его информировали. Установка параметра NotifyBehaviour на "NeverNotify" в событии гарантирует, что при обнаружении события, уведомление MGC не отправляется; однако, все вложенные сигналы или события выполняются. NeverNotify не имеет никакого отношения к свойству Notification Regulation (nb/notreg), и события, отмеченные таким образом, не включаются в расчеты регулирования.

E.15.5.1.2 NotifyImmediate

MGC может желать получать уведомления о важных событиях таких, как событие 'Emergency Call', даже если регулируются некоторые другие события. Это достигается путем установки параметра NotifyBehaviour на "NotifyImmediate" для определенного события. При обнаружении событий данный режим уведомления задан по умолчанию. NotifyImmediate не имеет никакого отношения к свойству Notification Regulation (nb/notreg), и события, отмеченные таким образом, не включаются в расчеты регулирования.

E.15.5.1.3 RegulatedNotify

E.15.5.1.3.1 Percentage Based Regulated Notify

Когда параметр NotifyBehaviour установлен на "NotifyRegulated" для определенного события, решение о том, посылать ли MGC уведомление о любом появлении определенного события, определяется свойством Notification Regulation (nb/notreg) на завершении Root. Значение данного свойства определяется и устанавливается MGC. Например, уровень регулирования может согласовываться с уровнем перегрузки MGC. В этом случае MG будет ответственным за определение верного процентного отношения сдерживаемых команд Notify. среди событий, отмеченных как "NotifyRegulated" по всему MG.

MG может реализовывать любой алгоритм достижения установки процентного отношения Notification Regulation. Алгоритм следует выбрать так, что риск синхронизации между различными MG был минимальным. Рекомендуемый алгоритм приводится ниже:

Для заданного процентного отношения регулирования (например, 10%) для событий, имеющих флаг RegulatedNotify, шлюзу MG следует сделать следующее:

- 1) Сдерживать число уведомлений (например, Notification 1).
- 2) Принять несколько последующих уведомлений, пока не будет достигнуто процентное отношение NotificationRegulation (например, принять 9 уведомлений).
- 3) Повторять пункты 1 и 2 пока не будет изменен Regulation Percentage
- 4) Во избежание проблем с синхронизацией, шлюзу MG следует начинать со случайной позиции в последовательности, предлагаемой выше.

Если параметр NotifyBehaviour для события выставлен на "RegulatedNotify", а свойство Notification Regulation выставлено на ноль, эффект будет тот же, как если бы событие было выставлено на "ImmediateNotify", что является заданным по умолчанию режимом уведомления о событии.

Во избежание неверных состояний алгоритм NotifyBehaviour следует использовать с осторожностью. Например, если уведомление о событии "трубка снята" подавляется MG, тогда MGC не сможет предоставить верный ответ (например, сигнала тонального вызова, тонального сигнала перегрузки). Данного неверного режима работы можно избежать посредством использования алгоритма вложенного сигнала или вложенных событий для возвращения соответствующего ответа без задействования MGC в то же самое время. MGC может установить, чтобы происходил запуск обоих вложенных дескрипторов Signals или Events при обнаружении и принятии события, отмеченного "NotifyRegulated" (т. е. уведомление не сдерживается), или запуск отдельных регулируемых вложенных дескрипторов Signals или Events, когда происходит обнаружение и регулирование событий (т. е. уведомление сдерживается).

ПРИМЕЧАНИЕ. – Даже в вышеупомянутом случае MG все еще не знает, из чего состоит соединение. Он просто выполняет предварительно запрограммированную задачу, а не функциональную возможность управления соединением.

E.15.5.2 Пример сценария

При условии сценария, при котором MGC и MG завершают аналоговые линии, MGC/MG могут работать в различном режиме на основе того, находится ли MGC в состоянии перегрузки или нет, и порождает ли аналоговая линия (ALN) аварийный вызов или нет. Различные режимы работы показаны в Таблице E.1 ниже:

Таблица Е.1/Н.248.1 – Потенциальные режимы работы MGC/MG Overload

Режим работы МG		1) Тип соединения/ контекста	
		Аварийный вызов	Не-аварийный вызов
Задача: "Управление сигнализации соединения аналоговой линии (ALS)" = трафик сигнализации управления соединением для оконечных устройств POTS (ALN; интерфейсы аналоговой линии)			
2) MGC Load State ПРИМЕЧАНИЕ. – подразумевается модель из двух состояний.	Не-состояние перегрузки (например, низкая, средняя, высокая загрузка)	I) MGC: Обычная обработка ALS с <i>высшим</i> приоритетом	II) MGC: Обычная обработка ALS с <i>низшим</i> приоритетом
	Состояние перегрузки	III) MGC: Обычная обработка ALS ==> Сигнализация управления соединением, направленная MGC в "обычном режиме Н.248"	IV) MGC: Программируемое управление локальным ALS ==> MGC <i>не</i> включается ==> Уступчивый протокол соединения отклоняется посредством "тонального сигнала перегрузки"

Е.15.5.2.1 Режим работы, квадранты I и II

На МG не требуется особого NotifyBehaviour, поскольку MGC не перегружен и может принимать все сообщения (уведомления) от МG. Однако, если MGC определил тип соединения, он может установить на МG приоритетность, используя соответствующий атрибут Context.

Е.15.5.2.2 Режим работы, квадранты III и IV

Если MGC определил, что он находится в состоянии перегрузки, он может отослать всем или выбранной группе завершений в нулевом контексте следующую команду.

Пример сообщения:

```
Transaction=1234 {
  Context = - {
    Modify = aln/* {          ; выберите завершения, как требуется
      Events = 1234 {
        al/of {
          RegulatedNotify {
            Embed {
              Signals {cg/dt},
; Если снятие трубки регулируется, применить тональный вызов.
              Events = 1235 {
                xdd/xce {
                  DigitMap = PriorityDialPlan1,
                  bc = 20,
                  mp = enhanced,
                  ImmediateNotify
                },
; В случае аварийного вызова генерировать уведомление для MGC
                xdd/xce {
                  DigitMap = NonPriorityDialPlan1,
                  bc = 20,
                  mp = enhanced,
```


Приложение F

Процедуры ServiceChange

F.1 Введение

В данном приложении описывается использование процедур ServiceChange, когда на MG или MGC происходят определенные события. Данное приложение служит для разъяснения процедур ServiceChange, описанных в 7.2.8 и 11. там, где между данным приложением и данной Рекомендацией возникают противоречия, процедуры основной части данной Рекомендации обладают приоритетом над процедурами, описанными в данном приложении.

Событиями, приводящими к отправке команды ServiceChange, описанной в данном приложении, являются:

- 1) MG Registration – MG регистрируется на MGC, и устанавливается связь управления.
- 2) MG Re-Registration – MG регистрируется на MGC после того, как MGC попросил его об этом.
- 3) MG Service Cancellation – MG информирует MGC о том, что завершает работу или MG полностью неисправен.
- 4) MG Service Restoration – MG восстановился после сбоя и информирует об этом MGC.
- 5) MG Redundant Takeover – Дополнительный MG берет на себя обязанности в случае сбоя или технического обслуживания основного MG.
- 6) MG Lost Communication – MG информирует MGC о том, что ранее он потерял связь с MGC, однако сейчас данная связь восстановлена.
- 7) MG Capability Change – MG информирует MGC о том, что возможности завершений или MG целиком изменились.
- 8) MGC Initiated MG Re-Registration – MGC информирует MG о том, что шлюзу MG следует перерегистрироваться на том же MGC или другом MGC.
- 9) MGC Initiated Service Restoration – MGC восстановился после сбоя и приказывает MG перевести важные завершения в первоначальное состояние.
- 10) MGC Initiated Service Cancellation – MGC информирует MG о том, что завершения или MG целиком прекращают работу.
- 11) MGC Redundant Failover – На основном MGC происходит сбой, и он направляет MG для регистрации на определенный дополнительный MGC.

На рисунке F.1 показаны данные запускающие события ServiceChange, объект, на котором появляется событие, и то, какие пары событий коррелируют. На рисунке F.2 показана модель состояний, которой подчиняются MG и завершения на MG.

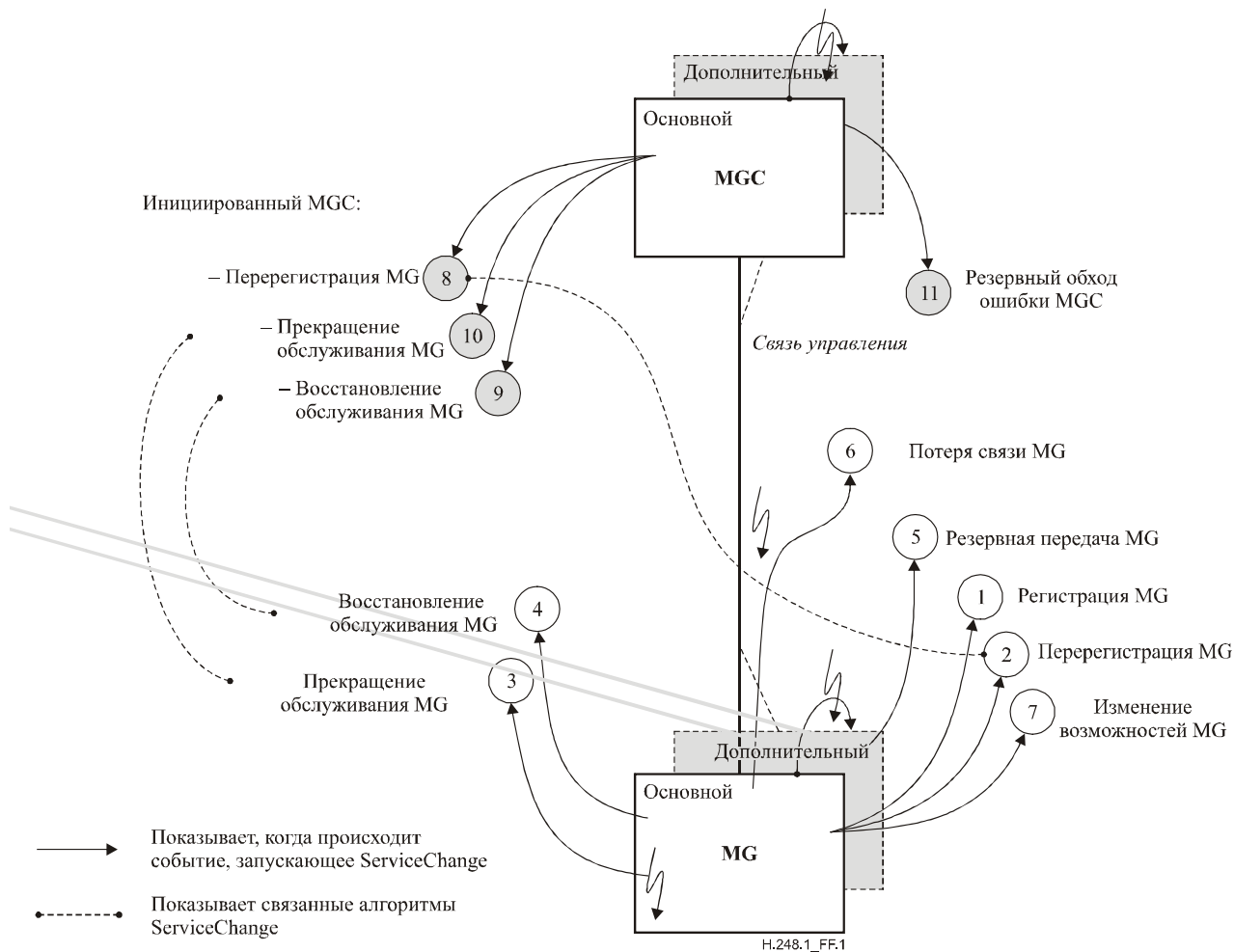


Рисунок F.1/Н.248.1 – Запускающие события для ServiceChange

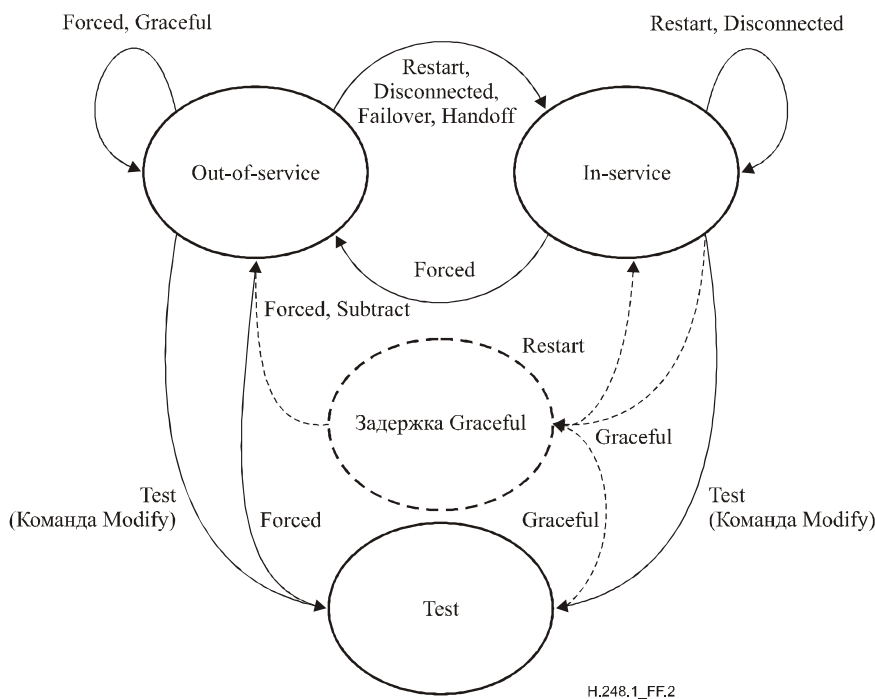


Рисунок F.2/Н.248.1 – Простая модель состояний ServiceChange

F.2 Определение связи управления

Связь управления представляет собой взаимосвязь, при которой MGC управляет MG. Связь управления устанавливается посредством регистрации. Связь управления прекращается если MG неисправен, был успешно передан альтернативному MGC, или успешно перешел к альтернативному MGC при обходе отказа. MG должен иметь не более одной связи управления в каждый момент времени, кроме случаев, когда физический MG разделен на один или более виртуальных MG. В случае в виртуальном MG, не более одной связи управления должен иметь каждый виртуальный. Связи управления могут создавать только шлюзы MG. Связи управления, определенные здесь, применяются только в рамках контекста сигнализации управления медиашлюзом. Данное определение не следует путать с определениями связей управления в контексте передачи (TCP, UDP, и т. п.).

Использование команд ServiceChange на завершении Root влияет только на виртуальный MG, к которому принадлежит завершение Root. На другие виртуальные MG на том же физическом MG они не влияют.

F.3 События, приводящие к процедурам ServiceChange

F.3.1 Регистрация MG (MG Registration)

MG может зарегистрироваться одним из трех способов:

- Объявить о своем первоначальном присутствии, или "регистрации", используя ServiceChangeMethod "Restart" с ServiceChangeReasons 901 ("Cold Boot" – "холодный запуск") or 902 ("Warm Boot" – "теплый запуск"),
- Показать, что изменяет свою связь путем ServiceChangeMethod "Failover" и ServiceChangeReason 909 ("MGC Impending Failure" – "предстоящий сбой MGC") или 908 ("MG Impending Failure" – "предстоящий сбой MGC"),
- Показать, что ему была дана команда изменить свою связь при помощи ServiceChangeMethod "Handoff" и ServiceChangeReason 903 ("MGC Directed Change" – "изменение, направленное MGC") (F.3.11).

Регистрация начинается, когда от MG к MGC посылается первоначальная команда ServiceChange, и завершается, когда ответ на команду приходит от MGC без альтернативного адреса или ошибки, а MGC и MG согласовали ServiceChangeProfile. Связь управления создается после окончание регистрации. Регистрация всегда происходит на завершении Root. Любая связь управления между MGC и MG прекращается при начале новой регистрации. Все команды из предыдущей связи управления нужно проигнорировать.

При регистрации используется ServiceChangeProfile. Подробно см. в F.5.5.

При регистрации используется ServiceChangeVersion. Подробно см. в F.5.6.

F.3.2 Пере-regистрация MG (MG Re-Registration)

Пере-regистрация MG происходит в двух случаях:

- 1) Когда MGC делает запрос MG о пере-regистрации (F.3.8). Затем MG должен послать ServiceChange с ServiceChangeMethod "Handoff", ServiceChangeReason 903 ("MGC Directed Change" – "изменение направленное MGC") и назначанный "ServiceChangeMgcID".
- 2) Когда MGC инициирует восстановление обслуживания (F.3.9). MG должен пере-regистриваться путем отправки ServiceChange с ServiceChangeMethod "Restart".

Пере-regистрация завершается, когда на команду приходит ответ от MGC, не содержащий альтернативного адреса или ошибки, MGC и MG согласовывают ServiceChangeProfile. Если от основного MGC не приходит ответа, MG должен следовать процедурам F.3.1. Рекомендуются ServiceChangeReasons 903 ("MGC Directed Change") или 909 ("MGC Impending Failure").

Связь управления устанавливается после завершения регистрации. Регистрация всегда происходит на завершении Root. Любая существующая связь управления между MGC и MG прекращается при начале новой регистрации. Все команды из предыдущей связи управления нужно игнорировать.

При пере-regистрации используется ServiceChangeProfile. Подробно см. в F.5.5.

При пере-regистрации используется ServiceChangeVersion. Подробно см. в F.5.6.

F.3.3 Прекращение обслуживания MG (MG Service Cancellation)

Для того чтобы поместить MG в "OutOfService", MG посылает команду ServiceChange со значением ServiceChangeMethod – "Forced" или "Graceful" на завершении Root. Подробно см. в F.4.1.1.

Для того, чтобы поместить завершение или группу завершений в "OutOfService", MG посылает ServiceChange со значением ServiceChangeMethod – "Forced" или "Graceful" на данное завершение (я). Подробно – см. F.4.1.2 и F.4.1.3.

ServiceChangeDelay показывает продолжительность времени до аннулирования обслуживания. Подробно см. в F.5.3.

Для отмены ранее отправленного (и подтвержденного) Graceful для всего MG, MG посылает команду ServiceChange на завершение Root со значением ServiceChangeMethod – "Restart" и ServiceChangeReason 918, "Cancel Graceful". MG должен продолжать работать, а обслуживание всех ранее необслуживаемых завершений возобновляется, если иначе не сигнализировано MG. В случае, если Cancel Graceful было получено после истечения таймера Delay, предполагается, что MG пере-regистривался, как если бы он послал Cold Boot.

Для отмены ранее отправленного (и подтвержденного) Graceful на завершении или группе завершений, MG посылает ServiceChange со значением ServiceChangeMethod – "Restart" и ServiceChangeReason – "Cancel Graceful" на данном завершении (яx). Завершение должно продолжать работать. В случае, если завершение уже прекратила работу, оно будет возвращено к работе, как при любом ServiceChange Restart.

F.3.4 Восстановление обслуживания MG (MG Service Restoration)

Событие MG Service Restoration происходит, когда MG возвращается к работе после сбоя или выполнения технического обслуживания.

При восстановлении обслуживания, инициированного MG, MG посылает MGC команду ServiceChange, объявляя о том, что произошел перезапуск, или что он желает заново согласовать

версию протокола профиля H.248. В таком случае используются ServiceChangeMethod "Restart" и ServiceChangeReason 900 ("Service Restored"). ServiceChangeReason покажет в каких действиях со стороны MGC может возникнуть необходимость.

F.3.5 Резервная передача MG (MG Redundant Takeover)

Существует два вида резервной передачи MG:

1) Передача, инициированная основным MG:

Когда MG заканчивает работу и желает передать обработку определенному дополнительному MG, он посылает MGC команду ServiceChange Command со значением ServiceChangeMethod "Failover" и ServiceChangeReason – 908 ("MG Impending Failure"). MGC прекратит посылать сообщения основному MG и завершит связь управления. Затем дополнительный MG должен послать команду ServiceChange со значением ServiceChangeMethod "Restart" и ServiceChangeReason – 900 ("Service Restored"). Затем MGC должен установить новую связь управления с дополнительным MG.

2) Передача, инициированная дополнительным MG:

Когда дополнительная единица MG обнаруживает аварийное отключение основного MG из-за сбоя или технического обслуживания, а основной MG не может уведомить MGC о сбое, дополнительный MG должен послать MGC команду ServiceChange со значением ServiceChangeMethod "Failover" и ServiceChangeReason – либо 919 ("Warm Failover"), либо 920 ("Cold Failover"), соответствующим образом. Применяются процедуры F.3.4.

MGC должен игнорировать все попытки резервной передачи со стороны неизвестных MG. MGC должен знать, посредством предоставления или других средств, что данный дополнительный MG, объявляющий о передаче, связан с неисправным основным MG и поэтому авторизован выполнить передачу для основного MG.

Если дополнительный MG известен и авторизован выполнить передачу для основного MG, MGC должен попытаться связаться с основным MG и определить, функционирует ли он. Это можно проделать при помощи пустой команды AuditValue или другого подходящего метода (см. п. 11.5). Если основной MG отвечает, MGC должен отклонить попытку передачи. Если основному MG не удастся ответить, MGC должен предположить, что на основном MG произошел сбой, и принять теплую или холодную обработку отказа (warm or cold failover). Когда MGC принимает теплую или холодную передачу, связь управления с основным MG удаляется и устанавливается новая связь управления с дополнительным MG.

F.3.6 Потеря связи MG (MG Lost Communication)

Когда MG обнаруживает потерю и последующее повторное установление связи с MGC, MG посылает MGC команду ServiceChange со значением ServiceChangeMethod – "Disconnected" в текущей связи управления. Если MGC не отвечает, тогда MG посылает команду ServiceChange со значением ServiceChangeMethod – "Failover" и ServiceChangeReason 909 ("MGC Impending Failure") каждому MGC из своего списка по очереди до тех пор, пока ему не удастся успешно установить новую связь управления, или пока список контроллеров MGC не закончится.

Если MGC в исходной связи управления отвечает на команду ServiceChange, связь управления продолжается, не прерываясь, и все команды обрабатываются так, как будто потери связи не было. В противном случае, когда MG посылает ServiceChange новому MGC, исходная связь управления завершается, и все команды из предыдущей связи управления игнорируются. Новая связь управления устанавливается, когда новый MGC отвечает на ServiceChange, завершая тем самым регистрацию.

Если у MG заканчивается список контроллеров MGC, а связь управления не была установлена успешно, MG выжидает случайное количество времени и пытается снова осуществить регистрацию на контроллерах MGC из списка, начиная с MGC из исходной связи управления. MG будет посылать ServiceChange со значением ServiceChangeMethod – "Disconnected" контроллеру MGC из исходной связи управления каждый раз, когда MG будет пытаться связаться с ним. MG посылает ServiceChange со значением ServiceChangeMethod – "Failover" всем остальным MGC.

Контроллеры MGC, получающие команды ServiceChange со значением ServiceChangeMethod – "Disconnected" должны проверить MG и определить, не произошло ли несоответствия состояний из-за потери сообщений.

В данных сценариях рекомендуется ServiceChangeReason 900 ("Service Restored"), хотя отдельные сценарии могут требовать других кодов ServiceChangeReason.

F.3.7 Изменение возможностей MG (MG Capability Change)

Для того, чтобы объявить об изменении своих возможностей, MG посылает ServiceChange с соответствующими ServiceChangeMethod и ServiceChangeReason – 916 ("Packages Change") или 917 ("Capabilities Change"). Контроллеру MGC следует проверить MG и определить новые возможности MG. Для работающего MG, посылающего команду ServiceChange со значением ServiceChangeMethod "Restart", во время изменения возможности связь управления продолжается, не прерываясь, и считается, что MG не подвергался системному перезапуску.

F.3.8 Перерегистрация MG, инициированная MGC (MGC Initiated MG Re-Registration)

MGC может сделать запрос MG о перерегистрации путем отправки команды ServiceChange на Root со значением ServiceChangeMethod – "Handoff", ServiceChangeReason 903 ("MGC Directed Change") и своим собственным ServiceChangeMgcID (то есть, текущего MGC).

Действия, предпринимаемые MG, см. в F.3.2.

F.3.9 Восстановление обслуживания, инициированное MGC (MGC Initiated Service Restoration)

Для запроса о том, чтобы MG выполнил собственный перезапуск, MGC посылает MG команду ServiceChange со значением ServiceChangeMethod – "Restart" и соответствующей ServiceChangeReason такой, как 900 ("Service Restored") или 901 ("Cold Boot"). MG должен установить новую связь управления в соответствии с F.3.2, используя обозначенную ServiceChangeReason.

Для восстановления обслуживания завершения или группы завершений, MGC посылает на данное завершение (я) ServiceChange со значением ServiceChangeMethod – "Restart". Действия, предпринимаемые MGC, см. в F.4.1.2 и F.4.1.3.

Задержка ServiceChangeDelay обозначает продолжительность времени до осуществления восстановления обслуживания. Подробно см. в F.5.3.

F.3.10 Прекращение обслуживания, инициированное MGC (MGC Initiated Service Cancellation)

F.3.10.1 Завершение Root

Для того, чтобы сделать MG необслуживаемым ("OutOfService"), MGC посылает на завершение Root команду ServiceChange со значением ServiceChangeMethod – "Forced" или "Graceful". Соответствующие ServiceChangeReason могут включать среди прочих 905 ("Termination taken out of service"). Действия, предпринимаемые MGC, см. в F.4.1.1.

ServiceChangeDelay показывает период времени, в конце которого произойдет прекращение обслуживания. Подробно см. в F.5.3.

Для аннулирования ранее отправленного (и подтвержденного) Graceful для всего MG, MGC посылает на завершение Root команду ServiceChange со значением ServiceChangeMethod – "Restart" и ServiceChangeReason 918 ("Cancel Graceful"). MG должен оставаться в работе, а все завершения, ранее установленные на OutOfService, возвращаются к работе, если иначе не сигнализировано MG. В случае, если получение Cancel Graceful происходит после истечения таймера Delay, MG должен предоставить MGC в качестве отчета Код ошибки 502 ("Not Ready"). Для возвращения к работе шлюзу MG требуется перерегистрироваться.

F.3.10.2 Физические завершения

Для того, чтобы сделать необслуживаемыми группу завершений, MGC посылает на данное завершение(я) команду ServiceChange со значением ServiceChangeMethod – "Forced" или "Graceful". Соответствующие ServiceChangeReason среди прочих могут включать в себя 904 ("Termination malfunctioning"), 905 ("Termination taken out of service"), 906 ("Loss of lower layer connectivity"), или 907 ("Transmission failure"). Действия, предпринимаемые MGC, см. в F.4.2 и F.4.3.

ServiceChangeDelay показывает период времени, по истечении которого произойдет прекращение обслуживания. Подробно см. в F.5.3.

Для аннулирования ранее отправленного (и подтвержденного) Graceful для завершения или группы завершений, MGC посылает на данное завершение(я) команду ServiceChange со значением ServiceChangeMethod – "Restart" и ServiceChangeReason 918 ("Cancel Graceful"). Завершение должно оставаться в работе. В случае, если завершение уже перешло в режим OutOfService, оно должно быть возвращено к работе так, как оно было бы возвращено при помощи любого ServiceChange.

F.3.10.3 Непродолжительные завершения (Ephemeral Terminations)

MGC не должен использовать ServiceChange для прекращения обслуживания непродолжительных завершений. Для удаления завершения достаточно использовать вычитание завершения из контекста.

F.3.11 Резервная передача MGC (MGC Redundant Failover)

Когда MGC, находящийся в связи управления, сталкивается с ситуацией технического обслуживания или сбоя и должен прекратить работу, он может направить MG определенному дополнительному MGC путем отправки команды ServiceChange со значением ServiceChangeMethod – "Handoff", ServiceChangeReason – 903 ("MGC Directed Change") и адресом нового MGC в параметре ServiceChangeMgcID. Связь управления завершается после получения ответа на команду от MG. Затем MG посылает команду ServiceChange заданному MGC со значением ServiceChangeMethod – "Handoff" и ServiceChangeReason – 903 ("MGC Directed Change"). Новая связь управления устанавливается после получения ответа на команду от MGC.

В случае, если заданный MGC отклоняет попытку передачи, или MGC не может послать команду ServiceChange со значением ServiceChangeMethod – Handoff из-за сбоя, MG должен перейти к процедурам для потери связи, описанным в F.3.6, начиная с MGC, заданного как основной.

F.4 Описание элемента ServiceChange

F.4.1 ServiceChangeMethod

В данном пункте описывается режим работы ServiceChangeMethod на различных типах завершений. Данный пункт состоит из:

- Завершение Root;
- Физические завершения;
- Непродолжительные завершения (Ephemeral terminations).

F.4.1.1 Режим работы ServiceChange Method на завершении Root

Команда ServiceChange на Root имеет различные результаты в зависимости от ServiceChangeMethod. Результаты каждого ServiceChangeMethod описаны ниже:

- 1) Restart – Посылается MG для объявления о том, что он был перезапущен, желает заново согласовать версию протокола, профиль, или объявляет о смене возможностей. ServiceChangeReason показывает какие действия, возможно, нужно будет предпринять MGC. Если отправляется со стороны MGC, MG должен осуществить собственный перезапуск, используя указанную ServiceChangeReason.
- 2) Forced – При отправлении со стороны MG, MG показывает, что он прекращает работу немедленно. Связь управления завершается MG после получения ответа на команду со стороны MGC. При отправлении со стороны MGC, MG должен прекратить работу (OutOfService), и завершить связь управления после отправки ответа на команду. Заметьте,

что MGC не может вернуть MG обратно к работе, даже если он может сделать запрос о прекращении работы MG (OutOfService). MG инициирует регистрацию на MGC для установления новой связи управления и указания на восстановление связи, также, как и для любой другой регистрации. ServiceChangeDelay в ServiceChangeMethod "Forced" не работает.

- 3) Graceful – При отправлении со стороны MG, ServiceChangeMethod "Graceful" показывает, что MG прекращает работу по завершении периода ServiceChangeDelay. Связь управления завершается в конце периода ServiceChangeDelay. При отправлении со стороны MGC, MG должен прекратить работу (OutOfService) и завершить связь управления в конце периода ServiceChangeDelay. Значение ServiceChangeDelay равно нулю или отсутствие задержки ServiceChangeDelay показывает, что MG должен прекратить работу (OutOfService) и завершить связь управления, когда последний контекст будет удален посредством вычитания его завершений, и то, что MGC не должен добавлять новые соединения. Шлюзу MG следует установить свойство ServiceStates завершения Root на значение "OutOfService" по завершении задержки ServiceChangeDelay или удалении всех завершений из активных контекстов (какое бы из событий ни было первым). Для удаления ранее отправленного (и подтвержденного) ServiceChange со значением ServiceChangeMethod – "Graceful", объект, инициирующий ServiceChangeMethod Graceful посылает команду ServiceChange со значением ServiceChangeMethod Restart и ServiceChangeReason 918 ("Cancel Graceful").
- 4) Failover – При отправлении со стороны шлюза MG контроллеру MGC, не находящемуся в данной связи управления, MG показывает, что на его текущем MGC произошел сбой, и что он пытается зарегистрироваться с получателем команды. При отправлении со стороны шлюза MG контроллеру MGC, находящемуся в текущей связи управления, основной MG показывает, что дополнительный MG принял полномочия от основного MG, давшего сбой. В любом случае, предыдущая связь управления завершается после отправки команды ServiceChange, и новая связь управления между MG и новым MGC, или MGC и новым MG устанавливается после получения ответа на команду. Контроллеры MGC не должны посылать команды ServiceChange со значением ServiceChangeMethod "Failover".
- 5) Handoff – При отправлении со стороны MGC, MGC показывает, что управление MG передается новому MGC. Данное действие завершает текущую связь управления после получения ответа на команду. При отправлении от MG к MGC, ServiceChangeMethod "Handoff" показывает, что MG пытается установить новую связь управления в соответствии с командой о переадресации вызова, полученной от MGC в предыдущей связи управления. При отправлении от MG к MGC, переадресация вызова является регистрацией, новая связь управления устанавливается после получения ответа на команду. MG не должен использовать Handoff без команды со стороны MGC.
- 6) Disconnected – При отправлении со стороны MG, показывает, что связь в текущей связи управления была потеряна, но в данный момент снова установлена. Происходит обновление текущей связи управления. Контроллеры MGC не должны отправлять команды ServiceChange со значением ServiceChangeMethod "Disconnected".

F.4.1.2 Режим работы ServiceChange Method на физических завершениях

Пока MG находится в активной связи управления отправление команды ServiceChange на физическом завершении имеет различные результаты в зависимости от ServiceChangeMethod. Результаты каждого метода описаны ниже:

- 1) Restart – При отправлении со стороны MG, MG объявляет, что на завершении(ях) произошел перезапуск, или объявляет об изменении возможностей. ServiceChangeReason показывает, какие действия, возможно, нужно будет предпринять MGC. При отправлении со стороны MGC, MG должен перезапустить завершение(ия) используя обозначенную ServiceChangeReason.
- 2) Forced – При отправлении со стороны MG, показывает, что завершение прекращает работу немедленно. При отправлении со стороны MGC, MG должен прекратить обслуживание завершения незамедлительно. В обоих случаях, значение параметра ServiceStates должно быть установлено на "OutOfService", а MGC отвечает за очистку всех контекстов или ресурсов, связанных с данным завершением. ServiceChangeDelay на ServiceChangeMethod "Forced" не работает.

- 3) Graceful – При отправлении со стороны MG, показывает, что завершение(я) прекращает работу (OutOfService) после ServiceChangeDelay. При отправлении со стороны MGC, MG должен прекратить обслуживание данного завершения(ий) в конце периода ServiceChangeDelay. Значение свойства ServiceStates должно быть выставлено на "OutOfService" после истечения периода ServiceChangeDelay или когда данное завершение (ия) будет удалено из активного контекста (какое бы из событий ни было первым), а MGC отвечает за очистку всех контекстов или ресурсов, связанных с данным завершением(ями). Значение ServiceChangeDelay равно нулю или отсутствующая ServiceChangeDelay обозначает, что обслуживание завершения должно прекратиться при удалении его из контекста путем вычитания. MGC не должен использовать обозначенные завершения для соединения пока Graceful не будет аннулирован, или пока обслуживание завершения не возобновлено при помощи последующей команды ServiceChange. Для аннулирования ранее отправленной (и подтвержденной) ServiceChange со значением ServiceChangeMethod – "Graceful", объект, инициирующий Graceful посылает команду ServiceChange Command со значением ServiceChangeMethod "Restart" и ServiceChangeReason 918 ("Cancel Graceful").
- 4) Failover – ServiceChangeMethod "Failover" не должен использоваться с завершениями не-Root.
- 5) Handoff – ServiceChangeMethod "Handoff" не должен использоваться с завершениями не-Root.
- 6) Disconnected – The ServiceChangeMethod "Disconnected" не должен использоваться с завершениями не-Root.

F.4.1.3 Режим работы ServiceChange Method на непродолжительных завершениях

Пока MG находится в активной связи управления отправление команды ServiceChange на непродолжительном завершении имеет различные результаты в зависимости от ServiceChangeMethod. Результаты каждого метода описаны ниже:

- 1) 1) Restart – При отправлении со стороны MG, MG объявляет, что на завершении(ях) произошел перезапуск, или объявляет об изменении возможностей. ServiceChangeReason показывает, какие действия, возможно, нужно будет предпринять MGC. MGC не должен отправлять ServiceChangeMethod "Restart" для непродолжительных завершений.
- 2) Forced – При отправлении со стороны MG, показывает, что завершение прекращает работу немедленно. За вычитание завершения отвечает MGC. MGC не должен отправлять ServiceChangeMethod "Forced" для непродолжительных завершений. ServiceChangeDelay на ServiceChangeMethod "Forced" не работает.
- 3) 3) Graceful – При отправлении со стороны MG, показывает, что завершение(я) прекращает работу (OutOfService) после ServiceChangeDelay. За вычитание завершения(ий) по истечении ServiceChangeDelay отвечает MGC. MGC не должен отправлять ServiceChangeMethod "Graceful" для непродолжительных завершений. Значение ServiceChangeDelay равно нулю показывает, что завершение должно быть разрушено при удалении из контекста путем вычитания. Шлюзу MG следует установить значение свойства ServiceStates завершения на "Out of Service" по истечении задержки ServiceChangeDelay или удаления завершения из активного контекста (какое бы из событий не было первым). Для аннулирования ранее отправленной (и подтвержденной) команды ServiceChange со значением ServiceChangeMethod – "Graceful", объект, инициирующий Graceful отправит команду ServiceChange со значением ServiceChangeMethod Restart и ServiceChangeReason – 918 "Cancel Graceful".
- 4) Failover – ServiceChangeMethod "Failover" не должен использоваться с завершениями не-Root.
- 5) Handoff – ServiceChangeMethod "Handoff" не должен использоваться с завершениями не-Root.
- 6) Disconnected – The ServiceChangeMethod "Disconnected" не должен использоваться с завершениями не-Root.

F.5 Использование параметров ServiceChange

F.5.1 ServiceChangeMethod

См. использование, описанное в F.4.

F.5.2 ServiceChangeReason

ServiceChangeReasons позволяет получающей стороне изменять свой режим работы для соответствия определенной ситуации. Например, если MG отправил MGC ServiceChangeReason 901 ("Cold Boot") при выполнении Restart, MGC мог предположить, что MG потерял все свои состояния и поэтому не будет выполнять задачи по технической поддержке и проверке для оценивания и очистки MG. И приведения его в приемлемое состояние.

В таблице F.1 показано, с какими ServiceChangeMethod могут быть отправлены определенные ServiceChangeReason.

Таблица F.1/Н.248.1 – Построение соответствий между ServiceChangeMethod и ServiceChangeReason

SC Reason	SC Method						Описание
	Restart	Forced	Graceful	Disconnected	Failover	Handoff	
900	X			Только Root Только MG			Service Restored
901	Только Root						Cold Boot
902	Только Root						Warm Boot
903						Только Root	MGC Directed Change
904		X	X				Term Malfunction
905		X	X				Term Taken OOS
906		X	X				Loss of lower layer connectivity
907		X	X				Transmission failure
908		Только Root Только MG	Только Root Только MG		Только Root Только MG		MG Impending Failure
909					Только Root Только MG		MGC Impending Failure
910	Только MG	X	X				Media Capability Failure
911	Только MG	X	X				Modem Capability Failure
912	Только MG	X	X				Mux Capability Failure
913	Только MG	X	X				Signal Capability Failure
914	Только MG	X	X				Event Capability Failure
915		X	X				State Loss
916	X			Только Root	Только		Packages Change

Таблица F.1/Н.248.1 – Построение соответствий между ServiceChangeMethod и ServiceChangeReason

SC Reason	SC Method						Описание
	Restart	Forced	Graceful	Disconnected	Failover	Handoff	
				Только MG	Root Только MG		
917	X			Только Root Только MG	Только Root Только MG		Capability Change
918	X						Cancel Graceful
919					Только Root Только MG		Warm Failover
920					Только Root Только MG		Cold Failover

F.5.3 ServiceChangeDelay

ServiceChangeDelay используется для предоставления задержки перед тем, как команда ServiceChange вступит в силу, и будет изменено свойство ServiceStates шлюза MG или завершения(ий). Ответ на команду ServiceChange отправляется так, как если бы выполнение ее происходило до получения сообщения, однако в действительности MG или завершение(я) не изменяют своего состояния до истечения ServiceChangeDelay. Значение ServiceChangeDelay равное нулю аналогично не использованию ServiceChangeDelay, за исключением случаев его использования совместно с ServiceChangeMethod "Graceful". Действие ServiceChangeDelay на различные ServiceChangeMethod – см. в F.4.

F.5.4 ServiceChangeAddress

Использование параметра ServiceChangeAddress описано в 7.2.8. Использование ServiceChangeAddress не рекомендуется. Если данный параметр присутствует, его можно использовать только с командами ServiceChange на завершении Root, а новые транзакции должны отсылаться на новый указанный адрес и/ или номер порта. Ответы должны присылаться обратно на адрес, с которого пришел соответствующий запрос.

F.5.5 ServiceChangeProfile

Параметр ServiceChangeProfile позволяет MGC и MG согласовывать профиль Н.248 для использования в связи управления. MGC все еще может проверять MG для определения других возможностей, поддерживаемых MG. Отправка ServiceChangeProfile должна осуществляться только в командах регистрации и перерегистрации.

F.5.6 ServiceChangeVersion

ServiceChangeVersion используется для согласования версии протокола H.248, используемой MGC и MG. ServiceChangeVersion является обязательной при первоначальной регистрации, и должна отправляться во всех других командах регистрации, где необходимо согласование версии протокола. Нельзя посылать ServiceChangeVersion в нерегистрирующих командах. Процедуры регистрации см. в F.3.1.

F.5.7 ServiceChangeMgcID

MGC может отправить данный параметр в команде ServiceChange, направленной завершению Root. После получения во время попытки регистрации, MG должен попытаться зарегистрироваться на MGC по заданному адресу. При получении в команде ServiceChange Handoff от основного контроллера MGC шлюза MG, MG должен применить процедуры, описанные в 11.5.

F.5.8 TimeStamp

Использование дополнительного параметра TimeStamp описано в 7.2.8. Параметр TimeStamp не оказывает влияния на выполнение команд ServiceChange, но может быть полезным получателю команды ServiceChange для других целей таких, как координация рассылки счетов и расчетов по времени.

F.6 ServiceChange в сравнении с TerminationState

Свойство ServiceStates в дескрипторе TerminationState содержит текущее значение состояния завершения. Свойство ServiceStates может иметь три значения: "InService", "OutOfService", и "Test".

Для установления значения свойства ServiceStates Property на или Test или с Test контроллер MGC должен использовать только команды Modify. Все остальные команды Modifies свойства ServiceStates являются нарушением протокола, и должны вызвать код ошибки 401 ("Protocol Error"). Для изменения состояния с "InService" на "OutOfService" или наоборот, нужно использовать команду ServiceChange.

Для завершений, находящихся в состоянии Test, MG должен послать MGC только ServiceChanges со значением ServiceChangeMethod – "Forced" или "Graceful". Затем последующие команды ServiceChange могут установить значение свойства ServiceState завершения на "InService".

Получателям следует сделать соответствующие попытки принять команды ServiceChange, однако в некоторых случаях отклонение команды неизбежно. Например, если MG отправил команду ServiceChange для того, чтобы возобновить обслуживание завершения, но у MGC нет ресурсов, распределенных для обслуживания данного завершения из-за недостатка подготовки к работе или по другим причинам, тогда MGC может отклонить команду. Если получающий объект отклоняет команду ServiceChange, состояние завершения остается неизменным. Отправитель может предпочесть выждать некоторое время и попытаться снова, или дожаться, пока получатель попытается изменить состояние завершения. Запросы о прекращении обслуживания завершения следует всегда выполнять.

В Таблице F.2 показано, какие команды могут быть использованы для выполнения изменения состояния на завершении.

Таблица F.2/H.248.1 – Команды изменения состояния и их результаты

Текущее состояние	Новое состояние	Команда	MGC разрешено?	MG разрешено?
InService	Test	ServiceChange	Невозможно	Невозможно
		Modify	Да	Невозможно
InService	OutOfService	ServiceChange	Да, физическое и только Root	Да
		Modify		Невозможно
OutOfService	InService	ServiceChange	Да, но не на Root	Да
		Modify	Нет	Н возможно

Таблица F.2/Н.248.1 – Команды изменения состояния и их результаты

OutOfService	Test	ServiceChange	Невозможно	Невозможно
		Modify	Да	Невозможно
Test	InService	ServiceChange	Нет	Нет
		Modify	Да	Невозможно
Test	OutOfService	ServiceChange	Нет	Да
		Modify	Да	Невозможно

Дополнение I

Примеры потоков соединений

Все применяющие Н.248.1 должны внимательно прочитать нормативную часть данной Рекомендации перед тем, как применять ее. Не следует использовать примеры из данного дополнения в качестве отдельных объяснений того, как создавать сообщения протокола.

В примерах данного дополнения для кодирования дескрипторов Local и Remote используется SDP. SDP определяется в RFC 2327. Если между SDP в примерах и RFC 2327 возникают какие-либо несоответствия, для уточнения следует обратиться к RFC. Используемые аудиопрофили определены в RFC 1890, другие профили зарегистрированы в IANA. Например, G.711 А-типа в SDP называется PCMA, и является назначенным профилем 0. G.723.1 называется G723 и является профилем 4; H.263 называется H263 и является профилем 34. См. также <http://www.iana.org/assignments/rtp-parameters>.

I.1 Соединение частный шлюз – частный шлюз

В данном примере иллюстрируется использование элементов протокола для установления соединения Частный шлюз – Частных шлюз по сети на основе IP. Для простоты в данном примере подразумевается, что оба частных шлюза, участвующих в соединении, контролируются одним контроллером медиашлюза.

I.1.1 Программирование завершений аналоговой линии на частных шлюзах для резервного режима работы

Далее иллюстрируется инициирование API со стороны контроллера медиашлюза и медиашлюза с целью программирования завершений данного сценария на резервный режим работы. Как исходящие, так и входящие медиашлюзы имеют резервные завершения аналоговой линии, запрограммированные на поиск событий инициирования соединения (т.е. снятия трубки) путем использования команды Modify с соответствующими параметрами. Нулевой контекст используется для обозначения того, что завершения еще не входят в контекст. Завершение Root используется для обозначения MG целиком вместо какого-либо завершения в рамках MG.

В данном примере, MG1 имеет IP адрес 124.124.124.222, MG2 – 125.125.125.111, а MGC – 123.123.123.4. Для всех порт Megaco – 55555.

1) MG регистрируется на MGC, используя команду ServiceChange:

от MG1 – MGC:

```
MEGACO/1 [124.124.124.222]
Transaction = 9998 {
    Context = - {
        ServiceChange = ROOT {Services {
```

```

        Method=Restart, Version=3,
        ServiceChangeAddress=55555, Profile=ResGW/1}
    }
}

```

2) MGC посылает ответ:

```

от MGC - MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
    Context = - {ServiceChange = ROOT {
        Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}

```

3) MGC программирует завершение в нулевом контексте. Значение TerminationID – A4444, streamID – 1, requestID в дескрипторе Events Descriptor – 2222. MID является идентификатором отправителя данного сообщения; в данном случае, это IP адрес и порт [123.123.123.4]:5555. Значение свойства Mode для данного потока установлено на SendRecv. "al" является комплектом контроля аналоговой линии. Подразумевается, что дескрипторы Local и Remote предоставляются.

```

От MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 9999 {
    Context = - {
        Modify = A4444 {
            Media { Stream = 1 {
                LocalControl {
                    Mode = SendRecv,
                    tdmc/gain=2,; in dB,
                    tdmc/ec=on
                },
            },
        },
        Events = 2222 {al/of {strict=state}}
    }
}
}

```

Скрипт плана набора можно загрузить в MG предварительно. Его задачей будет ожидать снятия трубки, включить тональный вызов и начать собирать цифры DTMF. Однако, в данном примере мы используем DigitMap, который вступает в силу после обнаружения снятия трубки (шаг 5) ниже).

Заметьте, что вложенный дескриптор Events можно использовать для объединения шагов 3) и 4) с шагами 8) и 9), исключая шаги 6) и 7).

4) MG1 принимает Modify посредством данного ответа:

```

от MG1 - MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 9999 {
    Context = - {Modify = A4444}
}

```

5) Между MG2 и MGC происходит аналогичный обмен, результатом которого является резервное завершение, называемое A5555.

1.1.2 Сбор цифр инициатора и инициирование завершения

Нижеследующее основывается на ранее указанных условиях. Далее иллюстрируются транзакции от контроллера медиашлюза и иницирующего медиашлюза (MG1) для того, чтобы провести иницирующее завершение (A4444) через все стадии сбора цифр, необходимого для инициирования соединения с завершающим медиашлюзом (MG2).

- 6) MG1 обнаруживает событие снятия трубки от Пользователя 1 и предоставляет отчет о нем контроллеру медиашлюза при помощи команды Notify.

```
от MG1 - MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10000 {
  Context = - {
    Notify = A4444 {ObservedEvents =2222 {
      19990729T22000000:al/of(init=OFF)}}
  }
}
```

- 7) И происходит подтверждение команды Notify.

```
от MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10000 {
  Context = - {Notify = A4444}
}
```

- 8) MGC модифицирует завершение так, чтобы оно воспроизводило тональный вызов, выполняло поиск цифр согласно Dialplan0, и производило поиск события "трубка положена" в настоящий момент.

```
От MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10001 {
  Context = - {
    Modify = A4444 {
      Events = 2223 {
        al/on(strict=state), dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
(0| 00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxxx|9011x.)}
      }
    }
  }
}
```

- 9) И происходит подтверждение команды Modify.

```
От MG1 - MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10001 {
  Context = - {Modify = A4444}
}
```

- 10) Затем MG1 накапливает цифры так, как они набираются Пользователем 1. Тональный вызов прекращается при обнаружении первой цифры. Когда из собранных цифр составляется соответствующая комбинация в отношении к запрограммированному текущему плану набора для A4444, контроллеру медиашлюза отправляется еще одна команда Notify.

```
От MG1 - MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10002 {
  Context = - {
    Notify = A4444 {ObservedEvents =2223 {
      19990729T22010001:dd/ce{ds="916135551212",Meth=UM}}}
  }
}
```

11) И происходит подтверждение команды Notify.

```
От MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10002 {
    Context = - {Notify = A4444}
}
```

12) Затем контроллер анализирует цифры и определяет, что необходимо установить соединение между MG1 и MG2. В новый контекст на MG1 добавляются как завершение TDM A4444, так и завершение RTP. Значение Mode – RecvOnly, поскольку значения дескриптора Remote еще не заданы. Предпочтительные кодеки расположены в порядке предпочтительности выбора MGC.

```
От MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10003 {
    Context = $ {
        Add = A4444,
        Add = $ {
            Media {
                Stream = 1 {
                    LocalControl {
                        Mode = RecvOnly,

                        nt/jit=40; in ms
                    },
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
                }
            }
        }
    }
}
```

ПРИМЕЧАНИЕ. – MGC устанавливает предпочтительные значения параметра в виде последовательностей блоков SDP в дескрипторе Local. MG заполняет дескриптор Local в ответе Reply.

13) MG1 подтверждает новое завершение в заполняет Local IP адрес и порт UDP. Он также выбирает кодек на основе предпочтений MGC в Local. MG1 устанавливает значение порта RTP на 2222.

```
MEGACO/3 [124.124.124.222]:55555
Reply = 10003 {
    Context = 2000 {
        Add = A4444,
        Add=A4445{
            Media {
                Stream = 1 {
                    Local {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
                }
            }
        }
    }
}
```



```

}
    }; профиль RTP G.723.1 - 4
}
}
}
}
}

```

16) Теперь шлюзу MG1 необходимо предоставить вышеупомянутые IPAddr и UDPport.

```

От MGC - MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10005 {
    Context = 2000 {
        Modify = A4444 {
            Signals {cg/rt}
        },
        Modify = A4445 {
            Media {
                Stream = 1 {
                    Remote {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
    }; профиль RTP для G.723.1 - 4
}
}
}
}
}

```

```

MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10005 {
    Context = 2000 {Modify = A4444, Modify = A4445}
}

```

17) Теперь два шлюза соединены, и Пользователь 1 слышит сигнал обратного вызова. Теперь MG2 ожидает, когда Пользователь 2 поднимет трубку, и будет установлено двустороннее соединение.

```

От MG2 к MGC:
MEGACO/3 [125.125.125.111]:55555
Transaction = 50005 {
    Context = 5000 {
        Notify = A5555 {ObservedEvents =1234 {
            19990729T22020002:al/of(init=off)}}
    }
}

```

```

От MGC к MG2:
MEGACO/3 [123.123.123.4]:55555
Reply = 50005 {
    Context = - {Notify = A5555}
}

```


От MGC к MG2:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 50006 {
  Context = 5000 {
    Modify = A5555 {
      Events = 1235 {al/on(strict=state)},
      Signals; to turn off ringing
    }
  }
}
```

От MG2 к MGC:

```
MEGACO/3 [125.125.125.111]:55555
Reply = 50006 {
  Context = 5000 {Modify = A4445}
}
```

18) Изменить режим MG1 на SendRecv, и прекратить сигнал обратного вызова.

От MGC – MG1:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 10006 {
  Context = 2000 {
    Modify = A4445 {
      Media {
        Stream = 1 {
          LocalControl {
            Mode=SendRecv
          }
        }
      }
    },
    Modify = A4444 {
      Signals
    }
  }
}
```

От MG1 к MGC:

```
MEGACO/3 [124.124.124.222]:55555
Reply = 10006 {
  Context = 2000 {Modify = A4445, Modify = A4444}}
```

19) MGC принимает решение провести проверку завершения RTP на MG2.

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 50007 {
  Context = 5000 {AuditValue = A5556{
    Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
  }
}
```

20) MG2 отвечает.

```
MEGACO/3 [125.125.125.111]:55555
Reply = 50007 {
  Context = 5000 {
    AuditValue = A5556 {
      Media {
        TerminationState { ServiceStates = InService,
          Buffer = OFF },
        Stream = 1 {
          LocalControl { Mode = SendRecv,

```

```

        nt/jit=40 },
    Local {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
a=ptime:30
    },
    Remote {
v=0

o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
    } } },
    Events,
    Signals,
    DigitMap,
    Packages {nt-1, rtp-1},
    Statistics { rtp/ps=1200,; отсылаемые пакеты
                nt/os=62300,; отсылаемые байты
                rtp/pr=700,; получаемые пакеты
                nt/or=45100,; получаемые байты
                rtp/pl=0.2,; % потеря пакетов
                rtp/jit=20,
                rtp/delay=40 }; среднее время задержки
    }
}
}

```

- 21) Когда MGC получает сигнал "трубка положена" от одного из шлюзов MG, он завершает соединение. В данном примере первым кладет трубку пользователь на MG2.

От MG2 к MGC:

```

MEGACO/3 [125.125.125.111]:55555
Transaction = 50008 {
    Context = 5000 {
        Notify = A5555 {ObservedEvents =1235 {
            19990729T24020002:al/on(init=off)}
        }
    }
}

```

От MGC к MG2:

```

MEGACO/3 [123.123.123.4]:55555
Reply = 50008 {
    Context = - {Notify = A5555}
}

```

- 22) Теперь MGC отправляет обоим шлюзам MG команду Subtract для снятия соединения. Здесь показаны только команды Subtract направленные MG2. Каждое завершение собирает свой собственный набор статистики. MGC может не испытывать необходимости в возвращении обоих наборов. A5555 является физическим завершением, а A5556 является завершением RTP.

От MGC к MG2:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 50009 {
  Context = 5000 {
    Subtract = A5555 {Audit{Statistics}},
    Subtract = A5556 {Audit{Statistics}}
  }
}
```

От MG2 к MGC:

```
MEGACO/3 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
    Subtract = A5555 {
      Statistics {
        nt/os=45123,; отправляемые байты
        nt/or=45123,; получаемые байты
        nt/dur=40000; в миллисекундах
      }
    },
    Subtract = A5556 {
      Statistics {
        rtp/ps=1245,; отправляемые пакеты
        nt/os=62345,; отправляемые байты
        rtp/pr=780,; получаемые пакеты
        nt/or=45123,; получаемые байты
        rtp/pl=10,; % потерянные пакеты
        rtp/jit=27,
        rtp/delay=48,; среднее время задержки
        nt/dur=38000; в миллисекундах
      }
    }
  }
}
```

- 23) Теперь MGC настраивает как MG1, так и MG2 на готовность к обнаружению события снятия трубки. (См. шаг 1). Заметьте это может быть заданным по умолчанию состоянием завершения, находящегося в нулевом контексте. В таком случае отправки сообщения от MGC к MG не требуется. Когда завершение возвращается в нулевой контекст, оно возвращается к заданным по умолчанию значениям завершения для данного завершения.

Дополнение II

Шаблон комплекта Н.248

Новые и обновленные комплекты Н.248 следует определять, используя нижеследующий шаблон. Он имеет формат Рекомендации МСЭ-Т. Редакторам не из МСЭ следует как минимум использовать структуры в пункте 6 данного шаблона. Ключевые слова комплектов Н.248 обозначены жирным шрифтом. Участки для заполнения обозначены символом "<>". Более подробную информацию о том, как создать новый комплект, см. в пункте 12.

ПРИМЕЧАНИЕ. – Коды ошибок, характерные для определенных комплектов, требуют регистрации в IANA. Присвоение кодов ошибок должно проходить по правилам, описанным в 4.1/Н.248.8, "*Assigning Error Codes*".

Рекомендация МСЭ-Т Н.248.<xxx>

Протокол управления шлюзом: <xxx> Комплекты

1 Сфера применения

<Сфера применения данного комплекта>

2 Справочная литература

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

- Рекомендация МСЭ-Т Н.248.1 (<xx/xxxx>), *Протокол управления шлюзом: Версия<x>*
- <Другая справочная литература>

3 Термины и определения

В данной Рекомендации используются следующие термины и определения:

<термины и определения>

4 Сокращения

В данной Рекомендации используются следующие сокращения:

<сокращения>

5 Соглашения

<соглашения>

6 <Название комплекта>

Название комплекта: <название>

PackageID: <PackageID "текст", двоичный <0х????>>

Описание: <описание>

Версия: <версия>
Расширения: <расширенный PackageID и версия>

6.1 Свойства

6.1.1 <Название свойства>

Название свойства: <название>
PropertyID: <текстовый ID, двоичный ID (0x????)>
Описание: <описание>
Тип: <тип>
Возможные значения: <значения>
Задано по умолчанию: <значение>
Определено в: <Local, Remote, LocalControl, TerminationState, ContextAttribute>
Характеристики: <ReadOnly, Read/Write>

6.2 События

6.2.1 <Название события>

Название события: <название события>
EventID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>

6.2.1.1 Параметры EventsDescriptor

6.2.1.1.1 <Название параметра>

Название параметра: <название>
ParameterID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>
Тип: <типы>
Дополнительный: <да/нет>
Возможные значения: <значения>
Задано по умолчанию: <значение>

6.2.1.2 Параметры ObservedEventsDescriptor

6.2.1.2.1 <Название параметра>

Название параметра: <название>
ParameterID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>
Тип: <типы>
Дополнительный: <да/нет>
Возможные значения: <значения>
Задано по умолчанию: <значение>

6.3 Сигналы

6.3.1 <Название сигнала>

Название сигнала: <название>
SignalID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>
Тип сигнала: <тип>
Продолжительность: <продолжительность>

6.3.1.1 Дополнительные параметры

6.3.1.1.1 <Название параметра>

Название параметра: <название>
ParameterID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>
Тип: <типы>
Дополнительный: <да/нет>
Возможные значения: <значения>
Задано по умолчанию : <значение>

6.4 Статистика

6.4.1 <Название статистики>

Название статистики: <название>
Statistic ID: <(текстовый ID), (двоичный ID (0x????))>
Описание: <описание>
Тип: <тип>
Возможные значения: <значения>
Уровень: <Termination, Stream, оба>

6.5 Коды ошибок

6.5.1 <Название кода ошибки>

Код ошибки #: <число>
Название: <название>
Определение: <определение>
Текст ошибки в дескрипторе Error: <возвращаемый текст ошибки>
Комментарий: <комментарий>

6.6 Процедуры

<Процедуры, связанные с комплектом>

Дополнение III

Шаблон определения профиля Н.248

Новые профили Н.248 следует определять, используя следующий шаблон профилей. Он имеет формат Рекомендации МСЭ. Редакторам не из МСЭ следует, как минимум, использовать структуры в пункте 6 ниже. Если шаблон формата не используется, редакторы профилей должны гарантировать, что их профиль охватывает все заголовки и пункты данного дополнения.

Заголовки в нижеследующей структуре представляют элементы, которые можно считать дополнительными. Для определений профиля определенный элемент Н.248.1 может не использоваться несмотря на то, что он является обязательным в Н.248.1. Данные элементы также включены в нижеследующую структуру. Не перечисленные элементы следует считать обязательными для протокола Н.248.1. В шаблоне профиля, приведенном ниже, элементы могут быть определены как "дополнительные" и "обязательные". "Дополнительный" означает, что для отправителя или получателя включение данного элемента в сообщение является необязательной возможностью. Если получающий объект получает дополнительный элемент, который он не реализует, он должен отправить Код ошибки 501 ("Not Implemented") согласно п. 6.2.3.

Редактору следует предоставить письменное описание в каждом из нижеследующих подпунктов, если в них далее разъясняется режим работы Н.248.1. Например, если использование команды Move (см. п. 6.8.4) ограничивается несколькими определенными типами завершений, это следует указать.

Текст, набранный курсивом, следует удалить.

<текст> в скобках следует заполнить.

Рекомендация МСЭ-Т <xxx>

<Название профиля>

1 Сфера применения

<Сфера применения данного профиля>

2 Справочная литература

<Справочная литература>

3 Термины и определения

В данной Рекомендации используются следующие термины и определения:

<термины и определения>

4 Сокращения

В данной Рекомендации используются следующие сокращения:

<сокращения>

5 Соглашения

<соглашения>

6 Описание профиля

6.1 Идентификация профиля

Название профиля:	<название из 1–64 символов>
Версия:	<версия 1–99>

Название и версия профиля, передаваемого в команде *ServiceChange*.

6.2 Резюме

<Описание>

Описание того, что представляет собой данный профиль.

6.3 Версия протокола управления шлюзом

<Номер версии>

Минимальная версия H.248 требующаяся для поддержки данного профиля. Она должна основываться на поддержке синтаксиса базы, а не на произвольном присвоении версии. Это относится к *ServiceChangeVersion* в 6.8.8.

6.4 Модель соединения

<Описать в словах и диаграммах>

Описание конфигураций завершений, разрешенных в контексте.

Максимальное число контекстов:	<Целое число>
Максимальное число завершений на контекст:	<Целое число>
Разрешенные комбинации типов окончаний в контексте:	<например, Context[a](IP,TDM), Context[b](TDM,AAL 2), и т. п.>

6.5 Атрибуты контекста

Атрибут контекста	Поддерживается	Поддерживаемые значения
Топология	<Да/Нет>	См. 6.7.8
Индикатор приоритетности	<Да/Нет>	<1-15>
Индикатор аварийного вызова	<Да/Нет>	NA
Индикатор IEPS	<Да/Нет>	NA
Дескриптор ContextAttribute	<Да/Нет>	Если "Да", детали поддерживаемых атрибутов см. п. 6.8.9.
Параметр ContextIDList	<Да/Нет>	NA

Поддерживается ли операция выбора атрибута контекста AND/OR?

Атрибут контекста AND/OR	<Да/Нет>	<AND/OR/BOTH>
--------------------------	----------	---------------

6.6 Завершения

6.6.1 Названия завершений

<Структура TerminationID>

Идентифицировать завершение, связанное с физическими, непродолжительными мультиплексными завершениями.

6.6.2 Мультиплексные завершения

Поддерживаются ли мультиплексные завершения?	<Да/Нет>
--	----------

Если да, то:

Поддерживаемые типы мультиплексирования:	<Н.221, Н.223, Н.226, V.76, N × 64К>
Максимальное число завершений, связанных в мультиплекс:	<Целое число>

Используются ли мультиплексные завершения? Если да, опишите.

6.7 Дескрипторы

6.7.1 Дескриптор Stream

Максимальное число потоков на тип завершения	<Тип завершения>	<Целое число>
--	------------------	---------------

Если более одного:

Конфигурация потока:	<Опишите разрешенные конфигурации. Разрешено ли более одного аудиопотока? И т. п.>
----------------------	--

6.7.1.1 Дескриптор LocalControl

Используются ли свойства *ReserveGroup* и *ReserveValue*?

Если не общие, перечислите соответствующие типы завершений и потоков.		Тип завершения	Тип потока
Используется <i>ReserveGroup</i> :	<Да/Нет>	<Тип>	<Тип>
Используется <i>ReserveValue</i> :	<Да/Нет>	<Тип>	<Тип>

Какие значения *StreamMode* используются?

Тип завершения	Тип потока	Разрешенные значения <i>StreamMode</i>
<Тип>	<Тип>	<SendOnly, RecvOnly, SendRecv, Loopback>

6.7.2 Дескриптор Events

Все события, содержащиеся в данном профиле могут быть установлены на любом завершении/потоке [со следующими исключениями].

ПРИМЕЧАНИЕ. – Текст в [] является дополнительным и включается, только если такие исключения существуют.

События, устанавливаемые на типах завершений и типах потоков:	<Да/Нет>		
<i>Если да</i>	EventID	Тип завершения	Тип потока
	<Название события и идентификационная информация, например, Событие Generic Error (g/cause, 0x0001/0x0001)>	<Тип>	<Тип потока, например, Audio/Video/Data или StreamID>

Используется ли EventBuffer Control?

EventBuffer Control используется:	<Да/Нет>
--	----------

Используется ли событие KeepActive?

KeepActive используется на событиях:	<Да/Нет>
---	----------

Используется ли вложение в дескрипторе Events?

Вложенные события в дескрипторе Events:	<Да/Нет>
Вложенные сигналы в дескрипторе Events:	<Да/Нет>

Поддерживаются ли регулируемые вложенные события?

Регулируемые вложенные события запускаются:	<Нет / Укажите отдельное событие>
--	-----------------------------------

Используется ли флаг ResetEventsDescriptor?

ResetEventsDescriptor используется с событиями:	<ALL / Нет / Укажите отдельное событие>
--	---

Какой режим Notification поддерживается?

NotifyImmediate:	<ALL события / Нет / Укажите отдельное событие>
NotifyRegulated:	<ALL события / Нет / Укажите отдельное событие>
NeverNotify:	<ALL события / Нет / Укажите отдельное событие>

6.7.3 Дескриптор EventBuffer

Поддерживается ли он?

Дескриптор EventBuffer используется:	<Да/Нет>	
<i>Если да</i>	EventIDs	<Название события и идентификационная информация, например, Событие Generic Error (g/cause, 0x0001/0x0001) или ALL>

6.7.4 Дескриптор Signals

Все сигналы, содержащиеся в данном профиле могут быть установлены на любом завершении/потоке[со следующими исключениями].

ПРИМЕЧАНИЕ. – Текст в [] является дополнительным и включается только тогда, когда такие исключения существуют.

Установка сигналов зависит от типов завершений и потоков:	<Да/Нет> ПРИМЕЧАНИЕ. – "Нет" означает, что на любом завершении или потоке может воспроизводиться любой сигнал. Если "Да", на любом завершении или потоке может воспроизводиться любой сигнал, не перечисленный ниже.		
<i>Если да</i>	SignalID	Тип завершения	Stream Type / ID
	<Название сигнала и идентификационная информация, например, Playtone (tonegen/pt, 0x0003/0x0001)>	<Тип>	<Тип потока, например, Audio/Video/Data или StreamID>

Поддерживаются ли списки сигналов? Если да, то каково максимальное число сигналов в списке на тип завершения/ потока, поддерживающего списки.

Списки сигналов поддерживаются:	<Да/Нет>	
<i>Если да</i>	Тип завершения, поддерживающего списки:	<Тип/ALL>
	Тип потока, поддерживающего списки:	<Тип/ALL>
	Максимальное число сигналов на список сигналов:	<Целое число>
	Параметр межсигнальной задержки поддерживается:	<Да/Нет>

Поддерживается ли замещение типа сигнала и продолжительности?

Тип сигнала и продолжительность поддерживаются:	<Да/Нет>	
<i>Если да</i>	SignalID	Замещение типа или продолжительности
	<Название сигнала и идентификационная информация, например, Playtone (tonegen/pt, 0x0003/0x0001) or "ALL">	<Тип, Продолжительность, Оба>

Поддерживается ли направление сигнала?

Signal Direction supported:	<Да/Нет>
------------------------------------	----------

Поддерживается ли "notifyCompletion"? Какие типы поддерживаются? Используется ли RequestID с "NotifyCompletion"?

NotifyCompletion поддерживается:	<Да/Нет>	
<i>Если да</i>	SignalID	Поддерживаемый тип завершения
	<Название сигнала и идентификационная информация, например, Playtone (tonegen/pt, 0x0003/0x0001) или ALL>	<ALL, TO, EV, ED, NC, PI>
Параметр RequestID поддерживается:	<Да/Нет>	

Может ли множество сигналов воспроизводиться одновременно?

Сигналы воспроизводятся одновременно:	<Да/Нет>	
<i>Если да</i>	Идентификаторы SignalID, которые могут воспроизводиться одновременно:	<Название сигнала и идентификационная информация, например, Playtone (tonegen/pt, 0x0003/0x0001) or ALL>

Поддерживается ли "KeepActive" для сигналов?

KeepActive используется на сигналах:	<Да/Нет>
---	----------

6.7.5 Дескриптор DigitMap

Поддерживаются ли DigitMap? Если да, укажите названия, структуры и таймеры.

DigitMap поддерживаются:	<Да/Нет>		
<i>Если да</i>	Название DigitMap	Структура	Таймеры
	<название>	<Описать>	<таймеры>

6.7.6 Дескриптор Statistics

Статистика поддерживается на завершениях, потоках или и том и другом?

Статистика поддерживается на:	<Завершение / Поток / Оба>
--------------------------------------	----------------------------

Необходимо ли предоставлять отчет по статистике?

Отчет по статистике предоставляется при выполнении команды Subtract:	<Да/Нет>	
<i>Если да</i>	Идентификаторы StatisticID, по которым предоставляется отчет:	<Название статистики и идентификационная информация, например, Packets Sent (rtp/ps, 0x000c/0x0004) или ALL>

6.7.7 Дескриптор ObservedEvents

Поддерживается ли время обнаружения?

Время обнаружения события поддерживается:	<Да/Нет>
--	----------

6.7.8 Дескриптор Topology

Если используется, каковы разрешенные установки?

Разрешены тройные:	<(T1, T2, односторонний) и т. п.>
---------------------------	-----------------------------------

6.7.9 Дескриптор Error

Какие коды ошибок определенных в комплекте и H.248.8 поддерживаются?

Коды ошибок, отправляемые MGC:

Поддерживаемые коды ошибок H.248.8:	<ALL H.248.8, список отдельных номеров>
Поддерживаемые коды ошибок, определяемых в комплекте:	Список кодов ошибок см. в п. 6.14.x <Ссылка на соответствующий подпункт 6.14 ниже>

Коды ошибок, отправляемые MG:

Поддерживаемые коды ошибок Н.248.8:	<ALL Н.248.8, список отдельных номеров>
Поддерживаемые коды ошибок, определяемых в комплекте	Список кодов ошибок см. в п. 6.14.x <Ссылка на соответствующий подпункт 6.14 ниже>

6.8 Команда API

ПРИМЕЧАНИЕ. – Подразумевается, что дескриптор *Error* может быть возвращен в любом ответе на команду.

6.8.1 Add

Какие дескрипторы могут использоваться в запросе Add?

Дескрипторы, используемые запросе Add:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
--	---

Какие дескрипторы могут использоваться в ответе Add?

Дескрипторы, используемые ответе Add:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
---------------------------------------	---

6.8.2 Modify

Какие дескрипторы могут использоваться в запросе Modify?

Дескрипторы, используемые запросе Modify:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
---	---

Какие дескрипторы могут использоваться в ответе Modify?

Дескрипторы, используемые ответе Modify:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
--	---

6.8.3 Subtract

Может ли дескриптор Audit быть использован в запросе Subtract?

Дескрипторы, используемые запросом Subtract:	<Audit>
--	---------

Может ли дескриптор Statistics быть использован в ответе Subtract?

Дескрипторы, используемые ответом Subtract:	<Statistics>
---	--------------

6.8.4 Move

Используется ли команда Move? Некоторые конфигурации контекста могут ее не использовать.

Команда Move используется:	<Да/Нет>
----------------------------	----------

Если используется:

Дескрипторы, используемые запросом Move:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
Дескрипторы, используемые ответом Move:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>

6.8.5 AuditValue

Проверка каких дескрипторов и/или отдельных свойств, сигналов, событий или статистики может проводиться?

Проверяемые свойства:	<Название свойства и идентификационная информация, например, Максимальное число контекстов (Root/maxNumberOfContexts, 0x0002/0x0001), ALL или Нет>	<Дескриптор: Local, Remote, LocalControl, TerminationState>
Проверяемая статистика	<Название статистики и идентификационная информация, например, Отправленные пакеты (rtp/ps, 0x000c/0x0004), ALL или Нет>	
Проверяемые сигналы:	<Название сигнала и идентификационная информация, например, Playtone (tonegen/pt, 0x0003/0x0001), ALL или None>	
Проверяемые события	<Название события и идентификационная информация, например, Событие общей ошибки (g/cause, 0x0001/0x0001), ALL или Нет>	
Возможные комплекты Audit:	Может ли проводиться проверка дескриптора Package? <Да/Нет>	

6.8.6 AuditCapabilities

Проверка каких дескрипторов и/или отдельных свойств, сигналов, событий или статистики может проводиться?

Проверяемые свойства:	<Название свойства и идентификационная информация, например, Максимальное число контекстов (Root/maxNumberOfContexts, 0x0002/0x0001), ALL или Нет>	<Дескриптор: Local, Remote, LocalControl, TerminationState>
Проверяемая статистика	<Название статистики и идентификационная информация, например, Отправленные пакеты (rtp/ps, 0x000c/0x0004), ALL или Нет>	
Проверяемые сигналы:	<Название события и идентификационная информация, например, Событие общей ошибки (g/cause, 0x0001/0x0001), ALL или Нет>	
Проверяемые события	<Название события и идентификационная информация, например, Событие общей ошибки (g/cause, 0x0001/0x0001), ALL или Нет>	

Возможна ли масштабная проверка?

Проверяемые свойства / Атрибуты контекста, используемые для масштабной проверки:	<Нет / ALL / Определить отдельные>
---	------------------------------------

6.8.7 Notify

Какие дескрипторы могут быть использованы в команде Notify?

Дескрипторы, используемые запросом или ответом Notify:	<ObservedEvents, Error>
---	-------------------------

6.8.8 ServiceChange

Какие ServiceChangeMethods и Reasons поддерживаются?

ServiceChangeMethod и **ServiceChangeReason**, отправляемые MGC:

Поддерживаемые ServiceChangeMethod:	Поддерживаемые ServiceChangeReason:
<Graceful, Forced, Restart, Handoff, Failover, ALL, Другие?>	<900-920>

ServiceChangeMethod и ServiceChangeReason, отправляемые MG:

Поддерживаемые ServiceChangeMethod:	Поддерживаемые ServiceChangeReason:
<Graceful, Forced, Restart, Disconnected, Handoff, Failover, ALL, Другие?>	<900-920>

Используется ли ServiceChangeAddress?

ServiceChangeAddress используется:	<Да/Нет>
---	----------

Используется ли ServiceChangeDelay?

ServiceChangeDelay используется:	<Да/Нет>
Если да	Допустимый период времени: <0-x> мс

Используется ли флаг ServiceChangeIncomplete?

Флаг ServiceChange Incomplete используется:	<Да/Нет>
--	----------

Какую версию Рек. МСЭ-Т использует ServiceChangeVersion? Низшее значение здесь должно быть минимальной версией, определенной в 6.3.

Версия, используемая в ServiceChangeVersion:	<1, 2, 3>
---	-----------

Могут ли поддерживаться многочисленные профили согласно Рек. МСЭ-Т Н.248.18?

Согласование профиля согласно Рек. МСЭ-Т Н.248.18:	<Да/Нет>
---	----------

6.8.9 Управление и проверка атрибутов контекста

Какие атрибуты контекста можно проверять и/или какими можно управлять?

Управляемые атрибуты контекста:	<Topology, Emergency, Priority, Индикатор IEPS, Дескриптор ContextAttribute (список названий атрибутов), ALL>
Проверяемые атрибуты контекста:	<Topology, Emergency, Priority, Индикатор IEPS, Дескриптор ContextAttribute (список названий атрибутов), ALL>

6.9 Общий синтаксис команд и кодирование

Определяется, какие виды кодирования поддерживаются профилем.

Поддерживаемые виды кодирования:	<Текстовое и двоичное, Двоичное, Текстовое>
---	---

6.10 Транзакции

Каково максимальное число запросов транзакций/ ответов на транзакции на одно сообщение?

Максимальное число запросов транзакций / ответов на транзакции / подтверждений ответов на транзакции / ответов сегментов на одно сообщение:	<Целое число>
--	---------------

Каково максимальное число команд на один запрос транзакции?

Максимальное число команд на запрос транзакции:	<Целое число>
---	---------------

Каково максимальное число команд на один ответ на транзакцию?

Максимальное число команд на ответ на транзакцию:	<Целое число>
---	---------------

Могут ли команды быть отмеченными как "Дополнительные"? Опишите.

Команды, которые могут быть помечены как "Дополнительные":	<Add, Modify, Move, Subtract, Auditvalue, Auditcapability, Servicechange, All, Ни одна>
--	---

Определите значения таймеров транзакций

Таймер транзакций:	Значение
normalMGExecutionTime	<Целое число или "Предоставляется">
normalMGCExecutionTime	<Целое число или "Предоставляется">
MGOriGinatedPendingLimit	<Целое число или "Предоставляется">
MGC OriGinatedPendingLimit	<Целое число или "Предоставляется">
MGProvisionalResponseTimerValue	<Целое число или "Предоставляется">
MGCProvisionalResponseTimerValue	<Целое число или "Предоставляется">

6.11 Сообщения

Соглашения о наименовании MGC/MG: адресация MID, связанная с названиями MGC/MG.

<Описать>

Показывает максимальное число транзакций на одно сообщение (Данную таблицу можно пропустить, если она не является важной)

Максимальное число транзакций на одно сообщение:	<Целое число>
--	---------------

6.12 Передача

Определяет, какие подсерии передач H.248 поддерживаются данным профилем.

Поддерживаемые передачи:	<UDP, TCP, SCTP, MTP3B, SSCOP/AAL 5, ALF/AAL 5>
--------------------------	---

Поддерживается ли сегментация, если да, то при помощи какого метода?

Поддерживаемая сегментация:	<Нет, Присущая передаче, Сегментация H.248>
-----------------------------	---

Используется ли мониторинг связи управления (см. п. 11.6), если да, то при помощи какого метода?

Поддерживаемый мониторинг связи управления:	<Нет, Присущая передаче, Пустое AuditValue на Root, H.248.14>
---	---

6.13 Безопасность

Определяет используемые алгоритмы безопасности.

Поддерживаемая безопасность:	<Нет, Промежуточная схема АН, IPSec>
------------------------------	--------------------------------------

6.14 Комплекты

Определяет комплекты, поддерживаемые в данном профиле.

Обязательные для выполнения: Определяет комплекты, которые должны поддерживаться в данном профиле.

Обязательные для выполнения комплекты:		
Название комплекта	PackageID	Версия
<Название>	<xxxx, (0x00xx)>	<1, 2, 3, ...>

Дополнительные: Определяет комплекты, которые могут поддерживаться в данном профиле.

Дополнительные комплекты:			
Название комплекта	PackageID	Версия	Поддержка зависит от:
<Название>	<xxx, 0x00??>	<1, 2, 3, ...>	<Описать>

Информация по использованию комплекта

В данной таблице определяется, как будут использоваться вышеописанные комплекты. Например:

- перечисляется, являются ли свойства / сигналы / события / статистика дополнительными или обязательными;
- следует ли определять значение предоставленного свойства / сигнала / события (например, названия и номера циклов для объявления Н.248.7);

Определяет значения свойств, которые заданы как предоставляемые.

Информация по использованию комплектов

6.14.x <Название комплекта>

Свойства	Обязательный/ Дополнительный	Используется в команде:	Поддерживаемые значения:	Предоставляемое значение:
<Название и идентификационная информация, например, Packets Sent (rtp/ps, 0x000с/0x0004), ALL или Ни один>	<M/O>	<ADD, MOD, MOVE, AUDITVALUE, AUDITCAP>	<Значения / ALL>	<Значение / Не применяется>
Сигналы	Обязательный/ Дополнительный	Используется в команде:		Предоставляемое значение продолжительности
<Название и идентификационная информация>	<M/O>	<ADD, MOD, MOVE, AUDITVALUE, AUDITCAP>		<Значение / не применяется>
	Параметры сигнала	Обязательный/ Дополнительный	Поддерживаемые значения:	Предоставляемое значение продолжительности:
<Название и идентификационная информация>	<M/O>	<Значения / ALL>	<Значение / Не применяется>	
События	Обязательный/ Дополнительный	Используется в команде:		
<Название и идентификационная информация>	<M/O>	<ADD, MOD, MOVE, NOTIFY, AUDITVALUE, AUDITCAP>		
	Параметры события	Обязательный/ Дополнительный	Поддерживаемые значения:	Предоставляемое значение:
	<Название и идентификационная информация>	<M/O>	<Значения / ALL>	<Значение / Не применяется>
	Параметры ObservedEvent	Обязательный/ Дополнительный	Поддерживаемые значения:	Предоставляемое значение:
<Название и идентификационная информация>	<M/O>	<Значения / ALL>	<Значение / Не применяется>	
Статистика	Обязательный/ Дополнительный	Используется в команде:	Поддерживаемые значения:	
<Название и идентификационная информация>	<M/O>	<ADD, MOD, MOVE, SUBTRACT, AUDITVALUE, AUDITCAP>	<Значения/ ALL>	
Коды ошибки	Обязательный/ Дополнительный			
<Номер>	<M/O>			

По желанию пользователя в таблицу могут быть занесены дополнительные ограничения.

6.15 Обязательная поддержка элементов информации SDP и Приложения С

Определяет, какие атрибуты SDP и элементы информации Приложения С должны поддерживаться.

Поддерживаемые элементы информации Приложения С и SDP:		
Элемент информации	Поддержка Приложения С	Поддержка SDP
<Название>	<Свойство Приложения С>	<Описать>

6.16 Дополнительная поддержка элементов информации SDP и Приложения С

Определяет, какие атрибуты SDP и элементы информации Приложения С могут поддерживаться.

Дополнительные элементы информации Приложения С и SDP:			
Элемент информации	Поддержка Приложения С	Поддержка SDP	Поддержка зависит от:
<Название>	<Свойство Приложения С>	<Описать>	<Описать>

6.17 Процедуры

Определяет процедуры, связанные с данным профилем.

Рекомендуется, чтобы процедуры имели следующий формат:

6.17.1 <Название процедуры>

Когда требуется <название> процедуры, запускается следующее.

От <MGC/MG> к <MG/MGC> посылается команда <ADD.req, MOD.req, MOV.req, SUB.req, AuditValue.req, AuditCapability.req, ServiceChange.req, Notify.req> со следующей информацией.

<Вставьте информацию в соответствующем формате. Например, дескрипторы, свойство, сигнал, событие, название статистики или абстракция действительных параметров на основе элементов информации.>

После получения команды <MG/MGC> должен:

- <Обозначить действия>

После завершения выполнения команды (1) от <MG/MGC> к <MGC/MG> посылается команда <ADD.reply, MOD.reply, MOV.reply, SUB.reply, AuditValue.reply, AuditCapability.reply, ServiceChange.reply, Notify.reply> со следующей информацией:

<Вставьте информацию в соответствующем формате. Например, дескрипторы, свойство, сигнал, событие, название статистики или абстракция действительных параметров на основе элементов информации.>

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 межсетевых протоколов и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи