



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

H.248.1

(03/2002)

СЕРИЯ H: АУДИОВИЗУАЛЬНЫЕ И
МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб – Процедуры
связи

Протокол управления шлюзом: Версия 1

Рекомендация МСЭ-Т H.248.1

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Н
АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	Н.100–Н.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ СЛУЖБ	
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 подвижных видеоизображений	Н.260–Н.279
Сопутствующие системные аспекты	Н.280–Н.299
СИСТЕМЫ И ОКОНЕЧНОЕ ОБОРУДОВАНИЕ ДЛЯ АУДИОВИЗУАЛЬНЫХ СЛУЖБ	Н.300–Н.399
ДОПОЛНИТЕЛЬНЫЕ УСЛУГИ ДЛЯ МУЛЬТИМЕДИЙНЫХ СЛУЖБ	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 служб серии Н	Н.510–Н.519
Приложения и службы мобильной мультимедийной совместной работы	Н.520–Н.529
Безопасность для мобильных мультимедийных систем и служб	Н.530–Н.539
Безопасность для приложений и служб 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

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

Рекомендация МСЭ-Т Н.248.1

Протокол управления шлюзом: Версия 1

Аннотация

В настоящей Рекомендации, с целью обеспечения лучшей масштабируемости, функция шлюза Н.323, определённая 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, распределённых среди нескольких физических платформ. Цель данной Рекомендации - дополнить возможности, определяемые в настоящее время для систем Н.323, и предназначается для обеспечения новых путей исполнения операций, уже поддерживаемых в Рекомендации МСЭ-Т Н.323.

ЗАМЕЧАНИЕ – Эта Рекомендация включает изменение нумерации Рекомендации МСЭ-Т Н.248, ее Приложений от А до Е, а также Дополнения I.

Источник

Рекомендация МСЭ-Т Н.248.1 подготовлена 16-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) и утверждена 29 марта 2002 года в соответствии с процедурой, предусмотренной Резолюцией 1 ВАСЭ.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи (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.

Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, разрабатывают 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 данной Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

На момент утверждения настоящей Рекомендации МСЭ не получил извещения об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации данной 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.

© МСЭ 2004

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена или использована в какой бы то ни было форме или с помощью каких-либо средств, электронных либо механических, включая изготовление фотокопий и микрофильмов, без письменного разрешения МСЭ.

СОДЕРЖАНИЕ

	Стр.
1 Область применения	1
2 Ссылки.....	1
2.1 Нормативные библиографические ссылки	1
2.2 Информационные библиографические ссылки.....	3
3 Определения.....	3
4 Сокращения.....	4
5 Соглашения.....	5
6 Модель соединения	5
6.1 Контексты	7
6.1.1 Качественные признаки контекста и дескрипторы.....	7
6.1.2 Создание, удаление и изменение контекстов	8
6.2 Окончания	8
6.2.1 Динамика окончания.....	10
6.2.2 Идентификаторы TerminationID	10
6.2.3 Комплекты	11
6.2.4 Свойства окончания и дескрипторы.....	12
6.2.5 Окончание Root	13
7 Команды	14
7.1 Дескрипторы.....	15
7.1.1 Определяющие аргументы	15
7.1.2 Дескриптор Modem	15
7.1.3 Дескриптор Multiplex.....	15
7.1.4 Дескриптор Media	16
7.1.5 Дескриптор TerminationState.....	16
7.1.6 Дескриптор Stream	17
7.1.7 Дескриптор LocalControl	17
7.1.8 Дескрипторы Local и Remote	18
7.1.9 Дескриптор Events	20
7.1.10 Дескриптор EventBuffer	23
7.1.11 Дескриптор Signals	23
7.1.12 Дескриптор Audit	24
7.1.13 Дескриптор ServiceChange	25
7.1.14 Дескриптор DigitMap	26
7.1.15 Дескриптор Statistics	29
7.1.16 Дескриптор Packages	30
7.1.17 Дескриптор ObservedEvents	30

	Стр.
7.1.18	Дескриптор Topology 30
7.1.19	Дескриптор Error 32
7.2	Стык прикладного программирования команд 32
7.2.1	Команда Add 32
7.2.2	Команда Modify 34
7.2.3	Команда Subtract 35
7.2.4	Команда Move 36
7.2.5	Команда AuditValue 37
7.2.6	Команда AuditCapabilities 39
7.2.7	Команда Notify 40
7.2.8	Команда ServiceChange 41
7.2.9	Изменение и проверка качественных признаков контекста 44
7.2.10	Общий синтаксис команд 45
7.3	Коды ошибок команд 45
8	Транзакции 45
8.1	Общие аргументы 46
8.1.1	Идентификаторы транзакций 46
8.1.2	Идентификаторы контекстов 46
8.2	Стык прикладного программирования транзакций 47
8.2.1	Аргумент TransactionRequest 47
8.2.2	Аргумент TransactionReply 47
8.2.3	Аргумент TransactionPending 49
8.3	Сообщения 49
9	Транспорт 49
9.1	Упорядочение команд 50
9.2	Защита от лавинообразной перегрузки 51
10	Соображения о безопасности 52
10.1	Защита соединений протокола 52
10.2	Временная структура АН 52
10.3	Защита медиасоединений 53
11	Стык управления MG-MGC 53
11.1	Несколько виртуальных шлюзов MG 54
11.2	Начальная загрузка 54
11.3	Согласование версии протокола 55
11.4	Сбой шлюза MG 56
11.5	Сбой контроллера MGC 56
12	Определение комплекта 57
12.1	Директивы по определению комплектов 57

	Стр.
12.1.1 Комплект	57
12.1.2 Свойства	58
12.1.3 События.....	59
12.1.4 Сигналы.....	59
12.1.5 Статистика	60
12.1.6 Процедуры	60
12.2 Директивы по определению аргументов событий и сигналов	60
12.3 Списки	61
12.4 Идентификаторы	61
12.5 Регистрация комплектов.....	61
13 Соображения о IANA	61
13.1 Комплекты	61
13.2 Коды ошибок	62
13.3 Причины ServiceChange.....	62
Приложение А – Двоичное кодирование протокола.....	63
А.1 Кодирование групповых символов замены.....	63
А.2 Спецификация синтаксиса ASN.1	64
А.3 Планы нумерации и имена маршрутов	78
Приложение В – Текстовое кодирование протокола	79
В.1 Кодирование групповых символов замены	79
В.2 Спецификация ABNF.....	79
В.3 Кодирование шестнадцатеричного октета.....	90
В.4 Последовательность шестнадцатеричного октета	90
Приложение С – Теги для свойств медиапотокa.....	90
С.1 Общие атрибуты среды передачи.....	91
С.2 Свойства мультиплексора	92
С.3 Общие свойства транспортного канала	92
С.4 Общие свойства АТМ	92
С.5 Frame Relay	95
С.6 Протокол IP.....	95
С.7 Уровень адаптации типа 2 АТМ	95
С.8 Уровень адаптации типа 1 АТМ	96
С.9 Возможности транспортных каналов.....	98
С.10 Свойства уровня адаптации типа 5.....	106
С.11 Эквиваленты SDP.....	106
С.12 Рекомендация МСЭ-Т Н.245	107

	Стр.
Приложение D – Транспорт через IP	107
D.1 Транспорт через IP/UDP с использованием прикладного уровня кадрирования (ALF)	107
D.1.1 Обеспечение самое большое однократных функциональных возможностей	108
D.1.2 Идентификаторы транзакций и трёхстороннее подтверждение соединения	108
D.1.3 Вычисление таймеров повторной передачи	109
D.1.4 Предварительные ответы.....	110
D.1.5 Повторение запросов, ответов и подтверждений.....	110
D.2 Использование TCP.....	111
D.2.1 Обеспечение самое большое однократных функциональных возможностей	111
D.2.2 Идентификаторы транзакций и трёхстороннее подтверждение соединения	112
D.2.3 Вычисление таймеров повторной передачи	112
D.2.4 Предварительные ответы.....	112
D.2.5 Упорядочение команд.....	112
Приложение E – Основные комплекты	112
E.1 Общее	112
E.1.1 Свойства	113
E.1.2 События.....	113
E.1.3 Сигналы.....	114
E.1.4 Статистика	114
E.2 Основной комплект Root	114
E.2.1 Свойства	114
E.2.2 События.....	116
E.2.3 Сигналы.....	116
E.2.4 Статистика	116
E.2.5 Процедуры	116
E.3 Комплект генератора тона.....	116
E.3.1 Свойства	116
E.3.2 События.....	116
E.3.3 Сигналы.....	116
E.3.4 Статистика	117
E.3.5 Процедуры	117
E.4 Комплект обнаружения тона.....	117
E.4.1 Свойства	117
E.4.2 События.....	117
E.4.3 Сигналы.....	119

	Стр.
Е.4.4	Статистика 119
Е.4.5	Процедуры 119
Е.5	Основной комплект DTMF генератора 119
Е.5.1	Свойства 119
Е.5.2	События 119
Е.5.3	Сигналы 120
Е.5.4	Статистика 120
Е.5.5	Процедуры 120
Е.6	Комплект обнаружения DTMF 121
Е.6.1	Свойства 121
Е.6.2	События 121
Е.6.3	Сигналы 122
Е.6.4	Статистика 122
Е.6.5	Процедуры 122
Е.7	Комплект генератора тонов прохождения соединения 122
Е.7.1	Свойства 122
Е.7.2	События 123
Е.7.3	Сигналы 123
Е.7.4	Статистика 123
Е.7.5	Процедуры 123
Е.8	Комплект обнаружения тонов прохождения соединения 123
Е.8.1	Свойства 124
Е.8.2	События 124
Е.8.3	Сигналы 124
Е.8.4	Статистика 124
Е.8.5	Процедуры 124
Е.9	Комплект контроля аналоговой линии 124
Е.9.1	Свойства 124
Е.9.2	События 124
Е.9.3	Сигналы 126
Е.9.4	Статистика 127
Е.9.5	Процедуры 127
Е.9.6	Код ошибки 127
Е.10	Основной комплект связности 127
Е.10.1	Свойства 127
Е.10.2	События 127
Е.10.3	Сигналы 128
Е.10.4	Статистика 128
Е.10.5	Процедуры 128
Е.11	Комплект сети 129

	Стр.
E.11.1 Свойства.....	129
E.11.2 События.....	129
E.11.3 Сигналы.....	130
E.11.4 Статистика	130
E.11.5 Процедуры	130
E.12 Комплект RTP.....	130
E.12.1 Свойства.....	131
E.12.2 События.....	131
E.12.3 Сигналы.....	131
E.12.4 Статистика	131
E.12.5 Процедуры	132
E.13 Комплект TDM канала.....	132
E.13.1 Свойства.....	132
E.13.2 События.....	133
E.13.3 Сигналы.....	133
E.13.4 Статистика	133
E.13.5 Процедуры	133
Приложение I – Пример прохождения соединения	133
I.1 Соединение частного шлюза с частным шлюзом	133
I.1.1 Программирование окончаний аналоговых каналов частного шлюза в режиме незанятости	133
I.1.2 Сбор исходных цифр и инициализация окончания	135

Рекомендация МСЭ-Т Н.248.1

Протокол управления шлюзом: Версия 1

1 Область применения

Данная Рекомендация определяет протоколы, используемые между элементами физически распределённого мультимедийного шлюза, применяемого в соответствии с архитектурой, указанной в Рекомендации МСЭ-Т Н.323. С системной точки зрения нет функциональных различий между шлюзом, разделённым на компоненты, потенциально распределёнными в более, чем одном физическом устройстве, и единым шлюзом, определяемым Рекомендацией МСЭ-Т Н.246. Эта Рекомендация не определяет, как работают шлюзы, многоточечные управляющие устройства или устройства интерактивного голосового ответа (IVRs). Вместо этого она создаёт общую инфраструктуру, подходящую для этих приложений.

Стыки пакетной сети могут включать IP, ATM или, возможно, и другие. Эти стыки 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.

2 Ссылки

Нижеследующие Рекомендации МСЭ-Т и другие источники содержат положения, которые путём ссылок на них в данном тексте образуют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники подвергаются пересмотру; поэтому пользователям этой Рекомендации следует рассматривать возможность применения самых последних изданий перечисленных ниже Рекомендаций и других источников. Список действующих Рекомендаций МСЭ-Т регулярно публикуется.

2.1 Нормативные библиографические ссылки

- Рекомендация МСЭ-Т Н.225.0 (2000), *Протоколы сигнализации о соединении и пакетизация медиапотоков для мультимедийных систем связи на основе пакетов.*
- Рекомендация МСЭ-Т Н.235 (2000), *Безопасность и шифрование для мультимедийных терминалов серии Н (Н.235 и других на основе Н.245).*
- Рекомендация МСЭ-Т Н.245 (2001), *Протокол управления для мультимедийной связи.*
- Рекомендация МСЭ-Т Н.246 (1998), *Взаимодействие мультимедийных терминалов серии Н с мультимедийными терминалами серии Н и голосовыми/голосового диапазона терминалами в КТСОП и ЦСИС.*
- Рекомендация МСЭ-Т Н.248.8 (2002), *Протокол управления шлюзом: Код ошибки и описание причины изменения службы.*
- Рекомендация МСЭ-Т Н.323 (2000), *Мультимедийные системы связи на основе пакетов.*
- Рекомендация МСЭ-Т I.363.1 (1996), *Спецификация уровня адаптации АТМ для Ш-ЦСИС: ААЛ типа 1.*
- Рекомендация МСЭ-Т I.363.2 (2000), *Спецификация уровня адаптации АТМ для Ш-ЦСИС: ААЛ типа 2.*

- Рекомендация МСЭ-Т I.363.5 (1996), *Спецификация уровня адаптации АТМ для Ш-ЦСИС: AAL типа 5.*
- Рекомендация МСЭ-Т I.366.1 (1998), *Служебно- ориентированной подуровень конвергенции сегментации и сборки для AAL типа 2.*
- Рекомендация МСЭ-Т I.366.2 (2000), *Служебно- ориентированной подуровень конвергенции AAL типа 2 для узкополосных служб.*
- Рекомендация МСЭ-Т I.371 (2000), *Управление трафиком и управление перегрузкой в Ш-ЦСИС.*
- Рекомендация МСЭ-Т Q.763 (1999), *Система сигнализации No. 7 – Форматы и коды пользовательской части ЦСИС.*
- Рекомендация МСЭ-Т Q.765.5 (2000), *Система сигнализации No. 7 – Прикладной транспортный механизм: Управление соединением, независимое от транспортного канала (BICC).*
- Рекомендация МСЭ-Т Q.931 (1998), *Спецификация стыка 3 уровня пользователь-сеть для основного управления соединением SDN user-network interface layer 3 specification for basic call control.*
- Рекомендация МСЭ-Т Q.2630.1 (1999), *Протокол сигнализации AAL типа 2 – Capability Set 1.*
- Рекомендация МСЭ-Т Q.2931 (1995), *Цифровая система абонентской сигнализации No. 2 – Стык “пользователь-сеть” (UNI) – Спецификация уровня 3 для управления основным вызовом/соединением.*
- Рекомендация МСЭ-Т Q.2941.1 (1997), *Цифровая система абонентской сигнализации No. 2 – Транспорт общего идентификатора.*
- Рекомендация МСЭ-Т Q.2961.1 (1995), *Дополнительные возможности сигнализации для поддержки характеристик трафика при факультативном тегировании и обоснованной установке характеристик скорости передачи ячеек.*
- Рекомендация МСЭ-Т Q.2961.2 (1997), *Поддержка возможности передачи АТМ в широкополосном информационном элементе с возможностями транспортного канала.*
- Рекомендация МСЭ-Т Q.2965.1 (1999), *Цифровая система абонентской сигнализации No. 2 – Поддержка классов качества обслуживания.*
- Рекомендация МСЭ-Т Q.2965.2 (1999), *Цифровая система абонентской сигнализации No. 2 – Сигнализация отдельных характеристик качества службы.*
- Рекомендация МСЭ-Т V.76 (1996), *Общий мультиплексор с использованием процедур, основанных на V.42 LAPM.*
- Рекомендация МСЭ-Т X.213 (2001), *Информационные технологии – Открытые системы взаимосвязи – Определение сетевых служб плюс поправка I(1997), Дополнение к идентификатору адресного формата протокола Интернет.*
- Рекомендация МСЭ-Т X.680 (1997), *Информационные технологии – Абстрактный синтаксис, нотация один (ASN.1): Спецификация основной нотации.*
- Рекомендация МСЭ-Т X.690 (2002), *Информационные технологии – Правила кодирования ASN.1: Спецификация основных правил кодирования (BER), Канонические правила кодирования (CER) и отмеченные правила кодирования (DER).*

- Форум АТМ (1996), *Спецификация сигнализации АТМ стыка пользователь-сеть (UNI) – Версия 4.0.*
- IETF RFC 1006, *Транспортная служба МОС поверх TCP, Версия 3.*
- IETF RFC 2234 (1997), *Спецификация синтаксиса для расширенного BNF: ABNF.*
- IETF RFC 2327 (1998), *SDP: протокол описания сессий.*
- IETF RFC 2402 (1998), *Заголовок с аутентификацией протокола IP.*
- IETF RFC 2406 (1998), *Безопасное закрытие содержания (ESP) протокола IP.*

2.2 Информационные библиографические ссылки

- Рекомендация МСЭ-Т E.180/Q.35 (1998), *Технические характеристики тональных сигналов для телефонной службы.*
- Рекомендация МСЭ-Т G.711 (1988), *Импульсно-кодовая модуляция (ИКМ) голосовых частот.*
- Рекомендация МСЭ-Т H.221 (1999), *Структура кадра канала от 64 до 1920 Кбит/с в аудиовизуальных телеслужбах.*
- Рекомендация МСЭ-Т H.223 (2001), *Мультиплексный протокол для низкоскоростной мультимедийной связи.*
- Рекомендация МСЭ-Т H.226 (1998), *Протокол объединения каналов для многоканальной работы в сетях с коммутацией каналов.*
- Рекомендация МСЭ-Т Q.724 (1988), *Процедуры сигнализации части пользователя телефона.*
- Рекомендация МСЭ-Т Q.764 (1999), *Система сигнализации № 7 – Процедуры сигнализации пользовательской части ЦСИС.*
- Рекомендация МСЭ-Т Q.1902.4 (2001), *Протокол управления соединением, независимый от транспортного канала (Совокупность возможностей 2): Основные процедуры соединения.*
- IETF RFC 768 (1980), *Датаграмный протокол пользователя.*
- IETF RFC 791 (1981), *Протокол Интернет.*
- IETF RFC 793 (1981), *Протокол управления передачей.*
- IETF RFC 1661 (1994), *Двухточечный протокол (PPP).*
- IETF RFC 1889 (1996), *RTP: Транспортный протокол для приложений реального времени.*
- IETF RFC 1890 (1996), *Профиль RTP для аудио и видео конференций с минимальным управлением.*
- IETF RFC 2401 (1998), *Безопасная архитектура для протокола Интернета.*
- IETF RFC 2460 (1998), *Спецификация версии 6 (IPv6) протокола Интернета.*
- IETF RFC 2543 (1999), *SIP: Протокол инициализации сессий.*
- IETF RFC 2805 (2000), *Архитектура и требования протокола управления медиашлюзом.*

3 Определения

Данная Рекомендация определяет следующие термины:

3.1 шлюз доступа: Тип шлюза, обеспечивающий стык “пользователь-сеть” (UNI), такой, как ЦСИС.

3.2 дескриптор: Синтаксический элемент протокола, группирующий связанные свойства. Например, свойства потока среды передачи MG могут быть установлены посредством MGC путём включения в команду подходящего дескриптора.

3.3 медиашлюз (MG): Медиашлюз преобразует информацию, имеющую место в сети одного типа, в формат, требуемый сетью другого типа. Например, медиашлюз MG может служить окончанием транспортных каналов сетей с коммутацией каналов (т.е. DS0) и медиапоток пакетных сетей (т.е. потоков RTP в IP сети). Этот шлюз может быть в состоянии обработать аудио, видео и T.120 как по одиночке, так и в любой комбинации и будет способен выполнить двусторонние преобразования носителей. Шлюз MG может также воспроизводить аудио/видео сообщения и выполнять другие IVR функции, или может выполнять организацию медиаконференции.

3.4 контроллер медиашлюза (MGC): Управляет составляющими состояниями соединения, имеющими отношение к управлению соединением для медиаканалов в MG.

3.5 многоточечный блок управления (MCU): Объект, управляющий начальной установкой и координацией конференции разнородных пользователей, что обычно включает обработку аудио, видео и данных.

3.6 частный шлюз: Шлюз, обеспечивающий взаимодействие аналоговой линии с пакетной сетью связи. Частный шлюз обычно содержит одну или две аналоговые линии и располагается в помещении покупателя.

3.7 шлюз сигнализации SCN FAS: Эта функция содержит стык сигнализации SCN, оканчивающий SS7, ЦСИС или другие звенья сигнализации, причём канал управления соединением и транспортные каналы располагаются в том же самом физическом пространстве.

3.8 шлюз сигнализации SCN NFAS: Эта функция содержит стык сигнализации SCN, оканчивающий SS7 или другие звенья сигнализации, причём каналы управления соединением разделены с В-каналами.

3.9 поток: Двухнаправленная среда передачи или поток управления, принятые/переданные медиашлюзом как часть соединения или конференции.

3.10 канал: Канал связи между двумя коммутационными системами, например, DS0 в системах T1 или E1.

3.11 канальный шлюз: Шлюз между сетями с коммутацией каналов и коммутацией пакетов, обычно являющийся окончанием для большого числа цифровых каналов.

4 Сокращения

Эта Рекомендация определяет следующие термины:

КТСОП	Коммутируемая телефонная сеть общего пользования (Public Switched Telephone Network)
ALF	Прикладной уровень кадрирования (Application Level Framing)
ATM	Асинхронный Режим Переноса (Asynchronous Transfer Mode)
CAS	Сигнализация, связанная с каналом (Channel Associated Signalling)

DTMF	Двухтоновый многочастотный набор номера (Dual Tone Multi-Frequency)
FAS	Сигнализация, связанная с услугой (Facility Associated Signalling)
GSM	Глобальная система мобильной связи (Global System for Mobile communications)
GW	Шлюз
IANA	Орган присвоения номеров Интернет (Internet Assigned Numbers Authority), заменённый Корпорацией Интернета по присвоению имён и номеров (Internet Corporation for Assigned Names and Numbers – ICANN)
IP	Межсетевой протокол (Internet Protocol)
ISUP	Пользовательская часть ЦСИС (ISDN User Part)
IVR	Интерактивный голосовой ответ (Interactive Voice Response)
MG	Медиашлюз (Media Gateway)
MGC	Контроллер медиашлюза (Media Gateway Controller)
NFAS	Сигнализация, не связанная с услугой (Non-Facility Associated Signalling)
PRI	Стык первичной скорости передачи
QoS	Качество обслуживания (Quality of Service)
RTP	Транспортный протокол реального времени (Real-Time Transport Protocol)
SCN	Сеть с коммутацией каналов (Switched Circuit Network)
SG	Шлюз сигнализации
SS7	Система сигнализации No. 7

5 Соглашения

В данной Рекомендации "Должен" ("SHALL") относится к обязательному требованию, в то время как "Следует" ("SHOULD") относится к предлагаемой, но необязательной характеристике или процедуре. Термин "Может" ("MAY") относится к необязательному образу действий без выражения предпочтительности.

6 Модель соединения

Модель соединения протокола описывает логические элементы, или объекты, в пределах медиашлюза, которые могут управляться контроллером медиашлюза. Основными абстракциями, используемыми в модели соединения, являются окончания и контексты.

Окончание является источником и/или приёмником одного или более потоков. В мультимедийной конференции окончанием может быть мультимедия, а также источники или приёмники нескольких медиапотоков. Аргументы медиапотока, также как и модем и аргументы транспортного канала, вложены в окончание.

Контекст является взаимосвязью между совокупностью окончаний. Существует специальный вид контекста, *нулевой* контекст, который содержит все окончания, не взаимосвязанные ни с каким другим окончанием. Например, в распределённом шлюзе доступа все нерабочие линии представлены окончаниями в нулевом контексте.

Ниже приведено графическое изображение этих понятий. Схема на рис. 1 даёт несколько примеров и не претендует на то, чтобы быть иллюстрацией, учитывающей всё. Прямоугольник со звёздочкой в каждом контексте представляет собой логическую взаимосвязь окончаний, подразумеваемых контекстом.

Медиашлюз

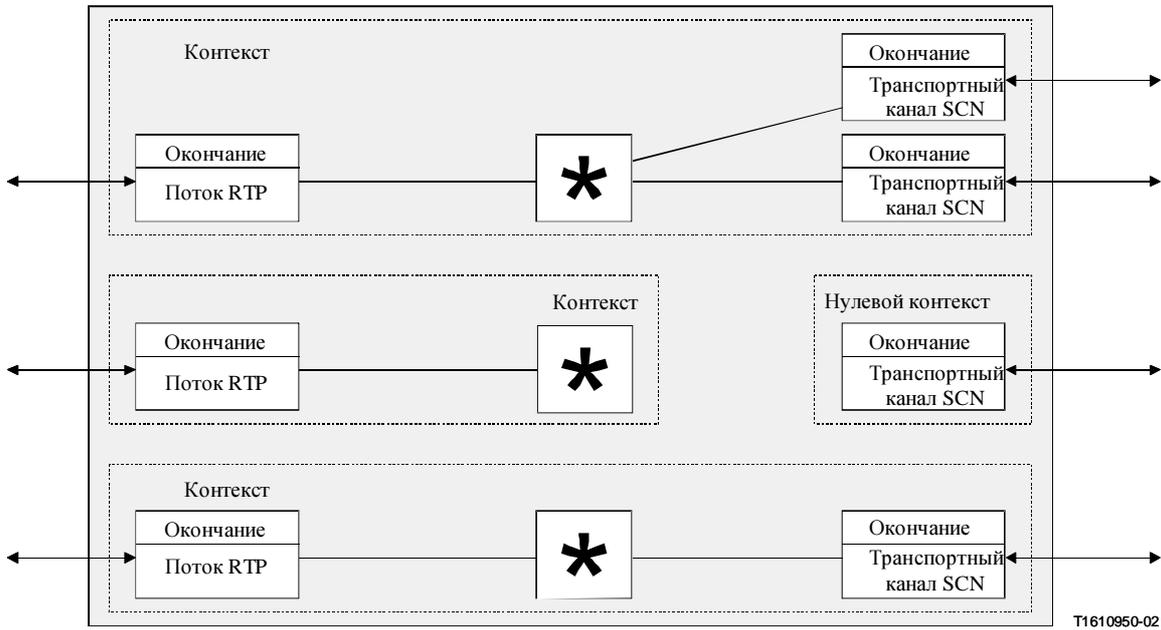


Рисунок 1/Н.248.1 – Пример модели соединений Н.248.1

Пример на рис. 2 является примером одного из способов осуществления сценария ожидания соединения в разделённом шлюзе доступа, иллюстрирующего перемещение окончания между контекстами. Окончания T1 и T2 принадлежат контексту C1 в двухстороннем аудио соединении. Второе аудио соединение ожидает T1 от окончания T3. T3 является единственным в контексте C2. T1 принимает соединение от T3, переводя T2 в ожидание. Это действие приводит к пересылке T1 в контекст C2, как это показано на рис. 3.

Медиашлюз

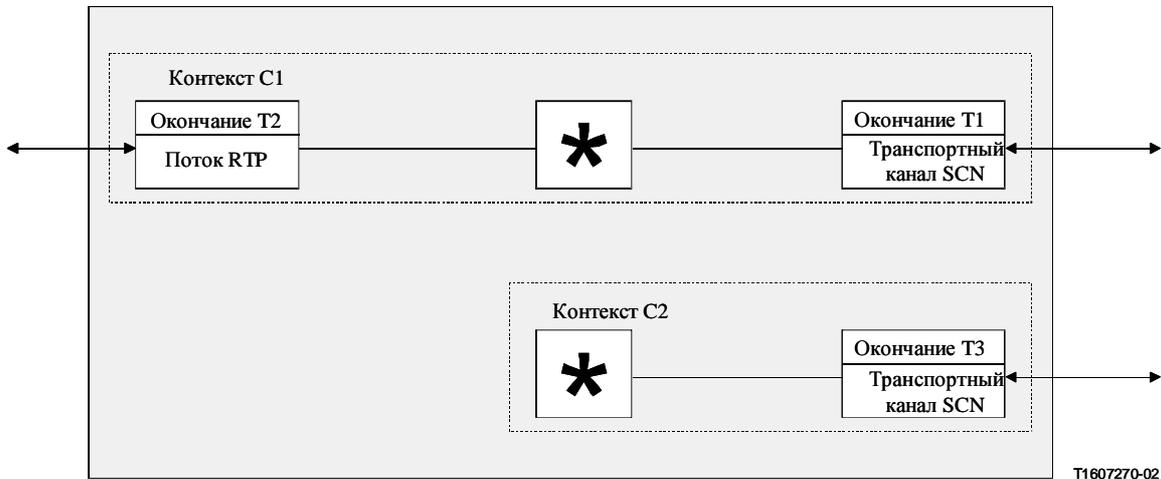


Рисунок 2/Н.248.1 – Пример сценария ожидания соединения /предупреждение, применяемое к T1

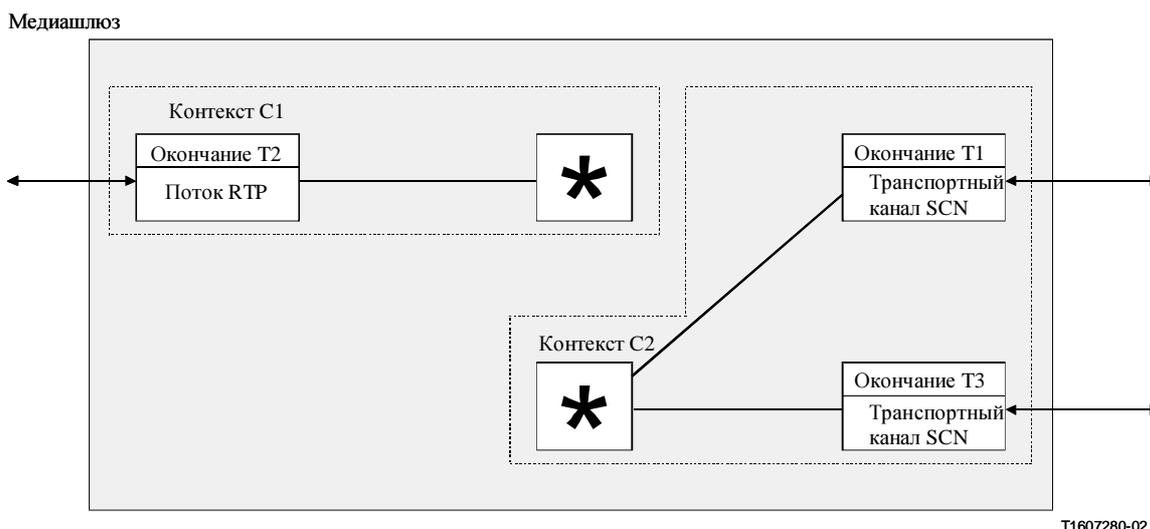


Рисунок 3/Н.248.1 – Пример сценария ожидания соединения /ответ T1

6.1 Контексты

Контекст является взаимосвязью между совокупностью окончаний. Контекст описывает топологию (кто кого слышит/видит) и микширование носителей и/или аргументы коммутации, если более, чем два окончания вовлечено во взаимосвязь.

Существует специальный контекст, называемый *нулевым (null)* контекстом. Он содержит окончания, которые не взаимосвязаны ни с каким другим окончанием. Окончания в *нулевом* контексте могут иметь свои проверенные или изменённые аргументы и могут иметь обнаруженные на них события.

Вообще команда Add применяется для присоединения окончаний к контекстам. Если контроллер MGC не указывает существующий контекст, к которому должно быть присоединено окончание, шлюз MG создаёт новый контекст. Окончание может быть удалено из контекста при помощи команды Subtract, и окончание может быть переслано из одного контекста в другой при помощи команды Move. В каждый момент окончание ДОЛЖНО существовать только в одном контексте.

Максимальное число окончаний в контексте является свойством шлюза MG. Медиашлюзы, предоставляющие лишь возможность соединения точка-точка, могут допустить максимум два окончания на каждый контекст. Медиашлюзы, поддерживающие многоточечные конференции, могут допустить три или более окончаний на контекст.

6.1.1 Качественные признаки контекста и определители

Качественными признаками контекста являются:

- Идентификатор ContextID.
- Топология (кто кого слышит/видит)

Топология контекста описывает поток среды передачи между окончаниями в пределах контекста. В противоположность этому, режим окончания (передача/приём/...) описывает поток среды передачи на входе/выходе медиашлюза.

- Приоритет используется в контексте для того, чтобы обеспечить шлюз MG информацией о некотором порядке обработке контекста. В некоторых ситуациях (например, рестарт) контроллер MGC может также использовать приоритет для

плавного автоматического управления порядком трафика, когда много контекста должно быть обработано одновременно. Приоритет 0 означает низший приоритет, а приоритет 15 соответствует самому высокому приоритету.

- Также обеспечивается признак аварийного соединения, чтобы разрешить предпочтительную обработку в шлюзе MG.

6.1.2 Создание, удаление и изменение контекстов

Протокол может быть использован (неявно) для создания контекста и изменения значений аргументов существующих контекстов. Протокол содержит команды присоединения окончаний к контекстам, вычитания их из контекстов и пересылки окончаний между контекстами. Контексты неявно удаляются, когда вычитается или пересылается последнее оставшееся окончание.

6.2 Окончания

Окончание является логическим объектом шлюза MG, представляющим собой источник и/или приёмник медиапоток и/или потоков управления. Окончание описывается рядом характерных свойств, группирующихся в набор дескрипторов, которые включаются в команды. Окончания имеют уникальные идентификаторы (TerminationIDs), присвоенные шлюзом MG во время их создания.

Окончания, представляющие физические объекты, имеют полупостоянное существование. Например, окончание, представляющее канал TDM, может существовать так долго, как это обеспечивается шлюзом. Окончания, представляющие эфемерные информационные потоки, такие как потоки RTP, обычно будут существовать только в процессе их использования.

Эфемерные окончания создаются при помощи команды Add. Они разрушаются посредством команды Subtract. В противоположность этому, когда окончание присоединяется к или вычитается из контекста, оно поступает из или в нулевой контекст, соответственно.

Окончания могут иметь применимые к ним сигналы (см. 7.1.11). Окончания могут программироваться для обнаружения событий, появление которых может инициировать уведомляющие сообщения к контроллеру MGC, или действие шлюза MG. На окончании может собираться статистика. Статистика сообщается MGC по запросу (посредством команды AuditValue, см. 7.2.5) и тогда окончание берётся из соединения, в котором оно находится.

Мультимедийные шлюзы могут обрабатывать мультиплексированные медиапоток. Например, Рекомендация МСЭ-Т Н.221 описывает структуру кадра для нескольких медиапоток, мультиплексированных из ряда цифровых 64 кбит/с каналов. Подобный случай обрабатывается в модели соединения следующим образом. Для каждого транспортного канала, по которому передаётся часть мультиплексированных потоков, существует физическое или эфемерное "окончание транспортного канала". Окончания транспортных каналов, являющихся источником и/или приёмником цифровых каналов, соединены с отдельным окончанием, называемым "мультиплексированное окончание". Мультиплексированное окончание является эфемерным окончанием, представляющим собой сеанс, связанный с кадром. Дескриптор MultiplexDescriptor для этого окончания описывает используемое мультиплексирование (например, Н.221 для сессии Н.320) и показывает порядок, в котором составляющие цифровые каналы размещаются в кадре.

Мультиплексированные окончания могут быть каскадными (например, мультиплексирование Н.226 цифровых каналов, питающее мультиплексирование Н.223, поддерживающее сессию Н.324).

Отдельные медиапоток, участвующие в сеансе, описываются на мультиплексированном окончании посредством дескрипторов StreamDescriptors. Эти медиапоток могут быть

взаимосвязаны с потоками создаваемыми/принимаемыми окончаниями в контексте ином, чем окончания транспортного канала, поддерживающие мультиплексированное окончание. Каждое окончание транспортного канала поддерживает только единственный поток данных. Эти потоки данных не проявляются явно как потоки на мультиплексированном окончании, а являются скрытыми от остального контекста.

На Рисунках 4, 5, и 6 показаны типичные применения мультиплексированного окончания и дескриптора Multiplex Descriptor.

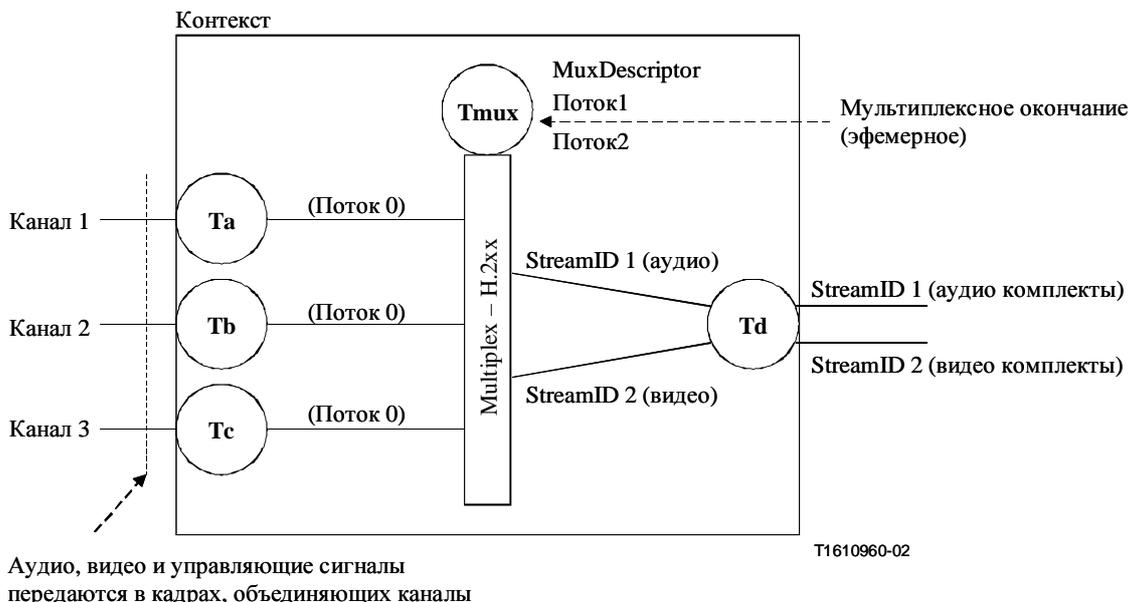


Рисунок 4/Н.248.1 – Сценарий мультиплексированного окончания – Канал с пакетом

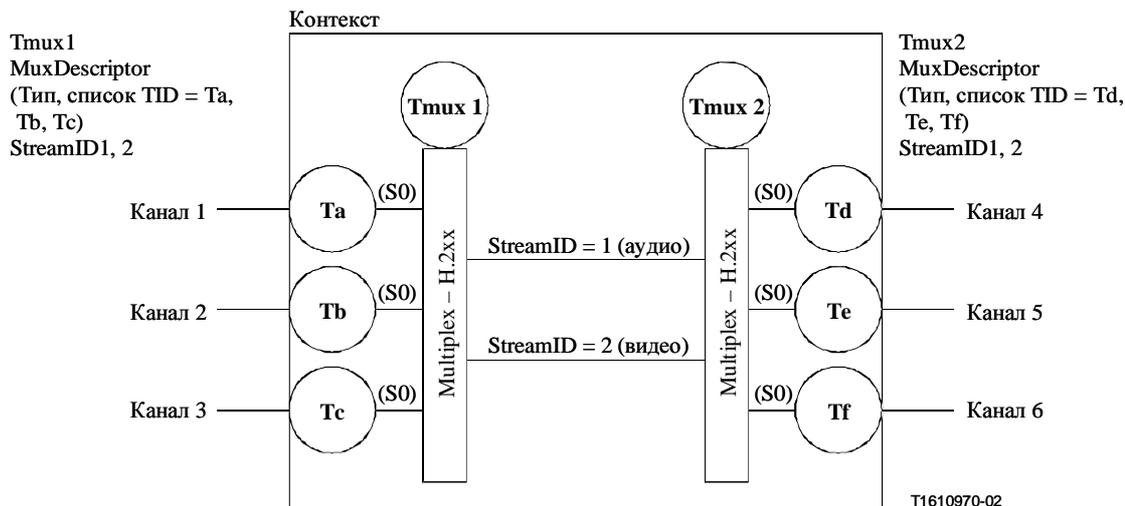


Рисунок 5/Н.248.1 – Сценарий мультиплексированного окончания – Канал с каналом

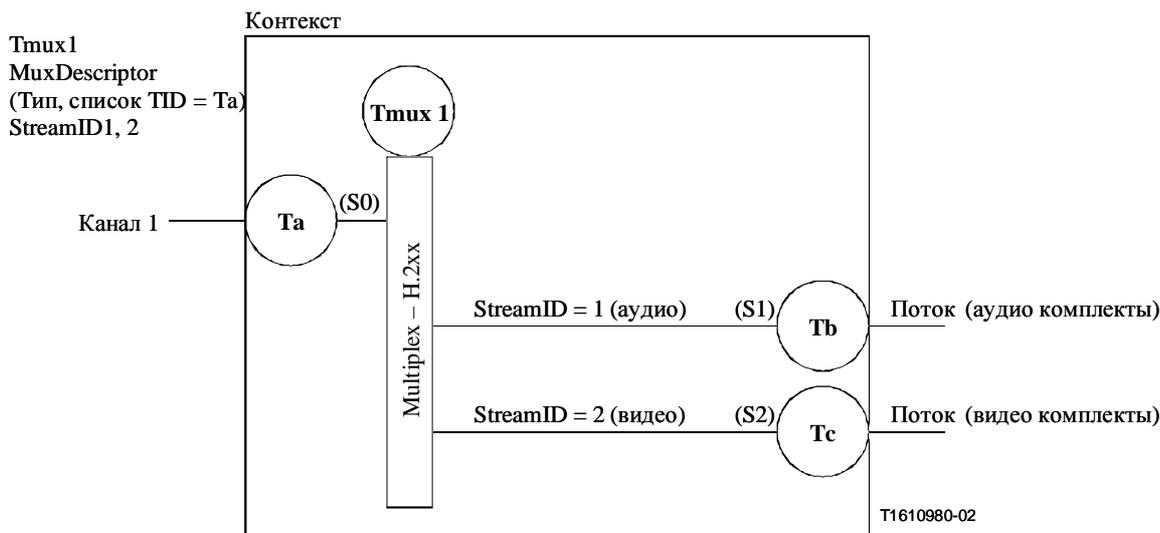


Рисунок 6/Н.248.1 – Сценарий мультиплексированного окончания – Единственное окончание с несколькими

Могут быть созданы окончания, представляющие мультиплексированные транспортные каналы, такие как транспортный канал АТМ ААЛ типа 2. При создании нового мультиплексированного транспортного канала, в контексте, предназначенном для этой цели, создаётся недолговечное окончание. При вычитании окончания мультиплексированный транспортный канал ликвидируется.

6.2.1 Динамика окончания

Протокол может применяться для создания новых окончаний и изменения значений свойств существующих окончаний. Эти изменения включают возможность присоединения и удаления событий и/или сигналов.

Свойства окончаний, событий, сигналов описывается в последующих подразделах. Контроллер МГС может только освободить/изменить окончания и ресурсы, которые представляет окончание и которые были захвачены перед этим.

6.2.2 Идентификаторы TerminationID

Ссылки на окончания осуществляются посредством идентификатора TerminationID (идентификатора окончания), являющегося произвольной схемой, выбранной шлюзом МГ.

Идентификаторы TerminationIDs физических окончаний обеспечиваются в медиашлюзе. Идентификаторы TerminationIDs могут быть выбраны так, чтобы иметь 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 одновременной адресации нескольких окончаний, в то время как последний используется для указания медиашлюзу в отношении выбора окончания, удовлетворяющего частично описанному идентификатору окончания. Это позволяет контроллеру МГС, например, дать команду шлюзу МГ относительно выбора канала в пределах канальной группы.

Когда групповой символ замены ALL используется в идентификаторе TerminationID команды, то результат идентичен повторению команды с каждым из соответствующих идентификаторов TerminationIDs. Применение группового символа замены ALL не относится к окончанию ROOT. В связи с тем, что каждая из этих команд может формировать ответ, размер полного ответа может быть большим. Если отдельные ответы не требуются, может потребоваться ответ с групповым символом замены. В этом случае, формируется единственный ответ, который содержит объединение всех отдельных ответов, которые в противном случае были бы сформированы с подавленными двойными значениями. Например, в случае окончания Ta со свойствами $p1 = a$, $p2 = b$ и окончания Tb со свойствами $p2 = c$, $p3 = d$, объединённый ответ будет состоять из идентификатора TerminationId с групповым символом замены и последовательности свойств $p1 = a$, $p2 = b, c$ и $p3 = d$. Ответ с групповым символом замены может быть особенно полезен в командах Audit.

Кодирование групповых символов замены подробно описывается в Приложениях А и В.

6.2.3 Комплекты

Различные типы шлюзов могут обеспечивать исполнение окончаний со значительно различающимися характеристиками. Различия в окончаниях сглаживаются в протоколе посредством разрешения окончаниям иметь факультативные свойства, события, сигналы и статистику, обеспечиваемую посредством шлюзов MGs.

Для достижения взаимодействия между шлюзом MG и контроллером MGC, такие факультативные свойства группируются в комплекты и окончание обычно осуществляет набор таких комплектов. Более подробная информация об определении пакетов может быть найдена в разделе 12.

Ссылки на свойства, события, сигналы и статистику, определённые в комплектах, а также на их аргументы, осуществляются посредством идентификаторов (Ids). Идентификаторы масштабируются. Для каждого комплекта, идентификаторы PropertyIds, EventIds, SignalIds, StatisticsIds и ParameterIds имеют уникальные именные пространства 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.п.

Для поддержки какого-либо определённого комплекта шлюз MG должен поддерживать все свойства, сигналы, события и статистику, определённые в комплекте. Он также должен поддерживать все аргументы сигнала и события. Шлюз MG может поддерживать подмножество значений определённого свойства или аргумента, перечисленных в комплекте.

При расширении комплектов ссылки на свойства, события, сигналы и статистику, определённые в основном комплекте, могут быть сделаны путём использования либо расширенного, либо основного имени комплекта. Например, если комплект А определяет событие e1, а комплект В расширяет пакет А, тогда В/e1 является событием для окончания, обеспечивающего выполнение комплекта В. По определению шлюз MG также ДОЛЖЕН обеспечить исполнение основного комплекта, однако публикация основного комплекта в качестве разрешённого стыка является факультативной. Если он действительно публикует А, то об А, как и о В, должно быть сообщено в составе дескриптора Package Descriptor команды AuditValue, а событие А/e1 должно быть доступно на окончании. Если шлюз MG не публикует А, то будет доступно только В/e1. Если публикация происходит посредством команды AuditValue, то А/e1 и В/e1 являются одним и тем же событием.

Для улучшенного взаимодействия и ухудшенной совместимости шлюз MG может опубликовать все комплекты, поддерживаемые его окончаниями, включая основные комплекты, из которых образуются расширенные комплекты. Исключением этому являются случаи, когда основные комплекты специально “Разработаны только для расширения”.

6.2.4 Свойства окончания и дескрипторы

Окончания имеют свойства. Свойства имеют уникальные идентификаторы PropertyIDs. Большинство свойств имеют значения по умолчанию, которые явно определяются в данной спецификации протокола, либо в пакете (см. раздел 12), либо устанавливаются посредством инициализации. Если же они не инициализируются иным образом, то, когда окончание впервые создаётся или возвращается нулевому контексту, свойства во всех дескрипторах, за исключением TerminationState и LocalControl, устанавливаются по умолчанию пустыми/"без значения". Содержимое по умолчанию двух исключений описывается в 7.1.5 и 7.1.7.

Инициализация значения свойства в шлюзе MG заменит любое значение, установленное по умолчанию, станет равным значению, приведенному в данной спецификации протокола или в пакете. Поэтому, если необходимо, чтобы контроллер MGC имел полный контроль над значениями свойств окончания, он должен, поставлять явные значения при присоединении окончания к контексту. В качестве альтернативы, в случае физического окончания контроллер MGC может определить любые инициализированные значения свойств путём проверки окончания в то время, как оно находится в нулевом контексте.

Имеется ряд общих свойств окончаний и свойств, характерных для медиапотоков. Эти общие свойства также называются свойствами состояния окончания. Для каждого медиапотока имеются местные свойства и свойства принимаемых и передаваемых потоков.

Свойства, не включённые в основной протокол, определяются в комплектах. К этим свойствам обращаются посредством имени, которое состоит из имени PackageName и идентификатора PropertyId. Большинство свойств имеют значения по умолчанию, описанные в описании комплекта. Свойства могут быть только для чтения или для чтения/записи. Возможные значения свойства могут быть проверены, как и их текущие значения. Контроллер MGC может установить значения свойств, используемых для чтения/записи. Свойство может быть заявлено как "Глобальное", если имеет единственное значение и используется совместно всеми окончаниями комплекта. Для удобства связанные между собой свойства группируются в дескрипторы.

Когда окончание присоединяется к контексту, значения его свойств для чтения/записи могут быть установлены путём включения соответствующих дескрипторов в качестве аргументов команды Add. Аналогично, свойство окончания в контексте может иметь значение, изменённое командой Modify. Свойства также могут иметь изменённые значения, когда окончание пересылается из одного контекста в другой при помощи команды Move. В некоторых случаях дескрипторы возвращаются в виде результата команды.

В общем, если одна из вышеуказанных команд полностью отсутствуют в дескрипторе, свойства в таком дескрипторе сохраняют свои прежние значения для окончаний, на которые действует команда. С другой стороны, если некоторые из свойств для чтения/записи отсутствуют в дескрипторе команды (т.е. дескриптор является лишь частично описанным), эти свойства будут приведены к их значениям по умолчанию для окончаний, на которые действует команда до тех пор, пока пакет не предпримет другое поведение. Более подробные сведения приведены в 7.1, связанным с индивидуальными дескрипторами.

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

Название дескриптора	Описание
Modem	Определяет тип модема и свойства, когда применимо.
Mux	Описывает тип мультиплексора для мультимедийных окончаний (например, Н.221, Н.223, Н.225.0) и окончаний, составляющих входной мультиплексор.
Media	Перечень спецификаций медиапотока (см. 7.1.4).
TerminationState	Свойства окончания (которое может быть определено в пакетах), не являющиеся характерными для потока.
Stream	Перечень дескрипторов Remote/Local/LocalControl одного потока.
Local	Содержит свойства, которые описывают медиапотоки, принимаемые шлюзом MG от удалённого объекта.
Remote	Содержит свойства, которые описывают медиапотоки, передаваемые шлюзом MG на удалённый объект.
LocalControl	Содержит свойства (которые могут быть определены в пакетах), представляющие интерес для шлюза MG и контроллера MGC
Events	Описывает события, обнаруживаемые шлюзом MG, а также то, что надо делать, когда событие обнаружено
EventBuffer	Описывает события, обнаруживаемые шлюзом MG, когда буферизация событий активна
Signals	Описывает сигналы (см. 7.1.11), применимые к окончаниям
Audit	В командах Audit, определяет требуемую информацию
Packages	В команде AuditValue, возвращает список комплектов, исполненных окончанием
DigitMap	Определяет образцы, с которыми должны сопоставляться последовательности определённого набора событий так, чтобы они могли быть зарегистрированы как группа, а не порознь
ServiceChange	В ServiceChange, какое изменение услуги, почему произошло и т.д.
ObservedEvents	В командах Notify или AuditValue, отчёт об отмеченных событиях
Statistics	В командах Subtract и Audit, отчёт о статистике, сохраняемой на окончании
Topology	Описывает направления потоков между окончаниями в контексте
Error	Содержит код ошибки и, факультативно, пояснение ошибки; может встречаться в ответах на команды и в запросах команды Notify

6.2.5 Окончание Root

Иногда команда должна относиться к целому шлюзу, а не к окончанию в его пределах. Для этой цели зарезервирован специальный идентификатор TerminationID (идентификатор окончания), названный "Root" ("Корневой"). Комплекты могут определяться на идентификаторе Root. Таким образом корневой идентификатор может иметь свойства, события и статистику (сигналы не подходят для корневого). Таким образом, корневой TerminationID может появиться в:

- команде Modify – для изменения свойства или набора событий;
- команде Notify – для уведомления о событии;
- возврате команды AuditValue – для проверки значений свойств и статистики, реализованных на корне;
- команде AuditCapability – для определения реализованных свойств корня;
- команде ServiceChange – для объявления о рабочем или нерабочем состоянии шлюза.

Любое иное использование корневого идентификатора TerminationID является ошибочным. В этих случаях должен возвращаться код ошибки 410 (неправильный идентификатор).

7 Команды

Протокол обеспечивает команды для изменения логических объектов модели соединения протокола, контекстов и окончаний. Команды обеспечивают управление на самом мелком уровне модульности, поддерживаемом протоколом. Например, существуют команды для присоединения окончаний к контексту, изменения окончаний, вычитания окончаний из контекста, аудита свойств контекстов и окончаний. Команды предусматривают полное управление свойствами контекстов и окончаний. Это включает указание, каким событиям окончание должно представлять сообщения, какие сигналы/действия должны быть применимы к окончанию, а также описание топологии контекста (кто кого слышит/видит).

Большинство команд специально предназначено для использования стыком медиашлюза, как инициатором команд, а также в управляющих медиашлюзах, как формирователях ответов на команды. Исключениями являются команды Notify и ServiceChange: команда Notify посылается из медиашлюза в контроллер медиашлюза, а команда ServiceChange может быть послана любым объектом. Ниже приведен обзор команд; более детальные пояснения даются в 7.2.

- 1) Add – Команда Add (присоединять) присоединяет окончание к контексту. Команда Add на первом окончании в контексте используется для создания контекста.
- 2) Modify – Команда Modify (изменять) изменяет свойства, события и сигналы окончания.
- 3) Subtract – Команда Subtract (вычитать) разъединяет окончание со своим контекстом и возвращает статистику участия окончания в контексте. Команда Subtract на последнем окончании контекста удаляет контекст.
- 4) Move – Команда Move (переслать) автоматически пересылает окончание в другой контекст.
- 5) AuditValue – Команда AuditValue (проверять значение) возвращает текущее состояние свойств, событий, сигналов и статистики окончаний.
- 6) AuditCapabilities – Команда AuditCapabilities (проверять возможности) возвращает все возможные значения свойств окончания, событий и сигналов, разрешённые медиашлюзом.
- 7) Notify – Команда Notify (уведомлять) разрешает медиашлюзу проинформировать контроллер медиашлюза о наличии событий в медиашлюзе.
- 8) ServiceChange – Команда ServiceChange (изменять службу) разрешает медиашлюзу уведомить контроллер медиашлюза о том, что окончание или группа окончаний собираются быть установленными в нерабочее состояние или только что были установлены в рабочее состояние. Команда ServiceChange также используется шлюзом MG для объявления своей доступности контроллеру MGC (регистрация) и уведомления контроллера MGC о предстоящей или завершившейся перезагрузке шлюза MG. Контроллер MGC может объявить переключение шлюзу MG посредством отправки ему команды ServiceChange. Контроллер MGC может также использовать команду ServiceChange для указания шлюзу MG об установке окончания или группы окончаний в рабочее или нерабочее состояние.

Эти команды подробно описываются в разделах 7.2.1 - 7.2.8.

7.1 Дескрипторы

Аргументы команды называются дескрипторами. Дескриптор содержит имя и список элементов. Некоторые элементы могут иметь числовые значения. Многие команды совместно используют общие дескрипторы. В настоящем подразделе перечисляются эти дескрипторы. Дескрипторы могут возвращаться как результат команды. При любом таком возврате содержания дескрипторов пустой дескриптор представляется своим именем, не сопровождаемым никаким списком. Аргументы и их применение, характерное для данного типа команды, приведены в подразделе, в котором описывается команда.

7.1.1 Определяющие аргументы

Аргументы команды структурированы в ряд дескрипторов. В общем, текстовый формат дескрипторов имеет вид `DescriptorName=<someID>{parm=value, parm=value....}`.

Аргументы могут быть полностью определены, переопределены и недоопределены:

- 1) Полностью определённые аргументы имеют единственное, точно определённое значение, которое инициатор команды указывает использовать командному ответчику для определённого аргумента.
- 2) Не полностью определённые аргументы, использующие значение CHOOSE, позволяют командному ответчику выбрать любое значение, которое он может поддерживать.
- 3) Сверхопределённые аргументы имеют список потенциальных значений. Порядок расположения в листе определяет предпочтительный порядок выбора командного инициатора. Командный ответчик выбирает одно значение из предложенного списка и возвращает это значение командному инициатору.

Если необходимый дескриптор иной, чем дескриптор Audit, точно не установлен из команды (т.е. полностью отсутствует), предыдущие значения, установленные в таком дескрипторе для этого окончания, если это имеет место, сохраняются. В иных командах, чем Subtract, пропущенный дескриптор Audit равнозначен пустому дескриптору Audit. Поведение шлюза MG относительно недоопределённых аргументов внутри дескриптора различно для конкретного дескриптора, как это указывается в последующих подразделах. Всякий раз, когда аргумент является недоопределённым или переопределённым, дескриптор, содержащий значение, выбранное ответчиком, включается в результат команды.

Каждая команда определяет идентификатор TerminationId, на котором работает команда. Идентификатор TerminationId может содержать "групповой символ замены". Когда идентификатор TerminationId команды содержит групповой символ замены, эффект должен быть таким же, как при повторении команды с каждым из подходящих идентификаторов TerminationIds.

7.1.2 Дескриптор Modem

Дескриптор Modem определяет тип модема и аргументы, если имеются, необходимые для применения, например, в H.324 и текстовом диалоге. Дескриптор включает следующие типы модемов: V.18, V.22, V.22 bis, V.32, V.32 bis, V.34, V.90, V.91, синхронная ЦСИС и предусматривает расширения. По умолчанию, дескриптор Modem в окончании отсутствует.

7.1.3 Дескриптор Multiplex

В мультимедийных соединениях ряд медиапотокa передается некоторым количеством (возможно различных) транспортных каналов. Дескриптор Multiplex (мультиплексирование) связывает среду и транспортные каналы. Дескриптор включает мультиплексированные типы:

- H.221;
- H.223;

- Н.226;
- V.76;
- возможные расширения,

а также набор идентификаторов TerminationIDs, представляющих собой мультиплексированные транспортные каналы по порядку. Например:

Мих = Н.221{ МутЗ/1/2, МутЗ/2/13, МутЗ/3/6, МутЗ/21/22 }

7.1.4 Дескриптор Media

Дескриптор Media (среда передачи) определяет аргументы для всех медиапотоков. Эти аргументы структурированы в двух дескрипторах: дескрипторе TerminationState, который определяет потоко-независимые свойства окончания, а также одним или более дескрипторах Stream, каждый из которых описывает единственный медиапоток.

Поток идентифицируется посредством StreamID. Идентификатор StreamID используется для связи потоков в том контексте, которому они все вместе принадлежат. Несколько потоков, исходящих из окончания должны быть синхронизированы друг с другом. Внутри дескриптора Stream имеется до трёх вспомогательных дескрипторов: LocalControl, Local и Remote. Таким образом, взаимосвязь между этими дескрипторами имеет следующий вид:

Дескриптор Media

Дескриптор TerminationState

Дескриптор Stream

Дескриптор LocalControl

Дескриптор Local

Дескриптор Remote

Для удобства дескрипторы LocalControl, Local и Remote могут быть включены в дескриптор Media без включающего дескриптора Stream. В этом случае подразумевается, что идентификатор потока равен 1.

7.1.5 Дескриптор TerminationState

Дескриптор TerminationState (состояние окончания) содержит свойство ServiceStates, свойство EventBufferControl, а также свойства окончания (определяемые в разделе о комплектах), не являющиеся потоко-ориентированными.

Свойство ServiceStates описывает общее состояние окончания (не потоко-ориентированными). Окончание может быть в одном из следующих состояний: "проверка", "нерабочее" или "рабочее". Состояние "проверка" показывает, что окончание проверяется. Состояние "нерабочее" показывает, что окончание не может быть использовано для трафика. Состояние "рабочее" показывает, что окончание может быть использовано или используется для обычного трафика. Состоянием по умолчанию является "рабочее" состояние.

Значения, присвоенные свойствам, могут быть простыми величинами (целое число/ряд/перечисление) или могут быть неопределёнными, когда поставляется более, чем одно значение и шлюз MG может сделать выбор:

- Альтернативные значения – перечень разнообразных значений, одно из которых должно быть выбрано;
- Пределы – минимальное и максимальная значения, должно быть выбрано любое значение между минимальным и максимальным, включая предельные значения;

- Больше, чем/Меньше, чем – значение должно быть больше/меньше, чем указанная величина;
- Групповой символ замены CHOOSE – шлюз MG выбирает из разрешённых значений свойства.

Свойство EventBufferControl определяет поступают ли события в буфер после обнаружения события в дескрипторе Events или сразу же обрабатываются. Подробности см. в 7.1.9.

7.1.6 Дескриптор Stream

Дескриптор Stream (поток) определяет аргументы единственного двустороннего потока. Эти аргументы структурированы в три дескриптора: один из них содержит свойства окончания, характерные для потока, а каждый из оставшихся – для локального и удалённого потоков. Дескриптор Stream включает в себя идентификатор StreamID, определяющий данный поток. Потоки создаются путем определения нового идентификатора StreamID на одном из окончаний в контексте. Поток удаляется путём установки пустых дескрипторов Local и Remote для потока со свойствами ReserveGroup и ReserveValue в дескрипторе LocalControl, установленными в значение "Ложь" на всех окончаниях в контексте, который ранее поддерживал этот поток.

Идентификаторы StreamIDs присваиваются контроллером MGC и имеют местное значение между контроллером MGC и шлюзом MG. В пределах контекста идентификатор StreamID является средством указания на то, какие из потоков носителей соединены: потоки с одинаковыми идентификаторами StreamID являются соединёнными.

Если окончание пересылается из одного контекста в другой, влияние на контекст, в который окончание пересылается, является таким же, как и в случае, когда присоединяется новое окончание с теми же идентификаторами StreamIDs, как и у пересланного окончания.

7.1.7 Дескриптор LocalControl

Дескриптор LocalControl (локальное управление) содержит свойство Mode, свойства ReserveGroup и ReserveValue, а также потоко-ориентированные свойства окончания (определённые в разделе о комплектах), играющие роль между шлюзом MG и контроллером MGC. Значения свойств могут быть неопределёнными, как в 7.1.1.

Разрешёнными значениями для свойства Mode являются только передача, только приём, передача/приём, пассивный и обратная связь. "Передача" и "приём" понимаются относительно внешней стороны контекста, таким образом, что, например, поток, установленный в режим = sendOnly, не проходит приёмную среду в контексте. Значением по умолчанию для свойства Mode является "Пассивный". Сигналы и события не зависят от свойства Mode.

Свойства окончания типа Reserve, а именно ReserveValue и ReserveGroup, принимающие булевы значения, показывают, что ожидается должен делать шлюз MG, когда он принимает дескрипторы Local и/или Remote.

Если свойство типа Reserve установлено в значение "Правда", шлюз MG ДОЛЖЕН зарезервировать ресурсы для всех альтернатив, определённых в дескрипторах Local и/или Remote, для которых в настоящее время у шлюза имеются доступные ресурсы. Шлюз ДОЛЖЕН ответить альтернативами, для которых он зарезервировал ресурсы. Если шлюз не может поддержать никакой альтернативы, он ДОЛЖЕН ответить контроллеру MGC, что содержит пустые дескрипторы Local и/или Remote. Если среда передачи начинает движение в то время, как зарезервирована более, чем одна альтернатива, пакеты среды передачи могут быть переданы/приняты на любой из альтернатив и должны быть обработаны, хотя только одна альтернатива может быть активной в данное время.

Если свойство типа Reserve установлено в значение "Ложь", шлюз MG ДОЛЖЕН выбрать одну из альтернатив, описанных в локальном дескрипторе (при наличии) и одну из альтернатив, описанных в дескрипторе Local (при наличии). Если шлюз MG ещё не зарезервировал ресурсы для поддержки выбранной альтернативы, он ДОЛЖЕН зарезервировать ресурсы. С другой стороны, если шлюз уже зарезервировал ресурсы для адресуемого окончания (вследствие предыдущего обмена со свойствами ReserveValue и/или ReserveGroup, установленными в значение "Правда"), он ДОЛЖЕН высвободить любое превышение ресурсов, зарезервированных им ранее. Окончательно шлюз MG должен послать ответ контроллеру MGC, содержащему альтернативы для дескрипторов Local и/или Remote, которые он выбрал. Если шлюз MG не имеет достаточных ресурсов для поддержки какой-либо из описанных альтернатив, он ДОЛЖЕН ответить ошибкой 510 (недостаточные ресурсы).

"Ложь" является значением по умолчанию свойств ReserveValue и ReserveGroup. Дополнительная информация по использованию этих двух свойств типа Reserve предоставлена в 7.1.8.

Новые установки дескриптора LocalControl полностью заменяют предыдущие установки этого дескриптора в шлюзе MG. Таким образом, для сохранения информации о предыдущей установке контроллер MGC должен включить эту информацию в новую установку. Если контроллер MGC желает удалить некоторую информацию из существующего дескриптора, он просто пересылает дескриптор (в команде Modify) с удалённой нежелательной информацией.

7.1.8 Дескрипторы Local и Remote

Контроллер MGC использует дескрипторы Local (локальный) и Remote (удалённый) для резервирования и предоставления ресурсов декодирования и кодирования среды передачи шлюза MG для данных потока(ов) и окончания, к которым они применяются. Шлюз MG включает эти дескрипторы в свой ответ для указания своей действительной готовности к поддержке. Шлюз MG ДОЛЖЕН включить дополнительные свойства и их значения в свой ответ даже, если эти свойства не обязательно присутствуют в запросах, сделанных контроллером MGC (например, путём указания подробных аргументов видео кодирования, когда контроллер MGC указал только тип нагрузки).

Дескриптор Local относится к среде, принятой шлюзом MG, а дескриптор Remote – к среде, переданной шлюзом MG.

При кодировании текста протокола дескрипторы состоят из дескрипторов сеансов, как определено в SDP (IETF RFC 2327). В описаниях сеансов, передаваемых от контроллера MGC к шлюзу MG, допускаются следующие исключения из синтаксиса IETF RFC 2327:

- строки "s=", "t=" and "o=" являются факультативными;
- допускается использование группового символа замены CHOOSE вместо единственного значения аргумента; и
- допускается использование альтернатив вместо единственного значения аргумента.

Дескриптор Stream определяет единственный двусторонний медиапоток, а поэтому описание единственного сеанса НЕ ДОЛЖНО включать более одного описания среды ("m=" строка). Дескриптор Stream может содержать, в качестве альтернатив, описания дополнительных сеансов. Каждый медиапоток для какого-либо окончания должен появляться в отдельных дескрипторах Stream. Когда в одном дескрипторе даются описания разнообразных сессий, в качестве разделителей необходимы строки "v="; в противном случае в описаниях сеансов, передаваемых в шлюз MG, они являются факультативными. Реализации должны допускать описания сессий, полностью соответствующих IETF RFC 2327. Как описывается в

Приложении С, при двоичном кодировании протокола дескриптор состоит из групп свойств (пар тег-значение). Каждая такая группа может содержать аргументы описания сеанса.

Семантика дескрипторов Local и Remote детально описываются ниже. Спецификация состоит из двух частей. В первой части приводится трактовка содержания дескриптора. Во второй части описываются те действия, которые должен выполнить шлюз MG при получении дескрипторов Local и Remote. Действия, которые должны быть выполнены шлюзом MG, зависят от значений свойств ReserveValue и ReserveGroup дескриптора LocalControl.

Дескрипторы Local и Remote могут быть либо оба, либо по отдельности:

- не определены (т.е. отсутствовать);
- пустые;
- недоопределены вследствие использования группового символа замены CHOOSE в значении свойства;
- полностью определены; или
- переопределены путём представления разнообразных групп свойств и возможно разнообразных значений свойств в одной или более из этих групп.

Там, где дескрипторы проходят от контроллера MGC к шлюзу MG, они трактуются в соответствии с правилами, приведенными в 7.1.1, со следующими дополнительными поясняющими комментариями:

- Предполагается, что в неопределённых дескрипторах Local и Remote отсутствует обязательный аргумент. Необходимо, чтобы шлюз MG использовал любой аргумент, определённый для этого дескриптора. Возможно, что значение, определённое ранее, отсутствует; в этом случае упомянутый дескриптор пропускается при дальнейшей обработке команды.
- Пустой дескриптор Local (Remote) в сообщении от контроллера MGC означает запрос на освобождение любых ресурсов, зарезервированных для принятого (переданного) медиапотока.
- Если в дескрипторах Local или Remote имеется несколько групп свойств или несколько значений в пределах группы, предпочтительный порядок- нисходящий.
- Недоопределённые или переопределённые свойства в пределах группы свойств, переданные контроллером MGC, являются запросами к шлюзу MG о выборе одного или более значений, которые он может поддержать для каждого из этих свойств. В случае переопределённого свойства список значений- в нисходящем порядке предпочтения.

При выполнении вышеуказанных правил последующее действие зависит от значений свойств ReserveValue и ReserveGroup в дескрипторе LocalControl.

Если свойство ReserveGroup установлено в значение "Правда", шлюз MG резервирует необходимые ресурсы, чтобы поддержать любые требуемые альтернативы групп свойств, которые он может поддержать в настоящее время. Если свойство ReserveValue установлено в значение "Правда", шлюз MG резервирует необходимые ресурсы, чтобы поддержать любые требуемые альтернативы групп свойств, которые он может поддержать в настоящее время.

ПРИМЕЧАНИЕ – Если дескрипторы Local и Remote содержат несколько групп свойств, а свойство ReserveGroup установлено в значение "Правда", то шлюзу MG необходимо зарезервировать ресурсы таким образом, чтобы он мог декодировать или кодировать медиапоток в соответствии с любой из альтернатив. Например, если локальный дескриптор содержит две группы свойств, одна-определяющая пакетизированный G.711 А-закон аудио, а другая- G.723.1 аудио, шлюз MG

резервирует ресурсы таким образом, чтобы он мог декодировать один аудио поток, закодированный либо в формате G.711 А-закона, либо в формате G.723.1. Шлюз MG не должен резервировать ресурсы для одновременного декодирования двух аудио потоков, одного- кодированного в соответствии с G.711 А-законом, и другого- в соответствии с G.723.1. Стремление в отношении использования свойства "ReserveValue- аналогично.

Если свойство ReserveGroup установлено в значение "Правда", или свойство ReserveValue установлено в значение "Правда", то применяются следующие правила:

- Если шлюз MG не имеет достаточных ресурсов для поддержки всех альтернатив, а также ресурсов в дескрипторах Local и Remote, запрошенных контроллером MGC, шлюз MG должен зарезервировать ресурсы для поддержки, по крайней мере, одной альтернативы в пределах каждого дескриптора Local и Remote.
- Если шлюз MG не имеет достаточных ресурсов для поддержки хотя бы одной альтернативы в пределах дескриптора Local (Remote), принятого от контроллера MGC, он должен вернуть в ответ пустой дескриптор Local (Remote).
- В своём ответе контроллеру MGC, когда контроллер MGC включает дескрипторы Local и Remote, шлюз MG ДОЛЖЕН включить дескрипторы Local и Remote для всех групп свойств и значений свойства, для которых шлюз зарезервировал ресурсы. Если шлюз MG не способен поддержать хотя бы одну из альтернатив в пределах дескриптора Local (Remote), принятого от MGC, он ДОЛЖЕН вернуть пустой дескриптор Local (Remote).
- Если свойство Mode дескриптора LocalControl находится в положениях RecvOnly, SendRecv, или LoopBack, то шлюз MG должен подготовиться для приёма среды, кодированной в соответствии с любой из альтернатив, включённых в свой ответ контроллеру MGC.

Если свойство ReserveGroup установлено в значение "Ложь", а свойство ReserveValue- в значение "Ложь", то MG ДОЛЖЕН применять следующие правила для приведения каждого дескриптора Local и Remote к единственной альтернативе:

- Шлюз MG выбирает первую альтернативу в дескрипторе Local, для которого он может поддержать по крайней мере одну альтернативу в дескрипторе Remote .
- Если шлюз MG не может поддержать по крайней мере одну Local и одну Remote альтернативу, он возвращает ошибку 510 (недостаточные ресурсы).
- Шлюз MG возвращает выбранную им альтернативу, в каждый дескриптор Local и Remote.

Новая установка дескриптора Local или Remote полностью заменяет предыдущую установку этого дескриптора в шлюзе MG. Таким образом, для того, чтобы сохранить информацию из предыдущей установки контроллер MGC должен включить эту информацию в новую установку. Если контроллер MGC хочет удалить некоторую информацию из существующего дескриптора, он просто передаёт дескриптор заново (в команде Modify) с вычеркнутой нежелательной информацией.

7.1.9 Дескриптор Events

Аргумент EventsDescriptor (дескриптор событий) содержит идентификатор RequestIdentifier и перечень событий, которые медиашлюзу задано обнаружить и составить отчёт. Идентификатор RequestIdentifier используется для соответствия запроса с уведомлениями, которые он может вызвать. Заданные события включают, например, тональные сигналы факса, непрерывность результатов измерений, а также переходы между состояниями снятой и положенной трубки. Идентификатор RequestIdentifier опускается, если аргумент EventsDescriptor является пустым (т.е. при отсутствии описанных событий).

Каждое событие в дескрипторе содержит имя события, факультативный идентификатор StreamID, факультативный признак KeepActive, а также факультативные аргументы. Имя события состоит из имени комплекта (в котором определено событие) и идентификатора EventID. Все групповые символы замены могут быть использованы для идентификатора EventID, показывающие, что все события из указанного пакета должны быть обнаружены. Значение идентификатора StreamID по умолчанию, равное 0, показывает, что событие, которое должно быть обнаружено не относится к данному медиапотoku. События могут иметь аргументы. Это позволяет единственному описанию события иметь некоторый разброс в значении без создания большого числа индивидуальных событий. Дальнейшие аргументы события определены в комплекте.

Если завершающее событие плана нумерации присутствует или выражено неявно в аргументе EventsDescriptor, аргумент EventDM используется для передачи либо имени, либо значения соответствующего плана нумерации. Более подробные сведения приведены в 7.1.14.

Когда событие обработано относительно содержимого активного дескриптора Events и найдено присутствующим в этом дескрипторе ("распознано"), действием шлюза MG по умолчанию является передача команды Notify в контроллер MGC. Уведомление может быть отложенным, если событие включается в текущую последовательность набора номера активного плана нумерации (см. 7.1.14). Любое другое действие является предметом дальнейшего изучения. Более того, распознавание события может вызвать прекращение сигналов, действующих в настоящее время, или может привести к замене текущего дескриптора событий и/или сигналов, как это описано в конце данного подраздела. Пока дескриптор событий не заменяется другим дескриптором событий он остаётся активным после распознавания события.

Если после обнаружения такого события значение свойства EventBufferControl становится равным LockStep, обычная обработка событий приостанавливается. Любое событие, которое обнаруживается впоследствии и происходит в дескрипторе EventBuffer, присоединяется к окончанию EventBuffer (очередь типа FIFO) с указанием времени обнаружения. Шлюз MG ДОЛЖЕН ждать загрузки нового аргумента EventsDescriptor. Новый аргумент EventsDescriptor может быть загружен либо в результате приёма команды с новым аргументом EventsDescriptor, либо путём активизации вложенного аргумента EventsDescriptor.

Если свойство EventBufferControl находится в состоянии Off, шлюз MG продолжает обработку, основанную на активном аргументе EventsDescriptor.

В случае активизации вложенного аргумента EventsDescriptor, шлюз MG продолжает обработку события, основанную на вновь активизированном аргументе EventsDescriptor.

ПРИМЕЧАНИЕ 1 – С целью обработки дескриптора EventBuffer активизация вложенного аргумента EventsDescriptor эквивалентна приёму нового аргумента EventsDescriptor.

Когда MG принимает команду с новым аргументом EventsDescriptor, одно или более событий могут быть буферизованы в дескрипторе EventBuffer в шлюзе MG. Тогда значение свойства EventBufferControl определяет как шлюз MG обрабатывает такие буферизованные события.

Пример 1

Если свойство EventBufferControl равняется LockStep и шлюз MG принимает новый аргумент EventsDescriptor, он проверит очередь FIFO дескриптора EventBuffer и предпримет следующие действия:

- 1) Если дескриптор EventBuffer является пустым, шлюз MG ожидает событий, базирующихся на новом аргументе EventsDescriptor.

- 2) Если дескриптор EventBuffer не является пустым, шлюз MG обрабатывает очередь FIFO, начиная с первого события:
- a) Если событие в очереди находится в списке событий нового аргумента EventsDescriptor, шлюз MG действует в соответствии с этим событием и удаляет событие из дескриптора EventBuffer. Отметка времени команды Notify должна быть временем действительного обнаружения события. Затем шлюз MG ожидает новый аргумент EventsDescriptor. На время ожидания нового аргумента EventsDescriptor, любые обнаруженные события, которые появляются в EventsBufferDescriptor, будут помещаться в дескриптор EventBuffer. При приёме нового аргумента EventsDescriptor, обработка события будет повторяться, начиная с шага 1).
 - b) Если событие не является частью нового аргумента EventsDescriptor, шлюз MG ДОЛЖЕН отклонить событие и повторить с шага 1).

Пример 2

Если свойство EventBufferControl равняется Off и шлюз MG принимает аргумент EventsDescriptor, то он обрабатывает новые события с новым аргументом EventsDescriptor.

Если шлюз MG принимает команду, которая указывает ему установить значение свойства EventBufferControl, равное Off, все события в дескрипторе EventBuffer ДОЛЖНЫ отклоняться.

Шлюз MG может сообщать в одной транзакции о нескольких событиях до тех пор, пока это излишне не задерживает сообщение об отдельных событиях.

Для процедур, касающихся передачи команды Notify, следует обращаться к соответствующему приложению или Рекомендации серии H.248.x в отношении характерных транспортных соображений.

Значение по умолчанию для свойства EventBufferControl равно Off.

ПРИМЕЧАНИЕ 2 – Т.к. свойство EventBufferControl находится в дескрипторе TerminationStateDescriptor, шлюз MG может принять команду, которая изменяет свойство EventBufferControl и не включает аргумент EventsDescriptor.

Обычно, распознавание события должно вызвать прекращение любых активных сигналов. Когда в событии указывается KeepActive, шлюз MG не должен прерывать никакие активные сигналы на том окончании, где это событие обнаружено.

Событие может включать дескриптор вложенных сигналов и/или дескриптор вложенных событий, который, в случае его наличия, заменяет текущий дескриптор сигналов/событий при распознавании события. Возможно, например, указать, что сигнал ответа станции будет генерироваться при распознавании события снятия трубки или, что сигнал ответа станции будет прекращён при распознавании цифры. Контроллер медиашлюза не должен передавать дескрипторы EventsDescriptors с событием, если оба помечены KeepActive и содержат вложенный дескриптор SignalsDescriptor.

Разрешается только один уровень вложения. Вложенный дескриптор EventsDescriptor НЕ ДОЛЖЕН содержать другой вложенный дескриптор EventsDescriptor; вложенный дескриптор EventsDescriptor МОЖЕТ содержать вложенный дескриптор SignalsDescriptor.

Дескриптор EventsDescriptor, принятый медиашлюзом, замещает любой предыдущий дескриптор событий. Текущее уведомление о событии должно завершиться, а события, обнаруженные после команды, содержащей инструкции нового дескриптора EventsDescriptor, должны быть обработаны в соответствии с новым дескриптором EventsDescriptor.

Пустой дескриптор Events Descriptor запрещает распознавание всех событий и отчётность. Пустой дескриптор EventBuffer очищает дескриптор EventBuffer и запрещает накопление всех событий в режиме LockStep: единственными сообщёнными событиями будут те, которые происходят пока дескриптор Events является активным. Если пустой дескриптор Events активизируется пока окончание работает в режиме LockStep, то буфер событий немедленно очищается.

7.1.10 Дескриптор EventBuffer

Дескриптор EventBuffer (буфер событий) содержит список событий с их аргументами, в случае их наличия, запрошенных шлюзом MG для обнаружения и буферизации при значении свойства EventBufferControl, равном LockStep (см. 7.1.9).

7.1.11 Дескриптор Signals

Сигналы представляют собой среду передачи, генерируемую MG, например, тональные сигналы и объявления, а также сигналы, относящиеся к транспортным каналам, такие как переключатель рычага телефонной трубки. Более сложные сигналы могут включать последовательность таких простых сигналов, как распределённые и восстановленные на приёме с учётом анализа среды, или сигналов, связанных с транспортными каналами. Примеры включают повторение принятых данных, как например, в комплекте Continuity Test. Сигналы также могут запросить подготовку содержания среды для будущих сигналов.

Дескриптор SignalsDescriptor является аргументом, содержащим набор сигналов, которые медиашлюз применяет к окончанию по запросу. Аргумент SignalsDescriptor содержит набор сигналов и/или списков последовательных сигналов. Аргумент SignalsDescriptor может содержать нулевые сигналы и списки последовательных сигналов. Поддержка списков последовательных сигналов является факультативной.

Сигналы определяются в комплектах. Сигналам должны быть присвоены имя комплекта (в котором определяется данный сигнал) и идентификатор SignalID. В идентификаторе SignalID не должны использоваться групповые символы замены. Сигналы, содержащиеся в аргументе SignalsDescriptor, имеют факультативный идентификатор StreamID (значение по умолчанию равно 0, чтобы показать, что сигнал не относится к индивидуальному медиапотoku), факультативный тип сигнала (см. ниже), факультативную длительность и возможные параметры, определённые в комплектах, определяющих сигнал. Это позволяет отдельному сигналу иметь некоторый разброс в значении, избегая необходимость создания большого числа индивидуальных сигналов.

В заключение, факультативный аргумент "notifyCompletion" позволяет MGC указать, что он хочет быть уведомленным, когда сигнал закончит существование. Возможными примерами является выход сигнала за временные рамки (или, иначе, самостоятельно прекращение), или прерывание сигнала каким-либо событием, или прекращение сигнала в результате замены дескриптора сигналов, или то, что сигнал прекратился или никогда не начинался по другим причинам. Если аргумент notifyCompletion не включён в дескриптор сигналов, уведомление генерируется только, если сигнал прекратился или никогда не начинался по другим причинам. Для осуществления отчётности в текущем активном дескрипторе событий должно быть разрешено событие завешения сигнала (см. E.1.2).

Длительность равна целой величине, которая выражается в сотых долях секунды.

Существует три типа сигналов:

- on/off – сигнал длится до тех пор, пока не выключится;
- timeout – сигнал длится до тех пор, пока не выключится или не закончится определённый интервал времени;

- `brief` – сигнал прекратится самостоятельно, если не будет использоваться новый дескриптор, который вызовет его остановку; величина интервала времени не требуется.

Аргумент длительности Если сигнал по умолчанию, отличающийся от `TO`, имеет свой тип в результате замещения типом `TO` в дескрипторе сигналов, то должен присутствовать аргумент длительности.

Если тип сигнала определяется в аргументе `SignalsDescriptor`, то он замещает тип сигнала по умолчанию (см. 12.1.4). Если длительность указывается для сигнала `on/off`, она ДОЛЖНА пропускаться.

Список последовательного сигнала состоит из идентификатора списка сигнала и последовательности сигналов, которые будут воспроизводиться последовательно. Только замыкающий элемент последовательности сигналов в списке последовательного сигнала может быть сигналом `on/off`. Длительность списка последовательного сигнала равна сумме длительностей сигналов, из которых он состоит.

Несколько сигналов и списки последовательного сигнала в одном и том же аргументе `SignalsDescriptor` должны воспроизводиться одновременно.

Сигналы определяются как дела от окончания в направлении к вне контекста до тех пор, пока не определены в комплекте по иному. Когда один и тот же сигнал применяется к разнообразным окончаниям в пределах одной транзакции, `MG` должен рассматривать использование того же самого ресурса для генерации этих сигналов.

Порождение сигнала на окончании прекращается посредством применения нового аргумента `SignalsDescriptor`, или обнаружения события на окончании (см. 7.1.9).

Новый аргумент `SignalsDescriptor` заменяет любой существующий аргумент `SignalsDescriptor`. Любые сигналы, применимые к окончанию не в заменяющем дескрипторе, должны прекращаться, а применяться новые сигналы, за исключением следующего. Сигналы, присутствующие в дескрипторе замены и содержащие признак `KeepActive`, должны продолжаться, если в текущий момент они воспроизводятся и ещё не закончены. Если дескриптор замены содержит список последовательного сигнала с тем же самым идентификатором, как и существующий дескриптор, тогда:

- тип сигнала и последовательность сигналов в списке последовательного сигнала дескриптора замены должны быть пропущены; и
- воспроизведение сигналов в списке последовательного сигнала существующего дескриптора не должно прерываться.

7.1.12 Дескриптор `Audit`

Дескриптор `Audit` (проверка) определяет, какая информация должна быть проверена. Дескриптор `Audit` определяет перечень дескрипторов, которые должны быть возвращены. Проверка может быть использована в любой команде, чтобы принудить возврат любого дескриптора, содержащего текущие значения его свойств, событий, сигналов и статистики даже, если этот дескриптор отсутствовал в команде или не имел недоопределённых аргументов. Возможными элементами в дескрипторе `Audit` являются:

Modem
Mux
Events
Media
Signals

ObservedEvents
DigitMap
Statistics
Packages
EventBuffer

Проверка может быть пустой, при этом дескрипторы не возвращаются. Это является полезным в команде Subtract, чтобы запретить возврат статистики, в особенности при использовании групповых символов замены.

7.1.13 Дескриптор ServiceChange

Дескриптор ServiceChangeDescriptor (изменение службы) содержит следующие аргументы:

- ServiceChangeMethod
- ServiceChangeReason
- ServiceChangeAddress
- ServiceChangeDelay
- ServiceChangeProfile
- ServiceChangeVersion
- ServiceChangeMGCIId
- TimeStamp
- Extension

См. 7.2.8.

7.1.14 Дескриптор DigitMap

7.1.14.1 Определение дескриптора DigitMap, его создание, изменение и удаление

Дескриптор DigitMap представляет собой план нумерации, резидентно находящийся в медиашлюзе и используемый для обнаружения и сообщения о номерных событиях, принятых на окончании. Дескриптор DigitMap содержит название DigitMap и присвоенный DigitMap. План нумерации может быть заранее загружен в медиашлюз администрацией и найден по имени в аргументе EventsDescriptor, может быть определён динамически и в последующем найден по имени, или же действующий план нумерации может быть определён в аргументе EventsDescriptor. Завершающему событию плана нумерации внутри дескриптора Events разрешается делать ссылки на DigitMap по имени, которое определяется дескриптором внутри той же самой команды, независимо от порядка передачи соответствующих дескрипторов.

Планы нумерации DigitMaps, определённые в дескрипторе DigitMapDescriptor, могут встречаться в любой из стандартных команд протокола, изменяющих окончания. План нумерации DigitMap, определённый однажды, может быть использован на всех окончаниях, определяемых в таких командах посредством идентификатора TerminationID (возможно, с использованием групповых символов замены). Планы нумерации DigitMaps, определённые на корневом окончании, являются глобальными и могут использоваться на каждом окончании в шлюзе MG, при условии, что план нумерации DigitMap с тем же самым именем не определяется на данном окончании. Когда DigitMap определяется динамически в дескрипторе DigitMap:

- Создаётся новый DigitMap путём указания имени, которое ещё не определено. Значение должно присутствовать.

- Значение DigitMap обновляется путём поставки нового значения для имени, которое уже определено. Окончания, ныне использующие digitmap, должны продолжать использовать старое определение; последующие аргументы EventsDescriptors, определяющие имя, включая любой аргумент EventsDescriptor в команде, содержащей дескриптор DigitMap, должны использовать новое определение.
- План нумерации DigitMap удаляется путём постановки пустого значения в уже определенное имя. Окончания, ныне использующие digitmap, должны продолжать использовать старое определение

7.1.14.2 Таймеры DigitMap

Совокупность цифр, соответствующих плану нумерации DigitMap может быть защищена тремя таймерами, а именно стартовым (T), коротким (S) и длинным (L).

- 1) Стартовый таймер (T) используется перед любыми набираемыми цифрами. Если стартовый таймер замещается значением, равным нулю ($T = 0$), то стартовый таймер должен быть отключён. Это подразумевает, что шлюз MG будет бесконечно ожидать цифр набора номера.
- 2) Если медиашлюз может определить, что для соответствия разрешённому эталону плана нумерации цифровой последовательности необходима, по крайней мере, ещё одна цифра, то значение межцифрового таймера должно быть установлено равным длиной (L) продолжительности (например, 16 секунд).
- 3) Если цифровая последовательность соответствует одному из эталонов плана нумерации, но возможно, что может быть принято ещё больше цифр, что приведёт к соответствию с другим эталоном, то вместо того, чтобы немедленно сообщать о соответствии, шлюз MG должен использовать короткий таймер (S) и ожидать прихода последующих цифр.

Таймеры являются реконфигурированными аргументами плана нумерации DigitMap. Значения по умолчанию этих таймеров должны быть изначально установлены в шлюзе MG, но могут замещаться значениями, определёнными внутри DigitMap.

7.1.14.3 DigitMap Синтаксис

Формальный синтаксис плана нумерации описывается правилом DigitMap в описании формального синтаксиса протокола (см. Приложения А и В). В соответствии с этим синтаксисом, план нумерации DigitMap определяется либо последовательностью, либо списком последовательностей. Каждая последовательность в списке является последовательностью альтернативных событий, определённой либо как последовательность, либо как регулярное выражение символов плана нумерации. Эти символы плана нумерации, цифры от "0" до "9" и буквы от "А" до максимального значения, зависящего от данной системы сигнализации, но никогда не превышающего "К", соответствуют определённым событиям в пределах комплекта, назначенного в дескрипторе событий на окончании, к которому применяется этот план нумерации. (Соответствие между событиями и символами плана нумерации определяется в документации на комплекты, относящиеся к системами сигнализации по каналам, таким как DTMF, MF, или R2. Цифры от "0" до "9" ДОЛЖНЫ соответствовать определённым цифровым событиям внутри данной системы сигнализации. Буквы должны размещаться в логическом порядке, облегчая использование диапазонной нотации для альтернативных событий.)

Буква "х" используется как групповой символ замены, обозначая любое событие, соответствующее символам в диапазоне "0"- "9". Последовательность может также содержать определённые диапазоны и, в более общем виде, определённые наборы символов, обозначающие альтернативные события, любое из которых удовлетворяет этой позиции плана нумерации. В заключение, точечный символ "." означает ноль или более повторений

селектора событий (событие, диапазон событий, набор альтернативных событий или групповой символ замены), которые ему предшествуют. Как следствие вышеупомянутого третьего правила работы таймера, меж-событийный отсчёт времени, соответствуя точечному символу окончания, использует по умолчанию короткий таймер.

В дополнение к этим символам событий, последовательность может содержать межсобытийные спецификаторы отсчёта времени "S" и "L", а также модификатор длительности "Z". Спецификаторы "S" and "L", соответственно, показывают, что шлюз MG должен использовать короткий (S) или длинный (L) таймеры для последующих событий, замещающих правила работы таймера, описанные выше. Если определённый спецификатор отсчёта времени в действительности находится в одной альтернативной последовательности событий, но ни одного не находится в любой другой возможной для применения альтернативе, должно использоваться значение таймера, установленное определённым спецификатором отсчёта времени. Если все последовательности с определённым управлением отсчётом времени удалены из возможного для применения набора, работа таймера возвращается к правилам по умолчанию, приведенным выше. В заключение, если конфликтующие спецификаторы отсчёта времени в действительности находятся в различных альтернативных последовательностях, должен использоваться длинный таймер.

Модификатор длительности "Z" обозначает событие большой длительности: помещённый перед символом(ами), обозначающим(ими) событие(я), удовлетворяющее(ие) данной позиции цифры, показывает, что эта позиция является удовлетворительной только, если длительность события превышает порог максимальной длительности. Предполагается, что значение этого порога указывается в шлюзе MG.

7.1.14.4 Завершающее событие DigitMap

План нумерации является активным до тех пор, пока вызвавший его дескриптор событий является активным и не завершился. План нумерации завершается, когда:

- таймер закончился; или
- альтернативная последовательность событий подобрана и другой альтернативной последовательности событий в плане нумерации не может быть подобрано путём обнаружения дополнительного события (точно выраженное соответствие); или
- обнаружено такое событие, что не будет возможно соответствие полной альтернативной последовательности событий плана нумерации, независимо от принятых дополнительных событий.

По завершении, должно быть сформировано завершающее событие плана нумерации, как это определяется в комплекте, обеспечивающем события для подбора плана нумерации. В этот момент план нумерации деактивируется. Последующие события в комплекте обрабатываются согласно механизмам обработки событий, активным в настоящее время.

7.1.14.5 Процедуры DigitMap

В ожидании завершения последующие события должны быть обработаны в соответствии со следующими правилами:

- 1) "Текущая последовательность набора номера", внутренняя переменная, вначале является пустой. Ряд возможных для применения альтернативных последовательностей событий включает все альтернативы, определённые в плане нумерации.
- 2) На каждом шаге таймер устанавливается в состояние ожидания следующего события на основе правил отсчёта времени по умолчанию, приведенных выше, или на определённом отсчёте времени, указанном в одном или более альтернативных

последовательностей событий. Если таймер истекает и представитель возможного для применения набора альтернатив полностью удовлетворён, сообщается о завершении лимита времени при полном соответствии. Если таймер истекает и часть альтернатив или ни одной возможной для применения альтернативы не оказывается удовлетворённой, сообщается о завершении лимита времени при частичном соответствии.

- 3) Если событие обнаруживается прежде, чем таймер истечёт, устанавливается его соответствие с каким-либо символом цифровой последовательности и оно временно присоединяется к окончанию текущей последовательности набора. Длительность события (длительное или не длительное) отмечается только, если это относится к текущей позиции символа (т.к. по крайней мере одна из возможных для применения последовательностей альтернативных событий содержит модификатор "Z" в последовательности на этой позиции).
- 4) Текущая последовательность набора номера сравнивается с возможными для применения последовательностями альтернативных событий. Если и только если последовательность, ожидающая событие большой продолжительности на этой позиции, совпадает (т.е. событие имело большую длительность и соответствовало спецификации для этой позиции), тогда любые альтернативные последовательности событий, не определяющие событие большой длительности на этой позиции, отклоняются, а текущая последовательность набора номера изменяется путём постановки "Z" перед символом, представляющим самое последнее событие. Любая последовательность, ожидающая событие большой длительности на этой позиции, но не совпадающая с наблюдаемым событием, вычёркивается из возможного набора. Если альтернативные последовательности событий, не определяющие события большой длительности в данной позиции, остаются в возможном для применения наборе после применения вышеупомянутых правил, наблюдаемая длительность события считается неподходящей для оценки совпадения с ними.
- 5) Если остаётся только одна возможность для применения и она полностью совпадает, создаётся завершающее событие, указывающее точное совпадение. Если возможностей для применения не остаётся, самое последнее событие удаляется из текущей последовательности набора номера и создаётся завершающее событие, указывающее полное совпадение, если один из кандидатов из предыдущего шага был полностью удовлетворён перед тем, как было обнаружено самое последнее событие, или частичное совпадение по-другому. О событии, удалённом из текущей последовательности набора номера будет затем сообщено, согласно текущим активным механизмам обработки событий.
- 6) Если по окончании шага 5) о завершающем событии не сообщается, обработка возвращается к шагу 2).

7.1.14.6 Активизация DigitMap

План нумерации активизируется всякий раз, когда новый дескриптор событий применяется к окончанию или активизируется вложенный дескриптор событий, причём дескриптор событий содержит завершающее событие плана нумерации. Завершающее событие плана нумерации содержит поле eventDM в поле действий по запросу. Каждая новая активизация плана нумерации начинается с шага 1) вышеупомянутой процедуры, с очищенной текущей номерной последовательностью. Любое предыдущее содержание текущей номерной последовательности от более ранней активизации теряется.

Завершающее событие плана нумерации, не содержащее поля eventDM в поле действий по запросу, считается ошибкой. После приёма такого события в аргументе EventsDescriptor, медиашлюз MG должен сформировать ответ ошибки, включая Ошибку 457 (Пропущенный аргумент в сигнале или событии).

7.1.14.7 Взаимодействие обработки дескрипторов DigitMap и Event

При активизированном плане нумерации для всех событий, определённых в комплекте, содержащем указанное завершающее событие плана нумерации, включается обнаружение. Обычное поведение события (например, прекращение сигналов до тех пор, пока включён флажок KeepActive завершающего события плана нумерации) продолжает применяться для каждого обнаруженного события за исключением:

- события в комплекте, содержащем указанное завершающее событие плана нумерации иное, чем само завершающее событие, индивидуально не уведомляются и не имеют побочных эффектов до тех пор, пока включаются по отдельности; и
- событие, вызывающее завершающее событие частичного совпадения, не распознаётся, а поэтому не имеет побочных эффектов до тех пор, пока снова не обработано вслед за распознаванием завершающего события плана набора номера.

7.1.14.8 Групповые символы замены

Заметим, что если комплект содержит завершающее событие плана нумерации, то событие спецификации, состоящее из имени комплекта с групповым символом замены ItemID (имя свойства), активизирует план нумерации; для этого событие спецификации должно включать поле eventDM в соответствии с 7.1.14.6. Если пакет также содержит сами цифровые события, то такая форма события спецификации вызовет сообщение в MGC при обнаружении индивидуальных событий.

7.1.14.9 Пример

В качестве примера рассмотрим следующий план набора номера:

0	Местный оператор
00	Междугородный оператор
Xxxx	Местный добавочный номер (начинается с 1-7)
8xxxxxxxx	Местный номер
#xxxxxxxx	Сторонний добавочный номер
*xx	Услуги по звёздочке
91xxxxxxxxxxx	Междугородный номер
9011 + до 15 цифр	Международный номер

Если для накопления набранных цифр используется комплект обнаружения DTMF, описанный в Е.6, то план набора номера, приведенный выше, приведёт к следующему плану нумерации:

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

7.1.15 Дескриптор Statistics

Дескриптор Statistics (статистическая информация) обеспечивает информацию, описывающую статус и использование окончания в течение его существования внутри определённого контекста. Для каждого окончания имеется набор сохраняемой, где это применимо, стандартной статистики (например, количество переданных и принятых октетов). Отдельные статистические свойства, о которых сообщается по данному окончанию, определяются комплектами, реализованными окончанием. По умолчанию статистика сообщается, когда окончание вычитается из контекста. Это поведение может быть замещено путём включения пустого аргумента AuditDescriptor в команду Subtract. Статистика может также возвращаться командой AuditValue, или любой другой командой Add/Move/Modify, использующей дескриптор Audit.

Статистика накапливается; сообщаемая статистика не сбрасывает её. Статистика сбрасывается, когда окончание вычитается из контекста.

7.1.16 **Дескриптор Packages**

Используется только с командой AuditValue; аргумент PackageDescriptor (дескриптор комплектов) возвращает список комплектов, реализованных окончанием.

7.1.17 **Дескриптор ObservedEvents**

Дескриптор ObservedEvents (наблюдаемые события) поставляется с командой Notify для информации контроллера MGC об обнаруженных событиях. Дескриптор ObservedEventsDescriptor, использующийся с командой AuditValue, возвращает события в буфер событий, который не был уведомлён. Дескриптор ObservedEvents содержит идентификатор RequestIdentifier дескриптора EventsDescriptor, который инициирует уведомление, обнаруженные события, факультативно моменты обнаружения, а также любые аргументы наблюдаемого события. О моментах обнаружения сообщается с точностью до сотых долей секунды.

7.1.18 **Дескриптор Topology**

Дескриптор Topology (топология) используется для определения направлений потоков между окончаниями в контексте. В противоположность дескрипторам предыдущих разделов, дескриптор Topology применяется в контексте вместо окончания. Топология контекста по умолчанию такова, что каждая передача окончания принимается всеми другими окончаниями. Дескриптор Topology является факультативным для реализации. Шлюз MG, который не поддерживает дескрипторы Topology, но принимает команду, содержащую дескриптор, возвращает сообщение об ошибке 444 (не поддерживаемый или не знакомый дескриптор), и факультативно включает последовательность, содержащую имя не поддерживаемого дескриптора ("Topology") в текст ошибки дескриптора ошибки.

Дескриптор Topology встречается перед командами в действии. Возможно действие, содержащее только дескриптор Topology при условии, что контекст, к которому это действие применяется, уже существует.

Дескриптор Topology состоит из последовательности треугольных форм ($T1$, $T2$, *взаимосвязь*). $T1$ и $T2$ определяют окончания в контексте, возможно с использованием групповых символов замены ALL или CHOOSE. *Взаимосвязь* определяет следующим образом, как среда передачи протекает между этими двумя окончаниями.

- ($T1$, $T2$, изоляция) означает, что окончания, соответствующие $T2$, не принимают среду передачи от окончаний, соответствующих $T1$, и наоборот.
- ($T1$, $T2$, односторонняя) означает, что окончания, соответствующие $T2$, принимают среду передачи от окончаний, соответствующих $T1$, но не наоборот. В этом случае не разрешается использование группового символа замены ALL так, как имеются окончания, соответствующие как $T1$, так и $T2$.
- ($T1$, $T2$, двусторонняя) означает, что окончания, соответствующие $T2$ принимают среду передачи от окончаний, соответствующих $T1$, и наоборот. В этом случае разрешается использование групповых символов замены так, как имеются окончания, соответствующие как $T1$, так и $T2$. Однако, если имеется окончание, которое соответствует обоим, петля обратной связи не воспроизводится.

Групповые символы замены CHOOSE также могут применяться в $T1$ и $T2$ со следующими ограничениями:

- действие (см. раздел 8), элементом которого является дескриптор топологии, содержит команду Add, в которой используется групповой символ замены CHOOSE;

- если групповой символ замены CHOOSE встречается в *T1* или *T2*, то неполное имя НЕ ДОЛЖНО определяться.

Групповой символ замены CHOOSE в дескрипторе Topology соответствует идентификатору TerminationID, который шлюз MG присваивает в первой команде Add, использующей групповой символ замены CHOOSE в том же самом действии. Существующее окончание, которое соответствует *T1* или *T2* в контексте, к которому присоединяется окончание, соединяется с вновь присоединённым окончанием, как определяется дескриптором Topology.

Рисунок 7 и следующая за ним таблица показывают несколько примеров результатов включения дескрипторов Topology в действия. В этих примерах предполагается, что дескрипторы Topology применяются последовательно.

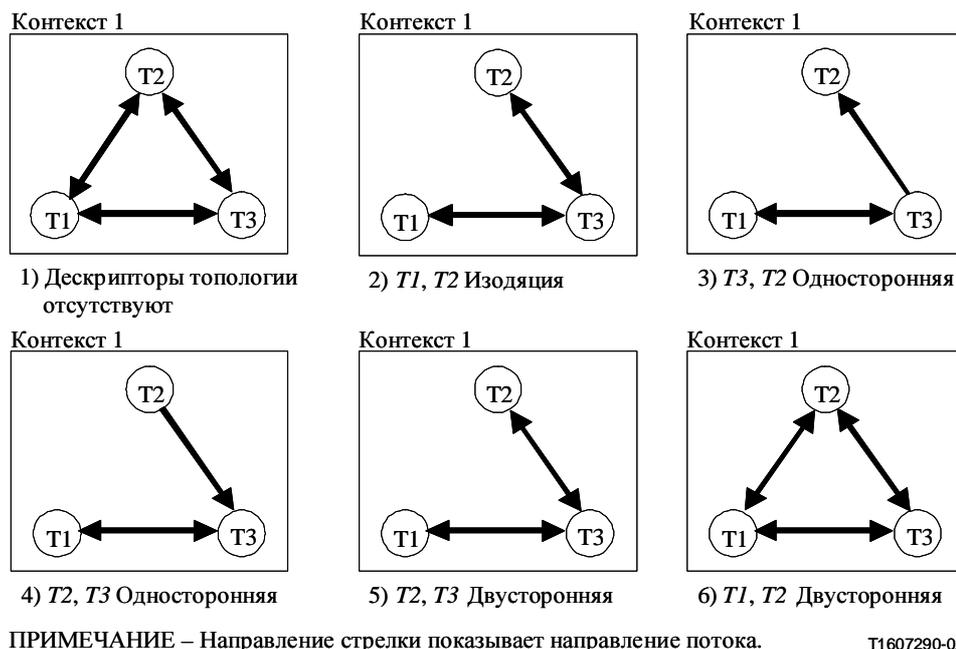


Рисунок 7/Н.248.1 – Пример топологий

Топология	Описание
1	Дескрипторы топологии отсутствуют При отсутствии включённых дескрипторов топологии все окончания имеют двусторонние соединения со всеми другими окончаниями.
2	<i>T1, T2</i> изоляция Разрывает соединение между <i>T1</i> и <i>T2</i> . <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 двусторонние соединения с <i>T3</i> .
3	<i>T3, T2</i> односторонняя Одностороннее соединение от <i>T3</i> к <i>T2</i> (т.е. <i>T2</i> принимает поток среды передачи от <i>T3</i>). Двустороннее соединение между <i>T1</i> и <i>T3</i> .
4	<i>T2, T3</i> односторонняя Одностороннее соединение от <i>T2</i> к <i>T3</i> . <i>T1</i> и <i>T3</i> остаются двусторонне соединёнными
5	<i>T2, T3</i> двусторонняя <i>T2</i> двусторонне соединено с <i>T3</i> . Это приводит к тому же, что и в 2.
6	<i>T1, T2</i> двусторонняя (<i>T2, T3</i> двусторонняя и <i>T1, T3</i> двусторонняя может быть

Топология	Описание
	неявной или явной). Все окончания имеют двусторонние соединения со всеми другими окончаниями.

Одностороннее соединение должно быть реализовано таким образом, чтобы другие окончания в контексте не знали об изменении топологии.

7.1.19 Дескриптор Error

Если в процессе обработки запроса на транзакцию отвечающая сторона обнаруживает ошибку, она должна включить в свой ответ дескриптор ошибки. Запрос уведомления также может содержать дескриптор ошибки.

Дескриптор ошибки состоит из кода ошибки, зарегистрированного IANA, факультативно сопровождаемого пояснением ошибки. Рекомендация МСЭ-Т Н.248.8 содержит список действующих кодов ошибок и описаний ошибок.

Дескриптор ошибки должен быть описан на "самом подробном уровне", который семантически соответствует описываемой ошибке и которому, возможно, оригинальным запросом создаются какие-либо проблемы синтаксического анализа. Дескриптор ошибки может делать ссылку на синтаксическую конструкцию иную, чем та, где она появилась. Например, дескриптор ошибки 422 (синтаксическая ошибка в действии), может появиться внутри команды, хотя она относится к большей конструкции – действию, а не к отдельной команде, внутри которой она появляется.

7.2 Стык прикладного программирования команд

Следующим является стык прикладного программирования (API), описывающий команды протокола. Стык API показан для иллюстрации команд и их аргументов и не предназначается для определения реализации (например, путём использования блокирующих функциональных соединений). Он описывает в скобках входные аргументы после имени команды и возвращаемые значения перед командой. Это делается только с целью описания; реальный синтаксис команды и кодирование определяются в последующих разделах. Порядок аргументов команд не фиксируется. Дескрипторы могут появляться как аргументы команд в любом порядке. Дескрипторы SHALL обрабатываются порядке их появления. Любой ответ на команду может содержать дескриптор ошибки; стык API специально не показывает этого.

Все аргументы, поставленные в квадратные скобки ([. . .]), считаются факультативными.

7.2.1 Команда Add

Команда Add (присоединять) присоединяет окончание к контексту.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

)

Идентификатор TerminationID определяет окончание, которое присоединяется к контексту. Окончание либо создаётся, либо берётся из нулевого контекста. Если групповой символ замены CHOOSE используется в идентификаторе TerminationID, будет возвращён выбранный идентификатор. В команде Add могут быть использованы групповые символы замены, однако такое использование было бы необычным. Если групповой символ замены соответствует более чем одному идентификатору TerminationID, проверяются все возможные соответствия, причём о результатах каждой проверки создаётся отчёт. Порядок проверок при соответствии нескольких идентификаторов TerminationID не определяется.

Факультативный дескриптор MediaDescriptor описывает все медиапотоки.

Факультативные дескрипторы ModemDescriptor и MuxDescriptor определяют, где применимо, модем и мультиплексор. В целях удобства, если дескриптор Multiplex присутствует в команде Add и перечисляет любые окончания, которые в настоящее время не находятся в контексте, то такие окончания присоединяются к контексту, как будто были активизированы отдельные команды Add, перечисляющие окончания. Если при исполнении такой неявной команды Add происходит ошибка, должна быть возвращена ошибка 471 (неявная команда Add вследствие сбоя дескриптора Multiplex) и дальнейшая обработка команды прекратится.

Аргумент EventsDescriptor является факультативным. При наличии, он обеспечивает список событий, которые должны быть обнаружены на окончании.

Аргумент EventBufferDescriptor является факультативным. При наличии, он обеспечивает список событий, которые запрашиваются шлюзом MG для обнаружения и буферизации, когда аргумент EventBufferControl равен LockStep.

Аргумент SignalsDescriptor является факультативным. При наличии, он обеспечивает список сигналов, которые должны быть применимы к окончанию.

Аргумент DigitMapDescriptor является факультативным. При наличии, он обеспечивает определение DigitMap, которое может использоваться в аргументе EventsDescriptor.

Аргумент AuditDescriptor является факультативным. При наличии, команда возвратит дескрипторы, как это определяется в аргументе AuditDescriptor.

Все дескрипторы, которые могут изменяться, могут возвращаться шлюзом MG, если аргумент был недоопределён или переопределён. Дескрипторы ObservedEvents, Statistics, Packages, а также EventBuffer возвращаются только, если запрашиваются в аргументе AuditDescriptor.

Команда Add НЕ ДОЛЖНА использоваться на окончании с serviceState в положении "OutOfService".

7.2.2 Команда Modify

Команда Modify (изменять) изменяет свойства окончания.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Modify(TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

)

Идентификатор TerminationID может быть положительно-выраженным, если должно изменяться единственное окончание в контексте. Для некоторых операций может быть характерно применение групповых символов замены в идентификаторе TerminationID. Если групповой символ замены соответствует более чем одному идентификатору TerminationID, проверяются все возможные соответствия, причём о результатах каждой проверки создаётся отчёт. Порядок проверок при соответствии нескольких идентификаторов TerminationID не определяется.

Возможность использования группового символа замены CHOOSE является ошибкой, т.к. команда Modify может применяться только на существующих окончаниях.

В целях удобства, если дескриптор Multiplex присутствует в команде Modify, то:

- если новый дескриптор Multiplex перечисляет любые окончания, которые в настоящее время не находятся в контексте, такие окончания присоединяются к контексту, как будто были активизированы отдельные команды Add, перечисляющие окончания.

- если какие-нибудь окончания, перечисленные прежде в дескрипторе Multiplex, больше не присутствуют в новом дескрипторе Multiplex, то они вычитаются из контекста, как будто были активизированы отдельные команды Subtract, перечисляющие окончания.

Остальные аргументы команды Modify являются теми же самыми, что и у команды Add. Возможные возвратные значения являются теми же самыми, что и у команды Add.

7.2.3 Команда Subtract

Команда Subtract (вычитать) разъединяет окончание с его контекстом и возвращает статистику по участию окончания в контексте.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
Subtract(TerminationID
        [, AuditDescriptor]
)
```

Идентификатор TerminationID во входных аргументах представляет окончание, которое вычитается. Идентификатор TerminationID может быть положительно-выраженным или может быть значением группового символа замены, показывающим, что все (или ряд имеющих отношение) окончания в контексте команды Subtract должны вычитаться. Если групповой символ замены соответствует более, чем одному идентификатору TerminationID, проверяются все возможные соответствия, причём о результатах каждой проверки создаётся отчёт. Порядок проверок при соответствии нескольких идентификаторов TerminationIDs не определяется.

Использование группового символа замены CHOOSE в идентификаторе TerminationID является ошибкой, т.к команда Subtract может использоваться только на существующих окончаниях.

Групповой символ замены ALL может использоваться как в идентификаторе ContextID, так и в идентификаторе TerminationId команды Subtract, что приведёт к удалению всех контекстов, удалению всех эфемерных окончаний и к возврату всех физических окончаний в нулевой контекст. Вычитание окончания из нулевого контекста не разрешается.

В целях удобства, если мультиплексирующее окончание является объектом команды Subtract, то любые окончания транспортных каналов, перечисленные в своём дескрипторе Multiplex вычитаются их контекста как будто были активизированы отдельные команды Subtract, перечисляющие окончания.

По умолчанию аргумент Statistics возвращается для сообщения информации, собранной на окончании или окончаниях, указанных в команде. Сообщённая информация применяется к существованию окончания или окончаний в контексте из которого оно было или они были вычтены.

Аргумент AuditDescriptor является факультативным. При наличии, команда возвратит только те дескрипторы, как это определяется в аргументе AuditDescriptor, которые могут быть пустыми. При отсутствии, дескриптор Statistics возвращается по умолчанию. Возможные возвращаемые значения являются теми же самыми, что и у команды Add.

Когда обеспеченное окончание вычитается из контекста, его значения свойства должны возвратиться к:

- значению по умолчанию, если оно указывается для свойства и не замещено инициализацией;
- в противном случае, обеспеченному значению.

7.2.4 Команда Move

Команда Move (переслать) пересылает окончание из существующего контекста в другой контекст за одну элементарную операцию. Команда Move является единственной командой, которая относится к окончанию в контексте, отличном от того, к которому команда применяется. Команда Move не должна применяться для пересылки окончаний в или из нулевого контекста.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Move(TerminationID

 [, MediaDescriptor]

 [, ModemDescriptor]

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

)

Идентификатор TerminationID определяет пересылаемое окончание. В нём могут использоваться групповые символы замены, однако групповой символ замены CHOOSE не должен использоваться в идентификаторе TerminationID. Если групповой символ замены соответствует более, чем одному идентификатору TerminationID, проверяются все возможные соответствия, причём о результатах каждой проверки создаётся отчёт. Порядок проверок при соответствии нескольких идентификаторов TerminationIDs не определяется. Контекст, в который пересылается окончание, определяется целевым идентификатором в действии. Контекст удаляется, если последнее оставшееся окончание пересылается за пределы контекста.

Команда Move не влияет на свойства окончания, на котором она работает, за исключением свойств, явно изменённых дескрипторами, включёнными в команду Move. Например, аргумент AuditDescriptor с факультативным дескриптором Statistics возвратит статистику на окончании как раз перед командой Move. Возможными дескрипторами, возвращёнными командой Move, являются те же самые, что и у команды Add.

В целях удобства, если мультиплексированное окончание является объектом команды Move, то любые окончания транспортных каналов, перечисленные в её дескрипторе Multiplex тоже пересылаются как будто были активизированы отдельные команды Move, перечисляющие окончания.

Команда Move НЕ ДОЛЖНА использоваться на окончании с serviceState в положении "OutOfService".

7.2.5 Команда AuditValue

Команда AuditValue (проверить значение) возвращает текущие значения свойств, событий, сигналов и статистики, связанных с окончаниями.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

AuditValue(TerminationID,

AuditDescriptor

)

Идентификатор TerminationID может быть необычным или с групповым символом замены. Если групповой символ замены соответствует более, чем одному идентификатору TerminationID, проверяются все возможные соответствия, причём о результатах каждой проверки создаётся отчёт. Порядок проверок при соответствии нескольких идентификаторов TerminationIDs не определяется. Если запрашивается ответ с групповым символом замены, то формируется возврат только одной команды, содержащий объединение значений всех

окончаний, соответствующих групповому символу замены. Это соглашение может сократить объём необходимых данных для проверки группы окончаний. Использование группового символа замены CHOOSE является ошибкой.

Походящие дескрипторы с текущими значениями для окончания возвращаются командой AuditValue. Значения, появляющиеся в нескольких примерах дескриптора, определяются как поддерживаемые альтернативные значения с каждым из аргументов дескриптора, считающимся независимым.

Дескриптор ObservedEvents возвращает список событий дескриптору EventBuffer. Если аргумент ObservedEventsDescriptor проверяется, пока план нумерации DigitMap является активным, возвращённый дескриптор ObservedEvents также включает завершающее событие плана нумерации, который показывает текущую последовательность набора номера, но не показывает способ окончания.

Дескриптор EventBuffer возвращает набор событий и связанных значений аргументов, включённых в настоящее время в EventBufferDescriptor. Дескриптор PackagesDescriptor возвращает список комплектов, реализованных окончанием. Дескриптор DigitMapDescriptor возвращает название или значение текущего плана нумерации DigitMap для окончания. План нумерации DigitMap, запрошенный в команде AuditValue с идентификатором TerminationID, равным ALL, возвращает все планы нумерации DigitMaps в шлюзе. Дескриптор Statistics возвращает текущие значения всей статистики, сохраняемой на окончании. Определяя пустой дескриптор Audit приводит к тому, что возвращается только идентификатор TerminationID. Это может быть полезным для получения списка идентификаторов TerminationIDs при использовании группового символа замены. В Приложениях А и В приводится специальный синтаксис для представления такого списка в сжатой форме, так что тэг команды AuditValue не должен повторяться для каждого идентификатора TerminationID.

Результаты команды AuditValue зависят от контекста, а именно положительно-выраженный, нулевой или с использованием группового символа замены. (Заметим, что идентификатор ContextID, равный All, не включает нулевой контекст). Идентификатор TerminationID может быть положительно-выраженным или с использованием группового символа замены.

Ниже приводятся примеры того, что возвращается в случае, если в контексте и/или окончании используется групповой символ замены и указывается ответ группового символа замены.

Предположим, что шлюз имеет 4 окончания: t1/1, t1/2, t2/1 и t2/2. Допустим, что окончания t1/* имеют осуществлённые комплекты aaa и bbb, а окончания t2/* - осуществлённые комплекты ccc и ddd. Предположим, что в контексте 1 имеются t1/1 и t2/1, а в контексте 2 - t1/2 и t2/2.

Команда:

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

возвращает:

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

Команда:

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

возвращает:

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

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

Команда:

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

возвращает:

```
Context={W-AuditValue=t1/{Packages{aaa,bbb}}}.
```

ПРИМЕЧАНИЕ – Ответ с использованием группового символа замены также может применяться для других команд, таких как команда Subtract.

В нижеследующей таблице приводится дополнительная информация, которая может быть получена с командой AuditValue:

ContextID	TerminationID	Information Obtained
Specific	Wildcard	Проверка соответствующих окончаний в контексте
Specific	Specific	Проверка единственного окончания в контексте
Null	Root	Проверка состояния медиашлюза и событий
Null	Wildcard	Проверка всех соответствующих окончаний в нулевом контексте
Null	Specific	Проверка единственного окончания за пределами любого контекста
All	Wildcard	Проверка всех соответствующих окончаний и контекста, с которым они связаны
All	Root	Список всех ContextIds (список ContextID должен возвращаться путём использования нескольких командных ответов, каждый из которых содержит ContextID из списка)
All	Specific	(Не нулевой) ContextID, в котором в настоящее время существует окончание

7.2.6 Команда AuditCapabilities

Команда AuditCapabilities (проверить возможности) возвращает возможные значения свойств, событий, сигналов и статистики, связанных с окончаниями.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor]

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

AuditCapabilities(TerminationID,
AuditDescriptor)

Команда AuditCapabilities возвращает соответствующие дескрипторы с возможными значениями для окончаний. Дескрипторы могут повторяться при нескольких возможных значениях. Если запрашивается ответ с групповым символом замены, то формируется только

один возврат команды, содержащий объединение значений всех окончаний, соответствующих групповому символу замены. Это соглашение может сократить объём необходимых данных для проверки группы окончаний.

Интерпретация того, какие возможности требуются для разных значений идентификаторов ContextID и TerminationID, является той же самой, как и в команде AuditValue.

Команда EventsDescriptor возвращает список возможных событий на окончании вместе со списком всех возможных значений аргументов дескриптора EventsDescriptor. Дескриптор EventBufferDescriptor возвращает ту же информацию, что и EventsDescriptor. Дескриптор SignalsDescriptor возвращает список возможных сигналов, которые могут применяться к окончанию вместе со списком всех возможных значений аргументов дескриптора Signals. Дескриптор StatisticsDescriptor возвращает имена статистики, сохраняемой на окончании. Дескрипторы DigitMap и Packages не допустимы для применения в команде AuditCapability.

В нижеследующей таблице приводится дополнительная информация, которая может быть получена с командой AuditValue:

ContextID	TerminationID	Information Obtained
Specific	Wildcard	Проверка соответствующих окончаний в контексте
Specific	Specific	Проверка единственного окончания в контексте
Null	Root	Проверка состояния шлюза MG и событий
Null	Wildcard	Проверка всех соответствующих окончаний в нулевом контексте
Null	Specific	Проверка единственного окончания за пределами любого контекста
All	Wildcard	Проверка всех соответствующих окончаний и контекста, с которым они связаны
All	Root	То же самое, что и для AuditValue
All	Specific	То же самое, что и для AuditValue

7.2.7 Команда Notify

Команда Notify (уведомлять) позволяет медиашлюзу уведомить контроллер медиашлюза о событиях, происходящих в пределах медиашлюза.

TerminationID

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

Аргумент TerminationID определяет окончание, вырабатывающее команду Notify. Идентификатор TerminationID должен быть полностью квалифицированным именем.

Дескриптор ObservedEventsDescriptor содержит идентификатор RequestID и список событий, обнаруженных медиашлюзом для того, чтобы они были обнаружены. Каждое событие в списке сопровождается аргументами, связанными с событием, и факультативно, указанием времени обнаружения события. Процедуры отправки команд Notify с идентификатором RequestID, равным 0, являются предметом дальнейшего изучения.

Команды Notify с идентификатором RequestID, не равным 0, должны встречаться только в результате обнаружения события, определённого дескриптором Events, являющегося активным на данном окончании.

Идентификатор RequestID возвращает аргумент RequestID дескриптора EventsDescriptor, вызвавшего команду Notify. Он используется для установления соотношения с вызвавшим его запросом. События в списке должны быть затребованы посредством вызывающего дескриптора EventsDescriptor или дескриптора вложенных событий пока идентификатор RequestID равен is 0 (что является предметом дальнейшего изучения).

Дескриптор ErrorDescriptor может передаваться в команде Notify как результат ошибки 518 (буфер события заполнен).

7.2.8 Команда ServiceChange

Команда ServiceChange позволяет медиашлюзу уведомить контроллер медиашлюза, что окончание или группа окончаний собираются быть установленными в нерабочее состояние или только что были установлены в рабочее состояние. Контроллер медиашлюза может показывать, что окончание(я) должны быть удалены из или возвращены в службу. Медиашлюз может уведомить контроллер MGC об изменении возможности окончания. Он также позволяет контроллеру MGC передать управление медиашлюзом другому контроллеру MGC.

TerminationID,

[ServiceChangeDescriptor]

```
ServiceChange(TerminationID,  
ServiceChangeDescriptor  
)
```

Аргумент TerminationID определяет окончание(я), которые удаляются или возвращаются в службу. Разрешается использование групповых символов замены в именах окончаний за исключением механизма CHOOSE, который применяться не должен. Использование идентификатора TerminationID типа "Root" показывает команду ServiceChange, затрагивающую целиком медиашлюз.

Дескриптор ServiceChangeDescriptor содержит по требованию следующие аргументы:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeDelay;
- ServiceChangeAddress;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMgcId;
- TimeStamp.

Аргумент ServiceChangeMethod определяет тип команды ServiceChange, которая произойдет или произошла:

- 1) Graceful – показывает, что указанные окончания будут установлены в нерабочее состояние после указанной ServiceChangeDelay; установленные соединения пока ещё не затрагиваются, но контроллер медиашлюза должен воздержаться от установления новых соединений и должен попытаться постепенно разорвать существующие соединения на окончании, затрагиваемом командой serviceChange. По окончании задержки ServiceChangeDelay или удалении окончания из активного контекста (что бы ни случилось первым) шлюз MG должен установить метод serviceState окончания в положение "нерабочее".
- 2) Forced – показывает, что указанные окончания были внезапно установлены в нерабочее состояние и любые, связанные с ними соединения, могут быть потеряны. Для окончаний типа non-Root, контроллер MGC отвечает за очистку контекста (если

это имеет место), с которым связано неудавшееся соединение. Как минимум окончание должно быть вычтено из контекста. Метод `serviceState` окончания должно быть в положении "нерабочее". В случае корневого окончания контроллер MGC может предположить, что все окончания в шлюзе MG потеряны, а поэтому можно полагать, что все окончания были вычтены.

- 3) `Restart` – показывает, что после истечения задержки `ServiceChangeDelay` на определённых окончаниях, служба будет восстановлена. По истечении `ServiceChangeDelay` метод `serviceState` должен быть установлен в положение "рабочее".
- 4) `Disconnected` – всегда применяется с идентификатором `Root TerminationID`, чтобы показать, что шлюз MG потерял связь с контроллером MGC, но затем связь с тем же самым контроллером MGC восстанавливается (вероятно после попыток с другими контроллерами, находящимися в предварительно представленном списке). Т.к. состояние шлюза MG может измениться, контроллер MGC может пожелать использовать команду `Audit`, чтобы вновь синхронизировать своё состояние с состоянием шлюза MG.
- 5) `Handoff` – передаётся контроллером MGC в шлюз MG чтобы показать, что контроллер MGC собирается перейти в нерабочее состояние и должна быть установлена новая взаимосвязь контроллера MGC. Передаётся шлюзом MG в контроллер MGC чтобы показать, что шлюз MG пытается установить новую взаимосвязь в соответствии с сигналом плавного переключения, принятым от контроллера MGC, с которым он был ранее взаимосвязан.
- 6) `Failover` – передаётся шлюзом MG в контроллер MGC чтобы показать, что первичный шлюз MG находится в нерабочем состоянии, а вторичный шлюз MG принимает работу на себя. Этот метод `serviceChange` также передаётся шлюзом MG в контроллер MGC, когда шлюз MG обнаруживает, что контроллер MGC вышел из строя.
- 7) Другое значение, чей смысл является взаимно понятным как шлюзу MG, так и контроллеру MGC.

Аргумент `ServiceChangeReason` определяет причину, почему произошло или произойдёт `ServiceChange`. Он содержит буквенно-цифровую лексему (зарегистрированную IANA) и, факультативно, пояснительную последовательность.

Факультативный аргумент `ServiceChangeAddress` определяет адрес (например, номер IP порта для IP сетей), который будет использоваться в последующих коммуникациях. Он может определяться дескриптором входного аргумента или дескриптором возвращённого результата. Аргументы `ServiceChangeAddress` и `ServiceChangeMgcId` не должны присутствовать вместе в дескрипторе `ServiceChangeDescriptor` или в дескрипторе `ServiceChangeResultDescriptor`. Аргумент `ServiceChangeAddress` обеспечивает адрес, который будет использоваться внутри контекста взаимосвязи, согласуемой в настоящее время, в то время, как аргумент `ServiceChangeMgcId` обеспечивает альтернативный адрес, по которому шлюз MG должен стремиться установить другую взаимосвязь. Заметим, что использование аргумента `ServiceChangeAddress` не поощряется. Контроллеры MGC и шлюзы MG должны быть способны работать с `ServiceChangeAddress`, который является либо полным адресом, либо только номером порта в случае TCP перемещений.

Факультативный аргумент `ServiceChangeDelay` измеряется в секундах. Если задержка отсутствует или равняется нулю, то значение задержки должно приниматься равной нулю. В случае аргумента `ServiceChangeMethod`, соответствующего "graceful", нулевая задержка показывает, что контроллер медиашлюза должен ожидать естественного удаления существующих соединений и не должен устанавливать новые соединения. Только для случая "graceful", нулевая задержка означает, что шлюз MG не должен устанавливать метод

serviceState в положение "out of service" до тех пор, пока окончание находится в нулевом контексте.

Факультативный аргумент ServiceChangeProfile определяет совокупность параметров (если это имеет место) поддерживаемого протокола. Аргумент ServiceChangeProfile включает версию поддерживаемой совокупности параметров.

Факультативный аргумент ServiceChangeVersion содержит версию протокола и используется, если происходит согласование версии протокола (см. 11.3).

Факультативный аргумент TimeStamp определяет действительное время, как оно отсчитывается отправителем. Как таковое, абсолютное время не является необходимым в соответствии с, например, местной временной зоной – оно просто устанавливает случайное начальное время, относительно которого должны сравниваться все будущие отметки времени, передаваемые отправителем в течение этой взаимосвязи. Оно может использоваться получателем для определения как его понятие времени отличается от понятия времени его корреспондента. Аргумент TimeStamp передаётся с точностью сотых долей секунды.

Факультативный аргумент Extension может содержать любое значение, чей смысл обоюдно понимается шлюзом MG и контроллером MGC.

Команда ServiceChange, определяющая "Root" для идентификатора TerminationID и метод ServiceChangeMethod, равный Restart, является регистрационной командой, посредством которой медиашлюз объявляет контроллеру медиашлюза о своём существовании. Медиашлюз может также объявить регистрационную команду путём определения "Root" для идентификатора TerminationID и метода ServiceChangeMethod, равного Failover, когда шлюз MG обнаруживает сбой контроллера MGC. Предполагается, что медиашлюзу будет дано имя одного первичного и, факультативно, нескольких альтернативных контроллеров медиашлюза. Подтверждение команды ServiceChange завершает процесс регистрации за исключением случая, когда контроллер MGC возвратил альтернативный аргумент ServiceChangeMgcId, как описывается в следующем параграфе. Шлюз MG может определить транспортный аргумент ServiceChangeAddress, который будет использоваться контроллером MGC для передачи сообщений в аргумент ServiceChangeAddress входящего дескриптора ServiceChangeDescriptor. Шлюз MG может определить адрес в аргументе ServiceChangeAddress запроса ServiceChange, а контроллер MGC также может сделать это в аргументе ответа ServiceChange. В любом случае получатель должен использовать поставленный адрес в качестве адреса доставки для всех последующих запросов на транзакции в пределах взаимосвязи. В то же время, как отмечено в разделе 9, ответы на транзакции и указания в состоянии ожидания должны посылаться по адресу источника соответствующих запросов. Это должно быть сделано даже, если это подразумевает дополнительный обмен сообщениями, т.к. команды и ответы не могут укомплектовываться вместе. Аргумент TimeStamp должен передаваться с регистрационной командой и её ответом.

Контроллер медиашлюза может вернуть аргумент ServiceChangeMgcId, описывающий контроллер медиашлюза, с которым предпочтительно должна быть установлена связь для дальнейшего обслуживания медиашлюзом. В этом случае медиашлюз должен повторно послать команду ServiceChange новому контроллеру медиашлюза. С контроллером MGC, определённым в аргументе ServiceChangeMgcId, если последний предоставляется, связь должна устанавливаться раньше, чем с любыми другими альтернативными контроллерами MGC. В случае сообщения HandOff от контроллера MGC к шлюзу MG, аргумент ServiceChangeMgcId является новым контроллером MGC, который принимает на себя управление от текущего контроллера.

Возврат команды ServiceChange является пустым за исключением случая, когда используется идентификатор terminationID типа Root. В этом случае она включает по требованию следующие аргументы:

- ServiceChangeAddress, если отвечающий контроллер MGC желает определить новое место назначения для сообщений шлюза MG, оставшихся во взаимосвязи;
- ServiceChangeMgcId, если отвечающий контроллер MGC не желает поддерживать взаимосвязь с шлюзом MG;
- ServiceChangeProfile, если отвечающая сторона желает согласовать совокупность параметров, которая будет использоваться для взаимосвязи;
- ServiceChangeVersion, если отвечающая сторона желает согласовать версию протокола, который будет использоваться для взаимосвязи.

Определяются следующие аргументы ServiceChangeReasons. Этот список может быть продолжен регистрацией IANA, как отмечается в 13.3.

```
900 Service Restored
901 Cold Boot
902 Warm Boot
903 MGC Directed Change
904 Termination malfunctioning
905 Termination taken out of service
906 Loss of lower layer connectivity (e.g. downstream sync)
907 Transmission Failure
908 MG Impending Failure
909 MGC Impending Failure
910 Media Capability Failure
911 Modem Capability Failure
912 Mux Capability Failure
913 Signal Capability Failure
914 Event Capability Failure
915 State Loss
```

7.2.9 Изменение и проверка качественных признаков контекста

Как обсуждалось в предыдущих разделах команды протокола применяются с окончаниями. Данный раздел определяет как изменяются и проверяются контексты.

Команды группируются в действия (см. раздел 8). Действие применяется к одному контексту. В дополнение к командам действие может содержать инструкции по изменению и проверке контекста.

Запрос на действие, посланный в шлюз MG, может включать запрос на проверку качественных признаков контекста. Действие может также включать запрос на изменение качественных признаков контекста.

Свойства контекста, которые могут быть включены в ответ на действие, используются для возврата информации контроллеру MGC. Это может быть информация, запрошенная проверкой качественных признаков контекста или элементов результата изменения контекста.

Если шлюз MG принимает действие, которое содержит как запрос на проверку качественных признаков, так и запрос на изменение этих атрибутов, ответ ДОЛЖЕН включать значения качественных признаков после обработки запроса на изменение.

7.2.10 Общий синтаксис команд

Протокол может быть закодирован в двоичном или текстовом формате. Контроллеры MGC должны поддерживать оба формата кодирования.

Синтаксис протокола для двоичного формата протокола описан в Приложении А. В Приложении С описывается кодирование дескрипторов Local и Remote при использовании бинарного формата.

В Приложении В приводится полный ABNF текстового кодирования протокола в соответствии с IETF RFC 2234. Для кодирования дескрипторов Local и Remote при использовании текстового кодирования с изменениями, приведенными в 7.1.8, применяется SDP.

7.3 Коды ошибок команд

Ошибки состоят из кода ошибки, зарегистрированного IANA, и поясняющей последовательности. Передача поясняющей последовательности является факультативной. Реализации призываются добавить диагностическую информацию в конце последовательности.

Когда шлюз MG сообщает контроллеру MGC об ошибке, он это делает в дескрипторе ошибки. Дескриптор ошибки состоит из кода ошибки и соответствующей пояснительной последовательности, являющейся факультативной.

Рекомендация МСЭ-Т Н.248.8 содержит коды ошибок, которые поддерживаются Рекомендациями серии Н.248.х.

8 Транзакции

Команды, используемые контроллером медиашлюза и медиашлюзом, группируются в транзакции, каждая из которых определяется идентификатором TransactionID. Транзакции состоят из одного или более действий. Действие состоит из непустой последовательности команд, изменений свойств контекста или проверок свойств контекста, функционирующих в пределах единственного контекста. Соответственно, каждое действие обычно определяет идентификатор ContextID. Однако, имеются два случая, когда характерный идентификатор ContextID не обеспечивается действием. Один из них является случаем изменения окончания вне контекста. Другой, когда контроллер запрашивает шлюз создать новый контекст. На Рисунке 8 графически представлена взаимосвязь транзакции, действия и команды.

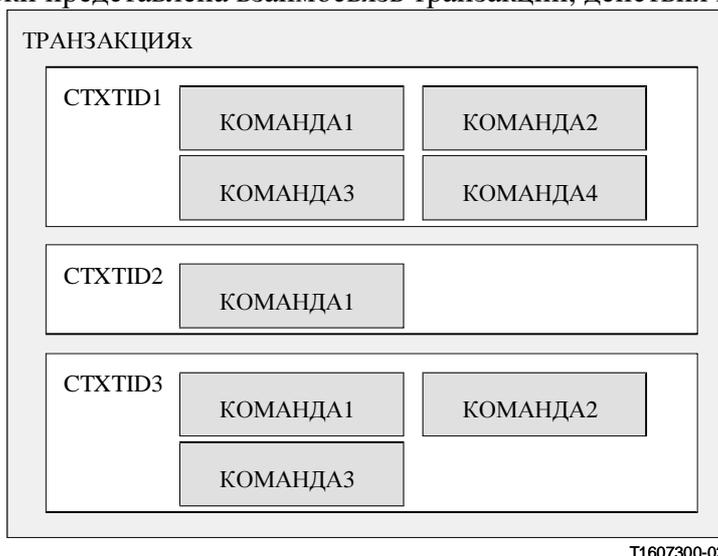


Рисунок 8/Н.248.1 – Транзакции, действия и команды

Транзакции представляются в виде запросов TransactionRequests. Соответствующие отклики на запрос TransactionRequest принимаются в единственном ответе, которому возможно предшествует ряд сообщений TransactionPending (см. 8.2.3).

Транзакции обеспечивают упорядоченную обработку команд. Таким образом, команды в пределах транзакции исполняются последовательно. Упорядочение транзакций не является гарантированным – транзакции могут исполняться в любом порядке или одновременно.

При первом сбое команды в транзакции обработка остающихся команд в этой транзакции останавливается. Если команда содержит идентификатор с использованием группового символа замены, команда проверяется с каждым из идентификаторов TerminationIDs, соответствующих групповому символу замены. Для каждого соответствующего идентификатора TerminationID в ответ TransactionReply включается отклик даже, если один или более запросов вызвали ошибку. Если какой-либо идентификатор TerminationID, соответствующий групповому символу замены, приводит в процессе исполнения к ошибке, любые команды, которые следуют за командой с групповым символом замены, не проверяются.

Команды могут быть помечены как "факультативные", что может заместить такое поведение (если команда, помеченная как факультативная, приводит к ошибке, то последующие команды в транзакции будут исполняться). Если команда сбоят, шлюз MG должен, насколько возможно, восстановить состояние, которое было до проверочного исполнения команды прежде, чем продолжать обработку команды.

Ответ TransactionReply включает результаты для всех команд в соответствующем запросе TransactionRequest. Ответ TransactionReply содержит возвратные значения для успешно исполненных команд, а также команду и дескриптор ошибки для любой неудачной команды. Сообщение TransactionPending используется для периодического уведомления получателя о том, что транзакция ещё не закончена, но энергично обрабатывается.

Приложения ДОЛЖНЫ реализовывать таймер прикладного уровня на транзакцию. Истечение таймера должно вызывать повторную передачу запроса. Приём ответа должен отменять таймер. Приём ожидания должен перезапустить таймер.

8.1 Общие аргументы

8.1.1 Идентификаторы транзакций

Транзакции определяются идентификатором TransactionID, который присваивается отправителем и является уникальными в рамках отправителя. Отклик, содержащий дескриптор ошибки для указания об отсутствии в запросе идентификатора TransactionID, должен использовать идентификатор TransactionID 0 в соответствующем ответе TransactionReply.

8.1.2 Идентификаторы контекстов

Контексты определяются идентификатором ContextID, который присваивается медиашлюзом и является уникальными в рамках медиашлюза. Контроллер медиашлюза должен использовать идентификатор ContextID, присвоенный медиашлюзом, во всех последующих транзакциях, относящихся к данному контексту. Протокол ссылается на отмеченное значение, которое может использоваться контроллером медиашлюза для ссылки на окончание, в настоящее время не связанное с контекстом, а именно на *нулевой* идентификатор ContextID.

Групповой символ замены CHOOSE применяется для запроса на создание медиашлюзом нового контекста.

Контроллер медиашлюза MGC может использовать групповой символ замены ALL для адресации всех контекстов медиашлюза. При использовании группового символа замены ALL нулевой контекст не включается.

Контроллер MGC не должен использовать частично определённые идентификаторы ContextID, содержащие групповые символы замены CHOOSE или ALL.

8.2 Стык прикладного программирования транзакций

Нижеследующее представляет собой интерфейс прикладного программирования (API), описывающий транзакции протокола. Этот API показывается для иллюстрации транзакций и их аргументов и не предназначается для определения реализации (например, путём использования блокирующих функциональных соединений). Он будет описывать входные аргументы и ожидаемые возвратные значения, которые будут использоваться различными транзакциями протокола из очень высокого уровня. Синтаксис транзакции и кодировки определяются в последующих разделах.

8.2.1 Аргумент TransactionRequest

Аргумент TransactionRequest вызывается отправителем. На вызов запроса имеется одна транзакция. Запрос содержит одно или более действий, каждое из которых определяет свой целевой контекст и одну или более команд на контекст.

```
TransactionRequest(TransactionId {  
    ContextID {команда ... команда},  
    ...  
    ContextID {команда ... команда } })
```

Аргумент TransactionID должен определять значение для последующей взаимосвязи с TransactionReply или с откликом TransactionPending от получателя.

Аргумент ContextID должен определять значение, относящееся ко всем командам, которые следуют либо за ближайшей спецификацией аргумента ContextID, либо за завершением запроса TransactionRequest, в зависимости от того, что приходит первым.

Аргумент Command представляет одну из команд, упомянутых в 7.2 (Стык прикладного программирования команд).

8.2.2 Аргумент TransactionReply

Аргумент TransactionReply вызывается получателем. На вызов ответа имеется одна транзакция. Ответ содержит одно или более действий, каждое из которых должно определять свой целевой контекст и одну или более команд на контекст. Аргумент TransactionReply вызывается ответчиком после обработки им TransactionRequest.

Аргумент TransactionRequest обрабатывается:

- когда обрабатываются все действия в данном TransactionRequest; или
- когда происходит ошибка при обработке данного TransactionRequest, за исключением того, когда ошибка происходит в факультативной команде.

Команда обрабатывается, когда обрабатываются все дескрипторы в данной команде.

Аргумент SignalsDescriptor считается обработанным, когда установлено, что данный дескриптор является синтаксически правильным, запрошенные сигналы поддерживаются и находятся в очереди на применение.

Аргументы EventsDescriptor или EventBufferDescriptor считаются обработанными, когда установлено, что данный дескриптор является синтаксически правильным, запрошенные события могут наблюдаться, любые вложенные сигналы могут создаваться, любые вложенные события могут обнаруживаться, а шлюз MG приведен в состояние, при котором события будут обнаружены.

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

Аргумент TransactionID должен быть таким же, как и у соответствующего запроса TransactionRequest.

Аргумент ContextID должен определять значение, относящееся ко всем откликам для данного действия. Аргумент ContextID может быть положительно-выраженным, целым или нулевым.

Каждый из аргументов отклика представляет собой возвращаемое значение, как отмечалось в 7.2, или дескриптор ошибки, если при исполнении команды произошла ошибка. После сбоя команды не обрабатываются, а поэтому отклики на них не создаются.

Исключение происходит, если команда отмечается в запросе на транзакцию как факультативная. Если факультативная команда создаёт ошибку, транзакция продолжает исполняться по-прежнему, так что в этом случае после ошибки ответ содержит отклики после ошибки.

Раздел 7.1.19, озаглавленный “Дескриптор ошибок”, определяет образование дескрипторов ошибок. Ниже обсуждаются несколько отдельных случаев.

Если получатель встречает ошибку при обработке ContextID, то запрошенный отклик на действие будет состоять из Context ID и единственного дескриптора ошибки 422 (синтаксическая ошибка в действии).

Если получатель встречает такую ошибку, что не сможет определить допустимое действие, то возвращается ответ TransactionReply, состоящий из TransactionID и единственного дескриптора ошибки 422 (синтаксическая ошибка в действии). Если конец действия не может быть определён надёжно, но одна или более команд могут быть проанализированы, они обрабатываются, а затем, как при последнем действии в транзакции, передаётся дескриптор ошибки 422 (синтаксическая ошибка в действии). Если получатель встречает такую ошибку, что не может определить допустимую транзакцию, то возвращается TransactionReply с нулевым TransactionID и единственным дескриптором ошибки 403 (синтаксическая ошибка в запросе на транзакцию).

Если конец транзакции не может быть определён надёжно, но одно или более действий могут быть проанализированы, то они обрабатываются, а затем, как при последнем ответе на действие в транзакции, возвращается дескриптор ошибки 403 (синтаксическая ошибка в запросе на транзакцию). Если ни какие действия не могут быть проанализированы, то, как единственный ответ, возвращается дескриптор ошибки 403 (синтаксическая ошибка в запросе на транзакцию).

Если идентификатор TerminationID не может быть определён надёжно, то, как ответ на действие, передаётся дескриптор ошибки 442 (синтаксическая ошибка в команде).

Если конец команды не может быть определён надёжно, то, как ответ на последнее действие, которое может быть проанализировано, возвращается дескриптор ошибки 442 (синтаксическая ошибка в команде).

8.2.3 Аргумент TransactionPending

Получатель вызывает TransactionPending. Аргумент TransactionPending показывает, что транзакция энергично обрабатывается, но обработка ещё не закончена. Он применяется для предотвращения предположения отправителем, что TransactionRequest потерян, когда для завершения транзакции потребуется некоторое время.

TransactionPending(TransactionID { })

Аргумент TransactionID должен быть таким же, как тот же аргумент соответствующего запроса TransactionRequest. Свойство корня (normalMGExecutionTime) устанавливается контроллером MGC, чтобы указать интервал, в пределах которого контроллер MGC ожидает отклика на любую транзакцию от шлюза MG. Другое свойство (normalMGCExecutionTime) устанавливается контроллером MGC, чтобы указать интервал, в пределах которого шлюз MG должен ожидать отклика на любую транзакцию от контроллера MGC. Отправители могут получить более, чем один аргумент TransactionPending на команду. Если в процессе ожидания принимается дублирующий запрос, получатель может послать дублирующий pending немедленно или продолжать ожидание своего таймера для того, чтобы инициировать другой TransactionPending.

8.3 Сообщения

Несколько транзакций могут быть объединены в сообщение. Сообщение имеет заголовок, который содержит информацию о личности отправителя. Идентификатор сообщения (MID) устанавливается на предоставленное имя (например, адрес домена/имя домена/имя устройства) объекта, передающего сообщение. Имя домена предлагается по умолчанию. Объект H.248.1 (MG/MGC) должен единообразно использовать тот же самый MID во всех сообщениях, источником которых является, в течение периода управляющего взаимодействия с сетевым устройством (MGC/MG).

Каждое сообщение содержит номер версии, определяющий версию протокола, которому соответствует сообщение. Номер версии состоит из одной или двух цифр, начиная с версии 1 для существующей версии протокола.

Транзакции в сообщении обрабатываются независимо. Предполагаемый порядок отсутствует; отсутствует сообщение подтверждения приложения или протокола.

Сообщение по существу является транспортным механизмом. Например, на сообщение X, содержащее запросы A, B, и C на транзакции, могут быть даны ответы в виде сообщения Y, содержащего ответы на A и C, а также сообщения Z, содержащего ответ на B. Более того, на сообщение L, содержащее запрос D, и сообщение M, содержащее запрос E, может быть дан ответ в виде сообщения N, содержащего ответы как на D, так и на E.

9 Транспорт

Транспортный механизм протокола должен обеспечивать надёжную транспортировку транзакций между контроллером MGC и шлюзом MG. Транспорт должен оставаться независимым от отдельных передаваемых команд и должен быть применим ко всем состояниям приложения. Имеется несколько типов транспорта, определённых протоколом, которые приводятся в приложениях к данной Рекомендации и другим Рекомендациям серии H.248.x. Дополнительные типы транспорта могут определяться в виде дополнительных Рекомендаций серии H.248. Для транспорта протокола поверх IP, контроллеры MGC должны реализовать как TCP, так и UDP/ALF, а шлюз MG должен реализовывать TCP или UDP/ALF, или оба протокола.

Шлюзу MG сообщаются имя или адрес (такие, как имя DNS или адрес IP) первичного, а также начального или более вторичных контроллеров MGC (см. 7.2.8), т.е. адрес, который используется шлюзом MG для отправки сообщений контроллеру MGC. Если в качестве транспорта протокола применяется TCP или UDP, а порт, на который должен быть отправлен начальный запрос ServiceChange, иным способом неизвестен, то такой запрос должен отправляться на номер порта по умолчанию для данного протокола. Этим номером порта для UDP или TCP в случае операции текстового кодирования является 2944, а в случае операции двоичного кодирования - 2945. Как описывается в 7.2.8, либо шлюз MG, либо контроллер

MGC могут обеспечивать адрес в аргумент ServiceChangeAddress, по которому должны направляться последующие запросы на транзакции, однако отклики (включая отклик на начальный запрос ServiceChange) должны всегда отправляться обратно по тому адресу, который являлся источником соответствующего запроса. Например, в случае IP сетей, им является адрес источника в заголовке IP и номер порта источника в заголовке TCP/UDP/SCTP.

9.1 Упорядочение команд

Данная рекомендация не обязывает, чтобы основной транспортный протокол гарантировал последовательность транзакций, отправленных объекту. Это свойство направлено на максимизацию своевременности действий, однако имеет несколько недостатков. Например:

- Команды Notify могут задерживаться и прибывать к контроллеру MGC после передачи новых команд, изменяющих дескриптор EventsDescriptor.
- Если новая команда передаётся до того, как подтверждена предыдущая, то нет гарантии, что предыдущая команда будет исполнена перед новой командой.

Контроллеры медиашлюзов, которые хотят гарантировать последовательную работу медиашлюза, могут применять следующие правила. Эти правила касаются команд, которые находятся в различных транзакциях. Команды, находящиеся в одной транзакции, исполняются по порядку (см. раздел 8).

- 1) Когда медиашлюз производит обработку нескольких окончаний, команды, принадлежащие различным окончаниям могут отправляться параллельно, например, в соответствии с моделью, когда каждое окончание (или группа окончаний) управляется своим собственным процессом или своим собственным тредом.
- 2) Обычно на окончании должна быть по большей части одна невыполненная команда (либо Add, либо Modify или Move), если невыполненные команды не находятся в той же транзакции. Однако команда Subtract может появляться в любое время. Как следствие, иногда медиашлюз может получать команду Modify, которая применяется к ранее вычтенному окончанию. Такие команды должны игнорироваться и должен возвращаться код ошибки.
- 3) Для транспортов, которые не гарантируют поочерёдную доставку сообщений (то есть UDP), обычно на данном окончании должна быть в любое время по крайней мере одна невыполненная команда Notify.
- 4) В некоторых случаях явная или неявная команда Subtract с использованием групповых символов замены, применяемая к группе окончаний, может сделать шаг перед ожидающей командой Add. Контроллер медиашлюза должен индивидуально удалить все окончания, для которых команда Add была в ожидании на время глобальной команды групповых символов замены (или подразумеваемых Subtract. Кроме того, новые команды Add для окончаний, названных в процессе использования в дескрипторе Multiplex), не должны передаваться до тех пор, пока не будет подтверждена команда Subtract.
- 5) Команды AuditValue и AuditCapability не являются предметом какого-либо упорядочения.
- 6) Команда ServiceChange должна быть всегда первой командой, переданной шлюзом MG, как это определяется процедурой перезагрузки. Любая другая команда или отклик должны передаваться после команды ServiceChange.

Эти правила не затрагивают устройство ответа на команды, которое всегда должно отвечать на команды.

9.2 Защита от лавинообразной перезагрузки

В том случае, когда электропитание одновременно подаётся на большое число медиашлюзов и все медиашлюзы должны начать транзакцию ServiceChange, контроллер медиашлюза будет с большой вероятностью переполнен, что приведёт к потере сообщений и перезагрузке в процессе важного периода восстановления службы. Чтобы избежать таких скоплений, предлагается следующее поведение:

- 1) Подача электропитания на медиашлюз вызывает перезагрузку таймера случайным числом, равномерно распределённым в интервале от 0 до максимального значения задержки ожидания (MWD). Необходима осторожность для того, чтобы избежать синхронизации при генерации случайных чисел между несколькими медиашлюзами, которые будут использовать тот же самый алгоритм.
- 2) Затем медиашлюз должен ожидать либо завершения этого таймера, или обнаружения активности локального пользователя, такой как, например, переход к снятию трубки в частном медиашлюзе.
- 3) По истечении таймера или при обнаружении активности, медиашлюз должен начать процедуру перезагрузки.

Процедура перезагрузки просто требует от шлюза MG гарантировать, чтобы первое сообщение, которое контроллер медиашлюза принимает от этого MG, является сообщением ServiceChange, информирующим контроллер медиашлюза о перезагрузке.

ПРИМЕЧАНИЕ – Значение MWD является аргументом конфигурации, зависящим от типа медиашлюза. Последующие рассуждения могут быть использованы для определения величины этой задержки в частных шлюзах.

Обычно контроллеры медиашлюзов рассчитываются таким образом, чтобы обрабатывать величину трафика часа наибольшей нагрузки, в течение которого, в среднем 10% линий будут заняты, устанавливая соединения, чья средняя длительность обычно равняется 3 минутам. Как правило, обработка соединения занимает 5-6 транзакций контроллера медиашлюза между каждым медиашлюзом и контроллером медиашлюза. Это простое вычисление показывает, что предполагается, что контроллер медиашлюза в среднем каждые 30 минут обрабатывает 5-6 транзакций для каждого окончания или, говоря по иному, в среднем каждые 5-6 минут примерно одну транзакцию на окончание. Это предполагает, что разумным значением MWD для резидентского шлюза было бы 10-12 минут. В отсутствие явной конфигурации частные шлюзы должны обладать значением MWD, равным 600 секундам.

Аналогичное рассуждение предполагает, чтобы величина MWD должна быть намного меньше для канальных или для деловых шлюзов, потому что они обрабатывают большое число окончаний, а также потому что степень использования этих окончаний много выше 10% в течении часа наибольшей нагрузки и обычно равна 60%. Таким образом, ожидается, что в течение часа наибольшей нагрузки эти окончания вносят вклад в нагрузку контроллера медиашлюза, равный примерно одной транзакции в минуту. Разумным алгоритмом является установка значения MWD на окончание "канала" в шесть раз меньшего MWD на частный шлюз, а также обратно пропорционального числу перезагружающихся окончаний. Например, для шлюза, обрабатывающего линию T1, величина MWD должна быть установлена, равной 2,5 секундам, а для шлюза, обрабатывающего линию T3, - равной 60 миллисекундам.

10 Соображения о безопасности

Этот раздел освещает безопасность при использовании протокола в среде IP.

10.1 Защита соединений протокола

Механизм обеспечения безопасности несомненно необходим, чтобы предотвратить использование протокола, определённого в данной Рекомендации, несанкционированными объектами для установления несанкционированных соединений или внесения помех санкционированным соединениям. Механизмом безопасности для данного протокола при его транспортировке по IP-сетям является IPsec (IETF RFC 2401-2411).

Заголовок АН (IETF RFC 2402) предоставляет аутентификацию исходных данных, безобрывную целостность и факультативную антиответную защиту сообщений, прошедших между шлюзом MG и контроллером MGC. Заголовок ESP (IETF RFC 2406) обеспечивает, при необходимости, конфиденциальность сообщений. Например, если описания сеансов используются для перемещения ключей сеансов, как определяется в SDP, должна запрашиваться служба кодирования ESP.

Реализации протокола, определённого в данной Рекомендации, в которых используется заголовок ESP, ДОЛЖНЫ соответствовать разделу 5 документа IETF RFC 2406, определяющего минимальный набор алгоритмов для проверки целостности и кодирования. Аналогично, реализации, в которых используется заголовок АН ДОЛЖНЫ соответствовать разделу 5 документа IETF RFC 2402, определяющего минимальный набор алгоритмов для проверки целостности при использовании ручных ключей.

Реализации ДОЛЖНЫ использовать IKE (IETF RFC 2409), чтобы допустить более устойчивые возможности манипуляции. Реализации, использующие IKE, ДОЛЖНЫ поддерживать аутентификацию с подписями RSA и кодированием публичных ключей RSA.

10.2 Временная структура АН

Реализация протокола IPsec требует, чтобы заголовки АН или ESP помещались непосредственно после IP-заголовка. На уровне приложений это сделать не легко. Вот почему это представляет проблему развёртывания для протокола, определённого в данной Рекомендации, когда реализация базовой сети не поддерживает протокол IPsec.

В качестве временного решения в пределах заголовка протокола H.248.1 определяется факультативный заголовок АН. Поля заголовка точно такие же, как и у SPI, поля SEQUENCE NUMBER (номер последовательности) и DATA (данные) - как определено в документе IETF RFC 2402. Семантика полей заголовка такая же, как в "режиме транспорта" документа IETF RFC 2402, за исключением вычисления значения проверки целостности (ICV). В протоколе Ipsec величина ICV вычисляется по целому IP-пакету, включая IP-заголовок. Это предотвращает спуфинг IP-адресов. Для сохранения той же самой функциональности вычисление ICV должно выполняться для всех транзакций (сцепленных) в сообщении с присоединённым спереди синтезированным IP-заголовком, состоящим из 32-битового IP-адреса источника, 32-битового IP-адреса доставки и 16-битового UDP-порта доставки, закодированным в виде 20 шестнадцатеричных цифр. При применении временного механизма АН, когда TCP является транспортным уровнем, вышеупомянутый порт UDP становится TCP-портом, а все остальные операции остаются теми же самыми.

Реализации протокола H.248.1 ДОЛЖНЫ использовать протокол IPsec там, где базовая операционная система и транспортная сеть поддерживают протокол IPsec. Реализации протокола с использованием IPv4 ДОЛЖНЫ применять временную структуру АН. Однако, эта временная структура НЕ ДОЛЖНА применяться, когда базовый сетевой уровень поддерживает протокол IPsec. Предполагается, что реализации протокола IPv6 поддерживают IPsec и НЕ ДОЛЖНЫ использовать временную структуру АН.

Все реализации временного АН-механизма ДОЛЖНЫ соответствовать разделу 5 документа IETF RFC 2402, определяющего минимальный набор алгоритмов для проверки целостности при использовании ручных ключей.

Временная АН-структура не обеспечивает защиты от перехвата, соответственно препятствуя третьим сторонам подслушивать соединения, установленные данным окончанием. Также она не обеспечивает защиты от атак воспроизведения. Эти процедуры не обязательно защищают от атак типа отказа в службе из-за неправильного функционирования шлюзов MG или контроллера шлюзов MGCs. Однако, они будут обеспечивать идентификацию этих неправильно функционирующих объектов, которые затем должны лишиться своего разрешения через процедуры технического обслуживания.

10.3 Защита медиасоединений

Протокол позволяет контроллеру MGC обеспечить шлюзы MG "ключами сеансов", которые могут быть использованы для шифрования аудио сообщений, защищающего от перехвата. Характерной проблемой пакетных сетей является "неконтролируемое вторжение". Эта атака может быть предпринята путём направления медиапакетов по IP-адресу и UDP-порту, используемому соединением. Если защита не реализована, пакеты должны распаковываться, а сигналы прослушиваться на "стороне линии".

Основной защитой от этой атаки является допуск пакетов только от известных источников, проверяющих, например, что IP-адрес источника и UDP-порт источника соответствуют значениям, объявленным в дескрипторе Remote. Это имеет два неудобства: защита замедляет установление соединения и может быть обманута спуфингом источника:

- Для того, чтобы задействовать адресную защиту, контроллер MGC должен получить описание удалённого сеанса исходящего шлюза MG и передать его входящему шлюзу MG. Это требует, по крайней мере, одного полного оборота по сети и ставит перед нами дилемму: либо позволить продолжать соединение без ожидания того, когда закончится этот оборот и рисковать, например, "клиппированием" удалённого объявления, или ждать окончания полного оборота по сети и установить более медленные процедуры установления соединения.
- Спуфинг источника только тогда эффективен, когда атакующий может получить действующие пары адресов источника назначения и портов, например, путём прослушивания отрывков трафика. Для борьбы со спуфингом источника можно попытаться управлять всеми точками доступа к сети. Однако практически это трудно осуществить.

Альтернативой проверке адреса источника является шифрование и аутентификация пакетов путём использования секретного ключа, который поставляется в процессе процедуры установления соединения. Это не замедлит скорость установления соединения и обеспечит прочную защиту от спуфинга адреса.

11 Стык управления MG-MGC

Взаимосвязь управления между шлюзом MG и контроллером MGC инициируется при начальной загрузке шлюза MG и извещается посредством сообщения ServiceChange, но может изменяться последующими событиями, такими как сбои или события ручной службы. В то время как протокол не имеет явного механизма для поддержки нескольких контроллеров MGC, управляющих физическим шлюзом MG, он был разработан для поддержки нескольких логических шлюзов MG (в пределах единственного физического шлюза MG), который может соединяться с различными контроллерами MGC.

11.1 Несколько виртуальных шлюзов MG

Физический медиашлюз может быть разделён на один или более виртуальных шлюзов MG. Виртуальный шлюз MG состоит из набора статистически разделённых физических окончаний и/или наборов эфемерных окончаний. Физическое окончание управляется одним контроллером MGC. Модель не требует, чтобы были статистически назначены другие ресурсы, только окончания. Механизм назначения окончаний виртуальным шлюзам MG является административным методом, выходящим за пределы настоящего протокола. Каждый из виртуальных шлюзов MGs представляется контроллеру MGC в качестве законченного пользователя шлюза MG.

Физический шлюз MG может иметь только один сетевой стык, который должен использоваться всеми виртуальными шлюзами MG. В таком случае окончание стороны пакета/элемента является общеиспользуемым. Однако необходимо отметить, что при использовании такие стыки требуют создания эфемерного экземпляра окончания на поток и таким образом, совместное использование окончания является простым. Этот механизм на самом деле ведёт к усложнению, а именно, если в стыке произойдёт случай, шлюз MG всегда должен знать, какому из его управляющих контроллеров MGC должно быть послано уведомление.

При нормальной работе контроллер MGC будет отдавать распоряжение виртуальному шлюзу MG в отношении создания сетевых потоков (если это отправляющая сторона) или ожидания потоковых запросов (если это получающая сторона) и беспорядок не возникнет. Однако, если происходит неожиданное событие, виртуальный MG должен знать, что делать с физическими ресурсами, которыми он управляет.

Если восстановление от события требует манипуляции состоянием физического стыка, только один контроллер MGC должен это сделать. Эти проблемы решаются посредством разрешения любому MGC создать дескрипторы EventsDescriptors для уведомления о таких событиях, однако, только один контроллер MGC может иметь доступ для чтения/записи к свойствам физического стыка; все другие контроллеры MGC имеют доступ только для чтения. Административный механизм используется для определения, какой MGC имеет возможность для чтения/записи и его назначения мастер-контроллером MGC.

Каждый виртуальный шлюз MG имеет своё собственное окончание типа Root Termination. В большинстве случаев значения свойств окончания типа Root устанавливаются независимо каждым контроллером MGC. Там, где может быть только одно значение, аргумент является для всех только для чтения, за исключением мастер-контроллера MGC.

Команда ServiceChange может применяться только к окончанию или ряду окончаний, разделённым к виртуальному шлюзу MG или созданным (в случае эфемерных окончаний) этим виртуальным шлюзом MG.

11.2 Начальная загрузка

Административный механизм, не входящий в рамки данного протокола, предварительно предоставляет шлюзу MG первичный и (факультативно) упорядоченный список вторичных контроллеров MGCs. При начальной загрузке шлюза MG, он вырабатывает команду ServiceChange с процедурой "Restart" на окончании Root своему первичному контроллеру MGC. Если контроллер MGC принимает шлюз MG, он посылает ответ на транзакцию, не включающий аргумент ServiceChangeMgcId. Если контроллер MGC не принимает регистрацию шлюзов MG, он посылает ответ на транзакцию Transaction Reply, обеспечивая адрес альтернативного контроллера MGC для контакта, путём включения аргумента ServiceChangeMgcId.

Если шлюз MG принимает ответ на транзакцию, который включает аргумент ServiceChangeMgcId, он посылает команду ServiceChange контроллеру MGC, определённому

в аргументе `ServiceChangeMgcId`. Он продолжает этот процесс до тех пор, пока не получит управляющий контроллер, который примет его регистрацию, или он не получает ответа. В случае неудачи получить ответ либо от первичного контроллера или назначенного преемника шлюз MG пробует по порядку свои предварительно предоставленные вторичные контроллеры MGC. Если шлюз MG не способен установить отношения управления с любым контроллером, он должен ждать случайное время, как описано в 9.2, а затем начать снова соединяться со своим первичным, а если необходимо, со своими вторичными контроллерами MGC.

Возможно, что ответ на команду `ServiceChange` с процедурой `Restart` будет потерян и команда будет принята шлюзом MG прежде, чем получен отклик на команду `ServiceChange`. Шлюз MG должен выдать сообщение `Error 505` (Запрос на транзакцию получен перед получением ответа на команду `ServiceChange`).

11.3 Согласование версии протокола

Первая команда `ServiceChange` от шлюза MG должна содержать в аргументе `ServiceChangeVersion` номер версии протокола, поддерживаемого шлюзом MG. Если контроллер MGC поддерживает только нижнюю версию протокола, то при получении такого сообщения контроллер MGC должен послать `ServiceChangeReply` с нижней версией и после этого все сообщения между шлюзом MG и контроллером MGC должны соответствовать нижней версии протокола. Если шлюз MG не может соответствовать и установил транспортное соединение с контроллером MGC, он должен закрыть это соединение. В любом случае он должен отвергнуть все последующие запросы со стороны контроллера MGC с сообщением `Error 406` (Версия не поддерживается).

Если контроллер MGC поддерживает более высокую версию, чем шлюз MG, но может поддержать и нижнюю версию, предложенную шлюзом MG, он должен послать сообщение `ServiceChangeReply` с нижней версией и после этого все сообщения между шлюзом MG и контроллером MGC должны соответствовать нижней версии протокола. Если контроллер MGC не может соответствовать, он должен отвергнуть взаимосвязь с сообщением `Error 406` (Версия не поддерживается).

Согласование версии протокола может также произойти при наличии "handoff" и "failover" команды `ServiceChanges`.

При расширении протокола новыми версиями должны выполняться следующие правила:

- 1) Существующие элементы протокола, т.е. процедуры, аргументы, дескриптор, свойство, значения не должны изменяться пока не исправлены ошибки протокола или не становится необходимым изменить работы службы, которая поддерживается протоколом.
- 2) Семантика команды, аргумент, дескриптор, свойство или значение не должны изменяться.
- 3) Установленные правила форматирования и шифрования сообщений и аргументов не должны изменяться.
- 4) При нахождении устаревших информационных элементов, они могут быть отмечены как не используемые. Однако, идентификатор такого информационного элемента будет помечен как зарезервированный. Таким образом он не может использоваться в будущих версиях.

11.4 Сбой шлюза MG

Если происходит сбой шлюза MG, но он в состоянии отправить сообщение контроллеру MGC, то он посылает команду `ServiceChange` с соответствующим методом (`graceful` или

forced) и определяет идентификатор TerminationID типа Root. При возвращении к обслуживанию, он посылает команду ServiceChange с методом "Restart".

Контроллер MGC, разрешающий отправить двойные сообщения обоим шлюзам MG, вмещает пары шлюзов MG, которые в состоянии резервированного обхода отказа одного из шлюзов MG. Только работающий шлюз MG должен принять или отклонить транзакции. При обходе отказа первичный MG отправляет команду ServiceChange с методом "Failover" и причиной "MG Impending Failure". Затем контроллер MGC использует вторичный шлюз MG в качестве активного шлюза MG. Когда причина ошибки исправлена, работающий шлюз MG может передать команду "ServiceChange" с методом "Restart".

ПРИМЕЧАНИЕ – Шлюзы с резервированным обходом отказа требуют надёжного транспорта, т.к. протокол не обеспечивает средств вторичному шлюзу MG, исполняющему ALF для подтверждения сообщений, переданных контроллером MGC.

11.5 Сбой контроллера MGC

Если шлюз MG обнаруживает сбой своего управляющего контроллера MGC, то он пытается соединиться со следующим контроллером MGC из своего предварительно предоставленного списка. Он начинает свои попытки с начального контроллера (первичный MGC), если последний не является вышедшим из строя контроллером MGC; в противном случае он начинает со своего первого вторичного контроллера. Шлюз MG отправляет сообщение ServiceChange с методом "Failover" и причиной "MGC Impending Failure". Если шлюз MG не способен установить отношения управления с любым контроллером MGC, он должен ждать случайное время, как описано в 9.2, а затем начать снова соединяться со своим первичным и (если необходимо) со своими вторичными контроллерами MGC. При соединении со своим предыдущим управляющим контроллером MGC, шлюз MG отправляет сообщение ServiceChange с методом "Disconnected".

При частичном сбое или по причинам неавтоматизированного технического обслуживания контроллер MGC может пожелать предписать контролируемым им шлюзам использовать другой контроллер. Для этого, он отправляет команду ServiceChange с методом "HandOff" в шлюз MG и её назначенную замену в ServiceChangeMgcId. Если поддерживается "HandOff", то шлюз MG должен послать сообщение ServiceChange с методом "Handoff" и причиной "MGC directed change" назначенному контроллеру MGC. Если он не получит ответ от назначенного контроллера MGC, шлюз MG должен вести себя так, как если бы его контроллер MGC вышел из строя, и начать соединяться с вторичными контроллерами MGC, как это указывалось в предыдущем параграфе. Если шлюз MG не может установить отношения управления с любым контроллером MGC, он должен ждать случайное количество времени, как описано в 9.2, а затем начать снова соединяться со своим первичным и, если необходимо, со своими вторичными контроллерами MGC.

Рекомендация о том, каким образом контроллеры MGC касаются информации состояния поддержки плавного переключения, не разработана; считается, что эта тема находится за рамками данной Рекомендации. Контроллер MGC и шлюз MG могут сделать следующие шаги, когда происходит плавное переключение. Когда контроллер MGC начинает HandOff, переключение должно быть прозрачным для действий на медиашлюзе. Транзакции могут исполняться в любом порядке и могут находиться в действии при исполнении команды ServiceChange. Соответственно, команды, находящиеся в действии, продолжают, а ответы на все команды от исходного контроллера MGC должны посылаться на транспортный адрес, из которого они были отправлены. Если служебные отношения с передающим контроллером MGC закончились, ответы должны отбрасываться. Шлюз MG может получить невыполненные ответы на транзакции от нового MGC. До тех пор, пока управляющее взаимодействие не установлено, новые сообщения не должны отправляться к новому контроллеру MGC. Запросы на повторные транзакции должны направляться к новым

контроллерам MGC. Шлюз MG должен поддерживать состояние на всех окончаниях и контекстах.

Возможно, что контроллер MGC может быть реализован таким образом, что вышедший из строя контроллер MGC заменяется работающим контроллером MGC, причём идентичность нового контроллера MGC является той же самой, что и вышедшего из строя. В таком случае аргумент ServiceChangeMgcId будет определяться с предыдущим значением, а поведение шлюза MG должно быть таким, как если значение изменилось и послать сообщение ServiceChange, как изложено выше.

Пары контроллеров MGC, допускающие резервированный обход отказа, могут уведомить управляемые шлюзы MG об обходе отказа посредством вышеупомянутого механизма.

12 Определение комплекта

Первичным механизмом расширения являются комплекты. Комплекты определяют дополнительные свойства, события, сигналы и статистику, которые могут происходить на окончаниях.

Комплекты, определённые IETF, будут опубликованы в отдельных RFC.

Комплекты, определённые ITU-T, могут быть опубликованы в соответствующих Рекомендациях (например, в Рекомендациях серии H.248.x).

- 1) Должен определяться общедоступный документ или стандартный документ собрания, на который можно ссылаться как на документ, который даёт описание комплектов, следуя изложенным выше директивам.
- 2) Документ должен определять версию комплекта, которую он описывает.
- 3) Документ должен быть доступен на общедоступном веб-сервере и должен иметь постоянный URL. Веб-узел должен обеспечить механизм для возврата замечаний и соответствующих откликов.

12.1 Директивы по определению комплектов

Комплекты определяют свойства, события, сигналы и статистику.

В соответствии с директивами, приведенными в 13.2, комплекты могут также определять новые коды ошибок. Это является предметом документального удобства: документация по комплекту представляется в IANA в поддержку регистрации кода ошибки. Если комплект изменяется, необходимо обеспечить IANA новым справочным документом в поддержку кода ошибки, пока не будет изменяться само описание кода ошибки. Имена всех таких определённых конструкций должны состоять из идентификатора PackageID (который однозначно определяет комплект) и идентификатора ID параграфа (который однозначно идентифицирует параграф в этом комплекте). При текстовом кодировании оба идентификатора должны разделяться знаком ("/") косой черты. Пример: togen/playtone представляет собой текстовое кодирование обращения к сигналу генерации тона в комплекте тоновой генерации.

Комплект будет содержать следующие секции:

12.1.1 Комплект

Общее описание комплекта, указывающее:

Имя комплекта: только описательное

PackageID: является идентификатором

Описание:

Версия:

Новая версия комплекта может только присоединить дополнительные свойства, случаи, сигналы, статистику и новые возможные значения для существующих аргументов, описанных в исходном комплекте. Никакие удаления и изменения не разрешаются. Версия представляет собой целое число в диапазоне от 1 до 99.

Предназначен только для расширения (факультативно): да.

Это показывает, что комплект специально предназначен для расширения другими, не для прямой ссылки. Например, комплект не может иметь никакой собственной функции или быть собственно бессмысленным. При сообщении о комплектах шлюз MG НЕ ДОЛЖЕН публиковать этот PackageID.

Extends (факультативный): существующий дескриптор комплекта

Комплект может расширить существующий комплект. Версия исходного комплекта должна быть определена. Когда комплект расширяет другой комплект, он должен только присоединить дополнительные свойства, события, сигналы, статистику и новые возможные значения для существующего аргумента, описанного в исходном комплекте. Расширенный комплект не должен переопределять или перегружать идентификатор, определённый в исходном комплекте и комплектах, которые он мог расширить (несколько уровней расширения). Следовательно, если комплект В версии 1 расширяет комплект А версии 1, то версия 2 комплекта В не сможет расширить комплект А версии 2, если комплект А версии 2 уже определяет имя в комплекте В версии 1.

12.1.2 Свойства

Свойства, определённые комплектом, указывающие:

Имя свойства: только описательное

PropertyID: является идентификатором

Описание:

Тип: Один из:

Булевый

Строка: строка типа UTF-8

Строка октета: Количество октетов. См. Приложения А и В.3 для кодирования

Целое число: целое число со знаком размером 4 байта

Двойное число: целое число со знаком размером 8 байт

Символ: кодирование одной буквы уникодом типа UTF-8. Может иметь более, чем один октет.

Перечисление: одно из списка возможных однозначных значений (см. 12.3)

Подпись: список нескольких значений из списка. Тип подписки также ДОЛЖЕН быть определён. Тип должен быть выбран из типов, определённых в этом параграфе (за исключением подписки). Например, Тип: подпись перечисления. Кодирование подписок определяется в Приложениях А и В.3.

Возможные значения:

Комплект ДОЛЖЕН определять либо определённый набор значений, или описание того, как определяются значения. Комплект также ДОЛЖЕН определять значение по умолчанию или поведение по умолчанию, когда в своём дескрипторе значение пропущено. Например, комплект может определять, что процедуры, относящиеся к свойству, приостанавливаются, когда его значение пропущено. Значение по умолчанию (но не процедуры) может быть определено как условное.

Определено в:

В каком дескрипторе H.248.1 определяется свойство. В дескрипторе LocalControl - потоко-зависимые свойства. В дескрипторе TerminationState – потоко-независимые свойства. Ожидается, что это наиболее общие случаи, однако возможно, что свойства определяются в других дескрипторах.

Характеристики: чтение/запись или и то и другое, а также всеобщая (факультативно):

Показывает является ли свойство и является ли оно всеобщим. Если Global пропущено, свойство не является всеобщим. Если свойство декларируется как всеобщее, значение свойства используется совместно всеми окончаниями, реализующими комплект.

12.1.3 События

События определяются комплектом, указывающим:

Имя события: только описательное

EventID: является идентификатором

Описание:

Аргументы дескриптора EventsDescriptor:

Аргументы используются контроллером MGC для конфигурирования события и использования в дескрипторе EventsDescriptor. См. 12.2.

Аргументы дескриптора ObservedEventsDescriptor:

Аргументы, возвращённые контроллеру MGC в запросах Notify и в откликах на запросы команд от контроллера MGC, которые проверяют дескриптор ObservedEventsDescriptor и используются в дескрипторе ObservedEventsDescriptor. См. 12.2.

12.1.4 Сигналы

Сигналы определяются комплектом, указывающим:

Имя сигнала: только описательное

SignalID: Является идентификатором. Идентификатор SignalID используется в дескрипторе SignalsDescriptor

Description

Тип сигнала: один из:

OO (включено/выключено)

TO (перерыв)

BR (короткий)

ПРИМЕЧАНИЕ – SignalType может быть определён так, что он является зависимым от значения одного или более аргументов. Комплект ДОЛЖЕН определять тип сигналов по умолчанию. Если типом сигнала по умолчанию является ТО, комплект ДОЛЖЕН определять продолжительность по умолчанию, которая может быть предоставлена. Продолжительность по умолчанию для сигнала типа BR не имеет смысла.

Длительность: в сотых долях секунды

Дополнительные аргументы: см. 12.2

12.1.5 Статистика

Статистика определяется комплектом, указывающим:

Имя статистики: только описательное

StatisticID: является идентификатором

Идентификатор StatisticID используется в дескрипторе StatisticsDescriptor

Описание:

Единицы: единица измерения, например, миллисекунды, комплекты

12.1.6 Процедуры

Дополнительное руководство по использованию комплекта.

12.2 Директивы по определению аргументов событий и сигналов

Имя аргумента: только описательное

ParameterID: является идентификатором. Текстовый идентификатор ParameterID аргументов для событий и сигналов не должен начинаться с "EPA" и "SPA", соответственно. Текстовый идентификатор ParameterID также не должен быть "ST", "Stream", "SY", "SignalType", "DR", "Duration", "NC", "NotifyCompletion", "KA", "Keepalive", "EB", "Embed", "DM" или "DigitMap".

Тип: Один из:

Булевый

Строка: октетная строка типа UTF-8

Октетная строка: количество октетов. См. Приложения А и В.3 для кодирования

Целое число: целое число со знаком размером 4 байта

Двойное число: целое число со знаком размером 8 байт

Символ: кодирование одной буквы уникодом типа UTF-8. Может иметь более, чем один октет.

Перечисление: одно из списка возможных однозначных значений (см. 12.3)

Подсписок: список нескольких значений из списка (не поддерживается для статистики). Тип подсписка также ДОЛЖЕН быть определён. Тип должен быть выбран из типов, определённых в этом параграфе (за исключением подсписка). Например, Тип: подсписок перечисления. Кодирование подсписков определяется в Приложениях А и В.3.

Возможные значения:

Комплект ДОЛЖЕН определять либо определённый набор значений, или описание того, как определяются значения. Комплект также ДОЛЖЕН определять значение по умолчанию или поведение по умолчанию, когда в своём дескрипторе значение

пропущено. Например, комплект может определять, что процедуры, относящиеся к аргументу, приостанавливаются, когда его значение пропущено. Значение по умолчанию (но не процедуры) может быть определено как условное.

Описание:

12.3 Списки

Возможные значения для аргументов включают перечисления. Перечисления могут определяться в списке. Рекомендуется, чтобы список был бы зарегистрирован IANA так что комплекты, которые расширяют список, могут определяться, не беспокоясь о конфликтующих именах.

12.4 Идентификаторы

При текстовом кодировании идентификаторы должны быть строками длиной до 64 символов, не содержать пробелов, начинаться с алфавитного символа и содержать буквенно-цифровые символы и/или цифры и, возможно, включать специальный знак подчёркивания ("_").

В случае двоичного кодирования идентификаторы имеют длительность 2 октета.

Как текстовые, так и двоичные значения должны быть определены для каждого идентификатора, включая идентификаторы, используемые как значения в перечислимых типах.

12.5 Регистрация комплектов

Комплект может быть зарегистрирован IANA по причинам возможности взаимодействия. См. Раздел 13 относительно соображений о IANA.

13 Соображения о IANA

13.1 Комплекты

Для регистрации комплекта в IANA ДОЛЖНЫ удовлетворяться следующие соображения:

- 1) Для каждого комплекта регистрируется уникальное имя строки, уникальный серийный номер и номер. Имя строки используется с текстовым кодированием. Серийный номер должен использоваться с двоичным кодированием. Серийные номера в диапазоне от 0x8000 до 0xFFFF резервируются для частного использования. Серийный номер 0 зарезервирован.
- 2) Для данного контакта должны быть определены контактное имя, адрес email (электронная почта) и почтовый адрес. По необходимости контактная информация должна обновляться определяющей организацией.
- 3) Ссылка на документ, который описывает комплект и который должен быть общедоступным:
Документ должен определять версию комплекта, которую он описывает.
Если документ является общедоступным, он должен размещаться на общедоступном веб-сервере и иметь постоянный URL. Веб-узел должен обеспечить механизм для возврата замечаний и соответствующих откликов.
- 4) Комплекты, зарегистрированные иными, чем признанные организациями по стандартизации, должны иметь минимальную длину имени комплекта, равную 8 символам.

- 5) При выполнении всех других условий все иные имена комплектов обслуживаются по принципу первым пришёл-первым обслужен.

13.2 Коды ошибок

Для регистрации комплекта в IANA ДОЛЖНЫ удовлетворяться следующие соображения:

- 1) Для каждой ошибки регистрируется номер ошибки и строка в одну линию (максимум 80 символов).
- 2) В общедоступный документ должно быть включено полное описание условий, при которых обнаруживается ошибка. Описание должно быть достаточно ясным, чтобы отличить данную ошибку от всех других существующих кодов ошибок.
- 3) Данный документ должен находиться на общедоступном веб-сервере и иметь постоянный URL.
- 4) Номера ошибок, регистрируемые признанными организациями стандартизации, должны иметь 3- или 4-символьные номера ошибок.
- 5) Номера ошибок, регистрируемые всеми другими организациями или лицами должны иметь 4-символьные номера ошибок.
- 6) Номер ошибки не должен переопределяться или изменяться той организацией или лицом, которые исходно его определили, а также их наследниками или правопреемниками.

13.3 Причины ServiceChange

Для регистрации комплекта в IANA ДОЛЖНЫ удовлетворяться следующие соображения:

- 1) Для каждой причины регистрируется уникальный код причины в одну фразу, максимальной длительностью 80 символов.
- 2) В общедоступный документ должно быть включено полное описание условий, при которых используется причина. Описание должно быть достаточно ясным, чтобы отличить данную причину от всех других существующих причин.
- 3) Данный документ должен находиться на общедоступном веб-сервере и иметь постоянный URL.

Приложение А

Двоичное кодирование протокола

В данном приложении определяется синтаксис сообщений с использованием нотации, приведенной в Рекомендации МСЭ-Т X.680, *Информационная технология – Австральный Синтаксис Нотация Один (ASN.1): Спецификация базовой нотации*. Сообщения должны кодироваться для передачи путём применения основных правил кодирования, определённых в Рекомендации МСЭ-Т X.690, *Информационная технология – Правила кодирования ASN.1: Спецификация Основных Правил Кодирования (BER), Канонические Правила Кодирования (CER) и Выделенные Правила Кодирования*.

А.1 Кодирование групповых символов замены

Протоколом разрешается использование групповых символов замены ALL и CHOOSE. Это позволяет контроллеру MGC частично определить идентификаторы Termination ID и даёт возможность шлюзу MG выбрать из значений, соответствующих частичной спецификации. Идентификаторы Termination ID могут кодировать иерархию имён. Эта иерархия

предоставляется. Например, идентификатор TerminationID может состоять из группы каналов, канала в пределах группы и линии. Применение групповых символов замены должно быть возможно на всех уровнях. Следующие параграфы объясняют как это достигается.

В описании ASN.1 для идентификаторов Termination ID используется октетная строка длиной до 8 октетов. Это означает, что идентификаторы состоят из, по большей части, 64 бит. Перед полностью определённым идентификатором Termination ID может быть последовательность полей групповых символов замены. Поле группового символа замены имеет длину в один октет. Бит 7 (старший разряд) этого октета определяет какой из групповых символов замены активизирован: если этот бит равен 1, то используется групповой символ замены ALL; если же этот бит равен 0, то используется групповой символ замены CHOOSE. Бит 6 поля групповых символов замены определяет принадлежит ли использование групповых символов замены к одному уровню в иерархической именной схеме (значение бита равно 0) или к уровню иерархии, определённого в поле групповых символов замены, включая все нижние уровни (значение бита равно 1). Биты от 0 до 5 поля групповых символов замены определяют позицию бита в идентификаторе Termination ID, с которой начинается использование групповых символов замены.

Мы иллюстрируем эту схему на нескольких примерах. В этих примерах старший разряд находится слева.

Допустим, что идентификаторы Termination ID имеют длину в 3 октета и что каждый октет представляет уровень в иерархической именной схеме. Правомерным идентификатором Termination ID является:

00000001 00011110 01010101.

Адресация имён ALL с префиксом 00000001 00011110 осуществляется следующим образом:

поле групповых символов замены: 10000111

идентификатор Termination ID: 00000001 00011110 xxxxxxxx.

Значение разрядов, отмеченных "x", не относится к теме и должно игнорироваться получателем.

Указание получателю, что он должен выбрать имя с 00011110 во втором октете, осуществляется следующим образом:

поля групповых символов замены: 00010111, за которыми следуют 00000111

идентификатор Termination ID: xxxxxxxx 00011110 xxxxxxxx.

Первое поле групповых символом замены указывает на групповой символ замены CHOOSE для уровня в именной иерархии, начинающегося с разряда 23, наивысшего уровня в нашей предполагаемой именной схеме. Второе поле групповых символов замены указывает на групповой символ замены CHOOSE для уровня в именной иерархии, начинающегося с разряда 7, наименьшего уровня в нашей предполагаемой именной схеме.

В заключение, имя с использованием группового символа замены CHOOSE с наивысшим уровнем имени, равным 00000001 определяется следующим образом:

поле групповых символов замены: 01001111

идентификатор Termination ID: 00000001 xxxxxxxx xxxxxxxx .

Значение бита 1 в разряде 6 первого октета поля групповых символов замены указывает, что использование групповых символов замены принадлежит к определённому уровню в именной иерархии и всем нижним уровням.

Идентификаторы Context IDs также могут использоваться с групповыми символами замены. Однако, в случае идентификаторов Context ID определение частичных имён не разрешается. Идентификатор Context ID 0x0 ДОЛЖЕН использоваться для указания НУЛЕВОГО контекста, идентификатор Context ID 0xFFFFFFFF SHALL be used to indicate a CHOOSE wildcard, and Context ID 0xFFFFFFFF ДОЛЖЕН использоваться для указания на групповой символ замены ALL.

Идентификатор TerminationID 0xFFFFFFFFFFFFFFFF ДОЛЖЕН использоваться для указания на окончание ROOT.

A.2 Спецификация синтаксиса ASN.1

Этот раздел содержит спецификацию ASN.1 синтаксиса протокола H.248.1.

ПРИМЕЧАНИЕ 1 – В случае применения транспортного механизма, в котором используется кадрирование прикладного уровня определение транзакции изменяется как это указывается ниже. За определением, применимым в этом случае, обращайтесь к приложению или к Рекомендации серии H.248.x, определяющим транспортный механизм.

ПРИМЕЧАНИЕ 2 – Нижеизложенная спецификация ASN.1 содержит раздел, определяющий TerminationIDList в виде последовательности идентификаторов TerminationID. Длина этой последовательности ДОЛЖНА быть равна одному, за возможным исключением при использовании в contextAuditResult.

ПРИМЕЧАНИЕ 3 – Данная спецификация синтаксиса не накладывает каких бы то ни было ограничений на включения элементов и значения. Некоторые дополнительные ограничения отмечаются в комментариях, а другие ограничения находятся в тексте данной Рекомендации. Эти дополнительные ограничения являются частью протокола, хотя и не навязываются данной Рекомендацией.

ПРИМЕЧАНИЕ 4 – В данном приложении модуль ASN.1 использует классы октетных строк для кодирования значений аргумента свойства, значений аргумента сигнала и аргумента события, а также статистики.

Сначала значение BER-кодируется на основании на своего типа при использовании нижеследующей таблицы. Затем результат этого BER-кодирования кодируется в виде октетной строки ASN.1, "дважды заворачивая" значение. Формат, определённый в Приложении С или в комплекте, связан с BER-кодированием следующей таблицей:

Тип, определённый в комплекте	BER-тип ASN.1
Строка	IA5String или UTF8String (Примечание 4)
Целое число (4 октета)	ЦЕЛОЕ ЧИСЛО
Двойное число (8 октетов, со знаком, целое)	ЦЕЛОЕ ЧИСЛО (Примечание 3)
Символ (UTF-8, примечание 1)	IA5String
Перечисление	ПЕРЕЧИСЛИМЫЙ
Булевый	БУЛЕВЫЙ
Целое без знака (примечание 2)	ЦЕЛОЕ ЧИСЛО (Примечание 3)
Октет (Строка)	ОКТЕТНАЯ СТРОКА

Тип, определённый в комплекте	BER-тип ASN.1
<p>ПРИМЕЧАНИЕ 1 – Может быть больше, чем один байт.</p> <p>ПРИМЕЧАНИЕ 2 – Целое число без знака представлено в Приложении С.</p> <p>ПРИМЕЧАНИЕ 3 – BER-кодирование ЦЕЛОГО ЧИСЛА не подразумевает использование 4 байт.</p> <p>ПРИМЕЧАНИЕ 4 – Строка должна быть закодирована как IA5String, когда содержимое-все символы ASCII, но как UTF8String, если она содержит какие-либо не ASCII-символы.</p>	

См. 8.7/X.690, где приведено определение кодирования значения октетной строки.

МЕДИАСЛЮЗ-АВТОМАТИЧЕСКИЕ ТЕГИ ОПРЕДЕЛЕНИЙ УПРАВЛЕНИЯ ::= BEGIN

MegacoMessage ::= ПОСЛЕДОВАТЕЛЬНОСТЬ

```
{
  authHeader      Аутентификация заголовка ФАКУЛЬТАТИВНО,
  mess            Сообщение
}
```

AuthenticationHeader ::= ПОСЛЕДОВАТЕЛЬНОСТЬ

```
{
  secParmIndex    SecurityParmIndex,
  seqNum          SequenceNum,
  ad              AuthData
}
```

SecurityParmIndex ::= ОКТЕТНАЯ СТРОКА(РАЗМЕР(4))

SequenceNum ::= ОКТЕТНАЯ СТРОКА(РАЗМЕР(4))

AuthData ::= ОКТЕТНАЯ СТРОКА(РАЗМЕР(12..32))

Message ::= ПОСЛЕДОВАТЕЛЬНОСТЬ

```
{
  version          ЦЕЛОЕ ЧИСЛО(0..99),
  -- Версия протокола, определённая здесь, равна 1.
  mId              Mid, -- Имя/адрес инициатора сообщения
  messageBody CHOICE
  {
    messageError   ErrorDescriptor,
    transactions   ПОСЛЕДОВАТЕЛЬНОСТЬ Транзакций
  },
  ...
}
```

Mid ::= ВЫБОР

```
{
  ip4Address       IP4Address,
  ip6Address       IP6Address,
  domainName       DomainName,
  deviceName       PathName,
  mtpAddress       ОКТЕТНАЯ СТРОКА(РАЗМЕР(2..4)),
  -- Структура адресации mtpAddress:
  --      25 - 15      0
  --      | PC        | NI |
  --      24 - 14 бит  2 бит
  -- ПРИМЕЧАНИЕ – 14 бит определены для международного использования.
  -- Существует две национальные опции с кодом точки, равным 16 или 24 бит.
```

```

-- Для октетного выравнивания mtpAddress, коды старших разрядов должны
-- быть равны 0.
...
}

DomainName ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    name      IA5String,
    -- Имя начинается с буквенно-цифрового числа, за которым следует
    -- последовательность буквенно-цифровых чисел, дефисов и точек. Две
    -- последовательные точки не должны встречаться.
    portNumber ЦЕЛОЕ ЧИСЛО(0..65535) ФАКУЛЬТАТИВНО
}

IP4Address ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    address      ОКТЕТНАЯ СТРОКА(РАЗМЕР(4)),
    portNumber   ЦЕЛОЕ ЧИСЛО(0..65535) ФАКУЛЬТАТИВНО
}

IP6Address ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    address      ОКТЕТНАЯ СТРОКА(РАЗМЕР(16)),
    portNumber   ЦЕЛОЕ ЧИСЛО(0..65535) ФАКУЛЬТАТИВНО
}

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

Transaction ::= ВЫБОР
{
    transactionRequest      TransactionRequest,
    transactionPending      TransactionPending,
    transactionReply        TransactionReply,
    transactionResponseAck  TransactionResponseAck,
    -- использование подтверждений откликов зависит от основного
    -- транспорта
    ...
}

TransactionId ::= ЦЕЛОЕ ЧИСЛО(0..4294967295)    -- 32-разрядное целое число
                                                    -- без знака

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

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

TransactionReply ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    transactionId      TransactionId,
    immAckRequired     NULL ФАКУЛЬТАТИВНО,
    transactionResult  ВЫБОР
    {
        transactionError      ErrorDescriptor,
        actionReplies          ПОСЛЕДОВАТЕЛЬНОСТЬ OF ActionReply
    }
}

```

```

    },
    ...
}

TransactionResponseAck ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF TransactionAck
TransactionAck ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    firstAck          TransactionId,
    lastAck           TransactionId ФАКУЛЬТАТИВНО
}

ErrorDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    errorCode         ErrorCode,
    errorText         ErrorText ФАКУЛЬТАТИВНО
}

ErrorCode ::= ЦЕЛОЕ ЧИСЛО(0..65535)
-- См. раздел 13 для соображений о IANA в отношении кодов ошибок

ErrorText ::= IA5String

ContextID ::= ЦЕЛОЕ ЧИСЛО(0..4294967295)

-- Значение NULL Контекста: 0
-- Значение CHOOSE Контекста: 4294967294 (0xFFFFFFFFE)
-- Значение ALL Контекста: 4294967295 (0xFFFFFFFF)

ActionRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    contextId         ContextID,
    contextRequest    ContextRequest ФАКУЛЬТАТИВНО,
    contextAttrAuditReq ContextAttrAuditRequest ФАКУЛЬТАТИВНО,
    commandRequests  ПОСЛЕДОВАТЕЛЬНОСТЬ OF CommandRequest
}

ActionReply ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    contextId         ContextID,
    errorDescriptor   ErrorDescriptor ФАКУЛЬТАТИВНО,
    contextReply      ContextRequest ФАКУЛЬТАТИВНО,
    commandReply      ПОСЛЕДОВАТЕЛЬНОСТЬ OF CommandReply
}

ContextRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    priority          ЦЕЛОЕ ЧИСЛО(0..15) ФАКУЛЬТАТИВНО,
    emergency         БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    topologyReq      ПОСЛЕДОВАТЕЛЬНОСТЬ OF TopologyRequest ФАКУЛЬТАТИВНО,
    ...
}

ContextAttrAuditRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    topology         NULL ФАКУЛЬТАТИВНО,
    emergency        NULL ФАКУЛЬТАТИВНО,
    priority         NULL ФАКУЛЬТАТИВНО,
    ...
}

CommandRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    command          Command,

```

```

    optional          NULL ФАКУЛЬТАТИВНО,
    wildcardReturn   NULL ФАКУЛЬТАТИВНО,
    ...
}

Command ::= ВЫБОР
{
    addReq           AmmRequest,
    moveReq          AmmRequest,
    modReq           AmmRequest,
    -- Запросы Add, Move, Modify имеют одинаковые аргументы
    subtractReq      SubtractRequest,
    auditCapRequest  AuditRequest,
    auditValueRequest AuditRequest,
    notifyReq        NotifyRequest,
    serviceChangeReq ServiceChangeRequest,
    ...
}

CommandReply ::= ВЫБОР
{
    addReply         AmmsReply,
    moveReply        AmmsReply,
    modReply         AmmsReply,
    subtractReply    AmmsReply,
    -- Запросы Add, Move, Modify имеют одинаковые аргументы
    auditCapReply    AuditReply,
    auditValueReply  AuditReply,
    notifyReply      NotifyReply,
    serviceChangeReply ServiceChangeReply,
    ...
}

TopologyRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    terminationFrom  TerminationID,
    terminationTo    TerminationID,
    topologyDirection PERЕЧИСЛИМЫЙ
    {
        bothway(0),
        isolate(1),
        oneway(2)
    },
    ...
}

AmmRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    terminationID    TerminationIDList,
    descriptors      ПОСЛЕДОВАТЕЛЬНОСТЬ OF AmmDescriptor,
    -- По большей части один дескриптор каждого типа (см. AmmDescriptor),
    -- разрешённый в последовательности.
    ...
}

AmmDescriptor ::= ВЫБОР
{
    mediaDescriptor  MediaDescriptor,
    modemDescriptor  ModemDescriptor,
    muxDescriptor    MuxDescriptor,
    eventsDescriptor EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor SignalsDescriptor,
    digitMapDescriptor DigitMapDescriptor,
    auditDescriptor  AuditDescriptor,

```

```

...
}

AmmsReply ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    terminationID          TerminationIDList,
    terminationAudit       TerminationAudit ФАКУЛЬТАТИВНО,
    ...
}

SubtractRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    terminationID          TerminationIDList,
    auditDescriptor        AuditDescriptor ФАКУЛЬТАТИВНО,
    ...
}

AuditRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    terminationID          TerminationID,
    auditDescriptor        AuditDescriptor,
    ...
}

AuditReply ::= ВЫБОР
{
    contextAuditResult     TerminationIDList,
    error                   ErrorDescriptor,
    auditResult             AuditResult,
    ...
}

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

TerminationAudit ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF AuditReturnParameter

AuditReturnParameter ::= ВЫБОР
{
    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 ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    auditToken БИТОВАЯ СТРОКА
    {

```

```

    muxToken(0), modemToken(1), mediaToken(2),
    eventsToken(3), signalsToken(4),
    digitMapToken(5), statsToken(6),
    observedEventsToken(7),
    packagesToken(8), eventBufferToken(9)
  } ФАКУЛЬТАТИВНО,
  ...
}

NotifyRequest ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
  terminationID          TerminationIDList,
  observedEventsDescriptor ObservedEventsDescriptor,
  errorDescriptor        ErrorDescriptor ФАКУЛЬТАТИВНО,
  ...
}

NotifyReply ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
  terminationID          TerminationIDList,
  errorDescriptor        ErrorDescriptor ФАКУЛЬТАТИВНО,
  ...
}

ObservedEventsDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
  requestId              RequestID,
  observedEventList      ПОСЛЕДОВАТЕЛЬНОСТЬ OF ObservedEvent
}
ObservedEvent ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
  eventName              EventName,
  streamID              StreamID ФАКУЛЬТАТИВНО,
  eventParList          ПОСЛЕДОВАТЕЛЬНОСТЬ OF EventParameter,
  timeNotation          TimeNotation ФАКУЛЬТАТИВНО,
  ...
}

EventName ::= PkgdName

EventParameter ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
  eventParameterName    Name,
  value                  Value,
  -- Для использования extraInfo см. комментарий относительно PropertyParm
  extraInfo             ВЫБОР
  {
    relation             Зависимость,
    range                БУЛЕВЫЙ,
    sublist              БУЛЕВЫЙ
  } ФАКУЛЬТАТИВНО,
  ...
}

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

ServiceChangeReply ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{

```

```

    terminationID          TerminationIDList,
    serviceChangeResult    ServiceChangeResult,
    ...
}

-- Для ServiceChangeResult, нет обязательных аргументов. Отсюда
-- различие между ServiceChangeParm и ServiceChangeResParm.

ServiceChangeResult ::= ВЫБОР
{
    errorDescriptor        ErrorDescriptor,
    serviceChangeResParms  ServiceChangeResParm
}

WildcardField ::= ОКТЕТНАЯ СТРОКА(РАЗМЕР(1))

TerminationID ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    wildcard ПОСЛЕДОВАТЕЛЬНОСТЬ OF WildcardField,
    id       ОКТЕТНАЯ СТРОКА(РАЗМЕР (1..8)),
    ...
}

-- См. А.1 Для пояснений по использованию механизма групповых символов замены.
-- Termination ID 0xFFFFFFFFFFFFFFFF указывает на окончание типа ROOT.

TerminationIDList ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF TerminationID

MediaDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    termStateDescrTerminationStateDescriptor ФАКУЛЬТАТИВНО,
    streams      ВЫБОР
    {
        oneStream StreamParms,
        multiStream ПОСЛЕДОВАТЕЛЬНОСТЬ OF StreamDescriptor
    } ФАКУЛЬТАТИВНО,
    ...
}

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

StreamParms ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    localControlDescriptor LocalControlDescriptor ФАКУЛЬТАТИВНО,
    localDescriptor        LocalRemoteDescriptor ФАКУЛЬТАТИВНО,
    remoteDescriptor       LocalRemoteDescriptor ФАКУЛЬТАТИВНО,
    ...
}

LocalControlDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    streamMode      StreamMode ФАКУЛЬТАТИВНО,
    reserveValue    БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    reserveGroup    БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    propertyParms   ПОСЛЕДОВАТЕЛЬНОСТЬ OF PropertyParm,
    ...
}

StreamMode ::= ПЕРЕЧИСЛИМЫЙ
{

```

```

    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
    ...
}

```

-- В PropertyParm, значение - это ПОСЛЕДОВАТЕЛЬНОСТЬ октетной строки. При отправке контроллером MGC трактовка следующая:

- пустая последовательность означает ВЫБИРАТЬ
- последовательность из одного элемента определяет значение
- Если поле подписка не выбрано, более длинная последовательность означает "выбери одно из значений" (т.е. значение1 ИЛИ значение2 ИЛИ ...)
- Если поле подписка выбрано, последовательность из более, чем одного элемента кодирует значение нормированного в виде списка свойства (т.е. значение1 И значение2 И ...).
- Поле отношения может выбираться только, если последовательность значений имеет длину 1. Это показывает, что шлюз MG должен выбрать значение для свойства. Например, $x > 3$ (использование значения больше, чем для зависимости) отдаёт распоряжение шлюзу MG выбрать для свойства x любое значение большее, чем 3.
- Поле диапазона может выбираться только, если последовательность значений имеет длину 2. Это показывает, что шлюз MG должен выбрать значение в диапазоне между первым октетом в последовательности значений и хвостовым октетом в последовательности значений, включающей граничные значения.
- При отправке шлюзом MG, только отклики на запрос AuditCapability могут содержать несколько значений, диапазон, или поле зависимости.

```

PropertyParm ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    name          PkgdName,
    value         ПОСЛЕДОВАТЕЛЬНОСТЬ OF ОКТЕТ STRING,
    extraInfo     ВЫБОР
    {
        relation   Зависимость,
        range      БУЛЕВЫЙ,
        sublist    БУЛЕВЫЙ
    } ФАКУЛЬТАТИВНО,
    ...
}

```

Name ::= ОКТЕТНАЯ СТРОКА (РАЗМЕР (2))

PkgdName ::= ОКТЕТНАЯ СТРОКА (РАЗМЕР (4))

- представляет имя комплекта (2 октета), включая имена свойства, события, сигнала, или идентификатор Statistics ID. (2 октета)
- Для применения групповых символов замены в комплекте используй 0xFFFF в первых двух октетах, choose не разрешается. Для ссылки на исходный тег свойства, определённый в Приложении C, используй 0x0000 в первых двух октетах.
- Для применения групповых символов замены в свойстве, событии, сигнале, или в идентификаторе Statistics ID, используй 0xFFFF в последних двух октетах, choose не разрешается.
- Применение групповых символов замены в имени комплекта разрешается только, если в свойстве, событии, сигнале, или идентификаторе Statistics ID также применяются групповые символы замены.

Relation ::= ПЕРЕЧИСЛИМЫЙ

```

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

```

```

}

LocalRemoteDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    propGrps ПОСЛЕДОВАТЕЛЬНОСТЬ OF PropertyGroup,
    ...
}

PropertyGroup ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF PropertyParm

TerminationStateDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    propertyParms        ПОСЛЕДОВАТЕЛЬНОСТЬ OF PropertyParm,
    eventBufferControl   EventBufferControl ФАКУЛЬТАТИВНО,
    serviceState         ServiceState ФАКУЛЬТАТИВНО,
    ...
}

EventBufferControl ::= ПЕРЕЧИСЛИМЫЙ
{
    off(0),
    lockStep(1),
    ...
}

ServiceState ::= ПЕРЕЧИСЛИМЫЙ
{
    test(0),
    outOfSvc(1),
    inSvc(2),
    ...
}

MuxDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    muxType              MuxType,
    termList             ПОСЛЕДОВАТЕЛЬНОСТЬ OF TerminationID,
    nonStandardData     NonStandardData ФАКУЛЬТАТИВНО,
    ...
}

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

StreamID ::= ЦЕЛОЕ ЧИСЛО(0..65535)    -- 16-разрядное целое число без знака

EventsDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    requestID           RequestID ФАКУЛЬТАТИВНО,
                        -- RequestID должен присутствовать, если eventList
                        -- не является ПУСТЫМ
    eventList           ПОСЛЕДОВАТЕЛЬНОСТЬ OF RequestedEvent,
    ...
}

RequestedEvent ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    pkgdName            PkgdName,
    streamID            StreamID ФАКУЛЬТАТИВНО,
}

```

```

    eventAction          RequestedActions ФАКУЛЬТАТИВНО,
    evParList            ПОСЛЕДОВАТЕЛЬНОСТЬ OF EventParameter,
    ...
}

RequestedActions ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    keepActive          БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    eventDM             EventDM ФАКУЛЬТАТИВНО,
    secondEvent        SecondEventsDescriptor ФАКУЛЬТАТИВНО,
    signalsDescriptor   SignalsDescriptor ФАКУЛЬТАТИВНО,
    ...
}

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

SecondEventsDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    requestID          RequestID ФАКУЛЬТАТИВНО,
    eventList          ПОСЛЕДОВАТЕЛЬНОСТЬ OF SecondRequestedEvent,
    ...
}

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

SecondRequestedActions ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    keepActive          БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    eventDM             EventDM ФАКУЛЬТАТИВНО,
    signalsDescriptor   SignalsDescriptor ФАКУЛЬТАТИВНО,
    ...
}

EventBufferDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF EventSpec

EventSpec ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    eventName          EventName,
    streamID           StreamID ФАКУЛЬТАТИВНО,
    eventParList       ПОСЛЕДОВАТЕЛЬНОСТЬ OF EventParameter,
    ...
}

SignalsDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF SignalRequest

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

SeqSigList ::= ПОСЛЕДОВАТЕЛЬНОСТЬ

```

```

{
    id          ЦЕЛОЕ ЧИСЛО(0..65535),
    sigallList  ПОСЛЕДОВАТЕЛЬНОСТЬ OF Signal
}

Signal ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    signalName      SignalName,
    streamID        StreamID ФАКУЛЬТАТИВНО,
    sigType         SignalType ФАКУЛЬТАТИВНО,
    duration        ЦЕЛОЕ ЧИСЛО (0..65535) ФАКУЛЬТАТИВНО,
    notifyCompletion NotifyCompletion ФАКУЛЬТАТИВНО,
    keepActive      БУЛЕВЫЙ ФАКУЛЬТАТИВНО,
    sigParList      ПОСЛЕДОВАТЕЛЬНОСТЬ OF SigParameter,
    ...
}

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

SignalName ::= PkgdName

NotifyCompletion ::= БИТОВАЯ СТРОКА
{
    onTimeOut(0), onInterruptByEvent(1),
    onInterruptByNewSignalDescr(2), otherReason(3)
}

SigParameter ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    sigParameterName  Name,
    value              Value,
    -- В отношении использования extraInfo см. комментарий относительно
    -- PropertyParm
    extraInfo CHOICE
    {
        relation      Зависимость,
        range          БУЛЕВЫЙ,
        sublist        БУЛЕВЫЙ
    } ФАКУЛЬТАТИВНО,
    ...
}

-- Для AuditCapReply со всеми событиями RequestID ДОЛЖЕН быть ALL.
-- ALL представлен посредством 0xffffffff.

RequestID ::= ЦЕЛОЕ ЧИСЛО(0..4294967295)  -- 32-разрядное целое число без знака

ModemDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    mtl ПОСЛЕДОВАТЕЛЬНОСТЬ OF ModemType,
    mpl ПОСЛЕДОВАТЕЛЬНОСТЬ OF PropertyParm,
    nonStandardData  NonStandardData ФАКУЛЬТАТИВНО
}

```

```

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

DigitMapDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    digitMapName    DigitMapName    ФАКУЛЬТАТИВНО,
    digitMapValue   DigitMapValue   ФАКУЛЬТАТИВНО
}

DigitMapName ::= Имя

DigitMapValue ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    startTimer      ЦЕЛОЕ ЧИСЛО(0..99) ФАКУЛЬТАТИВНО,
    shortTimer      ЦЕЛОЕ ЧИСЛО(0..99) ФАКУЛЬТАТИВНО,
    longTimer       ЦЕЛОЕ ЧИСЛО(0..99) ФАКУЛЬТАТИВНО,
    digitMapBody    IA5String,
    -- Для стартового, короткого и длинного таймеров единицами измерения
    -- являются секунды, а для таймера продолжительности – сотни миллисекунд.
    -- Таким образом, диапазон стартового, короткого и длинного таймеров от 1
    -- до 99 секунд, а таймера продолжительности от 100 мсек до 9.9 сек
    -- См. А.3 для пояснений синтаксиса плана нумерации
    ...
}

ServiceChangeParm ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    serviceChangeMethod    ServiceChangeMethod,
    serviceChangeAddress   ServiceChangeAddress ФАКУЛЬТАТИВНО,
    serviceChangeVersion   ЦЕЛОЕ ЧИСЛО(0..99) ФАКУЛЬТАТИВНО,
    serviceChangeProfile   ServiceChangeProfile ФАКУЛЬТАТИВНО,
    serviceChangeReason    Значение,
    -- serviceChangeReason состоит из числового кода причины
    -- и факультативного текстового описания.
    -- serviceChangeReason ДОЛЖЕН быть строкой, состоящей из
    -- десятичного кода причины, за которым факультативно следует
    -- один символ пробела и строка текстового описания.
    -- Эта строка является первой BER-кодированной, как IA5String.
    -- Результат этого BER-кодирования затем кодируется как
    -- ОКТЕТНАЯ СТРОКА типа ASN.1, "двойное заворачивание"
    -- значения как было сделано для элементов комплекта.
    serviceChangeDelay     ЦЕЛОЕ ЧИСЛО(0..4294967295) ФАКУЛЬТАТИВНО,
    -- 32-разрядное целое число без знака
    serviceChangeMgcId     MId ФАКУЛЬТАТИВНО,
    timeStamp              TimeNotation ФАКУЛЬТАТИВНО,
    nonStandardData        NonStandardData ФАКУЛЬТАТИВНО,
    ...
}

```

```

ServiceChangeAddress ::= ВЫБОР
{
    portNumber          ЦЕЛОЕ ЧИСЛО(0..65535),      -- номер порта TCP/UDP
    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName          DomainName,
    deviceName          PathName,
    mtpAddress          OCTET STRING(SIZE(2)),
    ...
}

ServiceChangeResParm ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    serviceChangeMgcId  MId ФАКУЛЬТАТИВНО,
    serviceChangeAddress ServiceChangeAddress ФАКУЛЬТАТИВНО,
    serviceChangeVersion ЦЕЛОЕ ЧИСЛО(0..99) ФАКУЛЬТАТИВНО,
    serviceChangeProfile ServiceChangeProfile ФАКУЛЬТАТИВНО,
    timestamp           TimeNotation ФАКУЛЬТАТИВНО,
    ...
}

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

ServiceChangeProfile ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    profileName          IA5String(РАЗМЕР (1..67))

    -- 64 символа для имени, 1 для "/", 2 для версии, чтобы отвечать
    -- требованиям ABNF
}

PackagesDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF PackagesItem

PackagesItem ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    packageName          Имя,
    packageVersion       ЦЕЛОЕ ЧИСЛО(0..99),
    ...
}

StatisticsDescriptor ::= ПОСЛЕДОВАТЕЛЬНОСТЬ OF StatisticsParameter

StatisticsParameter ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    statName             PkgdName,
    statValue            Значение ФАКУЛЬТАТИВНО
}

NonStandardData ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    nonStandardIdentifier NonStandardIdentifier,
    data                 ОКТЕТНАЯ СТРОКА
}

```

```

NonStandardIdentifier ::= ВЫБОР
{
    object                ИДЕНТИФИКАТОР ОБЪЕКТА,
    h221NonStandard       H221NonStandard,
    experimental          IA5String(РАЗМЕР(8)),
    -- первыми двумя символами должны быть "X-" или "X+"
    ...
}

H221NonStandard ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    t35CountryCode1       ЦЕЛОЕ ЧИСЛО(0..255),
    t35CountryCode2       ЦЕЛОЕ ЧИСЛО(0..255),           -- страна, согласно Т.35
    t35Extension          ЦЕЛОЕ ЧИСЛО(0..255),           -- присвоенная
                                                                -- национальность
    manufacturerCode      ЦЕЛОЕ ЧИСЛО(0..65535),         -- присвоенная
                                                                -- национальность
    ...
}

TimeNotation ::= ПОСЛЕДОВАТЕЛЬНОСТЬ
{
    date                  IA5String(SIZE(8)),           -- формат yyyymmdd
    time                  IA5String(SIZE(8)),           -- формат hhmmssss
                                                                -- согласно ISO 8601:1988
}

Value ::= ПОСЛЕДОВАТЕЛЬНОСТЬ ОКТЕТНОЙ СТРОКИ

```

END

А.3 Планы нумерации и имена маршрутов

С синтаксической точки зрения планы нумерации являются строками с наложенными на них синтаксическими ограничениями. Синтаксис правильных планов нумерации определяется в ABNF (IETF RFC 2234). Синтаксис планов нумерации, представленный в данном разделе предназначен только для целей иллюстрации. Определение digitMap, данное в Приложении В, имеет приоритет в случае расхождений между двумя документами, указанными выше.

```

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 ;цифры 0-9
                  / %x41-4B / %x61-6B ;a-k и A-K
                  / "L" / "S" ;Межсобытийные таймеры
                                                                ;(длинный, короткий)
                  / "Z" ;Длительное событие
DOT = %x2E ; "."
LWSP = *(WSP / COMMENT / EOL)
WSP = SP / HTAB
COMMENT = ";" *(SafeChar / RestChar / WSP) EOL
EOL = (CR [LF]) / LF
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
SafeChar = DIGIT / ALPHA / "+" / "-" / "&" / "!" / "_" / "/" /

```

```

"!" / "?" / "@" / "^" / "`" / "~" / "*" / "$" / "\" /
"(" / ")" / "%" / "."
RestChar = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
"<" / ">" / "=" / %x22
DIGIT = %x30-39 ; цифры от 0 до 9
ALPHA = %x41-5A / %x61-7A ; A-Z, a-z

```

Имя маршрута также является строкой с наложенными на неё синтаксическими ограничениями. Определяющее её порождение ABNF скопировано из Приложения В.

```

; Общая длина pathNAME не должна превышать 64 симв.
pathNAME = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$" )
["@" pathDomainName ]
; ABNF допускает два или более последовательных символа "." хотя это не имеет
; смысла в имени домена маршрута.
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 протокола, несмотря на то, что идентификаторы TerminationID являются произвольно выбранными путём разумного выбора имён, символ групповой замены "*" может быть сделан более полезным. Когда встречается символ групповой замены, он будет "подходить" всем идентификаторам TerminationID, имеющим одинаковые предыдущие и последующие символы (если это свойственно). Например, если имеются идентификаторы TerminationIDs R13/3/1, R13/3/2 and R13/3/3, то идентификатор TerminationID R13/3/* будет соответствовать всем из них. Имеются некоторые обстоятельства, когда ссылки должны быть сделаны на ВСЕ окончания. Идентификатор TerminationID "*" удовлетворяет этому требованию и ссылается на ВСЕ. Идентификатор TerminationID "\$" типа CHOOSE может использоваться для того, чтобы просигнализировать шлюзу MG, что он должен создать эфемерное окончание или выбрать свободное физическое окончание.

В.2 Спецификация ABNF

Синтаксис протокола представлен в ABNF в соответствии с IETF RFC 2234.

ПРИМЕЧАНИЕ 1 – Эта спецификация синтаксиса не накладывает каких либо ограничений на присоединения элементов и значения. Некоторые дополнительные ограничения указываются в комментариях, а другие ограничения приведены в тексте данной Рекомендации. Эти дополнительные ограничения являются частью протокола, хотя не обязываются данной Рекомендацией.

ПРИМЕЧАНИЕ 2 – Синтаксис является контекстно-зависимым. Например, "Add" может быть AddToken или NAME в зависимости от контекста, в котором встречается.

Всё в ABNF и текстовом кодировании не зависит от регистра. Это включает TerminationIDs, digitmap Ids и т.д. Согласно IETF RFC 2327 протокол SDP является зависимым от регистра.

```

; ПРИМЕЧАНИЕ – ABNF в данном разделе использует конструкцию VALUE (или списки
; конструкций VALUE) для кодирования различных значений элементов комплектов
; (свойств, параметров сигналов и т.д.). Типы этих значений разнообразны и
; определяются в соответствующих определениях комплекта. Несколько таких типов
; описывается в 12.2.
;

```

```

; Спецификация ABNF для VALUE допускает форму quotedString или
; совокупность SafeChars. Кодирование значений элементов комплекта в
; ABNF VALUES определяется ниже. Если кодирование типа допускает символы
; иные, чем SafeChars, форма quotedString ДОЛЖНА использоваться для всех
; значений этого типа, даже для особых значений, которые состоят только из
; SafeChars.
;
; Строка: Строка ДОЛЖНА использовать форму quotedString конструкции VALUE и
; может содержать всё, что допускается в форме quotedString.
;
; Целое число, двойное число и целое число без знака: десятичные значения могут
; закодированы, используя символы 0-9. Шестнадцатеричные значения должны иметь
; префикс '0x' и могут использовать символы 0-9, a-f, A-F. Октетный формат не
; поддерживается.
; Отрицательные целые числа начинаются с '-' и ДОЛЖНЫ быть десятичными. ДОЛЖНА
; использоваться форма SafeChar конструкции VALUE.
;
; Символ: Кодирование UTF-8 одной буквы заключается в двойные кавычки.
;
; Перечисление: Перечисление ДОЛЖНО использовать форму SafeChar конструкции
; VALUE и может содержать всё, что допускается в форме SafeChar.
;
; Булевый: Булевы значения кодируются как "on" и "off" и
; зависят от регистра. ДОЛЖНА использоваться форма SafeChar конструкции VALUE.
;
; Будущие типы: Любые определённые типы ДОЛЖНЫ соответствовать спецификации
; ABNF конструкции VALUE. В особенности, если кодирования типа
; допускает символы иные, чем SafeChars, ДОЛЖНА использоваться форма
; quotedString для всех значений этого типа, даже для особых значений, которые
; состоят только из SafeChars.
;
; Заметим, что нельзя использовать символ двойные кавычки в пределах
; значения.
;
; Заметим, что SDP не допускает пробел в начале строки, Megaco
; ABNF допускает пробел перед началом SDP в дескрипторе
; Local/Remote. Синтаксические анализаторы должны принять пробел между
; LBRKT, следующим за лексемой Local/Remote token и началом SDP.

```

```
megacoMessage      = LWSP [authenticationHeader SEP ] сообщение
```

```
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, которая здесь определяется, равна 1.
```

```
messageBody        = ( errorDescriptor / transactionList )
```

```
transactionList    = 1*( transactionRequest / transactionReply /
                      transactionPending / transactionResponseAck )
;Использование подтверждений откликов зависит от базового транспорта
```

```
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)

; at-most-once
contextProperty    = (topologyDescriptor / priority / EmergencyToken)

contextAudit       = ContextAuditToken LBRKT
                    contextAuditProperties *(COMMA
                    contextAuditProperties) RBRKT

; at-most-once
contextAuditProperties = ( TopologyToken / EmergencyToken /
                        PriorityToken )

; "O-" указывает факультативную команду
; "W-" указывает отклик на команду с использованием группового символа замены
commandRequestList= ["O-"] ["W-"] commandRequest *
                    (COMMA ["O-"] ["W-"] commandRequest)

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

transactionReply   = ReplyToken EQUAL TransactionID LBRKT
                    [ ImmAckRequiredToken COMMA]
                    ( errorDescriptor / actionReplyList ) RBRKT

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 and Modify have the same request parameters
ammRequest         = (AddToken / MoveToken / ModifyToken ) EQUAL
                    TerminationID [LBRKT ammParameter *(COMMA
                    ammParameter) RBRKT]

;at-most-once
ammParameter       = (mediaDescriptor / modemDescriptor /
                    muxDescriptor / eventsDescriptor /
                    signalsDescriptor / digitMapDescriptor /
                    eventBufferDescriptor / auditDescriptor)

ammsReply          = (AddToken / MoveToken / ModifyToken /
                    SubtractToken ) EQUAL TerminationID [ LBRKT

```

```

        terminationAudit RBRKT ]

subtractRequest = SubtractToken EQUAL TerminationID
                [ LBRKT auditDescriptor RBRKT]

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

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

auditOther      = EQUAL TerminationID [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 / auditItem)

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

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

notifyReply     = NotifyToken EQUAL TerminationID
                [ LBRKT errorDescriptor RBRKT ]

serviceChangeRequest = ServiceChangeToken EQUAL TerminationID
                LBRKT serviceChangeDescriptor RBRKT

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

errorDescriptor = ErrorToken EQUAL ErrorCode
                LBRKT [quotedString] RBRKT

ErrorCode       = 1*4(ЦИФРА) ; может быть расширен

TransactionID   = UINT32

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

; ABNF допускает два или более последовательных "." , хотя это не имеет
; значения в имени домена.
domainName      = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" /
                ".") ">"

deviceName      = pathNAME

;Значения 0x0, 0xFFFFFFFFE and 0xFFFFFFFF являются зарезервированными.
ContextID       = (UINT32 / "*" / "-" / "$")

domainAddress   = "[" (IPv4address / IPv6address) "]"
;RFC2373 contains the definition of IP6Addresses.

```

```

IPv6address      = hexpart [ ":" IPv4address ]
IPv4address      = V4hex DOT V4hex DOT V4hex DOT V4hex
V4hex            = 1*3(DIGIT) ; "0".."255"
; this production, while occurring in RFC 2373, is not referenced
; 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 вит   2 бит
; ПРИМЕЧАНИЕ - 14 бит определены для международного использования.
; Существуют две национальные опции с кодом точки, равным 16 или 24 бит.
; Для октетного выравнивания the mtpAddress коды старших разрядов должны
; быть равны 0.
; Октет должен представляться 2 шестнадцатеричными цифрами.
mtpAddress       = MTPToken LBRKT 4*8 (HEXDIG) RBRKT

terminationIDList = LBRKT TerminationID *(COMMA TerminationID) RBRKT

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

; ABNF допускает два и более последовательных ".", хотя это не имеет
; смысла в имени домена маршрута.
pathDomainName   = (ALPHA / DIGIT / "*" )
                 *63(ALPHA / DIGIT / "-" / "*" / ".")

TerminationID    = "ROOT" / pathNAME / "$" / "*"

mediaDescriptor  = MediaToken LBRKT mediaParm *(COMMA mediaParm) RBRKT

; по большей части один terminationStateDescriptor
; и либо streamParm(s), или streamDescriptor(s), но не оба
mediaParm         = (streamParm / streamDescriptor /
                    terminationStateDescriptor)

; по большей части один раз на пункт
streamParm        = ( localDescriptor / remoteDescriptor /
                    localControlDescriptor )

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

localControlDescriptor = LocalControlToken LBRKT localParm
                       *(COMMA localParm) RBRKT

; по большей части один раз на пункт, за исключением для 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
parmValue         = (EQUAL alternativeValue/ INEQUAL VALUE)
alternativeValue  = ( VALUE
                    / LSBRKT VALUE *(COMMA VALUE) RBRKT
                    ; sublist (i.e. A AND B AND ...)
                    / LBRKT VALUE *(COMMA VALUE) RBRKT
                    ; alternatives (i.e. A OR B OR ...)

                    / LSBRKT VALUE COLON VALUE RBRKT )
                    ; range

```

```

INEQUAL          = LWSP ">" / "<" / "#" ) LWSP
LSBRKT           = LWSP "[" LWSP
RSBRKT           = LWSP "]" LWSP

```

; ПРИМЕЧАНИЕ – Нулевой октет не находится среди допустимых символов октетной строки. Т.к. текущее определение ограничивается SDP, а нулевой октет не является допустимым символом в SDP, это не вызывает беспокойства.

```

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 ( "OFF" / LockStepToken )

```

```

terminationStateDescriptor = TerminationStateToken LBRKT
                             terminationStateParm *( COMMA terminationStateParm ) RBRKT

```

; по большей части один раз на пункт, за исключением для propertyParm
terminationStateParm =(propertyParm / serviceStates / eventBufferControl)

```

serviceStates    = ServiceStatesToken EQUAL ( TestToken /
                                              OutOfSvcToken / InSvcToken )

```

```

muxDescriptor     = MuxToken EQUAL MuxType  terminationIDList

```

```

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

```

```

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 ]

```

;по большей части один раз каждый из KeepActiveToken , eventDM и eventStream
;по большей части один из, либо embedWithSig, или embedNoSig, но не оба
;KeepActiveToken и embedWithSig не должны присутствовать оба
eventParameter = (embedWithSig / embedNoSig / KeepActiveToken

```

        /eventDM / eventStream / eventOther )

embedWithSig      = EmbedToken LBRKT signalsDescriptor
                   [COMMA embedFirst ] RBRKT
embedNoSig        = EmbedToken LBRKT embedFirst RBRKT

; по большей части один раз каждый из
embedFirst        = EventsToken [ EQUAL RequestID LBRKT
                               secondRequestedEvent *(COMMA secondRequestedEvent) RBRKT ]

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

; по большей части один раз каждый из embedSig , KeepActiveToken, eventDM или
; eventStream
; KeepActiveToken и embedSig не должны присутствовать оба
secondEventParameter = ( embedSig / KeepActiveToken / eventDM /
                        eventStream / eventOther )

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

;ровно один раз signalType, по большей части один раз в течение длительности и
;аргумента каждого сигнала
signalListParm   = signalRequest

signalName       = pkgdName
;по большей части один раз sigStream, по большей части один раз sigSignalType,
;по большей части один раз sigDuration, каждый signalParameterName по большей
;части один раз
sigParameter     = sigStream / sigSignalType / sigDuration / sigOther
                 / notifyCompletion / KeepActiveToken

sigStream        = StreamToken EQUAL StreamID
sigOther         = sigParameterName parmValue
sigParameterName = NAME
sigSignalType    = SignalTypeToken EQUAL signalType
signalType       = ( OnOffToken / TimeOutToken / BriefToken)
sigDuration      = DurationToken EQUAL UINT16
notifyCompletion = NotifyCompletionToken EQUAL (LBRKT
        notificationReason *(COMMA notificationReason) RBRKT)

notificationReason = ( TimeOutToken / InterruptByEventToken
                      / InterruptByNewSignalsDescrToken

```

```

        / OtherReasonToken )
observedEventsDescriptor = ObservedEventsToken EQUAL RequestID
        LBRKT observedEvent *(COMMA observedEvent) RBRKT

; время на случай, потому что оно может быть буферизовано
observedEvent
    = [ TimeStamp LWSP COLON] LWSP
      pkgdName [ LBRKT observedEventParameter
                *(COMMA observedEventParameter) RBRKT ]

; по большей части один раз eventStream, каждый eventParameterName по большей
; части один раз
observedEventParameter = eventStream / eventOther

; Для AuditCapReply со всеми событиями, идентификатор RequestID должен быть ALL.
RequestID
    = ( UINT32 / "*" )

modemDescriptor
    = ModemToken (( EQUAL modemType) /
                  (LSBRKT modemType *(COMMA modemType) RSBKRKT))
      [ LBRKT propertyParm
        *(COMMA propertyParm) RBRKT ]

; по большей части один раз за исключением 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] digitMap
Timer
    = 1*2DIGIT
; Единицы измерения для таймеров T, S, и L - секунды, а для таймера Z - сотни
; миллисекунд. Таким образом, диапазон T, S, и L таймеров от 1 до 99
; секунд, а таймера Z - от 100 мсек до 9.9 сек
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" ;Inter-event timers (long, short)
      / "Z" ;Long duration modifier

; по большей части один раз, а DigitMapToken и PackagesToken не допускаются
; в команду AuditCapabilities
auditItem
    = ( MuxToken / ModemToken / MediaToken /
        SignalsToken / EventBufferToken /
        DigitMapToken / StatsToken / EventsToken /
        ObservedEventsToken / PackagesToken )

serviceChangeDescriptor = ServicesToken LBRKT serviceChangeParm
                        *(COMMA serviceChangeParm) RBRKT

; каждый аргумент по большей части один раз
; по большей части один из либо serviceChangeAddress, или serviceChangeMgcId, но
; не не оба
; serviceChangeMethod и serviceChangeReason ТРЕБУЮТСЯ
serviceChangeParm
    = (serviceChangeMethod / serviceChangeReason /
        serviceChangeDelay / serviceChangeAddress /

```

```

        serviceChangeProfile / extension / TimeStamp /
        serviceChangeMgcId / serviceChangeVersion )

serviceChangeReplyDescriptor = ServicesToken LBRKT
        servChgReplyParm *(COMMA servChgReplyParm) RBRKT

; по большей части один раз. ТРЕБУЕТСЯ версия на первом ServiceChange отклике
; по большей части один из either serviceChangeAddress или serviceChangeMgcId,
; но не оба
servChgReplyParm      = (serviceChangeAddress / serviceChangeMgcId /
        serviceChangeProfile / serviceChangeVersion /
        TimeStamp)
serviceChangeMethod = MethodToken EQUAL (FailoverToken /
        ForcedToken / GracefulToken / RestartToken /
        DisconnectedToken / HandOffToken /
        extensionParameter)

; serviceChangeReason состоит из числового кода причины
; и факультативного текстового описания.
; serviceChangeReason ДОЛЖЕН быть закодирован, используя форму quotedString
; конструкции VALUE.
; quotedString ДОЛЖЕН содержать десятичный код причины,
; за которым факультативно 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 8601:1988
; Дата = yyyymmdd
Date               = 8(DIGIT)
; Время = hhmmssss
Time              = 8(DIGIT)
statisticsDescriptor = StatsToken LBRKT statisticsParameter
        *(COMMA statisticsParameter ) RBRKT

; по большей части один раз на параграф
statisticsParameter = pkgdName [EQUAL VALUE]

topologyDescriptor = TopologyToken LBRKT topologyTriple
        *(COMMA topologyTriple) RBRKT
topologyTriple     = terminationA COMMA
        terminationB COMMA topologyDirection
terminationA       = TerminationID
terminationB       = TerminationID
topologyDirection  = BothwayToken / IsolateToken / OnewayToken

priority           = PriorityToken EQUAL UINT16

extensionParameter = "X" ("-" / "+") 1*6(ALPHA / DIGIT)

; octetString используется, чтобы описать SDP, определённый в RFC 2327.

```

```

; Осторожность должна соблюдаться, если используется CRLF в RFC 2327.
; Для безопасности, используйте EOL в данном ABNF.
; Всякий раз, когда "}" появляется в SDP, он отключается посредством "\",
; т.е. "\"
octetString      = *(nonEscapeChar)
nonEscapeChar   = ( "\" / %x01-7C / %x7E-FF )
; ПРИМЕЧАНИЕ - Символ двойные кавычки не допускается в quotedString.
quotedString    = DQUOTE *(SafeChar / RestChar/ WSP) DQUOTE

```

```

UINT16          = 1*5(DIGIT) ; %x0-FFFF
UINT32          = 1*10(DIGIT) ; %x0-FFFFFFFF

```

```

NAME            = ALPHA *63(ALPHA / DIGIT / "_" )
VALUE           = quotedString / 1*(SafeChar)
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 ; " (Double Quote)
HEXDIG        = ( DIGIT / "A" / "B" / "C" / "D" / "E" / "F" )
SP            = %x20 ; пробел
HTAB          = %x09 ; горизонтальный таб
CR            = %x0D ; возврат каретки
LF            = %x0A ; перевод строки
LWSP          = *( WSP / COMMENT / EOL )
EOL           = ( CR [LF] / LF )
WSP           = SP / HTAB ; пробел
SEP           = ( WSP / EOL / COMMENT) LWSP
COMMENT       = ";" *(SafeChar/ RestChar/ WSP / %x22) EOL
RestChar      = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
                 "<" / ">" / "="

```

```

; Новые лексемы, присоединённые к sigParameter должны принимать SPA* формат
; * может быть любой формы, т.е. SPAM
; Новые лексемы, присоединённые eventParameter должны принимать EPA* формат
; * может быть любой формы, т.е. EPAD

```

```

AddToken       = ("Add" / "A")
AuditToken     = ("Audit" / "AT")
AuditCapToken  = ("AuditCapability" / "AC")
AuditValueToken = ("AuditValue" / "AV")
AuthToken      = ("Authentication" / "AU")
BothwayToken   = ("Bothway" / "BW")
BriefToken     = ("Brief" / "BR")
BufferToken    = ("Buffer" / "BF")
CtxToken       = ("Context" / "C")
ContextAuditToken = ("ContextAudit" / "CA")
DigitMapToken  = ("DigitMap" / "DM")
DisconnectedToken = ("Disconnected" / "DC")
DelayToken     = ("Delay" / "DL")
DurationToken  = ("Duration" / "DR")
EmbedToken     = ("Embed" / "EM")
EmergencyToken = ("Emergency" / "EG")
ErrorToken     = ("Error" / "ER")

```

EventBufferToken	= ("EventBuffer"	/ "EB")
EventsToken	= ("Events"	/ "E")
FailoverToken	= ("Failover"	/ "FL")
ForcedToken	= ("Forced"	/ "FO")
GracefulToken	= ("Graceful"	/ "GR")
H221Token	= ("H221")
H223Token	= ("H223")
H226Token	= ("H226")
HandOffToken	= ("HandOff"	/ "HO")
ImmAckRequiredToken	= ("ImmAckRequired"	/ "IA")
InactiveToken	= ("Inactive"	/ "IN")
IsolateToken	= ("Isolate"	/ "IS")
InSvcToken	= ("InService"	/ "IV")
InterruptByEventToken	= ("IntByEvent"	/ "IBE")
InterruptByNewSignalsDescrToken	= ("IntBySigDescr"	/ "IBS")
KeepActiveToken	= ("KeepActive"	/ "KA")
LocalToken	= ("Local"	/ "L")
LocalControlToken	= ("LocalControl"	/ "O")
LockStepToken	= ("LockStep"	/ "SP")
LoopbackToken	= ("Loopback"	/ "LB")
MediaToken	= ("Media"	/ "M")
MegacopToken	= ("MEGACO"	/ "!"
MethodToken	= ("Method"	/ "MT")
MgcIdToken	= ("MgcIdToTry"	/ "MG")
ModeToken	= ("Mode"	/ "MO")
ModifyToken	= ("Modify"	/ "MF")
ModemToken	= ("Modem"	/ "MD")
MoveToken	= ("Move"	/ "MV")
MTPToken	= ("MTP")	
MuxToken	= ("Mux"	/ "MX")
NotifyToken	= ("Notify"	/ "N")
NotifyCompletionToken	= ("NotifyCompletion"	/ "NC")
ObservedEventsToken	= ("ObservedEvents"	/ "OE")
OnewayToken	= ("Oneway"	/ "OW")
OnOffToken	= ("OnOff"	/ "OO")
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")
RestartToken	= ("Restart"	/ "RS")
RemoteToken	= ("Remote"	/ "R")
ReservedGroupToken	= ("ReservedGroup"	/ "RG")
ReservedValueToken	= ("ReservedValue"	/ "RV")
SendonlyToken	= ("SendOnly"	/ "SO")
SendrecvToken	= ("SendReceive"	/ "SR")
ServicesToken	= ("Services"	/ "SV")
ServiceStatesToken	= ("ServiceStates"	/ "SI")
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 Кодирование шестнадцатеричного октета

Октетное шестнадцатеричное кодирование является средством представления строки октетов в виде шестнадцатеричных цифр, с двумя цифрами, представляющими каждый октет. Данное октетное кодирование должно применяться для кодирования октетных строк в текстовой версии протокола.

Для каждого октета 8-битовая последовательность кодируется двумя шестнадцатеричными цифрами. Бит 0 передаётся первым; бит 7 передаётся последним.

Биты 7-4 кодируются как первая шестнадцатеричная цифра, с битом 7, как MSB, и битом 4, как LSB. Биты 3-0 кодируются как вторая шестнадцатеричная цифра, с битом 3, как MSB, и битом 0, как LSB.

Примеры:

Образец двоичного октета	Шестнадцатеричное кодирование
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

В.4 Последовательность шестнадцатеричного октета

Последовательность шестнадцатеричного октета представляет собой чётное количество цифр, заканчивающихся символом <CR>.

Приложение С

Теги для свойств медиапотоков

Если используется двоичное кодирование протокола, аргументы дескрипторов Local, Remote и LocalControl определяются как пары тег-значений. Данное приложение содержит названия свойств (PropertyID), теги (Property tag), тип свойства (Type) и значения (Value). Значения, представленные в поле Value, должны рассматриваться как "информация", если поле содержит ссылки. Ссылка содержит нормативные значения. Если поле значения не содержит ссылку, тогда значения в таком поле могут рассматриваться, как "нормативные".

В данном приложении теги приводятся в виде шестнадцатеричных чисел. При установке значения свойства, контроллер MGC может недоопределить значение, в соответствии с одним из механизмов, определённых в 7.1.1.

В данном приложении или любом из его подразделов поддержка свойств является факультативной. Например, только три свойства из С.3 и только пять свойств из С.8 могут быть реализованы.

Для типа "перечисление" значение представляется значением в скобках, например, Send(0), Receive(1). Свойства типов "N bits" или "M Octets", приведенные в Приложении С, должны трактоваться при кодировании протокола, как октетные строки. Свойства типа "N битовое целое число" должны трактоваться, как целые числа. При кодировании протокола "строка" должна трактоваться, как IA5String.

Если тип меньше, чем один октет, значение должно храниться в младших разрядах октетной строки размера 1.

С.1 Общие атрибуты среды передачи

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
Media	1001	Перечислимый	Аудио(0), Видео(1), Данные(2)
Transmission mode	1002	Перечислимый	Послать(0), Принять(1), Послать и Принять(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 – Единицы измерения: 100 бит/с.
ACodec	1006	Октетная строка	Тип аудио кодека: Ссылка: Рек. МСЭ-Т Q.765 Кодеки, не удовлетворяющие Рекомендациям МСЭ-Т, определяются идентификаторами соответствующих организаций по стандартизации
Samplepp	1007	Целое число без знака	Максимальное количество отсчётов или кадров на комплект: 0..65535
Silencesupp	1008	Булевый	Подавление пауз: Правда/Ложь
Encrypttype	1009	Октетная строка	Ссылка: Рек.. МСЭ-Т Н.245
Encryptkey	100A	Октетная строка размером (0..65535)	Ключ шифрования Ссылка: Рек. МСЭ-Т Н.235
Echocanc	100B		Не используется. См. в 1 Е.13 пример возможных свойств Echo Control.
Gain	100C	Целое число без знака	Усиление в дБ: 0..65535
Jitterbuff	100D	Целое число без знака	Размер буфера джиттера в мсек: 0..65535
PropDelay	100E	Целое число без знака	Задержка на распространение: 0..65535

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
		знака	Максимальная задержка на распространение в миллисекундах для транспортного соединения между двумя медиашлюзами. Максимальная задержка будет зависеть от технологии транспорта.
RTPpayload	100F	Целое число	Тип нагрузки в Профиле RTP для аудио и видео конференций с минимальным управлением Ссылка: IETF RFC 1890

С.2 Свойства мультиплексора

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
H222	2001	Октетная строка	H222LogicalChannelParameters Ссылка: Рек. МСЭ-Т Н.245
H223	2002	Октетная строка	H223LogicalChannelParameters Ссылка: Рек. МСЭ-Т Н.245
V76	2003	Октетная строка	V76LogicalChannelParameters Ссылка: Рек. МСЭ-Т Н.245
H2250	2004	Октетная строка	H2250LogicalChannelParameters Ссылка: Рек. МСЭ-Т Н.245

С.3 Общие свойства транспортного канала

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
Mediatx	3001	Перечислимый	Тип медиатранспорта Канал с временным уплотнением(0), ATM(1), FR(2), Ipv4(3), Ipv6(4), ...
BIR	3002	4 октета	Значение зависит от технологии транспорта
NSAP	3003	1-20 октетов	См. NSAP. Ссылка: Приложение А/Х.213

С.4 Общие свойства ATM

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
AESA	4001	20 октета	Адрес конечной ATM-системы
VPVC	4002	4 октета: VPCI в двух первых младших октетах, VCI в двух вторых октетах	VPCI/VCI Ссылка: Рек. МСЭ-Т Q.2931

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
SC	4003	Перечислимый	Категория службы: 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). Ссылка: Форум ATM UNI 4.0
BCOV	4004	5-битовое целое число	Класс широкополного транспортного канала Ссылка: Рек. МСЭ-Т Q.2961.2
BBTC	4005	7- битовое целое число	Возможность широкополосной передачи Ссылка: Рек. МСЭ-Т Q.2961.1
ATC	4006	Перечислимый	Возможность трафика I.371 ATM DBR(0), SBR1(1), SBR2(2), SBR3(3), АВТ/IT(4), АВТ/DT(5), ABR(6) Ссылка: Рек. МСЭ-Т I.371
STC	4007	2 бита	Чувствительность к ограничению: Биты <u>2 1</u> 0 0 не чувствителен к ограничению 0 1 чувствителен к ограничению Ссылка: Рек. МСЭ-Т Q.2931
UPCC	4008	2 бита	Конфигурация плана соединений пользователя: Биты <u>2 1</u> 0 0 точка-точка 0 1 точка-много точек Ref.: Рек. МСЭ-Т Q.2931
PCR0	4009	24-битовое целое число	Максимальная скорость передачи ячеек (для CLP = 0) Ссылка: Рек. МСЭ-Т Q.2931
SCR0	400A	24-битовое целое число	Средняя скорость передачи ячеек (для CLP = 0) Ссылка: Рек. МСЭ-Т Q.2961.1
MBS0	400B	24-битовое целое число	Максимальный размер пачки (для CLP = 0) Ссылка: Рек. МСЭ-Т Q.2961.1
PCR1	400C	24-битовое целое число	Максимальная скорость передачи ячеек (для CLP = 0 + 1) Ссылка: Рек. МСЭ-Т Q.2931
SCR1	400D	24-битовое целое число	Средняя скорость передачи ячеек (для CLP = 0 + 1) Ссылка: Рек. МСЭ-Т Q.2961.1
MBS1	400E	24-битовое целое число	Максимальный размер пачки (для CLP = 0 + 1) Ссылка: Рек. МСЭ-Т Q.2961.1

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value	
BEI	400F	Булевый	Индикатор негарантированной доставки Значение 1 показывает, что BEI должен включаться в сигнализацию ATM; значение 0 показывает, что BEI не должен включаться в сигнализацию ATM. Ссылка: Форум ATM UNI 4.0	
TI	4010	Булевый	Индикатор тегирования Значение 0 показывает, что тегирование не разрешается; значение 1 показывает, что есть запрос на тегирование. Ссылка: Рек. МСЭ-Т Q.2961.1	
FD	4011	Булевый	Сброс кадра Значение 0 показывает, что сброс кадра не разрешается; значение 1 показывает, что сброс кадра разрешается. Ссылка: Форум ATM UNI 4.0	
A2PCDV	4012	24-битовое целое число	Допустимое 2-точечное CDV Ссылка: Рек. МСЭ-Т Q.2965.2	
C2PCDV	4013	24-битовое целое число	Накопленное 2-точечное CDV Ссылка: Рек. МСЭ-Т Q.2965.2	
APPCDV	4014	24-битовое целое число	Допустимое P-P CDV Ссылка: Форум ATM UNI 4.0	
CPPCDV	4015	24-битовое целое число	Накопленное P-P CDV Ссылка: Форум ATM UNI 4.0	
ACLR	4016	8-битовое целое число	Допустимый коэффициент потерь ячеек Ссылка: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
MEETD	4017	16-битовое целое число	Максимальная сквозная транзитная задержка Ссылка: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
CEETD	4018	16-битовое целое число	Накопленная сквозная транзитная задержка Ссылка: Рек. МСЭ-Т Q.2965.2, Форум ATM UNI 4.0	
QoSClass	4019	Целое число 0-5	Класс QoS	
			Класс QoS	Значение
			0	QoS по умолчанию связано с АТС, как определяется в Рек. МСЭ-Т Q.2961.2
			1	Строгий
			2	Терпимый
			3	Двухуровневый
4	Неограниченный			

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value	
			5	Строгий двухуровневый
			Ссылка: Рек. МСЭ-Т Q.2965.1	
AALtype	401A	1 octet	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 пользовательский AAL Ссылка: Рек. МСЭ-Т Q.2931	

C.5 Frame Relay

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
DLCI	5001	Целое число без знака	Номер виртуального соединения
CID	5002	Целое число без знака	Номер субканала
SID/Noiselevel	5003	Целое число без знака	Дескриптор внесения пауз
Primary Payload type	5004	Целое число без знака	Тип первичной нагрузки Охватывает факс и кодеки

C.6 Протокол IP

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
IPv4	6001	32 бита Ipv4Address	Адрес Ipv4 Ссылка: IETF RFC 791
IPv6	6002	128 бит	Адрес IPv6 Ссылка: IETF RFC 2460
Port	6003	Целое число без знака	0..65535
Porttype	6004	Перечислимый	TCP(0), UDP(1), SCTP(2)

C.7 Уровень адаптации типа 2 ATM

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
AESA	7001	20 октетов	Адрес конечной точки службы AAL2, как определён в Рекомендации, указанной в ссылке. ESEA NSEA Ссылка: Рек. МСЭ-Т Q.2630.1
BIR	См. C.3	4 октетов	Справка, созданная обслуживаемым пользователем, в соответствии с Рекомендацией, указанной в ссылке.

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
			SUGR Ссылка: Рек. МСЭ-Т Q.2630.1
ALC	7002	12 октетов	Характеристики звена AAL2 в соответствии с Рекомендацией, указанной в ссылке. Максимальная/Средняя битовая скорость CPS-SDU; Максимальный/Средний размер CPS-SDU Ссылка: Рек. МСЭ-Т Q.2630.1
SSCS	7003	I.366.2: Аудио (8 октетов); Многоскоростной (3 октета), или I.366.1: Обеспеченный SAR (14 октетов); Необеспеченный SAR (7 октетов)	Информация служебно-ориентированного подуровня, как определяется в: – Рек. МСЭ-Т Q.2630.1, и используется в: – Рек. МСЭ-Т I.366.2: аудио/многоскоростной; – Рек. МСЭ-Т I.366.1: обеспеченный/ необеспеченный SAR. Ссылка: Рекомендации МСЭ-Т Q.2630.1, I.366.1 и I.366.2
SUT	7004	1..254 октетов	Транспортный аргумент обслуживаемого пользователя, как определяется в Рекомендации, указанной в ссылке. Ссылка: Рек. МСЭ-Т Q.2630.1
TCI	7005	Булевый	Индикатор измерения соединения, как определяется в Рекомендации, указанной в ссылке. Ссылка: Рек. МСЭ-Т Q.2630.1
Timer_CU	7006	32-битовое целое число	Таймер-CU Количество миллисекунд для удержания частично заполненной ячейки перед отправкой.
MaxCPSSDU	7007	8-битовое целое число	Максимальный блок данных общей части подуровня службы Ссылка: Рек. МСЭ-Т Q.2630.1
CID	7008	8 бит	Номер субканала: 0-255 Ссылка: Рек. МСЭ-Т I.363.2

С.8 Уровень адаптации типа 1 АТМ

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
BIR	См. таблицу в С.3	4-29 октетов	GIT (общий идентификатор транспорта) Ссылка: Рек. МСЭ-Т Q.2941.1
AAL1ST	8001	1 октет	Подтип AAL1

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
			0 0 1 0 0 0 1 1 22 × 64 Кбит/с неограниченное 0 0 1 0 0 1 0 0 23 × 64 Кбит/с неограниченное 0 0 1 0 0 1 0 1 запас 0 0 1 0 0 1 1 0 25 × 64 Кбит/с неограниченное 0 0 1 0 0 1 1 1 26 × 64 Кбит/с неограниченное 0 0 1 0 1 0 0 0 27 × 64 Кбит/с неограниченное 0 0 1 0 1 0 0 1 28 × 64 Кбит/с неограниченное 0 0 1 0 1 0 1 0 29 × 64 Кбит/cunrestricted 0 0 1 0 1 0 1 1 по резерв 1 1 1 1 1 1 1 1 Ссылка: Рек. МСЭ-Т Q.763
TMRSR	9002	1 октет	Диапазон скоростей по требованию среды передачи 0 не определено 1 8 Кбит/с 2 16 Кбит/с 3 32 Кбит/с
Contcheck	9003	Булевый	Проверка связности 0 в этом канале проверка связности не требуется 1 в этом канале проверка связности требуется Ссылка: Рек. МСЭ-Т Q.763
ITC	9004	5 бит	Возможность переноса информации 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 аудио 3.1 КГц 1 0 0 0 1 неограниченная цифровая информация тональными сигналами/объявлениями 1 1 0 0 0 видео Все другие значения резервируются. Ссылка: Рек. МСЭ-Т Q.763
TransMode	9005	2 бита	Режим переноса Биты <u>2 1</u> 0 0 канальный режим 1 0 пакетный режим Ссылка: Рек. МСЭ-Т Q.931

Название PropertyID	Тег свойства Property tag	Тип Туре	Значение Value
TransRate	9006	5 бит	Скорость переноса Биты <u>5 4 3 2 1</u> 0 0 0 0 этот код должен использоваться при соединениях в пакетном режиме 1 0 0 0 64 Кбит/с 1 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) Ссылка: Рек. МСЭ-Т Q.931
MULT	9007	7 бит	Множитель скорости Любое значение от 2 до n (максимальное число В-каналов) Ссылка: Рек. МСЭ-Т Q.931
layer1prot	9008	5 бит	Информация пользователя, протокол 1-го уровня Биты <u>5 4 3 2 1</u> 0 0 0 0 1 адаптация стандартизованной скорости МСЭ-Т V.110 и X.30. 0 0 0 1 0 Рек. МСЭ-Т G.711 μ-закон 0 0 0 1 1 Рек. МСЭ-Т G.711 А-закон 0 0 1 0 0 Рек. МСЭ-Т G.721 32 Кбит/с АДИКМ и Рек. МСЭ-Т I.460 0 0 1 0 1 Рекомендации МСЭ-Т Н.221 и Н.242 0 0 1 1 0 Рекомендации МСЭ-Т Н.223 И Н.245 0 0 1 1 1 адаптация скорости, не стандартизованной МСЭ-Т. 0 1 0 0 0 адаптация стандартизованной скорости МСЭ-Т V.120. 0 1 0 0 1 адаптация стандартизованной скорости МСЭ-Т X.31, вставка флага HDLC Все другие значения резервируются. Ссылка: Рек. МСЭ-Т Q.931
syncasync	9009	Булевый	Синхронный/Асинхронный 0 Синхронные данные 1 Асинхронные данные Ссылка: Рек. МСЭ-Т Q.931
negotiation	900A	Булевый	Согласование 0 возможно внутрисполосное согласование 1 не возможно внутрисполосное согласование Ссылка: Рек. МСЭ-Т Q.931

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
Userrate	900B	5 бит	<p>Пользовательская скорость</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 0 0 Скорость показывается в Е-битах, определённых в Рек. МСЭ-Т I.460 или может согласовываться внутриволноводно</p> <p>0 0 0 0 1 0.6 Кбит/с Рек-ции 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-ции 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-ции 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-ции 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-ции 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-ции МСЭ-Т V.6 и X.1</p> <p>1 1 0 0 0 1.2/0.075 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 0 0 1 0.050 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 0 1 0 0.075 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 0 1 1 0.110 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 1 0 0 0.150 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 1 0 1 0.200 Кбит/с Рек-ции МСЭ-Т V.6 и X.1</p> <p>1 1 1 1 0 0.300 Кбит/с Рек-ции 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 другие значения зарезервированы.</p> <p>Ссылка: Рек. МСЭ-Т Q.931</p>
INTRATE	900C	2 бита	<p>Промежуточная скорость</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>Ссылка: Рек. МСЭ-Т Q.931</p>
nictx	900D	Булевый	<p>Независимый генератор сети (NIC) на передаче</p> <p>0 не требуется отправлять данные с независимым генератором сети</p> <p>1 требуется отправлять данные с независимым генератором сети</p>

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
			Ссылка: Рек. МСЭ-Т Q.931
nicrx	900E	Булевый	Независимый генератор сети (NIC) на приёме 0 не может принимать данные с независимым генератором сети (т.е. отправитель не поддерживает эту факультативную процедуру) 1 может принимать данные с независимым генератором сети (т.е. отправитель поддерживает эту факультативную процедуру) Ссылка: Рек. МСЭ-Т Q.931
flowconttx	900F	Булевый	Управление потоком на передаче (Tx) 0 не требуется отправлять данные с механизмом управления потоком 1 требуется отправлять данные с механизмом управления потоком Ссылка: Рек. МСЭ-Т Q.931
flowcontrx	9010	Булевый	Управление потоком на приёме (Rx) 0 не может принимать данные с механизмом управления потоком (т.е. отправитель не поддерживает эту факультативную процедуру) 1 может принимать данные с механизмом управления потоком (т.е. отправитель поддерживает эту факультативную процедуру) Ссылка: Рек. МСЭ-Т Q.931
rateadapthdr	9011	Булевый	Скорость адаптации заголовков/нет заголовка 0 Скорость адаптации заголовков не включена 1 Скорость адаптации заголовков включена Ссылка: Рек. МСЭ-Т Q.931
multiframe	9012	Булевый	Поддержка в линии передачи данных образований из нескольких кадров 0 Нет поддержки образований из нескольких кадров. Разрешены только кадры типа UI 1 Поддержка образований из нескольких кадров Ссылка: Рек. МСЭ-Т Q.931
OPMODE	9013	Булевый	Режим работы 0 Двоичный прозрачный режим работы 1 Режим работы, чувствительный к протоколу Ссылка: Рек. МСЭ-Т Q.931
llidnegot	9014	Булевый	Согласование идентификатора логической линии 0 по умолчанию, LLI = только 256 1 согласование полного протокола Ссылка: Рек. МСЭ-Т Q.931
assign	9015	Булевый	Поручитель/правопреемник 0 источник сообщения- "правопреемник по умолчанию" 1 источник сообщения- "только поручитель" Ссылка: Рек. МСЭ-Т Q.931

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
inbandneg	9016	Булевый	Внутриполосное/внеполосное согласование 0 согласование выполнено с сообщениями ИНФОРМАЦИЯ ПОЛЬЗОВАТЕЛЯ на временном сигнальном соединении 1 внутриполосное согласование выполнено при использовании логической линии нуль Ссылка: Рек. МСЭ-Т Q.931
stopbits	9017	2 бита	Количество стоповых битов Биты <u>2 1</u> 0 0 не используется 0 1 1 бит 1 0 1.5 бита 1 1 2 бита Ссылка: Рек. МСЭ-Т Q.931
databits	9018	2 бита	Количество битов данных без учёта бита проверки начётность, при его наличии Биты <u>2 1</u> 0 0 не используется 0 1 5 бит 1 0 7 бит 1 1 8 бит Ссылка: Рек. МСЭ-Т Q.931
parity	9019	3 бита	Информация о проверке на чётность Биты <u>3 2 1</u> 0 0 0 нечётный 0 1 0 чётный 0 1 1 никакой 1 0 0 установлено в 0 1 0 1 установлено в 1 Все другие значения зарезервированы. Ссылка: Рек. МСЭ-Т Q.931
duplexmode	901A	Булевый	Дуплексный режим 0 Полудуплекс 1 Полный дуплекс Ссылка: Рек. МСЭ-Т Q.931

Название PropertyID	Тег свойства Property tag	Тип Туре	Значение Value
modem	901B	6 бит	<p>Тип модема</p> <p>Биты</p> <p><u>6 5 4 3 2 1</u></p> <p>0 0 0 0 0 0</p> <p>по Национальное использование</p> <p>0 0 0 1 0 1</p> <p>0 1 0 0 0 1 Рек. МСЭ-Т V.21</p> <p>0 1 0 0 1 0 Рек. МСЭ-Т V.22</p> <p>0 1 0 0 1 1 Рек. МСЭ-Т V.22 <i>bis</i></p> <p>0 1 0 1 0 0 Рек. МСЭ-Т V.23</p> <p>0 1 0 1 0 1 Рек. МСЭ-Т V.26</p> <p>0 1 1 0 0 1 Рек. МСЭ-Т V.26 <i>bis</i></p> <p>0 1 0 1 1 1 Рек. МСЭ-Т V.26 <i>ter</i></p> <p>0 1 1 0 0 0 Рек. МСЭ-Т V.27</p> <p>0 1 1 0 0 1 Рек. МСЭ-Т V.27 <i>bis</i></p> <p>0 1 1 0 1 0 Рек. МСЭ-Т V.27 <i>ter</i></p> <p>0 1 1 0 1 1 Рек. МСЭ-Т V.29</p> <p>0 1 1 1 0 1 Рек. МСЭ-Т V.32</p> <p>0 1 1 1 1 0 Рек. МСЭ-Т V.34</p> <p>1 0 0 0 0 0</p> <p>по Национальное использование</p> <p>1 0 1 1 1 1</p> <p>1 1 0 0 0 0</p> <p>по Определяется пользователем</p> <p>1 1 1 1 1 1</p> <p>Ссылка: Рек. МСЭ-Т Q.931</p>
layer2prot	901C	5 бит	<p>Информация пользователя, протокол 2-го уровня</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Рек. МСЭ-Т Q.921/L.441</p> <p>0 0 1 1 0 Рек. МСЭ-Т X.25, линейный уровень</p> <p>0 1 1 0 0 управление логическим каналом локальной сети (МОС/МЭК 8802-2)</p> <p>Все другие значения зарезервированы.</p> <p>Ссылка: Рек. МСЭ-Т Q.931</p>
layer3prot	901D	5 бит	<p>Информация пользователя, протокол 3-го уровня</p> <p>Биты</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Рек. МСЭ-Т Q.931</p> <p>0 0 1 1 0 Рек. МСЭ-Т X.25, пакетный уровень</p> <p>0 1 0 1 1 Рек. МСЭ-Т X.263 МОС/МЭК TR 9577 (Идентификация протокола на сетевом уровне)</p> <p>Все другие значения зарезервированы.</p> <p>Ссылка: Рек. МСЭ-Т Q.931</p>

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
addlayer3prot	901E	Октет	Дополнительная информация пользователя, протокол 3-го уровня Биты Биты <u>4 3 2 1</u> <u>4 3 2 1</u> 1 1 0 0 1 1 0 0 Протокол Интернет (IETF RFC 791) (Рек. МСЭ-Т X.263 МОС/МЭК TR 9577) 1 1 0 0 1 1 1 1 Протокол точка-точка (IETF RFC 1661) Ссылка: Рек. МСЭ-Т Q.931
DialledN	901F	30 октетов	Набранный номер
DiallingN	9020	30 октетов	Набираемый номер
ECHOIC	9021		Не используется. См. пример возможных свойств Echo Control в E.13/H.248.1.
NCI	9022	1 октет	Природа индикаторов соединения Биты <u>2 1</u> Спутниковый индикатор 0 0 нет спутникового канала в соединении 0 1 один спутниковый канал в соединении 1 0 два спутниковых канала в соединении 1 1 запас Биты <u>4 3</u> Индикатор проверки связности 0 0 проверка связности не требуется 0 1 в данном канале проверка связности требуется 1 0 в предыдущем канале проверка связности выполнена 1 1 запас Бит <u>5</u> Индикатор устройства управления эхо 0 исходящее устройство управления эхо не включено 1 исходящее устройство управления эхо включено Биты <u>8 7 6</u> Запас Ссылка: Рек. МСЭ-Т Q.763
USI	9023	Октетная строка	Информация службы пользователя Ссылка: 3.57/Q.763

С.10 Свойства уровня адаптации типа 5

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
FMSDU	A001	32-битовое целое число	Максимальный прямой размер CPCS-SDU: Максимальный размер CPCS-SDU, отправленный в направлении от вызывающего к вызываемому абоненту. Ссылка: Рек. МСЭ-Т Q.2931
BMSDU	A002	32-битовое целое число	Максимальный обратный размер CPCS-SDU: Максимальный размер CPCS-SDU отправленный в направлении от вызываемого к вызывающему абоненту. Ссылка: Рек. МСЭ-Т Q.2931
SSCS	См. таблицу в С.7	См. таблицу в С.7	См. таблицу в С.7 Дополнительные значения: VPI/VCI

С.11 Эквиваленты SDP

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
SDP_V	B001	Строка	Версия протокола Ссылка: IETF RFC 2327
SDP_O	B002	Строка	Владелец/создатель и идентификатор сессии Ссылка: IETF RFC 2327
SDP_S	B003	Строка	Название сеанса Ссылка: IETF RFC 2327
SDP_I	B004	Строка	Идентификатор сеанса Ссылка: IETF RFC 2327
SDP_U	B005	Строка	URI дескриптора Ссылка: IETF RFC 2327
SDC_E	B006	Строка	Адрес электронной почты Ссылка: IETF RFC 2327
SDP_P	B007	Строка	Номер телефона Ссылка: IETF RFC 2327
SDP_C	B008	Строка	Информация о соединении Ссылка: IETF RFC 2327
SDP_B	B009	Строка	Информация о полосе частот Ссылка: IETF RFC 2327
SDP_Z	B00A	Строка	Настройка временной зоны Ссылка: IETF RFC 2327
SDP_K	B00B	Строка	Ключ шифрования Ссылка: IETF RFC 2327
SDP_A	B00C	Строка	Нуль или более качественных признаков сессии Ссылка: IETF RFC 2327

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
SDP_T	B00D	Строка	Активное время сеанса Ссылка: IETF RFC 2327
SDP_R	B00E	Строка	Нуль или более раз повторений Reference: IETF RFC 2327
SDP_M	B00F	Строка	Тип среды передачи, порт, транспорт и формат Ссылка: IETF RFC 2327

C.12 Рекомендация МСЭ-Т Н.245

Название PropertyID	Тег свойства Property tag	Тип Type	Значение Value
OLC	C001	Октетная строка	Значение Н.245 структуры OpenLogicalChannel. Ссылка: Рек. МСЭ-Т Н.245
OLCack	C002	Октетная строка	Значение Н.245 структуры OpenLogicalChannelAck. Ссылка: Рек. МСЭ-Т Н.245
OLCcnf	C003	Октетная строка	Значение Н.245 структуры OpenLogicalChannelConfirm. Ссылка: Рек. МСЭ-Т Н.245
OLCrej	C004	Октетная строка	Значение Н.245 структуры OpenLogicalChannelReject. Ссылка: Рек. МСЭ-Т Н.245
CLC	C005	Октетная строка	Значение Н.245 структуры CloseLogicalChannel. Ссылка: Рек. МСЭ-Т Н.245
CLCack	C006	Октетная строка	Значение Н.245 структуры CloseLogicalChannelAck. Ссылка: Рек. МСЭ-Т Н.245

Приложение D

Транспорт через IP

D.1 Транспорт через IP/UDP с использованием прикладного уровня кадрирования (ALF)

Сообщения протокола, определённые в данной Рекомендации, могут передаваться через UDP. Когда сетевое устройство не обеспечивает порт (см. 7.2.8), команды должны посылаться на номер порта по умолчанию: 2944 - для действия с текстовым кодированием или 2945 – для действия с двоичным кодированием. Отклики должны отправляться на адрес и порт, из которых отправлены соответствующие команды.

Прикладной уровень кадрирования ALF представляет собой ряд технических приёмов, который позволяет приложению, в противоположность стеку, влиять на то, как сообщения отправляются другой стороне. Типичная техника ALF должна позволять приложению изменять порядок отправляемых сообщений, когда имеется очередь, после того, как оно

поставило их в очередь. Официальная спецификация ALF отсутствует. Процедуры, приведенные в Приложении D.1, содержат минимальный предлагаемый набор режимов работы ALF.

Исполнители, применяющие IP/UDP с ALF, должны знать об ограничениях MTU на максимальный размер сообщения.

D.1.1 Обеспечение самое большее однократных функциональных возможностей

Сообщения, передаваемые по UDP, могут быть потеряны. При отсутствии своевременного ответа команды повторяются. Большинство команд являются идемпотентными. Состояние шлюза MG будет становиться непредсказуемым, если, например, команды Add исполнялись бы несколько раз. Таким образом, процедуры передачи обеспечат "самое большее однократные" функциональные возможности.

Ожидается, что равноправные объекты протокола хранят в памяти список ответов, которые были ими посланы в последних транзакциях, а также список транзакций, которые теперь предстоит выполнить. Идентификатор транзакции каждого входящего сообщения сравнивается с идентификаторами транзакций последних ответов, направленных тому же MId. Если соответствие найдено, объект не исполняет транзакцию, а просто повторяет ответ. Если соответствие не найдено, сообщение будет сравниваться со списком транзакций, которые теперь предстоит выполнить. Если в этом списке найдено соответствие, указывающее на повторную транзакцию, объект не исполняет транзакцию (см. D.1.4 относительно процедур по передаче TransactionPending).

Процедура использует значение длинного таймера, в последующем обозначенного как LONG-TIMER. Таймер должен устанавливаться больше максимальной длительности транзакции, что должно принимать во внимание максимальное число повторений, максимальное значение таймера повторения и максимальную задержку распространения пакета в сети. Предлагаемое значение равно 30 секундам.

Копия ответов может быть разрушена либо в течение LONG-TIMER секунд после появления ответа, или, когда объект принимает подтверждение получения ответа посредством "Аргумента подтверждения ответа". Для транзакций, которые подтверждаются посредством этого аргумента, объект должен сохранять копию идентификатора транзакции в течение LONG-TIMER секунд после появления ответа для того, чтобы обнаружить и пренебречь двойной копией запроса на транзакцию, которая может быть сформирована сетью.

D.1.2 Идентификаторы транзакций и трёхстороннее подтверждение соединения

D.1.2.1 Идентификаторы транзакций

Идентификаторы транзакций являются 32-битовыми целыми числами. Контроллер медиашлюза может принять решение об использовании определённого числового пространства для каждого шлюза MG, которым они управляют, или использовать то же числовое пространство для всех шлюзов MG, которые принадлежат некоторой произвольно-выбранной группе. Контроллеры MGC могут принять решение распределить нагрузку управления большим шлюзом MG между несколькими независимыми процессами. Эти процессы будут совместно использовать одно и то же числовое пространство транзакции. Имеется несколько возможных реализаций такого совместного использования, как например, наличие централизованного выделения идентификаторов транзакций или предварительного выделения неперекрывающихся областей идентификаторов транзакций различным процессам. Реализации должны гарантировать, что уникальные идентификаторы транзакций выделяются всем транзакциям, исходящим от логического контроллера MGC (идентичный mId). Шлюзы MG могут просто обнаружить двойные транзакции только взглянув на идентификатор транзакции и mId.

D.1.2.2 Трёхстороннее подтверждение соединения

В любом сообщении может быть найден аргумент подтверждения TransactionResponse. Он содержит набор "диапазонов подтверждённых идентификаторов транзакций". Объекты могут выбрать удаление копий ответов на транзакции, чьи идентификаторы включены в "диапазон подтверждённых идентификаторов транзакций", принятых в сообщениях ответов на транзакции. Если идентификатор транзакции попадает в эти диапазоны, они должны молча отбросить последующие команды.

Значения "диапазонов подтверждённых идентификаторов транзакций" не должны использоваться, если более, чем LONG-TIMER секунд истекло с момента выдачи шлюзом MG своего последнего ответа этому контроллеру MGC, или, если шлюз MG возобновляет работу. В такой ситуации транзакции должны приниматься и обрабатываться без какой-либо проверки идентификатора транзакции.

Сообщения, которые содержат аргумент "подтверждение ответа на транзакцию" могут передаваться в любом порядке. Объект должен сохранять "диапазоны подтверждённых идентификаторов транзакций", принятых в течение LONG-TIMER секунд.

При двоичном кодировании, если в подтверждении ответа (см. A.2) присутствует firstAck, то подтверждается только одна транзакция. Если присутствуют оба firstAck и lastAck, то подтверждается диапазон транзакций от firstAck до lastAck. При текстовом кодировании для указания диапазона подтверждаемых транзакций используется тире (см. B.2).

D.1.3 Вычисление таймеров повторной передачи

Ответственностью запрашивающего объекта является обеспечение подходящих тайм-аутов для всех просроченных транзакций, а также повторение транзакций при превышении тайм-аутов. Более того, если повторенные транзакции не подтверждаются, ответственностью запрашивающего объекта является поиск резервных служб и/или очистка существующих или ожидающих соединений.

Спецификация специально избегает определение какого либо значения для таймеров повторной передачи. Как правило эти значения зависят от сети. Обычно таймеры повторной передачи должны оценивать значение таймера посредством измерения интервала времени между посылкой команды и возвращением ответа. Реализации ДОЛЖНЫ обеспечить, чтобы алгоритм, используемый для вычисления интервала повторной передачи, обеспечивал экспоненциально возрастающую выдержку времени тайм-аута повторной передачи для каждой повторной передачи или повторения после первой повторной передачи.

ПРИМЕЧАНИЕ – Одной из возможностей является использование алгоритма, реализованного в TCP-IP, который использует две переменные:

- Среднюю задержку подтверждения (AAD), оценка которой сделана при помощи экспоненциально сглаженного среднего наблюдаемых задержек.
- Среднее отклонение (ADEV), оценка которого сделана при помощи экспоненциально сглаженного среднего абсолютного значения разности между наблюдаемой задержкой и существующим средним. Таймер повторной передачи в TCP, устанавливается на сумму средней задержки плюс N раз среднее отклонение. Однако, для протокола, определяемого в данной Рекомендации, максимальное значение таймера должно быть ограничено для того, чтобы гарантировать, что после LONG-TIMER секунд шлюзы не получают повторного пакета. Предложенное максимальное значение равно 4 секундам.

После любой повторной передачи объект ДОЛЖЕН делать следующее:

- Он должен увеличить вдвое оценку значения средней задержки, AAD.
- Он должен вычислить случайное значение равномерно распределённое между 0.5 AAD и AAD.

- Он должен установить таймер повторной передачи на сумму этого случайного значения и N-кратного среднего отклонения.

Данная процедура имеет два следствия. В связи с тем, что процедура включает экспоненциально возрастающую составляющую, в случае перегрузки она будет автоматически замедлять поток сообщений. Т.к. процедура включает случайную составляющую, она будет прерывать потенциальную синхронизацию между уведомлениями, вызванными тем же самым внешним событием.

D.1.4 Предварительные ответы

Исполнение некоторых транзакций может потребовать продолжительное время. Продолжительные времена исполнения могут взаимодействовать с процедурой повторной передачи, основанной на таймере. Это может иметь результатом либо чрезмерное число повторных передач, или в значениях таймера которые становятся слишком продолжительными, чтобы быть эффективными. Объекты, которые могут предсказать, что транзакция потребует продолжительное время могут посылать предварительный ответ "Ожидание транзакции". Они ДОЛЖНЫ послать этот ответ, если получают повторение транзакции, которая всё ещё исполняется.

Объекты, которые получают ожидание транзакции должны переключиться к другому таймеру повторения для повторения запросов. Корневое окончание имеет свойство (значение таймера `ProvisionalResponse`), который может быть установлен на запрошенное максимальное число миллисекунд между приёмом команды и передачей ответа `TransactionPending`. После приёма окончательного ответа, следующего за приёмом предварительных ответов, должно отправляться немедленное подтверждение, а обычные таймеры повторения должны использоваться впоследствии. Объект, посылающий предварительный ответ, ДОЛЖЕН включить поле `immAckRequired` в последующий окончательный ответ, показывающий, что ожидается немедленное подтверждение. Приём ожидания транзакции после приёма ответа должно пренебрегаться.

D.1.5 Повторение запросов, ответов и подтверждений

Протокол организован в виде набора транзакций, каждая из которых содержит запрос и ответ, обычно упоминаемые, как подтверждение. Сообщения протокола, передаваемые по UDP, могут претерпевать потери. В отсутствие своевременного ответа транзакции повторяются. Ожидается, что объекты хранят в памяти список ответов, которые они отправляли недавним транзакциям, т.е. список всех ответов, которые они отправляли по заключительному LONG-TIMER секундам и список транзакций, которые в настоящее время исполняются.

Механизм повторения используется для дого, чтобы защитить от трёх типов возможных ошибок:

- ошибки передачи, когда, например, пакет теряется вследствие линейного шума или перегрузки в очереди;
- отказ составляющей, когда, например, стык к объекту становится недоступным;
- отказ объекта, когда, например, целый объект становится недоступным.

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 меньше 1%. Если контроллер медиашлюза или медиашлюз должен повторять сообщение больше нескольких раз, очень разумно предположить, что происходит что-то ещё, кроме ошибок передачи. Например, принимая коэффициент потерь равным 1%, вероятность того, что пять последовательных попыток передачи 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.

После повторной передачи пакета слишком большое число раз (обычно 7-11 раз) классический алгоритм повторной передачи просто подсчитывает число успешных последовательных повторений и придёт к заключению, что взаимосвязь разорвана. Для того, чтобы подсчитать вероятность необнаруженного или текущего "переключения", мы изменяем классический алгоритм так, что, если медиашлюз получает правильное сообщение ServiceChange, объявляющее переключение, он начнёт передачу просроченных команд этому новому контроллеру MGC. Ответы на команды ещё передаются по исходному адресу команды.

Для того, чтобы автоматически адаптироваться к нагрузке сети, данная Рекомендация определяет экспоненциально возрастающие таймеры. Если начальный таймер установлен на 200 миллисекунд, потеря пятой повторной передачи будет обнаруживаться примерно после 6 секунд. Возможно, что это допустимая задержка ожидания для обнаружения переключения. После такой задержки повторы передачи должны происходить не только для того, чтобы, возможно, преодолеть временную проблему связности, но также для того, чтобы позволить несколько больше времени для исполнения (возможно приемлимой является общая задержка ожидания, равная 30 секунд).

Однако важно, чтобы максимальная задержка повторных передач была бы ограниченной. Перед любой повторной передачей проверяется, что время, прошедшее с момента отправки начальной датаграммы, не превышает T-MAX. Если прошло время большее, чем T-MAX, шлюз MG приходит к заключению, что контроллер медиашлюза вышел из строя и начинает свой восстановительный процесс, как описывается в 11.5. Если шлюз MG пытается повторно соединиться с текущим контроллером MGC, он должен использовать ServiceChange с ServiceChangeMethod, установленным в разъединённое состояние так, чтобы новый контроллер медиашлюза MGC был бы уверен, что шлюз MG потерял одну или более транзакций. Значение T-MAX относится к значению LONG-TIMER: значение LONG-TIMER получается путём прибавления к T-MAX максимальной задержки распространения в сети.

D.2 Использование TCP

Сообщения протокола, как определяется в данной Рекомендации, могут передаваться по TCP. Если порт не определяется другой стороной (см. 7.2.8), команды должны посылаться на порт по умолчанию. В качестве единицы перемещения в определяемом протоколе используется сообщение в то время, как TCP является потоко-ориентированным протоколом. В соответствии с IETF RFC 1006, ТРКТ ДОЛЖЕН использоваться для очерчивания сообщений в пределах потока TCP.

В протоколе, ориентированном на транзакции, всё же имеются пути для потерь запросов или ответов на транзакции. По существу рекомендуется, чтобы объекты, использующие транспорт TCP, реализовывали таймеры уровня приложений для каждого запроса и ответа, подобно определённым для кадрирования на уровне приложений через UDP.

D.2.1 Обеспечение самое большее однократных функциональных возможностей

Сообщения, передаваемые по TCP, не подвержены транспортным потерям, но потеря запроса или ответа на транзакцию могут, однако, быть отмечены в реальных реализациях. В отсутствие своевременного ответа, команды повторяются. Большинство команд являются идемпотентными. Состояние шлюза MG будет становиться непредсказуемым, если, например, команды Add были исполнены несколько раз.

Для защиты от таких потерь рекомендуется, чтобы объекты следовали процедурам, описанным в D.1.1.

D.2.2 Идентификаторы транзакций и трёхстороннее подтверждение соединения

По тем же самым причинам возможно, чтобы ответы на транзакции могли быть потеряны даже с использованием протокола надёжной доставки такого, как TCP. Рекомендуется, чтобы объекты следовали процедурам, описанным в D.1.2.2.

D.2.3 Вычисление таймеров повторной передачи

При надёжной доставке ожидается, что коэффициент потери запроса или ответа на транзакцию будет очень низким. Поэтому необходимы только простые механизмы таймеров. Экспоненциальные алгоритмы выдержки времени не должны потребоваться, хотя они могут применяться там, где, как в шлюзе MGC, уже есть требование, чтобы код делал так, т.к. контроллеры MGC должны реализовывать ALF/UDP также, как и TCP.

D.2.4 Предварительные ответы

Как и с UDP, исполнение некоторых транзакций может потребовать продолжительного времени. Объекты, которые могут предсказать, что транзакция потребует продолжительного времени исполнения могут послать предварительный ответ "ожидание транзакции". Они должны послать этот ответ, если они получают повторение транзакции, которая ещё исполняется.

Объекты, которые принимают ожидание транзакции, должны переключиться к более длинному таймеру повторения для этой транзакции.

Объекты должны удерживать транзакции и ответы до тех пор, пока они не подтверждены. Необходимо следовать основной процедуре, описанной в D.1.4, но значений простого таймера должно быть достаточно. После приёма окончательного ответа нет необходимости посылать немедленное подтверждение.

D.2.5 Упорядочение команд

TCP обеспечивает упорядоченную доставку транзакций. Никаких специальных процедур не требуется. Необходимо отметить, что ALF/UDP позволяет отправляющему объекту изменять своё поведение при перегрузке и, в частности, может переупорядочить транзакции, когда происходит перегрузка. Протокол TCP не может достичь тех же результатов.

Приложение E

Основные комплекты

Это приложение содержит определения некоторых комплектов, используемых в данной Рекомендации.

E.1 Общее

PackageID: g (0x0001)

Версия: 1

Расширения: Нет

Описание:

Общий комплект для обычно встречающихся пунктов

Е.1.1 Свойства

Нет

Е.1.2 События

Причина

EventID: причина (0x0001)

Общее событие ошибки

Аргументы EventsDescriptor: Нет

Аргументы дескриптора ObservedEvents:

Общая причина

ParameterID: Generalcause (0x0001)

Этот аргумент группирует отказы в шесть групп, на которые может воздействовать контроллер.

Тип: перечисление

Возможные значения:

"NR" Обычный вариант (0x0001)

"UR" Недоступные ресурсы (0x0002)

"FT" Отказ, временный (0x0003)

"FP" Отказ, постоянный (0x0004)

"IW" Ошибка взаимодействия (0x0005)

"UN" Неподдерживаемый (0x0006)

Причина отказа

ParameterID: Failurecause (0x0002)

Возможные значения: ОКТЕТНАЯ СТРОКА

Описание: Причина отказа является значением, генерируемым освобождённым оборудованием, т.е. освобождённым сетевым соединением. Указанное значение определяется в соответствующем протоколе управления транспортным каналом.

Сигнал завершения

EventID: sc (0x0002)

Показывает окончание сигнала, которому был передан аргумент notifyCompletion для того, чтобы задействовать отчётность о завершении события. См. 7.1.1, 7.1.17 и 7.2.7 для более подробного описания процедур.

Аргументы EventsDescriptor: Нет

Аргументы дескриптора ObservedEvents:

Идентификация сигнала

ParameterID: SigID (0x0001)

Данный аргумент идентифицирует сигнал, который закончился. Для сигнала, который содержится в списке сигналов, должен также возвращаться аргумент идентификации списка сигналов, показывающий соответствующий список.

Тип: Двоичный: октет (строка), Текст: строка

Возможные значения: закончившийся сигнал. Сигнал должен идентифицироваться путём использования синтаксиса pkgdName без групповых символов замены.

Метод окончания

ParameterID: Meth (0x0002)

Показывает средства, которыми заканчивается сигнал.

Тип: перечисление

Возможные значения:

"TO" (0x0001) сигнал прервался или, по иному, закончился сам собой

"EV" (0x0002) прерван событием

"SD" (0x0003) остановлен новым дескриптором Signals

"NC" (0x0004) не закончился, другая причина

Список идентификаторов сигналов

ParameterID: SLID (0x0003)

Показывает какому списку сигналов принадлежит сигнал. В тех случаях, когда сигнал находится в списке сигналов, возвращается только идентификатор SignalList ID.

Тип: целое число

Возможные значения: любое целое число

Е.1.3 Сигналы

Нет.

Е.1.4 Статистика

Нет.

Е.2 Основной комплект Root

Основной комплект Root

PackageID: root (0x0002)

Версия: 1

Расширения: Нет

Описание:

Этот комплект определяет расширенные свойства шлюза.

Е.2.1 Свойства

MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

Значение этого свойства даёт максимальное число контекстов, которое может существовать в любое время. В это число не включается контекст NULL.

Тип: двойной

Возможные значения: 1 и больше

Определены в: TerminationState

Характеристики: только для чтения

MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

Максимальное число разрешённых окончаний в контексте, см. 6.1

Тип: целое число

Возможные значения: любое целое число

Определено в: TerminationState

Характеристики: только для чтения

normalMGExecutionTime

PropertyId: normalMGExecutionTime (0x0003)

Устанавливается контроллером MGC для указания интервала, в пределах которого контроллер MGC ожидает ответ на любую транзакцию со стороны шлюза MG (исключая задержку в сети)

Тип: целое число

Возможные значения: любое целое число, представляет миллисекунды

Определено в: TerminationState

Характеристики: для чтения/записи

normalMGCExecutionTime

PropertyId: normalMGCExecutionTime (0x0004)

Устанавливается контроллером MGC для указания интервала, в пределах которого шлюз MG ожидает ответ на любую транзакцию со стороны контроллера MGC (исключая задержку в сети)

Тип: целое число

Возможные значения: любое целое число, представляет миллисекунды

Определено в: TerminationState

Характеристики: для чтения/записи

MGProvisionalResponseTimerValue

PropertyId: MGProvisionalResponseTimerValue (0x0005)

Показывает время в пределах которого контроллер MGC должен ожидать задержанный ответ от шлюза MG, если транзакция не может завершиться. Вначале устанавливается на normalMGExecutionTime плюс задержка в сети, но может быть уменьшена.

Тип: Целое число

Возможные значения: любое целое число, представляет миллисекунды

Определено в: TerminationState

Характеристики: для чтения/записи

MGCProvisionalResponseTimerValue

PropertyId: MGCProvisionalResponseTimerValue (0x0006)

Показывает время в пределах которого шлюз MG должен ожидать задержанный ответ от контроллера MGC, если транзакция не может завершиться. Вначале устанавливается на normalMGExecutionTime плюс задержка в сети, но может быть уменьшена.

Тип: Целое число

Возможные значения: любое целое число, представляет миллисекунды

Определено в: TerminationState

Характеристики: для чтения/записи

Е.2.2 События

Нет.

Е.2.3 Сигналы

Нет.

Е.2.4 Статистика

Нет.

Е.2.5 Процедуры

Нет.

Е.3 Комплект генерации тона

PackageID: tonegen (0x0003)

Версия: 1

Расширения: Нет.

Описание:

Этот комплект определяет сигналы для генерации аудио тонов. Данный комплект не определяет значения аргументов. Он предназначен для расширения. Вообще, тоны определяются как отдельные сигналы с аргументами, ind, представляющими собой время задержки "между цифрами", а идентификатор тона должен использоваться с исполняемыми тонами **playtones**. Идентификатор тона должен поддерживаться совместимым с любой тоновой генерацией того же самого тона. Ожидается, что шлюзы MG обеспечиваются характеристиками соответствующих тонов для страны расположения шлюза MG.

Предназначен исключительно для расширения: Да

Е.3.1 Свойства

Нет.

Е.3.2 События

Нет.

Е.3.3 Сигналы

Исполнение тона

SignalID: pt (0x0001)

Исполняет аудио тон по аудио каналу

Тип сигнала: короткий

Длительность: обеспечивается

Дополнительные параметры:

Список идентификаторов тонов

ParameterID: tl (0x0001)

Тип: список идентификаторов тонов

Список тонов, исполняемых последовательно. Список ДОЛЖЕН содержать один или более идентификаторов тонов.

Длительность интервала между сигналами

ParameterID: ind (0x0002)

Тип: целое число

Интервал между двумя последовательными тонами сигналами в миллисекундах

В данном комплекте идентификаторы тонов не определяются. Комплекты, расширяющие данный комплект, могут добавлять возможные значения идентификаторам тонов, а также отдельные тональные сигналы.

Е.3.4 Статистика

Нет.

Е.3.5 Процедуры

Нет.

Е.4 Комплект обнаружения тона

PackageID: tonedet (0x0004)

Версия: 1

Предназначен исключительно для расширения: Да

Расширения: Нет

Этот комплект определяет события обнаружения аудио тонов. Тоны выбираются по имени (идентификатор тона). Ожидается, что шлюзы MG обеспечиваются характеристиками соответствующих тонов для страны расположения шлюза MG.

Этот комплект не определяет значения аргумента. Он предназначается для расширения.

Е.4.1 Свойства

Нет.

Е.4.2 События

Обнаружено начало тона

EventID: std, 0x0001

Обнаруживает начало тона. Характеристики реального обнаружения тона зависят от реализации.

Аргументы EventsDescriptor:

Список идентификаторов тонов

ParameterID: tl (0x0001)

Тип: список идентификаторов тонов

Возможные значения: Единственный идентификатор тона, определённый в данном комплекте, представляет собой "групповой символ замены", которым является "*" при текстовом кодировании и 0x0000 – при бинарном. Расширения к этому комплекту будут дополнять возможные значения идентификатора тона. Если tl это "групповой символ замены", обнаруживается любой идентификатор тона.

Аргументы ObservedEventsDescriptor:

Идентификатор тона

ParameterID: tid (0x0003)

Тип: перечисление

Возможные значения: "групповой символ замены", определённый выше, является единственным значением, определённым в данном комплекте. Расширения к этому комплекту будут добавлять дополнительные возможные значения идентификатору тона.

Обнаружен конец тона

EventID: etd, 0x0002

Обнаруживает конец тона.

Аргументы EventDescriptor:

Список идентификаторов тонов

ParameterID: tl (0x0001)

Тип: перечисление или список перечисленных типов

Возможные значения: В данном комплекте возможные значения не определяются. Расширения к данному комплекту будут добавлять возможные значения идентификатору тона.

Аргументы ObservedEventsDescriptor:

Идентификатор тона

ParameterID: tid (0x0003)

Тип: перечисление

Возможные значения: "групповой символ замены", определённый выше, является единственным значением, определённым в данном комплекте. Расширения к этому комплекту будут добавлять дополнительные возможные значения идентификатору тона.

Длительность

ParameterId: dur (0x0002)

Тип: целое число, в миллисекундах

Этот аргумент содержит длительность тона с момента обнаружения до его конца.

Обнаружен длинный тон

EventID: ltd, 0x0003

Обнаруживает что тон исполняется в течение, по крайней мере, определённого времени

Аргументы EventDescriptor:

Список идентификаторов тонов

ParameterID: tl (0x0001)

Тип: перечисление или список

Возможные значения: "групповой символ замены", определённый выше, является единственным значением, определённым в данном комплекте. Расширения к этому комплекту будут добавлять дополнительные возможные значения идентификатору тона.

Длительность

ParameterID: dur (0x0002)

Тип: целое число, длительность для измерения с

Возможные значения: любое допустимое целое число, выраженное в миллисекундах

Аргументы ObservedEventsDescriptor:

Идентификатор тона

ParameterID: tid (0x0003)

Тип: Перечисление

Возможные значения: В данном комплекте возможные значения не определяются. Расширения к данному комплекту будут добавлять возможные значения идентификатору тона.

E.4.3 Сигналы

Нет.

E.4.4 Статистика

Нет.

E.4.5 Процедуры

Нет.

E.5 Основной комплект генератора DTMF

PackageID: dg (0x0005)

Версия: 1

Расширения: tonegen версия 1

Этот комплект определяет основные тоны DTMF как сигналы и расширяет допустимые значения аргумента tl исполняемого тона в tonegen.

E.5.1 Свойства

Нет.

E.5.2 События

Нет.

Е.5.3 Сигналы

Символ DTMF 0

SignalID: d0 (0x0010)

Генерирует тон DTMF 0. Физическая характеристика DTMF 0 определяется в шлюзе.

Тип сигнала: Короткий

Длительность: Обеспеченная

Дополнительные аргументы:

Нет

Дополнительные значения:

d0 (0x0010) определяется как идентификатор тона исполняемого тона

Другие символы DTMF определяются точно таким же образом. Приводится таблица со всеми именами сигналов и идентификаторами сигналов. Заметим, что каждый символ DTMF определяется как сигнал, так и как идентификатор тона, расширяющий таким образом комплект основной генерации тона. Также заметим, что DTMF SignalIds отличаются от имён, используемых в плане нумерации.

Имя сигнала	Идентификатор сигнала/ Идентификатор тона
Символ DTMF 0	d0 (0x0010)
Символ DTMF 1	d1 (0x0011)
Символ DTMF 2	d2 (0x0012)
Символ DTMF 3	d3 (0x0013)
Символ DTMF 4	d4 (0x0014)
Символ DTMF 5	d5 (0x0015)
Символ DTMF 6	d6 (0x0016)
Символ DTMF 7	d7 (0x0017)
Символ DTMF 8	d8 (0x0018)
Символ DTMF 9	d9 (0x0019)
Символ DTMF *	ds (0x0020)
Символ DTMF #	do (0x0021)
Символ DTMF A	da (0x001a)
Символ DTMF B	db (0x001b)
Символ DTMF C	dc (0x001c)
Символ DTMF D	dd (0x001d)

Е.5.4 Статистика

Нет.

Е.5.5 Процедуры

Нет.

Е.6 Комплект обнаружения DTMF

PackageID: dd (0x0006)

Версия: 1

Расширения: tonedet версия 1

Этот комплект определяет обнаружение основных тонов DTMF. Данный комплект расширяет возможные значения идентификатора тона в событиях "обнаружено начало тона", "обнаружен конец тона" и "обнаружен длинный тон".

Дополнительные значения идентификатора тона являются какими бы то ни было идентификаторами тонов, описанными в комплекте dg (основной комплект генератора DTMF).

Следующая таблица устанавливает соответствия между событиями DTMF и символами плана нумерации, 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"

Е.6.1 Свойства

Нет.

Е.6.2 События

Цифры DTMF

EventIds определяются с теми же самыми именами как и SignalIds, определённые в таблице, приведенной в Е.5.3.

Событие завершения DigitMap

EventID: ce, 0x0004

Генерируются, когда план нумерации завершается, как описано в 7.1.14.

Аргументы EventsDescriptor: Нет.

Аргументы ObservedEventsDescriptor:

DigitString

ParameterID: ds (0x0001)

Тип: строка символов плана нумерации (возможно пустая), возвращённая как quotedString

Возможные значения: последовательность символов от "0" до "9", от "A" до "F", а также модификатор "Z" длиной продолжительности.

Описание: часть текущей строки набора, как описано в 7.1.14, которая соответствует части или всей последовательности альтернативных событий, описанных в плане нумерации.

Метод окончания

ParameterID: Meth (0x0003)

Тип: перечисление

Возможные значения:

"UM" (0x0001) Однозначное соответствие

"PM" (0x0002) Частичное соответствие, завершение таймером просроченного или не соответствующего события

"FM" (0x0003) Полное соответствие, завершение таймером просроченного или не соответствующего события

Описание: показывает причину генерации события. См. процедуры в 7.1.14.

Е.6.3 Сигналы

Нет.

Е.6.4 Статистика

Нет.

Е.6.5 Процедуры

Обработка плана нумерации приводится в действие только при активизации дескриптора событий, содержащего завершающее событие плана нумерации, как это определяется в Е.6.2, и это завершающее событие содержит в требуемых действиях поле eventDM в повторных действиях, как определяется в 7.1.9. Другие аргументы, такие как KeepActive или встроенные события дескрипторов сигналов, также могут присутствовать в дескрипторе событий и не оказывать влияния на приведение в действие обработки плана нумерации.

Е.7 Комплект генератора тонов прохождения соединения

PackageID: cg, 0x0007

Версия: 1

Расширения: tonegen версия 1

Этот комплект определяет основные тоны прохождения соединения как сигналы и расширяет допустимые значения аргумента tl исполняемого тона в tonegen.

Е.7.1 Свойства

Нет.

Е.7.2 События

Нет.

Е.7.3 Сигналы

Тон ответа станции

SignalID: dt (0x0030)

Генерировать тон ответа станции. Шлюз имеет физические характеристики тона ответа станции.

Signal Тип: TimeOut

Длительность: Обеспечивается

Дополнительные аргументы:

Нет

Дополнительные значения:

dt (0x0030) определяется как идентификатор тона для исполняемого тона

Другие тоны данного комплекта определяются точно таким же образом. Приводится таблица со всеми именами сигналов и идентификаторами сигналов. Заметим, что каждый тон определяется как сигнал и как идентификатор тона, расширяющий таким образом комплект основной генерации тона.

Имя сигнала	Идентификатор сигнала /Идентификатор тона
Тон ответа станции	dt (0x0030)
Тон звонка	rt (0x0031)
Тон занятости	bt (0x0032)
Тон перегрузки	ct (0x0033)
Специальный информационный тон	sit (0x0034)
Тон предупреждения	wt (0x0035)
Тон опознавания таксофона	prt (0x0036)
Тон ожидания соединения	cw (0x0037)
Тон ожидания вызывающего абонента	cr (0x0038)

Е.7.4 Статистика

Нет.

Е.7.5 Процедуры

ПРИМЕЧАНИЕ – Требуемый набор идентификаторов тонов соответствует Рек. МСЭ-Т Е.180/Q.35. Определение назначения этих тонов приведено в Рек. МСЭ-Т Е.180/Q.35.

Е.8 Комплект обнаружения тонов прохождения соединения

PackageID: cd (0x0008)

Версия: 1

Расширения: tonedet версия 1

Данный комплект определяет основные тоны обнаружения прохождения соединения. Этот комплект расширяет возможные значения идентификатора тона в "обнаружено начало тона", "обнаружен конец тона" и "обнаружен длинный тон".

События.

Дополнительные значения

Значения идентификатора тона определяются для обнаруженного начала тона, обнаруженного конца тона и обнаруженного длинного тона с теми же значениями как в комплекте sg (комплект генерирования тонов прохождения соединения).

Требуемый набор идентификаторов тонов соответствует Рек. МСЭ-Т E.180/Q.35. Определение значения этих тонов приведено в Рек. МСЭ-Т E.180/Q.35.

E.8.1 Свойства

Нет.

E.8.2 События

События определяются также, как и в комплекте генератора тонов прохождения вызова (sg) для тонов, перечисленных в таблице E.7.3.

E.8.3 Сигналы

Нет.

E.8.4 Статистика

Нет.

E.8.5 Процедуры

Нет.

E.9 Комплект контроля аналоговой линии

PackageID: al, 0x0009

Версия: 1

Расширения: Нет

Этот паке определяет события и сигналы для аналоговой линии.

E.9.1 Свойства

Нет

E.9.2 События

трубка положена на рычаг

EventID: on (0x0004)

Обнаруживает перемещение трубки в положение "трубка положена". Всякий раз, когда приводится в действие дескриптор событий, запрашивающий контроль события "трубка положена", а линия уже находится в состоянии "трубка положена", то шлюз MG должен вести себя в соответствии с установкой аргумента "strict".

Аргументы EventDescriptor:

Строгое перемещение

ParameterID: strict (0x0001)

Тип: перечисление

Возможные значения: "exact" (0x00), "state" (0x01), "failWrong" (0x02)

"exact" означает, что должен распознаваться только действительный переход состояния трубки в положение "трубка положена";

"state" означает, что событие должно распознаваться либо, если обнаруживается переход состояния трубки, или, если трубка уже находится в состоянии "трубка положена";

"failWrong" означает, что, если трубка уже находится в положении "трубка положена", то команда не исполняется и сообщается об ошибке.

Аргументы ObservedEventsDescriptor:

Начальное состояние

ParameterID: init (0x0002)

Тип: булевый

Возможные значения:

"True" означает, что сообщено о событии, потому что дескриптор событий, содержащий это событие, был приведён в действие, когда линия уже была в положении "трубка положена";

"False" означает, что событие представляет действительный переход состояния в положение "трубка положена".

трубка снята с рычага

EventID: of (0x0005)

Обнаруживает перемещение трубки в положение "трубка снята". Всякий раз, когда приводится в действие дескриптор событий, запрашивающий контроль события "трубка снята", а линия уже находится в состоянии "трубка снята", то шлюз MG должен вести себя в соответствии с установкой аргумента "strict".

Аргументы EventDescriptor:

Строгое перемещение

ParameterID: strict (0x0001)

Тип: перечисление

Возможные значения: "exact" (0x00), "state" (0x01), "failWrong" (0x02)

"exact" означает, что должен распознаваться только действительный переход состояния трубки в положение "трубка снята";

"state" означает, что событие должно распознаваться либо, если обнаруживается переход состояния трубки, или, если трубка уже находится в состоянии "трубка снята";

"failWrong" означает, что, если трубка уже находится в положении "трубка снята", то команда не исполняется и сообщается об ошибке.

Аргументы ObservedEventsDescriptor:

Начальное состояние

ParameterID: init (0x0002)

Тип: булевый

Возможные значения:

"True" означает, что сообщено о событии, потому что дескриптор событий, содержащий это событие, был приведён в действие, когда линия уже была в положении "трубка снята";

"False" означает, что событие представляет действительный переход состояния в положение "трубка снята".

световой индикатор трубки мигает

EventID: fl, 0x0006

Обнаруживает мигание светового индикатора трубки. Мигание светового индикатора происходит, когда за положением "трубка положена" следует положение "трубка снята" в интервале между минимальной и максимальной длительностями.

Аргументы EventDescriptor:

Минимальная длительность

ParameterID: mindur (0x0004)

Тип: целое число в миллисекундах

Значение по умолчанию обеспечивается.

Максимальная длительность

ParameterID: maxdur (0x0005)

Тип: целое число в миллисекундах

Значение по умолчанию обеспечивается.

Аргументы ObservedEventsDescriptor:

Нет

Е.9.3 Сигналы

звонки

SignalID: ri, 0x0002

Включает посылку вызова в линию

Тип сигнала: перерыв

Duration: обеспечивается

Дополнительные аргументы:

Каденция

ParameterID: cad (0x0006)

Тип: список целых чисел, представляющих длительности попеременных сегментов включено и выключено, представляющих собой полный цикл посылки вызова, начинающийся с сегмента включено. Единицы в миллисекундах

Состояние по умолчанию фиксировано или обеспечивается. Ограниченная функция шлюзов MG может игнорировать значения каденции, которые они не способны формировать.

Частота

ParameterID: freq (0x0007)

Тип: целое число в Гц

Состояние по умолчанию фиксировано или обеспечивается. Ограниченная функция шлюзов MG может игнорировать значения частоты, которые они не способны формировать.

Е.9.4 Статистика

Нет

Е.9.5 Процедуры

Если контроллер MGC выполняет установку EventsDescriptor, содержащего событие перехода состояния трубки (трубка положена или трубка снята) с аргументом "strict" (0x0001), установленным в "failWrong", а трубка уже находится в положении, которое подразумевается переходом, исполнение команды, содержащей этот EventsDescriptor, нарушается. Шлюз MG ДОЛЖЕН включить в свой ответ код ошибки 540 (неожиданное начальное состояние трубки).

Е.9.6 Код ошибки

Этот комплект определяет новый код ошибки:

540 (неожиданное начальное состояние трубки)

Процедура для использования этого кода приведена в Е.9.5.

Е.10 Основной комплект связности

PackageID: ct (0x000a)

Версия: 1

Расширения: Нет

Данный комплект определяет события и сигналы для проверки связности. Проверка связности включает обеспечение либо петли обратной связи или функциональности приёмопередатчика.

Е.10.1 Свойства

Нет.

Е.10.2 События

Завершение

EventID: cmp, 0x0005

Это событие выявляет завершение проверки в проверке связности.

Аргументы EventDescriptor:

Нет

Аргументы ObservedEventsDescriptor:

Результат

ParameterID: res (0x0008)

Тип: перечисление

Возможные значения: success (0x0001), failure (0x0000)

Е.10.3 Сигналы

Проверка связности

SignalID: ct (0x0003)

Начинает передачу тона проверки связности на окончание, к которому он применяется.

Тип сигнала: перерыв

Значение по умолчанию обеспечивается

Дополнительные аргументы:

Нет

Ответ

SignalID: rsp (0x0004)

Сигнал используется для ответа на проверку связности. Для дальнейших пояснений см Е.10.5.

Тип сигнала: включено/выключено

Длительность по умолчанию обеспечивается

Дополнительные аргументы:

Нет

Е.10.4 Статистика

Нет.

Е.10.5 Процедуры

Когда контроллер MGC хочет начать проверку связности, он посылает в шлюз MG команду, содержащую:

- дескриптор сигнала с сигналом ct; и
- дескриптор событий, содержащий событие smp.

По приёму команды, содержащей сигнал ct и событие smp, шлюз MG начинает тон проверки связности определённого окончания. Если обнаруживается обратный тон, а любые другие условия выполняются прежде, чем сигнал прервётся, то должно генерироваться событие smp, со значением аргумента результата, равным успеху. Во всех других случаях событие smp должно генерироваться со значением аргумента результата, равным неудаче.

Когда контроллер MGC хочет, чтобы шлюз MG ответил на проверку связности, он посылает в шлюз MG команду, содержащую дескриптор сигналов, с сигналом rsp. По приёму команды с сигналом rsp, шлюз MG либо использует петлю обратной связи или (для 2-проводных линий) ожидает приёма тона проверки связности. В случае петли обратной связи любая входящая информация должна отражаться обратно, как исходящая информация. В 2-проводном случае, соответствующий ответный тон должен посылаться когда бы не был получен соответствующий тон проверки связности. Контроллер MGC определяют когда удалить сигнал rsp.

Когда проверка связности выполняется на окончании, на этом окончании не должно быть активных устройств подавления эхо и кодеков.

Выполнение обеспечения голосового пути, как составной части проверки связности, обеспечивается двусторонним соглашением между операторами сетей.

(Информационное примечание) Пример тонов и деталей процедуры проверки приведен в разделах 7 и 8/Q.724, 2.1.8/Q.764, а также Рек. МСЭ-Т Q.1902.4.

Е.11 Комплект сети

PackageID: nt (0x000b)

Версия: 1

Расширения: Нет

Этот комплект определяет свойства окончаний сети, не зависящих от типа сети.

Е.11.1 Свойства

Максимальный буфер джиттера

PropertyID: jit (0x0007)

Это свойство устанавливает максимальный размер буфера джиттера.

Тип: целое число в миллисекундах

Возможные значения: Это свойство определяется в миллисекундах.

Определено в: LocalControlDescriptor

Характеристики: для чтения/записи

Е.11.2 События

отказ сети

EventID: netfail, 0x0005

Окончание формирует это событие после обнаружения отказа вследствие внешних или внутренних причин сети.

Аргументы EventDescriptor:

Нет

ObservedEventsDescriptor parameters:

причина

ParameterID: cs (0x0001)

Тип: строка

Возможные значения: любая текстовая строка

Этот аргумент может быть включён со случаем отказа для обеспечения диагностической информации о причине отказа.

Предупреждение о качестве

EventID: qualert, 0x0006

Это свойство позволяет шлюзу MG указать на потерю качества соединения в сети. Шлюз MG может это делать путём измерения потерь пакетов, разброса времени прихода пакетов, задержки распространения, а затем обратить внимание на потерю качества, используя процент потерь качества.

EventDescriptor parameters:

Порог

ParameterId: th (0x0001)

Тип: целое число

Возможные значения: 0 to 99

Описание: порог измеренного процента потерь качества, вычислен на основе предоставленного метода, который может принять во внимание потери пакетов, джиттер и, например, задержку. Событие инициализируется, когда результат вычисления превысит порог.

Аргументы ObservedEventsDescriptor:

Порог

ParameterId: th (0x0001)

Тип: целое число

Возможные значения: 0 to 99

Описание: измеренный процент потери качества вычислен на основе предоставленного метода, который может принять во внимание потери пакетов, джиттер и, например, задержку.

Е.11.3 Сигналы

Нет.

Е.11.4 Статистика

Длительность

StatisticsID: dur (0x0001)

Описание: обеспечивает длительность времени, в течение которого окончание находится в контексте.

Тип: двойной, в миллисекундах

Отправленный октет

StatisticID: os (0x0002)

Тип: двойной

Возможные значения: любое 64-битовое целое число

Принятый октет

StatisticID: or (0x0003)

Тип: двойной

Возможные значения: любое 64-битовое целое число

Е.11.5 Процедуры

Нет.

Е.12 RTP Package

PackageID: rtp (0x000c)

Версия: 1

Расширения: Сетевой комплект, версия 1

Этот комплект используется для поддержки мультимедийного перемещения данных, основанного на пакетах, посредством Транспортного протокола реального времени (RTP) (IETF RFC 1889).

Е.12.1 Свойства

Нет.

Е.12.2 События

Перемещение полезной нагрузки

EventID: pltrans, 0x0001

Это событие обнаруживает и уведомляет при наличии перемещения формата полезной нагрузки протокола RTP из одного формата в другой.

Аргументы EventDescriptor:

Нет

Аргументы ObservedEventsDescriptor:

ParameterName: rtppayload

ParameterID: rtppltype, 0x01

Тип: список перечисленных типов.

Возможные значения: Метод кодирования должен определяться путём использования одного или нескольких правильных имён кодирования, как определено в профиле AV протокола RTP или зарегистрировано IANA.

Е.12.3 Сигналы

Нет.

Е.12.4 Статистика

Отправленные комплекты

StatisticID: ps (0x0004)

Тип: двойной

Возможные значения: любое 64- битовое целое число

Принятые комплекты

StatisticID: pr (0x0005)

Тип: двойной

Возможные значения: любое 64- битовое целое число

Потеря пакета

StatisticID: pl (0x0006)

Описывает текущий коэффициент потерь пакетов в потоке RTP, как определено в IETF RFC 1889. Потеря пакетов выражается как процентное значение: число потерянных пакетов в интервале между двумя отчётами о приёме, делённое на количество комплектов, которые ожидалось в течение данного интервала.

Тип: двойной

Возможные значения: а 32- битовое целое число и 32- битовая дробь.

Джиттер

StatisticID: jit (0x0007)

Запрашивает текущее значение разброса времени прихода пакетов в потоке RTP, как определено в IETF RFC 1889. Джиттер измеряет отклонение во временном разбросе пакетов данных RTP.

Задержка

StatisticID:delay (0x0008)

Запрашивает текущее значение задержки распространения пакета, выраженное в единицах timestamp. То же самое, что средняя задержка.

Е.12.5 Процедуры

Нет.

Е.13 Комплект TDM канала

PackageID: tdmc (0x000d)

Версия: 1

Расширения: Сетевой комплект, версия 1

Этот комплект может использоваться любым окончанием, поддерживающим усиление и управление эхоподавлением. Вначале предназначалось для применения на каналах с временным уплотнением (TDM), но может использоваться более широко.

Новые версии или расширения этого комплекта должны принимать во внимание применение не на каналах TDM.

Е.13.1 Свойства

Подавление эха

PropertyID: ec (0x0008)

Тип: булевый

Возможные значения:

"on" (когда запрашивается подавление эха) и

"off" (когда оно выключается)

Значение по умолчанию обеспечивается.

Определено в: LocalControlDescriptor

Характеристики: для чтения/записи

Управление усилением

PropertyID: gain (0x000a)

Управление усилением или использование адаптации уровня сигнала и снижения уровня шума используется для подстройки уровня сигнала. Однако необходимо, например при модемных соединениях, выключить эту функцию.

Тип: целое число

Возможные значения:

Аргумент управления усилением может быть либо определённым как "автоматический" (0xffffffff) или как явное число децибелл усиления (любое другое целое значение). Значение по умолчанию обеспечивается в шлюзе MG.

Определено в: LocalControlDescriptor

Характеристики: для чтения/записи

E.13.2 События

Нет.

E.13.3 Сигналы

Нет.

E.13.4 Статистика

Нет.

E.13.5 Процедуры

Нет.

Приложение I

Пример прохождения соединения

Все те, кто реализует H.248.1, должны перед её реализацией внимательно прочитать нормативную часть данной Рекомендации. Примеры, приведенные в данном приложении, не должны использоваться в качестве автономных объяснений того, как создать сообщения протокола.

В примерах данного приложения для кодирования дескрипторов потоков Local и Remote используется SDP. SDP определяется в документе IETF RFC 2327. При наличии каких-либо различий между SDP в примерах и документом IETF RFC 2327, документ RFC должен использоваться в качестве верного документа. Используемые аудио параметры определяются в документе IETF RFC 1890, а другие- зарегистрированы IANA. Например, закон A Рекомендации МСЭ-Т G.711 называется в SDP законом PCMA и ему присвоен параметр 0. G.723.1 носит название G723 и ему присвоен параметр 4; H.263 называется H263 и ему присвоен параметр 34. См. также <http://www.iana.org/assignments/rtp-parameters>.

I.1 Соединение частного шлюза с частным шлюзом

Этот пример сценария иллюстрирует использование элементов протокола для установления соединения частного шлюза с частным шлюзом по IP сети. Для простоты этот пример подразумевает, что оба частных шлюза управляются одним и тем же контроллером медиашлюза.

I.1.1 Программирование окончаний аналоговых каналов частного шлюза в режиме незанятости

Нижеследующее иллюстрирует вызовы API со стороны контроллера медиашлюза и медиашлюзов для получения окончания, запрограммированного, в данном сценарии, для режима незанятости. Оба исходящий и входящий медиашлюзы имеют незанятые окончания AnalogLine, запрограммированные на поиск событий, инициализирующих соединения (т.е. снятие трубки), путём использования команды Modify с соответствующими параметрами. Нулевой контекст используется для указания того, что окончания пока ещё не участвуют в

контексте. Окончание ROOT используется для указания на целый медиашлюз, а не на окончание внутри шлюза MG.

В этом примере шлюз MG1 имеет адрес IP 124.124.124.222, шлюз MG2 – адрес IP 125.125.125.111, а контроллер MGC is 123.123.123.4. Для всех трёх устройств адрес по умолчанию порта Megaco 55555.

1) Используя команду ServiceChange, шлюз MG регистрируется в контроллере MGC:

Шлюз MG1 контроллеру MGC:

```
MEGACO/1 [124.124.124.222]
Transaction = 9998 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Restart,
      ServiceChangeAddress=55555, Profile=ResGW/1}
    }
  }
}
```

2) Контроллер MGC посылает ответ:

Контроллер MGC шлюзу MG1:

```
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
  Context = - {ServiceChange = ROOT {
    Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}
```

3) Контроллер MGC программирует окончание в нулевом контексте. Идентификатор terminationId равен A4444, идентификатор streamId - 1, идентификатор requestId в дескрипторе Events равен 2222. Идентификатор mId является идентификатором отправителя этого сообщения, в этом случае, это является адресом IP и порта [123.123.123.4]:55555. Для этого потока установлен режим SendReceive. "al" является комплектом контроля аналоговой линии.

Контроллер MGC шлюзу MG1:

```
MEGACO/1 [123.123.123.4]:55555
Transaction = 9999 {
  Context = - {
    Modify = A4444 {
      Media { Stream = 1 {
        LocalControl {
          Mode = SendReceive,
          tdmc/gain=2, ; in dB,
          tdmc/ec=on
        },
        Local {
          v=0
          c=IN IP4 $
          m=audio $ RTP/AVP 0
          a=fmtp:PCMU VAD=X-NNVAD ; special voice activity
          ; detection algorithm
        }
      }
    },
    Events = 2222 {al/of}
  }
}
```

Текст плана нумерации мог быть предварительно загружен в шлюз MG. Его функция должна состоять в ожидании снятия трубки, включения непрерывного тонального сигнала и начала сбора цифр DTMF. Однако, в этом примере мы используем план нумерации, загруженный после обнаружения снятия трубки [шаг 5) ниже].

Заметим, что вложенный EventsDescriptor может быть использован для объединения шагов 3) и 4) с шагами 8) и 9), пренебрегая шагами 6) и 7).

4) Шлюз MG1 принимает команду Modify с таким ответом:

```
Шлюз MG1 контроллеру MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 9999 {
  Context = - {Modify = A4444}
}
```

5) Аналогичный обмен происходит между шлюзом MG2 и контроллером MGC, который приводит к режиму ожидания окончания, обозначаемому A5555.

I.1.2 Сбор исходных цифр и инициализация окончания

Следующие построения основаны на условиях, приведенных выше. Они иллюстрируют транзакции от контроллера медиашлюза и исходящего медиашлюза (MG1) для получения исходящего окончания (A4444) посредством шагов сбора цифр, необходимых для начала соединения с входящим медиашлюзом (MG2).

6) Шлюз MG1 обнаруживает событие снятия трубки от пользователя 1 и сообщает об этом контроллеру медиашлюза посредством команды Notify.

```
Шлюз MG1 контроллеру MGC:
MEGACO/1 [124.124.124.222]:55555
Transaction = 10000 {
  Context = - {
    Notify = A4444 {ObservedEvents =2222 {
      19990729T22000000:a1/of}}
  }
}
```

7) Подтверждение команды Notify.

```
Контроллер MGC шлюзу MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 10000 {
  Context = - {Notify = A4444}
}
```

8) Контроллер MGC изменяет окончание для генерации тонального сигнала набора номера, для поиска цифр в соответствии с планом нумерации Dialplan0, а также для поиска события возврата трубки.

```
Контроллер MGC шлюзу MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10001 {
  Context = - {
    Modify = A4444 {
      Events = 2223 {
        a1/on, dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
(0| 00|[1-7]xxx|8xxxxxxxx|Fxxxxxxxx|Exx|91xxxxxxxxxxx|9011x.)}
    }
  }
}
```

```
}
```

9) Подтверждение команды Modify.

```
Шлюз MG1 контроллеру MGC:  
MEGACO/1 [124.124.124.222]:55555  
Reply = 10001 {  
  Context = - {Modify = A4444}  
}
```

- 10) Затем цифры накапливаются шлюзом MG1 в порядке их набора пользователем 1. После обнаружения первой цифры тональный сигнал набора намера прекращается. При подходящем совпадении накопленных цифр с запрограммированным планом нумерации для A4444, в контроллер медиашлюза посылается другая команда Notify.

Шлюз MG1 контроллеру MGC:

```
MEGACO/1 [124.124.124.222]:55555  
Transaction = 10002 {  
  Context = - {  
    Notify = A4444 {ObservedEvents =2223 {  
      19990729T22010001:dd/ce{ds="916135551212",Meth=FM}}}  
  }  
}
```

11) Подтверждение команды Notify.

```
Контроллер MGC шлюзу MG1:  
MEGACO/1 [123.123.123.4]:55555  
Reply = 10002 {  
  Context = - {Notify = A4444}  
}
```

- 12) Теперь контроллер проводит анализ цифр и определяет, какое соединение от MG1 к MG2 необходимо выполнить. Оба окончания с временным разделением A4444, а также окончание RTP присоединяются к новому контексту в шлюзе MG1. Режимом является ReceiveOnly, т.к. значения дескриптора Remote ещё не определены. Кодеки находятся в порядке выбора, предпочтительном для контроллера MGC.

```
Контроллер MGC шлюзу MG1:  
MEGACO/1 [123.123.123.4]:55555  
Transaction = 10003 {  
  Context = $ {  
    Add = A4444,  
    Add = $ {  
      Media {  
        Stream = 1 {  
          LocalControl {  
            Mode = ReceiveOnly,  
  
            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 – Контроллер MGC устанавливает свои предпочтительные значения как последовательность блоков SDP в дескрипторе Local. Шлюз MG заполняет дескриптор Local в Reply.

- 13) Шлюз MG1 подтверждает новое окончание и заполняет адрес IP дескриптора Local и порт UDP. Он также делает выбор для кодека, основанный на предпочтениях контроллера MGC в Local. Шлюз MG1 устанавливает порт RTP на 2222.

```

MEGACO/1 [124.124.124.222]:55555
Reply = 10003 {
  Context = 2000 {
    Add = A4444,
    Add=A4445{
      Media {
        Stream = 1 {
          Local {
v=0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
a=recvonly
          } ; RTP profile for G.723.1 is 4
        }
      }
    }
  }
}
}

```

- 14) Теперь контроллер MGC будет связывать A5555 с новым контекстом в шлюзе MG2 и образовывать поток RTP (т.е. будет присваиваться A5556) соединении SendReceive до исходящего пользователя, т.е. пользователя 1. Кроме того, контроллер MGC устанавливает вызов на A5555.

```

Контроллер MGC шлюзу MG2:
MEGACO/1 [123.123.123.4]:55555
Transaction = 50003 {
  Context = $ {
    Add = A5555 { Media {
      Stream = 1 {
        LocalControl {Mode = SendReceive} }},
    Events=1234{al/of},
    Signals {al/ri}
  },
  Add = $ {Media {
    Stream = 1 {
      LocalControl {
        Mode = SendReceive,
        nt/jit=40 ; in ms
      },
      Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
      },
      Remote {
v=0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
    } ; RTP profile for G.723.1 is 4
      }
    }
  }
}

```

```

    }
  }
}

```

- 15) Это подтверждение. Номер порта потока отличается от номера управляющего порта. В этом случае это 1111 (в случае SDP).

```

Шлюз MG2 контроллеру MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 50003 {
  Context = 5000 {
    Add = A5555,
    Add = A5556{
      Media {
        Stream = 1 {
          Local {
v=0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
}
          } ; RTP profile for G.723.1 is 4
        }
      }
    }
  }
}

```

- 16) Теперь вышеуказанные IPAddr и UDPport должны быть направлены шлюзу MG1.

```

Контроллер MGC шлюзу MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {cg/rt}
    },
    Modify = A4445 {
      Media {
        Stream = 1 {
          Remote {
v=0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
}
          } ; RTP profile for G.723.1 is 4
        }
      }
    }
  }
}

```

```

Шлюз MG1 контроллеру MGC:
MEGACO/1 [124.124.124.222]:55555
Reply = 10005 {
  Context = 2000 {Modify = A4444, Modify = A4445}
}

```

- 17) Теперь два шлюза соединены и пользователь 1 слышит RingBack. Теперь шлюз MG2 ожидает до тех пор, пока пользователь 2 снимет трубку и тогда установится двустороннее сообщение.

```

От шлюза MG2 к контроллеру MGC:
MEGACO/1 [125.125.125.111]:55555

```

```
Transaction = 50005 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1234 {
      19990729T22020002:al/of}}
  }
}
```

От контроллера MGC к шлюзу MG2:

```
MEGACO/1 [123.123.123.4]:55555
Reply = 50005 {
  Context = - {Notify = A5555}
}
```

От контроллера MGC к шлюзу MG2:

```
MEGACO/1 [123.123.123.4]:55555
Transaction = 50006 {
  Context = 5000 {
    Modify = A5555 {
      Events = 1235 {al/on},
      Signals { } ; to turn off ringing
    }
  }
}
```

От шлюза MG2 к контроллеру MGC:

```
MEGACO/1 [125.125.125.111]:55555
Reply = 50006 {
  Context = 5000 {Modify = A4445}
}
```

18) Изменение режима шлюза MG1 на SendReceive и прекращение ringback.

```
Контроллер MGC к шлюзу MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10006 {
  Context = 2000 {
    Modify = A4445 {
      Media {
        Stream = 1 {
          LocalControl {
            Mode=SendReceive
          }
        }
      }
    },
    Modify = A4444 {
      Signals { }
    }
  }
}
```

от шлюза MG1 к контроллеру MGC:

```
MEGACO/1 [124.124.124.222]:55555
Reply = 10006 {
  Context = 2000 {Modify = A4445, Modify = A4444}}
```

19) Контроллер MGC решает проверить окончание RTP на MG2.

```
MEGACO/1 [123.123.123.4]:55555
Transaction = 50007 {
  Context = - {AuditValue = A5556{
```

```

    Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
  }
}

```

20) Шлюз MG2 отвечает.

```

MEGACO/1 [125.125.125.111]:55555
Reply = 50007 {
  Context = - {
    AuditValue = A5556 {
      Media {
        TerminationState { ServiceStates = InService,
          Buffer = OFF },
        Stream = 1 {
          LocalControl { Mode = SendReceive,
            nt/jit=40 },
          Local {
v=0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
a=ptime:30
          },
          Remote {
v=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, ; packets sent
        nt/os=62300, ; octets sent
        rtp/pr=700, ; packets received
        nt/or=45100, ; octets received
        rtp/pl=0.2, ; % packet loss
        rtp/jit=20,
        rtp/delay=40 } ; avg latency
    }
  }
}

```

21) Когда контроллер MGC получает от одного из шлюзов MG сигнал о том, что трубка положена, он разрывает соединение. В этом примере пользователь шлюза MG2 кладёт трубку первым.

От шлюза MG2 к контроллеру MGC:

```

MEGACO/1 [125.125.125.111]:55555
Transaction = 50008 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1235 {
      19990729T24020002:al/on}
    }
  }
}

```

От контроллера MGC к шлюзу MG2:

```

MEGACO/1 [123.123.123.4]:55555
Reply = 50008 {
  Context = - {Notify = A5555}
}

```

}

- 22) Теперь контроллер MGC посылает на оба шлюза MG команду Subtract для разъединения соединения. Здесь показываются разъединения только в шлюзе MG2. У каждого окончания имеется свой собственный набор собираемой им статистической информации. Контроллеру MGC может быть нет необходимости запрашивать для возврата оба набора. A5555 представляет собой физическое окончание, а A5556 является окончанием RTP.

От контроллера MGC к шлюзу MG2:

```
MEGACO/1 [123.123.123.4]:55555
Transaction = 50009 {
  Context = 5000 {
    Subtract = A5555 {Audit{Statistics}},
    Subtract = A5556 {Audit{Statistics}}
  }
}
```

От шлюза MG2 к контроллеру MGC:

```
MEGACO/1 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
    Subtract = A5555 {
      Statistics {
        nt/os=45123, ; Octets Sent
        nt/dur=40 ; in seconds
      }
    },
    Subtract = A5556 {
      Statistics {
        rtp/ps=1245, ; packets sent
        nt/os=62345, ; octets sent
        rtp/pr=780, ; packets received
        nt/or=45123, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=27,
        rtp/delay=48 ; average latency
      }
    }
  }
}
```

- 23) Теперь контроллер MGC устанавливает оба шлюза MG1 и MG2 в состояние готовности для обнаружения следующего события снятия трубки. См. шаг 1). Заметим, что оно может быть состоянием окончания по умолчанию в нулевом контексте и, если это имеет место, то нет необходимости отправлять сообщения от контроллера MGC к шлюзу MG. Как только окончание возвратится в нулевой контекст, оно идёт обратно к значениям окончания по умолчанию для данного окончания.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и средства передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	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	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура и аспекты межсетевых протоколов (IP)
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи