МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т H.225.0

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

(07/2003)

СЕРИЯ H: АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб – Мультиплексирование и синхронизация при передаче

Протоколы сигнализации о соединении и пакетирование потоков носителей для мультимедийных систем связи на основе пакетов

Рекомендация МСЭ-Т H.225.0

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ H

АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	H.100–H.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ СЛУЖБ	
Общие положения	H.200–H.219
Мультиплексирование и синхронизация при передаче	**H.220–H.229**
Системные аспекты	H.230–H.239
Процедуры связи	H.240–H.259
Кодирование подвижных видеоизображений	H.260–H.279
Сопутствующие системные аспекты	H.280–H.299
СИСТЕМЫ И ОКОНЕЧНОЕ ОБОРУДОВАНИЕ ДЛЯ АУДИОВИЗУАЛЬНЫХ СЛУЖБ	H.300–H.399
ДОПОЛНИТЕЛЬНЫЕ УСЛУГИ ДЛЯ МУЛЬТИМЕДИЙНЫХ СЛУЖБ	H.450–H.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	H.500–H.509
Мобильность для мультимедийных систем и служб серии H	H.510–H.519
Приложения и службы мобильной мультимедийной совместной работы	H.520–H.529
Безопасность для мобильных мультимедийных систем и служб	H.530–H.539
Безопасность для приложений и служб мобильной мультимедийной совместной работы	H.540–H.549
Процедуры мобильного взаимодействия	H.550–H.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	H.560–H.569
ШИРОКОПОЛОСНЫЕ МУЛЬТИМЕДИЙНЫЕ СЛУЖБЫ И МУЛЬТИМЕДИЙНЫЕ СЛУЖБЫ В РЕЖИМЕ TRIPLE-PLAY	
Предоставление широкополосных мультимедийных услуг по VDSL	H.610–H.619

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

Рекомендация МСЭ-Т H.225.0

Протоколы сигнализации о соединении и пакетирование потоков носителей для мультимедийных систем связи на основе пакетов

Резюме

Настоящая Рекомендация посвящена техническим требованиям к узкополосным визуальным телефонным службам, определенным в Рекомендациях серий H.200 и F.720, для ситуаций, когда тракт передачи содержит одну или несколько пакетных сетей, каждая из которых настроена и управляется для обеспечения негарантированного Качества обслуживания, КО (Quality of Service, QoS), которое не эквивалентно КО сети У-ЦСИС, и поэтому в терминалах необходимо предусматривать механизмы защиты или восстановления в дополнение к тем, которые предписаны в Рекомендации МСЭ-Т H.320. Отмечается, что Рекомендация МСЭ-Т H.322 указывает на использование некоторых других локальных сетей (LAN), которые способны обеспечить рабочие характеристики нижележащего уровня, не предполагавшиеся в Рекомендациях МСЭ-Т H.323 и H.225.0.

В настоящей Рекомендации описывается, как можно управлять информацией аудио, видео, данных и управления в пакетной сети, чтобы обеспечить переговорные службы в оборудовании H.323.

В Приложении G описывается метод, позволяющий определять адрес между административными доменами в системе H.323 с целью организации соединений между такими административными доменами. Сам административный домен представляется другим административным доменам через некоторый тип логического элемента, называемый пограничным элементом.

Продукты, претендующие на соответствие с версией 5 (настоящей версией) H.225.0, должны выполнять все обязательные требования этой Рекомендации. Продукты версии 5 могут указываться сообщениями H.225.0, содержащими значение **protocolIdentifier**: {itu-t (0) recommendation (0) h (8) 2250 version (0) 5}.

Источник

Рекомендация МСЭ-Т H.225.0 утверждена 14 июля 2003 года 16-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т A.8.

ПРЕДИСЛОВИЕ

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

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

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

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

ПРИМЕЧАНИЕ

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

Соответствие положениям данной Рекомендации является добровольным делом. Однако в Рекомендации могут содержаться определенные обязательные положения (для обеспечения, например, возможности взаимодействия или применимости), и тогда соответствие данной Рекомендации достигается в том случае, если выполняются все эти обязательные положения. Для выражения требований используются слова "shall" ("должен", "обязан") или некоторые другие обязывающие термины, такие как "must" ("должен"), а также их отрицательные эквиваленты. Использование таких слов не предполагает, что соответствие данной Рекомендации требуется от каждой стороны.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ.

СОДЕРЖАНИЕ

Рекомендация МСЭ-Т H.225.0

Протоколы сигнализации о соединении и пакетирование потоков носителей для мультимедийных систем связи на основе пакетов

МСЭ-Т,

учитывая

широкое признание и возрастающее применение Рекомендации МСЭ-Т H.320 для служб видеотелефона и видеоконференции по сетям, соответствующим характеристикам узкополосной цифровой сети с интеграцией служб (У-ЦСИС), определенным в Рекомендациях серии I,

понимая

желательность и выгодность возможностей переносить вышеуказанные службы, полностью или частично, по локальным сетям, которые поддерживают также возможность взаимодействия с терминалами H.320,

и отмечая

характеристики и качество услуг многих типов локальных сетей, представляющих потенциальный интерес,

рекомендует,

чтобы для обеспечения этих возможностей использовались системы и оборудование, отвечающие требованиям Рекомендации МСЭ-Т H.322 или Рекомендации МСЭ-Т H.323.

1 Предмет рассмотрения

В этой Рекомендации описываются средства, с помощью которых информация аудио, видео, данных и управления объединяется, кодируется и пакетируется для транспортировки между устройствами H.323 по пакетной сети. Это охватывает использование шлюза H.323, который, в свою очередь, может быть соединен с терминалами H.320, H.324 или H.310/H.321 по сетям У-ЦСИС, Коммутируемой телефонной сети общего пользования (КТСОП) или широкополосной ЦСИС (Ш-ЦСИС) соответственно. Описания устройств и процедур приведены в Рекомендации МСЭ-Т H.323, а в настоящей Рекомендации даются протоколы и форматы сообщений. Возможна также связь через шлюз H.323 к шлюзу H.322 для локальных сетей с гарантированным качеством обслуживания (КО) и, следовательно, к конечным точкам H.322.

Настоящая Рекомендация предназначена для организации работы через разные пакетные сети, в том числе IEEE 802.3, Token Ring и др. Следовательно, эта Рекомендация определяется как находящаяся выше транспортного уровня, такого как TCP/IP/UDP, SPX/IPX и т. п. Специфичные профили для комплектов конкретных транспортных протоколов приведены в Добавлении IV. ***Таким образом, предмет рассмотрения связей H.225.0 находится между объектами H.323, включенными в одну и ту же пакетную сеть и использующими один и тот же транспортный протокол***. Эта пакетная сеть может быть одиночным сегментом или кольцом, либо она, по логике, может быть сетью передачи данных предприятия, состоящей из нескольких пакетных сетей, соединенных мостами или маршрутизаторами для создания одной взаимосвязанной сети. Следует подчеркнуть, что работа терминалов H.323 по всей сети Интернет или даже по нескольким соединенным пакетным сетям может привести к плохим рабочим характеристикам. Возможные средства, с помощью которых можно было бы гарантировать качество обслуживания по этой пакетной сети или по Интернету, вообще говоря, не входят в предмет рассмотрения этой Рекомендации. Однако эта Рекомендация предлагает для пользователей оборудования H.323 средства определения, что перегрузка пакетной сети вызвала трудности с качеством, а также предлагает процедуры исправляющих действий. Отмечается также, что использование нескольких шлюзов H.323, соединенных через сеть ЦСИС общего пользования, является прямым методом повышения качества обслуживания.

Рекомендация МСЭ-Т H.323 и эта Рекомендация предназначены для превращения соединений H.320 и H.221 в конференции, организуемые в среде пакетной сети с негарантированным КО. С учетом этого, первичной моделью конференции[1] является конференция с объемом в диапазоне от нескольких участников до нескольких тысяч в отличие от широкомасштабных вещательных операций, со строгим управлением допуском и слабым управлением конференцией.

В этой Рекомендации используются Транспортный протокол реального времени/Транспортный управляющий протокол реального времени (RTP/RTCP) для пакетирования потока носителей и синхронизации для всех нижележащих пакетных сетей (см. Приложения A, B и C). Отметьте, пожалуйста, что использование RTP/RTCP, определенных в этой Рекомендации, не связано каким-либо образом с использованием TCP/IP/UDP. В этой Рекомендации предполагается модель соединения, в которой начальная сигнализация по не-RTP-транспортному адресу используется для установления соединения и согласования возможностей (см. Рекомендации МСЭ-Т H.323 и H.245), за которыми следует установление одного или нескольких соединений RTP/RTCP. Эта Рекомендация содержит детали использования RTP/RTCP.

Согласно Рекомендации МСЭ-Т H.221 сигналы аудио, видео, данных и управления мультиплексируются в одно или несколько синхронизированных физических соединений по сети с коммутацией каналов (SCN). На стороне соединения H.323, связанной с пакетной сетью, ни одно из этих понятий не применимо. Нет необходимости от стороны SCN концепции H.221 переносить соединение P × 64 кбит/с, например, 2 по 64 кбит/с, 3 по 64 кбит/с и т. д. Поэтому на стороне пакетной сети, например, имеются только вызовы одиночных "соединений" с максимальной скоростью 128 кбит/с, а не соединения с фиксированными скоростями 2 × 64 кбит/с. В другом примере в пакетной сети имеются вызовы одиночных "соединений" с максимальной скоростью 384 кбит/с, которые взаимодействуют на стороне SCN[2] с 6 каналами по 64 кбит/с. Основная причина такого подхода состоит в желании разместить сложность не в терминале, а в шлюзе, а также в желании избежать проникновения в пакетную сеть особенностей H.320, которые тесно связаны с ЦСИС, если это не требуется.

Обычно терминалы H.323 прямо не осведомлены о скорости переноса терминала H.320 при взаимодействии через шлюз H.323; вместо этого шлюз использует сообщения **FlowControlCommand** из H.245 для ограничения скорости передачи носителей в каждом используемом логическом канале до той, которая разрешена мультиплексором H.221. Шлюз может разрешать для видео на стороне пакетной сети скорости, существенно меньшие скоростей на стороне SCN (или наоборот), путем использования функции снижения скорости и кадров заполнения H.261; детали этих операций не входят в предмет рассмотрения Рекомендации МСЭ-Т H.323 и настоящей Рекомендации. Заметим, что терминал H.323 косвенно осведомлен о скоростях переноса H.320 посредством полей максимальной битовой скорости видео из Рекомендации МСЭ-Т H.245 и не будет передавать на скоростях, превышающих эти скорости.

Настоящая Рекомендация разработана так, чтобы была возможность совместной работы шлюза H.323 с терминалами H.320 (1990 г.), H.320 (1993 г.) и H.320 (1996 г.). Однако некоторые особенности настоящей Рекомендации могут быть направлены на разрешенные улучшенные операции с будущими версиями Рекомендации МСЭ-Т H.320. Возможно также, что качество обслуживания на стороне H.320 может измениться на основе свойств и возможностей шлюза H.323 (см. рисунок 1).

---

[1] Факультативная модель конференции только для вещания находится на рассмотрении; по необходимости модель вещания не обеспечивает строгого контроля допуска или управления конференцией.

[2] Отметим, что скорости видео и данных на стороне LAN должны соответствовать скоростям видео и данных мультиплексора H.320 на стороне SCN; скорости аудио и управления не требуют соответствия. Сформулировав другой путь, кто-либо мог бы нормально ожидать, что шлюз LAN/SCN, используя управление потоком H.245, будет заставлять скорости видео и данных соответствовать мультиплексору SCN H.221. Однако, так как аудио обычно может транскодироваться в шлюзе, кто-либо часто будет находить, что скорость аудио LAN и скорость SCN не совпадают. Не будет также надежды, что битовая скорость H.221 для управления (800 бит/с) будет обычно совпадать с битовой скоростью H.245 на стороне LAN. Отметим также, что скорость LAN может отставать от скорости SCN для видео или данных либо для обоих, но она не может превышать максимальную сумму, которая подходит для мультиплексора стороны SCN.

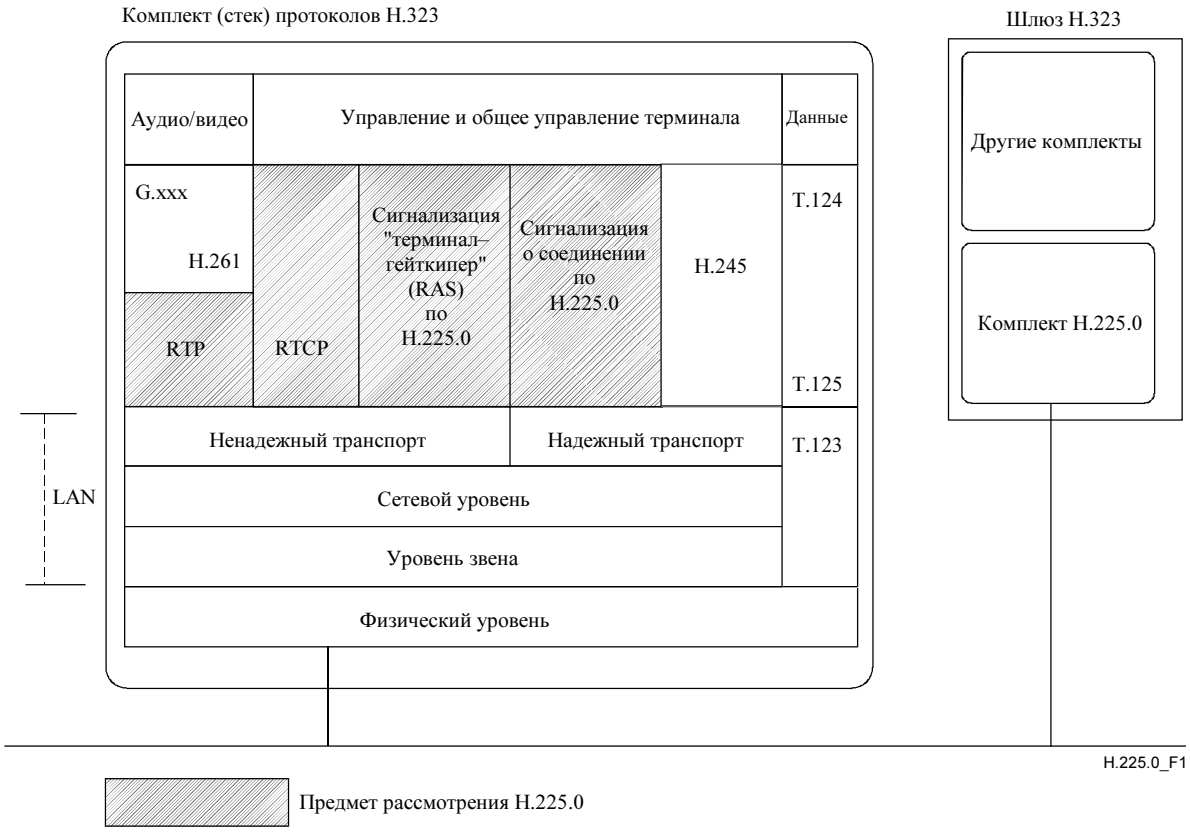


Рисунок 1/H.225.0 – Предмет рассмотрения H.225.0

Общим подходом в этой Рекомендации является обеспечение средств синхронизации пакетов, которые используют нижележащие средства пакетной сети/транспорта. В этой Рекомендации не требуется, чтобы все носители и управление были смешаны в одном потоке, который потом пакетируется. Механизмы образования кадров из Рекомендации МСЭ-Т H.221 не используются по следующим причинам:

- Неиспользование H.221 позволяет каждый носитель принимать с разной подходящей обработкой ошибок.
- H.221 относительно чувствительна к потере случайных групп битов; пакетирование позволяет получить лучшую ошибкоустойчивость в среде пакетной сети.
- Сообщения сигнализации о соединении по H.245 и H.225.0 могут передаваться по надежным звеньям, которые обеспечивает пакетная сеть.
- Гибкость и производительность H.245 сравнимы с H.242.

2 Библиографические ссылки

Нижеследующие Рекомендации МСЭ-Т и другие источники содержат положения, которые, путем ссылок на них в этом тексте, образуют положения настоящей Рекомендации. В момент публикации были действительны указанные издания. Все Рекомендации и другие источники подвергаются пересмотру; поэтому всем пользователям этой Рекомендации следует рассматривать возможность применения самых последних изданий перечисленных ниже Рекомендаций и других источников. Список текущих действующих Рекомендаций МСЭ-Т регулярно публикуется. Ссылка в этой Рекомендации на какой-либо документ, являющийся независимым документом, не дает ему статуса Рекомендации.

[1] ITU-T Recommendation G.711 (1988), *Pulse code modulation (PCM) of voice frequencies*.

[2] ITU-T Recommendation G.722 (1988), *7 kHz audio-coding within 64 kbit/s*.

[3] ITU-T Recommendation G.728 (1992), *Coding of speech at 16 kbit/s using low-delay code excited linear prediction*.

[4] ITU-T Recommendation G.723.1 (1996), *Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*.

[5] ITU-T Recommendation G.729 (1996), *Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)*.

[6] ITU-T Recommendation H.221 (1999), *Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices*.

[7] ITU-T Recommendation H.230 (1999), *Frame-synchronous control and indication signals for audiovisual systems*.

[8] ITU-T Recommendation H.233 (1995), *Confidentiality system for audiovisual services*.

[9] ITU-T Recommendation H.242 (1999), *System for establishing communication between audiovisual terminals using digital channels up to 2 Mbit/s*.

[10] ITU-T Recommendation H.243 (2000), *Procedures for establishing communication between three or more audiovisual terminals using digital channels up to 1920 kbit/s*.

[11] ITU-T Recommendation H.245 (2003), *Control protocol for multimedia communication*.

[12] ITU-T Recommendation H.261 (1993), *Video codec for audiovisual services at p × 64 kbit/s*.

[13] ITU-T Recommendation H.263 (1998), *Video coding for low bit rate communication*.

[14] ITU-T Recommendation H.320 (1999), *Narrow-band visual telephone systems and terminal equipment*.

[15] ITU-T Recommendation T.122 (1998), *Multipoint communication service – Service definition*.

[16] ITU-T Recommendation T.123 (1999), *Network-specific data protocol stacks for multimedia conferencing*.

[17] ITU-T Recommendation T.125 (1998), *Multipoint communication service protocol specification*.

[18] ITU-T Recommendation H.321 (1998), *Adaptation of H.320 visual telephone terminals to B-ISDN environments*.

[19] ITU-T Recommendation H.322 (1996), *Visual telephone systems and terminal equipment for local area networks which provide a guaranteed quality of service*.

[20] ITU-T Recommendation H.324 (1998), *Terminal for low bit-rate multimedia communication*.

[21] ITU-T Recommendation H.310 (1998), *Broadband audiovisual communication systems and terminals*.

[22] ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.

[23] ITU-T Recommendation Q.932 (1998), *Digital subscriber signalling system No. 1 – Generic procedures for the control of ISDN supplementary services*.

[24] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

[25] ITU-T Recommendation X.681/(2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.

[26] ITU-T Recommendation X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

[27] ITU-T Recommendation E.164 (1997), *The international public telecommunication numbering plan*.

[28] ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.

[29] ITU-T Recommendation Q.850 (1998), *Usage of cause and location in the digital subscriber Signalling System No. 1 and the Signalling System No. 7 ISDN user part*.

[30] ITU-T Recommendation Q.950 (2000), *Supplementary services protocols, structure, and general principles*.

[31] ITU-T Recommendation H.235 (2000), *Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals*.

[32] ISO/IEC 11571:1998, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Networks – Addressing*.

[33] IETF RFC 1738 (1994), *Uniform Resource Locators (URL)*.

[34] IETF RFC 2068 (1997), *Hypertext Transfer Protocol – HTTP/1.1*.

[35] IETF RFC 1766 (1995), *Tags for the Identification of Languages*.

[36] ITU-T Recommendation H.248 (2000), *Gateway control protocol*.

[37] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

[38] IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control*.

[39] IETF RFC 2032 (1996), *RTP Payload Format for H.261 Video Streams*.

[40] ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

3 Определения терминов

См. определения терминов в Рекомендации МСЭ-Т H.323. В Рекомендации МСЭ-Т H.323 термин "конечная точка" (endpoint) используется для обозначения терминалов, шлюзов и блоков MCU как элементов, которые способны принимать или инициировать соединения. В настоящей Рекомендации термин "терминал" (terminal) часто используется как общее средство в описаниях установления соединения и должен пониматься как ссылка на элемент, который может участвовать в установлении соединения, включая шлюз или блок MCU.

4 Соглашения

В этой Рекомендации "должен" (shall) указывает обязательное требование, а "следует" (should) указывает предлагаемую, но необязательную функцию или процедуру. Термин "может" (may) указывает необязательный образ действия без выражения желательности.

Когда применяется термин "MCU", он относится к MCU H.323. Если подразумевается MCU H.231, то это будет явно указываться.

В этой Рекомендации "килобит/секунду" сокращается до кбит/с и измеряется в единицах "1000". Следовательно, 64 кбит/с равняется точно 64 000 битов в секунду.

Если не указано другое, для всех описаний на языке ASN.1 в этой Рекомендации используется вариант ALIGNED уплотненных правил кодирования (PER) ASN.1.

Названия сообщений из Q.931 даются с заглавными первыми буквами, а элементы ASN.1 даются **полужирным шрифтом**.

5 Сокращения

В этой Рекомендации используются следующие сокращения:

5.1 Общие сокращения

ИКМ	Импульсно-кодовая модуляция (PCM)	
КО	Качество обслуживания (QoS)	
MA5	Международный алфавит № 5 (IA5)	
BAS	Bit rate Allocation Signal	Сигнал распределения скорости битов
CIF	Common Intermediate Format	Общий промежуточный формат
CRV	Call Reference Value	Значение справочного номера соединения
ECS	Encryption Control Signal	Сигнал управления шифрованием
FFS	For Further Study	Остается для изучения
GOB	Group of Blocks	Группа блоков
H-MLP	High speed Multi-Layer Protocol	Высокоскоростной многоуровневый протокол
HSD	High Speed Data	Высокоскоростные данные
IE	Information Element	Информационный элемент
IETF	Internet Engineering Task Force	Комитет по инженерным проблемам Интернета
IP	Internet Protocol	Межсетевой протокол
LAN	Local Area Network	Локальная сеть
LD-CELP	Low Delay – Code Excited Linear Prediction	Линейное предсказание с кодовым возбуждением с малой задержкой
LSB	Least Significant Bit	Бит младшего разряда
LSD	Low Speed Data	Низкоскоростные данные
MB	Macro Block (see ITU-T Rec. H.261)	Блок макросов, макроблок, см. Рек. МСЭ-Т H.261
MBE	Multi-Byte Extension	Многобайтовое расширение
MCC	Multipoint Command Conference	Многоточечная управляемая конференция
MCN	Multipoint Command Negating	Отрицание многоточечного управления
MCS	Multipoint Command Symmetrical data transmission	Симметричная передача данных многоточечного управления
MCS	Multipoint Communication Service	Многоточечная система связи
MCU	Multipoint Control Unit	Многоточечный блок управления
MF	MultiFrame	Многокадровый
MLP	Multi-Layer Protocol	Многоуровневый протокол
MPI	Minimum Picture Interval	Минимальный интервал изображения
MSB	Most Significant Bit	Бит старшего разряда
NA	Not Applicable	Не применимо
NS	Non-Standard	Нестандартный
NSAP	Network Service Access Point	Точка доступа к услуге сетевого уровня
PDU	Protocol Data Unit	Протокольный блок данных
QCIF	Quarter Common Intermediate Format	Четверть CIF

RAS	Registration, Admission and Status	Регистрация, допуск и статус
RTCP	Real-time Transport Control Protocol	Транспортный протокол управления в реальном времени
RTP	Real-time Transport Protocol	Транспортный протокол реального времени
SBE	Single Byte Extension	Однобайтовое расширение
SC	Service Channel	Служебный канал
SCM	Selected Communication Mode	Выбранный режим связи
SCN	Switched Circuit Network	Сеть с коммутацией каналов
TCP	Transport Control Protocol	Транспортный протокол управления
TSAP	Transport Service Access Point	Точка доступа к услуге транспортного уровня
UDP	User Datagram Protocol	Протокол датаграмм пользователя
URL	Uniform Resource Locator	Унифицированный указатель ресурсов
VCF	Video Command "Freeze picture Request"	Видеокоманда "Запрос стоп-кадра"
VCU	Video Command "Fast Update Request"	Видеокоманда "Запрос быстрого обновления"

5.2 Сокращения названий сообщений RAS

ACF	Admissions Confirm	Подтверждение допуска
ARJ	Admissions Reject	Отклонение допуска
ARQ	Admissions Request	Запрос допуска
BCF	Bandwidth Confirm	Подтверждение полосы пропускания
BRJ	Bandwidth Reject	Отклонение полосы пропускания
BRQ	Bandwidth Request	Запрос полосы пропускания
DCF	Disengage Confirm	Подтверждение разъединения
DRJ	Disengage Reject	Отклонение разъединения
DRQ	Disengage Request	Запрос разъединения
GCF	Gatekeeper Confirm	Подтверждение гейткипера
GRJ	Gatekeeper Reject	Отказ гейткипера
GRQ	Gatekeeper Request	Запрос гейткипера
IACK	Information request Acknowledgement	Подтверждение запроса информации
INAK	Information request Negative Acknowledgement	Отрицательное подтверждение запроса информации
IRQ	Information Request	Запрос информации
IRR	Information Request Response	Ответ на запрос информации
LCF	Location Confirm	Подтверждение местонахождения
LRJ	Location Reject	Отклонение местонахождения
LRQ	Location Request	Запрос местонахождения
RAC	Resource Availability Confirmation	Подтверждение доступности ресурсов
RAI	Resource Availability Indication	Индикация доступности ресурсов
RCF	Registration Confirm	Подтверждение регистрации
RIP	Request In Progress	Запрос продолжается
RRJ	Registration Reject	Отказ в регистрации

RRQ	Registration Request	Запрос регистрации
SCI	Service Control Indication	Индикация управления службой
SCR	Service Control Response	Ответ управления службой
UCF	Unregistration Confirm	Подтверждение отмены регистрации
URJ	Unregistration Reject	Отклонение отмены регистрации
URQ	Unregistration Request	Запрос отмены регистрации

6 Механизм пакетирования и синхронизации

6.1 Общий подход

До создания любого соединения конечная точка может обнаружить гейткипер и зарегистрироваться у него. Если это происходит, то желательно, чтобы конечная точка знала марку гейткипера, у которого она регистрируется. Желательно также, чтобы гейткипер знал марку конечной точки, которая регистрируется у него. По этим причинам обе последовательности *обнаружения* и регистрации содержат ИДЕНТИФИКАТОР ОБЪЕКТА типа H.245, который позволяет определять марку в терминах реализованной версии Рекомендации МСЭ-Т H.323. Эта последовательность может также содержать факультативные нестандартные части сообщения, позволяющие конечным точкам устанавливать нестандартные взаимоотношения. К концу этой последовательности как гейткиперы, так и конечные точки будут осведомлены о номере версии и нестандартном статусе друг друга.

В последовательности сообщений Установить/Соединить, описанной ниже, номер версии является обязательным, а нестандартная информация – факультативной, что позволяет двум конечным точкам информировать друг друга о своей марке и нестандартном статусе. Заметим, однако, что все сообщения сигнализации о сообщении H.225.0 имеют поле для факультативного нестандартного сообщения в информационном элементе Пользователь–пользователь и что все сообщения в канале RAS имеют факультативное поле для нестандартной информации. Кроме того, определено нестандартное сообщение RAS, которое может быть передано в любое время.

Ненадежный канал для передачи сообщений регистрации, допуска и статуса называется каналом RAS (Registration, Admission and Status). Общий подход к запуску соединения состоит в передаче обязательного запроса допуска по каналу RAS[3], за которым следует начальное сообщение Установить по транспортному адресу надежного канала (этот адрес мог быть передан в сообщении с подтверждением допуска или мог быть известен вызывающему терминалу). В результате этого начального сообщения начинается последовательность установления соединения, основанная на операциях сигнализации о соединении H.225.0 с улучшениями, описанными ниже. Эта последовательность завершается, когда терминал получит в сообщении Соединить надежный транспортный адрес, по которому следует передавать управляющие сообщения H.245[4].

Когда сообщения передаются по надежному каналу сигнализации о соединении H.225.0, только одно полное сообщение должно передаваться в пределах границ, определенных надежным транспортом; не должно быть фрагментации сообщений H.225.0 в транспортных блоках PDU. (В реализациях IP, как пояснено в Добавлении IV, такой PDU определяется форматом TPKT.)

Когда надежный управляющий канал H.245 уже установлен, может быть установлен дополнительный канал для аудио, видео и данных на основе результата обмена возможностями с помощью процедур логического канала H.245. Выполняется также согласование природы мультимедийной конференции на стороне пакетной сети (централизованной или распределенной/многопунктовой) для отдельного

---

[3] От терминала, который не зарегистрирован гейткипером, не требуется передавать запрос допуска.

[4] Заметим, что адрес H.245 может быть передан в сообщении Предупреждение или Готовность Вызова для сокращения времени установления соединения. Заметим, что канал H.245 может быть открыт сразу после получения адреса H.245 в сообщении Установить.

соединения[5]. Это согласование выполняется для каждого носителя; это означает, например, что аудио/видео могут быть распределенными, а данные и управление – централизованными.

Когда сообщения передаются по надежному управляющему каналу H.245, могут передаваться несколько сообщений в пределах границ, определенных PDU надежного транспорта, пока передаются полные сообщения; не должно быть фрагментации сообщений H.245 в транспортных PDU. (В реализациях IP, как пояснено в Добавлении IV, такой PDU определяется форматом TPKT.)

Терминалы H.225.0 должны быть способны передавать аудио и видео с помощью протокола реального времени (Real Time Protocol, RTP) через ненадежные каналы для минимизации задержки (времени переноса). Для преодоления влияния потери пакетов может применяться маскировка ошибок или другое восстанавливающее действие; обычно пакеты аудио/видео не передаются повторно, так как это привело бы к чрезмерной задержке в среде пакетной сети[6]. Предполагается, что ошибки по битам обнаруживаются на нижних уровнях и что пакеты с ошибками не отправляются к H.225.0. Заметим, что аудио/видео и сигнализация о соединении/управление H.245 никогда не передаются по одному и тому же каналу и не используют совместно структуру общего сообщения. Терминалы H.225.0 должны быть способны передавать и принимать аудио и видео по отдельным транспортным адресам с помощью отдельных экземпляров RTP, чтобы позволить специфичные для носителя порядковые номера кадров и раздельное обеспечение качества обслуживания для каждого носителя. Однако остается для изучения факультативный режим, при котором пакеты аудио и видео смешиваются в один кадр, который передается по одному транспортному адресу.

Возможности T.120 согласуются с помощью H.245; после получения соответствующих сообщений устанавливаются конференции T.120 с использованием подходящих стеков T.123 для транспортной/пакетной сети. Протокол T.120 должен переноситься по пакетной сети между конечными точками к другому транспортному адресу. В таблице 1 показан ряд идентификаторов TSAP, используемых для каждого носителя в двухточечном соединении. Справедливо также, что рассматриваемый терминал H.323 может быть способен участвовать одновременно более чем в одной конференции в результате использования дополнительных идентификаторов TSAP. Все используемые логические каналы H.245 являются однонаправленными, кроме тех, которые связаны с T.120 и являются двунаправленными.

Таблица 1/H.225.0 – Идентификаторы TSAP, используемые в H.225.0 для однопунктового соединения "точка-точка"

Использование идентификаторов TSAP	Надежный или ненадежный	Общеизвестный или динамический
Аудио/RTP	Ненадежный	Динамический
Аудио/RTCP	Ненадежный	Динамический
Видео/RTP	Ненадежный	Динамический
Видео/RTCP	Ненадежный	Динамический
Сигнализация о соединении	Надежный	Общеизвестный или динамический
H.245	Надежный	Динамический
Данные (T.120)	Надежный	Общеизвестный или динамический
RAS	Ненадежный	Общеизвестный или динамический
ПРИМЕЧАНИЕ. – Если используются общеизвестные идентификаторы TSAP, то может быть только одна конечная точка для сетевого адреса. В модели прямого соединения вызывающая сторона нуждается также в общеизвестном идентификаторе TSAP для канала сигнализации о соединении, чтобы начать соединение.		

---

[5] Конференция на стороне LAN может быть частично централизованной и частично распределенной по решению многоточечного контроллера (Multipoint Controller, MC), управляющего конференцией. Однако терминал не осведомлен об этом факте. Обычно все терминалы, конечно, будут видеть один и тот же Выбранный режим связи (Selected Communications Mode, SCM) (см. его определение в Рек. МСЭ-Т H.243).

[6] С помощью сигнализации H.245 может быть затребовано быстрое обновление полных кадров, блоков макросов (MB) или групп блоков (GOB).

Несмотря на то что транспортные адреса, например, для аудио и видео, могут совместно использовать один и тот же адрес пакетной сети, отличаясь только идентификаторами TSAP, некоторые изготовители могут использовать разные адреса пакетной сети для аудио и видео. Единственным требованием является выполнение соглашений из Приложений A и B при нумерации идентификаторов TSAP в сеансе RTP[7].

В таблице 1 был описан основной случай однопунктовых операций "точка-точка" между двумя терминалами. Чтобы упростить конструкцию шлюзов, блоков MCU и гейткиперов, могут использоваться динамичные идентификаторы TSAP вместо общеизвестных идентификаторов TSAP. В таблице 2 и 3 приводятся примеры использования идентификаторов TSAP для случая шлюз/MCU и для случая гейткипера.

Таблица 2/H.225.0 – Идентификаторы TSAP, используемые в одном порту MCU/шлюза (однопунктовый пример)

Использование идентификаторов TSAP	Надежный или ненадежный	Общеизвестный или динамический
Аудио/RTP	Ненадежный	Динамический
Аудио/RTCP	Ненадежный	Динамический
Видео/RTP	Ненадежный	Динамический
Видео/RTCP	Ненадежный	Динамический
Сигнализация о соединении	Надежный	Динамический (примечание)
H.245	Надежный	Динамический
Данные (T.120)	Надежный	Динамический
RAS	Ненадежный	Динамический (примечание)
ПРИМЕЧАНИЕ. – См. примечание 1 к таблице 3.		

Таблица 3/H.225.0 – Пример использования идентификаторов TSAP в гейткипере H.225.0, который поддерживает модель соединения при посредничестве гейткипера из рисунка 28/H.323 для двухточечного соединения

Использование идентификаторов TSAP	Надежный или ненадежный	Общеизвестный или динамический	Число каналов
Сигнализация о соединении	Надежный	Динамический или общеизвестный (примечание 1)	2 на соединение (примечание 2)
H.245	Надежный	Динамический	2 на соединение (примечание 2)
RAS	Ненадежный	Общеизвестный	1
ПРИМЕЧАНИЕ 1. – Если используется общеизвестный идентификатор TSAP, то гейткипер может быть сведен к одной конечной точке на устройство; поэтому должны использоваться динамические идентификаторы TSAP. ПРИМЕЧАНИЕ 2. – 0 для модели прямого соединения, 2 для модели соединения при посредничестве гейткипера.			

Заметим, что надежный транспортный адрес используется для установления соединения типа терминал-терминал, а также в случае с посредничеством гейткипера. Надежное соединение сигнализации о соединении должно удерживаться активным до приема сообщения "Освобождение завершено" для всех активных соединений, использующих этот канал сигнализации о соединении.

Заметим, что в рассматриваемый момент времени могут быть открытыми не один, а несколько каналов H.245, то есть конечная точка может участвовать одновременно в нескольких соединениях/конференциях. Заметим также, что во время конкретного соединения терминал может иметь открытыми несколько каналов одного и того же типа, например два аудиоканала для стерео-аудио. Ограничением является лишь то, что для каждого двухточечного соединения должен быть один и только один канал управления H.245.

---

[7] Заметим, что любой идентификатор TSAP может использоваться для начального сеанса RTP; главной причиной соблюдения соглашения RTP является возможное взаимодействие с RTP Комитета IETF.

Сигнализация по логическим каналам H.245 используется для начала и окончания применения протокола видео, аудио и данных. Этот процесс вызывает закрытие открытого канала и затем новое открытие с новым режимом работы. Как часть этого процесса открытия канала, перед посылкой подтверждения открытого логического канала, конечная точка использует последовательность ARQ/ACF или BRQ/BCF для обеспечения доступа к достаточной полосе пропускания для нового канала (если достаточная полоса пропускания не доступна из предшествующей последовательности ARQ/ACF или BRQ/BCF). В некоторых случаях шлюз может обнаружить, что смена режима на стороне SCN появилась быстрее, чем смена режима на стороне пакетной сети, что создает возможность пропадания аудиоинформации. Шлюз может иметь разные подходы по усмотрению изготовителя:

a) шлюз может транскодировать аудио, препятствуя таким образом распространению смены режима SCN;

b) шлюз может просто отбрасывать аудиоинформацию; или

c) шлюз может работать как MCU H.231, получая таким образом управление всеми изменениями режима на стороне SCN.

Нет общего правила о том, какие процедуры, H.245 или RTP (см. Приложения A, B и C), получают первенство; каждый конфликт и его разрешение специально рассматриваются в этой Рекомендации.

Заметим также, что нет фиксированной связи между идентификаторами источника синхронизации (Synchronization Source Identifier, SSRC) и логическими каналами; Рекомендация МСЭ-Т H.245 обеспечивает эту связь, которая может использоваться для синхронизации аудио/видео.

Обычно возможны два типа режимов работы конференции на стороне пакетной сети: распределенный и централизованный. Возможно также, что для разных носителей делается разный выбор, например, распределенные аудио/видео и централизованные данные. Процедуры определения типа устанавливаемой конференции содержатся в Рекомендации МСЭ-Т H.323; сообщения этой Рекомендации предназначены для поддержки всех разрешенных комбинаций, учитывая, что распределенные управление и данные остаются для изучения, хотя и поддерживаются сигнализацией о возможностях H.245.

6.2 Использование RTP/RTCP

Конечная точка H.225.0 должна быть способна использовать отдельные идентификаторы TSAP для аудио и видео и соответствующие каналы RTCP, как описано в Приложениях A и B. Факультативно конечные точки могут выбрать использование разных адресов пакетной сети для аудио и видео, но для каждого адреса пакетной сети следует соблюдать соглашения из Приложений A и B при использовании идентификаторов TSAP. При использовании сигнализации H.245 могут устанавливаться дополнительные каналы аудио и видео, если терминал имеет эту способность.

Факультативная способность использовать один транспортный адрес как для аудио, так и для видео, остается для изучения.

Реализации, кроме особого случая, специально отмеченного здесь, должны соблюдать те RTP, которые содержатся в Приложении A, если они не изменены в тексте этой Рекомендации. Реализации должны соблюдать профиль RTP (см. Приложение B) только так, как специально отмечается в этой Рекомендации.

Трансляторы и смесители RTP не являются элементами системы H.323, а любые сведения о них в Приложениях A и B должны рассматриваться как информативные. Заметим, что как шлюзы, так и блоки MCU имеют некоторые аспекты смесителей и трансляторов, поэтому сведения из Приложений A и B могут быть полезны при реализации шлюзов и MCU. Однако MCU не является смесителем, а смеситель не является MCU. Заметим, что шлюзы, например, могут работать как трансляторы в соединении от пакетной сети к пакетной сети через шлюз.

Версия (V): Должен использоваться RTP версии 2.

Подсчет CSRC (CC): Использование подсчета CSRC в этой Рекомендации является факультативным. Когда он не используется, значение CC должно быть нулем (0). CSRC может использоваться блоками MCU для обеспечения информации об участниках в аудиосумме, когда имеется обработка распределенного аудио. Заметим, что отсутствуют возможности, связанные со способностью понимать подсчет CSRC, поэтому Многоточечный блок управления/Многоточечный контроллер (Multipoint Control Unit, MCU/Multipoint Controller, MC) не имеют средств узнать, использует ли терминал и как именно эту информацию в конференции.

Каноническое имя (Canonical Name, CNAME): В простейшем случае двухточечного соединения по пакетной сети сигнал SSRC (Synchronization Source) используется для идентификации источника аудио/видео из терминала, а потоки связываются с помощью CNAME, как будто они выдаются одной и той же конечной точкой, как указано в Приложении A.

При использовании RTCP должны периодически передаваться пакеты RR или SR, как описано в Приложении A. Должно использоваться SDES-сообщение CNAME. Остальные SDES-сообщения (см. Приложение A) являются факультативными, но не должны использоваться для управления конференцией или для информации конференции, когда используются функции управления H.245 и/или T.120. Информация из Рекомендации МСЭ-Т H.245 и/или Рекомендации МСЭ-Т T.120 должна считаться правильной информацией.

Сообщение BYE из RTCP не должно передаваться при завершении сеанса RTP. Терминал H.323 определяет момент разъединения соединения с помощью процедур из Рекомендации МСЭ-Т H.323. Единственным обязательным применением пакета BYE из RTCP является преодоление столкновения SSRC.

Терминал H.323, вступивший в любую конференцию, двухточечную или многоточечную, должен ограничивать скорость передачи битов логического канала, усредненную за интервал, определенный в Рекомендации МСЭ-Т H.245, до значения, указанного в **FlowControlCommands** H.245, в командах логического канала H.245 и механизмом управления потоком из T.120.

Когда терминал H.323 соединен со шлюзом H.323, этот шлюз должен использовать средства из Рекомендации МСЭ-Т H.245 и Рекомендации МСЭ-Т T.120, чтобы вынуждать терминал H.323 передавать со скоростью, меньшей или равной скоростям передачи носителей на стороне SCN, и принимать на скорости, равной или превышающей скорость SCN, со следующими особыми случаями:

- Полоса пропускания для управления в пакетной сети может не соответствовать указанной в Рекомендации МСЭ-Т H.221.
- Полоса пропускания для аудио в пакетной сети может соответствовать указанной в Рекомендации МСЭ-Т H.221 для SCN, но при транскодировании в шлюзе соответствие не требуется.
- В случае использования сжатия данных в шлюзе: терминал H.323 на стороне пакетной сети не должен превышать указанную в H.245 скорость, которая будет, вероятно, меньше, чем скорость, передаваемая по SCN.

Шифрование для конечных точек остается для изучения.

6.2.1 Аудио

Перед рассмотрением пакетирования аудио с помощью RTP необходимо обратить внимание на то, как аудио передается через H.245, и на взаимоотношение этой передачи с RTP. Обычно, когда аудиоканал открыт, логический канал H.245 тоже открыт. Сигнализация H.245 в структуре **AudioCapability** дается в виде максимального числа кадров в пакете. Размер кадра для настоящей Рекомендации изменяется в соответствии с используемым аудиокодированием.

Все терминалы H.323, обеспечивающие аудиосвязь, должны поддерживать G.711. При всех аудиокодеках, основанных на кадрах, приемники должны сообщать максимальное число аудиокадров, которое они способны принимать в одном аудиопакете. Передатчики могут передавать любое целое число аудиокадров в каждом пакете, вплоть до максимума, установленного приемником. Передатчики не должны расщеплять аудиокадры по пакетам и должны передавать целое число октетов в каждом аудиопакете.

Кодеки, основанные на квантовании, такие как G.711 и G.722, должны рассматриваться как основанные на кадрах с размером кадра равным восьми выборкам. (См. в Приложении B дополнительную информацию с указаниями по кодированию аудио на основе квантования.) При аудиоалгоритмах, таких как G.723.1, в которых используются несколько размеров аудиокадра, границы аудиокадра внутри каждого пакета должны указываться внутриканальными средствами этого аудиоканала.

При аудиоалгоритмах с фиксированным размером кадра (см. в Рекомендациях МСЭ-Т G.728 и G.729 размер кадра, примененный в каждой из них) границы аудиокадра должны определяться отношением размера пакета к размеру аудиокадра; другими словами, в пакет RTP должны помещаться только целые аудиокадры.

Тип полезной нагрузки (Payload Type, PT): Для кодеков МСЭ должны использоваться только типы полезной нагрузки МСЭ-Т, такие как (0)[PCMU], (8)[PCMA], (9)[G722] и (15)[G728], которые передаются в Рекомендации МСЭ-Т H.245. Для любых типов полезной нагрузки МСЭ-Т, не перечисленных в Приложении B, должны использоваться динамические типы полезной нагрузки, передаваемые с помощью сигнализации H.245.

Рекомендуется, чтобы при обнаружении прерывания порядковых номеров приемник мог повторить самые последние принятые звуки так, чтобы амплитуда повторного звука снижалась до тишины; могут быть использованы другие похожие процедуры по усмотрению изготовителя.

Каждый октет G.711 должен быть октетом, сфазированным с пакетом RTP. Бит sign каждого октета G.711 должен соответствовать биту старшего разряда этого октета в пакете RTP (то есть, полагая, что выборки G.711 обрабатываются в хост-машине в виде октетов, бит sign должен быть битом старшего разряда в октете, как определено форматом хост-машины).

Когда к пакетной сети передается сигнал ИКМ со скоростью 48/56 кбит/с, шлюз H.323 должен заполнять лишние 1 или 2 бита в каждом октете согласно примечанию 2 в таблице 1b/G.711 и использовать значения RTP для PCMA или PCMU (8 или 0). При законе μ заполнение состоит из "1" в обоих битах 7 или 8. При законе A бит 7 устанавливается в 0, а бит 8 – в 1. В обратном направлении шлюз H.323 должен обрезать сигнал G.711 со скоростью 64 кбит/с на стороне пакетной сети, чтобы согласовать со скоростью G.711, примененной в H.320. Следовательно, на стороне пакетной сети должна использоваться только G.711 на 64 кбит/с.

Когда к пакетной сети передается сигнал G.722 со скоростью 48/56 кбит/с, шлюз H.323 должен заполнять лишние 1 или 2 бита в каждом октете и использовать динамические типы полезной нагрузки RTP, указанные в Рекомендации МСЭ-Т H.245, для определения различия между 64 кбит/с (при которой используется PT = 9) и сниженными скоростями. В обратном направлении шлюз H.323 должен обрезать G.722 – 64 кбит/с на стороне пакетной сети, чтобы согласовать со скоростью G.711, примененной в H.320. Следовательно, на стороне пакетной сети должна использоваться только G.722 на 64 кбит/с.

Если возможно, то в терминале H.323 следует использовать свойство подавления пауз из RTP, особенно в многопунктовой конференции. Терминал H.323 должен быть способен принимать потоки RTP с компрессированными паузами. Кодеры могут опускать передачу аудиосигналов во время пауз после передачи одного кадра паузы, либо могут передавать заполняющие кадры фона пауз, если такие методы указаны в используемой Рекомендации об аудиокодеке.

6.2.2 Видеосообщения

Тип полезной нагрузки (PT): Для кодеков МСЭ должны использоваться только типы полезной нагрузки МСЭ-Т, которые применены в Рекомендации МСЭ-Т H.261 или Рекомендации МСЭ-Т H.263 и передаются в Рекомендацию МСЭ-Т H.245. Для кодеков, которые могут передавать по H.245 и для которых не определены форматы пакетирования, могут использоваться динамические типы полезной нагрузки.

Маркер (M): Бит-маркер следует устанавливать согласно процедурам, описанным в Приложении A, кроме случаев, когда он будет увеличивать задержку (время переноса) "от конца до конца".

Чтобы защищаться от потери видеопакетов, должны поддерживаться **VideoFastUpdatePicture**, **VideoFastUpdateMB** и **VideoFastUpdateGOB** из H.245. Использование управляющих пакетов Full Intra Request (FIR) [передай мне полный кадр] и Negative Acknowledgment (NACK) [передай мне определенные пакеты] является факультативным и передается в возможностях H.245.

Метод 3) защиты от ошибок из раздела 5 документа RFC 2032 [39] может оказаться непригодным, если NACK не приходит в пределах времени одного кадра.

Сигнал H.261 пакетируется на стороне пакетной сети согласно Приложению C. Когда доступны достаточно длинные пакеты RTP, фрагментация по границам блоков макросов (MB) в передатчике не требуется. Однако, если терминал H.323 выполняет фрагментацию пакетов H.261 на уровне RTP, то эта фрагментация должна проводиться на границах MB. Все терминалы H.323 должны быть способны принимать MB-фрагментированные пакеты, а также GOB-фрагментированные пакеты либо пакеты со смесью MB и GOB. Заметим, что сбой поддержки MB-фрагментации в передатчике может привести к потере целой GOB и может также снизить скорость передачи пакетов. Для достижения максимальной устойчивости работы используемые пакеты RTP не должны превышать размера Максимального блока переноса (Maximum Transfer Unit, MTU) в заданной пакетной сети, но если самый маленький независимо кодируемый элемент кодирующей схемы (например, блок макросов) превышает размер

MTU, то не требуется разбивать пакет по блокам MTU. Блоки макросов (MB) не должны расщепляться по пакетам; все пакеты должны оканчиваться на границе GOB или MB. Передатчик H.323 может заполнять пакет, содержащий маленький GOB, дополнительными MB, но это не обязательно.

Чтобы избегать возможности искажения нескольких изображений из-за потери пакета RTP, пакетизатор RTP в конечной точке H.323 должен включать в пакет RTP видеосигнал только одного изображения.

SBIT – это количество битов старших разрядов, которые должны игнорироваться в первом октете данных. EBIT – это количество битов младших разрядов, которые должны игнорироваться в последнем октете данных.

Пакетизатор RTP не должен намеренно выравнивать видео по октетам в начале пакетов RTP. Другими словами, если EBIT = n (причем $0 < n < 8$) в пакете RTP, то SBIT в следующем пакете RTP должен равняться $8 – n$, а если EBIT = 0 в пакете RTP, то SBIT в следующем пакете RTP должен равняться 0. Это требование устраняет возможную дополнительную задержку "от конца до конца" из-за сдвига битов. Это требование должно применяться без учета границ изображений.

В Приложении D определяется расширение H.323 для заголовка видеопакета, которое содержит факультативный подсчет октетов. Использование этого факультативного расширения описано в Приложении D.

См. в Добавлении IV специфические для пакетной сети советы по пакетированию видео.

6.2.3 Сообщения данных

Специальных сообщений или форматов для данных не имеется; T.120 в пакетной сети используется так же, как для Рекомендации МСЭ-Т T.123. Централизованная организация конференции данных на пакетной сети в сравнении с распределенной организацией описана в Рекомендации МСЭ-Т H.323; согласование в ней выполняется с помощью H.245.

Управление потоком T.120 в пакетной сети, когда требуется командой H.245 **FlowControlCommand** и пределами **maxBitRate**, осуществляется с помощью протоколов пакетной сети.

См. в Рекомендации МСЭ-Т H.323 процедуры для соединения действующей конференции T.120 с конференцией H.323 либо для добавления соединения H.323 к конференции T.120.

Протокол в пакетной сети для использования с H.224 остается для изучения.

7 Определения сообщений H.225.0

Этот раздел посвящен определениям сообщений для установления соединения, управления соединением и связи между терминалами, шлюзами, гейткиперами и блоками MCU.

Определения на языке ASN.1 для всех сообщений H.225.0 даются в Приложении H.

7.1 Использование сообщений Q.931

Реализации должны выполнять Рекомендацию МСЭ-Т Q.931, как указано в настоящей Рекомендации. Терминалы могут также поддерживать факультативные блоки APDU из H.450 в информационных элементах (IE) "Пользователь-пользователь". Сообщения должны содержать все обязательные информационные элементы и могут содержать любые факультативные информационные элементы, определенные в Рекомендации МСЭ-Т Q.931, как описано в настоящей Рекомендации. Заметим, что конечная точка H.225.0 может согласно Рекомендации МСЭ-Т Q.931 игнорировать все факультативные сообщения, которые она не поддерживает, без вреда для взаимодействия, но должна отвечать на неизвестное сообщение сообщением Статус.

Каждая конечная точка H.225.0 должна быть способна принимать и распознавать входящее сообщение H.225.0 с сигнализацией о соединении, включая то, которое содержит APDU из H.450 в IE "Пользователь-пользователь". Она должна быть способна обрабатывать обязательные сообщения сигнализации о соединении H.225.0; она может быть способна обрабатывать факультативные сообщения сигнализации о соединении H.225.0. В любом случае, каждая конечная точка H.225.0 должна быть способна игнорировать сообщения, неизвестные ей, без нарушения работы.

Каждая конечная точка H.225.0 должна быть способна распознавать и генерировать информационные элементы, предписанные ниже для соответствующих сообщений сигнализации о соединении H.225.0,

и блоки APDU в IE "Пользователь-пользователь". Она может еще распознавать и генерировать факультативные информационные элементы, определенные ниже. Она может также распознавать другие информационные элементы из Q.931 или из других протоколов серии Q или H.450. Конечные точки должны быть способны игнорировать неизвестные информационные элементы, содержащиеся в сообщении сигнализации о соединении H.225.0 или в APDU H.450, без нарушения работы. Процедуры приема нераспознаваемых, "требующихся для полноты", информационных элементов должны применяться согласно 5.8.7.1/Q.931. Конечные точки H.225.0 не должны передавать несколько информационных элементов одного и того же типа в том же сообщении; например, они не должны передавать несколько информационных элементов Номер вызывающей стороны, описанных в Приложении A/Q.951.3.

Информационные элементы должны кодироваться согласно Рекомендации МСЭ-Т Q.931, кроме тех, которые изменены в настоящей Рекомендации. Однако Рекомендация МСЭ-Т Q.931 должна всегда определять правильный порядок следования информационных элементов внутри сообщения, независимо от порядка перечисления элементов в настоящей Рекомендации.

Промежуточные системы (шлюзы и гейткиперы) должны в отношении факультативных сообщений сигнализации о соединении H.225.0 и информационных элементов следовать приведенным ниже правилам:

1) Шлюзу следует, а гейткипер должен транслировать далее, после подходящих изменений, все информационные элементы (факультативные или обязательные), связанные с обязательными сообщениями сигнализации о соединении H.225.0, либо от терминала к шлюзу/терминалу, либо в обратном направлении. Это охватывает и такие информационные элементы, как информация "Пользователь-пользователь" и информация "Отображение".

2) Шлюзу следует транслировать далее все сообщения сигнализации о соединении, включая те, которые содержат блоки APDU и информационные элементы H.450, в обоих направлениях.

3) Гейткипер должен транслировать далее все сообщения сигнализации о соединении H.225.0, включая те, которые содержат блоки APDU и информационные элементы H.450, в обоих направлениях, после подходящего изменения. Заметим, что гейткипер может действовать как элемент сигнализации, который обеспечивает некоторые свойства (такие как свойства дополнительных услуг) и может поэтому изменять, получать или создавать сообщения сигнализации о соединении H.225.0.

Шлюзы H.323 могут быть способны преобразовывать дополнительные услуги серии H.450 и сообщения H.225.0 в соответствующие дополнительные услуги и сообщения ISO/IEC 11582, ISUP и других стандартов по сигнализации SCN. Детали составляют предмет рассмотрения Рекомендации МСЭ-Т H.246 и ее приложений.

Шлюзы H.323 могут быть способны пропускать неизмененные сообщения сигнализации ISO/IEC 11582, ISUP и других стандартов по сигнализации SCN с помощью туннелирования сигнализации не-H.323 в H.225.0. Детали имеются в Приложении M/H.323 (см. M.1/H.323, M.2/H.323 и т. д.).

В этой версии настоящей Рекомендации все ссылки сделаны на Рекомендации МСЭ-Т Q.931 версии 1998 года. Соблюдаются процедуры установления соединения с коммутацией каналов из 3.1/Q.931. Однако реализаторам напоминаем, что хотя сигнализируется о "переносчике" ("bearer"), никаких фактических "каналов B" типа ЦСИС (ISDN) нет на стороне пакетной сети. Успешное завершение "соединения" создает надежный канал "от конца до конца", который обеспечивает передачу сообщений H.245. Фактическое установление "переносчика" делается с помощью H.245. Однако использование Q.931 на стороне пакетной сети создает возможность взаимодействия с Q.931 на стороне SCN, а также обеспечивает хорошо испытанную основу для общих вызывающих средств, ориентированных на соединение.

Обычно используются симметричные процедуры из Приложения D/Q.931. Это означает, что конечный автомат Q.931 действует согласно Приложению D/Q.931, за исключением того, что процедуры из D.3/Q.931 (Столкновение вызовов) не должны выполняться; преодоление такого вредного состояния остается за прикладным уровнем.

Конечные точки, не поддерживающие наборы сдвинутых кодов (shifted code sets) Q.931, должны игнорировать все сообщения Q.931, использующие такие методы.

В таблице 4 показано, какие сообщения являются обязательными и факультативными для установления соединений H.323 и H.225.0 с использованием Q.931 в пакетной сети.

Таблица 4/H.225.0 – Использование сообщений Q.931/Q.932 в H.225.0

	Передача (M, F, O, CM) (примечание 1)	**Прием и выполнение (M, F, O (примечание 2), CM)**
Сообщения установления соединения		
Alerting (Предупреждение)	M	M
Call Proceeding (Готовность Вызова)	O	CM (примечания 3 и 6)
Connect (Соединить)	M	M
Connect Acknowledge (Подтверждение Соединения)	F	F
Progress (Прохождение)	O	CM (примечание 6)
Setup (Установить)	M	M
Setup Acknowledge (Подтверждение Установления)	O	O
Сообщения отбоя соединения		
Disconnect (Разъединить)	F	F
Release (Освободить)	F	F
Release Complete (Освобождение Завершено)	M (примечание 4)	M
Сообщения фазы "информация соединения"		
Resume (Возобновить)	F	F
Resume Acknowledge (Подтверждение Возобновления)	F	F
Resume Reject (Отклонение Возобновления)	F	F
Suspend (Приостановить)	F	F
Suspend Acknowledge (Подтверждение Приостановки)	F	F
Suspend Reject (Отклонение Приостановки)	F	F
User Information (Информация Пользователя)	O	O
Разные сообщения		
Congestion Control (Управление Перегрузкой)	F	F
Information (Информация)	O	CM (примечание 6)
Notify (Уведомление)	O	O
Status (Статус)	M (примечание 5)	M
Status Inquiry (Запрос Статуса)	O	M

Таблица 4/H.225.0 – Использование сообщений Q.931/Q.932 в H.225.0

	Передача (M, F, O, CM) (примечание 1)	**Прием и выполнение (M, F, O (примечание 2), CM)**
Сообщения Q.932/H.450		
Facility (Возможности)	M	M
Hold (Удержать)	F	F
Hold Acknowledge (Подтверждение Удержания)	F	F
Hold Reject (Отклонение Удержания)	F	F
Retrieve (Восстановить)	F	F
Retrieve Acknowledge (Подтверждение Восстановления)	F	F
Retrieve Reject (Отклонение Восстановления)	F	F
ПРИМЕЧАНИЕ 1. – M: Обязательное, F: Запрещенное, O: Факультативное, CM: Условно обязательное. Некоторое сообщение является CM, если оно требуется, когда поддержана некоторая опция (факультативное средство). ПРИМЕЧАНИЕ 2. – Заметим, что сообщение Статус не должно передаваться в ответ на сообщение, отмеченное здесь как "O"; приемник должен просто игнорировать такое сообщение, если он не поддерживает его. ПРИМЕЧАНИЕ 3. – Терминалы, предназначенные для использования со шлюзами, должны принимать и выполнять сообщение Готовность Вызова. ПРИМЕЧАНИЕ 4. – Освобождение Завершено требуется для закрытия надежного канала сигнализации о соединении H.225.0. Однако канал сигнализации о соединении должен остаться открытым, если все еще обрабатываются другие соединения, использующие тот же канал сигнализации о соединении. Кроме того, гейткипер может установить флаг **maintainConnection** в значение ИСТИНА, чтобы предотвратить закрытие канала сигнализации о соединении. ПРИМЕЧАНИЕ 5. – Конечная точка должна на неизвестное сообщение отвечать сообщением Статус; ответ на Запрос Статуса тоже обязателен. Однако от конечной точки не требуется посылать Запрос Статуса. На практике конечной точке следует иметь возможность понимать сообщение Статус, принятое в ответ на переданное сообщение, которое было неизвестно приемнику. ПРИМЕЧАНИЕ 6. – Конечные точки, которые поддерживают факультативные свойства, использующие эти сообщения (такие свойства, как туннелирование H.245, дополнительные услуги H.450, туннелирование протоколов сигнализации, или свойства, использующие **genericData**), должны обрабатывать эти сообщения.		

7.2 Общие информационные элементы Q.931

7.2.1 Информационные элементы заголовка

Для всех сообщений сигнализации о соединении H.225.0 имеются три общих поля, которые обязательны в дополнение к типу сообщения и которые описываются в этом разделе.

7.2.1.1 Дискриминатор протокола

Как определено в 4.2/Q.931.

Должен быть установлен в 08H, что определяет это сообщение как сообщение от пользователя в сеть по Q.931/I.451 (кодированное согласно рисунку 4-2/Q.931). Если гейткипер действует как сеть, обеспечивая дополнительные услуги, то может быть удобнее использовать другое значение. Это остается для изучения.

7.2.1.2 Справочный номер соединения

Как определено в 4.3/Q.931.

Длина значения справочного номера соединения, равная двум октетам, должна поддерживаться любой конечной точкой H.323.

Значение справочного номера соединения (CRV) выбирается на стороне, начинающей соединение, и должно быть уникальным в этом месте. При последующей связи вызывающая и вызываемая стороны должны использовать это значение справочного номера соединения во всех сообщениях, принадлежащих к этому конкретному соединению.

Значение кодируется согласно рисунку 4-5/Q.931 для двухоктетного значения справочного номера соединения. Октет старших разрядов значения справочного номера всегда кодируется в октете № 2.

Заметим, что CRV будет уникальным только на конкретной части соединения, например, между двумя терминалами или между терминалом и гейткипером. Если заданный терминал имеет два соединения в одной и той же конференции, то эти два соединения должны иметь одинаковые идентификаторы конференции, но разные CRV.

Флаг справочного номера соединения должен устанавливаться согласно процедурам, описанным в Рекомендации МСЭ-Т Q.931.

Заметим, что значения CRV, переносимые в сообщениях RAS, должны соответствовать структуре, определенной в Рекомендации МСЭ-Т Q.931. В частности, флаг справочного номера соединения должен включаться в качестве бита старшего разряда Значения Справочного Номера Соединения. Это ограничивает реальное CRV диапазоном от 0 до 32 767 включительно.

Глобальный Справочный Номер Соединения, показанный на рисунке 4-5/Q.931 и имеющий численное значение 0, используется для ссылок на все соединения определенного Канала сигнализации о соединении или Канала RAS.

7.2.1.3 Тип сообщения

Тип сообщения кодируется согласно рисунку 4-6/Q.931 с использованием значений, указанных в таблице 4-2/Q.931. Специфические для H.225.0 расширения остаются для изучения.

7.2.2 Информационные элементы, специфичные для сообщений

Общие правила кодирования для следующих информационных элементов определены в 4.5.1/Q.931 и в таблице 4-3/Q.931. Эти правила должны соблюдаться. Механизм перехода (escape) (см. рисунок 4-8/Q.931) является факультативным.

7.2.2.1 Возможность переноса

Этот информационный элемент кодируется согласно рисунку 4-11/Q.931 и таблице 4-6/Q.931. Если этот информационный элемент получен в соединении "пакетная сеть – пакетная сеть", то он может игнорироваться приемником. Если этот информационный элемент появился в сообщении Установить для соединения сигнализации, не зависящего от соединения, как определено в Рекомендации МСЭ-Т H.450.1, то кодирование должно подчиняться 7.2.2.1.2. Во всех других случаях кодирование должно подчиняться 7.2.2.1.1. Справочные номера октетов берутся из рисунка 4-11/Q.931.

7.2.2.1.1 Кодирование возможностей переноса "по умолчанию"

Объекты H.323 должны кодировать IE Возможность переноса следующим образом, если не указано другое в последующих разделах.

Бит расширения для октета № 3 (бит 8)

– Должен быть установлен в "1".

Стандарт кодирования (октет № 3, биты 6 и 7)

– Должны быть установлены в "00", указывая "МСЭ-Т".

Возможность переноса информации (октет № 3, биты 1–5)

– Для соединений, начатых конечной точкой ЦСИС, должна быть транслирована информация, указанная для шлюза.

  ПРИМЕЧАНИЕ. – Это позволяет к конечной точке H.323 транслировать некоторую предварительную информацию о природе соединения, например, "Только голос без данных и без видео"; это будет влиять на необходимую полосу пропускания, а также на способность/готовность принять или не принять соединение.

– В соединениях, начатых конечной точкой H.323, это поле должно использоваться для указания, что они желают поместить аудиовизуальное соединение. Поэтому поле должно быть установлено либо в "Неограниченная цифровая информация", то есть "01000", либо в "Ограниченная цифровая информация", то есть "01001". Если должно быть помещено соединение только для речи, то терминал H.323 должен установить Возможность переноса информации либо в "Голос" (то есть "00000"), либо в "3,1 кГЦ аудио" (то есть "10000").

Бит расширения для октета № 4 (бит 8)

– Должен быть установлен в "0", если Скорость переноса информации установлена в "Много скоростей"; должен быть установлен в "1" в остальных случаях.

Режим переноса (октет № 4, биты 6 и 7)

– Должны указывать "Канальный режим", значение "00".

Скорость переноса информации (октет № 4, биты 1–5)

– Должна кодироваться согласно таблице 4-6/Q.931, за исключением того, что значение "00000" (для пакетного режима) не разрешается, если шлюз не соединен с пакетной сетью.

Множитель скорости (октет № 4.1)

– Должен присутствовать, если Скорость переноса информации установлена в "Много скоростей".

– Бит расширения (бит 8) должен быть установлен в "1".

– Биты 1–7 должны указывать полосу пропускания, необходимую для соединения, как определяется ниже (отметим, что в отличие от Рекомендации МСЭ-Т Q.931 здесь разрешается значение "0000001").

– В соединении, начатом конечной точкой ЦСИС, шлюз должен просто переслать информацию, которую он получил от ЦСИС.

– В соединении, пришедшем от конечной точки H.324, шлюз должен устанавливать Множитель скорости в 01H.

– В соединении, пришедшем от Ш-ЦСИС, необходимо выполнить некоторое преобразование из Рекомендации МСЭ-Т Q.2931 в Рекомендацию МСЭ-Т Q.931. Это остается для изучения.

– В соединении, начатом конечной точкой H.323, множитель скорости должен использоваться для указания полосы пропускания, которую следует использовать для этого соединения. Если вызываемой системой является другая конечная точка H.323, то это значение может отражать полосу пропускания, которую следует использовать в пакетной сети, но принимающий терминал может не учитывать эту информацию. Если участвует шлюз, то это значение должно отражать число внешних соединений, которое следует установить. Полоса пропускания, необходимая для соединения, является полосой пропускания, необходимой на стороне SCN, и может соответствовать или не соответствовать полосе пропускания, разрешенной в пакетной сети сообщениями ACF/BCF.

Протокол уровня 1 (октет № 5)

– Бит расширения (бит 8) должен быть установлен в "1".

– Биты 6 и 7 должны указывать идентификатор уровня 1, то есть "01".

– Биты 1–5 должны указывать протокол уровня 1.

– Разрешенными значениями являются G.711 ("00011" для закона A и "00010" для закона μ) при указании на соединение "Только для голоса", а также H.221 и H.242 ("00101") при указании на "Видеофонное соединение" H.323.

Октеты № 5a, 5b, 5c, 5d, 6 и 7 не должны присутствовать.

7.2.2.1.2 Кодирование Возможностей переноса для соединений сигнализации H.450.1, не зависящих от соединения

Конечные точки H.323 должны кодировать IE Возможности переноса для соединений сигнализации, не зависящих от соединения, как определено в Рекомендации МСЭ-Т H.450.1, следующим образом.

Бит расширения для октета № 3 (бит 8)

– Должен быть установлен в "1".

Стандарт кодирования (октет № 3, биты 6 и 7)

– Должен быть установлен в "01", что указывает "Другой международный стандарт". Отметим, что если этот стандарт кодирования указывается, то должна применяться Рекомендация МСЭ-Т Q.931 для октетов 1 и 2 и бита 8 в октетах 3 и 4. Должны кодироваться, как указано, Возможности переноса информации, Режим переноса и Скорость переноса информации, а остальные октеты не включаются.

Возможности переноса информации (октет № 3, биты 1–5)

– Должно устанавливаться в "01000", что указывает "Неограниченная цифровая информация".

Бит расширения для октета № 4 (бит 8)

– Должен быть установлен в "1".

Режим переноса (октет № 4, биты 6 и 7)

– Должен быть установлен в "00", что указывает "Соединение сигнализации, не зависящее от соединения".

Скорость переноса информации (октет № 4, биты 1–5)

– Должен быть установлен в "00000", что указывает "Соединение сигнализации, не зависящее от соединения".

Октеты 4.1 и выше не должны включаться.

7.2.2.2 Идентификатор соединения

Возможное использование IE Идентификатор соединения остается для изучения. При этом изучении следует рассмотреть многостадийный набор номера, в том числе терминал-гейткипер-терминал и терминал-шлюз-терминал, а также свободную маршрутизацию от источника.

7.2.2.3 Состояние соединения

Этот информационный элемент кодируется согласно рисунку 4-13/Q.931.

Стандарт кодирования (октет № 3, биты 8 и 7)

– Установить в "00", что указывает "Стандартизованное МСЭ-Т кодирование".

Значение состояния соединения (октет № 3, биты 1–6)

– Установить согласно таблице 4-8/Q.931, но не использовать Глобальные значения состояния интерфейса. Значения распознаются как Состояние пользователя, применяя Приложение D/Q.931. Заметим, что большинство из перечисленных кодов не будут генерироваться в терминале H.323.

7.2.2.4 Номер вызываемой стороны

Этот информационный элемент кодируется согласно рисунку 4-14/Q.931 и таблице 4-9/Q.931.

Расширение для октета № 3 (бит 8)

– Установить в "1".

Тип номера (октет № 3, биты 5–7)

– Кодируется согласно значениям и правилам из таблицы 4-9/Q.931.

Идентификация плана нумерации (октет № 3, биты 1–4)

– Кодируется согласно значениям и правилам из таблицы 4-9/Q.931. Номер в форме набираемой цепочки цифр следует кодировать как "0000" (Неизвестен). Если установлена в "1001" (Частный план нумерации) в соединении, начатом пакетной сетью, то это указывает, что:

  1) в сообщении Установить отсутствует набираемая цепочка цифр; и

  2) сообщение будет маршрутизировано по адресу-псевдониму в информации "Пользователь-пользователь".

Тип номера (октет № 3, биты 5–7)

– Кодируется согласно значениям и правилам из таблицы 4-9/Q.931. Номер при Идентификации плана нумерации, закодированной как "0000" (Неизвестен), должен кодироваться как "000" (Неизвестен). Номер при Идентификации плана нумерации, закодированной как "0001" (ЦСИС/Телефонный план нумерации, Рекомендация МСЭ-Т E.164), с Типом номера, закодированным как "000" (Неизвестен), может использоваться для обратной совместимости (с прежними версиями Рекомендации).

"Цифры" номера

– Любое число знаков Международного алфавита № 5 (МА5) согласно форматам, указанным в соответствующем плане нумерации/набора номера.

ПРИМЕЧАНИЕ. – Номер по E.164 должен содержать только знаки МА5 "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" и "0".

7.2.2.5 Субадрес вызываемой стороны

Используется согласно Рекомендации МСЭ-Т Q.931.

7.2.2.6 Номер вызывающей стороны

Этот информационный элемент кодируется согласно рисунку 4-16/Q.931 и таблице 4-11/Q.931.

Тип номера (октет № 3, биты 5–7)

– Кодируется согласно значениям и правилам из таблицы 4-11/Q.931. Номер при Идентификации плана нумерации, закодированной как "0000" (Неизвестен), должен кодироваться как "000" (Неизвестен). Номер при Идентификации плана нумерации, закодированной как "0001" (ЦСИС/Телефонный план нумерации, Рекомендация МСЭ-Т E.164), с Типом номера, закодированным как "000" (Неизвестен), может использоваться для обратной совместимости.

Идентификация плана нумерации (октет № 3, биты 1–4)

– Кодируется согласно значениям и правилам из таблицы 4-11/Q.931. Номер в форме набираемой цепочки цифр следует кодировать как "0000" (Неизвестен). Если установлена в "1001" (Частный план нумерации) в соединении, начатом пакетной сетью, то это указывает, что:

  1) в сообщении Установить отсутствует набираемая цепочка цифр; и

  2) сообщение будет маршрутизировано по адресу-псевдониму в информации "Пользователь-пользователь".

Октет № 3a

– Кодируется согласно значениям и правилам из таблицы 4-11/Q.931.

"Цифры" номера

– Любое число знаков MA5 согласно форматам, указанным в соответствующем плане нумерации/набора номера.

ПРИМЕЧАНИЕ. – Номер по E.164 должен содержать только знаки MA5 "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" и "0".

Конечные точки H.323 не должны передавать несколько IE Номер вызываемой стороны в одном и том же сообщении. Шлюзы могут обеспечивать поддержку взаимодействия с сообщениями УСТАНОВИТЬ Q.931, которые содержат несколько IE Номер вызываемой стороны. Шлюзы, которые обеспечивают такую поддержку, должны отображать первый IE Номер вызывающей стороны Q.931 в IE Номер вызывающей стороны для сообщения Установить H.225.0 и отображать последующие IE Номер вызывающей стороны в поле **additionalSourceAddresses** этого сообщения Установить H.225.0.

7.2.2.7 Субадрес вызывающей стороны

Используется согласно Рекомендации МСЭ-Т Q.931.

7.2.2.8 Причина

Если причина получена, то применяются правила из Рекомендации МСЭ-Т Q.850. Заметим, что для Освобождения Завершено обязательна либо IE Причина, либо **ReleaseCompleteReason**; IE Причина факультативен в других сообщениях. IE Причина и **ReleaseCompleteReason** (как часть сообщения Освобождение Завершено) являются взаимно исключающими. Шлюз должен отображать **ReleaseCompleteReason** в IE Причина, когда передает сообщение Освобождение Завершено к стороне с коммутацией каналов от стороны пакетной сети (см. таблицу 5). (Обратное преобразование не требуется, так как объекты пакетной сети обязаны декодировать IE Причина.)

Таблица 5/H.225.0 – Отображение ReleaseCompleteReason в IE Причина

Код ReleaseCompleteReason	Соответствующее значение причины Q.931/Q.850
noBandwidth	34 – Нет доступного канала
gatekeeperResources	47 – Ресурс недоступен, без подробностей
unreachableDestination	3 – Нет маршрута к адресату
destinationRejection	16 – Нормальное разъединение соединения
invalidRevision	88 – Несовместимый адресат
noPermission	127 – Взаимодействие, без подробностей
unreachableGatekeeper	38 – Сеть повреждена
gatewayResources	42 – Перегрузка коммутационного оборудования
badFormatAddress	28 – Неправильный формат номера (неполный адрес)
adaptiveBusy	41 – Временная неисправность
inConf	17 – Пользователь занят
undefinedReason	31 – Нормально, без подробностей
facilityCallDeflection	16 – Нормальное разъединение соединения
securityDenied	31 – Нормально, без подробностей
securityWrongSyncTime	31 – Нормально, без подробностей
securityReplay	31 – Нормально, без подробностей
securityWrongGeneralID	31 – Нормально, без подробностей
securityWrongSendersID	31 – Нормально, без подробностей
securityMessageIntegrityFailed	31 – Нормально, без подробностей
securityWrongOID	31 – Нормально, без подробностей
securityDHmismatch	31 – Нормально, без подробностей
securityCertificateExpired	31 – Нормально, без подробностей
securityCertificateDateInvalid	31 – Нормально, без подробностей
securityCertificateRevoked	31 – Нормально, без подробностей
securityCertificateNotReadable	31 – Нормально, без подробностей
securityCertificateSignatureInvalid	31 – Нормально, без подробностей
securityCertificateMissing	31 – Нормально, без подробностей
securityCertificateIncomplete	31 – Нормально, без подробностей
securityUnsupportedCertificateAlgOID	31 – Нормально, без подробностей
securityUnknownCA	31 – Нормально, без подробностей
calledPartyNotRegistered	20 – Абонент отсутствует
callerNotRegistered	31 – Нормально, без подробностей
newConnectionNeeded	47 – Ресурс недоступен, без подробностей
nonStandardReason	127 – Взаимодействие, без подробностей
replaceWithConferenceInvite	31 – Нормально, без подробностей
genericDataReason	31 – Нормально, без подробностей
neededFeatureNotSupported	31 – Нормально, без подробностей
tunnelledSignallingRejected	127 – Взаимодействие, без подробностей
InvalidCID	3 – Нет маршрута к адресату
hopCountExceeded	3 – Нет маршрута к адресату

Шлюзы должны также отображать **AdmissionRejectReason** и **LocationRejectReason** в IE Причина при передаче сообщения Освобождение Завершено к стороне пакетной сети после приема **AdmissionReject** или **LocationReject** (таблица 6).

Таблица 6/H.225.0 – Отображение AdmissionRejectReason/LocationRejectReason в IE Причина

Код AdmissionRejectReason или LocationRejectReason	Соответствующее значение причины Q.931/Q.850
calledPartyNotRegistered	20 – Абонент отсутствует
invalidPermission	127 – Взаимодействие, без подробностей
requestDenied	31 – Нормально, без подробностей
undefinedReason	31 – Нормально, без подробностей
callerNotRegistered	31 – Нормально, без подробностей
routeCallToGatekeeper	Не применимо
invalidEndpointIdentifier	127 – Взаимодействие, без подробностей
resourceUnavailable	47 – Ресурс недоступен, без подробностей
securityDenial	31 – Нормально, без подробностей
qosControlNotSupported	63 – Услуга или вариант недоступны, без подробностей
incompleteAddress	28 – Неправильный формат номера
aliasesInconsistent	31 – Нормально, без подробностей
routeCallToSCN	3 – Нет маршрута к адресату
exceedsCallCapacity	41 – Временная неисправность
collectDestination	31 – Нормально, без подробностей
collectPIN	31 – Нормально, без подробностей
genericDataReason	31 – Нормально, без подробностей
neededFeatureNotSupported	31 – Нормально, без подробностей
securityWrongSyncTime	31 – Нормально, без подробностей
securityReplay	31 – Нормально, без подробностей
securityWrongGeneralID	31 – Нормально, без подробностей
securityWrongSendersID	31 – Нормально, без подробностей
securityIntegrityFailed	31 – Нормально, без подробностей
securityWrongOID	31 – Нормально, без подробностей
secuirtyDHMismatch	31 – Нормально, без подробностей
noRouteToDestination	3 – Нет маршрута к адресату
unallocatedNumber	1 – Неприсвоенный номер

7.2.2.9 Идентификация канала

Использование этого IE остается для изучения; может использоваться для обеспечения обратной связи при многократных попытках соединения.

7.2.2.10 Соединенный номер

Кодируется согласно 4.1/Q.951.5.

7.2.2.11 Соединенный субадрес

Кодируется согласно 4.2/Q.951.5.

7.2.2.12 Уровень перегрузки

Не должен использоваться.

7.2.2.13 Дата/время

Кодируется согласно рисунку 4-21/Q.931.

7.2.2.14 Отображение

Кодируется согласно рисунку 4-22/Q.931. Максимальная длина этого полного информационного элемента составляет 82 октета.

7.2.2.15 Информационный элемент Расширенные возможности

Любой IE Расширенные возможности, который используется для указания неизмененной семантики, определенной в Рекомендациях серии Q.95.x, должен кодироваться согласно 8.2.4/Q.932. В этом случае сервисные блоки ADU должны формироваться согласно ROSE (Remote Operations Service Element, сервисный элемент удаленных операций), определенному в Рекомендации МСЭ-Т X.229 (с использованием Рекомендации МСЭ-Т X.680 (Спецификация ASN.1) и Рекомендации МСЭ-Т X.690 (Спецификация базовых правил кодирования для ASN.1)).

7.2.2.16 Возможности

Чтобы передавать процедуры перенаправления вызова, специфичные для H.323 (переадресация вызова, перенаправление вызова к MC или принудительная маршрутизация вызова к гейткиперу), или в случае сигнализации о дополнительной услуге согласно Рекомендации МСЭ-Т H.450 используется информационный элемент Пользователь-пользователь в сообщении Возможности. Этот частный случай должен указываться кодированием IE Возможности с длиной нуль; то есть информационный элемент Возможности должен содержать точно 2 следующих октета:

- Октет № 1 (Идентификатор информационного элемента) должен быть установлен в "00011100" (шестнадцатеричное "1C") для указания IE Возможности.
- Октет № 2 (Длина информационного элемента) должен быть установлен в "0" для указания, что далее не следуют октеты, принадлежащие к этому информационному элементу.

Для указания на переприем вызова IE Возможности должен быть пустым, а **Facility-UUIE** должен указывать в **alternativeAddress** или **alternativeAliasAddress** тот терминал, к которому вызов должен быть перенаправлен. В этом случае **facilityReason** должна быть установлена в **callForwarded**.

Чтобы сообщить конечной точке о необходимости вызывать другую конечную точку, так как вызывающая конечная точка желает присоединиться к конференции, а вызываемая конечная точка не имеет MC, IE Возможности также должен оставаться пустым. Компонент **conferenceID** должен указывать конференцию для присоединения, а причиной в **Facility-UUIE** должна быть **routeCallToMC**.

IE Возможности остается пустым также для сообщения вызывающей конечной точке о необходимости связи с вызываемой конечной точкой через гейткипер этой вызываемой конечной точки. Компонент **conferenceID** в **Facility-UUIE** должен указывать конференцию для присоединения, а причиной в **Facility-UUIE** должна быть **routeCallToGatekeeper**.

Любой IE Возможности, который используется для указания неизменяемой семантики, определенной в Рекомендациях серии Q.95.x, должен кодироваться согласно 8.2.3/Q.932. В этом случае сервисные блоки ADU должен формироваться согласно ROSE, определенному в Рекомендации МСЭ-Т X.229 (с использованием Рекомендации МСЭ-Т X.680 (Спецификация ASN.1) и Рекомендации МСЭ-Т X.690 (Спецификация базовых правил кодирования для ASN.1)).

7.2.2.17 Совместимость верхних уровней

FFS.

7.2.2.18 Возможности клавиатуры

Кодируется согласно рисунку 4-24/Q.931. Использование символа восклицательного знака "!" должно представлять указание "Кратковременный отбой" (hookflash). Конечные точки, не поддерживающие прием указания "Кратковременный отбой", должны игнорировать "!", если он будет получен.

7.2.2.19 Совместимость нижних уровней

FFS.

7.2.2.20 Информационный элемент "Еще данные"

Не должен использоваться.

7.2.2.21 Возможности, специфичные для сети

Не должен использоваться.

7.2.2.22 Указатель уведомления

Кодируется согласно 4.5.22/Q.931.

7.2.2.23 Указатель прохождения

Кодируется согласно рисунку 4-29/Q.931 и таблице 4-20/Q.931.

Этот информационный элемент требуется только для стыковки терминала H.323 с терминалом на основе ЦСИС или ATM, где доступна детальная информация о готовности соединения. В этом случае шлюз должен направлять эту информацию к терминалу H.323. Оконечная система H.323 не нуждается в распознавании этого информационного элемента.

Если этот информационный элемент генерируется терминалом H.323, то применяются следующие ограничения:

Стандарт кодирования (октет № 3, биты 6 и 7)

– Должно указываться "МСЭ-Т" ("00").

Местонахождение

– Согласно таблице 4-20/Q.931.

– Разрешены значения "Пользователь" ("0000"), "Частная сеть, обслуживающая местного пользователя" ("0001") и "Частная сеть, обслуживающая удаленного пользователя" ("0101").

Описание прохождения

– Согласно таблице 4-20/Q.931.

7.2.2.24 Номер перенаправления

Кодируется согласно 4.6.7/Q.931. Заметим, что этот IE обеспечивается только для облегчения взаимодействия с SCN, а не для обеспечения какого-либо механизма для услуг изменения направления вызова. Услуги изменения направления вызова для H.323 определены в Рекомендации МСЭ-Т H.450.3.

7.2.2.25 Указатель повторения

Не должен использоваться.

7.2.2.26 Указатель рестарта

Не должен использоваться.

7.2.2.27 Сегментированное сообщение

Не должно использоваться. Заметим, что нет критичного верхнего предела для размера сообщения в Рекомендации МСЭ-Т H.323 и в настоящей Рекомендации.

7.2.2.28 Передача завершена

Кодируется согласно рисунку 4-33/Q.931.

Никакие ограничения не применяются.

7.2.2.29 Сигнал

Кодируется согласно рисунку 4-34/Q.931 и таблице 4-24/Q.931.

Никакие ограничения не применяются.

7.2.2.30 Выбор транзитной сети

Не должен использоваться.

7.2.2.31 Пользователь-пользователь

Кодируется согласно рисунку 4-36/Q.931 и таблице 4-26/Q.931, учитывая нижеприведенные изменения.

Информационный элемент Пользователь-пользователь должен применяться всеми объектами H.323 для переноса информации, относящейся к H.323. Реальная информация "пользователь-пользователь", которую следует передавать только между участвующими терминалами, вкладывается в поле **user-data** в PDU **H323-UserInformation** (к которому никакие ограничения не применяются).

Применяются следующие ограничения:

Длина содержимого "пользователь-пользователь"

– Должно быть 2 октета вместо 1 (показанного на рисунке 4-36/Q.931).

Дискриминатор протокола

– Должен указывать "Информация пользователя, кодированная по Рекомендациям МСЭ-Т X.680 и X.690 (ASN.1)" ("00000101").

ПРИМЕЧАНИЕ. – Это взято из Рекомендации МСЭ-Т Q.931 версии 1998 года, которая ссылается на более ранние версии ASN.1. Правильными ссылками на ASN.1 являются Рекомендации МСЭ-Т X.680 (синтаксис) и X.691 (кодирование PER).

Информация пользователя

– Должна содержать структуру ASN.1 (**H323-UserInformation**), которая кроме информации, относящейся к H.323, содержит реальные данные пользователя, как показано ниже. ASN.1 кодируется с использованием варианта ALIGNED правил уплотненного кодирования (PER), определенных в Рекомендации МСЭ-Т X.691.

Структура **H323-UserInformation** содержит поля **h323-uu-pdu** и **user-data**.

Поле **h323-uu-pdu** структуры **H323-UserInformation** содержит следующие поля. Заметим, что не все поля в **h323-uu-pdu** разрешаются в каждом сообщении. См. ограничения в описании каждого конкретного сообщения.

– **h323-message-body** – Это поле содержит информацию, специфичную для конкретного сообщения сигнализации H.225.0 и описанную в 7.3 и 7.4. Передатчик может выбрать вариант **empty**, если нет необходимости передать поле UUIE (**Facility-UUIE** и др.) в конкретном сообщении, например, когда используется сообщение Возможности для транспортировки информации, не связанной с соединением. Заметим, что начиная с версии 4 этой Рекомендации передатчик должен включать поле UUIE, если сообщение связано с конкретным соединением. Это нужно для обеспечения поля **callIdentifier**.

– **nonStandardData** – Это поле переносит информацию, не определенную в настоящей Рекомендации (например, данные о праве собственности).

– **h4501SupplementaryService** – Это поле переносит последовательность Протокольных блоков данных прикладного уровня (Application Protocol Data Unit, APDU) H4501SupplementaryService, определенных в таблице 3/H.450.1.

– **h245Tunnelling** – Этот элемент устанавливается в ИСТИНА, если имеется возможность туннелирования сообщений H.245. Системы, соответствующие H.225.0 версии 4 или выше, должны устанавливать этот элемент в ИСТИНА, если для установления соединения используется процедура Быстрое соединение.

– **h245Control** – Это поле переносит последовательность туннелированных PDU H.245. Каждая цепочка октета должна содержать точно один PDU H.245.

– **nonStandardControl** – Это поле содержит информацию управления, не определенную в этой Рекомендации (например, информацию об управлении правом собственности).

– **callLinkage** – Содержимое этого поля обычно управляется какой-либо услугой сцепления соединений. Процедуры и семантику этого поля см. в Рекомендации МСЭ-Т H.323.

– **tunnelledSignallingMessage** – Туннелированное полное сообщение сигнализации в его собственном формате для поддержки дополнительной сигнализации управления соединением "от конца до конца". Поле **tunnelledProtocolID** указывает протокол, подлежащий туннелированию. Поле **messageContent** является последовательностью реальных полных туннелированных сообщений с их собственным двоичным форматом; это позволяет объединять туннелированные сообщения в одно сообщение H.225.0. Если присутствует поле **tunnellingRequired**, то соединение должно продолжаться только при поддержке туннелирования.

– **provisionalRespToH245Tunnelling** – Этот флаг используется для сообщения о том, что вызываемый объект еще не решил, применимо ли туннелирование H.245 для этого соединения. Если это поле присутствует, то флаг **h245Tunnelling** должен игнорироваться принимающим объектом.

– **stimulusControl** – Это поле зарезервировано для будущего использования в МСЭ-Т протокола на основе стимула.

– **genericData** – Это поле является списком общих элементов, относящихся к свойствам, которые определяются вне основной спецификации H.225.0. Эти параметры могут использоваться, например, для информации о туннелировании, прозрачной для H.225.0.

Поле **user-data** структуры **H323-UserInformation** содержит следующие поля:

– **protocol-discriminator** – Это поле кодируется согласно таблице 4-26/Q.931.

– **user-information** – Это поле кодируется согласно 4.5.30/Q.931.

7.3 Детали сообщений сигнализации о соединении H.225.0 на базе Q.931

Заметим, что длины информационных элементов, указанные в приведенных ниже таблицах, относятся к сообщениям, которые генерируются только терминалами H.323. Размер показанного информационного элемента Пользователь-пользователь понимается как размер структуры **user-data** в **H323-UserInformation** и не охватывает **h323-UU-PDU**. Общий размер **H323-UserInformation** ограничен 65 536 октетами. Не учитывая указанные размеры, сообщения со стороны SCN могут иметь другие размеры (более длинные).

Заметим также, что информационные элементы, отмеченные ниже как обязательные, факультативные или запрещенные, относятся лишь к возможности их выдачи терминалами H.323.

7.3.1 Предупреждение

Это сообщение может передаваться вызываемым пользователем для указания, что начато предупреждение вызываемого пользователя. В обычных терминах – "телефонный аппарат вызывается".

Соответствует таблице 3-2/Q.931 (версия 1998 года) с учетом изменений из таблицы 7.

Таблица 7/H.225.0 – Предупреждение

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Возможность переноса	O	5–6
Расширенные возможности	O	8–*
Идентификация канала	FFS	н. д.
Возможности	O	8–*
Указатель прохождения	O	2–4
Указатель уведомления	O	2–*

Таблица 7/H.225.0 – Предупреждение

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Отображение	O	2–82
Сигнал	O	2–3
Совместимость верхних уровней	FFS	н. д.
Пользователь-пользователь	M	2–131

Информационный элемент Пользователь-пользователь содержит Alerting-UUIE, определенный в синтаксисе сообщения H.225.0. Компонент **Alerting-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

destinationInfo – Содержит **EndpointType**, позволяющий вызывающей стороне определять, включен ли шлюз в соединение.

h245Address – Это конкретный транспортный адрес, по которому вызываемая конечная точка или гейткипер, обрабатывающий соединение, собирается установить сигнализацию H.245. Этот адрес может передаваться также в сообщении Готовность Вызова, Прохождение или Соединить.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

h245SecurityMode – Объект H.323, который получает сообщение Установить с набором **h245SecurityCapability**, должен отвечать соответствующим приемлемым **h245SecurityMode** в сообщении Готовность Вызова, Предупреждение, Прохождение или Соединить**.**

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Используемый только в процедуре быстрого соединения, **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получать потоки носителей.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

alertingAddress – Содержит адреса-псевдонимы предупреждающей стороны.

presentationIndicator – Указывает, что представление **alertingAddress** следует разрешить или ограничить.

screeningIndicator – Указывает, что **alertingAddress** был выдан конечной точкой или сетью (гейткипером), и указывает, был ли **alertingAddress** проверен в гейткипере.

fastConnectRefused – Вызываемая конечная точка будет передавать этот элемент в любом сообщении до сообщения Соединить, включая его, при установлении соединения для указания, что она отказывается от процедуры Быстрое соединение.

serviceControl – Содержит данные, специфичные для услуги (либо ссылки на них), которые могут использоваться вызывающей оконечной точкой как часть процедуры установления (к примеру, меню опций для изменения направления соединения), как описано, например, в Приложении K/H.323.

capacity – Это поле указывает пропускную способность соединения, доступную передающей конечной точке в этот момент времени, предполагая, что это сообщение Предупреждение представляет реальное соединение. Когда конечная точка передает это поле, она должна включать элемент **currentCallCapacity**.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

7.3.2 Готовность вызова

Это сообщение может передаваться вызываемым пользователем для указания, что установление запрошенного соединения начато и что далее никакая информация об установлении соединения не будет приниматься. См. таблицу 8.

Таблица 8/H.225.0 – Готовность вызова

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Возможность переноса	O	5-6
Расширенные возможности	O	8–*
Идентификация канала	FFS	н. д.
Возможности	O	8–*
Указатель прохождения	O	2–4
Указатель уведомления	O	2–*
Отображение	O	2–82
Совместимость верхних уровней	FFS	н. д.
Пользователь-пользователь	M	2–131

Информационный элемент Пользователь-пользователь содержит **CallProceeding-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **CallProceeding-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

destinationInfo – Содержит **EndpointType**, позволяющий вызывающей стороне определять, включен ли шлюз в соединение.

h245Address – Это конкретный транспортный адрес, по которому вызываемая конечная точка или гейткипер, обрабатывающий соединение, собирается установить сигнализацию H.245.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

h245SecurityMode – Объект H.323, который получает сообщение Установить с набором **h245SecurityCapability**, должен отвечать соответствующим приемлемым **h245SecurityMode** в сообщении Готовность Вызова, Предупреждение, Прохождение или Соединить.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Используемый только в процедуре быстрого соединения **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получать потоки носителей.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

fastConnectRefused – Вызываемая конечная точка будет передавать этот элемент в любом сообщении до сообщения Соединить, включая его, при установлении соединения для указания, что она отказывается от процедуры Быстрое соединение.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

7.3.3 Соединить

Это сообщение должно передаваться вызываемой конечной точкой к вызывающей конечной точке (гейткиперу, шлюзу или вызывающему терминалу), чтобы указать на приемлемость этого вызова для вызываемого объекта. Соответствует таблице 3-4/Q.931 с учетом изменений из таблицы 9.

Таблица 9/H.225.0 – Соединить

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Возможность переноса	O	5–6
Расширенные возможности	O	8–*
Идентификация канала	FFS	н. д.
Возможности	O	8–*
Указатель прохождения	O	2–4
Указатель уведомления	O	2–*
Отображение	O	2–82
Дата/Время	O	8
Соединенный номер	O	2–*
Соединенный субадрес	O	2–23
Совместимость нижних уровней	FFS	н. д.
Совместимость верхних уровней	FFS	н. д.
Пользователь-пользователь	M	2–131

Информационный элемент Пользователь-пользователь содержит **Connect-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **Connect-UUIE** содержит:

protocolIdentifier – Вызываемая конечная точка указывает поддерживаемую версию H.225.0.

h245Address – Это конкретный транспортный адрес, по которому вызываемая конечная точка или гейткипер, обрабатывающий соединение, собирается установить сигнализацию H.245. Этот адрес должен передаваться, если был передан ранее в сообщении Предупреждение, Прохождение или Готовность Вызова.

destinationInfo – Содержит **EndpointType**, позволяющий вызывающей стороне определить, включен ли шлюз в соединение.

conferenceID – Будет содержать уникальный номер, позволяющий однозначно определить конференцию среди других, полученных в сообщении Установить.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

h245SecurityMode – Объект H.323, который получает сообщение Установить с набором **h245SecurityCapability**, должен отвечать соответствующим приемлемым **h245SecurityMode** в сообщении Готовность Вызова, Предупреждение, Прохождение или Соединить.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Используемый только в процедуре быстрого соединения **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получать потоки носителей.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

language – Указывает язык(и), на котором(ых) пользователь предпочел бы получать объявления и тревожные сообщения. Это поле содержит один или несколько маркеров (меток) языка, соответствующих RFC 1766.

connectedAddress – Содержит адреса-псевдонимы для соединенной (ответившей) стороны; набранная цепочка цифр этой соединенной стороны содержится в IE Соединенный номер.

presentationIndicator – Указывает, что представление поля **connectedAddress** следует разрешить или ограничить. Если присутствуют как **presentationIndicator**, так и указатель представления в IE Соединенный номер, но не совпадают, то должен использоваться указатель представления из IE Соединенный номер.

screeningIndicator – Указывает, что **connectedAddress** был выдан конечной точкой или сетью (гейткипером), и указывает, был ли **connectedAddress** проверен гейткипером. Если присутствуют как **screeningIndicator**, так и указатель проверки в IE Соединенный номер, но не совпадают, то должен использоваться указатель проверки из IE Соединенный номер.

fastConnectRefused – Вызываемая конечная точка будет передавать этот элемент в любом сообщении до сообщения Соединить, включая его, при установлении соединения для указания, что она отказывается от процедуры Быстрое соединение.

serviceControl – Содержит данные, специфичные для услуги (либо ссылки на них), которые могли бы использоваться конечной точкой или шлюзом (к примеру, для отображения меню опций для вызывающей стороны), как описано, например, в Приложении K/H.323.

capacity – Это поле указывает пропускную способность соединения, доступную передающей конечной точке в этот момент времени, предполагая, что это сообщение Соединить представляет реальное соединение. Когда конечная точка передает это поле, она должна включать элемент **currentCallCapacity**.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

7.3.4 Подтверждение соединения

Это сообщение не должно передаваться.

7.3.5 Разъединить

Это сообщение не должно передаваться объектом H.323.

Содержимое и семантика сообщения Разъединить, полученного от сети, определены в таблице 3-6/Q.931 и в 10.5 Стандарта ISO/IEC 11582.

7.3.6 Информация

Это сообщение может передаваться для предоставления дополнительной информации. Оно может использоваться для предоставления информации для установления соединения (например, при совмещенной передаче) либо различной информации, относящейся к соединению. Оно может использоваться для доставки свойств, связанных с правом собственности.

Это сообщение может передаваться каким-либо объектом H.323.

Это сообщение соответствует таблице 3-7/Q.931 с изменениями, показанными в таблице 10.

Таблица 10/H.225.0 – Содержимое сообщения Информация

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Передача завершена	O	1
Отображение	O	2–82
Возможности клавиатуры	O	2–34
Сигнал	O	2–3
Номер вызываемой стороны	O (примечание)	2–35
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ. – IE Номер вызываемой стороны будет использоваться для переноса номеров из какого-либо Частного плана нумерации, когда выполняется совмещенная передача согласно 8.1.12/H.323.		

Информационный элемент Пользователь-пользователь содержит **Information-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонет **Information-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Это поле не должно включаться и должно игнорироваться при получении.

fastConnectRefused – Это поле не должно включаться и должно игнорироваться при получении.

circuitInfo – Это поле предоставляет информацию о канале или каналах SCN, используемых для этого соединения.

7.3.7 Прохождение

Это сообщение может передаваться шлюзом H.323 для указания на прохождение вызова в случае взаимодействия с SCN. Это сообщение может передаваться также конечной точкой H.323 перед сообщением Соединить в зависимости от взаимодействия дополнительной услуги.

Соответствует таблице 3-9/Q.931 и 10.10 стандарта ISO/IEC 11582 с учетом изменений из таблицы 11.

Таблица 11/H.225.0 – Прохождение

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Возможность переноса	O	5–6
Причина	O	2–32
Расширенные возможности	O	8–*
Идентификация канала	FFS	н. д.
Возможности	O	8–*
Указатель прохождения	M	2–4
Указатель уведомления	O	2–*
Отображение	O	2–82
Совместимость верхних уровней	FFS	н. д.
Пользователь-пользователь	M	2–131

Информационный элемент Пользователь-пользователь содержит **Progress-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **Progress-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

destinationInfo – Содержит **EndpointType**, позволяющий вызывающей стороне определить, включен ли шлюз в соединение.

h245Address – Это конкретный транспортный адрес, по которому вызываемая конечная точка или гейткипер, обрабатывающий соединение, собирается установить сигнализацию H.245. Этот адрес должен передаваться, если был передан ранее в сообщении Готовность Вызова, Предупреждение или Соединить.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

h245SecurityMode – Объект H.323, который получает сообщение Установить с набором **h245SecurityCapability**, должен отвечать соответствующим приемлемым **h245SecurityMode** в сообщении Готовность Вызова, Предупреждение, Прохождение или Соединить.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Используемый только в процедуре быстрого соединения **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получать потоки носителей.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

fastConnectRefused – Вызываемая конечная точка будет передавать этот элемент в любом сообщении до сообщения Соединить, включая его, при установлении соединения для указания, что она отказывается от процедуры Быстрое соединение.

7.3.8 Освободить

Это сообщение не должно передаваться объектом H.323.

Содержимое и семантика принятого сообщения Освободить определены в таблице 3-10/Q.931 и 10.5 Стандарта ISO/IEC 11582.

7.3.9 Освобождение Завершено

Это сообщение должно передаваться терминалом для указания на освобождение соединения. После этого Значение справочного номера соединения (Call Reference Value, CRV) доступно для нового использования.

Последовательность Разъединить/Освободить/Освобождение Завершено не используется, поскольку ее главной целью является указание о завершении освобождения ресурсов с коммутацией каналов. Так как она не применима к среде с пакетной сетью, используется одношаговый метод с передачей только Освобождение Завершено.

Соответствует таблице 3-11/Q.931. Применяются изменения из таблицы 12.

Таблица 12/H.225.0 – Освобождение Завершено

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Причина	CM (примечание)	2–32
Возможности	O	8–*
Указатель уведомления	O	2–*
Отображение	O	2–82
Сигнал	O	2–3
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ. – Должен присутствовать либо IE Причина, либо элемент **ReleaseCompleteReason**.		

Если это сообщение передается в ответ на сообщение Возможности с пустым IE Возможности, то **ReleaseCompleteReason** должно быть установлено в **facilityCallDeflection**.

Если это сообщение передано шлюзом из SCN, то значение причины должно быть установлено согласно Рекомендации МСЭ-Т Q.931.

Информационный элемент Пользователь-пользователь содержит **ReleaseComplete-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **ReleaseComplete-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

reason – Различная информация о причине освобождения соединения. Причина **genericDataReason** указывает, что соединение получило отбой из-за общего элемента или свойства; в этом случае дополнительная информация может быть указана в поле **genericData** в **h323-uu-pdu** этого сообщения. Причина **neededFeatureNotSupported** указывает, что свойство, запрошенное одним объектом, не поддерживается другим. Причина **tunnelledSignallingRejected** передается, если соединение получило отбой, так как передатчик не разрешает туннелированную сигнализацию не-H.323, а туннелирование запрошено для успешного выполнения соединения. Причина **hopCountExceeded** указывает, что соединение отклонено, так как значение **hopCount** достигло 0 и поэтому соединение не может продолжаться далее.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

busyAddress – Содержит адреса-псевдонимы для занятой стороны.

presentationIndicator – Указывает, что представление **busyAddress** следует разрешить или ограничить.

screeningIndicator – Указывает, что **busyAddress** был выдан конечной точкой или сетью (гейткипером), и указывает, был ли **busyAddress** проверен гейткипером.

capacity – Указывает пропускную способность соединения, доступную передающей конечной точке после того, как соединение, указанное в этом сообщении Освобождение Завершено, было освобождено. Когда конечная точка передает это поле, она должна включать элемент **currentCallCapacity**.

serviceControl – Содержит данные, специфичные для услуги (либо ссылки на них), для услуг "поле соединения" (например, сообщение об ошибке или объявление), как описано, например, в Приложении K/H.323.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

7.3.10 Установить

Это сообщение должно передаваться вызывающим объектом H.323 для указания на его желание установить соединение к вызываемому объекту.

Соответствует таблице 3-15/Q.931 с учетом изменений из таблицы 13.

Таблица 13/H.225.0 – Установить

Информационный элемент	Статус в H.225.0 (M/F/O/CM)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M (примечание 2)	3
Тип сообщения	M	1
Передача завершена	O	1
Указатель повторения	F	н. д.
Возможность переноса	M	5–6
Расширенные возможности	O	8–*
Идентификация канала	FFS	н. д.
Возможности	O	8–*
Указатель прохождения	O	2–4
Услуги, специфичные для сети	F	н. д.
Указатель уведомления	O	2–*
Отображение	O	2–82
Возможности клавиатуры	O	2–34
Сигнал	O	2–3
Номер вызывающей стороны	O	2–131
Субадрес вызывающей стороны	CM (примечание 1)	н. д.
Номер вызываемой стороны	O	2–131
Субадрес вызываемой стороны	CM (примечание 1)	н. д.

Таблица 13/H.225.0 – Установить

Информационный элемент	Статус в H.225.0 (M/F/O/CM)	Длина в H.225.0
Номер перенаправления	O	2–*
Выбор транзитной сети	F	н. д.
Совместимость нижних уровней	FFS	н. д.
Совместимость верхних уровней	FFS	н. д.
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ 1. – Субадреса необходимы для некоторых сценариев соединения SCN; их не следует применять для соединений только на стороне пакетной сети. ПРИМЕЧАНИЕ 2. – Если ранее был передан ARQ, то использованный здесь CRV должен быть тем же самым.		

Информационный элемент Пользователь-пользователь содержит **Setup-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **Setup-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

h245Address – Это конкретный транспортный адрес, по которому вызывающая конечная точка или гейткипер, обрабатывающий соединение, собирается установить сигнализацию H.245. Он предоставляется отправителем только в случае, когда он способен обрабатывать процедуры H.245 до получения сообщения Соединить по каналу сигнализации о соединении.

sourceAddress – Содержит адреса-псевдонимы источника. Основной адрес должен быть первым. Заметим, что номер по E.164 для источника, если он имеется, должен содержаться внутри информационного элемента Номер вызывающей стороны.

sourceInfo – Содержит **EndpointType**, позволяющий вызываемой стороне определить, включен ли шлюз в соединение.

destinationAddress – Это адрес, к которому конечная точка желает получить соединение. Основной адрес должен быть первым. Когда вызывает конечная точка, использующая только цепочку цифр номеронабирателя, этот адрес должен присутствовать в IE Номер вызываемой стороны в сообщении сигнализации о соединении H.225.0. Элемент **destinationAddress**, если он доступен, должен включаться в сообщение Установить терминалами, соответствующими настоящей Рекомендации версии 2 или более.

destCallSignalAddress – Нужен для информирования гейткипера о транспортном адресе сигнализации о соединении у терминала-получателя; избыточен в случае прямой связи "терминал-терминал". Во всех случаях, когда эта информация доступна для отправителя сообщения Установить, это поле должно быть вставлено.

destExtraCallInfo – Нужен для создания возможных дополнительных вызовов канала, то есть для вызова 2 × 64 кбит/с на стороне SCN. Должен содержать только цепочки цифр номеронабирателя, номера по E.164 или частные номера, но не должен содержать номер исходного канала. (См. примечание.)

destExtraCRV – Это значения CRV для дополнительных вызовов к SCN, указанных в **destExtraCallInfo.** Использование этого элемента остается для изучения. Он может использоваться для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

activeMC – Указывает, что вызывающая конечная точки находится под влиянием активного многоточечного контроллера (MC).

conferenceID – Уникальный идентификатор конференции.

conferenceGoal:

- **create** – Начать новую конференцию.
- **invite** – Пригласить участника в существующую конференцию.
- **join** – Присоединиться к существующей конференции.

- **capability-negotiation** – Согласовать возможности для последующей свободно связанной конференции.
- **callIndependentSupplementaryService** – Транспортировать блоки APDU дополнительных услуг способом, не связанным с соединением.

callServices – Предоставляет информацию о поддержке факультативных протоколов серии Q для гейт-кипера и вызываемого терминала.

callType – Используя это значение, гейткипер вызываемой стороны может попытаться определить "реальное" использование полосы пропускания. Безусловным "по умолчанию") значением является **pointToPoint** для всех соединений; следует знать, что тип соединения может динамически изменяться за время существования соединения и что последний тип соединения может быть неизвестен при передаче сообщения Установить.

sourceCallSignalAddress – Содержит транспортный адрес источника; это значение должно использоваться в сообщении ARQ получателем сообщения Установить. Во всех случаях, когда эта информация доступна для отправителя сообщения Установить, это поле должно быть вставлено. Значение **sourceCallSignalAddress** должно быть эквивалентно значению, которое было использовано в ARQ отправителем сообщения Установить, и должно быть возвращено в виде эта конечной точкой, принявшей Установить в ее ARQ.

remoteExtensionAddress – Содержит адрес-псевдоним вызываемой конечной точки в случаях, когда эта информация нужна для прохождения через несколько шлюзов. Во всех случаях, когда эта информация доступна для отправителя сообщения Установить, это поле должно быть вставлено.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

h245SecurityCapability – Набор возможностей, которые отправитель может использовать для обеспечения безопасности в канале H.245.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

fastStart – Используемый только в процедуре быстрого соединения, **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получить потоки носителей.

mediaWaitForConnect – Если установлено в ИСТИНА, то указывает, что получатель сообщения Установить не должен передавать носители информации до передачи сообщения Соедининить.

canOverlapSend – Если установлено в ИСТИНА, то указывает, что отправитель сообщения Установить должен поддерживать совмещенную передачу.

endpointIdentifier – Это идентификатор конечной точки, который был присвоен терминалу в сообщении RCF. Это поле должно присутствовать, когда сообщение Установить передается к гейткиперу, в котором эта конечная точка зарегистрирована, и не должно присутствовать, когда Установить передается к какому-либо другому объекту.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

ConnectionParameters – Позволяет определить параметры, необходимые шлюзам, которые обеспечивают типы с несколькими соединениями и/или агрегирование (например, шлюзу H.323/H.320):

- **scnConnectionType** – Предоставляет шлюзу информацию о типе отдельного соединения, используемого для образования полного соединения по SCN. Конечные точки и гейткиперы будут заполнять это поле, если такая информация доступна им. Если указана опция "Несколько скоростей", то октет скорости переноса информации вIE Возможность переноса также должен указывать "Несколько скоростей", а октет множителя скорости должен

указывать число соединений. Во всех других случаях, если присутствует поле **scnConnectionType**, то оно отменяет любое указание о типе отдельного соединения, содержащееся в Скорости переноса (октет № 4) и Множителе скорости (октет № 4.1) в IE Возможность переноса.

- **numberOfSCNConnections** – Указывает число соединений типа **scnConnectionType**, которые агрегированы вместе для образования соединения по SCN. Это поле, помноженное на полосу пропускания отдельного соединения, указанную в **scnConnectionType**, будет означать полосу пропускания полного соединения по SCN. Конечные точки или гейткиперы должны заполнять это поле, если такая информация доступна им. Заметим, что если **scnConnectionType** установлен в "Неизвестен", то предполагается единица полосы пропускания 64 кбит/с. Если присутствуют как это поле, так и **scnConnectionType**, то общая указанная полоса пропускания должна совпадать с общей полосой пропускания в SCN, указанной Скоростью переноса (октет № 4) и Множителем скорости (октет № 4.1) в IE Возможность переноса.
- **scnConnectionAggregation** – Указывает, как отдельные соединения агрегируются вместе для образования полного соединения в SCN. Конечные точки или гейткиперы должны заполнять это поле, если такая информация доступна им. Когда фактический механизм агрегирования неизвестен, будет использоваться безусловный вариант "Авто". Когда известно, что используется связывание, но неизвестен точный режим связывания, то будет использоваться вариант "Связанная модель".

language – Указывает язык(и), на котором(ых) пользователь предпочел бы получать объявления и тревожные сообщения. Это поле содержит один или несколько маркеров языка, соответствующих RFC 1766.

presentationIndicator – Указывает, что представление **sourceAddress** следует разрешить или ограничить. Если присутствуют как **presentationIndicator**, так и указатель предоставления в IE Соединенный номер, но не совпадают, то должен использоваться указатель предоставления из IE Соединенный номер.

screeningIndicator – Указывает, что **sourceAddress** был выдан конечной точкой или сетью (гейткипером), и указывает, был ли sourceAddress проверен гейткипером. Если присутствуют как **screeningIndicator**, так и указатель проверки в IE Соединенный номер вызывающей стороны, но не совпадают, то должен использоваться указатель проверки из IE Номер вызывающей стороны.

serviceControl – Содержит данные, специфичные для услуги (либо ссылки на них), которые могут использоваться как часть процедуры установления в вызываемой конечной точке (к примеру, изображение или пиктограмма, отображаемые при предупреждении), как описано, например, в Приложении K/H.323.

symmetricOperationRequired – Если присутствует, то указывает, что вызываемая конечная точка должна выбрать одинаковые аудио-возможности для передачи и приема. Этот элемент не должен включаться, когда не включен также элемент **fastStart**.

capacity – Это поле указывает пропускную способность соединения, доступную передающей конечной точке в этот момент времени, предполагая, что это сообщение Соединить представляет реальное соединение. Когда конечная точка передает это поле, она должна включать элемент **currentCallCapacity**.

circuitInfo – Это поле предоставляет информацию о канале или каналах SCN, используемых для этого соединения.

desiredProtocols – Указывает в порядке предпочтения типы протоколов, которые вызывающая конечная точка желает использовать в своем соединении (например, голос или факс). Какой-либо разрешающий объект может использовать это поле для обнаружения конечной точки, которая тоже поддерживает такой протокол, с учетом порядка предпочтения.

neededFeatures – Это поле указывает список общих свойств, которые требуются, по порядку, для успешного выполнения соединения.

desiredFeatures – Это поле указывает список общих свойств, которые предпочтительны для соединения, но не требуются для его успешного выполнения.

supportedFeatures – Это поле указывает список общих свойств, которые отправитель поддерживает и выбрал для объявления.

parallelH245Control – Это поле переносит последовательность туннелированных блоков PDU Набор возможностей терминала H.245 и факультативных блоков PDU Определение ведущего-ведомого. Каждая цепочка октета должна содержать точно один PDU H.245.

additionalSourceAddresses – Это поле переносит последовательность адресов-псевдонимов, которые соответствуют второму и последующим информационным элементам Номер вызывающей стороны в сетях не-H.323. Например, в ЦСИС номера нескольких вызывающих сторон могут присутствовать для поддержки "Варианта с доставкой информационных элементов Два номера вызывающей стороны", определенного в Приложении A/Q.951.

hopCount – Это поле указывает целочисленное значение количества транзитных участков, которое сигнализация о соединении может пройти далее.

ПРИМЕЧАНИЕ. – Если присутствует **destExtraCallInfo**, то CRV для каждого организуемого соединения может выдаваться в **destExtraCRV**. Эти CRV будут использоваться для идентификации ответа на каждое начатое соединение. Эти процедуры остаются для изучения. Если **destExtraCRV** отсутствует, то шлюз должен собрать всю информацию о соединении в один ответ, что приведет к тому, что при неудаче одного соединения на стороне SCN будет считаться неудачным полное соединение.

7.3.11 Подтверждение Установления

Это сообщение может передаваться объектом H.323. Однако оно может передаваться и в обратную сторону, из сети через шлюз. Его обработка после приема является факультативной, но объект, который указал **canOverlapSend** в сообщении Установить, должен поддерживать Подтверждение Установления.

Содержимое и семантика сообщения Подтверждение Установления, принимаемого из сети, определены в таблице 3-16/Q.931 с учетом изменений из таблицы 14.

Таблица 14/H.225.0 – Подтверждение Установления

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Идентификация канала	FFS	н. д.
Отображение	O	2–82
Пользователь-пользователь	M	2–131

Для обратной совместимости с системами H.225.0 ранее версии 4 отправитель этого сообщения не должен включать поле **h4501SupplementaryService** или **h245Control** в поле **h323-message-body** информационного элемента Пользователь-пользователь.

Информационный элемент Пользователь-пользователь содержит **SetupAcknowledge-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **SetupAcknowledge-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

7.3.12 Статус

Сообщение Статус должно использоваться для ответа на неизвестное сообщение сигнализации о соединении или на сообщение Запрос Статуса.

Соответствует таблице 3-17/Q.931 с учетом изменений по таблице 15.

Таблица 15/H.225.0 – Статус

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения (примечание)	M	3
Тип сообщения	M	1
Причина	M	4–32
Состояние соединения	M	3
Отображение	O	2–82
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ. – Это сообщение может переносить глобальный справочный номер соединения, если это сообщение применено ко всем вызовам по соединению, переносящему несколько соединений.		

Для обратной совместимости с системами H.225.0 ранее версии 4 отправитель этого сообщения не должен включать поле **h4501SupplementaryService** или **h245Control** в поле **h323-message-body** информационного элемента Пользователь-пользователь.

Информационный элемент Пользователь-пользователь содержит **Status-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **Status-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

7.3.13 Запрос Статуса

Сообщение Запрос Статуса может использоваться для запроса статуса соединения, как описано в 8.4.2/H.323.

Соответствует таблице 3-18/Q.931 с учетом изменений по таблице 16.

Таблица 16/H.225.0 – Запрос Статуса

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения (примечание)	M	3
Тип сообщения	M	1
Отображение	O	2–82
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ. – Это сообщение может переносить глобальный справочный номер соединения, если это сообщение применено ко всем вызовам по соединению, переносящему несколько соединений.		

Для обратной совместимости с системами H.225.0 ранее версии 4 отправитель этого сообщения не должен включать поле **h4501SupplementaryService** или **h245Control** в поле **h323-message-body** информационного элемента Пользователь-пользователь.

Информационный элемент Пользователь-пользователь содержит **StatusInquiry-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **StatusInquiry-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

7.4 Детали сообщений сигнализации о соединении H.225.0 на базе Q.932

Сообщения, определяемые ниже, взяты из Рекомендаций МСЭ-Т Q.932 и H.450. Дальнейшие детали см. в Рекомендациях МСЭ-Т Q.932 и H.450.

7.4.1 Возможности

Сообщение Возможности должно использоваться для предоставления информации о том, куда следует направить вызов (**FacilityReason** = **routeCallToMC**), или использоваться для указания конечной точке, что входящий вызов должен пройти через гейткипер (**FacilityReason** = **routeCallToGatekeeper**).

Чтобы сигнализировать о перенаправлении вызова, специфичном для процедур H.323, используется информационный элемент Пользователь-пользователь сообщения Возможности. Этот частный случай должен указываться путем кодирования IE Возможности с длиной нуль. В этом случае информационный элемент Возможности должен содержать точно 2 октета. Объект H.323 должен правильно обработать пустой (специфичный для H.323) IE Возможности и должен быть способен опускать остальные IE Возможности, которые он не понимает.

Сообщение Возможности может использоваться для запроса или подтверждения дополнительной услуги согласно Рекомендациям серии H.450.x. По этой причине внутри информационного элемента Пользователь-пользователь сообщения Возможности должны переноситься один или несколько APDU Дополнительная услуга из H.450. Эти APDU Дополнительная услуга H.450 должны кодироваться согласно разделу 8/H.450.1. Информационный элемент Возможности должен быть заполнен длиной нуль. Заметим, что сообщение Возможности из H.225.0 версии 2 или версии 3, которое переносит только блоки APDU Дополнительная услуга H.450, могло бы не содержать Facility-UUIE, а вместо него использовать вариант **h323-message-body**, установленный в **empty**. В этом случае сообщение Возможности может не содержать в себе поле **callIdentifier**. При H.225.0 версии 4 или выше отправитель должен в каждое сообщение Возможности, связанное с соединением, включать Facility-UUIE, переносящее поле **callIdentifier**, и должен устанавливать поле **reason** в значение **transportedInformation**.

Если присутствует IE Возможности, переносящее семантику Рекомендации МСЭ-Т Q.932 и кодированное согласно Рекомендации МСЭ-Т Q.932 и Рекомендации МСЭ-Т Q.95.x, то он должен содержать по меньшей мере 8 октетов, как требуется в таблице 7-2/Q.932. Использование формационных элементов Возможности этого типа остается для изучения.

Сообщение Возможности может использоваться конечной точкой или гейткипером для запроса получателя установить канал H.245 между двумя объектами (**FacilityReason** = **startH245**).

Сообщение Возможности может использоваться конечной точкой или гейткипером для передачи нового набора маркеров (меток) в поле **tokens** и/или **cryptoTokens** этого сообщения Возможности (**FacilityReason** = **newTokens**). Это может быть полезным, например, для приложений, в которых маркеры используются для разрешения выполнять некоторое действие только в пределах ограниченного времени.

Соответствует 7.1.1/Q.932 и 10.8 Стандарта ISO/IEC 11582 с учетом изменений из таблицы 17.

Таблица 17/H.225.0 – Возможности

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения (примечание 1)	M	3
Тип сообщения	M	1
Расширенные возможности	O (примечание 2)	8–*
Возможности	O (примечание 2)	2 или 8–*
Указатель уведомления	O	2–*
Отображение	O	2–82
Номер вызывающей стороны	F	н. д.
Номер вызываемой стороны	F	н. д.
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ 1. – Это сообщение может переносить глобальный справочный номер соединения, если это сообщение применено ко всем вызовам по соединению, переносящему несколько соединений. ПРИМЕЧАНИЕ 2. – Если сообщение Возможности используется для переноса сигнализации о дополнительной услуге Q.95.x, то требуется информационный элемент Возможности или Расширенные возможности. Если сообщение Возможности используется для управления Дополнительной услугой согласно Рекомендациям серии H.450.x, либо если сообщение Возможности используется для перемаршрутизации к функциям МС/гейткипера, то требуется информационный элемент Возможности с нулевой длиной.		

Кодирование информационного элемента Тип сообщения

Информационный элемент Тип сообщения в сообщении Возможности должен кодироваться "0110 0010".

Информационный элемент Пользователь-пользователь содержит Facility-UUIE, определенный в синтаксисе сообщения H.225.0. Компонент Facility-UUIE содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

alternativeAddress – Это конкретный транспортный адрес, к которому вызывающая сторона собирается направить вызов; если он присутствует, то не требуется **alternativeAliasAddress**.

alternativeAliasAddress – Содержит псевдонимы, которые могут быть использованы для перенаправления вызова; если выдается псевдоним, то не требуется **alternativeAddress**.

conferenceID – Уникальный идентификатор конференции; не требуется, если используется поле **conferences**.

reason – Различная информация о сообщении Возможности. Установка **reason** в **featureSetUpdate** указывает, что целью этого сообщения является обновление информации **featureSet**, переданной ранее. Причина **forwardedElements** указывает, что целью сообщения является продвижение элементов другого сообщения в случае, когда это сообщение не может быть передано, что могло быть в случае, когда маршрутизирующий гейткипер получил сообщение Готовность Вызова после того, как он уже передал Готовность Вызова. Причина **transportedInformation** указывает, что целью сообщения является транспортировка информации более высокого уровня, например, в поле **h4501SupplementaryService**; в этом случае **Facility-UUIE** включается только для выдачи **callIdentifier**.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

destExtraCallInfo – Нужен для создания возможных дополнительных вызовов канала, то есть для вызова 2 × 64 кбит/с на стороне SCN. Должен содержать только цепочки цифр номеронабирателя, номера по E.164 или частные номера, но не должен содержать номер исходного канала.

remoteExtensionAddress – Содержит адрес-псевдоним вызываемой конечной точки в случаях, когда эта информация нужна для прохождения через несколько шлюзов.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

conferences – Одна или несколько конференций, которые могут быть присоединены.

h245Address – Это конкретный транспортный адрес, по которому конечная точка или гейткипер, передающие эту возможность, собираются установить сигнализацию H.245 с получателем. Заметим, что это поле может присутствовать в случае, когда промежуточный объект сигнализации переносит поле **h245Address** из сообщения Готовность Вызова. Принимающий объект должен начинать процедуры H.245 только в случае, когда **reason** установлена в **startH245**.

fastStart – Используемый только в процедуре быстрого соединения, **fastStart** поддерживает сигнализацию, необходимую для открытия логического канала. Использование структуры **OpenLogicalChannel** определено в Рекомендации МСЭ-Т H.245, но ее отправитель указывает режимы, в которых он предпочитает принимать и передавать, а также транспортные адреса, от которых он ожидает получить потоки носителей. Это поле присутствует в сообщении Возможности, когда маршрутизирующий гейткипер получил его в сообщении Готовность Вызова от вызываемого пользователя и транслирует эту информацию к вызывающему пользователю. Это поле не должно включаться конечной точкой.

multipleCalls – Если установлено в ИСТИНА, то указывает, что отправитель сообщения способен к сигнализации о нескольких соединениях по одному соединению сигнализации о соединении.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что передатчик сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

fastConnectRefused – Вызываемая конечная точка будет передавать этот элемент в любом сообщении до сообщения Соединить, включая его, при установлении соединения для указания, что она отказывается от процедуры Быстрое соединение. Это поле присутствует в сообщении Возможности, когда маршрутизирующий гейткипер получил его в сообщении Готовность Вызова от вызываемого пользователя и транслирует эту информацию к вызывающему пользователю.

serviceControl – Содержит данные, специфичные для услуги (или ссылки на них), которые могли бы использоваться конечной точкой или шлюзом (к примеру, для отображения меню опций для какого-либо участника соединения), как описано, например, в Приложении K/H.323.

circuitInfo – Это поле предоставляет информацию о канале или каналах SCN, используемых для этого соединения.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

destinationInfo – Содержит **EndpointType**, позволяющий вызывающей стороне определить, включен ли шлюз в соединение. Это поле присутствует в сообщении Возможности, когда маршрутизирующий гейткипер получил его в сообщении Готовность Вызова от вызываемого пользователя и транслирует эту информацию к вызывающему пользователю. Это поле отсутствует в сообщении Возможности H.225.0 ранее версии 4.

h245SecurityMode – Объект H.323, который получает сообщение Установить с набором **h245SecurityCapability**, отвечает соответствующим приемлемым **h245SecurityMode** в сообщении Готовность Вызова, Предупреждение, Прохождение или Соединить. Это поле присутствует в сообщении Возможности, когда маршрутизирующий гейткипер получил его в сообщении Готовность Вызова от вызываемого пользователя и транслирует эту информацию к вызывающему пользователю. Это поле отсутствует в сообщении Возможности H.225.0 ранее версии 4.

7.4.2 Уведомление

Это сообщение может передаваться объектом H.323. Его обработка после получения является факультативной.

Соответствует таблице 3-8/Q.931 с учетом изменений из таблицы 18.

Таблица 18/H.225.0 – Notify

Информационный элемент	Статус в H.225.0 (M/F/O)	Длина в H.225.0
Дискриминатор протокола	M	1
Справочный номер соединения	M	3
Тип сообщения	M	1
Возможность переноса	O (примечание)	5–6
Указатель уведомления	M	3
Отображение	O	2–82
Пользователь-пользователь	M	2–131
ПРИМЕЧАНИЕ. – Включена для указания на изменение возможности переноса.		

Для обратной совместимости с системами H.225.0 ранее версии 4 отправитель этого сообщения не должен включать поле **h4501SupplementaryService** или **h245Control** в поле **h323-message-body** информационного элемента Пользователь-пользователь.

Информационный элемент Пользователь-пользователь содержит **Notify-UUIE**, определенный в синтаксисе сообщения H.225.0. Компонент **Notify-UUIE** содержит:

protocolIdentifier – Указывает поддерживаемую версию H.225.0.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

7.4.3 Другие сообщения

Сообщения управления соединением, которые могут переносить факультативные элементы Возможности, Расширенные возможности или Указатель уведомления, определяются в 8.3.

7.5 Значения таймеров сигнализации о соединении H.225.0

Должны обеспечиваться следующие таймеры Q.931:

- "Таймер установки" T303 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, как долго вызывающая конечная точка должна ждать сообщение Предупреждение, Готовность Вызова, Соединить, Освобождение Завершено или другое сообщение от вызываемой конечной точки после того, как вызывающая конечная точка передала сообщение Установить. Значение этого тайм-аута должно быть не менее 4 секунд. Заметим, что в сетях могут появиться некоторые приложения, которые имеют присущие им более длинные задержки (например, сравните Интернет с местной сетью предприятия, то есть сетью интранет).
- "Таймер создания" T301 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, через какое время вызывающая конечная точка должна прекратить ожидание ответа от вызываемой конечной точки. Этот таймер запускается, когда получено сообщение Предупреждение, и нормально останавливается при получении сообщения Соединить, либо когда вызывающая сторона закончит попытку соединения и передаст сообщение Освобождение Завершено. Значение этого тайм-аута должно быть 180 секунд (3 минуты) или больше.
- "Таймер совмещенной передачи" T302 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, через какое время вызываемая конечная точка должна прекратить ожидание цифр набора номера от вызывающей конечной точки, выполняющей совмещенную передачу. Этот таймер запускается, когда передано сообщение Подтверждение Установления или получено

сообщение Информация, и нормально останавливается, когда получит указание о завершении передачи. Значение этого тайм-аута должно быть 10–15 секунд.

- "Таймер совмещенного приема" T304 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, через какое время вызывающая конечная точка должна прекратить ожидание цифр набора номера от пользователя вызываемой конечной точки, выполняющей совмещенный прием. Этот таймер запускается, когда получено сообщение Подтверждение Установления, перезапускается, когда передано сообщение Информация, и нормально останавливается при получении сообщения Готовность Вызова, Предупреждение или Соединить. Значение этого тайм-аута должно быть не менее 20 секунд.
- "Таймер обработки входящего вызова" T310 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, через какое время вызываемая конечная точка должна прекратить ожидание цифр набора номера от вызывающей конечной точки, выполняющей совмещенную передачу. Этот таймер запускается, когда получено сообщение Готовность Вызова и нормально останавливается при получении сообщения Предупреждение или Соединить, либо когда вызывающая сторона закончит попытку соединения и передаст сообщение Освобождение Завершено. Значение этого тайм-аута должно быть не менее 10 секунд.
- "Таймер статуса" T322 (см. таблицы 9-1/Q.931 и 9-2/Q.931) определяет, через какое время вызываемая конечная точка должна прекратить ожидание сообщения Статус в ответ на сообщение Запрос Статуса, которое она передала. Этот таймер запускается, когда передано сообщение Запрос Статуса и нормально останавливается, когда получено сообщение Статус. Значение этого тайм-аута должно быть не менее 4 секунд.

Заметим, что значения этих тайм-аутов на стороне пакетной сети будут такими же, какие используются в SCN.

Другие таймеры могут обеспечиваться как часть Рекомендаций серии H.450.x о факультативных дополнительных услугах.

7.6 Общие элементы сообщений H.225.0

В этом разделе описываются структуры языка ASN.1, которые используются более чем в одном сообщении протокола Регистрации, допуска и статуса (RAS). Некоторые могут также использоваться в части Пользователь-пользователь сообщений сигнализации о соединении.

requestSeqNum используется в сообщениях для слежения за несколькими запросами, ожидающими передачи. Любое связанное ответное сообщение (об успешности или неудаче) должно иметь соответствующий **requestSeqNum**, передаваемый в нем. Повторно передаваемые сообщения должны иметь один и тот же **requestSeqNum**. Этот **requestSeqNum** увеличивается шагами на 1 при модуле 65536.

protocolIdentifier включается как часть процедур Открытия, регистрации и Установить/Соединить, чтобы разрешать участвующим сторонам определять марку использованных реализаций.

nonStandardParameter – Этот параметр применяется факультативно в последовательностях открытия, регистрации и Установить/Соединить, чтобы разрешить участвующим сторонам определять нестандартные состояния используемых конечных точек. Гейткипер или шлюз не обязан пропускать **nonStandardData**, если он их не поддерживает или понимает, что они могли бы помешать его операциям.

Структура **TransportAddress** предназначена для сбора различных транспортных форматов и содержит некоторую схему, специфичную для транспортировки, в дополнение к возможной местной ссылке на идентификатор TSAP.

Адреса IPv4 и IPv6 должны кодироваться так, чтобы октет старших разрядов адреса был в первом октете соответствующей ЦЕПОЧКИ ОКТЕТОВ, например, адрес IPv4 класса B 130.1.2.97 должен иметь закодированное "130" в первом октете ЦЕПОЧКИ ОКТЕТОВ, затем следует "1" и так далее.

Адрес IPv6 a148:2:3:4:a:b:c:d должен иметь закодированное "a1" в первом октете, "48" – во втором, "00" – в третьем, "02" – в четвертом и так далее.

TransportAddress типа **ipSourceRoute**, в котором последовательность **route** не имеет введенных данных, должна считаться представляющей тот же адрес, что и тип **ipAddress**, который содержит те же значения **ip** и **port**.

Адреса IPX, **node, netnum** и **port** должны кодироваться так, чтобы октет старших разрядов каждого поля был в первом октете соответствующей ЦЕПОЧКИ ОКТЕТОВ.

Заметим, что эта структура не использует Транспортный адрес = язык Рекомендации МСЭ-Т H.323 "Адрес пакетной сети плюс Идентификатор TSAP". Вместо этого используются элементы, общие для всех транспортных доменов.

Структура **EndpointType** переносит информацию об объекте H.323, расположенном на конце звена сигнализации. Объект H.323 будет формировать один или несколько элементов сообщения **gatekeeper**, **gateway**, **mcu** или **terminal**. Если объект H.323 имеет MC, то **mc** будет иметь булево значение ИСТИНА. В разделе 6.3/H.323 описано представление MCU, расположенного совместно с шлюзом; в этом случае устройство H.323 может содержать элементы **gateway** и **mcu** в его определении **EndpointType**. Присутствие компонента **set** указывает, что объект является устройством Простого типа конечной точки (Simple Endpoint Type, SET), определенным, например, в Приложении F/H.323. Битовые позиции в компоненте **set** указывают тип устройства SET; их значения определены в Приложении F/H.323 и в других Рекомендациях, определяющих типы устройств SET. Поле **supportedTunnelledProtocols** дает список расположенных по приоритету поддерживаемых туннелированных протоколов (высший приоритет расположен первым).

Структура **TunnelledProtocol** указывает туннелированный протокол сигнализации, описанный, например, в Приложениях M.1 и M.2/H.323. Поле **tunnelledProtocolObjectID** представляет собой **OBJECT IDENTIFIER**, определяющий туннелируемый протокол. Поле **tunnelledProtocolAlternateID** дает формат переменного идентификатора. Поле **subIdentifier** позволяет определить конкретную версию стандартного протокола.

Структура **TunnelledProtocolAlternateIdentifier** предоставляет формат идентификатора в виде цепочки для туннелированного протокола. Поле **protocolType** дает общий тип протокола, например, ISUP. Поле **protocolVariant** дает конкретный вариант этого стандарта, например, ANSI.

Туннелированные протоколы, которые определены согласно настоящей Рекомендации, приведены в таблицах VI.1 и VI.2. Заметим, что туннелирование не ограничивается протоколами, перечисленными в этих таблицах.

Структура **GatewayInfo** содержит элемент **protocol**, который позволяет шлюзу указывать поддерживаемые им протоколы.

Структура **SupportedProtocols** указывает выбранные протоколы, с которыми объект H.323 может взаимодействовать. Например, выбор альтернативы **h310** указывает, что объект обеспечивает взаимодействие с H.310.

В каждой поддерживаемой структуре возможных протоколов (**H310Caps**, **H320Caps** и т. д.) элемент **dataRatesSupported** указывает скорости передачи данных, которые поддерживает каждый протокол устройства. Элемент **supportedPrefixes** указывает префиксы, связанные с поддерживаемым протоколом, а в некоторых случаях также со скоростями передачи данных.

Структура **McuInfo** содержит элемент **protocol**, который позволяет MCU указывать поддерживаемые им протоколы.

Структура **CapacityReportingCapability** указывает способность конечной точки сообщать информацию о пропускной способности соединения.

Структура **CapacityReportingSpecification** указывает информацию о пропускной способности, которую конечная точка запрашивает для сообщений. Поле **callStart** указывает информацию о запрашиваемой пропускной способности в начале соединения (то есть для ARQ или Установить). Поле **callEnd** указывает информацию о запрашиваемой пропускной способности в конце соединения (то есть для DRQ или Освобождение Завершено). Пустая последовательность **when** указывает на запрос, в котором оконечная точка не сообщает информацию о пропускной способности.

Структура **CallCapacityInfo** позволяет конечной точке указывать свою пропускную способность при приеме соединений для каждого типа соединения, который поддерживается этой конечной точкой. Следовательно, она представляет текущее состояние покоя конечной точки. Например, в голосовом шлюзе **CallCapacityInfo** будет представлять число свободных каналов.

Структура **CallCapacity** позволяет конечной точке указывать свою максимальную пропускную способность для каждого типа соединений и свою текущую доступную пропускную способность для каждого типа соединений, который поддерживается этой конечной точкой.

Структура **CallsAvailable** предоставляет некоторый поднабор общей пропускной способности конечной точки для соединений. Поле **group** может быть тем же, который сообщен в **CircuitIdentifier**.

Структура **DataRate** дает информацию о скорости протокола в шлюзе. Поле **channelRate** является основной канальной скоростью в сотнях бит/с. Поле **channelMultiplier** указывает число каналов с channelRate. Например, если шлюз поддерживает 3 канала B, то **channelMultiplier** = 3, а **channelRate** = 640 при канале 64 кбит/с.

Структура **VendorIdentifier** позволяет продавцу идентифицировать продукт. Элемент **vendor** позволяет идентификацию с помощью кода страны, некоторого расширения и кода производителя. Поля **productId** и **versionId** являются текстовыми цепочками, которые могут содержать информацию о продукте. Поле **enterpriseNumber** указывает изготовителя и присваивается Полномочным органом по присвоенным номерам Интернета (Internet Assigned Numbers Authority, IANA).

Структура **H221NonStandard** позволяет определять некоторое нестандартное поле. Элемент **t35CountryCode** должен определять страну согласно Приложению A/T.35. Элемент **t35Extension** должен содержать расширение кода страны, назначаемое в этой стране, если **t35CountryCode** не является двоичным "1111 1111", а если так, то это поле должно содержать код страны, находящийся в Приложении B/T.35. Элемент **manufacturerCode** должен быть присвоенным в стране кодом, определяющим изготовителя аппаратуры.

Структура **AliasAddress** предназначена для сбора различных внешних форматов адреса, которые указывают конкретное транспортное местоположение в пакетной сети. При регистрации гейткипером адреса, состоящего из цифр номеронабирателя, конечная точка должна использовать поле **dialledDigits** и применять только цифры 0–9. При регистрации гейткипером адреса по E.164 конечная точка должна использовать поле **e164Number** и применять только цифры 0–9. При регистрации и другом представлении префикса конечная точка должна использовать поле **dialledDigits** и применять только цифры 0–9 и знаки "#" и "*". Поле **mobileUIM** является идентификационным модулем для систем, совместимых со 2-м поколением и 3-м поколением радиосетей, и дает возможность взаимодействия с Сетями сухопутной подвижной связи общего пользования, которое описано, например, в Приложении E/H.246.

Структура **AddressPattern** позволяет указывать трафаретный **AliasAddress** или диапазон для **PartyNumber**. Поле **wildcard** представляет возможное трафаретное расширение структуры **AliasAddress**. Для цифр номеронабирателя или номеров E.164 это расширение возможно в конце номера. Для адресов e-mail такое расширение возможно вначале. Например, если трафарет равен "+1 303", то шаблон (Address Pattern) мог бы представлять любой номер в коде зоны Денвер. Поле **range** структуры **AddressPattern** представляет диапазон адресов, включая указанные начало и конец диапазона.

Механизмы, которые конечная точка использует для определения типа адреса, оставлены на усмотрение реализатора. Представление различных типов номеров для сообщений приведены в таблице 19. Заметим, что если конечная точка не знает тип или назначение какого-либо адреса, то она будет представлять его как Private Unknown при кодировании в сообщениях сигнализации о соединении H.225.0 и как **dialledDigits** в **AliasAddress**, когда кодирует в сообщениях RAS.

Таблица 19/H.225.0 – Отображение представлений типа номера

Тип номера	Представление Q.931	Представление информационного элемента H.225.0	Представление UUIE H.225.0
Неизвестен (по умолчанию и версия 1 режима взаимодействия)	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	Частный план нумерации, Тип номера = Неизвестен ("000")	**dialledDigits AliasAddress** (примечание 2)
Частный, неизвестен	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	**dialledDigits AliasAddress** (примечание 2)
Частный, Региональный номер уровня 2	Частный план нумерации, Тип номера = Региональный номер уровня 2 ("001")	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	**privateNumber** для **PartyNumber AliasAddress**, **TypeOfNumber** = **level2RegionalNumber**
Частный, Региональный номер уровня 1	Частный план нумерации, Тип номера = Региональный номер уровня 1 ("010")	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	**privateNumber** для **PartyNumber AliasAddress**, **TypeOfNumber** = **level1RegionalNumber**
Частный, Номер, специфичный для Частной сети с интеграцией служб (Private Integrated Service Network, PISN)	Частный план нумерации, Тип номера = Специфичный для PISN номер ("011")	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	**privateNumber** для **PartyNumber AliasAddress**, **TypeOfNumber** = **pISNSpecificNumber**
Частный, Региональный номер уровня 0 (Локальный)	Частный план нумерации, Тип номера = Региональный номер уровня 0 ("100")	Частный план нумерации, Тип номера = Неизвестен ("000") (примечание 1)	**privateNumber** для **PartyNumber AliasAddress**, **TypeOfNumber** = **localNumber**
Номер общего пользования E.164, неизвестный	План нумерации ЦСИС/телефонной сети, Тип номера = Неизвестен ("000")	План нумерации ЦСИС/телефонной сети, Тип номера = Неизвестен ("000")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **Unknown**
Номер общего пользования E.164, Международный номер	План нумерации ЦСИС/телефонной сети, Тип номера = Международный номер ("001")	План нумерации ЦСИС/телефонной сети, Тип номера = Международный номер ("001")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **internationalNumber**
Номер общего пользования E.164, Национальный номер	План нумерации ЦСИС/телефонной сети, Тип номера = Национальный номер ("010")	План нумерации ЦСИС/телефонной сети, Тип номера = Национальный номер ("010")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **nationalNumber**

Таблица 19/H.225.0 – Отображение представлений типа номера

Тип номера	Представление Q.931	Представление информационного элемента H.225.0	Представление UUIE H.225.0
Номер общего пользования E.164, Специфичный для сети номер	План нумерации ЦСИС/телефонной сети, Тип номера = Специфичный для сети номер ("011")	План нумерации ЦСИС/телефонной сети, Тип номера = Специфичный для сети номер ("011")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **networkSpecific Number**
Номер общего пользования E.164, Номер абонента	План нумерации ЦСИС/телефонной сети, Тип номера = Номер абонента ("100")	План нумерации ЦСИС/телефонной сети, Тип номера = Номер абонента ("100")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **subscriberNumber**
Номер общего пользования E.164, Сокращенный номер	План нумерации ЦСИС/телефонной сети, Тип номера = Сокращенный номер ("110")	План нумерации ЦСИС/телефонной сети, Тип номера = Сокращенный номер ("110")	**e164Number** для **PartyNumber AliasAddress**, **TypeOfNumber** = **abbreviatedNumber**
ПРИМЕЧАНИЕ 1. – Когда Идентификация плана нумерации = Частный, цифры частного номера кодируются в **privateNumber** для **PartyNumber**, который содержит тип номера. Поле Тип номера в информационном элементе должен игнорироваться на приеме и кодироваться согласно этой таблице на передаче. ПРИМЕЧАНИЕ 2. – Элемент **privateTypeOfNumber** = **Unknown PartyNumber AliasAddress** должен обрабатываться так же, как **dialledDigits AliasAddress**.			

Структура **MobileUIM** представляет идентифицирующий модуль для систем, совместимых со 2-м поколением и 3-м поколением радиосетей. Доступны следующие варианты:

- **ansi-41-uim** – Для радиосетей, определенных американскими стандартами.
- **gsm-uim** – Для радиосетей, определенных европейскими стандартами.

Структура **ANSI-41-UIM** указывает идентифицирующий модуль для систем, совместимых с американскими стандартами для радиосетей. Доступны следующие варианты:

- **imsi** – Для номеров International Mobile Station Identification.
- **min** – Для Mobile Identification Numbers.
- **mdn** – Для Mobile Directory Numbers.
- **msisdn** – Для номеров Mobile Station ISDN.
- **esn** – Для Electronic Serial Numbers.
- **mscid** – Для номеров Mobile Switching Center и номеров Market Identification или System Identification.
- **sid** – Для номеров System Identification.
- **mid** – Для номеров Market Identification.
- **systemMyTypeCode** – Для номеров идентификации продавцов.
- **systemAccessType** – Для типа доступа к системе.
- **qualificationInformationCode** – Для кода уточняющей информации.
- **sesn** – Для SIM Electronic Serial Numbers.
- **soc** – Для System Operator Codes.

Структура **GSM-UIM** указывает идентифицирующий модуль для систем, совместимых с европейскими стандартами для радиосетей. Доступны следующие варианты:

- **imsi** – Для International Mobile Station Identification.
- **tmsi** – Для Temporary Mobile Station Identification.
- **msisdn** – Для номеров Mobile Station ISDN.
- **imei** – Для номеров International Mobile Equipment Identification.
- **hplmn** – Для номеров Home Public Land Mobile Network.
- **vplmn** – Для номеров Visiting Public Land Mobile Network.

Структура **ExtendedAliasAddress** обеспечивает средства для связи общей информации с адресами-псевдонимами. Элемент **presentationIndicator** указывает, что наличие **address** будет разрешено или ограничено. Элемент **screeningIndicator** указывает, что **address** был выдан конечной точкой или сетью, и указывает, что он был или не был проверен сетью.

Структура **Endpoint** используется для указания вспомогательной избыточной или альтернативной информации о конечной точке:

- **nonStandardData** – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).
- **aliasAddress** – Это список адресов-псевдонимов, по которым другие конечные точки могут идентифицировать эту конечную точку.
- **callSignalAddress** – Это транспортный адрес сигнализации о соединении для этой конечной точки.
- **rasAddress** – Это транспортный адрес для регистрации и статуса этой конечной точки.
- **endpointType** – Это указывает тип конечной точки.
- **tokens** – Маркеры (метки), связанные с этой конечной точкой (то есть с конечной точкой, описанной в структуре **Endpoint**).
- **cryptoTokens** – **CryptoTokens**, связанные с этой конечной точкой (то есть с конечной точкой, описанной в структуре **Endpoint**).
- **priority** – Используется в случаях, когда присутствует ПОСЛЕДОВАТЕЛЬНОСТЬ из нескольких **Endpoints**. Конечные точки с меньшими номерами приоритета имеют преимущества перед конечными точками с более высокими номерами приоритета. Конечные точки без номеров приоритета эквивалентны тем, у которых приоритет 0 (высший приоритет).
- **remoteExtensionAddress** – Содержит адрес-псевдоним конечной точки в случае, когда эта информация необходима для прохождения нескольких шлюзов.
- **destExtraCallInfo** – Содержит внешние адреса для нескольких соединений.
- **alternateTransportAddresses** – Указывает поддержку транспортных протоколов, отличных от TCP.

Структура **alternateTransportAddresses** переносит адреса сигнализации о соединении для транспортов, отличных от TCP.

Структура **UseSpecifiedTransport** определяет выбранный транспортный протокол сигнализации. Значение **tcp** указывает протокол TCP, значение **annexE** указывает протокол, определенный в Приложении E/H.323, а значение **sctp** указывает на использование протокола SCTP (Stream Control Transmission Protocol).

Структура **AlternateGK** используется для указания списка альтернативных или резервных гейткиперов:

- **rasAddress** – Транспортный адрес, используемый для сигнализации RAS.
- **gatekeeperIdentifier** – Факультативно включается для определения резервного или альтернативного гейткипера. Если он представлен, то он должен включаться в будущие сообщения RAS, передаваемые к резервному гейткиперу.
- **needToRegister** – Устанавливается в ИСТИНА для указания, что конечная точка должна зарегистрироваться у другого гейткипера, прежде чем передавать другие запросы RAS.

- **priority** – Указывает приоритет резервного или альтернативного гейткипера. Меньший номер означает более высокий приоритет.

Структура **AltGKInfo** используется для предоставления информации об альтернативных гейткиперах:

- **alternateGatekeeper** – Последовательность альтернативных гейткиперов, расположенных по приоритету.
- **altGKisPermanent** – Устанавливается в ИСТИНА для указания, что все будущие сигналы RAS следует перенаправлять к гейткиперу, записанному в поле **alternateGatekeeper**; устанавливается в ЛОЖЬ в случае, когда следует перенаправлять только сообщение, которое привело к его отклонению. Этот флаг должен быть установлен в ИСТИНА, если флаг **needToRegister** установлен в ИСТИНА в поле **alternateGatekeeper**.

Структура **QseriesOptions** дает для гейткипера или других конечных точек информацию, относящуюся к поддержке терминалом факультативных протоколов серии Q. Она используется в сообщениях ARQ, Установить и GRQ. Использование опций серии Q еще не определено и подлежит дальнейшему изучению.

Структуры **GloballyUniqueID** и **ConferenceIdentifier** означают уникальные в глобальном масштабе идентификаторы (**GloballyUniqueID**), использование которых описано в Рекомендации МСЭ-Т H.323. Кодируется **GloballyUniqueID** с октетом нуль, который кодируется первым. Формируется **GloballyUniqueID** согласно таблице 20.

Таблица 20/H.225.0 – Формирование GloballyUniqueID

Поле	Тип данных	№ октета	Примечание
time_low	Целое число без знака, 32 бита	0–3	Нижнее поле метки времени
time_mid	Целое число без знака, 16 битов	4–5	Среднее поле метки времени
time_hi_and_version	Целое число без знака, 16 битов	6–7	Верхнее поле метки времени, объединенное с номером версии
clock_seq_hi_and_reserved	Целое число без знака, 8 битов	8	Верхнее поле синхрокомбинации, объединенное с вариантом
clock_seq_low	Целое число без знака, 8 битов	9	Нижнее поле синхрокомбинации
node	Целое число без знака, 48 битов	10–15	Уникальный для территории идентификатор узла

Структура **GloballyUniqueID** содержит запись 16-и октетов и не должна содержать заполнения между полями. Общий размер равен 128 октетам.

Чтобы минимизировать ошибочные присвоения битов внутри октетов, определение записи **GloballyUniqueID** выполняется только в виде полей, содержащих целое число октетов. Номер версии объединяется с меткой времени (*time_high*), а поле варианта объединяется с синхрокомбинацией (*clock_seq_high*).

Метка времени – это 60-битовое значение, представляющее Скоординированное всемирное время (UTC) в виде результата подсчета 100-наносекундных интервалов начиная с 00:00:00.00 15 октября 1582 г. (даты григорианской реформы с переходом на христианский календарь).

Номер версии размещается в 4-х битах старших разрядов поля *time_hi_and_version* и устанавливается в 1 (двоичное "0001").

Поле варианта определяет компоновку **GloballyUniqueID**. Структура DCE **GloballyUniqueID** фиксирована для всех различных версий. Другие варианты **GloballyUniqueID** могут не обеспечить взаимодействие с DCE **GloballyUniqueID**. Взаимодействием разных **GloballyUniqueID** считается применимость операций, таких как преобразование цепочек, сравнение и лексическое упорядочение, для всех разных систем. Поле *variant* содержит переменное число битов старших разрядов поля *clock_seq_hi_and_reserved* (см. таблицу 21).

Таблица 21/H.225.0 – Содержимое поля варианта DCE

msb1	msb2	msb3	Описание
0	–	–	Зарезервировано для обратной совместимости с системой управления сетью (NCS)
1	0	–	Вариант DCE
1	1	0	Зарезервировано для уникального в глобальном масштабе идентификатора (GUID) корпорации Microsoft
1	1	1	Зарезервировано для будущего определения

Синхрокомбинация нужна для обнаружения потерь монотонности синхронизации. Синхрокомбинация кодируется в 6-и битах младших разрядов поля *clock_seq_hi_and_reserved* и в поле *clock_seq_low*.

Поле *node* содержит адрес по IEEE, обычно адрес хоста. Для систем с несколькими узлами согласно IEEE 802 может быть использован любой доступный адрес узла. Младший адресный октет (октет с номером 10) содержит бит "глобальный/локальный" и бит "однопунктовый/многопунктовый", являясь первым октетом адреса, передаваемого по пакетной сети IEEE 802.3.

Значение синхрокомбинации (*clock sequence*) будет изменяться, когда:

- генератор **GloballyUniqueID** обнаружит, что местное значение UTC отстает; это может быть при нормальном функционировании Службы времени DCE.
- генератор **GloballyUniqueID** потерял свое положение в последнем использованном значении UTC, что указывает на возможное отставание времени; это обычно случается при перезагрузке.

Когда узел работает, генератор **GloballyUniqueID** обычно записывает последнее значение UTC, использованное для создания **GloballyUniqueID**. Каждый раз, когда создается новый **GloballyUniqueID**, текущее *UTC* сравнивается с записанным значением; если текущее значение меньше (случай немонотонного такта) или если записанное значение было потеряно, то синхрокомбинация увеличивается на 1 по модулю 16 384, что позволяет избежать выработки дублированных **GloballyUniqueID**.

Запуск *clock sequence* производится со случайного номера, чтобы минимизировать корреляцию между системами.

GloballyUniqueID генерируется согласно следующему алгоритму:

1) Определить значения для метки времени, основанной на UTC, и для синхрокомбинации, которые будут использоваться в **GloballyUniqueID**.
2) Установить поле *time_low* равным 32-м битам младших разрядов (битам с номерами от 0 до 31 включительно) из метки времени, в том же порядке разрядов.
3) Установить поле *time_mid* равным битам с номерами от 32 до 47 включительно из метки времени, в том же порядке разрядов.
4) Установить 12 битов младших разрядов (битов с номерами от 0 до 11 включительно) из поля *time_hi_and_version* равными битам с номерами от 48 до 59 включительно из метки времени, в том же порядке разрядов.
5) Установить 4 бита старших разрядов (биты с номерами от 12 до 15 включительно) из поля *time_hi_and_version* равными 4-битовому номеру версии, который соответствует создаваемой версии **GloballyUniqueID** согласно таблице 21.
6) Установить поле *clock_seq_low* равным 8-и битам младших разрядов (битам с номерами от 0 до 7 включительно) из *clock sequence*, в том же порядке разрядов.
7) Установить 6 битов младших разрядов (биты с номерами от 0 до 5 включительно) поля *clock_seq_hi_and_reserved* равными 6-и битам старших разрядов (битам с номерами от 8 до 13 включительно) из *clock sequence*, в том же порядке разрядов.
8) Установить 2 бита старших разрядов (биты с номерами 6 и 7) поля *clock_seq_hi_and_reserved* в 0 и 1 соответственно.

9) Установить поле *node* равным 48-битовому адресу по IEEE в том же порядке разрядов, как в адресе.

Если в системе желательно генерировать **GloballyUniqueID**, но не имеется сетевой платы, совместимой с IEEE 802, или другого источника адресов IEEE 802, то следует использовать альтернативный метод для генерации заменяющего значения для адреса. Идеальным решением является получение 47-битового случайного номера с криптографическим качеством и использование его в 47-и битах старших разрядов идентификатора узла, установив в 1 бит младшего разряда в первом октете этого идентификатора узла. Этот бит является битом "однопунктовый/многопунктовый", который не будет никогда так устанавливаться в адресах по IEEE 802, полученных из сетевой платы; поэтому не может возникнуть конфликта между двумя **GloballyUniqueID**, которые генерированы машинами с сетевой картой и без нее.

Если система не имеет элементарной функции для генерирования случайных номеров с криптографическим качеством, то в большинстве систем обычно имеется достаточно большое количество доступных источников случайности, с помощью которых можно генерировать что-либо. Такие источники специфичны для системы, но часто охватывают определенный процент используемой памяти, размер главной памяти в байтах, объем свободной главной памяти в байтах, размер файла перелистывания или перестановки в байтах, свободные байты файла перелистывания или перестановки, общий размер пользовательского виртуального адресного пространства в байтах, общее доступное пользовательское адресное пространство в байтах, размер загрузочного дисковода в байтах, свободную область дисковой памяти в загрузочном дисководе в байтах, текущее время, прошедшее время от начальной загрузки системы, отдельные размеры файлов в различных каталогах системы и т. п.

Для использования в тексте, который читается человеком, цепочка **GloballyUniqueID** представляется в виде последовательности полей, некоторые из которых разделяются одиночным тире.

Каждое поле считается целым числом, значение которого печатается в виде цепочки шестнадцатеричных цифр с заполнением нулями и с цифрой старшего разряда на первом месте. Шестнадцатеричные значения от "a" до "f" включительно выводятся как строчные знаки, а на входе – нечувствительны к регистру. Эта последовательность является такой же как составной тип **GloballyUniqueID**.

Формальное определение представления цепочки **GloballyUniqueID** дается ниже на языке "расширенная Форма Бэкуса-Наура" (Backus-Naur Form, BNF):

```
UUID                    = <time_low> <hyphen> <time_mid> <hyphen>
                          <time_high_and_version> <hyphen>
                          <clock_seq_and_reserved>
                          <clock_seq_low> <hyphen> <node>
time_low                = <hexOctet> <hexOctet> <hexOctet> <hexOctet>
time_mid                = <hexOctet> <hexOctet>
time_high_and_version   = <hexOctet> <hexOctet>
clock_seq_and_reserved  = <hexOctet>
clock_seq_low           = <hexOctet>
node                    = <hexOctet><hexOctet><hexOctet>
                          <hexOctet><hexOctet><hexOctet>
hexOctet                = <hexDigit> <hexDigit>p
hexDigit                = <digit> | <a> | <b> | <c> | <d> | <e> | <f>
digit                   = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                          "8" | "9"
hyphen                  = "-"
a                       = "a" | "A"
b                       = "b" | "B"
c                       = "c" | "C"
d                       = "d" | "D"
e                       = "e" | "E"
f                       = "f" | "F"
```

Примером представления цепочки **GloballyUniqueID** является:

f81d4fae-7dec-11d0-a765-00a0c91e6bf6

Элемент **timeToLive** – это число секунд, когда регистрация должна считаться действительной.

Структура **H248PackagesDescriptor** – это цепочка октетов, которая будет содержать **PackagesDescriptor** из H.248, закодированный согласно PER ASN.1.

Структура **H248SignalsDescriptor** – это цепочка октетов, которая будет содержать **SignalsDescriptor** из H.248, закодированный согласно PER ASN.1.

Структура **FeatureDescriptor** представляет собой элемент **GenericData**, который используется для общей идентификации свойства.

Структура **CircuitInfo**: предоставляет информацию о канале или каналах SCN, используемых для данного соединения. Поле **sourceCircuitID** дает информацию о канале источника, когда соединение начинается в SCN, и может использоваться входным шлюзом для выдачи идентификатора канала источника к гейткиперу. Поле **destinationCircuitID** дает информацию о канале получателя, когда соединение заканчивается в SCN, и может использоваться гейткипером для выбора канала получателя в выходном шлюзе.

Структура **CircuitIdentifier** обозначает средство для целей сообщения из шлюза или выбора гейткипером. Структура **CircuitIdentifier** поддерживает различные интерфейсы.

Структура **CicInfo** обозначает несущие каналы системы сигнализации № 7 (SS7). Поле **cic** – это код идентификатора канала, определенный в Рекомендации МСЭ-Т Q.763, у которого биты младших разрядов размещаются в первом октете, а биты старших разрядов – в последнем октете. Поле **pointCode** содержит код пункта, определенный в Рекомендации МСЭ-Т Q.763. Первый октет в **pointCode** указывает сеть (код индикатора сети), а остальные октеты указывают значение кода пункта SS7. Поля **cic** и **pointCode** переменны по длине, что позволяет применять национальные варианты.

Структура **GroupID** указывает физическую или логическую группу (**group**) и члена (**member**) (или **набор членов**) в этой группе. Например, **group** может определять физический интерфейс, а **member** – конкретный DS0 на этом интерфейсе. Если поле **member** пропущено, то шлюз может выбрать какое-либо доступное средство в указанной **group**.

Структура **CarrierInfo** содержит информацию о Выборе переносчика. Поле **carrierIdentificationCode** указывает переносчика (подобно коду идентификации переносчика в сообщении IAM подсистемы ISUP), выбранного абонентом или определенного приложениями маршрутизации в виде двоичной цепочки цифр. Поле **carrierName** является другим средством указания переносчика в виде цепочки знаков из кода ASCII.

Поле **carrier** – это код идентификатора/выбора переносчика для маршрутизации соединения, выбранный приложениями маршрутизации или предпочитаемый абонентом.

Структура **ServiceControlDescriptor** содержит данные, специфичные для услуги, или ссылки на них, которые предназначены для представления пользователю или для сообщения другим средствам управления услугой, как описано, например, в Приложении K/H.323. Возможны следующие опции (факультативные варианты):

- **url** – Этот вариант содержит протокол или ресурс, указанный в унифицированном указателе ресурсов (Uniform Resource Locator, URL).
- **signal** – Этот вариант содержит **SignalsDescriptor**, определенный в Рекомендации МСЭ-Т H.248, в двоичном формате. Факультативные элементы **streamID** и **notifyCompletion** должны опускаться из последовательности **Signal** в **SignalsDescriptor**.
- **nonStandard** – Этот вариант содержит информацию, не определенную в этой Рекомендации (например, данные о собственности).
- **callCreditServiceControl** – Этот вариант содержит информацию, относящуюся к управлению длительностью соединения и к уведомлению пользователя об остатке на счете.

Структура **ServiceControlSession** содержит описание сеанса управления службой, определенное, например, в Приложении K/H.323. Она содержит следующие поля:

- **sessionId** – Целое число, обозначающее данный сеанс и уникальное для клиента. Заметим, что такие идентификаторы, полученные через различные тракты сигнализации (например, RAS и сигнализация о соединении), являются ортогональными (независимыми) и могут перекрываться.
- **contents** – Это структура **ServiceControl** с соответствующим содержимым или механизмом связи.

- **reason** – Указывает, что этот сеанс является новым (**open**) или является модификацией существующего сеанса (**refresh**), либо указывает, что сеанс заканчивается провайдером (**close**), а имеющиеся ресурсы, такие как графический пользовательский интерфейс (Graphical User Interface, GUI) и т. п., следует закрыть.

Структура **RasUsageInfoTypes** перечисляет типы информации об использовании, которые могут посылаться конечной точкой к гейткиперу. Конечная точка применяет эту структуру для указания своих возможностей по сбору и выдаче информации об использовании, а гейткипер применяет эту структуру для запроса конкретных типов информации об использовании. Поле **nonStandardUsageTypes** позволяет продавцу указать типы информации об использовании, связанные с собственностью. Поля **startTime** и **endTime** указывают моменты времени, когда соединение началось и закончилось соответственно. Параметр **terminationCause** указывает причину, по которой соединение закончилось.

Структура **RasUsageSpecification** является шаблоном, который позволяет гейткиперу запросить конкретные типы информации об использовании в конкретный момент времени соединения. Поле **when** указывает момент или моменты в соединении, в которые конечная точка запрашивается для сообщения такой информации; поле **start** указывает начало соединения, поле **end** указывает конец соединения, а **inIrr** указывает незатребованные сообщения IRR. Поле **callStartingPoint** определяет момент или моменты в соединении, которые должны рассматриваться как начало соединения для целей выдачи информации об использовании; значение **connect** указывает на передачу или прием сообщения Соединить (Connect), а значение **alerting** указывает на передачу или прием сообщения Предупреждение (Alerting). Поле **required** указывает типы информации об использовании, которые конечная точка должна сообщить. Структура **RasUsageSpecification**, в которой ничего не выбрано в полях **when** или **required**, указывает запрос запрещения сообщать информацию об использовании.

Структура **RasUsageInformation** является совокупностью данных об использовании, относящихся к конкретному соединению. Поле **nonStandardUsageFields** позволяет продавцу перечислить информацию об использовании, связанную с собственностью. Поле **alertingTime** указывает время, когда сообщение Предупреждение было передано или принято. Поле **connectTime** указывает время, когда сообщение Соединить было передано или принято. Поле **endTime** указывает время, когда сообщение Освобождение Завершено было передано или принято.

Структура **CallTerminationCause** указывает причину окончания соединения. Поле **releaseCompleteReason** указывает **reason**, которая была приведена в сообщении Освобождение Завершено. Поле **releaseCompleteCauseIE** содержит IE Причина из сообщения Освобождение Завершено.

Структура **BandwidthDetails** определяет дополнительную информацию об использовании полосы пропускания, которая отсутствует в структуре **BandWidth**. Поле **sender** устанавливается в ИСТИНА, если сообщение передано отправителем потока, или в ЛОЖЬ, если передано получателем. Поле **multicast** устанавливается в ИСТИНА, если поток является многопунктовым, или в ЛОЖЬ при других случаях. Поле **bandwidth** указывает полосу пропускания, использованную для потока, в сотнях бит/с. Поле **rtcpAddresses** указывает адреса RTCP, использованные для потока носителей.

Структура **CallCreditCapability** указывает определенные возможности конечной точки, относящиеся к счетам за соединения. "По умолчанию" подразумевается, что конечная точка может не иметь эту факультативную возможность. Если какое-либо поле в этой структуре отсутствует, то это означает, что статус возможности, представляемой этим полем, не изменился с последнего сообщения о нем. Поле **canDisplayAmountString** указывает, может ли конечная точка отображать текстовую цепочку, содержащую количество денег на счету пользователя. Поле **canEnforceDurationLimit** указывает, имеет ли конечная точка возможность разъединять соединение при истечении лимита длительности, указанного гейткипером.

Структура **CallCreditServiceControl** позволяет гейткиперу выдавать к конечной точке определенные управляющие команды и информацию, относящиеся к счетам. Эта структура выдает следующие поля:

- **amountString** – Это поле указывает количество денег на счету пользователя, например, "$10.00". Эта цепочка должна содержать соответствующий символ валюты. Заметим, что стандартные сокращения для типов валюты, такие как "USD" для долларов США, определены в ISO 4217. Поле **amountString** должно кодироваться в Basic из ISO/IEC 10646-1 (уникод).

- **billingMode** – Это поле указывает режим составления счета для этого соединения. Режим **debit** означает, что соединение приведет к расходам из количества денег, имеющегося на счете пользователя. Режим **credit** означает, что соединение приведет к расходам, которые подлежат оплате в дальнейшем. Конечная точка может использовать эту информацию, например, для определения типа объявления при игре или отображении.
- **callDurationLimit** – Это поле указывает оставшийся интервал времени, разрешенный для конкретного соединения.
- **enforceCallDurationLimit** – Это поле указывает, имеет ли конечная точка требование разъединить соединение после истечения интервала времени, указанного в **callDurationLimit**. Если это поле отсутствует, то конечная точка должна понимать это как указание, что директива не изменилась по сравнению с ее предыдущим состоянием.
- **callStartingPoint** – Это поле указывает момент времени в соединении, который требуется считать началом, если проведение подсчета длительности соединения обеспечивается конечной точкой.

Структура **GenericData** содержит поле **id** для идентификации данных и поле **parameters** для переноса фактических параметров.

Структура **GenericIdentifier** дает различные способы идентификации объекта.

Структура **EnumeratedParameter** дает некоторый общий параметр. Она содержит поле **id** для идентификации параметра и поле **content** для переноса любых связанных данных.

Структура **Content** поддерживает несколько различных типов данных, включая **raw**, **text**, **unicode**, **bool**, **number8**, **number16**, **number32**, **id**, **alias**, **transport**, **compound** и **nested**. Это позволяет гибко определять общий параметр. Выбор **raw** допускает параметр или набор параметров, для которых фактическая структура данных определена еще где-нибудь; например, она может содержать символы ASN.1 с кодом PER, или содержать данные в форме "тип-длина-значение", или может быть вложенным (инкапсулированным) сообщением другого протокола сигнализации.

Структура **FeatureSet** позволяет объекту указывать информацию об общих свойствах. Объект указывает набор свойств, которые он требует для успешного завершения соединения, с помощью поля **neededFeatures**, набор свойств, которые он предпочитает, но не требует, с помощью поля **desiredFeatures** и набор свойств, которые он поддерживает, в поле **supportedFeatures**. Булево поле **replacementFeatureSet** устанавливается в ИСТИНА для указания, что этот набор свойств заменяет ранее переданный набор свойств, или в ЛОЖЬ в остальных случаях.

Структура **TransportChannelInfo** дает информацию о канале транспортировки носителей. Поле **sendAddress** является транспортным адресом отправителя, а поле **recvAddress** является транспортным адресом получателя.

Структура **RTPSession** дает описание сеанса RTP. Она имеет следующие поля:

- **rtpAddress** – Это поле дает передающий и приемный адреса для потока RTP.
- **rtcpAddress** – Это поле дает передающий и приемный адреса для сообщений RTCP.
- **cname** – Это поле дает CNAME, определенное в разделе 6 и в Приложении A.
- **ssrc** – Это поле используется для идентификации источника потока RTP, как описано в разделе 6 и в Приложении A.
- **sessionId** – Это поле содержит идентификатор данного сеанса RTP, описанной в Рекомендации МСЭ-Т H.245.
- **associatedSessionIds** – Это поле содержит идентификаторы связанных сеансов RTP, описанные в Рекомендации МСЭ-Т H.245.
- **multicast** – Это поле указывает, является ли данный сеанс многопунктовым.
- **bandwidth** – Это поле указывает полосу пропускания, используемую для данного потока, в сотнях бит/с.

7.7 Требуемая поддержка сообщений RAS

В таблице 22 показаны сообщения RAS, которые поддерживаются конечными точками разных типов.

Таблица 22/H.225.0 – Статусы сообщений RAS

Сообщение RAS	Конечная точка (передача)	Конечная точка (прием)	Гейткипер (передача)	Гейткипер (прием)
GRQ	O			M
GCF		O	M	
GRJ		O	M	
RRQ	M			M
RCF		M	M	
RRJ		M	M	
URQ	O	M	O	M
UCF	M	O	M	O
URJ	O	O	M	O
ARQ	M			M
ACF		M	M	
ARJ		M	M	
BRQ	M	M	O	M
BCF	M (примечание 1)	M	M	O
BRJ	M	M	M	O
IRQ		M	M	
IRR	M			M
IACK		O	CM	
INAK		O	CM	
DRQ	M	M	O	M
DCF	M	M	M	M
DRJ	M (примечание 2)	M	M	M
LRQ	O		O	M
LCF		O	M	O
LRJ		O	M	O
NSM	O	O	O	O
XRS	M	M	M	M
RIP	CM	M	CM	M
RAI	O			M
RAC		O	M	
SCI	O	O	O	O
SCR	O	O	O	O

M: Обязательное, O: Факультативное, F: Запрещенное, CM: Условно обязательное, пусто: "Не применимо".

ПРИМЕЧАНИЕ 1. – Если гейткипер передает BRQ с запросом меньшей скорости, то конечная точка должна ответить BCF, когда эта меньшая скорость поддержана, или BRJ в противном случае. Если гейткипер передает BRQ с запросом более высокой скорости, то конечная точка может ответить сообщением BCF или BRJ.

ПРИМЕЧАНИЕ 2. – Терминал не должен передавать DRJ в ответ на действительное DRQ от его гейткипера.

7.8 Сообщения обнаружения от терминала и шлюза

Сообщение GRQ запрашивает, чтобы любой гейткипер, получивший его, ответил сообщением GCF, предоставляющим ему разрешение регистрировать. Сообщение GRJ является отказом на этот запрос, означающим, что запрашивающей конечной точке следует поискать другой гейткипер.

7.8.1 Запрос Гейткипера (GatekeeperRequest, GRQ)

Заметим, что одно GRQ передается одной логической конечной точкой; следовательно, MCU или шлюз могут передать много GRQ.

Сообщение GRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

protocolIdentifier – Определяет марку по H.225.0 для передающей конечной точки.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

rasAddress – Это транспортный адрес, который эта конечная точка использует для сообщений регистрации и статуса. Гейткипер должен посылать сообщения RAS по этому адресу, а не по адресу, от которого сообщение было передано, кроме случаев, когда **rasAddress** не может быть декодирован.

endpointType – Указывает тип (типы) конечной точки, которая регистрируется (бит MC не должен быть установлен самой).

gatekeeperIdentifier – Цепочка для идентификации гейткипера, от которого терминал хотел бы получить разрешение на регистрацию. Отсутствие или нулевая цепочка поля **gatekeeperIdentifier** указывает, что терминал заинтересован в любом доступном гейткипере.

callServices – Дает информацию о поддержке факультативных протоколов серии Q к гейткиперу и к вызываемому терминалу.

endpointAlias – Список адресов-псевдонимов, по которым другие терминалы могут идентифицировать этот терминал.

alternateEndpoints – Последовательность альтернатив конечной точки, расположенных по приоритету, для **rasAddress**, **endpointType** или **endpointAlias**.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

authenticationCapability – Указывает механизмы аутентификации, поддерживаемые этой конечной точкой.

algorithmOIDs – Указывает полный набор алгоритмов шифрования, поддерживаемых этой конечной точкой.

integrity – Указывает для получателя, какой механизм проверки целостности должен быть применен к сообщениям RAS.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

supportsAltGK – Указывает, поддерживает ли эта конечная точка механизм запасного гейткипера.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.8.2 Подтверждение Гейткипера (GatekeeperConfirm, GCF)

Сообщение GCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в GRQ.

protocolIdentifier – Определяет марку принимающего гейткипера.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

gatekeeperIdentifier – Цепочка для идентификации гейткипера, который передает это GCF.

rasAddress – Это транспортный адрес, который гейткипер использует для сообщений регистрации и статуса.

alternateGatekeeper – Последовательность альтернатив, расположенных по приоритету, для **gatekeeperIdentifier** и **rasAddress**.

authenticationMode – Это указывает механизм аутентификации, который будет использован. Гейткипер должен выбрать **authenticationMode** из **authenticationCapability**, которые предоставила конечная точка в GRQ.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

algorithmOID – Указывает алгоритм шифрования, необходимый для этого гейткипера.

integrity – Указывает для получателя, какой механизм проверки целостности должен быть применен к сообщениям RAS.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.8.3 Отказ Гейткипера (GatekeeperReject, GRJ)

Сообщение GRJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в GRQ.

protocolIdentifier – Определяет марку отклоняющего гейткипера.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

gatekeeperIdentifier – Цепочка для идентификации гейткипера, который передает это GRJ.

rejectReason – Коды о том, почему GRQ отклонен этим гейткипером. Причина в **genericDataReason** указывает, что запрос отклонен из-за некоторого общего элемента или свойства; в этом случае дополнительная информация может быть приведена в поле **genericData**.

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.9 Сообщения регистрации от терминала и шлюза

Сообщение RRQ является запросом от терминала к гейткиперу о регистрации. Если гейткипер ответит RCF, то терминал должен использовать этот ответивший гейткипер для будущих соединений. Если гейткипер ответит сообщением RRJ, то терминал должен поискать другой гейткипер для регистрации.

7.9.1 Запрос Регистрации (RegistrationRequest, RRQ)

Сообщение RRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

protocolIdentifier – Определяет марку по H.225.0 для передающей оконечной точки.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

discoveryComplete – Установить в ИСТИНА, если эта отвечающая конечная точка перед этим сообщением применила процедуру обнаружения гейткипера; установить в ЛОЖЬ, если выполняется только регистрация. Заметим, что регистрация может устареть, тогда конечная точка будет переходить в состояние неисправности в сообщении RRQ или ARQ с кодом причины **discoveryRequired** или **notRegistered** соответственно. Это указывает, что конечной точке следует выполнить процедуру обнаружения (динамическую или статическую) перед выдачей RRQ с **discoveryComplete**, установленным в ИСТИНА.

callSignalAddress – Это транспортный адрес сигнализации о соединении для этой конечной точки. Если поддерживается несколько транспортов, то должны регистрироваться все сразу.

rasAddress – Это транспортный адрес регистрации и статуса для этой конечной точки. Гейткипер должен посылать сообщения RAS по этому адресу, а не по адресу, от которого сообщение было передано, кроме случаев, когда **rasAddress** не может быть декодирован.

terminalType – Указывает тип (типы) конечной точки, который (которые) регистрируются; заметим, что бит **mc** не должен быть установлен сам по себе; должен быть также установлен один из битов **terminal**, **mcu**, **gateway** или **gatekeeper**. Если представлена информация **vendor**, то эта информация должна быть идентична представленной в **endpointVendor**. Если **terminalType** будет **gateway** или **mcu**, то факультативное значение **supportedPrefixes** будет списком префиксных адресов, с помощью которых другие конечные точки могут идентифицировать протоколы SCN и скорости передачи данных, поддерживаемые этим объектом. Это поле может использоваться в дополнение к полям **terminalAlias** и **terminalAliasPattern**, либо как их альтернатива. Все префиксы, поддерживаемые конечной точкой, должны включаться в каждое RRQ, кроме случаев указания опции **additiveRegistration**, когда поддерживаемые префиксы в RRQ должны добавляться в список текущих зарегистрированных префиксов для этой конечной точки. При добавочном RRQ поддерживаемые префиксы, уже зарегистрированные для этой конечной точки, должны считаться еще зарегистрированными. Заметим, что префиксы не являются частью **PartyNumber** (E.164 или других).

Чтобы зарегистрировать какой-либо **PartyNumber** (или диапазон, или шаблон для них), конечная точка должна применить поля **terminalAlias** и **terminalAliasPattern**, как описывается ниже.

terminalAlias – Это факультативное значение является списком адресов-псевдонимов, по которым другие терминалы могут идентифицировать этот терминал. Это поле может использоваться в дополнение к полям **terminalAliasPattern** и **supportedPrefixes**, либо как их альтернатива. Если **terminalAlias** установлен в нуль, то некоторый адрес **terminalAlias** может быть присвоен гейткипером и включен в RCF. Если для конечной точки доступен **email-ID**, то его следует зарегистрировать. Заметим, что несколько адресов-псевдонимов могут ссылаться на одни и те же транспортные адреса. Все псевдонимы конечной точки, которые желательно зарегистрировать, должны быть включены в этот список, кроме случая указания опции **additiveRegistration**, когда псевдонимы конечной точки в RRQ должны быть добавлены к списку текущих зарегистрированных псевдонимов для этой конечной точки.

gatekeeperIdentifier – Цепочка для идентификации гейткипера, у которого желает зарегистрироваться этот терминал.

endpointVendor – Информация о продавце конечной точки.

alternateEndpoints – Последовательность альтернатив конечной точки, расположенных по приоритету, для **callSignalAddress**, **rasAddress**, **terminalType** или **terminalAlias**.

timeToLive – Интервал действительности регистрации, в секундах. После этого времени гейткипер может считать эту регистрацию устаревшей.

tokens – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

keepAlive – Если установлено в ИСТИНА, то указывает, что конечная точка передала это RRQ как "подтверждение активности". Конечная точка может передать упрощенное RRQ, содержащее только **rasAddress**, **keepAlive**, **endpointIdentifier**, **gatekeeperIdentifier**, **tokens**, и **timeToLive**. Гейткипер, получив RRQ с полем **keepAlive,** установленным в ИСТИНА, будет игнорировать все поля, кроме **endpointIdentifier**, **gatekeeperIdentifier**, **tokens** и **timeToLive**. Гейткипер должен использовать **rasAddress** из упрощенного RRQ только в качестве получателя для RRJ, когда эта конечная точка не зарегистрирована.

endpointIdentifier – Это **endpointIdentifier,** выданный гейткипером во время первоначального RCF.

willSupplyUUIEs – Если установлено в ИСТИНА, то указывает, что конечная точка будет поддерживать информацию сообщения сигнализации о соединении H.225.0 в сообщениях IRR, если будет запрос от гейткипера.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что отправитель этого сообщения способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

alternateTransportAddresses – Это поле переносит адреса сигнализации о соединении для транспортных протоколов, отличающихся от TCP. Включение какого-либо адреса означает поддержку соответствующего транспорта.

additiveRegistration – Если присутствует, то это поле указывает, что это сообщение RRQ является "добавочным", то есть конечная точка передала это RRQ для добавления информации к существующей регистрации. Конечная точка может передать добавочное RRQ, содержащее только **callSignalAddress**, **rasAddress**, **terminalType**, **terminalAlias**, **terminalAliasPattern**, **alternateEndpoints**, **endpointIdentifier**, **gatekeeperIdentifier** и **tokens**. Гейткипер, получив RRQ с присутствующим полем **additiveRegistration,** должен игнорировать все поля, кроме этих. Гейткипер должен использовать **rasAddress** из добавочного RRQ только в качестве получателя для последующего RRJ, если конечная

тока не зарегистрирована, либо если **terminalAlias** и/или **terminalAliasPattern** противоречат политике регистрации гейткипера.

terminalAliasPattern – Это факультативное значение является списком образцов адресов, Определяющим псевдонимы и адреса, по которым другие конечные точки могут идентифицировать эту конечную точку. Это поле может использоваться в дополнение к полям **terminalAlias** и **supportedPrefixes,** либо как их альтернатива. Все псевдонимы и адреса конечной точки должны включаться в каждое RRQ, кроме случая опции **additiveRegistration,** установленной в ИСТИНА, когда псевдонимы и адреса конечной точки из RRQ должны добавляться в список текущих псевдонимов, зарегистрированных для конечной точки.

supportsAltGK – Указывает, поддерживает ли эта конечная точка механизм запасного гейткипера.

usageReportingCapability – Это поле может быть включено конечной точкой для объявления своей способности собирать и выдавать различные типы информации об использовании.

multipleCalls – Если это поле установлено в ИСТИНА, то указывает, что отправитель этого сообщения способен сигнализировать о нескольких соединениях по одному соединению сигнализации о соединении.

supportedH248Packages – Это поле указывает список комплектов (package) H.248, которые поддерживаются этой конечной точкой.

callCreditCapability – Это поле описывает определенные возможности этой конечной точки, относящиеся к составлению счетов.

capacityReportingCapability – Это поле описывает способность конечной точки выдавать информацию о пропускной способности соединения.

capacity – Это поле указывает максимальную и текущую пропускную способность соединения для этой конечной точки.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

restart – Это поле, если установлено, указывает, что это RRQ – первое, переданное конечной точкой после ее перезагрузки или после ненормального события, которое вызвало потерю ее соединений. Это позволяет гейткиперу выполнить некоторую очистку или другие необходимые функции.

supportsACFSequences – Это поле, если установлено, указывает, что конечная точка способна принимать и обрабатывать последовательность сообщений ACF в ответ на одиночное сообщение ARQ.

7.9.2 Подтверждение Регистрации (RegistrationConfirm, RCF)

Сообщение RCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в RRQ.

protocolIdentifier – Определяет марку принимающего гейткипера.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

callSignalAddress – Это массив транспортных адресов для сообщений сигнализации о соединении H.225.0; один для каждого транспорта, на который будет отвечать гейткипер. В этом адресе содержится идентификатор TSAP.

terminalAlias – Это факультативное значение является списком адресов-псевдонимов, по которым другие терминалы могут идентифицировать этот терминал. Это поле может использоваться в дополнение к полям **terminalAliasPattern** и **supportedPrefixes,** либо как их альтернатива. Оно указывает адреса-псевдонимы, которые были приняты из предложенных в связанном сообщении RRQ. Если в RRQ ничего не было предложено, то этот список дает псевдонимы, присвоенные гейткипером. Если это поле отсутствует, а адреса-псевдонимы были предложены в RRQ, то гейткипер принял все эти предложенные адреса-псевдонимы. Если это поле присутствует и указывает поднабор адресов-псевдонимов, предложенных в RRQ, то гейткипер принял только эти адреса.

gatekeeperIdentifier – Цепочка для идентификации гейткипера, который принял регистрацию терминала.

endpointIdentifier – Цепочка идентификации терминала, присвоенная гейткипером; должна повторяться в последующих сообщениях RAS.

alternateGatekeeper – Последовательность альтернатив, расположенных по приоритету, для **gatekeeperIdentifer** и **rasAddress**.

timeToLive – Интервал действительности регистрации, в секундах. После этого времени гейткипер может считать эту регистрацию устаревшей.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

willRespondToIRR – Устанавливается в ИСТИНА, если гейткипер будет передавать сообщение IACK или INAK в ответ на нетребующееся сообщение IRR с его полем **needsResponse,** установленным в ИСТИНА.

preGrantedARQ – Указывает события, для которых гейткипер имеет заранее предоставленный допуск. Это дает более короткое время установления соединения в среде, где допуск гарантируется средствами, отличающимися от обмена ARQ/ACF. Заметим, что если даже эти поля установлены в ИСТИНА, конечная точка может еще передать ARQ к гейткиперу при таких причинах, как преобразование адреса, либо когда конечная точка не поддерживает этот измененный режим сигнализации. Если последовательность **preGrantedARQ** отсутствует, то во всех случаях используется сигнализация ARQ. Полями являются:

- **makeCall** – Если флаг **makeCall** установлен в ИСТИНА, то гейткипер имеет заранее предоставленный допуск для конечной точки начинать соединения, не передавая сигнала ARQ. Если флаг **makeCall** установлен в ЛОЖЬ, то конечная точка должна всегда передавать ARQ, чтобы получить допуск на выполнение соединения.
- **useGKCallSignalAddressToMakeCall** – Если флаги **makeCall** и **useGKCallSignalAddressToMakeCall** оба установлены в ИСТИНА, а конечная точка не передает ARQ к гейткиперу для выполнения соединения, то эта конечная точка должна передать всю сигнализацию о соединении H.225 к каналу сигнализации о соединении гейткипера.
- **answerCall** – Если флаг **answerCall** установлен в ИСТИНА, то гейткипер имеет заранее предоставленный допуск для конечной точки отвечать на соединение, не передавая сначала ARQ. Если флаг **answerCall** установлен в ЛОЖЬ, то конечная точка должна всегда передавать ARQ, чтобы получить допуск для ответа на соединение.
- **useGKCallSignalAddressToAnswer** – Если флаги **answerCall** и **useGKCallSignalAddressToAnswer** оба установлены в ИСТИНА, а конечная точка не передает ARQ к гейткиперу для ответа на соединение, тогда эта конечная точка должна контролировать поступления всей сигнализации о соединении H.225.0 от гейткипера. Если конечная точка была проинструктирована использовать гейткипер при ответе, но она не знает, пришел ли входящий вызов от гейткипера (что может затрагивать просмотр транспортного адреса), то эта конечная точка должна выдать ARQ независимо от состояния флага **useGKCallSignalAddressToAnswer**.
- **irrFrequencyInCall** – Это указывает частость, в секундах, передаваемых к гейткиперу сообщений IRR, когда конечная точка находится в состоянии одного или нескольких соединений. Когда это поле отсутствует, гейткипер не желает незатребованных сообщений IRR. Когда конечная точка передает эти сообщения IRR, значение справочного номера соединения должно быть сделано уникальным для терминала, каким он мог бы генерироваться в Запросе допуска. Однако это не является "нормальным" CRV и не может повторно использоваться при дальнейшей связи (DRQ, IRQ или BRQ). Идентификатор соединения должен быть таким же, какой использован в сообщениях канала сигнализации о соединении для соответствующего соединения.

- **totalBandwidthRestriction** – Это поле ограничивает общее использование полосы пропускания для конечной точки, имеющей соединения. Когда оно отсутствует, нет постоянного ограничения на полосу пропускания.
- **alternateTransportAddresses** – Это поле переносит адреса сигнализации о соединении для транспортных протоколов, отличающихся от TCP. Включение какого-либо адреса означает поддержку соответствующего транспорта.
- **useSpecifiedTransport** – Это поле позволяет гейткиперу инструктировать конечную точку о том, какой транспортный протокол сигнализации следует использовать для выполнения соединений. Если это поле включено, а указанным транспортным протоколом является не **tcp**, то в это сообщение должно также включаться **alternateTransportAddresses**.

maintainConnection – Если это поле установлено в ИСТИНА, то указывает, что гейткипер (в случае маршрутизации с гейткипером) способен поддерживать соединение сигнализации, когда по этому соединению нет текущей сигнализации о соединениях.

serviceControl – Содержит данные, специфичные для службы, или адресную информацию, которые конечная точка может использовать при связи с сетью для управления услугой, не относящейся к соединению, как описано, например, в Приложении K/H.323.

supportsAdditiveRegistration – Это поле, если присутствует, указывает, что гейткипер поддерживает добавочные возможности регистрации. Если отсутствует, то гейткипер не поддерживает добавочную регистрацию.

terminalAliasPattern – Это факультативное значение является списком образцов адресов, определяющим псевдонимы и адреса, по которым другие конечные точки могут идентифицировать эту конечную точку. Это поле может использоваться в дополнение к полям **terminalAlias** и **supportedPrefixes**, либо как их альтернатива. Оно определяет псевдонимы и адреса, которые были приняты из предложенных в соответствующем сообщении RRQ. Если в RRQ ничего не было предложено, то этот список дает псевдонимы и адреса, присвоенные гейткипером. Если это поле отсутствует, а образцы адресов были предложены в RRQ, то гейткипер принимает все эти предложенные образцы. Если это поле присутствует и указывает поднабор образцов адресов, предложенных в RRQ, то гейткипер принимает только эти образцы.

supportedPrefixes – Это факультативное значение является списком префиксов, по которым другие конечные точки могут идентифицировать эту конечную точку. Это поле может использоваться в дополнение к полям **terminalAlias** и **terminalAliasPattern,** либо как их альтернатива. Оно определяет адресные префиксы, которые были приняты из предложенных в соответствующем сообщении RRQ. Если в RRQ ничего не было предложено, то этот список дает префиксы, присвоенные гейткипером. Если это поле отсутствует, а адресные префиксы были предложены в RRQ, то гейткипер принимает все эти предложенные префиксы. Если это поле присутствует и указывает поднабор адресных префиксов, предложенных в RRQ, то гейткипер принимает только эти префиксы.

usageSpec – Это поле может быть включено гейткипером, чтобы запросить конечную точку собирать и выдавать указанную информацию об использовании соединения в определенные моменты времени.

featureServerAlias – Это поле зарезервировано для будущего использования в МСЭ-Т для протокола по базе внешнего воздействия.

capacityReportingSpec – Это поле указывает тип информации о пропускной способности соединения, которую конечная точка должна выдать.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.9.3 Отклонение Регистрации (RegistrationReject, RRJ)

Сообщение RRJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в RRQ.

protocolIdentifier – Определяет марку отклоняющего гейткипера.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

rejectReason – Причина отклонения регистрации. Это поле может содержать значение **invalidTerminalAliases,** которое содержит список псевдонимов, адресов и поддерживаемых префиксов, которые были признаны недействительными в соответствующем сообщении RRQ. В любом случае отклоняются все псевдонимы, адреса и поддерживаемые префиксы из соответствующего RRQ, вместе с указанными в поле **invalidTerminalAliases**. Причина **genericDataReason** указывает, что запрос отклонен из-за общего элемента или свойства; в этом случае дополнительная информация может приводиться в поле **genericData**.

gatekeeperIdentifier – Цепочка для идентификации гейткипера, который отклонил регистрацию терминала.

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.10 Сообщения отмены регистрации от терминала/гейткипера

7.10.1 Запрос отмены регистрации (UnregistrationRequest, URQ)

Сообщение URQ запрашивает, чтобы связь между терминалом и гейткипером была разрушена. Заметим, что отмена регистрации (лишение регистрации) является двунаправленной, то есть гейткипер может запросить терминал рассматривать себя незарегистрированным, и терминал может информировать гейткипер, что он аннулирует предыдущую регистрацию.

Сообщение URQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

callSignalAddress – Это один или несколько транспортных адресов сигнализации о соединении для этой конечной точки, регистрация которых должна быть отменена.

endpointAlias – Это факультативное значение является списком адресов-псевдонимов, по которым другие терминалы могут идентифицировать этот терминал. Это поле может использоваться в дополнение к полям **endpointAliasPattern** и **supportedPrefixes,** либо как их альтернатива. Если это поле, поле **endpointAliasPattern** и поле **supportedPrefixes** отсутствуют, то регистрация всех псевдонимов отменяется в некотором одном сообщении. Требуется значение **dialledDigits,** если оно присвоено. Отменяется регистрация только значений, перечисленных здесь; это позволяет, к примеру, отменить регистрацию **h323-ID,** а значение **dialledDigits** оставить зарегистрированным.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

endpointIdentifier – Подтверждение идентификатора; не передается гейткипером.

alternateEndpoints – Последовательность альтернатив конечной точки, расположенных по приоритету, для **callSignalAddress** или **endpointAlias**.

gatekeeperIdentifier – Некоторый **gatekeeperIdentifier,** который конечная точка получила от гейткипера в списке **alternateGatekeeper** в RCF, когда она регистрировалась, или в предыдущем сообщении URJ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это после должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

reason – Используется, когда гейткипер передает URQ для извещения, поему этот гейткипер считает регистрацию конечной точки отмененной. Значение **reason** и **maintenance** указывает, что гейткипер или конечная точка освобождены для технического обслуживания.

endpointAliasPattern – Это факультативное значение является списком образцов адресов, определяющим псевдонимы и адреса, по которым другие конечные точки могут идентифицировать эту конечную точку. Это поле может использоваться в дополнение к полям **endpointAlias** и **supportedPrefixes**. Если это поле, поле **endpointAlias** и поле **supportedPrefixes** отсутствуют, то регистрация всех псевдонимов и адресов отменяется в некотором одном сообщении. В остальных случаях отменяется регистрация только значений, перечисленных здесь.

supportedPrefixes – Это факультативное значение является списком префиксов, по которым другие конечные точки могут идентифицировать эту конечную току. Это поле может использоваться в дополнение к полям **terminalAlias** и **terminalAliasPattern,** либо как их альтернатива. Если это поле, поле **endpointAlias** и поле **endpointAliasPattern** отсутствуют, то регистрация всех псевдонимов и адресов отменяется в некотором одном сообщении. В остальных случаях отменяется регистрация только значений, перечисленных здесь.

alternateGatekeeper – Последовательность альтернатив, расположенных по периметру, для **gatekeeperIdentifier** и **rasAddress**.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.10.2 Подтверждение отмены Регистрации (UnregistrationConfirm, UCF)

Сообщение UCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в URQ.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.10.3 Отклонение отмены Регистрации (UnregistrationReject, URJ)

Сообщение URJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в URQ.

rejectReason – Причина отклонения отмены регистрации.

nonStandardDat – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.11 Сообщения допуска терминала гейткипером

Сообщение ARQ запрашивает, чтобы доступ к пакетной сети для конечной точки был разрешен гейткипером, который удовлетворяет запрос сообщением ACF, либо отказывает ему сообщением ARJ.

7.11.1 Запрос Допуска (AdmissionRequest, ARQ)

Сообщение ARQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

callType – Используя это значение, гейткипер может попытаться определить "реальное" использование полосы пропускания. Безусловным ("по умолчанию") значением является **pointToPoint** для всех соединений. Следует признать, что тип соединения может динамически измениться за время существования соединения и что окончательный тип соединения может быть неизвестен при передаче ARQ.

callModel – Если установлено в **direct**, то конечная точка запрашивает модель прямого соединения "терминал-терминал". Если установлено в **gatekeeperRouted**, то конечная точка запрашивает модель с промежуточным гейткипером. Гейткиперу не требуется исполнять этот запрос.

endpointIdentifier – Это идентификатор конечной точки, который был присвоен терминалу в сообщении RCF.

destinationInfo – Последовательность адресов-псевдонимов для получателя, таких как **dialledDigits**, **PartyNumber** (**e164Number** или **privateNumber**) либо **h323-ID**. Когда ARQ передается в ответ на соединение, **destinationInfo** указывает получателя того соединения (отвечающую конечную точку). Если гейткипером зарегистрирован по меньше мере один псевдоним и нет двух псевдонимов в ARQ, которые зарегистрированы для разных людей, то гейткипер должен воспринимать ARQ как ссылку на зарегистрированный идентификатор. В случае конфликта псевдонимов запрос допуска должен быть отклонен с причиной **AliasesInconsistent**. Если гейткипер не обеспечивает этой проверки, то он должен рассматривать первый зарегистрированный адрес в качестве получателя.

destCallSignalAddress – Транспортный адрес, используемый получателем для сигнализации о соединении.

destExtraCallInfo – Содержит внешние адреса для нескольких соединений.

srcInfo – Последовательность адресов-псевдонимов для конечных точек-источников, например, **dialledDigits**, **PartyNumber** (**e164Number** или **privateNumber**) либо **h323-ID**. При передаче ARQ в ответ на соединение **srcInfo** указывает инициатора соединения.

srcCallSignalAddress – Транспортный адрес, используемый источником для сигнализации о соединении.

bandWidth – Двунаправленная полоса пропускания, запрошенная для соединения, в сотнях бит/с. Например, сигнализацией о соединении со скоростью 128 кбит/с будет запрос на 256 кбит/с. Значение относится только к скорости передачи битов аудио и видео, без учета заголовка и служебных данных.

callReferenceValue – CRV из сообщений сигнализации о соединении H.225.0 для этого соединения; имеет только местное значение. Это поле используется гейткипером для связи ARQ с конкретным соединением.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

callServices – Дает информацию о поддержке гейткипером и вызываемой конечной точкой факультативных протоколов серии Q.

conferenceID – Уникальный идентификатор конференции.

activeMC – Если установлено в ИСТИНА, то вызывающая сторона имеет активный MC; в противном случае – ЛОЖЬ.

answerCall – Используется для изведения гейткипера о том, что соединение является входящим.

canMapAlias – Если установлено в ИСТИНА, то указывает, что когда полученное в ответ ACF будет содержать поля **destinationInfo**, **destExtraCallInfo** и/или **remoteExtensionAddress**, конечная точка должна скопировать эту информацию соответственно в поля **destinationAddress**, **destExtraCallInfo** и **remoteExtensionAddress** сообщения Установить, либо в IE Номер вызываемой стороны, когда это подходит. Если конечной точкой является шлюз, используемый для выхода из сети H.323, то этот шлюз преобразует информацию о получателе в соответствующий формат сигнализации, применяемый за пределами сети H.323 (например, в двухтональную многочастотную сигнализацию DTMF). Если гейткипер желает заменить адресную информацию из ARQ, а **canMapAlias** установлено в ЛОЖЬ, то этот гейткипер может отклонить ARQ. В системах, соответствующих H.225.0 версии 4 и выше, это поле должно устанавливаться в ИСТИНА.

callIdentifier – Уникальный в глобальном масштабе идентификатор соединения, который установлен инициирующей конечной точкой и который может использоваться для связи сигнализации RAS с измеренной сигнализацией Q.931, использованной в этой Рекомендации.

srcAlternatives – Последовательность расположенных по приоритету альтернатив конечной точки-источника для **srcInfo**, **srcCallSignalAddress**, или **rasAddress**.

destAlternatives – Последовательность расположенных по приоритету альтернатив конечной точки-получателя для **destinationInfo** или **destCallSignalAddress**.

gatekeeperIdentifier – Некоторый **gatekeeperIdentifier,** который конечная точка получила от гейткипера в списке **alternateGatekeeper** в RCF, когда она регистрировалась, или в предыдущем сообщении ARJ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщения, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

transportQOS – Конечная точка может использовать это поле для указания своей способности резервировать транспортные ресурсы. Структура **transportQOS** содержит следующее:

- **endpointControlled** – Конечная точка будет применять свой собственный механизм резервирования.
- **gatekeeperControlled** – Гейткипер будет выполнять резервирование ресурса для конечной точки.
- **noControl** – Резервирование ресурса не требуется.

willSupplyUUIEs – Это поле, когда установлено в ИСТИНА, указывает, что конечная точка будет выдавать информацию сообщений сигнализации о соединении H.225.0 в сообщениях IRR по запросу гейткипера.

callLinkage – Содержимое этого поля обычно контролируется услугой сцепления соединений. Процедуры и семантику этого поля см. в разделе 10/H.323.

gatewayDataRate – Запрошенная скорость передачи данных для стороны SCN соединения, проходящего через шлюз. Эта скорость передачи данных, если присутствует, должна равняться скорости передачи данных, указанной в IE Возможность переноса в сообщении Установить. Гейткипер может использовать это поле при выборе шлюза для образования соединения.

capacity – Это поле указывает доступную для передающей конечной точки пропускную способность соединения в данный момент времени, при этом предполагается, что гейткипер подтверждает ARQ путем передачи ACF. Конечная точка, передавая это поле, должна включать элемент **currentCallCapacity**.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

desiredProtocols – Указывает типы протоколов в порядке предпочтения, которые инициирующая конечная точка желает для своего соединения (например, голос или факс). Решающий объект может использовать это поле для определения местонахождения конечной точи, которая тоже поддерживает определенный протокол, с учетом порядка предпочтений.

desiredTunnelledProtocol – Это поле указывает протокол, который требуется туннелировать.

featureSet – Это поле указыает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

canMapSrcAlias – Это поле, если установлено в ИСТИНА, указывает, что когда полученное в ответ ACF будет содержать **modifiedSrcInfo**, конечная точка должна скопировать эту информацию в поле **sourceInfo** сообщения Установить и/или в IE Номер вызывающей стороны, когда это подходит. Если гейткипер желает заменить адресную информацию из ARQ, а **canMapSrcAlias** установлено в ЛОЖЬ, то этот гейткипер может отклонить ARQ.

ПРИМЕЧАНИЕ. – Как **destinationInfo,** так и **destCallSignalAddress** являются факультативными, но хотя бы одно из них должно присутствовать, кроме случаев ответа конечной точки на вызов. Общего правила предпочтительного выбора нет, так как он может зависеть от стороны соединения, но адрес следует выдавать, если он доступен. Следует учитывать, что наилучший результат будет получаться при учете природы используемого транспортного протокола.

7.11.2 Подтверждение Допуска (AdmissionConfirm, ACF)

Сообщение ACF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в ARQ.

bandWidth – Разрешенная максимальная полоса пропускания для этого соединения; может быть меньше запрошенной.

callModel – Сообщает терминалу, что сигнализация о соединении, передаваемая по **destCallSignalAddress,** идет к гейткиперу или к терминалу. Значение **gatekeeperRouted** указывает, что сигнализация о соединении прошла через гейткипер, а **direct** указывает, что используется режим соединения "конечная точка – конечная точка".

destCallSignalAddress – Транспортный адрес, к которому передается сигнализация о соединении H.225.0, но может быть адресом конечной точки или гейткипера в зависимости от используемой модели соединения.

irrFrequency – Частость, в секундах, с которой конечная точка должна передавать сообщения IRR к гейткиперу во время соединения, в том числе во время удержания. Если это поле отсутствует, то конечная точка не передает сообщения IRR во время активного соединения, но ожидается, что гейткипер будет опрашивать конечную точку.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

destinationInfo – Адрес первоначального канала, используемого при вызове через шлюз.

destExtraCallInfo – Необходимо для выполнения вызовов по возможным дополнительным каналам, то есть для вызова 2 × 64 кбит/с на стороне SCN. Должно содержать только адреса **dialledDigits** или **PartyNumber** и не должно содержать номер первоначального канала.

destinationType – Определяет тип конечной точки-получателя.

remoteExtensionAddress – Содержит адрес-псевдоним вызываемой конечной точки в случаях, когда эта информация требуется для прохождения нескольких шлюзов.

alternateEndpoints – Последовательность расположенных по приоритету альтернатив конечной точки для **destCallSignalAddress** или **destinationInfo**.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

transportQOS – Гейткипер может указать конечной точке, кто несет ответственность за резервирование ресурса. Если гейткипер получил **transportQOS** в ARQ, то он включает **transportQOS** (возможно, измененное согласно реализации гейткипера) в ACF.

willRespondToIRR – Устанавливается в ИСТИНА, если гейткипер будет посылать сообщение IACK или INAK в ответ на незапрошенное сообщение IRR, когда поле **needsResponse** в IRR установлено в ИСТИНА.

uuiesRequested – Гейткипер может запросить конечную точку уведомлять этот гейткипер о сообщениях сигнализации о соединении H.225.0, которые конечная точка передает или принимает, если эта конечная точка указала эту способность в ARQ путем установки **willSupplyUUIEs** в ИСТИНА. Поле **uuiesRequested** указывает набор сообщений сигнализации о соединении H.225.0, о которых конечная точка должна уведомлять гейткипер.

language – Указывает язык(и), на котором(ых) пользователь предпочел бы получать объявления и тревожные сообщения. Это поле содержит один или несколько маркеров языка, соответствующих RFC 1766.

alternateTransportAddresses – Это поле переносит адреса сигнализации о соединении для транспортных протоколов, отличающихся от TCP. Включение какого-либо адреса означает поддержку соответствующего транспорта.

useSpecifiedTransport – Это поле позволяет гейткиперу инструктировать конечную точку о том, какой транспортный протокол сигнализации следует использовать для выполнения соединения. Если это поле включено, а указанным транспортным протоколом является не **tcp**, то в это сообщение должно также включаться **alternateTransportAddresses**.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения. Например, оно позволяет гейткиперу проинструктировать выходной шлюз о выборе конкретных средств SCN, которые следует использовать для соединения.

usageSpec – Это поле может быть включено гейткипером, чтобы запросить конечную точку собирать и выдавать указанную информацию об использовании соединения в определенные моменты времени в этом соединении.

supportedProtocols – Это поле указывает протоколы, которые поддерживаются конечной точкой-получателем.

serviceControl – Содержит данные, специфичные для услуги (или ссылки на них), которые могли бы использоваться конечной точкой (к примеру, сообщение для воспроизведения на вызывающей стороне), как описано, например, в Приложении K/H.323.

multipleCalls – Если это поле установлено в ИСТИНА, то указывает, что конечная точка-получатель способна сигнализировать о нескольких соединениях по одному соединению сигнализации о соединении. Если установлено в ЛОЖЬ, то конечная точка-получатель не имеет этой способности. Если это поле отсутствует, то гейткипер не знает, имеет ли удаленная конечная точка эту способность.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

modifiedSrcInfo – Адрес-псевдоним, который следует использовать для конечной точки-источника, например, **dialledDigits**, **PartyNumber** (**e164Number** или **privateNumber**), либо **h323-ID**. Это поле следует использовать, когда адрес-псевдоним вызывающей конечной точки переносится/изменяется при попытке направить вызов к первоначальному получателю, либо к какой-либо альтернативной конечной точке. Эти адреса используются конечной точкой только для этого соединения.

7.11.3 Отклонение Допуска (AdmissionReject, ARJ)

Сообщение ARJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в ARQ.

rejectReason – Это причина, по которой запрос допуска был отклонен. Заметим, что **rejectReason,** установленная в **routeCallToSCN,** уместна только в случае, когда ARJ направляется к входному шлюзу (ARQ был послан шлюзом, а булево значение **answerCall** в ARQ установлено в ЛОЖЬ). Если **rejectReason** установлена в **routeCallToSCN**, то **rejectReason** при таком выборе содержит также телефонный номер, либо список телефонных номеров, к которым шлюз может перенаправить вызов в SCN, если шлюз поддерживает такую процедуру. Если **rejectReason** установлена в **exceedsCallCapacity**, то гейткипер определил, что получатель не имеет пропускной способности для приема этого соединения в данный момент времени. Поле **rejectReason** с причиной **collectDestination** означает, что гейткипер запрашивает, чтобы шлюз подобрал окончательный адрес получателя, и что поле **serviceControl** в ARJ указывает подсказку, которую следует выдать пользователю. Поле **rejectReason** с причиной **collectPIN** означает, что гейткипер запрашивает, чтобы шлюз подобрал персональный идентификационный номер или санкционированный код, и что поле **serviceControl** в ARJ указывает подсказку, которую следует выдать пользователю. Причина **genericDataReason** указывает, что запрос был отклонен из-за общего элемента или свойства; в этом случае в поле **genericData** может приводиться дополнительная информация. Конечной точке следует повторно регистрироваться с гейткипером, если она получила ошибку **invalidEndpointIdentifier**.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

callSignalAddress – Это адрес сигнализации о соединении, принадлежащий гейткиперу и выдаваемый от него, когда причиной отклонения является **routeCallToGatekeeper**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно

игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

serviceControl – Содержит данные, специфичные для услуги (или ссылки на них), которые могли бы использоваться конечной точкой (к примеру, для отображения причины неудачи вызова), как описано, например, в Приложении K/H.323.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.12 Запросы от терминала к гейткиперу об изменении полосы пропускания

Сообщение BRQ запрашивает, чтобы конечной точке была разрешена гейткипером измененная полоса пропускания в пакетной сети, а гейткипер либо удовлетворяет этот запрос с помощью сообщения BCF, либо отказывает с помощью BRJ.

Гейткипер может в BRQ запросить, чтобы конечная точка повысила или снизила используемую полосу пропускания. Если запрашивается повышение скорости, то конечная точка может ответить сообщением BRJ или BCF. Если запрашивается снижение скорости, то конечная точка должна ответить сообщением BCF, когда эта сниженная скорость поддерживается, или BRJ в остальных случаях.

7.12.1 Запрос Полосы Пропускания (BandwidthRequest, BRQ)

Сообщение BRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

endpointIdentifier – Это идентификатор конечной точки, который был присвоен терминалу в сообщении RCF.

conferenceID – Идентификатор соединения, у которого необходимо изменить полосу пропускания.

callReferenceValue – Значение справочного номера соединения из сообщений сигнализации о соединении H.225.0 для этого соединения; имеет только местный смысл. Оно используется гейткипером для связи BRQ с конкретным соединением.

callType – Используя это значение, гейткипер может попытаться определить "реальное" использование полосы пропускания.

bandWidth – Новая двунаправленная полоса пропускания, запрошенная для соединения, в сотнях бит/с. Это – абсолютное значение, которое охватывает только потоки битов аудио и видео, без учета заголовков и служебных данных. Одиночные многопунктовые потоки должны добавляться к общей полосе пропускания только один раз, даже если имеются несколько получателей этого потока носителя.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примеренной в настоящей Рекомендации.

gatekeeperIdentifier – Некоторый **gatekeeperIdentifier,** который конечная точка получила от гейткипера в списке **alternateGatekeeper** в RCF, когда она регистрировалась, или в предыдущем сообщении ARJ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и

секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

answeredCall – Устанавливается в ИСТИНА, указывая, что эта сторона была первоначальным получателем (эта сторона ответила на вызов).

callLinkage – Содержимое этого поле обычно контролируется услугой сцепления соединений. Процедуры и семантику этого поля см. в разделе 10/H.323.

capacity – Это поле указывает доступную для передающей конечной точки пропускную способность в данный момент времени, при этом предполагается, что гейткипер подтверждает BRQ путем передачи BCF. Конечная точка, передавая это поле, должна включать элемент **currentCallCapacity**.

usageInformation – Это поле позволяет конечной точке сообщать информацию об использовании для этого соединения. Гейткипер не должен включать это поле при передаче BRQ.

bandwidthDetails – Предоставляет информацию о полосе пропускания для каждого потока носителя, которые конечная точка в данное время передает или принимает в одном и том же блоке, например, в поле **bandwidth.** О каждом многопунктовом потоке сообщается только один раз, даже если имеются несколько получателей этого потока носителя.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.12.2 Подтверждение Полосы Пропускания (BandwidthConfirm, BCF)

Сообщение BCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в BRQ.

bandWidth – Максимальная разрешенная в этот момент времени полоса пропускания, кратная сотне бит/с.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

capacity – Это поле указывает доступную для передающей конечной точки пропускную способность в данный момент времени. Конечная точка, передавая это поле, должна включить элемент **currentCallCapacity.** Это поле не включается, когда BCF передается гейткипером.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.12.3 Отклонение Полосы Пропускания (BandwidthReject, BRJ)

Сообщение BRJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в BRQ.

rejectReason – Причина отклонения полосы пропускания гейткипером.

allowedBandWidth – Максимальная разрешенная в этот момент времени полоса пропускания, кратная сотне бит/с, включая текущее присвоение.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.13 Сообщения запроса местонахождения

Сообщение LRQ запрашивает о гейткипере, обеспечивающем преобразование адреса. Гейткипер отвечает сообщением LCF, содержащим транспортный адрес к получателю, или отклоняет запрос с помощью LRJ.

7.13.1 Запрос Местонахождения (LocationRequest, LRQ)

Сообщение LRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

endpointIdentifier – Это идентификатор конечной точки, который был присвоен терминалу в сообщении RCF.

destinationInfo – Последовательность адресов-псевдонимов для получателя, таких как **dialledDigits**, **partyNumber** (**e164Number** или **privateNumber**), либо **h323-ID**. Если гейткипером зарегистрирован по меньшей мере один псевдоним, и нет двух псевдонимов в LRQ, которые зарегистрированы для разных людей, то гейткипер должен воспринимать LRQ как ссылку на зарегистрированный идентификатор. В случае конфликта псевдонимов запрос местонахождения должен быть отключен с причиной **AliasesInconsistent**. Если гейткипер не обеспечивает этой проверки, то он должен рассматривать первый зарегистрированный адрес в качестве получателя.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

replyAddress – Транзитный адрес, к которому должны передаваться LCF/LRJ.

sourceInfo – Указывает отправителя LRQ. Гейткипер может использовать эту информацию для выбора способа ответа на LRQ.

canMapAlias – Это поле, если установлено в ИСТИНА, указывает, что когда полученное в ответ LCF будет содержать поля **destinationInfo**, **destExtraCallInfo** и/или **remoteExtensionAddress**, конечная точка может скопировать эту информацию соответственно в поля **destinationAddress**, **destExtraCallInfo** и **remoteExtensionAddress** сообщения Установить. Если гейткипер желает заменить адресную информацию из LRQ, а **canMapAlias** установлено в ЛОЖЬ, то этот гейткипер может отклонить LRQ. В системах, соответствующих H.225.0 версии 4 и выше, это поле должно устанавливаться в ИСТИНА.

gatekeeperIdentifier – Некоторый **gatekeeperIdentifier,** который конечная точка получила от гейткипера в списке **alternateGatekeeper** в RCF, когда она регистрировалась, или в предыдущем сообщении ARJ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

desiredProtocols – Указывает типы протоколов, в порядке предпочтения, которые инициирующая конечная точка желает для своего соединения (например, голос или факс). Решающий объект может использовать это поле для определения местонахождения конечной точки, которая тоже поддерживает определенный протокол, с учетом порядка предпочтений.

desiredTunnelledProtocol – Это поле указывает протокол, который требуется туннелировать.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

hopCount – Это поле определяет число гейткиперов, через которые это сообщение может пройти. Когда гейткипер получил LRQ и решил, что это сообщение следует переслать к другому гейткиперу, он сначала уменьшает на 1 поле **hopCount**. Если после этого **hopCount** больше 0, то гейткипер вводит новое значение числа участков в передаваемое сообщение. Если **hopCount** достигло 0, то гейткипер не должен пересылать это сообщение.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с сигнализацией управления соединением, применимой в настоящей Рекомендации. Передавая LRQ для поддержки сообщения ARQ или Установить, гейткипер должен скопировать идентификатор соединения из ARQ или Установить в LRQ. Конечная точка, которая передает LRQ для подготовки к инициации соединения, должна заполнить это поле идентификатором для соединения. Те LRQ, которые передаются вне связи с соединением, не будут содержать поля идентификатора соединения.

bandWidth – Двунаправленная полоса пропускания, запрошенная для соединения, в сотнях бит/с. Например, сигнализацией о соединении со скоростью 128 кбит/с будет запрос на 256 кбит/с. Значение относится только к скорости передачи битов аудио и видео, без учета заголовка и служебных данных.

sourceEndpointInfo – Последовательность адресов-псевдонимов для конечной точки-источника, таких как **dialledDigits**, **PartyNumber** (**e164Number** или **privateNumber**), либо **h323-ID**. Гейткипер скопирует информацию для конечной точки, по чьей заявке он посылает это LRQ, либо гейткипер, пересылая полученное LRQ, копирует **sourceEndpointInfo** из полученного LRQ.

canMapSrcAlias – Это поле, если установлено в ИСТИНА, указывает, что когда полученное в ответ LCF будет содержать **modifiedSrcInfo**, конечная точка может скопировать эту информацию в поле **sourceInfo** в сообщения Установить. Если LRQ передается гейткипером в результате получения ARQ, то гейткипер должен скопировать это поле из ARQ. Если гейткипер желает заменить адресную информацию из LRQ, а **canMapSrcAlias** установлено в ЛОЖЬ, то этот гейткипер будет отклонять LRQ.

7.13.2 Подтверждение Местонахождения (LocationConfirm, LCF)

Сообщение LCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в LRQ.

callSignalAddress – Транспортный адрес, к которому должна передаваться сигнализация о соединении H.225.0; использует надежный, широко известный или динамический порт, но может быть адресом конечной точки или гейткипера, в зависимости от используемой модели соединения.

rasAddress – Адрес регистрации, допуска и статуса (RAS) для конечной точки, местонахождение которой определяется.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

destinationInfo – Последовательность адресов-псевдонимов для получателя, таких как **dialledDigits**, **partyNumber** (**e164Number** или **privateNumber**), либо **h323-ID**.

destExtraCallInfo – Содержит внешние адреса для нескольких соединений.

destinationType – Определяет тип конечной точки-получателя.

remoteExtensionAddress – Содержит адрес-псевдоним вызываемой конечной точки в случаях, когда эта информация требует для прохождения нескольких шлюзов.

alternateEndpoints – Последовательность расположенных по приоритету альтернатив конечной точки для **callSignalAddress**, **rasAddress** или **destinationInfo**.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

alternateTransportAddresses – Это поле переносит адреса сигнализации о соединении для транспортных протоколов, отличающихся от TCP. Включение какого-либо адреса означает поддержку соответствующего транспорта.

supportedProtocols – Это поле указывает протоколы, которые поддерживаются конечной точкой.

multipleCalls – Если это поле установлено в ИСТИНА, то указывает, что конечная точка, местонахождение которой определяется, способна сигнализировать о нескольких соединениях по одному соединению сигнализации о соединении. Если установлено в ЛОЖЬ, то эта конечная точка не имеет этой способности. Если это поле отсутствует, то гейткипер не знает, имеет ли конечная точка эту способность.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

serviceControl – Это поле содержит адресную информацию, которую конечная точка может использовать при связи с сетью для управления услугой, относящейся к соединению, как описано, например, в Приложении K/H.323.

modifiedSrcInfo – Адрес-псевдоним, который следует использовать для конечной точки-источника, например **dialledDigits**, **PartyNumber** (**e164Number** или **privateNumber**), либо **h323-ID**. Это поле

следует использовать, когда адрес-псевдоним вызывающей конечной точки переносится/изменяется при попытке направить вызов к первоначальному получателю либо к какой-либо альтернативной конечной точке. Если сообщение LCF вызвало ответ ACF к конечной точке, то это поле должно быть скопировано в сообщение ACF.

bandWidth – Разрешенная максимальная полоса пропускания для соединения; может быть меньше запрошенной.

7.13.3 Отклонение Местонахождения (LocationReject, LRJ)

Сообщение LRJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в LRQ.

rejectReason – Это причина, по которой запрос местонахождения был отклонен. Если **rejectReason** установлена в **routeCallToSCN**, то **rejectReason** при таком выборе содержит также телефонный номер либо список телефонных номеров, к которым шлюз может перенаправить вызов в SCN, если шлюз поддерживает такую процедуру. Причина **resourceUnavailable** указывает, что полоса частот превышает используемую, либо что ни один объект, зарегистрированный в этом гейткипере, не имеет пропускной способности для организации соединения к запрошенному местонахождению в настоящее время. Причина **genericDataReason** указывает, что запрос был отклонен из-за общего элемента или свойства; в этом случае в поле **genericData** может приводиться дополнительная информация.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле указывает набор общих свойств, относящихся к этому соединению.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

serviceControl – Это поле содержит адресную информацию, которую конечная точка может использовать при связи с сетью для управления услугой, относящейся к соединению, как описано, например, в Приложении K/H.323.

7.14 Сообщения разъединения

7.14.1 Запрос Разъединения (DisengageRequest, DRQ)

Сообщение DRQ, передаваемое от конечной точки к гейткиперу, информирует гейткипер, что конечная точка отключается. Если оно передается от гейткипера к конечной точке, то DRQ вынуждает соединение отключиться; такой запрос не должен отвергаться. DRQ не передается непосредственно между конечными точками.

Заметим, что DRQ и сообщение Освобождение Завершено не идентичны, так как целью DRQ является информирование гейткипера об окончании соединения; гейткипер может не получить Освобождение Завершено, если он не оканчивает используемый канал сигнализации о соединении.

Сообщение DRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

endpointIdentifier – Это идентификатор конечной точки, который был присвоен терминалу в сообщении RCF.

conference ID – Идентификатор соединения, у которого необходимо освободить полосу пропускания.

callReferenceValue – Значение справочного номера соединения из сообщений сигнализации о соединении H.225.0 для этого соединения; имеет только местный смысл. Оно используется гейткипером для связи этого сообщения с конкретным соединением.

disengageReason – Причина изменения, которое было запрошено гейткипером или терминалом.

nonStandardData – Переносит информацию, не определенную этой Рекомендацией (например, данные о праве собственности).

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.

gatekeeperIdentifier – Некоторый **gatekeeperIdentifier,** который конечная точка получила от гейткипера в списке **alternateGatekeeper** в RCF, когда она регистрировалась, или в предыдущем сообщении ARJ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

answeredCall – Устанавливается в ИСТИНА, указывая, что эта сторона была первоначальным получателем (эта сторона ответила на вызов).

callLinkage – Содержимое этого поля обычно управляется услугой сцепления соединений. Процедуры и семантику этого поля см. в разделе 10/H.323.

capacity – Это поле указывает доступную для передающей конечной точки пропускную способность в данный момент времени, при этом предполагается, что гейткипер подтверждает DRQ путем передачи DCF. Конечная точка, передавая это поле, должна включать элемент **currentCallCapacity**. Это поле не включается, когда DRQ передается гейткипером.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

usageInformation – Это поле позволяет конечной точке сообщать информацию об использовании для этого соединения. Гейткипер не должен включать это поле при передаче DRQ.

terminationCause – Это поле описывает причину окончания соединения. Эта информация более конкретна, чем причина, приведенная в поле **disengageReason**. Гейткипер не должен включать это поле при передаче DRQ.

serviceControl – Содержит данные, специфичные для услуги (или ссылки на них), которые могли бы использоваться конечной точкой, как описано, например, в Приложении K/H.323. Гейткипер может использовать это поле для указания, что соединение закончилось из-за того, что окончился срок действия некоторого счета или исчерпана сумма, уплаченная за соединение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.14.2 Подтверждение Разъединения (DisengageConfirm, DCF)

Сообщение DCF содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в DRQ.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

capacity – Это поле указывает для передающей конечной точки пропускную способность соединения, доступную после того, как соединение, указанное в DCF, будет разъединено. Конечная точка, передавая это поле, должна включать элемент **currentCallCapacity**. Это поле не включается, когда DCF передается гейткипером.

circuitInfo – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

usageInformation – Это поле позволяет конечной точке сообщать информацию об использовании для этого соединения. Гейткипер не должен включать это поле при передаче DCF.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.14.3 Отклонение Разъединения (DisengageReject, DRJ)

Сообщение DRJ передается гейткипером, если конечная точка не зарегистрирована.

Сообщение DRJ содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в DRQ.

rejectReason – Причина отклонения запроса.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.15 Сообщения запроса статуса

Сообщение IRQ передается от гейткипера к терминалу, запрашивая информацию о статусе в форме сообщения IRR. Сообщение IRR может посылаться терминалом также через некоторые интервалы, указанные в сообщении ACF, без получения IRQ от гейткипера. Это сообщение не следует путать с сообщением Статуса сигнализации о соединении H.225.0.

Когда конечная точка посылает незатребованное IRR к гейткиперу версии 2 или выше, она может в поле **needResponse** указать, что она желает, чтобы гейткипер подтвердил получение IRR. В этом случае она заполняет поле **requestSeqNum** некоторым номером, не равным 1. Гейткипер отвечает сообщением IACK (положительное подтверждение) или INAK (отрицательное подтверждение), при этом должен возвратить тот же самый номер в поле **requestSeqNum**.

7.15.1 Запрос Информации (InfoRequest, IRQ)

Сообщение IRQ содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

callReferenceValue – Значение справочного номера соединения, о котором имеется вопрос. Если равно нулю, то это сообщение считается запросом сообщения IRR для каждого соединения, по которому терминал активен. Если терминал не активен по некоторым соединениям, то IRR должно посылаться в ответ на **callReferenceValue,** равное 9, со всеми соответствующими предусмотренными полями. Если **callReferenceValue** равно 0, то конечная точка должна игнорировать **callIdentifier** – в этом случае гейткипер должен установить **callIdentifier** = 0.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственника).

replyAddress – Транспортный адрес для передачи IRR, возможно, не адрес гейткипера.

callIdentifier – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примеренной в настоящей Рекомендации.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

uuiesRequested – Гейткипер может запросить конечную точку уведомлять этот гейткипер о сообщениях сигнализации о соединении H.225.0, которые конечная точка передает или принимает, если конечная точка указала эту способность в сообщении ARQ путем установки **willSupplyUUIEs** в ИСТИНА. Поле **uuiesRequested** указывает набор сообщений сигнализации о соединении H.225.0, о которых конечная точка должна посылать уведомления гейткиперу.

callLinkage – Содержимое этого поля обычно управляется услугой сцепления соединений. Процедуры и семантику этого поля см. в разделе 10/H.323.

usageInfoRequested – Это поле может быть включено гейткипером для запроса конечной точки о том, чтобы она выдавала в сообщении IRR указанную информацию об использовании соединения.

segmentedResponseSupported – Это поле указывает, будет ли гейткипер разрешать конечной точке выдавать информацию о соединении для всех соединений в нескольких сообщениях IRR, то есть "сегментами". Если это поле присутствует, то сегментация разрешена. В противном случае сегментация не разрешена. Это поле имеет смысл только тогда, когда гейткипер передает IRQ с **callReferenceValue,** установленным в 0, и не должно присутствовать в остальных случаях.

nextSegmentRequested – Если гейткипер передает сообщение IRQ с **callReferenceValue,** установленным в 0, и с включенным полем **segmentedResponseSupported**, то конечная точка может послать ответ IRR только с частью информации, отмеченной путем включения поля **segment** в IRR. Гейткипер может запросить следующий сегмент путем повторной передачи предыдущего сообщения IRQ с полем **nextSegmentRequested,** установленным в значение следующего сегмента, который гейткипер ожидает получить.

capacityInfoRequested – Это поле, если присутствует, указывает, что гейткипер запрашивает, чтобы конечная точка включила в IRR информацию о пропускной способности соединения.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.15.2 Ответ на Запрос Информации (InfoRequestResponse, IRR)

Сообщение IRR содержит следующее:

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

requestSeqNum – В случае запрошенного IRR это поле должно содержать порядковый номер из IRQ. В случае незапрошенного сообщения к гейткиперу версии 1 это поле должно содержать "единицу" (1). Во всех других незапрошенных IRR оно должно содержать некоторый монотонно увеличивающийся номер (который повторяется гейткипером в его ответе, если **needResponse** установлено в ИСТИНА).

endpointType – Дает информацию о конечной точке.

endpointIdentifier –Значение, назначенное гейткипером в RCF.

rasAddress – Адрес для регистрации, допуска и т. п.

callSignalAddress – Адрес сигнализации о соединении H.225.0.

endpointAlias – Псевдоним(ы) для конечной точки.

perCallInfo – Информация о конкретном соединении:

- **nonStandardData** – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).
- **callReferenceValue** – Значение справочного номера соединения (CRV) для того соединения, о котором дается ответ.
- **conferenceID** – Уникальный идентификатор конференции.
- **originator** – Если установлено в ИСТИНА, то конечная точка, получившая запрос, была инициатором соединения, а если установлено в ЛОЖЬ, то конечная точка была получателем соединения.
- **audio** – Информация об аудиоканале(ах). Должен быть включен элемент **multicast,** если сеанс является многопунктовым.
- **video** – Инфорация о видеоканале(ах). Должен быть включен элемент **multicast,** если сеанс является многопунктовым.
- **data** – Информация о канале(ах) передачи данных.
- **h245** – Транспортный адрес канала управления H.245.
- **callSignalling** – Транспортный адрес канала сигнализации о соединении H.225.0.
- **callType** – Дает информацию о топологии соединения.

- **bandwidth** – Текущее значение использования, кратное 100 бит/с; охватывает только аудио и видео, без учета заголовков и служебных данных.
- **callModel** – Указывает представление конечной точки о текущем использовании модели соединения.
- **callIdentifier** – Уникальный в глобальном масштабе, установленный инициирующей конечной точкой идентификатор соединения, который может быть использован для связи сигнализации RAS с измененной сигнализацией Q.931, примененной в настоящей Рекомендации.
- **tokens** – Это некоторые данные, которые могут потребоваться, чтобы разрешить определенную операцию. Эти данные, если они доступны, должны вводиться в сообщение.
- **cryptoTokens** – Зашифрованные **tokens**.
- **substituteConfIDs** – Перечисление всех идентификаторов конференции, полученных в сообщениях SubstituteCID H.245, относящихся к **conferenceID** первоначального поля **perCallInfo**.
- **pdu**:
  - **h323pdu** – Копия PDU H.225.0 и Q.931, запрошенного гейткипером в **uuiesRequested** сообщения ACF или IRQ.
  - **sent** – Установить в ИСТИНА, если конечная точка передана **h323pdu**; установить в ЛОЖЬ, если конечная точка получила **h323pdu**.
- **callLinkage** – Содержимое этого поля обычно управляется услугой сцепления соединений. Процедуры и семантику этого поля см. в разделе 10/H.323.
- **usageInformation** – Это поле позволяет конечной точке сообщать информацию об использовании для этого соединения.
- **circuitInfo** – Это поле дает информацию о канале или каналах SCN, используемых для этого соединения.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

needResponse – Если это поле установлено в ИСТИНА, а гейткипер указывал в RCF или ACF, что он будет отвечать на незапрошенные IRR (устанавливая **willRespondToIRR** в ИСТИНА), то гейткипер должен ответить сообщением IACK или INAK. Если гейткипер не указывал в RCF или ACF, что он будет отвечать на незапрошенные IRR (устанавливая **willRespondToIRR** в ЛОЖЬ), то гейткипер может игнорировать булево значение **needResponse**.

capacity – Указывает для передающей конечной точки пропускную способность соединения в этот момент времени. Передавая это поле, конечная точка должна включать элемент **currentCallCapacity** и будет включать **maximumCallCapacity** только в случае, когда отвечает на IRQ, содержащее элемент **capacityInfoRequested**.

irrStatus – Этот элемент следует выдавать в сообщении IRR в ответ на IRQ, переданное гейткипером. Отсутствие этого элемента означает, что это сообщение IRR содержит полную информацию о деталях соединения. Возможны следующие значения:

- **complete** – Указывает, что это IRR содержит последний сегмент информации о соединении в ответ на IRQ, который запросил все детали соединения. Когда сегментация не используется, это поле указывает, что это IRR содержит все детали соединения в одном сообщении IRR.

- **incomplete** – Указывает, что конечная точка не способна поместить всю запрошенную информацию о соединении в одно сообщение IRR при ответе на сообщение IRQ, содержащее **callReferenceValue,** установленное в 0.
- **segment** – Это поле указывает для этого сообщения IRR номер сегмента, который является значением, монотонно увеличивающимся по модулю 65536, когда передаются сегментированные IRR в ответ на IRQ, содержащее **callReferenceValue,** установленное в 0.
- **invalidCall** – Это поле указывает, что соединение, определенное в сообщении IRQ, не существует.

unsolicited – Конечные точки по H.323 версии 4 и выше должны устанавливать это поле в ИСТИНА в незапрошенных сообщениях IRR, как описано в 8.4.2/H.323, и должны устанавливать в ЛОЖЬ в запрошенных IRR.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.15.3 Подтверждение Запроса Информации (InfoRequestAck, IACK)

Сообщение IACK содержит следующее:

requestSeqNum – Это поле должно содержать **requestSeqNum,** которое было в IRR.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

7.15.4 Отрицательное Подтверждение Запроса Информации (InfoRequestNak, INAK)

Сообщение INAK содержит следующее:

requestSeqNum – Это поле должно содержать **requestSeqNum,** которое было в IRR.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

nakReason – Причина, по которой IRR был отрицательно подтвержден.

altGKInfo – Факультативная информация об альтернативных гейткиперах.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

7.16 Нестандартное Сообщение

NonStandardMessage (NSM) имеет следующую структуру:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.17 Сообщение Не Распознается

Это сообщение XRS посылается всякий раз, когда конечная точка H.323, получая сообщение RAS, не понимает его или не может декодировать его. В случаях, когда транспортный адрес получателя для этого сообщения XRS не доступен (то есть полученное сообщение RAS не могло быть декодировано), сообщение XRS может быть послано к транспортному адресу, от которого это нераспознаваемое сообщение RAS было получено. Этот транспортный адрес может быть получен из нижележащего транспортного уровня. Сообщение XRS не должно передаваться в ответ на входящее сообщение XRS. Конечные точки H.323 не будут передавать больше одного сообщения XRS в секунду к одному и тому же транспортному адресу, чтобы избежать перегрузки сети в ситуациях, когда принимаются искаженные сообщения.

requestSeqNum – Это должно быть **requestSeqNum** из неизвестного сообщения, если оно может быть декодировано. Если неизвестное сообщение не может быть декодировано, то это поле является монотонно увеличивающимся номером, уникальным для отправителя. Этот **requestSeqNum** следует использовать для обратной совместимости с конечными точками H.323 версии 3 и меньше. Конечные точки H.323 версии 4 и больше будут рассматривать параметр **messageNotUnderstood** для связывания этого XRS с каким-либо предыдущим переданным сообщением.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

messageNotUnderstood – Копия сообщения, которое было получено и не было распознано.

7.18 Сообщения доступности ресурсов шлюза

Индикация доступности ресурсов (RAI) является уведомлением от шлюза к гейткиперу о его текущей пропускной способности соединения для каждого протокола серии H и о скорости передачи данных для такого протокола. Гейткипер, получив RAI, отвечает Подтверждением доступности ресурсов (RAC) для уведомления о получении RAI.

7.18.1 Индикация Доступности Ресурсов (ResourcesAvailableIndicate, RAI)

Сообщение RAI содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

protocolIdentifier – Определяет марку передающей конечной точки.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

endpointIdentifier – Присвоенная гейткипером цепочка идентификатора конечной точки.

protocols – Указывает текущие скорости передачи данных для каждого протокола, который может поддерживаться при текущем состоянии устройства.

almostOutOfResources – Когда это поле установлено в ИСТИНА, устройство находится около или достигло пропускной способности. Любые действия на основе этого поля остаются на усмотрение изготовителя. Если устройство не находится около или не достигло пропускной способности, то это поле следует установить в ЛОЖЬ.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

capacity – Указывает для передающей конечной точки пропускную способность соединения в этот момент времени. Заметим, что если **capacity** выдается, то булево значение **almostOutOfResources** будет игнорироваться получателем так как поле **capacity** дает более детальную информацию; однако булево значение **almostOutOfResources** должно правильно устанавливаться, чтобы обеспечить обратную совместимость. Передавая поле **capacity,** конечная точка должна включать элементы **currentCallCapacity**.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.18.2 Подтверждение Доступности Ресурсов (ResourcesAvailableConfirm, RAC)

Сообщение RAC содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в RAI.

protocolIdentifier – Указывает марку принимающего гейткипера.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.19 Таймеры RAS и Запрос Продолжается (Request in Progress, RIP)

В таблице 23 показаны рекомендуемые безусловные ("по умолчанию") значения тайм-аутов для ответов на сообщения RAS и количество последующих повторений, если ответ не получен. (Эти значения могут быть изменены по результатам дальнейшего опыта и предложений реализаторов.)

Таблица 23/H.225.0 – Рекомендуемые значения тайм-аутов "по умолчанию"

Сообщение RAS	Значение тайм-аута (с)	Количество повторений
GRQ	5	2
RRQ (примечание 1)	3	2
URQ	3	1
ARQ	5	2
BRQ	3	2
IRQ	3	1
IRR (примечание 2)	5	2
DRQ	3	2
LRQ	5	2
RAI	3	2
SCI	3	2
ПРИМЕЧАНИЕ 1. – Значение тайм-аута следует пересчитывать на основе "времени жизни" (которое может быть указано гейткипером в сообщении RCF) и желательного числа повторений. ПРИМЕЧАНИЕ 2. – В случаях, когда от гейткипера ожидается ответ на незапрошенный IRR в виде IACK или INAK, может появиться тайм-аут, если на IRR не получен ответ.		

Если объект получает запрос от объекта версии 2 (или более поздней), ответ которому не может быть создан в пределах типового интервала тайм-аута с повторением, то он может послать сообщение RIP, указывающее интервал (в поле **delay**), после которого ответ будет создан. Как только ответ будет доступен, отвечающий объект передаст этот ответ, не ожидая окончания задержки из RIP. Если запросивший объект не получил ответа до истечения времени задержки из RIP, то он повторно передаст запрос. Отвечающий объект может затем послать дублирующий ответ или другое сообщение RIP. На рисунке 2 показан пример обмена сообщениями, демонстрирующий ряд аспектов стратегии повторений.

Поставщикам следует знать, что любые повторения будут влиять на время установления соединения, которое следует минимизировать. Поэтому желательны короткие интервалы повторения. Чтобы удаленные объекты могли предвидеть типовые интервалы повторения с целью решить, когда послать сообщение RIP, объектам следует избегать интервалов повторения менее 100 мс. Ожидается экспоненциальная функция выдержки и настройки для измеренных интервалов двойного времени

переноса. Объекты могут использовать измеренное двойное время переноса из процесса регистрации RRQ/RCF для изменения первоначальной установившейся оценки (порядка нескольких секунд) для этой цели. Объекты могут также использовать процесс регистрации для обмена номерами версий, чтобы гарантировать, что механизм повторения на основе RIP не используется, когда в сигнализации участвуют объекты версии 1.

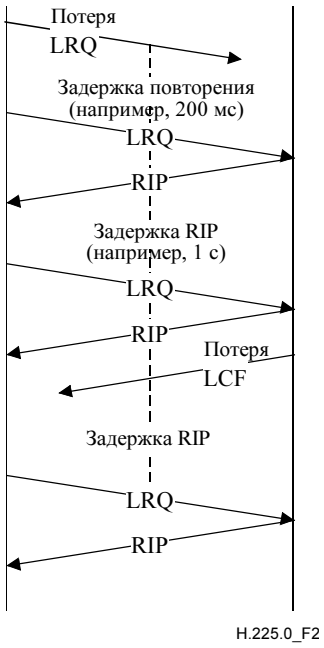


Рисунок 2/H.225.0 – Пример использования сообщения RIP

Сообщение RIP содержит следующее:

requestSeqNum – Это **requestSeqNum** из запроса, который в это время обрабатывается.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисленное значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

delay – Определяет количество времени в миллисекундах, в течение которого конечная точка должна ожидать перед попыткой повторения. Отвечающая конечная точка может ответить до окончания этого интервала.

7.20 Сообщения управления службой

7.20.1 Индикация Управления Службой (ServiceControlIndication, SCI)

Сообщение SCI передается от сервис-провайдера (поставщика службы) для указания пользователю службы, что может быть инициирован отдельный сеанс управления службой в направлении к заданному адресу. Оно может передаваться от гейткипера к конечной точке (например, для уведомления пользователя о свойствах службы) или от конечной точки к гейткиперу (например, для

загрузки в гейткипер сценария обработки соединения). Заметим, что объекты H.323 версии 3 или более ранние не способны декодировать это сообщение и поэтому не будут отвечать.

Сообщение SCI содержит следующее:

requestSeqNum – Это монотонно увеличивающийся номер, уникальный для отправителя. Номер должен передаваться обратно приемником в любых сообщениях, связанных с этим конкретным сообщением.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

serviceControl – Переносит набор информации для сеанса управления службой.

endpointIdentifier – Установлено в значение, полученное от гейткипера в сообщении RCF, если сообщение передается от конечной точки к ее гейткиперу.

callSpecific – Присутствует, если заданные сеансы относятся к одному конкретному соединению. Поля **callIdentifier**, **conferenceID** и **answeredCall** должны быть установлены в те же значения, которые были в сообщении ARQ, относящемся к этому служебному сеансу.

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.20.2 Ответ Управления Службой (ServiceControlResponse, SCR)

Сообщение SCR передается для подтверждения получения сообщения SCI, но не обязательно означает, что клиент службы будет инициировать сеанс, определенный в SCI.

Сообщение SCR содержит следующее:

requestSeqNum – Это должно быть тем же самым значением, которое было перенесено в SCI.

result – Это поле указывает результат обработки информации, содержащейся в сообщении SCI. Определены следующие значения:

- **started** – Запрошенное управление службой начато.
- **failed** – В запросе была некоторая ошибка, поэтому запрос неудачен.
- **stopped** – Управление службой остановлено.
- **notAvailable** – Запрошенное управление службой было недоступно в момент запроса.

nonStandardData – Переносит информацию, не определенную в этой Рекомендации (например, данные о праве собственности).

tokens – Это некоторые данные, которые могут быть необходимы для разрешения определенной операции. Эти данные должны вводиться в сообщение, если они доступны.

cryptoTokens – Зашифрованные **tokens**.

integrityCheckValue – Обеспечивает улучшенную целостность сообщения/аутентификацию сообщения в сообщениях RAS. Значение для проверки целостности на криптографической базе вычисляется отправителем, применяющим согласованный алгоритм обеспечения целостности и секретный ключ, по целому сообщению. До вычисления **integrityCheckValue** это поле должно игнорироваться и должно быть пустым. После вычисления отправитель помещает вычисляемое значение для проверки целостности в поле **integrityCheckValue** и передает это сообщение.

featureSet – Это поле определяет набор общих свойств.

genericData – Это поле является списком общих элементов, которые относятся к свойствам и которые определены за пределами основной спецификации H.225.0. Эти параметры могут использоваться, например, для прозрачного туннелирования информации через RAS.

7.21 Последовательность подтверждения допуска

AdmissionConfirmSequence является последовательностью из одного или нескольких сообщений ACF из RAS. Она может использоваться гейткипером для ответа на одно сообщение ARQ вместо одного сообщения ACF, когда он имеет разные маркеры безопасности, разную транслируемую информацию об источнике и т. п., которые нельзя просто выразить в одном сообщении ACF. Конечные точки указывают возможность приема AdmissionConfirmSequence, устанавливая в RRQ флаг **supportsACFSequences**.

8 Механизмы поддержания КО

8.1 Общий подход и допущения

Качество обслуживания (КО) при транспортировке по пакетной сети имеет следующие характеристики:

- коэффициент ошибок по битам;
- коэффициент потери пакетов;
- время переноса (задержка пакета в сети).

Любая транспортная сигнализация, относящаяся к КО (например, запрос к маршрутизатору о резервировании), выполняется терминалом, как только она возможна, либо гейткипером для него. Терминал может пожелать выполнить какие-либо резервирования, так как гейткипер может не быть, по логике, вблизи терминала или может быть неспособным выполнять запросы, касающиеся КО, от терминала. Средства, с помощью которых терминал или гейткипер выполняет КО или резервирования полосы пропускания, не входят в предмет рассмотрения этой Рекомендации.

Отчеты Передатчика и Приемника в протоколе RTCP должны быть теми средствами, с помощью которых КО будет оцениваться.

Имеются два типа задержек, связанных с перегрузкой, которые можно было бы измерять:

- кратковременные возрастания задержки, которые приводят к заметному, но не доставляющему беспокойства снижению скорости передачи кадров;
- общий рост задержки из-за перегрузки пакетной сети на такое время, что будет полезным применять механизм на основе обратной связи.

По существу, кратковременные пачки приравниваются к небольшому количеству ошибок, а долговременная перегрузка приравнивается к снижению мультимедийной нагрузки. Делается допущение, что все мультимедийные терминалы пакетной сети являются терминалами H.323 и что все они будут пытаться уменьшить использование пакетной сети, когда перегрузка растет быстрее, чем полоса пропускания, "украденная" друг у друга.

Ошибки по битам в пакетной сети обычно либо исправляются на нижележащем уровне, либо приводят к потере пакетов, поэтому они далее не рассматриваются в этом разделе.

Потеря пакетов требует, чтобы приемник был способен компенсировать потерянные пакеты способом, который снижает количество ошибок в максимально возможной степени. Для данных и сигналов управления используется повторная передача на транспортном уровне. Повторная передача для аудио и видео остается для изучения.

Заданный уровень транспортного КО приведет к некоторому уровню воспринимаемого пользователем КО аудио/видеоинформации, который частично зависит от эффективности методов, примененных для преодоления проблем транспортного КО.

8.2 Использование протокола RTCP при измерении КО

8.2.1 Отчеты передатчика

Отчет передатчика используется для трех главных целей:

1) позволяет синхронизировать несколько потоков RTP, таких как аудио и видео;

2) позволяет приемнику узнать ожидаемые скорость передачи данных и скорость передачи пакетов;

3) позволяет приемнику измерить расстояние до передатчика в виде времени переноса.

Первая цель из этих трех является самой близкой для этой Рекомендации. Изготовители могут использовать отчеты передатчика другими способами по своему усмотрению.

Полями, относящимися к синхронизации потоков, являются метка времени RTP и метка времени NTP в отчете передатчика RTCP. Метка времени NTP (если она имеется) дает время "стенных часов" и соответствует метке времени RTP, которая имеет те же единицы и случайный сдвиг, как метка времени захвата RTP в пакетах носителей информации.

8.2.2 Отчет приемника

Для измерения КО в этой Рекомендации используются четыре части отчета приемника:

1) потеря компонента;

2) совокупная потеря пакетов;

3) принятый увеличенный наибольший порядковый номер;

4) джиггер (разброс) интервалов между прибылями.

Пункты 2 и 3 используются для подсчета числа потерянных пакетов со времени предыдущего отчета приемника. Это может пониматься как долговременное измерение перегрузки пакетной сети. О вычислении замера см. раздел 6.4.4 RFC 3550 [37]. Если этот коэффициент потерь превышает значение, установленное изготовителем, то терминал H.225.0 должен снизить скорости передачи носителей на стороне пакетной сети согласно процедурам из 8.4. Если пункт 1 превышает значение, установленное изготовителем, то он тоже может быть пригодным для запуска корректирующего действия.

Если интервал между отчетами приемника превышает значение, установленное изготовителем, то терминал H.323 будет использовать пункт 1 в качестве индикатора серьезной перегрузки, требующей снижения скорости передачи носителей на стороне пакетной сети.

Пункт 4 следует использовать в качестве индикации приближающейся перегрузки. Если джиггер между прибытиями увеличивается до трех последовательных отчетов приемника, то передающему терминалу H.323 следует выполнить корректирующее действие.

8.3 Процедуры определения джиггера аудио/видео

В Рекомендации МСЭ-Т H.245 предусмотрены команды и процедуры для определения двойного времени переноса с помощью **RoundTripDelayRequest** и **RoundTripDelayResponse**. При многоточечных соединениях на запрос от конечной точки отвечает многоточечный контроллер (MC). RTCP содержит метод вычисления значений двойного времени переноса на основе сообщений Отчет Передатчика и Отчет Приемника. Заметим, что величины, измеряемые в этих случаях, не одни и те же, поэтому не будет конфликтов при использовании обоих методов измерения джиггера.

См. в 6.2.5/H.323 обсуждение возможности использования сигнализации на уровне H.245 для факультативного уменьшения задержек, связанных с джиггером.

8.4 Процедуры при расфазировке аудио/видео

См. в 6.2.6/H.323 обсуждение применения сигнализации на уровне H.245 для ограничения расфазировки между разными логическими каналами.

8.5 Процедуры поддержания КО

Существуют разные методы реагирования шлюзов/терминалов H.323 на увеличение потери пакетов или увеличение джиггера интервалов между прибытиями в удаленном приемнике. Эти методы можно сгруппировать в такие, которые подходят для быстрого реагирования на кратковременные проблемы, например, на потерянный или задержанный пакет, и в такие, которые подходят для реагирования на долговременные проблемы, например, на нарастающую перегрузку в пакетной сети. Заметим, что эти методы не пытаются поддерживать текущее качество обслуживания, а напротив, обеспечивают организованное ухудшение обслуживания. Должны соблюдаться приведенные ниже приоритеты, при которых носители, если они присутствуют, должны ухудшаться в следующем порядке: Видео, Данные, Аудио, Управление.

Кратковременные реакции

- уменьшение скорости передачи кадров на короткий промежуток времени: Это может вызвать в шлюзе H.323 передачу в пакетной сети в направлении к SCN дополнительных заполняющих кадров H.261, которые будут компенсировать пакеты в потоке;
- уменьшение скорости передачи пакетов путем переключения в факультативный режим, при котором аудио и видео смешиваются в одном пакете (для дальнейшего изучения);
- скорость передачи пакетов может быть уменьшена также путем использования фрагментации видеопотока на MB.

Долговременные реакции

- уменьшение битовой скорости передачи носителей (например, переключение с 384 кбит/с на 256 кбит/с): Это может охватывать простую инструкцию кодеру терминала или это может охватывать использование функции снижения скорости в шлюзе H.323. Об этих изменениях сигнализируется в командах **FlowControl** или с помощью сигнализации логического канала, смотря по обстоятельствам;
- включение носителя, имеющего меньшую важность (например, выключение видео, чтобы допустить большой объем трафика T.120);
- выдача к приемнику сигнала занятости (адаптивной занятости) в качестве указания на перегрузку пакетной сети. Это можно комбинировать с выключением носителя или даже всех носителей, кроме управления транспортным портом. Адаптивная занятость передается как значение причины Q.931 в сообщении Освобождение Завершено.

Следует отметить, что затруднено реагирование на джиггер интервалов между прибытиями в тракте с многими маршрутизаторами, где большой процент пакетов прибывает с нарушением порядка следования. Может оказаться невозможным различать этот источник джиггера от других источников или обосновывать стратегию исправления ошибок по измеренному джиггеру. Однако потеря пакетов является поддающейся количественному определению и точно выражающей величиной.

8.6 Управление эхом

Ответственность за управление акустическим эхом лежит на терминале серии H. Обычно, учитывая задержку, вносимую компрессией аудио/видео, предполагают, что все терминалы H.320, H.323 и H.324 имеют некоторую форму управления эхом (компенсацию или переключение).

Однако, когда терминал H.323 соединен с телефоном коммутируемой телефонной сети общего пользования (КТСОП), обычно телефон КТСОП не имеет управления эхом. Поэтому пользователь терминала H.323 может слышать акустическое эхо, возвращенное со стороны КТСОП. Это возвращенное акустическое эхо можно минимизировать путем применения устройства "громкой связи" с управлением эхом либо путем применения портативного или ушного телефона. Изготовители могут добавить затухание в аудиотракт, когда терминал H.323 соединяется с телефонным аппаратом обычной аналоговой телефонной связи КТСОП.

Управление эхом дифсистемы (перехода с 2-проводного тракта на 4-проводный). Дифсистема (дифференциальная система) обеспечивает стык между 4-проводными системами передачи и 2-проводными терминалами. Для голосовых соединений цифровой сети с интеграцией служб (ЦСИС), которые проходят через КТСОП на скорости 64 кбит/с, эхокомпенсатор не нужен. Для соединений передачи данных на 64 кбит/с эхокомпенсация не разрешается.

В случае распределенного шлюза с взаимосвязью по сети системы сигнализации № 7 (SS7) уведомления о наличии эхокомпенсации переносятся в сигнальных сообщениях подсистемы ISUP, как определено в Рекомендации МСЭ-Т Q.115. Контроллер шлюза носителей (Media Gateway Controller,

MGC) H.323 может распознавать эту информацию сигнализации и включать или нейтрализовать эхокомпенсатор в шлюзе носителей (Media Gateway, MG). Для голосовых соединений MGC может включать эхокомпенсатор, не внося вредного влияния на качество речи, даже если эхокомпенсатор имеется в КТСОП.

Для соединений передачи данных в полосе тональных частот (модемных соединений), которые проходят транзитом через сеть H.323 или заканчиваются в ней, управление эхокомпенсацией обеспечивается модемами с помощью внутриполосных тонов. Никакой внеполосной сигнализации не требуется для элементов КТСОП или для контроллеров MGC.

Приложение A

Протоколы RTP/RTCP

RTP и RTCP определены в источнике [37]. Этот источник указан также в Добавлении I. Настоящее Приложение и Добавление I сохранены в этой Рекомендации для поддержания эквивалентности с более ранними версиями этой Рекомендации.

Читателям следует заметить, что все ссылки в [37] относятся к библиографии и не являются нормативными, кроме ссылки на ISO/IEC 10646-1, которая включена также в раздел "Библиографические ссылки" этой Рекомендации.

Читателям следует также заметить, что используемая в [37] терминология отличается частично от той, которая используется в Рекомендации МСЭ-Т H.323 и в настоящей Рекомендации, как показано в таблице A.1.

Таблица A.1/H.225.0 – Соответствие терминологии

Термин из H.323 и H.225.0	Термин из [37] (RTP/RTCP)
media stream (поток носителей)	data (данные)
transport address (транспортный адрес)	transport address (транспортный адрес)
packet-based network address (адрес пакетной сети)	network address (сетевой адрес)
TSAP identifier (идентификатор TSAP)	port (порт)
Annex A (Приложение A)	specification or document (спецификация или документ)
shall (должен)	must (должен, обязан)
should (следует)	should (следует)

Кроме того, следует заметить что "трансляторы" и "смесители" не являются элементами системы H.323. Конечные точки H.323, такие как шлюзы и блоки MCU, имеют некоторые характеристики трансляторов и смесителей, поэтому данный текст сохранен в качестве руководства для реализатора. Однако поддержка трансляторов и смесителей не является частью H.323, и эти разделы должны рассматриваться как информативные.

Приложение B

Профиль RTP

Профиль RTP определен в источнике [38]. Этот источник указан также в Добавлении II. Настоящее Приложение и Добавление II сохранены в этой Рекомендации для поддержания эквивалентности с более ранними версиями этой Рекомендации.

См. введение к Приложению A; все сделанные там предупреждения применимы также к этому Приложению.

Приложение C

Формат полезной нагрузки RTP для видеопотоков H.261

Формат полезной нагрузки RTP для видеопотоков H.261 определен в источнике [39]. Этот источник указан также в Добавлении III. Настоящее Приложение и Добавление III сохранены в этой Рекомендации для поддержания эквивалентности с более ранними версиями этой Рекомендации.

См. введение к Приложению A; все сделанные там предупреждения применимы также к этому Приложению.

Приложение D

Формат полезной нагрузки RTP для видеопотоков H.261A

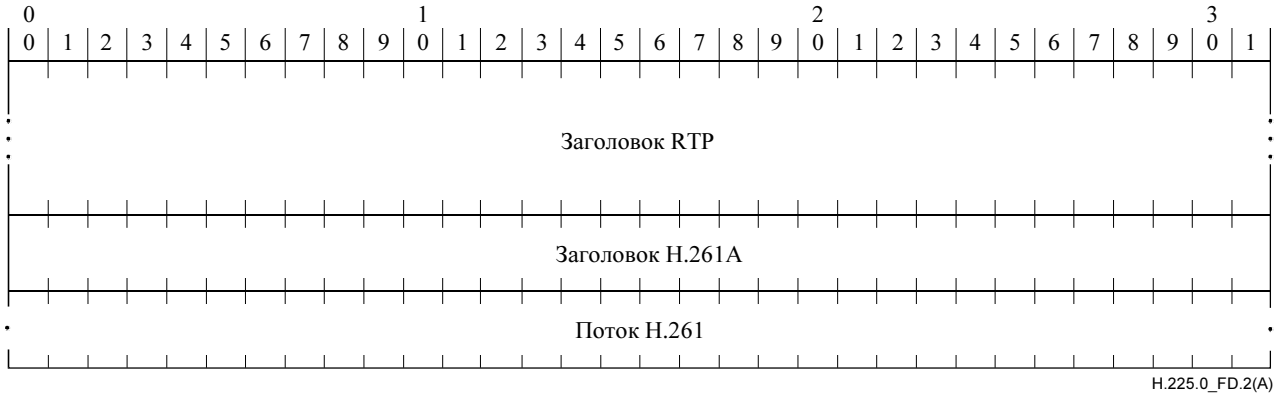
D.1 Введение

С целью упрощения стыковки видеопотоков H.323 с SCN через шлюзы, в Рекомендации МСЭ-Т H.323 определена модифицированная форма полезной нагрузки RTP H.261 для видео. Это облегчает управление буферным накопителем и взаимодействие с удаленными кодеками SCN. О поддержке типа полезной нагрузки H.261A сообщается с помощью набора возможностей H.245 и в сообщении **openLogicalChannel,** используя динамический тип полезной нагрузки RTP.

D.2 Пакетирование RTP в H.261A

Эта версия является расширением версии, описанной в Приложении C, за исключением того, что к заголовку H.261 прибавляется дополнительное 32-битовое слово. Процедуры, описанные в Приложении C, применимы также к этому Приложению.

Данные H.261A будут следовать за заголовком RTP, как показано ниже:



Заголовок H.261A определяется следующим образом:

Поля в заголовке H.261A имеют следующие значения:

Заголовок H.261: 32 бита – Как описано в Приложении C.

Последний номер GOB (LGOBN): 4 бита – Номер GOB последнего GOB в пакете RTP (максимальным номером GOB является 12 для Рекомендации МСЭ-Т H.261).

Res: Зарезервировано.

Подсчет байтов: 24 бита – Указывает накопленное число октетов, которые были переданы в части потока H.261 пакетов RTP. Если последний байт пакета заполнен только частично (что указывает EBIT), то он не считается в накопленном подсчете байтов. Этот подсчет байтов с модулем 2^{24} начинается с некоторого случайного значения и никогда не сбрасывается.

Оба эти дополнительные поля могут быть использованы, когда пакеты потеряны или доставлены с нарушением порядка следования. Подсчет байтов может использоваться для определения количества потребных стаффингов (вставок) в потоке SCN и облегчает управление буферным накопителем. Последний номер GOB упрощает определение тех GOB, которые были потеряны из-за потери пакетов.

Приложение E

Пакетирование видео

В этом приложении описываются детали пакетирования RTP для видеокодеков. В таблице E.1 приведены ссылки на определения форматов пакетирования видео, которые не определены в этой Рекомендации. В остальных разделах этого Приложения определяются дополнительные форматы пакетирования видео.

Таблица E.1/H.225.0 – Внешние определения форматов пакетирования видео

Имя кодирования	Определение пакетирования
ISO/IEC 14496-2 (MPEG-4 Video)	IETF RFC 3016, *RTP Payload Format for MPEG-4 Audio/Visual Streams*

E.1 H.263

Один из форматов полезной нагрузки RTP для видео H.263 определен в RFC 2190 Группы IETF для битовых видеотоков H.263, которые не имеют новых свойств, одобренных в версии 2 (версии 1998 года) Рекомендации МСЭ-Т H.263 (свойств, использующих PLUSPTYPE или приложения, следующие за Приложением H/H.263). Дополнительный формат полезной нагрузки, который поддерживает улучшенные свойства битовых потоков H.263 версии 2, будет определен позже. Существующий формат пакетирования, широко применяемый в промышленности (не определенный в IETF RFC 2190), может использоваться только в случаях, когда корреспондент сообщает о поддержке этого формата при обмене возможностями.

В разделе 5 RFC 3551 [38] описывается процедура, применяемая для передачи видеопотоков H.263.

Приложение F

Аудио- и мультиплексированное пакетирование

В этом приложении описываются детали пакетирования RTP для аудиокодеков. В таблице F.1 приведены ссылки на определения форматов пакетирования аудио, которые не определены в этой Рекомендации. В таблице F.2 приведены ссылки на определения форматов мультиплексированного пакетирования. В остальных разделах этого Приложения определяются дополнительные форматы пакетирования аудио.

Таблица F.1/H.225.0 – Внешние определения форматов пакетирования аудио

Имя кодирования	Определение пакетирования
ISO/IEC 14496-3 (MPEG-4 Audio)	IETF RFC 3016, *RTP Payload Format for MPEG-4 Audio/Visual Streams*

Таблица F.2/H.225.0 – Внешние определения форматов пакетирования мультиплексированного потока

Имя кодирования	Определение пакетирования
Мультиплексированные потоки H.222 (транспортные потоки MPEG-2)	IETF RFC 2250, *RTP Payload Format for MPEG1/MPEG2 Video*

F.1 Рекомендация МСЭ-Т G.723.1

Эта Рекомендация определяет кодированное представление, которое может использоваться для компрессии компонентов речевого сигнала мультимедийных служб на очень низкой битовой скорости. Кадр G.723.1 может иметь один из трех размеров: 24 байта (кадр для 6,3 кбит/с), 20 байтов (кадр для 5,3 кбит/с) или 4 байта. Эти 4-байтовые кадры называются кадрами Описателя введения пауз (Silence Insertion Descriptor frames, SID) и используются для определения параметров комфортного шума. Ограничения на то, как смешиваются 4-, 20- и 24-байтове кадры, отсутствуют. Два бита младших разрядов в первом октете кадра определяют размер кадра и тип кодека (см. в таблицах 5 и 6/G.723.1 дополнительную информацию о порядке следования битов). Возможно переключение между двумя скоростями на границах кадра каждые 30 мс. Обе скорости (5,3 кбит/с и 6,4 кбит/с) являются обязательным компонентом кодера и декодера. Такой кодер был оптимизирован для представления речи с качеством, близким к качеству междугородной связи, на вышеуказанных скоростях, с использованием ограниченного уровня сложности.

Все биты кодированного потока битов всегда передаются, начиная от бита младшего разряда к биту старшего разряда. Заметим, что это относится к порядку следования битов, представленных к транспортному уровню, а не к порядку следования битов в проводе.

Пакетирование G.723.1 соответствует Приложению B, за исключением интервала пакетирования (30 мс вместо 20 мс "по умолчанию"):

1) Первый пакет всплеска речи (первый пакет после интервала паузы) отмечается установкой бита-маркера в заголовке данных RTP.

2) Частота дискретизации (тактовая частота RTP) равна 8000 Гц.

3) Интервал пакетирования должен иметь длительность 30 мс (один кадр), что отличается от пакетизации "по умолчанию" 20 мс.

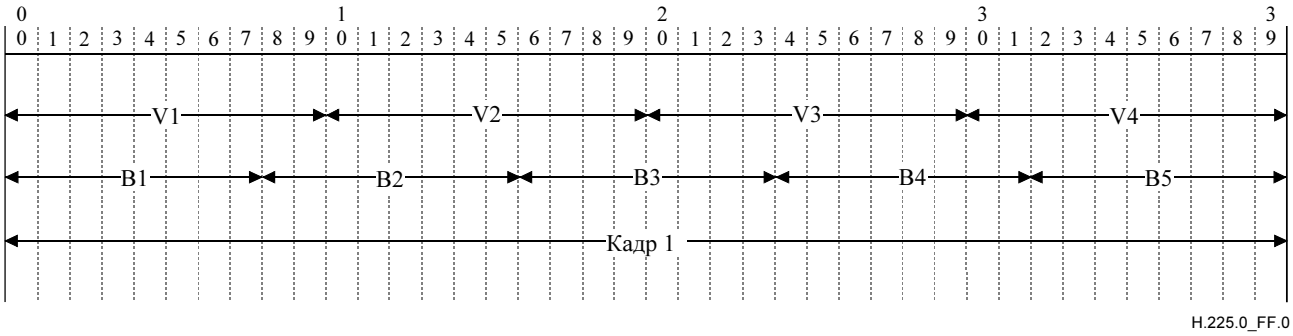
4) Кодеки будут способны кодировать и декодировать отдельные последовательно идущие кадры внутри одного пакета.

5) Приемник будет принимать пакеты, представляющие аудиоданные от 0 до 180 мс, а не от 0 до 200 мс "по умолчанию".

F.2 Рекомендация МСЭ-Т G.728

1) *Пакетирование кадра*

Кадр G.728 (4 массива: V1-V4, по 10 битов каждый, V1 является старшим – должен воспроизводиться первым) организован в 5 байтах (B1-B5). Как показано ниже на рисунке, принципом для порядка следования битов является "поддержание битовых разрядов". Биты из более старых массивов ("векторов") являются старшими разрядами по сравнению с битами новых массивов. Бит старшего разряда (MSB) кадра становится MSB в B1, а бит младшего разряда (LSB) кадра становится LSB в B5. Для ясности: биты более старших разрядов из каждого массива помещаются в биты более старших разрядов в B1-B5 (в биты более старших разрядов B байт с наименьшим номером).



Например:

B1 содержит 8 битов старших разрядов из V1, MSB из V1 является MSB в B1.

B2 содержит 2 бита младших разрядов из V2, более старший из этих двух – в его MSB, а также 6 битов старших разрядов из V2, самый старший разряд из них является также самым старшим разрядом в B2.

B1 должен помещаться в пакет первым (самый старший байт в RTP), а B5 – последним.

2) *Многокадровое пакетирование*

Окончание единственного кадра в пакете RTP могло бы привести к значительной избыточной нагрузке для сети. Поэтому допускается передача многокадрового пакета следующим образом:

Пакет RTP по G.728 должен содержать целое число кадров.

Более старые кадры (которые должны воспроизводиться первыми) должны помещаться первыми в пакет RTP.

Метка времени будет отражать время взятия первого отсчета в первом массиве (V1) первого кадра (самой старой информации в этом пакете).

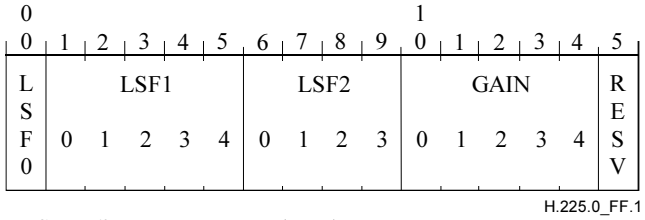
3) Бит-маркер должен сохранять тот же самый смысл, который был присвоен ему в этой Рекомендации.

F.3 Рекомендация МСЭ-Т G.729

Эта Рекомендация определяет кодированное представление, которое может использоваться для компрессии компонентов речевого сигнала мультимедийных служб на битовой скорости 8 кбит/с. Этот кодер был оптимизирован для представления речи на 8 кбит/с с качеством междугородной связи или проводной линии. Этот кодер имеет присущую ему устойчивость к случайным битовым ошибкам, а также к случайным или групповым стираниям кадров. Он представляет речь с высоким качеством при работе в среде с помехами. Версия алгоритма G.729 с уменьшенной сложностью описана в Приложении A/G.729. Версия с "плавающей запятой" для этих двух алгоритмов описана в Приложении C/G.729. Алгоритмы кодирования речи в основном тексте Рекомендация МСЭ-Т G.729 и в Приложениях A и C/G.729 обеспечивают полное взаимодействие друг с другом, поэтому не требуется далее их различать.

Рекомендуется алгоритм Детектора активности голоса (Voice Activity Detector, VAD) и Генератора комфортного шума (Comfort Noise Generator, CNG) из Приложения B/G.729. Этот алгоритм применен к Приложению F/G.729 (6,4 кбит/с с VAD/CNG), к Приложению G/G.729 (11,8 кбит/с с VAD/CNG),

к Приложению B/G.729 (G.729 и Приложение A/G.729 с VAD/CNG) и к Приложению I/G.729. Кадр G.729 или Приложения A/G.729 содержит 10 октетов; кадр Приложения D/G.729 содержит 8 октетов; кадр Приложения E/G.729 содержит 15 октетов; а кадр комфортного шума из Приложений B/G.729, F/G.729 и G/G.729 занимает 2 октета, как показано на рисунке F.1.



RESV = Зарезервировано (нуль)

Рисунок F.1/H.225.0 – Формат пакетирования кадра комфортного шума из Приложений B/G.729, F/G.729 и G/G.729

Передаваемые параметры кадра с длительностью 10 мс из G.729, Приложения A/G.729 или Приложения C/G.729, который содержит 80 битов, определены в таблице 8/G.729. Размещение этих параметров показано на рисунке F.2. Биты нумеруются согласно порядку следования в Интернете, то есть битом старшего разряда является бит 0.

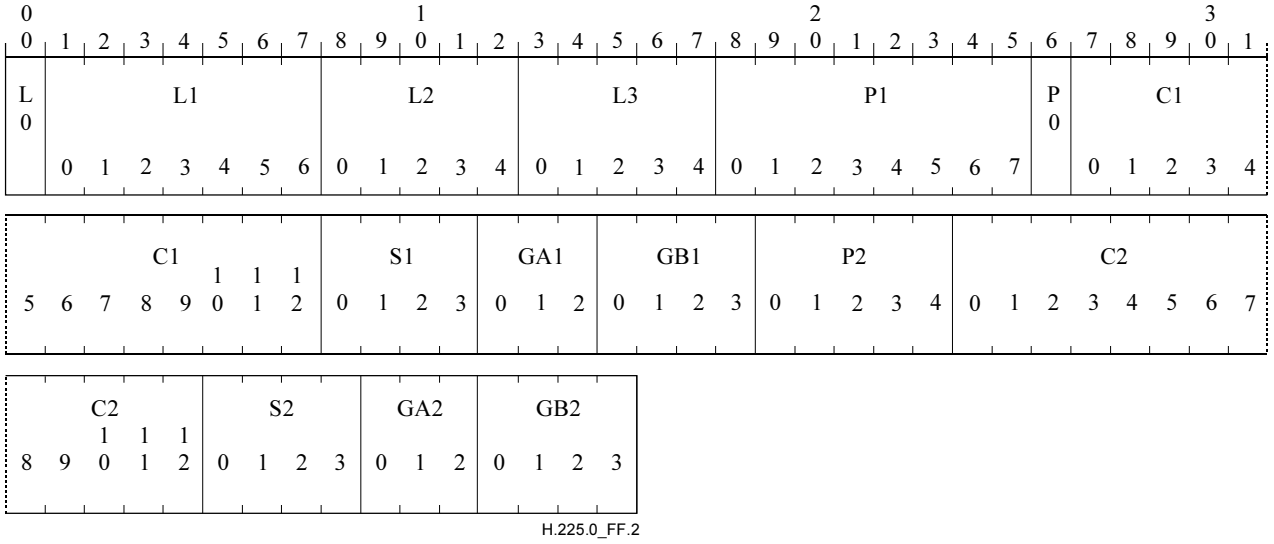


Рисунок F.2/H.225.0 – Формат пакетирования из G.729, Приложения A/G.729 и Приложения C/G.729

Приложение D/G.729 определяет расширение G.729 со скоростью 6,4 кбит/с для моментального снижения пропускной способности канала, например, для реагирования на состояния перегрузки. Приложение E/G.729 дает расширение G.729 со скоростью 11,8 кбит/с для получения лучших рабочих характеристик при широком диапазоне входных сигналов, таких как речь с фоновой помехой и музыка. Кроме того, Приложение E/G.729 имеет два рабочих адаптивных режима, обратный и прямой, о которых сигнализируют первые два бита в заголовке пакета.

Биты кадра G.729-6.4 форматируются, как показано на рисунке F.3 (см. таблицу D.1/G.729). Биты нумеруются согласно порядку следования в Интернете, то есть битом старшего разряда является бит 0. Используются всего 64 бита.

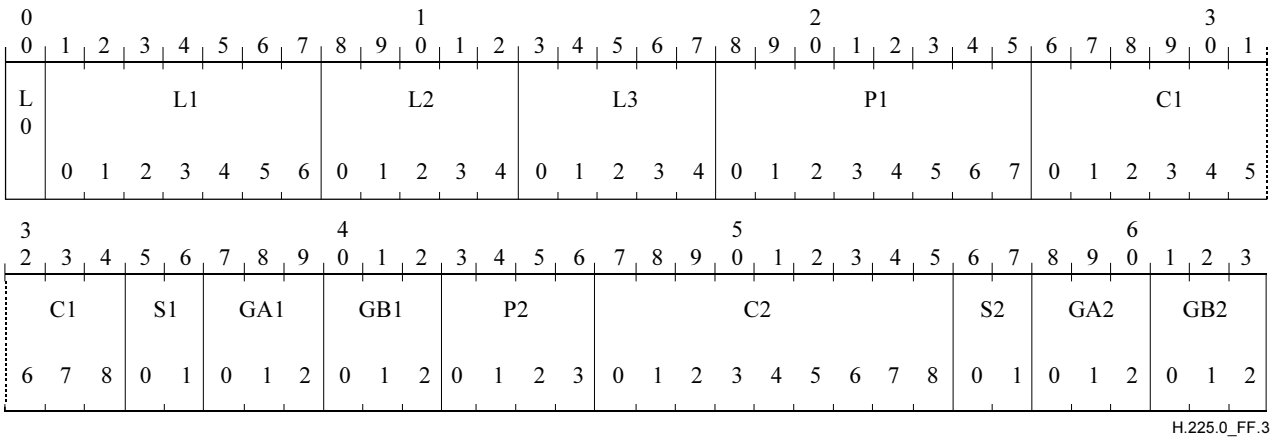


Рисунок F.3/H.225.0 – Формат пакетирования для G.729-6.4

Эффективная битовая скорость для алгоритма Приложения E/G.729 равна 11,8 кбит/с, а всего используется 118 битов. Биты кадра G.729-12 форматируются, как показано на рисунках F.4 и F.5 (см. таблицу E.1/G.729). Рисунки F.4 и F.5 для алгоритма Приложения E/G.729 определяют поля для прямого адаптивного режима и обратного адаптивного режима соответственно. Два бита младших разрядов включены в качестве битов "не обращать внимания", которые используются для завершения целого числа октетов в кадре.

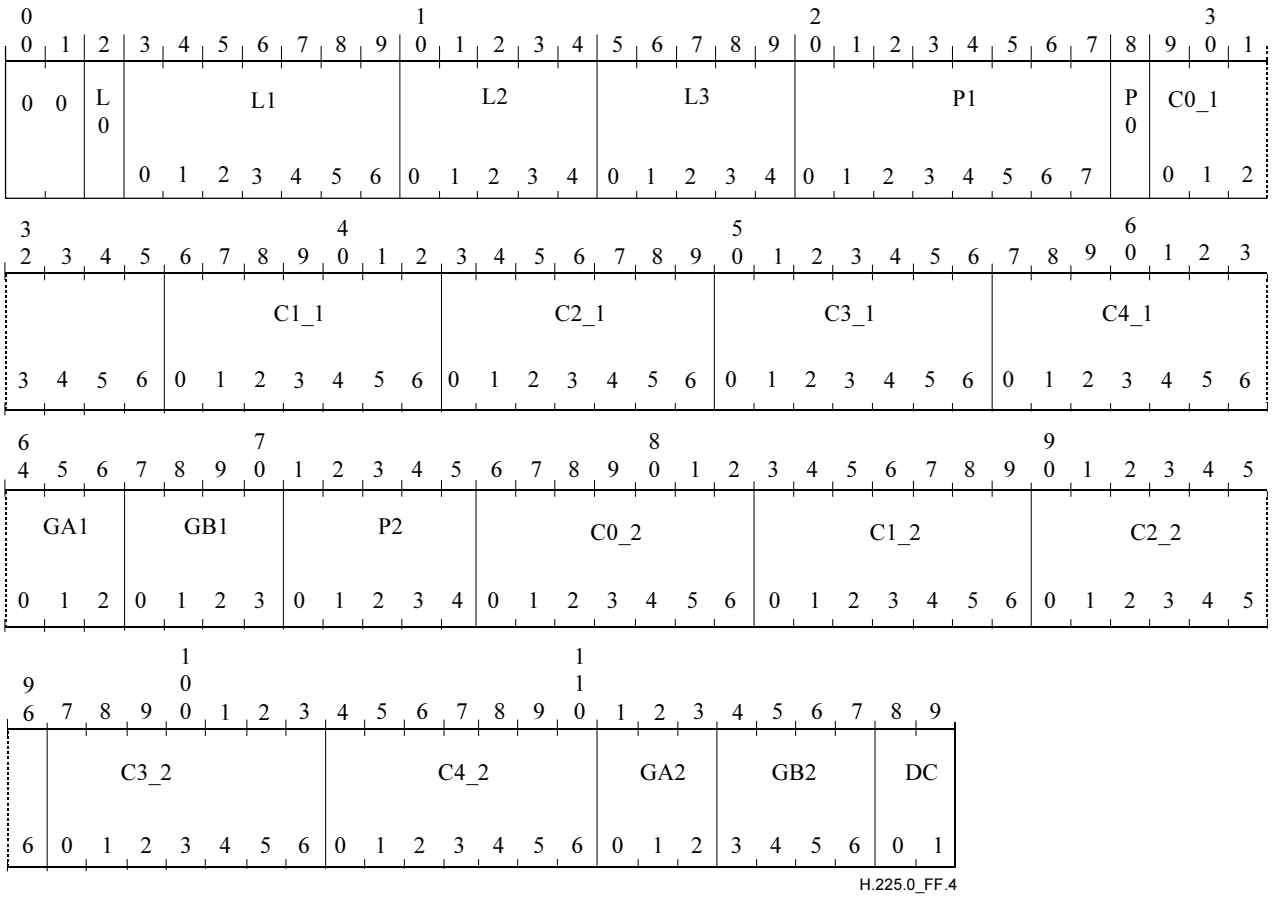


Рисунок F.4/H.225.0 – Формат пакетирования для G.729-12 в прямом адаптивном режиме

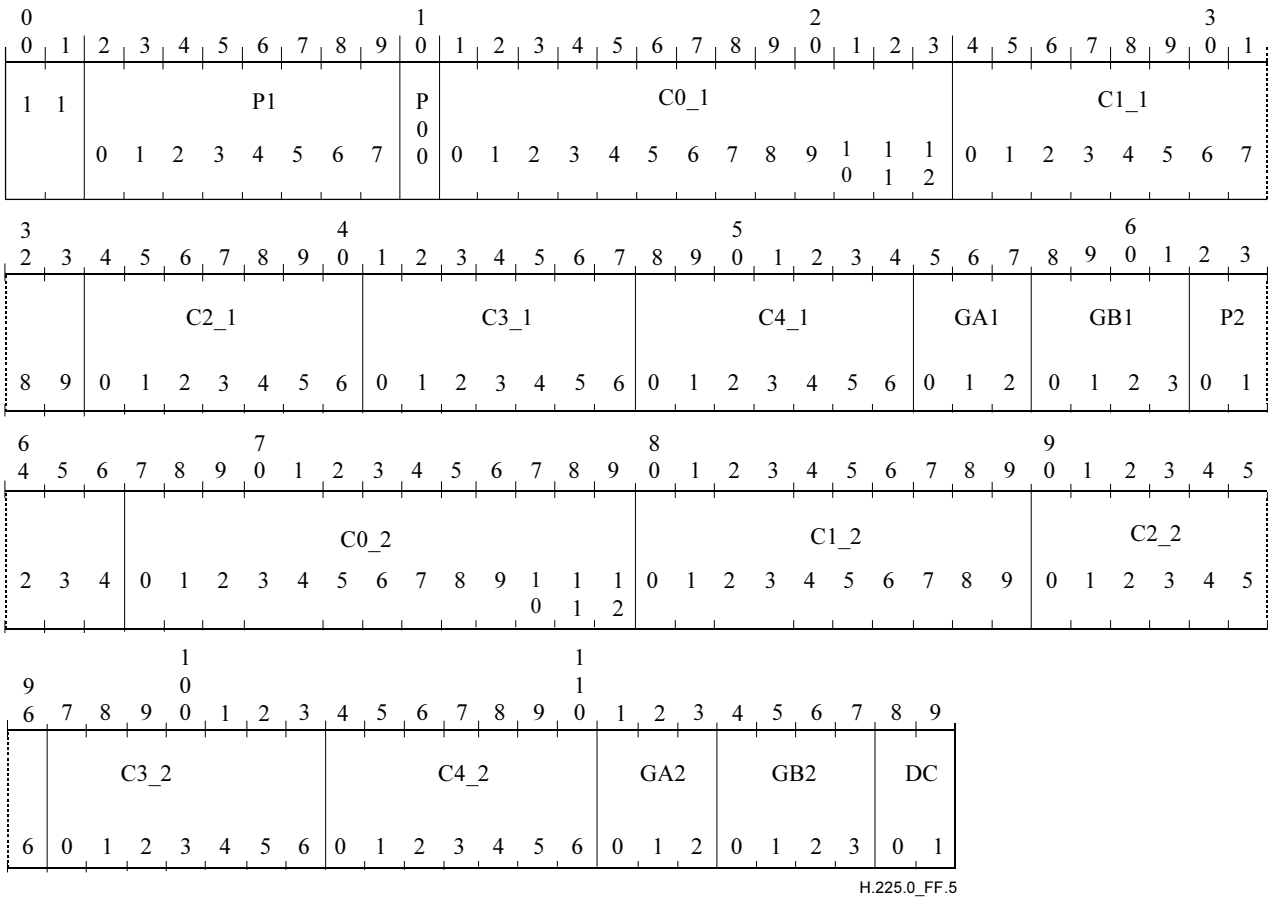


Рисунок F.5/H.225.0 – Формат пакетирования для G.729-12 в обратном адаптивном режиме

Пакет RTP может содержать нуль или больше кадров G.729 либо кадров Приложения A, C, D или E/G.729, за которыми следует нуль или одна полезная нагрузка RTP. Присутствие кадра комфортного шума может зависеть от длины полезной нагрузки RTP.

1) Первый пакет всплеска речи (первый пакет после интервала паузы) отмечается установкой бита-маркера в заголовке RTP.

2) Частота дискретизации (тактовая частота RTP) равна 8000 Гц.

3) Интервал пакетирования "по умолчанию" имеет длительность 20 мс. Несмотря на то, что 20 мс является определенно рекомендованным значением, в некоторых ситуациях может быть желательным передавать пакеты с длительностью 10 мс. Например, учтем переход "от голоса к не-голосу" в первые 10 мс пакета. Если интервал пакетирования 20 мс является обязательным, то передатчик должен был бы ждать, когда речь снова станет активной.

4) Кодеки будут способны кодировать и декодировать отдельные последовательно идущие кадры внутри одного пакета.

5) Приемник будет принимать пакеты, представляющие аудиоданные от 0 до 200 мс, а не от 0 до 200 мс "по умолчанию".

F.4 Подавление пауз

В Рекомендации МСЭ-Т H.225.0 установлено, что кодеры могут передавать кадры паузы перед остановкой передачи в интервале паузы. Так как не все аудиокодеры имеют внутриполосную сигнализацию о паузе, следует определить общий механизм на уровне RTP. Примером могла бы служить передача пустого пакета RTP. Это остается для изучения.

F.5 Кодеки GSM

Речевые кодеки GSM (Global System for Mobile communication, Глобальная система подвижной связи) охватывают: GSM с полной скоростью (FR) [F-1], GSM с половинной скоростью (HR) [F-2] и GSM с расширенной полной скоростью (EFR) [F-3]. Каждый кодек выдает три разных типа кадров речевого трафика, а именно:

- Речевые кадры: Содержат реальные речевые данные.
- Холостые кадры: Указывают на отсутствие голосовой активности, все биты данных установлены в единицу.
- Кадры Описателя паузы (Silence Descriptor, SID): Указывают начало интервала паузы, а данные описывают фоновый шум. Кадры SID отмечаются внутриполосно некоторой фиксированной комбинацией битов.

F.5.1 Пакетирование кадров

У всех трех кодеков GSM биты кадра речевого трафика пакетируются в кадр RTP, первым идет бит старшего разряда (MSB). Один пакет RTP может содержать один или больше кадров речевого трафика GSM. Все конечные точки должны быть способны принимать и распознавать холостой кадр. Холостой кадр речи GSM заполняется двоичными 1.

Если конечная точка устанавливает параметр comfortNoise в ИСТИНА, то она должна передавать кадры SID, как определено в спецификациях комфортного шума и прерывающейся передачи (Discontinuous Transmission, DTX) для конкретного кодека GSM. Во время интервала паузы периодически передается новый кадр SID с (возможно) обновленной шумовой информацией, а именно каждый 24-й кадр. После интервала паузы бит-маркер в заголовке RTP должен быть установлен в 1.

Кодек с полной скоростью

Кодек GSM с полной скоростью передает кадр из 260 битов (32,5 октета) каждые 20 мс. Эта информация должна пакетироваться в кадр RTP с четырехбитовым префиксом (0xD, или двоичный 1101), который называется сигнатурой. Поэтому полезная нагрузка кодера GSM FR в RTP должна содержать 33 октета. Кадр SID (Описателя паузы) отмечается внутриполосно некоторым кодовым словом SID, записанным в параметрах кодека, как описано в приведенном ниже источнике [F-4]. Размер полезной нагрузки кадра SID составляет 33 октета. Сигнатура в кадре SID с полной скоростью должна быть такой же, как в речевом кадре с полной скоростью (0xD). Речь FR, кодированная в RTP, должна иметь битовую скорость 13 200 бит/с без учета служебной информации пакетирования.

Кодек с половинной скоростью

Кодек GSM с половинной скоростью передает кадр из 112 битов (14 октетов) каждые 20 мс. Эта информация должна пакетироваться в заголовок RTP без каких-либо префиксов/сигнатур. Кадр SID отмечается внутриполосно некоторым кодовым словом SID, записанным в параметрах кодека как описано в приведенном ниже источнике [F-4]. Размер полезной нагрузки кадра SID составляет 14 октетов. Речь, кодированная в RTP, должна иметь битовую скорость 5600 бит/с без учета служебной информации пакетирования.

Расширенная полная скорость

Кодек GSM EFR передает кадр из 244 битов (30,5 октета) каждые 20 мс. Эта информация должна пакетироваться в заголовок RTP с четырехбитовым префиксом (0xC, или двоичный 1100), который называется "сигнатурой". Поэтому полезная нагрузка кодера GSM EFR в RTP должна содержать 31 октет. Кадр SID отмечается внутриполосно некоторым кодовым словом SID, записанным в параметрах кодека, как описано в приведенном ниже источнике [F-4]. Размер полезной нагрузки кадра SID составляет 31 октет. Речь EFR, кодированная в RTP, должна иметь битовую скорость 12 400 бит/с без учета служебной информации пакетирования.

F.5.2 Информативные библиографические ссылки

[F-1] GSM 06.10 (ETS 300 961), *Digital cellular telecommunications system; Full rate speech; Transcoding*.

[F-2] GSM 06.60 (ETS 300 726), *Digital cellular telecommunications system; Enhanced Full Rate (EFR) speech transcoding*.

[F-3] GSM 06.20 (ETS 300 969), *Digital cellular telecommunications system; Half rate speech; Half rate speech transcoding*.

[F-4] ETSI, TIPHON 03 001 (TS 101 318), *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Using GSM speech codecs within ITU-T Recommendation H.323*.

[F-5] GSM 06.31 (ETS 300 963), *Digital cellular telecommunications system; Full rate speech; Comfort noise aspect for full rate speech traffic channels*.

[F-6] GSM 06.81 (ETS 300 729), *Digital cellular telecommunications system; Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels*.

[F-7] GSM 06.41 (ETS 300 972), *Digital cellular telecommunications system; Half rate speech; Discontinuous Transmission (DTX) for half rate speech traffic channels*.

[F-8] GSM 06.12 (ETS 300 963), *Full rate speech; Comfort noise aspect for full rate speech traffic channels*.

[F-9] GSM 06.62 (ETS 300 728), *Digital cellular telecommunications system; Comfort noise aspects for Enhanced Full Rate (EFR) speech traffic channels*.

[F-10] GSM 06.22 (ETS 300 971), *Digital cellular telecommunications system; Half rate speech; Comfort noise aspect for the half Rate speech traffic channels*.

[F-11] GSM 08.60 (ETS 300 737), *Digital cellular telecommunications system; (Phase 2+) (GSM); In-band control of remote transcoders and rate adaptors for Enhanced Full Rate (EFR) and full rate traffic channels*.

F.6 Рекомендация МСЭ-Т G.722.1

Алгоритм кодирования речи, определенный в Рекомендации МСЭ-Т G.722.1, кодирует широкополосные аудиосигналы с полосой пропускания от 50 Гц до 7 кГц в одну из двух битовых скоростей, 24 кбит/с или 32 кбит/с, используя кадры 20 мс и тактовую частоту дискретизации 16 кГц. Битовая скорость может быть изменена на любой границе кадра длиной 20 мс, несмотря на то, что уведомление об изменении скорости не обеспечивается внутриполосно в этом потоке битов. При работе со скоростью 24 кбит/с вырабатываются 480 битов (60 октетов) на кадр, а при работе с 32 кбит/с вырабатываются 640 битов (80 октетов) на кадр. Следовательно, обе битовые скорости позволяют выравнивание по октетам, не требуя заполняющих битов.

Число битов в кадре постоянно. Однако внутри такого фиксированного кадра для G.722.1 используется кодирование с переменной длиной (например, кодирование Хаффмана) для представления большинства из кодированных параметров. За исключением параметра "биты управления категоризацией", все остальные параметры потока битов представляются кодами с переменной длиной, с переменным числом битов. Рисунок F.6 иллюстрирует это, показывая порядок передачи полей параметров. Все коды с переменной длиной и биты управления категоризацией передаются в порядке от самого левого бита (бита старшего разряда (MSB)) к самому правому биту (биту младшего разряда (LSB)). Использование кодирования Хаффмана означает, что невозможно определить различные параметры/поля кодера, содержащиеся в потоке битов, не выполнив вначале полное декодирование всего кадра.

На рисунке F.7 показано, как поток битов G.722.1 отображается в выровненную по октетам полезную нагрузку RTP. Поток битов кодера расщепляется на последовательность октетов (60 или 80 в зависимости от битовой скорости), а каждый октет, в свою очередь, отображается в октет RTP.

Пакет RTP должен содержать кадры G.722.1 только с одной и той же скоростью. Метка времени RTP должна выражаться в единицах "1/16 000 секунды".

Рисунок F.6/H.225.0 –Главные поля потока битов G.722.1 и порядок их передачи

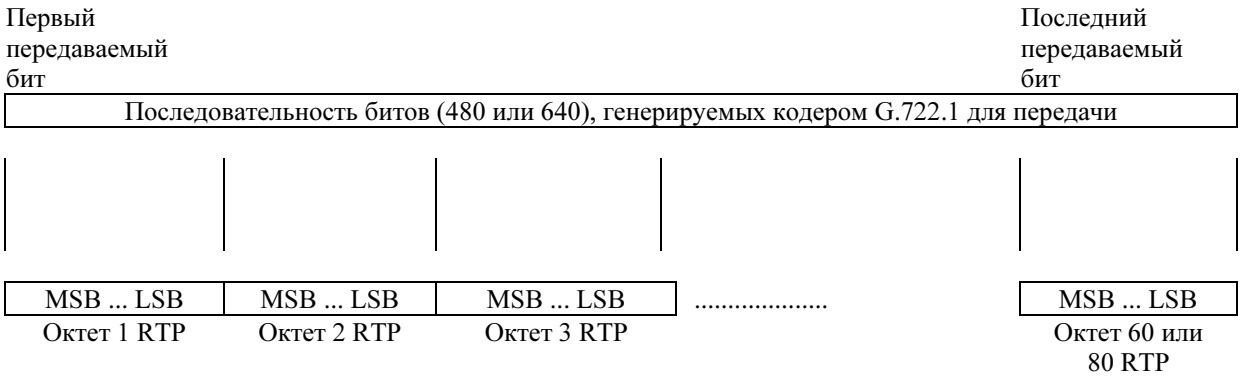


Рисунок F.7/H.225.0 – Отображение кодированного потока битов G.722.1 в RTP

F.7 ACELP стандарта TIA/EIA-136

Этот вокодер оптимизирован для цифровых сотовых систем с множественным доступом с разделением времени (TDMA) и систем персональной связи (PCS) из TIA/EIA-136. Он имеет возможности обнаружения активности голоса (VAD), замещения потерянного кадра и генерации комфортного шума (CNG). Частота дискретизации равна 8000 Гц, а длительность компрессированного голосового кадра равна 20 мс. Вокодер выдает 148-битовый массив (вектор), от s0 до s147, для каждого голосового кадра 20 мс. Бит s0 является битом старшего разряда (MSB). Дальнейшие детали см. в разделе 4 источника [F7-1].

F.7.1 Формат кадра TIA/EIA-136 ACELP

Бит флага "индикатор речи", SP, должен генерироваться вокодером и устанавливаться в "1" для указания речевого кадра или в "0" для указания кадра паузы (комфортного шума). Этот бит флага SP должен вводиться в битовую позицию 148. Битовая позиция 149 является "индикатором дефектного кадра или комфортного шума", BFI_CN, а битовая позиция 150 является флагом "обновление комфортного шума", CNU, битовая позиция 151 должна всегда устанавливаться в 0.

Логические комбинации этих трех флагов описываются ниже.

Передаваемый 152-битовый (19-октетный) кадр показан на рисунке F.8. Октеты формируются, начинаясь с LSB и продвигаясь к MSB. Первым передается LSB.

бит 0 (MSB)	**1 … 146**	**147**	**148**	**149**	**150**	**бит 151 (LSB)**
s0	s1 … s146	S147	SP	BFI_CN	CNU	Всегда 0
Речевой массив/комфортный шум			Флаг	Флаг	Флаг	Заполняющий бит

Рисунок F.8/H.225.0 – Голосовой кадр вокодера ACELP

F.7.2 Режим подавления пауз в TIA/EIA-136 ACELP

В режиме паузы вокодер генерирует представление кадра шума помещения. Этот кадр используется вокодером на приемном конце для восстановления шума помещения на передающем конце. Массив

параметров CN (комфортного шума) содержит только 38 битов, к которым добавляются три бита флагов и семь заполняющих битов (представляющих собой нули), чтобы образовать шестиоктетный кадр.

48-битовый (6-октетный) кадр CN показан на рисунке F.9. Октеты формируются, начинаясь с LSB и продвигаясь к MSB. Первым передается LSB.

бит 0 (MSB)	**1 … 37**	**38**	**39**	**40**	**41**	**41-47 (LSB)**
Cn0	cn1 … cn37	S147	SP	BFI_CN	CNU	Всегда 0
Речевой массив/Комфортный шум			Флаг	Флаг	Флаг	Заполняющий бит

Ключ:

SP Индикатор речи
BFI_CN Bad Frame Индикатор дефектного кадра/Индикатор комфортного шума
CNU Обновление комфортного шума

Логические значения этих флагов и их смысл определяются ниже:

SP: 1 = речевой кадр; 0 = неречевой (кадр комфортного шума)

BFI_CN:

Если SP = 1
и BFI_CN = 1,
то это является дефектным голосовым кадром.
В противном случае (BFI_CN = 0) это является хорошим голосовым кадром

Если SP = 0
и BFI_CN = 1,
то это является дефектным кадром комфортного шума
В противном случае (BFI_CN = 0) это является хорошим кадром комфортного шума

CN:

Если SP = 0,
BFI_CN = 0
и CN = 1,
то это является кадром обновления комфортного шума
В остальных случаях это является недействительным кадром CN

ПРИМЕЧАНИЕ. – Вокодер беспроводной подвижной свзи должен устанавливать BFI_CN в 0. Принимающая базовая станция может установить этот флаг в 1, если она не в состоянии исправить ошибки, внесенные радиоканалом.

Рисунок F.9/H.225.0 – Кадр подавления пауз для вокодера ACELP

F.7.3 Пакетирование TIA/EIA-136 ACELP

Пакетирование IS-ACELP должно соответствовать Приложению B.

1) Длительностью пакетирования должно быть целое число интервалов по 20 мс.

2) Каждый пакет может содержать один или больше кадров.

3) Кодеки будут способны кодировать и декодировать отдельные последовательно идущие кадры внутри одного пакета.

4) Все биты кодированного потока битов всегда передаются от бита младшего разряда к биту старшего разряда.

F.7.4 Библиографическая ссылка на стандарт TIA/EIA-136 ACELP

[F7-1] TIA/EIA-136, part 410, *TDMA Cellular/PCS – Radio Interface, Enhanced Full Rate Voice Codec (ACELP)*. Ранее IS-641.

F.8 US1 стандарта TIA/EIA-136

Этот вокодер оптимизирован для цифровых сотовых систем с TDMA и систем PCS из TIA/EIA-136. Источник [F8-1] содержит детальное описание этого вокодера.

F.8.1 Формат кадра TIA/EIA-136 US1

Частота дискретизации равна 8000 Гц, а длительность компрессированного голосового кадра равна 20 мс. Вокодер выдает 244 упорядоченных бита на один голосовой кадр. К речевому массиву добавляются три бита-флага BFI, SID и TAF. Один заполняющий бит (на битовой позиции 247) добавляется для образования целого числа октетов (31). Такой последний бит называется битом младшего разряда (LSB). Этот вокодер имеет также режим пауз DTX (прерывистой передачи).

Структура передаваемого голосового кадра показана на рисунке F.10.

MSB – бит 0	**1 … 243**	**244**	**245**	**246**	**247 (LSB)**
s0	s1 … s243	BFI	SID	TAF	Всегда 0
Речевой массив		Флаг	Флаг	Флаг	Заполняющий бит

Рисунок F.10/H.225.0 – Голосовой кадр вокодера US1

F.8.2 Кадры режима пауз (TX-DTX) вокодера TIA/EIA-136 US1

В режиме пауз передаются специальные кадры, называемые SID (для описателя паузы), согласно расписанию, описанному в разделе 1.3 источника [F8-1].

Кадр SID содержит такое же число битов, как и нормальные речевые кадры, но отмечается планом размещения битов. Детали см. в источнике [F8-1]. Кадр SID содержит параметры комфортного шума (CN) и 95-битовое кодовое слово SID. Кодовым словом SID являются все "0". Остальные неиспользуемые биты в 244-битовой полезной нагрузке массива также устанавливаются в "0". (См. рисунок F.11.)

MSB – бит 0	**1 … 243**	**244**	**245**	**246**	**247 (LSB)**
cn0	cn1 … cn243	BFI	SID	TAF	Всегда 0
Массив комфортного шума		Флаг	Флаг	Флаг	Заполняющий бит

Рисунок F.11/H.225.0 – Передаваемый кадр комфортного шума (для US1) от базовой станции к наземной линии

Логика флагов BFI, SID и TAF такая же, как у аналогичных флагов вокодера TIA/EIA-136 ACELP, описанных в F.7.

F.8.3 Пакетирование TIA/EIA -136 US1

Пакетирование должно соответствовать Приложению B.

1) Длительностью пакетирования должно быть целое число интервалов по 20 мс.

2) Каждый пакет может содержать нуль, один или больше кадров.

3) Кодеки будут способны кодировать и декодировать отдельные последовательно идущие кадры внутри одного пакета.

4) Все биты кодированного потока битов всегда передаются от бита младшего разряда к биту старшего разряда.

F.8.4 Библиографическая ссылка на стандарт TIA/EIA-136 US1

[F8-1] TIA/EIA-136, part 430, *TDMA Cellular/PCS – Radio Interface, US1 Full Rate Voice Codec*.

F.9 EVRC стандарта TIA/EIA IS-127

F.9.1 Описание IS-127 EVRC

F.9.1.1 Общие сведения

Усовершенствованный кодек с переменной скоростью (Enhanced Variable Rate Codec, EVRC) TIA/EIA IS-95 оптимизирован для цифровых сотовых систем с множественным доступом с кодовым разделением (CDMA) и для систем персональной связи (PCS) из TIA/EIA IS-95. Скорость дискретизации равна 8000 отсчетов в секунду, а длительность голосового кадра равна 20 мс (то есть 160 отсчетов на кадр). EVRC кодирует активную речь на полной скорости или половинной скорости, а фоновый шум (в отсутствии реи) на одной восьмой от скорости. Он доставляет речь с качеством междугородной связи на очень низкой средней битовой скорости. Детальное описание кодека EVRC можно найти в общедоступном стандарте TIA/EIA IS-127 [F9-1].

F.9.1.2 Скорости компрессии

Кодер EVRC компрессирует свой входной сигнал с использованием одной из трех скоростей: полной скорости (скорости 1), половинной скорости (скорости 1/2) и одной восьмой скорости (скорости 1/8). Полная и половинная скорости применяются в основном для кодирования активной речи, а одна восьмая скорость применяется для кодирования фонового шума (режим паузы). Все кадры имеют длительность 20 мс независимо от скорости кодирования.

F.9.1.3 Незаполненные пакеты

Чтобы обеспечить внутриполосную сигнализацию или передачу вторичного трафика (см. раздел 1.4.1 в [F9-1]), голосовые кадры не заполняются. Сгенерированный голосовой пакет просто не используется, а декодер обрабатывает его как аннулированный пакет. Детали см. в [F9-1].

F.9.1.4 Половинная скорость

Кодирование с половинной скоростью используется вместо нормальной полной скорости, когда в канал трафика нужно добавить сообщение сигнализации.

F.9.1.5 Нулевые данные канала трафика при одной восьмой скорости

Пакеты с одной восьмой скорости, в которых все биты установлены в "1", рассматриваются как нулевые данные Канала Трафика. Такие пакеты объявляются "аннулированными пакетами" и обрабатываются согласно разделу 5 из [F9-1].

Биты информации о скорости и кодировании канала добавляются к выходным битам вокодера для транспортировки по радиотракту согласно TIA/EIA IS-95.

Типы пакетов, число битов в пакете, битовые скорости непосредственно вокодера и средние скорости (вокодера плюс дополнительных битов) показаны в таблице F.3.

Таблица F.3/H.225.0 – Пакеты EVRC и битовые скорости

Тип пакета (3 бита)	Скорость	Битов в пакете	Битовая скорость вокодера, кбит/с	Средняя скорость, кбит/с
1	Полная	171	8,55	9,6
2	Половинная	80	4,0	4,8
3 (примечание)	Одна четверть (совместимость со служебной опцией-1)	40		
4	Одна восьмая	16	0,8	1,2
5	Незаполненные пакеты	0	–	–
6	Полная скорость, с ошибками	171	–	–
7	Дефектный кадр (аннулированный)	0	–	–
ПРИМЕЧАНИЕ. – Пакеты типа 3 могут генерироваться только старыми кодерами IS-96. Декодер IS-127 должен обрабатывать эти пакеты как аннулированные пакеты.				

F.9.2 Пакетирование IS-127 EVRC

F.9.2.1 Общие требования

Пакетирование для передачи должно соответствовать Приложению B.

1) Длительностью пакетирования должно быть целое число интервалов по 20 мс.

2) Передаваемый пакет может содержать нуль, один или больше кадров.

3) Кодеки будут способны кодировать и декодировать отдельные последовательно идущие кадры внутри одного пакета.

4) Все биты кодированного потока битов всегда передаются от бита младшего разряда к биту старшего разряда.

F.9.2.2 Форматы кадров

F.9.2.2.1 Полная скорость – F1

176-битовый (22-октетный) передаваемый кадр EVRC с полной скоростью (F1) показан на рисунке F.12. Октеты формируются, начинаясь с LSB и продвигаясь к MSB. Первым передается LSB (бит 175).

Бит 0 (MSB)	Биты от 1 до 170	Биты от 171 до 175 (LSB)
s0	s1 … s170	Всегда 0
Речевой массив		Заполняющие биты

Рисунок F.12/H.225.0 – Кадр EVRC с полной скоростью, F1

F.9.2.2.2 Половинная скорость – F2

80-битовый (10-октетный) передаваемый кадр EVRC с половинной скоростью (F2) показан на рисунке F.13. Октеты формируются, начинаясь с LSB и продвигаясь к MSB. Первым передается LSB (бит 79).

Бит 0 (MSB)	Биты от 1 до 79 (LSB)
s0	s1 … s79
Речевой массив	

Рисунок F.13/H.225.0 – Кадр EVRC с половинной скоростью, F2

F.9.2.2.3 Одна восьмая скорости – F3

18-битовый (2-октетный) передаваемый кадр EVRC с одной восьмой скорости (F3) показан на рисунке F.14. Октеты формируются, начинаясь с LSB и продвигаясь к MSB. Первым передается LSB (бит 15).

Бит 0 (MSB)	Биты от 1 до 15 (LSB)
s0	s1 … s15
Речевой массив	

Рисунок F.14/H.225.0 – Кадр EVRC с одной восьмой скорости, F3

F.9.3 Библиографические ссылки на стандарты IS-127 EVRC

[F9-1] TIA/EIA IS-127 (1997), *Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems*.

[F9-2] TIA/EIA IS-95-B (1999), *Mobile Station-Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems*.

F.10 Пакетирование H.223 MUX-PDU

F.10.1 Введение

Блоки данных H.223 MUX-PDU используются протоколом пакетного мультиплексирования, рассчитанным на обмен одним или несколькими информационными потоками между объектами более высокого уровня, такими как протоколы передачи данных и управленя или кодеки аудио и видео, как описано в Рекомендации МСЭ-Т H.223.

Каждый информационный поток представляется однонаправленным логическим каналом H.245, который определяется уникальным Номером логического канала (Logical Channel Number, LCN) – целым числом от 0 до 65 535. Канал с LCN=0 является постоянным логическим каналом, выделенным для канала управления H.245. Все остальные логические каналы динамически открываются и закрываются передатчиком, используя сообщения H.245 OpenLogicalChannel и CloseLogicalChannel. Все необходимые атрибуты логического канала указываются в сообщении OpenLogicalChannel message. В Рекомендации МСЭ-Т H.245 определена также процедура открытия двунаправленных логических каналов для приложений, требующих обратного канала.

Общая структура этого мультиплексора показана на рисунке 2/H.223. Мультиплексор содержит два различных уровня: уровень мультиплексирования (Multiplex Layer, MUX) и уровень адаптации (Adaptation Layer, AL).

О поддержке определенного типа полезной нагрузки H.223 сообщается с помощью наборов возможностей H.245, а также в сообщении openLogicalChannel с использованием динамических типов полезной нагрузки RTP.

F.10.2 Формат пакетирования MUX-PDU

Блок данных H.223 MUX-PDU, определенный на рисунке 3/H.223, переносится в виде данных полезной нагрузки внутри протокола RTP. Порядок передачи битов определен в 3.2.2/H.223, а правила распределения полей имеются в 3.2.3/H.223.

Хотя MUX-PDU может занимать более одного пакета RTP, MUX-PDU должен начинаться с первого октета полезной нагрузки пакета RTP.

Каждый пакет RTP содержит метку времени, получаемую от опорного генератора передатчика. Эта метка времени должна представлять заданное время передачи первого байта из H.223 MUX-PDU. Основной целью этой метки времени является оценка приемником и уменьшение любого джиггера, вносимого сетью, и воспроизведение потока битов H.223 с постоянной битовой скоростью.

Поля заголовка RTP должны использоваться следующим образом:

1) Используется какой-либо динамический тип полезной нагрузки.

2) Метка времени RTP представляет заданное время передачи первого байта из MUX-PDU в пакете, который переносится по каналу H.223 с постоянной битовой скоростью. Эта метка

времени получается из тактовой частоты со значением "по умолчанию" 90 кГц. Передатчик может изменить эту частоту, а выбранное значение будет сообщаться с помощью параметра **BitRate** в структуре **H223Capability** сообщения H.245. Если MUX-PDU занимает более одного пакета RTP, то метка времени должна быть одинаковой в последовательно идущих пакетах. Метка времени вычисляется исходя из числа байтов, включенных в переданные MUX-PDU.

3) Бит-маркер в заголовке RTP устанавливается в "1" в последнем пакете MUX-PDU, а в остальных пакетах должен быть "0". Следовательно, не нужно ждать следующего пакета, чтобы обнаружить границу MUX-PDU.

Блок данных H.223 MUX-PDU следует за заголовком RTP, как показано здесь:

Заголовок RTP	Данные MUX-PDU

Приложение G

Связь между административными доменами и внутри них

G.1 Предмет рассмотрения

Ожидается, что вся сеть H.323 будет состоять из более мелких подмножеств устройств, объединенных каким-либо способом, например, Административным доменом (областью). Ввиду потенциально большого числа элементов H.323, которые будут присутствовать в сетях H.323, необходим эффективный протокол, позволяющий организовать соединение между Административными доменами. Самым простым будет пример, когда пользователь (конечная точка) в одном Административном домене хочет вызвать пользователя (конечную точку), который обслуживается другим Административным доменом. Хотя протокол RAS H.225.0 может обращаться ко многим потребностям связи между Административными доменами, он является неполным и неэффективным для этой цели.

По той же причине нужен также некоторый эффективный протокол, который будет определен между элементами H.323 внутри одного и того же Административного домена.

В этом приложении описывается метод, позволяющий разрешать (находить) адрес, санкционировать доступ и отправлять отчет об использовании между Административными доменами и внутри них в системах H.323 с целью организации соединений. Элементы H.323, которые связываются с помощью описанных в этом приложении процедур, называются Равноправными элементами (Peer Elements). Административный домен объявляет себя для других Административных доменов с помощью некоторого типа логического элемента, который называется Пограничным элементом (Border Element). Пограничные элементы являются частными случаями равноправных элементов, причем по меньшей мере один из таких равноправных элементов принадлежит к другому Административному домену. Равноправный элемент может быть совмещен с любым другим объектом (например, с гейткипером). Приложение G не предписывает какую-либо конкретную архитектуру системы внутри Административного домена. Более того, Приложение G поддерживает использование любой модели соединения (с маршрутизацией гейткипером или с прямой связью конечных точек).

Общей процедурой для равноправных элементов является обмен информацией, относящейся к адресам, которые Административные домены могут разрешать. Обмен информацией Пограничного элемента может разрешать адреса своего Административного домена. Адреса могут определяться некоторым обычным образцом или в возрастающем конкретном порядке. Дополнительная информация разрешает элементы внутри Административного домена для определения наиболее подходящего Административного домена, который послужит пунктом назначения для определенного соединения. Пограничные элементы могут управлять доступом к объявленным ими адресам и запрашивать отчеты об использовании этих адресов во время соединений.

На рисунке G.1 показан ряд эталонных точек, представляющих сигнализацию между различными элементами в сети H.323. Административные домены на рисунке G.1 являются частью глобальной пакетной сети без границ. Заметим, что рисунок G.1 не является явным определением архитектуры системы H.323, а является средством показа эталонных точек сигнализации.

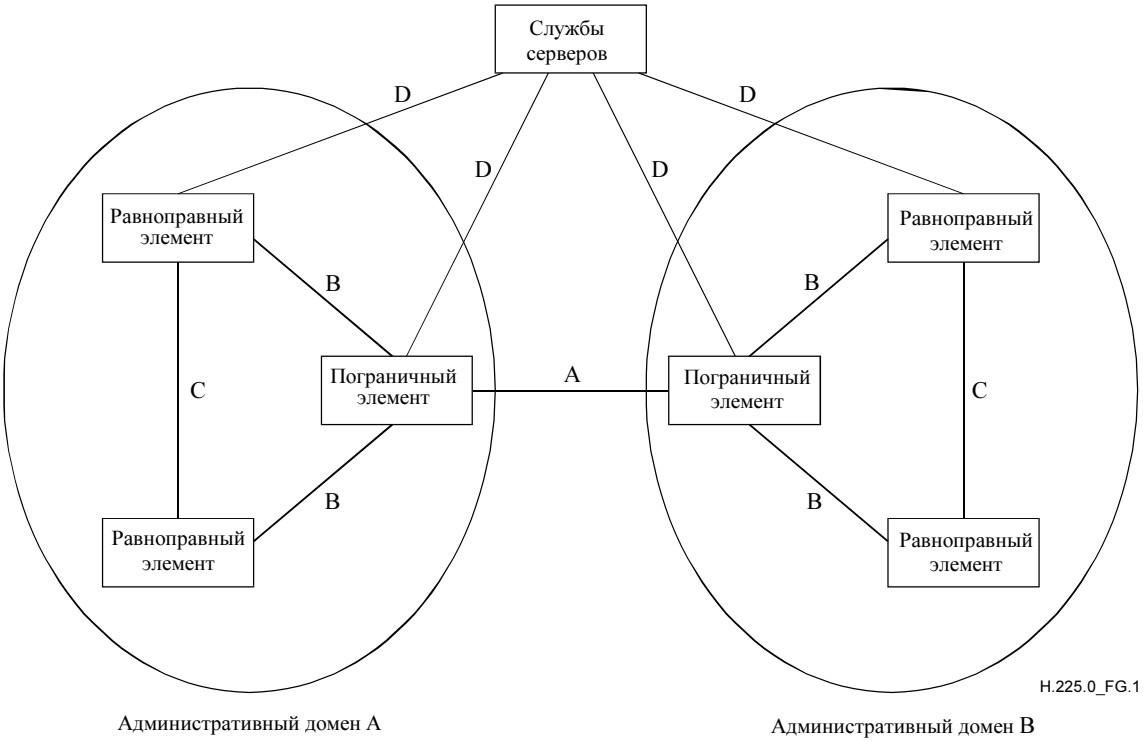


Рисунок G.1/H.225.0 – Эталонные точки системы

На этом рисунке показаны следующие эталонные точки:

A – Между Пограничными элементами, принадлежащими к разным Административным доменам.

B – Между Пограничными элементами и равноправными элементами внутри одного и того же домена.

C – Между равноправными элементами внутри одного и того же домена.

D – Между элементами H.323 и службами серверов (не входящими в предмет рассмотрения этого приложения).

Эталонные точки A, B и C являются основными предметами рассмотрения в Приложении G. Как было отмечено выше, равноправный элемент может быть совмещен с каким-либо другим элементом H.323.

В разделе G.7 (Примеры сигнализации) даются некоторые примеры сигнализации, которые могут помочь в понимании предмета рассмотрения.

G.2 Определение терминов

В этом приложении определяются следующие термины:

G.2.1 административный домен (область): Административным доменом является совокупность объектов H.323, управляемых одним администрирующим объектом. Один Административный домен может содержать один или больше гейткиперов (это значит, одну или больше зон).

G.2.2 службы серверов: Службы серверов являются функциями, такими как аутентификация или авторизация пользователя, составление счетов, биллинг, таксирование/тарификация и т. п. Службы серверов и протокол для обмена информацией со службами серверов (если он отличается от протоколов, определенных в этом приложении) не входят в предмет рассмотрения этого приложения.

G.2.3 равноправный элемент: Как определено в Рекомендации МСЭ-Т H.501, равноправным элементом является логический элемент, в котором создаются или принимаются сообщения сигнализации, определенные в настоящей Рекомендации. Этот элемент может существовать в комбинации с другими элементами H.323, например, в комбинации равноправного элемента, гейткипера и шлюза. Один Административный домен может содержать любое количество равноправных элементов.

G.2.4 пограничный элемент: Частный случай равноправного элемента, пограничный элемент является функциональным элементом, имеющим связь по меньшей мере с одним равноправным элементом, находящимся вне его Административного домена. Он обеспечивает широкий доступ в Административный домен для целей организации соединения или обеспечивает другие услуги, которые используют мультимедийную связь с другими элементами внутри Административного домена. Пограничный элемент управляет внешним видом Административного домена.

G.2.5 клиринг-центр (центр обмена информацией): Служба (возможно, в форме пограничного элемента), которая может обеспечивать разрешение (нахождение) для всех адресов (то есть некоторый тип собирающего пункта).

G.3 Сокращения

В этом приложении используются следующие сокращения:

AD Administrative Domain (Административный домен)

BE Border Element (Пограничный элемент)

CH Clearing House (Клиринг-центр, или Центр обмена информацией)

DST Daylight Saving Time (Летнее время)

EP Endpoint (Конечная точка)

GK Gatekeeper (Гейткипер)

GW Gateway (Шлюз)

PE Peer Element (Равноправный элемент)

SCN Switched Circuit Network (Сеть с коммутацией каналов)

T Terminal (Терминал)

G.4 Нормативные библиографические ссылки

Нижеследующие Рекомендации МСЭ-Т и другие источники содержат положения, которые путем 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 всем пользователям этой Рекомендации следует рассматривать возможность применения самых последних изданий перечисленных ниже Рекомендаций и других источников. Список действующих Рекомендаций МСЭ-Т регулярно публикуется. Ссылка в настоящей Рекомендации на какой-либо документ, являющийся независимым документом, не дает ему статуса Рекомендации.

[1] ITU-T Recommendation H.225.0 Version 4 (2000), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

[2] ITU-T Recommendation H.235 Version 2 (2000), *Security and encryption for H-series (H.323 and other H.245-based multimedia terminals)*.

[3] ITU-T Recommendation H.323 Version 4 (2000), *Packet-based multimedia communications systems*.

[4] ITU-T Recommendation H.323 (2000) Annex K, *HTTP-based Service Control Transport Channel in H.323*.

[5] ITU-T Recommendation H.501 (2002), *Protocol for mobility management and intra/inter-domain communication in multimedia systems*.

[6] ITU-T Recommendation H.460.2 (2001), *Number Portability Interworking between H.323 and SCN networks*.

G.5 Модели системы

Приложение G не предписывает какую-либо конкретную архитектуру системы между Административными доменами или внутри Административного домена. В последующих подразделах даются некоторые простые архитектуры, но их следует рассматривает как иллюстративные, а не полные.

Следует помнить, что равноправный элемент является функциональным элементом, который может существовать совместно с любым другим элементом H.323. На рисунке G.2 показаны некоторые примеры реализаций равноправного элемента в комбинации с другими элементами.

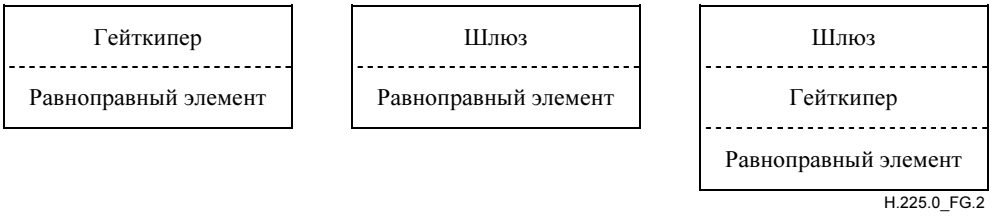


Рисунок G.2/H.225.0 – Примеры расположения равноправного элемента

Административный домен, в общем случае, считается состоящим из некоторого числа зон и некоторого числа равноправных элементов. Взаимоотношения между Административными доменами, а также между равноправными элементами внутри Административного домена могут быть любыми из множества видов организации. В последующих разделах описываются примеры взаимоотношений между Административными доменами, но иерархический, распределенный/"полная сетка" и агрегирующий примеры можно было бы использовать также для организации равноправных элементов внутри Административного домена.

Заметим снова, что последующие примеры являются иллюстративными и не исключают других возможных организаций.

G.5.1 Иерархическая организация

На рисунке G.3 показано простое иерархическое расположение Административных доменов. В таком расположении пограничный элемент одного Административного домена будет обращатья за справкой к пограничному элементу Административного домена, расположенного выше в иерархии, для разрешения адреса.

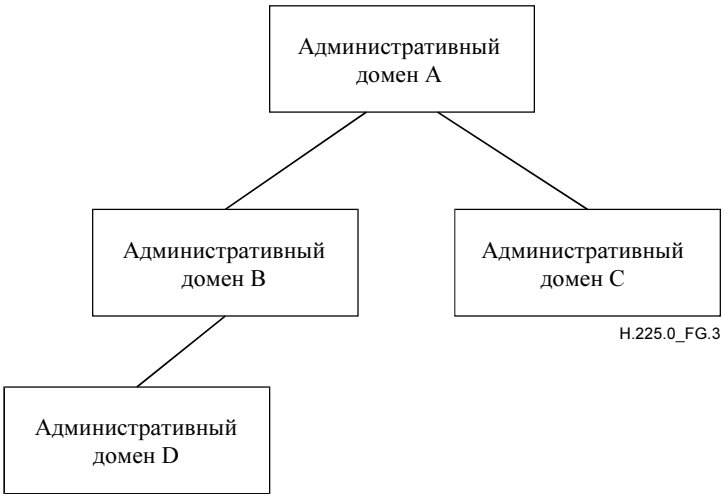


Рисунок G.3/H.225.0 – Простая иерархическая организация

G.5.2 Распределенная организация, или "полная сетка"

Возможна полностью распределенная модель, или модель "полная сетка", показанная на рисунке G.4. В этом примере пограничный элемент в каждом Административном домене связан с пограничными элементами в других известных Административных доменах.

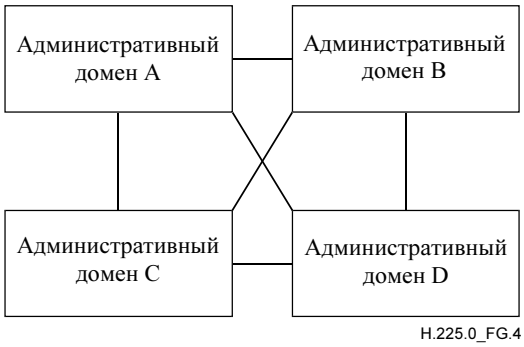


Рисунок G.4/H.225.0 – Простая распределенная организация

G.5.3 Организация с клиринг-центром

Пример расположения клиринг-центра показан на рисунке G.5. При таком расположении каждый Административный домен обращается за справкой к клиринг-центру для разрешения адресов. Заметим, что клиринг-центр является объектом, который существует вне какого-либо Административного домена, поэтому связанные с ним равноправные элементы являются, согласно определению, пограничными элементами.

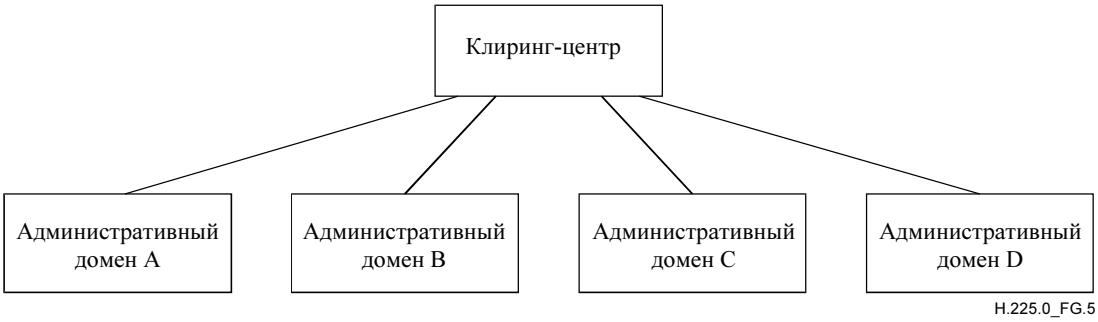


Рисунок G.5/H.225.0 – Простая организация с клиринг-центром

G.5.4 Агрегирующий пункт

На рисунке G.6 показан пример агрегирующего пункта. В этом примере Административный домен B является агрегирующим пунктом, который может обеспечивать разрешение адреса как для себя, так и для Административных доменов C и D. К примеру, Административный домен B может направить запросы на разрешение от Административного домена A к Административному домену C, либо может инструктировать Административный домен A, чтобы он связался прямо с Административным доменом C по поводу определенного пункта назначения. Если Административный домен B направляет запрос от Административного домена A к Административному домену C, то Административный домен B может записывать в память ответ Административного домена C.

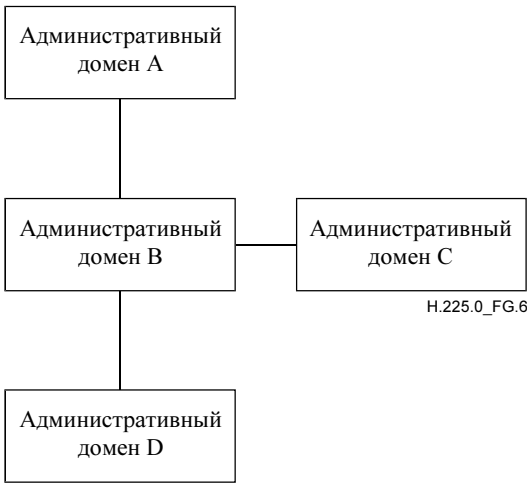


Рисунок G.6/H.225.0 – Пример агрегирующего пункта

G.5.5 Перекрывающиеся административные домены

Разрешить (найти) заданный адрес может не только один Административный домен. Например, несколько Административных доменов могут содержать шлюзы, которые могут организовать соединение к некоторому терминалу в КТСОП. За выбор подходящего Административного домена назначения несет ответственность вызывающий Административный домен. Применяемый алгоритм выбора Административного домена назначения зависит от реализатора.

G.6 Функционирование

G.6.1 Использование сообщений H.501

В реализациях Приложения G/H.225.0 должны применяться сообщения, определенные в Рекомендации МСЭ-Т H.501. Объекты, которые обмениваются сообщениями H.501, называются в настоящей Рекомендации равноправными элементами.

В Приложении G/H.225.0 используются следующие сообщения H.501:

ServiceRequest (запрос обслуживания)

ServiceConfirmation (подтверждение обслуживания)

ServiceRejection (отклонение обслуживания)

ServiceRelease (освобождение обслуживания)

DescriptorRequest (запрос описателя)

DescriptorConfirmation (подтверждение описателя)

DescriptorRejection (отклонение описателя)

DescriptorIDRequest (запрос идентификатора описателя)

DescriptorIDConfirmation (подтверждение идентификатора описателя)

DescriptorIDRejection (отклонение идентификатора описателя)

DescriptorUpdate (обновление описателя)

DescriptorUpdateAck (подтверждение обновления описателя)

AccessRequest (запрос доступа)

AccessConfirmation (подтверждение доступа)

AccessRejection (отклонение доступа)

RequestInProgress (прохождение запроса)

NonStandardRequest (запрос нестандартного)

NonStandardConfirmation (подтверждение нестандартного)

NonStandardRejection (отклонение нестандартного)

UnknownMessageResponse (ответ на неизвестное сообщение)

UsageRequest (запрос использования)

UsageConfirmation (подтверждение использования)

UsageRejection (отклонение использования)

UsageIndication (индикация использования)

UsageIndicationConfirmation (подтверждение индикации использования)

UsageIndicationRejection (отклонение индикации использования)

ValidationRequest (запрос проверки допустимости)

ValidationConfirmation (подтверждение проверки допустимости)

ValidationRejection (отклонение проверки допустимости)

Если равноправный элемент Приложения G/H.225.0 получит запросное сообщение H.501, не включенное в приведенный выше перечень, то он должен ответить сообщением UnknownMessageResponse.

Сообщения должны содержать все поля, определенные в Рекомендации МСЭ-Т H.501 как обязательные, и могут содержать факультативные поля, если требуется.

G.6.2 Шаблоны и описатели адресов

Равноправный элемент получает шаблоны следующими способами:

- статическая конфигурация;
- получение описателей (дескрипторов) от других равноправных элементов в ответ на общие запросы;
- получение ответов на специальные запросы.

G.6.2.1 Статическая конфигурация

Равноправный элемент будет поддерживать шаблоны для всех зон, за которые он несет ответственность. Эти шаблоны могут явно подготавливаться в равноправном элементе либо, в случае совмещения равноправного элемента с гейткипером, эти шаблоны могут формироваться путем суммирования информации, полученной от всех гейткиперов, с которыми этот равноправный элемент связывается. Равноправный элемент может делать эту информацию доступной для других равноправных элементов с помощью ответов на запросы. Административный домен может выбирать уровень детализации, который будет предоставлять его пограничным(и) элементом(ами). Примерами могут быть:

- Пограничный элемент, который желает скрывать внутреннюю структуру, мог бы выдавать один описатель (с указателем на передачу сообщения AccessRequest), который описывает свою целую зону и отсылает к гейткиперу, который будет обрабатывать все входящие вызовы.
- Пограничный элемент, которого не тревожит раскрытие внутренней структуры, мог бы выдавать набор шаблонов, каждый с описанием гейткипера зоны внутри домена.
- Пограничный элемент, который входит в состав брандмауэра (межсетевого средства защиты) (или является элементом, использующим модель с маршрутизацией гейткипером), мог бы выдавать шаблон о целой зоне с указанием на передачу сообщения Setup.
- Пограничный элемент, который имеет пробелы в своем домене (так как номера были переведены в другой Административный домен), выдает шаблоны с отметкой **sendAccessRequest,** указывающей пограничный элемент, который следует использовать для контакта с другим Административным доменом.
- Пограничный элемент клиринг-центра (например, имеющий полный экземпляр информации о ресурсе 44) мог бы иметь шаблон, обозначенный с помощью **sendAccessRequest,** для каждого Административного домена внутри 44.

От равноправных элементов не требуется сохранять копию полной базы данных. Если равноправный элемент не сохраняет копию полной базы данных, то ему следует иметь статически сформированные шаблоны **sendAccessRequest,** указывающие пограничный элемент клиринг-центра, который будет использоваться для удовлетворения других запросов.

G.6.2.2 Прием описателей

Равноправный элемент может запросить статически сформированные шаблоны от другого равноправного элемента. Решение об ответе на запрос принимает тот равноправный элемент, от которого запрашиваются шаблоны. Для запроса переноса равноправный элемент передает сообщение DescriptorRequest, указывающее описатели, которые он желает получить. Если владеющий равноправный элемент способен перенести эти описатели, то он отвечает сообщением DescriptorConfirmation, определяющим все шаблоны.

Запросивший равноправный элемент может записать в память копию шаблона, полученного таким способом, до истечения времени жизни шаблона; после этого равноправный элемент будет аннулировать свою копию шаблона. Если владеющий равноправный элемент изменит свои статически сформированные шаблоны до истечения их времени жизни, то он должен передать сообщение DescriptorUpdate к тем равноправным элементам, о которых он осведомлен. Получив сообщение DescriptorUpdate, равноправный элемент будет аннулировать, добавлять или изменять все указанные шаблоны в своей памяти либо запрашивать копии казанных описателей от владельца.

Промежуточный равноправный элемент (равноправный элемент между вызывающим и отвечающим Административными доменами, например, клиринг-центр или агрегирующий пункт) может публиковать свои собственные описатели, основанные на полученных им описателях. Например, клиринг-центр может указать себя в качестве контактного пункта для сообщения AccessRequest, даже если описатели, полученные им от других пограничных элементов, указывают, что контактным пунктом является другой пограничный элемент.

Равноправный элемент может в шаблоне указать требование к инициатору принять разрешение направить вызов в некоторый Административный домен. Когда в шаблоне установлен флаг **callSpecific,** а тип сообщения указывает, что должно быть передано сообщение AccessRequest, инициатор должен выдать информацию для каждого вызова в сообщении AccessRequest. Если равноправный элемент получил сообщение AccessRequest без информации для каждого вызова, а политика требует информацию для каждого вызова, то равноправный элемент должен ответить сообщением AccessRejection с причиной **needCallInformation**.

Равноправный элемент может передать сообщение DescriptorUpdate к другим известным равноправным элементам, либо равноправный элемент может передать DescriptorUpdate в режиме многопунктовой связи. Если сообщение DescriptorUpdate является многопунктовым, то равноправный элемент будет рассматривать область применения многопунктовой рассылки. Сообщение DescriptorUpdate может содержать описатели, которые были изменены. Альтернативно сообщение DescriptorUpdate может указывать только идентификаторы описателей, которые изменены, что позволяет получателю запросить эту новую информацию. Если изменено большое число описателей, то информацию следует передавать в нескольких сообщениях DescriptorUpdate, так чтобы конкретное сообщение DescriptorUpdate не превысило максимальный размер транспортного пакета.

G.6.2.3 Прием ответов на конкретные запросы

Равноправный элемент может передать сообщение AccessRequest к другому равноправному элементу, запрашивая разрешение полностью определенного или частично определенного адреса. Обычно AccessRequest передается по ненадежному транспорту (например, по UDP), хотя оно может передаваться и по надежному транспорту (например, по TCP).

Равноправный элемент, получив AccessRequest, проводит поиск в своей базе данных и отвечает наиболее конкретным шаблоном о пункте назначения. Если запросу удовлетворяют несколько шаблонов, то равноправный элемент должен выдать все согласующиеся шаблоны. Если равноправный элемент назначения фактически несет ответственность за указанный адрес-псевдоним, то этот равноправный элемент будет обычно отвечать шаблоном, указывающим, что следует передать сообщение AccessRequest или Setup. Если равноправный элемент назначения является клиринг-центром, то он будет обычно отвечать шаблоном, указывающим, что следует передать сообщение AccessRequest.

Равноправный элемент назначения может также добавить к ответу шаблоны, которые, по его мнению, будут полезны в будущем. Добавление этих шаблонов не должно делать ответ настолько большим, что транспортной сети будет необходимо фрагментировать его (например, 576 октетов для IPv4 или 1200 октетов для IPv6).

Например, пограничный элемент, который тесно связан с брандмауэром, может выдать два шаблона в своем ответе на сообщение AccessRequest: один шаблон с коротким временем жизни (порядка нескольких минут или секунд), указывающий местоположение, к которому следует передать сообщение Setup, и дополнительные шаблоны, указывающие, что сообщения AccessRequest следует передать пограничному элементу для других адресов-псевдонимов внутри Административного домена.

Равноправный элемент может хранить в памяти шаблон, полученный в сообщении AccessConfirmation, до истечения его времени жизни.

G.6.3 Обнаружение равноправного элемента или набора равноправных элементов

G.6.3.1 Статические процедуры

Равноправный элемент может иметь администрируемый набор других равноправных элементов, с которыми он может контактировать для разрешения адреса. Этот администрируемый набор может быть определен путем набора двусторонних соглашений, например, между одним Административным доменом и другими Административными доменами. Административные домены могут факультативно использовать службу клиринг-центра.

G.6.3.2 Динамические процедуры

В IP-сетях владение адресами типа Email-идентификатор определяется системой DNS (Domain Name System, система доменных имен). Таким образом, в отсутствие какой-либо лучшей информации пограничный элемент может провести поиск Записи местонахождения службы (SrRVice location record, SRV) в DNS относительно части идентификатора Email справа от символа "@" (например, поиск SRV DNS относительно **_h2250-annex-g._udp.example.org** для **person@example.org**). Ответ на этот поиск следует использовать для синтезирования шаблона **sendAccessRequest,** который может использоваться в процессе разрешения. Шаблоны, синтезированные из запросов DNS, не следует хранить в памяти дольше, чем время жизни, выданное в ответе DNS.

G.6.3.3 Иные методы

Использование иных методов определения местонахождения другого равноправного элемента остается для изучения.

G.6.4 Процедуры разрешения

G.6.4.1 Процедура разрешения внутри Административного домена

Когда равноправный элемент запрашивается о разрешении адреса-псевдонима (например, совместно расположенным шлюзом или гейткипером), он находит согласующиеся шаблоны в своей памяти.

Если согласуется более чем один шаблон, то подходящие шаблоны выбираются и сортируются согласно местной политике. Например, шаблоны могут сначала сортироваться по длине подстановочного символа (более конкретные шаблоны являются лучшими), затем сортируются по указанному типу протокола (**sendSetup** лучше, чем **sendAccessRequest**).

Если запросу удовлетворяют несколько шаблонов, то равноправный элемент должен выдавать все согласующиеся шаблоны.

Если эта процедура выбора шаблонов не дала шаблонов, отмеченных **sendSetup**, то равноправный элемент передает сообщение AccessRequest с конкретным адресом назначения к адресу, указанному в шаблоне. Когда он получил ответ от равноправного элемента, он может записать его в свою память и выдать запрашивающей стороне адрес, к которому будет передано сообщение Setup.

G.6.4.2 Процедура разрешения между Административными доменами

Когда пограничный элемент получил сообщение AccessRequest от пограничного элемента другого Административного домена, он выполняет поиск среди шаблонов в своей памяти и находит те, которые согласуются с запрошенным адресом.

Если согласуется более чем один шаблон, то согласующиеся шаблоны сначала сортируются по длине подстановочного символа (более конкретные шаблоны являются лучшими). Затем они сортируются по указанному типу протокола (**sendSetup** лучше, чем **sendAccessRequest**). В любом случае все шаблоны, не обладающие самой конкретной согласованностью, аннулируются.

Если согласующиеся шаблоны отмечены **sendAccessRequest,** то пограничный элемент может решить направить сообщение AccessRequest к пограничному(ым) элементу(ам), указанному(ым) в шаблоне(ах) либо может решить выдать шаблоны в том виде, в каком они имеются. Если счетчик

участков в принятом сообщении AccessRequest достиг нуля, то пограничный элемент не может направить это сообщение AccessRequest к другим равноправным элементам, а вместо этого выдаст любые согласующиеся шаблоны. Если счетчик участков достиг нуля, а пограничный элемент не имеет информации для включения в AccessConfirmation, то этот пограничный элемент ответит сообщением AccessRejection с указанием, что превышено число участков.

На этом этапе пограничный элемент может использовать другой пограничный элемент (например, клиринг-центр) для санкционирования запроса доступа. Чтобы сделать это, он передает сообщение ValidationRequest, которое переносит маркеры доступа, выданные запрашивающим пограничным элементом в полномочиях AccessRequest. Принимающий пограничный элемент подтверждает правильность маркеров и выдает ValidationConfirmation.

Затем пограничный элемент выдает сообщение AccessConfirmation, содержащее шаблоны, которые были им созданы (они будут иметь одни и те же поля адреса и типа сообщения), и любые другие шаблоны, которые он считает полезными.

Если запросу удовлетворяют несколько шаблонов, то пограничный элемент должен выдать все согласующиеся шаблоны.

Если запрос доступа содержит информацию о конкретном соединении, то выдаваемые шаблоны будут действительны только для запрошенного сообщения. Это применяется в случаях, когда Административный домен желает разрешать доступ для отдельного соединения. В этом случае Административный домен может требовать включать информацию о соединении в каждое сообщение AccessRequest, передаваемое к нему. Он делает это, установив некоторый флаг в шаблонах, которые ссылаются на него.

G.6.5 Обмен информацией об использовании

Равноправные элементы могут запрашивать другие равноправные элементы о выдаче им информации об использовании ресурсов в конкретных соединениях. Сообщения UsageIndication могут выдаваться на любой стадии соединения. Кроме того, для одного и того же соединения могут быть переданы несколько сообщений UsageIndication, каждое с возможной более свежей информацией, либо сообщающие о последовательных сегментах соединения или об использовании разных типов носителей.

Обмен сообщениями UsageIndication может производиться независимо от наличия у двух равноправных элементов служебных взаимоотношений между ними. Однако политика некоторого равноправного элемента может не разрешать такой обмен в отсутствие служебного взаимоотношения. В таком случае равноправный элемент может отказать сообщению UsageIndication с причиной отказа noServiceRelationship.

Запросы UsageIndication должны передаваться каждый раз, когда какой-либо равноправный элемент запросил их либо в шаблонах, в которых он служит контактным пунктом, либо путем указания в сообщении ServiceRequest, которое он передает во время установления служебного взаимоотношения с удаленным равноправным элементом, либо путем такого указания в сообщениях UsageRequest, AccessRequest, ValidationRequest и ValidationConfirmation, передаваемых в контексте соединения, для которого запрошена информация об использовании.

G.6.5.1 Несколько сообщений UsageIndications для одного и того же соединения

Несколько UsageIndications для одного и того же соединения обеспечивают все более свежую информацию об одних и тех же типах носителей или информацию об использовании новых типов носителей, появившихся в том же соединении. Кроме того, равноправные элементы могут менять соединения во время их существования, поэтому не обязательно, что все UsageIndications выдаются от одного и того же равноправного элемента. Следующие правила определяют эту семантику:

1) Сообщение UsageIndication, полученное с **`usageCallStatus`** в **`callInProgress,`** означает, что следует получить следующее UsageIndication с теми же **`callIdentifier`** и **`senderRole`**. Если получатель сконфигурирован для восстановления после неисправности, он может решить завершить после определенного интервала времени работу с дальнейшими сообщениями UsageIndication о появлении неисправности и может восстановить любые данные, какие сможет, из принятых сообщений UsageIndication.

2) Последующие сообщения UsageIndication с теми же идентификаторами **`usageField`** указывают **`startTime,`** согласующееся с **`endTime`** предыдущего сообщения (хотя это может быть невозможным для альтернативного равноправного элемента). Получатели должны

предполагать, что каждый отчет относится к другому интервалу. Другая информация в **usageField** отменяет информацию, полученную в предыдущих сообщениях с тем же идентификатором **usageField** id.

3) Равноправный элемент должен передавать новое сообщение UsageIndication для каждого изменения типа носителя в течение соединения, например, аудио оканчивается, а факс начинается, или изменился кодек. Если несколько типов носителей включены в одно и то же время (например, аудио и видео), то о них следует доложить в одном и том же сообщении UsageIndication.

G.6.5.2 Запрос и согласование информации об использовании во время установления служебного взаимоотношения

Равноправный элемент PE_A может включить элемент **UsageSpecification** в сообщение ServiceRequest ко второму равноправному элементу PE_B. Этот элемент **UsageSpecification** будет использоваться для определения безусловной ("по умолчанию") информации об использовании, которая должна сообщаться для всех соединений, происходящих в то время, когда существует служебное взаимоотношение между двумя равноправными элементами PE_A и PE_B. Эта **UsageSpecification** должна использоваться для всех соединений, для которых PE_B передает UsageIndications к PE_A.

Если элемент **UsageSpecification** прибывает в PE_B в другом сообщении от PE_A (например, в AccessConfirmation), то новая **UsageSpecification** отменяет безусловную **UsageSpecification** для всех соединений, относящихся к новому сообщению.

Равноправный элемент, получив ServiceRequest, который содержит **UsageSpecification,** будет действовать следующим образом:

i) Если принимающий равноправный элемент хочет принять ServiceRequest и содержащуюся в нем **UsageSpecification,** то он должен передать сообщение ServiceConfirmation, которое содержит такую же **UsageSpecification,** какая была принята в ServiceRequest. Эта **UsageSpecification** должна применяться как к входящим соединениям к принимающему равноправному элементу от запрашивающего равноправного элемента, так и к исходящим соединениям от принимающего равноправного элемента к запрашивающему равноправному элементу.

ii) Если принимающий равноправный элемент хочет принять ServiceRequest, но не хочет принять содержащуюся в нем **UsageSpecification,** то он должен передать либо сообщение ServiceConfirmation message, содержащее другую **UsageSpecification,** указывающую информацию об использовании, которую он способен предоставить запрашивающему равноправному элементу, либо сообщение ServiceRejection с причиной, установленной в **cannotSupportUsageSpec**.

iii) Если принимающий равноправный элемент совсем не поддерживает отчеты об использовании, то он должен выдать сообщение ServiceRejection с причиной, установленной в **usageUnavailable**.

Равноправный элемент, приняв ServiceConfirmation, будет действовать следующим образом:

i) Если **UsageSpecification** в ServiceConfirmation совпадает с переданной в ServiceRequest, то начавший равноправный элемент и принявший равноправный элемент установили служебное взаимоотношение между собой.

ii) Если **UsageSpecification** в ServiceConfirmation отличается от переданной в сообщении ServiceRequest, а начавший равноправный элемент хочет использовать новую **UsageSpecification**, то служебное взаимоотношение установлено. Если начавший равноправный элемент не хочет использовать новую **UsageSpecification**, то он должен передать сообщение ServiceRelease с причиной, установленной в **terminated**. Начинающий равноправный элемент затем может проанализировать **UsageSpecification,** выданную в ServiceConfirmation, чтобы построить новое сообщение ServiceRequest с измененной **UsageSpecification,** которая может быть приемлемой для обоих равноправных элементов.

iii) Если ServiceConfirmation не содержит **UsageSpecification** (а ServiceRequest ее содержал), то равноправный элемент, который передал ServiceConfirmation, не может или не хочет применять отчеты об использовании на уровне служебного взаимоотношения. Это, например, будет тем случаем, когда принимающий равноправный элемент реализует версию 1 этого приложения. В этом случае начавший равноправный элемент может либо закончить

служебное взаимоотношение (передав сообщение ServiceRelease с кодом причины, установленным в **`terminated`**), либо не закончить служебное взаимоотношение. В любом случае, если начинающий равноправный элемент заинтересован в получении информации об использовании для соединений, то он запросит информацию об использовании с помощью механизмов, описанных в версии 1 этого приложения (то есть передавая элементы **`UsageSpecification`** в любых сообщениях AccessRequest, AccessConfirmation (внутри выдаваемых адресных шаблонов), UsageRequest, ValidationRequest или ValidationConfirmation).

G.6.6 Передача информации о переносимости номера

В Рекомендации МСЭ-Т H.460.2 описаны механизмы переносимости номера (number portability) в сетях H.323. Поддерживая H.460.2, Приложение G должно обеспечить транспортировку информации о переносимости номера при обмене сообщениями для разрешения адреса. Интерфейс между пограничным элементом Приложения G и другими элементами сети H.323, с которыми он связывается, не рассматривается в этом приложении; предполагается, что этот интерфейс способен транспортировать переносимость номера по H.460.2 к пограничному элементу Приложения G и от него.

Когда передается сообщение AccessRequest, оно будет транспортировать информацию о переносимости номера, если она имеется, используя поле **`genericData`** в части "общая информация" этого сообщения.

Сообщения AccessConfirmation и AccessRejection будут передавать соответствующую ответную информацию о переносимости номера также в поле **`genericData.`** В случае AccessRejection причиной отказа должна быть **`genericDataReason`**.

G.7 Параметры сигнализации

Эти примеры сигнализации даются для иллюстрации основных операций. В этих примерах предполагается, что Административные домены имеют соглашения каждый с каждым, так что пограничные элементы обеспечены информацией друг о друге (например, о портах TCP). Во многих приведенных ниже примерах сообщения LRQ/LCF RAS передаются между гейткипером и пограничным элементом внутри одного и того же Административного домена. Это сделано только в иллюстративных целях; аналогичные сообщения Приложения G могут передаваться между пограничным элементом и равноправным элементом, расположенным внутри гейткипера.

G.7.1 Распределенная организация, или "полная сетка"

Пример распределенной сети показан на рисунке G.7.

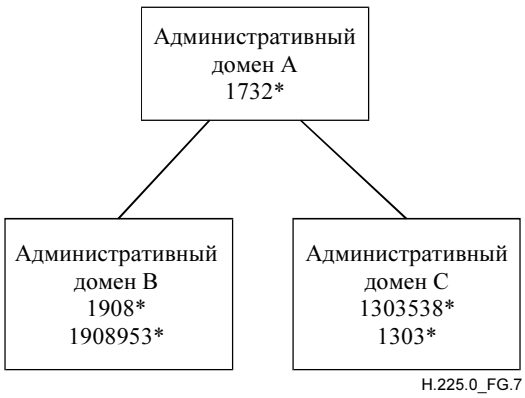


Рисунок G.7/H.225.0 – Распределенная сеть для примеров сигнализации

В этом примере предполагается, что в каждом Административном элементе имеется один пограничный элемент и что пограничные элементы сконфигурированы для разрешения адресов следующим образом:

Административный домен	Определение шаблона	Комментарий
A	Описатель "d1": Кодограмма = 1732* Транспортный адрес = адрес сигнализации о соединении для пограничного элемента A (BE_A) Тип сообщения = **sendSetup**	Сигнализация для любого соединения к Административному домену (AD) A будет проходить через пограничный элемент AD A.
B	Описатель "d2": Кодограмма = 1908* Транспортный адрес = адрес BE_B по Приложению G Тип сообщения = **sendAccessRequest** Описатель "d3": Кодограмма = 1908953* Транспортный адрес = адрес сигнализации о соединении для шлюза B1 (GW_{B1}) Тип сообщения = **sendSetup**	Для соединений 1908* необходимо сообщение AccessRequest для получения адреса сигнализации о соединении для пункта назначения (то есть шлюза). Для соединений к 1908953* Setup может быть передано прямо к этому конкретному шлюзу.
C	Описатель "d4": Кодограмма = 1303538* Транспортный адрес = адрес сигнализации о соединении для гейткипера C1 (GK_{C1}) Тип сообщения = **sendSetup** Описатель "d5": Кодограмма = 1303* Транспортный адрес = адрес BE_C по Приложению G Тип сообщения = **sendAccessRequest**	Соединения к 1303538* будут маршрутизироваться через этот конкретный гейткипер. Соединения к 1303* могут сигнализироваться прямо к шлюзу назначения, но AccessRequest должен быть послан для получения адреса сигнализации о соединении для шлюза

G.7.1.1 Обмен информацией о зоне

При распределенной организации, или "полной сетке" каждый Административный домен осведомлен о каждом другом Административном домене с помощью, вероятно, ряда двусторонних договорных соглашений. В любой момент времени пограничный элемент в каком-либо Административном домене может запросить другой Административный домен для получения адресной информации. Пример такой сигнализации приведен на рисунке G.8.

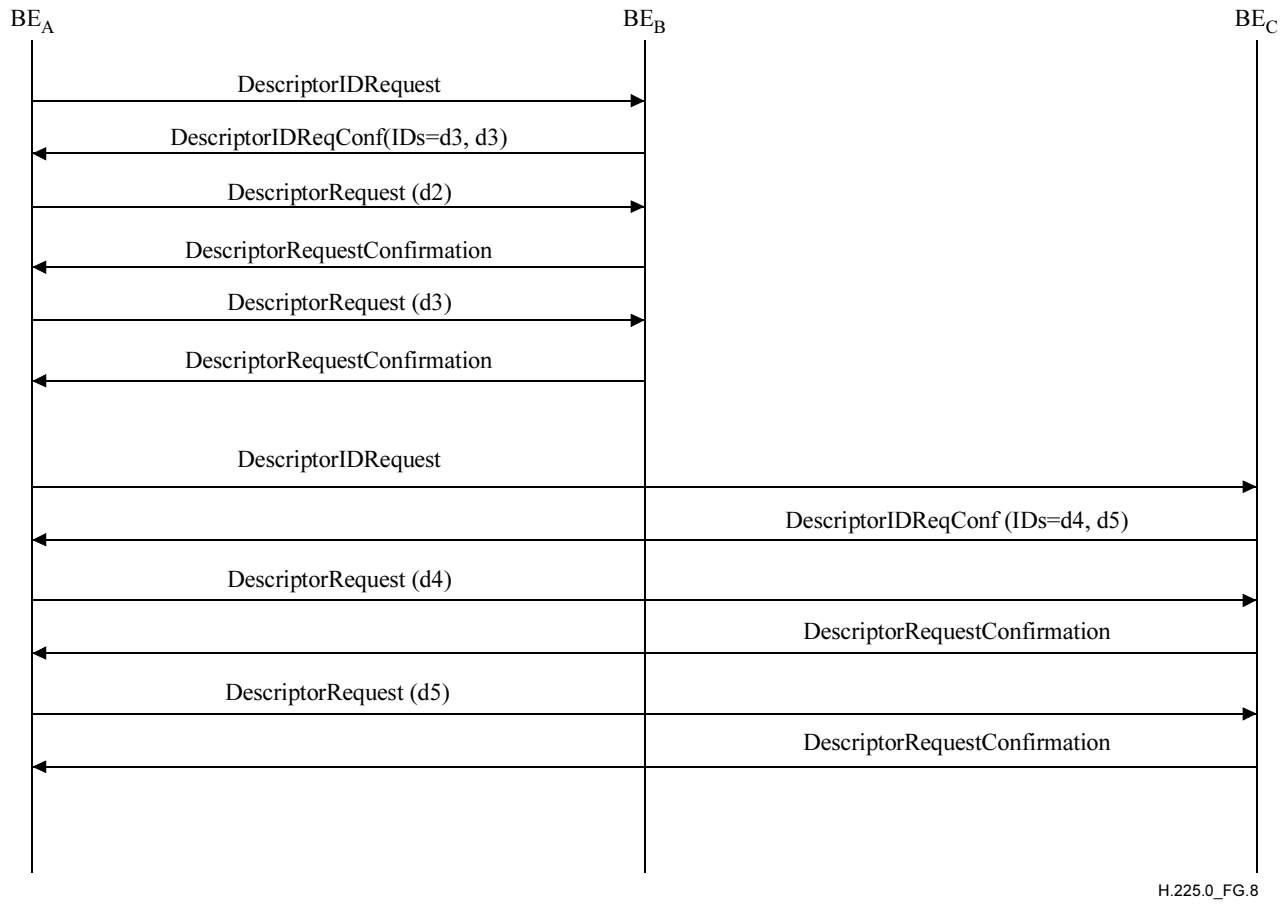


Рисунок G.8/H.225.0 – Пример обмена описателями

Аналогично BE_B запрашивает BE_A и BE_C, а BE_C запрашивает BE_A и BE_B.

G.7.1.2 Установка соединения

Предположим, что терминал T1 в Административном домене A инициирует соединение к 19085551515 (T2). Получив ARQ от T1, гейткипер этого T1 передаст LRQ. Пограничный элемент в Административном домене A, BE_A, ранее получил описатели зоны и знает, как обработать этот запрос. Как показано на рисунке G.9, BE_A передает сообщение AccessRequest к BE_B согласно описателю BE_A, полученному от BE_B. Далее BE_B ответит адресом сигнализации о соединении для T2 (в этом примере T2 может быть конечной точкой любого типа). Затем T1 передает сообщение Setup H.225.0 к адресу сигнализации о соединении для T2 после нормальных процедур, определенных в Рекомендации МСЭ-Т H.323 и ее приложениях.

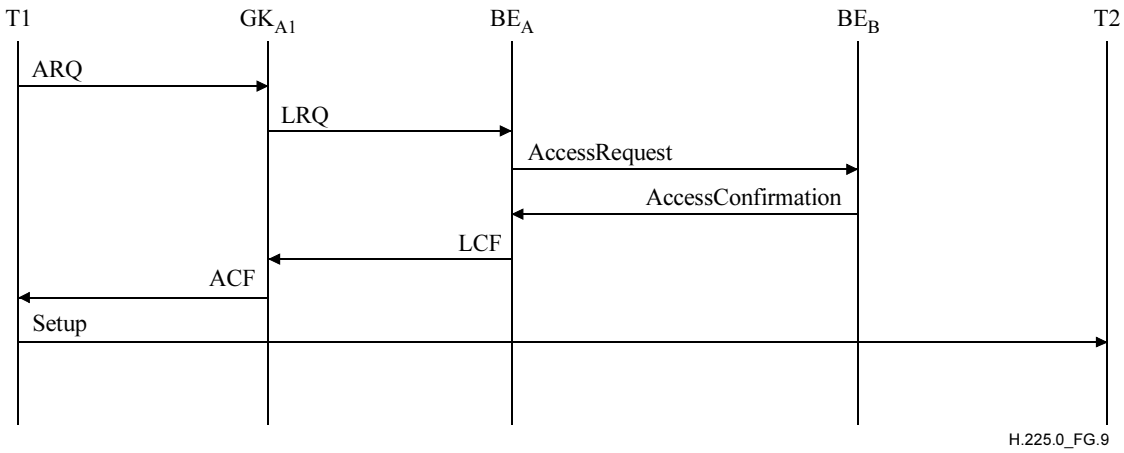


Рисунок G.9/H.225.0 – Пример простого вызова

Теперь предположим, что T1 инициирует соединение к 19089532000. В этом случае BE_A ранее получил адрес сигнализации о соединении для шлюза в Административном домене, который будет принимать это соединение. Как показано на рисунке G.10, BE_A может ответить на LRQ без какого-либо обмена сообщениями с Административным доменом B, разрешая для T1 передать сообщение Setup прямо к шлюзу.

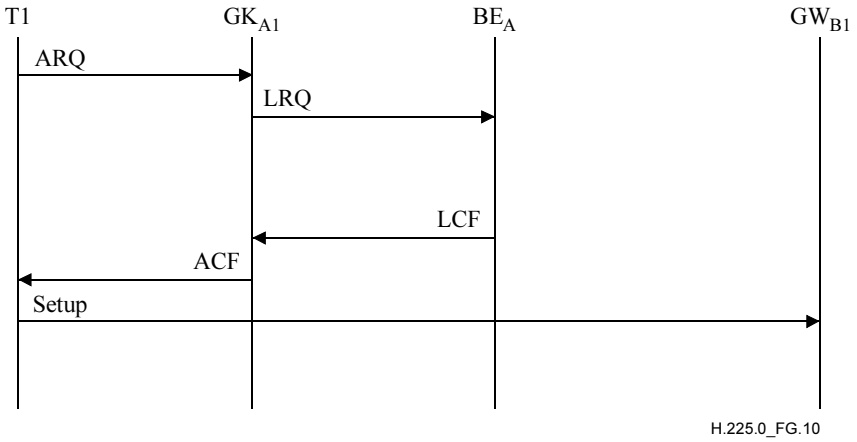


Рисунок G.10/H.225.0 – Пример соединения с адресом из памяти

В другом примере предположим, что T1 инициирует соединение к 13035382899. Административный домен C объявил о своей способности принять соединение к этому номеру и будет принимать сигнализацию о соединении через свой гейткипер при реализации модели с маршрутизацией через гейткипер. Как показано на рисунке G.11, BE_A может ответить на LRQ сообщением LCF, которое содержит адрес сигнализации о соединении для гейткипера в Административном домене C, без какого-либо обмена сообщениями с Административным доменом C.

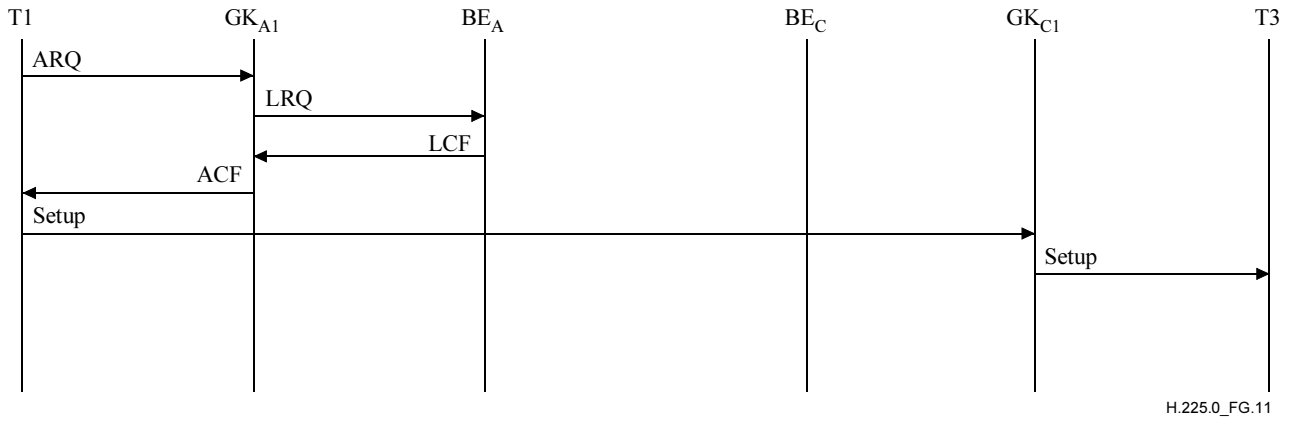


Рисунок G.11/H.225.0 – Пример соединения с маршрутизацией удаленным гейткипером

В другом случае модель с маршрутизацией через гейткипер может реализовать гейткипер терминала T1, как показано на рисунке G.12.

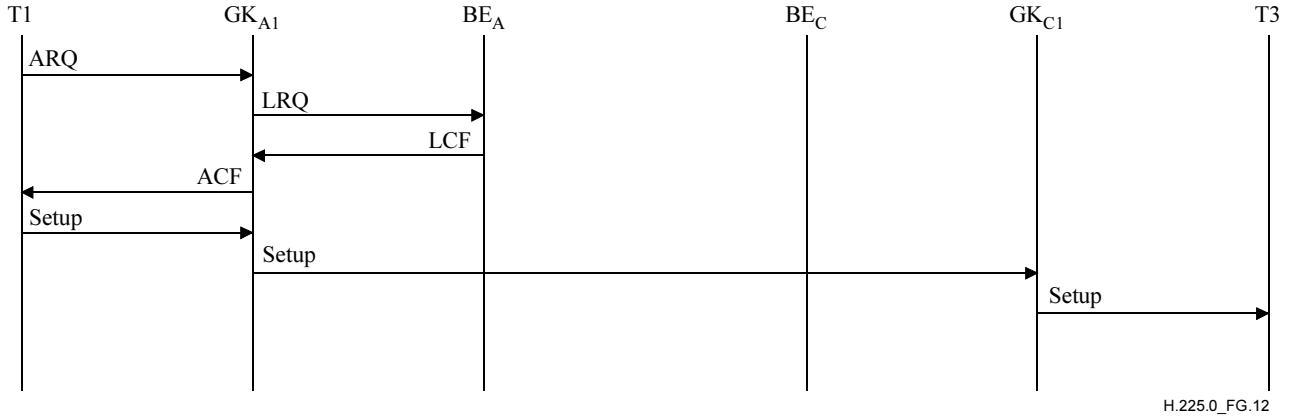


Рисунок G.12/H.225.0 – Пример соединения с маршрутизацией местным гейткипером

G.7.2 Организация с клиринг-центром

Пример конфигурации с использованием клиринг-центра показан на рисунке G.13. Будем учитывать этот рисунок в последующих примерах. В этом примере клиринг-центр хранит адресную информацию для всех Административных доменов, для которых этот клиринг-центр предоставляет обслуживание.

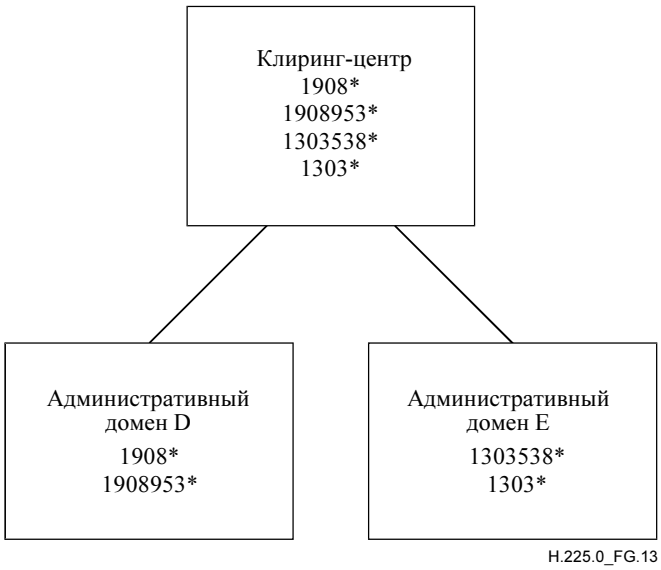


Рисунок G.13/H.225.0 – Простая конфигурация с центром обмена информацией

Пограничные элементы в Административных доменах D и E, а также клиринг-центр для этого примера содержат следующую информацию:

Административный домен	**Определение шаблона**	**Комментарий**
D	Описатель "d1": Кодограмма = 1908* Транспортный адрес = адрес BE_D по Приложению G Тип сообщения = **sendAccessRequest** Описатель "d2": Кодограмма = 1908953* Транспортный адрес = адрес сигнализации о соединении для GW_{D1} Тип сообщения = **sendSetup**	Для соединений к 1908* необходимо сообщение AccessRequest для получения адреса сигнализации о соединении для пункта назначения (то есть шлюза). Для соединений к 1908953* Setup может быть передано прямо к этому конкретному клюзу.
E	Описатель "d3": Кодограмма = 1303538* Транспортный адрес = адрес сигнализации о соединении для GK_{E1} Тип сообщения = **sendSetup** Описатель "d4": Кодограмма = 1303* Транспортный адрес = адрес BE_E по Приложению G Тип сообщения = **sendAccessRequest**	Соединения к 1303538* будут маршрутизироваться через этот конкретный гейткипер. Соединения к 1303* могут сигнализироваться прямо к шлюзу назначения, но AccessRequest должно быть послано для получения адреса сигнализации о соединении для шлюза.

Административный домен	Определение шаблона	Комментарий
CH	Описатель "d1": Описатель "d2": Кодограмма = 1908953* Транспортный адрес = адрес сигнализации о соединении для GW_{D1} Тип сообщения = **sendSetup** Описатель "d3": Кодограмма = 1303538* Транспортный адрес = адрес сигнализации о соединении для GK_{E1} Тип сообщения = **sendSetup** Описатель "d4": Кодограмма = 1303* Транспортный адрес = адрес BE_E по Приложению G Тип сообщения = **sendAccessRequest**	Клиринг-центр получает описатели от других AD и запоминает эту информацию для распространения при обмене описателями.

G.7.2.1 Обмен информацией о зонах

В этом примере клиринг-центр обменивается информацией с Административными доменами, которые подписались на обслуживание клиринг-центром. Клиринг-центр запоминает информацию, которую он получает от каждого Административного домена, и передает эту информацию ко всем другим Административным доменам. В этом примере клиринг-центр предстает перед Административным доменом D в качестве административного домена E, так что Административным доменам D и E не требуется быть осведомленными друг о друге. См. рисунок G.14.

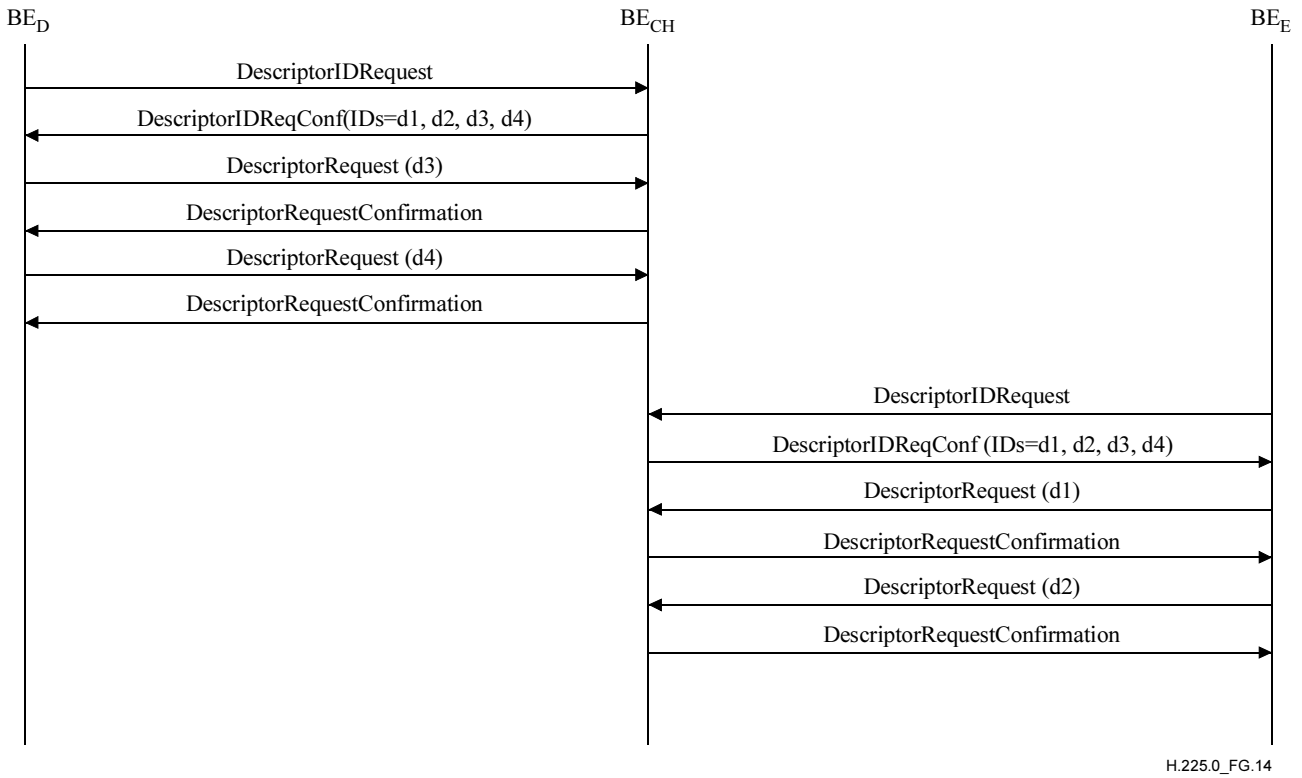


Рисунок G.14/H.225.0 – Пример обмена описателями с клиринг-центром

G.7.2.2 Установка соединения

Предположим, что T1 в Административном домене E начинает соединение к 19085551515. Пограничный элемент в Административном домене E получает от клиринг-центра описатели, которые указывают, что для такого соединения следует обращаться к клиринг-центру. Пограничный элемент передает AccessRequest в пограничный элемент клиринг-центра. Опираясь на описатели, полученные пограничным элементом клиринг-центра от пограничного элемента в Административном домене D, пограничный элемент клиринг-центра передает AccessRequest к пограничному элементу в Административном домене D. Когда пограничный элемент клиринг-центра выдаст подтверждение к пограничному элементу в Административном домене E, это подтверждение будет содержать информацию, переданную от пограничного элемента в Административном домене D. Гейткипер терминала T1 выдаст ACF с адресом destCallSignalAddress для T2, разрешая T1 передать сообщение Setup к T2. См. рисунок G.15.

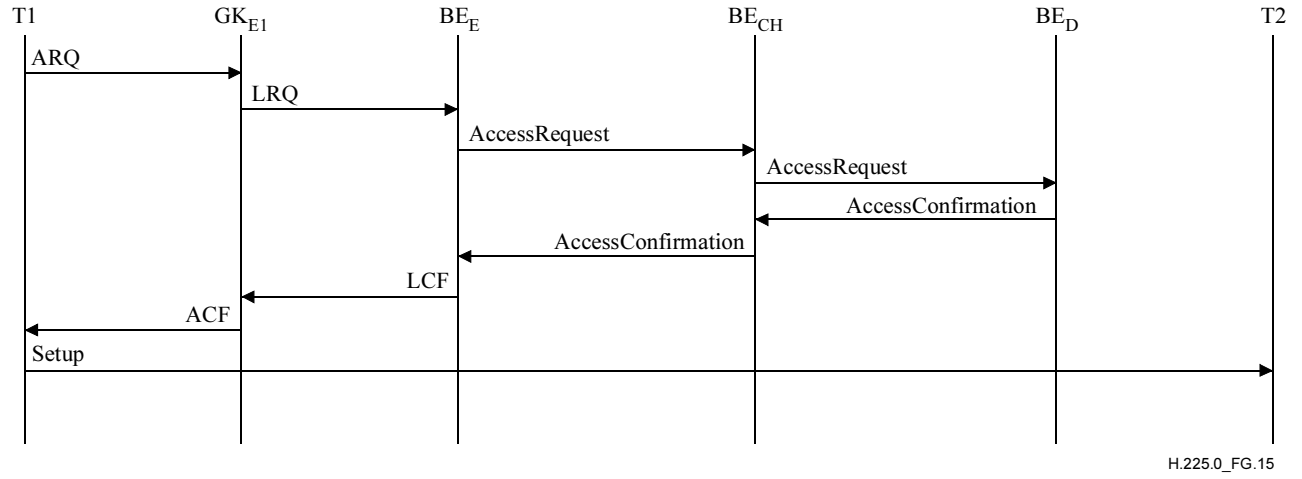


Рисунок G.15/H.225.0 – Пример соединения с участием клиринг-центра

В другом случае сигнализацию о соединении может маршрутизировать гейткипер терминала T1, как показано на рисунке G.16.

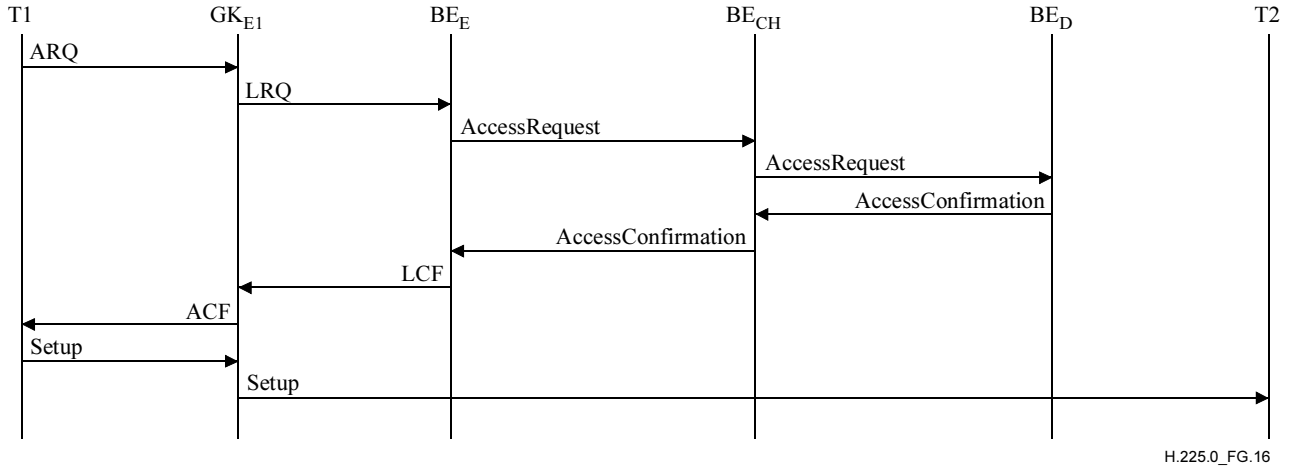


Рисунок G.16/H.225.0 – Пример соединения с участием клиринг-центра и с маршрутизацией местным гейткипером

Другой возможностью для клиринг-центра является ответ пограничному элементу в Административном домене E, содержащий контактную информацию о пограничном элементе в Административном домене D, как показано на рисунке G.17.

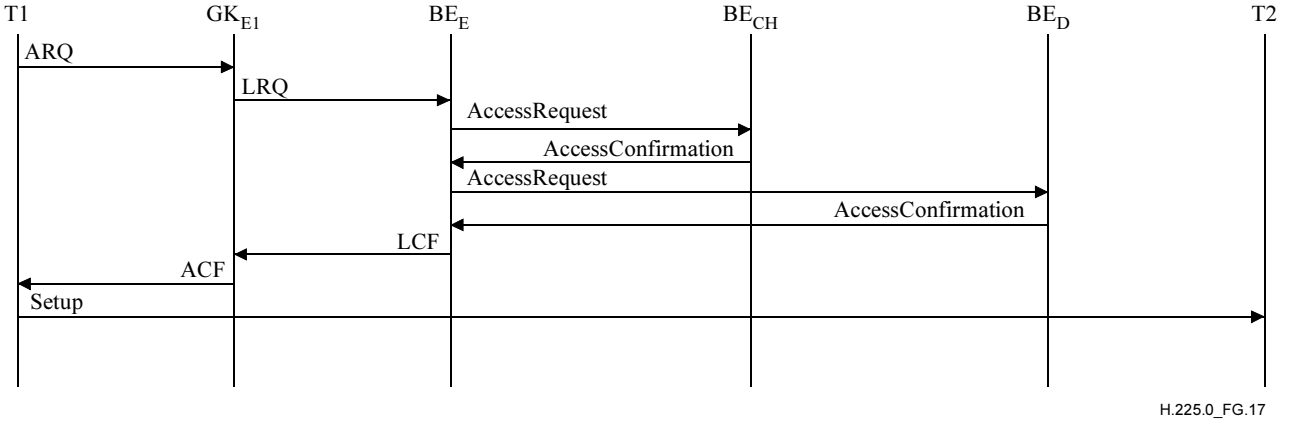


Рисунок G.17/H.225.0 – Пример маршрутизации с участием клиринг-центра и использованием контактной информации для удаленного пограничного элемента

Теперь предположим, что T1 начинает соединение к 19089532000. Ранее переданные описатели позволяют пограничному элементу выдать к T1 адрес сигнализации о соединении, не обращаясь к клиринг-центру, как показано на рисунке G.18.

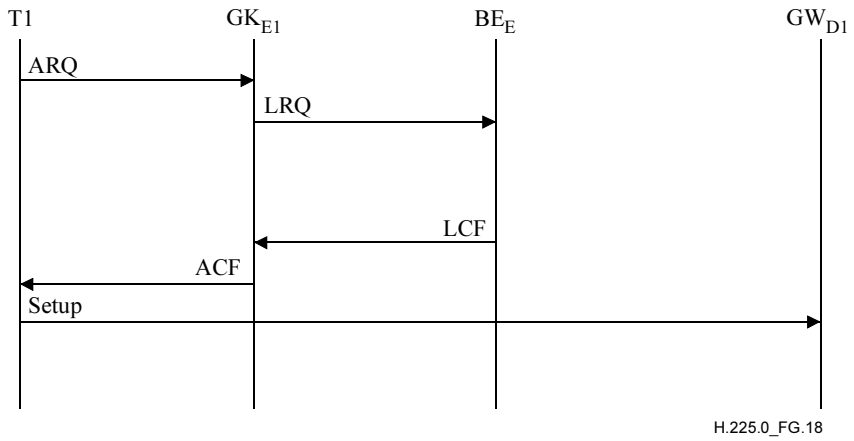


Рисунок G.18/H.225.0 – Пример соединения с использованием описателя, хранящегося в местном пограничном элементе

Далее рассмотрим сценарий, в котором T1 начинает соединение к 13035382899. Пограничный элемент в Административном домене E был ранее уведомлен, что соединения к 1303538* могут маршрутизироваться прямо к гейткиперу в Административном домене E, без потребности в сообщении AccessRequest, как показано на рисунке G.19. (Это уведомление не указывало, что объектом является гейткипер, а только указывало, что сообщение Setup может передаваться к указанному адресу.) Пограничный элемент в Административном домене D получает эту информацию от клиринг-центра, предполагая, что этот клиринг-центр в этом примере не обязан обеспечивать разрешение адреса для этих соединений.

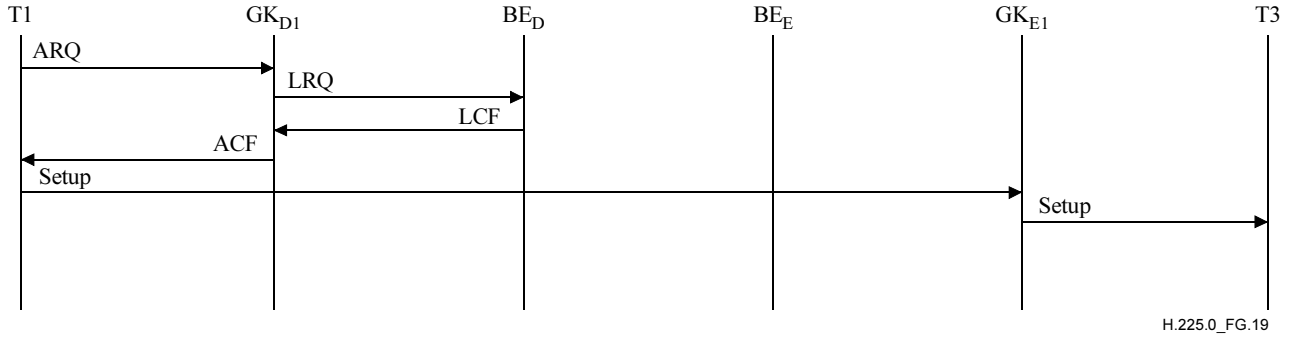


Рисунок G.19/H.225.0 – Пример соединения с маршрутизацией через гейткипер и с использованием хранящегося описателя

Вспомним, что неограниченный элемент может быть скомбинирован с гейткипером и тоже может маршрутизировать соединения в модели с маршрутизацией через гейткипер. Один из возможных примеров сигнализации показан на рисунке G.20. Возможно также использовать пограничный элемент как маршрутизирующий гейткипер в Административном домене, если описатели так конфигурированы.

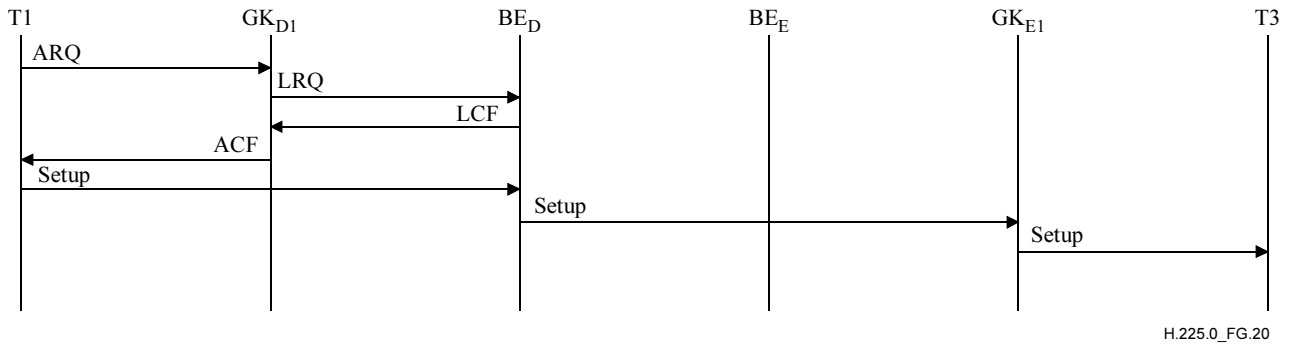


Рисунок G.20/H.225.0 – Пример соединения при комбинировании BE с маршрутизирующим GK

В примере, показанном на рисунке G.21, клиринг-центр подтверждает правильность соединения для Административного домена назначения. Клиринг-центр требует также от вызывающего и отвечающего пограничных элементов передать UsageIndication для этого соединения.

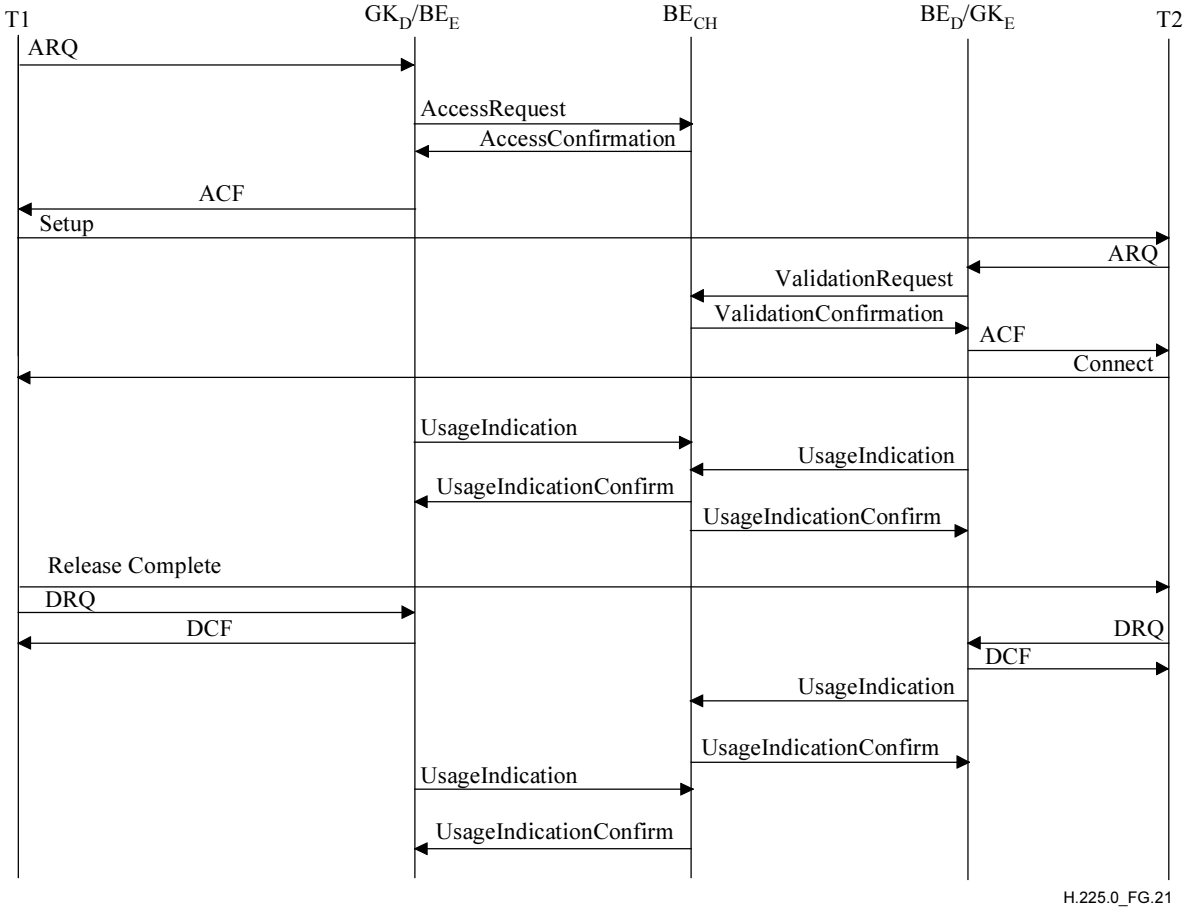


Рисунок G.21/H.225.0 – Пример с подтверждением правильности соединения и с отчетом об использовании при участии клиринг-центра

G.8 Профили Приложения G

G.8.1 Введение

В Рекомендации МЭС-Т H.501 дан богатый перечень сообщений и полей, которые могут быть использованы в Приложении G/H.225.0 для взаимодействия между Административными доменами и между равноправными элементами внутри одного Административного домена. Многие сообщения и поля являются факультативными и могут использоваться разными способами для реализации

различных услуг или опций услуг. В этом разделе описываются профили реализаций, в которых определяются сообщения, поля и процедуры, необходимые для того, чтобы заявлять о соответствии конкретному профилю.

G.8.1.1 Сигнализация о профиле и согласование профиля

Равноправный элемент может использовать общую расширяемую структуру H.323 для передачи к другому равноправному элементу набора профилей, которые ему нужны для успешной транзакции, набора профилей, которые он желает использовать, и набора профилей, которые он поддерживает. Эта сигнализация для согласования профиля может выполняться либо путем обмена отдельными сообщениями (например, путем обмена AccessRequest/AccessConfirmation), либо во время установления служебного взаимоотношения. Заметим, что установление служебного взаимоотношения между двумя равноправными элементами может не требоваться в профиле.

G.8.1.1.1 Обработка в запрашивающем объекте

Запрашивающий объект (равноправный элемент) использует элементы в структуре **`FeatureSet`** для описания различных запрашиваемых профилей. Он определяет набор профилей, нужных ему, с помощью поля **`neededFeatures`**, набор профилей, которые он желает, с помощью поля **`desiredFeatures`** и набор профилей, которые он поддерживает, в поле **`supportedFeatures`**. Все эти три поля находятся в структуре **`FeatureSet`**.

В ответ на свой запрос запрашивающий объект примет сообщение либо с подтверждением, либо с отказом.

Если запрос получает отказ, то отвечающий объект может включить набор **`neededFeatures`**, которые запрашивающий объект должен поддержать, чтобы запрос стал успешным. Если такой набор включен, а запрашивающий объект поддерживает нужные свойства (например, конкретный профиль), то запрашивающий объект может повторно выдать запрос, указывающий поддержку профиля, нужного отвечающему объекту.

Если запрос принимается, то необходимо применить специальные процедуры, обеспечивающие обратную совместимость процесса согласования. Это выполняется запрашивающим объектом, который проверит, что профиль, указанный им как нужный, имеется в ответе в списке **`supportedFeatures`**. Если запрашивающий объект не найдет профили, нужные ему, в поле **`supportedFeatures`** ответного сообщения, то он должен принять к сведению, что отвечающий объект не поддерживает профили, нужные ему. Если запрашивающий объект определит, что он не может продолжать при этих обстоятельствах, то он должен отказаться от операции, которую он пытался выполнить (то есть передать сообщение ServiceRelease, если первоначально он передал сообщение ServiceRequest), так чтобы состояние отвечающего объекта вернулось назад.

G.8.1.1.2 Обработка в отвечающем объекте

Отвечающий объект рассматривает профили, указанные в поле **`neededFeatures`** запроса, чтобы определить, может ли он принять запрос. Он рассматривает также поля **`neededFeatures`**, **`desiredFeatures`** и **`supportedFeatures`**, чтобы определить, поддерживает ли запрашивающий объект профили, нужные ему.

Если отвечающий объект определит, что нужные наборы профилей поддерживаются обоими объектами, то отвечающий объект может подтвердить запрос. Отвечающий объект перечислит набор профилей, которые он выбрал для поддержки, в поле **`supportedFeatures`** своего ответа. Если запрос принимается, то все **`neededFeatures`** из запроса должны быть включены в поле **`supportedFeatures`** ответа. Отвечающий объект может также включить **`desiredFeatures`**.

Если отвечающий объект нуждается в дополнительных профилях, которые должны поддерживаться запрашивающим объектом, то он должен отказать запросу. Если он желает объявить профили, которые должны быть поддержаны, чтобы запрос оказался успешным, то их следует указать в поле **`neededFeatures`** сообщения с отказом. Отвечающий объект может также включить в сообщение с отказом любые **`desiredFeatures`** и **`supportedFeatures`**.

G.8.1.1.3 Идентификаторы

Следующий идентификатор используется в FeatureDescriptor (Описатель свойств) для указания, что этот FeatureDescriptor применяется к профилям Приложения G/H.225.0.

Значение	Описание
idAnnexGProfiles	Этот идентификатор используется в поле "id" компонента FeatureDescriptor для указания, что этот FeatureDescriptor описывает нужные/желательные/поддерживаемые профили из Приложения G.

В следующей таблице содержится список идентификаторов, которые используются в общей расширяемой структуре, относящейся к Приложению G/H.225.0.

Стандартное целочисленное значение	Описание
0	Идентификатор в FeatureDescriptor, указывающий, что этот FeatureDescriptor описывает профили из Приложения G/H.225.0
1	Идентификатор в EnumeratedParameter, который идентифицирует Профиль "A" из Приложения G/H.225.0

G.8.2 Профиль "A": Межзоновая маршрутизация соединения к доверенному гейткиперу

Этот профиль определяет простую внутридоменную услугу: запросы для отдельного соединения, направляемые к другой доверенной зоне для определения конечной точки, когда адрес сигнализации по Приложению G для этой доверенной зоны предоставлен статически. Это – одно из простейших применений Приложения G, аналогичное пользованию LRQ RAS для запроса другой зоны о конечной точке. Тот же профиль может быть применен для запроса доверенного равноправного элемента, который выдает маршруты на основе знаний в масштабе домена или получает их с помощью последующих запросов согласно Приложению G.

G.8.2.1 Необходимые сообщения

Объекты, которые подчиняются этому профилю, должны поддерживать сообщения, отмеченные буквой M (Обязательные) в следующей таблице:

Сообщение	**Передача** **M** (Обязательное), **O** (Факультативное), **R** (Рекомендуемое)	**Прием и исполнение** **M** (Обязательное), **O** (Факультативное), **R** (Рекомендуемое)
ServiceRequest	O	M (примечание 1)
ServiceConfirmation	O	O
ServiceRejection	M	O
ServiceRelease	O	O
DescriptorRequest	O	M (примечание 1)
DescriptorConfirmation	R (примечание 2)	O
DescriptorRejection	M	O
DescriptorIdRequest	O	M (примечание 1)
DescriptorIdConfirmation	R (примечание 3)	O
DescriptorIdRejection	M	O
DescriptorUpdate	O	M (примечание 4)
DescriptorUpdateAck	M	O
AccessRequest	M	M
AccessConfirmation	M	M
AccessRejection	M	M
RequestInProgress	M	M
NonStandardRequest	O	M

Сообщение	Передача **M** (Обязательное), **O** (Факультативное), **R** (Рекомендуемое)	Прием и исполнение **M** (Обязательное), **O** (Факультативное), **R** (Рекомендуемое)
NonStandardConfirmation	O	O
NonStandardRejection	M	O
UnknownMessageResponse	M	M
UsageRequest	O	M (примечание 1)
UsageConfirmation	O	O
UsageRejection	M	O
UsageIndication	O	M (примечание 1)
UsageIndicationConfirmation	O	O
UsageIndicationRejection	M	O
ValidationRequest	O	M (примечание 1)
ValidationConfirmation	O	O
ValidationRejection	M	O
ПРИМЕЧАНИЕ 1. – Должно приниматься и, как максимум, получать отклонение. ПРИМЕЧАНИЕ 2. – Рекомендуется, чтобы объект выдавал, как минимум, один описатель для шаблона с SendAccessRequest, указывающим на себя. ПРИМЕЧАНИЕ 3. – Рекомендуется, чтобы объект выдавал, как минимум, один описатель шаблона с SendAccessRequest, указывающим на себя. ПРИМЕЧАНИЕ 4. – Должно приниматься и подтверждаться, но не требуется его обрабатывать.		

G.8.2.2 Необходимые поля

Все поля, определенные как обязательные в Рекомендации МСЭ-Т H.501, являются также обязательными в этом профиле.

Объекты, которые подчиняются этому профилю, должны также поддерживать поля, указанные в следующей таблице.

Другие поля, определенные в Рекомендации МСЭ-Т H.501 как факультативные, могут факультативно присутствовать.

Сообщение или структура	Необходимое поле	Комментарий
Сообщение AccessRequest	**destinationInfo**	Один адрес, содержащий полностью определенный адрес E.164 для пункта назначения
	sourceInfo	Содержит domainInfo и endpointType
	callInfo	
Сообщение AccessConfirmation	**templates**	Если какие-либо шаблоны присутствуют, то имеется по одному шаблону для каждого отвечающего шлюза/гейткипера
	partialResponse	Установлено в ЛОЖЬ
Структура AddressTemplate	**pattern**	Присутствует одна конкретная кодограмма, содержащая номер E.164
	routeInfo	Присутствует один экземпляр
	timeToLive	

Сообщение или структура	Необходимое поле	Комментарий
Структура RouteInformation	**messageType**	Присутствует
	callSpecific	Установлено в ЛОЖЬ
	contacts	Присутствует один экземпляр
	type	Должно присутствовать, если messageType = **sendSetup**
Структура ContactInformation	**transportAddress**	IP-адрес шлюза/гейткипера
	priority	

G.8.2.3 Необходимые процедуры

В этом профиле объекты могут использовать статические процедуры обнаружения из Приложения G (см. G.6.3.1) и поэтому будут иметь сконфигурированный список равноправных элементов или гейткиперов, к которым можно посылать запросы. Этот список может содержать альтернативы, которые следует использовать только в случаях, когда исходный элемент не может быть достигнут, либо можно просто добавлять альтернативы (если они имеются) к этому списку.

Объекты могут также использовать динамические процедуры обнаружения из Приложения G (см. G.6.3.2).

Объекты должны посылать сообщение AccessRequest к выбранному равноправному элементу или гейткиперу для каждого соединения. Если для заданного соединения доступны при запросе более чем один равноправный элемент или гейткипер, то не указывается, как они должны или могут запрашиваться – последовательно или одновременно. Этот выбор остается за запрашивающим объектом.

Ответ будет иметь нуль или больше шаблонов. Время жизни (**timeToLive**) может быть установлено в 60 секунд или менее для указания, что он не может использоваться для другого соединения.

Чтобы улучшить взаимодействие с более общими равноправными элементами, предлагается, чтобы в случае, когда равноправный элемент не реализует поддержку описателей, он выполнял следующие процедуры:

- Если получено сообщение DescriptorIDRequest, то равноправный элемент выдаст сообщение DescriptorIDConfirmation, содержащее одиночный **DescriptorInfo**. Этот **DescriptorInfo** определяет описатель, содержащий один шаблон, который определяет **sendAccessRequest**, указывающий на сам равноправный элемент.
- Если получено сообщение DescriptorRequest, то равноправный элемент выдаст сообщение DescriptorConfirmation, содержащее одиночный описатель. Этот описатель должен содержать один шаблон, который определяет **sendAccessRequest**, указывающий на сам равноправный элемент.

G.8.2.4 Идентификаторы для профиля "A"

Следующий идентификатор используется в **EnumeratedParameter** для указания, что этот **EnumeratedParameter** определяет профиль A из Приложения G/H.225.

Значение	Описание
idAnnexGProfileA	Этот идентификатор используется в поле "id" компонента **EnumeratedParameter** для указания, что профиль A из Приложения G нужен/желателен/поддерживается. Заметим, что поле "content" не присутствует в **EnumeratedParameter**.

Приложение H

Синтаксис сообщений H.225.0 (на языке ASN.1)

Эта Рекомендация определяет протоколы для RAS (в основном, протокол гейткипера) и сигнализацию о соединении (в основном, протокольные блоки данных, которые расположены в информационном элементе Пользователь-пользователь). Эти протоколы вместе определяются в следующем дереве ASN.1. Определения семантики для сообщений и различных элементов были изложены в предыдущих работах.

```
H323-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN


IMPORTS
          SIGNED{},
          ENCRYPTED{},
          HASHED{},
          ChallengeString,
          TimeStamp,
          RandomVal,
          Password,
          EncodedPwdCertToken,
          ClearToken,
          CryptoToken,
          AuthenticationMechanism
     FROM H235-SECURITY-MESSAGES
          DataProtocolCapability,
          T38FaxProfile
     FROM MULTIMEDIA-SYSTEM-CONTROL;
H323-UserInformation ::= SEQUENCE     -- корень для всех сообщений сигнализации о
                                      -- соединении H.225.0
{
     h323-uu-pdu   H323-UU-PDU,
     user-data     SEQUENCE
     {
          protocol-discriminator      INTEGER   (0..255),
          user-information            OCTET STRING (SIZE(1..131)),
          ...
     } OPTIONAL,
     ...
}

H323-UU-PDU ::= SEQUENCE
{
     h323-message-body   CHOICE
     {
          setup               Setup-UUIE,
          callProceeding      CallProceeding-UUIE,
          connect             Connect-UUIE,
          alerting            Alerting-UUIE,
          information         Information-UUIE,
          releaseComplete     ReleaseComplete-UUIE,
          facility            Facility-UUIE,
          ...,
          progress       Progress-UUIE,
          empty          NULL,     -- используется, когда передано сообщение
                                   -- Facility, но Facility-UUIE не должен
                                   -- вызываться (возможен, когда
                                   -- транспортируются сообщения дополнительных
                                   -- услуг в версиях H.225.0 до версии 4)
```

```
        status              Status-UUIE,
        statusInquiry       StatusInquiry-UUIE,
        setupAcknowledge    SetupAcknowledge-UUIE,
        notify              Notify-UUIE
    },
    nonStandardData         NonStandardParameter OPTIONAL,
    ...,
    h4501SupplementaryService    SEQUENCE OF OCTET STRING OPTIONAL,
                            -- каждая последовательность цепочки октетов
                            -- определена как APDU H4501SupplementaryService,
                            -- определенный в таблице 3/H.450.1

    h245Tunnelling          BOOLEAN,
                            -- если ИСТИНА, то разрешено туннелирование
                            -- сообщений H.245
    h245Control             SEQUENCE OF OCTET STRING OPTIONAL,
    nonStandardControl      SEQUENCE OF NonStandardParameter OPTIONAL,
    callLinkage             CallLinkage OPTIONAL,
    tunnelledSignallingMessage   SEQUENCE
    {
        tunnelledProtocolID     TunnelledProtocol, -- идентификатор
                                                   туннелированного протокола
                                                   сигнализации
    messageContent              SEQUENCE OF OCTET STRING, -- последовательность
                                                          -- целого(ых)
                                                          -- сообщения(ий)
        tunnellingRequired      NULL OPTIONAL,
        nonStandardData         NonStandardParameter OPTIONAL,
        ...
    } OPTIONAL,
    provisionalRespToH245Tunnelling  NULL OPTIONAL,
    stimulusControl                  StimulusControl OPTIONAL,
    genericData                      SEQUENCE OF GenericData OPTIONAL
}

StimulusControl ::= SEQUENCE
{
    nonStandard                      NonStandardParameter OPTIONAL,
    isText                           NULL OPTIONAL,
    h248Message                      OCTET STRING OPTIONAL,
    ...
}

Alerting-UUIE ::= SEQUENCE
{
    protocolIdentifier          ProtocolIdentifier,
    destinationInfo             EndpointType,
    h245Address                 TransportAddress OPTIONAL,
    ...,
    callIdentifier              CallIdentifier,
    h245SecurityMode            H245Security OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
    multipleCalls               BOOLEAN,
    maintainConnection          BOOLEAN,
    alertingAddress             SEQUENCE OF AliasAddress OPTIONAL,
    presentationIndicator       PresentationIndicator OPTIONAL,
    screeningIndicator          ScreeningIndicator OPTIONAL,
    fastConnectRefused          NULL OPTIONAL,
    serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
    capacity                    CallCapacity OPTIONAL,
    featureSet                  FeatureSet OPTIONAL
}
```

```
CallProceeding-UUIE ::= SEQUENCE
{
     protocolIdentifier          ProtocolIdentifier,
     destinationInfo             EndpointType,
     h245Address                 TransportAddress OPTIONAL,
     ...,
     callIdentifier              CallIdentifier,
     h245SecurityMode            H245Security OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
     multipleCalls               BOOLEAN,
     maintainConnection          BOOLEAN,
     fastConnectRefused          NULL OPTIONAL,
     featureSet                  FeatureSet OPTIONAL
}

Connect-UUIE ::= SEQUENCE
{
     protocolIdentifier          ProtocolIdentifier,
     h245Address                 TransportAddress OPTIONAL,
     destinationInfo             EndpointType,
     conferenceID                ConferenceIdentifier,
     ...,
     callIdentifier              CallIdentifier,
     h245SecurityMode            H245Security OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
     multipleCalls               BOOLEAN,
     maintainConnection          BOOLEAN,
     language                    SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
                                 -- маркер языка из RFC 1766
     connectedAddress            SEQUENCE OF AliasAddress OPTIONAL,
     presentationIndicator       PresentationIndicator OPTIONAL,
     screeningIndicator          ScreeningIndicator OPTIONAL,
     fastConnectRefused          NULL OPTIONAL,
     serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
     capacity                    CallCapacity OPTIONAL,
     featureSet                  FeatureSet OPTIONAL
}

Information-UUIE ::=SEQUENCE
{
     protocolIdentifier          ProtocolIdentifier,
     ...,
     callIdentifier              CallIdentifier,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
     fastConnectRefused          NULL OPTIONAL,
     circuitInfo                 CircuitInfo OPTIONAL
}

ReleaseComplete-UUIE ::= SEQUENCE
{
     protocolIdentifier          ProtocolIdentifier,
     reason                      ReleaseCompleteReason OPTIONAL,
     ...,
     callIdentifier              CallIdentifier,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     busyAddress                 SEQUENCE OF AliasAddress OPTIONAL,
     presentationIndicator       PresentationIndicator OPTIONAL,
```

```
    screeningIndicator          ScreeningIndicator OPTIONAL,
    capacity                    CallCapacity OPTIONAL,
    serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
    featureSet                  FeatureSet OPTIONAL
}

ReleaseCompleteReason ::= CHOICE
{
    noBandwidth                 NULL, -- полоса пропускания убрана или ARQ
                                      -- было отклонено
    gatekeeperResources         NULL, -- исчерпаны
    unreachableDestination      NULL, -- нет транспортного пути к пункту
                                      -- назначения
    destinationRejection        NULL, -- отклонение в пункте назначения
    invalidRevision             NULL,
    noPermission                NULL, -- отклонение от гейткипера вызываемой
                                      -- стороны
    unreachableGatekeeper       NULL, -- терминал не может связаться с
                                      -- гейткипером для ARQ
    gatewayResources            NULL,
    badFormatAddress            NULL,
    adaptiveBusy                NULL, -- соединение сброшено из-за загрузки LAN
    inConf                      NULL, -- вызываемая сторона занята
    undefinedReason             NULL,
    ...,
    facilityCallDeflection      NULL, -- вызов был отклонен с помощью сообщения
                                      -- Facility (возможности)
    securityDenied              NULL, -- несовместимые установки безопасности
    calledPartyNotRegistered    NULL, -- используется гейткипером, когда
                                      -- конечная точка имеет preGrantedARQ для
                                      -- обхода ARQ/ACF
    callerNotRegistered         NULL, -- используется гейткипером, когда
                                      -- конечная точка имеет preGrantedARQ для
                                      -- обхода ARQ/ACF
    newConnectionNeeded         NULL, -- указывает, что  Setup не было принято
                                      -- по этому соединению, но Setup может
                                      -- быть принято по новому сообщению
    nonStandardReason           NonStandardParameter,
    replaceWithConferenceInvite ConferenceIdentifier, -- соединение сброшено
                                                      -- из-за последующего
                                                      -- приглашения к
                                                      -- конференции
                                                      -- (см. 8.4.3.8/H.323)
    genericDataReason           NULL,
    neededFeatureNotSupported   NULL,
    tunnelledSignallingRejected NULL,
    invalidCID                  NULL,
    securityError               SecurityErrors,
    hopCountExceeded            NULL
}

Setup-UUIE ::= SEQUENCE
{
    protocolIdentifier      ProtocolIdentifier,
    h245Address             TransportAddress OPTIONAL,
    sourceAddress           SEQUENCE OF AliasAddress OPTIONAL,
    sourceInfo              EndpointType,
    destinationAddress      SEQUENCE OF AliasAddress OPTIONAL,
    destCallSignalAddress   TransportAddress OPTIONAL,
    destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCRV            SEQUENCE OF CallReferenceValue OPTIONAL,
    activeMC                BOOLEAN,
    conferenceID            ConferenceIdentifier,
    conferenceGoal          CHOICE
```

```
    {
        create              NULL,
        join                NULL,
        invite              NULL,
        ...,
        capability-negotiation                NULL,
        callIndependentSupplementaryService   NULL
    },
    callServices                QseriesOptions OPTIONAL,
    callType                    CallType,
    ...,
    sourceCallSignalAddress     TransportAddress OPTIONAL,
    remoteExtensionAddress      AliasAddress OPTIONAL,
    callIdentifier              CallIdentifier,
    h245SecurityCapability      SEQUENCE OF H245Security OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
    mediaWaitForConnect         BOOLEAN,
    canOverlapSend              BOOLEAN,
    endpointIdentifier          EndpointIdentifier OPTIONAL,
    multipleCalls               BOOLEAN,
    maintainConnection          BOOLEAN,
    connectionParameters        SEQUENCE  -- дополнительные параметры шлюза
    {
        connectionType              ScnConnectionType,
        numberOfScnConnections      INTEGER (0..65535),
        connectionAggregation       ScnConnectionAggregation,
        ...
    } OPTIONAL,
    language                    SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
                                -- маркер языка из RFC1766
    presentationIndicator       PresentationIndicator OPTIONAL,
    screeningIndicator          ScreeningIndicator OPTIONAL,
    serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
    symmetricOperationRequired  NULL OPTIONAL,
    capacity                    CallCapacity OPTIONAL,
    circuitInfo                 CircuitInfo OPTIONAL,
    desiredProtocols            SEQUENCE OF SupportedProtocols OPTIONAL,
    neededFeatures              SEQUENCE OF FeatureDescriptor OPTIONAL,
    desiredFeatures             SEQUENCE OF FeatureDescriptor OPTIONAL,
    supportedFeatures           SEQUENCE OF FeatureDescriptor OPTIONAL,
    parallelH245Control         SEQUENCE OF OCTET STRING OPTIONAL,
    additionalSourceAddresses   SEQUENCE OF ExtendedAliasAddress OPTIONAL,
    hopCount                    INTEGER (1..31) OPTIONAL
}

ScnConnectionType ::= CHOICE
{
    unknown     NULL,  -- следует выбрать, когда тип соединения неизвестен
    bChannel    NULL,  -- каждое отдельное соединение в SCN имеет 64 кбит/с.
                       -- Заметим, что если SCN доставляет полезные данные на
                       -- 56 кбит/с, то фактическая полоса пропускания,
                       -- выделенная в SCN, равна тем не менее 64 кбит/с.
    hybrid2x64  NULL,  -- каждое соединение является гибридным соединением
                       -- 128 кбит/с
    hybrid384   NULL,  -- каждое соединение является гибридным соединением H0
                       -- (384 кбит/с)
    hybrid1536  NULL,  -- каждое соединение является гибридным соединением H11
                       -- (1536 кбит/с)
    hybrid1920  NULL,  -- каждое соединение является гибридным соединением H12
                       -- (1920 кбит/с)
```

```
    multirate    NULL,  -- полоса пропускания, предоставленная в SCN с
                        -- использованием многоскоростного режима.
                        -- В этом случае октет "скорость переноса информации"
                        -- в возможности переноса должен быть установлен
                        -- в "многоскоростной", а октет "множитель скорости"
                        -- должен означать число каналов B.
    ...
}

ScnConnectionAggregation ::= CHOICE
{
    auto           NULL,     -- механизм объединения неизвестен
    none           NULL,     -- соединение, образованное с использованием
                             -- одиночного соединения SCN
    h221           NULL,     -- используйте для объединения соединений
                             -- формирования кадра по H.221
    bonded-mode1   NULL,     -- используйте режим 1 связывания по ISO/IEC 13871.
                             -- Используйте связанный режим 1 для передачи
                             -- связанного соединения, если точный режим
                             -- связывания, подлежащий применению, неизвестен.
    bonded-mode2   NULL,     -- используйте режим 2 связывания по ISO/IEC 13871
    bonded-mode3   NULL,     -- используйте режим 3 связывания по ISO/IEC 13871
    ...
}

PresentationIndicator ::= CHOICE
{
    presentationAllowed          NULL,
    presentationRestricted       NULL,
    addressNotAvailable          NULL,
    ...
}

ScreeningIndicator ::= ENUMERATED
{
    userProvidedNotScreened (0),
        -- номер был предоставлен удаленным пользователем и не был проверен
        -- гейткипером
    userProvidedVerifiedAndPassed (1),
        -- номер был предоставлен устройством пользователя (или удаленной
        -- сетью) и был проверен гейткипером
    userProvidedVerifiedAndFailed (2),
        -- номер был предоставлен устройством пользователя (или удаленной
        -- сетью), а гейткипер определил, что эта информация неправильна
    networkProvided (3),
        -- номер был предоставлен гейткипером
    ...
}

Facility-UUIE ::= SEQUENCE
{
    protocolIdentifier           ProtocolIdentifier,
    alternativeAddress           TransportAddress OPTIONAL,
    alternativeAliasAddress      SEQUENCE OF AliasAddress OPTIONAL,
    conferenceID                 ConferenceIdentifier OPTIONAL,
    reason                       FacilityReason,
    ...,
    callIdentifier               CallIdentifier,
    destExtraCallInfo            SEQUENCE OF AliasAddress OPTIONAL,
    remoteExtensionAddress       AliasAddress OPTIONAL,
    tokens                       SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                 SEQUENCE OF CryptoH323Token OPTIONAL,
    conferences                  SEQUENCE OF ConferenceList OPTIONAL,
    h245Address                  TransportAddress OPTIONAL,
    fastStart                    SEQUENCE OF OCTET STRING OPTIONAL,
```

```
    multipleCalls               BOOLEAN,
    maintainConnection          BOOLEAN,
    fastConnectRefused          NULL OPTIONAL,
    serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
    circuitInfo                 CircuitInfo OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    destinationInfo             EndpointType OPTIONAL,
    h245SecurityMode            H245Security OPTIONAL
}

ConferenceList ::= SEQUENCE
{
    conferenceID                ConferenceIdentifier OPTIONAL,
    conferenceAlias             AliasAddress OPTIONAL,
    nonStandardData             NonStandardParameter OPTIONAL,
    ...
}

FacilityReason ::= CHOICE
{
    routeCallToGatekeeper   NULL,     -- соединение должно использовать гейткипер
                                      -- в модели с гейткипером в качестве
                                      -- alternativeAddress
    callForwarded           NULL,
    routeCallToMC           NULL,
    undefinedReason         NULL,
    ...,
    conferenceListChoice    NULL,
    startH245               NULL,     -- получатель будет соединен с h245Address
    noH245                  NULL,     -- конечная точка не поддерживает H.245
    newTokens               NULL,
    featureSetUpdate        NULL,
    forwardedElements       NULL,
    transportedInformation  NULL
}

Progress-UUIE ::= SEQUENCE
{
    protocolIdentifier      ProtocolIdentifier,
    destinationInfo         EndpointType,
    h245Address             TransportAddress OPTIONAL,
    callIdentifier          CallIdentifier,
    h245SecurityMode        H245Security OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart               SEQUENCE OF OCTET STRING OPTIONAL,
    ...,
    multipleCalls           BOOLEAN,
    maintainConnection      BOOLEAN,
    fastConnectRefused      NULL OPTIONAL
}
```

```
TransportAddress ::= CHOICE
{
     ipAddress SEQUENCE
     {
          ip             OCTET STRING (SIZE(4)),
          port           INTEGER(0..65535)
     },
     ipSourceRoute      SEQUENCE
     {
          ip             OCTET STRING (SIZE(4)),
          port           INTEGER(0..65535),
          route          SEQUENCE OF OCTET STRING (SIZE(4)),
          routing        CHOICE
          {
               strict    NULL,
               loose     NULL,
               ...
          },
          ...
     },
     ipxAddress     SEQUENCE
     {
          node           OCTET STRING (SIZE(6)),
          netnum         OCTET STRING (SIZE(4)),
          port           OCTET STRING (SIZE(2))
     },
     ip6Address     SEQUENCE
     {
          ip             OCTET STRING (SIZE(16)),
          port           INTEGER(0..65535),
          ...
     },
     netBios            OCTET STRING (SIZE(16)),
     nsap               OCTET STRING (SIZE(1..20)),
     nonStandardAddress NonStandardParameter,
     ...
}

Status-UUIE ::= SEQUENCE
{
     protocolIdentifier ProtocolIdentifier,
     callIdentifier     CallIdentifier,
     tokens             SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
     ...
}

StatusInquiry-UUIE ::= SEQUENCE
{
     protocolIdentifier ProtocolIdentifier,
     callIdentifier     CallIdentifier,
     tokens             SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
     ...
}

SetupAcknowledge-UUIE ::= SEQUENCE
{
     protocolIdentifier ProtocolIdentifier,
     callIdentifier     CallIdentifier,
     tokens             SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
     ...
}
```

```
Notify-UUIE ::= SEQUENCE
{
     protocolIdentifier ProtocolIdentifier,
     callIdentifier     CallIdentifier,
     tokens             SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
     ...
}

-- Начало раздела об общих элементах сообщений

EndpointType ::= SEQUENCE
{
     nonStandardData    NonStandardParameter OPTIONAL,
     vendor             VendorIdentifier OPTIONAL,
     gatekeeper         GatekeeperInfo OPTIONAL,
     gateway            GatewayInfo OPTIONAL,
     mcu                McuInfo OPTIONAL,  -- мс должен быть также установлен
     terminal           TerminalInfo OPTIONAL,
     mc                 BOOLEAN,           -- не должен устанавливаться сам
     undefinedNode      BOOLEAN,
     ...,
     set           BIT STRING (SIZE(32)) OPTIONAL,
                           -- не должен использоваться с мс; кодовые точки
                           -- гейткипера для различных устройств SET (простых
                           -- типов конечных точек) определены в
                           -- соответствующих Приложениях о SET
     supportedTunnelledProtocols  SEQUENCE OF TunnelledProtocol OPTIONAL
                           -- перечислить поддерживаемые туннелированные
                           -- протоколы
}

GatewayInfo ::= SEQUENCE
{
     protocol           SEQUENCE OF SupportedProtocols OPTIONAL,
     nonStandardData    NonStandardParameter OPTIONAL,
     ...
}

SupportedProtocols ::= CHOICE
{
     nonStandardData    NonStandardParameter,
     h310               H310Caps,
     h320               H320Caps,
     h321               H321Caps,
     h322               H322Caps,
     h323               H323Caps,
     h324               H324Caps,
     voice              VoiceCaps,
     t120-only          T120OnlyCaps,
     ...,
     nonStandardProtocol     NonStandardProtocol,
     t38FaxAnnexbOnly        T38FaxAnnexbOnlyCaps,
     sip                SIPCaps
}

H310Caps ::= SEQUENCE
{
     nonStandardData    NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}
```

```
H320Caps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

H321Caps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

H322Caps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

H323Caps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

H324Caps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

VoiceCaps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

T120OnlyCaps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix
}

NonStandardProtocol ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix,
     ...
}
```

```
T38FaxAnnexbOnlyCaps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix,
     t38FaxProtocol          DataProtocolCapability,
     t38FaxProfile           T38FaxProfile,
     ...
}

SIPCaps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix OPTIONAL,
     ...
}

McuInfo ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     protocol                SEQUENCE OF SupportedProtocols OPTIONAL
}

TerminalInfo ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...
}

GatekeeperInfo ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     ...
}

VendorIdentifier ::= SEQUENCE
{
     vendor         H221NonStandard,
     productId      OCTET STRING (SIZE(1..256)) OPTIONAL,      -- для каждого
                                                               -- поставщика
     versionId      OCTET STRING (SIZE(1..256)) OPTIONAL,      -- для каждого
                                                               -- продукта
     ...,
     enterpriseNumber   OBJECT IDENTIFIER OPTIONAL
}

H221NonStandard ::= SEQUENCE
{    t35CountryCode     INTEGER(0..255),
     t35Extension       INTEGER(0..255),
     manufacturerCode   INTEGER(0..65535),
     ...
}

TunnelledProtocol ::= SEQUENCE
{
     id CHOICE
     {
     tunnelledProtocolObjectID             OBJECT IDENTIFIER,
          tunnelledProtocolAlternateID     TunnelledProtocolAlternateIdentifier,
          ...
     },
     subIdentifier                         IA5String (SIZE (1..64)) OPTIONAL,
     ...
}
```

```
TunnelledProtocolAlternateIdentifier ::= SEQUENCE
{
     protocolType            IA5String (SIZE (1..64)),
     protocolVariant         IA5String (SIZE (1..64)) OPTIONAL,
     ...
}

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

NonStandardIdentifier        ::= CHOICE
{
     object                  OBJECT IDENTIFIER,
     h221NonStandard         H221NonStandard,
     ...
}

AliasAddress ::= CHOICE
{
     dialledDigits IA5String (SIZE (1..128)) (FROM ("0123456789#*,")),
     h323-ID       BMPString (SIZE (1..256)), -- Базовый код ISO/IEC 10646-1
                                              -- (Уникод)
     ...,
     url-ID        IA5String (SIZE(1..512)),  -- адрес типа URL
     transportID   TransportAddress,
     email-ID      IA5String (SIZE(1..512)),  -- адрес email, соответствующий
                                              -- rfc822
     partyNumber   PartyNumber,
     mobileUIM     MobileUIM
}

AddressPattern ::= CHOICE
{
wildcard  AliasAddress,
range          SEQUENCE
{
startOfRange  PartyNumber,
endOfRange    PartyNumber
},
...
}

PartyNumber ::= CHOICE
{
     e164Number                   PublicPartyNumber,
                                       -- план нумерации, соответствующий
                                       -- Рек. МСЭ-Т E.163 и E.164.
     dataPartyNumber              NumberDigits,
                                       -- не используется, значение
                                       -- зарезервировано
     telexPartyNumber             NumberDigits,
                                       -- не используется, значение
                                       -- зарезервировано
     privateNumber                PrivatePartyNumber,
                                       -- план нумерации, соответствующий
                                       -- ISO/IEC 11571.
     nationalStandardPartyNumber  NumberDigits,
                                       -- не используется, значение
                                       -- зарезервировано
     ...
}
```

```
PublicPartyNumber ::= SEQUENCE
{
     publicTypeOfNumber      PublicTypeOfNumber,
     publicNumberDigits      NumberDigits
}

PrivatePartyNumber ::= SEQUENCE
{
     privateTypeOfNumber     PrivateTypeOfNumber,
     privateNumberDigits     NumberDigits
}

NumberDigits ::= IA5String (SIZE (1..128)) (FROM ("0123456789#*,"))

PublicTypeOfNumber ::= CHOICE
{
     unknown                 NULL,
                                       -- если используемые цифры номера
                                       -- переносят префикс, указывающий тип
                                       -- номера согласно национальным
                                       -- рекомендациям
     internationalNumber     NULL,
     nationalNumber          NULL,
     networkSpecificNumber   NULL,
                                       -- не используется, значение
                                       -- зарезервировано
     subscriberNumber        NULL,
     abbreviatedNumber       NULL,
                                       -- действителен только для номера
                                       -- вызываемой стороны при исходящем
                                       -- доступе; сеть подставляет подходящий
                                       -- номер
...
}

PrivateTypeOfNumber ::= CHOICE
{
     unknown                 NULL,
     level2RegionalNumber    NULL,
     level1RegionalNumber    NULL,
     pISNSpecificNumber      NULL,
     localNumber             NULL,
     abbreviatedNumber       NULL,
     ...
}

MobileUIM ::= CHOICE
{
     ansi-41-uim ANSI-41-UIM,     -- Американские стандарты радиосетей
     gsm-uim GSM-UIM,             -- Европейские стандарты радиосетей
     ...
}

TBCD-STRING ::= IA5String (FROM ("0123456789#*abc"))
```

```
ANSI-41-UIM ::= SEQUENCE
{
     imsi               TBCD-STRING (SIZE (3..16)) OPTIONAL,
     min                TBCD-STRING (SIZE (3..16)) OPTIONAL,
     mdn                TBCD-STRING (SIZE (3..16)) OPTIONAL,
     msisdn             TBCD-STRING (SIZE (3..16)) OPTIONAL,
     esn                TBCD-STRING (SIZE (16)) OPTIONAL,
     mscid              TBCD-STRING (SIZE (3..16)) OPTIONAL,
     system-id CHOICE
     {
          sid           TBCD-STRING (SIZE (1..4)),
          mid           TBCD-STRING (SIZE (1..4)),
          ...
     },
     systemMyTypeCode   OCTET STRING (SIZE (1)) OPTIONAL,
     systemAccessType   OCTET STRING (SIZE (1)) OPTIONAL,
     qualificationInformationCode OCTET STRING (SIZE (1)) OPTIONAL,
     sesn               TBCD-STRING (SIZE (16)) OPTIONAL,
     soc                TBCD-STRING (SIZE (3..16)) OPTIONAL,
     ...
     -- IMSI относится к International Mobile Station Identification
     -- MIN относится к Mobile Identification Number
     -- MDN относится к Mobile Directory Number
     -- MSISDN относится к Mobile Station ISDN-номерам
     -- ESN относится к Electronic Serial Number
     -- MSCID относится к Mobile Switching Center-номерам + Market ID
     -- (идентификатор) рынка или системы
     -- SID относится к System Identification, а MID относится к Market
     -- Identification
     -- SystemMyTypeCode относится к номеру идентификации поставщика
     -- SystemAccessType относится к типу доступа к системе, например refers to
     -- регистрация пропадания питания или инициирование соединения, или
     -- ответ на короткое сообщение и т. п.
     -- Qualification относится к правильности
     -- SESN относится к SIM Electronic Serial Number для целей безопасности
     -- пользователя
     -- SOC относится к System Operator Code
}

GSM-UIM ::= SEQUENCE
{
     imsi               TBCD-STRING (SIZE (3..16)) OPTIONAL,
     tmsi               OCTET STRING (SIZE (1..4)) OPTIONAL,
     msisdn             TBCD-STRING (SIZE (3..16)) OPTIONAL,
     imei               TBCD-STRING (SIZE (15..16)) OPTIONAL,
     hplmn              TBCD-STRING (SIZE (1..4)) OPTIONAL,
     vplmn              TBCD-STRING (SIZE (1..4)) OPTIONAL,
     -- IMSI относится к International Mobile Station Identification
     -- MSISDN относится к Mobile Station ISDN-номерам
     -- IMEI относится к International Mobile Equipment Identification
     -- VPLMN или HPLMN относится к Visiting (или Home) Public Land Mobile
     -- Network-номеру
     ...
}

ExtendedAliasAddress ::= SEQUENCE
{
     address                AliasAddress,
     presentationIndicator  PresentationIndicator OPTIONAL,
     screeningIndicator     ScreeningIndicator OPTIONAL,
     ...
}
```

```
Endpoint ::= SEQUENCE
{
     nonStandardData              NonStandardParameter OPTIONAL,
     aliasAddress                 SEQUENCE OF AliasAddress OPTIONAL,
     callSignalAddress            SEQUENCE OF TransportAddress OPTIONAL,
     rasAddress                   SEQUENCE OF TransportAddress OPTIONAL,
     endpointType                 EndpointType OPTIONAL,
     tokens                       SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                 SEQUENCE OF CryptoH323Token OPTIONAL,
     priority                     INTEGER(0..127) OPTIONAL,
     remoteExtensionAddress       SEQUENCE OF AliasAddress OPTIONAL,
     destExtraCallInfo            SEQUENCE OF AliasAddress OPTIONAL,
     ...,
     alternateTransportAddresses  AlternateTransportAddresses OPTIONAL,
     circuitInfo                  CircuitInfo   OPTIONAL,
     featureSet                   FeatureSet    OPTIONAL
}

AlternateTransportAddresses ::= SEQUENCE
{
     annexE             SEQUENCE OF TransportAddress OPTIONAL,
     ...,
     sctp               SEQUENCE OF TransportAddress OPTIONAL
}

UseSpecifiedTransport ::= CHOICE
{
     tcp                NULL,
     annexE             NULL,
     ...,
     sctp               NULL
}

AlternateGK ::= SEQUENCE
{
     rasAddress              TransportAddress,
     gatekeeperIdentifier    GatekeeperIdentifier  OPTIONAL,
     needToRegister          BOOLEAN,
     priority                INTEGER (0..127),
     ...
}

AltGKInfo ::=SEQUENCE
{
     alternateGatekeeper     SEQUENCE OF AlternateGK,
     altGKisPermanent        BOOLEAN,
     ...
}

SecurityServiceMode ::= CHOICE
{
     nonStandard        NonStandardParameter,
     none               NULL,
     default            NULL,
     ...                -- может быть расширен другими конкретными режимами
}
```

```
SecurityCapabilities ::= SEQUENCE
{
     nonStandard             NonStandardParameter OPTIONAL,
     encryption              SecurityServiceMode,
     authenticaton           SecurityServiceMode,
     integrity               SecurityServiceMode,
     ...
}

SecurityErrors     ::=  CHOICE
{
     securityWrongSyncTime       NULL,     -- либо проблема в сервере времени,
                                           -- либо задержка в сети
     securityReplay              NULL,     -- встретилась атака на ответ
     securityWrongGeneralID      NULL,     -- ошибочный общий идентификатор
     securityWrongSendersID      NULL,     -- ошибочный идентификатор
                                           -- передатчика
     securityIntegrityFailed     NULL,     -- неудачная проверка целостности
     securityWrongOID            NULL,     -- ошибочные OID (идентификаторы
                                           -- объекта) маркеров или крипто-
                                           -- алгоритмов
     securityDHmismatch          NULL,     -- несогласованность параметров
                                           -- DH (алгоритма Диффи-Хеллмана)
     securityCertificateExpired  NULL,     -- истек срок действия сертификата
     securityCertificateDateInvalid   NULL,     -- сертификат еще не действителен
     securityCertificateRevoked  NULL,     -- сертификат оказался аннулированным
     securityCertificateNotReadable   NULL,     -- ошибка декодирования
     securityCertificateSignatureInvalid   NULL,     -- ошибочная сигнатура в
                                                     -- сертификате
     securityCertificateMissing  NULL,     -- нет доступного сертификата
     securityCertificateIncomplete NULL,   -- отсутствуют ожидаемые расширения
                                           -- сертификата
     securityUnsupportedCertificateAlgOID  NULL, -- крипто-алгоритмы непонятны
     securityUnknownCA           NULL,     -- CA(источник сертификата)/корневой
                                           -- сертификат не найден
...
}

SecurityErrors2    ::=  CHOICE
{
     securityWrongSyncTime   NULL,     -- либо проблема в сервере времени, либо
                                       -- задержка в сети
     securityReplay          NULL,     -- встретилась атака на ответ
     securityWrongGeneralID  NULL,     -- ошибочный общий идентификатор
     securityWrongSendersID  NULL,     -- ошибочный идентификатор передатчика
     securityIntegrityFailed NULL,     -- неудачная проверка целостности
     securityWrongOID        NULL,     -- ошибочные OID (идентификаторы объекта)
                                       -- маркеров или крипто-алгоритмов
     ...
}

H245Security ::= CHOICE
{
     nonStandard             NonStandardParameter,
     noSecurity              NULL,
     tls                     SecurityCapabilities,
     ipsec                   SecurityCapabilities,
     ...
}
```

```
QseriesOptions ::= SEQUENCE
{
     q932Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.932
     q951Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.951
     q952Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.952
     q953Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.953
     q955Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.955
     q956Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.956
     q957Full       BOOLEAN,  -- если ИСТИНА, то указывает полную поддержку Q.957
     q954Info       Q954Details,
     ...
}

Q954Details ::= SEQUENCE
{
     conferenceCalling       BOOLEAN,
     threePartyService       BOOLEAN,
     ...
}

GloballyUniqueID        ::=  OCTET STRING (SIZE(16))
ConferenceIdentifier    ::=  GloballyUniqueID
RequestSeqNum           ::=  INTEGER (1..65535)
GatekeeperIdentifier    ::=  BMPString (SIZE(1..128))
BandWidth               ::=  INTEGER (0..4294967295) -- в сотнях битов
CallReferenceValue      ::=  INTEGER (0..65535)
EndpointIdentifier      ::=  BMPString (SIZE(1..128))
ProtocolIdentifier      ::=  OBJECT IDENTIFIER
TimeToLive              ::=  INTEGER (1..4294967295) -- в секундах
H248PackagesDescriptor  ::=  OCTET STRING  -- Эта цепочка октетов содержит
                                           -- PackagesDescriptor H.248
                                           -- с кодированием PER ASN.1

H248SignalsDescriptor   ::=  OCTET STRING  -- Эта цепочка октетов содержит
                                           -- SignalsDescriptor H.248
                                           -- с кодированием PER ASN.1

FeatureDescriptor       ::=  GenericData

CallIdentifier ::= SEQUENCE
{
     guid           GloballyUniqueID,
     ...
}

EncryptIntAlg ::= CHOICE
{    -- алгоритмы внутреннего шифрования для целостности сообщений RAS
     nonStandard        NonStandardParameter,
     isoAlgorithm       OBJECT IDENTIFIER, -- определен в ISO/IEC 9979
     ...
}
NonIsoIntegrityMechanism ::= CHOICE
{    -- использован механизм HMAC, без усечения, может быть необходимо
     -- тестирование!
     hMAC-MD5           NULL,
     hMAC-iso10118-2-s  EncryptIntAlg,     -- согласно ISO/IEC 10118-2
                                           -- с использованием EncryptIntAlg в
                                           -- качестве алгоритма внутреннего
                                           -- поблочного шифрования
                                           -- (короткий MAC)
     hMAC-iso10118-2-l  EncryptIntAlg,     -- согласно ISO/IEC 10118-2
                                           -- с использованием EncryptIntAlg
                                           -- в качестве алгоритма внутреннего
                                           -- поблочного шифрования
                                           -- (длинный MAC)
```

```
    hMAC-iso10118-3    OBJECT IDENTIFIER, -- согласно ISO/IEC 10118-3
                                          -- с использованием OID в качестве
                                          -- хеш-функции (OID является SHA-1,
                                          -- RIPE-MD160,
                                          -- RIPE-MD128)
    ...
}

IntegrityMechanism ::= CHOICE
{   -- for RAS message integrity
    nonStandard    NonStandardParameter,
    digSig         NULL,                -- указывает на применение цифровой подписи
    iso9797        OBJECT IDENTIFIER, -- согласно ISO/IEC 9797 с использованием
                                      -- OID в качестве алгоритма внутреннего
                                      -- шифрования (X-CBC MAC)
    nonIsoIM       NonIsoIntegrityMechanism,
    ...
}

ICV ::= SEQUENCE
{
    algorithmOID  OBJECT IDENTIFIER,    -- алгоритм, используемый для
                                        -- вычисления подписи
    icv           BIT STRING            -- вычисленное значение проверки
                                        -- криптографической целостности
                                        -- или подпись
}

FastStartToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, dhkey
PRESENT, generalID PRESENT
                                    -- установить в "alias" (псевдоним) -- }
EncodedFastStartToken ::= TYPE-IDENTIFIER.&Type (FastStartToken)
CryptoH323Token::= CHOICE
{
    cryptoEPPwdHash SEQUENCE
    {
        alias         AliasAddress,    -- псевдоним объекта, генерирующего
                                       -- хеш-код
        timeStamp     TimeStamp,       -- метка времени, используемая
                                       -- в хеш-коде
        token         HASHED    { EncodedPwdCertToken -- общий идентификатор,
                                          -- установленный в "alias" -- }
    },
    cryptoGKPwdHash  SEQUENCE
    {
        gatekeeperId  GatekeeperIdentifier,   -- идентификатор гейткипера,
                                              -- генерирующего хеш-код
        timeStamp     TimeStamp,              -- метка времени, используемая
                                              -- в хеш-коде
        token         HASHED    { EncodedPwdCertToken -- общий идентификатор,
                                                      -- установленный в
                                                      -- идентификатор
                                                      -- гейткипера -- }
    },
    cryptoEPPwdEncr    ENCRYPTED { EncodedPwdCertToken -- общий идентификатор,
                                                       -- установленный в
                                                       -- идентификатор
                                                       -- гейткипера --},
    cryptoGKPwdEncr    ENCRYPTED { EncodedPwdCertToken -- общий идентификатор,
                                                       -- установленный в
                                                       -- идентификатор
                                                       -- гейткипера --},
    cryptoEPCert       SIGNED { EncodedPwdCertToken -- общий идентификатор,
                                                    -- установленный
                                                    -- в идентификатор
                                                    -- гейткипера --},
    cryptoGKCert       SIGNED { EncodedPwdCertToken  -- общий идентификатор,
                                                     -- установленный в "alias" -- },
    cryptoFastStart    SIGNED { EncodedFastStartToken },
    nestedcryptoToken  CryptoToken,
    ...
}
```

```
DataRate ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     channelRate             BandWidth,
     channelMultiplier       INTEGER (1..256) OPTIONAL,
     ...
}

CallLinkage ::= SEQUENCE
{
     globalCallId            GloballyUniqueID  OPTIONAL,
     threadId                GloballyUniqueID  OPTIONAL,
     ...
}

SupportedPrefix ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     prefix                  AliasAddress,
     ...
}

CapacityReportingCapability ::= SEQUENCE
{
     canReportCallCapacity       BOOLEAN,
     ...
}

CapacityReportingSpecification ::= SEQUENCE
{
     when SEQUENCE
     {
          callStart     NULL OPTIONAL,
          callEnd       NULL OPTIONAL,
          ...
     },
     ...
}

CallCapacity ::= SEQUENCE
{
     maximumCallCapacity         CallCapacityInfo OPTIONAL,
     currentCallCapacity         CallCapacityInfo OPTIONAL,
     ...
}

CallCapacityInfo ::= SEQUENCE
{
     voiceGwCallsAvailable                SEQUENCE OF CallsAvailable OPTIONAL,
     h310GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     h320GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     h321GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     h322GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     h323GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     h324GwCallsAvailable                 SEQUENCE OF CallsAvailable OPTIONAL,
     t120OnlyGwCallsAvailable             SEQUENCE OF CallsAvailable OPTIONAL,
     t38FaxAnnexbOnlyGwCallsAvailable     SEQUENCE OF CallsAvailable OPTIONAL,
     terminalCallsAvailable               SEQUENCE OF CallsAvailable OPTIONAL,
     mcuCallsAvailable                    SEQUENCE OF CallsAvailable OPTIONAL,
     ...,
     sipGwCallsAvailable                  SEQUENCE OF CallsAvailable OPTIONAL
}
```

```
CallsAvailable ::= SEQUENCE
{
     calls          INTEGER (0..4294967295),
     group          IA5String (SIZE (1..128)) OPTIONAL,
     ...,
     carrier        CarrierInfo OPTIONAL
}

CircuitInfo ::= SEQUENCE
{
     sourceCircuitID             CircuitIdentifier OPTIONAL,
     destinationCircuitID        CircuitIdentifier OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL,
     ...
}

CircuitIdentifier ::= SEQUENCE
{
     cic            CicInfo OPTIONAL,  group         GroupID OPTIONAL,
     ...,
     carrier        CarrierInfo OPTIONAL
}

CicInfo ::= SEQUENCE
{
     cic            SEQUENCE OF OCTET STRING (SIZE (2..4)),
     pointCode      OCTET STRING (SIZE (2..5)),
     ...
}

GroupID ::= SEQUENCE
{
     member         SEQUENCE OF INTEGER (0..65535) OPTIONAL,
     group          IA5String (SIZE (1..128)),
     ...
}

CarrierInfo ::= SEQUENCE
{
     carrierIdentificationCode   OCTET STRING (SIZE (3..4)) OPTIONAL,
     carrierName                 IA5String (SIZE (1..128)) OPTIONAL,
     ...
}

ServiceControlDescriptor ::= CHOICE
{
     url                         IA5String (SIZE(0..512)),  -- указывает
                                                            -- протокол/ресурс
                                                            -- с ссылкой на URL
     signal                      H248SignalsDescriptor,
     nonStandard                 NonStandardParameter,
     callCreditServiceControl    CallCreditServiceControl,
     ...
}

ServiceControlSession ::= SEQUENCE
{
     sessionId      INTEGER (0..255),
     contents       ServiceControlDescriptor OPTIONAL,
     reason CHOICE
```

```
    {
        open            NULL,
        refresh         NULL,
        close           NULL,
        ...
    },
    ...
}

RasUsageInfoTypes ::= SEQUENCE
{
    nonStandardUsageTypes       SEQUENCE OF NonStandardParameter,
    startTime                   NULL OPTIONAL,
    endTime                     NULL OPTIONAL,
    terminationCause            NULL OPTIONAL,
    ...
}

RasUsageSpecification ::= SEQUENCE
{
    when SEQUENCE
    {
        start                   NULL OPTIONAL,
        end                     NULL OPTIONAL,
        inIrr                   NULL OPTIONAL,
        ...
    },
    callStartingPoint SEQUENCE
    {
        alerting            NULL OPTIONAL,
        connect             NULL OPTIONAL,
        ...
    } OPTIONAL,
    required                RasUsageInfoTypes,
    ...
}

RasUsageInformation ::= SEQUENCE
{
    nonStandardUsageFields      SEQUENCE OF NonStandardParameter,
    alertingTime                TimeStamp OPTIONAL,
    connectTime                 TimeStamp OPTIONAL,
    endTime                     TimeStamp OPTIONAL,
    ...
}

CallTerminationCause ::= CHOICE
{
    releaseCompleteReason       ReleaseCompleteReason,
    releaseCompleteCauseIE      OCTET STRING (SIZE(2..32)),
    ...
}

BandwidthDetails ::= SEQUENCE
{
    sender              BOOLEAN,                -- ИСТИНА=передатчик
                                                -- ЛОЖЬ=передатчик
    multicast           BOOLEAN,                -- ИСТИНА, если поток
                                                -- многопунктовый
    bandwidth           BandWidth,              -- Полоса пропускания,
                                                   использованная для потока
    rtcpAddresses       TransportChannelInfo,   -- Адреса RTCP для потока носителей
    ...
}

CallCreditCapability ::= SEQUENCE
```

```
{
     canDisplayAmountString       BOOLEAN OPTIONAL,
     canEnforceDurationLimit      BOOLEAN OPTIONAL,
     ...
}

CallCreditServiceControl ::= SEQUENCE
{
     amountString            BMPString (SIZE (1..512)) OPTIONAL,   -- (Уникод)
     billingMode CHOICE
     {
          credit             NULL,
          debit              NULL,
          ...
     } OPTIONAL,
     callDurationLimit       INTEGER (1..4294967295) OPTIONAL,    -- в секундах
     enforceCallDurationLimit    BOOLEAN OPTIONAL,
     callStartingPoint CHOICE
     {
          alerting           NULL,
          connect            NULL,
          ...
     } OPTIONAL,
     ...
}

GenericData ::= SEQUENCE
{
     id                 GenericIdentifier,
     parameters         SEQUENCE (SIZE (1..512)) OF EnumeratedParameter OPTIONAL,
     ...
}

GenericIdentifier ::= CHOICE
{
     standard           INTEGER(0..16383,...),
     oid                OBJECT IDENTIFIER,
     nonStandard        GloballyUniqueID,
     ...
}

EnumeratedParameter ::= SEQUENCE
{
     id                 GenericIdentifier,
     content            Content OPTIONAL,
     ...
}

Content ::= CHOICE
{
     raw                OCTET STRING,
     text               IA5String,
     unicode            BMPString,
     bool               BOOLEAN,
     number8            INTEGER (0..255),
     number16           INTEGER (0..65535),
     number32           INTEGER (0..4294967295),
     id                 GenericIdentifier,
     alias              AliasAddress,
     transport          TransportAddress,
     compound           SEQUENCE (SIZE (1..512)) OF EnumeratedParameter,
     nested             SEQUENCE (SIZE (1..16)) OF GenericData,
     ...
}
```

```
FeatureSet ::= SEQUENCE
{
     replacementFeatureSet   BOOLEAN,
     neededFeatures          SEQUENCE OF FeatureDescriptor OPTIONAL,
     desiredFeatures         SEQUENCE OF FeatureDescriptor OPTIONAL,
     supportedFeatures       SEQUENCE OF FeatureDescriptor OPTIONAL,
     ...
}

TransportChannelInfo ::= SEQUENCE
{
     sendAddress             TransportAddress OPTIONAL,
     recvAddress             TransportAddress OPTIONAL,
     ...
}

RTPSession ::= SEQUENCE
{
     rtpAddress                  TransportChannelInfo,
     rtcpAddress                 TransportChannelInfo,
     cname                       PrintableString,
     ssrc                        INTEGER (1..4294967295),
     sessionId                   INTEGER (1..255),
     associatedSessionIds        SEQUENCE OF INTEGER (1..255),
     ...,
     multicast                   NULL OPTIONAL,
     bandwidth                   BandWidth OPTIONAL
}

RasMessage ::= CHOICE
{
     gatekeeperRequest           GatekeeperRequest,
     gatekeeperConfirm           GatekeeperConfirm,
     gatekeeperReject            GatekeeperReject,
     registrationRequest         RegistrationRequest,
     registrationConfirm         RegistrationConfirm,
     registrationReject          RegistrationReject,
     unregistrationRequest       UnregistrationRequest,
     unregistrationConfirm       UnregistrationConfirm,
     unregistrationReject        UnregistrationReject,
     admissionRequest            AdmissionRequest,
     admissionConfirm            AdmissionConfirm,
     admissionReject             AdmissionReject,
     bandwidthRequest            BandwidthRequest,
     bandwidthConfirm            BandwidthConfirm,
     bandwidthReject             BandwidthReject,
     disengageRequest            DisengageRequest,
     disengageConfirm            DisengageConfirm,
     disengageReject             DisengageReject,
     locationRequest             LocationRequest,
     locationConfirm             LocationConfirm,
     locationReject              LocationReject,
     infoRequest                 InfoRequest,
     infoRequestResponse         InfoRequestResponse,
     nonStandardMessage          NonStandardMessage,
     unknownMessageResponse      UnknownMessageResponse,
     ...,
     requestInProgress           RequestInProgress,
     resourcesAvailableIndicate  ResourcesAvailableIndicate,
     resourcesAvailableConfirm   ResourcesAvailableConfirm,
```

```
    infoRequestAck              InfoRequestAck,
    infoRequestNak              InfoRequestNak,
    serviceControlIndication    ServiceControlIndication,
    serviceControlResponse      ServiceControlResponse,
    admissionConfirmSequence    SEQUENCE OF AdmissionConfirm
}

GatekeeperRequest ::= SEQUENCE --(сообщение GRQ)
{

    requestSeqNum               RequestSeqNum,
    protocolIdentifier          ProtocolIdentifier,
    nonStandardData             NonStandardParameter OPTIONAL,
    rasAddress                  TransportAddress,
    endpointType                EndpointType,
    gatekeeperIdentifier        GatekeeperIdentifier OPTIONAL,
    callServices                QseriesOptions  OPTIONAL,
    endpointAlias               SEQUENCE OF AliasAddress OPTIONAL,
    ...,
    alternateEndpoints          SEQUENCE OF Endpoint OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    authenticationCapability    SEQUENCE OF AuthenticationMechanism OPTIONAL,
    algorithmOIDs               SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
    integrity                   SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    supportsAltGK               NULL OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}

GatekeeperConfirm ::= SEQUENCE --(сообщение GCF)
{
    requestSeqNum               RequestSeqNum,
    protocolIdentifier          ProtocolIdentifier,
    nonStandardData             NonStandardParameter OPTIONAL,
    gatekeeperIdentifier        GatekeeperIdentifier  OPTIONAL,
    rasAddress                  TransportAddress,
    ...,
    alternateGatekeeper         SEQUENCE OF AlternateGK OPTIONAL,
    authenticationMode          AuthenticationMechanism OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    algorithmOID                OBJECT IDENTIFIER OPTIONAL,
    integrity                   SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}

GatekeeperReject ::= SEQUENCE --(сообщение GRJ)
{
    requestSeqNum               RequestSeqNum,
    protocolIdentifier          ProtocolIdentifier,
    nonStandardData             NonStandardParameter OPTIONAL,
    gatekeeperIdentifier        GatekeeperIdentifier  OPTIONAL,
    rejectReason                GatekeeperRejectReason,
    ...,
    altGKInfo                   AltGKInfo OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
```

```
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}

GatekeeperRejectReason ::= CHOICE
{
    resourceUnavailable         NULL,
    terminalExcluded            NULL,     -- отказ в разрешении, а не отказ
                                          -- в ресурсе
    invalidRevision             NULL,
    undefinedReason             NULL,
    ...,
    securityDenial              NULL,
    genericDataReason           NULL,
    neededFeatureNotSupported   NULL,
securityError               SecurityErrors}

RegistrationRequest ::= SEQUENCE --(сообщение RRQ)
{
    requestSeqNum               RequestSeqNum,
    protocolIdentifier          ProtocolIdentifier,
    nonStandardData             NonStandardParameter OPTIONAL,
    discoveryComplete           BOOLEAN,
    callSignalAddress           SEQUENCE OF TransportAddress,
    rasAddress                  SEQUENCE OF TransportAddress,
    terminalType                EndpointType,
    terminalAlias               SEQUENCE OF AliasAddress OPTIONAL,
    gatekeeperIdentifier        GatekeeperIdentifier  OPTIONAL,
    endpointVendor              VendorIdentifier,
    ...,
    alternateEndpoints          SEQUENCE OF Endpoint OPTIONAL,
    timeToLive                  TimeToLive OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    keepAlive                   BOOLEAN,
    endpointIdentifier          EndpointIdentifier OPTIONAL,
    willSupplyUUIEs             BOOLEAN,
    maintainConnection          BOOLEAN,
    alternateTransportAddresses AlternateTransportAddresses OPTIONAL,
    additiveRegistration        NULL OPTIONAL,
    terminalAliasPattern        SEQUENCE OF AddressPattern OPTIONAL,
    supportsAltGK               NULL OPTIONAL,
    usageReportingCapability    RasUsageInfoTypes OPTIONAL,
    multipleCalls               BOOLEAN OPTIONAL,
    supportedH248Packages       SEQUENCE OF H248PackagesDescriptor OPTIONAL,
    callCreditCapability        CallCreditCapability OPTIONAL,
    capacityReportingCapability CapacityReportingCapability OPTIONAL,
    capacity                    CallCapacity OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL,
    restart                     NULL     OPTIONAL,
    supportsACFSequences        NULL     OPTIONAL
}

RegistrationConfirm ::= SEQUENCE --(сообщение RCF)
{
    requestSeqNum                       RequestSeqNum,
    protocolIdentifier                  ProtocolIdentifier,
    nonStandardData                     NonStandardParameter OPTIONAL,
    callSignalAddress                   SEQUENCE OF TransportAddress,
    terminalAlias                       SEQUENCE OF AliasAddress OPTIONAL,
```

```
    gatekeeperIdentifier                GatekeeperIdentifier  OPTIONAL,
    endpointIdentifier                  EndpointIdentifier,
    ...,
    alternateGatekeeper                 SEQUENCE OF AlternateGK OPTIONAL,
    timeToLive                          TimeToLive OPTIONAL,
    tokens                              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue                 ICV OPTIONAL,
    willRespondToIRR                    BOOLEAN,
    preGrantedARQ                       SEQUENCE
    {
        makeCall                        BOOLEAN,
        useGKCallSignalAddressToMakeCall BOOLEAN,
        answerCall                      BOOLEAN,
        useGKCallSignalAddressToAnswer  BOOLEAN,
        ...,
        irrFrequencyInCall      INTEGER (1..65535) OPTIONAL, -- в секундах;
                                                             -- не присутствует,
                                                             -- если GK не
                                                             -- нуждается в
                                                             -- сообщениях IRR
        totalBandwidthRestriction   BandWidth OPTIONAL,      -- общий предел
                                                             -- для всех
                                                             -- одновременных
                                                             -- соединений
        alternateTransportAddresses     AlternateTransportAddresses OPTIONAL,
        useSpecifiedTransport           UseSpecifiedTransport OPTIONAL
    } OPTIONAL,
    maintainConnection              BOOLEAN,
    serviceControl                  SEQUENCE OF ServiceControlSession OPTIONAL,
    supportsAdditiveRegistration    NULL OPTIONAL,
    terminalAliasPattern            SEQUENCE OF AddressPattern OPTIONAL,
    supportedPrefixes               SEQUENCE OF SupportedPrefix OPTIONAL,
    usageSpec                       SEQUENCE OF RasUsageSpecification OPTIONAL,
    featureServerAlias              AliasAddress OPTIONAL,
    capacityReportingSpec           CapacityReportingSpecification OPTIONAL,
    featureSet                      FeatureSet OPTIONAL,
    genericData                     SEQUENCE OF GenericData OPTIONAL
}

RegistrationReject ::= SEQUENCE --(сообщение RRJ)
{
    requestSeqNum               RequestSeqNum,
    protocolIdentifier          ProtocolIdentifier,
    nonStandardData             NonStandardParameter OPTIONAL,
    rejectReason                RegistrationRejectReason,
    gatekeeperIdentifier        GatekeeperIdentifier  OPTIONAL,
    ...,
    altGKInfo                   AltGKInfo OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}

RegistrationRejectReason ::= CHOICE
{
    discoveryRequired           NULL,
    invalidRevision             NULL,
    invalidCallSignalAddress    NULL,
    invalidRASAddress           NULL,     -- предоставленный адрес
                                          -- недействителен
    duplicateAlias              SEQUENCE OF AliasAddress,
                                          -- псевдоним зарегистрирован для
                                          -- другой конечной точки
    invalidTerminalType         NULL,
```

```
    undefinedReason              NULL,
    transportNotSupported        NULL,     -- один транспорт или больше
    ...,
    transportQOSNotSupported     NULL,     -- КО для конечной точки
    resourceUnavailable          NULL,     -- ресурсы гейткипера исчерпаны
    invalidAlias                 NULL,     -- псевдоним не удовлетворяет
                                           -- правилам гейткипера
    securityDenial               NULL,
    fullRegistrationRequired     NULL,     -- срок разрешения от регистрации
                                           -- закончился
    additiveRegistrationNotSupported NULL,
    invalidTerminalAliases           SEQUENCE
    {
        terminalAlias            SEQUENCE OF AliasAddress OPTIONAL,
        terminalAliasPattern     SEQUENCE OF AddressPattern OPTIONAL,
        supportedPrefixes        SEQUENCE OF SupportedPrefix OPTIONAL,
        ...
    },
    genericDataReason            NULL,
    needed FeatureNotSupported   NULL,
    securityError                SecurityErrors
}

UnregistrationRequest ::= SEQUENCE --(сообщение URQ)
{
    requestSeqNum            RequestSeqNum,
    callSignalAddress        SEQUENCE OF TransportAddress,
    endpointAlias            SEQUENCE OF AliasAddress OPTIONAL,
    nonStandardData          NonStandardParameter OPTIONAL,
    endpointIdentifier       EndpointIdentifier OPTIONAL,
    ...,
    alternateEndpoints       SEQUENCE OF Endpoint OPTIONAL,
    gatekeeperIdentifier     GatekeeperIdentifier OPTIONAL,
    tokens                   SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue      ICV OPTIONAL,
    reason                   UnregRequestReason OPTIONAL,
    endpointAliasPattern     SEQUENCE OF AddressPattern OPTIONAL,
    supportedPrefixes        SEQUENCE OF SupportedPrefix OPTIONAL,
    alternateGatekeeper      SEQUENCE OF AlternateGK OPTIONAL,
    genericData              SEQUENCE OF GenericData OPTIONAL
}

UnregRequestReason ::= CHOICE
{
    reregistrationRequired   NULL,
    ttlExpired               NULL,
    securityDenial           NULL,
    undefinedReason          NULL,
    ...,
    maintenance              NULL,
    securityError            SecurityErrors2
}

UnregistrationConfirm ::= SEQUENCE --(сообщение UCF)
{
    requestSeqNum            RequestSeqNum,
    nonStandardData          NonStandardParameter OPTIONAL,
    ...,
    tokens                   SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue      ICV OPTIONAL,
    genericData              SEQUENCE OF GenericData OPTIONAL
}
```

```
UnregistrationReject ::= SEQUENCE --(сообщение URJ)
{

     requestSeqNum           RequestSeqNum,
     rejectReason            UnregRejectReason,
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     altGKInfo               AltGKInfo OPTIONAL,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     genericData             SEQUENCE OF GenericData OPTIONAL
}

UnregRejectReason ::= CHOICE
{
     notCurrentlyRegistered  NULL,
     callInProgress          NULL,
     undefinedReason         NULL,
     ...,
     permissionDenied        NULL,     -- запрашивающему пользователю не
                                       -- разрешается связь с незарегистрированным
                                       -- указанным пользователем
     securityDenial          NULL,
     securityError           SecurityErrors2
}

AdmissionRequest ::= SEQUENCE --(сообщение ARQ)
{
     requestSeqNum               RequestSeqNum,
     callType                    CallType,
     callModel                   CallModel OPTIONAL,
     endpointIdentifier          EndpointIdentifier,
     destinationInfo             SEQUENCE OF AliasAddress OPTIONAL,
     destCallSignalAddress       TransportAddress OPTIONAL,
     destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,
     srcInfo                     SEQUENCE OF AliasAddress,
     srcCallSignalAddress        TransportAddress OPTIONAL,
     bandWidth                   BandWidth,
     callReferenceValue          CallReferenceValue,
     nonStandardData             NonStandardParameter OPTIONAL,
     callServices                QseriesOptions  OPTIONAL,
     conferenceID                ConferenceIdentifier,
     activeMC                    BOOLEAN,
     answerCall                  BOOLEAN,  -- ответ на вызов
     ...,
     canMapAlias                 BOOLEAN,  -- может отрабатывать адрес-псевдоним
     callIdentifier              CallIdentifier,
     srcAlternatives             SEQUENCE OF Endpoint OPTIONAL,
     destAlternatives            SEQUENCE OF Endpoint OPTIONAL,
     gatekeeperIdentifier        GatekeeperIdentifier OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue         ICV OPTIONAL,
     transportQOS                TransportQOS OPTIONAL,
     willSupplyUUIEs             BOOLEAN,
     callLinkage                 CallLinkage OPTIONAL,
     gatewayDataRate             DataRate OPTIONAL,
     capacity                    CallCapacity OPTIONAL,
     circuitInfo                 CircuitInfo OPTIONAL,
     desiredProtocols            SEQUENCE OF SupportedProtocols OPTIONAL,
```

```
    desiredTunnelledProtocol    TunnelledProtocol OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL,
    canMapSrcAlias              BOOLEAN
}

CallType ::= CHOICE
{
    pointToPoint        NULL,       -- двухточечное
    oneToN              NULL,       -- без взаимодействия (FFS)
    nToOne              NULL,       -- без взаимодействия (FFS)
    nToN                NULL,       -- интерактивное (многоточечное)
    ...
}

CallModel ::= CHOICE
{
    direct              NULL,
    gatekeeperRouted    NULL,
    ...
}

TransportQOS ::= CHOICE
{
    endpointControlled      NULL,
    gatekeeperControlled    NULL,
    noControl               NULL,
    ...
}

AdmissionConfirm ::= SEQUENCE --(сообщение ACF)
{
    requestSeqNum               RequestSeqNum,
    bandWidth                   BandWidth,
    callModel                   CallModel,
    destCallSignalAddress       TransportAddress,
    irrFrequency                INTEGER (1..65535) OPTIONAL,
    nonStandardData             NonStandardParameter OPTIONAL,
    ...,
    destinationInfo             SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,
    destinationType             EndpointType OPTIONAL,
    remoteExtensionAddress      SEQUENCE OF AliasAddress OPTIONAL,
    alternateEndpoints          SEQUENCE OF Endpoint OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    transportQOS                TransportQOS OPTIONAL,
    willRespondToIRR            BOOLEAN,
    uuiesRequested              UUIEsRequested,
    language                    SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
    alternateTransportAddresses AlternateTransportAddresses OPTIONAL,
    useSpecifiedTransport       UseSpecifiedTransport OPTIONAL,
    circuitInfo                 CircuitInfo OPTIONAL,
    usageSpec                   SEQUENCE OF RasUsageSpecification OPTIONAL,
    supportedProtocols          SEQUENCE OF SupportedProtocols OPTIONAL,
    serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
    multipleCalls               BOOLEAN OPTIONAL,
    featureSet                  FeatureSet OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL,
    modifiedSrcInfo             SEQUENCE OF AliasAddress OPTIONAL
}
```

```
UUIEsRequested ::= SEQUENCE
{
     setup                    BOOLEAN,
     callProceeding           BOOLEAN,
     connect                  BOOLEAN,
     alerting                 BOOLEAN,
     information              BOOLEAN,
     releaseComplete          BOOLEAN,
     facility                 BOOLEAN,
     progress                 BOOLEAN,
     empty                    BOOLEAN,
     ...,
     status                   BOOLEAN,
     statusInquiry            BOOLEAN,
     setupAcknowledge         BOOLEAN,
     notify                   BOOLEAN
}

AdmissionReject ::= SEQUENCE --(сообщение ARJ)
{
     requestSeqNum            RequestSeqNum,
     rejectReason             AdmissionRejectReason,
     nonStandardData          NonStandardParameter OPTIONAL,
     ...,
     altGKInfo                AltGKInfo OPTIONAL,
     tokens                   SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
     callSignalAddress        SEQUENCE OF TransportAddress OPTIONAL,
     integrityCheckValue      ICV OPTIONAL,
     serviceControl           SEQUENCE OF ServiceControlSession OPTIONAL,
     featureSet               FeatureSet OPTIONAL,
     genericData              SEQUENCE OF GenericData OPTIONAL
}

AdmissionRejectReason ::= CHOICE
{
     calledPartyNotRegistered     NULL,     -- адрес не может быть транслирован
     invalidPermission            NULL,     -- срок разрешения закончился
     requestDenied                NULL,     -- полоса пропускания недоступна
     undefinedReason              NULL,
     callerNotRegistered          NULL,
     routeCallToGatekeeper        NULL,
     invalidEndpointIdentifier    NULL,
     resourceUnavailable          NULL,
     ...,
     securityDenial               NULL,
     qosControlNotSupported       NULL,
     incompleteAddress            NULL,
     aliasesInconsistent          NULL,     -- несколько псевдонимов в запросе
                                            -- определяют разных людей
     routeCallToSCN               SEQUENCE OF PartyNumber,
     exceedsCallCapacity          NULL,     -- пункт назначения не имеет
                                            -- пропускной способности для
                                            -- этого соединения
     collectDestination           NULL,
     collectPIN                   NULL,
     genericDataReason            NULL,
     neededFeatureNotSupported    NULL,
     securityError                SecurityErrors2,
     securityDHmismatch           NULL,     -- несогласованность параметров DH
     noRouteToDestination         NULL,     -- пункт назначения недоступен
     unallocatedNumber            NULL      -- нераспределенный номер пункта
                                            -- назначения
}
```

```
BandwidthRequest ::= SEQUENCE --(сообщение BRQ)
{
     requestSeqNum           RequestSeqNum,
     endpointIdentifier      EndpointIdentifier,
     conferenceID            ConferenceIdentifier,
     callReferenceValue      CallReferenceValue,
     callType                CallType OPTIONAL,
     bandWidth               BandWidth,
     nonStandardData         NonStandardParameter OPTIONAL,
     ...,
     callIdentifier          CallIdentifier,
     gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     answeredCall            BOOLEAN,
     callLinkage             CallLinkage OPTIONAL,
     capacity                CallCapacity OPTIONAL,
     usageInformation        RasUsageInformation OPTIONAL,
     bandwidthDetails        SEQUENCE OF BandwidthDetails OPTIONAL,
     genericData             SEQUENCE OF GenericData OPTIONAL
}

BandwidthConfirm ::= SEQUENCE --(сообщение BCF)
{
     requestSeqNum               RequestSeqNum,
     bandWidth                   BandWidth,
     nonStandardData             NonStandardParameter OPTIONAL,
     ...,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue         ICV OPTIONAL,
     capacity                    CallCapacity OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL
}

BandwidthReject ::= SEQUENCE --(сообщение BRJ)
{
     requestSeqNum               RequestSeqNum,
     rejectReason                BandRejectReason,
     allowedBandWidth            BandWidth,
     nonStandardData             NonStandardParameter OPTIONAL,
     ...,
     altGKInfo                   AltGKInfo OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue         ICV OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL
}

BandRejectReason ::= CHOICE
{
     notBound                    NULL,     -- разрешение обнаружения устарело
     invalidConferenceID         NULL,     -- возможен пересмотр
     invalidPermission           NULL,     -- полностью нарушенное разрешение
     insufficientResources       NULL,
     invalidRevision             NULL,
     undefinedReason             NULL,
     ...,
     securityDenial              NULL,
     securityError               SecurityErrors2}
```

```
LocationRequest ::= SEQUENCE --(сообщение LRQ)
{
     requestSeqNum               RequestSeqNum,
     endpointIdentifier          EndpointIdentifier OPTIONAL,
     destinationInfo             SEQUENCE OF AliasAddress,
     nonStandardData             NonStandardParameter OPTIONAL,
     replyAddress                TransportAddress,
     ...,
     sourceInfo                  SEQUENCE OF AliasAddress OPTIONAL,
     canMapAlias                 BOOLEAN,  -- может обрабатывать адрес-псевдоним
     gatekeeperIdentifier        GatekeeperIdentifier OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue         ICV OPTIONAL,
     desiredProtocols            SEQUENCE OF SupportedProtocols OPTIONAL,
     desiredTunnelledProtocol    TunnelledProtocol OPTIONAL,
     featureSet                  FeatureSet OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL,
     hopCount                    INTEGER (1..255) OPTIONAL,
     circuitInfo                 CircuitInfo OPTIONAL,
     callIdentifier              CallIdentifier OPTIONAL,
     bandWidth                   BandWidth OPTIONAL,
     sourceEndpointInfo          SEQUENCE OF AliasAddress OPTIONAL,
     canMapSrcAlias              BOOLEAN
}

LocationConfirm ::= SEQUENCE --(сообщение LCF)
{
     requestSeqNum               RequestSeqNum,
     callSignalAddress           TransportAddress,
     rasAddress                  TransportAddress,
     nonStandardData             NonStandardParameter OPTIONAL,
     ...,
     destinationInfo             SEQUENCE OF AliasAddress OPTIONAL,
     destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,
     destinationType             EndpointType OPTIONAL,
     remoteExtensionAddress      SEQUENCE OF AliasAddress OPTIONAL,
     alternateEndpoints          SEQUENCE OF Endpoint OPTIONAL,
     tokens                      SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue         ICV OPTIONAL,
     alternateTransportAddresses AlternateTransportAddresses OPTIONAL,
     supportedProtocols          SEQUENCE OF SupportedProtocols OPTIONAL,
     multipleCalls               BOOLEAN OPTIONAL,
     featureSet                  FeatureSet OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL,
     circuitInfo                 CircuitInfo OPTIONAL,
     serviceControl              SEQUENCE OF ServiceControlSession OPTIONAL,
     modifiedSrcInfo             SEQUENCE OF AliasAddress OPTIONAL,
     bandWidth                   BandWidth OPTIONAL
}

LocationReject ::= SEQUENCE --(сообщение LRJ)
{
     requestSeqNum          RequestSeqNum,
     rejectReason           LocationRejectReason,
     nonStandardData        NonStandardParameter OPTIONAL,
     ...,
     altGKInfo              AltGKInfo OPTIONAL,
     tokens                 SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue    ICV OPTIONAL,
     featureSet             FeatureSet OPTIONAL,
     genericData            SEQUENCE OF GenericData OPTIONAL,
```

```
    serviceControl          SEQUENCE OF ServiceControlSession OPTIONAL
}

LocationRejectReason ::= CHOICE
{
    notRegistered           NULL,
    invalidPermission       NULL,     -- исключено администратором или
                                      -- свойством
    requestDenied           NULL,     -- местонахождение не может быть найдено
    undefinedReason         NULL,
    ...,
    securityDenial              NULL,
    aliasesInconsistent         NULL,     -- несколько псевдонимов в запросе
                                          -- определяют разных людей
    routeCalltoSCN              SEQUENCE OF PartyNumber,
    resourceUnavailable         NULL,
    genericDataReason           NULL,
    neededFeatureNotSupported   NULL,
    hopCountExceeded            NULL,
    incompleteAddress           NULL,
    securityError               SecurityErrors2,
    securityDHmismatch          NULL,     -- несогласованность параметров DH
    noRouteToDestination        NULL,     -- пункт назначения недоступен
    unallocatedNumber           NULL      -- нераспределенный номер пункта
                                          -- назначения
}

DisengageRequest ::= SEQUENCE --(сообщение DRQ)
{
    requestSeqNum           RequestSeqNum,
    endpointIdentifier      EndpointIdentifier,
    conferenceID            ConferenceIdentifier,
    callReferenceValue      CallReferenceValue,
    disengageReason         DisengageReason,
    nonStandardData         NonStandardParameter OPTIONAL,
    ...,
    callIdentifier          CallIdentifier,
    gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    answeredCall            BOOLEAN,
    callLinkage             CallLinkage OPTIONAL,
    capacity                CallCapacity OPTIONAL,
    circuitInfo             CircuitInfo OPTIONAL,
    usageInformation        RasUsageInformation OPTIONAL,
    terminationCause        CallTerminationCause OPTIONAL,
    serviceControl          SEQUENCE OF ServiceControlSession OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

DisengageReason ::= CHOICE
{
    forcedDrop              NULL,     -- гейткипер принуждает к сбросу
    normalDrop              NULL,     -- относится к нормальному сбросу
    undefinedReason         NULL,
    ...
}

DisengageConfirm ::= SEQUENCE --(сообщение DCF)
{
    requestSeqNum           RequestSeqNum,
    nonStandardData         NonStandardParameter OPTIONAL,
    ...,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
```

```
    integrityCheckValue     ICV OPTIONAL,
    capacity                CallCapacity OPTIONAL,
    circuitInfo             CircuitInfo OPTIONAL,
    usageInformation        RasUsageInformation OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

DisengageReject ::= SEQUENCE --(сообщение DRJ)
{
    requestSeqNum           RequestSeqNum,
    rejectReason            DisengageRejectReason,
    nonStandardData         NonStandardParameter OPTIONAL,
    ...,
    altGKInfo               AltGKInfo OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

DisengageRejectReason ::= CHOICE
{
    notRegistered           NULL,     -- не зарегистрирован в гейткипере
    requestToDropOther      NULL,     -- не может запросить сброс для других
    ...,
    securityDenial          NULL,
    securityError           SecurityErrors2
}

InfoRequest ::= SEQUENCE --(сообщение IRQ)
{
    requestSeqNum           RequestSeqNum,
    callReferenceValue      CallReferenceValue,
    nonStandardData         NonStandardParameter OPTIONAL,
    replyAddress            TransportAddress OPTIONAL,
    ...,
    callIdentifier          CallIdentifier,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    uuiesRequested          UUIEsRequested OPTIONAL,
    callLinkage             CallLinkage OPTIONAL,
    usageInfoRequested          RasUsageInfoTypes OPTIONAL,
    segmentedResponseSupported  NULL OPTIONAL,
    nextSegmentRequested        INTEGER (0..65535) OPTIONAL,
    capacityInfoRequested       NULL OPTIONAL,
    genericData                 SEQUENCE OF GenericData OPTIONAL
}

InfoRequestResponse ::= SEQUENCE --(сообщение IRR)
{

    nonStandardData         NonStandardParameter OPTIONAL,
    requestSeqNum           RequestSeqNum,
    endpointType            EndpointType,
    endpointIdentifier      EndpointIdentifier,
    rasAddress              TransportAddress,
    callSignalAddress       SEQUENCE OF TransportAddress,
    endpointAlias           SEQUENCE OF AliasAddress OPTIONAL,
    perCallInfo             SEQUENCE OF SEQUENCE
    {
        nonStandardData         NonStandardParameter OPTIONAL,
        callReferenceValue      CallReferenceValue,
        conferenceID            ConferenceIdentifier,
```

```
        originator              BOOLEAN OPTIONAL,
        audio                   SEQUENCE OF RTPSession OPTIONAL,
        video                   SEQUENCE OF RTPSession OPTIONAL,
        data                    SEQUENCE OF TransportChannelInfo OPTIONAL,
        h245                    TransportChannelInfo,
        callSignalling          TransportChannelInfo,
        callType                CallType,
        bandWidth               BandWidth,
        callModel               CallModel,
        ...,
        callIdentifier          CallIdentifier,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
        substituteConfIDs       SEQUENCE OF ConferenceIdentifier,
        pdu                     SEQUENCE OF SEQUENCE
        {
            h323pdu       H323-UU-PDU,
            sent      BOOLEAN       -- ИСТИНА=передан, ЛОЖЬ=принят
        } OPTIONAL,
        callLinkage             CallLinkage OPTIONAL,
        usageInformation        RasUsageInformation OPTIONAL,
        circuitInfo             CircuitInfo OPTIONAL
    } OPTIONAL,
    ...,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    needResponse            BOOLEAN,
    capacity                CallCapacity OPTIONAL,
    irrStatus               InfoRequestResponseStatus OPTIONAL,
    unsolicited             BOOLEAN,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

InfoRequestResponseStatus ::= CHOICE
{
    complete            NULL,
    incomplete          NULL,
    segment             INTEGER (0..65535),
    invalidCall         NULL,
    ...
}

InfoRequestAck ::= SEQUENCE --(сообщение IACK)
{
    requestSeqNum               RequestSeqNum,
    nonStandardData             NonStandardParameter OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    ...
}

InfoRequestNak ::= SEQUENCE --(сообщение INAK)
{
    requestSeqNum               RequestSeqNum,
    nonStandardData             NonStandardParameter OPTIONAL,
    nakReason                   InfoRequestNakReason,
    altGKInfo                   AltGKInfo OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue         ICV OPTIONAL,
    ...
}
```

```
InfoRequestNakReason ::= CHOICE
{
     notRegistered      NULL,     -- не зарегистрирован в гейткипере
     securityDenial     NULL,
     undefinedReason    NULL,
     ...,
     securityError      SecurityErrors2

}

NonStandardMessage ::= SEQUENCE
{
     requestSeqNum           RequestSeqNum,
     nonStandardData         NonStandardParameter,
     ...,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     featureSet              FeatureSet OPTIONAL,
     genericData             SEQUENCE OF GenericData OPTIONAL
}

UnknownMessageResponse ::= SEQUENCE   -- (сообщение XRS)
{
     requestSeqNum           RequestSeqNum,
     ...,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     messageNotUnderstood    OCTET STRING
}

RequestInProgress ::= SEQUENCE        -- (сообщение RIP)
{
     requestSeqNum           RequestSeqNum,
     nonStandardData         NonStandardParameter    OPTIONAL,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     delay                   INTEGER(1..65535),
     ...
}

ResourcesAvailableIndicate ::= SEQUENCE --(сообщение RAI)
{
     requestSeqNum           RequestSeqNum,
     protocolIdentifier      ProtocolIdentifier,
     nonStandardData         NonStandardParameter OPTIONAL,
     endpointIdentifier      EndpointIdentifier,
     protocols               SEQUENCE OF SupportedProtocols,
     almostOutOfResources    BOOLEAN,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     ...,
     capacity                CallCapacity OPTIONAL,
     genericData             SEQUENCE OF GenericData OPTIONAL
}
```

```
ResourcesAvailableConfirm ::= SEQUENCE --(сообщение RAC)
{
     requestSeqNum           RequestSeqNum,
     protocolIdentifier      ProtocolIdentifier,
     nonStandardData         NonStandardParameter OPTIONAL,
     tokens                  SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue     ICV OPTIONAL,
     ...,
     genericData             SEQUENCE OF GenericData OPTIONAL
}

ServiceControlIndication ::= SEQUENCE --(сообщение SCI)
{
     requestSeqNum           RequestSeqNum,
     nonStandardData         NonStandardParameter OPTIONAL,
     serviceControl          SEQUENCE OF ServiceControlSession,
     endpointIdentifier      EndpointIdentifier OPTIONAL,
     callSpecific SEQUENCE
     {
          callIdentifier     CallIdentifier,
          conferenceID       ConferenceIdentifier,
          answeredCall       BOOLEAN,
          ...
     } OPTIONAL,
     tokens                       SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                 SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue          ICV OPTIONAL,
     featureSet                   FeatureSet OPTIONAL,
     genericData                  SEQUENCE OF GenericData OPTIONAL,
     ...
}

ServiceControlResponse ::= SEQUENCE --(сообщение SCR)
{
     requestSeqNum                     RequestSeqNum,
     result                            CHOICE
     {
          started                      NULL,
          failed                       NULL,
          stopped                      NULL,
          notAvailable                 NULL,
          neededFeatureNotSupported    NULL,
          ...
     } OPTIONAL,
     nonStandardData              NonStandardParameter OPTIONAL,
     tokens                       SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                 SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue          ICV OPTIONAL,
     featureSet                   FeatureSet OPTIONAL,
     genericData                  SEQUENCE OF GenericData OPTIONAL,
     ...
}


END       -- текста ASN.1
```

Приложение I

Пакетирование видео H.263+

В RFC 2429 Группы IETF определен формат полезной нагрузки RTP для битовых видеопотоков, который имеет новые свойства , "H.263+", одобренные в версии 2 (датированной 1998 годом) Рекомендации МСЭ-Т H.263 (включающей в себя свойства с использованием PLUSPTYPE, или Приложения от I/H.263 до T/H.263).

Способность поддерживать формат полезной нагрузки H.263 из RFC 2190 нужна, как указано в Приложении E, для битовых потоков H.263, которые не используют новые свойства версии 2 Рекомендации МСЭ-Т H.263, так как эта поддержка нужна для совместимости с предыдущими реализациями. Однако новый формат полезной нагрузки, определенный в RFC 2429, следует использовать даже для битовых потоков, которые не содержат новых свойств версии 2, если обеспечить, чтобы более новый формат полезной нагрузки был среди возможностей принимающих терминалов.

Добавление I

Алгоритмы RTP/RTCP

Исходный информативный материал можно найти в следующем предложении к Стандарту Интернет:

– SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.) and JACOBSON (V.): RFC 3550, RTP: A Transport Protocol for Real-Time Applications, *Internet Engineering Task Force*, 2003.

Добавление II

Профиль RTP

Исходный информативный материал можно найти в следующем предложении к Стандарту Интернет:

– SCHULZRINNE (H.), CASNER (S.): RFC 3551, RTP Profile for Audio and Video Conferences with Minimal Control, *Internet Engineering Task Force*, 2003.

Добавление III

Пакетирование H.261

Исходный информативный материал можно найти в следующем предложении к Стандарту Интернет:

– TURLETTI (T.), HUITEMA (C.): RFC 2032, RTP Payload Format for H.261 Video Streams, *Internet Engineering Task Force*, 1996.

Добавление IV

Работа H.225.0 при разных стеках протоколов пакетной сети

В настоящем Добавлении даются дополнительные детали, касающиеся работы H.225.0 при различных реальных стеках (наборах) протоколов пакетной сети. Пакетные сети, используемые в этой Рекомендации, должны обеспечивать как надежный, так и ненадежный режимы работы, включая средства для распознавания границ пакетов.

IV.1 Протоколы TCP/IP/UDP

Заметим, что UDP может фрагментировать и вновь собирать большие видеопакеты, но свойство выполнять пакетизацию MB может приводить к потере целого GOB.

Многопунктовую связь с протоколом IP следует использовать для распространения сообщения GRQ вместо вещания на уровне доступа к носителям.

<table>
<tr><th>Приложения для надежной доставки</th><th>Сигнализация о соединении и канал H.245</th></tr>
<tr><td rowspan="2">UDP</td><td>TPKT</td></tr>
<tr><td>TCP</td></tr>
<tr><td colspan="2">IP</td></tr>
<tr><td colspan="2">Уровень звена</td></tr>
<tr><td colspan="2">Физический уровень</td></tr>
</table>

TPKT – это пакетный формат, определенный в RFC 1006 Группы IETF. Он используется для разграничения отдельных сообщений (блоков PDU) внутри потока TCP, который сам дает непрерывный поток октетов без явных границ. TPKT содержит однооктетное поле номера версии, за которым следуют однооктетное зарезервированное поле, двухоктетное поле длины и далее реальные данные. Поле номера версии должно содержать значение "3", а зарезервированное поле должно содержать значение "0". Поле длины должно содержать длину всего пакета, включая поля номера версии, резерва и длины, в виде 16-битового слова с прямым порядком размещения байтов.

IV.1.1 Обнаружение гейткипера

IV.1.1.1 Обнаружение с использованием многопунктового адреса или широко известного порта

Следуя процедурам обнаружения гейткипера и регистрации, описанным в разделе 7/H.323, конечные точки будут использовать при попытке обнаружения гейткипера следующий многопунктовый адрес или широко известный порт в зависимости от конфигурации их сети:

– адрес UDP для многопунктовой связи с гейткиперами: 224.0.1.41,

– порт UDP для многопунктовой связи с гейткиперами: 1718,

– порт UDP для однопунктовой связи RAS, когда нет "другого соглашения": 1719.

Заметим, что "другое соглашение" может охватывать регистрацию конечной точки гейткипером.

Заметим, что при реализациях следует обращать внимание на область применения многопунктовой связи, так чтобы не перегрузить Интернет сообщениями обнаружения.

Предположим, что гейткипер имеет IP-адрес, например, 134.134.12.1, тогда может выполняться следующая сигнализация:

– LRQ или GRQ приходит к 134.134.12.1: порт 1719;

– LRQ или GRQ приходит к 134.134.12.1: порт 1718 (заметим, что это может произойти с гейткиперами версии 1);

– LRQ или GRQ приходит к 224.0.1.41: порт 1718.

Гейткипер может передать сообщение LRQ к следующим адресам:

– 224.0.1.41: порт 1718 (многопунктовая передача ко всем гейткиперам);

– X.X.X.X: порт 1719 (к конкретному гейткиперу).

Порт 1719 следует использовать только в случаях однопунктовой передачи запроса. Это позволяет приемнику знать, следует ли ему передать отказ (xRJ) к передатчику (следует передать во всех случаях).

Порт 1718 следует использовать только в случаях многопунктовой передачи запроса. Приемник ответит, в зависимости от сообщения, подходящим ответом. Для LRQ ответа не требуется, приемник не отвечает на многопунктовые запросы. Для GRQ следует передать направленное GRJ к источнику этого GRQ.

IV.1.1.2 Обнаружение с использованием DNS (для информации)

IV.1.1.2.1 URL для гейткиперов

Для начала отметим, что гейткипер идентифицируется транспортным адресом и gatekeeperIdentifier (идентификатором гейткипера), представляющим собой некоторую цепочку знаков. Любой гейткипер является конкретным ресурсом Интернет, поэтому целесообразно определять его в Унифицированном указателе ресурсов (Uniform Resource Locator, URL). Гейткипер общается с помощью протокола RAS, так что URL для гейткипера может быть следующим:

ras://gkID@domainname.

Здесь gkID – это идентификатор гейткипера, а domainname – это имя домена в DNS, которое определяет домен рассматриваемого гейткипера. Заметим, что это – не обязательно Полностью уточненное доменное имя (Fully Qualified Domain Name, FQDN) с записью типа A: не требуется, чтобы это доменное имя содержало физический транспортный интерфейс с IP-номером, записанным в DNS. Если, однако, именем является FQDN, то целесообразно утверждать, что его IP-адресом является адрес гейткипера, к которому отсылает URL. В этом случае разрешается добавить факультативный номер порта к URL:

ras://gkID@domainname:port_no.

Если не дан номер порта, то берется широко известное значение 1719 в качестве значения "по умолчанию".

Более интересным является случай, когда имя не является FQDN, и поэтому доменное имя не отсылает к транспортному адресу, имеющемуся в DNS. Это доменное имя тогда относится просто к "зоне полномочий гейткипера". В следующем разделе поясняется, как найти гейткипер в этом случае.

IV.1.1.2.2 Отыскание URL

URL не решает проблему местонахождения гейткипера, он просто дает стандартный формат для отыскания информации. Проблема состоит в следующем: как образовать транспортный адрес и gatekeeperIdentifier для сигнализации RAS, задавая доменное имя гейткипера.

Если гейткипер имеет идентификатор, соответствующий IETF RFC 822, то легко извлечь доменное имя из этого идентификатора гейткипера, соответствующего IETF RFC 822. В самом деле, может быть удобным дать идентификаторы, соответствующие IETF RFC 822, конечным точкам, а затем согласиться, что часть "доменное имя" в таком идентификаторе относится к домену гейткипера.

IV.1.1.2.2.1 Запрос записи о ресурсе в SRV

Первое решение использует тот факт, что гейткипер является, по сути, системной службой, а транспортный адрес поименованной системной службы можно выделить из DNS, используя запрос к новому типу записи ресурсов в DNS, который назван SRV (от SeRVice location record, запись местонахождения службы). Если задать доменное имя, то будет сделан запрос записи SRV

о транспортном адресе службы RAS для этого домена. Доменное имя само или то, которое выдано в ответе SRV, используется в качестве gatekeeperIdentifier. Запись SRV и ее использование определены в IETF RFC 2782.

IV.1.1.2.2.2 Запрос записи TXT

Все имеющиеся реализации DNS поддерживают текстовую запись (TXT) о ресурсах. По сути, она является некоторым свободным текстом, который может быть выдан для каждого доменного имени. Для одного домена возможно записать много TXT-ресурсов. Стандарт предусматривает, что будут выдаваться все записи TXT, когда к ним сделан запрос.

Если запросы SRV оказались неудачными, то возможно использование запросов TXT. Примем некоторые соглашения о выделении доменного имени, которые были предложены выше. Для gatekeeperIdentifier можно использовать либо цепочки, соответствующие IETF RFC 822 (похожие на email-имена), либо цепочки, соответствующие IETF RFC 1768 (имена URL). В любом случае используется доменное имя, чтобы сделать TXT-запрос в DNS для этого доменного имени. Выданные записи о ресурсах образуют строки свободного текста, а терминал будет затем искать в ответе строки следующей формы:

ras [< gk id>@]<domain name >[:<portno>] [<priority>].

Поле **<gk id>** является факультативным идентификатором гейткипера, который выделен из доменного имени. Если это поле отсутствует, то предполагается, что сам домен будет идентификатором гейткипера.

Поле **<domain name>** может быть либо именем A-записи, которое содержит IP-адрес гейткипера, либо исходным IP-адресом в форме с точками. Доменное имя не обязательно должно быть полностью уточненным; если это не так, то для формирования полностью уточненного имени A-записи следует к исходному IP-адресу добавить субдомен, в котором была найдена TXT-запись.

Факультативное поле **[:<portno>]** может использоваться для определения номера порта, отличающегося от стандартного порта RAS.

Факультативное поле **[<priority>]** указывает порядок, в котором следует выполнять доступ к перечисленным гейткиперам для обнаружения или посылки запросов LRQ, если имеется больше одной TXT-записи RAS. Меньшие номера имеют более высокий приоритет.

Заметим, что при этом формате, если поле **<gk id>** отсутствует, предполагается, что идентификаторы гейткиперов фактически являются законными доменными именами. Однако, если в одной хост-машине необходимо поддерживать несколько логических гейткиперов, каждый со своим идентификатором, то формат будет это обеспечивать. Это возможно благодаря тому, что разные A-записи могут содержать один и тот же IP-адрес.

Белые интервалы используются в качестве разделителей между **ras** и **gk id,** если он имеется, или **domain name**, а также между **porto** и **priority**. Белые интервалы содержат некоторое количество пробелов или шагов табуляции.

Примеры правильных TXT-записей для гейткипера:

– ras gk1

– ras gk1.company.com

– ras gk1:1500 3

– ras 172.11.22.33:1500 2.

Клиент анализирует выданные строки и из них получает транспортный адрес гейткипера внутри этого домена, к которому он может передавать сообщения RAS.

Так как DNS требует от сервера выдавать все TXT-записи, связанные с каким-либо доменным именем, клиент может отфильтровывать и обрабатывать только те записи, которые полезны для него. Разрешается для DNS также выдавать упорядоченный перечень гейткиперов, которые могут служить альтернативами и резервами, как определено в Рекомендации МСЭ-Т H.323.

Заметим, что сервер, указанный в таком запросе, может быть реальным транспортным адресом в десятичном обозначении с точками или он может быть FQDN, который сам требует запроса A-записи в DNS для определения транспортного адреса. Преимуществом использования FQDN является

обычное скрытие фактических IP-номеров. Преимуществом использования IP-номеров является избежание второго запроса в DNS, что сокращает это время перед установлением соединения.

IV.1.1.2.3 Обработка гейткипером email-идентификаторов при сообщениях ARQ и LRQ

Когда поле **destinationInfo** в сообщении ARQ или LRQ содержит адрес-псевдоним **email-ID,** гейткипер сначала проверяет свою базу данных регистрации на наличие этого псевдонима. Если он не может быть найден, то гейткипер анализирует псевдоним для восстановления его доменной части. Если домен не выдан, то гейткипер может генерировать домен "по умолчанию". Этот домен затем используется для нахождения одного или нескольких гейткиперов согласно процедурам из IV.1.1.2.2. Затем гейткипер может запросить все гейткиперы, найденные таким способом, об обмене сообщениями LRQ/LCF/LRJ.

Заметим, что более одного гейткипера могут иметь соответствующие TXT-записи в одном домене DNS. Следовательно, один домен DNS может "содержать" более одной зоны H.323. Поэтому даже если какой-либо гейткипер не может разложить email-ID, чья доменная часть является частью его доменов "по умолчанию", то он может тем не менее запросить другие зоны в том же домене DNS.

Если гейткиперу предоставлен незарегистрированный псевдоним, который является **h323-id,** причем этот идентификатор может быть признан допустимой пользовательской частью имени по IETF RFC 822, то гейткипер может признать, что этот псевдоним был email-ID в его домене "по умолчанию" и попытаться найти местонахождение этого псевдонима в каком-либо другом гейткипере. Аналогично, email-ID из входящего LRQ может лишиться в гейткипере своего доменного имени, так что он может быть размещен в качестве h323-ID.

IV.1.2 Связь от конечной точки к конечной точке

Конечные точки, которые желают получать вызовы от конечных точек, не входящих в зону их гейткипера, будут использовать следующий порт для канала сигнализации о соединении:

– TCP-порт сигнализации о соединении в конечной точке 1720.

Несмотря на то, что разрешается для этих портов использовать динамические значения, допускающие существование нескольких конечных точек в одном устройстве, необходимо понимать, что это будет препятствовать взаимодействию с конечными точками, не входящими в зону гейткипера, за исключением случаев работы через шлюз этой зоны.

IV.2 Протоколы SPX/IPX

Заметим, что в такой сети отсутствует сборка больших пакетов, поэтому использование фрагментации MB обязательно.

Приложения для ненадежной доставки	**Канал сигнализации о соединении для канала H.245**
PXP	SPX
IPX	
Уровень звена	
Физический уровень	

IV.2.1 Обнаружение гейткипера

В терминологии протокола IPX "разъем" (socket) эквивалентен "порту" в IP и "идентификатору TSAP" в этой Рекомендации и в Рекомендации МСЭ-Т H.323.

В сетях на базе протокола IPX (Intermetwork Packet exchange, межсетевой пакетный обмен) гейткиперы объявляют "тип обслуживания в гейткипере", определяемый ниже, чтобы позволять конечным точкам определять их местонахождение в сети. При этом конечные точки будут запрашивать "тип обслуживания в гейткипере" для определения местонахождения ближайшего гейткипера.

– Тип обслуживания в гейткипере FFS.

ПРИМЕЧАНИЕ. – Типом обслуживания (service type) называется разъем SAP в некоторых документах по IPX.

IV.2.2 Связь от конечной точки к конечной точке

Конечные точки, которые желают получать вызовы от конечных точек, не входящих в зону их гейткипера, будут использовать следующие разъемы для сигнализации о соединении:

– IPX-порт сигнализации о соединении в конечной точке FFS.

Несмотря на то что разрешается для этих разъемов использовать динамические значения, допускающие существование нескольких конечных точек в одном устройстве, необходимо понимать, что это будет препятствовать взаимодействию с конечными точками, не входящими в зону гейткипера, за исключением случаев работы через шлюз этой зоны.

IV.3 Протокол SCTP

Стек протоколов H.323 над протоколом SCTP (Stream Control Transmission Protocol, протокол передачи управления потоком) выглядит следующим образом:

Приложения для ненадежной доставки	**Сигнализация о соединении с управлением туннелированным соединением**
UDP	SCTP
IP	
Уровень звена	
Физический уровень	

Каждое сообщение сигнализации о соединении H.225.0 должно переноситься в отдельной порции данных SCTP. Не должны добавляться заголовки (то есть нет TPKT). Должна быть задана упорядоченная доставка.

IV.3.1 Потоки

Все сообщения одного соединения должны использовать один и тот же поток SCTP. В реализациях могут применяться различные потоки для разных соединений.

IV.3.2 Идентификаторы протокола полезной нагрузки

SCTP может использоваться с неопределенным идентификатором протокола полезной нагрузки (0) или с номером 13, присвоенным IANA (Internet Assigned Numbers Authority, Агентство по выделению номеров Интернет).

Добавление V

Использование ASN.1 в этой Рекомендации

В настоящем Добавлении перечисляются соглашения по ASN.1, которые использованы в этой Рекомендации. В будущих изменениях этой Рекомендации следует использовать только эти конструкции. Дополнительные конструкции ASN.1 будут приниматься во внимание только в исключительных обстоятельствах.

V.1 Тегирование

Все маркеры (теги) в этой Рекомендации являются AUTOMATIC TAGS.

V.2 Типы

В этой Рекомендации в определениях на языке ASN.1 могут присутствовать следующие типы.

BIT STRING (цепочка (строка) битов)	IA5String (цепочка MA5)	OCTET STRING (цепочка октетов)
BMPString (цепочка BMP (Basic Multilingnal Plene)	INTEGER (целочисленный)	SEQUENCE (последовательность)
BOOLEAN (булев)	NULL (вырожденный)	SEQUENCE OF (последовательность – из)
CHOICE (выборочный)	NumericString (числовая цепочка)	SET (множество)
GeneralString (общая цепочка)	OBJECT IDENTIFIER (идентификатор объекта)	SET OF (множество – из)

V.3 Ограничения и диапазоны

В этой Рекомендации используются ограничения на размер ("SIZE") для цепочек, SET OF и SEQUENCE OF, ограничения диапазона значений для целых чисел и разрешенных алфавитов ("FROM").

V.4 Расширяемость

В этой Рекомендации используется маркер расширения (многоточие "...").

Добавление VI

Идентификаторы туннелированных протоколов сигнализации в H.225.0

В этой Рекомендации поддерживается описанное в 10.4/H.323 туннелирование не определенных в H.323 протоколов сигнализации о соединении. В серии Приложений M/H.323 (M.1/H.323, M.2/H.323 и т. д.) определено туннелирование для конкретных протоколов. В настоящей Рекомендации туннелированный протокол определяется информацией из структуры ASN.1 **TunnelledProtocol,** описанной в 7.6 и в Приложении H. В этом Добавлении дается список идентификаторов **TunnelledProtocol,** которые были присвоены конкретным туннелированным протоколам.

Туннелированные протоколы, определенные для настоящей Рекомендации, приведены в таблицах VI.1 и VI.2. Заметим, что туннелирование не ограничено протоколами, перечисленными в этих таблицах.

Таблица VI.1/H.225.0 – Туннелированные протоколы, определенные tunnelledProtocolObjectID

Спецификация туннелирования	Спецификация протокола	tunnelledProtocolObjectID	subIdentifier
M.1/H.323	ISO/IEC 11572 and 11582	{iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9)}	(Нет)
M.2/H.323	Рек. МСЭ-T Q.763 (1988)	{itu-t (0) recommendation (0) q (17) 763}	"1988"
M.2/H.323	Рек. МСЭ-T Q.763 (1993)	{itu-t (0) recommendation (0) q (17) 763}	"1993"

Таблица VI.2/H.225.0 – Туннелированные протоколы, определенные TunnelledProtocolAlternateIdentifier

Спецификация туннелирования	Спецификация протокола	protocolType	protocolVariant	subIdentifier
M.2/H.323	ANSI T1.113-1988	"isup"	"ANSI T1.113-1988"	"1988"
M.2/H.323	ETS 300 121	"isup"	"ETS 300 121"	"121"
M.2/H.323	ETS 300 356	"isup"	"ETS 300 356"	"356"
M.2/H.323	BELLCORE GR-317	"isup"	"BELLCORE GR-317"	"317"
M.2/H.323	JT-Q761-4(1987-1992)	"isup"	"JT-Q761-4(1987-1992)"	"87"
M.2/H.323	JT-Q761-4(1993)	"isup"	"JT-Q761-4(1993)"	"93"

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия A	Организация работы МСЭ-Т
Серия B	Средства выражения: определения, символы, классификация
Серия C	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	**Аудиовизуальные и мультимедийные системы**
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	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
Серия Z	Языки программирования

Отпечатано в Швейцарии
Женева, 2004 г.