



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

J.181

(06/2004)

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

Цифровая передача телевизионных сигналов

**Сообщение с меткой о вставке цифровой
программы для систем кабельного
телевидения**

Рекомендация МСЭ-Т J.181

Рекомендация МСЭ-Т J.181

Сообщение с меткой о вставке цифровой программы для систем кабельного телевидения

Резюме

В настоящей Рекомендации поддерживается монтаж транспортных потоков MPEG-2 в целях вставки цифровых программ, включая вставку рекламы и содержательного материала других типов. Для сообщения о возможностях монтажа и вставки определяется внутривидеокадровый механизм обмена сообщениями. Описывается метод передачи уведомления о последующих монтажных точках в транспортном потоке.

Источник

Рекомендация МСЭ-Т J.181 утверждена 29 июня 2004 года 9-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

ПРЕДИСЛОВИЕ

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

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

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

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

ПРИМЕЧАНИЕ

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

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

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

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

© ITU 2005

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

СОДЕРЖАНИЕ

	Стр.
1 Область применения	1
2 Ссылки	1
2.1 Нормативные ссылки	1
2.2 Информативные ссылки	1
3 Определение терминов	2
4 Сокращения	3
5 Введение.....	4
5.1 Монтажные точки (информативные).....	4
5.2 Монтажные программные точки (информативные).....	4
5.3 События монтажа (информативные)	5
5.4 Выбор идентификатора PID.....	5
5.5 Поток сообщений (информативный)	6
6 Дескрипторы таблицы PMT	6
6.1 Регистрационный дескриптор	6
6.2 Дескриптор идентификатора метки	7
6.3 Дескриптор идентификатора потока.....	8
7 Таблица с монтажной информацией	9
7.1 Обзор.....	9
7.2 Информационный сегмент монтажа	10
7.3 Команды монтажа.....	13
7.4 Время	18
7.5 Ограничения.....	19
8 Дескрипторы монтажа	21
8.1 Обзор.....	21
8.2 Дескриптор монтажа	21
8.3 Конкретные дескрипторы монтажа.....	22
9 Шифрование	27
9.1 Обзор.....	27
9.2 Шифрование с фиксированным ключом	27
9.3 Алгоритмы шифрования	27

Сообщение с меткой о вставке цифровой программы для систем кабельного телевидения

1 Область применения

В настоящей Рекомендации поддерживается монтаж потоков MPEG-2 в целях вставки цифровых программ, включая вставку рекламы и содержательного материала других типов. Для сообщения о возможностях монтажа и вставки определяется внутривидеокадровый механизм обмена сообщениями; этот механизм не предназначен для обеспечения "бесшовного" монтажа. Сама по себе настоящая Рекомендация не содержит описания используемого метода монтажа или ограничений относительно монтируемых потоков, а также не рассматривает ограничения, налагаемые на монтажные устройства. В настоящей Рекомендации поддерживается также точная сигнализация о событиях, имеющих место в потоке.

Предполагается наличие транспортного потока (либо многопрограммного, либо однопрограммного транспортного потока), полностью соответствующего стандарту MPEG-2. На этот поток не налагаются никакие ограничения, кроме включения определенных сообщений с меткой.

В настоящей Рекомендации описывается метод передачи уведомления о последующих монтажных точках и прочей информации синхронизации в транспортном потоке. Определяется таблица с монтажной информацией для уведомления устройств в нисходящем направлении передачи о событиях монтажа, таких как перерыв в работе сети и возврат из состояния перерыва. Таблица с монтажной информацией, относящаяся к данной программе, переносится в одном или нескольких идентификаторах PID, на которые ссылается таблица структуры программы (PMT) для этой программы. Таким образом, уведомление о событиях монтажа может проходить через большинство ремультимплексов транспортных потоков без необходимости в специальной обработке.

2 Ссылки

2.1 Нормативные ссылки

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

- ITU-T Recommendation H.222.0 (2000) | ISO/IEC 13818-1:2000, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.
- Рекомендация МСЭ-Т H.262 (2000) | ИСО/МЭК 13818-2:2000, *Информационные технологии – Обобщенное кодирование движущихся изображений и связанной информации аудио: Видео*.
- ISO/IEC 13818-4:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 4: Conformance testing, plus Corrigendum 2 (1998)*.
- FIPS PUB 46-3-1999, *Data Encryption Standard (DES)*.
- FIPS PUB 81-1980, *DES Modes of Operation*.

2.2 Информативные ссылки

- SMPTE 312M, SMPTE Standard for Television – Splice Points for MPEG-2 Transport Streams.

3 Определение терминов

Во всем тексте настоящей Рекомендации приведенные ниже термины имеют специфические значения. Поскольку некоторые из терминов, которые определены в стандарте ИСО/МЭК 13818, имеют узкоспециальные технические значения, для получения их определений читатель должен обратиться к оригинальному источнику. Для терминов, используемых в настоящей Рекомендации, ниже приведены краткие определения.

3.1 блок доступа: Закодированное изображение компонента представления (см. в Рекомендации МСЭ-Т Н.262 | ИСО/МЭК 13818-2).

3.2 аналоговый тональный сигнал метки: Сигнал в аналоговой системе, который обычно является либо последовательностью двухтональных многочастотных (DTMF) сигналов, либо сигналом замыкания контакта, который означает для аппаратуры вставки рекламы, что время для рекламного блока начнется или закончится в ближайшее время.

3.3 время для рекламного блока; коммерческое время: Интервал времени, предоставляемый операторам кабельного телевидения службами программ кабельного телевидения во время программы для использования оператором CATV; обычно это время продается местным рекламодателям или используется для собственной рекламы канала.

3.4 перерыв: Время передачи рекламного блока или процесс фактической вставки программы.

3.5 режим монтажа компонентов: Режим сообщения с меткой, при котором бит поля `program_splice_flag` устанавливается на "0" и указывает на то, что каждый PID/компонент, предназначенный для монтажа, будет перечисляться отдельно в последующем синтаксисе. Компоненты, не перечисленные в сообщении, не подлежат монтажу.

3.6 сообщение с меткой: См. "сообщение".

3.7 событие: Событие монтажа или событие просмотра.

3.8 точка входа: Точка в потоке, пригодная для входа, которая находится на границе элементарного компонента представления. Фактически точка входа находится между двумя компонентами представления, а не входит в сам компонент представления.

3.9 внутрипотокное устройство: Устройство, непосредственно принимающее транспортный поток и способное извлекать информацию синхронизации непосредственно из транспортного потока.

3.10 сообщение: В контексте настоящей Рекомендации сообщение – это содержимое любого поля `splice_info_section`.

3.11 многопрограммный транспортный поток (MPTS): Транспортный поток со множеством программ.

3.12 внепотокное устройство: Устройство, принимающее сообщение с меткой от внутрипотокного устройства по отдельному логическому соединению от транспортного потока. Внепотокное устройство непосредственно не принимает и не передает транспортный поток.

3.13 точка выхода: Точка в потоке, пригодная для выхода, которая находится на границе элементарного компонента представления. Фактически точка выхода находится между двумя компонентами представления, а не входит в сам компонент представления.

3.14 поле `payload_unit_start_indicator`: Бит в заголовке транспортного пакета, который, среди прочего, указывает на то, что информационный сегмент начинается в последующей полезной нагрузке (см. Рекомендацию МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1).

3.15 идентификатор пакета (PID): Однозначно определяемое 13-битовое значение, используемое для идентификации типа данных, хранящихся в полезной нагрузке пакета (см. Рекомендацию МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1).

3.16 PID-поток: Все пакеты с одним и тем же идентификатором PID в транспортном потоке.

3.17 поле `pointer_field`: Первый байт полезной нагрузки транспортного пакета, необходимый в случае, когда информационный сегмент начинается в этом пакете (см. Рекомендацию МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1).

3.18 время представления: Время, в течение которого компонент представления присутствует в декодере приемника системы (см. Рекомендацию МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1).

- 3.19 компонент представления:** Декодированный блок аудиодоступа или декодированное изображение (см. Рекомендацию МСЭ-Т H.262 | ИСО/МЭК 13818-2).
- 3.20 программа:** Совокупность PID-поток видео, аудио и данных, совместно использующих общий номер программы в потоке MPTS (см. Рекомендацию МСЭ-Т H.222.0 | ИСО/МЭК 13818-1).
- 3.21 программная точка входа:** Группа точек входа PID-потока, взаимодействующих во время представления.
- 3.22 программная точка выхода:** Группа точек выхода PID-потока, взаимодействующих во время представления.
- 3.23 режим монтажа программы:** Режим сообщения с меткой, при котором бит поля `program_splice_flag` устанавливается в значение "1" и указывает на то, что сообщение относится к монтажной точке программы и что все PID/компоненты программы подлежат монтажу.
- 3.24 программная монтажная точка:** Программная точка входа или программная точка выхода.
- 3.25 приемное устройство:** Устройство, которое принимает или интерпретирует информационные сегменты, соответствующие настоящей Рекомендации. Примерами таких устройств могут служить сплайсеры (монтажные устройства), рекламные серверы, устройства сегментации и спутниковые приемники.
- 3.26 регистрационный дескриптор:** Переносится в PMT программы и указывает на то, что при передаче сигналов о событиях монтажа поля `splice_info_section` должны переноситься в PID-потоке в пределах данной программы. Наличие регистрационного дескриптора означает соответствие программы настоящей Рекомендации.
- 3.27 зарезервировано:** Термин "зарезервировано", когда он используется в пунктах, определяющих кодированный битовый поток, указывает на то, что это значение может быть использовано в будущем для расширения настоящей Рекомендации. Всем зарезервированным битам должно быть присвоено значение "1", если в настоящей Рекомендации не оговорено иначе, и это поле не должно приниматься во внимание приемной аппаратурой.
- 3.28 однопрограммный транспортный поток (SPTS):** Транспортный поток, содержащий одну программу стандарта MPEG.
- 3.29 событие монтажа:** Возможность монтажа одного или нескольких PID-поток.
- 3.30 режим немедленного монтажа:** Режим сообщения с меткой, при котором монтажное устройство выбирает по таблице `splice_info_table` ближайшую возможность для монтажа в потоке. Вне этого режима это сообщение задает значение поля `pts_time`, которое при модификации с помощью поля `pts_adjustment` задает время представления для назначенного момента монтажа.
- 3.31 монтажная точка:** Точка в PID-потоке, являющаяся либо точкой выхода, либо точкой входа.
- 3.32 событие просмотра:** Телевизионная программа или некоторый объем сжатого материала в рамках какой-либо службы; в противоположность событию монтажа, являющемуся моментом времени.

4 Сокращения

В настоящей Рекомендации используются следующие сокращения:

ATSC	Комитет по перспективным системам телевидения
bslbf	Битовая строка, левый бит обрабатывается первым, где "левый" означает порядок записи битовых строк
DVB	Цифровое телевизионное вещание системы DVB
MPTS	Многопрограммный транспортный поток
PMT	Таблица структуры программы (см. Рекомендацию МСЭ-Т H.222.0 ИСО/МЭК 13818-1)
PTS	Метка времени представления (см. Рекомендацию МСЭ-Т H.222.0 ИСО/МЭК 13818-1)
grchof	Коэффициенты остаточного многочлена; самый старший коэффициент обрабатывается первым

SPTS	Однопрограммный транспортный поток
STC	Системный таймер
uimsbf	Целое число без знака, старший бит обрабатывается первым

5 Введение

5.1 Монтажные точки (информативные)

Для обеспечения возможности монтажа сжатых битовых потоков в настоящей Рекомендации определены монтажные точки. Монтажные точки в транспортном потоке стандарта MPEG-2 предоставляют возможности для переключения элементарных потоков от одного источника к другому. Они указывают на место переключения или на место в битовом потоке, где может быть выполнено переключение. Монтаж в таких точках может или не может приводить к хорошему качеству видео- и аудиоинформации. Это определяется рабочими характеристиками монтажного устройства.

Транспортные потоки создаются путем мультиплексирования PID-потоков. В настоящей Рекомендации определены два типа монтажных точек для PID-потоков: точки выхода и точки входа. Точки входа – это места в битовых потоках, приемлемые для входа с точки зрения монтажа. Точки выхода – это места, приемлемые для выхода из битового потока. Определено группирование точек входа отдельных PID-потоков в программные точки входа с целью переключения целых программ (видео с аудио). Также определены программные точки выхода для выхода из программы.

Точки выхода и точки входа – это условные точки в битовом потоке, находящиеся между двумя элементарными компонентами представления потока. Точки выхода и точки входа необязательно синхронизированы с транспортными пакетами и пакетами потока PES. Точка выхода и точка входа могут быть расположены в одном месте; другими словами, граница одного компонента представления может служить как надежное место и для выхода из битового потока, и для входа в этот поток.

В результате операции простого переключения будут получены данные блока доступа одного потока вплоть до их точки выхода, за которой следуют данные из другого потока, начинающиеся с первого блока доступа вслед за точкой входа. Могут существовать более сложные операции монтажа, при которых монтажное устройство модифицирует данные перед точкой выхода или данные после точки входа. Монтажные устройства могут также вводить данные между точкой выхода одного потока и точкой входа другого потока. В настоящей Рекомендации поведение монтажных устройств не будет ни задаваться, ни ограничиваться.

5.2 Монтажные программные точки (информативные)

Программные точки входа и программные точки выхода – это множества точек входа или точек выхода PID-потока, которые взаимодействуют во время представления.

И хотя монтажные точки в программной монтажной точке согласуются во времени представления, они обычно не появляются в транспортном потоке близко одна к другой. Поскольку сжатая видеоинформация требует намного больше времени для декодирования, чем аудиоинформация, то монтажные точки аудиоинформации могут отставать от монтажных точек видеоинформации по времени на сотни миллисекунд и по количеству, которое может меняться во время программы.

В настоящей Рекомендации определены два способа сигнализации относительно того, какие монтажные точки в программе подлежат монтажу. Бит поля `program_splice_flag`, когда он установлен в значение "true" (истина), указывает на то, что режим монтажа программы активен и что все идентификаторы PID программы могут быть смонтированы (идентификатор PID таблицы с монтажной информацией является исключением; монтаж или прохождение этих сообщений выходят за рамки настоящей Рекомендации). Бит поля `program_splice_flag`, когда он установлен в значение "false" (ложь), указывает на то, что режим монтажа компонентов активен и что сообщение будет однозначно определять, какие идентификаторы PID подлежат монтажу, и может задавать однозначно определяемое время монтажа для каждого. Это требуется для того, чтобы указывать монтажному устройству, нужно ли монтировать или нет различные типы неопределенных данных, а также видеоинформацию и аудиоинформацию.

Хотя в настоящей Рекомендации допускается возможность задавать однозначно определяемое время монтажа для каждого компонента программы, ожидается, что в большинстве сообщений режима монтажа компонентов для всех компонентов будет использоваться одно время монтажа (время монтажа по умолчанию), как описано в п. 7. Когда один или несколько компонентов значительно отличаются от других компонентов в одном и том же сообщении по времени пуска или останова, то в этом случае предусматривается использование средств для факультативного определения отдельного времени монтажа для каждого компонента. Примером может служить загружаемая прикладная мини-

программа, которая должна поступить в декодер каналов кабельного телевидения за несколько секунд до рекламы.

5.3 События монтажа (информативные)

В настоящей Рекомендации приводится метод внутриволновой сигнализации о событиях монтажа с использованием команд монтажа, посылаемых монтажному оборудованию в нисходящем направлении передачи. При сигнализации о событии монтажа определяется, какая монтажная точка в потоке должна использоваться для монтажа. При получении сигнала о событии монтажное устройство может решать, выполнять или не выполнять монтаж (событие, о котором принят сигнал, должно интерпретироваться как возможность монтажа, а не как команда). Таблица с монтажной информацией содержит извещение о возможностях для события монтажа. Каждое событие монтажа, о котором передан сигнал, подобно аналоговому тональному сигналу метки. В таблице с монтажной информацией содержатся функции тональных меток, которые расширяются с целью предварительной диспетчеризации событий монтажа.

В настоящей Рекомендации устанавливается, что таблица с монтажной информацией для каждой программы переносится в одном или нескольких PID-потоках с обозначенным полем `stream_type`. Идентификаторы PID монтажной информации программы обозначены в таблице структуры программы (PMT) для этой программы. Таким образом, таблица с монтажной информацией включается программой при прохождении через операции ремultipлексирования. Общий тип потока (`stream_type`) определяет все PID-потоки, которые переносят таблицы с монтажной информацией. Ремultipлексоры или сплайсеры могут использовать это поле `stream_type`, чтобы проигнорировать монтажную информацию перед передачей транспортного потока устройству конечного пользователя.

Аппаратура ввода меток может периодически посылать сообщения в моменты, не являющиеся монтажными точками; эти сообщения используются в качестве контрольных сообщений, которые помогают гарантировать функционирование системы на должном уровне. Это может выполняться периодически путем выдачи сообщений `splice_null()` или путем передачи зашифрованных сообщений `splice_insert`, генерируемых с использованием нераспределенного ключа. Поскольку в настоящее время по типовой сети метки посылаются два раза в час, то средний интервал в 5 минут (интервал периодичности) может быть приемлемым интервалом. Если сообщение не было принято в течение 10-минутного интервала, то приемное устройство может посылать оператору тревожный сигнал о возможности нарушения работы системы (такое поведение зависит от того, кто реализует систему).

5.4 Выбор идентификатора PID

5.4.1 Выбор идентификатора PID (нормативный)

Монтажная информация может переноситься в нескольких идентификаторах PID. Максимальное число идентификаторов PID, которые могут переносить монтажную информацию, не должно быть больше 8. Эти идентификаторы PID могут быть либо в исходном состоянии (когда транспортные биты управления скремблированием (поле `scrambling_control`) установлены в "00"), либо скремблированы системой СА. Каждый идентификатор PID сообщения с меткой может включать дескриптор идентификатора метки (`cue identifier descriptor`), определенный в п. 6.2, для описания команд монтажа, включенных в PID. Когда для переноса монтажной информации используется несколько идентификаторов PID, то первый PID сообщения с меткой в таблице структуры программы должен содержать только типы команд монтажа: 0x00 (нулевой монтаж – `splice_null`), 0x04 (расписание монтажа – `splice_schedule`) и 0x05 (ввод монтажа – `splice_insert`). Кроме того, во всех идентификаторах PID монтажной информации в пределах данной программы должен быть однозначно определен идентификатор `splice_event_id`.

5.4.2 Выбор идентификатора PID (информативный)

Хотя использование нескольких идентификаторов PID сообщений с меткой допустимо на практике, следует отметить, что не все оборудование может реагировать одинаково на поток, содержащий несколько идентификаторов PID сообщений с меткой. Для некоторого оборудования могут иметься ограничения на число идентификаторов PID, которые могут быть переданы дальше или приняты этим оборудованием. Если в системе используется несколько идентификаторов PID, проходящих через различные устройства, с тем чтобы достичь декодера, предлагается выполнять тщательное сквозное тестирование.

Во многих системах нежелательна доставка идентификаторов PID, которые переносят монтажную информацию вне оборудования вставки рекламы в головном узле. В этих системах устройство монтажа или мультимплексирования будет игнорировать любое или все эти сообщения (идентификаторы PID), так что они не будут доставляться в декодер. В других системах для активизации функций декодера каналов кабельного телевидения к нему могут выборочно пропускаться определенные идентификаторы PID. Третья возможность состоит в том, что устройство монтажа или мультимплексирования будет объединять несколько идентификаторов PID, которые

переносят монтажную информацию, в один идентификатор PID для обработки выходной информации декодера в нисходящем направлении передачи с множеством идентификаторов PID. Рекомендуется, чтобы действие по игнорированию или передаче сообщения обеспечивалось пользователем при соответствующем поведении системы по умолчанию, выбираемом тем, кто реализует систему.

Если устройство монтажа или мультиплексирования принимает идентификатор PID, основанный на настоящей Рекомендации, с битами скремблирования, установленными в заголовке, то при операции по умолчанию этот PID должен игнорироваться и не пропускаться на выход. В идеальном случае это должна быть операция, обеспечиваемая пользователем, поскольку в ряде случаев этот идентификатор PID может быть дескремблирован устройством в нисходящем направлении передачи.

Доставка сообщений пользователю вне места приема может быть основана на соглашениях коммерческого характера. Примером может служить случай, когда один обладатель программ хочет, чтобы сообщения с меткой пропускались к декодерам каналов кабельного телевидения для активизации целевого метода объявлений, в то время как другой обладатель программ настаивает на том, чтобы эти сообщения игнорировались, чтобы гарантировать, что коммерческий злоумышленник не сможет использовать эти сообщения.

Когда в таблице PMT определено несколько идентификаторов PID монтажа, монтажное устройство должно обрабатывать все эти PID. Если используется `cue_identifier_descriptor`, устройство монтажа или мультиплексирования может использовать эту информацию для более избирательного подхода к выбору идентификатора PID, согласно которому оно будет действовать.

Некоторыми возможными причинами для использования нескольких идентификаторов PID для этого сообщения может быть выборочная доставка сообщений с меткой для различных уровней рекламы или для разграничения сообщений с меткой и сообщений сегментации. Хотя одним из возможных методов обработки этих сообщений является использование методов шифрования, заложенных в настоящей Рекомендации, многие механизмы доставки могут поддерживать условную доставку с помощью идентификатора PID в защищенном виде. Аппаратура доставки (спутниковые передатчик/приемник, ремультимплексор) может фильтровать поток, чтобы позволить пропустить внутри потока только один или малое число идентификаторов PID. Этот метод может быть использован для создания нескольких программ в линии передачи на основе наименований. Решение об использовании одного или нескольких идентификаторов PID будет основано на требованиях безопасности и имеющейся кабельной аппаратуре в системе.

5.5 Поток сообщений (информативный)

Сообщения, описываемые в настоящей Рекомендации, могут исходить из множества источников. Эти сообщения предназначены для внутрипоточковой передачи к устройствам в нисходящем направлении. Такие устройства могут предпринимать действия по этим сообщениям или посылать их внепоточковому устройству для обработки. Примером может служить сплайсер, взаимодействующий с сервером объявлений по протоколу SCTE 30. Внутрипоточковые устройства могут посылать эти сообщения следующему устройству в цепи передачи или же, как вариант, игнорировать эти сообщения. Тех, кто будет реализовывать систему, настоятельно просят обеспечить, чтобы эти решения принимались пользователем, а не были произвольным образом жестко закодированы в оборудовании.

Любое устройство, которое повторно регистрирует сообщения `pcr/pts/dts` и посылает эти сообщения с меткой устройству в нисходящем направлении передачи, должно модифицировать поле `pts_time` или поле `pts_adjustment` в сообщении во всех идентификаторах PID, соответствующих настоящей Рекомендации. Модификация поля `pts_adjustment` предпочтительна, поскольку регистрирующее устройство не будет иметь информацию о поле `pts_time`, которое может содержаться в нескольких командах (и, возможно, в будущих командах).

Сообщение `bandwidth_reservation()` (резервирование ширины полосы пропускания) предназначается в качестве сообщения, используемого на замкнутом тракте от исходящей спутниковой системы (кодера) до приемника. Предполагается также, что это сообщение будет проигнорировано (заменено пакетом НУЛЬ) приемником, но это необязательно. Если это сообщение должно достичь внутрипоточкового устройства (например, сплайсера), оно не должно перенаправляться к внепоточковому устройству (например, серверу объявлений) и может быть либо проигнорировано, либо передано дальше внутрипоточковым устройством. Рекомендуется, чтобы действие по игнорированию или передаче сообщения обеспечивалось пользователем при соответствующем поведении системы по умолчанию, выбираемом тем, кто реализует систему.

6 Дескрипторы таблицы PMT

6.1 Регистрационный дескриптор

Регистрационный дескриптор (Рекомендация МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, таблица 2-45 – Регистрационный дескриптор, п. 2.6.8) служит для однозначного определения программ,

соответствующих настоящей Рекомендации. Регистрационный дескриптор переносится в цикле `program_info` (программная информация) таблицы PMT для каждой программы, соответствующей настоящей Рекомендации. Этот дескриптор должен иметься во всех таблицах PMT всех соответствующих программ при мультиплексировании. Наличие регистрационного дескриптора также указывает на то, что при сигнализации о событиях монтажа информационные сегменты `splice_info_section` должны переноситься в одном или нескольких PID-потоках в пределах данной программы.

Наличие этого регистрационного дескриптора в таблице PMT говорит о следующем:

- 1) В число программных элементов не входит таблица с монтажной информацией, определяемая стандартом SMPTE 312M.
- 2) Единственными дескрипторами, которые могут присутствовать в цикле `ES_descriptor_loop` таблицы PMT для идентификатора(ов) PID, переносящего `splice_information_table` (таблица с монтажной информацией), являются дескрипторы, которые определены в настоящей Рекомендации, или частные дескрипторы пользователей.

Следует отметить, что данный дескриптор относится к указанной программе, а не ко всему объему мультиплексируемого материала. Содержимое регистрационного дескриптора описывается в таблице 6-1 и ниже:

Таблица 6-1/J.181 – Структура `registration_descriptor()`

Синтаксис	Биты	Мнемоника
<code>registration_descriptor() {</code>		
<code>descriptor_tag</code>	8	<code>uimsbf</code>
<code>descriptor_length</code>	8	<code>uimsbf</code>
<code>SCTE_splice_format_identifier</code>	32	<code>uimsbf</code>
<code>}</code>		

6.1.1 Семантическое определение полей в регистрационном дескрипторе

`descriptor_tag` (тег дескриптора): Поле `descriptor_tag` – это 8-битовое поле, определяющее каждый дескриптор. В целях регистрации это поле должно устанавливаться на `0x05`.

`descriptor_length` (длина дескриптора): Поле `descriptor_length` – это 8-битовое поле, определяющее число байтов дескриптора, следующего непосредственно за полем `descriptor_length`. Для этого регистрационного дескриптора поле `descriptor_length` должно быть установлено на `0x04`.

`SCTE_splice_format_identifier` (идентификатор формата монтажа SCTE): Этому 4-байтовому полю стандарта SCTE присваивается значение `0x43554549` (ASCII "CUEI") для идентификации программы (во всем мультиплексируемом материале), в которой он переносится как соответствующий настоящей Рекомендации.

6.2 Дескриптор идентификатора метки

Дескриптор идентификатора метки (`cue_identifier_descriptor`) может использоваться в таблице PMT для маркировки идентификаторов PID, которые переносят команды монтажа, с тем чтобы их можно было отличать по типу или уровню переносимых в них команд монтажа. Поле `cue_identifier_descriptor`, в случае его наличия, должно располагаться в цикле элементарного дескриптора. Если `cue_identifier_descriptor` не используется, то поток может переносить любую действительную согласно настоящей Рекомендации команду (см. таблицу 6-2).

Таблица 6-2/J.181 – Структура `cue_identifier_descriptor()`

Синтаксис	Биты	Мнемоника
<code>cue_identifier_descriptor() {</code>		
<code>descriptor_tag</code>	8	<code>uimsbf</code>
<code>descriptor_length</code>	8	<code>uimsbf</code>
<code>cue_stream_type</code>	8	<code>uimsbf</code>
<code>}</code>		

6.2.1 Семантическое определение полей в дескрипторе идентификатора метки

descriptor_tag (тег дескриптора): Поле descriptor_tag – это 8-битовое поле, определяющее каждый дескриптор. Для дескриптора идентификатора метки это поле должно устанавливаться на 0x8A.

descriptor_length (длина дескриптора): Поле descriptor_length – это 8-битовое поле, определяющее число байтов дескриптора, следующего непосредственно за полем descriptor_length. Для этого дескриптора поле descriptor_length должно устанавливаться на 0x01.

cue_stream_type (тип потока метки): Это 8-битовое поле, которое определено в таблице 6-3.

Таблица 6-3/J.181 – Значения поля cue_stream_type

cue_stream_type	Использование идентификатора PID
0x00	splice_insert, splice_null, splice_schedule
0x01	Все команды
0x02	Сегментация
0x03	Многоуровневый монтаж
0x04	Многоуровневая сегментация
0x05–0x7F	Зарезервировано
0x80–0xFF	Определяется пользователем

6.2.2 Описание использования поля cue_stream_type

0x00 – splice_insert, splice_null, splice_schedule: Только эти сообщения с метками допустимы в данном PID-потоке. Должен иметься максимум один идентификатор PID, определяемый этим cue_stream_type. Если такой идентификатор PID существует, то это должен быть первый поток, соответствующий настоящей Рекомендации, в цикле элементарного потока по таблице PMT.

0x01 – Все команды: По умолчанию, если этот дескриптор отсутствует. В этом идентификаторе PID могут быть использованы все сообщения.

0x02 – Сегментация: Этот идентификатор PID переносит команду time_signal (сигнал времени) и дескриптор сегментации. Он также может переносить все другие команды, если это необходимо для приложения, но его основное назначение – передать информацию сегментации содержания.

0x03 – Многоуровневый монтаж: Многоуровневый монтаж относится к системе вставки, где оператор предоставляет различные возможности для вставки программ в заданное коммерческое время для различных пользователей. Физическая и логическая реализация может быть осуществлена несколькими разными способами; некоторые из них выходят за рамки настоящей Рекомендации.

0x04 – Многоуровневая сегментация: Многоуровневая сегментация относится к системе, где оператор предоставляет разные возможности сегментации программ для различных пользователей. Физическая и логическая реализация может быть осуществлена несколькими разными способами; некоторые из них выходят за рамки настоящей Рекомендации.

0x05–0x7F: Зарезервированы для будущих расширений настоящей Рекомендации.

0x80–0xFF: Диапазон, определяемый пользователем.

6.3 Дескриптор идентификатора потока

Дескриптор идентификатора потока может быть использован в таблице PMT для маркировки компонентных потоков службы, с тем чтобы их можно было различать. Дескриптор идентификатора потока должен располагаться в цикле элементарного дескриптора вслед за соответствующим полем ES_info_length (длина информации элементарного потока). Дескриптор идентификатора потока должен использоваться, если либо program_splice_flag (флаг монтажа программы), либо program_segmentation_flag (флаг сегментации программы) равен нулю. Если используются дескрипторы идентификаторов потоков, то такой дескриптор должен присутствовать в каждом

появляющемся в таблице PMT цикле элементарного потока и должен иметь однозначно определяемый тег компонента в данной программе. См. таблицу 6-4.

Таблица 6-4/J.181 – Структура stream_identifier_descriptor()

Синтаксис	Биты	Мнемоника
stream_identifier_descriptor() { descriptor_tag descriptor_length component_tag }	8 8 8	uimsbf uimsbf uimsbf

6.3.1 Семантическое определение полей в дескрипторе идентификатора потока

descriptor_tag (тег дескриптора): Поле descriptor_tag – это 8-битовое поле, определяющее каждый дескриптор. Для дескриптора идентификатора потока поле stream_identifier_descriptor должно быть установлено на 0x52.

descriptor_length (длина дескриптора): Поле descriptor_length – это 8-битовое поле, определяющее число байтов дескриптора, следующего непосредственно за полем descriptor_length. Для этого дескриптора поле descriptor_length должно быть установлено на 0x01.

component_tag (тег компонента): Это 8-битовое поле, определяющее компонентный поток для его связи с описанием, приведенным в дескрипторе компонента. В пределах сегмента структуры программы каждый дескриптор идентификатора потока должен иметь свое значение для этого поля.

7 Таблица с монтажной информацией

7.1 Обзор

Таблица с монтажной информацией обеспечивает для сплайсера командную и управляющую информацию. Эта таблица заранее уведомляет сплайсер о событиях монтажа. Она предназначена для обеспечения вставки рекламы в поступающую сетевую информацию в линиях передачи. В этих условиях примерами событий монтажа могут служить:

- 1) монтаж с переходом от поступающей сетевой информации к рекламе; или
- 2) монтаж с переходом от рекламы к поступающей сетевой информации.

Таблица с монтажной информацией может посылаться много раз, а события монтажа могут аннулироваться. Синтаксис для информационного сегмента монтажа (splice_info_section) определяется для переноса таблицы с монтажной информацией. Информационный сегмент splice_info_section переносится в одном или нескольких PID-потоках с идентификатором(ами) PID, объявленным в таблице PMT программы.

Событие монтажа указывает на возможность выполнения монтажа одного или нескольких элементарных потоков в пределах программы. Каждое событие монтажа однозначно определяется идентификатором splice_event_id. О событиях монтажа может сообщаться тремя способами: их расписание может быть передано заранее, может быть подано предупреждение о предпуске или может быть выдана команда на выполнение события монтажа в определенных монтажных точках. Эти три типа сообщений посылаются через информационный сегмент splice_info_section. Поле splice_command_type (тип команды монтажа) определяет посылаемое сообщение. В зависимости от значения этого поля на остальной синтаксис налагаются различные ограничения.

Определены следующие типы команд: splice_null(), splice_schedule(), splice_insert(), time_signal() и bandwidth_reservation(). Если приемное устройство не поддерживает какую-либо команду, то оно может проигнорировать весь информационный сегмент splice_info_section.

Команда splice_null() предусмотрена для возможности расширений. Она может быть использована как средство передачи периодического контрольного сообщения на монтажное оборудование в нисходящем направлении.

Команда splice_schedule() – это команда, обеспечивающая возможность заблаговременной передачи расписания событий монтажа.

Команда `splice_insert()` должна посылаться по крайней мере один раз перед каждой монтажной точкой. Пакеты, содержащие всю `splice_info_table`, должны всегда предшествовать пакету, содержащему соответствующую монтажную точку (то есть первому пакету, который содержит первый байт блока доступа, время представления которого наиболее близко ко времени, указанному в `splice_info_section`).

Чтобы заранее выдать предупреждение о предстоящем монтаже (функция предпуска), команда `splice_insert()` может быть послана несколько раз до монтажной точки. Например, команда `splice_insert()` может быть послана за 8, 5, 4 и 2 секунды до пакета, содержащего соответствующую монтажную точку. Чтобы обеспечить соответствие другим срокам монтажа в системе, любое сообщение, принятое при предварительном уведомлении менее чем за 4 секунды, может не привести к желаемому результату. Сообщение `splice_insert()` должно быть послано по крайней мере один раз как минимум за 4 секунды до требуемого времени монтажа для состояния точки выхода сети.

Команда `time_signal()` предусмотрена для возможности расширений при сохранении точной синхронизации, допустимой в команде `splice_insert()`. Таким образом, учитываются новые функции, непосредственно не связанные с монтажом, где используются возможности синхронизации согласно настоящей Рекомендации при минимальном влиянии на монтажные устройства, соответствующие настоящей Рекомендации. Это дает возможность устройству, которое будет вводить время в сообщение с меткой, иметь определенное местоположение.

Команда `bandwidth_reservation()` позволяет устройствам ввода команд использовать согласованную часть ширины полосы пропускания для транспортного потока. В этой команде могут быть использованы дескрипторы, но нельзя ожидать, что они будут обработаны и переданы в нисходящем направлении для предоставления информации сигнализации.

После выдачи команды ее параметры можно изменить двумя способами. Один способ – это аннулирование выданной команды путем послышки `splice_info_section` с установленным индикатором `splice_event_cancel_indicator` и последующая передача нового `splice_info_section` с правильными/новыми параметрами. Другой способ – это просто передача последующего сообщения с новыми данными (без аннулирования старого сообщения через сообщение с меткой, которое имеет установленный бит индикатора `splice_event_cancel_indicator`).

7.1.1 Неоднородности оси времени

В случае неоднородности оси времени системы пакеты, содержащие команду `splice_insert()` или `time_signal()` с временем, выраженным по новой оси времени, не должны поступать до момента появления неоднородности оси времени. Пакеты, содержащие команду `splice_insert()` или `time_signal()` с временем, выраженным по предыдущей оси времени, не должны поступать после момента появления неоднородности оси времени. См. стандарт ИСО/МЭК 13818-4.

Ниже представлен весь синтаксис, сопровождаемый определением терминов с последующими ограничениями.

7.2 Информационный сегмент монтажа

Информационный сегмент монтажа (`splice_info_section`) должен переноситься в транспортных пакетах, при этом в любом транспортном пакете должен содержаться только один сегмент или его часть. Начало информационных сегментов `splice_info_section` должно всегда находиться в начале полезной нагрузки транспортного пакета. Когда сегмент начинается в транспортном пакете, то должно присутствовать поле `pointer_field`, равное 0x00, а бит индикатора `payload_unit_start_indicator` (индикатор начала блока полезной нагрузки) должен быть равен единице (согласно требованиям использования синтаксиса сегментов по Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1). См. таблицу 7-1.

Таблица 7-1/J.181 – Структура splice_info_section()

Синтаксис	Биты	Мнемоника	Зашифровано
splice_info_section() {			
table_id	8	uimsbf	
section_syntax_indicator	1	bslbf	
private_indicator	1	bslbf	
reserved	2	bslbf	
section_length	12	uimsbf	
protocol_version	8	uimsbf	
encrypted_packet	1	bslbf	
encryption_algorithm	6	uimsbf	
pts_adjustment	33	uimsbf	
cw_index	8	uimsbf	
reserved	12	bslbf	
splice_command_length	12	uimsbf	
splice_command_type	8	uimsbf	E
if (splice_command_type == 0x00)			E
splice_null()			
if (splice_command_type == 0x04)			E
splice_schedule()			
if (splice_command_type == 0x05)			E
splice_insert()			
if (splice_command_type == 0x06)			E
time_signal()			
if (splice_command_type == 0x07)			E
bandwidth_reservation()			
descriptor_loop_length	16	uimsbf	E
for (i = 0; i < N1; i++)			
splice_descriptor()			E
for (i = 0; i < N2; i++)			
alignment_stuffing	8	bslbf	E
if (encrypted_packet)			
E_CRC_32	32	rpchof	E
CRC_32	32	rpchof	
}			

7.2.1 Семантическое определение полей в структуре splice_info_section()

table_id (идентификатор таблицы): Это 8-битовое поле со значением 0xFC.

section_syntax_indicator (индикатор синтаксиса сегмента): Это 1-битовое поле, которое всегда должно быть установлено на "0", указывающее на то, что должны использоваться короткие сегменты по стандарту MPEG.

private_indicator (частный индикатор): Это 1-битовый флаг, который должен быть установлен на "0".

section_length (длина сегмента): Это 12-битовое поле, определяющее число оставшихся байтов в поле splice_info_section, непосредственно следующих за полем section_length вплоть до конца splice_info_section. Значение в этом поле не должно превышать 4093.

protocol_version (версия протокола): Это 8-битовое поле, представляющее целое число без знака; функция этого поля – разрешить в будущем, чтобы этот тип таблицы переносил параметры, которые могут иметь структуру, отличную от структуры параметров, определенных в текущем протоколе. В настоящее время единственным действительным значением для поля **protocol_version** является нуль. Ненулевые значения этого поля могут быть использованы в будущей версии настоящей Рекомендации для указания на структурно различные таблицы.

encrypted_packet (шифрованный пакет): Когда этот бит установлен на "1", он указывает на то, что части информационного сегмента splice_info_section, начинающиеся со splice_command_type и заканчивающиеся E_CRC_32 включительно, зашифрованы. Когда этот бит установлен на "0", никакая часть этого сообщения не шифруется. Потенциально зашифрованные части splice_info_table помечены в таблице 7-1 буквой E в столбце "Зашифровано".

encryption_algorithm (алгоритм шифрования): Это 6-битовое целое число без знака, определяющее, какой алгоритм шифрования был использован для шифрования текущего сообщения. Если бит encrypted_packet равен нулю, то поле encryption_algorithm присутствует, но оно не описано. Подробная информация об использовании этого поля содержится в п. 9, в частности в таблице 9-1.

pts_adjustment (коррекция pts): Это 33-битовое целое число без знака, которое не шифруется и которое должно использоваться монтажным устройством в качестве смещения, добавляемого (иногда) к зашифрованному полю pts_time, во время передачи всего этого сообщения для получения намечаемого времени монтажа. Когда это поле имеет нулевое значение, поле (поля) pts_time должно использоваться без смещения. Обычно формирователь сообщения с меткой будет заносить в это поле нулевое значение. Это значение коррекции является средством, с помощью которого устройство в восходящем направлении передачи, повторно регистрирующее pcr/pts/dts, может предоставить монтажному устройству средства преобразования поля pts_time сообщения во вновь введенный временной домен.

Предполагается, что первое устройство, которое повторно регистрирует pcr/pts/dts и пересылает сообщение с меткой, будет вводить некоторое значение в поле pts_adjustment, которое является приращением времени между доменом времени входа данного устройства и его доменом времени выхода. Все последующие устройства, которые также повторно регистрируют pcr/pts/dts, могут в дальнейшем изменять поле pts_adjustment путем сложения их приращения времени с существующим приращением времени поля и занесения результата обратно в поле pts_adjustment. При каждом изменении поля pts_adjustment устройство, осуществляющее такое изменение, должно повторно вычислять и обновлять содержимое поля CRC_32.

Поле pts_adjustment всегда должно иметь значение, пригодное для преобразования значения поля pts_time в значение времени по оси текущего времени. Это преобразование осуществляется путем суммирования значений двух полей. При наличии состояния переполнения или циклического наложения перенос должен игнорироваться.

cw_index (индекс управляющего слова): Это 8-битовое целое число без знака, которое передает информацию о том, какое управляющее слово (ключ) должно быть использовано для расшифровывания сообщения. В монтажном устройстве может храниться до 256 ключей, заранее предусмотренных для этой цели. Когда бит encrypted_packet равен нулю, тогда поле cw_index присутствует, но оно не описано.

splice_command_length (длина команды монтажа): Это 12-битовое поле длины команды монтажа. Если это поле равно 0xFFFF, тогда длина не определена.

splice_command_type (тип команды монтажа): Это 8-битовое целое число без знака, которому присваивается одно из значений, приведенных в таблице 7-2.

Таблица 7-2/J.181 – Значения поля splice_command_type

splice_command_type value	Команда
0x00	splice_null
0x01	Зарезервировано
0x02	Зарезервировано
0x03	Зарезервировано
0x04	splice_schedule
0x05	splice_insert
0x06	time_signal
0x07	bandwidth_reservation
0x08–0xFF	Зарезервировано

descriptor_loop_length (длина цикла дескриптора): Это 16-битовое целое число без знака, определяющее число байтов, используемых в цикле дескриптора монтажа, непосредственно следующем за данным полем.

alignment_stuffing (заполнение выравнивания): При использовании шифрования это поле является функцией определенного выбранного алгоритма шифрования. Поскольку для ряда алгоритмов шифрования требуется конкретная длина шифруемых данных, необходимо иметь возможность ввести байты заполнения. Например, в стандарте DES требуется длина данных, кратная 8 байтам, чтобы осуществить шифрование до самого конца пакета. Это позволяет использовать алгоритм согласно стандарту DES без необходимости в особой версии алгоритма шифрования.

Когда шифрование не используется, это поле может присутствовать, но не должно использоваться для переноса действительных данных.

E_CRC_32 (циклическая проверка четности с избыточностью и шифрованием): Это 32-битовое поле, содержащее значение кода CRC, задающее нулевое выходное значение регистров в декодере, определенном в Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, после обработки всей расшифровываемой части поля splice_info_section. Поле E_CRC_32 предназначено для индикации того, что расшифровывание было выполнено успешно. Таким образом, нулевое выходное значение получается после расшифровывания и в результате обработки полей, начиная с поля splice_command_type и заканчивая полем E_CRC_32.

CRC_32 (циклическая проверка четности с избыточностью): Это 32-битовое поле, содержащее значение кода CRC, задающее нулевое выходное значение регистров в декодере, определенном в Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, после обработки всего поля splice_info_section, которое включает поля начиная с поля table_id и заканчивая полем CRC_32. Обработка поля CRC_32 должна происходить до расшифровывания зашифрованных полей, при этом зашифрованные поля должны использоваться в их зашифрованном состоянии.

7.3 Команды монтажа

7.3.1 Команда splice_null()

Команда splice_null() предусмотрена для возможности расширения настоящей Рекомендации. Эта команда позволяет пересылать splice_info_table, которая может переносить дескрипторы без необходимости передавать одну из других определяемых команд. Эта команда может быть также использована как периодическое контрольное сообщение для текущего контроля целостности оборудования ввода метки и целостности канала (см. таблицу 7-3).

Таблица 7-3/J.181 – Команда splice_null()

Синтаксис	Биты	Мнемоника
splice_null() { }		

7.3.2 Команда splice_schedule()

Команда splice_schedule() предусмотрена для обеспечения возможности заблаговременной передачи расписания событий монтажа. См. таблицу 7-4.

Таблица 7-4/J.181 – Команда splice_schedule()

Синтаксис	Биты	Мнемоника
splice_schedule() { splice_count for (i = 0; i < splice_count; i++) { splice_event_id splice_event_cancel_indicator reserved if (splice_event_cancel_indicator == '0') { out_of_network_indicator program_splice_flag duration_flag reserved if (program_splice_flag == '1') utc_splice_time if (program_splice_flag == '0') { component_count for (j = 0; j < component_count; j++) { component_tag utc_splice_time } } if (duration_flag) break_duration() unique_program_id avail_num avails_expected } } }	8 32 1 7 1 1 1 5 32 8 8 32 16 8 8	uimsbf uimsbf bslbf bslbf bslbf bslbf bslbf bslbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf

7.3.2.1 Семантическое определение полей в команде splice_schedule()

splice_count (подсчет событий монтажа): Это 8-битовое целое число без знака, указывающее на число событий монтажа, определяемых в последующем цикле.

splice_event_id (идентификатор события монтажа): Это 32-битовый однозначно определяемый идентификатор события монтажа.

splice_event_cancel_indicator (идентификатор аннулирования события монтажа): Это 1-битовый флаг, который, будучи установленным на "1", указывает на то, что ранее посланное событие монтажа, определяемое идентификатором splice_event_id, аннулировано.

out_of_network_indicator (индикатор положения вне сети): Это 1-битовый флаг. Будучи установленным на "1", он указывает на то, что событие монтажа представляет собой возможность выйти из сетевой информации линии передачи и что значение поля utc_splice_time должно относиться к намеченной точке выхода или программной точке выхода. Будучи установленным на "0", этот флаг указывает на то, что событие монтажа представляет собой возможность возврата к сетевой информации линии передачи и что значение поля utc_splice_time должно относиться к намеченной точке входа или программной точке входа.

program_splice_flag (флаг монтажа программы): Это 1-битовый флаг. Будучи установленным на "1", он указывает на то, что сообщение относится к программной монтажной точке и что режим монтажа – это режим монтажа программы, при котором все идентификаторы PID/компоненты программы подлежат монтажу. Будучи установленным на "0", этот флаг указывает на то, что режим монтажа – это режим монтажа компонентов, при котором каждый компонент, который должен быть подвергнут монтажу, будет перечислен отдельно в последующем синтаксисе.

duration_flag (флаг продолжительности): Это 1-битовый флаг, указывающий на наличие поля `break_duration()`.

utc_splice_time (время UTC монтажа): Это 32-битовое целое число без знака, представляющее время события монтажа, о котором передано сигнальное сообщение, в виде числа секунд от 00 часов UTC 6 января 1980 года с подсчетом включенных сюда дополнительных секунд. Время `utc_splice_time` может быть преобразовано во время UTC без использования значения `GPS.UTC_offset` (сдвига времени UTC системы GPS) согласно таблице времени системы. Поле `utc_splice_time` используется только в команде `splice_schedule()`.

component_count (подсчет компонентов): Это 8-битовое целое число без знака, определяющее число экземпляров данных элементарных PID-поток в последующем цикле. Компоненты эквивалентны элементарным PID-потокам.

component_tag (тег компонента): Это 8-битовое значение, определяющее элементарный PID-поток, содержащий монтажную точку, задаваемую значением последующего поля `splice_time()`. Это значение должно быть таким же, как и значение, используемое в поле `stream_identification_descriptor()` для определения этого элементарного PID-потока.

unique_program_id (однозначно определяемый идентификатор программы): Это значение должно обеспечивать однозначно определяемую идентификацию для события просмотра в рамках службы. Поскольку настоящая Рекомендация предназначена для всеохватывающего стандарта, требуется значение, не зависящее от любого стандарта. Данное поле обеспечивает ту же функцию, что и поле `event_id` (идентификатор события) в стандартах ATSC или DVB.

avail_num (номер интервала коммерческого времени) (ранее "avail"): Это поле обеспечивает идентификацию конкретного интервала коммерческого времени в рамках одного поля `unique_program_id`. Ожидается, что это значение будет возрастать на единицу с каждым новым интервалом коммерческого времени в пределах события просмотра. Предполагается, что при первом интервале коммерческого времени в новом событии просмотра это значение будет переустанавливаться на единицу. Ожидается, что это поле будет возрастать на единицу при каждом новом интервале коммерческого времени. Для указания на то, что это поле не используется, ему факультативно может быть присвоено нулевое значение.

avails_expected (ожидаемые интервалы коммерческого времени) (ранее "avail_count"): Это поле обеспечивает подсчет ожидаемого числа отдельных интервалов коммерческого времени в рамках текущего события просмотра. Когда это поле равно нулю, это указывает на то, что поле `avail_num` не играет роли.

7.3.3 Команда `splice_insert()`

Команда `splice_insert()` должно посылаться по крайней мере один раз для каждого события монтажа. Использование данного сообщения рассматривается в п. 5.3. См. также таблицу 7-5.

Таблица 7-5/J.181 – Команда splice_insert()

Синтаксис	Биты	Мнемоника
splice_insert() {		
splice_event_id	32	uimsbf
splice_event_cancel_indicator	1	bslbf
reserved	7	bslbf
if (splice_event_cancel_indicator == "0") {		
out_of_network_indicator	1	bslbf
program_splice_flag	1	bslbf
duration_flag	1	bslbf
splice_immediate_flag	1	bslbf
reserved	4	bslbf
If ((program_splice_flag == "1") && (splice_immediate_flag == "0"))		
splice_time()		
if (program_splice_flag == "0") {		
component_count	8	uimsbf
for (i = 0; i < component_count; i++) {		
component_tag	8	uimsbf
if (splice_immediate_flag == "0")		
splice_time()		
}		
}		
if (duration_flag == "1")		
break_duration()		
unique_program_id	16	uimsbf
avail_num	8	uimsbf
avails_expected	8	uimsbf
}		

7.3.3.1 Семантическое определение полей в команде splice_insert()

splice_event_id (идентификатор события монтажа): Это 32-битовый однозначный идентификатор события монтажа.

splice_event_cancel_indicator (индикатор аннулирования события монтажа): Это 1-битовый флаг, который, будучи установленным на "1", указывает на то, что ранее посланное событие монтажа, определяемое идентификатором splice_event_id, аннулировано.

out_of_network_indicator (индикатор положения вне сети): Это 1-битовый флаг. Будучи установленным на "1", он указывает на то, что событие монтажа представляет собой возможность выйти из сетевой информации линии передачи и что значение поля splice_time(), измененное по значению поля pts_adjustment, должно относиться к намеченной точке выхода или программной точке выхода. Будучи установленным на "0", этот флаг указывает на то, что событие монтажа представляет собой возможность возврата к сетевой информации линии передачи и что значение поля splice_time(), измененное по значению поля pts_adjustment, должно относиться к намеченной точке входа или программной точке входа.

program_splice_flag (флаг монтажа программы): Это 1-битовый флаг. Будучи установленным на "1", он указывает на то, что сообщение относится к программной монтажной точке и что режимом монтажа является режим монтажа программы, при котором все идентификаторы PID/компоненты программы подлежат монтажу. Будучи установленным на "0", этот флаг указывает на то, что режимом является режим монтажа компонентов, при котором каждый компонент, который должен быть подвергнут монтажу, будет перечислен отдельно в последующем синтаксисе.

duration_flag (флаг продолжительности): Это 1-битовый флаг, который, будучи установленным на "1", указывает на наличие поля break_duration().

splice_immediate_flag (флаг немедленного монтажа): Когда этот флаг установлен на "1", он указывает на отсутствие поля splice_time() и на то, что режимом монтажа должен быть режим

немедленного монтажа, при котором монтажное устройство выбирает ближайшую по отношению к пакету монтажной информации возможность монтажа. Когда этот флаг равен "0", он указывает на наличие поля `splice_time()` по крайней мере в одном месте в команде `splice_insert()`.

component_count (подсчет компонентов): Это 8-битовое целое число без знака, определяющее число экземпляров данных элементарных PID-потоков в последующем цикле. Компоненты эквивалентны элементарным PID-потокам.

component_tag (тег компонента): Это 8-битовое значение, определяющее элементарный PID-поток, содержащий монтажную точку, задаваемую значением последующего поля `splice_time()`. Это значение должно быть таким же, как значение, используемое в поле `stream_identification_descriptor()` для определения этого элементарного потока.

unique_program_id (однозначно определяемый идентификатор программы): Это значение должно обеспечивать однозначно определяемую идентификацию для события просмотра в рамках службы. Поскольку настоящая Рекомендация предназначена для всеохватывающего стандарта, требуется значение, не зависящее от любого стандарта. Данное поле обеспечивает ту же функцию, что и поле `event_id` в стандартах ATSC или DVB.

avail_num (номер интервала коммерческого времени) (ранее "avail"): Это поле обеспечивает идентификацию конкретного интервала коммерческого времени в рамках одного поля `unique_program_id`. Ожидается, что это значение будет возрастать на единицу с каждым новым интервалом коммерческого времени в пределах события просмотра. Предполагается, что при первом интервале коммерческого времени в новом событии просмотра это значение будет переустанавливаться на "1". Ожидается, что это поле будет возрастать на единицу при каждом новом интервале коммерческого времени. Для указания на то, что это поле не используется, ему факультативно может быть присвоено нулевое значение.

avails_expected (ожидаемые интервалы коммерческого времени) (ранее "avail_count"): Это поле обеспечивает подсчет ожидаемого числа отдельных интервалов коммерческого времени в рамках текущего события просмотра. Когда это поле равно нулю, это указывает на то, что поле `avail_num` не играет роли.

7.3.4 Команда `time_signal()`

Команда `time_signal()` обеспечивает механизм доставки данных с синхронизацией по времени. В синтаксисе команды `time_signal()` учитывается синхронизация информации, которая переносится в этом сообщении, системным таймером (STC). Однозначно определяемая полезная нагрузка сообщения переносится в дескрипторе; однако синтаксис и возможности транспортировки, предоставляемые сообщениям `splice_insert()`, также предоставляются команде `time_signal()`. Но перенос может быть осуществлен в идентификаторе PID, отличном от PID, переносящего другие сообщения с меткой, используемые для уведомления о монтажных точках.

Если поле `time_specified_flag` установлено на "0", указывая на отсутствие поля `pts_time` в сообщении, тогда эта команда будет интерпретироваться как немедленная команда. Следует понимать, что такое использование этой команды приведет к неопределенному количеству систематических ошибок.

Поскольку в команде `time_signal()` используются дескрипторы для большей части конкретной информации, эта команда может превосходить по длине один транспортный пакет MPEG. Настоятельно рекомендуется, при возможности, передавать эту команду в одном пакете. Это не всегда возможно в ситуациях, когда, например, имеется большой объем уникальной информации или когда для описания этой уникальной информации используется другая спецификация. См. таблицу 7-6.

Таблица 7-6/J.181 – Команда `time_signal()`

Синтаксис	Биты	Мнемоника
<code>time_signal() { splice_time() }</code>		

7.3.4.1 Семантическое определение команды `time_signal()`

Команда `time_signal()` обеспечивает стандартный метод связывания отсчета времени согласно полю `pts_time` с произвольным дескриптором (или дескрипторами), как определяется синтаксисом поля `splice_info_section` (см. таблицу 7-1). Описание дескрипторов монтажа приводится в п. 8.

7.3.5 Команда `bandwidth_reservation()`

Команда `bandwidth_reservation()` служит для резервирования ширины полосы пропускания при мультимплексной передаче. Типичное использование эта команда находит в спутниковой системе доставки, где требуется постоянное наличие пакетов определенного идентификатора PID при желаемой частоте повторений, чтобы гарантировать определенную ширину полосы пропускания для этого идентификатора PID. Это сообщение отличается от команды `splice_null()`, так что эта команда может быть легко обработана приемной аппаратурой однозначным образом (то есть "извлечена" спутниковым приемником при мультимплексной передаче). Если дескриптор посылается с этой командой, нельзя ожидать, что он будет перенесен через всю цепь передачи; это должен быть частный дескриптор, используемый только в процессе резервирования ширины полосы пропускания. См. таблицу 7-7.

Таблица 7-7/J.181 – Команда `bandwidth_reservation()`

Синтаксис	Биты	Мнемоника
<code>bandwidth_reservation() {</code> <code>}</code>		

7.4 Время

7.4.1 Структура `splice_time()`

Структура `splice_time()`, будучи измененной с помощью поля `pts_adjustment`, определяет время события монтажа. См. таблицу 7-8.

Таблица 7-8/J.181 – Структура `splice_time()`

Синтаксис	Биты	Мнемоника
<code>splice_time() {</code> <code>time_specified_flag</code> <code>if (time_specified_flag == 1) {</code> <code>reserved</code> <code>pts_time</code> <code>}</code> <code>else</code> <code>reserved</code> <code>}</code>	1 6 33 7	<code>bslbf</code> <code>bslbf</code> <code>uimsbf</code> <code>bslbf</code>

7.4.1.1 Семантическое определение полей в структуре `splice_time()`

`time_specified_flag` (флаг заданного времени): Это 1-битовый флаг, который, будучи установленным на "1", указывает на наличие поля `pts_time` и соответствующих зарезервированных битов.

`pts_time` (время pts): Это 33-битовое поле, указывающее на время с точки зрения тактов генератора частоты 90 кГц программы. Это поле, будучи измененным с помощью поля `pts_adjustment`, представляет время намеченной монтажной точки.

7.4.2 Структура `break_duration()`

Структура `break_duration()` определяет продолжительность коммерческого перерыва (перерывов). Она может быть использована для указания сплайсеру, когда закончится перерыв и когда будет иметь место точка входа сети. См. таблицу 7-9.

Таблица 7-9/J.181 – Структура break_duration()

Синтаксис	Биты	Мнемоника
break_duration() { auto_return reserved duration }	1 6 33	bslbf bslbf uimsbf

7.4.2.1 Семантическое определение полей в структуре break_duration()

auto_return (автовозврат): Это 1-битовый флаг; будучи установленным на "1", означает, что продолжительность перерыва должна использоваться монтажным устройством для получения информации о том, когда должен иметь место возврат к сетевой информации линии передачи (окончание перерыва). Команда splice_insert() с битом поля out_of_network_indicator, установленным на "0", не предназначена для передачи с целью закончить этот перерыв. Когда этот флаг равен "0", то поле продолжительности перерыва, если имеется, не требуется для окончания перерыва, поскольку для этого будет посылаться новая команда splice_insert(). В этом случае наличие поля break_duration служит механизмом защиты при потере команды splice_insert() в конце перерыва.

duration (продолжительность): Это 33-битовое поле, указывающее на истекшее время с точки зрения тактов генератора частоты 90 кГц программы.

7.5 Ограничения

7.5.1 Ограничения на структуру splice_info_section()

Структура splice_info_section должна переноситься в одном или нескольких PID-потоках, которые характерны для программы и ссылки на которые имеются в таблице PMT. Идентификаторы PID информационного сегмента splice_info_section должны определяться в таблице PMT полем stream_type, равным 0x86.

Информационный сегмент splice_info_section, переносимый в одном или нескольких PID-потоках, ссылки на которые даются в таблице PMT, должен содержать только информацию о событиях монтажа, имеющих место в этой программе.

Событие монтажа должно определяться одним значением поля splice_event_id.

Если будет использован режим монтажа компонентов, каждый элементарный PID-поток должен определяться полем stream_identifier_descriptor, которое переносится в цикле таблицы PMT, по одному дескриптору на каждый идентификатор PID. Поле stream_identifier_descriptor должно переносить тег компонента, который однозначно соответствует одному PID-потoku среди тех PID-потоков, которые содержатся в программе и перечислены в таблице PMT для этой программы.

Любой splice_event_id, который посылается в структуре splice_info_section с использованием команды splice_schedule(), должен вновь посылаться перед событием монтажа с помощью команды splice_insert(). Следовательно, должно иметься соответствие между значениями поля splice_event_id, выбранными для отдельных событий монтажа, о которых сообщает команда splice_schedule() (далекое будущее), и значениями поля splice_event_id, используемыми в команде splice_insert() (ближайшее будущее), для указания на одни и те же события монтажа.

Значения поля splice_event_id не требуется посылать в возрастающем порядке в последующих сообщениях, и они не должны возрастать в хронологическом порядке. Эти значения могут выбираться произвольным образом. При использовании команды splice_schedule() значения поля splice_event_id должны быть однозначными в течение периода действия команды splice_schedule(). Значение поля splice_event_id может быть использовано повторно, когда прошло связанное с ним время монтажа.

Когда бит поля splice_immediate_flag установлен на "1", то время для монтажа должно интерпретироваться как текущее время. Этот режим называется "режимом немедленного монтажа". Когда этот режим используется вместе с командой splice_insert(), монтаж может происходить при ближайшей (предыдущей или последующей) возможности, выявленной сплайсером. "Режим немедленного монтажа" может быть использован как для монтажных точек входа, так и монтажных точек выхода, то есть для обоих состояний поля out_of_network_indicator.

Допускается, что любое коммерческое время может заканчиваться сообщением режима монтажа программы, режима монтажа компонентов или отсутствием сообщения (с помощью которого достигается `break_duration`), независимо от характера сообщения о начале коммерческого времени.

7.5.2 Ограничения на интерпретацию времени

7.5.2.1 Ограничения на структуру `splice_time()` для команды `splice_insert()`

Для поля `splice_command_type`, равного 0x05 (команда `splice_insert()`), должны налагаться следующие ограничения на структуру `splice_time()`:

Хотя бы одно сообщение для точки выхода сети должно поступать по меньшей мере за 4 секунды до сигнализируемого времени монтажа (значение поля `pts_time`, измененное согласно значению поля `pts_adjustment`), если это время определено. Для сетевой точки выхода предусматривается сообщение режима немедленного монтажа, но фактическое время монтажа не определяется, и рекомендуется, чтобы такие сообщения использовались только для раннего завершения перерывов. Когда для точек входа сети используются сообщения с меткой, не относящиеся к режиму немедленного монтажа, то сообщение с меткой должно поступать в сплайсер до поступления изображения, о котором сообщается, в точку входа в приемнике.

Точка выхода находится между двумя компонентами представления. Предполагаемая точка выхода для события монтажа, о котором передан сигнал, должна быть точкой выхода, которая находится непосредственно перед компонентом представления, время представления которого наиболее близко ко времени, указанному в поле `pts_time`, измененном по значению поля `pts_adjustment`.

Точка входа находится между двумя компонентами представления. Предполагаемая точка входа для события монтажа, о котором передан сигнал, должна быть точкой входа, которая находится непосредственно перед компонентом представления, время представления которого наиболее близко ко времени, указанному в поле `pts_time`, измененном по значению поля `pts_adjustment`.

Когда действует режим монтажа компонентов, а бит поля `out_of_network_indicator` равен "1" (начало перерыва), каждый компонент, перечисленный в цикле компонентов команды `splice_insert()`, должен переключаться с сетевого компонента на предоставляемый сплайсером компонент в указанное время. Компоненты, не содержащиеся в цикле компонентов сообщения, останутся неизменными: если выданный сплайсером компонент был сетевым компонентом, то он останется сетевым компонентом; если выданный сплайсером компонент был предоставляемым сплайсером компонентом, то он останется предоставляемым сплайсером компонентом.

Когда действует режим монтажа компонентов, а бит поля `out_of_network_indicator` равен "0" (конец перерыва), каждый компонент, перечисленный в цикле компонентов команды `splice_insert()`, должен переключаться с предоставляемого сплайсером компонента на сетевой компонент в указанное время. Компоненты, не содержащиеся в цикле компонентов сообщения, останутся неизменными: если выданный сплайсером компонент был сетевым компонентом, то он останется сетевым компонентом; если выданный сплайсером компонент был предоставляемым сплайсером компонентом, то он останется предоставляемым сплайсером компонентом.

Когда действует режим монтажа компонентов, а режим немедленного монтажа не действует, то первый компонент, содержащийся в цикле компонентов команды `splice_insert()`, должен иметь действительное значение поля `pts_time` в связанной с ним структуре `splice_time()`, и это значение поля `pts_time` именуется значением поля `pts_time` по умолчанию. Последующие компоненты, перечисленные в цикле компонентов того же сообщения, которые не имеют связанного с ними значения поля `pts_time`, должны использовать это значение поля `pts_time` по умолчанию. Допускается, что любой и все компоненты, следующие за первым компонентом, перечисленным в цикле компонентов команды `splice_insert()`, могут содержать однозначно определяемое значение поля `pts_time`, отличающееся от значения того же поля по умолчанию.

В режиме монтажа компонентов все значения поля `pts_time`, заданные в цикле компонентов команды `splice_insert()`, должны быть изменены по значению поля `pts_adjustment` для получения каждого предполагаемого значения для точки выхода или точки входа, о которых был передан сигнал. Значение поля `pts_adjustment`, предоставляемое любым устройством, которое генерирует или изменяет значение поля `pts_adjustment`, должно применяться ко всем полям `pts_time` в сообщении.

7.5.2.2 Ограничения на структуру break_duration() для команды splice_insert()

Для поля splice_command_type, равного 0x05 (команда splice_insert), на структуру break_duration() должны налагаться следующие ограничения:

Значение, заданное в структуре break_duration(), интерпретируется как намеченная продолжительность коммерческого перерыва. Это – факультативное поле, которое должно использоваться, когда бит поля out_of_network_indicator равен 1. Оно может быть использовано в той же команде splice_insert(), которая определяет время начала перерыва, с тем чтобы сплайсер мог вычислить время окончания перерыва.

Перерывы могут завершаться генерированием команды splice_insert() с битом поля out_of_network_indicator, установленным на "0". Может быть задана структура splice_time() либо может быть использован режим немедленного монтажа. При выдаче структуры break_duration() в начале перерыва (когда значение поля auto_return было установлено на "0") значение поля break_duration может быть использовано как механизм резервирования для гарантирования того, что в случае потери пакета с меткой фактически произойдет возврат к сети.

Перерывы также могут завершаться путем выдачи продолжительности перерыва в начале перерыва или возложением на монтажное устройство функции возврата к сетевой информации линии передачи в должное время. Флаг поля auto_return должен быть установлен на "1". Это будет соответствовать режиму автоматического возврата. Для перерывов в режиме автоматического возврата не требуется и не запрещается передача в конце перерыва сообщений с меткой с битом поля out_of_network_indicator, установленным на "0". Следовательно, приемное устройство не должно ожидать сообщения с меткой в конце перерыва, чтобы функционировать должным образом. Однако перерывы в режиме автоматического возврата могут заканчиваться рано. Для преждевременного окончания перерыва может быть выдана вторая команда splice_insert(), где бит поля out_of_network_indicator равен 0. Новое время возврата к сетевому монтажу может задаваться обновленной структурой splice_time(), либо может быть использовано сообщение режима немедленного монтажа. Сообщение с меткой с битом поля out_of_network_indicator, установленным на "0", должно всегда отменять поле продолжительности перерыва предыдущего сообщения с меткой (с битом поля out_of_network_indicator, установленным на "1"), если эта продолжительность перерыва, о которой было сообщено, все еще имеет место.

8 Дескрипторы монтажа

8.1 Обзор

Поле splice_descriptor – это прототип для добавления новых полей к полю splice_info_section. Во всех включенных дескрипторах используется один и тот же синтаксис для первых шести байтов. Чтобы допустить добавление частной информации, включен код "идентификатора". Этим устраняется необходимость в регистрационном дескрипторе в цикле дескрипторов.

Любая приемная аппаратура должна опускать все дескрипторы с неизвестными идентификаторами или неизвестными тегами дескрипторов. Что касается дескрипторов с неизвестными идентификаторами, то приемная аппаратура должна опускать дескрипторы с неизвестным значением поля splice_descriptor_tag.

Дескрипторы монтажа могут существовать в поле splice_info_section для целей расширения, характерного для различных команд. См. таблицу 8-1.

Таблица 8-1/J.181 – Теги дескрипторов монтажа

Тег	Дескрипторы для идентификатора "CUEI"
0x00	avail_descriptor
0x01	DTMF_descriptor
0x02	segmentation_descriptor
0x03-0xFF	Зарезервированы для будущих полей splice_descriptor

8.2 Дескриптор монтажа

Синтаксис дескриптора монтажа, приведенный в данном пункте, должен использоваться как шаблон для конкретных реализаций дескриптора, предназначенного для поля splice_info_section. Следует

отметить, что дескрипторы монтажа используются только в поле `splice_info_section`. Они не должны использоваться в синтаксисе стандарта MPEG, таком как таблица PMT, или в синтаксисе любого другого стандарта. Это позволяет опираться на весь диапазон тегов дескрипторов при определении новых дескрипторов. См. таблицу 8-2.

Таблица 8-2/J.181 – Структура `splice_descriptor()`

Синтаксис	Биты	Мнемоника
<code>splice_descriptor() {</code>		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Identifier	32	uimsbf
for (i = 0; i < N; i++) {		
private_byte	8	uimsbf
}		
<code>}</code>		

8.2.1 Семантическое определение полей в структуре `splice_descriptor()`

splice_descriptor_tag (тег дескриптора монтажа): Это 8-битовое число, определяющее синтаксис частных байтов, образующих тело дескриптора. Теги дескрипторов определяются владельцем дескриптора, зарегистрированным с использованием идентификатора.

descriptor_length (длина поля): Это 8-битовое число, задающее длину в байтах дескриптора, следующего за этим полем. Дескрипторы ограничены 256 байтами, так что это значение ограничено 254.

identifier (идентификатор): Это 32-битовое поле, как определено в Рекомендации МСЭ-Т H.222.0 | стандарте ИСО/МЭК 13818-1, пп. 2.6.8 и 2.6.9, для поля `format_identifier` структуры `registatuiou_descriptor()`. Должны использоваться только зарегистрированные значения идентификаторов, признанные органом регистрации общества SMPTE, а также LLC (см. сайт <http://www.smpite-ra.org/mpegreg.html>). Его использование в этом дескрипторе должно охватывать и определять только частную информацию, содержащуюся в данном дескрипторе. Это 32-битовое число используется для идентификации владельца дескриптора. Код 0x43554549 (ASCII "CUEI") для дескрипторов, определяемый в настоящей Рекомендации, зарегистрирован обществом SMPTE.

private_byte (частный байт): Это остаточный член дескриптора, предназначенный для полей данных, как того требует определяемый дескриптор.

8.3 Конкретные дескрипторы монтажа

8.3.1 Структура `avail_descriptor()`

Структура `avail_descriptor` (дескриптор коммерческого времени) является реализацией `splice_descriptor`. Она обеспечивает факультативное расширение команды `splice_insert()`, позволяющее передать идентификатор полномочий для интервала коммерческого времени. Может быть включено несколько копий этого дескриптора путем использования предоставленного механизма организации циклов. Этот идентификатор предназначен для копирования функций системы тональных сигналов меток, используемой в аналоговых системах для вставки рекламы. Этот дескриптор предназначен только для использования командой `splice_insert()` в поле `splice_info_section` См. таблицу 8-3.

Таблица 8-3/J.181 – Структура `avail_descriptor ()`

Синтаксис	Биты	Мнемоника
<code>avail_descriptor() {</code>		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
identifier	32	uimsbf
provider_avail_id	32	uimsbf
<code>}</code>		

8.3.1.1 Семантическое определение полей в структуре `avail_descriptor()`

splice_descriptor_tag (тег дескриптора монтажа): Это 8-битовое число, определяющее синтаксис частных байтов, образующих тело данного дескриптора. Поле `splice_descriptor_tag` должно иметь значение 0x00.

descriptor_length (длина дескриптора): Это 8-битовое число, задающее длину в байтах дескриптора, следующего за этим полем. Поле `descriptor_length` должно иметь значение 0x08.

identifier (идентификатор): Это 32-битовое число, используемое для идентификации владельца дескриптора. Идентификатор должен иметь значение 0x43554549 (ASCII "CUEI").

provider_avail_id (идентификатор коммерческого времени поставщика): Это 32-битовое число, обеспечивающее информацию, которую может использовать приемное устройство для изменения своего поведения в течение интервала коммерческого времени или вне этого интервала. Он может использоваться подобно аналоговому тональным сигналам метки. Примером может служить сеть, управляющая филиалом или головной станцией кабельного телевидения для гашения сигналов спортивной передачи.

8.3.2 Структура `DTMF_descriptor()`

Структура `DTMF_descriptor()` является реализацией `splice_descriptor`. Эта структура обеспечивает факультативное расширение команды `splice_insert()`, позволяющее приемному устройству генерировать обычную последовательность аналоговых сигналов DTMF на основе принимаемого информационного сегмента `splice_info_section`. См. таблицу 8-4.

Таблица 8-4/J.181 – Структура `DTMF_descriptor()`

Синтаксис	Биты	Мнемоника
<code>DTMF_descriptor() {</code>		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Identifier	32	uimsbf
Preroll	8	uimsbf
dtmf_count	3	uimsbf
reserved	5	bslbf
for (i = 0; i < dtmf_count ; i++) {		
DTMF_char	8	uimsbf
}		
<code>}</code>		

8.3.2.1 Семантическое определение полей в структуре `DTMF_descriptor()`

splice_descriptor_tag (тег дескриптора монтажа): Это 8-битовое число, определяющее синтаксис частных байтов, образующих тело этого дескриптора. Поле `splice_descriptor_tag` должно иметь значение 0x01.

descriptor_length (длина дескриптора): Это 8-битовое число, задающее длину в байтах дескриптора, следующего за этим полем.

identifier (идентификатор): Это 32-битовое число, используемое для идентификации владельца дескриптора. Идентификатор должен иметь значение 0x43554549 (ASCII "CUEI").

preroll (предпуск): Это 8-битовое число, представляющее время в десятках секунд, когда на аналоговый выход устройства подаются сигналы DTMF. Это определяет диапазон предпуска от 0 до 25,5 секунд. Информационный сегмент `splice_info_section` должен посылаться по меньшей мере на две секунды раньше этого значения времени. Предлагаемая минимальная продолжительность предпуска составляет 4 секунды.

dtmf_count (подсчет числа символов DTMF): Значение этого флага представляет число символов DTMF, которое должно быть генерировано устройством.

DTMF_char (символ DTMF): Это значение в коде ASCII для цифр от "0" до "9" и символов "*", "#". Устройство должно использовать эти значения для генерирования последовательности сигналов DTMF, подаваемых на аналоговый выход. Эта последовательность должна заканчиваться последним символом, передающим маркерный импульс синхронизации для предпуска.

8.3.3 Структура segmentation_descriptor()

Структура segmentation_descriptor() – это реализация структуры splice_descriptor(). Она обеспечивает факультативное дополнение к команде time_signal(), которое учитывает передачу сообщений сегментации программы по точному методу сегментации времени/видеоинформации. Этот дескриптор может быть использован в командах, отличных от команды time_signal().

Устройства, которые не распознают какое-либо значение в любом поле, должны игнорировать сообщение и не должны выполнять никаких действий. См. таблицу 8-5.

Таблица 8-5/J.181 – Структура segmentation_descriptor()

Синтаксис	Биты	Мнемоника
segmentation_descriptor() { splice_descriptor_tag descriptor_length identifier segmentation_event_id segmentation_event_cancel_indicator reserved if(segmentation_event_cancel_indicator == "0") { program_segmentation_flag segmentation_duration_flag reserved if(program_segmentation_flag == "0") { component_count for (i = 0; i < component_count; i++) { component_tag reserved pts_offset } } if(segmentation_duration_flag == "1") segmentation_duration() segmentation_upid_type segmentation_upid_length segmentation_upid segmentation_type_id chapter chapter_count } }	 8 8 32 32 1 7 1 1 6 8 8 7 33 8 8 8 8 8 8	 uimsbf uimsbf uimsbf uimsbf bslbf bslbf bslbf bslbf bslbf uimsbf uimsbf bslbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf

8.3.3.1 Семантическое определение полей в структуре segmentation_descriptor()

splice_descriptor_tag (тег дескриптора монтажа): Это 8-битовое число, определяющее синтаксис для частных байтов, образующих тело данного дескриптора. Поле splice_descriptor_tag должно иметь значение 0x02.

descriptor_length (длина дескриптора): Это 8-битовое число, указывающее длину в байтах дескриптора, следующего за данным полем.

identifier (идентификатор): Это 32-битовое число, используемое для идентификации владельца дескриптора. Этот идентификатор должно иметь значение 0x43554549 (ASCII "CUEI").

segmentation_event_id (идентификатор события сегментации): Это 32-битовый однозначно определяемый идентификатор события сегментации.

segmentation_event_cancel_indicator (идентификатор аннулирования события сегментации): Это 1-битовый флаг, который, будучи установленным на "1", указывает на то, что ранее посланное событие сегментации, определяемое полем segmentation_event_id, было аннулировано.

program_segmentation_flag (флаг сегментации программы): Это 1-битовый флаг, который должен быть установлен на "1", указывая на то, что сообщение относится к точке сегментации программы и что режимом сегментации является режим сегментации программы, при котором должны сегментироваться все идентификаторы PID/компоненты программы. Будучи установленным на "0", это поле указывает на то, что режимом сегментации является режим сегментации компонентов, при котором каждый компонент, подлежащий сегментации, будет перечисляться отдельно в последующем синтаксисе. Поле program_segmentation_flag может быть установлено в различные состояния при передаче в пределах программы различных сообщений дескрипторов.

segmentation_duration_flag (флаг продолжительности сегментации): Это 1-битовый флаг, который должен быть установлен на "1", указывая на наличие поля структуры segmentation_duration().

component_count (подсчет компонентов): Это 8-битовое целое число без знака, определяющее число экземпляров данных элементарных PID-поток в последующем цикле. Компоненты эквиваленты элементарным PID-поток.

component_tag (тег компонента): Это 8-битовое значение, определяющее элементарный PID-поток, содержащий точку сегментации, задаваемую значением последующего поля splice_time(). Это значение должно быть таким же, как и значение, используемое в структуре stream_identification_descriptor() для идентификации этого элементарного PID-потока. Наличие этого поля в цикле компонента означает наличие этого компонента ресурсов.

pts_offset (смещение pts): Это 33-битовое целое число без знака, которое должно использоваться монтажным устройством как смещение, прибавляемое к значению поля pts_time в сообщении time_signal() для получения намечаемого времени монтажа. Когда это поле имеет нулевое значение, поле (поля) pts_time должно использоваться без смещения. Если флаг заданного времени поля pts_time равен нулю или если команда, которая переносится этим дескриптором, не имеет поля splice_time(), тогда это поле должно использоваться для смещения полученного времени немедленного монтажа.

segmentation_duration(): Структура segmentation_duration() определяет продолжительность сегмента программы. Она может быть использована для выдачи сплайсеру индикации о том, когда закончится сегмент и когда будет иметь место следующее сообщение сегментации. Для конечных сообщений эта продолжительность должна быть равна нулю. См. таблицу 8-6.

Таблица 8-6/J.181 – Структура segmentation_duration()

Синтаксис	Биты	Мнемоника
segmentation_duration() { reserved duration }	7 33	bslbf uimsbf

duration (продолжительность): Это 33-битовое поле, указывающее на истекшее время с точки зрения тактов генератора частоты 90 кГц программы.

segmentation_upid_type (тип уникального программного идентификатора сегментации): Значение из последующей таблицы. Имеется несколько допустимых типов, обеспечивающих возможность использования программаторами идентификатора, поддерживаемого их системами. Ожидается, что пользователи этих идентификаторов будут располагать внеполосным методом сбора других данных, связанных с этими номерами, и поэтому не требуется, чтобы это были номера идентичных типов. Эти идентификаторы могут находиться в других дескрипторах в программе, и когда используется один и тот же идентификатор (например, V-ISAN), он должен быть согласован между программами. При использовании с вводом монтажа поле **segmentation_upid** не может быть передано или не может предполагаться совпадающим с полем **unique_program_id** в сообщении splice_insert. См. таблицу 8-7.

Таблица 8-7/J.181 – Поле segmentation_upid_type

Тип	Длина в байтах	Имя	Описание
0x00	0	Не используется	Структура segmentation_upid не определена и в дескрипторе не присутствует.
0x01	переменная	Определяется пользователем	Структура segmentation_upid не подчиняется стандартной схеме именования.
0x02	8	ISCI	Структура segmentation_upid – это 8-значный код, который обычно используется для содержимого рекламы.
0x03	12	Ad-ID	Определяется цифровой идентификацией рекламы, группа LLC. Это по существу 12-значный идентификатор. http://www.Ad-id.org/
0x04	24	UMID	SMPTE 330M
0x05	8	ISAN	Стандарт ИСО для аудио/видеоматериала с использованием шестнадцатеричных цифр. http://www.nlc-bnc.ca/iso/tc46sc9/standard/15706e.htm
0x06	12	V-ISAN	Расширение стандарта ISAN путем добавления 8 шестнадцатеричных символов для идентификатора версии.
0x07	12	TID	Программный идентификатор мультимедийных систем Tribune, 12 символов, 2 буквы, за которыми следуют 10 цифр.
0x08	8	Источник	Идентификатор Turner Origin.
0x09–0x1F	переменная	Зарезервированы	Зарезервированы для стандартизации в будущем.

segmentation_upid_length (длина однозначно определяемого программного идентификатора сегментации): Длина в байтах поля **segmentation_upid**.

segmentation_upid (однозначно определяемый программный идентификатор сегментации): Поле для длины и идентификации из таблицы 8-7. Содержимое и длина этого поля определяются полями **segmentation_upid_type** и **segmentation_upid_length**. Примером может служить тип 0x06 для V-ISAN и длина в 12 байтов. Это поле может тогда содержать идентификатор V-ISAN для содержимого, на которое ссылается данный дескриптор.

segmentation_type_id (идентификатор типа сегментации): Это 8-битовое значение должно содержать одно из значений, приведенных в таблице 8-8, для обозначения типа сегментации. Все неиспользованные значения резервируются.

Таблица 8-8/J.181 – Поле segmentation_type_id

Сообщение сегментации	Поле segmentation_type_id
Начало программы	0x10
Конец программы	0x11
Раннее завершение программы	0x12
Аварийное прекращение программы	0x13
Возобновление программы	0x14
Запланированный просмотр программы	0x15
Незапланированный просмотр программы	0x16
Начало главы	0x20
Конец главы	0x21
Начало национальной рекламы	0x30
Конец национальной рекламы	0x31
Начало местной рекламы	0x32

Таблица 8-8/J.181 – Поле segmentation_type_id

Сообщение сегментации	Поле segmentation_type_id
Конец местной рекламы	0x33
Начало незапланированного события	0x40
Конец незапланированного события	0x41

chapter (сегмент): Это поле обеспечивает идентификацию конкретного сегмента в поле segmentation_unique_program_id. Это значение, как ожидается, возрастает на единицу с каждым новым сегментом в событии сегментации. Требуется, чтобы это значение было переустановлено на единицу для первого сегмента в новом событии просмотра. Это поле должно увеличиваться на единицу для каждого нового сегмента.

chapter_count (подсчет сегментов): Это поле обеспечивает подсчет ожидаемого числа отдельных сегментов в текущем событии сегментации.

9 Шифрование

9.1 Обзор

Структура splice_info_section поддерживает шифрование части сегмента, чтобы можно было предотвратить доступ к коммерческому времени для всех приемников, кроме тех, которым разрешен доступ к этому коммерческому времени. В этом пункте дается описание различных алгоритмов шифрования, которые могут быть использованы. Шифрование сегмента является факультативным, поскольку является реализацией шифрования либо формирователем сообщения, либо любыми приемными устройствами. Считается, что использование шифрования является факультативным, чтобы позволить изготовителю транспортировать "незакодированные" системы, не заботясь об экспорте технологии шифрования. Если в систему включено шифрование, то любое приемное устройство должно реализовать все алгоритмы, перечисленные в настоящей Рекомендации, что дает возможность разработчику таблицы с монтажной информацией использовать любой из этих алгоритмов при передаче. Использование частной технологии шифрования является факультативным и в настоящей Рекомендации не рассматривается.

9.2 Шифрование с фиксированным ключом

В шифровании, применяемом в настоящей Рекомендации, предполагается использование фиксированного ключа. Один и тот же ключ предоставляется как передатчику, так и приемнику. Метод доставки ключа всем сторонам не задан. В настоящей Рекомендации допускается до 256 различных ключей, которые могут использоваться для расшифровывания. Поле cw_index используется для определения того, какой ключ должен использоваться при расшифровывании сегмента. Длина фиксированного ключа зависит от типа используемого алгоритма. Предполагается, что фиксированный ключ, доставляемый всем сторонам, будет иметь правильную длину для алгоритма, который предназначен для использования.

9.3 Алгоритмы шифрования

Поле encryption_algorithm (алгоритм шифрования) сегмента splice_info_section является шестибитовым значением, которое может содержать одно из значений, приведенных в таблице 9-1. Во всех вариантах стандарта шифрования данных для шифрования и расшифровывания блока из 8 байтов используется 64-битовый ключ (фактически 56 битов плюс контрольная сумма). В случае "тройного" стандарта DES будут необходимы три 64-битовых ключа, по одному на каждый из трех проходов алгоритма стандарта DES. В "типовом" тройном стандарте DES фактически используются два ключа, где первый и третий ключи идентичны. Соответствующая информация содержится в документах PUB 46-3 и PUB 81 стандарта FIPS.

Таблица 9-1/J.181 – Алгоритм шифрования

Значение	Алгоритм шифрования
0	Шифрование отсутствует
1	DES – режим ECB
2	DES – режим CBC
3	Тройной DES, режим EDE3 – ECB
4–31	Зарезервировано
32–63	Частный алгоритм пользователя

9.3.1 DES – режим ECB

В этом алгоритме используется "стандарт шифрования данных" (см. документ PUB 81 стандарта FIPS) в режиме электронной кодовой книги.

Чтобы использовать этот тип шифрования, зашифрованные данные должны содержать множество данных из 8 байтов начиная с поля `splice_command_type` и заканчивая полем `E_CRC_32`. Цикл поля `alignment_stuffing` может быть использован для заполнения любыми дополнительными байтами, которые могут потребоваться.

9.3.2 DES – режим CBC

В этом алгоритме используется "стандарт шифрования данных" (см. документ PUB 81 стандарта FIPS) в режиме цепочки шифрованных блоков. Основным алгоритм идентичен алгоритму DES – режим ECB. До выполнения шифрования с использованием ключа DES каждый 64-битовый блок открытого текста подвергается по каждому биту операции исключающего "ИЛИ" с предыдущим шифрованным блоком. Первый блок подвергается операции исключающего "ИЛИ" с начальным вектором. В настоящей Рекомендации начальный вектор будет иметь фиксированное нулевое значение.

Чтобы использовать этот тип шифрования, зашифрованные данные должны содержать множество данных из 8 байтов начиная с поля `splice_command_type` и заканчивая полем `E_CRC_32`. Цикл поля `alignment_stuffing` может быть использован для заполнения любыми дополнительными байтами, которые могут потребоваться.

9.3.3 Тройной DES, EDE3 – режим ECB

В этом алгоритме используются три 64-битовых ключа, при этом каждый ключ используется при одном проходе алгоритма DES – режим ECB. См. документ PUB 46-3 стандарта FIPS. Каждый блок данных в передающем устройстве сначала шифруется первым ключом, затем расшифровывается вторым ключом и в итоге шифруется третьим ключом. Каждый блок на приемной стороне сначала расшифровывается третьим ключом, затем шифруется вторым ключом и в итоге расшифровывается первым ключом.

Чтобы использовать этот тип шифрования, зашифрованные данные должны содержать множество данных из 8 байтов начиная с поля `splice_command_type` и заканчивая полем `E_CRC_32`. Цикл поля `alignment_stuffing` может быть использован для заполнения любыми дополнительными байтами, которые могут потребоваться.

9.3.4 Частные алгоритмы пользователя

В настоящей Рекомендации допускается использование частных алгоритмов шифрования. Как взаимодействуют передающие и приемные устройства при использовании алгоритма для любого частного кода пользователя, не определено. Также не определено, как осуществляется регистрация и управление действиями по координации частных значений для поля `encryption_algorithm`.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
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, аспекты межсетевого протокола и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи