



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

J.280

(03/2004)

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

Цифровая передача телевизионных сигналов

**Вставка цифровой программы:
Программный интерфейс приложения
монтажа**

Рекомендация МСЭ-Т J.280

Рекомендация МСЭ-Т J.280

Вставка цифровой программы: Программный интерфейс приложения монтажа

Резюме

В данной Рекомендации определяется прикладной программный интерфейс (API), который создает стандартизованный метод взаимодействия между серверами и монтажными устройствами для вставки содержания в любой выходной мультиплекс MPEG-2 в монтажном устройстве. Этот API достаточно гибок для поддержки одного или нескольких серверов, подключенных к одному или нескольким монтажным устройствам. Вставка цифровой программы включает такое содержание, как рекламные сообщения различной длины, замена программы, публичные объявления службы или программный материал, созданный путем разделения на части программы с сервера.

Источник

Рекомендация МСЭ-Т J.280 утверждена 15 марта 2004 года 9-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) в соответствии с процедурой, изложенной в Резолюции МСЭ-Т А.8.

Ключевые слова

Вставка программы, монтаж, монтажное устройство, сервер, API.

ПРЕДИСЛОВИЕ

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

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

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

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

ПРИМЕЧАНИЕ

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

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

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

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

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

© ITU 2005

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

СОДЕРЖАНИЕ

	Стр.
1	Область применения 1
2	Ссылки 1
2.1	Нормативные ссылки 1
2.2	Информативные ссылки..... 1
3	Определения 1
4	Сокращения 2
5	Обозначение соответствия 3
6	Введение..... 3
6.1	Блок-схема системы 3
6.2	Приоритеты для разрешения конфликтов 5
6.3	Аварийные завершения 6
6.4	Требования к монтажу 7
6.5	Взаимодействие 7
6.6	Дальнейшие исследования..... 7
7	Синтаксис API 7
7.1	Синтакс сообщения Splicing_API_Message..... 7
7.2	Соглашения и требования 9
7.3	Инициализация 10
7.4	Сообщения со встроенными метками..... 11
7.5	Сообщения монтажа 12
7.6	Сообщения о работе 15
7.7	Расширенные сообщения с данными 16
7.8	Сообщения аварийного завершения 17
7.9	Сообщение Abort_Request..... 18
7.10	Сообщение Abort_Response 18
7.11	Запрос параметров настройки конфигурации 18
7.12	Сообщение General_Response..... 18
8	Дополнительные структуры..... 19
8.1	Версия 19
8.2	Структура Hardware_Config..... 19
8.3	Структура splice_elementary_stream() 20
8.4	Определение поля time() 22
8.5	Определение поля splice_API_descriptor()..... 22
9	Синхронизация времени 25
10	Синхронизация систем 26
10.1	Поток сигналов монтажа DPI 26
10.2	Временная последовательность инициализации монтажа DPI 27

	Стр.
Добавление I – Коды результатов.....	29
ЛИТЕРАТУРА.....	31

Рекомендация МСЭ-Т J.280

Вставка цифровой программы: Программный интерфейс приложения монтажа

1 Область применения

В данной Рекомендации определяется прикладной программный интерфейс (API), который создает стандартизованный метод взаимодействия между серверами и устройствами монтажа для вставки содержания в любой выходной мультиплекс MPEG-2 в монтажном устройстве. Этот API достаточно гибок для поддержки одного или нескольких серверов, подключенных к одному или нескольким монтажным устройствам. Вставка цифровой программы включает такое содержание, как рекламные сообщения различной длины, замена программы, публичные объявления службы или программный материал, созданный монтажными частями программы с сервера.

Данная Рекомендация не поддерживает команды и управление операциями компоновки или редактирования (например, наложенные изображения или эффекты сжатия изображения), в ней не определяются порядок выполнения монтажа или достигаемый уровень непрерывности. Кроме того, важная проблема синхронизации асинхронных монтируемых потоков является элементом реализации, оставленным на усмотрение монтажного устройства [3].

2 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.

2.1 Нормативные ссылки

- [1] ITU-T Recommendation H.222.0 (2000) | ISO/IEC 13818-1:2000, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.
- [2] ITU-T Recommendation H.262 (2000) | ISO/IEC 13818-2:2000, *Information technology – Generic coding of moving pictures and associated audio information: Video*.
- [3] ITU-T Recommendation J.181 (2001), *Digital Program Insertion cueing message for cable television systems*.

2.2 Информативные ссылки

- [4] Appendix I to ITU-T Recommendation J.181 (2003), *Recommended practices and interpretation guide*.

3 Определения

В данной Рекомендации определяются следующие термины:

- 3.1 соединение API:** Соединение по каналу TCP/IP между сервером и монтажным устройством для передачи сообщений API.
- 3.2 вставка "конец к концу":** Два или более непрерывных во времени сеанса без возврата к первичному каналу между сеансами.

- 3.3 канал:** Канал – это синоним термина "услуга" (Service) в терминологии DVB или термина "программа" (Program) в терминологии MPEG.
- 3.4 канал вставки:** Мультиплексный канал (каналы) вставки, который заменяет первичный канал целиком или частично на время события монтажа.
- 3.5 мультиплекс вставки:** Это источник канала вставки. При некоторых обстоятельствах полученный сервером мультиплекс может исключать информацию PSI, следовательно, предполагается, что этот мультиплекс может быть несовместимым с MPEG-2 транспортным потоком.
- 3.6 мультиплекс:** Мультиплекс – это набор из одного или нескольких каналов, который может включать соответствующую служебную информацию. Мультиплекс – это транспортный поток MPEG-2 с возможным исключением мультиплекса вставки.
- 3.7 выходной канал:** Канал, который получается на выходе монтажного устройства.
- 3.8 выходной мультиплекс:** Транспортный поток MPEG-2, получаемый путем мультиплексирования одного или нескольких выходных каналов. Монтажное устройство всегда должно обеспечивать, чтобы выходной мультиплекс имел достоверную PSI.
- 3.9 первичный канал:** Первичный мультиплексный канал, который замещается целиком или частично. Один первичный канал может иметь несколько выходных каналов.
- 3.10 первичный мультиплекс:** Это источник первичного канала (каналов).
- 3.11 сервер:** Устройство, которое инициирует монтаж канала (каналов) вставки в первичный канал (каналы). Это устройство обменивается информацией с монтажным устройством относительно времени начала монтажа и его объекта.
- 3.12 сеанс:** Сеанс – это вставка содержания (такого, как рекламные сообщения различной длины, замена программы, публичные объявления службы или программный материал, созданный монтажными частями программы с сервера). Каждый сеанс идентифицируется уникальным идентификатором **SessionID**.
- 3.13 входная монтажная точка:** Монтаж в начале вставки. Он происходит в момент, указанный в сообщении **Splice_Request**.
- 3.14 выходная монтажная точка:** Монтаж в конце вставки. Ожидаемое время завершения вставки рассчитывается путем сложения времени начала монтажа и его продолжительности, указанной в сообщении **Splice_Request**; однако завершение вставки может произойти раньше из-за наличия условий ошибки.
- 3.15 монтажное устройство:** Устройство, которое монтирует канал (каналы) вставки в первичный канал (каналы). Оно может получать сообщения с меткой по J.181. Это устройство также обменивается информацией с сервером относительно времени начала монтажа и его объекта.

4 Сокращения

В данной Рекомендации используются следующие сокращения:

API	Интерфейс прикладного программирования
CNN	Сеть кабельного вещания
DVB-ASI	Цифровое телевизионное вещание – асинхронный последовательный интерфейс
ID	Идентификатор
ИСО	Международная организация по стандартизации
МСЭ	Международный союз электросвязи
MPEG	Экспертная группа по движущимся изображениям
MPTS	Многопрограммный транспортный поток
NCTA	Национальная ассоциация кабельного телевидения

NTP	Сетевой протокол системного времени
PAT	Таблица взаимосвязи программ
PCR	Эталонное время программы
PID	Идентификатор пакета
PMT	Таблица преобразования программ
PSI	Относящаяся к программе информация
SCTE	Общество инженеров по кабельной связи
TCP/IP	Протокол управления передачей – межсетевой протокол (IP)
uimsbf	Целое число без знака, самый старший бит записывается первым
UTC	Универсальное скоординированное время

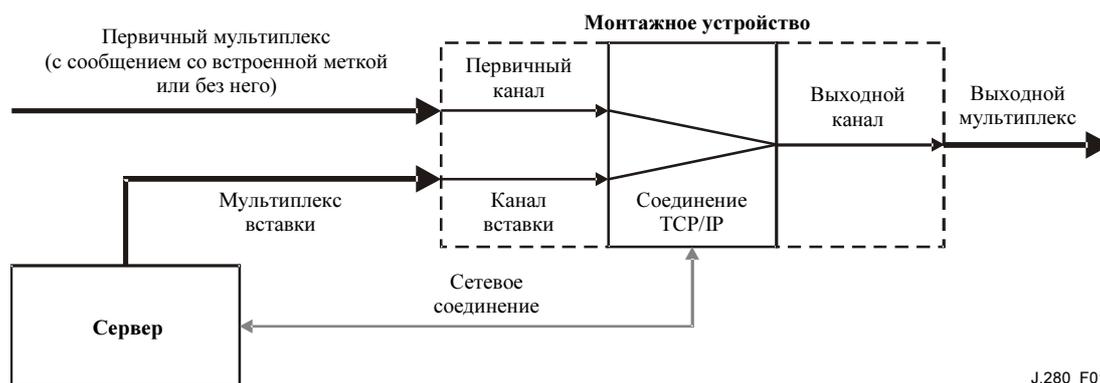
5 Обозначение соответствия

При использовании в данной Рекомендации слово "должен" обозначает обязательное положение этого стандарта. Слово "следует" обозначает рекомендуемое, но не обязательное положение. Слово "может" означает возможность, наличие которой не исключает соответствия и которая может присутствовать или отсутствовать по выбору производителя.

6 Введение

6.1 Блок-схема системы

Данный API может использоваться со многими различными конфигурациями сервера (серверов) и монтажного устройства (устройств). Данный API сконцентрирован на конфигурации с одним сервером и одним монтажными устройством, показанной на рисунке 1. Однако она может быть расширена до конфигурации с несколькими серверами и несколькими монтажными устройствами, как показано на рисунке 2.



J.280_F01

Рисунок 1/J.280 – Один сервер/одно монтажное устройство

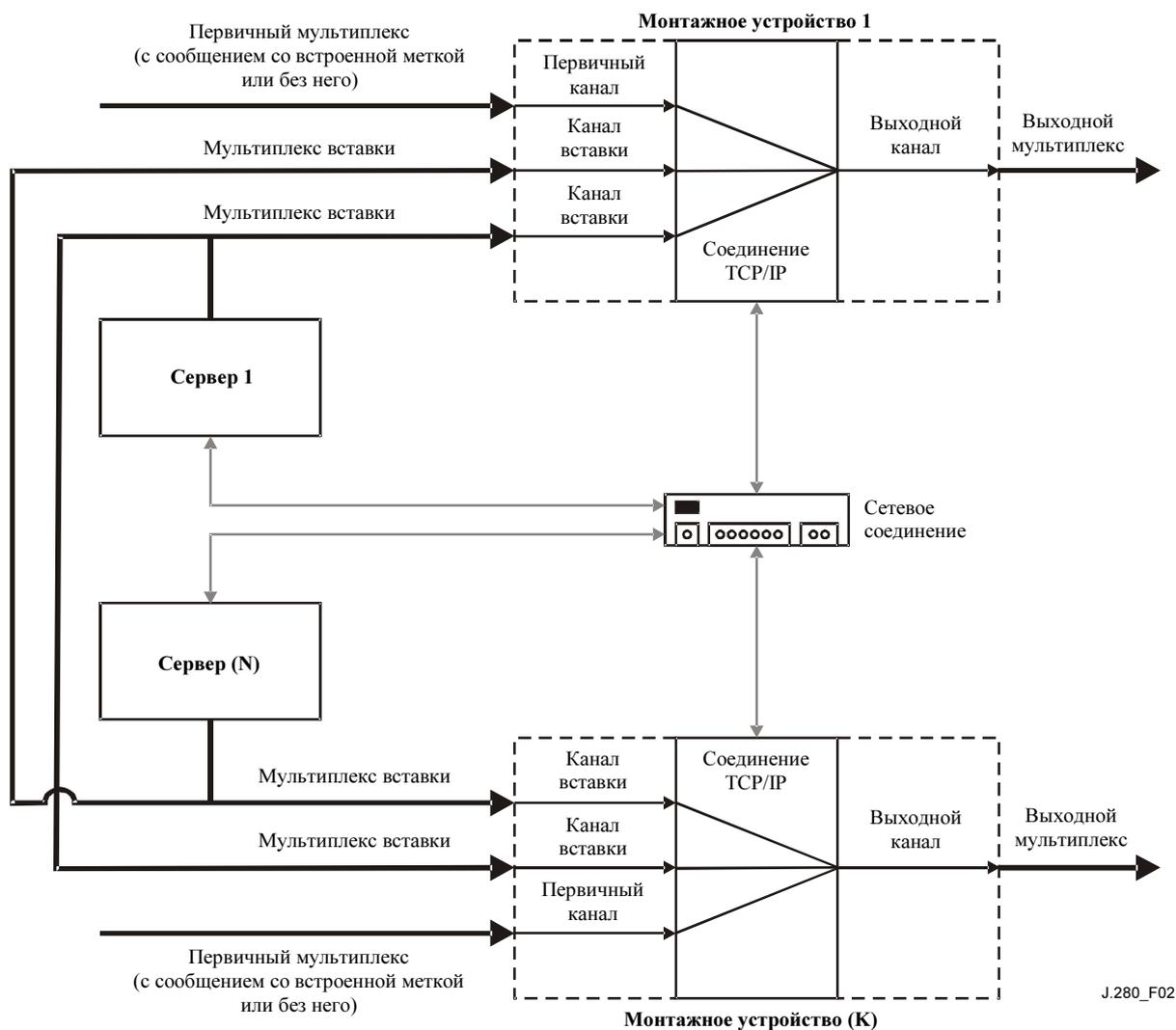


Рисунок 2/J.280 – Несколько серверов/несколько монтажных устройств

Модель в данном API содержит монтажное устройство, которое имеет один или несколько мультиплексных входов. Монтажное устройство логически отделяет канал (каналы) в мультиплексе (мультиплексах) и предоставляет этот канал (каналы) коммутатору. Данный коммутатор может преобразовать любой входной сигнал в любой выходной канал. При первоначальной конфигурации первичный канал (каналы) преобразуется в выходной канал (каналы). При этом сервер может передать монтажному устройству команду переключения с первичного канала на канал вставки на указанное время. Затем он может передать монтажному устройству команду переключения на другой канал вставки после первоначального переключения.

Логически монтаж включает два входных канала и один выходной канал. Монтажное устройство отвечает за соединение различных элементарных потоков (аудио, видео и данных). Оптимальные монтажные точки отдельных элементарных потоков могут немного различаться по времени, поэтому монтажное устройство должно производить монтаж, который обеспечит выход с наилучшим качеством. Монтаж не всегда может выполняться из "программной сети" первичного канала к рекламному сообщению "канала вставки" с возвращением к "программной сети" первичного канала. Монтажное устройство может смонтировать содержание, которое хранится только на сервере и поступает по одному входному мультиплексу. Можно использовать этот API в ситуации, когда сервер имеет один выход многопрограммного транспортного потока (MPTS), который содержит программу и промежуточный материал и использует монтажное устройство для создания требуемого монтажа между частями содержания.

Данный API поддерживает все комбинации с одним или несколькими серверами, взаимодействующими с одним или несколькими монтажными устройствами. Отдельное соединение API соответствует каждому выходному каналу.

В некоторых конфигурациях к монтажному устройству может быть подключено либо несколько серверов, либо несколько каналов с мультиплексом вставки. В этих случаях монтажное устройство будет иметь несколько соединений API с выходным каналом. Если в первичном канале получено сообщение с меткой по J.181, то требуется передать сообщение **Cue_Request** серверам по всем соединениям API, которые были созданы для соответствующего выходного канала (каналов). Кроме того, возможно, что одновременно по нескольким соединениям API для одной и той же вставки для выходного канала будет транспортироваться сообщение **Splice_Request**.

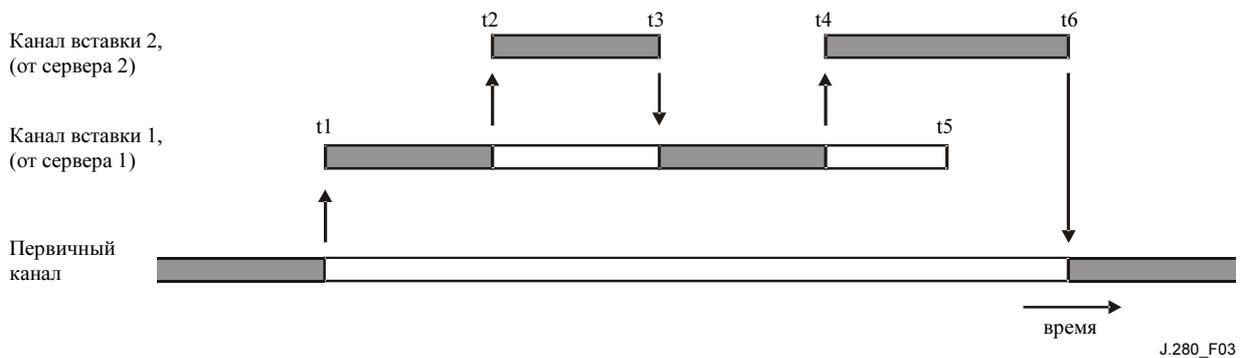
6.2 Приоритеты для разрешения конфликтов

Для обеспечения подключения соответствующего канала вставки используются различные уровни доступа. Существует 10 различных уровней доступа – от 0 до 9. Уровень 9 означает наивысший приоритет, он может отменить любое соединение с более низким приоритетом. Флаг **OverridePlaying** (отмена воспроизведения) в сообщении **Splice_Request** (запрос монтажа) определяет, будет ли выполняться запрос вставки, когда монтажное устройство в текущий момент ставит в очередь или осуществляет вставку. Если флаг установлен на 1, то вставка с более высоким приоритетом может прервать воспроизведение вставки с таким же или более низким приоритетом. Если флаг установлен на 0, то монтажное устройство не будет заменять воспроизводимую в настоящий момент вставку, даже если новый запрос имеет более высокий приоритет.

Сообщение **Splice_Request** следует передавать по крайней мере за три секунды до времени монтажа `time()`, чтобы оно было действительным. Если минимальный период в три секунды не обеспечен, то результат сообщения **Splice_Request** не определяется в этом API. Если несколько серверов одновременно инициализируют запросы монтажа с одинаковым приоритетом, монтажное устройство примет решение о том, какой запрос должен быть принят, по принципу "первым пришел – первым обслужен". Все остальные запросы будут отклонены, а в ответном сообщении **Splice_Response** (ответ о монтаже) будет передана ошибка конфликта (если не был установлен флаг **OverridePlaying**).

В период, непосредственно предшествующий инициализации вставки, справедливо следующее. Если запрос **Splice_Request** с приоритетом 5 получен в течение того же времени монтажа, что и запрос **Splice_Request** с приоритетом 3, то ошибка конфликта передается в обратном направлении для запроса с приоритетом 3. Если позднее в течение того же времени монтажа поступает запрос **Splice_Request** с приоритетом 7, то ошибка конфликта передается для запроса с приоритетом 5, а запрос с приоритетом 7 ставится в очередь. Если поступает второй запрос с приоритетом 7 с флагом **OverridePlaying**, установленным на 0, то второй запрос с приоритетом 7 получает ошибку конфликта. Однако если во втором запросе с приоритетом 7 флаг **OverridePlaying** установлен на 1, то первоначальный запрос с приоритетом 7 получает ошибку конфликта и отменяется.

На рисунке 3 показаны три входа монтажного устройства. Заштрихованные области показывают, какой входной источник будет направлен в выходной канал в любой заданный момент времени.



J.280_F03

- t1 – Сервер 1 выдает сообщение запроса **Splice_Request** и начинает передавать свой поток в монтажное устройство. Монтажное устройство переключает этот поток канала вставки на выходной канал. В сообщении **Splice_Request** запрашивается продолжительность вставки t1–t5. Монтажное устройство должно передать серверу 1 сообщение ответа **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_in** (входная монтажная точка) и кодом результата, равным 100 – "успешный ответ".
- t2 – Сервер 2 выдает сообщение запроса **Splice_Request** с флагом **OverridePlaying**, установленным на 1, и таким же или более высоким приоритетом. Во время, указанное в запросе **Splice_Request**, поток канала вставки 2 переключается на выходной канал (заменяя текущий поток от сервера 1). В сообщении сервера 2 **Splice_Request** запрашивается продолжительность t2–t3. Монтажное устройство должно передать серверу 1 сообщение ответа **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_out** (выходная монтажная точка), и кодом результата, равным 125 – "отмена канала". Монтажное устройство должно передать серверу 2 сообщение **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_in**, и кодом результата, равным 100 – "успешный ответ". Если сервер 1 определит, что отмена канала является ошибкой, он может передать запрос **Abort_Request** и закончить передачу своего потока в этот момент времени. Такое поведение не показано на рисунке 3.
- t3 – Время вставки прошло, и монтажное устройство возвращается к материалу от сервера 1 для направления его в выходной канал. Следует отметить, что монтажное устройство не возвращается к первичному каналу для направления его в выходной канал. Монтажное устройство должно передать серверу 1 сообщение ответа **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_in**, и кодом результата, равным 125 – "отмена канала". Монтажное устройство должно передать серверу 2 сообщение **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_out**, и кодом результата, равным 100 – "успешный ответ".
- t4 – Сервер 2 выдает другое сообщение запроса **Splice_Request** с флагом **OverridePlaying**, установленным на 1. Во время, указанное в запросе **Splice_Request**, поток канала вставки 2 переключается на выходной канал (заменяя текущий поток от сервера 1). В сообщении сервера 2 **Splice_Request** запрашивается продолжительность t4–t6. Монтажное устройство должно передать серверу 1 сообщение ответа **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_out**, и кодом результата 125 – "отмена канала". Монтажное устройство должно передать серверу 2 сообщение **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_in**, и кодом результата 100 – "успешный ответ".
- t5 – Поток вставки сервера 1 заканчивается двумя воспроизведенными и двумя отмененными потоком сервера 2 порциями.
- t6 – Продолжительность последней вставки завершена и монтажное устройство возвращается к материалу первичного канала для направления в выходной канал. Монтажное устройство должно передать серверу 2 сообщение ответа **SpliceComplete_Response** с флагом **SpliceType**, установленным на **Splice_out**, и кодом результата, равным 100 – "успешный ответ".

Рисунок 3/J.280 – Действие флага **OverridePlaying**

Также возможно, что несколько серверов пожелают разбить сообщение **Cue_Request** (запрос метки). В качестве примера рассмотрим возможность 60-секундного монтажа, при которой один сервер будет использовать первые 30 секунд, а второй сервер будет использовать последние 30 секунд. В зависимости от приоритетов и времени получения сообщений запроса **Splice_Request** монтажное устройство должно указать код результата 109 (Конфликт монтажа) при его наличии. Данный API не координирует способность этих двух серверов выполнять такие функциональные возможности. Они могут быть обеспечены по взаимному соглашению между серверами или API сервер–сервер.

6.3 Аварийные завершения

Возможно, что вставка в какой-либо момент в ходе воспроизведения будет отменена вставкой с более высоким приоритетом. В этом случае монтажное устройство должно возвратиться к отмененной вставке по окончании вставки с более высоким приоритетом. Если вставка с более высоким приоритетом прервана сообщением **Abort_Request** (запрос аварийного завершения), монтажное устройство должно возвратиться к отмененной вставке. Если начальный канал вставки стал недоступен, то монтажное устройство должно возвратиться к первичному каналу, если это возможно.

Если сервер запрашивает монтаж в первичном канале, который в настоящее время не имеет допустимого ввода, монтажное устройство должно выполнить монтаж, но передать в сообщении ответа **SpliceComplete_Response** серверу код результата 111 (Первичный канал не найден).

Аналогичным образом возвращение с канала вставки на первичный канал, который не имеет допустимого ввода, должно завершиться с кодом результата 111 (Первичный канал не найден).

Монтажное устройство может использовать дополнительное программное обеспечение для постоянного гарантированного возврата монтажного устройства к первичному каналу. Весьма желательно иметь монтажное устройство, которое переключается на первичный канал при любом состоянии ошибки, которое вызывает прекращение передачи выходного канала. В этом случае монтажное устройство должно возвращать ответ **SpliceComplete_Response** с кодом результата 110 (Канал вставки не найден).

6.4 Требования к монтажу

Монтажному устройству требуется информация о канале вставки прежде, чем он может быть смонтирован в первичный канал. Часть этой информации должна передаваться по соединению API, а часть может передаваться в мультиплексе MPEG. Вся информация требуется перед монтажом.

Имя канала **ChannelName** используется для идентификации выходного канала. Это уникальное имя, присвоенное каждому выходному каналу (например, CNN) в монтажном устройстве; оно необходимо серверу для определения того, какой первичный канал должен быть заменен каждым каналом вставки.

Монтажное устройство должно знать, какой канал вставки должен монтироваться в первичный канал. Эта информация включает местоположение мультиплекса вставки и то, какой канал должен использоваться в мультиплексе вставки. Такая информация имеется в сообщении **Splice_Request**.

6.5 Взаимодействие

Связь между сервером и монтажным устройством осуществляется по одному соединению двунаправленного канала TCP/IP на один выходной канал. Двунаправленный канал TCP/IP определяется IP-адресом и номером порта TCP сервера и монтажного устройства. IP-адрес уникален для каждого сервера или монтажного устройства, а номер порта TCP уникален для каждой монтируемой программы. Следовательно, для каждого выходного канала однозначно определяются сервер рекламы, монтажное устройство и соответствующее соединение. Раз данное соединение API установлено, оно остается установленным до тех пор, пока одно из устройств не разъединит соединение API, в это время необходима повторная инициализация для возобновления монтажа.

Все сообщения, которыми обмениваются монтажное устройство и сервер, имеют совместно используемый общий формат, который детально описан в подразделе 7.1. Для связи между монтажным устройством и сервером должны использоваться только сообщения, соответствующие этому формату. Этот формат допускает класс сообщений "определяемого пользователем" типа, которые могут использоваться в качестве шаблона для обмена частными сообщениями с данными между сервером и монтажным устройством, которые выходят за рамки настоящей Рекомендации.

Все сообщения запроса требуют ответа либо от монтажного устройства, либо от сервера в зависимости от того, какое устройство формирует запрос. В большинстве сообщений ответа указывается только результат и не содержатся какие-либо другие данные. Они необходимы для того, чтобы уверить запросившую систему, что ее сообщение было получено и правильно интерпретировано. При наличии ошибок сообщение может быть передано повторно.

6.6 Дальнейшие исследования

В данной Рекомендации определяется, что минимальная задержка при планировании монтажа между получением запроса монтажа и фактической операцией монтажа должна составлять три секунды. Определение, параметры и приложение "немедленного запроса монтажа" с существенно меньшей задержкой составят предмет дальнейшего изучения.

7 Синтаксис API

7.1 Синтаксис сообщения Splicing_API_Message

Все сообщения в этом API содержат общую структуру сообщения, в которую внедряются данные определенного передаваемого сообщения. Это сделано для того, чтобы при получении сообщения общая подпрограмма синтаксического анализа могла сохранить сообщение, определить структуру его данных и гарантировать, что сообщение было получено правильно.

Таблица 7-1/J.280 – Splicing_API_Message

Синтаксис	Байты	Тип
Splicing_API_Message {		
MessageID	2	uimsbf
MessageSize	2	uimsbf
Result	2	uimsbf
Result_Extension	2	uimsbf
data()	*	*
}		

MessageID (Идентификатор сообщения) – целочисленное значение, которое указывает на то, какое сообщение передается. См. таблицу 7-2.

MessageSize – Длина в байтах передаваемого поля data().

Result – Результаты для требуемого сообщения. Подробное описание кодов результатов приведено в Добавлении I. Для сообщений запроса его значение равно 0xFFFF.

Result_Extension – Расширение результата должно быть установлено равным 0xFFFF, если не используется для передачи дополнительной информации результата в сообщении ответа.

data() – Специальная структура данных для передаваемого сообщения. Ниже приведено подробное описание каждого сообщения, содержащего данные. Размер этого поля равен **MessageSize** и определяется длиной данных, добавляемых в сообщение. Поле data() используется не во всех сообщениях.

Таблица 7-2/J.280 – Значения идентификатора MessageID

MessageID	Имя сообщения	Отправлено	Описание
0x0000	General_Response	Монтажным устройством или сервером	Используется для передачи асинхронной информации между устройствами. Отсутствует связанное с этим сообщением поле data().
0x0001	Init_Request	Сервером	Начальное сообщение монтажному устройству по порту 6021.
0x0002	Init_Response	Монтажным устройством	Первоначальный ответ серверу по установленному соединению.
0x0003	ExtendedData_Request	Сервером	Запрос детальной информации воспроизведения от монтажного устройства.
0x0004	ExtendedData_Response	Монтажным устройством	Уникальный для производителя ответ с расширенными данными воспроизведения для требуемого события воспроизведения.
0x0005	Alive_Request	Сервером	Передает сообщение "Alive", чтобы получить текущий статус.
0x0006	Alive_Response	Монтажным устройством	Ответ на сообщение "Alive", указывающее текущий статус.
0x0007	Splice_Request	Сервером	Запрос монтажа в определенное время.
0x0008	Splice_Response	Монтажным устройством	Ответ, который указывает, что сообщение Splice_Request было получено и что монтажное устройство готовится к монтажу.
0x0009	SpliceComplete_Response	Монтажным устройством	Ответ на входную монтажную точку и выходную монтажную точку.
0x000A	GetConfig_Request	Сервером	Запрос на получение текущей конфигурации монтажа для данного соединения API.

Таблица 7-2/J.280 – Значения идентификатора MessageID

MessageID	Имя сообщения	Отправлено	Описание
0x000B	GetConfig_Response	Монтажным устройством	Содержит всю информацию о монтаже для соединения API.
0x000C	Cue_Request	Монтажным устройством	Передача монтажным устройством секции информации метки на сервер.
0x000D	Cue_Response	Сервером	Подтверждение получения секции информации метки.
0x000E	Abort_Request	Сервером	Запрос немедленного возвращения на первичный канал или отмененный канал вставки.
0x000F	Abort_Response	Монтажным устройством	Подтверждение получения сообщения Abort_Request . В случае необходимости должен также формироваться ответ SpliceComplete_Response .
0x0010 - 0x7FFF 0xFFFF	Зарезервировано		Диапазон, зарезервированный для стандартизации в будущем.
0x8000 - 0xFFFFE	Определяемое пользователем		Диапазон, доступный для определяемых пользователем функций.

7.2 Соглашения и требования

- 1) Каждое содержащее данные сообщение описывается ниже с помощью его полей данных и типов. Дополнительные структуры обозначены как функции, их описания приведены в разделе 8.
- 2) В длине каждой строки зарезервировано место для символа завершающего нуля, и должны использоваться только строки с завершающим нулем. Например, строка, которая определена как 16 символов, может иметь максимум 15 символов данных, за последним из которых сразу следует символ завершающего нуля (0x00). Как только при сканировании строки встречается нуль, остальные символы в строке не определяются. Определенный для строки размер является постоянным и не будет изменяться в зависимости от длины строки. В настоящей Рекомендации в строках используются 8-битовые символы ASCII.
- 3) Все значения времени используются в формате UTC.
- 4) В данной Рекомендации для состояния DON'T CARE (НЕ ИМЕЕТ ЗНАЧЕНИЯ) используются последовательности единиц. Для 4-байтового поля это значение равно 0xFFFFFFFF.
- 5) Сообщение ответа должно посылаться без ненужной задержки. Устройство, ожидающее ответ, должно считать отсутствие ответа в течение 5 секунд указанием тайм-аута. Если сервер подозревает тайм-аут, он должно отправить сообщение **Alive_Request**. Если монтажное устройство не отвечает так, как определено в данной Рекомендации, соединение для этого канала должно быть отключено и установлено вновь.
- 6) Сервер, получающий сообщение ответа, означающее неудачный анализ сообщения (код ошибки 123), должен отправить сообщение **Alive_Request**. Если он не получит соответствующее сообщение **Alive_Response**, соединение для этого канала должно быть отключено и восстановлено.
- 7) Поле результата в сообщении **Splicing_API_Message** используется для возврата кода результата. Можно вернуть несколько кодов ответа, отправив несколько сообщений **General_Response** в любое время.
- 8) Если монтажное устройство или сервер не сможет разобрать сообщение запроса, оно должно передать в обратном направлении сообщение **General_Response** с кодом результата 123.

7.3 Инициализация

Первоначальный обмен сообщениями начинается с того, что монтажное устройство прослушивает предварительно определенный порт **6021**, а сервер устанавливает соединение API с монтажным устройством. Сервер передает монтажному устройству сообщение **Init_Request** (запрос инициализации). Затем сервер слушает ответ от монтажного устройства по установленному соединению API. Вся дальнейшая связь между сервером и монтажным устройством происходит по этому соединению API. Монтажное устройство или сервер может завершить связь, разъединив это соединение API. Каждое устройство ответственно за обнаружение и соответствующую обработку разъединенного соединения API. Когда монтажное устройство инициализирует приемник TCP по порту 6021, оно должно поддерживать втрое большее, чем число каналов вставки, количество соединений API с монтажным устройством. Например, если монтажное устройство управляет 70 каналами, в 40 из которых может проводиться монтаж, то оно должно поддерживать 120 (40×3) одновременных соединений API.

7.3.1 Сообщение Init_Request

Поле data() этого сообщения содержит структуру Init_Request_Data, описанную в таблице 7-3.

Таблица 7-3/J.280 – Init_Request_Data

Синтаксис	Байты	Тип
Init_Request_Data { Version() ChannelName SplicerName Hardware_Config() for (i=0; i<N; i++) splice_API_descriptor() }		
	32	Строка
	32	Строка

Version() – См. подраздел 8.1.

ChannelName – Логическое имя, присвоенное выходному каналу этого соединения. Оно также используется для проверки правильности соединения API, когда монтажное устройство отвечает серверу.

SplicerName – Имя монтажного устройства, если сервер использует API для связи с устройством, которое управляет несколькими монтажными устройствами.

Hardware_Config() – См. подраздел 8.2.

splice_API_descriptor() – Дескриптор, который должен следовать синтаксису, определенному в подразделе 8.5. Дескриптор missing_Primary_Channel_action_descriptor() является подходящим дескриптором для этого запроса.

7.3.2 Сообщение Init_Response

После отправки запроса **Init_Request** монтажное устройство передает сообщение **Init_Response** (ответ инициализации) по установленному соединению API. Сервер проверяет, что посланная монтажным устройством версия поддерживается и что он имеет соединение API с правильным первичным каналом.

Поле data() этого сообщения содержит структуру Init_Response_Data, описанную в таблице 7-4.

Таблица 7-4/J.280 – Init_Response_Data

Синтаксис	Байты	Тип
<pre> Init_Response_Data { Version() ChannelName } </pre>	32	Строка

Version() – См. подраздел 8.1. Монтажное устройство должно указать в ответе самый большой номер версии API, которую оно может поддерживать.

ChannelName (имя канала) – Возвращается серверу для индикации правильного соединения.

7.4 Сообщения со встроенными метками

Монтажные устройства могут иметь возможность получать сообщения со встроенными метками, основанные на Рекомендации МСЭ-Т J.181. Как только эти сообщения с метками поступят на монтажное устройство, их необходимо сразу передать на сервер. Сообщение **Cue_Request** используется для передачи сообщений с метками с монтажного устройства на сервер. Когда монтажное устройство получает сообщение с меткой, оно посылает всю структуру splice_info_section() наряду с указанием времени монтажа на сервер. Сервер подтверждает прием этого сообщения сообщением **Cue_Response**. Сообщение **Cue_Response** состоит только из сообщения Splicing_API_Message и не имеет присоединенных данных data(), но может иметь код возврата. Монтажное устройство расшифровывает splice_info_section() перед отправкой ее на сервер, если она зашифрована.

Если монтажное устройство получает поврежденное сообщение с меткой, оно должно передать серверу сообщение **General_Message** (общее сообщение) с кодом результата 117 (Недействительное сообщение с меткой). В этом случае монтажное устройство не должно отправлять сообщение **Cue_Request**.

7.4.1 Сообщение Cue_Request

Поле data() этого сообщения содержит структуру Cue_Request_Data, описанную в таблице 7-5.

Таблица 7-5/J.280 – Cue_Request_Data

Синтаксис	Байты	Тип
<pre> Cue_Request_Data { time() splice_info_section() } </pre>		

time() – Это время получается из splice_time() в структуре splice_info_section() сообщения с меткой по J.181 монтажным устройством. Если в splice_info_section по J.181 используется режим компонентного монтажа, то time() относится к значению времени монтажа по умолчанию, которое подробно описано в подразделе 7.5.2.1/J.181. В том случае, когда структура splice_info_section() не содержит pts_time(), которое требует преобразования, как в команде splice_schedule(), тогда структура времени должна быть заполнена всеми единицами, чтобы обозначить, что время не указано. Монтажное устройство должно решить, как преобразовать время PTS в UTC для связи с сервером. Преобразование может быть различным в различных монтажных устройствах, чтобы они могли надлежащим образом управлять их внутренними буферами. Описание синтаксиса структуры time() приведено в подразделе 8.4.

splice_info_section() – Подробное описание этой структуры приведено в Рекомендации МСЭ-Т J.181.

7.5 Сообщения монтажа

После инициализации и конфигурирования монтажного устройства сервер может выдать сообщение **Splice_Request** для инициализации сеанса. В ответ на сообщение **Splice_Request** передаются два сообщения – сообщение **Splice_Response** и сообщение **SpliceComplete_Response** (ответ "монтаж завершен"). Сервер должен послать сообщение **Splice_Request** по крайней мере за 3 секунды до времени `time()`, указанного в сообщении **Splice_Request**. Это позволяет монтажному устройству настроить свою конфигурацию и приготовиться к монтажу. Поток канала вставки для сеанса должен начинаться в пределах 300–600 миллисекунд до времени `time()`. Эталонное время программы (PCR) должно передаваться в первом блоке видеодоступа потока канала вставки или перед ним. Видеопоток вставки должен начинаться с заголовка последовательности и I-кадра. Монтажное устройство должно допускать очередь, состоящую минимум из 10 сообщений **Splice_Request**, для данного соединения API. Если очередь сообщений монтажного устройства полна, он направит в ответном сообщении код результата 114 (Очередь монтажа полна).

Детали физического соединения передаются в сообщении **Init_Request**. Существует два способа указать используемые каналы в мультиплексе вставки и PID:

- Если идентификатор **ServiceID** (идентификатор услуги) в сообщении **Splice_Request** не равен 0xFFFF, то поле **ServiceID** определяет номер программы в PAT, который указывает на соответствующую PMT. PAT и PMT должны устанавливаться в канале вставки по крайней мере за 200 мс до отправки сообщения **Splice_Request** и должны сохраняться в течение сеанса. Они должны быть допустимыми таблицами MPEG с соответствующими связанными с пересмотрами приращениями.
- Если идентификатор **ServiceID** равен 0xFFFF, то в сообщении **Splice_Request** используется структура `splice_elementary_stream()` (PCR, PID видео, аудио и данных).

ПРИМЕЧАНИЕ. – Если используется этот метод, то идентификатор **ServiceID** должен быть равен 0xFFFF. Монтажное устройство должно предоставлять совместимый с MPEG-2 транспортный поток на выходной мультиплекс, хотя мультиплекс вставки не должен содержать PSI.

Имеет значение порядок передачи сообщений монтажа. Первое сообщение, передаваемое в какой-либо заданной последовательности вставок "конец к концу", должно использовать время `time()`, в то время как все другие сообщения **Splice_Request** могут использовать **PriorSession**. Номер **PriorSession** должен ссылаться на существующий сеанс, который еще не завершился. Во всех других случаях возвращается ошибка 123, указывающая на **PriorSession** или поле `time()`.

Сервер выбирает идентификаторы PID элементарных потоков внутри мультиплекса вставки. Идентификаторы PID не могут быть общими для соседних сеансов одного и того же сервера в одном мультиплексе вставки. Причина этого в том, что иногда происходит небольшое наложение по времени потоков смежных сеансов из-за требований к данному API.

7.5.1 Сообщение Splice_Request

Поле data() этого сообщения содержит структуру Splice_Request_Data, описанную в таблице 7-6.

Таблица 7-6/J.280 – Splice_Request_Data

Синтаксис	Байты	Тип
Splice_Request_Data {		
SessionID	4	uimsbf
PriorSession	4	uimsbf
time()		
ServiceID	2	uimsbf
If (ServiceID = 0xFFFF)		
{		
PcrPID	2	uimsbf
PIDCount	4	uimsbf
for (j=0; j< PidCount ; j++)		
splice_elementary_stream()		
}		
Duration	4	uimsbf
SpliceEventID	4	uimsbf
PostBlack	4	uimsbf
AccessType	1	uimsbf
OverridePlaying	1	uimsbf
ReturnToPriorChannel	1	uimsbf
for (i=0; i< N ; i++)		
splice_API_descriptor()		
}		

SessionID – Идентификатор сеанса. Он используется для того, чтобы отличить этот запрос от других запросов, которые были переданы или формируются. Не допускаются несколько одновременных сообщений **Splice_Request** с одинаковым идентификатором **SessionID**. Если используется сообщение **ExtendedData_Request**, то для этого идентификатора **SessionID** должно быть получено сообщение **ExtendedData_Response** прежде, чем будет повторно использоваться данный идентификатор.

PriorSession – Это поле допускает использование упрощенного метода выполнения вставки "конец к концу". Значение этого поля будет содержать идентификатор **SessionID** сеанса, который непосредственно предшествует ему. Значение данного поля, равное 0xFFFFFFFF, указывает на то, что этот сеанс использует поле time() для инициализации его вставки, а не идентификатор **SessionID** предыдущего сеанса. Данное поле должно иметь действительный идентификатор **SessionID** только тогда, когда непосредственно предшествующий сеанс был инициирован тем же сервером. При создании вставок "конец к концу" от нескольких серверов должно использоваться поле time(), а не поле **PriorSession**.

time() – Время монтажа события. Обычно таким полем будет поле time() из сообщения **Cue_Request**, которое возвращается монтажному устройству. Если событие было запущено не сообщением **Cue_Request**, то это будет временем, когда сервер намеревается вызвать событие монтажа. Данное поле игнорируется, если поле **PriorSession** не равно 0xFFFFFFFF. Если это значение не связано с сообщением с меткой по J.181, то возможны различия во времени фактического монтажа для различных монтажных устройств, зависящие от их буфера и моделей монтажа. Описание синтаксиса структуры time() приведено в подразделе 8.4.

ServiceID – Номер программы канала в мультиплексе вставки, который будет монтироваться вместо первичного канала. Если он установлен равным 0xFFFF, то требуются `splice_elementary_stream()` и **PIDCount**.

PCR – Указывает PID PCR.

PIDCount – Число PID в канале вставки. (Не включая PCR PID.)

Duration – Число тактов 90 кГц, которые сервер просит монтажное устройство вставить. Это поле может отменить значение продолжительности по J.181. Оно может быть установлено на 0, чтобы указать, что монтажное устройство должно переключиться на канал вставки, пока не поступит новое сообщение **Splice_Request**.

SpliceEventID – Этот идентификатор используется, чтобы связать данное событие вставки с сообщением с меткой по J.181, которое, возможно, вызвало этот монтаж. Он должен быть эквивалентен идентификатору `splice_event_id` команды `splice_insert` соответствующего сообщения с меткой по J.181. Он должен быть одинаковым для всех сообщений **Splice_Request**, относящихся к одному и тому же сообщению с меткой по J.181. Для события, которое не было инициализировано сообщением с меткой по J.181, это поле будет установлено равным 0xFFFFFFFF.

PostBlack – Число тактов 90 кГц сигнала черного поля и заблокированного звука, которые должны воспроизводиться в конце воспроизведения содержания вставки. Интервал **PostBlack** следует за отрезком времени, указанным **Duration** (продолжительность), и не включен в него. Если параметр **PostBlack** не запрошен, то это поле будет установлено на 0. **PostBlack** не должно рассматриваться как часть воспроизводимой в текущий момент вставки для флага **OverridePlaying**.

AccessType – Указывает тип доступа для данного соединения. Это целое число от 0 до 9, причем 0 означает самый низкий приоритет, а 9 – самый высокий приоритет.

OverridePlaying – Если этот флаг равен 0, данное сообщение **Splice_Request** не может отменить воспроизводимую в текущий момент вставку. Если этот флаг установлен на 1, то данное сообщение **Splice_Request** отменит любую воспроизводимую в текущий момент вставку с равным или более низким приоритетом. Воспроизводимая в текущий момент вставка имеет место между входной монтажной точкой и выходной монтажной точкой.

ReturnToPriorChannel – Если этот флаг равен 0, монтажное устройство не должно возвращаться к первичному каналу или отмененному каналу вставки при завершении данного запроса монтажа **Splice_Request**. Ожидается, что новое сообщение **Splice_Request** будет выдано прежде, чем завершится данная вставка. Если новое сообщение **Splice_Request** не получено, то монтажное устройство должно прекратить передачу по этому выходному каналу. Если этот флаг равен 1, оно должно возвратиться к предшествующему каналу, если только не получено следующее сообщение **Splice_Request**, в котором указано иное.

`splice_API_descriptor()` – Дескриптор, который должен следовать синтаксису, определенному в подразделе 8.5. Дескрипторы `playback_descriptor()` и `muxpriority_descriptor()` являются подходящими дескрипторами для этого оператора.

7.5.2 Сообщение **Splice_Response**

Сообщение ответа **Splice_Response** не содержит никаких данных и указывает, что было получено сообщение **Splice_Request**. Это сообщение может содержать в соответствующих случаях код ошибки.

7.5.3 Сообщение **SpliceComplete_Response**

Сообщение **SpliceComplete_Response** передается, когда начинается и завершается вставка. Оно применяется также и для вставок "конец к концу". Например, если воспроизводятся две части содержания, возвращаются четыре сообщения **SpliceComplete_Response**, одно в начале первой части содержания, другое после завершения первой части содержания, третье в начале второй части содержания и еще одно после завершения второй части содержания. Код результата в заголовке должен надлежащим образом указывать на причину отказа, если монтаж завершился неудачно, чтобы сервер мог предпринять соответствующие действия. Входная и выходная монтажные точки – это отдельные события, и они должны обрабатываться как отдельные события. В случае отказа при монтаже двух частей содержания выходная монтажная точка должна указывать допустимое состояние, если текущая часть содержания воспроизведена полностью. Сообщение **SpliceComplete_Response** должно передаваться немедленно после отказа любого события монтажа и не должно ждать в течение ожидаемой продолжительности вставленного содержания.

Поле data() этого сообщения содержит структуру SpliceComplete_Response_Data, описание которой приведено в таблице 7-7.

Таблица 7-7/J.280 – SpliceComplete_Response_Data

Синтаксис	Байты	Тип
SpliceComplete_Response_Data {		
SessionID	4	uimsbf
SpliceTypeFlag	1	uimsbf
Bitrate	4	uimsbf
PlayedDuration	4	uimsbf
}		

SessionID – Идентификатор сеанса, используемый сообщением **Splice_Request**.

SpliceTypeFlag – Это поле должно быть равным 0, чтобы означать начальную монтажную точку (начало), и 1, чтобы означать конечную монтажную точку (конец).

Bitrate – Для конечной монтажной точки это средняя скорость передачи информации для сеанса. Это поле задает скорость передачи в битах в секунду (бит/с), включая заголовок транспортного пакета для этого канала.

PlayedDuration – Для конечной монтажной точки это число тактов 90 кГц, которые фактически воспроизводятся.

7.6 Сообщения о работе

Сразу по завершении инициализации сервер может передавать сообщения **Alive_Request**, чтобы гарантировать, что монтажное устройство все еще работает. Каждое сообщение **Alive_Response** содержит передаваемую серверу информацию о состоянии. Эта информация указывает состояние устройства. Если в течение предыдущих 60 секунд не было никаких передач по соединению TCP/IP, то должно быть отправлено сообщение **Alive_Request**.

7.6.1 Сообщение Alive_Request

Поле data() сообщения **Alive_Request** содержат структуру **Alive_Request_Data**, описание которой приведено в таблице 7-8.

Таблица 7-8/J.280 – Alive_Request_Data

Синтаксис	Байты	Тип
Alive_Request_Data {		
time()		
}		

time() – Текущее время UTC передающего устройства, выбираемое как можно ближе к передаче сообщения. Оно предназначено для использования монтажным устройством и сервером с целью проверки степени синхронизации времени в этих двух системах. Считается, что оно не позволит обеспечить синхронизацию систем для надежного монтажа, но разработчики могут использовать его для любых целей. Описание синтаксиса структуры time() приведено в подразделе 8.4.

7.6.2 Сообщение Alive_Response

Поле data() сообщение **Alive_Response** содержит структуру **Alive_Response_Data**, описание которой приведено в таблице 7-9.

Таблица 7-9/J.280 – Alive_Response_Data

Синтаксис	Байты	Тип
Alive_Response_Data { State SessionID time() }	4 4	uimsbf uimsbf

State – Это значение описывает состояние выходного канала.

Таблица 7-10/J.280 – Состояния сообщения Alive_Response

Состояние	Описание
0x00	Нет вывода
0x01	По первичному каналу
0x02	По каналу вставки

SessionID – Идентификатор сеанса воспроизводимой в текущее время вставки. Он действителен только для состояния **State** = 0x02.

time() – Текущее время UTC передающего устройства, выбираемое как можно ближе к передаче сообщения. Оно предназначено для использования монтажным устройством и сервером с целью проверки того, насколько хорошо синхронизировано время в этих двух системах. Считается, что оно не позволит обеспечить синхронизацию систем для надежного монтажа, но разработчики могут использовать его для любых целей. Описание синтаксиса структуры time() приведено в подразделе 8.4.

7.7 Расширенные сообщения с данными

Это определяемая монтажным устройством структура для передачи подробных данных о воспроизведении на сервер. После получения сообщения **SpliceComplete_Response** можно найти расширенные данные, используя **ExtendedData_Request**. Идентификатор **SessionID**, используемый в этом сообщении, совпадает с идентификатором **SessionID**, используемым при настройке этого сеанса и в сообщении **SpliceComplete_Response**.

7.7.1 Сообщение ExtendedData_Request

Поле data() этого сообщения содержит структуру **ExtendedData_Request_Data**, описание которой приведено в таблице 7-11.

Таблица 7-11/J.280 – ExtendedData_Request_Data

Синтаксис	Байты	Тип
ExtendedData_Request_Data { SessionID ExtendedDataType }	4 4	uimsbf uimsbf

SessionID – Идентификатор **SessionID** законченного сеанса.

ExtendedDataType – Запрошенный тип данных ответа монтажного устройства в сообщении **ExtendedData_Response**. Это значение может быть установлено равным 0xFFFFFFFF, чтобы указать, что должен быть возвращен тип данных по умолчанию. В данной Рекомендации диапазон от 0x00000000 до 0x7FFFFFFF зарезервирован для стандартизации в будущем. Диапазон от 0x80000000 до 0xFFFFFFFFE предназначен для использования только поставщиками.

7.7.2 Сообщение **ExtendedData_Response**

Сервер должен использовать поле **MessageSize** для определения объема данных, которые он обязан считать с помощью сообщения **ExtendedData_Response**.

Поле `data()` этого сообщения содержит структуру **ExtendedData_Response_Data**, описанную в таблице 7-12.

Таблица 7-12/J.280 – **ExtendedData_Response_Data**

Синтаксис	Байты	Тип
<pre>ExtendedData_Response_Data { SessionID for (i=0;i<n;i++) splice_API_descriptor() }</pre>	4	uimsbf

SessionID – Идентификатор сеанса **SessionID**, для которого действительны эти данные.

`splice_API_descriptor()` – Определяемый монтажным устройством дескриптор формата, описываемый в подразделе 8.5.

7.8 Сообщения аварийного завершения

Сервер может в любой момент передать сообщение **Abort_Request**, которое заставит монтажное устройство немедленно возвратиться в отмененный канал вставки или в первичный канал. Монтажное устройство должно передать сообщение **Abort_Response**, чтобы подтвердить получение сообщения **Abort_Request**. Сообщение **SpliceComplete_Response** с кодом результата 116 (Аварийное завершение вставки) передается, если сообщение **Abort_Request** вызвало формирование выходной монтажной точки вставки. Если выходная монтажная точка не потребовалась, то сообщение **SpliceComplete_Response** не должно передаваться.

Все ждущие обработки вставки "конец к концу", связанные через поле **PriorSession** сообщения **Splice_Request** с идентификатором сеанса **SessionID** сообщения **Abort_Request**, также должны быть аварийно завершены. Для каждого аварийного завершения сеанса с **SessionID** должно быть возвращено сообщение об ошибках. Рассмотрим следующий пример: три вставки получают метки для последовательного выполнения в рамках блока времени, причем первое событие основано на времени; второе событие связано с первым идентификатором сеанса **SessionID** с помощью поля **PriorSession**; третье событие связано со вторым **SessionID** с помощью этого поля **PriorSession**. Если в этом примере первое событие вставки завершено аварийно, то два последующих события вставки с метками также будут завершены подобным образом. Сообщение аварийного завершения не прерывает вставки, которые используют другое соединение API между сервером и монтажным устройством. Для следующего монтажа в первичном канале требуется, чтобы в сообщении "монтаж" поле **PriorSession** было равно 0xFFFFFFFF.

7.9 Сообщение Abort_Request

Поле data() этого сообщения содержит структуру Abort_Request_Data, описанную в таблице 7-13.

Таблица 7-13/J.280 – Abort_Request_Data

Синтаксис	Байты	Тип
Abort_Request_Data { SessionID }	4	uimsbf

SessionID – Идентификатор сеанса **SessionID** и все последующие сеансы, связанные через поле **PriorSession**, которые должны быть аварийно завершены.

7.10 Сообщение Abort_Response

Сообщение **Abort_Response** не содержит никаких данных и указывает, что было получено сообщение **Abort_Request**. Это сообщение может содержать в соответствующих случаях код результата.

7.11 Запрос параметров настройки конфигурации

Текущие параметры настройки конфигурации для соединения API могут быть возвращены. Это относится к части информации в сообщении **Init_Request**. Сообщение **GetConfig_Request** не содержит никаких дополнительных данных.

7.11.1 Сообщение GetConfig_Request

Сообщение **GetConfig_Request** не содержит никаких данных.

7.11.2 Сообщение GetConfig_Response

Поле data() этого сообщения содержит структуру GetConfig_Response_Data, описанную в таблице 7-14.

Таблица 7-14/J.280 – GetConfig_Response_Data

Синтаксис	Байты	Тип
GetConfig_Response_Data { ChannelName Hardware_Config() TS_program_map_section() }	32	Строка

ChannelName – Логическое имя, присвоенное выходному каналу этого соединения.

Hardware_Config() – Описание синтаксиса структуры **Hardware_Config()** приведено в подразделе 8.2.

TS_program_map_section() – Это вся секция PMT для выходного канала, как определено в Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1. Если монтажное устройство изменяет PMT, ему следует сообщить об этом изменении на сервер в сообщении **General_Response** с кодом результата 128.

7.12 Сообщение General_Response

Это сообщение используется для передачи асинхронной информации между сервером и монтажным устройством. С этим сообщением не связано какое-либо поле data(). В нем можно передать любой код результата. Обычно такое сообщение используется, чтобы уведомить об изменении PMT выходного канала или недействительном сообщении запроса.

8 Дополнительные структуры

8.1 Версия

Структура Version (Версия) используется, чтобы поддержать надлежащий механизм отслеживания версий API. Ожидается, что данный API будет изменяться со временем, и для учета этого расширения в сообщениях **Init_Request** и **Init_Response** указывается версия, чтобы гарантировать, что монтажное устройство поддерживает ту же версию, что и сервер.

Таблица 8-1/J.280 – Version()

Синтаксис	Байты	Тип
Version { Revision_Num }	2	uimsbf

Revision_Num – Это поле равно нулю в этой версии.

8.2 Структура Hardware_Config

Данная структура описывает аппаратный интерфейс между сервером и монтажным устройством. Монтажному устройству важно точно знать, где подключен сервер, с тем чтобы знать, на какой мультиплекс ссылается сервер. Примером такой связи может служить соединение сервера с монтажным устройством по DVB-ASI.

Таблица 8-2/J.280 – Hardware_Config()

Синтаксис	Байты	Тип
Hardware_Config { Length Chassis Card Port Logical_Multiplex_Type Logical_Multiplex }	2 2 2 2 2 2	uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf

Length – Указывает длину данной структуры, следующей за этим полем, в байтах.

Chassis – Целое число, показывающее, какой блок монтажного устройства подключен к мультиплексу вставки сервера. В случаях, когда платы обозначены буквами алфавита, производится преобразование в целочисленное значение (т. е. А – 1; В – 2; и т. д.).

Card – Целое число, указывающее плату монтажного устройства, к которой подключен мультиплекс вставки сервера. В случаях, когда платы обозначены буквами алфавита, производится преобразование в целочисленное значение (т. е. А – 1; В – 2; и т. д.).

Port – Номер аппаратного порта, к которому подключен мультиплекс вставки сервера.

Logical_Multiplex_Type – Значение из таблицы 8-3.

Таблица 8-3/J.280 – Тип логического мультиплекса

Тип	Длина	Имя	Описание
0x0000	0	Не используемый	Поле Logical_Multiplex не является необходимым для идентификации мультиплекса.
0x0001	Переменная	Определяемый пользователем	В данной Рекомендации не определено использование поля Logical_Multiplex , оно должно быть согласовано монтажным устройством и сервером.
0x0002	6	MAC-адрес	Поле Logical_Multiplex содержит адрес управления доступом к среде (MAC) IEEE мультиплекса в виде 6-байтового адреса.
0x0003	6	Адрес IPv4	Старшие 4 байта поля Logical_Multiplex содержат адрес межсетевых протоколов (IP) мультиплекса, а остальные 2 байта содержат номер IP-порта мультиплекса.
0x0004	18	Адрес IPv6	Старшие 16 байтов поля Logical_Multiplex содержат адрес межсетевых протоколов (IPv6) мультиплекса, а остальные 2 байта содержат номер IP-порта мультиплекса.
0x0005	5	ATM-адрес	Поле Logical_Multiplex содержит координаты канала ATM (режима асинхронной передачи), по которому передается мультиплекс. Два старших байта поля Logical_Multiplex содержат идентификатор виртуального тракта (VPI). Следующие два байта содержат идентификатор виртуального канала (VCI). Самый младший байт содержит номер уровня адаптации ATM (AAL).
0x0006-0xFFFF	Переменная	Зарезервирован	Зарезервировано для стандартизации в будущем.

Logical_Multiplex – Если через порт (**Port**) передается несколько мультиплексов вставки по одному входу, то это поле позволяет монтажному устройству определить, какой мультиплекс используется при монтаже с данного сервера. Значение и формат данного поля определяются полем **Logical_Multiplex_Type**. Если требуется нестандартное определение для **Logical_Multiplex**, то **Logical_Multiplex_Type** должен быть установлен равным 1, что означает определяемый пользователем (User Defined) тип.

8.3 Структура `splice_elementary_stream()`

Идентификаторы пакетов (PID) – это идентификаторы частей транспортного потока, видео, аудио, данных и т. д. Эта структура используется для описания одного из элементов в программе в MPTS. Сообщение **Splice_Request** может содержать структуру `splice_elementary_stream()` для каждого компонента транспортного потока (за исключением PID PCR). **StreamTypes** (типы потоков) основаны на определениях таблицы PMT MPEG.

В данной Рекомендации не определяется порядок преобразования нескольких PID аудио/видео/данных в выходные PID. В ней также не определяется поведение монтажного устройства при наличии или отсутствии нескольких звуковых дорожек в канале вставки по сравнению с первичным каналом.

Таблица 8-4/J.280 – Структура splice_elementary_stream()

Синтаксис	Байты	Тип
splice_elementary_stream {		
Length	1	uimsbf
PID	2	uimsbf
StreamType	2	uimsbf
AvgBitrate	4	uimsbf
MaxBitrate	4	uimsbf
MinBitrate	4	uimsbf
Hresolution	2	uimsbf
Vresolution	2	uimsbf
for (i=0;i<N;i++)		
descriptor()		
}		

Требуется PID PCR.

Length – Полная длина структуры splice_elementary_stream().

PID – Используемый номер PID. Это двухбайтовое (16-битовое) поле, оно должно содержать 13-битовый PID, выровненный вправо в виде 16-разрядного целого числа (от 0x0000 до 0x1FFF).

StreamType – Тип PID (аудио, видео и т. д.) Этот номер соответствует спецификации PMT, определенной в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

AvgBitrate – Средняя скорость передачи информации для данного PID по всему содержанию (биты в секунду, бит/с), на которой было закодировано содержание. Она устанавливается равной 0xFFFFFFFF, если закодированная скорость передачи информации неизвестна.

MaxBitrate – Максимальная скорость передачи информации для данного PID. Она устанавливается равной 0xFFFFFFFF, если скорость передачи информации неизвестна.

MinBitrate – Минимальная скорость передачи информации для данного PID. Она устанавливается равной 0xFFFFFFFF, если скорость передачи информации неизвестна.

HResolution – Ширина видеоизображения в пикселах при использовании этого PID. Если PID не содержит видеоизображений или если сервер не может предоставить это значение, оно должно быть установлено равным 0xFFFF.

VResolution – Высота видеоизображений в пикселах при использовании этого PID. Если PID не содержит видеоизображений или если сервер не может предоставить это значение, оно должно быть установлено равным 0xFFFF.

descriptor() – Это может быть любой действительный дескриптор, используемый в PMT. Для нескольких PID аудио требуются дескрипторы языка, определенные в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

8.4 Определение поля time()

Структура time используется для определения различного времени монтажа в данной Рекомендации.

Таблица 8-5/J.280 – Структура time()

Синтаксис	Байты	Тип
time {		
Seconds	4	uimsbf
MicroSeconds	4	uimsbf
}		

Seconds – Секунды UTC, истекшие с 12:00 01 января 1970 г.

MicroSeconds – Смещение относительно поля **Seconds** в микросекундах.

8.5 Определение поля splice_API_descriptor()

Это шаблон для добавления дескрипторов в любое сообщение, определенное в данной Рекомендации. Дескрипторы могут использоваться в сообщениях **Splice_Request**, **ExtendedData_Response** и **Init_Request**. Использование дескрипторов в сообщениях, определенных в данной Рекомендации, является необязательным. В таблице 8-6 приведен общий формат дескрипторов, используемых в данной Рекомендации.

Таблица 8-6/J.280 – Структура splice_API_descriptor()

Синтаксис	Байты	Тип
splice_API_descriptor {		
Splice_Descriptor_Tag	1	uimsbf
Descriptor_Length	1	uimsbf
Splice_API_Identifier	4	uimsbf
for (i=0;i<n;i++)		
Private_Byte	1	uimsbf
}		

Splice_Descriptor_Tag – Значение от 0x00 до 0xFF обозначает определенный используемый дескриптор. Значения тега от 0x00 до 0xFF зарезервированы для использования данной Рекомендацией. Производитель может использовать уникальный для производителя идентификатор **Splice_API_Identifier**, позволяющий расширить диапазон тегов и обеспечить более устойчивый метод добавления дескрипторов, уникальных для производителя.

Descriptor_Length – Указывает длину дескриптора, следующего за этим полем, в байтах. Длина дескриптора ограничена 256 байтами, поэтому это значение не должно превышать 254.

Splice_API_Identifier – Идентификатор организации, которая определила данный дескриптор. Для всех дескрипторов в данной Рекомендации этот идентификатор равен 0x53415049 (ASCII "SAPI"). Это значение было выбрано, чтобы не создавать конфликтов с дескрипторами любого другого известного идентификатора.

Private_Byte – Остающаяся часть дескриптора выделена для полей данных, требуемых определяемым дескриптором.

8.5.1 Определения поля `playback_descriptor()`

Структура `playback_descriptor()` – это реализация `splice_API_descriptor()`, предназначенная для использования в сообщении `Splice_Request`.

Критерии аварийного завершения исследуют скорость воспроизведения, определяемую как скорость передачи информации в битах в выходном канале, усредненную за односекундный период. Рекомендуемая ширина скользящего смещение окна усреднения составляет одну секунду или менее.

Таблица 8-7/J.280 – Структура `playback_descriptor()`

Синтаксис	Байты	Тип
<code>playback_descriptor {</code>		
Splice_Descriptor_Tag	1	uimsbf
Descriptor_Length	1	uimsbf
Splice_API_Identifier	4	uimsbf
BitrateRule	1	uimsbf
MinPlaybackRate	4	uimsbf
<code>}</code>		

Splice_Descriptor_Tag – 0x01.

Descriptor_Length – 0x09.

Splice_API_Identifier – 0x53415049, ASCII "SAPI".

BitrateRule – Флаг, используемы для указания правил для **MinPlaybackRate**.

Таблица 8-8/J.280 – Значения **BitrateRule**

BitrateRule	Описание
0x00	Игнорировать MinPlaybackRate
0x01	Возвратить код результата 127 немедленно, используя сообщение General_Response , если скорость воспроизведения падает ниже MinPlaybackRate , но соединение не разрывается
0x02	Аварийное прерывание, если скорость воспроизведение падает ниже MinPlaybackRate
0x03	Отменить сеанс до входной монтажной точки, если монтажное устройство решает, что скорость передачи MinPlaybackRate не будет обеспечена. Монтажное устройство передаст сообщение SpliceComplete_Response или General_Response с кодом результата 127.

MinPlaybackRate – Минимальная суммарная скорость передачи в битах в выходном канале, усредненная за одну секунду во время монтажа, с которой может осуществляться воспроизведение, до того как будет запущено правило **BitrateRule**. Установка этого значения на 0 указывает, что минимальная скорость отсутствует.

8.5.2 Определения поля muxpriority_descriptor()

Структура muxpriority_descriptor() – это реализация splice_API_descriptor(), которая предназначена для использования в сообщении Splice_Request.

Таблица 8-9/J.280 – Структура muxpriority_descriptor()

Синтаксис	Байты	Тип
muxpriority_descriptor {		
Splice_Descriptor_Tag	1	uimsbf
Descriptor_Length	1	uimsbf
Splice_API_Identifier	4	uimsbf
MuxPriorityValue	1	uimsbf
}		

Splice_Descriptor_Tag – 0x02

Descriptor_Length – 0x05

Splice_API_Identifier – 0x53415049, ASCII "SAPI".

MuxPriorityValue – Это число может варьироваться от 1 до 10 (1 означает самый низкий приоритет, 5 – среднее значение, а 10 – наиболее высокий приоритет). Это число изменяет хранимый приоритет **MuxPriorityValue** первичного канала в монтажном устройстве. Если **MuxPriorityValue** равно 5, то приоритет выходного канала не будет изменяться. Если **MuxPriorityValue** менее 5, то оно вычитается из приоритета выходного канала, а если **MuxPriorityValue** больше 5, то оно добавляется к приоритету выходного канала.

Использование **MuxPriorityValue** не гарантирует, что содержание воспроизводится с любым определенным уровнем качества. Фактический результат воздействия **MuxPriorityValue** зависит от общей конфигурации мультимедиа монтажа и от того, насколько устройство монтажа должно снизить полную скорость передачи мультимедиа в любое заданное время. Он будет также зависеть от того, как работает монтажное устройство, поэтому это поле будет очень сильно зависеть от производителя монтажного устройства.

8.5.3 Определения поля missing_Primary_Channel_action_descriptor()

Структура missing_Primary_Channel_action_descriptor() – это реализация splice_API_descriptor(), которая предназначена для использования в сообщении Init_Request.

Если первичный канал завершен по какой-либо причине в течение вставки, результат в декодере может состоять в воспроизведении стоп-кадра последнего вставленного кадра при завершении вставки. Этот дескриптор позволяет дать монтажному устройству команду осуществить вставку черных полей без звука, чтобы очистить буфер декодера при отсутствии первичного канала, когда он обычно становится выходным источником аудио/видео.

Таблица 8-10/J.280 – Структура missing_Primary_Channel_action_descriptor ()

Синтаксис	Байты	Тип
missing_Primary_Channel_action_descriptor {		
Splice_Descriptor_Tag	1	uimsbf
Descriptor_Length	1	uimsbf
Splice_Api_Identifier	4	uimsbf
MissingPrimaryChannelAction	1	uimsbf
}		

Splice_Descriptor_Tag – 0x03

Descriptor_Length – 0x05

Splice_API_Identifier – 0x53415049, ASCII "SAPI".

MissingPrimaryChannelAction – Этот параметр имеет три возможных значения – 0, 1 и 2. Значение 0 означает "ничего не делать". 1 означает вставку одного черного I-кадра и одного кадра без звука. 2 означает продолжение передачи черных полей без звука до восстановления первичного сигнала.

9 Синхронизация времени

Синхронизация времени требуется из-за передачи времени между сервером и монтажным устройством. Задержка сообщения TCP/IP несколько непредсказуема, на нее влияют другие машины в сети. При синхронизации машин можно передавать время между этими двумя машинами, не беспокоясь об обычных задержках в сети и обеспечивая очень точный монтаж. Один из возможных необязательных методов состоит в использовании сетевого протокола системного времени (NTP) для поддержания синхронизации сервера и монтажного устройства. Вероятно, что серверы уже синхронизованы в течение некоторого времени, и поэтому они должны обеспечить службу NTP, причем монтажное устройство будет клиентом NTP. Можно также использовать общую в сети хостовую систему NTP-сервера, поскольку она обычно имеется на кабельной головной станции с сетевой инфраструктурой.

Система синхронизации времени должна быть способна сохранять относительную ошибку синхронизации монтажного устройства и сервера в пределах ± 15 мс. Такая точность синхронизации (то есть в пределах одного видеокadra) считается достаточной для надлежащего взаимодействия монтажного устройства и сервера в данной Рекомендации. Система может использовать сообщения **Alive_Request/Alive_Response** для проверки требуемой синхронизации двух устройств и предупреждения оператора в случае потери синхронизации.

Битовый поток, представляющий первичный канал, подвержен различным задержкам, которые могут включать монтаж восходящего потока данных, спутниковые каналы связи и другие процессы передачи и подготовки. В сумме эти задержки могут составлять от миллисекунд до секунд, и привязка синхронизации потока для компенсации таких задержек может быть получена из значений PCR, передаваемых в транспортных потоках MPEG-2. Однако эти задержки не влияют на точность сообщения с метками, встроенного в первичный канал. Сообщение с метками использует PCR для указания правильного времени вставки, поэтому оно сохраняет ее первоначальную точность по отношению к содержанию.

Сервер, обеспечивающий содержание канала вставки, знает только время (UTC), и окна вставки, с которыми он был запрограммирован, связаны со временем. Однако монтажное устройство должно сообщить ему фактический точный момент, когда он должен начать воспроизводить это содержание.

Когда монтажное устройство получает битовый поток программы, все задержки этого потока уже произошли. Монтажное устройство может получить PCR и связать его со временем, затем передать сообщение серверу, который определяет точное время UTC, когда должна начаться передача содержания. Теперь канал вставки от сервера поступает в монтажное устройство при точной синхронизации с первичным каналом, и можно произвести точный монтаж. Любые дополнительные задержки, которые происходят внутри монтажного устройства, не имеют значения, поскольку входные битовые потоки были синхронизированы.

10 Синхронизация систем

10.1 Поток сигналов монтажа DPI

На рисунках 4 и 5 показаны конкретные детали, касающиеся использования и упорядочения различных сообщений, допускаемых рассматриваемым API. Фактическое применение сообщений API не может быть ограничено этими примерами.

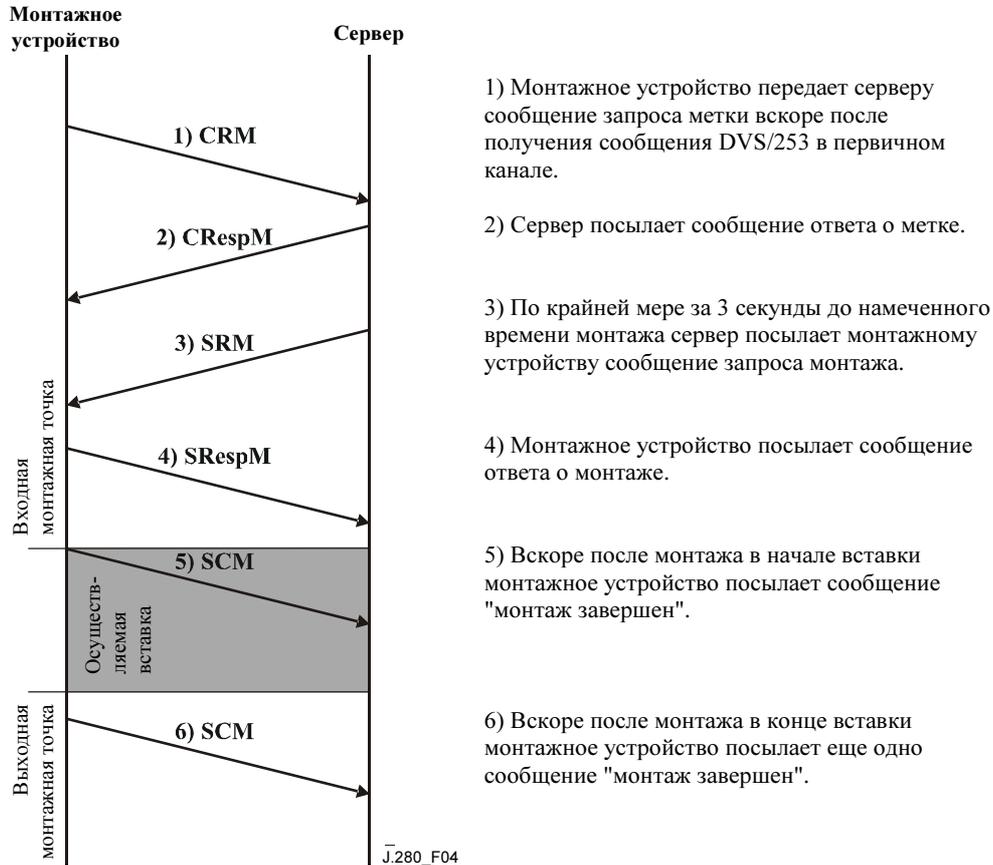


Рисунок 4/J.280 – Монтаж одного события

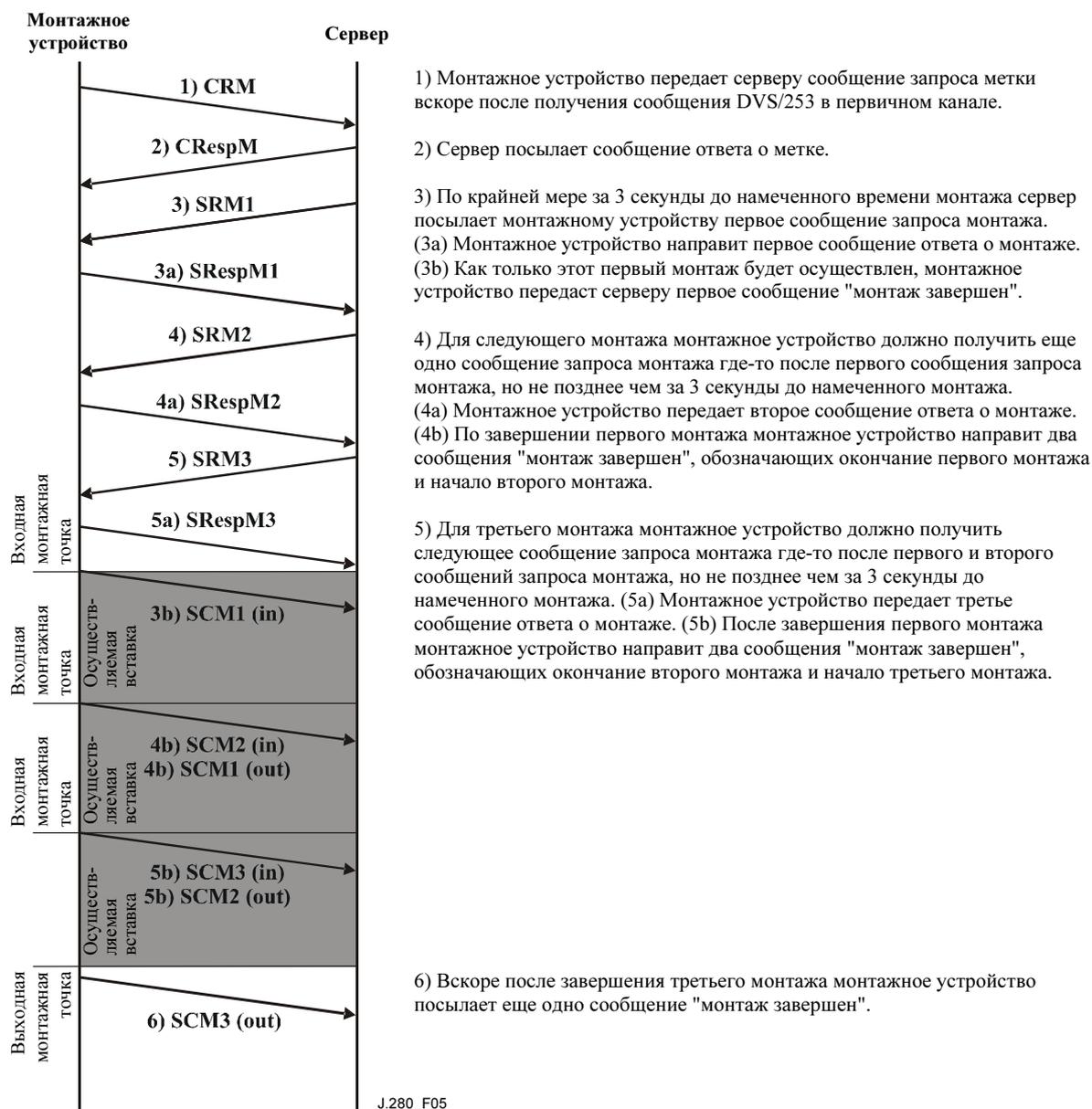


Рисунок 5/J.280 – Монтаж нескольких событий

10.2 Временная последовательность инициализации монтажа DPI

На рисунке 6 приведен пример синхронизации событий, приводящих к началу вставки программы (или рекламы). В реальной ситуации время может отличаться от согласования по времени, показанного на этом рисунке. Показанный временной интервал применим для разрешения конфликтов на основе приоритетов, что обсуждалось в подразделе 6.2. Кроме того, показана совместная работа с сообщениями с метками согласно Рекомендации J.181.

На этом рисунке жирные черные линии указывают поток информации MPEG в первичном канале и в канале вставки. Тонкие черные линии указывают, что информация MPEG либо не передается в этот момент, либо является неважной (т. е. она не выбрана для вывода в выходной канал).

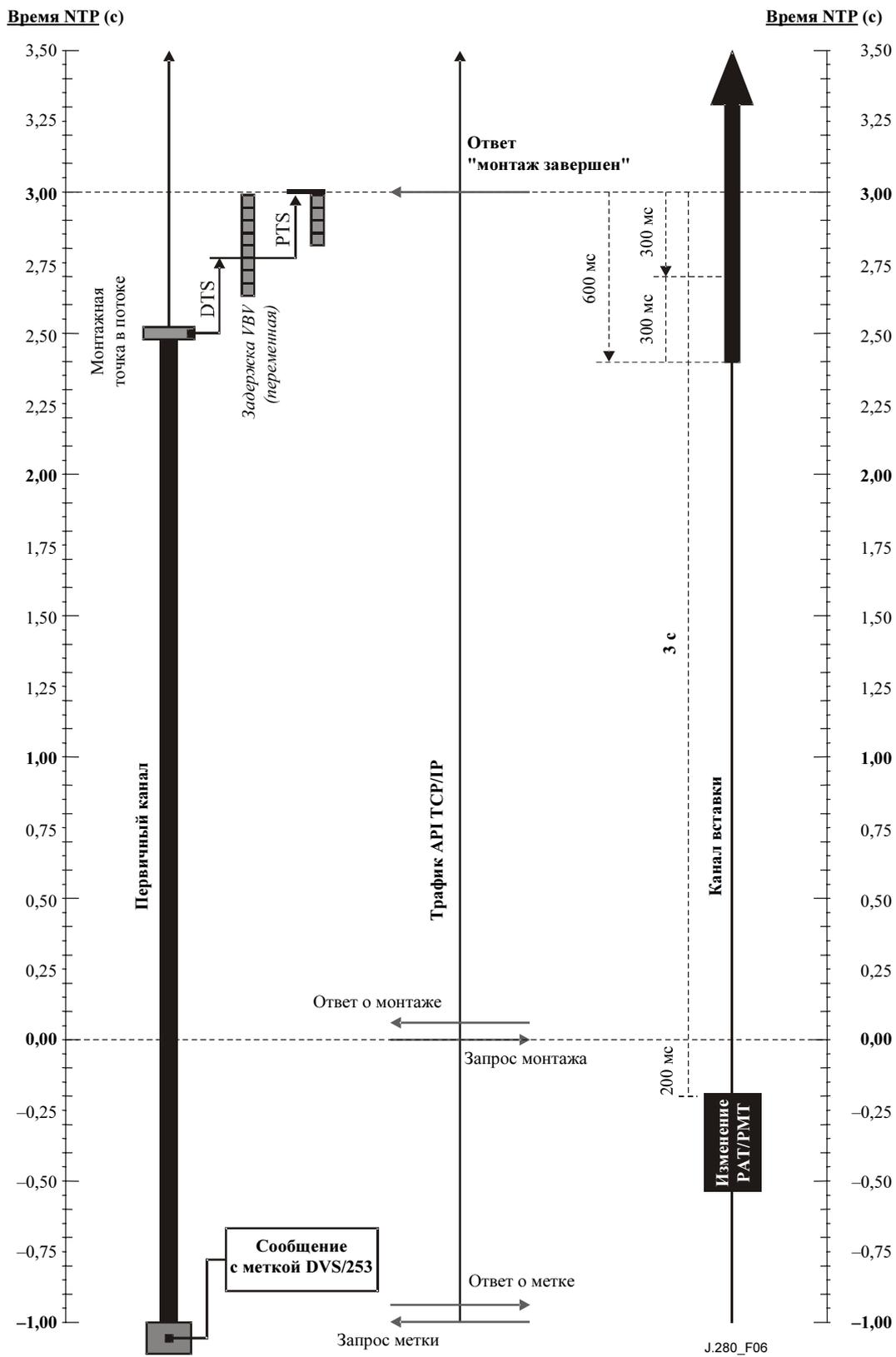


Рисунок 6/J.280 – Временная последовательность инициализации монтажа DPI

Приложение I

Коды результатов

Результат	Имя результата	Описание	Сообщение ответа
100	Успешный ответ		Все
101	Неизвестный отказ		Все
102	Неправильная версия	Сервер и монтажное устройство используют различные версии данного API.	Init_Response
103	Отказ доступа	Возможна проблема с лицензией.	Init_Response
104	Неправильное/Неизвестное имя канала ChannelName	Возможна ошибка конфигурации.	Init_Response
105	Неправильное физическое соединение	Возможна ошибка конфигурации.	Init_Response
106	Конфигурация не обнаружена	Монтажное устройство не может определить конфигурацию этого соединения.	GetConfig_Response
107	Неправильная конфигурация	Один или несколько параметров в конфигурации этого соединения недействительны.	GetConfig_Response
108	Отказ монтажа – неизвестный отказ		SpliceComplete_Response
109	Конфликт монтажа	Монтажу уже присвоен более высокий или тот же приоритет.	Splice_Response SpliceComplete_Response
110	Канал вставки не найден	Эта ошибка должна возвращаться, если в начале монтажа отсутствует канал вставки.	SpliceComplete_Response
111	Первичный канал не найден	Эта ошибка должна возвращаться, если первичный канал отсутствует во входной или выходной монтажной точке.	SpliceComplete_Response
112	Сообщение Splice_Request получено слишком поздно	Сообщение Splice_Request не было получено вовремя (за 3 секунды), чтобы монтажное устройство могло инициализировать монтаж.	Splice_Response
113	Монтажная точка не найдена	Монтажное устройство не смогло найти правильную монтажную точку в первичном канале.	SpliceComplete_Response
114	Очередь монтажа полна	Слишком много невыполненных сообщений Splice_Request .	Splice_Response
115	Подозрительное воспроизведение сеанса	Монтажное устройство обнаружило несоответствия видео или аудио, которые, возможно, оказали влияние на воспроизведение.	SpliceComplete_Response
116	Аварийное завершение вставки	Сообщение Abort_Request вызвало формирование выходной монтажной точки.	SpliceComplete_Response
117	Недействительное сообщение с меткой	Монтажное устройство или сервер не смогли произвести анализ сообщения с метками.	General_Response Cue_Response
118	Монтажное устройство не существует	SplicerName не найдено.	Init_Response

Результат	Имя результата	Описание	Сообщение ответа
119	Отказ Init_Request	Монтажное устройство не разрешает серверу подключиться.	Init_Response
120	Неизвестный идентификатор MessageID	Используйте сообщение Splicing_API_Message , чтобы послать ответ запрашивающей стороне. Возвратите неизвестный идентификатор MessageID .	Все
121	Недействительный идентификатор SessionID	Монтажное устройство не знает указанный идентификатор сеанса SessionID .	Splice_Response Abort_Response ExtendedData_Response
122	Сеанс не был полным	Монтажное устройство не смогло воспроизвести всю продолжительность. Сюда же относится случай, в котором сервер не смог предоставить достаточное содержание.	SpliceComplete_Response
123	Неправильное поле data() в сообщении запроса	Монтажное устройство или сервер не смогли успешно проанализировать поле в сообщении запроса. Положение неправильного поля возвращено в поле Result_Extension сообщения Splicing_API_Message .	Все
124	Дескриптор не реализован	В настоящее время монтажное устройство не понимает требуемый дескриптор или он не реализован.	Ответы на все сообщения, которые допускают использование дескрипторов.
125	Отмена канала	Этот код результата используется для указания воспроизводимой в настоящее время вставки, что она была отменена с состоянием выходной монтажной точки или вновь введена с состоянием входной монтажной точки (Splice_in).	SpliceComplete_Response
126	Преждевременный запуск канала вставки	Эта ошибка может быть выдана, если канал вставки начался преждевременно, и монтажное устройство не смогло определить правильное начало потока вставки.	SpliceComplete_Response
127	Скорость воспроизведения ниже пороговой скорости	Детали приведены в описании структуры playback_descriptor() .	SpliceComplete_Response
128	Изменение PMT	Он используется для указания серверу об изменении PMT для этого первичного канала.	General_Response
129	Неправильный размер сообщения	Сообщение не имело правильную длину, определенную в данной Рекомендации.	Все
130	Неправильный синтаксис сообщения	Поля, определенные данной Рекомендацией, не находятся в пределах допустимого диапазона.	Все
ПРИМЕЧАНИЕ. – Все коды результатов можно использовать в сообщении General_Response .			

ЛИТЕРАТУРА

Список библиографических документов

- KAR (M.), NARASIMHAN (S.), PRODAN (R.): Local Commercial Insertion in the Digital Headend, *Proceedings of NCTA 2000 Conference*, New Orleans, USA.
- Cable Television Laboratories: Cable Advertising, white paper, Louisville, CO, March 1997.

Приобретение литературы

- The National Cable Television Association, 1724 Massachusetts Ave., NW, Washington, D.C. 20036-1969; телефон: 202-775-3669; URL: <http://www.ncta.com>
- CableLabs, 400 Centennial Parkway, Louisville, CO 80027; телефон: 303-661-9100; факс: 303-661-9199; URL: <http://www.cablelabs.com>

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 (IP) и сети следующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи