ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 55714— 2013

Телевидение вещательное цифровое

ОБОРУДОВАНИЕ СИГНАЛИЗАЦИИ МЕТОК О ВСТАВКЕ (СПЛАЙСИНГЕ) РЕГИОНАЛЬНЫХ ПРОГРАММ В ТРАНСПОРТНЫЙ ПОТОК MPEG-2 ВЕЩАТЕЛЬНОГО ТЕЛЕВИДЕНИЯ

Основные параметры

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Автономной некоммерческой организацией «Научно-технический центр информатики» (АНО «НТЦИ»)

2 ВНЕСЕН Управлением технического регулирования и стандартизации Федерального агентства по техническому регулированию и метрологии

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 8 ноября 2013 г. № 1368-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений Американского национального стандарта/Общества инженеров кабельных телекоммуникаций, подкомитет Цифрового Видео «Формирование меток вставки цифровых программ» (ANSI/SCTE 35 2007 «Digital Program Insertion Cueing Message for Cable», NEQ)

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru).

© Стандартинформ, 2014

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины, определения, сокращения и аббревиатуры 1
4 Введение в систему формирования меток 5
4.1 Общие замечания 5
4.2 Точки вставки (сплайсинга) 5
4.3 Точки вставки (сплайсинга) программ 6
4.4 События вставки 6
4.5 Условия обработки контента, находящегося на хранении 7
4.6 Выбор PID для решения задач вставки 7
4.7 Поток сообщений при вставке 8
5 Дескрипторы PMT 8
5.1 Дескриптор регистрации 8
5.2 Дескриптор идентификатора метки 9
5.3 Дескриптор идентификатора потока 10
6 Таблица информации о вставке 11
6.1 Краткий обзор 11
6.2 Секция информации вставки 12
6.3 Команды вставки 15
6.4 Параметры структур времени вставки события и продолжительности события вставки 19
6.5 Ограничения 20
7 Дескрипторы вставки 22
7.1 Краткий обзор 22
7.2 Дескриптор вставки 22
7.3 Специальные дескрипторы вставки 23
8 Шифрование секций 29
8.1 Краткий обзор 29
8.2 Шифрование фиксированным ключом 30
8.3 Алгоритмы шифрования 30
Библиография 32

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Телевидение вещательное цифровое

ОБОРУДОВАНИЕ СИГНАЛИЗАЦИИ МЕТОК О ВСТАВКЕ (СПЛАЙСИНГЕ) РЕГИОНАЛЬНЫХ ПРОГРАММ В ТРАНСПОРТНЫЙ ПОТОК MPEG-2 ВЕЩАТЕЛЬНОГО ТЕЛЕВИДЕНИЯ

Основные параметры

Digital video broadcasting (DVB). Signaling equipment of splicing of regional programs in the transport stream of MPEG-2 broadcast television. Basic parameters

Дата введения — 2014—09—01

1 Область применения

Настоящий стандарт распространяется на оборудование цифровой вставки (сплайсинга) региональных программ и рекламных вставок в транспортные потоки MPEG-2 вещательного телевидения. Настоящий стандарт устанавливает основные параметры оборудования формирования сообщений меток цифровых вставок региональных программ и рекламных вставок в транспортные потоки MPEG-2, размещаемого на головных станциях (центрах) формирования программ вещания.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт:

ГОСТ 28147—89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по соответствующим ежемесячно издаваемым информационным указателям, опубликованным в текущем году. Если ссылочный стандарт заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться заменяющим (измененным) стандартом. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения, сокращения и аббревиатуры

3.1 В настоящем стандарте применены термины с соответствующими определениями:

3.1.1 **аналоговая тональная метка** (analog cue tone): В аналоговой системе сигнал, который представлен или последовательностью тонов DTMF или замыканием контакта. Для оборудования вставки рекламы это означает, что начинается или заканчивается время, доступное для рекламы (ВДР).

3.1.2 **вещатель** (broadcaster): Организация, которая собирает последовательность событий или программ для доставки.

3.1.3 **внепотоковое устройство:** Устройство, которое получает сообщение метки от другого устройства, работающего в транспортном потоке. Внепотоковое устройство непосредственно не получает и не передает транспортный поток.

3.1.4 **временная метка декодирования** (Decoding Time Stamp; DTS): Число, указывающее момент времени, когда данный видеокадр должен быть декодирован.

Издание официальное

3.1.5 **время представления (воспроизведения)** (Presentation Time): Время, когда модуль представления (воспроизведения) представляется (воспроизводится) в системном целевом декодере согласно стандарту ISO/IEC [1].

3.1.6 **время, доступное для рекламы;** ВДР (Avail): Время вещания, доступное для размещения рекламы (рекламных пауз и других промежутков эфирного времени, которые могут быть куплены под рекламу).

3.1.7 **вставка (сплайсинг, сращивание)** (Splice (Splicing)): 1. Процесс замещения основного канала каналом ввода, который характеризуется точкой входа в вставку и точкой выхода из вставки. 2. Сращивание рекламных пауз с основной программой вещания.

3.1.8 **глава** (chapter): Короткий раздел более длинной программы, обычно располагаемой так, чтобы разрешить средству просмотра легко определять местоположение сцены или раздела программы.

3.1.9 **идентификатор типа пакета** (Packet Identifier; PID): Тринадцатибитовый указатель в заголовке транспортного пакета MPEG-2, определяющий принадлежность пакета тому или иному потоку данных (стандарт ISO/IEC [1]).

3.1.10 **дескриптор регистрации** (Registration Descriptor): дескриптор регистрации переносится в PMT программы и указывает, что при сигнализации события вставки секции splice_info_sections нужно переносить в потоке PID этой программы. Присутствие дескриптора регистрации указывает на соответствие программы настоящему стандарту.

3.1.11 **контент** (content): Содержание, мультимедийный продукт канала ввода (например, рекламное объявление, анонс услуг общего пользования, замещающая телевизионная программа или материал программы, созданный путем соединения частей программы с сервера).

3.1.12 **метка времени в потоке PES** (Elementary Stream Clock Reference; ESCR): Метка, по которой декодеры потоков PES могут установить синхронизацию.

3.1.13 **метка времени представления (воспроизведения)** (Presentation Time Stamp; PTS): Число, указывающее время, когда данный видеокадр должен появиться на выходе декодера (ISO/IEC [1]).

3.1.14 **многопрограммный транспортный поток** (Multi Pogram Transport Stream; MPTS): Транспортный поток, содержащий множество программ.

3.1.15 **модуль доступа** (access unit): Кодированное представление модуля представления (воспроизведения). В случае аудио модуль доступа включает кодированное представление аудиокадра. В случае видео модуль доступа включает все кодированные данные изображения и биты стаффинга, которые следуют за ним, до начала следующего модуля доступа в соответствии со стандартом ISO/IEC [1] (2.1.1).

3.1.16 **модуль представления (воспроизведения)** (Presentation Unit): Декодируемый (декодированный) модуль доступа аудио или декодируемый (декодированный) модуль доступа видео согласно стандарту ISO/IEC [1] (2.1.40).

3.1.17 **пакетированный элементарный поток;** ПЭП (Packetized Elementary Stream; PES): Пакетированный элементарный поток, в котором данные разбиты на пакеты и снабжены заголовками.

3.1.18 **пауза** (off-time): Интервал времени между двумя пачками квантования времени.

Примечание — На интервале паузы в соответствующем элементарном потоке transport_packets не поставляются.

3.1.19 **пользователь** (user): Оконечная система, которая может передавать или принимать информацию от других таких же оконечных систем с использованием сети и которая может функционировать как клиент, сервер или как клиент и сервер одновременно.

3.1.20 **поток PID** (PID stream): Совокупность пакетов в транспортном потоке с одинаковыми PID.

3.1.21 **потоковое устройство:** Устройство, которое получает транспортный поток непосредственно и получает информацию о синхронизации непосредственно из транспортного потока.

3.1.22 **приемное устройство** (Receiving Device): Устройство, которое принимает или интерпретирует секции, соответствующие настоящему стандарту. Примерами этих устройств являются сплайсеры, серверы объявления, сегментаторы и спутниковые приемники.

3.1.23 **приложение** (application): 1. Программное обеспечение, предоставляющее клиенту возможность решения определенной задачи и реализуемое в среде клиента. 2. Функциональная реализация программного обеспечения, обслуживающего один или несколько взаимодействующих аппаратных объектов.

3.1.24 **программа («продолжительный» контент)** (program): Набор видео-, аудио- и потоков PID данных, которые в MPTS имеют общий номер программы в соответствии со стандартом ISO/IEC [1].

Используется в контексте дескриптора сегментации, качества функционирования или информативной телевизионной вещательной передачи, обычно с продолжительностью более 5 мин.

3.1.25 **программный поток данных** (Program Stream; PS): Поток данных, образованный путем мультиплексирования элементарных потоков видеоданных и звукоданных цифрового вещательного телевидения, имеющих одну общую тактовую частоту, и сформированный из программных пакетов вещательного телевидения переменной длины.

3.1.26 **профиль** (profile): 1. Описание группы минимальных конфигураций, определяющих параметры потока битов, формируемого одной из совокупностей рассматриваемых систем кодирования (или параметры приемников-декодеров этих потоков) и отображающих функции, которые характеризуют контекст опций службы. 2. Набор средств и инструментов обработки видеосигнала (видео) или аудиосигнала (аудио), использующий предусмотренную стандартом кодирования технологию и формирующий кодированный поток битов.

3.1.27 **размер пачки** (burst size): Количество битов сетевого уровня на интервале пачки квантования времени.

3.1.28 **разрыв (пауза)** (break): Время, выделенное для рекламы, или время фактически происходящей вставки (сплайсинга).

3.1.29 **режим вставки (сплайсинга) компонент (Component Splice Mode):** Режим выдачи сообщения метки, вследствие чего program_splice_flag устанавливается в «0» и указывает, что каждый PID/компонент, который предназначается для вставки (сплайсинга), будет перечислен отдельным синтаксисом. Компоненты, не перечисленные в сообщении, не должны быть вставлены.

3.1.30 **режим вставки (сплайсинга) программы (Program Splice Mode):** Режим формирования сообщения метки, в соответствии с чем флаг program_splice_flag устанавливается в «1» и указывает, что сообщение относится к точке вставки программы и что все PID/компоненты программы должны быть соединены.

3.1.31 **режим немедленной вставки** (Splice Immediate Mode): Режим выдачи сообщения метки, вследствие чего устройство вставки должно выбрать в потоке для выполнения вставки наиболее близкую возможность относительно splice_info_table.

3.1.32 **реклама (рекламное объявление, объявление, «краткий» контент)** (advertisement; ad): Сообщение о предлагаемых товарах или услугах, предстоящих событиях и т. д., обычно с продолжительностью не более 2 мин.

3.1.33 **секция** (section): Синтаксическая структура, используемая для отображения всей сервисной информации в пакетах транспортного потока.

3.1.34 **семантика** (semantics): Система правил, предназначенная для определения смысловых значений отдельных конструкций алгоритмического языка.

3.1.35 **сервис (служба, услуга)** (service): 1. Последовательность программ, которая под управлением вещателя может быть в режиме вещания передана как часть расписания. 2. Логический объект в системе предоставляемых функций и интерфейсов, поддерживающий одно или множество приложений, отличие которого от других объектов заключается в доступе конечного пользователя к управлению шлюзом сервисов.

3.1.36 **синтаксис** (syntax): Часть языка программирования, которая описывает структуру программ как наборов символов.

3.1.37 **системная тактовая частота** (System Time Clock; STC): Тактовая частота, используемая для формирования отсчетов аудио- PTS и видео- PTS.

3.1.38 **событие** (event): Событие вставки или просмотра (визуализации).

3.1.39 **событие вставки** (Splice Event): Возможность соединения одного или более потоков PID; точка на шкале времени.

3.1.40 **событие просмотра (визуализации)** (Viewing Event): Телевизионная программа или фрагмент компрессированного материала в службе.

3.1.41 **сообщение** (message): В контексте этого документа сообщение является содержанием любого поля splice_info_section.

3.1.42 **сплайсер** (splicer): Устройство, выполняющее переключение элементарных потоков.

3.1.43 **точка вставки** (Splice Point): Точка в потоке PID (точка выхода или точка входа).

3.1.44 **точка вставки (сплайсинга) программы** (Program Splice Point): Точка входа программы или точка выхода программы.

3.1.45 **точка входа** (In Point): Точка в потоке, подходящая для ввода, которая находится на границе элементарного модуля представления. Точка входа расположена между двумя модулями представления. Это позиция в потоках битов, где допустимо выполнение процедуры вставки.

3.1.46 **точки входа программы** (Program In Point): Группа точек входа потока PID, которые соответствуют времени представления (воспроизведения) программы.

3.1.47 **точка входа сети** (network In Point): Точка входа сети ленты новостей.

3.1.48 **точка выхода** (Out Point): Точка в потоке, пригодная для выхода, которая находится на границе элементарного модуля представления. Точка фактически расположена между двумя модулями представления. Это позиции в потоке битов, где допустим выход из потока битов.

3.1.49 **точка выхода программы** (Program Out Point): Группа точек выхода потока PID, которые соответствуют времени представления (воспроизведения) программы.

3.1.50 **транспортный поток;** ТП (transport stream; TS): Набор из нескольких программных потоков данных цифрового вещательного телевидения, сформированный из программных пакетов постоянной длины с коррекцией ошибок и независимым тактированием от своих источников синхронизации. Параметры транспортного потока определяются ISO/IEC [1] (подраздел 2.4).

3.1.51 **payload_unit_start_indicator:** Бит в заголовке транспортного пакета, который сигнализирует о том, что секция начинается в следующей за ним полезной нагрузке (стандарт ISO/IEC [1]).

3.1.52 **pointer_field:** Первый байт транспортного пакета полезной нагрузки обязательный, если секция начинается с этого пакета (ISO/IEC [1]).

3.2 В настоящем стандарте применены следующие сокращения:

ВДР — время вещания, доступное для рекламы;

МСЭ (International Telecommunication Union, ITU) — Международный союз электросвязи;

МЭК (International Electrotechnical Commission/Committee, IEC) — Международная электротехническая комиссия;

ОПД — опережающее продвижение данных;

ПЭП (Packetized Elementary Stream, PES) — пакетированный элементарный поток;

ТП (Transport Stream, TS) — транспортный поток (цифрового вещательного телевидения);

ЭП (Elementary Stream, ES) — элементарный поток;

ad (advertisement) — реклама (рекламное объявление, объявление, «краткий» контент);

ADI (Asset Distribution Interface) — интерфейс распределения файлов программ;

ASCII (American Standard Code for Information Interchange) — Американский стандартный код обмена информацией;

CBC (Cipher Block Chaining) — режим сцепления блоков шифра;

DES (Data Encryption Standard) — стандарт шифрования данных;

DTS (decoding time stamp) — временная метка декодирования;

DTMF (Dual-Tone Multiple-Frequency (Touch-Tone)) — многочастотный способ передачи сигналов набора номера;

DVB (Digital Video Broadcasting) — цифровое телевизионное вещание;

EIDR (Entertainment ID Registry Association) — ассоциация регистрации идентификаторов кинофильмов и телевизионных программ;

ES (Elementary Stream) — элементарный поток; ЭП;

ESCR (Elementary Stream Clock Reference) — метка времени в потоке PES;

IEC (International Electrotechnical Commission/Committee) — Международная электротехническая комиссия; МЭК;

INT (IP/MAC Notification Table) — таблица извещения IP/MAC;

IP (Internet Protocol) — Интернет протокол;

ISO (International Standards Organizations) — Международная организация по стандартизации;

ITU (International Telecommunications Union) — Международный союз электросвязи; МСЭ;

LLC(Logical Link Control) — управление логическим каналом;

MPEG (Motion Pictures Expert Group) — группа экспертов по движущимся изображениям (группа стандартов сжатия аудио- и видеоинформации);

MPEG-2 — стандарт цифрового сжатия аудио- и видеоинформации;

MPTS (Multi-Program Transport Stream) — многопрограммный транспортный поток;

NIT (Network Information Table) — таблица сетевой информации;

PCR (Program Clock Reference) — ссылка на программные часы;

PES (Packetized Elementary Stream) — пакетированный элементарный поток; ПЭП;

PID (Packet Identifier) — идентификатор типа пакета;
PMT (Program Map Table) — таблица состава программы;
PS (Program Stream) — программный поток данных;
PSI (Program Specific Information) — программно-зависимая информация;
PTS (Presentation Time Stamp) — метка времени воспроизведения;
RAM (Ramdom Access Memory) — оперативное запоминающее устройство с произвольным доступом;
SI (Service Information) — информация о службах;
SMPTE (Society of Motion Picture and Television Engineers) — Общество инженеров кино и телевидения (США);
STC (System Time Clock) — системная тактовая частота;
TDT (Time and Date Table) — таблица времени и даты;
TOT (Time Offset Table) — таблица смещения времени;
TPS (Transmission Parameter Signalling) — сигнализация параметра передачи;
TS (Transport Stream) — транспортный поток (цифрового вещательного телевидения); ТП;
UP (User Packet) — пакет пользователя;
UPL (User Packet Length) — длина пакета пользователя;
UTC (Universal Time Coordinated) — всемирное координированное время;
XOR (Exclusive OR) — исключающее ИЛИ — логическая булева функция.

3.3 В настоящем стандарте применены следующие аббревиатуры:

bslbf (bit string left bit first**)** — мнемоническое обозначение: «последовательность битов, левый бит первый»;

uimsbf (unsigned integer, most significant bit first) — мнемоническое обозначение последовательности битов: «целое число без знака, старший бит первый».

4 Введение в систему формирования меток

4.1 Общие замечания

Оборудование системы формирования и сигнализации меток обеспечивает формирование и передачу (в том числе в составе транспортного потока MPEG-2) меток, определяющих точки вставки (сплайсинга) в основную программу региональных программ, рекламы, объявлений. Метки представляют собой дополнительные PID, идентифицирующие каждую программу, каждое событие вставки (сплайсинга), время начала и окончания врезки в канал оператора вещания, для которого выполняется врезка. Оборудование системы формирования меток обеспечивает безопасность функционирования шифрованием дополнительных PID.

Настоящий стандарт определяет параметры следующих составляющих процесса формирования меток вставки:

- точек вставки (сплайсинга);
- точек вставки программ;
- событий вставки;
- условий вставки (обработки) контента, находящегося на хранении;
- выбора PID для решения задач вставки;
- потоков сообщений при вставке.

4.2 Точки вставки (сплайсинга)

Точки вставки определяют возможность сращивания компрессированных потоков битов. Они указывают на место в потоке битов, в котором может быть выполнено переключение пакетированного элементарного потока от одного источника (например, основной программы) на пакетированный элементарный поток другого источника (например, региональной программы). Качество изображения и звука при сращивании в таких точках может определяться производительностью сплайсера.

Транспортные потоки создаются мультиплексированием потоков PID. В настоящем стандарте определены два типа точек вставки для потоков PID: точки входа и точки выхода. Группирование точек входа отдельных потоков PID в точки входа программы определяет активирование переключения всех программ (видео- и аудио-). Точки выхода программы инициируют процедуру выхода из программы.

Точки входа и выхода являются воображаемыми точками в потоке битов, расположенными между двумя модулями представляемых элементарных потоков. Точки входа и выхода не обязательно должны быть ориентированы на транспортные пакеты или пакеты PES. Точки входа и выхода могут быть совмещены: граница модуля представления может служить и безопасным местом для выхода из потока битов, и безопасным местом для ввода потока битов.

Простая операция врезки (переключения) формирует данные модуля доступа от одного потока до точки выхода включительно, данные от другого потока начинаются после точки входа первого модуля доступа. Возможна реализация более сложных операций врезки, посредством которых данные до точки выхода или данные после точки входа могут быть изменены устройством вставки (сплайсером). Сплайсер может вставить данные между точкой выхода одного потока и точкой входа другого потока. Настоящий стандарт не нормирует и не ограничивает алгоритм поведения сплайсера во всех случаях.

Примечание — Требования настоящего подраздела информативные и необязательны для исполнения.

4.3 Точки вставки (сплайсинга) программ

Точки входа программы и точки выхода программы представляют собой группы потоков PID точек входа или точек выхода, соответствующего времени воспроизведения.

Точки вставки и точки вставки программы совпадают на интервале воспроизведения, но, как правило, они не размещены друг около друга в транспортном потоке. Это связано с тем, что декодирование компрессированного видео требует больше времени, чем декодирование аудио, точки вставки видео могут отставать от точек вставки аудио на сотни миллисекунд.

Настоящий стандарт определяет два способа сигнализации о точках вставки:

- в случае, когда флаг program_splice_flag установлен в «1», активизируется режим сплайсинга программы, в котором все PID программ могут быть соединены (табличный PID информации о вставке является исключением);
- в случае, когда флаг program_splice_flag установлен в «0», активизируется режим вставки компонентов и в сообщении однозначно определяется, какие PID должны быть соединены, и устанавливается уникальное время вставки каждого компонента. Это сообщение должно быть направлено на сплайсер для обработки данных, включающих потоки и видео, и аудио.

Настоящий стандарт устанавливает уникальное время сплайсинга для каждого компонента программы. Ожидается, что большинство сообщений режима сплайсинга компонентов использует единое время вставки (время вставки по умолчанию) для всех компонентов, как описано в разделе 6 настоящего стандарта. Опционально может устанавливаться отдельное время вставки для каждого компонента. Такой режим должен использоваться, когда один или более компонентов значительно отличаются по времени точки входа или времени точки выхода относительно других компонентов в том же самом сообщении. Примером является загруженный апплет (прикладная программа), который должен достигнуть цифрового приемника за несколько секунд до вставки рекламы.

Примечание — Требования настоящего подраздела информативные и необязательны для исполнения.

4.4 События вставки

Настоящий стандарт устанавливает параметры метода внутриполосной сигнализации о событиях вставки использованием команд вставки для оборудования вставки «нижнего уровня» транспортного потока. Сигнализация события вставки для выполнения вставки идентифицирует в потоке точку вставки. Устройство вставки может действовать или игнорировать сообщенное событие вставки (сообщенное событие должно быть интерпретировано как возможность вставки, а не как команда). Таблица информации о вставке переносит уведомление о возможности события вставки.

Таблица информации о вставке переносится в одном или более потоках PID с назначенным stream_type. PID информации о вставке программ определен в PMT.

Общий stream_type идентифицирует все потоки PID, которые переносят таблицы информации о вставке. Ремультиплексоры или сплайсеры могут использовать поле stream_type, чтобы отбросить информацию о вставке до отправки транспортного потока к устройству конечного пользователя.

Оборудование инжекции метки может отправлять сообщения метки с интервалами, которые не указывают на точку вставки, а используются в качестве сообщений о тактовом сигнале. Такие сообщения позволяют обеспечить правильное функционирование системы. Эта операция может выполняться периодически, выдавая сообщения splice_null() или передавая зашифрованные сообщения

splice_insert. Интервал передачи сообщений длительностью 5 мин можно считать достаточным. В этом случае, если сообщение не было бы получено на 10-минутном интервале, то приемное устройство могло бы подать сигнал оператору о неправильном функционировании системы. Такая функция системы является опциональной.

Примечание — Требования настоящего подраздела информативные и необязательны для исполнения.

4.5 Условия обработки контента, находящегося на хранении

Требования уникальности идентификатора устанавливаются в предположении, что контент будет воспроизводиться в режиме реального времени. Если контент сохранен, то процесс считывания контента не предъявляет требований к оборудованию воспроизведения для изменения идентификаторов (таких как splice_event_id или segmentation_event_id). Правила работы оборудования вставки, расположенного на «нижнем уровне» транспортного потока, анализирующего идентификаторы, должны учитывать, что при появлении нарушения требований уникальности идентификатора необходимо, по возможности, использовать и другую информацию, подтверждающую это нарушение. Рекомендуется формировать отрицательную реакцию на нарушение требования уникальности идентификатора после подтверждения нарушения уникальности идентификатора.

Настоящий стандарт обеспечивает (опционально) инструменты, помогающие формировать более короткие секции сегментированного контента, которые могут быть или главами рекламных объявлений, или полными рекламными объявлениями в соответствии с 7.3.3 настоящего стандарта.

Примечание — Требования настоящего подраздела информативные и необязательны для исполнения.

4.6 Выбор PID для решения задач вставки

4.6.1 Выбор PID

Информация о вставке может переноситься несколькими PID. Максимальное количество таких PID не должно превышать 8. Эти PID могут быть размещены в любых свободных полях (где транспортируемые биты scrambling_control установлены в «0») или скремблированы системой условного доступа. Каждый PID сообщения метки может включать дескриптор cue_identifier_descriptor, описывающий команды вставки, включенные в PID. При использовании нескольких PID для переноса информации о вставке первый PID сообщения метки в таблице PMT должен содержать только команды вставки типа 0x00 (splice_null), 0x04 (splice_schedule) и 0x05 (splice_insert). Кроме того, splice_event_id должен быть уникальным во всех PID информации о вставке в программу.

4.6.2 Выбор PID

В соответствии с 4.6.1 настоящего стандарта допускается использование нескольких PID сообщений метки, однако не все типы оборудования могут обрабатывать потоки, содержащие такое множество PID. Ограничение может возникать из-за ограниченного количества PID, которые оборудование может обработать. Если система использует передачу нескольких PID при необходимости обеспечения высокого технологического уровня системы, рекомендуется выполнять полное тестирование сквозного тракта.

Возможны следующие варианты построения таких систем:

- доставка PID с информацией вставки рекламы на оборудование головной станции не нужна, так как в этих системах устройства вставки или устройства мультиплексирования удалят любые PID с информацией о вставке;

- в системах другого типа возможна выборочная передача определенных PID (с информацией вставки рекламы) к оборудованию головной станции в целях повышения ее функциональности;

- в третьем варианте системы устройства вставки или мультиплексирования группируют множество PID, переносящих информацию о вставке в единственный PID для обработки нисходящего потока.

Выбор варианта системы с игнорированием PID с информацией вставки рекламы или передачи сообщения PID с информацией вставки рекламы рекомендуется оставлять за пользователем с возможностью использования наиболее применимого алгоритма вставки «по умолчанию», выбранного проектировщиком.

Когда устройства вставки или мультиплексирования получают PID со скремблированными битами в заголовке, то «по умолчанию» они должны отбрасывать этот PID и не передавать его на выход. В идеальном случае эту операцию должны настраивать пользователи, поскольку в некоторых случаях этот PID может быть дескремблирован устройством в нисходящем потоке.

В случае идентификации в PMT нескольких PID вставки устройство вставки должно обрабатывать все эти PID. Если используется cue_identifier_descriptor, устройства вставки или устройства мультиплексирования могут использовать эту информацию для повышения селективности обрабатываемых PID.

Возможности использования нескольких PID для идентификации сообщений метки позволяют реализовать выборочную поставку сообщений метки как для разделения различных уровней рекламы, так и для разделения сообщений метки от сообщений сегментации. Один из допустимых методов обработки этих проблем использует методы шифрования, предусмотренные в настоящем стандарте, многие механизмы доставки могут поддерживать поставку PID условно безопасным способом. Оборудование доставки (спутниковый передатчик/приемник, ремультиплексор) может фильтровать поток PID, чтобы разрешить передачу в потоке одного или небольшого количества PID. Решение об использовании одного или более PID будет основано на необходимой безопасности и аппаратных средствах условного доступа, доступных системе.

Примечание — Требования настоящего пункта информативные и необязательны для исполнения.

4.7 Поток сообщений при вставке

Сообщения, описанные в настоящем стандарте, могут формироваться многими источниками. Они разработаны для отправки в потоке к устройствам нисходящего потока. Устройства нисходящего потока могут воздействовать на сообщения или отправлять их на устройства, которые работают «не в потоке», чтобы воздействовать на них. Примером может быть сплайсер, связывающийся через протокол стандарта ANSI [2] с сервером рекламы. Устройства, работающие в потоке (потоковые устройства), могут передать сообщения к следующему устройству по цепочке устройств тракта передачи или (опционально) отбросить сообщения.

Любое устройство, которое повторно маркирует поля PCR/PTS/DTS и передает эти сообщения метки к устройству в нисходящем потоке, должно изменить поле pts_time или поле pts_adjustment в сообщении во всех PID, соответствующих этому стандарту. Изменение поля pts_adjustment более предпочтительно, потому что устройство повторной маркировки может быть недостаточно хорошо информировано относительно содержания поля pts_time, которое может появиться в нескольких командах.

Сообщение bandwidth_reservation() предназначено для использования в замкнутой спутниковой системе (от кодера до приемника). Приемник это сообщение отбросит (заменит нуль-пакетом). Если это сообщение достигает устройства, работающего «в потоке» (например, сплайсера), то оно не должно быть передано на устройство «вне потока» (например, на сервер рекламы) и может быть проигнорировано или передано на устройство «в потоке». Решение об игнорировании сообщения или о его передаче рекомендуется принимать пользователю. Один из этих вариантов решения должен быть установлен в режиме «по умолчанию».

Примечание — Требования настоящего подраздела информативные и необязательны для исполнения.

5 Дескрипторы PMT

Дескрипторы PMT в составе: Дескриптор Регистрации, дескриптор метки cue_identifier_descriptor, дескриптор идентификатора потока обеспечивают идентификацию программ, участвующих в процессах вставки, формирования меток вставки и формирование потоков компонентов.

5.1 Дескриптор регистрации

Дескриптор регистрации согласно ISO/IEC [1] (2.6.8, таблица 2-46) обеспечивает идентификацию программ, которые соответствуют настоящему стандарту. Дескриптор регистрации переносится в цикле PMT program_info для каждой программы, соответствующей настоящему стандарту. Он должен находиться во всех PMT соответствующих программ мультиплекса. Присутствие дескриптора регистрации также указывает, что сигнализацию события вставки splice_info_sections нужно переносить в одном или более потоках PID в этой программе.

Присутствие дескриптора регистрации в PMT сигнализирует о следующем:

- элементы программы не содержат таблицу информации о вставке, определенную в стандарте SMPTE [3];
- в цикле ES_descriptor_loop PMT для PID могут присутствовать только дескрипторы, переносящие splice_information_table, которые определены в этом стандарте, или дескрипторы частных пользователей.

Примечание — Дескриптор регистрации применяется к обозначенной программе, а не ко всему мультиплексу. Контент дескриптора регистрации определяется в таблице 1.

Таблица 1 — Контент дескриптора регистрации

Синтаксис	Количество битов	Мнемосхема
registration_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
SCTE_splice_format_identifier	32	uimsbf
}		

Параметры семантики полей дескриптора регистрации:

descriptor_tag: Поле идентифицирует каждый дескриптор. Для целей регистрации в этом поле должно быть установлено 0x05.

descriptor_length: Поле определяет количество байтов в дескрипторе сразу после поля descriptor_length. Для дескриптора регистрации в descriptor_length должно быть установлено 0x04.

SCTE_splice_format_identifier: Поле, которому SCTE присвоило значение 0x43554549 (ASCII «CUEI»), чтобы идентифицировать программу (в мультиплексе).

5.2 Дескриптор идентификатора метки

5.2.1 Дескриптор cue_identifier_descriptor может использоваться в PMT, чтобы маркировать PID, которые переносят команды вставки так, чтобы их можно было различать по типу или по уровню команд вставки, которые они переносят. Дескриптор cue_identifier_descriptor должен быть расположен в элементарном цикле дескриптора. Если cue_identifier_descriptor не используется, поток может перенести любую команду, допустимую в этой спецификации. Кодирование дескриптора идентификатора метки осуществляется в соответствии с таблицей 2.

Таблица 2 — Кодирование дескриптора идентификатора метки cue_identifier_descriptor

Синтаксис	Количество битов	Мнемосхема
cue_identifier_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
cue_stream_type	8	uimsbf
}		

5.2.2 Параметры семантики полей в дескрипторе идентификатора метки:

descriptor_tag: Поле идентифицирует каждый дескриптор. Для дескриптора cue_identifier_descriptor в этом поле должно быть установлено 0x8A.

descriptor_length: Поле определяет количество байтов дескриптора сразу после поля descriptor_length. Для этого дескриптора в descriptor_length должно быть установлено 0x01.

cue_stream_type: Поле определяет значения сообщений, передаваемых в поле **cue_stream_type,** в соответствии с таблицей 3.

Таблица 3 — Определение поля **cue_stream_type** дескриптора идентификатора метки

Значение поля **cue_stream_type**	Использование PID
0x00	splice_insert, splice_null, splice_schedule
0x01	Все команды

Окончание таблицы 3

Значение поля **cue_stream_type**	Использование PID
0x02	Сегментация
0x03	Вставка, разделенная на уровни
0x04	Сегментация, разделенная на уровни
0x05 — 0x71	Зарезервировано
0x80 — 0xff	Определяется пользователем

5.2.3 Описание использования **cue_stream_type**:

0x00: splice_insert, splice_null, splice_schedule — только эти сообщения сигнала допускаются в этом потоке PID. Должно быть не более одного PID, идентифицированного с этим **cue_stream_type**. Если этот PID будет существовать, то это должен быть первый поток, выполняющий настоящий стандарт в цикле PMT элементарного потока.

0x01: все команды — значение устанавливается «по умолчанию», если этот дескриптор не присутствует. В этом PID могут использоваться все сообщения.

0x02: сегментация — этот PID переносит команду time_signal и дескриптор сегментации. Он может перенести все другие команды, если это нужно для приложения, но его основная цель состоит в том, чтобы передать информацию о сегментации контента.

0x03: вставка, разделенная на уровни, — вставка, разделенная на уровни, ссылается на систему вставки, в которой оператор обеспечивает различные возможности вставки программы в данном ВДР для различных клиентов. Физическая и логическая реализация может быть выполнена несколькими способами, некоторые из них не рассматриваются в настоящем стандарте.

0x04: сегментация, разделенная на уровни, — разделенная на уровни сегментация обращается к системе, где оператор предоставляет различные возможности сегментации программы различным клиентам. Физическая и логическая реализация может быть выполнена несколькими способами, часть их в настоящем стандарте не рассматривается.

0x05-0x7F: зарезервировано для будущих расширений настоящего стандарта.

0x80-0xFF: определяется пользователем.

5.3 Дескриптор идентификатора потока

Дескриптор идентификатора потока может использоваться в PMT для маркировки потоков компонентов службы для их дифференцирования. Дескриптор идентификатора потока должен быть расположен в элементарном цикле дескриптора после соответствующего поля ES_info_length. Дескриптор идентификатора потока должен использоваться, если program_splice_flag или program_segmentation_flag будет иметь нулевое значение. Если используются дескрипторы идентификатора потока, то они должны быть представлены в PMT при каждом возникновении цикла элементарного потока и должны иметь уникальный тег компонента данной программы.

Кодирование дескриптора идентификатора потока осуществляется в соответствии с таблицей 4.

Таблица 4 — Кодирование дескриптора идентификатора потока

Синтаксис	Количество битов	Мнемосхема
stream_identifier_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
component_tag	8	uimsbf
}		

Параметры семантики полей дескрипторов идентификатора потока:

descriptor_tag: Поле идентифицирует каждый дескриптор. Для stream_identifier_descriptor в этом поле должно быть установлено 0x52.

descriptor_length: Поле определяет количество байтов дескриптора сразу после поля descriptor_length. Для этого дескриптора в поле descriptor_length должен быть установлен 0x01.

component_tag: Поле идентифицирует поток компонентов для того, чтобы связать его с описанием, данным в дескрипторе компонента. В секции карты программы каждый дескриптор идентификатора потока должен иметь различное значение этого поля.

6 Таблица информации о вставке

6.1 Краткий обзор

Таблица информации о вставке предоставляет информацию о командах и управлении. Она уведомляет сплайсер о событиях вставки до наступления этих событий. Таблица информации о вставке предназначена для размещения вставки рекламы в сетевых каналах новостей. В этой среде примеры событий вставки могут включать два варианта:

1) вставка из сетевой ленты (канала) новостей в сервер рекламы;

2) вставка из сервера рекламы в сетевую ленту (канал) новостей.

Таблица информации о вставке может отправляться многократно, событие вставки событий может быть отменено. Синтаксис splice_info_section определен в целях передачи таблицы информации о вставке. Секция splice_info_section переносится в одном или в нескольких потоках PID с PID( ), объявленным в PMT этой программы.

Каждое событие вставки однозначно определяется splice_event_id. Информация о событии вставки может быть передана тремя способами:

- заблаговременным планированием;
- опережающим предупреждением;
- командой на выполнение события вставки в указанных точках вставки.

Эти три типа сообщений передаются в секции splice_info_section. Поле splice_command_type определяет тип отправляемого сообщения. В зависимости от значения этого поля к синтаксису применяются различные ограничения.

Нормируются следующие типы команды:

- splice_null();
- splice_schedule();
- splice_insert();
- time_signal();
- bandwidth_reservation().

Если приемное устройство не поддерживает какую-либо команду, оно может проигнорировать всю секцию splice_info_section.

Команда splice_null() обеспечивает расширяемость функций. Она может использоваться в качестве средства подтверждения работоспособности оборудования вставки, работающего в нисходящем потоке.

Команда splice_schedule() позволяет заблаговременную передачу расписания событий вставки.

Команда splice_insert() должна отправляться не менее одного раза перед каждой точкой вставки. Пакеты, содержащие полную splice_info_table, всегда должны предшествовать пакету, который содержит соответствующую точку вставки (то есть первый пакет, который содержит первый байт модуля доступа, время представления которого наиболее близко соответствует времени, сообщенному в splice_info_section).

Для заблаговременного предупреждения о предстоящей вставке (способ опережающего предупреждения) команда splice_insert() перед точкой вставки может быть отправлена несколько раз. Например, команда splice_insert() может быть отправлена за 8, 5, 4 и 2 секунды до отправки пакета с соответствующей точкой вставки. Однако при приеме любого сообщения, полученного менее чем за 4 с после заблаговременного предупреждения, необходимый результат может быть не достигнут. Сообщение splice_insert() должно отправляться не менее одного раза, не менее чем за 4 с перед требуемым временем вставки для условия точки выхода сети. Рекомендуется при отправлении сообщения о возврате к сети (точка входа) обеспечивать опережающую отправку предварительного уведомления не менее чем за 4 с.

Команда splice_insert() предусматривает дополнительную структуру break_duration(), идентифицирующую длину рекламной паузы. Рекомендуется, чтобы сообщения splice_insert() с полем out_of_network_indicator, установленным в 1 (точка выхода из сети канала новостей), включали структуру

break_duration(), чтобы предоставить сплайсеру указание относительно того, когда произойдет точка входа сети канала новостей. Структура break_duration() предусматривает опционально флаг auto_return, который, установленный в 1, указывает, что сплайсер должен возвратиться к сети в конце перерыва ТВ-передачи (определяемого как режим автоматического возврата согласно 6.5.2.2 настоящего стандарта). Рекомендуется, чтобы режим автоматического возврата использовался для динамической поддержки продолжительности ВДР.

Команда time_signal() обеспечивает расширяемость при сохранении точной синхронизации, разрешенной командой splice_insert(). Расширяемость обеспечивает при использовании новых функций, непосредственно косвенно не связанных со вставкой, использование возможностей синхронизации, предусмотренных настоящим стандартом, при минимальном воздействии на устройства вставки. Это позволяет устройству, вставляющему время в сообщение метки, обеспечивать определенное расположение меток.

Команда bandwidth_reservation() позволяет устройствам вставки команды использовать согласованную пропускную способность транспортного потока. Дескрипторы могут использоваться в этой команде, но они не будут обработаны, а будут отправлены в нисходящий поток, чтобы обеспечить информацию сигнализации.

Предусмотрены два метода, позволяющие изменить параметры введенной команды. Один метод отменяет данную команду, отправляя секцию splice_info_section с набором splice_event_cancel_indicator и затем отправляя новую секцию splice_info_section с корректными или новыми параметрами. Второй метод предусматривает отправление следующего сообщения с новыми данными (не отменяя старое сообщение через сообщение метки с набором битов splice_event_cancel_indicator).

6.1.1 Нарушения непрерывности базового времени

При нарушении непрерывности базового времени пакеты, содержащие команду splice_insert() или time_signal() со временем, выраженным в новом базовом времени, не должны прибывать до возникновения разрыва базового времени. Пакеты, содержащие команды splice_insert() или time_signal() со временем, выраженным в предыдущей базе, не должны прибывать после возникновения разрыва базового времени (стандарт ISO/IEC [4]).

Полный синтаксис и сопровождающие его ограничения представляются ниже.

6.2 Секция информации вставки

Секция splice_info_section должна передаваться в транспортных пакетах. В любом транспортном пакете может быть размещена только одна секция или частичная секция. Splice_info_sections должна всегда начинаться в начале полезной нагрузки транспортного пакета. В начале секции в транспортном пакете должно присутствовать поле pointer_field с установленным 0x00, в поле payload_unit_start_indicator битов должна быть установлена «1» (в соответствии с требованием раздела об использовании синтаксиса стандарта ISO/IEC 13818 [1]).

Кодирование секции информации вставки осуществляется в соответствии с таблицей 5.

Таблица 5 — Кодирование секции информации вставки

Синтаксис	Количество битов	Мнемосхема	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	

Окончание таблицы 5

Синтаксис	Количество битов	Мнемосхема	Зашифровано
reserved	12	bslbf	
splice_command_length	12	uimsbf	
splice_command_type	8	uimsbf	E
if(splice_command_type == 0x00)			
splice_null()			E
if (splice_command_type == 0x04)			
splice_schedule()			E
if (splice_command_type == 0x05)			
splice_insert()			E
if (splice_command_type == 0x06)			
time_signal()			E
if (splice_command_type == 0x07)			
bandwidth_reservation()			E
if (splice_command_type == 0xff)			
private_command()			E
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	
}			

Параметры семантики полей в секции splice_info_section():

table_id: Значение поля должно быть 0xFC.

section_syntax_indicator: Поле, которое должно всегда устанавливаться в «0», что является указанием необходимости использования короткой секции MPEG.

private_indicator: Поле, в котором должен быть установлен «0».

section_length: Поле определяет количество остающихся байтов в splice_info_section, сразу следующей за полем section_length до конца splice_info_section. Значение этого поля не должно превышать «4093».

protocol_version: Поле предназначено для использования в будущем для переноса параметров, которые могут быть структурированы иначе, чем определенные в настоящем стандарте. В настоящее время единственным допустимым значением поля protocol_version является «0». Ненулевые значения поля protocol_version могут использоваться в будущих версиях этого стандарта для обозначения таблиц с различной структурой.

encrypted_packet: Поле, установленное в «1», означает, что шифруются части splice_info_section, начинающейся с поля splice_command_type и заканчивающейся полем E_CRC_32. Когда этот бит устанавливается в «0», части этого сообщения не шифруются. Части, которые потенциально могут быть зашифрованы, в splice_info_table обозначаются символом «E» в столбце «Зашифровано» таблицы 5.

encryption_algorithm: Поле определяет примененный алгоритм шифрования текущего сообщения. Если в поле encrypted_packet установлен «0», то это означает, что шифрование не применяется. Детализация использования этого поля показана в подразделе 8.3 настоящего стандарта.

pts_adjustment: Поле используется устройством вставки в качестве значения смещения времени, которое будет добавлено к (иногда) зашифрованному полю или полям pts_time, чтобы получить предназначенное время вставки. В случае, если в этом поле установлен «0», поле pts_time должно ис-

пользоваться без смещения. Как правило, создатель сообщения метки в этом поле устанавливает «0». Это значение является средством, которое устройство, расположенное «вверх по потоку» и которое повторно маркирует PCR/PTS/DTS, может передать устройству вставки средства, которыми можно преобразовать сообщения поля pts_time к недавно наложенному временному интервалу.

В этом случае первое устройство, которое повторно маркирует PCR/PTS/DTS и передает сообщение метки, вставит в поле pts_adjustment значение, которое является разницей между входным временным интервалом этого устройства и его выходным временным интервалом (время «дельта»). Все последующие устройства, которые повторно маркируют PCR/PTS/DTS, могут изменять pts_adjustment поле, добавляя их собственное время «дельта» к существующему полю время «дельта» и помещая результат назад в поле pts_adjustment. После каждого изменения поля pts_adjustment устройство, вносящее изменения, должно повторно вычислить и обновить поле CRC_32.

Поле **pts_adjustment** должно иметь собственное значение для преобразования поля pts_time к оси текущего времени. Преобразование выполняется суммированием этих двух полей. В случае переполнения перенос поля должен быть проигнорирован.

cw_index: Поле передает слово управления (ключ) для дешифровки сообщения. Устройство вставки может хранить до 256 ключей, введенных ранее. Когда бит флага encrypted_packet установлен в «0», это поле присутствует, но не содержит информации.

splice_command_length: Поле определяет длину команды вставки. Устройства, совместимые с настоящим стандартом, должны заполнить это поле с указанием фактической длины. Значение 0xfff позволяет поддерживать обратную совместимость.

splice_command_type: Поле принимает одно из значений, приведенных в таблице 6.

Таблица 6 — Значения, принимаемые полем **splice_command_type**

Значение поля **splice_command_type**	Команда
0x00	splice_null
0x01	Зарезервировано
0x02	Зарезервировано
0x03	Зарезервировано
0x04	splice_schedule
0x05	splice_insert
0x06	time_signal
0x07	bandwidth_reservation
0x08 — 0x0fe	Зарезервировано
0xff	private_command

descriptor_loop_length: Поле определяет число байтов, используемых в цикле дескриптора вставки сразу после этого поля.

alignment_stuffing: При шифровании это поле является функцией выбранного алгоритма шифрования. Так как некоторые алгоритмы шифрования требуют определенной длины для зашифрованных данных, допускается вставка байтов стаффинга. Например, алгоритм DES требует для шифрования пакета кратности 8 байтам последовательности данных. Это позволяет использовать стандартный DES в отличие от специальной версии алгоритма шифрования. Это поле не должно использоваться для переноса действительных данных , если шифрование не используется, но передача этого поля допустима.

E_CRC_32: Поле, которое содержит значение CRC, в соответствии с ISO/IEC 13818 [1], после обработки дешифрованной части поля splice_info_section. Это поле предназначается для индикации успешного выполнения дешифрования. Следовательно, нулевой вывод получается после дешифрования и обработки полей splice_command_type через E_CRC_32.

CRC_32: Поле, которое содержит значение CRC в соответствии с ISO/IEC 13818 [1], после обработки всего поля splice_info_section, включающего поле table_id. Обработка CRC_32 должна произойти до дешифрования зашифрованных полей и должна использовать зашифрованные поля в их зашифрованном состоянии.

6.3 Команды вставки

6.3.1 Команда splice_null()

Команда splice_null() обеспечивает расширяемость стандарта. Команда splice_null() позволяет отправлять таблицу splice_info_table, которая может перенести дескрипторы, при отсутствии необходимости отправлять одну из определенных команд. Эта команда может также использоваться в качестве сообщения «подтверждения работоспособности» для того, чтобы контролировать целостность оборудования инжекции метки и целостность канала.

6.3.2 Команда splice_schedule()

Команда splice_schedule() обеспечивает передачу расписания событий вставки с опережением. Кодирование команды splice_schedule() выполняется в соответствии с таблицей 7.

Таблица 7 — Кодирование команды splice_schedule()

Синтаксис	Количество битов	Мнемосхема
splice_schedule() {		
splice_count	8	uimsbf
for (i=0; i <splice_count; i ++) {		
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
reserved	5	bslbf
if (program_splice_flag == '1')		
utc_splice_time	32	uimsbf
if (program_splice_flag == '0') {		
component_count	8	uimsbf
for (j=0; j <component_count; j ++) {		
component_tag	8	uimsbf
utc_splice_time	32	uimsbf
}		
}		
if (duration_flag)		
break_duration ()		
unique_program_id	16	uimsbf
avail_num	8	uimsbf
avails_expected	8	uimsbf
}		
}		
}		

Параметры семантики полей в splice_schedule():

splice_count: Поле указывает на количество событий вставки, определенных в цикле, который следует в настоящее время.

splice_event_id: Поле является уникальным идентификатором события вставки.

splice_event_cancel_indicator: Значение флага «1» указывает, что ранее отправленное событие вставки, идентифицированное splice_event_id, было отменено.

out_of_network_indicator: Значение флага «1» указывает, что событие вставки означает возможность выхода из сети ленты новостей и что значение utc_splice_time должно относиться к намеченной точке выхода или к точке выхода программы. Значение флага «0» означает, что событие вставки может возвратиться в сеть ленты новостей и что значение utc_splice_time должно относиться к намеченной точке входа или к точке входа программы.

program_splice_flag: Значение флага, установленного в «1», указывает, что сообщение относится к точке вставки программы, и что режим является режимом вставки программы, при котором все PID/компоненты программы должны быть соединены. Когда флаг установлен в «0», это означает, что имеет место режим вставки компонент. Синтаксис режима вставки компонент перечислен ниже.

duration_flag: Флаг указывает на присутствие поля break_duration().

utc_splice_time: Поле представляет сигнализированное время события вставки, включающее количество секунд, прошедших с 00 часов UTC 6 января 1980. Поле **utc_splice_time** может быть преобразовано в UTC без использования значения GPS_UTC_offset, обеспеченного таблицей системного времени. Поле **utc_splice_time** используется только в команде **splice_schedule()**.

component_count: Поле определяет количество экземпляров компонентов (элементарных потоков PID) в цикле, который следует в настоящее время. Компоненты эквивалентны элементарным потокам PID.

component_tag: Поле идентифицирует элементарный поток PID, содержащий точку вставки, определенную значением **splice_time()**. Значение поля должно соответствовать значению, используемому в дескрипторе stream_identification_descriptor(), для идентификации этого элементарного потока PID.

unique_program_id: Поле обеспечивает уникальную идентификацию события визуализации службы. Детализированные указания об установке значений для этого поля должны быть в соответствии со стандартом ANSI/SCTE [5].

avail_num: Поле обеспечивает идентификацию определенного ВДР в одном поле **unique_program_id**. Его значение будет увеличиваться с каждым новым ВДР с событием просмотра (визуализации).

avails_expected (ранее — **avail_count**): Поле содержит результаты подсчета количества ожидаемых отдельных ВДР в текущем событии визуализации. Когда в этом поле установлен «0», оно указывает, что у поля **avail_num** значения нет.

6.3.3 Команда splice_insert()

Команда splice_insert() должна передаваться не менее одного раза для каждого события вставки. Справка об использовании этого сообщения приведена в подразделе 5.3 настоящего стандарта. Кодирование команды splice_insert() осуществляется в соответствии с таблицей 8.

Таблица 8 — Кодирование команды splice_insert()

Синтаксис	Количество битов	Мнемосхема
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		

Окончание таблицы 8

Синтаксис	Количество битов	Мнемосхема
for(i=0; i <component_count; i ++) {	8	uimsbf
component_tag		
if(splice_immediate_flag == '0')	8	uimsbf
splice_time()		
}		
}		
if(duration_flag == '1')		
break_duration()		
unique_program_id	16	uimsbf
avail_num	8	uimsbf
avails_expected	8	uimsbf
}		
}		

Параметры семантики полей в splice_insert():

splice_event_id: Поле является уникальным идентификатором вставки события.

splice_event_cancel_indicator: Значение флага, установленного в «1», указывает, что ранее отправленное сообщение о вставке события, идентифицированное полем splice_event_id, было отменено.

out_of_network_indicator: Значение флага, установленного в «1», указывает, что сообщение о событии вставки означает возможность выхода из сети новостей и что значение splice_time(), измененное полем pts_adjustment, должно относиться к намеченному значению для точки выхода или точки выхода программы. Этот флаг, установленный в «0», указывает, что сообщение о событии вставки дает возможность возвратиться к сети новостей, и что значение splice_time(), измененное pts_adjustment, должно вернуться к намеченному значению для точки выхода или точки выхода программы.

program_splice_flag: Значение флага, установленного в «1», указывает, что сообщение относится к точке вставки программы и что режим является режимом вставки программы, в котором все PID/компоненты программы должны быть соединены. Этот флаг, установленный в «0», указывает, что режим является режимом вставки компонентов, в котором каждый компонент будет перечислен отдельно с синтаксисом, описанным ниже.

duration_flag: Значение флага, установленного в «1», указывает на присутствие поля break_duration().

splice_immediate_flag: Значение флага, установленного в «1», указывает на отсутствие поля splice_time() и что вставка должна быть выполнена в режиме немедленной вставки, в котором устройство вставки должно выбрать самую близкую возможность вставки в поток относительно пакета информации о вставке. Когда этот флаг установлен в «0», он указывает на присутствие поля splice_time(), по крайней мере, в одной команде splice_insert().

component_count: Поле определяет количество экземпляров в цикле элементарных потоков PID, который следует в настоящее время. Компоненты эквивалентны элементарным потокам PID.

component_tag: Поле идентифицирует элементарный поток PID, содержащий точку вставки, определенную значением splice_time(), которое следует. Значение поля должно соответствовать значению, используемому в дескрипторе stream_identification_descriptor(), для идентификации этого элементарного потока PID.

unique_program_id: Поле обеспечивает уникальную идентификацию события визуализации службы. Детализированные указания об установке значений для этого поля должны быть в соответствии со стандартом ANSI/SCTE [5].

avail_num (ранее — avail): Поле обеспечивает идентификацию определенного ВДР в одном поле unique_program_id.

avails_expected (ранее — avail_count): Поле обеспечивает результаты подсчета количества ожидаемых отдельных ВДР в текущем событии визуализации. Когда поле установлено в «0», оно указывает, что поле avail_num не имеет значения.

6.3.4 Команда time_signal()

Команда time_signal() обеспечивает синхронизацию механизма поставки данных. Синтаксис команды time_signal() позволяет выполнять синхронизацию информации, которую переносят в сообщении системной тактовой частоты (STC). Уникальная полезная нагрузка сообщения переносится в дескрипторе, однако синтаксис и возможности транспортирования, предоставляемые сообщением splice_insert(), обеспечиваются также time_signal().

Если флаг time_specified_flag будет установлен в «0», что указывает на отсутствие в сообщении pts_time, то команда не должна интерпретироваться как команда непосредственной вставки. В этом случае при непосредственной вставке точность синхронизации может быть неопределенно низкой.

Так как команда time_signal() использует дескрипторы для большей части конкретной информации, то длина этой команды может превысить длину одного транспортного пакета MPEG. Рекомендуется, при возможности, размещать эту команду в одном пакете. Это не всегда возможно в ситуациях, например, когда уникальная информация имеет большой объем или когда используется другая спецификация для определения этой уникальной информации.

Кодирование команды time_signal() должно выполняться в соответствии с таблицей 9.

Таблица 9 — Кодирование команды time_signal()

Синтаксис	Количество битов	Мнемосхема

time_signal() {
splice_time()
}		

Семантика команды time_signal() ассоциируется с семантикой любого дескриптора (или дескрипторов) pts_time, как это предусмотрено синтаксисом splice_info_section согласно таблице 5 настоящего стандарта. Дескрипторы вставки описаны в разделе 7 настоящего стандарта.

6.3.5 Команда bandwidth_reservation()

Команда bandwidth_reservation() предназначена для резервирования пропускной способности мультиплекса. Обычно она используется в спутниковой системе доставки, которая требует, чтобы пакеты некоторых PID всегда присутствовали с предусмотренной частотой повторения для того, чтобы гарантировать определенную пропускную способность для этих PID. Отличия этого сообщения от команды splice_null() обеспечивает приемному оборудованию возможность его удаления из мультиплекса. Если дескриптор передается с этой командой, то нельзя ожидать, что он перенесен через весь тракт передачи. В этом случае должен применяться частный дескриптор, который используется только процессом резервирования пропускной способности.

6.3.6 Параметры структуры private_command()

Структура private_command() обеспечивает возможность распределения команд, определяемых пользователем при использовании протокола настоящего стандарта. Первое поле последовательности битов в каждой определяемой пользователем команде является 32-битовым идентификатором, уникальным для каждого участвующего поставщика. Оборудование должно пропускать любое сообщение splice_info_section(), содержащее структуры private_command() с неизвестными идентификаторами.

Кодирование структуры private_command() выполняется в соответствии с таблицей 10.

Таблица 10 — Кодирование структуры private_command()

Синтаксис	Количество битов	Мнемосхема
private_command() {		
identifier	32	uimsbf
for(i=0; i <N; i++) {		
private_byte	8	uimsbf
}		
}		

Параметры семантики полей в private_command():

identifier: Поле для registration_descriptor() format_identifier используется для идентификации пользователя команды в соответствии с ISO/IEC [1] (2.6.8, 2.6.9). Должны использоваться только значения идентификатора, зарегистрированного SMPTE Registration Authority, LLC. Его использование в структуре private_command() должно быть в рамках контекста и идентификации только частной информации, содержащейся в этой команде.

private_byte: Остаток от дескриптора выделяется для полей данных в соответствии с требованиями дескриптора.

Использование таких данных допустимо только при применении частных средств для передачи для поставщика детализированной уникальной вспомогательной информации.

6.4 Параметры структур времени вставки события и продолжительности события вставки

6.4.1 Параметры структуры splice_time()

Структура splice_time() при изменении pts_adjustment определяет время события вставки. Кодирование структуры splice_time() выполняется в соответствии с таблицей 11.

Таблица 11 — Кодирование структуры splice_time()

Синтаксис	Количество битов	Мнемосхема
splice_time() {		
time_specified_flag	1	bslbf
if(time_specified_flag == 1) {		
reserved	6	bslbf
pts_time	33	uimsbf
}		
else		
reserved	7	bslbf
}		

Семантическое определение полей в структуре splice_time()

time_specified_flag: Значение флага, установленного в «1», указывает на присутствие поля pts_time и связанных резервированных битов.

pts_time: Поле указывает время в периодах частоты 90 кГц. Это поле, в случае изменения pts_adjustment, представляет время намеченной точки вставки.

6.4.2 Параметры структуры break_duration()

Структура break_duration() позволяет определять продолжительность рекламной передачи (коммерческой паузы). Она может применяться сплайсером как информация о времени окончания рекламной паузы и о времени точки входа сети.

Кодирование структуры break_duration() выполняется в соответствии с таблицей 12.

Таблица 12 — Кодирование структуры break_duration()

Синтаксис	Количество битов	Мнемосхема
break_duration() {		
auto_return	1	bslbf
reserved	6	bslbf
duration	33	uimsbf
}		

Определение семантики полей в структуре break_duration():

auto_return: Флаг, установленный в «1», означает, что поле duration (продолжительность) должно использоваться устройством вставки для определения времени возврата к сети новостей (конец разрыва (паузы)). Команда splice_insert() с установкой флага out_of_network_indicator в «0» не предназначается для передачи в целях завершения этой паузы. Когда флаг auto_return установлен в «0», поле duration (продолжительность) не должно использоваться для окончания паузы, потому что для окончания паузы будет отправлена новая команда splice_insert(). В этом случае наличие поля break_duration действует как дублирующий механизм, если команда splice_insert() в конце паузы будет потеряна.

duration: Поле, которое указывает на прошедшее время в отсчетах периодов частоты на 90 кГц.

6.5 Ограничения

6.5.1 Ограничения на splice_info_section()

Секция splice_info_section переносится в одном или более потоках PID, которые являются специфичными для программы и привязаны к PMT. PID секции (или секций) splice_info_section() должен быть идентифицирован в PMT stream_type значением 0x86.

Секция splice_info_section должна содержать только информацию о событиях вставки, которые происходят в этой программе.

Событие вставки должно быть определено единственным значением splice_event_id.

Если будет использоваться режим вставки компонентов, то каждый элементарный поток PID должен быть идентифицирован stream_identifier_descriptor, который переносят в цикле PMT, единственном для каждого PID. Дескриптор stream_identifier_descriptor должен переносить component_tag, который уникально соответствует одному потоку PID среди содержащихся в программе и перечисленных в PMT этой программы.

Любой splice_event_id, который передается в секции splice_info_section, используя команду splice_schedule(), должен быть отправлен еще раз до события, используя команду splice_insert(). Следовательно, должно обеспечиваться соответствие между значениями splice_event_id, выбранными для определенных событий, сообщенных командой splice_schedule() (далекое будущее), и значениями splice_event_id, используемыми в команде splice_insert() (ближайшее будущее), чтобы индицировать одни и те же события.

Значения идентификатора в сообщениях splice_event_id не должны увеличиваться с каждым последующим сообщением, в том числе они не должны увеличиться хронологически. Значения splice_event_id выбираются случайным образом. Используя команды splice_schedule (), splice_event_id, значения должны быть уникальными за период команды splice_schedule(). Значение splice_event_id может быть использовано вновь лишь после окончания связанного с ним времени вставки.

Когда флаг splice_immediate_flag установлен в «1», время вставки должно интерпретироваться как текущее время. Такой режим называют режимом немедленной вставки. Когда эта форма используется с командой splice_insert(), вставка может произойти при самой близкой (предшествующий или последующий) возможности, которую обнаруживает сплайсер. Режим немедленной вставки может использоваться и для точек входа и для точек выхода, то есть для обоих состояний индикатора out_of_network_indicator.

Любое ВДР может быть закончено сообщением «режим вставки программы», сообщением «режим вставки компонентов» или отсутствием сообщения (посредством использования break_duration) независимо от природы сообщения в начале ВДР.

6.5.2 Ограничения интерпретации времени

6.5.2.1 Ограничения на splice_time() для splice_insert()

Для поля splice_command_type, равного 0x05 (соответствует команде splice_insert согласно таблице 6 настоящего стандарта), применяются следующие ограничения на параметры splice_time():

- не менее одного сообщения для точки выхода сети должно прибыть не менее чем за 4 с перед сигнализированным временем вставки (pts_time), если время задано. Сообщение режима немедленной вставки разрешено применять для точки выхода сети, но фактическое время вставки не задано и рекомендуется, чтобы сообщения режима немедленной вставки использовались для раннего завершения рекламных пауз. Когда сообщение метки режима не-непосредственной вставки применяется для точек входа сети, сообщение метки должно прибывать на сплайсер перед прибытием на приемник изображения, сигнализированного в точке входа;
- точка выхода находится между двумя модулями представления (воспроизведения). Намеченная точка выхода просигнализированного события вставки должна быть такой, которая появляется сразу перед модулем представления, время воспроизведения которого наиболее близко соответствует сообщенному pts_time;

- точка входа находится между двумя модулями представления. Намеченная точка входа просигнализированного события вставки должна размещаться (устанавливаться) до начала модуля представления, время представления которого наиболее близко соответствует сообщенному сигнализированному pts_time;

- если в режиме вставки компонентов в поле out_of_network_indicator установлена «1» (начало паузы), то каждый компонент, перечисленный в цикле компонента splice_insert(), должен быть в указанное время переключен от компонента сети к сплайсеру. Компоненты, не перечисленные в цикле компонентов сообщения splice_insert(), останутся не замененными:

- если выходной компонент сплайсера был сетевым компонентом, то он останется сетевым компонентом;

- если выходной компонент сплайсера был компонентом, предоставленным сплайсером, то он останется компонентом, предоставленным сплайсером;

- если в режиме вставки компонентов out_of_network_indicator установлен в «0» (конец паузы), то каждый компонент, перечисленный в цикле компонентов splice_insert(), в обозначенное время должен быть переключен от сплайсера, предоставившего компонент, к сетевому компоненту. Компоненты, не перечисленные в компонентном цикле сообщения, останутся неизменными:

- если выходной компонент сплайсера был сетевым компонентом, то он останется сетевым компонентом;

- если выходной компонент сплайсера был предоставленным компонентом сплайсера, то он останется компонентом, предоставленным сплайсером;

- если в режиме вставки компонентов режим непосредственной вставки не активизирован, то у команды первого компонента, перечисленного в цикле компонентов splice_insert(), должен быть допустимый pts_time в его связанном splice_time() и этот pts_time упоминается как значение pts_time по умолчанию. Следующие за первым компоненты, перечисленные в цикле компонента того же самого сообщения, у которых нет соотвествующего pts_time, используют это значение pts_time «по умолчанию». Однако нееобходимо учитывать, что любой компонент после первого перечисленного компонента команды splice_insert() может содержать уникальный pts_time, отличающийся от значения pts_time «по умолчанию»;

- в режиме вставки компонентов все значения pts_time, содержащиеся в цикле компонентов splice_insert, должны быть изменены полем pts_adjustment, чтобы получить для каждого конкретного случая значение для сигнализированной точки выхода или точки входа. Значение поля pts_adjustment, предоставленного любым устройством, которое генерирует или изменяет pts_adjustment, должно применяться ко всем полям pts_time в сообщении.

6.5.2.2 Ограничения на break_duration() для splice_insert()

При значении поля splice_command_type, равного 0x05, к полю break_duration() будут применяться следующие ограничения:

- значение, данное в break_duration(), интерпретируется как предполагаемая продолжительность рекламной паузы. Это поле используется опционально, когда out_of_network_indicator установлен в «1». Это поле может использоваться в той же самой команде splice_insert(), которая определяет время начала разрыва так, чтобы сплайсер мог вычислить время, когда разрыв будет закончен;

- паузы могут быть завершены выпуском команды splice_insert() с установкой out_of_network_indicator в «0». Поле splice_time() может быть задано или может быть использован переход в режим непосредственной вставки. Когда поле break_duration было дано в начале разрыва (где auto_return был обнулен), значение break_duration может быть использовано как резервный механизм для того, чтобы обеспечить выполнение возврата к сети в случае потери пакета метки;

- паузы могут также быть завершены, устанавливая продолжительность паузы в ее начале и полагаясь на устройство вставки, чтобы возвратиться к сети новостей в свое время, при этом флаг auto_return должен быть установлен в «1». Далее этот режим будет упоминаться как режим автоматического возврата. В состоянии «пауза» (рекламная вставка) в режиме автоматического возврата сообщения метки в конце паузы с установкой индикатора out_of_network_indicator в «0» не требуются и не отвергаются. Следовательно, приемное устройство при нормальном функционировании не должно ожидать сообщения метки в конце паузы. В режиме автоматического возврата паузы могут быть завершены досрочно. Для досрочного завершения паузы может быть выпущена вторая команда splice_insert(), где в out_of_network_indicator устанавливается «0». Новое время паузы может быть дано обновленным сообщением splice_time() или использованием сообщения режима непосредственной вставки. Сообщение метки с установкой индикатора out_of_network_indicator в «0» должно переопределять поле продолжительности предыдущего сообщения метки (с установкой out_of_network_indicator в «1»), если просигнализированная продолжительность этой паузы будет все еще действительна.

7 Дескрипторы вставки

7.1 Краткий обзор

Дескриптор вставки splice_descriptor является образцом (шаблоном) для добавления новых полей в секцию splice_info_section. Все дескрипторы используют одинаковый синтаксис для первых шести байтов. Для добавления частной информации используется код «идентификатора». Это устраняет необходимость применения в цикле дескриптора регистрации.

Любое оборудование приема должно обрабатывать любые дескрипторы с неизвестными идентификаторами или неизвестными тегами дескриптора. В случае дескрипторов с известными идентификаторами оборудование приема должно игнорировать дескрипторы с неизвестными splice_descriptor_tag.

Дескрипторы вставки могут существовать в секциях splice_info_section для специфичных расширений команд, представленных в таблице 13.

Таблица 13 — Теги дескриптора вставки для идентификатора «CUEI»

Тег	Дескрипторы для идентификатора «CUEI»
0x00	avail_descriptor
0x01	DTMF_descriptor
0x02	segmentation_descriptor
0x03 — 0xFF	Зарезервировано для будущих splice_descriptors

7.2 Дескриптор вставки

Синтаксис дескриптора вставки splice_descriptor(), рассматриваемый ниже, должен использоваться в качестве шаблона для конкретных реализаций дескрипторов, предназначенных для секции splice_info_section. Дескрипторы вставки используются только в splice_info_section. Они не должны использоваться с синтаксисом MPEG, таким как PMT, или с синтаксисом какого-либо другого стандарта. Это позволяет использовать весь диапазон тегов дескриптора, определяя новые дескрипторы.

Кодирование дескриптора вставки splice_descriptor() осуществляется в соответствии с таблицей 14.

Таблица 14 — Кодирование дескриптора вставки

Синтаксис	Количество битов	Мнемосхема
splice_descriptor() {		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
identifier	32	uimsbf
for(i=0; i <N; i ++) {		
private_byte	8	uimsbf
}		
}		

Семантическое определение полей в splice_descriptor():

splice_descriptor_tag: Поле определяет синтаксис частных байтов, которые составляют тело этого дескриптора. Теги дескриптора устанавливаются владельцем дескриптора, зарегистрировавшего идентификатор.

descriptor_length: Поле обозначает длину в байтах дескриптора, который следует после этого поля. Длина дескриптора не превышает 256 байтов, таким образом, значение descriptor_length ограничено 254 байтами.

identifier: Поле в соответствии с ISO/IEC [1] (2.6.8, 2.6.9) для registration_descriptor() format_identifier. Должны использоваться только значения идентификатора, зарегистрированного SMPTE Registration Authority, LLC. Его использование в структуре private_command() должно быть в рамках контекста и

для идентификации только частной информации, содержащейся в этом дескрипторе. Идентификатор используется для идентификации пользователя команды. Код 0x43554549 (ASCII «CUEI») для дескрипторов, определенных в настоящем стандарте, зарегистрирован в SMPTE.

private_byte: Остаток от дескриптора выделяется для полей данных при необходимости, определяемой дескриптором.

7.3 Специальные дескрипторы вставки

7.3.1 avail_descriptor()

Дескриптор avail_descriptor является реализацией splice_descriptor и обеспечивает дополнительное расширение команды splice_insert(). Несколько копий этого дескриптора могут быть включены при использовании предусмотренного механизма цикла. Этот идентификатор предназначается для тиражирования функциональных возможностей системы тональных меток, используемой в аналоговых системах для ввода объявлений (ad). Этот дескриптор предназначается только для использования с командой splice_insert() в секции splice_info_section.

Кодирование дескриптора avail_descriptor осуществляется в соответствии с таблицей 15.

Таблица 15 — Кодирование дескриптора avail_descriptor

Синтаксис	Количество битов	Мнемосхема
avail_descriptor() {		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
identifier	32	uimsbf
provider_avail_id	32	uimsbf
}		

Определение семантики полей в avail_descriptor():

splice_descriptor_tag: Поле определяет синтаксис для частных байтов, которые составляют тело этого дескриптора. Тег splice_descriptor_tag должен иметь значение 0x00.

descriptor_length: Поле определяет длину в байтах дескриптора, который следует после этого поля. Поле descriptor_length должно иметь значение 0x08.

identifier: Поле идентифицирует владельца дескриптора. Идентификатор должен иметь значение 0x43554549 (ASCII «CUEI»).

provider_avail_id: Поле предоставляет информацию, которую приемное устройство может использовать для корректировки алгоритма обработки данных на интервале ВДР или за пределами интервала ВДР. Способ использования этого поля может быть аналогичен способу использования аналоговых меток, например, в сети, направляющей на главный узел или на его филиал команду для блокирования спортивного мероприятия.

7.3.2 DTMF_descriptor()

DTMF_descriptor() является одной из реализацией splice_descriptor. Он обеспечивает опциональное расширение команды splice_insert(), которая позволяет устройству приема генерировать аналог существующей последовательности DTMF, базирующийся на принимаемой секции splice_info_section.

Кодирование дескриптора DTMF_descriptor() осуществляется в соответствии с таблицей 16.

Таблица 16 — Кодирование дескриптора DTMF_descriptor()

Синтаксис	Количество битов	Мнемосхема
DTMF_descriptor() {		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
identifier	32	uimsbf

Окончание таблицы 16

Синтаксис	Количество битов	Мнемосхема
preroll	8	uimsbf
dtmf_count	3	uimsbf
reserved	5	bslbf
for(i=0; i <dtmf_count; i ++) {		
DTMF_char	8	uimsbf
}		
}		

Определение семантики полей в DTMF_descriptor():

splice_descriptor_tag: Поле определяет синтаксис для частных байтов, которые составляют тело этого дескриптора. Тег splice_descriptor_tag должен иметь значение 0x01.

descriptor_length: Поле дает длину в байтах дескриптора, следующего после этого поля.

identifier: Поле используется для того, чтобы идентифицировать владельца дескриптора. У идентификатора должно быть значение **0x43554549** (ASCII «CUEI»).

preroll (опережающее продвижение данных; ОПД): Поле определяет время опережающего продвижения данных, представляемое выходному сигналу аналогового устройства DTMF с дискретностью равной 0,1 с. Диапазон значений времени опережающего продвижения данных лежит в интервале от 0 до 25,5 с. Секция информации сплайсинга должна опережать время ОПД не менее чем на 2 с. Ориентировочное минимальное значение времени опережающего продвижения данных составляет 4,0 с.

dtmf_count: Поле определяет количество символов DTMF, которое должно генерировать устройство.

DTMF_char: Поле определяет значение ASCII для цифр от «0» до «9», для символов «*», «#». Устройство должно использовать эти значения, чтобы генерировать последовательность DTMF, которая будет выведена на аналоговый выход. Последовательность должна завершиться последним знаком — меткой времени ОПД.

7.3.3 segmentation_descriptor()

Дескриптор segmentation_descriptor() является реализацией splice_descriptor(). Он обеспечивает дополнительное расширение команды time_signal(). Этот дескриптор должен использоваться только с командами time_signal() и splice_null(). Сообщение time_signa() должно передаваться не менее одного раза не менее чем за 4 с перед сообщением splice_time(), чтобы позволить устройству вставки правильно помещать splice_info_section().

Устройства, которые не распознают значение этого дескриптора в любом поле, должны игнорировать сообщение и не предпринимать никаких действий.

Кодирование дескриптора segmentation_descriptor() должно выполняться в соответствии с таблицей 17.

Таблица 17 — Кодирование дескриптора segmentation_descriptor()

Синтаксис	Количество битов	Мнемосхема
segmentation_descriptor () {		
splice_descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
identifier	32	uimsbf
segmentation_event_id	32	uimsbf
segmentation_event_cancel_indicator	1	bslbf
reserved	7	bslbf
if(segmentation_event_cancel_indicator == '0') {	1	bslbf
program_segmentation_flag	1	bslbf
segmentation_duration_flag	6	bslbf

Окончание таблицы 17

Синтаксис	Количество битов	Мнемосхема
reserved		
if(program_segmentation_flag == '0') {	8	uimsbf
component_count		
for(i=0;i <component_count;i ++) {	8	uimsbf
component_tag	7	bslbf
reserved	33	uimsbf
pts_offset		
}		
}		
if(segmentation_duration_flag == '1')	40	uimsbf
segmentation_duration	8	uimsbf
segmentation_upid_type	8	uimsbf
segmentation_upid_length		
segmentation_upid()	8	uimsbf
segmentation_type_id	8	uimsbf
segment_num	8	uimsbf
segments_expected		
}		
}		

7.3.3.1 Определение семантики полей в segmentation_descriptor():

splice_descriptor_tag: Поле определяет синтаксис частных байтов, которые составляют тело этого дескриптора. Тег splice_descriptor_tag должен иметь значение 0x02.

descriptor_length: Поле обозначает длину в байтах поля дескриптора, которое следует после этого поля.

identifier: Поле используется для идентификации владельца дескриптора. У идентификатора должно быть значение 0x43554549 (ASCII «CUEI»).

segmentation_event_id: Поле является уникальным идентификатором события сегментации. В любой момент должно быть активным только одно значение segmentation_event_id. Детализированное описание синтаксиса приведено в 7.3.3.2.

segmentation_event_cancel_indicator: Значение флага, установленного в «1», указывает, что было отменено ранее отправленное событие сегментации, идентифицированное segmentation_event_id. Идентификатор segmentation_type_id не должен сопоставлять события сегментации между исходным и отмененным сообщением и сообщением с истинным segmentation_event_cancel_indicator. При отмене события сегментации segmentation_event_id может быть снова использован для идентификации контента или формирования нового сегмента.

program_segmentation_flag: Значение флага, установленного в «1», указывает, что сообщение относится к точке сегментации программы и что это режим сегментации программы, в котором все PID/компоненты программы должны быть сегментированы. Когда флаг установлен в «0», это означает, что режим является режимом сегментации компонентов, вследствие чего каждому компоненту, предназначенному для сегментации, по отдельности будет указан синтаксис.

segmentation_duration_flag: Значение флага, установленного в «1», указывает на присутствие поля segmentation_duration. Если в поле segmentation_type_id будет установлено 0x01 (идентификация контента), то этот флаг должен быть установлен в «0».

component_count: Поле определяет количество элементарных потоков PID в данном цикле (элементарные потоки PID эквивалентны компонентам).

component_tag: Поле идентифицирует элементарный поток PID, содержащий точку сегментации, определенную значением поля splice_time(). Значение поля должно соответствовать значению, используемому в дескрипторе stream_identification_descriptor() для идентификации этого элементарного потока PID.

pts_offset: Поле должно добавляться устройством к значению pts_time в сообщении time_signal(), чтобы получить намеченное время вставки. Если это поле имеет нулевое значение, тогда поле pts_time должно использоваться без смещения. Если в поле splice_time() флаг time_specified_flag равен «0» или если команда этого дескриптора не переносит поле splice_time(), то это поле должно использоваться для смещения полученного времени режима немедленной вставки.

segmentation_duration: Поле определяет продолжительность сегмента в количестве периода часов тактовой частоты 90 кГц. Оно может использоваться сплайсером в качестве индикатора времени окончания сегмента и времени появления следующего сообщения сегментации. Для конечных сообщений должна устанавливаться величина «0».

segmentation_upid_type: Поле определяет тип уникального идентификатора программ (unique program identifier; UPID) версий аудиовизуальных продуктов контента, полученного из этих продуктов или тесно связанного с аудиовизуальными продуктами, позволяющими определить возможность использования их систем поддержки. В таблице 18 представлены типы upid:

- значения кода segmentation_upid_type, определяющего тип идентификатора;
- длина в байтах segmentation_upid_type;
- условное наименование типа идентификатора;
- краткое описание типа идентификатора.

Таблица 18 — Типы и параметры идентификатора segmentation_upid_type

segmentation_upid_type (тип)	segmentation_upid_length (длина в байтах)	segmentation_upid() (имя)	Описание
0x00	0	Не используется	Egmentation_upid не нормируется и не присутствует в дескрипторе
0x01	Переменная	Определяется пользователем	Segmentation_upid не соответствует стандартной схеме именования
0x02	8	Industry Standard Commercial Identifier; ISCI	Устаревшее: используйте 8 символов типа 0x03; 4 альфа-символа затем 4 числа
0x03	12	Ad-ID	Определен группой Advertising Digital Identification, LLC. 12 символов, 4 альфа-символа (префикс идентификации компании), сопровождаемые 8 алфавитно-цифровыми символами
0x04	32	Unique Material Identifier; UMID	В соответствии со стандартом SMPTE [7]
0x05	8	International Standard Audiovisual Number; ISAN (согласно ISO [8])	Не рекомендуется: используйте тип 0x06 двоичное кодирование согласно ISO [8]
0x06	12	Версия ISAN (ядро ISAN плюс номер версии; V-ISAN)	Известный ранее как V-ISAN. В соответствии с ISO [9]
0x07	12	TID	Идентификатор Tribune Media Systems Program. 12 символов, 2 альфа-символа, затем 10 чисел
0x08	8	TI	Идентификатор Turner
0x09	Переменная	ADI	Стандарт [12]
0x0A	21	EIDR	Идентификатор Цифровых Объектов с предполагаемым префиксом 10.123/, указывающим на номер, зарегистрированный Entertainment ID Registry Association (EIDR)
0x0в-0xFF	Переменная	Зарезевирован	Зарезервирован для будущей стандартизации

Детализированные правила применения идентификатора segmentation_upid_type должны быть в соответствии со стандартом ANSI/SCTE [6] (8.3.3).

segmentation_upid_length: Поле определяет длину поля segmentation_upid() в байтах в соответствии с таблицей 18.

segmentation_upid(): Поле идентифицирует и определяет длину в соответствии с таблицей 18. Содержание этой структуры и ее длина определяются полями segmentation_upid_type и segmentation_upid_length. Примером может быть тип 0x06 для V-ISAN длиной 12 байтов. Это поле тогда содержало бы V-ISAN идентификатор для контента, к которому обращается этот дескриптор.

segmentation_type_id: Поле должно содержать одно из значений в соответствии с таблицей 19, определяющей тип сегментации. Резервируются все неиспользованные значения. Когда segmentation_type_id будет 0x01 (идентификация контента), значение segmentation_upid_type должно быть отличным от нуля.

Таблица 19 — Тип сегментации

Сообщение сегментации	Segmentation_type_id	segment_num	segments_expected
Не обозначено	0x00	0	0
Идентификация контента	0x01	0	0
Начало программы	0x10	1	1
Конец программы	0x11	1	1
Раннее завершение программы	0x12	1	1
Выключение (остановка, разрыв) программы	0x13	1	1
Восстановление программы	0x14	1	1
Плановое превышение объема программы	0x15	1	1
Неплановое превышение объема (рановер) программы	0x16	1	1
Начало главы	0x20	Не равно нулю	Не равно нулю
Конец главы	0x21	Не равно нулю	Не равно нулю
Начало рекламы провайдера	0x30	0 или не равно нулю	0 или не равно нулю
Конец рекламы провайдера	0x31	0 или не равно нулю	0 или не равно нулю
Начало рекламы дистрибутора	0x32	0 или не равно нулю	0 или не равно нулю
Конец рекламы дистрибутора	0x33	0 или не равно нулю	0 или не равно нулю
Unscheduled_event_start (начало внепланового события)	0x40	0	0
Unscheduled_event_end (конец внепланового события)	0x41	0	0

segment_num: Поле обеспечивает идентификацию короткой секции или рекламы в segmentation_upid(). Значение этого поля должно быть в соответствии с таблицей 19.

segments_expected: Поле обеспечивает подсчет ожидаемого числа отдельных сегментов (таких как главы) в передаваемом событии сегментации. Значение этого поля должно быть в соответствии с таблицей 19.

7.3.3.2 Сегментация контента — дополнительная семантика

Использование дескриптора segmentation_descriptor() должно сигнализировать о сегментах контента. Сегменты имеют логическую иерархию со следующими уровнями:

- программы (высший уровень);
- главы;
- рекламные объявления (согласно таблице 19).

Провайдер и дистрибьютор рекламных объявлений совместно используют самый низкий логический уровень, их сегменты не должны перекрываться.

В соответствии с таблицей 19 при определении семантики, установленной в этой секции, применяются следующие определения: сегментом должна быть или программа, или глава, или реклама провайдера, или реклама дистрибьютора, или незапланированное событие. В случаях употребления segmentation_descriptor() сегменты обычно образуют пары. Допустимы следующие соединения сегментов в пары:

- начало программы/конец программы (окончание Программы может быть переопределено ранним завершением программы);
- пауза программы/возобновление программы;
- начало главы/конец главы;
- начало рекламы провайдера/конец рекламы провайдера;
- начало рекламы дистрибьютора/конец рекламы дистрибьютора;
- Unscheduled_event_start (начало внепланового события)/Unscheduled_event_end (конец внепланового события).

Следующие типы сегментации segmentation_types (таблица 19) поддерживают Сегменты, но не соединяются в пары:

- запланированное превышение объема программы;
- незапланированное превышение объема программы.

Следующие segmentation_types (таблица 19) находятся за пределами контекста раздела настоящего стандарта. Они не поддерживают сегментацию контента. К этим типам относятся следующие:

- не идентифицировано (Not Indicated);
- идентификация контента (Content Identification).

Дескрипторы начала сегмента и дескрипторы конца сегмента должны соединяться в пары. Каждый конец сегмента может сопровождаться другим сегментом, начинающимся с того же самого логического уровня сегмента. Дополнительная семантика приведена в 7.3.3.3 (программы) и в 7.3.3.4 (главы) настоящего стандарта. Когда продолжительность сегмента обеспечивается и она заканчивается без просигнализированного конца сегмента, тогда, в случае необходимости, значение segmentation_event_id может быть использовано снова. Использование таких случаев окончания сегмента не рекомендуется.

Использование ассоциаций различных типов конструкций сегментации (таких как ассоциации конструкции уровня программы с конструкциями уровня главы), позволяет использовать один и тот же segmentation_upid() в связанных конструкциях. Использование этой возможности не рекомендуется.

Семантика полей в segmentation_descriptor() для сегментирования контента представлена ниже:

segmentation_event_id: После того, как было просигнализировано начало сегмента, значение segmentation_event_id становится активным. После этого это значение не должно использоваться для идентификации других событий сегментации. После того, когда будет просигнализирован конец сегмента, значение segmentation_event_id должно соответствовать значению начала сегмента segmentation_event_id, это значение тогда становится неактивным и, следовательно, может использоваться снова для возникающего нового segmentation_descriptor().

program_segmentation_flag: Должен быть установлен в «1».

segmentation_duration_flag: Если флаг установлен в «1», допустимый segmentation_duration должен быть включен в дескриптор. Если segmentation_type_id устанавливается в 0x10 (начало программы), тогда этот флаг может быть установлен в «0».

segment_num: Должен иметь ненулевые значения для глав в пределах от 1 до значения segments_expected. Для сегментов программ это значение должно быть установлено в «1». Это поле может быть дополнительно использовано для рекламных объявлений тем же самым способом, как и для глав.

segments_expected: Должен быть установлен в ненулевое значение в соответствии с количеством глав в программе (и дополнительно рекламных объявлений). Для сегментов программы это значение должно быть установлено в «1».

7.3.3.3 Программы — дополнительная семантика

В соответствии с сигнализированным сообщением программа должна начаться с дескриптора segmentation_descriptor(), содержащего значение **segmentation_type_id** 0x10 (начало программы). Программа должна использовать единственное и уникальное значение **segmentation_event_id** во всех дескрипторах, которые принадлежат этой программе. Использование segmentation_upid() является опциональным, но если это поле используется, то его значение должно быть уникально присвоено этой программе и не использоваться совместно другими программами, которые встраиваются в эту программу. Программа должна закончиться segmentation_descriptor(), содержащим **segmentation_type_id** значение 0x11 (окончание программы) или 0x12 (раннее завершение программы).

Только следующие сообщения сегментации должны иметь место между началом программы и окончанием программы (или ранним завершением программы):

- пауза (Breakaway) программы (segmentation_type_id значение 0x13);
- возобновление программы (segmentation_type_id значение 0x14);
- запланированное превышение объема программы (segmentation_type_id значение 0x15);
- незапланированное превышение объема программы (segmentation_type_id значение 0x16). Возобновление программы может иметь место только за паузой (Breakaway) программы. Программа может быть закончена в состоянии паузы (Breakaway) программы.

После паузы (Breakaway) программы может иметь место другая последовательность с началом программы и концом программы и с новыми значениями **segmentation_event_id** и **segmentation_upid()**. Вся вложенная программа или сегменты вложенной программы должны быть расположены только между паузой (Breakaway) программы и возобновлением программы. Может иметь место множество экземпляров встроенных программ.

Примечание — Сообщения переполнения объема программы являются асинхронными уведомлениями и могут иметь место в любое время между началом и окончанием программы, включая в другую встроенную активную программу.

При наличии поля segmentation_duration в поле splice_time() принимают команду time_signal, используя настоящее время или время принятого сообщения. Время продолжительности программы непрерывно увеличивает значение во время паузы программы. Значение поля segmentation_duration может быть увеличено использованием сообщениями запланированного или незапланированного переполнения. Значение поля, предоставленного в новом сообщении является обновлением полной продолжительности программы и представляет собой время, прошедшее с реального момента нового сообщения до конца сегмента. Это значение не является добавлением прошедшего времени. Если значение поля segmentation_duration будет определено, то в случае, когда значение продолжительности будет превышено, программу нужно считать завершенной.

Если в начале программы значение продолжительности не установлено, оно может быть установлено в более позднее время использованием сообщения «запланированное переполнение программы» или «незапланированное переполнение программы».

Если реальная продолжительность программы установлена в начале программы или будет установлена позже, то поле segmentation_duration может быть обнулено отправлением сообщений «запланированное переполнение программы» или «незапланированное переполнение программы» с установкой флага segmentation_duration_flag в «0».

Сообщение идентификации контента (значение segmentation_type_id 0x01) со значением **segmentation_upid()**, соответствующим в настоящий момент активной программе, может отправляться периодически, чтобы повысить устойчивость реализации. Если сообщение отправлено, оно должно соответствовать значениям **segmentation_event_id** и **segmentation_upid()**, используемым в аналогичных (родственных) сообщениях программы. Это не ограничивает возможность передачи сообщений идентификации контента, которые не соответствуют **segmentation_event_id** и **segmentation_upid()**, используемым в похожих сообщениях программы.

7.3.3.4 Главы — дополнительная семантика

Сегмент главы должен быть представлен началом главы и заканчиваться к концу главы. Для конца главы значение **segmentation_event_id** должно соответствовать значению **segmentation_event_id** для начала главы. Существующий **segmentation_upid()** должен быть одним и тем же в обоих экземплярах пары дескрипторов segmentation_descriptor().

Сегменты главы могут быть связаны с сегментами программы использованием одного и того же **segmentation_upid()** в сообщениях главы и программы.

Главы могут накладываться. Главы могут быть пронумерованы, используя поле **segment_num**. В поле **Segments_expected** должно быть указано ожидаемое количество глав. Использование ненулевых значений поля **segmentation_duration** в начале главы является опциональным.

8 Шифрование секций

8.1 Краткий обзор

Для ограничения доступа приемников к ВДР, не авторизованных для этого ВДР, сообщение splice_info_section поддерживает шифрование части секции. Этот раздел стандарта описывает различ-

ные алгоритмы шифрования, которые могут использоваться для шифрования части секции. Процедуры шифрования секции являются опциональными как реализация шифрования, или являются создателем сообщения, или производителем приемного устройства. Опциональность применения шифрования позволяет производителю поставлять «прозрачные системы», не занимаясь задачами технологии шифрования. Если в системе предусматривается шифрование, то любое приемное устройство реализует все алгоритмы, перечисленные в настоящем стандарте, что позволяет создателю таблицы информации вставки использовать любой из алгоритмов. Использование частной технологии шифрования так же является опциональным.

8.2 Шифрование фиксированным ключом

Шифрование, используемое в соответствии с настоящим стандартом, предполагает использование фиксированного ключа. Для передатчика и для приемника используется один и тот же ключ. Метод доставки ключа для передатчика и для приемника настоящий стандарт не определяет. Допускается применение для дешифрования до 256 различных ключей. Для определения ключа, который должен использоваться при дешифровании секции, используется поле cw_index. Длина фиксированного ключа определяется типом используемого алгоритма и должна быть корректной.

8.3 Алгоритмы шифрования

Настоящий стандарт предусматривает использование алгоритмов шифрования, наименования которых переносятся в поле encryption_algorithm секции splice_info_section и перечислены в таблице 20. Поле encryption_algorithm секции splice_info_section является шестиразрядным. Применяемые разновидности стандарта шифрования данных (DES) используют 64-разрядный ключ (56 битов плюс контрольная сумма) для шифрования или дешифрования блоков размером 8 байтов. В случае тройного DES используется три 64-разрядных ключа, то есть по одному ключу для каждого из трех проходов алгоритма DES. «Стандартный» тройной DES фактически использует два ключа: первый и третий ключи идентичны в соответствии с FIPS [10], [11].

Таблица 20 — Алгоритмы шифрования

Значение	Наименование алгоритма шифрования
0	Шифрование не применяется
1	Режим DES — ECB
2	Режим DES — CBC
3	Режим тройной DES EDE3 — ECB
4—31	Зарезервированы
32—63	Определяет частный пользователь

8.3.1 DES — режим ECB

Этот алгоритм использует DES (FIPS[11]) в режиме электронной кодовой книги (Electronic Code Book, ECB) или в соответствии с ГОСТ 28147 в режиме простой замены. Для этого режима шифрования блоки данных должны содержать 8 байтов. Для ввода дополнительных байтов допускается использование цикла alignment_stuffing.

8.3.2 DES — режим CBC

Этот алгоритм использует DES (FIPS[11]) в режиме сцепления блоков шифра (Cipher Block Chaining; CBC). Основной алгоритм идентичен алгоритму DES — ECB. Каждый 64-битовый блок текста (кроме первого блока) побитно складывается по модулю 2 (операция XOR) с предыдущим блоком шифрованного текста, после чего шифруется ключом DES. Первый блок побитно складывается по модулю 2 с начальным вектором. Начальный вектор должен иметь фиксированное значение, равное нулю.

Для использования этого типа шифрования зашифрованные данные должны содержать 8 байтов с полями от splice_command_type до E_CRC_32. При необходимости ввода байтов стаффинга может применяться цикл alignment_stuffing.

8.3.3 Тройной DES EDE3 — режим ECB

Этот алгоритм использует три 64-разрядных ключа, каждый ключ используется на одном проходе алгоритма ECB DES (FIPS [10]). Каждый блок данных в устройстве передачи сначала шифруется

первым ключом, дешифруется вторым ключом и затем шифруется третьим ключом. Каждый блок на стороне приема сначала дешифруется третьим ключом, шифруется вторым ключом и затем дешифруется первым ключом.

Для использования этого типа шифрования зашифрованные данные должны быть кратны 8 байтам на интервале от поля splice_command_type до поля E_CRC_32. При необходимости ввода байтов стаффинга может применяться цикл alignment_stuffing.

8.3.4 Частные алгоритмы пользователя

Настоящий стандарт разрешает использование частных алгоритмов шифрования. Стандарт не определяет способ информирования устройства приема об алгоритме шифрования при использовании частного кода любого пользователя. Настоящий стандарт не определяет координации частных значений поля encryption_algorithm, которые должны быть зарегистрированы в установленном порядке.

Библиография

[1]	ISO/IEC 13818-1:2000-12-01	Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems
[2]	SCTE 30 2006	Digital Program Insertion Splicing API
[3]	SMPTE 312M — 2001	SMPTE STANDARD for Television — Splice Points for MPEG-2 Transport Streams
[4]	ISO/IEC 13818—4: 2004	Information Technology — Generic coding of moving pictures and associated audio information — Part 4: Conformance testing
[5]	ANSI/SCTE 118-2:2006	Program-Specific Ad Insertion — Content Provider to Traffic Communication Applications Data Model
[6]	ANSI/SCTE 35:2007	Digital Program Insertion Cueing Message for Cable
[7]	SMPTE 330M-2004	SMPTE Standard for Television — Unique Material Identifier
[8]	ISO 15706:2002	Information and Documentation — International Standard Audiovisual Number (ISAN)
[9]	ISO 15706-2:200x (In Process)	Information and Documentation — International Standard Audiovisual Number (V-ISAN)
[10]	FIPS PUB 46-3, 1999 October 25	Data Encryption Standard
[11]	FIPS PUB 81, 1980 December 2	DES Modes of Operation
[12]	MD-SP-ADI2.0-AS-I03-070105	Metadata 2.0 Specifications ADI 2.0 Specification Asset Structure

УДК 621.397:681.327.8:006.354 ОКС 33.170 ОКП 65 7400

Ключевые слова: телевидение вещательное цифровое, транспортные потоки, вставка (сплайсинг), сигнализация меток, контент

Редактор *В.В. Фролов*
Технический редактор *В.Н. Прусакова*
Корректор *М.В. Бучная*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 01.09.2014. Подписано в печать 30.10.2014. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 3,27. Тираж 34 экз. Зак. 4411.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru