ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56950— 2016

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ

РАСШИРЕННАЯ СПЕЦИФИКАЦИЯ ОБЩЕГО ИНТЕРФЕЙСА В СИСТЕМАХ ОГРАНИЧЕНИЯ ДОСТУПА CI Plus™

Основные параметры

ETSI TS 103 205 V1.1.1 (2014-03)
Digital Video Broadcasting (DVB);
Extensions to the CI Plus™ Specification
(NEQ)

Издание официальное

Москва
Стандартинформ
2016

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием Ордена Трудового Красного Знамени научно-исследовательским институтом радио, Самарский филиал «Самарское отделение научно-исследовательского института радио» (филиал ФГУП НИИР — СОНИИР)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 7 июня 2016 г. № 543-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений стандарта Европейского института по стандартизации в области телекоммуникаций (ETSI) ЕТСИ ТС 103 205 V1.1.1 (2014-03) «Расширенная спецификация общего интерфейса в системах ограничения доступа CI Plus™» [ETSI TS 103 205 V1.1.1 «Digital Video Broadcasting (DVB); Extensions to the CI Plus™ Specification», NEQ]

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены 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, уведомления и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2016

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 символы . . . 1
4 Общие сведения о расширенной спецификации общего интерфейса CI Plus™ . . . 6
4.1 Введение . . . 6
4.2 Расширенные возможности приема нескольких потоков . . . 7
4.3 Доставка контента через каналы с IP . . . 7
4.4 Расширенные функции браузера CI Plus™ . . . 9
4.5 Запуск приложений CICAM . . . 9
4.6 Извлечение файла из CICAM . . . 9
4.7 Расширенная информация URI о правилах использования . . . 10
4.8 Водяные знаки и транскодирование . . . 10
5 Общие требования . . . 10
5.1 Требования обратной совместимости . . . 10
5.2 Водяные знаки и транскодирование . . . 10
5.3 Скремблирование на уровне PES . . . 11
6 Многопоточный режим работы . . . 11
6.1 Общие замечания . . . 11
6.2 Интерфейс транспортного потока и мультиплексирование локальных TS . . . 13
6.3 Выбор PID . . . 15
6.4 Ресурсы для обработки нескольких потоков . . . 16
7 Хост в режиме проигрывателя контента IP . . . 35
7.1 Общие замечания . . . 35
7.2 Режимы интерфейса транспортного потока . . . 36
7.3 Интерфейс команд . . . 36
7.4 Ресурс дескремблирования семпла . . . 37
7.5 Интерфейс транспортного потока . . . 45
8 CICAM в режиме проигрывателя контента IP . . . 57
8.1 Общие замечания . . . 57
8.2 Управление проигрывателем . . . 58
8.3 Инициализация сеанса проигрывания . . . 58
8.4 Обработка ошибок обмена . . . 59
8.5 Поддержка режима спецэффектов . . . 59
8.6 Завершение сеанса . . . 59
8.7 Ресурс CICAM в режиме проигрывателя . . . 60
9 Поиск файла CICAM . . . 72
9.1 Общие замечания . . . 72
9.2 APDU системы файлов . . . 72
9.3 APDU FileSystemAck подтверждения поддержки . . . 73
9.4 APDU FileRequest запроса файла . . . 73
9.5 APDU FileAcknowledge подтверждения файла . . . 74
9.6 Сводная информация о ресурсах системы вспомогательных файлов . . . 74
9.7 Ресурсы системы вспомогательных файлов и ресурсы приложения MMI . . . 74

10 Ресурс низкоскоростных соединений . . . 74
10.1 Общие замечания . . . 74
10.2 Информация о конфигурации IP Хоста . . . 74
10.3 Информация о потоках IP . . . 75
10.4 Многоадресный IP . . . 75
10.5 Порты источника IP . . . 75
10.6 Доставка данных от Хоста к CICAM . . . 75
10.7 Доставка данных от CICAM к Хосту . . . 75
10.8 Управление потоком IP . . . 76
10.9 APDU запроса comms_info . . . 76
10.10 APDU comms_IPconfig запроса конфигурации IP Хоста . . . 77
10.11 Параметры модифицированного Comms Cmd . . . 79
10.12 Модификация типов ресурсов низкоскоростных соединений . . . 82
11 Расширенные правила использования информации, версия 3 . . . 84
12 Приложения CICAM . . . 86
12.1 Общие замечания . . . 86
12.2 Спецификации браузера CI Plus™ с расширенными функциями . . . 86
12.3 Управление жизненным циклом приложения . . . 87
12.4 Платформа координации приложений . . . 88
12.5 Среда приложения Хоста . . . 92
13 Управление Хостом DVB . . . 93
13.1 Общие замечания . . . 93
13.2 APDU управления Хостом DVB . . . 93
14 Виртуальный канал CICAM . . . 97
14.1 Введение . . . 97
14.2 Расширенный ресурс информации приложения, версия 4 . . . 97
15 Профиль оператора . . . 98
15.1 Вводные замечания . . . 98
15.2 Параметры профиля оператора, тип 2 . . . 98
15.3 Параметры виртуального канала CICAM . . . 99
15.4 Службы, предоставленные IP . . . 102
16 Параметры интерфейса человек-машина высокого уровня . . . 104
Приложение А (рекомендуемое) Идентификация и местоположение дескрипторов . . . 105
Приложение Б (обязательное) Полный набор ресурсов . . . 106
Приложение В (обязательное) Учетная спецификация: параметры обмена в APDU . . . 116
Приложение Г (обязательное) Схемы XML . . . 118
Приложение Д (обязательное) Сигнализация приложения CICAM . . . 124
Библиография . . . 125

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ

РАСШИРЕННАЯ СПЕЦИФИКАЦИЯ ОБЩЕГО ИНТЕРФЕЙСА В СИСТЕМАХ ОГРАНИЧЕНИЯ ДОСТУПА CI PLUS™

Основные параметры

Digital Video Broadcasting. Extensions to the CI Plus™ Specification. Basic parameters

Дата введения — 2017—06—01

1 Область применения

Настоящий стандарт содержит расширенные характеристики общего интерфейса в системах ограничения доступа CI Plus™, определенного в [1]. Стандарт распространяется на оборудование защиты информации, передаваемой по каналам вещания и по каналам интернет-протокола (Internet Protocol; IP).

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 52210—2004 Телевидение вещательное цифровое. Термины и определения

ГОСТ Р 52591—2006 Система передачи данных пользователя в цифровом телевизионном формате. Основные параметры

ГОСТ Р 53528—2009 Телевидение вещательное цифровое. Требования к реализации протокола высокоскоростной передачи информации DSM-CC. Основные параметры

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 ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины, определения, сокращения и символы

3.1 В настоящем стандарте применены термины по ГОСТ Р 52210, ГОСТ Р 52591, ГОСТ Р 53528, а также следующие термины с соответствующими определениями:

3.1.1 **виртуальный канал** (Virtual Channel): Канал, формируемый из совокупности каналов Хоста и запускаемый из меню CICAM как элемент перечня каналов Хоста.

Издание официальное

3.1.2 **дескриптор** (descriptor): Ключевое слово, определяющее тип передаваемых данных.

3.1.3 **идентификатор локального транспортного потока (transport stream; TS)** (Local TS identifier; LTS): Уникальный номер, присвоенный локальному (местному) TS; замещает байт синхронизации в каждом заголовке пакета TS.

3.1.4 **интерфейс сети IP** (IP network interface): Проводной или беспроводной интерфейс интегрированного приемника-декодера (Integrated Receiver Decoder; IRD), поддерживающего базовые соединения IP.

3.1.5 **контент IP** (IP-delivered content): Аудиовидеоконтент, принятый через сетевой интерфейс IP IRD.

3.1.6 **листинг** (listing): Список, составление списка.

3.1.7 **локальный (местный) TS** (Local TS): Последовательность пакетов TS, в которой каждый пакет TS имеет идентификатор локального TS.

3.1.8 **настройка приоритетная** (foreground tune): Операция настройки в многопоточном режиме для приема службы, которая будет представляться пользователю.

3.1.9 **настройка фоновая** (background tune): Операция настройки в многопоточном режиме для приема службы, не предназначенной для представления пользователю.

3.1.10 **общий интерфейс** (Common Interface; CI): Метод обеспечения доступа к скремблированному сигналу, при котором все узлы Хоста, имеющие отношение к защите информации, устанавливаются в модуле защиты.

3.1.11 **орган** (trust authority): Объект (организация), который регламентирует соответствие и надежность реализаций CICAM и Хоста в соответствии с настоящим стандартом.

3.1.12 **платформа координации приложений** (application coordination framework): Совокупность конкретных правил CI Plus™ координации приложений вещания и приложений CICAM.

Примечание — В соответствии с 12.4 настоящего стандарта.

3.1.13 **приложение вещания** (broadcast application): Приложение, сигнализируемое в службе вещания, или приложение, запускающее приложение вещания, даже если оно само не сигнализируется в службе DVB.

3.1.14 **приложение вещания CICAM** (CICAM broadcast application): Приложение вещания, загруженное от системы вспомогательных файлов.

3.1.15 **приложение, сигнализированное вещанием** (broadcast signalled application): Приложение, сигнализированное в службе DVB, которое может переноситься во внутриполосной службе (например, с каруселью объектов DSM-CC) или во внеполосной службе [например, с помощью протокола передачи гипертекста (HyperText Transfer Protocol; HTTP) или системы вспомогательных файлов CICAM].

3.1.16 **приложение AppMMI CICAM** (CICAM AppMMI application): Приложение, использующее ресурсы приложения MMI CI Plus™ и запущенное CICAM.

3.1.17 **приложение CICAM** (CICAM application): Приложение, предоставленное CICAM с помощью системы вспомогательных файлов или использования ресурса приложения MMI.

3.1.18 **профиль пользователя** (consumer profile): Данные, представляющие интересы и предпочтения пользователя.

3.1.19 **режим ввода** (Input Mode): Режим работы интерфейса TS, в котором CICAM генерирует для Хоста новые TS.

3.1.20 **режим доставки контента IP Хосту в режиме проигрывателя** (IP delivery Host player mode): Режим обработки контента IP, в процессе которого Хост обрабатывает протоколы доставки и формат инкапсуляции контента.

3.1.21 **режим доставки контента IP CICAM в режиме проигрывателя** (IP delivery CICAM player mode): Режим обработки доставленного контента IP, в процессе которого CICAM обрабатывает протоколы и формат инкапсуляции контента.

3.1.22 **режим многопоточный** (Multi-stream Mode): Режим работы, обеспечивающий перенос нескольких TS через интерфейс TS.

3.1.23 **режим нормальный** (Normal Mode): Режим работы локального TS, переносящего обычный TS.

3.1.24 **режим обработки одиночного потока** (single-stream mode): Режим работы интерфейса TS, соответствующего [2].

3.1.25 **режим семпла** (Sample Mode): Режим работы локального TS при переносе семпла.

3.1.26 **режим спецэффектов** (trick mode): Режим, позволяющий устанавливать начало проигрывания контента, ускоренную перемотку вперед или назад, ускоренное (относительно реального времени) считывание, запрос и считывание блоков контента, например I-кадров, в пределах элемента контента.

3.1.27 **ресурс приложения MMI** (application MMI resource): Ресурс, обеспечивающий обмен файлами и данными в обоих направлениях и обеспечивающий возможность возвращать информацию о статусе от домена приложения к модулю (CICAM).

3.1.28 **сеанс проигрывателя** (player session): Интервал времени, на котором выполняется проигрывание контента.

3.1.29 **сеанс CICAM в режиме проигрывателя** (CICAM player mode session): Интервал времени, на котором CICAM выполняет проигрывание контента.

3.1.30 **семпл** (Sample): Логический сегмент данных.

Примечание — Контент семпла содержит сегмент данных, скремблированных постоянным набором параметров скремблирования.

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

3.1.32 **служба DVB (**DVB service): Служба, которая сигнализирована или объявлена в соответствии со спецификациями DVB, включая [3], [4] и OSDT, приведенный в разделе 15 и приложении А настоящего стандарта.

3.1.33 **транспортный поток** (transport stream; TS): Набор из нескольких программных потоков данных цифрового телевизионного вещания, сформированный из программных пакетов постоянной длины с коррекцией ошибок и независимым тактированием от своих источников синхронизации.

3.1.34 **трек** (Track): Последовательность связанных семплов в файле контента IP, доставляемая Хосту в режиме проигрывателя.

3.1.35 **тюнер** (tuner): IRD, имеющий функциональную возможность доставки TS, содержащего не менее одной службы DVB.

3.1.36 **формат преобразования юникода** (Unicode Transformation Format, 8-bit; UTF-8): Восьмибитный набор символов для протоколов, выходящих за рамки кодировки ASCII.

3.1.37 **Хост** (Host): IRD, включающий в себя CI Plus™ со слотом CICAM.

3.1.38 **элемент контента** (content item): Объект, который может быть приобретен как единое целое, например, файл аудио/видео, аудиопоток.

В настоящем стандарте применены следующие сокращения:

ADQ — запрос домена приложения (Application Domain Query);

AIT — таблица информации приложения (Application Information Table);

AKH — аутентификация (проверка подлинности) ключа Хоста (Authentication Key Host);

AKM — аутентификация (проверка подлинности) ключа модуля (CICAM) [Authentication Key Module (CICAM)];

APDU — модуль данных протокола приложения (Application Protocol Data Unit);

API — интерфейс прикладных программ (Application Programming Interface);

ASCII — американский 8-битный стандартный код для обмена информацией (American Standart Code for Information Interchange);

ASN.1 — абстрактная синтаксическая нотация, версия 1 (Abstract Syntax Notation One);

AV — аудио/видео (Audio-Video);

BCG — путеводитель по широкополосному контенту (Broadband Content Guide);

BER — базовые правила кодирования (Basic Encoding Rules);

BLT — таблица уровня буфера (Buffer Level Table);

bslbf — строка битов, левый бит первый (bit string, left bit first);

CA — условный доступ (Conditional Access);

CA_PMT — таблица состава программ CA (Conditional Access Program Map Table);

CAS — система условного доступа (CA System);

CASD — дескриптор CAS (Content Access Streaming Descriptor);

CAT — таблица условного доступа (Conditional Access Table);

CC — управление контентом (Content Control);

CCK — ключ управления контентом (Content Control Key);

CENC — общее скремблирование (Common Encryption);

CI — общий интерфейс (Common Interface);

CICAM — модуль CI CA (CI CA Module);

CIS — передача сообщения CI (CI Send Message);

CIV — вектор инициализации контента (Content Initialisation Vector);

DASH — динамическая адаптивная потоковая передача по HTTP (Dynamic Adaptive Streaming over HTTP);

DH — криптографический протокол Диффи — Хеллмана (Diffie Hellman);

DHCP — протокол динамического конфигурирования Хоста (Dynamic Host Configuration Protocol);

DHPH — открытый ключ DH Хоста (DH Public key Host);

DHPM — открытый ключ DH модуля CICAM (DH Public key CICAM Module);

DNS — служба доменных имен (Domain Name Service);

DRM — цифровое управление правами (Digital Rights Management);

DSD — дескриптор системы доставки (Delivery System Descriptor);

DSM-CC — система команд и управления для средств цифровой записи (Digital Storage Media — Command and Control);

DTCP — защита контента при цифровой передаче (Digital Transmission Content Protection);

DVB — цифровое телевизионное вещание (Digital Video Broadcasting);

DVB-C — стандарт цифрового телевизионного вещания по кабелю первого поколения (DVB Cable);

DVB-C2 — стандарт цифрового телевизионного вещания по кабелю второго поколения (Digital Video Broadcasting-Cable 2);

DVB-S — стандарт спутникового цифрового телевизионного вещания первого поколения (DVB Satellite);

DVB-S2 — стандарт спутникового цифрового телевизионного вещания второго поколения (Digital Video Broadcasting-Satellite 2);

DVB-T — стандарт наземного цифрового телевизионного вещания первого поколения (DVB Terrestrial);

DVB-T2 — стандарт наземного цифрового телевизионного вещания второго поколения (Digital Video Broadcasting-Terrestrial 2);

EBU — Европейский вещательный союз (European Broadcasting Union);

ECM — сообщение управления доступом (Entitlement Control Message);

EMM — сообщение разрешения доступа (Entitlement Management Message);

EPG — электронный путеводитель по программам (Electronic Programme Guide);

ES — элементарный поток MPEG-2 (MPEG-2 Elementary Stream);

ETSI — Европейский институт по стандартизации в области телекоммуникаций (European Telecommunications Standards Institute);

FEC — прямая коррекция ошибок (Forward Error Correction);

FLT — таблица сброса (очистки) (Flush Table);

HbbTV® — гибридное широкополосное телевизионное вещание (Hybrid Broadcast Broadband Television);

HD — высокая четкость (High Definition);

HDCP — система защиты широкополосного цифрового контента (High-bandwidth Digital Content Protection system);

HTTP — протокол передачи гипертекста (HyperText Transfer Protocol);

ID — идентификатор (Identifier);

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

IETF — Комитет по инженерным вопросам Интернета (Internet Engineering Task Force);

IGMP — протокол управления группами Интернета (Internet Group Management Protocol);

IP — Интернет-протокол (Internet Protocol);

IPTV — телевидение IP (IP Television);

IRD — интегрированный приемник-декодер (Integrated Receiver Decoder);

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

ISOBMFF — базовый формат медиафайла ISO (ISO Base Media File Format);

ITU-T — сектор стандартизации электросвязи МСЭ (International Telecommunications Union-Telecommunication Standardization Sector);

IV — вектор инициализации (Initialisation Vector);
KID — идентификатор ключа (key identifier);
LCN — номер логического канала (Logical Channel Number);
LSC — низкоскоростное соединение (Low Speed Communication);
LTS — локальный транспортный поток (Local Transport Stream);
MAC — управление доступом медиа (Media Access Control);
MHEG — группа экспертов по мультимедиа и гипермедиа (Multimedia and Hypermedia Experts Group);
MHP®, MHP — мультимедийная домашняя платформа (Multimedia Home Platform);
MMI — интерфейс человек-машина (Man-Machine Interface);
MPD — дескриптор представления медиа (Media Presentation Description);
MPEG — 1 Группа экспертов ISO по вопросам стандартизации, обработки и записи движущихся объектов; 2 Набор стандартов кодирования и сжатия цифрового изображения и звука (Motion Pictures Expert Group);
MSB — старший значащий бит (Most Significant Bit);
NIT — таблица сетевой информации (Network Information Table);
OSDT — онлайн SDT (Online SDT);
PAT — таблица объединения программ (Program Association Table);
PES — пакетизированный элементарный поток (Packetised Elementary Stream);
PID — идентификатор пакета (Packet Identifier);
PIN — персональный идентификационный номер (Personal Identification Number);
PMT — таблица состава программы (Program Map Table);
ROT — полномочный орган [Root of Trust (i.e. Trust Authority)];
RTP — транспортный протокол реального времени (Real Time Transport Protocol);
RTSP — протокол потоковой передачи в реальном времени (Real Time Streaming Protocol);
SAC — безопасный канал с проверкой аутентификации (Secure Authenticated Channel);
SAS — поддержка специфического приложения (Specific Application Support);
SD — стандартная четкость (Standard Definition);
SDT — таблица описания служб (Service Description Table);
SET — таблица окончания семпла (Sample End Table);
SI — информация о службах (Service Information);
SPTS — однопрограммный транспортный поток (Single Programme Transport Stream);
SRM — сообщение реабилитации системы (System Renewability Message);
SSM — режим набора субтитров (Set Subtitle Mode);
SST — таблица старта (начала) семпла (Sample start table);
tcimsbf — мнемоническое условное обозначение: целое число в дополнительном коде, старший бит следует первым (twos complement integer, most significant bit first);
TCP — протокол управления передачей (Transmission Control Protocol);
TS — транспортный поток (transport stream);
TSC — транспорт управления скремблированием (Transport Scrambling Control);
UDP — протокол дейтаграмм пользователя (User Datagram Protocol);
uimsbf — целое число без знака, сначала старший значащий бит (unsigned integer most significant bit first);
URI — информация о правилах использования (Usage Rules Information);
URL — унифицированный указатель (определитель расположения) ресурса (Uniform Resource Locator);
URN — унифицированное имя ресурса (Uniform Resource Name);
UTF-8 — формат преобразования юникода, 8-битный (Unicode Transformation Format, 8-bit);
UUID — универсальный уникальный идентификатор (Universally Unique Identifier);
VOD — видео по запросу (Video on Demand);
XML — расширяемый язык разметки (Extensible Markup Language).

3.2 В настоящем стандарте применены следующие символы:

0x — префикс шестнадцатеричного числа;
0b — префикс двоичного числа.

4 Общие сведения о расширенной спецификации общего интерфейса CI Plus™

4.1 Введение

Настоящий стандарт устанавливает параметры расширенной спецификации общего интерфейса системы условного доступа, определенного в [1].

Спецификация CI Plus™ [1] в рамках настоящего стандарта расширяется дополнением следующих функций и возможностей:

- функцией обработки нескольких локальных транспортных потоков, переносимых через интерфейс TS;
- функцией обработки контента, доставляемого по каналам IP;
- расширением функций браузера CI Plus™;
- функцией активизации приложения CICAM;
- расширением URI при обработке режима спецэффектов;
- функциями маркирования дескремблированных служб водяными знаками;
- функциями транскодирования служб.

На рисунке 1 представлена структурная схема расширенной системы CI Plus™, которая включает новые функции и возможности обработки потоков, определенные в настоящем стандарте.

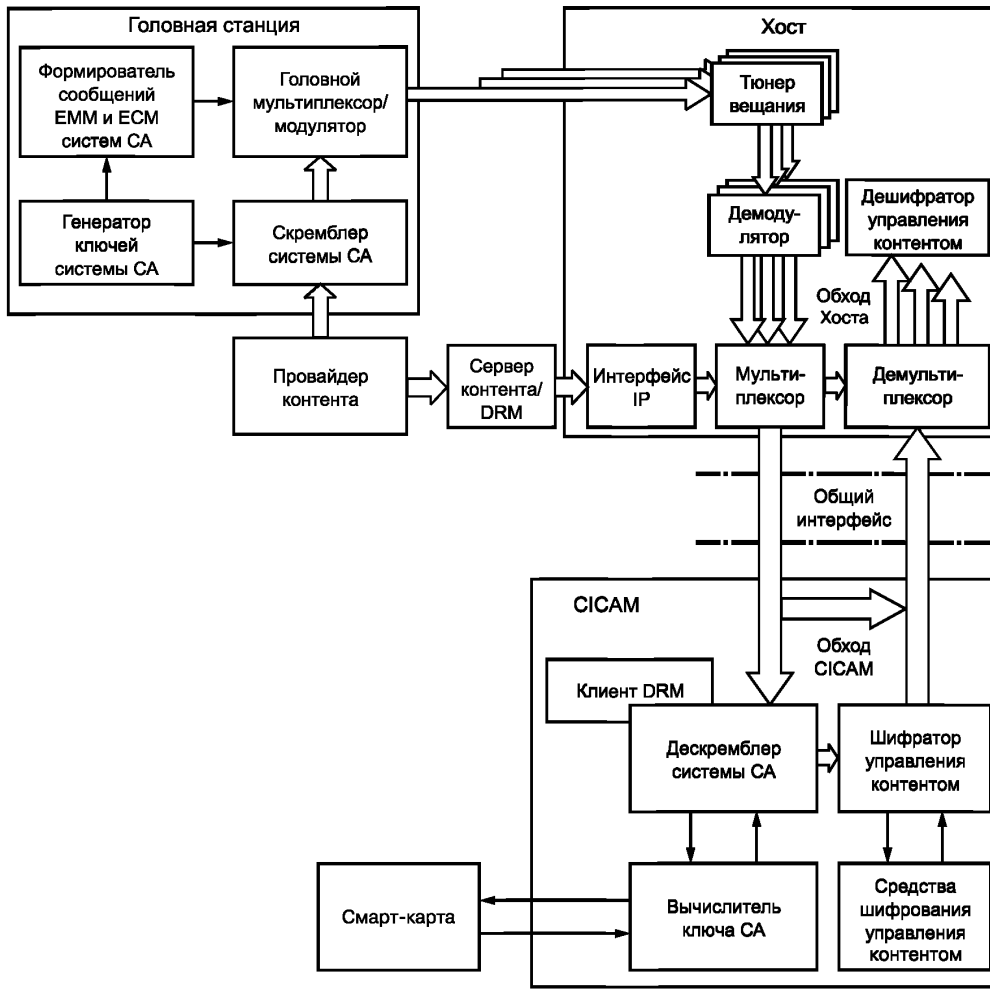


Рисунок 1 — Структурная схема расширенной системы CI Plus™

Хост при обработке нескольких потоков направляет и маршрутизирует к CICAM службы вещательной передачи, полученные от вещательных тюнеров. Хост фильтрует входящие транспортные потоки для выделения компонентов потоков, необходимых в CICAM, и мультиплексирует потоки от разных тюнеров через интерфейс транспортных потоков. Процесс обработки нескольких потоков специфицируется в соответствии с разделом 6 настоящего стандарта.

При обработке нескольких потоков Хост маршрутизирует контент, полученный от интерфейса IP, к CICAM или направляет контент непосредственно к CICAM, если Хост поддерживает маршрутизацию к CICAM только одной службы. Хост форматирует контент, доставленный через интерфейс IP, для передачи через интерфейс транспортного потока. В 4.3 настоящего стандарта содержатся вводные сведения о доставке контента средствами IP. Параметры детализированных функций доставки контента представлены в разделах 7 и 8 настоящего стандарта.

На рисунке 2 показана совокупность схем защиты контента, представляющая расширенную версию схем защиты, в соответствии с [1] (рисунок 5.1). Эта область включает схему защиты системы цифрового управления правами (Digital Rights Management; DRM) для контента, поставляемого средствами IP.

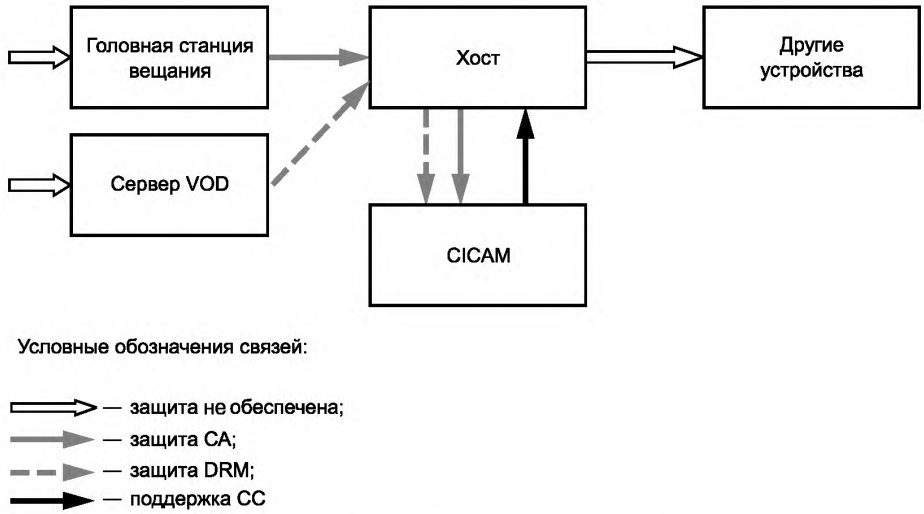


Рисунок 2 — Совокупность схем защиты контента

Настоящий стандарт расширяет спецификацию [1] (приложение H) новым идентификатором типа данных datatype_id и расширяет [1] (Приложение L) новыми ресурсами и APDU, определенными в настоящем стандарте. Другие приложения [1] в настоящем стандарте не изменены.

Раздел 5 настоящего стандарта содержит общие требования к реализациям Хоста и CICAM.

Разделы 6—16 настоящего стандарта содержат параметры новых функций, а также изменения и ограничения параметров интерфейса CI Plus™.

4.2 Расширенные возможности приема нескольких потоков

Настоящий стандарт определяет расширение для CI Plus™, которое позволяет выполнять передачу через интерфейс TS с одним CICAM различных служб от нескольких вещательных тюнеров и/или интерфейсов IP. Эти службы могут быть защищены системами CAS или DRM при доступе к службам с помощью одного CICAM. На рисунке 1 Хост содержит три вещательных тюнера, создающих несколько потоков. Обработка этих потоков должна выполняться интерфейсом TS в многопоточном режиме.

Параметры процессов в многопоточном режиме работы нормированы в разделе 6 настоящего стандарта.

4.3 Доставка контента через каналы с IP

4.3.1 Общие положения

Многие провайдеры вещательных служб и операторы сетей доставляют контент по каналам с IP (контент IP). Современные Хосты в дополнение к традиционной доставке контента средствами вещания

могут подключаться к широкополосной сети IP и имеют возможности обработки гибридного широкополосного вещания. Предыдущие версии спецификации CI предусматривали обработку контента IP через интерфейс TS, защищенного CAS. Доступ пользователя к контенту IP, защищаемому DRM, определял необходимость наличия в Хосте отдельного агента DRM. В настоящем стандарте процессы маршрутизации и дескремблирования защищенного контента IP на сетевом интерфейсе Хоста расширены при размещении в CICAM агента DRM. Это позволяет провайдерам служб и операторам сетей иметь единый CICAM, который может принимать и дескремблировать как контент вещания, так и контент IP.

Характеристики форматов контейнеров и протоколов доставки контента IP, используемых для передачи контента между сервером и Хостом, настоящий стандарт не нормирует, за исключением случаев, когда они влияют на поведение контента, передаваемого через CI. Предоставление защищенного контента IP с помощью Хоста на CICAM обеспечивается для всех форматов, применяемых до марта 2014 г. включительно.

Настоящий стандарт определяет параметры поддержки контента IP как для случая TS, так и для контента IP, инкапсулированного в формате ISOBMFF. Контент IP в формате TS не требует повторной инкапсуляции для переноса через интерфейс TS. Контейнер контента IP ISOBMFF [5] или MP4FF [6] инкапсулируется в специфичный контейнер базового формата TS для доставки IP CI Plus™. Этот контейнер применен для передачи от Хоста к CICAM и от CICAM к Хосту. При переносе контента в контейнере, отличающемся от контейнера TS, интерфейс TS работает в режиме семпла. В случае переноса в стандартных контейнерах TS он работает в нормальном режиме.

В режиме семпла метаданные DRM и информация управления, относящиеся к IP-элементам контента, перед запуском проигрывания передаются от Хоста на CICAM через интерфейс команд. Метаданные DRM, которые изменяются между семплами, переносятся с данными семпла через интерфейс TS.

В режиме семпла данные, входящие от Хоста к CICAM, как правило, хранятся в буфере до тех пор, пока они не будут готовы к дескремблированию семплов. Настоящий стандарт предусматривает два режима доставки контента IP, различающиеся уровнем взаимодействия Хоста с IP протоколами и форматами доставки. Они вводятся в 4.3.2 настоящего стандарта.

4.3.2 Режимы доставки IP

4.3.2.1 Хост в режиме проигрывателя

При работе в режиме проигрывателя Хост подключен к широкополосной сети и обрабатывает протоколы, доставляющие контент. Кроме того, Хост обрабатывает контейнеры с инкапсулированным контентом. Хост передает контент в CICAM на дескремблер DRM. CICAM возвращает дескремблированный контент, который он может скремблировать вновь с использованием системы управления контентом, определенной в спецификации Хоста CI Plus™. Последовательность обмена APDU данных между Хостом и CICAM при подготовке к дескремблированию и в процессе дескремблирования контента представлена в 7.4 настоящего стандарта.

Аналогичным образом Хост и CICAM обрабатывают контент вещания, когда для безопасной поставки CICAM предоставляет клиенту собственную защиту дескремблированного контента и может безопасно предоставить Хосту дескремблированный контент, используя систему управления контентом CI Plus™.

Параметры доставки IP Хосту в режиме проигрывателя определены в разделе 7 настоящего стандарта.

4.3.2.2 CICAM в режиме проигрывателя

При работе CICAM в режиме проигрывателя Хост подключен к широкополосной сети, но не обрабатывает протоколы UDP или TCP, с помощью которых доставляется контент, и не обрабатывает форматы инкапсуляции контента и коды, а передает полезную нагрузку пакетов IP в CICAM. CICAM деинкапсулирует контент, управляет протоколами доставки и переводит их в случае необходимости в формат, поддерживаемый Хостом. Контент возвращается из CICAM в Хост в виде SPTS, который Хост способен обрабатывать самостоятельно.

Параметры CICAM в режиме проигрывателя при доставке контента IP представлены в разделе 8 настоящего стандарта.

4.3.3 Методы использования контента IP

4.3.3.1 Общие положения

Три метода использования контента IP описаны в 4.3.3.2—4.3.3.4 настоящего стандарта.

4.3.3.2 Передача видео по запросу в потоке «вытягиванием»

При передаче видео по запросу в потоке «вытягиванием» контент от сервера передается в потоке, используя протокол для VOD, аналогичный HTTP. Хост может обеспечить проигрывание контента в режиме спецэффектов.

Этот вариант предусматривает адаптивную передачу контента в потоке, при которой сервер обеспечивает представление различных версий одного и того же контента, кодированного с различными скоростями передачи. Адаптивная передача контента в потоке сопровождается декларацией, содержащей данные для инициализации контента.

Защищенный контент дескремблируется CAS или клиентом DRM в CICAM, позволяя Хосту представить контент пользователю.

4.3.3.3 Передача видео по запросу в потоке «проталкиванием»

При передаче видео по запросу в потоке с протоколами «проталкивания» контент передается от сервера при использовании протоколов RTSP/RTP. Хост выполняет проигрывание контента в режиме спецэффектов в границах элемента контента при передаче соответствующих команд.

Защищенный контент дескремблирован CAS или клиентом DRM в CICAM, позволяя Хосту представить контент пользователю.

4.3.3.4 Использование загруженного контента

Использование загруженного контента возможно только в случае, когда Хост работает в режиме проигрывателя контента IP. Контент был загружен ранее для локального хранения и защищен скремблированием DRM. Контент считывается Хостом. Хост может обеспечить обработку контента в режиме спецэффектов.

Защищенный контент дескремблирован клиентом DRM в CICAM, что позволяет Хосту представить контент пользователю.

Поигрывание контента возможно только при наличии соединения с сервером о DRM.

4.4 Расширенные функции браузера CI Plus™

Браузер CI Plus™ с расширенными функциями выполняет поиск и извлечение контента, используя соединение IP приложения браузера CI Plus™ и используя профиль интерактивного канала MHEG профиля вещания MHEG-5 [7]. Параметры браузера с расширенными функциями определены в 12.2 настоящего стандарта.

Настоящий стандарт устанавливает обязательной поддержку браузером CI Plus™ масштабируемого видео. Параметры процесса обработки масштабируемого видео браузером CI Plus™ определены в 12.2.4 настоящего стандарта.

Расширения для CI Plus™, предусмотренные настоящим стандартом, позволяют браузеру CI Plus™ использовать гибридное широкополосное вещание и облегчают пользователю доступ к приложениям оператора, например, для работы VOD и EPG.

4.5 Запуск приложений CICAM

Реализации Хоста для поддержки вещания и онлайн интерактивных приложений включают в себя промежуточное программное обеспечение (ПО), например, HbbTV® или MHP®. Настоящий стандарт определяет механизмы CICAM, позволяющие установить возможность Хоста поддерживать системы промежуточного ПО и возможность запуска на Хосте промежуточного ПО приложений. Настоящий стандарт определяет только платформу, в рамках которой могут быть запущены из CICAM приложения для Хоста, поддерживающие среды, отличные от среды браузера CI Plus™. Параметры идентификаторов, необходимых для использования этой платформы, устанавливаются конкретной средой.

Платформа координации приложений, возникающих между приложениями CICAM и приложениями вещания, определена в разделе 12 настоящего стандарта. Настоящий стандарт определяет механизм формирования CICAM виртуального канала, который при запуске выбранного приложения CICAM добавляется к совокупности каналов Хоста. Пользователь получает доступ к виртуальному каналу с помощью обычных методов выбора канала, доступных интерфейсу пользователя. Параметры механизма формирования виртуального канала CICAM должны быть в соответствии с разделом 14 настоящего стандарта.

4.6 Извлечение файла из CICAM

Настоящий стандарт определяет механизм, с помощью которого приложение Хоста, например HbbTV® или MHP®, может получать файлы, которые хранятся в CICAM. Целью использования этого механизма является извлечение файла для приложений, работающих в Хосте для считывания активов (цифровых объектов) интерфейса пользователя (иконок, графики) CICAM, а не от сервера через соединение IP Хоста. Предполагается, что приложению заранее известны доступные активы файла

CICAM. Запрос файла или успешно выполнится, или перестанет работать, если файл недоступен. Параметры процесса поиска файла в CICAM должны быть в соответствии с разделом 9 настоящего стандарта.

4.7 Расширенная информация URI о правилах использования

Расширение параметров сигнализации URI управления контентом позволяет поддерживать ограничение операций в режиме спецэффектов, которые будут включены или отключены для соответствующих элементов управляемого контента. Это расширение параметров сигнализации применяется к управляемому контенту со статусом управления копией только одной генерации. Правила интерпретации URI замедления режима перемотки в Хосте настоящий стандарт не определяет.

Параметры режима спецэффектов, ограничивающего расширение URI, в новой версии URI управления контентом должны быть в соответствии с разделом 11 настоящего стандарта.

4.8 Водяные знаки и транскодирование

В настоящем стандарте предусмотрены приложения CICAM, которые обеспечивают добавление водяного знака на дескремблированные службы или на их компоненты и транскодирование службы или любого из ее компонентов.

Добавление цифровых водяных знаков позволяет определять источник незаконного копирования встраиванием данных в один или несколько компонентов потоков (компрессированный или некомпрессированный) в режиме реального времени.

В соответствии с настоящим стандартом CICAM создает водяные знаки на компрессированном видео на стороне получателя.

Транскодирование применяется в случае доставки IP CICAM в режиме проигрывателя, когда Хост не поддерживает кодек, используемый для доставленных компонентов службы. CICAM транскодирует каждый соответствующий поток компонентов в формат, поддерживаемый Хостом.

При доставке IP Хосту в режиме проигрывателя выполнение водяных знаков и транскодирование операций с контентом ограничивается в случаях, рассмотренных в 5.2 настоящего стандарта.

Настоящий стандарт позволяет CICAM создавать водяные знаки и транскодировать операции на контенте или службах, которые дескремблируются. Методы создания водяных знаков или форматы перекодирования стандарт не устанавливает. Нормативные положения, касающиеся создания водяных знаков и транскодирования операций, выполняемых в CICAM, должны быть в соответствии с 5.2 настоящего стандарта.

5 Общие требования

5.1 Требования обратной совместимости

Хосты, соответствующие настоящему стандарту, должны работать с CICAM, удовлетворяющим требованиям предыдущих версий CI Plus™, т.е. [1], и с CICAM, соответствующим требованиям общего интерфейса и требованиям технологий DVB [2], [8].

Конкретные вопросы, касающиеся работы с несколькими потоками нескольких CICAM, рассматриваются в 6.2.7 настоящего стандарта.

В каждом случае доступные функции должны обеспечиваться при использовании взаимных возможностей CICAM и Хоста.

5.2 Водяные знаки и транскодирование

Как отмечалось в 4.8 настоящего стандарта, интерфейсы CICAM CI Plus™, выполняющие дескремблирование защищенных служб или компонентов потока, могут выполнять добавление водяных знаков и транскодирование компонентов потоков.

Независимо от функции обработки контента, выполняемой CICAM, для Хоста должен предоставляться совместимый с ним TS.

Водяные знаки могут быть добавлены только к контенту или службам, которые дескремблированы или обработаны CICAM. Входящие пакеты, не скремблированные в CICAM, должны быть возвращены на Хост в неизменном виде. Водяные знаки не должны добавляться к контенту или службам, которые получены не скремблированными.

Операции транскодирования в формате ISOBMFF при доставке IP контента сопровождаются изменением размера семпла между входом и выходом CICAM, поэтому они могут выполняться CICAM только при работе в режиме проигрывателя.

Операции транскодирования приведут к изменению скорости потока битов, возвращаемых к Хосту, относительно скорости входного потока в CICAM в режиме проигрывателя, следовательно, тактовая частота выходного потока может существенно отличаться от тактовой частоты входного потока. Транскодирование операций может привести к увеличению задержки потоков, обрабатываемых CICAM.

При добавлении водяных знаков или при перекодировании CICAM может вставлять новые пакеты TS, удалять часть входных пакетов и/или изменять контент входных пакетов TS.

5.3 Скремблирование на уровне PES

Спецификация [1] предусматривает защиту контента, поставляемого скремблированным на уровне PES, и выполнение повторного скремблирования CICAM на уровне TS при доставке на Хост. Настоящий стандарт не допускает повторного скремблирования в CICAM контента на уровне TS. Это объясняется повышением уязвимости системы безопасности при повторном скремблировании.

6 Многопоточный режим работы

6.1 Общие замечания

В этом разделе определяются параметры интерфейса TS в составе Хоста, обеспечивающего обработку нескольких потоков в многопоточном режиме.

На рисунке 3 в качестве примера показана схема расширенной архитектуры CI Plus™, обеспечивающей поддержку многопоточного режима реализации Хоста с тремя тюнерами вещания и с возможностью доставки IP.

В контексте настоящего стандарта тюнер обеспечивает доставку TS MPEG-2 на физическом уровне наземных DVB, спутниковых и кабельных каналов и интерфейса сетей IP. При использования тюнера IP он может принимать обычные TS, защищенные CA. Контент может быть доставлен в формате ISOBMFF или в активах MPEG DASH. Общие вопросы процесса обработки IP контента CI Plus™ кратко описаны в 4.3 настоящего стандарта. Параметры обработки IP контента установлены в разделе 7 (Хост в режиме проигрывателя) и в разделе 8 (CICAM в режиме проигрывателя) настоящего стандарта.

В расширенной архитектуре каждая служба для дескремблирования в CICAM извлекается из поступающего TS при выборе PID этой службы с последующим формированием локального TS. Локальные TS мультиплексируются перед отправкой в CICAM, который дескремблирует каждую службу независимо. Когда мультиплексированные дескремблированные локальные TS возвращаются к Хосту, они демультиплексируются на отдельные SPTS, которые Хост может декодировать. Совокупность потоков, совместимых с Хостом, должна обрабатываться в соответствии с [2] до тех пор, пока CICAM не завершит инициализацию многопоточного ресурса, определенного в 6.4.2 настоящего стандарта, передав APDU CICAM_multistream_capability.

После создания сеанса многопоточного ресурса CICAM должен передать APDU CICAM_ multistream_capability и ожидать приема одного или нескольких локальных TS в многопоточном режиме.

Настоящий стандарт предусматривает расширение ряда ресурсов CI [2] и [1] добавлением возможностей обработки многопоточного ресурса. Многопоточный Хост должен предлагать CICAM как многопоточные типы ресурса, так и стандартные типы ресурса единого потока. Это позволяет CI V1.3 Plus™ и более ранним версиям, совместимым с CICAM, нормально функционировать в режиме одиночного потока.

При поддержке Хостом многопоточного ресурса многопоточный CICAM может запросить сеанс или к ресурсам многопоточного типа, или к ресурсам типа одиночного потока. Если будет открыт какой-либо ресурс многопоточного типа, то CICAM должен продолжать работать в многопоточном режиме и должен запрашивать сеансы только к ресурсу многопоточного типа до тех пор, пока не будут закрыты все ресурсы многопоточного типа.

Если CICAM затребует ресурсы одиночного потока, то Хост должен работать в режиме одиночного потока в соответствии с [2].

Если CICAM затребует ресурсы многопоточного типа, то Хост, совместимый с многопоточным режимом, должен работать в многопоточном режиме. Работая в многопоточном режиме, Хост может отправить CICAM одиночный локальный TS.

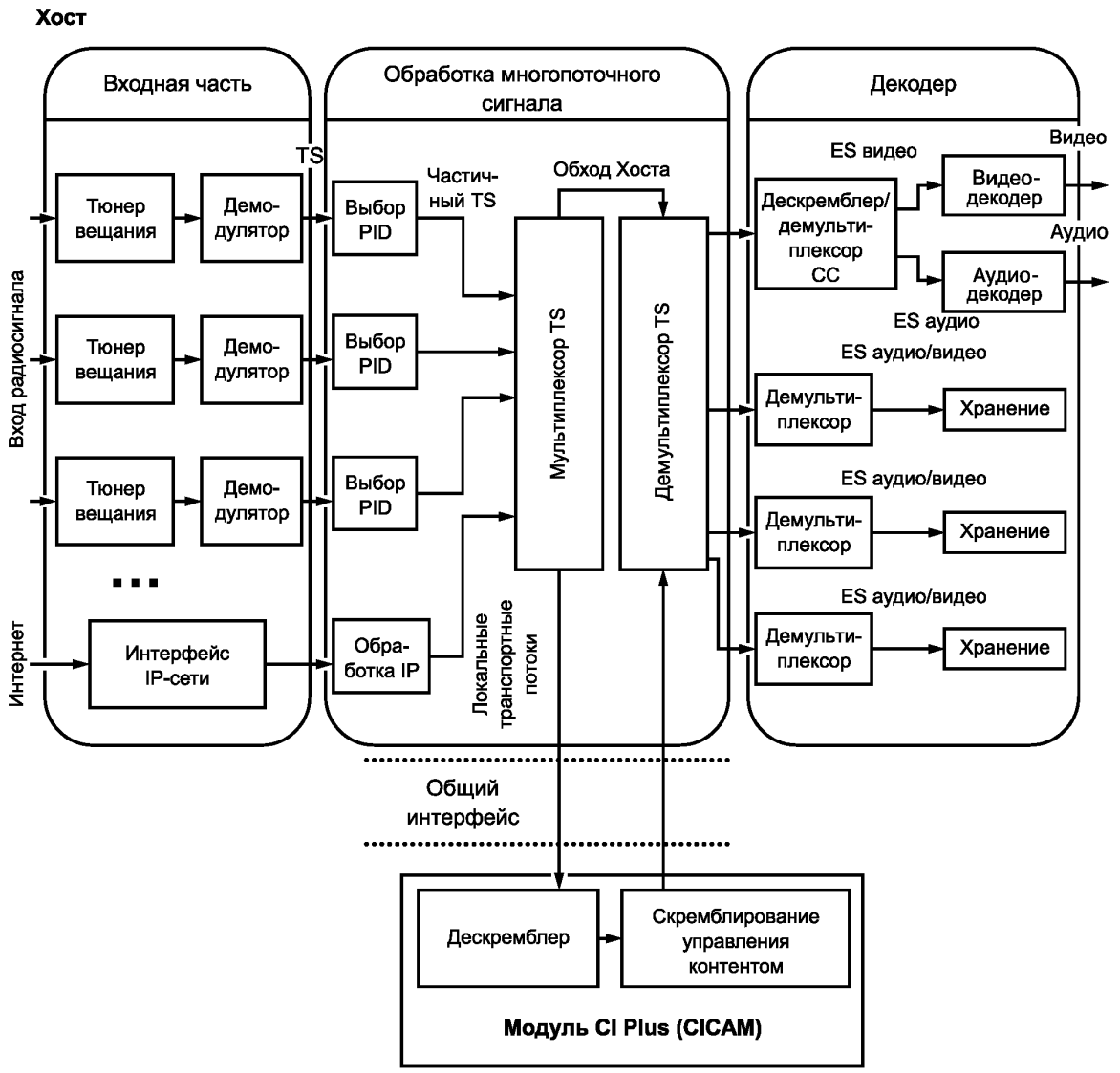


Рисунок 3 — Схема расширенной архитектуры CI Plus™, обеспечивающей поддержку многопоточного режима реализации Хоста с тремя тюнерами вещания и с возможностью доставки IP

В многопоточном режиме только одна служба, выбранная для дескремблирования, должна переноситься в каждом локальном TS. Если две службы, исходящие из одного и того же физического тюнера, выбраны для дескремблирования, то для каждой службы должен формироваться отдельный локальный TS.

Количество служб, которые могут быть реализованы через интерфейс TS в многопоточном режиме, ограничивает только общая пропускная способность интерфейса TS (96 Мбит/с) и возможности Хоста и CICAM. Максимальная скорость передачи службы вещания, пропускная способность, выделяемая для каждой службы в многопоточном режиме, и критерии возможности мультиплексирования Хостом дополнительных потоков определяются при оценке параметров реализации Хоста и характеристик доступных сетей вещания.

Дополнительные правила для работы в многопоточном режиме, когда в многопоточном Хосте устанавливаются многопоточный CICAM и однопоточный CICAM, определены в 6.2.7 настоящего стандарта.

Спецификации многопоточного ресурса представлены в 6.4 настоящего стандарта.

При необходимости использования ресурсов многопоточности типы ресурсов, указанные в [2] и [1], были обновлены с добавлением параметра LTS_id, идентифицирующего локальные TS, и определены в 6.4 настоящего стандарта.

Состав ресурса (идентификатор ресурса 00 90 00 41), необходимого для поддержки многопоточной функциональности, представлен в приложении Б настоящего стандарта.

6.2 Интерфейс транспортного потока и мультиплексирование локальных TS

6.2.1 Идентификатор локального TS

Каждому локальному TS Хост должен присваивать уникальный идентификатор локального TS (LTS_id). LTS_id заменяет фиксированное значение 0x47 в поле синхронизации байтов заголовка пакета TS каждого пакета TS, передаваемого от Хоста на CICAM. Хост устанавливает поле синхронизации байтов в заголовке TS пакетов LTS_id каждого соответствующего пакета TS, отправленного на CICAM. Значение LTS_id должно быть уникальным для каждого локального TS и не должно изменяться при выборе локального TS для дескремблирования и для передачи по интерфейсу TS. Рекомендуется, чтобы значения, присвоенные LTS_id, начинались от 0x47, т.е. LTS_id первого локального TS устанавливается на 0x47, LTS_id второго локального TS устанавливается на 0x48 и т. д.

Так как байт синхронизации локального TS может отличаться от фиксированного стандартного значения 0x47, это значение байта не должно использоваться для обнаружения первого байта каждого пакета транспортного потока.

CICAM не должен вносить изменения в поля LTS_id входящих локальных TS.

LTS_id используется демультиплексорами Хоста и CICAM для идентификации пакетов TS, связанных с каждым локальным TS. Хост может регенерировать каждый локальный TS из совместимого TS, заменяя LTS_id синхронизирующим байтом фиксированного стандартного значения 0x47. Хост не должен выполнять регенерацию TS до выполнения операций декодирования или хранения, если результирующий TS не будет использоваться устройствами или приложениями, требующими полностью совместимого TS.

Если интерфейс TS эксплуатируется в режиме одиночного потока, то LTS_id должен иметь значение 0x47.

6.2.2 Мультиплексирование вещательного контента и контента IP

Обмен контентом IP между Хостом и CICAM должен выполняться в формате TS. В режиме одиночного потока контент IP через интерфейс TS переносится как одиночный TS. В многопоточном режиме контент IP поставляется или как одиночный локальный TS, или как мультиплексированный с локальными TS вещания или потоками IP.

6.2.3 Параметры задержки мультиплексированных пакетов транспортного потока

Требования к параметрам задержки пакетов, вносимых CICAM для отдельных TS, проходящих через CICAM в Хост и из Хоста в CICAM, устанавливает [2]. В таблице 1 определены требования к порядку следования пакетов TS, к задержке пакетов TS и к изменениям задержки байтов TS на входе CICAM для различных вариантов использования. Первый вариант использования в случае вещания расширяет требования, содержащиеся в [2] для многопоточного режима. Следующие два варианта использования относятся к режиму доставки IP Хосту в режиме проигрывателя, указанному в разделе 7 настоящего стандарта, и к режиму IP доставки CICAM в режиме проигрывателя, указанному в разделе 8 настоящего стандарта. Четвертый вариант использования применяется в случае переноса через интерфейс TS контента вещания и контента IP.

Таблица 1 — Требования к порядку следования пакетов TS, к задержке пакетов TS и к изменениям задержки байтов TS на входе CICAM для различных вариантов использования

Вариант использования	Порядок следования пакетов TS	Задержка пакетов TS	Изменения задержки байтов TS на входе CICAM
Только вещание	CICAM должен поддерживать порядок пакетов с одинаковыми LTS_id	При обработке входных транспортных пакетов CICAM должен обеспечивать постоянную задержку и требования в соответствии с [2] (5.4.2)	CICAM должен обеспечивать выполнение требований к максимальным изменениям задержки в соответствии с [2] (5.4.2)

Окончание таблицы 1

Вариант использования	Порядок следования пакетов TS	Задержка пакетов TS	Изменения задержки байтов TS на входе CICAM
Только в режиме IP-доставки Хосту в режиме проигрывателя	CICAM не должен поддерживать порядок пакетов с одинаковыми LTS_id. Порядок следует поддерживать для пакетов с одинаковыми PID	Допускается несоответствие CICAM требованию постоянной задержки [2] (5.4.2) при обработке входных IP-пакетов	Допускается несоответствие CICAM требованиям к максимальным изменениям задержки [2] (5.4.2)
Только в режиме IP-доставки CICAM в режиме проигрывателя	Не применяется, так как CICAM создает новый TS	Не применяется, так как CICAM создает новый TS	Не применяется, так как CICAM создает новый TS
Вещание и IP-доставка Хосту в режиме проигрывателя	CICAM должен поддерживать порядок пакетов вещания с одинаковыми LTS_id. CICAM не должен поддерживать порядок пакетов с одинаковыми LTS_id в режиме доставки IP Хосту в режиме проигрывателя. Порядок следует поддерживать для пакетов с одинаковыми PID	Допускается несоответствие CICAM требованию постоянной задержки [2] (5.4.2) при обработке входных транспортных пакетов вещания. Допускается несоответствие CICAM требованию постоянной задержки [2] (5.4.2) при обработке входных IP-пакетов	Допускается несоответствие CICAM требованиям к максимальным изменениям задержки [2] (5.4.2)
Вещание и IP-доставка CICAM в режиме проигрывателя	К CICAM не предъявляется требование поддержки порядка пакетов вещания с одинаковыми LTS_id. Не предъявляется к IP контенту, так как CICAM создает новый TS	Допускается несоответствие CICAM требованию постоянной задержки [2] (5.4.2) при обработке входных транспортных пакетов вещания. Не предъявляется контенту при доставке IP, так как CICAM создает новый TS	Допускается несоответствие CICAM требованиям к максимальным изменениям задержки [2] (5.4.2). Не предъявляется к контенту при доставке IP, так как CICAM создает новый TS

6.2.4 Правила применения шифра скремблирования и ключа управления контентом

Хост и CICAM должны применять одинаковые правила выбора шифра скремблирования в соответствии с [1] (таблица 5.6). Такой же шифр должен быть использован для всех локальных TS.

6.2.5 Услуга игнорирования, предоставляемая Хостом

В соответствии с [1] Хост предоставляет услугу игнорирования службы, позволяющую предотвратить отображение контента в случае использования пользователем CICAM, не совместимого с интерфейсом CI Plus™.

По требованию оператора Хост не должен предоставлять игнорируемую службу в CICAM для дескремблирования. Услуга игнорирования службы, предоставляемая Хостом, активизируется в соответствии с [1] (раздел 10). В многопоточном режиме Хост не должен выполнять мультиплексирование локальных TS, содержащих игнорируемые службы в многопоточном TS, отправляемом на CICAM, т. е. услуга игнорирования применяется только к той службе, о которой сообщил оператор службы.

Услуга игнорирования, выполняемая Хостом, не распространяется на виды доставки IP, описанные в разделах 7 и 8 настоящего стандарта.

6.2.6 Тактовая частота транспортного потока

Вследствие того, что тактовая частота мультиплексируемых локальных TS может постоянно изменяться, может изменяться и скорость передачи по интерфейсу TS, поэтому Хост и CICAM должны функционировать в условиях переменных тактовых частот входного и выходного сигналов MPEG.

6.2.7 Обработка нескольких транспортных потоков несколькими CICAM

В настоящем стандарте многопоточный Хост должен работать с несколькими CICAM, соответствующими предыдущим версиям спецификации CI Plus™ [2]. В случае, когда в многопоточный Хост устанавливаются несколько CICAM, допускаются следующие варианты использования многопоточного Хоста:

- при работе Хоста только с многопоточными CICAM допускается обработка Хостом мультиплекса локальных TS в гирлянде последовательно соединенных CICAM;

- при работе Хоста только с однопоточными CICAM допускается работа Хоста в режиме одиночного потока при работе одиночных TS в гирлянде последовательно соединенных однопоточных CICAM;
- при работе Хоста с многопоточными и с однопоточными CICAM допускается работа Хоста:
  - в многопоточном режиме при обработке мультиплекса локальных TS в гирлянде последовательно соединенных многопоточных CICAM с обходом однопоточных CICAM;
  - в однопоточном режиме при обработке одиночных TS в гирлянде последовательно соединенных CICAM.

6.3 Выбор PID

6.3.1 Общие замечания

С целью уменьшения необходимой полосы пропускания интерфейса TS Хост должен выбирать PID только тех служб, которые будут дескремблироваться с формированием новых частичных TS, при отбрасывании остальных входящих PID служб.

Для каждой выбранной службы Хост должен поддерживать список выбранных PID. Первоначальное содержание этого списка определяется Хостом и включает в себя минимальный набор PID «по умолчанию», как указано в 6.3.2 или 6.3.3 настоящего стандарта, в зависимости от обстоятельств. Список может также включать в себя и любые другие PID, которые Хост может выбрать для своих собственных потребностей.

Список выбранных PID может обновляться в следующих случаях:

- пользователь инициирует изменение в выбранном элементарном потоке, например, изменением активного аудиотрека;
- CICAM инициирует изменение списка для добавления или удаления PID. Запрос состоит из списка PID, перечисленных в порядке уменьшения очередности;
- любое иное событие, требующее изменения Хостом списка выбранных PID.

6.3.2 Выбор Хостом PID по умолчанию

После выбора службы по умолчанию Хост должен отобрать следующие PID и направить соответствующие пакеты TS через интерфейс TS:

- ES (PID Elementary_PID), объявленные в соответствующем APDU ca_pmt;
- CA_PID, объявленные в CA_descriptor, присутствующие в соответствующем APDU ca_pmt;
- PID PMT, содержащую выбранную службу;
- EIT PID в соответствии с [3];
- SDT PID в соответствии с [3].

Если для дескремблирования выбирается не менее двух служб от одного тюнера, то PID SI должны быть продублированы в каждом из частичных TS, порождаемых для каждой службы.

Хост в частичных TS может включать дополнительные PID.

6.3.3 Выбор PID частоты настройки тюнера по умолчанию

В частном случае, когда CICAM выдает частоту настройки при помощи APDU tune_broadcast_req без ссылки на конкретную службу, устанавливая в поле APDU service_id значение 0x0000, тогда выбранные по умолчанию значения в списке PID должны содержать:

- PID PAT в соответствии с [9];
- PID CAT в соответствии с [9];
- PID EIT в соответствии с [3];
- PID SDT в соответствии с [3];
- список PID NIT в соответствии с [3].

В локальный TS Хост может включать дополнительные PID.

6.3.4 Приоритет выбора PID

CICAM предоставляет список PID, которые он хочет получить для выбранной службы (в порядке убывания приоритета). Наиболее приоритетными должны быть PID, необходимые для дескремблирования. Это атрибут, который может быть установлен для каждого запрашиваемого PID в APDU PID_ select_req. Хост должен выбирать PID в соответствии с приоритетом списка PID. Если Хост не может включать в себя PID, которые CICAM обозначает приоритетными для дескремблирования (в APDU PID_ select_req), то CICAM может неправильно дескремблировать службу.

6.3.5 Инициация обновления CICAM

CICAM может потребовать обновления списка PID, выбранного Хостом. Для изменения списка PID по запросу CICAM должны использоваться APDU PID_select_req. Этот список может включать в

себя PID, на которые нет ссылок в PMT выбранной службы. Хост должен ответить на такой запрос передачей APDU PID_select_reply. Совокупность PID ES для служб, перечисленных в ca_pmt, не может быть изменена с помощью CICAM.

Результирующий список PID содержит следующие группы:

- PID ES (Elementary_PID), объявленные в последнем появлении APDU ca_pmt для службы;
- PID, объявленные CICAM в APDU PID_select_req APDU;
- любые дополнительные PID, выбранные Хостом для собственных нужд.

Если Хост не может выбрать все вышеуказанные PID, то он должен выбрать PID в следующем порядке:

- PID ES (PID Elementary_PID), декларированные в последнем событии ca_pmt, связанном со службой, и поэтому имеющие высший приоритет;
- PID, запрошенные CICAM по списку в порядке очередности, и дополнительные PID, выбранные Хостом для собственных нужд. Выбор приоритетов собственного набора необходимых PID из перечня дополнительных PID, выбранных CICAM, определяется реализацией Хоста.

6.3.6 Изменение выбранных ES

Всякий раз когда Хост посылает обновление APDU ca_pmt для выбранной службы (ca_pmt_list_management = 0x05), Хост должен вернуться к набору PID «по умолчанию», как определено в 6.3.2 настоящего стандарта.

6.3.7 Корректировка списка PID по инициативе Хоста

Хост может корректировать список PID с добавлением или удалением PID в соответствии с собственными потребностями. В случае изменения списка PID, запрошенных CICAM, Хост должен информировать об этом CICAM отправлением APDU PID_select_reply.

PID, запрошенные CICAM, в том числе и не критические для дескремблирования, должны быть предоставлены и должны оставаться в локальном TS в соответствии с возможностями Хоста.

Хост может повторно выбрать один из PID, запрошенных CICAM, и должен соответственно информировать CICAM об этом, отправив APDU PID_select_reply. Хост не должен информировать CICAM, если он добавляет или удаляет PID, которые не были в списке PID, запрошенных CICAM.

Удаление или повторный выбор PID выполняются Хостом в соответствии с приоритетом, установленным CICAM.

6.3.8 Службы, выбираемые из одного транспортного потока

Если из TS вещательной передачи выбираются службы с PID, которые являются общими для нескольких служб, то пакеты TS, соответствующие общему PID, должны быть дублированы в локальных TS, формируемых для этих служб.

6.4 Ресурсы для обработки нескольких потоков

6.4.1 Общие замечания

Ресурсы обработки нескольких потоков являются модифицированными видами следующих ресурсов управления контентом:

- управления Хостом;
- поддержки системы условного доступа;
- MMI;
- приложения MMI.

Ресурсы обработки нескольких потоков специфицированы в 6.4.2 настоящего стандарта.

Перечень идентификаторов типов данных многопоточной функциональности должен быть в соответствии с приложением Б.

6.4.2 Многопоточный ресурс

6.4.2.1 Общие замечания

Многопоточный ресурс содержит APDU, показывающие возможности для CICAM, связанные с многопоточностью, и APDU для управления выбором PID в локальном TS. Ресурс многопоточности реализуют Хосты и CICAM. Параметры APDU многопоточного ресурса должны быть в соответствии с таблицей 2.

6.4.2.2 APDU от CICAM к Хосту о возможностях работы CICAM в многопоточном режиме

При открытом сеансе многопоточного ресурса CICAM должен направить Хосту APDU CICAM_multistream_capability с сообщением о максимальном количестве TS и ES, которые CICAM способен дескремблировать одновременно.

Учитывая возможности CICAM, Хост должен ограничить количество локальных TS, которые он одновременно мультиплексирует, и количество ES, которые запрашивает CICAM для дескремблирования.

Синтаксис APDU CICAM_multistream_capability должен быть в соответствии с таблицей 3.

Таблица 2 — Параметры APDU многопоточного ресурса

Многопоточный ресурс				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00 90 00 41	144	1	1	CICAM_multistream_capability	9F 92 00	←	
				PID_select_req	9F 92 01	←	
				PID_select_reply	9F 92 02	→	

Таблица 3 — Синтаксис APDU CICAM_multistream_capability

Синтаксис	Количество битов	Мнемоника
CICAM_multistream_capability () {		
CICAM_multistream_capability_tag	24	uimsbf
length_field ()		
max_local_TS	8	uimsbf
max_descramblers	16	uimsbf
}		

Семантика полей APDU CICAM_multistream_capability:

- **CICAM_multistream_capability_tag**: поле 24 бита является тегом со значением 0x9F9200;
- **length_field**: поле указывает значение длины полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **max_local_TS**: поле 8 битов указывает максимальное количество локальных TS, которые CICAM может получить одновременно;
- **max_descramblers**: поле 16 битов указывает общее количество дескремблеров, которые CICAM может предоставить одновременно для всех локальных TS.

6.4.2.3 APDU от CICAM к Хосту запроса выбора PID

CICAM посылает Хосту APDU PID_select_req с запросом на конкретный набор PID для включения в принятый локальный TS.

CICAM должен предоставить список PID в порядке приоритета списка. Каждый PID в списке должен сопровождаться флагом critical_for_descrambling_flag, который CICAM устанавливает для PID, необходимых для дескремблирования контента. Список PID не должен содержать PID с critical_for_descrambling_flag, установленным в 0b1 после первого PID, который имеет critical_for_descrambling_flag, установленный в 0b0.

Если Хост не может включить PID с установленным флагом critical_for_descrambling_flag, то CICAM не может корректно дескремблировать службу.

В любой момент времени Хост может отменить или повторно выбрать один из PID, выбранных CICAM ранее. Об этом случае Хост должен информировать CICAM, отправив APDU PID_select_reply об обновленном состоянии тех PID, которые не были выбраны или были выбраны повторно.

В случае отмены Хостом PID он должен отменять PID с самым низким приоритетом из списка PID, представленного CICAM первоначально при отборе PID.

При повторном выборе Хостом PID из списка ранее выбранных он должен выбирать PID с высшим приоритетом.

Если службы для дескремблирования выбираются от одного и того же тюнера и в списке PID, выбранных CICAM, есть общие PID, то транспортные пакеты, соответствующие общим PID, должны быть продублированы в каждом из локальных TS, создаваемых для каждой службы.

Синтаксис APDU PID_select_req должен быть в соответствии с таблицей 4.

Таблица 4 — Синтаксис APDU PID_select_req

Синтаксис	Количество битов	Мнемоника
PID_select_req () {		
PID_select_req_tag	24	uimsbf
length_field ()		
LTS_id	8	uimsbf
num_PID	8	uimsbf
for (i = 0; i < num_PID; i++) {		
reserved	2	bslbf
critical_for_descrambling_flag	1	bslbf
PID	13	uimsbf
}		
}		

Семантика полей APDU PID_select_req:

- **PID_select_req_tag**: поле 24 бита является тегом со значением 0x9F9201;
- **length_field**: длина полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **num_PID**: поле 8 битов указывает количество PID, содержащихся в передаваемом цикле;
- **critical_for_descrambling_flag**: при флаге, установленном на 0b1, связанный с ним PID имеет решающее значение для дескремблирования. При установке на 0b0 связанный с ним PID не имеет решающего значения для дескремблирования;
- **PID**: поле 13 битов содержит запрашиваемое значение PID. Хост должен игнорировать любые запросы при значении PID 0x1FFF.

6.4.2.4 APDU ответа Хоста на запрос набора PID

Когда Хост получает запрос на PID от CICAM, он должен подтвердить способность обеспечивать выделение запрошенных PID, используя APDU PID_select_reply. Хост должен указать статус для всех PID, которые были запрошены для выбора в предыдущем APDU PID_select_req, установив соответствующим образом флаг PID_selected_flag.

CICAM будет ожидать приема APDU PID_select_reply до выдачи следующего APDU PID_select_req для того же LTS_id.

Всякий раз когда Хост отменяет выбор PID или выбирает PID повторно из числа запрошенных CICAM, Хост должен сообщить CICAM об изменениях и послать APDU PID_select_reply. APDU PID_select_reply указывает статус всех PID, которые присутствовали в предыдущем APDU PID_select_req.

Синтаксис APDU PID_select_reply должен быть в соответствии с таблицей 5.

Таблица 5 — Синтаксис APDU PID_select_reply

Синтаксис	Количество битов	Мнемоника
PID_select_ reply () {		
PID_select_ reply _tag	24	uimsbf
length_field ()		
LTS_id	8	uimsbf
reserved	7	uimsbf
PID_selection_flag	1	uimsbf

Окончание таблицы 5

Синтаксис	Количество битов	Мнемоника
num_PID	8	uimsbf
for (i=0; i < num_PID; i++)		
reserved	2	bslbf
PID_selected_flag	1	uimsbf
PID	13	uimsbf
}		
}		

Семантика полей APDU PID_select_reply:

- **PID_select_reply_tag**: поле 24 бита является тегом со значением 0x9F9202;
- **length_field**: поле содержит длину полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **PID_selection_flag**: флаг указывает статус PID, он должен быть установлен в 0b0, если весь TS передается как локальный TS, а в поле num_PID должно быть установлено 0x0. Если будет применена селекция PID, то флаг должен быть установлен в 0b1 и Хост должен информировать CICAM о выбранных PID, используя поле num_PID и список выбранных PID.
- **num_PID**: поле 8 битов указывает количество PID, содержащихся в передаваемом цикле;
- **PID_selected_flag**: статус соответствующего запрашиваемого PID. PID, который выбран успешно, должен иметь этот флаг, установленный в 0b1. Если PID не может быть выбран Хостом, то его значение должно быть 0b0.
- **PID**: поле 13 битов содержит PID, для которого установлен флаг PID_selected_flag.

6.4.3 Ресурсы управления контентом

6.4.3.1 Общие замечания

Поддержка приема нескольких потоков должна обеспечиваться дополнением контента ресурса управления новым типом ресурсов (0x008C1041).

Контент ресурса управления в обобщенной форме должен быть в соответствии с таблицей 6.

Таблица 6 — Контент ресурса управления в обобщенной форме

Ресурс управления контентом				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00 8C 10 41	140	65	1	cc_open_req	9F 92 01	←	
				cc_open_cnf	9F90 02	→	
				cc_data_req	9F 9003	←	
				cc_data_cnf	9F 90 04	→	
				cc_sync_req	9F 90 05	←	
				cc_sync_cnf	9F 90 06	→	
				cc_sac_data_req (Примечание 1)	9F 90 07	← →	
				cc_sac_data_cnf (Примечание 1)	9F 90 08	← →	
				cc_sac_sync_req	9F 90 09	←	
				cc_sac_sync_cnf	9F 90 10	→	

Окончание таблицы 6

Ресурс управления контентом				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00 8C 10 41	140	65	1	cc_PIN_capabilities_req	9F 90 11	→	
				cc_PIN_capabilities_reply	9F 90 12	←	
				cc_PIN_cmd	9F 90 13	→	
				cc_PIN_reply (Примечание 2)	9F 90 14	←	
				cc_PIN_event (Примечание 2)	9F 90 15	←	
				cc_PIN_playback	9F 90 16	→	
				cc_PIN_MMI_req	9F 90 17	→	

Примечания

1 Синтаксис APDU не расширен, но поле LTS_id входит в отдельные протоколы SAC, как указано в 6.4.3.3 настоящего стандарта.

2 Этот APDU расширен и включает поле LTS_id.

Пункт 6.4.3.2 настоящего стандарта определяет APDU, предназначенные для расширения ресурса управления контентом в многопоточном режиме. Упомянутое расширение ресурса выполняется включением поля идентификатора локального TS (LTS_id). Остальные APDU этого типа ресурсов соответствуют синтаксису, приведенному в [1] (11.3.1 и 11.3.2).

В 6.4.3.3 настоящего стандарта определены расширенные протоколы управления контентом, специфицированным в [1].

6.4.3.2 APDU расширенного ресурса управления контентом в многопоточном режиме

6.4.3.2.1 APDU cc_PIN_reply

Хост должен использовать APDU cc_PIN_reply для протокола начала записи. В случае активности многопоточного режима APDU должен дополняться включением LTS_id.

Синтаксис этого расширенного APDU должен быть в соответствии с таблицей 7.

Таблица 7 — Синтаксис APDU cc_PIN_reply

Синтаксис	Количество битов	Мнемоника
cc_PIN_reply() {		
cc_PIN_reply_tag	24	uimsbf
length_field()		
reserved	7	uimsbf
LTS_bound_flag	1	uimsbf
if (LTS_bound_flag == 1) {		
LTS_id	8	uimsbf
} else {		
reserved	8	uimsbf
}		
PINcode_status_field	8	uimsbf
}		

Семантика полей APDU cc_PIN_reply:

- **PID_select_reply_tag**: поле 24 бита является тегом со значением 0x9F9202;
- **length_field**: поле содержит длину полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_bound_flag**: при установке на 0b1 флаг показывает, что APDU cc_PIN_reply связан с конкретным локальным TS. При установке на 0b0 показывает, что APDU cc_PIN_reply не связан с LTS_id, например, когда передается в ответ на cc_PIN_cmd, APDU cc_PIN_playback или cc_PIN_MMI_req;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **PINcode_status_field**: поле 8 битов, семантика этого поля должна быть в соответствии с [1] (11.3.2.3).

6.4.3.2.2 APDU cc_PIN_event

APDU cc_PIN_event используется для протокола начала записи. В случае активности многопоточного режима он расширяется включением LTS_id. Синтаксис APDU cc_PIN_event должен быть в соответствии с таблицей 8.

Таблица 8 — Синтаксис APDU cc_PIN_event

Синтаксис	Количество битов	Мнемоника
cc_PIN_event () {		
cc_PIN_reply_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
program_number	16	uimsbf
PINcode_status_field	8	uimsbf
rating	8	uimsbf
pin_event_time_utc	40	uimsbf
pin_event_time_centiseconds	8	uimsbf
private_data	8x15	uimsbf
}		

Семантика полей APDU cc_PIN_event:

- **cc_PIN_event_tag**: поле 24 бита является тегом со значением 0x9F9015;
- **length_field**: поле содержит длину полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS.

Семантика других полей должна быть в соответствии с [1] (11.3.2.4).

6.4.3.3 Расширенные протоколы управления

6.4.3.3.1 Протокол передачи и подтверждения URI

В случае активности многопоточного режима URI передается от CICAM к Хосту в составе APDU cc_sac_data_req. Хост должен ответить CICAM передачей APDU cc_sac_data_cnf.

Параметры модифицированного протокола передачи и подтверждения URI должны быть в соответствии с таблицей 9.

Таблица 9 — Параметры модифицированного протокола передачи и подтверждения URI

Операция	APDU	Содержание		
1 CICAM передает URI на Хост	cc_sac_data_req	send_datatype_nbr = 3		
		i	datatype_id	datatype_len, бит
		0	25 (operating_mode)	64

Окончание таблицы 9

Операция	APDU	Содержание		
1 CICAM передает URI на Хост	cc_sac_data_req	1	26 (program_number)	16
		2	50 (LTS_id)	8
		request_datatype_nbr = 1		
		i	datatype_id	
		0	27 (uri_confirm)	
2 Хост передает CICAM подтверждение	cc_sac_data_cnf	send_datatype_nbr = 1		
		i	datatype_id	datatype_len, бит
		0	27 (uri_confirm)	256

CICAM должен получить от Хоста APDU cc_sac_data_cnf перед отправлением следующего сообщения протокола передачи и подтверждения URI.

6.4.3.3.2 Протокол начала записи

В случае активности многопоточного режима Хост, информируя CICAM о начале записи, передает APDU cc_sac_data_req, на который CICAM должен ответить Хосту подтверждением о приеме и передать APDU cc_sac_data_cnf.

Параметры модифицированного протокола начала записи, расширенного при использовании многопоточного режима, должны быть в соответствии с таблицей 10. Хост должен дождаться от CICAM подтверждения APDU cc_sac_data_cnf перед отправлением следующего сообщения протокола начала записи.

Таблица 10 — Параметры модифицированного протокола начала записи

Операция	APDU	Содержание		
1 Хост информирует CICAM о начале записи	cc_sac_data_req	send_datatype_nbr = 3 или 4		
		i	datatype_id	datatype_len, бит
		0	38 (operating_mode)	8
		1	26 (program_number)	16
		2	39 (PINcode data)	переменное количество (опционально)
		3	50 (LTS_id)	8
		request_datatype_nbr = 1		
		i	datatype_id	
		0	40 (record_start_status)	
2 CICAM передает Хосту подтверждение	cc_sac_data_cnf	send_datatype_nbr = 1		
		i	datatype_id	datatype_len, бит
		0	40 (record_start_status)	8

При выполнении протокола начала записи Хост указывает, поддерживается выбранная программа пользователем или не поддерживается. Если выбранная программа пользователем не поддерживается, то CICAM должен воздержаться от использования MMI высокого уровня или приложения MMI для диалога с пользователем по поводу выбранной программы. CICAM должен рассматривать выбранную программу как необслуживаемую, когда operating_mode будет установлен равным или 0x01 (Timeshift), или 0x02 (Unattended_Recording).

6.4.3.3.3 Расширение протокола остановки записи

В случае активности многопоточного режима остановка записи выполняется передачей от Хоста на CICAM APDU cc_sac_data_req. CICAM должен ответить Хосту передачей APDU cc_sac_data_cnf.

Параметры модифицированного протокола остановки записи, расширенного при обслуживании нескольких потоков, должны быть в соответствии с таблицей 11.

Таблица 11 — Параметры модифицированного протокола остановки записи

Операция	APDU	Содержание		
1 Хост информирует CICAM об остановке записи	cc_sac_data_req	send_datatype_nbr = 2		
		i	datatype_id	datatype_len, бит
		0	26 (program_number)	16
		1	50 (LTS_id)	16
		request_datatype_nbr = 1		
		i	datatype_id	
		0	42 (record_stop_status)	
2 CICAM передает Хосту подтверждение	cc_sac_data_cnf	send_datatype_nbr = 1		
		i	datatype_id	datatype_len, бит
		0	42 (record_stop_status)	8

Хост должен дождаться от CICAM подтверждения перед отправлением следующего сообщения протокола остановки записи.

6.4.3.3.4 Расширение протокола изменения режима работы

В случае активности многопоточного режима Хост информирует CICAM об изменении режима работы и посылает APDU cc_sac_data_req. CICAM должен ответить Хосту передачей APDU cc_sac_data_cnf.

Параметры модифицированного протокола изменения режима работы должны быть в соответствии с таблицей 12.

Таблица 12 — Параметры модифицированного протокола изменения режима работы

Операция	APDU	Содержание		
1 Хост информирует CICAM об изменении режима работы	cc_sac_data_req	send_datatype_nbr = 3		
		i	datatype_id	datatype_len, бит
		0	38 (operating_mode)	8
		1	26 (program_number)	16
		2	50 (LTS_id)	8
		request_datatype_nbr = 1		
		i	datatype_id	
		0	41 (mode_change_status)	
2 CICAM передает Хосту подтверждение	cc_sac_data_cnf	send_datatype_nbr = 1		
		i	datatype_id	datatype_len, бит
		0	41 (mode_change_status)	8

Перед отправлением следующего сообщения протокола изменения режима работы Хост должен дождаться подтверждения от CICAM.

6.4.3.3.5 Расширение протокола замены лицензии между CICAM и Хостом

В случае активности многопоточного режима расширение протокола замены лицензии между CICAM и Хостом выполняется передачей от CICAM на Хост APDU cc_sac_data_req с контентом лицензии. Хост должен ответить CICAM APDU cc_sac_data_cnf с подтверждением получения.

Параметры протокола замены лицензии между CICAM и Хостом, расширенного для работы в многопоточном режиме, должны быть в соответствии с таблицей 13.

Таблица 13 — Параметры протокола замены лицензии между CICAM и Хостом, расширенного для работы в многопоточном режиме

Операция	APDU	Содержание		
1 CICAM поставляет Хосту контент лицензии	cc_sac_data_req	send_datatype_nbr = 4 или 5		
		i	datatype_id	datatype_len, бит
		0	26 (program_number) (Примечание 1)	16
		1	34 (license_status) (Примечание 2)	8
		2	25 (uri_message)	64
		3	33 (cicam_license)	переменное количество (опционально)
		4	50 (LTS_id)	8
		request_datatype_nbr = 1		
		i	datatype_id	
		0	35 (license_rcvd_status) (Примечание 3)	
2 Хост подтверждает получение	cc_sac_data_cnf	send_datatype_nbr = 1		
		i	datatype_id	datatype_len, бит
		0	35 (license_rcvd_status) (Примечание 3)	8

Примечания

1 program_number соответствует program_number в сообщениях Record Start (начало записи).

2 Допустимые значения и значение этого поля содержатся в [1] (таблица 11.45).

3 Допустимые значения и значение этого поля содержатся в [1] (таблица 11.42).

CICAM должен ожидать от Хоста подтверждения перед передачей других сообщений от CICAM к Хосту протокола замены лицензии.

6.4.4 Ресурсы поддержки условного доступа

6.4.4.1 Общие замечания

Для поддержки многопоточного режима ресурс поддержки условного доступа должен обеспечиваться новым resource_type, определенным как (resource_type = 2, версия 1), в котором APDU ca_pmt и ca_pmt_reply расширены добавлением к синтаксису APDU идентификатора локального TS (LTS_id). APDU ca_pmt расширяется также включением поля PMT_PID с целью экономии ресурсов CICAM для выполнения задачи синтаксического анализа локального TS при его получении.

Хост и CICAM должны использовать эти расширенные APDU в многопоточном режиме.

6.4.4.2 APDU ca_pmt

APDU ca_pmt расширяется включением идентификатора локального TS и поля PMT_PID. Синтаксис APDU ca_pmt должен быть в соответствии с таблицей 14.

Таблица 14 — Синтаксис APDU ca_pmt

Синтаксис	Количество битов	Мнемоника
ca_pmt () {		
ca_pmt_tag	24	uimsbf

Окончание таблицы 14

Синтаксис	Количество битов	Мнемоника
length_field ()		
LTS_id	8	uimsbf
ca_pmt_list_management	8	uimsbf
program_number	16	uimsbf
reserved	3	bslbf
PMT_PID	13	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
reserved	4	bslbf
program_info_length	12	uimsbf
if (program_info_length != 0) {		
ca_pmt_cmd_id /* at program level */	8	uimsbf
for (i = 0; i < n; i++) {		
ca_descriptor() /* CA descriptor at programme level */		
}		
}		
for (i = 0; i < n; i++) {		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID /* elementary stream PID */	13	uimsbf
reserved	4	bslbf
es_info_length	12	uimsbf
if (es_info_length != 0) {		
ca_pmt_cmd_id /* at ES level */	8	uimsbf
for (i = 0; i < n; i++) {		
ca_descriptor () /* CA descriptor at elementary stream level */		
}		
}		
}		
}		

Семантика полей APDU ca_pmt:

- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **ca_pmt_list_management**: поле 8 битов в режиме одиночного потока указывает, что пользователь выбрал одну программу (состоит из одного или нескольких ES) или несколько программ. В многопоточном режиме каждая программа должна соответствовать отдельному локальному TS, в этом случае

могут быть использованы значения, приведенные в таблице 15. Это подмножество определено в [2] (8.4.3.4).

Таблица 15 — Значения поля ca_pmt_list_management для отдельного локального TS в многопоточном режиме

Значение	Содержание
0x03	Only (только)
0x05	Update (обновление)
Другие значения	Reserved (зарезервировано)

В многопоточном режиме «Only (только)» используется для запуска новой программы в связанном локальном TS. Эта операция не влияет на другие локальные работающие TS:

- **PMT_PID**: поле 13 битов содержит PID PMT выбранной службы. Всякий раз когда PID PMT выбранной службы изменяется, Хост должен отправить APDU ca_pmt с полем ca_pmt_list_management, содержащим 0x05 (обновление);
- **другие поля** должны быть в соответствии с [2] (таблица 25).

6.4.4.3 APDU ca_pmt_reply

В случае активности многопоточного режима APDU ca_pmt_reply расширяется добавлением поля идентификатора локального TS (LTS_id). Синтаксис расширенного APDU ca_pmt_reply должен быть в соответствии с таблицей 16.

Таблица 16 — Синтаксис расширенного APDU ca_pmt_reply

Синтаксис	Количество битов	Мнемоника
ca_pmt_reply () {		
ca_pmt_reply_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
ca_enable_flag	1	bslbf
if (ca_enable_flag == 1) {		
ca_enable /* at programme level */	7	uimsbf
} else {		
reserved	7	bslbf
}		
for (i = 0; i < n; i++) {		
reserved	3	bslbf
elementary_PID	13	uimsbf
ca_enable_flag	1	bslbf
if (ca_enable_flag == 1) {		
ca_enable /* at elementary stream level */	7	uimsbf

Окончание таблицы 16

Синтаксис	Количество битов	Мнемоника

} else {
reserved
}
}
}
	7	bslbf

Семантика полей APDU ca_pmt_reply:

- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **семантика других полей** должна быть в соответствии с [2] (таблица 26).

6.4.5 Ресурсы Хоста управления несколькими потоками

6.4.5.1 Общие замечания

Многопоточный ресурс управления Хоста (с ID ресурса 0x00200081) базируется на версии 3 управления Хоста DVB. Эта версия ресурса позволяет CICAM запрашивать Хост о необходимом типе настройки или для презентации (представления), или фоновой. Хост должен ответить на запрос информацией о типе настройки с LTS_id потока.

Если Хост предоставит CICAM фоновую настройку, то CICAM должен ответить на APDU ask_release в течение одной секунды «Release_OK» в APDU ask_release_reply и должен выделить этот ресурс, не используя приложения MMI или MMI высокого уровня. Эти операции не должны оказывать влияние на поток презентации.

Перечень APDU, поддерживающих многопоточный тип ресурса управления Хоста, должен быть в соответствии с таблицей 17. Все другие APDU ресурса управления Хоста DVB (версия 3) не должны поддерживаться.

Таблица 17 — Перечень APDU, поддерживающих многопоточный тип ресурса управления Хоста

APDU	Направление передачи	
	Хост	CICAM
tune_broadcast_req	←	
tune_triplet_req	←	
une_lcn_req	←	
tune_ip_req	←	
tune_reply	→	
ask_release	→	
ask_release_reply	←	
tuner_status_req	←	
tuner_status_reply	→	

CICAM не должен на интервале одного сеанса запрашивать настройку для презентации и фоновую настройку. При необходимости выполнения двух видов настроек (для презентации и фоновой настройки) CICAM должен запрашивать другой сеанс к ресурсу управления Хостом. Многопоточный Хост должен поддерживать не менее двух сеансов многопоточного ресурса управления Хоста DVB.

На интервале многопоточного сеанса управления Хоста любой запрос настройки отменяет и заменяет предыдущий запрос в этом сеансе.

В многопоточном режиме может потребоваться фильтрация PID для сохранения пропускной способности интерфейса TS на уровне ниже максимальной. После подтверждения успешного запроса настройки APDU tune_reply Хост должен обеспечить минимально необходимое подмножество PID в за-

висимости от типа запроса настройки. Если целью запроса настройки от CICAM является служба, то Хост должен обеспечить необходимые PID в соответствии с 6.3.2 настоящего стандарта. CICAM может запрашивать дополнительные PID, используя APDU pid_select_req.

Если целью запроса настройки от CICAM является не служба, а частота настройки, то Хост должен предоставить совокупность PID в соответствии с 6.3.3 настоящего стандарта. CICAM может запрашивать дополнительные PID, используя APDU pid_select_req. Синтаксис APDU tune_broadcast_req, tune_triplet_req, tune_lcn_req, tune_ip_req и tune_reply изменен в их многопоточных версиях, чтобы позволить CICAM запрашивать настройку для презентации пользователю или фоновую настройку, означающую, что поток не предназначен для презентации. Все другие APDU сохраняют синтаксис ресурса версии 3 управления Хостом DVB, определенный в разделе 13 настоящего стандарта. В 6.4.5.2—6.4.5.6 настоящего стандарта определены измененные синтаксисы соответствующих APDU для многопоточных операций.

6.4.5.2 APDU tune_broadcast_req

Версия APDU tune_broadcast_request в многопоточном режиме позволяет CICAM указать, необходимо или нет представлять запрос настройки пользователю. Хост должен ответить APDU tune_reply, информируя CICAM, с каким LTS_id будет отправлен требуемый поток. Синтаксис APDU tune_broadcast_req представлен в таблице 18.

Таблица 18 — Синтаксис APDU tune_broadcast_req

Синтаксис	Количество битов	Мнемоника
tune_broadcast_req(){		
tune_broadcast_req_tag	24	uimsbf
length_field()		
reserved	4	uimsbf
background_tune_flag	1	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
pmt_flag	1	uimsbf
service_id	16	uimsbf
reserved	4	uimsbf
descriptor_loop_length	12	uimsbf
for (i=0; i<N; i++){		
descriptor()		
}		
if (pmt_flag == 1) {		
program_map_section()		
}		
}		

Семантика полей APDU tune_broadcast_request:

- **background_tune_flag**: флаг указывает режим, в котором должна быть выполнена настройка: в фоновом режиме или для презентации пользователю. Значение 0b0 указывает, что настройка должна быть в режиме презентации; значение 0b1 указывает, что настройка должна быть выполнена в фоновом режиме. В случае запроса фоновой настройки флаги tune_quietly_flag и keep_app_running_flag должны игнорироваться;
- **семантика других полей** должна быть в соответствии с 13.2.1 настоящего стандарта.

6.4.5.3 APDU tune_triplet_req

Версия APDU tune_triplet_req в многопоточном режиме добавляет возможность, позволяющую CICAM указать, должен быть запрос настройки представлен пользователю или нет. Хост должен ответить APDU tune_reply, сообщая CICAM, с каким LTS_id будет отправлен запрошенный поток настройки.

Синтаксис APDU tune_triplet_req представлен в таблице 19.

Таблица 19 — Синтаксис APDU tune_triplet_req

Синтаксис	Количество битов	Мнемоника
tune_triplet_req () {		
tune_triplet_req_tag	24	uimsbf
length_field()		
reserved	5	uimsbf
background_tune_flag	1	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
delivery_system_descriptor_tag	8	uimsbf
if (delivery_system_descriptor_tag == 0x7f){		
descriptor_tag_extension	8	uimsbf
}		
else {		
reserved	8	uimsbf
}		
}		

Семантика полей APDU tune_triplet_req:

- **background_tune_flag**: в соответствии с 6.4.5.2 настоящего стандарта;
- **семантика других полей**: в соответствии с 13.2.2 настоящего стандарта.

6.4.5.4 APDU tune_lcn_req

Версия APDU tune_lcn_req в многопоточном режиме добавляет возможность CICAM указывать, должен представляться пользователю запрос настройки или нет. Хост должен ответить APDU tune_reply, информируя CICAM, на который LTS_id будет отправлен запрошенный поток настройки.

Синтаксис APDU tune_lcn_req представлен в таблице 20.

Таблица 20 — Синтаксис APDU tune_lcn_req

Синтаксис	Количество битов	Мнемоника
tune_lcn_req () {		
tune_lcn_req_tag	24	uimsbf
length_field()		
reserved	7	uimsbf
background_tune_flag	1	uimsbf

Окончание таблицы 20

Синтаксис	Количество битов	Мнемоника
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
logical_channel_number	14	uimsbf
}		

Семантика полей APDU tune_ lcn_req:

- **background_tune_flag**: в соответствии с 6.4.5.2 настоящего стандарта;
- **семантика других полей**: в соответствии с 13.2.3 настоящего стандарта.

6.4.5.5 APDU tune_ip_req

Версия APDU tune_ip_req в многопоточном режиме добавляет средство, позволяющее CICAM указать, должен представляться пользователю запрос настройки или нет. Хост должен ответить APDU tune_reply, информируя CICAM, на который LTS_id будет отправлен запрошенный поток настройки.

Синтаксис APDU tune_ip_req представлен в таблице 21.

Таблица 21 — Синтаксис APDU tune_ip_req

Синтаксис	Количество битов	Мнемоника
tune_ip_req () {		
tune_ip_req_tag	24	uimsbf
length_field()		
reserved	1	uimsbf
background_tune_flag	1	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
service_location_length	12	uimsbf
for (i=0; i<N; i++) {		
service_location_data	8	uimsbf
}		
}		

Семантика полей APDU tune_ip_req:

- **background_tune_flag**: в соответствии с 6.4.5.2 настоящего стандарта;
- **семантика других полей**: в соответствии с 13.2.4 настоящего стандарта.

6.4.5.6 APDU tune_reply

APDU tune_reply отправляет Хост в ответ на каждый из трех предыдущих типов запроса настройки, определенных для многопоточного ресурса управления Хоста DVB. Он используется для того, чтобы сообщить CICAM об успешной настройке и LTS_id, с которым требуемый поток будет отправлен.

Синтаксис APDU tune_reply представлен в таблице 22.

Таблица 22 — Синтаксис APDU tune_reply

Синтаксис	Количество битов	Мнемоника
tune_reply () {		
tune_reply_tag	24	uimsbf
length_field ()		

Окончание таблицы 22

Синтаксис	Количество битов	Мнемоника
LTS_id	8	uimsbf
status_field	8	uimsbf
}		

Семантика полей APDU tune_reply:

- **LTS_id**: поле 8 битов содежит идентификатор локального TS, который может быть найден по запросу настройки локального TS. LTS_id должен игнорироваться, если запрос настройки будет неудачным;
- **все другие поля** должны быть в соответствии с [1] (таблица 14.30).

6.4.6 Ресурс приложения MMI

6.4.6.1 Общие замечания

Поддержка функций многопоточного режима должна обеспечиваться новым типом ресурса приложения MMI (resource_type 2, версия 1), в котором APDU RequestStart расширен добавлением к синтаксису APDU LTS_id. Хост может использовать дополнительный LTS_id для того, чтобы связать запрос с ситуацией, когда Хост отображает одновременно более одной программы (например, картина в картине, мозаика или предоставляет функцию двойного дисплея).

Многопоточный ресурс приложения MMI базируется на ресурсе приложения MMI версия 3, тип 1, специфицированного в 12.3 настоящего стандарта, и опционально включает ADQ.

6.4.6.2 APDU RequestStart

Синтаксис расширенного APDU RequestStart, содержащего идентификатор локального TS, должен быть в соответствии с таблицей 23.

Таблица 23 — Синтаксис расширенного APDU RequestStart, содержащего идентификатор локального TS

Синтаксис	Количество битов	Мнемоника
RequestStart(){		
RequestStartTag	24	uimsbf
length_field()		
reserved	7	bslbf
LTS_bound_flag	1	bslbf
if (LTS_bound_flag == 1) {		
LTS_id	8	uimsbf
} else {		
reserved	8	bslbf
}		
AppDomainIdentifierLength	8	uimsbf
InitialObjectLength	8	uimsbf
for (i=0; i<AppDomainIdentifierLength; i++){		
AppDomainIdentifier	8	uimsbf
}		
for (i=0; i< InitialObjectLength; i++){		
InitialObject	8	uimsbf
}		
}		

Семантика полей APDU RequestStart:

- **LTS_bound_flag**: флаг, установленный на 0b1, указывает, что запрос связан с определенным локальным TS. Когда этот флаг установлен на 0b0, это означает, что запрос не связан с определенным локальным TS, например, в ответ на отправленный Хостом APDU enter_menu;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, используется CICAM для того, чтобы ассоциировать объект RequestStart с локальным TS, когда LTS_bound_flag установлен в 0b1. Хост может использовать значение LTS_id для того, чтобы ассоциировать запрос с областью отображения, где в настоящее время отображается соответствующая программа, если дескремблируется более одной программы;
- **семантика других полей** должна быть в соответствии с [8] (таблица 62).

6.4.6.3 APDU RequestStartAck

APDU RequestStartAck определен в [8] (6.5.3). Семантика поля AckCode в расширенной версии при значении AckCode = 0x05 соответствует случаю «не обслуживается пользователем», когда конечный пользователь отсутствует.

После приема APDU RequestStartAck с AckCode = 0x05 CICAM не должен передавать еще один APDU RequestStart для этого LTS_id до тех пор, пока:

- Хост не укажет, что пользователь передал протокол начала записи или протокол остановки записи;
- Хост не укажет, что LTS_id предназначен для другой программы, передачей APDU ca_pmt или APDU sd_start.

6.4.6.4 APDU FileRequest

APDU FileRequest должен быть в соответствии с [8] (6.5.4).

6.4.6.5 APDU FileAck

APDU FileAck должен быть в соответствии с [8] (6.5.5).

6.4.6.6 APDU AppAbortRequest

APDU AppAbortRequest должен быть в соответствии с [8] (6.5.6).

6.4.6.7 APDU AppAbortAck

APDU AppAbortAck должен быть в соответствии с [8] (6.5.7).

6.4.7 Ресурсы MMI высокого уровня

6.4.7.1 Общие замечания

Поддержка многопоточного режима ресурсом MMI высокого уровня обеспечивается применением нового типа ресурса (resource_type = 2, версия = 1), в котором APDU enq, menu и list расширены добавлением к синтаксису APDU идентификатора LTS_id. Хост может использовать дополнительный LTS_id для того, чтобы связать запрос с отображением Хостом более одной программы одновременно (например, картинка в картинке, мозаика или сдвоенный дисплей).

Хост и CICAM должны использовать расширенный APDU в случае активности многопоточного режима.

6.4.7.2 APDU enq

Расширенный APDU enq включает в себя идентификатор локального TS. Синтаксис APDU enq должен быть в соответствии с таблицей 24.

Таблица 24 — Синтаксис APDU enq

Синтаксис	Количество битов	Мнемоника
enq(){		
enq_tag	24	uimsbf
length_field()		
reserved	6	bslbf
LTS_bound_flag	1	bslbf
blind_answer	1	bslbf
if (LTS_bound_flag == 1) {		
LTS_id	8	uimsbf
}else {		

Окончание таблицы 24

Синтаксис	Количество битов	Мнемоника
reserved	8	bslbf
}		
answer_text_length	8	uimsbf
for (i=0; i<enq_length - 3; i++){		
text_char	8	uimsbf
}		
}		

Семантика полей APDU enq:

- **LTS_bound_flag**: флаг при установке на 0b1 указывает, что запрос связан с определенным локальным TS. Когда этот флаг установлен на 0b0, это означает, что запрос не связан с определенным локальным TS, например, в ответ на отправленный Хостом APDU enter_menu;
- **LTS_id**: идентификатор локального TS используется CICAM для того, чтобы связать объект запроса с локальными TS, когда LTS_bound_flag установлен в 0b1. Хост может использовать значение LTS_id для того, чтобы связать объект запроса с зоной отображения, в которой отображается соответствующая программа, когда дескремблируется более одной программы;

семантика полей **blind_answer, answer_text_length, text_char** должна быть в соответствии с [2] (таблица 47).

6.4.7.3 APDU answ

APDU answ определяется в соответствии с [2] (8.6.5.3). Кодирование поля answ_id должно быть в соответствии с таблицей 25.

Таблица 25 — Кодирование поля answ_id

answ_id	Значение
Cancel (отменить)	0x00
Answer (ответ)	0x01
Reserved (зарезервировано)	от 0x02 до 0xFE
Unattended (не обслуживается)	0xFE

Значение Unattended (не обслуживается) означает, что LTS_id соответствует программе, которая в настоящий момент не обслуживается конечным пользователем (например, Хост в настоящий момент записывает необслуживаемую программу), и Хост не может вывести на экран требуемый объект запроса. После приема APDU answ с answ_id «не обслуживается пользователем» CICAM должен воздерживаться от отправки другого объекта MMI высокого уровня для того же LTS_id до тех пор, пока:

- Хост не укажет, что программа снова обслуживается пользователем или при запуске протокола начала записи, или при запуске протокола остановки записи;
- Хост не укажет, что LTS_id выделяется для другой программы при помощи APDU ca_pmt или APDU sd_start.

6.4.7.4 APDU menu

APDU menu определяется в соответствии с [2] (8.6.5.4). Семантика APDU menu расширяется включением полей локального TS. Синтаксис APDU menu в соответствии с таблицей 26.

Таблица 26 — Синтаксис APDU menu

Синтаксис	Количество битов	Мнемоника
menu(){		
menu_tag	24	uimsbf

Окончание таблицы 26

Синтаксис	Количество битов	Мнемоника
length_field()		
reserved	7	bslbf
LTS_bound_flag	1	bslbf
if (LTS_bound_flag == 1) {		
LTS_id	8	uimsbf
}else {		
reserved	8	bslbf
}		
choice_nb	8	uimsbf
TEXT() /* title text */		
TEXT() /* sub-title text */		
TEXT() /* bottom text */		
for (i=0; i<choice_nb; i++) { /* when choice_nb != 'FF' */		
TEXT()		
}		
}		

Семантика полей APDU menu:

- **LTS_bound_flag**: флаг, когда он установлен в 0b1, указывает, что объект запроса связывается с определенным локальным TS. Когда этот бит устанавливается в 0b0, это указывает, что запрос не связан с определенным локальным TS, например, он может быть ответом Хосту на APDU enter_menu;

- **LTS_id**: поле 8 битов, идентификатор локального TS используется CICAM для связывания объекта меню с локальным TS, когда LTS_bound_flag установлен в положение 0b1. Хост может использовать значение LTS_id для связывания объекта меню с областью отображения, в которой отображается соответствующая программа, при дескремблировании более одной программы.

6.4.7.5 APDU menu_answ

APDU menu_answ определен в [2] (8.6.5.5). Семантика поля choice_ref расширяется в соответствии с таблицей 27.

Таблица 27 — Кодирование поля choice_ref

choice_ref	Значение
Cancel (отменить)	0x00
Количество выборов, сделанных пользователем	от 0x01 до 0xFE
Unattended (не обслуживается)	0xFE

Значение Unattended (не обслуживается) означает, что LTS_id соответствует программе, которая в настоящий момент не обслуживается конечным пользователем (например, Хост в настоящий момент записывает необслуживаемую программу), и Хост не может вывести на экран требуемый объект запроса. После приема APDU answ с answ_id Unattended (не обслуживается), CICAM должен воздерживаться от отправки другого объекта MMI высокого уровня для того же LTS_id до тех пор, пока:

- Хост не укажет, что программа снова обслуживается или при запуске протокола начала записи, или при запуске протокола остановки записи;

- Хост не укажет, что LTS_id выделяется для другой программы при помощи APDU ca_pmt или APDU sd_start.

6.4.7.6 APDU list

APDU list определен в [2] (8.6.5.5). Синтаксис расширенного APDU list представлен в таблице 28.

Таблица 28 — Синтаксис расширенного APDU list

Синтаксис	Количество битов	Мнемоника
list(){		
list_tag	24	uimsbf
length_field()		
reserved	7	bslbf
LTS_bound_flag	1	bslbf
if (LTS_bound_flag == 1) {		
LTS_id	8	uimsbf
}else {		
reserved	8	bslbf
}		
item_nb	8	uimsbf
TEXT() /* title text */		
TEXT() /* sub-title text */		
TEXT() /* bottom text */		
for (i=0; i<item_nb; i++){ /* when item_nb != 'FF' */		
TEXT()		
}		
}		

Семантика APDU list расширяется включением полей локального TS:

- **LTS_bound_flag**: флаг указывает, что объект запроса связывается с определенным локальным TS, когда он установлен в 0b1. Когда флаг устанавливается в 0b0, это указывает, что запрос не связан с определенным локальным TS, например, он может быть ответом Хосту на APDU enter_menu;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS и используется CICAM для связывания объекта меню с локальным TS, когда LTS_bound_flag установлен в положение 0b1. Хост может использовать значение LTS_id для связывания объекта меню с областью отображения, в которой отображается соответствующая программа, при дескремблировании более одной программы.

6.4.7.7 APDU close_mmi

APDU close_mmi должен быть в соответствии с [2] (8.6.2.1).

6.4.7.8 APDU display_control

APDU display_control должен быть в соответствии с [2] (8.6.2.2).

6.4.7.9 APDU display_reply

APDU display_reply должен быть в соответствии с [2] (8.6.2.3).

7 Хост в режиме проигрывателя контента IP

7.1 Общие замечания

В разделе 7 настоящего стандарта нормируются параметры обработки контента IP, представленного в форматах: TS, ISOBMFF, MPEG DASH, содержащего контент или в формате TS, или в формате ISOBMFF.

Контент IP может поставляться в любом формате контейнера TS, или ISOBMFF, или в любом формате файла, встроенного в объект DASH. Если контент поставляется не в формате TS, Хост управляет деинкапсуляцией семплов контента в специфический базовый формат контейнера TS CI Plus™ IP контента. Этот формат применяется при обмене между Хостом и CICAM.

При работе Хоста в режиме проигрывателя CICAM может применить скремблирование управления контента CI Plus™ для защиты созданных элементарных потоков.

При переносе IP контента в контейнере формата TS или в контейнере какого-либо другого формата, исключая формат стандартных TS, защищенных CA, интерфейс TS работает в режиме семпла, а в случаях обычного TS он работает в нормальном режиме.

В режиме семпла метаданные DRM и управляющая информация, связанная с поставленным IP-элементом контента, передаются от Хоста на CICAM через интерфейс команд прежде, чем проигрывание может быть запущено. Метаданные DRM, изменяющиеся между семплами, например IV, переносят с данными семпла внутри полосы через интерфейс TS.

В режиме семпла CICAM накапливает в буфере входящие данные для дескремблирования. По этой причине, как правило, дескремблирование обычных TS с постоянной задержкой пакетов через CICAM в режиме семпла не применяется.

Если информация о дате и времени от тюнера вещания недоступна, то Хост и CICAM могут получить информацию о дате и времени через сетевой интерфейс IP. Настоящий стандарт не определяет правила получения информации о дате и времени через сетевой интерфейс IP.

Возможности выполнения Хостом услуги игнорирования, определенные в [1] (раздел 10), применяемые к вещательным службам, не должны использоваться для служб, доставленных IP через Хост в режиме проигрывателя.

7.2 Режимы интерфейса транспортного потока

При дескремблировании семпла CICAM может буферизировать принимаемый семпл перед его дескремблированием с последующей передачей обратно на Хост через интерфейс TS. Буферизация семпла в CICAM позволяет дескремблировать семпл, который будет создан, как правило, с новым IV. Метод буферизации в CICAM и количество дескремблеров, доступных в CICAM, настоящим стандартом не определяются.

Хост сообщает CICAM о формате передаваемого контента, устанавливая интерфейс TS (или локальный TS в контексте многопоточного режима) в режим семпла или в нормальный режим.

Хост управляет конфигурацией интерфейса TS с помощью APDU sd_start и ca_pmt. APDU sd_start является составной частью ресурса дескремблирования семпла, который определен в 7.4 настоящего стандарта.

При изменении режима интерфейса TS данные, относящиеся к предыдущему режиму в Хосте или CICAM, должны быть отброшены.

7.3 Интерфейс команд

7.3.1 Общие замечания

При работе интерфейса TS в режиме семпла Хост должен координировать доставку данных мультимедийных файлов в CICAM для обеспечения правильного дескремблирования контента. С этой целью CICAM также должен получать конкретную информацию о метаданных в медиафайле.

7.3.2 Инициирование проигрывания

Хост управляет конфигурацией интерфейса TS, который по умолчанию находится в нормальном режиме. Хост должен использовать APDU sd_start, чтобы переключить интерфейс TS или локальный TS из мультипотокового режима в режим семпла.

При приеме от Хоста APDU sd_start CICAM должен завершить дескремблирование любого контента нормального режима, ожидать контент, инициируемый APDU ca_pmt, и подтвердить конфигурацию TS, отправляя APDU sd_start_reply. После передачи APDU sd_start интерфейс TS или локальный TS должны считаться находящимися в режиме семпла.

С помощью APDU sd_start Хост может запрашивать дескремблирование семплов, поступающих из разных треков. Эти треки, находящиеся или в одном и том же медиафайле или в разных медиафайлах, обозначаются как треки семплов и могут иметь свои собственные метаданные DRM.

В частном случае TS мультимедийного файла Хост не должен выполнять синтаксический анализ файла перед передачей на интерфейс TS, так как предполагается, что CICAM способен извлекать данные, необходимые ему для дескремблирования. В частности, Хосту нет необходимости определять элементарные потоки, содержащиеся в TS, перед передачей на интерфейс TS. При этом считается, что медиафайл TS представляет собой один трек семплов.

После того как CICAM будет готов к получению семплов, он должен подтвердить готовность отправкой APDU sd_start_reply. После получения этого подтверждения Хост может начать передачу первых семплов. Если CICAM не может дескремблировать семпл, например, пока он находится в процессе получения лицензии от сервера DRM, то это состояние должно быть указано в подтверждении готовности. Когда CICAM будет способен дескремблировать семплы, он должен послать Хосту второе подтверждение, используя APDU sd_start_reply с обновленным статусом. Теперь CICAM должен запустить дескремблирование семплов и отослать дескремблированные семплы назад к Хосту через обратный интерфейс TS. Хост не должен отправлять данные семпла в количестве, превышающем возможности буфера CICAM, до получения второго подтверждения от CICAM и подтверждения уведомления от буферного механизма об уровне заполнения буфера в обратном TS о том, что буфер CICAM очищен. Информация о размере буфера CICAM для LTS сообщается CICAM в APDU sd_start_reply.

Механизм подтверждения, выполняемого в два этапа, обеспечивает возможность ускорения подачи контента и, следовательно, улучшения качества обслуживания пользователя.

7.3.3 Выполнение проигрывания

Хост может начать передачу семплов через интерфейс TS при получении первого APDU sd_start_reply. В этом случае TS соответствует структуре, определенной в 7.5 настоящего стандарта. Хост должен соблюдать ограничения буфера CICAM, как определено в 7.5.4 настоящего стандарта.

В процессе проигрывания Хост может обновить список участвующих треков семплов. Например, пользователь может изменять язык звукового сопровождения, что может привести к изменению используемого аудиотрека. Такие изменения треков должны быть указаны Хостом с помощью APDU sd_update.

В процессе проигрывания Хост может обновлять метаданные DRM, связанные со списком треков семплов с помощью APDU sd_update.

В процессе проигрывания Хост может полностью изменить список используемых треков семплов с помощью другого APDU sd_start, как правило, при выборе пользователем другого элемента контента, который требует работы в режиме семпла. После приема APDU sd_start CICAM, работающий в режиме семпла, должен прекратить какое-либо дескремблирование, очистить буфер и приготовиться к дескремблированию семплов, соответствующих новому списку файлов. CICAM направляет Хосту не менее одного подтверждения готовности к проигрыванию.

7.3.4 Прекращение проигрывания

Когда Хост заканчивает передачу мультимедийных семплов для CICAM или в связи с окончанием медиафайла, или потому, что пользователь решил остановить его просмотр, Хост может передать APDU ca_pmt. Этот APDU переводит интерфейс TS или локального TS в нормальный режим и находится в состоянии готовности начать дескремблирование потока вещания.

7.4 Ресурс дескремблирования семпла

7.4.1 Последовательность использования ресурса

Ресурс дескремблирования семпла обеспечивает управление дескремблированием CICAM последовательности медиасемплов, упакованных в TS системы MPEG-2 [9].

При необходимости дескремблирования семплов Хост посылает APDU sd_info_req, запрашивая список систем DRM, внедренных в CICAM и поддерживающих дескремблирование семплов. CICAM должен ответить APDU sd_info_reply со списком поддерживаемых drm_system_id и UUID систем защиты контента.

Хост должен декларировать треки семплов, передаваемые через интерфейс TS, для дескремблирования в CICAM с помощью APDU sd_start. Когда CICAM будет готов получить семплы, он должен ответить первым APDU sd_start_reply, а когда он будет готов дескремблировать семплы, он должен ответить вторым APDU sd_start_reply. Если CICAM обладает высоким быстродействием, то он может возвратить только один APDU sd_start_reply, содержащий обе области, соответствующие обновленным полям состояния.

Хост может обновлять список треков семплов, участвующих в процессе дескремблирования семплов, использованием APDU sd_update. Хост может добавить в список новые треки семплов или может удалить из списка несколько треков семплов, которые больше не используются. CICAM должны подтверждать обновления с помощью APDU sd_update_reply.

Хост может обновлять метаданные DRM, связанные с действующими треками семплов, применяя APDU sd_update. Трек семпла определяется по track_PID, выделенным Хостом в APDU sd_start.

Параметры ресурса дескремблирования семпла должны быть в соответствии с таблицей 29.

Таблица 29 — Параметры ресурса дескремблирования семпла

Ресурс дескремблирования семпла				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00920041	146	1	1	sd_info_req	9F 98 00	→	
				sd_info_reply	9F 98 01	←	
				sd_start	9F 98 02	→	
				sd_start_reply	9F 98 03	←	
				sd_update	9F 98 04	→	
				sd_update_reply	9F 98 05	←	

7.4.2 APDU sd_info_req

Запрос списка систем DRM, встроенных в CICAM и поддерживающих дескремблирование семплов, Хост выполняет отправлением в CICAM APDU sd_info_req. Длина поля length_field в составе APDU sd_info_req должна быть в соответствии с форматом BER ASN.1 [2] (8.3.1).

CICAM должен ответить APDU sd_info_reply.

7.4.3 APDU sd_info_reply

CICAM направляет APDU sd_info_reply к Хосту в ответ на APDU sd_info_req. В нем перечислены drm_system_id и UUID для каждой системы защиты контента или системы DRM, которые CICAM поддерживает для дескремблирования семплов. Если система DRM может быть идентифицирована либо drm_system_id, либо его UUID, то должны быть сообщены оба идентификатора.

Синтаксис APDU sd_info_reply представлен в таблице 30.

Таблица 30 — Синтаксис APDU sd_info_reply

Синтаксис	Количество битов	Мнемоника
sd_info_reply() {		
sd_info_reply_tag	24	uimsbf
length_field()	16	
number_of_drm_system_id	8	uimsbf
for (i=0; i<n; i++) {		
drm_system_id	16	uimsbf
}		
number_of_drm_uuid	8	uimsbf
for (i=0; i<n; i++) {		
drm_uuid	128	uimsbf
}		
}		

Ниже представлена семантика полей в составе APDU sd_info_reply:

- **sd_info_reply_tag**: поле 24 бита является тегом этого APDU со значением 0x9F9801;
- **length_field**: поле определяет длину полезной нагрузки этого APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **number_of_drm_system_id**: поле определяет количество записей, содержащихся в следующем списке идентификаторов системы DRM. Этот список не должен содержать идентификаторы систем CA

вещания, которые могут поддерживаться CICAM, но система CA/DRM не поддерживает дескремблирование и CA вещания и дескремблирование семпла.

Примечание — Список поддерживаемых систем CA вещания предоставляет APDU CA_info_reply в ответ на APDU CA_info в соответствии с [2] и [1] (E.17.3);

- **drm_system_id**: поле содержит значения, аналогичные значениям ca_system_id, в соответствии с распределением идентификаторов и кодов для систем DVB [10];
- **number_of_drm_uuid**: поле содержит количество записей в следующем списке UUID DRM;
- **drm_uuid**: поле содержит записи UUID системы DRM, поддерживаемой CICAM.

7.4.4 APDU sd_start

Для инициирования дескремблирования CICAM треков семплов Хост должен послать для CICAM APDU sd_start, который содержит данные о типе интерфейса — интерфейс TS или локальный TS. Перед отправлением APDU Хост должен прекратить отправку любых пакетов, относящихся к этому локальному TS (в случае нескольких потоков) (например, пакеты TS от тюнера вещания).

Хост должен объявить program_number, которые будут использоваться в CICAM в сообщениях URI.

Для каждого трека Хост указывает PID (track_PID), с которыми будут переданы соответствующие семплы. Track_PID затем используется в качестве идентификатора трека в любом следующем APDU sd_update. Хост должен убедиться, что сигнализация PID в этом APDU является уникальной и не используется для других целей. Значения PID в области от 0x0000 до 0x001F зарезервированы.

Если Хост идентифицировал метаданные, связанные с системой DRM, поддерживаемой CICAM, то он должен добавить цикл метаданных DRM для каждого трека в APDU sd_start. Если метаданные DRM применяются ко всем трекам, то Хост должен повторить их для каждого трека.

Синтаксис APDU sd_start представлен в таблице 31.

Таблица 31 — Синтаксис APDU sd_start

Синтаксис	Количество битов	Мнемоника
sd_start() {		
sd_start_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
program_number	16	uimsbf
reserved	7	bslbf
ts_flag	1	uimsbf
if (ts_flag == 1){		
number_of_metadata_records	8	uimsbf
for (i=0; i<number_of_metadata_records; i++) {		
drm_metadata_source	8	uimsbf
drm_system_id	16	uimsbf
drm_uuid	128	uimsbf
drm_metadata_length	16	uimsbf
for (i=0; i<N; i++) {		
drm_metadata_byte	8	uimsbf
}		
}		
}		

Окончание таблицы 31

Синтаксис	Количество битов	Мнемоника
if (ts_flag == 0){		
number_of_sample_tracks	8	uimsbf
for (i=0;i<number_of_sample_tracks;i++){		
reserved	3	bslbf
track_PID	13	uimsbf
number_of_metadata_records	8	uimsbf
for (i=0; i<number_of_metadata_records; i++)		
drm_metadata_source	8	uimsbf
drm_system_id	16	uimsbf
drm_uuid	128	uimsbf
drm_metadata_length	16	uimsbf
for (i=0; i<N; i++) {		
drm_metadata_byte	8	uimsbf
}		
}		
}		
}		
}		

Семантика полей APDU sd_start:

- **sd_start_tag**: поле 24 бита является тегом со значением 0x9F9802, идентифицирует этот APDU;
- **length_field**: поле определяет длину полезной нагрузки в формате BER ASN.1, определенной в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **program_number**: поле 16 битов содержит номер программы, который будет использоваться CICAM в сообщениях URI. Значение TS не должно совпадать со значением поля program_number в PMT, включенной в TS. Для контента, представляемого не в формате TS, поля program_number не должны применяться, и Хост может устанавливать в них любое значение;
- **ts_flag**: имеет значение 0b1, если сделан запрос на дескремблирование семпла в формате TS, и имеет значение 0b0, если запрос сделан на дескремблирование семпла не в формате TS;
- **number_of_Sample_Tracks**: поле 8 битов содержит количество треков семплов, в которых семплы должны быть дескремблированы;
- **track_PID**: поле 13 битов содержит PID, с которым семплы дескремблированных треков семплов отправляются Хосту;
- **number_of_metadata_records**: поле 8 битов содержит количество записей DRM метаданных в этом цикле;
- **drm_metadata_source**: поле 8 битов определяет источник метаданных DRM в соответствии с таблицей 32;

Таблица 32 — Источник метаданных DRM

Источник метаданных DRM	Значение
Не определен	0x00

Окончание таблицы 32

Источник метаданных DRM	Значение
Дескриптор потокового доступа к контенту (Content Access Streaming Descriptor; CASD) — общие данные DRM. Элемент DRMGenericData из элемента дескриптора потокового доступа к контенту копируется в массив drm_metadata_byte как закодированная строка UTF-8. Дескриптор потокового доступа к контенту определен в [11] (4.7 и E.2). Элемент DRMControlInformation определяется в [12] (3.3.2)	0x01
Дескриптор потокового доступа к контенту (CASD) — частные данные DRM. Элемент DRMPrivateData из элемента дескриптора потокового доступа к контенту копируется в массив drm_metadata_byte как закодированная строка UTF-8. Дескриптор потокового доступа к контенту определен в [11] (4.7 и E.2). Элемент DRMControlInformation определяется в [12] (3.3.2)	0x02
Общее шифрование (Common Encryption; CENC) — поле «pssh» заголовка специфичной системы защиты. Данные поля «pssh» копируются в массив drm_metadata_byte. Поле «pssh» определено в [13]	0x03
Описание представления медиа (Media Presentation Description, MPD) — элемент защиты контента. Элемент ContentProtection от элемента Representation (представление) для трека, который будет дескремблирован, копируется в массив drm_metadata_byte как закодированная строка UTF-8. Описание представления медиа определяется в [14]. Элемент ContentProtection определяется в [14] (5.8.4.1 и 5.8.5.2)	0x04
ISOBMFF — поле («sinf») схемы защиты информации. Данные поля «sinf» копируются в массив drm_metadata_byte. Поле «sinf» определено в [5]	0x05
Онлайн SDT (OSDT) — общие данные DRM. Элемент DRMGenericData из элемента OSDT DRMControlInformation копируется в массив drm_metadata_byte как закодированная строка UTF-8. OSDT определена в соответствии с приложением Г настоящего стандарта. Элемент DRMControlInformation определен в [12] (3.3.2)	0x06
Онлайн SDT (OSDT) — частные данные DRM. Элемент DRMPrivateData элемента DRMControlInformation OSDT копируется в массив drm_metadata_byte как закодированная строка UTF-8. OSDT определена в соответствии с приложением Г настоящего стандарта. Элемент DRMControlInformation определен в [12] (3.3.2)	0x07
Зарезервировано	от 0x08 до 0xFF

- **drm_system_id**: идентификация системы DRM со связанными метаданными. Значения для drm_system_id аналогичны значениям для ca_system_id, как определено размещением идентификаторов и кодов для систем DVB [10]. Если поле drm_system_id не будет использоваться для идентификации DRM, то в этом поле должно быть установлено 0xFFFF;
- **drm_uuid**: поле 128 битов содержит идентификатор UUID системы DRM, к которому относятся метаданные. Если drm_uuid не используется для идентификации DRM, то во всех байтах этого поля должно быть установлено 0xFF;
- **drm_metadata_length**: поле содержит длину в байтах метаданных DRM;
- **drm_metadata_source**: поле 8 битов содержит источник метаданных, определенный в таблице 32.

7.4.5 APDU sd_start_reply

CICAM в случае готовности к получению первого семпла для дескремблирования должен направить Хосту APDU sd_start_reply как ответ на APDU sd_start.

При передаче APDU sd_start_reply CICAM сообщает Хосту:

- о готовности CICAM к получению транспортных пакетов от интерфейса TS (transmission_status);
- о способности CICAM к дескремблированию транспортных пакетов (drm_status);
- о размере буфера (buffer_size), выделенного CICAM для всех объявленных треков семплов в APDU sd_start. Минимальное значение возвращаемых buffer_size составляет 5000 пакетов TS.

После получения APDU sd_start CICAM способен отправить более одного sd_start_reply. Например, CICAM может отправить первый sd_start_reply с transmission_status, равным 0x00 (готов к приему), и drm_status, установленным в 0x01 (статус в настоящее время не определен), а затем второй

sd_start_reply с transmission_status, равным 0x00 (готов к приему), и drm_status, установленным в 0x00 (дескремблирование возможно). Это позволяет Хосту начать заполнение буфера CICAM с момента получения первой sd_start_reply, в то время как CICAM готовится для дескремблирования семплов, и, следовательно, улучшает условия работы пользователя за счет ускорения представления информации.

Значение buffer_size, установленное в первом sd_start_reply, не может быть изменено CICAM в последующем sd_start_reply.

Синтаксис APDU sd_start_reply должен быть в соответствии с таблицей 33.

Таблица 33 — Синтаксис APDU sd_start_reply

Синтаксис	Количество битов	Мнемоника
sd_start_reply() {		
sd_start_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
transmission_status	8	uimsbf
drm_status	8	uimsbf
drm_system_id	16	uimsbf
drm_uuid	16*8	uimsbf
buffer_size	16	uimsbf
}		

Семантика полей APDU sd_start_reply:

- **sd_start_tag**: поле 24 бита с установленным значением 0x9F9803 идентифицирует этот APDU;
- **length_field**: поле 8 битов содержит длину полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов идентифицирует локальный TS;
- **transmission_status**: поле 8 битов содержит оценку готовности CICAM к приему. Значение 0x00 (готов к приему) указывает, что CICAM готов к приему семплов для дескремблирования. Другие значения указывают, что CICAM не готов к приему. В таблице 34 перечислены возможные значения поля transmission_status;

Таблица 34 — Возможные значения поля transmission_status

transmission_status	Значение
Готовность к приему	0x00
Ошибка — CICAM занят	0x01
Ошибка — другая причина	0x02
Зарезервированы	от 0x03 до 0xFF

- **drm_status**: поле 16 битов содержит статус DRM в CICAM. Таблица 35 содержит возможные значения поля drm_status;

Таблица 35 — Возможные значения поля drm_status

drm_status	Значение
Дескремблирование возможно	0x00
В настоящий момент состояние неопределенное	0x01

Окончание таблицы 35

drm_status	Значение
Ошибка — нет права	0x02
Зарезервированы	от 0x03 до 0xFF

- **drm_system_id**: поле 16 битов идентифицирует систему DRM. CICAM будет использовать этот идентификатор для дескремблирования семплов. Значения drm_system_id идентичны значениям ca_system_id, определенным в распределении идентификаторов и кодов для системы и оборудования DVB [10]. Если drm_system_id не используется для идентификации DRM, то в поле должно быть установлено 0xFFFF;

- **drm_uuid**: UUID из DRM, который CICAM будет использовать для дескремблирования семплов. Если drm_uuid_id не используется для идентификации DRM, все байты этого поля должны быть установлены в 0xFF;

- **buffer_size**: размер буфера CICAM, выделенный для дескремблирования треков семпла, выраженных в числе транспортных пакетов. Если в APDU sd_start был объявлен больше чем один трек семпла, то буфер будет использован треками совместно.

7.4.6 APDU sd_update

Хост должен послать к CICAM APDU sd_update в тех случаях, если необходимо:

- указать на изменения в списке треков семплов, вследствие которых новые семплы должны быть дескремблированы. Этот APDU должен быть отправлен после изменения выбора пользователя, что приводит к изменению в наборе треков семплов для дескремблирования;

- предоставить дополнительные метаданные DRM, которые могут быть необходимы CICAM для продолжения дескремблирования семпла. Как правило, этот APDU может быть отправлен до ротации ключей с тем, чтобы обеспечить необходимые метаданные для CICAM с целью определения ключей, которые доступны для применения после ротации ключей.

Для контента, полученного не в форме транспортного потока, каждый трек семплов, связанный с его track_PID в соответствии с объявлением в одном из предыдущих APDU sd_start или sd_update, должен обрабатываться по следующим правилам:

- CICAM должен продолжать дескремблирование соответствующих семплов, если track_PID находится в списке;

- CICAM не должен останавливать дескремблирование соответствующих семплов и очищать свой буфер от всех пакетов TS с этим PID, если track_PID исчезнет из списка. Хост должен прекратить передачу любых пакетов TS с этим PID перед передачей этого APDU;

- все треки семплов, которые будут дескремблированы, должны быть перечислены в списке, даже если они были сообщены CICAM в предыдущих APDU. Однако в этом случае метаданные, если они не изменились, не должны повторяться;

- CICAM должен подготовиться к дескремблированию, если новый track_PID будет указан в списке и при готовности ответить sd_update_reply.

Для контента TS существует единственный трек семплов, это исключает возможность добавления или удаления трека. Поэтому отсутствует необходимость в идентификаторе трека, и APDU sd_update может быть использован только для предоставления дополнительных метаданных DRM.

Синтаксис APDU sd_update должен быть в соответствии с таблицей 36.

Таблица 36 — Синтаксис APDU sd_update

Синтаксис	Количество битов	Мнемоника
sd_update() {		
sd_update_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
reserved	7	bslbf

Окончание таблицы 36

Синтаксис	Количество битов	Мнемоника
ts_flag	1	bslbf
if (ts_flag == 1){		
number_of_metadata_records	8	uimsbf
for (i=0; i<number_of_metadata_records; i++) {		
drm_metadata_source	8	uimsbf
drm_system_id	16	uimsbf
drm_uuid	128	uimsbf
drm_metadata_length	16	uimsbf
for (i=0; i<N; i++) {		
drm_metadata_byte	8	uimsbf
}		
}		
}		
if (ts_flag == 0){		
number_of_Sample Tracks	8	uimsbf
for (i=0; i<number_of_Sample_Tracks; i++){		
reserved	3	bslbf
Sample_track_PID	13	uimsbf
number_of_metadata_records	8	uimsbf
for (i=0; i<number_of_metadata_records; i++) {		
drm_metadata_source	8	uimsbf
drm_system_id	16	uimsbf
drm_uuid	128	uimsbf
drm_metadata_length	16	uimsbf
for (i=0; i<N; i++) {		
drm_metadata_byte	8	uimsbf
}		
}		
}		
}		
}		

Семантика полей APDU sd_update:

- **sd_update_tag**: поле 24 бита при значении 0x9F9804 идентифицирует этот APDU;
- **length_field**: содержит длину полезной нагрузки APDU в формате BER ASN.1, определенном в [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор обновляемого локального TS;

- **ts_flag**: флаг, установленный в 0b1, указывает, что запрос дескремблирования семпла относится к семплу трека TS. В других случаях флаг устанавливается в 0b0. Ts_flag имеет то же значение, что и в APDU sd_start;

- **number_of_Sample Tracks**: поле 8 битов указывает количество треков семплов, для которых семплы должны быть дескремблированы;

- **Sample_track_PID**: поле 13 битов указывает PID семпла, описанного трека семплов;

- **number_of_metadata_records**: поле 8 битов содержит количество записей метаданных DRM в настоящем цикле;

- **drm_metadata_source**: поле 8 битов указывает источник метаданных в соответствии с таблицей 32;

- **drm_system_id**: поле 16 битов содержит идентификатор DRM, который связывает метаданные. Значения для drm_system_id идентичны значениям для ca_system_id в соответствии с [10]. Если drm_system_id не используется для идентификации DRM, то в этом поле должно быть установлено 0xFFFF;

- **drm_uuid**: поле 128 битов содержит идентификатор UUID DRM для связанных метаданных. Если drm_uuid_id не используется для идентификации DRM, то все байты этого поля должны иметь значение 0xff;

- **drm_metadata_length**: поле 16 битов содержит длину в байтах метаданных DRM;

- **drm_metadata_byte**: метаданные DRM в соответствии с таблицей 32.

7.4.7 APDU sd_update_reply

CICAM направляет Хосту APDU sd_update_reply как ответ на APDU sd_update.

Если APDU sd_update добавляет треки семплов, то Хост может начать передачу пакетов TS, содержащих семплы, соответствующие новым трекам семплов.

Если sd_update предоставляет некоторые дополнительные метаданные DRM, то Хост может начать передачу пакетов TS со скремблированными семплами и с дополнительными метаданными.

Синтаксис APDU sd_update_reply должен быть в соответствии с таблицей 37.

Таблица 37 — Синтаксис APDU sd_update_reply

Синтаксис	Количество битов	Мнемоника
sd_update_reply() {		
sd_update_reply_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
drm_status	8	uimsbf
}		

Семантика полей APDU sd_update_reply:

- **sd_update_reply_tag**: поле 24 бита при значении 0x9F9805 идентифицирует этот APDU;

- **length_field**: поле содержит полезную нагрузку APDU в формате BER ASN.1 в соответствии с [2] (8.3.1);

- **LTS_id**: поле 8 битов содержит идентификатор локального TS;

- **drm_status**: поле 8 битов содержит статус DRM в CICAM. В таблице 35 перечислены его возможные значения.

7.5 Интерфейс транспортного потока

7.5.1 Общие замечания

Трек семплов в составе TS передается непосредственно через интерфейс TS без дополнительной инкапсуляции.

Если треки семплов не относятся к типу TS, то дополнительная сигнализация и инкапсуляция для семплов должны применяться в соответствии с 7.5.3 настоящего стандарта.

7.5.2 Передача контента в транспортном потоке

7.5.2.1 Выход Хоста

В тех случаях, когда трек семплов относится к типу TS, Хост должен посылать пакеты TS исходного трека семплов TS в CICAM через интерфейс TS.

Хост не должен выполнять синтаксический анализ и делать оценки трека семплов TS перед передачей на интерфейс TS. CICAM должен выполнять анализ принятого TS в объеме, включающем:

- извлечение PSI;
- идентификацию PID ECM;
- идентификацию PID для дескремблирования.

Хост может отправить таблицы сброса FLT на comms_PID.

Хост может передать таблицы SET с теми же идентификаторами, которые используются для передачи треков семплов, аналогично тому, как это описано в 7.5.5.3 настоящего стандарта. Таблица SET не должна направляться на comms_PID, так как не гарантируется правильная последовательность поступающих SET по отношению к заключительным данным трека семплов.

Хост не должен отправлять таблицу SST.

7.5.2.2 Выход CICAM

Для дескремблирования трека семплов TS CICAM должен выводить транспортные пакеты в том же порядке, в каком они были получены от Хоста.

CICAM может буферизировать полученные транспортные пакеты.

CICAM должен непрерывно сообщать Хосту об уровне заполнения буфера CICAM в соответствии с 7.5.4.3 настоящего стандарта.

7.5.2.3 Треки семплов нескольких TS

С целью устранения коллизий PID локального TS, сконфигурированного в режиме семпла, должны выполняться следующие условия:

- локальный TS должен содержать не более одного трека семплов TS;
- локальный TS не должен содержать одновременно треки семплов в формате TS и в формате, отличающемся от формата TS.

Одновременное дескремблирование нескольких локальных TS при обработке нескольких потоков в зависимости от поддерживающих функций и доступности ресурса возможно в соответствии с разделом 6 настоящего стандарта.

7.5.3 Передача контента в формате, отличающемся от формата транспортного потока

7.5.3.1 Выход Хоста

7.5.3.1.1 Общие замечания

Когда интерфейс TS конфигурируется в режим семпла, а треки семплов не относятся к формату TS, то TS, отправленные Хостом к CICAM, должны содержать пакеты TS только с идентификаторами, объявленными в APDU sd_start или sd_update.

Каждый трек семплов TS, посланный Хостом через интерфейс TS, должен содержать последовательность семплов, имеющих отношение к указанному треку с присвоенным PID.

7.5.3.1.2 Передача семплов

Перед передачей семпла Хост должен объявить (декларировать) семпл, отправляя таблицу SST в TS и используя PID трека семплов. Хост может повторить SST при условии, что последнее повторение SST предшествует запуску семпла. SST содержит учетные данные, необходимые для дескремблирования семпла [обычно это вектор инициализации (IV) и идентификатор ключа].

Таблица SST содержит tsc_parity_bit, который указывает на биты четности transport_scrambling_control транспортных пакетов, содержащих скремблированные данные объявленного семпла. Хост должен переключать tsc_parity_bit в SST на следующий семпл трека семплов.

Хост начинает передачу семпла после передачи его SST. Если ошибка при передаче не возникает, то Хост должен завершить передачу предыдущего семпла, отправляя SET перед отправкой секции SST следующего семпла того же трека.

После передачи транспортного пакета, содержащего окончание семпла, Хост должен отправить в TS SET с тем же PID, как и трек семплов (track_PID) аналогично изложенному в 7.5.5.3 настоящего стандарта. Хост должен завершить передачу семпла до отправления SET этого семпла.

В любое время Хост может просить CICAM очистить свой буфер семплов, отправляя таблицу FLT в TS, используя значение comms_PID, равное 0x001C, как определено для локальной связи внутриполосной сигнализации в [3] (5.1.3). Хост должен остановить передачу семплов, прежде чем отправить FLT. Сразу после передачи FLT Хост может запустить передачу новых семплов, объявленных в SST.

На рисунке 4 представлена последовательность операций переноса контента при использовании таблиц SST и SET.

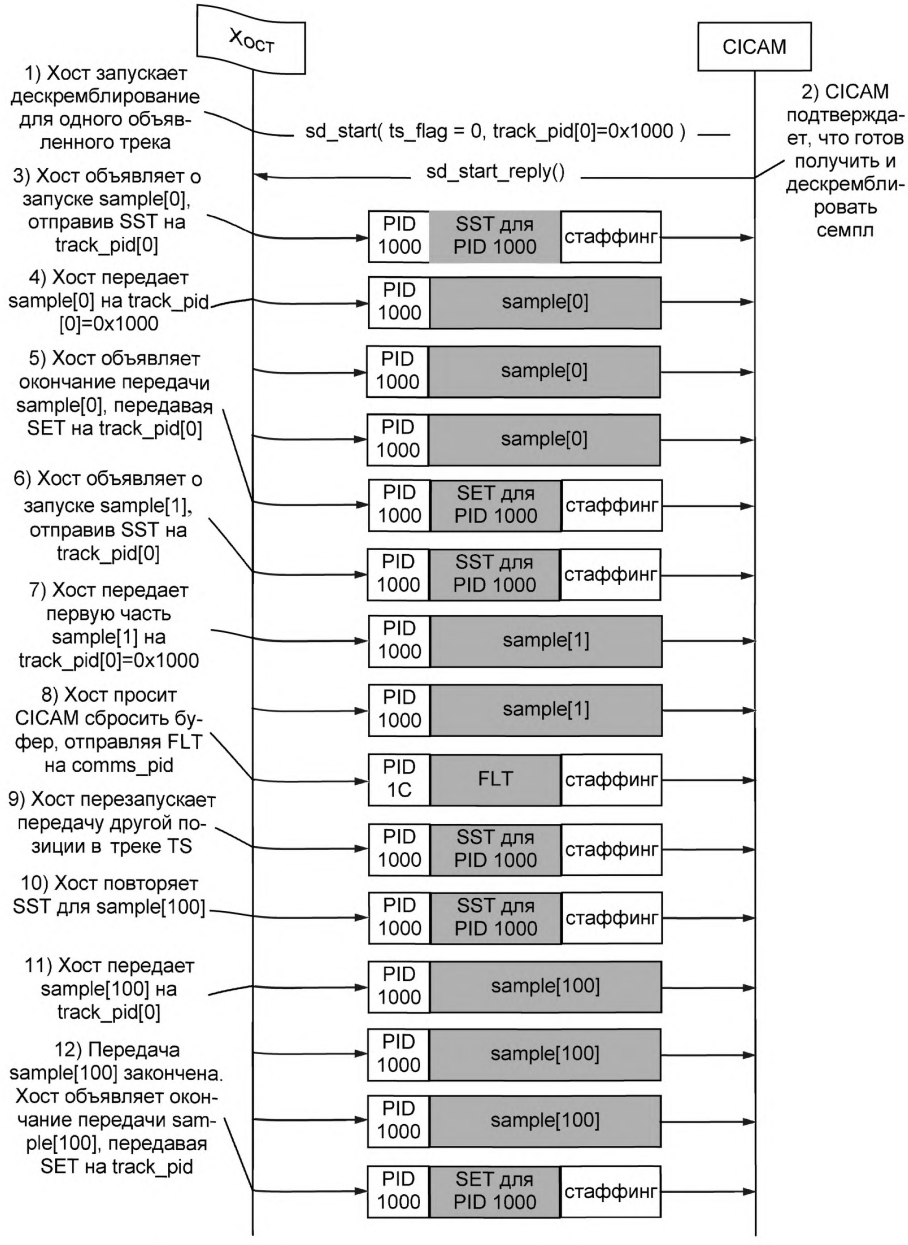


Рисунок 4 — Последовательность операций переноса контента при использовании таблиц SST и SET

Примечание — Нумерация PID шестнадцатиричная.

1) Хост запускает дескремблирование семпла по интерфейсу TS, отправляя APDU sd_start. Есть только один объявленный трек [0] на PID 0x1000;

2) CICAM APDU sd_start_reply подтверждает, что он готов получить и дескремблировать семпл;

3) Хост объявляет о запуске sample[0], отправив SST на track_PID;

4) Хост передает sample[0] на track_PID [0]=0x1000;

5) Передача sample[0] закончена. Хост объявляет окончание передачи sample[0], передавая SET на track_PID;

6) Хост объявляет о запуске sample[1], отправив SST на track_PID;

7) Хост передает первую часть sample[1] на track_PID [0]=0x1000;

8) Хост просит CICAM сбросить буфер, отправляя FLT на comms_PID;

9) Хост перезапускает передачу другой позиции в треке TS, Хост не ожидает получения назад FLT от CICAM. Хост объявляет о запуске sample[100], отправляя SST на track_PID;

10) Хост повторяет SST для sample[100];

11) Хост передает sample[100] на track_PID[0]=0x1000;

12) Передача sample[100] закончена. Хост объявляет окончание передачи sample[100], передавая SET на track_PID.

В любое время Хост может запросить CICAM очистить буфер семпла, отправляя FLT в TS, использованием comms_PID со значением 0x001C. Хост должен остановить передачу семплов прежде, чем отправить FLT. Сразу после передачи FLT Хост может запустить передачу новых семплов, объявленных в SST.

7.5.3.1.3 Управление несколькими треками

При декларировании передачи более одного трека Хост может чередовать семплы из различных треков.

7.5.3.1.4 Обновление списка треков

Хост может обновлять список треков, находящихся в ожидании, отправив APDU sd_update. Если трек добавлен, то Хост не должен отправлять семплы, связанные с этим дополнительным треком, пока от CICAM не получит положительный ответ sd_update_reply. Если ранее декларированный трек будет удален, то Хост должен прекратить отправку семплов, связанных с удаленным треком, перед отправкой APDU sd_update.

7.5.3.2 Заголовки пакетов TS

В пакете, включающем данные семпла, передаваемым от Хоста к CICAM, заголовок пакета битов TS должен формироваться следующим образом:

- в байте синхронизации должен быть установлен LTS_id локального TS;
- в поле transport_error_indicator должен быть установлен 0b0;
- в поле payload_unit_start_indicator пакета, содержащего начало семпла, должен быть установлен 0b1. В противном случае payload_unit_start_indicator должен быть установлен в 0b0;
- в поле pointer_field должен быть установлен 0, когда payload_unit_start_indicator установлен в 0b1, в противном случае он не должен присутствовать;
- в поле transport_priority должен быть установлен 0b0;
- PID должен быть идентификатором соответствующего track_PID трека, как определено в APDU sd_start или sd_update;
- в поле transport_scrambling_control должно быть установлено или 0b10, или 0b11 для пакета, содержащего скремблированные данные семпла. Для данного семпла Хост должен использовать значение transport_scrambling_control в соответствии с tsc_parity_bit, указанным в секции SST соответствующего семпла, как это определено в 7.5.5.5 настоящего стандарта для всех пакетов, содержащих скремблированные данные из указанного семпла. В поле transport_scrambling_control будет установлено 0b00 для пакета, содержащего прозрачные (нескремблированные) данные семпла;
- в поле adaptation_field_control должно быть установлено: или 0b01, когда пакет TS содержит только данные семпла, или 0b11, когда пакет TS содержит поле для заполнения (адаптации), следующее после данных семпла;
- поле continuity_counter должно быть в соответствии с определением системы MPEG-2 [9]. Поле continuity_counter может быть отброшено для пакета, содержащего начало семпла.

Пакет полезной нагрузки TS должен содержать только данные от одного семпла.

Хост должен использовать adaptation_field только для заполнения (стаффинга). Когда используется поле adaptation_field, оно должно содержать только stuffing_byte, все индикаторы и флаги во втором байте adaptation_field должны быть установлены в ноль.

Хост должен запретить использование поля adaptation_field в следующих случаях:

- пакет TS содержит окончание семпла;
- пакет TS содержит прозрачные данные, когда следующий транспортный пакет с тем же track_PID содержит скремблированные данные;
- пакет TS содержит скремблированные данные, когда следующий транспортный пакет с тем же track_PID содержит прозрачные данные.

7.5.3.3 Выход CICAM

Для данного трека семплов CICAM должен выводить транспортные пакеты с track_PID в том же порядке, в котором они были получены. Структура транспортных пакетов не должна изменяться. Исключением являются следующие случаи:

- CICAM дескремблировал полезную нагрузку некоторых транспортных пакетов;
- CICAM повторно скремблировал некоторые из дескремблированных транспортных пакетов: биты пакетов transportscrambling_control должны быть установлены в соответствии с [1] (5.6.1);
- CICAM использовал различные исходные значения continuity_counter. В любом случае CICAM должен использовать в пакетах семпла совместимую последовательность значений continuity_counter.

Порядок транспортных пакетов для различных треков на выходе из CICAM может отличаться от порядка пакетов, посланных Хостом.

CICAM может буферизовать полученные пакеты TS. CICAM должен постоянно информировать Хост об уровне заполнения буфера CICAM, как указано в 7.5.4.3 настоящего стандарта.

После приема FLT CICAM должен избавиться от всех буферизованных пакетов TS, а затем отправить неизмененную FLT обратно на comms_PID.

7.5.3.4 Семплы ISOBMFF

7.5.3.4.1 Секция SST

Для объявления прозрачного семпла, происходящего из файла ISOBMFF, дескриптор цикла секции SST не должен содержать initialization_vector_descriptor или key_identifier_descriptor. Допускается присутствие частных дескрипторов.

Для объявления скремблированного семпла, происходящего из файла ISOBMFF, Хост должен включать один initialization_vector_descriptor и один key_identifier_descriptor в дескрипторе цикла секции SST. Initialization_vector_descriptor должен содержать InitializationVector (вектор инициализации) семпла в байтах iv_data в соответствии с [5] (Приложение I).

Дескриптор key_identifier_descriptor в байтах key_id_data байт должен содержать KID семпла в соответствии с [5] (Приложение I).

7.5.3.4.2 Пакетирование семпла ISOBMFF

Пакетирование семплов ISOBMFF в TS должно выполняться в соответствии с правилами, описанными в 7.5.3.4.1. Схема пакетирования семплов ISOBMFF в пакеты транспортного потока в соответствии с рисунком 5.

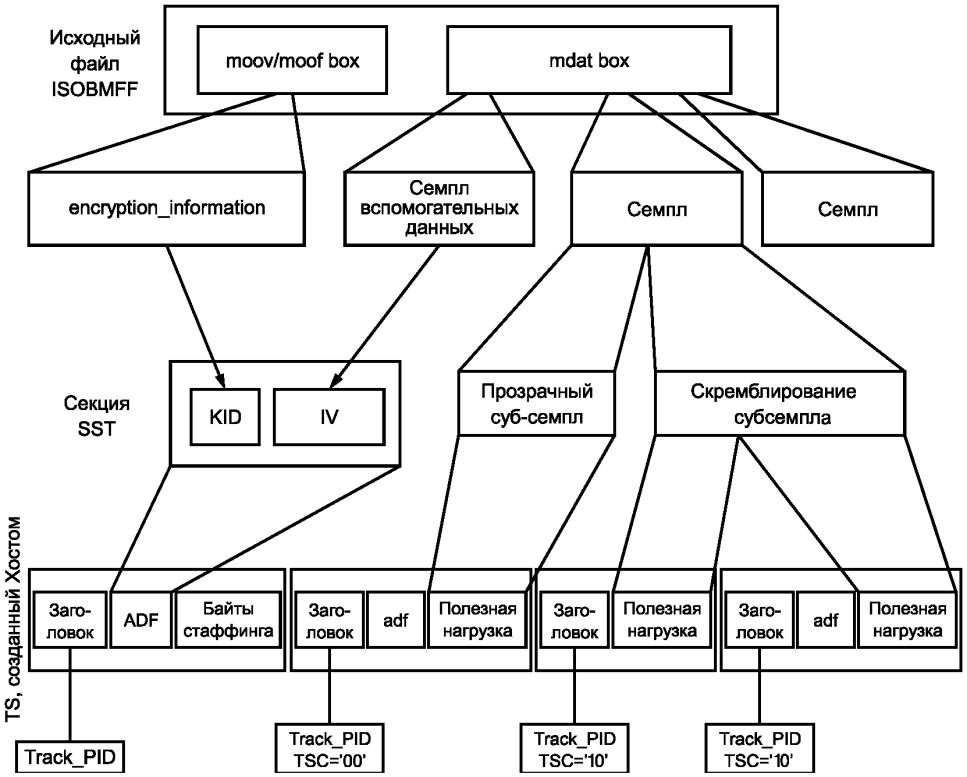


Рисунок 5 — Схема пакетирования семплов ISOBMFF в пакеты транспортного потока

7.5.4 Буферизация CICAM

7.5.4.1 Общие замечания

В случае конфигурирования интерфейса TS в режим семпла CICAM может использовать буфер входящих транспортных пакетов с различной задержкой.

7.5.4.2 Индикация размера буфера

CICAM должен предоставлять полный размер буфера, выделенного для всех треков семплов интерфейса TS, использованием APDU sd_start_reply.

7.5.4.3 Индикация уровня заполнения буфера

CICAM должен периодически предоставлять Хосту индикацию уровня заполнения буфера, вставляя таблицу уровня буфера BLT в TS, отправляемую обратно на Хост. Минимальный период времени между последовательными BLT должен быть 180 миллисекунд. Максимальный период времени между последовательными BLT должен быть 220 миллисекунд. CICAM должен вставлять секции BLT, используя значение comms_PID 0x001C, как установлено для локальной связи в полосе сигнализации в [3] (5.1.3). Информация об уровне буфера, обеспечиваемая CICAM, должна формироваться не менее чем за 100 мс до начала выхода на Хост соответствующей секции BLT.

Рекомендуется, чтобы Хост оценивал буферное пространство в буфере CICAM для локального TS каждый раз, когда принимается секция BLT, для того чтобы гарантировать непереполнение буфера CICAM при передаче на CICAM дополнительных пакетов TS. Хост не должен делать предположения о свободном пространстве буфера в CICAM и не отправлять больший объем данных, чем указано в последнем BLT.

7.5.4.4 Сброс буфера CICAM

Хост может потребовать от CICAM очистить буфер от всех транспортных пакетов, отправив FLT на comms_PID через интерфейс TS. При получении FLT CICAM должен очистить буфер от всех транспортных пакетов, полученных до приема FLT. CICAM подтверждает завершение операции очистки, вернув FLT на comms_PID через интерфейс TS. Хост может начать передачу новых семплов после отправки FLT, не дожидаясь FLT от CICAM (на основе информации, посланной CICAM в последнем полученном BLT).

7.5.5 Коммуникационные таблицы (comms)

7.5.5.1 Общие замечания

Когда интерфейс TS конфигурирован в режим семпла, Хост и CICAM через интерфейс TS обмениваются коммуникационными таблицами (comms) 39—42, представленными в 7.5.5.5—7.5.5.8 настоящего стандарта.

Коммуникационные таблицы 39—42 перед вставкой в пакеты TS должны быть сегментированы на частные секции с применением синтаксиса частной секции, определенного в системе MPEG-2 [9].

Пакеты TS, содержащие коммуникационные таблицы 39—42, не относящиеся к треку, т. е. FLT и BLT, должны использовать значение PID, равное 0x001C, назначенное локальной связью внутриполосной сигнализации в [3] (5.1.3), и должны быть отображены непосредственно в пакеты TS, как указано в 7.5.5.2 настоящего стандарта.

Пакеты TS, содержащие таблицы comms, которые имеют отношение к треку, т. е. SST и SET, должны использовать соответствующий track_PID и должны быть отображены в поле адаптации, как указано в 7.5.5.3 настоящего стандарта.

7.5.5.2 Отображение секций comms в пакеты TS

Секции comms, соответствующие таблицам comms, которые не ассоциируются с треком (FLT и BLT), должны быть отображены непосредственно в пакеты TS. Начало первой секции в полезной нагрузке пакета TS должно указываться полем pointer_field, и таблица comms может запускаться в начале полезной нагрузки пакета TS сразу после pointer_field. В пакете TS не должно быть больше одного pointer_field, а начало другой секции может быть идентифицировано подсчетом длины первой и последующих секций, так как интервалы между секциями в пакете TS синтаксисом недопустимы.

В пакетах TS с одним значением PID секция должна быть закончена, прежде чем будет разрешен запуск следующей секции. Если секция заканчивается до окончания пакета TS, а запуск новой секции невозможен, то должен использоваться механизм стаффинга для заполнения пустого пространства.

Стаффинг должен выполняться заполнением каждого оставшегося байта пакета TS значением 0xFF. Если после последнего байта секции байт сразу же принимает значение 0xFF, то остальная часть пакета TS должна быть заполнена байтами стаффинга, имеющими значения 0xFF.

Стаффинг не должен выполняться использованием механизма adaptation_field.

Более подробное описание механизма и функциональности представлено в [9] (2.4.4 и приложение C).

Для пакета, содержащего часть секции comms, биты заголовка пакета TS должны быть установлены следующим образом:

- в поле байта синхронизации должен быть установлен LTS_id локального TS;
- в поле transport_error_indicator должно быть установлено 0b0;
- в поле payload_unit_start_indicator пакета, содержащего начало секции comms, должно быть установлено 0b1, поле pointer_field должно указывать на первый байт первой секции, запускающейся в пакете TS. В противном случае в поле payload_unit_start_indicator должно быть установлено 0b0;
- в поле transport_priority должно быть установлено 0b0;
- PID должен быть установлен в comms_PID со значением 0x001C для SST и SET, которые направляются в поле адаптации соответствующих медиапотоков пакетов TS;
- в поле transport_scrambling_control должно быть установлено 0b00. Секции comms не должны быть скремблированы;
- в поле adaptation_field_control должно всегда быть установлено 0b01: поле адаптации не должно использоваться для транспортного пакета, содержащего часть секции comms;
- поле continuity_counter должно быть в соответствии с системой MPEG-2 [9].

Пакет TS, содержащий конец секции comms, может содержать начало следующей секции.

Максимальное количество секций в пакете TS ограничивает размер пакета TS.

CICAM должен добавлять секции BLT в comms_PID и может изменять местоположение полученных секций comms в пакетах TS, но CICAM всегда должен генерировать совместимый пакет счетчика непрерывности для всех PID (в том числе comms_PID) на его выходе, за исключением того, что continuity_counter может быть сброшен для пакета TS, содержащего начало семпла.

Примечание — continuity_counter 4-битовое беззнаковое целое поле в транспортном потоке MPEG-2, который сигнализирует текущее значение continuity_counter для этого пакета ID. Каждый раз, когда кодер передает continuity_counter, счетчик для этого пакета ID увеличивается на 1, за исключением случая, когда максимальное значение 15 увеличивается, величина цикла возвращается в 0.

7.5.5.3 Отображение секций comms в полях адаптации пакета TS

Секции comms, соответствующие таблицам comms, имеющим отношение к треку (SST и SET), размещены в поле адаптации пакета TS. Такое размещение позволяет сигнализировать таблицы начала (SST) и окончания (SET) семпла в том же потоке PID, что и сами семплы.

Для пакета, содержащего секции comms, биты заголовка пакета TS должны быть установлены следующим образом:

- в поле байта синхронизации должен быть установлен LTS_id локального TS;
- в поле transport_error_indicator должно быть установлено 0b0;
- в поле payload_unit_start_indicator пакета должно быть установлено 0b0;
- в поле transport_priority должно быть установлено 0b0;
- PID должен соответствовать track_PID, с которым он ассоциирован;
- в поле transport_scrambling_control должно быть установлено 0b00: секции comms не должны быть скремблированы;
- в поле adaptation_field_control должно всегда устанавливаться 0b10: этот пакет не должен содержать data_bytes;
- поле continuity_counter должно соответствовать требованиям системы MPEG-2 [9].

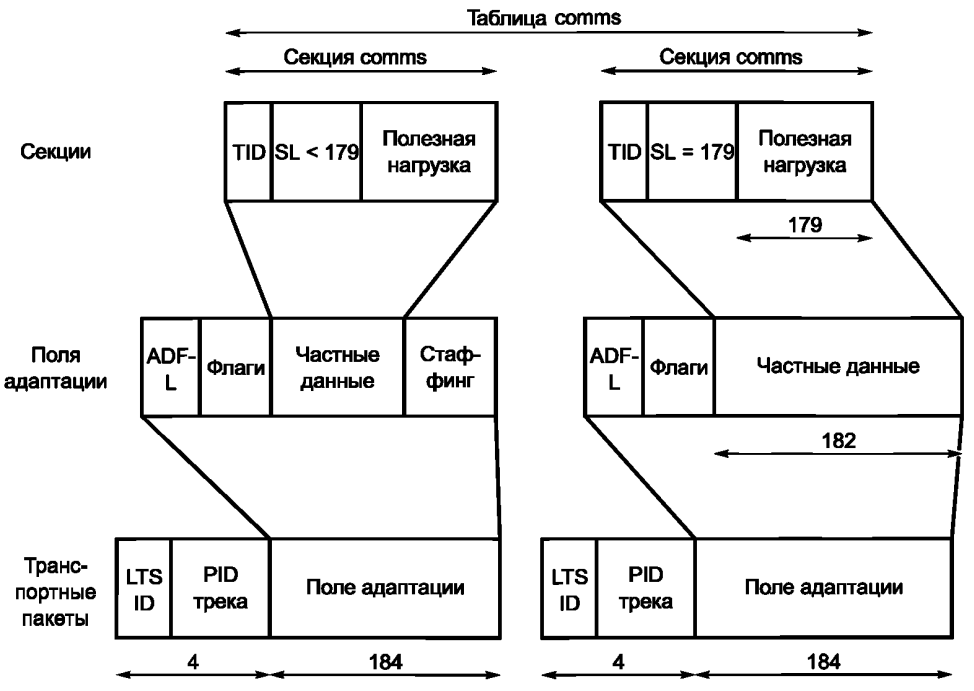
Поле адаптации должно содержать:

- Adaptation_field_length: полная длина поля адаптации. Поскольку пакет TS содержит только поле адаптации, длина поля адаптации должна быть установлена равной 183 байта;
- Discontinuity_indicator: 0b0;
- Random_access_indicator: 0b0;
- Elementary_stream_priority_indicator: 0b0;
- PCR_flag: 0b0;
- OPCR_flag: 0b0;
- Splicing_point_flag: 0b0;
- Adaptation_field_extension_flag: 0b0;
- Transport_private_data_flag: 0b1;
- private_data_byte: первый байт частных данных — первый байт секции comms. Частные данные данного пакета должны содержать всю секцию comms. Секция comms не должна разделяться по нескольким пакетам TS. Таблица comms может быть разделена на последовательные пакеты TS.

Таблица comms, размещаемая в поле адаптации, может содержать дескрипторы, которые показаны в таблице 30 настоящего стандарта, а также дескипторы, определенные в [1]:

- ciplus_initialization_vector_descriptor;
- ciplus_key_identifier_descriptor.

Схема отображения секций comms в полях адаптации пакетов TS показана на рисунке 6.



ADFL — полная длина поля адаптации;
LTS ID — идентификатор LTS;
SL — длина секции, байт;
TID — table_id

Рисунок 6 — Схема отображения секций comms в полях адаптации пакетов TS

7.5.5.4 Кодирование значений table_id

В таблице 38 приведены значения table_id для коммуникационных таблиц. Использование тега 0xFF запрещено из-за возможности неправильной интерпретации байта стаффинга.

Таблица 38 — Значения table_id для коммуникационных таблиц

Идентификатор таблицы table_id	Описание
от 0x00 до 0xCF	зарезервировано
0xD0	sample_start_section
0xD1	sample_end_section
0xD2	flush_section
0xD3	buffer_level_section
от 0xD4 до 0xFE	зарезервировано
0xFF	запрещено

7.5.5.5 Секция таблицы начала семпла

Синтаксис секции SST соответствует синтаксису для частных секций, определенному в системе MPEG-2 [9].

Синтаксис секции SST должен быть в соответствии с коммуникационной таблицей 39.

Таблица 39 — Коммуникационная таблица. Синтаксис секции SST

Синтаксис	Количество битов	Мнемоника
sample_start_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserve_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	3	bslbf
tsc_parity_bit	1	bslbf
descriptor_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
}		

Семантика полей секции SST:

- **table_id**: поле 8 битов, установленное в 0xD0, является идентификатором sample_start_section;
- **section_syntax_indicator**: индикатор 1 бит должен быть установлен в 0b0;
- **section_length**: поле 12 битов определяет количество байтов секции после поля section_length и до конца секции. Максимальная длина секции должна составлять 179 байтов и вписываться в пакет поля адаптации TS, первые 4 бита должны быть установлены в 0b0000;
- **tsc_parity_bit**: поле 1 бит указывает на четность битов transport_scrambling_control, которые будут использоваться для пакета TS, содержащего скремблированные данные соответствующего семпла. Если tsc_parity_bit устанавливается в 0b0, то в заголовке скремблированного пакета TS transport_scrambling_control должен быть установлен 0b10. Если tsc_parity_bit устанавливается в 0b1, то transport_scrambling_control должен быть установлен в 0b11;
- **descriptor_loop_length**: поле 12 битов содержит полную длину применяемых дескрипторов в байтах.

Дескрипторы в SST должны быть в соответствии с таблицей 43. Все дескрипторы должны иметь стандартный формат, включающий тег дескриптора, поле длины, данные дескриптора.

7.5.5.6 Секция таблицы окончания семпла

Синтаксис секции таблицы окончания семпла (SET) для частных секций должен быть в соответствии с [9] и коммуникационной таблицей 40.

Таблица 40 — Коммуникационная таблица. Синтаксис секции SET

Синтаксис	Количество битов	Мнемоника
sample_end_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserve_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf

Окончание таблицы 40

Синтаксис	Количество битов	Мнемоника
reserved	4	bslbf
descriptor_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
}		

Семантика полей секции SET:

- **table_id**: поле 8 битов содержит идентификатор sample_start_section; он должен быть установлен в 0xD1;
- **section_syntax_indicator**: однобитовый индикатор должен быть установлен в 0xb0;
- **section_length**: поле 12 битов определяет количество байтов секции после поля section_length и до конца секции. Максимальная длина секции должна составить 179 байтов и вписываться в пакет поля адаптации TS, первые 4 бита должны быть установлены в 0b0000;
- **descriptor _loop_length**: поле 12 битов содержит полную длину применяемых дескрипторов в байтах.

Дескрипторы в SET должны быть в соответствии с таблицей 43. Все дескрипторы должны иметь стандартный формат, включающий тег дескриптора, поле длины, данные дескриптора.

7.5.5.7 Секция таблицы очистки

Синтаксис секции таблицы очистки (FLT) должен соответствовать синтаксису для частных секций, определенному системой MPEG-2 [9] и коммуникационной таблицей 41.

Таблица 41 — Коммуникационная таблица. Синтаксис секции FLT

Синтаксис	Количество битов	Мнемоника
flush_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserve_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	4	bslbf
descriptor_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
}		

Семантика полей секции FLT:

- **table_id**: поле 8 битов содержит идентификатор flush_section. Он должен быть установлен в 0xD2;
- **section_syntax_indicator**: однобитовый индикатор должен быть установлен в 0xb0;
- **section_length**: поле 12 битов определяет количество байтов секции после поля section_length и до конца секции;
- **descriptor _loop_length**: поле 12 битов содержит полную длину применяемых дескрипторов в байтах.

Дескрипторы в FLT должны быть в соответствии с таблицей 43. Все дескрипторы должны иметь стандартный формат, включающий тег дескриптора, поле длины, данные дескриптора.

7.5.5.8 Секция таблицы уровня заполнения буфера

Синтаксис секции таблицы уровня заполнения буфера (BLT) должен соответствовать синтаксису для частных секций, определенному системой MPEG-2 [9] и коммуникационной таблицей 42.

Таблица 42 — Коммуникационная таблица. Синтаксис секции BLT

Синтаксис	Количество битов	Мнемоника
buffer_level_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserve_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
buffer_level	16	uimsbf
reserved	5	bslbf
buffer_empty_flag	1	bslbf
10%_flag	1	bslbf
20%_flag	1	bslbf
30%_flag	1	bslbf
40%_flag	1	bslbf
50%_flag	1	bslbf
60%_flag	1	bslbf
70%_flag	1	bslbf
80%_flag	1	bslbf
90%_flag	1	bslbf
buffer_full_flag	1	bslbf
}		

Семантика полей секции BLT:

- **table_id**: поле 8 битов содержит идентификатор buffer_level_section, когда он установлен в 0xD3;
- **section_syntax_indicator**: однобитовый индикатор должен быть установлен в 0xb0;
- **section_length**: поле 12 битов определяет количество байтов секции после поля section_length и до конца секции;
- **buffer_level**: поле 16 битов определяет текущий уровень свободного места в буфере, который выражается в количестве транспортных пакетов. Эта мера неиспользуемого пространства в буфере. buffer_level должна соответствовать уровню, измеренному CICAM за 100 мс до передачи этой секции Хосту;
- **buffer_empty_flag**: этот бит должен быть установлен в 0b1, когда буфер пуст, и должен быть установлен в 0b0, когда буфер полон;
- **10 (20, 30, 40, 50, 60, 70, 80, 90) %_flag**: бит этих флагов должен быть установлен в 0b1, когда buffer_level ниже 10 (20, 30, 40, 50, 60, 70, 80, 90) % от максимума, и должен быть установлен в 0b0, когда уровень буфера выше 10 (20, 30, 40, 50, 60, 70, 80, 90) % от максимума;
- **buffer_full_flag**: флаг должен быть установлен в 0b1, когда буфер заполнен, и в 0b0, когда он не заполнен.

7.5.5.9 Дескрипторы

7.5.5.9.1 Общие замечания

В 7.5.5.9 настоящего стандарта определены дескрипторы, которые могут использоваться в таблице comms. Эти дескрипторы могут быть вставлены в любую из таблиц, определенных в 7.5.5.4, 7.5.5.5, 7.5.5.6 и 7.5.5.7 настоящего стандарта, в виде конструкции «дескриптор ()».

Все дескрипторы, определенные в 7.5.5.9 настоящего стандарта, имеют следующую семантику:

- descriptor_tag: тег дескриптора поле 8 битов, однозначно идентифицирует дескриптор. Значения descriptor_tag определены в таблице 43;
- descriptor_length: длина дескриптора поле 8 битов с указанием общего количества байтов блока данных дескриптора, следующего после поля descriptor_length.

Таблица 43 содержит список дескрипторов, представленных в настоящем стандарте, значения тегов дескрипторов и варианты рекомендуемого использования в таблицах comms. Символы «да» в пересечении между дескриптором и типом таблицы показывают, что дескриптор может быть включен в таблицу этого типа.

Таблица 43 — Значения дескрипторов для таблицы comms

Дескриптор	Значение тега	SST	SET	FLT	BLT
ciplus_initialization_vector_descriptor	0xCD	да	—	—	—
ciplus_key_identifier_descriptor	0xCF	да	—	—	—
Зарезервировано	от 0xD0 до 0xEF	—	—	—	—
Определяется Хостом	от 0xF0 до 0xFE	да	да	да	—
Запрещено использовать	0xFF	—	—	—	—

7.5.5.9.2 Дескриптор вектора инициализации CI Plus™

Дескриптор вектора инициализации CI Plus™ Ciplus_initialization_vector_descriptor используется Хостом для предоставления вектора инициализации, связанного со следующим семплом.

Синтаксис дескриптора вектора инициализации CI Plus™ ciplus_initialization_vector_descriptor должен быть в соответствии с таблицей 44.

Таблица 44 — Синтаксис дескриптора вектора инициализации CI Plus™ ciplus_initialization_vector_descriptor

Синтаксис	Количество битов	Мнемоника
ciplus_initialization_vector_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
IV_data_byte	8	uimsbf
}		
}		

Семантика полей вектора инициализации CI Plus™:

- **descriptor_tag**: поле 8 битов определяет тег дескриптора и имеет значение 0xCD;
- **descriptor_length**: поле 8 битов содержит количество байтов IV_data_bytes;
- **IV_data_byte**: поле 8 битов содержит последовательность полей IV_data_byte, включающих IV.

7.5.5.9.3 Дескриптор идентификатора ключа CI Plus™

Дескриптор идентификатора ключа CI Plus™ ciplus_key_identifier_descriptor используется Хостом для предоставления идентификатора ключа контента, связанного со следующим семплом.

Синтаксис дескриптора идентификатора ключа CI Plus™ ciplus_key_identifier_descriptor должен быть в соответствии с таблицей 45.

Таблица 45 — Синтаксис дескриптора идентификатора ключа CI Plus™ ciplus_key_identifier_descriptor

Синтаксис	Количество битов	Мнемоника
ciplus_key_identifier_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
key_id_data_byte	8	uimsbf
}		
}		

Семантика полей дескриптора ciplus_key_identifier_descriptor:

- **descriptor_tag**: поле 8 битов определяет тег дескриптора при значении 0xCF;
- **key_id_data_byte**: поле 8 битов содержит идентификатор ключа.

7.5.6 Информация о правилах использования

CICAM должен связывать информацию о URI с программой дескремблирования сампла по правилам, описанным в [1] (5.7).

CICAM должен использовать program_number, указанный в APDU sd_start, при отправке сообщения URI в Хост.

Хост должен применять URI, полученную от CICAM, в соответствии с [1] (5.7).

В течение периода, когда URI для программы еще неизвестна Хосту, например сразу после того, как было инициировано дескремблирование сампла, Хост должен использовать начальную URI по умолчанию, как определено в [1] (5.7).

8 CICAM в режиме проигрывателя контента IP

8.1 Общие замечания

Параметры доставки данных IP через интерфейс команд нормированы в [1]. Однако этот вариант доставки имеет ограниченную пропускную способность. Для увеличения скорости передачи данных для доставки видеоданных настоящий стандарт нормирует параметры передачи данных IP от Хоста к CICAM через интерфейс TS. Восходящий поток (Upstream) данных IP в сеть отправляется через интерфейс команд.

В некоторых реализациях Хоста поддержка приема и деинкапсуляции данных, полученных по IP, не поддерживается. В этих вариантах реализации Хост предоставляет ресурсы, позволяющие CICAM получать данные полезной нагрузки UDP/TCP через стек IP Хоста по интерфейсу TS. Этот режим будет использоваться в тех случаях, когда Хост не поддерживает транспортные механизмы UDP/TCP. Далее в стандарте этот режим работы обозначается как гибридный.

Данные, возвращенные CICAM, должны быть в формате SPTS, который Хост в режиме проигрывателя может использовать непосредственно.

При доставке IP на CICAM в режиме проигрывателя по требованию Хоста должен выполнять операции транскодирования или создания водяных знаков. При выполнении этих операций семплы на входе и выходе CICAM будут иметь различные размеры. Эти операции при доставке IP Хост в режимах проигрывателя или сампла выполнять не может.

CICAM может применить управления скремблированием CI Plus™ элементарных потоков, созданных CICAM в режиме проигрывателя.

Если информация о дате и времени от вещательного тюнера Хосту и CICAM недоступна, то Хост и CICAM могут получать информацию о дате и времени через сетевой интерфейс IP. Метод получения информации о дате и о времени через сетевой интерфейс IP настоящий стандарт не определяет.

Хост не может выполнять услуги игнорирования, определенные в [1] (раздел 10), для служб, предоставленных через IP CICAM в режиме проигрывателя.

8.2 Управление проигрывателем

Для управления потоком данных CICAM должен иметь данные о времени выполнения конкретных действий пользователя.

Для CICAM может возникнуть необходимость запрета использования отдельных средств управления проигрывателем, если они не поддерживаются CICAM или по условиям формата потока. Способ, при котором CICAM получает средства для управления проигрывателем, зависит от метода инициирования проигрывания.

Если проигрывание инициирует CICAM, то предполагается, что CICAM запускает приложение CICAM AppMMI и что приложение будет использоваться для захвата ключей пользователя.

Если проигрывание инициирует Хост, то для обработки актива службы или контента Хост запрашивает ресурс у CICAM, использует ресурс CICAM в режиме проигрывателя и сообщает CICAM о действиях пользователя.

8.3 Инициализация сеанса проигрывания

8.3.1 Общие замечания

Потребление службы при доставке IP CICAM в режиме проигрывателя может инициировать CICAM использованием приложения CICAM или это инициирование может выполнить Хост через канал IP.

8.3.2 Проигрывание, инициируемое Хостом

В режиме проигрывания, инициируемого Хостом, пользователь передает Хосту запрос на проигрывание контента. Хост при использовании собственных ресурсов не может выполнить этот запрос и производит проверку CICAM на возможность выполнить проигрывание контента. Это достигается передачей APDU **CICAM_player_verify_req**. Такая проверка может быть выполнена Хостом и на более раннем этапе, например во время установки профиля каналов IP, используя механизм, описанный в 15.4 настоящего стандарта, при открытии и обновлении служб в соответствии с 15.4.2 настоящего стандарта.

Если CICAM подтверждает, что он может выполнить запрос проигрывания контента, то Хост при помощи APDU **CICAM_player_play_req** запрашивает CICAM о возможности проигрывания службы. После получения этого запроса CICAM должен отправить к Хосту APDU **CICAM_player_start_req**, сообщая PMT в SPTS и скорость, необходимую для передачи данных, через интерфейс TS. Хост должен ответить APDU **CICAM_player_start_reply**, в котором сообщается о локальном TS, который будет использоваться для передачи данных для этого сеанса проигрывателя.

CICAM после этого устанавливает необходимое количество соединений гибридных LSC, как определено в разделе 10 настоящего стандарта, с использованием выделенного LTS_id для доставки данных AV.

В дополнение к гибридным соединениям CICAM может также установить дополнительные соединения для управления, например, использованием протокола RTSP. Эти соединения должны выполняться в рамках нового сеанса LSC.

CICAM принимает данные, расшифровывает их и преобразует в локальный TS (SPTS) и, прежде чем отправить локальный TS через интерфейс TS в Хост, может (опционально) скремблировать его, используя управление скремблированием контента CI Plus™.

Во время проигрывания контента команды проигрывателя могут быть направлены Хостом на CICAM для управления проигрыванием.

Когда Хост прекращает сеанс проигрывателя, он должен отправить к CICAM APDU **CICAM_player_stop**. CICAM в качестве подтверждения направляет Хосту APDU **CICAM_player _end** с сообщением о том, что CICAM прекратил проигрывание. Хост не освобождает занятые ресурсы для этого сеанса проигрывателя до окончания получения APDU **CICAM_player_end.** Интерфейс TS затем возвращается в нормальный режим как один из режимов TS, описанных в 7.2 настоящего стандарта.

8.3.3 Проигрывание, инициируемое CICAM

CICAM может установить сеанс Хоста проигрывателя в ответ на событие, которое настоящий стандарт не определяет. Установление такого сеанса выполняется передачей Хосту APDU CICAM_player_start_req, содержащего PMT в SPTS и скорость данных, необходимые для передачи данных через интерфейс TS, для включения этого запроса потребления контента. Хост отвечает APDU CICAM_player_start_reply, который сообщает локальный TS, который будет использоваться для передачи данных через интерфейс TS для этого сеанса проигрывателя.

CICAM устанавливает LSC гибридные соединения, как правило, с использованием выделенного LTS_id для доставки данных AV.

В случае, когда CICAM осуществляет прямые извлечения данных IP без использования ресурса LSC, CICAM может не устанавливать LSC гибридные соединения, однако во всех других отношениях взаимодействия между Хостом и CICAM одинаковы.

CICAM запрашивает данные, расшифровывает и преобразует входной SPTS в локальный TS, и может скремблировать пакеты TS, используя скремблирование управления контентом CI Plus™, перед отправкой Хосту локального TS через интерфейс TS. В процессе потребления контента приложение режима спецэффектов может взаимодействовать с CICAM в режиме проигрывателя для управления режимами проигрывания в соответствии с протоколом, определение которого выходит за рамки настоящего стандарта. Когда CICAM будет завершать сеанс проигрывателя, он отправит Хосту APDU CICAM_player_session_end (0) о прекращении проигрывания. Этот APDU заставляет Хост освободить ресурсы, занятые для этого сеанса проигрывателя, и вернуть интерфейс TS в нормальный режим.

8.4 Обработка ошибок обмена

При обнаружении Хостом ошибки обмена на сетевом интерфейсе IP он должен передать эту информацию к CICAM через интерфейс команд использованием APDU LSC comms_reply, как определено в [2] (8.7.1.5). Ошибки сервера, объявленные через протоколы более высоких уровней протоколов TCP или UDP (например, HTTP по TCP), должны быть возвращены как пакеты IP, инкапсулированные в TS через интерфейс TS в гибридном режиме.

8.5 Поддержка режима спецэффектов

В тех случаях, когда CICAM поддерживает формат контента, протокол доставки и режим спецэффектов, CICAM может управлять потоком данных и контентом для поддержки режима спецэффектов.

CICAM должен продолжать выводить совместимый TS через интерфейс TS во время сеанса проигрывателя. Например, в случае проигрывания при ускоренной перемотке вперед CICAM может выводить последовательность I-кадров одновременно с аудиокадрами, содержащими в фонограмме паузу. В случае команды «пауза» CICAM может продолжать отправлять локальный TS, содержащий повторения последнего видеокадра, наряду с аудиокадрами, содержащими в фонограмме паузу.

Точные характеристики и метод управления режимом спецэффектов CICAM в настоящем стандарте не определяются.

8.6 Завершение сеанса

8.6.1 Общие замечания

Сеанс проигрывателя может быть прекращен по любой из причин, указанных в 8.6.2, 8.6.3, 8.6.4 настоящего стандарта. После окончания сеанса Хосту может потребоваться повторная инициализация его функций декодирования и визуализации AV, чтобы избежать нежелательных артефактов, вызванных потерей входного потока.

8.6.2 Неисправимая ошибка

В том случае, когда CICAM обнаружил ошибку, приводящую к необходимости завершения сеанса проигрывателя, CICAM должен сигнализировать пользователю о деталях ошибки, например через приложение MMI, и направлять к Хосту APDU CICAM_player_ end.

Правила поведения в случае обнаружения неисправимой ошибки определяет инициатор сеанса: Хост или CICAM.

8.6.3 Завершение сеанса пользователем

Сеанс проигрывателя, инициированный Хостом, может быть завершен пользователем в любое время, например, при смене канала. Всякий раз когда Хост хочет завершить сеанс проигрывателя, он должен отправить к CICAM APDU CICAM_player_stop, в ответ на который CICAM должен завершить сеанс, отправляя APDU CICAM_player_end.

Хост не должен переключаться на новый канал в локальном TS сеанса проигрывателя, пока не получит для сеанса CICAM APDU CICAM_player_end.

В случае сеанса, инициируемого CICAM, замена канала пользователем должна привести к передаче Хостом к CICAM APDU CICAM_player_stop, в ответ на который CICAM должен завершить сеанс.

Хост не должен отправлять к CICAM APDU ca_pmt, сигнализирующий об изменении канала, пока он не получил от CICAM APDU CICAM_player_end для сеанса проигрывателя.

Хост не должен использовать этот локальный TS для передачи данных AV любого нового канала и не должен отправлять к CICAM ни APDU ca_pmt, ссылающийся на этот локальный TS, сигнализиру-

ющий об изменении канала, ни APDU sd_start, пока не получит для сеанса от CICAM APDU CICAM_player_end.

8.6.4 Окончание контента

Если в инициированном Хостом сеансе CICAM проигрывателя будет достигнуто окончание актива контента, то CICAM должен сигнализировать Хосту об окончании контента, используя APDU CICAM_player_asset_end.

Когда Хост получает APDU CICAM_player_asset_end, то он должен дать CICAM команду завершения сеанса проигрывателя, отправляя к CICAM APDU CICAM_player_stop.

Процедуры при окончании контента в инициированном CICAM сеансе CICAM в режиме проигрывателя настоящим стандартом не определяются.

8.7 Ресурс CICAM в режиме проигрывателя

8.7.1 Общие замечания

Ресурс CICAM в режиме проигрывателя позволяет Хосту предложить CICAM инициировать сеанс проигрывателя и проигрывать службу от имени Хоста. Это позволяет CICAM по своей инициативе запрашивать инициирование сеанса проигрывателя. Ресурс CICAM проигрывателя обеспечивается Хостом. CICAM может запросить сеанс к ресурсу CICAM в режиме проигрывателя, только если Хост объявил о ресурсе CICAM проигрывателя во время передачи протокола администратора ресурсов в соответствии с [2] (8.4.1.1). Хост может поддерживать ресурсы нескольких сеансов CICAM проигрывателя.

Следующие 8.7.2—8.7.19 настоящего стандарта определяют APDU, используемые Хостом для управления потреблением контента CICAM.

Примечание — Термин «сеанс проигрывателя» используется для описания сеанса проигрывания контента при поставке IP CICAM в режиме проигрывателя. Это не связано с понятием сеанса ресурса CI Plus™.

8.7.2 APDU ресурсов CICAM в режиме проигрывателя

В таблице 46 перечислены APDU ресурса CICAM в режиме проигрывателя и приведены краткие описания назначения каждого APDU.

Таблица 46 — APDU ресурса CICAM в режиме проигрывателя

APDU	Направление	Назначение APDU
CICAM_player_verify_req	Хост → CICAM	Запрос CICAM о возможности употребления контента
CICAM_player_verify_reply	CICAM → Хост	Сообщение CICAM Хосту о способности употребления контента
CICAM_player_capabilities_req	CICAM → Хост	Запрос о типах кодеков, поддерживаемых Хостом
CICAM_player_capabilities_reply	Хост → CICAM	Типы поддерживаемых Хостом кодеков
CICAM_player_start_req	CICAM → Хост	Запрос о начале сеанса проигрывания
CICAM_player_start_reply	Хост → CICAM	Ответ на запрос начала сеанса проигрывания
CICAM_player_play_req	Хост → CICAM	Запрос Хоста о проигрывании элемента контента
CICAM_player_status_error	CICAM → Хост	Сигнализация об обнаружении ошибки
CICAM_player_control_req	Хост → CICAM	Поручение CICAM установить позицию проигрывания текущего сеанса и/или изменить скорость проигрывания
CICAM_player_info_req	Хост → CICAM	Запрос скорости проигрывания, позиции и длительности проигрывания
CICAM_player_info_ reply	CICAM → Хост	Информацию о продолжительности проигрывания контента, позиции и скорости проигрывания
CICAM_player_stop	Хост → CICAM	Поручение CICAM для завершения сеанса проигрывания
CICAM_player_end	CICAM → Хост	Сеанс проигрывания свободен
CICAM_player_asset_end	CICAM → Хост	CICAM уведомляет Хост о том, что был достигнут конец актива контента

Окончание таблицы 46

APDU	Направление	Назначение APDU
CICAM_player_update_req	CICAM → Хост	CICAM запросит предоставить обновленные компоненты
CICAM_player_update_reply	Хост → CICAM	Хост отвечает на запрос обновления проигрывателя

8.7.3 APDU CICAM_player_verify_req

Хост может передать этот APDU в CICAM для проверки способности CICAM проигрывания актива контента при IP-доставке в случае работы CICAM в режиме проигрывателя. CICAM должен ответить APDU CICAM_player_verify_reply. Синтаксис APDU CICAM_player_verify_req показан в таблице 47.

Таблица 47 — Синтаксис APDU CICAM_player_verify_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_verify_req() {		
CICAM_player_verify_req_tag	24	uimsbf
length_field()		
service_location_length	16	uimsbf
for (i=0; i<service_location_length; i++) {		
service_location_byte	8	
}		
}		

Семантика полей APDU CICAM_player_verify_req:

- **CICAM_player_verify_req_tag**: поле 24 бита содержит значение тега, установленное на 0x9FA000;
- **service_location_length**: поле 16 битов содержит количество байтов, формирующих поле ServiceLocation;
- **service_location_byte**: поле 8 битов содержит структуру данных XML одиночного элемента ServiceLocation, описывающего службу, проверяемую Хостом. Схема элемента XML ServiceLocation специфицируется в приложении Г настоящего стандарта.

8.7.4 APDU CICAM_player_verify_reply

CICAM должен отправить этот APDU Хосту в ответ на APDU CICAM_player_verify_req. Этим APDU он указывает, что CICAM способен воспроизводить контент IP.

Синтаксис APDU CICAM_player_verify_reply показан в таблице 48.

Таблица 48 — Синтаксис APDU CICAM_player_verify_reply

Синтаксис	Количество битов	Мнемоника
CICAM_player_verify_reply() {		
CICAM_player_verify_reply_tag	24	uimsbf
length_field()= 1		
player_verify_status	8	uimsbf
}		

Семантика полей APDU CICAM_player_verify_reply:

- **CICAM_player_verify_reply_tag**: поле 24 бита содержит значение тега, установленное на 0x9FA001;

- **player_verify_status**: поле 8 битов содержит возвращаемый Хосту статус проигрывателя на интервале сеанса проигрывателя. В таблице 49 перечислены возможные значения player_verify_ status.

Таблица 49 — Значения player_verify_status

Program_start_status	Значение
Проигрывание службы возможно	0x00
Ошибка — проигрывание службы невозможно	0x01
Зарезервировано	от 0x02 до 0xFF

8.7.5 APDU CICAM_player capabilities_req

CICAM направляет этот APDU Хосту с запросом о наборе кодеков, поддерживаемых Хостом. Эта информация будет использована CICAM для определения необходимости транскодирования исходного контента.

Синтаксис APDU CICAM_player_play_ capabilities_req показан в таблице 50.

Таблица 50 — Синтаксис APDU CICAM_player_play_ capabilities_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_capabilities_req() {		
CICAM_player_capabilities_req_tag	24	uimsbf
length_field()= 0		
}		

Семантика APDU CICAM_player_play_ capabilities_req:

CICAM_player_capabilities_req_tag: поле 24 бита содержит значение тега, установленное на 0x9FA002.

8.7.6 APDU CICAM_player_capabilities_reply

Хост должен отправить этот APDU к CICAM в ответ на APDU CICAM_player_ capabilities_req, чтобы сообщить о кодеках, поддерживаемых Хостом. Эта информация используется CICAM для определения необходимости транскодирования исходного контента. Синтаксис APDU CICAM_player_capabilities_ reply показан в таблице 51.

Таблица 51 — Синтаксис APDU CICAM_player_capabilities_reply

Синтаксис	Количество битов	Мнемоника
CICAM_player capabilities_reply() {		
CICAM_player capabilities_reply_tag	24	uimsbf
length_field()		
number_of_component_types	16	uimsbf
for (i=0; i<number_of_component_types; i++) {		
stream_content	4	uimsbf
Reserved	4	bslbf
component_type	8	uimsbf
}		
}		

Семантика полей APDU CICAM_player_capabilities_reply:

- **CICAM_player_player_capabilities_reply_tag**: поле 24 бита содержит значение тега, установленное на 0x9FA003;
- **number_of_component_types**: поле 16 битов содержит количество поступающих компонент на входе, которые Хост может обрабатывать;
- **stream_content**: поле 4 бита определяет тип потока (например, видео, аудио или данные EBU). Кодирование этого поля должно быть в соответствии с правилом кодирования дескриптора компонент, определенным в [3];
- **component_type**: поле 8 битов определяет тип видео, аудио или данных EBU. Кодирование этого поля должно быть в соответствии с правилом кодирования дескриптора компонент, определенным в [3].

8.7.7 APDU CICAM_player_start_req

CICAM сообщает Хосту о начале сеанса проигрывателя, передавая поля, содержащие:

- необходимые скорости передачи данных через интерфейс TS от Хоста к CICAM и от CICAM к Хосту;
- флаг, характеризующий тип сеанса канала или без поддержки временного сдвига пакетов TS или канала VOD с поддержкой временного сдвига пакетов TS;
- количество байтов, формирующих PMT;
- таблицу PMT.

APDU может содержать PMT с описаниями компонентов, которые будут поставлены на CICAM от Хоста через интерфейс TS. Перед PMT в TS должна быть передана PAT. Если APDU содержит PMT, то она должна быть идентична PMT в TS.

Если во время передачи APDU CICAM_player_start_req PMT CICAM неизвестна, то CICAM будет ожидать PMT, которая будет определена до начала поставки TS. Когда PMT известна, CICAM должен информировать Хост передачей APDU CICAM_player_update_req и ожидать от Хоста APDU CICAM_player_update_reply. После того как Хост получит PMT, CICAM может начать передачу TS с гарантией корректной интерпретации Хостом пакетов TS с самого начала доставки локальных TS.

Хост для этого сеанса должен выделить локальный TS и должен переключиться на режим ввода. Если значение поля input_max_bitrate не будет равно нулю (т. е. CICAM хочет получить данные от Хоста на этом локальном TS), то CICAM должен установить гибридное соединение LSC, используя APDU comms_cmd в соответствии с разделом 10 настоящего стандарта.

CICAM должен выбрать соответствующие значения PID для выходных компонентов и для PMT. Требования идентичности этих PID и PID, выделенных для гибридных соединений LSC, не предъявляются.

Синтаксис APDU CICAM_player_start_req представлен в таблице 52.

Таблица 52 — Синтаксис APDU CICAM_player_start_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_start_req() {		
CICAM_player_start_req_tag	24	uimsbf
length_field()		
input_max_bitrate	16	uimsbf
output_max_bitrate	16	uimsbf
linearChannel	1	bslbf
reserved	7	bslbf
PMT_length	16	uimsbf
for(int i=0; i<PMT_length; i++) {		
PMT_byte	8	bslbf
}		
}		

Семантика полей APDU CICAM_player_update_req:

- **CICAM_player_start_req_tag**: поле 24 бита определяет тег APDU и имеет значение 9FA004;
- **input_max_bitrate**: поле 16 битов определяет скорость передачи, которую CICAM запрашивает на поставку обрабатываемых данных через интерфейс TS от Хоста к CICAM, с шагом 10 Кбит/с, округленных до следующей границы 10 Кбит/с. Например, скорость передачи 512 Кбит/с будет кодирована как 0x0034;
- **output_max_bitrate**: поле 16 битов определяет скорость передачи, которую CICAM запрашивает на поставку обрабатываемых данных через интерфейс TS от CICAM к Хосту с шагом 10 Кбит/с;
- **linearChannel:** флаг, установленный в 0b1, сигнализирует сеанс канала без возможности смещения во времени. При установке в 0b0 он сигнализирует о сеансе актива VOD или канала со сдвигом во времени;
- **PMT_length**: поле 16 битов определяет число байтов, составляющих PMT;
- **PMT_byte**: поле 8 битов содержит таблицу PMT MPEG, где первый байт содержит table_id PMT.

8.7.8 APDU CICAM_player_start_reply

Хост должен переслать этот APDU как ответ на APDU CICAM_player_start_req. Хост должен предоставить идентификатор локального TS, выделенного для доставки данных через интерфейс TS для этого сеанса. Этот идентификатор является также уникальным идентификатором сеанса CICAM проигрывателя.

До отправки этого APDU Хост должен определить необходимость создания нового локального TS. Если Хост определил для использования локальный TS, он должен прекратить передачу данных на этот локальный TS перед отправкой этого APDU. Хост может не передавать TS или отправлять нулевые пакеты до тех пор, пока не будет выполнено гибридное соединение LSC. В этом случае Хост должен использовать этот локальный TS для передачи данных IP в CICAM.

При приеме этого APDU CICAM должен отбросить любые данные TS на соответствующем локальном TS, полученные перед приемом этого APDU, и может начать поставлять SPTS Хосту, используя идентификатор локального TS, предоставленный Хостом.

Синтаксис APDU CICAM_player_start_reply показан в таблице 53.

Таблица 53 — Синтаксис APDU CICAM_player_start_reply

Синтаксис	Количество битов	Мнемоника
CICAM_player_start_reply() {		
CICAM_player_start_reply_tag	24	uimsbf
length_field()=2		
LTS_id	8	uimsbf
input_status	8	uimsbf
}		

Семантика полей APDU CICAM_player_start_reply:

- **CICAM_player_start_reply_tag**: поле 24 бита определяет тег APDU и имеет значение 9FA005;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS для проигрывателя, установленного Хостом. Он также используется для идентификации сеанса проигрывателя;
- **input_status**: поле 8 битов сообщает статус Хоста. Возможные значения статуса приведены в таблице 54.

Таблица 54 — Значения input_status

input_status	Значение
OK — локальный TS переключился на режим ввода	0x00
Отказ запроса	0x01
Слишком низкая доступная скорость передачи	0x02

Окончание таблицы 54

input_status	Значение
Отсутствуют доступные сеансы проигрывателя	0x03
Зарезервирован	от 0x04 до 0xFF

Если в поле input_status установлено значение, отличное от нуля, то поле LTS_id должно быть проигнорировано.

8.7.9 APDU CICAM_player_play_req

Хост направляет этот APDU к CICAM с запросом к CICAM о проигрывании службы от имени Хоста. CICAM должен ответить или APDU CICAM_player_status_error в случае, когда обнаружена ошибка, или APDU CICAM_player_start_req.

Синтаксис APDU CICAM_player_play_req должен быть в соответствии с таблицей 55.

Таблица 55 — Синтаксис APDU CICAM_player_play_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_play_req() {		
CICAM_player_play_req_tag	24	uimsbf
length_field()		
service_location_length	16	uimsbf
for (i=0;i<service_location_length;i++) {		
service_location_byte	8	uimsbf
}		
}		

Семантика полей APDU CICAM_player_play_req:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA006;
- **service_location_length**: поле 16 битов, содержит значение длины поля ServiceLocation;
- **service_location_byte**: поле 8 битов, формирует структуру данных XML с одиночным элементом ServiceLocation, описывающим службу, проверяемую Хостом. Схема XML ServiceLocation определена в приложении Г настоящего стандарта.

8.7.10 APDU CICAM_player_status_error

CICAM направляет этот APDU к Хосту в ответ на APDU CICAM_player_play_req, если обнаружена ошибка. В случае отсутствия ошибки CICAM направляет APDU CICAM_player_start_req.

Если обнаружена ошибка, то CICAM может отправить APDU CICAM_player_play_req в любое время сеанса проигрывателя или после сеанса проигрывателя, если обнаруживается состояние ошибки.

Отображение сообщения об ошибке выполняет CICAM с помощью приложения MMI.

CICAM направляет APDU CICAM_player _end для Хоста с предложением освободить ресурсы Хоста в режиме проигрывателя.

Синтаксис APDU CICAM_player_status_error показан в таблице 56.

Таблица 56 — Синтаксис APDU CICAM_player_status_error

Синтаксис	Количество битов	Мнемоника
CICAM_player_status_error() {		
CICAM_player_status_error_tag	24	uimsbf
length_field() = 3		

Окончание таблицы 56

Синтаксис	Количество битов	Мнемоника
reserved	7	uimsbf
valid_LTS_id	1	uimsbf
LTS_id	8	
player_status	8	
}		

Семантика полей APDU APDU CICAM_player_status_error:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA007;
- **valid_LTS_id**: флаг устанавливается в 0b1, когда сообщение об ошибке относится к установленному сеансу проигрывателя с известным LTS_id. Если установлено значение 0b0, сообщение об ошибке относится к сеансу проигрывателя, который еще не установлен;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, используется сеансом проигрывателя. Он также используется для однозначной идентификации сеанса проигрывателя. Это поле не определено, если valid_LTS_id установлен в 0b0;
- **player_status**: поле 8 битов содержит возвращаемый статус сеанса проигрывателя. В таблице 57 перечислены возможные значения.

Таблица 57 — Значения play_status

play_status	Значение
Зарезервировано	0x00
Ошибка — проигрывание контента невозможно (например, не поддерживается формат контента или протокол)	0x01
Ошибка — неисправимая ошибка	0x02
Ошибка — контент заблокирован (например, нет лицензии контента)	0x03
Зарезервирован	от 0x04 до 0xFF

8.7.11 APDU CICAM_player_control_req

Хост должен отправить этот APDU в CICAM для того, чтобы установить позицию проигрывания или изменение скорости на сеансе проигрывателя. CICAM должен ответить APDU CICAM_player_info_reply, который сигнализирует нужную позицию и скорость.

Синтаксис APDU CICAM_player_control_req показан в таблице 58.

Таблица 58 — Синтаксис APDU CICAM_player_control_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_control_req() {		
CICAM_player_control_req_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
Command	8	uimsbf
if(command == 0x01) {		
seek_mode	8	uimsbf
seek_position	32	tcimsbf

Окончание таблицы 58

Синтаксис	Количество битов	Мнемоника

} else if(command == 0x02) {
Speed
}
}
	16	tcimsbf

Семантика полей APDU CICAM_player_control_req:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA008;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя. Он также используется для идентификации сеанса проигрывателя;
- **command**: поле 8 битов определяет действие, которое должно выполняться в соответствии со значениями, указанными в таблице 59;

Таблица 59 — Значения command

command	Значение
Зарезервировано	0x00
Установить позицию	0x01
Установить скорость	0x02

- **seek_mode**: поле 8 битов, интерпретация его значений должна быть в соответствии с таблицей 60;

Таблица 60 — Значения seek_mode

seek_mode	Значение
Абсолютное значение	0x00
Значение по сравнению с текущей позицией	0x01

- **seek_position**: поле 12 битов содержит целое число со знаком, представляющее искомую позицию проигрывания, выраженную в миллисекундах. Значение 0xFFFFFFFF для потока в реальном времени означает «переход в реальное время», и для VOD это означает конец актива контента;
- **speed**: поле 8 битов содержит данные о скорости проигрывания, выраженной в сотых долях номинальной скорости. Отрицательная величина означает, что запрашивается скорость обратного направления.

Если CICAM не поддерживает запрос скорости или позиции проигрывания, то CICAM должен выбрать ближайшую скорость или позицию проигрывания, которые он поддерживает, или же CICAM может игнорировать команду. Поэтому APDU CICAM_player_info_reply должен оцениваться для определения фактической скорости и позиции проигрывания.

8.7.12 APDU CICAM_player_info_req

Хост должен отправить этот APDU к CICAM для того, чтобы получить информацию о проигрываемом контенте, в том числе о длительности проигрывания контента, о позиции и о скорости проигрывания. CICAM должен ответить APDU CICAM_player_info_reply.

Синтаксис APDU CICAM_player_info_req должен быть в соответствии с таблицей 61.

Таблица 61 — Синтаксис APDU CICAM_player_info_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_info_req() {		
CICAM_player_info_req_tag	24	uimsbf

Окончание таблицы 61

Синтаксис	Количество битов	Мнемоника
length_field() = 1		
LTS_id	8	uimsbf
}		

Семантика полей APDU CICAM_player_info_req:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA009;

- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя. Он также используется для идентификации сеанса проигрывателя.

8.7.13 APDU CICAM_player_info_reply

CICAM направляет этот APDU как ответ на APDU CICAM_player_info_req. Синтаксис APDU CICAM_player_info_reply показан в таблице 62.

Таблица 62 — Синтаксис APDU CICAM_player_info_reply

Синтаксис	Количество битов	Мнемоника
CICAM_player_info_reply() {		
CICAM_player_info_reply_tag	24	uimsbf
length_field() = 11		
LTS_id	8	uimsbf
duration	32	uimsbf
position	32	uimsbf
speed	16	uimsbf
}		

Семантика полей APDU CICAM_player_info_reply:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00A;

- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя и для идентификации сеанса проигрывателя;

- **duration**: поле 32 бита содержит общую продолжительность проигрывания контента в секундах. Если она неизвестна, то в поле должно быть установлено 0xFFFFFFFF;

- **position**: поле 32 бита сигнализирует текущую позицию проигрывания, выраженную в секундах, прошедших с начала проигрывания контента. Если позиция неизвестна, то в поле должно быть установлено 0xFFFFFFFF;

- **speed**: поле 16 битов содержит целое число, отображающее текущую скорость проигрывания, выраженную в сотых долях от номинальной скорости проигрывания.

8.7.14 APDU CICAM_player_stop

Хост посылает этот APDU к CICAM для прекращения сеанса проигрывателя.

Синтаксис APDU CICAM_player_stop показан в таблице 63.

Таблица 63 — Синтаксис APDU CICAM_player_stop

Синтаксис	Количество битов	Мнемоника
CICAM_player_stop() {		
CICAM_player_stop_tag	24	uimsbf

Окончание таблицы 63

Синтаксис	Количество битов	Мнемоника
length_field() = 1		
LTS_id	8	uimsbf
}		

Семантика полей APDU CICAM_player_info_reply:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00B;

- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя и является идентификатором сеанса проигрывателя.

8.7.15 APDU CICAM_player_end

CICAM направляет этот APDU для завершения сеанса проигрывателя CICAM. Это позволяет Хосту освободить занятые ресурсы, например, интерфейса TS. CICAM должен закрыть все гибридные LSC, используемые сеансом проигрывателя CICAM, до отправления этого APDU.

Синтаксис APDU CICAM_player_end показан в таблице 64.

Таблица 64 — Синтаксис APDU APDU CICAM_player_end

Синтаксис	Количество битов	Мнемоника
CICAM_player_end () {		
CICAM_player_end_tag	24	uimsbf
length_field() = 1		
LTS_id	8	uimsbf
}		

Семантика полей APDU APDU CICAM_player_end:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00C;

- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя и является идентификатором сеанса проигрывателя.

8.7.16 APDU CICAM_player_asset_end

CICAM должен отправить этот APDU для уведомления Хоста о том, что были достигнуты начало или конец актива контента.

Синтаксис APDU CICAM_player_asset_end показан в таблице 65.

Таблица 65 — Синтаксис APDU CICAM_player_asset_end

Синтаксис	Количество битов	Мнемоника
CICAM_player_asset_end () {		
CICAM_player_session_end_tag	24	uimsbf
length_field() = 2		
LTS_id	8	uimsbf
reserved	7	bslbf
beginning	1	bslbf
}		

Семантика полей APDU CICAM_player_asset_end:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00D;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя и является идентификатором сеанса проигрывателя;
- **reserved**: поле 7 битов должно содержать 0x7F;
- **beginning**: флаг при установке 0b1 означает, что достигнуто начало актива. Установка 0b0 означает, что достигнут конец актива.

8.7.17 APDU CICAM_player_update_req

CICAM должен отправить этот APDU, чтобы уведомить Хост о поставке Хосту описания компонентов PMT через интерфейс TS. Этот APDU должен отправляться при каждом изменении PMT в соответствующем локальном TS.

Синтаксис APDU CICAM_player_update_req показан в таблице 66.

Таблица 66 — Синтаксис APDU CICAM_player_update_req

Синтаксис	Количество битов	Мнемоника
CICAM_player_update_req () {		
CICAM_player_update_req_tag	24	uimsbf
length_field()		
LTS_id	8	uimsbf
PMT_length	16	uimsbf
for (int i=0; i< PMT_length; i++) {		
PMT_byte	8	bslbf
}		
}		

Семантика полей APDU CICAM_player_update_req:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00E;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, который используется сеансом проигрывателя и является идентификатором сеанса проигрывателя;
- **PMT_length**: поле 16 битов содержит количество байт, составляющих PMT. Оно не должно быть равно нулю;
- **PMT_byte**: поле 8 битов содержит PMT, представленную в виде таблицы MPEG, где первый байт содержит table_id PMT.

8.7.18 APDU CICAM_player_update_reply

Хост должен отправить этот APDU в ответ на APDU CICAM_player_update_req. После приема этого APDU CICAM может доставлять на Хост достоверные SPTS с гарантией того, что Хост готов интерпретировать транспортные пакеты от начала доставки SPTS.

Синтаксис APDU CICAM_player_update_reply показан в таблице 67.

Таблица 67 — Синтаксис APDU CICAM_player_update_reply

Синтаксис	Количество битов	Мнемоника
CICAM_player_update_reply() {		
CICAM_player_start_reply_tag	24	uimsbf
length_field()= 2		
LTS_id	8	uimsbf

Окончание таблицы 67

Синтаксис	Количество битов	Мнемоника
update_status }	8	uimsbf

Семантика полей APDU CICAM_player_update_reply:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9FA00F;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS для сеанса проигрывателя;
- **update_status**: поле 8 битов содержит значения статуса Хоста. Возможные значения update_ status приведены в таблице 68.

Таблица 68 — Значения update_status

update_status	Значение
Хост обработал обновленную PMT и готов получить локальный TS	0x00
Запрос отвергнут	0x01
Зарезервировано	от 0x02 до 0xFF

8.7.19 Сводный перечень ресурсов CICAM в режиме проигрывателя

В таблице 69 представлен сводный перечень атрибутов ресурса CICAM в режиме проигрывателя.

Таблица 69 — Сводный перечень атрибутов ресурса CICAM в режиме проигрывателя

Ресурсы				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00 93 00 41	147	1	1	CICAM_player_verify_req	9F A0 00	→	
				CICAM_player_verify_reply	9F A0 01	←	
				CICAM_player_capabilities_req	9F A0 02	←	
				CICAM_player_capabilities_reply	9F A0 03	→	
				CICAM_player_start_req	9F A0 04	←	
				CICAM_player_start_reply	9F A0 05	→	
				CICAM_player_play_req	9F A0 06	→	
				CICAM_player_status_error	9F A0 07	←	
				CICAM_player_control_req	9F A0 08	→	
				CICAM_player_info_req	9F A0 09	→	
				CICAM_player_info_reply	9F A0 0A	←	
				CICAM_player_stop	9F A0 0B	→	
				CICAM_player_end	9F A0 0C	←	
				CICAM_player_asset_end	9F A0 0D	←	
				CICAM_player_update_req	9F A0 0E	←	
				CICAM_player_update_reply	9F A0 0F	→	

9 Поиск файла CICAM

9.1 Общие замечания

Поиск файла CICAM выполняется с помощью нового ресурса системы вспомогательных файлов CI Plus™. Ресурс системы вспомогательных файлов дает CICAM универсальный механизм предоставления Хосту файлов, например приложений вещания CICAM, которые могут быть запущены для работы на Хосте. Метод, используемый Хостом для доступа к ресурсу, зависит от соответствующей среды приложения Хоста и настоящим стандартом не нормируется. Смотри 12.4.3.3 настоящего стандарта об использовании CICAM системы вспомогательных файлов для предоставления приложений.

Этот новый ресурс наследует большую часть своих сообщений и базовой семантики ресурсов из приложения MMI v2, которые определены в [1] (14.5).

Хост предоставляет ресурс системы вспомогательных файлов с идентификатором ресурса 0x00910041. Этот ресурс позволяет Хосту извлекать файлы от CICAM, например:

- для использования приложениями, работающими на Хосте;
- для сигнализации вещания, позволяющей ссылаться на приложение, хранящееся на CICAM.

Кроме того, ресурс системы вспомогательных файлов позволяет Хосту получать детализированную информацию о доступных приложениях на CICAM.

Система файлов предназначена только для чтения, она дает возможность запрашивать и принимать конкретные известные файлы, если они существуют, при их отсутствии она возвращает сообщение об ошибке.

CICAM предлагает Хосту свою систему файлов, используя APDU FileSystemOffer. Хост запрашивает файл, используя APDU FileRequest. Файлы отправляются от CICAM к Хосту, используя APDU FileAcknowledge.

CICAM за сеанс ресурса системы вспомогательных файлов должен обеспечивать только одну систему файлов. Если у CICAM будет несколько предложений системы, то он должен запрашивать один сеанс к ресурсу для каждой системы файлов, которые он предлагает.

Хост должен поддерживать не менее трех сеансов к данному ресурсу.

Если CICAM по какой-либо причине должен остановить доступ к своей системе файлов, например из-за обновления системы файлов, то он должен закрыть сеанс к ресурсу системы вспомогательных файлов.

9.2 APDU системы файлов

APDU FileSystemOffer определяет характеристики системы файлов, предоставляемой CICAM.

CICAM должен запрашивать новый сеанс для каждой системы файлов CICAM, которую он хочет предложить.

Только один APDU FileSystemOffer должен направляться на каждый сеанс дополнительного ресурса системы файлов, открытого CICAM.

Синтаксис APDU FileSystemOffer показан в таблице 70.

Таблица 70 — Синтаксис APDU FileSystemOffer

Синтаксис	Количество битов	Мнемоника
FileSystemOffer() {		
FilesystemOffer_tag	24	uimsbf
length_field()		
DomainIdentifierLength	8	uimsbf
for (i = 0; I < DomainIdentifierLength; i ++) {		
DomainIdentifier	8	uimsbf
}		
}		

Семантика полей APDU FileSystemOffer:

- **FileSystemOffer_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F9400;
- **length_field**: поле 8 битов содержит длину полезной нагрузки APDU в формате BER ASN.1, определенном в [2] (8.3.1);
- **DomainIdentifierLength**: поле 8 битов определяет количество байтов в поле DomainIdentifier;
- **DomainIdentifier**: поле 8 битов содержит информацию, необходимую Хосту для идентификации системы файлов, представленной CICAM. DomainIdentifier может быть идентичным AppDomainIdentifier, который используется приложением ресурса MMI. Поле DomainIdentifier не имеет определенного формата. Значение его байтов определяется промежуточным ПО и настоящим стандартом не определяется. Примерами возможного содержания поля DomainIdentifier являются следующие:
- **URL-адрес**, который может быть использован для идентификации системы файлов, которая принадлежит организации, управляющей системой файлов, например: «gost.e-ecolog.ru
API может использовать это значение в качестве прекурсора адреса URL или путей доступа к файлам, предлагаемым CICAM через этот ресурс.
- **UUID** должно быть в соответствии с [15], например: «urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6»;
- **идентификатор**, зарегистрированный в DVB.

9.3 APDU FileSystemAck подтверждения поддержки

APDU FileSystemAck должен быть направлен Хостом в ответ на APDU FileSystemOffer для подтверждения поддержки Хостом предлагаемого домена приложения.

Синтаксис APDU FileSystemAck показан в таблице 71.

Таблица 71 — Синтаксис APDU FileSystemAck

Синтаксис	Количество битов	Мнемоника
FileSystemAck() {		
FilesystemAck_tag	24	uimsbf
length_field()		
AckCode	8	bslbf
}		

Семантика полей APDU CICAM_player_update_reply:

- **FileSystemAck_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F9401;
- **length_field**: длина поля полезной нагрузки определяется как формат BER ASN.1 в соответствии с [2] (8.3.1);
- **AckCode**: поле 8 битов содержит ответ на APDU FileSystemOffer.

Значения кода Аск представлены в таблице 72.

Таблица 72 — Значения кода Аск

AckCode	Значение кода
0x00	Зарезервировано
0x01	Хост поддерживает среду приложений
0x02	Хост не поддерживает среду приложений
0x03 до 0xFF	Зарезервировано

9.4 APDU FileRequest запроса файла

APDU FileRequest копируется из ресурса приложения MMI v2, он полностью описан в [1] (14.5.1). Он позволяет Хосту поддерживать кэширование файлов и обнаруживать типы запросов, поддерживаемых CICAM (файл, данные).

9.5 APDU FileAcknowledge подтверждения файла

APDU FileAcknowledge скопирован из ресурса приложения MMI v2 и полностью описан в [1] (14.5.2).

9.6 Сводная информация о ресурсах системы вспомогательных файлов

Информация об атрибутах ресурса системы вспомогательных файлов приведена в таблице 73.

Таблица 73 — Информация об атрибутах ресурса системы вспомогательных файлов

Ресурсы				Объект приложения		Направление передачи	
Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM
00 91 00 41	145	1	1	FileSystemOffer	9F 94 00	←	
				FileSystemAck	9F 94 01	→	
				FileRequest	9F 94 02	→	
				FileAcknowledge	9F 94 03	←	

9.7 Ресурсы системы вспомогательных файлов и ресурсы приложения MMI

Ресурсы системы вспомогательных файлов и ресурсы приложения MMI обеспечивают доступ к системе файлов CICAM. Когда оба ресурса открыты, Хост должен использовать эти ресурсы в соответствии со следующими правилами:

- приложения AppMMI CICAM и приложения, запущенные приложениями AppMMI CICAM, должны использовать ресурс приложения MMI;
- все другие приложения, включая случай, когда приложение CICAM передается средствами вещания и является приложением MHEG-5, получающим доступ к резидентной программе CI_SendMessage (CIS), определенной стандартом [7] (11.10.11), должны использовать ресурс системы вспомогательных файлов.

10 Ресурс низкоскоростных соединений

10.1 Общие замечания

В этом разделе нормируются параметры ресурса низкоскоростных соединений версии 4, которая облегчает режим доставки IP CICAM в режиме проигрывателя к ресурсам LSC версии 3, которая определена в [1] (14.1).

Версия 4 ресурса LSC в качестве базовой использует версию 3 ресурса LSC и расширяет ее с добавлением поддержки следующих функций:

- многоадресных соединений (IGMPv3), специфичных для источника;
- доставки данных ответа через интерфейс TS;
- передачи информации в сеансе LSC от Хоста до CICAM новым APDU comms_info;
- передачи информации о сетевых адаптерах IP Хоста на CICAM новым APDU comms_IPconfig;
- передачи CICAM сообщений при повторном использовании порта IP источника предыдущего сеанса с помощью нового дескриптора источника IP.

Типы ресурса LSC расширяются для усиления поддержки гибридных соединений. В этом разделе нормированы дополнения к расширениям IP низкоскоростных соединений, определенным в [1] (14.2). Для облегчения доставки IP CICAM в режиме проигрывателя стек IP Хоста должен удовлетворять требованиям [16] — [20].

Для этой версии ресурс поддержки IPv6 для Хоста является опциональным. Поддержка Хостом IPv6 должна быть совместимой с [21] и [22].

10.2 Информация о конфигурации IP Хоста

Для некоторых механизмов доставки CICAM должен делать вызовы в сети IP низкого уровня для получения информации о сети. Например, это может потребоваться в следующих случаях:

- для поиска DNS, чтобы установить IP-адрес сервера с именем DNS;
- для выполнения запроса DHCP с целью получения домена, в котором расположен Хост.

Это означает, что CICAM должен иметь доступ к:
- IP-адресу Хоста;
- адресу сервера DHCP;
- адресам серверов DNS;
- к возможности оценки состояния соединения сетевого интерфейса;
- адресу шлюза IP.

Новый APDU должен позволять CICAM получать информацию о конфигурации Хоста в сети для разрешения перечисленных функций.

10.3 Информация о потоках IP

Настоящий стандарт обеспечивает поддержку обмена информацией о соединении IP между Хостом и CICAM. Определен новый APDU comms_info_request, который позволяет CICAM получать информацию об установлении IP соединения. В частности, для гибридных соединений он определяет PID, по которому данные IP будут доставлены через интерфейс TS. Он также определяет информацию об адресе источника IP и о порте, используемых Хостом для данного соединения на сети IP. Информация о порте источника IP полезна в случае использования транспортного протокола, требующего согласования портов данных и управления портами во время инициализации сеанса, например, RTSP. Кроме того, она может использоваться при установлении другого соединения с повторным использованием того же самого IP исходного порта в соответствии с 10.5 настоящего стандарта.

10.4 Многоадресный IP

Настоящий стандарт дополнительно поддерживает многоадресные потоки от Хоста к CICAM через интерфейс TS. Когда CICAM обращается к многоадресному соединению (через multicast_descriptor), Хост пошлет пакет присоединения IGMP и передаст данные, полученные от многоадресного источника, через интерфейс TS к CICAM. Когда многоадресное соединение закрыто CICAM, Хост должен послать на сервер пакет <leave> IGMP.

10.5 Порты источника IP

В IP-соединениях может представлять интерес возможность использования исходящими данными порта источника IP, который уже используется Хостом.

Например, в случае телевидения с ретрансляцией или механизмов быстрой смены канала в случае DVB-IPTV, как это определено в [16], данные должны исходить от одного порта-источника. Для этой цели настоящий документ добавляет поддержку создания сеанса LSC дескриптором, который использует исходный порт IP, который одновременно уже используется в сеансе LSC.

10.6 Доставка данных от Хоста к CICAM

При работе с гибридным соединением IP при получении данных IP Хост должен инкапсулировать данные IP в пакеты TS, как это определено в 10.12.4 настоящего стандарта, и в результате должен переслать в CICAM пакеты TS через интерфейс TS.

CICAM должен поддерживать все необходимые протоколы сеанса проигрывателя.

10.7 Доставка данных от CICAM к Хосту

При установлении гибридного соединения CICAM должен сопрягаться с Хостом транспортным потоком, совместимым с [9], поддерживая модели буфера TS. Этот TS может быть защищен при использовании существующего механизма скремблирования на уровне транспортного потока CI Plus™. TS наряду с компонентами AV должен содержать таблицы PAT и PMT для обслуживания. PAT и PMT могут быть повторены в TS, но CICAM должен вставлять новую PMT, если возникают какие-либо изменения в структуре TS.

TS на выходе CICAM должен соответствовать модели синхронизации, определенной в системе MPEG 2 [9]. Должна быть предусмотрена возможность использования TS непосредственно Хостом.

Скорость передачи на выходе CICAM не должна превышать значений скорости во время установления сеанса проигрывателя. Измерение скорости передачи на выходе CICAM должно выполняться не менее чем через 100 мс.

Гибридные соединения не должны оказывать влияния на другие локальные TS, которые могут одновременно работать в многопоточном режиме.

Для передачи данных в сети IP CICAM должен использовать сервер управления потоком и APDU comms_send, определенный в [1] (14.1.4).

10.8 Управление потоком IP

В режиме доставки IP на CICAM в режиме проигрывателя управление потоком осуществляет CICAM. При работе с протоколом вытягивания, например HTTP, CICAM будет запрашивать сегменты данных в соответствии с его моделью буфера. В случае протокола проталкивания Хост должен передавать поток данных в CICAM с минимальной задержкой.

Примечание — Рекомендуется предусматривать превышение допустимой скорости передачи через интерфейс TS относительно фактической скорости передачи битов аудио/видео с учетом доставки в сети IP с переменной скоростью.

10.9 APDU запроса comms_info

10.9.1 Общие замечания

Настоящий стандарт вводит новый APDU для запроса информации о сеансе LSC. Применение нового APDU целесообразно в том случае, когда CICAM запрашивает IP-имя Хоста, тип соединения: гибридное соединение или подключением источника IP с использованием соответствующего connection_descriptor_type.

CICAM должен использовать этот APDU, устанавливая гибридное соединение, после того как он получил APDU comms_reply, чтобы получить детализированную информацию о PID, на который будут доставлены данные IP. CICAM может также отправить этот APDU, если он хочет узнать порт источника, поставляющего данные.

10.9.2 APDU comms_info_req

CICAM посылает этот APDU к Хосту для запроса детализированной информации о сеансе соединения LSC, работающего в гибридном режиме.

Синтаксис APDU comms_info_req показан в таблице 74.

Таблица 74 — Синтаксис APDU comms_info_req

Синтаксис	Количество битов	Мнемоника
comms_info_req () {		
comms_info_req_tag	24	uimsbf
length_field() = 1		
}		

Семантика полей APDU comms_info_req:

- **comms_info_req_ tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F8C07;
- **length_field**: поле содержит длину полезной нагрузки APDU в формате BER ASN.1 в соответствии с [2] (8.3.1).

10.9.3 APDU comms_info_reply

Этот APDU должен направляться в ответ на APDU comms_info_req с Хоста на CICAM, чтобы сообщить CICAM IP-адрес источника, используемого Хостом для связи, и PID, используемого для передачи этой информации от Хоста к CICAM для гибридных соединений.

Синтаксис APDU comms_info_reply показан в таблице 75.

Таблица 75 — Синтаксис APDU comms_info_reply

Синтаксис	Количество битов	Мнемоника
comms_info_reply () {		
comms_info_reply_tag	24	uimsbf
length_field() = 21		
LTS_id	8	uimsbf
reserved	7	bslbf

Окончание таблицы 75

Синтаксис	Количество битов	Мнемоника
status	1	bslbf
source_IPaddress	128	uimsbf
source_port	16	uimsbf
reserved	3	bslbf
inputDeliveryPID	13	uimsbf
}		

Семантика полей APDU comms_info_reply:

- **CICAM_player_play_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F8C08;
- **length_field**: длина полезной нагрузки APDU должна быть в формате BER ASN.1 в соответствии с [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS;
- **status**: поле определяет статус соединения. Значение 0b1 означает, что соединение установлено. Значение 0b0 означает, что соединение не установлено. CICAM должен игнорировать последующие поля, если в поле status установлено 0b0;
- **source_IPaddress**: поле 128 битов содержит IP-адрес источника, который используется Хостом для сеанса LSC. Если Хост не может определить IP-адрес источника, то в поле должен быть установлен ноль. Для адресов IPv4 первые 12 байтов должны быть установлены в 0x00;
- **source_port**: поле 16 битов содержит адрес порта источника, который используется Хостом для сеанса LSC. Если Хост не знает порт источника, то в этом поле должно быть установлено значение 0x0000;
- **inputDeliveryPID**: поле 13 битов используется для доставки данных IP через интерфейс TS для гибридных соединений. Эти PID выделены Хостом, они должны находиться в диапазоне значений от 0x0020 до 0x1FFE. В тех случаях, когда соединение LSC не является гибридным, это значение должно быть 0x0000. В случае нескольких одновременных сеансов LSC с гибридными соединениями с одним и тем же локальным TS каждому из сеансов должен быть выделен уникальный PID.

10.10 APDU comms_IPconfig запроса конфигурации IP Хоста

10.10.1 Общие замечания

Настоящий стандарт вводит параметры нового APDU, позволяющего CICAM получать информацию о конфигурации IP Хоста. Он включает в себя адреса IP по умолчанию шлюза и адрес сервера DNS, назначенные Хосту. Эта информация может быть использована CICAM для прямой передачи запросов и справок о сервере DNS, сконфигурированном для Хоста, при открытии сеанса дескриптором LSC IP_descriptor.

10.10.2 APDU comms_IP_config_req

Этот APDU должен быть направлен от CICAM к Хосту для запроса информации о конфигурации IP Хоста.

Синтаксис APDU comms_IP_config_req показан в таблице 76.

Таблица 76 — Синтаксис APDU comms_IP_config_req

Синтаксис	Количество битов	Мнемоника
comms_IP_config_req () {		
comms_IP_config_req_tag	24	uimsbf
length_field() = 0		
]		

Семантика полей APDU comms_IP_config_req:

- **comms_info_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F8C08;
- **length_field**: длина полезной нагрузки APDU должна быть в формате BER ASN.1 в соответствии с [2] (8.3.1).

10.10.3 APDU comms_IP_config_reply

Этот APDU должен быть направлен Хостом в ответ на comms_IP_config_req, чтобы сообщить CICAM о конфигурации IP-порта Хоста. Хост сообщает о конфигурации основного адаптера, который в настоящее время используется Хостом для IP-соединения. Синтаксис APDU comms_IP_config_reply показан в таблице 77.

Таблица 77 — Синтаксис APDU comms_IP_config_reply

Синтаксис	Количество битов	Мнемоника
comms_IP_config_reply () {		
comms_IP_config_reply_tag	24	uimsbf
length_field ()		
connection_state	2	uimsbf
reserved	6	bslbf
physical_address	48	uimsbf
if (connection_state == 01) {		
IP_address	128	uimsbf
network_mask	128	uimsbf
default_gateway	128	uimsbf
DHCP_server_address	128	uimsbf
num_DNS_servers	8	uimsbf
for (int i=0; i<num_DNSservers; i++) {		uimsbf
DNS_server_address	128	uimsbf
}		
}		
}		

Семантика полей APDU comms_IP_config_reply:

- **comms_info_req_tag**: поле 24 бита содержит значение тега, которое должно быть 0x9F8C08;
- **connection state**: поле 2 бита определяет состояние соединения IP-адаптера. Поле содержит следующие значения:
  - 0x00 отключен (сетевой интерфейс неактивен или отключен);
  - 0x01 связан (сетевой интерфейс активен, подключен и имеет действительный IP-адрес);
  - от 0x10 до 0x11 значения зарезервированы;
- **physical_address**: поле 48 битов содержит MAC-адрес этого IP сетевого адаптера;
- **IP_address**: поле 128 битов содержит IP-адрес, назначенный этому IP-адаптеру;
- **network_mask**: поле 128 битов содержит маску подсети для этого IP сетевого адаптера;
- **default_Gateway**: поле 128 битов содержит IP-адрес шлюза, используемого для этого IP сетевого адаптера;
- **DHCP_server_address**: поле 128 битов содержит IP-адрес DHCP-сервера, используемого для этого IP сетевого адаптера. Если сервер DHCP отсутствует, то это поле должно быть установлено в ноль;
- **num_DNS_servers**: поле 8 битов содержит количество DNS-серверов, используемых для этого IP сетевого адаптера;
- **DNS_server_address**: поле 128 битов содержит IP-адрес сервера DNS.

10.11 Параметры модифицированного Comms Cmd

10.11.1 Общие замечания

Объект дескриптора соединения дополняется двумя новыми типами соединения, которые обеспечивают следующие параметры:

- источника многоадресной доставки;
- IP прямого канала по тракту LSC и обратного канала через интерфейс TS.

Дополнительно вводится новое поле, позволяющее CICAM сообщать исходный порт, который должно использовать соединение.

Дескриптор connection_descriptor определен в [1] (таблица 14.5). Он изменяется добавлением типов дескрипторов гибридной связи и многоадресных источников.

Синтаксис connection_descriptor показан в таблице 78.

Таблица 78 — Синтаксис connection_descriptor

Синтаксис	Количество битов	Мнемоника
connection_descriptor () {		
connection_descriptor_tag	24	uimsbf
length_field()		
source_port_flag	1	bslbf
connection_descriptor_type	7	uimsbf
if (source_port_flag == 0b1) {		
IP_source_port	16	uimsbf
}		
if (connection_descriptor_type == SI_telephone_descriptor) {		
telephone_descriptor ()		
}		
if (connection_descriptor_type ==cable_return_channel_descriptor) {		
channel_id	8	uimsbf
}		
if (connection_descripotr_type == IP_descriptor) {		
IP_descriptor ()		
}		
if (connection_descriptor_type == hostname_descriptor) {		
hostname_descriptor ()		
}		
if (connection_descriptor_type == hybrid_descriptor) {		
hybrid_descriptor ()		
}		
if (connection_descriptor_type == IP_multicast_descriptor) {		
IP_multicast_descriptor ()		
}		
}		

Семантика полей APDU CICAM_player_update_reply:

- **connection_descriptor_tag**: поле 24 бита содержит значение тега 0x9FA00F.

Типы дескрипторов гибридной связи и многоадресных источников представлены в таблице 79.

Таблица 79 — Типы дескрипторов гибридной связи и многоадресных источников

connection_descriptor_type	Значения типов
SI_telephone_descriptor	0x01
cable_return_channel_descriptor	0x02
IP_descriptor	0x03
hostname_descriptor	0x04
hybrid_descriptor	0x05
multicast_descriptor	0x06
Зарезервировано	от 0x07 до 0x7F

В состав дескрипторов входят следующие поля:

- **sourcePortFlag**: флаг при значении 0b1 в поле connection_descriptor указан IP_source_port.

Примечание — MSB, размещенный в connection_descriptor_type, для sourcePortFlag сохраняет обратную совместимость с дескриптором соединения [1];

- **IP_source_port**: поле 16 битов содержит адрес порта источника IP, который должен быть использован Хостом для этого соединения. Этот порт источника должен быть одним из исходных портов, которые уже используются другим сеансом LSC, как это предусмотрено в CICAM через APDU comms_info_reply.

В случае если CICAM использует IP_source_port, который уже используется в сеансе LSC, Хост может игнорировать это поле и выбрать свой собственный исходный порт.

Адреса источника и порта назначения, используемые в этом IP-соединении, должны отличаться от адреса источника и порта назначения, используемых в сеансе LSC, который использует тот же исходный порт IP. Это позволяет предотвратить создание одновременно двух сеансов LSC, использующих одинаковую комбинацию IP-адреса, порта источника и порта назначения.

10.11.2 Команда сообщения hybrid_descriptor

Синтаксис команды сообщения hybrid_descriptor показан в таблице 80.

Таблица 80 — Синтаксис команды сообщения hybrid_descriptor

Синтаксис	Количество битов	Мнемоника
hybrid_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
LTS_id	8	uimsbf
IP_connection_type	8	uimsbf
if (IP_connection_type == IP_descriptor) {		
IP_descriptor()		
}		
if (IP_connection_type == multicast_descriptor) {		
multicast_descriptor()		
}		

Окончание таблицы 80

Синтаксис	Количество битов	Мнемоника

if (IP_connection_type == hostname_descriptor {
hostname_descriptor()
}		

Семантика полей команды сообщения hybrid_descriptor:

- **descriptor_tag**: поле 8 битов содержит тег hybrid_descriptor и имеет значение 0x05;
- **descriptor_length**: поле 8 битов указывает общее количество байтов части данных hybrid_ descriptor после байта, определяющего значение этого поля;
- **LTS_id**: поле 8 битов содержит идентификатор локального TS, с которым ассоциируется это соединение, а следовательно, и локальный TS, на который должны быть доставлены данные IP. Это значение возвращается в APDU CICAM_player_start_reply после APDU CICAM_player_start_req. Если ни один сеанс проигрывателя не был установлен с указанным LTS_id, то Хост должен сообщить об ошибке;
- **IP_connection_type**: поле 8 битов указывает тип гибридного соединения в соответствии с таблицей 81;

Таблица 81 — Тип гибридного соединения

IP_connection_type	Значения типов
Зарезервировано	0x01
Зарезервировано	0x02
IP_descriptor	0x03
hostname_descriptor	0x04
Зарезервировано	0x05
multicast_descriptor	0x06
Зарезервировано	от 0x07 до 0xFF

- **IP_descriptor**: в соответствии с [1] (14.2.1.1);
- **hostname_descriptor**: в соответствии с [1] (14.2.1.2);
- **multicast_descriptor**: в соответствии с 10.11.3 настоящего стандарта.

10.11.3 Дескриптор команды многоадресной доставки Comms

Синтаксис дескриптора команды многоадресной доставки multicast_descriptor показан в таблице 82.

Таблица 82 — Синтаксис дескриптора команды многоадресной доставки multicast_descriptor

Синтаксис	Количество битов	Мнемоника
multicast_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
IP_protocol_version	8	uimsbf
IP_address	128	uimsbf
multicast_port	16	uimsbf
reserved	7	bslbf

Окончание таблицы 82

Синтаксис	Количество битов	Мнемоника
include_sources	1	bslbf
num_source_addresses	8	uimsbf
for (i=0; i<num_source_addresses; i++) {		
source_address	128	uimsbf
}		
}		

Семантика полей multicast_descriptor:

- **descriptor_tag**: поле 8 битов содержит тег дескриптора и имеет значение 0x06;
- **descriptor_length**: поле 8 битов указывает общее количество байтов части данных multicast_descriptor после поля descriptor_tag;
- **IP_protocol_version**: поле 8 битов указывает версию протокола IP: величине 0x01 соответствует IPv4, величине 0x02 соответствует IPv6;
- **IP_address**: поле 128 битов указывает IP-адрес многоадресной службы. Для адресов IPv4 первые 12 байтов должны быть установлены в 0x00;
- **multicast_port**: поле 16 битов указывает адрес многоадресного порта, который будет использоваться Хостом;
- **includeSources**: флаг при установке 0b1 сигнализирует, что CICAM желает получить IP многоадресные потоки от любого из IP-адресов источников в списке. При установке 0b0 он сообщает, что CICAM желает получить IP многоадресные потоки от всех IP-адресов источников, кроме тех, которые перечислены в списке. Это поле имеет значение только тогда, когда в поле num_source_addresses установлено значение больше чем 0;
- **num_source_addresses**: поле 8 битов указывает количество адресов, используемых многоадресных источников. В случае когда многоадресные источники отсутствуют, в поле будет установлен 0, и Хост принимает IP многоадресные потоки от источника с любым адресом. При отсутствии адреса источника Хост должен считать, что для многоадресного соединения связи должен использоваться IGMPv2;
- **source_address**: поле 128 битов содержит адрес источника для многоадресной рассылки.

10.12 Модификация типов ресурсов низкоскоростных соединений

10.12.1 Общие замечания

Новое значение типа ресурса LSC добавляется для поддержки гибридных соединений. Значения типов устройства для ресурсов LSC версии 4 определены в таблице 83.

Таблица 83 — Значения типов ресурса

Описания	Значения типов
Модемы	0x00 — 0x3F
Последовательные порты	0x40 — 0x4F
Обратный кабельный канал	0x50
Зарезервировано	0x51 — 0x5F
Соединение IP	0x60
Зарезервировано	0x61 — 0x6F
Гибридное соединение	0x70
Зарезервировано	от 0x71 до 0xFF

Если CICAM выдает APDU comms_cmd с дескриптором соединения, который не поддерживается Хостом, Хост должен ответить APDU comms_reply с comms_reply_id = Connect_Ack и установить в поле return_value значение в соответствии с таблицей 84.

Таблица 84 — Значения comms_reply

Описание	Значения
Хост поддерживает дескриптор соединения	0x00
Зарезервировано	от 0x1 до 0x7F
Частная ошибка	от 0x80 до 0xFD
Протокол соединения не поддерживается	0xFE
Ошибка нетипичная	0xFF

10.12.2 Выполнение разъединения

При инициировании разъединения конечной точкой сети Хост должен передать все данные в буферы, ожидающие приема в CICAM, и затем должен отправить незапрошенный APDU comms_reply в соответствии с [1] (14.1.2).

10.12.2.1 Передача данных через интерфейс TS

10.12.2.2 Синтаксис пакетов TS

При использовании гибридного соединения данные IP, полученные Хостом, поставляются через интерфейс TS инкапсулированными в пакетах TS. Таблица 85 показывает синтаксис заголовка пакета TS для инкапсуляции.

Таблица 85 — Синтаксис заголовка пакета TS для инкапсуляции

Синтаксис	Количество битов	Мнемоника
transport_packet () {		
sync_byte	8	bslbf
transport_error_indicator	1	bslbf
payload_unit_start_indicator	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
transport_scrambling_control	2	bslbf
adaptation_field_control	2	bslbf
continuity_counter	4	uimsbf
if(adaptation_field_control == '01' || adaptation_field_control =='11') {		
adaptation_field()		
}		
for(i=0; i<N; i++) {		
IP_datagram_byte	8	bslbf
}		
}		

Поля должны быть определены в соответствии с требованиями системы MPEG-2 [9] со следующими исключениями:

- **sync_byte**: поле 8 битов используется для идентификации локального TS на интерфейсе TS. Содержит значение LTS_id, выделенное Хостом для сеанса проигрывателя;

- **transport_error_indicator**: флаг должен быть установлен в 0b0;
- **payload_unit_start_indicator**: флаг должен быть установлен в 0b0;
- **transport_priority**: флаг должен быть установлен в 0b0;
- **PID**: поле 13 битов, значение этого поля присваивается сервером, оно является уникальным в рамках локального TS. Это означает, что у каждого сеанса LSC, имеющего гибридное соединение, в котором используется этот локальный TS, для пакетов IP, связанных с этим сеансом, будет другой PID;
- **transport_scrambling_control**: поле 2 бита должно быть установлено в 0b00;
- **IP_datagram_byte**: поле 8 битов должно содержать полезную нагрузку пакета TCP или UDP.

10.12.2.3 Использование поля адаптации

Поле адаптации должно использоваться для заполнения пакетов TS, когда последняя секция дейтаграммы данных IP заполняет пакет TS частично. Это достигается следующим образом:

- **adaptation_field_length**: это поле должны занимать байты, оставшиеся свободными в пакете TS, так как поле адаптации должно заполнить пакет TS;
- **discontinuity_indicator**: флаг должен быть установлен в 0b0;
- **random_access_indicator**: флаг должен быть установлен в 0b0;
- **elementary_stream_priority_flag**: флаг должен быть установлен в 0b0;
- **PCR_flag**: флаг должен быть установлен в 0b0;
- **OPCR_flag**: флаг должен быть установлен в 0b0;
- **splicing_point_flag**: флаг должен быть установлен в 0b0;
- **transport_private_data_flag**: флаг должен быть установлен в 0b0;
- **adaptation_field_extension_flag**: флаг должен быть установлен в 0b0;

для (INT I = 0; i <N; i ++) {

- **stuffing_byte**: в этом поле должно быть установлено 0xFF.

11 Расширенные правила использования информации, версия 3

Определения URI спецификации CI Plus™ в версиях 1 и 2 установлены в [1] (5.7.1). Настоящий стандарт расширяет правила использования информации URI для CI Plus™ и устанавливает параметры версии 3 URI CI Plus™. Для обеспечения совместимости с устаревшими устройствами URI версии 3 применяется только к ресурсу управления контентом типа 64 версии 3 и к более поздним версиям или к ресурсу управления контентом типа 65 в соответствии с приложением Б настоящего стандарта. Управление контентом типа 64 версии 3 использует те же APDU, как и в случае версии 2.

Хост может декларировать поддержку версии 3 URI во время согласования версии URI только на сеансе управления контентом, который использует идентификатор ресурса, как определено в 6.4.3 настоящего стандарта. CICAM может отправить версию 3 URI только тогда, когда Хост декларировал поддержку этой версии.

URI версии 3 CI Plus™ добавляет возможность управления режимом спецэффектов, применимую к контенту с набором emi_copy_control. Возможность управления разрешена для одной копии, с возможностью сигнализировать управление режимом спецэффектов, которое может быть включено или отключено.

Таблица 86 содержит значения по умолчанию для URI версии 3 CI Plus™, включая «trick_mode_control_info».

Таблица 86 — Значения по умолчанию для URI версии 3 CI Plus™

Поле	Исходное значение по умолчанию
Версия протокола	0x03
emi_copy_control_info	0b11
aps_copy_control_info	0b00
ict_copy_control_info	0b0
rct_copy_control_info	0b0
dot_copy_control_info	0b0

Окончание таблицы 86

Поле	Исходное значение по умолчанию
rl_copy_control_info	0b00000000
trick_mode_control_info	0b0
Зарезервированные биты	0b0

В таблице 87 представлен синтаксис версии 3 URI, расширяющий существующие версии 1 и 2 URI, содержащиеся в [1] (5.7.5.2).

Таблица 87 — Синтаксис сообщения URI версии 3

Поле	Количество битов	Мнемоника
uri_message() {		
protocol_version	8	uimsbf
aps_copy_control_info	2	uimsbf
emi_copy_control_info	2	uimsbf
ict_copy_control_info	1	uimsbf
if (emi_copy_control_info == 00) {		
rct_copy_control_info	1	uimsbf
}		
else {		
reserved = 0	1	uimsbf
}		
reserved for future use	1	uimsbf
if (emi_copy_control_info == 11) {		
dot_copy_control_info	1	uimsbf
rl_copy_control_info	8	uimsbf
}		
else {		
reserved = 0x00	9	uimsbf
}		
if (emi_copy_control_info == 10) {		
trick_mode_control_info	1	uimsbf
}		
else {		
reserved = 0	1	uimsbf
}		
reserved for future use	39	uimsbf
}		

Кодирование и семантика поля trick_mode_control_info выполняются следующим образом:

trick_mode_control_info:

- параметр принимает значение 0b0, если управление режимом спецэффектов отключено;
- параметр принимает значение 0b1, если управление режимом спецэффектов включено.

Правила интерпретации сигналов управления режимом спецэффектов в Хосте выходят за рамки настоящего стандарта.

12 Приложения CICAM

12.1 Общие замечания

Раздел 12 настоящего стандарта содержит спецификации приложений CICAM в следующих трех аспектах, связанных с приложением MMI, приложениями CICAM в целом и с их координацией с приложениями вещания:

- расширения спецификации браузера [1] (12.2) определены в 12.2 настоящего стандарта;
- расширения приложений жизненного цикла CI, представленных в [1] (12.6) и определенных в 12.3 настоящего стандарта. Эти расширения не зависят от изменений, содержащихся в 12.4 настоящего стандарта;
- платформа координации приложений, которая позволяет разрешать конфликты между приложениями вещания и приложениями CICAM, которые могут быть запущены в Хосте, определена в 12.4 настоящего стандарта.

В 12.5 настоящего стандарта нормируются следующие характеристики среды приложений Хоста:

- характеристики, позволяющие использовать платформу для предоставления и запуска приложений из CICAM;
- методы, которыми CICAM определяет среды приложения, поддерживаемые Хостом.

12.2 Спецификации браузера CI Plus™ с расширенными функциями

12.2.1 Система файлов InteractionChannelExtension

Профиль механизма CI Plus™ расширяется включением системы файлов InteractionChannelExtension в соответствии [6] (15.1.2) со следующим исключением: в [6] (15.12.2), разрешение ссылок гибридной системы файлов по умолчанию должно отображать имя «CI://» вместо «DSM://».

12.2.2 ICStreamingExtension

Механизм профиля CI Plus ™ расширен включением требований в соответствии с [6] (14.2.9). Исключением являются требования, которые касаются аудио, видео и форматов субтитров, являющихся опциональными (рекомендуемыми).

Рекомендуется спецификацию системы, ссылающуюся на настоящий стандарт для CI Plus™, дополнять требованиями к форматам медиа для IP-контента.

12.2.3 ICEncryptedStreamExtension

Профиль механизма ICEncryptedStreamExtension расширяется в соответствии [7] (15.16) профиля HTTP для доставки скремблированных потоков, является опциональным.

Рекомендуется спецификацию системы, ссылающуюся на настоящий стандарт для CI Plus™, дополнять требованиями к скремблированию потока для контента при IP-доставке.

12.2.4 Масштабирование видео

12.2.4.1 Общие замечания

Настоящий стандарт дополняет спецификацию профиля механизма [1] функцией масштабирования видео. В общем случае базовые возможности масштабирования видео определяются спецификацией профиля вещания MHEG-5 [7].

12.2.4.2 Набор функций масштабирования видео

В таблице 88 перечислены функции GetEngineSupport профиля механизма CI Plus™.

Таблица 88 — Функции GetEngineSupport профиля механизма CI Plus™

Функция	Примечания
Кэширование	Не устанавливается
Формат кадра	Не устанавливается
UniversalEngineProfile	Должен поддерживаться профиль вещания MHEG-5 [7] и должно поддерживаться значение профиля CI Plus™

12.2.4.3 GetEngineSupport

В таблице 89 перечислены функции профиля механизма CI Plus™.

Таблица 89 — Функции GetEngineSupport профиля механизма CI Plus™

Наименование функции	Ограничение
MultipleAudioStreams(N)	Может возвратить «истинно» для N ≤ 1
MultipleVideoStreams(N)	Может возвратить «истинно» для N ≤ 1
DownloadableFont(CHook)	Должен возвращать «истинно» для значения функции, которое поддерживает класс шрифта. Должен возвращать «ложь» для всех других значений N
Примечание — Определения функций профиля механизма CI Plus™ должны быть в соответствии с [7].	

12.3 Управление жизненным циклом приложения

Инструменты управления жизненным циклом приложения CI, специфицированные [1] (12.6), настоящий стандарт расширяет добавлением возможности CICAM передачи:

- запроса о поддержке Хостом среды конкретного приложения при попытке CICAM запуска приложения AppMMI;
- запроса о возможности терминации работающего приложения Хоста при запуске CICAM приложения AppMMI.

Для реализации этого расширения варианты запуска идентификатора домена приложения в [1] (12.6.2, таблица 12.8), которые применимы к APDU Requeststart, расширены (опционально), как показано в таблице 90, и включают запрос домена приложения (Application Domain Query; ADQ).

Это расширение относится к ресурсу приложения MMI версии 3, определенного в настоящем стандарте. Все другие APDU из набора APDU ресурса приложения MMI остаются неизменными в редакции версии 2.

Таблица 90 — Варианты запуска идентификатора домена приложения (опциональные)

Имя	Возможные значения	Комментарии
Статус SSM	SSM=0	RTGraphic (субтитры) должны быть отключены перед запуском приложения CI Plus. При завершении приложения CI Plus™ субтитры должны быть возвращены в их рабочее состояние
	SSM=1	RTGraphic (субтитры) должны отображаться, когда включено любое предпочтение пользователя; если приложения CI Plus™ и субтитры не могут сосуществовать, то приложения CI Plus™ не должны запускаться
	SSM=2	RTGraphic (субтитры) отображаются (опционально) при включении любого предпочтения пользователя; если приложение CI Plus™ и субтитры не могут сосуществовать, то субтитры должны быть отключены, а приложение CI Plus™ должно запускаться. Если в соответствии со статусом предпочтения пользователя субтитры выключены, то состояние субтитров должно быть восстановлено при завершении приложения
ADQ	ADQ=0	Хост должен игнорировать исходный объект в сообщении RequestStart. Среда выполнения приложения не должна пытаться загружать и выполнять приложение. Если домен приложения будет поддержан и запуск приложения с этим доменом приложения не вызовет завершения любого другого работающего приложения, то RequestStartAck должен возвратить AckCode 0x01. Если домен приложения не будет поддержан, то RequestStartAck должен возвратить AckCode 0x02. Если домен приложения будет поддержан и запуск приложения с этим доменом приложений может вызвать завершение любого другого работающего приложения, то RequestStartAck должен возвратить AckCode 0x04. Опция ADQ доступна в приложении MMI версии 3 типа 1 и в более поздних версиях. Эта опция не должна использоваться в сеансе приложения MMI версии 1 или 2 типа 1. Опция ADQ доступна в приложении MMI типа 2 (для операции с мультипотоками, определенной в 6.4.6 настоящего стандарта), начиная с версии 1

Значения AckCode, возвращаемые APDU RequestStartAck, определенные в [8] (6.5.3), расширены, уточнены и представлены в таблице 91.

Таблица 91 — Значения AckCode

Значение AckCode	Содержание
0x00	Зарезервировано
0x01	Домен приложения будет поддержан. Работающее приложение не будет терминировано. Если ADQ отсутствует, то среда исполнения приложений должна попытаться загрузить и выполнить исходный объект, указанный в APDU requestStart. Если опция ADQ присутствует и ADQ=0, то домен приложения поддерживается, и запуск приложения при данном домене приложения не вызовет прекращения любого другого запущенного приложения
0x02	Неверный API. Домен приложения не поддерживается
0x03	API занят. Домен приложения поддерживается, но в настоящее время недоступен
0x04	Конфликт API. Если ADQ присутствует и ADQ=0, то домен приложения поддерживается, и запуск приложения с данного домена приложения может привести к завершению какого-либо другого запущенного приложения
0x05	Зарезервировано для использования с ресурсом тип 2 для работы с несколькими потоками в соответствии с 6.4.6, 6.4.6.3 настоящего стандарта
от 0x06 до 0x7F	Зарезервировано
от 0x80 до 0xFF	Домен конкретного API занят. Ответ домена конкретного приложения эквивалентен ответу 0x03, но предоставляется конкретная информация о том, почему среда выполнения занята или недоступна по какой-то другой причине, например, из-за конфликта ресурсов

12.4 Платформа координации приложений

12.4.1 Общие замечания

Реализации Хоста CI Plus™ обеспечивают возможность загрузки и запуска не менее одной среды приложений, которые доставлены или через канал вещания, или при использовании ссылок, сигнализированных в этих каналах. Эти приложения в общем случае называются приложениями вещания.

Между приложениями вещания и приложениями AppMMI CICAM (запущенными с помощью ресурса приложения MMI CI Plus™) могут возникать конфликты.

Платформа (структура) координации приложений, определенная в 12.4 настоящего стандарта, позволяет Хосту принимать решения при запуске приложения в случае конфликта между заявками приложения вещания и приложения AppMMI CICAM с предложением работы в Хосте. Выбор варианта выполняется с учетом предпочтений пользователя, сообщений сигнализации вещания и согласования функций с CICAM. Настоящий стандарт не устанавливает фиксированные параметры приоритета между приложениями AppMMI CICAM и приложениями вещания.

12.4.2 Изменения ссылочных спецификаций

Спецификации, представленные в [1] и [8], допускают различные интерпретации приоритета приложений CICAM AppMMI и приложений вещания, которые хотят работать на Хосте. Изменения этих спецификаций, уточняющие условия разрешения конфликтов, должны быть следующими.

Изменения в [1]:

1) В целом параметры, специфицированные в [1] (раздел 12), определяются только базовым MHEG-5 доменом приложения «CIEngineProfile1». Конкретные изменения представлены в 2), 3), 4);

2) При запуске приложение AppMMI CICAM должно иметь фокус ввода и должно быть показано в передней части приложения вещания либо приложение вещания не должно быть показано. Опционально допускается представление этого условия на собственном графическом плане отображения Хоста, как показано в совокупности концептуальных плоскостей в [1] (рисунок 12.3);

3) В контексте настоящего стандарта не должно применяться требование из [1] (12.3.3) о том, что приложение CI Plus должно иметь фокус ввода и отображать на экране приоритет, при условии возможности одновременной работы приложение CI Plus MMI с другим приложением;

4) Требования [1] (12.6.2.2) должны быть дополнены: приложение AppMMI CICAM, запущенное приложением AppMMI, должно иметь фокус ввода и должно быть показано перед приложением вещания, в противном случае приложение вещания не должно быть показано.

Изменение в [8] — Требование [8] (6.5.2), касающееся уничтожения любого приложения, выполняющегося на платформе приложения, по запросу RequestStart, должно быть аннулировано и заменено в соответствии с 12.4.4 настоящего стандарта.

12.4.3 Механизмы запуска приложений CICAM

12.4.3.1 Общие замечания

Запуск приложений, предоставляемых CICAM, выполняют два независимых механизма.

Первый механизм предназначен для приложений AppMMI CICAM и квалифицирован в контексте платформы координации приложений, представленной в 12.4 настоящего стандарта.

Второй механизм определен для приложения вещания CICAM, его характеристики должны быть в соответствии с 12.4.3.3 настоящего стандарта.

12.4.3.2 Приложения AppMMI CICAM

CICAM использует APDU requestStart, запрашивая Хост о запуске приложения. В этом случае файлы приложения поставляются при использовании ресурса приложения MMI, как определено в [8].

12.4.3.3 Приложения вещания CICAM

12.4.3.3.1 Общие замечания

Приложение вещания CICAM предоставляется системой вспомогательных файлов (специфицированной в разделе 9 настоящего стандарта), которую Хост запускает после получения соответствующего файла. Приложение вещания CICAM может быть запущено в ответ на запрос сигнализации вещания или работающего приложения вещания.

12.4.3.3.2 Сигнализация приложений вещания CICAM в потоке вещания

Способ сигнализации приложения вещания был определен в [23] и [24]. Настоящий стандарт расширяет возможности этого способа и дополняет его вещанием сигналов для приложений вещания CICAM, как определено в приложении Д настоящего стандарта. Сигнализация приложения вещания CICAM на основе указанных выше спецификаций DVB должна выполняться в соответствии с Д.2 приложения Д настоящего стандарта.

Расширенная сигнализация приложений DVB предоставляет механизм, который позволяет сигнализировать соответствующие приоритеты вещания и приложения вещания CICAM. Это позволяет использовать, например, случай, когда версия приложения в канале вещания, предоставленная по умолчанию, имеет более низкий приоритет относительно версии приложения более высокого качества, хранящейся в системе вспомогательных файлов CICAM и дающей приоритет пользователям, имеющим соответствующий CICAM, установленный в Хост.

В тех случаях, когда приложения вещания и приложения вещания CICAM переданы для службы средствами сигнализации приложения вещания, приоритеты этих приложений должны устанавливаться механизмами, определенными в этой системе, или другими альтернативными механизмами в соответствии со спецификацией сигнализации приложения вещания.

Другие решения сигнализации приложения вещания выходят за рамки настоящего стандарта.

12.4.3.3.3 Объявление приложения вещания CICAM

В этом пункте определяется метод, с помощью которого CICAM предъявляет Хосту приложения, хранящиеся на его системе вспомогательных файлов, которая специфицирована в разделе 9 настоящего стандарта.

Каждое приложение вещания CICAM должно предоставляться Хосту файлом AIT XML в соответствии с [23] (5.4), который также хранится на системе вспомогательных файлов с контентом. Файл XML должен удовлетворять следующим условиям:

- файл XML должен содержать записи обнаружения приложений, содержащих один или более элементов приложения с одинаковыми значениями orgId и appId, и все элементы, содержащие тип приложения, соответствующий домену приложения системы вспомогательных файлов, в которой находится файл XML;

- местоположение исходного объекта приложения вещания CICAM должно быть указано с помощью элемента applicationLocation, который должен быть установлен в соответствии с условием, приведенным в настоящем пункте;

- элемент applicationLocation должен ссылаться на исходный объект, предоставляемый CICAM. Если исходный объект обеспечивается системой вспомогательных файлов, то идентификатор домена не должен включаться, а должен быть получен из поля тип приложения;

- элемент type дескриптора приложения должен быть скремблирован. Элемент type дескриптора приложения идентифицирует среду приложения, так как это определено в той же среде или применением технологии промежуточного программного обеспечения, используемой приложением, описанной в 12.5.2 настоящего стандарта. Домен приложения «CIEngineProfile1» определен в [1] (12.2) и, измененный в настоящем стандарте, должен использовать тип приложения «application/vnd.dvb.ciplus.mheg»;

- допускается кодирование иконки элемента;

- если среда приложения определяет использование AIT XML, то использование других полей и элементов должно быть в соответствии со спецификацией среды приложений. В противном случае использование других полей и элементов не нормируется.

Примечание — Использование синтаксиса AIT на основе XML в этих обстоятельствах не предусматривает использования таблицы и синтаксиса AIT в потоке вещания.

Эти файлы XML должны быть включены в «/ dvbapplication /» в системе вспомогательных файлов.

Доставка файлов XML от CICAM к Хосту должна поддерживаться схемой XML, представленной на рисунке 7.

```
<xsd:schema xmlns:xsd="gost.e-ecolog.ru" xmlns:mhp="urn:dvb:mhp:2009"
    xmlns:dvb="urn:dvb:ciplus:xmlait:2013" targetNamespace="urn:dvb:ciplus:xmlait:2013">
        <xsd:import namespace="urn:dvb:mhp:2009" schemaLocation="imports/mis_xmlait.xsd"/>
        <xsd:complexType name="CITransportType">
            <xsd:complexContent>
            <xsd:extension base="mhp:TransportProtocolDescriptorType">
                <xsd:sequence>
                    <xsd:element name="initialObject" type="xsd:anyURI"
                        minOccurs="1" maxOccurs="1" />
                </xsd:sequence>
            </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
    </xsd:schema>
```

Рисунок 7 — Схема XML доставки файлов XML от CICAM к Хосту

12.4.4 Координация приложений вещания и приложений CICAM

12.4.4.1 Общие замечания

В 12.4.4 описывается техническая реализация платформы координации приложений.

Предпочтения пользователя, вводимые на Хосте, должны переопределять любое утверждение, относящееся к приоритету приложения.

Приоритет платформы координации приложений во всех ситуациях, когда Хост изменяет службу DVB, устанавливается в соответствии с 12.4.4.2 настоящего стандарта.

Приоритет платформы приложения в ситуации, когда уже запущено и работает приложение вещания или приложение AppMMI CICAM в результате использования приоритета, определенного выше, устанавливается в соответствии с 12.4.4.3 настоящего стандарта.

Приоритет платформы приложения при выборе виртуального канала CICAM устанавливается в соответствии с 12.4.4.4 и разделом 14 настоящего стандарта.

Пункт 12.4.4.5 устанавливает механизмы платформы завершения приложений.

В целом запущенное приложение, имеющее приоритет, должно иметь фокус ввода, т. е. оно выводится на экран пользователя с последующим взаимодействием с пользователем, происходящим в этом приложении. Детали этого механизма выходят за рамки настоящего стандарта, так как они зависят от

среды приложения, или от включенных сред, или от реализации Хоста в случае координирования приложений в других средах приложения.

Если запускаемое приложение имеет приоритет над уже запущенными приложениями или получает усиление приоритета над другими работающими приложениями, то дальнейшие события определяются способностями сред приложений и Хоста запускать одновременно более одного приложения.

Хосты, которые имеют возможность запускать несколько приложений или сред приложений, могут содержать механизм, позволяющий пользователю переключать фокус между приложениями. Использование этого механизма приведет к потере фокуса ввода приложением, которое ранее было приоритетным.

Любые приложения, которые Хост не может обрабатывать одновременно как приоритетные, должны быть завершены при условии подтверждения завершения пользователем.

Замена среды приложения должна выполняться с учетом того, что процессы завершения одной среды приложения и запуска другой могут занимать значительное время и приводить к отрицательной реакции пользователя. На рынках, где развернуты приложения вещания, рекомендуется использовать в CICAM ту среду приложений для приложений CICAM AppMMI, которая используется на этом рынке для приложений вещания.

12.4.4.2 Изменение службы DVB

После завершения изменения службы DVB и приема соответствующего TS Хост должен контролировать TS для приема сигнализации приложения вещания, интерпретировать сигналы, предусмотренные для этой службы DVB, а также при необходимости запускать приложение вещания.

Хост не должен запускать приложение вещания, если в процессе его запуска возникают следующие события:

- сигнализация приложения вещания отсутствует (в случае DVB это отсутствие ссылки на AIT в PMT и отсутствие AIT непосредственно);
- сигнализация приложения вещания присутствует, но приложение вещания не запускается. Это может произойти из-за Хоста, не способного запустить приложение, или из-за синтаксической ошибки в сигнализации или в переносимом приложении.

Если Хост решает, что приложение вещания не будет запущено и что конфликты с другими запущенными приложениями отсутствуют, то по требованию CICAM могут быть запущены Хостом приложения AppMMI CICAM.

Если CICAM запросит запуск приложения AppMMI прежде, чем Хост завершит проверку, и примет решение не запускать приложение вещания, которое может занять несколько секунд, то Хост не должен отвечать на APDU RequestStart от CICAM, пока эта проверка не будет выполнена. Когда проверка будет выполнена, Хост должен ответить APDU RequestStartAck, описанного в 12.3 настоящего стандарта, в соответствии с тем, будет приложение AppMMI запущено или нет.

Если выполняется автоматический запуск приложения вещания, то Хост должен отклонить запросы CICAM о запуске приложения AppMMI CICAM. Об этом CICAM должен быть извещен APDU requestStartAck со значением AckCode из 0x03 («API занят» в соответствии с 12.3 настоящего стандарта) или со значением «домен конкретного API занят», если это определено для домена приложения, как показано в [8] (6.5.3).

Необходимость сохранения приложения при изменениях службы DVB определяется спецификацией этой среды приложения. Настоящий стандарт определяет необходимость сохранения приложения AppMMI CICAM при изменении службы при использовании решений в сответствии с 13.2.1 настоящего стандарта.

В тех случаях, когда стандартные приложения вещания и приложения вещания CICAM для службы передаются в сигнализации вещания, выбор приложения для запуска должен выполняться по приоритету, указанному в системе сигнализации. Этот механизм расширения приложения сигнализации вещания DVB [23] представлен в приложение Д настоящего стандарта.

12.4.4.3 Условия функционирования приложения AppMMI CICAM

Если приложение AppMMI CICAM функционирует и имеет фокус, то Хост не должен запускать приложение вещания. Исключением является часть процесса, выполняющая изменения службы DVB, как было определено выше, в 12.4.4.2.

Это означает, что допускается запрещать промежуточному ПО приложения вещания следовать за изменениями в сигнализации приложения, например, во время работы приложения AppMMI CICAM. Хост может уведомить пользователя о приложении вещания, ставшего доступным, средствами, которые выходят за рамки настоящего стандарта.

При выполнении приложения вещания может быть запущено приложение вещания CICAM под управлением соответствующего промежуточного ПО среды приложения, работающего в Хосте. Аналогичным образом выполнение CICAM приложения вещания может запустить другое приложение вещания или приложение вещания CICAM.

Механизм, позволяющий приложению вещания запускать приложение AppMMI CICAM, настоящим стандартом не определен.

12.4.4.4 Условия переключения на виртуальный канал CICAM

В 14.2.1 настоящего стандарта предусматривается использование приложений MMI при вводе виртуального канала. Когда пользователь решает начать запуск виртуального канала, приложение, указанное в APDU requestStart, должно иметь фокус пользователя и любое работающее вещание или приложение CICAM может потерять фокус или терминироваться в зависимости от возможностей Хоста.

12.4.4.5 Условия ввода Хостом меню CICAM

В результате выбора пользователя Хост отправляет к CICAM APDU enter_menu, чтобы пользователь мог войти в меню CICAM.

Если CICAM отвечает на APDU enter_menu APDU requestStart, то приложение, идентифицированное в APDU requestStart, должно иметь фокус пользователя и любое функционирующее вещание в противном случае приложение CICAM может потерять фокус или завершиться в зависимости от возможностей Хоста.

Примечание — В зависимости от реализации CICAM может запустить высокий уровень MMI после получения от Хоста APDU enter_menu.

12.4.4.6 Условия завершения приложения

После изменения службы среда приложения не должна выполнять требования, установленные для Хоста в [1] или [8], для завершения приложения AppMMI CICAM при следующих условиях:

- среда приложения поддерживает идентификацию приложений;
- среда приложения поддерживает сохранение приложения при изменении службы;
- приложению AppMMI CICAM разрешается работать в новой службе согласно приложению вещания, сигнализированному в этой службе.

12.5 Среда приложения Хоста

12.5.1 Общие замечания

В 12.5.2 настоящего стандарта определяется спецификации среды приложения, которая использует платформу для предоставления и запуска приложений из CICAM.

В 12.5.3 настоящего стандарта перечислены методы, которыми CICAM определяет среды приложения, поддерживаемые Хостом.

12.5.2 Предоставление приложения на CICAM

При предоставлении приложения на CICAM система промежуточного ПО (или среда приложения) должна определить свой идентификатор и идентификатор объекта, запускающего приложение.

Промежуточное ПО идентифицируется с помощью «AppDomainIdentfiier» в соответствии с [8] (6.5.2) и «DomainIdentifier» (см. 9.2 настоящего стандарта). Промежуточное ПО должно задавать значения для этих двух идентификаторов. Для браузера на основе MHEG CI Plus™ значением «AppDomainIdentfiier» и «DomainIdentfiier» является «CIEngineProfile1».

Объект, который разрешает Хосту запускать приложение в соответствии с [8] (6.5.2), называется «InitialObject». Промежуточное ПО должно определить соответствующее значение для этого объекта. Для браузера MHEG на основе MHEG CI Plus™ «InitialObject» этим значением является URI. Если промежуточное ПО работает с URI, то оно должно создать URI, который указывает на CICAM. В случае MHEG URI, указывающим на CICAM, является CI://.

12.5.3 Определение среды приложения, поддерживаемой Хостом

CICAM может обнаруживать среду приложения, поддерживаемую Хостом двумя способами:

- по значению AckCode в APDU FileSystemAck, полученному от Хоста для каждого предложения файловой системы в APDU FileSystemOffer, который CICAM передал Хосту. Эта операция описывается в разделе 9 настоящего стандарта;
- при использовании APDU RequestStart с параметром «ADQ=0» и известным идентификатором домена приложения среды приложения, поддержка которой запрашивается в соответствии с 12.3 настоящего стандарта.

13 Управление Хостом DVB

13.1 Общие замечания

Этот раздел определяет параметры механизма, с помощью которого пользователь может выбрать запуск сеанса приложения CICAM или сеанса MMI высокого уровня. Сеансы приложения CICAM или MMI высокого уровня предлагаются как вход в виртуальный канал в наборе каналов Хоста. При выборе пользователем виртуального канала запускается сеанс приложения CICAM или сеанс MMI высокого уровня.

Версия 3 ресурса управления Хостом DVB (с ID ресурса 0x00200043) является расширением версии 2 и добавляет следующие возможности ресурса управления Хостом DVB:

- возможность настраиваться непосредственно на номер логического канала, включая виртуальный канал CICAM и службы, доставляемые по IP-соединениям;
- возможность настраиваться непосредственно на местоположение IP;
- возможность настраиваться непосредственно на DVB в трех форматах вещания (спутниковое, кабельное, эфирное);
- возможность сохранять приложение, запущенное приложением MMI, после замены канала, инициированного CICAM;
- возможность выполнить запрос доступных систем доставки, поддерживаемых Хостом.

CICAM не должен держать этот ресурс открытым, если он не выдан по одному из APDU с запросом настройки.

13.2 APDU управления Хостом DVB

13.2.1 APDU tune_broadcast_req

APDU tune_broadcast_req определен в [1] (14.6.2.1), его расширения представлены в таблице 92. Хост должен ответить на этот APDU APDU tune_reply.

Таблица 92 — Синтаксис APDU tune_broadcast_req

Синтаксис	Количество битов	Мнемоника
tune_broadcast_req() {		
tune_broadcast_req_tag	24	uimsbf
length_field()		
reserved	5	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
pmt_flag	1	uimsbf
service_id	16	uimsbf
reserved	4	uimsbf
descriptor_loop_length	12	uimsbf
for (i=0; i<n; i++){		
descriptor()		
}		
if (pmt_flag == 1) {		
program_map_section()		
}		
}		

Семантика полей APDU tune_broadcast_req:

- **tune_quietly_flag**: флаг сообщает о необходимости информировании пользователя об изменении службы;

- **keep_app_running_flag**: флаг сообщает, должна ли затребованная настройка сохранить запущенное неидентифицированное приложение, конфликтующее со службой настройки. Значение 0b1 указывает, что настройка не должна быть разрушающей и должна сохранять приложение, работающее в процессе настройки и после настройки, при следующих ограничениях:

- если приложение сигнализировано вещанием, которое Хост может запустить на службу, на которую настраивается CICAM, то любое запущенное неидентифицированное приложение должно быть или завершено, или потерять фокус, в зависимости от возможностей Хоста, как описано в 12.4.4.1 настоящего стандарта;

- если работа приложения прекращается в связи с выполнением предыдущего условия, то tune_quietly_flag и keep_app_running_flag должны быть проигнорированы;

- запуск приложения, имеющего идентификатор, должен соблюдать правила жизненного цикла, допустимые для соответствующей среды приложения. Дополнительные сведения о координации приложений приведены в 12.4.4 настоящего стандарта.

Если CICAM завершает выполнение приложения AppMMI из-за сигнализированного приложения вещания, то Хост должен отправить к CICAM APDU AppAbortRequest.

Другие поля должны определяться как параметры в APDU tune_broadcast_req в соответствии с [1] (14.6.2.1).

13.2.2 APDU tune_triplet_req

APDU tune_triplet_req повторяет APDU tune, определенный в [2] (8.5.1.1), но в нем исключен параметр network_id и добавлен параметр dsd_type, для того чтобы CICAM мог установить систему вещания, из которой должна быть получена служба. Кроме того, добавляется tune_quietly_flag и keep_app_running_flag, определенные в 13.2.1 настоящего стандарта. Хост должен ответить на это APDU tune_reply.

Синтаксис APDU tune_triplet_req показан в таблице 93.

Таблица 93 — Синтаксис APDU tune_triplet_req

Синтаксис	Количество битов	Мнемоника
tune_triplet_req () {		
tune_lcn_triplet_tag	24	uimsbf
length_field()		
reserved	6	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
delivery_system_descriptor_tag	8	uimsbf
if (delivery_system_descriptor_tag == 0x7f){		
descriptor_tag_extension	8	uimsbf
}		
else {		
reserved	8	uimsbf
}		
}		

Семантика полей APDU tune_triplet_req:

- **tune_triplet_req_tag**: тег 24 бита имеет значение 0x9F8409;

- **tune_quietly_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;

- **keep_app_running_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;
- **original_network_id**: поле 16 битов содержит идентификатор сети original_network_id запрошенной службы;
- **transport_stream_id**: поле 16 битов содержит идентификатор потока transport_stream_id запрошенной службы;
- **service_id**: поле 16 битов содержит идентификатор запрошенной службы service_id;
- **delivery_system_descriptor_tag**: поле 8 битов определяет тег системы доставки затребованной службы. Значения тега определяются в [3];
- **descriptor_tag_extension**: поле 8 битов совместно с delivery_system_descriptor_tag определяет тип системы доставки затребованной службы. Значения тега определяются в [3].

13.2.3 APDU tune_lcn_req

APDU tune_lcn_req передается от CICAM к Хосту для настройки Хоста непосредственно на службу по LCN. Его синтаксис приведен в таблице 94. Хост должен ответить APDU tune_reply.

Таблица 94 — Синтаксис APDU tune_lcn_req

Синтаксис	Количество битов	Мнемоника
tune_lcn_req () {		
tune_lcn_req_tag	24	uimsbf
length_field()		
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
logical_channel_number	14	uimsbf
}		

Семантика полей APDU tune_lcn_req:

- **une_lcn_req_tag**: поле 24 бита содержит тег, который имеет значение 0x9F8407;
- **tune_quietly_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;
- **keep_app_running_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;
- **logical_channel_number**: поле 14 битов указывает логический номер канала, присвоенный службе, может содержать четыре десятичных цифры.

13.2.4 APDU tune_ip_req

APDU tune_ip_req передается от CICAM к Хосту для того, чтобы предоставить Хосту местоположение запрашиваемой Хостом одиночной IP службы. Его синтаксис приведен в таблице 95. Хост должен ответить на этот APDU APDU tune_reply.

Таблица 95 — Синтаксис APDU tune_ip_req

Синтаксис	Количество битов	Мнемоника
tune_ip_req () {		
tune_ip_req_tag	24	uimsbf
length_field()		
reserved	2	uimsbf
tune_quietly_flag	1	uimsbf
keep_app_running_flag	1	uimsbf
service_location_length	12	uimsbf
for (i=0; i<N; i++) {		

Окончание таблицы 95

Синтаксис	Количество битов	Мнемоника
service_location_data	8	uimsbf
}		
}		

Семантика полей APDU tune_ip_req:

- **tune_lcn_req_tag**: поле 24 бита содержит тег, который имеет значение 0x9F8408;
- **tune_quietly_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;
- **keep_app_running_flag**: флаг в соответствии с синтаксисом APDU tune_broadcast_req в 13.2.1 настоящего стандарта;
- **service_location_length**: поле 12 битов содержит длину поля ServiceLocation;
- **service_location_data**: поле 8 битов представляет текстовую строку, которая описывает допустимое описание XML, содержащее один элемент ServiceLocation, соответствующий схеме XML, определенный в приложении Г настоящего стандарта.

13.2.5 APDU tuner_status_req

APDU tuner_status_req передается от CICAM к Хосту для определения системы доставки вещания, поддерживаемой Хостом, и возможности настройки в зависимости от типа системы доставки. Синтаксис APDU tuner_status_req приведен в таблице 96. Хост должен ответить APDU tuner_status_reply.

Таблица 96 — Синтаксис APDU tuner_status_req

Синтаксис	Количество битов	Мнемоника
tuner_status_req() {		
tuner_status_req_tag	24	uimsbf
length_field()= 0		
}		

Семантика поля в APDU tuner_status_req:

tuner_status_req_tag: поле 24 бита содержит тег, который имеет значение 0x9F840A.

13.2.6 APDU tuner_status_reply

APDU tuner_status_reply передается от Хоста в ответ на APDU tuner_status_req. Хост использует этот APDU, чтобы сообщить CICAM, какие системы доставки вещания он поддерживает, на какие из них он может быть успешно настроен и будет ли он поддерживать службы IP. Синтаксис APDU показан в таблице 97.

Таблица 97 — Синтаксис APDU tuner_status_reply

Синтаксис	Количество битов	Мнемоника
tuner_status_reply() {		
tuner_status_reply_tag	24	uimsbf
length_field()		
IP_tune_capable_flag	1	uimsbf
num_dsd	7	uimsbf
for (i=0; i<num_dsd; i++){		
reserved	7	uimsbf
connected_flag	1	uimsbf
delivery_system_descriptor_tag	8	uimsbf
if (delivery_system_descriptor_tag == 0x7f){		

Окончание таблицы 97

Синтаксис	Количество битов	Мнемоника
descriptor_tag_extension	8	uimsbf
}		
else {		
reserved	8	uimsbf
}		
}		
}		

Семантика полей APDU tuner_status_req:

- **tuner_status_reply_tag**: поле 24 бита, тег имеет значение 0x9F840B;
- **IP_tune_capable_flag**: этот флаг должен быть установлен в 0b1, если Хост способен принять APDU tune_ip_req для IP поставляемых служб, и должен быть установлен в 0b0, если Хост не в состоянии выполнить эту настройку;
- **num_dsd**: поле 7 битов определяет количество используемых типов DSD, поддерживаемых Хостом. Если один физический тюнер содержит несколько DSD, то должны быть перечислены все DSD;
- **connected_flag**: флаг определяет возможность Хоста поддержки одним тюнером соединения этого типа dsd с сетью вещания. Если возможность настройки существует, то устанавливается 0b1. Значение 0b0 указывает, что возможность настройки не существует;
- **delivery_system_descriptor_tag**: поле 8 битов определяет системы доставки, которые Хост поддерживает, и доступен для фоновых настроек. Значения тега определяются в [3];
- **descriptor_tag_extension**: поле 8 битов вместе с полем delivery_system_descriptor_tag определяет системы доставки, поддерживаемые Хостом и доступные для фоновых настроек. Значения тега определяются в [3].

14 Виртуальный канал CICAM

14.1 Введение

В этом разделе определяется механизм, позволяющий пользователю выбирать запуск приложения или запуск сеанса MMI высокого уровня, которые предлагаются CICAM. Приложение или сеанс MMI высокого уровня предлагаются в форме ввода в виртуальный канал в совокупности каналов Хоста. После выбора пользователем виртуального канала запускается приложение CICAM или сеанс MMI высокого уровня.

Версия 3 и более ранние версии ресурса информации приложения уже содержат APDU запроса запуска сеанса MMI от CICAM, настоящий стандарт добавляет новый APDU для запуска специального виртуального канала CICAM, предоставляемого Хосту, который будет включаться в список каналов Хоста. Ресурс информации приложения в результате такого расширения обозначается как версия 4. Параметры расширенного ресурса информации, версия 4 и правила его применения определены в 14.2 настоящего стандарта.

CICAM информирует Хост о виртуальном канале включением его в NIT CICAM, используя новую версию профиля оператора, который нормирован в разделе 15 настоящего стандарта.

14.2 Расширенный ресурс информации приложения, версия 4

14.2.1 Общие замечания

Расширенный ресурс информации приложения, версия 4 (идентификатор ресурса 0x00020044), содержит новый APDU enter_cicam_channel, который запрашивает запуск приложения, ассоциированного с виртуальным каналом CICAM.

Обычный механизм уведомления CICAM о выборе пользователя передачей CA_PMT выбранной службы в данном случае неприменим. После выбора пользователем виртуального канала Хост не должен посылать CA_PMT, и CICAM не должен ожидать от Хоста CA_PMT или TS.

После приема APDU enter_cicam_channel CICAM запрашивает или приложение MMI, или сеанс высокого уровня MMI. Сеанс приложения MMI должен соответствовать идентификатору домена приложения,

связанного с виртуальным каналом CICAM и определенного во время установления профиля оператора. Хост должен гарантировать создание такого сеанса после передачи APDU enter_cicam_channel.

Хост должен послать APDU enter_cicam_channel только тогда, когда виртуальный канал CICAM будет создан и предоставлен пользователю.

При приеме APDU enter_cicam_channel CICAM должен остановить дескремблирование локальных TS, указанных параметром LTS_id, которые он все еще может получать.

14.2.2 APDU enter_cicam_channel

Синтаксис APDU enter_cicam_channel показан в таблице 98.

Таблица 98 — Синтаксис APDU enter_cicam_channel

Синтаксис	Количество битов	Мнемоника
enter_cicam_channel () {		
enter_cicam_channel_tag	24	uimsbf
length_field()= 1		
LTS_id	8	uimsbf
}		

Семантика полей APDU enter_cicam_channel:

- **enter_cicam_channel_tag**: поле 24 бита, тег имеет значение 0x9F8025;
- **length_field**: поле указывает длину полезной нагрузки APDU в соответствии с форматом BER ASN.1, определенным в [2] (8.3.1);
- **LTS_id**: поле 8 битов содержит идентификатор локального TS.

15 Профиль оператора

15.1 Вводные замечания

Профиль оператора версии 2 (идентификатор ресурса 0x008F1002) является расширением профилей оператора версии 0 и 1 с добавлением profile_type, который позволяет Хосту создавать в расширенной NIT CICAM объединенный список логических каналов из SI, полученной в процессе вещания, и из дополнительных служб, предоставленных CICAM, сигнализированных в расширенной NIT CICAM.

Профиль оператора версии 2 дополняется следующими возможностями:

- способностью CICAM добавлять службы IP в список логических каналов Хоста. При этом в первый цикл расширенной NIT CICAM добавляется дескриптор uri_linkage_descriptor. Процесс обработки этого дескриптора определен в 15.4.2 настоящего стандарта. Дескриптор uri_linkage_descriptor переносит URI, который связывает онлайн SDT (OSDT), содержащую описание доставленных служб IP;
- способностью CICAM добавлять виртуальный канал в список логических каналов Хоста. Виртуальный канал CICAM определен в разделе 14 настоящего стандарта. Признаками виртуального канала CICAM являются: LCN, имя службы и информация о событии. Эти признаки сообщены в дескрипторе виртуального канала CICAM, который переносится в расширенной NIT CICAM. Дескриптор виртуального канала CICAM определен в 15.3 настоящего стандарта.

15.2 Параметры профиля оператора, тип 2

15.2.1 Вводные замечания

В дополнение к profile_type = 0 и profile_type = 1, специфицированных в [1], новый profile_type = 2 является профилем операции, при которой Хост создает список логических каналов из SI вещания и NIT CICAM. SI вещания имеет приоритет над NIT CICAM. Хост сообщает о любых коллизиях между SI вещания и NIT CICAM, позволяя CICAM предоставить разрешение этих коллизий обновленной NIT CICAM. Этот тип профиля позволяет CICAM собирать информацию о правах. Profile_type = 2 позволяет NIT CICAM выполнять службы, предоставляемые по IP и по виртуальному каналу CICAM.

15.2.2 Инициализация и открытие профиля оператора

CICAM, имеющий profile_type = 2, поддерживает частичный список каналов, который будет интегрироваться со списком каналов, созданным из SI вещания. Хост определит списки каналов вещания до установления списка каналов профиля оператора. После этого Хост сообщает о коллизиях, возни-

кающих в затребованных LCN от CICAM, что позволяет CICAM обеспечить альтернативный LCN для каналов профиля оператора.

Хост должен открыть сеанс ресурса профиля оператора при запросе сеанса CICAM. CICAM должен отправить APDU operator_status для передачи Хосту информации о профиле оператора.

CICAM должен сохранять информацию о профиле оператора в соответствии с [1] (14.7.4.1.2).

Открытие профиля оператора для profile_type 2 выполняет механизм, аналогичный механизму profile_type типа 1 в соответствии с [1]. CICAM может потребовать сканирования профилей, чтобы инициализировать профиль оператора и создать NIT CICAM.

Если Хост поддерживает системы доставки, представленные через APDU operator_status, то он должен определить поддержку profile_type = 2 и установить каналы профиля оператора в список логических каналов Хоста. Эти каналы связаны с системой доставки, в соответствии с указанием системы доставки в ближайшее время при минимальном ущербе для пользователя. CICAM должен обеспечивать однозначную идентификацию каждой службы в списке по ее LCN, а также и в том случае, когда в подсказке системы сообщается несколько систем, содержащих службы из различных источников системы доставки. APDU operator_status в соответствии с настоящим стандартом сигнализирует о предоставляемых службах IP в профиле оператора типа 2, как указано в 15.4.1 настоящего стандарта.

Хост может отказаться от хранения профиля оператора, если дескриптор доставки профиля оператора отличается от заранее выбранного дескриптора доставки службы вещания.

Хост должен сохранять виртуальный канал CICAM, если среда приложений для канала поддерживается промежуточным ПО Хоста. Если Хост не поддерживает запрошенную среду приложений, то он должен сообщить об этом CICAM так, чтобы CICAM мог попытаться выбрать другую среду приложения для виртуального канала.

Если Хост поддерживает дескремблирование DRM, выполняемое CICAM (доставка IP Хосту в режиме проигрывателя), или гибридное соединение LSC (поставка IP CICAM в режиме проигрывателя), то он должен сохранять список доступных доставленных IP служб.

15.2.3 Возможность работы Хоста с профилями оператора разных типов

Не допускается изменение Хостом профиля оператора (изменение профиля на тип 2 или изменение профиля типа 2 на тип 0 или тип 1). Поэтому Хост не должен сообщать CICAM, что он находится в профиле типа 2.

15.2.4 Управление коллизиями между логическими каналами

При приеме NIT CICAM Хост должен добавить службы профиля оператора в список логических каналов. Если Хост будет видеть, что LCN двух каналов не уникальны, то он должен сохранить службу вещания в своем списке логических каналов. Хост должен сообщить CICAM, какой LCN и какая служба от NIT CICAM не уникальны в списке вещания, одновременно предоставляя ближайший доступный свободный LCN. CICAM должен обновить NIT CICAM с новой версией и должен выделить новый LCN для рассматриваемой службы.

Для CICAM, работающих в profile_type типа 2, Хост должен использовать APDU operator_nit_management для того, чтобы сообщать CICAM о любом LCN, который не уникален в NIT CICAM.

15.3 Параметры виртуального канала CICAM

15.3.1 Дескриптор виртуального канала CICAM

Дескриптор виртуального канала CICAM, синтаксис которого определен в таблице 99, должен присутствовать в первом цикле NIT CICAM и должен соответствовать частному спецификатору данных CI Plus™. Виртуальный канал CICAM может интерпретироваться Хостом как служба вещания данных. Дескриптор виртуального канала предоставляет LCN, имя службы, метаданные события, идентификатор домена приложения службы.

NIT CICAM не должна содержать больше одного дескриптора виртуального канала.

Таблица 99 — Синтаксис дескриптора виртуального канала CICAM

Синтаксис	Количество битов	Мнемоника
cicam_virtual_channel_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	2	uimsbf

Окончание таблицы 99

Синтаксис	Количество битов	Мнемоника
logical_channel_number	14	uimsbf
service_provider_name_length	8	uimsbf
for (i=0; i<n; i++) {		
service_provider_name_char	8	uimsbf
}		
service_name_length	8	uimsbf
for (i=0; i<n; i++) {		
service_name_char	8	uimsbf
}		
event_information_length	8	uimsbf
for (i=0; i<n; i++) {		
event_information_char	8	uimsbf
}		
AppDomainIdentifierLength	8	uimsbf
for (i=0; i<n; i++) {		
AppDomainIdentifier	8	bslbf
}		
}		

Параметры полей дескриптора виртуального канала должны быть следующими:

- **descriptor_tag**: поле 8 битов является тегом, имеет значение 0xCF и должно быть интерпретировано только в контексте частного дескриптора спецификатора данных CI Plus™;
- **logical_channel_number**: поле 14 битов указывает номер логического канала соответствующей службы, должны использоваться только 4 десятизначных номера каналов от 0 до 9999. Логические номера каналов должны быть уникальными во всей NIT CICAM, т.е. номер логического канала должен быть использован только один раз;
- **service_provider_name_length**: поле 8 битов определяет количество байтов, следующих после поля service_provider_name_length, которые описывают символы имени провайдера служб;
- **service_provider_name_char, service_name_char, event_information_char**: поля по 8 битов содержат строки символьных полей, определяющих имя провайдера, имя службы или информацию о событии. Информация о тексте кодируется при использовании наборов символов и методов, описанных в [3], в соединении с APDU operator_info;
- **service_name_length**: поле 8 битов определяет количество байтов, следующих за полем service_name_length и описывающих символы имени службы;
- **event_information_length**: поле 8 битов определяет длину в байтах элемента текста, который может использоваться для заполнения информационного поля EPG;
- **AppDomainIdentifierLength**: поле 8 битов определяет длину в байтах строки домена приложения;
- **AppDomainIdentifier**: поле 8 битов определяет необходимый домен приложения в конкретной предметной области. Идентификатор домена приложения используется и как идентификатор ресурса приложения MMI.

Если Хост не поддерживает идентификатор домена приложения, запрашиваемого CICAM, то он должен сообщить об этом CICAM, используя APDU operator_nit_management для того, чтобы CICAM выбрал другую среду приложений для канала. CICAM информирует Хост о новой среде приложения, обеспечивая новую NIT CICAM, при обновлении версии NIT.

15.3.2 Выбор виртуального канала CICAM

Хост должен использовать ресурс информации приложения, версия 4, описанный в разделе 14 настоящего стандарта, для того чтобы информировать CICAM о том, что виртуальный канал CICAM был выбран пользователем.

15.3.3 Правила решения проблем обслуживания NIT CICAM

При появлении проблем, возникающих при обслуживании NIT CICAM, Хост передает APDU operator_nit_management. При приеме этого APDU CICAM может обновить NIT CICAM, чтобы разрешить проблему, о которой сообщает Хост. Если CICAM не предоставит обновленную NIT CICAM, то Хост не должен хранить службу, у которой есть проблемы в списке логических каналов.

Если этот APDU не направляется Хостом в ответ на передачу от CICAM APDU operator_nit, то CICAM должен предположить, что все затребованные LCN доступны и что Хост поддерживает указанный домен приложения.

Синтаксис APDU управления NIT оператора (operator_nit_management) должен быть в соответствии с таблицей 100.

Таблица 100 — Синтаксис APDU operator_nit_management

Синтаксис	Количество битов	Мнемоника
operator_nit_management() {		
operator_nit_management_tag	24	uimsbf
length_field()		
error_field	8	uimsbf
number_of_services	8	uimsbf
for (i=0; i<number_of_services; i++) {		
channel_issue_type	2	uimsbf
logical_channel_number	14	uimsbf
service_name_length	8	uimsbf
for (i=0; i<service_name_length; i++){		
char	8	uimsbf
}		
if (channel_issue_type == 0) {		
free_lcn_number	8	uimsbf
for (i=0; i<free_lcn_number; i++) {		
reserved	2	uimsbf
logical_channel_number	14	uimsbf
}		
}		
}		
}		

Семантика полей APDU operator_nit_management:

- **operator_nit_management_tag**: поле 24 бита содержит тег со значением 0x9F9C0F;
- **error_field**: поле 8 битов указывает на проблему, возникающую при работе с NIT CICAM. Допустимые значения и содержание проблем определены в таблице 101.

Таблица 101 — Значения поля error_field

Значение	Содержание проблем
0x00	Нет проблем. OSDT и/или NIT найдены
0x01	Ошибка NIT: NIT не может быть применена. Службы от NIT CICAM не были добавлены в список логических каналов Хоста
0x02	Ошибка OSDT: URL для OSDT не может быть найден. Службы от OSDT не были добавлены в список логических каналов Хоста, но служба от NIT CICAM будет добавлена, если по APDU operator_nit_management не обозначено иное решение
0x03	Ошибка OSDT. Не поддерживается
от 0x04 до 0xFF	Зарезервировано

- **number_of_services**: поле 8 битов идентифицирует количество каналов от NIT CICAM, которые не сохранены в списке логических каналов Хоста из-за проблемы, детализированной в поле channel_ issue_type;

- **service_name_length**: поле 8 битов определяет количество байтов, которые следуют за полем service_name_length для символов имени службы. Имя службы должно соответствовать предоставленному CICAM в APDU, которому соответствует APDU operator_nit_management. Происхождение имени службы может быть одним из следующих:

- имя службы предоставляется дескриптором ciplus_service_descrptor во втором цикле NIT CICAM;

- имя службы предоставляется дескриптором cicam_virtual_channel_descriptor в первом цикле NIT CICAM; первое имя службы, найденное в списке возможных имен службы в OSDT, указано в первом NIT CICAM loop;

- **logical_channel_number**: поле 14 битов указывает номер логического канала, назначенный службе (NIT CICAM), который не был сохранен в списке логических каналов Хоста;

- **channel_issue_type**: поле 2 бита содержит описание проблем обслуживания NIT CICAM. Значения поля и описание проблем обслуживания NIT CICAM показаны в таблице 102;

Таблица 102 — Значения channel_issue_type и описания проблем обслуживания NIT CICAM

Значение	Описание проблем обслуживания NIT CICAM
0x0	Коллизия LCN: это означает, что LCN, предоставленный NIT CICAM, не является уникальным и имеет конфликт с LCN, предоставляемым информацией службы вещания
0x1	Домен приложения: это означает, что применение домена, запрошенного в NIT CICAM, не поддерживается Хостом
0x2	Служба не сохраняется: это означает, что механизм доставки не поддерживается
0x3	Зарезервировано

- **free_lcn_number**: поле 8 битов определяет количество свободных значений LCN, на которые может быть перемещена служба, не вызывая коллизий. Хост должен обеспечивать (при наличии) первый доступный LCN ниже и выше запрашиваемого LCN из NIT CICAM.

15.4 Службы, предоставленные IP

15.4.1 Расширенный APDU operator_status

APDU operator_status расширяется (опционально) добавлением поля delivery_system_hint в зарезервированные биты профиля оператора в версии 1 в соответствии с [1] (таблица 14.42). В таблице 103 показаны значения delivery_system_hint для профиля оператора версии 2.

Таблица 103 — Значения delivery_system_hint

Значение	Описание
0b0001	Это кабельная сеть, возможно в формате DVB-C и/или DVB-C2
0b0010	Это спутниковая сеть, возможно в формате DVB-S и/или DVB-S2

Окончание таблицы 103

Значение	Описание
0b0100	Это сеть эфирного вещания, возможно в формате DVB-T и/или DVB-T2
0b1000	Это IP-сеть

15.4.2 Открытие и обновление IP служб

Для служб, предоставляемых IP, NIT CICAM в первом цикле может включать в себя не менее одного дескриптора uri_linkage_descriptors. Uri_linkage_descriptor определяется в [3] (6.4.14).

Для использования в данном контексте предназначены только uri_linkage_descriptors, содержащие uri_linkage_type со значением 0x00. Uri_linkage_descriptor несет единственный URI, который обеспечивает локацию списка служб, предоставляемых IP. Этот список, объявленный в структуре XML данных, называемый онлайн SDT (OSDT), содержит один элемент IPServiceList. Схема OSDT XML определяется в приложении Г настоящего стандарта. Полная нормативная схема XML доступна в файле osdt_v1.1.1.xsd в архиве ts_103205v010101p0.zip, который сопровождает настоящий стандарт.

Службы, перечисленные в OSDT, могут иметь следующие особенности:

- службы, для которых представлена информация о местоположении, но ни одно из этих местоположений не поддерживается Хостом или CICAM. Хост не может добавить эти службы в список каналов;
- службы, для которых информация о местоположении предоставлена, но Хост не поддерживает ни одно из этих местоположений, в то время как CICAM поддерживает по крайней мере одно из них. Хост определяет, что CICAM поддерживает местоположение службы при использовании APDU CICAM_player_verify_req. В этом случае при выборе служб Хост должен инициировать поиск службы, используя APDU CICAM_player_play_req, представленный в 8.7.9 настоящего стандарта;
- службы, для которых обеспечено предоставление не менее одного местоположения, и Хост поддерживает не менее одного из них. При выборе службы Хост может использовать режим семпла для передачи контента к CICAM в соответствии с 7.2 настоящего стандарта, если служба скремблирована;
- службы, для которых информация о местоположении не предоставляется. Если информация о местоположении приложения предоставлена, то это службы данных (если в поле ServiceType не указывается иное). При выборе службы Хост должен выполнить запуск приложения, однако правила запуска приложения настоящим стандартом не нормируются.

Если URI в uri_linkage_descriptor является «cicam:», то Хост должен использовать APDU operator_osdt_request с запросом OSDT от CICAM. В этом случае Хост должен игнорировать поле min_polling_interval в uri_linkage_descriptor. Если URI определяет другую схему (например, «gost.e-ecolog.ru path/list.xml»), то Хост должен получить OSDT с помощью указанного URI и должен проверять наличие обновлений в OSDT не чаще, чем через интервал опроса, указанный в поле in_polling_interval. Схемы URI, отличающиеся от «http:» или «https:», Хост может игнорировать.

В каждый файл OSDT Хост должен добавлять все перечисленные службы IP таким же образом, как он это делает для других служб, на которые ссылается NIT CICAM. Этот процесс может включать отбраковывание служб, не совместимых с Хостом и имеющих внутренние конфликты LCN.

Всякий раз когда CICAM объявляет новую версию CICAM_NIT, Хост должен восстановить OSDT независимо от ее местоположения, если он сохраняет доставляемые CICAM IP-службы.

15.4.3 APDU operator_osdt_request

Хост посылает этот APDU в CICAM с запросом текущего файла OSDT. Синтаксис APDU operator_osdt_request приведен в таблице 104. CICAM должен ответить APDU operator_osdt_reply, возвращая OSDT к Хосту.

Таблица 104 — Синтаксис APDU operator_osdt_request

Синтаксис	Количество битов	Мнемоника
operator_osdt_request() {		
operator_osdt_request_tag	24	uimsbf
length_field()		
}		

Поле **operator_osdt_request_tag** 24 бита является тэгом и имеет значение 0x9F9C0D.

15.4.4 APDU operator_osdt_reply

Этот APDU CICAM направляет к Хосту в ответ на APDU operator_osdt_request. В случае успешного обмена он должен содержать последнюю версию файла OSDT. Синтаксис APDU operator_osdt_reply приведен в таблице 105.

Таблица 105 — Синтаксис APDU operator_osdt_reply

Синтаксис	Количество битов	Мнемоника
operator_osdt_reply () {		
operator_osdt_reply_tag	24	uimsbf
length_field()		
osdt_file_data_length	32	uimsbf
for (i=0; i< osdt_file_data_length; i++) {		
osdt_file_data_byte	8	uimsbf
}		
}		

Семантика полей operator_osdt_reply:

- **operator_osdt_reply_tag**: поле 24 бита является тегом и имеет значение 0x9F9C0E;
- **file_data_data_length**: поле 32 бита содержит длину файла OSDT в байтах. Если CICAM не имеет OSDT для возврата, то osdt_file_data_length должен быть установлен в ноль;
- **osdt_file_data_byte**: поле 8 битов содержит данные файла XML OSDT.

16 Параметры интерфейса человек-машина высокого уровня

Реализация на Хосте MMI высокого уровня не должна приводить к прерыванию работы запущенных приложений.

Дополнительные требования, уточняющие минимальные возможности Хоста, связанные с MMI высокого уровня, в соответствии с [2] (8.6), должны быть следующими:

- Хост должен отображать не менее 40 символов в строке (элементы текста, титры, субтитры, нижняя линия);
- Хост должен отображать объект menu, содержащий до 254 элементов;
- Хост должен иметь возможность отображать объект list, содержащий до 254 элементов;
- при представлении объектов menu или list Хост должен отображать не менее 5 элементов одновременно и осуществлять прокручивание объекта, если не все элементы размещены на отображаемой области.

Приложение А
(рекомендуемое)

Идентификация и местоположение дескрипторов

Таблица А.1 содержит:

- список дескрипторов, объявленных или определенных в настоящем стандарте и в предыдущих версиях спецификации CI Plus™;
- значения тегов дескрипторов со ссылками на соответствующие нормативные документы, определяющие каждый дескриптор;
- наименования таблиц, в которых в основном размещаются дескрипторы.

Таблица А.1 — Определения и местоположения дескрипторов

Дескриптор	Значение тега	Определено в нормативном документе	Возможное местоположение
ciplus_content_label_descriptor	0xCB	[1] (N.1.2.4)	CICAM NIT
ciplus_service_descriptor	0xCC	[1] (N.1.2.3)	CICAM NIT
ciplus_initialization_vector_descriptor	0xCD	7.5.5.9.2 настоящего стандарта	SST
ci_protection_descriptor	0xCE	[1] (10.1.1.1)	SDT
ciplus_key_identifier_descriptor	0xCF	7.5.5.9.3 настоящего стандарта	SST
Зарезервировано	от 0xD0 до 0xEF	—	—
Определено Хостом	от 0xF0 до 0xEF	—	SST, SET, FLT
Запрещено	0xFF	—	—

Приложение Б
(обязательное)

Полный набор ресурсов

В таблице Б.1 приведен полный набор ресурсов, включающий ресурсы, предусмотренные в предыдущих спецификациях CI V1.3 Plus™, и новые ресурсы и типы ресурсов, которые определены в настоящем стандарте.

Таблица Б.1 — Полный набор ресурсов

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
Администратор ресурса	00 01 00 41	1	1	1	profile_enq	9F 80 10	⟷		[2]
					profile	9F 80 11	⟷		
					profile_change	9F 80 12	⟷		
	00 01 00 42	1	1	2	profile_enq	9F 80 10	⟷		[8]
					profile	9F 80 11	⟷		
					profile_change	9F 80 12	⟷		
					CICAM_id_send	9F 80 13	←		
					CICAM_id_command	9F 80 14	→		
Информация приложения	00 02 00 41	2	1	1	application_info_enq	9F 80 20	→		[2]
					application_info	9F 80 21	←		
					enter_menu	9F 80 22	→		
	00 02 00 42	2	1	2	application_info_enq	9F 80 20	→		[8]
					application_info	9F 80 21	←		
					enter_menu	9F 80 22	→		
	00 02 00 43	2	1	3	application_info_enq	9F 80 20	→		[1]
					application_info	9F 80 21	←		
					enter_menu	9F 80 22	→		
					request_CICAM_reset	9F 80 23	←		
					data_rate_info	9F 80 24	→		
	00 02 00 44	2	1	4	application_info_enq	9F 80 20	→		Настоящий стандарт
					application_info	9F 80 21	←		
					enter_menu	9F 80 22	→		
					request_CICAM_reset	9F 80 23	←		
					data_rate_info	9F 80 24	→		
					enter_cicam_channel	9F 80 25	→		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
Поддержка условного доступа	00 03 00 41	3	1	1	ca_info_enq	9F 80 30	→		[2]
					ca_info	9F 80 31	←		
					ca_pmt	9F 80 32	→		
					ca_pmt_reply	9F 80 33	←		
	00 03 00 81 (Примечание 1)	3	2	1	ca_info_enq	9F 80 30	→		Настоящий стандарт
					ca_info	9F 80 31	←		
					ca_pmt	9F 80 32	→		
					ca_pmt_reply	9F 80 33	←		
Управление Хостом	00 20 00 41	32	1	1	tune	9F 84 00	←		[2]
					replace	9F 84 01	←		
					clear_replace	9F 84 02	←		
					ask_release	9F 84 03	→		
	00 20 00 42	32	1	2	tune	9F 84 00	←		[1]
					replace	9F 84 01	←		
					clear_replace	9F 84 02	←		
					ask_release	9F 84 03	→		
					tune_broadcast_req	9F 84 04	←		
					tune_reply	9F 84 05	→		
					ask_release_reply	9F 84 06	←		
	00 20 00 43	32	1	3	tune	9F 84 00	←		Настоящий стандарт
					replace	9F 84 01	←		
					clear_replace	9F 84 02	←		
					ask_release	9F 84 03	→		
					tune_broadcast_req	9F 84 04	←		
					tune_reply	9F 84 05	→		
					ask_release_reply	9F 84 06	←		
					tune_lcn_req	9F 84 07	←		
					tune_ip_req	9F 84 08	←		
					tune_triplet_req	9F 84 09	←		
					tune_status_req	9F 84 0A	←		
					tune_status_reply	9F 84 0B	→		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
	00 20 00 81 (Примечание 1)	32	2	1	ask_release	9F 84 03	→		Настоящий стандарт
					tune_broadcast_req	9F 84 04	←		
					tune_reply	9F 84 05	→		
					ask_release_reply	9F 84 06	←		
					tune_lcn_req	9F 84 07	←		
					tune_ip_req	9F 84 08	←		
					tune_triplet_req	9F 84 09	←		
					tune_status_req	9F84 0A	←		
					tune_status_reply	9F84 0B	→		
Дата-время	00 24 00 41	36	1	1	date_time_enq	9F 84 40	←		[2]
					date_time	9F 84 41	→		
MMI	00 40 00 41	64	1	1	close_mmi	9F 88 00	→		[2]
					display_control	9F 88 01	←		
					display_reply	9F 88 02	→		
					text_last	9F 88 03	←		
					text_more	9F 88 04	←		
					keypad_control	9F 88 05	←		
					keypress	9F 88 06	→		
					enq	9F 88 07	←		
					answ	9F 88 08	→		
					menu_last	9F 88 09	←		
					menu_more	9F 88 0A	←		
					menu_answ	9F 88 0B	→		
					list_last	9F 88 0C	←		
					list_more	9F 88 0D	←		
					subtitle_segment_last	9F 88 0E	←		
					subtitle_segment_more	9F 88 0F	→		
					display_message	9F 88 10	←		
					scene_end_mark	9F 88 11	←		
					scene_done	9F 88 12	←		
					scene_control	9F 88 13	→		
					subtitle_download_last	9F 88 14	←		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
					subtitle_download_ more	9F 88 15	→		
					flush_download	9F 88 16	←		
					download_reply	9F 88 17	←		
	00 40 00 81 (Примечание 1)	64	2	1	close_mmi	9F 88 00	→		Настоящий стандарт
					display_control	9F 88 01	←		
					display_reply	9F 88 02	→		
					enq	9F 88 07	←		
					answ	9F 88 08	→		
					menu_last	9F 88 09	←		
					menu_more	9F 88 0A	←		
					menu_answ	9F 88 0B	→		
					list_last	9F 88 0C	←		
					list_more	9F 88 0D	←		
Соединение с низкой скоростью	00 60 xx x1	96	—	1	comms_cmd	9F 8C 00	←		[2]
					connection_ descriptor	9F 8C 01	←		
					comms_reply	9F 8C 02	→		
					comms_send_last	9F 8C 03	←		
					comms_send_more	9F 8C 04	←		
					comms_rcv_last	9F 8C 05	→		
					comms_rcv_more	9F 8C 06	→		
	00 60 xx x2	96	—	2	comms_cmd	9F 8C 00	←		[1]
					connection_ descriptor	9F 8C 01	←		
					comms_reply	9F 8C 02	→		
					comms_send_last	9F 8C 03	←		
					comms_send_more	9F 8C 04	←		
					comms_rcv_last	9F 8C 05	→		
					comms_rcv_more	9F 8C 06	→		
	00 60 xx x3	96	—	3	comms_cmd	9F 8C 00	←		[1]
					connection_descriptor	9F 8C 01	←		
					comms_reply	9F 8C 02	→		
					comms_send_last	9F 8C 03	←		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
					comms_send_more	9F 8C 04	←		
					comms_rcv_last	9F 8C 05	→		
					comms_rcv_more	9F 8C 06	→		
	00 60 xx x4	96	—	4	comms_cmd	9F 8C 00	←		Настоящий стандарт
					connection_descriptor	9F 8C 01	←		
					comms_reply	9F 8C 02	→		
					comms_send_last	9F 8C 03	←		
					comms_send_more	9F 8C 04	←		
					comms_rcv_last	9F 8C 05	→		
					comms_rcv_more	9F 8C 06	→		
					comms_info_req	9F 8C 07	←		
					comms_info_reply	9F 8C 08	→		
					comms_IP_config_req	9F 8C 09	←		
					comms_IP_config_reply	9F 8C 0A	→		
Управление контентом	00 8C 10 01	140	64	1	cc_open_req	9F 90 01	←		[1]
					cc_open_cnf	9F 90 02	→		
					cc_data_req	9F 90 03	←		
					cc_data_cnf	9F 90 04	→		
					cc_sync_req	9F 90 05	←		
					cc_sync_cnf	9F 90 06	→		
					cc_sac_data_req	9F 90 07	←		
					cc_sac_data_cnf	9F 90 08	→		
					cc_sac_sync_req	9F 90 09	←		
					cc_sac_sync_cnf	9F 90 10	→		
	00 8C 10 02	140	64	2	cc_open_req	9F 90 01	←		[1]
					cc_open_cnf	9F 90 02	→		
					cc_data_req	9F 90 03	←		
					cc_data_cnf	9F 90 04	→		
					cc_sync_req	9F 90 05	←		
					cc_sync_cnf	9F 90 06	→		
					cc_sac_data_req	9F 90 07	←		
					cc_sac_data_cnf	9F 90 08	→		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
					cc_sac_sync_req	9F 90 09	←		
					cc_sac_sync_cnf	9F 90 10	→		
					cc_PIN_capabilities_req	9F 90 11	→		
					cc_PIN_capabilities_ reply	9F 90 12	←		
					cc_PIN_cmd	9F 90 13	→		
					cc_PIN_reply	9F 90 14	←		
					cc_PIN_event	9F 90 15	←		
					cc_PIN_playback	9F 90 16	→		
					cc_PIN_MMI_req	9F 90 17	→		
	00 8C 10 03	140	64	3	cc_open_req	9F 90 01	←		Настоящий стандарт
					cc_open_cnf	9F 90 02	→		
					cc_data_req	9F 90 03	←		
					cc_data_cnf	9F 90 04	→		
					cc_sync_req	9F 90 05	←		
					cc_sync_cnf	9F 90 06	→		
					cc_sac_data_req	9F 90 07	←		
					cc_sac_data_cnf	9F 90 08	→		
					cc_sac_sync_req	9F 90 09	←		
					cc_sac_sync_cnf	9F 90 10	→		
					cc_PIN_capabilities_req	9F 90 11	→		
					cc_PIN_capabilities_ reply	9F 90 12	←		
					cc_PIN_cmd	9F 90 13	→		
					cc_PIN_reply	9F 90 14	←		
					cc_PIN_event	9F 90 15	←		
					cc_PIN_playback	9F 90 16	→		
					cc_PIN_MMI_req	9F 90 17	→		
	00 8C 10 41 (Примечание 1)	140	65	1	cc_open_req	9F 90 01	←		Настоящий стандарт
					cc_open_cnf	9F 90 02	→		
					cc_data_req	9F 90 03	←		
					cc_data_cnf	9F 90 04	→		
					cc_sync_req	9F 90 05	←		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
					cc_sync_cnf	9F 90 06	→		
					cc_sac_data_req	9F 90 07	←		
					cc_sac_data_cnf	9F 90 08	→		
					cc_sac_sync_req	9F 90 09	←		
					cc_sac_sync_cnf	9F 90 10	→		
					cc_PIN_capabilities_req	9F 90 11	→		
					cc_PIN_capabilities_ reply	9F 90 12	←		
					cc_PIN_cmd	9F 90 13	→		
					cc_PIN_reply	9F 90 14	←		
					cc_PIN_event	9F 90 15	←		
					cc_PIN_playback	9F 90 16	→		
					cc_PIN_MMI_req	9F 90 17	→		
Язык Хоста и страна	00 8D 10 01	141	64	1	Host_country_enq	9F 81 00	←		[1]
					Host_country	9F 81 01	→		
					Host_language_enq	9F 81 10	←		
					Host_language	9F 81 11	→		
CICAM _Upgrade	00 8E 10 01	142	64	1	cicam_firmware_ upgrade	9F 9D 01	←		[1]
					cicam_firmware_ upgrade_reply	9F 9D 02	→		
					cicam_firmware_ upgrade_progress	9F 9D 03	←		
					cicam_firmware_ upgrade_complete	9F 9D 04	←		
Профиль оператора	00 8F 10 01	143	64	1	operator_status_req	9F 9C 00	→		[1]
					operator_status	9F 9C 01	←		
					operator_nit_req	9F 9C 02	→		
					operator_nit	9F 9C 03	←		
					operator_info_req	9F 9C 04	→		
					operator_info	9F 9C 05	←		
					operator_search_start	9F 9C 06	→		
					operator_search_status	9F 9C 07	←		
					operator_exit	9F 9C 08	→		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
					operator_tune	9F 9C 09	←		
					operator_tune_status	9F 9C 0A	→		
					operator_entitlement_ack	9F 9C 0B	→		
					operator_search_cancel	9F 9C 0C	→		
	00 8F 10 02	143	64	2	operator_status_req	9F 9C 00	→		Настоящий стандарт
					operator_status	9F 9C 01	←		
					operator_nit_req	9F 9C 02	→		
					operator_nit	9F 9C 03	←		
					operator_info_req	9F 9C 04	→		
					operator_info	9F 9C 05	←		
					operator_search_start	9F 9C 06	→		
					operator_search_status	9F 9C 07	←		
					operator_exit	9F 9C 08	→		
					operator_tune	9F 9C 09	←		
					operator_tune_status	9F 9C 0A	→		
					operator_entitlement_ack	9F 9C 0B	→		
					operator_search_cancel	9F 9C 0C	→		
					opetaror_osdt_request	9F 9C 0D	→		
					operator_osdt_reply	9F 9C 0E	←		
					operator_nit_management	9F 9C 0F	→		
SAS	00 96 10 01	150	64	1	SAS_connect_rqst	9F 9A 00	→		[1]
					SAS_connect_cnf	9F 9A 01	←		
					SAS_data_rqst (Примечание 2)	9F 9A 02	←→		
					SAS_data_av (Примечание 2)	9F 9A 03	←→		
					SAS_data_cnf (Примечание 2)	9F 9A 04	←→		
					SAS_server_query (Примечание 2)	9F 9A 05	←→		
					SAS_server_reply (Примечание 2)	9F 9A 06	←→		
					SAS_async_msg	9F 9A 07	←→		

Продолжение таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
Приложение MMI	00 41 00 41	65	1	1	RequestStart	9F 80 00	←		[8]
					RequestStartAck	9F 80 01	→		
					FileRequest	9F 80 02	→		
					FileAcknowledge	9F 80 03	←		
					AppAbortRequest	9F 80 04	←→		
					AppAbortAck	9F 80 05	←→		
	00 41 00 42	65	1	2	RequestStart	9F 80 00	←		[1]
					RequestStartAck	9F 80 01	→		
					FileRequest	9F 80 02	→		
					FileAcknowledge	9F 80 03	←		
					AppAbortRequest	9F 80 04	←→		
					AppAbortAck	9F 80 05	←→		
	00 41 00 43	65	1	3	RequestStart	9F 80 00	←		Настоящий стандарт
					RequestStartAck	9F 80 01	→		
					FileRequest	9F 80 02	→		
					FileAcknowledge	9F 80 03	←		
					AppAbortRequest	9F 80 04	←→		
					AppAbortAck	9F 80 05	←→		
	0 41 00 81 (Примечание 1)	65	2	1	RequestStart	9F 80 00	←		Настоящий стандарт
					RequestStartAck	9F 80 01	→		
					FileRequest	9F 80 02	→		
					FileAcknowledge	9F 80 03	←		
					AppAbortRequest	9F 80 04	←→		
					AppAbortAck	9F 80 05	←→		
Многопоточность	00 90 00 41 (Примечание 1)	144	1	1	CICAM_multistream_ capability	9F 92 00	←		Настоящий стандарт
					PID_select_req	9F 92 01	←		
					PID_select_reply	9F 92 02	→		
Внешняя файловая система	00 91 00 41	145	1	1	FileSystemOffer	9F 94 00	←		Настоящий стандарт
					FileSystemAck	9F 94 01	→		
					FileRequest	9F 94 02	→		
					FileAcknowledge	9F 94 03	←		

Окончание таблицы Б.1

Ресурсы					Объекты приложений		Направление передачи		Нормативный документ
Имя ресурса	Идентификатор ресурса	Класс	Тип	Версия	Тег APDU	Значение тега	Хост	CICAM	
Дескремблирование семпла	00 92 00 41	146	1	1	sd_info_req	9F 98 00	→		Настоящий стандарт
					sd_info_reply	9F 98 01	←		
					sd_start	9F 98 02	→		
					sd_start_reply	9F 98 03	←		
					sd_update	9F 98 04	→		
					sd_update_reply	9F 98 05	←		
CICAM в режиме проигрывателя	00 93 00 41	147	1	1	CICAM_player_verify_ req	9F A0 00	→		Настоящий стандарт
					CICAM_player_verify_ reply	9F A0 01	←		
					CICAM_player_ capabilities_ req	9F A0 02	←		
					CICAM_player_ capabilities_ reply	9F A0 03	→		
					CICAM_player_start_ req	9F A0 04	→		
					CICAM_player_start_ reply	9F A0 05	→		
					CICAM_player_play_ req	9F A0 06	→		
					CICAM_player_status_ error	9F A0 07	←		
					CICAM_player_control_ req	9F A0 08	→		
					CICAM_ player_info_ req	9F A0 09	→		
					CICAM_player_info_ reply	9F A0 0A	←		
					CICAM_player_stop	9F A0 0B	→		
					CICAM_player_ session_end	9F A0 0C	←		
					CICAM_player_asset_ end	9F A0 0D	←		

Примечания

1 Этот ресурс применяется для работы с несколькими потоками, как указано в разделе 6 настоящего стандарта.

2 APDU cc_sac_data_req и cc_sac_data_cnf используют LTS_id как поле типа данных для передачи и подтверждения.

Приложение В
(обязательное)

Учетная спецификация: параметры обмена в APDU

Перечень параметров обмена в APDU и иентификаторов Datatype представлен в таблице В.1.

Таблица В.1 — Перечень параметров обмена в APDU

Ключ или переменная	Размер, бит	Комментарии	Идентификатор Datatype
Зарезерв.	—	—	1
Зарезерв.	—	—	2
Зарезерв.	—	—	3
Зарезерв.	—	—	4
HOST_ID	64	Создается с помощью ROT и включено в сертификат X.509	5
CICAM_ID	64	Создается с помощью ROT и включено в сертификат X.509	6
Host_BrandCert	(Примечание 1)	Сертификат бренда Хоста	7
CICAM_BrandCert	(Примечание 1)	Сертификат бренда CICAM	8
Зарезерв.	—	—	9
Зарезерв.	—	—	10
Зарезерв.	—	—	11
Kp	256	Прекурсор ключа CICAM Хоста для CCK	12
DHPH	2048	Открытый ключ DH Хоста	13
DHPM	2048	Открытый ключ DH CICAM/ CICAM	14
Host_DevCert	(Примечание 1)	Данные сертификата устройства Хост	15
CICAM_DevCert	(Примечание 1)	Данные сертификата устройства CICAM	16
Signature_A	2048	Сигнатура открытого ключа DH Хоста	17
Signature_B	2048	Сигнатура открытого ключа DH CICAM	18
auth_nonce	256	Случайный одноразовый код 256 битов, сгенерированный CICAM и переданный CICAM Хосту для использования в протоколе аутентификации	19
Ns_Host	64	Вызов Хоста к CICAM для SAC	20
Ns_CICAM	64	Вызов CICAM к Хосту для SAC	21
AKH	256	Ключ аутентификации Хоста	22
AKM	256	Ключ аутентификации CICAM	23
Зарезерв.	—	—	24
uri_message	24	Данные сообщения, переносящие информацию о правилах использования	25
program_number	16	Номер программы MPEG	26

Окончание таблицы В.1

Ключ или переменная	Размер, бит	Комментарии	Идентификатор Datatype
uri_confirm	256	Хэш на данные подтвержден Хостом	27
key register	8	(в формате uimsbf) 0 = четные, 1 = нечетные, другие значения не поддерживаются	28
uri_versions	256	Битовая маска, выражающая версии URI, которые могут поддерживаться Хостом. Формат — «uimsbf»	29
status_field	8	Поле состояния в APDU подтверждает сообщения	30
Srm_data_hdcp	перем	SRM для HDCP	31
Srm_confirm	256	Хэш на данные подтвержден Хостом	32
CICAM_license	перем	Лицензия от CICAM, связанная с контентом (Примечание 2)	33
license status	8	Текущее состояние лицензии контента	34
license_rcvd_status	8	Состояние замены лицензии контента	35
Host_license	перем	Лицензия, для которой Хост требует текущее состояние (Примечание 2)	36
play_count	8	Счетчик остатка проигрывания	37
operating_mode	8	Режим записи	38
PINcode data	перем	PIN-код CICAM один байт для каждой цифры PIN-кода	39
record_start_status	8	Состояние CICAM после протокола record_start	40
mode_change_status	8	Состояние CICAM после протокола изменения режима работы	41
record_stop_status	8	Состояние CICAM после протокола record_stop	42
srm_data_dtcp	перем	SRM для DTCP	43
Зарезерв.	—	—	44
Зарезерв.	—	—	45
Зарезерв.	—	—	46
Зарезерв.	—	—	47
Зарезерв.	—	—	48
Зарезерв.	—	—	49
LTS_id	8	Идентификатор локального TS	50

Примечания

1 Длины сертификатов являются переменными.

2 Лицензии не должны быть нулевой длины и должны быть дополнены до следующей границы байта. Размер лицензии не должен превышать 1024 байта.

Приложение Г
(обязательное)

Схемы XML

Г.1 Схемы OSDT

Г.1.1 Общие замечания

Онлайн SDT (OSDT) является структурой XML, которая содержит список доступных служб, предоставляемых IP.

В Г.1.2, Г.1.3 приложения Г настоящего стандарта определяются различные типы и элементы, которые используются в схемах XML OSDT.

Полная нормативная схема XML содержится в файле «osdt.xsd» в архиве ts_103205v010101p0.zip, который сопровождает настоящий стандарт.

В Г.1.4 приложения Г настоящего стандарта предоставлен пример использования OSDT.

Г.1.2 Пространство имен

Пространством имен схемы OSDT является urn: dvb: metadata: ciplus: osdt: 2013.

Г.1.3 Сложные типы и группы атрибутов

Г.1.3.1 Элемент типа SubRegion

Элемент типа SubRegion представлен на рисунке Г.1.3.1.

```
<complexType name="SubRegionType">
    <sequence>
        <element name="Region" type=" mpeg7:TextualType" maxOccurs="unbounded"/>
        <element name="SubRegion" type="dvb:SubRegionType" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="RegionListType">
    <sequence>
        <element name="PrimaryRegion" type="dvb:SubRegionType" minOccurs="0"/>
    </sequence>
    <attribute name="CountryCodes" type=" sds:ISO-3166-List"/>
</complexType>

<complexType name="TargetRegionType">
    <sequence>
        <element name="RegionList" type="dvb:RegionListType" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="AccessibleOutOfRegion" type="boolean" default="true"/>
</complexType>
```

Рисунок Г.1.3.1 — Тип SubRegion

Семантика элементов и атрибутов типа SubRegionType представлена в таблице Г.1.

Таблица Г.1 — Семантика элементов и атрибутов типа SubRegion

Элемент/Атрибут	Семантика
SubRegion	Тип содержит наименование субрегиона
Region	Наименование субрегиона. Допускается применение нескольких элементов этого типа, если они выполнены на различных языках
RegionList	Тип определяет регион и указывает имя региона
PrimaryRegion	Подробная информация о регионе определена в иерархическом порядке, начиная с основного региона

Окончание таблицы Г.1

Элемент/Атрибут	Семантика
CountryCodes	Список стран, которые составляют регион, который далее определяется элементом PrimaryRegion
TargetRegion Type	Тип используется для представления страны и региона в стране, где служба предназначена для приема. Там, где предполагается эквивалентность дескриптору региона, использование субрегионов должно быть ограничено двумя уровнями (т.е. основными регионами, содержащими субрегионы, которые содержат субрегионы)
RegionList	Список регионов в странах
AccessibleOutOfRegion	Флаг, указывающий необходимость доступа к службе, когда Хост не находится ни в одном из перечисленных регионов

Г.1.3.2 Элемент типа ServiceLocation

Элемент ServiceLocation представлен на рисунке Г.1.3.2.

```
        <complexType name="ServiceLocationType">
            <sequence>
                <any processContents="lax" id="ProtocolDependentInformation" />
                <element name="DRMControlInformation" type="casd:DRMControlInformationType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="ContentAttributes" type="dvb:ContentAttributesType" minOccurs="0"/>
                <choice>
                    <element name="IPMulticastAddress" type="sds:McastType"/>
                    <element name="RTSPURL" type="sds:RTSPURLType"/>
                    <element name="UriBasedLocation" type="dvb:ExtendedURIType"/>
                </choice>
            </sequence>
            <attribute name="priority" type="integer" default="0"/>
        </complexType>

        <complexType name="ExtendedURIType">
            <sequence>
                <element name="URI" type="anyURI"/>
            </sequence>
            <attribute name="contentType" type="mpeg7:mimeType" use="required"/>
        </complexType>

        <complexType name="ContentAttributesType">
            <sequence>
                <element name="AudioAttributes" type="tva:AudioAttributesType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="VideoAttributes" type="tva:VideoAttributesType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="CaptionLanguage" type="tva:CaptionLanguageType" minOccurs="0"
                maxOccurs="unbounded"/>
                <element name="SignLanguage" type="tva:SignLanguageType" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
        </complexType>

        <complexType name="LCNType">
            <attribute name="LCN" type="positiveInteger" use="required"/>
            <attribute name="subscribed" type="boolean" use="optional"/>
            <attribute name="selectable" type="boolean" use="optional" default="true"/>
            <attribute name="visible" type="boolean" use="optional" default="true"/>
        </complexType>
```

Рисунок Г.1.3.2, лист 1 — Элемент **ServiceLocation**

```
        <complexType name="IPServiceType">
            <sequence>
                <element name="UniqueIdentifier" type=" sds:TextualIdentifier"/>
                <element name="DVBTriplet" type="sds:DVBTriplet" minOccurs="0"/>
                <element name="ServiceLocation" type="dvb:ServiceLocationType" minOccurs="0"
maxOccurs="unbounded" />
                <element name="LCN" type="dvb:LCNType" minOccurs="0"/>
                <element name="TargetRegions" type="dvb:TargetRegionType" minOccurs="0"/>
                <element name="ServiceName" type="sds:MultilingualType" maxOccurs="unbounded"/>
                <element name="ApplicationLocation" type="anyURI" minOccurs="0"/>
                <element name="ServiceGenre" type="sds:Genre" minOccurs="0"/>
                <element name="ServiceType" type="sds:ServiceType" minOccurs="0"/>
                <element name="ContentAttributes" type="dvb:ContentAttributesType" minOccurs="0"/>
                <element name="BCG" type="sds:BCGOffering" minOccurs="0"/>
            </sequence>
        </complexType>

        <complexType name="IPServiceListType">
            <sequence>
                <element name="IPService" type="dvb:IPServiceType" maxOccurs="unbounded"/>
                <element name="BCG" type="sds:BCGOffering" minOccurs="0"/>
            </sequence>
            <attribute name="Version" type="positiveInteger" use="required"/>
</complexType>
```

Рисунок Г.1.3.2, лист 2

Семантика элементов и атрибутов элемента **ServiceLocation** представлена в таблице Г.2.

Таблица Г.2 — Семантика элементов и атрибутов элемента **ServiceLocation**

Элемент/Атрибут	Семантика
ServiceLocation	Используется для предоставления информации о местоположении службы вместе с информацией DRM и информацией аудио/видео
DRMControlInformation	Используется для обеспечения DRM информации, в том числе системы DRM ID и другие метаданные для этой версии службы. В DoNotRecord и DoNotTimeShift элементы могут быть расценены устройством в качестве подсказки: URI предоставленные CICAM всегда должны иметь приоритет
ContentAttributes	Атрибуты аудио, видео, субтитры и скрепление подписью этой версии службы
IPMulticastAddress	Сигнализирует использование IGMP, обеспечивает транспортный адрес и другие параметры, с которыми можно получить доступ к службе
RTSPURL	Сигнализирует использование RTSP, чтобы получить доступ к службе и обеспечивает URL, при котором можно получить доступ к описанию службы. Этот URL является совокупностью URL, когда имеет место управление URL для потоков FEC
UriBasedLocation	Обеспечивает URI местоположения службы, где целью URI является тип MIME как предусмотрено в атрибуте contentType
priority	Приоритет этого элемента ServiceLocationType относительно других элементов ServiceLocationType для службы
ExtendedURIType	Тип используется для предоставления URI с дополнительной информацией. Этот тип может быть использован для ProtocolDependentInformation
contentType	Тип MIME объекта, идентифицированного в URI
URI	URI, предоставляющее местоположение службы
ContentAttributesType	Тип используется для предоставления аудио, видео и других атрибутов службы
AudioAttributes	Атрибуты аудиослужбы
VideoAttributes	Атрибуты видеослужбы

Окончание таблицы Г.2

Элемент/Атрибут	Семантика
CaptionLanguage	Язык титров от службы
SignLanguage	Язык сигнализации службы
LCNType	Используется для предоставления номер логического канала для службы
LCN	Номер логического канала. Семантика этого атрибута такая же, как для поля logicial_channel_number в ciplus_service_descriptor в соответствии с [1] (N.1.2.3)
subscribed	Флаг, указывающий, подписан пользователь на эту службу или нет. В случае «ложь» устройство может предположить, что оно не сможет представить эту службу. Если этот атрибут не указан, статус подписки неизвестен
selectable	Флаг указывает, следует или не следует позволять устройству выбирать службу прямым цифровым вводом логического номера канала. Этот флаг интерпретируется, только если флаг установлен в «ложь». При установке «истина» закрытая служба выбирается прямым вводом логического номера канала; при установке на «ложь» скрытая служба непосредственно выбирается пользователем (но может выбираться LCN из среды приложений)
visible	Флаг, указывающий, должно ли устройство включать эту службу в список служб или EPG, представленные зрителю. Когда флаг установлен в «истинно», то он указывает, что служба обычно видима как служба Хоста или в списке каналов и EPG и т.д. Когда установлено в «ложь», то он указывает, что приемник, как правило, не предложит службу пользователю в нормальных режимах навигации, но приемник должен обеспечить механизм получения доступа к этим службам прямой записью логического номера канала в зависимости от установки выбираемого флага
IPServiceType	Используется для предоставления подробной информации о службе
UniqueIdentifier	Уникальный идентификатор службы. Этот идентификатор никогда не должен изменяться для службы, даже если изменились все остальные параметры службы. Если дочерний атрибут DomainName отсутствует, то должно приниматься значение домена, в котором этот файл был расположен
DVBTriplet	Триплет DVB, который может использоваться для обращения к этой службе, даже если служба не поставляется в TS
ServiceLocation	Местоположение или местоположения, в которых может быть найден контент A/V службы. Если присутствует множество элементов этого типа, то элемент с самым высоким значением приоритета атрибута имеет самый высокий приоритет
LCN	Номер логического канала службы
TargetRegions	Регионы, в которых предусмотрен прием службы
ServiceName	Имя службы. Может существовать несколько элементов этого типа, если у них всех есть различные атрибуты языка
ApplicationLocation	Местоположение файла AIT XML, в котором может быть найдено приложение, связанное со службой, в соответствии с [23]
ServiceGenre	Жанр службы
ServiceType	Тип службы, определенный в [3]
ContentAttributes	Атрибуты контента службы
BCG	Детализированное руководство по широкополосному контенту, переносящему метаданные нескольких служб, включенных в это руководство
IPServiceListType	Перечень типов всех доступных служб и BCG, относящихся к этим службам
IPService	Детализированные службы
BCG	Детализированное руководство по широкополосному контенту, переносящему метаданные нескольких служб, включенных в это руководство
Version	Номер версии родительского элемента

Г.1.3.3 IPServiceList и ServiceLocation

Элементы IPServiceList и ServiceLocation представлены на рисунке Г.1.3.3.

```
<element name="IPServiceList" type="dvb:IPServiceListType"/>
<element name="ServiceLocation" type="dvb:ServiceLocationType"/>
```

Рисунок Г.1.3.3 — Элементы IPServiceList и ServiceLocation

Семантика элементов IPServiceList и ServiceLocation представлена в таблице Г.3.

Таблица Г.3 — Семантика элементов IPServiceList и ServiceLocation

Элемент/Атрибут	Семантика
IPServiceList	Перечень служб
ServiceLocation	Элемент высокого уровня, содержащий местоположение службы

Г.1.4 Пример файла OSDT

Этот файл OSDT, в качестве примера представленный на рисунке Г.1.4, содержит одну службу с логическим номером канала 1, который может быть предоставлен или как SD, или как HD, обе версии используют DASH MPEG. Служба BCG содержит ссылки на информацию EPG, которая может быть получена для службы.

```
<?xml version="1.0" encoding="UTF-8"?>
<dvb:IPServiceList Version="1" xmlns:dvb="urn:dvb:metadata:ciplus:osdt:2013"
xmlns:tva="urn:tva:metadata:2005"
xmlns:xsi="gost.e-ecolog.ru )01/XMLSchema-instance"
xsi:schemaLocation="urn:dvb:metadata:ciplus:osdt:2013 osdt.xsd">
  <dvb:IPService>
    <dvb:UniqueIdentifier ServiceName="channel-1" DomainName="www.broadcaster.com"/>
    <dvb:ServiceLocation>
      <dvb:ContentAttributes>
        <dvb:AudioAttributes>
          <tva:Coding href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:3.2">
            <tva:Name>MPEG-1 Audio Layer II</tva:Name>
          </tva:Coding>
          <tva:NumOfChannels>2</tva:NumOfChannels>
        </dvb:AudioAttributes>
        <dvb:VideoAttributes>
          <tva:Coding href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:2.2.2">
            <tva:Name>MPEG-2 Video Main Profile @ Main Level</tva:Name>
          </tva:Coding>
        </dvb:VideoAttributes>
      </dvb:ContentAttributes>
      <dvb:UriBasedLocation contentType="application/dash+xml">
        <dvb:URI>gost.e-ecolog.ru ter.com/mpd/channel-1-sd.mpd</dvb:URI>
      </dvb:UriBasedLocation>
    </dvb:ServiceLocation>
    <dvb:ServiceLocation>
      <dvb:ContentAttributes>
        <dvb:AudioAttributes>
          <tva:Coding href="urn:dvb:metadata:cs:AudioCodecCS:2007:3.1">
            <tva:Name>E-AC3</tva:Name>
          </tva:Coding>
          <tva:NumOfChannels>5</tva:NumOfChannels>
        </dvb:AudioAttributes>
        <dvb:VideoAttributes>
          <tva:Coding href="urn:dvb:metadata:cs:VideoCodecCS:2007:1.4.12">
```

Рисунок Г.1.4, лист 1 — Файл OSDT

```
                                <tva:Name>H264 High Profile @ Level 4.0</tva:Name>
                            </tva:Coding>
                        </dvb:VideoAttributes>
                    </dvb:ContentAttributes>
                    <dvb:UriBasedLocation contentType="application/dash+xml">
                        <dvb:URI>gost.e-ecolog.ru                                                    >
                    </dvb:UriBasedLocation>
                </dvb:ServiceLocation>
                <dvb:LCN LCN="1" selectable="true"/>
                <dvb:TargetRegions>
                    <dvb:RegionList CountryCodes="GBR">
                        <dvb:PrimaryRegion>
                            <dvb:Region>London</dvb:Region>
                        </dvb:PrimaryRegion>
                    </dvb:RegionList>
                </dvb:TargetRegions>
                <dvb:ServiceName Language="eng">Channel 1</dvb:ServiceName>
            </dvb:IPService>
            <dvb:BCG DomainName="gost.e-ecolog.ru                 ›
                <sds:BCG Id="BCG1" xmlns:sds="urn:dvb:metadata:iptv:sdns:2008-1">
                    <sds:Name Language="ENG">Operator BCG</sds:Name>
                    <sds:TransportMode>
                        <sds:HTTP Location="gost.e-ecolog.ru                                        >
                    </sds:TransportMode>
                </sds:BCG>
            </dvb:BCG>
</dvb:IPServiceList>
```

Рисунок Г.1.4, лист 2

Приложение Д
(обязательное)

Сигнализация приложения CICAM

Д.1 Общие замечания

Это приложение содержит спецификацию сигнализации CICAM, которая является устаревшей и будет в дальнейшем пересматриваться.

Д.2 Сигнализации приложения вещания CICAM

Если приложение вещания CICAM необходимо сигнализировать в таблице и секции кодирования AIT MPEG-2 в соответствии с [23] (5.3), то это должно быть сделано в соответствии с [23] со следующими изменениями:

- приложение должно быть передано с дескриптором транспортного протокола protocol_id 0x0004, байты селектора не должны использоваться;
- поле application_type должно указывать на технологию промежуточного программного обеспечения, используемую приложением, зарегистрированным в DVB. Домен приложения «CIEngineProfile1», определенный в [1] (12.2) и измененный в настоящем стандарте, должен использовать тип приложения 0x0013;
- дескриптор местоположения приложения должен ссылаться на исходный объект, предоставленный CICAM. Когда исходный объект будет обеспечен системой вспомогательных файлов CICAM, идентификатор домена должен быть получен в поле типа приложения.

Рекомендуется, чтобы путь к исходному объекту включал organisation_id и application_id. Это позволит избежать случайного запуска Хостом неверного приложения только потому, что у другого приложения или CICAM есть исходный объект с тем же именем в том же местоположении. Например, приложение, organisation_id которого имел значение 0x21, application_id которого имел значение 0x42 и чей исходный объект вызвался «index.html», могло бы включать следующий путь в дескриптор расположения приложения — «/0x21/0x42/index.html».

Д.3 Приоритет приложений вещания и приложений вещания CICAM

Если оба приложения: приложение вещания и приложение вещания CICAM сигнализируются в AIT службы, то для выбора запускаемого приложения должно использоваться поле application_priority AIT.

Библиография

[1]	CI Plus™ specification V1.3.1 (2011-09)	Content Security Extensions to the Common Interface Расширения контента безопасности общего интерфейса
[2]	CENELEC EN 50221 (1997-02)	Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications Спецификация общего интерфейса для условного доступа и других приложений декодеров цифрового телевизионного вещания
[3]	ETSI EN 300 468 V1.13.1 (2012-08)	Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems Цифровое телевизионное вещание (DVB); Спецификация информации о службах в системах DVB
[4]	ETSI TS 102 034 V1.4.1 (2009-08)	Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks Цифровое телевизионное вещание (DVB); Передача транспортного потока MPEG-2 основных служб DVB через базовые сети IP
[5]	ISO/IEC 14496-12 (2012)	Information technology — Coding of audio-visual objects — Part 12: ISO base media file format Информационные технологии. Кодирование аудиовидеообъектов. Часть 12. Базовый формат медиафайлов ISO
[6]	ISO/IEC 14496-14 (2003)	Information technology — Coding of audio-visual objects — Part 14: MP4 file format Информационные технологии. Кодирование аудиовидеообъектов. Часть 14. Формат файла MP4
[7]	ETSI ES 202 184 V2.3.1 (2012-12)	MHEG-5 Broadcast Profile Профиль вещания MHEG-5
[8]	ETSI TS 101 699 V1.1.1 (1999-11)	Digital Video Broadcasting (DVB); Extensions to the Common Interface Specification Расширения спецификации общего интерфейса
[9]	ITU-T Recommendation H.222.0 (2006)/ ISO/IEC 13818-1:2007	Information technology — Generic coding of moving pictures and associated audio information: Systems Информационные технологии. Универсальное кодирование движущихся изображений и связанной с ними звуковой информации. Системы
[10]	ETSI TS 101 162	Digital Video Broadcasting (DVB); Allocation of identifiers and codes for Digital Video Broadcasting (DVB) systems Цифровое телевизионное вещание (DVB); Распределение идентификаторов и кодов для систем цифрового телевизионного вещания (DVB)
[11]	Open IPTV Forum	Release 1 Specification, Volume 5 — Declarative Application Environment, V1.2, August 2012 Спецификация. Реализация 1, том 5. Декларативная среда приложения, v1.2, август 2012
[12]	Open IPTV Forum	Release 1 Specification, Volume 3 — Content Metadata, V1.2, August 2012 Спецификация. Реализация 1, том 3. Метаданные контента, v1.2, август 2012
[13]	ISO/IEC 23001-7 (2012)	Information technology — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files Информационные технологии. Технологии системы MPEG. Часть 7. Общее кодирование основных ISO форматов медиафайлов
[14]	ISO/IEC 23009-1	Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats Информационные технологии. Динамическая адаптивная потоковая передача по HTTP (DASH). Часть 1. Описание представления медиа и форматов сегментов
[15]	IETF RFC 4122	A Universally Unique IDentifier (UUID) URN Namespace Универсальный уникальный идентификатор (UUID) пространства имен URN
[16]	IETF RFC 768	User Datagram Protocol Протокол дейтаграмм пользователя

[17]	IETF RFC 793	Transmission Control Protocol Протокол управления передачей
[18]	IETF RFC 791	Internet Protocol Интернет-протокол
[19]	IETF RFC 3376	Internet Group Management Protocol, Version 3 Протокол управления группами Интернет, версия 3
[20]	IETF RFC 1112	Host extensions for IP multicasting Расширения Хоста для многоадресного вещания IP
[21]	IETF RFC 2460	Internet Protocol, Version 6 (IPv6) Specification Интернет-протокол, версия 6 (IPv6). Спецификация
[22]	IETF RFC 4443	Internet Control Message Protocol (ICMPv6) for the Internet Protocol, Version 6 (IPv6) Specification Протокол межсетевых управляющих сообщений (ICMPv6) для спецификации Интернет-протокола, версия 6 (IPv6)
[23]	ETSI TS 102 809 (V1.1.1)	Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid broadcast/broadband environments Цифровое телевизионное вещание (DVB); Сигнализация и передача интерактивных приложений и служб в гибридных вещательных и широкополосных средах
[24]	ETSI TS 102 727 (01-2010)	Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2 Цифровое телевизионное вещание (DVB); Мультимедийная домашняя платформа, спецификация 1.2.2

УДК 621.397:681.327.8:006.354 ОКС 33.170 ОКП 657400

Ключевые слова: телевидение вещательное цифровое, система ограничения доступа, виртуальный канал, платформа координации приложений, трек, семпл, общий интерфейс, модуль общего интерфейса

Редактор *А.Л. Безлюдникова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Ю.В. Поповой*

Сдано в набор 11.06.2016. Подписано в печать 12.07.2016. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 14,88. Уч.-изд. л. 13,54. Тираж 25 экз. Зак. 1928.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Набрано в ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
gost.e-ecolog.ru gost.e-ecolog.ru

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru