ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 54708— 2011

СИСТЕМА ЦИФРОВОГО ЗВУКОВОГО РАДИОВЕЩАНИЯ DRM

Протокол распределения и коммуникации (DCP)

Издание официальное

Москва
Стандартинформ
2020

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ФГУП «ВНИИНМАШ») и Федеральным государственным унитарным предприятием «Ордена Трудового Красного Знамени научно-исследовательский институт радио», Самарский филиал «Самарское отделение научно-исследовательского института радио» (филиал ФГУП «НИИР-СОНИИР»)

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

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 13 декабря 2011 г. № 869-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений документа Европейского института по стандартизации в области телекоммуникаций (ЕТСИ) «Всемирное цифровое радио (DRM). Протокол распределения и коммуникации (DCP)» (ETSI TS 102 821 v1.3.1 (2010—12) «Digital Radio Mondiale (DRM); Distribution and Communications Protocol (DCP)», NEQ)

5 ВВЕДЕН ВПЕРВЫЕ

6 ПЕРЕИЗДАНИЕ. Июнь 2020 г.

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, оформление, 2012, 2020

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, обозначения и сокращения . . . 1
4 Общее описание . . . 2
4.1 Обзор системы . . . 2
4.2 Архитектура системы . . . 2
4.2.1 TAG элементы, AF пакеты и PFT фрагменты . . . 4
5 TAG уровень . . . 4
5.1 TAG пакет . . . 4
5.1.1 Основные правила . . . 5
5.2 TAG элемент . . . 5
5.2.1 Иерархические TAG элементы — пример кодирования . . . 5
5.2.2 Специальные TAG элементы . . . 6
5.2.2.1 Тип протокола и версия, *ptr . . . 6
5.2.2.1.1 Нумерация версий . . . 7
5.2.2.2 Фиктивное заполнение, *dmy . . . 7
6 Уровень разделения на фреймы . . . 7
6.1 Структура AF пакета . . . 7
6.2 История версий . . . 8
7 Уровень PFT . . . 8
7.1 Структура фрагмента PFT . . . 9
7.2 Определения . . . 10
7.2.1 Известные значения . . . 10
7.2.2 Расчетные значения . . . 10
7.3 Кодирование . . . 11
7.3.1 Код Рида-Соломона . . . 11
7.3.2 Фрагментация . . . 12
7.3.3 Транспортная адресация . . . 13
7.4 Процесс декодирования . . . 13
7.4.1 Синхронизация . . . 13
7.4.2 Транспортная адресация . . . 14
7.4.3 Дефрагментация . . . 14
7.4.4 Декодирование Рида-Соломона . . . 14
Приложение А (обязательное) Расчет слова CRC . . . 15
Приложение Б (обязательное) Физическое отображение . . . 16
Б.1 Пакетные связи . . . 16
Б.1.1 UDP/IP . . . 16
Б.2 Потоковые связи . . . 16
Б.3 Файл . . . 16
Б.3.1 Файл IO (fio_) . . . 16

Б.3.1.1 AF пакет/PFT фрагмент (afpf) . . . 17
Б.3.1.2 Метка времени (time) . . . 17
Приложение В (справочное) Сигнализация основных уровней передачи и параметров . . . 18
В.1 DCP через UDP/IP . . . 19
В.2 DCP через транспарентные (последовательные) каналы связи . . . 19
В.3 DCP в/из файла с использованием файла IO . . . 19
В.4 DCP через TCP/IP . . . 20
Приложение Г (обязательное) DCP профили и DCP параметры . . . 21
Г.1 Определения профилей DCP . . . 21
Г.1.1 Профиль A DCP . . . 21
Г.1.2 Профиль B DCP . . . 21
Г.1.3 Профиль C DCP . . . 21
Г.2 Определения параметров DCP . . . 21
Г.2.1 AFRevision . . . 21
Г.2.2 AFMaxLen . . . 21
Г.2.3 PFTMaxLen . . . 22
Г.2.4 PFTMaxFragCnt . . . 22
Г.2.5 PFTMaxAFFragCache . . . 22
Приложение Д (справочное) Рекомендации по реализации DCP . . . 23
Д.1 Используемый интерфейс . . . 23
Д.2 Переупорядочение AF пакетов . . . 23
Д.3 Ошибочное поведение . . . 23
Приложение Е (справочное) DCP структура двунаправленной связи . . . 24
Е.1 Типовые требования связи . . . 24
Е.1.1 Транзакции . . . 24
Е.1.2 Классы транзакций . . . 24
Е.1.2.1 Транзакция передачи . . . 24
Е.1.2.2 Транзакция выборки . . . 25
Е.1.2.3 Транзакция подписки/отказа от подписки . . . 25
Е.2 DCP пакет сообщения . . . 25
Е.2.1 Общая структура . . . 25
Е.2.2 Типы транзакций . . . 26
Е.2.2.1 Транзакция передачи (запрос "*tsq", ответ "*tss") . . . 26
Е.2.2.2 Транзакция выборки (запрос "*tfq", ответ "*tfs") . . . 26
Е.2.2.3 Транзакция подписки/отказа от подписки (запрос "*tuq"/"*tnq", ответ "*tus"/"*tns"). Транзакция доставки (запрос "*tdq", ответ "*tds") . . . 26
Е.2.3 Поле флага типа ответа . . . 27
Е.2.4 Данные команды . . . 28
Е.2.4.1 Поле данных команды, несущее полный TAG пакет . . . 28
Е.2.4.2 Связанные данные команды в главном уровне иерархии пакета сообщения . . . 28
Библиография . . . 29

Введение

ETSI TS 102 821 v1.3.1 (2010—12) создан Объединенным техническим комитетом (JTC) «Радиовещание» Европейского радиовещательного союза (EBU), Европейского комитета по стандартизации в электротехнике (CENELEC) и Европейского института по стандартизации в области телекоммуникаций (ETSI).

Большое количество протоколов связи было разработано, чтобы обеспечить надежный обмен данными при широком использовании различной техники. Некоторые основаны на использовании двухсторонней связи, чтобы обеспечить запросы на повторение пропавших или искаженных сообщений, в то время как другие основаны на прямой коррекции ошибок с использованием кода Рида-Соломона, чтобы восстановить оригинальное сообщение. К сожалению, большинство протоколов были разработаны для конкретных применений и не могли достаточно успешно использоваться в многоадресных сетях или не подходили для использования в однонаправленных схемах, часто являющихся основой в распределительных системах. Когда рассматривался вопрос о разработке протокола распределения для DRM, ни один из доступных протоколов не сочли подходящим, вследствие чего было решено разработать надежный протокол связи нижнего уровня, подходящий для однонаправленных и двунаправленных каналов связи, который отвечал бы потребностям DRM, но также был бы достаточно гибким, чтобы отвечать потребностям других применений.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

СИСТЕМА ЦИФРОВОГО ЗВУКОВОГО РАДИОВЕЩАНИЯ DRM

Протокол распределения и коммуникации (DCP)

Digital audio broadcasting system DRM.
Distribution and communication protocol (DCP)

Дата введения — 2012—09—01

1 Область применения

Настоящий стандарт относится к системе DRM, осуществляющей цифровое звуковое вещание в соответствии ETSI [1].

Стандарт устанавливает требования для распределения в системе DRM коммуникаций общего применения, подходящих для многоадресной передачи данных многим получателям (абонентам), использующим однонаправленные сети связи.

2 Нормативные ссылки

В настоящем стандарте нормативные ссылки не использовались.

3 Термины, определения, обозначения и сокращения

3.1 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1.1 **байт** (byte): Совокупность из 8 битов.

3.1.2 **протокол распределения и коммуникации** (Distribution and Communication Protocol; DCP): Протокол связи транспортного уровня, предусматривающий фрагментацию, адресацию и/или надежную передачу данных по каналам с ошибками с использованием кода Рида-Соломона для обеспечения прямой коррекции ошибок.

3.1.3 **разделение на фреймы** (Application Framing; AF): Уровень DCP, обеспечивающий логическую группировку множества TAG элементов.

3.1.4 **AF пакет** (AF Packet): Совокупность TAG элементов с заголовком, несущая связанный и независимый блок данных.

3.1.5 **TAG значение** (TAG Value): Полезная нагрузка TAG элемента.

3.1.6 **TAG название** (TAG Name): Название поля в индивидуальном TAG элементе, используемое для идентификации индивидуальной части информации.

3.1.7 **TAG пакет** (TAG Packet): Набор TAG элементов, переносящий связанный и модульный блок данных.

3.1.8 **TAG элемент** (TAG Item): DCP элементный тип, объединяющий в единых логических данных название, длину и значение данных.

3.2 Обозначения

В настоящем стандарте применены следующие обозначения:

N_x — значение «N», выраженное в основании «*x*». Основание «*x*» должно быть десятичным, таким образом $2A_{16}$ есть шестнадцатиричное представление десятичного числа 42,

Издание официальное

$\lceil x \rceil$ — наименьшее целое число, численно большее, чем x. Иногда известно как функция «потолка»;

$\lfloor x \rfloor$ — наибольшее целое число, численно меньшее, чем x. Иногда известно как функция «пола»;

$\frac{x}{y}$ — результат деления величины x на величину y;

MIN $\{a, \ldots, z\}$ — наименьшая величина в перечне.

3.3 Сокращения

В настоящем стандарте применены следующие сокращения:

AF (Application Framing (a DCP Protocol Layer) — разделение на фреймы (уровень DCP протокола);

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

CRC (Cyclic Redundancy Check) — циклический контроль с избыточностью (метод обнаружения ошибок с использованием полиномиального кода);

DCP (Distribution and Communication Protocol) — протокол распределения и коммуникации;

DRM (Digital Radio Mondiale) — Всемирное цифровое радио;

FEC (Forward Error Correction) — прямая коррекция ошибок;

IP (Internet Protocol) — Интернет-протокол;

LSb (Least Significant bit) — младший значащий бит;

LSB (Least Significant Byte) — младший значащий байт;

MSb (Most Significant bit) — старший значащий бит;

MSB (Most Significant Byte) — старший значащий байт;

MTU (Maximum Transmit Unit) — максимальный блок передачи;

PFT (Protection, Frangmentation and Transportation) — защита, фрагментация, транспортировка;

PPP (point-to-point protocol) — протокол двухточечного соединения;

RS (Reed Solomon) — Рид-Соломон;

TAG (Tag, Length, Value) — тег, длина, значение;

TCP (Transmission Control Protocol) — протокол управления передачей, один из основных сетевых протоколов Интернета, предназначенный для управления передачей данных в сетях TCP/IP;

UDP (User Datagram Protocol) — протокол передачи пользовательских дейтаграмм.

Примечание — В тексте стандарта, если не указано иное, принято следующее соглашение о порядке следования битов:

- на рисунках бит или байт, показанный слева, рассматривается как первый;
- в таблицах бит или байт, показанный слева, рассматривается как первый;
- в полях байта старший значащий бит (MSb) рассматривается первым и обозначается большим числом. Например, MSb одного байта обозначается "b_7" , а младший значащий бит (LSb) обозначается "b_0";
- в векторах (математических выражениях) бит с наименьшим индексом рассматривается как первый.

Порядок передачи (MSb — сначала или LSb — сначала) должен использоваться установленным порядком относительно физической линии связи. Там, где оба порядка действительны, сначала будет использоваться MSb.

4 Общее описание

4.1 Обзор системы

Протокол распределения и коммуникации специально разработан, чтобы обеспечить надежную многоадресную связь центрального сервера с множеством приемников. Ошибки на линии(ях) связи могут быть обнаружены и исправлены путем применения кода Рида-Соломона с прямым исправлением ошибок. В результате используемые линии связи могут быть однонаправленными, что обеспечивает потенциально существенную экономию затрат.

Примечание — Этот универсальный DCP протокол не определяет никаких правил или ограничений при выборе передачи. Выбор правил определяется конкретным приложением и поэтому рассматривается индивидуально для каждого определения прикладного протокола.

4.2 Архитектура системы

Приложения данных передаются с сервера на приемник через ряд уровней, как показано на рисунке 1.

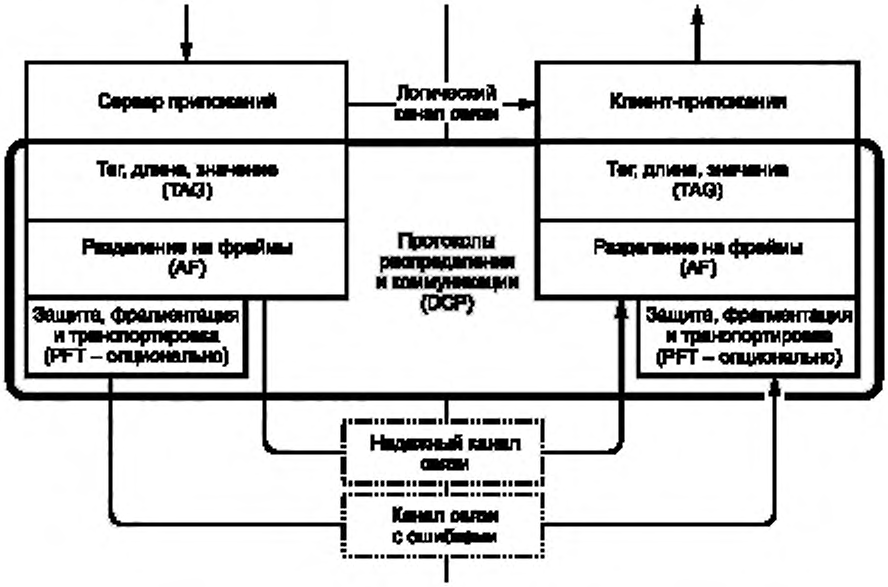


Рисунок 1 — Стек DCP протокола

Данные каждого уровня инкапсулированы в серии пакетов. Уровень TAG инкапсулирует элементарные элементы данных произвольной длины, в то время как уровень AF комбинирует элементарные данные в единый блок связанных данных. Дополнительный (опционально) уровень PFT позволяет осуществлять фрагментацию потенциально больших AF пакетов и добавляет возможность адресации и прямого исправления ошибок (FEC). Пакеты AF или фрагменты PFT могут тогда транспортироваться по любому из множества физических каналов, включая (не ограничивая) асинхронный последовательный, UDP/IP и даже сохраненный как файл на диске. Пример возможного варианта уровней связи показан на рисунке 2.

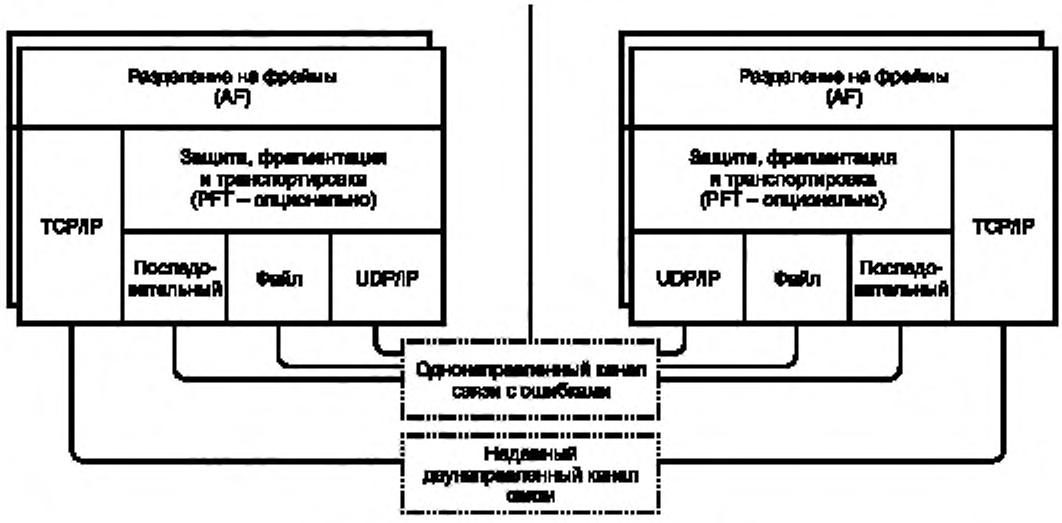


Рисунок 2 — Пример уровней связи DCP

Хотя в представленном примере используется канал передачи данных с ошибками, уровень PFT также полезен при использовании надежных каналов передачи данных, которые не обеспечивают функцию адресации транспортировки.

4.2.1 TAG элементы, AF пакеты и PFT фрагменты

Краткий обзор структуры данных на различных уровнях приведен на рисунке 3.

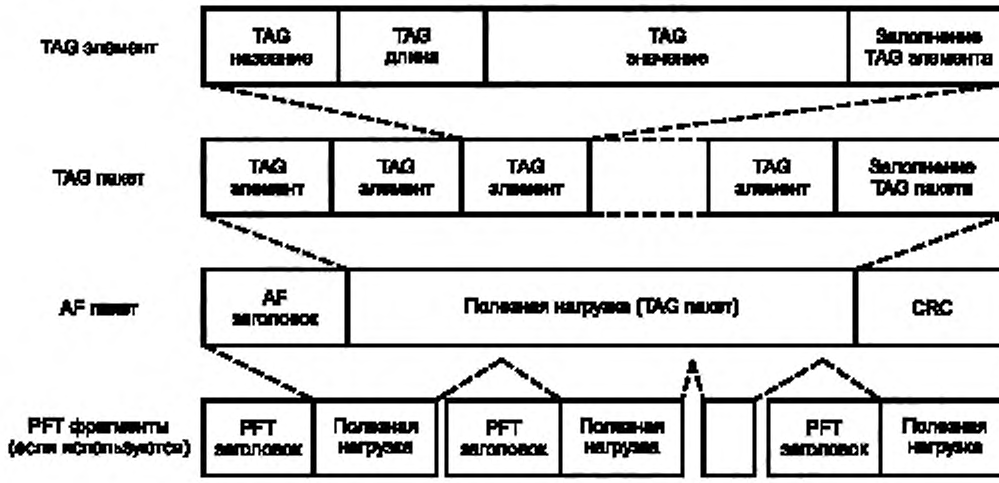


Рисунок 3 — TAG элементы, AF пакеты и PFT фрагменты

5 TAG уровень

TAG уровень формирует интерфейс между приложением и DCP. В пределах TAG уровня TAG элементы инкапсулированы как индивидуальные элементы данных. TAG элементы объединены в TAG пакет, чтобы сформировать логически связанный блок данных согласно приложению. Таким образом, чтобы определить новое приложение, необходимо просто определить ряд TAG элементов и наложить любые необходимые ограничения на особенности DCP, например, определяя, что PFT уровень должен всегда использоваться или что физической линией должен всегда быть Ethernet и т. д.

5.1 TAG пакет

TAG пакет — название, данное связной группе TAG элементов, которая имеет значение для общего приложения. TAG пакеты не содержат никакой синхронизации или корректировки ошибок и, как правило, существуют только в конкретном оборудовании.

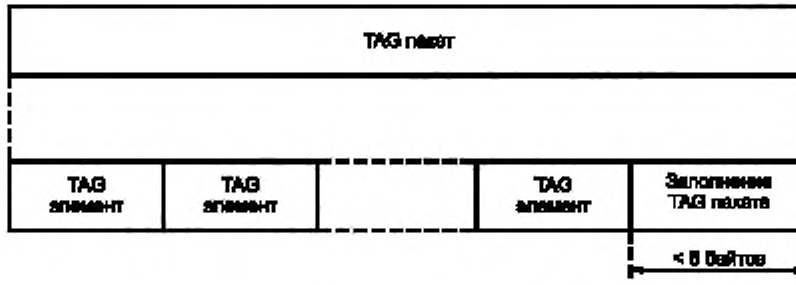


Рисунок 4 — TAG пакет

TAG пакет (рисунок 4) может включать до 7 байтов заполнения после последнего TAG элемента — данные, которые содержатся в заполнении, должны быть неопределенными. Такое заполнение должно

игнорироваться всеми приемниками. Так как самая короткая длина TAG элемента составляет 8 байтов, заполнение TAG пакета может быть легко идентифицировано. Если требуется больше чем 7 байтов заполнения, то должен быть использован специальный TAG элемент *dmy согласно 5.2.2.2.

Сам TAG пакет не имеет никакого заголовка, и нет никакого способа определить полную длину TAG элементов в пакете: эти функции выполняются, используя AF уровень, описанный ниже. Таким образом, TAG пакет — не подходящая структура для передачи данных от одной части оборудования к другой, но удобная абстракция, которая может при желании использоваться.

5.1.1 Основные правила

Приложение может определить, имеет ли порядок TAG элементов в пределах TAG пакета какое-нибудь значение. Очень настоятельно рекомендуется, чтобы порядок TAG элементов не был существенен.

Приложение может определить, может ли один TAG пакет содержать многие TAG элементы с тем же самым названием.

Реализации должны игнорировать любые TAG элементы, включенные в TAG пакет, которые не распознаны. Это позволит использовать частные расширения к существующим протоколам с обеспечением обратной совместимости.

Название TAG элемента может включить любые четыре байта и не должно быть ограничено символами ASCII. Одно общее ограничение дано в 5.2.2. Приложения могут определять дополнительные ограничения.

5.2 TAG элемент

Структура одного TAG элемента приведена на рисунке 5.

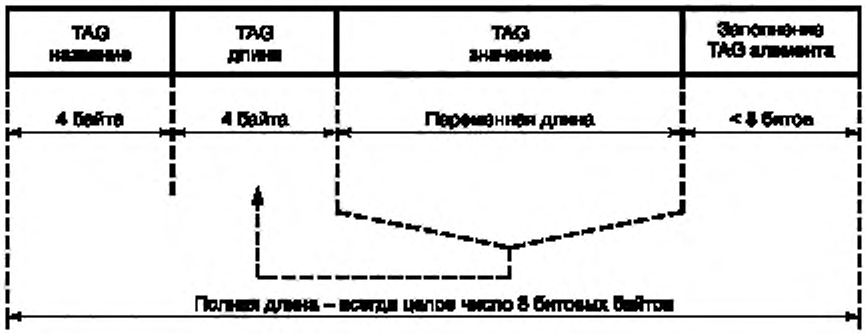


Рисунок 5 — Структура TAG элемента

TAG название: четырехбайтовое название, используемое для идентификации значения данных, переносимых в TAG элементе.

TAG длина: четырехбайтовая величина, представляющая число битов в поле TAG значение.

TAG значение: любое значение, требуемое приложением.

Заполнение TAG элемента: до семи битов неопределенного значения, как требуется, чтобы сделать полную длину TAG элемента целым числом байтов.

5.2.1 Иерархические TAG элементы — пример кодирования

Если требуется приложением, один TAG элемент может инкапсулировать в себе дополнительные TAG элементы, как показано на рисунке 6. Глубина иерархии может быть при необходимости ограничена приложением.

Так как каждый TAG элемент самого низкого уровня будет содержать заполнение TAG элемента, чтобы гарантировать, что его полная длина всегда будет составлять целое число 8-битовых байтов, TAG элементы более высокого уровня никогда не требуют наличия заполнения TAG элемента.

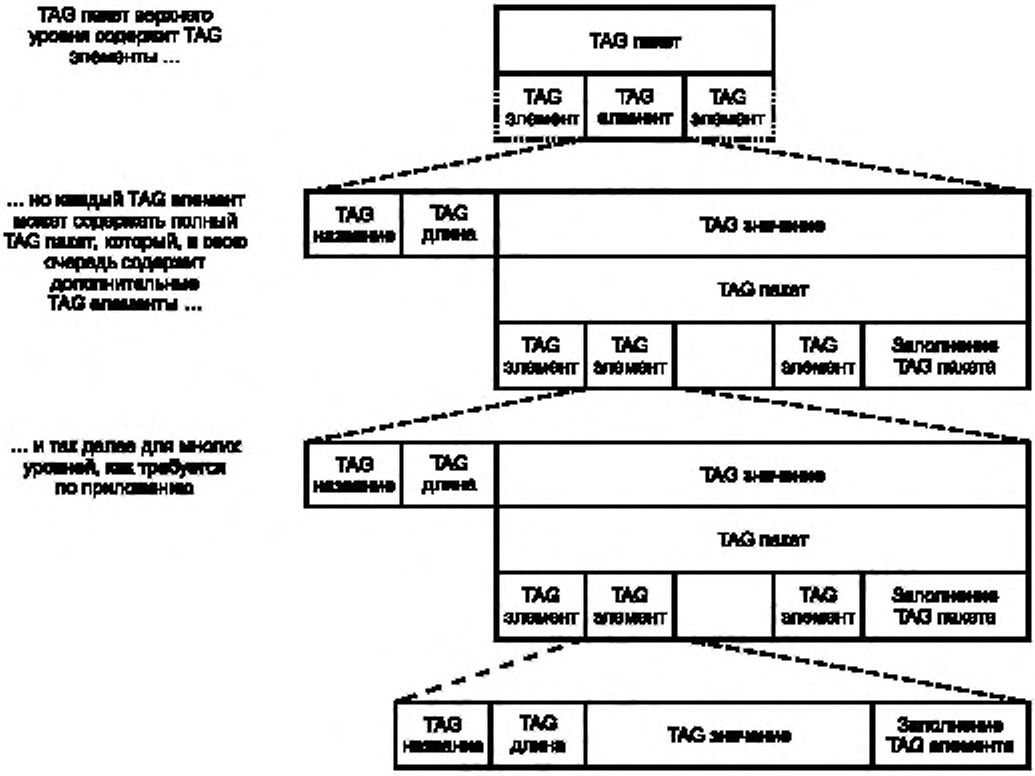


Рисунок 6 — Иерархические TAG элементы

5.2.2 Специальные TAG элементы

Каждое приложение может свободно определить соответствующие названия TAG элементов, единственное исключение — все названия, начинающиеся с символа ASCII "*", 42 (десятичное число) или 2A16, зарезервированы как управляющие TAG элементы.

5.2.2.1 Тип протокола и версия, *ptr

Каждое приложение, использующее DCP, должно объявить тип протокола и версию в каждом TAG пакете, используя TAG элемент *ptr, как показано на рисунке 7.

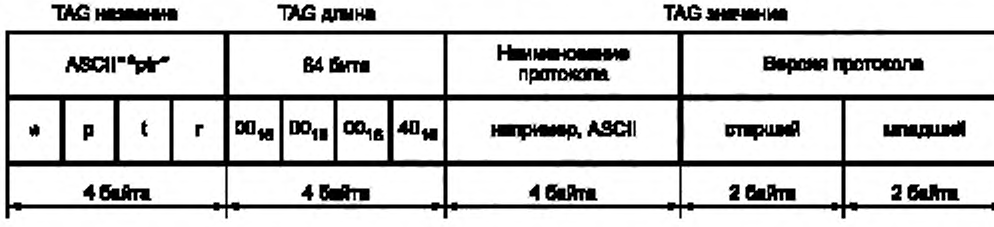


Рисунок 7 — Тип протокола и версия

Тип протокола: наименование протокола. Как правило, это будет кодироваться с использованием значений кода ASCII в диапазоне от 20_{00} до $7F_{16}$, но значения вне этого диапазона могут при желании использоваться.

Старшая версия: двоичный счетчик, представляющий старший номер версии протокола, начинающийся от 0000_{16}.

Младшая версия: двоичный счетчик, представляющий младший номер версии протокола, начинающийся от 0000_{16}.

TAG элемент *ptr не требует заполнения TAG элемента.

5.2.2.1.1 Нумерация версий

Каждому приложению разрешено использовать любую нумерацию версий в соответствии с требованиями, однако рекомендуется, чтобы версиии были совместимы с ближайшими предыдущими версиями. Таким образом, реализация, осуществляя версию 4.3 протокола WXYZ, должна быть в состоянии декодировать версии 4.0, 4.1 и 4.2 в дополнение к 4.3, но не обязательно должна поддерживать версию 3.1 или 5.0. Дополнительно пакеты версии 4.5 должны быть обработаны, как будто они были версией 4.3 с любыми новыми особенностями, добавленными в проигнорированной версии 4.4 или 4.5.

Как общий принцип, дополнение новых TAG элементов или определение ранее зарезервированных битов должны быть отражены изменением младшего номера версии. Обратно совместимые переопределения (включая удлинение) существующих TAG элементов должны быть отражены изменением старшего номера версии.

5.2.2.2 Фиктивное заполнение, *dmy

Чтобы обеспечить выравнивание слова, когда это необходимо, TAG элемент *dmy позволяет вставить в TAG пакет больше 8 байтов заполнения (неопределенные данные): если требуется заполнение меньше 8 байтов, то будет использоваться обычное заполнение TAG пакета (рисунок 8).

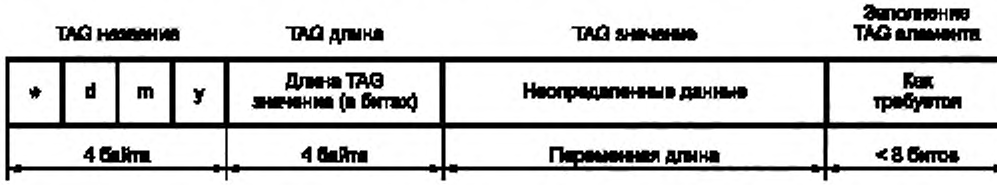


Рисунок 8 — Фиктивное заполнение TAG элемента

6 Уровень разделения на фреймы

AF уровень инкапсулирует одиночный TAG пакет в простую структуру, которая является подходящей для прохождения между оборудованием, связанным каналами связи без ошибок. Такие каналы связи можно обеспечить сетями, такими как TCP/IP или PFT уровня, описанные в разделе 7.

6.1 Структура AF пакета

Базовая структура AF пакета приведена на рисунке 9.

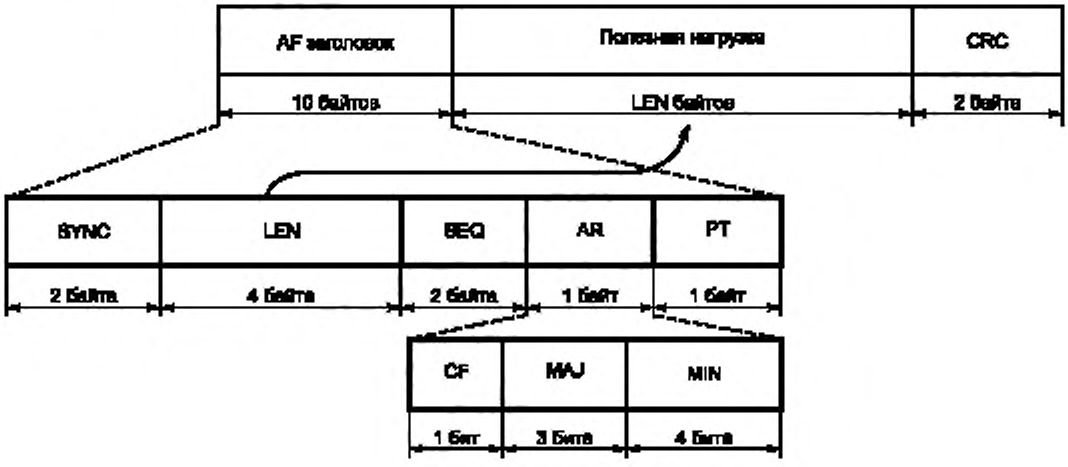


Рисунок 9 — AF уровень

SYNC (синхронизация): двухбайтовое представление "AF" в коде ASCII.

LEN: длина полезной нагрузки в байтах.

SEQ: порядковый номер. Каждый посланный AF пакет должен увеличить порядковый номер на единицу независимо от контента (содержания). Не должно быть никакого требования, чтобы первый полученный пакет имел определенную величину. Счетчик делает свертку от $FFFF_{16}$ до 0000_{16}, таким образом величина будет принимать значения $FFFE_{16}$, $FFFF_{16}$, 0000_{16}, 0001_{16} и т. д.

AR: пересмотр AF протокола — область, комбинирующая CF, MAJ и MIN поля.

CF: CRC флаг: имеет значение 0, если CRC поле не используется (CRC значение должно быть 0000_{16}) или 1, если CRC поле включает действительный CRC.

MAJ: используется старший пересмотр AF протокола (см. 6.2).

MIN: используется младший пересмотр AF протокола (см. 6.2).

PT (тип протокола): один байт, кодирующий протокол данных, переносимых в полезной нагрузке. Для TAG пакетов, значение в коде ASCII представляется как «T».

CRC: CRC рассчитывается, как описано в приложении А, по полю AF заголовка и полезной нагрузки, если поле CF есть 1, в противном случае (CF равно нулю) CRC значение равно 0000_{16}.

6.2 История версий

История версий AF протокола приведена в таблице 1.

Таблица 1 — История версий

Старшая версия	Младшая версия	Дата	Изменения
01_{16}	00_{16}	2003—01—28	Начальный публичный выпуск

7 Уровень PFT

Уровень PFT (дополнительная защита, фрагментация и транспортировка) осуществляет согласно названию три отдельные функции. Первая — защита от ошибок с использованием кода Рида-Соломона, который может обнаруживать и исправлять индивидуальные битовые ошибки, а также восстанавливать целые потерянные пакеты. Вторая — фрагментация, разделение больших пакетов на меньшие части, подходящие для каналов передачи данных и которые предписываются ниже MTU. Наконец, уровень PFT позволяет использовать такую ограниченную форму адресации транспортировки, чтобы те нижние уровни, которые не включают адресацию (например, RS-232 для последовательных связей), могли бы использоваться с многократными транспортными потоками. Совсем не обязательно, чтобы все три функции использовались одновременно, и использование дополнительных полей заголовка минимизирует затраты, когда определенных требований не предъявляется.

Разрешены следующие сочетания:

- инкапсулирование;
- простая фрагментация;
- использование кода Рида-Соломона с FEC и фрагментация.

Кроме того, каждое из вышеупомянутых трех сочетаний может быть объединено при желании с адресацией транспортировки. Эти варианты представлены на рисунке 10.

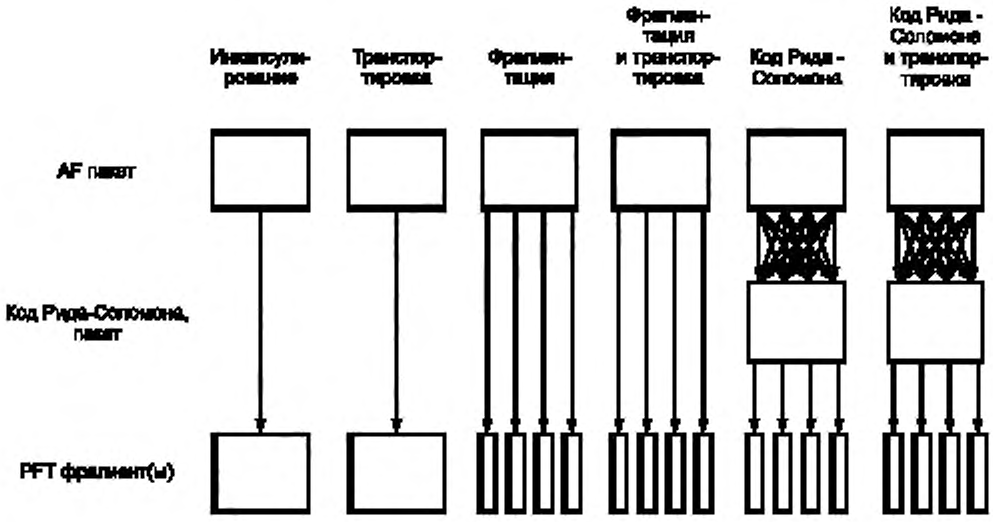


Рисунок 10 — Доступные опции с использованием уровня PFT

7.1 Структура фрагмента PFT

Структура фрагмента PFT представлена на рисунке 11.

Psync: ASCII строка "PF" используется как слово синхронизации для уровня PFT.

Pseq: 16-битовый счетчик, увеличивающийся на единицу с каждым полученным AF пакетом. Значение должно быть в пределах от 2^{16}—1 до 0, например…., 2^{16}—2, 2^{16}—1, 0, 1,…. Приемник не должен ожидать определенное значение в первом полученном фрагменте. Значение поля Pseq не имеет никакой связи со значением поля SEQ AF пакетов.

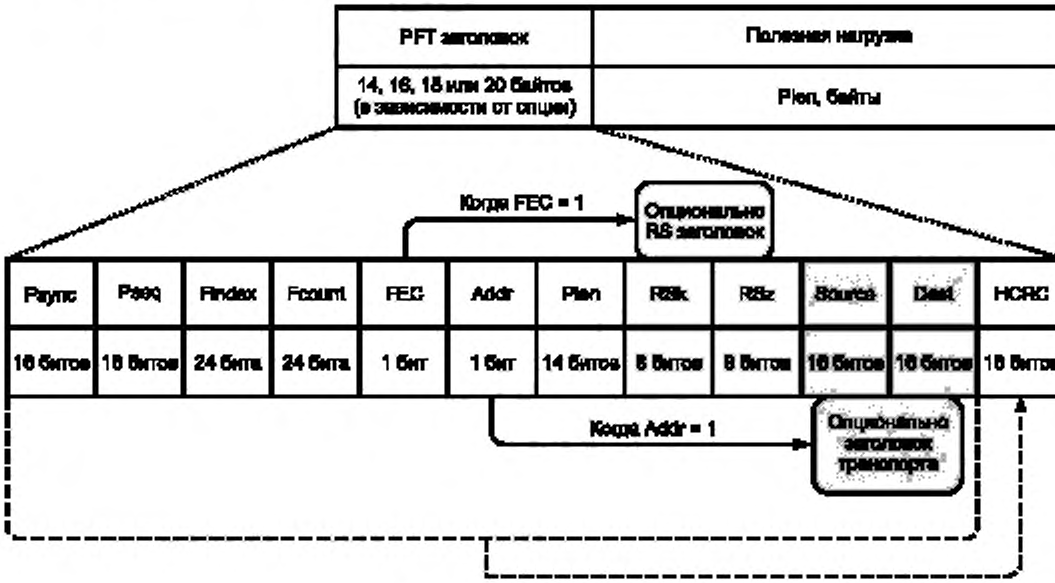


Рисунок 11 — Фрагмент PFT

Findex: 24-битовый счетчик, увеличивающийся на единицу с каждым фрагментом, который является частью единого AF пакета. Первый фрагмент из каждого AF пакета должен иметь значение нуль. Это значение не должно свертываться, налагая таким образом предел на максимальный размер AF пакета, который может быть перенесен. Максимальный размер может изменяться в зависимости от максимального блока данных (MTU) связи, но обычно составляет несколько гигабайтов.

Fcount: число фрагментов, полученных из данного AF пакета, в диапазоне от 1 до 2^{24}—1. Значение нуль не должно использоваться.

FEC: когда этот однобитовый флаг установлен на 1, то дополнительный RS заголовок присутствует.

Addr: когда этот однобитовый флаг установлен на 1, то дополнительный транспортный заголовок присутствует.

Plen: длина, в байтах, полезной нагрузки этого фрагмента.

RSk: длина слова данных кода Рида-Соломона — см. 7.2.2. Представлено только для случая, когда поле FEC равно 1.

RSz: число байтов заполнения в последнем блоке кода Рида-Соломона — см. 7.2.2. Представлено только для случая, когда поле FEC равно 1.

Source: свободный формат 16-битового идентификатора источника. Представлено для случая, когда поле Addr равно 1.

Dest: свободный формат 16-битового идентификатора предназначения. Представлено для случая, когда поле Addr равно 1.

HCRC: PFT заголовок CRC, рассчитанный через поля PFT заголовка от *Psync*, включая любой опциональный заголовок. CRC должен быть рассчитан, как описано в приложении А.

Когда FEC и Addr установлены на 1 (когда оба опциональных заголовка присутствуют), эти два заголовка должны появляться в порядке, приведенном на рисунке 11.

7.2 Определения

Далее по всему тексту стандарта должны применяться следующие определения.

7.2.1 Известные значения

l — общая длина оригинального AF пакета, включающая заголовок и CRC;

k_{max} — максимальное значение k, имеющее значение 207;

p — число байтов четности кода Рида-Соломона в фрагменте, имеющее значение 48;

m — максимальное число потерь фрагментов на пакет, который код Рида-Соломона должен быть в состоянии восстановить. Когда восстановление после потери фрагмента не требуется или когда код Рида-Соломона не используется, m должно быть нулем. Величина m больше 5 не рекомендуется из-за увеличения затрат на передачу многих маленьких фрагментов;

MTU — максимальный передаваемый размер блока (в байтах) для основного транспортного уровня. Когда транспортный уровень не имеет никакого *MTU* и когда *MTU* больше чем 2^{14}, тогда значение *MTU* должно быть 2^{14};

h — длина заголовка PFT в байтах. Значения должны быть 12, 14, 16 или 18 байтов в зависимости от вариантов использования.

7.2.2 Расчетные значения

c — число фрагментов Рида-Соломона (фиксируется как нуль, если код Рида-Соломона не используется);

k — длина данных каждого фрагмента, передается в поле RSk заголовка PFT (нуль, если код Рида-Соломона не используется);

z — число нулевых байтов, добавленных к последнему фрагменту Рида-Соломона, передается в поле RSz заголовка PFT (нуль, если код Рида-Соломона не используется);

s_{max} — промежуточный результат, представляющий максимальный размер полезной нагрузки в байтах для одного фрагмента;

f — число фрагментов, переносимых в поле Fcount PFT заголовка;

s — действительная величина фрагмента(ов), в байтах;

L — длина (в байтах) пакета, который будет фрагментирован. Когда код Рида-Соломона используется, L имеет величину $f \cdot s$, если нет, то l.

$$c = \left\lceil \frac{l}{k_{\max}} \right\rceil; \quad (1)$$

$$k = \left\lceil \frac{l}{c} \right\rceil; \quad (2)$$

$$z = c \cdot k - l; \quad (3)$$

$$s_{\max} = MIN\left\{ \left\lfloor \frac{c \cdot p}{m} \right\rfloor, MTU - h \right\} \quad \text{для } m > 0; \quad (4)$$

$$s_{\max} = MTU - h \quad \text{для } m = 0; \quad (5)$$

$$f = \left\lceil \frac{l + c \cdot p + z}{s_{\max}} \right\rceil; \quad (6)$$

$$s = \left\lceil \frac{l + c \cdot p + z}{f} \right\rceil. \quad (7)$$

Примечание — В предыдущей версии настоящего стандарта алгоритм расчета $s_{\max}$ для случая ($m > 0$) определялся следующим образом

$$s_{\max} = MIN\left\{ \left\lfloor \frac{c \cdot p}{m+1} \right\rfloor, MTU - h \right\}.$$

Следствием этой формулы является то, что даже при низких степенях защиты $m = 1$ (т. е. предназначенных для защиты от потери одного фрагмента) DCP кодер генерировал большее количество фрагментов, чем технически было необходимо, а DCP декодер исправлял два фрагмента вместо необходимого одного фрагмента. Кроме того, для каждого более высокого значения m еще один потерянный фрагмент, чем указанное значение m, могло быть восстановлено DCP декодером за счет неоправданно большого количества сгенерированных фрагментов.

Для копирования точного поведения DCP кодера по старой версии алгоритма значение m может быть выбрано на единицу выше, чем на самом деле необходимо. Например, чтобы копировать прежнюю "защиту против потери одного единственного пакета" ($m = 1$), m должно быть установлено в значение 2 вместо 1 с новой версией алгоритма, однако тогда фактическая защита будет от потерь двух фрагментов.

7.3 Кодирование

Следующие шаги должны быть выполнены, чтобы закодировать один AF пакет. Когда опция (например, код Рида-Соломона) недоступна, этот шаг просто не выполняется.

7.3.1 Код Рида-Соломона

Оригинальный пакет сначала делится на c фрагментов Рида-Соломона по k-байтов каждый; z нулевых байтов заполнения добавляются к последнему фрагменту при необходимости. Четные байты Рида-Соломона тогда рассчитываются и добавляются к каждому фрагменту, и полученный RS блок подвергается перемежению до формы RS пакета. Схематически это показано на рисунке 12.

Полный код Рида-Соломона должен быть RS (255, 207), вычисленный через поле Галуа (Galois Field GF) (2^8) с использованием полиномиального генератора

$$P(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

Когда расчетное значение для k меньше чем 207, байты k до 206 (включительно), закодированные RS (255, 207) кодом, должны все быть равны нулю и не должны быть включены в результирующий RS блок, таким образом производя RS ($k + p$, k) код.

Полиномиальный код должен быть

$$G(x) = \prod_{i=1}^{48} (x - \alpha^i). \quad (8)$$

Байты k данных и p проверочные байты кода Рида-Соломона отражаются как коэффициенты соответствующих полиномиалов в порядке уменьшения степени x. Если порядок байтов данных в RS блоке — от d_0 до d_{k-1}, сопровождаемому проверочными байтами от rs_0 до rs_{p-1}, то от d_0 до d_{k-1} — коэффициенты от x^{254} до x^{255-k} соответственно, и от rs_0 до rs_{p-1} — коэффициенты от x^{p-1} до x^0 в кодовом слове полинома, который имеет G(x) как фактор.

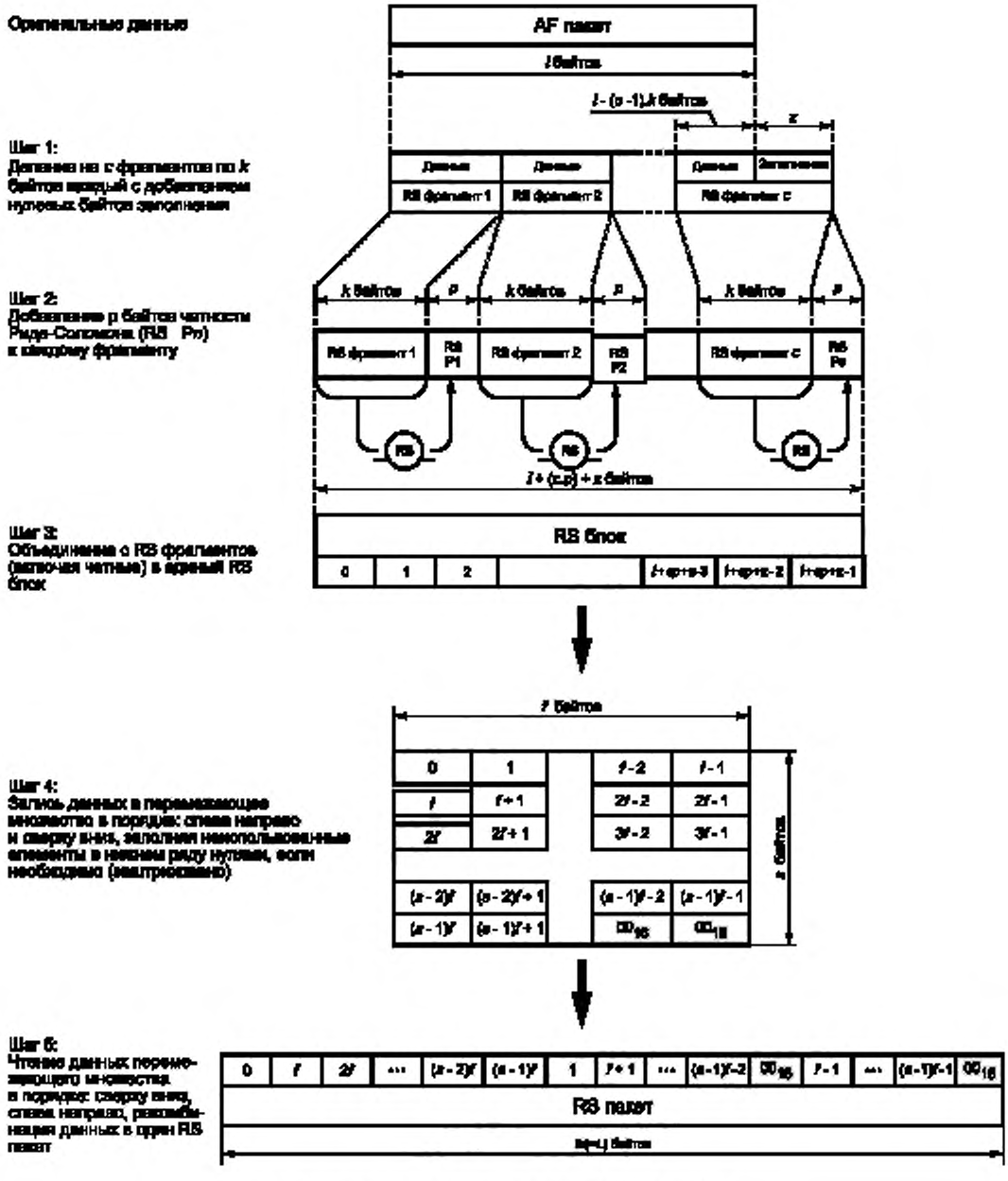
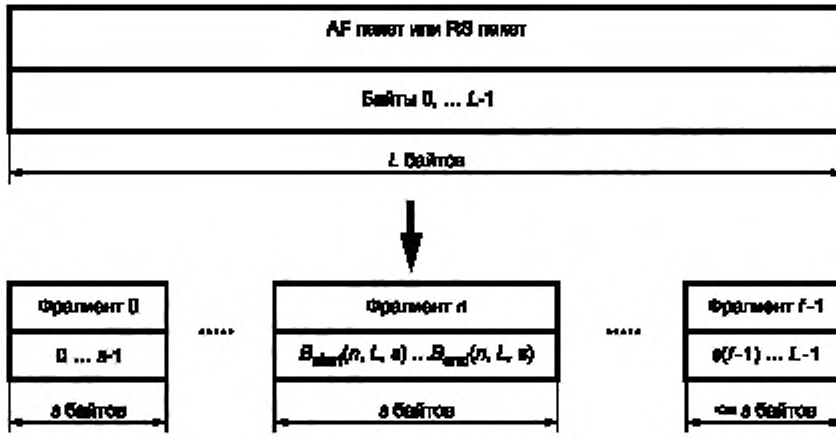


Рисунок 12 — Формирование пакета Рида-Соломона

7.3.2 Фрагментация

Фрагментация может быть применена непосредственно к AF пакету или к предварительно обработанному RS пакету. Фрагментация разделяет данные оригинального AF или RS пакета на множество отдельных фрагментов. Когда передается перемежеванный пакет Рида-Соломона, до *m* этих фрагментов может быть потеряно из каждого пакета без потери данных. Процесс фрагментации показан схематически на рисунке 13.



$$B_{start}\ (n,\ L,\ s) = n.s; \tag{9}$$

$$B_{end}\ (n,\ L,\ s) = MIN[s(n + 1) - 1,\ L - 1]. \tag{10}$$

Рисунок 13 — Фрагментация AF или RS пакета

Каждый фрагмент PFT, сформированный из одного AF или RS пакета, должен иметь те же самые значения во всех полях заголовка PFT, за исключением полей Findex, Plen и HCRC.

Поле Findex должно содержать счет, который должен начаться с нуля и увеличиваться на единицу для каждого фрагмента.

Поле Plen всех фрагментов должно иметь величину *s* для начальных *f* – 1 фрагментов и *s*-(*L*%f) (оператор модуля) для финального фрагмента. Когда используется код Рида-Соломона, все фрагменты будут иметь длину *s*.

Поле HCRC должно быть вычислено правильно для каждого фрагмента PFT.

7.3.3 Транспортная адресация

Поля адресации Source и Dest уровня PFT предназначены, чтобы использоваться при идентификации отправителя (Source) и получателя (Dest) пакета. Значение $FFFF_{16}$ должно использоваться, чтобы указать "передачу", все другие значения указывают определенный адрес. Если устройство конфигурировано с определенным источником и/или адресами назначения, то должны игнорироваться все фрагменты PFT, принятые с неправильным невещательным адресом.

7.4 Процесс декодирования

Процесс декодирования PFT фрагментов состоит из 4 стадий в следующем порядке:

1) синхронизация;
2) отказ от неправильно адресованных фрагментов (если транспортная адресация позволяет);
3) дефрагментация (если используется код Рида-Соломона либо простая фрагментация);
4) код Рида-Соломона обнаруживает и исправляет ошибки (если код Рида-Соломона разрешен).

7.4.1 Синхронизация

Для потоковых каналов связи (например, асинхронный последовательный или TCP/IP) должны использоваться следующие процессы для синхронизации поступающего потока; синхронизация может также применяться при чтении файла:

1) обнаружение битовой комбинации 0101000001000110_2 (5046_{16}), соответствующей синхронизирующему слову "PF" в коде ASCII, чтобы найти начало заголовка PFT кандидата;

2) вычисление CRC по заголовку кандидата — это может быть выполнено эффективно непрерывной байтовой реализацией CRC, используя тот факт, что CRC заголовка PFT, включающее поле CRC, будет иметь постоянное значение $1D0F_{16}$;

3) проверка длины заголовка, которая соответствует отобранным вариантам: если слишком короткая — продолжите от шага (2), если слишком длинная, возвратитесь к шагу (1), если соответствует, синхронизация была достигнута и поле Plen может использоваться, чтобы определить длину фрагмента.

Для пакетных каналов связи (например, UDP/IP) понятие синхронизации не нужно, поскольку транспортный протокол будет предъявлять полные фрагменты уровню PFT. Необходимо только проверить, что CRC и длина пакета правильны перед передачей пакета для дальнейшей обработки. Когда CRC и длина неправильны, пакет считается поврежденным и может не учитываться.

7.4.2 Транспортная адресация

Каждый фрагмент PFT может содержать источник и адрес назначения. Если представлено, поля адреса должны быть проверены, и если они не будут соответствовать конфигурации модуля, то весь фрагмент не учитывается.

Фрагменты PFT, которые не содержат дополнительный транспортный заголовок, никогда не учитываются.

7.4.3 Дефрагментация

Дефрагментация — процесс, обратный фрагментации. Управление памятью сохранено простым, так как известно, что все фрагменты (кроме последнего) — имеют одинаковый размер. Последний фрагмент будет того же самого размера или меньше, чем предшествующие фрагменты.

Если код Рида-Соломона не используется, каждый фрагмент должен быть принят правильно и полностью, для того чтобы восстановить оригинальный AF пакет.

При использовании кода Рида-Соломона применяется метод прямой коррекции ошибок, чтобы попытаться восстановить оригинальный AF пакет прежде, чем будут получены все фрагменты (см. 7.4.4).

Примечание 1 — Не все фрагменты PFT поступают в установленном порядке. Это может быть результатом установления специальных условий на транспортном уровне (например, изменение порядка пакетов, посылаемых через публичный Интернет) или может быть инициировано передающей стороной перемежение фрагментов PFT, принадлежащих различным AF пакетам. Перемежение может быть применено для защиты от продолжительных потерь связи.

Примечание 2 — PFT фрагменты, которые совпадают с ранее полученными PFT фрагментами и затем собранными в AF пакеты, должны быть отброшены.

7.4.4 Декодирование Рида-Соломона

Декодирование Рида-Соломона — процесс, обратный кодированию.

Длина RS пакета может быть рассчитана следующим образом:

$$c_{max} = \left\lfloor \frac{f \cdot s}{k + p} \right\rfloor; \tag{11}$$

$$Rx_{min} = \left\lfloor \frac{c_{max} \cdot p}{s} \right\rfloor, \tag{12}$$

где f — число фрагментов, полученных из этого пакета, несущего поле с заголовком Fcount;

s — размер в байтах фрагментов PFT, переносимых в поле заголовка Plen всех фрагментов, за исключением последнего фрагмента (где Fcount равняется [Findex – 1]);

k — размер данных в байтах кода Рида-Соломона, переносимых в поле заголовка RSk;

p — число байтов четности кода Рида-Соломона, должно иметь значение 48;

c_{max} — максимальное число фрагментов кода Рида-Соломона, которые могут быть посланы;

Rx_{min} — минимальное число фрагментов, которые должны быть получены, прежде чем может быть начато декодирование кода Рида-Соломона. Дополнительные фрагменты могут потребоваться, если произошли ошибки или если один из полученных фрагментов является последним.

Rx_{min} PFT фрагментов могут быть получены один раз, оставшиеся байты могут быть заполнены нулями и предпринята попытка декодирования кода Рида-Соломона с успешным получением неискаженного AF пакета с помощью правильного CRC.

В случае обнаружения битовых ошибок потребуется большее количество PFT фрагментов, прежде чем оригинальный AF пакет может быть правильно декодирован.

Байты заполнения, добавленные в течение процесса перемежения кода Рида-Соломона, могут в результате привести к большему количеству восстановленных данных, чем первоначально передавалось для очень больших пакетов. Эти дополнительные данные все будут нулями, и могут быть дифференцированы, начиная с z нулевых байтов, добавленных при шифровании кодом Рида-Соломона, используя значение поля заголовка RSz (z — число байтов заполнения кода Рида-Соломона, переносимых в поле заголовка RSz (z— число байтов заполнения кода Рида-Соломона, переносимых в поле заголовка RSz). Размер конечного AF пакета может быть определен из поля LEN AF.

Приложение А
(обязательное)

Расчет слова CRC

Использование циклических кодов с избыточностью (CRC кодов) позволяет обнаруживать ошибки передачи на стороне приемника. Эти слова CRC должны быть определены в результате процедуры, описанной в этом приложении.

CRC код определяется полиномом в степени n:

$$G(x) = x^n + g_{n-1}x^{n-1} + \ldots + g_2x^2 + g_1x + 1,$$

где: $n \geq 1$.

$g_i \in \{0,1\}$, $i = 1 \ldots n-1$.

Расчет CRC может быть выполнен посредством сдвигового регистра, содержащего n ступеней, эквивалентных степени полинома (см. рисунок А.1). Ступени обозначены от b_0 до b_{n-1}, где b_0 соответствует 1, b_1 соответствует x, b_2 соответствует x^2, b_{n-1} соответствует x^{n-1}.

Сдвиговый регистр функционирует, включая элементы XORs (исключающее ИЛИ) на входах тех регистров, где корреспондирующий коэффициент g_i полинома равен "1".

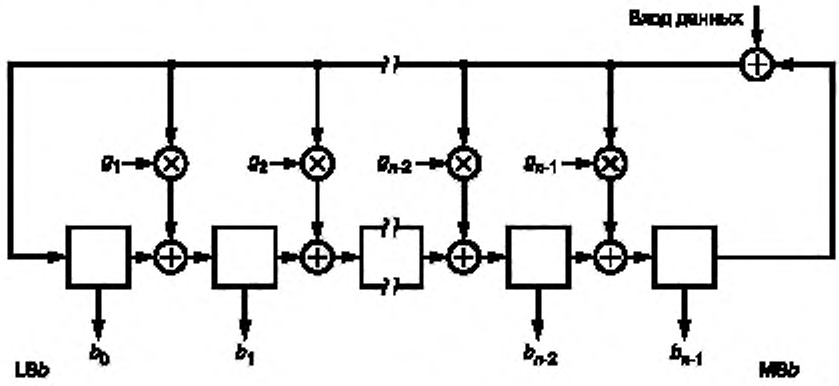


Рисунок А.1 — CRC генератор

В начале вычисления CRC все содержимое регистров калибруется одинаково.

После поступления на вход регистра первого бита блока данных (первый — MSb), сдвиговые тактовые импульсы заставляют регистр перемещать его содержимое вперед на одну ступень к MSb(b_{n-1}), в то же время загружая следующие ступени результатом соответствующих XOR операций. Процедура затем повторяется для каждого бита данных. Вслед за сдвигом после поступления на вход последнего бита (LSb) блока данных, сдвиговый регистр содержит слово CRC, которое затем считывается. Данные и CRC слова передаются, начиная с MSb.

CRC должен быть инвертирован (дополнен единицами) перед передачей.

Должен использоваться генератор полинома $G(x) = x^{16} + x^{12} + x^5 + 1$.

Если CRC будет приложен к оригинальным данным, то второй CRC, рассчитанный по всей длине, приведет к постоянному значению $1D0F_{16}$.

Приложение Б
(обязательное)

Физическое отображение

Фактическая передача PFT (или AF) пакетов должна быть возможна по многим видам существующих систем передачи с разной возможной инфраструктурой. Три возможных физических отображения определены в этом приложении, однако этот список не является ни исчерпывающим, ни предписывающим. Новые отображения могут быть определены в будущем.

Б.1 Пакетные связи

Сети пакетной коммутации становятся все более обычными, и во многих случаях либо PFT фрагменты, либо AF пакеты могут быть отображены точно в пакеты связи. MTU для уровня связи должен быть соблюден, и PFT фрагментация или PFT фрагментация Рида-Соломона должны использоваться, чтобы гарантировать, что единственный исходный пакет не будет фрагментирован на канальном уровне связи.

Б.1.1 UDP/IP

UDP/IP — одна из самых популярных сетей пакетной коммутации в настоящее время. PFT фрагменты и AF пакеты могут быть отображены точно в пакеты UDP /IP, если предусмотреть соблюдение MTU уровня UDP/IP. PFT фрагментация или PFT фрагментация Рида-Соломона должна использоваться, чтобы гарантировать, что единственный исходный пакет не будет фрагментирован. Любой определенный IP транспортный интерфейс для UDP/IP может использоваться, включая (но не ограничивая), 10Base-T Ethernet или PPP (двухточечной) связи.

UDP/IP предоставляет источник и номера портов назначения, следовательно, необходимость использования дополнительного PFT транспортного заголовка маловероятна, однако его использование не запрещено.

UDP/IP не гарантирует доставку пакетов, следовательно использование заголовка PFT Рида-Соломона настоятельно рекомендуется, но не является обязательным.

Б.2 Потоковые связи

Потоковые коммуникации в этом контексте описывают любой тип не пакетных коммуникаций. Примерами таких коммуникаций служат асинхронные последовательные каналы связи, синхронные последовательные каналы связи и каналы связи TCP/IP. Отличительная особенность таких каналов состоит в том, что наивысшие уровни (например, AF или PFT уровень) обязаны устанавливать пакетную синхронизацию прежде, чем пытаться декодировать поток.

Использование PFT уровня настоятельно рекомендуется, поскольку это было разработано для улучшения надежности синхронизации по сравнению с необработанным AF уровнем, однако этот необработанный AF уровень может использоваться непосредственно, если требуется.

Для физических каналов, которые не гарантируют безошибочный прием, рекомендуется использование дополнительного механизма кода Рида-Соломона с FEC, обеспеченного PFT уровнем. Каналы типа TCP/IP гарантируют обеспечение безошибочной связи, не нуждающейся в использовании защитного кода Рида-Соломона, однако линии связи, использующие RS-232, могут быть подвержены ошибкам и, следовательно, использование дополнения в виде кода Рида-Соломона с FEC целесообразно.

Б.3 Файл

PFT фрагменты или AF пакеты могут быть сохранены в виде файла для офлайнового распределения, архивации или любой другой цели. Стандартное отображение было определено путем использования опционально доступного иерархического TAG элемента, однако в будущем другие отображения могут быть определены, либо предложено использование его расширения. TAG элемент верхнего уровня имеет TAG название *fio_* и используется для инкапсуляции TAG пакета, одна часть которого является AF пакетом или PFT фрагментом в TAG элементе *afpf*. Были определены дополнительные TAG элементы, чтобы контролировать прием или управлять воспроизведением пакетов.

Б.3.1 Файл IO (fio_) (рисунок Б.1)

TAG элемент "fio_" — самый высокий уровень TAG элемента в иерархии файлов TAG элементов.

Этот TAG элемент действует как контейнер для TAG элемента afpf. TAG элемент time (метка времени) может быть представлен опционально.

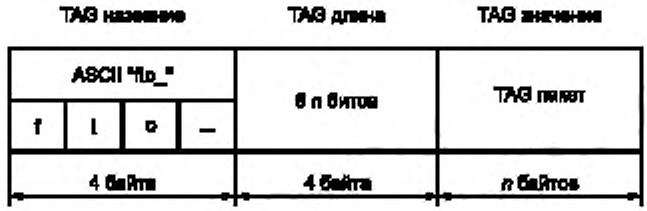


Рисунок Б.1 — Файл вход/выход

Б.3.1.1 AF пакет/PFT фрагмент (afpf) (рисунок Б.2)
TAG элемент afpf содержит весь AF пакет или PFT фрагмент как TAG значение.



Рисунок Б.2 — AF пакет или PFT фрагмент

Б.3.1.2 Метка времени (time) (рисунок Б.3)
TAG элемент time может быть отмечен в полезной нагрузке любого TAG элемента fio_. Это может быть записано как время приема полезной нагрузки или это может быть указано как намеченное время воспроизведения. Значение времени, данное в полях TI_SEC и TI_NSEC, может быть относительно начала файла или любой другой желательной ссылки.

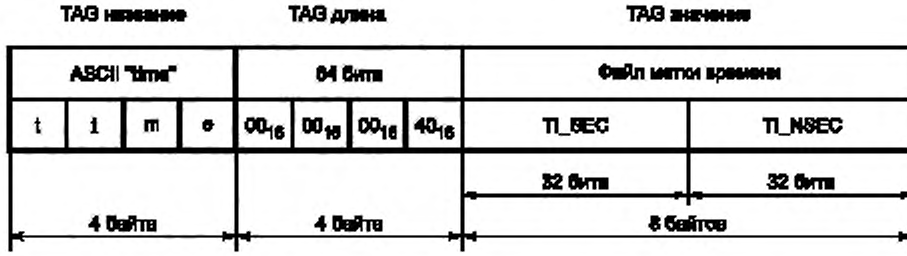


Рисунок Б.3 — Файл метки времени

TI_SEC: число целых SI секунд, в диапазоне от 0 до $(2^{32} - 1)$.
TI_NSEC: число целых SI наносекунд, в диапазоне от 0 до 999 999 999. Значения вне этого диапазона не определены и не должны использоваться.

Приложение В
(справочное)

Сигнализация основных уровней передачи и параметров

Поскольку несколько основных протоколов передачи для PFT (или AF) пакетов определены, целесообразно также определить общий метод для адресации и идентификации каждого из этих протоколов вместе с их индивидуальными параметрами. Эти информационные определения предназначены для организаций, использующих протокол DCP, чтобы быть в состоянии описать источники и цели (адреса протокола DCP) общепринятым способом.

Следующие определения соответствуют общему стилю универсального идентификатора ресурса (URI), как определено в IETF [2], но не полностью с ним совместимы и не должны синтаксически рассматриваться как противопоставление URI спецификациям. Семантически также потоки, указанные в настоящем приложении, не идентифицируют уникальные ресурсы; скорее они обеспечивают одну из возможных спецификаций для приложения, в которой будет приведена информация относительно необходимых элементов протокола DCP, осуществимых на данном канале связи.

Синтаксис, используемый для описания адресов протокола DCP, следующий:

- элементы в квадратных скобках идентифицируют дополнительные элементы;
- элементы в угловых скобках идентифицируют названные элементы;
- схему строк и названия/значения параметров следует рассматривать без различения прописных и строчных букв;
- большинство параметров являются дополнительными; если такой параметр не будет определен, то должно быть принято значение «по умолчанию»; если никакое (заданное по умолчанию) значение не определено, то параметр не является дополнительным;
- дополнительные параметры могут быть определены приложениями с использованием протокола распределения и коммуникации (например, максимальное время передачи или максимальное число зарегистрированных байтов);
- параметры могут появляться в любом порядке в секции параметра (см. ниже);
- неизвестные параметры и их значения должны игнорироваться и могут быть сообщены пользователю; о неизвестных значениях известных параметров необходимо сообщить пользователю.

Компоновка адреса сновного DCP протокола выглядит следующим образом:

- <scheme>:<target>[:[<src-addr>:]<dst-addr>]][?<param>=<val>[&<param>=<val>[&...]]].

Основные элементы:

- <scheme> (схема) определяет основной протокол, который будет использоваться, например, *"dcp.udp.pft"* или *"dcp.tcp"*. Все основанные на DCP схемы должны начинаться со строки *"dcp"*. Если *"pft"* — суффикс строки <scheme>, то протокол PFT используется вместе с указанным основным транспортным протоколом; если он отсутствует, AF пакеты передаются непосредственно, используя указанный основной транспортный протокол;
- <target> (цель) имеет различную точную семантику в зависимости от используемой схемы. Начиная со знака двоеточия (":") используется как разделитель элемента, это не должно использоваться как часть <target> строки;
- <src-addr> и <dst-addr> имеют различную точную семантику в зависимости от используемой схемы.

Примечание — Адреса DCP протокола, которые опускают src-addr компонент, синтаксически совместимы с URI спецификацией в соответствии с IETF [2].

Основные параметры AF уровня (доступны одинаково для всех основных транспортных уровней):

- "crc" (disabled (блокировано): "f", "false" (ошибка), "0"; enabled (разрешено): "t", "true" (верно), "1"; default (значение по умолчанию): "1") позволяет или запрещает вычисление CRC для AF уровня.

Основные параметры PFT уровня (доступны одинаково для всех основных транспортных уровней, только если сегмент <scheme> содержит суффикс ".pft"):

- "saddr", "daddr" (оба по умолчанию — 0). Если один или оба параметра "saddr" и "daddr" присутствуют, то параметры "saddr" и "daddr" переносятся в адресе заголовка PFT пакета. Если оба параметра опущены, рекомендуется, чтобы дополнительные адреса заголовка не были включены в заголовок PFT пакета. Для обратной совместимости, если ни "saddr" ни "daddr" не присутствуют, приложение может послать адреса заголовка, используя значения в <src-addr> и <dst-addr> сегментах, или 0, если <src-addr> и/или <dst-addr> сегменты отсутствуют;
- "fec" (disabled (блокировано): "0"; enabled (разрешено): "sp", "1" ..."9"; default (значение по умолчанию): "0") включает/отключает механизм FEC защиты PFT уровня; значение "sp" указывает на "однопакетный режим FEC" (FEC включено, но FEC не ведет к фрагментации; однако, фрагментация может еще следовать из "maxpaklen" параметра!); значения 1...9 определяют степень защиты (можно, например, указать число восстанавливаемых потерь фрагментов);
- "maxpaklen" (положительное целое число; без ограничения: "0"; default (значение по умолчанию): "0"), определяет MTU канала связи для уровня PFT в байтах.

DCP будет часто использоваться для передачи той же самой информации по многим реализациям транспортировки. Например, один и тот же поток данных можно послать одному адресату через последовательный порт, а другому — используя UDP/IP. Даже по одному типу транспортировки, например, UDP/IP, одна связь может осуществляться по местной локально-вычислительной сети и быть очень надежной, а другая может быть направлена через Интернет и потребует прямой коррекции ошибок. Если желательно поддержать различные возможные опции AF CRC и/или PFT для одной и той же логической DCP связи, тогда они должны быть определены как отдельные адреса DCP по всем объектам коммуникаций. В настоящее время не существует протокола, определенного в пределах DCP, чтобы договориться об определенном AF/PFT профиле во время установления связи.

B.1 DCP через UDP/IP

Этот формат адреса протокола DCP поддерживает как многоадресную, так и одноадресную доставку сообщений. Адрес протокола DCP относится к назначению DCP передачи, т. е. целевому хосту (узлу сети) при одноадресной передаче и целевой группе для многоадресной доставки или для местного приема DCP передачи.

Основные элементы:

- <scheme> (схема) имеет значение "dcp.udp[.pft]";
- <target> (цель) должна быть именем хоста или IP адресом (как определено в IETF [2] (3.2.2) предшествующим "//". Для совместимости приложения не обязаны поддерживать IP-буквенный синтаксис;
- <src-addr> — номер порта UDP/IP в исходном хосте в диапазоне от 0 до ($2^{16} - 1$). Если <src-addr> опущен или "0", тогда приложение может использовать любой номер порта;
- <dst-addr> — номер порта UDP/IP в хосте назначения в диапазоне от 0 до ($2^{16} - 1$).

Опустить <dst-addr> невозможно; 0 — действительный номер порта согласно UDP спецификации, но фактически исключен в реальных системах.

Если адрес протокола DCP описывает местные параметры приема DCP передачи, <target> идентифицирует адрес группы вещания в случае многоадресного приема или местного хоста (<localhostl>, "127.0.0.1" и т. д.) для одноадресного приема; <dst-addr> идентифицирует номер местного порта, который должен использоваться, чтобы получить входящие пакеты UDP.

Для этой схемы определены следующие дополнительные параметры:

- "interface" (ip адрес или системное специфическое наименование устройства, например "192.168.0.2", "eth0"; default (значение по умолчанию): используются системные таблицы маршрутизации). Этот параметр указывает, что платформа маршрутизации таблиц должна быть обойдена, и дейтаграммы будут переданы или получены через указанный интерфейс сети. Для групповых адресов определено, что это имеет отношение к выбору гнезда IP_MULTICAST_IF;
- "ttl" (числовой, от 0 (ограничено тем же самым хостом) до 255 (не ограничено); значения по умолчанию — по базе значений по умолчанию). Определяет значение IP_MULTICAST_TTL (время существования) выбора гнезда. Имеет значение только для многоадресной рассылки и существенное, если административная область действия не определена для указанной группы многоадресной рассылки.

Пример 1: dcp.udp.pft://192.168.0.1:3002?fec=9&crc=0&saddr=7&daddr=6.

Пример 2: dcp.udp://224.10.1.20:3002?interface=192.168.0.2.

Пример 3: dcp.udp://transmitter2.drm.org:1234:3114.

Пример 4: dcp.udp.pft://192.168.0.1:3002?fec=sp&crc=0.

B.2 DCP через транспарентные (последовательные) каналы связи

Основные элементы:

- <scheme> (схема) имеет значение "dcp.ser[.pft]";
- <target> (цель) — системный определенный идентификатор устройства, например, "COM4" или "/dev/ttyS1";
- <src-addr> имеет значения поля заголовка пакета PFT SRC в диапазоне от 0 до ($2^{16} - 1$);
- <dst-addr> имеет значения поля заголовка пакета PFT DST в диапазоне от 0 до ($2^{16} - 1$).

Определены следующие дополнительные параметры для этой схемы:

- "bitrate"(числовой, например "115200");
- "flowctrl" ("xonxoff", "rtscts"/"hw" или "none" (по умолчанию).

Пример 1: dcp.ser.pft:/dev/ttyS3:1:2?bitrate=4800&fec=4&flowctrl=hw.

Пример 2: dcp.ser:COM2:200?bitrate=115200.

B.3 DCP в/из файла с использованием файла IO

Основные элементы:

- <scheme> (схема) имеет значение "dcp.file[.pft]";
- <target> (цель) — является системным конкретным названием файла, включая любой требуемый путь. Начиная со знака двоеточия (":"), используется как разделитель элемента, но не должен использоваться как часть строки <target>. Сценарием, где это не может быть опущено, является DOS совместимые строки тракта, включая название дисковода ("C:\temp\test.mdi"). В данном случае знак двоеточия должен всегда сопровождаться нечисловым знаком (например, знак наклонной черты влево или нечисловой знак начала названия файла). В случае неоднозначных предписаний, как "C:5:6" (где анализатор не может ясно решить, является ли "C" именем файла,

"5" источником и "6" адресом назначения, или опущен исходный адрес и файл имеет название "5" на диске "C"), анализатор должен предположить, что и источник и адрес назначения определены. Если предписание может стать неоднозначным, рекомендуется использовать параметры "saddr" и "daddr" PFT уровня вместо <src-addr> и <dst-addr> адресных элементов;

- <src-addr> имеет значение поля заголовка пакета PFT SRC, в диапазоне от 0 до ($2^{16} - 1$);
- <dst-addr> имеет значение поля заголовка пакета PFT DST, в диапазоне от 0 до ($2^{16} - 1$).

Пример 1: dcp.file:/temp/record_1/test.dcp.

Пример 2: dcp.file.pft:c:\temp\test.dcp:99:100.

Пример 3: dcp.file.pft:\\myhost\myshare\temp\test.dcp?saddr=99.

B.4 DCP через TCP/IP

При использовании DCP через TCP/IP имеется (запускается) активное окончание ("клиент") и пассивное окончание ("сервер", ожидающий запросы связи). Пассивное окончание может произвольно поддержать многократные сеансы DCP через TCP/IP последовательно или одновременно. Формат адреса протокола DCP определяет пассивное окончание любой связи локально (на "сервере") либо удаленно (удаленный адрес "клиента" может достичь соединения с "сервером").

Основные элементы:

- <scheme> (схема) имеет значение "dcp.tcp[.pft]";
- <target> (цель) должна быть именем хоста или IP адресом (как определено в IETF [2] (3.2.2), предшествующим знаку "//". Для совместимости приложения не обязаны поддерживать IP-буквенный синтаксис. Если <target> определит местные параметры приема DCP передачи (на "сервере"), это описывается на местном хосте ("localhost", "127.0.0.1" и т. д.);
- <src-addr> — номер порта TCP/IP в активном хосте ("клиент") в диапазоне от 0 до ($2^{16} - 1$). Если он определен и отличен от нуля в активном хосте ("клиент"), тогда этот номер порта должен использоваться как локальный исходящий номер порта для связи; если <src-addr> опущен или равен нулю, тогда можно использовать любой номер порта. Если он определен в пассивном хосте ("сервер"), тогда требование других исходных номеров порта будет отклонено; значение "нуль" эквивалентно исключению этого параметра;
- <dst-addr> — номер порта TCP/IP в пассивном хосте ("сервер") в диапазоне от 0 до ($2^{16} - 1$).

0 — действительный номер порта, но фактически не используется в реальных системах. Если номер определен и отличен от нуля в активном хосте ("клиент"), тогда соединения должны быть установлены с этим номером порта. Если порт определен в пассивном хосте ("сервер"), тогда хост должен принять поступающие на этот порт соединения.

Определены следующие дополнительные параметры для этой схемы:

- "interface" (ip адрес или системное специфическое название устройства, например, "192.168.0.2", "eth0"; default (значение по умолчанию): прием осуществляется на всех интерфейсах для входящих связей). Этот параметр важен только для пассивного хоста ("сервер"). Если он несет адрес одного из интерфейсов хоста из множества интерфейсов, тогда хост должен принимать соединения только на этом интерфейсе.

Пример 1 (на "сервере"): dcp.tcp://localhost:3002?interface=eth0.

Пример 2 (на "клиенте"): dcp.tcp://yourserver:3002.

Приложение Г
(обязательное)

DCP профили и DCP параметры

Не все области использования и, соответственно, не все реализации протокола распределения и коммуникации (DCP) могут потребоваться для поддержки всех функциональных возможностей, обеспечиваемых DCP. Поэтому это приложение перечисляет три доступных уровня поддержки особенностей DCP, названных профилями A, B и C DCP. Определение профилей DCP является наилучшим способом ясно показать, какие функциональные возможности обеспечиваются конкретной реализацией DCP.

Кроме того, дополнительные параметры DCP обеспечивают простой способ более детально представить некоторые частные аспекты применения DCP.

Профили DCP и приведенные ниже параметры DCP относятся к реализациям кодирования и декодирования DCP. Любая реализация по кодированию и декодированию DCP, не поддерживающая профиль A DCP (полная поддержка всех особенностей DCP), должна заявить этот факт.

Г.1 Определения профилей DCP

Г.1.1 Профиль A DCP

Реализация профиля A DCP поддерживает все особенности, обеспечиваемые DCP:

- AF уровень;
- PFT уровень с адресацией, фрагментацией и прямым исправлением ошибок.

Профиль A DCP — это заданный по умолчанию профиль DCP и неявно принятый, если не обозначена другая определенная реализация DCP.

Г.1.2 Профиль B DCP

Реализация профиля B DCP поддерживает следующие особенности, обеспечиваемые DCP:

- AF уровень;
- PFT уровень с адресацией и фрагментацией.

Данная реализация DCP не поддерживает следующую особенность DCP: PFT уровень с прямым исправлением ошибок.

Реализация декодирования DCP, поддерживающего профиль B, должна быть в состоянии успешно декодировать PFT защищенный AF пакет с использованием прямого исправления ошибок до успешного получения всех фрагментов PFT.

Г.1.3 Профиль C DCP

Реализация профиля C DCP поддерживает следующую особенность, обеспечиваемую DCP: AF уровень.

Реализация профиля C DCP не поддерживает следующую особенность DCP: PFT уровень (никакой адресации или фрагментации, никакого прямого исправления ошибок).

Г.2 Определения параметров DCP

Следующие параметры DCP могут использоваться, чтобы при необходимости описать некоторые аспекты использования конкретного DCP кодера или DCP декодера более подробно. Эти параметры DCP, если заявлено, обычно сопровождают спецификацию поддерживаемого DCP профиля.

Примечание — Если использование DCP более ограничено относительно определенного параметра, чем максимальное техническое ограничение, наложенное в соответствии с протоколом DCP (упомянутого ниже), об этом факте должно быть заявлено.

Г.2.1 AFRevision

Этот параметр указывает старшие и младшие номера версий протокола, поддерживаемые DCP кодером или DCP декодером, в форме "<major>.<minor>" (числовые значения).

Для декодеров DCP: если реализация DCP поддерживает диапазон версий, это можно указать, используя знак "-": "<major_first>.<minor_first>-<major_last>.<minor_last>" и/или заменяя номер версии <minor> знаком "x" (который указывает, что по крайней мере версия <minor> 0 конкретного номера версии <major> полностью поддерживается).

Более высокие <minor> версии, чем обозначенные для того же самого номера <major> версии, поддерживаются автоматически (обратная совместимость).

Пример: AF пересмотр 1.x—2.x.

Г.2.2 AFMaxLen

Этот параметр описывает максимальную длину байта (AF) полезной нагрузки, которая может быть обработана с использованием DCP (исключая AF заголовок).

Примечание — Максимальный размер AF полезной нагрузки, о которой технически можно сигнализировать в заголовке AF поля LEN, составляет $2^{32} - 1$.

Г.2.3 PFTMaxLen

Этот параметр описывает максимальную длину байта полезной нагрузки пакета PFT, которая может быть обработана с использованием DCP.

Примечание — Максимальный размер полезной нагрузки PFT, о которой можно технически сигнализировать в заголовке PFT поля PLEN, составляет $2^{14} - 1$.

Г.2.4 PFTMaxFragCnt

Этот параметр описывает максимальное число фрагментов PFT пакета AF, которое может быть обработано с использованием DCP.

Примечание — Максимальное число фрагментов PFT, о которых можно технически сигнализировать в заголовке PFT, составляет $2^{24} - 2$.

Г.2.5 PFTMaxAFFragCache

Этот параметр используется только при выполнении декодирования DCP.

Он описывает максимальное число AF пакетов, фрагменты PFT которых могут быть обработаны одновременно при их получении в произвольном порядке.

Приложение Д
(справочное)

Рекомендации по реализации DCP

Это приложение дает некоторые рекомендации относительно деталей реализации и ошибочных решений при реализации DCP декодера.

Д.1 Используемый интерфейс

Интерфейс между реализацией DCP и приложением с использованием этой реализации DCP не определен в пределах области применения настоящего документа.

Необходимо до реализации DCP обеспечить выбор достаточно обоснованного и документированного интерфейса приложения. Термин "достаточно" может существенно изменяться в зависимости от индивидуальных требований того или иного типа приложения и условий эксплуатации.

Д.1.1 Для выполнения DCP кодирования должны приниматься данные, переданные наряду с необходимыми параметрами и целевыми адресами, встроенными в требуемые AF, PFT и, вероятно, FileFraming пакеты; вывод результирующих пакетов должен осуществляться через указанный основной уровень передачи.

Д.1.2 Реализация DCP декодирования должна обрабатывать DCP пакеты, полученные через указанный основной уровень передачи, и отправлять полученные AF пакеты (или их содержание (контент) плюс дополнительная информация управления соответственно) согласно приложению. В большинстве случаев декодирование AF уровня (AF заголовок, вычисление CRC и т. д.), PFT уровня (переупорядочение, FEC декодирования/коррекции и т. д.), так же как уровня FileFraming (если используется), должно быть полностью выполнено внутренней реализацией DCP.

Д.2 Переупорядочение AF пакетов

Переупорядочение принятых AF пакетов, как правило, не может обрабатываться декодером DCP (в отличие от переупорядочения PFT фрагментов). Вместо этого реализация DCP может отправить любой успешно полученный контент AF пакетов (наряду с порядковым номером AF пакета и другой соответствующей информацией управления) непосредственно приложению.

Эти действия рекомендуются вследствие того, что некоторые приложения могут зависеть от обработки в реальном времени входных данных, и поэтому может считаться более необходимым заменить недостающие AF пакеты немедленно, чем вводить любую задержку на ожидание более позднего получения недостающих AF пакетов.

Даже если конкретное приложение может обрабатывать определенное количество переупорядоченных AF пакетов (например, путем буферизации входа), это будет зависеть от принятого приложением решения о размере входного буфера и связанных с ним параметров.

Разрешение обработки переупорядочения AF пакетов позволяет применять более общую реализацию DCP декодирования в комбинации с любым типом приложения — приложения в реальном времени с/без буферизации входных данных или приложения не в реальном времени.

Д.3 Ошибочное поведение

Эта DCP спецификация не включает никаких правил для обработки ошибочных ситуаций.

Д.3.1 При выполнении DCP кодирования об ошибочных ситуациях, как правило, сообщается в приложении (неправильный контент, неправильные целевые параметры, цель не достижима и т. д.).

Д.3.2 При выполнении DCP декодирования при конкретной реализации регистрируется и документируется его ошибочное поведение и информация об ошибке направляется в приложение. Обычно при выполнении DCP декодирования сообщается об ошибках, связанных с основным уровнем передачи, который был обозначен в приложении (например, "последовательный порт не может быть открыт", "неправильный номер UDP порта" и т. д.).

Однако при обработке полученных входных данных через успешно действующий основной уровень передачи реализация DCP декодирования может выбрать и передать только успешно декодированные AF пакеты (или их контент плюс дополнительную информацию управления соответственно) приложения и просто отказаться от любых неверно сформированных данных (которые могут фактически даже быть не связанными с DCP) без предварительного уведомления.

Приложение Е
(справочное)

DCP структура двунаправленной связи

Это приложение обрисовывает в общих чертах предложение о структуре двунаправленной связи общего назначения, основанной на протоколе распределения и коммуникации (DCP).

Примечание 1 — Основа DCP структуры для двунаправленной связи, представленная в этом приложении, — только информативная и доступна для использования любым приложением на основе DCP. Однако, любое приложение на основе DCP может выбрать собственную обработку двунаправленных связей через DCP, полагаясь только на использование основного уровня передачи (например, TCP/IP), либо расширяя определение этой структуры согласно своим индивидуальным требованиям.

Примечание 2 — Структура для двунаправленной связи, основанной на DCP, не дает никаких рекомендаций относительно выбора времени (ретрансляции оставшихся без ответа или неподтвержденных запросов, максимального перерыва между запросом и ответом и т. д.). Определение перерывов очень сильно зависит от специфики приложения и условий эксплуатации. Так как структура DCP для двунаправленной связи должна подходить для широкого спектра приложений, определение перерывов относится к техническим требованиям конкретного приложения, использующего эту DCP структуру для двунаправленной связи.

Е.1 Типовые требования связи

Этот раздел выделяет некоторые общие подходы для связи на базе DCP, на которых базируется следующая DCP структура для двунаправленной связи.

Е.1.1 Транзакции

Основу связи составляют транзакции (протоколы обмена). Транзакции включают в себя:

- одно сообщение запроса (специальный DCP пакет), посланное от "клиента" на "сервер";
- различное количество сообщений ответа (специальные DCP пакеты), посланные от "сервера" к "клиенту", со ссылкой на сообщение запроса.

Е.1.2 Классы транзакций

Send Transaction — транзакция передачи:

"Клиент" посылает информацию на "сервер".

Fetch Transaction — транзакция выборки:

"Клиент" запрашивает информацию от "сервера".

Subscribe/Unsubscribe Transaction; Delivery Transaction — транзакция подписки/отказа от подписки; транзакция доставки:

Запрос на "сервер" инициирует доставку информации.

"Клиент" подписывается на информацию, предлагаемую "сервером".

"Сервер" впредь будет доставлять информацию "клиенту" каждый раз, когда будут доступны (новые) данные.

"Сервер" останавливает доставку данных на явную команду отказа от подписки или, например, при пропадании переподписки (перерыв).

Е.1.2.1 Транзакция передачи

Типичные примеры типов транзакций передачи:

- команды:

  установить частоту, выключить через 5 минут, полить цветы и т. д.;

- уведомления:

  новый пользователь только загрузился, огонь на пятом этаже, саморазрушение активировано и не может быть остановлено.

Типичные примеры потоковых коммуникаций:

- "клиент" посылает команду "серверу" и не требует подтверждения; связь заканчивается немедленно, успешный прием команды не может быть проверен;
- "клиент" посылает команду "серверу" и требует подтверждения приема; "сервер" посылает уведомление подтверждения приема;
- "клиент" посылает команду "серверу" и требует подтверждения обработки запроса; приложение на "сервере" по команде запускает процесс и уведомляет "клиента".

Потенциальные типы ответа:

- нет ответа;
- подтверждение приема сообщения запроса;
- уведомление о статусах обработки "сервером" (принятая команда, обработанная команда, ошибка выполнения/ОК (правильно) и т. д.) — возможные состояния определены в соответствии с приложением.

Е.1.2.2 Транзакция выборки

Типичные примеры типов транзакций выборки:

- получите текущую частоту, получите список доступных услуг, получите индикатор качества приема и т. д.

Типичные примеры потока коммуникаций:

- "клиент" запрашивает данные от "сервера", который немедленно доступен;
- "клиент" запрашивает данные от "сервера", при этом от "сервера" требуются определенные действия (пред/пост обработка).

Потенциальные типы ответа:

- подтверждение приема сообщения запроса;
- уведомление о статусах обработки "сервером" (принятая команда, обработанная команда, ошибка/ОК и т. д.) — возможные состояния определены в соответствии с приложением;
- поставка запрашиваемой информации.

Е.1.2.3 Транзакция подписки/отказа от подписки

Типичные примеры типов транзакций подписки:

- уведомьте меня обо всех будущих пользовательских логинах (учетных записей пользователей);
- доставка потока данных X.

Типичные примеры потока коммуникаций:

- "клиент" подписывается на сервис X; "сервер" автоматически поставляет данные "клиенту" всякий раз, когда новые данные для сервиса X становятся доступными;
- "клиент" подписывается на все предупреждающие сообщения от "сервера".

"Сервер" автоматически сообщает "клиенту" обо всех происходящих предупреждениях.

Потенциальные типы ответа:

- нет ответа;
- подтверждение приема сообщения запроса;
- уведомление о статусах обработки "сервером" (принятая команда, обработанная команда, ошибка/ОК и т. д.) — возможные состояния, определенные в соответствии с приложением;
- доставка требуемой информации, инициируемой "сервером" как индивидуальная транзакция доставки (в этом случае исходящая от "сервера", а не от "клиента").

Е.2 DCP пакет сообщения

DCP пакет сообщения — специально сформированный пакет DCP, несущий информацию, используемую в пределах транзакции от "клиента" к "серверу" или наоборот. Части, которые составляют такой DCP пакет, будут описаны в последующих пунктах.

Е.2.1 Общая структура

Структура DCP пакета сообщения показана на рисунке Е.1.

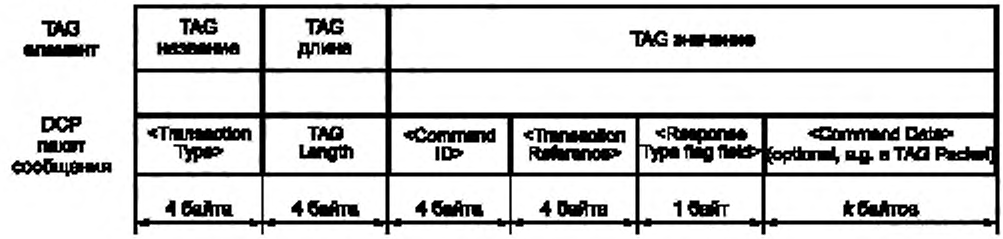


Рисунок Е.1 — Структура DCP пакета сообщения

Тип транзакции: одно из специальных TAG названий (4 символа ASCII), указывающее на DCP сообщение и его тип (подробнее см. Е.2.2) (таблица Е.1).

Таблица Е.1 — Типы транзакции

Транзакция	Запрос тип сообщения	Ответ тип сообщения
передачи	*tsq	*tss
выборки	*tfq	*tfs
подписки	*tuq	*tus
отказа от подписки	*tnq	*tns
доставки	*tdq	*tds

Длина TAG: длина секции TAG значение в битах (это касается каждого TAG пакета).

Идентификатор команды: указывает на индивидуальную команду; может быть закодирован как 4 "удобочитаемых" символа ASCII или как двоичное число; величины определены при использовании этой DCP структуры для двунаправленной связи.

Ссылка транзакции: номер ссылки, выбранный в качестве инициатора транзакции с новой командой запроса (например, случайное число), и используемый в любой последующей команде ответа, чтобы идентифицировать индивидуальную транзакцию (поток коммуникации); никакие два сообщения запросов, передаваемые между одним "клиентом" и одним "сервером", не должны, насколько возможно, использовать ту же самую комбинацию значения ID команды и значения ссылки транзакции; приложение должно определить минимальный интервал перед повторным использованием ссылок транзакции.

Поле флага типа ответа:

- для сообщений запроса указывает, какой тип ответов отправитель конкретного сообщения запроса желает получать; один или несколько флагов могут быть разрешены (установлены в "1");
- для сообщений ответа указывает, какие типы ранее затребованных ответов обработаны в пределах текущего сообщения ответа; каждый ответ может быть послан как индивидуальное сообщение ответа или несколько ответов могут содержаться в том же самом сообщении ответа; флаги, неизвестные получателю пакета, должны быть обработаны как "0" (и таким образом проигнорированы); подробнее — см. Е.2.3.

Данные команды:

- опционально — например, поле данных или полный TAG пакет;
- наличие, расположение и информационное содержание зависят от примененной спецификации и от индивидуального идентификатора команды;
- если представлены, могут нести подробную информацию, требуемую, например, для выборки или подписки на сообщения;
- если же форматированы как полный TAG пакет, могут всегда нести заранее известные TAG элементы, определенные в Е.2.4.

Примечание — Нет никакого ограничения на число сообщений запроса или ответа, которые переносятся в одном DCP пакете.

Е.2.2 Типы транзакций

Следующие типы транзакций определены для использования в пределах DCP структуры для двунаправленной связи.

Е.2.2.1 Транзакция передачи (запрос "*tsq", ответ "*tss")

"Клиент" запрашивает информацию от "сервера" (сообщение запроса) (рисунок Е.2). Произвольно одно или более сообщений ответа могут быть потребованы.

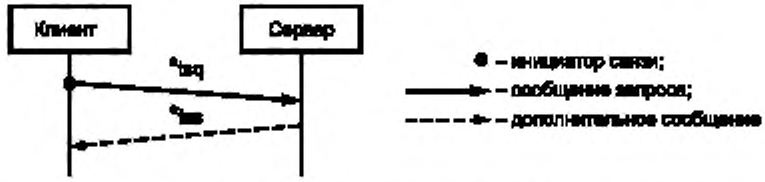


Рисунок Е.2 — Транзакция передачи

Е.2.2.2 Транзакция выборки (запрос "*tfq", ответ "*tfs")

"Клиент" запрашивает информацию от "сервера" (сообщение запроса) (рисунок Е.3). По крайней мере, одно сообщение ответа, содержащее требуемую информацию, отсылается назад.

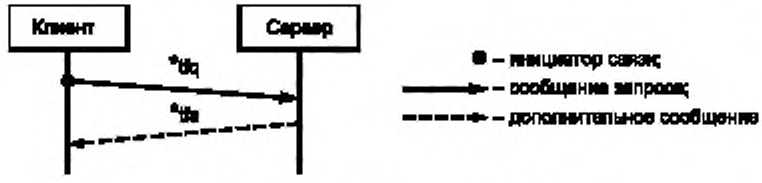


Рисунок Е.3 — Транзакция выборки

Е.2.2.3 Транзакция подписки/отказа от подписки (запрос "*tuq"/"*tnq", ответ "*tus"/"*tns"). Транзакция доставки (запрос "*tdq", ответ "*tds")

"Клиент" подписывается на, вероятно, не оперативную информацию, предлагаемую "сервером" (транзакция подписки) (рисунок Е.4). "Сервер" передает информацию "клиенту" всякий раз, когда (новые) данные становятся

доступны (транзакция доставки) (рисунок Е.4). "Сервер" приостанавливает доставку данных при приеме транзакции отказа от подписки или, например, после сбоя (перерыва).

Любая указанная выше транзакция состоит, по крайне мере, из сообщения запроса и, произвольно, одного или более сообщений ответа (например, как подтверждение приема).

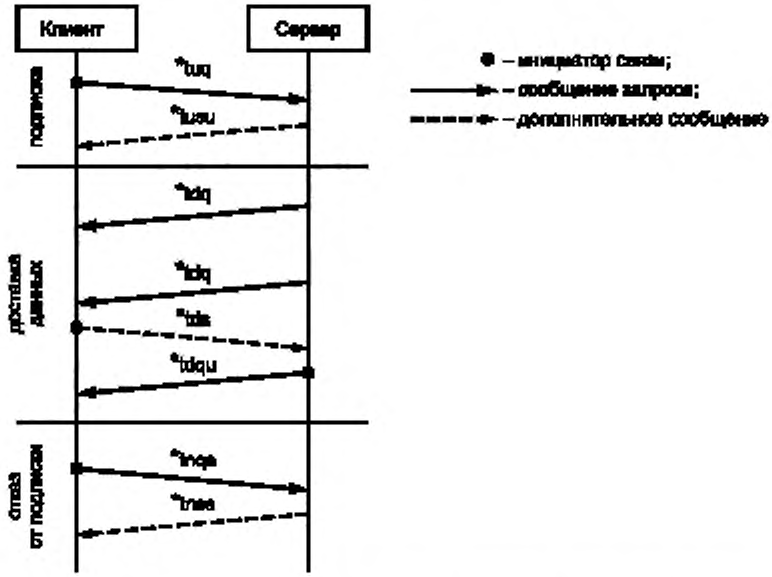


Рисунок Е.4 — Транзакция подписки/отказа от подписки; транзакция доставки

Е.2.3 Поле флага типа ответа

Поле флага типа ответа состоит из 8 битов. Каждый бит представляет специальный тип требуемого сообщения ответа (см. ниже).

В сообщении запроса сообщение ответа требуется для всех типов ответов, обозначенных флагами.

В сообщении ответа один или несколько наборов битов в связанном сообщении запроса могут указывать, что отмеченные ответы обработаны в соответствии с текущим сообщением ответа. Число отдельных сообщений ответа на конкретное сообщение запроса находится поэтому от 0 (флаг не был установлен) до числа флагов, указанных в сообщении запроса (каждый тип сообщения ответа посылается как отдельный DCP пакет). В последнем случае только один бит поля флага устанавливается в сообщении ответа.

Поле флага типа ответа имеет структуру, представленную на рисунке Е.5.



Рисунок Е.5 — Структура поля флага типа ответа

Стандартизированные типы ответа имеют следующее выражение:

- подтверждение приема: сообщение запроса было успешно получено (передача получила подтверждение);
- начало обработки: идентификатор известен и обработка может технически быть выполнена; произвольно, дополнительно представленные данные, связанные с командой, находятся в разрешенных пределах (контент относится к общему подтверждению);
- конец обработки: команда окончательно выполнена (контент относится к определенному подтверждению). Дополнительный TAG элемент состояния (часть данных команды) может предоставить информацию об успехе или неудаче.

Определение и использование всех других флагов определяется приложением, использующим DCP структуру для двунаправленной связи.

Примечание — Любое приложение, использующее эту DCP структуру для двунаправленной связи, должно, по крайней мере, поддерживать подтверждение приема. Поддержка и точное значение других двух стандартизированных флагов могут изменяться от приложения к приложению, однако они должны соответствовать значениям, указанным выше.

Е.2.4 Данные команды

Данные команды — необходимая информация, чтобы обработать текущее сообщение. Это может быть значение параметра, требуемое для конкретной команды в пределах отосланного сообщения запроса, значение, которое возвращено внутри ответного сообщения транзакции выборки, или доставки данных клиенту, который на них предварительно подписался.

Дополнительная секция данных команды пакета сообщения может нести в себе значение или полный TAG пакет. Наличие, расположение и содержание этого поля зависят от приложения и конкретного идентификатора команды (см. Е.2.4.1).

Кроме того, данные соответствующей команды могут также переноситься в верхнем иерархическом уровне сообщения DCP пакета, например, чтобы поддержать совместимость подписки данных доставки (см. Е.2.4.2).

Е.2.4.1 Поле данных команды, несущее полный TAG пакет

Если поле данных команды несет полный TAG пакет, оно может использоваться, чтобы транспортировать параметры, связанные с конкретным идентификатором команды, или транспортировать любую другую форму информации. Определение TAG элементов, содержащееся в TAG пакете, приложением, использующим DCP структуру для двунаправленной связи, может также зависеть от значения идентификатора команды.

Однако некоторые TAG элементы (таблица Е.2) предопределены в настоящем пункте, чтобы обеспечить единообразное решение для кодирования данных общего назначения (применяется, только если поле данных команды несет полный TAG пакет).

Таблица Е.2 — TAG элементы

TAG название	TAG значение	Описание
*sta	2 байта <индикатор состояния > + *n* байтов описания состояния; < индикатор состояния >: 4 бита — класс состояния + 12 битов — код состояния	Эти TAG элементы несут информацию о статусе, например как ответ на команду (успех/неудача и т. д.); описание состояния — простой форматированный текст UFT-8; класс состояния: 0 = OK (хорошо); 1 — предупреждение; 2 — ошибка; 3 — 14: специальное применение; 15: неопределенный/неизвестный; код состояния: специальное применение
*rqu	m байтов	Формат и длина определяются конкретным приложением и зависят от идентификатора команды
*rsp	p байтов	Формат и длина определяются конкретным приложением и зависят от идентификатора команды

TAG элементы "*rqu" и "*rsp" могут использоваться для транспортировки дополнительной информации, относящейся к подписке или сообщению ответа соответственно.

Е.2.4.2 Связанные данные команды в главном уровне иерархии пакета сообщения

В некоторых случаях целесообразно передавать информацию в главном уровне иерархии DCP пакета сообщения.

Пример этого — MDI поток, предварительно подписанный на Клиента и теперь доставленный Сервером: чтобы быть обработанным стандартным декодером MDI, фактические данные MDI должны быть расположены в главном уровне иерархии. Эти данные MDI только сопровождаются с сообщением доставки.

Другой пример — обратно совместимое расширение протоколов, в настоящее время несущих команды и связанные с командой данные на главном уровне. Любой запрос или ответ (в настоящее время находящиеся на главном уровне) могут просто дополнительно сопровождаться с отправленным сообщением или сообщением выборки соответственно (один дополнительный TAG элемент на команду). Это переданное сообщение или сообщение выборки использует, например, то же самое название команды, как определено для TAG названия команды (или идентификатора команды, где это применимо), и добавляет все особенности DCP структуры для двунаправленной связи без дублирования фактически полезных данных от главного уровня в секцию данных команды сообщения.

Библиография

[1] ETSI ES 201 980 v3.1.1 (2009—08) Digital Radio Mondiale (DRM): System Specification

[2] IETF RFC 3986 Uniform Resource Identifier (URI): Generic Syntax

УДК 621.396.97:681.327.8:006.354 ОКС 33.170

Ключевые слова: радиовещание, цифровое, DRM, радио, протокол DCP

Редактор переиздания *Н.Е. Рагузина*
Технический редактор *В.Н. Прусакова*
Корректор *И.А. Королева*
Компьютерная верстка *И.А. Налейкиной*

Сдано в набор 24.07.2020. Подписано в печать 01.09.2020. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 4,21.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru gost.e-ecolog.ru

gost.e-ecolog.ru