



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

Н.271

(05/2006)

СЕРИЯ Н: АУДИОВИЗУАЛЬНЫЕ И
МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб –
Кодирование движущихся видеоизображений

**Сообщения обратного канала
видеоизображения для доставки информации
о состоянии и запросов от приемника
видеоизображения к отправителю
видеоизображения**

Рекомендация МСЭ-Т Н.271

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Н
АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	H.100–H.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ УСЛУГ	
Общие положения	H.200–H.219
Мультиплексирование и синхронизация при передаче	H.220–H.229
Системные аспекты	H.230–H.239
Процедуры связи	H.240–H.259
Кодирование движущихся видеоизображений	H.260–H.279
Сопутствующие системные аспекты	H.280–H.299
Системы и оконечное оборудование для аудиовизуальных услуг	H.300–H.349
Архитектура услуг справочника для аудиовизуальных и мультимедийных услуг	H.350–H.359
Качество архитектуры обслуживания для аудиовизуальных и мультимедийных услуг	H.360–H.369
Дополнительные услуги для мультимедиа	H.450–H.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	H.500–H.509
Мобильность для мультимедийных систем и услуг серии Н	H.510–H.519
Приложения и услуги мобильной мультимедийной совместной работы	H.520–H.529
Защита мобильных мультимедийных систем и услуг	H.530–H.539
Защита приложений и услуг мобильной мультимедийной совместной работы	H.540–H.549
Процедуры мобильного взаимодействия	H.550–H.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	H.560–H.569
ШИРОКОПОЛОСНЫЕ И МУЛЬТИМЕДИЙНЫЕ TRIPLE-PLAY УСЛУГИ	
Предоставление широкополосных мультимедийных услуг по VDSL	H.610–H.619

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

Рекомендация МСЭ-Т Н.271

Сообщения обратного канала видеоизображения для доставки информации о состоянии и запросов от приемника видеоизображения к отправителю видеоизображения

Резюме

В этой Рекомендации определяется формат сообщений обратного канала для доставки информации о состоянии и запросов от приемника видеоизображения к отправителю видеоизображения.

Синтаксис сообщений был разработан в общем виде, чтобы сделать возможным его применение для большинства существующих международных стандартов кодирования видеоизображений. Определено применение таких общих сообщений к Рекомендациям МСЭ-Т Н.261, Н.263 и Н.264 | ISO/IEC 14496-10.

Источник

Рекомендация МСЭ-Т Н.271 утверждена 16-й Исследовательской комиссией МСЭ-Т 29 мая 2006 года (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

Ключевые слова

Сообщение обратного канала, обратная связь приемника, выбор эталонного изображения, видеоизображение.

ПРЕДИСЛОВИЕ

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

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

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

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

ПРИМЕЧАНИЕ

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

Соблюдение положений данной Рекомендации осуществляется на добровольной основе. Однако данная Рекомендация может содержать некоторые обязательные положения (например, для обеспечения функциональной совместимости или возможности применения), и в таком случае соблюдение Рекомендации достигается при выполнении всех указанных положений. Для выражения требований используются слова "следует", "должен" ("shall") или некоторые другие обязывающие выражения, такие как "обязан" ("must"), а также их отрицательные формы. Употребление таких слов не означает, что от какой-либо стороны требуется соблюдение положений данной Рекомендации.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или выполнение настоящей Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, действительности или применимости заявленных прав интеллектуальной собственности, независимо от того, доказываются ли такие права членами МСЭ или другими сторонами, не относящимися к процессу разработки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения настоящей Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что вышесказанное может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2009

Все права сохранены. Ни одна из частей данной публикации не может быть воспроизведена с помощью каких бы то ни было средств без предварительного письменного разрешения МСЭ.

СОДЕРЖАНИЕ

	Стр.
1 Сфера применения	1
2 Нормативные справочные документы	1
3 Определения	1
4 Сокращения	2
5 Соглашения о терминах.....	2
5.1 Арифметические операторы.....	2
5.2 Логические операторы.....	3
5.3 Операторы отношений.....	3
5.4 Поразрядные операторы	3
5.5 Операторы присваивания	3
5.6 Переменные, элементы синтаксиса и таблицы.....	4
5.7 Текстовое описание логических операций	4
5.8 Метод описания синтаксиса в табличной форме	5
5.9 Спецификация функций синтаксиса, категорий и дескрипторов	7
6 Полезная нагрузка сообщений.....	8
6.1 Синтаксис	8
6.2 Семантика	9
7 Особенности использования сообщений в стандартах.....	11
7.1 Особенность использования сообщений в стандарте Н.261	11
7.2 Особенность использования сообщений в стандарте Н.263	12
7.3 Особенность использования сообщений в стандарте Н.264	13

Введение

0.1 Цель

Для некоторых применений передача дополнительных данных (внутри или вне полосы) полезна для улучшения качества видеоизображения. В этой Рекомендации определяются данные полезной нагрузки для использования с различными технологиями кодирования видеоизображения. Данные полезной нагрузки определены в общем виде, для того чтобы сделать их применение возможным для общепринятых существующих стандартов кодирования видеоизображений. Определено применение таких общих сообщений к Рекомендациям МСЭ-Т Н.261, Н.263 и Н.264 | ИСО/МЭК 14496-10.

0.2 Обзор

Следующая информация может быть передана от приемника видеоизображения к отправителю видеоизображения с использованием сообщения(й) обратного канала, определенного в этой Рекомендации:

- Сообщения о состоянии:
 - одно или более изображений без обнаружения ошибки несовпадения битового потока;
 - потери на уровне изображения и/или уровня макроблока;
 - информация о важных данных, содержащихся в заголовке.
- Запросы на обновление:
 - запрос на "сброс", указывающий на то, что отправитель должен полностью обновить битовый поток видеоизображения, как если бы никакие прежде отправленные данные не были получены.

Определено применение сообщений обратного канала к Рекомендациям МСЭ-Т Н.261, Н.263 и Н.264.

Сообщения обратного канала видеоизображения для доставки информации о состоянии и запросов от приемника видеоизображения к отправителю видеоизображения

1 Сфера применения

В данной Рекомендации определяются сообщения обратного канала для поблочного кодирования видеоизображения.

2 Нормативные справочные документы

В перечисленных ниже Рекомендациях МСЭ-Т и другой справочной литературе содержатся положения, которые посредством ссылок на них в этом тексте составляют основные положения данной Рекомендации. На момент опубликования, действовали указанные редакции документов. Все Рекомендации и другая справочная литература, являются предметом корректировки, в связи с чем пользователям данной Рекомендации настоятельно рекомендуется изыскать возможность для использования самых последних изданий Рекомендации и справочной литературы, перечисленной ниже. Регулярно публикуется перечень действующих Рекомендаций МСЭ-Т. Ссылка на документ в рамках этой Рекомендации не даёт ему, как отдельному документу, статуса Рекомендации.

- ITU-T Recommendation H.261 (1993), *Video codec for audiovisual services at $p \times 64$ kbit/s*.
 - Рекомендация МСЭ-Т Н.263 (2005 г.), *Кодирование видеосигнала для низкоскоростной связи*.
 - Рекомендация МСЭ-Т Н.264 (2005 г.), *Усовершенствованное кодирование изображений для общих аудиовизуальных услуг*.
- ISO/IEC 14496-10:2005, *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*.

3 Определения

В этой Рекомендации определены следующие термины:

- 3.1 обратный канал:** Средство доставки сообщений обратного канала от приемника битового потока видеоизображения к отправителю битового потока видеоизображения.
- 3.2 сообщение обратного канала:** Сообщение, которое создается приемником *битового потока* видеоизображения, и в котором содержится информация о состоянии приемника или запросы.
- 3.3 битовый поток:** Последовательность битов, которая образует представление кодированных *изображений* и связанных с ними данных, формирующая одну или более кодированных видео последовательностей.
- 3.4 ошибка битового потока:** Искаженный или неполный *битовый поток*.
- 3.5 ошибка несовпадения битового потока:** Разница между значением или количеством декодированных *изображений*, появившаяся из-за одной или более *ошибок битового потока* в сравнении со значением или количеством декодированных *изображений*, полученных в процессе декодирования битового потока, не содержащего ошибок.
- 3.6 блок:** Матрица $M \times N$ (M -столбцов на N -рядов) образцов яркости и соответствующих образцов цветности.
- 3.7 обнаруженная ошибка битового потока:** Ошибка *битового потока*, обнаруженная приемником.
- 3.8 обнаруженная ошибка несовпадения битового потока:** Ошибка несовпадения *битового потока*, которая может появиться из-за ошибок в *битовом потоке*.
- 3.9 порядок декодирования:** Порядок обработки *элементов синтаксиса* в процессе декодирования определенный технологией кодирования видеоизображения.

3.10 макроблок: Один блок 16x16 образцов яркости и два соответствующих блока образцов цветности.

3.11 множество значений параметра: Структура синтаксиса, содержащая некоторое количество элементов синтаксиса, которые могут быть использованы в процессе декодирования для одного или более изображений.

3.12 изображение: Собираемый термин для поля или кадра видеоизображения, которое кодируется как отдельная единица с помощью технологии кодирования видеоизображения.

3.13 эталонное изображение: Изображение, содержащее образцы, которые могут быть использованы для предсказания между изображениями в процессе декодирования последовательных изображений в битовом потоке видеоизображения.

3.14 зарезервировано: Элемент зарезервирован, если его использование в разделах, определяющих некоторые значения конкретного элемента синтаксиса, отложено для будущего применения организацией МСЭ-Т. Эти значения не должны использоваться в сообщениях обратного канала, соответствующих этой Рекомендацией, но могут быть использованы в будущих расширениях этой Рекомендации организацией МСЭ-Т.

3.15 элемент синтаксиса: Элемент данных, представленный в битовом потоке или в сообщении обратного канала.

4 Сокращения

В этой Рекомендации используются следующие сокращения:

CRC	Cyclic Redundancy Code	Циклический избыточный код
LSB	Least Significant Bit	Младший значащий бит
MSB	Most Significant Bit	Старший значащий бит

5 Соглашения о терминах

По всему тексту этой Рекомендации высказывания, появляющиеся с преамбулой "ПРИМЕЧАНИЕ" – являются информативными и не представляют собой неотъемлемую часть данной Рекомендации.

ПРИМЕЧАНИЕ. – Математические операторы, использованные в этой Рекомендации подобны применяемым в языке программирования C. Нумерация и подсчет соглашений обычно начинаются с 0.

5.1 Арифметические операторы

Следующие арифметические операторы задаются таким образом:

+	сложение;
–	вычитание (как оператор двух аргументов) или отрицание (как унарный оператор префикса);
*	умножение;
/	целочисленное деление с округлением результата в сторону нуля. Например, для $7/4$ и $(-7)/(-4)$ округление до 1, а для $(-7)/4$ и $7/(-4)$ округление до -1 ;
$x \% y$	модуль. Остаток от деления x на y , определяется только для целочисленных значений x и y , при $x \geq 0$ и $y > 0$.

Если порядок приоритетных действий не указан явно с использованием скобок, то применяются следующие правила:

- операции умножения и деления выполняются до сложения и вычитания;
- операции умножения и деления выполняются последовательно слева направо;
- операции сложения и вычитания выполняются последовательно слева направо.

5.2 Логические операторы

Следующие логические операторы определяются таким образом:

<code>x && y</code>	Булево логическое "и" для <code>x</code> и <code>y</code> ;
<code>x y</code>	Булево логическое "или" для <code>x</code> и <code>y</code> ;
<code>!</code>	Булево логическое "не";
<code>x ? y : z</code>	если <code>x</code> TRUE или не равно 0, то он устанавливается равным <code>y</code> ; иначе он устанавливается равным <code>z</code> ;

5.3 Операторы отношений

Следующие операторы отношения определяются таким образом:

<code>></code>	больше чем;
<code>>=</code>	больше чем или равно;
<code><</code>	меньше чем;
<code><=</code>	меньше чем или равно;
<code>==</code>	равно;
<code>!=</code>	не равно.

5.4 Поразрядные операторы

Следующие поразрядные операторы определяются таким образом:

<code>&</code>	поразрядное "и". Если действие выполняется над целочисленными аргументами, то оперируют целочисленными значениями, представленными в дополнительном двоичном коде. Если действие выполняется над двоичным аргументом, который содержит меньше разрядов, чем другой аргумент, то более короткий аргумент удлиняется прибавлением дополнительных значащих битов, равных 0;
<code> </code>	поразрядное "или". Если действие выполняется над целочисленными аргументами, то оперируют целочисленными значениями, представленными в дополнительном двоичном коде. Если действие выполняется над двоичным аргументом, который содержит меньше разрядов, чем другой аргумент, то более короткий аргумент удлиняется прибавлением дополнительных значащих битов, равных 0;
<code>^</code>	поразрядное "исключающее или". Если действие выполняется над целочисленными аргументами, то оперируют целочисленными значениями, представленными в дополнительном двоичном коде. Если действие выполняется над двоичным аргументом, который содержит меньше разрядов, чем другой аргумент, то более короткий аргумент удлиняется прибавлением дополнительных значащих битов, равных 0;
<code>x >> y</code>	арифметический сдвиг вправо целочисленного <code>x</code> , выраженного в дополнительном двоичном представлении, на <code>y</code> двоичных разрядов. Эта функция определена только для положительных целочисленных значений <code>y</code> . Разряды, сдвинутые в сторону MSB (старших значащих битов) в результате сдвига вправо, имеют значения, равные MSB аргумента <code>x</code> до операции сдвига;
<code>x << y</code>	арифметический сдвиг влево целочисленного <code>x</code> , выраженного в дополнительном двоичном представлении, на <code>y</code> двоичных разрядов. Эта функция определена только для положительных целочисленных значений <code>y</code> . Разряды, сдвинутые в сторону LSB (младших значащих битов) в результате сдвига влево, имеют значения, равные 0.

5.5 Операторы присваивания

Следующие арифметические операторы определяются таким образом:

<code>=</code>	оператор присваивания;
<code>++</code>	приращение, например, <code>x++</code> эквивалентно <code>x = x + 1</code> , если используется в индексе множества, то равно значению переменной до операции приращения;
<code>--</code>	отрицательное приращение, например, <code>x--</code> эквивалентно <code>x = x - 1</code> ; если используется в индексе множества, то равно значению переменной до операции отрицательного приращения;

- $+=$ приращение на определенную величину, например, $x += 3$ эквивалентно $x = x + 3$, а $x += (-3)$ эквивалентно $x = x + (-3)$;
- $- =$ отрицательное приращение на определенную величину, например, $x - = 3$ эквивалентно $x = x - 3$, а $x - = (-3)$ эквивалентно $x = x - (-3)$.

5.6 Переменные, элементы синтаксиса и таблицы

Элементы синтаксиса в битовом потоке представлены **жирным** шрифтом. Каждый элемент синтаксиса представлен своим названием (все строчные буквы с символами подчеркивания), собственной одной или двумя категориями и одним или двумя дескрипторами для собственного способа кодированного представления. Проведение процесса декодирования зависит от значения элемента синтаксиса и значений, ранее декодированных элементов синтаксиса. Если значение элемента синтаксиса используется в таблицах синтаксиса или в тексте, то оно обозначается обычным (не жирным) шрифтом.

В некоторых случаях в таблицах синтаксиса могут использоваться значения других переменных, образованных от значений элементов синтаксиса. Такие переменные появляются в таблицах синтаксиса или в тексте и их названия состоят из набора строчных и заглавных букв и не содержат символов подчеркивания. Переменные, названия которых начинаются с заглавных букв, образованы для декодирования текущей структуры синтаксиса и всех зависимых структур синтаксиса. Переменные, названия которых начинаются с заглавных букв, могут использоваться в процессе декодирования для более поздних структур синтаксиса, в которых упоминается первоначальная структура синтаксиса данной переменной. Переменные, названия которых начинаются со строчных букв, используются только в рамках того подпункта, в котором они образованы.

В некоторых случаях "мнемонические" названия значений элемента синтаксиса или значений переменных используются попеременно с их численными значениями. Иногда "мнемонические" названия используются без каких-либо связанных численных значений. Связь значений и названий определяется в тексте. Названия образуются от одной или более групп букв, разделенных символом подчеркивания. Каждая группа начинается с заглавной буквы и может содержать и другие заглавные буквы.

ПРИМЕЧАНИЕ. – Описание синтаксиса сходно с описанием конструкций синтаксиса языка-C.

Функции определяются их названиями, которые образованы как названия элементов синтаксиса с направленными влево и вправо круглыми скобками, содержащими ноль или более названий переменных (для определений), или значения (для использования), разделенные между собой запятыми (если имеется более одной переменной).

Двоичное представление указывается ограничением строки значений разрядов одинарными кавычками. Например, "01000001" представляет восьмиразрядную строку, в которой только второй и последний разряды равны 1.

Шестнадцатеричная система счисления, которая указывается префиксным числом "0x", может использоваться вместо двоичной системы, если количество разрядов является целым числом, кратным 4. Например, 0x41 представляет восьмиразрядную строку, в которой второй и последний разряды равны 1.

Численные значения, не заключенные в одинарные кавычки и не имеющие префикса "0x", являются десятичными значениями.

Значение, равное 0, представляет условие FALSE в инструкции проверки. Значение TRUE представлено любым другим значением, отличающимся от нуля.

5.7 Текстовое описание логических операций

В тексте, инструкция логических операций, что была бы описана в псевдокоде как

```
if (condition 0)
    statement 0
else if (condition 1)
    statement 1
...
else /* информативное замечание об оставшемся условии */
    statement n;
```

может быть описана следующим образом:

... as follows / ... the following applies.

- If condition 0, statement 0;
- Otherwise, if condition 1, statement 1;
- ...;
- Otherwise (информативное замечание об оставшемся условии), statement n.

Каждая инструкция "If...Otherwise, if...Otherwise, ..." в тексте вводится с помощью "... as follows" или "... the following applies", за которым сразу же следует "If ... ". Последнее условие "If...Otherwise, if...Otherwise, ..." это всегда "Otherwise, ...". Чередующиеся инструкции "If...Otherwise, if...Otherwise, ..." могут быть идентифицированы согласованием "... as follows" или "... the following applies" с окончанием "Otherwise, ...".

В тексте, инструкция логических операций, что была бы описана в псевдокоде как

```
if( condition 0a && condition 0b )
    statement 0
else if ( condition 1a || condition 1b )
    statement 1
...
else
    statement n;
```

может быть описана следующим образом:

... следующим образом / ... применяется следующее.

- Если все следующие условия справедливы, то инструкция 0:
 - condition 0a;
 - condition 0b.
- Иначе, если любые из следующих условий справедливы, то инструкция 1:
 - condition 1a;
 - condition 1b.
- ...
- Иначе, инструкция n.

В тексте, инструкция логических операций, что была бы описана в псевдокоде как

```
if ( condition 0 )
    statement 0
if ( condition 1 )
    statement 1;
```

может быть описана следующим образом:

When condition 0, statement 0.

When condition 1, statement 1.

5.8 Метод описания синтаксиса в табличной форме

В следующих таблицах перечислены примеры псевдокода, использующегося для описания синтаксиса. Появление **syntax_element** означает, что элемент синтаксиса взят из битового потока, а указатель битового потока перемещается на следующую позицию ниже элемента синтаксиса в процессе анализа битового потока.

	Дескриптор
/* Инструкция может состоять из элемента синтаксиса со связанной категорией синтаксиса и дескриптором или это может быть выражение, которое используется для задания условий существования, типа и количества элементов синтаксиса, как в следующих двух примерах */	
syntax_element	ue(v)
Создание условий для инструкции или структуры {	
/* Группа инструкций, заключенная в фигурные скобки, является составной инструкцией, с которой функционально обращаются как с единой инструкцией. */	
statement	
statement	
...	
}	
/* Структура "while" является инструкцией создания условия, которая задает проверку истинности условия и, если условие истинно, то задаётся выполнение инструкции (или составной инструкции) повторно, до тех пор, пока условие не перестанет быть истинным */	
while(condition)	
statement	
/* Структура "do ... while" является структурой создания условия, которая задает выполнение инструкции однократно, затем следует проверка истинности условия и, если условие истинно, то она задает повторяющееся выполнение инструкции до тех пор, пока условие не перестанет быть истинным */	
do	
statement	
while(condition)	
/* Структура "if ... else" задает проверку истинности условия и, если условие истинно, то задает выполнение первоначальной инструкции, иначе (если условие ложно), она задает проверку альтернативной инструкции. Часть "else" этой структуры и связанная альтернативная инструкция опускаются, если не требуется выполнение альтернативной инструкции*/	
if(condition)	
primary statement	
else	
alternative statement	
/* Структура "for" задает выполнение исходной инструкции, за которой следует проверка условия и, если условие истинно, то она задает повторяющееся выполнение первоначальной инструкции и последующей инструкции до тех пор, пока условие не перестанет быть истинным. */	
for(initial statement; condition; subsequent statement)	
primary statement	

5.9 Спецификация функций синтаксиса, категорий и дескрипторов

Представленные здесь функции используются в описаниях синтаксиса. Эти функции предполагают существование указателя битового потока с указанием позиции следующего бита, который будет считан из битового потока в процессе декодирования.

`byte_aligned()` определяется следующим образом:

- если текущая позиция в битовом потоке находится на границе байта, т. е., следующий бит в битовом потоке является первым битом в байте, то возвращаемое значение элемента `byte_aligned()` равно значению `TRUE`;
- иначе, возвращаемое значение элемента `byte_aligned()` равно значению `FALSE`.

`more_msg_data()` задается следующим образом:

- если в структуре синтаксиса `msg_data()` имеется больше данных, то возвращаемое значение элемента `more_msg_data()` равно значению `TRUE`;
- иначе, возвращаемое значение элемента `more_msg_data()` равно значению `FALSE`.

Метод, запускающий определение наличия большего количества данных в структуре синтаксиса `msg_data()`, задается программным приложением:

`next_bits(n)` предоставляет следующие биты в битовом потоке для сравнения, без перемещения указателя битового потока. Обеспечивает просмотр следующих `n` бит в битовом потоке, где `n` является его аргументом;

`read_bits(n)` считывает следующие `n` бит в битовом потоке и перемещает указатель битового потока на `n`-бит позиций. Если `n` равно 0, то элемент `read_bits(n)` задается для возвращения значения равного 0 и для того, чтобы указатель битового потока не был перемещен.

Процесс анализа каждого элемента синтаксиса задается следующими дескрипторами:

- `f(n)`: фиксированная структурная битовая строка, в которой используются `n` бит, написанных (слева направо), в которой первый бит находится на левой позиции. Процесс анализа для этого дескриптора задается возвращаемым значением элемента `read_bits(n)`;
- `u(n)`: целое число без знака, в котором используются `n` бит. Если `n` – это "v" в таблице синтаксиса, то количество бит изменяется в зависимости от значения других элементов синтаксиса. Процесс анализа для этого дескриптора задается возвращаемым значением элемента `read_bits(n)`, интерпретированным как двоичное представление целого числа без знака со старшим значащим битом, написанным на первом месте;
- `ue(v)`: элемент синтаксиса, являющийся целым числом без знака и закодированный в системе Exp-Golomb, в котором первый бит находится на левой позиции. Процесс анализа для этого дескриптора задается следующим образом.

Элементы синтаксиса закодированы как `ue(v)` – это означает, что они закодированы в системе Exp-Golomb. Процесс анализа для этих элементов синтаксиса начинается со считывания тех битов, которые начинаются на текущей позиции в битовом потоке, и продолжается далее, включая первый ненулевой бит, и с подсчетом количества начальных битов, равных 0. Этот процесс должен быть эквивалентен следующему:

```
leadingZeroBits = -1;
for ( b = 0; !b; leadingZeroBits++ ),
    b = read_bits( 1 ).
```

Переменная `codeNum` затем задается следующим образом:

```
codeNum = 2leadingZeroBits - 1 + read_bits( leadingZeroBits ),
```

где значение, возвращаемое от элемента `read_bits( leadingZeroBits )`, интерпретируется как двоичное представление целого числа без знака со старшим значащим битом, написанным на первом месте.

Значение элемента синтаксиса равно переменной `codeNum`.

6 Полезная нагрузка сообщений

6.1 Синтаксис

msg_data() {	Дескриптор
do	
message()	
while(more_msg_data())	
}	

message() {	Дескриптор
payloadType = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* равно 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* равно 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
msg_payload(payloadType, payloadSize)	
}	

msg_payload(payloadType, payloadSize) {	Дескриптор
if(payloadType <= 4)	
ref_pic_id	u(32)
if(payloadType == 0) {	
num_ref_pics_minus1	ue(v)
for(i = 1; i <= num_ref_pics_minus1; i++)	
good_ref_pic_id [i]	u(32)
} else, if(payloadType == 1)	
delta_ref_pic_id	ue(v)
else, if(payloadType == 2) {	
data_partition_idc	ue(v)
run_length_flag	u(1)
if(run_length_flag) {	
first_blk_lost	ue(v)

num_blks_lost_minus1	ue(v)
} else {	
top_left_blk	ue(v)
bottom_right_blk	ue(v)
}	
} else,if((payloadType == 3) (payloadType == 4)) {	
param_set_type	ue(v)
param_set_crc	u(16)
if(payloadType == 3)	
param_set_id	ue(v)
}	
stop_one_bit /* равно 1 */	f(1)
while(!byte_aligned())	
alignment_zero_bit /* равно 0 */	f(1)
}	

6.2 Семантика

ff_byte должно быть равно 0xFF.

last_payload_type_byte является последним байтом, использованным для задания полезной нагрузки сообщения.

last_payload_size_byte является последним байтом, использованным для задания размера сообщения.

Размер структуры синтаксиса msg_payload() должен быть равен 8 битам payloadSize *.

В соответствии со значениями payloadType, в таблице 6-1 указаны типы сообщений.

Таблица 6-1/Н.271 – Типы сообщений

payloadType	Сообщение
0	Одно или более изображений без обнаружения ошибки несовпадения битового потока.
1	Одно или более изображений, потерянных частично или полностью.
2	Набор блоков одного изображения, потерянных частично или полностью.
3	Контроль CRC (циклический избыточный код) для одного множества значений параметра.
4	Контроль CRC для всех множеств значений параметра конкретного типа.
5	Запрос на "сброс", указывающий на то, что отправитель должен полностью обновить битовый поток видеоизображения, как если бы никакие прежде отправленные данные не были получены.
> 5	Зарезервировано для будущего применения организацией МСЭ-Т.

Сообщения, в которых payloadType больше 5, должны быть удалены и отброшены в соответствии с размером сообщения, как указано переменной payloadSize.

ref_pic_id идентифицирует изображение. В зависимости от значения переменной payloadType, в таблице 6-2 определяется семантика элемента ref_pic_id.

Таблица 6-2/Н.271 – Семантика элемента ref_pic_id

payloadType	Семантика ref_pic_id
0	Задаёт изображение, потерянное частично или полностью без обнаружения ошибки несовпадения битового потока.
1	Задаёт изображение, потерянное частично или полностью.
2	Задаёт изображение, потерянное частично.
3	Задаёт изображение. Присутствует контроль CRC для одного множества значений параметра, связанного с заданным изображением.
4	Задаёт изображение. Присутствует контроль CRC для всех множеств значений параметра конкретного типа, накопленных за время, связанное с декодированием заданного видеоизображения.

num_ref_pics_minus1 плюс 1 задаёт количество изображений без обнаружения ошибки несовпадения битового потока. Значение num_ref_pics_minus1 должно быть в диапазоне от 0 до 31, включительно.

good_ref_pic_id[i] задаёт 1-е декодированное изображение, указанное в текущем сообщении, без обнаружения ошибки несовпадения битового потока.

delta_ref_pic_id задаёт приращение относительно элемента ref_pic_id, распознающего набор изображений, которые потеряны частично или полностью. Значение delta_ref_pic_id должно быть в диапазоне от 0 до 31, включительно.

data_partition_idc указывает на то, что все данные (если data_partition_idc равно 0) или часть данных (если data_partition_idc не равно 0) для блоков, заданных с помощью top_left_blk и bottom_right_blk потеряны. Значение элемента data_partition_idc должно быть в диапазоне от 0 до 15, включительно.

run_length_flag равное 1, указывает наличие элементов синтаксиса first_blk_lost и num_blks_lost_minus1. run_length_flag равное 0, указывает присутствие элементов синтаксиса top_left_blk и bottom_right_blk.

first_blk_lost указывает адрес первого блока в порядке растрового сканирования. Адрес блока – это индекс блока в растровом сканировании блока изображения, начинающийся с нуля для самого верхнего левого блока изображения.

num_blks_lost_minus1 плюс 1 задаёт количество последовательных блоков в порядке растрового сканирования, первый блок в котором задается с помощью first_blk_lost.

top_left_blk и **bottom_right_blk** задаёт адреса самого верхнего левого и самого нижнего правого блоков, соответственно, прямоугольной области, которая полностью или частично потеряна. Следующие ограничения должны соблюдаться для значений элементов синтаксиса top_left_blk и bottom_right_blk, где переменная PicWidthInBlks представляет собой ширину изображения, единицей измерения которой служит блок.

- top_left_blk должно быть меньше или равно bottom_right_blk.
- bottom_right_blk должно быть меньше, чем размер изображения, выраженный в блоках.
- top_left_blk % PicWidthInBlks должно быть меньше или равно bottom_right_blk % PicWidthInBlks.

param_set_type задаёт тип множеств(а) значений параметра. Значение элемента param_set_type должно быть в диапазоне от 0 до 15, включительно.

param_set_crc является контролем CRC для множества значений параметра, заданных с помощью param_set_id и param_set_type, если payloadType равно 3 и имеется контроль CRC для всех множеств значений параметра этого типа, заданных с помощью param_set_type, если payloadType равно 4.

Значение param_set_crc должно быть равно значению crcVal , полученному при выполнении следующего процесса псевдо-кодирования.

```
paramSet[pLen] = 0
paramSet[pLen + 1] = 0
crcVal = 0xFFFF
for( bitIdx = 0; bitIdx < ( pLen + 2 ) * 8; bitIdx++ ) {
    crcMsb = ( crcVal >> 15 ) & 1
    bitVal = ( paramSet[bitIdx >> 3] >> ( 7 - ( bitIdx & 7 ) ) ) & 1
    crcVal = ( ( ( crcVal << 1 ) + bitVal ) & 0xFFFF ) ^ ( crcMsb * 0x1021 )
},
```

(6-1)

где переменная paramSet – это строка байтов, содержащая в начальных байтах данные, для которых производится контрольное вычисление CRC; переменная pLen – это количество байтов данных, для которых производится контрольное вычисление CRC; а строка байтов переменной paramSet достаточно длинная для присоединения двух дополнительных байтов с нулевыми значениями к концу данных, для которых производится контрольное вычисление CRC.

Если payloadType равно 3, то paramSet содержит в последовательности байтов сети байты принятого множества значений параметра, которое определено с помощью param_set_id и param_set_type.

Если payloadType равно 4, то paramSet содержит в последовательности байтов сети сцепление байтов всех множеств значений параметра, тип которых определяется с помощью param_set_type, в порядке возрастания идентификатора множества значений параметра. Любое множество значений параметра, которое никогда не было принято, должно быть представлено двумя байтами, первый из которых содержит восемь MSB, а второй байт содержит восемь LSB 16-битного двоичного представления без знака значения идентификатора множества значений параметра.

param_set_id задает идентификатор множества значений параметра, тип которого задается с помощью param_set_type, для которого существует контроль CRC. Значение param_set_id должно быть в диапазоне от 0 до 65535, включительно.

stop_one_bit должно быть равно 1.

alignment_zero_bit должно быть равно 0.

7 Особенности использования сообщений в стандартах

В этом пункте n-ый бит переменной относится к n-му младшему значащему биту двоичного представления переменной. LSB считается нулевым битом.

7.1 Особенность использования сообщений в стандарте H.261

Значение payloadType должно быть равно 0, 1, 2, или 5. Сообщения с другими значениями payloadType должны быть проигнорированы.

Если имеются ref_pic_id или good_ref_pic_id[i], то 5 разрядов LSB этих элементов синтаксиса показывают значение TR изображения, которое равно ref_pic_id & 0x1F или good_ref_pic_id[i] & 0x1F. Оставшиеся биты ref_pic_id или good_ref_pic_id[i], зарезервированные для будущего применения организацией МСЭ-Т, должны быть равны 0, и не должны учитываться.

В зависимости от значения `payloadType`, применяется следующее:

- если `payloadType` равно 0, то изображение, у которого `TR` равно `ref_pic_id & 0x1F` или `good_ref_pic_id[i] & 0x1F`, является изображением без обнаружения ошибки несовпадения битового потока;
- иначе, если `payloadType` равно 1, то это указывает на то, что все изображения, поступающие в порядке декодирования и начинающиеся с изображения, у которого `TR` равно `ref_pic_id & 0x1F`, вплоть до и включая изображение, у которого `TR` равно $((\text{ref_pic_id} \& 0x1F) + \text{delta_ref_pic_id}) \% 32$, если такие изображения были представлены в битовом потоке видеоизображения, были полностью или частично потеряны;
- иначе, если `payloadType` равно 2, то изображение, у которого `TR` равно `ref_pic_id & 0x1F` было частично потеряно. Каждый блок является макроблоком. Значение `data_partition_idc` должно быть равно 0, что указывает на то, что все данные идентифицированных макроблоков считаются потерянными. Все другие значения `data_partition_idc` зарезервированы для будущего применения организацией МСЭ-Т и не должны учитываться;
- иначе, (`payloadType` равно 5), дальнейшая спецификация отсутствует.

7.2 Особенность использования сообщений в стандарте H.263

Функции стандарта H.263, которые поддерживаются имеющимся синтаксисом, включают в себя следующее:

- долговременные изображения;
- разделение данных.

Следующие функции стандарта H.263 не поддерживаются имеющимся синтаксисом:

- чередующиеся поля.

Значение `payloadType` должно быть равно 0, 1, 2, или 5. Сообщения с другими значениями `payloadType` не должны учитываться.

Если присутствует `ref_pic_id` или `good_ref_pic_id[i]`, то 12 LSB показывают значение переменной `picIdentifier`, которое равно `ref_pic_id & 0x0FFF` или `good_ref_pic_id[i] & 0x0FFF`. Если используется Приложение U/H.263 и `payloadType` равно 0, то 12-ый бит показывает, является ли сохраненное изображение долговременным изображением (если этот бит равен 1) или кратковременным изображением (если этот бит равен 0). 13-ый бит должен быть равен 0 до тех пор, пока не используется дополнительный Временной, SNR и режим Пространственный масштабируемости (Приложение O/H.263). Если 13-ый бит равен 1, то сообщение относится к расширенному уровню (а не к базовому уровню), а биты с 14 до 17, включительно, содержат значение `ELNUM`, которое равно `ref_pic_id & 0x03C000` или `good_ref_pic_id[i] & 0x03C000`. Оставшиеся биты `ref_pic_id` или `good_ref_pic_id[i]` зарезервированы для будущего применения организацией МСЭ-Т; они должны быть равны 0 и не должны учитываться,

- если Приложение U/H.263 не используется, то `picIdentifier` указывает значение `TR` изображения. Значение `TR` должно быть меньше, чем `MaxTR`, что равно максимально возможному значению `TR` плюс 1. В зависимости от значения `payloadType` применяется следующее:
 - если `payloadType` равно 0, то 12-ый бит элемента `ref_pic_id` или `good_ref_pic_id[i]` должен быть равен 0. Изображение, у которого `TR` равно `picIdentifier`, и, если 13-ый бит элемента `ref_pic_id` или `good_ref_pic_id[i]` равен 1, причем `ELNUM` равно `ref_pic_id & 0x03C000` или `good_ref_pic_id[i] & 0x03C000`, то оно является изображением без обнаружения ошибки несовпадения битового потока;
 - иначе, если `payloadType` равно 1, то 12-ый бит `ref_pic_id` должен быть равен 0. Все изображения, поступающие в порядке декодирования и в базовом уровне или в том же самом расширенном уровне с `ELNUM` равным `ref_pic_id & 0x03C000`, начинающиеся с изображения, у которого `TR` равно `ref_pic_id & 0x0FFF`, вплоть до и включая изображение, у которого `TR` равно $((\text{ref_pic_id} \& 0x0FFF) + \text{delta_ref_pic_id}) \% \text{MaxTR}$, если такие изображения были представлены в битовом потоке видеоизображения, считаются полностью или частично потерянными;

- иначе, если payloadType равно 2, то 12-ый бит ref_pic_id должен быть равен 0. Изображение, у которого TR равно picIdentifier и, если 13-ый бит ref_pic_id равен 1, а ELNUM равно ref_pic_id & 0x03C000, частично потеряно. Значение data_partition_idc указывает на то, что все данные (если data_partition_idc равно 0), раздел данных заголовка (если data_partition_idc равно 1), раздел вектора перемещения (если data_partition_idc равно 2) или раздел коэффициента данных (если data_partition_idc равно 3) обнаруженных макроблоков потеряны. Все другие значения data_partition_idc зарезервированы для будущего применения организацией МСЭ-Т и не должны учитываться;
- иначе, (payloadType равно 5), дальнейшая спецификация отсутствует.
- Иначе (используется Приложение U/H.263), в зависимости от значения типа полезной нагрузки, применяется следующее,
 - если payloadType равно 0, то ref_pic_id или good_ref_pic_id[i] распознает сохраненное изображение без обнаружения ошибки несовпадения битового потока. Переменная picIdentifier показывает значение PN сохраненного изображения (если сохраненное изображение является кратковременным) или LPIN (если сохраненное изображение является долговременным):
 - если изображение является кратковременным изображением, то значение picIdentifier должно быть меньше, чем MaxPN, которое является максимально возможным значением PN плюс 1;
 - иначе (изображение является долговременным), значение picIdentifier должно быть меньше, чем MaxLPIN, которое является максимально возможным значением LPIN плюс 1.
- Иначе, если payloadType равно 1, то 12-ый бит ref_pic_id должен быть равен 0, а picIdentifier показывает значение PN изображения. Значение PN должно быть меньше, чем MaxPN, что равно максимально возможному значению PN плюс 1. Все изображения, поступающие в порядке декодирования в одном и том же расширенном или базовом уровне, начинающиеся с изображения, у которого PN равно ref_pic_id & 0x0FFF, вплоть до и включая изображение, у которого PN равно ((ref_pic_id & 0x0FFF) + delta_ref_pic_id) % MaxPN, если такие изображения были представлены в битовом потоке видеоизображения, считаются полностью или частично потерянными. MaxPN равно максимально возможному значению PN плюс 1.
- Иначе, если payloadType равно 2, то 12-ый бит ref_pic_id должен быть равен 0, а picIdentifier показывает значение PN изображения. Значение PN должно быть меньше, чем MaxPN, которое является максимально возможным значением PN плюс 1. Изображение, у которого TR равно picIdentifier и, если 13-ый бит ref_pic_id равен 1, а ELNUM равно ref_pic_id & 0x03C000, является частично потерянным. Каждый блок является макроблоком. Значение data_partition_idc показывает, что все данные (если data_partition_idc равно 0), раздел данных заголовка (если data_partition_idc равно 1), раздел вектора перемещения (если data_partition_idc равно 2) или раздел коэффициента данных (если data_partition_idc равно 3) распознанных макроблоков потеряны. Все другие значения data_partition_idc зарезервированы для будущего применения организацией МСЭ-Т и не должны учитываться.
- Иначе (payloadType равно 5), дальнейшая спецификация отсутствует.

7.3 Особенность использования сообщений в стандарте H.264

Функции стандарта H.264, которые поддерживаются имеющимся синтаксисом, включают в себя следующее:

- долговременные эталонные изображения;
- разделение данных;
- последовательность и множество значений параметра изображения.

Следующие функции стандарта H.264 не поддерживаются имеющимся синтаксисом:

- кадры, закодированные с помощью кодирования кадр-поле, приспособленного для макроблоков;
- изображения поля.

Если `ref_pic_id` или `good_ref_pic_id[ i ]` присутствует, то применяется следующее. 16 разрядов LSB показывают значение переменной `picIdentifier`, которое равно `ref_pic_id & 0xFFFF` или `good_ref_pic_id[ i ] & 0xFFFF`. Если `payloadType` равно 0, то 16-ый бит показывает, является ли эталонное изображение долговременным эталонным изображением (если этот бит равен 1) или кратковременным эталонным изображением (если этот бит равен 0). Оставшиеся биты `ref_pic_id` или `good_ref_pic_id[ i ]` зарезервированы для будущего применения организацией МСЭ-Т, должны быть равны 0 и не должны учитываться.

В зависимости от значения `payloadType`, применяется следующее.

- Если `payloadType` равно 0, то `ref_pic_id` или `good_ref_pic_id[ i ]` распознает изображение без обнаружения ошибки несовпадения битового потока. Переменная `picIdentifier` показывает значение `FrameNum` эталонного изображения (если эталонное изображение является кратковременным эталонным изображением) или `LongTermFrameIdx` (если эталонное изображение является долговременным эталонным изображением):
 - если изображение является кратковременным эталонным изображением, то значение `picIdentifier` должно быть меньше, чем `MaxFrameNum`;
 - иначе (изображение является долговременным эталонным изображением), то значение `picIdentifier` не должно быть больше, чем `MaxLongTermFrameIdx`.
- Иначе, если `payloadType` равно 1, то 16-ый бит должен быть равен 0, а переменная `picIdentifier` показывает значение `FrameNum` эталонного изображения. Значение `FrameNum` должно быть меньше, чем `MaxFrameNum`. Все изображения, поступающие в порядке декодирования, начинающиеся с изображения, у которого `FrameNum` равно `ref_pic_id & 0xFFFF`, вплоть до и включая изображение, у которого `FrameNum` равно $((\text{ref_pic_id} \& 0xFFFF) + \text{delta_ref_pic_id}) \% \text{MaxFrameNum}$, если такие изображения были представлены в битовом потоке видеоизображения, считаются полностью или частично потерянными.
- Иначе, если `payloadType` равно 2, то 16-ый бит должен быть равен 0, а переменная `picIdentifier` показывает значение `FrameNum` эталонного изображения, которое частично потеряно. Значение `FrameNum` должно быть меньше, чем `MaxFrameNum`. Каждый блок является макроблоком. Значение `data_partition_idc` показывает, что все данные (если `data_partition_idc` равно 0), раздел данных А (если `data_partition_idc` равно 1), раздел данных В (если `data_partition_idc` равно 2) или раздел данных С (если `data_partition_idc` равно 3) распознанных макроблоков потеряны. Все другие значения `data_partition_idc` зарезервированы для будущего применения организацией МСЭ-Т и не должны учитываться.
- Иначе, если `payloadType` равно 3 или 4, то 16 разрядов LSB элемента `ref_pic_id` показывают `FrameNum`, который равен `ref_pic_id & 0xFFFF`, эталонного изображения, для которого имеется CRC для одного или всех множеств значений параметра связанной последовательности или множеств значений параметра изображения. Значение `ref_pic_id & 0xFFFF` должно быть меньше, чем `MaxFrameNum`. Оставшиеся биты `ref_pic_id` зарезервированы для будущего применения организацией МСЭ-Т, должны быть равны 0, и не должны учитываться. Множество значений параметра с `param_set_type` равным 0 является множеством значений параметра последовательности блока NAL. Множество значений параметра с `param_set_type` равным 1 является множеством значений параметра изображения блока NAL. При использовании `forbidden_zero_bit` и `nal_ref_idc` в вычислении CRC множеств значений параметров последовательности или изображения блока NAL, они устанавливаются равными 0 и 3, соответственно.
- Иначе (`payloadType` равно 5), дальнейшая спецификация отсутствует.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты протокола Интернет и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи