

Internet Engineering Task Force (IETF)
Request for Comments: 5795
Obsoletes: 4995
Category: Standards Track
ISSN: 2070-1721

K. Sandlund
G. Pelletier
Ericsson
L-E. Jonsson
March 2010

Модель отказоустойчивой компрессии заголовков ROHC

The RObust Header Compression (ROHC) Framework

Аннотация

Протокол ROHC обеспечивает концепцию гибкого, эффективного и готового к будущему сжатию заголовков. Он разработан для обеспечения эффективности и надёжности при работе на каналах с различными технологиями и характеристиками.

Модель ROHC вместе с набором профилей сжатия была изначально предложена в RFC 3095. Для развития и упрощения спецификаций ROHC в этом документе явно разделены определения модели ROHC и профиля без компрессии. Приводимое здесь определение модели не меняет и не обновляет модели, описанной в RFC 3095.

Эта спецификация отменяет RFC 4995. Решена проблема взаимодействия, ошибочно внесённая в RFC 4995, и добавлены некоторые разъяснения.

Статус документа

Этот документ является проектом стандарта Internet (Internet Standards Track)..

Документ подготовлен IETF¹ и содержит согласованный взгляд сообщества IETF. Документ обсуждался публично и одобрен для публикации IESG². Дополнительная информация о стандартах Internet приведена в разделе 2 RFC 5741.

Информацию о текущем состоянии данного документа, обнаруженных ошибках и способах обратной связи можно найти по ссылке <http://www.rfc-editor.org/info/rfc5795>.

Авторские права

Авторские права ((с) 2010) принадлежат IETF Trust и лицам, являющимся авторами документа. Все права защищены.

К этому документу применимы права и ограничения, перечисленные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Документ может содержать материалы из IETF Document или IETF Contribution, опубликованных или публично доступных до 10 ноября 2008 года. Лица, контролирующие авторские права на некоторые из таких документов, могли не предоставить IETF Trust права разрешать внесение изменений в такие документы за рамками процессов IETF Standards. Без получения соответствующего разрешения от лиц, контролирующих авторские права этот документ не может быть изменён вне рамок процесса IETF Standards, не могут также создаваться производные документы за рамками процесса IETF Standards за исключением форматирования документа для публикации или перевода с английского языка на другие языки.

Оглавление

1. Введение.....	2
2. Уровни требований.....	3
2.1. Сокращения.....	3
2.2. Терминология ROHC.....	3
3. Основы (информационный раздел).....	4
3.1. Основы компрессии заголовков.....	4
3.2. Краткая история компрессии заголовков.....	4
4. Обзор ROHC (информационный раздел).....	4
4.1. Общие принципы.....	4
4.2. Эффективность, отказоустойчивость и прозрачность компрессии.....	5
4.3. Развитие протокола ROHC.....	5
4.4. Операционные характеристики канала ROHC.....	6
4.5. Компрессия и порядковый номер MSN.....	6
4.6. Статические и динамические части контекста.....	6
5. Модель ROHC (нормативный раздел).....	7
5.1. Канал ROHC.....	7
5.1.1. Контексты и их идентификаторы.....	7
5.1.2. Поканальные параметры.....	7
5.1.3. Продолжительность контекста декомпрессора.....	7
5.2. Пакеты ROHC и их типы.....	8

¹Internet Engineering Task Force.

²Internet Engineering Steering Group.

5.2.1. Общий формат пакетов ROHC.....	8
5.2.1.1. Формат октета заполнения.....	8
5.2.1.2. Формат октета Add-CID.....	8
5.2.1.3. Общий формат заголовка.....	8
5.2.2. Типы пакетов инициализации и обновления (IR).....	9
5.2.2.1. Формат заголовка ROHC IR.....	9
5.2.2.2. Формат заголовка ROHC IR-DYN.....	9
5.2.3. Изначальная обработка в декомпрессоре ROHC.....	10
5.2.4. Обратная связь ROHC.....	10
5.2.4.1. Формат обратной связи ROHC.....	11
5.2.5. Сегментация ROHC.....	11
5.2.5.1. Вопросы использования сегментации.....	11
5.2.5.2. Протокол сегментации.....	12
5.3. Общие методы кодирования.....	12
5.3.1. Контрольные суммы при компрессии заголовка.....	12
5.3.1.1. 8-битовая сумма CRC в заголовках IR и IR-DYN.....	12
5.3.1.2. 3-битовая сумма CRC в сжатых заголовках.....	12
5.3.1.3. 7-битовая контрольная сумма в сжатых заголовках.....	13
5.3.1.4. 32-битовая контрольная сумма сегментации.....	13
5.3.2. Самоописывающиеся значения переменного размера.....	13
5.4. ROHC UNCOMPRESSED - без компрессии (профиль 0x0000).....	13
5.4.1. Пакет IR.....	13
5.4.2. Нормальный пакет.....	14
5.4.3. Инициализация контекста.....	14
5.4.4. Работа декомпрессора.....	14
5.4.5. Обратная связь.....	15
6. Обзор профилей ROHC (информационный раздел).....	15
7. Благодарности.....	15
8. Согласование с IANA.....	15
9. Вопросы безопасности.....	16
10. Литература.....	16
10.1. Нормативные документы.....	16
10.2. Дополнительная литература.....	16
Приложение А. Алгоритм CRC.....	17

1. Введение

Для многих сетей снижение расходов при развёртывании и эксплуатации за счёт более эффективного использования пропускной способности является жизненно важным вопросом. Компрессия заголовков в каналах возможна благодаря тому, что часть содержащейся в заголовках пакетов информации не меняется для разных пакетов одного потока.

Для каналов, где используются заголовки IP, суммарный размер этих заголовков может быть достаточно большим. Приложения, передающие данные по протоколу RTP [RFC3550], в дополнение к кадрингованию канального уровня используют заголовок IPv4 [RFC0791] (20 октетов), заголовок UDP [RFC0768] (8 октетов) и заголовок RTP (12 октетов), что в сумме добавляет 40 октетов. При использовании IPv6 [RFC2460] заголовок IPv6 составит 40 октетов, а общий размер достигнет 60 октетов. Приложения, использующие протокол TCP [RFC0793], будут добавлять 20-октетный заголовок транспортного уровня и общий размер составит 40 октетов для IPv4 и 60 октетов для IPv6.

Относительный эффект компрессии для конкретных потоков (или приложений) зависит от используемого размера пакетов. Для приложений типа VoIP¹, где размер данных, служащих для передачи голоса, может быть достаточно мал (15-20 октетов), компрессия заголовков может давать существенный эффект. Относительная эффективность для потоков TCP с большими пакетами (например, копирование файлов) будет значительно меньше, чем для потоков с мелкими пакетами (например, сигнализация или организация сессий).

По мере распространения беспроводных технологий передачи трафика IP требуется реализация специальных мер, учитывающих параметры каналов, обеспечиваемых этими технологиями, в части сжатия заголовков. Традиционные схемы сжатия заголовков (такие, как [RFC2507] и [RFC2508]), недостаточно эффективны для каналов с существенным уровнем потерь пакетов и достаточно большим периодом кругового обхода (например, беспроводных каналов и туннелей IP).

Кроме того, схема компрессии должна обрабатывать зачастую нетривиальные ситуации с остаточными ошибками, когда нижележащий уровень передаёт декомпрессору пакеты с не обнаруженными битовыми ошибками. Следует также обрабатывать ситуации с потерей и нарушением порядка доставки пакетов до точки компрессии, а также на канале между точками сжатия и декомпрессии [RFC4224].

Протокол ROHC обеспечивает концепцию эффективной, гибкой и готовой к будущему компрессии заголовков. Он разработан для обеспечения эффективности и надёжности при работе на каналах с различными технологиями и характеристиками.

В RFC 3095 [RFC3095] определена схема ROHC и стартовый набор профилей сжатия. Для развития и упрощения спецификации модель и профили были разнесены в отдельные документы. Этот документ явно определяет модель ROHC, не меняя и не обновляя определения, данного в RFC 3095 (оба документа могут использоваться независимо один от другого). Реализации, основанные на определениях того и другого документа будут совместимы и смогут взаимодействовать. Однако, эту спецификацию следует рассматривать как замену RFC 3095 в качестве базовой спецификации для всех профилей, определяемых в будущем.

В этом документе исправления возникшая в RFC 4995 ошибка, приводящая к проблемам совместимости. Исправление приведено в параграфе 5.2.4.1 и связано с уточнением интерпретации поля Size в обратной связи ROHC.

¹Voice over IP - голосовая связь через IP.

2. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с [RFC2119].

2.1. Сокращения

В этом параграфе приведено большинство сокращений, используемых в документе.

ACK	Acknowledgment - подтверждение.
CID	Context Identifier - идентификатор контекста.
CO	Compressed Packet Format - формат сжатого пакета.
CRC	Cyclic Redundancy Check - контрольная сумма.
IR	Initialization and Refresh - инициализация и обновление.
IR-DYN	Initialization and Refresh, Dynamic part - инициализация и обновление, динамическая часть.
LSB	Least Significant Bit - наименее значимый (младший) бит.
MRRU	Maximum Reconstructed Reception Unit - максимальный восстанавливаемый на приеме блок.
MSB	Most Significant Bit - наиболее значимый (старший) бит.
MSN	Master Sequence Number - основной порядковый номер.
NACK	Negative Acknowledgment - негативное подтверждение.
ROHC	RObust Header Compression - сильная компрессия заголовков.

2.2. Терминология ROHC

Context - контекст

Контекст компрессора представляет собой состояние, используемое для сжатия заголовков, контекст декомпрессора - состояние, используемое для декомпрессии заголовков. Любое из этих состояний или оба вместе обычно будут называться контекстом, если не возникает неоднозначностей. Контекст содержит относящуюся к делу информацию из заголовков предыдущих пакетов потока (такую, как статические поля и возможные справочные значения для сжатия и декомпрессии). Кроме того, в контекст входит дополнительная информация, описывающая поток, например, сведения об изменении поведения полей (скажем, поведение поля IP Identifier или типовое увеличение порядковых номеров и временных меток от пакета к пакету).

Context damage - повреждение контекста

Когда контекст декомпрессора не согласуется с контекстом сжатия, при попытке восстановления исходного заголовка могут происходить отказы. Такая ситуация может возникать в результате некорректной инициализации контекста декомпрессора, а также при потере или повреждении пакетов между компрессором и декомпрессором. Когда пакеты не могут быть декомпрессированы по причине несоответствия контекстов, говорят о потере в результате повреждения контекста. О пакетах, которые декомпрессированы, но содержат ошибки вследствие несовместимости контекстов, говорят, как о повреждённых в результате повреждения контекста.

Context repair mechanisms - механизмы исправления контекста

Механизмы, используемые для ресинхронизации контекстов. Эти механизмы достаточно важны, поскольку повреждение контекста вызывает распространение потерь. Примерами могут служить механизмы на базе NACK и периодическое обновление важной информации из контекста, выполняемое обычно в односторонней операции. Имеются механизмы, способные снизить вероятность несоответствия контекстов (например, повторение одного типа информации во множестве пакетов и использование CRC для защиты данных обновления контекста).

CRC-8 validation - проверка CRC-8

Проверка целостности в плане битовых ошибок для принятого заголовка IR и IR-DYN с использованием 8-битовой контрольной суммы CRC, включённой в заголовок IR/IR-DYN.

CRC verification - проверка CRC

Проверка результата попытки декомпрессии с использованием 3-битовых или 7-битовых контрольных сумм CRC, включённых в сжатый заголовок пакета.

Damage propagation - распространение повреждений

Доставка некорректно декомпрессированных заголовков в результате повреждений контекста, например, ошибок (т. е., потери или повреждения) в предыдущих заголовках или обратной связи.

Error detection - детектирование ошибок

Обнаружение ошибок нижележащими уровнями. Если детектирование не полное, возникают остаточные ошибки.

Error propagation - распространение ошибок

Распространение повреждений или потерь.

ROHC profile - профиль ROHC

Протокол компрессии, определяющий механизмы сжатия для конкретных комбинаций заголовков. Профиль ROHC может быть адаптирован для работы с конкретным набором характеристик канала (например, уровень потерь, нарушение порядка доставки между точками компрессии и т. п.). Профили ROHC детализируют модель компрессии заголовков, определённую в этом документе, и с каждым профилем компрессии связан уникальный идентификатор профиля ROHC [ROHC-ids]. При организации канала ROHC согласуется набор профилей, поддерживаемых на обеих сторонах канала, и идентификатор профиля из согласованного набора используется для связывания каждого контекста сжатия с одним из профилей.

Link - канал

Физический путь передачи, образующий один интервал IP (hop).

Loss propagation - распространение потерь

Потеря заголовков в результате ошибок (например, потери или повреждения) в предыдущих пакетах или обратной связи.

Packet flow - поток пакетов

Последовательность пакетов, в которой значения полей и картина изменения этих значений таковы, что заголовки можно сжимать в одном контексте.

Residual error - остаточная ошибка

Ошибка, возникшая в процессе передачи и не обнаруженная на нижележащих уровнях.

ROHC channel - канал ROHC

Логический односторонний канал «точка-точка», передающий пакеты ROHC от одного компрессора к одному декомпрессору и опционально передающий данные обратной связи ROHC от имени другой пары компрессор-декомпрессор, работающей по отдельному каналу ROHC в противоположном направлении (см. также [RFC3759]).

В этом документе используется также концептуальная терминология из документа ROHC Terminology and Channel Mapping Examples [RFC3759].

3. Основы (информационный раздел)

В этом разделе приведена базовая информация по сжатию заголовков. Описаны фундаментальные идеи и рассмотрена история схем компрессии заголовков. Обсуждаются также мотивы разработки и обнаруженные недостатки разных схем, послуживших основой при создании модели и профилей ROHC [RFC3095].

3.1. Основы компрессии заголовков

Компрессия заголовков стала возможной, благодаря существенной избыточности в полях заголовков пакетов (в частности, последовательных пакетов одного потока). На пути от начальной точки до конечной нужна вся информация из заголовков каждого пакета в потоке, но на одном канале часть информации становится избыточной и её можно убрать перед отправкой и прозрачно восстановить на приёмной стороне канала. Размер заголовка может быть сокращён в первую очередь за счёт полей, которые предполагаются статичными (или почти статичными) в течение всего срока существования потока пакетов. Дополнительная компрессия обеспечивается за счёт сжатия полей с более динамичной информацией на основе методов, учитывающих поведение таких полей.

Для обеспечения компрессии и декомпрессии некоторая важная информация из предшествующих пакетов сохраняется в контексте. Компрессор и декомпрессор обновляют свои контексты при некоторых (не обязательно) синхронизированных) событиях. Рассинхронизация событий может приводить к несогласованности в контексте декомпрессора (т. е., повреждению контекста), способной привести к некорректной декомпрессии. Схема ROHC требует использования механизмов минимизации возможного повреждения контекста в комбинации с механизмами исправления контекста.

3.2. Краткая история компрессии заголовков

Первая схема компрессии заголовков STCP¹ [RFC1144] была предложена Van Jacobson. Схема STCP, которую называют также компрессией VJ, позволяет уменьшить 40-октетные заголовки TCP/IP до 4 октетов. STCP использует дельта-кодирование для последовательно изменяющихся полей. Компрессор STCP детектирует повторы передачи на транспортном уровне и передаёт в таких случаях заголовок, обновляющий весь контекст. Такой механизм исправления не требует явной сигнализации между компрессором и декомпрессором.

В общей схеме компрессии заголовков IP (IPHC²) [RFC2507] достигнуты некоторые улучшения по сравнению с STCP. IPHC позволяет сжимать любые заголовки IP, TCP и UDP. При сжатии заголовков, отличных от TCP, в IPHC не используется дельта-кодирование и это обеспечивает отказоустойчивость. Механизм исправления STCP дополнен здесь негативными подтверждениями (их называют сообщениями CONTEXT_STATE), которые повышают скорость исправления. Скорость работы этого механизма исправления ограничивается временем кругового обхода для канала. IPHC не сжимает заголовки RTP.

CRTP [RFC2508] является расширением IPHC для протокола RTP. CRTP сжимает 40 октетов заголовков IPv4/UDP/RTP до 2 октетов при отключённых контрольных суммах UDP. При использовании UDP минимальный размер заголовков CRTP составляет октета.

На каналах с потерями и большим периодом кругового обхода CRTP не обеспечивает достаточной эффективности [CRTP-eval]. Каждый потерянный в канале пакет приводит к отказу при декомпрессии нескольких последующих пакетов, поскольку контекст повреждается на время не менее одного периода кругового обхода с момента потери пакета. К сожалению, большие заголовки, передаваемые CRTP при обновлении контекста, приводят к дополнительному расходу полосы.

CRTP использует механизм локального исправления TWICE, который был предложен в IPHC. Имя TWICE происходит от того, что для регулярных сжатых пакетов корректное предсказание потери пакета между точками компрессии определяется двойным (twice) обновлением в текущем пакете. Хотя механизм TWICE значительно повышает производительность CRTP, в [CRTP-eval] отмечено, что даже при его использовании протокол CRTP удваивает число теряемых пакетов.

Улучшенный вариант протокола CRTP, названный eCRTP [RFC3545], повышает отказоустойчивость CRTP при наличии разупорядочивания и потери пакетов, сохраняя сам протокол CRTP почти без изменений. В результате eCRTP обеспечивает повышение уровня отказоустойчивости, но за счёт добавочных издержек несколько уступает протоколу CRTP по эффективности компрессии.

4. Обзор ROHC (информационный раздел)

4.1. Общие принципы

Как было отмечено выше, возможность поканальной компрессии заголовков обусловлена значительной избыточностью содержащейся в заголовках информации, что особенно существенно для последовательных пакетов, относящихся к одному потоку. Для использования этих свойств при компрессии заголовков следует рассмотреть несколько этапов.

Первый этап включает идентификацию и группировку пакетов в различные «потоки» с максимальной избыточностью в рамках каждого потока для обеспечения максимальной сжимаемости. Группировка пакетов в потоки обычно осуществляется по адресам хостов отправителя и получателя (IP), типу протокола транспортного уровня (например, UDP или TCP), номерам процессов (портов) и, возможно, дополнительным уникальным идентификаторам приложений

¹Compressed TCP.

²IP header compression.

типа источника синхронизации (SSRC¹) в RTP [RFC3550]. Компрессор и декомпрессор организуют контекст для потока пакетов и обозначают его идентификатором контекста CID² включаемым в сжатый заголовок.

Вторым этапом является понимание картины изменения разных полей заголовков. На верхнем уровне поля заголовков можно разделить на несколько классов.

INFERRED - производные

Эти поля содержат значения, которые можно вывести из других полей или внешних источников (например, размер кадра, содержащего пакет, часто можно получить от протокола канального уровня, что позволяет не передавать его при использовании компрессии).

STATIC - статический

Статические поля предполагаются неизменными в течение срока жизни потока. Значение такого поля достаточно передать один раз в начале потока.

STATIC-DEF - статические поля, определяющие принадлежность к потоку

Поля типа STATIC-DEF используются для определения потока, как отмечено выше. Пакеты, в которых соответствующие поля различаются, относят к разным потокам. В общем случае эти поля сжимаются, как STATIC.

STATIC-KNOWN - статические известные поля

Предполагается, что поля, относящиеся к типу STATIC-KNOWN, содержат общеизвестные значения, передавать которые, следовательно, не требуется.

CHANGING - меняющиеся

Предполагается, что такие поля могут содержать произвольные значения, принадлежащие к заданному диапазону или заданные иным способом. Изменяющиеся поля обычно приходится обрабатывать более изолированно, а для картины изменения значений таких полей требуется более детальная классификация.

Последним этапом является выбор методов кодирования различных полей в соответствии с их классификацией. Методы кодирования в комбинации с определенным поведением полей обеспечивают информацию для выбора формата сжатых заголовков. Анализ распределения вероятностей идентифицированной динамики полей обеспечивает пути оптимизации формата пакетов, когда более часто встречающиеся картины полей кодируются в более эффективных форматах.

Однако эффективность компрессии требует компромисса с двумя другими свойствами - устойчивостью кодирования к потерям и ошибкам между компрессором и декомпрессором, а также возможностью обнаружения и корректировки ошибок в процессе декомпрессии.

4.2. Эффективность, отказоустойчивость и прозрачность компрессии

Производительность протокола компрессии заголовков можно описать тремя параметрами - эффективностью, отказоустойчивостью и прозрачностью компрессии.

Эффективность сжатия

Эффективность сжатия определяется средним снижением размера заголовков в результате применения протокола.

Отказоустойчивость

Отказоустойчивость протокола характеризует устойчивость к потере пакетов, остаточным битовым ошибкам и нарушению порядка доставки в канале, на котором используется сжатие заголовков, без потери дополнительных пакетов или внесения дополнительных ошибок при декомпрессии заголовков.

Прозрачность сжатия

Прозрачность сжатия определяется степенью поддержки семантики исходных заголовков. Если все поля после декомпрессии побитово совпадают с исходными полями, схема компрессии обеспечивает прозрачность.

4.3. Развитие протокола ROHC

При разработке протокола компрессии заголовков требуется найти компромисс между эффективностью сжатия и отказоустойчивостью при сохранении прозрачности, поскольку повышение отказоустойчивости всегда ведёт к снижению эффективности сжатия и наоборот. Схема должна быть достаточно гибкой и минимизировать влияние таких факторов, как изменение времени кругового обхода и доли теряемых пакетов для каналов, на которых используется сжатие заголовков.

Для достижения требуемого компромисса схема компрессии должна обеспечивать декомпрессору средства проверки декомпрессии и детектирования потенциального повреждения контекста, а также механизмы восстановления (такие, как обратная связь). Схема компрессии заголовков, разработанные до создания рабочей группы ROHC, создавались без должного учёта этих требований.

Рабочая группа ROHC разработала решения для компрессии заголовков с учётом требований современных и будущих технологий. Хотя при разработке уделялось особое внимание требованиям, основанным на характеристиках беспроводных каналов, результаты применимы и к многим другим технологиям канального уровня.

Документ RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed [RFC3095] был опубликован в 2001 году, как первый результат ROHC WG. Модель ROHC является обобщённой и расширяемой схемой компрессии заголовков, на основе которой могут определяться профили сжатия для различных протоколов. В RFC 3095 предложено множество новых методов компрессии, показавших соответствие предъявляемым требованиям, как описано в [RFC3096].

Тестирование взаимодействия RFC 3095 подтверждает возможности ROHC в части заявленных целей, но сообщения от разработчиков показали, что спецификация чересчур сложна и не всегда понятна. Один из основных недостатков заключался в том, что базовую модель и профили было достаточно сложно разделить в [RFC3095], что также осложняет разработку дополнительных профилей. Поэтому цель данного документа заключается в явной спецификации модели ROHC, а пересмотренные профили сжатия RFC 3095 приведены в отдельном документе [RFC5225].

¹Synchronization source.

²Context Identifier.

4.4. Операционные характеристики канала ROHC

Отказоустойчивая компрессия заголовков может использоваться со многими технологиями канального уровня. Модель ROHC обеспечивает гибкость профилей для решения задач различных приложений и в этом параграфе кратко рассмотрены некоторые операционные характеристики каналов ROHC (см. также [RFC3759]).

Мультиплексирование через один логический канал

Канал ROHC обеспечивает механизм идентификации контекста в базовом формате пакетов ROHC. CID обеспечивает логическому каналу возможность использования ROHC для доставки множества потоков с компрессией заголовков, сохраняя возможность использования канала для одного потока пакетов без использования CID. Говоря точнее, ROHC использует отдельное пространство CID для логического канала и поле CID может быть опущено при использовании канала ROHC для одного потока.

Организация канальных параметров

Канальный уровень, определяющий поддержку для каналов ROHC, должен обеспечивать способ организации параметров канала с компрессией заголовков (см. параграф 5.1). Это может быть достигнуто с помощью механизмов согласования, статического предоставления или сигнализации по отдельному каналу.

Идентификация типа пакетов

Канал ROHC определяет пространство идентификаторов типа пакетов и устанавливает ограничения на использование идентификаторов, которые являются общими для всех профилей ROHC. Идентификаторы, которые не имеют ограничений (т. е., не определённые в данном документе), доступны для всех профилей. Идентификатор является частью каждого сжатого заголовка и обеспечивает каналу, используемому для поддержки канала ROHC возможность использования одного типа данных канального уровня для ROHC.

Нарушение порядка доставки пакетов между конечными точками компрессии

Каждый профиль определяет свой уровень отказоустойчивости, включая устойчивость к нарушению порядка доставки до точек компрессии и, особенно, между ними.

Для профилей, заданных в [RFC3095], требуется, чтобы канал между компрессором и декомпрессором поддерживал упорядоченную доставку (т. е., в определениях этих профилей предполагается, что декомпрессор всегда получает пакеты в том порядке, в котором их передал компрессор). Влияние нарушения порядка на производительность этих профилей описано в [RFC4224]. Однако нарушение порядка доставки до точки компрессии обрабатывается этими профилями (т. е., в них не делается допущения о том, что компрессор получает все пакеты с сохранением порядка).

Профили ROHCv2, заданные в [RFC5225], предполагают, что декомпрессор может получать пакеты с нарушением порядка (т. е., не в том порядке, в котором они были переданы компрессором). Нарушение порядка доставки в точку компрессии также учитывается в этих профилях.

Дублирование пакетов

От канала, поддерживающего ROHC, требуется отсутствие дублирования пакетов (однако, такое дублирование может возникать до точки компрессии, поэтому предположений о получении компрессором единственного экземпляра каждого пакета не делается).

Кадрование

Канальный уровень должен обеспечивать кадрование, которое позволяет различать границы кадров и отдельные кадры.

Детектирование ошибок и защита от них

Профили ROHC следует разрабатывать так, чтобы они справлялись с остаточными ошибками в заголовках, доставляемых декомпрессору. Для детектирования отказов при декомпрессии и предотвращения или снижения уровня распространения повреждений используются контрольные суммы CRC. Однако рекомендуется реализовать детектирование ошибок на нижележащих уровнях для заголовков ROHC и предотвращение доставки заголовков ROHC с большим числом остаточных ошибок.

4.5. Компрессия и порядковый номер MSN

Компрессия полей заголовков основывается на организации функции от номера пакета (MSN¹), которая описывает картину изменения полей по мере изменения MSN.

Картина изменения включает, например, поля с монотонным изменением, поля с незначительными изменениями, поля с редкими изменениями, а также поля, значения которых не меняются в течение срока жизни потока пакетов (в этом случае функция MSN эквивалентна константе).

Компрессор сначала организует функции для каждого типа полей, а потом - гарантированное взаимодействие с MSN. Когда картина поля не соответствует организованной функции (т. е., существующая функция даёт результат, отличающийся от поля в сжимаемом заголовке) может быть передана дополнительная информация для обновления параметров этой функции.

MSN определяется на уровне профиля. Значение может извлекаться непосредственно из какого-либо поля сжимаемого заголовка (например, RTP SN [RFC5225]) или создаваться и поддерживаться самим компрессором (например, MSN для сжатия UDP в профиле 0x0102 [RFC5225] или MSN в ROHC-TCP [RFC4996]).

4.6. Статические и динамические части контекста

Контекст компрессии можно концептуально разделить на две части - статическую и динамическую, - каждая из которых основывается на свойствах сжимаемых полей.

Статическая часть включает информацию, требуемую для сжатия и декомпрессии полей класса STATIC, STATIC-KNOWN или STATIC-DEF (см. параграф 4.1).

Динамическая часть включает состояния, поддерживаемые для всех прочих полей (т. е. полей класса CHANGING).

¹Master sequence number - основной порядковый номер.

5. Модель ROHC (нормативный раздел)

В этом разделе даются нормативные определения частей, общих для всех профилей ROHC (т. е. для модели в целом). Модель задаёт требования и функциональность канала ROHC включая способы обслуживания множества потоков сжатых пакетов в одном канале.

В заключение этот раздел задаёт методы кодирования, используемые в форматах пакетов, которые являются общими для всех профилей. Эти методы кодирования могут использоваться в спецификациях профилей для кодирования полей в определяемых профилем частях без их повторного определения.

5.1. Канал ROHC

5.1.1. Контексты и их идентификаторы

С каждым сжатым потоком связывается контекст, представляющий собой состояние компрессора и декомпрессора, поддерживаемое для корректного сжатия и декомпрессии заголовков в пакетах из данного потока. Для идентификации контекста служит специальное значение CID.

Контекст считается новым, когда CID связывается с профилем в первый раз с момента создания канала ROHC или CID связывается с получением IR (не IR-DYN) для профиля, отличающегося от профиля контекста.

Данные контекста концептуально хранятся в таблице. Таблица контекстов индексируется по значениям CID, передаваемым со сжатыми заголовками и данными обратной связи.

Пространство CID может быть небольшим (значения от 0 до 15) или большим (от 0 до $2^{14} - 1 = 16383$). Решение о размере пространства CID **должно** приниматься (возможно, путём согласования) до того, как можно будет передать первый пакет через канал ROHC.

Пространство CID организуется независимо для каждого канала. Например, CID 3 для канала А и CID 3 для канала В не указывают на один контекст даже в тех случаях, когда конечные точки каналов А и В совпадают. В частности значения CID для любой пары каналов ROHC не связаны между собой (двум связанным каналам ROHC, каждый из которых используется в качестве канала обратной связи для другого, даже не требуется использовать пространства CID одного размера).

5.1.2. Поканальные параметры

Канал ROHC имеет множество параметров, которые формируют часть организованного состояния канала и состояния для каждого контекста. Состояние канала ROHC **должно** организовываться до того, как может быть передан первый пакет ROHC, что может быть реализовано за счёт использования протоколов согласования, обеспечиваемых канальным уровнем (см. также [RFC3241], где описана опция согласования параметров ROHC для PPP). В этом параграфе приведено абстрактное описание отдельных компонент информации о состоянии канала.

LARGE_CIDS

Логическая переменная. Значение false говорит о малом пространстве CID (0 октетов или один префиксный октет, значения CID от 0 до 15), значение true - о большом пространстве CID (1 или 2 октета CID, значения CID от 0 до 16383). См. также параграфы 5.1.1 и 5.2.1.3.

MAX_CID

Неотрицательное целое число. Максимальный номер CID, который будет использоваться компрессором (отметим, что этот параметр не связан с LARGE_CIDS, но в дальнейшем может быть ограничен этим параметром). Это значение представляет согласие декомпрессора на выделение ресурсов памяти, достаточных для обслуживания по крайней мере MAX_CID+1 контекстов; декомпрессор **должен** поддерживать организованные контексты в этом пространстве, пока значение CID не будет использовано снова путём организации нового контекста или канала не прекратит существование.

PROFILES

Множество неотрицательных целых чисел, каждое из которых указывает профиль, поддерживаемый компрессором и декомпрессором. Профили идентифицируются 16-битовыми значениями, в которых 8 младших битов указывают реальный профиль, а 8 старших - вариант этого профиля. Формат сжатого заголовка ROHC идентифицирует используемый профиль только 8 младшими битами LSB - это означает, что при наличии у профиля вариантов во множество PROFILES после согласования **недопустимо** включать более одного варианта данного профиля. Компрессору **недопустимо** сжимать заголовки с использованием профилей, не включённых в PROFILES.

FEEDBACK_FOR

Необязательная ссылка на канал ROHC в обратном направлении между теми же конечными точками. При наличии этого параметра он указывает в какой канал ROHC передаётся вся обратная связь для данного канала ROHC (см. [RFC3759]).

MRRU¹

Неотрицательное целое число. Показывает максимальный размер (в октетах) восстанавливаемого блока, который декомпрессор может собрать из сегментов (см. параграф 5.2.5). Это значение включает контрольную сумму сегментации. Если согласовано MRRU = 0, сегментацию **недопустимо** использовать на канале, а полученные сегменты **должны** отбрасываться декомпрессором.

5.1.3. Продолжительность контекста декомпрессора

Как часть согласования параметров канала, компрессор и декомпрессор с помощью параметра MAX_CID договариваются о максимальном используемом идентификаторе контекста (CID). Соглашаясь с MAX_CID, декомпрессор принимает на себя обязательство по выделению памяти для поддержки не менее MAX_CID+1 контекстов и контексты со значениями CID в этом согласованном пространстве декомпрессору **следует** сохранять, пока значение CID не будет использовано заново или пока канал не будет разорван или согласован заново.

¹Maximum Reconstructed Reception Unit - максимальный восстанавливаемый блок для приёма.

5.2. Пакеты RONS и их типы

В этом параграфе для диаграмм, показывающих различные типы пакетов, форматов и полей RONS, двоеточие (:) указывает необязательную часть, а символ дробной черты (/) - переменный размер.

Схема индикации типа пакета RONS разработана так, чтобы обеспечивалось необязательное поле заполнения, поле типа пакетов обратной связи, необязательный октет Add-CID (включает 4 бита CID) и простой механизм сегментации-сборки.

На уровне модели RONS зарезервировано несколько типов пакетов:

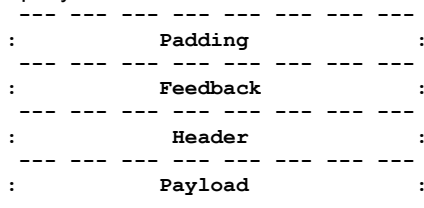
```
11100000 : заполнение
1110nnnn : октет Add-CID (nnnn=CID со значениями от 0x1 до 0xF)
11110    : обратная связь
11111000 : пакет IR-DYN
1111110  : пакет IR
1111111  : сегмент
```

В отдельных профилях могут определяться и использоваться другие типы пакетов:

```
0      : доступно (не резервируется моделью RONS)
10     : доступно (не резервируется моделью RONS)
110    : доступно (не резервируется моделью RONS)
1111101 : доступно (не резервируется моделью RONS)
11111001 : доступно (не резервируется моделью RONS)
```

5.2.1. Общий формат пакетов RONS

Базовый формат пакета RONS показан на рисунке.



Padding - заполнение

Произвольное (0 или больше) число октетов заполнения, формат которых определен в параграфе 5.2.1.1.

Feedback - обратная связь

Произвольное (0 или больше) число элементов обратной связи, формат которых определен в параграфе 5.2.4.1.

Header - заголовок

Специфический для профиля заголовок CO (см. параграф 5.2.1.3), заголовок IR или IR-DYN (см. параграф 5.2.2) или сегмент RONS (см. параграф 5.2.5). В пакете RONS не может быть более одного заголовка, который может отсутствовать, если пакет содержит только обратную связь (поле Feedback).

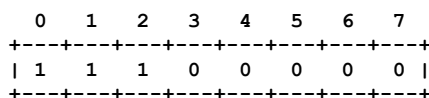
Payload - данные

Соответствует некоторому (0 или больше) числу октетов данных из несжатого пакета, начиная с первого октета в несжатом пакете вслед за последним заголовком, сжимаемым в этом профиле компрессии.

В пакете **должно** присутствовать по крайней мере одно поле Feedback или Header.

5.2.1.1. Формат октета заполнения

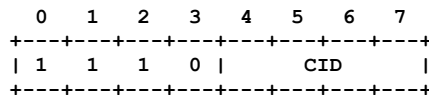
Октет заполнения показан на рисунке.



Отметим, что октет Padding **недопустимо** интерпретировать, как октет Add-CID для CID 0.

5.2.1.2. Формат октета Add-CID

Октет Add-CID показан на рисунке.



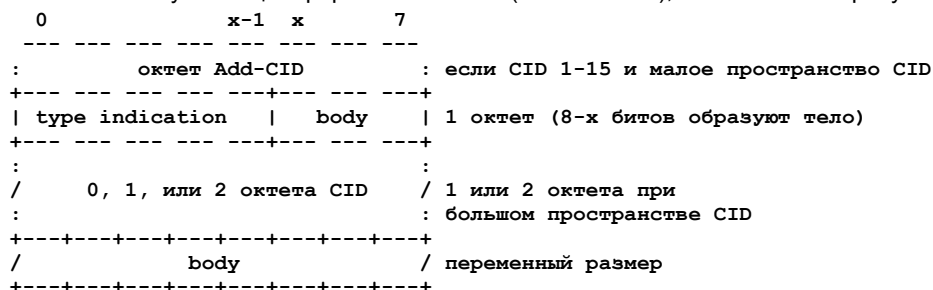
CID

Значения от 0x1 до 0xF указывают CID от 1 до 15.

Отметим, что для CID 0 октет Add-CID совпадает с октетом заполнения (Padding).

5.2.1.3. Общий формат заголовка

Все типы пакетов RONS используют общий формат заголовка (поле Header), показанный на рисунке.



type indication

Тип пакета ROHC.

body

Интерпретируется в зависимости от типа пакета и данных CID в соответствии с определением профиля.

Заголовок начинается с индикации типа пакета или включает индикации, следующую сразу после октета Add-CID.

Когда канал ROHC настроен на использование малого пространства CID:

- если поле Add-CID непосредственно предшествует индикации типа пакета, пакет имеет значение CID = Add-CID, в противном случае CID = 0;
- CID = 0 из малого пространства представляется без использования каких-либо битов (0), следовательно, для потоков с CID 0 не возникает издержек на передачу CID в сжатом заголовке; в этом случае поле Header начинается с индикации типа пакета;
- CID из малого пространства со значениями от 1 до 15 представляются октетом Add-CID, как описано выше; поле Header начинается с октета Add-CID, за которым следует индикация типа пакета;
- CID из большого пространства не включаются в поле Header.

Если канал ROHC настроен на использование большого пространства CID:

- большое значение CID всегда указывается с использованием схемы кодирования, описанной в параграфе 5.3.2 (не более 2 октетов); в таких случаях поле Header начинается с индикации типа пакета.

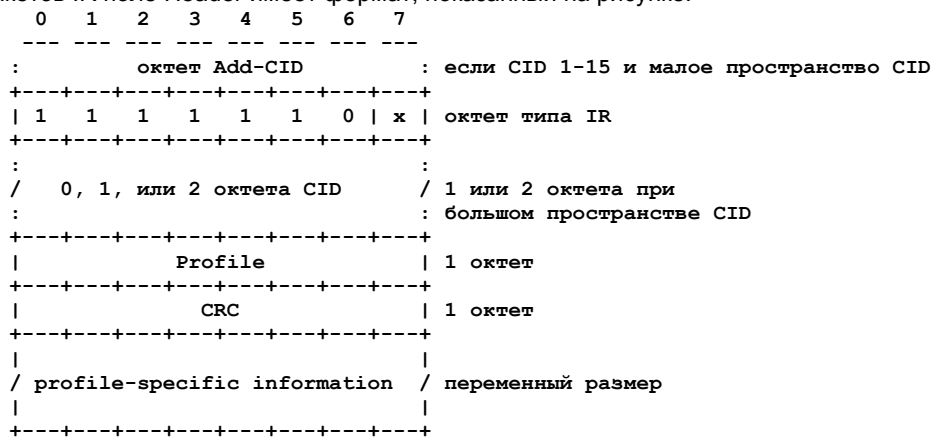
5.2.2. Типы пакетов инициализации и обновления (IR)

Тип пакетов IR¹ содержит идентификатор профиля, определяющий интерпретацию остальной части заголовка и связывающий профиль с контекстом. Сохранённый параметр профиля в дальнейшем определяет синтаксис и семантику идентификаторов типа пакетов и типы, используемые в конкретном контексте.

Пакеты IR и IR-DYN всегда обновляют контекст в части содержащихся в заголовке полей обновления. Они никогда не сбрасывают контекст, за исключением случаев инициализации нового контекста (см. параграф 5.1.1) и ситуаций, когда профиль, указанный полем Profile задаёт иное.

5.2.2.1. Формат заголовка ROHC IR

Заголовок IR связывает CID с профилем и обычно также инициализирует контекст, а также обновляет все или часть контекстов. Для пакетов IR поле Header имеет формат, показанный на рисунке.



x

Зависящая от профиля информация, интерпретируемая в соответствии с профилем, указанным в поле Profile заголовка IR.

Profile

Профиль, связанный с CID. В заголовке IR профиль представляется 8 младшими битами идентификатора (см. параграф 5.1.2).

CRC

8-битовое значение контрольной суммы (см. параграф 5.3.1.1).

Profile-specific information

Содержимое этой части заголовка IR определяется конкретным профилем и интерпретируется в соответствии с профилем, указанным в поле Profile.

5.2.2.2. Формат заголовка ROHC IR-DYN

В отличие от заголовка IR заголовок IR-DYN не может инициализировать неинициализированный контекст. Однако он может переопределить связанный с контекстом профиль, если указанный в заголовке IR-DYN профиль позволяет это. Этот тип пакетов также резервируется на уровне модели. Заголовок IR-DYN обычно также инициализирует или обновляет части контекста. Формат заголовка IR-DYN показан на рисунке.

Profile

Профиль, связанный с CID. Задаётся так же, как для пакетов IR.

CRC

8-битовое значение контрольной суммы (см. параграф 5.3.1.1).

Profile-specific information

Содержимое этой части заголовка IR-DYN определяется конкретным профилем и интерпретируется в соответствии с профилем, указанным в поле Profile.

¹Initialization and Refresh.

0	1	2	3	4	5	6	7	

: октет Add-CID								: если CID 1-15 и малое пространство CID
+---+---+---+---+---+---+---+---+								
1	1	1	1	1	0	0	0	октет типа IR-DYN
+---+---+---+---+---+---+---+---+								
:								:
/ 0-2 октета CID								/ 1 или 2 октета при
:								: большом пространстве CID
+---+---+---+---+---+---+---+---+								
Profile							1 октет	
+---+---+---+---+---+---+---+---+								
CRC							1 октет	
+---+---+---+---+---+---+---+---+								
/ profile-specific information								/ переменный размер
+---+---+---+---+---+---+---+---+								

5.2.3. Изначальная обработка в декомпрессоре RONS

Изначально контекст не имеет какого-либо состояния. Таким образом, ни какие пакеты, указывающие на неинициализированный контекст, за исключением пакетов, содержащих достаточную информацию в статических полях, не могут быть декомпрессированы.

При получении декомпрессором пакета типа IR указанный в нем профиль определяет обработку.

- Если проверка целостности заголовка с помощью 8-битового поля CRC приводит к отказу, пакет **недопустимо** декомпрессировать и доставлять на вышележащие уровни. Если в контексте указан профиль, логика этого профиля определяет передачу данных обратной связи, если они имеются. Если профиль в контексте не отмечен, логика передачи данных обратной связи (при их наличии) определяется реализацией. Однако может оказаться разумным отказ от каких-либо дальнейших действий, поскольку причиной отказа может быть любая часть заголовка IR, покрываемая контрольной суммой.

При получении декомпрессором пакета IR-DYN указанный в нем профиль определяет обработку данного пакета.

- Если проверка целостности заголовка с помощью 8-битового поля CRC приводит к отказу, пакет **недопустимо** декомпрессировать и доставлять на вышележащие уровни. Если в контексте указан профиль, логика этого профиля определяет передачу данных обратной связи, если они имеются. Если профиль в контексте не отмечен, логика передачи данных обратной связи (при их наличии) определяется реализацией. Однако может оказаться разумным отказ от каких-либо дальнейших действий, поскольку причиной отказа может быть любая часть заголовка IR-DYN, покрываемая контрольной суммой.
- Если контекст ещё не был инициализирован, пакет **недопустимо** декомпрессировать и доставлять на вышележащие уровни. При наличии данных обратной связи логику действий для них определяет профиль, указанный в заголовке IR-DYN (при совпадении 8-битовой контрольной суммы CRC).

При возникновении ошибок разбора пакетов любого типа декомпрессор **должен** отбрасывать такие пакеты без дальнейшей обработки. Например, поле CID присутствует в заголовке пакета при указанном использовании большого пространства CID для канала RONS и поле представлено с использованием самопишущего кодирования с переменным размером (см. параграф 5.3.2); если поле начинается с 110 или 111 это будет вызывать ошибку при разборе, поскольку такое поле не может быть представлено с размером, превышающим 2 октета.

Для профилей **рекомендуется** запрещать декомпрессору предпринимать попытки декомпрессии для пакетов, содержащих лишь 3-битовую контрольную сумму CRC после полной или частичной инвалидации динамического контекста, пока не будет получен, декомпрессирован и проверен по 7- или 8-битовой контрольной суммы CRC пакет с достаточной информацией о динамических полях.

5.2.4. Обратная связь RONS

Обратная связь обеспечивает компрессору информацию от декомпрессора. Для обратной связи может использоваться любой канал RONS, работающий в нужном направлении.

Общий формат пакетов RONS позволяет передавать данные обратной связи с использованием «размывания» (interspersed), «наложения» (piggybacking) (см. [RFC3759]) или их комбинации по любому каналу RONS. Это обеспечивается за счёт перечисленных ниже свойств

Reserved packet type - зарезервированный тип пакетов

Тип пакетов для обратной связи зарезервирован на уровне модели. В пакетах можно передавать данные переменного размера.

CID information - информация CID

Данные обратной связи, передаваемые по отдельному каналу, направляются компрессору и интерпретируются им. Для этого в каждом пакете обратной связи содержится информация CID из канала, к которому относится обратная связь. Схема обратной связи в RONS требует, чтобы канал обратной связи доставлял данные не более, чем одному компрессору. Связь компрессора с обратной связью для конкретного канала выходит за пределы данной спецификации (см. [RFC3759]).

Length information - информация о размере

Размер элемента обратной связи можно определить путем проверки нескольких первых октетов данных. Это позволяет добавлять данные обратной связи «в нагрузку» или объединять несколько элементов обратной связи в один пакет. Информация о размере, таким образом, отвязывает декомпрессор от связанного с ним компрессора на той же стороне, поскольку декомпрессор может извлечь данные обратной связи из сжатого заголовка без разбора его содержимого и передать найденную информацию.

Связь между парами компрессоров-декомпрессоров, работающих в противоположном направлении, для передачи данных обратной связи **следует** поддерживать в течение всего срока жизни канала RONS. Если это не удаётся,

рекомендуется уведомлять компрессор о недоступности канала обратной связи - компрессору в таких случаях **следует** перезапустить компрессию путем создания нового контекста для каждого потока пакетов, а также **следует** использовать значение CID, которое раньше не было связано с профилем, используемым для компрессии потока.

5.2.4.1. Формат обратной связи ROHC

ROHC определяет три разных категории сообщений обратной связи - подтверждение (ACK), негативное подтверждение (NACK) и NACK для контекста в целом (STATIC-NACK). Для профилей могут определяться другие типы сообщений.

ACK

Подтверждает успешную декомпрессию пакета и показывает, что декомпрессор считает свой контекст корректным.

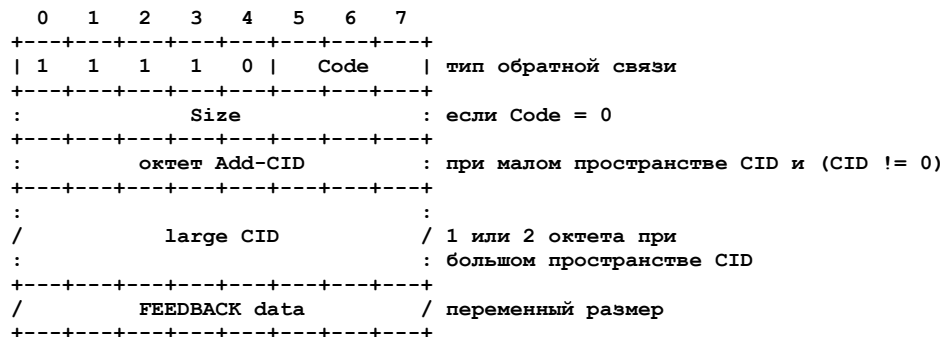
NACK

Показывает, что декомпрессор считается динамическую часть контекста или её отдельные компоненты некорректными.

STATIC-NACK

Показывает, что декомпрессор считает некорректным статический контекст или этот контекст отсутствует.

Обратная связь передаётся в канал ROHC в виде одного или нескольких (конкатенация) элементов, формат которых показан на рисунке.



Code

Значение 0 указывает на присутствие октета Size, 1-7 показывает общий размер полей FEEDBACK data и CID (если оно имеется) в октетах.

Size

Показывает общий размер полей FEEDBACK data и CID (если оно имеется) в октетах.

FEEDBACK data

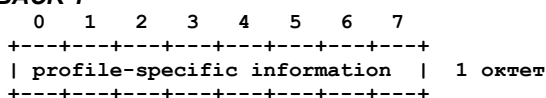
FEEDBACK-1 или FEEDBACK-2 (см. ниже).

Информация CID в элементах обратной связи показывает контекст, для которого предназначена обратная связь. Параметр LARGE_CIDS, контролирующей использование больших значений CID, берётся из состояния канала принимающего компрессора, а не из состояния канала, служащего для обратной связи.

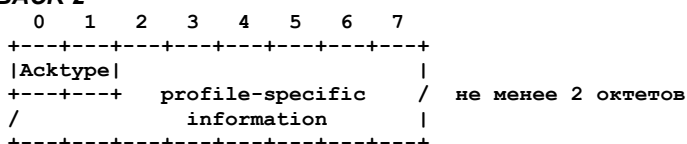
Поле большого CID (при его наличии) кодируется, как описано в параграфе 5.3.2 и для его представления **недопустимо** использовать более 2 октетов.

Поле FEEDBACK может содержать данных в любом из двух показанных ниже форматов.

FEEDBACK-1



FEEDBACK-2



Acktype: 0 = ACK

1 = NACK

2 = STATIC-NACK

3 резерв (использование **недопустимо**; приводит к невозможности разбора).

5.2.5. Сегментация ROHC

ROHC определяет простой протокол сегментации. Компрессор может выполнить сегментацию, например, для передачи пакетов, размер которых превосходит заданное для канала значение.

5.2.5.1. Вопросы использования сегментации

Протокол сегментации ROHC не слишком эффективен. Он не предназначен для замены функций сегментации канального уровня, которые по возможности **следует** использовать для повышения эффективности.

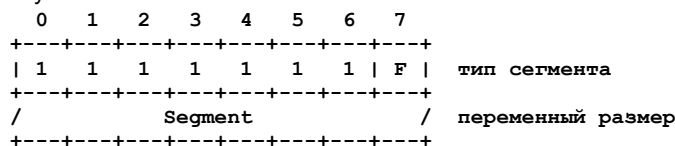
Протокол сегментации ROHC был разработан в предположении сохранения порядка доставки пакетов между компрессором и декомпрессором, использования для детектирования ошибок только контрольной суммы CRC и отсутствия порядковых номеров. Если нет гарантии упорядоченной доставки, использовать сегментацию ROHC **недопустимо**.

Протокол сегментации также предполагает, что все сегменты пакета ROHC, относящиеся к одному контексту, принимаются без конфликтов с другими пакетами ROHC в том же канале, включая все пакеты ROHC, относящиеся к другим контекстам. С учётом этого допущения в сегменты не включается информация CID и они, следовательно, не могут быть связаны с соответствующим контекстом, пока не будут получены все сегменты и собран весь блок.

5.2.5.2. Протокол сегментации

Сегментация ROHC применяется к комбинации полей Header и Payload пакетов ROHC, как определено в параграфе 5.2.1.

Формат сегмента показан на рисунке.

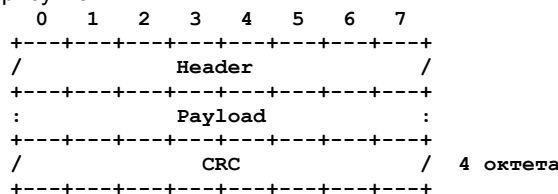


F

Финальный бит, указывающий последний сегмент восстанавливаемого блока. Октету типа сегмента могут предшествовать поля Padding и/или Feedback. Для сегментов не указывается CID, но информация CID включается в восстановленный блок. В этот блок **недопустимо** включать заполнение, сегменты или обратную связь.

При получении заключительного сегмента декомпрессор собирает полученный в пакете сегмент вместе с остальными сегментами, которые непосредственно предшествовали заключительному сегменту в один блок в порядке приёма сегментов. Все сегменты одного восстанавливаемого блока принимаются один за другим с сохранением порядка доставки пакетов декомпрессору. Если за сегментом, который не является финальным, следует несегментированный пакет ROHC, сборка текущего блока прерывается и декомпрессор **должен** отбросить все полученные в данном канале сегменты этого блока.

Восстановленный блок показан на рисунке.



Header

См. параграф 5.2.1

Payload

См. параграф 5.2.1

CRC

32-битовое значение CRC вычисляется с использованием полинома, приведённого в параграфе 5.3.1.4.

Если размер восстановленного блока не превышает 4 октета, проверка CRC завершается отказом или размер блока превышает значение параметра MRRU (см. параграф 5.1.2) для канала, восстановленный блок **должен** быть отброшен декомпрессором. При совпадении CRC восстановленный блок может быть подвергнут дальнейшей обработке.

5.3. Общие методы кодирования

5.3.1. Контрольные суммы при компрессии заголовка

В этом разделе описан расчёт контрольных сумм CRC, используемых ROHC. Для всех значений CRC используется алгоритм, описанный в [RFC1662] и определённый в Приложении А к данному документу, с полиномами, заданными в последующих параграфах.

5.3.1.1. 8-битовая сумма CRC в заголовках IR и IR-DYN

Покрытие 8-битовой контрольной суммы в заголовках IR и IR-DYN зависит от профиля, но **должно** включать, по крайней мере, начальную часть заголовка, завершающуюся полем Profile и включающую CID или октет Add-CID. Обратная связь и заполнение не относятся к заголовку Header (параграф 5.2.1) и, таким образом, не включаются в расчёт CRC. В спецификации профилей **следует** также включать в покрытие CRC все прочую информацию, используемую для инициализации контекста декомпрессора.

Говоря более точно, 8-битовое значение CRC не покрывает полностью исходный несжатый заголовок и, следовательно, не может обеспечить декомпрессору способа проверки результата декомпрессии или корректности контекста декомпрессии в целом. Однако применение контрольных сумм обеспечивает декомпрессору достаточную отказоустойчивость для обновления своего контекста с помощью информации из заголовков IR или IR-DYN.

Полином CRC для 8-битовой контрольной суммы имеет вид

$$C(x) = 1 + x + x^2 + x^8$$

При расчёте контрольной суммы значение самого поля CRC в заголовке принимается нулевым, а в качестве начального содержимого регистра CRC устанавливаются все 1.

5.3.1.2. 3-битовая сумма CRC в сжатых заголовках

3-битовая контрольная сумма заголовка рассчитывается для всех октетов исходного заголовка до его компрессии.

В качестве начального содержимого регистра CRC устанавливаются все 1.

Для расчёта контрольной суммы используется полином

$$C(x) = 1 + x + x^3$$

3-битовое значение CRC позволяет декомпрессору проверять результат декомпрессии для небольших сжатых заголовков и обнаруживать повреждения контекста на основе агрегированной вероятности для множества попыток декомпрессии. Однако эти суммы слишком слабы даже для того, чтобы гарантировать отсутствие ошибок в одном заголовке. По этой причине 3-битового значения CRC в сжатом заголовке обычно недостаточно для исправления контекста декомпрессора. Следовательно, профилям рекомендуется воздерживаться от декомпрессии таких заголовков, когда контекст декомпрессора или его часть предполагается некорректным.

5.3.1.3. 7-битовая контрольная сумма в сжатых заголовках

7-битовая контрольная сумма заголовка рассчитывается для всех октетов исходного заголовка до его компрессии .

В качестве начального содержимого регистра CRC устанавливаются все 1.

Для расчёта контрольной суммы используется полином

$$C(x) = 1 + x + x^2 + x^3 + x^6 + x^7$$

7-битовые значения CRC предназначены для обеспечения декомпрессору возможности проверки результата декомпрессии для больших заголовков и восстановления контекста декомпрессора. 7-битовых значений CRC достаточно для того, чтобы предполагать успешное исправление контекста при декомпрессии одного заголовка. Следовательно, профили могут разрешать декомпрессию заголовка с 7-битовой контрольной суммой даже в предположении повреждения контекста декомпрессора.

5.3.1.4. 32-битовая контрольная сумма сегментации

32-битовые контрольные суммы используются схемой сегментации для проверки восстановленных блоков и, поэтому, рассчитываются для сегментируемого блока, т. е., включают поля Header и Payload пакетов ROHC.

В качестве начального содержимого регистра CRC устанавливаются все 1.

Для расчёта контрольной суммы используется полином

$$C(x) = x^0 + x^1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$$

Назначением 32-битовых контрольных сумм CRC является проверка восстановленных блоков.

5.3.2. Самоописывающиеся значения переменного размера

Значения многих полей и параметров декомпрессии могут меняться в широких пределах. Для оптимизации передачи таких значений для их кодирования применяется переменное число октетов, которое указывается несколькими первыми битами первого октета, как показано ниже.

Первый бит равен 0: 1 октет.

7 битов передаётся

десятичное значение до 127

Шестнадцатеричное представление октетов от 00 до 7F

Первые биты равны 10: 2 октета.

14 битов передаётся

десятичное значение до 16 383

Шестнадцатеричное представление октетов от 80 00 до BF FF

Первые биты равны 110: 3 октета.

21 бит передаётся

десятичное значение до 2 097 151

Шестнадцатеричное представление октетов от C0 00 00 до DF FF FF

Первые биты равны 111: 4 октета.

29 битов передаётся

десятичное значение до 536 870 911

Шестнадцатеричное представление октетов от E0 00 00 00 до FF FF FF FF

5.4. ROHC UNCOMPRESSED - без компрессии (профиль 0x0000)

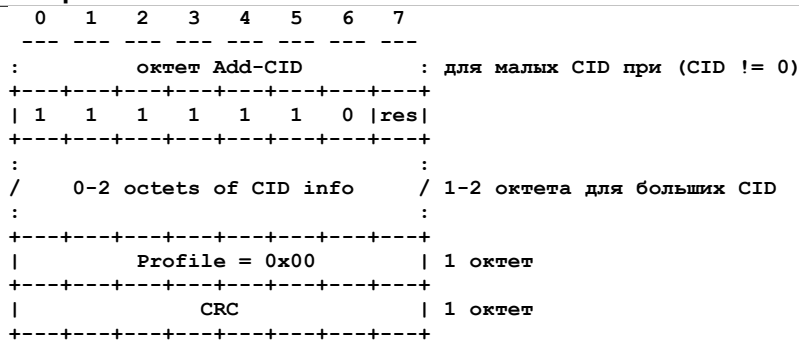
В этом разделе описан профиль ROHC без компрессии, имеющий идентификатор 0x0000.

Профиль 0x0000 обеспечивает способ передачи пакетов IP без их сжатия. Он может использоваться с любыми пакетами, для которых недоступны профили из набора поддерживаемых для канала ROHC или в тех случаях, когда компрессия нежелательна по тем или иным причинам.

После инициализации единственной издержкой при передаче пакетов с использованием Profile 0x0000 является размер CID. Когда несжимаемые пакеты встречаются часто, Profile 0x0000 следует связывать с CID размером 0 или 1 октет. Profile 0x0000 не **следует** связывать с несколькими CID.

5.4.1. Пакет IR

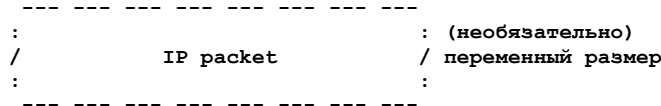
Пакет инициализации и обновления (IR) для Profile 0x0000 имеет формат, показанный на рисунке.



res это поле **должно** иметь значение 0; в противном случае декомпрессор **должен** отбрасывать пакет.

Profile
0x00

CRC
8-битовое значение CRC, рассчитанное с использованием полинома из параграфа 5.3.1.1. CRC покрывает первый октет IR Header до октета Profile в IR Header, т. е., не включает самого поля CRC. В контрольной сумме также не учитываются предшествующие поля Padding или Feedback и Payload.
Для пакетов IR поле Payload имеет формат, показанный на рисунке.

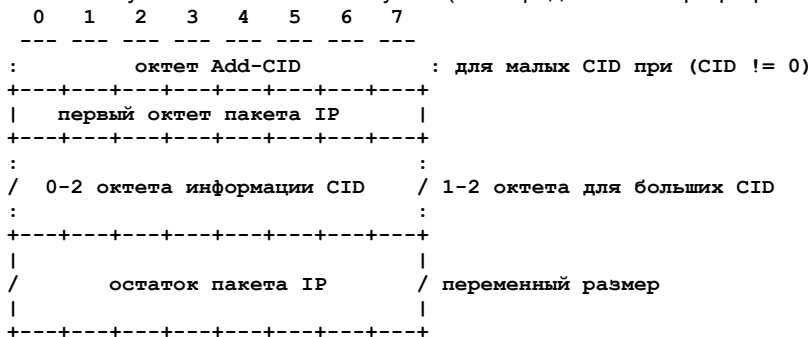


IP packet

Несжатый пакет IP может быть включён в пакет IR. Декомпрессор определяет присутствие пакета IP по размеру пакета IR.

5.4.2. Нормальный пакет

Пакет Normal представляет собой обычный пакет IP с дополнительной информацией CID. Для пакетов Normal формат, показанный на рисунке, соответствует полям Header и Payload (как определено в параграфе 5.2.1).



Отметим, что первый октет пакета IP начинается с битовой последовательности 0100 (IPv4) или 0110 (IPv6). Это не вызывает конфликтов с какими-либо из зарезервированных типов пакетов.

При использовании в канале малых значений CID и связывании профиля 0x0000 с CID > 0, пакету IP предшествует октет Add-CID. Если на канале используются большие значения CID, идентификатор CID начинается со второго октета комбинированного формата Header/Payload, описанного выше.

Пакет Normal может содержать поле Padding и/или Feedback, как любые другие пакеты ROHC, перед комбинированным полем Header/Payload.

5.4.3. Инициализация контекста

Компрессор инициализирует статический контекст, связанный с профилем UNCOMPRESSED путем передачи пакетов IR (см. параграф 5.4.1). В процессе инициализации контекста компрессору **рекомендуется** передавать пакеты IR, пока не будет уверенности в том, что декомпрессор получил хотя бы один такой пакет. Подтверждением получения пакета может служить обратная связь от декомпрессора или сведения о характеристиках канала.

Компрессору **следует** периодически передавать пакеты IR для контекста, связанного с профилем UNCOMPRESSED по крайней мере до получения от декомпрессора обратной связи для этого контекста. Компрессор **может** прекратить периодическую отправку пакетов IR после получения обратной связи.

5.4.4. Работа декомпрессора

При получении пакета IR декомпрессор сначала проверяет его корректность, используя 8-битовое значение CRC.

- Если проверка завершилась отказом для декомпрессора **недопустимо** передавать пакет IP вышележащим уровням;
- при совпадении контрольных сумм декомпрессор
 1. инициализирует контекст, если у него нет корректного контекста для данного CID, связанного с этим профилем,
 2. доставляет пакет IP вышележащим уровням при наличии таковых,

3. может передать подтверждение ACK.

При получении любых других пакетов декомпрессором в отсутствие требуемого контекста такие пакеты отбрасываются без выполнения с ними каких-либо действий.

При получении пакета Normal и наличии корректного контекста у декомпрессора из него извлекается пакет IP, который передаётся вышележащим уровням.

5.4.5. Обратная связь

Единственным типом обратной связи для 0x0000 является подтверждение ACK с использованием формата FEEDBACK-1 (см. параграф 5.2.4.1) с нулевым значением октета profile-specific в поле FEEDBACK-1. Формат FEEDBACK-2 не определён для Profile 0x0000.

6. Обзор профилей ROHC (информационный раздел)

Протокол ROHC включает модель и профили. Модель определяет общие для всех профилей механизмы, а профили определяют конкретный алгоритм сжатия и специфичные для профиля форматы пакетов.

В разделе 5 детально описана модель ROHC. В данном разделе даётся обзор элементов, включаемых в спецификации профилей. Нормативные спецификации конкретных профилей выходят за пределы этого документа.

Профиль ROHC определяет элементы, образующие протокол сжатия, и включает перечисленные ниже элементы.

Форматы пакетов

- Bits-on-the-wire - биты в линии

Профиль определяет схему битов в специфичных для профиля типах пакетов и специфичных для профиля частях типов пакетов общего назначения (например, IR и IR-DYN).

- Field encodings - кодирование полей

Биты и группы битов из схемы формата пакетов, называемые сжатыми полями (Compressed), представляют собой специфичный для метода результат кодирования данного поля в конкретном формате пакета. Эти методы кодирования определяются профилем.

- Updating properties - свойства обновления

Специфичные для профиля форматы пакетов могут обновлять состояние декомпрессора и могут выполнять это разными способами. Профиль определяет обновление контекста декомпрессора с помощью отдельных полей, специфичных для этого профиля, или специфичных для него типов пакетов.

- Verification - верификация

Пакеты, обновляющие состояние декомпрессора, проверяются с целью предотвращения некорректных обновлений контекста декомпрессора. Профиль определяет механизмы, используемые для верификации декомпрессии пакета.

Управление контекстом

- Robustness logic - логика отказоустойчивости

Между компрессором и декомпрессором пакеты могут теряться, а их порядок может изменяться. Профиль определяет механизмы минимизации влияния таких событий и предотвращения распространения повреждений.

- Repair mechanism - механизм исправления

Несмотря на логику отказоустойчивости, нарушения могут приводить к отказам декомпрессии и даже к повреждению контекста в декомпрессоре. Профиль определяет механизмы исправления контекста, включая логику обратной связи, если таковая используется.

7. Благодарности

Авторы выражают свою признательность всем, кто внёс свой вклад в предшествующие работы по ROHC, и особенно авторам документа RFC 3095 [RFC3095], который послужил технической основой для данного документа. Спасибо всем, кто внёс свои правки и разъяснения в корректирующий RFC 3095 документ [RFC4815], - их технические предложения (когда они были применимы) включены в этот документ. Спасибо Jani Juvaan за обнаружение несоответствия между структурой обратной связи, описанной в [RFC4995], и аналогичной структурой из [RFC3095] — это сделало необходимым обновление [RFC4995].

Назначенными рабочей группой рецензентами документа были Carl Knutsson, Biplab Sarkar и Robert Stangarone — они принимали участие в работе над документом вплоть до её завершения. Отдельная благодарность Bert Wijnen и Brian Carpenter за их комментарии в процессе IETF Last Call.

8. Согласование с IANA

Реестр IANA «RObust Header Compression (ROHC) Profile Identifiers» [ROHC-ids] был создан RFC 3095 [RFC3095]. Правила распределения, предложенные в RFC 3095, включают нижеперечисленное.

Идентификатор профиля ROHC представляет собой неотрицательное целое число. Во многих протоколах согласования оно представляется 16-битовым значением. В результате сжатия этих идентификаторов в пакетах ROHC младшие 8 битов идентификатора имеют специальное значение - два идентификатора профилей, совпадающие по восьми младшим битам, следует выделять лишь в тех случаях, когда идентификатор с большим значением предназначен для замены идентификатора с меньшим значением. Для того, чтобы не упустить это из виду,

идентификаторы следует указывать в шестнадцатеричном формате (например, идентификатор 0x1234 можно выделить взамен 0x0A34).

Приведённые далее правила были включены в [RFC5226]. Политика IANA при выделении новых значений идентификаторов профиля должна соответствовать процедуре Specification Required¹ - значения и их трактовки должны быть документированы в RFC или аналогичном документе, доступном по известной и постоянной ссылке, достаточно подробно для обеспечения возможности взаимодействия независимых реализаций. В младших 8 битах идентификатора значения от 0 до 127 резервируются для спецификаций проектов стандартов IETF, диапазон от 128 до 254 доступен для других спецификаций, удовлетворяющих требованиям (таких, как информационные RFC). Значение 255 в младших восьми битах зарезервировано для будущих расширений данной спецификации.

Выделенные идентификаторы профилей перечислены в таблице.

Идентификатор профиля	Применение	Документ
0x0000	ROHC uncompressed	RFC 5795
0x0001	ROHC RTP	RFC 3095
0x0002	ROHC UDP	RFC 3095
0x0003	ROHC ESP	RFC 3095
0x0004	ROHC IP	RFC 3843
0x0005	ROHC LLA	RFC 3242
0x0105	ROHC LLA with R-mode	RFC 3408
0x0006	ROHC TCP	RFC 4996
0x0007	ROHC RTP/UDP-Lite	RFC 4019
0x0008	ROHC UDP-Lite	RFC 4019
0x0101	ROHCv2 RTP	RFC 5225
0x0102	ROHCv2 UDP	RFC 5225
0x0103	ROHCv2 ESP	RFC 5225
0x0104	ROHCv2 IP	RFC 5225
0x0107	ROHCv2 RTP/UDP-Lite	RFC 5225
0x0108	ROHCv2 UDP-Lite	RFC 5225

Для новых профилей потребуется выделение агентством IANA новых идентификаторов, но этот документ не требует от IANA каких-либо действий.

9. Вопросы безопасности

Поскольку шифрование устраняет избыточность, которую пытаются использовать в схемах компрессии, возникает желание отказаться от шифрования заголовков для обеспечения работы по низкоскоростным каналам.

Неисправный или вредоносный компрессор может привести к тому, что пакеты после декомпрессии не будут соответствовать оригиналу, но сохранят корректные заголовки и, возможно, корректные контрольные суммы транспортного уровня. Такие повреждения могут быть обнаружены при использовании механизмов сквозной аутентификации и контроля целостности, на которые компрессия не может оказать своего влияния. Кроме того, в модели компрессии заголовков ROHC используются свои контрольные суммы для проверки реконструированных заголовков, которые снижают вероятность незаметного появления после декомпрессии заголовков, не соответствующих оригиналу.

Атаки на службы (DoS²) будут возможны, если атакующий способен ввести в канал обманные пакеты IR, IR-DYN или пакеты обратной связи, снижая тем самым эффективность компрессии. Однако атакующий с возможностью вставки на канальном уровне произвольных пакетов способен нанести более серьёзный урон, нежели снижение эффективности компрессии заголовков.

10. Литература

10.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.

10.2. Дополнительная литература

[CRTP-eval] Degermark, M., Hannu, H., Jonsson, L., and K. Svanbro, "Evaluation of CRTP Performance over Cellular Radio Networks", IEEE Personal Communication Magazine, Volume 7, number 4, pp. 20-25, August 2000.", 2000.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.

[RFC1144] Jacobson, V., "Compressing TCP/IP headers for low-speed serial links", [RFC 1144](#), February 1990.

[RFC1662] Simpson, W., "PPP in HDLC-like Framing", STD 51, [RFC 1662](#), July 1994.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.

[RFC2507] Degermark, M., Nordgren, B., and S. Pink, "IP Header Compression", RFC 2507, February 1999.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "Robust

¹Требуется спецификация.

²Denial-of-service.

- Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.
- [RFC3096] Degermark, M., "Requirements for robust IP/UDP/RTP header compression", RFC 3096, July 2001.
- [RFC3241] Bormann, C., "Robust Header Compression (ROHC) over PPP", RFC 3241, April 2002.
- [RFC3545] Koren, T., Casner, S., Geevarghese, J., Thompson, B., and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, July 2003.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3759] Jonsson, L-E., "RObust Header Compression (ROHC): Terminology and Channel Mapping Examples", RFC 3759, April 2004.
- [RFC4224] Pelletier, G., Jonsson, L-E., and K. Sandlund, "Robust Header Compression (ROHC): ROHC over Channels That Can Reorder Packets", RFC 4224, January 2006.
- [RFC4815] Jonsson, L-E., Sandlund, K., Pelletier, G., and P. Kremer, "RObust Header Compression (ROHC): Corrections and Clarifications to RFC 3095", RFC 4815, February 2007.
- [RFC4995] Jonsson, L-E., Pelletier, G., and K. Sandlund, "The RObust Header Compression (ROHC) Framework", RFC 4995, July 2007.
- [RFC4996] Pelletier, G., Sandlund, K., Jonsson, L-E., and M. West, "RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP)", RFC 4996, July 2007.
- [RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, April 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [ROHC-ids] IANA, "RObust Header Compression (ROHC) Profile Identifiers", <<http://www.iana.org>>.

Приложение А. Алгоритм CRC

```
#!/usr/bin/perl -w
use strict;
#####
# ROHC CRC demo - Carsten Bormann cabo@tzi.org 2001-08-02
#
# Эта маленькая программа показывает 4 типа CRC, используемых в 3095
# для отказоустойчивого сжатия заголовков. Введите данные в
# шестнадцатеричном формате и нажмите Control+D.
#
#-----
#
# utility
#
sub dump_bytes($) {
    my $x = shift;
    my $i;
    for ($i = 0; $i < length($x); ) {
        printf("%02x ", ord(substr($x, $i, 1)));
        printf("\n") if (++$i % 16 == 0);
    }
    printf("\n") if ($i % 16 != 0);
}

#-----
#
# Алгоритм расчёта CRC
#
sub do_crc($$$) {
    my $nbits = shift;
    my $poly = shift;
    my $string = shift;

    my $crc = ($nbits == 32 ? 0xffffffff : (1 << $nbits) - 1);
    for (my $i = 0; $i < length($string); ++$i) {
        my $byte = ord(substr($string, $i, 1));
        for (my $b = 0; $b < 8; $b++) {
            if (($crc & 1) ^ ($byte & 1)) {
                $crc >>= 1;
                $crc ^= $poly;
            } else {
                $crc >>= 1;
            }
            $byte >>= 1;
        }
    }

    printf "%2d bits, ", $nbits;
    printf "CRC: %02x\n", $crc;
}
}
```

```
#-----  
#  
# Тестирование  
#  
$/ = undef;  
$_ = <>; # читать до конца файла  
my $string = ""; # извлечь все шестнадцатеричное  
s/([0-9a-fA-F]{0-9a-fA-F})/$string .= chr(hex($1)), ""/eg;  
dump_bytes($string);  
  
#-----  
#  
# 32-битовое значение CRC для сегментации  
# Отметим, что в тексте предполагается дополнение, как в PPP  
# (это отличается от 8, 7 и 3-битовых CRC)  
#  
#  $C(x) = x^0 + x^1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} +$   
#  $x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$   
#  
do_crc(32, 0xedb88320, $string);  
  
#-----  
#  
# 8-битовая сумма для IR/IR-DYN  
#  
#  $C(x) = x^0 + x^1 + x^2 + x^8$   
#  
do_crc(8, 0xe0, $string);  
  
#-----  
#  
# 7-битовая сумма для FO/SO  
#  
#  $C(x) = x^0 + x^1 + x^2 + x^3 + x^6 + x^7$   
#  
do_crc(7, 0x79, $string);  
  
#-----  
#  
# 3-битовая сумма FO/SO  
#  
#  $C(x) = x^0 + x^1 + x^3$   
#  
do_crc(3, 0x6, $string);
```

Адреса авторов

Kristofer Sandlund
Ericsson
Box 920
Lulea SE-971 28
Sweden
Phone: +46 (0) 8 404 41 58
EMail: kristofer.sandlund@ericsson.com

Ghyslain Pelletier
Ericsson
Box 920
Lulea SE-971 28
Sweden
Phone: +46 (0) 8 404 29 43
EMail: ghyslain.pelletier@ericsson.com

Lars-Erik Jonsson
Optand 737
Ostersund SE-831 92
Sweden
Phone: +46 76 830 03 12
EMail: lars-erik@lejonsson.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru