Network Working Group
Request for Comments: 4413
Category: Informational

M. West
S. McCann
Siemens/Roke Manor Research

March 2006

Поведение полей TCP/IP

TCP/IP Field Behavior

Статус документа

В этом документе содержится информация для сообщества Internet. Документ не задает каких-либо стандартов Internet. Допускается свободное распространение документа.

Аннотация

В этом документе описано поведение полей TCP/IP в контексте сжатия заголовков. Такое сжатие возможно благодаря тому, что большинство полей заголовка незначительно отличается от пакета к пакету. Многие из полей являются статическими или меняются более или менее предсказуемо. При разработке схем компрессии заголовков весьма важно понимать поведение полей. Примером такого анализа может служить RFC 3095. Данный документ играет аналогичную роль для сжатия заголовков TCP/IP.

Оглавление

1. Введение

В этом документе описывается формат заголовков TCP/IP и поведение полей заголовка (т. е., изменение этих полей в потоке TCP). Описание приводится в контексте сжатия заголовков.

Поскольку поведение заголовков IP несколько отличается от описанного ранее в RFC 3095 [31] для протоколов UDP и RTP, эти заголовки также рассматриваются здесь.

Этот документ заимствует множество классификационных фрагментов из RFC 3095 вместо включения ссылок на этот документ.

Согласно формату, представленному в RFC 3095 [31], поля заголовков TCP/IP классифицируются и анализируются в два этапа. Сначала мы проводим общую классификацию (глава 2), в которой поля подразделяются на основе установившихся сведений и допущений. Эта общая классификация не принимает во внимание изменение характеристик полей, в той или иной степени зависящее от реализации или используемого приложения. В главе 3 рассматривается вопрос использования значений полей для оптимизации короткоживущих потоков. Более детальный анализ изменения характеристик приводится в главе 4. В заключение глава 5 описывает обработку полей заголовков с учетом их оптимального сжатия.

Основным вопросом является определение базы для рассмотрения имеющихся реализаций TCP/IP. Этот обзор основан на анализе развернутых в настоящее время реализаций TCP, которые поддерживают стандартизованные IETF механизмы.

Представляют интерес также общие требования в части прозрачности. Множество предложенных недавно расширений TCP используют ранее зарезервированные биты полей в заголовках TCP. Поэтому не следует полагаться на то, что зарезервированные биты имеют нулевые значения - они могут изменяться. В идеальном варианте схема компрессии должна принимать это во внимание.

Множество зарезервированных битов доступно для использования в будущих расширениях. Трактовка поведения полей не может предсказать использование этих битов в будущем, но мы ожидаем, что они будут применяться в некоторых случаях. С учетом этого схему компрессии можно оптимизировать для текущей ситуации, но следует обеспечить возможность поддержки любого использования зарезервированных битов. Однако обеспечить оптимизацию для битов, которые еще не определены, не представляется возможным.

2. Общая классификация

Приведенные ниже определения (и часть текста) скопированы из Приложения A в RFC 3095 [31]. Отличия в поведении полей IP от RFC 3095 [31] (например, поведение IP/UDP/RTP для аудио и видео-приложений) явно указываются в этом документе.

Далее в документе термин «сессия» будет использоваться для потока пакетов TCP, представляющего собой серию пакетов с одинаковыми адресами IP и номерами портов. Поток пакетов определяется некими полями (см. ниже STATIC-DEF) и может рассматриваться как подмножество сессии. Более детально разделение сессий на потоки пакетов для сжатия заголовков рассматривается в документе [31].

Заголовки пакетов делятся на 5 классов:

INFERRED - опосредованные
: Эти поля содержат значения, которые могут быть определены на основе других значений (например, размер кадра, содержащего пакет), и по этой причине не обрабатываются в процессе сжатия.

STATIC - статические
: Значения этих полей предполагаются неизменными в течение срока существования потока пакетов. Статическая информация должна тем или иным путем передаваться однократно.

STATIC-DEF - статические определяющие
: Поля типа STATIC, значения которых определяют поток пакетов. В общем случае эти значения обрабатываются как STATIC.

STATIC-KNOWN - статические известные
: Поля типа STATIC, которые предположительно содержат общепринятые (well-known) значения и, следовательно, могут не передаваться.

CHANGING - изменяющиеся
: Предполагается, что такие поля могут принимать произвольные значения из ограниченного набора или диапазона.

В этой главе каждое поле заголовков IP и TCP относится к тому или иному классу. Для всех полей, кроме класса CHANGING, приводится также обоснование этой классификации. В главе 4 проводится дополнительное рассмотрение и классификация полей CHANGING на основе их предполагаемого поведения.

2.1. Поля заголовка IP

2.1.1. Поля заголовка IPv6

Version - версия
: Это поле указывает номер версии протокола IP. Пакеты с отличающимися значениями этого поля должны обрабатываться разными стеками IP. Все пакеты одного потока должны, следовательно, иметь одинаковую версию IP. Это позволяет отнести поле к классу STATIC.

Flow Label - метка потока
: Это поле используется для идентификации пакетов, относящихся к одному потоку. Если идентификатор потока не используется, в этом поле следует устанавливать нулевое значение. Во всех остальных случаях пакеты одного потока должны использовать одинаковое значение идентификатора, которое является одним из полей, определяющих поток. Это поле, следовательно, классифицируется, как STATIC-DEF.

Payload Length - размер данных
: Предполагается, что информация о пакете (включая размер содержащихся в нем данных) обеспечивается канальным уровнем. Это поле, следовательно, относится к классу INFERRED.

Поле	*Размер в битах*	*Класс*
Version	4	STATIC
DSCP[1]	6	ALTERNATING
Флаг ECT[1]	1	CHANGING
Флаг CE[1]	1	CHANGING
Flow Label	20	STATIC-DEF
Payload Length	16	INFERRED
Next Header	8	STATIC
Hop Limit	8	CHANGING
Source Address	128	STATIC-DEF
Destination Address	128	STATIC-DEF

Рисунок 1. Поля заголовка IPv6

Next Header - следующий заголовок

Обычно имеет одинаковое значение во всех пакетах одного потока и указывает тип следующего заголовка. Значение поля в течение срока существования потока пакетов может изменяться только в результате отсутствия расширенных заголовков, поэтому поле классифицируется, как STATIC. Такая классификация унаследована от RFC 3095 [31]. Однако следует отметить, что поле Next Header в действительности определяется типом следующего заголовка. Возможно, что более корректно было бы отнести это поле к типу опосредованных, хотя это зависит от конкретной реализации схемы компрессии.

Source Addresses и Destination Addresses - адреса отправителя и получателя

Эти поля являются частью определения потока пакетов и остаются неизменными для данного потока. Таким образом, поля классифицируются, как STATIC-DEF. Такое представление может показаться несколько упрощенным, но в данном документе адреса IP, связанные с соединением транспортного уровня, рассматриваются как часть определения потока. Естественно, что могут существовать и более сложные определения для разделения потоков (дополнительное обсуждение этого вопроса можно найти в RFC 3095 [31]). При использовании туннелей адреса IP во внешних заголовках туннеля также относятся к классу STATIC-DEF.

Суммарные размеры полей каждого класса показаны на рисунке 2.

Класс	*Размер в октетах*
INFERRED	2
STATIC	1,5
STATIC-DEF	34,5
STATIC-KNOWN	0
CHANGING	2

Рисунок 2. Размеры полей.

2.1.2. Поля заголовка IPv4

Поле	*Размер в битах*	*Класс*
Version	4	STATIC
Header Length	4	STATIC-KNOWN
DSCP[1]	6	ALTERNATING
Флаг ECT[1]	1	CHANGING
Флаг CE[1]	1	CHANGING
Packet Length	16	INFERRED
Identification	16	INFERRED
Reserved Flag[1]	1	CHANGING
Флаг запрета фрагментирования[1]	1	CHANGING
Флаг наличия дополнительных фрагментов	1	STATIC-KNOWN
Fragment Offset	13	STATIC-KNOWN
Time To Live	8	CHANGING
Protocol	8	STATIC
Header Checksum	16	INFERRED
Source Address	32	STATIC-DEF
Destination Address	32	STATIC-DEF

Рисунок 3. Поля заголовка IPv4

Version - версия

Это поле указывает номер версии протокола IP. Пакеты с отличающимися значениями этого поля должны обрабатываться разными стеками IP. Все пакеты одного потока должны, следовательно, иметь одинаковую версию IP. Это позволяет отнести поле к классу STATIC.

Packet Length - размер пакета

Предполагается, что информация о размере пакета обеспечивается канальным уровнем. Это поле, следовательно, относится к классу INFERRED.

Flags - флаги

Поле резервного флага (Reserved) должно имеет значение 0 в соответствии с RFC 791 [1]. Поэтому в RFC 3095 [31] поле классифицируется как STATIC-KNOWN. Однако предполагается возможность использования зарезервированных полей в будущем. Нежелательно выбирать такое представление, которое будет препятствовать использованию профиля компрессии при изменении заголовка в результате использования резервных полей. По этой причине используется классификация поля как CHANGING. Коммуникационный профиль, естественно, может быть оптимизирован с учетом текущей ситуации, когда значение поля известно заранее (0).

[1]Отличается от RFC 3095 [31], где флаги DSCP, ECT и CE были объединены в октет TOS (поведение флага DF обсуждается позднее, остальные флаги рассмотрены ниже в этом параграфе).

Флаг наличия дополнительных фрагментов (MF) предполагается нулевым, поскольку фрагментации в идеальном случае не ожидается. Однако следует понимать, что в некоторых сценариях работы (например, в некоторых вариантах архитектуры туннелирования) фрагментация может возникать. В общем случае, тем не менее, предполагается отсутствие фрагментации в Internet (благодаря начальному согласованию MSS и последующему применению механизма Path-MTU discovery). RFC 3095 [31] указывает, для протокола RTP только первый фрагмент будет содержать заголовок транспортного уровня и последующие фрагменты будут сжиматься с использованием другого профиля. Это нормальная ситуация для TCP. Если возникает фрагментация, первый фрагмент по определению будет относительно большим для минимизации относительного размера заголовков. Последующие фрагменты будут сжиматься с использованием другого профиля. Следовательно, представляется нежелательной оптимизация фрагментирования при сжатии заголовков. Флаг наличия дополнительных фрагментов в результате классифицируется, как STATIC- KNOWN.

Fragment Offset - смещение фрагмента

В предположении отсутствия фрагментации смещение фрагмента всегда равно нулю. Следовательно, это поле классифицируется, как STATIC-KNOWN. Даже если принимать во внимание фрагментацию, только первый фрагмент будет содержать заголовок TCP и смещение фрагмента в этом заголовке будет равно нулю.

Protocol - протокол

Это поле обычно имеет одинаковое значение во всех пакетах одного потока. Данное поле определяет тип последующего заголовка.

Это поле меняет свое значение лишь в случае изменения последовательности заголовков (например, вставляется или удаляется расширение заголовка или туннельный заголовок). Следовательно, поле относится к классу STATIC. Будет ли такое изменение приводить к тому, что последовательность пакетов станет трактоваться, как новый поток (с точки зрения сжатия заголовков), определяется профилем. Профили ROHC[1] должны обеспечивать возможность работы с расширенными заголовками и туннелями, но выбор стратегии не входит в задачи данного документа.

Header Checksum - контрольная сумма заголовка

Контрольная сумма заголовка позволяет предотвратить обработку поврежденных пакетов. Когда почти вся информация заголовка IP представлена в сжатом виде, не возникает необходимости в использовании дополнительной контрольной суммы. Вместо этого значение контрольной суммы восстанавливается при декомпрессии. Следовательно, поле классифицируется, как INFERRED.

Отметим, что контрольная сумма заголовка TCP защищает не все заголовки TCP/IP, а только псевдозаголовок TCP (и данные).

ROHC [31] использует значение CRC[2] для проверки несжатого заголовка. С учетом необходимости проверки корректности всего заголовка TCP/IP, затрат на расчет контрольной суммы TCP с учетом всех данных и известные недостатки контрольных сумм TCP [37], дополнительная проверка является необходимой. Следовательно, весьма желательно использование дополнительной контрольной суммы (такой, как CRC) для проверки корректности декомпрессии.

Source Addresses и Destination Addresses - адреса отправителя и получателя

Эти поля являются частью определения потока пакетов и, следовательно, являются неизменными для конкретного потока. Таким образом, поля классифицируются как STATIC-DEF.

Общие размеры заголовков разного класса показаны на рисунке 4.

Класс	***Размер в октетах***
INFERRED	4
STATIC[3]	1,5
STATIC-DEF	8
STATIC-KNOWN[3]	2,25
CHANGING[3]	4,25

Рисунок 4. Размеры полей.

2.2. Поля заголовка TCP

Source Addresses и Destination Addresses - адреса отправителя и получателя

Эти поля являются частью определения потока пакетов и, следовательно, остаются неизменными для конкретного потока. Таким образом, поля классифицируются, как STATIC-DEF.

Data Offset - смещение данных

Число 4-октетных слов в заголовке TCP, показывающее начало данных (всегда выровнено по 4-октетной границе). Это значение может быть восстановлено из размера всех опций, следовательно не возникает необходимости в его явной передаче. В результате поле классифицируется как INFERRED.

[1] Robust Header Compression.
[2] Контрольная сумма. *Прим. перев.*
[3] Отличается от RFC 3095 [31].

Поле	*Размер в битах*	*Класс*
Source Port	16	STATIC-DEF
Destination Port	16	STATIC-DEF
Sequence Number	32	CHANGING
Acknowledgement Num	32	CHANGING
Data Offset	4	CHANGING
Резерв	4	CHANGING
Флаг CWR	1	CHANGING
Флаг ECE	1	CHANGING
Флаг URG	1	CHANGING
Флаг ACK	1	CHANGING
Флаг PSH	1	CHANGING
Флаг RST	1	CHANGING
Флаг SYN	1	CHANGING
Флаг FIN	1	CHANGING
Window	16	CHANGING
Checksum	16	CHANGING
Urgent Pointer	16	CHANGING
Options	0(-352)	CHANGING

Рисунок 5. Поля заголовка TCP.

2.3. Общие размеры для IP/TCP

В целом поля разных классов в заголовках IP/TCP имеют размеры, показанные на рисунке 6

Класс	*Число октетов*	
	IPv6	IPv4
INFERRED	2,5	4,5
STATIC	1,5	1,5
STATIC-DEF	38,5	12
STATIC-KNOWN	-	2,25
CHANGING	17,25	19,75
Всего	60	40

Рисунок 6. Суммарные размеры полей.

Опции класса CHANGING не учитывались.

3. Классификация повторяющихся полей заголовков

В тех случаях, когда множество потоков перекрываются по времени или используются последовательно в течение короткого времени, приходиться иметь дело с похожими значениями полей заголовков. Такое сходство полей возникает и в контексте сжатия. Таким образом, следует использовать сходство полей различных потоков для повышения степени сжатия. Для решения этой задачи важно найти «повторяющиеся» характеристики различных полей заголовков.

Ключевым моментом «репликации» характеристик является использование текущего контекста в качестве базы при создании нового контекста. То, что было изменено, обновляется или переписывается с использованием значений из пакета, вызвавшего репликацию. В этой главе рассматриваются общие характеристики полей из различных потоков.

Отметим, что репликация основывается на контексте (а не просто на значениях полей) и созданные при сжатии поля также могут включаться в этот контекст. Эти поля, естественно, зависят от используемого протокола сжатия (профиля ROHC).

Варианты возможности репликации для полей TCP/IP перечислены ниже.

N/A: поле не рассматривается в процессе репликации, поскольку оно относится к числу опосредованных (inferred) или известно a priori (и, следовательно, не появляется в контексте).

No: поле не может реплицироваться, поскольку отсутствует корреляция между значениями этого поля в двух потоках пакетов.

Yes: поле может реплицироваться. Это не гарантирует совпадения значений в двух потоках-кандидатах и лишь позволяет использовать репликацию для повышения коэффициента сжатия. Для повышения эффективности компресии могут применяться различные методы кодирования.

3.1. Заголовок IPv4 (внутренний и/или внешний)

Поле	*Класс*	*Репликация*
Version	STATIC	N/A
Header Length	STATIC-KNOWN	N/A
DSCP	ALTERNATING	No (1)
Флаг ECT	CHANGING	No (2)
Флаг CE	CHANGING	No (2)
Packet Length	INFERRED	N/A
Identification	INFERRED	Yes (3)
Reserved Flag	CHANGING	No (4)
Флаг DF	CHANGING	Yes (5)
Флаг MF	STATIC-KNOWN	N/A
Fragment Offset	STATIC-KNOWN	N/A
Time To Live	CHANGING	Yes
Protocol	STATIC	N/A
Header Checksum	INFERRED	N/A
Source Address	STATIC-DEF	Yes
Destination Address	STATIC-DEF	Yes

Рисунок 7. Заголовок IPv4.

(1) Поле DSCP маркируется в соответствии с требованиями приложения. Если можно предположить, что реплицируемые соединения относятся к одному классу diffserv, очевидно, что значения DSCP будут реплицируемыми. Значение DSCP может устанавливать не только отправитель, но и любой, кому позволено маркировать пакеты. Таким образом, пакет может использовать множество значений DSCP в разных точках сети. Однако компрессия заголовках используется на соединениях «точка-точка», поэтому значение данного можно считать сравнительно стабильным. Если выполняется перемаркировка на основе состояния измерителя, значение поля может измениться посреди потока. В целом мы полагаем, что репликация DSCP будет полезна для сжатия заголовков.

(2) Поле ECN невозможно реплицировать (отметим, что ожидается использование схемы ECN nonce [19]). Однако представляется очевидным, что все потоки TCP между хостами, поддерживающими ECN, будут применять ECN, поэтому использование ECN (или отказ от него) для потоков между одной парой хостов можно рассматривать как реплицируемое. См. также (4).

(3) Реплицируемый контекст для этого поля включает флаги IP-ID, NBO и RND (как описано в ROHC RTP). Это подчеркивает, что репликация происходит для контекста, а не просто для отдельных значений полей и, таким образом, должна рассматриваться с учетом точной природы компрессии, используемой для каждого поля.

(4) Поскольку будущее поведение поля Reserved Flag предсказать невозможно, не представляется возможным рассматривать и вопрос его репликации. Однако можно предполагать, что поведение этого поля в разных пакетах между одной парой конечных точек будет похожим. В этом случае любой выбор формата пакетов (например) может передаваться в новый поток. В случае формата пакетов решение может приниматься локально системой сжатия заголовков.

(5) Теоретически бит DF можно реплицировать. Однако, практическая польза этого не очевидна. С точки зрения сжатия заголовков очевидно, что явная передача этого 1-битового флага не потребует большего объема, нежели индикация возможности реплицирования. Мы не включаем флаг DF в число реплицируемых.

3.2. Заголовок IPv6 (внутренний и/или внешний)

Поле	*Класс*	*Репликация*
Version	STATIC	N/A
Traffic Class	CHANGING	Yes (1)
Флаг ECT	CHANGING	No (2)
Флаг CE	CHANGING	No (2)
Flow Label	STATIC-DEF	N/A
Payload Length	INFERRED	N/A
Next Header	STATIC	N/A
Hop Limit	CHANGING	Yes
Source Address	STATIC-DEF	Yes
Destination Address	STATIC-DEF	Yes

Рисунок 8. Заголовок IPv6.

(1) См. примечание для поля DSCP в заголовке IPv4 (п. 1, выше).

(2) См. примечание для флагов ECT и CE в заголовке IPv4 (п. 2, выше).

3.3. Заголовок TCP

Поле	*Класс*	*Репликация*
Source Port	STATIC-DEF	Yes (1)
Destination Port	STATIC-DEF	Yes (1)
Sequence Number	CHANGING	No (2)
Acknowledgement Num	CHANGING	No
Data Offset	CHANGING	N/A
Резерв	CHANGING	No (3)
Флаг CWR	CHANGING	No (4)
Флаг ECE	CHANGING	No (4)
Флаг URG	CHANGING	No
Флаг ACK	CHANGING	No
Флаг PSH	CHANGING	No
Флаг RST	CHANGING	No
Флаг SYN	CHANGING	No
Флаг FIN	CHANGING	No
Window	CHANGING	Yes
Checksum	CHANGING	No
Urgent Pointer	CHANGING	Yes (5)

Рисунок 9. Заголовок TCP

(1) Очевидно, что номер порта на серверной стороне относится к числу общепринятых (well-known). На клиентской стороне номер порта обычно выбирается стеком протоколов автоматически. Возможность репликации номера зависит от того, как стек протоколов выбирает номера портов. Хотя наиболее популярные реализации используют последовательное выделение номеров, расчет на такое поведение может оказаться нежелательным.

(2) Рекомендация (и ожидаемое развертывание) использования случайных значений для начальных порядковых номеров TCP[1] в соответствии с RFC 1948 [10] делает невозможным совместное использование порядковых номеров. Таким образом, это поле не может считаться реплицируемым.

(3) См. выше комментарий (4) для полей заголовка IPv4.

(4) См. выше комментарий (2) для флага ECN в заголовке IPv4.

(5) Указатель срочности используется очень редко. Это означает, что на практике поле может рассматриваться как реплицируемое.

3.4. Опции TCP

Опция	*Только SYN (1)*	*Репликация*
End of Option List	No	No (2)
No-Operation	No	No (2)
Maximum Segment Size	Yes	Yes
Window Scale	Yes	Yes
SACK-Permitted	Yes	Yes
SACK	No	No
Timestamp	No	No

Рисунок 10. Опции TCP.

(1) Эта колонка показывает, что данная опция может использоваться только в пакетах SYN (Yes) или в других пакетах также (No). Многие опции TCP используются только в пакетах SYN. Некоторые опции (например, MSS, Window Scale и SACK-Permitted) имеют тенденцию сохранять свои значения в потоке пакетов.

Таким образом, для поддержки контекста совместного использования системе компрессии следует такие опции TCP в контексте (даже если они появляются только в сегментах SYN).

(2) Поскольку эти опции имеют фиксированные значения, они могут рассматриваться как реплицируемые. Однако единственно, что интересно передавать об этих опциях - это их присутствие. Если известно, что такая опция существует, ее значение также известно.

3.5. Сводные данные по репликации

Из приведенного выше анализа можно видеть, что существуют разумные основания для использования избыточности, присутствующей в разных потоках пакетов и в пакетах одного потока. Просто рассмотрите преимущества, которые можно получить, опуская адреса отправителя и получателя для повторяющихся соединений между парой конечных точек IPv6. Существует также цена (в терминах сложности и устойчивости) реплицируемого контекста и она должна приниматься во внимание при выборе решения.

В заключение отметим, что использование репликации требует, чтобы система компрессии имела уверенность в том, что нужные данные (отправителя) присутствуют и корректны на стороне декомпрессии. Это может вносить некоторые ограничения на использование «изменяемых» полей в процессе репликации.

4. Анализ картины изменения полей заголовков

Для создания подходящего механизма эффективной компрессии всех полей заголовка следует проанализировать картину изменения этих полей. Для такого анализа здесь вводится дополнительная классификация полей, которые в главе 2 были отнесены к классу CHANGING (изменяющиеся).

Поля класса CHANGING разделены на 5 дополнительных субклассов:

[1]TCP Initial Sequence Number

- **STATIC** - статические

  Эти поля были отнесены к классу CHANGING при общем рассмотрении, но они квалифицируются как статические с учетом некоторых добавочных допущений.

- **SEMISTATIC** - полустатические

  Эти поля относятся к типу STATIC большую часть времени. Однако время от времени значение может меняться и после известного числа пакетов возвращаться к первоначальному.

- **RARELY-CHANGING (RC)** - редкое изменение

  Эти поля изменяют свое значение достаточно редко и сохраняют новое значение.

- **ALTERNATING** - чередование

  В этих полях чередуется небольшой набор отличающихся значений.

- **IRREGULAR** - непредсказуемые изменения

  Это поля, для которых нет возможности идентифицировать ту или иную регулярность изменений.

Для дополнительного расширения этой классификации без ее усложнения можно использовать значения полей и/или диапазоны их изменения.

При классификации полей принимались во внимание дополнительные сведения и/или диапазоны возможных изменений. Для полей класса STATIC или SEMISTATIC значение поля может относиться не только к классу STATIC но быть также заранее **известным** (KNOWN) общепринятым значением (два состояния для полей SEMISTATIC). Для полей с непредсказуемым поведением может быть известно, что обычно изменения происходят в **ограниченном** (LIMITED) диапазоне всех возможных значений. Для остальных полей значения совершенно **неизвестны** (UNKNOWN).

На рисунке 11 показана классификация полей класса CHANGING на основе предполагаемой картины их изменения. (4) относится к полям IPv4, а (6) - к полям IPv6.

В следующих параграфах приведено детальное обсуждение различных полей заголовков. Отметим, что рисунок 11[1] и и это обсуждение не учитывают потери или нарушения порядка доставки пакетов до точки сжатия.

Поле	***Значение/диапазон***	***Класс***	***Дополнительные сведения***
DSCP(4) / Traffic-Class (6)	Значение	ALTERNATING	UNKNOWN
Флаг IP ECT (4)	Значение	RC	UNKNOWN
Флаг IP CE (4)	Значение	RC	UNKNOWN
IP Id (4) последовательный	Диапазон	STATIC	KNOWN
IP Id (4) - увеличение	Диапазон	RC	LIMITED
IP Id (4) случайный	Значение	IRREGULAR	UNKNOWN
Флаг IP DF (4)	Значение	RC	UNKNOWN
IP TTL(4) / Hop Lim(6)	Значение	ALTERNATING	LIMITED
Порядковый номер TCP	Диапазон	IRREGULAR	LIMITED
Номер подтверждения TCP	Диапазон	IRREGULAR	LIMITED
TCP Reserved	Значение	RC	UNKNOWN
Флаг ECN	Значение	IRREGULAR	UNKNOWN
Флаг CWR	Значение	IRREGULAR	UNKNOWN
Флаг ECE	Значение	IRREGULAR	UNKNOWN
Флаг URG	Значение	IRREGULAR	UNKNOWN
Флаг ACK	Значение	IRREGULAR	KNOWN
Флаг PSH	Значение	IRREGULAR	UNKNOWN
Флаг RST	Значение	IRREGULAR	UNKNOWN
Флаг SYN	Значение	IRREGULAR	KNOWN
Флаг FIN	Значение	IRREGULAR	KNOWN
Окно TCP	Значение	ALTERNATING	KNOWN
Контрольная сумма TCP	Значение	IRREGULAR	UNKNOWN
Указатель срочности TCP	Значение	IRREGULAR	KNOWN
Опции TCP	Значение	IRREGULAR	UNKNOWN

Рисунок 11. Классификация полей CHANGING.

4.1. Заголовок IP

4.1.1. IP Traffic-Class / Type-Of-Service (TOS)

Предполагается, что поля Traffic-Class (IPv6) и Type-Of-Service/DSCP (IPv4) могут изменять свои значения в течение срока существования потока пакетов. Этот анализ принимает во внимание несколько RFC, описывающих изменения исходной спецификации протокола в RFC 791 [1].

Байт TOS описан в исходной спецификации RFC 791 [1] как 3-битовое поле уровня предпочтения, за которым следует 3 бита TOS и 2 резервных бита (по определению равные 0). В RFC 1122 [21] размер поля TOS расширен до 5 битов, но значения двух добавленных битов не определены. RFC 1349 [23] определяет четвертый бит TOS как minimize monetary cost[2]. Следующее важное изменение содержится в RFC 2474 [14] (этот документ отменяет действие RFC 1349 [23]). RFC 2474 переопределяет октет TOS как 6-битовое поле DSCP (DiffServ Code Point), за которым следует 2 неиспользуемых бита. Позднее в RFC 2780 [30] эти два резервных бита октета TOS (или traffic class - класс трафика) определены для экспериментов с ECN.

[1]В оригинале здесь приведена ссылка на отсутствующую в документе таблицу 1. *Прим. перев.*
[2]Минимизация расходов в денежном выражении.

Поэтому предполагается классифицировать октет TOS (или traffic class) как 6 битов DSCP и 2 дополнительных бита. Эти два бита могут предполагаться нулевыми или содержащими данные ECN. С учетом перспективы предпочтительней предполагать использование ECN, особенно в случае TCP.

Следует также обеспечить работу профиля со старыми вариантами стека, поскольку они будут применяться еще достаточно долго. Для простоты мы будем рассматривать поле как 6 битов TOS и 2 бита данных ECN во всех случаях.

Значение DSCP (как TOS в ROHC RTP) не предполагается изменяющимся часто (хотя оно может смениться посреди потока, например, в результате изменения маршрута).

4.1.2. Флаги ECN

Сначала мы опишем флаги ECN в соответствии со спецификациями RFC 2481 [15] и RFC 3168 [18]. После этого будет описанное предлагаемое обновление, которое будет менять поведение флагов.

В RFC 2481 [15] определены 2 отдельных флага - ECT (ECN Capable Transport[1]) и CE (Congestion Experienced[2]). Флаг ECT, если он согласован стеком TCP, будет иметь значение 1 для всех пакетов с данными и 0 - для пакетов, содержащих только подтверждения. Это означает, что поведение флага ECT связано с поведением стека TCP. Непонятно, можно ли это использовать для компрессии.

Флаг CE используется только при установке ECT = 1. Отправитель устанавливает для флага значение 0, которое может быть заменено на 1 поддерживающим ECN маршрутизатором при возникновении в сети насыщения. Таким образом, предполагается, что флаг CE может быть в любой момент установлен в 1 в зависимости от состояния насыщения сети и местоположения системы компрессии на пути. Следовательно, система компрессии, расположенная близко к перегруженной сети часто будет видеть установленных флаг CE, а система сжатия, расположенная вблизи отправителя будет редко (или не будет совсем) видеть CE = 1.

Недавние экспериментальные предложения [19] используют иную трактовку этих битов, рассматривая их совместно как код, способный принимать 4 значения:

00 - ECN не поддерживается

01 - поддерживается ECN, насыщение отсутствует (обозначается также ECT(0))

10 - поддерживается ECN, насыщение отсутствует (обозначается также ECT(1))

11 - присутствует насыщение.

Использование двух кодов ECT позволяет отправителю детектировать ситуации, когда получатель не может надежно отвечать на информацию о перегрузке.

С точки зрения компрессии это изменение означает, что значения ECT(0) и ECT(1) равновероятны (для поддерживающего ECN потока), а значение 11 будет встречаться достаточно редко. Вероятность увидеть индикацию насыщения обсуждалась выше при описании флага CE.

В целях сжатия заголовков предлагается рассматривать значения ECN nonce в качестве базовых, хотя схема компрессии должна обеспечивать прозрачную компрессию для исходной схемы использования ECN.

4.1.3. Идентификация IP

Поле Identification (IP ID) в заголовке IPv4 показывает какой фрагмент содержит дейтаграмма при сборке фрагментов пакета. Спецификация IPv4 не задает порядок присвоения значений идентификатора, указывая лишь, что каждому пакету следует давать уникальное для пары отправитель-получатель и используемого протокола значение IP ID. Уникальность должна обеспечиваться в течение интервала времени, когда дейтаграмма (или любой из ее фрагментов) может находиться в сети. Это означает, что присвоение значений IP ID может выполняться различными путями, которые мы будем делить на три класса:

- **Sequential jump** - нарастание

  Это наиболее распространенная политика присвоения идентификаторов в современных вариантах стека IP. Один счетчик IP ID используется для всех потоков пакетов. Когда отправитель работает с множеством потоков одновременно, значение IP ID между последовательными пакетами одного потока может увеличиваться более, чем на 1. Значения IP ID являются легко предсказуемыми, для их передачи требуется меньшее количество битов, а увеличение идентификатора от пакета к пакету (определяется числом активных потоков исходящих пакетов и частотой передачи) обычно бывает ограничено.

- **Random** - случайные значения

  Некоторые варианты стека IP используют в качестве IP ID псевдослучайные значения. В этом случае отсутствуют корреляции между значениями ID в последовательных пакетах. Следовательно, отсутствует возможность предсказать значение IP ID для следующей дейтаграммы. С точки зрения компрессии заголовков это означает, что поле IP ID нужно передавать без сжатия, что приводит к использованию двух лишних октетов[3] в сжатом заголовке. Реализациям стека IP в терминалах сотовых сетей требуется эффективная компрессия, поэтому им не следует использовать данный вариант задания значений IP ID.

- **Sequential** - равномерное увеличение

  В этом варианте используется отдельный счетчик для каждого исходящего потока пакетов и значения IP ID будут увеличиваться на единицу для каждого следующего пакета (за исключением случаев достижения максимума и возврата к началу отсчета). Следовательно, изменение значения этого поля постоянно и заранее известно. Такой

[1]Поддерживающий ECN транспорт
[2]Наблюдается насыщение (перегрузка).
[3]Это не совсем точно, поскольку даже при использовании компрессии для передачи идентификаторов используется отличное от нуля количество битов сжатого заголовка. *Прим. перев.*

вариант присвоения идентификаторов наиболее подходит для сжатия заголовков. Однако его использование не столь широко, как следовало бы ожидать[1].

Во избежание нарушения требований спецификации RFC 791 [1], пакеты, использующие одну пару адресов «отправитель-получатель» и номер протокола IP, не могут иметь одинаковых значений IP ID. Следовательно, при выделении значений идентификатора по порядку нужно разделять значения идентификаторов для разных потоков одного протокола между одной парой конечных точек. Это можно сделать разными путями, каждый из которых использует случайные пропуски в порядке нумерации, что делает распределение идентификаторов не вполне последовательным. С точки зрения компрессии желательно реже использовать пропуски в значениях идентификаторов.

Отметим, что эти идентификаторы используются только в IPv4 и, следовательно, не рассматриваются в IPv6. Для IPv4 поля ID могут обрабатываться тремя разными способами. Во-первых, имеется малоэффективное, но надежное решение, в котором поле идентификации передается без изменений во всех пакетах, что ведет к увеличению сжатых заголовков на 2 октета[3]. Этот способ является наилучшим вариантом обработки полей ID в тех случаях, когда отправитель использует случайные значения ID. Во-вторых, может использоваться более гибкий механизм, который обеспечит снижение числа битов для полей ID при нарастающей нумерации. Такие механизмы могут в некоторых случаях увеличивать число требуемых битов[2] при использовании отправителем случайных значений идентификаторов. Следовательно, информация об используемом отправителем механизме выделения идентификаторов полезна при выборе между двумя упомянутыми выше решениями. Наконец, даже для IPv4 могут быть созданы схемы компрессии, не включающие в сжатый заголовок никакой дополнительной информации о поле ID. Для использования таких схем нужно знать какой из механизмов выделения значений поля ID применяется отправителем. Получение такой информации возможно не во всех случаях, что ведет к весьма редкому применению таких механизмов. Однако разработчикам стеков IPv4 для терминалов сотовых сетей следует использовать политику выделения идентификаторов, близкую к последовательной.

В контексте компрессии TCP поведение поля IP ID остается таким же. Однако в RFC 3095 [31] значения IP ID в общем случае получаются из порядковых номеров RTP. Для случая TCP нет подходящего кандидата на роль «первичного порядкового номера».

Очевидно, что наблюдаемое поведение загруженного сервера может быть достаточно изменчивым. В некоторых случаях возможность совместного использования контекста компрессии IP для множества потоков (между одной парой конечных точек) может давать некоторые преимущества. Однако реальное воздействие может наблюдаться лишь в случаях использования большого числа потоков между конкретной машиной и сервером. Если рассматривается совместное использование контекста, предпочтительным будет совместное использование связанной с IP части контекста.

4.1.4. Флаг запрета фрагментации (DF)

Механизм Path-MTU discovery (RFC 1191 для IPv4 [6] и RFC 1981 для IPv6 [11]) широко используется для протокола TCP, но весьма редко применяется сегодня для потоков пакетов UDP. Этот механизм использует флаг DF = 1 для определения необходимости фрагментирования на сквозном пути и нахождения минимального значения MTU вдоль этого пути. Можно предполагать, что конечные хосты, использующие этот механизм, будут передавать все пакеты с DF = 1, хотя хост может прекратить использование PMTU, установив DF = 0. С точки зрения компрессии мы считаем значение этого флага стабильным.

4.1.5. IP Hop-Limit / Time-To-Live (TTL)

Поля Hop-Limit (IPv6) и Time-To-Live (IPv4) предполагаются постоянными в течение всего срока существования потока пакетов или чередующими значения из небольшого набора при изменении маршрутов.

4.2. Заголовок TCP

Обсуждение сжимаемости полей TCP заимствовано из RFC 1144 [22]. Однако условия компрессии несколько отличаются, а используемые протоколы изменились.

4.2.1. Порядковый номер

Понимание поведения порядковых номеров и номеров подтверждений имеет важное значение для схем сжатия TCP.

На простейшем уровне поведение порядковых номеров можно объяснить достаточно легко. Однако имеется ряд осложняющих факторов, которые также следует рассмотреть.

Для последовательно передаваемых пакетов порядковые номера будут увеличиваться для каждого пакета на величину от 0 до верхнего предела, определяемого значением MSS[3] и, если этот механизм используется, Path-MTU discovery.

Существуют общепринятые значения MSS, но они могут изменяться в значительных пределах и непредсказуемыми для любого конкретного потока. С учетом этого факта и широкого диапазона размеров окна для представления порядковых номеров сложно (по сравнению с RTP, например) выбрать «одно решение на все случаи жизни».

Отметим, что увеличение порядковых номеров пакетов определяется размером данных в этом пакете (включая флаги SYN и FIN). Естественно, существуют точные соотношения, которые RFC 1144 [22] использует для сжатия порядковых номеров в наиболее эффективном случае. Этот метод не применим напрямую для устойчивых к ошибкам систем, но рассмотреть его полезно.

Однако в любой точке пути (например, там, где может быть развернута система компрессии) порядковые номера могут оказаться в любой точке окна TCP. Это зависит от множества параметров (размер буфера на стороне отправителя,

[1] В точки зрения безопасности это вариант выделения идентификаторов является наименее предпочтительным, поэтому не следует удивляться тому, что он не используется повсеместно. *Прим. перев.*
[2] Сверх обычных двух октетов. *Прим. перев.*
[3] Maximum Segment Size - максимальный размер сегмента.

анонсированный получателем размер буфера, алгоритм контроля насыщения TCP). Потеря пакетов или повтор передачи также могут вызывать флуктуации порядковых номеров в пределах этого окна.

Желательно иметь возможность предсказания порядковых с некоторой регулярностью. Однако сделать это сложно. Например, при передаче большого объема данных порядковые номера имеют тенденцию увеличения на величину MSS для каждого пакета (если предположить отсутствие потерь). Параметры вышележащих уровней также могут оказывать влияние поведение порядковых номеров может повторяться с интервалом 8 кбайт (5 сегментов по 1460 байтов), за которыми следует 1 сегмент размером 892 байта. Реализация TCP и управления буферами в стеке протоколов могут воздействовать на поведение порядковых номеров.

Система компрессии может отслеживать размер окна TCP, что позволяет ограничить размер этих переходов.

Для интерактивных потоков (например, telnet), порядковые номера будут увеличиваться на небольшие и нерегулярные значения. В этом случае обычно применим алгоритм Nagle [3], объединяющий по возможности мелкие пакеты для снижения объема служебного трафика (заголовков). Это может также вести к тому, что предсказать изменение порядковых номеров станет сложнее. Алгоритм Nagle предназначен для оптимизации и его не требуется использовать (приложения могут отключать этот алгоритм). Однако он включен по умолчанию во всех популярных реализациях TCP.

Отметим, что флаги SYN и FIN (которые будут подтверждаться) также занимают 1 байт пространства порядковых номеров.

4.2.2. Номер подтверждения

Большая часть информации, относящейся к порядковым номерам, применима и для номеров подтверждений. Однако имеются и некоторые важные отличия.

При передаче больших объемов данных обычно передается 1 подтверждение для каждой пары сегментов данных. Этот алгоритм описан в RFC 2581 [16]. Пакеты ACK не требуется передавать сразу же после получения сегмента данных, но подтверждения должны передаваться в течение 500 мсек и их следует генерировать по крайней мере для каждого второго полноразмерного (MSS) сегмента принятых данных. Может показаться, что увеличение номеров подтверждений приблизительно вдвое больше увеличения порядковых номеров. Однако это верно не во всех случаях и нерегулярность изменения порядковых номеров должна приниматься во внимание.

Отметим также распространенную ошибку реализаций stretch ACKs [33] (подтверждения могут генерироваться реже, чем для каждого второго полноразмерного сегмента данных). Такая же картина может возникать при потерях на пути возврата подтверждений.

Поскольку номера подтверждений являются кумулятивными, отбрасывание пакетов на прямом пути будет приводить к тому, что номер подтверждения в течение некоторого времени не будет изменяться для обратного направления. Повторная передача отброшенных сегментов будет приводить к существенному увеличения номера подтверждения. Размер этого увеличения ограничен окном TCP, как и для разового увеличения порядковых номеров. Для номеров подтверждений границу увеличения номера задает анонсируемый передающей стороной размер окна.

4.2.3. Резерв

Значение резервного поля должно быть нулевым. Но это поле может использоваться в будущем и делать предположения о его значении не следует.

4.2.4. Флаги

- **ECN-E** (Explicit Congestion Notification - явное уведомление о перегрузке)

  Флаг устанавливается (1), как эхо бита CE в заголовке IP. Это значение будет наблюдаться в нескольких последовательных заголовках (пока не будет подтвержден с помощью CWR). При использовании ECN nonce в этом флаге будет передаваться бит суммы nonce (NS). Как обычно, прозрачность резервного бита важна для работы схемы компрессии в будущем. С точки зрения соотношения эффективность/сжатие бит NS не будет использоваться (всегда 0) или будет произвольно изменяться. Сумма nonce представляет собой 1-битовое значение суммы кодов ECT, как описано в [19].

- **CWR** (Congestion Window Reduced - окно насыщения уменьшено)

  Флаг устанавливается для обозначения уменьшения размера окна насыщения в ответ на ECN. Этот флаг в общем случае устанавливается для отдельного пакета. Флаг устанавливается однократно в ответ на потерю пакета. Таким образом, вероятность установки этого флага пропорциональна степени насыщения сети, но очевидно меньше вероятности установки флага CE.

- **ECE** (Echo Congestion Experience - сигнал о возникновении перегрузки в сети)

  При получении в заголовке IP сигнала о перегрузке в сети, в ответ возвращается эхо-сигнал (бит ECE) в сегментах, передаваемых получателем до приема подтверждения о получении сигнала в виде установленного бита CWR. Очевидно, что в периоды насыщения и/или при большом значении RTT этот флаг часто будет иметь значение 1.

  При организации соединения (пакеты SYN и SYN/ACK) бит ECN имеет специальное значение:

  - Флаги CWR и ECN-E устанавливаются в пакетах SYN для индикации желания использовать ECN.
  - Флаг CWR устанавливается в пакетах SYN-ACK для подтверждения использования ECN.

    (различие в битовых последовательностях для согласования сделано для того, чтобы можно было работать со старыми стеками, не понимающими расширение).

- **URG** (Urgent Flag - флаг срочности)

  1 указывает на срочность данных (маловероятно использование этого флага с какими-либо флагами, кроме ACK).

- **ACK** (Acknowledgement - подтверждение)

1 во всех случаях, кроме стартового пакета SYN.

- **PSH** (Push Function Field - выталкивание данных)

  В общем случае произвольно меняется между 0 и 1. Однако одно из значений может встречаться чаще другого. В основном определяется используемым стеком протоколов.

- **RST** (Reset Connection - сброс соединения)

  1 устанавливается для сброса соединения (маловероятно использование этого флага вместе с какими-либо флагами, кроме ACK).

- **SYN** (Synchronize Sequence Number - синхронизация порядковых номеров)

  1 устанавливается для пакетов SYN/SYN-ACK на этапе организации соединения.

- **FIN** (End of Data: FINished - завершение передачи данных)

  1 показывает отсутствие данных для передачи (маловероятно использование этого флага вместе с какими-либо флагами, кроме ACK).

4.2.5. Контрольная сумма

Контрольная сумма служит для сквозного контроля отсутствия ошибок в данных TCP. Обсуждение вопросов работы с контрольными суммами содержится в RFC 1144 [22]. Схемам компрессии заголовков не следует полагаться на контрольную сумму TCP, хотя им следует применять свой подходящий механизм детектирования ошибок. Контрольные суммы TCP рассматриваются как произвольно изменяющиеся значения.

4.2.6. Окно (Window)

Размер окна может изменяться от 0 до установленного получателем предела (для данного соединения).

На практике размер окна сохраняется постоянным или чередуется небольшой набор значений. В частности, при снижении размера окна ясно, что это связано с размером сегмента, но какие-то преимущества с точки зрения компрессии не очевидны. При увеличении размера окна следует помнить об эффекте Silly-Window Syndrome[1], который может заставить отправителя передавать данные в очень мелких сегментах.

При обсуждении поведения полей в последовательности сегментов TCP следует отметить, что получатель может генерировать сегменты window update, в которых изменяется только анонсируемый размер окна.

4.2.7. Указатель срочности (Urgent)

С точки зрения компрессии поле Urgent Pointer представляет интерес по той причине, что оно служит показательным примером того, как «семантически идентичная» компрессия отличается от идентичной на битовом уровне. Это связано с тем, что поле указателя срочности имеет смысл только при установленном флаге URG.

Однако для обеспечения сквозного контроля целостности требуется прозрачная передача контрольной суммы TCP. Поскольку контрольная сумма учитывает поле Urgent Pointer, это требует обеспечивать побитовую идентичность для поля Urgent Pointer. Таким образом, поле Urgent Pointer требуется сжимать так, чтобы сохранялось его значение.

При установленном флаге URG поле Urgent Pointer указывает на окончание срочных данных, которое может находиться в любой точке окна. Срочные данные могут содержаться в нескольких сегментах и значение указателя срочности может изменяться. Отметим, что пользуются указателем срочности достаточно редко, поскольку он не обеспечивает эффективного способа управления данными на практике.

4.3. Опции

Опции размещаются в конце заголовка TCP и учитываются при вычислении контрольной суммы. Опция может начинаться на любой границе байта. Заголовок TCP должен дополняться нулями для выравнивания по 32-битовой границе.

Необязательные поля заголовка идентифицируются полем типа опции. Опции типа 0 и 1 занимают один октет. Все остальные опции имеют 1-октетное поле типа, за которым следует октет размера (length) и поле данных, размером length-2 октета.

4.3.1. Обзор опций

Тип	*Размер в октетах*	*Значение*	*RFC*	*Применение*
0	-	End of Option List	RFC 793	*
1	-	No-Operation	RFC 793	*
2	4	Maximum Segment Size	RFC 793	*
3	3	WSopt - Window Scale	RFC 1323	*
4	2	SACK Permitted	RFC 2018	*
5	N	SACK	RFC 2018	*
6	6	Echo (отменено опцией 8)	RFC 1072	
7	6	Echo Reply (отменено опцией 8)	RFC 1072	
8	10	TSopt - Time Stamp Option	RFC 1323	*
9	2	Partial Order Connection Permitted	RFC 1693	*
10	3	Partial Order Service Profile	RFC 1693	
11	6	CC	RFC 1644	
12	6	CC.NEW	RFC 1644	
13	6	CC.ECHO	RFC 1644	
14	3	Alternate Checksum Request	RFC 1146	

[1] Синдром тупого окна.

Тип	Размер в октетах	Значение	RFC	Применение
15	N	Alternate Checksum Data	RFC 1146	
16		Skeeter		
17		Bubba		
18	3	Trailer Checksum Option		
19	18	MD5 Signature Option	RFC 2385	
20		SCPS Capabilities		
21		Selective Negative Acks		
22		Record Boundaries		
23		Corruption experienced		
24		SNAP		
25		Unassigned (c 18.12.2000)		
26		TCP Compression Filter		

Рисунок 12. Опции TCP общего назначения.

Агентство IANA поддерживает официальный список определенных опций TCP. На рисунке 12 показан список опций, определенных на момент публикации документа. Любая опция имеет идентификатор типа, выделенный IANA. Список опций доступен на сайте [20]. В тех случаях, когда это применимо, список опций содержит ссылки на RFC.

Знак * в колонке «Применение» отмечает опции, которые чаще встречаются в потоках TCP. Отметим также, что RFC 1072 [4] был заменен RFC 1323 [7], хотя исходное использование битов определено в 1072.

4.3.2. Поведение поля опций

В общем случае все поля опций классифицируются как изменяющиеся. В этом параграфе рассматривается поведение каждой опции, определенное в RFC, указанном для данного типа опций.

0: End of Option List - конец списка опций

Этот тип указывает на завершение списка опций, которое может не совпадать с концом заголовка TCP, определяемым полем Data Offset. Эта опция указывается после завершения всех опций, а не какой-то отдельной опции, но ее необходимо использовать лишь в тех случаях, когда точки завершения заголовка TCP и списка опций не совпадают. Опция определена в RFC 793 [2].

С этой опцией не связано никаких данных, поэтому схеме компрессии достаточно просто закодировать присутствие опции. Однако следует помнить о возможном присутствии после опции битов заполнения для выравнивания заголовка TCP по 4-октетной границе (биты заполнения имеют значение 0 в соответствии с документом [2]).

1: No-Operation - нет операции

Эта опция может включаться между другими опциями заголовка (например, для выравнивания следующей опции по границе слова). Использование такого выравнивания на передающей стороне не гарантируется, поэтому получатель должен быть готов к обработке опций, которые начинаются не на границе слова (RFC 793 [2]). С этой опцией не связано никаких данных, поэтому схеме компрессии достаточно показать наличие опции. Это можно сделать путем обозначения наличия выравнивания и необходимости передачи информации о нем. В этом случае биты заполнения могут быть удалены.

2: Maximum Segment Size - максимальный размер сегмента

При наличии этой опции она определяет максимальный размер сегмента, который может быть передан данному конечному хосту. Этот поле следует устанавливать только в стартовом запросе на организацию соединения (т. е., в сегменте с флагом SYN). Если опция не используется, можно использовать сегменты любого размера в соответствии с RFC 793 [2].

Эта опция применяется очень часто. Размеры сегментов задаются с гранулярностью 16 битов. Теоретически может использоваться любой размер, однако на практике обычно применяется небольшой ряд значений. Например, значение 1460 байтов характерно для использования TCP/IPv4 в сетях Ethernet (хотя с ростом популярности туннелей все чаще используется значение 1400). По умолчанию для MSS используется значение 536. Распространенные значения поля могут кодироваться более эффективно.

3: Window Scale Option (Wsopt) - опция масштабирования окна

Эта опция может передаваться в сегменте SYN конечным хостом TCP для

(1) индикации передающему хосту TCP возможности масштабирования окон приема и передачи;

(2) индикации коэффициента масштабирования окна приема.

Коэффициент масштабирования окна задается в виде двоичного логарифма (возможно[1] потому, что используется битовый сдвиг). Отметим, что окно в самом сегменте SYN никогда не масштабируется (RFC 1072 [4]). Данная опция может передаваться в начальном сегменте (т. е., в сегменте с флагом SYN, но без флага ACK). Возможно использование опции и в последующих сегментах, но только при условии получения опции Window Scale в начальном сегменте. Опции Window Scale в сегментах без флага SYN следует игнорировать. Поле Window в самом сегменте SYN никогда не масштабируется (RFC 1323 [7]).

Масштабирование окна не влияет на кодирование других полей в течение срока существования соединения. Важно лишь кодирование самой опции масштабирования окна. Значения масштабирования размера окна должны находиться в диапазоне от 0 до 14 (включительно). В общем случае следует ожидать не слишком больших значений, поскольку 14 соответствует размеру окна 1 Гбайт (слишком много).

4: SACK-Permitted - возможность использования SACK

[1] В RFC 1323, определяющем это расширение явно указано, что логарифм используется для того, чтобы размер окна можно было получать с помощью операции сдвига. *Прим. перев.*

Эта опция может передаваться при организации соединения в пакетах SYN узлами TCP, которые готовы принимать (и, вероятно, обрабатывать) опцию SACK (RFC 2018 [12]). Опция не включает каких-либо данных, которые нужно было бы сохранять при ее кодировании.

5: SACK - частичное подтверждение

Эта опция может использоваться для передачи в существующих соединениях расширенных подтверждений доставки. В частности, получатель блоков данных с нарушением порядка может с помощью этой опции уведомить отправителя о приеме и буферизации таких блоков. Получатель ждет приема недостающих блоков в повторных пакетах для заполнения имеющихся в данных пропусков. В это же время получатель подтверждает данные, обычно указывая левый край окна в поле Acknowledgment Number заголовка TCP. Важно понимать, что опция SACK не меняет трактовку поля Acknowledgment Number, значение которого по прежнему указывает на левый край окна (т. е., на один байт больше порядкового номера последних данных, которые получены полностью RFC 2018 [12]).

Если использование SACK согласовано сторонами (путем обмена опциями SACK-Permitted), данная опция может применяться в тех случаях, когда получатель уведомляет о потерянных сегментах. Поскольку опция идентифицирует диапазоны блоков в окне приема, она может рассматриваться как базовое значение и набор значений смещения. Базовое значение (левый край первого блока) можно рассматривать как смещение от номера подтверждения TCP. В одной опции может указываться до 4 блоков SACK. Блоки SACK могут возникать во многих сегментах (если в сети часто нарушается порядок доставки) и будут, таким образом, расширять размер уже указанных блоков или добавлять новые блоки.

Дополнительные расширения типа DSACK RFC 2883 [17] не меняют существенно поведения блоков SACK в плане содержащейся в этих блоках информации.

6: Echo - эхо

Эта опция содержит информацию, которую принимающая сторона TCP может вернуть в последующей опции TCP Echo Reply (см. ниже). TCP может передать опцию TCP Echo в любом сегменте при условии, что в сегменте SYN при организации соединения была получена опция TCP Echo. При использовании опции TCP echo для измерения RTT эта опция включается в сегменты данных, а четыре информационных байта будут определять время передачи сегмента данных в удобном для отправителя формате (см. RFC 1072 [4]).

Опция Echo обычно не используется на практике, поскольку она заменена опцией Timestamp. Однако для обеспечения прозрачности схемам компрессии следует обеспечивать возможность поддержки этой опции. В результате какой-то специальной трактовки этой опции (отличной от «стандартного» для опций подхода) не возникает никаких дополнительных преимуществ .

7: Echo Reply - эхо-отклик

Модуль TCP, получивший опцию TCP Echo, содержащую 4 байта данных, будет возвращать эти байты в опции TCP Echo Reply. Опция TCP Echo Reply должна возвращаться при передаче следующего сегмента (например, ACK). Если до отправки следующего сегмента будут получены новые опции Echo, модуль TCP должен выбрать только одну из них, игнорируя остальные (должен выбираться самый новый сегмент с самым старым порядковым номером - RFC 1072 [4]).

Опция Echo Reply обычно не используется на практике, поскольку она заменена опцией Timestamp. Однако для обеспечения прозрачности схемам компрессии следует обеспечивать возможность поддержки этой опции. В результате какой-то специальной трактовки этой опции (отличной от «стандартного» для опций подхода) не возникает никаких дополнительных преимуществ .

8: Timestamps - временные метки

Эта опция передает два 4-байтовых поля с временными метками. Поле Timestamp Value (TSval) содержит текущее значение временной метки передавшего опцию модуля TCP. Поле Timestamp Echo Reply (TSecr) имеет смысл только при наличии в заголовке TCP флага ACK и в этом случае содержит временную метку, переданную удаленным модулем TCP в поле TSval опции Timestamps. Когда поле TSecr не имеет смысла, оно должно содержать 0. В общем случае значение TSecr берется из последней принятой опции Timestamp, однако ниже рассмотрено несколько исключений из этого правила. TCP может включать опцию Timestamps (TSopt) в начальный сегмент (сегмент с флагом SYN, но без флага ACK), а также может передавать TSopt в других сегментах, если в начальном сегменте соединения было получено поле TSopt (см. RFC 1323 [7]).

Временные метки используются достаточно часто. Если эта опция используется на этапе организации соединения, следует ожидать ее появления и во всех последующих сегментах. Если в стартовых сегментах опции не было, она не может присутствовать в других сегментах данного потока.

Поскольку поле содержит значение временной метки, логично предположить, что его не требуется передавать целиком. Однако картина изменения временных меток может быть различной.

Одной из важных причин использования временных меток является детектирование повторного использования порядковых номеров (механизм PAWS[1], см. RFC 1323 [7]). Для соединений, использующих компрессию заголовков TCP, такая угроза обычно отсутствует, но важно обеспечить целостность опции. Этот вопрос рассматривается в документе RFC 3150 [32]. Предложенный алгоритм Эйфеля (Eifel) [35] рекомендует использовать временные метки на каналах сотовых сетей [34].

В части компрессии отметим, что диапазон разрешения для временных меток, предложенный в RFC 1323 [7] достаточно широк (от 1 мсек до 1 сек на «тик»). Вкупе с возможностью значительных вариаций RTT это осложняет выбор кодирования, которое было бы оптимальным для всех случаев.

9: Partial Order Connection (POC) permitted - возможность неполного упорядочивания

Эта опция является простым индикатором, с помощью которого обе стороны соединения могут согласовать использование протокола POC (см. RFC 1693 [9]).

[1]Protection Against Wrapped Sequence-number - защита от повтора порядковых номеров.

Соединения с неполным упорядочиванием редко используются (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

10: POC service profile - профиль POC

Эта опция служит для обмена информацией, обеспечивающей работу протокола (обычная информация из заголовков TCP). Соединения с неполным упорядочиванием редко используются (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

11: Connection Count (CC) - счетчик соединений

Эта опция является частью реализации TAO (TCP Accelerated Open), которая позволяет обойтись без трехэтапного согласования TCP Three-Way Handshake (3WHS). TAO использует 32-битовый номер инкарнации, называемый счетчиком соединений (connection count или CC), который передается в опции TCP каждого сегмента. Для каждого из направлений соединения TCP используется свое значение CC. Реализации присваивают монотонно возрастающие значения CC для последующих соединений, которые открываются активно или пассивно (см. RFC 1644 [8]). Эта опция редко используется (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

12: CC.NEW

Для корректной работы механизма TAO от клиентов требуется генерация монотонно возрастающих значений CC при успешной организации соединений. Получение опции CC.NEW заставляет сервер объявить некорректной кэшированную запись и выполнить процедуру 3WHS (см. RFC 1644 [8]). Эта опция редко используется (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

13: CC.ECHO

Когда сервер передает сегмент, он возвращает значение счетчика соединений из начальной опции CC.ECHO, которое используется клиентом для проверки корректности сегмента (см. RFC 1644 [8]). Эта опция редко используется (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

14: Alternate Checksum Request - запрос на изменение алгоритма расчета контрольной суммы

Эта опция может быть передана в сегменте SYN для индикации того, что модуль TCP готов генерировать и принимать контрольные суммы, рассчитанные с использованием альтернативного алгоритма. В течение срока существования соединения такие суммы будут использоваться в заголовках TCP вместо обычных контрольных сумм TCP. Если нестандартная контрольная сумма имеет размер более 2 октетов, она может полностью перемещаться в поле опции TCP Alternate Checksum Data Option с установкой нулевого значения в поле контрольной суммы заголовка TCP или указываться по частям в поле заголовка и опции. Для расчета контрольный суммы по иному алгоритму используется тот же набор данных, что и для обычных контрольных сумм TCP (см. RFC 1146 [5]).

Эта опция редко используется (возможно, не используются совсем) в современной сети Internet, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции. Опция может использоваться только на этапе организации соединения (SYN-пакеты). Даже при наличии этой опции она не будет оказывать влияния на обработку контрольной суммы, поскольку для нее должна обеспечиваться прозрачность во всех случаях.

15: Alternate Checksum Data - альтернативная контрольная сумма

Это поле используется лишь в тех случаях, когда согласованная контрольная сумма имеет размер более 16 битов. Такие суммы не помещаются в стандартное поле заголовка TCP и должны, по крайней мере, частично, указываться в данной опции. Расщепление контрольной суммы между заголовком TCP и данной опцией или включение всей контрольной суммы в поле опции определяется независимо для каждой контрольной суммы. Размер этой опции будет зависеть от выбранного механизма генерации контрольных сумм для данного соединения (см. RFC 1146 [5]).

Если использование нестандартного алгоритма подсчета контрольных сумм согласовано при организации соединения, эта опция может появляться во всех последующих пакетах (если она требуется). Однако данная опция на практике почти не используется, поэтому единственным требованием к схеме компрессии является поддержка кодирования этой опции.

16 - 18:

Эти опции, не включенные в RFC, не рассматриваются в данном документе.

19: MD5 Digest - сигнатура MD5

Каждый сегмент, передаваемый в соединение TCP, будет защищаться от подмены путем включения в него 16-битовой сигнатуры MD5, получаемой с помощью алгоритма MD5 для собранного вместе блока данных [13].

При получении подписанного сегмента получатель должен проверить его путем расчета сигнатуры для тех же данных (с использованием своего ключа) и сравнения с полученной сигнатурой. При обнаружении различий сегмент должен отбрасываться без передачи серверу какого-либо отклика на него. Разумно также сохранять информацию о таких событиях в системном журнале.

В отличие от других расширений TCP (например, от опции Window Scale [7]), отсутствие данной опции в сегменте SYN-ACK не должно заставлять отправителя запрещать использование подписей в своих сегментах. Такое согласование обычно делается для предотвращения некорректного поведения некоторых реализаций TCP при получении опции в отличных от SYN сегментах. Это не вызывает проблем для данной опции, поскольку сегменты SYN-ACK, передаваемые в процессе организации соединений, не будут подписаны и, следовательно, будут

игнорироваться. В результате этого соединение не будет организовано и опция просто не может появиться в сегментах без флага SYN. Более важно то, что передача сигнатур должна вестись под полным контролем приложения, а не по милости удаленного хоста, который не понимает эту опцию. Сигнатуры MD5 следует, подобно любым криптографически защищенным данным, передавать без использования компрессии. Следовательно, схема компрессии должна просто обеспечивать прозрачную передачу этой опции.

20 - 26;

Эти опции, не включенные в RFC, не рассматриваются в данном документе. Это означает, что детали их поведения не описываются и не ожидается, что схема компрессии оптимизирована для работы с такими опциями. Однако любые нераспознанные опции должны прозрачно передаваться схемой компрессии TCP для обеспечения эффективной работы с редкими и новыми опциями.

Приведенный выше список включает опции, известные на момент создания этого документа. Предполагается, что могут быть определены другие опции. Важно обеспечить обработку будущих опций системой компрессии заголовков. Обработка неизвестных пока опций не может быть оптимизирована, но они, по крайней мере, должны прозрачно передаваться.

Современная модель использования опций TCP включает согласование опций в процессе обмена сегментами SYN и последующее использование согласованных сторонами опций. Это ведет к возможности некоторых допущений о присутствии тех или иных опций (только в пакетах с флагом SYN, в каждом пакете и т. п.). Когда такие допущения корректны, они помогают несколько оптимизировать сжатие заголовков. Однако такие допущения нежелательны, поскольку нет гарантии, что обработка и согласование опций не изменится в будущем. Отметим также, что система компрессии может не обрабатывать SYN-пакеты потока и, следовательно, не знать о том, какие опции согласованы для использования.

5. Другие наблюдения

5.1. Неявные подтверждения

Для потоков TCP существует небольшое число намеков на неявные подтверждения. Даже если система компрессии видит только одно направление потока TCP, она может судить о получении пакета SYN просто по наличию передаваемых пакетов без флага SYN.

Для систем компрессии существует четкое требование независимости от топологии. Это означает невозможность на практике гарантировать, что просмотр пакетов данных на уровне системы компрессии позволяет гарантированно судить о получении пакета SYN системой декомпрессии (поскольку пакет SYN может идти по иному пути).

Однако существуют некоторые косвенные признаки, которые могут использоваться в определенных обстоятельствах для повышение эффективности компрессии.

5.2. Совместно используемые данные

Представляется разумным рассмотреть два случая - (i) когда прямой (данные) и обратный (ACK) пути проходят по общему каналу и (ii) когда прямой (данные) и обратный (ACK) пути проходят по разным каналам, каждый из которых используется только для определенного направления.

В первом случае компрессор и декомпрессор могут располагаться в одном месте. Тогда компрессор и декомпрессор на каждой стороне соединения могут обмениваться информацией. Такой обмен может повысить уровень эффективности.

Например, номера подтверждений обычно берутся из порядковых номеров противоположного направления. Поскольку подтверждение не может генерироваться для пакета, который еще не прошел через канал, это обеспечивает эффективный способ кодирования подтверждений.

5.3. Доля заголовков TCP

При передаче больших объемов данных TCP доля заголовков TCP невелика по сравнению с общим размером пакетов (например, < 3% для пакетов размером 1460 октетов) и сравнима с долей заголовков в типичном голосовом потоке RTP. Очевидно, что спектральная эффективность является важной целью. Однако «выжимание последних битов» при компрессии дает незначительное повышение эффективности при существенном росте сложности. При создании профилей компрессии TCP нужно искать компромисс между эффективностью и сложностью.

Однако в направлении подтверждений (т. е., для заголовков «чистых» подтверждений) долю заголовков можно считать бесконечной, поскольку объем передаваемых данных равен нулю. Поэтому оптимизация для пути подтверждений может считаться полезной.

Существует множество схем работы с подтверждениями TCP, позволяющих снизить расход полосы на передачу ACK. Многие из таких схем описаны в документах [33] и [32]. Большинство этих схем полностью совместимо с компрессией заголовков без каких-либо дополнительных усилий. Хотя и не предполагается оптимизация схем компрессии для экспериментальных опций, полезно учитывать эти опции при разработке схем компрессии (и наоборот, учитывать схемы компрессии при создании новых опций). Схема компрессии заголовков должна быть способна поддерживать любые опции, включая те, которые еще не определены.

5.4. Независимость полей и поведение пакетов

Ясно, что прямое сравнение с сильно ориентированной на пакеты компрессией RTP достаточно сложно. Поля заголовков RTP имеют тенденцию регулярных изменений от пакета к пакету, а многие поля (например, IPv4 IP ID, порядковый номер RTP, временные метки RTP) изменяются в зависимости одно от другого. Однако поля TCP (такие, как порядковый номер) менее предсказуемы, отчасти в результате влияния внешних факторов (размеры окна TCP, поведение приложение и т. п.). Значения полей изменяются независимо. Все это в целом дает дополнительные стимулы для выполнения компрессии и осложняет выбор набора вариантов кодирования, который обеспечит эффективность в сочетании с устойчивостью к ошибкам.

5.5. Короткоживущие потоки

Сложно предположить, как можно повысить производительность для отдельного, непредсказуемого и короткоживущего соединения. Однако существует множество типовых ситуаций, когда организуется множество соединений TCP между одной парой хостов. Одним из таких примеров может служить web-серфинг (это более относится к HTTP/1.0 [25], нежели к HTTP/1.1 [26]).

Когда соединение закрывается, оно может быть последним между данной парой хостов, но чаще в течение сравнительно короткого времени создается новое соединение. В этом случае связанная с заголовком IP часть контекста (т. е. поля, описанные в параграфе 2.1) будет сохраняться почти без изменений. Некоторые аспекты контекста TCP также сохранят сходство.

Поддержка репликации более детально рассматривалась в главе 3. В целом, поддержка совместного использования части контекста или генерация одного контекста из другого обеспечивает оптимизацию для последовательности соединений с небольшим сроком существования.

Отметим, что несмотря на то, что протокол TCP использует прямые соединения, компрессору сложно сказать, когда завершится поток TCP. Например, даже для «двухстороннего» соединения получение подтверждение (ACK) пакета FIN в компрессоре/декомпрессоре не означает, что не может быть повтора передачи FIN. Поэтому может оказаться более полезной инициализация нового контекста на основе существующего, нежели повторное использование имеющегося контекста.

Как было отмечено в параграфе 4.1.3, заголовок IP может совместно использоваться множеством транспортных потоков между одной парой конечных точек. Это может быть использовано при инициализации новых заголовков TCP из существующих заголовков. Начинают обычно с номеров портов.

5.6. Первичный порядковый номер

Как было отмечено в параграфе 4.1.3, в TCP не существует очевидного кандидата на роль «первичного порядкового номера». Более того, отмечено, что такой «первичный» номер требуется только для того, чтобы позволить декомпрессору подтверждать пакеты в двухстороннем режиме. Понятно, также, что такой порядковый номер не будет требоваться для каждого пакета.

Хотя пространство порядковых номеров и представляется обширным, очевидно, что имеется необходимость в его расширении. Не существует очевидного способа обеспечить гарантию уникальности пакетов без такого расширения пространства номеров (порядковые номера и номера подтверждений могут по-прежнему использовать общее пространство).

5.7. Ограничение размера опций TCP

Как можно видеть из приведенного выше обсуждения, большинство опций TCP, таких, как MSS, Wsopt или SACK-Permitted может появляться только в сегментах SYN. Каждой реализации следует (и предполагается, что это выполняется на практике) игнорировать неизвестные опции в сегментах SYN. Опции TCP будут передаваться в сегментах без флага SYN лишь в тех случаях, когда обмен опциями в SYN-сегментах показал, что обе стороны понимают данное расширение. Другие опции TCP, такие, как MD5 Digest или Timestamp, также обычно передаются в процессе организации соединений (т. е. в пакетах SYN).

Общий размер заголовка также является предметом обсуждения. Заголовок TCP показывает начало сегмента данных с помощью 4-битового поля, определяющего общий (с учетом опций) размер заголовка в 32-битовых словах. Это означает, что общий размер заголовка и опций не может превышать 60 байтов (т. е., на опции остается не более 40 байтов).

6. Вопросы безопасности

Поскольку этот документ лишь описывает поведение полей TCP и не создает новых проблем безопасности.

Документ предназначен для использования в целях компрессии заголовков TCP/IP. При работе с механизмами аутентификации типа IPsec AH [24] важно обеспечить прозрачность сжатия. При использовании шифрования (например, IPsec ESP [27]) поля TCP могут стать невидимыми, что будет препятствовать их сжатию.

7. Благодарности

Множество посвященных протоколам IP и TCP документов RFC (надеемся, что ниже перечислены все эти документы), вместе с посвященными схемам компрессии RFC 1144 [22], RFC 3544 [36] и RFC 3095 [31], а также подробный анализ RTP/UDP/IP в RFC 3095 послужили источниками идей и информации для этой работы. Дополнительная информация по основам компресии содержится также в документах [28] и [29].

Этот документ основан также на обсуждениях в почтовой конференции ROHC и различных коридорах (виртуальных и иных) множества ключевых вопросов. Особая благодарность Qian Zhang, Carsten Bormann и Gorry Fairhurst.

Qian Zhang и Hongbin Liao внесли свой вклад в анализ используемых совместно полей заголовков.

Все ошибки и погрешности представления и интерпретации остаются на совести авторов этого документа.

8. Литература

8.1. Нормативные документы

[1] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[2] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[3] Nagle, J., "Congestion control in IP/TCP internetworks", RFC 896, January 1984.

[4] Jacobson, V. and R. Braden, "TCP extensions for long-delay paths", RFC 1072, October 1988.

[5] Zweig, J. and C. Partridge, "TCP alternate checksum options", RFC 1146, March 1990.

[6] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[7] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[8] Braden, B., "T/TCP -- TCP Extensions for Transactions Functional Specification", RFC 1644, July 1994.

[9] Connolly, T., Amer, P., and P. Conrad, "An Extension to TCP: Partial Order Service", RFC 1693, November 1994.

[10] Bellovin, S., "Defending Against Sequence Number Attacks", RFC 1948, May 1996.

[11] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[12] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, October 1996.

[13] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[14] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[15] Ramakrishnan, K. and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.

[16] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[17] Floyd, S., Mahdavi, J., Mathis, M., and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", RFC 2883, July 2000.

[18] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[19] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

8.2. Дополнительная литература

[20] IANA, "IANA", IANA TCP options, February 1998, <http://www.iana.org/assignments/tcp-parameters>.

[21] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

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

[23] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.

[24] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[25] Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.

[27] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[26] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

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

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

[30] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[31] 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 Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[32] Dawkins, S., Montenegro, G., Kojo, M., and V. Magret, "End-to-end Performance Implications of Slow Links", BCP 48, RFC 3150, July 2001.

[33] Balakrishnan, Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", RFC 3449, December 2002.

[34] Inamura, H., Montenegro, G., Ludwig, R., Gurtov, A., and F. Khafizov, "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks", RFC 3481, February 2003.

[35] Ludwig, R. and M. Meyer, "The Eifel Detection Algorithm for TCP", RFC 3522, April 2003.

[36] Engan, M., Casner, S., Bormann, C., and T. Koren, "IP Header Compression over PPP", RFC 3544, July 2003.

[37] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

Адреса авторов

Mark A. West

Siemens/Roke Manor Research

Roke Manor Research Ltd.

Romsey, Hants SO51 0ZN

UK

Phone: +44 (0)1794 833311

EMail: mark.a.west@roke.co.uk

URI: http://www.roke.co.uk

Stephen McCann

Siemens/Roke Manor Research

Roke Manor Research Ltd.

Romsey, Hants SO51 0ZN

UK

Phone: +44 (0)1794 833341

EMail: stephen.mccann@roke.co.uk

URI: http://www.roke.co.uk

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Интеллектуальная собственность

Подтверждение

Финансирование функций RFC Editor обеспечено IETF Administrative Support Activity (IASA).