Network Working Group
Request for Comments: 2883
Category: Standards Track

S. Floyd
ACIRI
J. Mahdavi
Novell
M. Mathis
Pittsburgh Supercomputing Center
M. Podolsky
UC Berkeley

July 2000

Расширение для опции селективных подтверждений (SACK) TCP

An Extension to the Selective Acknowledgement (SACK) Option for TCP

Статус документа

Документ содержит спецификацию стандартного протокола для сообщества Internet и является приглашением к дискуссии в целях развития и совершенствования протокола. Сведения о стандартизации и состоянии данного протокола можно найти в документе «Internet Official Protocol Standards» (STD 1). Допускается свободное распространение данного документа.

Авторские права

Тезисы

В этом документе дано определение расширения опции SACK[1] [RFC2018] для протокола TCP. В RFC 2018 определено применение опции SACK для подтверждения данных, доставленных с нарушением порядка и не покрываемых полем кумулятивного подтверждения TCP. Этот документ служит расширением RFC 2018, задавая применение опции SACK для подтверждения пакетов-дубликатов. Данный документ предлагает при получении дубликатов использовать первый блок поля опции SACK для передачи порядковых номеров пакетов, которые вызвали это подтверждение. Такое расширение использования опции SACK позволяет отправителю узнать о порядке доставки пакетов на приемную сторону и идентифицировать ненужные повторы передачи. Отправитель TCP может использовать полученную информацию для повышения эффективности работы в средах с нарушением порядка доставки [BPS99], потерей подтверждений ACK, репликацией пакетов или слишком ранним повтором передачи.

1. Уровни требований

Ключевые слова "MUST" (**необходимо**), "MUST NOT" (**недопустимо**), "REQUIRED" (**требуется**), "SHALL" (**следует**), "SHALL NOT" (**не следует**), "SHOULD" (**следует**), "SHOULD NOT" (**не следует**), "RECOMMENDED" (**рекомендуется**), "MAY" (**возможно**) и "OPTIONAL" (**необязательно**) в данном документе трактуются в соответствии с [B97].

2. Введение

Опция селективных подтверждений (SACK), определенная в RFC 2018, используется при передаче данных по протоколу TCP для подтверждения блоков данных с разрывами (пропусками), которые не покрываются полем кумулятивного подтверждения (Cumulative Acknowledgement). Однако в RFC 2018 не рассматривается использование опции SACK для случая приема дубликатов сегментов. В данном документе рассмотрено использование опции SACK при подтверждении пакетов-дублей [F99]. Для обозначения блока SACK с информацией о дубликате сегмента используется аббревиатура D-SACK (duplicate-SACK).

Данный документ ни коим образом не меняет использования поля кумулятивных подтверждений TCP и не влияет на принятие получателем TCP решения о моменте передачи пакетов с подтверждением. Данный документ относится только к содержимому опции SACK при передаче подтверждений.

Данное расширение совместимо с имеющимися реализациями опции SACK в TCP. Т. е., если один из узлов в соединении TCP не поддерживает расширение D-SACK, а другой поддерживает, предполагается, что использование одним из узлов TCP расширения D-SACK не будет создавать проблем.

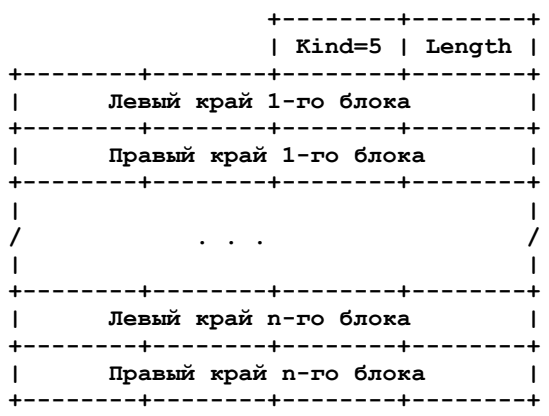
Использование опции D-SACK не требует дополнительного согласования между отправителем и получателем TCP, которые уже согласовали между собой использование SACK. Отсутствие отдельного согласования D-SACK означает, что получатель TCP может передавать блоки D-SACK даже в тех случаях, когда отправитель TCP не понимает данного расширения SACK. В таких случаях отправитель TCP будет просто отбрасывать блоки D-SACK, обрабатывая блоки SACK в поле опции SACK обычным путем.

[1]Selective Acknowledgement — селективные подтверждения.

3. Формат опции SACK, определенный в RFC 2018

Опция SACK определена в RFC 2018 следующим образом:

Опция селективных подтверждений (SACK) в заголовке TCP содержит число блоков SACK, каждый из которых указывает левый и правый край блока данных, принятого получателем TCP. В частности, блок представляет непрерывный диапазон порядковых номеров полученных данных, где «левый край» указывает первый порядковый номер для блока, а «правый край» - порядковый номер, непосредственно следующий за последним порядковым номером блока.



RFC 2018 предполагает, что первый блок SACK указывает сегмент, который вызвал данное подтверждение. В RFC 2018 сказано, когда получатель данных решает передать опцию SACK: «Первый блок SACK ... **должен** указывать непрерывный блок данных, содержащий сегмент, который вызвал передачу данного сегмента ACK, если он не опережает значение поля Acknowledgment Number в заголовке[1]».

Однако в RFC 2018 не рассматривается использование опции SACK при подтверждении дубликата сегмента. Например, в RFC 2018 сказано: «Если опции SACK передаются, их **следует** включать во все сегменты ACK, которые не являются подтверждением для старшего порядкового номера в приемной очереди получателя[1]». RFC 2018 не задает использования опции SACK при получении дубликата сегмента и поле кумулятивного подтверждения в ACK подтверждает все данные в очереди получателя.

4. Использование опции SACK для информирования о дубликатах

В этом разделе описано использование блоков SACK для случаев применения опции SACK с целью информирования о получении дубликата сегмента. При использовании D-SACK в качестве первого блока опции SACK **следует** помещать блок D-SACK, указывающий порядковые номера для сегмента-дубликата, вызвавшего подтверждение. Если дубликат является частью более крупного блока с разрывами из очереди на приемной стороне, для указания этого блока следует использовать последующий блок SACK. Дополнительные блоки SACK могут служить для указания других блоков с разрывами, как указано в RFC 2018.

Ниже приведены рекомендации по информированию о получении сегментов-дубликатов:

(1) Блок D-SACK используется только для информировании о дублировании непрерывной последовательности данных, принятых получателем в последнем (самом свежем) пакете.

(2) Рапорт о получении каждой дублирующей непрерывной последовательности передается не более, чем в одном блоке D-SACK (т. е., передача получателем двух идентичных блоков D-SACK в последовательных пакетах возможна лишь при получении двух дубликатов сегментов).

(3) Левый край блока D-SACK указывает первый порядковый номер в непрерывной последовательности дубликата, а правый край блока D-SACK указывает порядковый номер, следующий непосредственно за последним порядковым номером в непрерывной цепочке дубликата.

(4) Если блок D-SACK сообщает о непрерывной последовательности дубликатов из (возможно более крупного) блока данных в приемной очереди получателя, во втором блоке SACK данной опции SACK следует указать этот (возможно более крупный) блок данных.

(5) Вслед за упомянутыми выше блоками SACK могут размещаться дополнительные блоки SACK, служащие для информировании о дополнительных блоках данных в соответствии с RFC 2018.

Отметим, что по причине информирования о каждом сегменте-дубликате только в одном пакете ACK эта информация будет теряться при отбрасывании пакета ACK в сети.

4.1 Информирование о полном дубликате

Приведенные выше рекомендации проиллюстрированы тремя примерами. В каждом примере предполагается, что получатель данных имеет первые пять принятых сегментов по 500 байтов в каждом и подтверждает их в поле кумулятивных подтверждений со значением 4000 (предполагается отсчет номеров с 0). Блок D-SACK в каждом примере подчеркнут.

4.1.1. Пример 1. Рапорт о дубликате сегмента.

По причине потери нескольких пакетов ACK отправитель данных повторно передает пакеты 3000-3499 и на стороне получателя возникает дубликат сегмента с номерами 3000-3499. Получатель отправляет подтверждение со значением 4000 в поле кумулятивного подтверждения и блоком D-SACK, указывающим номера 3000-3500.

Переданные сегменты	**Принятые сегменты**	**Переданные подтверждения (включая SASK)**
3000-3499	3000-3499	3500 (ACK отброшено)
3500-3999	3500-3999	4000 (ACK отброшено)
3000-3499	3000-3499	4000, SACK=3000-3500

4.1.2. Пример 2: Рапорт о нарушении порядка и дублировании сегмента.

После потери данных получатель принимает сегмент с нарушением порядка, который инициирует опцию SACK в соответствии с RFC 2018. По причине потери нескольких пакетов ACK отправитель повторно передает пакет данных. Получатель, приняв дубликат, сообщает о нем в первом блоке D-SACK:

[1]Цитируется по переводу RFC 2018. *Прим. перев.*

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
3000-3499	3000-3499	3500 (ACK отброшено)
3500-3999	3500-3999	4000 (ACK отброшено)
4000-4499	(пакет данных потерян)	
4500-4999	4500-4999	4000, SACK=4500-5000 (ACK отброшено)
3000-3499	3000-3499	4000, SACK=3000-3500, 4500-5000

4.1.3. Пример 3: Отчет о дубликате сегмента, принятого с нарушением порядка.

В результате потери пакета данных на приемной стороне возникает два сегмента с нарушением порядка. После этого получатель принимает дубликат одного из этих сегментов:

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
3500-3999	3500-3999	4000
4000-4499	(пакет данных потерян)	
4500-4999	4500-4999	4000, SACK=4500-5000
5000-5499	5000-5499	4000, SACK=4500-5500
	(пакет-дубликат)	
	5000-5499	4000, SACK=5000-5500, 4500-5500

4.2. Информирование о частичном дублировании сегментов

Отправитель может передать пакет, содержащий один или несколько частичных дубликатов ранее отправленных сегментов (т. е., часть пакета может уже быть получена, но остальная часть передается впервые). Это может происходить в тех случаях, когда размер передаваемых отправителем сегментов увеличивается (например, в результате обнаруженного отправителем роста PMTU в действующем сеансе TCP). Приведенные в параграфе 4 рекомендации применимы и к информированию о частичном дублировании сегментов. Ниже приводятся примеры реализации этих рекомендаций для информирования о частичном дублировании сегментов.

При использовании опции SACK для информирования о частичном дублировании сегментов первый блок D-SACK сообщает о первом частичном дубликате. Если подтверждаемый пакет включает более одного частичного дубликата, в опции SACK сообщается только о первом таком дубликате. Примеры рапортов приведены ниже.

4.2.1. Пример 4: Рапорт об одном сегменте с частичным дублированием.

Отправитель увеличивает размер пакета от 500 до 1000 байтов. Получатель после этого принимает 1000-байтовый пакет, содержащих 500-байтовый субсегмент, который уже был передан и новый субсегмент такого же размера. Получатель сообщает только о полученном ранее субсегменте, используя один блок D-SACK.

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	500-999	1000
1000-1499	(задержан)	
1500-1999	(пакет данных потерян)	
2000-2499	2000-2499	1000, SACK=2000-2500
1000-2000	1000-2000	1500, SACK=2000-2500
	1000-2000	2500, SACK=1000-1500

4.2.2. Пример 5: 2 дубликата с разрывом, покрываемые кумулятивным ACK.

После увеличения отправителем размера пакета от 500 до 1500 байтов получатель принимает пакет с двумя не последовательными (разрыв в номерах) дубликатами 500-байтовых субсегментов, которые включены в поле кумулятивного подтверждения. Получатель информирует отправителя только о первом из этих дубликатов, используя один блок D-SACK.

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	500-999	1000
1000-1499	(задержан)	
1500-1999	(пакет данных потерян)	
2000-2499	(задержан)	
2500-2999	(пакет данных потерян)	
3000-3499	3000-3499	1000, SACK=3000-3500
1000-2499	1000-1499	1500, SACK=3000-3500
	2000-2499	1500, SACK=2000-2500, 3000-3500
	1000-2499	2500, SACK=1000-1500, 3000-3500

4.2.3. Пример 6: 2 дубликата с разрывом, не покрываемые кумулятивным ACK.

Этот случай отличается от рассмотренного в примере 5 лишь тем, что после увеличения отправителем размера пакетов получатель принимает два не последовательных (с разрывом в номерах) субсегмента с дубликатами, номера которых превышают номер в поле кумулятивного подтверждения. В этом случае блок D-SACK информирует о первом субсегменте-дубликате, а следующий за ним блок SACK информирует о наибольшем блоке не последовательных данных.

Переданные сегменты	**Принятые сегменты**	**Переданные подтверждения (включая SASK)**
500-999	500-999	1000
1000-1499	(пакет данных потерян)	
1500-1999	(задержан)	
2000-2499	(пакет данных потерян)	
2500-2999	(задержан)	
3000-3499	(пакет данных потерян)	1000, SACK=3000-3500
3500-3999	3500-3999	1000, SACK=3500-4000
1000-1499	(пакет данных потерян)	
1500-2999	1500-1999	1000, SACK=1500-2000, 3500-4000
	2500-2999	1000, SACK=2500-3000, 1500-2000, 3500-4000[1]
	1500-2999	1000, SACK=1500-2000, 1500-3000, 3500-4000

4.3. Взаимодействие D-SACK и PAWS

RFC 1323 [RFC1323] задает алгоритм защиты порядковых номеров PAWS[2]. Этот метод позволяет отличить порядковые номера новых данных от номеров из предыдущего цикла нумерации[3]. Дубликаты сегментов могут детектироваться PAWS, как относящиеся к предыдущему циклу нумерации.

В RFC 1323 указано, что для таких пакетов:

> «в ответ передается подтверждение, как указано в RFC 793 (стр. 69[4]), а сегмент отбрасывается[5].»

Поскольку PAWS требует передачи ACK, конфликтов между PAWS и применением D-SACK не возникает. Блок D-SACK может быть включен в опцию SACK пакета ACK, как сказано в разделе 4, независимо от использования PAWS на приемной стороне TCP и независимо от решения PAWS по части корректности порядкового номера сегмента данных.

Отправителям TCP, получающим блоки D-SACK, следует принимать во внимание, что сегмент, указанный, как дубликат, может относиться к предыдущему циклу нумерации. Это не зависит от использования PAWS отправителем данных TCP. Мы не предполагаем возникновения в результате серьезных проблем для отправителей, использующих информацию D-SACK.

5. Детектирование дубликатов пакетов

Данное расширение опции SACK дает получателю возможность точного информирования отправителя о получении дубликатов. Поскольку о каждом приеме дубликата отчет передается в единственном пакете ACK, потеря такого пакета приведет к тому, что отправитель не узнает о дубликате. Кроме того, следует отметить, что отправитель не обязан верить в точность передаваемой получателем информации [SCWA99].

Чтобы убедиться в том, что первый блок (D)SACK в подтверждении действительно подтверждает получение дубликата, отправителю следует сравнить пространство порядковых номеров в первом блоке SACK с кумулятивным ACK из **того же пакета**. Если номера в пространстве SACK меньше, чем в кумулятивном ACK, это говорит о том, что указанный блоком SACK сегмент был принят получателем более одного раза. Для реализации **недопустимо** сравнивать пространство номеров в блоке SACK с переменной состояния TCP **snd.una** (общее кумулятивное подтверждение ACK), поскольку это может приводить к ошибочным заключениям в случае нарушения порядка доставки пакетов ACK.

Если номера из пространства первого блока SACK больше кумулятивного ACK, отправитель будет сравнивать пространство из первого блока SACK с пространством из второго блока SACK (если он есть). Такое сравнение позволяет определить, говорит ли первый блок SACK о дубликате данных, которые «лежат выше» кумулятивного ACK.

Реализации TCP, поддерживающие RFC 2581 [RFC2581], могут видеть дубликаты пакетов в каждой из четырех рассмотренных далее ситуаций. Данный документ не задает действий реализации TCP в этих случаях. Расширение опции SACK просто позволяет отправителю детектировать каждый из таких случаев. Отметим, что четыре перечисленных условия не исчерпывают все возможные ситуации дублирования пакетов, но представляют наиболее распространенные (типичные) случаи. Последующие документы будут описывать основанные на опыте предложения в части отклика отправителей на обнаружение ненужных повторов в результате нарушения порядка, потери пакетов ACK или ускоренного тайм-аута повторной передачи.

---

[1] В оригинале ошибочно указано иное — см. https://www.rfc-editor.org/errata_search.php?eid=365. *Прим. перев.*
[2] Protection Against Wrapped Sequence Numbers — защита от номеров из прошлого цикла нумерации.
[3] До перехода от максимального номера к 0. *Прим. перев.*
[4] Страница 23 опубликованного на сайте www.protocols.ru перевода RFC 793. *Прим. перев.*
[5] Цитируется по переводу RFC 1323. *Прим. перев.*

5.1. Репликация в сети

Данное расширение опции SACK позволяет обнаружить репликацию пакетов в сети. Например:

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	500-999	1000
1000-1499	1000-1499	
	(репликация)	
	1000-1499	1500, SACK=1000-1500

В этом случае второй пакет реплицируется в сети. Пакет ACK с блоком D-SACK, номера в котором меньше, нежели в поле ACK и отличаются от переданного ранее повторного сегмента, указывают на репликацию пакета в сети.

Без D-SACK:

Если D-SACK не используется и последнее подтверждение ACK было прицеплено к пакету данных, отправитель не узнает о репликации пакета в сети. Если D-SACK не используется и ни одно из двух последних подтверждений ACK не было прицеплено к пакетам данных, отправитель может обоснованно предположить, что какой-то из пакетов данных или пакет ACK был реплицирован в сети. Получение пакета D-SACK дает отправителю подтверждение репликации данного пакета в сети (предполагается, что получатель не лжет).

Нужно исследовать:

Опция SACK сама по себе уже позволяет отправителю идентифицировать дубликаты пакетов ACK, которые не подтверждают новых данных, но опция D-SACK обеспечивает отправителю более строгое подтверждение того, что дубликат ACK не подтверждает новых данных. Информация о том, что дубликат ACK не подтверждает новых данных, позволяет отправителю воздержаться от использования дубликатов ACK, как индикатора потери пакетов (например, включение Fast Retransmit) или передачи дополнительных данных (например, включение Fast Recovery).

5.2. Ложный повтор передачи в результате нарушения порядка

Если порядок пакетов изменяется в сети так, что более 3 пакетов в сегмент принимаются с нарушением порядка, алгоритм TCP Fast Retransmit будет повторно передавать пакет. Пример этого показан ниже:

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	500-999	1000
1000-1499	(задержан)	
1500-1999	1500-1999	1000, SACK=1500-2000
2000-2499	2000-2499	1000, SACK=1500-2500
2500-2999	2500-2999	1000, SACK=1500-3000
1000-1499	1000-1499	3000
	1000-1499	3000, SACK=1000-1500

В этом случае пакет ACK с блоком SACK, указывающим номера меньше, чем поле ACK, и идентичным уже переданному повторно сегменту, указывает на существенное изменение порядка доставки и вызывает ложный (ненужный) повтор передачи.

Без D-SACK:

При использовании D-SACK, как показано выше, отправитель узнает, что переданный изначально сегмент 1000-1499 был задержан в сети или первый повтор сегмента 1000-1499 был отброшен, а второй повтор был сдублирован. С учетом отсутствия фактов дублирования в сети других пакетов второй вариант может рассматриваться, как маловероятный.

Без использования D-SACK отправитель получит лишь информацию о том, что либо первая передача сегмента 1000-1499 задержалась в сети, либо один из сегментов данных или финальный пакет ACK был продублирован в сети. Таким образом, использование D-SACK позволяет отправителю получить более надежную информацию о том, что первая передача сегмента 1000-1499 не была отброшена в сети.

В работах [AP99], [L99] и [LK00] отмечается, что отправитель может однозначно детектировать ненужные повторы передачи с помощью опции timestamp (временная метка). В [LK00] предложен основанный на временных метках алгоритм, который минимизирует «расплату» (penalty) за ненужный повтор передачи. В [AP99] предлагается эвристический метод детектирования ненужных повторов в средах без временных меток и SACK. В [L99] предлагается двухбитовое поле (как альтернатива опции timestamp) для однозначного маркирования первых трех повторов пакета. Похожая идея была предложена в [ISO8073].

Нужно исследовать:

Использование D-SACK позволяет отправителю детектировать некоторые случаи (например, при отсутствии потерь пакетов ACK), когда механизм Fast Retransmit был использован в результате нарушения порядка, а не потери пакетов. Это позволяет отправителю TCP настроить порог числа дубликатов подтверждений, позволяющий предотвратить необоснованное применение механизма Fast Retransmit в будущем. Наряду с этим, когда отправитель пост-фактум обнаруживает ненужное снижение размера окна, он может отменить это снижение в окне насыщения путем сброса ssthresh к значению прежнего окна насыщения и использовать механизм slow-start, пока окно насыщения не достигнет этой точки.

Во всех предложениях по отмене снижения размера окна насыщения следует принимать во внимание возможность наличия ложной информации о принятых пакетах от получателя TCP [SCWA99].

5.3. Повторная передача в результате потери ACK

При потере целого окна пакетов ACK будет возникать тайм-аут, пример которого показан ниже.

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	500-999	1000 (ACK теряется)
1000-1499	1000-1499	1500 (ACK теряется)
1500-1999	1500-1999	2000 (ACK теряется)
2000-2499	2000-2499	2500 (ACK теряется)
тайм-аут		
500-999	500-999	2500, SACK=500-1000

В этом случае все пакеты ACK отбрасываются и в результате возникает тайм-аут. Такую ситуацию можно идентифицировать по наличию в первом пакете ACK после тайм-аута блока D-SACK, говорящего о получении дубликатов данных.

Без D-SACK:

Без использования D-SACK отправитель в таких случаях не сможет принять решения об отсутствии потери пакетов с данными.

Нужно исследовать:

Для реализаций TCP, поддерживающих контроль перегрузок с использованием ACK [BPK97], такая возможность различить потерю пакетов данных и потерю пакетов ACK будет полезна на практике. В этом случае для соединения может быть реализован контроль перегрузок на пути возврата (ACK) независимо от контроля перегрузок на прямом пути (пакеты данных).

5.4. Ранний тайм-аут повтора передачи

Если на стороне отправителя значение RTO[1] слишком мало, тайм-аут может возникать даже при отсутствии потери пакетов в сети. Пример этого показан ниже.

Переданные сегменты	Принятые сегменты	Переданные подтверждения (включая SASK)
500-999	(задержан)	
1000-1499	(задержан)	
1500-1999	(задержан)	
2000-2499	(задержан)	
тайм-аут		
500-999	(задержан)	
	500-999	1000
1000-1499	(задержан)	
	1000-1499	1500
...		
	1500-1999	2000
	2000-2499	2500
	500-999	2500, SACK=500-1000
	1000-1499	2500, SACK=1000-1500
...		

В этом случае после тайм-аута повторяется передача первого пакета. Далее исходное окно пакетов поступает к получателю и это ведет к генерации пакетов ACK для принятых сегментов. Вслед за этим поступают переданные повторно сегменты, для которых генерируются пакеты ACK с блоками SACK, указывающими дублирование.

Это можно идентифицировать, как ранний тайм-аут повторной передачи, поскольку пакет ACK для байта 1000 приходит после тайм-аута без данных SACK, а за ним следует пакет ACK с информацией SACK (500-999), показывающей, что переданный повторно пакет уже был получен.

Без D-SACK:

Если D-SACK не используется и один из дубликатов ACK был совмещен с пакетом данных, отправитель не узнает числа принятых дубликатов. Если D-SACK не используется и все пакеты ACK не совмещены с пакетами данных, отправитель получит N дубликатов ACK при передаче N повторов. В этом случае отправитель может обоснованно сделать вывод о том, что часть пакетов ACK была реплицирована в сети или тайм-аут возник чересчур рано (при условия доверия к данным от получателя).

Нужно исследовать:

После того, как отправитель идентифицирует слишком ранний (ненужный) тайм-аут повтора, он может изменить значение RTO для предотвращения ненужных тайм-аутов. Наряду с этим при определении отправителем (пост-фактум) ненужного снижения размера окна, он может отказаться от этого снижения в рамках окна насыщения.

[1] Тайм-аут повтора передачи. *Прим. перев.*

6. Вопросы безопасности

Этот документ не меняет текущего состояния вопросов безопасности TCP.

7. Благодарности

Авторы выражают свою благодарность Mark Handley, Reiner Ludwig и Venkat Padmanabhan за обсуждения по теме документа и благодарят Mark Allman за полезные отклики.

8. Литература

[AP99] Mark Allman and Vern Paxson, On Estimating End-to-End Network Path Properties, SIGCOMM 99, August 1999.

[BPS99] J.C.R. Bennett, C. Partridge, and N. Shectman, Packet Reordering is Not Pathological Network Behavior, IEEE/ACM Transactions on Networking, Vol. 7, No. 6, December 1999, pp. 789-798.

[BPK97] Hari Balakrishnan, Venkata Padmanabhan, and Randy H. Katz, The Effects of Asymmetry on TCP Performance, Third ACM/IEEE Mobicom Conference, Budapest, Hungary, Sep 1997.

[F99] Floyd, S., Re: TCP and out-of-order delivery, Message ID <199902030027.QAA06775@owl.ee.lbl.gov> to the end-to-end-interest mailing list, February 1999.

[ISO8073] ISO/IEC, Information-processing systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification, Internation Standard ISO/IEC 8073, December 1988.

[L99] Reiner Ludwig, A Case for Flow Adaptive Wireless links, Technical Report UCB//CSD-99-1053, May 1999.

[LK00] Reiner Ludwig and Randy H. Katz, The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions, SIGCOMM Computer Communication Review, V. 30, N. 1, January 2000.

[RFC1323] Jacobson, V., Braden, R. and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[RFC2018] Mathis, M., Mahdavi, J., Floyd, S. and A. Romanow, "TCP Selective Acknowledgement Options", RFC 2018, April 1996.

[RFC2581] Allman, M., Paxson,V. and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[SCWA99] Stefan Savage, Neal Cardwell, David Wetherall, Tom Anderson, TCP Congestion Control with a Misbehaving Receiver, ACM Computer Communications Review, pp. 71-78, V. 29, N. 5, October, 1999.

Адреса авторов

Sally Floyd

AT&T Center for Internet Research at ICSI (ACIRI)

Phone: +1 510-666-6989

EMail: floyd@aciri.org

URL: http://www.aciri.org/floyd/

Jamshid Mahdavi

Novell

Phone: 1-408-967-3806

EMail: mahdavi@novell.com

Matt Mathis

Pittsburgh Supercomputing Center

Phone: 412 268-3319

EMail: mathis@psc.edu

URL: http://www.psc.edu/~mathis/

Matthew Podolsky

UC Berkeley Electrical Engineering & Computer Science Dept.

Phone: 510-649-8914

EMail: podolsky@eecs.berkeley.edu

URL: http://www.eecs.berkeley.edu/~podolsky

Перевод на русский язык

Николай Малых

nmalykh@protocols.ru

Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.