Network Working Group
Request for Comments: 2018
Category: Standards Track

M. Mathis
J. Mahdavi
PSC
S. Floyd
LBNL
A. Romanow
Sun Microsystems

October 1996

Опции селективных подтверждений TCP

TCP Selective Acknowledgment Options

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.

Аннотация

TCP может сталкиваться с проблемой снижения производительности при потере множества пакетов из одного окна данных. Из ограниченной информации кумулятивных подтверждений отправитель TCP может лишь узнать о потере одного пакета за период кругового обхода. В этом случае отправитель с агрессивным поведением может принять решение об ускоренном повторе передачи пакетов, однако передаваемые повторно сегменты могут оказаться уже полученными.

Механизм селективных подтверждений (SACK[1]) при совместном использовании с политикой селективного повтора передачи может помочь в решении этой проблемы. Приемная сторона TCP передает отправителю пакеты SACK, информирующие того о приеме отправленных сегментов. В этом случае отправитель может ограничить повтор передачи лишь отсутствующими сегментами.

Этот документ предлагает реализацию SACK и рассматривает вопросы производительности при использовании этой опции, а также связанные с этим темы.

Благодарности

Значительная часть текста в этом документе заимствована из RFC 1072 TCP Extensions for Long-Delay Paths, который написали Bob Braden и Van Jacobson. Авторы выражают свою признательность Kevin Fall (LBNL), Christian Huitema (INRIA), Van Jacobson (LBNL), Greg Miller (MITRE), Greg Minshall (Ipsilon), Lixia Zhang (XEROX PARC и UCLA), Dave Borman (BSDI), Allison Mankin (ISI) и другим за просмотр документа и конструктивные комментарии.

1. Введение

Потеря множества пакетов из одного окна данных может приводить к катастрофическому влиянию на пропускную способность соединений TCP. Протокол TCP [Postel81] использует схему кумулятивных подтверждений, в которой принятые сегменты, не относящиеся к левому краю окна приема, не подтверждаются. Это ведет к тому, что отправитель должен ждать в течение периода кругового обхода для принятия решения о потере пакета или без необходимости повторять передачу сегментов, которые были корректно приняты [Fall95]. При использовании схемы кумулятивных подтверждений TCP обычно теряет ACK-синхронизацию, что ведет к снижению пропускной способности.

Стратегия селективных подтверждений SACK исправляет отмеченное поведение для случаев отбрасывания множества сегментов. При селективном подтверждении получатель может информировать отправителя обо всех сегментах, полученных на приемной стороне, что позволяет отправителю повторно передавать лишь те сегменты, которые реально были потеряны.

Селективные подтверждение использует несколько транспортных протоколов, включая NETBLT [Clark87], XTP [Strayer92], RDP [Velten84], NADIR [Huitema81] и VMTP [Cheriton88]. Имеются эмпирические подтверждения эффективности селективных подтверждений - простые эксперименты с протоколом RDP показывают, что запрет селективных подтверждений существенно увеличивает число повторно передаваемых сегментов на путях с высокими задержками и потерей пакетов в Internet [Partridge87]. Недавнее моделирование, выполненное Kevin Fall и Sally Floyd [Fall95], показало, преимущества TCP с SACK по сравнению с реализациями TCP non-SACK Tahoe и Reno.

В RFC 1072 [VJ88] описана одна из возможных реализаций опций SACK для TCP. К сожалению эта модель не была развернута в Internet по причине разногласий по вопросу совместного использования опций SACK с опцией сдвига окна TCP (исходно описанной в RFC1072 и пересмотренной в [Jacobson92]).

Мы предлагаем незначительное изменение опций SACK по сравнению с RFC 1072. В частности, передача селективного подтверждения для полученных последними данных снижает потребность в длинных опциях SACK [Keshav94, Mathis95]. Кроме того, опция SACK в предлагаемом варианте передает полные 32-битовые порядковые

[1]Selective Acknowledgment.

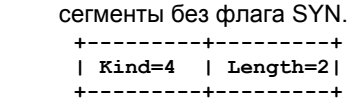
номера. Эти два изменения исчерпывают список отличий от предложенной в RFC 1072 модели. Они упрощают реализацию SACK и решают проблему отказоустойчивости.

Расширение для селективных подтверждений использует две опции TCP. Первая опция SACK-permitted[1] включает механизм и может передаваться в сегменте SYN для индикации возможности использования опции SACK после организации соединения. Другая опция (собственно, SACK) может передаваться в существующем соединении, для которого при организации была включена возможность селективного подтверждения с помощью опции SACK-permitted.

Опция SACK включается в сегмент, передаваемый приемным модулем TCP передающей стороне TCP (будем называть эти стороны получателем и отправителем данных, соответственно). Мы будем рассматривать простейший, односторонний поток данных. Для двухсторонних потоков селективные подтверждения могут независимо использоваться для каждого из направлений.

2. Опция Sack-Permitted

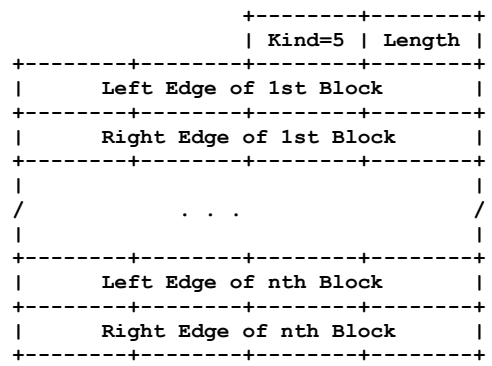
Эта двухбайтовая опция может передаваться в сегменте SYN модулем TCP, который способен принимать (и, предположительно, обрабатывать) опции SACK после организации соединения. **Недопустимо** включать эту опцию в сегменты без флага SYN.



Опция TCP Sack-Permitted:

Kind: 4

3. Формат опции SACK



Опция SACK используется для передачи расширенной информации о доставке (подтверждений) от получателя к отправителю через существующее соединение TCP.

Опция TCP SACK:

Kind: 5

Length: переменное значение

Опция SACK передается получателем данных для информирования отправителя блока данных с разрывами[2] о доставке данных и размещении их в приемной очереди. Получатель ждет приема данных (возможно, с использованием повторной передачи), заполняющих пропуски в пространстве порядковых номеров между полученными блоками. При получении отсутствующих сегментов получатель данных подтверждает их обычным способом, сдвигая левый край окна в поле Acknowledgement Number заголовка TCP. Опция SACK не меняет трактовки поля Acknowledgement Number.

Опция содержит список непрерывных блоков порядковых номеров попадающих в окно данных, которые были приняты и помещены в очередь.

Каждый непрерывный блок данных, помещенный в очередь на стороне получателя, определяется в опции SACK двумя 32-битовыми целыми числами с использованием сетевого порядка байтов:

- **Left Edge of Block**

  Первый порядковый номер для данного блока.

- **Right Edge of Block**

  Порядковый номер, непосредственно следующий за последним порядковым номером данного блока.

Каждый блок представляет принятые байты данных, составляющие непрерывное изолированное множество (т. е., данные с порядковыми номерами Left Edge of Block - 1 и Right Edge of Block еще не получены).

Опция SACK, содержащая n блоков, будет иметь размер 8*n+2 байтов, поэтому 40 байтов доступного пространства опций TCP позволяют указать не более 4 блоков. Предполагается, что опция SACK часто будет использоваться вместе с опцией Timestamp, применяемой для RTTM[3] [Jacobson92], для которой требуется 10 байтов (плюс два байта заполнения), что сокращает число указываемых в опции SACK блоков до 3.

Опция SACK является справочной в том смысле, что она уведомляет отправителя данных о том, что получатель принял указанные в опции сегменты. Однако получатель может впоследствии отбросить данные, доставку которых он подтвердил в опции SACK. В параграфе 8 обсуждаются следствия применения SACK (в частности, возможность получателя отказаться от подтверждения или отбросить подтвержденные в SACK данные).

[1] Селективные подтверждения разрешены.

[2] Вследствие потери пакетов или нарушения порядка доставки. *Прим. перев.*

[3] Round-trip time measurement - измерение времени кругового обхода. *Прим. перев.*

4. Генерация опций SACK - поведение получателя

Если получатель данных обнаруживает опцию SACK-Permitted в сегменте SYN для данного соединения, он **может** принять решение о генерации опций SACK, как описано ниже. Если получатель данных генерирует опции SACK при каких-либо обстоятельствах, ему **следует** генерировать их при любых разрешенных условиях. Если получатель данных не обнаружил опции SACK-Permitted в сегменте SYN для данного соединения, для него **недопустимо** использовать опцию SACK в данном соединении.

Если опции SACK передаются, их **следует** включать во все сегменты ACK, которые не являются подтверждением для старшего порядкового номера в приемной очереди получателя. Такие ситуации говорят о возникновении в сети потери или нарушения порядка доставки данных. В параграфе 4.2.2.21 RFC 1122 обсуждаются причины, по которым получатель может передавать сегменты ACK в качестве отклика на дополнительные сегменты, принятые в таком состоянии. Получателю **следует** передавать ACK для каждого принятого корректного сегмента, содержащего новые данные, и в каждом из таких «дубликатов» ACK **следует** передавать опцию SACK.

Если получатель решил передавать опцию SACK, используются приведенные ниже правила.

- Первый блок SACK (т. е., блок, следующий сразу после полей kind и length в этой опции) **должен** указывать непрерывный блок данных, содержащий сегмент, который вызвал передачу данного сегмента ACK, если он не опережает значение поля Acknowledgment Number в заголовке. Это гарантирует, что сегмент ACK с опцией SACK будет отражать самое свежее изменение в приемной очереди получателя.
- Получателю данных **следует** включать столько разных блоков SACK, сколько позволяет размер опции SACK. Отметим, что размера опции может оказаться недостаточно для передачи информации обо всех блоках, находящихся в приемной очереди получателя.
- Опцию SACK следует заполнять самыми новыми блоками SACK (с учетом первого блока SACK в предыдущей опции SACK), которые не являются подмножествами блоков SACK, уже включенных в создаваемую опцию. Это гарантирует, что при нормальной работе любой сегмент, являющийся частью разорванного блока данных, удерживаемого получателем, будет указан в одной из трех последовательных опций SACK даже реализаций TCP с большим размером окна [RFC1323]). После первого блока SACK можно включать остальные SACK-блоки в опцию SACK в произвольном порядке.

Важно, что опция SACK всегда сообщает о блоке, содержащем сегмент, принятый последним, поскольку эта информация дает отправителю наиболее свежие данные о состоянии сети и приемной очереди получателя.

5. Интерпретация опций SACK и стратегия повтора - отправитель

При получении сегмента ACK с опцией SACK отправителю **следует** записать селективное подтверждение для использования в будущем. Предполагается, что отправитель данных имеет буфер повторной передачи для хранения сегментов, которые уже переданы, но еще не подтверждены, в соответствии с порядковыми номерами. Если отправитель перед повтором передачи выполняет процедуру репакетизации, границы блоков в полученной опции SACK могут не совпадать с границами сегментов в очереди для повторной передачи, однако это не создает серьезных сложностей для отправителя.

Ниже описан один из возможных вариантов реализации поведения отправителя. Предположим, что для каждого сегмента в очереди повторной передачи имеется (новый) флаг SACKed[1], используемый для индикации того, что доставка этого конкретного сегмента подтверждена в опции SACK.

Когда приходит сегмент подтверждения, содержащий опцию SACK, отправитель данных будет устанавливать флаг SACKed для сегментов, которые были селективно подтверждены. Говоря более конкретно, для каждого блока из опции SACK отправитель данных будет устанавливать флаги SACKed во всех сегментах очереди на повторную передачу, которые полностью содержатся в этом блоке. Это требует прямого сравнения порядковых номеров.

После того, как флаг SACKed установлен (в результате обработки полученной опции SACK), отправитель данных будет пропускать сегменты с таким флагом при последующих повторах передачи. Любой сегмент без флага SACKed с номером, который меньше старшего (по номеру) сегмента с флагом SACKed, остается доступным для повторной передачи.

По истечении тайм-аута повторной передачи отправителю данных **следует** сбросить флаги SACKed, поскольку тайм-аут может показывать, что получатель данных «отрекся» от селективных подтверждений. Отправитель данных **должен** повторно передать сегмент, расположенный на левом краю окна после тайм-аута повторной передачи независимо от наличия для этого сегмента селективного подтверждения. Сегмент не будет удаляться из очереди с освобождением буфера, пока не уйдет за левый край окна.

5.1 Вопросы контроля насыщения

Этот документ не является попыткой детальной спецификации алгоритмов контроля насыщения для реализаций TCP с поддержкой SACK. Однако алгоритмы контроля насыщения, имеющиеся в реализациях TCP, ставших фактическими стандартами, **должны** быть сохранены [Stevens94]. В частности, для сохранения устойчивости к нарушениям порядка доставки пакетов в сети восстановление не включается по одному сегменту ACK, информирующему о нарушении порядка доставки на приемной стороне. Более того, в процессе восстановления отправитель данных ограничивает число сегментов, передаваемых в ответ на получение каждого сегмента ACK. Существующие реализации ограничивают отправителя данных передачей одного сегмента в процессе ускоренного восстановления в стиле Reno или двух сегментов в течение процедуры замедленного старта [Jacobson88]. Другие аспекты контроля насыщения (такие, как снижение размера окна насыщения в ответ на перегрузку) должны сохраняться аналогичным образом.

Использование тайм-аута повтора в качестве механизма детектирования отброшенных пакетов не меняется для режима селективных подтверждений. Поскольку получателю данных разрешено отбрасывать данные, подтвержденные в SACK, при возникновении тайм-аута повтора передачи отправителю **следует**[2] игнорировать предшествующую информацию SACK при определении данных для повтора передачи.

[1]Подтвержден с помощью SACK.

Будущие исследования алгоритмов контроля насыщения могут использовать преимущества, обеспечиваемые дополнительной информацией из SACK. Одна из таких областей для будущих исследований касается изменения TCP для беспроводных и спутниковых сетей, где потеря пакетов не обязательно является индикацией перегрузки.

6. Эффективность и поведение в плохих условиях

Если на пути возврата сегментов ACK и опций SACK не возникает потери пакетов, одного блока на пакет с опцией SACK будет достаточно в любом случае. Каждый сегмент, прибывающий в то время, когда у получателя имеются разорванные блоки данных, будет вызывать передачу получателем сегмента ACK с опцией SACK, содержащей один измененный блок из приемной очереди. Отправитель данных будет способен создать точную реплику приемной очереди получателя путем объединения всех первых блоков SACK.

Поскольку путь доставки подтверждений может вызывать потери пакетов, для опции SACK определено включение более одного блока SACK в пакет подтверждения. Резервные блоки в пакете с опцией SACK повышают отказоустойчивость при доставке SACK в случае потери сегментов ACK. Для получателей, использующих также опцию временных меток [Jacobson92], в опцию SACK можно поместить до 3 блоков SACK. Таким образом, каждый блок SACK будет в общем случае повторяться по крайней мере трижды, если нужно, по одному разу в каждом из трех последовательных пакетов ACK. Однако, если все пакеты ACK, сообщающие о конкретном блоке SACK, будут отброшены, отправитель данных может предположить, что данные из этих блоков SACK не были получены на приемной стороне и без необходимости повторно передать эти сегменты.

Использование других опций TCP может дополнительно снижать число доступных блоков SACK до 2 и даже до 1. Это будет снижать уровень избыточности при доставке SACK в условиях возможной потери сегментов ACK. Но даже в этом случае использование SACK существенно снижает вероятность ненужных повторов передачи по сравнению с обычными реализациями TCP. Наихудшие условия, требующие от отправителя без необходимости повторять передачу сегментов более детально рассмотрены в отдельном документе [Floyd96].

Старые реализации TCP, которые не поддерживают опцию SACK, не будут необоснованно ущемляться при конкуренции с TCP, поддерживающими SACK. Этот вопрос более детально рассмотрен в работе [Floyd96].

7. Примеры опции SACK

Приведенные здесь примеры служат попыткой продемонстрировать корректное поведение при генерации SACK на приемной стороне.

Предположим, что на левом краю окна размещается порядковый номер 5000, а отправитель передает подряд 8 сегментов, содержащих по 500 байтов данных.

Случай 1: Первые 4 доставлены, но 4 оставшихся отброшены в сети.

Получатель будет возвращать обычный сегмент ACK, подтверждающий порядковый номер 7000, без опции SACK.

Случай 2: Первый сегмент отброшен в сети, а оставшиеся 7 доставлены получателю.

Сегмент-триггер	Номер ACK	Левый край окна	Правый край окна
5000	потеря		
5500	5000	5500	6000
6000	5000	5500	6500
6500	5000	5500	7000
7000	5000	5500	7500
7500	5000	5500	8000
8000	5000	5500	8500
8500	5000	5500	9000

При приеме каждого из семи доставленных сегментов получатель данных будет возвращать сегмент TCP ACK, подтверждающий порядковый номер 5000 и содержащий опцию SACK, указывающую один блок данных из приемной очереди, как показано в таблице справа.

Случай 3: Сегменты 2, 4, 6 и 8 (последний) отброшены в сети.

Получатель обычным способом подтверждает первый пакет. Третий, пятый и седьмой пакеты инициируют опции SACK, как показано в таблице ниже.

Сегмент-триггер	ACK	Первый блок		Второй блок		Третий блок	
		Левый край	Правый край	Левый край	Правый край	Левый край	Правый край
5000	5000						
5500	потеря						
6000	5500	6000	6500				
6500	потеря						
7000	5500	7000	7500	6000	6500		
7500	потеря						
8000	5500	8000	8500	7000	7500	6000	6500
8500	потеря						

Предположим, что четвертый пакет принят с нарушением порядка (это может быть результатом нарушения порядка пакетов в сети или следствием того, что второй пакет был передан повторно и потерян, а четвертый пакет был передан повторно). В этом случае получатель данных будет иметь только два блока SACK и ответит отправителю данных следующим образом:

[2] В исходном документе сказано «**должен**», но впоследствии это признано ошибкой - http://www.rfc-editor.org/errata_search.php?eid=1610. *Прим. перев.*

Сегмент -триггер	ACK	Первый блок		Второй блок		Третий блок	
		Левый край	Правый край	Левый край	Правый край	Левый край	Правый край
6500	5500	6000	7500	8000	8500		

Предположим теперь, что второй сегмент получен. Получатель данных в этом случае будет передавать селективное подтверждение:

Сегмент-триггер	ACK	Первый блок		Второй блок		Третий блок	
		Левый край	Правый край	Левый край	Правый край	Левый край	Правый край
5500	7500	8000	8500				

8. «Отречение» получателя от данных

Отметим, что получателю данных разрешено отбрасывать неподтвержденные данные из своей очереди даже в тех случаях, когда о них уже сообщено отправителю с помощью опции SACK. Такое отбрасывание не рекомендуется, но может быть использовано при нехватке на приемной стороне буферной емкости.

Получатель данных **может** принять решение об отбрасывании данных, для которых уже переданы селективные подтверждения в опциях SACK. В этом случае для генерации SACK на приемной стороне возникают дополнительные ограничения:

- Первый блок SACK **должен** отражать самый новый сегмент. Если этот сегмент планируется отбросить и получатель уже отбросил смежные сегменты, первый блок SACK **должен** информировать (как минимум) о левом и правом краях самого нового сегмента.
- Кроме новейшего сегмента в блоках SACK **недопустимо** сообщать о каких-либо более старых данных, которые уже не удерживаются получателем.

Поскольку получатель может позднее отбросить данные, указанные в опции SACK, отправителю **недопустимо** отбрасывать данные, пока они не подтверждены полем Acknowledgment Number в заголовке TCP.

9. Вопросы безопасности

Этот документ не оказывает влияния на параметры безопасности TCP.

10. Литература

[Cheriton88] Cheriton, D., "VMTP: Versatile Message Transaction Protocol", RFC 1045, Stanford University, February 1988.

[Clark87] Clark, D., Lambert, M., and L. Zhang, "NETBLT: A Bulk Data Transfer Protocol", RFC 998, MIT, March 1987.

[Fall95] Fall, K. and Floyd, S., "Comparisons of Tahoe, Reno, and Sack TCP", ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z, December 1995.

[Floyd96] Floyd, S., "Issues of TCP with SACK", ftp://ftp.ee.lbl.gov/papers/issues_sa.ps.Z, January 1996.

[Huitema81] Huitema, C., and Valet, I., An Experiment on High Speed File Transfer using Satellite Links, 7th Data Communication Symposium, Mexico, October 1981.

[Jacobson88] Jacobson, V., "Congestion Avoidance and Control", Proceedings of SIGCOMM '88, Stanford, CA., August 1988.

[Jacobson88] Jacobson, V. and R. Braden, "TCP Extensions for Long-Delay Paths", RFC 1072, October 1988.

[Jacobson92] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[Keshav94] Keshav, presentation to the Internet End-to-End Research Group, November 1994.

[Mathis95] Mathis, M., and Mahdavi, J., TCP Forward Acknowledgment Option, presentation to the Internet End-to-End Research Group, June 1995.

[Partridge87] Partridge, C., "Private Communication", February 1987.

[Postel81] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, DARPA, September 1981.

[Stevens94] Stevens, W., TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994.

[Strayer92] Strayer, T., Dempsey, B., and Weaver, A., XTP - the xpress transfer protocol. Addison-Wesley Publishing Company, 1992.

[Velten84] Velten, D., Hinden, R., and J. Sax, "Reliable Data Protocol", RFC 908, BBN, July 1984.

11. Адреса авторов

Matt Mathis и Jamshid Mahdavi

Pittsburgh Supercomputing Center

4400 Fifth Ave

Pittsburgh, PA 15213

mathis@psc.edu

mahdavi@psc.edu

Sally Floyd

Lawrence Berkeley National Laboratory
One Cyclotron Road
Berkeley, CA 94720
floyd@ee.lbl.gov

Allyn Romanow
Sun Microsystems, Inc.
2550 Garcia Ave., MPK17-202
Mountain View, CA 94043
allyn@eng.sun.com

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru