Internet Engineering Task Force (IETF)
Request for Comments: 8961
BCP: 233
Category: Best Current Practice
ISSN: 2070-1721

M. Allman
ICSI

November 2020

Requirements for Time-Based Loss Detection

Требования к детектированию потерь по времени

Аннотация

Многим протоколам нужно по той или иной причине детектировать потерю пакетов (например, для гарантированной доставки за счет повтора или понимания уровня перегрузки на пути через сеть). Хотя для обнаружения потерь было разработано много механизмов, протоколы на деле могут рассчитывать лишь на прошедшее без подтверждений время, чтобы счесть пакет потерянным. Каждая реализация механизма обнаружения потерь на основе времени вынуждена соблюдать баланс между корректностью и своевременностью, поэтому нет решения на все случаи. В этом документе представлены требования высокого уровня к детекторам потерь на основе времени для использования в базовых индивидуальных (unicast) коммуникациях через Internet. В рамках этих требований реализации могут определять параметры, наиболее подходящие для каждой ситуации.

Статус документа

Документ относится к категории Internet Best Current Practice.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошел открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc8961.

Оглавление

1. Введение

Будучи сетью сетей, Internet включает множество разных каналов и систем, которые поддерживают широкий класс задач. Предоставляемые через сеть услуги варьируются по качеству от доступного (best-effort) для множества слабо связанных элементов до надежно предсказуемого в контролируемых средах (например, между физически соединенными узлами или в строго контролируемом ЦОДе). У каждого пути через сеть имеется набор параметров, например, доступная пропускная способность, задержка и потеря пакетов. Учитывая разнородность сетей в Internet, можно считать эти свойства варьирующимися от статических до быстро меняющихся.

В этом документе рассматривается одно из свойств пути - потеря пакетов. В частности, предложены рекомендации по разработке и реализации детекторов потерь на основе времени, которые были выработаны за прошедшие десятки лет. Рассматривается достаточно общий случай, когда свойства потери пакетов в пути (a) не известны заранее и (b) меняются с течением времени. Кроме того, несмотря на возможность различных причин потери пакетов, здесь принят

[1]Internet Engineering Task Force.
[2]Internet Engineering Steering Group.

консервативный подход о том, что потери являются неявным признаком перегрузки [RFC5681]. Хотя эта позиция верна не во всех случаях, она хорошо послужила в качестве базового допущения, использованного еще в [Jac88]. Как будет отмечено в разделе 2, рекомендации этого документа следует считать базовыми для индивидуального трафика в сетях с доставкой по возможности (best-effort) и неоптимальными, хотя и применимыми в других ситуациях.

С учетом того, что в сетях best-effort потеря пакетов является обычным делом, обнаружение таких потеря является важной задачей для многих протоколов и приложений. Это связано с двумя основными причинами, указанными ниже.

(1) Обеспечение гарантированной доставки данных.

Отправитель должен понимать, какие из переданных пакетов не достигли адресата, чтобы можно было повторить их передачу.

(2) Контроль насыщения (перегрузок).

Как уже было отмечено, потеря пакетов часто служит неявной индикацией того, что отправитель передает пакеты слишком быстро и перегружает ту или иную часть пути через сеть. Поэтому отправителю нужно знать о потерях для снижения скорости.

Для обнаружения потерь в потоке пакетов применяются разные механизмы. Часто для этого служат постоянные или периодические подтверждения от получателя для информирования отправителя об отсутствии пакетов. Однако, несмотря на добрые намерения и отказоустойчивые механизмы, невозможно полностью доверять таким подтверждениям, поскольку они могут зависеть от времени и реальным подтверждением потери может являться лишь тайм-аут. То есть отправитель задает те или иные ожидания в части времени подтверждения доставки той или иной порции данных. Когда это время проходит без доставки подтверждения, отправитель считает данные потерянными.

Специфика схем обнаружения потерь на основе времени является компромиссом между корректностью и быстродействием, поскольку хочется одновременно:

- ждать достаточно долго для корректного детектирования;
- сократить задержку для приложений (перед повтором) и сети (перед снижением нагрузки).

Достичь обеих целей сложно, поскольку они противонаправлены [AP99]. Без долгого ожидания можно своевременно повторять передачу, но это ведет к риску ненужных (ложных) повторов и неоправданногоу снижения скорости передачи. Если ждать долго для достижения уверенности в потере пакетов, не будет своевременного восстановления и возникает риск продления перегрузки в сети.

Многие протоколы и приложения, такие как TCP [RFC6298], SCTP [RFC4960] и SIP [RFC3261], включают механизмы обнаружения потерь на основе времени. Опыт использования этих механизмов показывает, что зачастую конкретные настройки, отклоняющиеся от стандартизованных детекторов на основе времени, не оказывают нужного влияния на безопасность сети в части контроля перегрузок [AP99]. Поэтому в данном документе представлен набор независимых от протоколов требований высокого уровня к детектирования потерь на основе времени. Цель документа заключается в создании надежной основы для реализаций, обеспечивающих гибкие механизмы решения конкретной задачи.

1.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

2. Контекст

Этот документ отличается от того, как хотелось видеть идеальную систему. Обычно требования высокого уровня служат стартовой точкой, а затем разрабатываются конкретные протоколы, алгоритмы и системы, отвечающие этим требованиям. В рамках процесса стандартизации IETF разработано много схем детектирования потерь на основе времени без наличия какого-либо документа с требованиями, поскольку не было понимания, как написать такой документ. В результате принимались решения, которые представлялись подходящими.

К настоящему времени накопленный сообществом опыт позволяет задать базовые требования высокого уровня для схем детектирования на основе времени. Понятно, как отделить стратегию этих механизмов, имеющую важное значение для безопасности сети, от мелких деталей, не оказывающих существенного влияния. Приведенные здесь требования могут не оказаться подходящими для всех случаев. В частности, рекомендации раздела 4 относятся к базовому случаю, но в некоторых конкретных ситуациях могут быть более гибкие в плане обнаружения потерь решения, учитывающие аспекты конкретной среды (например, при работе по одному физическому каналу или в ЦОД с единым контролем). Поэтому в некоторых случаях могут быть полезны и даже необходимы варианты, отклонения или совершенно иные решения по детектированию потерь на основе времени. Этот документ следует рассматривать как принятый по умолчанию вариант, а не универсальное решение во всех ситуациях.

Добавление «зонтика» требований к имеющимся спецификациям по сути не является аккуратным решением и возникает риск несовместимости как с прошлыми, так и с будущими механизмами. Поэтому здесь приведены несколько допущений о связи этого документа с имеющимися и будущими спецификациями.

- Документ не обновляет и не отменяет имеющиеся RFC. Прежние спецификации в общем случае соответствуют требованиям документа и отражают согласованное мнение сообщества, поэтому сохранены.
- Требования документа нацелены на обеспечение защиты сети, поэтому их **следует** применять в будущих механизмах обнаружения потери пакетов на основе времени.
- Требования документа применимы не во всех случаях, поэтому в будущем могут потребоваться варианты и отклонения (поэтому применен термин **следует**). Однако несовместимости **должны** быть (a) объяснены и (b) получить согласие сообщества.

3. Область действия

Описанные в документе принципы не зависят от протокола и широко применимы. Ниже представлены заявления о применимости требований, приведенных в разделе 4.

(S.1) Хотя в протоколах применяется множество таймеров (от контроля скорости до обнаружения отказов в соединениях и т. п.), здесь рассматривается лишь обнаружение потерь.

(S.2) Приведенные здесь требования к механизмам обнаружения потерь на основе времени предназначены для первичного (последняя надежда) механизма детектирования потерь, независимо от того, является он единственным способом восстановления потерь или применяется совместно с другими механизмами.

Хотя для таких простых протоколов, как DNS [RFC1034] [RFC1035], достаточно простого детектора потерь, более сложные протоколы часто применяют более совершенные механизмы обнаружения потерь для повышения производительности. Например, в TCP и SCTP имеются методы обнаружения (и восстановления) потерь на основе явного обобщения состояния конечной точки [RFC2018] [RFC4960] [RFC6675]. Такие механизмы часто обеспечивают более своевременное и точное обнаружение потерь, нежели детекторы на основе времени. Однако эти механизмы не избавляют от необходимости поддерживать тайм-аут повтора (retransmission timeout) или RTO, как отмечено в разделе 1, и в конечном счете могут полагаться лишь на прохождение времени для детектирования потери. Иными словами, нет возможности рассчитывать на прибытие подтверждения отправителю данных как на указание пакетов, которые не пришли к получателю. В таких случаях все равно нужен детектор на основе времени, который сработает в крайнем случае.

Отметим также, что некоторые недавние предложения включают время как часть метода обнаружения потерь в качестве агрессивного детектирования первой потери в некоторых ситуациях или вместе с обобществлением состояний конечных точек [DCCM13] [CCDJ20] [IS20]. Хотя эти механизмы могут способствовать своевременному восстановлению, протокол в конечном итоге опирается на более консервативный таймер для обеспечения надежности при выходе этих механизмов из строя. Требования этого документа напрямую применимы лишь к крайнему варианту обнаружения потерь (last-resort). Однако предполагается, что многие из требований послужат полезным руководством для менее агрессивных таймеров, не предназначенных для крайних случаев.

(S.3) Требования этого документа относятся лишь к взаимодействию между парами конечных точек по индивидуальным (unicast) адресам. Протоколы группового взаимодействия (например, [RFC5740]) явно выходят за рамки документа.

Такие протоколы, как SCTP [RFC4960] и Multipath TCP (MP-TCP) [RFC6182], использующие unicast-адресацию для множества конечных точек, могут применять требования документа при условии отслеживания состояний и независимого выполнения требований каждой точкой. Т.е. при взаимодействии хоста A с хостами B и C хост A должен применять независимое детектирование потерь на основе времени для трафика, передаваемого B и C.

(S.4) В некоторых случаях общее состояние используется несколькими соединениями или потоками (например, [RFC2140] и [RFC3124]). Состояние, относящееся к обнаружению потерь по времени, часто считается доступным для совместного использования. Такие ситуации вызывают вопросы, которые простой механизм обнаружения связанных с потоком потерь по времени, рассматриваемый здесь, не решает (например, продолжительность сохранения состояний между соединениями). Поэтому совместное использование потоками общей информации о потерях на основе времени выходит за рамки документа, хотя к нему и применимы общие принципы раздела 4.

4. Требования

Здесь приведены требования, применимые при разработке основных (primary) или крайних (last-resort) механизмов обнаружения потерь на основе времени. По историческим причинам и для простоты описания время между отправкой пакета и фиксацией его потери по отсутствию подтверждения называется тайм-аутом повтора (retransmission timeout или RTO). По истечении RTO без подтверждения доставки отправитель может в суверенностью считать пакет потерянным. Однако, как было отмечено выше, обнаруженную потерю не требуется восстанавливать (т. е. потеря принимается для контроля перегрузок, но не для обеспечения гарантий доставки).

(1) Как отмечено выше, обнаружение потери происходит, когда отправитель не получает подтверждения доставки в ожидаемом интервале времени. В отсутствие информации о задержке на пути, начальное значение RTO **должно** быть не меньше 1 секунды.

Корректность имеет важнейшее значение при передаче в сеть с неизвестными свойствами по ряду причин.

- Преждевременное обнаружение потери может вызвать ложные повторы, усугубляющие проблемы уже перегруженной сети.
- Преждевременное обнаружение потери может привести к неоправданному снижению разрешенной отправителю скорости передачи, поскольку скорость уже достаточно мала на этом этапе взаимодействия. Восстановление после такого снижения скорости может быть достаточно долгим.
- Как отмечено выше, иногда использование основанного на времени обнаружения потерь и повтора передачи может приводить к неоднозначности определения задержки на пути через сеть. Поэтому особо важно, чтобы первое измерение задержки в сети не включало неоднозначности и можно было создать базу для последующего взаимодействия.

Конкретная константа (1 сеунда) получена из анализа времени кругового обхода в Internet (RTT), приведенного в Приложении A [RFC6298].

(2) Здесь заданы 4 требования, относящиеся к установке ожидаемого интервала подтверждения доставки.

Часто измерение времени, требуемого для подтверждения доставки, воспринимается как определение RTT для сетевого пути. RTT - это минимальное время, которое требуется для подтверждения доставки, которое

часто зависит от поведения протокола в части скорости генерации подтверждения при доставке. Например, это относится к RTO, используемому в TCP [RFC6298] и SCTP [RFC4960]. Однако это иной раз вводит в заблуждение и ожидаемую задержку лучше означит как время обратной связи (feedback time или FT). Иными словами, ожидаемое время не всегда отражает свойства сети и может включать дополнительную задержку, которую следует учитывать отправителю.

Рассмотрим, например, UDP-запрос DNS от клиента к рекурсивному распознавателю [RFC1035]. При обслуживании запроса из кэша распознавателя, время обратной связи (FT) будет близко к RTT между клиентом и распознавателем. Однако при отсутствии записи в кэше распознаватель будет запрашивать нужную информацию у одного или нескольких полномочных серверов DNS, что приведет к тредно оцениваемому росту FT по сравнению с RTT между клиентом и распознавателем.

Поэтому требования выражаются в терминах FT. Для простоты описания по-прежнему используется RTO в качестве индикатора интервала между отправкой пакета и принятием решения о потере, независимо от повтора передачи пакета.

(a) Для установки RTO **следует** использовать несколько наблюдений FT, если они доступны.

Иными словами, значение RTO должно представлять эмпирически доступное разумное время, в течение отправителю следует ждать подтверждения доставки, прежде чем принимать решение о потере данных. Пути в сети по природе динамичны, поэтому важно учитывать в RTO несколько недавних измерений FT.

Например, TCP RTO [RFC6298] удовлетворяет этим требованиям благодаря использованию экспоненциально взвешенного скользящего среднего значения (EWMA[1]) для объединения множества измерений FT в «сглаженное время RTT». Во имя консервативности TCP идет дальше, включая явный учет дисперсии при расчете RTO.

Несмотря на то, что использование нескольких измерений FT очень важно для учета динамики задержки в пути, здесь явно не задается число и срок действия таких измерений для расчета RTO, поскольку это может зависеть от ситуации и задач конкретного детектора потерь.

Измерения FT выполняются на основе обмена пакетами между партнерами. Разработчикам протоколов (особенно новых) рекомендуется обеспечивать сложность подделки обратной связи злоумышленниками в сети с целью исказить оценку хостом значения FT. В идеале все сообщения следует защищать криптографически, но это не всегда возможно (особенно в устаревших протоколах), поэтому рекомендуется использовать в пакетах разумный объем случайных значений.

(b) Измерение FT **следует** выполнять и включать в RTO не менее 1 раза за период RTT или с частотой обмена пакетами, если пакеты передаются с интервалами больше RTT.

Измерения в Internet показывают, что однократное измерение FT для соединения TCP приводит к достаточно плохой работе механизма RTO [AP99], поэтому введено требование многократного измерения FT в течение всего времени связи.

Например, TCP может оценивать FT 1 раз в интервале RTT или для каждого приема подтверждения с временной меткой [RFC7323]. В [AP99] показано, что оба варианта дают близкие оценки RTO.

(c) Оценки FT **можно** делать не только на основе обмена данными.

Некоторые протоколы по тем или иным причинам передают не только данные, но и служебные сообщения keepalive, heartbeat, управляющие сообщения. Задержки при таких обменах могут применяться при оценке FT для механизма RTO в той мере, в какой они отражают задержки при обмене данными. Такие измерения могут помочь протоколам сохранить точность RTO при возникновении перерывов в обмене данными. Однако с учетом того, что задержки этих сообщений могут отличаться от задержки при передаче данных, они могут быть полезны не всегда.

(d) Механизму RTO **недопустимо** применять неоднозначные измерения FT.

Предположим, что были переданы две копии пакета X в моменты t0 и t1, затем в момент t2 отправитель получил подтверждение доставки X. В некоторых случаях невозможно узнать, какая из копий X вызвала подтверждение, поэтому значением FT может быть как t2-t1, так и t2-t0. В такой ситуации реализации **недопустимо** использовать любое из этих значений FT для обновления RTO ([KP87] и [RFC6298]).

В некоторых случаях при отправке двух копий данных можно различить, какую из них подтверждает принятое сообщение ACK. Например, временные метки TCP [RFC7323] позволяют точно установить подтвержденный пакет. Неоднозначности не возникает и измерение пригодно для обновления RTO.

(3) Потеря, обнаруженная механизмом RTO, **должна** служить индикацией перегрузки в сети и вызывать корректировку скорости передачи стандартным механизмом (например, TCP сжимает окно насыщения до одного пакета [RFC5681]). Это обеспечивает защиту сети.

Исключением являются случаи, когда стандартизованный IETF механизм определяет, что данная потеря не связана с перегрузкой (например, повреждение пакета), поэтому контроль перегрузки включать не нужно. Кроме того, действия по контролю перегрузки на основе детектирования потерь по времени могут быть отменены, когда стандартный механизм постфактум определяет, что потеря не связана с перегрузкой (например, [RFC5682]).

(4) При каждом использовании RTO для обнаружения потери значение RTO **должно** экспоненциально увеличиваться, чтобы следующее срабатывание происходило через более долгий интервал. Изменение тайм-аута **следует** отменять, если (a) последующая передача произошла без потерь или (b) в течение RTO не было обнаружено дополнительных потерь. Первый вариант обычно наступает быстрее, а второй относится к случаям, когда потеря обнаружена но не устранена. Это обеспечивает защиту сети.

[1]Exponentially weighted moving average.

Для RTO **можно** задать максимальное значение, которое **недопустимо** делать меньше 60 секунд, как указано в [RFC6298].

Как и в случае (3), имеется исключение, если стандартизованный IETF механизм определяет, что потеря не связана с перегрузкой.

5. Обсуждение

Отметим, что исследования показали фундаментальность противоречия между своевременностью и точностью при обнаружении проблем по времени в контексте TCP [AP99]. Т. е. более энергичное применение RTO (например, изменение прироста EWMA, снижение минимального RTO и т. п.) может ускорить обнаружение действительной потери. Однако при этом будет больше ошибочных срабатываний, когда отмеченные потери не будут таковыми на деле. Поэтому максимальная энергичность, разрешаемая приведенными выше требованиями не будет лучшим решением, ведь детектирование потерь (даже ложное) требует реакции на перегрузку, снижающей в итоге скорость передачи.

Хотя противоречие между точностью и своевременностью детектирования является фундаментальным, его важность можно снизить, если отправитель может обнаружить и скомпенсировать ложные срабатывания детектора потерь. Для этого было предложено несколько механизмов, таких как Eifel [RFC3522], F-RTO[1] [RFC5682], DSACK[2] [RFC2883] [RFC3708]. Применение таких механизмов позволяет отправителю данных реагировать быстрее, но без сопутствующих издержек, связанных с ошибочным детектированием потерь.

Следует также отметить, что в дополнение к экспериментам, описанным в [AP99], реализация Linux TCP много лет использует различные нестандартные механизмы RTO без каких-либо серьезных проблем (например, использование коэффициентов усиления EWMA, отличных от [RFC6298]). Кроме того, во многих реализациях TCP используется минимальное значение RTO в установившемся состоянии, которое меньше 1 секунды, заданной в [RFC6298]. Хотя эти отклонения от стандартов могут приводить к росту числа ложных обнаружений потерь (согласно [AP99]), неизвестно о каких-либо значимых проблемах с безопасностью сетей, вызванных изменением минимального значения RTO. Это учтено в последней рекомендации раздела 4, где не задано минимальное значение RTO.

В заключение следует отметить, что хотя более энергичное поведение реализаций может вести к росту числа ненужных повторов, приведенные выше требования не будут работать в той части, где требуется экспоненциальное снижение тайм-аута и уменьшение скорости передачи. Поэтому предоставление разработчикам большей свободы по сравнению с действующими спецификациями IETF для механизмов RTO не открывает ворота для такого энергичного поведения. Это поведение имеет оборотную сторону и механизм сохраняет стимулы к реализации корректного поведения.

6. Вопросы безопасности

Этот документ не меняет свойств безопасности основанных на времени механизмов обнаружения потерь. Рассмотрение этого вопроса в контексте TCP приведено в [RFC6298].

7. Взаимодействие с IANA

Документ не требует действий со стороны IANA.

8. Литература

8.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Дополнительная литература

[AP99] Allman, M. and V. Paxson, "On Estimating End-to-End Network Path Properties", Proceedings of the ACM SIGCOMM Technical Symposium, September 1999.

[CCDJ20] Cheng, Y., Cardwell, N., Dukkipati, N., and P. Jha, "The RACK-TLP loss detection algorithm for TCP", Work in Progress, Internet-Draft, draft-ietf-tcpm-rack-13, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-tcpm-rack-13>.

[DCCM13] Dukkipati, N., Cardwell, N., Cheng, Y., and M. Mathis, "Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail Losses", Work in Progress, Internet-Draft, draft-dukkipati-tcpm-tcp-loss-probe-01, 25 February 2013, <https://tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe-01>.

[IS20] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", Work in Progress, Internet-Draft, draft-ietf-quic-recovery-32, 20 October 2020, <https://tools.ietf.org/html/draft-ietf-quic-recovery-32>.

[Jac88] Jacobson, V., "Congestion avoidance and control", ACM SIGCOMM, DOI 10.1145/52325.52356, August 1988, <https://doi.org/10.1145/52325.52356>.

[KP87] Karn, P. and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[1]Forward RTO-Recovery - ускоренное восстановление RTO.
[2]Duplicate Selective Acknowledgement - селективное подтверждение дубликатов.

[RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, DOI 10.17487/RFC2018, October 1996, <https://www.rfc-editor.org/info/rfc2018>.

[RFC2140] Touch, J., "TCP Control Block Interdependence", RFC 2140, DOI 10.17487/RFC2140, April 1997, <https://www.rfc-editor.org/info/rfc2140>.

[RFC2883] Floyd, S., Mahdavi, J., Mathis, M., and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", RFC 2883, DOI 10.17487/RFC2883, July 2000, <https://www.rfc-editor.org/info/rfc2883>.

[RFC3124] Balakrishnan, H. and S. Seshan, "The Congestion Manager", RFC 3124, DOI 10.17487/RFC3124, June 2001, <https://www.rfc-editor.org/info/rfc3124>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3522] Ludwig, R. and M. Meyer, "The Eifel Detection Algorithm for TCP", RFC 3522, DOI 10.17487/RFC3522, April 2003, <https://www.rfc-editor.org/info/rfc3522>.

[RFC3708] Blanton, E. and M. Allman, "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions", RFC 3708, DOI 10.17487/RFC3708, February 2004, <https://www.rfc-editor.org/info/rfc3708>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC5682] Sarolahti, P., Kojo, M., Yamamoto, K., and M. Hata, "Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP", RFC 5682, DOI 10.17487/RFC5682, September 2009, <https://www.rfc-editor.org/info/rfc5682>.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, DOI 10.17487/RFC5740, November 2009, <https://www.rfc-editor.org/info/rfc5740>.

[RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, DOI 10.17487/RFC6182, March 2011, <https://www.rfc-editor.org/info/rfc6182>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/info/rfc6298>.

[RFC6675] Blanton, E., Allman, M., Wang, L., Jarvinen, I., Kojo, M., and Y. Nishida, "A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP", RFC 6675, DOI 10.17487/RFC6675, August 2012, <https://www.rfc-editor.org/info/rfc6675>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <https://www.rfc-editor.org/info/rfc7323>.

Благодарности

Этот документ основан на многолетнем обсуждении с Ethan Blanton, Sally Floyd, Jana Iyengar, Shawn Ostermann, Vern Paxson, а также с членами рабочих групп TCPM и TCPIMPL. Полезные комментарии к предварительным вариантам документа предоставили Ran Atkinson, Yuchung Cheng, David Black, Stewart Bryant, Martin Duke, Wesley Eddy, Gorry Fairhurst, Rahul Arvind Jadhav, Benjamin Kaduk, Mirja Kühlewind, Nicolas Kuhn, Jonathan Looney, Michael Scharf.

Адрес автора

Mark Allman

International Computer Science Institute

2150 Shattuck Ave., Suite 1100

Berkeley, CA 94704

United States of America

Email: mallman@icir.org

URI: https://www.icir.org/mallman

Перевод на русский язык

Николай Малых

nmalykh@protocols.ru