Network Working Group
Request for Comments: 4341
Category: Standards Track

S. Floyd
ICIR
E. Kohler
UCLA

March 2006

Профиль контроля насыщения CCID 2: TCP-like Congestion Control для протокола DCCP

Profile for Datagram Congestion Control Protocol (DCCP)

Congestion Control ID 2: TCP-like Congestion Control

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Аннотация

Этот документ содержит профиль механизма контроля насыщения CCID 2 (TCP-like Congestion Control[1]) для протокола DCCP[2]. Механизм CCID 2 следует использовать отправителям, которые хотят использовать всю доступную полосу в среде с быстро изменяющимися условиями и могут адаптироваться к внезапным изменениям размера окна насыщения, характерным для механизма AIMD[3] в TCP.

Оглавление

1. Введение....1
2. Использование терминов....2
3. Применение....2
3.1. Связь с TCP....2
3.2. Пример полусоединения....2
4. Организация соединения....3
5. Контроль насыщения для пакетов данных....3
5.1. Отклик на периоды бездействия и ограничений от приложения....4
5.2. Реакция на отбрасывание данных и медленных получателей....4
5.3. Размер пакетов....4
6. Подтверждения....4
6.1. Контроль насыщения для подтверждений....5
6.1.1. Детектирование потерянных и помеченных подтверждений....5
6.1.2. Изменение Ack Ratio....5
6.2. Подтверждения подтверждений....5
6.2.1. Определение бездействия....6
7. Явное уведомление о насыщении....6
8. Опции и возможности....6
9. Вопросы безопасности....6
10. Взаимодействие с IANA....6
10.1. Коды сброса....6
10.2. Типы опций....6
10.3. Номера свойств....6
11. Благодарности....6
Приложение A. Алгоритм снижения Ack Ratio....6
Приложение B. Влияние неточного учёта потерь на Ack Ratio....7
Нормативные документы....7
Дополнительная литература....8

1. Введение

Этот документ содержит профиль механизма контроля насыщения CCID 2 (TCP-like Congestion Control) для протокола DCCP[RFC4340]. Протокол DCCP использует идентификаторы механизмов контроля насыщения или CCID для того, чтобы указать механизм контроля насыщения, применяемый в полусоединении.

Механизм контроля в стиле TCP передаёт данные с использованием механизма, близкого к механизму контроля насыщения в TCP и включающего вариант селективных подтверждений (SACK) [RFC2018, RFC3517]. Механизм CCID 2

[1]Контроль насыщения в стиле TCP
[2]Datagram Congestion Control Protocol
[3]Additive Increase Multiplicative Decrease – аддитивное увеличение и мультипликативное уменьшение.

подходит для отправителей, которые могут адаптироваться к внезапным изменениям размера окна насыщения, типичным для алгоритма AIMD в TCP, и особенно полезен для приложений, которые хотят воспользоваться всей доступной полосой в среде с быстро изменяющимися условиями. Более подробное описание применения механизма содержится в главе 3.

2. Использование терминов

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

Полусоединение DCCP включает данные приложения, передаваемые одной конечной точкой, и соответствующие подтверждения, передаваемые другой конечной точкой. Термины HC-Sender и HC-Receiver обозначают конечные точки, передающие данные приложения и подтверждения, соответственно. Поскольку CCID используется на уровне полусоединения, в этом документе вместо HC-Sender мы будем иногда писать просто «отправитель», а вместо HC-Receiver – «получатель». Более подробную информацию можно найти в [RFC4340].

Для простоты мы будем предполагать, что отправители передают пакеты DCCP-Data, а получатели – пакеты DCCP-Ack. Пакеты DCCP-DataAck относятся к обеим категориям.

Термины «маркированный ECN» и «маркированный» относятся к пакетам, содержащим в себе маркер насыщения ECN Congestion Experienced, если явно не указано иное.

3. Применение

CCID 2 (TCP-like Congestion Control[1]) подходит для потоков DCCP, которым хочется на продолжительное время получить максимально возможную пропускную способность согласованно со сквозным контролем насыщения. Потоки CCID 2 должны также быть устойчивы к значительным вариациям характеристик скорости передачи контроля насыщения AIMD, включая снижение вдвое окна насыщения по факту возникновения перегрузки.

Приложениям, которым просто нужно передать как можно больше данных за минимально возможное время, следует использовать CCID 2. Этот метод отличается от CCID 3 (TFRC[2]) [RFC4342], который подходит для потоков, предпочитающих минимизировать скачки скорости передачи. Например, CCID 2 предпочтительней CCID 3 для потоковых приложений, которые буферизуют значительный объем данных на приёмной стороне до их реального вывода, что обеспечивает устойчивость к значительным колебаниям скорости передачи. Такие приложения предпочтут DCCP CCID 2 обычному протоколу TCP, возможно с некоторым повышением уровня надёжности в самих приложениях. CCID 2 также будет предпочтительней CCID 3 для приложений, вдвое снижающих скорость передачи в ответ на перегрузку, поскольку в этом случае не будет возникать конфликтов на уровне приложения.

Дополнительным преимуществом CCID 2 является то, что похожие на TCP механизмы контроля насыщения достаточно хорошо известны и динамика трафика подобна динамике для TCP. Хотя исследовательское сообщество продолжает изучать динамику TCP после того, как в течение 15 он является доминирующим транспортным протоколом в Internet, для некоторых приложений может оказаться предпочтительной хорошо известная динамика похожего на TCP контроля насыщения по сравнению с более новыми механизмами такого контроля, которые ещё не получили широкого распространения в Internet.

3.1. Связь с TCP

Описанные здесь механизмы контроля насыщения весьма похожи на стандартизованные IETF механизмы для использования в TCP на основе SACK и частично опираются на имеющиеся документы TCP [RFC793], [RFC2581], [RFC3465] и [RFC3517]. Контроль насыщения в TCP продолжает развиваться, но реализациям CCID 2 **следует** ждать явных обновлений CCID 2, а не отслеживать непосредственно развитие TCP.

Различия между CCID 2 и прямым контролем насыщения TCP включают перечисленные ниже аспекты.

- CCID 2 применяет контроль насыщения к подтверждениям - для использования в TCP такой механизм ещё не стандартизован.
- DCCP представляет собой протокол передачи дейтаграмм, поэтому некоторые параметры, задаваемые для TCP в байтах (например, окно насыщения cwnd), в DCCP задаются числом пакетов.
- Как протокол без гарантий доставки, DCCP никогда не повторяет передачу пакетов, поэтому был разработан механизм контроля насыщения, позволяющий отличать новые пакет от повторов в контексте протокола DCCP.

3.2. Пример полусоединения

Этот пример показывает процесс создания полусоединения с использованием похожего на TCP контроля насыщения CCID 2. Пример не является нормативным и служит для иллюстрации.

1. Отправитель передаёт пакеты DCCP-Data, при этом число переданных пакетов определяется окном насыщения cwnd, как в TCP. В каждом пакете DCCP-Data используется порядковый номер. Отправитель также передаёт опцию Ack Ratio, задающую число пакетов данных, «покрываемых» подтверждением Ack от получателя (по умолчанию Ack Ratio = 2). Поле CCVal в заголовке DCCP имеет значение 0.

   Предположим, что полусоединение поддерживает ECN[3] (по умолчанию ECN Incapable = 0) и каждый пакет DCCP-Data передаётся с полем ECN Capable, имеющим значение ECT(0) или ECT(1), как описано в [RFC3540].

2. Получатель передаёт пакет DCCP-Ack подтверждающий доставку пакетов данных для каждых Ack Ratio пакетов данных, переданных отправителем. Каждый пакет DCCP-Ack использует порядковый номер и содержит Ack Vector.

[1]Похожий на TCP контроль насыщения.
[2]TCP-Friendly Rate Control - дружественный к TCP контроль скорости.
[3]Explicit Congestion Notification - явный контроль насыщения.

Порядковый номер, подтверждаемый в DCCP-Ack, является старшим из порядковых номеров в принятых пакетах; это не кумулятивное подтверждение, подобное используемому в TCP.

Получатель возвращает набор полученных ECN Nonce с помощью опции Ack Vector, позволяя отправителю вероятностно проверить корректность поведения получателя. Пакеты DCCP-Ack от получателя передаются, как поддерживающие ECN (ECN Capable), поскольку отправитель будет контролировать скорость подтверждений подобно TCP, используя опцию Ack Ratio. Получателю нет необходимости проверять значения nonce в своих пакетах DCCP-Ack, поскольку отправитель не сможет получить существенных преимуществ в результате некорректного представления скорости подтверждений.

3. Отправитель продолжает передачу пакетов DCCP-Data под контролем окна насыщения. При получении пакетов DCCP-Ack отправитель проверяет свои Ack Vector для определения маркированных или отброшенных пакетов данных и соответствующей подстройки своего окна насыщения. Поскольку этот транспорт не обеспечивает гарантий, отправитель не повторяет передачу отброшенных пакетов.

4. Поскольку в пакетах DCCP-Ack используются порядковые номера, у отправителя есть некоторая информация о потерянных или промаркированных пакетах DCCP-Ack. Отправитель отвечает на потерю или маркировку пакетов DCCP-Ack изменением значения Ack Ratio отправляемого получателю.

5. Отправитель подтверждает получателю пием подтверждений хотя бы один раз в окне насыщения. Если оба полусоединения активны, подтверждение отправителем приёма подтверждений от получателя включается в подтверждение отправителем приёма от получателя пакетов данных. Если обратное полусоединение бездействует, отправитель передаёт по крайней мере один пакет DCCP-DataAck на окно насыщения.

6. Отправитель оценивает интервалы кругового обхода путём отслеживания времени возврата подтверждений, как это делает TCP, или с помощью явной опции Timestamp и рассчитывает значение TimeOut (TO), как в TCP рассчитывается значение RTO (Retransmit Timeout - тайм-аут повтора). Значение TO определяет, когда может быть передан новый пакет DCCP-Data, если отправитель был ограничен окно насыщения и от получателя не приходило обратной связи.

4. Организация соединения

Использование Ack Vector **обязательно** на полусоединениях CCID 2, поэтому отправитель **должен** передать опцию Change R(Send Ack Vector, 1) в процессе организации соединения. Отправителю **недопустимо** передавать данные, пока он не получит соответствующей опции Confirm L(Send Ack Vector, 1) от получателя. Исключением является **возможность** передавать пакеты DCCP-Request.

5. Контроль насыщения для пакетов данных

Механизмы контроля насыщения CCID 2 основаны на методах, используемых TCP на основе SACK [RFC3517], поскольку Ack Vector обеспечивает всю информацию, которая может быть передана в опциях SACK.

Отправитель данных CCID 2 поддерживает 3 целочисленных параметра (число пакетов), описанных ниже.

1. Окно насыщения cwnd задаёт максимальное число пакетов данных, которые могут находиться в сети в любой момент времени (пакетами данных считаются любые пакеты DCCP, содержащие данные пользователя, - DCCP-Data, DCCP-DataAck, а в некоторых случаях DCCP-Request и DCCP-Response).

2. Порог замедленного старта (slow-start) ssthresh, определяющий подстройку значения cwnd.

3. Параметр pipe, показывающий оценку отправителем числа пакетов данных, остающихся в сети.

Эти параметры являются изменяемыми, а их начальные значения определяются в соответствии с поведением, заданным для SACK TCP, за исключениям того, что значения задаются в пакетах, а не байтах. В оставшейся части этого параграфа даны более конкретные рекомендации.

Отправитель **может** передать пакет данных, когда pipe < cwnd, но передача **недопустима**, если pipe >= cwnd. Каждый отправленный пакет данных увеличивает значение pipe на 1.

Отправитель уменьшает значение pipe, когда он делает вывод о том, что пакет покинул сеть, т. е. был доставлен получателю или отброшен. Ниже рассмотрены конкретные ситуации.

1. Пакет данных подтверждён. Отправитель уменьшает pipe на 1 для каждого пакета данных, недавно подтверждённого в принятом (Ack Vector State 0 или State 1) DCCP-Ack.

2. Пакет данных отброшен. Отправитель уменьшает pipe на 1 для каждого пакета данных, который он может считать потерянным в соответствии с DCCP-эквивалентом дубликата подтверждения TCP. Это зависит от параметра NUMDUPACK, задающего число дубликатов подтверждений, требуемых для фиксации потери. Для параметра NUMDUPACK устанавливается значение 3, как принято сейчас в TCP. Пакет P считается потерянным, а не задержанным, когда не менее NUMDUPACK пакетов, переданных после P были подтверждены (Ack Vector State 0 или 1) получателем. Отметим, что пакеты подтверждения, следующие за пропуском, могут быть пакетами DCCP-Ack или другими пакетами без данных.

3. Тайм-аут при передаче. Наконец, отправителю нужны тайм-ауты передачи, обрабатываемые подобно тайм-аутам повтора TCP, когда потеряно целое окно пакетов. Отправитель оценивает время кругового обхода не более одного раза на окно данных и применяет алгоритмы TCP для поддержки среднего значения периода кругового обхода и значения тайм-аута [RFC2988] (если для расчёта было использовано более одного измерения за период кругового обхода, нужно будет отрегулировать весовые коэффициенты усреднителей, чтобы обеспечить эффективный вывод среднего времени кругового обхода из множества измерений). Поскольку DCCP не повторяет передачу данных, здесь не требуется рекомендуемого TCP минимального тайм-аута в 1 секунду. Экспоненциальное изменение таймера совпадает с используемым в TCP. При возникновении тайм-аута передачи отправитель устанавливает pipe=0. Настройка cwnd и ssthresh описана ниже.

Отправителю **недопустимо** декрементировать pipe более одного раза на пакет данных. Например, **недопустимо** влияние на pipe действительных подтверждений. Отправителю также **недопустимо** уменьшать pipe при получении подтверждения пакета, который считался потерянным. Кроме того, отправителю **недопустимо** уменьшать pipe для пакетов, не содержащих данных (таких как DCCP-Ack), даже если Ack Vector содержит информацию о них.

Фауты перегрузки заставляют CCID 2 снижать своё окно насыщения. Событие перегрузки включает хотя бы одну потерю или помеченный пакет. Как и в TCP, две потери или маркировки считаются одним событием, если второй пакет был передан до обнаружения потери или маркировки первого. В качестве аппроксимации отправитель может считать две потери или маркировки частью одного события, если пакеты были переданы в интервале между двумя оценками RTT с использованием текущей оценки RTT в моменты передачи пакетов. Для каждого факта перегрузки, указанного явно как подтверждение Ack Vector State 1 (маркировка ECN) или выведенного из подтверждений дубликатов, значение cwnd делится пополам, а затем для ssthresh устанавливается новое значение cwnd. Значение cwnd никогда не бывает меньше 1. После тайм-аута для порога slow-start устанавливается значение cwnd/2, затем устанавливается cwnd = 1. При делении пополам значения cwnd и ssthresh округляются в сторону уменьшения, если при этом cwnd не становится меньше 1, а ssthresh меньше 2.

Когда cwnd < ssthresh (это означает режим slow-start у отправителя), окно насыщения увеличивается на 1 пакет для каждой пары недавно подтверждённых пакетов данных с Ack Vector State 0 (без маркера ECN) до максимального значения Ack Ratio/2 пакетов на подтверждение. Это изменённый вариант Appropriate Byte Counting [RFC3465], соответствующий текущему стандарту TCP (не включает подсчёт байтов), но позволяющий CCID 2 увеличиваться так же быстро, как и TCP, когда Ack Ratio в CCID 2 больше принятого по умолчанию значения 2. Когда cwnd >= ssthresh, окно насыщения увеличивается на 1 пакет для каждого окна данных, подтверждённого без потери и маркировки пакетов. Параметр cwnd инициализируется значением не более 4 пакетов для новых соединений в соответствии с правилами [RFC3390], параметр ssthresh инициализируется произвольно высоким значением.

Отправители **могут** использовать задание темпа на основе скорости при отправке множества пакетов данных, освобождённых одним пакетом подтверждения, вместе передачи всех освобождённых пакетов данных одним махом.

5.1. Отклик на периоды бездействия и ограничений от приложения

CCID 2 разработан с учётом максимального соответствия механизмам контроля перегрузок TCP, но эти механизмы TCP не имеют полной стандартизации откликов контроля насыщения на периоды бездействия (пакеты данных не передаются) или ограничения передачи от приложений (скорость отправки меньше разрешённой cwnd). В этом параграфе дано краткое руководство по стандартам TCP в этой области.

Для периодов бездействия в соответствии с [RFC2581] отправителю TCP **следует** перейти в состояние slow-start после интервала бездействия, который определяется как интервал, превосходящий значение тайм-аута. [RFC2861][1], имеющий статус Experimental, предлагает более умеренный механизм, где окно насыщения делится пополам за каждый интервал кругового обхода, в течение которого отправитель не передавал данных.

В настоящее время нет стандартов, определяющих использование окна насыщения TCP в периоды ограниченной приложениям передачи данных. В частности, окно насыщения TCP может стать достаточно большим в течение продолжительного интервала без перегрузки, когда отправитель передаёт данные с малой скоростью. [RFC2861], по сути, предлагает не увеличивать окно насыщения TCP в периоды ограниченной приложением передачи данных, когда окно насыщения не используется полностью.

5.2. Реакция на отбрасывание данных и медленных получателей

Опция Data Dropped в DCCP позволяет получателю сообщить, что пакет данных был отброшен на конечном узле до его передачи приложению (например, по причине повреждения пакета или переполнения приёмного буфера). Опция Slow Receiver позволяет получателю сообщить о наличии проблем с пакетами отправителя, хотя ни один из них не был отброшен. Отправители CCID 2 отвечают на эти опции в соответствии с [RFC4340] и приведёнными ниже уточнениями.

- Drop Code 2 (отбрасывание в буфере приёма). Окно насыщения cwnd уменьшается на 1 для каждого пакета, вновь подтверждённого как Drop Code 2, но никогда не снижается до значений меньше 1.
- Отправитель **должен** выйти из состояния slow-start при получении соответствующей опции Data Dropped или Slow Receiver.

5.3. Размер пакетов

CCID 2 оптимизирован для приложений, обычно использующих пакеты фиксированного размера и меняющих скорость их отправки в ответ на перегрузку. CCID 2 не подходит для приложений, которым нужен фиксированный между пакетами и которые могут менять размер пакетов при перегрузке вместо снижения скорости отправки.

CCID 2 поддерживает окно насыщения в пакетах и не увеличивает его при снижении размера пакетов. Однако требуется некоторое внимание к приложениям, использующим CCID 2, которые меняют размер пакетов не в результате перегрузки а в качестве реакции на другие требования прикладного уровня.

Реализации CCID 2 **могут** проверять приложения, которые представляются неправомерно изменяющими размер пакетов. Например, приложение может передавать мелкие пакеты, наращивая их скорость, а затем увеличить размер пакетов для достижения более высокой скорости (предварительное моделирование показывает, что приложения не смогут повысить таким путём общую скорость, поэтому применение таких действий на практике не очевидно [V03]).

6. Подтверждения

Подтверждения CCID 2 обычно передаются отправителем пакетов данных. Каждое требуемое подтверждение **должно** содержать опции Ack Vector, точно указывающие принятые пакеты и наличие маркеров ECN. Данным подтверждения в опциях Ack Vector обычно **следует** учитывать все окно подтверждений (Acknowledgement Window, см. параграф 11.4.2 в [RFC4340]. Любым опциям Data Dropped также **следует** охватывать все окно подтверждений получателя.

[1] Заменён RFC 7661, который также имеет статус экспериментального. *Прим. перев.*

Отправители CCID 2 используют функцию Ack Ratio в DCCP для влияния на скорость, с которой получатель генерирует пакеты DCCP-Ack, что позволяет контролировать перегрузку обратного пути. Это отличается от TCP, где в настоящее время нет контроля насыщения для чистого трафика подтверждений. Контроль перегрузки обратного пути в CCID 2 не пытается быть дружественным к TCP, а лишь старается избежать перегрузок и быть несколько лучше TCP при высокой частоте потери или маркировки пакетов на обратном пути. По умолчанию Ack Ratio = 2 и CCID 2 с таким Ack Ratio ведёт себя подобно TCP с отложенными подтверждениями. В параграфе 11.3 [RFC4340] приведено более подробное описание Ack Ratio, включая связь с темпом отправки подтверждений и пакетов DCCP-DataAck. В параграфе 6.1.1 этого документа описано, как отправитель CCID 2 детектирует потерю и маркировку подтверждений, а в параграфе 6.1.2 описано изменение Ack Ratio.

6.1. Контроль насыщения для подтверждений

При Ack Ratio = R получатель передаёт пакет DCCP-Ack на каждые R пакетов данных (более или менее). Поскольку отправитель передаёт cwnd пакетов данных ха период кругового обхода, скорость подтверждений составляет cwnd/R DCCP-Ack за период кругового обхода. Отправитель поддерживает скорость подтверждений примерно на уровне TCP, выполняя мониторинг потока подтверждений на предмет потери и маркировки пакетов DCCP-Ack и соответственно меняет R. Для каждого RTT с событием перегрузки для DCCP-Ack (потеря или маркировка DCCP-Ack) отправитель снижает вдвое скорость отправки подтверждений путём удваивания Ack Ratio. Для каждого RTT без перегрузки для DCCP-Ack скорость отправки подтверждений увеличивается путём постепенного снижения Ack Ratio.

6.1.1. Детектирование потерянных и помеченных подтверждений

Все пакеты от получателя содержат порядковые номера, чтобы отправитель мог обнаружить потерю и маркировку таких пакетов. Отправитель делает это точно так же, как детектируется потеря пакетов данных. Пакет от получателя считается потерянным после получения как минимум NUMDUPACK пакетов с большими порядковыми номерами.

Пакеты DCCP-Ack обычно малы, поэтому они могут создавать меньшую нагрузку в насыщенной сети по сравнению с пакетами DCCP-Data и DCCP-DataAck. По этой причине Ack Ratio зависит от частоты потери и маркировки пакетов без данных, а не от суммарной частоты потери и маркировки всех пакетов от получателя. Категория пакетов без данных включает пакеты, которые не могут содержать данных приложений - DCCP-Ack, DCCP-Close, DCCP-CloseReq, DCCP-Reset, DCCP-Sync, DCCP-SyncAck. Отправитель может легко отличит маркировку пакетов без данных от иной маркировки. Для потери это сложней, поскольку отправитель не всегда может знать о наличии или отсутствии данных в потерянном пакете. Если у отправителя нет более надёжной информации, ему **следует** считать для целей расчёта Ack Ratio, что каждый потерянный пакет не содержал данных. Более надёжную информацию можно получить с помощью опции DCCP NDP Count, если это нужно (в приложении B рассмотрены издержки, связанные с ошибочным учётом потери пакетов с данными как потери пакетов без данных)

Получатель, реализующему свой контроль перегрузок для подтверждений, независимый от Ack Ratio, **не следует** снижать темп передачи DCCP-Ack в результате потери или маркировки его пакетов данных.

6.1.2. Изменение Ack Ratio

Ack Ratio всегда следует трём ограничениям: (1) является целым числом, (2) не превышает cwnd/2 с округлением вверх, за исключением то, что Ack Ratio = 2 всегда приемлемо, (3) Ack Ratio имеет значение не меньше 2 для окна насыщения размером не меньше 4 пакетов.

Отправитель меняет Ack Ratio с учётом этих ограничений. Для каждого окна насыщения данных с потерянными или маркированными пакетами DCCP-Ack значение Ack Ratio удваивается, а для каждых последовательных cwnd/(R^2 - R) окон насыщения данных без потерь и маркировки DCCP-Ack значение Ack Ratio уменьшается на 1 (приложение A). Изменения Ack Ratio указываются через согласование возможностей (см. параграф 11.3 в [RFC4340]).

При постоянном окне насыщения это даёт скорость передачи Ack близкую к TCP. Конечно, cwnd обычно меняется со временем и динамика достаточно сложна, но близка к TCP. Отправителю рекомендуется использовать последнее значение cwnd при решении вопроса об уменьшении Ack Ratio на 1.

Отправитель не обязан постоянно обновлять Ack Ratio. Например, он **может** ограничить частоту согласования Ack Ratio до одного раза за 4 - 5 периодов кругового обхода или до одного раза за каждые 2 секунды. Отправителю **недопустимо** пытаться согласовать Ack Ratio больше одного раза за время кругового обхода. Кроме того, он **может** установить минимальное значение Ack Ratio = 2 или **может** задать Ack Ratio = 1 для полусоединений с постоянными окнами насыщения в 1 или 2 пакета.

С учётом сказанного, получатель отправляет не менее 1 подтверждения на окно данных при cwnd = 1, и не менее 2 в остальных случаях. Таким образом, получатель может передать 2 подтверждения на окно данных даже при сильном насыщении обратного пути. Отметим, однако, что при достаточно сильной перегрузке все подтверждения будут отбрасываться и отправитель вернётся к экспоненциальному снижению тайм-аута как в TCP. Таким образом, при достаточно сильной перегрузке обратного пути отправитель снижает скорость передачи на прямом пути, что ведёт к снижению скорости передачи на обратном пути.

6.2. Подтверждения подтверждений

Активный отправитель DCCP A **должен** время от времени подтверждать подтверждения своего партнёра DCCP B, чтобы DCCP B мог высвободить состояние Ack Vector. Когда активны оба полусоединения, подтверждения от A для подтверждения B автоматически включаются в подтверждения A для данных от B. Если соединение от B к A бездействует, DCCP A должен время от времени отправлять подтверждения заблаговременно, например, путём отправки пакета DCCP-DataAck с Acknowledgement Number в заголовке.

Активному отправителю **следует** подтверждать подтверждения от получателя по крайней мере 1 раз за период кругового обхода. Конечно, приложение отправителя может «замолчать», но это не создаёт проблемы, поскольку отправитель может ждать сколь угодно долго перед отправкой подтверждения.

6.2.1. Определение бездействия

В этом параграфе описано, как получатель CCID 2 определяет, что соответствующий отправитель не передаёт данные. Общая информация о молчании отправителей приведена в параграфе 11.1 [RFC4340].

Путь T равно большему из значений 0,2 сек. и удвоенное время кругового обхода (получатель может знать это время из своей роли отправителя в другом полусоединении; если это время ему неизвестно, следует использовать принятое по умолчанию RTT = 0,2, как описано в параграфе 3.4 [RFC4340]). Как только отправитель подтверждает Ack Vector от получателя и не передаёт других данных по меньшей мере T секунд, получатель может сделать вывод о молчании сервера. Более точно, получатель делает вывод о молчании отправителя по прошествии не менее T секунд без передачи им каких-либо данных, когда сервер уже подтвердил Ack Vector для всех данных, принятых получателем.

7. Явное уведомление о насыщении

CCID 2 поддерживает явные уведомления о перегрузке (ECN[1]) [RFC3168]. Отправитель будет применять ECN Nonce для пакетов данных, а получатель будет возвращать (эхо) эти nonce в Ack Vector, как указано в параграфе 12.2 [RFC4340]. Информация о помеченных пакетах также будет возвращаться в Ack Vector. Поскольку информация Ack Vector передаётся гарантированно, DCCP флаги TCP в ECN-Echo и Congestion Window Reduced.

Для непомеченных пакетов получатель рассчитывает ECN Nonce Echo как в [RFC3540] и возвращает как часть опций Ack Vector. Получателю **следует** проверять соответствие ECN Nonce Echoe ожидаемым значениям, что обеспечит защиту от случайного или намеренного сокрытия помеченных пакетов.

Поскольку для подтверждений CCID 2 обеспечивается контроль перегрузки, ECN можно также применять для этих подтверждений. В этом случае ECN Nonce не используются, поскольку непросто обеспечить защиту от сокрытия помеченных отправителем пакетов подтверждения, а также потому, что у отправителя нет серьёзных мотивов скрывать частоту маркировки подтверждений.

8. Опции и возможности

Опция DCCP Ack Vector и свойства ECN Capable, Ack Ratio, Send Ack Vector применимы в CCID 2.

9. Вопросы безопасности

Вопросы безопасности DCCP рассмотрены в [RFC4340], а для TCP - в [RFC2581].

В [RFC2581] рассмотрены способы, с помощью которых атакующий может снизить производительность соединений TCP за счёт отбрасывания пакетов и подделки подтверждений дубликатов или подтверждений новых данных. Авторам не известны какие-либо новые проблемы безопасности, связанные с этим документом при использовании похожего на TCP контроля перегрузок.

10. Взаимодействие с IANA

Эта спецификация определяет значение 2 в пространстве имён DCCP CCID, поддерживаемом IANA. Это назначение указано также в [RFC4340]. CCID 2 также добавляет три набора чисел, значения которых должны выделяться IANA - коды сброса (Reset Code) CCID 2, типы опций и номера свойств. Эти диапазоны будут предотвращать «засорение» соответствующих глобальных пространств имён DCCP будущими значениями для CCID 2 (см. параграф 10.3 в [RFC4340]). Однако этот документ не задаёт никаких значений из этих диапазонов, кроме тестов и экспериментального применения [RFC3692]. Документ указывает процедуру Standards Action в соответствии с [RFC2434].

10.1. Коды сброса

Каждая запись в реестре кодов сброса DCCP содержит относящееся к CCID 2 значение Reset Code из диапазона 128-255, краткое описание кода и ссылку на RFC с определением кода. Коды 184-190 и 248-254 выделены для тестов и экспериментов. Оставшиеся значения 128-183, 191-247 и 255 являются резервными и распределять их следует по процедуре Standards Action, которая требует рецензии IESG и публикации RFC со статусом standards-track.

10.2. Типы опций

Каждая запись в реестре типов опций DCCP содержит связанный с CCID 2 тип (число из диапазона 128-255), имя опции и ссылку на RFC с определением типа. Типы 184-190 и 248-254 выделены для тестов и экспериментов. Оставшиеся значения 128-183, 191-247 и 255 являются резервными и распределять их следует по процедуре Standards Action, которая требует рецензии IESG и публикации RFC со статусом standards-track.

10.3. Номера свойств

Каждая запись в реестре номеров свойств DCCP содержит связанный с CCID 2 номер свойства (число из диапазона 128-255), имя свойства и ссылку на RFC с определением номера свойства. Номера 184-190 и 248-254 выделены для тестов и экспериментов. Оставшиеся значения 128-183, 191-247 и 255 являются резервными и распределять их следует по процедуре Standards Action, которая требует рецензии IESG и публикации RFC со статусом standards-track.

11. Благодарности

Авторы благодарны Mark Handley и Jitendra Padhye за помощь в определении CCID 2. Спасибо Mark Allman, Aaron Falk, Nils-Erik Mattsson, Greg Minshall, Arun Venkataramani, Magnus Westerlund и членам рабочей группы DCCP за отклики.

Приложение A. Алгоритм снижения Ack Ratio

В этом приложении обоснован алгоритм роста и снижения Ack Ratio, заданного в параграфе 6.1.2.

[1] Explicit Congestion Notification.

Фаза предотвращения перегрузки TCP снижает cwnd вдвое на каждое окно с насыщением. Точно так же CCID 2 удваивает Ack Ratio для каждого окна с насыщением на обратном пути, снижая примерно вдвое частоту DCCP-Ack.

Фаза предотвращения перегрузки TCP увеличивает cwnd на значение MSS для каждого окна без насыщения. Когда такое поведение применяется к трафику подтверждений, это будет соответствовать росту числа пакетов DCCP-Ack на 1 после каждого окна без перегрузки для пакетов DCCP-Ack. Это условие не выполняется точно, поскольку Ack Ratio является целым числом. Вместо этого нужно увеличивать Ack Ratio на 1 после каждых K без насыщения на обратном пути, где значение K выбирается так, чтобы долгосрочное число пакетов DCCP-Ack на окно насыщения было близко к значению TCP, задаваемому контролем перегрузки AIMD.

В CCID 2 грубое приближение к TCP для трафика подтверждений может быть достигнуто установкой $K = cwnd/(R^2 - R)$, где R - текущее значение Ack Ratio.

Расчёт результата показан ниже

```
R = Ack Ratio = # число пакетов данных / число пакетов подтверждения
W = Congestion Window = # число пакетов данных / окно
W/R = # число пакетов подтверждения / окно
```

Требование. Нужно увеличивать W/R на 1 для каждого окна без перегрузки. Поскольку R можно снижать лишь на 1, определив K так, что после K окон без перегрузки значение W/R + K будет равно W/(R-1).

```
(W/R) + K = W/(R-1)
        K = W/(R-1) - W/R = W/(R^2 - R)
```

Приложение B. Влияние неточного учёта потерь на Ack Ratio

Как указано в параграфе 6.1.1, отправитель зачастую не может определить, содержал ли потерянный пакет данные. Это ограничивает возможность отделить потерю пакетов без данных от других потерь. При отсутствии более точной информации отправитель при расчёте Ack Ratio считает, что все потерянные пакеты не содержали данных. Это может приводить к переоценке числа потерь и слишком большим значениям Ack Ratio, что ведёт к слишком редкой отправке подтверждений. Вся подтверждающая информация будет доставляться (подтверждения DCCP доставляются с гарантией), но подтверждения будут приходить в форме «пиков». При отсутствии той или иной формы контроля темпа отправки на основе скорости это может приводить в росту всплесков (пиков) трафика данных отправителя.

В некоторых случаях проблема слишком большого Ack Ratio и связанного с этим роста пиков трафика данных не возникает. Для получателя DCCP B и отправителя DCCP A эти случаи рассмотрены ниже.

- Проблема не возникает, пока DCCP B не передаёт значительного объёма данных. Когда полусоединение от B к A бездействует или имеет малую скорость, большинство переданных DCCP B пакетов будут фактически чистыми подтверждениями и оценка узлом DCCP A частоты потери DCCP-Ack будет достаточно точной.

- Проблема не возникает, если DCCP B обычно «цепляет» подтверждения к своим пакетам данных. Такие подтверждения не ограничиваются значением Ack Ratio, поэтому они приходят достаточно часто и пиков не будет.

- Проблема не возникает, если скорость передачи DCCP A мала, поскольку пики трафика при малой скорости не создают проблем.

- Проблема не возникает, если скорость передачи DCCP B достаточно высока по сравнению со скоростью передачи DCCP A, поскольку частота потерь от B к A должна быть низкой для поддержки скорости передачи DCCP B. Это ограничивает Ack Ratio разумными значениями даже в том случае, когда DCCP A связывает каждую потерю с DCCP-Ack.

- Проблема не возникает, если DCCP B передаёт опции NDP Count, когда это возможно (Send NDP Count/B имеет значение true). Отправитель в этом случае может использовать опции получателя NDP Count, чтобы корректно отличать потерю пакетов данных от потери DCCP-Ack.

- Проблема не возникает, если DCCP A ускоряет темп передачи своих пакетов данных.

Остаётся случай, когда DCCP B передаёт примерно одинаковой число пакетов с данными и без данных, не используя NDP Count, и вся информация подтверждений содержится в пакетах DCCP-Ack. Оценим возможное влияние слишком высоких значений Ack Ratio в результате учёта потерь пакетов с данными как потерь DCCP-Ack. Для простоты предполагается среда крупномасштабного статистического мультиплексирования, где частота отбрасывания пакетов не зависит от скорости передачи в отдельном соединении.

Предположим, что при корректном учёте узлом DCCP A потери пакетов без данных значение Ack Ratio устанавливается так, что скорости передачи данных и подтверждений от B к A имеют значение D пакетов в секунду. Тогда при некорректном учёте в DCCP A потери пакетов данных как потери пакетов без данных, скорость передачи трафика данных от B к A останется D, а для скорости передачи подтверждений от B к A будет установлено значение f*D (f < 1). Предположим, что частота потери пакетов равна p. Отправитель ошибочно оценивает коэффициент потери пакетов без данных как (pD+pfD)/fD или (эквивалентно) как p(1 + 1/f). Поскольку механизм контроля насыщения для трафика подтверждений приблизительно соответствует TCP и скорости отправки пакетов с данными и без данных будут расти как 1/sqrt(x), где x - частота отбрасывания пакетов, получим

```
fD/D = sqrt(p)/sqrt(p(1 + 1/f)),
```

и

```
f^2 = 1/(1 + 1/f).
```

Решение даёт значение f = 0,62. Если в таком случае отправитель некорректно считает потерю пакетов с данными потерей пакетов без данных, скорость передачи подтверждений снизится до 0,62. Это приведёт к некоторому росту всплесков трафика от A к B, которые могут быть ослаблены использованием опций NDP Count, включением подтверждений в пакеты данных или управлением скоростью передачи данных.

Нормативные документы

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's Initial Window", RFC 3390, October 2002.

[RFC3517] Blanton, E., Allman, M., Fall, K., and L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517, April 2003.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

Дополнительная литература

[RFC2861] Handley, M., Padhye, J., and S. Floyd, "TCP Congestion Window Validation", RFC 2861[1], June 2000.

[RFC3465] Allman, M., "TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465, February 2003.

[RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

[RFC4342] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.

[V03] Arun Venkataramani, August 2003. Citation for acknowledgement purposes only.

Адреса авторов

Sally Floyd
ICSI Center for Internet Research
1947 Center Street, Suite 600
Berkeley, CA 94704
USA
EMail: floyd@icir.org

Eddie Kohler
4531C Boelter Hall
UCLA Computer Science Department
Los Angeles, CA 90095
USA
EMail: kohler@cs.ucla.edu

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).

[1] Заменён RFC 7661. *Прим. перев.*