Network Working Group
Request for Comments: 5348
Obsoletes: 3448
Updates: 4342

S. Floyd
ICIR
M. Handley
University College London
J. Padhye
Microsoft
J. Widmer
DoCoMo

September 2008

Дружественный к TCP контроль скорости (TFRC) - спецификация протокола

TCP Friendly Rate Control (TFRC): Protocol Specification

Статус документа

В этом документе содержится проект стандартного протокола Internet для сообщества Internet и приглашение к дискуссии в целях развития и совершенствования протокола. Информацию о текущем состоянии стандартизации протокола можно найти в текущей версии документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Аннотация

В этом документе приведена спецификация протокола TFRC1, который представляет собой механизм контроля насыщения для потоков с индивидуальной адресацией в среде Internet, обеспечивающей доставку по возможности2. Этот механизм обеспечивает достаточно беспристрастное деление полосы с конкурирующими потоками TCP, но отличается значительно меньшими временными вариациями пропускной способности по сравнению с TCP, что делает этот механизм более подходящим для таких приложений, как телефония или потоковая передача, где важна постоянная скорость потока данных.

Данный документ отменяет действие RFC 3448 и является обновлением RFC 4342.

Оглавление

1. Введение

В этом документе содержится спецификация TFRC - механизма контроля насыщения, предназначенного для потоков с индивидуальной адресацией в среде Internet при одновременной передаче с трафиком TCP [FHPW00]. Вместо задания полного протокола в данном документе просто даётся спецификация механизма контроля насыщения, который может использоваться в транспортных протоколах типа DCCP (Datagram Congestion Control Protocol) [RFC4340] для приложений, включающих сквозной контроль насыщения на прикладном уровне, или в контексте контроля насыщения на оконечных точках [BRS99]. В документе не рассматриваются форматы пакетов и вопросы надёжности доставки. Связанные с реализацией механизма вопросы рассмотрены кратко в главе 8.

Механизм TFRC разработан для обеспечения разумной беспристрастности распределения полосы при конкуренции с потоками TCP. Разумная беспристрастность означает, что скорость передачи отличается от скорости потока TCP при таких же условиях не более, чем вдвое. Однако TFRC обеспечивает существенно меньшие временные вариации пропускной способности по сравнению с TCP, что делает этот механизм более подходящим для телефонии и потоковых приложений, где относительное постоянство скорости передачи играет важную роль.

Платой за более стабильную пропускную способность по сравнению с TCP в условиях конкуренции за полосу является более медленная реакция TFRC на изменение доступной полосы пропускания. Таким образом, TFRC следует использовать лишь в тех случаях, когда приложениям требуется стабильная пропускная способность и, в частности, предотвращение двухкратного снижения скорости передачи, принятого в TCP в ответ на отбрасывание одного пакета. Для приложений, которым просто нужно передать данные за возможно кратчайшее время, рекомендуется использовать TCP или, в тех случаях, когда надёжность не требуется, механизм контроля насыщения AIMD[1] с параметрами, близкими к тем, которые применяются в TCP.

Механизм TFRC разработан для приложений, которые используют пакеты фиксированного размера и меняют скорость передачи таких пакетов в ответ на возникновение перегрузок (насыщения). TFRC может также использоваться, возможно с менее оптимальной производительностью, в приложениях, не использующих фиксированный размер сегмента и меняющих его в соответствии с потребностями приложения (например, видео-приложения).

Для некоторых приложений (например, звуковых) требуется обеспечение фиксированного интервала времени между передачей последовательных пакетов и варьирование размера сегментов (вместо изменения скорости передачи пакетов) в ответ на возникновение перегрузки. Механизм контроля насыщения, предложенный в этом документе, для таких приложений не подходит, однако эту задачу решает механизм TFRC-PS[2], являющийся вариантом TFRC для приложений с фиксированной частотой передачи и изменением размера пакетов при возникновении насыщения. Спецификация TFRC-PS содержится в [RFC4828].

Механизм TFRC основан на работе принимающей стороны с расчётом параметров контроля насыщения (частоты фактов потери пакетов) на стороне получателя, а не отправителя. Такой способ хорош для приложений, в которых отправителем является большой сервер, обслуживающий множество одновременных соединений, а получатели имеют достаточно памяти и процессорных ресурсов для выполнения требуемых расчётов. Кроме того, реализованный на приёмной стороне механизм лучше подходит для контроля насыщения в системах с групповой адресацией. Однако возможна реализация TFRC на серверной стороне, как в профиле CCID-3[3] протокола DCCP [RFC4342].

Этот документ отменяет действие RFC 3448. В транспортном протоколе DCCP[4] [RFC4340] профили CCID-3 [RFC4342] и CCID-4 [CCID-4] задают использование TFRC в соответствии с RFC 3448. Разработчикам CCID-3 и CCID-4 **следует** использовать этот документ взамен RFC 3448 для спецификации TFRC.

Нормативная часть спецификации TFRC приведена в главах 3 - 6. В главе 7 рассматриваются реализации механизма на серверах, в главе 8 - вопросы реализации механизма, а в главе 9 рассмотрены отличия от RFC 3448.

2. Терминология

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].

В Приложении A приводится список использованных в документе технических терминов.

3. Механизм протокола

Для контроля насыщения TFRC напрямую использует уравнение пропускной способности для разрешённой скорости передачи, как функции от частоты фактов потери пакетов и времени кругового обхода. Для беспристрастной конкуренции с TCP, механизм TFRC использует принятое в TCP уравнение для пропускной способности, выражающее

[1] Additive-Increase, Multiplicative-Decrease - аддитивный рост, мультипликативное снижение.
[2] TFRC-PacketSize.
[3] Congestion Control ID 3.
[4] Datagram Congestion Control Protocol - протокол передачи дейтаграмм с контролем насыщения.

скорость передачи TCP, как функцию от частоты фактов потери пакетов, времени кругового обхода и размера сегментов. Определим факт потери, как случай утраты одного или более маркированного пакета из окна данных; маркированными считаются пакеты, помеченные явным индикатором насыщения ECN[1] [RFC3168].

В общем виде механизм контроля насыщения TFRC можно описать следующим образом:

- Получатель определяет частоту фактов потери пакетов и передаёт эту информацию отправителю.
- Отправитель использует эти сообщения от получателя для определения времени кругового обхода (RTT[2]).
- Значения частоты фактов потери и RTT передаются в уравнение пропускной способности TFRC и результирующая скорость передачи ограничена значением не более удвоенной скорости приёма.
- Отправитель подстраивает скорость передачи в соответствии с допустимой скоростью передачи X.

Динамика TFRC чувствительна к способам проведения измерений и применения результатов. В этом документе приводятся рекомендации по конкретному механизму. Возможно использование иных механизмов, но при этом следует понимать, как этот механизм будет воздействовать на динамику TFRC.

3.1. Уравнение для пропускной способности TCP

Любое реалистичное выражение пропускной способности TCP, как функции RTT и вероятности потери пакетов, следует рассматривать, как подходящее для использования с TFRC. Однако следует отметить, что используемое для пропускной способности TCP уравнение должно отражать поведение повторов передачи по тайм-ауту, поскольку это поведение доминирует при определении пропускной способности TCP в условиях высокой вероятности потерь. Отметим также, что допущения, принятые относительно вероятности потерь в уравнении для пропускной способности, зависят от реального механизма определения вероятности потерь. Хотя это допущение не вполне соответствует приведённому ниже уравнению для пропускной способности и описанным механизмам измерения, оно достаточно хорошо подходит на практике.

Уравнение пропускной способности, которое в настоящее время **требуется** использовать в TFRC, является слегка упрощённым вариантом уравнения для Reno TCP из работы [PFTK98]. В идеальном случае уравнение для пропускной способности следовало бы создавать на основе SACK[3] TCP, однако тесты и эксперименты показывают, что различия между двумя уравнениями достаточно малы [FF99] (Приложение B).

Уравнение для средней скорости передачи TCP (в байтах в секунду) X_Bps имеет вид:

```
                                s
X_Bps = --------------------------------------------------------
        R*sqrt(2*b*p/3) + (t_RTO * (3*sqrt(3*b*p/8)*p*(1+32*p^2)))
```

где:

X_Bps - средняя скорость передачи TCP (байт/сек); параметр X_Bps идентичен X_calc в RFC 3448;

s - размер сегмента (байт) с учётом заголовков транспортного уровня и IP;

R - время кругового обхода в секундах;

p - вероятность потерь (0 - 1.0) - доля потерянных пакетов в от общего числа переданных пакетов;

t_RTO - тайм-аут повторной передачи TCP в секундах;

b - максимальное число пакетов, подтверждаемых одним пакетом TCP ACK.

Выбор значения тайм-аута повторной передачи TCP - t_RTO.

Реализациям **следует** использовать значение t_RTO = 4*R. **Возможно** использование более точного метода расчёта t_RTO. Реализации **могут** также выбирать в качестве t_RTO значение max(4*R, 1 секунда) в соответствии с рекомендуемым для значения RTO минимумом [RFC2988].

Выбор параметра b для отложенных подтверждений.

Некоторые современные реализации TCP используют отложенные подтверждения, передавая один пакет подтверждения для каждой пары принятых пакетов данных. Однако TCP также позволяет передавать подтверждения для каждого принятого пакета данных. Для пересмотренных механизмов контроля насыщения TCP [RFC2581bis] в настоящее время задаёт алгоритм отложенных подтверждений, который следует использовать. Однако [RFC2581bis] рекомендует увеличивать размер окна насыщения при предотвращении перегрузки по одному сегменту за период RTT даже при использовании отложенных подтверждений, в соответствии с уравнением пропускной способности TCP для случая b = 1. На основе экспериментальных данных [RFC2581bis] позволяет увеличивать окно насыщения в процессе замедленного старта, что также соответствует уравнению пропускной способности TCP для случая b = 1. Таким образом, использование b = 1 согласуется с [RFC2581bis]. **Рекомендуется** устанавливать значение b = 1.

При t_RTO=4*R и b=1 уравнение для пропускной способности X_Bps (скорость передачи TCP в байтах за секунду) может быть упрощено до:

```
                          s
X_Bps = ------------------------------------------
        R * (sqrt(2*p/3) + 12*sqrt(3*p/8)*p*(1+32*p^2))
```

В будущем обновлённые варианты этого документа могут использовать иные уравнения для TCP взамен приведённого здесь. Требование к таким уравнениям заключается в том, что они должны давать разумное приближение скорости передачи TCP для соответствия с механизмом контроля насыщения TCP.

Пропускную способность можно также выразить в X_pps (пакет/сек)

[1]Explicit Congestion Notification.
[2]Round-trip time.
[3]Selective acknowledgment - селективное подтверждение.

```
X_pps = X_Bps/s
```

Параметры s (размер сегмента), p (вероятность потерь) и R (RTT) должны измеряться или рассчитываться реализацией TFRC. Измерение s описано в параграфе 4.1, измерение R - в параграфе 4.3, а измерение p - в разделе 5. В оставшейся части документа скорости передачи данных измеряются байтами в секунду, если явно не указано иное.

3.2. Содержимое пакетов

Прежде, чем описывать функциональность отправителя и получателя, рассмотрим содержимое пакетов данных, передаваемых отправителем, и пакетов обратной связи, передаваемых получателем. Мы не будем задавать формат пакетов, поскольку TFRC будет использоваться с протоколом транспортного уровня, который и определяет этот формат.

3.2.1. Пакеты данных

Каждый пакет данных, передаваемый отправителем, содержит следующую информацию:

- Порядковый номер. Этот номер **должен** увеличиваться на 1 с каждым переданным пакетом. Поле должно быть достаточно большим, чтобы в списке недавних пакетов получателя не появлялись разные пакеты с одинаковым порядковым номером.

- Временная метка момента передачи. Будем обозначать ts_i временную метку пакета с порядковым номером i. Разрешение для временных меток в общем случае **следует** измерять в миллисекундах.

  Эти временные метки используются получателем для определения потерь пакетов, которые следует отнести к одному событию (факту потери). Метки получатель возвращает отправителю в качестве «эхо» для того, чтобы тот мог оценить время кругового обхода (это нужно для отправителей, не сохраняющих временных меток переданных пакетов).

  Отметим, что существует альтернативный вариант использования временных меток, когда значение метки инкрементируется каждую четверть периода кругового обхода; такие метки **могут** служить для отнесения потерь пакетов к одному факту потери в контексте протокола, где это понятно как отправителю, так и получателю, а отправитель сохраняет временные метки переданных пакетов.

- Оценка времени кругового обхода отправителем. Оценка, передаваемая в пакете i обозначается R_i. Оценка времени кругового обхода используется получателем вместе с временными метками для определения множества потерь пакетов, относящихся к одному событию.

  Если отправитель передаёт передаёт грубые «временные метки», которые увеличиваются каждую четверть периода кругового обхода, как описано выше, такому отправителю не нужно передавать свою оценку времени кругового обхода.

3.2.2. Пакеты обратной связи

Каждый пакет обратной связи, передаваемый получателем данных, содержит следующую информацию:

- Временная метка последнего принятого пакета (t_recvdata). Если последний принятый пакет имеет номер i, то t_recvdata = ts_i. Эта временная метка используется отправителем для оценки времени кругового обхода и требуется только в тех случаях, когда отправитель не сохраняет временные метки переданных пакетов данных.

- Интервал времени между приёмом последнего пакета и генерацией данного пакета обратной связи. Будем обозначать этот интервал t_delay.

- Оценка получателем скорости приёма. данных в предыдущем периоде кругового обхода. Будем обозначать этот интервал X_recv.

- Текущее значение вероятности потери по оценке получателя (p).

4. Протокол отправителя данных

Отправитель шлёт получателю поток пакетов данных с определённой скоростью. При получении пакета обратной связи от получателя, отправитель данных меняет скорость передачи на основе информации, содержащейся в таком пакете. Если отправитель не получает пакетов обратной связи в течение четырёх интервалов кругового обхода, он снижает скорость передачи вдвое. Для контроля времени используется таймер обратной связи[1].

Для протокола на передающей стороне зададим следующие этапы:

- измерение среднего размера передаваемого сегмента;
- инициализация отправителя;
- реакция отправителя на получение пакета обратной связи;
- поведение отправителя по завершении отсчёта таймера обратной связи;
- предотвращение осцилляций (опционально);
- планирование передачи пакетов и допустимых пиков.

4.1. Измерение размера сегмента

Отправитель TFRC использует размер сегмента s в уравнении пропускной способности, при установке максимальной скорости приёма., минимальной и начальной скорости передачи, а также значения таймера обратной связи. Получатель TFRC **может** использовать средний размер сегмента s при инициализации истории потерь. Как указано в

[1]В оригинале используется термин nofeedback timer. *Прим. перев.*

параграфе 6.3.1, если получатель TFRC не знает размера сегмента, используемого отправителем, он **может** использовать взамен при инициализации истории потерь среднее число принимаемых в секунду пакетов.

Размер сегмента s обычно известен приложению, но возможны два исключения:

1) Размер сегментов меняется естественным образом в зависимости от данных. В этом случае, хотя размер сегментов меняется, его вариации не отражаются на скорости передачи. Отправитель TFRC может рассчитать размер сегмента или использовать максимальное значение размера сегмента s.

2) Для контроля насыщения приложение может менять размер сегментов, а не скорость их передачи. Это обычная практика для аудио-приложений, в которых пакеты данных передаются с фиксированным интервалом, требуемым для представления каждого пакета. Для таких приложений требуется совершенно иной способ измерения параметров.

Для первого класса приложений, где размер сегментов меняется в зависимости от данных, отправителю **следует** оценить размер сегмента s, как среднее значение в течение четырёх последних интервалов между фактами потери. При желании отправитель **может** оценивать среднее значение размера за более продолжительный период.

Второй класс приложений рассматривается отдельно в документе, посвящённом TFRC-PS [RFC4828]. В оставшейся части этого раздела предполагается, что отправитель может оценить размер сегмента и контроль насыщения осуществляется за счёт управления числом пакетов, передаваемых за секунду.

4.2. Инициализация отправителя

Начальные значения X (допустимая скорость передачи, в байт/сек) и tld[1] (время последнего удвоения в процессе замедленного старта, в секундах) являются неопределёнными, пока не будут установлены, как описано ниже. Если отправитель готов передавать данные, когда у него ещё нет результатов измерения периода кругового обхода, в качестве X используется скорость s байт/с для сегмента размером s, для таймера обратной связи устанавливается значение 2 секунды, а для tld - 0 или -1 (если это подходит). При получении первого результата измерения времени кругового обхода (например, после первого пакета обратной связи, обмена SYN-пакетами на этапе организации соединения или из предшествующего соединения [RFC2140]), в качестве tld используется текущее значение времени кругового обхода, для X устанавливается значение initial_rate, определённое, как W_init/R для W_init в соответствии с [RFC3390]:

```
initial_rate = W_init/R
W_init = min(4*MSS, max(2*MSS, 4380))
```

При расчёте W_init вместо максимального размера сегмента (MSS[2]) отправителю TFRC **следует** использовать максимальный размер сегмента, который используется в начальный период кругового обхода, если это значение известно отправителю TFRC в момент инициализации X.

При отклике на начальный пакет обратной связи эта процедура заменяет этап 4), описанный ниже в параграфе 4.3.

В Приложении B объясняются причины для установки начального значения таймера обратной связи TFRC 2 секунды, взамен значения 3 секунды, рекомендуемого для таймера повтора передачи TCP [RFC2988].

4.3. Поведение отправителя при получении пакета обратной связи

Отправитель знает текущее значение допустимой скорости передачи X и поддерживает оценку текущего времени кругового обхода R. Отправитель также поддерживает параметр X_recv_set, включающий несколько недавних значений X_recv (обычно два).

Инициализация. X_recv_set инициализируется одним значением Infinity[3]. **Возможна** инициализация X_recv_set взамен Infinity достаточно большим числом (например, наибольшим целым числом в системе).

При получении отправителем пакета обратной связи в момент t_now (текущее время в секундах) **должны** быть выполнены следующие операции.

1) Расчёт нового значения периода кругового обхода по формуле:

```
R_sample = (t_now - t_recvdata) - t_delay
```

Как было указано в параграфе 3.2.2, значение t_delay показывает время, затраченное на приёмной стороне.

2) Обновление оценки времени кругового обхода:

```
If no feedback has been received before {
    R = R_sample;
} Else {
    R = q*R + (1-q)*R_sample;
}
```

Точное значение постоянной q не имеет существенного значения для TFRC, но по умолчанию **рекомендуется использовать значение** 0,9.

3) Обновление значения тайм-аута:

```
RTO = max(4*R, 2*s/X)
```

4) Обновление значения допустимой скорости передачи с использованием переменных t_mbi и recv_limit:

t_mbi - максимальный интервал снижения скорости 64 секунды.

recv_limit - предельное значение скорости передачи, рассчитанной из X_recv_set.

Эта процедура также использует процедуры Maximize X_recv_set() и Update X_recv_set(), определённые ниже.

Псевдокод процедуры обновления значения допустимой скорости имеет вид:

[1]Time Last Doubled.
[2]Maximum Segment Size.
[3]Бесконечность.

```
If (если весь интервал, закрываемый пакетом обратной связи, характеризуется
      ограниченной передаче данных) {
    If (пакет обратной связи говорит о новом факте потерь или росте частоты
                потерь p) {
        уменьшить вдвое элементы X_recv_set;
        X_recv = 0.85 * X_recv;
        Maximize X_recv_set();
        recv_limit = max(X_recv_set);
    } Else {
        Maximize X_recv_set();
        recv_limit = 2 * max (X_recv_set);
    }
} Else {                        // типичное поведение
    Update X_recv_set();
    recv_limit = 2 * max (X_recv_set);
}
If (p > 0) {           // фаза предотвращения перегрузки
    рассчитать X_Bps с использованием уравнения пропускной способности TCP
    X = max(min(X_Bps, recv_limit), s/t_mbi);
} Else if (t_now - tld >= R) {
    // начальная процедура замедленного старта
    X = max(min(2*X, recv_limit), initial_rate);
    tld = t_now;
}
```

5) При использовании механизма подавления осцилляций рассчитывается мгновенная скорость передачи X_inst в соответствии с параграфом 4.5.

6) Таймер обратной связи сбрасывается и устанавливается на RTO секунд.

Процедура максимизации X_recv_set сохраняет наибольший элемент X_recv_set и новое значение X_recv:

```
Maximize X_recv_set():
    добавить X_recv к X_recv_set;
    удалить начальное значение Infinity из X_recv_set, если оно ещё не удалено;
    установить текущее время в качестве временной метки наибольшего элемента;
    удалить все элементы, кроме наибольшего.
```

Процедура обновления X_recv_set сохраняет набор значений X_recv с временными метками из двух последних периодов кругового обхода.

```
Update X_recv_set():
    добавить X_recv к X_recv_set;
    Удалить из X_recv_set значения, не относящиеся к двум последним периодам RTT.
```

Определение интервала ограниченной передачи данных.

Определим для сервера интервал ограниченной передачи данных, как любой промежуток времени, в течение которого сервер не передавал того количества данных, которое позволяет допустимая скорость. Вопрос идентификации «интервалов ограниченной передачи» рассматривается в параграфе при обсуждении вопросов реализации. Термин «интервал ограниченной передачи используется в первом условии «if» этапа 4), когда решается вопрос предотвращения снижения отправителем скорости передачи в результате приёма. сообщения обратной связи о скорости приёма. в период ограниченной передачи данных.

В качестве примера рассмотрим отправителя, который передаёт данный с максимально дозволенной скоростью, но при этом пакеты передаются не поодиночке, а парами. Такой отправитель часть времени работает в режиме ограниченной передачи, поскольку пакеты не отправляются сразу по мере готовности. Однако, в соответствии с приведённым выше определением, при рассмотрении интервала, в течение которого отправитель передал пару пакетов, этот интервал нельзя отнести к интервалу ограниченной передачи данных, поскольку отправитель ограничивает передачу только в течение части интервала.

Если пакет обратной связи говорит, что скорость X_recv равна 0 (т. е., первый пакет обратной связи), отправитель не рассматривает весь интервал, покрываемый этим пакетом, как интервал с ограниченной передачей данных.

X_recv_set и первый пакет обратной связи.

Поскольку X_recv_set инициализируется с одним элементом Infinity, для recv_limit устанавливается значение Infinity на два первых интервала кругового обхода в соединении. В результате скорость передачи не ограничивается скоростью приёма. в течение данного периода. Это позволяет избежать проблем, связанных с ограничением скорости передачи данных по значению X_recv из первого пакета обратной связи.

Интервал, покрываемый пакетом обратной связи.

Как отправитель может определить период, к которому относится пакет обратной связи? Этот вопрос более детально рассматривается в параграфе 8.2. В общем случае, получатель будет передавать один пакет обратной связи за период кругового обхода, поэтому обычно отправитель может определить точный период, покрываемый текущим пакетом обратной связи из предыдущего пакета такого типа. Однако в случаях потери предыдущего пакета обратной связи или при более ранней передаче такого пакета в результате обнаружения пакета с маркером ECN отправителю потребуется оценка интервала, покрываемого пакетом обратной связи. Как указано в параграфе 6.2, каждый переданный получателем пакет обратной связи покрывает период кругового обхода в R_m (оценка, поддерживаемая получателем) секунд до отправки пакета обратной связи.

Отклик на потери в период ограниченной передачи.

В TFRC после начальной процедуры замедленного старта отправитель всегда обновляет расчётное значение скорости передачи X_Bps (после того, как будет получен пакет обратной связи) и допустимая скорость передачи X всегда ограничена значением X_Bps. Однако в периоды ограниченной передачи, когда реальная скорость обычно ниже X_Bps, скорость передачи остаётся ограниченной значением recv_limit, получаемым из X_recv_set. Если отправитель ограничивает передачу данных (возможно с изменением скорости от одного периода кругового обхода к другому) и

наблюдаются потери данных, мы уменьшаем значение элемента X_recv_set для снижения допустимой скорости передачи.

Отправитель может обнаружить факт потери в период ограниченной передачи по явному сигналу от получателя или по возросшему значению вероятности потерь. Когда отправитель получает пакет обратной связи, сообщающий о таком факте потерь в период ограниченной передачи, он ограничивает дозволенный рост скорости передачи в период ограниченной передачи данных.

Начальная процедура замедленного старта.

Отметим, что p=0 говорит о том, что отправитель ещё не знает о фактах потерь и находится в начальной фазе замедленного старта. В этой фазе отправитель может приблизительно удваивать скорость передачи в каждый период кругового обхода, пока не начнутся потери. Параметр initial_rate term в п. 4) определяет минимальную дозволенную скорость передачи в течение замедленного старта.

Отметим, что в тех случаях, когда отправитель ограничивает передачу во время замедленного старта или полоса соединения ограничена, отправитель может не иметь возможности удваивать скорость передачи в каждый период кругового обхода; скорость передачи ограничена большим из двух значений - удвоенная скорость приёма. в предыдущий период кругового обхода или initial_rate. Это напоминает поведение TCP, где скорость передачи ограничена скоростью входящих пакетов подтверждений, а также размером окна насыщения. Таким образом при замедленном старте TCP в более агрессивном случае, когда получатель подтверждает каждый пакет, скорость передачи TCP ограничена удвоенной скоростью приёма. пакетов подтверждения.

Минимальная дозволенная скорость передачи.

Параметр s/t_mbi при p > 0 обеспечивает отправителю возможность передачи по крайней мере одного пакета каждые 64 секунды.

4.4. Завершение отсчёта таймера обратной связи

В этом параграфе описано поведение отправителя при завершении отсчёта таймера обратной связи. Отсчёт этого таймера может завершиться в результате продолжительного бездействия или по причине отбрасывания пакетов обратной связи в сети.

В этом параграфе используется переменная recover_rate. Если отправитель TFRC бездействует с момента запуска таймера обратной связи, допустимая скорость передачи не устанавливается ниже recover_rate. В этой спецификации для recover_rate устанавливается значение initial_rate (см. параграф 4.2). В будущих вариантах спецификации могут быть заданы другие значения recover_rate.

При завершении отсчёта таймера обратной связи отправитель **должен**:

1) Снизить дозволенную скорость передачи вдвое.

   Если в момент завершения отсчёта таймера у отправителя имелось хотя бы одно измерение RTT, допустимая скорость передачи снижается путём изменения X_recv_set в соответствии с приведённым ниже псевдокодом (включая п. 2)). В общем случае скорость передачи ограничена значением не более двух X_recv. Изменение X_recv_set ограничивает скорость передачи, но позволяет отправителю выполнять процедуру замедленного старта с удвоением скорости передачи в каждый период RTT, если пакеты обратной связи не говорят о потерях.

   Если отправитель бездействовал с момента запуска таймера обратной связи и X_recv < recover_rate, дозволенная скорость передачи не снижается вдвое и X_recv_set не меняется. Это предотвращает снижение дозволенной скорости до значений меньше половины recover_rate в результате бездействия.

   В общем случае допустимая скорость передачи снижается вдвое в ответ на завершение отсчёта таймера обратной связи. Детали приведённого ниже псевдокода зависят от состояния отправителя - процесс замедленного старта, предотвращение перегрузки с ограничением по X_recv или предотвращение перегрузки с ограничением на основе уравнения пропускной способности.

```
X_recv = max (X_recv_set);
If (у отправителя нет значения RTT, не было получено пакетов
    обратной связи и не наблюдалось бездействия с момента
    запуска таймера обратной связи) {
    // У отправителя пока нет значений X_Bps и recover_rate.
    // Дозволенная скорость снижается вдвое.
    X = max(X/2, s/t_mbi);
} Else if (((p>0 && X_recv < recover_rate) or (p==0 && X < 2 * recover_rate))
       и отправитель бездействовал с момента запуска таймера обратной связи) {
    // Не снижать вдвое дозволенную скорость передачи.
    Ничего не делать
} Else if (p==0) {
    // Ещё нет значения X_Bps.
    // Снизить вдвое дозволенную скорость передачи.
    X = max(X/2, s/t_mbi);
} Else if (X_Bps > 2*X_recv)) {
    // Значение 2*X_recv уже ограничивает скорость передачи.
    // Снизить вдвое дозволенную скорость передачи.
    Update_Limits(X_recv;)
} Else {
    // Скорость передачи ограничена X_Bps, а не X_recv.
    // Снизить вдвое дозволенную скорость передачи.
    Update_Limits(X_Bps/2);
}
```

   Значение s/t_mbi ограничивает снижение скорости (по крайней мере 1 пакет за 64 секунды).

Процедура Update_Limits() использует переменную timer_limit для ограничения скорости передачи, рассчитанной по завершению отсчёта таймера обратной связи, как показано ниже:

```
Update_Limits(timer_limit):
    If (timer_limit < s/t_mbi)
        timer_limit = s/t_mbi;
    Заменить содержимое X_recv_set одним элементом timer_limit/2;
    Пересчитать X в соответствии с п. 4) параграфа 4.3;
```

2) Запустить таймер обратной связи на max(4*R, 2*s/X) секунд.

Если отправитель ограничивал передачу, но не простаивал с момента запуска таймера обратной связи, возможно завершение отсчёта таймера в результате потери пакетов обратной связи в сети. В таких случаях таймер обратной связи является механизмом детектирования таких потерь, подобно таймеру повтора передачи в TCP.

Отметим, что при остановке передачи данных отправителем, получатель прекращает передачу пакетов обратной связи. При завершении отсчёта таймера обратной связи у отправителя последний может использовать описанную выше процедуру для ограничения скорости передачи. Если отправитель возобновляет передачу, для ограничения скорости будет использоваться X_recv_set и процедура замедленного старта, пока скорость передачи не достигнет значения X_Bps.

Снижение отправителем TFRC скорости передачи по таймеру обратной связи похоже на уменьшение размера окна насыщения TCP по истечении каждых RTO периода бездействия для TCP с поддержкой Congestion Window Validation [RFC2861].

4.5. Предотвращение осцилляций

Для снижения осцилляций скорости и задержек в очередях в средах с низким уровнем статистического мультиплексирования при высокой нагрузке отправителю **рекомендуется** снизить скорость передачи данных при увеличении задержки в очереди (и, следовательно, RTT). Для реализации этого отправитель поддерживает оценку среднего значения RTT за достаточно большой период (R_sqmean) и меняет свою скорость передачи в зависимости от того, как квадратный корень из R_sample (недавнее значение RTT) отличается от среднего (R_sqmean). Среднее значение R_sqmean определяется следующим образом:

```
Если ранее не было получено пакетов обратной связи
    R_sqmean = sqrt(R_sample);
иначе
    R_sqmean = q2*R_sqmean + (1-q2)*sqrt(R_sample);
```

Таким образом, R_sqmean изменяется пропорционально квадратному корню измеренного значения RTT. Константу q2 следует устанавливать по аналогии с q; по умолчанию **рекомендуется** использовать значение q2=0,9.

При sqrt(R_sample) > R_sqmean текущее время кругового обхода превышает среднее значение, что говорит о возможном увеличении задержки в очереди. В таких случаях скорость передачи снижается для минимизации осцилляций задержки в очередях.

Отправитель получает базовое значение дозволенной скорости передачи X, как описано в п. 4) параграфа 4.3. После этого рассчитывается новое мгновенное значение скорости передачи:

```
X_inst = X * R_sqmean/sqrt(R_sample);
If (X_inst < s/t_mbi)
    X_inst = s/t_mbi;
```

Благодаря использованию квадратного корня в общем случае мгновенная скорость передачи X_inst незначительно отличается от дозволенной скорости X. Например, в некой экстремальной ситуации, когда текущее значение RTT (R_sample) вдвое превышает среднее значение, sqrt(R_sample) будет отличаться от R_sqmean приблизительно в 1, 44 раза и дозволенная скорость будет снижаться путём умножения на коэффициент приблизительно равный 0,7.

Отметим, что такое изменение поведения для предотвращения осцилляций требуется не во всех случаях, особенно в тех случаях, когда уровень статистического мультиплексирования в сети достаточно велик. Отметим также, что подавление осцилляций может вызывать проблемы для соединений, на которых время кругового обхода слабо связано с задержкой в очередях (например, в беспроводных средах при частых изменениях маршрутизации). Однако такую возможность **следует** реализовать поскольку она улучшает поведение TFRC в некоторых средах с низким уровнем статистического мультиплексирования. Работа этого механизма проиллюстрирована в параграфе 3.1.3 документа [FHPW00]. Если подавление осцилляций не поддерживается, реализации **следует** использовать очень малое значение весового коэффициента q при определении среднего времени кругового обхода.

4.6. Планирование передачи пакетов

Поскольку TFRC работает на основе скорости, а операционные системы обычно не способны точно планировать события, необходимо с осторожностью относиться к передаче данных, чтобы поддерживалось корректное среднее значение скорости, несмотря на грубое или нерегулярное планирование в операционной системе. Для поддержки корректного среднего значения скорости передачи отправитель TFRC **может** передавать некоторые пакеты раньше номинального времени.

В дополнение к этому планирование передачи пакетов управляет допустимыми пиками передачи после периодов бездействия и ограниченной передачи. Отправитель TFRC **может** накапливать «кредиты на передачу» за прошлые периоды бездействия; это позволяет отправителю TFRC передавать данные с пиковой скоростью после периодов бездействия или ограничения передачи. TCP, для сравнения, может передать в одном пике число пакетов, передаваемое за период кругового обхода, но не более того. Например, пиковое число пакетов может быть передано TCP при получении пакетов ACK, подтверждающих окно данных, или когда отправитель внезапно получает окно данных для передачи после задержки приблизительно на время кругового обхода.

Для ограничения пиков трафика реализация TFRC **должна** предотвращать выборы трафика произвольной величины. Предел допустимых пиков **должен** быть не больше числа пакетов, передаваемых за один период кругового обхода. Реализация TFRC **может** ограничивать пики значением меньше числа пакетов, передаваемых за период кругового

обхода. Кроме того, реализация TFRC **может** использовать ограничение пиков по скорости в целях выравнивания потока трафика.

Например, реализация протокола может рассчитывать интервал между пакетами (t_ipi) следующим образом:

```
t_ipi = s/X_inst;
```

Пусть t_now показывает текущее время, а i является натуральным числом (i = 0, 1, ...) и t_i показывает номинальное время передачи i-го пакета. Тогда номинальное время t_(i+1) можно рекурсивно определить следующим образом:

```
t_0 = t_now,
t_(i+1) = t_i + t_ipi.
```

Отправителя TFRC разрешается накапливать «кредиты на передачу» за неиспользованное для передачи время в течение последних T секунд отправителю разрешено передавать за неиспользованное номинальное время t_j в пока t_j < t_now - T для T, равного времени кругового обхода.

5. Расчёт частоты потерь (p)

Точное и стабильное измерение частоты потерь имеет первоочередное значение для TFRC. Измерение частоты потери пакетов осуществляется на приёмной стороне путём детектирования потери пакетов по порядковым номерам прибывающих пакетов или при получении маркированного пакета. Опишем этот процесс прежде, чем разбираться с остальной частью приёмного протокола. Если получатель ещё не обнаружил потери или маркировки пакетов, он не рассчитывает вероятность потерь и сообщает о нулевой частоте потерь.

5.1. Детектирование потерь и маркированных пакетов

TFRC предполагает, что в каждом пакете содержится порядковый номер и эти номера увеличиваются на 1 в каждом переданном пакете. Данная спецификация требует, чтобы при повторе передачи потерянного пакета использовался новый порядковый номер в соответствии с последовательностью нумерации. Если транспортный протокол требует при повторе передачи использовать исходный порядковый номер, разработчики транспортного протокола должны обеспечить механизм, позволяющий отличить задержанные пакеты от переданных повторно, и механизм детектирования потери пакетов при повторе передачи.

TFRC предполагает, что каждый пакет содержит порядковый номер и номера увеличиваются на единицу в каждом переданном пакете. Данная спецификация **требует**, чтобы при повторной передаче потерянных пакетов использовались новые порядковые номера, а не номера, с которыми потерянные пакеты были переданы изначально. Если используемый транспортный протокол требует повтора передачи с исходными порядковыми номерами, разработчики транспортного протокола должны обеспечить механизм, позволяющий отличить исходный пакет от переданного повторно.

Получатель поддерживает структуру данных, в которой хранится информация о полученных и пропущенных пакетах. В данной спецификации предполагается, что эта структура представляет собой список полученных пакетов и временных меток момента приёма. каждого такого пакета. На практике такая структура может использовать более компактное представление, выбранное разработчиками.

Получатель поддерживает структуру данных, в которой хранится информация о полученных и пропущенных пакетах. В данной спецификации предполагается, что эта структура представляет собой список полученных пакетов и временных меток момента приёма. каждого такого пакета. На практике такая структура может использовать более компактное представление, выбранное разработчиками.

Потеря пакета детектируется по факту прибытия по крайней мере NDUPACK пакетов с большими порядковыми номерами; для NDUPACK задано значение 3. Требование приёма. NDUPACK последующих пакетов совпадает с аналогичным требованием TCP и обеспечивает TFRC устойчивость к нарушению порядка доставки пакетов. В отличие от TCP, если пакет приходит позднее (после доставки NDUPACK следовавших за ним пакетов), информация об этом пакете может заполнить пробел в записях TFRC и получатель сможет пересчитать вероятность потерь. В будущих версиях TFRC значение NDUPACK для детектирования потери может быть заменено адаптивным значением, учитывающим реальное нарушение порядка доставки, но в данной спецификации механизм такого учёта не рассматривается.

Для соединений, поддерживающих ECN, прибытие маркированных пакетов трактуется как факт насыщения без ожидания доставки последующих пакетов.

Если перед пакетами с маркерами ECN была возможна потеря пакетов[1], детектированием факта насыщения считается потеря первого пакета. Например, если получатель принимает пакет с порядковым номером n-1, за которым следует немаркированный пакет с порядковым номером n+1, а затем маркированный пакет с порядковым номером n+2, тогда получатель обнаруживает насыщение по приёму пакета n+2. Начало обнаруженного факта перегрузки связано с потерей пакета n. Рекомендации параграфа 5.2 позволяют определить принадлежность потерь и маркированных пакетов к одному или нескольким фактам потерь.

5.2. Преобразование истории потерь в факты потерь

TFRC требует устойчивости к нескольким последовательным потерям пакетов или приёму маркированных пакетов, когда эти события относятся к одному факту потери. Это похоже на поведение протокола TCP, который (обычно) лишь однократно за период RTT уменьшает наполовину окно насыщения. Таким образом, получатель должен отобразить историю потери пакетов в запись о факте потери, трактуя как факт потери утрату одного или более пакетов или приём маркированных одного или более пакетов в течение периода RTT. Для выполнения такого отображения получателю нужно знать значение RTT, которое обычно периодически сообщается отправителем в форме управляющей информации, присоединяемой в конце пакета данных. Для TFRC способ передачи результатов измерения RTT получателю не имеет значения, однако рекомендуется использовать для этого рассчитанное отправителем значение RTT (R в параграфе 4.3).

[1] Имеются пропуски в порядковых номерах. *Прим. перев.*

Для того, чтобы определить, относится потеря или маркированный пакет к новому факту потери или является продолжением существующего, нужно сравнить порядковые номера и временные метки пакетов, принятых получателем. Для маркированного пакета S_new время приёма. T_new можно определить напрямую. Для потерянного пакета нужно использовать интерполяцию, чтобы определить номинальное «время прибытия». Предположим, что:

S_loss - порядковый номер потерянного пакета;

S_before - порядковый номер последнего пакета, прибывшего с номером меньше S_loss прежде любого пакета с номером, превышающим S_loss;

S_after - порядковый номер первого пакета, прибывшего с после пакета S_before, с порядковым номером больше S_loss;

S_max - наибольший порядковый номер.

Следовательно, S_before < S_loss < S_after <= S_max.

T_loss - номинальное время, когда должен был прибыть потерянный пакет;

T_before - время приёма. пакета S_before;

T_after - время прибытия пакета S_after.

Отметим, что T_before < T_after.

Для потерянного пакета S_loss можно интерполировать номинальное «время прибытия» на основе времени прибытия пакетов S_before и S_after:

```
T_loss = T_before + ((T_after - T_before) * (S_loss - S_before)/(S_after - S_before))
```

Для случая перехода порядковых номеров через максимальное значение (0), предположим, что S_MAX = 2^b, где b - размер принятого в реализации поля порядковых номеров в битах. В этом случае мы можем интерполировать время прибытия T_loss следующим образом:

```
T_loss = T_before + (T_after - T_before) * Dist(S_loss, S_before)/Dist(S_after, S_before)
```

где

```
Dist(S_A, S_B) = (S_A + S_MAX - S_B) % S_MAX
```

Если было определено, что потеря пакета S_old явилась началом предыдущего факта потери и мы детектировали потерю пакета S_new, можно интерполировать номинальное время прибытия пакетов S_old и S_new (T_old и T_new, соответственно).

Если T_old + R >= T_new, потеря пакета S_new относится к текущему факту потерь. В противном случае S_new относится к новому факту потери.

5.3. Размер интервала без потерь

После детектирования первого факта потери, получатель делит пространство порядковых номеров на интервалы без потерь. Если интервал без потерь A определён, как начинающийся с пакета S_A, а следующий интервал B начинается с пакета S_B, тогда к интервалу A относятся (S_B - S_A) пакетов. Таким образом, интервал A включает все пакеты, переданные отправителем с момента отправки первого пакета интервала A до первого пакета интервала B (не включая этот пакет).

Текущий интервал I_0 определяется как интервал без потерь, включающий последний факт потерь. Если интервал без потерь начинается с пакета S_A, а S_C является максимальным номером пакета в этом интервале, размер I_0 составляет S_C - S_A + 1. Например, если текущий интервал состоит из одного пакета с маркером ECN, то S_A == S_C и размер интервала без потерь равен одному пакету.

5.4. Средний интервал без потерь

Для расчёта частоты потерь p сначала вычислим продолжительность среднего интервала без потерь. Это осуществляется с помощью взвешенного фильтра, определяющего средний интервал на основе значений n последних интервалов так, чтобы значение частоты потерь изменялось достаточно непрерывно. Если получатель ещё не имеет данных о потерях или маркированных пакетах, средний интервал без потерь не рассчитывается.

Веса w_0 - w_(n-1) определяются следующим образом:

```
If (i < n/2) {
    w_i = 1;
} Else {
    w_i = 2 * (n-i)/(n+2);
}
```

Таким образом, для n=8 значения w_0 - w_7 составляют:

```
1.0, 1.0, 1.0, 1.0, 0.8, 0.6, 0.4, 0.2
```

Значение n для числа интервалов без потерь, используемых при расчёте частоты потерь, определяет скорость отклика TFRC на изменение уровня насыщения. **Рекомендуется** использовать в качестве значения этого параметра 8. TFRC **не следует** использовать значения n > 8 для трафика, который может конкурировать в глобальной сети Internet с трафиком TCP. В крайнем случае при использовании значений n > 8 для обеспечения безопасной работы потребуется незначительное изменение механизмов TFRC для обеспечения более резкого отклика на два и более периода RTT с высоким уровнем потерь.

При расчёте среднего интервала без потерь требуется решить вопрос о включении последнего интервала. Предлагается включать его только в тех случаях, когда он существенно увеличивает значение среднего интервала.

Пусть I_0 - I_n обозначают интервалы без потерь и I_0 относится к текущему факту потерь. Если имеется по крайней мере n интервалов без потерь, установим k = n, в остальных случаях в качестве значения k будем использовать

максимальный номер имеющегося интервала без потерь. Тогда средний интервал рассчитывается следующим образом:

```
I_tot0 = 0;
I_tot1 = 0;
W_tot = 0;
for (i = 0 to k-1) {
    I_tot0 = I_tot0 + (I_i * w_i);
    W_tot = W_tot + w_i;
}
for (i = 1 to k) {
    I_tot1 = I_tot1 + (I_i * w_(i-1));
}
I_tot = max(I_tot0, I_tot1);
I_mean = I_tot/W_tot;
```

Частота (вероятность) потерь p составит:

```
p = 1/I_mean;
```

5.5. Дисконтирование истории

Как было показано в параграфе 5.4, при расчёте среднего значения по n интервалам без потерь последний интервал даёт 1/(0.75*n) часть общего веса, независимо от продолжительности этого интервала. В этом параграфе описан **дополнительный** механизм «дисконтирования истории»[1], рассмотренный в работах [FHPW00a] и [W00], который позволяет приёмному узлу TFRC подбирать весовые параметры, придавая больший вес последнему интервалу без потерь, когда этот интервал более чем вдвое превышает рассчитанное значение среднего интервала.

Для дисконтирования истории свяжем коэффициент DF_i (число с плавающей запятой) с каждым интервалом L_i (для i > 0). Общая история дисконтирования для каждого интервала без потерь будет храниться в массиве коэффициентов. В начальный момент значения элементов массива DF_i устанавливаются в 1:

```
for (i = 0 to n) {
    DF_i = 1;
}
```

Дисконтирование истории также использует общий коэффициент DF (число с плавающей запятой), который также имеет начальное значение 1. Сначала посмотрим, как коэффициенты используются при расчёте среднего интервала без потерь, а затем опишем изменение коэффициентов с течением времени.

Как описано в параграфе 5.4, средний интервал без потерь вычисляется с использованием n значений предыдущих интервалов I_1, ..., I_n и значения I_0 для текущего интервала без потерь. Расчёт среднего интервала с использованием коэффициентов дисконтирования незначительно отличается от процедуры, описанной в параграфе 5.4:

```
I_tot0 = I_0 * w_0;
I_tot1 = 0;
W_tot0 = w_0;
W_tot1 = 0;
for (i = 1 to n-1) {
    I_tot0 = I_tot0 + (I_i * w_i * DF_i * DF);
    W_tot0 = W_tot0 + w_i * DF_i * DF;
}
for (i = 1 to n) {
    I_tot1 = I_tot1 + (I_i * w_(i-1) * DF_i);
    W_tot1 = W_tot1 + w_(i-1) * DF_i;
}
p = min(W_tot0/I_tot0, W_tot1/I_tot1);
```

Значение общего коэффициента DF обновляется по прибытии каждого пакета в соответствии с приведённым ниже описанием. Сначала получатель определяет средневзвешенное значение I_mean для интервалов без потерь I_1, ..., I_n:

```
I_tot = 0;
W_tot = 0;
for (i = 1 to n) {
    W_tot = W_tot + w_(i-1) * DF_i;
    I_tot = I_tot + (I_i * w_(i-1) * DF_i);
}
I_mean = I_tot / W_tot;
```

Значение I_mean сравнивается с размером текущего интервала без потерь I_0. Если I_0 превышает I_mean более, чем вдвое, это говорит о том, что новый интервал без потерь существенно превышает старые значения и значение общего коэффициента DF изменяется для снижения относительного веса более старых интервалов:

```
if (I_0 > 2 * I_mean) {
    DF = 2 * I_mean/I_0;
    if (DF < THRESHOLD) {
        DF = THRESHOLD;
    }
} else {
    DF = 1;
}
```

Отличное от 0 значение порога THRESHOLD обеспечивает гарантию того, что информация о более ранних интервалах в периоды высокого насыщения не будет полностью обесценена. Рекомендуется устанавливать THRESHOLD = 0,25. Отметим, что прибытие каждого нового пакета ведёт к дополнительному росту I_0 и коэффициент DF будет обновляться.

[1] History discounting mechanism.

При новом факте потерь текущий интервал переходит из I_0 в I_1, интервал I_i - в I_(i+1), а интервал I_n отбрасывается. Предыдущий коэффициент DF включается в массив коэффициентов дисконтирования. Поскольку DF_i показывает коэффициент, связанный с интервалом I_i, значения DF_i в массиве также смещаются при новом факте потерь. Процедура сдвига имеет вид:

```
for (i = 1 to n) {
    DF_i = DF * DF_i;
}
for (i = n-1 to 0 step -1) {
    DF_(i+1) = DF_i;
}
I_0 = 1;
DF_0 = 1;
DF = 1;
```

На этом описание **дополнительного** механизма дисконтирования истории заканчивается. Подчеркнём что этот механизм является опциональным и позволяет TFRC более быстро реагировать на стремительное прекращение перегрузок, демонстрируемое ростом интервала без потерь.

6. Протокол получателя данных

Получатель периодически направляет отправителю сообщения обратной связи. Пакеты обратной связи в обычных условиях **следует** передавать по крайней мере по одному за период RTT, если отправитель не передаёт менее 1 сообщения за период RTT (в последнем случае пакеты обратной связи **следует** передавать в ответ на каждый принятый пакет). Пакеты обратной связи **следует** также передавать при новых фактах потерь без ожидания завершения цикла RTT и при получении пакетов с нарушением порядка доставки, когда это ведёт к удалению факта потерь из истории.

Если отправитель передаёт пакеты с высокой скоростью (много пакетов за период RTT), передача пакетов обратной связи несколько раз в течение RTT может обеспечивать преимущества, поскольку позволит быстрее реагировать на изменение результатов измерения RTT и повысит устойчивость к потере пакетов обратной связи. Однако эти преимущества с ростом числа пакетов обратной связи за период RTT растут достаточно медленно.

Если получатель передал k пакетов обратной связи за период RTT, для k>1 п. 4) параграфа 6.2 изменяется с установкой для таймера обратной связи значения R_m/k секунд. Однако каждый пакет обратной связи уведомляет пакета о скорости приёма. а течение последнего периода RTT, а не части этого периода. В этом документе не задаются изменения, которые могут потребоваться для передачи получателем множества пакетов обратной связи за период RTT. Отметим, что передача множества пакетов обратной связи за время RTT даёт незначительные преимущества.

6.1. Поведение получателя при поступлении пакета данных

При получении пакета данных приёмный узел выполняет ряд действий:

1) Добавление пакета в историю принятых пакетов.

2) Если новый пакет ведёт к обнаружению нового факта потерь или не было передано пакета обратной связи на момент завершения отсчёта таймера обратной связи, выполняется п. 3. В остальных случаях дополнительных действий не предпринимается (за исключением оптимизации, описанной в следующем параграфе).

   Для **дополнительной** оптимизации может выполняться проверка заполнения полученным пакетом пропуска в порядковых номерах (истории) и связанное с этим объединение интервалов без потерь. Если пакет заполняет пропуск, получатель может незамедлительно передать пакет обратной связи. Эффект от такой оптимизации в нормальных условиях предполагается незначительным.

3) Расчёт p. Предыдущее значение p рассматривается, как p_prev. Рассчитывается новое значение в соответствии с разделом 5.

4) Завершение отсчёта таймера обратной связи. Если p > p_prev, отсчёт таймера считается завершённым и выполняются действия, описанные в параграфе 6.2.

   Если p <= p_prev и к моменту последнего завершения отсчёта таймера обратной связи пакет обратной связи не был передан, выполняются действия, описанные в параграфе 6.2. Если же пакет обратной связи к этому моменту был передан, никаких дополнительных действий не требуется.

6.2. Завершение отсчёта таймера обратной связи

При завершении отсчёта таймера обратной связи на принимающей стороне должны выполняться определённые действия в зависимости от наличия или отсутствия пакетов, принятых с момента отправки последнего сообщения обратной связи.

Для m-го завершения отсчёта таймера обратной связи предположим, что на приёмной стороне максимальный номер полученного пакета имеет значение S_m, а включенный в этот пакет результат измерения RTT имеет значение R_m. Как описано в параграфе 3.2.1, R_m представляет собой наиболее свежую оценку отправителем периода кругового обхода, переданную в пакетах данных. Если с момента отправки предыдущего сообщения обратной связи были получены пакеты данных, получатель выполняет следующие операции:

1) Расчёт средней частоты потерь с использованием алгоритма, описанного в разделе 5.

2) Расчёт значения скорости приёма. X_recv на основе пакетов, полученных за предыдущие R_(m-1) секунд. Это выполняется в тех случаях, когда отсчёт таймера обратной связи завершается естественным путём или констатируется в результате новой потери или приёма. маркированного пакета, как описано в п. 3) параграфа 6.1.

В типичной ситуации, когда получатель передаёт только один пакет обратной связи за период кругового обхода и отсчёт таймера обратной связи не завершается раньше времени в результате новой потери, время с момента предыдущего завершения отсчёта таймера обратной связи будет составлять R_(m-1) секунд.

Отметим, что при завершении отсчёта таймера обратной связи в результате потери или приёма. маркированного пакета, время, прошедшее с момента предыдущего завершения отсчёта, будет явно меньше R_(m-1) секунд.

Для упрощения реализации в тех случаях, когда время с момента предыдущего завершения отсчёта таймера обратной связи отлично от R_(m-1) секунд, скорость приёма. **может** рассчитываться за более продолжительный интервал вплоть до момента завершения отсчёта таймера обратной связи, которое произошло не менее R_(m-1) секунд назад.

3) Подготовка и передача сообщения обратной связи с информацией, описанной в параграфе 3.2.2.

4) Сброс и повторный запуск таймера обратной связи на время R_m секунд.

Отметим, что приведённое выше правило 2) даёт минимальное значение измеренной скорости приёма. X_recv, равное 1 пакету за период кругового обхода. Если отправитель ограничивает скорость передачи данных до значений меньше 1 пакета за RTT, это будет обусловлено потерей пакетов, а не ограничениями, вносимыми измеренным значением скорости приёма.

Если с момента передачи последнего пакета обратной связи не было принято ни одного пакета, новый пакет обратной связи не передаётся и таймер обратной связи перезапускается на время R_m секунд.

6.3. Инициализация получателя

Получатель инициализируется первым принятым пакетом данных. Предположим, что этот пакет имеет порядковый номер i.

В момент получения первого пакета:

- устанавливается p = 0;
- устанавливается X_recv = 0;
- подготавливается и передаётся пакет обратной связи;
- таймер обратной связи запускается на время R_i секунд.

Если первый пакет данных не содержит оценки периода кругового обхода R_i, получатель передаёт пакет обратной связи для каждого принимаемого пакета, пока не будет получен пакет данных с оценкой времени кругового обхода.

Если отправитель использует грубые временные метки, которые инкрементируются каждую четверть периода кругового обхода, таймер обратной связи не требуется и для определения момента передачи пакета обратной связи используется приведённая ниже процедура из RFC 4342.

- Всякий раз при передаче пакета обратной связи получатель устанавливает для переменной last_counter наибольшее значение счётчика окон с момента последней передачи сообщения обратной связи, если с этого момента были получены какие-либо пакеты данных.
- Если получатель принимает пакет данных со значением счётчика окон, равным значению last_counter + 4 или превышающим его, получатель передаёт новый пакет обратной связи (отношения «больше или равно» определяются в циклическом пространстве счётчика окон).

6.3.1. Инициализация истории потерь после первого факта потери

В этом параграфе рассматриваются процедуры, которые **должны** использоваться для инициализации истории потерь после первого факта потерь.

Число пакетов, принятых до первого факта потери, не может напрямую использоваться для расчёта допустимой скорости передачи, поскольку в течение этого периода скорость передачи меняется достаточно быстро. TFRC предполагает, что корректная скорость передачи данных после первой потери равна половине скорости передачи перед потерей. TFRC аппроксимирует эту скорость X_target по максимальному значению X_recv (в течение процедуры замедленного старта для отдельного периода кругового обхода скорость передачи данных отправителем в общем случае равна удвоенной скорости приёма. данных получателем за предыдущий период кругового обхода).

После первой потери вместо инициализации первого интервала без потерь числом пакетов, принятых до первой потери, получатель TFRC рассчитывает интервал без потерь, который нужен для получения скорости X_target, и использует этот искусственный интервал без потерь для запуска механизма истории потерь.

TFRC рассчитывает первый интервал без потерь, находя значение p, для которого уравнение пропускной способности в параграфе 3.1 даёт скорость передачи отличающуюся от X_target не более, чем на 5%, для данного времени кругового обхода R - для первого интервала без потерь устанавливается значение 1/p. Если получатель знает размер сегмента s, используемый отправителем, он **может** использовать уравнение пропускной способности, в противном случае получатель **может** измерить скорость приёма. в пакетах за секунду вместо байт/сек и использовать уравнение пропускной способности для X_pps (допуск 5% обусловлен тем, что уравнение пропускной способности трудно обратить и без такого допуска пришлось бы использовать ресурсоёмкие численные методы расчёта p).

Специальным случаем инициализации первого интервала без потерь является ситуация, когда первый пакет теряется или приходит с маркером насыщения. При потере первого пакета в TCP отправитель повторяет передачу этого пакета по завершении отсчёта таймера повторной передачи. Если первый пакет TCP имеет маркер ECN, отправитель сбрасывает и заново запускает таймер повтора передачи и отправляет новый пакет данных только после того, когда будет завершён отсчёт запущенного заново таймера [RFC3168] (параграф 6.1.2). Для TFRC, если первый пакет

потерян или имеет маркер ECN, первый интервал без потерь не содержит пакетов данных. В этом случае размер этого (пустого) интервала **следует** устанавливать так, чтобы скорость передачи задавалась как в TCP (см. ниже).

Когда первый интервал без потерь в TFRC пуст (первый пакет потерян или имеет маркер ECN), для обеспечения аналогии с поведением TCP, TFRC желает установить допустимую скорость передачи в 1 пакет за каждые 2 периода кругового обхода (или 0,5 на RTT). Таким образом, получатель TFRC рассчитывает интервал без потерь так, чтобы он обеспечивал значение X_target = 0,5/R пакетов в секунду для периода кругового обхода R и использовал это расчётное значение для первого интервала без потерь. Получатель TFRC использует значение 0,5/R пакетов в секунду в качестве минимального значения X_target при инициализации первого интервала без потерь.

Отметим, что несмотря на использование получателем TFRC искусственного значения интервала без потерь после первой потери, получатель продолжает сообщать о скорости передачи X_recv, как указано в параграфе 6.2.

7. Серверные варианты

В серверных вариантах TFRC получатель использует транспорт с гарантированной доставкой для передачи информации о потере пакетов отправителю, а отправитель рассчитывает частоту потерь и допустимую скорость передачи.

Основным преимуществом серверной реализации TFRC является то, что отправитель не должен доверять расчётам частоты потерь на стороне получателя. Однако требование гарантированной доставки информации о потерях от получателя к отправителю вносит существенные ограничения в процесс выбора транспортного протокола для поддержки серверных вариантов TFRC.

Вариант TFRC, реализованный на приёмной стороне в соответствии с данной спецификацией, напротив, не требует гарантированной доставки пакетов обратной связи. Этот вариант также лучше подходит для таких приложений, как потоковые службы на web-серверах, для которых желателен максимальный перенос нагрузки с серверной стороны на клиентскую.

Документы RFC 4340 и RFC 4342 совместно задают спецификацию механизма DCCP CCID 3, который может применяться в серверных вариантах TFRC. В CCID 3 каждый пакет обратной связи от получателя содержит опцию Loss Intervals, показывающую размеры последних интервалов без потерь. Пакеты обратной связи могут также включать опцию Ack Vector, позволяющую отправителю точно определить, какие пакеты были отброшены или маркированы и проверить информацию из опций Loss Intervals. Опция Ack Vector может также включать маркеры ECN Nonce Echo, позволяющие отправителю проверить данные получателя о принятых пакетах данных без маркеров. Опция Ack Vector также позволяет отправителю самому увидеть, какие пакеты данных были потеряны или маркированы ECN для детектирования интервалов без потерь и расчёта частоты потерь. Раздел 9 документа RFC 4342 включает обсуждение вопросов проверки информации, принятой от получателя.

8. Вопросы реализации

В этом документе приведена спецификация механизма контроля насыщения TFRC для использования прикладными и транспортными протоколами. В этом разделе кратко рассматриваются некоторые вопросы реализации механизма.

8.1. Расчёт пропускной способности

Для t_RTO = 4*R и b = 1 уравнение пропускной способности из параграфа 3.1 можно представить в форме:

```
                 s
X_Bps =  --------
         R * f(p)
```

где

```
f(p) =  sqrt(2*p/3) + (12*sqrt(3*p/8) * p * (1+32*p^2))
```

Значения функции f(p) могут сохраняться в специальной таблице.

Многие операции умножения (например, q и 1-q для расчёта среднего времени кругового обхода, умножение на 4 для тайм-аута) могут быть реализованы с помощью операций сдвига регистра.

8.2. Поведение отправителя при получении пакета обратной связи

В этом параграфе рассматриваются вопросы реализации, связанные поведением отправителя при получении пакетов обратной связи, как описано в параграфе 4.3.

8.2.1. Детектирование интервалов с ограниченной передачей данных

При получении пакета обратной связи отправитель проверяет, относится ли покрываемый пакетом интервал к периоду ограниченной передачи данных. В этом параграфе рассматривается один из возможных вариантов реализации этого.

Если все пакеты обратной связи содержат временную метку последнего принятого пакета, предположим, что t_new - это временная метка из данного пакета обратной связи. Поскольку все пакеты обратной связи покрывают интервал по крайней мере в один период кругового обхода, отправителю достаточно этой метки для того, чтобы определить наличие в интервале (t_old, t_new] времени с ограниченной передачей данных; если оценка времени кругового обхода отправителем составляет R, то в качестве t_old используется значение t_new - R (это оценка покрываемого пакетом обратной связи интервала, а не его точное определение, однако точность такой оценки вполне достаточна).

Ниже приведён псевдокод для проверки ограничения передачи данных в течение всего интервала, покрываемого пакетом обратной связи. Переменные NotLimited1 и NotLimited2 представляют время, когда отправитель не ограничивал передачу данных.

Инициализация

```
NotLimited1 = NotLimited2 = t_new = t_next = 0;
t_now = текущее время;
```

После передачи сегмента

```
If (отправитель передал все, что ему было дозволено) {
    // отправитель не ограничивал передачу данных в этом интервале
    If NotLimited1 <= t_new
        // цель: NotLimited1 > t_new.
        NotLimited1 = t_now
    Else if (NotLimited2 <= t_next)
        // цель: NotLimited2 > t_next.
        NotLimited2 = t_now;
}
```

При получении пакета обратной связи, если в этом интервале передача данных ограничена

```
t_new = временная метка из пакета обратной связи
t_old = t_new - R                          // локальная переменная
t_next = t_now;
If ((t_old < NotLimited1 <= t_new) or (t_old < NotLimited2 <= t_new))
    отправитель не ограничивал передачу данных в этом интервале;
Else
    отправитель ограничивал передачу данных в этом интервале.
If (NotLimited1 <= t_new && NotLimited2 > t_new)
    NotLimited1 = NotLimited2;
```

Времена передачи указывают на отправку сегмента или сегментов нижележащему уровню.

В промежутке между пакетами обратной связи (t_old, t_new] даёт интервал передачи по оценке покрываемый последним пакетом обратной связи, а t_next показывает время последнего периода кругового обхода после t_new. Предполагается, что следующий пакет обратной связи будет покрывать интервал (t_new, t_next], если получатель не передаст пакет обратной связи раньше в ответ на новый факт потери. Целью является сохранение в переменной NotLimited1 времени передачи без ограничений в интервале (t_new, t_next], если таковая происходила, а в переменной NotLimited2 - времени неограниченной передачи после t_next.

Если при получении пакета обратной связи одно из значений NotLimited1, NotLimited2 попадает в интервал, покрываемый данным пакетом, этот интервал считается интервалом передачи без ограничений (т. е., отправитель передавал данные без ограничения по крайней мере один раз в течение данного интервала). Если ни одно из значений NotLimited1, NotLimited2 не относится к интервалу, покрываемому пакетом обратной связи, предполагается, что отправитель ограничивал передачу данных в течение всего интервала.

Отметим, что эта процедура является эвристической и в некоторых случаях отправитель может некорректно определять наличие ограничений на передачу в течение интервала, покрываемого пакетом обратной связи. Эвристический подход не учитывает возможных осложнений, связанных с нарушением порядка доставки.

Рассмотренное ограничение вполне приемлемо. Для повышения точности идентификации наличия ограничений на передачу в период, покрываемый пакетом обратной связи отправитель может сохранять больше значений NotLimited.

В некоторых реализациях TFRC отправитель передаёт временные метки с грубым разрешением, которые увеличиваются каждую четверть периода кругового обхода, и пакеты обратной связи сообщают наибольший порядковый номер принятого пакета вместо временной метки последнего полученного пакета. В таких случаях отправитель может поддерживать состояние для каждого пакета, чтобы определить время t_new, когда был передан подтверждаемый пакет, или оценить t_new по оценке периода кругового обхода и прошедшему времени t_delay из пакета обратной связи.

8.2.2. Поддержка X_recv_set

Для упрощения поддержки X_recv_set достаточно ограничить размер массива X_recv_set тремя элементами (N=3). В этом случае процедура Update X_recv_set() будет иметь вид:

```
Update X_recv_set():
    Добавить X_recv к X_recv_set;
    Удалить из X_recv_set значения с возрастом более 2 периодов кругового обхода;
    Сохранить только N последних значений.
```

Поддержка лишь пары элементов в X_recv_set будет достаточна отправителю для сохранения старого значения X_recv перед периодом ограниченной передачи данных и позволяет отправителю не ограничивать себя по первому пакету обратной связи после периода бездействия, сообщающему о получении одного пакета за период кругового обхода. Однако возможны ситуации, когда поддержка только двух элементов в X_recv_set не обеспечит столь же активной работы, как в случае с поддержкой трёх элементов. Поддержка трёх элементов в X_recv_set позволяет сохранять в X_recv_set значения X_recv из двух последовательных пакетов обратной связи и значение X_recv для последнего факта потери.

8.3. Передача пакетов раньше номинального времени

В этом параграфе рассматривается один из механизмов планирования передачи пакетов на стороне отправителя для операционных систем с грубой гранулярностью отсчёта времени (параграф 4.6).

Пусть t_gran задаёт гранулярность таймера планирования в операционной системе, а t_ipi - интервал между передачей пакетов (как указано в параграфе 4.6). Если операционная система не обеспечивает нужной гранулярности таймеров или по иным причинам не может поддерживать короткие интервалы t_ipi, отправитель TFRC будет ограничен возможностью передачи не более 1 пакета за каждые t_gran секунд или ему должна быть разрешена передача множества пакетов сразу в форме коротких пиков. В дополнение к разрешению отправителю накапливать кредиты на передачу пакетов за прошлые неиспользованные периоды, полезным может оказаться разрешение отправителю передавать пакет раньше запланированного времени, как описано ниже в этом параграфе.

Параметр t_delta может использоваться для того, чтобы разрешить передачу пакетов раньше номинального времени. Рассмотрим приложение, которое переходит в режим бездействия и запрашивает следующую передачу в момент t_i = t_(i-1) + t_ipi, где t_(i-1) показывает время передачи предыдущего пакета. Когда приложение снова активизируется, оно

проверяет значение текущего времени t_now. Если t_now > t_i - t_delta, пакет передаётся. Когда рассчитывается номинальное время передачи следующего пакета t_i, может оказаться, что t_now > t_i - t_delta. В таких случаях пакет передаётся незамедлительно.

Для того, чтобы раньше номинального времени передавалось не более одного пакета и пакеты никогда не передавались ранее, чем за период кругового обхода до номинального времени передачи, параметр t_delta следует выбирать, как показано ниже:

```
t_delta = min(t_ipi, t_gran, rtt)/2;
```

(гранулярность планирования t_gran в некоторых старых системах Unix составляет 10 мсек.).

В качестве примера рассмотрим поток TFRC с допустимой скоростью передачи X = 10 пакетов за период кругового обхода (PPR), временем кругового обхода 100 мсек, гранулярностью планирования операционной системы t_gran = 10 мсек и возможностью аккумулирования неиспользованных кредитов на передачу в течение периода кругового обхода. В этом случае t_ipi будет иметь значение 1 мсек. Отправителю TFRC будет разрешено передавать пакеты за 0,5 мсек до номинального времени и сохранять неиспользованные кредиты на передачу в течение 100 мсек. Гранулярность планирования в 10 мсек не будет оказывать существенного влияния на работу соединения.

В качестве другого примера рассмотрим поток TFRC с гранулярностью планирования хуже периода кругового обхода - пусть время кругового обхода составляет 0,1 мсек, а гранулярность планирования в операционной системе - 1 мсек и обеспечивается возможность накопления кредитов на передачу в течение периода кругового обхода. Отправителю TFRC будет дозволено сохранять неиспользованные кредиты на передачу в течение 0,1 мсек. Если гранулярность планирования «не оказывает» влияния на отклики отправителя на получение пакетов обратной связи, отправитель TFRC сможет передать RTT пакетов (в соответствии с разрешённой скоростью) за каждый период RTT в ответ на принятый пакет обратной связи. В этом случае грубая гранулярность планирования не позволит существенно снизить скорость передачи, но передача может носить пиковый характер с передачей данных в течение периода кругового обхода в ответ на каждый пакет обратной связи.

Однако в этом случае производительность будет отличаться, если гранулярность планирования операционной системы будет влиять на время отклика на получение пакетов обратной связи, наряду с влиянием на обычное планирование передачи. В этом случае производительность отправителя будет существенно ограничена гранулярностью планирования, превышающей период кругового обхода, - при возможности отправителя передавать RTT данных с дозволенной скоростью передачи не более 1 пакета за каждую мсек. Такое ограничение скорости передачи является неизбежным результатом разрешения пиковой передачи не более периода кругового обхода.

8.4. Расчёт среднего интервала без потерь

Расчёт среднего интервала без потерь в параграфе 5.4 включает умножение на весовые коэффициенты w_0 - w_(n-1), которые для n=8 имеют значения:

```
1.0, 1.0, 1.0, 1.0, 0.8, 0.6, 0.4, 0.2.
```

С незначительной потерей точности можно использовать в качестве весовых коэффициентов значения степеней числа два или суммы таких значений, например:

```
1.0, 1.0, 1.0, 1.0, 0.75, 0.5, 0.25, 0.25.
```

8.5. Опциональный механизм History Discounting

Необязательный механизм дисконтирования истории, описанный в параграфе 5.5, служит для расчёта усреднённой частоты потерь. Механизм дисконтирования вводится в действие лишь в тех случаях, когда наблюдаются необычно долгие интервалы передачи без потерь. Для более эффективной работы значения коэффициента DF_i могут быть ограничены степенями числа 2.

9. Отличия от RFC 3448

9.1. Обзор изменений

В этом разделе рассматриваются отличия от RFC 3448. На верхнем уровне основным отличием является добавление механизмов обработки случаев ограниченной передачи данных отправителем. Данный документ также явно разрешает отправителю TFRC накапливать до RTT неиспользованных кредитов на передачу, обеспечивающих возможность пиковой передачи пакетов при получении от приложения данных после периода ограниченной передачи. В RFC 3448 этот вопрос в явном виде не рассматривался.

В данном документе, в отличие от RFC 3448, принимаются высокие начальные значения скорости передачи TCP из RFC 3390. данный документ также отличается от RFC 3448 в части разрешения RFC 4342 использовать грубые временные метки в пакетах данных взамен более тонкой гранулярности таких меток.

Другие отличия включают процедуру замедленного старта, отклик на отбрасывание первого пакета данных и т. п. В данном документе исправлены также отмеченные ошибки RFC 3448.

Данный раздел не является нормативным; нормативные материалы приведены выше.

9.2. Изменения в отдельных параграфах

Параграф 4.1, оценка среднего размера сегмента. Приведён вариант алгоритма, который может использоваться для оценки среднего размера сегмента.

Параграф 4.2, значение начальной скорости передачи. В RFC 3448 начальная скорость передачи имеет значение 2 пакета за время кругового обхода. В настоящем документе начальная скорость передачи может достигать 4 пакетов за период кругового обхода в соответствии с RFC 3390. Начальная скорость была изменена в терминах размера сегмента s, а не в терминах MSS.

В параграфе 4.2 данного документа сказано, что tld[1] во время замедленного старта может инициализироваться значениями 0 или -1. В параграфе 4.2 также даны разъяснения по поводу того, что измерения RTT могут осуществляться не только по пакетам обратной связи, но и иными способами (например, из обмена SYN-пакетами).

Параграф 4.3, отклик на пакеты обратной связи. Изменён способ использования скорости приёма. для ограничения дозволенной скорости передачи. Используется набор значений скорости приёма. за два последних периода кругового обхода, инициализируемый достаточно большим значением скорости приёма.

Большое начальное значение скорости приёма. (параграф 4.2) используется недолго, если получатель отправляет пакет обратной связи после получения первого пакета и отправитель в ответ на этот пакет снижает дозволенную скорость передачи до значения не более 2 пакетов за период RTT, которое является удвоенной скоростью приёма. Благодаря изменению обработки данных о скорости приёма. на стороне отправителя, последний не снижает дозволенную скорость приёма. до удвоенного значения скорости приёма. в ответ на первый пакет обратной связи.

В период ограниченной передачи данных отправитель сохраняет значение скорости приёма., предшествующее периоду ограниченной передачи, если это значение превышает скорость приёма. в период ограниченной передачи. Отправитель также уменьшает сохранённые значения X_recv_set в ответ на продолжительный период ограниченной передачи. Этот вопрос более подробно рассмотрен в Приложении C.

Параграф 4.4, отклик на период бездействия. В соответствии с параграфом 5.1 документа [RFC4342] в данном документе указано, что при снижении скорости после периода бездействия, покрывающего время с момента запуска таймера обратной связи, дозволенная скорость передачи не снижается до значений меньше начальной скорости передачи (в параграфе 4.4 для переменной recover_rate устанавливается значение начальной скорости передачи).

Параграф 4.4, исправление ошибки [RFC3448Err]. В RFC 3448 содержится противоречивый текст о снижении отправителем скорости передачи вдвое по истечении двух периодов кругового обхода без получения пакетов обратной связи или по истечении 4 периодов кругового обхода. В данном документе указано, что снижение скорости вдвое происходит после 4 периодов кругового обхода без получения пакетов обратной связи [RFC3448Err].

Параграф 4.4, разъяснение процедуры замедленного старта. В параграфе 4.4 указано, что X_Bps не может использоваться, если при завершении отсчёта таймера обратной связи p = 0, поскольку у отправителя ещё нет значения X_Bps. В параграфе 4.4 также разъяснена ситуация, когда у отправителя ещё нет результата определения RTT, но с момента запуска таймера обратной связи уже был передан пакет.

Параграф 4.6: кредиты за неиспользованную передачу. В параграфе 4.6 отмечено, что отправитель TFRC может накапливать до RTT кредитов за неиспользованные передачи. Параграф 4.6 был переписан для того, чтобы чётко разделить требования спецификации и вопросы реализации механизма TFRC.

Параграф 5.4, разъяснение. Параграф 5.4 был переписан для прояснения расчёта получателем среднего интервала между потерями в тех случаях, когда число таких интервалов ещё не достигло n.

Параграф 5.5, корректировка. Параграф 5.5 был исправлен в части того, что интервал без потерь I_0 включает все переданные пакеты с учётом потерянных или маркированных пакетов (как определено в параграфе 5.3).

Параграф 5.5, исправление ошибки [RFC3448Err]. В параграфе 5.5 строка

```
for (i = 1 to n) { DF_i = 1; }
```

была заменена на

```
for (i = 0 to n) { DF_i = 1; }
```

[RFC3448Err].

Параграф 5.5, дисконтирование истории. Значение параметра THRESHOLD, определяющее нижнюю границу параметра дисконтирования истории DF, было изменено с 0,5 до 0,25 для обеспечения возможности дисконтирования при большой продолжительности интервала без потерь.

Раздел 6, множество пакетов обратной связи. В разделе 6 расширено обсуждение процедур обработки ситуаций, когда получатель шлёт множество пакетов обратной связи за период кругового обхода.

Параграф 6.3, инициализация таймера обратной связи. В параграфе 6.3 описана инициализация таймера обратной связи на стороне получателя в тех случаях, когда в первом принятом пакете нет оценки периода кругового обхода.

Параграф 6.3, грубые временные метки. Параграф 6.3 был изменён с включением в качестве опции грубых временных меток от отправителя, инкрементируемых один раз за каждую четверть периода кругового обхода, вместо более прецизионных меток. Такое поведение соответствует RFC 4342.

Параграф 6.3.1, поведение после первой потери. В параграфе 6.3.1 настоящего документа сказано, что для инициализации истории потерь после первого факта потери получатель использует максимальную скорость приёма. взамен скорости приёма. в последнем периоде кругового обхода.

Параграф 6.3.1, отбрасывание первого пакета. В параграфе 6.3.1 описана инициализация истории потерь в тех случаях, когда первый пакет данных утерян или имеет маркировку ECN.

Раздел 7, серверные варианты. В разделе 7 рассматриваются варианты реализации механизма TFRC на серверах с учётом RFC 4342.

10. Вопросы безопасности

TFRC не является транспортным протоколом, а представляет собой механизм управления, предназначенный для использования с транспортным протоколом. Следовательно, вопросы безопасности требуется рассматривать прежде всего в контексте соответствующих транспортных протоколов и поддерживаемых ими механизмов аутентификации.

Механизмы контроля насыщения потенциально могут использоваться для организации атак на отказ служб. Такая атака может быть реализована путём передачи ложных сообщений обратной связи. Поэтому транспортным

[1]Time Last Doubled - время последнего удвоения

протоколам, использующим TFRC, следует принимать меры по защите от приёма. фальсифицированных пакетов обратной связи. Точные механизмы такой защиты зависят от выбранного транспортного протокола.

Кроме того, механизм контроля насыщения может использоваться «жадными» получателями, которые хотят получать данных больше, нежели позволяет беспристрастное деление полосы. Получатель может предпринять такую попытку за счёт передачи серверу обманной информации о приёме пакетов, которые реально были потеряны в результате насыщения. Возможной защитой от такого поведения является включение той или иной формы специальных сигналов (nonce), которые получатель должен возвращать отправителю для подтверждения приёма. Однако детали такой защиты зависят от наличия гарантий доставки пакетов на уровне транспортного протокола.

Предполагается, что протоколы, использующие ECN с TFRC, будут также поддерживать обратную связь от получателя с использованием ECN nonce [RFC3540]. ECN nonce представляет собой модификацию ECN с защитой отправителя от нечаянного или злонамеренного сокрытия маркированных пакетов. Однако детали использования таких механизмов зависят от транспортного протокола и не рассматриваются в этом документе.

10.1. Вопросы безопасности для TFRC в DCCP

TFRC в настоящее время используется механизмом контроля насыщения CCID 3[1] [RFC4342] протокола DCCP[2] [RFC4340]. Раздел «Вопросы безопасности»[3] в RFC 4340 [RFC4340] (раздел 18) включает обсуждение некоторых аспектов безопасности DCCP, в том числе проверку корректности порядковых номеров для защиты от перехвата соединений. В разделе 18 RFC 4340 также рассматриваются механизмы DCCP, способные ограничить влияние потенциальных атак на службы (DoS).

В RFC 4342 дана спецификация использования TFRC в CCID 3. RFC 4342 включает расширенное обсуждение механизмов, которые отправитель может использовать для проверки информации, переданной получателем. При использовании ECN с CCID 3 получатель возвращает отправителю информацию ECN Nonce, позволяющую последнему проверить достоверность переданных получателем данных. Для случаев когда ECN не используется в разделе 9 RFC 4342 рассматривается возможность использования отправителем различных методов, которые позволяют предотвратить проблемы, связанные с ошибочными сообщениями получателя о перегрузках в сети. Однако, как отмечено в RFC 4342, эти методы не являются столь же стойкими к ошибкам и неразрушающими, как ECN Nonce.

11. Благодарности

Авторы благодарят за отклики и обсуждении основанного на уравнении пропускной способности механизма контроля насыщения многих людей, включая членов исследовательской группы Reliable Multicast, рабочей группы Reliable Multicast Transport и исследовательской группы End-to-End. Благодарим также Dado Colussi, Gorry Fairhurst, Ladan Gharai, Wim Heirman, Eddie Kohler, Ken Lofgren, Mike Luby, Ian McDonald, Vladimir Moltchanov, Colin Perkins, Michele R., Gerrit Renker, Arjuna Sathiaseelan, Vladica Stanisic, Randall Stewart, Eduardo Urzaiz, Shushan Wen и Wendy Lee (lhh@zsu.edu.cn) за отклики на ранние версии этого документа и Mark Allman за многочисленные отклики по использованию [RFC3448] для создания работоспособной реализации.

Приложение A. Параметры

В этом документе используется целый ряд параметров. Предполагается, что временные переменные (например, t_now, tld) выражаются в секундах, а разрешение таймеров не хуже 1 миллисекунды.

data-limited interval

Интервал, в течение которого отправитель ограничивает передачу данных (т. е., передаёт их со скоростью меньше дозволенной) в течение интервала времени (параграф 4.3).

DF

Коэффициент дисконтирования для интервалов между потерями (параграф 5.5).

initial_rate

Дозволенная начальная скорость передачи.

last_counter

Наибольшее полученное значение размера окна (параграф 6.3).

n

Число интервалов без потерь.

NDUPACK

Число пакетов для принятия решения о потере (константа) (параграф 5.1).

nofeedback timer

Таймер на стороне отправителя (раздел 4).

p

Оценённая вероятность (частота) потерь.

p_prev

Предыдущее значение p (параграф 6.1).

q

[1]Congestion Control ID 3.
[2]Datagram Congestion Control Protocol - протокол дейтаграмм с контролем насыщения.
[3]Security Considerations.

Константа фильтрации для RTT (параграф 4.3).

q2

Константа фильтрации для долговременной оценки RTT (параграф 4.6).

R

Оценка времени кругового обхода для пути.

R_m

Конкретная оценка времени кругового обхода для пути (параграф 4.3, 6).

R_sample

Измеренное значение RTT (параграф 4.3).

R_sqmean

Долговременная оценка квадратного корня из RTT (параграф 4.6).

recover_rate

Дозволенная скорость для восстановления передачи после периода бездействия (параграф 4.4).

recv_limit

Предел скорости передачи, рассчитанный по X_recv_set (параграф 4.3).

s

Номинальный размер пакета в байтах.

S

Порядковый номер.

t_delay

Сообщённое получателем время между приёмом последнего пакета и генерацией пакета обратной связи (параграф 3.2.2).

t_delta

Параметр, позволяющий гибко управлять временем передачи (параграф 8.3).

t_gran

Гранулярность таймера планирования операционной системы (константа) (параграф 8.3).

t_ipi

Межпакетный интервал при передаче (параграф 4.6).

t_mbi

Максимальное значение RTO для TCP (константа) (параграф 4.3).

t_recvdata

Временная метка последнего принятого пакета (параграф 3.2.2).

timer_limit

Предел скорости передачи в результате завершения отсчёта таймера обратной связи (параграф 4.4).

tld

Время последнего удвоения (параграф 4.2).

t_now

Текущее время (параграф 4.3).

t_RTO

Estimated RTO of TCP (параграф 4.3).

X

Допустимая скорость передачи, ограничиваемая по скорости приёма.

X_Bps

Рассчитанная скорость передачи в байт/сек (параграф 3.1).

X_pps

Рассчитанная скорость передачи в пакет/сек (параграф 3.1).

X_inst

Разрешённая мгновенная скорость передачи (параграф 4.6).

X_recv

Оценённая получателем скорость приёма. (параграф 3.2.2).

X_recv_set

Небольшой набор недавних значений X_recv (параграф 4.3).

X_target

Результирующая скорость передачи после первого факта потерь (параграф 6.3.1).

W_init

Начальное окно TCP (константа) (параграф 4.2).

Приложение B. Начальное значение таймера обратной связи

Почему в качестве начального значения таймера обратной связи TFRC используется значение 2 секунды вместо 3 секунд, рекомендуемых в качестве начального значения таймера повтора передачи TCP в соответствии с [RFC2988]? Нет каких-либо существенных причин, по которым таймер обратной связи TFRC должен иметь начальное значение, совпадающее с начальным значением таймера повтора TCP. Таймер повторной передачи в TCP используется не только для снижения скорости в ответ на насыщение, но и для повтора передачи пакетов, которые предполагаются потерянными в сети. Таймер обратной связи TFRC, напротив, используется только для снижения дозволенной скорости передачи и не служит триггером для передачи новых пакетов. В результате не возникает какой-либо опасности для сети в результате того, что начальное значение таймера обратной связи TFRC меньше рекомендуемого значения таймера повторной передачи TCP.

Далее, пока отсчёт таймера обратной связи не завершился TFRC обеспечивает более медленный отклик на насыщение, нежели TCP, и применяемое в TFRC ограничение скорости передачи на основе скорости приёма. менее точно, нежели используемые в TCP окна и синхронизация подтверждений, поэтому таймер обратной связи является важным механизмом защиты TFRC. С учётом сказанного, представляется вполне разумным и обоснованным выбор в качестве начального значения таймера обратной связи значения, меньшего по сравнению с начальным значением таймера повторной передачи TCP.

Приложение C. Отклик на период ограниченной передачи

В последующих работах может использоваться иной отклик на значение скорости приёма. в период ограниченной передачи данных и факты потерь пакетов в такой период.

В частности [RFC2861] предлагает экспериментальный механизм проверки корректности окна насыщения (CWV[1]) для TCP. Далее мы будем использовать термин «стандартный TCP» для механизмов контроля насыщения, описанных в [RFC2581] и [RFC2581bis]. [RFC2861] задаёт иной отклик на периоды бездействия или ограниченной передачи, нежели принято в стандартном TCP. При использовании CWV отправитель TCP снижает вдвое размер окна насыщения после каждого RTO в течение периода бездействия, вплоть до начального размера окна. Аналогично при использовании CWV отправитель TCP уменьшает вдвое окно насыщения после каждого RTO в периоды ограниченной передачи.

В этом документе уже задан отклик TFRC на периоды бездействия, похожий на поведение TCP с CWV. Однако этот документ не задаёт отклик TFRC на периоды ограниченной передачи данных по аналогии с CWV. Добавление такого механизма в TFRC будет требовать изменения одной строки в псевдокоде п. 4) параграфа 4.3. В частности, отклик отправителя на пакет обратной связи:

```
If (если весь интервал, закрываемый пакетом обратной связи, характеризуется
     ограниченной передаче данных) {
   If (пакет обратной связи говорит о новом факте потерь или росте частоты
                потерь p) {
       уменьшить вдвое элементы X_recv_set;
       X_recv = 0.85 * X_recv;
       Maximize X_recv_set();
       recv_limit = max(X_recv_set);
   } Else {
       Maximize X_recv_set();
       recv_limit = 2 * max (X_recv_set);
   }
}
```

будет заменён на:

```
If (если весь интервал, закрываемый пакетом обратной связи, характеризуется
      ограниченной передаче данных) {
    Старые элементы X_recv_set умножаются на 0,85;
    If (пакет обратной связи говорит о новом факте потерь или росте частоты
                потерь p) {
        Новое значение X_recv умножается на 0,85.
    }
    Maximize X_recv_set();
    recv_limit = 2 * max (X_recv_set);
}
```

В частности, если скорость приёма. из предыдущего периода ограниченной передачи сохранена в X_recv_set, приведённое выше изменение п. 4) будет приводить к умножению этой скорости на 0,85 всякий раз при получении пакета обратной связи и выполнении приведённого выше псевдокода. В результате после 4 последовательных периодов кругового обхода скорость приёма. из предыдущего периода ограниченной передачи будет умножаться на $0{,}85^4 = 0{,}52$. Таким образом, изменение одной строки в п. 4) параграфа 4.3 будет приводить к снижению дозволенной скорости вдвое за каждые 4 периода кругового обхода, в течение которых отправитель ограничивал передачу данных.

[1]Congestion Window Validation

По самой природе X_recv_set этот механизм никогда не будет снижать дозволенную скорость передачи ниже удвоенного последнего значения скорости приёма.

Отметим, что для сохранения похожего на CWV стиля откликов на ограничение передачи данных сохраняется значение

```
recv_limit = 2 * max (X_recv_set);
```

взамен

```
recv_limit = max (X_recv_set);
```

для реагирования на потери в периоды ограниченной передачи. Такой ослабленный отклик на факт потери дозволен потому, что поведение в стиле CWV само по себе ограничивает скорость флуктуаций скорости передачи в периоды ограниченной передачи данных.

C.1. Долгое бездействие или ограниченный объем данных

Таблица 1. Отклики на долгое бездействие и ограничение передачи данных.

Протокол	***Долгие периоды бездействия***	***Долгие периоды ограниченной передачи***
Стандартный TCP	Начальный размер окна	Окно увеличивается для каждого cwnd
TCP с CWV	Половина окна (не меньше cwnd)	Уменьшение окна наполовину
Стандартный TFRC	Снижение скорости вдвое (не менее 2 пакетов за RTT). В течение 1 RTT после передачи пакета скорость ограничена значением X_recv	Скорость передачи ограничена удвоенной скоростью приёма
Обновленный TFRC	Снижение скорости вдвое (не ниже начальной)	Скорость ограничена удвоенным значением max (текущее значение X_recv, скорость приёма до периода ограничения)

Таблица 1 описывает отклики стандартного TCP [RFC2581], TCP с контролем насыщения CWV[RFC2861], стандартного TFRC [RFC3448] и обновлённого TFRC (данный документ) на периоды длительного бездействия или ограниченной передачи. Длинным считается период, продолжительность которого не менее RTO.

Стандартный TCP после долгого бездействия.

Для стандартного TCP [RFC2581] указывает, что после периода бездействия не менее RTO TCP **следует** установить окно насыщения не более изначального размера окна (точнее говоря, RFC 2581 указывает, что отправителю TCP следует установить для cwnd значение изначального размера окна, если отправитель не передавал данных в течение интервала, превышающего тайм-аут повтора).

Стандартный TCP после длительного ограничения передачи.

Стандартный TCP [RFC2581] не снижает размер окна насыщения после периода ограниченной передачи данных, в течение которого окно насыщения не использовалось полностью. Стандартный TCP в [RFC2581] использует значения FlightSize (объем остающихся в сети данных) только для установки порога замедленного старта по истечении тайм-аута повтора передачи. Стандартный TCP не ограничен синхронизацией подтверждений в течение периода ограниченной передачи.

Слабый отклик стандартного TCP на периоды ограниченной передачи существенно отличается от строго отклика на периоды бездействия.

TCP с CWV после долгого бездействия.

В качестве экспериментального варианта [RFC2861] предлагает более сдержанный отклик на период бездействия, нежели принято в стандартном TCP, где в период бездействия отправитель TCP вдвое снижает значение cwnd после каждого периода RTO (вплоть до начального значения cwnd).

TCP с CWV после длительного ограничения передачи.

В качестве экспериментального варианта [RFC2861] предлагает более строгий отклик на периоды ограниченной передачи данных, нежели принято в стандартном TCP, где после каждых RTO секунд ограниченной передачи окно насыщения уменьшается вдвое по сравнению с используемым размером окна насыщения.

Отклик TCP с CWV на период бездействия похож на отклик в периоды ограниченной передачи данных. TCP с CWV вносит меньшие по сравнению со стандартным TCP ограничений в отклики на период бездействия и большие ограничения в отклики на периоды ограниченной передачи данных.

Стандартный TFRC после долгого бездействия.

Для стандартного TFRC в [RFC3448] указано, что дозволенная скорость передачи снижается вдвое после каждых RTO секунд периода бездействия. Дозволенная скорость передачи не снижается до значений меньше 2 пакетов за период RTT после периода бездействия. Первый пакет обратной связи после периода бездействия сообщает о скорости приёма. в пакетах за период кругового обхода - эта скорость используется для ограничения скорости передачи. Стандартный TFRC эффективно выполняет процедуру замедленного старта, начиная с этого значения дозволенной скорости.

Стандартный TFRC после длительного ограничения передачи.

[RFC3448] не делает различий между периодами бездействия и периодами ограниченной передачи. По этой причине дозволенная скорость передачи ограничивается значением, не превышающим удвоенную скорость приёма. в течение и после периода ограниченной передачи. Это очень жёсткое ограничение, более жёсткое, нежели в стандартном TCP и TCP с CWV.

Обновленный TFRC после долгого бездействия.

Для обновлённого TFRC в данном документе указано, что дозволенная скорость снижается вдвое за каждые RTO секунд периода бездействия. В результате бездействия дозволенная скорость не снижается до значений меньше

начальной скорости передачи. Первый пакет обратной связи после периода бездействия сообщает о скорости приёма. в 1 пакет за период кругового обхода. Однако отправитель в обновлённом TFRC не использует это значение скорости приёма. для ограничения скорости передачи. Таким образом, обновленный TFRC отличается от стандартного тем, что устанавливается более низкий предел снижения скорости передачи, а отклик на первый пакет обратной связи после периода бездействия является более эффективным.

Обновленный TFRC после длительного ограничения передачи.

Для обновлённого TFRC в настоящем документе внесены различия между периодами бездействия и периодами ограниченной передачи. Как указано в параграфе 4.3, в периоды ограниченной передачи обновленный TFRC помнит скорость приёма. до начала ограничений и не снижает дозволенную скорость ниже удвоенного значения сохранённой в памяти скорости приёма. Это похоже на отклик стандартного TCP, но достаточно сильно отличается от весьма ограниченного отклика стандартного TFRC на периоды ограниченной передачи. Однако отклик обновлённого TFRC не столь консервативен, как отклик TCP с CWV, где окно насыщения постепенно уменьшается до реального размера окна в период ограниченной передачи.

Отметим, что для современных реализаций TCP окно насыщения в общем случае не увеличивается в период ограниченной передачи (когда текущее окно насыщения не используется полностью) [MAF05] (параграф 5.7). В обновлённом TFRC подобного механизма нет.

Восстановление после периодов бездействия или ограниченной передачи.

Когда TCP снижает размер окна насыщения после бездействия или ограниченной передачи, TCP может установить порог замедленного старта ssthresh, позволяющий отправителю TCP начать замедленный возврат к старому значению скорости передачи по завершении периода бездействия или ограниченной передачи. Однако в TFRC даже при ограничении скорости передачи отправителя TFRC удвоенным значением предшествующей скорости приёма. отправитель может удвоить скорость передачи по сравнению с предыдущим периодом кругового обхода, если это позволяет уравнение пропускной способности. Таким образом, TFRC не требуются механизмы, типа используемого в TCP порога замедленного старта ssthresh для выполнения процедуры замедленного старта после периодов бездействия или ограниченной передачи.

В перспективе одним из направлений развития является добавление механизмов проверки окна насыщения (CWV) для откликов TFRC на периоды ограниченной передачи. В настоящее время, следуя стандартному TCP, в периоды ограниченной передачи обновленный TFRC не ограничивают скорость передачи в зависимости от скорости приёма.

C.2. Короткое бездействие или ограниченный объем данных

Таблица 2 показывает отклики стандартного TCP [RFC2581], TCP с CWV [RFC2861], стандартного TFRC [RFC3448] и обновлённого TFRC (данный документ) на короткие периоды бездействия или ограниченной передачи. Коротким считается период, продолжительность которого меньше RTT.

Таблица 2. Отклики на короткое бездействие и ограничение передачи данных.

Протокол	***Короткие периоды бездействия***	***Короткие периоды ограниченной передачи***
Стандартный TCP	Пик размером до cwnd	Пик размером до cwnd
TCP с CWV	Пик размером до cwnd	Пик размером до cwnd
Стандартный TFRC	?	?
Обновленный TFRC	Пик размером до RTT неиспользованных кредитов на передачу	Пик размером до RTT неиспользованных кредитов на передачу

Таблица 2 показывает, что отклики обновлённого TFRC на короткие периоды бездействия или ограниченной передачи подобны откликам стандартного TCP и TCP с CWV. Для коротких периодов бездействия или ограниченной передачи TCP ограничивается только размером неиспользуемого окна насыщения, а обновлённых TFRC - только число неиспользованных кредитов на передачу (до RTT). Для стандартного TFRC [RFC3448] не задаёт в явном виде поведения в части неиспользованных кредитов на передачу.

C.3. Среднее бездействие или ограниченный объем данных

Таблица 3 показывает отклики стандартного TCP [RFC2581], TCP с CWV [RFC2861], стандартного TFRC [RFC3448] и обновлённого TFRC (данный документ) на периоды бездействия или ограниченной передачи средней продолжительности. Средними считаются периоды продолжительностью более RTT, но меньше RTO.

Таблица 3. Отклики на среднее по продолжительности бездействие и ограничение передачи данных.

Протокол	***Средние периоды бездействия***	***Средние периоды ограниченной передачи***
Стандартный TCP	Пик размером до cwnd	Пик размером до cwnd
TCP с CWV	Пик размером до cwnd	Пик размером до cwnd
Стандартный TFRC	?	Ограничено X_recv
Обновленный TFRC	Пик размером до RTT неиспользованных кредитов на передачу	Пик размером до RTT неиспользованных кредитов на передачу

Таблица 3 показывает, что отклики обновлённого TFRC на периоды бездействия или ограниченной передачи средней продолжительности подобны откликам стандартного TCP и TCP с CWV. Для средних по продолжительности периодов бездействия или ограниченной передачи TCP ограничивается только размером неиспользуемого окна насыщения. В таких же ситуациях обновленный TFRC ограничивается только числом неиспользованных кредитов на передачу (до RTT). Для периодов ограниченной передачи средней продолжительности стандартный TFRC будет ограничен значением X_recv из последнего пакета обратной связи. Обновленный TFRC, напротив, не ограничивается скоростью приёма. в течение периода ограниченной передачи, которая покрывает весь подтверждённый пакетом обратной связи период кругового обхода. Для стандартного TFRC [RFC3448] не задаёт явно поведение в части неиспользованных кредитов на передачу.

C.4. Потери в период ограниченной передачи данных

В этом параграфе обсуждаются отклики на потери в период ограниченной передачи данных.

Таблица 4. Отклики на потери в период ограниченной передачи данных.

Протокол	***Отклик на потери***
Стандартный TCP	Установить для ssthresh и cwnd значение FlightSize/2
TCP с CWV	Установить для ssthresh и cwnd значение FlightSize/2
Стандартный TFRC	Рассчитать X_Bps и передавать не более 2*X_Bps
Обновленный TFRC	Рассчитать X_Bps и передавать не более recv_limit. Изменить X_recv_set

В TCP [RFC2581] отклик на потери в период ограниченной передачи данных не отличается от откликов на потери в любом другом состоянии TCP. Этот отклик состоит в установке окна насыщения в размере половины FlightSize, где FlightSize определяет размер переданных, но ещё не подтверждённых данных. Таким образом, после потери в период ограниченной передачи отправитель TCP должен вдвое снизить скорость передачи, как это происходит при обычных условиях в ответ на потерю.

В стандартном TFRC отклик на потери в период ограниченной передачи совпадает с откликом на потери в любом другом состоянии TFRC. Скорость передачи ограничивается значением X_Bps из уравнения пропускной способности, а также удвоенным последним значением скорости приёма. X_recv. В результате после потери в период ограниченной передачи может передавать данные со скоростью до 2*X_recv, даже в тех случаях, когда X_Bps из уравнения пропускной способности разрешает большую скорость.

В обновлённом TFRC использование скорости приёма. X_recv в периоды ограниченной передачи для отклика на потери в такие периоды изменено; скорость передачи ограничена значением recv_limit и отправитель может запоминать значения скорости приёма. X_recv непосредственно перед началом периода ограниченной передачи. Это позволяет отправителю увеличивать скорость передачи в период ограничений более, чем вдвое, вплоть до скорости приёма. перед началом ограничений (если это дозволяет значение X_Bps, полученное из уравнения пропускной способности). Такое поведение похоже на принятую в стандартном TCP практику отказа от снижения размера окна в период ограниченной передачи данных (при отсутствии потерь).

Как и в стандартном TFRC отправитель в обновлённом TFRC передаёт данных меньше, нежели позволяет значение X_Bps из уравнения пропускной способности. После потери отправитель по-прежнему может не хотеть передавать данных больше, чем позволяет новое значение X_Bps, которое учитывает факт потери. В обновленный TFRC добавлен механизм постепенного ограничения скорости передачи после потерь в период ограниченной передачи. В отличие от отклика TCP в форме установки для размера окна насыщения cwnd значения в половину FlightSize, дополнительные механизмы в обновлённом TFRC используют принятую в TFRC практику медленного изменения откликов как для увеличения, так и для снижения допустимой скорости передачи.

Это делается в обновлённом TFRC (п. 4) параграфа 4.3) путём уменьшения элемента X_recv_set после потери в период ограниченной передачи и позволяет отправителю передать до max(X_recv_set) данных, вместо 2*max(X_recv_set), в период кругового обхода, следующий непосредственно за получением информации о потере. Таким образом, возможность передачи более, чем 2*X_recv (последнее полученное значение) в интервале ограниченной передачи достигается за счёт введения дополнительного механизма снижения дозволенной скорости после потерь в период ограниченной передачи.

Для иллюстрации отклика TFRC на потери в период ограниченной передачи данных рассмотрим несколько примеров.

Таблица 5. Потеря после периода ограниченной передачи.

Этап 1:	Нет ограничения передачи.
	Передаётся 100 пакетов за период кругового обхода (RTT).
Этап 2:	Ограниченная передачи, 10 пакетов за RTT.
Этап 3:	Нет ограничения передачи.
	Передаётся 100 пакетов за RTT, как позволяет X_Bps.
	Потеря пакета в первом RTT этапа 3.
	Обновление X_Bps.
Отклик обновлённого TFRC: незначительное снижение дозволенной скорости передачи в зависимости от числа пакетов с момента последней потери.	

Пример 1, потеря **после** периода ограниченной передачи. Этот пример показывает, что потери после завершения периода ограниченной передачи данных решаются с помощью уравнения пропускной способности для X_Bps.

В примере 1, где пакет теряется в первый RTT этапа 3, это будет отражаться измененным значением X_Bps и дальнейшие потери пакетов будут приводить к дополнительному снижению X_Bps. В частности, следуя стандартному для TFRC уравнению пропускной способности [FHPW00] (параграф A.2), дозволенная скорость передачи TFRC будет снижена вдвое после пяти периодов кругового обхода подряд с потерями пакетов.

Пример 2, незначительное ограничение передачи данных. В этом примере рассматривается потеря пакета в период ограниченной передачи, когда отправитель передаёт чуть меньше данных, чем ему разрешено.

Рассмотрим соединение обновлённого TFRC, где отправитель передаёт 100 пакетов за RTT и начинает ограничивать передачу 99 пакетами по причине нехватки данных от приложения (т. е., за каждый период интервала ограниченной передачи отправитель может передать ещё один пакет). Если в период ограниченной передачи теряется пакет,

Таблица 6. Потеря при незначительном ограничении передачи.

Этап 1:	Нет ограничения передачи. 100 пакетов за RTT.
Этап 2:	Ограниченная передачи, 99 пакетов за RTT.
	Пакет теряется на этапе 2.
Отклик обновлённого TFRC: незначительное снижение дозволенной скорости передачи (до 85 пакетов за RTT или чуть меньше) в зависимости от числа пакетов с момента последней потери.	

дозволенная скорость передачи снижается до min(X_Bps, recv_limit), где оба значения X_Bps и recv_limit незначительно отличаются от дозволенной скорости передачи.

Таблица 7. Потеря одного пакета.

Этап 1:	Нет ограничения передачи. 100 пакетов за RTT.
Этап 2:	Ограниченная передачи, 10 пакетов за RTT.
Этап 3:	Нет передачи в течение 2 периодов RTT.
Этап 4:	Передаётся 1 пакет, который приходит с маркером ECN.
Отклик обновлённого TFRC: снижение дозволенной скорости передачи до 50 пакетов за RTT. Для каждой потери пакета в период ограниченной передачи сохранённое значение X_recv до начала ограничения передачи уменьшается вдвое.	

Пример 3, потеря 1 пакета в период ограниченной передачи. В этом примере рассматривается потеря единственного пакета в период ограниченной передачи после того, как отправитель не передавал пакетов в течение 2 RTT.

Рассмотрим соединение обновлённого TFRC, где отправитель передавал 100 пакетов за RTT и начал ограничивать передачу на уровне 10 пакетов за RTT, а потом не передавал пакетов в течение 2 периодов RTT, после чего передал один пакет, который был принят с маркером ECN. В этом случае обновленный TFRC для каждого факта потери в период ограниченной передачи будет снижать вдвое сохранённое значение скорости перед началом периода ограниченной передачи X_recv.

Таблица 8. Потери после увеличения скорости передачи.

Этап 1:	Нет ограничения передачи. 100 пакетов за RTT.
Этап 2:	Ограниченная передачи, 1 пакет за RTT.
Этап 2:	Ограниченная передачи, 20 пакетов за RTT.
	Теряется несколько пакетов в каждом RTT этапа 3.
	В течение этапа 3 отправитель желает передавать 20 пакетов за RTT.
Отклик обновлённого TFRC: при каждой потере пакетов в течение периода ограниченной передачи сохранённое значение скорости приёма до начала ограничения X_recv уменьшается вдвое, а последнее полученное значение X_recv умножается на 0,85.	

Пример 4, Потери после увеличения скорости передачи в период ограничения. В этом примере рассматриваются потери в то время, когда отправитель существенно повышает скорость передачи данных в период ограниченной передачи.

Рассмотрим соединение обновлённого TFRC, где отправитель передавал 100 пакетов за RTT, затем ограничил передачу до 1 пакета за RTT и снова увеличил до 20 пакетов. После этого неоднократно возникали потери пакетов.

В этом случае обновленный TFRC при каждом факте потери данных в период ограниченной передачи будет снижать вдвое сохранённое значение скорости приёма. до ограничения X_recv, а последнее значение X_recv будет умножаться на 0,85.

C.5. Другие варианты

Другим путём оценки обновлённого TFRC является сравнение поведения TCP, стандартного TFRC и обновлённого TFRC для соединений с чередованием периодов занятости и бездействия, периодов бездействия и ограниченной передачи, а также чередования бездействия и ограниченной передачи в процессе замедленного старта.

C.6. Оценка отклика TFRC на периоды бездействия

В этом параграфе проводится оценка отклика обновлённого TFRC на периоды бездействия и ограниченной передачи.

Одним из недостатков стандартного TFRC является то, что жёсткий отклик на периоды бездействия или ограниченной передачи может вызывать у приложения желание дополнить поток данных ненужной информацией для предотвращения простоев. Поэтому в обновлённом TFRC используется менее жёсткий отклик на периоды бездействия или ограниченной передачи. Ведутся работы (например, Faster Restart [KFS07]), которые могут также снизить желание приложений заполнять паузы пустыми данными за счёт ускорения процедуры восстановления после периода бездействия. Будут полезны дальнейшие исследования для более детального понимания взаимодействия между механизмами контроля насыщения TCP или TFRC и приложениями, стремящимися заполнить паузы пустыми данными в периоды бездействия или ограниченной передачи.

TCP с контролем окна насыщения (CWV), описанный в параграфе C.1, является экспериментальным стандартом, задающим для отправителя TCP медленное снижение размера окна насыщения в период бездействия или ограниченной передачи [RFC2861]. Хотя отклики TFRC и обновлённого TFRC на периоды бездействия похожи на отклики TCP с CWV, отклик обновлённого TFRC на периоды ограниченной передачи менее консервативны, нежели аналогичные отклики TCP с CWV (а отклики стандартного TFRC на периоды ограниченной передачи были более консервативны по сравнению с откликами CWV). В будущих работах этот документ может пересматриваться и в отклик обновлённого TFRC на периоды ограниченной передачи может включать медленное снижение допустимой скорости передачи; в Приложении C приводится один из возможных механизмов такого снижения. Такие модификации станут более вероятными, если механизм контроля окна насыщения CWV получит в IETF статус Proposed Standard[1] для TCP.

Литература

Нормативные документы

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

[RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

Дополнительная литература

[BRS99] Balakrishnan, H., Rahul, H., and Seshan, S., "An Integrated Congestion Management Architecture for Internet Hosts,"[2] Proc. ACM SIGCOMM, Cambridge, MA, September 1999.

[CCID-4] Floyd, S., and E. Kohler, "Profile for DCCP Congestion Control ID 4: the Small-Packet Variant of TFRC Congestion Control", Work in Progress, February 2008.

[FHPW00] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications"[3], August 2000, Proc SIGCOMM 2000.

[FHPW00a] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications: the Extended Version"[4], ICSI tech report TR-00-03, March 2000.

[FF99] Floyd, S., and K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking[5], August 1999.

[KFS07] E. Kohler, S. Floyd, and A. Sathiaseelan, "Faster Restart for TCP Friendly Rate Control (TFRC)", Work in Progress, November 2007.

[MAF05] A. Medina, M. Allman, and S. Floyd, "Measuring the Evolution of Transport Protocols in the Internet", ACM Computer Communications Review[6], April 2005.

[PFTK98] Padhye, J. and Firoiu, V. and Towsley, D. and Kurose, J., "Modeling TCP Throughput: A Simple Model and its Empirical Validation"[7], Proc ACM SIGCOMM 1998.

[RFC2140] Touch, J., "TCP Control Block Interdependence", RFC 2140, April 1997.

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

[RFC2581bis] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", Work in Progress, April 2008.

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

[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.

[RFC3448Err] RFC 3448 Errata, <http://www.rfc-editor.org/errata_search.php?rfc=3448>.

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

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

[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.

[RFC4828] Floyd, S. and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007.

[W00] Widmer, J., "Equation-Based Congestion Control", Diploma Thesis, University of Mannheim, February 2000, <http://www.icir.org/tfrc/>.

Адреса авторов

Sally Floyd

[1]Предложенный стандарт.
[2]Документ доступен по ссылке http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-771.pdf. *Прим. перев.*
[3]Документ доступен по ссылке http://www.icir.org/tfrc/tcp-friendly.pdf. *Прим. перев.*
[4]Документ доступен по ссылке. *Прим. перев.*
[5]Документ доступен по ссылке http://www.icir.org/floyd/papers/collapse.may99.pdf. *Прим. перев.*
[6]Документ доступен по ссылке http://www.icir.org/floyd/papers/TCPevolution-May2004.pdf. *Прим. перев.*
[7]Документ доступен по ссылке http://www.sigcomm.org/sigcomm98/tp/paper25.pdf. *Прим. перев.*

ICSI
1947 Center St, Suite 600
Berkeley, CA 94708
EMail: floyd@icir.org

Mark Handley,
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
UK
EMail: M.Handley@cs.ucl.ac.uk

Jitendra Padhye
Microsoft Research
EMail: padhye@microsoft.com

Joerg Widmer
DoCoMo Euro-Labs
Landsberger Strasse 312
80687 Munich
Germany
EMail: widmer@acm.org

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Интеллектуальная собственность