Internet Engineering Task Force (IETF)
Request for Comments: 6182
Category: Informational
ISSN: 2070-1721

A. Ford
Roke Manor Research
C. Raiciu
M. Handley
University College London
S. Barre
Universite catholique de Louvain
J. Iyengar
Franklin and Marshall College

March 2011

Architectural Guidelines for Multipath TCP Development

Архитектурные рекомендации для разработки Multipath TCP

Аннотация

Хосты часто соединены множеством путей, но TCP ограничивает связь одним путем на соединение. Использование ресурсов сети будет более эффективным при возможности использовать одновременно несколько путей. Это должно улучшить взаимодействие с пользователем за счет повышения отказоустойчивости и пропускной способности.

В документе представлены архитектурные рекомендации для разработки транспортного протокола с множеством путей (Multipath Transport Protocol) и указаны способы объединения этих архитектурных элементов для создания протокола Multipath TCP (MPTCP). В документе перечислены некоторые высокоуровневые решения, обеспечивающие основу для создания протокола MPTCP на основе рассмотренных архитектурных требований.

Статус документа

Документ не содержит какого-либо стандарта (Internet Standards Track) и публикуется с информационными целями.

Документ является результатом работы IETF[1] и представляет согласованное мнение сообщества IETF. Документ был представлен на общее обозрение и одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 5741.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке http://www.rfc-editor.org/info/rfc6182.

Оглавление

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

1. Введение

По мере развития Internet потребности в ресурсах постоянно растут, но зачастую эти ресурсы (в частности, пропускная способность) не могут быть использованы полностью по причине протокольных ограничений, присущих как конечным системам, так и сети. Если эти ресурсы можно было бы использовать одновременно, взаимодействие с конечным пользователем можно было бы существенно улучшить. Это также снизило бы расходы на сетевую инфраструктуру, которая требуется для реализации потребностей конечных пользователей. За счет использования пулов ресурсов [3] доступные ресурсы можно объединить так, чтобы они представлялись пользователю единым ресурсом.

Транспортировка по множеству путей предназначена для реализации целей объединения ресурсов путем одновременного использования множества несвязанных (или частично связанных) путей через сеть. Это обеспечивает два важных преимущества.

- Рост отказоустойчивости соединений за счет предоставления множества путей.
- Рост эффективности использования ресурсов и доступных хостам возможностей сети.

Multipath TCP является измененным вариантом TCP [1], который реализует доставку по нескольким путям, объединяя их в транспортное соединение, прозрачное для приложения. Протокол Multipath TCP в первую очередь связан со сквозным использованием множества путей, когда один или оба конечных хоста являются многодомными. Протокол может также применяться при наличии в сети множества путей, которыми конечный хост может манипулировать, например, с помощью разных портов для ECMP[1] [4].

Протокол MPTCP, определенный в [5], задает концепцию Multipath TCP. В этом документе рассмотрены базовые принципы архитектуры Multipath TCP для достижения целей, заявленных в разделе 2, а также основные решения, лежащие в основе MPTCP (раздел 5).

Хотя функции для работы с многодомными хостами и множеством путей не являются новыми для транспортных протоколов (примером может служить SCTP[2] [6]), целями MPTCP является более широкое развертывание и совместимость с сетью. Эти цели, подробно рассматриваемые в разделе 2, относятся к представлению MPTCP сети (чтобы не знающие о MPTCP элементы воспринимали его как TCP) и приложению (путем предоставления услуг, эквивалентных TCP не поддерживающим MPTCP приложениям).

Основными целями этого документа служат (i) описание назначения транспортировки по нескольким путям (назначение MPTCP), (ii) описание архитектурной основы MPTCP (обсуждение другого транспорта со множеством путей) и (iii) обсуждение и документирование высокоуровневых решений, принятых в MPTCP, а также их влияние.

Сопровождающий этот архитектурный обзор документ детализирует расширения протокола [5], алгоритмы контроля перегрузок [7] и вопросы прикладного уровня [8]. Совместно эти документы полностью описывают устройство Multipath TCP. Отметим, что конкретные компоненты могут заменяться в соответствии с определенной здесь декомпозицией уровней и функций.

1.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с RFC 2119 [2].

1.2. Терминология

Regular/Single-Path TCP - обычный (единственный) путь TCP
Стандартная версия TCP [1], применяемая сегодня для взаимодействия между парой адресов IP и портов.

Multipath TCP - TCP с множеством путей
Измененная версия протокола TCP для одновременной поддержки множества путей между хостами.

Path - путь
Последовательность каналов между отправителем и получателем, определяемая в контексте документа парой адресов отправителя и получателя.

Host - хост
Конечный хост, инициирующий или завершающий соединение Multipath TCP.

MPTCP
Предложенное в [5] расширение протокола для реализации Multipath TCP.

Subflow - субпоток
Поток сегментов TCP, передаваемых по отдельному пути, являющийся частью соединения Multipath TCP.

(Multipath TCP) Connection - соединение (Multipath TCP)
Множество из одного или нескольких субпотоков, объединенных для предоставления услуги Multipath TCP приложению на хосте.

[1] Equal Cost MultiPath - множество равноценных путей.
[2] Stream Control Transmission Protocol - протокол управления передачей потоков.

1.3. Эталонный сценарий

Схема на рисунке 3 иллюстрирует типичный пример использования Multipath TCP, где два хоста A и B взаимодействуют между собой. Эти хосты являются многодомными и многоадресными, обеспечивая два несвязанных соединения через Internet. Адреса хостов обозначены A1, A2, B1 и B2. В результате можно организовать до 4 путей между хостами A1-B1, A1-B2, A2-B1, A2-B2.

Рисунок 1. Простой пример применения Multipath TCP.

Сценарий будет работать с любым числом адресов (от 1) на каждом хосте, а также при любом числе путей между хостами, начиная с 2 (т. е. num_addr(A) * num_addr(B) > 1). Пути, образуемые комбинациями адресов, не обязаны быть полностью разделенными в сети Internet и вопросы беспристрастности при возникновении «пробок» на общем пути должны обрабатываться контроллером перегрузок Multipath TCP. Кроме того, пути через Internet зачастую не обеспечивают «чистого» сквозного обслуживания и могут включать промежуточные устройства, такие как NAT и МСЭ[1].

2. Цели

В этом разделе означены основные цели Multipath TCP, разделенные на две категории - функциональные цели, управляющие службами и функциями, которые требуются от Multipath TCP, и цели совместимости, определяющие представление Multipath TCP взаимодействующими с этим расширением объектами.

2.1. Функциональные цели

С точки зрения работы по нескольким путям Multipath TCP имеет две функциональных цели.

- *Повышение пропускной способности*. Протокол Multipath TCP **должен** поддерживать одновременное использование множества путей. Для выполнения требований к росту производительности соединению Multipath TCP по нескольким путям **следует** обеспечивать производительность не хуже, чем у отдельного соединения TCP по лучшему из путей.
- *Повышение отказоустойчивости*. Протокол Multipath TCP **должен** поддерживать замену используемых путей для обеспечения устойчивости к отказам, разрешая передавать и повторять сегменты по любому из доступных путей. Из этого следует, что в худшем случае протокол **должен** быть не менее устойчив к отказам, чем наиболее устойчивый из отдельных путей TCP.

Распределение трафика по доступным путям и отклики на перегрузку выполняются в соответствии с принципами объединения ресурсов [3], а дополнительным эффектом достижения этих целей является широкое применение Multipath TCP в Internet, которое должно повысить общую производительность сети за счет избавления от «пробок» и использования по возможности резервных «емкостей» сети.

Кроме того, Multipath TCP **следует** поддерживать автоматическое согласование применения. Хосту, поддерживающему Multipath TCP нужна возможность надежного определения возможностей хоста-партнера в части поддержки требуемых расширений, поскольку при их отсутствии он вынужден будет вернуться к использования обычного протокола TCP.

2.2. Цели совместимости

В дополнение к указанным выше функциональным целям протокол Multipath TCP должен соответствовать множеству целей совместимости для поддержки развертывания в сети Internet. Эти цели делятся на несколько категорий.

2.2.1. Совместимость с приложениями

Совместимость с приложениями относится к представлению Multipath TCP в плане доступных для использования API и предоставляемой модели обслуживания.

Протокол Multipath TCP **должен** следовать модели сервиса, используемой TCP [1] - упорядоченная, надежная, ориентированная на байты доставка. Кроме того, соединению Multipath TCP **следует** предоставлять приложению пропускную способность не хуже ожидаемой при работе через одно соединение TCP по любому из доступных путей. Однако Multipath TCP не может обеспечить такой же уровень согласованности пропускной способности и задержки, как одиночное соединение TCP. Эти и другие вопросы применения подробно рассматриваются в [8].

Поддерживающий множество путей эквивалент TCP **должен** сохранять некоторый уровень совместимости с имеющимися TCP API, чтобы существующие приложения могли использовать новый транспорт, просто обновив операционные системы конечных хостов. Это не предполагает использования расширенного API, позволяющего приложениям с поддержкой нескольких путей задать свои предпочтения или пользователям настроить свои системы не так, как принято по умолчанию, например, включив или отключив автоматическое использование расширений для множества путей.

Обычные сессии TCP могут пережить короткие перебои связности, сохраняя состояние на конечных хостах до тайм-аута. Желательная поддержка такой непрерывности сессий и в MPTCP, однако обстоятельства могут быть разными. Если в обычном TCP адреса IP остаются неизменными после потери связности, в MPTCP это может быть не так. Желательно (но не требуется) поддерживать непрерывность сессий «break-before-make[2]». Это вносит ограничения для механизмов защиты, описанные в параграфе 5.8. Тайм-ауты для такой функции настраиваются локально.

2.2.2. Совместимость с сетью

[1]Межсетевой экран - firewall.
[2]Разрыв до создания.

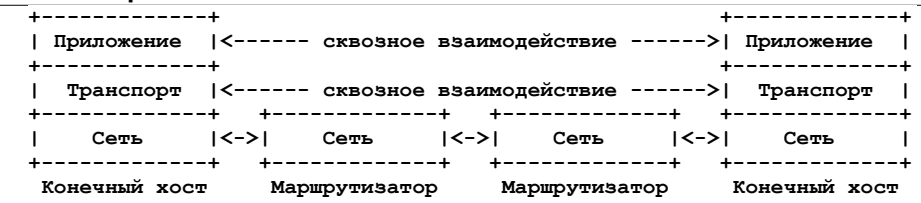

Рисунок 2. Традиционная архитектура Internet.

В традиционной архитектуре Internet сетевые устройства работают на сетевом (L3) и нижележащих уровнях, а вышележащие уровни создаются лишь конечными хостами. Хотя показанная на рисунке 2 архитектура изначально в основном соблюдалась, сейчас она уже не отражает реалии Internet в связи с наличием большого числа промежуточных устройств [9], которые обычно помещаются на транспортный уровень и иной раз полностью перехватывают транспортные соединения, делая первым «сквозным» уровень приложений, как показано на рисунке 3.

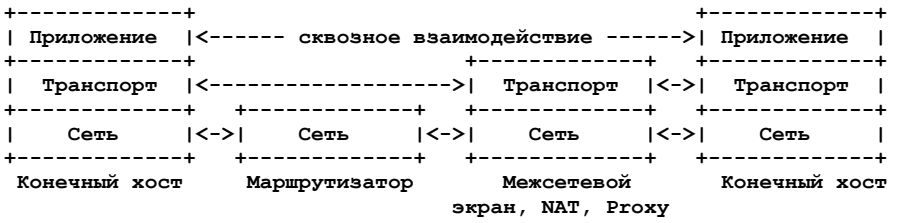

Рисунок 3. Реалии Internet.

Перехват соединений промежуточными устройствами приводит к потере «общей судьбы» (fate-sharing) [10] - «жестким» состояниям, потеря или повреждение которых ведет к потере или повреждению сквозного транспортного соединения.

Цель совместимости с сетью требует от расширения TCP на множество путей сохранения совместимости с сегодняшней сетью Internet, включая разумные усилия по работе через преобладающие промежуточные устройства, такие как МСЭ, трансляторы NAT и прокси для повышения производительности [9]. Это требование основано на осознании важной роли промежуточных устройств в создании «пробок» для любого транспорта, отличного от TCP и UDP, что ограничивает представление Multipath TCP как TCP в соединительных линиях для использования расширений TCP там, где они нужны. Для обеспечения сквозной работы транспорта в Multipath TCP **должна** сохраняться «общая судьба» без каких-либо допущений о поведении промежуточных устройств.

Подробный анализ поведения промежуточных устройств и их влияния на архитектуру Multipath TCP дан в разделе 7. Кроме того, должна сохраняться совместимость с сетью, обеспечивающая возврат от Multipath TCP к обычному TCP, который **должен** происходить для преодоления несовместимости с использованием нескольких путей.

Промежуточные устройства могут также приводить к тому, что некоторые свойства TCP могут поддерживаться в одном субпотоке, но отсутствовать в другом. Обычно это происходит на уровне субпотока (например, селективные подтверждения SACK [11]) и поэтому не будет влиять на свойства соединения в целом. В будущем для любых расширений на уровне соединения TCP следует учитывать возможность сосуществования с MPTCP.

Изменения для поддержки Multipath TCP остаются на транспортном уровне, хотя требуются некоторые сведения о нижележащем сетевом уровне. Расширению Multipath TCP **следует** работать взаимозаменяемо с IPv4 и IPv6, т. е. одно соединение может работать сразу через сети IPv4 и IPv6.

2.2.3. Совместимость с другими пользователями

Для совместимости с сетью и приложениями архитектура должна поддерживать сосуществование потоков Multipath TCP с обычными потоками TCP без чрезмерной или слабой (пока не запрошены специально низкоприоритетные операции такими приложениями, как LEDBAT) конкуренции за пропускную способность. При использовании множества путей **недопустимо** причинять ущерб обычным потокам TCP в узких местах сети, за исключением воздействий, которые могут исходить от других обычных потоков TCP. Множество потоков Multipath TCP в узком месте сети **должно** разделять пропускную способность между субпотоками так же беспристрастно, как для обычных потоков TCP.

2.3. Безопасность

Расширение TCP с поддержкой множества путей будет создавать множество новых угроз, проанализированных в [12]. Защитные цели Multipath TCP заключаются в обеспечении не менее безопасных услуг, нежели для обычных потоков TCP. Это достигается путем комбинирования имеющихся механизмов защиты TCP (возможно измененных для работы с расширениями Multipath TCP) и защиты от новых угроз, связанных с множеством путей. Защитные решения на основе такого подхода представлены в параграфе 5.8.

2.4. Смежные протоколы

Имеется ряд сходств между SCTP [6] и MPTCP, поскольку оба могут использовать множество адресов конечного хоста для поддержки нескольких путей. Основным применением SCTP является поддержка избыточности (резервирования) и мобильности для многодомных хостов (т. е. при изменении адреса конечного хоста на одном из путей), а одновременное использование множества путей не поддерживается. Расширения, предложенные для транспортировки одновременно по нескольким путям [13], еще не стандартизованы. Наиболее широко для транспортировки потоков применяется протокол TCP [1], а SCTP не соответствует требованиям по совместимости с сетью и приложениями, указанным в параграфе 2.2. В части совместимости с сетью имеются проблемы, связанные с тем, что промежуточные устройства (прежде всего NAT) не знают о протоколе SCTP и в результате блокируют его. В части совместимости с приложениями нужно активно выбирать использование протокола SCTP, а из-за проблем с развертыванием это делается достаточно редко. Цели совместимости MPTCP частично основаны на наблюдении за развертыванием SCTP.

3. Архитектурные основы Multipath TCP

В этом разделе представлен один из возможных вариантов транспортной архитектуры, которые по мнению авторов позволит эффективно достичь целей Multipath TCP. Предложенная здесь новая модель Internet основана на идеях, предложенных ранее в Tng[1] [14]. Не будучи единственной возможной архитектурой для поддержки транспорта со множеством путей, Tng включает многие уроки, извлеченные из исследований транспорта и практики разработок, предлагая хорошую отправную точку для рассмотрения имеющейся архитектуры Internet и ее влияния на разработку новых транспортных решений и расширений транспорта Internet.

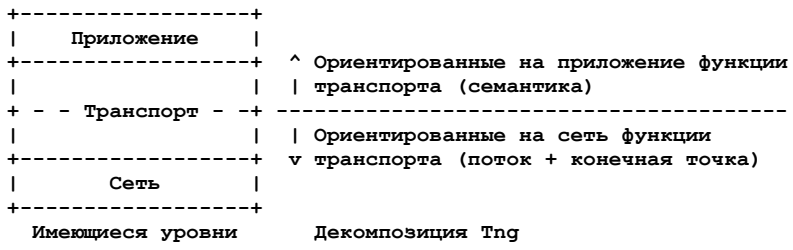

Рисунок 4. Декомпозиция транспортных функций.

Tng делит транспортный уровень на два уровня, ориентированные на приложение и сеть, как показано на рисунке 4. Ориентированный на приложения уровень (семантика) реализует функции, связанные в первую очередь с поддержкой и защитой сквозных коммуникаций между приложениями. Ориентированный на сеть уровень (поток + конечная точка) реализует такие функции, как идентификация конечной точки (использование номеров портов) и контроль перегрузок. Эти функции, традиционно размещаемые на якобы «сквозном» транспортном уровне, на практике вызывают существенную озабоченность операторов и промежуточных устройств, развернутых для исполнения правил использования сети [15] [16] или оптимизации производительности [17]. На рисунке 5 показано взаимодействие промежуточного устройства с разными уровнями в этой модели транспорта с декомпозицией. Ориентированный на приложения уровень обеспечивает сквозную работу, а ориентированный на сеть работает на уровне сегментов и может быть реализована на промежуточных устройствах.

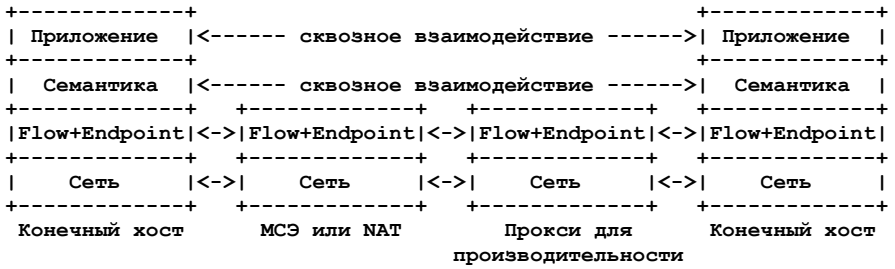

Рисунок 5. Промежуточные устройства в новой модели.

Архитектура MPTCP соответствует декомпозиции Tng, как показано на рисунке 6. Протокол MPTCP, который обеспечивает совместимость приложений за счет предоставления семантики в стиле TCP для глобального упорядочивания данных приложения и надежной доставки, является экземпляром ориентированного на приложения уровня семантики. Компоненты субпотоков TCP, которые обеспечивают совместимость с сетью, проявляя себя как сетевые потоки TCP, являются экземплярами ориентированного на сеть уровня Flow+Endpoint.

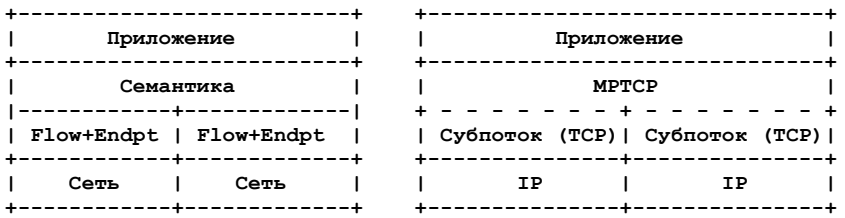

Рисунок 6. Связь между Tng (слева) и MPTCP (справа).

Будучи расширением протокола TCP, MPTCP явно признает наличие промежуточных устройств и задает протокол, работающий «в двух масштабах» - MPTCP обеспечивает сквозное взаимодействие, позволяя компонентам TCP работать на уровне сегментов.

4. Функциональная декомпозиция MPTCP

В двух предыдущих разделах рассмотрены цели разработки Multipath TCP и представлена основа для декомпозиции функций транспортного протокола, чтобы лучше понять предлагаемое решение. Этот раздел основан на проведенном анализе и представляет функциональные компоненты, используемые в MPTCP.

MPTCP использует (с точки зрения сети) стандартные сессии TCP, называемые субпотоками, для обеспечение базового транспорта на каждом пути и сохранения желаемой совместимости с сетью. Связанная с MPTCP информация передается совместимым с TCP способом, хотя это отличается от передаваемой фактически информации и может измениться в будущих версиях. Уровни архитектуры показаны на рисунке 7.

Расположенное ниже приложения расширение MPTCP управляет множеством расположенных под ним субпотоков TCP и должно выполнять перечисленные ниже функции.

[1]Transport next-generation - следующее поколение транспорта.

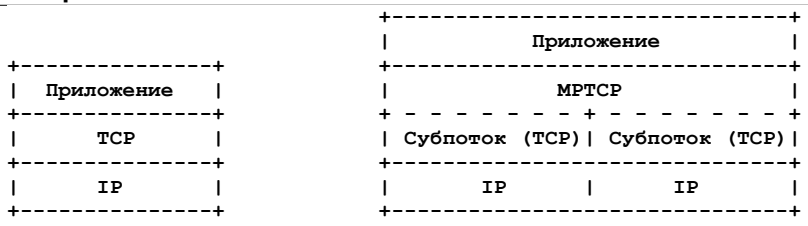

Рисунок 7. Сравнение стеков стандартного TCP и MPTCP.

- *Поддержка путей*. Эта функция находит и использует множество путей между хостами. MPTCP использует наличие множества адресов IP на одном или обоих хостах в качестве индикатора. Функции поддержки путей в протоколе MPTCP служат механизмами сигнализации хостам о дополнительных адресах и организации новых субпотоков, связанных с имеющимся соединением MPTCP.

- *Планирование пакетов*. Эта функция разбивает поток байтов от приложения на сегменты, передаваемые в одном из доступных субпотоков. MPTCP использует отображение последовательности данных, связывая сегменты, отправленные в разных субпотоках, порядковыми номерами на уровне соединения, что позволяет передавать сегменты в разных субпотоках с сохранением порядка на приемной стороне. Планировщик пакетов зависит от информации о доступности путей, предоставляемой средствами поддержки путей, что позволяет передавать сегменты по разным путям с восстановлением порядка у получателя. Планировщик на основе данных о доступности путей, предоставляемых средствами поддержки путей, использует субпотоки для передачи сегментов из очереди. Эта функция отвечает также за восстановление порядка принятых в разных субпотоках TCP пакетов на уровне соединения в соответствии с отображением порядковых номеров.

- *Интерфейс с субпотоками* (отдельные пути TCP). Субпотоки принимают сегменты от планировщика пакетов и передают их по конкретному пути, обеспечивая подтверждаемую доставку хосту. MPTCP использует услуги TCP для совместимости с сетью, а TCP обеспечивает надежную, упорядоченную доставку. TCP добавляет свои порядковые номера в сегменты, используемые для обнаружения потерь и повторной передачи пакетов на уровне субпотока. На приемной стороне субпоток передает собранные данные планировщику пакетов для сборки на уровне соединения. Отображение последовательности данных от планировщика на стороне отправителя позволяет корректно восстановить поток байтов.

- *Контроль перегрузок* координирует распределение трафика между субпотоками при возникновении насыщения в сети. Как указано, этот алгоритм **должен** обеспечить для соединения MPTCP пропускную способность больше пропускной способности отдельного потока TCP через сеть с «пробками». Алгоритм для поддержки механизма задан в [7].

Эти функции работают совместно, как описано ниже. Функция поддержки после обнаружения путей (при необходимости их инициализации) следит за множеством путей между хостами. Планировщик пакетов получает поток данных от приложения для отправки в сеть и выполняет требуемые для передачи операции (такие как сегментирование данных и добавление порядковых номеров на уровне соединения). Затем субпотоки добавляют свои номера, подтверждения (ACK) и передают сегменты в сеть. Принимающий субпоток (при необходимости) восстанавливает порядок в потоке данных и передает поток приложению. Функция контроля перегрузок работает как часть планировщика пакетов, чтобы определить, какие сегменты нужно передать, с какой скоростью и в какой субпоток.

5. Устройство протокола

Очевидно, что имеется много вариантов разработки расширения TCP с поддержкой нескольких путей. Однако рассмотренные выше цели ограничивают набор возможных решений, оставляя сравнительно небольшой выбор во многих областях. В этом разделе очерчены высокоуровневые решения, основанные на архитектуре, описанной в разделе 3, с учетом устройства расширения MPTCP [5].

5.1. Порядковые номера

MPTCP использует два типа порядковых номеров - на уровне соединения и на уровне каждого субпотока. Это позволяет сегментировать данные на уровне соединения, а также собирать и (при необходимости) передавать повторно с теми же порядковыми номерами для соединения но с другими номерами на уровне субпотока.

Другим вариантом является использование общего пространства номеров на уровне соединения, которые передаются в разных субпотоках. Здесь возникают две проблемы. Во-первых, отдельные субпотоки будут появляться в сети как сессии TCP с пропусками порядковых номеров, что может нарушить работу некоторых промежуточных устройств (например, систем обнаружения вторжений) и будет противоречить целям совместимости с сетью. Во-вторых, отправитель не сможет связать потерю или доставку пакетов с нужным путем при передаче одного сегмента по разным путям (т. е. не сможет управлять повторной передачей).

Отправитель должен быть способен сказать получателю, как собрать данные для доставки приложению. Для сборки получатель должен определить, как данные из субпотока (переносящие свои номера) отображаются на уровень соединения. Это называется «отображением последовательности данных» (data sequence mapping). Такое отображение можно представить кортежем (порядковый номер данных, порядковый номер в субпотоке, размер), т. е. для данного числа байтов (размер) порядковые номера в субпотоке, начиная с указанного номера, отображаются на порядковые номера в соединении (начиная с указанного). Эта информация предположительно может приходить из разных источников.

Одним из вариантов отображения является использование полей в сегменте TCP (порядковый номер в субпотоке, размер) и добавлять в каждый сегмент порядковый номер в соединении, например, как опцию TCP. Однако это может создать проблемы на промежуточных устройствах, которые собирают или заново сегментируют данные, поскольку для них не задано определенного поведения при объединении опций TCP. Указание порядкового номера в соединении и размера не решает проблемы, поскольку собирающее сегменты промежуточное устройство, столкнувшись с непонятной опцией MPTCP не сможет правильно переписать опции.

В По причине наличия отмеченных проблем выбранное для протокола MPTCP решение состоит в том, что всякий раз при возникновении необходимости отправки отображения данных субпотока другому хосту protocol должны передаваться все 3 части данных (data seq, subflow seq, length). Для сокращения издержек можно отправлять отображения периодически, включая в них более одного сегмента. Нужны дополнительные эксперименты для поиска компромиссов, связанных с частотой передачи отображений. Ее можно исключить полностью в соединении, где пока не используется более одного субпотока и пространства номеров соединения и субпотока совпадают.

5.2. Надежность и повторы передачи

MPTCP обеспечивает подтверждения на уровне соединения и отдельных субпотоков, что обеспечивает устойчивое к отказам обслуживание приложения.

В нормальных условиях MPTCP может использовать отображение номеров данных и ACK субпотоков для информирования о получении сегмента на уровне соединеия. Передача TCP ACK для субпотоков полностью обрабатывается на уровне субпотока для сохранения семантики TCP и управления повторной передачей на уровне субпотока. Это оказывает определенное влияние на сквозную семантику, означая, что подтверждение доставки сегмента (ACK) на уровне потока не позволяет отбросить этот сегмент в буфере упорядочивания на уровне соединения. Кроме того, в отличие от стандартного TCP, получатель не может просто отбросить нарушающие порядок сегменты (например, при нехватке памяти). В некоторых случаях может оказаться желательным отбрасывание сегмента после подтверждения в субпотоке, но до доставки приложению и этому может способствовать подтверждение на уровне соединения.

Кроме того, можно представить случаи, когда подтверждения на уровне соединения будут повышать отказоустойчивость. Рассмотрим субпоток, проходящий через прозрачный прокси - если в прокси возникает отказ после отправки ACK, отправитель не будет повторять потерянный сегмент в другом субпотоке, считая, что сегмент получен. В результате соединение прерывается, несмотря на наличие других работающих субпотоков, и отправитель не сможет определить причину проблемы. Примером ситуации, где это может случиться, является перемещение между точками беспроводного доступа, на каждой из которых работает прокси транспортного уровня. Наконец, в качестве оптимизации может оказаться возможной передача подтверждения на уровне соединения по пути с наименьшим временем кругового обхода (Round-Trip Time или RTT), что может снизить требования к буферу передачи (параграф 5.3).

Следовательно для обеспечения полной отказоустойчивости Multipath TCP с учетом указанных ограничений MPTCP, работающий в публичной сети Internet, **должен** поддерживать явные подтверждения на уровне соединения в дополнение к подтверждениям на уровне субпотока. Подтверждения на уровне соединения требуются лишь для сигнализации о перемещении окна приема вперед и эвристика такой сигнализации более подробно рассматривается в спецификации протокола [5].

В части повторной передачи для сегментов **должна** обеспечиваться возможность повтора в другом субпотоке (не том, где он был передан изначально). Это одна из основных целей MPTCP для поддержки целостности при наличии временных или сохраняющихся отказов субпотоков, реализуемая за счет поддержки двух разных порядковых номеров.

Планирование повторной передачи оказывает значительное влияние на работу пользователей MPTCP. Текущая спецификация MPTCP предполагает, что оставшиеся необработанными данные субпотока при возникновении тайм-аута следует запланировать для передачи через другой субпоток. Это нацелено на минимизацию повреждений при отказе пути и использует в качестве индикатора первый тайм-аут. Более осторожные версии могут использовать второй или третий тайм-аут для одного и того же сегмента.

Обычно ускоренный повтор передачи в отдельном субпотоке не будет вызывать повтора в другом субпотоке, но в некоторых случаях это может оказаться желательным, например, для снижения требований к приемному буферу. Однако во всех случаях повтора передачи в другом субпотоке потерянные сегменты **следует** продолжать отправлять по пути, где они были потеряны. В настоящее время считается, что это нужно для сохранения целостности, как часть цели совместимости с сетью. При этом снижается эффективность и стратегия повторов пока не ясна.

Поэтому нужны масштабные эксперименты для определения подходящей стратегии повторов и после этого рекомендации могут быть уточнены.

5.3. Буферы

Для обеспечения упорядоченной доставки протокол MPTCP должен использовать приемный буфер на уровне соединения, где сегменты хранятся до тех пор, пока не наступит очередь их передачи приложению.

В TCP с одним путем обычно рекомендуется устанавливать размер приемного буфера 2*BDP (произведение пропускной способности и задержки, т. е. BDP = BW*RTT, где BW - пропускная способность, а RTT - время кругового обхода). Один размер BDP позволяет восстановить порядок, нарушенный в сети, другой BDP позволяет сохранять соединение при ускоренном повторе передачи, когда от получателя требуется возможность сохранять данные в течение дополнительного интервала RTT.

Для MPTCP хранение усложняется. Конечной целью является устранения влияния потери пакетов в субпотоке или отказа субпотока на другие субпотоки. Получателю следует иметь буфер, достаточный для размещения данных, пока потерянный сегмент не будет передан повторно и принят получателем.

Худшим случаем является тайм-аут субпотока с наибольшим отношением RTT/RTO (время кругового обхода к тайм-ауту повторной передачи), когда получатель буферизует данные из всех субпотоков за интервал RTO. Таким образом, минимальный размер приемного буфера для соединения, который позволит избежать остановки при отказе субпотока, составляет sum(BW_i)*RTO_max, где BW_i = пропускная способность каждого субпотока i, а RTO_max - наибольшее значение RTO среди субпотоков.

Это на порядок больше размера буфера для одного потока и, возможно, слишком много для практических целей. Более разумным требованием будет предотвращение остановки в отсутствие тайм-аутов. Поэтому **рекомендуется** приемный буфер размером 2*sum(BW_i)*RTT_max, где RTT_max - наибольшее значение RTO среди субпотоков. Такой размер приемного буфера предотврати остановку субпотоков при использовании ускоренного повтора любым из них.

Результирующий размер буфера должен быть достаточно небольшим для практического применения. Однако могут возникать экстремальные ситуации, когда быстрый путь с высокой пропускной способностью (например, 100 Мбит/с с RTT 10 мсек.) применяется вместе с медленными путями (например, 1 Мбтит/с и RTT 1000 мсек.). В таком случае требуемый размер буфера составит 12,5 Мбайт, что явно слишком много. В подобных случаях может оказаться разумным использовать в соединении MPTCP лишь самые быстрые пути, оставляя медленные в качестве резерва.

Для буфера передачи **рекомендуется** такой же размер, как для приемного буфера, т. е. 2*sum(BW_i)*RTT_max. Это обусловлено тем, что отправитель локально хранить переданные, но не подтвержденные на уровне соединения сегменты. Размер буфера передачи особенно важен для хостов, поддерживающих много исходящих соединений. Если требуемый размер слишком велик, хосту следует ограничить передачу данных только быстрыми субпотоками, оставляя другие на случай отказа.

5.4. Сигнализация

Поскольку MPTCP использует TCP в качестве транспортного механизма субпотоков, соединение MPTCP начинается как одиночное соединение TCP. Тем не менее, нужно уведомить партнера о поддержке MPTCP и желании использовать протокол в соединении. Для передачи этих сведений применяется опция TCP, указывающая дополнительную функциональность сессии TCP. Кроме того, нужна дополнительная сигнализация в процессе работы сессии MPTCP для управления сборкой во множестве субпотоков и информирования другой стороны о доступных адресах IP.

Протокол MPTCP использует для дополнительной сигнализации опции TCP, как наиболее подходящий механизм с учетом целей, указанных в разделе 2. С помощью этого механизма требуемые для работы MPTCP сигналы доставляются отдельно от данных, позволяя создавать и обрабатывать их независимо от потока данных и сохраняя архитектурную совместимость с элементами сети.

Это решение является соглашением рабочей группы (после подробного обсуждения на IETF78), основные мотивы которого приведены ниже.

- Опции TCP являются традиционным методом сигнализации для TCP.
- Опции TCP в SYN обеспечивают конечным хостам наиболее совместимый способ указать поддержку MPTCP.
- При передаче ACK для соединения в данных (payload) они могут пропадать при потере пакетов или подвергаться влиянию контроля перегрузок, что может влиять на пропускную способность данных от отправителя и приводить к блокировке head-of-line.
- Промежуточные устройства (например, трансляторы NAT) легко разбирают опции TCP (скажем, меняют адреса).

С другой стороны, основными недостатками опций TCP по сравнению с кодированием TLV в данных являются:

- ограниченное пространство опций для сигнальных сообщений;
- возможность отбрасывания промежуточными устройствами пакетов с незнакомыми опциями;
- доставка данных управления может оказаться ненадежной.

При разработке MPTCP эти проблемы устранялись по мере возможности за счет аккуратного учета размера опций MPTCP и плавного возврата к обычному TCP при потере данных управления.

Оба варианта сигнализации могут конфликтовать с выгрузкой обработки TCP в быстрые интерфейсные платы, например для сегментации, контрольных сумм и сборки. Для сетевых плат, поддерживающих MPTCP, сигнализация в опциях TCP должна упростить их выгрузку за счет раздельной обработки сигнализации и данных в MPTCP.

5.5. Управление путями

В настоящее время сеть не поддерживает разные пути между парой адресов IP. Для поддержки разделения путей в современных сетях IP как типичный случай MPTCP использует множество адресов на одном или обоих взаимодействующих хостах. Предполагается, что эти пути, которые не обязательно различаются полностью, будут достаточно разными, чтобы позволить рост пропускной способности и отказоустойчивости за счет использования нескольких путей. Использование множества адресов IP является простейшим вариантом, не требующих дополнительных свойств сети.

Таким образом, в большинстве случаев будет использоваться множество адресных пар (отправитель, получатель) в качестве селекторов путей. Однако каждый путь будет указываться стандартным квинтетом, включающим адреса и номера портов отправителя и получателя, а также номер протокола, что позволяет MPTCP использовать в качестве селекторов пути адреса и номера портов. Это позволяет хостам распределять нагрузку в MPTCP по номерам портов, например, при их маршрутизации по разным путям (что может обеспечиваться такими технологиями, как ECMP[1] [4]). Однако следует отметить, что ISP часто используют организацию трафика для оптимизации ресурсов в своих сетях, поэтому следует проявлять осторожность (как разработчикам, так и ISP), чтобы в MPTCP не применялись практически совпадающие пути.

Для повышения шансов успешной организации дополнительных субпотоков (например, при размещении одной стороны за МСЭ, NAT или иным промежуточным устройством с ограничениями), каждому из хостов **следует** быть способным добавлять новые субпотоки в соединение MPTCP. Протокол MPTCP **должен** быть способен обслуживать пути, которые могут появляться и исчезать в процессе работы соединения (например, в результате активации дополнительного сетевого интерфейса).

Поддержка путей отделена от планирования пакетов, интерфейса субпотоков и контроля перегрузок в MPTCP, как указано в разделе a4. Поэтому будет целесообразно заменить схему, основанную на адресах IP, другим механизмом выбора путей без существенного влияния на другие функциональные компоненты.

[1] Equal-Cost Multipath - множество равноценных путей.

5.6. Идентификация соединений

Поскольку соединение MPTCP невозможно связать с традиционным квинтетом из адресов и номеров портов у отправителя и получателя, а также номером протокола на весь срок существования соединения, желательно обеспечит иной механизм идентификации соединений. Для осведомленных о MPTCP приложений и реализаций MPTCP (а также знающих о MPTCP промежуточных устройств) будет полезно иметь уникальный идентификатор, связанный с множеством субпотоков.

Поэтому для каждого соединения MPTCP нужен идентификатор на каждом из хостов, уникальный в рамках этого хоста. Во многих случаях это напоминает использование эфемерных номеров порта в обычном TCP. Заявление и назначение такого идентификатора выходит за рамки документа.

Однако не знающие о MPTCP приложения не будут иметь доступа к такому идентификатору и в таких случаях соединение MPTCP будут указываться обычным квинтетом первого субпотока TCP. Задание поведения реализации MPTCP при отказе этого субпотока выходит за рамки документа. При наличии не понимающих MPTCP приложений, делающих предположения о продолжении работы исходной пары адресов поведение может быть нарушено. Предполагается, что влияние этого будет незначительным и, возможно, пренебрежимым. MPTCP **недопустимо** использовать в приложениях, запрашивающих привязку к конкретному адресу или интерфейсу, поскольку такая привязка задает осознанный выбор используемого пути.

Поскольку требования приложений на данном этапе не очевидны, нет возможности указать сейчас влияние потери исходной пары адресов на работу приложения. Это поведение должно определяться зависимым от реализации решением и предполагается, что такое решение будут принимать разработчики после проведения дополнительных исследований.

5.7. Контроль перегрузок

Как отмечено в требования к совместимости на сетевом уровне параграфа 2.2.3, для алгоритма контроля перегрузок в MPTCP имеется 3 цели - повышение пропускной способности (не хуже одиночного соединения TCP), отсутствие помех для других пользователей сети (использование на любом пути не больше пропускной способности, чем получит отдельный поток на этом маршруте, что особенно важно для общих «пробочных» участков) и балансирование загрузки путем переноса трафика с наиболее насыщенных путей. Для достижения этих целей алгоритмы контроля перегрузок каждого потока должны быть связаны между собой. Подходящие алгоритмы предложены в [7].

5.8. Безопасность

Подробный анализ угроз для Multipath TCP представлен в [12]. Этот документ рассматривает лавинные рассылки и атаки с захватом, которые могут быть использованы против соединений Multipath TCP.

Основной целью защиты Multipath TCP, как указано в параграфе 2.3, является обеспечение безопасности не хуже TCP.

С учетом этой цели и анализа угроз можно выделить 3 основных требования безопасности, которые **следует** выполнять многоадресной реализации Multipath TCP.

- Обеспечить механизм контроля соответствия сторон согласования субпотока участникам исходного соединения (например, требование обмена ключами при начальном согласовании субпотока для ограничения возможностей захвата субпотоков).
- Проверка возможности получения партнером трафика по новому адресу до его добавления (т. е. проверка принадлежности адреса в тому же хосту для предотвращения лавинных атак).
- Обеспечение защиты от повторного использования (т. е. проверка «свежести» запросов на добавление и удаление субпотоков).

Были развернуты дополнительные механизма (как часть стандартных стеков TCP) для обеспечения устойчивости к DoS-атакам[1]. Например, имеются различные механизмы для защиты от атак со сбросом TCP [18] и Multipath TCP следует поддерживать такую защиту. Разработан механизм TCP SYN Cookie [19], позволяющий серверу TCP отложить создание соединения в состоянии SYN_RCVD и оставаться без состояния до перехода в ESTABLISHED. Multipath TCP следует в идеальном случае поддерживать такую функциональность или хотя бы избегать значительных вычислений до перехода в состояние ESTABLISHED (для всего соединения Multipath TCP).

Следует отметить, что аспекты пространства разработки Multipath TCP вносят ограничения для защитных решений:

- использование опций TCP существенно ограничивает передаваемую при согласовании информацию;
- необходимость работы через промежуточные устройства требует обработки изменяемых пакетов;
- желание поддержать подходы break-before-make[2] и make-before-break[3] при добавлении субпотоков (в ограниченный срок) предполагает, что хост не может полагаться на использование уже имеющегося субпотока для поддержки добавления нового.

Протокол MPTCP будет разрабатываться с учетом требований безопасности и спецификация [5] будет документировать их выполнение.

6. Программные взаимодействия

6.1. Взаимодействие с приложениями

Для приложений, использующих вызовы имеющегося API с целью привязки к конкретному адресу или интерфейсу, использовать расширение MPTCP **недопустимо**. Это обусловлено тем, что приложение явно задает выбор пути и ожидает соответствующего поведения для поддержки совместимости.

[1]Denial-of-Service - отказ в обслуживании.
[2]Прервать до создания.
[3]Создать до прерывания.

Взаимодействия с приложениями представлены в [8], включая ожидаемое изменение производительности и новые функции, которые могут быть запрошены через расширенный интерфейс API.

TCP имеет возможность отправки срочных (Urgent) данных, доставка которых приложению может (не обязательно) обеспечиваться по отдельному каналу (out-of-band). Использовать эту возможность не рекомендуется по причине ее влияния на безопасность и различий в реализациях [20]. MPTCP требует непрерывных данных для поддержки отображения их последовательности на множество сегментов, поэтому указатель Urgent не может прерывать имеющееся отображение. Реализация MPTCP **может** поддерживать отправку срочных данных и в этом случае ей **следует** отправлять такие данные в ближайшем доступном не выделенном пространстве номеров субпотока. Входящие срочные данные **следует** отображать на номера в соединении и доставлять приложению как в TCP.

6.2. Взаимодействие с системами управления

Для обеспечения взаимодействия между TCP и системами управления сетью были определены базы MIB TCP [21] и TCP Extended Statistics (ESTATS) [22]. MPTCP следует использовать эти MIB для аспектов, которые должны быть прозрачны для приложений.

Предполагается определение в будущем MPTCP MIB по мере обретения опыта развертывания MPTCP. Эта база обеспечит доступ к специфическим свойствам MPTCP, таким как включение MPTCP, а также число и свойства используемых путей.

7. Взаимодействие с промежуточными устройствами

Как отмечено в параграфе 2.2, целью MPTCP является возможность развертывания сейчас и это требует совместимости с большинством промежуточных устройств. В этом параграфе рассматриваются проблемы, которые могут быть связаны с NAT, МСЭ, прокси, системами детектирования вторжений и другими промежуточными устройствами, которые нужно учесть при разработке протокола во избежание помех развертыванию.

Этот параграф нацелен лишь на описание вариантов и соображений, а зависящие от протоколов решения проблем будут рассмотрены в сопутствующих документах.

Multipath TCP будет развертываться в сети, которая уже не просто обеспечивает доставку дейтаграмм. В сети имеется огромное число промежуточных устройств для оптимизации и смягчения различных проблем, связанных с протоколами Internet - трансляторы NAT решают проблему нехватки адресов IP [15], прокси для повышения производительности (Performance Enhancing Proxy или PEP) оптимизируют TCP для различных каналов [17], МСЭ [16] и системы детектирования вторжений пытаются блокировать вредоносное содержимое от попадания на хосты, а нормализаторы трафика [23] обеспечивают согласованное представление потоков трафика в системах детектирования вторжений (Intrusion Detection System или IDS) и на хостах.

Все эти промежуточные устройства оптимизируют имеющиеся приложения за счет будущих. Фактически от будущих приложений часто требуется поведение, похожее не имеющееся для повышения шансов успешного развертывания. Кроме того, точное поведение промежуточных устройств не задано, а ошибки реализации усугубляют ситуацию, дополнительно осложняя развертывание новых технологий.

Ниже перечислены классы промежуточных устройств с указанием поведения, которое может влиять на применение MPTCP. Этот список применяется в [5] для описания свойств протокола MPTCP, служащих для снижения влияния промежуточных устройств.

NAT (Network Address Translator) - транслятор сетевых адресов
: Отвязывает локальный IP-адрес хоста (в случае NAPT и номер порта) от адреса, появляющегося в Internet при прохождении пакетов через NAT. Это усложняет работу и снижает шансы на успех при передаче адресов IP.

PEP (Performance Enhancing Proxy) - прокси для повышения производительности
: Служат для повышения производительности протоколов при работе по каналам с низкой производительностью (например, с высокой задержкой или большими потерями). В результате соединения TCP могут «расщепляться» с возникновением таких явлений, как упреждающие ACK, что не гарантирует прямого взаимодействия между хостами. PEP, МСЭ и другие промежуточные устройства могут также изменять объявленный размер окна приема.

Traffic Normalizer - нормализатор трафика
: Служат для предотвращения неоднозначности и возможных атак на сетевом уровне, а также вряд ли позволят пропуски в порядковых номерах TCP (что влияет на повторную передачу MPTCP и нумерацию в субпотоках).

Firewall - межсетевой экран
: Помимо предотвращения входящих соединений МСЭ могут пытаться обеспечить дополнительную защиту, такую как случайные порядковые номера (в результате отправитель не сможет точно узнать порядковый номер TCP у получателя).

IDS (Intrusion Detection System) - система детектирования вторжений
: Могут искать в трафике определенные шаблоны в целях защиты сети и давать ложные срабатывания для MPTCP, отбрасывая соединения при обычной работе. В будущих промежуточных устройствах, знающих о MPTCP, потребуется сопоставление используемых путей.

МСЭ с проверкой содержимого
: Некоторые промежуточные устройства могут менять содержимое пакетов, включая переписывание URI в HTTP.

Кроме того, все классы промежуточных устройств могут влиять на указанные ниже аспекты трафика TCP.

Опции TCP
: Некоторые промежуточные устройства могут отбрасывать пакеты с неизвестными опциями TCP или вырезать такие опции из пакетов.

Сегментация и объединение
: Промежуточные устройства (иной раз достаточно близкие к конечным хостам, как в случае TSO (TCP Segmentation Offloading) на сетевых адаптерах (Network Interface Card или NIC) могут менять границы пакетов по сравнению с заданными отправителем. Это может выполняться путем расщепления или объединения пакетов. В результате возникает два основных эффекта - границы пакетов не могут гарантироваться и поведение промежуточных устройств применительно к опциям TCP в таких случаях нельзя предсказать точно (пакеты могут повторяться, отбрасываться или передаваться лишь однократно).

8. Участники работы

Авторы признательны Andrew McDonald и Bryan Ford за их вклад в документ.

Авторы также благодарны за детальное рецензирование Olivier Bonaventure, Gorry Fairhurst, Iljitsch van Beijnum, Philip Eardley, Michael Scharf, Lars Eggert, Cullen Jennings, Joel Halpern, Juergen Quittek, Alexey Melnikov, David Harrington, Jari Arkko и Stewart Bryant.

9. Благодарности

Alan Ford, Costin Raiciu, Mark Handley и Sebastien Barre поддерживались в рамках исследовательского проекта Trilogy (http://www.trilogy-project.org, ICT-216372), частично финансируемого Европейской комиссией в рамках программы Seventh Framework. Выраженные здесь мнения принадлежат лишь авторам и Европейская комиссия не несет ответственности за любое использование представленной в документе информации.

10. Вопросы безопасности

Этот информационный документ содержит обзор архитектуры Multipath TCP и сам по себе не вызывает проблем безопасности. В отдельном документе [12] приведен анализ угроз, которые могут возникать при использовании Multipath TCP. Однако к протоколу, основанному на описанной в документе архитектуре, будет предъявляться много требований безопасности. Высокоуровневые цели для такого протокола указаны в параграфе 2.3, а параграф 5.8 содержит более подробное рассмотрение требований безопасности и проектных решений, применимых при разработке протокола MPTCP [5].

11. Литература

11.1. Нормативные документы

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

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

11.2. Дополнительная литература

[3] Wischik, D., Handley, M., and M. Bagnulo Braun, "The Resource Pooling Principle", ACM SIGCOMM CCR vol. 38 num. 5, pp. 47-52, October 2008, <http://ccr.sigcomm.org/online/files/p47-handleyA4.pdf>.

[4] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, November 2000.

[5] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", Work in Progress[1], March 2011.

[6] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[7] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", Work in Progress[2], March 2011.

[8] Scharf, M. and A. Ford, "MPTCP Application Interface Considerations", Work in Progress[3], March 2011.

[9] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[10] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.

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

[12] Bagnulo, M., "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6181, March 2011.

[13] Becke, M., Dreibholz, T., Iyengar, J., Natarajan, P., and M. Tuexen, "Load Sharing for the Stream Control Transmission Protocol (SCTP)", Work in Progress, December 2010.

[14] Ford, B. and J. Iyengar, "Breaking Up the Transport Logjam", ACM HotNets, October 2008.

[15] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[16] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[17] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, June 2001.

[18] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", RFC 5961, August 2010.

[19] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, August 2007.

[20] Gont, F. and A. Yourtchenko, "On the Implementation of the TCP Urgent Mechanism", RFC 6093, January 2011.

[21] Raghunarayan, R., "Management Information Base for the Transmission Control Protocol (TCP)", RFC 4022, March 2005.

[22] Mathis, M., Heffner, J., and R. Raghunarayan, "TCP Extended Statistics MIB", RFC 4898, May 2007.

[23] Handley, M., Paxson, V., and C. Kreibich, "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics", Usenix Security 2001, 2001, <http://www.usenix.org/events/sec01/full_papers/handley/handley.pdf>.

[1] Работа опубликована в RFC 6824, замененном RFC 8684. *Прим. перев.*
[2] Работа опубликована в RFC 6356. *Прим. перев.*
[3] Работа опубликована в RFC 6897. *Прим. перев.*

Адреса авторов

Alan Ford
Roke Manor Research
Old Salisbury Lane
Romsey, Hampshire SO51 0ZN
UK
Phone: +44 1794 833 465
EMail: alan.ford@roke.co.uk

Costin Raiciu
University College London
Gower Street
London WC1E 6BT
UK
EMail: c.raiciu@cs.ucl.ac.uk

Mark Handley
University College London
Gower Street
London WC1E 6BT
UK
EMail: m.handley@cs.ucl.ac.uk

Sebastien Barre
Universite catholique de Louvain
Pl. Ste Barbe, 2
Louvain-la-Neuve 1348
Belgium
Phone: +32 10 47 91 03
EMail: sebastien.barre@uclouvain.be

Janardhan Iyengar
Franklin and Marshall College
Mathematics and Computer Science
PO Box 3003
Lancaster, PA 17604-3003
USA
Phone: 717-358-4774
EMail: jiyengar@fandm.edu

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru