Internet Engineering Task Force (IETF)
Request for Comments: 9330
Category: Informational
ISSN: 2070-1721

B. Briscoe, Ed.
Independent
K. De Schepper
Nokia Bell Labs
M. Bagnulo
Universidad Carlos III de Madrid
G. White
CableLabs

January 2023

Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture

Архитектура службы Internet L4S

Аннотация

Этот документ описывает архитектуру L4S с малыми задержками и потерями в сочетании с расширяемым управлениям пропускной способностью. Основой L4S является осознание того. Что основной причиной задержки в очередях являются контроллеры перегрузки у отправителей, а не сами очереди. Архитектура L4S позволяет (но не требует) всем приложениям Internet отказаться от алгоритмов контроля перегрузок, которые ведут к существенным задержкам в очередях, и применять вместо них новый класс средств контроля перегрузок, способных определять пропускную способность с очень малыми задержками в очередях. Это достигается за счёт изменения явных уведомлений о перегрузке (Explicit Congestion Notification или ECN) из сети. Новая архитектура обеспечивает приложениям малые задержки и высокую пропускную способность.

Архитектура сосредоточена в основном на поэтапном развёртывании и задаёт механизм, обеспечивающий сосуществование новых механизмов контроля перегрузок L4S с «классическим» контролем в общей сети. Цель заключается в обеспечении с помощью L4S задержки и пропускной способности существенно лучших (редко, худших) обычно без влияния на работу «классических» механизмов.

Статус документа

Документ не относится к категории Internet Standards Track и публикуется с информационными целями.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Не все документы, одобренные IESG, претендуют на статус стандартов Internet, см. раздел 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc9330.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Трафик из узких каналов (например, домашний доступ в Internet или Wi-Fi) все чаща приходит из приложений, которые предпочитают малые задержки - интерактивные web-приложения и службы, голосовая связь, видео со звуком и интерактивные видеосистемы, интерактивное удалённое присутствие, мгновенный обмен сообщениями, сетевые и облачные игры, удалённые рабочие столы, облачные приложения и дополненная реальность, а также дистанционное управления промышленным оборудованием и процессами с визуальным сопровождением. За последнее десятилетие или около того было много сделано для сокращения задержки путём кэширования и переноса серверов ближе к пользователям. Однако очереди остаются основным, хотя и меняющимся компонентом задержки. Например, всплески задержки до сотен миллисекунд не являются редкостью даже при использовании современного активного управления очередями (Active Queue Management или AQM) [COBALT] [DOCSIS3AQM]. Классический механизм AQM в узких местах сетей доступа обычно настраивается для буферизации пилообразности (sawteeth) отдельных потоков, что может приводить примерно к удвоению задержки во время продолжительных потоков по сравнению с ожидаемой задержкой для незагруженного пути [BufferSize]. Малые потери тоже важны, поскольку для интерактивных приложений потери транслируются в дополнительные задержки, связанные с повтором передачи.

Было показано, что при достижении в сетях доступа скоростей, распространённых сегодня в развитых странах, повышение пропускной способности канала ведёт к снижению скорости отдачи, если проблема задержки не решена. [Dukkipati06] [Rajiullah15]. Поэтому целью является служба Internet с очень малыми задержками в очередях и потерями, а также расширяемой пропускной способностью. Очень малыми считаются задержки меньше 1 мсек в среднем и меньше 2 мсек при 99-м процентиле. Сквозная задержка выше 50 мсек [Raaen14] или даже 20 мсек [NASA04] начинает казаться неестественной для требовательных интерактивных приложений. Поэтому устранение ненужной изменчивости задержек увеличивает зону охвата таких приложений (дальность, при которой использование остаётся комфортным) и/или обеспечивает дополнительный запас времени, который можно использовать для улучшения обработки. Этот документ описывает архитектуру L4S для достижения указанных целей.

Дифференцированное обслуживание (Differentiated service или Diffserv) предлагает ускоренную пересылку (Expedited Forwarding или EF) [RFC3246] для некоторых пакетов за счёт других, но даёт эффекта, когда весь (или большая часть) трафик в узких местах требует малой задержки. L4S хорошо работает в таких случаях, когда весь трафик относится к L4S, поскольку служба «отдаёт не забирая», не требует настройки и управления (регулирование трафика или контракты), связанных с предпочтением одних потоков перед другими.

Задержка в очередях время от времени ведёт к снижению производительности [Hohlfeld14]. Это происходит i) при наличии потока с достаточно большой потребностью в пропускной способности (например, TCP) наряду с другим пользовательским трафиком в узком канале или ii) когда само приложение с малой задержкой запрашивает высокую пропускную способность или адаптивное управление скоростью (например, интерактивное видео). В настоящее время повышенид производительности за счёт L4S должно быть достаточным мотивом для внедрения операторами сетей.

AQM служит частью решения для очередей под нагрузкой, повышая производительность всего трафика, но этому методу присущи ограничения в части сокращения задержки в очереди при изменении сети без устранения причин проблемы.

Первопричиной проблемы является присутствие стандартного контроля перегрузок (Reno [RFC5681]) или совместимых с ним вариантов (например, CUBIC [RFC8312]) в TCP и другом транспорте, таком как QUIC [RFC9000]. Далее в документе похожие на Reno механизмы контроля перегрузок называются «классическими». Эти механизмы вызывают сравнительно большие пилообразные колебания заполнения очередей. Если оператор наивно пытается сократить задержки в очереди путём настройки AQM для работы с уменьшенной очередью, классический контроль перегрузок приведёт к значительному недоиспользованию канала в ниженей части каждого зубца пилы. Продолжительность зубьев увеличивается по мере роста скорости потока (см. параграф 5.1 и [RFC3649]).

Было показано, что при замене на передающем узле классического контроля перегрузок расширяемым (Scalable) производительность упомянутых выше приложений под нагрузкой может существенно возрастать после внедрения в сети подходящего механизма AQM. В приведённом ниже примере решения с использованием ЦОД TCP (Data Center TCP или DCTCP) [RFC8257] Dual-Queue Coupled AQM [RFC9332] на канале DSL или Ethernet задержка в очереди при значительной нагрузке составляет 1-2 мсек при 99-м процентиле без потери полезной загрузки (utilization) [L4Seval22] [DualPI2Linux] (для других типов каналов ситуация описана в параграфе 6.3). Это сопоставимо со средней задержкой 5-20 мсек для классического контроля перегрузки и современных AQM, таких как Flow Queue CoDel [RFC8290], Proportional Integral controller Enhanced (PIE) [RFC8033], DOCSIS PIE [RFC8034] и 20-30 мсек при 99-м процентиле [DualPI2Linux].

Архитектура L4S предназначена для поэтапного внедрения. Можно развернуть службу L4S в узком месте одновременно с имеющимися службами best efforts [DualPI2Linux], чтобы неизмененные приложения могли начать использование службы сразу же по обновлении сетевого стека отправителя. В сетях доступа обычно узким место является одиночный канал для каждого сайт (дом, небольшое предприятие или мобильное устройство), поэтому

развёртывание службы на одной или обеих сторонах такого канала должно обеспечить почти все преимущества в соответствующем направлении. Для TCP [ACCECN] отправитель проверяет предоставление получателем более точных откликов, а для иного транспорта, такого как QUIC [RFC9000] и DCCP[1] [RFC4340] подходят любые получатели.

В этом документе представлена архитектура L4S, состоящая из 3 компонентов: изоляция в сети трафика L4S от классического, свойства протокола, позволяющие элементам сети идентифицировать трафик L4S и поддержка контроля перегрузок L4S на хосте. Протокол определён отдельно в [RFC9331] как экспериментальное изменение явных уведомлений о перегрузке (ECN). Этот документ описывает и обосновывает составные части архитектуры и их взаимодействия для обеспечения малой задержки и потерть, а также расширяемых услуг Internet. Кроме того, в документе подробно рассматривается поэтапное развёртывание, упомянутое здесь.

1.1. Структура документа

Этот документ описывает архитектуру L4S в три приёма. В разделе 2 приведён краткий обзор идеи на высоком уровне и описаны основные компоненты с минимальным обоснованием. Это предназначено лишь для организации контекста определения терминов в разделе 3 и разъяснения структуры оставшейся части документа. Раздел 4 содержит более подробное описание каждого компонента с обоснованиями, но по-прежнему описывает архитектуру, а не способы реализации. В разделе 5 объясняется выбор каждого элемента решения (5.1. Почему эти компоненты выбраны основными?) и отличия от других подходов (5.2. Что L4S добавляет к имеющимся подходам).

После описания архитектуры в разделе 6 рассмотрена её применимость путём описания возможных применений и вариантов использования, которые послужили мотивами для разработки. Рассмотрены проблемы, связанные с применением архитектуры для разных технологий канального уровня, модели поэтапного внедрения (включая 2 основных топологии развёртывания, различные последовательности внедрения и взаимодействие с существующими подходами). Документ завершается обычными заключительными частями, включая подробное рассмотрение вопросов организации трафика и соображений безопасности в разделе 8.

2. Обзор архитектуры L4S

Ниже очерчены три основных компонента архитектуры L4S: 1) расширяемые (Scalable) элементы контроля перегрузок на передающем хосте, 2) механизм AQM в узких местах сети (bottleneck), 3) протокол взаимодействия.

Сначала нужно уяснить главное - малые задержки не обеспечиваются сетью, а являются результатом осторожного поведения контроллеров перегрузки с масштабированием, используемых отправителями L4S. Сеть играет определённую роль в первую очередь для изоляции трафика L4S с малыми задержками от очередей с большой задержкой, требуемых для имеющегося трафика с «классическим» поведением. Сеть также изменяет способ сигнализации транспорту о росте очередей. Сеть использует протокол ECN, но сигнализирует о начале роста очереди незамедлительно без задержки, связанной со сглаживанием, характерной для Classic AQM. Поскольку поддержка ECN важна для L4S, отправители используют поле ECN как протокол, позволяющий сети отличать пакеты L4S от классических.

1) Хост

   Расширяемые элементы контроля перегрузок уже имеются. Они решают задачи масштабирования для классических элементов контроля перегрузок, таких как Reno или CUBIC. Поскольку скорости потоков выросли с момента разработки контроля перегрузок TCP в 1988 г., для достаточно продолжительных потоков на восстановление (в результате маркировки ECN или потери пакетов) затрачиваются сотни круговых обходов (round trip), как показано в примерах параграфа 5.1 и в [RFC3649], и их число продолжает расти. Поэтому контроль за постановкой в очередь и её использованием ослабляется и малейшие нарушения (например, появление новых потоков) препятствуют достижению высокой скорости.

   При расширяемом контроле перегрузок среднее время от одного сигнала перегрузки до следующего (время восстановления) инвариантно к изменению скорости потока при сохранении прочих условий. Это обеспечивает одинаковый уровень контроля за постановкой в очередь и её использованием независимо от скорости потока, а также гарантирует устойчивость к нарушениям при высокой скорости потока. Наиболее распространенным расширяемым контролем перегрузок (в контролируемых средах) является протокол DCTCP [RFC8257], реализованный и развёрнутый в серверных версиях Windows (Server Edition), начиная с 2012 г., а также в Linux и FreeBSD. Несмотря на хорошую работу функций DCTCP в широком диапазоне значений времени кругового обхода (round-trip time или RTT), в большинстве реализацию отсутствуют некоторые функции защиты, которые требуются для работы за пределами контролируемых сред, таких как ЦОД (см. 6.4.3. Поток L4S с узким местом без ECN). Поэтому нужно реализовать расширяемые элементы контроля перегрузок в TCP и других транспортных протоколах (QUIC, SCTP[2], RTP/RTCP, RMCAT[3] и т. п.). За время подготовки и публикации этого документа были реализованы расширяемые элементы контроля перегрузок Prague для TCP и QUIC [PRAGUE-CC] [PragueLinux], вариант L4S контроллера RMCAT SCReAM [SCReAM-L4S] и связанная с L4S ECN часть BBRv2[4] [BBRv2] для транспорта TCP и QUIC.

2) Сеть

   Трафик L4S нужно изолировать от задержки в очередях классического трафика. Одним из путей решения является поддержка одной очереди на поток приложения (FQ), например, FQ-CoDel [RFC8290]. Однако достаточно использовать лишь две очереди и не требовать проверки транспортных заголовков в сети, что возможно не всегда (см. 5.2. Что L4S добавляет к имеющимся подходам). При наличии лишь двух очередей может показаться невозможным узнать, какая пропускная способность нужно запланировать для каждой очереди, без определения числа потоков, использующих каждую очередь в каждый момент. И нежелательно произвольно делить пропускную способность сети доступа на две части. Для решения этой задачи с минимальными сложностями был разработан механизм Dual-Queue Coupled AQM. Он действует как

[1]Datagram Congestion Control Protocol - протокол дейтаграмм с контролем перегрузок.
[2]Stream Control Transmission Protocol - протокол управления потоковой передачей.
[3]RTP Media Congestion Avoidance Techniques - методы предотвращения перегрузки сред RTP.
[4]Bottleneck Bandwidth and Round-trip propagation time - пропускная способность узких мест и время кругового обхода.

полупроницаемая мембрана, которая разделяет задержку, но не пропускную способность. Таким образом, две очереди предназначены для перехода от классического поведения к L4S, а не для приоритизации полосы.

В разделе 4 дано высокоуровневое объяснение работы вариантов FQ и DualQ для L4S, а в [RFC9332] дано полное объяснение модели DualQ Coupled AQM. Конкретный алгоритм маркировки не задан для L4S AQM и в приложениях к [RFC9332] приведены ненормативные примеры, которые были реализованы и оценены, а также даны рекомендации для принятых по умолчанию значений параметров. Предполагается, что эксперименты с L4S улучшат понимание настройки параметров и выбора алгоритмов маркировки.

3) Протокол

Передающий хост должен отмечать пакеты L4S и классические разными идентификаторами, чтобы сеть могла классифицировать их для разной обработки. В спецификации идентификатора L4S [RFC9331] сделано заключение, что все варианты требуют компромиссов, но коды ECT(1) и CE[1] в поле ECN представляют работоспособное решение. Как уже отмечено, сеть также использует ECN для незамедлительной сигнализации транспорту о самом начале роста очереди.

3. Терминология

Classic Congestion Control - классический контроль перегрузок

Поведение контроля перегрузок, способное сосуществовать со стандартным Reno [RFC5681] без существенного негативного влияния на скорость потока [RFC5033]. Проблема расширяемости классического контроля перегрузок рассмотрена на примерах в параграфе 5.1. Почему эти компоненты выбраны основными? и [RFC3649].

Scalable Congestion Control - расширяемый контроль перегрузок

Контроль перегрузок, где среднее время от одного сигнала насыщения до следующего (время восстановления) не зависит от скорости потока при прочих равных условиях. Например, DCTCP усредняет 2 сигнала пересылки за интервал кругового обхода, независимо от скорости потока, как и другие недавно разработанные расширяемые механизмы контроля перегрузок, такие как Relentless TCP [RELENTLESS], Prague для TCP и QUIC [PRAGUE-CC] [PragueLinux], BBRv2 [BBRv2] [BBR-CC], L4S-вариант SCReAM для потоков в реальном масштабе времени [SCReAM-L4S] [RFC8298]. Дополнительные сведения можно найти в параграфе 4.3 [RFC9331].

Classic Service - классический сервис

Классический сервис предназначен для всех вариантов поведения контроля перегрузок, сосуществующих с Reno [RFC5681] (например, сам Reno, CUBIC [RFC8312], Compound [CTCP], TFRC [RFC5348]). «Классической очередью» называется очередь, обеспечивающая классический сервис.

Low Latency, Low Loss, and Scalable throughput (L4S) service - сервис с малыми задержками и потерями, а также расширяемой пропускной способностью

Сервис L4S предназначен для трафика с расширяемыми алгоритмами контроля перегрузок, такими как Prague [PRAGUE-CC], который был выведен из DCTCP [RFC8257]. Сервис L4S предназначен для более широкого класса трафика, нежели просто Prague, и позволяет развивать набор средств контроля перегрузок, аналогичных Prague, таких как отмечены выше (Relentless, SCReAM и т. п.). Очередью L4S называется очередь, обеспечивающая услуги L4S.

Атрибуты Classic и L4S могут применяться к очередям (queue), кодам (codepoint), идентификаторам (identifier), классификации (classification), пакетам (packet), потокам (flow). Например термин «пакет L4S» относится к пакету с идентификатором L4S, переданному из системы контроля перегрузок L4S.

Оба типа сервиса (Classic и L4S) могут справляться с некоторой долей неотзывчивого и слабо отзывающегося трафика, но в случае L4S скорость должна быть достаточно плавной или низкой, чтобы не возникала очередь (например, DNS, VoIP, синхронизация игр и т. п.).

Reno-friendly - совместимость с Reno

Часть классического трафика, совместимая со стандартным контролем перегрузок Reno, заданным для TCP в [RFC5681]. Спецификация TFRC [RFC5348] косвенно подразумевает, что дружественностью (совместимостью) считается «нахождение обычно в пределах двухкратного отличия скорости передачи потока TCP при одинаковых условиях». Термин Reno-friendly используется здесь вместо TCP-friendly, поскольку последнее выражение стало неточным, так как протокол TCP сейчас использует много разных вариантов контроля перегрузок, а Reno применяется не только в TCP, но и в других транспортных протоколах (например, QUIC [RFC9000]).

Classic ECN - классический механизм явных уведомлений о перегрузке

Исходный механизм явных уведомлений о перегрузке (ECN) [RFC3168] требует считать сигналы ECN эквивалентом отбрасывания пакетов как при генерации в сети, так и отвечающим хостом.

Для L4S имена кодов 2-битового поля IP-ECN не отличаются от заданных в спецификации ECN [RFC3168] - Not-ECT, ECT(0), ECT(1), CE, где ECT указывает поддержку ECN в транспорте (ECN-Capable Transport), а CE - возникновение перегрузки (Congestion Experienced). Пакеты с кодом CE называют промаркированными ECN (ECN-marked), а иногда - просто маркированными, если наличие ECN ясно из контекста.

Site - сайт

Дом, мобильное устройство, небольшое предприятие или кампус, где узким местом сети является канал доступа. Этому определению соответствуют не все сети, но оно является полезным и широко распространенным.

Traffic Policing - надзор за трафиком

Ограничение трафика путём отбрасывания пакетов или их перевода в более низкий класс обслуживания (в отличие от внесения задержки, называемого формированием или формовкой трафика - traffic shaping). Надзор может включать ограничение скорости и/или размера всплесков (пиков). Надзор, сосредоточенный на ограничении очередей, а не средней скорости потока, в этом документе называется надзором за перегрузкой (congestion policing), задержкой (latency policing), всплесками (burst policing) или защитой очередей (queue protection). В иных случаях применяется термин «надзор за скоростью (rate policing).

4. Компоненты архитектуры L4S

Элементы архитектуры L4S описаны в следующих 3 параграфах.

[1]Congestion Experienced - наблюдается перегрузка.

4.1. Протокольные механизмы

Архитектура L4S включает a) отмену прежнего использования идентификатора, b) новое назначение того же идентификатора, c) дополнение необязательных идентификаторов, как описано ниже.

a. Важным аспектом расширяемого контроля перегрузок является использование явных сигналов о перегрузке. В классическом ECN [RFC3168] требуется считать сигнал ECN эквивалентом отбрасывания независимо от генерации сигнала в сети или на отвечающем хосте. L4S требует от сетей и хостов поддерживать более чёткую трактовку каждого сигнала ECN, который менее важен, чем отбласывание. Поэтому сигнала L4S:

   - могут быть более частыми;
   - могут передаваться незамедлительно (без задержки, требуемой для сглаживания флуктуаций очереди.

   Для поддержки L4S, пришлось изменить стандартную спецификацию ECN [RFC3168], чтобы пакеты L4S могли передаваться не только как эквивалент отбрасывания. [RFC8311] является обновлением стандарта, смягчающим требования [RFC3168] (и некоторых других Standards Track RFC) и обеспечивающим возможность экспериментальных изменений, предложенных для L4S. Кроме того, ког ECT(1), ранее считавшийся экспериментальным ECN nonce [RFC3540], в [RFC8311] перенесён в число устаревших (historic), чтобы код можно было использовать заново.

b. В [RFC9331] указано, что ECT(1) служит идентификатором для пакетов L4S при отдельной от классических пакетов обработке. Это соответствует требованиям по указанию альтернативной обработки ECN в [RFC4774].

   Ко CE служит для индикации возникновения перегрузки в обоих случаях (L4S и Classic). Это вызывает опасение, что прежний механизм Classic AQM на пути может пометить некоторые пакеты ECT(0) как CE и затем они будут ошибочно отнесены к очереди L4S. В Приложении B к [RFC9331] разъяснено, что должны совпасть 5 маловероятных условий, чтобы возникли нежелательные эффекты, которые и в этом случае приведут лишь к пренебрежимо малой вероятности ненужного повтора передачи.

c. Оператор сети может захотеть включать определённый не отвечающий трафик, не относящийся к L4S, в очередь L4S, если он представляется достаточно сглаженным и низкоскоростным, чтобы не создавать очередь (например, VoIP, дейтаграммы синхронизации сетевых игр с малой скоростью, DNS, LDAP[1] и т. п.). Такой трафик требуется помечать конкретными идентификаторами, например кодом Diffserv для малой задержки, таким как EF [RFC3246], NQB[2] [NQB-PHB], или своим идентификатором оператора.

4.2. Сетевые компоненты

Архитектура L4S нацелена на обеспечение малой задержки без необходимости выполнять в элементах сети операции на уровне отдельных потоков. Тем не менее, архитектура не отвергает решений по потокам. Ниже описаны известные схемы: a) DualQ Coupled AQM с L4S AQM в одной очереди, связанным с Classic AQM в другой, b) очереди по потокам с экземпляром Classic и L4S AQM в каждой очереди, c) двойные очереди с AQM для каждого потока, но без очередей по потокам.

a. Dual-Queue Coupled AQM (Рисунок 1) обеспечивает упомянутую выше полупроницаемую мембрану.

   - Изоляция задержки. Применяются 2 отдельных очереди для изоляции задержки в очереди L4S от большей задержки, требуемой для поддержки полной загрузки канала классическим трафиком.
   - Объединение пропускной способности. Две очереди действуют так, будто они имеют общую пропускную способность, где потоки любого типа получают примерно равные доли без необходимости идентификации потоков в планировщике. Это достигается за счёт наличия AQM в каждой очереди с передачей сигналов перегрузки от Classic AQM в обе очереди так, чтобы обеспечить согласованную реакцию обоих классов контроля перегрузки. В частности, Classic AQM выдаёт сигналы отбрасывания и маркировки на основе перегрузки в его очереди и это влияет на вероятность маркировки в очереди L4S. Степень связывания сигналов перегрузки между двумя очередями достаточна для того, чтобы потоки L4S замедлялись, оставляя нужную долю пропускной способности классическим потокам (как будто это потоки одного типа, использующие общую очередь).

   После этого планировщик может обслуживать очередь L4S с приоритетом (вход с высшим приоритетом указан 1 на рисунке), поскольку объём трафик L4S недостаточно велик для полного использования предоставленного приоритета. Поэтому:

   - для кратковременной изоляции задержек (доли RTT) приоритизация очереди L4S защищает малую задержку, позволяя всплескам (пикам) быстро рассеиваться;
   - для объединения пропускной способности в более продолжительных интервалах времени (RTT и больше) классическая очередь создаёт равное и противонаправленное давление на трафик L4S, чтобы ни один из типов не получил преимущества при разделении пропускной способности; противоречие между приоритизацией L4S и связыванием с маркировкой из Classic AQM обеспечивает приблизительную беспристрастность по отношению к потокам.

   Для предотвращения приоритизации трафика L4S, на некоторое время блокирующей классическую очередь в отдельных реализациях, рекомендуется задавать приоритет по условию, а не строго (см. Приложение A к спецификации DualQ [RFC9332]).

   При отсутствии классического трафика в дело включается AQM очереди L4S, начиная маркировку перегрузки с очень малой очереди, поэтому для трафика L4S обеспечивается очень малая задержка в очереди.

   Если какая-либо из очередей становится постоянно перегруженной, происходит отбрасывание некоторых пакетов с поддержкой ECN, как рекомендует раздел 7 спецификации ECN [RFC3168] и параграф 4.2.1

[1] Lightweight Directory Access Protocol - облегчённый протокол доступа к каталогам.
[2] Non-Queue-Building - без создания очереди.

рекомендация для AQM [RFC7567]. Компромиссы разных подходов обсуждаются в параграфе 4.2.3 спецификации DualQ [RFC9332] (на рисунке 1 не показано).

Механизм Dual-Queue Coupled AQM задан максимально обобщенно [RFC9332] без указания конкретных AQM в очередях, чтобы разработчики могли реализовать разные идеи. Информационные приложения к спецификации содержат примеры псевдокода для двух разных подходов AQM - DualPI2 (произносится как Dual PI Squared) [DualPI2Linux] с использованием PI2-варианта PIE и не требующий настройки (zero-config) вариант RED, называемый Curvy RED. DualQ Coupled AQM на основе PIE был также задан и реализован для Low Latency DOCSIS [DOCSIS3.1].

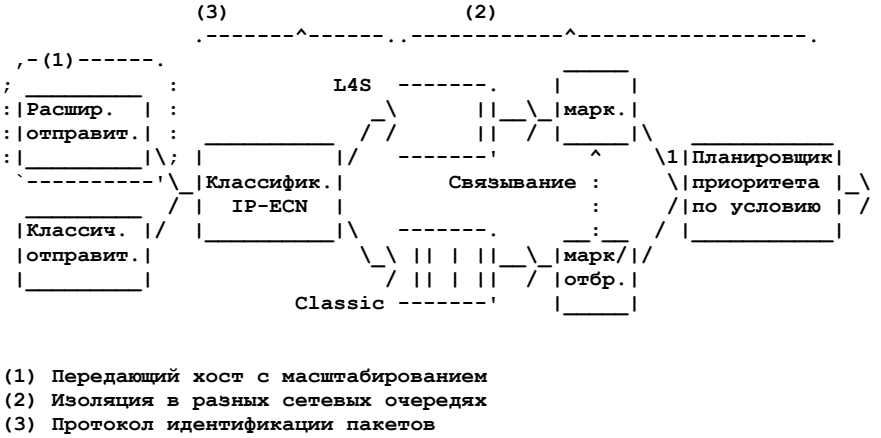

Рисунок 1. Компоненты решения L4S DualQ Coupled AQM.

b. Очереди по потокам и AQM. Для L4S можно использовать планировщик с очередями по потокам, такой как FQ-CoDel или FQ-PIE. Например, в каждой очереди системы FQ-CoDel, а также CoDel AQM обычно имеется опция маркировки ECN с незамедлительным (не сглаженным) низким порогом для поддержки использования в ЦОД (см. параграф 5.2.7 спецификации FQ-CoDel [RFC8290]). В Linux это было изменено так, что низкий порог может применяться исключительно к пакетам ECT(1) [FQ_CoDel_Thresh]. Затем при наличии в очереди потока потока пакетов Not-ECT или ECT(0) применяется Classic AQM (например, CoDel), а если в очереди имеется поток пакетов ECT(1), применяется более низкий порог (обычно доли миллисекунды). Кроме того пакеты ECT(0) и Not-ECT могут выделяться в отдельную от пакетов ECT(1) и CE очередь, чтобы избежать смешивания при использовании ими общего идентификатора потока (например, в VPN).

c. Двойные очереди с AQM по потокам. Следует также обеспечивать возможность использовать двойные очереди для изоляции, но с маркировкой по потокам для управления их скоростями (вместо связанной маркировки по потокам Dual-Queue Coupled AQM). Одна из двух очередей будет изолировать пакеты L4S, которые помечались бы кодом ECN. Скорости потоков можно регулировать с помощью маркировки в конкретном потоке. Целью маркировки может быть дифференциация (например, [Nadas20], где требуется передача дополнительных сигнальных значений по потокам) или выравнивание скоростей потоков (возможно, аналогично Approx Fair CoDel [AFCD] [CODEL-APPROX-FAIR], но с двумя очередями).

Отметим, что при использовании DualQ без указания маркировки по очередям или потокам это означает AQM с двойной очередью и маркировкой по очередям.

4.3. Механизмы хостов

Архитектура L4S включает на конечных хостах 2 механизма, указанных ниже.

a. Расширяемый контроль перегрузок у отправителя. В разделе 2 для расширяемого контроля перегрузок задано поведение расширяемого контроля перегрузок, при котором среднее время от одного сигнала перегрузки до следующего (время восстановления) не зависит от скорости потока при прочих равных условиях. Наиболее распространенным примером является DCTCP, описанный в информационном документе [RFC8257] для протокола, применяемого в настоящее время в контролируемых средах. Составлен список улучшений безопасности и производительности для расширяемого контроля перегрузок, применимого в общедоступной сети Internet (см. Приложение A. Обоснование требований Prague L4S к [RFC9331]). Часть требований, с которыми может быть связан ущерб для других, перечислена в нормативном разделе 4 [RFC9331]. TCP Prague [PRAGUE-CC] реализован в Linux в качестве образца для выполнения этих требований [PragueLinux].

Транспортные протоколы, отличные от TCP, используют различные элементы контроля перегрузок, разработанные для совместимости с Reno. Прежде, чем они смогут использовать услуги L4S, эти механизмы нужно обновить для реализации масштабируемых откликов на перегрузку, которые используют код ECT(1). Рассматриваются расширяемые варианты для недавних транспортных протоколов (например, QUIC), а связанная с L4S ECN часть of BBRv2 [BBRv2] [BBR-CC] является расширяемым контролем перегрузок, предназначенным, среди прочего, для транспорта TCP и QUIC. Реализован также L4S-вариант контроллера RMCAT SCReAM [RFC8298] для потоков, доставляемых по протоколу RTP [SCReAM-L4S].

В параграфе 4.3 спецификации L4S ECN [RFC9331] расширяемый контроль перегрузок определён более подробно и заданы требования для L4S Scalable congestion control.

b. Отклики ECN в некоторых транспортных протоколах уже достаточно детализированы для L4S (в частности, DCCP [RFC4340] и QUIC [RFC9000]), но другие нужно обновлять или они находятся в процесс обновления.

- Для TCP протокол обратной связи ECN использует допущение из Classic ECN [RFC3168] об эквивалентности маркировки ECN отбрасыванию пакета, что делает его непригодным для Scalable TCP. Поэтому реализации получателей TCP нужно обновлять [RFC7560]. Работы по стандартизации и реализации более точных откликов ECN для TCP (AccECN) ещё не завершены [ACCECN] [PragueLinux].

- Отклики ECN лишь очерчены в приложениях к признанной устаревшей второй спецификации SCTP [RFC4960], а более полная спецификация предложена в давно просроченном документе [ECN-SCTP]. Нужно разработать и внедрить новое решение для поддержки L4S в SCTP.

- Для RTP отклики ECN заданы в [RFC6679], а [RFC8888] предлагает последние улучшения Standards Track.

5. Обоснование

5.1. Почему эти компоненты выбраны основными?

Явные сигналы о перегрузке (протокол)

Явные сигналы о перегрузке являются важнейшей частью L4S. Использование отбрасывания в качестве сигнала перегрузки создаёт напряжённость, поскольку отбрасывание является одновременно ущербом (чем меньше, тем лучше) и полезным сигналом (чем больше, тем лучше).

- Явные сигналы о перегрузке можно использовать неоднократно в интервале кругового обхода для обеспечения жёсткого контроля без нанесения вреда. При высокой нагрузке могут передаваться ещё более явные сигналы, чтобы очередь оставалась короткой при любой нагрузке. В отличие от этого, механизмы Classic AQM должны часто отбрасывать пакеты при высокой нагрузке для сохранения короткой очереди. При использовании ECN в контроле перегрузки L4S сокращается размер зубьев пилы и возврат к рабочей точке происходит чаще и можно не заботиться о росте числа сигналов из-за увеличения числа зубьев. Сокращение амплитуды зубьев соответствует меньшему интервалу между пустой очередью и очень низким порогом маркировки (~1 мсек в общедоступной сети Internet), поэтому вариации задержки будут очень малы без риска недогрузки каналов.
- Явные сигналы о перегрузке могут передаваться незамедлительно для отслеживания флуктуаций очереди. L4S переносит сглаживание от сетевых устройств к хостам. Сеть не знает RTT для каких-либо потоков, поэтому при сглаживании в сети (как в классическом подходе) предполагается худшее значение RTT, поскольку иначе потоки с большим RTT становятся нестабильными. Это задерживает классические сигналы перегрузки на 100-200 мсек. В отличие от сети, хосты знают своё значение RTT, поэтому в модели L4S хост может сглаживать каждой поток по своему значению RTT, внося при этом лишь незначительную задержку (обычно несколько миллисекунд). Хост может даже не вносить задержку на сглаживание, если это приемлемо (например, при запуске потока).

Оба указанных выше пункта неосуществимы, если явная сигнализация о перегрузке считается «эквивалентом отбрасывания» (как в Classic ECN [RFC3168]), поскольку отбрасывание является не только сигналом, но и «повреждением». Поэтому отбрасывание не может быть очень частым и не может происходить сразу же, иначе пакеты будут слишком часто отбрасываться просто из-за кратковременных флуктуаций в очереди. В результате в L4S AQM очередь L4S использует новый вариант L4S ECN, который не эквивалентен отбрасыванию (параграф 5.2 в спецификации L4S ECN [RFC9331]), тогда как в классических очередях применяется Classic ECN [RFC3168] или отбрасывание, которые по-прежнему эквивалентны друг другу.

До стандартизации Classic ECN были различные предложения придать маркировке ECN значение, отличное от отбрасывания. Однако не было веских причин принимать эти предложения, поэтому был выбран вариант эквивалентности. В [RFC3168] указано:

Среда, где все конечные узлы поддерживают ECN, позволяет разработать новые критерии установки маркера CE и механизмы контроля перегрузки для реакции конечных узлов на CE-пакеты. Однако рассмотрение этого вопроса выходит за рамки данного документа.

Изоляция очередей (сеть)

Контроль перегрузки L4S сохраняет задержку в очереди достаточно низкой, тогда как классическому контролю нужна задержка в очереди порядка RTT, чтобы избежать недогрузки канала. Одна очередь не может иметь двух размеров, поэтому трафик L4S нужно выделять в отдельную очередь (например, DualQ) или очереди (скажем, FQ).

Связанные уведомления о перегрузке

Связывание уведомлений о перегрузке между двумя очередями, как в DualQ Coupled AQM, не требуется, но обеспечивает простой способ позволить отправителям определять свою скорость пакет за пакетом, вместо её переопределения сетевым планировщиком. Альтернативой является управление скоростью каждого прикладного потока сетевым планировщиком (см. обсуждение в параграфе 5.2. Что L4S добавляет к имеющимся подходам).

Идентификатор пакета L4S (протокол)

При наличии в сети хотя бы 2 вариантов обработки пакетов, хостам нужен идентификатор на уровне IP для указания желаемого типа обработки.

Расширяемые уведомления о перегрузке

Расширяемый контроль перегрузки на хосте поддерживает высокую частоту передачи сигналов из сети независимо от скорости потока, поэтому вариации задержки в очередях могут быть малы при стабильных условиях, а скорость может отслеживать изменения доступной пропускной способности насколько возможно быстро.

Низкие потери

Задержки не являются единственной проблемой, решаемой L4S. Связанная с малыми потерями (Low Loss) часть имени указывает, что L4S обычно обеспечивает отсутствие потерь при перегрузке благодаря применению ECN. Иначе сами потери вызывали бы задержку из-за повтора передачи, особенно для коротких потоков [RFC2884].

Расширяемая пропускная способность

Связанная с расширяемостью пропускной способности (Scalable throughput) часть имени означает расширяемые элементы контроля перегрузки по потокам с неограниченным расширением, что позволяет избежать проблем, неизбежных при использовании алгоритмов Reno-friendly [RFC3649]. При разработке в 1988 г. механизмов предотвращения перегрузки TCP было известно, что это не будет расширяться до случаев с большим произведением пропускной способности на загрузку (bandwidth-delay product, см. примечание 6 в [TCP-CA]). Сегодня скорости потоков в широкополосных каналах WAN уже вышли за пределы расширяемости контроля перегрузок Classic Reno, поэтому были развёрнуты более расширяемые варианты TCP CUBIC [RFC8312] и Compound [CTCP]. Однако они тоже уже приближаются к своим пределам масштабируемости.

Рассмотрим в качестве примера сценарий с максимальным RTT 30 мсек на пике каждого зуба пилы. Если скорость передачи пакетов Reno увеличивается в 8 раз (с 1250 до 10000 пакет/с или с 15 до 120 Мбит/с при размере пакетов 1500 байт), время восстановления после перегрузки также растёт в 8 раз от 422 мсек до 3,38 сек. Очевидно, что контролю перегрузки потребуется на восстановление после каждого факта перегрузки несколько

секунд. Механизм CUBIC [RFC8312] был разработан для увеличения масштабируемости, но он приближается к своему пределу. При том же максимальном RTT 30 мсек и скорости 120 Мбит/с CUBIC остаётся в режиме полной совместимости с Reno и восстановление займёт около 4,3 сек. Однако при следующем увеличении скорости в 8 раз до 960 Мбит/с алгоритм переходит в режим CUBIC со временем восстановления 12,2 сек. С этого момента каждое следующее увеличение скорости в 8 раз удваивает время восстановления CUBIC (кубический корень от 8 равен 2), например, при скорости 7,68 Гбит/с время восстановления будет 24,3 сек. При расширяемом контроле перегрузок, таком как DCTCP или Prague выдаётся в среднем 2 сигнала на интервал кругового обхода независимо от скорости потока, что делает динамический контроль очень строгим.

Представление о скорости отдельного потока загрузки (download) на момент написания документа (2021 г.) можно составить по данным [BDPdata] - усреднённая глобально средняя скорость стационарного доступа в 2020 г. составляла 103 Мбит/с, а среднее базовое значение RTT для CDN[1] - от 25 до 34 мсек (2019 г.). Усреднение по странам выполнялось с учётом числа пользователей Internet (данные, собранные по миру, имеют разную достоверность, но в документе применена двойная проверка и результаты хорошо согласуются с другим источником). Таким образом, одиночному потоку CUBIC в лучшем случае потребовалось бы около 200 интервалов RTT (5 секунд) для восстановления после каждого спада пилы, если поток был достаточно долгим. Это указано как «лучший случай», поскольку предполагается, что все применяют AQM, тогда как на деле большинство пользователей имеет (возможно раздутый) буфер tail-drop. В этом случае (tail-drop) вероятное среднее время восстановления было бы не меньше 20 сек. (4 раза по 5 сек), поскольку RTT под нагрузкой будет по меньшей мере вдвое превышать значение для AQM, а время восстановления потоков Reno-friendly пропорционально квадрату RTT.

Хотя работа по масштабированию средств контроля перегрузок обычно начинается с транспорта TCP, сказанное выше относится и к другому транспорту (например, SCTP и QUIC) и менее гибким алгоритмам (например, RMCAT), которые обычно основаны на похожих разработках.

5.2. Что L4S добавляет к имеющимся подходам

Ниже указаны связанные с тем же пространством проблем подходы, для которых L4S что-то добавляет или улучшает.

Diffserv

Diffserv решает задачу выделения пропускной способности для важного трафика, а также сокращения задержки в очередях для трафика, чувствительного к задержкам. L4S решает лишь задачу сокращения задержки в очереди. Потребность в Diffserv сохраняется там, где нужно отдать приоритет важному трафику (например, из коммерческих соображений или для защиты важного инфраструктурного трафика) [L4S-DIFFSERV]. L4S может обеспечивать малые задержки для всего трафика в каждом классе Diffserv (включая случай наличия лишь принятого по умолчанию класса Diffserv).

Diffserv может сокращать задержки лишь для малой части трафика в узком месте канала. Как уже сказано, этот метод влияет лишь на часть приложений, которым нужны малые задержки, работающих одновременно на сайте (например, дома, в небольшом офисе или на мобильном устройстве). L4S, напротив, работает для всего трафика и не вносит издержек управления (правила или соглашения для трафика), связанных с предпочтением для отдельных пакетов по отношению к другим. Это даёт L4S больше шансов на сквозное внедрение.

В частности, если сеть не доверяет конечным системам в части приоритизации пакетов, она самостоятельно задаёт для пакетов класс Diffserv. Однако доступные в таких сетях методы, такие как проверка идентификаторов потоков или более глубокий анализ сигнатур приложений, не всегда совмещаются с шифрованием на уровне выше IP [RFC8404]. В таких случаях пользователям приходится выбирать между конфиденциальностью и качеством обслуживания (Quality of Service или QoS).

Как и в Diffserv, идентификатор L4S размещается в заголовке IP. Но, в отличие от Diffserv, идентификатор L4S не указывает желание или необходимость некого уровня обслуживания. Скорее, этот метод обещает некое поведение (расширяемые отклики на перегрузку), которое сеть может при необходимости проверить объективно. Это связано с тем, что малые задержки зависят от коллективного поведения хостов, а приоритизация пропускной способности - от сети.

Современные механизмы AQM

Механизмы AQM для классического трафика, такие как PIE и FQ-CoDel, значительно снижают задержку в очередях по сравнению с работой без AQM. L4S дополняет эти AQM и ему не следует препятствовать как можно более широкому развёртыванию этих механизмов. Тем не менее, AQM сами по себе не могут существенно сократить задержку в очередях без значительного снижения уровня использования канала, поскольку корнем проблемы является хост, где классические методы контроля перегрузок вносят большие пилообразные вариации скорости. L4S разрешает этот конфликт между задержкой и загрузкой канала, позволяя хосту минимизировать амплитуду зубьев пилы. Classic AQM с одной очередью недостаточно, чтобы хосты могли использовать меньшую амплитуду пилы, по двум причинам: i) сокращение не будет снижать задержку в AQM, разработанном для большей амплитуды классической пилы, поскольку очередь в каждый момент может иметь лишь один размер и ii) сокращение амплитуды предполагает повышение частоты зубьев, поэтому потоки L4S будут вызывать в Classic AQM частую маркировку ECN, которая будет выглядеть как очень значительная перегрузка классических потоков и в результате значительно снижать их скорость (6.4.4. Поток L4S с Classic ECN в узком месте).

Очереди и маркировка по потокам

Подходя на уровне потоков, такие как FQ-CoDel и Approx Fair CoDel [AFCD], не совместимы с L4S. Однако очереди на уровне потока самой по себе недостаточно, она лишь изолирует один поток от других, но не от себя. В реализации на уровне потоков требуется добавлять расширяемый контроль перегрузок, что уже сделано в Linux FQ-CoDel (см. параграф 5.2.7 в [RFC8290] и [FQ_CoDel_Thresh]). Без такого простого изменения AQM на уровне потоков, такие как FQ-CoDel, не способны поддерживать приложения которым нужны сразу высокая пропускная способность и малая задержка, например, основанное на видео управление удалёнными процедурами или интерактивное видео на основе облака[2].

Хотя методы, работающие на уровне потока, не совместимы с L4S, важно иметь альтернативу DualQ, поскольку обработка сквозных потоков (L4) в сети (L3 и L2) препятствует некоторым сквозным функциям.

---

[1] Content Delivery Network - сеть доставки содержимого. *Прим. перев.*

[2] Может показаться, что задержку, созданную самой очередью, не следует учитывать, поскольку исключение задержки в сети просто переносит её к отправителю. Однако современные адаптивные приложения, например, HTTP/2 [RFC9113] и некоторые интерактивные приложения (6.1. Приложения), могут хранить объекты с малой задержкой в начале своей локальной очереди отправки, перемешивая приоритеты других объектов в зависимости от хода других передач (см. например, [Iowat]). После выпуска объектов в сеть перемешивание уже невозможно.

a. Варианты L4S на уровне потоков, подобные FQ-CoDel, не совместимы с полным сквозным шифрованием идентификаторов транспортного уровня для приватности и конфиденциальности (например, IPsec или шифрованные туннели VPN в отличие от DTLS через UDP), поскольку им нужно заглядывать в пакеты для получения идентификаторов сквозных транспортных потоков.
   В отличие от этого DualQ L4S не требует заглядывать в пакет глубже уровня IP. Пока операторы применяют подход DualQ, пользователи получают очень малые задержки при сквозном шифрованием [RFC8404].

b. При использовании L4S на уровне потоков сеть берет на себя контроль относительной скорости каждого прикладного потока. Некоторые видят в этом преимущество, поскольку сеть не позволяет отдельному потоку работать быстрей других, другие считают неотъемлемым свойством Internet возможность каждого потока контролировать свою скорость, принимая во внимание другие потоки через сигналы перегрузки. Последние считают, что это позволяет развивать приложения, заинтересованные в скорости, например, i) видеопотоки с переменной скоростью, которая выделяется примерно поровну для каждого вместо принудительного постоянного значения, или ii) сквозная «сборка мусора» [RFC6817] с использованием меньшей (чем поровну) пропускной способности [LEDBAT_AQM].
   Архитектура L4S не требует от IETF предпочтения одного подхода перед другими, поскольку поддерживает оба, позволяя «рынку» принять решение. Тем не менее, в духе принципа «делать что-то одно и делать это хорошо» [McIlroy78] вариант DualQ обеспечивает малые задержки, не предрешая вопрос управления скоростью потоков, которое при желании может быть добавлено отдельно. В отличие от планирования, «регулировщик» (policer) будет разрешать приложению контролировать скорость до определённого момента, но сеть все равно может установить точку вмешательства для предотвращения захвата ресурсов отдельным потоком.

ABE

L4S является не альтернативой, а дополнением к ABE (Alternative Back-off ECN), обеспечивающим меньшую задержку в очереди. ABE [RFC8511] меняет поведения хоста при откликах на маркеры ECN для более полного использования канала связи и обеспечения потокам ECN большей пропускной способности. ABE использует ECT(0) и предполагает, в сети одинаковую трактовку ECN и отбрасывания, используя меньшую задержку в очередях, которую могут обеспечить методы AQM. Однако, как отмечено выше, AQM все равно не могут существенно сократить задержки без потери загрузки канала (для не связанных с ABE потоков).

BBR

BBR[1] [BBR-CC] контролирует сквозную задержку в очередях без какой-либо специальной логики в сети (такой как AQM) и поэтому может работать практически на любом пути. BBR сохраняет достаточно малую задержку в очередях, но иногда не столь малую, как в современных AQM, таких как PIE или FQ-CoDel, и уж точно не такую малую, как при использовании L4S. Задержка не остаётся малой постоянно из-за регулярных всплесков при зондировании пропускной способности и энергичной фазы запуска потока.
L4S дополняет BBR. В BBRv2 может применяться L4S ECN (при доступности) и расширяемый контроль перегрузок L4S в ответ на сигналы ECN из точек пути [BBRv2]. Сигналы L4S ECN дополняют основанные на задержке аспекты контроля перегрузки BBR явной индикацией, которую хосты могут использовать для схождения на беспристрастной скорости и сохранения задержки ниже установленной сетью цели. Без L4S ECN оба эти аспекта требуется предполагать или оценивать.

6. Применимость

6.1. Приложения

Транспортный уровень, решающий текущие проблемы с задержками, обеспечит новые возможности для услуг, продукции и приложений. Благодаря L4S указанные ниже имеющиеся приложения будут значительно лучше работать под нагрузкой:

- игры, включая облачные;
- VoIP;
- видеоконференции;
- просмотр web-страниц;
- (адаптивные) видео-потоки;
- мгновенные сообщения.

Значительное сокращение задержек позволит перенести в облако некоторые функции интерактивных приложений, включая:

- облачные интерактивные видео-приложения;
- виртуальная и добавленная реальность на основе облака.

Два упомянутых приложения были успешно продемонстрированы с L4S при работе по каналу доступа 40 Мбит/с загруженному множеством чувствительных к задержкам приложений из приведённого выше списка работающих одновременно через одну очередь в узком месте [L4Sdemo16] [L4Sdemo16-Video]. В первом случае видео-панораму футбольного стадиона можно было поворачивать и сжимать так, что прокси в облаке мог на лету генерировать субокно видео потока матча под управлением движениями пальцев со стороны каждого пользователя. Во втором гарнитура виртуальной реальности отображала картинку с 360-градусной камеры в гоночном автомобиле. Отображаемое поле определялось поворотом головы пользователя и извлекалось из облачного прокси. В обоих случаях при базовой сквозной задержке в 7 мсек дополнительная задержка в очереди (около 1 мсек) была столь мала, что изображение казалось созданным локально.

Управление пальцами или поворотом головы при просмотре круговой панорамы очень чувствительно к задержкам (значительно сильнее, чем VoIP), поскольку человеческий глаз способен замечать очень малую задержку (порядка 1 мсек) при отставании изображения от действия (движение пальцев или поворот головы), воспринимаемого вестибулярным аппаратом внутреннего уха. При использования вариант AQM изображение заметно отставало.

[1]Bottleneck Bandwidth and Round-trip propagation time - пропускная способность в узком месте и время кругового обхода.

Без малой задержки в очереди, обеспеченной L4S, облачным приложениям, подобным этим, потребовалась бы гораздо большая пропускная способность сети доступа (выгрузка из облака полной панорамы) и дополнительная локальная обработка, которая повысила бы энергопотребление и массу устройств отображения в шлеме. При выполнении всей интерактивной обработки в облаке передавать нужно лишь данные, отображаемые конечному пользователю.

Другие высокоскоростные приложения с малыми задержками, такие как интерактивное удалённое присутствие и удалённое управления машинами и производственными процессами с использованием видео, просто не могут работать без обеспечения малой задержки в очереди. Потерю времени здесь не скомпенсировать локальной обработкой и расширением пропускной способности при доступе.

6.2. Варианты применения

Ниже приведены варианты применения L4S, рассматриваемые различными заинтересованными сторонами.

- Узкое место в какой-либо сети доступа, например, DSL, пассивная оптическая сеть (Passive Optical Network или PON), кабель DOCSIS, мобильная или спутниковая сеть, канал Wi-Fi (см. 6.3. Применимость для конкретных канальных технологий, где рассмотрены различные канальные технологии).
- Частные сети с неоднородными ЦОД без единого администрирования, позволяющего одновременно вносить изменения для отправителей, получателей и сетей, требуемые для внедрения DCTCP:
  - частные ЦОД, соединённые через распределенные сети, с раздельным администрированием внутри одной компании;
  - ЦОД, управляемые разными компаниями и соединённые в сеть с общими интересами (например, финансы);
  - ЦОД с арендаторами (облако), использующими стек выбранной операционной системы (Infrastructure as a Service или IaaS);
- Различный контроль перегрузок на уровне транспорта (или приложений):
  - эластичный контроль (TCP/SCTP);
  - приложения в реальном масштабе времени (RTP, RMCAT);
  - запрос-отклик (DNS/LDAP).
- Потребность в QoS с малой задержкой но без проверки и вмешательства со стороны уровня IP [RFC8404]:
  - мобильные и другие сети, как правило, проверяют вышележащие уровни для понимания потребностей приложений в QoS, однако с ростом потребностей в обеспечении приватности и шифрования L4S предоставляет альтернативное решение; не нужно выбирать, предпочтительный трафик в очереди, когда L4S может обеспечить благоприятную постановку в очередь для всего трафика.
- При минимизации задержки в очередях приложения с фиксированным бюджетом задержки могут взаимодействовать на больших расстояниях или по обходным путям, например, через более длинные цепочки сервисных функций [RFC7665] или маршрутизаторы для анонимизации (onion router).
- При минимизации вариаций задержки можно сократить размер компенсационных буферов (dejitter) на приёмной стороне, что должно улучшить интерактивный интерфейс.

6.3. Применимость для конкретных канальных технологий

Некоторые технологии канального уровня объединяют множество пакетов в блоки (burst), буферизуя для этого входящие пакеты. Такое агрегирование применяется в Wi-Fi, PON, кабельных модемах, тогда как проводные сети Ethernet и DSL не делают этого. Ни один отправитель (независимо от L4S) не может каким-либо способом сократить буферизацию, требуемую для агрегирования пакетов. Механизмам AQM не следует считать эти буферы частью управляемой очереди, поскольку никакие сигналы перегрузки не могут сократить размер буфера.

Некоторые канальные технологии добавляют буферизацию по иным причинам.

- Радиоканалы (сотовые и спутниковые сети, Wi-Fi) с удаленным источником являются наиболее сложным случаем. Пропускная способность такого канала может быстро изменяться на порядки, поэтому считается желательным наличие постоянной очереди для использования внезапно возросшей пропускной способности.
- В сотовых сетях дополнительные сложности связаны с буферизацией требуемой для незаметного перехода между сотами (hand-over).

L4S не может устранить необходимость этих различных форм буферизации. Однако, устраняя буферизацию для больших зубьев пила классического контроля перегрузки, L4S раскрывает более мелкие зубья для наблюдения.

До сих пор буферизация, обусловленная этими дополнительными причинами, обычно завышалась под предлогом того, что не одна из них не считалась самой большой (наиболее важной). Однако после устранения такой причины становится возможным их минимизация, например, за счёт снижения размера блоков агрегирования и интервалов планирования MAC.

Кроме того, некоторые каналы (особенно радио) более подвержены потерям при передаче. В параграфе 6.4.3 указано, реакция L4S на потери должна быть столь же резкой, как и при классическом отклике. Однако упомянутое там же исследование показало возможность существенно более эффективного восстановления потерь на канальном уровне за счёт смягчения ограничений для упорядоченности пакетов L4S.

6.4. Вопросы развёртывания

Механизмы L4S AQM, будь то DualQ [RFC9332] или FQ [RFC8290], сами являются механизмами поэтапного внедрения L4S, где трафик L4S может сосуществовать с имеющимся классическим трафиком (Reno-friendly). В параграфе 6.4.1 описано, как внедрение L4S AQM лишь на одном узле с каждой стороны канала доступа обеспечивает почти все преимущества L4S.

L4S включает как сеть, так и конечные системы и в параграфе 6.4.2 предложены некоторые типовые последовательности внедрения каждой части и разъяснено, почему внедрение лишь одной части сразу обеспечивает значительные преимущества.

В параграфах 6.4.3 и 6.4.4 описан случай обратного поэтапного внедрения, где L4S AQM не развёртывается в узком месте сети и любой поток L4S, проходящий через это место, должен учитывать конкуренцию с классическим трафиком.

6.4.1. Топология развёртывания

L4S AQM не потребуется внедрять во всей сети Internet, пока L4S не принесёт пользы кому бы то ни было. Операторы открытых сетей доступа в Internet обычно проектируют свои сети так, что узкое место почти всегда возникает на одном известном (логическом) канале. Это сокращает издержки на управление очередями.

Ситуация в многосвязных (mesh) сетях отличается и будет рассмотрена ниже. Вариант с известным узким местом в общем случае применим к доступу в Internet для всех сайтов, включая домашние сети, мелкие и средние кампусы и предприятия и даже сотовые устройства (Рисунок 2). Кроме того, вариант с известным узким местом обычно применим к разным канальным технологиям доступа, таким как xDSL, кабельные модемы, PON, сотовые сети, прямые оптические каналы, спутниковые системы.

Поэтому всем достоинствам услуг L4S следует быть доступными при внедрении L4S AQM на входе в узкий канал. Все достоинства в восходящем направлении обычно становятся доступными при внедрении L4S AQM на входе восходящего канала. Многодомные сайты смогут полностью воспользоваться такими преимуществами после внедрения служб на всех своих каналах доступа.

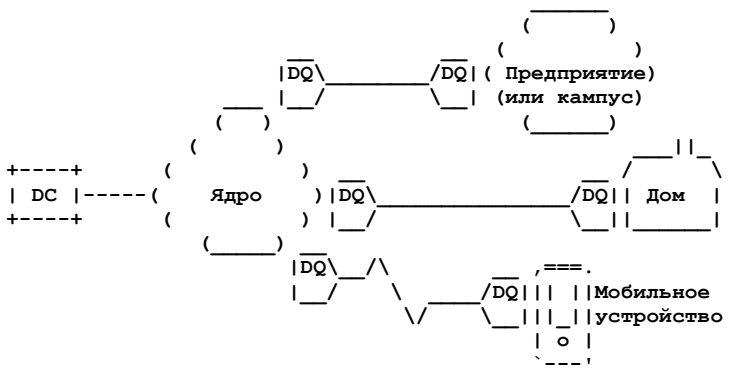

Рисунок 2. Вероятное размещение DualQ (DQ) в базовой топологии доступа.

Внедрение в mesh-топологии зависит от уровня перегрузки в ядре сети. Если перегрузки в ядре нет или производительность достаточно высока и узкими местами почти всегда являются ребра, достаточно будет внедрить L4S AQM на таких рёбрах. Например, некоторые сети ЦОД спроектированы так, что узким местом является гипервизор или сетевые контроллеры (Network Interface Controller или NIC) хостов, а другая узость возникает в коммутаторе стойки (top-of-rack), на портах, обращённых к хосту и ядру.

Затем L4S AQM потребуется там, где узким местом в доме становятся каналы Wi-Fi. L4S AQM может потребоваться в любых сохраняющихся узких местах, таких как соединения между сетями, например в некоторых точках обмена Internet, а также на входах и выходах каналов WAN между ЦОД.

6.4.2. Последовательность внедрения

Чтобы отдельный поток L4S обеспечивал преимущества нужно развернуть 3 (иногда 2) элемента: i) контроль перегрузки у отправителя, ii) AQM в узком месте и iii) обновление старого транспорта (а именно, TCP) в части обратной связи от получателя. С этими же проблемами столкнулось внедрение ECN [RFC8170], откуда были извлечены уроки.

Во-первых, при развёртывании L4S используется наличие DCTCP уже на многих хостах Internet (например, клиенты и серверы Windows, FreeBSD, Linux). Поэтому L4S AQM можно внедрить в узком месте сети, чтобы сразу же получить рабочее развёртывание всех частей L4S для тестирования при условии смены кода ECT(0) на ECT(1). В DCTCP нужно устранить несколько проблем безопасности для базового применения в общедоступной сети Internet (см. параграф 4.3 спецификации L4S ECN [RFC9331]), но протокол DCTCP по умолчанию не включён и эти вопросы можно решать в рамках контролируемых внедрений или экспериментов.

Во-вторых, повышение производительности с помощью L4S столь значительно, что позволяет создавать новые интерактивные службы и продукцию, которое до этого не были возможны. Компаниям гораздо проще начать работы по внедрению, если имеется бюджет на пробные версии продукции. Если бы происходило лишь постепенное повышение производительности (как в Classic ECN), расходы на внедрение оправдать, скорей всего, было бы гораздо сложнее.

В-третьих, идентификатор L4S выбран так, что сетевые операторы могут сначала включить L4S лишь для некоторых клиентов или отдельных приложений. Выбор идентификатора тщательно продуман, чтобы не поставить под угрозу будущее развитие в направлении внедрения L4S как всеобщего сервиса Internet. Это обусловлено тем, что идентификатор L4S задан не только как сквозное поле ECN, но и может сочетаться с другими заголовками пакетов или статусом клиента или его канала доступа (см. параграф 5.4 в [RFC9331]). Операторы могли бы сделать это даже без одобрения IETF, однако IETF лучше указать, что локальный идентификатор должен применяться в сочетании с идентификатором IETF - ECT(1). Если оператор выбрал подход лишь с локальным применением, он просто будет удалять дополнительное правило, чтобы сервис работал через Internet (промежуточные устройства, партнёров и т. п.).

	Серверы или прокси	*Канал доступа*	*Клиенты*
0	DCTCP (имеющийся)		DCTCP (имеющийся)
1		Добавление L4S AQM в нисходящем канале	
		Работа в нисходящем направлении по внедрению в контролируемой среде (пробному)	
2	Обновление DCTCP до TCP Prague		Замена откликов DCTCP на AccECN
		Полностью работающее нисходящее направление	
3		Добавление L4S AQM в восходящем канале	Обновление DCTCP до TCP Prague
		Полностью работающие нисходящее и восходящее направление	

Рисунок 3. Пример последовательности развёртывания L4S.

На рисунке 3 приведён пример последовательности внедрения элементов L4S при наличии DCTCP с обеих сторон.

1. DCTCP не применяется в общедоступной сети Internet, поэтому здесь подчёркнуто, что поток DCTCP полностью локализован в контролируемой среде развёртывания. Внутри этой среды сразу после внедрения L4S AQM пробный поток DCTCP получает выгоду даже без дополнительного внедрения. В этом примере внедрение происходит сначала в нисходящем направлении, но в других ситуациях внедрение может начинаться с восходящего направления. Если ранее для нисходящего доступа не был развернут механизм AQM, L4S AQM существенно улучшает классические услуги (а также добавляет услуги L4S). Если AQM уже был внедрён, классические услуги не изменяются (а L4S добавит улучшения).

2. Здесь TCP Prague [PRAGUE-CC] представляет вариант DCTCP, разработанный для применения в рабочей среде Internet (т. е. он соответствует всем требованиям раздела 4 в спецификации L4S ECN [RFC9331], что означает возможность использования в общедоступной сети Internet). Если приложение в основном однонаправленное, TCP Prague на передающей стороне обеспечит все требуемые преимущества при условии поддержки принимающей стороной откликов Accurate ECN (AccECN) [ACCECN].

   Для TCP нужны отклики AccECN от другой стороны, но это базовые отклики ECN, внедрение которых уже запланировано для других целей, таких как DCTCP и BBR. Внедрение на обеих сторонах может происходить в любом порядке, поскольку в TCP контроль перегрузок L4S включается лишь при указании поддержки AccECN в процессе начального согласования. Таким образом, внедрение TCP Prague на сервере позволяет пробному внедрению L4S перейти в рабочее состояние для одного направления, независимо от развёртывания AccECN на другой стороне. Дополнительным мотивом для этого этапа может быть большая производительность TCP Prague по сравнению с DCTCP (см. Приложение A.2 к спецификации L4S ECN [RFC9331]).

   В отличие от TCP, отклики QUIC ECN [RFC9000] с самого начала поддерживают L4S. Поэтому для транспорта QUIC внедрение на этом этапе контроля перегрузок Prague является простым и достаточным.

   Если при использовании транспорта QUIC прокси размещается на пути между серверами-источниками и узким местом доступа для множества клиентов, обновление этого прокси с внедрением расширяемого контроля перегрузок обеспечит преимущества L4S в нисходящем направлении сразу для всех таких клиентов независимо от обновления серверов источников. Если же прокси не обновить, клиенты не смогут воспользоваться преимуществами L4S в нисходящем направлении даже при обновлении любого из серверов-источников для поддержки L4S.

   В случае TCP обновление прокси для поддержки TCP Prague обеспечит преимущества L4S в нисходящем направлении всем клиентам, поддерживающим AccECN (независимо от поддержки ими L4S). В восходящем направлении прокси будет поддерживать AccECN как получатель и любой клиент, внедривший у себя L4S получит преимущества в восходящем направлении независимо от поддержки AccECN серверами за прокси.

3. Этот этап состоит из 2 шагов для включения L4S в восходящем направлении. L4S AQM или TCP Prague можно развернуть в любом порядке, как уже описано. Для мотивирования первого шага отложенная выгода от включения новых услуг после второго шага должна превышать инвестиционные риски первого шага. Как уже отмечено, потенциал новых интерактивных услуг обеспечивает такую мотивацию. L4S AQM также значительно улучшает классические услуги в восходящем направлении, если механизм AQM не был внедрён ранее.

Отметим, что последовательность внедрения может быть иной. Например, сначала можно обновить восходящее направление, использовать сквозной протокол, отличный от TCP (например, QUIC и RTP). Устройства, подобные 3GPP, могут потребовать внедрения L4S в пользовательском оборудовании 5G и т. д.

6.4.3. Поток L4S с узким местом без ECN

Если L4S включён между двумя хостами, отправитель L4S должен безопасно сосуществовать с Reno при реакции на какое-либо отбрасывание пакета (см. параграф 4.3 в спецификации L4S ECN [RFC9331]). К сожалению, помимо защиты классического трафика это ухудшает работу сервиса L4S при возникновении какой-либо потери, даже если её причиной не является перегрузка в узком месте. Это может быть, например,

- потеря в другом узком месте пути (например, в результате всплеска пакетов в мелких очередях);
- ошибки при передаче (например, электрические помехи);
- правила управления скоростью.

Для решения этой проблемы разрабатывается 3 взаимодополняющих подхода, но они ещё в стадии исследования.

- В контроле перегрузок Prague игнорируются некоторые потери, связь которых с перегрузкой представляется маловероятной (применяются некоторые идеи из BBR [BBR-CC] в части изолированных потерь). Это позволяет маскировать любые из указанных выше потерь, сосуществуя с контролем перегрузки по потерям.
- Сочетание недавнего подтверждения (Recent Acknowledgement или RACK) [RFC8985], L4S и повторов передачи на канальном уровне без изменения порядка может исправлять ошибки передачи без задержки из-за блокировки в начале линии (head of line), обычно связанной с повтором передачи на канальном уровне [UnorderedLTE] [RFC9331].
- Гибридное управление скоростью по ECN и отбрасыванию (8.3. Взаимодействие ограничения скорости и L4S).

Сценарии внедрения L4S, минимизирующие эти проблемы (например, через проводные сети), могут реализоваться в параллель с исследованиями в надежде на то, что успех исследований может расширить применимость L4S.

6.4.4. Поток L4S с Classic ECN в узком месте

Поддержка Classic ECN начинает реализовываться в Internet как повышение уровня (объёма) маркировки CE. Сложно понять, связано это с добавлением поддержки ECN в реализации FQ-CoDel и/или FQ-COBALT, что обычно не вызывает проблем, поскольку планирование в очередях по потокам (flow queue или FQ) по своей природе предотвращает превышение потоком «справедливой» скорости независимо от энергичности этого потока. Однако некоторые маркировки Classic ECN могут быть вызваны внедрением ECN с одной очередью. Этот случай рассматривается в параграфе 4.3 спецификации L4S ECN [RFC9331].

6.4.5. Развёртывание L4S AQM внутри туннеля

В L4S AQM поле ECN служит для сигналов о перегрузке. Поэтому, как и в Classic ECN, при размещении AQM внутри туннеля или на нижележащем уровне для корректной работы сигнализации ECN нужно соответствующее стандартам распространение поля ECN по уровням [RFC6040] [ECN-SHIM] [ECN-ENCAP].

7. Взаимодействие с IANA

Этот документ не требует действий IANA.

8. Вопросы безопасности

8.1. Ограничение скорости трафика

8.1.1. Ограничение скорости на уровне потока

В современной сети Internet провайдеры (ISP) обычно принудительно разделяют пропускную способность общих каналов, выделенных различным сайтам (например, домовладениям, организациям или мобильным пользователям, см. 3. Терминология), с помощью той или иной формы планировщика [RFC0970]. ISP также применяют различные методы (например, перенаправление на средства очистки) для борьбы с лавинными рассылками (flooding). Однако никогда не возникало универсальной необходимости контролировать скорость отдельных потоков приложений, Internet обычно полагается на самоограничение контроля перегрузки у отправителей для совместного использования «внутрисайтовой» пропускной способности.

При разработке L4S это сложившееся состояние было сохранено. Для предоставления услуг L4S с использованием DualQ параграф 4.2 в [RFC9332] разъясняет, как неотзывчивым потокам предоставляется не больше преимуществ, чем AQM с одной очередью, независимо от наличия перегрузки.

Если когда либо потребуется управление скоростью по потокам, его можно будет добавить, поскольку оно ортогонально разделению классического трафика и L4S. Как разъяснено в параграфе 5.2, L4S с DualQ обеспечивает малые задержки без необходимости управлять скоростью на уровне потока. Поэтому при потребности в контроле скорости по потокам можно использовать очереди по потокам (FQ) с поддержкой L4S или добавить контроль скорости потока как модульное расширение DualQ. Однако контроль скорости на уровне потока обычно не внедряется в качестве механизма защиты, поскольку активный злоумышленник может просто разделить свой трафик между разными идентификаторами потоков, если скорость каждого потока ограничена.

8.1.2. Ограничение скорости для L4S

В параграфе 5.2 разъяснено, что Diffserv различает пакеты лишь в случае, когда одним пакетам нужно более приоритетное обслуживание, чем другим, и это обычно требует управления скоростью для класса трафика с малой задержкой. Для L4S, напротив, не возникает потребности в ограничении скорости доступа к услугам L4S для защиты классического обслуживания, поскольку L4S снижает свои задержки без ущерба для задержки и скорости любого классического трафика.

На ранних этапах развёртывания (а, возможно, и всегда) некоторые сети не будут предоставлять услуг L4S. В общем случае этим сетям не нужны правила для трафика L4S. От них требуется (в соответствии со спецификациями ECN [RFC3168] и L4S ECN [RFC9331]) не менять идентификатор L4S, поскольку это нарушит сквозной контроль перегрузок. Если сеть уже рассматривает трафик ECN как Not-ECT, она может относить и трафик L4S к Not-ECT. В узком месте такой сети может возникать очередь и отбрасывание пакетов. При обнаружении отбрасывания расширяемым протоколом контроля перегрузки он должен реагировать безопасно для классического контроля перегрузки, как требует параграф 4.3 в [RFC9331]. Это ухудшит сервис L4S, чтобы он был не лучше (и не хуже) классического сервиса best efforts при наличии на пути узкого места без ECN (6.4.3. Поток L4S с узким местом без ECN).

В представляющихся редкими случаях сеть с поддержкой только Classic ECN [RFC3168] в узком месте с одной очередью, может начать ограничивать трафик L4S для защиты конкурирующего с ним трафика Classic ECN (см., например, параграф 6.1.3 эксплуатационных рекомендация L4S [L4SOPS]). Однако параграф 4.3 спецификации L4S ECN [RFC9331] рекомендует отправителю адаптировать свою реакцию на перегрузку для надлежащего сосуществования с потоками Classic ECN, т. е. вернуться к самоограничению.

Некоторые операторы сетей могут ограничивать доступ к услугам L4S, предоставляя его лишь части клиентов. Их классификаторы пакетов (2 на рисунке 1) могут идентифицировать клиентов но некоторым полям (например, диапазону адресов отправителей) в дополнение к классификации по полю ECN. Если соответствует лишь идентификатор ECN L4S, но не адрес отправителя (например), классификатор может направить пакеты (клиентов без доступа к L4S) в классическую очередь. Чёткое разъяснение способов использования операторами дополнительных локальных классификаторов (см. параграф 5.4 в [RFC9331]) призвано устранить любые мотивы сброса идентификаторов L4S. Тогда сквозное сохранение идентификатора L4S ECN будет более вероятным даже без поддержки сервиса на некоторых интервалах пересылки. Такие локальные соглашения будут требовать лишь простой классификации по факту регистрации, а не управляемой и зависящей от приложения проверки трафика на соответствие контракту, как принято в Diffserv.

8.2. Дружественность к задержкам

Подобно классическому сервису, L4S полагается на самоограничение скорости в ответ на перегрузку. Кроме того, L4S требует самоограничения с точки зрения задержки (скачкообразности). Есть надежда, что личной заинтересованности и руководства по динамическому поведению (особенно при запуске потока, который, возможно, придётся стандартизировать) будет достаточно для предотвращения на транспортном уровне чрезмерных всплесков (burst) трафика L4S с учётом того, что от таких всплесков больше всего пострадает задержка передающего приложения.

Поскольку служба L4S может сокращать задержку без заметного роста задержки классического трафика, не должно возникать необходимости ограничивать трафик L4S для защиты задержки классического трафика. Однако ещё предстоит выяснить, не потребуется ли ограничивать всплески для защиты другого трафика L4S. Без этого может возникнуть возможность атак для увеличения задержки сервиса L4S. Для решения этой задачи можно использовать разные механизмы, указанные ниже.

Локальная защита очереди в узком месте
: Функция защиты очередей по потокам (5-tuple) [DOCSIS-Q-PROT] была разработана для очереди с малой задержкой в DOCSIS, где применяется архитектура DualQ L4S. Она защищает сервис с малой задержкой от любых потоков, создающих очереди, которые случайно или злонамеренно классифицируют себя в очередь с малой задержкой. Функция предназначена для оценки потоков исключительно по их вкладу в очередь, а не по скорости потока. Если для общей очереди с малой задержкой возникает риск превышения порога, функция перенаправляет достаточное число пакетов с высокой оценкой в классическую очередь, чтобы сохранить малую задержку.

Очистка распределенного трафика
: Вместо локального ограничения в каждом узком месте можно решать проблему реактивно, например, «наказывая» любые развёртывания новых вредоносных программ со всплесками трафика, аналогично перенаправлению трафика от источников лавинных атак через средства очистки.

Локальное планирование по потокам в узком месте
: Планированию по потокам следует по своей природе изолировать потоки без всплесков (non-bursty) от потоков со всплесками (пиками) трафика (сравнение ограничения и планирования на уровне потоков дано в параграфе 5.2).

Защита очереди подсети распределенного доступа
: Защита очередей по потокам может быть организована для структуры очередей, распределенной по подсети, использующей управляющие сообщения нижележащих уровней (см. параграф 2.1.4 в [QDyn]). Например, в сети радиодоступа пользовательское оборудование уже передаёт регулярные отчёты о состоянии буфера контроллеру сети, который может использовать эти сведения для удалённого ограничения отдельных потоков.

Раскрытие распределенной перегрузки входным ограничителям
: Архитектура раскрытия перегрузки (Congestion Exposure или ConEx) [RFC7713] использует аудит на выходе для побуждения отправителей правдиво сообщать о перегрузке пути по основному каналу, где это может использоваться входными ограничителями. Возможен и сквозной вариант этой архитектуры.

Распределенное кондиционирование трафика на границе домена
: Может быть предпочтительной архитектура, подобная Diffserv [RFC2475], где трафик заранее (проактивно) кондиционируется на входе в домен вместо реактивного ограничения при возникновении очереди в узком месте после объединения с другим трафиком.

Защита очередей распределенного ядра сети
: Функция ограничения (policing) может быть распределена между механизмами на уровне потоков на входе в сеть, которые характеризуют склонность к пикам (burstiness) каждого потока сигналом, передаваемым с трафиком, и механизмами на уровне классов в узком месте, которые воздействуют на эти сигналы, если постановка в очередь действительно происходит после схождения трафика. Это чем-то похоже на [Nadas20], что, в свою очередь, похоже на идею беспристрастной очереди без учёта состояний.

Ни один из этих вариантов защиты очередей не считается неотъемлемой частью архитектуры L4S, которая работает без всех этих вариантов при отсутствии атак (примерно так же, как Internet обычно работает без ограничения скорости по потокам). Действительно, даже при внедрении правил для задержки при нормальных условиях эти механизмы не вмешивались бы и операторы могли бы отключить их. Частью экспериментов с L4S будет выяснение потребности в такой функции и выбор наиболее подходящих механизмов.

8.3. Взаимодействие ограничения скорости и L4S

Как отмечено в параграфе 5.2, L4S следует исключать потребность в классах Diffserv с малой задержкой. Однако классы Diffserv, дающие некоторым приложениям или пользователям преимущество в выделении пропускной способности, останутся применимыми в определённых ситуациях (например, в корпоративных сетях). В рамках таких классов Diffserv зачастую может применяться L4S для обеспечения трафику малой задержки и малых потерь. В рамках таких классов Diffserv доступная пользователю или приложению пропускная способность часто ограничивается регулятором скорости. Аналогично, в принятом по умолчанию классе Diffserv иногда применяются регуляторы скорости для разделения общей пропускной способности.

Классические ограничители скорости отбрасывают все пакеты, превышающие установленный предел скорости, обычно разрешая всплески (есть варианты, где ограничитель перемаркировывает несоответствующий трафик кодом Diffserv, подходящим для отбрасывания, поэтому пакеты при перегрузке могут быть отброшены где угодно). Когда трафик L4S сталкивается с таким регулятором скорости, он испытывает потери пакетов и источник будет возвращаться к классическому контроля перегрузки, теряя преимущества L4S (6.4.3. Поток L4S с узким местом без ECN). Поэтому в сетях где уже имеются регуляторы скорости и планируется внедрение L4S, предпочтительно перепроектировать ограничители скорости, чтобы они стали более дружественны к сервисуL4S. Совместимое с L4S ограничение скорости в настоящее время является областью исследований (отметим, что это отличается от управления задержкой). Можно было бы установить порог начала маркировки ECN чуть ниже порога ограничения скорости или чуть ниже величины пиков, при которой начинается отбрасывание. Например, при 3-цветной маркировке в двумя скоростями [RFC2698] или пороге PCN и маркере избыточной скорости [RFC5670] можно применять маркировку ECN при низкой скорости и отбрасывание - при высокой. Или можно было добавить к имеющемуся ограничителю скорости управление «скоростью перегрузки», например, с использованием «локального» варианта агрегатного регулятора ConEx [CONG-POLICING]. Может быть удастся разработать элементы расширяемого контроля перегрузок, позволяющие менее катастрофически реагировать на потери, которым не предшествовал интервал роста задержки.

Устройству совместимых с L4S регуляторов скорости должен быть посвящен отдельный специальный документ. Обсуждение взаимодействия L4S и Diffserv приведено в [L4S-DIFFSERV].

8.4. Целостность ECN

Разработаны разные варианты защиты целостности в контуре откликов на перегрузку (по потерям, Classic ECN, L4S ECN) от некорректного поведения получателей, отправителей и сети. Краткое описание таких методов в рассмотрением применимости, доводов за и против дано в Приложении C.1 к спецификации L4S ECN [RFC9331].

8.5. Вопросы приватности

Как отмечалось в параграфе 5.2, архитектура L4S не исключает подходов с проверкой сквозных идентификаторов транспортного уровня. Например, поддержка L4S добавлена в FQ-CoDel, где происходит классификация в сети по идентификатору потока приложения. Однако основным нововведением L4S является модель DualQ AQM, не требующая заглядывать внутрь пакета дальше внешнего заголовка IP, поскольку идентификатор L4S помещается в поле IP-ECN.

Таким образом, архитектура L4S обеспечивает очень малую задержку в очереди без необходимости просмотра информации выше уровня IP. Это означает, что пользователям, желающим шифровать идентификаторы потоков приложений, например в туннелях IPsec или иных туннелях VPN с шифрованием, не придётся жертвовать малой задержкой [RFC8404].

Поскольку L4S может обеспечить малые задержки для широкого класса приложений, которе решат использовать эту службу, не возникает потребности тем или иным способом различать отдельные приложения или классы при прохождении через сеть. Это в значительной степени исключает возможность сопоставить требования к задержке трафика с другими идентифицирующими признаками [RFC6973]. Возможно, некоторые типы трафика предпочтут не применять L4S, но грубая бинарная классификация трафика мало что даёт для нарушения приватности.

9. Литература

[ACCECN] Briscoe, B., Kühlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", Work in Progress, Internet-Draft, draft-ietf-tcpm-accurate-ecn-22, 9 November 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-accurate-ecn-22>.

[AFCD] Xue, L., Kumar, S., Cui, C., Kondikoppa, P., Chiu, C-H., and S-J. Park, "Towards fair and low latency next generation high speed networks: AFCD queuing", Journal of Network and Computer Applications, Volume 70, pp. 183-193, DOI 10.1016/j.jnca.2016.03.021, July 2016, <https://doi.org/10.1016/j.jnca.2016.03.021>.

[BBR-CC] Cardwell, N., Cheng, Y., Hassas Yeganeh, S., Swett, I., and V. Jacobson, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-cardwell-iccrg-bbr-congestion-control-02, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-02>.

[BBRv2] "TCP BBR v2 Alpha/Preview Release", commit 17700ca, June 2022, <https://github.com/google/bbr>.

[BDPdata] Briscoe, B., "PI2 Parameters", TR-BB-2021-001, arXiv:2107.01003 [cs.NI], DOI 10.48550/arXiv.2107.01003, October 2021, <https://arxiv.org/abs/2107.01003>.

[BufferSize] Appenzeller, G., Keslassy, I., and N. McKeown, "Sizing Router Buffers", SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 281-292, DOI 10.1145/1015467.1015499, October 2004, <https://doi.org/10.1145/1015467.1015499>.

[COBALT] Palmei, J., Gupta, S., Imputato, P., Morton, J., Tahiliani, M. P., Avallone, S., and D. Täht, "Design and Evaluation of COBALT Queue Discipline", IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), DOI 10.1109/LANMAN.2019.8847054, July 2019, <https://ieeexplore.ieee.org/abstract/document/8847054>.

[CODEL-APPROX-FAIR] Morton, J. and P. Heist, "Controlled Delay Approximate Fairness AQM", Work in Progress, Internet-Draft, draft-morton-tsvwg-codel-approx-fair-01, 9 March 2020, <https://datatracker.ietf.org/doc/html/draft-morton-tsvwg-codel-approx-fair-01>.

[CONG-POLICING] Briscoe, B., "Network Performance Isolation using Congestion Policing", Work in Progress, Internet-Draft, draft-briscoe-conex-policing-01, 14 February 2014, <https://datatracker.ietf.org/doc/html/draft-briscoe-conex-policing-01>.

[CTCP] Sridharan, M., Tan, K., Bansal, D., and D. Thaler, "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks", Work in Progress, Internet-Draft, draft-sridharan-tcpm-ctcp-02, 11 November 2008, <https://datatracker.ietf.org/doc/html/draft-sridharan-tcpm-ctcp-02>.

[DOCSIS-Q-PROT] Briscoe, B., Ed. and G. White, "The DOCSIS® Queue Protection Algorithm to Preserve Low Latency", Work in Progress, Internet-Draft, draft-briscoe-docsis-q-protection-06, 13 May 2022, <https://datatracker.ietf.org/doc/html/draft-briscoe-docsis-q-protection-06>.

[DOCSIS3.1] CableLabs, "MAC and Upper Layer Protocols Interface (MULPI) Specification, CM-SP-MULPIv3.1", Data-Over-Cable Service Interface Specifications DOCSIS 3.1 Version i17 or later, 21 January 2019, <https://specification-search.cablelabs.com/CM-SP-MULPIv3.1>.

[DOCSIS3AQM] White, G., "Active Queue Management Algorithms for DOCSIS 3.0: A Simulation Study of CoDel, SFQ-CoDel and PIE in DOCSIS 3.0 Networks", CableLabs Technical Report, April 2013, <https://www.cablelabs.com/wp-content/uploads/2013/11/Active_Queue_Management_Algorithms_DOCSIS_3_0.pdf>.

[DualPI2Linux] Albisser, O., De Schepper, K., Briscoe, B., Tilmans, O., and H. Steen, "DUALPI2 - Low Latency, Low Loss and Scalable (L4S) AQM", Proceedings of Linux Netdev 0x13, March 2019, <https://www.netdevconf.org/0x13/session.html?talk-DUALPI2-AQM>.

[Dukkipati06] Dukkipati, N. and N. McKeown, "Why Flow-Completion Time is the Right Metric for Congestion Control", ACM SIGCOMM Computer Communication Review, Volume 36, Issue 1, pp. 59-62, DOI 10.1145/1111322.1111336, January 2006, <https://dl.acm.org/doi/10.1145/1111322.1111336>.

[ECN-ENCAP] Briscoe, B. and J. Kaippallimalil, "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-ecn-encap-guidelines-17, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-ecn-encap-guidelines-17>.

[ECN-SCTP] Stewart, R., Tuexen, M., and X. Dong, "ECN for Stream Control Transmission Protocol (SCTP)", Work in Progress, Internet-Draft, draft-stewart-tsvwg-sctpecn-05, 15 January 2014, <https://datatracker.ietf.org/doc/html/draft-stewart-tsvwg-sctpecn-05>.

[ECN-SHIM] Briscoe, B., "Propagating Explicit Congestion Notification Across IP Tunnel Headers Separated by a Shim", Work in Progress, Internet-Draft, draft-ietf-tsvwg-rfc6040update-shim-15, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-rfc6040update-shim-15>.

[FQ_CoDel_Thresh] "fq_codel: generalise ce_threshold marking for subset of traffic", commit dfcb63ce1de6b10b, October 2021, <https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git/commit/?id=dfcb63ce1de6b10b>.

[Hohlfeld14] Hohlfeld, O., Pujol, E., Ciucu, F., Feldmann, A., and P. Barford, "A QoE Perspective on Sizing Network Buffers", IMC '14: Proceedings of the 2014 Conference on Internet Measurement, pp. 333-346, DOI 10.1145/2663716.2663730, November 2014, <https://doi.acm.org/10.1145/2663716.2663730>.

[L4S-DIFFSERV] Briscoe, B., "Interactions between Low Latency, Low Loss, Scalable Throughput (L4S) and Differentiated Services", Work in Progress, Internet-Draft, draft-briscoe-tsvwg-l4s-diffserv-02, 4 November 2018, <https://datatracker.ietf.org/doc/html/draft-briscoe-tsvwg-l4s-diffserv-02>.

[L4Sdemo16] Bondarenko, O., De Schepper, K., Tsang, I., Briscoe, B., Petlund, A., and C. Griwodz, "Ultra-Low Delay for All: Live Experience, Live Analysis", Proceedings of the 7th International Conference on Multimedia Systems, Article No. 33, pp. 1-4, DOI 10.1145/2910017.2910633, May 2016, <https://dl.acm.org/citation.cfm?doid=2910017.2910633>.

[L4Sdemo16-Video] "Videos used in IETF dispatch WG 'Ultra-Low Queuing Delay for All Apps' slot", <https://riteproject.eu/dctth/#1511dispatchwg>.

[L4Seval22] De Schepper, K., Albisser, O., Tilmans, O., and B. Briscoe, "Dual Queue Coupled AQM: Deployable Very Low Queuing Delay for All", TR-BB-2022-001, arXiv:2209.01078 [cs.NI], DOI 10.48550/arXiv.2209.01078, September 2022, <https://arxiv.org/abs/2209.01078>.

[L4SOPS] White, G., Ed., "Operational Guidance for Deployment of L4S in the Internet", Work in Progress, Internet-Draft, draft-ietf-tsvwg-l4sops-03, 28 April 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-l4sops-03>.

[LEDBAT_AQM] Al-Saadi, R., Armitage, G., and J. But, "Characterising LEDBAT Performance Through Bottlenecks Using PIE, FQ-CoDel and FQ-PIE Active Queue Management", IEEE 42nd Conference on Local Computer Networks (LCN), DOI 10.1109/LCN.2017.22, October 2017, <https://ieeexplore.ieee.org/document/8109367>.

[lowat] Meenan, P., "Optimizing HTTP/2 prioritization with BBR and tcp_notsent_lowat", Cloudflare Blog, October 2018, <https://blog.cloudflare.com/http-2-prioritization-with-nginx/>.

[McIlroy78] McIlroy, M.D., Pinson, E. N., and B. A. Tague, "UNIX Time-Sharing System: Foreword", The Bell System Technical Journal 57: 6, pp. 1899-1904, DOI 10.1002/j.1538-7305.1978.tb02135.x, July 1978, <https://archive.org/details/bstj57-6-1899>.

[Nadas20] Nádas, S., Gombos, G., Fejes, F., and S. Laki, "A Congestion Control Independent L4S Scheduler", ANRW '20: Proceedings of the Applied Networking Research Workshop, pp. 45-51, DOI 10.1145/3404868.3406669, July 2020, <https://doi.org/10.1145/3404868.3406669>.

[NASA04] Bailey, R., Trey Arthur III, J., and S. Williams, "Latency Requirements for Head-Worn Display S/EVS Applications", Proceedings of SPIE 5424, DOI 10.1117/12.554462, April 2004, <https://ntrs.nasa.gov/api/citations/20120009198/downloads/20120009198.pdf?attachment=true>.

[NQB-PHB] White, G. and T. Fossati, "A Non-Queue-Building Per-Hop Behavior (NQB PHB) for Differentiated Services", Work in Progress, Internet-Draft, draft-ietf-tsvwg-nqb-15, 11 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-nqb-15>.

[PRAGUE-CC] De Schepper, K., Tilmans, O., and B. Briscoe, Ed., "Prague Congestion Control", Work in Progress, Internet-Draft, draft-briscoe-iccrg-prague-congestion-control-01, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-briscoe-iccrg-prague-congestion-control-01>.

[PragueLinux] Briscoe, B., De Schepper, K., Albisser, O., Misund, J., Tilmans, O., Kühlewind, M., and A.S. Ahmed, "Implementing the 'TCP Prague' Requirements for Low Latency Low Loss Scalable Throughput (L4S)", Proceedings Linux Netdev 0x13, March 2019, <https://www.netdevconf.org/0x13/session.html?talk-tcp-prague-l4s>.

[QDyn] Briscoe, B., "Rapid Signalling of Queue Dynamics", TR-BB-2017-001, arXiv:1904.07044 [cs.NI], DOI 10.48550/arXiv.1904.07044, April 2019, <https://arxiv.org/abs/1904.07044>.

[Raaen14] Raaen, K. and T-M. Grønli, "Latency Thresholds for Usability in Games: A Survey", Norsk IKT-konferanse for forskning og utdanning (Norwegian ICT conference for research and education), 2014, <http://ojs.bibsys.no/index.php/NIK/article/view/9/6>.

[Rajiullah15] Rajiullah, M., "Towards a Low Latency Internet: Understanding and Solutions", Dissertation, Karlstad University, 2015, <https://www.diva-portal.org/smash/get/diva2:846109/FULLTEXT01.pdf>.

[RELENTLESS] Mathis, M., "Relentless Congestion Control", Work in Progress, Internet-Draft, draft-mathis-iccrg-relentless-tcp-00, 4 March 2009, <https://datatracker.ietf.org/doc/html/draft-mathis-iccrg-relentless-tcp-00>.

[RFC0970] Nagle, J., "On Packet Switches With Infinite Storage", RFC 970, DOI 10.17487/RFC0970, December 1985, <https://www.rfc-editor.org/info/rfc970>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, DOI 10.17487/RFC2698, September 1999, <https://www.rfc-editor.org/info/rfc2698>.

[RFC2884] Hadi Salim, J. and U. Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", RFC 2884, DOI 10.17487/RFC2884, July 2000, <https://www.rfc-editor.org/info/rfc2884>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3246] Davie, B., Charny, A., Bennet, J.C.R., Benson, K., Le Boudec, J.Y., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, DOI 10.17487/RFC3246, March 2002, <https://www.rfc-editor.org/info/rfc3246>.

[RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, DOI 10.17487/RFC3540, June 2003, <https://www.rfc-editor.org/info/rfc3540>.

[RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", RFC 3649, DOI 10.17487/RFC3649, December 2003, <https://www.rfc-editor.org/info/rfc3649>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[RFC4774] Floyd, S., "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field", BCP 124, RFC 4774, DOI 10.17487/RFC4774, November 2006, <https://www.rfc-editor.org/info/rfc4774>.

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

[RFC5033] Floyd, S. and M. Allman, "Specifying New Congestion Control Algorithms", BCP 133, RFC 5033, DOI 10.17487/RFC5033, August 2007, <https://www.rfc-editor.org/info/rfc5033>.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, DOI 10.17487/RFC5348, September 2008, <https://www.rfc-editor.org/info/rfc5348>.

[RFC5670] Eardley, P., Ed., "Metering and Marking Behaviour of PCN-Nodes", RFC 5670, DOI 10.17487/RFC5670, November 2009, <https://www.rfc-editor.org/info/rfc5670>.

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

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <https://www.rfc-editor.org/info/rfc6679>.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, DOI 10.17487/RFC6817, December 2012, <https://www.rfc-editor.org/info/rfc6817>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7560] Kuehlewind, M., Ed., Scheffenegger, R., and B. Briscoe, "Problem Statement and Requirements for Increased Accuracy in Explicit Congestion Notification (ECN) Feedback", RFC 7560, DOI 10.17487/RFC7560, August 2015, <https://www.rfc-editor.org/info/rfc7560>.

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <https://www.rfc-editor.org/info/rfc7567>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <https://www.rfc-editor.org/info/rfc7713>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8034] White, G. and R. Pan, "Active Queue Management (AQM) Based on Proportional Integral Controller Enhanced (PIE) for Data-Over-Cable Service Interface Specifications (DOCSIS) Cable Modems", RFC 8034, DOI 10.17487/RFC8034, February 2017, <https://www.rfc-editor.org/info/rfc8034>.

[RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <https://www.rfc-editor.org/info/rfc8170>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <https://www.rfc-editor.org/info/rfc8290>.

[RFC8298] Johansson, I. and Z. Sarker, "Self-Clocked Rate Adaptation for Multimedia", RFC 8298, DOI 10.17487/RFC8298, December 2017, <https://www.rfc-editor.org/info/rfc8298>.

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/info/rfc8311>.

[RFC8312] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", RFC 8312, DOI 10.17487/RFC8312, February 2018, <https://www.rfc-editor.org/info/rfc8312>.

[RFC8404] Moriarty, K., Ed. and A. Morton, Ed., "Effects of Pervasive Encryption on Operators", RFC 8404, DOI 10.17487/RFC8404, July 2018, <https://www.rfc-editor.org/info/rfc8404>.

[RFC8511] Khademi, N., Welzl, M., Armitage, G., and G. Fairhurst, "TCP Alternative Backoff with ECN (ABE)", RFC 8511, DOI 10.17487/RFC8511, December 2018, <https://www.rfc-editor.org/info/rfc8511>.

[RFC8888] Sarker, Z., Perkins, C., Singh, V., and M. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", RFC 8888, DOI 10.17487/RFC8888, January 2021, <https://www.rfc-editor.org/info/rfc8888>.

[RFC8985] Cheng, Y., Cardwell, N., Dukkipati, N., and P. Jha, "The RACK-TLP Loss Detection Algorithm for TCP", RFC 8985, DOI 10.17487/RFC8985, February 2021, <https://www.rfc-editor.org/info/rfc8985>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[RFC9331] De Schepper, K. and B. Briscoe, Ed., "The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9331, DOI 10.17487/RFC9331, January 2023, <https://www.rfc-editor.org/info/rfc9331>.

[RFC9332] De Schepper, K., Briscoe, B., Ed., and G. White, "Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9332, DOI 10.17487/RFC9332, January 2023, <https://www.rfc-editor.org/info/rfc9332>.

[SCReAM-L4S] "SCReAM", commit fda6c53, June 2022, <https://github.com/EricssonResearch/scream>.

[TCP-CA] Jacobson, V. and M. Karels, "Congestion Avoidance and Control", Laurence Berkeley Labs Technical Report , November 1988, <https://ee.lbl.gov/papers/congavoid.pdf>.

[UnorderedLTE] Austrheim, M., "Implementing immediate forwarding for 4G in a network simulator", Master's Thesis, University of Oslo, 2018.

Благодарности

Спасибо Richard Scheffenegger, Wes Eddy, Karen Nielsen, David Black, Jake Holland, Vidhi Goel, Ermin Sakic, Praveen Balasubramanian, Gorry Fairhurst, Mirja Kuehlewind, Philip Eardley, Neal Cardwell, Pete Heist, Martin Duke за их рецензии и комментарии. Спасибо также рецензентам от направления Marco Tiloca, Lars Eggert, Roman Danyliw, Éric Vyncke.

Работа Bob Briscoe и Koen De Schepper частично финансировалась Европейской комиссией в рамках программы Seventh Framework через проект Reducing Internet Transport Latency (RITE) (ICT-317700). Работа Koen De Schepper частично финансировалась также в рамках проектов 5Growth и DAEMON EU H2020, работа Bob Briscoe частично финансировалась также Research Council of Norway по проекту TimeIn, CableLabs и Comcast Innovation Fund. Выраженные здесь мнения принадлежат исключительно авторам.

Адреса авторов

Bob Briscoe (editor)
Independent
United Kingdom
Email: ietf@bobbriscoe.net

URI: https://bobbriscoe.net/

Koen De Schepper
Nokia Bell Labs
Antwerp
Belgium
Email: koen.de_schepper@nokia.com
URI: https://www.bell-labs.com/about/researcher-profiles/koende_schepper/

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
28911 Madrid
Spain
Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: https://www.it.uc3m.es

Greg White
CableLabs
United States of America
Email: G.White@CableLabs.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru