Network Working Group
Request for Comments: 970

John Nagle
FACC Palo Alto

December 1985

On Packet Switches With Infinite Storage

Коммутаторы пакетов с бесконечным хранилищем

Статус документа

Цель этого RFC состоит в обсуждении конкретных проблем ARPA-Internet и возможных методов их решения. На данный момент ни одно из предложенных в документе решений на рассматривается в качестве стандартов ARPA-Internet. Однако есть надежда, обрести общее согласие в части подходящего решения таких проблем, приводящее в конечном итоге к принятию стандартов. Распространение документа не ограничивается.

Аннотация

Большая часть предшествующих работ, связанных с перегрузками в системах на основе дейтаграмм, была сосредоточена на управлении буферами. Мы считаем полезным рассмотреть вариант коммутатора пакетов с бесконечным хранилищем, в котором никогда не возникает переполнения буферов. Однако и такой коммутатор может войти в насыщение и здесь рассматривается значение перегрузки с системе с бесконечным хранилищем. Продемонстрирован неожиданный результат - сеть передачи дейтаграмм с бесконечным хранилищем и очередями FIFO[1], содержащая хотя бы пару коммутаторов пакетов с конечным сроком жизни, при перегрузке будет отбрасывать все пакеты. Решая проблему перегрузки для случая бесконечного хранилища, мы открываем новые решения, применимые для коммутаторов с конечным размером хранилища.

Введение

Коммутация пакетов впервые была внедрена, когда хранилища компьютерных данных стоили на несколько порядков больше, чем сегодня. По началу были приложены значительные усилия по созданию коммутаторов пакетов с абсолютным минимумом хранилища, требуемого для работы. Проблема контроля перегрузки обычно рассматривалась как предотвращение истощения буфера в коммутаторах пакетов. Здесь представлен иной подход. Анализ начинается с допущения возможности бесконечной памяти. Это позволяет иначе взглянуть на насыщение. Больше не нужно беспокоиться о блокировке и решении вопросов по отбрасыванию пакетов, а вместо этого нужно думать, какой мы хотим видеть работу системы.

Системы на основе лишь дейтаграмм особенно подвержены перегрузкам. Механизмы блокировки, имеющиеся в системах с виртуальными каналами (устройствами), здесь отсутствуют. Достаточно эффективных решений проблемы перегрузки систем на основе дейтаграмм неизвестно. Большинство таких систем плохо ведут себя при перегрузке. Здесь показано, что можно существенно продвинуться в решении задачи контроля перегрузок для систем на основе дейтаграмм, когда ставится задача определения порядка передачи пакетов, а не выделения буферного пространства.

Коммутатор пакетов с бесконечным хранилищем

Рассмотрим сначала простой коммутатор пакетов с бесконечным хранилищем, имеющий входные и выходные каналы. Каждый канал имеет фиксированную скорость передачи данных и скорости разных каналов могут различаться. Пакеты, поступающие во входные каналы, сразу же связываются с выходными каналами через тот или иной механизм маршрутизации (не рассматривается здесь). Каждый из выходных каналов имеет очередь, из которой пакеты извлекаются и передаются в выходной канал с поддерживаемой им скоростью. Изначально предполагается, что пакеты извлекаются из очереди в порядке их включения туда (FIFO).

Предполагается, что время жизни пакетов ограничено. В протоколе DoD IP пакеты имеют поле TTL (time-to-live), указывающее число секунд, по истечении которого пакет должен отбрасываться, как не представляющий интереса[2]. При прохождении пакета через сеть это поле декрементируется и, по достижении 0, пакет должен отбрасываться. Начальное значение поля фиксировано и в протоколе DoD IP по умолчанию устанавливается 15.

Механизм TTL предотвращает неограниченный рост очередей - когда очередь становится достаточно длинной, пакеты просто устаревают до их отправки. Это задаёт верхнюю границу для общего размера всех очередей, определяемою суммарной скоростью всех входных каналов и верхним пределом TTL. Однако это не предотвращает насыщение.

Рассмотрим простой узел с одним входящим и одним исходящим каналом. Предположим, что скорость прибытия превышает скорость отправки. Размер очереди исходящего канала будет расти, пока время прохождения через очередь не превысит TTL входящих пакетов. С этого момента при обработке выходной очереди будут встречаться пакеты с нулевым значением TTL, которые будут отбрасываться с переходом к следующему пакету из очереди. Пакеты с ненулевым значением TTL передаются в исходящий канал.

Фактически передаются лишь пакеты с ненулевым значением TTL. Как только будет достигнуто установившееся состояние с перегрузкой, число передаваемых пакетов будет мало, поскольку пакет будет находиться в очереди чуть меньше максимального значения TTL. Фактически, если скорость отправки больше одного пакета за интервал TTL, значение TTL любого передаваемого пакета будет 1. Это следует из наблюдения, что при передаче более 1 пакета за интервал TTL, последовательные пакеты из очереди будут иметь значения TTL, отличающиеся не более чем на 1. Таким образом при работе компонента коммутатора, извлекающего пакеты из очереди, он будет отбрасывать устаревшие пакеты (с нулевым или отрицательным TTL) и отправлять пакеты с TTL=1.

[1] First-in-first-out - первый на входе - первый на выходе.
[2] В современных сетях значение TTL уменьшается не по времени, а по числу пересылок, поэтому пакеты не будут устаревать просто из-за долгого пребывания в очереди. *Прим. перев.*

Итак, достаточно ясно, что на следующий узел сети с коммутацией пакетов они будут прибывать с TTL=1. Поскольку значение TTL всегда сокращается по меньшей мере на 1 при прохождении пакета через сеть, этот узел будет декрементировать TTL до 0 и затем отбрасывать пакет.

Таким образом, мы показали, что сеть с бесконечной очередью FIFO и конечным значением TTL при перегрузке будет отбрасывать все пакеты. Это довольно неожиданный, но вполне реальный результат. Отбрасывание всех пакетов не является следствием допущения о бесконечном размере буфера. Проблема возникает и в сетях с конечным размером хранилища, но эффект менее очевиден. Известно, что сети на основе дейтаграмм плохо ведут себя при перегрузке, но анализ их поведения не был проведён ранее. Для случая с бесконечным буфером такой анализ достаточно прост и даёт хорошее представление проблемы.

Можно было ожидать обнаружение этого эффекта ранее, но предшествующие работы по контролю перегрузок были сосредоточены на управлении буферами. Анализ варианта в бесконечным буфером был, по-видимому, проведён автором впервые.

Этот результат напрямую применим и к сетям с конечными ресурсами. Размер хранилища, требуемый для реализации коммутатора, у которого буферы никогда не переполняются, оказывается вполне разумным. Рассмотрим коммутатор дейтаграмм для сети, подобной ARPANET. Для коммутатора пакетов с 4 каналами 56 Кбит/с и верхним пределом времени жизни пакетов в 15 секунд максимальный размер буфера, который может потребоваться, составляет 420 Кбайт[1]. Коммутатору с таким, достаточно скромным, размером буфера не потребуется отбрасывать пакеты из-за истощения буфера.

Эта проблема не является сугубо теоретической и была продемонстрирована в экспериментальной сети с коммутатором на основе supermini с несколькими мегабайтами памяти. Это показало, что описанные выше явления встречаются на практике. Первый эксперимент с применением Ethernet на одной стороне коммутатора и линии 9600 Бод на другой привёл к тому, что в пике коммутатор буферизовал 916 дейтаграмм IP. Однако нагрузка обеспечивалась в транспортном соединении TCP и возникал тайм-аут транспортного соединения до достижения предела времени жизни в очереди из-за большого времени кругового обхода (round trip), поэтому фактически не возникало стабильного состояния с максимальной длиной очереди, как предсказывает приведённый выше анализ. Это условие можно создать принудительно с точки зрения пользовательского приложения на основе транспорта TCP и это заслуживает дальнейшего анализа.

Взаимодействие с транспортными протоколами

До сих пор предполагалось, что источник передаёт пакеты с фиксированной скоростью. Это справедливо для некоторых источников, таких как как пакетные системы голосовой связи. Системы, использующие протоколы класса ISO TP4 или DoD TCP, должны вести себя лучше. Ключевым моментом является то, что транспортные протоколы в системах на основе дейтаграмм должны вести себя так, чтобы сеть не перегружалась, даже если в ней нет средств предотвращения перегрузок. Это вполне возможно. В предшествующей работе автора [1] разъясняется поведение транспортного протокола TCP в сети с перегрузкой. Там показано, что реализация транспортного протокола с некорректным поведением может нанести серьёзный вред сети на основе дейтаграмм и рассмотрено желаемое поведение реализаций. В этой статье предложено несколько конкретных рекомендаций в части поведения реализаций TCP и показано, что корректное поведение в некоторых случаях может снижать загрузку сети на порядок. Выводы этой статьи заключаются в том, что транспортному протоколу для обеспечения надлежащего поведения не следует использовать время повторной передачи меньше интервала кругового обхода между вовлечёнными хостами, а при получении из сети информации о перегрузке протоколу следует принять меры по сокращению числа пакетов, остающихся в соединении.

Ссылка на предыдущую работу приведена для указания того, что нагрузка вносимая транспортным протоколом в сеть, может не фиксироваться спецификацией протокола. Некоторые из имеющихся реализаций транспортных протоколов работают хорошо, другие - нет. Имеющиеся реализации TCP достаточно сильно различаются. Есть основания предполагать, что реализации ISO TP4 будут более однородными в соответствии с более жёсткой спецификацией, но в стандарте TP4 имеется достаточно открытых мест, допускающих значительную вариативность. Можно предположить, что будут маргинальные с точки зрения сети реализации TP4, как уже имеются такие реализации TCP. Такие реализации могут работать достаточно хорошо, пока они не столкнутся с тяжело нагруженной сетью с большими задержками. Тогда будет видно, какие из них корректно ведут себя.

Даже при достаточно хорошем поведении всех хостов возможны неполадки. Каждый хост обычно может получить больше пропускной способности сети, передавая больше пакетов за единицу времени, поскольку стратегия FIFO отдаёт больше ресурсов отправителю большего числа пакетов. Но по мере загрузки сети общая пропускная способность падает и, как показано выше, может упасть до 0. Таким образом, оптимальная для отдельного хоста стратегия сильно неоптимальна для сети в целом.

Аспекты теории игр для насыщения сети

С точки зрения теории игр сети на основе дейтаграмм напоминают проблему нестабильности многопользовательских игр. Системы, где оптимальная стратегия каждого игрока неоптимальна для всех игроков, стремятся к неоптимальному состоянию. Примером такого свойства в теории игр является хорошо известная дилемма заключённого. Но более близкой аналогией является проблема общего достояния в экономике. Там, где каждый может улучшить своё благосостояние, используя больше бесплатных ресурсов, общее число ресурсов сокращается по мере роста популяции и личные интересы приводят к истощению ресурсов и коллапсу. Исторически этот анализ применялся к использованию пастбищных угодий, но он применим и к таким ресурсам, как качество воздуха и системы распределения времени. В общем случае опыт показывает, что системы с множеством игроков с таким типом нестабильности имеют тенденцию возникновения серьёзных проблем.

Решения проблемы общего достояния делятся на три класса: кооперативные, авторитарные и рыночные. Кооперативные решения, конда все согласны вести себя хорошо, подходят для небольшого числа игроков, но обычно перестают работать с ростом числа участников. Авторитарные решения эффективны, если можно легко отслеживать поведение каждого, но обычно сталкиваются с проблемами, когда определение хорошего поведения является тонким

[1]Размер буфера для 1 соединения Ethernet 10 Мбит,с с верхней границей TTL 255 составляет 318 миллионов байтов. *Прим. редакт.*

делом. Рыночное решение возможно при условии изменения правил игры так, чтобы оптимальная для каждого стратегия вела к ситуации, оптимальной для всех. Там, где это возможно, рыночное решение весьма эффективно.

Приведённый выше анализ обычно применим к игрокам-людям. В случае с сетью игроками являются компьютеры, исполняющими запрограммированную стратегию. Само по себе это не гарантирует хорошего поведения, стратегия компьютера может быть запрограммирована на оптимизацию его производительности без учёта соображений сети. Аналогичная ситуация наблюдается для телефонных устройств повторного набора номер, где пользовательское устройство пытается повысить вероятность своего успеха в перегруженной сети за счёт быстрого повтора набора номера при отказе в сети. Поскольку ресурсы систем организации телефонных соединений ограничены, такое поведение может существенно влиять на сеть и в некоторых странах применение таких устройств запрещено (например, в Бразилии). Это административное решение иногда эффективно, иногда нет, в зависимости от административной силы запрещающего органа и пользователей.

По мере коммерциализации транспортных протоколов и появления конкурирующих систем следует ожидать попыток настроить протокол оптимально с точки зрения одного хоста, но неоптимально для сети в целом. Это уже проявилось в реализации транспортного протокола одного популярного производитель рабочих станций.

Возвращаясь к анализу сети сетей на основе дейтаграмм, ясно, что авторитарное решение указало бы всем хостам «вести себя хорошо», но это может оказаться сложным, поскольку определение хорошего поведения хоста является достаточно сложным. В кооперативном решении возникает та же проблема, наряду со сложностью применения социального давления в распределенной системе. Рыночное решение требует платы за хорошее поведение и изменения правил игры.

Беспристрастность в системах коммутации пакетов

Хотелось бы защитить сеть от хостов с плохим поведением. Точнее, хотелось бы при наличии хостов с разным поведением обеспечить лучшее обслуживание хостам с подобающим поведением. Средства для этого разработаны.

Рассмотрим сеть, состоящую из широкополосных ЛВС на основе дейтаграмм без управления потоком данных (к этому классу относятся Ethernet и большинство дейтаграммных систем IEEE 802.x, независимо от определения несущей или передачи маркеров), подключённым к этим ЛВС хостов и соединяющих ЛВС распределенной сети на основе коммутаторов пакетов и протяжённых каналов. В распределенной сети может применяться внутреннее управление потоками данных, но нет возможности внести обязательное управление потоком на передающих хостах. Этой модели соответствуют сети DoD Internet, сети сетей Xerox Network Systems и созданные на их основе системы.

Если какой-либо из хостов локальной сети создает пакеты, маршрутизируемые в распределенную сеть со скоростью, превышающей возможности этой сети, возникает перегрузка в коммутаторе, соединяющем эту ЛВС с распределенной сетью. Если очереди коммутаторов пакетов работают строго на основе модели FIFO, хост с плохим поведением будет препятствовать передаче данных хостами с хорошим поведением.

Здесь вводится концепция беспристрастности (fairness) коммутатора пакетов, чтобы каждый хост-источник мог получить свю долю ресурсов на каждом коммутаторе пакетов. Это можно сделать, заменив одну очередь FIFO на каждом выходном канале несколькими очередями (по одной для каждого источника трафика в сети). Эти очереди перебираются по кругу (round-robin) с извлечением по одному пакету с неистексим сроком жизни из каждой непустой очереди для передачи в связанный выходной канал и отбрасыванием просроченных пакетов. Пустые очереди просто пропускаются.

Этот механизм беспристрастен и пропускная способность выходного канала делится между хостами-источниками. Каждый такой хост с пакетами в очереди коммутатора для указанного выходного канала, получает возможность передать в точности 1 пакет за каждый цикл обхода очередей. Это является одной из форм балансировки нагрузки.

Система также улучшается с точки зрения теории игр. Оптимальной стратегией данного хоста больше не является отправка как можно большего числа пакетов. Сейчас оптимально передавать пакеты со скоростью, при котором в каждом коммутаторе остаётся 1 пакет, ожидающий отправки, поскольку в этом случае хост будет обслуживаться в каждом цикле кругового перебора и пакеты хоста будут сталкиваться с минимальной транзитной задержкой. Такая стратегия вполне приемлема и с точки зрения сети, поскольку каждая очередь будет содержать 1 или 2 пакета.

Хостам нужны рекомендации из сети для оптимизации их стратегии. Имеющийся в DoD IP механизм Source Quench для этого вполне подходит, не смотря на его минималистичность. Коммутаторам пакетов следует передавать сообщение Source Quench хосту-отправителю, когда число пакетов в очереди для этого хоста превышает некоторое небольшое значения, возможно, 2. Если хост поддерживает свой трафик чуть ниже этого порога, сети следует работать со средней очередью для каждого хоста менее 2.

Хосты с плохим поведением могут передать все дейтаграммы, которые они хотят, но это не увеличит получаемую таким хостом долю ресурсв сети. Они добьются лишь того, что их пакеты будут проходить через сеть с большой транзитной задержкой. Хост с достаточно плохим поведением может передать довольно много дейтаграмм, увеличив до предела времени жизни транзитную задержку своих пакетов в сети и ни одна из его дейтаграмм не пройдёт. Это произойдёт быстрее при использовании беспристрастных очередей, нежели FIFO, поскольку хосты с плохим поведением будут получить лишь часть пропускной способности, обратно пропорциональную числу хостов, использующих в данный момент коммутатор пакетов. Это много меньше доли, которую хост получил бы в прежней системе, где более активным хостам доставалось больше пропускной способности. Беспристрастное распределение ресурсов служит хорошим стимулом для улучшения поведения хостов.

Следует отметить, что вредоносные хосты, в отличие от просто плохо работающих, могут перегружать сеть, используя множество адресов отправителя в своих дейтаграммах и выдавая себя за множество хостов для получения большей доли пропускной способности сети. Это является атакой на сеть и маловероятной случайное возникновение такой атаки. Таким образом, это является проблемой безопасности и не рассматривается здесь.

Хотя коммутаторы пакетов сделаны беспристрастными, это не обеспечивает беспристрастности всей сети и в этом состоит слабость подхода. Описанная здесь стратегия лучше всего применима к коммутатору пакетов в узкой точке сети, такой как входной узел распределенной сети или межсетевой шлюз. В качестве стратегии для промежуточного узла большой сети с коммутацией пакетов, через который проходят дейтаграммы от хостов разных частей сети, этот

подход не так хорош. Автор не утверждает, что описанное здесь решение проблемы перегрузки в сетях на основе дейтаграмм является полным. Однако описанный подход решает достаточно серьёзную проблему и задаёт направление для будущей работы над более общим решением.

Реализация

На первый взгляд поддержка отдельной очереди для каждого хоста-источника на каждом выходном канале каждого коммутатора пакетов значительно усложняет механизм очередей в коммутаторах пакетов. Это, действительно, сопряжено с некоторыми сложностями, но значительно проще, например, манипуляций со сбалансированными бинарными деревьями. Одна простая реализация включает предоставление пространства для указателей как части заголовка каждого буфера дейтаграммы. Очередь для каждого хоста-источника требует лишь одной привязки, а начало очередей (первый буфер каждой очереди) должны иметь двойную привязку, чтобы можно было удалить всю очередь, когда она пуста. Таким образом, для каждого буфера нужны 3 указателя. Для ускорения процесса при длинных очередях могут применяться более сложные стратегии, но такие сложности вряд ли оправданы практически.

С учётом конечного размера буфера можно столкнуться с его истощением (заполнением). В таком случае следует отбросить пакет в конце самой диннной очереди, поскольку именно он бвл бы передан последним. Это, конечно, неблагоприятно для хоста с наибольшим число дейтаграмм в сети, но соответствует цели обеспечения беспристрастности.

Заключение

Отказавшись от сложившейся исторически привязки к управлению буферами, мы получаем некоторые новые представления о контроле перегрузок в системах на основе дейтаграмм и решение для некоторых известных проблем реальных систем. Есть надежда, что другие на основе этого нового представления продолжат добиваться реального продвижения в решении общей задачи контроля перегрузки в сетях на основе дейтаграмм.

Литература

[1] Nagle, J. "Congestion Control in IP/TCP Internetworks", ACM Computer Communications Review, October 1984.

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru