Network Working Group
Request for Comments: 2992
Category: Informational

C. Hopps
NextHop Technologies

November 2000

Анализ алгоритма равноценных путей

Analysis of an Equal-Cost Multi-Path Algorithm

Статус документа

Этот документ содержит информацию для сообщества Internet и не задаёт каких-либо стандартов Internet. Документ можно распространять без ограничений.

Авторские права

Аннотация

ECMP[1] представляет собой метод маршрутизации для передачи пакетов по множеству равноценных путей. Машина пересылки идентифицирует пути по следующему интервалу (next-hop). При пересылке пакета маршрутизатор должен выбрать используемый путь (next-hop). В этом документе приведён анализ одного из методов принятия таких решений. Анализ учитывает производительность алгоритма и нарушения, вызываемые изменением набора next-hop.

1. Хэш-порог

Одним из методов выбора следующего этапа пересылки (next-hop) при маршрутизации с ECMP является hash-threshold. Маршрутизатор сначала выбирает ключ путём хэширования (например, CRC16) полей заголовка пакета, идентифицирующих поток. Для N значений next-hop назначаются уникальные области (region) в пространстве ключей. Маршрутизатор использует ключ для определения области и выбора используемого next-hop.

В качестве примера использования хэш-порога рассмотрим случай, когда маршрутизатор при получении пакета выполняет расчёт CRC16 для полей заголовка, определяющих поток (например, адресов отправителя и получателя пакета) и результат служит ключом. Предположим, что для этого адресата имеется 4 варианта next-hop, каждому из которых назначена область в 16-битовом пространстве ключей. Для равномерного использования маршрутизатор может поделить пространство равномерно, чтобы каждая область имела размер 65536/4 или 16K. Значение next-hop выбирается путём определения области, содержащей ключ (т. е. результат CRC).

2. Анализ

Имеется несколько проблем при выборе алгоритма определения используемого next-hop. Одной из них является производительность, связанная со сложностью расчётов алгоритма. Другая проблема связана с изменением пути, используемого для потока. Третьей проблемой является балансировка, однако по причине тесной связи характеристик алгоритма балансировки с выбранной хэш-функцией в этом документе не делается попытки глубокого анализа.

При анализе предполагается использование областей одного размера. Если результат хэш-функции имеет равномерное распределение, потоки между путями также будут распределяться равномерно и алгоритм обеспечит подобающую реализацию ECMP. Можно реализовать неравнозначное распределение по путям, задав области разного размера, однако это выходит за рамки документа.

2.1. Производительность

Производительность алгоритма hash-threshold определяется тремя компонентами - выбор областей для next-hop, расчёт ключей и сопоставление ключа с областями для выбора используемого next-hop.

Алгоритм не задаёт хэш-функцию, применяемую для создания ключа. Производительность выбранной функции будет напрямую влиять на общую производительность алгоритма. Предполагается возможность реализации хэш-функции на аппаратном уровне параллельно с другими операциями, которые нужно выполнить до выбора next-hop.

Поскольку области имеют равный размер, расчёт их границ является тривиальной задачей. Каждая граница отделена от предыдущей точно на размер области, начиная со значения 0 для первой области. Как будет показано ниже, при использовании областей одного размера не требуется хранить значения их границ.

Для выбора next-hop требуется определить область, к которой относится ключ. Поскольку области имеют одинаковый размер, это можно сделать путём простого деления.

```
regionsize = keyspace.size / #{nexthops}
region = key / regionsize;
```

Таким образом, время на поиск next-hop зависит от способа хранения значений next-hop в памяти. Обычно используется массив, индексируемый по областям O(1).

2.2. Нарушения в работе потоков

Протоколы типа TCP работают лучше, если путь потока не изменяется в процессе работы соединения. Степень нарушения измеряется числом потоков, пути которых меняются в результате тех или иных изменений в маршрутизаторе. Будем измерять степень нарушения долей числа потоков, чьи пути поменялись в ответ на некие

[1]Equal-cost multi-path - множество равноценных путей.

изменения в маршрутизаторе. Это может оказаться важным при наличии флуктуаций (flapping) на одном или нескольких путях. Описание влияния таких нарушений на работу протоколов типа TCP приведено в работе [1].

Некоторые алгоритмы типа кругового обхода (round-robin, когда при получении пакета для него выбирается значение next-hop, которое использовалось наиболее давно) нарушают потоки независимо от каких-либо изменений в маршрутизаторе. Ясно, что для hash-threshold этого не происходит. Пока границы областей не меняются, для данного потока будет выбираться одно и то же значение next-hop.

Поскольку области должны иметь одинаковый размер, единственным случаем изменения границ является изменение числа next-hop. В этом случае области будут расти или сокращаться. Начнём анализ с примеров.

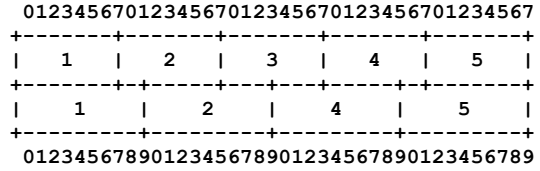

Рисунок 1. До и после удаления области 3.

На рисунке 1 удаляется область 3. Оставшиеся области расширяются и сдвигаются для заполнения всего пространства. В данном случае четверть области 2 переходит в область 1, половина (две четверти) области 3 переходит в область 2, вторая половина - в область 4, а четверть области 4 в область 5. Поскольку каждая из исходных областей представляет 1/5 потоков, суммарное нарушение составит 1/5*(1/4 + 1/2 + 1/2 + 1/4) или 3/10.

Отметим, что нарушение потоков при добавлении области будет эквивалентно возникающему при удалении. Т. е. при переходе от N областей к N-1 изменится такая же часть потоков, как при переходе от N-1 к N.

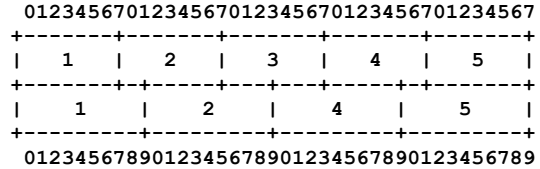

Рисунок 2. До и после удаления области 4.

На рисунке 2 показано удаление области 4. Здесь оставшиеся области снова расширяются и смещаются. Четверть области 2 переходит в область 1, половина области 3 - в область 2, три четверти области 4 - в область 3 и четверть области 4 - в область 5. Поскольку каждая исходная область представляет пятую часть потоков, общее нарушение составит 7/20.

Для обобщения отметим, что при удалении области K оставшиеся N-1 области растут для заполнения освободившегося пространства 1/N. Этот рост делится на N-1 область и размер каждой области меняется на 1/N/(N-1) или 1/(N(N-1)). Это изменение приводит также к сдвигам областей, не расположенных по краям. Первая область расширяется, вторая в дополнение к этому смещается в направлении K на величину изменения первой области. 1/(N(N-1)) потоков из области 2 учитываются в изменении размера области 1. 2/(N(N-1)) потоков из области 3 учитываются в области 2. Это обусловлено тем, что область 2 смещается на 1/(N(N-1)) и расширяется на 1/(N(N-1)). Этот процесс продолжается с обеих сторон для границ области K. Расчёт числа потоков, переходящих из области K в соседние области учитывает удаление области K. Выражения для расчёта приведены ниже.

$$\text{нарушение} = \sum_{i=1}^{K-1} \frac{i}{(N)(N-1)} + \sum_{i=K+1}^{N} \frac{(i-K)}{(N)(N-1)}$$

Коэффициент 1/((N)(N-1)) можно вынести за скобки

$$= \frac{1}{(N)(N-1)} \left(\sum_{i=1}^{K-1} i + \sum_{i=K+1}^{N} (i-K) \right)$$

Рассчитаем сейчас значения обеих сумм. Первая сумма даёт значение (K)(K-1)/2. Для второй суммы отметим, что складываются целые числа от 1 до N-K, что даёт в результате (N-K)(N-K+1)/2.

$$= \frac{(K-1)(K) + (N-K)(N-K+1)}{2(N)(N-1)}$$

Результаты суммирования показывают, что минимальное нарушение потоков возникает в том случае, когда K находится ближе всего к середине между 1 и N. Это можно доказать, находя минимум конкретного выражения для K при постоянном N. Преобразуем выражение как показано ниже

$$= \frac{2K*K - 2K - 2NK + N*N + N}{2(N)(N-1)}$$

$$= \frac{K*K - K - NK}{(N)(N-1)} + \frac{N + 1}{2(N-1)}$$

С учётом поиска минимума по K постоянную правую часть выражения (N+1)/2(N-1) можно отбросить вместе с неизменным знаменателем (N)(N-1).

$$\frac{d}{} (K*K - (N+1)K)$$

$$\frac{dk}{} = 2K - (N+1)$$

Очевидно, что последнее выражение даст 0 для K = (N+1)/2.

В последнем выражении предполагается целочисленное значение K. При нечётном N выражение (N+1)/2 даёт целое число, тогда как при чётном N выражение (N+1)/2 будет давать целое число + 1/2. Минимальные нарушения в последнем случае будут при K, равном N/2 или N/2 + 1.

Поскольку выражение является квадратичным с глобальным минимумом посередине между 1 и N, наибольшие нарушения будут возникать при удалении крайних областей 1 и N и оно составит 1/2.

Минимально возможное нарушение будет наблюдаться при K=(N+1)/2 и составит 1/4 + 1/(4*N). Диапазон возможных значений составит (1/4, 1/2].

Для минимизации вносимых нарушений рекомендуется добавлять новые значения в середину, а не по краям.

3. Сравнение с другими алгоритмами

Существуют и другие алгоритмы выбора следующего интервала (next-hop) с другими параметрами производительности и вносимых нарушений. Из этих алгоритмов рассмотрим лишь те, которые не вносят нарушений по своей природе (т. е. при отсутствии изменений в наборе next-hops пути потоков не меняются). Это явно исключает перебор по кругу (round-robin) и случайный выбор. Рассмотрим алгоритмы modulo-N и HRW[1].

Алгоритм Modulo-N является «простейшей» формой hash-threshold. Для N вариантов next-hop используется хэш-функция полей заголовка, определяющих поток и результат приводится к модулю N. Это обеспечивает прямое отображение на один из вариантов next-hop. Modulo-N является одним из вносящих наибольшие нарушения алгоритмов и при удалении или добавлении next-hop нарушено будет (N-1)/N потоков. Производительность Modulo-N эквивалентна производительности hash-threshold.

Метод HRW является сравнительным и в некотором смысле похож на hash-threshold с областями разных размеров. Для каждого варианта next-hop маршрутизатор генерирует псевдослучайное значение на основе полей заголовка, описывающих поток и следующего интервала, это значение задаёт вес для next-hop. Для передачи выбирается вариант next-hop с наибольшим весом. Преимуществом HRW являются минимальные нарушения (в результате добавления или удаления next-hop доля нарушений составляет 1/N), а недостатком - большие по сравнению с hash-threshold издержки на выбор next-hop. Описание HRW и его сравнение с другими методами дано в работе [2]. Пример использования HRW (не для выбора next-hop) приведён в [3].

Поскольку каждый из трёх отмеченных методов (modulo-N, hash-threshold и HRW) требует хэширования полей заголовка, определяющих поток, влияние функции хэширования на производительность не учитывалось при сравнении. Если функцию хэширования можно сделать «недорогой» (например, за счёт аппаратной реализации), её также следует принимать во внимание при использовании любого из трёх методов.

Производительность поиска для hash-threshold, как и для подобно modulo-N, составляет O(1), для HRW - O(N).

Нарушение поведения потоков ведёт себя обратно по отношению к производительности методов. HRW обеспечивает наименьшее нарушение 1/N, hash-threshold - от 1/4 до ½, Modulo-N - (N-1)/N.

Если сложность метода HRW не является препятствием, его следует рассматривать в качестве альтернативы hash-threshold. Это может быть, например, в случаях когда сохраняется состояние по потокам и выбор next-hop происходит нечасто.

Однако в тех случаях, когда издержки HRW слишком велики, метод hash-threshold обеспечивает преимущество по сравнению с modulo-N, поскольку производительность методов одинакова, а нарушений он создаёт меньше.

4. Вопросы безопасности

Этот документ содержит анализ алгоритма, используемого для реализации ECMP-маршрутизации. Анализ не оказывает непосредственного влияния на проблемы безопасности инфраструктуры Internet.

5. Литература

[1] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast", RFC 2991, November 2000.

[2] Thaler, D. and C.V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates", IEEE/ACM Transactions on Networking, February 1998.

[3] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P. and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.

6. Адрес автора

Christian E. Hopps

NextHop Technologies, Inc.

517 W. William Street

Ann Arbor, MI 48103-4943

U.S.A

Phone: +1 734 936 0291

EMail: chopps@nexthop.com

[1]Highest random weight - максимальный случайный вес.

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

7. Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.