Network Working Group
Request for Comments: 2439
Category: Standards Track

C. Villamizar
ANS
R. Chandra
Cisco
R. Govindan
ISI

November 1998

Демпфирование осцилляций маршрутов BGP

BGP Route Flap Damping

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Авторские права

Тезисы

Описан способ использования протокола BGP, позволяющий снизить объем маршрутного трафика, передаваемого партнерам и, следовательно, уровень нагрузки на этих партнеров без существенного воздействия на время схождения для относительно стабильных маршрутов. Этот метод был реализован в коммерческих приложениях, поддерживающих протокол BGP. Метод применим также к IDRP.

Основными целями являются:

- обеспечение механизма, позволяющего снизить нагрузку на маршрутизатор, вызванную нестабильностью;
- предотвращение устойчивых осцилляций маршрутов;
- выполнение предыдущих операций без существенного роста времени схождения для стабильных маршрутов.

При этом должны приниматься во внимание другие задачи протокола BGP:

- упаковка данных об изменениях в небольшое число обновлений;
- сохранение согласованной маршрутизации;
- минимизация дополнительного расхода ресурсов.

Избыточная скорость обновления анонсов доступности для подмножества префиксов Internet часто встречается в сети Internet. Это было отмечено еще в начале 1990 годов многими людьми, вовлеченными поддержку работы Internet. Для обозначения этого эффекта используется неформальный термин route flap. Описанный здесь метод в настоящее время широко распространен и обычно обозначается термином route flap damping[1].

1 Обзор

Для обеспечения масштабируемости системы маршрутизации необходимо снизить количество изменений состояния маршрутизации, распространяемых посредством BGP, что позволит снизить нагрузку на маршрутизаторы. Основным вкладом в нагрузку, создаваемую обработкой обновлений BGP, вносит процесс выбора маршрутов BGP (decision process) вкупе с добавлением и удалением записей в таблицах пересылки.

Рассмотрим простой пример. В популярных реализациях BGP могут возникать отказы, связанные с высоким уровнем маршрутных обновлений. Например, при достаточно высоком уровне нагрузки может оказаться невозможной поддержка сессий BGP или IGP. Отказ одного маршрутизатора может привести к дополнительному росту нагрузки на другие маршрутизаторы. Эта дополнительная нагрузка может привести к отказу других экземпляров той же реализации протокола или других реализаций с похожими недостатками. В тяжелых случаях могут возникать стабильные осцилляции маршрутов. Подобные ситуации уже наблюдались на практике.

Реализация BGP должна быть готова к обработке больших объемов маршрутных данных. Реализации протокола BGP не следует надеяться на то, что отправитель защитит ее от нестабильности маршрутов. Разработаны рекомендации по предотвращению устойчивых осцилляций, но эти рекомендации не избавляют от необходимости разработки эффективных и устойчивых реализаций протокола. Описанный здесь механизм позволяет сдерживать нестабильность маршрутов на граничных маршрутизаторах AS.

[1]Демпфирование осцилляций маршрутов.

Даже в устойчивых реализациях BGP производительность обработки маршрутов не является безграничной. Ограничение распространения ненужных изменений является частью задачи обеспечения разумного времени схождения при обработке маршрутных изменений в условиях роста числа маршрутов.

2 Методы ограничения маршрутных анонсов

Здесь описываются два метода контроля частоты маршрутных анонсов. Первый метод включает таймеры с фиксированными значениями. Этот метод не требует дополнительных ресурсов, но увеличивает время схождения для нормальных ситуаций, когда маршруты не имеют нестабильностей. Второму методу эти ограничения не присущи, но он требует для реализации некоторых ресурсов небольшой объем данных по состояниям маршрутов и совсем незначительное увеличение нагрузки на процессор при обработке).

Возможно и желательно использовать комбинации обоих методов. На практике таймеры с фиксированным значением устанавливаются на очень короткое время и полезны для упаковки маршрутов в меньшее число обновлений для тех случаев. Когда маршруты приходят в виде отдельных обновлений. Протокол BGP называет это упаковкой NLRI[1] [5].

Изредка таймеры с фиксированным значением устанавливают на время от десятков минут до нескольких часов – это может потребоваться для демпфирования осцилляций маршрутов. Однако такая установка будет давать побочный в виде замедления процесса схождения при выборе маршрутов.

2.1 Рекомендации по установке фиксированных значений для таймеров

BGP-3 не включает конкретных рекомендаций по этому вопросу [1]. Короткая глава Frequency of Route Selection[2] просто рекомендует что-нибудь делать и содержит рассуждения общего плана о желательных и нежелательных свойствах.

BGP4 сохраняет главу Frequency of Route Advertisement[3] и добавляет главу Frequency of Route Origination[4]. BGP-4 описывает метод ограничения частоты маршрутных анонсов, включающий таймеры с фиксированными значениями MinRouteAdvertisementInterval (задается в конфигурации) и MinASOriginationInterval [5]. Рекомендуемое значение для таймера MinRouteAdvertisementInterval составляет 30 секунд, для MinASOriginationInterval - 15 секунд.

2.2 Желаемые свойства алгоритмов демпфирования

Прежде, чем описывать алгоритм демпфирования осцилляций, нужно ясно определить цели. Для обоснования алгоритма были проведены проверки некоторых ключевых свойств..

Основной задачей является снижение нагрузки, вызванной маршрутными обновлениями, без влияния на время схождения для хорошо себя ведущих маршрутов. Для решения этой задачи нужно определить что такое маршруты с “хорошим” и “плохим” поведением. Должен быть также определен алгоритм, позволяющий идентифицировать маршруты с плохим поведением. В идеальном случае этот параметр мог бы служить для прогнозирования стабильности маршрута.

Любые задержки распространения маршрутов с хорошим поведением следует минимизировать. Некоторые задержки вполне допустимы в целях лучшей упаковки обновлений. Задержки для маршрутов с плохим поведением по возможности следует делать пропорциональными мере ожидаемой стабильности маршрута. Задержки при распространении нестабильных маршрутов должны вести к демпфированию таких маршрутов до тех пор, пока не будет достигнута некоторая уверенность в стабилизации маршрута.

Если за короткий промежуток времени получено множество маршрутных изменений в виде отдельных обновлений и эти обновления потенциально могут быть объединены, такое объединение следует выполнить с максимально возможной эффективностью прежде, чем распространять их дальше. Незначительная задержка в распространении маршрутов с хорошим поведением вполне допустима и даже необходима в целях лучшей упаковки обновлений.

Для нестабильных маршрутов следует подавлять их анонсирование, а не удаление. В тех случаях, когда к одному адресату ведет стабильный маршрут и имеется также нестабильный маршрут, возможно, что демпфирование нестабильного маршрута следует делать более интенсивным, нежели в случаях отсутствия альтернативного пути.

Согласованность картины маршрутизации внутри AS имеет очень важное значение. Задержки в распространении данных IBGP следует делать минимальными. Согласованность картины маршрутизации между AS также достаточно важна. Весьма нежелательно анонсировать маршрут, отличающийся от того, который будет использоваться, за исключением очень коротких интервалов времени. Более важно подавить восприятие маршрута (и, следовательно, его дальнейшее распространение в IGP), нежели подавлять только дальнейшее распространение.

Очевидно, что точное предсказание будущей стабильности маршрута не представляется возможным. Ретроспектива стабильности маршрута в общем случае может служить хорошей основой для оценки перспектив его стабильности. Критерии отличия маршрутов с хорошим поведением от маршрутов с плохим поведением основываются, следовательно, на истории состояния стабильности маршрута в недавнем прошлом. Не существует простого количественного выражения истории стабильности маршрута, поэтому требуется определить показатель качества[5] (figure of merit). Некоторыми желательными характеристиками уровня нестабильности может быть учет времени существования нестабильности (более давние нестабильности оказывают меньшее влияние на уровень) и кумулятивный эффект (уровень нестабильности учитывает всю предысторию, а не только недавнее прошлое).

Алгоритму следует вести себя так, чтобы для стабильных маршрутов маршрутов с незначительным количеством переходов в нестабильное состояние такие переходы происходили быстро. Если переходы в нестабильное состояние продолжаются, анонсирование маршрута следует подавлять. Следует сохранять некоторую информацию о предшествующих случаях нестабильности. Степень влияния прежних случаев нестабильности следует постепенно уменьшать по мере того, как маршрут продолжает анонсироваться в качестве стабильного.

---

[1] Network Layer Reachability Information – информация о доступности на сетевом уровне.
[2] Частота выбора маршрутов.
[3] Частота анонсирования маршрутов.
[4] Частота порождения маршрутов.
[5] По-русски будет лучше сказать уровень нестабильности и мы будет в дальнейшем использовать именно такой термин. *Прим. перев.*

2.3 Выбор алгоритма

После того, как маршрут воспринят, его реанонсирование будет задержано на короткое время для более эффективной упаковки обновлений. Для внешних маршрутов время такой задержки может быть более продолжительным. Длительность задержки определяется на основе уровня нестабильности, который предполагается коррелирующим с вероятностью будущей нестабильности маршрута. Маршруты с большим значением уровня нестабильности будут подавляться. Для уменьшения значения уровня нестабильности с течением времени выбран закон экспоненциального спада. Этот выбор следует рассматривать как предложение для разработчиков.

Функция экспоненциального спада позволяет сохранять информацию о предшествующих фактах нестабильности в течение достаточно долгого времени. Скорость уменьшения значения уровня нестабильности при использовании этой функции с течением времени падает. Функция экспоненциального спада имеет следующее свойство:

```
f(f(figure-of-merit, t1), t2) = f(figure-of-merit, t1+t2)
```

Это свойство позволяет в один прием вычислить уменьшение значения за длительный период независимо от текущего значения уровня нестабильности (figure-of-merit). В целях оптимизации производительности уменьшение значений уровня нестабильности следует выполнять по истечении фиксированного интервала времени (периода). Зная время «полураспада»[1], можно рассчитать с упреждением значение уровня нестабильности для следующего периода. Снижение уровня нестабильности за несколько периодов можно рассчитать по формуле:

```
f(figure-of-merit, n*t0) = f(figure-of-merit, t0)**n = K**n
```

Значения K ** n для разумного числа n могут быть рассчитаны заранее и сохранены в массиве. Значение K всегда меньше 1. размер массива может быть ограничен, поскольку значения достаточно быстро стремятся к нулю. Это упрощает расчет снижения уровня нестабильности, позволяя ограничиться проверкой выхода за границу массива, выборкой значения из массива и одной операцией умножения независимо от прошедшего времени.

3 Ограничение анонсирования маршрутов с использованием фиксированных таймеров

Этот метод ограничения анонсирования маршрутов включает использование таймеров с фиксированными значениями, применяемых в процессе передачи маршрутов. Основной задачей метода является повышение эффективности упаковки обновлений BGP. Задержку при обновлении стабильных маршрутов следует ограничивать и минимизировать. Задержку при анонсировании недоступных маршрутов не требуется делать нулевой, но также следует минимизировать и ограничивать. Возможна установка отдельной границы для задержки распространения таких анонсов, значение которой не превышает верхнюю границу для задержки анонсов доступных маршрутов.

Протокол BGP определяет использование баз маршрутной информации RIB[2]. Маршруты, для которых требуется повторное анонсирование, могут помечаться в базе RIB или внешней структуре данных, связанной с RIB.

Периодически часть отмеченных маршрутов может сбрасываться. Это настоятельно рекомендуемая мера, соответствующая целям. Нагрузка при расчетах для достаточно простой реализации может квадратично зависеть от N. Для того, чтобы избежать квадратичного роста нагрузки требуется создание той или иной структуры данных для группировки маршрутов с одинаковыми атрибутами.

Реализации следует эффективно паковать обновления, обеспечивая минимальную задержку реанонсирования. Верхняя граница для такой задержки будет определяться только алгоритмом, используемым для минимизации задержки. Должна обеспечиваться эффективность расчетов при наличии очень большого числа кандидатов на реанонсирование.

4 Демпфирование анонсов с учетом уровня стабильности

Этот метод ограничения маршрутных анонсов использует меру стабильности маршрутов независимо для каждого маршрута. Данный метод применяется в тех случаях, когда обновления приходят только от внешних партнеров (EBGP). Применение этого метода к маршрутам, полученным от IBGP, или к анонсам, передаваемым в IBGP или EBGP после выбора маршрута, может приводить к возникновению маршрутных петель.

Показатель качества, основанный на уровне нестабильности маршрута, поддерживается для каждого маршрута независимо. Этот показатель используется в процессе принятия решений о необходимости демпфирования использования маршрута. Маршруты с высоким уровнем нестабильности подавляются. При каждом отзыве маршрута значение уровня нестабильности увеличивается на постоянное значение. Если маршрут сохраняет стабильность, значение уровня нестабильности экспоненциально уменьшается (скорость уменьшения зависит от того, в каком состоянии стабилизируется маршрут – доступном или недоступном). Скорость снижения уровня нестабильности может быть меньше для недоступного маршрута или даже быть для таких ситуаций нулевой (т. е., уровень нестабильности не будет снижаться с течением времени, если маршрут остается недоступным). Выбор скорости уменьшения уровня нестабильности для стабильно недоступного маршрута определяется реализацией. Рекомендуется замедлять снижение уровня для недоступных маршрутов.

Очень эффективная реализация рассматривается в следующих параграфах. Эта реализация требует расчетов лишь для маршрутов, указанных в обновлении (анонс или отзыв), в отличие от простейшего варианта с периодическим снижением уровня нестабильности для всех маршрутов. Рассмотренная реализация включает лишь небольшое число простых операций, которые могут быть выполнены с целыми числами.

Поведение нестабильных маршрутов слабо предсказуемо. Зачастую осциллирующие маршруты будут анонсироваться и отзываться через регулярные интервалы времени, соответствующие значению таймера того или иного протокола (IGP или внешнего протокола). Малоэффективные устройства или умеренное насыщение в сети могут приводить к отзыву маршрута на продолжительный срок с кратковременными периодами доступности такого маршрута.

[1] Время, в течении которого значение уровня нестабильности уменьшается вдвое. *Прим. перев.*
[2] Routing Information Base

4.1 Один или множество наборов параметров?

Поведение алгоритма зависит от множества параметров. Можно задать отдельные наборы параметров для контроля жестких кратковременных осцилляций или хронических умеренных колебаний маршрутов (когда происходят сравнительно редкие «падения» маршрута на достаточно долгий срок). Первый набор будет требовать быстрого снижения уровня нестабильности и низкого порога, заставляющего подавлять маршрут после небольшого числа осцилляций, но позволяющего снова использовать маршрут после сравнительно короткого периода стабильности. Второй вариант требует медленного снижения уровня нестабильности с высоким порогом и может использоваться для маршрутов, имеющих вариант обхода с близкой полосой.

Может также оказаться желательной установка различных пороговых уровней для маршрутов, имеющих близкие по полосе пути обхода, и маршрутов с более медленными или склонными к насыщению путями обхода. Это сделать, связывая разные наборы параметров с различными диапазонами уровня предпочтения. Выбор параметров можно проводить на основе значений BGP LOCAL_PREF.

Выбирать параметры можно также на основе данных о наличии альтернативного маршрута. Маршрут будет рассматриваться, если для любого применимого набора параметров существует альтернативный маршрут с заданным уровнем предпочтения и связанный с набором параметров уровень нестабильности не показывает необходимость демпфирования этого маршрута. При отсутствии альтернативного маршрута будет применяться менее агрессивное демпфирование. В простейшем случае при наличии любого альтернативного маршрута будет применяться более жесткое демпфирование. Поскольку диапазоны уровней предпочтения могут перекрываться, требуется задать лишь верхний (наиболее предпочтительный) уровень.

Может также оказаться желательным задание различных пороговых уровней для маршрутов, использующих коммутируемые соединения, которые могут отключаться из соображений экономии. Такие маршруты могут менять состояние достаточно часто, но подавлять их следует лишь в тех случаях, когда продолжающиеся смены состояния говорят о нестабильности.

Хотя это и не играет важной роли, но может оказаться желательной возможность создания множества наборов конфигурационных параметров для маршрута. Желательной может быть и возможность настройки наборов параметров для некоторого множества маршрутов (определяемого AS path, партнером, адресатом или иными критериями). Опыт может определять требования к гибкости и способам выбора наилучшего набора параметров. Возможность использования различных наборов параметров демпфирования для разных маршрутов и поддержки множества значений уровня нестабильности (figure of merit) для маршрута определяется реализацией.

Выбор параметров может быть основан также на размере префикса. Смысл этого заключается в том, что более длинные префиксы соответствуют маршрутам к меньшему числу конечных систем и такие менее важные префиксы можно подавлять более агрессивно. Такой подход используется достаточно широко. Небольшие сайты и достаточно крупные многодомные сайты часто доступны через длинные префиксы, которые обычно сложно агрегировать. Наличие таких сайтов делает спорным вопрос применимости размера префикса в качестве критерия выбора параметров демпфирования. Сторонники такого выбора говорят, что он помогает повысить эффективность агрегирования маршрутов.

4.2 Параметры конфигурации

В процессе настройки может задаваться множество параметров. Параметры задаются в понятных пользователю единицах. Параметры же, используемые в процессе работы, выражаются в удобных для расчетов единицах. Эти параметры определяются на основе конфигурационных параметров, заданных пользователем. Предлагаемые для реализаций конфигурационные параметры перечислены ниже.

`cutoff threshold (cut)`
Верхний порог числа отзывов маршрута, после достижения которого анонсирование данного маршрута будет подавляться.

`reuse threshold (reuse)`
Нижний порог числа отзывов маршрута, используемый для восстановления анонсов демпфированного ранее маршрута (когда число отзывов станет ниже порогового уровня).

`maximum hold down time (T-hold)`
Максимальное время, на которое маршрут может быть демпфирован независимо от того, насколько он был нестабилен перед данным периодом стабильности.

`decay half life while reachable (decay-ok)`
Время (число минут или секунд), в течение которого накопленное значение уровня нестабильности будет снижено вдвое, если маршрут рассматривается как доступный (независимо от того, подавляется ли он в это время).

`decay half life while unreachable (decay-ng)`
Время (число минут или секунд), в течение которого накопленное значение уровня нестабильности будет снижено вдвое, если маршрут рассматривается как недоступный. Если этот параметр не задан или указано нулевое значение, снижение уровня нестабильности не будет происходить для недоступных маршрутов.

`decay memory limit (Tmax-ok or Tmax-ng)`
Максимальное время хранения информации о предыдущем состоянии нестабильности при условии, что состояние маршрута не изменяется (независимо от того, доступен он или нет). Этот параметр в общем случае служит для определения размеров массива.

Перечисленный выше набор параметров может использоваться во множестве экземпляров, как было сказано в параграфе 4.1. Перечисленные ниже конфигурационные параметры имеют общесистемное значение. К числу общих параметров относится гранулярность отсчета времени при всех расчетах, а также параметры, используемые для переоценки маршрутов, которые были демпфированы ранее.

`time granularity (delta-t)`
Гранулярность времени (в секундах) при выполнении всех расчетов снижения уровня нестабильности.

reuse list time granularity (delta-reuse)

Интервал времени между оценками списков повторного использования (reuse list).

reuse list memory reuse-list-max

Время, соответствующее последнему списку повторного использования. Это может быть максимальное значение из числа параметров T-hold для всех наборов.

number of reuse lists (reuse-list-size)

Число списков повторного использования. Значение параметра может определяться на основе reuse-list-max или задаваться явно.

Рекомендуемая оптимизация, описанная в параграфе 4.8.6, включает массив, который называют массивом индексов повторного использования[1]. Такой массив требуется для каждой используемой скорости снижения уровня нестабильности. Массив служит для оценки и выбора списка повторного использования (reuse list), в который следует поместить маршрут при его демпфировании. Корректное размещение избавляет от необходимости периодической оценки снижения уровня для определения момента, когда можно будет снова использовать маршрут или можно будет восстановить место. Использование массива избавляет от необходимости вычисления логарифма для определения места размещения. Может быть введен дополнительный параметр общесистемного значения.

reuse index array size (reuse-index-array-size)

Этот параметр задает размер массива индексов повторного использования. Данный размер определяет точность, с которой демпфированные маршруты могут быть помещены в набор списков повторного использования на длительное время.

4.3 Рекомендации по установке параметров

Время снижения уровня нестабильности вдвое (decay half life) следует устанавливать существенно большим, чем период осцилляций, для которого используется этот параметр. Например, при установке для снижения уровня времени 10 минут для маршрута, который отзывается и заново анонсируется каждые 10 минут, осцилляции маршрута будут продолжаться, если для верхнего порога (cutoff) установлено значение 2 или выше.

Стабильность показателя качества (уровня нестабильности) определяется накопленным в течение времени общим снижением уровня нестабильности. Это должно приниматься во внимание при установке времени снижения, а также значений порога (cutoff) и повторного использования (reuse). Уровень нестабильности увеличивается при каждом переходе маршрута из доступного состояния в недоступное. Уровень нестабильности уменьшается с течением времени со скоростью, пропорциональной текущему значению уровня. Увеличение скорости осцилляций, следовательно, приводит к более частому увеличению уровня нестабильности и достижению заданного порога в более короткий срок. Когда отображается реакция на постоянную скорость осцилляций, она выглядит как пик резким нарастанием и медленным спадом. Поскольку абсолютное снижение пропорционально уровню нестабильности, при продолжающемся

время	уровень нестабильности как функция времени (в минутах)			
0.00	0.000	0.000	0.000	0.000
0.08	0.000	0.000	0.000	0.000
0.16	0.000	0.000	0.000	0.973
0.24	0.000	0.000	0.000	0.920
0.32	0.000	0.000	0.946	1.817
0.40	0.000	0.953	0.895	2.698
0.48	0.000	0.901	0.847	2.552
0.56	0.953	0.853	1.754	3.367
0.64	0.901	0.807	1.659	4.172
0.72	0.853	1.722	1.570	3.947
0.80	0.807	1.629	2.444	4.317
0.88	0.763	1.542	2.312	4.469
0.96	0.722	1.458	2.188	4.228
1.04	1.649	2.346	3.036	4.347
1.12	1.560	2.219	2.872	4.112
1.20	1.476	2.099	2.717	4.257
1.28	1.396	1.986	3.543	4.377
1.36	1.321	2.858	3.352	4.141
1.44	1.250	2.704	3.171	4.287
1.52	2.162	2.558	3.979	4.407
1.60	2.045	2.420	3.765	4.170
1.68	1.935	3.276	3.562	4.317
1.76	1.830	3.099	4.356	4.438
1.84	1.732	2.932	4.121	4.199
1.92	1.638	2.774	3.899	3.972
2.00	1.550	2.624	3.688	3.758
2.08	1.466	2.483	3.489	3.555
2.16	1.387	2.349	3.301	3.363
2.24	1.312	2.222	3.123	3.182
2.32	1.242	2.102	2.955	3.010
2.40	1.175	1.989	2.795	2.848
2.48	1.111	1.882	2.644	2.694
2.56	1.051	1.780	2.502	2.549
2.64	0.995	1.684	2.367	2.411
2.72	0.941	1.593	2.239	2.281
2.80	0.890	1.507	2.118	2.158
2.88	0.842	1.426	2.004	2.042
2.96	0.797	1.349	1.896	1.932
3.04	0.754	1.276	1.794	1.828
3.12	0.713	1.207	1.697	1.729
3.20	0.675	1.142	1.605	1.636
3.28	0.638	1.081	1.519	1.547
3.36	0.604	1.022	1.437	1.464
3.44	0.571	0.967	1.359	1.385

Рисунок 1: Нестабильность показателя качества при постоянной частоте осцилляций

[1]Reuse index array.

постоянстве осцилляций, базовый уровень имеет тенденцию к прекращению роста и схождению, если он не будет зажат верхним ограничением.

Если базовый уровень пика зажат верхним ограничением, он просто будет достигать верхней границы быстрее при более высокой частоте осцилляций. Например, если переключение маршрута происходит 4 раза за период снижения уровня вдвое, наблюдается следующая последовательность значений. Когда маршрут становится недоступным первый раз, значение становится равным 1. При следующем переключении добавляется 1 к предыдущему значению, которое уменьшилось на корень четвертой степени из 2 (уменьшение уровня за ¼ периода “полураспада”, если снижение происходит по экспоненциальному закону). В результате будет получаться последовательность значений 1, 1.84, 2.55, 3.14, 3.64, 4.06, 4.42, 4.71, 4.96, 5.17, ..., сходящаяся приблизительно к 6.285. Когда переключение происходит 4 раза за период снижения вдвое, уровень будет достигать 3 за 4 цикла, 4 – за 6, 5 – за 10 и сойдется к значению около 6.3. При двух переключениях за период снижения вдвое, уровень достигнет 3 за 7 циклов и сойдется при значении, меньшем 3.5.

На рисунке 1 показана стабильность показателя качества (уровня нестабильности) для постоянной скорости переключения маршрутов. По временной оси значения отложены в периодах снижения уровня нестабильности вдвое. Графики представляют переключение маршрутов с периодом 1/2, 1/3, ¼ и 1/8 от срока снижения уровня вдвое. Установлен верхний порог 4.5, который, как можно видеть на рисунке, оказывает влияние на три графика, эффективно ограничивая время, через которое произойдет реанонсирование маршрута независимо от предыстории. При пороговых уровнях cutoff и reuse 1,5 и 0,75, соответственно, маршруты будут подавляться после того, как будет декларирована из недоступность 2-3 раза и будут снова использоваться приблизительно через по истечении периода стабильности приблизительно в два срока снижения уровня нестабильности вдвое.

Эту функцию можно выразить формально. Доступность маршрута может быть представлена как переменная R с возможными значениями 0 (недоступен) и 1 (доступен). В любой момент времени R принимать только одно значение. Уровень нестабильности увеличивается на 1 при каждом переходе от R=1 к R=0 и зажат верхним ограничением (ceiling). Снижение уровня нестабильности может быть выражено через набор дискретных интервалов следующим образом:

```
figure-of-merit(t) = K * figure-of-merit(t - delta-t)
K = K1 для R=0, K=K2 для R=1
```

Четыре графика представлены вертикально. В силу ограниченного пространства показан ограниченный набор точек по временной оси. Значения уровня нестабильности показаны также в числовой форме. Разрешение по оси уровня нестабильности крайне низко по причине ограниченности изобразительных средств ASCII. График дает лишь грубое качественное представление роста и спада уровня нестабильности. Линии не показаны на графика в силу ограниченных возможностей текстового формата. По причине низкого разрешения на графиках пики и спады видны, но точные значения доступны лишь в виде чисел.

С момента максимального удержания (T-hold) может быть определено отношение значения reuse к верхнему порогу (ceiling). После этого может быть выбрано целое значение верхнего порога (ceiling) так, чтобы не возникало проблем переполнения и остальные значения могли быть подобающим образом масштабированы. Если заданы оба порога или установлено множество наборов параметров, будет использовано наибольшее значение верхнего порога.

На рисунке 2 показан эффект задания различных скоростей спада для периодов доступности и недоступности маршрута (в периоды недоступности скорость снижения устанавливалась в 5 раз меньше). На трех показанных графиках период осцилляций был равен времени снижения уровня нестабильности вдвое, а период доступности маршрута составлял 1/8, 1/2 и 7/8 от периода снижения уровня нестабильности вдвое. В последнем случае маршрут не подавлялся до того, как он стал недоступным в третий раз (когда уровень нестабильности превысил порог после того, как маршрут снова стал доступным).

Основным назначением рисунка 2 является демонстрация эффекта изменения рабочего цикла изменений переменной R при фиксированной частоте переключений. Если константы снижения уровня нестабильности выбраны так, что снижение происходит медленнее, когда R=0 (маршрут недоступен), уровень нестабильности растет более медленно (точнее, более медленно растет базовый уровень пиков), когда маршрут сохраняет доступность более долго. Эффект при восстановлении постоянной доступности маршрута может быть весьма незначительным, если величина пика зажата верхним порогом (ceiling), но он становится более значимым, если пик не

время	уровень нестабильности как функция времени (в минутах)		
0.00	0.000 .	0.000 .	0.000 .
0.20	0.000 .	0.000 .	0.000 .
0.40	0.000 .	0.000 .	0.000 .
0.60	0.000 .	0.000 .	0.000 .
0.80	0.000 .	0.000 .	0.000 .
1.00	0.999 .	0.999 .	0.999 .
1.20	0.971 .	0.971 .	0.929 .
1.40	0.945 .	0.945 .	0.809 .
1.60	0.919 .	0.865 .	0.704 .
1.80	0.894 .	0.753 .	0.613 .
2.00	1.812 .	1.657 .	1.535 .
2.20	1.762 .	1.612 .	1.428 .
2.40	1.714 .	1.568 .	1.244 .
2.60	1.667 .	1.443 .	1.083 .
2.80	1.622 .	1.256 .	0.942 .
3.00	1.468 .	1.094 .	0.820 .
3.20	2.400 .	2.036 .	1.694 .
3.40	2.335 .	1.981 .	1.475 .
3.60	2.271 .	1.823 .	1.284 .
3.80	2.209 .	1.587 .	1.118 .
4.00	1.999 .	1.381 .	0.973 .
4.20	2.625 .	2.084 .	1.727 .
4.40	2.285 .	1.815 .	1.503 .
4.60	1.990 .	1.580 .	1.309 .
4.80	1.732 .	1.375 .	1.139 .
5.00	1.508 .	1.197 .	0.992 .
5.20	1.313 .	1.042 .	0.864 .
5.40	1.143 .	0.907 .	0.752 .
5.60	0.995 .	0.790 .	0.654 .
5.80	0.866 .	0.688 .	0.570 .
6.00	0.754 .	0.599 .	0.496 .
6.20	0.656 .	0.521 .	0.432 .
6.40	0.571 .	0.454 .	0.376 .
6.60	0.497 .	0.395 .	0.327 .
6.80	0.433 .	0.344 .	0.285 .
7.00	0.377 .	0.299 .	0.248 .
7.20	0.328 .	0.261 .	0.216 .
7.40	0.286 .	0.227 .	0.188 .
7.60	0.249 .	0.197 .	0.164 .
7.80	0.216 .	0.172 .	0.142 .
8.00	0.188 .	0.150 .	0.124 .

Рисунок 2: Использование раздельных параметров снижения уровня нестабильности.

достигает верхнего порога (ceiling). На рисунке 2, где нестабильность маршрута достаточно кратко-временна, верхний порог не достигается, следовательно, маршруты, которые доступны в течение большей части периода переключения маршрутов (route flap cycle) начинают снова использоваться (включаются в RIB и анонсируются партнерам) после восстановления стабильности (R становится равным 1, показывая, что маршрут стал доступным и сохраняет это состояние)быстрее, чем другие маршруты.

На рисунках 1 и 2 маршруты будут подавляться. Маршруты, переключающиеся с периодом снижения уровня нестабильности или быстрее того, будут отзываться 2 или 3 раза и после этого сохраняться как отозванные до тех пор, пока они не начнут стабильно анонсироваться и сохранять стабильность в течение периода от 1,5 до 2,5 периодов снижения уровня нестабильность.

Целью демпфирования осцилляций маршрутов BGP является снижение нагрузки на процессоры промежуточного маршрутизатора и “нисходящих” (downstream) маршрутизаторов (BGP-партнеры и партнеры этих партнеров, которые будут видеть анонсы маршрутов от промежуточного маршрутизатора). Расчет уровня нестабильности для каждого дискретного интервала времени по формуле

```
figure-of-merit(t) = K * figure-of-merit(t - delta-t)
```

будет весьма неэффективным и неподходящим для этой цели. Проблема решается как можно более долгим откладыванием момента проведения расчета и однократным простым расчетом с учетом снижения уровня нестабильности за время, прошедшее с момента последнего обновления значения этого уровня. Использование массивов значений снижения уровня обеспечивает однократный простой расчет. Применение списков повторного использования (см. ниже) обеспечивает способ задержки проведения расчетов. Маршрут становится доступным для использования, если не происходит дальнейших изменений в течение заданного периода и маршрут пока является недоступным. Структура данных восстанавливается, если состояние маршрута не изменялось в течение заданного периода времени и маршрут является недоступным. Массив повторного использования обеспечивает способ оценки возможного срока задержки расчетов при отсутствии изменений.

Большая временная гранулярность будет сохранять размер таблицы параметров. Временную гранулярность следует делать меньше, чем минимальное разумное время между переключениями маршрута в худшем из ожидаемых случае. Может оказаться разумным установка для этого параметра фиксированного значения на этапе компиляции или установка значения по умолчанию с настоятельной рекомендацией не изменять его. При экспоненциальном снижении размер массива можно значительно уменьшить путем установки периода полной стабильности, по истечении которого значение уровня нестабильности можно просто считать нулевым, не выполняя расчетов. Кроме того, для реализации очень длительного снижения можно использовать неоднократное умножение при достижении границы массива.

Списки повторного использования содержат маршруты, сгруппированные по времени ожидания возможности повторного использования. Периодически каждый список будет перемещаться на одну позицию вперед, а один список будет удаляться, как описано в параграфе 4.8.7. Все подавляемые маршруты из подлежащего удалению списка будут оцениваться заново и включаться в число используемых, либо помещаться в другой список в соответствии с дополнительным временем, которое должно пройти до того, как маршрут можно будет использовать снова. Последний список будет содержать все маршруты, которые не будут анонсироваться дольше, чем допустимо для остальных списков. Кода последний список будет перемещаться вперед, некоторые из содержащихся в нем маршрутов не будут готовые к повторному использованию и будут помещены в очередь снова. Временной интервал переоценки демпфированных маршрутов и число списков следует делать настраиваемыми. Разумными значениями для использования по умолчанию будут 30 секунд и 64 списка. Маршруты, остающиеся демпфированными более длительное время, следует оценивать заново каждые 32 минуты.

4.4 Рабочие (Run Time) структуры данных

Требуется небольшой фиксированный объем памяти для хранения данных общесистемного значения. В тех случаях, когда используется множество наборов конфигурационных параметров, потребуется память для каждого из таких наборов. Требуется также небольшой объем памяти для хранения информации, связанной с каждым маршрутом. Требуется набор списков, которые используются для сохранения демпфированных маршрутов до того момента, когда их можно будет использовать снова.

Может применяться отдельный список повторного использования для сохранения недоступных маршрутов с целью последующего восстановления в тех случаях, когда маршруты недоступны слишком долго (этот список точнее было бы назвать термином recycling list). Такой список позволяет максимально быстро делать доступными освобожденные структуры данных. В дополнение к этому структуры данных могут просто помещаться в очередь, а данные восстанавливаться, когда маршрут окажется в начале очереди, если требуется пространство для хранения. Этот вариант менее оптимален, но прост.

Если разрешено использование множества наборов конфигурационных параметров для маршрута, возникает необходимость в организации связей между уровнем нестабильности и параметрами конфигурации для каждого маршрута. Построение связного списка таких объектов представляется одним из многих возможных вариантов реализации таких связей. Подобно этому требуется создание ассоциаций между маршрутами и списком повторного использования. Будет требоваться также незначительный объем служебной информации для реализации какой-либо структуры данных для списков повторного использования. Рекомендуемая реализация использует два связных списка и требует наличия двух указателей для каждого показателя уровня нестабильности.

Каждый набор конфигурационных параметров может ссылаться на массивы снижения уровня нестабильности и массивы повторного использования. Эти массивы следует совместно использовать для множества наборов параметров, поскольку для хранения массивов требуется память. В таком случае для маршрутизатора в целом может использоваться один набор ссылок на начала массивов повторного использования.

4.4.1 Структуры данных для наборов конфигурационных параметров

На основе конфигурационных параметров, рассмотренных в предыдущем параграфе, можно рассчитать перечисленные ниже значения, как масштабируемые целые числа, непосредственно на основе соответствующих параметров конфигурации.

- Фактор масштабирования снижения уровня нестабильности (decay-array-scale-factor)

- Значение порога cutoff (cut)
- Значение параметра повторного использования (reuse)
- Верхний порог уровня нестабильности (ceiling)

Каждый набор параметров будет содержать ссылку на один или два массива decay и один или два массива reuse. Если снижение уровня нестабильности происходит с одинаковой скоростью независимо от доступности маршрута или для недоступных маршрутов уровень нестабильности не снижается, требуется только один массив decay.

4.4.2 Структуры данных для массивов Decay и Reuse Index

Приведенные ниже значения также рассчитываются на основе конфигурационных параметров, но этот расчет не делается напрямую. Описание процедур расчета приведено в параграфе 4.5.

- Скорость снижения уровня нестабильности (decay-delta-t)
- Размер массива decay (decay-array-size)
- Массив decay (decay[])
- Размер массива индексов повторного использования (reuse-index-array-size)
- Массив индексов повторного использования (reuse-index-array[])

Для каждой заданной скорости снижения уровня (decay rate) будет использоваться массив, в котором хранятся значения рассчитанных параметров, возводимых в степень, задаваемую индексом каждого элемента массива. Такой подход позволяет ускорить расчеты. Снижение уровня нестабильности за один период (decay rate per tick) является промежуточным значением, выражаемым действительным числом, и используется при расчете значений, хранящихся в массивах decay. Размер массива рассчитывается с учетом заданного в конфигурации предела распределения памяти для decay, заданного как размер массива или максимальное время удержания.

Размер массива decay должен быть достаточным большим. Критерием выбора размера может быть временная гранулярность, возможность получения нулевого уровня при целочисленном округлении значений или заданное реализацией значение, предохраняющее от чрезмерного расхода памяти. Разработчики могут также создать массив небольшого размера и использовать его неоднократно для расчета снижения уровня нестабильность.

Массив индексов повторного использования (reuse index) используется с теми же целями, что и массивы decay. В BGP маршрут считается используемым, если он выбран как лучший (best route). В этом контексте используемые маршруты включают в RIB и выбирают для анонсирования партнерам BGP. Если маршрут отзывается (с помощью анонса BGP, показывающего недоступность данного маршрута), он больше не относится к числу используемых. После того, как доступность маршрута будет восстановлена, он может не сразу попасть в число используемых, если уровень нестабильности для него будет указывать на недавнюю нестабильность. После того, как маршрут достаточно долго будет сохранять стабильность и уровень нестабильности станет ниже порога повторного использования ("reuse" threshhold), этот маршрут может начать использоваться снова (будет трактоваться как действительно доступный, помещен в RIB и передаваемые партнерам анонсы). Время, по истечение которого может начаться повторное использование маршрута, может определяться просмотром массива. Массив может создаваться на основе скорости снижения уровня нестабильности. Массив индексируется с использованием целочисленного коэффициента, пропорционального отношению уровня нестабильности к порогу начала повторного использования.

4.4.3 Состояние для каждого маршрута

Информация должна поддерживаться в форме некого «кортежа» (tuple), представляющего маршрут. В кортеже должна содержаться (как минимум) информация (префикс BGP и его размер). Возможно включение (исключение) различных атрибутов BGP в зависимости от ситуации. По умолчанию в кортеж следует включать также AS path. Опционально могут включаться и другие атрибуты BGP — такие, как MULTI_EXIT_DISCRIMINATOR (MED).

Представление маршрута в целях демпфирования осцилляций имеет вид:

Элемент	По умолчанию	Опционально
Префикс NLRI	требуется	
Размер NLRI	требуется	
AS path	включен	может быть исключен
Последний AS set в пути	исключен	может быть включен
next hop	исключен	может быть включен
MED	исключен	может быть включен (только для сравнения)

Атрибут AS path обычно включается для идентификации нестабильности в нисходящем направлении, которая не подавляется или ослабляется недостаточно и является чередованием стабильного и нестабильного пути. В редких случаях может оказаться желательным исключение AS path для всего или части набора префиксов. Если AS path заканчивается в AS set, на практике такой путь всегда является агрегируемым. Изменение «трейлерного» AS set следует игнорировать. В идеале сравнение AS должно давать по крайней мере одну AS, которая сохраняется в старом и новом AS set, но допускается и полное игнорирование содержимого «трейлерного» AS set.

Включение изменений hop и MED помогает подавить использование AS с внутренней нестабильностью и избежать интервалов next hop , которые ближе к нестабильному пути IGP в смежной AS. При использовании большого числа значений MED рост количества состояний может создать проблему. В силу этого обстоятельства MED не используется по умолчанию и включается лишь в качестве части сравнения «кортежей» с использованием одного элемента состояния независимо от значения MED. Включение MED будет подавлять использование смежной AS даже в тех случаях, когда изменения не распространяются дальше. Использование MED является единственно безопасной

практикой для случаев, когда известно о существовании пути через другую AS или имеется достаточно партнерских со смежной AS сайтов и подавляться будут только маршруты для части партнеров.

4.4.4 Структуры данных для маршрута

Перечисленные ниже данные должны поддерживаться на уровне маршрутов. Маршрутом в данном случае считается кортеж, обычно содержащий NLRI, next hop и AS path, как определено в параграфе 4.4.3.

stability figure of merit (figure-of-merit) — стабильность добротности (добротность)

Каждый маршрут должен обеспечивать стабильность добротности для применимого набора параметров.

last time updated (time-update) — время последнего обновления (время обновления)

Точное время последнего обновления должно сохраняться для того, обеспечить возможность отложить экспоненциальное снижение набранного уровня добротности до того момента, когда маршрут можно будет считать подходящим для изменения статуса (переход из недоступного в доступные или анонсирование в списках повторного использования).

config block pointer — указатель конфигурационного блока

Любая реализация, поддерживающая множество наборов параметров, должна обеспечивать способ быстрой идентификации набора параметров, который соответствует рассматриваемому маршруту. Для реализаций, поддерживающих только один набор параметров, где все маршруты дожны трактоваться, как один и тот же, такой указатель не требуется.

reuse list traversal pointers

При использовании двойных связанных списков (doubly linked list) для реализации списков повторного использования требуется два указателя — на следующий и предыдущий элемент. В общем случае имеется двойной связанный список, который не используется, когда применение маршрута демпфировано, что позволяет повторно проходить по списку и избавляет от необходимости хранения дополнительного указателя.

4.5 Обработка конфигурационных параметров

Из конфигурационных параметров можно заранее определить число значений, которые могут использоваться повторно и сохранить их для ускорения последующих часто повторяющихся расчетов.

Масштабирование обычно зависит от наибольшего значения, которого может достигнуть добротность (потолок). Реальное значение потолка обычно определяется приведенным ниже уравнением. Для потолка можно также указать конкретное значение, которое, в свою очередь, определяет T-hold.

```
ceiling = reuse * (exp(T-hold/decay-half-life) * log(2))
```

В этом уравнении параметр reuse указывает порог повторного использования, описанный в параграфе 4.2.

Методы масштабирования в целочисленной арифметике здесь не описываются детально. Приведены методы расчета действительных значений. Преобразование в целочисленные значения и детали масштабирования в целочисленной арифметике оставлены для отдельной работы.

В качестве значения потолка может быть установлено максимальное целое число, помещающееся в половину битов целого числа без знака. Это позволит при масштабировании умножать целое число на коэффициент масштабирования (scaled decay), а затем сдвигать вниз. Реализации могут воспользоваться действительными числами или применять любое целочисленное масштабирование, подходящее для их архитектуры.

penalty value and thresholds (как пропорционально масштабируемые целые числа)

Снижение уровня добротности (пенальти) для отзыва маршрута и отсечку значений должны быть масштабируемыми в соответствии с описанным выше масштабным коэффициентом.

decay rate per tick (decay[1])

Величина снижения за единицу приращения времени (определяется гранулярностью отсчета времен) должна быть определена (по крайней мере изначально как действительное число). Снижение за один «тик» будет числом слегка меньше 1. Это корень N-ой степени из половины, где N составляет половину времени жизни, поделенного на гранулярность отсчета времени.

```
decay[1] = exp ((1 / (decay-half-life/delta-t)) * log (1/2))
```

decay array size (decay-array-size)

Размера массива decay array size в памяти decay, поделенный на гранулярность отсчета времени. Если отсечка целой части делает значение элемента массива нулевым, массив можно уменьшить. Реализации следует также задавать максимальный приемлемый размер массива или разрешать более одного умножения.

```
decay-array-size = (Tmax/delta-t)
```

decay array (decay[])

Каждый i-й элемент массива decay является задержкой на «тик», возведенной в степень i. Лучшим способом выполнить это может оказаться последовательное умножение действительных чисел с дальнейшим целочисленным округлением или отсечкой. Сам массив требуется рассчитывать только при старте.

```
decay[i] = decay[1] ** i
```

4.6 Индексирование списков повторного использования

Списки повторного использования могут восприниматься дружественно, если число маршрутов для демпфирования осцилляций достаточно велико. Предложен метод ускорения выбора списка повторного использования, который может быть применен для данного маршрута. Метод описан в параграфе 4.2, его конфигурация — в параграфе 4.4.2, а алгоритмы в параграфах 4.8.6 - 4.8.7. В данном параграфе рассматривается индексирование списков повторного использования.

Отношение добротности рассматриваемого маршрута к значению отсечки служит основой для просмотра массива. Отношение масштабируется и отсекается до целого значения, после чего применяется в качестве индекса массива. Элемент массива является целым числом, служащим для выбора списка повторного использования.

reuse array maximum ratio (max-ratio)

Максимальное отношение между текущим стабильным уровнем добротности и целевым значением повторного использования, которое может быть индексировано массивом повторного использования. Оно может быть ограничено потолком, вносимым максимальным временем удержания или продолжительностью периода, покрываемого списком повторного использования.

```
max-ratio = min(ceiling/reuse, exp((1 / (half-life/reuse-array-time)) * log(2)))
```

reuse array scale factor (scale-factor)

Поскольку массив повторного использования служит для оценки, коэффициент масштабирования для него рассчитывается так, чтобы использовался полный размер массива.

```
scale-factor = reuse-index-array-size / (max-ratio - 1)
```

reuse index array (reuse-index-array[])

Каждый элемент массива должен содержать индекс массива списков повторного использования, указывающий на начало одного из списков. Этот индекс должен соответствовать списку, который будет оцениваться после того, как маршрут будет выбран для повторного использования данным отношением текущего уровня стабильности к целевому уровню повторного использования, соответствующему элементу массива повторного использования.

```
reuse-index-array[j] = integer((decay-half-life / reuse-time-granularity) * log(1/(reuse * (1
+ (j / scale-factor)))) / log(1/2))
```

Для определения очереди, куда следует поместить маршрут, который будет подавляться, используется описанная далее процедура. Значение текущей добротности делится на значение отсечки, к результату добавляется 1 и сумма умножается на коэффициент масштабирования. Полученное значение будет индексом для массива индексов повторного использования (reuse-index-array[]). Значение, полученное из массива индексов повторного использования (reuse-index-array[]) является индексом для массива списков повторного использования (reuse-array[]). Если этот индекс указывает на конец массива, используется последняя очередь. В остальных случаях массив просматривается и из него выбирается очередь с соответствующим номером. Этот механизм работает достаточно быстро, легко устанавливается и не расходует большого объема памяти.

4.7 Пример конфигурации

Ниже представлен простой пример, в котором оценивается перекрытие пространства для установки конфигурационных параметров. Предполагается, что:

1. имеется один набор параметров для всех маршрутов,
2. «распад» неиспользуемых маршрутов происходит медленней «распада» используемых,
3. массив должен иметь полный размер вместо того, чтобы разрешать более одного умножения на одну операцию «распада» с целью снижения размера массива.

Этот пример используется в последующих параграфах. Применение множества наборов параметров усложняет ситуацию. Там, где разрешены множества наборов параметров для одного маршрута, «распадная» часть алгоритма просто повторяется для каждого набора. Если разные маршруты могут использовать разные наборы параметров, эти маршруты должны иметь указатели на соответствующие наборы для сокращения времени поиска. Остальная часть алгоритма остается неизменной.

Наборы конфигурационных параметров и параметров реализации имеют вид:

1. Конфигурационные параметры
   - cut = 1.25
   - reuse = 0.5
   - T-hold = 15 мин.
   - decay-ok = 5 мин.
   - decay-ng = 15 мин.
   - Tmax-ok, Tmax-ng = 15, 30 мин.
2. Параметры реализации
   - delta-t = 1 сек.
   - delta-reuse = 15 сек.
   - reuse-list-size = 256
   - reuse-index-array-size = 1024

С помощью этих параметров конфигурации и реализации, а также уравнений из параграфа 4.5 можно рассчитать дополнительное пространство (space overhead). Это фиксированное пространство не будет зависеть от числа маршрутов. Указанные требования к пространству связаны со стабильным маршрутом. Для нестабильных маршрутов возникают дополнительные требования к пространству. Пространственные требования для приведенных выше параметров указаны в списке.

1. Фиксированное пространство (с использованием параметров из предыдущего примера)
   - 900 * integer — массив «распада»
   - 1,800 * integer — массив «распада»

- 120 * pointer — начала списков повторного использования
- 2,048 * integer — массивы индексов повторного использования

2. Дополнительное пространство для стабильного маршрута
   - pointer — содержит нулевой элемент
3. Дополнительное пространство для нестабильного маршрута
   - pointer — указатель на структуру демпфирования, содержащую
   - integer — добротность + бит для состояния
   - integer — время последнего обновления
   - 2 * pointer — указатели на списки повторного использования (предыдущий, следующий)

Размер массивов «распада» определяется в соответствии с delta-t и Tmax-ok или Tmax-ng. Число заголовков списков повторного использования определяется значением delta-reuse и большим из Tmax-ok и Tmax-ng. Имеется два массива индексов повторного использования, размер которых задается конфигурационным параметром.

На рисунке 3 показано поведение алгоритма с указанными выше параметрами. Рассмотрены 4 случая, для каждого из которых присутствует 12-минутный интервал маршрутных осцилляций. Используется два периода осцилляций в 2 и 4 минуты продолжительностью. Используются два цикла загрузки, в одном из которых маршрут доступен в течение 20% продолжительности цикла, в другом - 80%. Во всех четырех случаях маршрут подавляется после того, как он во второй раз становится недоступным. Демпфирование маршрута сохраняется в течение некоторого времени после восстановления его стабильности. Маршруты, которые осциллируют с периодом 4 минуты перестанут подавляться в течение 9-11 после восстановления их стабильности. Маршруты с периодом осцилляций 2 минуты будут подавляться не более 15 минут после восстановления их стабильности.

4.8 Обработка операций протокола маршрутизации

В предыдущих параграфах рассмотрены параметры конфигурации и их связи с параметрами и массивами, используемыми при работе а также обеспечивающими алгоритмы инициализации рабочей области памяти. В этом параграфе рассматриваются этапы обработки маршрутных событий и таймеров.

Маршрутные события включают:

1. Первое появление (или восстановление после продолжительного отсутствия) нового партнера BGP или маршрута (параграф 4.8.1).
2. Переход маршрута в недоступное состояние (параграф 4.8.2)
3. Восстановление доступности маршрута (параграф 4.8.3)
4. Изменения маршрутов (параграф 4.8.4)
5. Потеря партнера (параграф 4.8.5)

Список повторного использования служит для обеспечения способов быстрой оценки демпфированного маршрута, который достаточно долго сохранял стабильность и может быть восстановлен или был достаточно долго демпфирован и может рассматриваться, как новый. Описаны две операции:

1. Вставка в список повторного использования (параграф 4.8.6).
2. Обработка списка повторного использования каждые delta-t секунд (параграф 4.8.7).

```
время    уровень нестабильности как функция времени (в минутах)
0.00    0.000 .         0.000 .         0.000 .         0.000 .
0.62    0.000 .         0.000 .         0.000 .         0.000 .
1.25    0.000 .         0.000 .         0.000 .         0.000 .
1.88    0.000 .         0.000 .         0.000 .         0.000 .
2.50    0.977  .        0.968  .        0.000 .         0.000 .
3.12    0.949  .        0.888  .        0.000 .         0.000 .
3.75    0.910  .        0.814  .        0.000 .         0.000 .
4.37    1.846    .      1.756    .      0.983  .        0.983  .
5.00    1.794    .      1.614    .      0.955  .        0.935  .
5.63    1.735    .      1.480    .      0.928  .        0.858  .
6.25    2.619      .    2.379      .    0.901  .        0.786  .
6.88    2.544      .    2.207      .    0.876  .        0.721  .
7.50    2.472      .    2.024      .    0.825  .        0.661  .
8.13    3.308        .  2.875        .  1.761    .      1.608    .
8.75    3.213        .  2.698        .  1.711    .      1.562    .
9.38    3.122        .  2.474        .  1.662    .      1.436   .
10.00   3.922          . 3.273         . 1.615    .      1.317   .
10.63   3.810          . 3.107         . 1.569    .      1.207   .
11.25   3.702          . 2.849         . 1.513    .      1.107   .
11.88   3.498        .  2.613        .  1.388   .        1.015   .
12.50   3.904          . 3.451         . 2.312      .    1.953    .
13.13   3.580          . 3.164         . 2.120      .    1.791    .
13.75   3.283        .  2.902        .  1.944    .      1.643    .
14.38   3.010        .  2.661        .  1.783    .      1.506    .
15.00   2.761      .    2.440      .    1.635    .      1.381   .
15.63   2.532      .    2.238      .    1.499    .      1.267   .
16.25   2.321     .     2.052     .     1.375    .      1.161   .
16.88   2.129     .     1.882     .     1.261    .      1.065   .
17.50   1.952    .      1.725    .      1.156    .      0.977  .
18.12   1.790    .      1.582    .      1.060    .      0.896  .
18.75   1.641    .      1.451    .      0.972  .        0.821  .
19.38   1.505    .      1.331    .      0.891  .        0.753  .
20.00   1.380   .       1.220   .       0.817  .        0.691  .
20.62   1.266   .       1.119   .       0.750  .        0.633  .
21.25   1.161   .       1.026   .       0.687  .        0.581  .
21.87   1.064   .       0.941   .       0.630  .        0.533  .
22.50   0.976  .        0.863  .        0.578  .        0.488 .
23.12   0.895  .        0.791  .        0.530  .        0.448 .
23.75   0.821  .        0.725  .        0.486 .         0.411 .
24.37   0.753  .        0.665  .        0.446 .         0.377 .
25.00   0.690  .        0.610  .        0.409 .         0.345 .
```

Рисунок 3: Несколько достаточно продолжительных циклов осцилляций (в течение 12 минут), за которыми следует стабильное состояние.

4.8.1 Обработка новых партнеров или маршрутов

При появлении партнера никаких действий не требуется, если у маршрутов нет истории нестабильности (например, партнер новый или впервые анонсирует свои маршруты). Для каждого маршрута указатель на структуру демпфирования будет нулевым, а сам маршрут будет использоваться. Такая же ситуация возникает для новых маршрутов или маршрутов, которые долгое время не использовались, их уровень качества достиг нулевого значения, а структура для демпфирования была удалена.

4.8.2 Обработка сообщений Unreachable

При отзыве или изменении маршрута (обработка изменений описана в параграфе 4.8.4) используется описанная ниже процедура.

Если истории стабильности нет (нулевой указатель на структуру демпфирования):

```
1. выделяется структура демпфирования;
2. устанавливается figure-of-merit = 1
3. маршрут отзывается.
```

При наличии истории:

```
1.  устанавливается t-diff = t-now - t-updated
2.  если (t-diff указывает на конец массива) {
            setfigure-of-merit =1
      } иначе {
            setfigure-of-merit = figure-of-merit *decay-array-ok [t-diff ]+ 1
            если (figure-of-merit > ceiling) {
                    setfigure-of-merit = ceiling
            }
    }
3.  маршрут удаляется из списка повторного использования, если он там присутствует
4.  маршрут отзывается, если он уже не был демпфирован.
```

В любом случае:

```
1.  устанавливается t-updated = t-now
2.  маршрут помещается в список повторного использования (см. параграф 4.8.6).
```

При наличии истории стабильности предшествующее значение уровня добротности уничтожается с помощью массива распада (decay-array). Индекс определяется путем деления разности между текущим временем и временем последнего обновления на гранулярность отсчета времени. Если индекс получается нулевым, значение добротности не меняется (не уничтожается). Если значение индекса превышает размер массива, индекс предполагается нулевым. В остальных случаях значение индекса используется для выбора элемента в массиве распада и на значение этого элемента умножается значение уровня добротности. При использовании предложенного метода целочисленного масштабирования происходит сдвиг вниз на половину целого. Добавляется отмасштабированный штраф для недоступного более маршрута (показан выше, как 1). Если результат превышает потолок, он заменяется значением для потолка. После этого обновляется значения поля времени последнего обновления (предпочтительно, с учетом времени, отсеченного перед расчетом распада).

Когда маршрут становится недоступным, должны рассматриваться дополнительные пути. Этот процесс несколько усложняется, если используемые конфигурационные параметры зависят от наличия действующих дополнительных путей. Если все эти дополнительные пути были демпфированы по причине существования другого дополнительного пути, а данный отзыв маршрута изменил это условие, демпфированные дополнительные пути могут быть оценены заново. Переоценивать их следует в порядке обычного предпочтения маршрутов. Когда один из демпфированных ранее дополнительных маршрутов становится применимым по причине отсутствия иных дополнительных путей, дальнейшая переоценка маршрутов не проводится. Это применимо лишь в тех случаях, когда маршруты имеют два разных порога повторного использования — один для применения при наличии дополнительного пути и другой (более высокий) для использования в тех случаях, когда демпфирование маршрута будет вызывать полную недоступность адресатов.

4.8.3 Обработка маршрутных анонсов

При повторном анонсировании маршрута в случае отсутствия структуры демпфирования используется процедура, описанная в параграфе 4.8.1.

```
1.  не создавать новую структуру демпфирования;
2.  использовать маршрут.
```

При наличии структуры демпфирования значение уровня добротности устраняется, поля figure-of-merit и t-updated обновляются. Далее принимается решение о незамедлительном использовании маршрута или сохранении демпфированного состояние в течение некоторого времени.

```
1.  установить t-diff = t-now - t-updated
2.  если (t-diff выходит за пределы массива) {
       установить figure-of-merit = 0
    } иначе {
       установить figure-of-merit = figure-of-merit * decay-array-ng[t-diff]
    }
3.  если (нет демпфирования и figure-of-merit < cut) {
       использовать маршрут
    } иначе, если (имеется демпфирование и figure-of-merit < reuse) {
       установить состояние «не подавлять»
       удалить маршрут из списка повторного использования
       использовать маршрут
    } иначе {
       установить состояние «подавлять»
```

```
            не использовать маршрут
            поместить маршрут в список повторного использования (см. параграф 4.8.6)
        }
    4.  если (figure-of-merit > 0) {
            установить t-updated = t-now
        } иначе {
            восстановить память для структуры damping
            обнулить указатель на структуру damping
        }
```

Если маршрут считается применимым, должен быть проведен поиск наилучшего в данный момент маршрута. Обнаруженные недавно маршруты оцениваются в соответствии с правилами выбора маршрутов протокола BGP.

Если маршрут применим, проверяется маршрут, который ранее был лучшим. Перед сравнением маршрутов текущий лучший маршрут оценивается заново, если используются разные наборы параметров а зависимости от наличия альтернативного пути. Если альтернативного пути не было, предшествующий лучший маршрут может быть демпфирован.

Если новый маршрут подавляется, его включают в список повторного использования только в том случае, когда он имел бы преимущество перед текущим лучшим маршрутом, если бы новый маршрут считался стабильным. Нет оснований включать в список повторного использования маршрут, который после признания его применимым не использовался про причине наличия лучших путей. Такой маршрут не будет переоцениваться, пока предпочтительный маршрут не утратит доступности. Как отмечено здесь, менее предпочтительный маршрут может быть переоценен и потенциально использован или помещен в список повторного использования при обработке отзыва более предпочтительного маршрута.

4.8.4 Обработка изменений маршрутов

Если маршрут заменяется маршрутизатором-партнером, предлагающим новый путь, заменяемый маршрут следует трактовать, как будто для него получен анонс недоступности (см. параграф 4.8.2). Это будет происходить в тех случаях, когда партнер, расположенный в AS path позади, непрерывно переключается между двумя AS и не подавляет маршрутных осцилляций (или использует недостаточное демпфирование). Не существует способа различить ситуации, когда определить, что AS path является стабильным, а другой осциллирует или осциллируют оба пути. Если цикл осцилляций достаточно короток по сравнению со временем схождения, ни один из маршрутов не будет обеспечивать надежной доставки пакетов. По причине отсутствия возможности повлиять на выбор стабильного маршрута партнером, единственным вариантом остается наказание для обоих маршрутов, путем трактовки каждого изменения, как недоступности маршрута, за которой следует его анонсирование.

4.8.5 Обработка потери маршрутизатора-партнера

При разрыве сеанса с партнеров все анонсируемые этим партнером маршруты маршруты помечаются, как нестабильные, или сама сессия отмечается, как нестабильная. Маркировка сессии в целом обеспечивает существенную экономию памяти. Поскольку анонсы недоступности маршрутов передаются за пределы проблемной области, могут меняться состояния маршрутов за пределами области прямых соседей участников прерванной сессии BGP. Если нестабильность сохраняется, непосредственным соседям требуется лишь сохранять историю нестабильности для партнера. Маршрутизаторы за пределами зоны соседей не будут далее получать анонсов или отзывов маршрутов и с течением времени будут удалять структуру демпфирования.

В будущем могут быть предложены уведомления BGP с использованием необязательных переходных атрибутов для снижения издержек, связанных со структурами демпфирования.

4.8.6 Вставка в список Reuse Timer List

Список повторного использования служит для обеспечения возможности быстрой оценки ранее демпфированных маршрутов, которые показали стабильность, достаточную для возобновления их применения. Структура данных включает заголовков списков (голова - list head). Каждый список включает набор маршрутов, для которых планируется переоценка примерно в одно время. Множество таких заголовков трактуется, как циклический массив (см. рисунок 4).

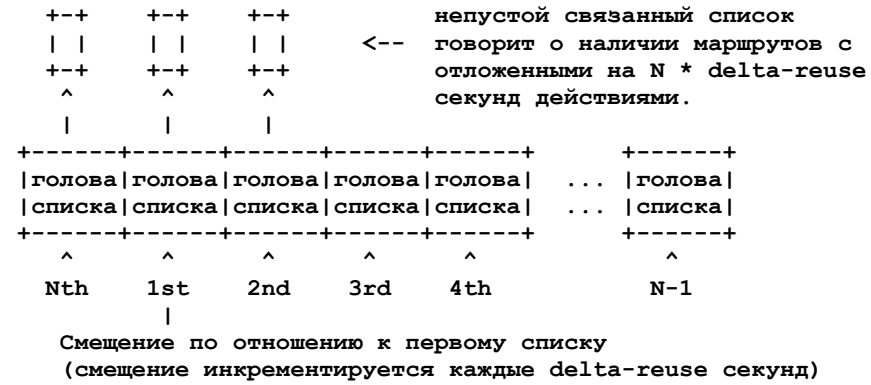

Рисунок 4: Структура данных списка повторного использования.

Простой реализацией циклического массива заголовков может быть массив таких заголовков. Доступ к элементам массива организуется по их смещению от начала. N-й список будет располагаться со смещением N-го плюс размер одного заголовка. В приведенных ниже примерах предполагается такая структура массива.

Основным требованием является возможность включения элемента в наиболее подходящую очередь с минимальными затратами на вычисления. Рассчитывается только текущее значение уровня добротности. Вместо расчетов с использованием логарифмов применяется массив повторного использования reuse-array[], описанный в параграфе 4.6. Массив, масштаб и границы рассчитываются заранее для отображения значений figure-of-merit на ближайший заголовок списка без расчета логарифмов (см. параграф 4.5).

Отметим, что в последующих параграфах записи вида modulo a b означают b % a в нотации языка C.

Например число 1023 по модулю 16 (modulo 16 1023) будет иметь значение 15.

```
1.  масштабировать добротность для поиска в массиве индексов
2.  проверить индекс на предмет выхода за границы массива
3.  если (в границах массива) {
      установить index = reuse-array[index]
    } иначе {
      установить index = reuse-list-size-1
    }
4.  поместить в список
      reuse-list[moduloreuse-list-size (index + offset)]
```

Выбор списка повторного использования включает только операции умножения и сдвига для выполнения масштабирования и отсечки до целого значения индекса после чего выполняется поиск в массиве reuse-array[]. Найденное в массиве значение используется для выбора списка повторного использования. Список повторного использования является циклическим. Наиболее распространенным способом реализации циклического списка является использование массива и смещения, с фиксированным модулем. Для перехода по циклическому списку смещение инкрементируется с учетом модуля.

4.8.7 Обработка событий, связанных с таймерами

Гранулярность таймера повторного использования следует делать более грубой, чем для таймера распада (decay). В результате по завершении отсчета таймера повторного использования демпфированные маршруты следует дополнительно «разрушать» путем множественного инкрементирования времени распада. Части расчетов можно избежать за счет вставки в список повторного использования, соответствующий однократному инкрементированию прошлой желательности повторного использования. В тех случаях, когда списки повторного использования имеют более долгую «память», нежели «память распада» (см. выше), все маршруты из первой очереди могут применяться незамедлительно, если они доступны или запись истории для недоступного маршрута может быть устранена.

Когда наступает время продвижения списков, должна обрабатываться первая очередь с перемещением циклического списка. Используя массив и смещение в качестве циклического массива (см. параграф 4.8.6), приведенный ниже алгоритм повторяют каждые delta-reuse секунд.

```
1.  установить указатель на текущую нулевую очередь и обнулить элемент списка заголовков
2.  установить offset = modulo reuse-list-size (offset + 1), перемещая тем самым очередь
    в циклическом списке
3.  если (сохраненный указатель на заголовок не пуст)
    для каждого элемента {
      sett-diff = t-now - t-updated
      set figure-of-merit = figure-of-merit * decay-array-ok[t-diff]
      sett-updated = t-now
      если ( figure-of-merit < reuse)
        снова использовать маршрут
      else
        поместить маршрут в другой список (см. параграф 4.8.6)
    }
```

Значение головы нулевого списка будет сохраняться, а сам элемент массива — обнуляться. Головы списков будут продвигаться путем инкрементирования смещения. Начиная с сохраненной головы старого нулевого списка, каждый маршрут будет оцениваться заново и использоваться, полностью уничтожаться или заново помещаться в очередь для последующего использования. Если маршрут используется, он должен трактоваться, как полученный из нового маршрутного анонса (см. параграф 4.8.3).

5 Опыт реализации

Первыми реализациями демпфирования маршрутных осцилляций (route flap damping) были демон rsd[1], разработанный Ramesh Govindan (ISI) и реализация Cisco IOS от Ravi Chandra. Обе реализации стали доступны в 1995 и широко использовались. Демон rsd применялся в серверах маршрутов финансируемых NSF точках доступа в сеть NAP[2] и других основных узлах Internet. Версия Cisco IOS использовалась Internet-провайдерами по всему миру. Реализация rsd была встроена в демон gated (see http://www.gated.org) и доступна на коммерческих маршрутизаторах, использующих gated.

Опыт использования систем демпфирования осцилляций маршрутов BGP составляет уже более 2 лет. В развернутых системах обнаружены некоторые проблемы. До сих пор эти проблемы решались за счет тщательной реализации алгоритма и осторожного развертывания систем. В некоторых топологиях координированной развертывание механизма демпфирования может оказаться полезным и во всех случаях раскрытие информации о применении системы демпфирования осцилляций и параметрах механизма обеспечивает значимые преимущества при решении проблем отладки связности.

Некоторые из проблем были связаны со скрытыми ошибками в реализациях. Демпфирование маршрутов никогда не следует применять для маршрутов, полученных от IBGP, поскольку такое демпфирование может приводить к возникновению устойчивых маршрутных петель. Несогласованность маршрутов IBGP внутри AS легко может приводить к возникновению петель. Демпфирование полученных от IBGP маршрутов как раз и будет приводить к таким несогласованностям. Следовательно, разработчикам следует запрещать активизацию демпфирования маршрутов для партнеров IBGP.

Наказания за нестабильность следует применять лишь в случаях удаления или замены маршрутов, но не при их добавлении. При согласованном применении параметров демпфирования такое ограничение будет приводить к предпочтительности стабильного вторичного пути перед нестабильным основном в результат демпфирования основного маршрута вблизи его источника.

[1]Route server daemon — демон сервера маршрутов.
[2]Network Access Point.

В топологиях с множеством AS path к данному адресату осцилляции на основном пути могут приводить к демпфированию вторичного пути. Это может происходить в тех случаях, когда вблизи точки осцилляция их демпфирование не происходит или в удаленной AS используется более агрессивное демпфирование осцилляций. Проблему можно решить двумя способами. Можно выполнить демпфирование осцилляций вблизи их источника или согласовать параметры демпфирования. Кроме того, удаленная AS, где применяются более агрессивные параметры демпфирования осцилляций, может отметить наложение санкций на маршруты при изменении AS path, сохранив их только для полного отзыва маршрутов. Для этого реализация должна поддерживать специальную опцию, описанную в параграфе 4.4.3.

Осцилляции маршрутов следует подавлять вблизи источника. Для однодомных адресатов можно использовать статические маршруты. Другой вариант демпфирования может обеспечивать агрегирование маршрутов. Провайдерам следует демпфировать свои внутренние проблемы, однако демпфирование осцилляций для протоколов IGP еще не реализовано производителями маршрутизаторов. Провайдерам, использующим в своей сети множество AS, следует использовать демпфирование и между этими AS. Провайдерам также следует применять демпфирование для смежных AS.

Демпфирование позволяет ограничить распространение избыточных переключений маршрутов в сильно перемежающихся средах. После решения проблемы состояние демпфирования, соответствующее префиксам, для которых применялось демпфирование, может быть просто сброшено вручную. Провайдерам следует использовать ручной сброс для конкретных префиксов или AS path по запросам других провайдеров, сопровождаемым гарантиями решения соответствующих проблем.

Демпфируя свою маршрутную информацию, провайдеры могут снизить свои потребности в запросах к другим провайдерам для сброса использованного в их отношении демпфирования. Провайдерам следует применять упреждающие меры и мониторинг демпфируемых префиксов и путей в дополнение к мониторингу состояний каналов и сессий BGP.

Благодарности

Этот документ и работа в целом не были бы завершены без советов, комментариев и поддержки Yakov Rekhter (Cisco). Dennis Ferguson (MCI) представил описания алгоритмов в реализации gated BGP, а также множество идей и комментариев. David Bolen (ANS) и Jordan Becker (ANS) внесли важные комментарии, особенно в части ранних моделей. Прошло четыре года между представлением изначального чернового варианта в BGP WG (октябрь 1993 г.) и выходом данного документа. За время работы был обретен значительный опыт применения двух реализаций, развернутых в 1995 г. Одна из них была предложена Ramesh Govindan (ISI) для проекта NSF Routing Arbiter, а вторая - Ravi Chandra (Cisco). Sean Doran (Sprintlink) и Serpil Bayraktar (ANS) были среди первых независимых тестировщиков предварительной реализации Cisco. Значимые комментарии и обратная связь были представлены многими членами рабочих групп IETF IDR WG и RIPE Routing, а также NANOG и IEPG.

Благодарим также Rob Coltun (Fore Systems), Sanjay Wadhwa (Fore), John Scudder (IENG), Eric Bennet (IENG) и Jayesh Bhatt (Bay Networks) за обнаружение ошибок в более свежих реализациях. Эти ошибки в деталях, внесенных после завершения двух ранних реализаций. Спасибо Vern Paxson за внимательное рассмотрение документа и множество внесенных уточнений.

Литература

[1] Gross, P., and Y. Rekhter, "Application of the border gateway protocol in the internet", RFC 1268, October 1991.

[2] ISO/IEC. ISO/IEC 10747 - information technology - telecommunications and information exchange between systems - protocol for exchange of inter-domain routeing information among intermediate systems to support forwarding of iso 8473 pdus. Technical report, International Organization for Standardization, August 1994. ftp://merit.edu/pub/iso/idrp.ps.gz[1].

[3] Lougheed, K., and Y. Rekhter, "A border gateway protocol 3 (BGP-3)", RFC 1267, October 1991.

[4] Rekhter, Y., and P. Gross, "Application of the border gateway protocol in the internet", RFC 1772, March 1995.

[5] Rekhter, Y., and T. Li, "A border gateway protocol 4 (BGP-4)", RFC 1771, March 1995.

[6] Rekhter, Y., and C. Topolcic,"Exchanging routing information across provider boundaries in the CIDR environment", RFC 1520, September 1993.

[7] Traina, P., "BGP-4 protocol analysis", RFC 1774[2], March 1995.

[8] Traina, P., "Experience with the BGP-4 protocol", RFC 1773, March 1995.

Вопросы безопасности

Описанная в документе практика не вносит дополнительного снижения уровня безопасности протоколов маршрутизации. Атаки на отказ служб возможны и в имеющейся незащищенной среде маршрутизации, но описанная практика может лишь способствовать сохранению таких атак, не оказывая влияния на методы защиты и определения источников атак.

Адреса авторов

Curtis Villamizar

ANS

EMail: curtis@ans.net

[1]Приведенная в оригинале ссылка устарела. На момент перевода документ был доступен по ссылке ftp://ftp.merit.edu/oldmerit/pub/iso/idpr-fnl.tar.gz. *Прим. перев.*
[2]Этот документ устарел и заменен RFC 4274. *Прим. перев.*

Ravi Chandra

Cisco Systems

EMail: rchandra@cisco.com

Ramesh Govindan

ISI

EMail: govindan@isi.edu

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

Полное заявление авторских прав