

## СПОСОБ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ В ПЕРЕДАЧИ ИНФОРМАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЯХ

Национальный технический университет Украины «КПИ»

*Предложен новый способ разделения потоков трафика на граничном узле MPLS, который позволяет повысить эффективность в передаче информации в телекоммуникационных сетях*

### **Введение**

Расширение сферы использования компьютерных сетей (КС) явилось причиной создания нового класса сети – объединенных, которые представляют собой объединение различных сетей в единую телекоммуникационную сеть (ТС).

Инфраструктура подобной ТС характеризуется значительной сложностью, как по своей организации, так и по управлению. В связи с этим возникает необходимость в разработке и анализе новых интеллектуальных технологий в управлении сетевой инфраструктурой, которые гарантируют заданное качество обслуживания при минимальных затратах сетевых ресурсов.

Традиционные способы и протоколы маршрутизации, основанные на обработке пакетов сетевого уровня, не могут эффективно решать эти поставленные задачи в современных объединенных сетях.

### **Постановка задачи**

В статье [1] приведены результаты исследования многопротокольной коммутации по меткам (*MPLS*) и сравнение ее с традиционными способами маршрутизации. Выявлено, что *MPLS* ходя и имеет базовые средства для решения многих задач, возникающих при построении и использовании объединенных КС, но данная технология предполагает перенос основных функций по определенному пути прохождения пакета через сеть на граничные маршрутизаторы, что усложняет их реализацию и приводит к необходимости создания более адаптивной маршрутизации.

Таким образом, использование технологии *MPLS* позволяет повысить эффективность функционирования в ТС за

счет усовершенствования способа распределения трафика на граничных узлах.

### **Обзор существующих методов решения**

Технология *MPLS* предоставляет базовые средства, которые позволяют реализовать эффективное управление трафиком.

Явные пути коммутации по меткам (*LSP*) и возможность гибкого определения классов *FEC* (основываясь на таких данных заголовков пакетов, как адреса отправителя и получателя, порты и другие поля заголовка) направлены на устранение первого недостатка.

Маршрутизация с ограничениями направлена на устранение второго недостатка.

Пути *LSP* могут точно определять путь для пакетов, принадлежащих некоторому классу *FEC*, между входной и выходной точкой сети *MPLS*. Следовательно, переключение потока пакетов, принадлежащих одному классу *FEC*, с одного пути *LSP* на другой затрагивает только данный трафик, при условии, что перегрузки нового канала не произойдет. Таким образом в сетях *MPLS* преодолевается третий недостаток стандартных *IP*-сетей.

Последняя проблема, не позволяющая эффективно применять управление трафиком в *IP*-сетях, полностью не решена и в сетях многопротокольной коммутации по меткам, и один из способов её решения рассмотрен в данной работе.

В *MPLS* механизмы распределения трафика могут зависеть от времени или состояния. В случае механизма временной зависимости для предварительного задания *LSP* и назначения трафика ис-

пользуется информация о периодических изменениях трафика. Предварительно заданная схема *LSP* обычно изменяется на протяжении достаточно длительного промежутка времени (например, суток). Механизмы с использованием временной зависимости не пытаются приспособиться к непредсказуемым изменениям трафика или состояния сети. Примером может быть алгоритм оптимизации, описанный в [2], где входными данными системы является матрица трафика и требования *QoS*.

Так как случаются существенные фактические изменения трафика, которые не могут быть предсказаны, используя информацию о прошедших периодах времени, механизм зависимости о времени может не предотвратить значительный дисбаланс в нагрузке и перегрузку каналов. В такой ситуации может быть использован ситуационно-зависимый механизм, который предусматривает применение адаптивного назначения трафика по установленным *LSP* в соответствии с текущим состоянием сети, основанным на коэффициенте использования, задержке пакетов, потере пакетов.

Отметим некоторые особенности разрабатываемого алгоритма, и требования, которым он должен соответствовать:

- алгоритм является распределенным алгоритмом балансировки нагрузки;
- не требуется нового аппаратного обеспечения;
- не требуется нового программного обеспечения на промежуточных узлах (узлах *LSR*);
- изменяется только алгоритм обработки пакетов на граничных узлах сети *MPLS* (узлах *LER*);
- не изменяется протокол обмена метками;
- минимизируется переупорядочивание пакетов *TCP*-соединений;
- матрица трафика не должна быть известна заранее.

При использовании алгоритма предполагается, что существуют несколько установленных явных *LSP* путей (например, вручную или с помощью протокола *CR-LDP* или *RSVP-TE*) между входным узлом и выходным узлом в сети *MPLS*.

Такая конфигурация часто встречается в крупных корпоративных сетях или сетях провайдеров, когда между двумя узлами существует более одного пути, возможно с разными метриками. Одной из задач управления трафиком на входящем узле является оптимальное распределение трафика по этим нескольким путям, например такое, что нагрузка на каналы передачи данных будет равномерной. Рассматривая эту задачу в обобщенном виде, можно вывести функцию стоимости использования данного пути, зависящую от его загрузки и других переменных.

Так как для трафика, требующего высокого качества обслуживания и резервирования полосы пропускания, применяются специальные методы установки путей *LSP*, данный алгоритм предназначен для трафика, который не требует выделения полосы пропускания является трафиком, который нужно доставить по возможности наилучшим образом (*best-effort traffic*). Следует отметить, что такой трафик является значительной, если не наибольшей частью всего трафика в современных сетях.

На рис. 1 показан обобщенный пример сети *MPLS* с несколькими парами входных и выходных узлов, которые являются граничными узлами сети *MPLS*.

Предлагаемый алгоритм будет выполняться на узлах  $I_1..I_n$ .

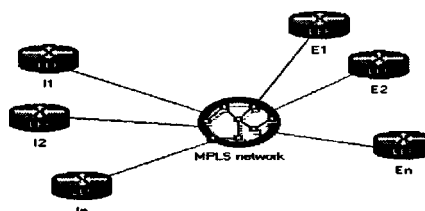


Рис. 1.

На выходных узлах  $E1..En$  также потребуется внести незначительные изменения в программное обеспечение, реализующее обработку пакетов. Внутренняя структура промежуточных узлов *LSR* и их программное обеспечение для обработки помеченных пакетов не меняется, как и протоколы распространения меток, используемые при первоначальной установке путей между *LSR*.

Облако сети *MPLS*, как было рассмотрено в обзоре, состоит из маршрути-

затов коммутации по меткам и физических каналов связи между ними. В сети может использоваться любой протокол канального уровня, а также любой протокол сетевого уровня. Множество путей *LSP* и маршрутизаторов *LSR* представляют собой виртуальную топологию, организованную поверх физической сети. При этом различные пути *LSP* могут разделять одни и те же физические каналы передачи данных между маршрутизаторами *LSR*. Поэтому при разработке алгоритма необходимо учитывать при определении загрузки каналов кроме потоков данных, передающихся непосредственно от выбранного входного узла к выходному узлу, еще и трафик, проходящий через разделяемые каналы и узлы от других входных узлов.

Рис. 2 показывает функциональную схему блока распределения трафика, расположенного во входном узле.

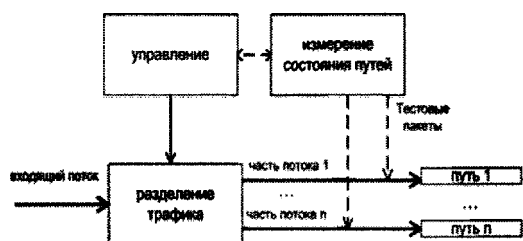


Рис. 2.

Входящий трафик попадает процедуру фильтрации и распределения, целью которой является упрощение переключения трафика среди *LSP* способом, уменьшающим возможности прихода пакетов к получателю не по порядку. Этому механизму не нужна статистика требований трафика или информация о состоянии потока. В процедуре распределения трафика определяется, как и когда переключать трафик между *LSP*. Это делается на основе статистики *LSP*, которая получается из измерений с использованием тестовых пакетов. Процедура распределения трафика состоит из двух этапов: этап мониторинга и этап балансировки нагрузки. На этапе мониторинга в случае обнаружения значительного и постоянного изменения состояния сети выполняется переход на этап балансировки. На этапе балансировки алгоритм пытается выровнять нагрузку среди *LSP*. Как только нагрузка выровни-

вается, алгоритм переходит на этап мониторинга и весь процесс повторяется.

Роль процедуры измерения и анализа заключается в получении статистики *LSP* в одну сторону, такой как задержка пакетов и потеря пакетов. Это делается путем периодической посылки тестовых пакетов от входного узла к выходному узлу и обратно. Тесты могут проводиться для каждого класса, то есть тестовые пакеты имеют тот же значение поля "тип сервиса" в заголовке пакета, что и класс распределяемого трафика. Основываясь на информации о возвращении тестовых пакетов, входной узел может считать *LSP*-статистику в одну сторону. Оценочная функция *LSP* статистики может быть получена эффективно и надежно с помощью технологий приближения и самонастройки. Эти технологии предоставляют динамический механизм посылки тестовых пакетов так, что при изменении условий трафика автоматически выбирается наименьшее их количество, обеспечивающее нужный уровень точности. Недавние измерения в *Internet* показали небольшие колебания агрегатного трафика в каналах за 5-ти минутные интервалы [3]. Такие условия, когда статистика трафика изменяется относительно медленно (значительно дольше, чем время обращения пакета между входным и выходным узлом), способствуют распределению трафика и балансировке нагрузки, основанной на измерениях статистики *LSP*.

На основании анализа математических моделей разделения трафика [4], рассмотрим различные способы разделения трафика, поступающего на входной маршрутизатор сети *MPLS*. Надо отметить, для получения полностью равномерной (или требуемой) загрузки каналов передачи данных, общий поток пакетов необходимо разделить на множество более мелких потоков, вплоть до разбиения на отдельные потоки данных для каждого соединения между прикладными процессами на узле отправителя и узле получателя (например, *TCP*-потоки). Однако такой уровень гранулярности приведет к необходимости поддерживать множество записей в памяти граничного маршрутизатора и существенно увеличит время вы-

бора соответствующего пути *LSP* для поступившего пакета. Более целесообразным является агрегирование множества потоков таких в один (например, основываясь на адресе сети отправителя и получателя).

Разделение трафика случайным образом на уровне пакетов - предусматривает разделение поступающих пакетов для передачи по одному из подходящих путей *LSP* случайным образом. При большой интенсивности поступления пакетов и правильной реализации генератора случайных чисел загрузка каждого из путей (и соответствующих физических каналов передачи) будет равномерной. Если в соответствии со стратегией управления трафиком необходимо загружать пути неравномерно, можно ввести

Главным преимуществом данного способа является его простота, и, как следствие этого, возможность реализации на его основе алгоритма с небольшой временной сложностью. Основным недостатком этого способа является то, что пакеты одного потока мелкой гранулярности (например, *TCP*-потока, определяемого парой отправитель-получатель и парой портов) могут быть направлены (и, скорее всего, будут направлены) по разным путям. При этом происходит переупорядочивание пакетов, принадлежащих одному *TCP*-потoku. Хотя протокол *TCP* гарантирует доставку пакетов на верхние уровни модели *OSI* в правильном порядке, переупорядочивание пакетов на сетевом уровне приводит к уменьшению размера *TCP*-окна (так как пришедший не вовремя пакет может посчитаться потерянным) и уменьшению скорости обмена данными между прикладными процессами на узле-отправителе и узле-получателе, или необходимостью увеличивать размеры буфера на узлах, установивших соединение.

Разделение в соответствии с циклической дисциплиной предполагает очередное направление поступающих пакетов для передачи через подходящие пути *LSP*. Как и при случайном разделении, данный способ прост в реализации и алгоритм на его основе имеет малую временную сложность. Его недостатками яв-

ляется переупорядочивание пакетов одного потока, а также некоторая вероятность неравномерной загрузки каналов передачи данных при определенных условиях (например, при существовании нескольких интенсивных входных потоков с пакетами разной длины, поступающими поочередно).

Еще одним преимуществом двух названных способов является также отсутствие необходимости поддерживать таблицы проверки соответствия пришедшего пакета тому или иному потоку.

Разделение на основе адреса отправителя и получателя является создание и поддержка в памяти маршрутизатора таблиц соответствия входящих потоков и предпочитаемых путей *LSP* для них. Ниже приведен пример такой таблицы.

Таблица 1. Соответствие потоков и *LSP*

<i>source IP</i>	<i>destination IP</i>	<i>interface</i>	<i>label</i>
195.245.194.127	212.40.44.3	1	3
195.245.194.127	161.2.3.1	2	2
195.245.194.142	212.40.44.3	1	3
61.1.23.4	161.2.3.1	2	2
61.1.23.4	212.40.44.3	1	3

В таблице может также быть дополнительное поле, в котором будет храниться информация об интенсивности данного потока (например, количество пакетов в секунду) или требуемой пропускной способности (байт в секунду). Эта информация будет собираться за некоторый период времени мониторинга данного потока (и постоянно обновляться), и использоваться при принятии решения о переключении данного потока на другой путь *LSP* при изменении состояния загруженности путей. Данная таблица может быть совмещена с таблицей соответствия *FEC* и *NHLFE*, но тогда при большом количестве активных потоков таблица займет большой объем памяти. Фактически, это приводит к определению класса *FEC* для каждого *TCP*-соединения, пакеты которого проходят через данный граничный маршрутизатор.

Преимуществом данного способа является возможность достаточно гибко (и при необходимости равномерно) распределять входящий трафик между несколькими подходящими путями *LSP*, соблюдая при этом условие пересылки пакетов одного соединения по одному *LSP*, а недостатками - высокие требования к объему памяти граничного маршрутизатора и снижение его быстродействия при поиске соответствующего пакету класса *FEC* в большой таблице. Кроме того, при некоторых условиях (например, при установке между некоторой парой узлов нескольких *TCP*-соединений, пакеты по которым передаются с большой интенсивностью) данный способ не позволяет оптимально распределить нагрузку.

Разделение на основе адреса отправителя и получателя, и номеров портов является модификацией предыдущего, но для идентификации потоков используются не только адреса отправителя и получателя, а еще и номера портов (в том случае, если установлено *TCP*-соединение). С одной стороны, это приводит к еще большим требованиям к объему памяти и снижению быстродействия (вследствие увеличения размеров таблиц), что является существенным недостатком, с другой стороны, появляется возможность более гибко распределить нагрузку. Как и в предыдущем способе, при его применении переупорядочивания пакетов не происходит.

Разделение по значению *hash*-функции от заголовка пакетов предполагает вычисление значения *hash*-функции, используя в качестве аргумента некоторые поля *IP*-пакета (например, адрес отправителя, получателя и номера портов) [3]. Как и в предыдущем способе, необходимо создать и постоянно обновлять таблицы в памяти маршрутизатора (в которых содержатся значения *hash*-функций и номера соответствующих предпочитаемых путей *LSP*), но размер их можно выбрать в соответствии с объемом доступной памяти и требуемым быстродействием. Это решение является компромиссом между возможностью гибко распределять потоки между путями *LSP* и требованием к быстродействию граничного маршрути-

затора, а также требуемым объемом памяти. Так как в качестве аргументов *hash*-функции выбираются поля заголовка пакета, которые постоянны для одного *TCP*-соединения, переупорядочивания пакетов при использовании этого способа не происходит. Среди недостатков существующих способов можно выделить следующие:

- переупорядочивание пакетов *TCP*-потока,
- большой объем памяти для хранения таблиц,
- невозможность гибкой настройки.

Во всех рассмотренных выше способах разделения потоков для обеспечения поступления пакетов на исходящий узел в правильном порядке необходимо анализировать значение заголовка пакета. Хотя такая процедура в любом случае выполняется на граничном маршрутизаторе сети *MPLS* для определения класса *FEC*, соответствующего данному пакету, при использовании методов разделения потоков пакетов по адресам отправителя и получателя приходится выполнять просматривать две таблицы: одна из них устанавливает соответствие между заголовком пакета и классом *FEC*, а другая - между значениями заголовка пакетов и подходящими путями *LSP*. Было бы целесообразным объединить эти две таблицы, но это приводит фактически к определению множества классов *FEC*, вплоть до отдельного класса для каждого *TCP*-потока. Из-за ограниченного объема памяти маршрутизатора, и снижения его производительности при поиске в таблицах с большим количеством записей, этот подход является неприемлемым.

### Решение

Суть предложенного модифицированного способа состоит в следующем: для хранения записей соответствия *FEC* и *LSP* используется модифицированная таблица (Таблица 2). Кроме стандартных полей (*IP*-адреса отправителя и получателя, адреса портов, а также исходящей метки и исходящего интерфейса, определяющего *LSP*), в таблице присутствуют такие поля:

- значение типа протокола (*TCP*, *UDP*),
- счетчик байт данного потока за некоторый промежуток времени ( $\Delta t$ ),
- начальное значение *hash*-функции, которая берется от полей *IP*-пакета, одинаковых для одного *TCP*-потока.

Для упрощения вычисления *hash*-функции берется сумма по модулю 28, таким образом, максимально возможное количество записей для одного предварительно определенного класса *FEC* равняется 256.

В качестве адреса отправителя и получателя может использоваться как адреса подсети, так и отдельные узлы. Это по-

зволяет осуществить агрегирование редко используемых комбинаций адресов, объединив их в одну запись.

Специальный формат данной таблицы позволяет использовать как настройку классов *FEC* вручную администратором сети, так и эффективное автоматическое распределение трафика с помощью специального программного модуля, осуществляющего мониторинг загрузки путей и разделяющие более обобщенные записи *FEC* на несколько более определенных при возрастании интенсивности трафика этого класса, и производящего агрегирование при уменьшении его интенсивности.

Таблица 2. Модифицированная таблица соответствия потоков и *LSP*

Proto	dst IP/mask	src IP/mask	src port	dst port	hash		interface	label	bytes
					from	to			
<i>TCP</i>	195.245.194/24	any	any	any	0	63	1	2	35354
<i>TCP</i>	195.245.194/24	any	any	any	64	127	2	3	32532
<i>TCP</i>	195.245.194/24	any	any	any	128	255	3	3	35325
<i>UDP</i>	12.2.3.1	80.54.1.20	any	any	0	255	5	1	14143
<i>UDP</i>	62/8	any	any	80	0	255	1	5	3000
any	61.1.2.3	212.55.5.1	1024	22	0	255	8	2	32320
any	212.11.7.2	any	any	any	0	255	5	2	1500

Например, запись с адресом получателя 212.40.4/24 и неопределенными адресами портов может быть разделена на две записи: 212.40.4.0/25 и 212.40.4.128/25

Для упрощения процедуры разделения обобщенной записи на более определенные, а также увеличения производительности маршрутизаторов предлагается использовать *hash*-функцию от значений полей *IP*-заголовка.

Надо отметить, что, как правило, входной граничный маршрутизатор соединен путями *LSP* с несколькими выходными маршрутизаторами, а не с одним, и пакеты, поступающие на него, не обязательно покидают сеть *MPLS* в одном и том же выходном граничном маршрутизаторе. Поэтому, чтобы поступающие пакеты не были направлены по неправильному пути из-за совпадения значений *hash*-функций у пакетов, принадлежащих

разным классам *FEC* (что вполне возможно при наборе всего из 256 значений), некоторые поля таблицы являются обязательными для рассмотрения. Таким полем, прежде всего, является адрес получателя (заметим, что в традиционной *IP*-маршрутизации именно это поле пакета определяет следующий узел для передачи пакета). Поэтому предлагается использовать флаг (принимающий значения 0 или 1) для полей *dst IP*, *src IP*, *dst port* и *src port*. При установке флага в 1 поле не может игнорироваться при просмотре и выборе подходящего *LSP*, и проверка заголовка пакета на соответствие его полей данным значениям обязательна. Если же значения флага равно 0, решение принимается на основании значения *hash*-функции.

Поле, содержащее значение *hash*-функции, разделено на две части. Это

сделано для того, чтобы уменьшить количество записей в таблице, предоставив возможность использовать диапазоны значений. Например, вместо использования 256 записей для двух диапазонов значений (Таблица 3), используется всего 2 записи (Таблица 4). Хотя поля *src IP*, *src port*, *dst port* содержат значение "any" (указывает на то, что любое значение подходит), при определении *hash-*

функции действительные значения данных полей из заголовков пакетов будут учитываться. Содержимое данных таблиц одинаково определяет соответствие между классом *FEC* и путем *LSP*. Так, для значений *hash*-функции от 0 до 127 используется путь *LSP* с исходящим интерфейсом №1 и меткой 5, а для значений *hash*-функции от 128 до 255- используется другой *LSP* (интерфейс №2 и метка 4)

Таблица 3.

<i>Proto</i>	<i>dst IP/mask</i>	<i>src IP/mask</i>	<i>src port</i>	<i>dst port</i>	<i>hash</i>	<i>interface</i>	<i>label</i>	<i>bytes</i>
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	0	1	5	5253255
...	...	...	...	...	...	...	...	...
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	127	1	5	5543345
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	128	2	4	6754544
...	...	...	...	...	...	...	...	...
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	255	2	4	4355994

Диапазоны значений *hash*-функций для одного набора обязательных полей

(отмеченных флагом со значением 1) не должны пересекаться.

Таблица 4.

<i>Proto</i>	<i>dst IP/mask</i>	<i>src IP/mask</i>	<i>src port</i>	<i>dst port</i>	<i>hash</i>		<i>interface</i>	<i>label</i>	<i>bytes</i>
					<i>from</i>	<i>to</i>			
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	0	127	1	5	79743644
<i>TCP</i>	195.245.194.142	<i>any</i>	<i>any</i>	<i>any</i>	128	255	2	4	46465456

### Выводы

Предложен новый способ разделения потоков трафика на граничном узле *MPLS*, который поддерживает архитектуру интегрированных служб и обеспечивает равномерную загрузку сети при адаптивной маршрутизации и тем самым повышает эффективность управления в *ТС*.

Среди достоинств предложенного способа можно выделить такие: невысокие требования к объему памяти для размещения таблиц (по сравнению со способом распределения на основе адресов отправителя и получателя), отсутствие переупорядочивания *TCP*-пакетов (что является

недостатком способа распределения в соответствии с циклической дисциплиной), и гибкость настройки.

### Список литературы

1. Семенов Ю.А. Введение в *MPLS*, *TE* и *QoS* // <http://book.itper.ru>.
2. Mitra D., Ramakrishnan K. A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks // *Globecom'99*. – 1999.
3. Elwalid A., Jin C., Low S., Widjaja I. *MATE: MPLS Adaptive Traffic Engineering* // *IEEE INFOCOM* – 2003.
4. Elwalid A., Jin C., Low S., Widjaja I. *MATE: MPLS Adaptive Traffic Engineering* // *IEEE INFOCOM-2003*.