

УДК 621.372

К.М. Руккас, К.А. Овчинников

Харьковский национальный университет радиоэлектроники, Харьков

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ МАРШРУТИЗАЦИИ В СЕТЯХ MPLS

Рассмотрены основные алгоритмы маршрутизации, использующиеся в сетях с коммутацией по меткам. Приведена математическая постановка задачи маршрутизации с минимальными взаимными влияниями передаваемых потоков. Приведены алгоритмы вычисления оптимальных маршрутов для протоколов на основе алгоритма минимума взаимных влияний. Оценена вычислительная сложность алгоритмов маршрутизации в MPLS. Приведен сравнительный анализ эффективности использования доступных канальных ресурсов протоколами маршрутизации в MPLS на основании результатов имитационного моделирования.

**Ключевые слова:** маршрутизация, оптимизация, коммутация по меткам, алгоритм минимума интерференции, имитационное моделирование.

### Введение

Ключевым недостатком протоколов маршрутизации, использующих алгоритм поиска кратчайшего пути, является возникновение коллизий и потерь трафика на участках, являющихся частью кратчайших путей для других пар источник-получатель. Данное условие особенно важно для сетей MPLS, имеющих иерархическую структуру (разделение сети на «периферию», где сконцентрированы основные функции управления и «ядро», обеспечивающее исключительно продвижение пакетов через сеть). Для устранения коллизий был предложен алгоритм минимума взаимных влияний MIRA (Minimum Interference Routing Algorithm).

### Постановка задачи минимума взаимных влияний

Обозначим через  $G(N,L,V)$  граф сети, где  $N$  – множество маршрутизаторов (вершин),  $L$  – множество каналов передачи (дуг) и  $V$  – пропускные способности каналов. Пусть  $n$  – количество узлов, а  $m$  – количество каналов. Значения пропускных способностей и величина маршрутизируемого трафика считаются целыми положительными числами. Обозначим через  $P$  множество пар маршрутизаторов (источник и получатель) мощностью  $p$ . Элемент этого множества будем обозначать как  $(s,d)$ . Путь коммутации меток LSP может быть проложен только между этими парами. Матрица  $M$  представляет собой матрицу сопряженности между узлами и дугами. Строки матрицы соответствуют узлам, а столбцы – дугам. В каждом столбце матрицы существуют только два неотрицательных значения: столбец, соответствующий дуге  $(v,w)$  равен «+1» в строке  $v$ , «-1» в строке  $w$  и «0» для всех остальных строк. Вектор  $x^{sd}$  размерности  $m$  соответствует паре  $(s,d) \in P$ . Обозначим через  $\theta_{sd}$  сумму свободных пропускных способностей всех путей между парой

$(s,d)$ . Вектор  $R$  имеет размерность  $m$  и содержит значения доступных пропускных способностей всех каналов связи. В начальный момент  $R$  равен  $V$ . Вектор  $e^{sd}$  размерности  $n$  содержит сведения об источнике и получателе трафика, элемент  $s$  этого вектора равен «+1», элемент  $d$  равен «-1». Предполагается, что в один момент времени поступает только один запрос. Под запросом понимается требуемая полоса пропускания (целое неотрицательное значение количества единиц пропускной способности)  $D$  для передачи трафика между маршрутизаторами  $a$  и  $b$ ,  $(a,b) \in P$ .

Положим, задача максимального потока решается для пары  $(s,d)$  с учетом доступных пропускных способностей каналов. Максимальный поток равен максимальному значению пропускной способности, доступной между узлами  $s$  и  $d$ .

Под взаимным влиянием будем понимать уменьшение максимально доступной полосы между данной парой в связи с передачей некоторого трафика по выбранному пути. Поскольку трафик, который надо будет передать между этой парой в будущем, неизвестен, задачей маршрутизации является выбор пути с условием максимизации минимально доступной полосы пропускания между всеми парами  $P$  за исключением рассматриваемого пути, т.е.  $P \setminus (a,b)$ .

Таким образом MAX-MIN-MAX задача сводится к задаче целочисленного программирования:

$$\max z$$

$$Mx^{sd} = \theta_{sd} e^{sd}; \quad \forall (s,d) \in P \setminus (a,b); \quad (1)$$

$$Mx^{ab} = -De^{ab}; \quad (2)$$

$$x^{sd} + x^{ab} \leq R; \quad \forall (s,d) \in P \setminus (a,b); \quad (3)$$

$$z \leq \alpha_{sd} \theta_{sd}; \quad \forall (s,d) \in P \setminus (a,b); \quad (4)$$

$$x^{sd} \geq 0; \quad \forall (s,d) \in P; \quad (5)$$

$$x^{ab} \in \{0, d\}^m. \quad (6)$$

Выражение (1) отражает задачу максимального потока для всех пар в  $P$  за исключением текущей пары  $(a,b)$ .  $\theta_{sd}$  – значение максимального потока для пары  $(s,d)$ . Выражение (2) указывает на то, что  $D$  единиц трафика должны быть переданы между парой  $(a,b)$ . Выражение (3) связывает переменные между собой и обеспечивает условие использования пропускных способностей, не задействованных для обслуживания текущего трафика. Условия (5) и (6) означают, что трафик должен быть маршрутизирован только по одному пути между  $a$  и  $b$ .

Следует отметить, что единственной оптимизируемой переменной выступает минимальное значение максимального потока, и оно не зависит от других потоков. Альтернативной формулировкой задачи служит максимизация взвешенной суммы максимальных потоков (WSUM-MAX):

$$\max \sum_{(s,d) \in P \setminus (a,b)} \alpha_{s,d} \theta_{s,d}, \quad (7)$$

где  $\alpha_{s,d}$  – вес пары  $(s,d)$ , который задается администратором для обозначения относительной важности пары. Если вес не задан, то все пары считаются равновесными. Т.о. из формулировки задачи исключается ограничение (4) и она уже не является задачей целочисленного программирования.

Для решения задачи WSUM-MAX необходимо сначала определить веса дуг, а затем найти кратчайший путь с учетом полученных весов дуг. Дополнительные ограничения для маршрутов можно применить уже после нахождения кратчайшего пути. Веса дуг подбираются таким образом, чтобы маршрутизируемый поток минимально пересекался с будущими потоками в сети. Для решения задачи определения весов дуг исключается ограничение (2), т.е. требование передачи трафика в  $D$  единиц пропускной способности между  $a$  и  $b$ . Это эквивалентно случаю  $D = 0$ . Затем решается задача минимального потока для всех пар в сети  $(s,d) \in P \setminus (a,b)$ . Обозначим результат решения этой задачи для каждой пары через  $\hat{\theta}_{sd}$ . Тогда оптимальным решением задачи WSUM-MAX для  $D = 0$  будет  $\alpha_{sd} \hat{\theta}_{sd}$ . Теперь, в случае передачи трафика через канал, его свободная пропускная способность уменьшится на величину  $D$ , что приведет к отклонению текущего решения задачи WSUM-MAX от оптимума. Таким образом, вес дуги есть отношение изменения значения оптимального решения WSUM-MAX к изменению значения свободной пропускной способности канала. Обозначим через  $\partial \hat{\theta}_{sd} / \partial R(l)$  изменение значения максимального потока пары  $(s,d)$  при постепенном изменении значения свободной пропускной способности канала  $l$ . Частная производная отражает уменьшение значения максимального потока для

пары  $(s,d)$  при передаче трафика через канал  $l$ . Тогда вес канала  $\omega(l)$  равен

$$\omega(l) = \sum_{(s,d) \in P \setminus (a,b)} \alpha_{sd} \frac{\partial \hat{\theta}_{sd}}{\partial R(l)}. \quad (8)$$

Вес дуги (8) характеризует изменение значения целевой функции WSUM-MAX в зависимости от изменения текущей загрузки канала и не учитывает зависимость между каналами и условие передачи трафика между парой  $(a,b)$ . Особенностью данной задачи является сопоставление каждому максимальному потоку минимального сечения (minimum cut).

Максимальный поток отдельно взятой пары уменьшается при уменьшении пропускной способности любого канала из минимального сечения этой пары. В то же время уменьшение пропускной способности любого канала, не принадлежащего к минимальному сечению, влиять на значение максимального потока не будет.

### Minimum Interference Routing Algorithm

Введем определение критического канала. Канал является критическим для заданной пары  $(s,d)$  в случае если он принадлежит к любому из минимальных сечений этой пары. Минимальное сечение вычисляется на основе текущих значений свободной пропускной способности каналов. Обозначим множество критических каналов для пары  $(s,d)$  как  $C_{sd}$ . Исходя из сказанного выше, можно заключить:

$$\frac{\partial \hat{\theta}_{sd}}{\partial R(l)} = \begin{cases} 1, & l \in C_{sd} \\ 0, & l \notin C_{sd} \end{cases},$$

откуда

$$\omega(l) = \sum_{(s,d): l \in C_{sd}} \alpha_{sd}. \quad (9)$$

Таким образом, задача вычисления весов дуг сократилась до задачи нахождения множества критических каналов для всех пар  $(s,d)$ . Для заданной пары возможно существование нескольких минимальных сечений, тогда критическими являются все каналы, принадлежащие этим сечениям.

После вычисления весов дуг каналы со значением свободной полосы  $R(l) < D$  исключаются из топологии и маршрут вычисляется как кратчайший путь между  $a$  и  $b$ .

При вычислении кратчайшего пути веса дуг приравниваются к  $\omega(l)$ . Для обеспечения выбора кратчайшего пути по критерию минимального количества переприемов через каналы, имеющие нулевой вес (т.е. не являющиеся критическими), их вес принимается равным некоторому небольшому неотрицательному значению.

Общее описание алгоритма [2]:

*Исходные данные:* граф  $G(N,L)$  и множество значений свободных пропускных способностей каналов  $B$ , узел-источник  $a$  и узел-получатель  $b$ , требуемая полоса пропускания  $D$ .

*Выходные данные:* маршрут между  $a$  и  $b$ , имеющий достаточную пропускную способность  $D$ .

*Алгоритм:*

1. Вычисление значений максимального потока для  $\forall(s,d) \in P \setminus (a,b)$ .

2. Вычисление множества критических каналов  $C_{sd}$  для  $\forall(s,d) \in P \setminus (a,b)$ .

3. Определение весов дуг

$$\omega(l) = \sum_{(s,d) \in C_{sd}} \alpha_{sd}, \quad \forall l \in L.$$

4. Исключение из топологии всех каналов с недостаточной полосой  $R(l) < D$ .

5. Вычисление кратчайшего пути по алгоритму Дейкстры с использованием весов дуг  $\omega(l)$ .

6. Маршрутизация потока в  $D$  единиц пропускной способности и пересчет значений свободной пропускной способности.

### Dynamic On-line Routing Algorithm

Алгоритм маршрутизации в реальном времени DORA (Dynamic On-line Routing Algorithm) является развитием алгоритма MIRA. Главными задачами, решаемыми алгоритмом, являются поиск маршрута с максимальным значением свободной пропускной способности и минимизация использования каналов, потенциально являющихся частью других маршрутов.

Алгоритм DORA осуществляет установление LSP с гарантированной пропускной способностью. Целевой функцией алгоритма является минимизация отказов в обслуживании трафика в связи с недостаточным количеством канальных ресурсов.

Алгоритм состоит из двух частей. Первая часть представляет реакцию на изменение топологии сети и осуществляет поиск новых путей, а вторая – решает задачу поиска маршрута для трафика, поступившего на обслуживание.

Для первой части ключевым этапом является вычисление PPV (path potential value, относительная важность пути) для каждого канала каждой пары источник-получатель. PPV отражает вероятность того, что канал используется несколькими путями сразу.

Большее значение PPV соответствует большей вероятности того, что канал является частью других путей, и его использование нежелательно. Напротив, меньшее значение PPV указывает на то, что этот канал является приоритетным при выборе маршрута и степень влияния его загруженности на другие пути минимальна. Каждой паре  $(s,d)$  ставится в

соответствие вектор  $PPV_{sd}$  размерностью  $L$ , где  $L$  – количество каналов. Вычисление значений элементов  $PPV_{sd}$  базируется на частоте появления канала в непересекающемся множестве путей между парами.

Для каждой пары необходимо проверить, принадлежит ли этот канал множеству путей между  $(a,b)$ , уменьшая значение  $PPV_{sd}(l)$  на единицу каждый раз, когда он оказывается частью пути, и также определить, является ли канал частью множества путей между парой  $(s,d)$ , для  $\forall(s,d) \in P \setminus (a,b)$ , и увеличивать значение  $PPV_{sd}(l)$  на единицу каждый раз, когда канал является частью пути между  $(s,d)$ .

Вторая часть алгоритма использует значения  $PPV_{sd}(l)$  в сочетании с соответствующим значением свободной пропускной способности для вычисления веса канала. Значение веса канала регулируется параметром BWP (bandwidth proportion),  $BWP \in [0,1]$ , значение которого определяет степень влияния свободной пропускной способности на вес канала, а  $(1-BWP)$ , соответственно, определяет значимость  $PPV_{sd}(l)$  для веса текущего канала.

Общее описание алгоритма [3]:

Этап 1:

1. Для каждой пары  $(s,d)$  определяется множество всех возможных путей  $DP_{sd}$ . Одним из возможных вариантов является использование алгоритма Дейкстры для определения кратчайшего пути (в смысле минимума переприемов) между  $(s,d)$  и добавление этого пути в  $DP_{sd}$ , после чего все каналы, составляющие полученный кратчайший путь, удаляются из топологии, и цикл начинается заново, до тех пор, пока узел  $d$  для  $s$  не становится недоступным.

2. Для пары источник-получатель  $(a,b)$  создается массив нулевых элементов  $PPV_{sd}(a,b)$ , размерностью  $L$ , где  $L$  – количество каналов связи.

3. Для каждой пары  $(a,b)$ :

а) Для каждого канала сети  $l$  в случае, если этот канал является частью любого из путей, принадлежащих  $DP_{ab}$ , уменьшить  $PPV_{ab}(l)$  на 1.

б) Для  $\forall(s,d) \in P \setminus (a,b)$  проверить каждый канал  $l$  и определить количество путей  $X$ , для которых  $l$  является частью пути, после чего увеличить значение  $PPV_{sd}(l)$  на единицу.

4. Повторить шаги 3 и 4 для всех пар источник-получатель.

5. Для каждой пары  $(s,d)$  нормализовать все значения  $PPV_{sd}$  таким образом, чтобы минимальный элемент из всех пар в  $PPV$  был равен 0, а максимальное значение  $PPV$  – 100. Нормализованные значения  $PPV_{sd}(L)$  обозначим через  $NPPV_{sd}(L)$ .

Этап 2:

1. Предположим, что поступает запрос на передачу  $Y$  единиц пропускной способности между парой  $(s_1, d_1)$ . Исключаем все каналы, значение свободной пропускной способности которых не превышает  $Y$ .

2. Для каждого канала сети  $L$  определяем значение свободной пропускной способности  $RB(L)$ , берем обратное значение и нормализуем  $RB(L)-1$  в пределах от 0 до 100 таким образом, чтобы минимальное значение  $RB(L)-1$  равнялось 0, а максимальное – 100. Обозначим нормализованные значения  $RB(L)-1$  как  $NRB(L)$ .

3. Для пары  $(s_1, d_1)$  построить таблицу весов каналов  $LWT_{(s_1, d_1)}(L)$  по следующему правилу:

$$LWT_{(s_1, d_1)}(L) = NPPV_{(s_1, d_1)}(L) \times (1 - BWP) + NRB(L) \times BWP. \quad (10)$$

4. Для вычисления оптимального взвешенного пути между парой  $(s_1, d_1)$  применяется алгоритм Дейкстры.

Вычислительная сложность алгоритмов маршрутизации приведена в табл. 1.

Из приведенных алгоритмов SPF обладает минимальной вычислительной сложностью, но его производительность намного хуже других алгоритмов. Второй этап DORA обладает такой же вычис-

лительной сложностью, что и SPF и выполняется при каждом поступлении трафика на вход сети. Второй этап выполняется только при изменении топологии сети. Следует отметить, что при различных топологиях вычислительная сложность DORA может быть ниже, чем у SPF, но, тем не менее, всегда выше, чем у MIRA.

Таблица 1

Вычислительная сложность алгоритмов маршрутизации

Алгоритм	Вычислительная сложность
SPF	$O(M, N)$
MIRA	$O(N^5) + O(M^2)$
DORA этап 1	$O(N^3 M^2)$
DORA этап 2	$O(NM)$

### Экспериментальные данные

Сравнение эффективности рассмотренных алгоритмов маршрутизации было проведено при помощи имитационного моделирования. Топология моделируемой сети приведена на рис. 1.

Между каждой парой передавалось 5 видов трафика, время между двумя заявками и длительность распределены по экспоненциальному закону. Числовые характеристики трафика приведены в табл. 2.

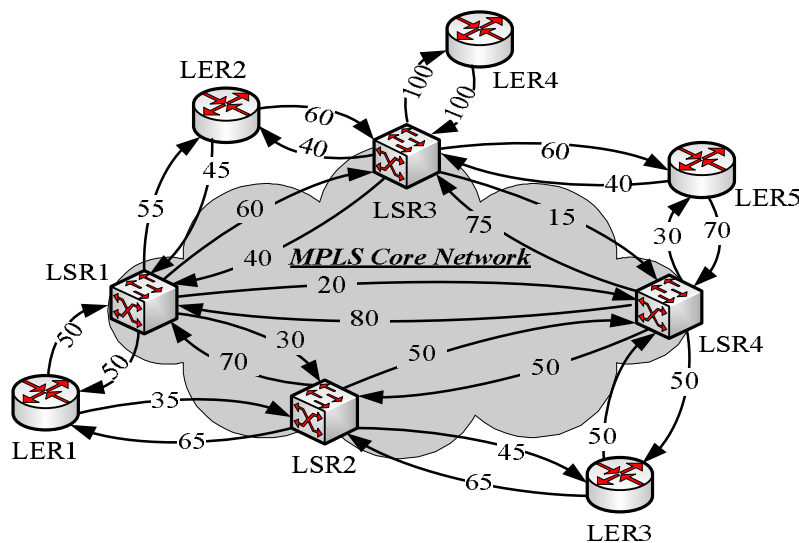


Рис. 1. Топология моделируемой сети

Таблица 2  
Характеристики трафика.

№ п/п	$B$ , Мбит/с	Появление, с	Длительность, с
1	0,01	5	120
2	0,5	10	60
3	1	20	90
4	2	60	30
5	5	120	60

Ограничения для протокола CSPF [3] принимались равными средней ожидаемой нагрузке на каждую пару граничных маршрутизаторов в 20 Мбит/с. Интервал расчета маршрутных таблиц составляет 30 секунд (для OSPF, CSPF и WPF).

В результате моделирования работы сети в течение 30 минут были получены следующие характеристики, представленные в табл. 3, 4.

При использовании протокола OSPF не использовались каналы LSR1–LSR2, LSR1–LSR4, LSR2–

LSR4, LSR4–LSR2, LSR4–LSR3. Протокол CSPF оставляет неиспользованным только канал LSR4–LSR2. WPF, MIRA и DORA используют все доступные каналы передачи моделируемой сети.

Таблица 3

Потери трафика  
из-за отсутствия свободной полосы пропускания

№ п/п	Потери OSPF, %	Потери WPF, %	Потери CSPF, %	Потери MIRA, %	Потери DORA, %
1	0,14	0,11	0,5	0,01	–
2	5,17	4,71	10,05	1,08	1,25
3	11,58	12,26	9,67	3,85	4,03
4	24,43	27,3	24,8	13,23	11,62
5	51	47,21	44,68	37,8	33,11

Таблица 4

Загруженность каналов

	min	max	среднее
OSPF	0	0,84	0,49
WPF	0	0,86	0,5
CSPF	0,22	0,73	0,58
MIRA	0,15	0,86	0,65
DORA	0,16	0,83	0,64

При использовании OSPF суммарные потери составили 10,5%, для WPF – 12,3% (в основном за счет самого требовательного к полосе пропускания трафика), CSPF – 7,8%, 4,45% и 3,9% для MIRA и DORA соответственно.

### Выводы

Как видно из полученных результатов имитационного моделирования, эффективность алгоритмов WPF и CSPF между собой отличается в малой

степени, также как и эффективность алгоритмов MIRA и DORA; в то же время алгоритмы группы реального времени позволяют значительно снизить коэффициент отказов и повысить эффективность использования каналов передачи.

Следует отметить, что эффективность CSPF сильно зависит от административно заданных значений ограничений на минимальную полосу пропускания, и при некорректном задании ограничений выбор маршрута может практически не отличаться от простого OSPF.

Также очевидно, что традиционные алгоритмы однопутевой маршрутизации на основе единичных метрик являются крайне неэффективными в плане использования доступных сетевых ресурсов. При выборе алгоритма маршрутизации, относящегося к группе реального времени, следует учитывать вычислительную сложность алгоритма.

### Список литературы

1. Kar K., *Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications* / K. Kar, M. Kodialam and T.V. Lakshmn // *IEEE Journal on Selected Areas in Communications: Quality of Service in the Internet*. – 2000. – Vol. 18, No. 12. – P. 921-940.
2. Boutaba R., *DORA: Efficient Routing for MPLS Traffic Engineering* / R. Boutaba, W. Szeto and Y. Iraqi // *IEEE Journal of Network and System Management*. – 2002. – Vol. 10, No. 3. – pp 309-305.
3. Ziegelmann M. *Constrained Shortest Paths and Related Problems: Constrained Network Optimization* / M. Ziegelmann // *VDM Verlag*. – Saarbrücken, Germany, 2007.

Поступила в редакцию 13.06.2012

**Рецензент:** д-р техн. наук, проф. Ю.И. Лосев, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

### ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ МАРШРУТИЗАЦІЇ В МЕРЕЖАХ MPLS

К.М. Руккас, К.А. Овчинников

*Розглянуто основні алгоритми маршрутизації, які використовуються в мережах з комутацією за мітками. Наведено математичну постановку задачі маршрутизації з мінімальним взаємним впливом переданого трафіку. Наведено алгоритми розрахунку оптимальних маршрутів для протоколів мінімуму взаємних впливів. Оцінена обчислювальна складність алгоритмів маршрутизації в MPLS. Наведено порівняльний аналіз ефективності використання доступних каналів ресурсів протоколами маршрутизації в MPLS на підставі результатів імітаційного моделювання.*

**Ключові слова:** маршрутизація, оптимізація, комутація за мітками, мінімум взаємних впливів, імітаційне моделювання.

### COMPARATIVE ANALYSIS OF ROUTING ALGORITHMS FOR MPLS NETWORKS

K.M. Rukkas, K.A. Ovchinnikov

*The basic routing algorithms for networks with label switching are considered. The mathematical formulation for minimum interference routing problem is given. Algorithms for calculating the optimal routes for minimum interference routing protocols are given. The computational complexity of algorithms for routing in MPLS is estimated. The simulation-based comparative analysis of the efficiency of the available channel resources distribution for main routing protocols in MPLS is given.*

**Keywords:** routing, optimization, label switching, minimum interference, network simulation.