

Моделирование двухуровневой маршрутизации в задаче последовательного заполнения сети потоками продуктов*

А. П. АФАНАСЬЕВ, Я. Р. ГРИНБЕРГ, И. И. КУРОЧКИН, А. В. КОРХ

Аннотация. Данная работа касается моделирования маршрутизации потоков данных в телекоммуникационных сетях. Рассматриваются результаты применения двухуровневой маршрутизации для организации работы существующих эвристических алгоритмов заполнения сети потоками продуктов последовательно поступающими в сеть. При этом даны два алгоритма реализации двухуровневой маршрутизации, основанных как на эвристических предположениях, так и с помощью решения оптимизационной задачи. Также рассматриваются модификации эвристических алгоритмов, позволяющие получить точное решение задачи о возможности распределения в сети возникшей в данный момент совокупности элементарных потоков. Производится сравнение результатов работы одно- и двухуровневых алгоритмов на сетях с кластерной и стохастической топологией по таким параметрам, как: время работы имитационного моделирования и суммарное количество проведенного потока.

Ключевые слова: *сеть, последовательные алгоритмы, потоки в сетях, суперприоритет, двухуровневая маршрутизация.*

Введение

Данная работа является развитием работ [1–4] и посвящена проблеме динамической маршрутизации потоков данных в телекоммуникационных сетях, которая также может быть рассмотрена в рамках многопродуктовой задачи теории потоков в сетях [5]. Один из аспектов многопродуктовой проблемы — задача допустимости — вполне соответствует задаче распределения каналов связи в сети в каждый момент времени, т. е. при учете всей совокупности заявок.

Моделирование динамической маршрутизации потоков предполагает последовательное по времени поступление на полюса заявок на передачу некоторого объема данных. Такие заявки на передачу потоков данных могут быть формализованы в виде потока продуктов между абонентами. В работах [1, 2] были предложены алгоритмы заполнения сети последовательно поступающими во времени потоками продуктов. В основе всех этих алгоритмов лежит принцип экономии дефицитного ресурса сети, такого как пропускные способности дуг сети или про-

пускные способности минимальных разрезов между всеми парами абонентских узлов. Маршрут для очередного потока ищется как путь минимальной стоимости, где стоимость дуг сети может быть найдена в зависимости от:

- 1) остаточной пропускной способности дуг, дуги с меньшей остаточной пропускной способностью получают больший вес (*дуговой, субоптимальный дуговой алгоритмы*); (1.1)
- 2) величины минимального разреза, в который входит дуга, наибольший вес получает дуга, входящая в наименьший из возможных минимальных разрезов, и, таким образом, пути не прокладывались по «узким местам» сети (*минимально-разрезный, субоптимальный минимально-разрезный, гибридный алгоритмы*); (1.2)
- 3) всем ребрам назначаются одинаковые веса (единичные), и поиск пути наименьшей стоимости становится поиском кратчайшего пути (*простой алгоритм*). (1.3)

Наилучшие результаты, в частности по величине проложенного через сеть потока, показали субоптимальный минимально-разрезный и гибридный алгоритмы, однако их особенность заключается в большой вычислительной сложности, например, время работы минимально-разрезного алгоритма равно $(|V| |E|^2 P)$, где V — число узлов в сети,

* Исследование выполнено при поддержке ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» (госконтракт № 11.519.11.4012) и РФФИ (проект № 12–07–00415-а).

E — число дуг графа, описывающего телекоммуникационную сеть, P — число пар корреспондирующих вершин (полюсов сети). В качестве возможного варианта уменьшения времени работы алгоритмов заполнения сети последовательными заявками, была предложена декомпозиция задачи динамической маршрутизации и введение двухуровневой маршрутизации, предложенной в работах [3, 4]. Напомним, что двухуровневая маршрутизация предполагает организацию локальной маршрутизации в сети посредством суперпиров — выделенных узлов с дополнительной функциональностью, образующих связную подсеть. Потоки продуктов проходят от узла-источника через из суперпиров, далее по подсети суперпиров с окончанием на узле-адресате.

Второй особенностью решаемой задачи является то, что поступающие в сеть заявки не могут быть точно предсказаны заранее и, следовательно, решение многопродуктовой задачи, как задачи линейного программирования, становится затруднительным. Тем не менее, для имитационных экспериментов на основе существующих эвристических алгоритмов были разработаны составные алгоритмы, решающие задачу допустимости многопродуктового потока в заданный момент времени.

В настоящей работе будут приведены следующие результаты:

1. Описание составного алгоритма, в котором возможность удовлетворения текущего множества заявок в сети решалась как задача допустимости многопродуктового потока методом линейного программирования. Этот алгоритм позволяет получить точный ответ на вопрос, возможно ли удовлетворение всей совокупности заявок, с учетом времен жизни каналов связи, вне зависимости от того, какой именно последовательный алгоритм был использован.
2. Сравнительный анализ времени работы имитационной модели для одно- и двухуровневых модификаций последовательных алгоритмов заполнения сети потоками продуктов, а также сравнительный анализ суммарных потоков, получаемых этими алгоритмами. Дополнительно дано описание алгоритма выделения подсети суперпиров, как задачи линейного программирования с булевыми переменными.

Во второй части работы будет приведена точная постановка задачи и описание используемых алгоритмов. Третья часть будет посвящена описанию численных экспериментов, четвертая — их результатам. Выводы по проведенным экспериментам будут даны в пятом, заключительном, разделе.

1. Выделение подсети узлов с дополнительной функциональностью. Построение алгоритмов двухуровневой маршрутизации

1.1. Постановка задачи

Определим задачу нахождения многопродуктового потока в сети можно как задачу допустимости. Напомним, что в основе математической модели телекоммуникационной сети лежит связный граф $G = (V, E)$, где каждому ребру $e_{ij} \in E$, $\{i, j\} \in V$ графа G поставлено в соответствие неотрицательное число $c_{ij} \geq 0$ — пропускная способность ребра.

$$\begin{aligned} & \text{minimize } \sum_{l=1}^n \sum_{e \in E} f_e^l \\ & \sum_{l=1}^n f_{ij}^l + \sum_{l=1}^n f_{ji}^l \leq c_{ij} \quad \forall (i, j) \in E \\ & f_{ij}^l \geq 0 \quad \forall (i, j) \in E, \quad \forall l \in [1 \dots n] \\ & \sum_{(j, i) \in E} f_{ji}^l - \sum_{(i, k) \in E} f_{ik}^l = 0 \quad \forall i \in V, \\ & i \notin \{s_1, t_1, \dots, s_n, t_n\} \quad \forall l \in [1 \dots n] \\ & \sum_{(s_l, j) \in E} f_{sj}^l = d_l \quad \sum_{(j, s_l) \in E} f_{js_l}^l = 0 \quad \forall l \in [1 \dots n] \\ & \sum_{(j, t_l) \in E} f_{jt_l}^l = d_l \quad \sum_{(t_l, j) \in E} f_{t_lj}^l = 0 \quad \forall l \in [1 \dots n] \end{aligned} \quad (1)$$

Здесь f_{ij}^l это поток l -го продукта по дуге e_{ij} . Ограничение неравенства определяет то, что сумма потоков всех продуктов по любому ребру не превышает ее пропускной способности. Первое ограничение равенства отвечает за ненакопление потока l -го продукта во всех промежуточных узлах, кроме своих полюсов. Второе и третье ограничение равенства определяют интенсивности потоков каждого продукта, т. е. количества заявок на каждой паре корреспондирующих вершин в данный момент времени. На основе такой формулировки задачи допустимости многопродуктового потока строится составной алгоритм последовательного заполнения сети потоками.

1.2. Составной алгоритм решения динамической многопродуктовой задачи

Идея составного алгоритма заполнения сети потоками связи, поступающими в сеть последовательно во времени, состоит в том, чтобы использовать

линейное программирование (ЛП) как точный метод решения задачи допустимости (1) в тех случаях, когда последовательные эвристические алгоритмы не могут удовлетворить очередную заявку. Если с помощью решения задачи ЛП удалось удовлетворить заявку, то идет перестроение путей уже существующих потоков, если нет — то заявка сбрасывается и фиксируется «отказ» составного алгоритма. Основные этапы работы составной модификации существующих последовательных алгоритмов (1.1)–(1.3) следующие:

1. Действие алгоритма начинается тогда, когда очередная (k -я) заявка не может быть удовлетворена алгоритмами (1.1)–(1.3).
2. Строится вектор действительных (с неистекшим временем жизни) заявок.
3. Формируется вектор ограничений равенства для задачи ЛП, определяющий величины продуктов между парами корреспондирующих узлов.
4. Производится решение задачи (1) методом линейного программирования.
5. Если задача не имеет решения при полученных ограничениях, то заявка сбрасывается и считается неудовлетворенной, переходим к шагу 7.
6. Если решение существует, то перестраиваем пути существующих потоков, которые не могут быть «вписаны» в полученное решение, плюс путь для новой заявки, и назначаем им те оставшиеся времена жизни, которые были у «старых» путей плюс время жизни новой удовлетворенной заявки.
7. Берем очередную ($k + 1$)-ю заявку. Если она не может быть удовлетворена с помощью алгоритмов последовательного заполнения, переходим к шагу 1.

Схематично составной алгоритм можно представить в следующем виде (см. рис. 1).

Важным элементом работы составного алгоритма является переопределение простых путей потоков после решения задачи допустимости многопродуктового потока после решения задачи линейного программирования. В приведенной формулировке задачи допустимости (1) решением будет следующая структура данных: $\| (v_i, v_j) \| \| f_{ij}^l \|$, где v_i и v_j — начальная и конечная вершины дуги графа $G = (V, E)$, l — номер продукта, эквивалентный номеру пары полюсов, между которыми проходит данный продукт, f_{ij}^l — величина потока l -го продукта, протекающего по ребру (v_i, v_j) . Соответственно для каждого продукта можно выделить подмножество дуг, по которым протекает поток с суммарной интенсивностью равной количеству действительных заявок, удовлетворяемых на данной паре полюсов. Выделив это подмножество дуг, восстанавливаются все простые пути потоков, ведущих из источника в сток. По минимальной величине потока через ребро, определяется максимальная мощность потока, протекающего через каждый простой путь. Далее выбирается подмножество простых путей так, чтобы сумма потоков по этим простым путям была равна суммарной интенсивности потока между данной парой полюсов, т. е. пропорциональная числу заявок с неистекшим временем жизни между этими терминальными вершинами. Далее последовательно сравниваются пути, полученные с помощью алгоритма линейного программирования (ЛП) и пути, сформированные до поступления k -й заявки. Если при этом «старые» пути «вписываются» в ЛП-решение, то они остаются в сети без изменения их времен жизни. В противном случае «старые» пути заменяются на новые, и на них переносятся времена жизни

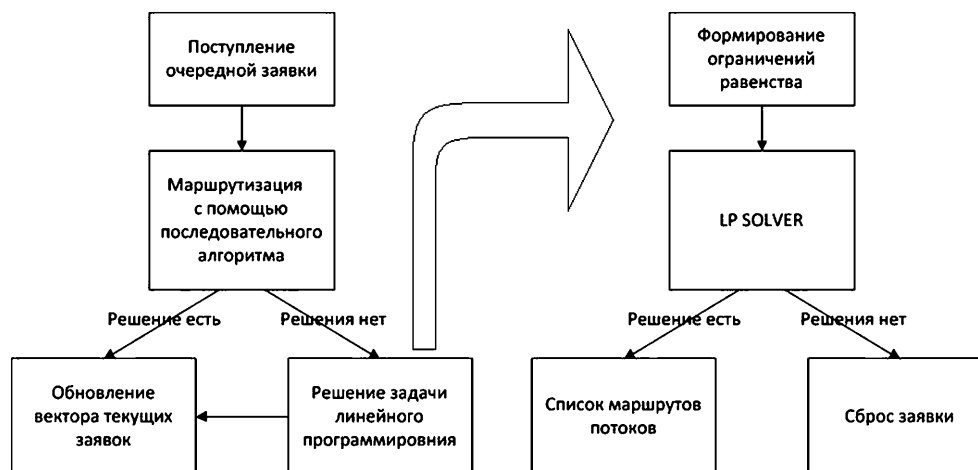


Рис. 1. Схема работы составного алгоритма

«старых» путей.

«Отказ» составного алгоритма на k -м шаге означает «абсолютную» невозможность удовлетворить k -ю заявку, вне зависимости от того, посредством какого именно эвристического последовательного алгоритма заполнялась сеть. Таким образом, характеристики составного алгоритма являются абсолютными или наилучшими, и не могут быть улучшены никакими эвристическими или иными последовательными и другими алгоритмами.

Далее перейдем к декомпозиции задачи заполнения сети потоками последовательно поступающих в сеть заявок, а именно к двухуровневой маршрутизации потоков в сетях.

1.3. Двухуровневая маршрутизация потоков

Как было сказано ранее, основным принципом двухуровневой маршрутизации потоков является прокладка потоков через несколько выделенных узлов-суперпиров Sp (*SuperPeers*), и для любой пары $\{s_i, t_i\}$ каждый простой поток между ними проходит через один и более суперпиров, см. рис. 2.

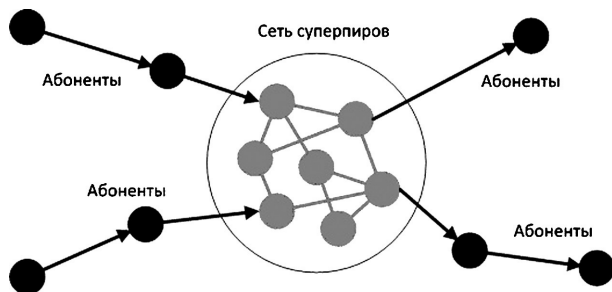


Рис. 2. Схема двухуровневой маршрутизации

Эта схема предполагает, что первым уровнем маршрутизации в сети будет прокладка пути от узла абонента до ближайшего суперпира и обратно. Второй уровень маршрутизации — прокладка канала данных в подсети суперпиров. Подобные схемы маршрутизации можно встретить в некоторых типах пиринговых сетей [8], VoIP сетях [9], пиринговых видеосетях [10, 11]; в работах [3, 4] приведены более подробные описания систем, использующих многоуровневую парадигму маршрутизации потоков данных.

Для выделения суперпиров была введена такая характеристика, как разгруженность узла — мера способности узла принять, обработать и передать дальше поток. В ходе предварительных работ было предложено несколько функций разгруженности [3, 4] из которых наилучшей оказалась следующая:

$$U(p) = \lambda_1 * \left(\frac{rescap(p)}{\max_p(rescap(p))} \right) + \lambda_2 * \left(\frac{d(p)}{\max_p(rescap(p))} \right),$$

где $rescap(p)$ — сумма остаточных пропускных способностей ребер, инцидентных вершине p , $d(p)$ — степень рассматриваемой вершины, обе эти величины нормированы на максимальные значения этих величин, встречающиеся в сети. λ_1 и λ_2 — нормирующие коэффициенты, определяющие решающий вклад степени или остаточной пропускной способности вершины. Было показано, что при использовании такой функции разгруженности удастся получить множество суперпиров с меньшим числом связанных компонент, нежели для остальных.

Для выделения подмножества суперпиров был использован метод агрегации данных [4–7], позволяющий каждому узлу в сети «самостоятельно» получать гистограмму зависимости числа узлов в сети от их разгруженности и определять порог U_{Limit} вхождения узла во множество суперпиров в зависимости от требуемого количества суперпиров в сети.

Найденное множество узлов сети-суперпиров в общем случае представляет собой несвязный граф, что является серьезным недостатком метода агрегации данных, и предполагает необходимость дополнительных усилий по преодолению этого недостатка.

Между тем задача нахождения подграфа суперпиров может быть поставлена и решена как задача выделения связного подграфа с числом вершин не больше заданной величины и весами вершин не меньше некоторого граничного значения. Оказывается, что все ограничения и целевая функция в такой задаче оптимизации линейны, следовательно, она может быть решена методом линейного программирования. Правда, такая формулировка задачи выделения связного подграфа суперпиров предполагает введение дополнительных булевых переменных¹.

Точная формулировка задачи приведена ниже.

$$\max \sum_{i,j \in V} E_{ij} * Cap_{ij},$$

$$\sum_{i \in V} X_i \leq k, \quad (2)$$

$$\forall (i, j) \in E \quad 0 \leq Y_{ij} \leq (k - 1) * X_j \quad (3)$$

$$i = v_r \quad \sum_{j \in V, (j,i) \in E} Y_{ji} = 0, \quad (4)$$

¹ Возможность сформулировать задачу выделения связного подграфа с линейной целевой функцией как задачу линейного программирования была в частной беседе высказана Волошиновым В. В.

$$i = v_r \quad \sum_{j \in V, (i,j) \in E} Y_{ij} = k - 1, \quad (5)$$

$$i \neq v_r \quad \sum_{j \in V, (j,i) \in E} Y_{ji} = \sum_{j \in V, (i,j) \in E} Y_{ij} + X_i, \quad (6)$$

$$\forall (i, j) \in E \quad 1 - E_{ij} \leq (1 - X_i) + (1 - X_j), \quad (7)$$

$$\forall (i, j) \in E \quad E_{ij} \leq X_i, \quad (8)$$

$$\forall (i, j) \in E \quad E_{ij} \leq X_j, \quad (9)$$

В ней используются следующие переменные:

1. E_{ij} — бинарная переменная, определяющая принадлежность дуги графа к искомому подграфу, i, j — индексы вершин графа.
2. Cap_{ij} — вектор пропускных способностей всех дуг графа.
3. X_i — бинарная переменная, определяющая принадлежность вершины к искомому подграфу.
4. Y_{ij} — вспомогательная переменная, имеющая смысл виртуального потока, и отвечающая за связность выделяемого подграфа.
5. v_r — затравочная вершина, с которой начинается построение подграфа, ею являлась вершина с наибольшей загруженностью.
6. k — требуемое число вершин подграфа.

В данном случае осуществляется максимизация суммы пропускных способностей дуг подграфа суперпиров. Первое неравенство (2) отвечает за ограничение по размеру выделяемого подграфа. Неравенства (3)–(6) отвечают за связность подграфа; неравенства (7)–(9) определяют те ребра, которые войдут в подграф суперпиров (доказательство корректности всех приведенных неравенств можно найти, например, в [12]).

После того как удалось выделить подсеть узлов с дополнительной функциональностью, необходимо определить схему работы двухуровневых модификаций алгоритмов последовательного заполнения сети потоками продуктов.

1.4. Выделение подсети суперпиров

После определения подсети суперпиров идет построение кластера клиентских узлов вокруг каждого суперпира таким образом, чтобы гарантированно существовал маршрут из любого клиентского узла к суперпиру, его обслуживающему. Приведем описание алгоритма такой кластеризации.

```

init  $Clust\_set \leftarrow \{Sp_1, Sp_2, \dots, Sp_k\}$  //инициализируем
      кластеры суперпирами
for  $i = 1 : (N - k)$  // для всех оставшихся узлов
       $Clust_j = find\_bcap(Cl_i, Clust\_set)$  // для  $i$ -го
      клиента находим кластер, с которым он соеди-
      нен дугой с наибольшей остаточной пропускной
      способностью.
    
```

$Clust_j \leftarrow Cl_i$ // включаем i -го клиента в най-
денный кластер

$Clust_set \leftarrow Clust_j$

end

Дальнейший алгоритм маршрутизации выглядит следующим образом:

- 1) при поступлении заявки на клиентский узел Cl_i прокладывается кратчайший маршрут к суперпиру Sp_1 кластера, к которому этот клиент принадлежит;
- 2) после чего по подсети суперпиров с помощью одного из алгоритмов последовательного заполнения сети находится маршрут $Sp_1 \rightarrow Sp_2$ к суперпиру кластера, в котором находится адресат;
- 3) уже внутри него рассчитывается путь $Sp_2 \rightarrow Cl_2$.

2. Описание организации численных экспериментов

Для анализа предложенных алгоритмов маршрутизации была использована имитационная модель, моделирующая телекоммуникационные сети и поток заявок, поступающий на них в каждый дискретный момент времени (более подробно она описана в [3, 4]). Заполнение сети потоками продуктов происходило в динамическом режиме, т. е. заявки имели конечное время жизни. Выполненные численные эксперименты основывались на результатах численного моделирования по заполнению сетей потоками последовательных заявок согласно описанному выше сценарию, также описанному в работах [1, 2].

В проведенных экспериментах рассматривались сети с кластерной [1] и стохастической топологиями [2]. В этих сетях количество вершин колебалось от 30 до 45, плотность заполнения сети ребрами колебалась от 10 % до 20 %. Под кластерной топологией сети мы понимаем связные между собой подграфы с большой плотностью ребер (4–5 кластеров узлов на сеть). Длина очереди заявок равнялась 45 000, время жизни заявок колебалось от 1 до 1000 единиц времени с математическим ожиданием в 300 единиц времени. В каждом классе топологий было взято по 60 сетей.

В данном исследовании представлены результаты следующих численных экспериментов:

1. Сравнение времени работы одноуровневых реализаций алгоритмов последовательного заполнения сети и их двухуровневых модификаций. Для экспериментов были выбраны простой (Simple), субоптимальный дуговой (EmpEdge) и субоптимальный минимально-разрезный алгоритмы

(EmpCut), чтобы охватить все эвристики назначения весов дугам в последовательных алгоритмах.

2. Сравнение суммарного количества потока, пропущенного через сеть за все время работы имитационной модели для одноуровневых и двухуровневых реализаций последовательных алгоритмов маршрутизации потоков продуктов.
3. Сравнение результатов, полученных посредством последовательных и составного алгоритма.

Несмотря на самоочевидный характер результатов в **первом эксперименте**, а именно очевидное ускорение работы алгоритмов последовательного заполнения на меньшем множестве узлов, представляет интерес зависимость ускорения маршрутизации от типа последовательного алгоритма (простой, дуговой или минимально-разрезный).

Во **втором эксперименте** интересна разница между суммарным потоком, проложенным одноуровневым и двухуровневыми алгоритмами. Чем она меньше, тем больше оснований пользоваться двухуровневыми реализациями последовательных алгоритмов.

Третий эксперимент позволяет оценить «абсолютную» погрешность последовательных алгоритмов маршрутизации, являющихся по сути эвристическими.

3. Результаты численных экспериментов

На рис. 3 показано ускорение работы многопродуктовой маршрутизации для сетей с кластерной и стохастической топологией при использовании двухуровневой маршрутизации, основанной на методе агрегации данных. В качестве входного параметра метода агрегации данных было выделение числа суперпиров, равного 30 % от общего числа узлов. В качестве одноуровневых алгоритмов были использованы простой (Simple), *субоптимальный минимально-разрезный* (EmpCut) и *субоптимальный дуговой* (EmpEdge).

Таким образом, введение двухуровневой маршрутизации позволяет получить, в среднем, ускорение в 1,05 раз для простого алгоритма, в 17,9 раз для субоптимального минимально-разрезного и 2,9 раз для субоптимального дугового алгоритмов одноуровневой маршрутизации на сетях с кластерной топологией. На сетях со стохастической топологией аналогичные результаты таковы: ускорение в 1,15 раз для простого, в 22,5 для субоптимального минимально-разрезного, в 4,1 раз для субоптимального дугового алгоритмов.

Аналогичные результаты по двухуровневой маршрутизации с использованием модифицированного ал-

горитма двухуровневой маршрутизации (нахождение подграфа суперпиров с помощью задачи линейного программирования) приведены на рис. 4.

В этом случае получается ускорение в 1,23 раз для простого алгоритма, в 9,1 раз для субоптимального минимально-разрезного и 2,48 раз для субоптимального дугового алгоритмов одноуровневой маршрутизации на сетях с кластерной топологией. На сетях со стохастической топологией аналогичные результаты равны: ускорение в 1,28 раз для простого, в 9,1 для субоптимального минимально-разрезного, в 2,45 раз для субоптимального дугового алгоритмов. Однако необходимо заметить то, что само по себе решение задачи линейного программирования по выделению связной подсети с максимальной суммой остаточных пропускных способностей ребер является вычислительно сложной, что затрудняет на практике широкое использование этого метода.

Далее приведем результаты, которые показывают, на сколько процентов суммарный поток, полученный последовательными алгоритмами и их двухуровневыми модификациями меньше суммарного потока, полученного их составными аналогами. Результаты для последовательных алгоритмов отмечены значком «○», результаты их двухуровневых реализаций — «+», результаты для двухуровневых алгоритмов, выделяющий подсеть суперпиров с помощью решения задачи линейного программирования (2)-(9) — «▽».

В данном случае необходимо отметить отсутствие качественного различия в результатах для различных эвристик нахождения весов дуг при построении путей минимальной стоимости. Например, в случае минимально-разрезной эвристики, среднее уменьшение суммарного потока относительно составного алгоритма для последовательного алгоритма составляет 0,62 % (дисперсия 1,55 % суммарного потока), для двухуровневого алгоритма маршрутизации с применением метода агрегации данных уменьшение составляет 8,67 % (дисперсия 4,46 %), для двухуровневого алгоритма с точным нахождением подграфа суперпиров уменьшение суммарного потока составляет 3,44 % (дисперсия 5 %) от результата составного алгоритма. Для сетей со стохастической топологией аналогичные результаты показаны на рис. 6.

Уменьшение суммарного потока относительно составного алгоритма в случае минимально-разрезной эвристики назначения весов дугам сети равно: 2 % ($\sigma = 2,95\%$) для простого алгоритма; 13 % ($\sigma = 4,11\%$) для двухуровневого алгоритма маршрутизации с привлечением метода агрегации данных; 7,34 % ($\sigma = 6,91\%$) для двухуровневого алгоритма с точным нахождением подграфа суперпиров. Таким

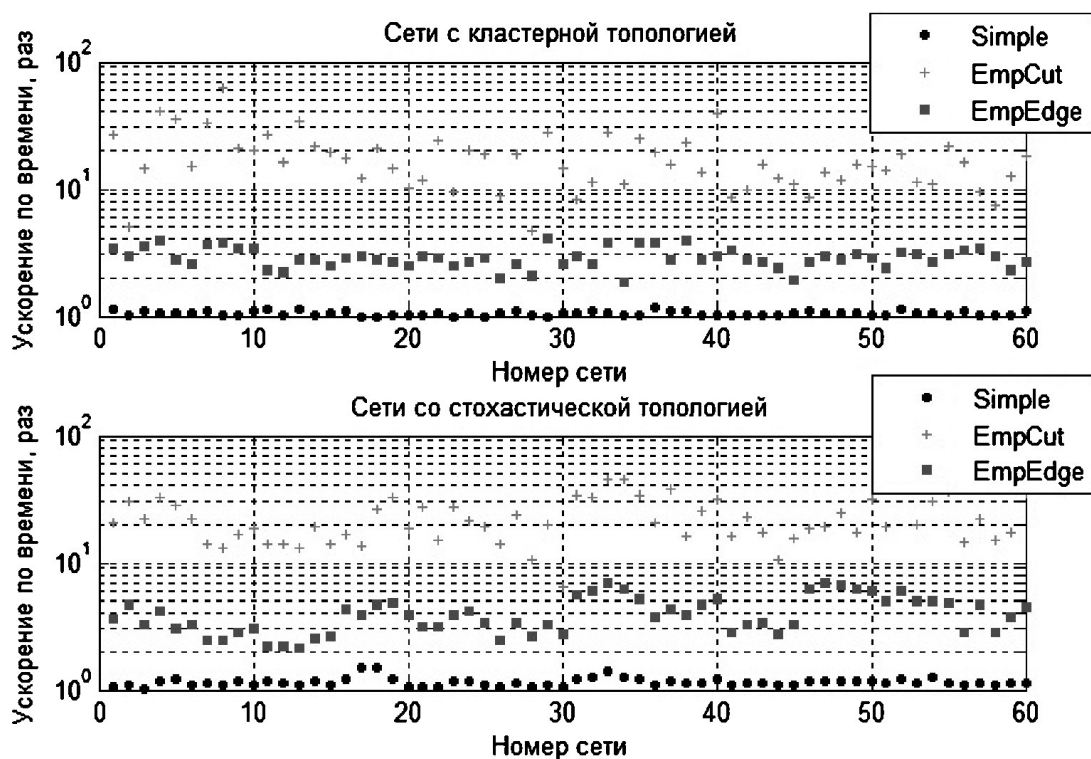


Рис. 3. Ускорение работы последовательных алгоритмов маршрутизации при введении второго уровня

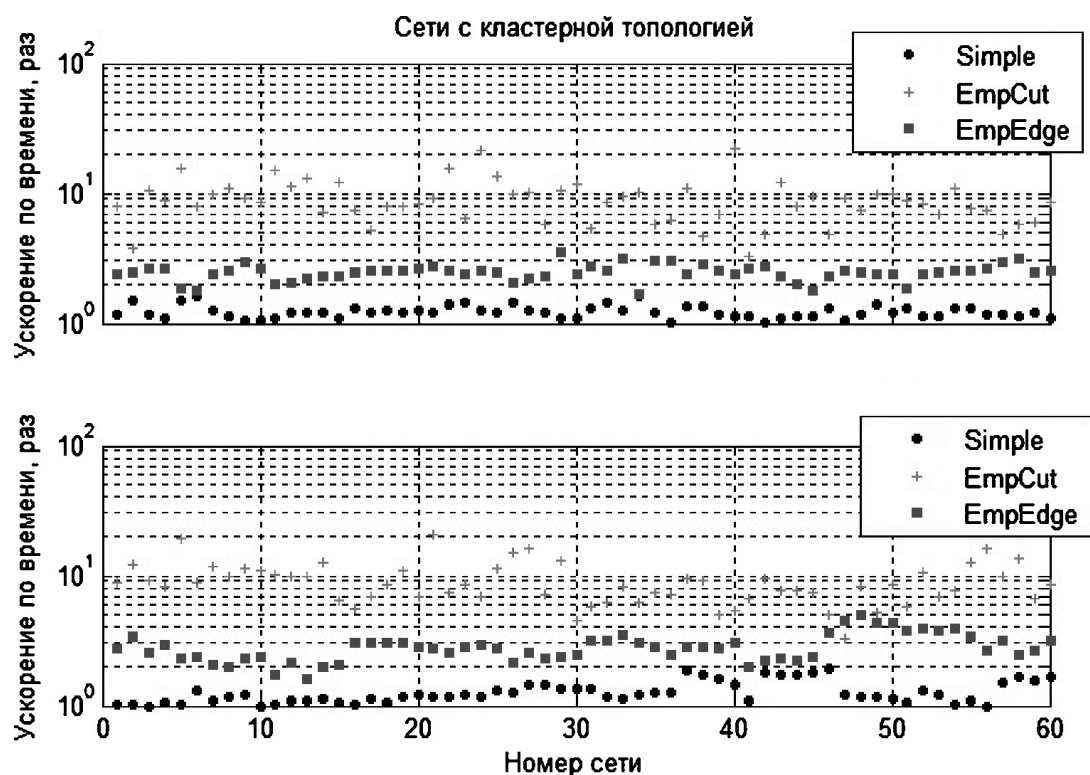


Рис. 4. Ускорение работы последовательных алгоритмов маршрутизации для модифицированного алгоритма двухуровневой маршрутизации

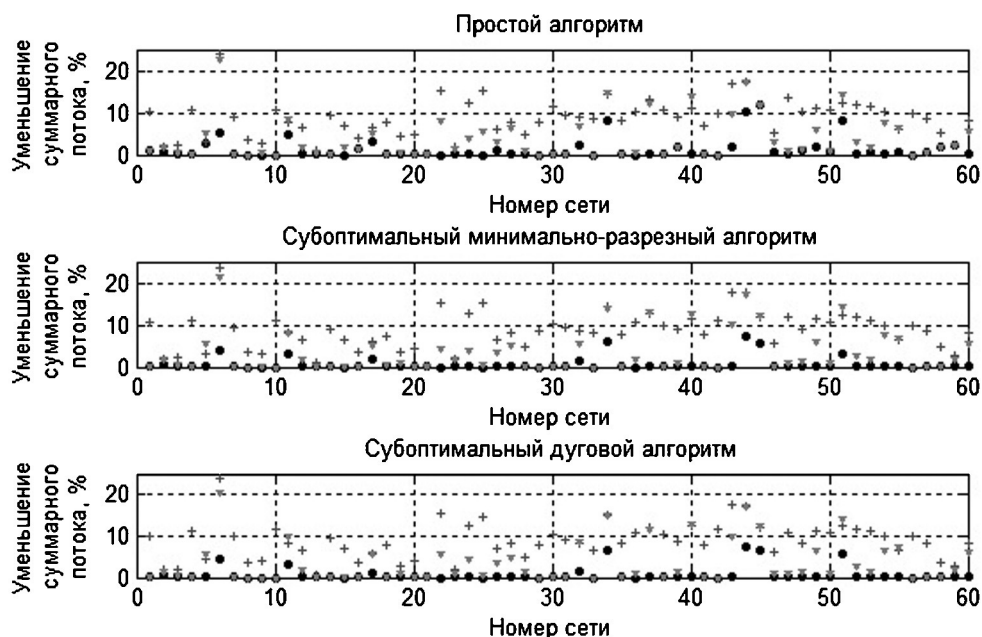


Рис. 5. Уменьшение суммарного потока для последовательных алгоритмов и их двухуровневых модификаций, относительно точного решения задачи многопродуктовой маршрутизации. Сети с кластерной топологией

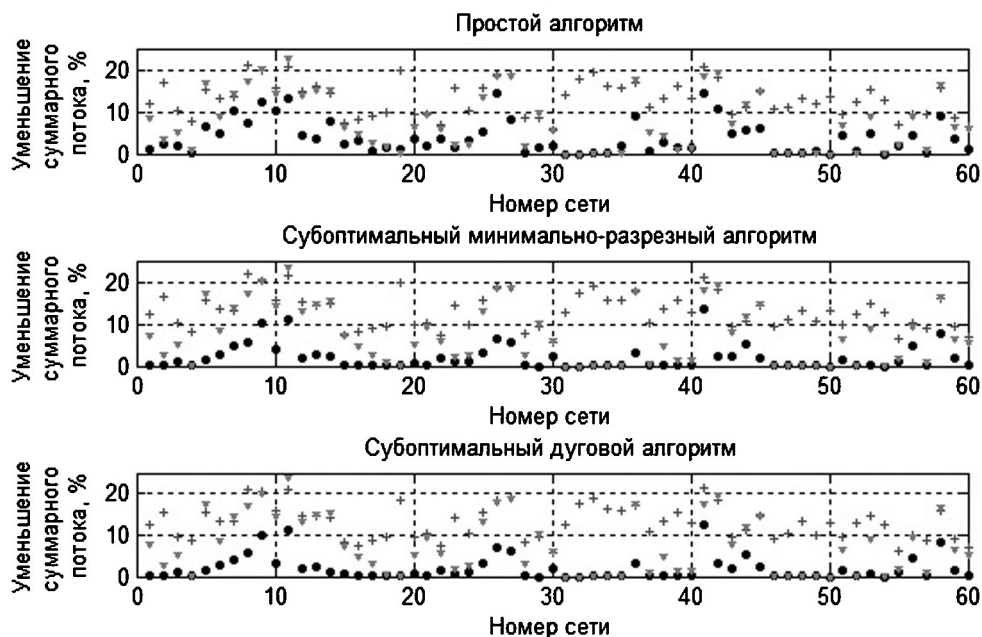


Рис. 6. Уменьшение суммарного потока для последовательных алгоритмов и их двухуровневых модификаций, относительно точного решения задачи многопродуктовой маршрутизации. Сети со стохастической топологией

образом, результат, полученный с помощью простой двухуровневой маршрутизации на сетях обоих типов дает уменьшение суммарного потока, в среднем, на 10–12 % по сравнению с точным решением задачи динамической многопродуктовой маршрутизации и эвристическими алгоритмами последователь-

ного заполнения сети. Если при выделении подсети суперипров ориентироваться на максимизацию суммы остаточных пропускных способностей, то тогда двухуровневая маршрутизация дает уменьшение суммарного потока, в среднем на 5 % от точного решения.

Выводы

В данной работе были представлены:

- 1) способ оценки эффективности работы последовательных алгоритмов заполнения телекоммуникационной сети потоками продуктов, посредством введения составного алгоритма;
- 2) механизм организации двухуровневой маршрутизации потоков продуктов заявок, последовательно поступающих во времени, на основе выделения узлов с дополнительной функциональностью для организации локальной маршрутизации.

Как составной алгоритм, так и алгоритмы двухуровневой маршрутизации были представлены, как модификации соответствующих одноуровневых последовательных алгоритмов, основанных на принципе экономии дефицитного ресурса: простом, субоптимальном минимально-разрезном и субоптимальном дуговом алгоритмах.

Численные эксперименты показали, что существующие эвристические алгоритмы последовательного заполнения сетей потоками продуктов дают суммарный поток, близкий по величине к суммарному потоку, получаемому с помощью точного решения динамической многопродуктовой задачи (уменьшение суммарного потока, в среднем, составляет 1,5 %).

Результаты численных экспериментов на сетях с кластерной и стохастической топологиями показали ускорение работы указанных алгоритмов маршрутизации, при замене их двухуровневыми модификациями:

1. На сетях с кластерной топологией, в среднем, для простого алгоритма ускорение составило 1,05 раз, для субоптимального минимально-разрезного алгоритма в 17,9 раз, для субоптимального дугового, в среднем, в 2,9 раз.
2. На сетях со стохастической топологией, в среднем, для простого алгоритма ускорение составило 1,16 раз, для субоптимального минимально-разрезного алгоритма в 22,5 раз, для субоптимального дугового, в среднем, в 4 раза.

Тем не менее, количество удовлетворенных заявок для двухуровневых модификаций алгоритмов составляла, в среднем, 91,34 % от числа заявок, удовлетворенных составными алгоритмами для сетей с кластерной топологией, и 87,0 % для сетей со стохастической топологией. Тем не менее, данные результаты улучшаются в два раза, если при выделении подграфа суперпиров использовать требование максимизации суммы остаточных пропускных способностей дуг этого подграфа.

Таким образом было получено, что в данной модели функционирования телекоммуникационной сети введение двухуровневой маршрутизации может быть эффективно ввиду увеличения скорости работы алгоритмов и возможности такого построения подсети суперпиров, что увеличение числа неудовлетворенных заявок будет минимальным.

Литература

1. *Гринберг Я. Р., Курочкин И. И.* Математическое моделирование динамического последовательного заполнения сетей потоками связи // Проблемы вычислений в распределенной среде: Труды ИСА РАН / Под ред. С. В. Емельянова, А. П. Афанасьева Т. 46 М.: Красанд/URSS, 2009. С. 233–258.
2. *Гринберг Я. Р., Курочкин И. И.* Анализ результатов численного эксперимента по последовательному заполнению сетей со стохастической топологией // Проблемы вычислений в распределенной среде: распределенные приложения, коммуникационные системы, математические модели и оптимизация: Сборник трудов ИСА РАН / Под ред. А. П. Афанасьева Т. 25 М.: Ком-Книга/URSS, 2006. С. 99–128.
3. *Гринберг Я. Р., Курочкин И. И., Корх А. В.* Алгоритм кластеризации сетей передачи данных // Информационные технологии и вычислительные системы. Т. 3. М.: Ленанд/URSS, 2012. С. 18–30.
4. *Гринберг Я. Р., Курочкин И. И., Корх А. В., Алыгулиев Р. М., Гашимов М. А.* Применение метода агрегации данных в задаче нахождения узлов с дополнительной функциональностью // Информационные технологии и вычислительные системы. Т. 2. М.: Ленанд/URSS, 2013. С. 27–39.
5. *Ху Т.* Целочисленное программирование и потоки в сетях. М.: Мир, 1973.
6. *Sacha J., Dowling J., Canningham R., Meier R.* Using aggregation for adaptive superpeer discovery on the gradient topology / In Proceedings of 2nd IEEE International Workshop on Self-Managed Networks, Systems and Services. 2006. P. 77–90.
7. *Jelasy M., Montresor A., Babaoglu O.* Gossip-based aggregation in large dynamic networks // ACM Transactions on Computer Systems. August 2005. V. 23. P. 219–252.
8. *Li Y., Huang X., Ma F., Zou F.* Building eclient super-peer overlay network for DHT systems // In Proceedings of the 4th International Conference on Grid and Cooperative Computing. 2005. P. 787–798.

9. Guha S., Daswani N., Jain R. An Experimental Study of the Skype Peer-to-Peer VoIP System / In Proceedings of the 5th International Workshop on Peer-to-Peer Systems. 2007. P. 1–6.
10. Xinyan Zhang, Jiangchuan Liuy, Bo Liz, and Tak-Shing Peter Yum. CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming / In Proceedings of IEEE INFOCOM, March 2005.
11. Abbasi U., Ahmed T. SWOR: an Architecture for P2P Scalable Video Streaming using Small World Overlay // IEEE CCNC 2010 proceedings.
12. Алексеева Е. В. Построение математических моделей целочисленного линейного программирования. Примеры и задачи: Учеб. пособие. Новосибирск: Изд-во Новосиб. гос. ун-та, 2012. 131 с.

Афанасьев Александр Петрович. Заведующий ИППИ РАН. Д. ф.-м. н., профессор. Окончил МФТИ в 1968 г. Количество печатных работ: более 120 (в том числе 3 монографии и 2 учебника). Область научных интересов: алгоритмы распределенных вычислений, оптимизационные задачи. E-mail: apa@isa.ru

Гринберг Яков Рувимович. Заведующий лабораторией ИППИ РАН, к. т. н. Окончил МФТИ в 1968 г. Количество печатных работ: 14. Область научных интересов: математическое моделирование потоков в сетях, алгоритмы канальной маршрутизации, задачи многопродуктовой маршрутизации. E-mail: greenjak@isa.ru

Курочкин Илья Ильич. С. н. с. ИППИ РАН, к. т. н. Окончил МГТУ им. Баумана в 2003 г. Количество печатных работ: 19. Область научных интересов: имитационное математическое моделирование телекоммуникационных сетей, распределенные и параллельные вычисления, анализ текстур на двумерных изображениях. E-mail: qurochkin@gmail.com

Корх Артем Владимирович. Инженер. Окончил МФТИ в 2010 г. Количество печатных работ: 2. Область интересов: моделирование телекоммуникационных сетей, сегментация растровых и спектральных изображений, алгоритмы сжатия видеопотоков. E-mail: artkorkh@gmail.com