Основную часть времени, как видно из табл. 3, занимают операция копирования данных из памяти CPU к видеокарте и обратная запись результата. Время непосредственной обработки занимает доли процента. Для ускорения обработки необходимо минимизировать объем вывода. На уменьшение коэффициента ускорения существенно влияют: время ожидания в коммуникациях, ограниченная пропускная способность, неэффективные алгоритмы, значительные задержки при запуске большого количества процессов (планирование).

Безусловно, для решения на GPU необходимо тщательно выбирать задачи. Основные приложения для GPU включают моделирование в физике, обработку и анализ изображений, финансовые расчеты, динамику газов и жидкостей, криптографию, обработку звука (методы сжатия), анализ данных (data mining), моделирование свертываемости белка, квантовую химию, вычислительную математику.

В последнее время nVidia выпустила CUDA Toolkit 3.0, поддерживающий OpenCL. В какой-то мере OpenCL можно считать альтернативой CUDA. OpenCL – это фреймворк, позволяющий разрабатывать параллельные программы для вычислений на GPU и CPU, включающий язык программирования (расширение стандарта C99) и API. Это реализация технологии GPGPU, обеспечивающей параллелизм как на уровне команд, так и на уровне данных.

В настоящее время графические процессоры трансформировались из устройств с фиксированным набором функций для обработки трехмерной графики в вычислительную платформу для общих вычислений (GPGPU). По сути, теперь каждый, кому необходимо осуществлять быстрые вычисления, может иметь недорогой суперкомпьютер на рабочем столе. Компания nVidia уже интегрировала язык C++ в свою технологию CUDA, что расширяет круг пользователей GPU. Тем не менее многое еще необходимо делать вручную, например создавать потоки и планировать доступ к памяти. Кроме того, алгоритмы решаемых задач для достижения существенного ускорения должны хорошо распараллеливаться на сотни потоков, содержать минимальное количество сложных для обработки операций (деление, возведение в отрицательную степень и т. д.), иметь незначительное количество ветвлений.

Однако при сравнимых трудозатратах на подготовку программ для CPU и CUDA последние будут работать значительно быстрее. Существенное ускорение вычислений ожидается при использовании квантовых компьютеров.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Валях Е. Параллельно-последовательные вычисления. М., 1985.

2. Буза М. К., Кондратьева О. М. Проектирование программ для многоядерных процессоров : учеб.-метод. пособие. Минск, 2013.

3. Буза М. К. Параллельная обработка данных в модульных архитектурах // Вестн. компьютерных и информационных технологий. 2007. № 6. С. 51–56.

4. Архитектура CUDA следующего поколения, кодовое название Fermi. Сердце суперкомпьютера в теле GPU // Сайт компании NVIDIA [Электронный ресурс]. 2013. Режим доступа: http://www.nvidia.ru/object/fermi_architecture_ru.html (дата обращения: 10.05.2014).

5. Aaftab Munshi. OpenCL. Programming Guide. Addison, 2011.

6. Пахомов С. Революция в мире графических процессоров // КомпьютерПресс [Электронный ресурс]. 2006. Режим доступа: http://www.compress.ru/Article.aspx?d=16963 (дата обращения: 20.10.2013).

7. Luebke D., Humhreys G. How GPUs Work // IEEE Computer Society [Electronic resource]. 2007. URL: http://www.research.nvidia.com/sites/default/files/publication/04085637.pdf/ (date of access: 20.10.2013).

Поступила в редакцию 12.11.2014.

Михаил Константинович Буза – доктор технических наук, профессор, заведующий кафедрой многопроцессорных систем и сетей факультета прикладной математики и информатики БГУ.

УДК 004.722.2+004.021

Н. И. ЛИСТОПАД, Ю. И. ВОРОТНИЦКИЙ, А. А. ХАЙДЕР (ИРАК)

МАРШРУТИЗАЦИЯ В МУЛЬТИСЕРВИСНЫХ СЕТЯХ ТЕЛЕКОММУНИКАЦИЙ НА ОСНОВЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА ДЕЙКСТРЫ

Рассматривается проблема поиска оптимальных маршрутов на графе мультисервисной телекоммуникационной сети. Для данных сетей, кроме полосы пропускания, должны приниматься во внимание такие параметры качества обслуживания (*QoS*), как потери пакетов, задержка пакетов, вариация времени задержки (джиттер). Задачу маршрутизации в мультисервисных сетях предлагается решать на основе критериев, учитывающих перечисленные параметры, согласно требованиям конкретных приложений. Эта задача сформулирована как многокритериальная задача поиска маршрута с минимальной стоимостью, причем поиск выполняется только на подмножестве осуществимых путей, удовлетворяющих ограничениям на параметры качества сервиса. Предложена модификация алгоритма Дейкстры, которая позволяет осуществлять многокритериальный поиск оптимального маршрута с учетом ограничений на каждый критерий в отдельности, а также в случае, когда стоимость маршрута неаддитивна. Приведены примеры расчетов, показывающие эффективность предложенного подхода в рамках сервис-ориентированной архитектуры.

Ключевые слова: маршрутизация; мультисервисная сеть; качество обслуживания; оптимальный маршрут; алгоритм Дейкстры.

The problem of optimal routes search for a multiservice telecommunication network graph is considered. For those networks, unless bandwidth, quality of service (*QoS*) parameters such as error rates, transmission delay, jitter should be taken into account. It is proposed to solve the problem of routing in multiservice networks, using criteria that take into account the *QoS* parameters, as well as the requirements of specific applications. This problem is formulated as a multi-objective problem of searching for the route with the lowest cost (the shortests path), while the search is performed only on a feasible paths subset satisfying the constraints on the *QoS* parameters. In this case, the path cost is defined as the four criteria convolution to ensure the minimization of error rates, transmission delay and jitter as well as maximize bandwidth. A modification of Dijkstra's algorithm, which enables multi-criteria search for an optimal route taking into account the constraints on each criterion separately, when the cost of the route is not additive, is proposed. Examples of calculations showing the effectiveness of the proposed approach within a service-oriented architecture.

Key words: routing; multi-service network; quality of service; the optimal route; Dijkstra's algorithm.

Мультисервисные сети предоставляют определенный комплекс услуг: широкополосный доступ в интернет, IP-телефония, цифровое телевидение, видеоконференцсвязь и др. Для данных сетей, кроме полосы пропускания, важны такие параметры, как потери пакетов, задержка пакетов, вариация времени задержки (джиттер). Задача маршрутизации в мультисервисных сетях должна решаться на основе критериев, учитывающих перечисленные параметры, и согласно требованиям конкретных приложений. В целом обеспечение заданного качества обслуживания (*QoS*) в этих сетях является сложной задачей многокритериальной оптимизации, для решения которой необходимы новые методы и подходы [1–3].

Постановка задачи

Рассмотрим процесс передачи данных между узлами в телекоммуникационной сети, топология которой описывается в виде графа $G(V, E)$, где V – множество узлов сети; E – множество дуг e_{ij}, соединяющих узлы $i, j \in V$, телекоммуникационной сети (рисунок) [4–5].

Обозначим как p путь, соединяющий узел источника s и узел получателя t. В качестве *QoS*-параметров каждого канала связи (ребра графа) будем рассматривать полосу пропускания Y_e, задержку D_e, вариации задержки J_e и вероятность потери пакетов Z_e. Для каждого из возможных путей из s в t будут справедливы следующие соотношения:

$$Y_{s,t} = \max_{e \in p}\{Y_e\}; \quad D_{s,t} = \sum_{e \in p} D_e; \quad J_{s,t} = \sum_{e \in p} J_e; \quad Z_{s,t} = 1 - \prod_{e \in p}(1 - Z_e).$$

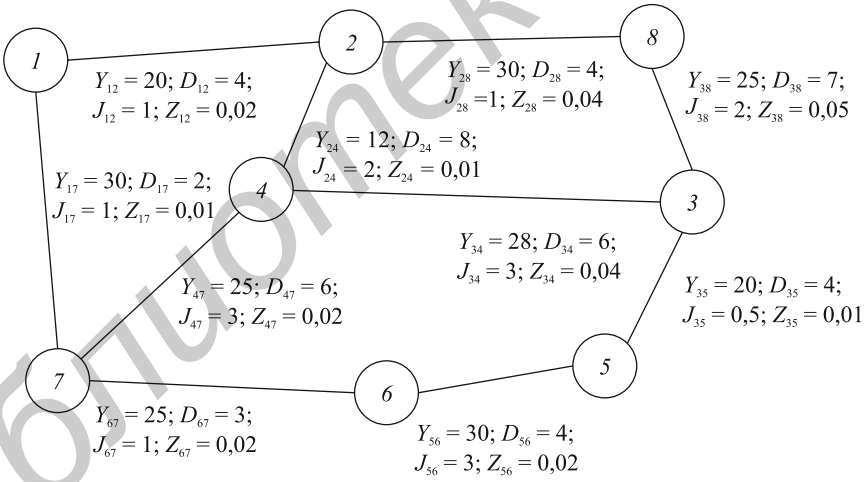

Схема сети: *1–8* – узлы сети

В целях упрощения дальнейшего анализа полагаем, что параметры качества обслуживания являются постоянными и не меняются в ходе эксплуатации сети, т. е. не зависят от ее загрузки. Данное предположение является корректным, если в процессе анализа используются среднестатистические величины.

Для удобства дальнейшего анализа вместо вероятности потери пакетов Z_e рассматривают такой параметр, как логарифм вероятности прохождения пакетов $X_e = \ln(1 - Z_e)$ [6]. Такой подход удобен тем, что значение данного параметра на всем пути маршрутизации вычисляется суммированием:

$$X_{s,t} = \sum_{e \in p} X_e.$$

Если на всем пути p для каждого из *QoS*-параметров будут выполнены ограничения вида

$$Y_{s,t} \geq Y^{\min};\ \ X_{s,t} \geq X^{\max};\ \ D_{s,t} \leq D^{\max};\ \ J_{s,t} \leq J^{\max}, \tag{1}$$

то такой путь называют *QoS*-осуществимым (feasible) путем.

Учет нескольких *QoS*-параметров и различных требований приложений к значениям этих параметров существенно усложняет задачу маршрутизации. Для ее решения более эффективным представляется подход, базирующийся на сервис-ориентированной архитектуре (SOA) [6]. Следуя методике, изложенной в [6], SOA можно представить в виде двух уровней.

1. Уровень резервирования необходимых телекоммуникационных ресурсов для обеспечения заданного качества обслуживания.

2. Уровень предоставления соответствующих телекоммуникационных услуг для обеспечения требуемого качества обслуживания.

Задача резервирования ресурсов может быть сформулирована следующим образом: телекоммуникационные ресурсы необходимо зарезервировать так, чтобы пропускные способности каналов связи обеспечивали маршрутизацию требуемого объема трафика при соблюдении требований заданного качества обслуживания (*QoS*) в безаварийной и во всех аварийных ситуациях. Другими словами, необходимо выбрать такой путь (пути), чтобы было обеспечено выполнение основных *QoS*-требований вида (1) [5].

Пусть $P_{fes}(s,t)$ есть множество *QoS*-осуществимых путей из s в t. Тогда проблема *QoS*-маршрутизации может быть сформулирована как модель оптимальной маршрутизации, рассматриваемая на множестве *QoS*-осуществимых путей $P_{fes}(s,t)$ [6, 7]:

$$\max_{P_{fes}} Y_{s,t} \Leftrightarrow \min_{P_{fes}} -Y_{s,t};\ \max_{P_{fes}} X_{s,t} \Leftrightarrow \min_{P_{fes}} X_{s,t};\ \min_{P_{fes}} D_{s,t};\ \min_{P_{fes}} J_{s,t}; \tag{2}$$

$$Y_{s,t} - Y^{\min} \geq 0;\ X_{s,t} - X^{\min} \geq 0;\ D^{\max} - D_{s,t} \geq 0;\ J^{\max} - J_{s,t} \geq 0; \tag{3}$$

$$Y_{s,t} = \max_{e \in p}\{Y_e\};\ D_{s,t} = \sum_{e \in p} D_e;\ J_{s,t} = \sum_{e \in p} J_e;\ X_{s,t} = \sum_{e \in p} X_e. \tag{4}$$

Построение вычислительного алгоритма

Можно выделить следующие особенности модели (2–4), которые не позволяют непосредственно применить для ее решения хорошо известные методы нахождения пути наименьшей стоимости.

1. Критерий оптимальности (2), включающий четыре *QoS*-параметра, является векторным. Его можно свернуть, используя, например, аддитивную свертку, и решать задачу минимизации на графе скалярной функции $R_{s,t}$

$$\min_p R_{s,t} = \min_p \left(-w_Y \frac{Y_{s,t}}{Y^{\min}} + w_D \frac{D_{s,t}}{D^{\max}} + w_J \frac{J_{s,t}}{J^{\max}} + w_X \frac{X_{s,t}}{X^{\min}} \right). \tag{5}$$

Здесь w_Y, w_D, w_J, w_X – весовые коэффициенты, определяющие значимость соответствующих параметров *QoS*. Знак «+» перед последним слагаемым объясняется тем, что максимизируемая величина $X_{s,t}$ при нормировке делится на отрицательную константу $X^{\min}$. Тем не менее в рассматриваемом случае в силу неаддитивности полосы пропускания $Y_{s,t} = \max_{e \in p}\{Y_e\}$ описать каждое ребро графа функцией стоимости R_e такой, что

$$R_{s,t} = \sum_{e \in p} R_e,$$

оказывается невозможным.

2. Поиск решения должен осуществляться только на подмножестве *QoS*-осуществимых путей, удовлетворяющих условиям (3).

Для решения задачи (4), (5) с учетом ограничений (3) предлагается использовать эвристический алгоритм, представляющий собой модификацию алгоритма Дейкстры [8]. Модификация заключается в отбрасывании в процессе поиска тех путей, на которых не выполняются ограничения (3), и новом способе описания и вычисления меток узлов. В исходном алгоритме Дейкстры метки каждого j-го узла,

которого можно достичь из узла *s* через соседний узел *i*, имеют вид $\left[R_{s,j}, i\right]$, где величина стоимости $R_{s,j}$, соответствующая данному пути, аддитивна и вычисляется по формуле $R_{s,j} = R_{s,i} + R_{ij}$, где R_{ij} – стоимость ребра e_{ij}; величина $R_{s,i}$ берется из метки *i*-го узла. Вместо этого введем метку, имеющую 6 компонент: $\left[R_{s,j}, Y_{s,j}, D_{s,j}, J_{s,j}, X_{s,j}, i\right]$. Новую метку при переходе из узла *i* в узел *j* будем вычислять следующим образом:

$$D_{s,j} = D_{s,i} + D_{ij}; \tag{6}$$

$$J_{s,j} = J_{s,i} + J_{ij}; \tag{7}$$

$$X_{s,j} = X_{s,i} + X_{ij}; \tag{8}$$

$$Y_{s,j} = \min\left\{Y_{s,i}, Y_{ij}\right\}; \tag{9}$$

$$R_{sj} = \begin{cases} r, \text{ если для } D_{sj}, J_{sj}, X_{sj}, Y_{sj} \text{ выполняются условия (3)}, \\ \infty, \text{ если для } D_{sj}, J_{sj}, X_{sj}, Y_{sj} \text{ не выполняется хотя бы} \\ \text{одно из условий (3)}, \end{cases} \tag{10}$$

где

$$r = -w_Y \frac{Y_{s,j}}{Y^{\min}} + w_D \frac{D_{s,j}}{D^{\max}} + w_J \frac{J_{s,j}}{J^{\max}} + w_X \frac{X_{s,j}}{X^{\min}}. \tag{11}$$

Предложенный способ формирования метки обеспечивает вычисление параметров *QoS*, изменяющихся при прохождении пакета из узла *i* в узел *j* по ребру e_{ij}, не только путем суммирования, но фактически по любой формуле (например, (10)) или алгоритму, которые могут описывать *QoS*-маршрутизацию. Это позволяет в дальнейшем уточнять рассматриваемую модель в части способов вычисления параметров *QoS* по пути *p*. Поиск оптимального пути только среди *QoS*-осуществимых путей обеспечивается на основе формирования функции вида (10). При этом алгоритм просмотра узлов, правила замены меток узлов, предложенные Дейкстрой, остаются неизменными и используют в качестве стоимостей путей величины $R_{s,j}$.

Доказательство корректности предложенного алгоритма основывается на том, что прохождение любого ребра сети не может улучшить (уменьшить) значение интегрального критерия качества, вычисляемого по формулам (6)–(11). Далее доказательство аналогично доказательству корректности базового алгоритма Дейкстры [8].

Вычислительная сложность рассмотренной модификации алгоритма Дейкстры, как и базового алгоритма, зависит от способа поиска вершины с минимальным значением метки, а также способов хранения множества непосещенных вершин и обновления меток и может варьироваться от $O\left(n^2\right)$ до $O\left(n^2 + m\right)$, где *n* – число узлов; *m* – число ребер сети.

Для вычисления меток по формулам (7)–(12) необходимо хранить информацию обо всех параметрах *QoS* каждого телекоммуникационного канала (каждого ребра графа). Простейшим образом это можно сделать, используя матрицу смежности **A** размера $n \times n$, каждый элемент которой представляет собой вектор, имеющий следующие компоненты:

$$\vec{A}_{ij} = \left(D_{ij}, J_{ij}, X_{ij}, Y_{ij}\right), \tag{12}$$

где значения $D_{ij} = D_e, J_{ij} = J_e, X_{ij} = X_e, Y_{ij} = Y_e$ есть величины параметров качества сервиса, заданные для дуги *e*, соединяющей узлы *i* и *j*. Такую матрицу смежности удобно представлять в виде четырех отдельных матриц для каждого из параметров. Например, для сети на рисунке эти матрицы будут выглядеть следующим образом (для наглядности вместо матрицы логарифмов вероятности прохождения пакетов **X** приведена матрица вероятности потерь пакетов **Z**):

$$Y = \begin{pmatrix} 0 & 20 & 0 & 0 & 0 & 0 & 30 & 0 \\ 20 & 0 & 0 & 12 & 0 & 0 & 0 & 30 \\ 0 & 0 & 0 & 28 & 20 & 0 & 0 & 25 \\ 0 & 12 & 28 & 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 20 & 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 & 25 & 0 \\ 30 & 0 & 0 & 30 & 0 & 25 & 0 & 0 \\ 0 & 30 & 25 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ Мбит/с} \quad (13)$$

$$Z = \begin{pmatrix} 0 & 0,02 & 0 & 0 & 0 & 0 & 0,01 & 0 \\ 0,02 & 0 & 0 & 0,01 & 0 & 0 & 0 & 0,04 \\ 0 & 0 & 0 & 0,04 & 0,01 & 0 & 0 & 0,05 \\ 0 & 0,01 & 0,04 & 0 & 0 & 0 & 0,02 & 0 \\ 0 & 0 & 0,01 & 0 & 0 & 0,02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,02 & 0 & 25 & 0 \\ 0,01 & 0 & 0 & 0,02 & 0 & 0,02 & 0 & 0 \\ 0 & 0,04 & 0,05 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (14)$$

$$D = \begin{pmatrix} 0 & 4 & 0 & 0 & 0 & 0 & 2 & 0 \\ 4 & 0 & 0 & 8 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 6 & 4 & 0 & 0 & 7 \\ 0 & 8 & 6 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 3 & 0 \\ 2 & 0 & 0 & 6 & 0 & 3 & 0 & 0 \\ 0 & 4 & 7 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ мс} \quad (15)$$

$$J = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 0,05 & 0 & 0 & 2 \\ 0 & 2 & 4 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ мс} \quad (16)$$

Численные эксперименты

Варьируя в соответствии с некоторой стратегией значения весовых коэффициентов в (11), можно получить несколько допустимых решений, каждое из которых будет отличаться значимостью того или иного параметра, определяемой его весом. Это открывает широкие возможности поиска оптимального маршрута для конкретных приложений с соответствующими требованиями к параметрам качества обслуживания.

Например, для сети, показанной на рисунке и описываемой матрицами (13)–(16), решим с помощью предложенного алгоритма две задачи маршрутизации из узла 1 в узел 3 при следующих значениях параметров ограничений:

$$Y^{\min} = 20 \text{ Мбит/с}; \quad Z^{\max} = 0{,}2; \quad D^{\max} = 16 \text{ мс}; \quad J^{\max} = 8 \text{ мс}. \tag{17}$$

В первом случае решим задачу оптимальной маршрутизации для сервиса, ориентированного на обмен файлами. Для этого весовые коэффициенты выберем следующим образом:

$$w_Y = 0{,}7; \quad w_D = 0{,}1; \quad w_J = 0{,}1; \quad w_X = 0{,}1,$$

так как приоритетным параметром для передачи файлов является полоса пропускания. Решением будет маршрут 1–7–4–3 с параметрами:

- полоса пропускания $Y_{13} = 28$ Мбит/с;
- вероятность потери пакетов $Z_{13} = 0{,}07$;
- среднее время задержки $D_{13} = 14$ мс;
- среднее время вариации задержки $J_{13} = 8$ мс.

Вторую задачу будем решать для обеспечения голосового сервиса. В этом случае значения весовых коэффициентов будут следующими:

$$w_Y = 0{,}1; \quad w_D = 0{,}3; \quad w_J = 0{,}3; \quad w_X = 0{,}3.$$

Оптимальным путем маршрутизации станет путь 1–7–6–5–3 с параметрами *QoS*:

- полоса пропускания $Y_{13} = 20$ Мбит/с;
- вероятность потери пакетов $Z_{13} = 0{,}06$;
- среднее время задержки $D_{13} = 13$ мс;
- среднее время вариации задержки $J_{13} = 5{,}5$ мс.

Отметим, что оба пути являются *QoS*-осуществимыми, т. е. удовлетворяют ограничениям вида (1).

Теперь попробуем получить маршрут, который позволит сократить вариацию времени задержки за счет ухудшения других параметров. Для этого будем увеличивать весовой коэффициент w_D, уменьшая значения других. Отметим, что при построении интегрального критерия качества можно положить значения одного или нескольких весовых коэффициентов равными 0. Соответствующие параметры *QoS* при этом не будут влиять на значение критерия качества, однако ограничения на значения этих параметров по-прежнему остаются в силе и поиск выполняется только на подмножестве *QoS*-осуществимых путей. Так, при значениях весовых коэффициентов $w_Y = 0$; $w_D = 0{,}8$; $w_J = 0{,}1$; $w_X = 0{,}1$ получаем оптимальный маршрут 1–2–8–3 со следующими параметрами *QoS*:

- полоса пропускания $Y_{13} = 20$ Мбит/с;
- вероятность потери пакетов $Z_{13} = 0{,}11$;
- среднее время задержки $D_{13} = 15$ мс;
- среднее время вариации задержки $J_{13} = 4$ мс.

Формирование системы весовых коэффициентов для наилучшего обеспечения различных сервисов является отдельной инженерной задачей. Фактически эти коэффициенты определяются исходя из требований обеспечиваемого сервиса к параметрам *QoS*. При этом они могут зависеть от нормирования параметров качества обслуживания при вычислении аддитивной свертки (11), т. е. от задаваемых ограничений. На практике подбор этих коэффициентов выполняется в процессе исследования соответствующей модели в ходе вычислительного эксперимента. В результате, варьируя весовые коэффициенты по определенному алгоритму, можно получить набор резервируемых маршрутов (телекоммуникационных ресурсов) для предоставления комплекса услуг в мультисервисной сети.

Задачей второго уровня является предоставление самих услуг из набора зарезервированных. В этом случае также необходимо решить задачу целочисленного программирования. Используя ранее введенные обозначения, сформулируем задачу следующим образом:

$$\min \sum_{e \in p} \sum_{k=1}^{m} k_p(e) x_p(e) + d_k(e) L_k(e), \tag{18}$$

где k_p – стоимость выбора пути на всем отрезке от s к t; $x_p(e) \in \{0,1\}$; $d_p(e)$ – единица дополнительной стоимости, необходимой для обеспечения *QoS*-требования.

В настоящей статье представлены результаты исследования лишь первого уровня сервис-ориентированной архитектуры – уровня резервирования необходимых телекоммуникационных ресурсов для обеспечения заданного качества обслуживания. Что касается второго уровня – предоставления соответствующих телекоммуникационных услуг на основе зарезервированных ресурсов для обеспечения требуемого качества обслуживания, то данная проблема (18) находится в стадии активного исследования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Листопад Н. И. Модели оптимальной маршрутизации в компьютерных сетях // Труды БГТУ. Сер. VI, Физико-математические науки и информатика. 2006. Вып. XVI. С. 130–132.

2. Листопад Н. И., Матрук Аль Даллаен А., Копачев А. Г. Модели обеспечения живучести компьютерных сетей при оптимальной маршрутизации информационных потоков // Информатика. 2006. Вып. 4. С. 39–48.

3. Листопад Н. И., Величкевич И. О. Методы балансировки трафика в IP сетях // Докл. БГУИР. 2010. № 7 (53). С. 18–24.

4. Листопад Н. И., Величкевич И. О. Оптимальная маршрутизация информационных потоков с учетом параметров QOS // Докл. БГУИР. 2012. № 4 (66). С. 111–116.

5. Girlich E., Kovalev M. M., Listopad N. I. Optimal choice of the capacities of telecommunication networks to provide QoS-routing. Magdeburg, 2009.

6. A Scalable Approach to QoS-Aware Self-adaptation in Service-Oriented Architectures / V. Cardellini [et al.] // Proc. of 6th Intern. ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Q Shine 2009 and 3rd Intern. Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, AAA-IDEA 2009 (Las Palmas, Gran Canaria, Nov. 23–25, 2009). Berlin ; Heidelberg ; New York, 2009. P. 431–447.

7. Listopad N. I., Kopachev A. G., Matruk A. A. Quality of Service at the Computer Networks Based on Internet // Системні дослідження та інформаційні технології. 2006. № 4. С. 71–76.

8. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. 1959. Vol. 1. P. 269–271.

Поступила в редакцию 14.10.2014.

Николай Измайлович Листопад – доктор технических наук, профессор, заведующий кафедрой информационных радиотехнологий факультета радиотехники и электроники УО «Белорусский государственный университет информатики и радиоэлектроники», директор Учреждения «Главный информационно-аналитический центр Министерства образования Республики Беларусь».

Юрий Иосифович Воротницкий – кандидат физико-математических наук, доцент, заведующий кафедрой телекоммуникаций и информационных технологий факультета радиофизики и компьютерных технологий БГУ, начальник Центра информационных технологий БГУ.

Ахмед Абдулазиз Хайдер – аспирант кафедры информационных радиотехнологий факультета радиотехники и электроники УО «Белорусский государственный университет информатики и радиоэлектроники». Научный руководитель – Н. И. Листопад.

УДК 003.26:51:004(075.8)

Т. В. ГАЛИБУС, Г. В. МАТВЕЕВ

ВЕРИФИКАЦИЯ ПАРАМЕТРОВ МОДУЛЯРНОГО РАЗДЕЛЕНИЯ СЕКРЕТА

Предложены метод верификации модулярной схемы разделения секрета в кольце полиномов и алгоритм проверки дилера, исключающий возможность распределения некорректных данных. Алгоритм является модификацией метода Фельдмана и основан на вычислительной сложности решения задачи дискретного логарифмирования. В отличие от существующих алгоритмов верификации для модулярных схем в кольце целых чисел наш метод автоматически гарантирует верификацию как промежуточного значения секрета, так и порогового значения *t*. Тем самым завершено построение криптостойкой верифицируемой пороговой схемы разделения секрета. Попутно проанализированы методы верификации параметров схем разделения секрета в кольце целых чисел.

Ключевые слова: разделение секрета; метод Фельдмана; модулярные параметры; верификация разделения секрета; пороговые схемы.

We construct a verification scheme for the polynomial modular secret sharing and propose a dealer verification algorithm that guarantees the correctness of the distributed data. We compare the constructed algorithm with the same for the integer modular secret sharing scheme. Our method is the modification of Feldman's verifiable secret sharing scheme and is based on the complexity of discrete logarithm computation. The proposed method guarantees the verification of all modular (t, k)-threshold secret sharing scheme parameters including the intermediate secret value and the threshold value t. We construct the Feldman verification in the univariate polynomial ring over F_q and provide conditions on the size of the finite field in which the scheme is constructed. Our algorithm constructs the verifiable cryptographically secure or perfect modular secret sharing scheme.

Key words: secret sharing; Feldman method; modular parameters; secret sharing verification; threshold schemes.

Схемы разделения секрета (СРС) лежат в основе многих криптографических протоколов в распределенных системах. Разделение секрета применяется для совместных конфиденциальных вычислений [1], шифрования на основе атрибутов [2] и электронного защищенного голосования [3]. Важной задачей в разделении секрета является построение таких схем, с помощью которых пользователи могут проверить корректность секрета и тем самым не допустить обман со стороны остальных участников и дилера. Такие схемы строятся на основе протоколов с нулевым разглашением [4]. В схемах верифицируемого разделения секрета (СВРС) дилер распределяет информацию о секретном значении среди участников таким образом, что для честных пользователей гарантируется получение ими значения секрета, а для нечестных – невозможность восстановить секрет.

СВРС позволяет честным пользователям, т. е. тем, которые следуют протоколу восстановления секрета, проверить корректность частичных секретов при их распределении и восстановлении исходного