

УДК 004.7

И.В. КОРОЛЬКОВ

Запорожский национальный технический университет, Украина

**ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ МОДЕЛИРОВАНИЯ  
МАРШРУТИЗАЦИИ В БОЛЬШИХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ**

Рассмотрен метод алгоритмической маршрутизации как механизм абстракции при моделировании маршрутизации в вычислительных сетях. Предложены некоторые модификации метода, нацеленные на повышение его производительности и масштабируемости. Проведен анализ возможности применения исходного и модифицированных методов для моделирования сетей разных размеров.

**вычислительная сеть, маршрутизация, моделирование, граф, дерево, узел, маршрут****Введение**

Ключевая особенность маршрутизации в Интернет состоит в том, что каждый маршрутизатор поддерживает свою собственную таблицу маршрутизации, даже если она построена путем обмена сообщениями со своими соседями. Эта необходимость обусловлена децентрализованной природой сети Интернет.

Тем не менее, в отличие от проектирования реального маршрутизатора, при моделировании можно не принимать во внимание некоторые существующие ограничения. Так, например, моделируемый маршрутизатор может иметь доступ к глобальной информации о сети, которая недостижима в реальных условиях, что значительно облегчает задачу расчета маршрутов. Приемлемость такого допущения определяется целями моделирования. Если подробности работы протоколов маршрутизации не представляют интерес, то с точки зрения моделирования более эффективным является применение сохраненных предварительно рассчитанных маршрутов. Централизованный расчет маршрутных схем является общепринятым допущением и имеет ряд преимуществ: в процессе моделирования нет необходимости во включении сложных протоколов, снижается вычислительная нагрузка и требуемый объем памяти.

Такой подход требует выделения памяти порядка  $O(N^2)$ , где  $N$  – число узлов в сети. Каждому маршрутизатору должен быть известен путь к каждому другому маршрутизатору. Так, например, для сети из 10 000 узлов плоская таблица маршрутизации, где каждая запись представлена единственным четырехбайтовым целым, займет 400 МБ памяти.

В [1] показано, что таблица маршрутизации в иерархической сети имеет минимальный размер, когда она разделена на  $\log N$  уровней. Для сети, разделенной на  $\log_k N$  уровней, где  $k$  – коэффициент ветвления, общая пространственная сложность имеет вид:  $O(kN \log_k N)$ . Сеть Интернет иерархична по своей природе, однако имеет только два уровня иерархии, следовательно:

$$\log_k N = 2 \Rightarrow k = \sqrt{N}.$$

Таким образом, общая пространственная сложность имеет вид:  $O(2N\sqrt{N})$ .

**Постановка задачи.** Метод алгоритмической маршрутизации (АМ) был выдвинут в [2], далее развит в [3] и внедрен в систему моделирования NS [4]. Идея метода основана на работе с сетями бинарных деревьев [5] и предполагает снижение использования памяти; при этом несколько возрастает вычислительная сложность. Метод АМ накладывает  $k$ -ое дерево на исходную топологию сети. Это дерево

может быть сгенерировано путем поиска в ширину или в глубину сети. При этом выбор алгоритма поиска оказывает влияние на точность маршрутизации. Значение  $k$  соответствует максимальному числу потомков, охваченных любым узлом в дереве. В случае, если в сети присутствуют петли, они разрушаются на этапе отображения.

На рис. 1 показан пример отображения топологии сети на дерево. Разорванные связи обозначены пунктирными линиями.

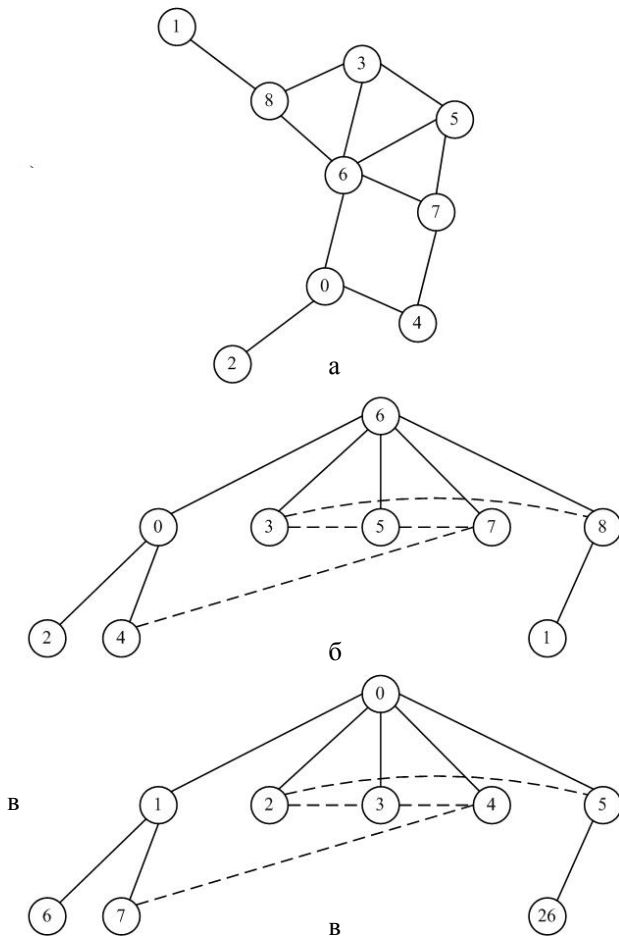


Рис. 1. Пример отображения на дерево простой сети:  
 а – исходная топология сети;  
 б – топология сети, отображенная на дерево;  
 в – дерево с новым адресами узлов

Второй этап вводной стадии состоит в назначении новой системы нумерации узлов. Корень дерева обозначается 0. Другие узлы нумеруются, как показано на рис. 1, в, где потомок узла  $i$  имеет вид  $ki + j$ ,  $j \in 1 \dots k$ , а предок любого узла  $i$  определя-

ется как:

$$Parent(i) = \left\lfloor \frac{i-1}{k} \right\rfloor. \quad (1)$$

Размер памяти, необходимый для новой схемы адресации –  $O(N)$ .

Каждый раз, когда пакет достигает промежуточного маршрутизатора, он должен быть отправлен на один шаг далее. Предположим, что пакет находится в узле, обозначенном  $S$  в соответствии с новой схемой  $k$ -го дерева, а пункт его назначения – узел  $D$ ; теперь рассчитаем следующую пересылку в маршруте. Поиск начинаем от узла-источника  $D$  и следуем от потомка к предку вверх вдоль дерева, вплоть до достижения корневого узла или узла-назначения  $S$ . Если вначале достигнут узел  $S$ , тогда узел, предшествующий  $S$ , считается следующей пересылкой в маршруте. Если же вначале достигнут корневой узел, тогда следующей пересылкой в маршруте является предок  $S$ . Таким образом, пакет пересылается по линии, ведущей к следующему узлу маршрута.

Поскольку дерево имеет примерную глубину  $\log N$ , алгоритм имеет вычислительную сложность  $O(\log N)$ .

Существует ряд проблем, связанных с реализацией АМ. Данный метод требует назначения каждому узлу сети нового идентификатора. Если дерево имеет глубину  $d$ , тогда наибольшее число, необходимое для отображения:

$$L = \sum_{i=0}^{d-1} k^i = \frac{k^d - 1}{k - 1} - 1 \approx k^{d-1}.$$

Даже в небольшой сети при условии сочетания умеренной глубины и наличия отдельного узла с большим количеством соседей использование АМ может привести к переполнению четырехбайтового целого.

Такое явление при переадресации отображенных узлов влечет за собой две проблемы. Во-первых, объем памяти, необходимый для адресации узлов,

быстро превышает стандартное четырехбайтовое целое и требует размер  $O(N \log_2(k^{d-1}))$  бит. Во-вторых, новые адреса редко распределены в диапазоне  $0 \dots L$ , в особенности это имеет место в случае, когда максимальное число соседей намного больше среднего. Для указания на ячейку памяти, в которой хранятся данные об узле с исходными адресами в диапазоне  $0 \dots (N-1)$ , может использоваться простой массив длиной  $N$ , в то время как для отображенных адресов потребуется массив длиной  $L$ , что весьма неэффективно. Для решения этой проблемы могут быть использованы методы разреженных матриц или древовидная структура данных, однако оба эти подхода ведут к увеличению потребления памяти, а также вносят дополнительную вычислительную сложность.

**Целью данного исследования является** модификация исходного метода АМ для улучшения производительности и масштабируемости моделирования больших вычислительных сетей.

### Прямая алгоритмическая маршрутизация

На этапе отображения сети на дерево каждому узлу назначается новый адрес. По данному адресу узла можно определить его родительский узел и любые возможные узлы-потомки. Для работы алгоритма этой информации более чем достаточно, поскольку для вычисления следующей пересылки не-обходимой является лишь возможность определения узла-родителя.

Прямой алгоритмической маршрутизацией (ПАМ) назовем модификацию исходного метода, устраняющую необходимость в переадресации отображенных узлов. При этом предполагается, что каждый узел хранит в массиве список своих соседей. Когда сеть отображается на дерево, вместо того, чтобы назначить узлу новый адрес, список соседних узлов сортируется таким образом, чтобы узел-предок имел определенную позицию в списке,

например, в начале или в конце. При этом корень дерева должен храниться явно. В случае если сортировка списка соседей является излишне вычислительно затратной, предок каждого узла может быть сохранен явно (для этого требуется  $O(N)$  памяти). Даже в таком случае данная модификация позволяет улучшить производительность.

Рассмотрим небольшую сеть, изображенную на рис. 2. Изначально она может быть описана, как показано на рис. 3, а. После отображения сети на дерево, отмечается корень, а списки соседей сортируются таким образом, чтобы предок находился на первой позиции в каждом списке (рис. 3, б). Таким образом, получаем систему, в которой древовидная структура неявно присутствует в обычной структуре сети. Предок узла может быть непосредственно определен путем просмотра первого элемента списка. Кроме того, могут быть определены потомки узла, т.е. узлы-соседи, для которых исходный узел является предком. Определение узлов-потомков становится более сложным, что, однако, не составляет значительную проблему, поскольку в этом нет необходимости для определения следующей пересылки в маршруте.

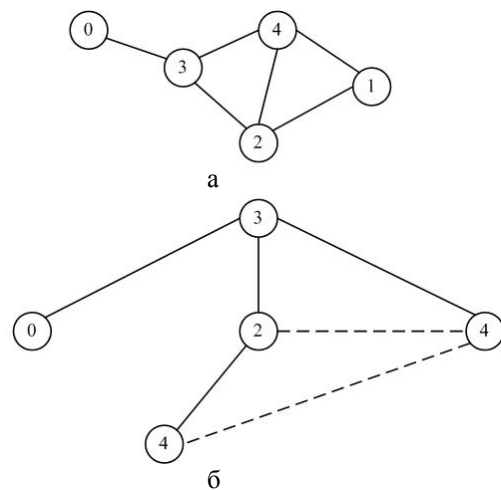


Рис. 2. Пример простой сети:  
а – исходный граф сети;  
б – та же сеть, отображенная на дерево

Метод ПАМ имеет несколько ощутимых преимуществ. Прежде всего, для его реализации не тре-

бується виділення додаткових об'ємів пам'яті для збереження нових адресів вузлів. Оскільки не вводиться нова система адресації, алгоритм не підвержен затронутій раніше проблемі переповнення адресів. Нарешті, знаходження батьківського вузла шляхом прямого пошуку проводиться швидше, ніж виконання розрахунку  $\lfloor (i-1)/k \rfloor$ , тим більше, що операція ділення має відносно високу вичисельну складність. При цьому вичислення наступної пересилки так же, як і при використанні початкового алгоритму АМ, має складність  $O(\log N)$ .

Вузел	Сусіди
0	3
1	2 4
2	1 3 4
3	2 4 0
4	1 2 3

а

Вузел	Сусіди
0	3
1	2 4
2	3 1 4
3 (корінь)	2 4 0
4	3 1 2

б

Рис. 3. Два варіанти представлення мережі, зображеної на рис. 2: а – до відображення; б – після відображення

### Алгоритмічна маршрутизація з фіксованими вичисельними витратами

Розглянемо рис. 4: якщо вузли пронумеровані за допомогою системи нумерації, передбаченої методом АМ, то повинна бути можливість визначити, знаходиться чи вузол-адресат в ділянці А, В чи С, тобто. являється чи наступна пересилка до вузла-предка чи до одного з вузлів-потомків. Подібна схема нумерації і пов'язаний з нею алгоритм складають основу методу алгоритмічної маршрутизації з фіксованими вичисельними витратами (ФАМ). Алгоритм ФАМ потребує  $O(N)$  пам'яті і має вичисельну складність  $O(1)$ .

Замість використання одного числа як адреси вузла запропонована схема передбачає використання двох чисел: перша частина адреси є унікальною серед всіх інших вузлів, знайдених на тій же глибині в дереві; друга частина адреси соот-

ветствует глубине узла в дереве. При этом корневой узел имеет адрес  $(0; 0)$ . Как и в методе ПАМ, список соседних узлов сортируется на каждом узле таким образом, чтобы узел-предок находился в конце списка.

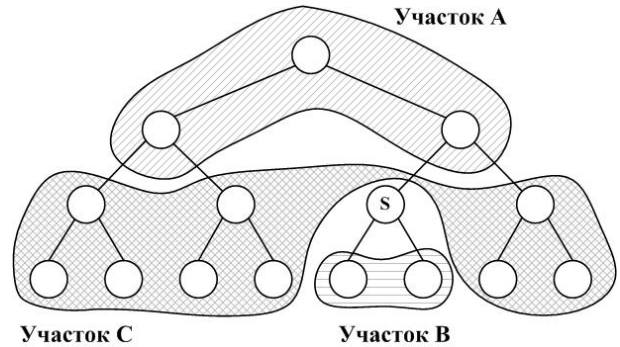


Рис. 4. Пример дерева сети, разбитого на три участка

Пусть  $k$  – максимальное число соседей, принадлежащих любому узлу в сети. Родительский узел обозначим как  $P$ , а узел-потомок – как  $C$ , тогда  $(X_P, Y_P)$  – адрес  $P$ , а  $(X_C, Y_C)$  – адрес  $C$ . Пусть  $C$  будет  $i$ -ым узлом в списке соседей узла  $P$ , где  $i \in \{0 \dots (k-1)\}$ . Тогда  $(X_P, Y_P)$  и  $(X_C, Y_C)$  связаны следующим образом:

$$Y_C = Y_P + 1; \tag{2}$$

$$X_C = X_P \times k + i. \tag{3}$$

Пример такой схемы адресации для исходной сети (рис. 4) представлен на рис. 5.

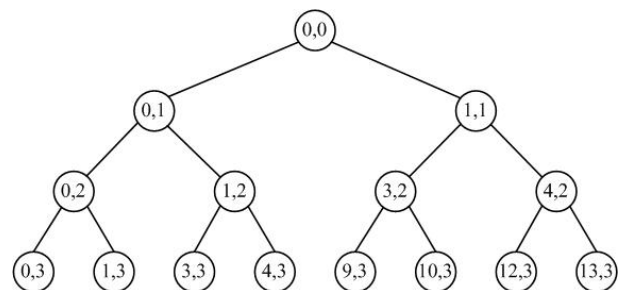


Рис. 5. Пример дерева сети, разбитого на три участка

Пусть  $S$  и  $D$  соответственно обозначают источник и пункт назначения пакета. Изначально, на основании адресов узлов, узел  $D$  может быть отне-

сен к одному из трех участков (рис. 4). Первый участок А содержит все узлы, находящиеся в дереве выше  $S$ , т.е. если  $Y_D < Y_S$ , то узел  $D$  находится на участке А. Второй участок В содержит все узлы, находящиеся в дереве непосредственно ниже узла  $S$ , т.е. выполняется условие  $Y_D > Y_S$ . Таким образом, из выражения (3) следует, что если узел  $D$  находится в этом участке, то:

$$X_D \in \{X_C \times k^{(Y_D - Y_S)}, \dots, X_C \times k^{(Y_D - Y_S)} + k - 1\}. \quad (4)$$

Если узел  $D$  не принадлежит ни одному из первых двух участков, тогда он принадлежит третьему участку С, содержащему узлы, находящиеся в обо-  
собленных поддеревьях ниже узла  $S$ .

Если узел назначения находится на участке А или С, тогда следующей пересылкой из узла-источника  $S$  будет пересылка к предку  $S$ . Если узел назначения находится на участке В, то может существовать более чем один узел-потомок, который может быть использован для следующей пересылки. Пусть

$$Q = X_D - X_S \times k^{(Y_D - Y_S)}, \quad (6)$$

тогда положение узла следующей пересылки в списке соседей узла  $S$  определяется как целая часть от

$$\left[ \frac{Q}{k^{(Y_D - Y_S - 1)}} \right]. \quad (7)$$

Метод ФАМ предполагает фиксированные вычислительные затраты для расчетов всех следующих пересылок, что, однако, само по себе не делает моделирование всех сетей более эффективным. В настоящее время осуществимым является моделирование сетей, содержащих менее чем один миллион узлов. Метод АМ для нахождения следующего узла в цепочке маршрута требует в среднем  $\log N$  вычислений – примерно 14 вычислений для сети с миллионом узлов. При использовании ФАМ для этого требуется единственное действие, однако это действие может иметь более чем в 14 раз большую сложность. При выполнении поиска узла-предка с ис-

пользованием исходного метода АМ наиболее вычислительно сложным действием является операция деления. Метод ФАМ дважды требует нахождения  $k^{\Delta Y}$ , а также выполнения нескольких операций умножения, сложения и деления. Операция возведения в степень характеризуется значительной вычислительной сложностью. Существует простое решение этой проблемы: поскольку  $\Delta Y$  может принимать сравнительно малый диапазон значений, как правило, от 1 до  $\log N$ , значительно проще выполнить предварительное вычисление, а затем производить поиск по таблице, чем выполнять повторные операции возведения в степень.

Метод ФАМ, как и исходный метод АМ, подвержен проблеме в работе с большими значениями адресов. Наибольшее число, используемое для адресации, имеет порядок  $k^d$ , где  $k$  – наибольшая степень узла, а  $d$  – глубина дерева. Если размер адреса ограничен четырех- или восьмибайтным целым числом, то размер возможной моделируемой сети ограничен. Для преодоления существующих ограничений могут быть использованы такие библиотеки для работы с арифметикой с многократно увеличенной точностью (МУТ), как, например, GNU MP [6]. Недостатком такого решения является наличие нелинейной зависимости от  $N$  объема требуемой памяти, а также появление нефиксированных вычислительных затрат.

### Масштабируемость и производительность

Проведем сравнение рассмотренных выше методов для моделирования сетей разных размеров с точки зрения их требований к памяти и вычислительным ресурсам.

Следует отметить, что при использовании для адресов узлов библиотек для работы с арифметикой с многократно увеличенной точностью объем памяти для каждого адреса не является фиксированным, а зависит от максимальной степени узла и глубины

дерева. Число машинных операций, необходимых для работы с такими адресами, зависит от размеров операндов, что усложняет задачу оценки вычислительных затрат.

Результаты сравнения производительности рассмотренных выше методов для моделирования сетей разных размеров представлены на рис. 6. Сравнение производилось по результатам выполнения фиксированного числа операций поиска следующей пересылки.

Все топологии были все сформированы с помощью GT-ITM [7]. При этом сети подражали структуре Интернет: узлы были разделены на магистральные и периферийные.

В табл. 1 подытожены теоретические свойства пяти описанных методов. Принимая во внимание эти свойства, рассмотрим экспериментальные результаты, представленные на рис. 6.

Таблица 1  
Требования к памяти и вычислительным ресурсам некоторых методов АМ

Алгоритм	Память	Операция поиска
АМ	$O(N)$	$O(\log N)$ делений
ПАМ	$O(0)$	$O(\log N)$ поисков
ФАМ	$O(N)$	1 деление, 1 умножение
АМ с МУТ	$< O(N \log k^{d-1})$	$O(\log N)$ делений МУТ
ФАМ с МУТ	$< O(N \log k^{d-1})$	1 деление МУТ, 1 умножение МУТ

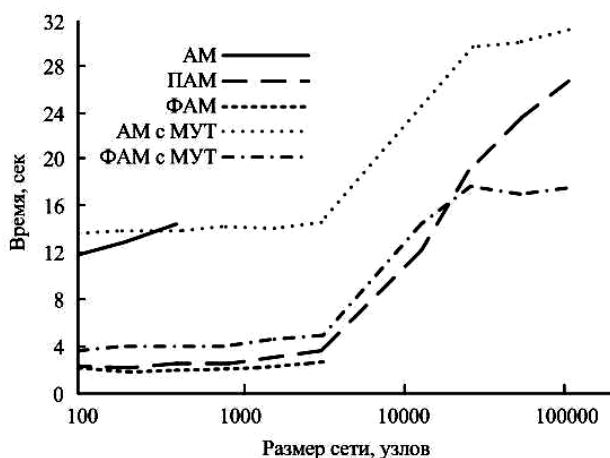


Рис. 6. Время, необходимое для моделирования сети с использованием разных методов

Заметно, что рис. 6 имеет разделение на три четкие области, соответствующие сетям, содержащим до 4 000 узлов, сетям с количеством узлов от 4 000 до 20 000 и сетям с более чем 20 000 узлов. Наилучшую производительность при моделировании небольших сетей имеет метод ФАМ, хотя его результат довольно близок к ПАМ. Как исходный метод АМ, так и метод ФАМ не способны обеспечить моделирование сетей больших некоторого размера (800 узлов для исходного метода АМ и 3000 – для ФАМ), поскольку адреса отображенных узлов выходят за пределы стандартных целых чисел.

Весьма неожиданные данные получены для сетей средних размеров: относительная разница производительности методов сохраняется неизменной, однако присутствует значительное увеличение времени выполнения моделирования. Это может быть обусловлено влиянием кэш-памяти: сеть, содержащая более чем 2 000 узлов, не может храниться в кэш-памяти, поэтому производительность снижается сравнительно резко.

Для сетей с более чем 20 000 узлов, большая часть которых хранится в основной памяти, полученные данные вновь подтверждают ожидаемые. Метод ПАМ и исходный метод АМ проявляют зависимость  $O(\log N)$ , в то время как метод с постоянными вычислительными затратами сохраняет примерно одинаковую производительность.

## Выводы

Важной составной частью моделирования вычислительных сетей является моделирование протоколов маршрутизации. При выполнении моделирования должны быть сбалансированы следующие факторы: скорость, точность и масштабируемость. Наиболее точный метод предполагает подробное моделирование протоколов маршрутизации и построение таблиц маршрутизации для каждого узла сети. Тем не менее, если каждый из моделируемых маршрутизаторов поддерживает полные таблицы

маршрутизації, значительные требования к памяти препятствуют моделированию больших сетей. Для решения этой проблемы в модель может быть введен некоторый уровень абстракции.

Метод АМ предусматривает отображение сети на дерево и использует простой алгоритм расчета маршрута между двумя заданными узлами. При этом за счет небольшого увеличения вычислительной сложности предполагается значительное снижение требуемого объема памяти.

Поскольку метод алгоритмической маршрутизации предполагает хорошую масштабируемость, предложены некоторые модификации, касающиеся его производительности.

Проведено изучение производительности методов АМ для моделирования сетей разных размеров. Метод ПАМ имеет преимущество в виде отсутствия дополнительных требований памяти и хорошей масштабируемости приблизительно до 40 000 узлов. Также несомненным преимуществом метода является его простота. Для небольших сетей, где не стоит вопрос нехватки памяти, наилучшую производительность имеет метод ФАМ. Для больших сетей выбор метода сводится к нахождению баланса между скоростью моделирования и требованиями к памяти. Метод ФАМ, использующий арифметику с многократно увеличенной точностью, имеет примерно постоянную производительность, однако требует выделения дополнительной памяти, в то время как метод ПАМ не предъявляет таких требований, но является более медленным.

Несмотря на то, что использование алгоритмической маршрутизации добавляет в задачу моделирования некоторую вычислительную сложность, она компенсируется за счет снижения требований к объемам памяти. Задачу моделирования сверхбольших сетей невозможно решить без использования таких методов абстракции, как АМ.

## Литература

1. Kleinrock L., Kamoun F. Hierarchical Routing for Large Networks, Performance Evaluation and Optimization // Computer Networks. – January 1977. – Vol. 1, No. 3. – P. 155-174.
2. Polly Huang. Enabling Large-scale Network Simulations: A Selective Abstraction Approach. – PhD thesis, University of Southern California, 1999.
3. Huang Polly, Heidemann John. Minimizing routing state for light-weight network simulation // Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. – Cincinnati, Ohio, USA, August 2001. – IEEE.
4. Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zapala. Improving simulation for network research. Technical Report 99-702b, University of Southern California, March 1999 (revised September 1999).
5. Raman S., McCanne S., Shenker S. Asymptotic scaling behavior of global recovery in SRM // Proceedings of SIGMETRICS/PERFORMANCE 98, Joint International Conference on Measurement and Modeling of Computer Systems. – 1998.
6. Free Software Foundation. GNU MP, 2006. [Электронный ресурс]. – Режим доступа: <http://www.swox.com/gmp>.
7. Kenneth L. Calvert, Matthew B. Doar, Ellen W. Zegura. Modeling internet topology // IEEE Communications Magazine. – June 1997. – 35(6): – P. 160-163.

*Поступила в редакцию 25.01.2007*

**Рецензент:** д-р техн. наук, проф. И.А. Жуков, Национальный авиационный университет, Киев.