

**А. И. Миков**, д-р физ.-мат. наук, проф. кафедры информационных технологий,  
e-mail: Alexander\_Mikov@mail.ru,

Кубанский государственный университет, г. Краснодар,

**Е. Б. Замятина**, канд. физ.-мат. наук, доц. кафедры информационных технологий в бизнесе,  
e-mail: ezamyatina@hse.ru,

Национальный исследовательский университет "Высшая школа экономики", г. Пермь,

**С. М. Калашников**, студент,

Пермский государственный национальный исследовательский университет

## Исследования алгоритмов маршрутизации в SON сетях с использованием программных средств имитационного предсказательного моделирования<sup>1</sup>

*Представлены результаты исследования одного из алгоритмов маршрутизации в ad-hoc сетях, а именно, в SON (самоорганизующихся сетях, от англ. self-organizing network), методом имитационного моделирования. Рассматриваются разновидности ad-hoc сетей и особенности реализации алгоритмов маршрутизации в этих сетях. Внимание сфокусировано на инструментальных средствах имитационного моделирования, указываются требования к программным средствам подобного рода, приводятся конструкции языка имитационного моделирования Triad. Моделирование выполняется в инструментальной среде TriadNS, разработанной авторами.*

**Ключевые слова:** ad-hoc сети, SON, алгоритмы маршрутизации, имитационное моделирование, язык имитационного моделирования

### Введение

Классическая ad-hoc сеть представляет собой динамическую мобильную сеть без четкой инфраструктуры, которая формируется автоматически в результате возникающих соединений между мобильными узлами. Набор этих мобильных узлов располагается в некоторой области и характеризуется тем, что центральный управляющий узел отсутствует. Каждый узел ad-hoc сети может функционировать как маршрутизатор, организуя и поддерживая процесс маршрутизации как к другим узлам в той же сети, так и к устройствам, которые являются передатчиками или приемниками. Узлы могут выполнять как пересылку данных, так и выполнять работу, связанную с конкретным назначением сети.

К особенностям ad-hoc сетей можно отнести также динамически изменяющиеся топологии и ограниченные энергетические ресур-

сы. Узлы сети могут изменять свое местоположение, условия распространения сигнала также могут меняться. Более того, может изменяться полоса пропускания и зона видимости радиосигнала.

Цели, для которых используют ad-hoc сети, могут быть самыми разнообразными. Известно, что ad-hoc сети могут быть использованы структурами МЧС, в транспортных системах, БПЛА (беспилотными летательными аппаратами) и т.д., нередко в тех ситуациях, когда надо развернуть сеть на некоторое время для передачи данных.

В настоящее время существует большое число систем имитационного моделирования для проектирования и анализа компьютерных сетей. Эти системы моделирования можно разделить на специализированные (COMNET [1], OMNet + + [2], OPNET [3], NS-2[13] и т.д.) и системы моделирования общего назначения (к примеру, ANYLOGIC [4]).

Авторы представляют в статье инструментальные средства автоматизированного проек-

<sup>1</sup>Работа поддержана Российским фондом фундаментальных исследований, проект № 18-01-00359.

тирования и имитационного моделирования TriadNS[5,6]. Одной из особенностей TriadNS является способ представления имитационной модели в виде трех слоев: слоя структур (вычислительные узлы и линии связи компьютерной сети), слоя рутин (описывает поведение вычислительных узлов), слоя сообщений (описание сообщений и данных сложной структуры, предполагается, что вычислительные узлы обмениваются сообщениями).

Для описания имитационной модели в TriadNS используют язык Triad [7, 8], синтаксические конструкции языка будут представлены ниже.

Далее статья организована следующим образом: в первом разделе описаны особенности самоорганизующихся сетей (SON), во втором разделе рассмотрены динамические графы, которые являются математической моделью SON. Далее обсуждается система моделирования TriadNS и результаты эксперимента.

### Самоорганизующиеся сети (SON)

Один из подходов к классификации беспроводных сетей предполагает их разделение на сети с централизованной инфраструктурой и самоорганизующиеся сети. Особенностью SON [9, 10] является их способность при отсутствии централизованной инфраструктуры организовывать обмен информацией между любыми двумя вычислительными узлами в зоне радиопокрытия.

Как уже отмечалось ранее, вычислительные узлы в SON можно разделить на конечные хосты и маршрутизаторы. Соединения между конечными узлами-хостами организуются на длинные расстояния через промежуточные узлы-маршрутизаторы с помощью специальных протоколов маршрутизации. Такое соединение называют "многошаговым" или "многоэтапным" (multi-hop). Каждый шаг — это участие одного из узлов-маршрутизаторов сети в передаче данных от одного узла до другого. Вычислительные узлы сети обладают способностью находить друг друга, чтобы обеспечить передачу данных и формировать сеть. В случае выхода из строя одного из узлов вычислительные узлы SON организуют новые пути для передачи данных.

К классу самоорганизующихся сетей можно отнести:

- а) беспроводные мобильные ad-hoc сети (MANET) [15—17, 25, 26];
- б) беспроводные сенсорные сети (WSN) [13, 14, 23];

- с) беспроводные ячеистые сети (WMN) [13];
- д) оппортунистические сети WSN DTN (ODTN, Opportunistic Delay Tolerant Networks) [21];
- е) автотранспортные ad-hoc сети (VANET) [18—20];

- ф) гетерогенные ad-hoc сети (HANET) [12];
- г) ad-hoc сети для летательных аппаратов (FANET) [11, 24].

SON могут быть использованы:

- а) в системах безопасности;
- б) для мониторинга оборудования;
- в) в интернете вещей.

Итак, как уже упоминалось ранее, вычислительные узлы SON сетей могут устанавливать новые маршруты для передачи данных, таким образом, координаты вычислительных узлов могут динамически изменяться.

Структура классических проводных сетей может быть представлена виде графа  $G = (V, E)$ , где  $V$  — множество вершин (вычислительных узлов сети),  $E$  — множество ребер (дуг) (линии связи между вычислительными узлами).

SON могут быть описаны с помощью динамических геометрических графов. Рассмотрим их более подробно.

### Динамические геометрические графы

Геометрический граф — это набор  $GG = \{V; R; P; r\}$ , где  $V$  — множество вершин;  $R$  — ориентированная ограниченная поверхность, на которой располагаются вершины;  $P$  — отображение,  $P: V \rightarrow R$  (геометрическая конфигурация — локализация вершин);  $r$  — числовой параметр, интерпретируемый как радиус действия приемопередатчиков компьютерных узлов сети. Геометрический граф является математической моделью гомогенной беспроводной компьютерной сети, его вершины соответствуют узлам сети.

В мобильных компьютерных сетях вершины перемещаются в области  $R$ , поэтому их модели являются динамические геометрические графы  $DGG = \{V; R; T; S; r\}$ , где  $T$  — упорядоченное множество чисел, обычно интерпретируемое как множество моментов времени;  $S$  — отображение,  $S: V \times T \rightarrow R$  (поток геометрических конфигураций — локализации вершин) [27, 28]. В динамическом геометрическом графе ad-hoc сети координаты узлов, определяемые отображением  $S$ , как правило, являются непрерывными функциями времени.

Для решения задач маршрутизации сообщений модель динамического геометрического графа является в некотором смысле избыточной. Например, смещение всех вершин

в одном направлении на одинаковое расстояние никак не влияет на выбор оптимального маршрута. Изотропное изменение размера области с соответствующим изменением величины  $r$ , а также масштаба временной оси также сохраняет свойства маршрутов. Поэтому для решения задач, поставленных в статье, целесообразно построить отображение множества динамических геометрических графов сетей в семейства обыкновенных графов. Для этого фиксируем некоторый момент времени  $t \in T$ , выбирая одну геометрическую конфигурацию из потока. Она определяет расстояния между вершинами, часть из которых меньше  $r$ . Соответствующие пары узлов сети "слышат" друг друга, следовательно, в обыкновенном графе  $G(t)$  между ними имеется ребро, которое можно использовать для построения маршрутов передачи данных. Переход к новому моменту времени  $t + \Delta t$  может не изменить ситуации, и обыкновенный граф  $G(t + \Delta t)$  останется таким же,  $G(t + \Delta t) = G(t)$ , либо граф изменится. Хотя отображение  $S$  непрерывно, изменение графа  $G(t)$  во всех случаях дискретно, т.е. динамический граф  $DGG$  сводится к последовательности  $G_0, G_1, G_2, \dots, G_i, \dots$  обыкновенных графов.

Такой подход позволяет естественным образом вводить классы динамических геометрических графов, определяя их на основе свойств обыкновенных графов.

Назовем динамический геометрический граф *связным на интервале*  $[a, b]$ , если он отображается в такое семейство  $\{G_i\}$ , что все обыкновенные графы  $G(t)$ ,  $a \leq t \leq b$ , связны.

Назовем динамический геометрический граф *стабильным на интервале*  $[a, b]$ , если он отображается в такое семейство  $\{G_i\}$ , что все обыкновенные графы  $G(t)$ ,  $a \leq t \leq b$ , изоморфны.

Если перемещение узлов мобильной компьютерной сети заранее неизвестно, непредсказуемо, то используется математическая модель случайного геометрического графа. В этой модели позиции вершин в графе  $G_0$  (геометрические конфигурации) считаются случайными векторами с заданным распределением вероятностей. В исследовании, как правило, для всех вершин задается одно и то же распределение, для ограниченной области расположения вершин сети — равномерное.

В динамических случайных геометрических графах происходит перемещение вершин, а в соответствующей сети — перемещение узлов, в результате которого изменяются условия приема и передачи сигналов. При сближении двух узлов уровень сигнала становится достаточным для установления двусторонней связи, при удалении узлов друг от друга связь

теряется. Это происходит в случайные моменты времени  $t_1, t_2, \dots, t_i, \dots$ , поэтому последовательности  $G_0, G_1, G_2, \dots, G_i, \dots$  обыкновенных графов можно рассматривать как реализации случайного процесса. При указанных выше предположениях о распределениях вероятностей поток событий "изменение обыкновенного графа" является ординарным, две пары вершин не могут одновременно приобрести ребро или лишиться ребра, поэтому соседние графы в последовательности  $\{G_i\}$  отличаются в точности одним ребром.

Представляют практический интерес следующие постановки проблем для динамических геометрических графов при различных условиях:

1) ограничения среды  $R$ . Варианты ограничений:

- среда представляет собой односвязную выпуклую область (например, круг, прямоугольник); нет ограничений, перемещение возможно в любом направлении;
- среда представляет собой совокупность одномерных криволинейных пересекающихся отрезков; перемещение узлов по ребрам (или дугам) заданного графа — сети маршрутов на местности;
- среда представляет собой многосвязную область; перемещение в обход запретных или труднодоступных областей, для движения в которых нужно разрешение или специальное оборудование;

2) решение задачи  $Q$  распределенным алгоритмом  $A$ . В процессе решения задачи участвуют все узлы сети. Первоначально сеть  $N(t_0)$  связна, что позволяет начать решение задачи. В результате перемещения узлов в момент  $t_1$  сеть  $N(t_1)$  становится несвязной. Может быть несколько вариантов продолжения:

а) алгоритм  $A$  решил задачу  $Q$  к моменту времени  $t_1$ ;

б) алгоритм  $A$  не завершён к моменту времени  $t_1$  и продолжит свою работу в момент времени  $t_2$  (восстановления связности сети);

в) алгоритм  $A$  не завершён и может возобновить свою работу в момент  $t_2$  с частичной потерей выполненной ранее работы (с некоторой контрольной точкой);

г) алгоритм  $A$  не завершён, и его исполнение начинается заново в момент времени  $t_2$  восстановления связности сети ввиду неактуальности прежних исходных данных;

д) алгоритм  $A$  продолжает свою работу в промежутке времени от  $t_1$  до  $t_2$  в  $k$  компонентах сети, но в момент  $t_2$  должно происходить согласование частичных решений, на которое требуется время, зависящее от числа компонент.

Потеря связности сети приводит к одному из следующих последствий:

- невозможности решения задачи  $Q$  алгоритмом  $A$ ;
- снижению эффективности, задержкам решения задачи  $Q$  алгоритмом  $A$ ;
- снижению безопасности для сети  $N$  при решении задачи  $Q$  алгоритмом  $A$ ;
- вероятной потере отдельных узлов сети  $N$ , оказавшихся изолированными, и постепенному уничтожению сети  $N$  (например, при боевых действиях).

Для того чтобы полноценно использовать динамические графы в построении ad-hoc сети, вершины должны использовать некоторые алгоритмы маршрутизации, представленные в следующем разделе.

Рассмотрим особенности реализации алгоритмов маршрутизации в ad-hoc сетях. Существует большое число алгоритмов для классических проводных сетей. Поскольку беспроводные сети не имеют централизованной инфраструктуры, применение классических алгоритмов не является эффективным. Алгоритмы маршрутизации, применяемые в ad-hoc сетях, можно разделить по топологическим свойствам на топологические (учитываются связи между вычислительными узлами сети) и географические (используются географические координаты вычислительных узлов). Согласно принципам функционирования алгоритмы маршрутизации делятся на проактивные, реактивные и гибридные.

Если учитывать наилучшие маршруты, алгоритмы маршрутизации могут быть разделены на алгоритмы вектора расстояния (вектор расстояния, шагов) и алгоритмы со сложными метриками (используют оценку маршрутов по нескольким параметрам) [23].

Итак, существует несколько различных типов ad-hoc сетей, однако их особенности следует учитывать при разработке алгоритмов маршрутизации. Далее рассмотрим более подробно алгоритмы маршрутизации в оппортунистических сетях [21].

### Алгоритмы маршрутизации

Рассмотрим более подробно *алгоритмы маршрутизации в оппортунистических сетях без инфраструктуры*.

*Маршрутизация на основе рассылки* [29]. Методы маршрутизации на основе рассылки выполняют доставку сообщения по назначению, просто распространяя его по всей сети.

Эвристика, лежащая в основе этой политики, заключается в том, что поскольку нет сведений о возможном пути к месту назначения или сведений о соответствующем узле следующего перехода, сообщение следует отправлять повсюду. Оно в конечном итоге достигнет места назначения путем передачи от одного узла к другому. Методы, основанные на распространении, очень ресурсоемки. Кроме того, из-за значительного числа передач методы на основе распространения страдают от высокого уровня конкуренции и могут потенциально привести к перегрузке сети.

Чтобы увеличить пропускную способность сети, радиус распространения сообщения обычно ограничивают введением максимального числа ретрансляционных переходов для каждого сообщения или даже путем ограничения общего числа копий сообщений, присутствующих в сети в одно и то же время. Если ретрансляция дальше не разрешена, узел может отправлять сообщение только непосредственно получателю при встрече, если таковая происходит.

*Маршрутизация на основе сетевого кодирования*. Маршрутизация на основе сетевого кодирования [30] также относится к категории алгоритмов, основанных на распространении данных, но использует собственный подход для ограничения потока сообщений. Сообщения перед отправкой объединяются (кодируются) в узлах. Затем полученные коды отправляются вместо исходных сообщений. Коды распространяются в разных направлениях, как и в других протоколах маршрутизации, основанных на рассылке. Число генерируемых кодов выше, чем число оригинальных сообщений вместе взятых. Это необходимо, чтобы обеспечить большую устойчивость к потерям пакетов и путей. Кодирование выполняется как на исходном, так и на промежуточном узлах.

Далее рассмотрим *алгоритмы маршрутизации в оппортунистических сетях на базе инфраструктуры*.

*Маршрутизация на основе контекста*. В протоколе Context-Aware Routing (CAR) [31] каждый узел в сети отвечает за создание таких показателей, как вероятность доставки к каждому известному узлу назначения. Происходит периодический обмен вероятностями доставки, так что, в конечном счете, каждый узел может вычислить лучшего оператора для каждого узла назначения. Лучшие операторы вычисляются на основе контекста узлов. Среди атрибутов контекста, необходимых для выбора лучшего носителя, есть, например, остаточный уровень заряда батареи, скорость изменения связности, вероятность быть в пределах досягаемости места назначения, степень

мобильности. Когда лучший оператор получает сообщение для пересылки, он сохраняет его в локальном буфере и, в конечном счете, пересылает его на узел назначения, если таковой доступен, или же другому узлу с более высокой вероятностью доставки.

В *MobySpace Routing* [32] показатели мобильности узлов представляют контекстную информацию, используемую для маршрутизации. Протокол создает многомерное евклидово пространство по имени MobySpace, где каждая ось представляют собой возможный контакт между двумя узлами, а расстояние вдоль оси служит мерой вероятности этого контакта. Два узла, которые имеют похожие наборы контактов и вероятности связи с ними, близки в MobySpace. Лучшим узлом пересылки для сообщения является узел, который находится на минимальном расстоянии к месту назначения в этом пространстве. Это увеличивает вероятность того, что сообщение в конечном итоге достигнет пункта назначения. Очевидно, что в вышеописанном виртуальном контактном пространстве знание всех осей пространства также требует знания всех узлов, которые циркулируют в пространстве (в работе [30] предлагается оптимизация, которая не требует знания всех контактов между узлами).

*Алгоритм перемещения на основе кодов с избыточностью.* Общая идея — это кодирование с избыточностью [29]. Коды избыточности работают путем преобразования сообщения в больший набор кодовых блоков, так что любое достаточно большое подмножество генерируемых кодовых блоков может быть использовано для восстановления исходного сообщения.

Для более точной реализации алгоритма можно использовать агентное моделирование, иными словами, узлы сети будут являться интеллектуальными агентами, которые будут самостоятельно принимать решение о пересылках сообщений.

### **Система имитационного моделирования и автоматизированного проектирования TriadNS**

Система имитационного моделирования (СИМ) TRIadNS [5] была разработана на основе системы автоматизированного проектирования и моделирования TRIAD [6, 7].

Рассмотрим представление имитационной модели в TriadNS и представим лингвистические конструкции, предназначенные для описания компьютерных сетей. Имитационная модель в TriadNS представлена несколькими объектами, функционирующими в соответ-

ствии с некоторым сценарием и взаимодействующими друг с другом путем отправки сообщений. Вычислительные узлы сети и взаимосвязи между ними могут быть описаны на языке Triad [22] в слое структур. Следует отметить, что имитационная модель в Triad может быть представлена тремя слоями. Основным является слой структур. Структура компьютерной сети обычно представлена графом  $G(V, E)$ , где  $V$  — множество вершин (вычислительные узлы сети);  $E$  — ребра (дуги) между вершинами (линии связи между вычислительными узлами).

Поведение вычислительных узлов описывают в слое рутин. Третий уровень — это слой сообщений. Он предназначен для описания сообщений со сложными структурами данных.

Лингвистическая конструкция для описания структуры компьютерной сети представлена ниже:

*structure* <имя структуры> *def* (<список общих параметров>) (<список входных и выходных параметров>) <описание списка переменных> <оператор> *endstr*

Входные и выходные параметры связаны с входными и выходными полюсами объекта, представляющего компьютерный узел сети. Действительно, каждый компьютерный узел может отправлять сообщение  $s$  через выходные полюсы на другой узел и получать сообщения через входные полюсы. Каждый объект может быть описан (расшифрован) структурой нижнего уровня. Таким образом, имитационная модель является иерархической.

Вычислительные узлы компьютерной сети действуют по определенному сценарию. Сценарий описывают с помощью лингвистических и программных средств (процедур, которые называют еще "рутинами") слоя рутин. Каждая процедура задается множеством событий  $E$ , линейно упорядоченным множеством временных моментов ( $T$ ) и набором состояний  $\{Q\}$ . Состояние задается значениями локальных переменных. Локальные переменные определяются в рутине. Состояние изменяется только в случае возникновения события. События планируют друг друга (*schedule* (event1, t1) — процедура планирования события event1 на момент времени t1). В рутине могут быть определены входные и выходные полюсы ( $Pr_{in}$  и  $Pr_{out}$ ). Входной полюс служит для приема сообщений, выходные полюсы — для их отправки. Среди множества событий выделяют входные сообщения. Все входные полюсы обрабатываются входным событием, выходные полюсы — обычным событием. Специальный оператор

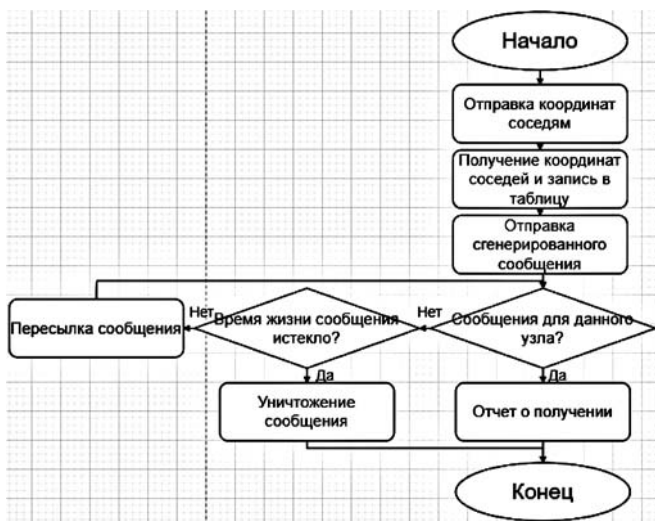


Рис. 1. Блок-схема алгоритма маршрутизации на основе рассылки

(*out* <сообщение> *through* <имя полюса>) служит для отправки сообщения.

На рис. 1 представлена блок-схема алгоритма маршрутизации на основе рассылки.

Сценарий имитационного эксперимента, критерии завершения имитационного запуска определяют в лингвистической конструкции "условия моделирования". Входными параметрами является список элементов имитационной модели (переменные, события, входные и выходные сообщения), которые необходимо изучить и обработать во время выполнения имитации с помощью информационных процедур. Список информационных процедур, которые используют для обработки элементов имитационной модели, включают в условия моделирования. Условия моделирования и информация о процедурах представляют собой "алгоритм исследования".

Система моделирования TriadNS включает также набор стандартных информационных процедур.

Кроме того, на языке Triad пользователь имеет возможность написать свой собственный алгоритм обработки элементов модели в ходе имитационного эксперимента.

Исследование слоя структур является только статическим процессом. Процесс моделирования может происходить только после определения поведения всех узлов модели в структурном слое. Прогон имитационного эксперимента инициализируется оператором *simulate*:

*Simulate* < список моделей > *on condition of simulation* < имя условия моделирования > (<параметры настроек>) (< параметры интерфейса >) (< список информационных процедур >; < последовательность операторов >).

Следует отметить, что одним оператором *simulate* можно запустить сразу несколько моделей.

Кроме того, можно выполнить моделирование с разными "условиями":

- предположим, что пользователь хочет разработать компьютерную сеть. Для этого необходимо определить структуру компьютерной сети и характеристики рабочих станций и других устройств компьютерной сети;
- предположим, что пользователь хочет определить конфигурацию компьютерной сети и технические характеристики самой дешевой сети, более эффективной сети, более надежной и т. д. Для этого необходимо определить разные критерии во время моделирования, и можно сделать это, используя различные "условия моделирования". Теперь продемонстрируем способность TriadNS для разработки и анализа алгоритма маршрутизации на основе рассылки.

### Реализация алгоритма маршрутизации оппортунистических сетей в TriadNS

Итак, для моделирования алгоритмов маршрутизации в оппортунистических компьютерных сетях была выбрана СИМ TriadNS. Напомним алгоритм маршрутизации, который мы будем исследовать.

Каждый узел, отправляя свое сообщение, рассылает его всем ближайшим узлам. Ближайшие узлы выполняют однотипные действия, они рассылает сообщения во все узлы, исключая исходный отправляющий узел и узел, из которого было отправлено сообщение (если это разные узлы). Таким образом, в определенный момент времени все копии сообщения должны быть у получателя. Если копии не доходят до получателя, они уничтожаются по истечении их времени жизни. Время жизни — число возможных пересылок сообщений. Время жизни указывается во входных параметрах и обязательно должно быть общим для всех узлов.

Представим результаты тестирования алгоритма.

Прежде всего, необходимо определить входные параметры: координаты конкретного узла, идентификатор этого узла, радиус видимости, время жизни сообщения.

Рассмотрим тест № 1 (рис. 2). Пусть в сети в передаче сообщений участвуют четыре узла, радиус видимости которых позволяет им достичь только двух других узлов. Время жизни сообщения в этом случае — только одна пересылка.

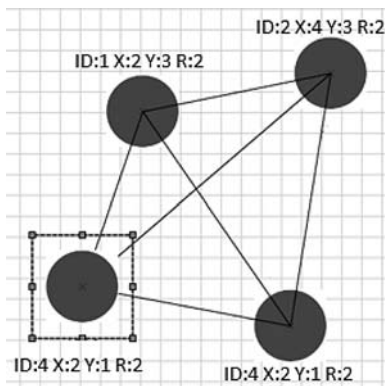


Рис. 2. Тест № 1. Отслеживание сообщений при выполнении алгоритма маршрутизации на основе рассылки. Число узлов — 4

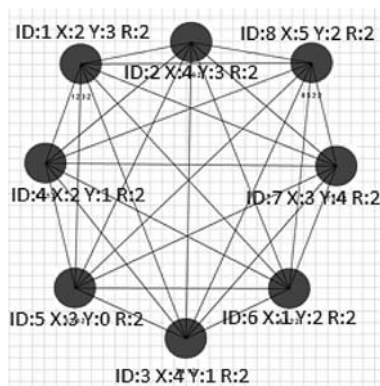


Рис. 3. Тест № 2, 3. Отслеживание сообщений при выполнении алгоритма маршрутизации на основе рассылки. Число узлов — 8

Результат теста № 1 были зафиксированы в журнале. Анализ записей в журнале показал, что многие сообщения являются "мертвыми", только небольшая часть сообщений была доставлена в нужное место.

Тест № 2 является более сложным: имеется восемь узлов, и каждый узел видит три других узла. Срок службы сообщения остается таким же, как в предыдущем примере — одна пересылка (рис. 3).

Анализ результатов эксперимента показал, что увеличился разрыв между отправкой сообщения и его прибытием, а также увеличилось число уничтоженных сообщений.

В тесте № 3 число узлов было таким же, что и во втором тесте, но время жизни сообщений было увеличено. В этом случае время жизни равно времени двух пересылок. Результат эксперимента свидетельствует о том, что число сообщений, отправленных с одного узла на другой, увеличивается при увеличении срока службы сообщений.

На основе трех испытаний можно сделать вывод, что при увеличении числа узлов следует увеличить жизненный цикл сообщения, чтобы сообщения достигли цели с высокой степенью надежности.

## Заключение

В статье кратко представлены компьютерные сети нового поколения, а именно самоорганизующиеся сети (SON), и несколько алгоритмов, предназначенных для реализации передачи данных в самоорганизующихся сетях.

В качестве модели самоорганизующихся сетей используют динамические геометрические графы; в самоорганизующихся сетях геогра-

фические координаты вычислительных узлов могут изменяться.

Для исследования эффективности алгоритмов маршрутизации авторы предлагают использовать разработанное ими программное обеспечение — симулятор компьютерных сетей TriadNS и высокоуровневый язык Triad.

Симулятор компьютерных сетей имеет ряд особенностей, которые позволяют строить сложные структуры сетей, задавая графовые константы (соответствуют основным топологиям компьютерных сетей) и выполняя операции над структурой модели (добавление вершины или ребра, удаление вершины или ребра, объединение графов, пересечение графов и т.д.). Алгоритм построения структуры сети записывают с помощью параметризованной процедуры.

Использование этой процедуры позволяет сгенерировать соответствующую структуру сети с произвольным числом узлов.

Наряду с лингвистическими средствами описания структур исследователь имеет возможность описать функционирование вычислительных узлов, разработать сценарий проводимых исследований, используя информационные процедуры и условия моделирования (информационные процедуры собирают и обрабатывают необходимые данные, а условия моделирования задают критерии проведения эксперимента).

Поскольку ad-hoc сети в настоящее время находят широкое применения, а разработка алгоритмов маршрутизации является достаточно нетривиальной задачей, предлагаемые авторами программные средства и проводимые исследования являются актуальными.

## Список литературы

1. **COMNETIII**. Planning for Network Managers. Release 1.3. URL: <http://eent3.lsbu.ac.uk/staff/baoyb/acs/Comnet/comnet%20III.pdf> (Accessed: 10.06.2018).
2. **OMNeT++** Community Site. URL: <http://www.omnetpp.org> (Accessed: 10.06.18).
3. **OPNET**. URL: <http://www.riverbed.com/products/performance-management-control/opnet.html?redirect=opnet> (Accessed: 10.06.18) Zhang Y.
4. **ANYLOGIC**. URL: [www.anylogic.ru](http://www.anylogic.ru). (Accessed: 10.06.18).
5. **Mikov A., Zamiatina E.** Program Tools and Language for Networks Simulation and Analysis // Proceedings SDN & NFV — The Next Generation of Computational Infrastructure: 2014 International Science and Technology Conference "Modern Networking Technologies (MoNeTec)". October 27–29. 2014. P. 94–102.
6. **Zamyatina E., Mikov A., Mikheev R.** Linguistic and Program Tools For Debugging and Testing Of Simulation Models Of Computer Networks // International Journal "Information Models and Analyses". 2013. Vol. 2, N. 1. P. 70–80.

7. **Mikov A. I.** Simulation and Design of Hardware and Software with Triad // Proc.2nd Intl. Conf. on Electronic Hardware Description Languages. Las Vegas, USA. 1995. P. 15–20.
8. **Миков А. И.** Автоматизация синтеза микропроцессорных управляющих систем. Иркутск: Издательство Иркут. ун-та, 1987.
9. **Проскочило А. В., Воробьев А. В., Зряхов М. С., Кравчук А. С.** Анализ состояния и перспективы развития самоорганизующихся сетей // Научные ведомости. Сер. "Экономика. Информатика. 2015, № 19(216). Вып. 36/1. С. 177–187.
10. **Белфер Р. А.** Сети SON. Материал из Национальной библиотеки им. Баумана, 2012. URL: [https://ru.bmstu.wiki/Seti\\_SON](https://ru.bmstu.wiki/Seti_SON). (Accessed: 10.06.18).
11. **Oubbatia O. S., Lakasb A., Zhouc F., Güne M., Yagoubia M. B.** A survey on position-based routing protocols for Flying Ad-hoc Networks (FANETs) // Vehicular Communications. 2017. V. 10. P. 29–56.
12. **Qiu T., Chen N., Li K., Qiao D., Fu Z.** Heterogeneous ad hoc networks: Architectures, advances and challenges // Elsevier Ad Hoc Networks. 2017. Vol. 55. P. 143–152.
13. **Kucheryavy A.** State of the Art and Research Challenges for USN Traffic Flow Models // 16th International Conference on Advanced Communication Technology (ICACT). 2014. P. 336–340.
14. **Чинь Б. Х., Парамонов А. И.** Исследование параметров маршрутов в самоорганизующихся сетях связи // Информационные технологии и телекоммуникации. 2016. С. 81–88.
15. **Li M., Salmanian M., Willink T. J.** Network layer connectivity awareness with application to investigate the OLSR protocol in tactical MANETs // Proceedings of the 2017 Winter Simulation Conference. P. 4093–4105.
16. **Liu X., Li Z., Yang P., Dong Y.** Information-centric mobile ad hoc networks and content routing: A survey // Ad Hoc Networks. 2017. Vol. 58. P. 255–268.
17. **Ali H. A., Areed M. F., Elewely D.** An on-demand power and load-aware multi-path node-disjoint source routing scheme implementation using NS-2 for mobile ad-hoc networks // Simulation Modelling Practice and Theory. 2018. Vol. 80. P. 50–65.
18. **Boussoufa-Lahlaha S., Semchedinea F., Bouallouche-Medjkounea L.** Geographic routing protocols for Vehicular Ad hoc NETWORKS (VANETs): A survey // Vehicular Communications. 2018. Vol. 11. P. 20–31.
19. **Kitsisa R., Dattaa S.** Layer 3 Enhancements for Vehicular Ad Hoc Networks // Proceedings of the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018). Elsevier. Procedia Computer Science. 2018. Vol. 130. P. 628–635.
20. **Dharani Kumari N. V., Shylaja B. S.** AMGRP: AHP-based Multimetric Geographical Routing Protocol for Urban environment of VANETs // Journal of King Saud University – Computer and Information Sciences. 2017. URL: <http://dx.doi.org/10.1016/j.jksuci.2017.01.001>.
21. **Григорьев А. А.** Алгоритмы маршрутизации в оппортунистических сетях // Труды МФТИ. 2013. Т. 5, № 3. С. 112–120.
22. **Mikov A. I.** Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design. Mexico. 1995. P. 105–127.
23. **Винокуров В. М., Пуговкин А. В., Пшенников А. А., Ушарова Д. Н., Филатов А. С.** Маршрутизация в беспроводных сетях // Труды ТУСУР. № 2(22). 2010. С. 288–292.
24. **Леонов А. В., Чаплышкин В. А.** Сети FANET // Омский научный вестник. 2015. № 3 (143). С. 297–301.
25. **Павлов А. А., Датъев И. О., Шишаев М. Г.** Разработка имитационных моделей для тестирования протоколов маршрутизации беспроводных многошаговых сетей // Вестник Иркутского государственного технического университета. 2016. № 7. С. 90–101. DOI: 10.21.285/1814-3520-2016-7-90-101.
26. **Щерба Е. В., Никонов В. И., Литвинов Г. А.** Обеспечение безопасности протоколов маршрутизации // Доклады ТУСУР. 2018. Т. 21, № 3. С. 19–70.
27. **Миков А. И.** Связность автономных беспроводных компьютерных сетей в местностях с плохой инфраструктурой // Экологический вестник научных центров. 2015. С. 70–75.
28. **Миков А. И.** Динамические геометрические графы мобильных ad hoc сетей // Информатизация и связь. 2017. № 2. С. 66–70.
29. **Vahdat A., Becker D.** Epidemic routing for partially connected ad hoc networks: Tech. Rep. CS-2000-06. Department of Computer Science, Duke University. Durham, NC, 2000.
30. **Musolesi M., Hailes S., Mascolo C.** Adaptive Routing in Intermittently Connected Mobile Ad Hoc Networks // Proceedings of the 6th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005). Naxos, 2005. P. 183–189.
31. **Leguay J., Friedman T., Conan V.** Evaluating Mobility Pattern Space Routing for DTNs // Proceedings of the IEEE Infocom 2006. Barcelona, 2006. P. 1–10.
32. **Widmer J., Le Boudec J.-Y.** Network Coding for Efficient Communication in Extreme Networks // Proceedings of ACM SIGCOMM 2005 Workshop on delay tolerant networks. Philadelphia, 2005. P. 284–291.

**A. I. Mikov**, Full Professor, Professor, e-mail: [Alexander\\_Mikov@mail.ru](mailto:Alexander_Mikov@mail.ru),

Computer Technologies Department, Kuban State University, Krasnodar, Russian Federation,

**E. B. Zamyatina**, Ph.D., Associate Professor, e-mail: [ezamyatina@hse.ru](mailto:ezamyatina@hse.ru),

Business Informatics Department, National Research University Higher School of Economics, Perm, Russian Federation,

**S. M. Kalashnikov**, Graduate Student, e-mail: [ser.00@list.ru](mailto:ser.00@list.ru),

Computer Science Department, Perm State National Research University, Perm, Russian Federation

## Research of Routing Algorithms in SON Networks Using Simulation Predictive Modeling Software

*The paper presents the results of a study of one of the routing algorithms in ad-hoc networks, namely, in SON (self-organizing) networks, by the method of simulation. Some kinds of ad hoc networks and features of the implementation of routing algorithms in these networks are considered. Attention is focused on simulation modeling tools, the requirements for software of this kind are indicated, and the constructions of the simulation language Triad are given. The simulation is performed in the TriadNS tool environment developed by the authors.*

**Keywords:** ad hoc networks, simulation modeling, simulation systems, dynamic graphs, routing algorithms

**Acknowledgements:** This work was supported by the Russian Foundation for Basic Research, project no. 18-01-00359.

DOI: 10.17587/it.26.302-310

### References

1. **COMNET III.** Planning for Network Managers. Release 1.3, available at: <http://eent3.lsbu.ac.uk/staff/baoyb/acs/Comnet/comnet%20III.pdf> (date of access: 10.02.2020).
2. **OMNeT++** Community Site, available at: <http://www.omnetpp.org> (date of access: 10.02.20).
3. **OPNET**, available at: <http://www.riverbed.com/products/performance-management-control/opnet.html?redirect=opnet> (date of access: 10.06.18).
4. **ANYLOGIC**, available at: [www.anylogic.ru](http://www.anylogic.ru) (date of access: 10.06.18).
5. **Mikov A., Zamiatina E.** Program Tools and Language for Networks Simulation and Analysis, *Proceedings SDN & NFV – The Next Generation of Computational Infrastructure: 2014 International Science and Technology Conference "Modern Networking Technologies (MoNeTec)"*, October 27–29, 2014, Lomonosov Moscow State University, pp. 94–102.
6. **Zamyatina E., Mikov A., Mikheev R.** Linguistic and Program Tools For Debugging and Testing Of Simulation Models Of Computer Networks, *International Journal "Information Models and Analyses"*, 2013, vol. 2, no. 1, pp. 70–80.
7. **Mikov A. I.** Simulation and Design of Hardware and Software with Triad, *Proc. 2nd Intl. Conf. on Electronic Hardware Description Languages*, Las Vegas, USA, 1995, pp. 15–20.
8. **Mikov A. I.** Avtomatizatsiya sinteza mikroprocessornyh upravlyaushih system, Irkutsk, Publishing house of Irkut. un-t, 1987 (in Russian).
9. **Proskochylo A. V., Vorobyov A. V., Zriakhov M. S., Kravchuk A. S.** Analysis of state and development perspectives of self-organizing networks, *Nauchnye vedomosti, Seriya Ekonomika. Informatika*, 2015, no. 19 (216), iss. 36/1, pp. 177–187 (in Russian).
10. **Belfer R. A.** Seti SON, Material is Nacionalnoy biblioteki im. N. E. Baumana, 2012, available at: [https://ru.bmstu.wiki/Seti\\_SON](https://ru.bmstu.wiki/Seti_SON) (date of access: 10.06.18) (in Russian).
11. **Oubbatia O. S., Lakas A., Zhouc F., Güne M., Yagoubia M. B.** A survey on position-based routing protocols for Flying Adhoc Networks (FANETs), *Vehicular Communications*, 2017, vol. 10, pp. 29–56.
12. **Qiu T., Chen N., Li K., Qiao D., Fu Z.** Heterogeneous ad hoc networks: Architectures, advances and challenges, *Elsevier Ad Hoc Networks*, 2017, vol. 55, pp. 143–152.
13. **Koucheryavyi A.** State of the Art and Research Challenges for USN Traffic Flow Models, *16th International Conference on Advanced Communication Technology (ICACT)*, 2014, pp. 336–340.
14. **Trinh B., Paramonov A.** Research of parameters of routes in self-organizing networks, *Information technologies and telecommunication*, 2016, vol. 4, no. 4, pp. 81–88 (in Russian).
15. **Li M., Salmanian M., Willink T. J.** Network layer connectivity awareness with application to investigate the OLSR protocol in tactical MANETs, *Proceedings of the 2017 Winter Simulation Conference*, pp. 4093–4105.
16. **Liu X., Li Z., Yang P., Dong Y.** Information-centric mobile ad hoc networks and content routing: A survey, *Ad Hoc Networks*, 2017, vol. 58, pp. 255–268.
17. **Ali H. A., Areed M. F., Elewely D.** An on-demand power and load-aware multi-path node-disjoint source routing scheme implementation using NS-2 for mobile ad-hoc networks, *Simulation Modelling Practice and Theory*, vol. 80, 2018, pp. 50–65.
18. **Boussoufa-Lahlaha S., Semchedinea F., Bouallouche-Medjkoune L.** Geographic routing protocols for Vehicular Ad hoc Networks (VANETs): A survey, *Vehicular Communications*, 2018, vol. 11, pp. 20–31.
19. **Kitsisa R., Dattaa S.** Layer 3 Enhancements for Vehicular Ad Hoc Networks, *Proceedings of the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018), Procedia Computer Science*, 2018, vol. 130, pp. 628–635.
20. **Dharani Kumari N. V., Shylaja B. S.** AMGRP: AHP-based Multimetric Geographical Routing Protocol for Urban environment of VANETs, *Journal of King Saud University – Computer and Information Sciences*, 2017, available at: <http://dx.doi.org/10.1016/j.jksuci.2017.01.001>.
21. **Grigoriev A. A.** *Trudy MFTI*, 2013, vol. 5, no. 3, pp. 112–120 (in Russian).
22. **Mikov A. I.** Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design. Mexico. 1995. P. 105–127.
23. **Vinokurov V. M., Pugovkin A. V., Pshennikov A. A., Usharova D. N., Filatov A. S.** *Trudy TUSUR*, 2010, no. 2 (22), pp. 288–292 (in Russian).
24. **Leonov A. V., Chaplyshkin V. A.** *Omskiyi nauchnyi vestnik*, 2015, no. 3 (143), pp. 297–301 (in Russian).
25. **Pavlov A. A., Datiev I. O., Shishaev M. G.** *Vestnik Irkutskogo gosudarstvennogo universiteta*, 2016, no. 7, pp. 90–101, DOI: 10.21285/1814-3520-2016-7-90-101 (in Russian).
26. **Sherba E. V., Nikonov V. I., Litvinov G. A.** *Doclady TUSUR*, 2018, vol. 21, no. 3, pp. 19–70 (in Russian).
27. **Mikov A. I.** *Ekologicheskii vestnik nauchnykh centrov CHES*, 2015, pp. 70–75 (in Russian).
28. **Mikov A. I.** *Informatizacia i Svyaz*, 2017, no. 2, pp. 66–70 (in Russian).
29. **Vahdat A., Becker D.** Epidemic routing for partially connected ad hoc networks: Tech. Rep. CS-2000-06, Department of Computer Science, Duke University, Durham, NC, 2000.
30. **Musolesi M., Hailes S., Mascolo C.** Adaptive Routing in Intermittently Connected Mobile Ad Hoc Networks, *Proceedings of the 6th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005)*, Naxos, 2005, pp. 283–289.
31. **Leguay J., Friedman T., Conan V.** Evaluating Mobility Pattern Space Routing for DTNs, *Proceedings of the IEEE Infocom 2006*, Barcelona, 2006, pp. 1–10.
32. **Widmer J., Boudec J.-Y. L.** Network Coding for Efficient Communication in Extreme Networks, *Proceedings of ACM SIGCOMM 2005 Workshop on delay tolerant networks*, Philadelphia, 2005, pp. 284–291.