

## ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ СОЦИАЛЬНЫХ СЕТЕЙ

**И.В. Дмитриев, Е.Б. Замятина (Пермь)**

В настоящее время виртуальные социальные сети находят все более широкое распространение. Виртуальная социальная сеть – это специальный Интернет - ресурс, позволяющий ее участникам, вне зависимости от их текущего местоположения, общаться со своими родственниками, коллегами и друзьями, обмениваться с ними различной информацией, а также искать интересующие данные.

На сегодняшний день насчитывается более миллиарда пользователей социальных сетей. И если ранее социальные сети использовались в основном для общения между людьми, то в настоящее время они используются различными компаниями для решения рабочих вопросов, для работы с клиентами, для поиска информации, для распространения рекламы.

Структура виртуальных социальных сетей сходна со структурой социальных сетей в реальном мире. Таким образом, изучение социальных сетей позволяет изучить принципы распространения информации, организации групп пользователей, способы привлечения клиентов.

Авторы работы [1] выделяют два основных подхода к анализу социальных сетей: статический и динамический. Первый подход подразумевает изучение структуры сети (топологии), ее основных свойств (смежность, степень центральности, расстояние и др.). В рамках данного подхода делается «снимок» социальной сети и анализируется ее текущее состояние.

Основное внимание уделяется геометрической форме сети, а также различным связям между вершинами (участниками социальной сети). Статический (структурный) подход позволяет достаточно точно характеризовать текущее состояние системы, однако не позволяет увидеть многие закономерности, которые становятся заметны только при изучении развития структуры сети.

Рассмотрение социальной сети в динамике дает возможность проследить различные этапы ее формирования, выделить основные закономерности образования связей между вершинами и ход образования кластеров в графе.

Часто исследования проводятся с интернет - графами, свойства которых соответствуют свойствам реальных социальных сетей. Однако и реальные социальные интернет-сети привлекают внимание исследователей.

### **Требования к программным средствам моделирования социальных сетей**

К программным средствам симуляторов социальных сетей можно предъявить следующие требования:

– Симулятор должен располагать программными средствами построения интернет-графов. При их построении необходимо соблюдать соответствие свойств генерируемого графа свойствам реальных социальных сетей. Так, показатели распределения степеней вершин генерируемого моделью графа должны быть близки к экспериментальным. Например, показатели распределения степеней вершин многих социальных сетей меньше 2. Еще одним критерием является гибкость программных средств, которые позволяют оперативно менять параметры моделей интернет-графов.

– Симулятор должен располагать программными средствами исследования интернет-графов.

– Симулятор должен уметь имитировать поведение пользователей. Пользователи социальной сети играют определенные роли и взаимодействуют друг с другом (передача

сообщений, публикация ссылок и т.д.). Для имитации поведения пользователей более всего подходит агентная парадигма имитационного моделирования, часто программное обеспечение.

– Симулятор должен уметь работать с большими объемами данных, т.е. должен обладать программными средствами, позволяющими использовать вычислительные мощности нескольких вычислительных узлов или процессоров.

Рассмотрим, какими свойствами из перечисленных выше обладает симулятор компьютерных сетей TriadNS, а именно, укажем на особенности представления имитационной модели.

### Представление имитационной модели в TRIADNS

Симулятор компьютерных TriadNS был разработан на основе системы автоматизированного проектирования и моделирования Triad, которая была спроектирована и реализована сотрудниками кафедры математического обеспечения ВС ПГНИУ в 80-е годы прошлого века. Система Triad предназначалась для проектирования и моделирования вычислительных систем. В 2002 году работы были возобновлены (Triad.Net, язык разработки C#) и продолжают по сей день.

В TriadNS принято трехуровневое представление имитационной модели:  $M = (STR, ROUT, MES)$ , где STR – слой структур, ROUT – слой рутин, MES – слой сообщений.

Слой структур представляет собой совокупность объектов, взаимодействующих друг с другом посредством посылки сообщений. Каждый объект имеет полюсы (входные и выходные), которые служат соответственно для приёма и передачи сообщений. Основа представления слоя структур – графы. В качестве вершин графа следует рассматривать отдельные объекты (концентраторы, маршрутизаторы, серверы, рабочие станции). Дуги графа определяют связи между объектами.

Имитационная модель имеет иерархическое представление. Отдельные объекты, представляющие вершины графа, могут быть расшифрованы подграфом более низкого уровня и т.д.

Объекты действуют по определённому сценарию, который описывают с помощью рутины. Рутинка представляет собой последовательность событий  $e_i$ , планирующих друг друга ( $E$  – множество событий; множество событий рутины является частично упорядоченным в модельном времени). Выполнение события сопровождается изменением состояния объекта. Состояние объекта определяется значениями переменных рутины. Таким образом, система имитации является событийно-ориентированной.

Рутинка так же, как и объект, имеет входные и выходные полюса. Входные полюса служат соответственно для приёма сообщений, выходные полюса – для их передачи. В множестве событий рутины выделено входное событие  $ein$ . Все сообщения, которые поступают на входные полюса рутины, обрабатываются входным событием. Обработка сообщений, которые генерируются на выходных полюсах рутины, осуществляется обычными событиями рутины. Для передачи сообщения служит специальный оператор **out** (**out** <сообщение> **through** <имя полюса>). Совокупность рутин определяет слой рутин ROUT.

Слой сообщений (MES) предназначен для описания сообщений сложной структуры.

Система моделирования Triad реализована таким образом, что пользователю необязательно описывать все слои. Так, если возникает необходимость в исследовании структурных особенностей модели, то можно описать только слой структур.

Triad-модель рассматривается как переменная. Она может быть построена с помощью операций над моделью.

Как уже было сказано ранее, модель включает описание структуры, рутин и слоя сообщений.

Синтаксис слоя структур: `structure <имя структуры> def (<список настроечных параметров>) (<список входных и выходных параметров>) <описание переменных> <операторы> endstr`

В качестве переменных можно использовать переменные типа «вершина», «граф», «полюс» и т.д. Операторы выполняют операции над вершинами, графами, полюсами, структурами.

Ниже приведен пример описания структуры компьютерной сети, состоящей из сервера и нескольких клиентов (в слое структур).

```
structure КлиентСервер[integer числоКлиентов ] def
КлиентСервер := node Сервер<ПРИЕМ,ВЫДАЧА> +
node Клиент[0:числоКлиентов-1] <ПРИЕМ,ВЫДАЧА>;
integer i;
for i := 0 by 1 to числоКлиентов - 1 do
КлиентСервер := КлиентСервер +
arc( Клиент[ i ].ВЫДАЧА -- Сервер.ПРИЕМ )
+arc(Сервер.ВЫДАЧА--Клиент[ i ].ПРИЕМ );
endf;
endstr
```

Выше представлена структура сети «КлиентСервер», которая строится в виде вершины «Сервер» с присоединенным к ней массивом вершин «Клиент».

Связи между вершинами устанавливаются в цикле `for` с помощью дуг, при этом указываются входные и выходные полюса (`arc(Сервер.ВЫДАЧА--Клиент[i].ПРИЕМ)`).

Слой структур представляет собой параметризованную процедуру. Изменив значение входного параметра «числоКлиентов», можно получить в результате модель со структурой, в которой определено уже другое количество клиентов.

Переопределить количество клиентов можно перед началом или в ходе имитационного эксперимента. Во втором случае переопределение выполняется в условии моделирования.

Сценарий работы клиента описывают с помощью рутины, синтаксис и текст которой приведены ниже:

Синтаксис:

```
routine<имя>(<список настроечных параметров>)(<список входных и выходных
формальных параметров>) initial <последовательность операторов> endi
event <последовательность операторов> ende
event <наименование события> <последовательность операторов> ende ...
event<наименование события><последовательность операторов> ende endrout
```

Пример рутины:

```
routine Клиент (input ПРИЕМ; output ВЫДАЧА ) [ real deltaT ]
initial
boolean запросПослан;
запросПослан := false;
schedule ЗАПРОС in 0;
Print «Инициализация клиента»; endi
event ЗАПРОС;
out «Запрос на обслуживание» through ВЫДАЧА;
Print «Клиент послал запрос серверу»;
schedule ЗАПРОС in deltaT;
ende
endrout
```

Итак, слой структур в TriadNS позволяет построить модель Интернет - графа, вершины – узлы социальной сети, ребра указывают на связи между узлами.

### Случайные графы

В настоящее время для построения различных моделей социальных сетей применяют теорию случайных графов. Существует много видов моделей, генерирующих случайные графы, близкие по свойствам к реальным сетям [5]. Их можно разделить на несколько основных классов:

- модели случайных графов (модель Эрдёша-Реньи);
- простейшие модели безмасштабных сетей (модель Боллобаша-Риордана, модель копирования и др.);
- более гибкие модели безмасштабных сетей (модель Чунг-Лу, модель Янсона-Лучака);
- модель стохастических графов Кронекера.

Рассмотрим более подробно некоторые из моделей Интернет - графов. Первой обсуждаемой моделью является модель Эрдёша-Реньи.

Модель Эрдёша-Реньи на данный момент является самой изученной моделью случайных графов [6]. Но в начале 2000-х выяснилось, что она плохо подходит для описания графов, возникающих в реальных социальных сетях.

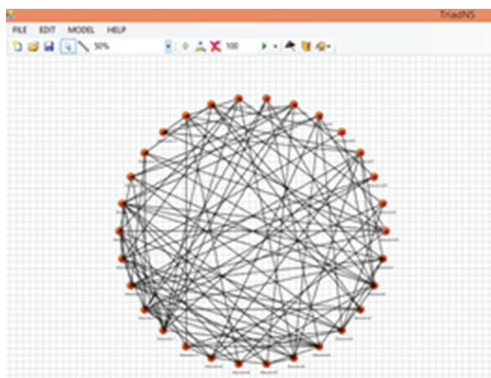


Рис. 1. Случайный граф модели Эрдеши-Реньи с 30 вершинами и  $p=0,25$

В настоящее время в симуляторе компьютерных сетей TriadNS возможно построение такого случайного графа. Для построения графа выбирают соответствующий пункт меню в графическом редакторе TriadNS, задают соответствующие параметры, а именно, количество вершин в графе и вероятность, с которой будет проведено ребро между любыми двумя вершинами, и запускают соответствующую процедуру.

Пример модели Эрдёша-Реньи для случайного графа из 30 вершин с вероятностями 0,25 проведения ребер между любыми двумя вершинами приведен на рис. 1. При вероятности, равной 1, получаем полный граф (рис. 2).

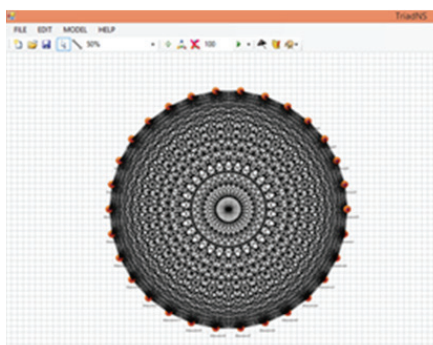


Рис. 2. Случайный граф модели Эрдеши-Реньи с 30 вершинами и  $p=1$

Полный граф ( $p=1$ ) может быть построен с помощью графовой константы  $\text{comple}(n)$ , где  $n$  - количество вершин. Используя графовые константы и операции над структурой: добавление вершин, добавление ребер, объединение и пересечение графов и т.д., – можно строить модели случайных графов.

Следующей обсуждаемой моделью является модель графа Барабаши – Альберта. Введем прежде всего понятие Интернет - графа [5] (или веб - графа). Интернет - граф - это граф, вершинами которого являются какие-либо конкретные структурные единицы в Интернете: страницы, сайты, хосты.

Обычно принято считать вершинами веб-графа сайты, а ребрами соединять те вершины, между которыми есть ссылки. При этом число ребер между вершинами равно числу ссылок между соответствующими сайтами, возможны ссылки сайта на себя (т.е. петли), ребра разумно полагать направленными.

Таким образом, Интернет - граф является ориентированным, он имеет кратные ребра и петли [6].

Интернет - графы обладают следующими свойствами:

1. Интернет - графы формируются добавлением к нему новых вершин, соединяемых ребрами со старыми вершинами.

2. Диаметр Интернет - графа мал (примерно 5–7). Это свойство соответствует известному свойству любой социальной сети, поскольку известно, что «мир тесен».

3. Интернет - граф имеет степенное распределение вершин. Этот факт роднит Интернет со многими реальными сетями – социальными, биологическими и транспортными. Они также подчиняются степенному распределению, только имеют другой показатель распределения  $\lambda$ . Графы со степенным распределением степеней вершин называют *безмасштабными*. Исходя из своих наблюдений, Барабаши и Альберт ввели понятие *предпочтительного присоединения* [8].

Модель графа Барабаши – Альберта (граф БА) представляет собой алгоритм генерации случайных безмасштабных сетей с использованием правила предпочтительного связывания (ПС) [11]. Правило предпочтительного связывания говорит, что, чем большую степень связности имеет вершина, тем выше вероятность присоединения к ней новых вершин. Если для присоединения выбирать вершину случайным образом, то вероятность выбора определённой вершины будет пропорциональна её степени связности [9]. Правило соответствует принципу «богатый становится богаче». Данный граф строится из графа, у которого степень связности каждой вершины должны быть не меньше единицы. Каждая новая вершина присоединяется к уже существующим вершинам с вероятностью, пропорциональной степени связности этих вершин. В качестве такого графа берется полный граф из 3 вершин. Пользователь вводит количество вершин и количество ребер, которые будут прибавляться при создании новой вершины.

Каждая новая вершина соединяется с существующими вершинами с вероятностью, пропорциональной числу связей этих вершин [10].

Наиболее связанные узлы («хабы»), как правило, накапливают ещё больше связей, тогда как узлы с небольшим числом связей вряд ли будут выбраны для присоединения новых узлов. Новые узлы имеют «предпочтение» соединяться с наиболее связанными узлами.

Пример построения графа с 30 вершинами и добавлением 3 вершин на каждом шаге приведен на рис.3.

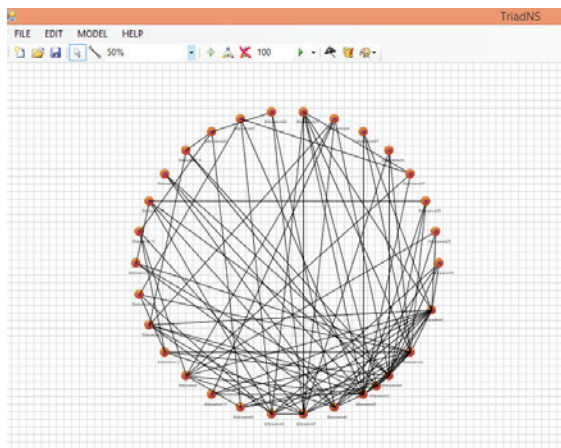


Рис. 3. Модель графа Барабаши-Альберта

Итак, в настоящее время разработаны процедуры для построения ряда моделей социальных сетей (класс случайных графов, класс безмасштабных сетей) [12]. Процедуры являются параметризованными.

### Программные средства исследования графа

Социальные сети могут характеризоваться множеством разных метрик, ниже приведены некоторые из них:

1. Взаимная направленность – свойство показывает, является ли отношение между вершинами бинарным (является ли связь двунаправленной).
2. Гомогенность – указывает на степень появления связей между похожими акторами (по полу, возрасту, интересам) [11].
3. Транзитивность связей – увеличение вероятности появления связей между акторами, у которых есть связи с одними и теми же вершинами [13]. Разница в распределении указывает на большое количество связей у одних узлов и минимальное у других. Важным в данном случае является феномен «богатый становится богаче», который приводит к высокой дисперсии вершин.
4. Центральность – метрика, которая позволяет определить значительность или влияние определенного узла или группы в сети [14].
5. Ассортативность – склонность к образованию связей между вершинами большой степени [15].
6. Диаметр.

Для определения характеристик графа в TriadNS существует большое количество стандартных процедур. На рис. 4 представлено окно, в котором следует указать те характеристики, которые интересуют пользователя.

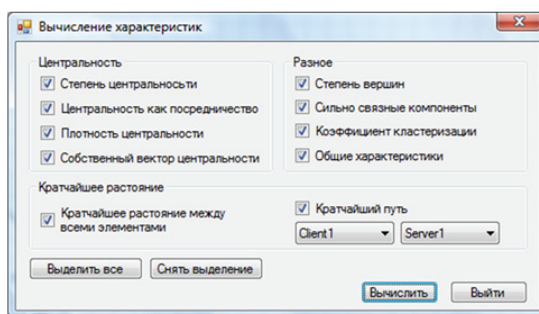


Рис. 4. Окно «Вычисление характеристик графа»

### Заключение

В работе представлены программные и языковые средства симулятора компьютерных сетей TriadNS, которые позволяют строить случайные и Интернет-графы с целью исследования социальных сетей.

Программные средства TriadNS позволяют получить характеристики Интернет-графов.

Наряду с возможностями исследований характеристик графа возможно динамическое исследование социальных сетей.

Динамическое исследование социальных сетей в настоящей статье подробно не рассматриваются. Однако представленные языковые и программные средства TriadNS указывают на то, что TriadNS может успешно применяться для этого (слой рутин, информационные процедуры для сбора статистики во время имитационного эксперимента).

TriadNS дает возможность выполнять распределенное (параллельное) моделирование[16]. Как известно, исследование социальных сетей может быть связано с обработкой больших объемов данных.

### Литература

1. **Gatti M, Appel A.P., Nogueira dos Santos C. Pinhanez C.Z.** A simulation-based approach to analyze the information diffusion in Microblogging Online Social Network // Proceedings of Winter Simulation Conf., ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, Piscataway, New Jersey, 8-11 Dec. 2013; IEEE. P. 1685–1696.
2. **Замятина Е.Б., Миков А.И., Михеев Р.А.** Лингвистические и интеллектуальные инструментальные средства симулятора компьютерных сетей TRIADNS // International Journal “Information theories & Applications (IJ ITA). 2012. Vol. 19. Number 4. P.355–368. ITNEA, Sofia, 1000, P.O.B. 775, Bulgaria.
3. **Замятина Е.Б., Миков А.И.** Программные средства системы имитации Triad.Net для обеспечения ее адаптируемости и открытости // Информатизация и связь. 2012. № 5. АНО «Редакция журнала «Информатизация и связь». С. 130–133.
4. **Mikov A.I.** Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Gero J.S. and F.Sudweeks F.(eds), Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design, Mexico, Mexico, 1995. P. 105–127.
5. **Райгородский А.М.** Математические модели интернета. // Квант/ 2012/ № 4. URL: <http://elementy.ru> (Проверено: 15 октября 2015).
6. **Райгородский А. М.** Модели случайных графов и их применение // Труды МФТИ. 2010. Т. 2. №4. С. 130-140.
7. **Mislove A.E., Druschel P.** Online social networks: measurement, analysis, and applications to distributed information systems. Houston, Texas, USA: Rice University, 2009. 244 p.
8. **Barabási L.-A., Albert R., Jeong H.** Scale-free characteristics of random networks: the topology of the world-wide web // Physica. 2000. Vol. A281. P. 69–77.
9. **L.-A. Barabási, R. Albert, H. Jeong.** Diameter of the world-wide web // Nature. 1999. Vol. 401. P. 130–131.
10. **Barabási L.-A., Albert R.** Emergence of scaling in random networks // Science. 1999. Vol. 286. P. 509–512.
11. **Aiello W., Chung F., Lu L.** A Random Graph Model for Massive Graphs // Experimental Mathematics. 2001. Vol. 10. P. 53–56.

12. **Берновский М.М., Кузюрин Н.Н.** Случайные графы, модели и генераторы безмасштабных графов // Труды Института системного программирования РАН. 2012. Т. 22. С. 419–432.
13. **Newman M.E.J.** A measure of betweenness centrality based on random walks. URL: <http://aps.arxiv.org/pdf/cond-mat/0309045.pdf>. (Проверено: 10 октября 2015).
14. **Newman M.E.J.** The mathematics of networks. URL: <http://www-personal.umich.edu/~mejn/papers/palgrave.pdf>. (Проверено: 13 октября 2015).
15. Анализ социальных сетей. URL: <http://datareview.info>. (Проверено: 1 ноября 2015)/
16. **Mikov A., Zamyatina E., Kozlov A., Ermakov S.** Some Problems of the Simulation Model Efficiency and Flexibility // Proceedings of «2013 8th EUROSIM Congress on Modelling and Simulation EUROSIM 2013», Cardiff, Wales, United Kingdom, 10–13 of September. P. 532–538.