

УДК 004.414.23

ВЛИЯНИЕ СБАЛАНСИРОВАННОСТИ НА ВРЕМЯ ИСПОЛНЕНИЯ РАСПРЕДЕЛЕННОЙ ИМИТАЦИОННОЙ МОДЕЛИ

Д.Н. Шинкарук^а

^а Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

Адрес для переписки: dimashink@gmail.com

Информация о статье

Поступила в редакцию 21.11.14, принята к печати 25.12.14

doi: 10.17586/2226-1494-2015-15-1-101-106

Язык статьи – русский

Ссылка для цитирования: Шинкарук Д.Н. Влияние сбалансированности на время исполнения распределенной имитационной модели // Научно-технический вестник информационных технологий, механики и оптики. 2015. Том 15. № 1. С. 101–106

Аннотация. Представлены результаты исследования влияния распределения нагрузки между вычислительными узлами на время выполнения распределенной имитационной модели компьютерной сети. Рассматриваются два основных типа балансировки – по вычислительной нагрузке и по объему передаваемых данных. Моделирование проводилось на одном компьютере, и распределение осуществлялось между ядрами одного процессора. В результате проведенных имитационных экспериментов на моделях сетей сетчатой топологии показано, что отсутствие сбалансированности по объему передаваемых данных между частями распределенной модели ведет к снижению скорости выполнения модели в несколько раз, что объясняется накладными расходами на передачу данных между логическими процессами распределенной модели в связи с использованием MPI. Изменение времени выполнения модели при неравномерном распределении вычислительной нагрузки в значительной степени зависит от нагрузки, которую создают приложения, работающие на узлах моделируемой сети. Показано, что даже при использовании приложений, не требующих значительных вычислительных ресурсов, сбалансированная модель выполняется быстрее несбалансированного варианта. Для уменьшения времени распределенного моделирования предлагается разделение модели таким образом, чтобы уменьшить количество передаваемых данных между ее частями и сократить разброс нагрузок, создаваемый приложениями в разных логических процессах модели.

Ключевые слова: распределенное моделирование, компьютерные сети, NS-3, балансировка нагрузок.

RUNTIME BALANCING EFFECT IN DISTRIBUTED SIMULATION MODEL

D.N. Shinkaruk^а

^а ITMO University, Saint Petersburg, 197101, Russian Federation

Corresponding author: dimashink@gmail.com

Article info

Received 21.11.14, accepted 25.12.14

doi: 10.17586/2226-1494-2015-15-1-101-106

Article in Russian

Reference for citation: Shinkaruk D.N. Runtime balancing effect in distributed simulation model. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, vol. 15, no. 1, pp. 101–106 (in Russian)

Abstract. The paper deals with study of the load distribution effect between computing nodes for simulation run-time of a distributed computer network model. Two main types of balancing are considered: computational load balancing and reduction of transmitted amounts of data. Simulation was performed on one computer, and distribution was carried out between the cores of one processor. Simulation experiments showed that the lack of balance in the amounts of data, transferred between parts of a distributed model, leads to decrease of the simulation speed by several times due to overhead charges for data transmission between logical processes because of MPI usage. Model run-time changing at uneven distribution of computational load depends to a large extent on the load, which is created by the applications running on the simulated network nodes. It is shown that a balanced model is performed much faster than unbalanced version even when using applications that do not require significant computing resources. Simulation time reduction can be achieved by model separation in such manner as to reduce amounts of data transferred between its parts and reduce variability of loads generated by applications in different logical processes.

Keywords: distributed simulation, computer networks, NS-3, load balancing.

Введение

Большие системы состоят из тысяч, десятков тысяч элементов и еще большего количества связей между ними. Такие системы характеризуются неоднородностью элементов (большим количеством раз-

личных типов элементов) и неоднородностью связей. Это приводит к тому, что классические математические методы становятся практически непригодными для описания и анализа больших систем [1]. Примером большой системы может быть компьютерная сеть провайдера или сеть крупного предприятия. Несмотря на то, что отдельные элементы или связи прекрасно описываются моделями дискретной математики или теории массового обслуживания [2], о системе в целом этого сказать нельзя. Естественной альтернативой является использование имитационного моделирования, позволяющего соединить между собой разнородные математические модели элементов системы [3].

Имитационное моделирование успешно используется для анализа систем различного масштаба [4]. Но при исследовании больших систем возникают специфические проблемы, требующие использования параллельного или распределенного моделирования и балансировки нагрузки [5–10].

В настоящей работе рассматривается применение системы имитационного моделирования NS-3 для моделирования больших компьютерных сетей с поддержкой возможности распределения вычислений между узлами для ускорения процесса моделирования [11]. NS-3 позволяет строить компьютерные сети с глубоким уровнем детализации процессов, происходящих в узлах и каналах моделируемой сети.

Уменьшение времени исполнения распределенной модели относительно последовательной зависит от того, насколько удачно пользователь разбил модель на подмодели до начала процесса моделирования. Однако предугадать нагрузку на узлы моделируемой сети или объем передаваемых данных по линиям связи довольно сложно, и в некоторых случаях необходим детальный анализ структуры моделируемой сети [12, 13]. Таким образом, пользователю сложно эффективно разделить модель, что приведет к увеличению времени выполнения модели. В ситуациях, когда параметры модели меняются в ходе моделирования, эффективно разбить модель перед запуском модельного прогона невозможно [14]. В этом случае необходимо использовать динамическую балансировку нагрузки в процессе моделирования. Цель настоящей работы – экспериментально оценить зависимость времени имитационного эксперимента от способа разбиения модели на логические процессы.

Особенности распределенной системы моделирования NS-3

Система моделирования NS-3 служит для моделирования процессов, протекающих в компьютерных сетях [15]. В связи с этим внутреннее устройство системы во многом обусловлено этой спецификой, включая наличие различных стеков сетевых протоколов и передачу данных между узлами при помощи пакетов, структура которых полностью повторяет структуру реальных сетевых пакетов, что также делает возможным использование системы в реальных компьютерных сетях [16].

Модель в NS-3 состоит из узлов, выполняющих роль компьютеров, к которым подключаются сетевые адаптеры, которые, в свою очередь, соединяются при помощи каналов передачи данных. Для выполнения какой-либо полезной работы на узлы устанавливаются приложения, которые выполняют необходимую работу (генерация, обработка, прием данных) [17].

Распределение модели в NS-3 выполняется следующим образом: узлам при создании присваивается определенный номер, который соответствует номеру логического процесса (ЛП), в котором будет обрабатываться этот узел. ЛП – это последовательная часть модели, которая имеет собственный набор объектов, управляющую программу и часы локального времени. В ЛП происходит моделирование только тех процессов, которые относятся к узлам, привязанным к данному ЛП. При установке приложений на узлы учитывается то, к какому ЛП принадлежит данный узел, и приложения, создающие наибольшую вычислительную нагрузку, не дублируются между подмоделями и выполняются только в пределах своего ЛП. В отличие от приложений, все узлы модели и связи между ними создаются во всех ЛП, т.е. каждый ЛП содержит полную топологию моделируемой сети. Сделано это для упрощения процессов маршрутизации [18].

Связь логических процессов между собой осуществляется при помощи MPI (*Message Passing Interface*). Если канал, соединяющий два моделируемых узла, пересекает границы логических процессов, то пакет, передаваемый по этому каналу, упаковывается в сообщение MPI, передается сетевыми средствами операционной системы, в которой выполняется модель, и после приема пакета ЛП-адресатом извлекается из сообщения и преобразовывается в структуру пакета NS-3 для дальнейшей передачи по каналам связи.

Несбалансированность по вычислительной нагрузке

Для демонстрации результатов неэффективного разбиения модели был проведен ряд экспериментов. Были созданы сети сетчатой топологии различных размеров: 20×10 , 40×10 , 60×10 . В каждой моделируемой сети было запущено различное количество приложений. Приложения делятся на два вида: генераторы и приемники трафика. Модель разбивалась на два логических процесса, каждый из которых содержал одинаковое количество моделируемых узлов. Генераторы трафика посылали данные на узлы, расположенные в том же ЛП (трафик между логическими процессами не передавался). Различное распределение приложений между логическими процессами позволяло создавать сбалансированную или несбалансированную распределенную модель. Каждое приложение-генератор создает равномерный поток UDP-пакетов в направлении приемника. Скорость потока равна 1 Мбит/с, размер пакетов – 512 Б. Генераторы

и приемники трафика располагаются равномерно по краям сетки в моделируемой сети в данном ЛП (рис. 1). Таким образом, каждая пара генератор–приемник создает одинаковую нагрузку в рамках одного ЛП, так как между ними располагается одинаковое количество промежуточных узлов. Моделировалось 5 мин работы такой сети. Все эксперименты проводились на компьютерах с процессором Intel Core i5 (4×2,6 GHz) и 8 ГБ оперативной памяти.

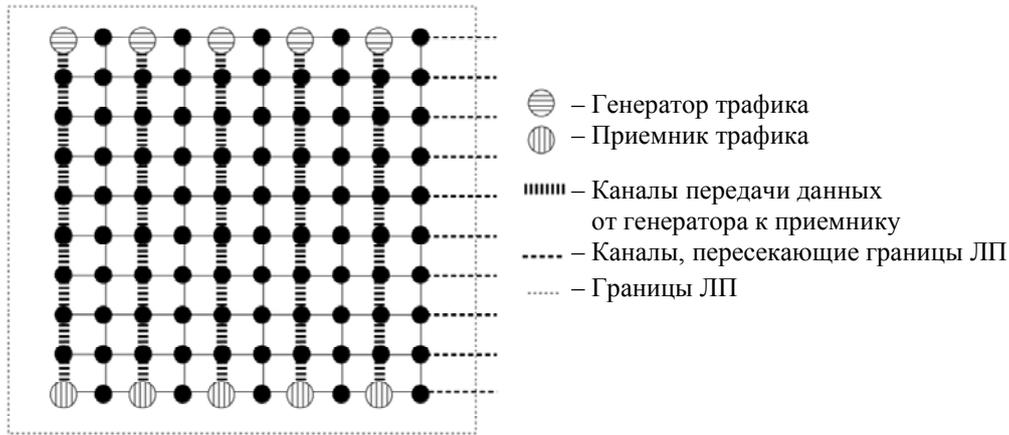


Рис. 1. Структура моделируемой сети 20×10 для одного из логических процессов

В табл. 1 представлены результаты моделирования. Отдельно от времени моделирования (T_m) выделено время построения топологии (T_t), так как для больших сетей оно составляет большую часть общего времени выполнения модели (T_f) и является постоянным для каждой отдельной сети. Большую часть этого времени занимают создание связей между узлами и настройка различных параметров для них (пропускная способность, ip-адреса интерфейсов). Количество таких связей в сети сетчатой топологии будет выражаться формулой $M \times (N-1) + N \times (M-1)$, где M и N размеры сетки.

Из табл. 1 видно, что время моделирования сбалансированной модели и худшего варианта несбалансированной модели различается в 2,47 раза для сети размером 20×10 и в 3,54 раза для сетей 40×10 и 60×10. Однако такое сравнение различных моделей некорректно, так как отношение количества приложений в разных ЛП больших и малых моделируемых сетей различно. Так, для худшего варианта несбалансированности сети 20×10 оно составляет 1/9, а для сети 60×10 – 1/29. Если же сравнивать время моделирования для сетей с одинаковым соотношением несбалансированности, то получим, что при соотношении 1/9 модель сети 20×10 выполнялась в 2,47 раза медленнее сбалансированной модели, тогда как модель сети 60×10 (распределение приложений 27×3) в 3,14 раз медленнее. Связано это с тем, что с увеличением количества приложений в ЛП возрастает нагрузка на вычислительное ядро системы, на котором обрабатывается данный ЛП, в том числе и из-за возрастающего объема передаваемых данных, что влечет за собой увеличение нагрузки на подсистему работы с памятью.

Размер сети	Количество приложений	Распределение приложений между ЛП	T_f , с	T_t , с	T_m , с
20×10	10	9×1	2,1	0,2	1,9
		7×3	1,5	0,2	1,3
		5×5	0,9	0,2	0,7
40×10	20	19×1	23,1	7,7	15,4
		17×3	20,1	7,6	12,4
		15×5	17,9	7,6	10,2
		13×7	15,3	7,6	7,6
		10×10	12,0	7,7	4,3
60×10	30	29×1	171,1	108,7	62,3
		27×3	165,5	110	55,5
		25×5	158,8	110,2	48,5
		23×7	152	109,8	42,2
		20×10	142,8	109,9	32,8
		17×13	133,8	109,9	23,8
		15×15	129,6	111,9	17,6

Таблица 1. Сравнение времени выполнения сбалансированной и несбалансированной по вычислительной нагрузке модели

Несбалансированность по объему передаваемых данных

В предыдущем эксперименте в разных ЛП модели работало различное количество приложений, которые создавали разную нагрузку на вычислительные узлы. Модель была построена таким образом, что трафик между логическими процессами не передавался. Однако, так как для взаимодействия ЛП используется MPI, то пакеты, пересекающие границы ЛП, должны быть упакованы в сообщения MPI и переданы через сетевую инфраструктуру операционной системы. Это создает дополнительные накладные расходы и должно влиять на время выполнения модели [19, 20].

Для следующего эксперимента также были построены модели сетей сетчатой топологии (10×10, 20×20, 30×30), однако трафик между узлами-генераторами и приемниками пересекал границы ЛП (рис. 2). Модель построена так, чтобы в каждом ЛП было одинаковое количество приложений-генераторов и приемников. Таким образом, исключается влияние количества приложений в логических процессах на время выполнения модели.

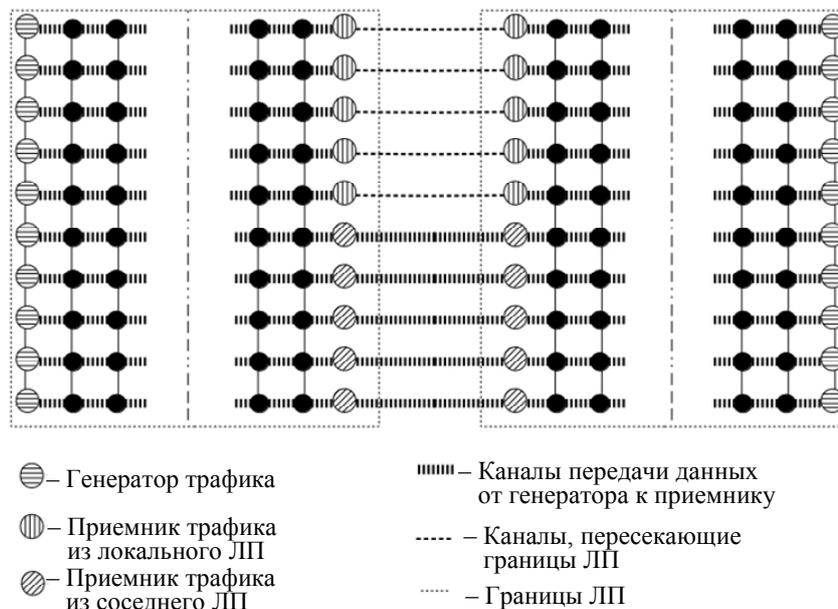


Рис. 2. Структура модели компьютерной сети, в которой трафик пересекает границы логических процессов

Размер сети	Количество приложений	Распределение потоков данных	T_f , с	T_m , с
10×10	20	20×0	3,53	3,33
		16×4	2,92	2,72
		12×8	2,34	2,15
		8×12	1,81	1,58
		4×16	1,16	0,96
		0×20	0,57	0,36
20×20	40	40×0	48,71	40,97
		32×8	45,97	38,23
		24×16	43,77	36,14
		16×24	37,24	29,58
		8×32	30,75	23,13
		0×40	23,94	16,24
30×30	60	60×0	294,43	189,72
		48×12	288,39	184,08
		36×24	279,5	176,44
		24×36	257,15	151,04
		12×48	220,42	116,92
		0×60	185,34	81,19

Таблица 2. Время выполнения модели с пересечением потоками данных границ ЛП

В табл. 2 представлены результаты проведенных экспериментов. Так как количество приложений-генераторов в каждом ЛП одинаково, то в таблице представлено распределение приложений, трафик ко-

торых пересекает и не пересекает границы логических процессов. Так, соотношение 12×28 для 40 приложений означает, что потоки данных 12 приложений пересекают границы ЛП, а 28 приложений передают данные только в пределах одного ЛП.

Из табл. 2 видно, что при отсутствии передачи данных между логическими процессами моделирование выполняется в 2,2–2,5 раза быстрее для больших сетей и до 9,2 раза для сети 10×10 узлов, чем в ситуации, когда все приложения передают данные в соседний ЛП.

На рис. 3 представлена зависимость времени моделирования от распределения приложений, передающих данные только внутри ЛП и пересылающих данные в соседний ЛП.

Таким образом, время моделирования для представленных моделей могло быть сокращено за счет уменьшения объема передаваемых данных между логическими процессами. Сделать это можно, изменяя границу разбиения модели на логические процессы таким образом, чтобы максимально локализовать трафик внутри ЛП. Для представленной на рис. 2 модели можно проводить границу не в вертикальной плоскости, а в горизонтальной, в таком случае весь трафик будет локализован внутри отдельных ЛП. Однако для большинства моделей такое идеальное разбиение затруднительно, так как распределение узлов генераторов и приемников может быть случайным. В таком случае перераспределение генераторов и приемников необходимо выполнять отдельно на уровне моделируемых узлов.

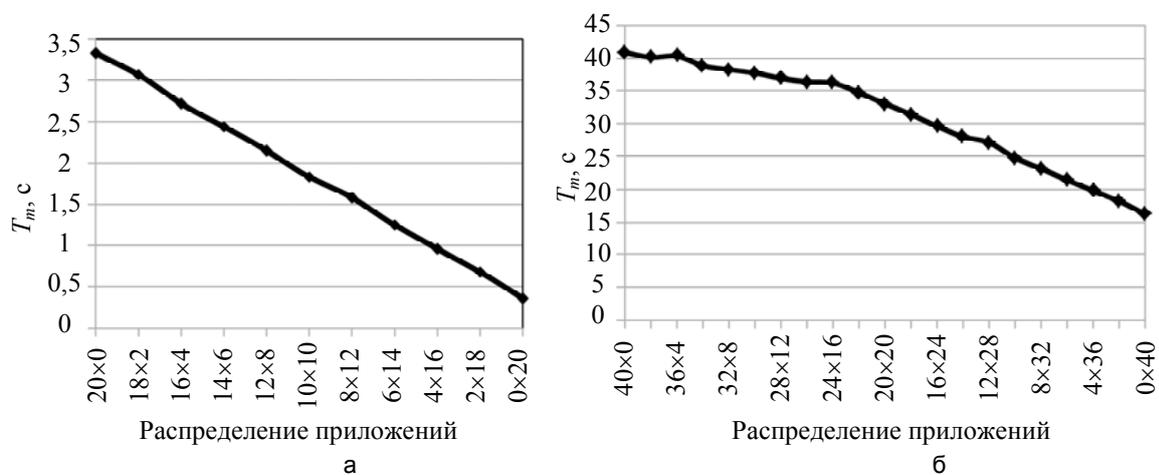


Рис. 3. Зависимость времени моделирования от распределения трафика между ЛП: 20 активных приложений в сети 10×10 (а); 40 активных приложений в сети 20×20 (б)

Заключение

В результате проведенных экспериментов показано, что несбалансированность распределенной имитационной модели может значительно влиять на время моделирования. В некоторых случаях расчет плохо сбалансированной распределенной модели может выполняться значительно дольше аналогичной нераспределенной модели из-за накладных расходов, связанных с синхронизацией времени и передачей данных между логическими процессами.

В работе рассмотрена балансировка нагрузки между вычислительными узлами при распределении модели сети передачи данных с сетчатой топологией. Экспериментально показано, что время выполнения несбалансированной модели, в которой логические процессы обрабатывают разное количество приложений, может в 2,5–3 раза превышать время выполнения сбалансированной модели. При моделировании больших компьютерных сетей с большим количеством приложений, создающих значительную нагрузку на вычислительные узлы, эта разница будет значительно больше.

Также показано, что изменение количества передаваемых данных между логическими процессами сильно влияет на время выполнения модели из-за значительных накладных расходов на передачу данных между логическими процессами. Для малых сетей соотношение между временем исполнения сбалансированной и несбалансированной модели достигло 9 раз. Для больших моделей (20×20 и 30×30) соотношение составило 2,2–2,5 раза. Связано это с тем, что в больших моделях значительную часть времени выполнения тратится на моделирование работы приложений. Таким образом, большая часть времени работы логических процессов тратится на обработку приложений и передачу данных между узлами внутри отдельного логического процесса.

Для уменьшения времени исполнения распределенных моделей компьютерных сетей следует учитывать два основных фактора, влияющих на скорость моделирования: равномерность загрузки логических процессов и минимизация передаваемых между логическими процессами данных. В большинстве моделей эти два фактора могут быть связаны друг с другом различными соотношениями, в такой ситуации необходимо минимизировать функцию времени от этих двух параметров.

References

1. Mikov A.I., Zamyatina E.B. Tekhnologiya imitatsionnogo modelirovaniya bol'shikh sistem [Simulation technology of large systems]. *Trudy Vserossiiskoi Nauchnoi Konferentsii "Nauchnyi Servis v Seti Internet"* [Proc. All-Russian Conference Scientific Service in the Internet]. Moscow, MGU Publ., 2008, pp. 199–204.
2. Aliev T.I. *Osnovy Modelirovaniya Diskretnykh Sistem* [Basics of Discrete Systems' Modeling]. St. Petersburg, SPbSU ITMO, 2009, 363 p.
3. Fujimoto R.M. *Parallel and Distributed Simulation Systems*. Wiley, 2000, 320 p.
4. Perumalla K., Park A., Wu H., Ammar M.H., Riley G.F. Large-scale network simulation: how big? How fast? *IEEE 20th Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2003, pp. 116–123.
5. Manjikian N., Loucks W.M. High performance parallel logic simulations on a network of workstations. *ACM SIGSIM Simulation Digest*, 1993, vol. 23, no. 1, pp. 76–84.
6. Misra J. Distributed discrete-event simulation. *ACM Computing Surveys*, 1989, vol. 18, no. 1, pp. 39–65.
7. Chandy K.M., Misra J. Distributed simulation: a case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering*, 1979, vol. SE-5, no. 5, pp. 440–452.
8. Wentong C., Turner S.J., Hanfeng Z. A load management system for running HLA-based distributed simulations over the grid. *Distributed Simulation and Real-Time Applications*, 2002, pp. 7–14.
9. Deelman E., Szymanski B.K. Dynamic load balancing in parallel discrete event simulation for spatially explicit problems. *Proc. Workshop on Parallel and Distributed Simulation, PADS'98*. Banff, Canada, 1998, pp. 46–53.
10. Schlagenhaft R., Ruhwandl M., Bauer H., Sporrer C. Dynamic load balancing of a multi-cluster simulator on a network of workstations. *Proc. 9th Workshop on Parallel and Distributed Simulation, PADS'95*. Lake Placid, USA, 1995, vol. 25, no. 1, pp. 175–180.
11. Renard K., Peri C., Clarke J. *A performance and scalability evaluation of the NS-3 distributed scheduler*. 2012. Available at: <http://eudl.eu/pdf/10.4108/icst.simutools.2012.247679> (accessed 18.11.2014).
12. Nandy B., Loucks W.M. An algorithm for partitioning and mapping conservative parallel simulation onto multicomputers. *Proc. 6th Workshop on Parallel and Distributed Simulation*, 1992, pp. 139–146.
13. Nandy B., Loucks W.M. On a parallel partitioning technique for use with conservative parallel simulation. *Proc. 7th Workshop on Parallel and Distributed Simulation*. San Diego, USA, 1993, vol. 23, no. 1, pp. 43–51.
14. Ferscha A. Parallel and Distributed Simulation of Discrete Event Systems. In: *Handbook of Parallel and Distributed Computing*. McGraw-Hill, 1995, 49 p.
15. Henderson T.R., Floyd S., Riley G.F. *NS-3 Project Goals*. Available at: <http://www2.nsnam.org/docs/meetings/wns2/wns2-ns3.pdf> (accessed 15.11.2014)
16. Aliev T.I., Sosnin V.V., Shinkaruk D.N., Tikhonov M.Yu., Burmakin N.G. SAPR marshrutiziruemoi komp'yuternoi seti na osnove komponentov s otkrytymi iskhodnymi kodami [Creating A CAD Simulator of a routed computer network using open source components]. *Izv. vuzov. Priborostroenie*, 2012, vol. 55, no. 10, pp. 47–53.
17. Sosnin V.V., Shinkaruk D.N. Osobennosti proektirovaniya imitatsionnoi modeli marshrutiziruemoi komp'yuternoi seti [Designing features of routed network simulation model]. *Sbornik Trudov Molodykh Uchenykh i Sotrudnikov Kafedry VT. Vypusk 3* [Proc. of Young Scientists and Staff of the Department VT. Issue 3]. St. Petersburg, NIU ITMO, 2012, pp. 57–63.
18. Pelkey J., Riley G. *Distributed simulation with MPI in NS-3*. 2011. Available at: <http://users.ece.gatech.edu/~riley/ece6110/handouts/DistNS3.pdf> (accessed 15.11.2014).
19. Liu X., Chien A.A. Traffic-based load balance for scalable network emulation. *Proc. 2003 ACM/IEEE Conference on Supercomputing*, 2003, pp. 40. doi: 10.1145/1048935.1050190
20. Zhou S. Trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, 1988, vol. 14, no. 9, pp. 1327–1341. doi: 10.1109/32.6176

Шинкарук Дмитрий Николаевич – ассистент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, dimashink@gmail.com

Dmitry N. Shinkaruk – assistant, ITMO University, Saint Petersburg, 197101, Russian Federation, dimashink@gmail.com