

О.В. Кузьмін, к.т.н., доцент, Ю.З. Знак, бакалавр каф. АСУ Національного університету «Львівська політехніка», Львів

ПРОГРАМНІ ЗАСОБИ МОДЕЛЮВАННЯ СЕНСОРНИХ МЕРЕЖ

У статті наведено огляд програмних засобів моделювання сенсорних мереж. Розглядаються найбільш поширені системи моделювання такі як: TOSSIM, NS, OMNET++, AnyLogic. Наведена порівняльна оцінка розглянутих програмних засобів.

Keywords – sensor, software, framework, protocol, interface.

In article provides an overview of the software for modeling sensor networks. The most common modeling systems are considered: TOSSIM, NS, OMNET ++, AnyLogic. The comparative estimation of the considered software is presented.

Ключові слова – сенсор, програмне забезпечення, фреймворк, протокол, інтерфейс.

Вступ. Бездротові сенсорні мережі (БСМ), знаходять широке застосування в сферах моніторингу навколишнього середовища, попередження катастроф, охорони здоров'я, контролю за технологічними процесами на виробництвах, обліку угідь, безпеки, оборони та багатьох інших [1].

Підприємства промисловості різних галузей інтегрують у свої технологічні процеси сенсорні мережі, датчики яких безперервно надсилають на контрольну станцію дані для контролю за станом роботи виробництва [2-3]. Незважаючи на те, що безпроводні технології набули широкого розповсюдження, все ще не існує уніфікованих апаратно-програмних рішень, а тому кожна специфічна прикладна задача потребує окремого підходу для її реалізації [5]. Варто зазначити, що проведення натурних випробувань пов'язане із значними витратами часу та людських ресурсів. Тому використання програмних засобів імітаційного моделювання БСМ дасть змогу значно зменшити витрати на їх проектування.

Метою даної роботи є аналіз сучасних засобів моделювання БСМ, виявлення особливостей їх застосування, переваг та можливих вузьких місць, щоби в подальшому розробити систему моделювання з відкритою архітектурою, яка безпосередньо була би призначена для моделювання БСМ.

TOSSIM. Система моделювання TOSSIM є складовою частиною проекту TinyOS [8]. TinyOS – це операційна система для мікроконтролерів, які входять до складу нодів, що є вузлами БСМ. Вона дає можливість оперувати всіма низькорівневими функціями вузла, серед яких керування сенсорами та передача даних мережею. Ця ОС написана мовою nesC (розширення мови C). Сама ж система TOSSIM є фреймворком, бібліотека якого містить велику кількість пакетів із імплементаціями найрізноманітніших програмних компонентів, а також функцій, алгоритмів та протоколів, що використовуються в БСМ. За їх допомогою можна будувати

програмну імітацію реальних складових елементів вузла БСМ, програмувати внутрішні процеси цих елементів та особливості взаємодії між ними [9].

TOSSIM є симулятором дискретних подій, призначеним для високоточного моделювання систем на TinyOS. Симуляція подій у БСМ відбувається на рівні бітів та апаратних переривань. Але, є можливість виконувати моделювання системних подій більш високого рівня (наприклад, пакетний прийом).

Нижче наведено характеристики TOSSIM, які визначають його сферу застосування, та основні переваги [9]:

- масштабованість – фреймворк здатний ефективно моделювати БСМ із тисячами вузлів в широкому діапазоні можливих конфігурацій;
- повнота – система дозволяє отримувати та відслідковувати точну поведінку мережі у всіх аспектах роботи, виявляючи навіть неочікувані взаємодії;
- перенесення – програмний код, створений для TOSSIM-моделі, може бути відразу розгорнутий на її апаратній реалізації.

Хоча TOSSIM забезпечує дуже широкий функціонал для опису внутрішньої будови БСМ, але, при цьому, не надає достатніх можливостей для моделювання подій реального світу [10].

Оскільки TOSSIM працює із операційною системою TinyOS, тому програмний код, написаний для імітаційної моделі, може бути імпортований в реально існуючу сенсорну мережу, побудовану згідно з цією моделлю. Але, як наслідок, зосередженість на одній єдиній ОС накладає обмеження на використання TOSSIM.

Варто зазначити, що оскільки TOSSIM - це фреймворк, то є необхідність окремого створення зовнішньої програмної оболонки, яка налаштовуватиме симуляцію та запускатиме її [10]. Для цього TOSSIM підтримує два інтерфейси для роботи із мовами програмування Python та C++. Також існує пакет для візуалізації TOSSIM-моделей TinyViz.

Сімейство систем моделювання Network Simulator. NS (Network Simulator, або симулятор мережі) – це назва лінійки об'єктно-орієнтованих середовищ для імітації мережевих подій у часі [11-13]. Існує три версії: NS-1 (більше не підтримується), NS-2 (підтримка і розвиток продовжується за рахунок користувачів) та NS-3 (швидко розвивається розробниками). Network Simulator версій 2 та 3 в наш час широко використовується для моделювання БСМ.

Модель на NS-2 реалізується за допомогою комбінації OTcl (об'єктно-орієнтована версія мови TCL) – мови для написання сценаріїв, що описують топологію мережі, конфігурації, різні затримки, моделі втрат сигналів; та C++, на якій створюється ядро імітації (на цій мові написано багато вже готових компонентів, тому, часто немає потреби писати специфічний код). Через таку «двомовність» спостерігаються деякі втрати швидкодії при роботі інтерпретатора OTcl та через накладні витрати на взаємодію цих двох мов через

спеціальний інтерфейс TCLCL [14]. До того ж, скрипти OTcl, як правило, дуже ускладнюють модель через великі об'єми коду (часто переплетеного із кодом C++), що є серйозним недоліком при тестуванні та супроводі.

Ключовою перевагою популярності NS-2 є його багата бібліотека протоколів для майже всіх мережевих шарів і протоколів для численних механізмів маршрутизації. В NS-2 на рівні ядра реалізовано майже всі відомі мережеві протоколи на всіх рівнях, серед яких: IPv6, MPLS, OSPF, різноманітні версії TCP, UDP, RAP, RTP і дуже багато інших. Також є готові реалізації для цілої лінійки дисциплін обслуговування черг: WFQ, RED, SFQ тощо. Серед додаткових можливостей NS-2 варто згадати сильний математичний апарат, який дозволяє створювати різні види трафіку всіх рівнів складності. Середовище наділено потужним інструментарієм моделювання помилок. Таке моделювання можливо проводити на рівні бітів, пакетів або імовірності на секунду. Можлива і реалізація користувацької моделі помилок.

Процес моделювання в середовищі NS-2 є відносно складним і довготривалим [11].

При створенні БСМ слід враховувати, що NS-2 стандартно не підтримує розповсюджену для БСМ специфікацію ZigBee, не передбачає можливості задати переміщення вузлів БСМ в просторі та не дозволяє відразу отримувати показники енерговитрат вузлів [15].

Середовище NS-3 є подальшим розвитком лінійки Network Simulator і призначене для виправлення більшості недоліків свого попередника. В NS-3 розробники відмовилися від використання окремої скриптової мови, тому весь код пишеться за допомогою C++. Для написання скриптів, крім C++, обмежено підтримується Python. Для візуалізації процесу моделювання та його результатів використовують Python-пакет PyViz.

Завдяки єдиній мові програмування моделі на NS-3 є краще оптимізованими через більшу кількість інструментів для гнучкого керування пам'яттю [16]. Результати дослідження свідчать про те, що NS-3 демонструє один з найкращих показників швидкодії серед середовищ для розробки моделей БСМ. Його швидкодія значно перевищує показники NS-2.

Великою перевагою NS-3 є те, що код моделей може бути відразу успішно перенесений на реально існуючу систему, наприклад БСМ. Крім того NS-3 підтримує взаємодію імітаційної моделі, розгорнутої на окремих віртуальних машинах, із реальними системами.

Основну увагу середовище NS-3 приділяє моделюванню бездротових/IP-моделей, що включають моделі для різних шарів OSI, а також різні статичні або динамічні протоколи маршрутизації, такі як OLSR та AODV для IP-додатків [17]. Судячи з відгуків користувачів, NS-3 значно більше підходить для моделювання на рівнях від транспортного до прикладного включно. З іншого боку, попри існування моделей для 1 і 2 шарів для NS-3, з метою серйозної роботи на фізичному, каналному та мережевому рівнях, рекомендують NS-2 [18].

Хоча NS-3 усуває більшість недоліків NS-2 та надає користувачу ширший функціонал, однією з найсерйозніших вад цієї платформи є

підтримка відносно малої кількості протоколів.

OMNET++. Фреймворк OMNET++ отримав широку популярність серед науковців та інженерів як надійна та ефективна платформа для моделювання мережеских подій [19]. А такі розширення, як Castalia та MiXiM зробили OMNET++ відмінним засобом для створення імітаційних моделей БСМ. Система надає компонентну архітектуру для опису моделей. Компоненти (модулі) запрограмовані мовою C++ і є не подільними елементами модульної ієрархії. Найчастіше вони виконують передачу та отримання повідомлень [20]. Ці компоненти об'єднані в більші компоненти та моделі, що використовують високорівневу мову програмування NED. В свою чергу такі компоненти можуть об'єднуватися в ще більші моделі, таким чином утворюючи необмежену ієрархію. Така модульна архітектура дозволяє гнучко використовувати існуючі компоненти в користувацьких програмах. Передбачена можливість описання параметричних топологій. Ієрархічність компонентів є особливо зручною при створенні моделей БСМ. Функціональні можливості OMNET++ та підтримка багатьох протоколів для БСМ дозволяють створювати ефективні імітаційні моделі БСМ різноманітних типів.

Визначальною рисою OMNET++ є розділення різноманітних аспектів моделювання. Поведінка компонентів моделі описана мовою C++, а високорівневі топології - за допомогою NED [21].

Система OMNET++ володіє потужним графічним редактором, який входить до складу інтегрованого середовища розробки. Зручність редактора полягає у тому, що користувач має змогу працювати над топологією моделі в графічному режимі або безпосередньо в NED коді. В цьому є одна з переваг роботи з OMNET++ над NS-2, де об'єднання логіки на C++ та опису топології на Tcl не дозволяє створити динамічний графічний редактор коду.

На відміну від середовищ NS, OMNET++ містить бібліотеки, які відповідають стандартам специфікації ZigBee та різноманітні протоколи маршрутизації. Крім цього в моделях OMNET++ є можливість передбачити рух вузлів у просторі та розрахувати показники енерговитрат у вузлах [15].

Ефективність роботи великих моделей на OMNET++ досягається, в тому числі, за рахунок підтримки виконання паралельних імітацій [22]. Детальна імітація роботи БСМ із тисячами вузлів може бути дуже довготривалою. Функціонал OMNET++ дозволяє розподіляти процес моделювання роботи БСМ між декількома процесорами, що дозволяє відчутно зекономити час та розподілити навантаження.

AnyLogic. Середовище AnyLogic – це потужна програмна платформа, призначена для проведення імітаційного моделювання широкого кола бізнес-процесів. Це кросплатформне програмне забезпечення написано мовою Java та дозволяє використовувати її для розробки моделей [23]. Всі проекти, створені в AnyLogic, є повноцінними Java-проектами. Такі проекти працюватимуть на всіх системах, де встановлено середовище виконання для Java (Java Runtime Environment). З цим пов'язано високий рівень адаптованості робочих моделей на різноманітні операційні системи або веб-сервіси.

Середовище AnyLogic надає можливість використовувати різноманітні парадигми моделювання: неперервне, дискретно-подійне та багатоагентне. Ці парадигми можна гнучко комбінувати в межах однієї моделі. Важливими перевагами AnyLogic є об'єктно-орієнтований підхід до моделювання, потужні можливості мови Java та GUI середовище з багатим функціоналом. Засоби графічного інтерфейсу користувача цієї системи настільки багатофункціональні, що переважно при створенні моделі немає потреби в написанні коду [7]. Окрім того, процес моделювання супроводжується наочною графічною візуалізацією. Проте, якщо цього не є достатньо, функціональність моделі можна легко розширити, написавши власну логіку мовою Java, або під'єднавши до моделі вже готові Java-бібліотеки.

В середовищі AnyLogic введено і активно підтримується поняття активного об'єкту (АО), яке фактично є розширенням терміну об'єкта в об'єктно-орієнтованих мовах програмування. Окрім стандартних полів та методів, АО можуть включати в себе параметри моделі, зовнішні змінні для зв'язування станів двох різних АО, таймери, призначені задавати часові затримки та періодичні події, порти для обміну повідомлень, діаграми станів АО, математичні функції тощо. Кожному АО середовище AnyLogic присвоює власний потік управління. Активні об'єкти взаємодіють шляхом передачі повідомлень через посередництво зовнішніх змінних. За підходом багатоагентного моделювання, кожен такий АО може розглядатися, як окремих незалежний агент.

Моделі в AnyLogic створюються шляхом побудови ієрархічних взаємозв'язків між активними об'єктами [25]. Існує один єдиний кореневий АО, який включає в себе всі інші. Загалом, будь-який АО може містити в собі довільну кількість інших із визначеними зв'язками між ними. Порядок таких взаємозв'язків розробник встановлює за допомогою блоків умовних переходів та блоків об'єднань. Такий ієрархічний модульний підхід до побудови архітектури дозволяє розробляти складні та адаптивні моделі за короткий час при відносно невеликих супутніх витратах.

Перелічені вище особливості середовища AnyLogic роблять його придатним для проведення фахового високорівневого імітаційного моделювання БСМ [4, 6, 24-25]. Розробник може з легкістю змінювати функціональність на одному з шарів моделі, при цьому не перебудовуючи структуру на всіх інших. Важливою перевагою цього середовища при імітаційному моделюванні БСМ є застосування принципів багатоагентного моделювання, які дозволяють всебічно досліджувати цілісну поведінку БСМ шляхом аналізу станів її кожної окремої компоненти.

Для AnyLogic існують стандартні пакети об'єктів для створення моделей БСМ. Проте, оскільки AnyLogic – це середовище загального застосування, тут не передбачено великого розмаїття готових специфічних компонентів та протоколів, які надавали б всебічний функціонал для розробки моделей та топологій БСМ будь-якого типу. Це може стати причиною певних незручностей, якщо виникає необхідність в моделюванні нестандартних рішень.

Порівняльний аналіз розглянутих систем моделювання наведений у таблиці 1.

Таблиця 1

Порівняльна таблиця систем моделювання

	TOSSIM	NS-2	NS-3	OMNET	AnyLogic
Підтримка коду моделювання	тільки nesC	ні	так	так	ні
Моделювання динамічної зміни топології мережі	ні	так	так	так	так
Моделювання енергоспоживання	так	ні	так	так	так
Загальне призначення моделювання	БСМ на TinyOS	Мережеві системи	Мережеві системи	Мережеві системи	Бізнес-процеси
Основна мова програмування	nesC, Python, C++	C++, OTCL	C++, Python	NED, C++	Java
Операційні системи	Windows, Linux	Linux, FreeBSD, macOS	Linux, FreeBSD, macOS	Windows, Linux, macOS	Windows, Linux, macOS
Масштабованість (n – кількість вузлів)	n < 1000	n < 100	n < 100	n < 1000	n < 1000
Парадигми імітаційного моделювання	дискретно-подійна	дискретно-подійна	дискретно-подійна	дискретно-подійна	дискретно-подійна, системна динаміка, агентна
Підтримка користувацьких розширень	так	так	так	так	так
Можливість надійного моделювання в реальному часі	ні	частково	так	так	так
Наявність потужного GUI	так	ні	ні	так	так

Висновок. Наведені в роботі програмні засоби забезпечують можливість побудови адекватних моделей БСМ. Використання імітаційного моделювання дає можливість зменшити витрати на розробку та експлуатацію реальних систем. Вибір конкретного середовища моделювання впливає на рівень абстракції моделі. Наприклад система TOSSIM дозволяє управляти найнижчим рівнем подій кожного TMSOS вузла TinyOS. AnyLogic дає розробникам можливість створювати моделі на рівні архітектури, протоколів

та зовнішніх подій. Деякі з цих систем, у тому числі OPNET ++ та AnyLogic, мають надзвичайно потужні графічні редактори, за допомогою яких створюються моделі, і не вимагають створення програмного коду, хоча така можливість існує. А системи NS-2 і NS-3 мають лише засоби візуалізації процесів, але написання програмного коду є необхідністю під час розробки моделей.

1. *Antoine Bagula* “Applications of wireless sensor networks”, UTC, February 2012. – pp.1-67
2. *Juan Aponte-Luis, Juan Antonio Gómez-Galán, Fernando Gómez-Bravo, Manuel Sánchez-Raya, Javier Alcina-Espigado, Pedro Miguel Teixido-Rovira*, An Efficient Wireless Sensor Network for Industrial Monitoring and Control, 2017. – pp. 1-17
3. http://www.ipmce.ru/img/release/is_sensor.pdf.
4. *Kuzmin O., Legky L.* Research Reliability Wireless Sensor Network of Cluster Topology, The VIth International Scientific and Technical Conference Computer Science & Informational Technologies 2011 (CSIT’2011), 2011, Lviv, Ukraine. - 3 p.
5. *Галелюка І. Б.* Моделювання бездротових сенсорних мереж / І. Б. Галелюка // Комп’ютерні засоби, мережі та системи, 2015. - № 14. - С. 141-150.
6. *Кузьмін О., Головка В.* Синтез топології сенсорної мережі методом імітаційного моделювання.. Матеріали 4-ї міжнародної науково-технічної конференції CSIT2009 “Комп’ютерні науки та інформаційні технології 2009”, 15-17 жовтня 2009 р., Україна, Львів, 2009. - С.108-110.
7. *Дорошенко А.Е., Жереб К.А., Шевченко Р.С.*, О моделировании сенсорных сетей средствами высокого уровня, Проблемы програмування. 2006. №2-3. Спеціальний випуск.-С. 718-727.
8. <http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM>.
9. *Tanveer Zia, Albert Zomaya*, “An Analysis of Programming and Simulations in Sensor Networks”, International Workshop on Sensor Networks and Applications (SNA05), SAS 2005, Beijing, China, 2005 – pp. 15-19.
10. *Philip Levis, Nelson Lee*, “TOSSIM: A Simulator for TinyOS Networks”, Berkeley, 2003 – pp. 1-17
11. [https://en.wikipedia.org/wiki/Ns_\(simulator\)#cite_ref-hackers_1-0](https://en.wikipedia.org/wiki/Ns_(simulator)#cite_ref-hackers_1-0).
12. *М.Ю.Зеляновський, Алхімі Мухамад, Аль-бдур Нашат, Самі Аскар* «Засоби для моделювання спеціалізованих та сенсорних мереж бездротового доступу: симулятори роботи комп’ютерних мереж NS-2 та NS-3», Зб. наук. пр. ІПМЕ ім.Г.Є.Пухова НАН України. — К.: ІПМЕ ім. Г.Є.Пухова НАН України, 2009. — Вип. 51. — С. 203-210.
13. <http://wrc-ejust.org/crn/images/Tutorials/ns2vsns3.pdf>.
14. *Кучерявий Е. А.* NS2 как универсальное средство имитационного моделирования сетей связи , Tampere University of Technology, Telecommunications Laboratory, Tampere, Finland. – 2001. – С. 1-6.
15. *А.С. Борисенко*, Адекватность моделей беспроводных сенсорных сетей в средах имитационного моделирования , Восточно-Европейский журнал передовых технологий. - 2013. - № 4(9). - С. 52-55.
16. *Rehman Khana , Sardar M. Bilal , Mazliza Othman*, A Performance Comparison of Network Simulators for Wireless Networks, Conference: IEEE International Conference on Control System, Computing and Engineering, Madrid, Spain, 2012. – pp. 1-6
17. *Rajan Patel, Dr. Pariza Kamboj*, Investigation of Network Simulation Tools and Comparison Study: NS3 vs NS2, Journal of Network Communications and Emerging

Technologies (JNCET) Volume 5, Special Issue 2, December, 2015 – pp. 137-142

18. https://www.researchgate.net/post/How_do_you_simulate_WSN_with_Ns2_Is_NS2_or_NS3_better.

19. <https://www.omnetpp.org/intro>.

20. *András Varga, Rudolf Hornig*, An overview of the OMNeT++ simulation environment, Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, SimuTools 2008, Marseille, France, March 3-7, 2008.

21. *X. Xian, W. Shi, H. Huang*, Comparison of OMNET++ and other simulator for WSN simulation, 3rd IEEE Conference on Industrial Electronics and Applications 2008. ICIEA 2008, 2008 – pp. 1439-1443.

22. *A. Varga*, Parallel Simulation Made Easy With OMNeT++, 2009.

23. <https://www.anylogic.com/upload/iblock/fd4/fd4d353e5afbbdf421ff4efea586373c.pdf>

24. *V.G. Gavrilenko, A. Yu. El'tsov, A. A. Kiryushin, S. V. Lobanov, and A. N. Sadkov*, Detailed Simulation of the Physical Level in a Simulator of Wireless Sensor Networks, Radiotekhnika i Elektronika, 2009, Vol. 54, No. 4, pp. 465–475.

25. *Кузьмін О., Головка В.* Оптимізація кластерної структури сенсорної мережі методом імітаційного моделювання. // Вісник НУ “Львівська політехніка”, № 663, Комп'ютерні науки та інформаційні технології. - Львів, 2010, с.103-107.

Поступила 22.02.2018р.

УДК 629.7.072.8

О.А. Машков¹, д.т.н., проф., Б.В.Дурняк², д.т.н., проф., Ю.В. Мамчур¹,
О.В. Тимченко^{2,3}, д.т.н., професор

СИНТЕЗ АЛГОРИТМУ ПРОГРАМНОГО КЕРУВАННЯ НА ТРЕНАЖЕРІ ДИСТАНЦІЙНО ПІЛОТОВАНОГО ЛІТАЛЬНОГО АПАРАТА НА ОСНОВІ АЛГОРИТМІЧНОЇ ПРОЦЕДУРИ РІШЕННЯ ЗВОРотноЇ ЗАДАЧІ ДИНАМІКИ (СТОХАСТИЧНА ПОСТАНОВКА)

Запропонована методика синтезу керування дистанційно пілотованим літальним апаратом в умовах нештатних (аварійних) ситуацій в імітаторі динаміки польоту тренажера. Розроблено алгоритмічний підхід до синтезу програмного керування у стохастичній постановки. Запропоновані теоретичні положення розглянуто на модельному прикладі.

Ключові слова: алгоритм керування, дистанційно пілотований літальний апарат, ефективність керування, оптимізація, об'єкт керування, оперативне програмування траєкторія, стохастична система.

¹ Державна екологічна академія післядипломної освіти та управління, Київ

² Українська академія друкарства, Львів

³ Uniwersytet Warmińsko-Mazurski w Olsztynie