

АРХИТЕКТУРА ТА ГРАФІЧНИЙ ІНТЕРФЕЙС СИМУЛЯТОРА ДЛЯ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ПОБУДОВИ ПЕРСОНАЛЬНИХ ЛОКАЛЬНИХ МЕРЕЖ БЕЗДРОТОВОГО ДОСТУПУ

Розглянуто архітектуру та графічний інтерфейс симулятора реального часу «SNOW», розробленого в межах дисертаційної роботи «Моделі та методи побудови персональних локальних мереж бездротового доступу». Наведено показники продуктивності симулятора для моделювання контролю топології типу K-NEIGH для варіантів із послідовним та паралельним виконанням кроків симуляції.

The architecture and graphical user interface of real-time simulation tool called “SNOW” and developed within the scope of dissertation project “Models and construction methods for personal local networks with wireless access” is proposed for review. Performance characteristics are given for modeling of KNEIGH-based topology control methods with sequential and parallel simulation options of simulation steps execution.

Вступ

Аналіз існуючих програмних продуктів для симуляції методів побудови та контролю топології сенсорних та актуаторних мереж бездротового доступу проведений у [1], [2] дозволив виявити переваги, недоліки та засоби покращення, що в свою чергу призвело до розробки симулятора «Сніг» (від англ. “SNOW” – Sensor Network Over Wireless). Наступні вимоги були виділені як основні, що забезпечать необхідне середовище для проведення експериментів в межах задач поставлених у дисертаційній роботі:

- Підтримка негомогенної структури мережі;
- Можливість незалежного опису та одночасного використання у експериментах різних компонент моделі мережі;
- Максимальне наближення опису поведінки вузла та протоколу до опису їх у програмному забезпеченні вузла;
- Можливість опису довільних методів побудови та протоколів обміну даними у бездротовій мережі;
- Простота опису поведінки вузлів мережі;
- Здатність до розширення симулятора шляхом додавання нових моделей;
- Можливість зміни рівня деталізації для кожної з моделей;
- Можливість зняття довільних характеристик мережі або окремих вузлів в реальному часі;

- Потужні засоби для візуалізації результатів: комунікаційні графи, графіки;
- Можливість збереження початкових параметрів та результатів проведення експериментів;
- Можливість проведення серій експериментів з різними налаштуваннями;
- Відсутність обмежень по розміру досліджуваної мережі;
- Висока швидкодія.

Засоби розробки

Було прийнято рішення застосовувати при розробці платформу .NET4.0. Для опису графічного інтерфейсу було використано XAML (eXtensible Application Markup Language) [4] – розширювану мову розмітки графічного інтерфейсу; використання цього засобу у складі WPF (Windows Presentation Foundation) [5] – графічної (презентаційної) підсистеми у складі .NET дозволило максимально ефективно використовувати можливості сучасних ПК для відображення графічних результатів моделювання (комунікаційні графи, графіки реального часу тощо). Для графічної підсистеми симулятора було використано також бібліотеки WPF Extensions [6] та WPF Dynamic Data Display [7]. Для паралельного зняття характеристик в реальному часі було використано LINQ (Language-INtegrated Query) [8] – розширення мови програмування для запитів, трансформації та модифікації множинних об'єктів. Для опису всіх моделей та алгоритмів було використано об'єктно-орієнтовану мову програмування C# [9].

Архітектура симулятора

У [3] проведено загальний опис архітектури симулятора, коротко описано взаємозв'язки між компонентами моделі мережі, реалізованої у розробленому симуляторі, пояснено функціональність ядра симулятора та описано способи проведення експериментів, що передбачаються симулятором. У даній статті детальніше описано структуру класів у симуляторі та дано пояснення обраний декомпозиції.

На діаграмах, зображених на Рис.1 та Рис. 2 показано основні програмні класи та підкласи розробленого симулятора відповідно до:

- а) реалізації різних типів пристроїв ІС для БСАМ (“Специфікація Типів Керуючого Елемента”)
- б) реалізації їх поведінки, тобто обміну повідомленнями у БСАМ (“Специфікація Стадій Роботи”).

Побудована таким чином архітектура дозволяє отримати необхідну гнучкість в описі усіх компонент моделі. Також досягається можливість дослідження довільних методів контролю топології для мережі (див. “Модель Методу Побудови” на Рис.1) з одночасною підтримкою негомогенних пристроїв (див. “Модель Вузла Мережі” на Рис.1).

Зокрема, для “Моделі Вузла Мережі” можемо описати довільні комбінації наступних її складових:

- Радіо Пристрій – надсилання та приймання даних з ефіру, дозволяє налаштувати потужність передавача та чутливість приймача, дозволяє контролювати його увімкнення;
- Мережевий Пристрій – ідентифікація в мережі; збереження, оновлення та накопичення інформації про сусідів, маршрути тощо;
- Модель Живлення – контроль заряду батареї, витрати на передачу та прийом даних;
- Керуючий Елемент – реалізує машину станів роботи самого вузла та перехід по них у відповідності до роботи поточного методу КТ; базовий набір включає чотири стани роботи вузла;

Кожна зі складових описаних вище може бути доповнена новими властивостями та дозволяє внесення змін у вже існуючі.

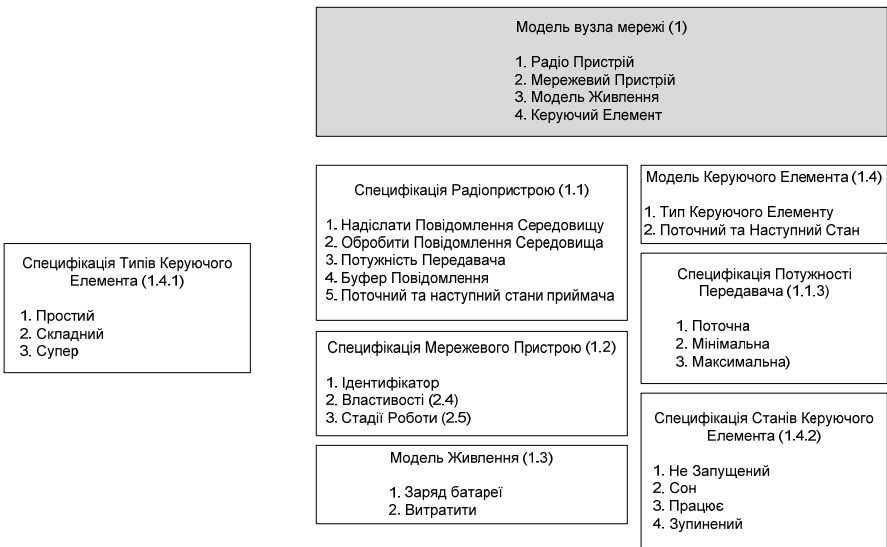


Рис.1 Керуючий елемент – модель вузла мережі та її складові.

Для “Моделі Контролю Топології” можемо описати довільні:

- типи та структуру повідомлень,;
- стани протоколу та функціональність кожного із станів;
- обробники вхідних повідомлень;
- функцію призначення комунікаційного радіусу;

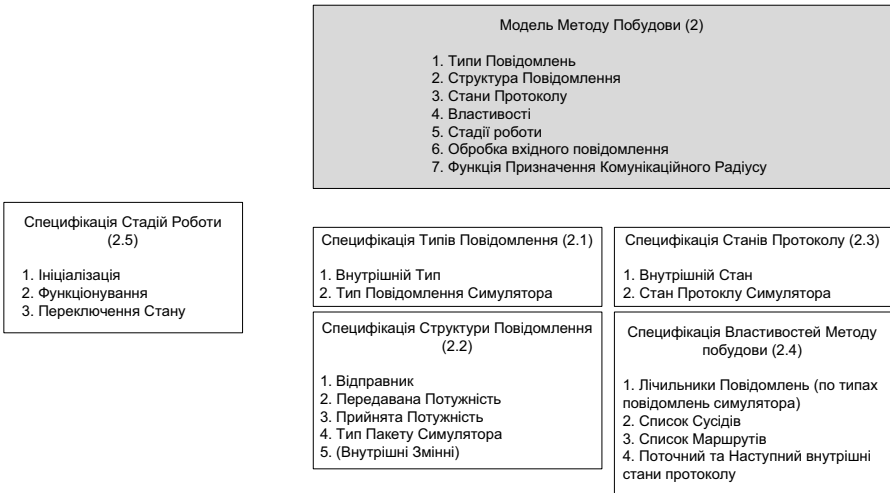


Рис.2. Опис поведінки – модель методу побудови та її складові.

На Рис.2 показано зв'язки між основними класами програми. На рисунку не показано графічної підсистеми.

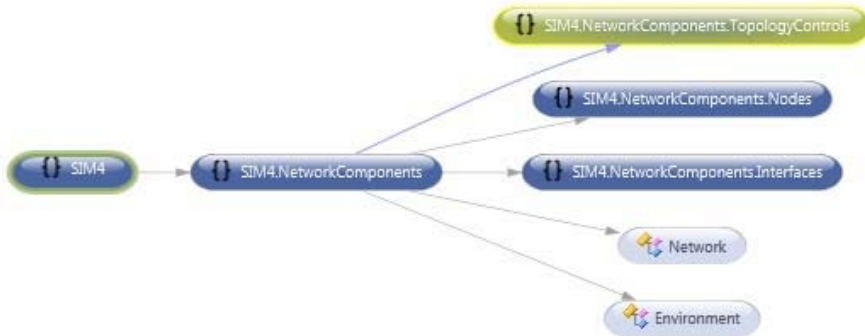


Рис.1 Основні класи симулятора.

На Рис. 3. показано інтерфейси описані для компонент моделі мережі у складі симулятора.



Рис.2 Типи вузлів.

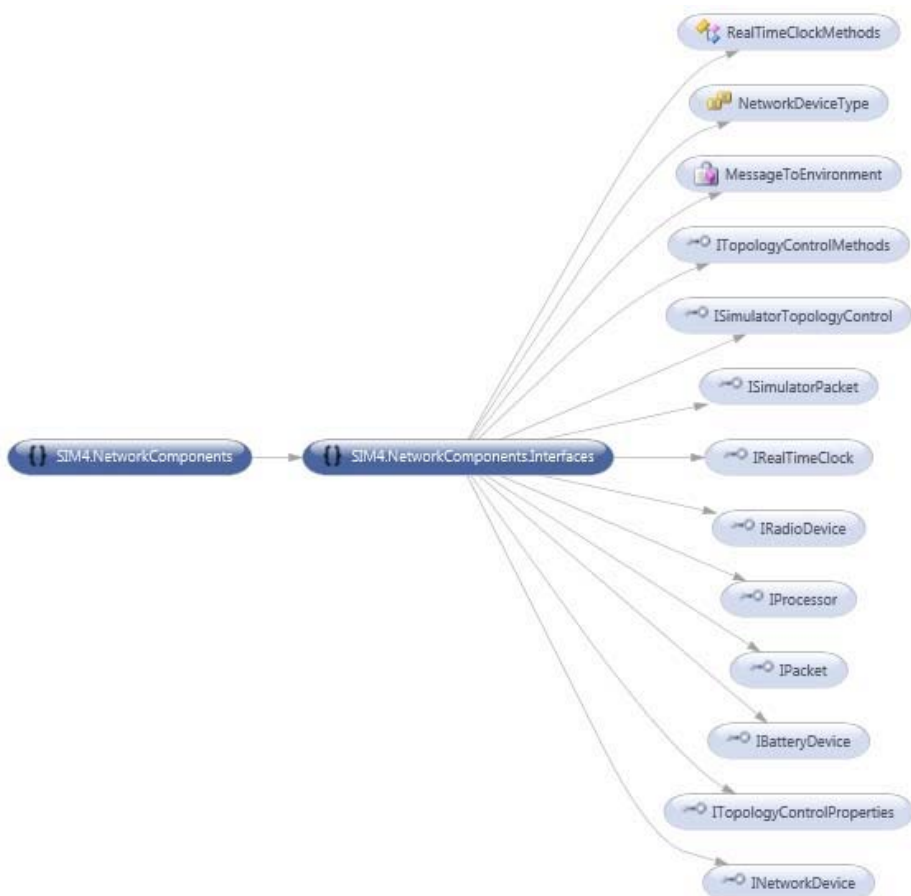


Рис.3. Інтерфейси.



Рис.4. Модель середовища та його зв'язки.

Графічний інтерфейс симулятора

Головне вікно

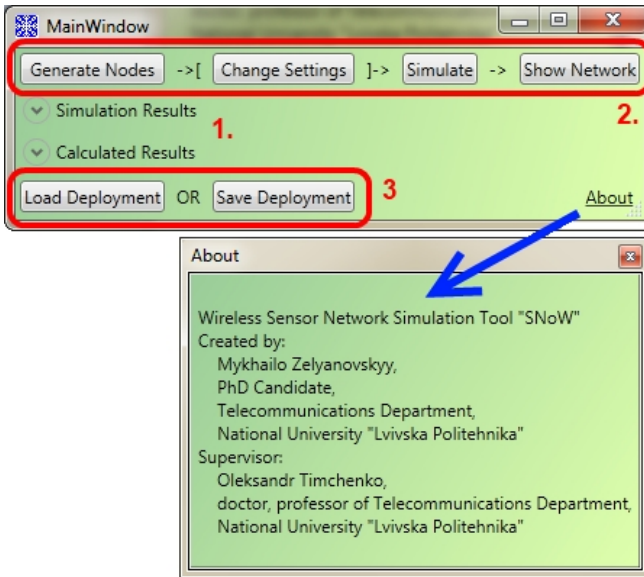


Рис.5. Головне вікно

- 1 – Елементи відображення обчислень та результатів симуляції
- 2 – Основні засоби налаштування та контролю симуляції (зліва-направо): генерування мережі, налаштування експерименту, запуск симуляції, відображення комунікаційного графа;
- 3 – збереження поточної та завантаження існуючої конфігурації розташування;

Вікно генерування вузлів

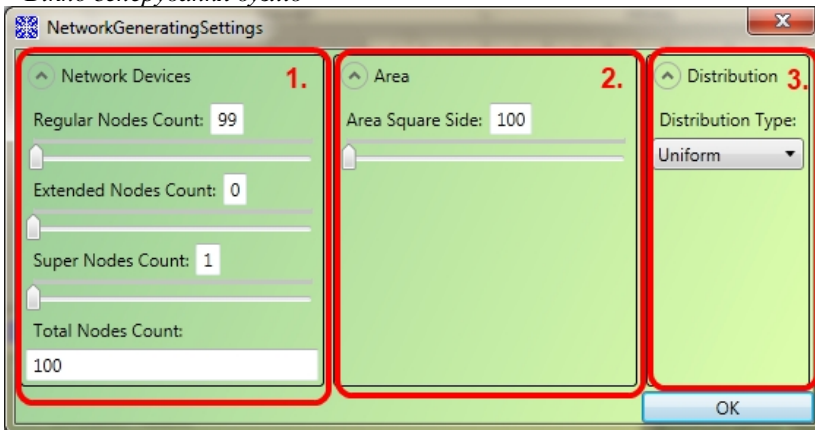


Рис.6. Вікно генерування вузлів

- 1 – Розподіл кількості вузлів по типах;
- 2 – розмір області розташування;
- 3 – розподіл вузлів на площині;

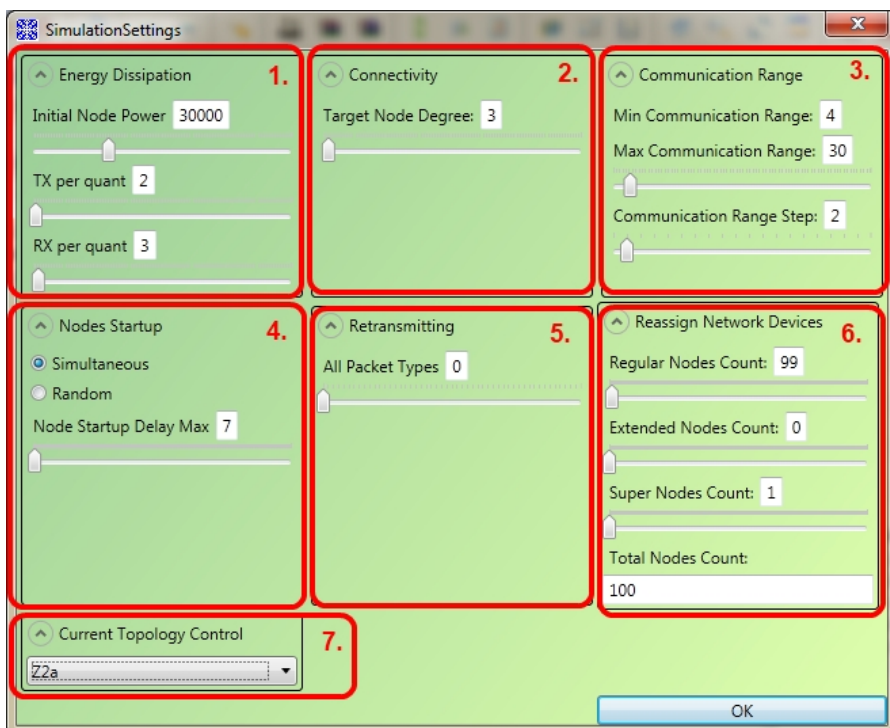


Рис.7. Вікно генерування вузлів

- 1 – параметри енергоспоживання;
- 2 – цільова зв'язність;
- 3 – параметри області покриття для вузла;
- 4 – тип старту вузлів;
- 5 – параметри повторної передачі;
- 6 – зміна розподілу вузлів по типах;
- 7 – вибір методу побудови;

Продуктивність симулятора

У табл.1 наведено тривалість 100 кроків симуляції для експериментів з різною кількістю вузлів. Розмір сторони квадратної області розташування вузлів змінювався для збереження однакової густини розташування. Сторону квадратного регіону обчислюємо за формулою:

$$r = \sqrt{N/q} , \quad (1)$$

де N – множина вузлів, r – сторона квадратної області розташування вузлів, q – густина розташування вузлів; густину приймаємо рівну 0,01; r заокруглено до найближчого більшого числа, що є кратним 10.

Вікно симуляції



Рис.8. Вікно генерування вузлів

- 1 – вкладки знятих характеристик;
- 2 – область відображення графіків реального часу знятих характеристик;
- 3 – прогрес виконання експерименту;
- 4 – час, затрачений на симуляцію;
- 5 – управління симуляцією;

Вікно перегляду комунікаційного графа

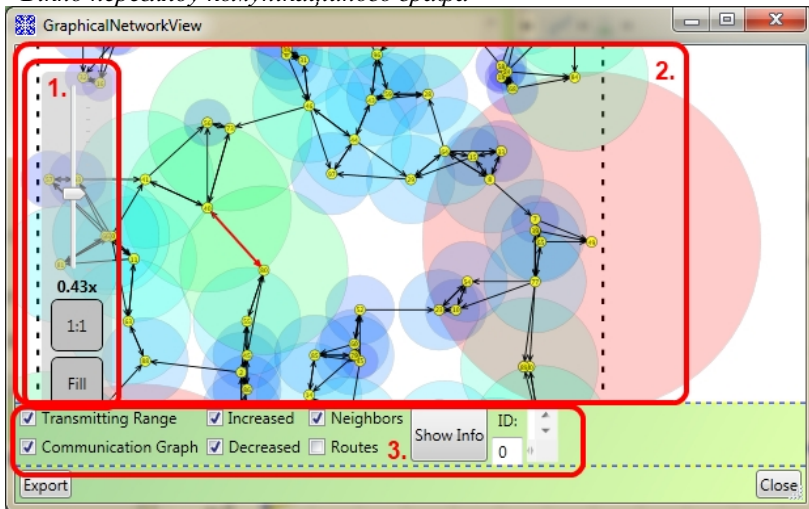


Рис.9. Вікно генерування вузлів

- 1 – управління масштабуванням;
- 2 – область відображення комунікаційних графів;
- 3 – управління відображенням.

Було проведено ряд експериментів для мереж різного розміру, що мали на меті визначити продуктивність симулятора на прикладі методу контролю топології K-NEIGH. Проводились експерименти з вимкненим (послідовна симуляція), частково увімкненим та увімкненим розпаралеленням процесів у ядрі симулятора. У табл. Наведено отримані дані.

Таблиця 1.

Час виконання симуляції у різних режимах роботи симулятора

Розмір мережі	R, од.	T, послідовна, мс	T, частк. паралельна, мс	T, паралельна, мс
100	100	916	1389	914
200	140	1978	2347	1663
500	220	8343	7647	5073
800	280	20047	16726	11886
1000	320	30350	22228	17243
1200	350	42924	30352	24575
1500	390	67447	44069	39762
1700	410	88466	55174	47393
2000	450	117378	74771	66867
2500	500	186351	110318	98441
3000	550	264805	155691	142234

Послідовна симуляція тут означає використання послідовних циклів при виконанні як машин станів усіх вузлів так і при обробці (перенаправленні) повідомлень середовищем. При частковому розпаралеленні обробка повідомлень здійснюється середовищем і паралельному циклі.

Однією з можливостей підвищення продуктивності є більш проріджене зняття характеристик. В проведених експериментах зняття усіх характеристик відбувалось кожних 10 кроків симуляції при загальній кількості кроків, рівній 200. Більш часте зняття характеристик забезпечує точніші проміжні результати і відповідно згладжені графіки отриманих характеристик у часі.

Зазначимо також, що експерименти по швидкодії здійснювались в режимі відлагодження програми; виключення збору відлагоджувальної інформації дозволяє зберегти до 30% часу. Щоб перевірити це припущення, було повторно проведено експерименти для випадків, виділених у Табл.1. кольором. Було підтверджено, що при запуску програми в режимі нормального функціонування виграш коливається від 26,6% до 33,8%.

Часові затрати при експериментах в різних режимах показано на рис.10.

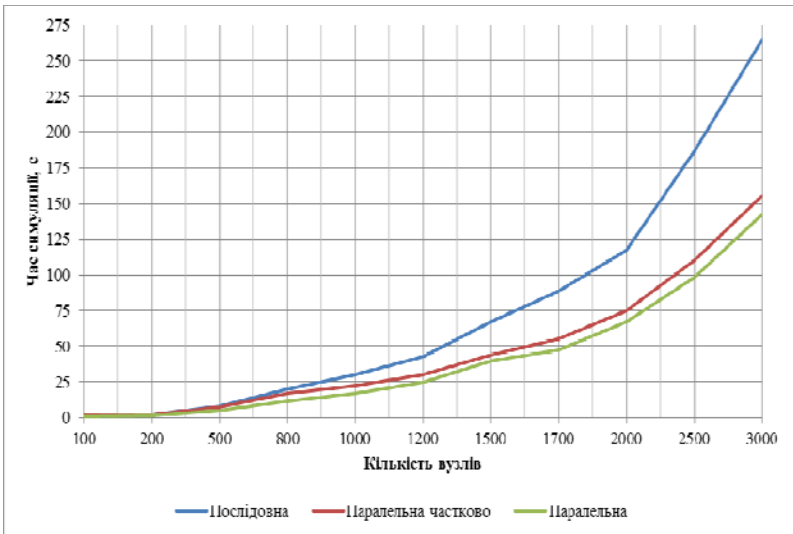


Рис.10. Результати експериментів по визначенню впливу застосування паралельних обчислень на продуктивність симулятора

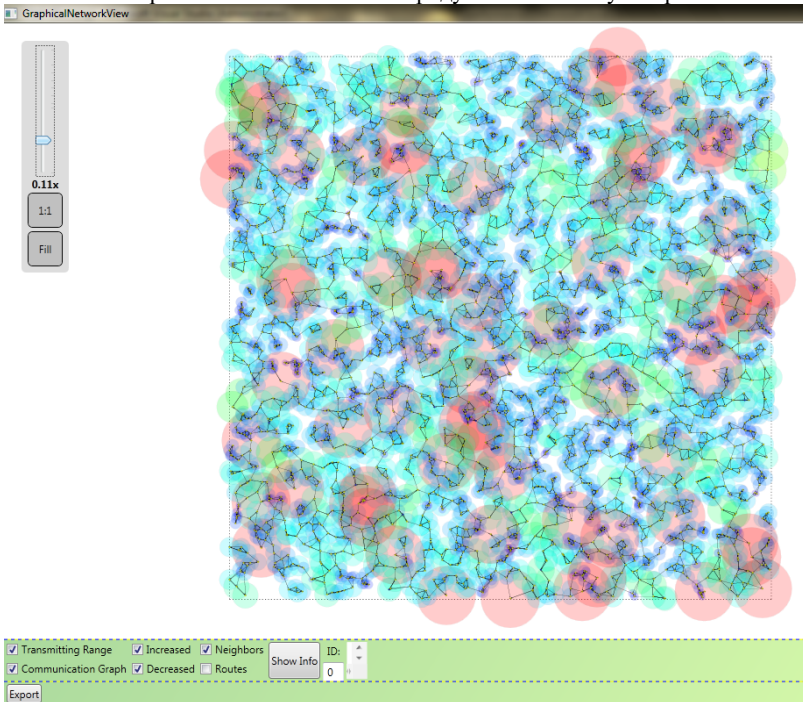


Рис.11. Комунікаційний граф після перебігу симуляції методу K-NEIGH у мережі розміром у 3000 елементів.

Як бачимо з рис.10 застосування паралельних обчислень у симуляторі дозволяє суттєво зменшувати часові затрати на симуляцію, досягаючи водночас тих самих результатів. Такий вплив зростає в міру збільшення розміру мережі. Так, симуляція методу побудови K-NEIGH для мережі з 3000 елементів засобами симулятора «Сніг» була проведена за 100 секунд (процесор Intel(R) Core(TM)2 Duo CPU P8400 @ 2.26GHz, 2267 МГц, ядер 2, логічних процесорів 2). Комунікаційний граф для такої мережі подано на рис. 11.

Висновки

Розглянуто модель бездротової мережі, адаптовану для досліджень процесу побудови та підтримки зв'язності бездротової мережі, що складається з елементів ІС. Описано будову програми-симулятора, у якій реалізована модель мережі. Описано компоненти та реалізацію симулятора.

Наведено приклад роботи з симулятором. В якості прикладу взято один з найпростіших способів побудови бездротової мережі.

Запропоновано спосіб оцінки ефективності функції призначення комунікаційного радіусу. Можливість використання такого способу підтверджено рядом експериментів.

Запропоновано способи оптимізації описаного методу побудови мережі.

1. *Зеляновський М.Ю., Алхіхі 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. ПІМЕ НАН України. - Вип.51. - К.: 2009. - С. 203-210.
2. *Зеляновський М.Ю., Тимченко О.В.* Засоби для моделювання спеціалізованих та сенсорних мереж бездротового доступу: симулятор SHAWN // Моделювання та інформаційні технології. Зб. наук. пр. ПІМЕ НАН України. - Вип.54. - К.: 2009. - С. 52-62.
3. *Зеляновський М.Ю., Тимченко О.В.* Інтелектуальна система для локальних мереж бездротового доступу: моделювання методів побудови мережі // Зб. наук. пр. ПІМЕ НАН України. – Вип.53. – К.: 2009. – С. 185-196.
4. <http://msdn.microsoft.com/en-us/library/ms752059.aspx>
5. <http://msdn.microsoft.com/en-us/library/ms754130.aspx>
6. <http://wpfextensions.codeplex.com/>
7. <http://dynamicdatadisplay.codeplex.com/>
8. <http://msdn.microsoft.com/en-us/vcsharp/aa904594>
9. <http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx>

Поступила 10.02.2011р.