

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління

Курсова робота
з курсу «Комп'ютерні мережі»
на тему:
«Моделювання сенсорних мереж за допомогою симулятора CUPCARBON»

Виконав:
студент групи КН-312
Крохмалюк Б. В.

Прийняла:
доц. каф. АСУ
Обельовська К. М.

Львів - 2019

Завдання на курсове проектування

Бездротова сенсорна мережа яка відслідковує позицію та переміщення внутрішніх мобільних сенсорів. До прикладу має бути побудована мережа на основі території Львівської Політехніки, яка виконує задачу відстеження позиції викладачів/студентів, тим самим на основі цієї інформації в подальшому можна скласти статистику відвідування, а також дізнатись в будь-який момент робочого часу де знаходиться той чи інший викладач Університету.

Зміст

Вступ	4
Характеристики об'єкту проектування	6
Огляд літературних джерел	7
Вибір та обґрунтування методу проектування	14
Моделювання системи	15
Висновки	20
Список використаної літератури	21

Вступ

Із самого початку розвитку індустрії сенсорних мереж для об'єднання різнопланових пристроїв була необхідна технологія для об'єднання усіх пристроїв у єдину мережу на базі протоколу бездротового зв'язку, котра мала бути простою та дешевою у використанні, проте, у той же час, достатньо надійною для передачі даних на відстані, відповідні до розміру окремої будівлі.

Ще кілька років тому розробники могли вибирати лише між протоколом X10 та технологіями, які використовувалися у рішеннях окремих фірм. Пристрої на базі X10 успішно використовувалися для дистанційного контролю освітлення та управління побутовими приладами. Проте, цей протокол має ряд недоліків, таких, як: низька швидкість, мала надійність передачі та слабкий захист від колізій. Це призвело до пошуку нових рішень в індустрії. Додатковим недоліком було з'єднання окремих приладів дротами.

До останнього часу не існувало бездротового стандарту, що відповідав специфічним потребам пристроїв, найважливішим параметром яких є довгострокове використання батарей та підтримка великої кількості пристроїв у мережі. їм необхідні не велика пропускна можливість, а низький рівень латентності та економічне енергоспоживання. Розробка різними фірмами спеціалізованих пристроїв на запатентованих стандартах призводить до проблем взаємодії та ускладненню переходу на нові технології.

Однак, після того як у травні 2003 року було випущено першу версію стандарту 802.15.4, у наступні два роки було створено одразу два консорціуму, котрі розробляли та використовували технології, засновані на цьому стандарті, та призначені впровадити ці технології в сферу автоматизації – Z-Wave Alliance і ZigBee Alliance.

Новітні технології бездротового зв'язку і прогрес в області виробництва мікросхем дозволили протягом останніх кількох років перейти до практичної розробки та впровадження нового класу розподілених комунікаційних систем – сенсорних мереж.

Бездротове середовище поступово входить у наше життя. Як тільки технологія остаточно формується, виробники пропонують широкий вибір продукції за прийнятними цінами, що призведе до росту попиту на неї, і до збільшення обсягу продажів. У свою чергу, це викличе подальше вдосконалення й розвиток бездротового середовища. Словосполучення «бездротове середовище» може ввести в оману, оскільки означає повна відсутність проводів у мережі. У дійсності ж звичайно бездротові компоненти взаємодіють із мережею, у якій - як середовище передачі - використовується кабель.

Характеристики об'єкту проектування

1. Основна станція-ресівер - 1 шт.
2. Сенсорні датчики - 15-20 шт.
3. Мобільний датчик - 1-2шт.
4. Стандарт передачі даних Zigbee на основі стандарту 802.15.14
5. Локація проектування: територія Львівської Політехніки

Огляд літературних джерел

Актуальність WSN

З досить великого числа прикладів використання бездротових сенсорних мереж виділимо два. Найбільш відомим є, мабуть, розгортання мережі на борту нафтового танкера компанії BP. Там за допомогою мережі, побудованої на основі обладнання Intel, здійснювався моніторинг стану судна з метою організації його профілактичного обслуговування. Компанія BP проаналізувала, чи може сенсорна мережу працювати на борту судна в умовах екстремальних температур, високої вібрації і значного рівня радіочастотних перешкод, наявних в деяких приміщеннях судна. Експеримент пройшов успішно, кілька разів автоматично здійснювалися реконфігурація і відновлення працездатності мережі.

Прикладом ще одного реалізованого пілотного проекту є розгортання сенсорної мережі на базі військово – повітряних сил США у Флориді. Система продемонструвала хороші можливості по розпізнаванню різних металевих об'єктів, у тому числі рухомих. Застосування сенсорної мережі дозволяє виявляти проникнення людей і автомобілів в контрольовану зону і відстежувати їх переміщення. Для вирішення цих завдань використовувалися моти, оснащені магнітоелектричними і температурними датчиками. Відповідне прикладне програмне забезпечення розробляється кількома американськими університетами.

Актуальність досліджень бездротових сенсорних мереж очевидна. Вже зараз у багатьох галузях їх починають використовувати. Це і моніторинг екології, авто трафіку, погоди. З удосконаленням технологій і ускладнення різних виробництв потреба в бездротових сенсорних мережах буде тільки рости. Але перш ніж впроваджувати мережі їх необхідно ретельно протестувати, тому і потрібно проводити дослідження в цій області.

Бездротові сенсорні мережі (wireless sensor networks) складаються з мініатюрних обчислювально-комунікаційних обладнань — мотів (від англ. motes — порошини), або сенсорів. Мот являє собою плату розміром зазвичай не більш одного кубічного дюйма. На платі розміщуються процесор, пам'ять — флеш і оперативна, цифро-аналогові й аналого-цифрові перетворювачі, радіочастотний прийомопередавач, джерело живлення й датчики, які підключаються через цифрові й аналогові конектори (частіше інших використовуються датчики температури, тиску, вологості, освітленості, вібрації). Набір застосовуваних датчиків залежить від функцій, що виконуються бездротовими сенсорними мережами. Живлення мота здійснюється від невеликої батареї. Моти використовуються тільки для збору, первинної обробки й передачі сенсорних даних. Головна функціональна обробка даних, що збираються мотами, здійснюється на вузлі, або шлюзі, який являє собою потужний комп'ютер. Для одержання даних вузол оснащується антеною. Але доступними для вузла виявляються тільки моти, що перебувають досить близько від нього; інакше кажучи, вузол не одержує інформацію безпосередньо від кожного мота. Проблема одержання сенсорної інформації, що збирається мотами, вирішується в такий спосіб. Моти можуть обмінюватися між собою інформацією за допомогою прийомопередавачів, що працюють у радіодіапазоні. Це, по-перше, сенсорна інформація, зчитування з датчиків, а по-друге, інформація про стан обладнань і результати процесу передачі даних. Інформація передається від одних мотів іншим за ланцюжком, і в підсумку найближчі до шлюзу моти передають йому всю акумульовану інформацію. Якщо частина мотів виходить із ладу, робота сенсорної мережі після реконфігурації повинна тривати. Але в цьому випадку, природно, зменшується число джерел інформації.

Стандарти бездротового зв'язку

Серед найбільш відомих бездротових технологій можна виділити: Wi-Fi, Wi-Max, Bluetooth, Wireless USB і відносно нову технологію – ZigBee, яка спочатку розроблялася з орієнтацією на промислові застосування.

Кожна з цих технологій має свої унікальні характеристики, які визначають відповідні області застосування. Спробуємо сформулювати вимоги, яким повинна задовольняти технологія зв'язку для її успішного застосування в промисловості. Припустимо, є якийсь промисловий об'єкт, що складається з декількох насосних електроприводів, пристроїв збору інформації з різних технологічних датчиків, наприклад, датчиків тиску, температури, витрати, у тому числі встановлених віддалено, операторського пульта і диспетчерського пункту. Управління насосами проводиться з операторського пульта, а в диспетчерському пункті проводиться безперервний моніторинг системи.

Зрозуміло, що оптимальним варіантом з точки зору простоти і зручності було б об'єднання всіх пристроїв, що беруть участь в обміні інформацією, в єдину інформаційну мережу, що працює в одному стандарті. Оскільки на промисловому об'єкті можуть бути встановлені пристрої різної складності і, відповідно, вартості, то програмно-апаратний комплекс, що забезпечує доступ кожного пристрою в інформаційну мережу, повинен бути досить дешевим. Також технологія зв'язку повинна забезпечувати необхідну дальність і швидкість зв'язку. А якщо взяти до уваги те, що промислова установка може бути доповнена новими вузлами (наприклад, ще одним насосом або пристроєм збору інформації), то від технології зв'язку потрібна можливість масштабування. Розглянутий випадок є типовим прикладом розподіленої системи управління, де кожен з вузлів, будучи інтелектуальним, виконує свою локальну задачу автоматизації, а зв'язки між вузлами є "слабкими" – в основному по мережі передаються команди оперативного управління та зміни установок регульованих змінних, повідомлення

про стан обладнання і технологічного процесу. Кожен вузол, наприклад, на базі перетворювача частоти має власні канали зв'язку з технологічними датчиками, і необхідність в передачі великих потоків даних відсутня.

Аналіз бездротових технологій показує, що високошвидкісні технології Wi-Fi, Wi-Max, Bluetooth, Wireless USB призначені в першу чергу для обслуговування комп'ютерної периферії і пристроїв мультимедіа. Вони оптимізовані для передачі великих обсягів інформації на високих швидкостях, працюють в основному по топології "точка–точка" або "зірка" і малопридатні для реалізації складних розгалужених промислових мереж з великою кількістю вузлів. Технологія ZigBee має достатньо скромні показники швидкості передачі даних і відстані між вузлами, але володіє наступними важливими, з точки зору застосування в промисловості, перевагами :

- Вона орієнтована на переважне використання в системах розподіленого мульти–мікропроцесорного керування зі збором інформації з інтелектуальних датчиків, де питання мінімізації енергоспоживання і процесорних ресурсів є визначальними.
- Надає можливість організації самостійно конфігуруючих мереж зі складною топологією, в яких маршрут повідомлення автоматично визначається не тільки числом справних або вмикання/вимикання на поточний момент пристроїв (вузлів), але і якістю зв'язку між ними, яке автоматично визначається на апаратному рівні.
- Забезпечує масштабованість – автоматичне введення в роботу вузла або групи вузлів відразу після початку живлення.
- Гарантує високу надійність мережі за рахунок вибору альтернативного маршруту передачі повідомлень при відключеннях/збоях в окремих вузлах.
- Підтримує вбудовані апаратні механізми шифрації повідомлень AES–128, виключаючи можливість несанкціонованого доступу в мережу.

Протокол ZigBee

Протокол Zigbee був спеціально розроблений для мінімізації споживання електроенергії мотами. Тому на мотах проводиться тільки первинна обробка, орієнтована на зменшення обсягу переданої інформації.

Для вироблення стандарту, у тому числі стека протоколів для бездротових сенсорних мереж, Zigbee використовував розроблений раніше стандарт IEEE 802.15.4, який описує фізичний рівень і рівень доступу до середовища для бездротових мереж передачі даних на невеликі відстані (до 75 м) з низьким енергоспоживанням, але з високим ступенем надійності.

На початку роботи відбувається ідентифікація всіх мотів, а потім вже формується схема маршрутизації. Взагалі всі моти в стандарті Zigbee за рівнем складності розбиваються на три класи. Перший і вищий з них — координатор — управляє роботою мережі, зберігає дані про її топологію й слугує шлюзом для передачі даних, що збираються всією бездротовою сенсорною мережею, для подальшої обробки. У сенсорних мережах, як правило, використовується один координатор. До наступного класу належать моти маршрутизатори, вони приймають і передають дані, а також можуть визначати напрямок передачі. І нарешті, найпростіший мот може лише передавати дані найближчому маршрутизатору. Таким чином, виходить, що стандарт Zigbee підтримує мережа із кластерною архітектурою. Кластер утворюють маршрутизатор і найпростіші моти, у яких він запитує сенсорні дані. Маршрутизатори кластерів ретранслюють дані один одному, і в остаточному підсумку дані передаються координаторові. Координатор звичайно має зв'язок з Ір-Мережею, куди й направляються дані для остаточної обробки. На сучасному етапі розвитку в протоколі Zigbee окремо моти-маршрутизатори й найпростіші моти в більшості випадків об'єднані в один, який виконує одночасно обидві функції. Приклад такої топології зображено на рисунок 1.

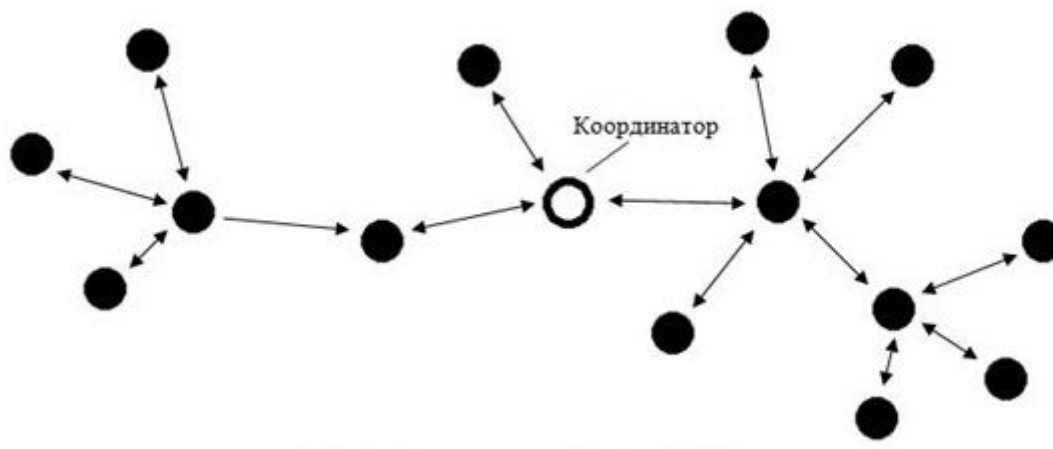


Рисунок 1 - Топологія мережі WSN

Cupcarbon симулятор

CupCarbon - це симулятор бездротової сенсорної мережі Smart City та Internet of Things (SCI-WSN). Його метою є розробка, візуалізація, налагодження та перевірка розподілених алгоритмів моніторингу, збору даних про навколишнє середовище тощо. Це не тільки допомагає візуально пояснити основні поняття сенсорних мереж і як вони працюють; він також може підтримувати вчених для перевірки їх бездротових топологій, протоколів тощо.

CupCarbon пропонує два середовища моделювання. Перше середовище моделювання дозволяє створювати сценарії мобільності та генерувати природні події, такі як пожежі та газ, а також моделювання мобільних пристроїв, таких як транспортні засоби та літаючі об'єкти (наприклад, безпілотники, комахи тощо). Друге середовище моделювання являє собою дискретне моделювання подій бездротових сенсорних мереж, яке враховує сценарій, розроблений на основі першого середовища.

Мережі можуть бути розроблені за допомогою ергономічного та легкого у використанні інтерфейсу з використанням OpenStreetMap (OSM) для розгортання датчиків безпосередньо на карті. Вона включає в себе скрипт під назвою

SenScript, який дозволяє програмувати і налаштовувати кожен сенсорний вузол окремо. З цього сценарію також можна генерувати коди для апаратних платформ, таких як Arduino / XBee. Ця частина не повністю реалізована в CupCarbon, вона дозволяє генерувати коди для простих мереж і алгоритмів.

Моделювання CupCarbon базується на прикладному шарі вузлів. Це робить його реальним доповненням до існуючих тренажерів. Вона не імітує всі рівні протоколу через складну природу міських мереж, які потребують іншої складної та ресурсомісткої інформації, наприклад, будівель, доріг, мобільності, сигналів тощо.

CupCarbon пропонує можливість моделювати алгоритми і сценарії в декілька кроків. Наприклад, може існувати крок для визначення вузлів, що представляють інтерес, за яким слідує крок, пов'язаний з характером зв'язку між цими вузлами для виконання даного завдання, такого як виявлення події, і, нарешті, етап, що описує природу маршрутизації до базової станції у разі виявлення події. Поточна версія CupCarbon дозволяє динамічно конфігурувати вузли, щоб розділити вузли на окремі мережі або приєднатися до різних мереж - завдання, яке базується на мережних адресах і каналі. Споживання енергії можна розрахувати і відобразити як функцію імітованого часу. Це дозволяє уточнити структуру, доцільність і реалістичну реалізацію мережі до її реального розгортання. Видимість розповсюдження та моделі інтерференції інтегровані і включають протоколи ZigBee, LoRa та WiFi.

CupCarbon є головним ядром проекту ANR PERSEPTEUR, який має на меті розробити алгоритми для точного моделювання поширення та інтерференції сигналів у 3D-міському середовищі.

Вибір та обґрунтування методу проектування

Вибір системи моделювання за допомогою засобів програми Cupcarbon є оптимальним, порівняно з іншими симуляторами, оскільки Cupcarbon має такі переваги:

- Наявність повної інструкції користувача, що включає 17 основних прикладів і 4 додаткових приклади
- Новий ергономічний і професійний дизайн і графічний інтерфейс користувача, заснований на новій базі JavaFX. Будь-яку дію можна скасувати та відновити.
- Багато опцій надаються користувачеві для того, щоб вибрати бажаний візуалізацію (темні карти, світлі карти, маркування вузлів та ребер, динамічна зміна параметрів моделювання, ...) і отримати найкращу і чисту візуалізацію мережі та робочого середовища.
- Дуже проста скриптова мова, заснована на SenScript, що включає інтуїтивні команди
- Можливість врахувати топологію міста (будівлі), а також радіо видимість і поширення радіо в цьому середовищі.
- Можливість включити в моделювання перешкоди сигналів на рівні РНУ для повідомлень підтвердження. Перешкоди використовують дуже точні моделі на основі альфа-стабільного розподілу
- Можливість розглянути багато радіомодулів і стандартів (802.15.4, ZigBee, WiFi і LoRa) в одному і тому ж сенсорному вузлі. Радіомодулі можна змінювати або в середовищі, або під час моделювання .
- 2D / 3D візуалізація (з OpenStreetMap)
- Генерація коду Arduino / XBee (для простих мереж)
- UART / Серіалізація / Десеріалізація між радіо і мікроконтролером

Моделювання системи

Підготовка і налаштування проекту

Моделювання системи починається зі створення нового проекту в програмному забезпеченні Cupcarbon (Рисунок 2).

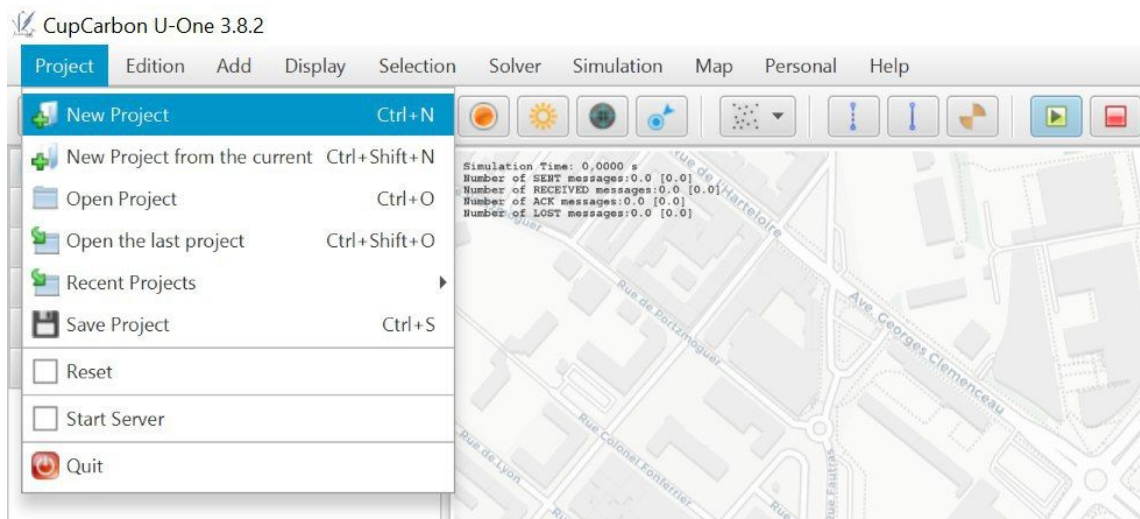


Рисунок 2 - Створення проекту

Шукаємо на карті територію Львівської Політехніки і для зручності вибираємо чорно-білу тему (Рисунок 3).

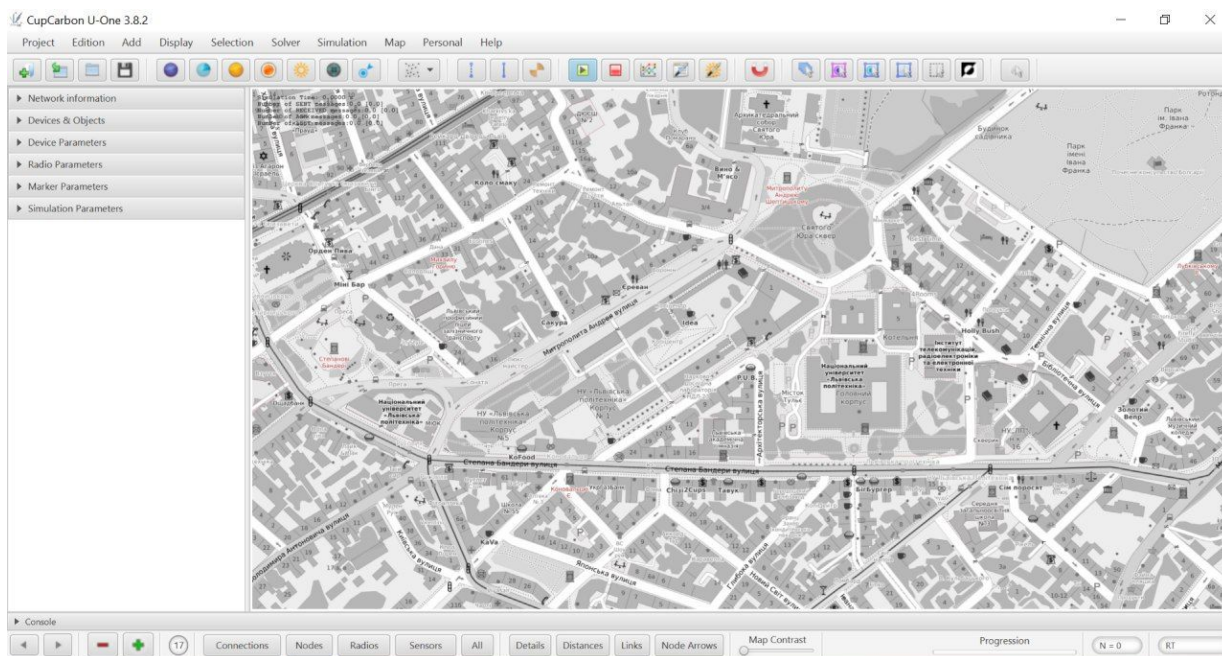


Рисунок 3 - Візуалізація проекту

Створюємо першу базову точку до якої будуть надсилати сигнали, параметри базової станції залишаємо стандартними (Рисунок 4).

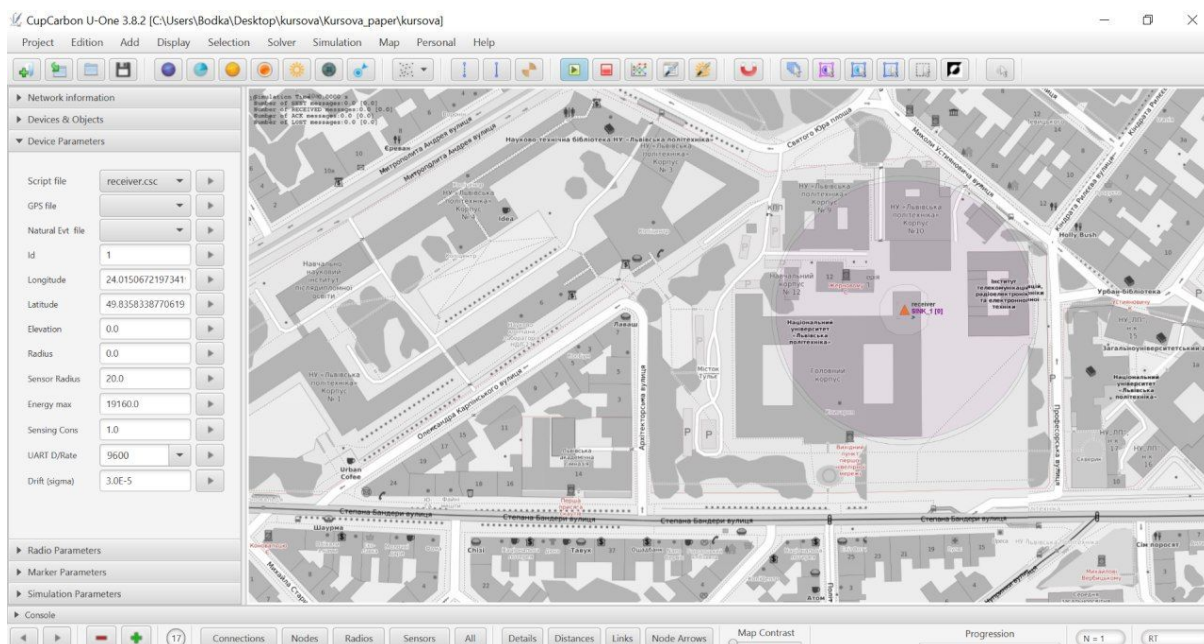


Рисунок 4 - Базова станція та її параметри

Додаємо довільну кількість сенсорів по всій території НУ “ЛП”, таким чином щоб вся територія була покрита радіусом дії сенсорів. Сенсори налаштовуємо так, щоб активний радіус покриття був 60% (Рисунок 5) (територія покриття сенсорів зображена в Додатку 3)

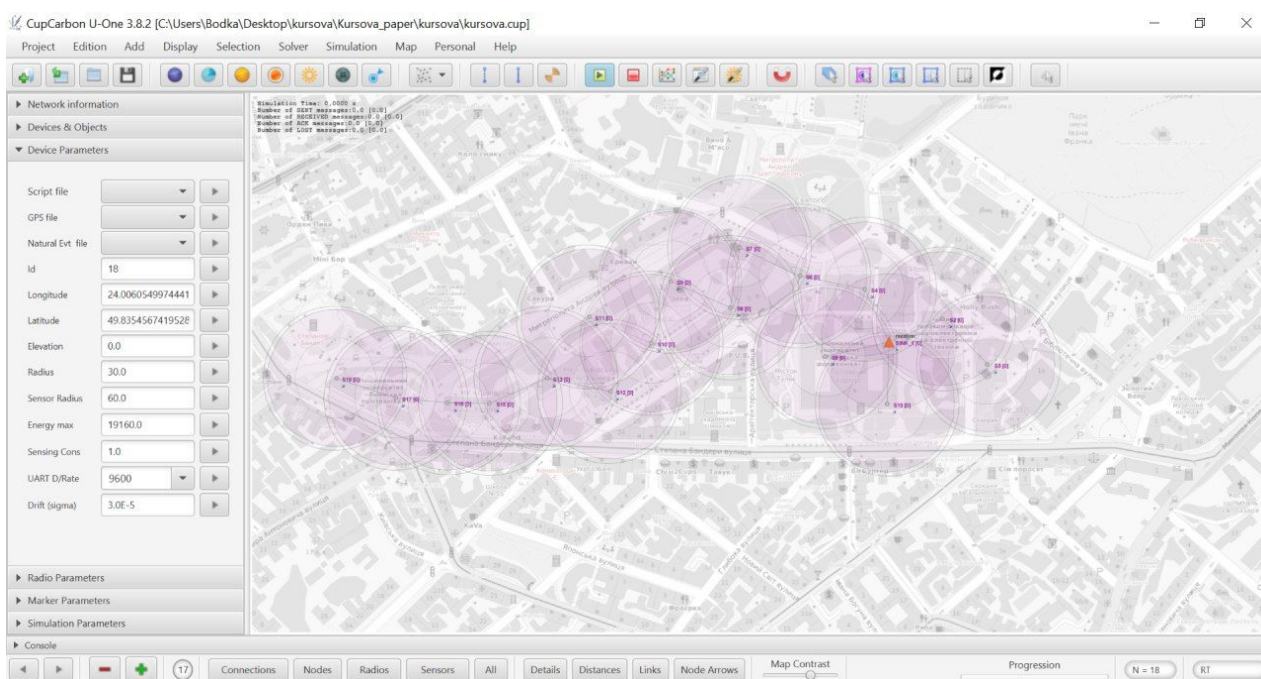


Рисунок 5 - Сенсори та їх параметри

Написання коду

Пишемо скрипти. Окремий для базової станції і один для всіх сенсорів.

Використовуємо вбудований SenScripts Editor.

Алгоритм:

1. На початку симуляції базова станція відправляє в broadcast повідомлення зі своїм ID номером.
2. Станції які мають зв'язок з базовою станцією ловлять повідомлення від неї, та встановлюють значення змінної 'nid' ID номером отриманим з повідомлення.
3. Після того як змінна 'nid' встановлена. Сенсор уже ігнорує всі наступні повідомлення які містять в собі значення наступного ідентифікатора.
4. Після цього сенсор відсилає значення вже зі своїм ідентифікатором в broadcast.
5. Пункти 2-4 повторюються допоки не охоплюються всі сенсори в мережі.
6. Увесь час сенсор прослуховує чи немає якогось мобільного сенсору в його радіусі дії, якщо є, сенсор відправляє повідомлення типу 'id#latitude#longitude' з параметрами id, x, y (локацією та ідентифікатором мобільного сенсору) наступному сенсору, id якого записані в змінній 'nid'.
7. Увесь час сенсор прослуховує чи не приходить сигнал типу 'id#latitude#longitude', коли сигнал приходить сенсор відправляє такий сигнал наступному сенсору з id = 'nid'

Код програм для базової станції (Додаток 1) та сенсорів (Додаток 2) знаходяться в розділі “Додатки”

Створення об'єкту відслідковування

Об'єкт який будуть відслідковувати сенсори (в прикладі завдання відслідковуються викладачі/студенти) у програмі Cupcarbon представлений об'єктом Mobile Sensor.

Для того щоб Mobile Sensor працював - потрібно скласти для нього маршрут, для цього позначаємо маркерами наш маршрут (Рисунок 6), після цього у вкладці “Marker Parameters” зберігаємо маршрут у файл з розширенням *.gps (Наприклад: route1.gps, route2.gps...)

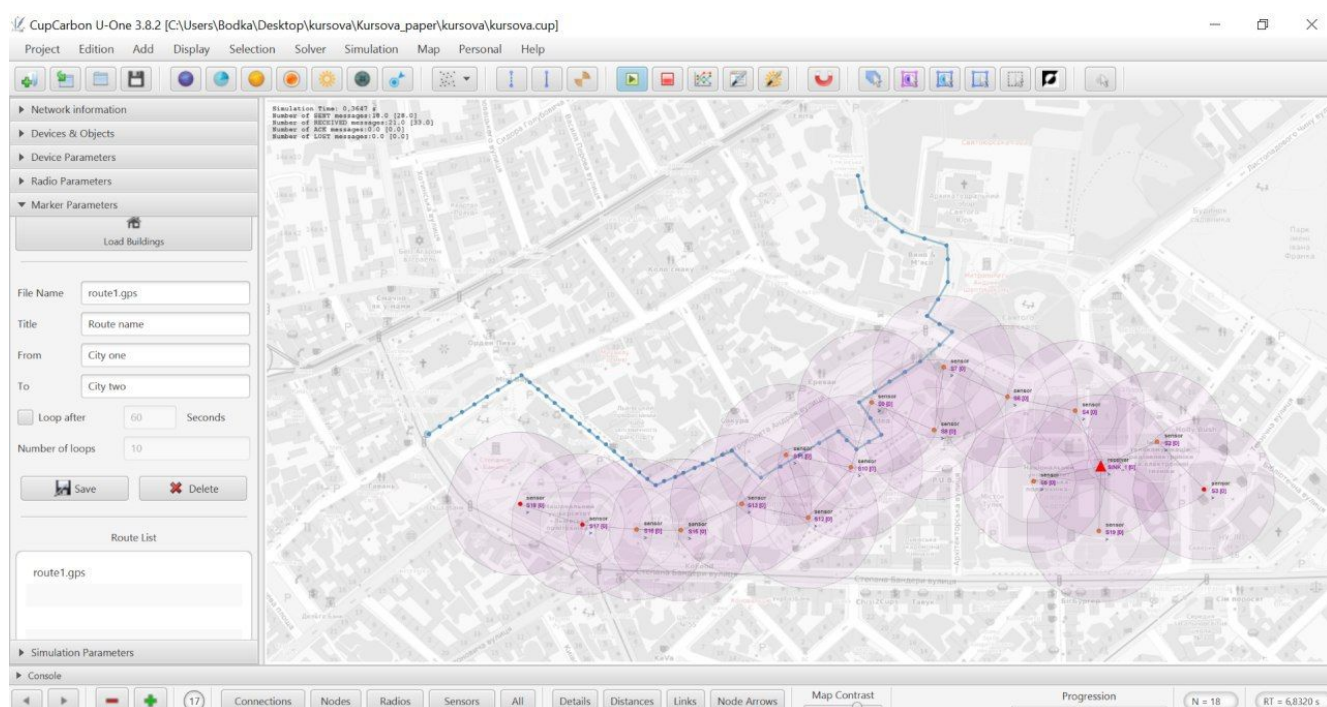


Рисунок 6 - Створення маршруту

Створюємо об'єкт типу “Mobile sensor” на карті, і прописуємо файл із збереженим маршрутом у властивостях даного об'єкту (Рисунок 7). Якщо в середині Mobile sensor світиться оранжевий кружечок - значить маршрут присвоєний даному девайсу. Для того щоб девайси могли пересуватись по маршруту у вкладці “Simulation parameters” потрібно поставити галочку у чекбоксі “Mobility/Events”

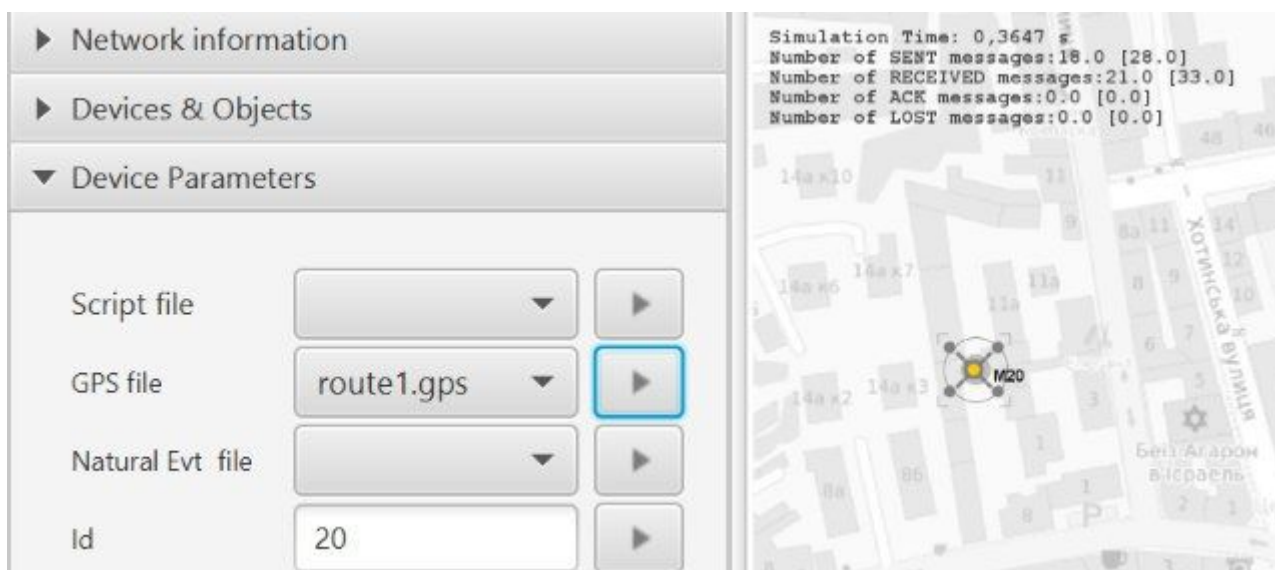


Рисунок 7 - Mobile Sensor

Створюємо ще кілька маршрутів та сенсорів і поєднуємо їх між собою, для кращого представлення симуляції. Отримуємо щось таке як на рисунку 8

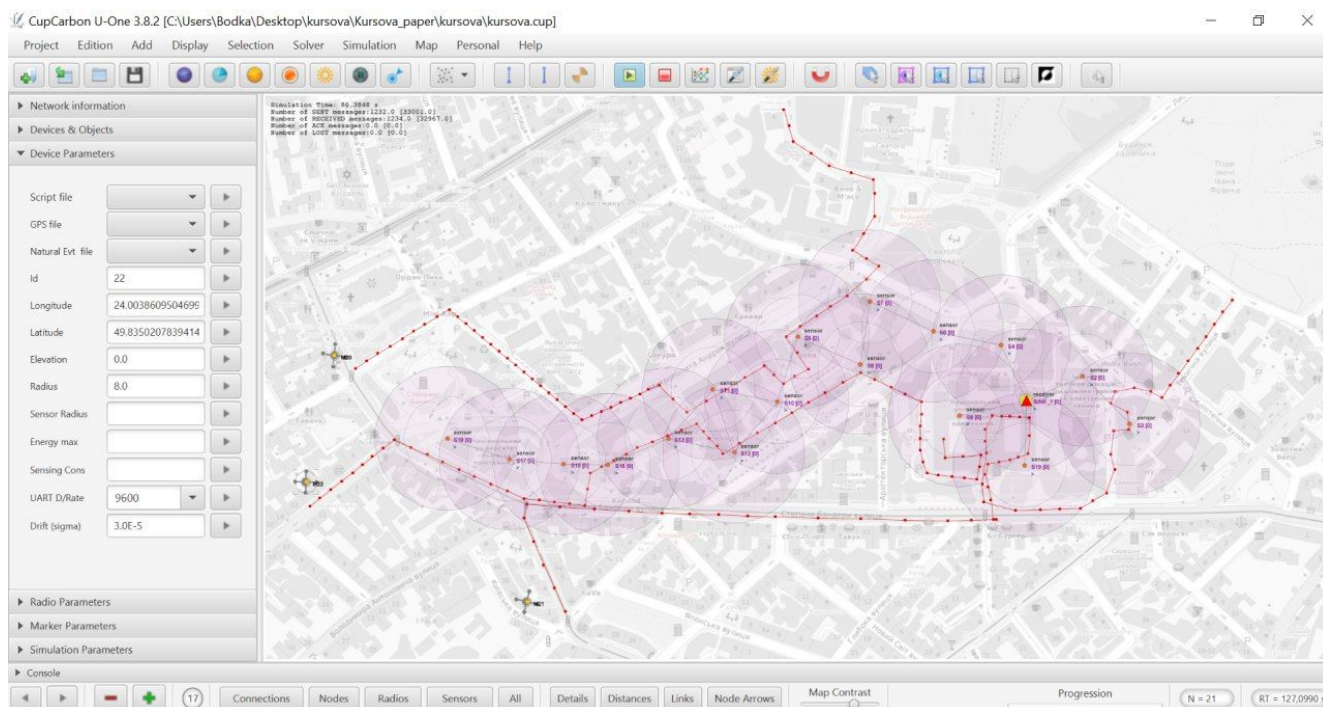


Рисунок 8 - Система готова то емуляції

Симуляція

Для правильної симуляції, в якій видно послідовність передачі даних та правильність їх відображення в консолі - використовуємо такі налаштування симулятора (Рисунок 9)

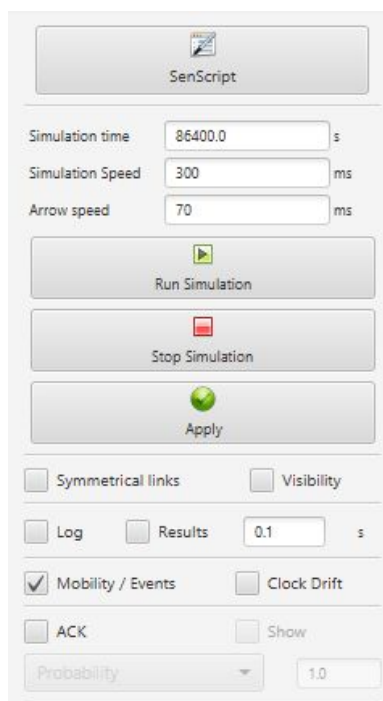


Рисунок 9 - налаштування симулятора

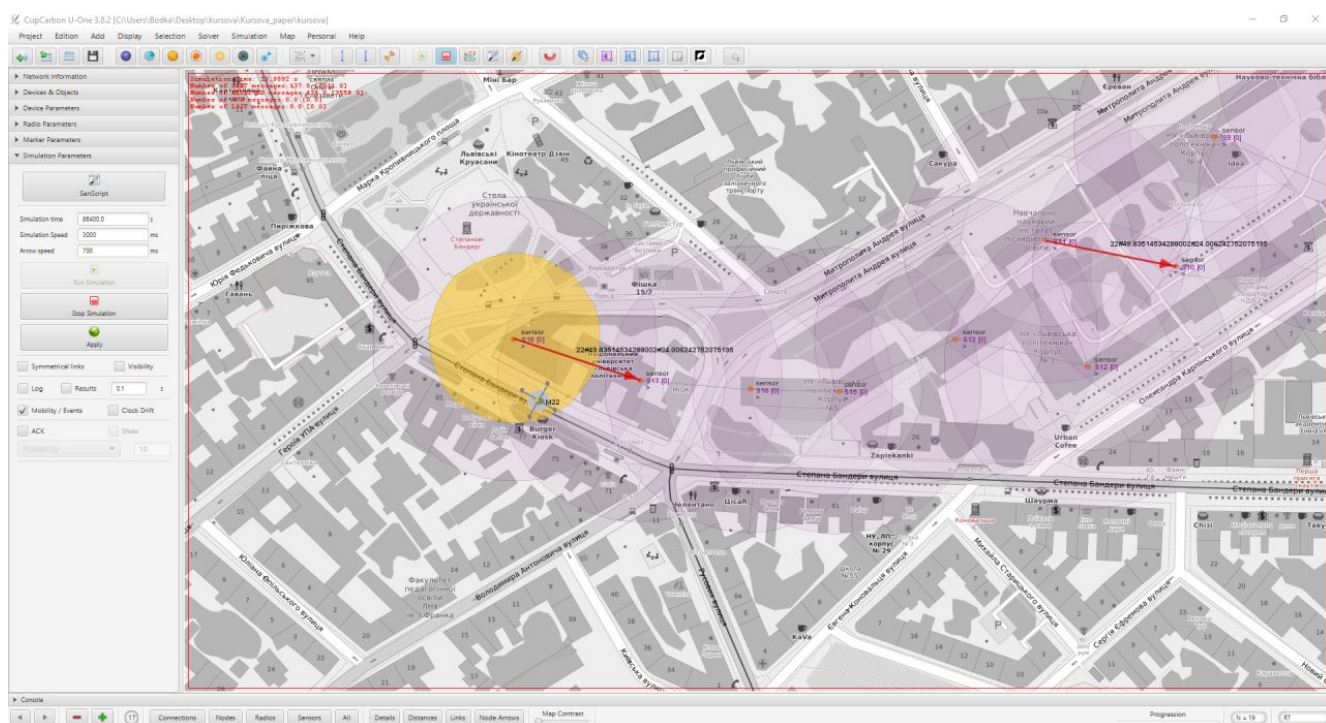


Рисунок 10 - Хід Симуляції

Результати прийнятих даних базовою станцією (Рисунок 11)

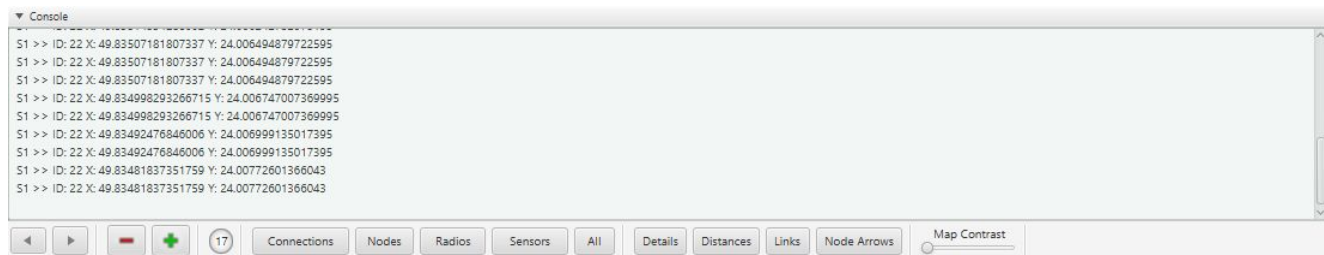


Рисунок 11 - Результати базової станції

Висновки

Перспективність розвитку бездротових сенсорних мереж очевидна. Вже зараз у багатьох галузях починають використовувати WSN. Це і моніторинг екології, авто трафіку, моніторинг погоди. З удосконаленням технологій і ускладнення різних виробництв потреба в бездротових сенсорних мережах буде тільки рости. Але перш ніж впроваджувати мережі їх необхідно ретельно протестувати, тому й потрібен хороший емулятор з чіткою візуалізацією роботи майбутньої мережі.

Список використаної літератури

1. Kufmann M., Wagner D. Drawing Graphs. Methods and Models // Berlin Heidelberg - 2001, 325 с.
2. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка визуализация, применение // С-П. - 2003, 1104 с.
3. Levis P. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications // 2009, 23 с.
4. Maneesh Varshney. Detailed Models for Sensor Network Simulations and their Impact on Network Performance // L-A. – 2006, 10 с
5. Levis P. TOSSIM System Description // Los Angeles – 2009, 43 с.
6. Karl H. and Willig A. Protocols and Architectures for Wireless Sensor Networks. // John Wiley & Sons - 2005
7. <http://cupcarbon.com/>
8. https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%B4%D1%80%D0%BE%D1%82%D0%BE%D0%B2%D1%96_%D1%81%D0%B5%D0%BD%D1%81%D0%BE%D1%80%D0%BD%D1%96_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96

Додаток 1

Приклад коду програми для базової станції

```
set iden null
loop
```

```
if($iden == null)
    atget id x
    send $x
    set iden 1
    delay 300
end
```

```
wait
read v
mark $v
```

```
rdata $v id x y
```

```
cprint ID: $id X: $x Y: $y
```


Додаток 2

Приклад коду програми для сенсорів

```

set nid null
loop

if($nid == null)
    wait
    read v
    set nid $v
    atget id x
    send $x * $nid
end

dreadsensor s
if($s==1)
    getinfo p
    send $p $nid
    cbuffer
    delay 300
end

read v
set a 0
set b A
conc a $v $b

if($a!=A)
    cprint $nid
    if($v!=0)
        cprint $v
        send $v $nid
    end
end
delay 100

```

