

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

_____ І. А. Жуков
(підпис)

«_____» _____ 2020 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

ЗА СПЕЦІАЛЬНІСТЮ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

Тема: Прикладне програмно-апаратне забезпечення системи теледіагностики
в умовах гібридного середовища передачі даних

Виконавець: студент КС231М Казадаєв Іван Олексійович
(студент, група, прізвище, ім'я, по батькові)

Керівник: д.т.н., професор Печурін Микола Капітонович
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: _____
(підпис) Малярчук В.О.
(ПІБ)

Засвідчую, що у дипломній роботі немає
запозичень праць інших авторів без
відповідних посилань

Студент _____ Казадаєв І.О.
(підпис) (ПІБ)

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних систем та мереж

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

І. А. Жуков
(підпис)

« ____ » _____ 2020 р.

ЗАВДАННЯ на виконання дипломної роботи (проекту)

Казадаєва Івана Олексійовича

(прізвище, ім'я, по батькові)

1. Тема роботи: Прикладне програмно-апаратне забезпечення системи теледіагностики в умовах гібридного середовища передачі даних

затверджена наказом ректора від " 25 " вересня 2020 року № 1793/ст

2. Термін виконання проекту (роботи): з 05.10.2020 до 30.12.2020

3. Вихідні дані до роботи: модифікація системи теледіагностики

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1. Системний аналіз прикладного програмного та апаратного забезпечення систем теледіагностики в умовах гібридного середовища передачі даних;

2. Вибір інструментарію для моделювання та реалізації прикладного програмно-апаратного забезпечення систем теледіагностики;

3. Реалізація підсистеми збирання, обробки та збереження даних в теледіагностичній системі

5. Перелік обов'язкового графічного матеріалу:

Презентація *Power Point*

6. Календарний план-графік

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Ознайомитись з постановкою задачі дипломного проектування	22.09.20	
2	Вивчити спеціальну літературу і технічну документацію	23.09.20	
3	Провести аналіз необхідності використання засобів теледіагностування	27.09.20	
4	Спроектувати систему збору і зберігання результатів теледіагностування	28.10.20	
5	Дослідити розроблену систему	09.11.20	
6	Написати пояснювальну записку	10.12.20	
7	Підготувати графічний демонстраційний матеріал	15.12.20	

7. Дата видачі завдання « 25 » вересня 2020 р. _____

Керівник дипломної роботи

(підпис)

Печурін М.К.

(П.І.Б.)

Завдання прийняв до виконання

(підпис студента)

Казадаєв І.О.

(П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи “Прикладне програмно-апаратне забезпечення системи теледіагностики в умовах гібридного середовища передачі даних”: 90 с., 23 рис., 2 табл., 26 літературних джерел.

ТЕЛЕДІАГНОСТИКА, СЕНСОРНА МЕРЕЖА, МЕДИЧНА СИСТЕМА,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,

Мета дипломної роботи – розробити підсистему збирання і збереження результатів діагностики пацієнтів до єдиної бази.

Об’єкт дослідження – бездротові комп’ютерні мережі.

Предмет дослідження – підсистема теледіагностики на основі бездротових технологій.

Метод дослідження – реалізація підсистеми збирання і збереження результатів діагностики пацієнтів до єдиної бази та аналіз бездротової комп’ютерної мережі з використанням алгоритмів керування чергами трафіку як *FIFO*, *PQ*, *WFQ*, *MDRR*.

Дипломна робота сприяє створенню нових моделей для оцінки перерозподілу трафіку в ієрархічних мережах. Розроблена система рекомендується до використання у вищих навчальних закладах у якості прикладу реалізації сенсорних мереж.

За результатами досліджено прийнято участь у науково-практичній конференції та опубліковано наукову роботу у збірнику тез доповідей:

Коцюр А.Б., Казадаєв І.О. Підсистема теледіагностики на основі бездротових технологій// Тези доповідей наук.-практ. конф. “Сучасні тенденції розвитку системного програмування” (25-26 листопада 2020 р.). – К.: НАУ, 2020. – С. 42.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРИКЛАДНОГО ПРОГРАМНОГО ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ ТЕЛЕДІАГНОСТИКИ В УМОВАХ ГІБРИДНОГО СЕРЕДОВИЩА ПЕРЕДАЧІ ДАНИХ	13
1.1. Суть системного аналізу комп'ютерних систем та мереж із визначенням особливостей його застосування для об'єктів з закритими від випромінювання каналами зв'язку середовища передачі даних.....	13
1.2. Особливості гібридних (гетерогенних) середовищ передачі даних для комп'ютерних систем медичного спрямування.....	21
1.3. Аналіз існуючих методів побудови сенсорних мереж для медико- екологічних та промислово-технологічних потреб	31
1.3.1. Принципи організації сенсорних мереж.....	31
1.3.2. Тестування швидкості роботи мереж з wi-fi та mesh концепціями побудови	34
Висновки за розділом.....	42
РОЗДІЛ 2 ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЇ ПРИКЛАДНОГО ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ ТЕЛЕДІАГНОСТИКИ.....	44
2.1. Топологія і структура радіомережі передачі даних.....	44
2.1.1. Базові топології мереж <i>LR_WPAN</i>	44
2.1.2. Асоціація - входження вузла до складу <i>PAN</i>	45
2.1.3. Створення мережі <i>MESH-BLE</i> з топологією кластерів " <i>line-mesh</i> ".	46

2.2. Вибір стека протоколів для платформи розробки радіомережі <i>MESH-BLE</i>	51
2.3. Методи аналітичного, імітаційного і натурного моделювання	58
2.4. Моделі теорії масового обслуговування	60
2.5. Спеціалізовані системи імітаційного моделювання обчислювальних мереж	62
2.6. Вибір інструментарію для реалізації структури апаратного забезпечення захищеної від випромінювання частини системи теледіагностики.....	64
2.7. Вибір реактивного веб-фреймворку для розробки інтерфейсу системи	67
Висновки за розділом.....	70
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ЗБИРАННЯ, ОБРОБКИ ТА	
ЗБЕРЕЖЕННЯ ДАНИХ В ТЕЛЕДІАГНОСТИЧНІЙ СИСТЕМІ.....	71
3.1. Описання розгортання системи в медичному закладі.....	71
3.2. Розробка інтерфейсу системи	78
3.3. Розробка програмного функціоналу підключення смарт-браслету за допомогою технології <i>Web Bluetooth</i>	79
3.4. Розробка функціоналу для динамічного отримання даних зі смарт- браслету (пульсоксиметра) та їх аналіз	82
Висновки за розділом.....	84
ВИСНОВКИ	85
СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
Додаток А	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>ATM</i>	–	<i>Asynchronous Transfer Mode</i>
<i>BLE</i>	–	<i>Bluetooth Low Energy</i>
<i>LR WPAN</i>	–	<i>Low Rate Wireless Personal Area Network</i>
<i>MDRR</i>	–	<i>Modified Weighted Round-Robin Algorithm</i>
<i>WFQ</i>	–	<i>Weighted Fair Queuing</i>
БКП	–	багатоканальний пристрій
ЛКМ	–	локальна комп'ютерна мережа
ОКП	–	одноканальний пристрій
ОС	–	операційна система
ПК	–	персональний комп'ютер

ВСТУП

Низькошвидкісна персональна радіомережа *Low Rate Wireless Personal Area Network – LR WPAN* (що має назву *BLE mesh*), на відміну від локальних радіомереж *WLAN* що не вимагає інфраструктури, пропонує реалізацію малогабаритних, енергоефективних, недорогих рішень для широкого спектру практичних завдань автоматизації і збору інформації, які вимагають контролювати стан безлічі датчиків або виконавчих пристроїв, розміщених на великих відстанях, і до яких складно або небажано підводити дротяні лінії. *BLE mesh* - це розподілена мережа, що здатна до самоорганізації, яка об'єднує між собою за допомогою радіоканалів в єдину мережу тисячі датчиків і виконавчих пристроїв, що не вимагають передач великих об'ємів інформації, - аби вони були простими, дешевими, з наднизьким споживанням енергії і з нескладним механізмом підключення до мережі. Економічна, автономна і надійна мережа *BLE mesh* зі швидкістю передачі 250 Кбіт/с на базі стандартів *LR_WPAN* і *DQDB_R* забезпечує автоматизацію виробництва і збір даних від різних типів датчиків.

Радіомережі *BLE mesh* є зручною основою для систем домашньої та промислової автоматизації, медичних систем моніторингу і охоронних комплексів при побудові надійних мереж збору/передачі даних і управління, у тому числі в операційному просторі тунелів шахти на відстанях в десятки кілометрів.

- безпроводні системи моніторингу і автоматизації шахт;
- зняття свідчень з біометричних датчиків персоналу шахти;
- зняття свідчень з газоаналітичних датчиків метану і вуглекислого газу;
- управління вентиляцією і кондиціонуванням шахти;
- передача аварійних сигналів про пожежі, вибухи і перевищення ПДУ газу.

- контроль температури і вібрації гірського устаткування шахт;
- передача даних тільки при виході параметрів за заданий діапазон.
- опитування протипожежних і охоронних датчиків.

Впровадження подібних систем можливе у різних галузях:

1. Безпроводні системи управління рухом міським транспортом у складі інформ-табло: датчики визначення місця розташування об'єктів, що рухаються, для моніторингу за міським транспортом

2. Забезпечення безпеки будівлі/території:

- датчики реєстрації руху різного типу;
- відео камерами спостереження;
- контролю появи/виходу відвідувача або співробітника з

радіоелектронним пропуском.

3. Системи автоматизації будівлі "розумний будинок":

- управління доступом і освітленням у будівлях, включення дверей, кондиціонерів;

- комерційний облік для ЖКГ спожитого тепла, електроенергії, газу і води (збір свідчень лічильників);

- контроль і управління теплопостачанням, освітленням і вентиляцією в комунальному господарстві;

- радіочастотне дистанційне керування аудіо/відеоапаратами (телевізорами і телеприставками);

- системи пожежної безпеки і автоматичної пожежогасінні;

- системи управління ліфтовим устаткуванням;

- опитування аварійних датчиків.

4. Промислове управління і моніторинг:

- збір даних про процеси, стан верстатів і агрегатів;

- датчики вібрацій для промислового устаткування;

- управління технологічним устаткуванням;

- передача інформації від об'єктів (конвеєрів, роботів), що рухаються;

- передача інформації від об'єктів під високою напругою або у вибухо- і

пожежонебезпечних умовах.

5. Індивідуальне медичне діагностичне устаткування: опитування датчиків, розміщених на тілі хворих, що пересуваються по території лікарні і санаторію.

Технологія радіомережі збору і передачі вимірювальних даних від датчиків орієнтована на передачу за розкладом невеликих об'ємів інформації від безлічі джерел. Вузли і сенсори мережі *BLE mesh* мають тільки акумуляторне живлення і більше 99 % часу знаходяться в сплячому режимі з низьким рівнем споживання енергії, забезпечуючи тим самим тривалішу роботу акумуляторів. Розвиток і модернізація мережі *BLE mesh* не вимагають заміни вже створеної апаратури.

Стандарт *IEEE 802.15.4* дозволяє створити *LR_WPAN* із швидкістю передачі даних 250 кбіт/с при радіусі дії вузлів мережі до 500 м. Стандарт орієнтований на розробку дешевих радіомодулів, що працюють від автономного енергоживлення з низьким споживанням. Використовуються короткі 16 байт контейнери, кадри ≤ 118 байт і пакети даних ≤ 240 байт. Передача коротких пакетів – відмінна риса систем управління, моніторингу і збору даних від сенсорів.

Проблеми розробки мережі радіодатчиків *BLE mesh* і методи їх вирішення.

Мережі *BLE mesh* мають специфіку їх розгортання і технічного обслуговування порівняно з іншими мережевими технологіями. Урахування цієї специфіки спрощує інженерний процес розробки мереж. Проблеми впровадження мереж *BLE mesh* усуваються під час їх розробки з ухваленням нових технічних рішень:

- масштабування мережі;
- маршрутизація;
- асоціація;

- фрагментація;
- форматування;
- відмовостійкість;
- локалізація;
- зручність розгортання.

З огляду на це велика необхідність у розробці спеціалізованих програмних засобів для проведення моніторингу звертань користувачів по мережі, що утворює передачу пакетів даних – дана передача називається трафіком.

При цьому ключові характеристики, що поєднані з керуванням трафіком, можуть належати до наступних категорій:

- 1) орієнтовані на трафік;
- 2) орієнтовані на ресурси.

Завдання, що орієнтовані на керування трафіком, містять у собі аспекти поліпшення інформаційних потоків. У моделі "оптимальних зусиль" для Інтернет-сервісу ключове завдання керування трафіком містить у собі: мінімізацію втрат пакетів і затримок, оптимізацію пропускну здатності й узгодження найкращого рівня послуг. У даній моделі мінімізація ймовірності втрати пакетів є найбільш важливим аспектом. Статистично задані характеристики трафіка (такі як розкид часу доставки пакетів, ймовірність втрати й максимальний час доставки) стають важливими в майбутніх диференційованих послугах Інтернет.

Одним з підходів вирішення таких проблем є оптимізація використання всіх наявних ресурсів мережі. Зокрема, бажано гарантувати, щоб субнабори мережних ресурсів не були перевантажені, у той час як аналогічні ресурси на альтернативних маршрутах недовантажені. Смуга пропускання є критичним ресурсом сучасних мереж. Отже, центральною функцією керування трафіком є

ефективне керування пропускнуою здатністю.

З огляду на вище сказане можна визначити мету, об'єкт та предмет дослідження.

Мета дипломної роботи – розробити підсистему збирання і збереження результатів діагностики пацієнтів до єдиної бази.

Об'єкт дослідження – бездротові комп'ютерні мережі.

Предмет дослідження – підсистема теледіагностики на основі бездротових технологій.

РОЗДІЛ 1

СИСТЕМНИЙ АНАЛІЗ ПРИКЛАДНОГО ПРОГРАМНОГО ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ ТЕЛЕДІАГНОСТИКИ В УМОВАХ ГІБРИДНОГО СЕРЕДОВИЩА ПЕРЕДАЧІ ДАНИХ

В даному розділі розкриваються основні можливості використання *mesh*-мереж для реалізації автоматизованих систем моніторингу стану розподілених об'єктів медико-екологічних та промислово-технологічних комплексів, які набувають останнім часом все більшого розповсюдження. Впровадження *mesh*-мереж для даних потреб супроводжується низкою проблем, які пов'язані не тільки з їх використанням, підтримкою та обслуговуванням, а також з використанням алгоритмів організації мереж і спеціалізованого програмного забезпечення для збору, накопичення, пересилання і обробки розподіленої інформації.

1.1. Суть системного аналізу комп'ютерних систем та мереж із визначенням особливостей його застосування для об'єктів з закритими від випромінювання каналами зв'язку середовища передачі даних

В умовах гібридного середовища передачі даних в останні роки набули популярності комірчасті мережі (*mesh*-мережі). Говорячи про комірчасті мережі необхідно розуміти, що практично всі бездротові мережі передачі даних можуть працювати з подібною топологією. Тобто майже будь-яка бездротова мережа може бути використана як основа.

При цьому необхідно розуміти, чим вони відрізняються від традиційних мереж, для чого вони призначені, які завдання вирішують і відповідно чому були створені.

Однако, відразу варто зазначити, що для цього в більшості випадків потрібні ті чи інші зміни (доопрацювання). Оскільки часто, коли з'являлися стандарти бездротової передачі даних про це не замислювалися. У той же час

існують стандарти, які вже починаючи з самого заснування були орієнтовані на роботу з даною топологією.

Як приклад, можна навести широко відомий *Wi-Fi*. Підтримка комірчастих мереж з'явилася в стандарті починаючи з специфікації *IEEE 802.11s* (з'явився в 2011), в той час як первісна версія стандарту була випущена в 1997 році, перші пристрої вийшли на ринок ще раніше в 1991, а попередні розробки почалися всередині 80 років.

Такі стандартні, як *ZigBee* і *6LoWPAN* (рис. 1.1) мають комірчасту структуру в своїй основі з моменту розробки. Розглянемо більш докладно принцип дії комірчастих мереж на прикладі *Wi-Fi 802.11s*.

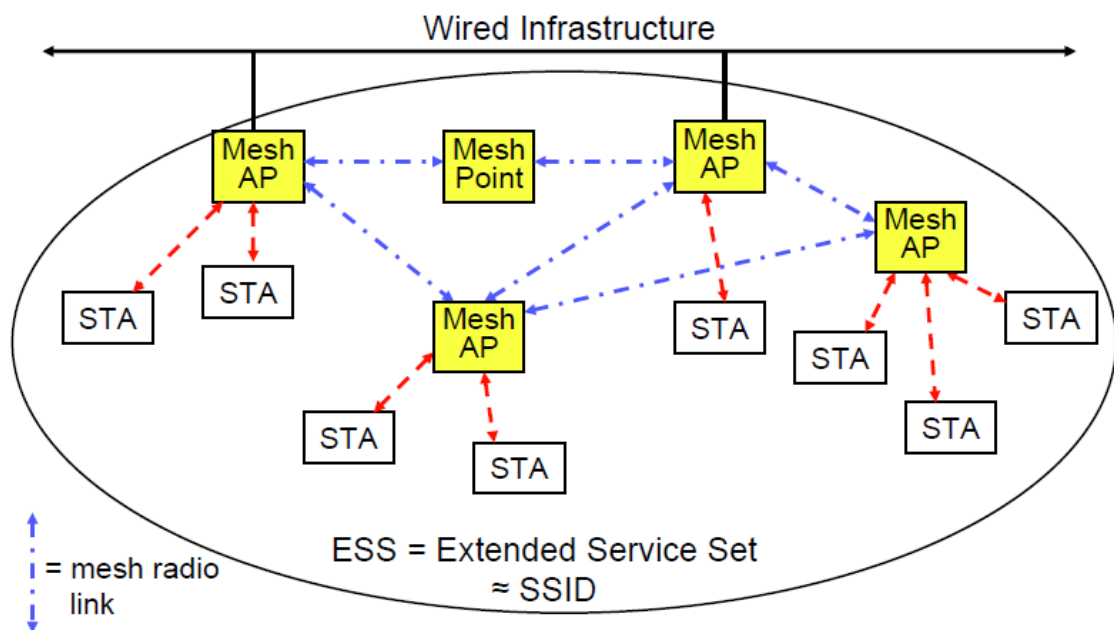


Рис. 1.1. Структура *mesh*-мережі для роботи в умовах гібридного середовища передачі даних

Візьмемо групу кінцевих клієнтів (станцій *STA*) розподілених в деякому просторі кожної з яких потрібен доступ до зовнішньої мережі. Вважаємо, що доступ до зовнішньої мережі здійснюється через дротову мережу (*Wired Infrastructure*). Традиційно кожна точка доступу (*AP*) утворює свою власну мережу топології зірка з унікальним *SSID*. До точкам доступу в свою чергу підключаються станції. Однак, радіус дії кожної точки доступу обмежений,

тому неможливо підключити всі станції до однієї точки доступу, тому необхідно встановлювати кілька точок доступу для забезпечення рівномірного і повного покриття. Але, тут виникають інші проблеми:

- ручний розподіл *STA* по *AP* довго і незручно, тому що потрібно визначити найбільш відповідну для підключення пару *STA* / *AP* по ряду факторів (якість сигналу, пропускна здатність, розподіл навантаження і ін.),

- крім того конфігурація мережі може змінитися (можуть з'явитися додаткові точки доступу і / або станції, також як число учасників мережі може і зменшитися, або ж *STA* / *AP* можуть поміняти своє розташування в просторі), що потребують додаткової переконфігурації мережі,

- при цьому кожна точка доступу утворює свою власну мережу з унікальним ідентифікатором і адресним простором, що ще більше ускладнює ручне конфігурування мережі і передачу даних між вузлами.

- *AP* не можуть мати один *SSID*, тому що будуть заважати один одному.

Тому і був розроблений стандарт *IEEE802.11s*, що усуває описані недоліки. Станції підключаються до точок доступу (з'єднання позначені червоним пунктиром), в подальшому точки доступу організовують порожнисту мережу між собою (синій пунктир) із загальним ідентифікатором *ESS*. Система автоматично визначає шлях передачі даних (здійснюється маршрутизація) серед вузлів мережі (*802.11s* працює на 2 рівні моделі *OSI/ISO*). При необхідності одна або кілька точок доступу можуть бути підключені до кабельної мережі.

При підключенні / відключенні тих чи інших вузлів відбувається перестроювання топології мережі і відповідно визначення нових оптимальних маршрутів. Управління мережею забезпечується за допомогою точок доступу, в деяких реалізаціях *mesh*-мереж для цього використовуються додаткові пристрої типу координатора / роутера, які займаються управлінням на більш високому рівні.

В рамках інших типів комірчастих мереж певні учасники мережі можуть мати відмінні назви, але принцип залишається в загальному випадку таким же.

Наприклад, в *ZigBee* є такі компоненти:

- координатор запускає мережу і керує нею. Він формує мережу, виконує функції центру управління мережею і довірчого центру (*trust*-центру) - встановлює політику безпеки, задає налаштування в процесі приєднання пристроїв до мережі, управляє ключами безпеки;

- маршрутизатор передає пакети, здійснює динамічну маршрутизацію, відновлює маршрути при перевантаженнях в мережі або відмові будь-якого пристрою. Маршрутизатор при формуванні мережі приєднуються до координатора та / або іншим маршрутизаторів, і можуть підключати дочірні пристрою - маршрутизатори і кінцеві пристрої. Маршрутизатор працюють в безперервному режимі, мають стаціонарне живлення і можуть обслуговувати «сплячі» пристрої. Маршрутизатор може обслуговувати до 32 сплячих пристроїв;

- кінцевий пристрій може приймати і відправляти пакети, але не займається їх трансляцією і маршрутизацією. Кінцеві пристрої можуть підключатися до координатора або маршрутизатора, але не можуть мати дочірніх пристроїв. Кінцеві пристрої можуть переводитися в сплячий режим для економії заряду акумуляторів. Саме кінцеві пристрої мають справу з датчиками, локальними контролерами і виконавчими механізмами.

Питання «сплячих пристроїв» в комірчастої мережі є одним з найбільш важливих, особливо для інтернету речей. Традиційно в сплячому режимі можуть перебувати тільки кінцеві пристрої. Це обґрунтовується принципами роботи мережі, зокрема автоматичною конфігурацією мережі і маршрутів за допомогою точок доступу (маршрутизаторів). Однак, існують альтернативні реалізації, які дозволяють переводити в режим сну не тільки кінцеві вузли, але і маршрутизатори. Прикладом такої мережі можна назвати *6LoWPAN*.

Режим «сплячих роутерів» в ній реалізований із застосуванням технології *Radio Duty Cycling*. У ній ній використовується принцип періодичного пробудження роутерів і перевірки наявності передачі. Велику частину часу роутер (в даному випадку приймач) знаходиться в режимі сну, але періодично

пробуджується для перевірки наявності в радіоефірі передачі. Якщо за відведений інтервал часу вікно прийому (виділено блакитним на малюнку нижче) приймач не знайшов в радіоефірі передачі, то він знову переходить в режим сну. Якщо ж їм була отримана певна посилка (чорний блок *D* (рис. 1.2)), то він відповідає, що отримав її, відправивши передавача підтверджує посилку (чорний блок *A*). У 6LoWPAN реалізована технологія фіксації фази передачі (*Transmission phase-lock*) дозволяє після визначення успішної передачі першої посилки надалі передавати дані тільки в ті інтервали, коли приймач пробуджується. Це дозволяє значно знизити енергоспоживання.

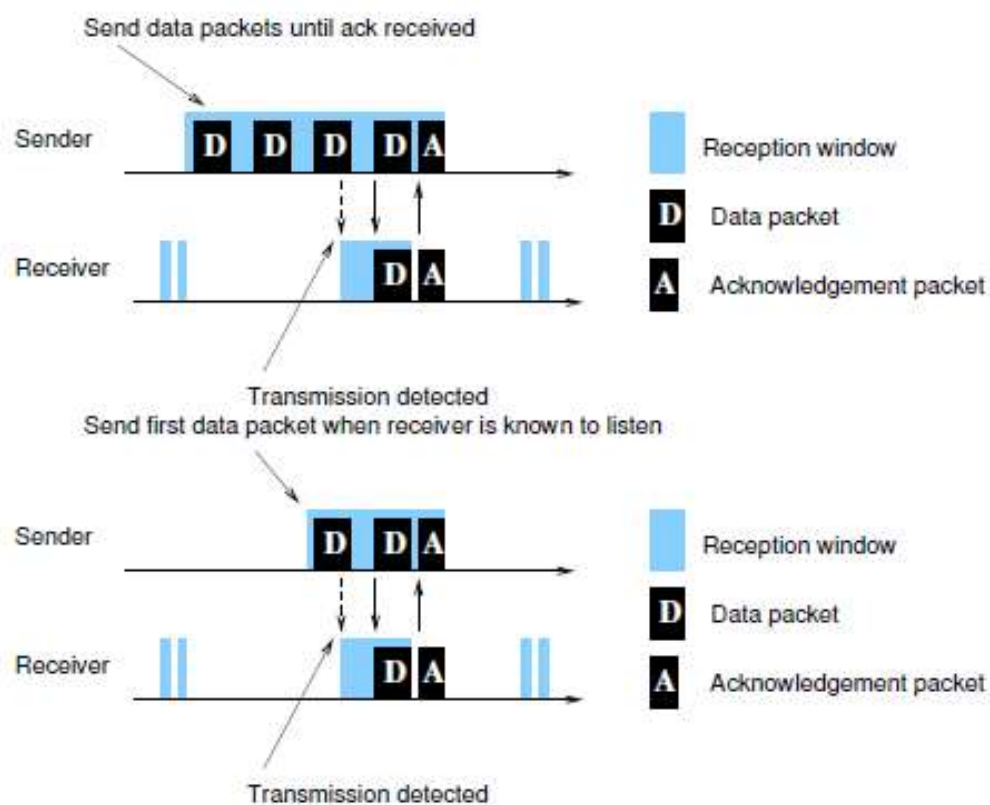


Рис. 1.2. Аналіз роботи роутерів в *mesh*-мережі

Мережа *MESH* відноситься до класу так званої *ad hoc* - систем, в яких комунікації для сенсорів організовуються тільки через вузли (роутери і ретранслятори), при цьому самі сенсори не беруть участь в мережевих операціях, передаючи повноваження своїм вузлам-батькам на вибір маршрутів і доставку пакетів даних. Розміри мережі *MESH*, як правило, не перевищують 20 км.

У мережі *MESH* група сенсорів (як стаціонарних, так і мобільних), що складається з одного вузла-батька і декількох сенсорів-нащадків, утворює топологію типу "зірка". Безліч груп сенсорів створюють складнішу просторову конфігурацію на базі радіотрактів - типу "лінійна двонаправлена шина".

У складі радіотракту завжди є вузол, що генерує суперкадри - координатор радіотракту *Radio Patch Coordinator - RPC*, що виконує функції створення і управління радіотрактом, визначення меж усіх інтервалів і розподілу *GTS* слотів між групами сенсорів за запитом з їх боку.

Наявність декількох радіоканалів (11-26) має на увазі, що у безпосередній близькості може працювати до 16-ти радіотрактів в даний момент часу (за умови організації шлюзів між ними).

Безліч радіотрактів мережі об'єднуються в *mesh*-кластер радіомережі під управлінням одного з вузлів радіотракту № 0, координатора кластера *Radio Cluster Coordinator*, що виконує функцію, - *RCC*.

Можливе формування пізніше гіпермережі на основі декількох незалежних сусідніх *mesh* -кластерів.

Організовує роботу гіпермережі з довільною топологією її координатор *Giper Net Coordinator - GNC*.

Інформаційний обмін в мережі *MESH* асинхронними пакетами заснований на циклах суперкадрів кожного з радіотрактів. Цикл суперкадру виконує одночасно і функції маяка "*beacon*", і функції слотового доступу для даних. Кожен суперкадр згідно протоколу *DQDB_R* містить:

- *PHY/MAC* -заголовок згідно стандарту *IEEE 802.15.4*;
- інтервал ізохронного доступу (слот 0) для передачі *MAC*-команд і ізохронних коротких сигналів.

У ізохронному слоті суперкадру передається інформація про розподіл *GTS* слотів: для кожного з них задається момент початку і тривалість.

- інтервал конкурентного асинхронного доступу набір з п'яти слотів 1-5, пакетів даних, що призначаються для передачі, від сенсорів вузлам-одержувачам, що маршрутизуються;

– гарантовані тайм-слоти *GTS* набір тимчасових інтервалів *GTS*, *призначених* певним сенсорам, і тільки за допомогою яких виконується доступ сенсорів до мережі. Призначення *GTS* для якого-небудь сенсора означає, що ніякий інший сенсор у цей момент не може працювати на передачу. Слоти *GTS* виділяються динамічно для асинхронних даних і фіксовано для ізохронних даних.

Під час процедур MAC асоціації доступ вузлів до *RPC* відбувається на основі механізму контролю такою, що несе із запобіганням колізіям - *CSMA/CA* (як в стандарті *IEEE 802.15.4*), тобто хто перший зайняв канал, той і працює. У цей період вузол-координатор передає радіомаяки, MAC-команди і асинхронні дані.

Структура суперкадру каналного *LLC* -рівня передбачає слоти для доступу тільки вузлів радіотракту (але не сенсорів) згідно з механізмом розподіленої черговості, тобто. в один момент інформацією можуть обмінюватися усі вузли, що отримали у своє користування слоти суперкадру.

Стандарт передбачає взаємодію вузлів не лише у рамках одного кластера, але і між сусідніми кластерами гіпермережі, для чого потрібна розвинена система адресації. Для спрощення обміну усередині кластера його *RSC* координатор привласнює кожному вузлу мережі 16-бит адресу MAB. При міжкластерній взаємодії використовуються 16-бит ідентифікатори кластерів, *GNC* -координатором, що призначаються. Формат суперкадру визначає *RSC* координатор радіотракту. Координатор – це головний мережевий вузол, що управляє передачею усіх потоків даних. Суперкадр передається з циклом 1 с *RSC* координатором, виконуючи при цьому функції мережевого маркера - кадру маяка "*beacon*" - для синхронізації і управління роботою усіх активних вузлів. Після відправки суперкадру координатор передає функції управління мережею вузлам-нащадкам, які самостійно визначають свої дії. Для передачі асинхронних даних в суперкадр вводиться спеціальний інтервал безконкурентного доступу вузлів до радіотракту, який розбитий на фіксовані тимчасові ділянки, - тимчасові слоти "*slots*", впродовж яких вузли гарантовано

зможуть відправити або отримати інформацію. Використання структури суперкадру з функцією маяка дає економічний режим роботи: після періоду гарантованого доступу слідує період "неактивності", і ніякі дані в мережі не передаються до появи чергового суперкадру.

MESH BLE дозволяє три формати обміну даними: MAC-кадри команд/даних/підтверджень, радіомаяки і суперкадри. Використання того або іншого з них визначається топологією мережі і типом пристроїв (вузол або сенсор). У топології "зірка" можливі тільки два перші види транзакцій.

MESH BLE є програмною надбудовою на базі специфікацій *IEEE 802.15.4*, *DQDB_R*, *JenNet* і *BlueTooth*, що визначають діапазон частот, формат радіосигналів і алгоритми реалізації протокольних процедур.

Мережа *MESH BLE* будується з елементів "зірка", "двонаправлена шина" і "комірка" під управлінням одного головного вузла – координатора мережі. Число сенсорів в мережі *MESH BLE* – до 30 тисяч, а число вузлів – до 6 тисяч. Відстань між двома вузлами мережі може бути. в межах 10-500 метрів при потужності передавача 1-100 мВт. Для побудови мережі *MESH BLE* в діапазоні 2.4 ГГц передбачено 16 радіоканалів шириною по 5 МГц кожен.

Мережа працює в напівдуплексному режимі: прийом і передача даних не відбуваються одночасно. У мережі *MESH BLE* дані передаються з граничною швидкістю передачі по радіоканалу 250 кБит/сек, проте реальна швидкість доставки даних на порядок нижча, т. до.:

- час передачі помітно збільшується при проходженні пакету через безліч вузлів мережі,
- цикли передач ініціюються із заздалегідь заданим періодом суперкадрів, наприклад, кожні 1 с
- окрім корисної інформації в заголовок радіопакету включені також і службові дані протоколів
- високі швидкості реально і не потрібні, так як об'єми інформації, що генерується, невеликі.

1.2. Особливості гібридних (гетерогенних) середовищ передачі даних для комп'ютерних систем медичного спрямування

Починаючи роботу з комірчастими мережами, більшість розробників задається питанням: яку мережу віддати перевагу? Пов'язано це з тим, що на ринку представлено кілька конкуруючих рішень і необхідно вибрати з них найбільш підходяще під конкретну задачу або коло завдань. Насправді питання не настільки простий, як може здатися спочатку.

Пов'язано це з багатьма причинами, наведемо основні:

- відмінності в підходах і метриках у різних мереж;
- вплив факторів технічної реалізації пристроїв;
- вплив факторів навколишнього середовища;
- вплив конфігурації мережі (і окремо взятих вузлів).

Відсутність відповідних метрик для порівняння пов'язано з історичним контекстом розвитку пористих мереж. Різні мережі розроблялися різними групами і в різний час із застосуванням різноманітних парадигм, результативний також різний (табл. 1.1).

Таблиця 1.1

Зведений аналіз типологій комірчастих мереж

Параметр	<i>ZigBee</i>	<i>Thread</i>	<i>Bluetooth Mesh</i>
Основні застосування	Освітлення, лічильники ресурсів, домашньої автоматизації	Промисловість і комерційні продукти, універсальне рішення	Освітлення, домашньої автоматизації
Основна ідея (відміну)	Перша масова чарункова мережа	<i>IPv6</i>	Легка інтеграція зі смартфонами та іншими <i>Bluetooth</i> -пристроями
Маршрутизація	повна	повна	кероване повинь
Підключення до хмари	Шлюз з перетворенням протоколу	граничний шлюз	Смартфон (тимчасово) або шлюз з перетворенням протоколу

Слід звернути увагу на особливості пористих мереж (табл. 1.1). Крім реалізації однієї з початкових задач підвищення надійності мережі в цілому, кожна мережа орієнтувалася на додаткові чинники, що визначали характер її функціонування.

Першопрохідцем став *ZigBee*, в 2003 році запропонував рішення для трьох найбільш масових застосувань: освітлення, лічильники ресурсів і домашньої автоматизації. Для виконання кожної з цих завдань були розроблені свої подстандарту, на основі яких компанії і створювали вже свої варіанти. Наприклад, пропозиції *ZigBee* були досить обмежені функціонально, так як проектувалися під конкретні моделі використання, а завдання інтеграції з зовнішніми системами нерідко представляли собою окрему проблему. В результаті рішення хоча і залишалися сумісними між собою формально, на практиці це несло безліч обмежень, і системи на *ZigBee* завжди залишалися локальними. В значній мірі усунення зазначеної проблеми і присвячений проект *6LoWPAN*, на основі якого в подальшому був побудований *Thread*.

В даному випадку розробники застосовували кардинально інший підхід - на відміну від закритого *ZigBee* практично кожна частина *Thread* розроблялася «Інженерним радою Інтернету» *IETF*, що складається з фахівців безлічі компаній. В результаті вийшов відкритий, загальнодоступний, універсальний варіант і, що не менш важливо, прекрасно сумісний з поточними рішеннями звичайного Інтернету. При цьому враховувалися питання безпеки, які з кожним роком стають все більш актуальною. Третій підхід продемонстрував *Bluetooth SIG*, що представив *Bluetooth Mesh*. На відміну від перших двох протоколів *Bluetooth Mesh* використовує технологію керованого повені (*Managed Flooding*), коли кожен вузол-ретранслятор, який прийняв пакет, відправляє його далі, якщо до цього ще не пересилав його ж.

Такий спосіб дозволяє значно знизити вимоги до ПЗ і збільшити кількість учасників мережі. Оскільки практично не потрібна додаткова обробка з боку ретрансляторів, багато пристроїв можуть виконувати цю функцію на додаток до основної. Однак, в свою чергу, цей підхід несе і додаткові обмеження, пов'язані

з підвищеною ретрансляцією пакетів учасниками мережі при щільному розміщенні.

З важливих аспектів потрібно відзначити і тип підключення до хмари. Найбільш простим і гнучким рішенням, як було сказано, є *Thread*, оскільки пристрої вже часто працюють на стандартних протоколах, адаптованих для звичайного Інтернету, зокрема *MQTT* і *COAP*, і шлюзу необхідно лише перетворити один *IP*-інтерфейс в інший, але це бачиться типовим завданням для *Linux*. *ZigBee* і *Bluetooth Mesh* використовують свої протоколи для передачі даних, не сумісні з *IP*, що веде до потреби застосовувати шлюзи, що перетворюють протоколи між різними мережами відразу на декількох рівнях моделі *OSI / ISO*. До переваг *Bluetooth Mesh* треба віднести можливість використання смартфонів в якості шлюзу як тимчасове рішення, яке, однак, може бути застосовано в багатьох побутових областях.

Питання впливу апаратного забезпечення пов'язаний з тим, що різні рішення були реалізовані на різних платформах, звідки вони історично пішли. Відповідно виникали питання в продуктивності обчислювального ядра, радіоядра і радіотракта, так як у різних виробників вони були свої. Системи на кристалі останнього покоління, як правило включають *Cortex-M4*, радіо *IEEE802.15.4* і 0,5-1 Мбайт *Flash*, дозволяють реалізувати всі три мережі на одній мікросхемі і, відповідно, тієї ж самої друкованої плати. Однак до недавнього часу практично не існувало легкого способу запустити всі три мережі на одній платформі. *TI* досі не підтримує *BLE Mesh*, *ST* не підтримує *ZigBee*, у *Silabs* сертифікований тільки *ZigBee*. Рішенням в даному випадку став *Nordic Semiconductor*, який восени минулого року сертифікував все три ніздрюваті мережі. Ймовірно, що для ряду розробок це може стати вирішальним фактором. Перед Новим роком *Nordic Semiconductor* випустив порівняльну статтю про ефективну дальності з'єднання [12], Результати якої ми розглянемо далі.

Особливо варто відзначити можливість одночасного запуску двох мереж різного типу на чіпах *Nordic* - наприклад, *Thread* + *BLE*, *Zigbee* + *BLE mesh* або

в інших комбінаціях. Досягається це за рахунок поділу часу знаходження в радіоефірі (рис. 1.3).

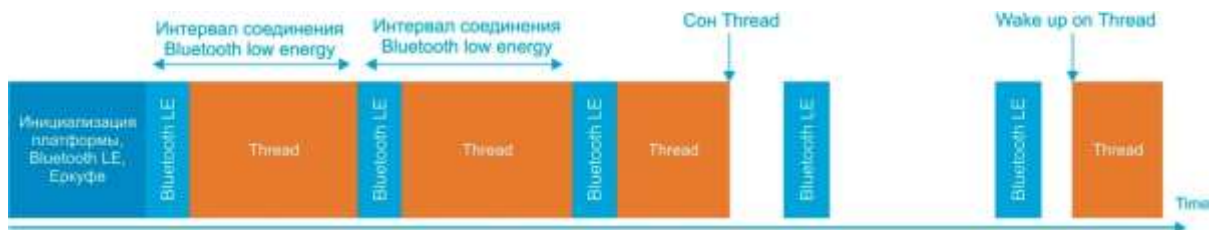


Рис. 1.3. Спільна робота *Thread* і *BLE*

Порівняння пористих мереж додатково ускладнюється необхідністю враховувати розташування їх вузлів відносно один одного і середовища передачі радіосигналу між ними. Оскільки ніздрюваті мережі призначені для роботи в неідеальних умовах (стіни, перегородки, відображення сигналу, джерела шуму і тощо) і часто складно модельованих, то результат може значно відрізнятися, навіть під час переміщення вузлів всього на 1 м. Свого часу ніздрюваті мережі були розроблені саме для зменшення впливу цього фактора, але повністю усунути його не вдалося і, швидше за все, не вийде через фізичної природи світу. Це слід враховувати при розгортанні мереж на конкретному об'єкті.

Найбільш цікавими дослідженнями пористих мереж є роботи *Ericsson* [13] і *Silabs* [14], присвячені *BLE Mesh*.

Питання дальності роботи пористих мереж наочно продемонстрував *Nordic Semiconductor* (табл. 1.2) в статті [2].

Таблиця 1.2

Ефективна дальність з'єднання в метро

протокол	Вихідна потужність, дБм		
	0	+4	+8
<i>ZigBee</i>	196	231	280
<i>Thread</i>	209	270	328
<i>BLE @ 1Mbps (mesh)</i>	248	276	345
<i>BLE @ 2Mbps</i>	238	273	333
<i>BLE @ 125kbps (long-range)</i>	-	-	756

Тут на однакових налагоджувальних платах *nRF52840-DK* були протестовані всі три мережі на трьох режимах вихідної потужності з антеною нульового посилення.

Висновки, які можна зробити в результаті аналізу таблиці:

- найбільшу дальність дає *BLE mesh*;
- *Thread* відстає зовсім небагато - на 2-5%;
- *ZigBee* показує найменшу дальність - на 16-18%.

Особливо варто відзначити режим *BLE Long Range*, що надає більш ніж дворазове збільшення дальності. На жаль, поки *Long Range* і *Mesh* в рамках *BLE* не сумісні. Однак це питання задається настільки часто, що, цілком можливо, через деякий час він буде реалізований.

У тесті на передачу 100-байтних даних можна зробити наступні висновки:

- пористі мережі з маршрутизацією показують явну залежність від кількості стрибків;
- при цьому навіть в найгіршому випадку (6 стрибків) такі мережі показують результат в 2-4 рази більше, ніж мережу без маршрутизації;
- для випадку одного стрибка *Thread* показує швидкість близько 47 кбіт / с, що становить 75% від символічної швидкості радіопередавача (62,5 кбіт / с); це говорить про вкрай високу ефективність протоколів маршрутизації;
- *Thread* показує приблизно в 2 рази більшу пропускну здатність щодо *ZigBee* при будь-яких умовах;
- пропускну здатність *Bluetooth Mesh* практично не залежить від числа вузлів і становить приблизно 2 кбіт / с;
- враховуючи, що в більшості застосувань кількість стрибків становить 2 або 3, то можна говорити про різницю в 4-6 разів для випадку *ZigBee* / *BLE Mesh* і про різницю в 8-12 разів для випадку *Thread* / *BLE Mesh*.

Затримки передачі даних для мережі з 4 стрибків. На основі графіка (рис. 1.4) можна зробити наступні висновки:

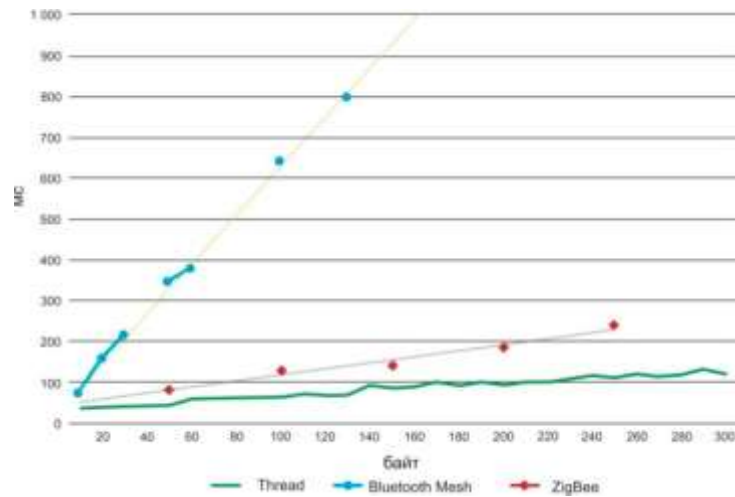


Рис. 1.4. Затримка в залежності від обсягу переданих даних

- на малих переданих даних всі протоколи забезпечують приблизно однаковий результат (~ 50 мс).
- практично всі мережі забезпечують затримки, непомітні для людини (менше 300 мс) [3];
- всі мережі забезпечують лінійну залежність затримок від розміру переданих даних;
- кращий результат показує *Thread*, побудований на технології фрагментизація 6LoWPAN;
- *ZigBee* показує хорошу ефективність, але більшу фрагментизація на рівні додатку.

Затримки в *Bluetooth mesh* значно виростають у міру збільшення переданих даних через малі розміри пакетів і сегментації, як наслідок.

Варто трохи розповісти про стиснення заголовків і фрагментизація в пакетах *Thread* (рис. 1.5). Ці властивості успадковані від протоколу *IP* і дозволяють стискати заголовні дані в пакетах для найбільш ефективного використання радіоефіру. Ефект може досягати до двох разів відносно звичайного *IEEE 802.15.4*, розмір корисного навантаження збільшується з 40 до 80 байт в кожному пакеті. Крім того, якщо розмір переданих даних більше максимально допустимого для пакета, то дані будуть автоматично розділені на фрагменти і передані по радіо, а на зворотному боці автоматично зібрані

силами *6LoWPAN* [5]. На великих навантаженнях це може бути дуже затребуване. Треба нагадати, що *6LoWPAN* працює на більш високому каналному рівні моделі *OSI*, ніж сам *IEEE 802.15.4*, на якому діє в тому числі і *ZigBee*. Це в значній мірі пояснює перевага в продуктивності.

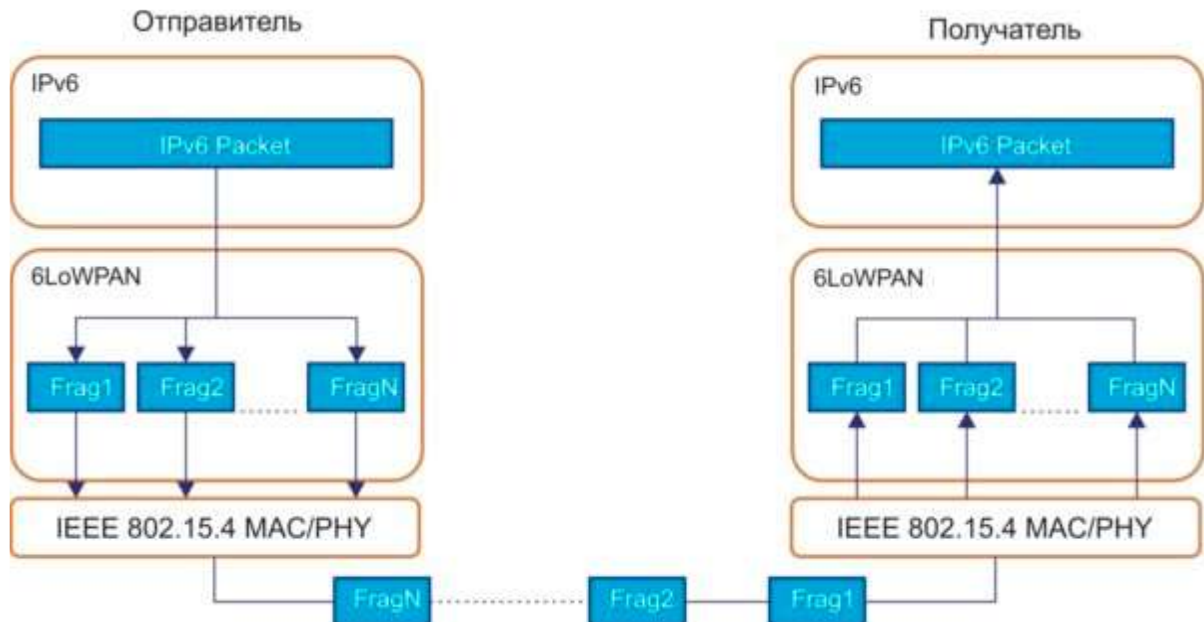


Рис. 1.5. Фрагментизація і перезбірка пакетів *6LoWPAN*

Розглянемо вплив розміру мережі та розміру переданих даних на затримки (рис. 1.6).

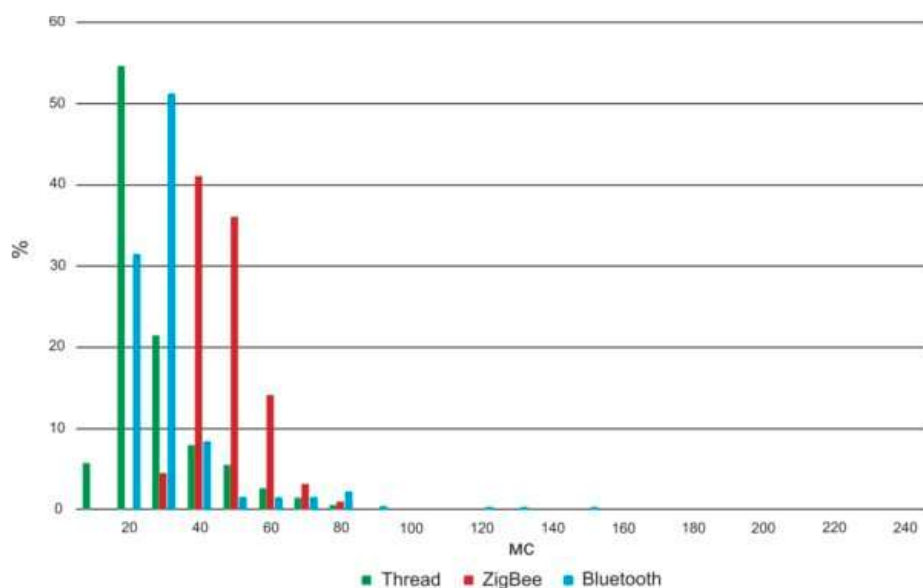


Рис. 1.6. Затримка приходу пакету для мережі в 24 вузла при посилці 5 байт (8 байт для *BLE*)

Для малих мереж (в тесті мережа складалася з 24 вузлів) і малої навантаженні (5-8 байт) всі мережі показують відмінний результат:

- типове значення: до 50 мс;
- максимальне: до 90 мс;
- практично немає втрати даних.

При збільшенні навантаження на мережу до 50 і 32 байт картинка змінюється вже значно (рис. 1.7):

- *Thread* залишається в лідерах, лише трохи змінюючи форму розподілу (більш 70% пакетів приходить до 20 мс, і все до 90 мс).
- *ZigBee* збільшує середній час приходу приблизно вдвічі - до 80 мс з максимальним значенням в 130 мс.
- *BLE Mesh* має найменш передбачуваний результат, основна частина розподілу лежить в області 40-170 мс, з піками в 20 і 220 мс.

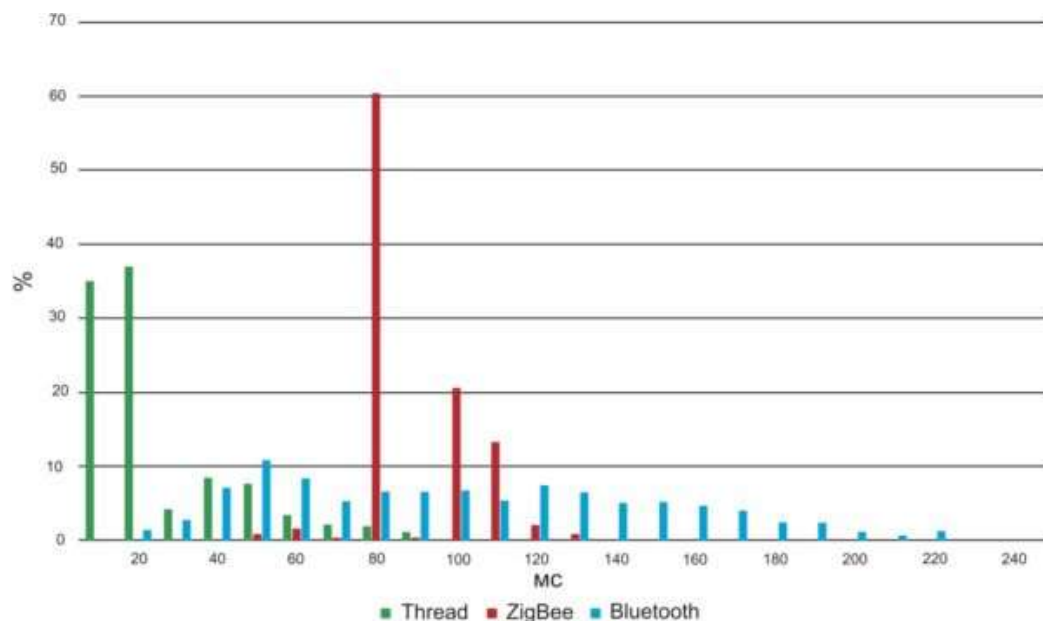


Рис. 1.7. Затримка приходу пакету для мережі в 192 вузла при посилці 50 байт (32 байта для *BLE*)

Велика мережа (192 вузла) на малому навантаженні (5-8 байт) поводить себе схожим чином (рис. 1.8):

- *Thread* забезпечує кращу затримку, трохи збільшуючи середнє значення до 40 мс і гарантовано вкладаючись в діапазон до 100 мс;

– розподіл *ZigBee* стає більш рівномірним, зберігаючи при цьому середнє значення порядку 80 мс при максимальному значенні в 190 мс.

– більшість пакетів *Bluetooth Mesh* має затримку в 60 мс, в той час як максимальне значення збільшується ще більше і може перевищувати 250 мс.

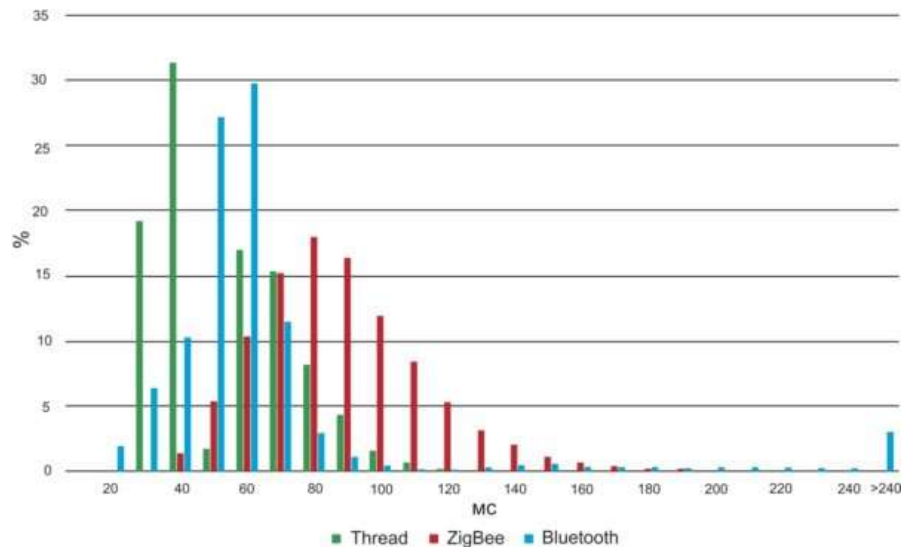


Рис. 1.8. Затримка приходу пакету для мережі в 192 вузла при поси́лці 5 байт (8 байт для *BLE*)

Найбільш цікаві результати показують великі мережі (192 вузла) на помірних навантаженнях (25 байт для *Thread* / *ZigBee* або 16 байт для *BLE Mesh*) (рис. 1.9):

– *Thread* демонструє стабільний результат в діапазоні 30-70 мс, з максимальною затримкою до 170 мс;

– *ZigBee* збільшує середній час до 100 мс і максимальне до 250 мс;

– *Bluetooth* збільшує середній час приблизно в 2,5 рази до 150 мс, а максимальне значення може досягати 800 мс.

Ефект збільшення максимального часу затримки було розглянуто в статті *Ericsson* [3], де досліджувалася велика *BLE Mesh*-мережа у 2000 вузлів, розташованих на одному поверсі. Пояснюється він тим, що занадто велика кількість ретрансляторів в мережі такого типу може заважати один одному, якщо будуть постійно передавати дані.

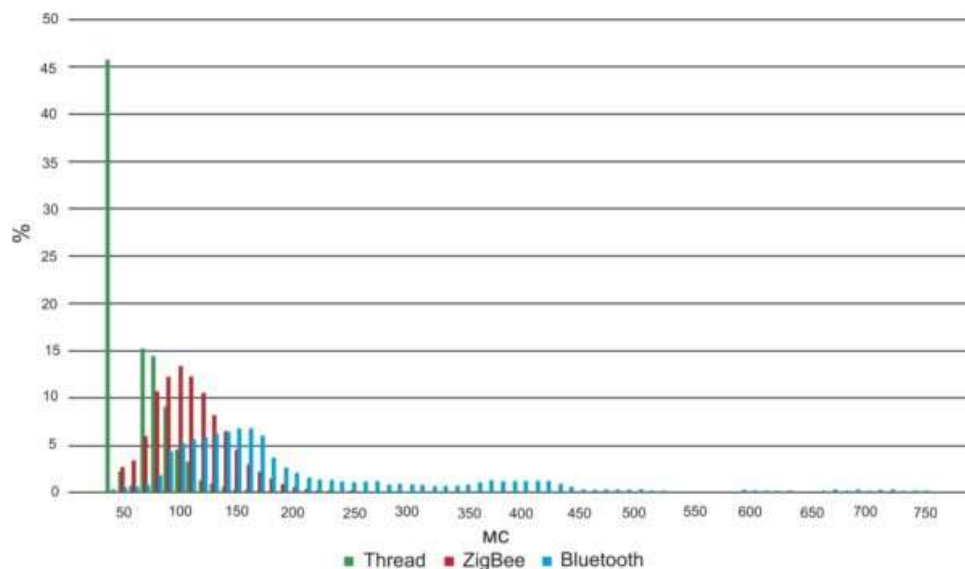


Рис. 1.9. Затримка приходу пакету для мережі в 192 вузла при посилці 25 байт (16 байт для *BLE*)

Щоб уникнути цього ефекту, необхідно реорганізувати мережу за рахунок зменшення числа ретрансляторів, оптимального їх розташування і варіації часу передачі даних (затримка на стрибок) щодо інших вузлів. В цьому випадку можна стабільно отримати затримку менше 300 мс, що і було підтверджено в лабораторії *Ericsson*.

У табл. 1.3 представлені результати впливу оптимізації мережі *BLE Mesh* на час доставки пакета. Рекомендовані значення, отримані за підсумками експерименту, дають кращий показник, - 6 ретрансляторів на 1000 кв. м і 1,5% від загального числа вузлів в мережі.

Таблиця 1.3.

Відсоток доставки пакетів за період менш 300 мс

Щільність розташування вузлів	базове рішення			оптимізоване рішення		
	низький трафік	середній трафік	високий трафік	низький трафік	середній трафік	високий трафік
низька	99,1	95,4	84,3	> 99,9	> 99,9	> 99,9
висока	97,5	88,7	69,2	> 99,9	> 99,9	99,1

Thread динамічно визначає ролі вузлів в мережі і призначає ретрансляторами найбільш ефективні вузли.

1.3. Аналіз існуючих методів побудови сенсорних мереж для медико-екологічних та промислово-технологічних потреб

1.3.1. Принципи організації сенсорних мереж

Зараз про концепцію IoT («інтернету речей») говорять скрізь. З'являється «розумна» побутова техніка, яка може підключитися до мережі (Bluetooth / Wi-Fi) по бездротовому інтерфейсу і почати розсилати повідомлення про те, що завдання з прання / готуванні їжі / кип'ятіння води завершена і непогано б щось з цим зробити. Більшість таких «розумних» пристроїв отримує живлення безпосередньо з електромережі. Але як бути, якщо хочеться отримувати інформацію від бездротового термометра і при цьому не міняти батарейку щотижня? Або мати бездротової вимикач з невеликим акумулятором для якого не знадобиться руйнувати стіни? І добре б об'єднати такі пристрої в єдину розподілену мережу, якою можна керувати віддалено і яка сама, ґрунтуючись на показаннях датчиків / сповіщувачів / лічильників, могла б приймати якісь рішення.

Спеціально для вирішення таких завдань була створена бездротова технологія ZigBee, BLE mesh, тощо. Порівняння мережевих топологій «Зірка» vs. «Mesh» зображено на рисунку 1.10.

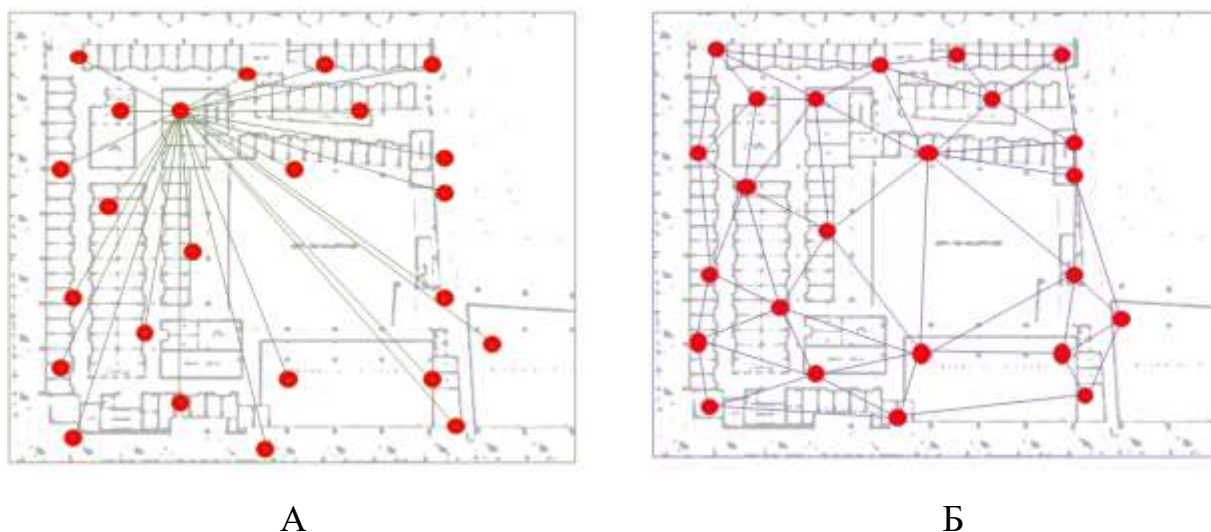


Рис. 1.10. Приклад топологій: а – зірка, б – mesh

У мережах Bluetooth і Wi-Fi мережева взаємодія йде через центральний шлюз. І якщо він вийде з ладу, то обмін даними стане неможливим. Крім цього окремі вузли можуть залишитися без зв'язку, якщо несподівано виникла перешкода на шляху проходження радіосигналу.

У мережах ZigBee і Thread надійність зв'язку підвищується за рахунок наявності надлишкових зв'язків між пристроями. Всі пристрої, які не йдуть в сплячий режим, виконують роль роутерів, які відповідальні за маршрутизацію мережевого трафіку, вибору оптимального маршруту слідування і ретрансляцію пакетів. Навіть якщо з ладу вийде пристрій, яке виступало в якості організатора мережі, ZigBee-мережу продовжить функціонувати далі. Виникнення перешкоди або перешкоди, а також вихід будь-якого з роутерів з ладу не є критичним за рахунок наявності надлишкових зв'язків. Тому з введенням додаткових вузлів, які мають стаціонарне харчування і можуть виконувати завдання роутера, мережу стає надійніше.

Типова структура мереж mesh складається з наступних елементів:

- координатор;
- роутер;
- кінцевий пристрій.

Координатор – це вузол, який організував мережу. Саме він вибирає політику безпеки мережі, дозволяє або забороняє підключення до мережі нових пристроїв, а також при наявності перешкод в радіоефірі ініціює процес перекладу всіх пристроїв в мережі на інший частотний канал.

Роутер – це вузол, який має стаціонарне харчування і отже може постійно брати участь в роботі мережі. Координатор також є роутером. На вузлах цього типу лежить відповідальність за маршрутизацію мережевого трафіку. Роутери постійно підтримують спеціальні таблиці маршрутизації, які використовуються для прокладки оптимального маршруту і пошуку нового, якщо раптом якесь пристрій вийшов з ладу. Наприклад, роутерами в мережі ZigBee можуть бути розумні розетки, блоки управління освітлювальними приладами або будь-яке інше пристрій, який має підключення до мережі електроживлення.

Кінцевий пристрій – це пристрій, який підключається до мережі через батьківський вузол - роутер або координатор - і не бере участі в маршрутизації трафіку. Все спілкування з мережею для них обмежується передачею пакетів на «батьківський» вузол або зчитуванням даних, що надійшли з нього ж. «Батьком» для таких пристроїв може бути будь-який роутер або координатор. Кінцеві пристрої більшу частину часу перебувають у сплячому режимі і відправляють управляє або інформаційне повідомлення зазвичай тільки по певній події (натискання кнопки вимикача, відкриття вікна або дверей). Це дозволяє їм довго зберігати енергію вбудованого джерела живлення. Прикладом кінцевих пристроїв в мережах ZigBee можуть бути бездротові вимикачі, що керують роботою світильників і працюють від батарейок, датчики протікання води, датчики відкриття / закриття дверей. Варто сказати,

Так як кінцеві пристрої більшу частину часу перебувають у сплячому режимі і прокидаються лише для опитування батьківського вузла на наявність повідомлень для себе, або для передачі даних, то це дозволяє економно витрачати енергію батарейковому джерела живлення.

Приклад того, як може бути реалізована система управління освітленням з використанням стандартної бібліотеки кластерів. Подробиці ми розберемо в інший раз, а зараз досить зрозуміти наступне:

- у лампочки є набір збережених атрибутів (стан - включена / вимкнена; рівень яскравості)

- вимикач може відправляти команди для зміни доступних атрибутів лампочки

- при отриманні команди, лампочка приймає зазначене стан

У бібліотеці кластерів вказується, які атрибути і команди є обов'язковими для тих чи інших пристроїв, а які опціональними. Це дозволяє реалізувати стандартний інтерфейс взаємодії між ZigBee-пристроями.

Якщо стандартних можливостей прошивки не вистачає, то можна використовувати реалізацію програмного стека ZigBee від компанії Silicon Labs - Ember ZNet PRO. Так як модулі виконані на базі мікросхеми EM357, то

перехід від стандартної прошивки до розробки власного додатка зажадає лише придбання програматора-відладчикаISA3за допомогою якого можна робити як внутрісхемний налагодження пристрою, так і налагоджувати додаток на мережевому рівні.Справа на малюнку показаний приклад того, як відображаються дані про шляху проходження пакету і його розшифровка.

Для спрощення процесу створення програми надається компоновщик додатків, який для обраної конфігурації ZigBee-пристрої генерує каркас додатка і створює функції, в яких розробник повинен дописати необхідну логіку пріложенія.Все утиліти входять в програмний пакетSimplicity Studio, Куди також входить демонстраційна версія стека Ember ZNet PRO. Тому можна прямо зараз завантажити і подивитися як це працює.

1.3.2. Тестування швидкості роботи мереж з wi-fi та mesh концепціями побудови

Для звичайного користувача, який підключається по Wi-Fi до мережі і дивиться ролики в Youtube ніякої різниці не буде, якою системою користуватися. А ось з точки зору організації нормального Wi-Fi покриття ці системи принципово різні і у кожної є як плюси, так і мінуси. Почнемо з Wi-Fi системи.

У мережі звичайних роутерів, які можуть працювати незалежно, виділяється один провідний маршрутизатор і інші стають відомими. При цьому перехід між роутерами залишається непомітним для клієнта, а з точки зору самих маршрутизаторів, клієнт переміщається з однієї стільники в іншу. Таку систему можна порівняти з стільниковим зв'язком, адже утворюється єдина локальна мережа з роутерами-трансляторами. Плюси системи очевидні: мережа можна нарощувати поступово, додаючи нові пристрої в разі потреби. Причому досить буде купувати недорогі роутери з підтримкою цієї технології. Мінус один, але істотний: до кожного роутера повинен бути підведений Ethernet-кабель і живлення.

Тобто, якщо ви вже зробили ремонт і не заклали UTP кабель, то його доведеться або протягувати по плінтусу, де це можливо, або розглядати іншу систему (рис. 1.10).

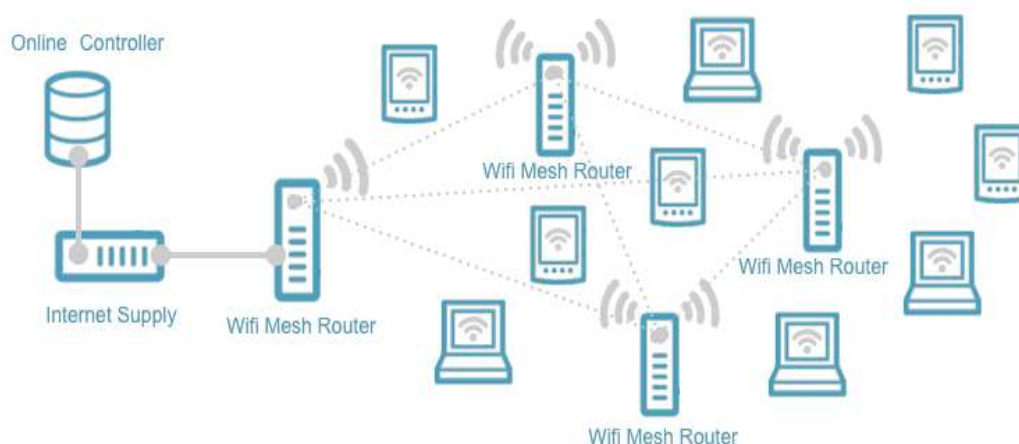


Рис. 1.10. Система Mesh

Це мережа спеціалізованого обладнання, яке також утворює мережу з декількох пристроїв, створюючи суцільне покриття Wi-Fi сигналу. Ці точки, як правило, дводіапазонні, тому працювати можна як в мережах 2,4 ГГц, так і в мережах 5 ГГц. Великим плюсом є те, що для підключення кожного нового пристрою немає необхідності тягти кабель - вони зв'язуються за окремим передавача, створюючи свою мережу і через неї передаються дані. В подальшому ці дані передаються на звичайний Wi-Fi адаптер, досягаючи користувача.

Перевага очевидна: не потрібно ніяких додаткових дротів - тільки вставити в розетку адаптер нової точки і прив'язав її до головного маршрутизатора. Але є і недоліки. Наприклад, ціна. Вартість головного маршрутизатора в рази перевищує вартість звичайного роутера, а вартість додаткового адаптера також істотна.

Виробник часто заявляє в прес-релізах про надзвичайно широкому покритті бездротовою мережею своїх пристроїв. Але це діє на відкритому майданчику без стін, що відбивають поверхонь і радіоперешкод. У реальності багато хто стикався зі зниженням швидкості і втратою пакетів в квартирах, де на смартфоні видно півтора-два десятка бездротових мереж. У тому числі тому

ефективніше задіяти ще не так зашумлений діапазон 5 ГГц.

Для простоти, будемо називати головні пристрої Wi-Fi і Mesh системи роутерами. Кожен з роутерів може бути просто тримати бездротовий пристрій. Але мені цікаво, скільки пристроїв і на якій швидкості роутер може забезпечити доступом до мережі. По першому питанню ситуація виглядає наступним чином. Кількість підтримуваних пристроїв залежати від Wi-Fi модуля. Наприклад, у Zyxel Multy X і Multy mini це буде 64 + 64 пристрої на кожен діапазон (2,4 + 5 ГГц), тобто при наявності двох точок можна підключити 128 пристроїв по 2.4 ГГц і 128 пристроїв по 5 ГГц.

Створення Mesh мережі зроблено максимально простим і наочним: для роботи достатньо мати смартфон і встановити туди додаток Zyxel Multi. І неважливо, чи буде у вас пристрій iOS або Android. Слідуючи підказкам майстра установки створюється мережа і підключаються всі наступні пристрої. Що дивно, для початкового створення мережі треба включити геопозиціонування і мати підключення до Інтернету. Так що доведеться, як мінімум, мати доступ в мережу зі смартфона.

У роутерів Keenetic ситуація виглядає дещо інакше. Кількість підключаються клієнтських пристроїв залежить від моделі. Нижче наведемо назву роутерів і можливості по підключенню клієнтів в діапазонах 2,4 і 5 ГГц.

- Giga III і Ultra II: 99 + 99;
- Giga KN-1010 і Viva KN-1910: 84 на обидва діапазони;
- Ultra KN-1810: 90 + 90;
- Air, Extra II, Air KN-1610, Extra KN-1710: 50 + 99;
- City KN-1510: 50 + 32;
- Duo KN-2110: 58 + 99;
- DSL KN-2010: 58;
- Lite KN-1310, Omni KN-1410, Start KN-1110, 4G KN-1210: 50.

Конфігурувати роутери можна як з комп'ютера, так і зі смартфона. І якщо в локальній мережі це легко реалізується через web-інтерфейс, то для смартфона існує спеціальний додаток, яке в подальшому дасть можливість

користуватися додатковими функціями, на кшталт торрент-качалки або доступу до файлів на підключеному накопичувачі по USB. У Keenetic є відмінна функція - KeenDNS, яка дозволяє при наявності сірого IP адреси підключатися до web-службам опублікованих сервісів із зовнішньої мережі. Тобто можна підключитися до інтерфейсу роутера за NAT, можна підключитися до інтерфейсу відеореєстратора або web- сервера за NAT. Але оскільки цей матеріал все-таки про мережі, то треба зазначити, що організація Wi-Fi мережі також дуже проста: чільний роутер стає провідним пристроєм, а на інших роутерах включається режим ведених адаптерів. При цьому, ведені роутери можуть створювати VLAN, можуть працювати в єдиному адресному просторі, на них може бути виставлена потужність роботи кожного бездротового адаптера з кроком в 10%. Таким чином, мережа можна розширити багаторазово. Але є одне але: для організації Wi-Fi мережі все роутери повинні бути підключені за допомогою Ethernet.

Тестування проходили окремо пристрої Zyxel Multy X + Multiy mini і Keenetic Giga + Keenetic Viva. Щоб уникнути вплив провайдера, в локальній мережі було піднято сервер перед головним пристроєм. А на призначеному для користувача пристрої був організований клієнт. У підсумку, топологія була наступною: сервер-провідний роутер-точка доступу-клієнт.

Всі тести проводилися за допомогою утиліти Iperf, що емулює безперервну передачу даних. Щоразу тести проводилися для 1, 10 і 100 потоків, що дозволяє оцінити продуктивність бездротової мережі при різній завантаженні. Емулюватися як однопотокові передача даних, на кшталт перегляду ролика на Youtube, так і многопотоковая, на кшталт роботи торрент качалки. Тести проводилися окремо при підключенні по мережі 2,4 і 5 ГГц.

Крім цього, так як пристрої Zyxel Multy і Zyxel mini можуть виступати в якості адаптера, вони були підключені по Ethernet-інтерфейсу до призначеного для користувача комп'ютера на швидкості 1000 Мбіт / с і також проведено три тести швидкості. В аналогічному тесті роутер Keenetic Vivo брав участь в якості Wi-Fi адаптера, будучи підключеним патч-кордом до ноутбука.

Відстані між точками близько 10 метрів, залізобетонне перекриття і дві стіни. Відстань від ноутбука до кінцевої точки доступу - 1 метр.

Всі дані занесені в таблицю і побудовані графіки швидкості (рис. 1.11).

На графіках ланцюжки підключення наступні:

– Zyxel mini: сервер - провід - Zyxel Multy X - беспровод - Zyxel Multy mini - ноутбук (адаптер Intel Dual Band Wireless-AC 7265);

– Zyxel Multy: сервер - провід - Zyxel Multy X - беспровод - Zyxel Multy X - ноутбук (адаптер Intel Dual Band Wireless-AC 7265);

– Keenetic Wi-Fi: сервер - провід - Keenetic Giga - провід - Keenetic Viva - ноутбук (адаптер Intel Dual Band Wireless-AC 7265);

– Keenetic підсилювач: сервер - провід - Keenetic Giga - беспровод - Keenetic Viva (як репитера) - ноутбук (адаптер Intel Dual Band Wireless-AC 7265);

– Keenetic адаптер: сервер - провід - Keenetic Giga - беспровод - Keenetic Viva (в режимі адаптера) - провід – ноутбук;

– Zyxel mini адаптер: сервер - провід - Zyxel Multy X - беспровод - Zyxel Multy mini - провід – ноутбук;

– Zyxel Multy адаптер: сервер - провід - Zyxel Multy X - беспровод - Zyxel Multy X - провід – ноутбук.

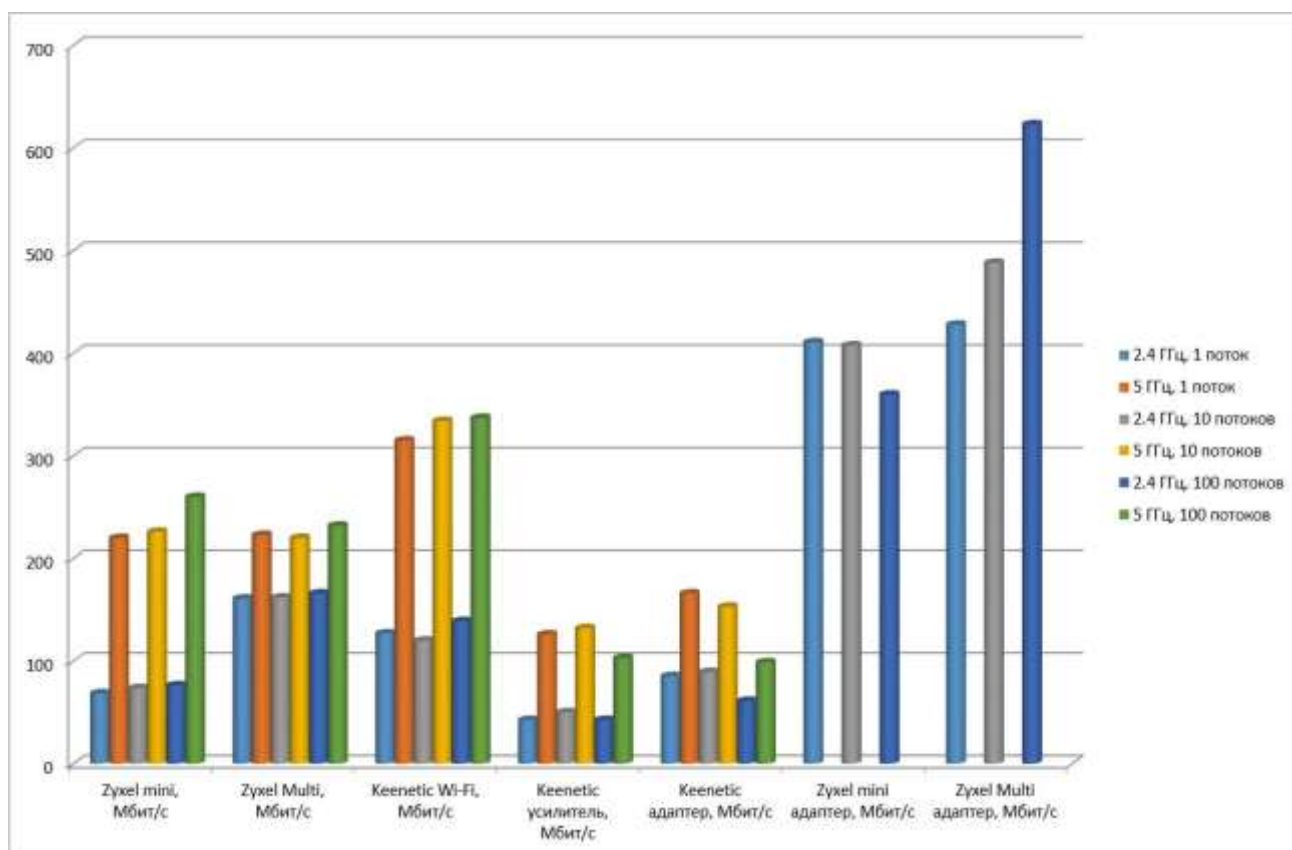


Рис. 1.11. Результати тестування різних ланцюгів підключення

На зображенні видно, що всі пристрої на частоті 2,4 ГГц менш продуктивні, ніж на частоті 5 ГГц. І це при тому, що навколо не було шуму з сусідських заважають мереж, так як в разі наявності зашумленості по частоті 2,4 ГГц результат був би помітно гірше. Проте, наочно видно, що швидкість передачі даних на частоті 5 ГГц майже вдвічі вище, ніж на 2,4 ГГц. Крім того, помітно, що деякий вплив робить і кількість потоків одночасного завантаження, тобто при збільшенні кількості потоків більш щільно використовується канал передачі даних, хоча різниця не така істотна.

Дуже добре видно, коли роутер Keenetic виступав в якості репитера, що швидкість передачі ділиться надвоє, так що варто це враховувати, якщо хочете передавати великі обсяги інформації з великою швидкістю, а не просто розширити покриття Wi-Fi мережі.

На останньому тесті, де Zykel Multy X і Zykel Multy mini виступали в якості адаптера для проводового підключення віддаленого пристрою (зв'язок між базовим Zykel Multy X і пристроєм одержувача була бездротовий)

продемонструвала переваги Multy X, особливо при багатопотокової передачі даних. Позначилося більшу кількість антен у Zyxel Multy X: 9 штук проти 6 у Zyxel Multy mini.

Таким чином, очевидно, що навіть при незавантаженому ефірі на частоті 2,4 ГГц має сенс переходити на 5 ГГц, коли потрібно передавати великі обсяги інформації досить швидко. При цьому, навіть на частоті 2,4 ГГц цілком можна дивитися фільми в FullHD якості, використовуючи роутер як репитера. А ось кіно в 4K з нормальним бітрейтом вже почне заїкатися, тому роутер і пристрій відтворення повинні вміти працювати на частоті 5 ГГц. При цьому найбільша швидкість досягається, якщо в якості бездротового адаптера використовується комплект з двох Zyxel Multy X або Zyxel Multi X + Multy mini.

Mesh система Zyxel може забезпечити широке покриття на дуже пристойній швидкості без прокладки додаткових проводів. Це особливо актуально, коли ремонт вже зроблений, а додатковою кручений пари не закладено. До того ж організація такої мережі максимально проста через додаток на смартфоні. Треба додати до цього, що Mesh мережа може складатися з 6 пристроїв і мати як топологію зірка, так і деревоподібну. Тобто кінцевий пристрій може перебувати дуже далеко від стартового роутера, який підключений до мережі Інтернет.

При цьому Wi-Fi система на базі роутерів Keenetic куди більш функціональна і забезпечує більш дешеву організацію мережі. Але для цього потрібно кабельне підключення. Дистанція між роутерами може бути до 100 метрів, а швидкість абсолютно не буде знижуватися через передачу по гігабітного провідного з'єднанню. При цьому пристроїв в такій мережі може бути більше 6, а роумінг Wi-Fi пристроїв при переміщенні буде безшовним.

Вивчення бездротових сенсорних мереж є складним завданням, оскільки воно вимагає величезної широти знань з величезного різноманіття дисциплін. У цьому розділі ми намічаємо комунікаційні мережі, бездротові сенсорні мережі та інтелектуальні датчики, принципи фізичної трансдукції, комерційно доступні бездротові сенсорні системи, самоорганізацію, обробку сигналів та прийняття

рішень, і, нарешті, деякі концепції домашньої автоматизації [23].

Основною проблемою в мережах зв'язку є передача повідомлень для досягнення заданої пропускної спроможності (кількість послуг) та якості обслуговування (*QoS*). *QoS* можна вказати в термінах затримки повідомлень, термінів надсилання повідомлень, бітових помилок, втрат пакетів, економічної вартості передачі, потужності передачі тощо. В залежності від *QoS*, середовища встановлення, економічних міркувань та програми, одного з декількох базових Топології мережі можуть бути використані.

Мережа зв'язку складається з вузлів, кожен з яких має обчислювальну потужність і може передавати та отримувати повідомлення через лінії зв'язку, бездротові або кабельні [23]. Основні мережеві топології показані на малюнку і включають повністю з'єднані, сітку, зірку, кільце, дерево, автобус. Єдина мережа може складатися з декількох взаємозв'язаних підмереж різних топологій. Мережі також класифікуються як локальні мережі (*LAN*), наприклад, Всередині однієї будівлі або широкосмугових мереж (*WAN*), наприклад Між будинками.

Повністю з'єднані мережі страждають від проблем *NP*-складності [17]; Коли додаються додаткові вузли, кількість посилок збільшується експоненціально. Тому для великих мереж проблематика маршрутизації є обчислювально важкодоступною навіть за наявності великих обсягів обчислювальної потужності.

Мережеві мережі є регулярно розподіленими мережами, які, як правило, дозволяють передачу лише найближчим сусідам вузла. Вузли в цих мережах, як правило, ідентичні, тому мережеві сітки також називаються мережами однорангової мережі. Мережеві сітки можуть бути хорошими моделями для великомасштабних мереж бездротових датчиків, які поширюються по географічному регіону, наприклад, персонал або системи контролю безпеки транспортного засобу. Зауважте, що регулярна структура відображає топологію зв'язку; Фактичний географічний розподіл вузлів не повинен бути звичайною сіткою. Оскільки зазвичай існує кілька шляхів маршрутизації між вузлами, ці

мережі є надійними для відмови окремих вузлів або посилянь. Перевага сітчастих сіток полягає в тому, що, хоча всі вузли можуть бути однаковими і мають однакові обчислювальні та передавальні можливості, певні вузли можуть бути визначені як "лідери групи", які приймають додаткові функції. Якщо лідер групи відключений, інший вузол може взяти на себе ці обов'язки.

Всі вузли топології зірки з'єднані з одним вузлом вузла. Центр вимагає більшої обробки повідомлень, маршрутизації та прийняття рішень, ніж інші вузли. Якщо лінія зв'язку скорочується, вона впливає тільки на один вузол. Однак, якщо концентратор недієздатний, мережа буде знищена. У кільцевій топології всі вузли виконують одну й ту ж функцію, і немає вузла-лідера. Повідомлення, як правило, рухаються по кільцю в одному напрямку. Однак, якщо кільце зрізане, все спілкування буде втрачено. Самоцелезуюча кільцева мережа (*SHR*) має дві кільця і більше відмовостійкість.

У топології шини повідомлення транслуються по шині до всіх вузлів. Кожен вузол перевіряє адресу призначення в заголовку повідомлення та обробляє повідомлення, адресовані йому. Топологія шини є пасивною, оскільки кожен вузол просто слухає повідомлення та не несе відповідальності за повтор передавання повідомлень.

Висновки за розділом

Проведений системний аналіз прикладного програмного та апаратного забезпечення систем теледіагностики в умовах гібридного середовища передачі даних показав можливість використання для даних потреб пористих мереж (*mesh*). А порівняльний аналіз протоколів побудови *mesh*-мереж дозволив зробити наступні висновки:

- *BLE Mesh* забезпечує максимальну дальність серед розглянутих мереж. З незначним відставанням в 2-5% за ним слід *Thread*. Із запропонованих мереж дальність покриття *ZigBee* менше на 16-18% відносно лідера;

- *BLE Mesh* підтримує найбільшу кількість вузлів в мережі (теоретично

до 32 тис., На практиці використовуються мережі до 1-2 тис.) І орієнтований на передачу невеликих обсягів інформації;

- *Thread* забезпечує найбільшу пропускну здатність мережі і найменші затримки, а також найбільш стабільний результат - середній час до 50 мс;

- *ZigBee* відстає приблизно в 2 рази відносно *Thread* по пропускій здатності і затримки в передачі даних практично у всіх тестах. Однак забезпечує при цьому гідний результат, що дозволяє використовувати мережу в більшості застосувань;

- *BLE Mesh* показує найменшу швидкість передачі даних. Ця мережа також є найменш стабільною і схильна до впливу як числа вузлів, так і розміру переданих даних. Затримка може досягати в результаті максимального значення більш 300 мс;

- для оптимальної роботи мережі *BLE Mesh* потрібно оптимізація кількості вузлів ретрансляторів, а також їх розташування, в той час як *Thread* і *ZigBee* мають вбудовані механізми оптимізації.

- *BLE Mesh* - найбільш проста мережа для реалізації і, відповідно, вимагає найменше ресурсів, однак це обертається додатковими обмеженнями.

При виборі мережі потрібно враховувати додаткові чинники, такі як простота інтеграції з іншими сервісами, можливість подальшого розвитку проекту:

- основними застосуваннями для *BLE Mesh* передбачаються системи освітлення, а також рішення з максимально простою інтеграцією зі смартфонами, планшетами, тощо;

- *Thread* – універсальне рішення з кращими показниками швидкості передачі даних, мінімальними затримками і легкої інтеграцією з *IP*-мережами і «Інтернетом речей»;

- *ZigBee* доцільно застосовувати у випадках, коли необхідно інтегрувати нове рішення з уже існуючою мережею на цьому протоколі.

РОЗДІЛ 2

ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЇ ПРИКЛАДНОГО ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ ТЕЛЕДІАГНОСТИКИ

2.1. Топологія і структура радіомережі передачі даних

2.1.1. Базові топології мереж *LR_WPAN*

Зірка - *star* (проста топологія). Кожен сенсор встановлює зв'язок і веде обмін тільки зі своїм батьком, яким є центральний вузол мережі – *PAN* координатор мережі *LR_WPAN*. Ця топологія реалізується тільки за допомогою стандарту *IEEE 802.15.4*. Усі вузли чують лише координатор *PAN* і тому можуть звертатися один одному тільки через нього. У центрі зірки - координатор *PAN*: центр збору даних [14].

Дерево - *tree* (складніша топологія), при якій у кожного вузла м би. один батьківський і декілька дочірніх вузлів. Тут також використовується один *PAN* координатор, але є і інші вузли-координатори.

Комірка - *mesh*. Комірчаста топологія пропонує альтернативні варіанти вибору маршруту між вузлами. Тут також використовується один *PAN* координатор мережі, усі інші вузли ідентичні. Пакети транслуються від вузла до вузла, поки не досягнуть кінцевого одержувача. Можливі різні шляхи проходження пакетів, що підвищує доступність мережі у разі виходу з ладу тієї або іншої ланки.

Точка-точка - *point - to - point*. Вузли у великих мережах з топологією кластерне дерево також і пов'язані між собою (без використання координатора), що дозволяє будувати масштабовані системи.

Точка – багато точок - *point - to - multipoint*. Вузол по черзі або ширококомовно передає пакети усім вузлам-сусідам з таблиці сусідів. Ця топологія – окремий випадок топології зірка при взаємодії двох вузлів і також

містить координатор *PAN*. Це означає, що один з двох вузлів д. би. ведучим *master*, інший веденим *slave*. Взаємодія цих вузлів відбудеться лише за двох умов:

1) у провідного вузла має бути адреса *PAN*, 2) адреса веденого вузла "прописана" в пам'яті у провідного вузла.

Кластерне дерево - *cluster_tree*. В порівнянні із зіркою вводяться ретранслятори пакетів з базовою функцією маршрутизації пакетів, які також здійснюють контроль маршрутів в мережі, вибір часу ретрансляції залежно від завантаженості радіоефіру. Вузли в цій топології самі розраховують або прокладають маршрут, залежно від методу маршрутизації. Якщо вузол, вказаний в маршрутній карті, або вийшов з ладу, або сильно погіршав зв'язок з ним, або виник затор, він може змінити маршрут на один з резервних або вибрати інший ретранслятор в області радіовидимості.

2.1.2. Асоціація - входження вузла до складу *PAN*

Асоціація виконується автоматично. Для цього координатор *PAN* використовує кадр маяка, який розпізнає будь-який вузол. Вузол-кандидат, що отримав кадр маяка, може звернутися до координатора *PAN*, видає запит на приєднання до мережі. Якщо координатор *PAN* дозволить цьому вузлу асоціацію, він присвоїть йому статус координатора і додасть новий координатор в якості дочірнього координатора у свій список сусідів, передавши йому у відповідь номер слоту доступу і адресу "MAB".

Потім тільки що асоційований координатор додасть *PAN* координатор, як свого батька, у свій список сусідів, і почне передавати маяки; інші кандидат-координатори потім приєднуються до мережі. Якщо початковий *PAN* координатор не може підключити вузол-кандидат до мережі, останній шукатиме іншого батька [12].

У кожного вузла є декілька сусідів. Кожного з них він чує і може вести з

ним обмін кадрами. Група сусідів утворює осередок *mesh*, в якій усі можуть обмінюватися один з одним. Якщо вузли не сплять постійно і слухають своїх сусідів, не економлячи енергоспоживання, то мережа з багатьох осередків може мати властивість *Self_healing* -самовідновлення - виявлення і виправлення несправності в роботі вузлів і зв'язків між вузлами без участі людини. Ця здатність заснована на перевірці якості ліній зв'язку між вузлами завдяки постійному прослуховуванню ефіру і наявності функції *LQI*. У мережі доставка даних не координується з центру - шлюзу *MESH BLE_Gate*, а здійснюється за принципом ретрансляції від свого сусіда до його сусіда в спеціально виділених для цього слотах. Мережа *mesh* побудована на базі топології "кожен-з-кожним" з децентралізованою доставкою даних. Роль координатора *PAN* зводиться лише до функції першого вузла мережі, адреси *PAN*, що є носієм. Для передачі даних по мережі координатора *PAN* не потрібно. Метод реалізації мережі *LR_WPAN* є предметом технології типу *BlueTooth*. Можливість самоорганізації і самовідновлення мережі на базі топології *Cluster_Tree* з осередками *Mesh* дозволяє спонтанно і оперативно створювати мережі. Мережа автоматично формується з великого числа вузлів, розкиданих по території кластера, по "початковому імпульсу" від *PAN* координатора - коли в мережу його сусідам починають поступати від нього кадри маяка. Якщо усі вузли мають автономне живлення, існує проблема обмеженого часу функціонування такої мережі. Зменшується воно і зі збільшенням території охоплення мережею. Оперативність і відсутність витрат на координування є головним критерієм. Спонтанно створювані мережі іменують термін *Ad_Hoc* "випадково" з можливістю розширення в часі і в просторі.

2.1.3. Створення мережі *MESH-BLE* з топологією кластерів "*line-mesh*".

Топологія кластера мережі *MESH-BLE* має дворівневу структуру *Line - Mesh*, яка дозволяє максимально ефективно використовувати виділені 16-ть каналів в смузі частот 2.4-2.5 ГГц, поєднуючи швидкість передачі даних і

надійність функціонування мережі. *MESH-BLE* має динамічну мережеву топологію, в якій маршрути передачі даних постійно міняються, але сам вузол мережі встановлений стаціонарно. При додаванні або видаленні вузла мережі відбувається автоматична переконфігурація мережі. Кластер мережі складається з його координатора і асоційованих з ним радіотрактів, до складу яких входять роутери – вузли з включеною функцією маршрутизації. Координатор кластера виконує і функції концентратора – вузла вищого 0-го рівня топології, який може обмінюватися даними з будь-якими вузлами, що входять до складу його мережі.

Роутер є посередник між вузлами мережі для транзиту даних від видалених сенсорів до концентратора. Кожен вузол і сенсор мережі має свою унікальну адресу MAB. Сенсор *MESH-BLE* – це вузол без функції маршрутизації; він може бути і одержувачем, і джерелом даних. Будь-який вузол може бути кінцевим одержувачем даних

MESH-BLE підтримує ланцюжок *line-mesh-tree* топології: сенсори-вузли-радіотракти-кластери-гіпермережа:

1) *line* -топологія: радіотракт має двонаправлений трафік і містить:

- генератор і термінатор суперкадрів, розташовані в крайніх точках радіотракту;

- набір вузлів - роутери і ретранслятори;

- групи асоційованих з вузлами стаціонарних і мобільних сенсорів.

Лінійну топологію визначає структура шахти - тунелі і штреки, або маршрути міського транспорту.

2) *mesh*-топологія: визначає набір радіотрактів, що утворюють кластер мережі – два і більше радіотрактів в межах мережі, що взаємодіють в одному і тому ж фізичному радіоканалі.

Мережевий кластер включає один координатор, який, як правило, виконує і функції концентратора [3].

У *line-mesh* топології координатор відповідає за старт мережі і вибір її ключових параметрів. Використання роутерів дозволяє розширювати мережу. У

line мережі роутер/ретранслятор пересилає пакети і управляє мережею без прокладення маршрутів, використовуючи метод лінійної маршрутизації з повторним прийняттям. *Line*-мережа застосовує комунікації, орієнтовані на суперкадри. *Mesh*-мережі забезпечують повні *peer-to-peer* зв'язки між радіотрактами, що забезпечуються тільки роутерами. Проект використовує квазі-*mesh* мережа, в якій створення мережі і генерація регулярних суперкадрів відбувається із застосуванням структури *line*-мережі, а обмін даними виконують роутери з логічною структурою *mesh*-сети. Цей проект описує тільки *intra*-сети: зв'язок починається і завершується в одному і тому ж кластері мережі.

MESH-BLE має дворівневу топологію: *line* для організації радіотракту мережі в тунелі або штреку з підключенням сенсорів і квазі-*mesh* для з'єднання радіотрактів між собою. Вузол використовується як точка ініціації, завершення і маршрутизації зв'язку для мережевих комунікацій. Координатор кластера є основним контролером в мережі. Усі вузли в мережі з будь-якою топологією мають унікальну адресу, що складається з двох частин: 16-бит ідентифікатор *MESH-BLE* мережі і власне 16-біт коротка "Мережева Адреса Вузла - МАВ" в цій мережі, використовуваний для прямого спілкування у рамках одного кластера, і кластера, що виділяється координатором, коли вузол входить до складу мережі (процедура асоціації, оординатором, що проводиться). Сенсори також мають власний МАВ, призначений йому вузлом-батьком під час асоціації. Мобільні сенсори в мережі асоціюються як в просторі, так і в часі з будь-яким вузлом за наявності ресурсів (слотів і МАВ), а стаціонарні сенсори - тільки в просторі.

Топології *line-mesh* мають тільки координатори і роутери, при цьому будь-який вузол може спілкуватися з будь-якими іншими сусідніми вузлами (до 14-ти в цьому проекті), якщо вони знаходяться у сфері охоплення один з одним. Однорангова мережа *mesh* являється вузькоспеціалізованою мережею з самовідновленням і дозволяє застосовувати т. н. хопи для маршрутизації повідомлень по методу "маршрутизація_від_джерела" від будь-якого вузла до

будь-якого іншого вузла в мережі. Хоп (стрибок, стрибок) є одиницею виміру довжини маршруту в числі проміжних вузлів, включаючи також вузол призначення. У проекті з топологією *line - mesh* застосований на мережевому рівні тільки метод маршрутизації від джерела з повторним прийманнями і обхідними шляхами. Кожен кластер мережі має унікальний 16-бітовий ідентифікатор *PAN ID*, який дозволяє вузлам зв'язуватися в мережі з використанням 16-бітових МАС і дає можливість передачі між вузлами через кластери гіпермережі. Механізм вибору ідентифікатора і створення мережі проводиться на *NWK* рівні.

Охоплення мережі може розширюватися за межі *MESH-BLE*. Виразно вираженої області охоплення не існує для радіосередовища, тобто параметри поширення – динамічні і невизначені. Невеликі зміни місця розташування або напрямку можуть привести до великої зміни сили сигналу або якості сеансу зв'язку. Ці ефекти відбуваються незалежно від пристрою - нерухомого або мобільного, т. до. переміщення об'єктів може впливати на поширення сигналу від вузла до вузла [10].

Нижче представлений короткий опис мережі з топологією кластерів "*line-mesh*".

Використовується комбінована топологія датчикової мережі: зірка – двонаправлена радіюшина – комірка. Мережа складається з безлічі вузлів, причому один з них виступає в ролі координатора мережі.

Приклад топології кластера мережі *line - mesh* як варіант однорангової мережі *mesh*, де більшість вузлів - роутери/ретранслятори. Мобільний сенсор-нащадок підключається до кластера в якості тимчасового елементу і він не дозволяє іншим вузлам спілкуватися через нього. До складу вузла-роутера стека ПО мережевого рівня входить програма маршрутизації.

Після активізації 1-го вузла якого-небудь тунеля (штреку) і привласнення йому статусу генератора суперкадру він може створити свій власний радіотракт мережі. Вузол-генератор забезпечує функцію автономного створення

радіотракту мережі і синхронізації його вузлів за допомогою своїх суперкадрів. Вузли-генератори в інших тунелях, у свою чергу, створюють свої радіотракти. Можливий режим роботи, коли суміжні радіотракти створюються і працюють на різних радіоканалах 11-26. Один з координаторів є загальним *PAN* координатором з великими обчислювальними ресурсами, чим будь-який інший координатор. Цей координатор оголошується в якості координатора кластера, внаслідок того, що він став першим координатором, що встановив зв'язок в кластері. Базова структура *line-mesh* має вибраний *PAN ID*, зараз не використовуваний будь-якою іншою мережею в межах радіосфери впливу.

Після того, як призначений мережі *PAN ID* вибраний (механізм вирішення конфліктів *PAN ID* не вимагається, т. до. у мережі відсутні координатори з ідентичними *PAN ID*), координатор кластера формує 1-й радіотракт, дозволяючи іншим радіотрактам увійти до складу свого кластера і передаючи широкомовні кадри створення кластера до сусідніх радіотрактів. Далі 1-й радіотракт підключає до себе інші суміжні радіотракти, якщо вони раніше були вже незалежно створені своїми вузлами-генераторами. Структура мережі будується за схемою *mesh* і відображає топологію тунелів і штретків при формуванні мережі.

Процедура асоціації радіотрактів в кластер подібна до асоціації вузлів *PAN* мережі і розглядається нижче. Створення мережі *mesh* відбувається за розкладом хвилями від центру до периферії від координатора 1-го радіотракту, яке упорядковує процес створення мережі. Радіотракти топології *line* одного і того ж кластера не залежать один від одного і кожним з них управляє координатор радіотракту. У топології *mesh* кожен вузол спілкується тільки з сусідніми вузлами у рамках своєї радіосфери впливу, створюючи таблицю сусідів.

MESH-BLE - протокол передачі даних для створення безпроводних систем моніторингу і управління. Протокол *MESH-BLE* реалізований на базі ядра реального часу *MESH-BLECore* і апаратно незалежний, на відміну від

BlueTooth. Він дозволяє створювати радіомережі датчиків і старанних модулів в діапазоні 2.4 МГц. Досягається це завдяки модульній структурі ПО, для переходу на іншу елементну базу необхідно тільки перепідключити модуль, що відповідає за фізичний рівень протоколу.

У рамках протоколу *MESH-BLE* реалізовані наступні механізми взаємодії вузлів:

- конфігурація;
- самоорганізація і самовідновлення;
- маршрутизація;
- контроль присутності вузлів в мережі;
- перехід на резервний маршрут
- аутентифікація;
- гарантована доставка пакетів даних.

Гіпермережа формується з декількох сусідніх кластерів, наприклад, кластер утворює мережу одного горизонту шахти, а гіпермережа – інтеграція кластерів в межах однієї шахти. Як тільки заздалегідь встановлені Застосування або мережеві вимоги виконані, перший *PAN* координатор може доручити одному з граничних координаторів, щоб стати *PAN* координатором нового кластера, прилеглого до першого [11].

Інші координатори поступово підключаються, утворюючи структуру мультикластерної гіпермережі. Вузли в мережі мають між собою відношення "батько-нащадок", а не зв'язки потоків кадрів. Перевага гіпермережі є збільшення зони покриття, а недолік – збільшення затримки повідомлень (рис. 2.1).

2.2. Вибір стека протоколів для платформи розробки радіомережі *MESH-BLE*

Специфікація стека – це зведення правило, по яких створюється програма для конкретного *RISC* процесора.

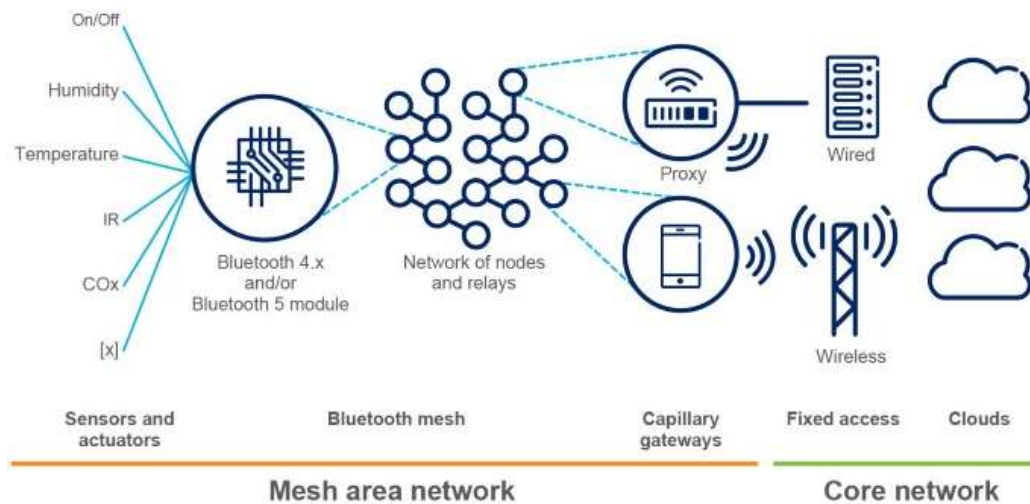


Рис. 2.1 Структура топології радіомережі *MESH-BLE*

Стек *BlueTooth* – входить до складу тільки сенсорів і його специфікація включає профілі елементів мережі - формальний опис властивостей конкретних датчиків і старанних пристроїв, що виконують однакові функції. *BlueTooth* - профіль описує типи сенсорів і інтерфейсів для конкретного Застосування, наприклад, профілі для *BlueTooth* -пристроїв автоматизації будинку і моніторингу підприємств (різноманітні датчики в промисловому і медичному устаткуванні, "розумні" будинки і т. п.).

Стек *BlueTooth* відповідає за структуру мережі, безпеку і вибір маршруту. *BlueTooth* - стандарт організації низькошвидкісних *LR_WPAN* радіомереж з топологією, схожою на зигзагоподібну траєкторію польоту бджоли). *BlueTooth* виділяє: проста інсталяція, надійна передача, низька вартість і споживання.

Основу *BlueTooth* складає специфікація *IEEE 802.15.4*, що відповідає за реалізацію усіх основних функцій, - від фізичної передачі даних до об'єднання в радіомережу, а також організація доступу сенсорів до мережі.

Стек *BlueTooth* - *PRO* описаний в документі "Посібник користувача *BlueTooth PRO Stack JN - UG - 3048 v2.0*" (рис. 2.2).

Стек *MESH-BLE* призначений головним чином для вузлів радіотракту (генератор і термінатор суперкадру, роутери і ретранслятори) і його специфікації визначають властивості вузлів при створенні мережі, процедури

передачі пакетів між вузлами мережі, вибір координаторів мережі, процедури асоціації (входження до складу мережі) нових вузлів і сенсорів, принципи взаємин "батько-нащадок". Для реалізації стека програм *MESH-BLE* потрібно *RISC* процесор з об'ємом ОЗУ і флеш-пам'яті - по 128 кБайт.

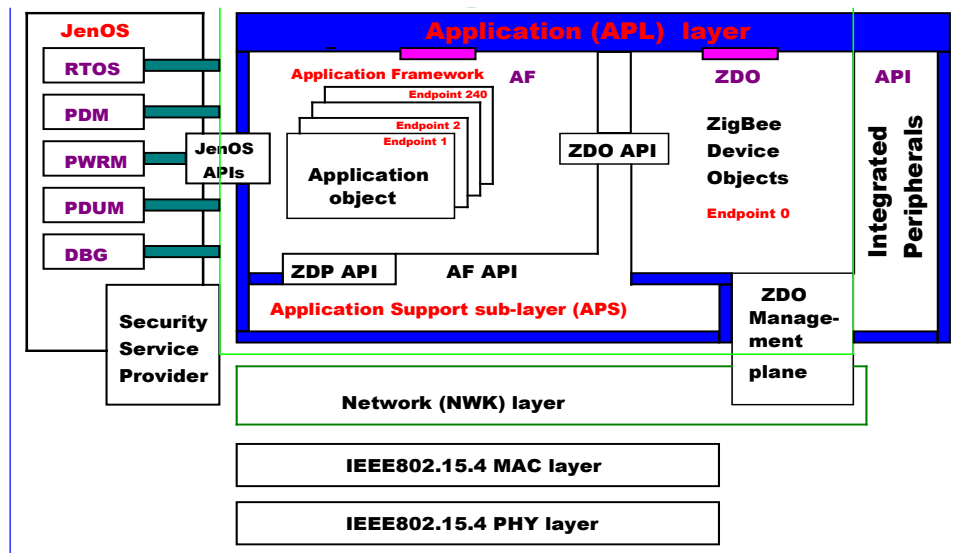


Рис. 2.2. Стек протоколів радіомережі

Для вирішення завдань автоматизації і збору інформації розроблена технологія і протоколи *MESH-BLE* на базі наступних стандартів *IEEE* 802.15.4, *IEEE* 802.6 і специфікацій фірми *NXP*:

- *IEEE* 802.15.4-2006 - протоколи фізичного рівня *PHY* (визначає параметри радіотрансивера) і канального підрівня *MAC* доступу до середовища (специфіковані ф. *NXP* у вигляді бібліотек для РМК *JN5148*);

- *DQDB_R* - протокол *LLC* рівня мультислотового доступу до двонаправленої радіюшини з розподіленою черговістю на базі модифікованого стандарту *DQDB IEEE* 802.6 стосовно радіомережі *MESH-BLE*;

- специфікації *JenNet* і *BlueTooth* стосовно радіомережі *MESH-BLE* із скороченням зайвої складності і вибором оптимальної реалізації стандартів для мереж великої протяжності.

Архітектура мережі *MESH-BLE* визначена в термінах ряду блоків, які називають рівнями. Кожен рівень відповідає за одну частину проекту і пропонує свої послуги у верхні рівні. Інтерфейси між рівнями визначають

логічні зв'язки. Радіомережа *MESH-BLE* специфікує 6 рівнів протоколів моделі *OSI-7*:

- *PHY* → фізичний на базі протоколу *IEEE 802.15.4* [нижчий рівень стека];
- *MAC* → управління доступом до середовища передачі [доступ до радіоканалу] на базі протоколу *IEEE 802.15.4*;
- *LLC* → управління логічним каналом [доступ до радіотракту] на базі *DQDB_R* (тільки для вузлів);
- *NWK* → мережевий плюс маршрутизація (тільки для вузлів);
- *APS* → транспортний плюс сеансовий (тільки для стаціонарних і мобільних сенсорів);
- *APL* → прикладний (тільки для стаціонарних і мобільних сенсорів)[вищий рівень стека].

Наступні чинники враховані в процесі вибору алгоритмів для різних рівнів протоколів: фізичний (потужність передавача, тип антени). Вузли сумісні із стандартом *IEEE 802.15.4*.

Чіп РМК *JN5148* має ВЧ-модуль на кристалі з потужністю передавача *2.5 dBm*, пасивне об'язування з 50-м інтерфейсом антени, аналогову/цифрову периферію. Стандартні модулі *2.5 dBm* покривають відстань до 100 м, а високопотужні модулі *20 dBm* □ понад 500 м (залежно від рельєфу місцевості і умов видимості). Модуль має два виконання:

а) зі вбудованим невеликим кристалом чіп-антени, змонтованим на друкованій платі радіомодуля, який займає менше місце, але пред'являє більше вимог до навколишніх поверхонь і не забезпечує потрібної потужності для необхідного додатку.

б) із зовнішньою антеною, що під'єднується до модуля через чіп підсилювача потужності *RFX2411* і *SMA* ВЧ-роз'єм.

Канальний (*MAC/LLC* підрівні). У Застосуваннях стандарту *IEEE 802.15.4 MAC* -підрівень виконує:

- встановлення зв'язку вузлів з мережею і від'єднання від неї (проблеми створення і асоціації радіотрактів);

- контроль над доступом до каналів, використовуваних різними вузлами і сенсорами в мережі;

- генерацію кадрів маяків і підтверджень, а також управляє тайм-слотами.

Мережевий і транспортний рівні в конфігураціях радіодатчикових мереж використовують технології *BlueTooth* і *Z - wave*, а також менш вживані альтернативні методи.

MESH-BLE - радіомережа додатків, що самоорганізуються, з обмеженою потужністю і низькими вимогами по продуктивності. Основні її цілі – зручність установки, надійна передача даних, операції з великим радіусом дії, низький рівень вартості, термін служби акумуляторів місткістю 8 А*ч до 6 років, надійний і гнучкий протокол. Вузол мережі забезпечує збір даних і передачу "повідомлень – пакетів – слотів – кадрів цих/команд" за розкладом на основі вбудованого календаря *RTC*, обробляє запити від координатора для отримання поточних даних, відправляючи аварійні/діагностичні повідомлення.

Вузол *MESH-BLE* містить *PHY* -рівень, що містить радіочастотний трансивер з низькорівневим механізмом управління, і *MAC* -рівень, доступ, що забезпечує, до фізичного каналу для усіх типів передач. Показано ці блоки в графічному представленні, які детальніше описані у відповідних розділах. Верхні рівні складаються з мережевого рівня *NWK*, який забезпечує налаштування мережі, маніпуляції і маршрутизацію повідомлень, *APS* [сеансовий плюс транспортний] і прикладного рівня, який забезпечує передбачені функції координатора.

MESH-BLE архітектура реалізована у вигляді вбудовуваних модулів без підтримки зовнішніх пристроїв (ПК).

PHY -рівень надає послуги *PHY* даних і *PHY* управління, що взаємодіють з точкою доступу до послуги (*SAP - Service Access Point*) за допомогою примітив управління *PHY* рівнем *PLME - Physical Layer Management Entity* [*PLME - SAP*]. Послуга *PHY* даних забезпечує передачу/прийом *PHY* протокольних блоків даних *PPDU - PHY Protocol Data Units* через фізичний радіоканал. Специфікації *PHY* см в п. 6. Функції *PHY* рівня:

включення/виключення радіотрансивера, *ED*, *LQI*, оцінка статусу каналу *CCA* "*Clear Channel Assessment*", а також передача/прийом кадрів через фізичне середовище.

MAC-рівень надає послуги *MAC* даних і *MAC* управління, що взаємодіють з точкою доступу до послуги *SAP* через примітиви управління *MLME* - *MAC Layer Management Entity* [*MLME* - *SAP*]. Послуга *MAC* даних дозволяє передавати і приймати протокольні блоки даних *MAC Protocol Data Unit* [*MPDU*] через службу *PHY* дані. Функції *MAC* рівня: управління маяком, синхронізація з маяком, доступ до каналу, управління *GTS* і *CSMA - CD*, контроль дійсності кадрів, підтвердження доставки кадрів, асоціація і дисасоціація, виявлення конфліктів адрес і вузлів-сиріт.

Радіо "*JN5148*" працює в неліцензійованому діапазоні 2.4 *GHz* в смузі частот 2400.0 – 2483.5 МГц.

Пересилка інформаційних пакетів. Висока швидкість передачі інформ-пакетів *JN5148* в активному режимі 250 кБод з буферизацією даних помітно зменшує середнє споживання струму і вірогідність конфліктних ситуацій з іншими системами в тому ж частотному діапазоні. Чутливість і тому діапазон передач зменшені для високошвидкісних передач пакетів.

Малопотужні роутери *Low power routers* "*LPR*" працюють від акумуляторів декілька років при регулярному виключенні свого радіотрансивера і пізнаються координаторами, аналізуючи біт-поля *capability* впродовж фази *joining*. Тип вузла:= *LPR* з можливістю приймати мережеві кадри команд, які є широкомовними в мережі. Це досягається установкою адреси призначення в *NWK* заголовку як ШІВА для усіх роутерів і координаторів.

Функціональне загальне представлення *MESH-BLE* включає інформацію про модель передачі даних, підвищення вірогідності успішної доставки, споживаної потужності і безпеці.

Модель передачі даних. Три види операцій передачі: 1) непряма передача даних від сенсора до вузла;

2) отримання даних сенсором від вузла; 3) пряма передача даних між двома вузлами. Топологія зірка використовує операції 1 і 2 (даними обмінюються тільки вузол і сенсор). У топології *line - mesh* даними в мережі обмінюються будь-які два сусідні вузли. Механізми для кожного типу передачі залежать від протоколу передачі кадрів.

PAN з дозволеними маяками використовує синхронізацію і підтримку для низько-латентних пристроїв.

Маяк потрібний для відкриття мережі, асоціації вузлів і передачі мережевих команд.

Передача даних до вузла. Коли сенсор-нащадок бажає передавати дані до вузла-батька в мережі з суперкадрами, він прослуховує їх. Коли суперкадр знайдений, вузол синхронізується з його структурою.

У заданий час сенсор передає вузлу свій кадр даних, використовуючи свій слот в структурі суперкадру.

Вузол визнає успішне отримання даних шляхом передачі кадру АСК. Транзакція завершена.

Передача даних від вузла. Коли вузол-батько бажає передати дані до сенсора-нащадка в *PAN* з суперкадрами, він вказує в суперкадрі, що є очікуючий кадр даних. Сенсор періодично слухає свій суперкадр, і якщо в нім вказано, що для цього сенсора є очікуючий передачі кадр, передає команду *MAC* запиту даних у своєму слоті. Вузол підтверджує успішне отримання запиту даних шляхом передачі кадру АСК. Очікуючий кадр даних потім передається в цьому ж слоті услід за кадром АСК. Сенсор звітує про успішний прийом даних передачею кадру АСК. Транзакція завершена. Після успішного завершення транзакції передачі даних, кадр видаляється із списку очікуючих кадрів.

Передача даних в мережі з топологією "*mesh*". У *mesh* -топології вузол спілкується з будь-яким іншим вузлом-сусідом у своїй радіосфері впливу. Для ефективності операцій охочі передавати вузли постійно контролюють суперкадри один одного і синхронізуються один з одним. Вузол після

синхронізації передає свої дані у виділеному для нього слоті. Заходи для забезпечення синхронізації описані нижче.

Підвищення вірогідності успішної доставки даних: *MESH-BLE* використовує механізми перевірки достовірності даних і наскрізного підтвердження кадру.

Чинник споживаної потужності. Фізична реалізація *MESH-BLE* вимагає особливе ПЗ управління живленням. Вузол живиться тільки від акумулятора. Пасивну частину періоду суперкадру вузол проводить в сплячому режимі а в активній частині суперкадру він виконує операції по управлінню вузлами і по обміну даними з вузлами-сусідами. У апріорі відомі інтервали часу вузол періодично слухає *RF* канал для визначення, чи є для нього очікуючі відправки кадри від його сенсорів-нащадків.

Цей механізм регулює баланс між споживанням акумулятора і часом очікування повідомлень.

2.3. Методи аналітичного, імітаційного і натурного моделювання

Моделювання являє собою потужний метод наукового пізнання, при використанні якого досліджуваний об'єкт замінюється простішим об'єктом, який називається моделлю. Основними різновидами процесу моделювання можна вважати два його види - математичне і фізичне моделювання. При фізичному (натурному) моделюванні досліджувана система замінюється відповідною їй іншою матеріальною системою, яка відтворює властивості досліджуваної системи зі збереженням їх фізичної природи.

Можливості фізичного моделювання досить обмежені. Воно дозволяє вирішувати окремі завдання при завданні невеликої кількості поєднань досліджуваних параметрів системи. Дійсно, при натурному моделюванні обчислювальної мережі практично неможливо перевірити її роботу для варіантів з використанням різних типів комунікаційних пристроїв - маршрутизаторів, комутаторів тощо Перевірка на практиці близько десятка

різних типів маршрутизаторів пов'язана не тільки з великими зусиллями і тимчасовими витратами, а й з чималими матеріальними витратами.

Але навіть і в тих випадках, коли при оптимізації мережі змінюються не типи пристроїв і операційних систем, а тільки їх параметри, проведення експериментів в реальному масштабі часу для величезної кількості всіляких поєднань цих параметрів практично неможливо за доступний для огляду час. Навіть проста зміна максимального розміру пакету в якому-небудь протоколі вимагає переконфігурування операційної системи в сотнях комп'ютерів мережі, що вимагає від адміністратора мережі проведення дуже великої роботи.

Тому, при оптимізації мереж у багатьох випадках переважним виявляється використання математичного моделювання. Математична модель являє собою сукупність співвідношень (формул, рівнянь, нерівностей, логічних умов), що визначають процес зміни стану системи залежно від її параметрів, вхідних сигналів, початкових умов і часу.

Особливим класом математичних моделей є імітаційні моделі. Такі моделі являють собою комп'ютерну програму, яка крок за кроком відтворює події, що відбуваються в реальній системі. Стосовно до обчислювальних мереж їх імітаційні моделі відтворюють процеси генерації повідомлень додатками, розбиття повідомлень на пакети і кадри певних протоколів, затримки, пов'язані з обробкою повідомлень, пакетів і кадрів усередині операційної системи, процес отримання доступу комп'ютером до поділюваного мережевому середовищі, процес обробки надходять пакетів маршрутизатором і т.д. При імітаційному моделюванні мережі не потрібно купувати дороге обладнання - його роботи імітується програмами, досить точно відтворюють всі основні особливості і параметри такого устаткування.

Перевагою імітаційних моделей є можливість підміни процесу зміни подій в досліджуваній системі в реальному масштабі часу на прискорений процес зміни подій в темпі роботи програми. У результаті за кілька хвилин можна відтворити роботу мережі протягом декількох днів, що дає можливість оцінити роботу мережі в широкому діапазоні варіюваних параметрів.

Результатом роботи імітаційної моделі є зібрані в ході спостереження за протікають подіями статистичні дані про найбільш важливих характеристиках мережі: часах реакції, коефіцієнтах використання каналів і вузлів, вірогідності втрат пакетів і т.п.

Існують спеціальні мови імітаційного моделювання, які полегшують процес створення програмної моделі в порівнянні з використанням універсальних мов програмування. Прикладами мов імітаційного моделювання можуть служити такі мови, як *SIMULA*, *GPSS*, *SIMDIS*.

Існують також системи імітаційного моделювання, які орієнтуються на вузький клас систем, що вивчаються і дозволяють будувати моделі без програмування.

2.4. Моделі теорії масового обслуговування

Використовувані в даний час в локальних мережах протоколи канального рівня використовують методи доступу до середовища, засновані на її спільне використання кількома вузлами за рахунок розділення в часі. У цьому випадку, як і у всіх випадках поділу ресурсів з випадковим потоком запитів, можуть виникати черги. Для опису цього процесу зазвичай використовуються моделі теорії масового обслуговування.

Механізм поділу середовища протоколу *Ethernet* спрощено описується найпростішою моделлю типу $M/M/1$ - одноканальною моделлю з пуассоновським потоком заявок і показовим законом розподілу часу обслуговування.

Вона добре описує процес обробки випадково вступників заявок на обслуговування системами з одним обслуговуючим приладом з випадковим часом обслуговування і буфером для зберігання заявок на час, поки обслуговуючий прилад зайнятий виконанням іншої заявки (рис. 2.3).

Середовище *Ethernet*, в режимі передавання інформації, представлено в цій моделі обслуговуючим приладом, а пакети відповідають заявкам.

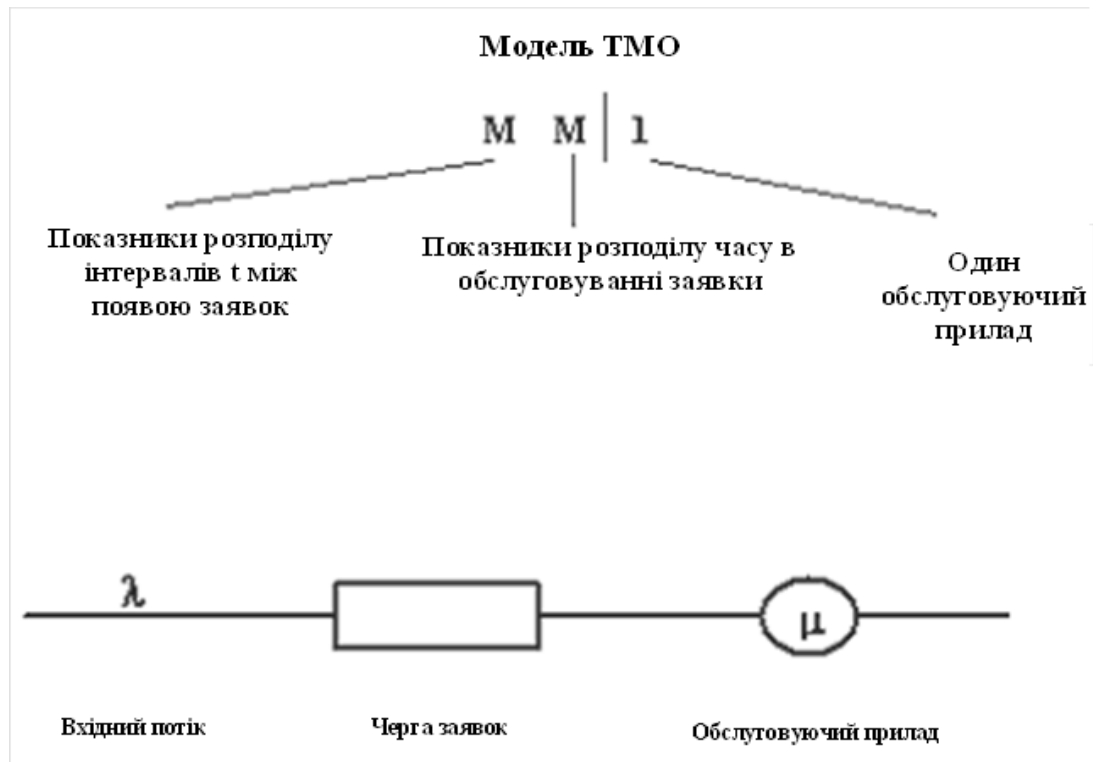


Рис. 2.3. Застосування моделі теорії масового обслуговування $M/M/1$ для аналізу трафіку в мережі *Ethernet*

Введемо позначення: l - інтенсивність надходження заявок, в даному випадку це середнє число пакетів, що претендують на передачу в середовищі в одиницю часу, b - середній час обслуговування заявки (без урахування часу очікування обслуговування), тобто середній час передачі пакету в середовищі з урахуванням паузи між пакетами в 9.6 мкс, r - коефіцієнт завантаження обслуговуючого приладу, в даному випадку це коефіцієнт використання середовища, $r = lb$.

У теорії масового обслуговування для даної моделі отримані наступні результати:

- показники розподілу інтервалів t між появою заявок $A(t) = 1 - e^{-\lambda t}$;
- показники розподілу часу в обслуговуванні заявки $B(x) = 1 - e^{-\mu x}$, $h = 1/\mu$, $\rho = lb$;
- середній час очікування заявки в черзі (час очікування пакетом доступу до середовища) $W = \rho b / (1 - \rho)$.

2.5. Спеціалізовані системи імітаційного моделювання обчислювальних мереж

Існують спеціальні, орієнтовані на моделювання обчислювальних мереж програмні системи, в яких процес створення моделі спрощений. Такі програмні системи самі генерують модель мережі на основі вихідних даних про її топології і використовуваних протоколах, про інтенсивності потоків запитів між комп'ютерами мережі, протяжності ліній зв'язку, про типи використовуваного обладнання та програм. Програмні системи моделювання можуть бути вузько спеціалізованими і достатньо універсальними, що дозволяють імітувати мережі найрізноманітніших типів.

Програмні системи моделювання мереж - інструмент, який може стати в нагоді будь-якому адміністратору корпоративної мережі, особливо при проектуванні нової мережі або внесенні кардинальних змін у вже існуючу. Продукти даної категорії дозволяють перевірити наслідки впровадження тих чи інших рішень ще до оплати придбаного обладнання.

Програми імітаційного моделювання мережі використовують у своїй роботі інформацію про просторове розташування мережі, числі вузлів, конфігурації зв'язків, швидкостях передачі даних, використовуваних протоколах і типі обладнання, а також про виконувані в мережі додатках.

Таблиця 2.1.

Огляд систем імітаційного моделювання проектування мереж

Компанія і продукт	Тип мережі	Примітки
1	2	3
<i>American HYTech, Prophecy</i>	ЛМ	Оцінювання продуктивності при роботі з текстовими і графічними даними по окремих сегментах і мережі в цілому
<i>CACI Product, COMNET III</i>	ЛМ, ГМ	Моделює мережі X.25, ATM, Frame Relay, зв'язку LAN-WAN, SNA, DECnet, протоколи OSPF, RIP. Доступ CSMA / CD і токені доступ, FDDI і ін Вбудована бібліотека маршрутизаторів 3COM, Cisco, DEC, HP, Wellfleet, ...

Продовження табл. 2.1

1	2	3
<i>Make System, NetMaker XA</i>	ЛМ, ГМ	Перевірка даних про топологію мережі; імпорт інформації про трафік, одержуваної в реальному часі
<i>NetMagic System, StressMagik</i>	ЛМ	Підтримка стандартних тестів виміру продуктивності; імітація пікового навантаження на файл-сервер
<i>Network Analysis Center, MIND</i>	ГМ	Засіб проектування, оптимізації мережі, містить дані про вартість типових конфігурацій з можливістю точного оцінювання продуктивності
<i>Network Design and Analysis Group, AutoNet / Designer</i>	ГМ	Визначення оптимального розташування концентратора в ГМ, можливість оцінки економії коштів за рахунок зниження тарифної плати, зміни постачальника послуг та оновлення обладнання; порівняння варіантів зв'язку через найближчу і оптимальну точку доступу, а також через міст і місцеву телефонну мережу
<i>Network Design and Analysis Group, AutoNet / MeshNET</i>	ГМ	Моделювання смуги пропускання і оптимізація витрат на організацію ГМ шляхом імітації пошкоджених ліній, підтримка тарифної сітки компаній <i>AT & T, Sprint, WiTel, Bell</i>
<i>Network Design and Analysis Group, AutoNet / Performance-1</i>	ГМ	Моделювання продуктивності ієрархічних мереж шляхом аналізу чутливості до тривалості затримки, часу відповіді, а також вузьких місць у структурі мережі
<i>Network Design and Analysis Group, AutoNet / Performance-3</i>	ГМ	Моделювання продуктивності багатопрокольних об'єднань локальних і глобальних мереж; оцінювання затримок у чергах, прогнозування часу відповіді, а також вузьких місць у структурі мережі; облік реальних даних про трафік, що надходять від мережевих аналізаторів
<i>System & Networks, BONES</i>	ЛМ, ГМ	Аналіз впливу додатків клієнт-сервер і нових технологій на роботу мережі
<i>MIL3, Opnet</i>	ЛМ, ГМ	Має бібліотеку різних мережевих пристроїв, підтримує анімацію, генерує карту мережі, моделює смугу пропускання.

Зазвичай імітаційна модель будується не з нуля. Існують готові імітаційні моделі основних елементів мереж: найбільш поширених типів маршрутизаторів, каналів зв'язку, методів доступу, протоколів і т.п. Ці моделі окремих елементів мережі створюються на підставі різних даних: результатів тестових випробувань реальних пристроїв, аналізу принципів їх роботи, аналітичних співвідношень. У результаті створюється бібліотека типових елементів мережі, які можна налаштовувати за допомогою заздалегідь передбачених в моделях параметрів.

Системи імітаційного моделювання зазвичай включають також набір засобів для підготовки вихідних даних про досліджувану мережі - попередньої обробки даних про топологію мережі і вимірюваний трафік. Ці кошти можуть бути корисні, якщо мережа, що моделюється, являє собою варіант існуючої мережі і є можливість провести в ній вимірювання трафіку і інших параметрів, потрібних для моделювання. Крім того, система забезпечується засобами для статистичної обробки отриманих результатів моделювання.

2.6. Вибір інструментарію для реалізації структури апаратного забезпечення захищеної від випромінювання частини системи теледіагностики

На шляху масового впровадження пористих мереж варті уваги кілька завдань, які необхідно вирішити. Як буде працювати мережа в багатоповерховій будівлі? Чи буде передача між поверхами? Чи можлива робота передачі по ланцюжку? Яка реальна дальність передачі даних всередині будівлі по прямої видимості?

Для наочної відповіді на питання були проведені тести в 9-поверховій будівлі клініки (будівлі радянських часів). Для тестів була обрана платформа *Mi Smart Home Multifunction Gateway 3* (оновлена версія шлюзу від *Xiaomi*), Так як вона має дуже зручний форм-фактор для польових випробувань (рис. 2.4). Виробником також надається готовий набір ПЗ і додатки конфігурації для *Android, iOS*.



Рис. 2.4. Вигляд *Mi Smart Home Multifunction Gateway 3*

Особливості *Mi Smart Home Multifunction Gateway 3*:

- інтерфейси *Wi-Fi* 2,4 ГГц, *Zigbee* 3.0 та *Bluetooth* 5.0 *Mesh*;
- дальність зв'язку з іншими приладами – до 200 метрів;
- звукова сигналізація і сповіщення при спрацьовуванні датчиків;
- керування через мобільний додаток;
- сумісність з *Mi Home* та *Apple HomeKit*;
- живлення від будь-якого джерела з інтерфейсом *Micro-USB*.

Тест проводився в конфігурації, коли всі пристрої були об'єднані в одну мережу. При цьому один пристрій виконував роль клієнта (видавця), який посилав на адресу 0xC001 команду перемикання світлодіода при натисканні на кнопку, а всі інші п'ять були підписані на адресу 0xC001 і при отриманні команди запалювали або гасили світлодіод. Всі пристрої в мережі були сконфігуровані, як роутери (ретранслятори).

Команда увімкнути / вимкнути відправляється з періодом 1 раз в секунду з 7 поверху. Таким чином натискання на одну кнопку повинно привести до запалювання світлодіодів на всіх пристроях, можна наочно встановити дальність проходження сигналу по всьому ланцюжку.

Тестувалася передача даних у вертикальній площині будівлі (між поверхами), так як міжповерхові перекриття складаються з набагато більш товстого і міцного бетону, і якщо мережа буде працювати в таких умовах, то мережа гарантовано буде працювати і по поверху в горизонтальній площині.

Пристрої були розташовані на кожному поверсі будівлі в зоні діаметром приблизно 8-10 метрів. На шляху проходження сигналу між деякими вузлами до поверховим горизонтальним перекриттям додаються 2-3 вертикальні перегородки, в залежності від місця розташування пристроїв, так як пристрої були розташовані не строго по прямій, а в різних кабінетах (рис. 2.5).

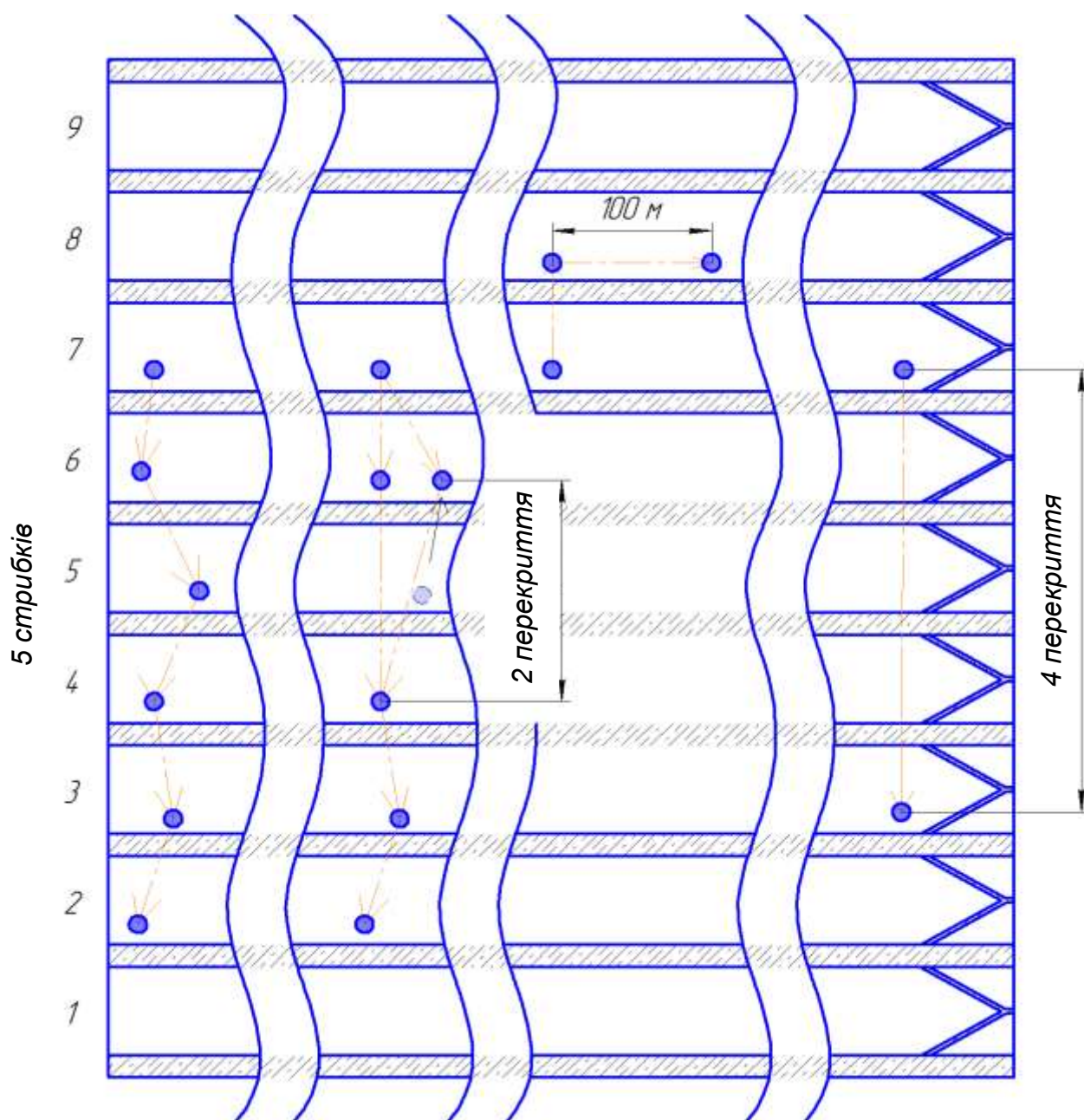


Рис. 2.5. Розріз будівлі для тестування ретрансляторів *mesh*-мережі

Були протестовані наступні 4 конфігурації розташування вузлів мережі (див. рис. 2.5):

– тестування роботи по ланцюжку. Пристрої розташовані на кожному

поверсі будівлі з 7 по 2 поверх. Отримано дальність в 5 стрибків. Пристрої на всіх поверхах відпрацьовують команду без збоїв;

– тестування максимальної пробивної здатності (кількості перекриттів через яких проходить сигнал). Роутер, розташований на 5 поверсі, переміщається на 6 поверх з утворенням стрибка між вузлами в два поверхових перекриття (з 6 на 4 поверх). Мережа працює при стрибку через два міжповерхових перекриття. Збільшення довжини стрибка в три поверхових перекриття призводить до зникнення зв'язку.

– тестування максимальної дальності зв'язку на поверху в прямій видимості. На 8 поверсі з одного боку коридору встановлюється вузол мережі, другий вузол переміщається по коридору до моменту зникнення мережі. Дальність зв'язку в прямої видимості - до кінця коридору (100 метрів). При зміщенні пристрою з лінії прямої видимості зв'язок втрачається;

– тестування проходження сигналу в умовах сходових прольотів. Ретранслятор розташовується на сходовій клітці 7-го поверху, отримує сигнал і спускається сходами вниз до моменту зникнення мережі. Досягнуто стабільна зв'язок при стрибку на 4 поверхи (з 7 до 3).

2.7. Вибір реактивного веб-фреймворку для розробки інтерфейсу системи

Фреймворки розробляються для того, щоб спростити нам життя і звільнити від написання однотипного коду. Але, у міру того як кодова база деяких фреймворків сильно розростається, вони починають привносити свою частку якої складності в проект. Через це при плануванні розробки ми повинні враховувати два фактори:

- складність нашого застосування;
- складність фреймворку, який ми використовуємо.

Трохи абстрагуємося і назвемо це внутрішньої складністю (тобто складністю, від якої ми не можемо позбутися при розробці) і складністю

інструменту, за допомогою якого ми досягаємо тієї чи іншої мети. Природно маніпулювати ми можемо тільки складністю інструменту.

З вищесказаного, у нас виходить два варіанти вибору інструменту:

– варіант недостатності. Коли інструменту недостатньо, щоб перекрити внутрішню складність. Функції, необхідні для реалізації програми, відсутні в фреймворку, і нам доводиться вручну допрацьовувати і додавати необхідний інструментарій.

– Варіант надмірності. Коли необхідний для додатка функціонал перекривається тільки невеликою частиною інструменту. Інша ж частина інструментарію висить мертвим вантажем і тільки створює нам проблеми: обмеження при розробці, уповільнення завантажити програму і т.п.

При плануванні нам необхідно шукати золоту середину так, щоб складність (і, отже, функціональність) додатки були на одному рівні.

Аналіз фреймворку *Vue.js*

Творцем *Vue.js* є *Evan You*, колишній співробітник *Google* і *Meteor Dev Group*. Почав він розробляти фреймворк в 2013-му, а в лютому 2014 го відбувся перший публічний реліз. *Vue* широко використовується серед китайських компаній, наприклад: *Alibaba*, *Baidu*, *Xiaomi*, *Sina Weibo* і ін. Він входить в ядро *Laravel* і *PageKit*. Нещодавно вільна система управління репозиторіями *GitLab* теж перейшла на *Vue.js*.

В кінці вересня 2016- го вийшов в реліз *Vue.js* 2.0, ще крутіше і з упором на продуктивність — тепер використовується віртуальний *DOM*, підтримується серверний рендеринг, можливість використовувати *JSX* і т.д. Хоча зараз він підтримується тільки співтовариством, він тримається гідно навіть на рівні продуктів таких гігантів, як *Google* і *Facebook* (*Angular2* і *React* 15), і поступово наздоганяє їх по популярності.

Основними складові *Vue* є:

- конструктор;
- компоненти;
- директиви;

– переходи.

Конструктор

app.js

```
new Vue({
  el: 'jQueryStyleSelector',
  template: '<id || inline template>',
  data: {
    props: 'Это видно в шаблонах',
    first_name: "Вася",
    last_name: "Пупкин"
  },
  computed: {
    full_name: function(){
      return this.first_name + this.last_name; //Вася Пупкин
    }
  },
  methods: {
    // методы жизненного цикла
    beforeCreate: function() {},
    created: function() {},
    beforeMount: function() {},
    mounted: function() {},
    beforeUpdate: function() {},
    updated: function() {},
    beforeDestroy: function() {},
    destroyed: function() {},

    customMethodsAlso: function(){
      //здесь у нас тоже есть доступ к data
    }
  }
})
```

Рис. 2.6. Конструктор *Vue.js*, основні методи та концепції.

Робота з *Vue.js* починається зі створення нового інстансу *new Vue*. У *el* у нас елемент, за яким стежить *Vue*. У *template* обраний елемент, куди *Vue* буде рендерити додаток. У *data* зберігається поточний стан компонента, а метод *computed* надає нам обчислювані властивості [24].

Vue – дуже зручний фреймворк для старту цього проекту! Можна просто підключити його на сторінці і почати працювати без необхідності налаштовувати систему збирання. З ним дуже просто почати працювати, навіть якщо ви ніколи не працювали з *JavaScript* фреймворками. Це ідеальне поєднання зручності і потужності.

Висновки за розділом

В даному розділі було проведено вибір інструментарію для моделювання та реалізації прикладного програмно-апаратного забезпечення системи теледіагностики.

За мережну архітектуру було обрано *BLE Mesh*, тому що результати, які були отримані при використанні ретранслятора *Mi Smart Home Multifunction Gateway 3* у старій будівлі з бетонними перекриттями показали.

- при вибудовуванні пристроїв ланцюжком на кожному поверсі дані передаються стабільно;

- мережа працює при стрибку (відстані між вузлами) в 2 міжповерхових перекриття;

- при розташуванні пристроїв на сходах зв'язок зберігається при стрибку в 4 поверхи;

- дальність передачі даних по поверху в прямої видимості в умовах виробничої будівлі досягає 100 метрів.

Фреймворком для розробки інтерфейсу ПЗ було обрано *Vue.js* за реактивність,, яка дозволить інтерфейсу реактивно оновлювати дані без необхідності перезавантаження сторінки і велику кількість готових шаблонів розробки.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ЗБИРАННЯ, ОБРОБКИ ТА ЗБЕРЕЖЕННЯ ДАНИХ В ТЕЛЕДІАГНОСТИЧНІЙ СИСТЕМІ

Одним з перспективних напрямків розвитку портативної електроніки є пристрої медичного призначення, які відстежують один або кілька параметрів життєдіяльності людського організму і володіють такими функціями як, наприклад, зняття ЕКГ, вимірювання частоти серцевих скорочень, частоти дихання, температури тіла, підрахунок кількості кроків, вимірювання кількості цукру в крові і так далі. Потенціал технології портативних медичних пристроїв величезний – від простого моніторингу стану їх власника до розподіленого збору статистики для визначення ранніх ознак захворювань, оскільки початок багатьох захворювань може починатися відносно непомітно і супроводжуватися, на перший погляд, непов'язаними між собою ознаками.

3.1. Описання розгортання системи в медичному закладі

Одним з основних показників нормально функціонуючого організму є насиченість артеріальної крові киснем. Цей параметр відображається на числі еритроцитів, а визначити його допомагає пульсоксиметрія (пульсова оксиметрія).

У терапевтичній практиці пульсоксиметрія проводиться в разі патології органів дихання, порушеннях сну з зупинкою дихання, передбачуваному ціанозі різної етіології, з метою контролю терапії хронічної патології.

Постійне спостереження пульсу і вмісту кисню в крові може допомогти виявити схильність до вказаних нездужань і вжити превентивні заходи.

Елементна база, яку пропонує компанія *Texas Instruments*, дозволяє реалізувати безліч варіантів датчиків пульсу і вмісту кисню в крові з різним функціоналом. Одним з найбільш бажаних для кінцевого користувача варіантів є наявність у портативного датчика бездротового інтерфейсу, що дозволяє при

необхідності відображати дані в реальному часі, вести їх облік або переглядати статистику.

Проект *TIDA-010029* (рис. 3.1) є опорною розробкою бездротового безконтактного пульсоксиметра з інтерфейсом *BLE*, що вимірює вміст кисню в крові (*SpO2*) і частоту серцевих скорочень (*HRM*).

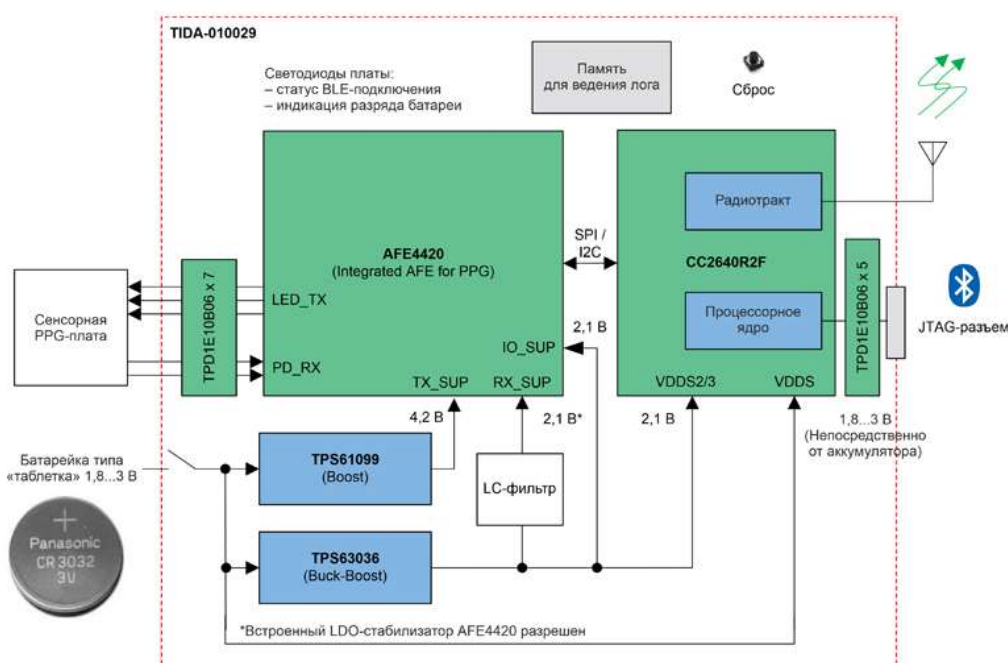


Рис. 3.1. *TIDA-010029* - розробка бездротового безконтактного пульсоксиметра з інтерфейсом *BLE*

У його апаратної частини є мікросхема *AFE4420*, яка представляє собою однокристалний пристрій для вимірювань за методом фотоплетізмографія (*PPG*). *AFE4420* підтримує до чотирьох світлодіодів і фотодіодів, що дозволяє отримувати до 16 різних вимірювальних комбінацій з корекцією засвічення. Бездротова система-на-кристалі (СНК) *CC2640R2F* передає виміряні дані для подальшої обробки. На виході доступні необроблені дані для розрахунку частоти серцевих скорочень, *SpO2* та інших пов'язаних параметрів [12]. Два вбудованих світлодіоди індикують виявлення критичного розряду батареї і наявність з'єднання *Bluetooth*. Пристрій має батарейне живлення від елемента типу *CR3032*.

AFE4420 і *CC2640R2F* зв'язуються за допомогою інтерфейсів *SPI* або *I2C*. Обидва зазначених ключові компоненти забезпечують високу продуктивність і виконання необхідних функцій при невисоких, що важливо для портативних пристроїв, витратах енергії. Так, при рекомендованій для проекту батареї

тривалість автономної роботи складе близько 30 днів при передачі набору даних один раз в секунду.



Сумарне енергоспоживання *TIDA-010029* залежить від періодичності передачі даних по радіоканалу, режиму роботи *AFE4420* (частоти опитування датчиків) і робочого струму світлодіодів. Вимірювання дають значення споживаного струму в межах 5 мА, що при ємності *CR3032* в 500 мА дає близько 98 годин роботи.

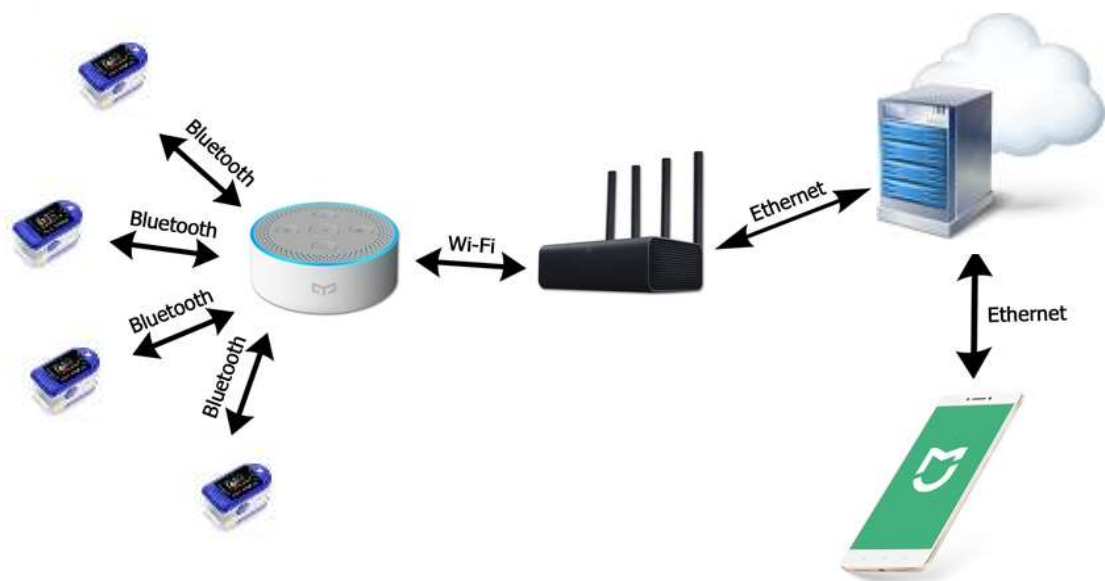


Рис. 3.3. Схема передачі даних на мобільні пристрої з системи теледіагностики

Дана схема передбачає отримання показників з датчиків, які в свою чергу самоорганізовані у *mesh*-мережу. Показники через ретранслятори і роутер, що має вихід у мережу Інтернет, передаються на хмарний сервер, звідки ці дані можуть зчитувати користувачі програмного додатку, який через *web*-інтерфейс відображає дані з серверу.

В даній структурі не передбачено синхронізацію зняття показників з датчиків та випадки відсутності зв'язку внутрішньої мережі з віддаленим сервером. Даний вибір архітектури розгортання обумовлено тим, що це експериментальна система, яка надає можливість ознайомитись користувачам з можливостями використання мережі теледіагностики та має низьку вартість за рахунок відсутності витрат на власний сервер в організації і додаткове ПЗ збереження даних на проміжному сервері.

Для тестування зони покриття використовувалась програмна система *NetSpot*. На рис. 3.4 вказано точки встановлення шлюзів, які було описано в розділі 2.6, а на рис. 3.5 представлено карту рівня сигналів (червоний – максимальний, жовтий – середній).

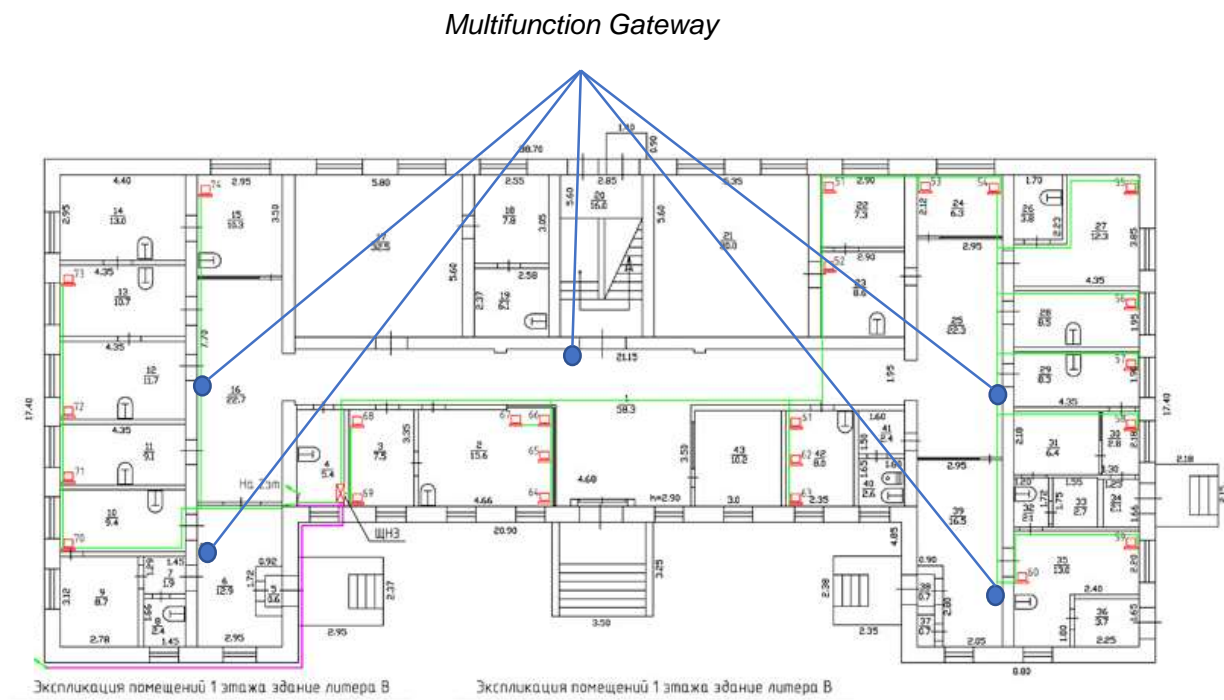


Рис. 3.4. Карта першого поверху медичного закладу з розташуванням шлюзів

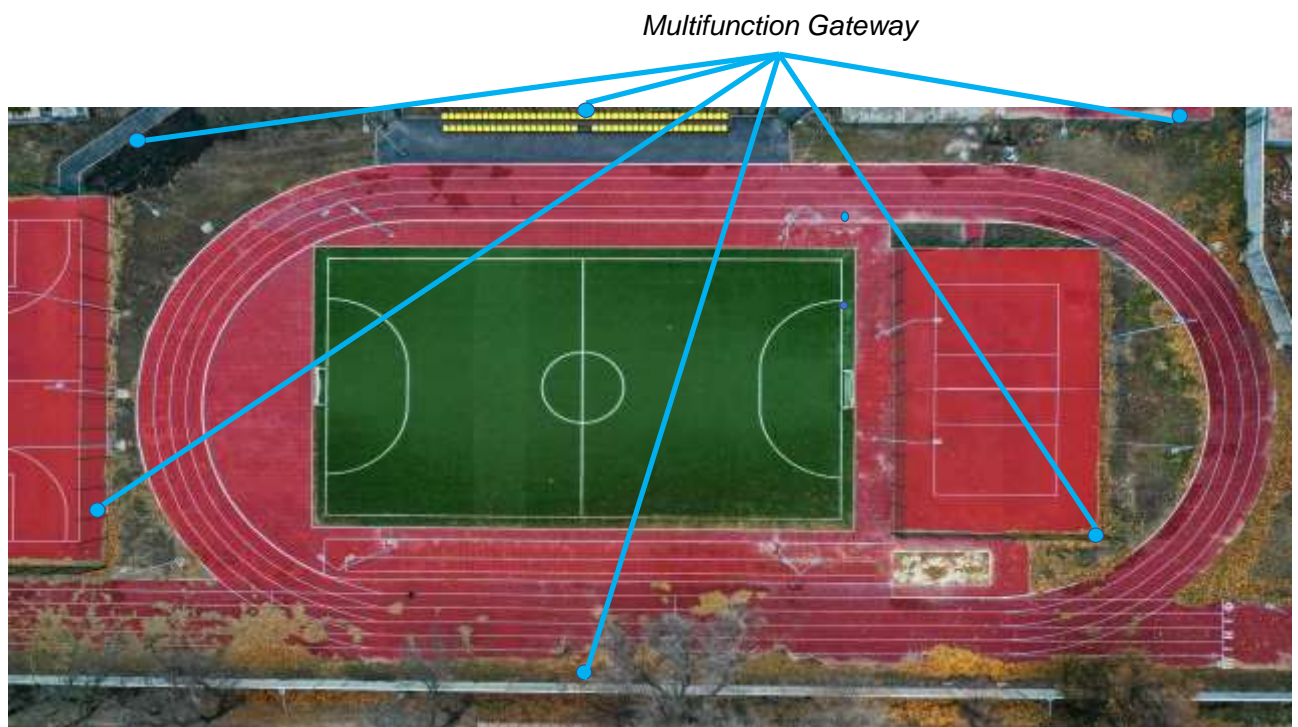


Рис. 3.5. Розташування шлюзів на стадіоні

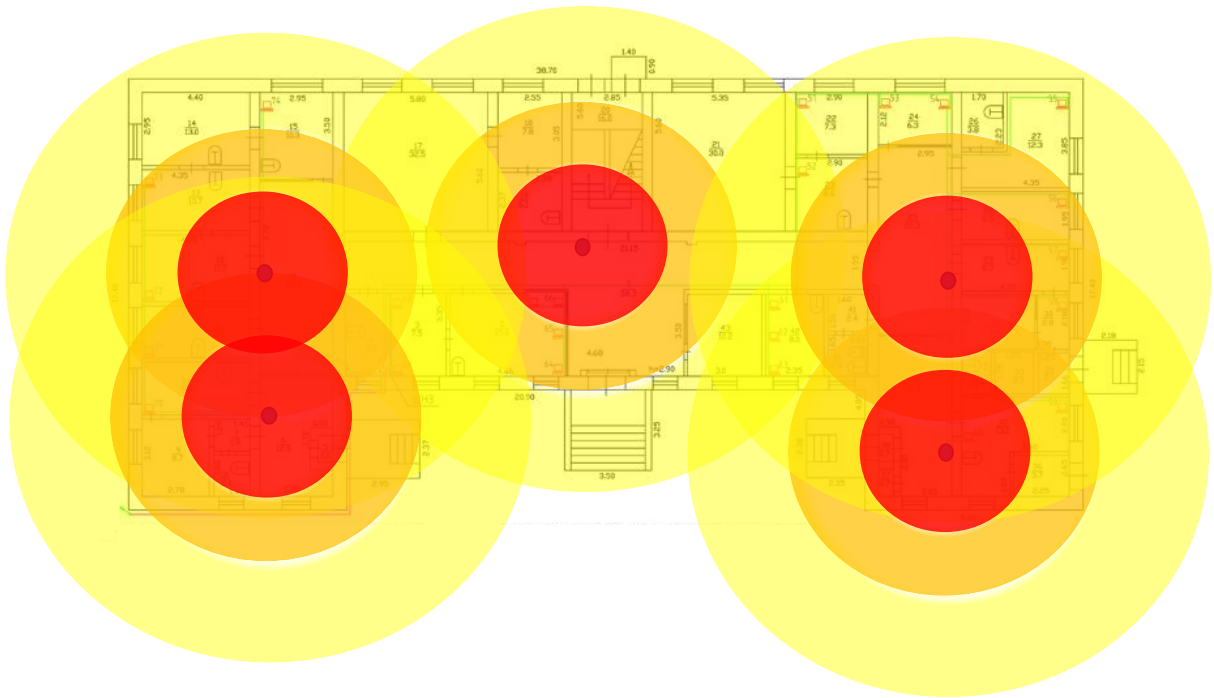


Рис. 3.6. Теплова карта рівня радіо-сигналу у приміщенні медичного закладу

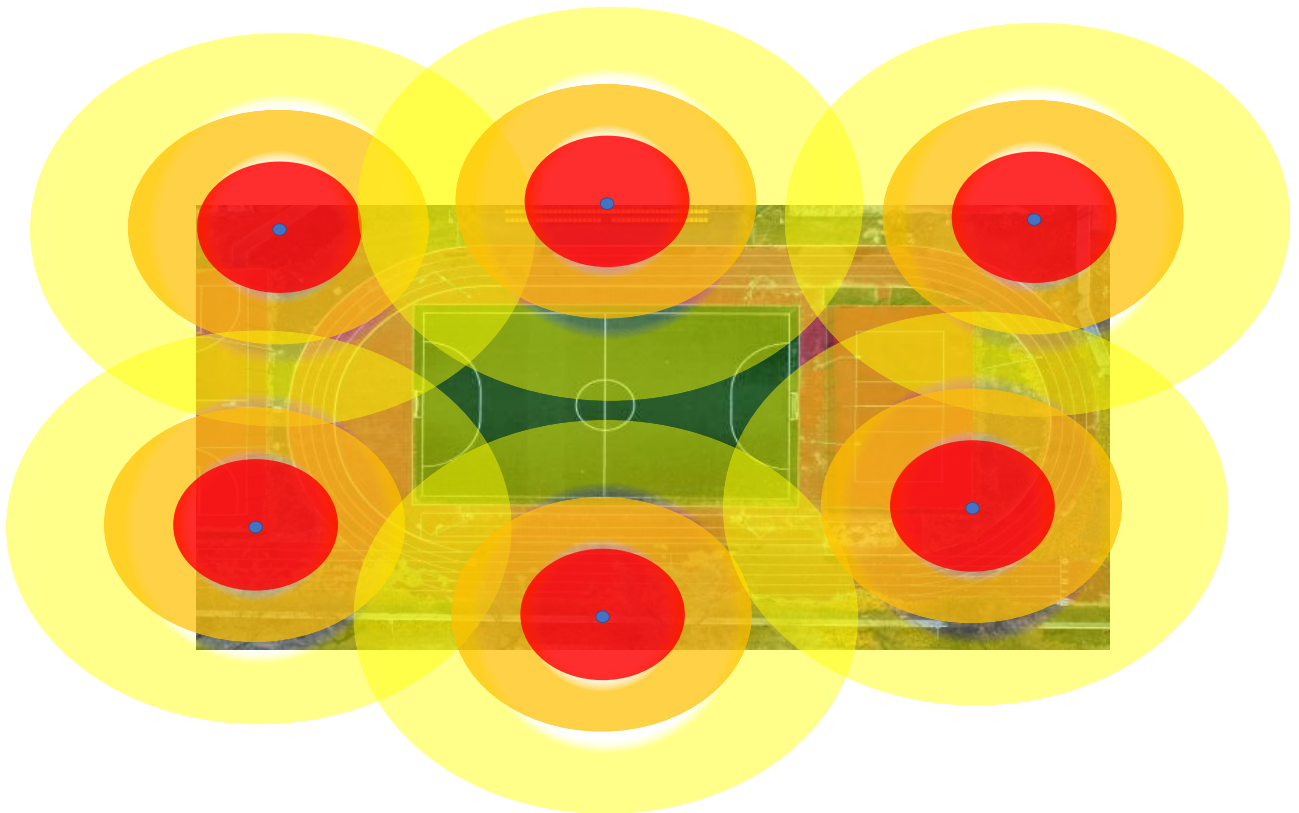


Рис. 3.7. Теплова карта рівня радіо-сигналу на стадіоні

На стадіоні за рахунок умов використання на відкритому просторі і низького рівня електро-магнітних шумів рівень сигналу від шлюзів вище, ніж у закритих приміщеннях.

Для реалізації системи з повним покриттям приміщень медичного закладу і стадіону необхідно 51 шлюзів *Mi Smart Home Multifunction Gateway 3* (45 для лікарні, 6 – для стадіону). З урахуванням оплати за встановлення і налаштування шлюзів, технічне забезпечення проекту коштує 51 000 грн.

Після розгортання системи на різних поверхах будівлі (див. розділ 2.6) було перевірено роботу мережі пристроїв (рис. 3.8) та команди обміну повідомленнями (рис. 3.9).



Рис. 3.8. Результати підключення мережі пристроїв

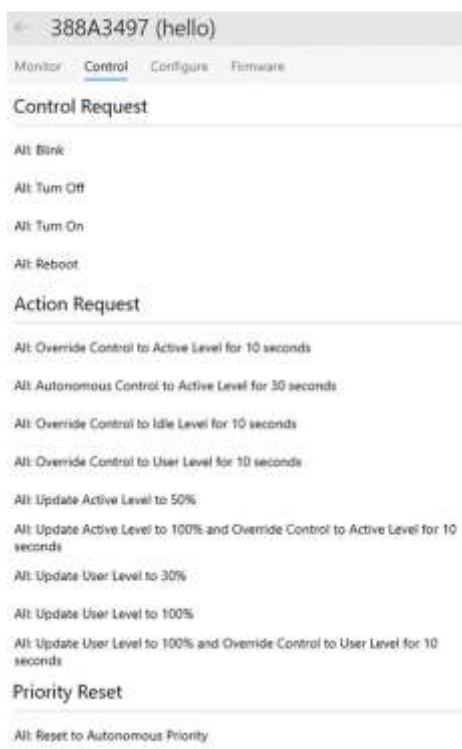


Рис. 3.9. Результат обміну повідомленнями в системі теледіагностики

3.2. Розробка інтерфейсу системи

Для початку потрібно створити проект. Для цього більш зручно буде використати *vue-cli*. *Vue CLI* – повноцінна система для швидкої розробки на *Vue.js*, що надає [24]:

- Інтерактивне створення проекту через *@ vue / cli*;
- прототипування через *@vue/cli+@vue/cli-service-global* без змін.

Runtime-залежність (*@ vue / cli-service*) надає [24]:

- можливість поновлення;
- створена поверх *webpack*, з оптимальними настройками за замовчуванням;
- налаштування за допомогою конфігураційного файлу в проекті;
- велика колекція офіційних полігонів, інтегруючих кращі інструменти екосистеми фронтенда;
- повноцінний графічний користувальницький інтерфейс для створення і управління проектами *Vue.js*.

У *methods* можна виділити наступні кастомні методи і методи життєвого циклу *Vue*:

- *beforeCreate* – дивиться дані і ініціалізує події;
- *created* – дивиться, чи є *el* або *template*. Якщо є, то рендерить в них; якщо немає, то шукає метод *render*;
- *beforeMount* – створює *vm*. \$ *el* і замінює в них *el*;
- *mounted* – елемент був зрендерений.

При зміні стану (рис. 3.12) [25]:

- *beforeUpdate* – знову рендерить *VDOM* і порівнює з реальним *DOM*-му, застосовує зміни;
- *updated* – зміни отрендерити;
- *beforeDestroy* – повний демонтаж вотчерів, внутрішніх компонентів і слухачів подій;
- *destroyed* – викликається, коли виконання операції зупиняється.

```
Ivan@DESKTOP-DVMHBN0 MINGW64 /c/University/heart-rate-app (master)
$ yarn serve
yarn run v1.13.0
$ vue-cli-service serve
[INFO] Starting development server...
40% building 133/134 modules 1 active ...ef--0-0!C:\University\heart-rate-app\node_modules\vue-loader\lib\index.js??vue-loader-options!C:\University\heart-rate-
40% building 133/135 modules 2 active ..._modules\vue-loader\lib\index.js??vue-l
oader-options!C:\University\heart-rate-app\src\components\HeartRate.vue?vue&type
98% after emitting CopyPlugin

[DONE] Compiled successfully in 2740ms 20:42:15

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.0.101:8080/

Note that the development build is not optimized.
To create a production build, run yarn build.
```

Рис. 3.10. Запуск проекту за допомогою *Vue-cli* [25]

3.3. Розробка програмного функціоналу підключення смарт-браслету за допомогою технології *Web Bluetooth*

В даному проекті основна задача дизайну – демонстрація поточного пульсу та демонстрація графу рівнів пульсу.

Web Bluetooth – це нова технологія, яка з'єднує *Web* з Інтернетом Речей.

По суті, *Web Bluetooth* дозволяє вам керувати будь-яким пристроєм *Bluetooth Low Energy* (розумної побутовою технікою, медичними приладами, такими як монітор серцевого ритму, монітор глюкози, датчики температури і т.д.) Безпосередньо з вашого ПК або смартфона без необхідності установки програми.

Web Bluetooth, в кінцевому підсумку, дозволяє розробникам створювати одне рішення, яке буде працювати на всіх платформах, включаючи як мобільні, так і настільні, що означає більш низькі витрати на розробку, більше інтерфейсів управління з відкритим вихідним кодом для різних фізичних продуктів і більше інновацій.

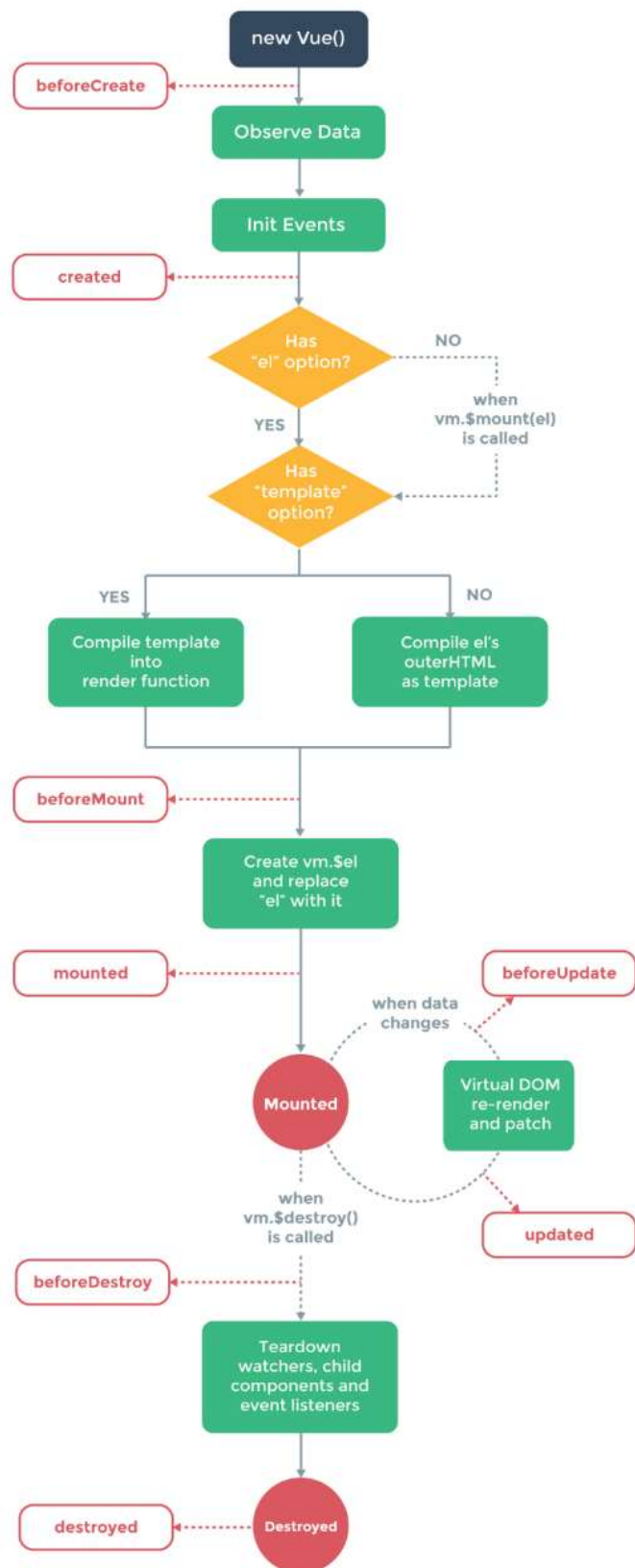


Рис. 3.11. Принцип роботи компонента *Vue.js* при зміні стану компонентів

Це послідовність дій, яка буде поширюватися для всіх додатків з *Web Bluetooth*. Основні кроки:

- пошук відповідного пристрою;
- підключення до нього;
- вибір *Service*, який вас цікавить;
- викидами *Characteristic*, яка вас цікавить;
- читання, запис або підписка на *Characteristic* (рис. 3.12).

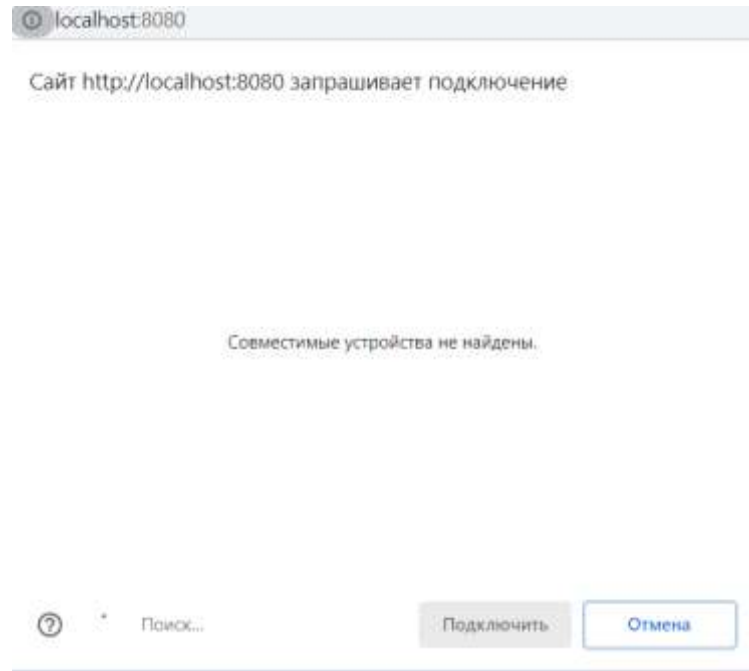


Рис. 3.12. Результат роботи вищевказаної функції

Лістинг функції підключення до пульсометра:

```
async connect() {
  try {
    const device = await navigator.bluetooth.requestDevice({
      filters: [{ services: ['heart_rate'] }],
    });
    const server = await device.gatt.connect();
    const service = server.getPrimaryService('heart_rate');
    const characteristic = service.getCharacteristic('heart_rate_measurement');
    await characteristic.startNotifications();
    characteristic.addEventListener('characteristicvaluechanged',
      this.handleCharacteristicValueChanged.bind(this));
  }
}
```

```

    } catch (error) {
        console.log(error);
    }
}

```

3.4. Розробка функціоналу для динамічного отримання даних зі смарт-браслету (пульсоксиметра) та їх аналіз

Для того, щоб отримувати дані з пульсометра динамічно, потрібно написати функцію-прослуховувач, яка буде «слухати» та приймати нові дані з пульсометра.

Лістинг функції прослуховача *handleCharacteristicValueChanged*:

```

characteristic.addEventListener('characteristicvaluechanged',
this.handleCharacteristicValueChanged.bind(this))

```

```

pushPatientData(i, v) {
    this.heartRatePatientData[i].push({ x: Date.now(), y: v });
    this.heartRateData = [].concat(this.heartRateData);
},
handleCharacteristicValueChanged(e) {
    const heartRate = parseHeartRateData(e.target.value).heartRate;
    this.heartRate = heartRate;
    switch (heartRate) {
        case heartRate > 104 && heartRate < 114:
            this.pushPatientData(1, heartRate);
            break;
        case heartRate > 114 && heartRate < 133:
            this.pushPatientData(2, heartRate);
            break;
        case heartRate > 133 && heartRate < 152:
            this.pushPatientData (3, heartRate);
            break;
    }
}

```

```

    case heartRate > 152 && heartRate < 172:
      this.pushPatientData pushData(4, heartRate);
      break;
    case value > 172:
      this.pushPatientData (5, heartRate);
      break;
    default: this.pushPatientData (0, heartRate);
  }
},
},
},

```

Графік поточного пульсу (рис. 3.13) та різних рівнів пульсу (рис. 3.14) будемо створювати за допомогою бібліотеки *vue-chartjs* (програмний код представлено в Додатку А).



Рис. 3.13. Веб-інтерфейс системи моніторингу пульсу пацієнта

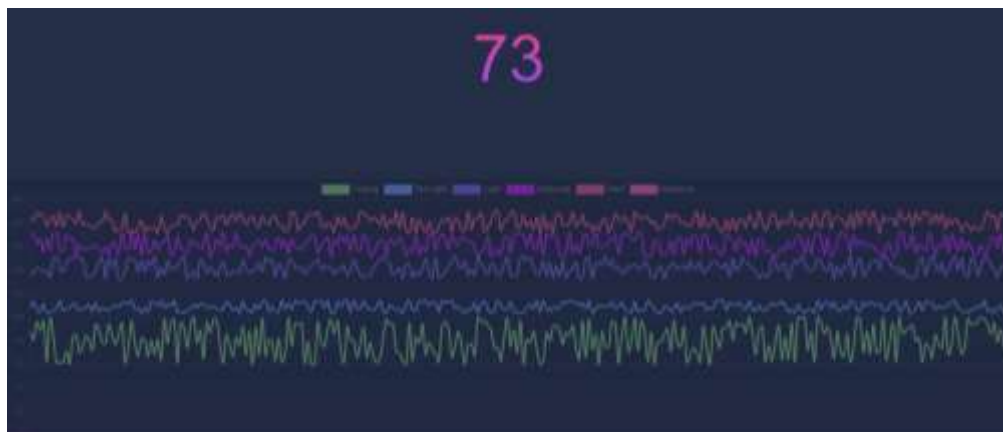


Рис. 3.14. Результат роботи програми. Поточний пульс пацієнта та графік

Висновки за розділом

В даному розділі покроково описано розгортання системи з описанням пристроїв, які задіяні при цьому. Для зняття показників з бездротових безконтактних пульсоксиметрів *TIDA-010029* було обрано архітектуру мережі *mesh-BLE*, в якій у якості шлюзів використовувались *Mi Smart Home Multifunction Gateway 3*.

В даній структурі не передбачено синхронізацію зняття показників з датчиків та випадки відсутності зв'язку внутрішньої мережі з віддаленим сервером. Даний вибір архітектури розгортання обумовлено тим, що це експериментальна система, яка надає можливість ознайомитись користувачам з можливостями використання мережі теледіагностики та має низьку вартість за рахунок відсутності витрат на власний сервер в організації і додаткове ПЗ збереження даних на проміжному сервері.

Для тестування зони покриття використовувалась програмна система *NetSpot*. За рахунок якої визначено зони з низьким рівнем радіосигналу.

Web-інтерфейс реалізовано у фреймворку *Vue.js* та адаптовано для показу на мобільних пристроях.

В подальших планах передбачено впровадження внутрішнього серверу з спеціалізованим ПЗ для реалізації функцій попередньої обробки результатів теледіагностики, синхронізації роботи датчиків і забезпечення роботи системи у разі відсутності зв'язку з зовнішнім сервером.

ВИСНОВКИ

Для реалізації системи теледіагностування було проведено системний аналіз апаратно-програмних методів і засобів роботи подібних систем в умовах гібридного середовища передачі даних. Даний аналіз показав можливість використання для даних потреб пористих мереж (*mesh*) та за результатами порівняльного аналізу було обрано технологію *BLE Mesh*. Яка забезпечує максимальну дальність серед розглянутих мереж, підтримує найбільшу кількість вузлів в мережі, але орієнтована на передачу невеликих обсягів інформації.

Визначено, що для оптимальної роботи мережі *BLE Mesh* необхідна оптимізація кількості вузлів ретрансляторів, а також їх розташування.

У другому розділі було проведено вибір інструментарію для моделювання та реалізації прикладного програмно-апаратного забезпечення системи теледіагностики.

У мережі було вирішено використовувати ретранслятори *Mi Smart Home Multifunction Gateway 3*, які були протестовані у багатоповерховій старій будівлі з бетонними перекриттями.

Фреймворком для розробки інтерфейсу ПЗ було обрано *Vue.js* за реактивність,, яка дозволить інтерфейсу реактивно оновлювати дані без необхідності перезавантаження сторінки і велику кількість готових шаблонів розробки.

У третьому розділі покроково описано розгортання системи з описанням пристроїв, які задіяні при цьому. Передбачено схему розгортання, яка передбачає отримання показників з датчиків, організованих у *mesh*-мережу. Показники через ретранслятори і роутер, що має вихід у мережу Інтернет, передаються на хмарний сервер, звідки ці дані можуть зчитувати користувачі програмного додатку, який через *web*-інтерфейс відображає дані з серверу.

В даній структурі не передбачено синхронізацію зняття показників з

датчиків та випадки відсутності зв'язку внутрішньої мережі з віддаленим сервером. Даний вибір архітектури розгортання обумовлено тим, що це експериментальна система, яка надає можливість ознайомитись користувачам з можливостями використання мережі теледіагностики та має низьку вартість за рахунок відсутності витрат на власний сервер в організації і додаткове ПЗ збереження даних на проміжному сервері.

Для тестування зони покриття використовувалась програмна система *NetSpot*. За рахунок якої визначено зони з низьким рівнем радіосигналу.

Web-інтерфейс реалізовано у фреймворку *Vue.js* та адаптовано для показу на мобільних пристроях.

В подальших планах передбачено впровадження внутрішнього серверу з спеціалізованим ПЗ для реалізації функцій попередньої обробки результатів теледіагностики, синхронізації роботи датчиків і забезпечення роботи системи у разі відсутності зв'язку з зовнішнім сервером.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – 39 с.
2. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету / С.В. Бойченко, О.В. Іванченко – К.: НАУ, 2017. – 63 с.
3. Akyildiz I.F. *Wireless multimedia sensor networks: applications and testbeds* / Akyildiz I. F., Melodia T., Chowdury K.R. // *Proceedings of the IEEE (invited paper)*, 2008. – Vol. 96. – № 10 – Pp. 1588-1605.
4. Akyildiz I.F., Vuran M.C. *Wireless Sensor Networks* / I.F. Akyildiz, M.C. Vuran – New York: John Wiley & Sons, 2010. – 571 p.
5. Basagni S. *Mobility – Adaptive Protocols for Managing Large AD Hoc Network* / S. Basagni, D. Turgut, S.K. Das // *Proceedings of the IEEE International Conference on Communications (ICC)*. – 2011.-P. 1539-1543.
6. Broch, J. *A performance comparison of multihop wireless ad hoc network routing protocols* / J. Broch, D.A. Maltz, D.B. Johnson, J. Jetcheva // *Proc. ofMOBICOM'08*. – 2008.
7. Kawadia V. *System services for implementing Ad-Hoc routing: Architecture, Implementation and Experiences*/ V. Kawadia, Y. Zhang, B. Gupta // *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys)*. San Francisco, CA. – June 2013. – P. 99-112.
8. Mockapetris. *P Domain Names – Concepts and Facilities*. – Network Working Group. – 2017.
9. Perkins, CE. *Ad hoc On-Demand Distance Vector (AODV) Routing* / C.E. Perkins, C.E. BeldingRoyer // *RFC 3561*. – July 2013.
10. Schulman A. *The Extent of Systematic Monitoring of Employee E-mail and Internet Use, US, 2010-2012*. Постійне посилання:

11. Shin, KG. *Performance Analysis of Distributed Routing Strategies Free of Ping-Pong-Type Looping* / K.G. Shin, M. Chen // *IEEE Trans. Computers.* – February 2007. – V. COMP-36, M 2. – P. 129-137.

12. Бодрова А. А., Логвин В.И. Позиционирование и взаимодействие в беспроводных сенсорных сетях / А.А. Бодрова, В.И. Логвин // Молодой ученый. – 2015. – №6. – С. 129-132.

13. Болілій В.О., Котяк В.В. Комп'ютерні мережі. Навчальний посібник / В.О. Болілій, В.В. Котяк – Кіровоград: ЦОП Авангард, 2008. – 146 с.

14. Власенко В.О. Методи самоорганізації безпроводних сенсорних мереж // Сучасний захист інформації №1, 2017. – С. 96-102.

15. Вороненко А.В., Романов В.А. Выбор унифицированных средств процессорной обработки данных на основе *cortex*-ядра для биосенсорных приборов // Комп'ютерні засоби, мережі та системи: зб. наук. пр. / К.: Ін-т кібернетики ім. В. М. Глушкова НАН України. – 2013 (№12). – С. 44-101.

16. Высокопроизводительные сети. Энциклопедия пользователя./ Под ред. Марка А. Спортака: Пер. с англ. – К.: ДияСофт, 2008. – 836 с.

17. Груша В.М., Артеменко Д.М., Пацко О.В. Використання бездротового зв'язку для моніторингу стану насаджень методом індукції флуоресценції хлорофілу / В.М. Груша, Д.М. Артеменко, О.В. Пацко // XVIII Міжнародна конференція з автоматичного управління, 28-30 вересня 2011 року: мат. конф. – Львів: Видавництво Львівської політехніки. – 2011. – С. 392-393.

18. Коцюр А.Б., Казадаєв І.О. Підсистема теледіагностики на основі бездротових технологій / А.Б. Коцюр, І.О. Казадаєв // Тези доповідей наук.-практ. конф. “Сучасні тенденції розвитку системного програмування” (25-26 листопада 2020 р.). – К.: НАУ, 2020. – С. 42.

19. Коробейников А.В. Структура системы мониторинга для медицинских учреждений / Математическое моделирование и

интеллектуальные системы. – Ижевск: ИжГТУ, 2003. – № 1. – С. 18-21.

20. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Учебник. / В.Г.Олифер, Н.А. Олифер – М.:, 2014. – 1026 с.

21. Педжман Р., Джонатан Л. Основы построения беспроводных локальных сетей стандарта 802.11 / Р. Педжман, Л. Джонатан – М.:, 2006 – 315 с.

22. Рагозин Д. В. Моделирование синхронизированных сенсорных сетей // Проблемы програмування. – 2008. – № 2-3. Спеціальний випуск – 721-729 с.

23. Романов В.О. Нова інформаційна технологія експрес-оцінювання стану рослин в умовах дії стресових факторів / Романов В.О., Галелюка І.Б., Вороненко О.В., Груша В.М. // Комп'ютерні засоби, мережі та системи: зб. наук. пр. – К.: Ін-т кібернетики ім. В. М. Глушкова НАН України. – 2016 (№15). – С. 44-101.

24. *Vue.js* для сомневающихся. Все, что нужно знать / Сайт: *Habr* – Режим доступа: <https://habr.com/ru/post/329452/>, вільний.

25. Начинаем разработку с *Web Bluetooth* и *Progressive Web Apps* / Сайт *jem-space*. Режим доступа: <https://jem-space.ru/nachinaiem-razrabotku-s-pomoshchiu-vieb-bluetooth-i-proghriessivnykh-vieb-prilozhienii-kontrol-mira-s-poddierzhkoi-bluetooth-s-komfortom-intiernieta/>, вільний.

26. Стандарты по локальным вычислительным сетям: Справочник./ Под ред. С.И. Самойленко. – М.: Радио и связь, 2000.

Лістинг коду компонента графа

```
<script>
import { Scatter } from 'vue-chartjs';

export default {
  extends: Scatter,
  name: 'HeartRateChart',
  props: {
    values: {
      type: Array,
      default: () => [],
    },
  },
  watch: {
    values() {
      this.updateChart();
    },
  },
  mounted() {
    this.updateChart();
  },
  methods: {
    updateChart() {
      this.renderChart({
        datasets: [
          {
            label: 'Chilling',
```

```

    data: this.values[0],
    backgroundColor: '#4f775c',
    borderColor: '#4f775c',
    showLine: true,
    fill: false,
  },
  {
    label: 'Very light',
    data: this.values[1],
    backgroundColor: '#465f9b',
    borderColor: '#465f9b',
    showLine: true,
    fill: false,
  },
  {
    label: 'Light',
    data: this.values[2],
    backgroundColor: '#4e4491',
    borderColor: '#4e4491',
    showLine: true,
    fill: false,
  },
  {
    label: 'Moderate',
    data: this.values[3],
    backgroundColor: '#6f2499',
    borderColor: '#6f2499',
    showLine: true,
    fill: false,
  },

```

```

{
  label: 'Hard',
  data: this.values[4],
  backgroundColor: '#823e62',
  borderColor: '#823e62',
  showLine: true,
  fill: false,
},
{
  label: 'Maximum',
  data: this.values[5],
  backgroundColor: '#8a426f',
  borderColor: '#8a426f',
  showLine: true,
  fill: false,
},
],
}, {
  animation: false,
  responsive: true,
  maintainAspectRatio: false,
  elements: {
    point: {
      radius: 0,
    },
  },
},
scales:
{
  xAxes: [{
    display: false,

```

```

    }],
    yAxes: [{
      ticks: {
        beginAtZero: true,
        fontColor: '#394365',
      },
      gridLines: {
        color: '#2a334e',
      },
    }],
  },
});
},
},
};
</script>

```

Виклик компонента:

```

<div class=heart-rate-chart-wrapper>
  <HeartRateChart :values='heartRateData' />
</div>

```