МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТУСА

В. Г. Крижановський, С. П. Сергієнко

Енергоефективні пристрої Інтернету речей (IoT)

Навчально-методичний посібник

Вінниця
ДонНУ імені Василя Стуса
2020

УДК 004.77:681.5(075.8)
К 822

Рекомендовано до друку вченою радою факультету інформаційних та прикладних технологій (протокол № 3 від 25 листопада 2020 р.)

Автори:
В. Г. Крижановський, проф. кафедри радіофізики та кібербезпеки;
С. П. Сергієнко, доц. кафедри радіофізики та кібербезпеки.

Рецензент: *П. К. Ніколюк,* д-р фіз.-мат. наук, проф., проф. кафедри комп'ютерних наук та інформаційних технологій.

К 822
Енергоефективні пристрої інтернету речей (IoT): навчально-методичний посібник./ Вінниця: ДонНУ імені Василя Стуса, 2020. 63 с.

Розглядаються особливості побудови систем Інтернету речей (IoT) зі зниженим рівнем споживання енергії, які мають найбільші перспективи широкого впровадження.

Посібник рекомендовано для студентів вищих навчальних закладів за напрямами «Прикладна фізика. Технології Інтернету речей», «Кібербезпека», «Комп'ютерні науки та інформаційні технології. Інтелектуальні інформаційні технології».

УДК 004.77:681.5(075.8)

ВСТУП

Інтернет речей (Internet of Things, IoT) – концепція, яка зародилася з різних джерел, а зараз широкому загалу найбільш відома завдяки поняттям «Розумний будинок» та «Індустріальний Інтернет речей». Фахівці пов'язують виникнення «Інтернету речей» з розвитком сенсорних мереж (мереж давачів) та з різновидами автоматизованих систем управління технологічними процесами. Спільною у цих галузях є потреба забезпечення гнучкого та економічного (з малим споживанням енергії) процесу передачі інформації з невизначеного кола джерел, який до того ж має бути переважно бездротовим. Зрозуміло, що Інтернет як глобальна мережа стає зразком для створення розподіленої, гнучкої, надійної та побудованої з однотипних елементів системи.

Водночас потреба у використанні пристроїв зі зниженим енергоспоживанням породжує нові вимоги до побудови системи зв'язку Інтернету речей, обчислювальних пристроїв та програмного забезпечення для функціонування всієї екосистеми Інтернету речей. У цьому посібнику розглядаються актуальні питання розвитку відповідних технічних систем. Частково питання побудови, експлуатації та безпеки інтернету речей розглядалися в посібниках [1, 2].

1. СТАНДАРТИ ЗВ'ЯЗКУ З НИЗЬКИМ СПОЖИВАННЯМ ЕНЕРГІЇ ДЛЯ ІНТЕРНЕТУ РЕЧЕЙ (IoT)

1.1. Мережі з низькою потужністю, але з великою площею покриття

Мережі з низькою потужністю та широкою площею покриття (LPWA) являють собою нову парадигму спілкування, яка доповнить традиційні стільникові та бездротові технології для задоволення різноманітних вимог застосування IoT. Технології LPWA пропонують унікальний набір функцій, включно із широкосмуговим підключенням для пристроїв малої потужності та низьку швидкість передачі даних, не передбачених застарілими бездротовими технологіями. Очікується, що їх ринок буде величезним. Приблизно четверта частина загальних 30 мільярдів пристроїв IoT/M2M має бути підключена до Інтернету за допомогою мереж LPWA і за допомогою власних або стільникових технологій [3]. На рис. 1.1 висвітлено різноманітність застосувань у кількох галузях бізнесу, які можуть використовувати технології LPWA для підключення своїх кінцевих пристроїв. Ці сектори діяльності включають, але не обмежуються цим: розумне місто, програми IoT для персонального використання, інтелектуальну мережу, інтелектуальне вимірювання, логістика, промисловий моніторинг, сільське господарство тощо.

Мережі LPWA унікальні тим, що вони створюють компромісні рішення, на відміну від традиційних технологій, що переважають у ландшафті IoT, таких як бездротові мережі короткого діапазону відстаней, наприклад, Zig-Bee, Bluetooth, Z-Wave, застарілі бездротові локальні мережі (WLAN), наприклад, Wi-Fi, і стільникові мережі: глобальна система мобільного зв'язку (GSM), система 4G (LTE – довгострокова еволюція) тощо. Старі неклітинні бездротові технології не є ідеальними для підключення пристроїв малої потужності, розподілених у великих географічних районах. Характеристики цих технологій обмежені у кращому випадку відстанню у кілька сотень метрів. Отже, пристрої не можна довільно розгортати або переміщувати куди завгодно, що є вимогою до багатьох програм для розумного міста, логістики та особистого здоров'я. Діапазон відстані цих технологій розширюється за допомогою щільного розгортання пристроїв і шлюзів, підключених за допомогою сіткових мереж. Розгортання на великій площі є надзвичайно дорогими. Наявні бездротові локальні мережі, з іншого боку, характеризуються більш вузькими зонами покриття та більшим споживанням електроенергії для зв'язку машинного типу (MTC).

Широке охоплення території забезпечують стільникові мережі, що є причиною широкого впровадження технологій другого покоління (2G)

та третього покоління (3G) для зв'язку МТС. Однак майбутнє виведення з експлуатації цих технологій, як оголосили деякі оператори мобільного зв'язку, розширить технологічний розрив у підключенні пристроїв малої потужності. Загалом, традиційні стільникові технології не досягають достатньої енергоефективності, щоб запропонувати десять років роботи від акумулятора. Складність і вартість стільникових пристроїв висока через їхню здатність працювати зі складними формами хвиль, оптимізованими під голос, швидкісними послугами передачі даних та текстом. Для малопотужного зв'язку M2M існує очевидна необхідність усунути складність, щоб зменшити витрати. Розробки в цьому напрямі проводяться для стільникових мереж у межах проєкту партнерства третього покоління та обговорюються в розділі **1.4.**

Рис. 1.1. Можливі застосування мереж LPWA

Завдяки феноменальному діапазону від кількох до десятків кілометрів [3] та ресурсу акумулятора десять років і більше технології LPWA є перспективними для Інтернету з малопотужними, недорогими та з малими пропускними можливостями. Велике різноманіття технологій LPWA дає змогу пристроям розміщуватися та переміщуватися по

великих географічних районах. Пристрої IoT та M2M, підключені за технологіями LPWA, можна вмикати в будь-якому місці та в будь-який час, щоб миттєво взаємодіяти зі своїм оточенням. Варто уточнити, що технології LPWA мають здатність роботи на великій відстані та малої потужності за рахунок низької швидкості передачі даних (зазвичай порядку десятків кілобітів на секунду) та більшої затримки (зазвичай порядку секунд або хвилин). Тому зрозуміло, що технології LPWA не призначені для вирішення кожного випадку використання IoT та обслуговують нішу в зоні IoT. Зокрема, технології LPWA розглядаються для тих випадків використання, які толерантні до затримки, не потребують високих швидкостей передачі даних і переважно потребують низького енергоспоживання та низької вартості, остання є важливим аспектом. Такі програми МТС класифікуються як Massive МТС на відміну від критичних програм МТС, які потребують наднизької затримки та надвисокої надійності. Останні виразно виходять зі сфери використання технологій LPWA, оскільки їх жорсткі вимоги до продуктивності, такі аж до п'яти знаків (99,999 %) надійності та затримки до 1÷10 мс, не можуть бути гарантовані рішенням із низькими витратами та низькою потужністю. Хоча технології LPWA з цієї причини не підходять для багатьох промислових застосувань IoT, зв'язку автомобілів та транспортних засобів (V2V) та транспортних засобів з інфраструктурою (V2I), вони все ще задовольняють потреби безлічі додатків для розумних міст, розумних міст вимірювання, домашньої автоматизації, електроніки, яка здійснюється людиною, логістики, моніторингу навколишнього середовища тощо (рис. 1.1), які обмінюються малою кількістю даних, до того ж нечасто. Тому привабливість технологій LPWA, хоча й обмежена її низькою швидкістю передачі даних, все ще широка. Саме тому технології LPWA викликали такий великий інтерес після того, як на ринок потрапили фірмові технології, наприклад SIGFOX та LORa.

На сьогодні існує кілька конкуруючих технологій LPWA, кожна з яких використовує різні методи для досягнення дальньої дальності, малої потужності та високої масштабованості. У розділі 1.2 представлені ці цілі дизайну та описано, як комбінація різних нових технологій насправді їх досягає. У розділі 1.3 обговорюються декілька ранніх фірмових технологій LPWA та їх технічні особливості, підкреслюється потреба у стандартизації для нашої екосистеми IoT. Для цього кілька відомих організацій які розвивають стандарти (SDO), серед яких Європейський інститут стандартів зв'язку (ETSI), Проєкт партнерства третього покоління (3GPP), Інститут інженерів електротехніки та електроніки (IEEE), і робоча група з Інтернет-інженерії (IETF) працюють над відкритими стандартами для технологій LPWA. Крім того, декілька промислових альянсів будуються

навколо окремих технологій LPWA для просування нових стандартів. LORa™ Alliance, WEIGHTLESS-SIG та DASH7 Alliance – кілька прикладів таких спеціальних груп інтересів (SIG). Розділ 1.4 охоплює зусилля зі стандартизації, які ведуть усі ці SDO та SIG.

З технічної сторони, постачальникам LPWA необхідно наполягати на інноваційних рішеннях для подолання труднощів підключення величезної кількості пристроїв IoT та M2M. Це дійсно непросте завдання, особливо коли різнорідні технології LPWA діляться обмеженими радіоресурсами, щоб зробити масштабованим та безпечним підключення до малопотужних та недорогих кінцевих пристроїв. Багаторазові компроміси, здійснені за допомогою технологій LPWA, викликають декілька проблем, які обговорюються у розділі 1.5, також там обговорюються можливі напрями досліджень щодо їх вирішення.

1.2. Цілі та техніки проєктування

Успіх технологій LPWA полягає у їх здатності пропонувати підключення з низькими витратами енергії до величезної кількості пристроїв, розподілених на великих географічних територіях за безпрецедентно низькою вартістю. У цьому розділі описуються технології LPWA, що використовуються для досягнення цих часто суперечливих цілей. Треба підкреслити, що технології LPWA поділяють деякі цілі дизайну з іншими бездротовими технологіями. Однак ключовою метою технологій LPWA є досягнення великого діапазону з низьким енергоспоживанням та низькою вартістю, на відміну від інших технологій, для яких досягнення вищої швидкості передачі даних, меншої затримки та більшої надійності може бути більш важливим.

А. Велика дальність

Технології LPWA розроблені для широкого покриття та відмінного поширення сигналу у важкодоступних приміщеннях, наприклад підвали. Передбачається збільшення сигналу +20 дБ над застарілими стільниковими системами. Це дає змогу кінцевим пристроям підключатися до базових станцій на відстані від кількох до десятків кілометрів залежно від середовища їх розгортання (сільського, міського тощо). Для досягнення цієї мети використовуються діапазони суб-ГГц і спеціальні схеми модуляції, які розглядатимуся.

1) Використання діапазону нижче 1ГГц: за винятком кількох технологій LPWA (наприклад, WEIGHTLESS-W та INGENU), більшість з них використовують діапазон Sub-GHz, який забезпечує надійну комунікацію при низьких рівнях енергетичного бюджету. По-перше, порівняно з діапазоном 2,4 ГГц, сигнали нижчої частоти відчувають менше послаблення та багатошарове згасання, викликане перешкодами та щільними

поверхнями, наприклад бетонні стіни. По-друге, суб-ГГц менш перевантажений, ніж 2,4 ГГц, діапазон, який використовується найпопулярнішими бездротовими технологіями, наприклад, Wi-Fi, бездротовими телефонами, Bluetooth, ZigBee та іншими побутовими приладами. Отримана більш висока надійність дозволяє забезпечити зв'язок на великій відстані та за низької потужності. Тим не менш, технологія RPMA INGENU є винятком, який досі використовує діапазон 2,4 ГГц завдяки більш ліберальному регулюванню спектру частот та більшій максимальній потужності передачі в цій смузі частот у різних регіонах.

2) Методи модуляції: технології LPWA розроблені для досягнення бюджету зв'язку 150 ± 10 дБ, що дає змогу охопити відповідно до кількох кілометрів і десятків кілометрів у містах та сільській місцевості. Апаратний рівень проєктується як компромісний відносно високої швидкості передачі даних і сповільнює швидкість модуляції, щоб вкласти більше енергії у кожен переданий біт (або символ). З цієї причини приймачі можуть правильно декодувати дуже ослаблені сигнали. Типова чутливість найсучасніших приймачів LPWA досягає –130 дБм. За різними технологіями LPWA були прийняті два класи методів модуляції, а саме методи вузькосмугового та розширеного спектру.

Методи вузькосмугової модуляції забезпечують високий бюджет зв'язку за допомогою кодування сигналу з низькою пропускною здатністю (переважно, менше 25 кГц). Присвоюючи кожному несучому сигналу дуже вузьку смугу, ці методи модуляції ефективно розподіляють загальний спектр між кількома ланками. Рівень шуму в одному вузькому діапазоні також мінімальний. Отже, для декодування сигналу на приймачі не потрібно посилювати обробку через розширення частоти, що призводить до простої та недорогої конструкції приймача. NB-IoT та WEIGHTLESS-P – приклади вузькосмугових технологій.

Кілька технологій LPWA видають кожен несучий сигнал у надвузькій смузі (UNB) шириною до 100 Гц (наприклад, у SIGFOX), що додатково зменшує відчутний шум і збільшує кількість підтримуваних кінцевих пристроїв на одиницю пропускної здатності. Однак ефективна швидкість передачі даних для окремих кінцевих пристроїв також зменшується, збільшуючи кількість часу, коли радіо потребує увімкнення. Ця низька швидкість передачі даних у поєднанні з регламентом спектру щодо спільного використання базових діапазонів може обмежувати максимальний розмір та частоту передачі пакетів даних, звужуючи кількість випадків використання бізнесу. SIGFOX, WEIGHTLESS-N та TELENSA – кілька прикладів технологій LPWA, які використовують модуляцію UNB.

Методи розширення спектру поширюють вузькосмуговий сигнал по більш широкій смузі частот, але з однаковою щільністю потужності. Фактична передача – це сигнал, подібний до шуму, який важче виявити підслуховувачем, більш стійкий до перешкод і надійний до атак глушення. Однак потрібна посилена обробка на стороні приймача для декодування сигналу, який зазвичай приймається нижче рівня шуму. Поширення вузькосмугового сигналу по широкій смузі призводить до менш ефективного використання спектру. Але ця проблема, зазвичай, долається за допомогою використання кількох ортогональних послідовностей. Поки кілька кінцевих пристроїв використовують різні канали та / або ортогональні послідовності, всі вони можуть декодуватися одночасно, що призводить до збільшення загальної ємності мережі. За різними варіантами методів розповсюдження спектру використовуються наявні стандарти, про які йдеться у розділах 1.3-B та розділі 1.3-C. Метод розширення за допомогою чирпу (CSS) та метод прямої послідовності розширення спектру (DSSS) використовуються LORa та RPMA відповідно.

B. Робота з наднизьким споживанням потужності

Робота з низьким енергоспоживанням – головна вимога, щоб скористатися величезною можливістю бізнесу, яку забезпечують пристрої IoT/M2M, що працюють від акумуляторів. Бажано для зменшення вартість технічного обслуговування батареї мати термін роботи – 10 та більше років з батареями типу AA або дисковими.

1. Топологія: хоча сітчаста топологія широко використовується для розширення покриття бездротових мереж близької відстані, їх висока вартість розгортання є головним недоліком підключення великої кількості географічно розподілених пристроїв. Крім того, коли трафік передається через декілька стрибків до шлюзу, деякі вузли отримують більше перевантаженості, ніж інші, залежно від їх розташування чи структури мережі. Тому вони швидко виснажують свої акумулятори, обмежуючи загальний термін експлуатації мережі лише кількома місяцями замість років.

З іншого боку, дуже довга відстань технологій LPWA долає ці обмеження, підключаючи кінцеві пристрої безпосередньо до базових станцій, уникаючи потреби у щільному та дорогому розгортанні ретрансляторів та шлюзів взагалі. Отримана топологія – це зірка, яка широко використовується в стільникових мережах і приносить величезні переваги енергозбереження. На відміну від сітчастої топології, пристрої не повинні витрачати дорогоцінну енергію на зайняте прослуховування інших пристроїв, які хочуть передати свій трафік через них. Базова станція, яка завжди знаходиться робочою, забезпечує зручний та швидкий доступ, коли цього вимагають кінцеві пристрої.

Окрім зірки, кілька технологій LPWA підтримують деревоподібні та сітчасті топології, але мають додаткову складність у розробці протоколів.

2) Обов'язкове циклічне використання: робота з низькою потужністю досягається опортуністичним відключенням компонентів пристроїв M2M/IoT, наприклад, передавача даних. Циклічна радіопередача дає змогу кінцевим пристроям LPWA вимикати їх трансивери, коли вони не потрібні. Тільки коли дані повинні бути передані або отримані, трансивер увімкнений.

Механізми робочого циклу LPWA адаптовані до застосування, типу джерела живлення та схеми руху транспорту серед інших факторів. Якщо програмі потрібно перенести дані лише по висхідній лінії зв'язку, кінцеві пристрої можуть прокинутися лише тоді, коли дані будуть готові до передачі. На відміну від цього, якщо потрібна також передача низхідній лінії зв'язку, кінцеві пристрої обов'язково слухають, коли базова станція насправді передає. Кінцеві пристрої досягають цього, узгоджуючи графік прослуховування. Наприклад, кінцеві пристрої можуть прослуховувати недовго після передачі висхідної лінії зв'язку, щоб отримати відповідь. Крім того, вони можуть прокинутися в запланований час, погоджений із базовою станцією. Для кінцевих пристроїв з основним живленням, які потребують наднизької затримки зв'язку низхідній лінії зв'язку, радіопередавач може залишатися у режимі, що постійно працює. Різні стандарти LPWA, такі, як LORAWAN, визначають декілька класів кінцевих пристроїв на основі їх потреб у зв'язку у висхідній та низхідній лінії зв'язку.

Уцарині технологій LPWA обов'язкова циклічна передача даних – це не лише механізм енергозбереження, а й законодавча вимога. Регіональні положення про спільне використання спектру можуть обмежувати час, який може займати один передавач, щоб забезпечити його співіснування з іншими пристроями, що діляться тим самим каналом.

Циклічна робота також може бути розширена за межі трансивера на інші апаратні компоненти, як це було досліджено в контексті багатьох вбудованих мереж малої потужності. Модульна конструкція обладнання може забезпечити можливість вибору різних режимів роботи та включення або вимкнення окремих апаратних компонентів (наприклад, допоміжних компонентів та накопичувачів і мікроконтролерів). Використовуючи ці методи управління енергією, розробники додатків LPWA можуть додатково скоротити споживання енергії та збільшити термін служби акумулятора.

3) Легкий контроль канального доступу: найпоширеніші протоколи управління канального доступу (MAC) для стільникових мереж або

бездротових мереж малої відстані занадто складні для технологій LPWA. Наприклад, стільникові мережі точно синхронізують базові станції та користувальницьке обладнання (UE), щоб отримати користь від складних схем MAC, що використовують різноманітність частот та часу. Контроль над цими схемами, хоча й виправданий для потужних стільникових UE, є значним для кінцевих пристроїв LPWA. Інакше кажучи, управління цими протоколами MAC може бути навіть дорожчим, ніж коротка та нечаста комунікаційна система пристроїв LPWA. Крім того, дуже щільну синхронізацію, необхідну для цих схем, складно задовольнити наднизькими (1–5 доларів США) кінцевими пристроями, що мають низькі якості дешевих тактових генераторів. Під час доступу до спектру ці пристрої відчувають зміну як у часовій, так і в частотній областях, що робить ексклюзивний доступ до спільного середовища основним завданням для конкуруючих пристроїв. З цієї причини прості схеми випадкового доступу є більш популярними для технологій LPWA.

Багаторазовий доступ із контролем несучої та уникненням зіткнень (CSMA/CA) є одним з найпопулярніших протоколів MAC, що успішно розгортається в WLAN та інших бездротових мережах короткого діапазону. Кількість пристроїв на базовій станції обмежена для таких мереж, що підтримує приховану проблему вузла. Однак у міру зростання кількості цих пристроїв у мережах LPWA зондування оператора стає менш ефективним і дорогим для надійного виявлення поточних передач, що негативно впливає на продуктивність мережі. Коли віртуальне зондування оператора за допомогою механізму Запити на відправку / очищення для відправки (RTS/CTS) використовується для подолання цієї проблеми, воно вводить додаткові накладні комунікації по висхідній і низхідній лінії зв'язку. Завдяки величезній кількості пристроїв, технології LPWA зазвичай не можуть дозволити собі надмірні сигнали. Крім того, асиметрія зв'язку, властивість багатьох технологій LPWA сьогодні, знижує практичність віртуального зондування.

Через ці причини багато технологій LPWA, такі, як SIGFOX та LORAWAN, вдаються до використання ALOHA[1], протоколу MAC з випадковим доступом, в якому кінцеві пристрої передають без будь-якого зондування. Як вважається, простота ALOHA підтримує дизайн приймача простим і з низькою вартістю. Проте протоколи MAC на базі TDMA також розглядаються INGENU та NB-IoT для більш ефективного розподілу радіоресурсів, хоча через більшу складність та витрати на кінцеві пристрої.

[1] ALOHA – протокол попередження колізій при передачі даних у мережі, з гавайської мови – «Привіт» та вираження гарного ставлення

4) Велика складність від кінцевих пристроїв: більшість технологій спрощують конструкцію кінцевих пристроїв шляхом передачі складних завдань базовим станціям або резервній системі. Щоб зберегти конструкцію приймача для кінцевих пристроїв простою і з низькою вартістю, базові станції або резервна система повинні бути складнішими. Зазвичай, базові станції використовують різноманітність апаратних засобів і здатні передавати та прослуховувати з кількох кінцевих пристроїв, використовуючи одночасно кілька каналів або ортогональних сигналів. Це дозволяє кінцевим пристроям надсилати дані, використовуючи будь-який доступний канал або ортогональний сигнал, та дістатися до базової станції без необхідності дорогої сигналізації для ініціювання зв'язку. Вбудовуючи певний інтелект у систему бекенда, кінцеві пристрої можуть отримати додаткові переваги від надійнішого та енергоефективнышого зв'язку останньої милі. Помітним прикладом є LORAWAN, в якому система доповнення адаптує параметри зв'язку (наприклад, параметри швидкості передачі даних / модуляції) для підтримки хороших з'єднань висхідної та низхідній ліній зв'язку. Крім того, система бекенда також відповідає за надання підтримки кінцевим пристроям для переміщення по кількох базових станціях і придушення повторюваних прийомів, якщо такі є. Вибір збереження складності на базових станціях та резервних системах, які мають меншу кількість, забезпечує низьку вартість і конструкції з низькою потужністю споживання для багатьох кінцевих пристроїв.

Окрім комунікацій, обробка даних також може бути відокремлена від кінцевих пристроїв, але нам потрібно зрозуміти кілька компромісів. Зважаючи на різноманітність програм IoT, кожен може мати різні вимоги, зокрема частоту звітування даних. Можливо, деякі програми вимагають від кінцевих пристроїв часто повідомляти дані (наприклад, раз на кілька хвилин). З іншого боку, у нас можуть бути програми, які вимагають від кінцевих пристроїв рідше повідомляти дані, можливо, раз на день. З погляду споживання енергії загальновідомий факт, що операція зв'язку споживає більше енергії, ніж операція обробки. Тому ключовим питанням, яке часто виникає, є те, чи повідомляти про всі дані такими, які вони є, або провести якусь локальну обробку та повідомити про оброблений результат (зменшена потреба у спілкуванні). Колишній підхід не потребує значної можливості обробки кінцевого пристрою, що означає, що можуть бути реалізовані пристрої з низькою вартістю. Однак в останньому випадку, залежно від складності необхідної обробки, вартість кінцевого пристрою, ймовірно, зросте, хоча і зменшить споживання енергії, необхідноъ для транспортування даних. Вибір між цими двома дійсно зумовлений базовим бізнесом. Хоча завжди бажано мати недорогі кінцеві пристрої, особливо зважаючи на великі обсяги пристроїв, може бути

корисним місцева обробка, якщо вартість зв'язку значна. Так само, якщо вартість зв'язку не залежить від обсягу даних (через ціноутворення по фіксованій ставці), то вигідніше мати більш прості кінцеві пристрої. Необхідно також оцінити витрати, пов'язані з експлуатацією кінцевого пристрою з/і без складної обробки. Інакше кажучи, як збільшується вартість, якщо часто замінювати кінцевий пристрій через виснаження акумулятора, спричинене більшою кількістю сеансів зв'язку проти використання трохи дорожчого кінцевого пристрою в першому місці, яке спілкується рідше, але не вичерпує акумулятор так часто. З позиції оператора мережі, можливо, буде бажано зменшити кількість трафіку в їхній мережі шляхом локальної обробки на вузлах, оскільки це може зменшити ймовірність виникнення проблем із продуктивністю. Однак це може бути небажаним, якщо бізнес-модель оператора покладається на ціноутворення, що не ґрунтується на обсязі даних.

Парадигма обробки даних поблизу від кінцевого пристрою, яку останнім часом називають крайовими обчисленнями, набуває все більшої популярності, як це очевидно з посилення таких ініціатив, як OpenFog та Mobile Edge Computing. Однак, не існує простої однобічної бінарної відповіді на проблему, чи потрібно транспортувати необроблені дані або транспортувати локально оброблений результат. Як вже згадувалося раніше, це дійсно зводиться до вимог програми та аналізу рентабельності інвестицій (ROI) для тих, хто хоче розгорнути такі рішення.

C. Низька вартість

Комерційний успіх мереж LPWA пов'язаний із підключенням великої кількості кінцевих пристроїв, водночас зберігаючи вартість апаратного забезпечення нижче 5 доларів, а підписка на підключення на одиницю становить лише 1 долар. Ця доступність дозволяє технологіям LPWA не лише вирішувати широкий спектр застосувань, але й вигідно конкурувати в тих галузях, де бездротові технології та стільникові мережі вже налагоджені. Технології LPWA використовують декілька способів зменшення капітальних витрат (CAPEX) та експлуатаційних витрат (OPEX) як для кінцевих споживачів, так і для мережевих операторів. Конструкція кінцевих пристроїв із низькою вартістю стає можливою за допомогою декількох прийомів, деякі з яких розглядались вище. Використання підключення типу «зірка» (а не сітчастого), простих протоколів MAC та прийомів для зменшення складності кінцевих пристроїв дає змогу виробникам проєктувати прості і недорогі кінцеві пристрої. Ще деякі методи, механізми та підходи обговорюються так:

1) Зменшення складності обладнання: порівняно із бездротовими технологіями стільникового та близького діапазонів, приймачі LPWA повинні обробляти менш складні форми хвиль. Це дозволяє їм знижувати

розмір трансивера, максимальну швидкість передачі даних та розмір пам'яті, мінімізуючи складність обладнання та, отже, вартість [3]. Виробники чіпів LPWA націлені на велику кількість підключених кінцевих пристроїв, а також можуть знизити витрати з економією залежно від масштабу.

2) Мінімальна інфраструктура: традиційні бездротові та дротові технології страждають від обмеження відстані, що вимагає щільного і, отже, дорогого розгортання інфраструктури (шлюзи, лінії електропередач, релейні вузли тощо). Однак одна базова станція LPWA з'єднує десятки тисяч кінцевих пристроїв, розподілених на кілька кілометрів, що значно скорочує витрати операторів мережі.

3) Використання смуг частот, на які не потрібна ліцензія, або право власності: вартість операторів мережі на отримання ліцензій на новий спектр для технологій LPWA пов'язана з низькою вартістю розгортання, коротким часом виходу на ринок та конкурентоспроможністю їхніх передплатних пропозицій для клієнтів. Тому більшість технологій LPWA розглядають використання в смугах, що не належать до ліцензійних, включно з промисловим, науковим та медичним (ISM) діапазоном або телевізійним простором. NB-IoT, стандарт LPWA від 3GPP, може ділитися стільниковими смугами, які вже належать МНО, щоб уникнути додаткових витрат на ліцензування. Однак для досягнення кращої продуктивності можна придбати й окремий ліцензований діапазон, з часом може з'явитися тенденція захищених технологій LPWA, щоб уникнути погіршення продуктивності через збільшення кількості підключених пристроїв, які використовують спільний спектр.

D. Масштабованість

Підтримка значної кількості пристроїв, що надсилають невеликі обсяги трафіку, є однією із ключових вимог для технологій LPWA. Ці технології повинні добре працювати зі збільшенням кількості та щільності підключених пристроїв. Для вирішення цієї проблеми масштабованості розглядається кілька методик.

1) Різноманітні методи: для розміщення якомога більшої кількості підключених пристроїв життєво важливою є ефективна експлуатація різноманітностей у каналах, часі, просторі та апаратурі. Через низьку потужність і недорогий характер кінцевих пристроїв багато чого досягається завдяки співпраці більш потужних компонентів у мережах LPWA, таких як базові станції та резервні системи. Технології LPWA використовують багатоканальний та багатоантенний зв'язок для паралелізації передач на підключені пристрої та від них. Крім того, зв'язок стає стійким до перешкод, використовуючи кілька каналів і виконуючи надмірні передачі.

2) Ущільнення: щоб впоратися зі збільшенням щільності кінцевих пристроїв у певних районах, мережі LPWA, як і традиційні стільникові мережі, вдаватимуться до щільних розгортань базових станцій. Однак проблема полягає в тому, щоб зробити це, не викликаючи занадто сильних перешкод між кінцевими пристроями та щільно розгорнутими базовими станціями. Нові підходи до ущільнення для мереж LPWA потребують подальшого дослідження, оскільки наявні стільникові методи покладаються на добре узгоджене управління радіоресурсами всередині і між комірками, припущення не відповідає дійсності у більшості технологій LPWA.

3) Вибір адаптивного каналу та швидкість передачі даних: не тільки системи LPWA повинні масштабуватися до кількості підключених пристроїв, але окремі ланки мають бути оптимізовані для надійного та енергоефективного зв'язку. Адаптація схем модуляції, вибір кращих каналів для надійного досягнення відстаней або здійснення адаптивного управління потужністю передачі вимагають ефективного моніторингу якості зв'язку та координації між кінцевими пристроями та мережею.

Ступінь можливого вибору та модуляції адаптивного каналу залежить від основної технології LPWA. Різні фактори, такі як асиметрія зв'язку та максимально допустимий радіозахисний цикл, можуть обмежувати можливість для дуже надійних адаптивних механізмів. У випадках, коли базова станція не може надати зворотний зв'язок щодо якості зв'язку висхідної лінії зв'язку та / або повідомити кінцеві пристрої для адаптації їх параметрів, кінцеві пристрої вдаються до дуже спрощеного механізму для покращення якості зв'язку. Такий механізм містить передачу одного і того ж пакета кілька разів по кількох випадково обраних каналах з надією, що принаймні одна копія успішно досягне базової станції. Такі механізми, ймовірно, підвищують надійність цього зусилля для висхідної лінії зв'язку, зберігаючи водночас складність та вартість кінцевих пристроїв дуже низькими. У випадках, коли зв'язок по низхідній лінії зв'язку може забезпечити адаптацію параметрів висхідної лінії зв'язку, базові станції або резервні системи можуть відігравати важливу роль у виборі таких оптимальних параметрів, як канал або оптимальна швидкість передачі даних для підвищення надійності та енергоефективності.

Підбиваючи підсумок, існує чіткий компроміс між масштабованістю мережі та простотою пристроїв з низькою ціною. Більшість технологій LPWA дають змогу кінцевим пристроям малої потужності отримати доступ до обмежених радіоресурсів здебільшого неузгоджено та випадково, обмежуючи кількість пристроїв, які можуть підтримуватися мережами. Збільшення нещодавно опублікованих досліджень [3] виявляє

практичні обмеження щодо масштабованості мереж LPWA. У розділі 1.5 це розглядається як цікавий напрям майбутніх досліджень.

Е. Якість обслуговування

Технології LPWA орієнтовані на різноманітний набір програм із різними вимогами. Щонайменше він забезпечує затримку толерантних програм розумного вимірювання, водночас, він повинен доставляти тривожний сигнал, створений домашніми програмами безпеки за мінімальний час. Тому мережа повинна забезпечувати якість обслуговування (QoS) за тією ж базовою технологією LPWA. Для клітинних стандартів, де базові радіоресурси можуть бути розподілені між програмами LPWA та мобільними широкосмуговими програмами, необхідно визначити механізми співіснування різних типів трафіку. Наскільки нам відомо, сучасні технології LPWA не забезпечують або обмежують якість QoS.

1.3. Фірмові технології

У цьому розділі висвітлюються та порівнюються нові фірмові технології, показані на рис. 1.2, та їх технічні аспекти. Деякі з цих технологій відповідають вимогам стандартів, запропонованих різними SDO та SIG. У розділі 1.4 коротко описані ці стандарти та їх зв'язок з будь-якими фірмовими технологіями, які розглядатимутися далі.

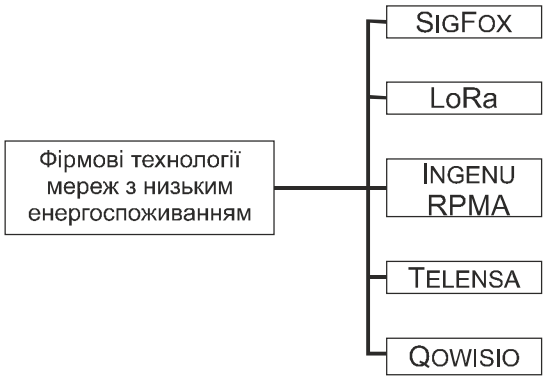

Рис. 1.2. Види технологій LPWA

А. SIGFOX

Сам SIGFOX або у партнерстві з іншими операторами мережі пропонує комплексне рішення для підключення LPWA на основі своїх запатентованих технологій. Мережеві оператори SIGFOX (SNO) розгортають власні базові станції, оснащені когнітивними програмними радіостанціями, і підключають їх до сервера, що працює за допомогою резервного

сервера, використовуючи мережу на базі IP. Кінцеві пристрої підключаються до цих базових станцій за допомогою двійкової фазової модуляції (BPSK) в ультравузькій смузі несучої SUB-GHZ ISM діапазону (100 Гц). Використовуючи UNB, SIGFOX ефективно застосовує пропускну здатність і має дуже низький рівень шуму, що призводить до високої чутливості приймача, наднизького споживання енергії та недорогих конструкцій антен. Усі ці переваги надходять завдяки максимальній пропускній здатності лише 100 біт/с. Досягнута швидкість передачі даних явно падає на нижній кінець пропускної здатності, запропонованої більшістю інших технологій LPWA, і так обмежує кількість випадків використання для SIGFOX. Крім того, SIGFOX спочатку підтримував лише зв'язок висхідної лінії зв'язку, але пізніше перетворився на двосторонню технологію, хоча зі значною асиметрією зв'язку.

Зв'язок низхідної лінії зв'язку може передувати лише комунікаційному каналу висхідної лінії зв'язку, після чого кінцевому пристрою необхідно чекати, щоб прослухати відповідь із базової станції. Кількість та розмір повідомлень по висхідній лінії зв'язку обмежуються 140-ка 12-байтними повідомленнями на день, щоб відповідати регіональним нормам щодо використання безліцензійного спектру. Посилання для доступу до радіо несиметричне, що дає змогу передавати максимум лише 4 8-байтних повідомлень на день по низхідній лінії зв'язку від базових станцій до кінцевих пристроїв. Це означає, що підтвердження кожного повідомлення висхідної лінії зв'язку не підтримується.

Без належної підтримки для підтверджень надійність зв'язку висхідної лінії зв'язку покращується за допомогою різноманітності часу та частоти, а також надмірних передач. Одне повідомлення з кінцевого пристрою може передаватися кілька разів по різних каналах частоти. З цією метою в Європі смуга між 868.180-868.220 МГц ділиться на 400 каналів 100 ГГц, з яких 40 каналів зарезервовані і не використовуються. Оскільки базові станції можуть сканувати всі канали для декодування повідомлень, кінцеві пристрої можуть автономно вибирати випадковий частотний канал для передачі своїх повідомлень. Це спрощує конструкцію для кінцевих пристроїв. Крім того, одне повідомлення передається кілька разів (за замовчуванням 3), щоб збільшити ймовірність успішного прийому базовими станціями.

Б. LOR*a*

LORa – це технологічний рівень фізичного рівня, який модулює сигнали в діапазоні ISM SUB-GHZ, використовуючи власну техніку розширення спектру, розроблену та комерціалізовану корпорацією Semtech [4]. Двонаправлений зв'язок забезпечується спеціальною технікою чирппоширення (CSS) (використовуються сигнали з частотною модуляцією

всередині імпульсу), яка розширює вхідний сигнал вузької смуги по більш широкій смузі каналів. Отриманий сигнал має властивості, подібні до шуму, що ускладнює їх виявлення або глушіння.

Передавач генерує чирп (the chirp) сигнали, що змінюють свою частоту з часом, не змінюючи фази між сусідніми символами. Поки ця зміна частоти відбувається досить повільно, щоб поставити більшу енергію на символ щебетання (чирпу), віддалені приймачі можуть декодувати дуже ослаблений сигнал на кілька дБ нижче рівня шумового сигналу. LORa підтримує множинні коефіцієнти розширення (між 7–12), щоб вирішити компроміс між діапазоном і швидкістю передачі даних. Більш високі коефіцієнти поширення забезпечують велику дальність завдяки меншої швидкості передачі даних і навпаки. LORa також поєднує попередню корекцію помилок (FEC) з технікою розширення спектру для подальшого підвищення чутливості приймача. Швидкість передачі даних коливається від 300 bps до 37,5 kbps залежно від коефіцієнта поширення та пропускної здатності каналу. Крім того, базова станція LORa може приймати одночасно кілька передач з використанням різних коефіцієнтів розширення. По суті, множинні фактори, що розширюють спектр, забезпечують третій ступінь різноманітності за часом та частотою.

Кілька досліджень оцінювали LORAWAN у реальних умовах, як назовні так і в приміщенні. Були оцінені LORa та SIGFOX завдяки експериментам, проведеним із тестового розгортання в Ірландії. Результати показують, що базова станція LORa, розміщена на 470 м над рівнем моря, могла б обслуговувати зону покриття 1 380 квадратних кілометрів у тестовій установці і що технологія SIGFOX змогла забезпечити 25-кілометровий тестовий зв'язок між клієнтом, що використовує сигнал 14 дБм, і базовою станцією за співвідношення сигнал / шум, що стабільно перевищує 20 дБ. В іншому дослідженні спостерігалася дальність зв'язку 15 км та 30 км для LORAWAN на землі та воді відповідно у Фінляндії. Крім того, в іншому дослідженні, проведеному в університеті, кінцеві пристрої передавали при 14 дБм, використовуючи найвищий коефіцієнт розширення (12) до базової станції, яка знаходилася в радіусі 420 м. Зафіксовано коефіцієнт доставки пакетів на базовій станції 96,7 %.

Повідомлення, що передаються кінцевими пристроями, приймаються не одиницею, а всіма базовими станціями в діапазоні, що відповідає топології «зірки зірок». Використовуючи у такий спосіб різноманітність прийому, LORa покращує співвідношення успішно отриманих повідомлень. Однак для досягнення цього потрібні кілька базових станцій у мікрорайоні, які можуть збільшити CAPEX та OPEX. Отримані повторювані прийоми фіксуються в системі резервного копіювання. Крім того, LORa використовує ці багаторазові прийоми одного і того ж

повідомлення на різних базових станціях для локалізації передавального кінцевого пристрою. Для цього використовується техніка локалізації на основі різниці в часі прибуття (TDOA), підтримувана дуже точною синхронізацією часу між кількома базовими станціями.

Спеціальна група за інтересами, що складається з кількох комерційних та промислових партнерів, названих Альянсом LORa™, запропонувала LORAWAN, відкритий стандарт, що визначає архітектуру та шари над фізичним шаром LORa. Ми коротко описуємо LORAWAN відповідно до стандартів у розділі 1.4.

C. INGENU RPMA

INGENU (раніше відомий як On-Ramp Wireless) запропонував фірмову технологію LPWA, яка, на відміну від більшості інших технологій, не покладається на кращі властивості поширення смуги SUB-GHZ. Натомість він працює в діапазоні ISM 2,4 ГГц і використовує більш розслаблені правила щодо використання спектру в різних регіонах. На приклад, норми в США та Європі не встановлюють максимального обмеження робочого циклу для діапазону 2,4 ГГц, що забезпечує більш високу пропускну здатність та більшу потужність, ніж інші технології, що працюють у діапазоні SUB-GHZ.

Найголовніше, що INGENU використовує запатентовану схему фізичного доступу, іменовану як випадковий множинний доступ (RPMA) Direct Sequence Spread Spectrum, яку вона використовує лише для зв'язку в висхідній лінії зв'язку. Як сама версія множинного доступу з кодовим поділом (CDMA), RPMA дає можливість декільком передавачам ділитися одним часовим слотом. Однак, RPMA насамперед збільшує тривалість інтервалу часу традиційного CDMA, а потім розсіює доступ до каналу в цьому слоті, додаючи затримку випадкового зміщення для кожного передавача. Не надаючи каналу доступ до передавачів точно одразу (тобто на початку прорізу), RPMA зменшує перекриття між переданими сигналами і так збільшує відношення сигналу до перешкод для кожної окремої ланки [18]. На приймальній стороні базові станції використовують кілька демодуляторів для декодування сигналів, що надходять у різний час у проріз. INGENU забезпечує двосторонній зв'язок, хоча і з невеликою асиметрією зв'язку. Для зв'язку по низхідній лінії базові станції розширюють сигнали по спектру для окремих кінцевих пристроїв і потім передають їх за допомогою CDMA.

Повідомляється, що RPMA досягає чутливості приймача до –142 дБм та бюджету зв'язку 168 дБ. Крім того, кінцеві пристрої можуть регулювати свою потужність передачі для досягнення найближчої базової станції та обмеження перешкод для сусідніх пристроїв.

INGENU докладає зусиль щодо стандартизації специфікацій фізичного рівня відповідно до стандарту IEEE 802.15.4k. Технологія RPMA відповідає стандартам IEEE 802.15.4k.

D. TELENSA

TELENSA пропонує комплексні рішення для програм LPWA, що містять повністю розроблені вертикальні мережеві стеки з підтримкою інтеграції з програмним забезпеченням сторонніх виробників.

Для бездротового зв'язку між їх кінцевими пристроями та базовими станціями TELENSA розробила фірмову техніку модуляції UNB, яка працює в безліцензійному діапазоні ISM SUB-GHZ ISM з низькою швидкістю передачі даних. Хоча менше відомо про впровадження їх бездротових технологій, TELENSA прагне стандартизувати свою технологію, використовуючи специфікації ETSI з низькою пропускною здатністю (LTN) для легкої інтеграції в додатки.

Наразі TELENSA зосереджується на таких розумних містах, як інтелектуальне освітлення, розумне паркування тощо. Для посилення своїх пропозицій LPWA в інтелектуальному освітленні TELENSA бере участь у консорціумі TALQ, визначаючи стандарти моніторингу та управління системами зовнішнього освітлення.

E. QOWISIO

QOWISIO використовує двомодні мережі LPWA, поєднуючи власну технологію UNB та LORa. Вони забезпечують підключення LPWA як послугу для кінцевих користувачів: пропонуються кінцеві пристрої, розгортання мережевої інфраструктури, розробку власних додатків та розміщення їх у хмарі. Однак менше відомо про технічні характеристики їх основної технології UNB та інших компонентів системи.

1.4. Стандарти

Різноманітні шаблони стандартизації проводяться різними установленими органами стандартизації, включно з Інститутом інженерів з електротехніки та електроніки (IEEE), Європейським інститутом стандартів зв'язку (ETSI) та Проєктом партнерства третього покоління (3GPP), а також промисловими консорціумами, серед яких WEIGHTLESS-SIG, Альянс LORa™ і DASH7 Alliance. На рис. 1.3 впорядковані запропоновані стандарти відповідно до організацій, що їх розробляють. Більшість цих зусиль також стосується декількох фірмових постачальників зв'язку LPWA, про які йшлося у попередньому розділі. Цілі цих SDO та SIG досить різноманітні. У перспективі можна сподіватися, що прийняття цих стандартів, ймовірно, зменшить фрагментацію ринку LPWA та дасть змогу співіснувати кільком конкуруючим технологіям.

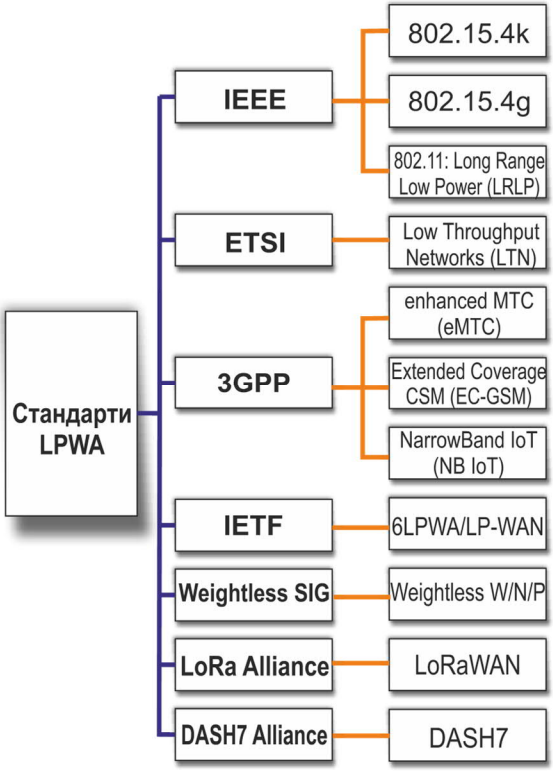

Рис. 1.3. LPWA стандарти та їх організації-розробники

A. IEEE

IEEE розширює діапазон і зменшує енергоспоживання своїх стандартів 802.15.4 та 802.11 з набором нових специфікацій для фізичного шару та шару MAC. Два стандарти LPWA пропонуються як поправки до базового стандарту IEEE 802.15.4 для низькошвидкісних бездротових персональних мереж (LR-WPAN), які розглянемо в цьому розділі. Також описані зусилля щодо внесення змін до стандарту IEEE 802.11 для бездротових локальних мереж (WLAN) для більшої дальності.

1) IEEE 802.15.4k: Мережі моніторингу критичної інфраструктури з низьким рівнем енергії: IEEE 802.15.4k цільова група (TG4k) пропонує стандарт для цілей моніторингу критичної інфраструктури з низькою енергією для роботи в діапазонах ISM (SUB-GHZ і 2,4 ГГц). Це було відповіддю на той факт, що у попередньому стандарті не вистачає діапазону та щільності вузлів, необхідних для програм LPWA. Поправка IEEE

802.15.4k усуває цей проміжок, прийнявши DSSS та FSK як два нових шари PHY. Можна використовувати кілька дискретних смуг пропускання каналів у межах від 100 кГц до 1 МГц. Специфікації рівня MAC також вносяться до змін для нових фізичних шарів. Стандарт підтримує звичайний CSMA/CA без пріоритетного каналу доступу (PCA), CSMA та ALOHA з PCA. За допомогою PCA пристрої та базові станції можуть надавати пріоритет їхній роботі в доступі до середовища, надаючи поняття якості обслуговування. Як і більшість стандартів LPWA, кінцеві пристрої підключені до базових станцій зірковою топологією і здатні обмінюватися асинхронними та запланованими повідомленнями.

Описано розгортання LPWA на базі IEEE 802.15.4k для моніторингу якості повітря. Була розгорнута зоряна топологічна мережа, де 1 точка доступу та 5 вузлів були розгорнуті в радіусі 3 км від центру університетського містечка. Точка доступу працює в смузі 433 МГц. Використовуючи потужність передачі 15 дБм, приймач може підтримувати різну чутливість залежно від вимог швидкості передачі даних, наприклад, чутливість до –129 дБм, –123 дБм і –110 дБм може бути досягнута для швидкостей передачі даних, що відповідають відповідно 300 біт/с, 1,2 кбіт/с і 50 кбіт/с відповідно.

INGENU, постачальник технології RPMA LPWA, є прихильником цього стандарту. Шари PHY та MAC технології LPWA INGENU відповідають цьому стандарту.

2) IEEE 802.15.4g: З низькою швидкістю передачі даних, бездротові, інтелектуальні мережі вимірювання: цільова група IEEE 802.15 WPAN 4g (TG4g) пропонує перший набір поправок для фізичного рівня (PHY) для розширення портфеля базових стандартів IEEE 802.15.4 для короткого діапазону. Випуск стандарту у квітні 2012 року стосується програм управління процесом, серед яких мережі інтелектуального обліку, які за своєю суттю складаються з величезної кількості налагоджених кінцевих пристроїв, розгорнутих у містах чи країнах. Стандарт визначає три шари PHY, а саме FSK, ортогональний частотний поділ із множинним доступом (OFDMA) та квадратурний фазовий зсув (QPSK), які підтримують різні швидкості передачі даних від 40 кбіт до 1 Мбіт/с у різних регіонах. За винятком одного ліцензованого діапазону в США, PHY переважно працює в діапазонах ISM (SUB-GHZ та 2,4 ГГц) і, отже, співіснує з іншими перешкоджаючими технологіями в тому ж діапазоні. PHY розроблений для доставки кадрів розміром до 1500 байт, щоб уникнути фрагментації пакетів Internet Protocol (IP).

Зміни рівня MAC для підтримки нових PHY визначаються IEEE 802.15.4e, а не самим стандартом IEEE 802.15.4g.

3) IEEE 802.11: Бездротові локальні мережі: технології WLAN відіграватимуть важливу роль в IoT. Зусилля щодо розширення дальності та зменшення енергоспоживання для WLAN докладаються IEEE 802.11 цільовою групою AH (TGah) та IEEE 802.11.

TGah запропонував специфікації IEEE 802.11ah для PHY та MAC для роботи Wi-Fi на великій відстані в діапазоні ISM SUB-GHZ. Порівняно зі стандартом IEEE 802.11ac, було впроваджено кілька нових функцій для досягнення 1 км дальності у зовнішніх умовах та швидкості передачі даних понад 100 кбіт/с. PHY приймає OFDM, які передають зі швидкістю в 10 разів повільніше, ніж IEEE 802.11ac, більш ранній стандарт, щоб розширити діапазон зв'язку. На шарі MAC накладні витрати, пов'язані з кадрами, заголовками та маяками, зменшуються для продовження роботи від батареї. Протокол MAC пристосований до тисяч (8191) підключених кінцевих пристроїв, щоб зменшити виникаючі між ними зіткнення. Кінцеві пристрої оснащені режимом для економії енергії протягом неактивних періодів, але зберігають свій зв'язок / синхронізацію з точками доступу. Завдяки всім цим механізмам енергозбереження та розширення діапазону, IEEE 802.11ah дійсно забезпечує значно більший діапазон і нижче споживання енергії, ніж інші стандарти WLAN, ZigBee та Bluetooth, але не стільки, скільки інші технології LPWA, обговорені в цій роботі. З цієї причини, все більша кількість нещодавно опублікованих досліджень та проєктів документів IETF не відносять IEEE 802.11ah як технологію LPWA. Насправді IEEE 802.11ah задовольняє ті програми, які потребують відносно більшої пропускної здатності через більше енергоспоживання, ніж інші технології LPWA.

Доцільність використання IEEE 802.11ah для випадків використання IoT/2M вивчена в [5]. Автори показують, що при використанні діапазону 900 МГц, для випадку низхідній лінії, просто досягти дальності в 1 км і вище 100 кбіт/с, швидкість передачі даних, оскільки AP використовує більшу потужність передачі (20-30 дБм). Однак у випадку висхідної лінії зв'язку досягти цих цілей досить складно, оскільки клієнти працюють з низьким енергоспоживанням (0 дБм) і повинні спрацьовувати циклично, щоб забезпечити багато років роботи акумулятора. У такому разі автори підкреслювали, що отримати діапазон до 400 м можливо при використанні схем кодування, вищої потужності передачі та антен з більшим посиленням. Однак це може статися через скорочення часу автономної роботи у клієнтів, що може виявитися небажаним. Вони також припускають, що якщо вимоги до надійності будуть зменшені, діапазон може бути додатково збільшений, наприклад, їм вдалося досягти 1 км дальності для надійності зв'язку менше 60 %.

У 2016 році в межах 802.11 було створено нову тематичну групу (TIG), щоб вивчити доцільність нового стандарту для малої дальності малої потужності (LRLP). На ранній стадії цієї роботи TIG визначив деякі випадки використання та функціональні вимоги до цієї технології, але не міг чітко обґрунтувати потребу в цій діяльності в Комітеті стандартів IEEE LAN/MAN (LMSC). Тому робота над LRLP закінчилася передчасно.

B. ETSI

ETSI спряморвує зусилля на стандартизацію двонаправленого стандарту LPWA з низькою швидкістю передачі даних. Отриманий стандарт, який отримав назву «Малопропускна мережа» (LTN), був випущений у 2014 році у вигляді трьох специфікацій групи. Ці характеристики визначають а) випадки використання; б) функціональну архітектуру; в) протоколи та інтерфейси. Однією з його головних цілей є зменшення електромагнітного випромінювання через використання коротких розмірів корисного навантаження та низьких швидкостей передачі даних зв'язку M2M/IoT.

Окрім рекомендацій щодо бездротових інтерфейсів, LTN визначає різні інтерфейси та протоколи для співпраці між кінцевими пристроями, базовими станціями, мережевим сервером, операційними та бізнес-системами управління.

Мотивований тим, що нові LPWA-мережі використовують як ультравузьку смугу (наприклад, SIGFOX, TELENSA), так і методи розширення спектру ортогональної послідовності (OSSS) (наприклад, LORa) модуляції, стандарт LTN не обмежується однією категорією. Це забезпечує гнучку можливість операторам LPWA розробити та розгорнути власні фірмові схеми модуляції UNB або OSSS в діапазоні ISM SUB-GHZ до того часу, поки кінцеві пристрої, базові станції та мережеві сервери реалізують інтерфейси, описані специфікаціями LTN. Ці характеристики рекомендують використовувати BPSK у висхідній лінії зв'язку та GFSK у низхідній лінії зв'язку для реалізації UNB. Альтернативно, будь-яка схема модуляції OSSS може використовуватися для підтримки двосторонньої комунікації. Шифрування даних, а також процедури аутентифікації користувачів визначаються як частина специфікацій LTN.

Кілька таких постачальників технологій LPWA, як SIGFOX, TELENSA та Semtech, активно беруть участь у ETSI для стандартизації своїх технологій.

C. 3GPP

Щоб вирішити питання щодо ринку M2M та IoT, 3GPP розробляє наявні стандарти стільникового зв'язку, щоб позбавити складності та вартості, покращити діапазон і проникнення сигналу, а також продовжити термін служби акумулятора. Багаторазові ліцензовані рішення, серед

яких вдосконалення довгострокової еволюції (LTE – 4G) для комунікацій типу машин (eMTC), розширене покриття GSM (EC-GSM) та вузькосмуговий IoT (NB-IoT), пропонують різні вигоди між вартістю, покриттям, швидкістю передачі даних та енергоспоживання для задоволення різноманітних потреб додатків IoT та M2M. Однак загальною метою всіх цих стандартів є максимальне використання наявної стільникової інфраструктури та власного радіочастотного спектру.

1) Удосконалення LTE для комунікацій між машинами (eMTC): Звичайні пристрої LTE-кінцевих пристроїв пропонують послуги з високою швидкістю передачі даних за вартістю та енергоспоживанням, неприйнятними для деяких випадків використання MTC. Щоб знизити витрати, дотримуючись системних вимог LTE, 3GPP знижує пікову швидкість передачі даних з категорії LTE 1 до LTE категорії 0, а потім до LTE категорії M, різних стадій процесу еволюції LTE. Подальше зниження витрат досягається завдяки підтримці опціонального напівдуплексного режиму у категорії 0. Цей вибір зменшує складність конструкції модему та антени. Від категорії 0 до категорії M1 (також відомий як eMTC), більш виражене падіння пропускної здатності прийому з 20 МГц до 1,4 МГц у поєднанні зі зниженою потужністю передачі приведе до конструкції більш економічної та з меншою потужністю.

Щоб продовжити термін служби акумулятора для eMTC, 3GPP використовує дві функції: режим енергозбереження (PSM) та розширений перерваний прийом (eDRx). Вони дають змогу кінцевим пристроям переходити в режим глибокого сну годинами або навіть днями, не втрачаючи реєстрації в мережі. Кінцеві пристрої уникають моніторингу каналу управління низхідній лінії зв'язку протягом тривалого періоду часу для економії енергії. Такі функції енергозбереження використовуються і в описаному далі EC-GSM.

2) EC-GSM: Хоча в деяких регіонах оголошено, що глобальна система мобільного зв'язку (GSM) буде виведена з експлуатації, оператори мобільної мережі (МНО) можуть захотіти продовжити свою діяльність на кількох ринках. За цим припущенням, 3GPP зараз пропонує стандарт GSM із розширеним покриттям (EC-GSM), який має на меті розширити охоплення GSM на +20 дБ, використовуючи смугу SUB-GHZ для кращого проникнення сигналу в приміщення. Бюджет зв'язку в діапазоні 154 дБ-164 дБ спрямований залежно від потужності передачі. За допомогою лише оновлення програмного забезпечення мереж GSM старий GPRS-спектр може упакувати нові логічні канали, визначені для розміщення пристроїв EC-GSM. EC-GSM використовує повторювані передачі та методи обробки сигналів, щоб покращити охоплення та потужність застарілих GPRS. Два методи модуляції, а саме Гаусівська двопозиційна

частотна маніпуляція з мінімальним зсувом (GMSK) та 8-арний фазовий зсув (8PSK), забезпечують різну швидкість передачі даних з піковою швидкістю 240 кбіт/с за останньою методикою. Стандарт був випущений в середині 2016 року і спрямований на підтримку 50 тисяч пристроїв на базову станцію та покращені функції безпеки і конфіденційності порівняно зі звичайними рішеннями на основі GSM.

3) NB-IoT: NB-IoT – це вузькосмугова технологія, яка була доступна в межах випуску-13 близько середини 2016 року. NB-IoT має на меті забезпечити розгорнуту функціональність, тривалий термін служби акумулятора, низьку вартість пристрою і складність та сигнал розширення покриття NB-IoT несумісний з 3G, але може співіснувати з GSM, GPRS та LTE. NB-IoT може підтримуватися лише оновленням програмного забезпечення на основі наявної інфраструктури LTE. Вона може бути розгорнута всередині одного GSM-носія потужністю 200 кГц, всередині одного блоку фізичних ресурсів LTE (PRB) 180 кГц або всередині діапазону охорони LTE. Порівняно з eMTC, NB-IoT додатково скорочує витрати та енерговитрати, зменшуючи вимоги до швидкості передачі даних та пропускної здатності (потребує лише 180 кГц) та спрощує розробку протоколу та підтримку мобільності. Крім того, підтримується автономне розгортання у виділеному ліцензованому спектрі.

NB-IoT має на меті покриття 164 дБ, обслуговуючи до 50 тисяч кінцевих пристроїв на клітинку з потенціалом для збільшення потужності, додаючи більше носіїв NB-IoT. NB-IoT використовує множинний доступ з одночасним частотним поділом (FDMA) у висхідній лінії зв'язку, а ортогональний FDMA (OFDMA) у низхідній лінії зв'язку. Швидкість передачі даних обмежена 250 кбіт/с для багатотонного зв'язку низхідної лінії зв'язку та 20 кбіт/с для однотонного зв'язку висхідної лінії зв'язку. Як було підкреслено, до втрати зв'язку 164 дБ радіостанція на базі NB-IoT може досягти тривалості роботи від батареї 10 років при передачі в середньому 200 байт даних на день.

Після публікації специфікацій випуску-13 стандарт NB-IoT був підданий критиці:

• У NB-IoT підтверджено лише половину повідомлень через обмежену потужність низхідної лінії зв'язку. Це передбачає неможливість реалізувати програми IoT, які потребують підтвердження всіх потоків даних висхідної лінії зв'язку, якщо додаток не реалізує певну форму механізмів надійності. Останнє може призвести до збільшення складності додатків та більшого споживання енергії через додаткову обробку.

• Використання агрегації пакетів (комбінування кількох пакетів та відправлення їх у вигляді одного більшого пакету) у рішеннях на основі

3GPP покращує ефективність, але ціною додаткової затримки, яка може бути небажаною для чутливих додатків IoT, що спричиняють затримку.

• Трафік NB-IoT – це найкращі зусилля, і тому в часи важкого трафіку голосу / даних динамічне перерозподілення спектра для зменшення перевантаженості для останнього класу трафіку може вплинути на ефективність додатків NB-IoT. Крім того, після розгортання пристрою NB-IoT, ймовірно, залишається на 10–20 років, що досить вище за цикл оновлення пристрою порівняно з традиційними мобільними телефонами (зазвичай, 2 роки). Деякі програми можуть зайняти більше часу, щоб забезпечити беззбитковість та окупність інвестицій. Навіть, якщо з'являться нові покоління стільникового зв'язку, можуть виникнути питання щодо довговічності розгорнутого рішення, наприклад, ситуація, подібна до деяких операторів, які припиняють обслуговувати свою мережу GSM для відновлення спектра для LTE. Це може залишити клієнтів на межі, оскільки, можливо, не тривіально / економічно доцільно оновлювати термінали, і це вагомий аргумент.

• Відсутність комерційного розгортання залишає відкритими питання щодо реального часу роботи акумулятора та його продуктивності, досяжної в реальних умовах.

D. IETF

IETF має на меті підтримати екосистему LPWA з переважними фірмовими технологіями шляхом стандартизації кінцевих підключень на основі IP для пристроїв та додатків наднизької потужності. IETF вже розробив стек IPv6 для бездротових мереж особистої області низької потужності (6LoWPAN). Однак ці зусилля зі стандартизації зосереджуються на застарілих бездротових мережах на базі IEEE 802.15.4, які підтримують порівняно більш високі швидкості передачі даних, довші розміри корисного навантаження та менший діапазон, ніж сьогодні більшість технологій LPWA. Однак відмінні особливості технологій LPWA створюють реальні технічні проблеми для зв'язку з IP. По-перше, технології LPWA неоднорідні: кожна технологія маніпулює даними у різних форматах, використовуючи різні фізичні та MAC-шари. По-друге, більшість технологій використовують смуги ISM, на які поширюються суворі регіональні норми, що обмежують максимальну швидкість передачі даних, час в ефірі та частоту передачі даних. По-третє, для багатьох технологій характерна сильна асиметрія зв'язку між висхідною та низхідною лініями, що зазвичай обмежує можливості низхідної лінії зв'язку. Отже, запропоновані стеки IP повинні бути досить легкими, щоб увійти в ці дуже жорсткі обмеження базових технологій. На жаль, ці виклики ще не вирішені у попередніх спробах зі стандартизації IETF.

У квітні 2016 року була сформована робоча група з мереж з низькою потужністю широкої площі (LPWAN) під парасолькою IETF. Ця група визначила проблеми та проєктний простір для підключення IPv6 для технологій LPWA. Майбутні зусилля, ймовірно, можуть досягати кількох стандартів, що визначають повний стек IPv6 для LPWA (6LPWA), який може безпечно та масштабовано з'єднувати пристрі LPWA між собою та їх зовнішньою екосистемою. Більш конкретні технічні проблеми, з якими повинна вирішуватися група IETF, описані нижче:

• Стиснення заголовка. Максимальний розмір корисного навантаження для технологій LPWA обмежений. Методи стиснення заголовків повинні бути пристосовані до цих невеликих розмірів корисної навантаження, а також до рідкісного та нечастого трафіку пристроїв LPWA.

• Фрагментація та повторна збірка. Більшість технологій LPWA не підтримують фрагментацію та повторну збірку на рівні 2 (L2). Оскільки пакети IPv6 часто занадто великі, щоб не знаходитися в одному пакеті L2, механізми фрагментації та повторної збірки пакетів IPv6 мають бути визначені.

• Управління. Для управління кінцевими пристроями, додатками, базовими станціями та серверами існує потреба у надлегких протоколах сигналізації, які можуть ефективно працювати над обмеженою технологією L2. З цією метою IETF може розглянути ефективні протоколи сигналізації рівня застосування.

• Безпека, цілісність та конфіденційність. Підключення до IP повинно зберігати безпеку, цілісність та конфіденційність даних, що обмінюються через мережі доступу до радіо LPWA та за її межами. Більшість технологій LPWA використовують симетричну ключову криптографію, в якій кінцеві пристрої та мережі поділяють один і той же секретний ключ. Більш надійні та стійкі методи та механізми можуть бути досліджені.

E. LORa™ Alliance

Як описано в розділі 1.3, LORa є власним фізичним рівнем для підключення LPWA. Однак верхні шари та архітектура системи, визначені Альянсом LORa™ згідно зі специфікацією LORaWAN™, були оприлюднені у липні 2015 р.

На шарі MAC використовується проста схема ALOHA, яка, поєднуючись з фізичним шаром LORa дає змогу отримати кілька пристроїв для одночасного спілкування, але використовуючи різні канали та / або ортогональні коди (тобто коефіцієнти розповсюдження). Кінцеві пристрої можуть підключитися до будь-якої базової станції без додаткових надмірних сигналів. Базові станції з'єднують кінцеві пристрої за допомогою зворотного зв'язку до мережевого сервера, мозку системи

LORAWAN, який пригнічує повторювані прийоми, адаптує радіолінії доступу та передає дані на відповідні сервери додатків. Потім сервери додатків обробляють отримані дані та виконують визначені користувачем завдання.

LORAWAN передбачає, що пристрої матимуть різні можливості відповідно до вимог програми. Отже, LORAWAN визначає три різні класи кінцевих пристроїв, які підтримують двонаправлену комунікацію, але мають різні затримки в низхідній лінії зв'язку та потужність. Пристрій класу А досягає найдовшого терміну експлуатації, але з найвищою затримкою. Він прослуховує зв'язок по низхідній лінії зв'язку лише незабаром після передачі висхідної лінії зв'язку. Пристрій класу В, крім того, може планувати прийом низхідній лінії зв'язку від базової станції через певні інтервали часу. Отже, лише в ці узгоджені епохи програми можуть надсилати керуючі повідомлення кінцевим пристроям (можливо, для виконання функції приводу). Нарешті, пристрій класу С, зазвичай, працює від мережі, має можливість постійно слухати та отримувати передачі по низхідній лінії зв'язку з найкоротшою можливою затримкою в будь-який час.

Стандарт LORAWAN використовує криптографію симетричного ключа для автентифікації кінцевих пристроїв у мережі та збереження конфіденційності даних програми.

F. WEIGHTLESS-SIG

Спеціальна цільова група WEIGHTLESS запропонувала три відкритих стандарти LPWA, кожен із яких має різні функції, діапазон та енергоспоживання. Ці стандарти можуть діяти як в ліцензійному, так і в ліцензійному спектрі.

WEIGHTLESS-W використовує чудові властивості розповсюдження сигналу під час пауз рядків у телевізорі. Він підтримує декілька схем модуляції, включно із 16-квадратурною амплітудною модуляцією (16-QAM) та диференціальною BPSK (DBPSK) а також широкий спектр факторів розповсюдження. Залежно від бюджету зв'язку пакети, розміри яких перевищують 10 байт, можуть передаватися зі швидкістю від 1 кбіт до 10 Мбіт/с. Кінцеві пристрої передають на базові станції у вузькій смузі, але з меншим рівнем потужності, ніж базові станції для економії енергії. WEIGHTLESS-W має один недолік. Спільний доступ до білих просторів телевізора дозволений лише в кількох регіонах, тому WEIGHTLESS-SIG визначає два інші стандарти в діапазоні ISM, який є загальнодоступним для спільного доступу.

WEIGHTLESS-N – це стандарт UNB для лише одностороннього зв'язку від кінцевих пристроїв до базової станції, що забезпечує значну ефективність енергії та нижчу вартість, ніж інші стандарти

WEIGHTLESS. Він використовує схему модуляції DBPSK у діапазонах SUB-GHZ. Однак одностороння комунікація обмежує кількість випадків використання для WEIGHTLESS-N.

WEIGHTLESS-P поєднує двосторонній зв'язок між двома фізичними шарами, які не є власною розробкою. Він модулює сигнали, використовуючи GMSK і Quadrature Phase Shift Keying (QPSK), дві добре відомі схеми, прийняті в різних комерційних продуктах. Тому кінцеві пристрої не потребують фірмового чипсета. Кожен окремий вузький канал 12,5 кГц у діапазоні ISM SUB-GHZ пропонує швидкість передачі даних в діапазоні від 0,2 кбіт до 100 кбіт/с. Повна підтримка підтверджень та двосторонніх комунікаційних можливостей дають змогу здійснити оновлення програмного забезпечення у прямому ефірі.

Як і LORAWAN, усі стандарти WEIGHTLESS використовують симетричну ключову криптографію для аутентифікації кінцевих пристроїв та цілісності даних програми.

G. DASH7 Альянс

Альянс DASH7 – це галузевий консорціум, який визначає повний вертикальний мережевий стек для підключення LPWA, відомий як DASH7 Alliance Protocol (D7AP). Починаючи зі стандарту ISO/IEC 18000-7 радіо-інтерфейсу для пристроїв активної радіочастотної ідентифікації (RFID), D7AP перетворився на стек, що забезпечує підключення середнього діапазону до датчиків і приводів малої потужності.

DASH7 використовує вузькосмугову модуляційну схему із застосуванням дворівневої GFSK у смугах SUB-GHZ. Порівняно з більшістю інших технологій LPWA, DASH7 має кілька помітних відмінностей. Спочатку використовується за замовчуванням деревоподібна топологія з можливістю вибору також зіркового макета. У першому випадку кінцеві пристрої спочатку підключаються до робочих підконтролерів, які потім підключаються до базових станцій, які завжди ввімкнено. Цей робочий циклічний механізм приносить більшу складність конструкції верхніх шарів. По-друге, протокол DACH7 MAC змушує кінцеві пристрої періодично перевіряти канал на можливу передачу низхідній лінії зв'язку, додаючи значну вартість простою на прослуховування. Так DASH7 отримує набагато меншу затримку для зв'язку по низхідній лінії зв'язку, ніж інші технології LPWA, через більше енергоспоживання. По-третє, на відміну від інших технологій LPWA, DASH7 визначає повний мережевий стек, що дозволяє додаткам та кінцевим пристроям спілкуватися один з одним без необхідності мати справу з тонкощами базових фізичних або MAC-шарів.

DASH7 реалізує підтримку виправлення помилок вперед та криптографію із симетричними ключами.

1.5. Завдання і напрями досліджень

Гравці LPWA дуже наполегливо намагаються внести інноваційні рішення, які можуть забезпечити так звану вищу продуктивність носія. Для цього виробники пристроїв, мережеві оператори та фахівці системної інтеграції сконцентрували свої зусилля на дешевому дизайні обладнання, надійному підключенні та повноцінній інтеграції додатків. Що стосується бізнесу, постачальники фірмових рішень поспішають вивести свої послуги на ринок і захопити свою частку в кількох вертикалях. У цій гонці легко, але контрпродуктивно проігнорувати важливі проблеми, з якими стикаються технології LPWA. У цьому розділі розглядаються ці виклики та деякі напрями досліджень для їх подолання та підвищення ефективності в довгостроковій перспективі.

А. Масштабування мереж до величезної кількості пристроїв Технологія LPWA з'єднує десятки мільйонів пристроїв, що передають дані безпрецедентного масштабу через обмежені та часто спільні радіоресурси. Ця складна проблема розподілу ресурсів ще більше ускладнюється кількома іншими факторами. По-перше, щільність пристроїв може суттєво змінюватися в різних географічних районах, що створює так звану проблему гарячих точок. Ці гарячі точки поставлять базові станції LPWA на стрес-тест. По-друге, перехресні технології втручання можуть дуже погіршити ефективність технологій LPWA. Ця проблема, безумовно, є більш серйозною для технологій LPWA, що функціонують у групах ISM, які не належать до ліцензій і не мають спільного доступу. Навіть ліцензовані стільникові технології LPWA, що працюють в межах діапазону із широкосмуговими послугами (як голосові та відеозаписи), однаково піддаються цьому ризику. Нескладно уявити сценарій, коли кілька каналів UNB за технологією LPWA одночасно перешкоджають одному широкосмуговому сигналу. Крім того, більшість технологій LPWA використовують прості протоколи MAC на основі ALOHA або CSMA, які не відповідають масштабам кількості підключених пристроїв.

У кількох роботах досліджується, чи LPWA-технології зможуть підтримувати велику кількість кінцевих пристроїв, що очікується при розгортанні в майбутньому систем масштабу міста та країни. На момент написання роботи для LORAWAN присутні лише кілька досліджень. Оцінка обмеження кількості вузлів, які можуть підтримуватися типовим розгортанням LORAWAN, становить 120 на 3,8 га, щільність пристроїв набагато менша, ніж очікувалося в міських умовах. Джорджоу та Раза також оприлюднили, що ймовірність покриття LORAWAN знижується експоненціально з кількістю кінцевих пристроїв через перешкоди. Як видається, обидва дослідження припускають, що кінцеві пристрої повинні адаптувати параметри зв'язку LORa, можливо, за допомогою більш

потужних базових станцій та використовувати різноманітність базових станцій для подолання цього обмеження.

Можна визначити кілька напрямів дослідження для вирішення проблеми потенціалу технологій LPWA. Вони включають використання різноманітності каналів, умовно-патогенний доступ до спектра та адаптивну стратегію передачі. Застосування стрибкової зміни каналів та базових станцій з кількома модемами може використовувати різноманітність каналів та обладнання, і це вже застосовується для наявних технологій LPWA. Різношарові рішення можуть адаптувати стратегії передачі до специфічних моделей перевезень пристроїв LPWA та зменшити вплив перехресних технологій. Крім того, потрібні вдосконалення наявних протоколів MAC для технологій LPWA для їх значного масштабування для великої кількості пристроїв, що передають лише короткі повідомлення.

У контексті стільникових мереж LPWA, якщо надмірний трафік IoT/M2M перевищує традиційний стільниковий трафік, оператори можуть розглянути можливість розгортання підтримки LPWA у неліцензованому спектрі. Таке опортуністичне використання радіочастотного спектру може отримати користь від використання когнітивних програм, визначених програмним забезпеченням (SDR). SDR можуть стати в нагоді, коли кілька технологій будуть конкурувати за спільний спектр.

Для обслуговування областей з більшою щільністю пристроїв мережі доступу LPWA можуть запозичити методи ущільнення із стільникового домену. Однак такі особливості технологій LPWA, як їх спеціалізовані методи модуляції, сильна асиметрія зв'язків та здебільшого некоординована робота кінцевих пристроїв, створюють серйозні проблеми, щоб утримати рівень перешкод на низькому рівні у щільних розгортаннях.

B. Контроль та зменшення інтерференції

Надалі кількість підключених пристроїв зазнаватиме експоненціальне збільшення, спричиняючи більш високий рівень перешкод один одному. Пристрої, що працюють у спільних діапазонах ISM, зазнають безпрецедентного рівня як перехресних технологічних втручань, так і самовтручання. Деякі дослідження вимірювання перешкод вже вказують на можливий негативний вплив на охоплення та потужність мереж LPWA. Крім того, багато технологій LPWA, серед яких LORa та SIGFOX, вдаються до простої схеми ALOHA для надання каналу доступу до кінцевих пристроїв малої потужності. Такий вибір говорити випадковим чином, не слухаючи інших, може не тільки погіршити продуктивність, але й породжує більш високі перешкоди. Крім того, ущільнення розгортання базової станції для розміщення більшої кількості пристроїв є основним джерелом перешкод в осередках LPWA і вимагає ретельного розгортання та проєктування базових станцій.

В анархії десятків бездротових технологій та величезної кількості пристроїв усі спільні канали, стійкі до перешкод зв'язку та ефективний обмін спектром є ключовими проблемами як з технічної, так і з регуляторної думки. Оскільки перешкоди змінюються у залежності від частоти, часу та простору, пристрої повинні адаптувати свої графіки передачі, щоб відчувати найменші перешкоди та найкращу надійність. Проєкти шарів PHY та MAC, що використовують таке різноманіття в таких значних масштабах, потребують подальшого дослідження. Регулюючим органам також може знадобитися рухатися вперед, щоб запропонувати правила для забезпечення ефективного обміну та співпраці між різними бездротовими технологіями в неліцензованих смугах.

C. Високошвидкісні методи модуляції передачі даних

Технології LPWA роблять компромісні рішення у швидкості передачі даних для досягнення великих відстаней. Деякі технології, особливо ті, що використовують модуляцію UNB у спільних діапазонах ISM, пропонують дуже низькі швидкості передачі даних та короткі розміри корисного навантаження, що обмежує їх потенційні випадки використання бізнесу. Для підтримки випадків використання обмеженої пропускної здатності доцільно реалізувати кілька схем модуляції для пристроїв. Відповідно до потреб програми пристрої можуть перемикатися між різними схемами модуляції, щоб одночасно забезпечити високу ефективність енергії, дальність дії та високу швидкість передачі даних.

Для цього потрібна гнучка та недорога апаратна конструкція, яка може підтримувати кілька фізичних рівнів, кожен з яких може запропонувати додаткові компроміси, щоб відповідати діапазону та вимогам швидкості передачі даних.

D. Взаємодія між різними технологіями LPWA

Зважаючи на те, що ринок спрямований на інтенсивну конкуренцію між різними технологіями LPWA, можна припустити, що і в майбутньому їх може існувати декілька. Отже, сумісність між цими неоднорідними технологіями має вирішальне значення для їх довгострокової успішності. Оскільки майже не підтримується взаємодія між різними технологіями, потреба в стандартах, що їх об'єднють, є великою. Деякі зусилля зі стандартизації в межах ETSI, IEEE, 3GPP та IETF, які обговорені в розділі 1.4, будуть розглянуті у контексті цієї проблеми взаємодії.

Однак для повної сумісності необхідно вивчити кілька напрямів. По-перше, IP вже може підключати бездротові пристрої короткої відстані за допомогою сітчастої мережі. Особливості технологій LPWA обмежують безпосередню реалізацію одного і того ж стека IP на пристроях LPWA. Альтернативні рішення, що базуються на шлюзах або серверних рішеннях, є життєздатними кандидатами. Однак усі подібні рішення

мають добре поєднуватися з кількістю пристроїв, не погіршуючи продуктивність. По-друге, використання проміжного програмного забезпечення IoT та методів віртуалізації може зіграти вирішальне значення у підключенні пристроїв LPWA. Проміжне програмне забезпечення IoT може підтримувати безліч технологій радіодоступу та у такий спосіб робити простою інтеграцію LPWA-технологій із рештою IoT-технологій. Ці проміжні програми також можуть консолідувати дані з різних джерел, щоб запропонувати кінцевим споживачам послуги із доданою вартістю.

Інтероперабельність залишається відкритим викликом. Тестування та розвинені інструменти з відкритим кодом для технологій LPWA ще не доступні для оцінки механізмів взаємодії.

E. Позиціювання

Мережі LPWA очікують, що вони отримають значний прибуток від логістики, управління низкою поставок та особистих додатків IoT, де розташування мобільних об'єктів, транспортних засобів, людей та тварин може викликати величезний інтерес. Точна підтримка локалізації є важливою особливістю для відстеження цінностей, дітей, людей похилого віку, домашніх тварин, вантажів, автомобілів тощо. Насправді це розглядається як важлива особливість для нових програм.

Позиціювання мобільних пристроїв здебільшого досягається властивостями прийнятих сигналів та вимірюванням часу розповсюдження сигналу. Усі такі методи вимагають дуже точної синхронізації часу та достатньої щільності розгортання базових станцій. Це досить легко досягти при ретельному розгортанні та плануванні мережі. Однак дуже обмежена пропускна здатність каналів технологій LPWA та часто відсутність прямого шляху між кінцевими пристроями та базовими станціями вводять дуже велику помилку локалізації. Отже зробити точну локалізацію лише за допомогою приймачів LPWA є справжньою проблемою.

Мережам LPWA потрібні нові методи, які не лише використовують властивості фізичного рівня, але й комбінують інші усталені методи позиціювання, щоб переконатися, що точність є достатньо хорошою для реальних додатків відстеження.

F. Оптимізація та адаптованість посилань

Якщо технологія LPWA дозволяє, кожну окрему ланку потрібно оптимізувати для високої якості зв'язку та низького енергоспоживання, щоб максимізувати загальну ємність мережі. Кожна технологія LPWA дає змогу здійснити кілька налаштувань рівня зв'язку, які запроваджують компроміси між різними показниками продуктивності – швидкість передачі даних, час в етері, охоплення області тощо.

Однак для роботи таких методів зазвичай потрібен зворотний зв'язок від шлюзу до кінцевих пристроїв по низхідній лінії зв'язку.

Асиметрія посилань, що викликана низхідною лінією багатьох технологій LPWA (наприклад, SIGFOX), яка має меншу ємність, ніж висхідна лінія, є головною перешкодою в цьому випадку.

G. Випробувальні панелі та інструменти LPWA

Технології LPWA дають змогу використовувати декілька розумних міських додатків. Кілька пробних тестів для міста, наприклад, SmartSantander з'явилося в останнім часом. Такі тестові панелі містять датчики, оснащені такими різними бездротовими технологіями, як мережі Wi-Fi, мережі IEEE 802.15.4 та стільникові мережі. Однак наразі немає відкритих тестових панелей для мереж LPWA. Тому широко випускати системи LPWA та порівнювати їх ефективність у столичному масштабі не вигідно. На сьогодні лише кілька емпіричних досліджень порівнюють дві нові технології LPWA за тих же умов. На наш погляд, це є важливим бар'єром для входу для потенційних клієнтів. Забезпечення технологій LPWA як наукового інструментарію для широкої громадськості через міські органи влади може виступати як міра побудови упевненості. Тимчасом аналітичні моделі та тренажери були нещодавно запропоновані для популярних технологій LPWA.

H. Автентифікація, безпека та конфіденційність

Автентифікація, безпека та конфіденційність – це найважливіші особливості будь-якої системи зв'язку. Стільникові мережі забезпечують перевірені механізми аутентифікації, безпеки та конфіденційності. Використання модулів посвідчення абонента (SIM) спрощує ідентифікацію та автентифікацію стільникових пристроїв. Технології LPWA, зважаючи на їх вартість та енергоспоживання, не лише встановлюють простіші протоколи зв'язку, але й відходять від аутентифікації на основі SIM. Отже, необхідні методи та протоколи, щоб забезпечити еквівалентну або кращу підтримку аутентифікації для технологій LPWA. Крім того, щоб переконатися, що кінцеві пристрої не піддаються жодним ризикам безпеки протягом тривалого часу, важливою особливістю є підтримка оновлень у прямому ефірі (OTA). Відсутність належної підтримки оновлень OTA становить великий ризик для безпеки більшості технологій LPWA.

Margelis та ін. [6] висвітлити кілька вразливих місць безпеки трьох відомих технологій LPWA, а саме SIGFOX, LORAWAN та INGENU. Наприклад, кінцеві пристрої у мережах SIGFOX та LORAWAN не шифрують корисну навантаження програми та запит на приєднання до мережі відповідно, що може призвести до підслуховування. Крім того, більшість технологій LPWA використовують симетричну ключову криптографію, в якій кінцеві пристрої та мережі поділяють один і той же секретний ключ. Надійні та малопотужні механізми аутентифікації, безпеки та конфіденційності потребують подальшого дослідження.

I. Мобільність та роумінг

Роумінг пристроїв між різними мережевими операторами – життєво важлива особливість, що відповідає комерційному успіху стільникових мереж. Хоча деякі технології LPWA не мають поняття роумінгу (робота в глобальному масштабі, наприклад, SIGFOX), є й інші, які не мають підтримки для роумінгу на момент написання цього повідомлення. Головною проблемою є забезпечення роумінгу без шкоди для терміну експлуатації пристроїв. З цією метою підтримка роумінгу повинна покласти мінімальне навантаження на кінцеві пристрої, що працюють від акумулятора. Оскільки кінцеві пристрої спрацьовують агресивно, доцільно припустити, що пристрої малої потужності не можуть постійно отримувати трафік по низхідній лінії зв'язку. Обмін даними по висхідній лінії зв'язку слід використовувати більш потужно. Призначення мережі має бути вирішено в резервних системах на відміну від мережі доступу. Усі питання, пов'язані зі спритністю роумінгового процесу та ефективним управлінням ресурсами, мають бути вирішені.

Необхідно узгодити подальші моделі виставлення рахунків та розподілу доходів для роумінгу в різних мережах.

Міжнародний роумінг у регіонах, контрольованих різними правилами спектра (наприклад, США, Європа чи Китай), є ще більш складним завданням. Щоб відповідати різним нормам спектра, кінцеві пристрої мають бути обладнані можливостями для виявлення першої області, а потім дотримуватися відповідних регіональних вимог під час передачі даних. Це додає складності кінцевим пристроям, а отже, і вартості. Отже, необхідна проста конструкція з низькими витратами для підтримки міжнародного роумінгу.

J. Підтримка угод про рівень обслуговування

Можливість надання певних гарантій якості може бути конкурентним диференціатором між різними операторами LPWA. Хоча пропонувати гарантії QoS у ліцензованому спектрі порівняно просто, більшість фірмових технологій вибирають спектр ліцензійних випробовувань для швидшого виходу на ринок. Як наслідок, вони повинні дотримуватися регіональних норм щодо використання спільного спектра, які можуть обмежувати радіотехнічний цикл та передану радіочастоту. Перехресне технологічне втручання також впливає на ефективність технологій LPWA.

Забезпечення продуктивності на рівні носіїв на спектрі, що ділиться на кілька неузгоджених технологій та десятків тисяч пристроїв на базовій станції, є вагомим завданням. Угоди про рівень обслуговування (SLAs), ймовірно, будуть порушені через фактори, що не підпадають під контроль мережевих операторів. Тому очікується, що підтримка SLAs буде обмежена в групах, звільнених від ліцензій. Вивчення таких

надзвичайно галасливих середовищ, для визначення, чи можуть бути надані якісь послаблені гарантії статистичного обслуговування, є хорошим потенційним напрямом досліджень.

Можуть бути різні випадки використання, коли кілька технологій співпрацюватимуть між собою. Специфікація ETSI LTN перераховує кілька таких випадків використання для співпраці стільникового зв'язку та LPWA. На приклад, коли зв'язок стільникового зв'язку недоступний, технології LPWA все ще можуть бути використані як резервний варіант для надсилання лише критичного трафіку з низькою швидкістю передачі даних. Крім того, періодичні постійні повідомлення стільникових мереж можуть бути делеговані енергоефективним мережам LPWA. Можуть бути й інші нові способи співпраці між LPWA та стільниковими мережами. Наприклад, технології LPWA можуть сприяти формуванню маршруту для зв'язку між пристроєм та пристроєм у стільникових мережах. Коли для деяких пристроїв поза стільниковим покриттям потрібно побудувати маршрут з кількома стрибками, щоб досягти стільникової інфраструктури, підключення LPWA може допомогти виявити близькість до інших обслуговуваних пристроїв. Ці випадки використання можуть мати сильне звернення до громадських програм безпеки. Крім того, як відомо, технології LPWA розроблені спеціально для наднизьких швидкостей передачі даних. Необхідність періодично надсилати великі обсяги трафіку може задовольнятись додатковим стільниковим зв'язком, який можна активувати лише на вимогу.

Спільна власність LPWA та стільникових мереж у поєднанні зі зниженням цін на пристрої LPWA та підключення є вагомим бізнесом для вищезазначених випадків використання. Однак існує потреба подолати багато проблем, пов'язаних із системою.

L. Підтримка аналітики даних

Порівняно з абонентським абонентом, середній дохід, отриманий одним підключеним пристроєм M2M / IoT, досить невеликий. Тому оператори мережі бачать чіткий стимул у розширенні свого бізнесу за межі чистого зв'язку задля вищої прибутковості. Один зі способів зробити це – розширення мереж LPWA з витонченою підтримкою аналітики даних, яка може перетворити необроблені дані в контекстно-релевантну інформацію для кінцевих користувачів. Такі знання можуть підтримувати кінцевих користувачів у прийнятті розумних рішень, зароблянні вищих прибутків або зниженні їхніх експлуатаційних витрат. Отже, оператори мережі можуть монетизувати це, продаючи знання кінцевим користувачам.

Однак існують величезні проблеми, пов'язані із наданням мережі LPWA як послуги кінцевим споживачам. Це вимагає єдиного управління бізнес-платформою та масштабованої інтеграції із хмарою. Однією з

головних проблем є також змога пропонувати послуги на замовлення для багатьох вертикальних галузей, ефективно охоплюючи різні випадки використання, в ідеалі, за допомогою єдиної технології LPWA.

1.6. Відповіді бізнесу

З встановленням парадигми зв'язку M2M використання системи 2G здавалося розумним результатом для задоволення вимог цих програм. Враховуючи дефіцит спектра у всьому світі та великі капітальні витрати, здійснені на придбання нового спектра, оператори постають перед дилемою, чи продовжувати використовувати системи 2G для обслуговування клієнтів M2M, чи переробляти спектр, створюючи такі нові технології, як LTE та його варіанти. Оголошення від кількох операторів про перехід до останнього створили дірку на ринку. Відтоді кілька нових технологій LPWA наполегливо намагаються усунути цей проміжок з надією подати свою пропозицію на позицію полюса.

Варто підкреслити, що для кожного з цих підходів не існує жодного параметра, який би не мав своїх плюсів і мінусів (рис. 1.4). Ринок все ще готовий до захоплень, і гравці мають кілька стратегічних варіантів, які необхідно розглянути, залежно від їхніх обставин. Тим, хто потребує негайного впровадження рішення IoT, доведеться хеджувати свої ставки на LORa, SIGFOX, INGENU, WEIGHTLESS-N тощо, тимчасом як інші можна дозволити собі зачекати, поки 3GPP не вкаже такі стандарти, як NB-IoT, який ще не працює. В одночас, оператори стільникового зв'язку, схоже, захищали свої ставки на LORa та SIGFOX, доки кілька операторів робили великі інвестиції в те чи інше. У будь-якому разі це виглядає як безпрограшна ситуація для операторів, незалежно від того, як ситуація складається, оскільки ці технології можуть відігравати допоміжну роль до потенційного стандарту NB-IoT, який зараз використовується. Також той факт, що оператори інвестували в ці технології, зменшує невизначеність з позиції довговічності для тих, хто приймає ці рішення.

Передбачається, що LORa, SIGFOX та INGENU продовжуватимуть кидати виклик гегемонії стільникових гравців, і всі четверо, наймовірніше, поділять пиріг у довгостроковій перспективі. Очікується, що буде застосовано різний ступінь прийняття у різних сегментах ринку та моделі ціноутворення, які, ймовірно, матимуть значний вплив на успіх різних технологій.

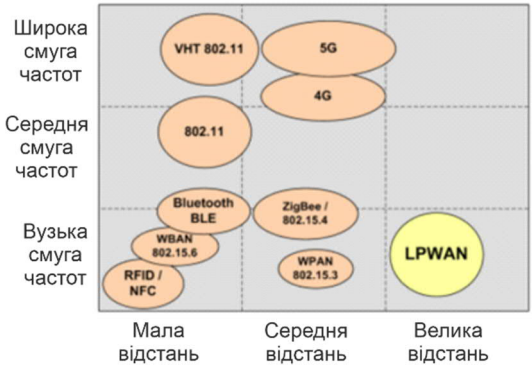

Рис. 1.4. Характеристики різних технологій зв'язку

2. ОПЕРАЦІЙНІ СИСТЕМИ ДЛЯ ПРИСТРОЇВ НИЗЬКОГО КЛАСУ В ІНТЕРНЕТІ РЕЧЕЙ

Інтернет речей (IoT) випливає з наявності безлічі дешевих, крихітних, енергоефективних комунікаційних пристроїв (тобто речей). Кілька стандартних протоколів зв'язку були розроблені на різних рівнях для мережевого стеку IoT, водночас IPv6, зазвичай є вузьким місцем на мережевому рівні. Наявність таких протоколів дає змогу різнорідним пристроям бути взаємопов'язаними та мати доступ до Інтернету.

З апаратного погляду, Інтернет речей складається з різнорідного обладнання – навіть більше, ніж у традиційному Інтернеті. Пристрої IoT можна класифікувати на дві категорії залежно від їхніх можливостей та продуктивності. Перша категорія – це висококласні пристрої IoT, до яких належать одноплатні комп'ютери (Rasberry Pi [7]), та смартфони. Високоякісні пристрої IoT мають достатньо ресурсів та адекватних характеристик для запуску програмного забезпечення на базі традиційних операційних систем (ОС), таких як Linux або BSD.

Друга категорія – це пристрої IoT низького класу, які занадто обмежені для використання цих традиційних ОС. Популярні приклади низькопродуктивних пристроїв IoT вміщують Arduino, Econotag, Zolertia Z1, IoT-LAB M3 вузли, Open-Mote вузли та TelosB motes, деякі з яких показано на рис. 2.1. У цьому розділі зупинимося на таких низькоякісних пристроях IoT, оскільки вони створюють нові завдання для дизайнерів ОС, коли йдеться про обробку даних за дуже обмежених апаратних ресурсів [8].

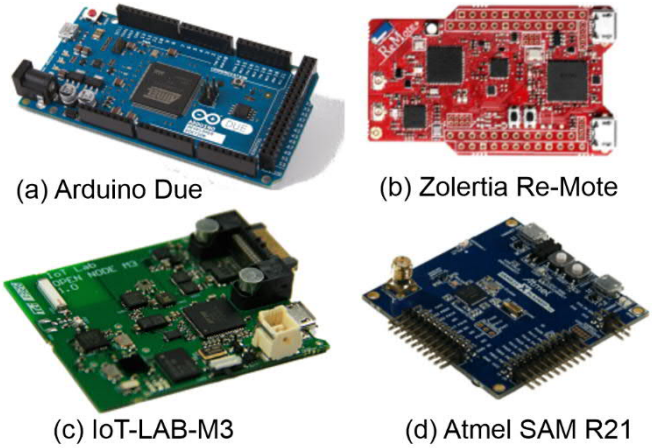

Рис. 2.1. Приклади IoT пристроїв низького класу

2.1. Пристрої IoT низького класу

Пристрої IoT низького класу, зазвичай дуже обмежені з точки зору ресурсів, включно з енергією, центральним процесором та обсягом пам'яті. Нещодавно Робоча група Інтернет-інженерії (IETF) стандартизувала класифікацію таких пристроїв за трьома підкатегоріями на основі обсягу пам'яті.

• Пристрої класу 0 мають найменші ресурси (~ 10 кБ оперативної пам'яті та ~ 100 кБ Flash), наприклад, спеціалізований компонент у бездротовій сенсорній мережі (WSN).

• Пристрої класу 1 мають ресурси середнього рівня (~ 10 кБ оперативної пам'яті та ~ 100 кБ флеш-пам'яті), що дає змогу розширити додатки та вдосконалити функції, а не елементарні частки, наприклад протоколи маршрутизації та безпечного зв'язку.

• Пристрої класу 2 мають більше ресурсів, але все ще дуже обмежені порівняно з висококласними пристроями IoT та традиційними Інтернет-хостами.

На пристроях класу 0 надзвичайна спеціалізація та обмеження ресурсів зазвичай роблять використання відповідної ОС непридатним. Тому програмне забезпечення, що працює на такому обладнанні переважно розробляється простими засобами та дуже специфічне для обладнання.

Однак пристрої IoT класу 1 і вище, зазвичай, менш спеціалізовані. Програмне забезпечення може альтернативно перетворити такий пристрій на Інтернет-маршрутизатор, хост або сервер із стандартним

мережевим стеком та перепрограмованими / взаємозамінними програмами, що працюють понад цього стека. Тому зараз з'являються нові бізнес-моделі, засновані (частково) на портативному апаратно-незалежному програмному забезпеченні та додатках, що працюють на пристроях IoT класу 1 і вище. Отже, кілька великих компаній нещодавно анонсували нові ОС, розроблені спеціально для роботи на пристроях IoT, включно з Huawei, ARM та Google. Дійсно, на такому обладнанні часто бажано мати програмні примітиви, що дають змогу легко виробляти код, незалежний від апаратного забезпечення. Більш загально, існує потреба в інтерфейсах програмування програм (API), окрім простого програмування, які можуть задовольнити широкий спектр випадків використання IoT, щоб полегшити масштабну розробку, розгортання та обслуговування програмного забезпечення. Такі примітиви програмного забезпечення, зазвичай, надаються ОС. Тому зосередимося на ОС, які підходять для пристроїв класу 1 та класу 2.

Ми зазначаємо, що, на жаль, закон Мура не допоможе у цьому контексті: передбачається, що пристрої IoT стануть меншими, дешевшими та енергоефективнішими, замість того, щоб забезпечити значно більшу пам'ять чи потужність центрального процесора. Тому в найближчому майбутньому пристрої IoT низького класу з кількома кілобайтами пам'яті, а це пристрої класу 1 та класу 2, ймовірно, залишаться домінуючими в IoT.

2.2. Операційні системи для пристроїв IoT низького класу

Як зазначалося раніше, традиційні операційні системи, такі, як Linux або BSD, не застосовуються на пристроях IoT низького класу, оскільки вони не можуть працювати на обмежених ресурсах, що надаються на такому обладнанні. Як наслідок, IoT страждає від недостатньої сумісності між багатьма несумісними вертикальними ізольованими рішеннями. Ми стверджуємо, що IoT не реалізує свій потенціал доти, доки не відбудеться великий вибух програмного забезпечення, що призведе до появи кількох фактично стандартних ОС, що забезпечують узгоджені API та SDK на неоднорідних апаратних платформах IoT.

У цій роботі проведено аналіз ОС, які можуть стати фактично стандартною ОС для пристроїв низького класу IoT. Зазначається, що рішення, які забезпечують якнайменший розмір пам'яті, зазвичай, обмежуються конкретним випадком використання, а тому не можуть стати загальною ОС для пристроїв IoT. На відміну від цього, ми будемо орієнтуватися на універсальні рішення (або, принаймні, на однакові величини), які забезпечують найкращий рівень комфорту, одночасно задовольняючи вимоги до середньої пам'яті близько 10 КБ оперативної пам'яті або

більше , і ~100 кБ флеш або більше; тобто пристрої класу 1 і вище, згідно з класифікацією IETF.

Під рівнем комфорту ми маємо на увазі взаємодію з рештою Інтернету, включно а) сумісністю з протоколами IP з точки зору мережі, та б) з погляду систем сумісність із стандартними засобами програмування, моделями та мовами використовується на Інтернет-хостах. У цій роботі ми зосередимося на ОС з відкритим кодом, але ми також розглянемо опитування щодо альтернатив із закритим кодом. Однією з причин цього фокусу є те, що кілька найбільш розповсюджених ОС для пристроїв IoT низького класу є відкритими, і що вони пропонують більші можливості для детального вивчення їх проєктування та реалізації, як це потрібно для цього дослідження. Низку додаткових причин зосередження уваги на відкритому коді також буде зазначено далі.

2.3. Вимоги до операційних систем для IoT

У цьому розділі дається огляд різноманітних вимог, яким має відповідати загальна ОС для пристроїв низького класу IoT.

А. Невеликий обсяг пам'яті

У зіставленні з іншими підключеними машинами, пристрої IoT набагато обмеженішими ресурсами, особливо з погляду пам'яті. Одже, однією з вимог до загальної ОС для IoT є відсутність таких обмежень пам'яті. Тимчасом як ПК, смартфони, планшети або ноутбуки забезпечують пам'ять Гіга- або ТераБайт, пристрої IoT зазвичай забезпечують кілька кілобайт пам'яті, тобто в мільйон разів менше. Це спостереження стосується як енергонезалежної (RAM), так і постійної (ROM) пам'яті. Для того, щоб не вкластися в обмеження розміру пам'яті, дизайнери додатків IoT повинні забезпечити набір оптимізованих бібліотек (потенційно міжшарових), що забезпечують загальну функціональність IoT та ефективні структури даних.

Визначення правильного компромісу між продуктивністю, зручним API та невеликим розміром пам'яті ОС є нетривіальною проблемою. Наприклад, у багатьох випадках дизайнеру ОС доводиться визначати найкраще місце між використанням оперативної пам'яті та ПЗУ. Крім того, необхідно знайти баланс між розумними керівними принципами програмування та правилами кодування, яких потрібно дотримуватися, з одного боку, та високим ступенем модульності та налаштованості, який бажаний для широкого кола випадків використання, з іншого боку.

В. Підтримка різнорідного обладнання

Хоча різноманітність обладнання та протоколів, що використовуються в сучасному Інтернеті, порівняно невелике з архітектурного погляду, в IoT ступінь неоднорідності вибухає. Велика різноманітність

випадків використання призвела до розвитку великої різноманітності апаратних та комунікаційних технологій. Пристрої IoT базуються на різних архітектурах та сімействах мікроконтролерів (MCU), включно з 8 бітними (наприклад, Intel 8051/52, Atmel AVR), 16 бітними (наприклад, TI MSP430), 32 бітні (ARM7, ARM Cortex-M, MIPS32 і навіть x86) архітектури – 64-розрядні архітектури також можуть з'явитися з часом. Крім того, ключові характеристики системи дуже відрізняються: наприклад, деякі пристрої IoT забезпечують сотні кілобайт оперативної пам'яті, але не мають постійної пам'яті для зберігання виконуваного коду (і, отже, виникає потреба завантажувати як код, так і дані в оперативну пам'ять). Однією з таких плат є все ще популярна плата Redwire Econotag, яка базується на Freescale MC13224V. Інші пристрої IoT дуже обмежені в плані оперативної пам'яті, але оснащені великою кількістю ПЗУ, наприклад, мікроконтролер STM32F100VC ARM Cortex-M3. Аналогічно, пристрої IoT можуть бути оснащені широким спектром комунікаційних технологій, як описано нижче у підрозділі 2.3-C. Зауважте, що така неоднорідність може виникати навіть під час одного розгортання, завдяки чому багато різних типів пристроїв беруть участь у різних завданнях для досягнення загальної мети. Отже, однією з вимог – і ключовою проблемою – для загальної ОС для IoT є підтримка цієї неоднорідності в апаратних архітектурах та комунікаційних технологіях.

C. Мережеве підключення

Основний сенс наявності пристроїв IoT полягає в тому, що вони можуть взаємозв'язуватися та спілкуватися між собою або з Інтернетом. Тому пристрої IoT зазвичай оснащені одним (або декількома) мережевими інтерфейсами. Методи зв'язку, що використовуються в IoT, охоплюють не лише широкий спектр радіотехнологій з низьким енергоспоживанням (наприклад, IEEE 802.15.4, Bluetooth/BLE, DASH7 та EnOcean), а й різні дротові технології (наприклад, PLC, Ethernet або кілька шин) системи. На відміну від сценаріїв WSN, зазвичай, очікується, що пристрої IoT безперешкодно інтегруються з Інтернетом; тобто може взаємодіяти наскрізно з іншими машинами в Інтернеті. Поєднання необхідності підтримувати технології декількох рівнів зв'язку та необхідності спілкування з іншими хостами Інтернету привело до використання мережевих стеків на основі протоколів IP безпосередньо на пристроях IoT. Отже, ключовою вимогою до загальної ОС для IoT є підтримка різнорідних технологій рівня зв'язку та мережевого стеку на основі протоколів IP, що мають відношення до IoT. Крім того, як вказує розвиток Linux протягом багатьох років (що є очевидним прикладом майбутнього дизайну), також бажано, щоб ОС могла обслуговувати кілька мережевих стеків та постійний розвиток мережевих стеків.

D. Енергоефективність

Багато пристроїв IoT працюватимуть від акумуляторів або інших обмежених джерел енергії. Наприклад, розумні лічильники та інші пристрої автоматизації будинків / будівель повинні працювати роками з одним зарядом батареї. На глобальному рівні енергоефективність також потрібна через велику кількість пристроїв IoT, які передбачається використовувати (десятки мільярдів). Обладнання IoT загалом – мікроконтролери, радіоприймачі, датчики – надає функції для ефективної енергетичної роботи. Якщо програмне забезпечення IoT не використовує ці функції (наприклад, переведення пристроїв у найглибший режим сну якомога частіше), енергоефективність не досягається. Отже, ключовою вимогою до ОС для IoT є а) забезпечити варіанти енергозбереження для верхніх шарів та б) максимально використовувати ці функції, наприклад, використовуючи такі методи, як циклічний радіозв'язок, або шляхом мінімізації кількості періодичних завдань, які потрібно виконати. Наприклад, періодичний системний таймер, який планувальники використовують для зрізування часу, веде до системи, яка ніколи не переходить у глибокі режими відключення, і, отже, її необхідно уникати, якщо це можливо.

E. Можливості в реальному часі

Точні терміни та своєчасне виконання мають вирішальне значення в різних випадках використання IoT, наприклад, інтелектуальних додатках для охорони здоров'я: як мережі на тілі (BAN), з кардіостимуляторами, що забезпечують бездротовий моніторинг та управління, або в інших сценаріях, включно виконавчі механізми та / або роботи в контексті промислової автоматизації, або Спеціальна мережа автомобілів (VANET). ОС, яка може виконувати вимоги до своєчасного виконання, називається операційною системою в режимі реального часу (RTOS) і розроблена у такий спосіб, щоб гарантувати працездатність при найгірших часу виконання та часу затримки обробки переривань. Отже, ще однією вимогою до загальної ОС для IoT є RTOS, що зазвичай означає, що функції ядра повинні працювати з детермінованим часом роботи. Японський відкритий стандарт для операційної системи в режимі реального часу, ITRON, популярний у цій галузі, хоча він спрямований здебільшого на побутову електроніку.

F. Безпека

З одного боку, деякі системи IoT є частиною критичної інфраструктури або промислових систем із наслідками для безпеки життя. З іншого боку, оскільки вони підключені до Інтернету, пристрої IoT, зазвичай, відповідають високим стандартам безпеки та конфіденційності. Окрім всеохоплюючого виклику управління довірою, виклики безпеки IoT містять цілісність даних, автентифікацію та контроль доступу в різних частинах

архітектури IoT. Отже, вимога (і виклик) для ОС для IoT полягає у забезпеченні необхідних механізмів (криптографічних бібліотек та протоколів безпеки), водночас зберігаючи гнучкість та зручність використання. І останнє, але не менш важливе: оскільки програмне забезпечення з певним ступенем складності ніколи не може забезпечити на 100% від помилок, а стандарти безпеки розвиваються (за участю різних зацікавлених сторін, серед як промисловість, уряд, споживачі тощо), потрібні механізми оновлення програмного забезпечення на вже розгорнутих пристроях IoT. Також потрібно якомога більше використовувати відкритий код.

2.4. Ключовий вибір дизайну

На успіх і застосовність ОС для IoT впливають як технічні, так і політичні чи організаційні фактори. У цьому розділі ми розглянемо ключові технічні альтернативи проєктування ОС, а також відповідні нетехнічні міркування.

А. Технічні властивості

Вибір дизайну, що стосується, загальної моделі ОС, стратегії планування або апаратної абстракції, має великий вплив на можливості та гнучкість системи. У цьому розділі ми розглянемо такий вибір та як вони впливають на придатність ОС для випадків використання IoT.

Загальна архітектура та модульність. Першим дизайнерським рішенням, яке необхідно прийняти для будь-якої ОС, є вибір типу ядра. Цей вибір має великий вплив на загальну архітектуру системи та її модульність. Загальна архітектура ОС IoT зображена на рис. 2.2. Можна розрізнити підхід екзоядра, мікроядра, монолітний підхід або гібридний підхід. Основною ідеєю підходу екзоядра є якомога менше абстракцій між додатком та обладнанням і необхідність зосередитись переважно на уникненні конфліктів ресурсів та перевірці рівнів доступу. Підхід до мікроядра спрямований на більшу кількість функціональних можливостей (мінімалістичний набір функцій) в ядрі, але водночас вимагає дуже мало пам'яті, а також забезпечує багато місця та гнучкості для решти системи, а також надійність (оскільки драйвер пристрою, що аварійно завершує роботу не впливає на стабільність всієї системи). Однак через типову відсутність блоку керування пам'яттю (MMU) на низькоякісних пристроях IoT, буферні та стекові потоки все ще можуть відбуватися і мати серйозний вплив на систему. Нарешті, основна ідея монолітного підходу полягає в тому, що всі компоненти системи розробляються разом, що може призвести до більш простого та загального більш ефективного проєктування.

Резюме: потрібно вибирати між більш надійним та еластичним мікроядром або менш складним і більш ефективним монолітним ядром – або вибрати гібридний підхід.

Модель планування. Іншою важливою частиною будь-якої ОС є планувальник, який впливає на інші важливі властивості, серед яких енергоефективність, можливості реального часу або модель програмування. Зазвичай є два типи планувальників: попереджувальні планувальники та непередбачувальні (або кооперативні) планувальники. ОС може надавати різні планувальники, які можна вибрати під час збірки. Попереджувальний планувальник може перервати будь-яке (неядерне) завдання в будь-якій заданій точці, щоб дати змогу виконувати інше завдання протягом обмеженого часу. У кооперативній моделі кожен потік відповідає за поступку, оскільки жодне інше завдання, а в деяких випадках навіть ядро, не може перервати завдання.

У багатьох випадках попереджувальний планувальник вимагає періодичної галочки таймера, яку іноді називають *систиком (systick)*, для того, щоб призначати зрізи часу для кожного завдання. Ця вимога зазвичай не дає змогу пристрою IoT перейти в найглибший режим економії енергії, оскільки принаймні один апаратний таймер повинен залишатися активним. Крім того, MCU переходить у повний активний режим на кожному *систику*. Часовий графік часто використовується для ОС із користувальницьким інтерфейсом (UI), щоб імітувати паралельне виконання кількох завдань. Для ОС IoT це здебільшого непотрібно, оскільки вони не мають безпосереднього користувача і, отже, не потребують користувацького інтерфейсу.

Резюме: попереджувальний планувальник призначає час процесора кожному завданню, тимчасом як різні завдання повинні поступатися самим у кооперативній моделі.

Виділення пам'яті. Як описано в розділі 2.2, пам'ять, зазвичай, є дуже дефіцитним ресурсом на пристроях IoT. Отже, потрібна вишукана обробка пам'яті. Важливе питання полягає в тому, чи виділяється пам'ять статичним чи динамічним способом, і цей вибір також впливає на інші критерії проєктування системи. Розподіл статичної пам'яті, здебільшого вимагає певного надмірного забезпечення та робить систему менш при-датною до змін вимог під час роботи. Динамічний розподіл пам'яті ускла-днює дизайн системи із двох основних причин. По-перше, такі функції, як malloc () та пов'язані з ними функції, зазвичай реалізовуються у визна-чений час недетермінованим способом у стандартних бібліотеках C і, отже, порушують будь-які гарантії в режимі реального часу. Для того, щоб використовувати динамічний розподіл пам'яті для додатків із вимо-гами в режимі реального часу, ОС має забезпечити спеціальні реалізації.

для детермінованого malloc (), такого як TLSF. По-друге, динамічний розподіл пам'яті створює необхідність обробляти ситуації, коли немає пам'яті або під час виконання, що може бути складно вирішити. Крім того, реалізації malloc на основі купи зазвичай спричиняють фрагментацію пам'яті, що змушує системи швидше заповнювати пам'ять.

Резюме: статичний розподіл пам'яті вносить певні накладні витрати на пам'ять через надмірне забезпечення та призводить до менш гнучких систем, тимчаом динамічне розподілення пам'яті веде до більш складної системи і може суперечити вимогам у режимі реального часу.

Управління мережевим буфером. Центральним компонентом ОС IoT є мережевий стек, де шматки пам'яті, наприклад, пакети, мають спільно використовувати між рівнями. Двома можливими рішеннями для досягнення цього є копіювання пам'яті (memcpy ()) або передача покажчиків між кількома шарами. Хоча перше рішення є дорогим з погляду ресурсу, останнє породжує питання, хто відповідає за розподіл пам'яті. Делегуючи це завдання на верхні шари, робіть розробку додатків більш складною та менш зручною. Залишаючи це завдання для нижчих шарів, таких, як драйвер пристрою, можна зробити систему менш життєздатною. Можливим підходом до вирішення цього конфлікту є розробка центрального диспетчера пам'яті, запропоноване для TinyOS або RIOT.

Резюме: пам'ять для обробки пакетів у мережевому стеку може виділятися кожним шаром або передаватися як еталон між шарами

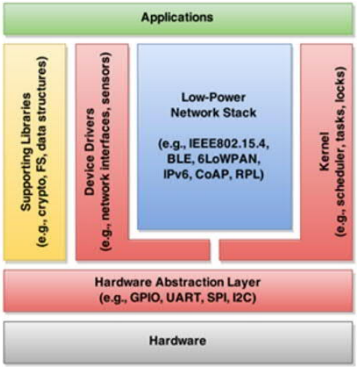

Рис. 2.2. Типові компоненти Операційної системи для простих пристроїв IoT, включно зі стеком загального протоколу IPv6 з низьким споживанням енергії

Модель програмування. Модель програмування визначає, як розробник програми може моделювати програму. Типові моделі програмування у сфері операційних систем IoT можна розділити на керовані

подіями системи та багатопотокові системи. У керованій подіями системі, яка, наприклад, широко використовується для ОС WSN, кожне завдання має ініціюватися (зовнішньою) подією, наприклад, перериванням. Цей підхід часто супроводжується простим циклом подій (замість більш складного планувальника) та моделлю спільного стеку. Модель програмування, заснована на багатопотоковості, надає розробнику можливість запускати кожне завдання у власному контексті потоку та обмінюватися даними між собою за допомогою API Inter Process Communication (IPC).

Резюме: системи, керовані подіями, можуть бути більш ефективними в пам'яті, а багатопотокові системи полегшують дизайн програми.

Мови програмування. Основним вибором мови програмування ОС є вибір між стандартною мовою програмування, переважно, ANSI C або C++, та специфічною для ОС мовою або діалектом. З одного боку, надання специфічних для ОС функцій мови дає змогу покращити продуктивність або безпеку, які мови низького рівня, зокрема C, не підтримують. З іншого боку, вони перешкоджають використанню усталених і зрілих засобів розвитку. Специфікація стандартів для мов програмування, особливо специфікація ANSI для C та C ++, означала значний поштовх для розвитку програмного забезпечення загалом та для ОС зокрема. Незважаючи на свій вік (і зростання нових мов програмування), мова програмування C все ще залишається найважливішою і найбільш широко використовуваною мовою програмування (поряд з Assembler), коли йдеться про програмування на ОС, а також для частин нижчого рівня, таких як планування або драйвер пристрою. Однак, для вдосконалення прикладного програмування на вищих рівнях можуть бути доступні більш досконалі мови з більшим набором функцій.

Резюме: стандартні мови програмування спрощують портативність і дають змогу використовувати добре відомі засоби розробки. Спеціальні для ОС мови та розширення мов можуть підвищити продуктивність та безпеку системи.

Модель драйвера та апаратний рівень абстракції. Системи IoT багато в чому взаємодіятимуть із навколишнім середовищем або пасивно, здійснюючи зондування через всі види датчиків, або активно – через виконавчі механізми, серед яких двигуни або системи освітлення. Отже, мікроконтролери для цих систем зазвичай оснащені безліччю різних периферійних пристроїв, таких, як АЦП / ЦАП, інтерфейси, серед яких SPI, I^2C, шина CAN або послідовних ліній, і GPIO. Отже, гнучкий та досить зручний інтерфейс драйвера має вирішальне значення для ОС IoT.

На додаток до моделі драйвера для підключення зовнішніх пристроїв, наприклад, датчиків, приводів, трансиверів, модель може також абстрагуватися від базового обладнання загалому. Апаратний рівень

абстракції може забезпечити чітко визначений інтерфейс для процесора, пам'яті та обробки переривань, щоб зробити перенесення на нові платформи простою задачею.

Резюме: добре визначений апаратний рівень абстракції та модель драйвера можуть суттєво покращити дизайн системи, але вводить певну кількість накладних витрат – або з погляду рядків коду, або з погляду накладних витрат під час виконання.

Інструменти налагодження. Як зазначалося раніше, вибір мов програмування також зумовлює можливі інструменти для використання, включно з інструментами для налагодження. Добре налагоджені ланцюжки інструментів, такі, як навколо колекції компіляторів GNU (GCC), зазвичай, мають відповідні інструменти налагодження, наприклад, налагоджувач GNU (GDB). Однак, щоб запустити живу систему налагодження, цільова плата повинна забезпечити адекватний інтерфейс, такий, як JTAG або Spy-Bi-Wire. На жаль, не кожен пристрій IoT надає такий інтерфейс, і тому потрібні інші засоби налагодження.

Типовим допоміжним інструментом є використання printf () для простого налагодження послідовного інтерфейсу, наприклад, USART. У деяких випадках навіть простий алгоритм миготіння світлодіодів іноді можна знайти як примітивний замінник налагодження. Якщо комусь бракує доступу до пристроїв, як це часто буває у розгорнутих мережах IoT, необхідно надати інші засоби доступу до інформації про налагодження. Наприклад, цього можна досягти за допомогою періодичних діагностичних повідомлень, що надсилаються по мережі, або за допомогою журналів, записаних на зовнішній флеш-пам'яті.

Резюме: використання стандартних мов програмування загалом дає змогу використовувати стандартні засоби налагодження, але апаратні обмеження можуть спричинити потребу в інших, більш простих засобах налагодження за допомогою послідовного виводу або навіть світлодіодного блимання.

Набір функцій. ОС можна розділити на ядро та функціональні можливості вищого рівня. Зазвичай ядро забезпечує планувальник, модель для завдань, взаємного виключення (мьютекс) та інших форм синхронізації та таймери. Якщо ОС підтримує багатопотоковість, API, зазвичай, також включатиме функції для IPC. На вищих шарах можна знайти системні бібліотеки, серед яких оболонка, ведення журналу, криптографічні функції або мережеві стеки. Через зазвичай відсутні модулі MMU на пристроях IoT, такі програми та бібліотеки програм, здебільшого, працюють в тому ж адресному просторі, що і операції ядра, і тому можуть знизити стабільність системи.

На додаток до мережевих протоколів, функції вищих рівнів, які викликають особливий інтерес для ОС для пристроїв низького класу IoT, включають бездротові оновлення, динамічне завантаження та зв'язування або бібліотеки для полегшеного шифрування та дешифрування.

Резюме: загальний набір функцій ОС може бути описаний розміром її API.

Тестування. Як і для всіх програмних систем, тестування відіграє вирішальну роль у розробці ОС IoT. Зокрема, для високо розподілених потоків робіт із розвитку, як це часто можна зустріти у великих проєктах із відкритим кодом, розгортання середовища безперервної інтеграції (CI) неминуче. Цей IP зазвичай містить тести побудови та інтеграції, а також модульні та регресійні тести. Конкретні випробування тестування систем IoT виникають із розподіленого характеру цих систем та того факту, що вони глибоко вбудовані та часто дуже обмежені. Широко застосовуваний підхід до вирішення апаратної частини тестування, наприклад, тестування драйверів пристроїв, полягає у використанні апаратних засобів емуляції, наприклад, MSPSim або Emul8. Мережеві емулятори та симулятори, такі, як Cooja або ns-2/ns-3, які дають змогу інтегрувати код ОС, дуже корисні в цьому контексті.

Резюме: розподілений характер та обмеження апаратного забезпечення роблять ретельне тестування складним, але вирішальним завданням.

B. Нетехнічні властивості

На придатність технічно складної ОС – зокрема для комерційного використання – також впливають такі аспекти, як ліцензія, ремонтопридатність, робочий потік або постачальник ОС. У цьому підрозділі ми розглядаємо нетехнічні аспекти.

(Відкриті) Стандарти. Найважливішою характеристикою будь-якої ОС є її здатність забезпечувати перенесення додатків на апаратних платформах та архітектурах – в ідеалі, без будь-яких додаткових зусиль. Для спрощення перенесення програмного забезпечення між кількома ОС були також розроблені стандартизовані API (такі, як POSIX, визначені IEEE та Open Group). Однак на пристроях IoT низького класу впровадження стандартного API, призначеного для операційних систем загального призначення, таких, як Linux, може бути складним через обмеження розміру програмного забезпечення (і насправді навіть на ПК мало хто з ОС може претендувати на повну відповідність POSIX). Однак для безперебійного портування програмного забезпечення між кількома ОС, має бути передбачена додаткова підтримка стандартів мови програмування, таких, як ANSI C99 або C++. Нарешті, стандарти не тільки важливі на системному рівні, але й неминучі на рівні мережі. Що до стандартів на

мережевому рівні досвід показує, що використання специфікацій із відкритим доступом, таких як стандартизовані, наприклад, Інженерною робочою групою (IETF), за замовчуванням є кращим за інші підходи.

Резюме: використання стандартів покращує портативність та сумісність.

Сертифікація. Для деяких випадків використання, зокрема для критично важливих систем у таких додатках, як автоматизація будівель, найважливішими властивостями системи є можливості в режимі реального часу, надійність або детермінованість. У цих випадках сертифікація через незалежні установи стає невідворотною вимогою для ОС. Типовим і широко відомим прикладом такої сертифікації є стандарт IEC 61508, який називається «Функціональна безпека електричних / електронних / програмованих електронних систем, пов'язаних з безпекою». Додаткові сертифікації, які стосуються ОС на пристроях IoT, – це програма логотипу IPv6 Forum «IPv6 Ready» та нещодавно розпочата програма відповідності та сертифікації Альянсу IPSO.

Резюме: спеціально для розгортання у промислових програмах та програмах, що мають найважливіше значення для безпеки, сертифікація всього програмного забезпечення, що працює в системі IoT, може бути обов'язковою.

Зрілість коду. Зрілість програмного забезпечення навіть важче виміряти, ніж якість документації. Дуже грубим показником є вік проєкту у поєднанні з кількістю учасників та користувачів. Хоча сертифікація в багатьох випадках є переважно правовою гарантією, фактичну надійність та правильність системи набагато важче оцінити.

Резюме: у багатьох випадках ретельне тестування та широке впровадження в комерційних програмах є кращим показником зрілості ОС, ніж просто вік проєкту або сертифікацій.

Ліцензія коду. Загалом можна розрізнити три категорії ліцензій: невільні, дозвільні відкриті джерела та ліцензії copyleft. Якщо ОС випущена за невільною ліцензією, ОС доступна лише у вигляді двійкових даних, або з клієнтів стягується додаткова плата за отримання вихідного коду, що заважає виправленням помилок та вдосконаленням третіх сторін, обмежуючи кількість учасників. Дозвільні ліцензії, напр. BSD, MIT або Apache License надають розробникам та користувачам високий ступінь свободи і часто легше приймаються промисловістю, ніж ліцензії copyleft – хоча для деяких компаній все навпаки. Можливим недоліком дозвільних ліцензій є потенційна фрагментація спільноти та бази коду, що часто призводить до ситуації, коли не всі функції доступні – або, принаймні, не в межах одного сховища. Навпаки, ліцензії copyleft, такі, як GPL (з винятками або без них) та LGPL, менш легко приймаються

деякими галузями промисловості, але можуть призвести до набагато більш інтегративного співтовариства та загальної кодової бази, як це можна бачити на показовому прикладі Linux.

Резюме: відкритий код – зокрема ліцензії copyleft – не завжди може бути першим вибором галузі, але пропонує шанси на підвищення якості та безпечніший код завдяки збільшенню кількості авторів та рецензентів.

Постачальник ОС. Код ОС може надаватися у різних формах та різними об'єктами (залежно від обраного типу ліцензії). Його може надавати постачальник, який фактично розробляє програмне забезпечення, або третя сторона, яка може також надавати комерційну підтримку. У випадку рішень із відкритим кодом код часто надається самим співтовариством розробників через сховища систем контролю версій, таких як Git, Subversion або Mercurial. Спільнота, зазвичай, надає підтримку у вигляді розумних відповідей через онлайн-форуми, відстежувачі відкритих видань та списки розсилки для таких типів проєктів. Ця підтримка є надзвичайно важливою на практиці, і тому настійно рекомендується віддавати перевагу проєкту з відкритим кодом з активною спільнотою, що працює над відкритим кодом, ніж проєкту без активної спільноти, або з раніше активною спільнотою. Зауважте, що іноді професійний консалтинг програмного забезпечення пропонується не лише для комерційних ОС, а й для безкоштовних ОС з відкритим кодом.

Резюме: спосіб розповсюдження та ступінь підтримки ОС сильно залежить від її ліцензії.

2.5. ОС, які перспективні для IoT

У цьому розділі ми коротко оглядаємо ОС, які подають найбільш перспективні підходи до загальної ОС IoT. Мета цього розділу – вичерпний, а не поглиблений аналіз (що є основним предметом наступного розділу). Ми будемо розрізняти ОС з відкритим кодом, ОС із закритим кодом та інші бібліотеки програмного забезпечення або проміжне програмне забезпечення для IoT. Якщо не зазначено інше, всі ОС написані мовою програмування C, тимчасом як деякі специфічні для апаратного забезпечення частини можуть бути реалізовані мовою асемблера.

A. ОС з відкритим кодом

У цьому розділі перераховані переважні ОС з відкритим кодом, націлені на пристрої IoT.

1) Contiki: Contiki спочатку розроблявся як ОС для мереж WSN, що працюють на 8-бітних мікроконтролерах, обмежених пам'яттю, але тепер також працює на 16-бітних мікроконтролерах та сучасних пристроях IoT на базі ARM 32-бітові MCU. Він заснований на підході

спільного планування, керованого подіями, з підтримкою полегшеної псевдопоточності. Написані мовою програмування С, деякі частини ОС використовують абстракції на основі макросів (наприклад, Protothreads) і фактично вимагають від розробників врахування певних обмежень щодо того, який тип мовних функцій вони можуть використовувати. Код Contiki доступний за ліцензією BSD на GitHub3 та інших платформах, водночас велика різноманітність форків (процеси, майже копії) розробляється самостійно (включно з багатьма версіями ОС із закритим кодом). Contiki має кілька мережевих стеків, включно з популярним стеком uIP, з підтримкою IPv6, 6LoWPAN, RPL та CoAP; та стек Rime, який забезпечує набір розподілених абстракцій програмування. Contiki розробляється з 2002 року і дотепер є однією з найбільш часто використовуваних ОС з обмеженими вузлами.

2) RIOT: RIOT був розроблений з урахуванням особливих вимог IoT і спрямований на зручну для розробників модель програмування та API, наприклад, подібно до того, що зазнає Linux. RIOT – це RTOS на основі мікроядра з підтримкою багатопоточності, що використовує архітектуру, успадковану від FireKernel. Хоча ОС написана на C (ANSI99), програми та бібліотеки також можуть бути реалізовані на C++. Вихідний код доступний на GitHub4 під LGPLv2.1. RIOT має кілька мережевих стеків, включно з власною реалізацією повного стека 6LoWPAN (стек gnrc), порт стека 6TiSCH OpenWSN та порт інформаційно-орієнтованого мережевого стека CCN-lite. RIOT розробляється як така з 2012 року зростаючим у всьому світі співтовариством з відкритим кодом.

3) FreeRTOS: FreeRTOS – це популярний RTOS, який переноситься на багато MCU. Його переважне мікроядро має підтримку багатопоточності. Зараз він розроблений Real Time Engineers Ltd., а його код доступний на сторінці проєкту під модифікованим GPL, що сприяє комерційному використанню із закритими програмами (лише ядро має залишатися відкритим кодом). Хоча він не забезпечує власного мережевого стека, сторонні мережеві стеки можуть бути використані для підключення до Інтернету. FreeRTOS розробляється з 2002 року і до цього часу є однією з найбільш часто використовуваних RTOS для відкритих кодів для обмежених вузлів.

4) TinyOS: Разом з Contiki TinyOS є найвидатнішою ОС для WSN-програм, орієнтована на дуже обмежені 8-бітові та 16-бітові платформи і відома своїм витонченим дизайном. TinyOS та nesC розробили мовні примітиви та абстракції програмування, щоб запобігти якомога більшій кількості помилок завдяки структурі програмного забезпечення та підвищити ефективність пам’яті, зменшивши фактичний пов’язаний код до мінімуму. Однак досить складний дизайн у поєднанні з індивідуальною

мовою програмування ускладнює навчання і тому йому бракує більшої спільноти розробників. Він схожий на підхід, керований подіями, коли декілька компонентів або модулів можуть бути фактично підключені, як це описано в конфігураціях відповідно до вимог. Він написаний на діалекті мови програмування C, який називається nesC. Його вихідний код доступний в Інтернеті за ліцензією BSD на GitHub5. Включений мережевий стек BLIP реалізує стек 6LoWPAN. TinyOS розробляється з 2000 року і дотепер є однією з найбільш часто використовуваних ОС з обмеженими вузлами з Contiki.

5) OpenWSN: OpenWSN містить мережевий стек 6TiSCH, базовий планувальник та пакет підтримки плат (BSP), тобто просту апаратну абстракцію, що дає можливість запускати OpenWSN на дюжині апаратних платформ IoT. Отже, OpenWSN – це більше мережевий стек, ніж повнофункціональна ОС. Код OpenWSN доступний в Інтернеті за ліцензією BSD на GitHub6. Основним напрямом роботи OpenWSN є мережевий стек 6TiSCH, включно з реалізацією поправки MAC IEEE 802.15.4e. OpenWSN розробляється з 2010 року зростаючою у всьому світі спільнотою з відкритим кодом.

6) nuttX: ОС nuttX націлена на повну відповідність стандартам POSIX та ANSI і підтримує мікроконтролери, що варіюються від 8 до 32-бітових архітектур. NuttX може бути побудований і як мікроядро, і як монолітна версія. Він дуже модульний і має можливості в режимі реального часу, а також безпроблемний планувальник. Вихідний код доступний за ліцензією BSD на Sourceforge7. Інтегрований мережевий стек містить підтримку IPv4 та IPv6 з різними протоколами верхнього рівня. NuttX розробляється з 2007 року.

7) eCos: Вбудована конфігурувана операційна система (eCos) підтримує 16, 32 та 64 біти вбудованого обладнання. Код eCos доступний за спеціальною ліцензією, заснованою на GPL, з виключенням зв'язування (визнано FSF). Хоча версія eCos з відкритим кодом здається досить неактивною, комерційна версія (eCosPro від eCosCentric) знаходиться в стадії активної розробки. eCos не надає власний мережевий стек як такий, але підтримує сторонні мережеві стеки (lwIP та мережевий стек FreeBSD). Вихідний код доступний у сховищі Mercurial8. eCos розробляється з 2002 року, але частини кодової бази застаріли.

8) mbedOS: нещодавно ARM випустила випуск технологічного попереднього перегляду (з маркуванням 15.11) своєї майбутньої ОС для пристроїв низького класу IoT, яка називається mbed OS. Виходячи з попереднього перегляду, mbedOS фокусується виключно на 32-розрядній вбудованій архітектурі ARM та підтримує невелику кількість платформ (поки що 5, хоча найближчим часом ми можемо очікувати набагато

більше). Серед експериментальних функцій, продемонстрованих у попередньому перегляді, є реалізація 6LoWPAN (із закритим джерелом), яка претендує на реалізацію специфікації Thread 1.0, кілька визначень інтерфейсу, порт PolarSSL та підтримку Bluetooth Low Energy. mbed розробляється з 2009 року, але до цього часу він був зосереджений на забезпеченні апаратного рівня абстракції, а не на ОС.

9) Сімейство мікроядер L4: ОС L4 відповідають суворому дизайну мікроядер та спочатку були створені для подолання низької продуктивності попередніх ОС на основі мікроядер у середині 1990-х. Пізніші реалізації були розроблені для незалежності платформи, поліпшення безпеки, ізоляції та надійності. Відомим представником цього сімейства є seL4, розроблений у 2006 році групою NICTA з особливим акцентом на безпеку, надійність та офіційну перевірку. Однак більшість ОС на основі мікроядер L4 не відповідають обмеженням пристроїв класу 1. Виняток становить мікроядро F9, орієнтоване на конкретні пристрої на базі ARM Cortex-M3/M4. Хоча багато членів цієї родини мають ліцензію за ліцензією GPL або BSD, не всі вони мають відкритий код.

10) uClinux: Це порт ядра Linux 2.x для процесорів без MMU і зі значно меншим розміром пам'яті, ніж Linux. Незважаючи на те, що uClinux виграє від багатого набору функцій Linux (включно з API, повним стеком TCP/IP та чудовою підтримкою файлової системи), він має вимоги до пам'яті, які не відповідають низькоякісним пристроям IoT, таким, як пристрої класу 1 [8], які є предметом цього розгляду. Вихідний код доступний на Sourceforge9. uClinux розробляється з 1998 року.

11) Android та Brillo: ця мобільна ОС Android, розроблена Google, є варіантом Linux, орієнтованим переважно на смартфони та планшети, але також використовується в автомобілях, годинниках, телевізорах та іншій побутовій електроніці. Концепція програм, доступних через Інтернет-магазини, де користувачі можуть купувати та завантажувати прикладне програмне забезпечення, стимулювала еволюцію смартфонів. Незважаючи на те, що ядро Android є відкритим кодом – як вимагає GPL від Linux, – багато драйверів пристроїв та апаратна підтримка є закритим вихідним кодом. Подібно до інших систем на базі Linux, Android не може працювати на низькоякісних пристроях IoT, таких, як пристрої класу 1.

У 2015 році Google оголосив про Brillo, зменшену версію Android, яка зможе працювати на пристроях IoT, пропонуючи кілька десятків мегабайт пам'яті. Отже, Brillo вимагає значно менше апаратних ресурсів, ніж Android. Оскільки це все ще варіант Linux, однак його не можна використовувати на пристроях IoT низького класу, які є предметом цього опитування, і тому ми не будемо викладати його технічні деталі.

12) Інші ОС з відкритим кодом: для повноти ми згадаємо нижче інші ОС з відкритим кодом. Однак, оскільки вони не такі помітні, ми опишемо їх менш докладно.

• ChibiOS/RT – це RTOS, розроблена з 2007 року в межах модифікованого GPL, за винятком зв'язування, та спрямована на високу продуктивність 8, 16 та 32-бітних мікроконтролерів.

• CooCox CoOS – це безкоштовний та відкритий RTOS, спеціально розроблений для платформ ARM Cortex-M, який постачається разом із повноцінною IDE, розробленою у 2009 році.

• ERIKA Enterprise – це RTOS, орієнтована на вбудовані автомобільні системи. Він підтримує 8, 16 і 32-бітні мікроконтролери, має підтримку багатоядерних систем і ліцензується згідно з GPL v2, за винятком зв'язків.

• MansOS – це ще одна ОС WSN, яка спрямована на легку розробку, налагодження та підтримує 8-бітні AVR та 16-бітні MSP430 MCU.

• NanoQplus, розроблений на ETRI, націлений на пристрої WSN класу 0 і забезпечує багатопоточність та механізм захисту пам'яті.

• nanoRK – це RTOS для WSN з акцентом на резервування ресурсів для завдань, розроблений з 2005 року для платформ MSP430.

• Nut/OS з'явився внаслідок RTOS під назвою Liquorice, Nut/OS фокусується на обмежених пристроях із дротовим (Ethernet) з'єднанням.

• RTEMS – це відкритий RTOS з акцентом на API відкритого стандарту, підтримку мультипроцесорів та жорсткі гарантії реального часу.

• Є й інші ОС з відкритим кодом із домену WSN, такі як SOS, MANTIS OS, Lorien або LiteOS, але вони переважно неактивні і ніколи не націлені на сценарії IoT .

У [8] узагальнюється, що ОС, такі як Contiki, FreeRTOS або RIOT, відповідають більшості вимог, виведених у 2.3, водночас інші такі підходи, як uClinux, Arduino та Android, не можуть їх виконати.

Б. ОС із закритим кодом

На додаток до вищезазначених ОС з відкритим кодом, кілька ОС із закритим кодом мають характеристики, придатні для домену IoT. Деякі постачальники, хоч і є власністю, пропонують обмежений доступ до свого вихідного коду для клієнтів, зареєстрованих користувачів чи академічних інститутів. Однак ці ОС часто спочатку розроблені для інших доменів і, зазвичай не мають важливих функцій, таких, як енергозберігаючі механізми або нещодавно стандартизовані протоколи IoT. Тим не менше, деякі ОС із закритим кодом можуть бути адаптовані для роботи на пристроях класу 0 та класу 1, і ми назвемо деякі найбільш відповідні приклади нижче.

1. ThreadX: ThreadX – це RTOS, розроблена Express Logic, Inc., яка нещодавно була придбана ARM (і може стати ядром mbed OS 3.0). ThreadX базується на мікроядрі RTOS (іноді його називають пікоядром), яке підтримує багатопотоковість та використовує превентивний планувальник. Ядро надає два прийоми для усунення інверсії пріоритетів: успадкування пріоритету, що підвищує рівень пріоритету завдання під час виконання критичного розділу, та поріг випередження, що відключає попередження випуску потоків нижче вказаного пріоритету. Додаткові функції, такі як мережевий стек, підтримка USB, файлова система або графічний інтерфейс, можна придбати як окремі продукти.

2) QNX: спочатку розроблений компанією Quantum Software Systems у 1982 році, QNX був придбаний компанією Research in Motion (RIM) у 2010 році. Він був одним із перших комерційно успішних RTOS на базі мікроядер та забезпечував UNIX-подібний API. Потужний IPC від QNX послужив натхненням для багатьох наступних ОС, таких як RIOT. Поточна версія, яка називається QNX Neutrino, підтримує численні архітектури, але жодна з них не відповідає вимогам пристроїв класу 1.

3) VxWorks: Розроблений спочатку у 1987 році Wind River (який зараз належить Intel), VxWorks є монолітним ядром, яке здебільшого підтримує ARM-платформи та платформи Intel, включно з новим Quark SoC. VxWorks підтримує IPv6 та інші функції Інтернету речей, але не має підтримки стеку 6LoWPAN і не може працювати на обмежених пристроях Інтернету речей, як визначено RFC 7228, які є предметом цього опитування.

4) Wind River Rocket: іншою ОС, розробленою Wind River, є Rocket, яка націлена на конкретні сценарії IoT. Наразі Rocket підтримує єдину апаратну платформу: плата Intel Galileo Gen 2, яка пропонує кілька мегабайт оперативної пам'яті та ПЗУ. ОС тісно пов'язана із використанням хмарної платформи Wind River Helix.

5) PikeOS: PikeOS розробляється з 1991 року компанією під назвою SYSGO AG (нині належить Thales). PikeOS – це RTOS на основі мікроядра, який забезпечує безпеку та захист та діє як гіпервізор для інших ОС. PikeOS, який спочатку називався P4, є нащадком сімейства мікроядер L4. PikeOS пропонує декілька API, може розміщувати різні гостьові ОС та сертифіковані відповідно до декількох відповідних стандартів, включно з IEC 61508 або EN 50128.

6) embOS: embOS розроблена компанією Segger Microcontroller Systems, компанією, що надає засоби розробки та програмування, а також програмне забезпечення для вбудованих пристроїв. embOS – це текст, написаний на ANSI C, що містить пріоритетний, безпроблемний, попереджувальний планувальник і націлений на різні обмежені 8-бітні, 16-бітні

та 32-бітні MCU. Мережевий стек (включно з ZigBee), підтримка USB, графічний інтерфейс та система файлів доступні як окремі додаткові продукти.

7) Nucleus RTOS: Nucleus – це RTOS, розроблена компанією Mentor Graphics, компанією, що займається електронною автоматизацією проектування, яка придбала колишнього постачальника Nucleus, Accelerated Technology у 2002 році. Nucleus дає змогу програмувати на C++, сумісний із POSIX та інтерфейс Micro ITRON. Nucleus має багатий набір функцій, включно з IP-мережевим стеком, і його можна зменшити до десятків кілобайт, проте він не належить до RTOS з найменшими розмірами пам'яті.

8) Sciopta: Sciopta – це RTOS, що надається SCIOPTA Systems AG, з акцентом на критично важливі програми. Його мікроядро (з прямим повідомленням, що передає IPC) та планувальник написані на асемблері. Підтримувані архітектури містять ARM7, ARM9, ARM Cortex-M, ARM Cortex-A та PowerPC. SCIOPTA Systems також пропонує додаткові модулі, наприклад система файлів FAT або мережевий стек на основі IP.

9) µC/OS-II та µC/OS-III: µC/OS-II та µC/OS-III – це дві версії RTOS, що надаються Micrium Inc. Ці RTOS базуються на мікроядрі з можливостями багатопоточності та IPC. Порівнянно з версією µC/OS-II, випущеною у 2009 році, µC/OS-III містить деякі розширені функції, такі як необмежена кількість завдань та пріоритетів. Додаткові пакети програмного забезпечення, такі як графічний інтерфейс, файлова система або мережевий стек TCP/IP, також надаються Micrium і можуть бути інтегровані в µC/OS-III.

10) µ-velOSity: µ-velOSity – це безкоштовний RTOS, розроблений Green Hills Software (GHS). Добре інтегрована в IDE Green Hills (так звана MULTI), µ-velOSity записана на MISRA-сумісному ANSI C і базується на мікроядрі. Подібно до інших комерційних ОС IoT, додаткові необхідні функції (наприклад, мережевий стек) надаються окремо. Однак зауважимо, що стек 6LoWPAN недоступний.

11) Windows CE: Windows CE – це версія ОС Windows для обмежених пристроїв, яка розробляється корпорацією Майкрософт з 1996 року. Windows CE працює в режимі реального часу та має багатий набір функцій. Однак для цього потрібні ПЗП та оперативна пам'ять у порядку мегабайт, а отже, націлені на пристрої, які не мають обмежених ресурсів, ніж пристрої низького класу IoT, які є предметом цього розгляду.

12) LiteOS Huawei: у 2015 році Huawei оголосила, що випустить LiteOS, операційну систему для пристроїв IoT. В оголошенні стверджується, що LiteOS від Huawei вміщуватиметься в межах 10 кбайт пам'яті і

буде найлегшою операційною системою IoT. На сьогодні код недоступний, і незрозуміло, чи дійсно ОС буде з відкритим вихідним кодом, отже, віднесемо його в поточну категорію. Крім того, технічні характеристики цієї ОС невідомі, і, зокрема, незрозуміло, як вона пов'язана з ОС з відкритим кодом під назвою LiteOS, яку ми згадували у попередньому розділі.

C. Інше програмне забезпечення

Для повноти ми також узагальнимо в цьому розділі колекцію інших програм, які іноді згадуються як потенційні претенденти, але насправді не є повноцінними ОС або не застосовуються на пристроях класу 1.

1) Arduino: Arduino, що походить із університетського проєкту, є апаратно-програмною компанією з відкритим кодом. У комплекті з IDE, орієнтованою на людей, які не знайомі з програмуванням, це дає змогу легко прототипувати. Хороша підтримка апаратних функцій досягається тим, що Arduino надає як платформи, так і програмне забезпечення. Однак Arduino не забезпечує реального планувальника, підтримки потоків або будь-яких функцій вищого рівня, що робить його придатним насамперед для більш простих додатків.

2) Espruino: Espruino пропонує кілька вбудованих платформ та програмне середовище з відкритим кодом. Програмна частина – це дуже ефективний інтерпретатор для JavaScript, що робить можливим запуск коду JavaScript на обмежених пристроях з обсягом оперативної пам'яті менше 100 кБ. Однак, як і Arduino, Espruino не має на меті замінити повнофункціональну ОС, очевидно має забезпечити сценарій для любителів та виробників. Він не забезпечує базових функціональних можливостей ОС, таких як планувальник або управління потоками. Через природу мови сценаріїв вона, крім того, не здатна виконувати гарантії в режимі реального часу або не надходити на пристроях IoT низького класу.

3) node OS: Node OS – це набір інструментів, повністю написаний на Javascript. Хоча з назви випливає, що це ОС, node OS – це, ймовірніше проміжне програмне забезпечення, ніж сама ОС. Він не працює безпосередньо на апаратному забезпеченні, а працює поверх ядра Linux. Вимоги до Linux, у поєднанні з накладними витратами на Javascript, роблять Node OS непридатною для низькоякісних пристроїв IoT, таких, як пристрої класу 1.

2.6. Різноманіття ОС

Було проаналізовано різні вимоги, які повинна виконувати ОС для низькоякісних пристроїв IoT, які занадто обмежені ресурсами для запуску традиційних операційних систем, таких, як Linux. Розглянуто ключові аспекти такої ОС як з технічного, так і з нетехнічного погляду.

Беручи до уваги ці аспекти, було обстежено доступні ОС, які можуть претендувати на роль ОС для пристроїв IoT.

Здебільшого розглядалися операційні системи з відкритим кодом, оскільки в контексті IoT варто очікувати гострих проблем щодо конфіденційності та безпеки. Таке занепокоєння є великою проблемою, яку легше вирішити за допомогою відкритого вихідного коду, що забезпечує більший потенціал прозорості, надійності та безпеки. Зазначено, що для того, щоб повністю виграти від переваг відкритого коду з позиції надійності, також необхідно використовувати інструменти з відкритим кодом для створення та розгортання двійкових файлів на пристроях IoT (і унеможливити залежність від ненадійних сторонніх серверів / хмарних служб для створення та розгортання цих двійкових файлів). У довгостроковій перспективі спільний характер більшості розробок із відкритим кодом збільшує ймовірність виявлення помилок та покращує потреби МСП. Згідно з нещодавніми дослідженнями, такі компанії будуть стимулювати інновації IoT найближчим часом, але частіше, ніж більші компанії, потребуватимуть розробки програмного забезпечення IoT та розподілу витрат на обслуговування.

У цьому розділі виявили три категорії ОС, в межах яких деякі мають потенціал стати еквівалентом Linux в IoT. Багатопотокові ОС технічно найближчі до Linux, і в цій категорії RIOT наразі є найбільш відомою ОС з відкритим кодом. Орієнтовані на події ОС використовують іншу парадигму програмування, щоб працювати на пристроях з ще меншими ресурсами, і в цій категорії Contiki насьогодні є найбільш відомою ОС з відкритим кодом. RTOS зосереджуються на гарантіях на найменший час виконання та найменший час затримки переривання. У цій категорії FreeRTOS натепер є найбільш відомою ОС з відкритим кодом.

Можна дійти до висновку [8], що існує безліч різних ОС для IoT, це дозволяє користувачам вибрати ОС, яка найкраще відповідає їхнім критеріям. Розглянуто багато компромісів, зроблених дизайнерами систем щодо вимог та обмежень поточних програм IoT та апаратних платформ. Оскільки галузь IoT розвивається стрімкими темпами, остаточне слово ще не сказано щодо того, який тип архітектури та які можливості повинна мати ідеальна ОС для IoT.

ВИСНОВКИ

Розглянуто два аспекти створення сучасних систем Інтернету речей (IoT) – впровадження систем передачі даних з низьким енергоспоживанням та розробка операційних систем, які здатні працювати в умовах обмежених ресурсів та сприяти економії електричної енергії. Вирішення цих задач дасть змогу забезпечити прогнозований ріст систем Інтернету речей. Хоча, звісно, на шляху широкого застосування цих систем є й інші проблеми – побудова когнітивних мереж та систем, вирішення проблеми доступу до хмарних (практично до «туманних») сховищ та обчислень, забезпечення комплексу проблем кібербезпеки та багато інших.

Список використаної літератури

1. Крижановський В.Г. Зв'язок у близькому полі (Near field communication): методичний посібник. Вінниця: ДонНУ імені Василя Стуса, 2018. 32 с.
2. Крижановський В. Г., Сергієнко С. П. До практичної реалізації пристроїв Інтернету речей (IoT): навчально-методичний посібник. Вінниця: ДонНУ імені Василя Стуса, 2019. 48 с.
3. U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, Secondquarter 2017, doi: 10.1109/COMST.2017.2652320.
4. Semtech. [Online]. Available: http://www.semtech.com/
5. A. Hazmi, J. Rinne, and M. Valkama, "Feasibility study of i 802.11ah radio technology for iot and m2m use cases," in 2012 IEEE Globecom Workshops, Dec 2012, pp. 1687–1692.
6. G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, Dec 2015, pp. 181–186.
7. E. Upton and G. Halfacree, Meet the Raspberry Pi. John Wiley & Sons, 2012.
8. Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, Nicolas Tsiftes. Operating Systems for Low-End Devices in the Internet of Things: a Survey. IEEE internet of things journal, IEEE, 2016, 3 (5), pp.720–734.

ЗМІСТ

Навчальне видання

Кржановський *Володимир Григорович*
Сергієнко *Сергій Петрович*

ЕНЕРГОЕФЕКТИВНІ ПРИСТРОЇ ІНТЕРНЕТУ РЕЧЕЙ (IoT)

Навчально-методичний посібник

Редактор І. М. Колесникова

Підписано до друку 02.12.2020
Формат 60 х 84/16. Папір офсетний.
Друк – цифровий. Умовн. друк. арк. 3,72
Тираж 10 прим. Зам. 77.

Донецький національний університет імені Василя Стуса
21021, м. Вінниця, 600-річчя, 21
Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру
серія ДК № 5945 від 15.01.2018