

ДОСЛІДЖЕННЯ СТРУКТУРНИХ ОСОБЛИВОСТЕЙ РОЗПОДІЛЕНИХ ПРОГРАМОВАНИХ БЕЗПРОВІДНИХ СИСТЕМ

У статті досліджуються структурні особливості розподілених програмованих безпроводних систем, які базуються на принципах *Software Defined Radio* з відкритою архітектурою *Software Communication Architecture*.

Вступ

Значне розширення за останнє десятиліття частотної смуги мереж передачі даних привело до появи нового класу безпроводних систем – розподілених програмованих безпроводних систем (РПБС), які базуються на принципах *SDR* (*software defined radio*). Розподілені програмовані безпроводні системи містять декілька обчислювальних вузлів, об'єднаних високошвидкісним інтерфейсом передачі даних [1]. Зазначені системи використовують апаратні засоби для виконання функцій під керуванням програмного забезпечення. Розроблювачі цих засобів покладають великі надії на відкриту архітектуру *SDR* мереж наступного покоління, де обладнання безпроводних мереж можуть випускати одні виробники, а функції й режими роботи буде визначати програмне забезпечення інших розроблювачів.

Розподілені програмовані безпроводні системи дозволяють підвищити якість обслуговування за допомогою високої адаптивності, спрощують розробку програмної частини обладнання й створюють нові можливості при реалізації нових стандартів [2].

У вітчизняній літературі не приділено достатньої уваги структурним властивостям розподілених програмованих безпроводних систем, в результаті виникає завдання їх аналізу.

Метою роботи є аналіз структурних властивостей розподілених програмованих безпроводних систем.

Виклад основного матеріалу

РПБС являє собою центральний процесор (або протокольний процесор, ПП), оснащений прийомними і передавальними блоками. Передавальний блок має комунікаційний процесор (КП), основним завданням якого є модуляція сигналів певної системи зв'язку, які надходять на цифро-аналоговий перетворювач і далі на радіоінтерфейс. Приймний блок містить аналого-цифровий перетворювач, комунікаційний процесор, який здійснює демодуляцію сигналу і перетворення демодульованих символів системи зв'язку в біти даних. Роль центрального процесора полягає в обробці протоколів обміну даними. РПБС можуть бути реалізовані як на обчислювальних засобах загального призначення, так і на сучасних програмованих логічних інтегральних схемах (ПЛІС), що дозволяє створювати радіосистеми на кристалі. При цьому за межі ПЛІС виносяться радіоінтерфейси, а також АЦП і ЦАП. Застосування ПЛІС не призводить до зниження гнучкості системи, оскільки ПЛІС можуть бути в будь-який момент повністю або частково перепрограмовані.

При використанні сучасних ПЛІС стає можливим створення системи на основі принципу *SDR* на одному кристалі (рис. 1). ПЛІС, на відміну від дискретних цифрових сигнальних процесорів, дозволяють створювати безліч різних програмованих блоків обробки на одному кристалі, що в підсумку призводить до збільшення кількості радіоканалів, які одночасно обслуговуються.

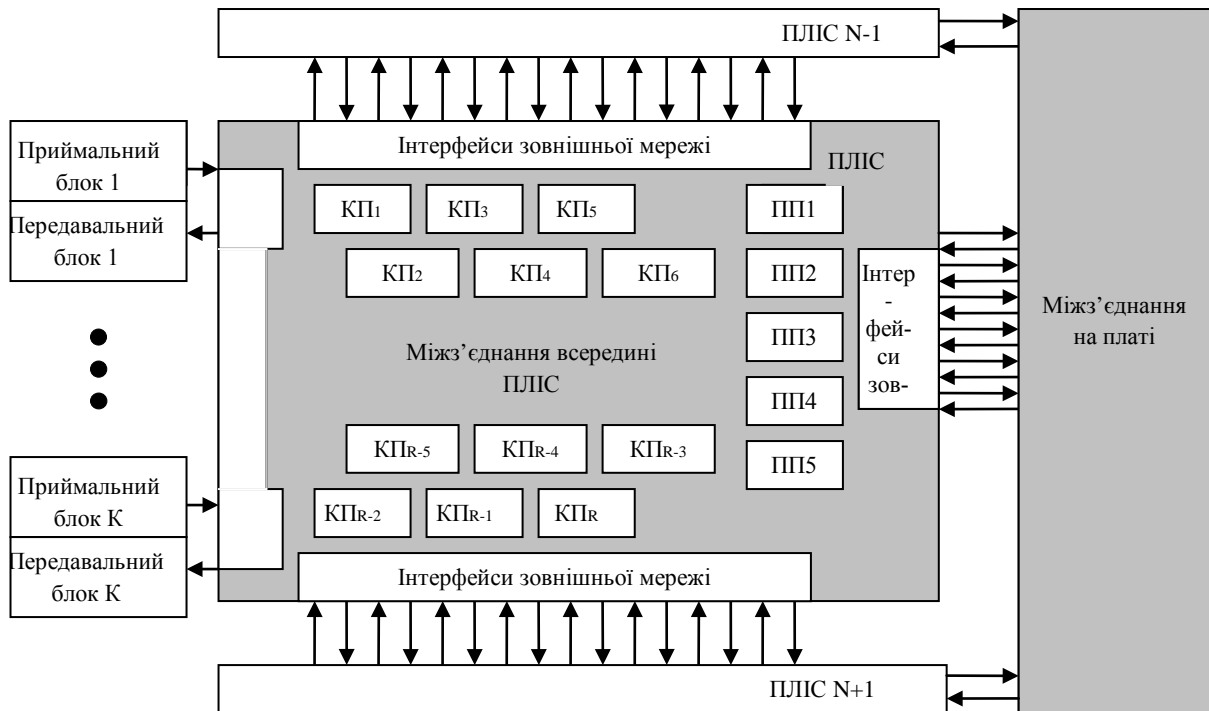


Рис. 1. Функціональна схема РПБС, заснованої на застосуванні ПЛІС

РПБС може складатися з декількох ПЛІС і обслуговувати декілька незалежних радіоканалів. Велика кількість комунікаційних процесорів забезпечує одночасну обробку декількох потоків даних. Самі по собі комунікаційні процесори можуть бути декількох типів, кожен з яких оптимізований для роботи з певним типом сигналів. Також окремі типи процесорів можуть бути виділені для аналізу сигналів, збору статистики або фільтрації пакетів. Можливості перепрограмування (повного або часткового) дозволяють змінювати кількість і склад комунікаційних процесорів в залежності від поточних умов роботи. Використання високошвидкісних сучасних послідовних прийомопередавачів, а також великої кількості паралельних каналів дозволяє поширити структуру між'єднань за межі одного кристала і з малими витратами об'єднувати декілька ПЛІС в систему.

Основним завданням, яке доводиться вирішувати при розробці таких систем, є ефективна і гнучка структура між'єднань. Основними вимогами до систем між'єднань є:

можливість керування швидкістю передачі для кожного зв'язку;

наявність високошвидкісного матричного комутатора; наявність простих мостів для швидкого об'єднання декількох мікросхем в систему обробки даних;

автоматичне виявлення підключення / відключення пристроїв, що необхідно для забезпечення перепланування ресурсів, які використовуються.

Найбільш придатними для реалізації структури між'єднань, що задовольняють цим вимогам, є мережі на кристалі (*Network-on-Chip*). З метою спрощення маршрутизації потоків даних найбільш зручною є пакетна передача даних.

Систему, що складається з обчислювальних *SDR*-вузлів, об'єднаних в прозору мережу (всередині і поза кристалів), можна назвати РПБС. Для реалізації мереж можуть використовуватися *Gigabit Ethernet*, *InfiniBand*, *PCI-Express*, *RapidIO* та інші. Характерною рисою розподілених *SDR*-систем є можливість зміни кількості та складу комунікаційних процесорів при зміні кількості обслуговуваних радіоканалів і / або об'єму оброблюваних даних.

Для створення ефективних розподілених систем обробки даних необхідні адекватні стеки протоколів

обміну даними між елементами системи, які здатні забезпечити передачу цифрових сигналів в режимі реального часу і з прозорістю мережевого місцезнаходження. Також важливою вимогою до протоколів є їх відносна простота, яка повинна дозволити обробляти радіодані не навантажуючи комунікаційні процесори і інші елементи системи зайвими обчисленнями.

Слід зазначити, що специфіка РПБС полягає в тому, що самі по собі комунікаційні процесори можна спростити за рахунок розділення їх по ступенях обробки. При цьому можна досягти зменшення необхідних ресурсів ПЛІС за рахунок того, що деякі ступені обробки можуть використовуватися при обробці різних типів сигналів.

Вибір протоколу транспортного рівня залежить від архітектури системи і вирішуваних завдань. Для передачі в реальному часі з контролем втрат використовуються спеціальні протоколи, такі як *RTP* і *RTCP* [3]. При передачі пакетів сигнальних потоків між мікросхемами вони можуть бути інкапсульовані в протокол каналного рівня, який відповідає обраній шині. При передачі даних усередині мікросхеми з метою спрощення внутрішніх каналів і комутації може використовуватися спрощений транспортний протокол. Перетворення протоколів з внутрішнього в зовнішній і навпаки повинні виконуватися в мостах на основі таблиць перетворення, таких як *ARP*-таблиці в мережах *Ethernet* і програмовані таблиці відповідності внутрішніх і зовнішніх адрес.

Важливою складовою РПБС є розробка переносимого та розширюваного програмного забезпечення. Для уніфікації інтерфейсів програмних компонентів РПБС була розроблена специфікація *SCA* (*Software Communication Architecture*), яка визначає архітектурний каркас розподілених комунікаційних *SDR*-систем. Використання архітектури *SCA* дозволяє уніфікувати інтерфейси для роботи з програмними компонентами та апаратним забезпеченням, завдяки чому програмне забезпечення стає переносимим, а радіосистеми масштабованими [4].

Архітектура *SCA* визначає не тільки способи трансляції радіосигналу через комунікаційну мережу, але й принципи програмного перепрограмування апаратних вузлів (ПЛІС, *DSP*-процесорів), віддаленого налаштування РПБС, управління апаратурою та програмним забезпеченням.

Першою вдалою спробою в області РПБС з'явився проєкт МО США *SPEAKeasy I*, метою якого було створення мультидіапазонної, багаторежимної *SDR*-системи, що дозволяє здійснювати обмін інформацією підрозділам різних військових відомств [4]. Експеримент по перевірці сумісності системи *SPEAKeasy I* був проведений ВПС США в 1993 році. Станція могла працювати у двох різних частотних діапазонах, підтримуючи два різних інтерфейси для зв'язку між звичайними аналоговими радіостанціями *CX*-діапазону (сухопутні сили) і станціями цифрової системи зв'язку з розширенням спектра методом стрибкоподібної зміни частоти (СЗЧ) (тактична авіація). В 1996-1997 роках компанія *Motorola* розробила систему *SPEAKeasy II*, відповідно до якої планувалася розробка радіостанцій, що володіють відкритою архітектурою, і працюють у діапазоні до 2 ГГц [4].

У цей час у США основна програма, що використовує технологію *SDR*, – об'єднана тактична система радіозв'язку *JTRS* (*Joint Tactical Radio System*), яку можна розглядати як розвиток проєкту *SPEAKeasy* [5-7]. Метою програми є випуск сімейства радіостанцій тактичного призначення для передачі мовних повідомлень, відео й даних у діапазоні від 2 МГц до 2 ГГц в умовах прямої і непрямой видимості з використанням одного з 40 різних способів модуляції. Багато із цих видів модуляції використовуються наявними на озброєнні радіостанціями [7].

Роботами по програмі *JTRS* займається “команда комунікацій і електроніки” (*Communications Electronics Command*) армії США під керівництвом Департаменту оборони США, однак всі роботи проводяться на контрактній основі. Так, архітектуру радіостанцій для *JTRS*

розробляє консорціум компаній на чолі з *Raytheon Command, Control, Communications and Information Systems*, що включає також *ITT Industries, Rockwell Collins* і *Marconi CNI Division*.

Система *JTRS* включає наступні програми ведення тактичних дій [7]: *WNW-BEAM, -OFDM, -AJ, -LPI, HAVEQUICK I, II, UHF SATCOM, SINCGARS, LINK-4A, -11B, -16, -22, EPLRS, LMR, DWTS, SATURN, Iridium, Globalstar*, а також підтримує *WLAN*, стільниковий зв'язок різних стандартів (*GSM, CDMA*) і транкінговий зв'язок стандартів *TETRA, TETRAPOL*. Радіозасоби цих програм перекривають частотний діапазон від 2 МГц до 2 ГГц і використовуються для ведення наземних, повітряних і морських операцій. У цей час на стадії розробки з перерахованих програм перебуває тільки *WNW*.

Компанія *Harris* у роботах по проєкті використовує радіостанцію *AN/PRC-117(C)* сімейства *Falcon II*, яка вже знаходиться на озброєнні ВПС США [8]. Основна особливість цієї мультидіапазонної багаторежимної станції – можливість програмного перепрограмування. Компанія бачить пріоритетне завдання в підвищенні каналної ємності своїх радіосистем за рахунок збільшення їхньої спектральної ефективності у високочастотному діапазоні. Для цього досліджуються нові види модуляції й радіоінтерфейси. Компанія *Motorola* на етапі *Step 2* займається верифікацією архітектури *SCA* по програмі програмувальних радіостанцій *DMR (Digital Modular Radio)*, створюваної за окремим контрактом із ВПС США. *DMR* – попередниця *JTRS*, підтримує радіоінтерфейси систем зв'язку *DAMA, HAVE QUICK* і *SINCGARS*, а також обмін даними по *Link 4A* і *Link 11*. Продовженням проєкту *DMR* стала розроблена компанією *Motorola* технологія *WITS (Wireless Information Transfer System)*, відповідно до якої була випущена *SDR-станція AN/USC-61(C)*. Ця радіостанція підтримує більшість тактичних програм і працює в діапазоні частот 2МГц-2ГГц. Як модуляція

використовується *AM, FM, FSK, BPSK, SBPSK, QPSK* [9].

Компанія *Rohde & Schwarz* розробила серію універсальних цифрових багатодіапазонних радіостанцій *M3TR* для тактичної ланки з метою заміни великої кількості різнотипних несумісних між собою засобів *KX* і *УКХ* радіозв'язку для всіх видів збройних сил Німеччини. В 2004 році радіостанції компанії *Rohde & Schwarz M3TR* замінили радіостанції, що стояли на озброєнні швейцарської армії, *SE-229*, одержавши нове умовне найменування *SE-239*. Основна перевага цієї радіостанції – простота зміни режиму роботи за рахунок завантаження відповідного ПЗ. *M3TR* може підтримувати радіоінтерфейси натовських систем *HAVE QUICK I, II, SATURN* і *SECOS*, а також таких стандартів, як *SINCGARS, PR4G* і *SEM*. Радіостанція *M3TR* працює в діапазоні частот 1,5-512 МГц, може підключатися до мереж *TCP/IP*, обмінюватися даними зі швидкістю до 72 Кбіт/с, одночасно передаючи мовні повідомлення [10].

В теперішній час лідером у розробці *SDR-станцій* по програмі *JTRS* є корпорація *Spectrum Signal Processing Inc* (США). Ця корпорація випустила радіостанції, що володіють відкритою архітектурою *SCA* версії 2.2, *SDR-3000* і *SDR-4000*. Ці радіозасоби призначені для передачі мови й даних у діапазоні частот від 0,5 МГц до 3 ГГц із використанням розширення спектра методом СЗЧ зі швидкістю 5000 стрибків/сек для *SDR-3000* і 1000 стрибків/сек для *SDR-4000*. У даних радіозасобах крім підтримки більшості тактичних програм також використовуються протоколи взаємодії з різними мережами мобільного радіозв'язку другого покоління й мережею *WCDMA* третього покоління [11].

Хоча архітектура *SCA* розробляється і просувається Міністерством оборони США з метою створення та впровадження *JTRS* [11], з 2005 р. існує спільнота розробників компонентів *SCA-систем* с відкритим вихідним кодом *OSSIE* [12]. До складу розробленого і розповсюдженого *OSSIE* програмного забезпечення входять інтерфейси і реалізації

ключових абстракцій *SCA*, засоби створення алгоритмів обробки радіоданих вузлами *SCA*, засоби проектування комунікаційних протоколів і відповідних їм схем модуляції / демодуляції, налагоджувальні засоби для імітації та перегляду реалізацій цифрових радіосигналів.

Архітектура *SCA* була розроблена із застосуванням об'єктно-орієнтованого підходу (ООП). Мова *UML (Unified Modeling Language)* – уніфікована мова моделювання) використовувалась для графічного подання інтерфейсів, у той час як самі інтерфейси визначалися мовою *IDL (Interface Definition Language)* – мова опису інтерфейсів). Оба опису (*UML* і *IDL*) можуть бути реалізовані за допомогою стандартних засобів розробки, що дозволяє почати безпосередню розробку з визначення базових концепцій архітектури.

Під архітектурою ПРС розуміється цілісний, взаємопов'язаний і достатній набір функцій, компонентів і правил проектування разом з відповідними інтерфейсами, відповідно до яких можна розробити й реалізувати систему. Конкретна архітектура складається в розподілі функцій між компонентами, таким чином, щоб кожна функція була привласнена якому-небудь компоненту і інтерфейси між компонентами відповідали інтерфейсам між функціями. Якщо такі функції й інтерфейси визначаються шляхом суспільного обговорення, отримана архітектура вважається «відкритою».

В основу архітектури *SCA* покладений стандарт (технологія) *CORBA (Common Object Request Broker Architecture)* – загальна архітектура брокера об'єктних запитів), що визначає загальні принципи побудови розподілених додатків. Стандарт *SCA* базується на об'єктній моделі в рекомендації *X.733ITU-T*, спеціалізованій мові *IDL* і архітектурі керування об'єктами *OMA (Object Management Architecture)*. Використання *CORBA* дозволяє досягти максимальної універсальності модулів *SDR* за рахунок універсальності стандарту й забезпечити незалежність від протоколу

передачі, використовуваної структури даних і апаратної бази.

Відповідно до керівного принципу стандарту *TEMPEST NACSAM 5203* вся множина використовуваного обладнання ділиться на «червоні» і «чорні» підмножини («червону» і «чорну» область). «Червона» область – обладнання, по ланцюгах якого поширюється секретна інформація у відкритому виді. «Чорна» область – обладнання, що обробляє сигнали, які є невизначеними через зміст тексту або через те, що відомості вже зашифровані.

Ключовими перевагами використовуваної архітектури *SDR* є [13]:

1. Максимальне використання комерційних протоколів і продуктів.
2. Ізолювання додатків рівня ядра й додатків нижнього рівня з використанням множинних рівнів відкритої комерційної програмної інфраструктури.
3. Використання середовища розподіленої обробки, що має в основі технологію *CORBA*, що забезпечує програмну переносимість, повторне використання коду й масштабованість.

Архітектура *SCA* визначає операційне середовище (*OE – Operating Environment*) як об'єднання набору сервісів ядра (*Core Framework Services*) і інфраструктурного програмного забезпечення (ПЗ), включаючи проміжне програмне забезпечення (ППЗ). Операційне середовище є інтегрованим компонентом *SCA*.

Структура ПЗ системи, при проектуванні якої була застосована архітектура *SCA*, показана на рис. 2. Операційному середовищу відповідають блоки, зображені нижче логічної шини.

Основними компонентами операційного середовища, згідно [13-16], є:

CORBA Middleware – ППЗ *CORBA*. Насамперед, *ORB (Object Request Broker)* – брокер об'єктних запитів – ядро *CORBA* і будь-якого додатка, написаного з використанням цієї технології, забезпечує пересилання запитів між *CORBA*-об'єктами), а також різні сервіси передбачені специфікацією *CORBA*, як-то сервіс іменування, сервіс подій і т.д. ППЗ

забезпечує роботу додатків, написаних з використанням даної технології. Інтерфейси взаємодії різних компонентів визначаються за допомогою мови *IDL*.

CF Services & Application – сервіси і додатки, що відносяться до інфраструктури ядра. Забезпечують багаторівневий підхід при програмуванні й проектуванні прикладних систем.

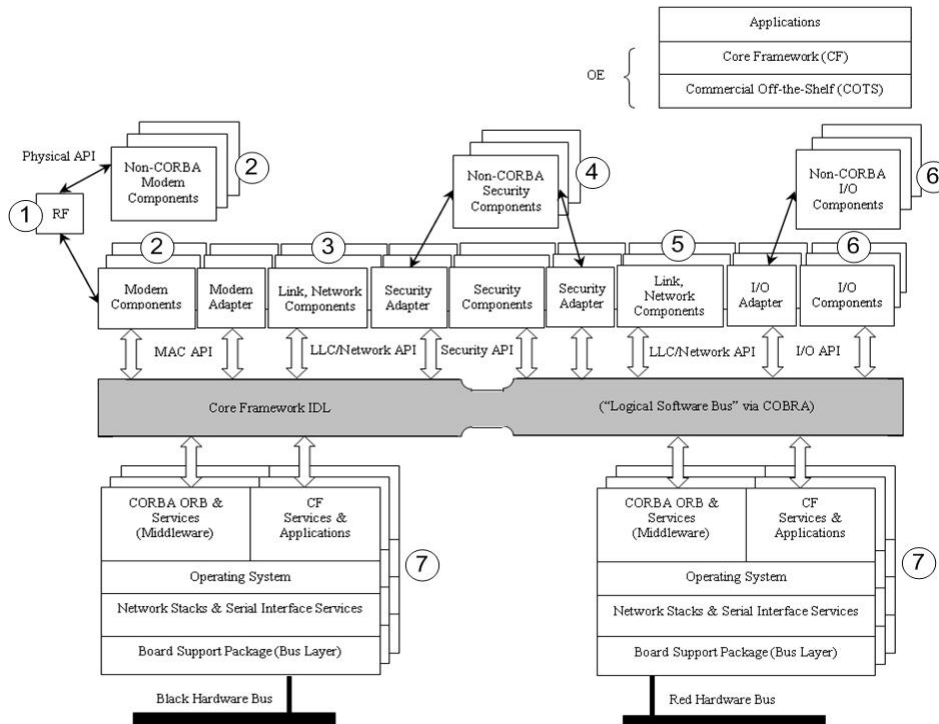


Рис. 2. Структура ПЗ архітектури SCA

Обидва розглянутих компонента функціонують під керівництвом операційної системи – *Operation System*. Як приклад подібної ОС пропонується *POSIX*, через сумісність зі стандартом *CORBA OMG*. Передбачається використання *POSIX 1003.13* – підвид *POSIX*, спеціально розроблений для систем керування реального часу (*PSE52*) (надалі – *SCA POSIX Profile*). Всі інтерфейси взаємодії для даного компонента також визначаються за допомогою мови *IDL*.

Network Stacks & Serial Interface Services – стеки мережних протоколів і послідовні інтерфейси. Забезпечують взаємодію в мережі. Як підтримувані протоколи використовуються *PPP*, *SLIP*, *LAPx*, а також *TCP/IP*. Крім мережних протоколів цей блок відповідає за підтримку різних послідовних інтерфейсів передачі даних: *RS-232*, *RS-422*, *RS-423*, *RS-485*, *Ethernet* і *802.x*.

Board Support Packages – засіб забезпечення кросплатформеності, для

організації апаратного інтерфейсу з конкретним середовищем передачі або системною шиною. У якості таких використовується *VME*, *PCI*, *CompactPCI*, *Firewire (IEEE-1394)*, *Ethernet*. Передбачається, що система з *SCA* архітектурою може функціонувати не тільки в розподіленому виді, але й на різних апаратних платформах.

Розглянуте операційне середовище являє собою базу, на основі якої функціонує основний додаток. Операційне середовище і її компоненти не виконують корисної роботи, однак забезпечують необхідну функціональність для основного рівня. Основним рівнем розглянутої архітектури є рівень додатка – *Application* (на рис. 2 зображений над логічною шиною). Згідно [13-16], на рівні додатка виділяються компоненти (рівні) різної функціональності: 1. *Modem-level digital signal processing* – обробка цифрового сигналу модемом; 2. *Link-level protocol processing* – обробка протоколу з'єднання;

3. *Network-level protocol processing* – обробка протоколу мережі;
4. *Internet routing* – міжмережна маршрутизація;
5. *External input/output (I/O) access* – зовнішній ввід/вивід;
6. *Security* – безпека;
7. *Embedded utilities* – впроваджені програми.

Всі зазначені компоненти додатків при роботі використовують сервіси *CF*.

Взаємодія компонентів додатків безпосередньо з ОС обмежено специфікацією *SCA POSIX Profile*. Взаємодія компонентів додатків з мережею за допомогою ОС не обмежується.

Основні компоненти *SCA*-дodatка показані на рис. 3, числові позначення різних видів функціональності яких використовуються на рис 2, 4-9.



Рис. 3. Рівні з різною функціональністю в додатку *SCA*

Кожний подібний компонент додатка розглядається як такий, що складається із двох частин: безпосередньо компонента (*component*), що виконує прикладне завдання, і адаптера (*adapter*), завданням якого є інтеграція компонента (який може

бути реалізований і без використання технології *CORBA*) у систему, що використовує архітектуру *SCA*. Процес передачі повідомлення в *SCA*-системі показаний на рис. 4.

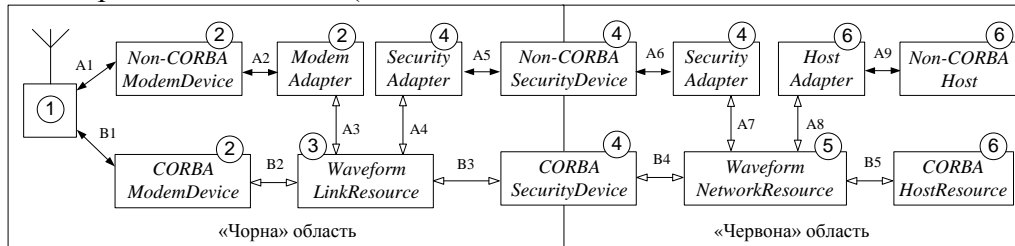


Рис. 4. Передача повідомлення в *SCA*-системі

При передачі повідомлення з використанням винятково ресурсів *CORBA* повідомлення буде проходити по шляху *B1-B5*. При використанні ресурсів, реалізованих з використанням альтернативних технологій, повідомлення в загальному випадку буде передаватися по шляху *A1-A9*.

Компоненти додатка складаються з одного або більше ресурсів (*resource*) [7-10]. Різні ресурси системи визначають режими роботи радіостанції, перемикання між якими здійснюється з використанням принципу програмного керування. Ресурс реалізує інтерфейс керування й конфігурації програмного компонента.

Архітектура *SCA* передбачає використання особливого виду ресурсів – пристроїв (*devices*), які використовуються як програмні представники реальних пристроїв. Розробка внутрішньої функціональності пристроїв не декларується *SCA*, залишаючи рішення за розроблювачем.

На рис. 5 зображені основні класи ресурсів, властивих *SCA* архітектурі (*ModemDevice*, *I/ODevice*, *SecurityDevice*, *NetworkResource*, *LinkResource* і *UtilityResource*), а також показані їхні підкласи, що володіють більш вузькою функціональністю.

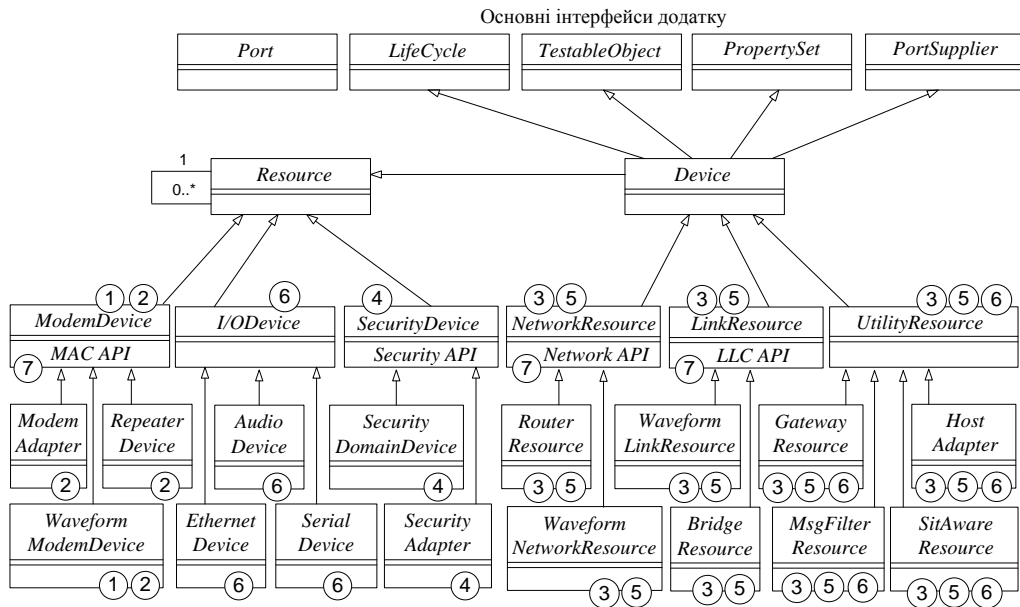


Рис. 5. Ресурси й інтерфейси додатку

Як можна бачити, ресурс може обслуговувати кілька рівнів додатка SCA. Так, ресурс мережі (*NetworkResource*) функціонує на рівнях мережного з'єднання в чорній (3) і червоній (5) області, при цьому реалізує мережний API, який відноситься до рівня операційного середовища (7).

Методи й атрибути, надавані основними інтерфейсами ресурсів, дозволяють організувати їхню взаємодію

в середовищі SCA. Для обміну повідомленнями між ресурсами й пристроями використовується передача повідомлень через порти (*port*).

Більш детально функціональність ресурсів рівня модему (*ModemDevice*), ресурсів вводу/виводу (*I/ODevice*), ресурсів безпеки (*SecurityDevice*) і ресурсів забезпечення мережної взаємодії (*NetworkResource*) показана на рис. 6-9.

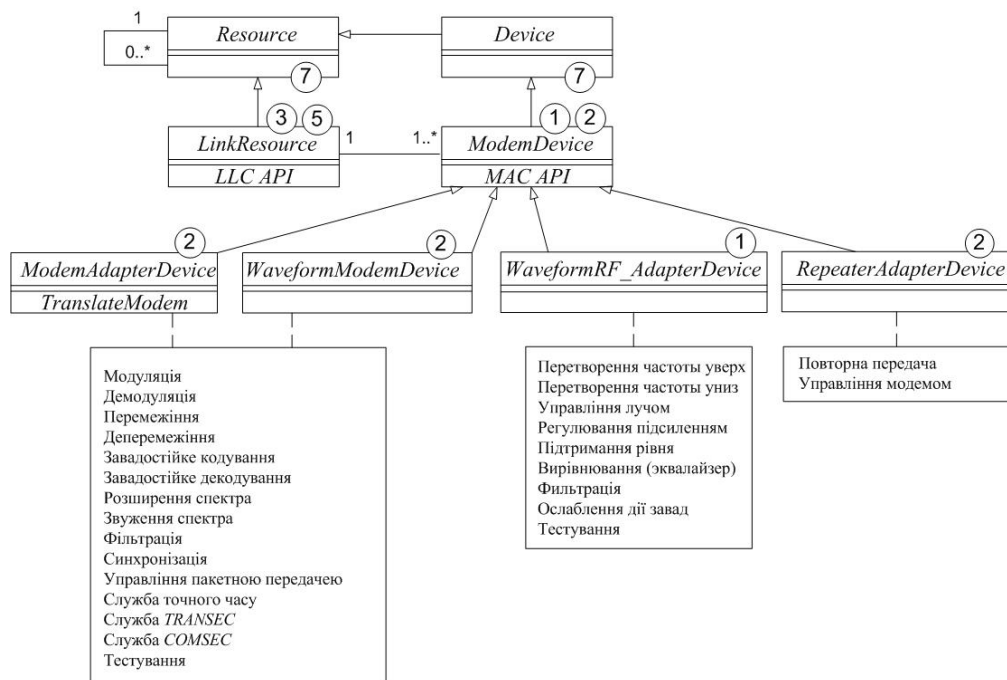


Рис. 6. Функції різних ресурсів модему

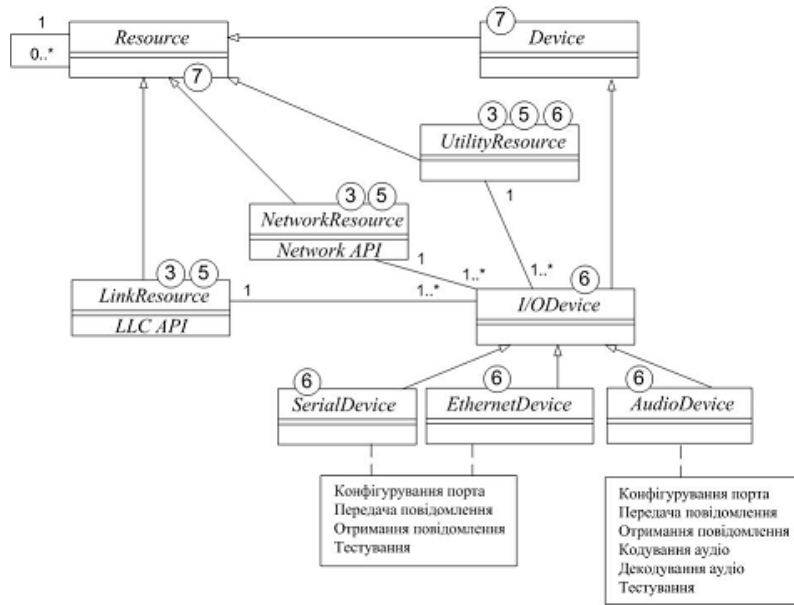


Рис. 7. Функції різних ресурсів забезпечення вводу/виводу

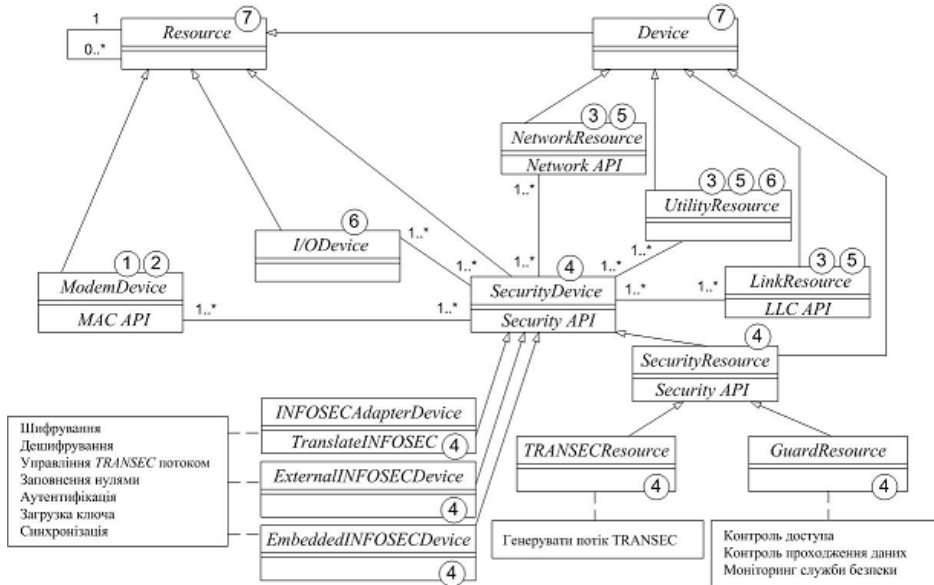


Рис. 8. Функції різних ресурсів забезпечення безпеки

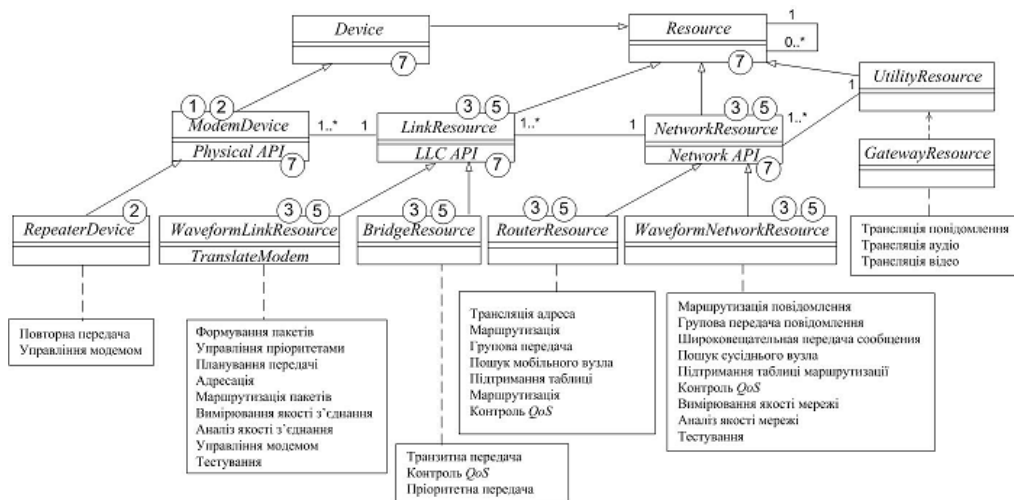


Рис. 9. Функції різних ресурсів забезпечення мережної взаємодії

Функціональність ресурсів SCA описана з використання діаграм UML. У верхній частині прямокутника-ресурсу зазначено його назва, у нижній – реалізована функція. Зв'язок «1..* до 1» вказує на те, що до безлічі ресурсів першого виду ставиться один ресурс другого виду, відповідно «1..* до 1..*» вказує відношення «багато хто до багатьох» і «1 до 0..*» – «один до багатьох» або до порожньої множини. Порожня стрілка вказує на відношення, при якому ресурс, з якого вона виходить, є більше конкретизованим нащадком (у термінах ООП), що інкапсулює частину функціональності батька. API (*Application Programming Interface*) – інтерфейс прикладного програмування, набір функцій, надаваний для використання в прикладних програмах. LLC (*Logical Link Control*) – керування логічним каналом, верхній підрівень канального рівня в семирівневій моделі ISO/OSI.

Ресурс модему (*ModemDevice*) являє собою практично автономний пристрій передачі даних, у зв'язку із чим має більше число функцій, що розділені між декількома підресурсами. Ресурс модему в стандарті SCA крім властивих йому операцій по модуляції/демодуляції також виконує функції завадостійкого кодування/декодування. На додаток до цього модем управляє частотою й шириною смуги, в якій ведеться передача, ослаблює дію перешкод, фільтрацію й т.п. Архітектура SCA крім наявного в її складі криптомодуля (*SecurityDevice*) передбачає наявність додаткових протоколів захисту: *Communication Security – COMSEC* і *Transition Security TRANSEC*. Служба точного часу, необхідна для синхронізації частин розподіленого додатка, запущених на різних вузлах обчислювальної мережі, також функціонує на рівні цього ресурсу.

Найбільш важливими особливостями реалізації РПС по програмі JTRS є використання в їхній структурі сучасних методів модуляції, завадостійкого кодування й технологій розширення спектра.

Ресурси вводу/виводу (*I/ODevice*) служать для організації взаємодії радіозасобів з віддаленими вузлами (використання мережі *Ethernet*, послідовних інтерфейсів і т.п.), а також для безпосереднього вводу інформації (з мікрофона, клавіатури, файлу й т.п.) або її виводу (на гучномовець, файл, печатку й т.п.). Додатковою функцією цього класу ресурсу є організація кодування/декодування аудіо й відео інформації з використанням стандартних методів стиснення.

Ресурси безпеки (*SecurityDevice*) призначені для забезпечення криптозахисту переданих даних. Функції даного ресурсу наступні: шифрування й дешифрування, виконання аутентифікації, операції роботи із ключами й ін.

Ресурси забезпечення мережної взаємодії (*NetworkResource*) призначені для інтеграції різних вузлів SCA системи в єдину комунікаційну мережу. Функції, що виконуються ресурсом цього типу, багато в чому схожі з функціями звичайного мережного маршрутизатора: здійснення маршрутизації, трансляція адрес, підтримка маршрутних таблиць, групові (*multicast*) і широкомовні (*broadcast*) передачі й ін.

Серед додаткових функцій даного ресурсу варто виділити вимір і аналіз якості з'єднання, на підставі якого може прийматися рішення про маршрутизації й виконуватися переконфігурування модему, контроль якості обслуговування (*QoS – Quality of Service*), організація передач потоків даних з різними пріоритетами й ін.

Висновки

Проведений аналіз показує підвищений інтерес у розвинених країнах світу до програмної реалізації функцій обробки сигналів у сучасних безпроводних системах, що визначає необхідність подальшого дослідження архітектури РПС, розробки вітчизняних програмувальних додатків, які могли б використовуватися в даних РПС.

Однією із ключових проблем, що виникають при побудові РПБС, є проблема захисту інформації в розподілених безпроводних системах. Ця проблема складається із двох напрямків: захист програмного забезпечення (програмувальних додатків) і захист оброблюваної інформації (даних).

Напрямок подальших досліджень вважаємо дослідження і оптимізацію фізичного рівня РПБС.

Список літератури

1. Сорохтин Е.М. Распределенные программно-определяемые радиосистемы / Е.М Сорохтин, С.А. Минеев // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2010. – № 5 (2). – С. 383-388.
2. Burns P. Software Defined Radio for 3G / Burns P. – Boston: Artech House, 2003. – 279 p.
3. Jones E. Software Defined Radios, Cognitive Radio and the Software Communications Architecture (SCA) in relation to COMMS, radar and ESM // Cognitive Radio and Software Defined Radios: Technologies and Techniques. – 2008 IET Seminar.
4. Щербак Н. Программируемые радиостанции – будущее тактической связи // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. – 2001. – С. 16-19.
5. Maier K. Mapping waveforms to systems: What would a wideband networking waveform system require? // Military EMBEDDED SYSTEMS. – 2005. – P. 38-41.
6. Uhm M. Adaptivity in Action for SDR and Cognitive Radio // COTS Journal. – 2006. (<http://www.cotsjournalonline.com>).
7. Joint Tactical Radio System (Encyclopedia) // (<http://www.absoluteastronomy.com>).
8. <http://www.rfcomm.harris.com>.
9. <http://www.gdc4s.com>.
10. <http://www.rohde-schwarz.com>.
11. <http://www.spectrumsignal.com>.
12. Dietrich C.B., Reed J.H., Edwards S.H., Kragh F.E. Experiences from the OSSIE Open Source Software Defined Radio Project // Open Source Business Resource. – 2010.
13. Software Communications Architecture Specification // MSRC-5000SCA, V2.2. Contract No. DAAB15-00-3-0001. – 2001. (<http://jtrs.spawar.navy.mil>).
14. Gonzalez A., Hess R. Joint Tactical Radio System. SCA Developer's Guide // Raytheon Company. Contract No. DAAB15-00-3-0001, Document Number: Rev 1.1. – 2002. (<http://jtrs.spawar.navy.mil>).
15. Software Communications Architecture Specification // JPEO JTRS-5000, SCA V3.0. – 2004. (<http://jtrs.spawar.navy.mil>).
16. Software Communications Architecture Specification // JPEO JTRS, Version 2.2. 2. – 2006. (<http://jtrs.spawar.navy.mil>).