

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG

*



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA,
TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG**

Sinh viên thực hiện: **Hoàng Minh Mạnh**

Mã số sinh viên: **20132513**

Lớp: **ĐT-TT 04 - K58**

Giảng viên hướng dẫn: **Ts. Trần Quang Vinh**

Hà Nội, 6/2018

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA,
TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG

Sinh viên thực hiện: **Hoàng Minh Mạnh**

Mã số sinh viên: **20132513**

Lớp: **ĐT-TT 04 - K58**

Giảng viên hướng dẫn: **Ts. Trần Quang Vinh**

Cán bộ phản biện:

Hà Nội, 6/2018

Đánh giá quyển đồ án tốt nghiệp
(Dùng cho giảng viên hướng dẫn)

Giảng viên đánh giá:

TS. Trần Quang Vinh

Họ và tên Sinh viên:

Hoàng Minh Mạnh

MSSV:

20132513

Tên đồ án: **THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA, TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG.**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án.	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (quốc tế và trong nước).	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/ giải quyết vấn đề.	1	2	3	4	5
4	Có kết quả mô phỏng/ thực nghiệm và trình bày rõ ràng kết quả đạt.	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả .(15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống.	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4	5
Kỹ năng viết (10)						

8	Đồ án trình bày đúng biểu mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số đúng thứ tự và được giải thích hay đề cập trong đồ án, có cẩn lè, dấu cách sau chấm, dấu phẩy, v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và trích dẫn theo quy định.	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu văn chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp, v.v).	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5)(chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh sáng chế.					5
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.					2
10c	Không có thành tích nghiên cứu khoa học.					0
Điểm Tổng						.. /50
Điểm tổng quy đổi về thang 10						

Nhận xét thêm của Thầy/Cô (Giảng viên hướng dẫn nhận xét về thái độ và tinh thần làm việc của sinh viên)

.....
.....
.....
.....
.....

Hà Nội, ngày ...tháng ...năm 2018
 Người nhận xét
 (Ký và ghi rõ họ tên)

Đánh giá quyển đồ án tốt nghiệp (Dùng cho cán bộ phản biện)

Giảng viên đánh giá:

Họ và tên Sinh viên: **Hoàng Minh Mạnh**

MSSV: **20138643**

Tên đồ án: **THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA, TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG.**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án.	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (quốc tế và trong nước).	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/ giải quyết vấn đề.	1	2	3	4	5
4	Có kết quả mô phỏng/ thực nghiệm và trình bày rõ ràng kết quả đạt.	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống.	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4	5
Kỹ năng viết (10)						

8	Đồ án trình bày đúng biểu mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số đúng thứ tự và được giải thích hay đề cập trong đồ án, có cẩn lè, dấu cách sau chấm, dấu phẩy, v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và trích dẫn theo quy định.	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu văn chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp, v.v).	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5)(chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh sáng chế.	5				
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.	2				
10c	Không có thành tích nghiên cứu khoa học.	0				
Điểm Tổng						.. /50
Điểm tổng quy đổi về thang 10						

Nhận xét thêm của Thầy/Cô

.....

.....

.....

.....

.....

Hà Nội, ngày ...tháng ...năm 2018

Người nhận xét

(Ký và ghi rõ họ tên)

Lời nói đầu

Trong cuộc các mạng công nghiệp lần thứ tư, các hệ thống nhúng và cơ sở sản xuất thông minh sẽ được kết nối với nhau để tạo ra sự hội tụ kỹ thuật số giữa công nghiệp, kinh doanh, chức năng và quy trình bên trong. Một trong những yếu tố cốt lõi của cách mạng công nghiệp lần thứ tư là Internet Vạn Vật - IoT. IoT là một khái niệm về một viễn cảnh mà các thiết bị ở đó được định danh và có thể chia sẻ, giao tiếp với nhau.

Với số lượng thiết bị vô cùng lớn, để kết nối chúng với nhau thì cần sử dụng một mạng truyền thông không dây. Có rất nhiều công nghệ truyền thông không dây đã được đề xuất như mạng viễn thông, Bluetooth, WiFi, ZigBee,... nhưng chúng đều không đáp ứng được đầy đủ yêu cầu mà IoT đặt ra là truyền dữ liệu khoảng cách xa và tiết kiệm năng lượng. Do đó, nhiều công nghệ truyền thông không dây khác được đề xuất để áp dụng hệ thống IoT như Sigfox, NB-Fi, RPMA và LoRa.

Là một công nghệ mới, công nghệ LoRa cần có những kỹ sư trẻ, nhiệt huyết tìm hiểu, ứng dụng vào thực tiễn đời sống và phát triển. Trước yêu cầu đó, em quyết định chọn đề tài "**THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA, TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG**" nhằm góp phần xây dựng được một mạng cảm biến không dây sử dụng công nghệ LoRa để thu thập dữ liệu về những tham số môi trường cần thiết và đáp ứng được các tiêu chí về năng lượng tiêu thụ, chi phí, khoảng cách truyền dữ liệu.

Qua đây, em cũng xin chân thành cảm ơn thầy giáo **TS. TRẦN QUANG VINH** đã trực tiếp định hướng, tạo điều kiện cần thiết và tận tình hướng dẫn để em có thể hoàn thành đồ án của mình. Em cũng xin chân thành cảm ơn các thành viên trong phòng nghiên cứu Hệ thống mạng và các ứng dụng thông minh

SANSLAB (Smart Applications and Network System Laboratory) đã nhiệt tình hỗ trợ, giúp đỡ em trong quá trình nghiên cứu và thực hiện đồ án.

Trong quá trình làm đồ án, do kiến thức của em còn nhiều hạn chế và hiểu biết chưa rộng nên đồ án không tránh khỏi thêum sót. Em rất mong nhận được sự chỉ bảo và nhận xét quý báu của các thầy cô.

Em xin cam đoan các kết quả được trình bày trong đồ án là công trình nghiên cứu của em dưới sự hướng dẫn của giáo viên hướng dẫn. Các số liệu, kết quả trong đồ án là hoàn toàn trung thực, chưa được công bố trong bất kỳ công trình nào trước đây. Các kết quả được dùng để tham khảo đều được trích dẫn đầy đủ và theo đúng quy định.

Em xin chân thành cảm ơn!

Hà Nội, ngày 30 tháng 5 năm 2018

Tác giả

Hoàng Minh Mạnh

Tóm tắt báo cáo

Internet Vạn Vật - IoT là một kịch bản của cuộc cách mạng công nghiệp 4.0. Khi đó, các thiết bị sẽ được định danh, chúng có thể tương tác với con người và cũng có thể tương tác nhau. Do vậy, chúng ta cần tạo ra và phát triển những công nghệ truyền thông không dây, cũng như các thuật toán phù hợp để giải quyết các vấn đề mà hệ thống IoT đặt ra.

Xuất phát từ những yêu cầu đó, đồ án này tập trung vào việc xây dựng thuật toán đa truy nhập cho mạng sử dụng công nghệ truyền thông LoRa. Từ đó, tiến hành tích hợp vào thiết bị IoT.

Đồ án được trình bày qua các chương như sau:

- CHƯƠNG 1: trình bày tổng quan về đồ án, mục tiêu cùng với các phương pháp nghiên cứu sử dụng trong đồ án.
- CHƯƠNG 2: trình bày những kiến thức cơ bản về công nghệ LoRa, phương pháp đa truy nhập trong mạng không dây, cùng với các thiết bị và công cụ hỗ trợ quá trình thiết kế.
- CHƯƠNG 3: tiến hành các bước trong quá trình thiết kế như: phân tích yêu cầu; thiết kế, kiểm tra và đánh giá thuật toán.
- CHƯƠNG 4: trong phần này, em tiến hành thử nghiệm thực tế, đưa ra kết quả và tiến hành đánh giá.

Abstract

Internet of Things – IoT is a script of the fourth industry revolution (Industry 4.0). At that time, devices will be identified and they will be able to interact with us and other devices. Therefore, we need to design and develop wireless communication technology, as well as suitable algorithms to solve problems imposed by the IoT system.

Based on that requirements, the project focuses on design a multiple access control algorithm for network using LoRa technology, which can be integrated into the IoT device.

The project was showed on chapters:

- Chapter 1: showing overview, target and exhaustive method of the project,
- Chapter 2: including basic knowledge of LoRa communicative technique, medium access control protocol in wireless network and devices and tools for supporting design period,
- Chapter 3: present steps of processing design,
- Chapter 4: presentation and evaluation of experimental result.

Mục lục

Lời nói đầu	v
Tóm tắt báo cáo	vii
Abstract	viii
Mục lục	ix
Danh mục hình vẽ	xi
Danh mục bảng biểu	xiii
Danh mục các từ viết tắt	xiv
Chương 1 Đặt vấn đề	1
1.1 Tổng quan về đề tài	1
1.2 Mục tiêu của đề tài	2
1.3 Các phương pháp sử dụng trong nghiên cứu và thiết kế	3
1.4 Kết luận	3
Chương 2 Cơ sở lý thuyết	4
2.1 Công nghệ LoRa	4
2.1.1 Kỹ thuật điều chế CSS	5
2.1.2 Kiến trúc mạng LoRa	6
2.1.3 Thông số hoạt động chính của LoRa	8
2.2 Đa truy nhập trong mạng không dây	10
2.2.1 ALOHA	11
2.2.2 Giao thức đa truy nhập dựa vào việc kiểm tra kênh truyền	13
2.2.3 Giao thức đa truy nhập tránh va chạm MACA	17
2.3 Module LoRa Ai Thinker SX1278 Ra-02	18
2.4 Vi điều khiển STM32F103	19

2.5	Thiết bị giám sát các tham số môi trường nước BKRES	21
2.6	Một số công cụ hỗ trợ khác	25
2.6.1	Phần mềm hỗ trợ cấu trình vi điều khiển	25
2.6.2	Phần mềm lập trình và biên dịch firmware Keil C	26
2.6.3	Phần mềm hỗ trợ giao tiếp UART Hercules	26
2.7	Kết luận	27
Chương 3 Xây dựng thuật toán đa truy nhập		28
3.1	Phân tích yêu cầu bài toán đa truy nhập	28
3.2	Hệ thống BKRES-LoRa	29
3.2.1	Kiến trúc tổng thể hệ thống	29
3.2.2	Nút mạng cảm biến tích hợp module SX1278	30
3.2.3	Mô hình hoạt động của hệ thống BKRES-LoRa	31
3.3	Xây dựng thuật toán đa truy nhập	32
3.3.1	Thuật toán đa truy nhập của Gateway	32
3.3.2	Thuật toán đa truy nhập của nút	33
3.4	Xây dựng chi tiết và đánh giá	35
3.4.1	Bản tin yêu cầu tham gia mạng	35
3.4.2	Bản tin gửi dữ liệu	37
3.5	Kết luận	38
Chương 4 Kết quả và đánh giá		39
4.1	Các kết quả đạt được	39
4.1.1	Phân tích kết quả các thử nghiệm	39
4.1.2	Hệ thống BKRES-LoRa	41
4.2	Đánh giá thuật toán và định hướng phát triển	42
4.3	Kết luận	43
Kết luận		44
Tài liệu tham khảo		44
Phụ Lục		46

Danh mục hình vẽ

2.1	Nguyên lý điều chế dữ liệu của công nghệ LoRa	6
2.2	Kiến trúc mạng LoRa [4]	6
2.3	Kiến trúc các lớp của thiết bị LoRa [4]	7
2.4	Mối quan hệ giữa tốc độ truyền dữ liệu với các hệ số SF, CR [5] .	8
2.5	Thời gian truyền gói tin trong không khí của LoRa. [5]	9
2.6	Cấu trúc một gói dữ liệu được truyền đi [5].	9
2.7	Các gói tin được truyền trong ALOHA [6]	12
2.8	Hiệu năng của hai giao thức ALOHA và Slotted ALOHA	13
2.9	Lưu đồ thuật toán của hai giao thức 1-persistent CSMA và non-persistent CSMA	14
2.10	Lưu đồ thuật toán giao thức p-persistent CSMA.	15
2.11	So sánh hiệu năng sử dụng kênh truyền của các phương pháp truy nhập kênh truyền ngẫu nhiên [6].	15
2.12	Quá trình truyền tin của 2 trạm sử dụng giao thức CSMA/CA .	16
2.13	Quá trình truyền tin của hai trạm sử dụng giao thức MACA .	17
2.14	Module LoRa Ai Thinker SX1278 Ra-02	19
2.15	Thông số họ vi điều khiển STM32F103xxx [9]	20
2.16	Sơ đồ chân của dòng vi điều khiển STM32F103RCT6 64 chân [9] .	21
2.17	Sơ đồ khối một thiết bị BKRES (client)	23
2.18	Cấu trúc bản tin gửi dữ liệu trao đổi giữa máy chủ và client . . .	24
2.19	Giao diện phần mềm STM32CubeMX hỗ trợ cấu hình	25
2.20	Giao diện cấu hình chi tiết cho từng ngoại vi của STM32CubeMX	25
2.21	Giao diện phần mềm lập trình firmware Keil C μ Vision	26
2.22	Giao diện phần mềm HERCULES	27
3.1	Kiến trúc hệ thống	30
3.2	Nút cảm biến tích hợp chip LoRa SX1278	30
3.3	Mô hình hoạt động của mạng	32

3.4	Luồng trao đổi dữ liệu trong quá trình yêu cầu tham gia mạng	36
3.5	Luồng dữ liệu trong hai trường hợp	38
4.1	Quá trình xử lý của Gateway	39
4.2	Kết quả thí nghiệm đánh giá độ mất gói	40
4.3	Kit điều khiển BKRES-LoRa	41
4.4	Hiển thị dữ liệu trên ứng dụng web	42

Danh mục bảng biểu

2.1	Dòng điện tiêu thụ của module LoRa Ai Thinker SX1278 Ra-02 [7]	18
3.1	Bảng thông số cấu hình của các nút trong mạng	31
3.2	Cấu trúc bản tin yêu cầu tham gia mạng	36
3.3	Cấu trúc bản tin dữ liệu	37
4.1	Bảng kết quả độ mất gói trong điều kiện thuận lợi	40
4.2	Bảng quả độ mất gói trong điều kiện thách thức	41

Danh mục các từ viết tắt

Ký hiệu	Tiếng Anh	Tiếng Việt
IoT	Internet of Thing	Internet vạn vật
WSN	Wireless Sensor Network	Mạng cảm biến không dây
LoRa	Long range	Truyền thông LoRa
FSK	Frequency Shift Key	Điều chế số theo tần số tín hiệu
CSS	Chirp Spread Spectrum	
IES	Inter-symbol Interference	Nhiều giữa các ký tự
SF	Spreading factor	Hệ số trải phổ
BW	Bandwidth	Băng thông
CR	Coding Rate	Tốc độ mã hóa
CRC	Cyclic Redundancy Check	Kiểm tra vòng
MAC	Medium Access Control	Kiểm soát truy nhập kênh truyền
LAN	Local Area Network	Mạng cục bộ
FDM	Frequency Division Multiplexing	Ghép kênh phân chia theo tần số
FDMA	Frequency Division Multiple Access	Đa truy nhập phân chia theo tần số
TDMA	Time Division Multiple Access	Đa truy nhập phân chia theo thời gian
CDMA	Code Division Multiple Access	Đa truy nhập phân chia theo mã
SDMA	Space Division Multiple Access	Đa truy nhập phân chia theo không gian

CSMA	Carrier Sense Multiple Access	Đa truy nhập dựa vào cảm nhận kênh truyền
CSMA-CA	CSMA with Collision Avoidance	Đa truy nhập CSMA thêm cơ chế tránh va đập
RTT	Round Trip Time	Trẽ hai chiều
MACA	Multiple Access with Collision Avoidance	Đa truy nhập có thêm cơ chế tránh va đập
RTS	Ready-to-Send	Bản tin thông báo sắp gửi tin
CTS	Clear-to-Send	Bản tin thông báo sẵn sàng nhận tin
SPI	Serial Peripheral Interface	Giao tiếp nối tiếp ngoại vi
USB	Universal Serial Bus	Chuẩn kết nối USB
GSM	Global System for Mobile Communications	Hệ thống toàn cầu dành cho việc liên lạc di động
AES	Advanced Encryption Standard	Tiêu chuẩn mã hóa tiên tiến
TCP-TP	Transmission Control Protocol/Internet Protocol	Giao thức kiểm soát truyền tải dùng cho mạng Internet
ID	Identify	Định danh
IC	Intergrated Circuit	Vi mạch tích hợp
IMEI	International Mobile Equipment Identity	Số nhận dạng thiết bị di động

Chương 1

Đặt vấn đề

Trong chương 1, em xin trình bày khái quát về đề tài, mục tiêu mà đề tài hướng đến cũng như các phương pháp nghiên cứu được sử dụng để hoàn thành được đề tài [?].

1.1 Tổng quan về đề tài

Đề tài “THIẾT KẾ, PHÁT TRIỂN GIAO THỨC ĐA TRUY NHẬP LORA TÍCH HỢP VÀO MẠNG WSN GIÁM SÁT MÔI TRƯỜNG” là đề tài nghiên cứu và tích hợp công nghệ LoRa vào thiết bị IoT. Trong cuộc cách mạng công nghiệp 4.0, IoT là một trong các xu thế, nó không chỉ được áp dụng trong các mô hình sản xuất mà còn được ứng dụng để phục vụ cuộc sống hằng ngày con người như nhà ở thông minh (Smart Home), thành phố thông minh (Smart City), giao thông thông minh (Smart Transport),...

LoRa (Long Range) là kỹ thuật điều chế dựa trên kỹ thuật trải phổ, có phạm vi hoạt động lớn hơn rất nhiều so với các kỹ thuật cạnh tranh khác. Công nghệ truyền thông không dây LoRa được phát triển bởi Cycleo SAS, sau đó được mua lại bởi Semtech. Theo tài liệu [1], công nghệ LoRa đã được xuất hiện ở nhiều quốc gia trên thế giới. Ở châu Âu, nhiều dự án được triển khai trên toàn lãnh thổ hoặc được thí điểm trên khu vực rộng lớn. Một số dự án ở châu Âu như:

- KPN (Hà Lan), nó là dự án đầu tiên hoàn thành việc sử dụng mạng lưới LoRa (phổ sóng cả nước) cho thiết bị IoT,

- Orange (Pháp), trong quý đầu năm 2016, dự án đã bắt đầu triển khai ở 18 khu vực đô thị và được kỳ vọng sẽ triển khai trên khắp cả nước,
- Unidata, một công ty viễn thông của Ý đã ra mắt mạng LoRaWAN tại Rome và sớm sẽ triển khai trên toàn quốc.

Tại Mỹ, hai thành phố Philadelphia và San Francisco đã được chọn cho việc thử nghiệm các ứng dụng IoT như theo dõi tài sản và giám sát môi trường. Vào ngày 20/12 năm 2016, Semtech đã triển khai một mạng LoRaWAN tại New Zealand và hứa rằng một nửa dân số New Zealand có thể được sử dụng trong một tháng. Ở các khu vực khác trên thế giới như châu Á - Thái Bình Dương và châu Phi, công nghệ LoRa cũng bắt đầu được ứng dụng. Tại Việt Nam, công nghệ LoRa đang trong quá trình nghiên cứu để có thể áp dụng vào các dự án thực tế.

Trong mạng cảm biến không dây WSN (Wireless Sensor Network), các nút trong mạng sẽ có nhiệm vụ thu thập dữ liệu môi trường xung quanh qua các cảm biến. Những dữ liệu đó sẽ được gửi đến Gateway qua các phương pháp truyền thông không dây. Sau đó, chúng được gửi lên máy chủ để tổng hợp, phân tích và xử lý. Từ những dữ liệu đã được xử lý này, con người sẽ đưa ra các biện pháp để ứng phó hoặc chính những nút này cũng có thể đưa ra các biện pháp.

1.2 Mục tiêu của đề tài

Mục tiêu của đề tài này là thiết kế, phát triển giao thức đa truy nhập cho mạng sử dụng module truyền thông LoRa và tích hợp giao thức truy nhập vào thiết bị BKRES, là các thiết bị có kích thước nhỏ, yêu cầu tiêu thụ năng lượng thấp. Dựa vào những mục tiêu trên, những yêu cầu cụ thể được đưa ra như sau:

- Truyền dữ liệu 1 – 1 giữa nút và Gateway,
- Truyền dữ liệu từ nhiều nút đến Gateway (n-1),
- Các nút có khả năng tự động tham gia mạng,
- Truyền nhận dữ liệu chính xác, ổn định,
- Tìm ra được các nguyên nhân gây mất dữ liệu và khắc phục,
- Tích hợp thuật toán đa truy nhập vào thiết bị IoT (BKRES).

1.3 Các phương pháp sử dụng trong nghiên cứu và thiết kế

Các phương pháp mà em sử dụng gồm có:

- Tham khảo tài liệu: tham khảo tài liệu từ sách báo về điện tử, từ mạng internet, kế thừa và phát triển các kết quả nghiên cứu đã có,
- Quan sát, học hỏi: thảo luận cùng thầy và các bạn để đưa ra được hướng đi tốt nhất, đạt hiệu quả cao,
- Thực hành và sửa lỗi: xây dựng thuật toán thành các phần sau đó thí nghiệm nhiều lần và tìm hiểu, thảo luận để đạt được kết quả tối ưu nhất.

1.4 Kết luận

Mục tiêu của đề tài này là thiết kế, phát triển giao thức đa truy nhập cho mạng sử dụng module truyền thông LoRa và tích hợp giao thức đa truy nhập vào thiết bị BKRES. Để hoàn thành đề tài cần có những kiến thức về công nghệ LoRa, đa truy nhập trong mạng không dây cùng với những hiểu biết về hệ thống BKRES. Những kiến thức này sẽ được trình bày trong Chương 2.

Chương 2

Cơ sở lý thuyết

Chương này em sẽ trình bày những kiến thức cần có để xây dựng được một mạng hình sao sử dụng công nghệ truyền thông LoRa. Đây là những kiến thức cơ bản về LoRa, đa truy nhập trong mạng không dây, các thiết bị phần cứng được sử dụng cũng như một số công cụ phần mềm hỗ trợ thiết kế và lập trình.

2.1 Công nghệ LoRa

LoRaTM (Long Range Radio) [2] là một công nghệ truyền thông không dây có phạm vi hoạt động lớn, tiêu tốn ít năng lượng và sử dụng dải tần miễn phí (unlicensed radio spectrum), dải tần này được sử dụng trong công nghiệp, khoa học và y tế (ISM band). LoRa được phát triển bởi Cycleo SAS (sau này được mua lại bởi SemTech). Mục đích tạo ra công nghệ LoRa là nhằm loại bỏ repeater, giảm chi phí đầu tư vào thiết bị, tăng cường thời gian hoạt động, tăng dung lượng mạng và hỗ trợ cho mạng có số thiết bị lớn. Công nghệ LoRa nằm lớp vật lý, được sử dụng cho truyền thông phạm vi lớn. Hầu hết công nghệ không dây sử dụng kỹ thuật điều chế số theo tần số tín hiệu FSK (Frequency Shift Key). Tuy nhiên, LoRa lại sử dụng kỹ thuật điều chế Chirp Spread Spectrum (CSS) để tiết kiệm năng lượng nhằm tăng khoảng cách truyền. LoRa là công nghệ đầu tiên sử dụng kỹ thuật điều chế CSS được thương mại hóa. Trước đó, kỹ thuật điều chế CSS chỉ được dùng trong quân sự và hàng không vũ trụ. Nhờ những ưu điểm về năng lượng và tầm hoạt động, LoRa phù hợp để ứng dụng trong mạng WSN cũng như các thiết bị IoT.

2.1.1 Kỹ thuật điều chế CSS

Theo tài liệu [3], LoRa dựa trên kỹ thuật điều chế Chirp Spread Spectrum (CSS) (Hình 2.1). CSS được xem là một kỹ thuật Direct-Sequence Spread Spectrum. CSS phù hợp với nhu cầu mạng IoT bởi vì nó cho phép chống nhiễu tốt, phạm vi sử dụng lớn và tiêu thụ ít năng lượng. Trong kỹ thuật trai phổ CSS, tín hiệu chirp được mô tả bằng hàm pha tức thời $\phi(t)$ hoặc một hàm thời gian đặc biệt $f_c(t)$. $f_c(t)$ được gọi là hàm tín hiệu chirp chưa được xử lý (the raw chirp) do:

- Khi tăng tần số, một tín hiệu up-chirp bắt đầu từ $-\frac{B}{2}$ đến $\frac{B}{2}$,
- Khi giảm tần số, một tín hiệu down-chirp bắt đầu từ $\frac{B}{2}$ đến $-\frac{B}{2}$,

trong đó, B là băng thông của dải tần ISM được sử dụng. Hàm $f_c(t)$ được xác định bởi công thức:

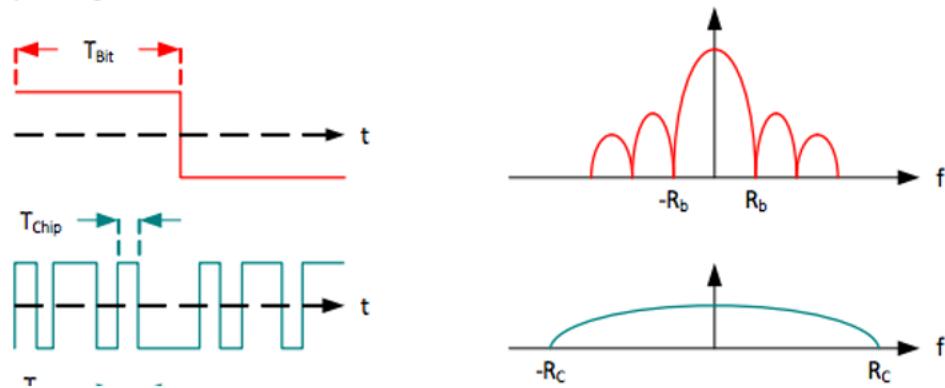
$$f_c(t) = \pm \frac{B}{T_s} t \quad (2.1)$$

trong đó T_s là chu kỳ ký hiệu (symbol period). Mối quan hệ giữa băng thông và chu kỳ ký hiệu được thể hiện trong công thức $T_s = \frac{2^{SF}}{B}$ trong đó SF là hệ số trai phổ nằm trong khoảng [7,..., 12]. D_s là tỷ lệ ký hiệu trong tín hiệu được truyền và D_b là tỷ lệ bit, $D_s = \frac{D_b}{SF}$.

Trong LoRa, một ký hiệu được tạo từ SF bit nhị phân. Mỗi ký hiệu được mô tả bởi một chirp duy nhất. Các chirp khác nhau trực giao với nhau để khi được khôi phục ở bên nhận các ký hiệu tránh được nhiễu (Inter-symbol interference). Nếu chúng ta có bộ M ký hiệu, ký hiệu m , $m \in [0, M - 1]$, thu được bởi chế raw chirp $f_c(t)$ và $\tau_m = \frac{m}{B}$. Những chirp nằm bên ngoài khoảng $[-\frac{T_s}{2}, \frac{T_s}{2}]$ được chuyển dịch chu kỳ $[-\frac{T_s}{2}, -\frac{T_s}{2} + \tau_m]$. Do đó, chirp liên quan đến việc truyền ký tự thứ m được chia làm hai phần:

- Nếu $t \in [-\frac{T_s}{2}, -\frac{T_s}{2} + \tau_m]$, raw chirp nhỏ hơn $(T_s - \tau_m)$,
- Nếu $t \in [-\frac{T_s}{2} + \tau_m, \frac{T_s}{2}]$, raw chirp lớn hơn τ_m .

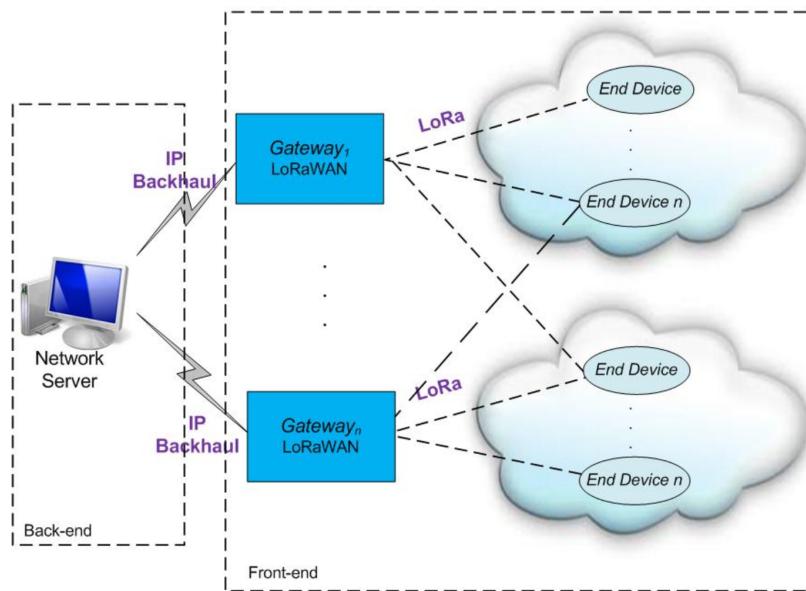
Nhờ sử dụng tín hiệu chirp mà các tín hiệu LoRa với các tỷ lệ chirp khác nhau có thể hoạt động trong cùng 1 khu vực mà không gây nhiễu cho nhau. Điều này cho phép nhiều thiết bị LoRa có thể trao đổi dữ liệu trên nhiều kênh đồng thời (mỗi kênh cho 1 chirprate).



Hình 2.1: Nguyên lý điều chế dữ liệu của công nghệ LoRa

2.1.2 Kiến trúc mạng LoRa

Một mạng sử dụng công nghệ truyền thông LoRa [4] sẽ được triển khai theo cấu trúc hình sao và gồm hai phần là back-end và front-end (Hình 2.2). Phần back-end là máy chủ mạng chứa thông tin nhận được từ các cảm biến. Còn phần front-end chứa module Gateway và các nút (end-device nodes). Gateway có nhiệm vụ làm cầu nối giữa các nút và máy chủ. Thông tin mà Gateway nhận được sẽ được gửi đến máy chủ qua kết nối IP.



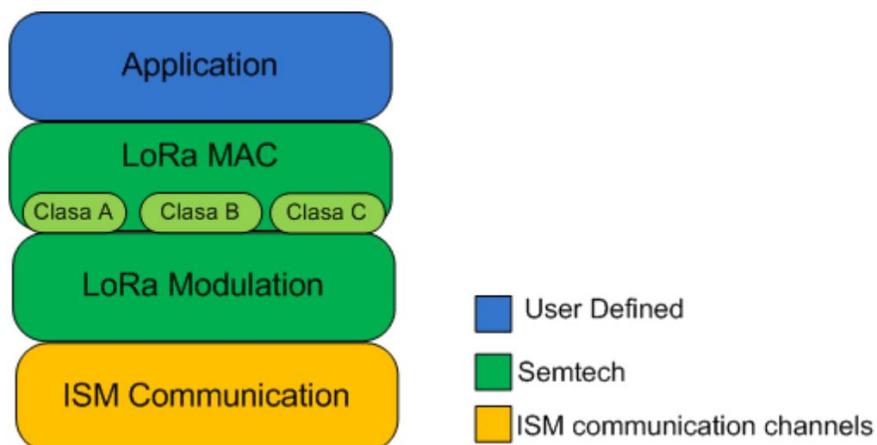
Hình 2.2: Kiến trúc mạng LoRa [4]

LoRa hoạt động dải tần ISM [4], tuy nhiên ở mỗi nước (khu vực) thì dải tần này lại khác nhau. Cụ thể là:

- Ở Châu Âu, dải tần ISM là: 433MHz hoặc 863-870MHz, gồm có 3 kênh (tương ứng với mỗi dải tần): 433.175 MHz, 433.375 MHz, 433.575 MHz hoặc 868.10 MHz, 868.30 MHz, 868.50 MHz,
- Dải tần ISM ở Mỹ là: 902-928MHz,
- Dải tần ISM ở Trung Quốc là: 779-787MHz,
- Ở Úc, dải tần ISM là: 915-928MHz.

Dựa vào cơ chế truyền thông, các nút LoRa (end-device) được chia thành 3 loại [4] gồm:

- Class A (Baseline): Khung truyền được chia thành uplink và downlink, bao gồm 1 khe thời gian uplink, tiếp theo là 2 khe thời gian downlink. Thiết bị loại này chỉ có thể nhận dữ liệu từ máy chủ sau khi nó đã gửi dữ liệu,
- Class B (Beacon): Thiết bị đầu cuối mở thêm một khe nhận trong khoảng thời gian downlink, cùng thêm hai khe thời gian được chỉ định trong loại A. Ngoài khả năng nhận dữ liệu như các thiết bị loại A, các thiết bị loại B cũng có thể sử dụng một loạt khe nhận được kích hoạt bởi các bản tin kiểu Beacon được gửi từ gateway,
- Class C (Continuous): Các thiết bị loại này có thể nhận dữ liệu bất kể khi nào trừ lúc truyền dữ liệu.

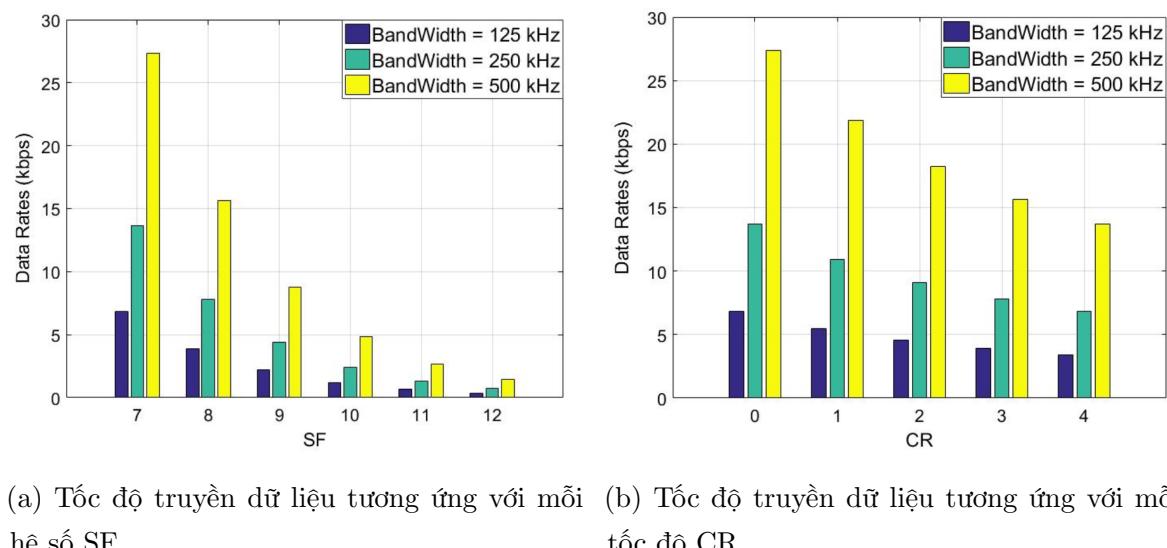


Hình 2.3: Kiến trúc các lớp của thiết bị LoRa [4]

Thiết bị loại A là đơn giản nhất và đảm bảo tiêu thụ năng lượng tối thiểu nhưng lại bị hạn chế khả năng nhận tin nhắn. Nên theo tiêu chuẩn LoRa, các thiết bị phải thực hiện tối thiểu được cơ chế truyền thông loại A. Hình 2.3 sẽ cho ta thấy rõ hơn cấu trúc các lớp của một thiết bị LoRa.

2.1.3 Thông số hoạt động chính của LoRa

Đối với chipset SX1278 của Semtech, hệ số trải phổ SF (Spreading Factor) có giá trị từ 7 đến 12 [5]. Hệ số SF thể hiện mối quan hệ giữa tốc độ truyền dữ liệu và khoảng cách truyền. Lựa chọn hệ số SF cao có thể làm tăng phạm vi hoạt động nhưng giảm tốc độ truyền dữ liệu và ngược lại. Mỗi ký hiệu được mã hóa bởi một mā có độ dài 2^{SF} chip. Do việc mã hóa một ký hiệu bằng nhiều tín hiệu chip, điều này làm ảnh hưởng đến tốc độ truyền dữ liệu. Mối quan hệ giữa tốc độ truyền dữ liệu và hệ số SF được thể hiện trong Hình 2.4a tại các băng thông khác nhau.

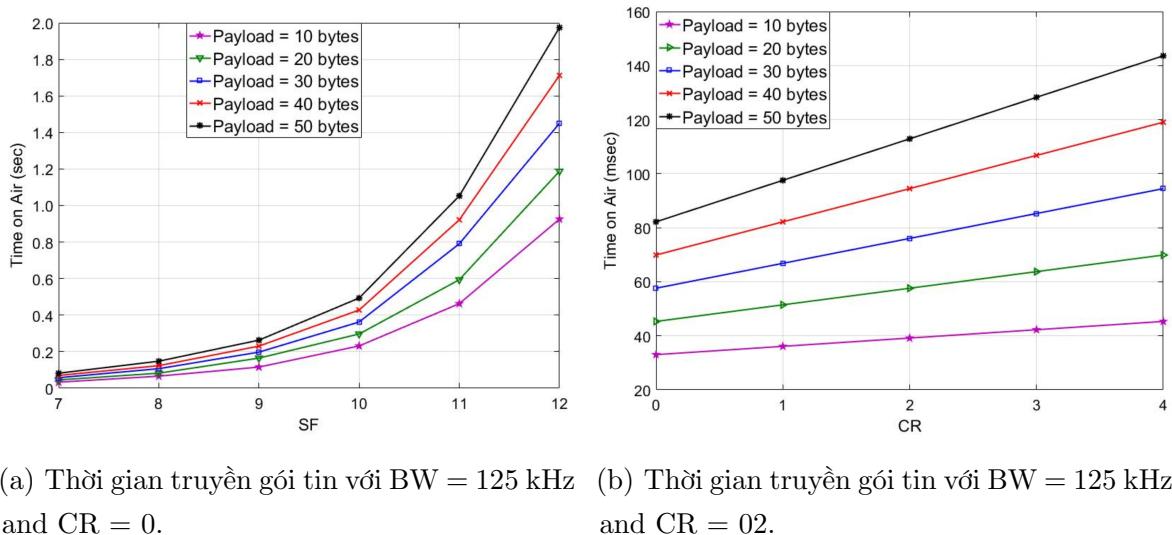


(a) Tốc độ truyền dữ liệu tương ứng với mỗi hệ số SF.
(b) Tốc độ truyền dữ liệu tương ứng với mỗi tốc độ CR.

Hình 2.4: Mối quan hệ giữa tốc độ truyền dữ liệu với các hệ số SF, CR [5]

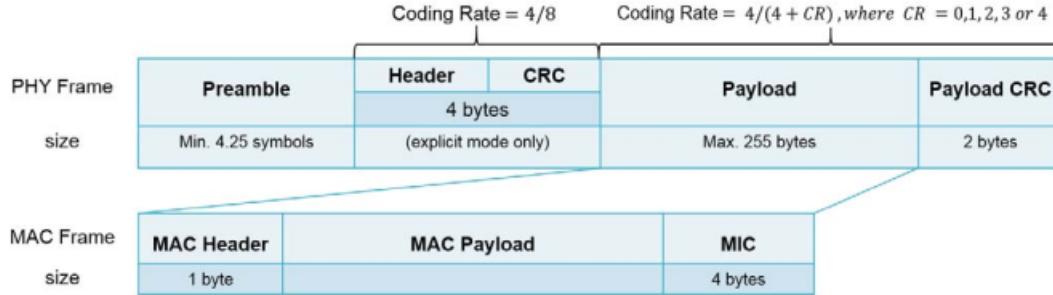
Theo tài liệu [5], LoRa thường sử dụng băng thông BW Bandwidth) có giá trị là 125 KHz nhưng cũng có thể mở rộng lên thành 250 KHz và 500 KHz để truyền tín hiệu. Với băng thông có độ rộng lớn, LoRa hạn chế được nhiều kênh truyền, hiệu ứng doppler và fading. Máy phát gửi dữ liệu với tốc độ chip bằng với băng thông nên với băng thông là 125 kHz thì tốc độ chip là 125 kcps.

Tốc độ mã hóa CR (Coding Rate) của chipset SX1278 của SemTech thì CR có 4 giá trị là $4/5$, $4/6$, $4/7$ và $4/8$. Nếu tốc độ mã hóa là k/n , thì k là số bit có ích (chứa thông tin được mã hóa) và n là số bit ở đầu ra. Do đó $n - k$ là số bit dự phòng, những bit này cho phép người nhận phát hiện và sửa lỗi trong bản tin nhưng nó cũng làm giảm tốc độ truyền dữ liệu. Tốc độ truyền dữ liệu tương ứng với mỗi CR được thể hiện trong Hình 2.4b ở mỗi băng thông khác nhau. Ngoài ra, tốc độ truyền dữ liệu cũng phụ thuộc vào độ dài gói tin (Hình 2.5).



(a) Thời gian truyền gói tin với BW = 125 kHz and CR = 0.
(b) Thời gian truyền gói tin với BW = 125 kHz and CR = 02.

Hình 2.5: Thời gian truyền gói tin trong không khí của LoRa. [5]



Hình 2.6: Cấu trúc một gói dữ liệu được truyền đi [5].

Kích thước lớn nhất của một gói tin LoRa là 256 byte, cấu trúc một gói tin (Hình 2.6) gồm các thành phần sau:

- Preamble: là trường chuỗi bit để dò tìm tín hiệu của LoRa trong không gian,

- Header: chứa thông tin về kích thước của Payload và xem có Payload CRC hay không. Giá trị của Header cũng được check CRC kèm theo,
- Payload: chứa dữ liệu ứng dụng truyền qua LoRa, có kích thước từ 2 đến 255 byte. Trường này còn chứa các trường sau:
 - MAC header: phụ thuộc vào kiểu của frame (là dữ liệu hoặc ACK), phiên bản của giao thức và hướng của nó (uplink hoặc downlink).
 - MAC payload: chứa dữ liệu.
 - MIC: được sử dụng như là chữ ký số của dữ liệu.
- CRC: là trường tùy chọn và chứa các byte kiểm tra vòng CRC (Cyclic Redundancy Check).

2.2 Đa truy nhập trong mạng không dây

Đa truy nhập (Multiple Access) là một kỹ thuật cho phép nhiều thiết bị sử dụng chung một kênh truyền. Kỹ thuật này nằm ở lớp MAC (Medium Access Control), đây là một lớp đặc biệt quan trọng trong những mạng LAN (Local Area Network), trong đó có mạng không dây bởi vì các thiết bị trong mạng này sử dụng chung môi trường truyền.

Giả sử có nhiều thiết bị đang sử dụng kênh truyền, trong cùng một khoảng thời gian với cùng tần số, khi đó sẽ xảy ra hiện tượng va chạm. Do đó, cần phải có một biện pháp kiểm soát các thiết bị tham gia kênh truyền. Ví dụ như một mạng có băng thông là B , có N thiết bị đang tham gia kênh truyền, mạng sử dụng phương pháp ghép kênh phân chia theo tần số FDM (Frequency Division Multiplexing), khi đó băng thông được chia thành N dải tần, mỗi thiết bị sẽ sử dụng một dải tần. Tuy nhiên, với số lượng thiết bị lớn, thì những dải tần này bị chia quá nhỏ, do đó dễ gây nên can nhiễu. Do đó, cần có thêm những phương pháp kiểm soát kênh truyền. Đây là 4 phương pháp điều khiển truy nhập kênh truyền:

- Đa truy nhập phân chia theo tần số FDMA (Frequency Division Multiple Access): dải tần số sẽ được chia thành các dải tần số nhỏ không chồng lên nhau, và mỗi dải đó được sử dụng bởi một thiết bị khác nhau,

- Đa truy nhập phân chia theo thời gian TDMA (Time Division Multiple Access): mỗi thiết bị trong mạng sử dụng chung một dải tần nhưng chỉ được phép truyền trong một khoảng thời gian cụ thể,
- Đa truy nhập phân chia theo mã CDMA (Code Division Multiple Access): các thiết bị trong mạng sẽ sử dụng chung một dải tần và có thể gửi bản tin cùng một thời gian nhưng những tín hiệu đó sẽ được mã hóa bằng các mã ngẫu nhiên khác nhau,
- Đa truy nhập phân chia theo không gian SDMA (Space Division Multiple Access): các thiết bị trong mạng sẽ sử dụng chung một dải tần, truyền/nhận bản tin cùng một lúc và không sử dụng mã ngẫu nhiên để mã hóa. Tại trạm cơ sở, yêu cầu phải có một dãy ăng-ten lớn để khử nhiễu.

Từ những phương pháp điều khiển kênh truyền trên, các giao thức đa truy nhập đã được phát triển.

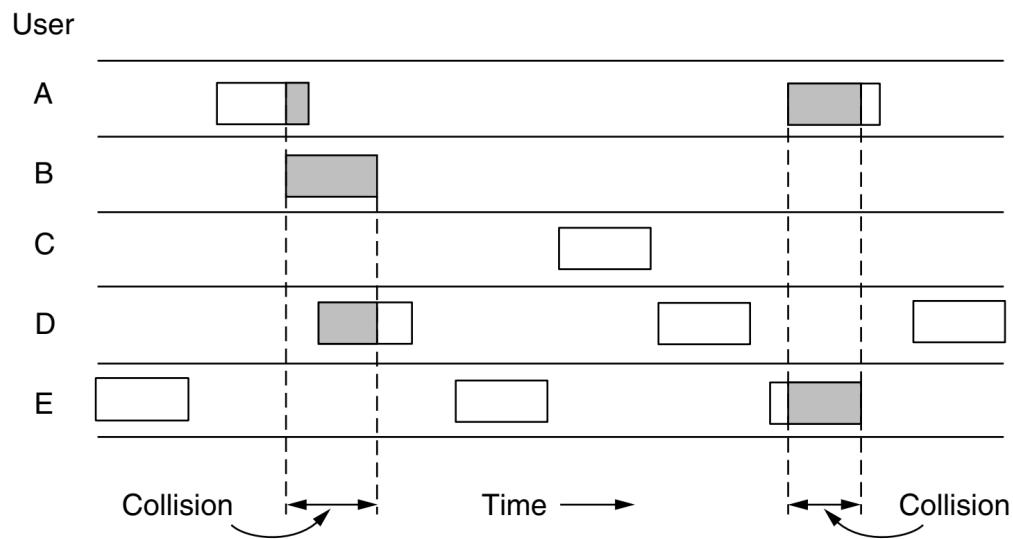
2.2.1 ALOHA

Theo tài liệu [6], giao thức đa truy nhập đầu tiên bắt đầu một cách sơ khai ở Hawaii vào đầu những năm 1970, do Norman Abramson đề xuất. Gọi là "sơ khai" bởi vì nó không được sử dụng cho mạng di động. Mục đích của Norman Abramson và các cộng sự khi tạo ra mạng này để kết nối những thiết bị ở những hòn đảo xa tới máy chủ ở Honolulu. Sau đó, công trình của ông, hệ thống pure ALOHA (ALOHA nguyên thủy), đã thu hút sự chú ý của rất nhiều nhà nghiên cứu và nó đã được phát triển thêm thành giao thức slotted ALOHA.

Pure ALOHA

Ý tưởng của hệ thống ALOHA là rất đơn giản: người sử dụng sẽ truyền bản tin bất khi nào họ có dữ liệu. Do đó, va chạm trong ALOHA là điều dễ hiểu. Trong hệ thống ALOHA, sau khi trạm đã gửi bản tin của nó đến máy chủ, máy chủ sẽ phát quang bá bản tin đó đến tất cả các trạm. Trạm gửi bản tin có thể biết được máy chủ đã nhận được bản tin của nó hoặc bản tin được gửi đã đúng chưa qua việc nhận lại bản tin từ máy chủ. Nếu việc gửi không thành công, bên gửi sẽ chờ một khoảng thời gian ngẫu nhiên nào đó rồi gửi lại. Thời gian chờ này

phải ngẫu nhiên để tránh việc va chạm với bản tin quay trở lại từ máy chủ. Khi hệ thống có nhiều thiết bị cùng chia sẻ một kênh, điều đó dẫn đến việc xung đột kênh truyền. Hình 2.7 mô tả quá trình gửi bản tin một cách tùy ý trong hệ thống ALOHA.



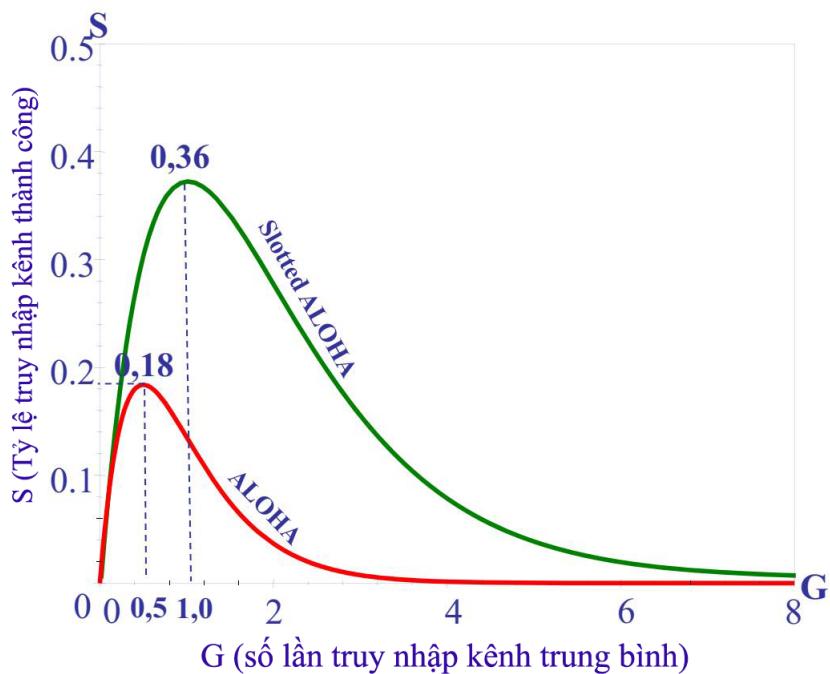
Hình 2.7: Các gói tin được truyền trong ALOHA [6]

Do không có cách nào để kiểm soát xung đột trong ALOHA, nên hiệu năng của giao thức này khá thấp như thể hiện trong Hình 2.8.

Slotted Aloha

Theo tài liệu [6], ngay sau ALOHA, vào năm 1972, Roberts đã công bố một phương pháp giúp tăng gấp đôi hiệu năng của hệ thống ALOHA. Đề xuất của ông là chia thời gian thành những khe thời gian rời rạc gọi là slot, bắt đầu của mỗi khe thời gian là lúc cho mỗi trạm tham gia kênh truyền. Phương pháp này yêu cầu các trạm phải thống nhất với nhau về khe thời gian. Một cách để có sự đồng bộ này là một trạm đặc biệt sẽ phát ra tín hiệu để thông báo bắt đầu khe thời gian. Hiệu năng của Slotted Aloha được mô tả trong Hình 2.8.

Trong những năm 1970, sau khi được tạo ra, Slotted ALOHA được sử dụng trong các thí nghiệm và rồi nhanh chóng bị lãng quên cũng như ALOHA.



Hình 2.8: Hiệu năng của hai giao thức ALOHA và Slotted ALOHA

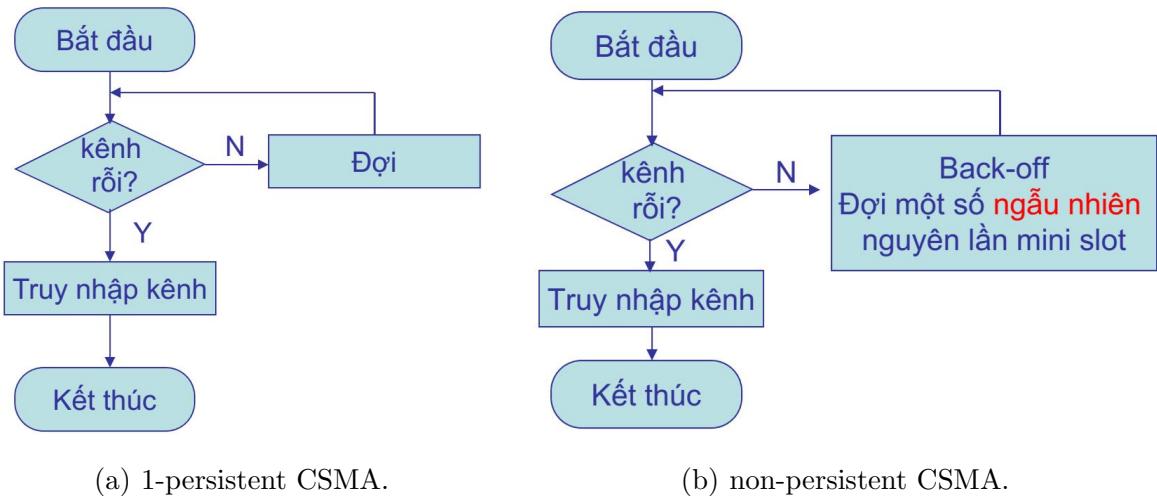
2.2.2 Giao thức đa truy nhập dựa vào việc kiểm tra kênh truyền

Khi một trạm truyền bản tin, nó sẽ không biết các trạm khác đang làm gì, do vậy ở vùng giao của vùng phát sóng hai trạm sẽ xảy ra nhiều xung đột. Do đó, cần có một giao thức mà trạm có thể cảm nhận được kênh truyền. Có nhiều giao thức nghe kênh truyền đã được đưa ra, sau đây em xin trình bày một số phiên bản của giao thức đa truy nhập dựa vào việc cảm nhận kênh truyền CSMA (Carrier Sense Multiple Access).

1-persistent CSMA

1-persistent CSMA [6] là giao thức đơn giản nhất trong các giao thức CSMA. Khi trạm có dữ liệu, đầu tiên nó sẽ cảm nhận kênh truyền xem các trạm khác có đang gửi tin không. Nếu kênh truyền rảnh, trạm sẽ gửi dữ liệu. Ngược lại, nếu kênh truyền bận, trạm sẽ chờ đến khi kênh truyền rảnh. Sau đó trạm sẽ truyền bản tin. Nếu va trạm xảy ra, trạm sẽ chờ một khoảng thời gian ngẫu nhiên và bắt đầu gửi lại. Quá trình gửi tin của trạm sẽ được mô tả trong lưu đồ thuật toán Hình 2.9a. Tuy nhiên, trong trường hợp hai trạm cùng chờ chạm thức 3

thực hiện xong quá trình gửi tin, khi đó hai trạm sẽ cùng gửi tin, xung đột sẽ xảy ra. Giao thức 1-persistent CSMA có hiệu năng tốt hơn so với ALOHA do có quá trình kiểm tra kênh truyền.



Hình 2.9: Lưu đồ thuật toán của hai giao thức 1-persistent CSMA và non-persistent CSMA

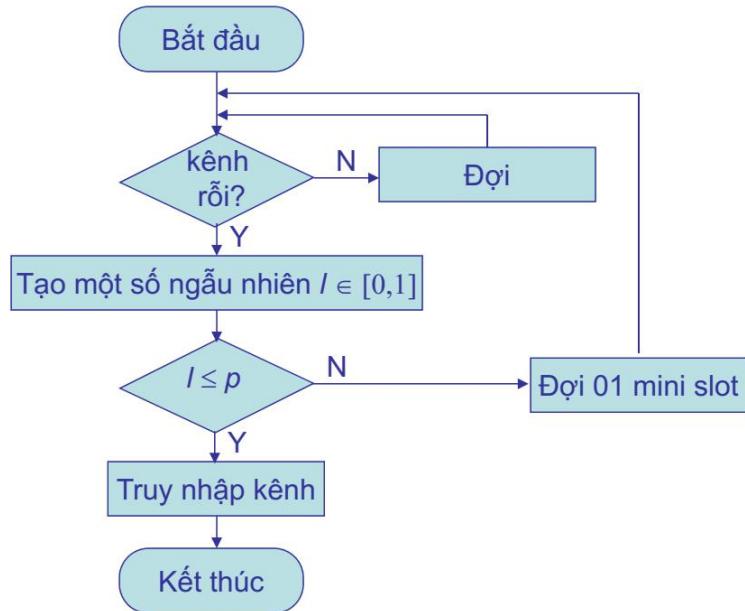
Non-persistent CSMA

Giao thức kiểm tra kênh truyền thức hai là nonpersistent CSMA. Theo tài liệu [6], cũng giống như 1-persistent CSMA, khi cảm nhận được kênh truyền rỗi, trạm sẽ gửi luôn bản tin. Tuy nhiên, khi kênh truyền bận, trạm sẽ không tiếp tục cảm nhận kênh truyền nữa mục đích để không cần ngay lập tức xác định thời điểm kết thúc của quá trình truyền tin trước. Thay vào đó, nó sẽ chờ một khoảng thời gian ngẫu nhiên rồi tiếp tục thuật toán. Hệ quả của quá trình này là sử dụng kênh truyền tốt hơn nhưng thời gian trễ dài hơn so với 1-persistent CSMA. Lưu đồ thuật toán Hình 2.9b mô tả giao thức nonpersistent CSMA.

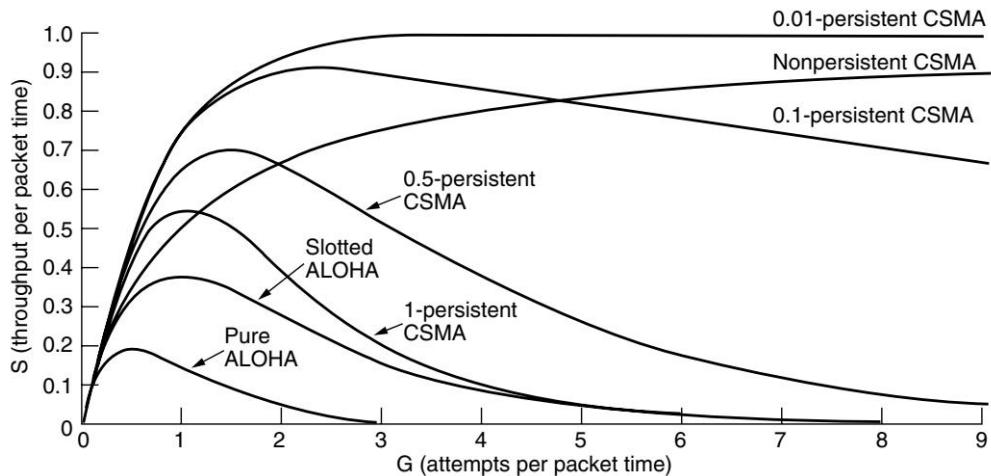
p-persistent CSMA

Giao thức p-persistent CSMA [6] khắc phục những nhược điểm của 1-persistent CSMA. Khi trạm sẵn sàng gửi dữ liệu, nó sẽ kiểm tra kênh truyền. Nếu kênh truyền rảnh, nó sẽ gửi dữ liệu với một xác suất p hoặc đợi một mini slot (mini slot là khoảng thời gian nhỏ hơn rất nhiều so với thời gian truyền tối đa của tín

hiệu) với xác suất $q = 1 - p$ sau đó kiểm tra lại kênh truyền. Khi kênh bận thì đợi kênh rỗi. Hình 2.10 mô tả thuật toán của giao thức p-persistent CSMA. Giá trị xác suất p ảnh hưởng nhiều đến hiệu suất của kênh truyền.



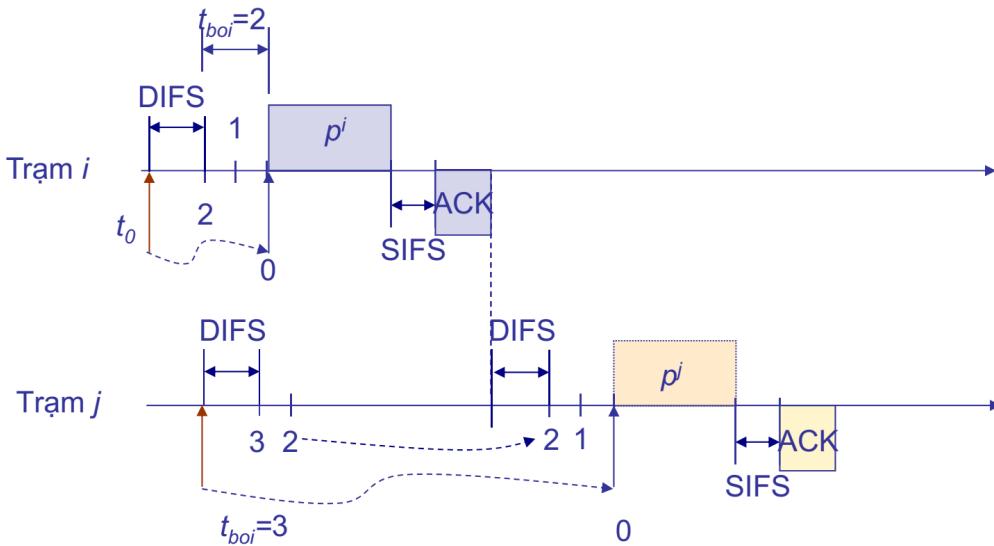
Hình 2.10: Lưu đồ thuật toán giao thức p-persistent CSMA.



Hình 2.11: So sánh hiệu năng sử dụng kênh truyền của các phương pháp truy nhập kênh truyền ngẫu nhiên [6].

Hình 2.11 so sánh hiệu suất của 3 giao thức 1-persistent CSMA, nonpersistent CSMA, p-persistent CSMA cũng như ALOHA và slotted ALOHA.

CSMA/CA



Hình 2.12: Quá trình truyền tin của 2 trạm sử dụng giao thức CSMA/CA

Do cơ chế kiểm tra trạng thái kênh truyền và cơ chế phát hiện va đập hoạt động không hiệu quả trong môi trường vô tuyến, nên cần có một giao thức CSMA được phát triển trong mạng vô tuyến đó là CSMA/CA (CSMA with Collision Avoidance). Cơ chế hoạt động của CSMA/CA (Hình 2.12):

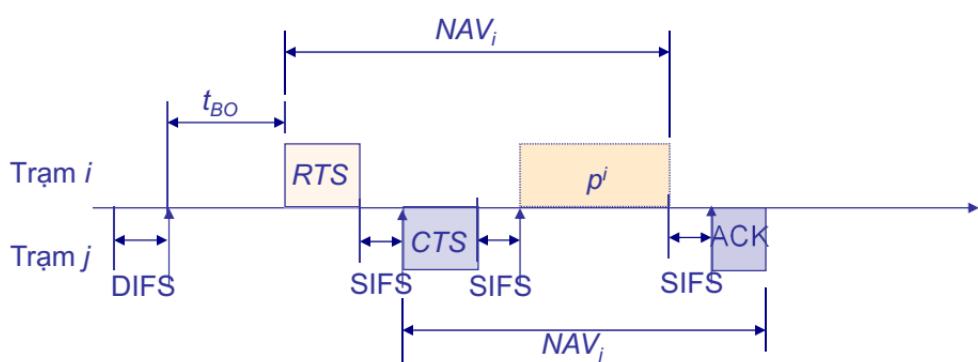
- Trước khi truy nhập kênh truyền, kiểm tra trạng thái kênh truyền. Nếu xảy ra va đập, các trạm sẽ dừng truyền tin và gửi bản tin JAM SIGNAL để báo hiệu cho các trạm khác và back-off theo hàm mũ. Điều này có nghĩa, số lần truy nhập kênh truyền không thành công càng lớn thì thời gian back-off càng tăng,
- Nếu kênh truyền bận: đợi đến khi kênh truyền rỗi,
- Tiếp tục đợi thêm khoảng thời gian DIFS (DCF Inter-Frame Space) cho trước ($DIFS = RTT$),
- Back-off một số mini slot t_{BO} ngẫu nhiên,
- Sau mỗi mini slot: $t_{BO} = t_{BO} - 1$,
- Nếu trong thời gian back-off kênh truyền lại bận thì trạm dừng đếm lùi và bảo toàn giá trị t_{BO} tại thời điểm dừng,

- Sau khi kênh truyền chuyển sang trạng thái rỗi một khoảng thời gian DIFS, trạm tiếp tục đếm lùi,
- Nếu $t_{BO} = 0$ thì truy nhập kênh và gửi gói.

2.2.3 Giao thức đa truy nhập tránh va chạm MACA

Ý tưởng cơ bản của giao thức MACA [6] (Multiple Access with Collision Avoidance) (Karn, 1990) là sử dụng cơ chế tránh va đập. Quá trình truyền tin của trạm sử dụng giao thức MACA được mô tả trong Hình 2.13. Cơ chế tránh va đập gồm có các thành phần:

- Trước khi gửi: bên gửi sẽ quảng bá bản tin RTS (Ready-To-Send) cho các nút xung quanh biết là nó sắp sửa gửi bản tin,
- Khi nhận được RTS: bên thu quảng bá bản tin CTS (Clear-To-Send) để cho bên nhận và các nút xung quanh nó biết nó sẵn sàng nhận bản tin,
- Trong RTS và CTS mang theo bản tin NAV (Network Allocation Vector) chứa thời gian chiếm kênh truyền của bên phát,
- Các trạm khác dừng việc truy nhập kênh truyền trong khoảng thời gian NAV.



Hình 2.13: Quá trình truyền tin của hai trạm sử dụng giao thức MACA

Ở trên là một số giao thức đa truy nhập có thể áp dụng cho mạng không dây, mỗi giao thức có những ưu điểm và cũng có nhược điểm riêng. Đối với đề tài của mình, em xin lựa chọn việc xây dựng giao thức đa truy nhập của mình dựa

trên giao thức ALOHA. Bởi vì, giao thức ALOHA đơn giản, dễ dàng triển khai. Ngoài ra, do dữ liệu cần truyền là nhỏ (dữ liệu lớn nhất trong một lần truyền là 255 byte), cùng với số lượng nút trong mạng ít nên rất phù hợp với việc ứng dụng giao thức ALOHA.

2.3 Module LoRa Ai Thinker SX1278 Ra-02

Ai Thinker là một hãng chuyên về lĩnh vực thiết kế và sản xuất module truyền thông không dây (WiFi, LoRa, GPRS,...) của Trung Quốc.

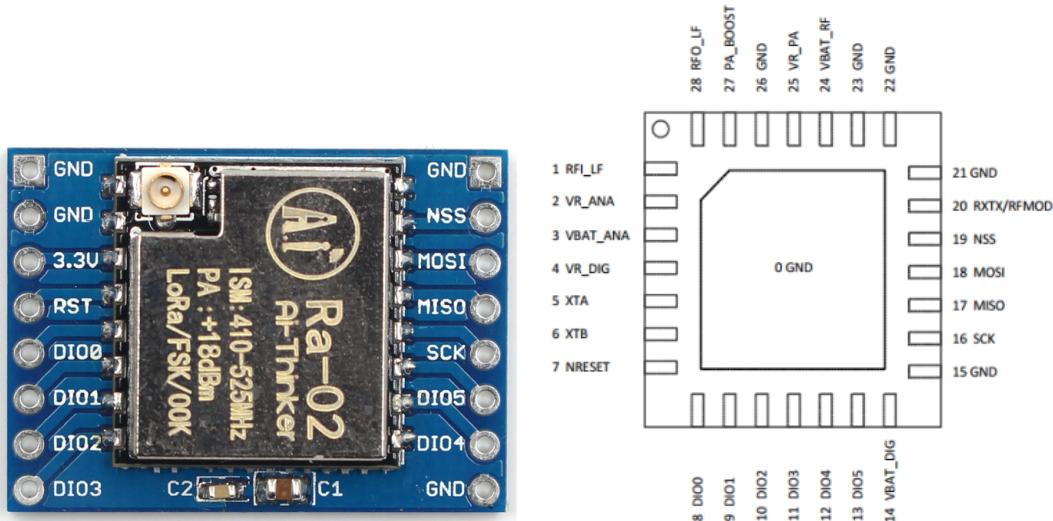
Module truyền thông LoRa SX1278 Ra-02 là một trong số những module sử dụng công nghệ LoRa có giá thành rẻ, sử dụng IC truyền thông LoRa SX1278 của Semtech. Hình ảnh module SX1278 cùng với sơ đồ chân và sơ đồ khối được thể hiện trong Hình 2.14.

Thông số của module LoRa Ai Thinker SX1278 Ra-02 [7] gồm có:

- Kích thước: 17 x 16 mm,
- Hỗ trợ chuẩn giao tiếp SPI,
- Hoạt động trên dải tần: 410 – 525 MHz,
- Hỗ trợ công nghệ điều chế FSK, GFSK, GMSK và LoRa,
- Điện áp hoạt động: 2.5 – 3.7,
- Dòng điện tiêu thụ (Bảng 2.1),
- Công suất phát: 18 ± 1 dBm,
- Nhiệt độ hoạt động: -30 - 85 °C .

Bảng 2.1: Dòng điện tiêu thụ của module LoRa Ai Thinker SX1278 Ra-02 [7]

Tần số	RX	TX	Standby
433 MHz	93 mA	12,15 mA	1,6 mA
470 MHz	97 mA	12,15 mA	1,5 mA



(a) Hình ảnh thực tế Module LoRa Ai Thinker SX1278 Ra-02 []

(b) Sơ đồ chân IC SX1278 [8]

(c) Sơ đồ khối IC SX1278 [8]

Hình 2.14: Module LoRa Ai Thinker SX1278 Ra-02

Theo datasheet của IC SX1278 do Semtech công bố [8], IC SX1278 là một trong những chip thu phát sóng điện từ sử dụng công nghệ LoRa, cung cấp một vùng phủ sóng rộng, tiêu thụ ít năng lượng và khả năng chống nhiễu tốt.

2.4 Vi điều khiển STM32F103

Vi điều khiển STM32F103 là loại vi điều khiển 32 bit với lõi ARM Cortex-M3 trên kiến trúc RICS của hãng STMicroelectronics (ST). Vi điều khiển lõi được sử dụng rộng rãi trong các thiết bị điện tử - tự động bởi khả năng xử lý vượt trội, kích thước nhỏ, dễ dàng cho việc lập trình. Vi điều khiển lõi ARM Cortex-M3

dựa trên kiến trúc ARMv7-M được thiết kế để tối ưu hóa hiệu suất cho các ứng dụng vi xử lý, có nhiều ngoại vi, khả năng đa kết nối đa năng, tương thích với nhiều công cụ (tools) nạp của nhiều hãng khác nhau, cho phép lập trình và phát triển các ứng dụng một cách nhanh chóng. Vi điều khiển Cortex-M3 tiêu thụ ít điện năng và chi phí thấp, đồng thời cung cấp khả năng tính toán cao. Dưới đây là một số thông số hoạt động của vi điều khiển STM32F103 [9]:

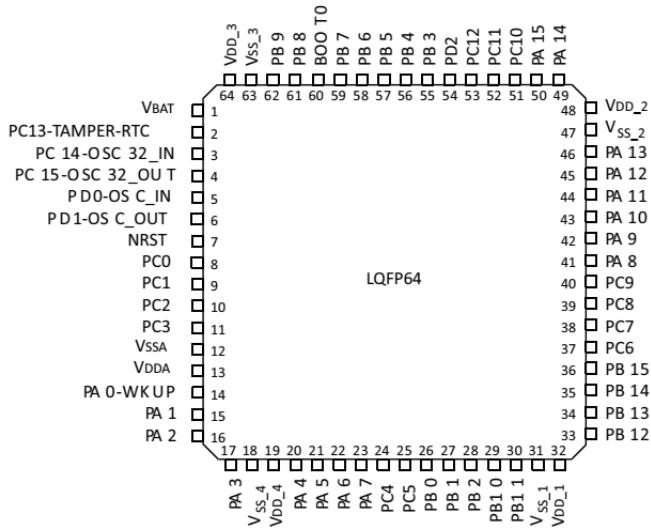
- Tần số cao nhất: 72MHz,
- Dung lượng bộ nhớ flash: 256 KB,
- Dung lượng RAM: 48 KB,
- Dải điện áp 2,0 V – 3,6 V.

Pinout	Low-density devices		Medium-density devices		High-density devices		
	16 KB Flash	32 KB Flash	64 KB Flash	128 KB Flash	256 KB Flash	384 KB Flash	512 KB Flash
	6 KB RAM	10 KB RAM	20 KB RAM	20 KB RAM	48 KB RAM	64 KB RAM	64 KB RAM
144	-	-	-	-	5 × USARTs 4 × 16-bit timers, 2 × basic timers 3 × SPIs, 2 × I ² Ss, 2 × I ² Cs USB, CAN, 2 × PWM timers 3 × ADCs, 2 × DACs, 1 × SDIO FSMC (100 and 144 pins)		
100	-	-					
64	2 × USARTs 2 × 16-bit timers 1 × SPI, 1 × I ² C, USB, CAN, 1 × PWM timer	3 × USARTs 3 × 16-bit timers 2 × SPIs, 2 × I ² Cs, USB, CAN, 1 × PWM timer 2 × ADCs					
48	2 × ADCs						
36							

Hình 2.15: Thông số vi điều khiển STM32F103xxx [9]

Bên cạnh đó, vi điều khiển STM32F103 còn hỗ trợ các tính năng sau [9]:

- Bộ chuyển đổi ADC 12-bit,
- Tích hợp Timer 16-bit với hai bộ điều khiển PWM,
- Đầy đủ các kết nối thông dụng: I²C, SPI, I²S, CAN, USART,
- Hỗ trợ các loại ngoại vi như SDIO, USB,
- Hỗ trợ giao tiếp với ngoại vi thông qua các kênh DMA,
- Nạp chương trình thông qua 2 phương thức SWD và Bootloader,
- Tích hợp cảm biến nhiệt ngay trong CPU với độ phân giải 1 °C.



Hình 2.16: Sơ đồ chân của dòng vi điều khiển STM32F103RCT6 64 chân [9]

2.5 Thiết bị giám sát các tham số môi trường nước BKRES

Thiết bị giám sát tham số môi trường nước BKRES là sản phẩm của đề tài cấp Sở “NGHIÊN CỨU MÔ HÌNH MẠNG CẢM BIẾN KHÔNG DÂY GIÁM SÁT MÔI TRƯỜNG NUỐC PHỤC VỤ PHÁT TRIỂN NUÔI TÔM CHÂN TRẮNG TẠI QUẢNG NINH” do thầy giáo TS. Nguyễn Hữu Phát làm chủ nhiệm. Đây là hệ thống cảm biến tự động thu thập dữ liệu về môi trường nước năm của hồ nuôi trồng thủy sản và cảnh báo kịp thời khi có dấu hiệu bất thường trong môi trường nước.

Hệ thống tích hợp nhiều loại cảm biến, có thể đo đồng thời nhiều thông số của nước cùng lúc, cảnh báo tại chỗ mỗi khi các thông số vượt ngưỡng an toàn, kết hợp với trung tâm dữ liệu tập hợp, phân tích và dự đoán khả năng có thể xảy ra.

Các chức năng chính của hệ thống BKRES gồm có:

- Hệ thống đo đạc được 7 thông số môi trường nước (với thông số chu kỳ thời gian cho trước) bao gồm:
 - Bốn thông số được đo trực tiếp (sử dụng cảm biến) gồm có:
 - * Hàm lượng Oxy trong nước,

* Độ pH,

* Nhiệt độ,

* Nồng độ muối.

- Ba thông số nội suy (tính toán từ các thông số đã đo được qua cảm biến):

* Nồng độ NH_3 ,

* Nồng độ H_2S ,

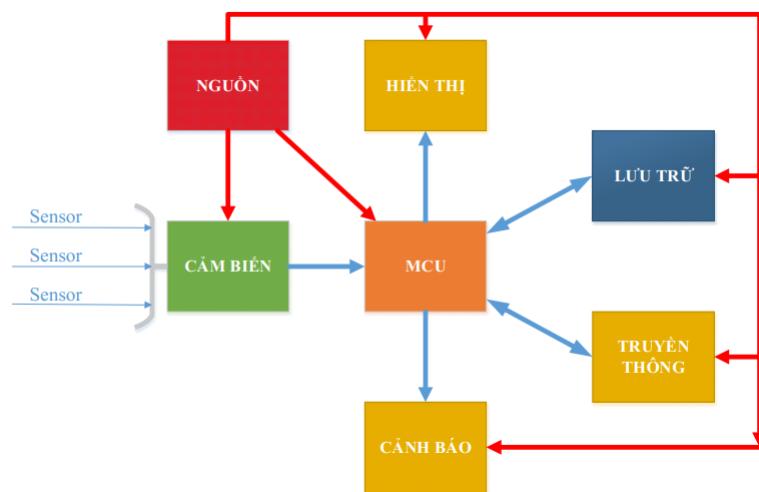
* Nồng độ NO_2 .

- Hiển thị tại chỗ các thông số đo được lên LCD,
- Cảnh báo cho người dùng khi các thông số vượt quá ngưỡng an toàn qua còi, đèn và qua SMS,
- Lưu cấu hình vào bộ nhớ Flash để có thể phục hồi cấu hình khi bị mất điện,
- Lưu lại nhật ký hệ thống vào thẻ nhớ (Có tùy chọn Bật/Tắt),
- Mã hóa rồi gửi dữ liệu về máy chủ thông qua mạng di động,
- Có khả năng điều chỉnh các thông số hệ thống từ xa qua máy chủ hoặc SMS (có mật khẩu),
- Chu kỳ lấy dữ liệu, gửi dữ liệu về máy chủ, chu kỳ lưu giữ vào thẻ nhớ,
- Cập nhật giá trị ngưỡng trên và dưới của thông số (Cho phép tự động cập nhật từ máy chủ),
- Truy vấn thông tin hệ thống gồm các thông số cấu hình và tình trạng hệ thống,
- Cập nhật và bảo trì hệ thống.

Yêu cầu khoa học cần đạt:

- Module phần cứng hoạt động ổn định trong môi trường khắc nghiệt, dễ dàng mở rộng, nâng cấp khi cần,

- Xử lý các dữ liệu từ cảm biến với tốc độ nhanh, tiêu tốn ít tài nguyên hệ thống nhất,
- Đảm bảo các phép tính toán có sai số ít nhất,
- Đảm bảo truyền thông tin cậy,
- Kích thước nhỏ gọn, tiêu thụ năng lượng thấp.



Hình 2.17: Sơ đồ khái niệm một thiết bị BKRES (client)

Máy chủ và thiết bị trao đổi dữ liệu qua việc trao đổi bản tin trên hạ tầng mạng GSM sử dụng giao thức TCP/IP nhằm đảm bảo dữ liệu truyền tải được chính xác và đầy đủ. Để máy chủ có thể nhận diện và giải mã dữ liệu được gửi đến thì phần đầu hiệu nhận diện bản tin và giá trị chỉ định kích thước bản tin cần được giữ nguyên không được mã hóa. Phần dữ liệu mã hóa chỉ bao gồm nội dung bản tin. Do đó, cấu trúc bản tin (Bảng 2.3) được xây dựng gồm có:

- HEADER: phần đầu bản tin và kích thước của toàn bộ bản tin,
- Phần nội dung bản tin (phần được mã hóa) bao gồm:
 - Thông tin về ClientID,
 - Loại bản tin (CommandID),
 - Thông tin về thời gian (Timestamp),
 - Dữ liệu bản tin (Payload).

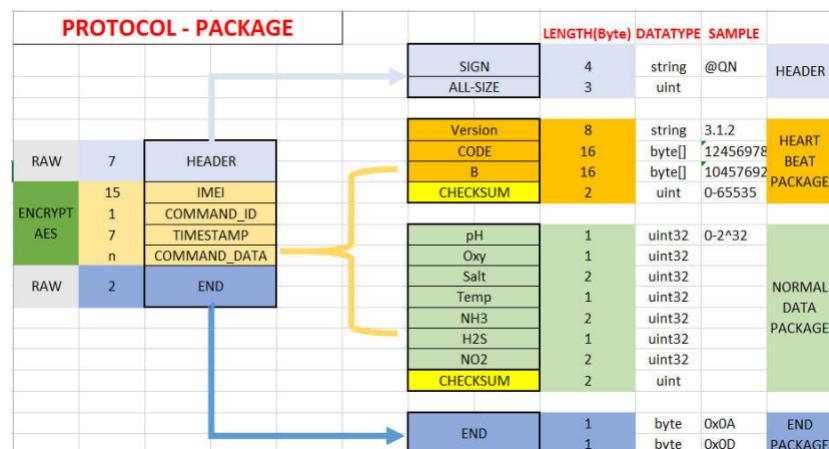
- END: nhận diện kết thúc bản tin.

Nội dung bản tin xác thực gồm:

- Thông tin về phiên bản hệ thống,
- Mã CODE (một thành phần tạo khóa AES),
- Mã B (một thành phần tạo khóa AES),
- Checksum.

Nội dung bản tin dữ liệu gồm:

- Thông tin về dữ liệu cảm biến: pH, Oxy, Salt, Temp, NH3, H2S,
- Checksum.



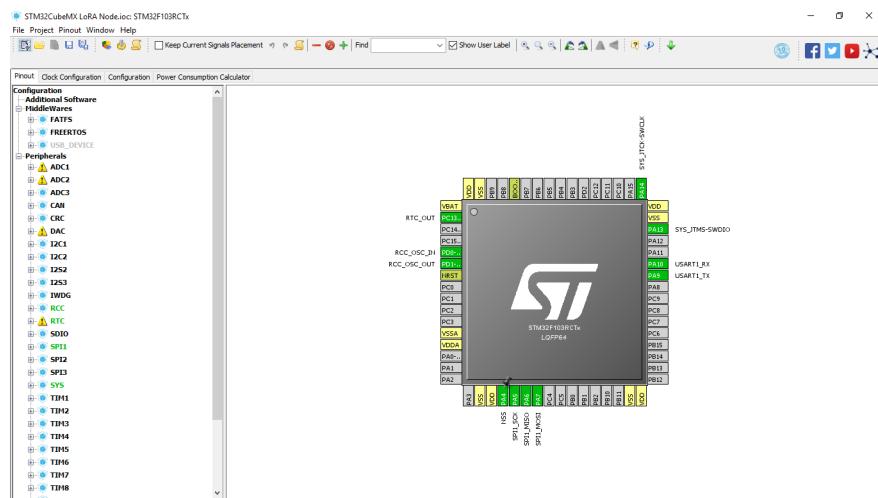
Hình 2.18: Cấu trúc bản tin gửi dữ liệu trao đổi giữa máy chủ và client

Hệ thống BKRES, sử dụng phương pháp truyền thông qua hệ thống GSM. Với việc sử dụng phương pháp này, mỗi thiết bị trong mạng cần phải có một module sim để truyền dữ liệu cho server, bên cạnh đó cần phải tốn chi phí hàng tháng để duy trì dịch vụ cho mỗi thiết bị. Phương pháp này có một số nhược điểm lớn là yêu cầu về cơ sở hạ tầng cao (cần có hệ thống GSM), khi hệ thống BKRES mở rộng (số lượng thiết bị lớn hơn) thì yêu cầu chi phí về việc duy trì hệ thống là lớn. Do đó, hệ thống BKRES hướng đến việc sử dụng công nghệ truyền thông LoRa để thu thập dữ liệu.

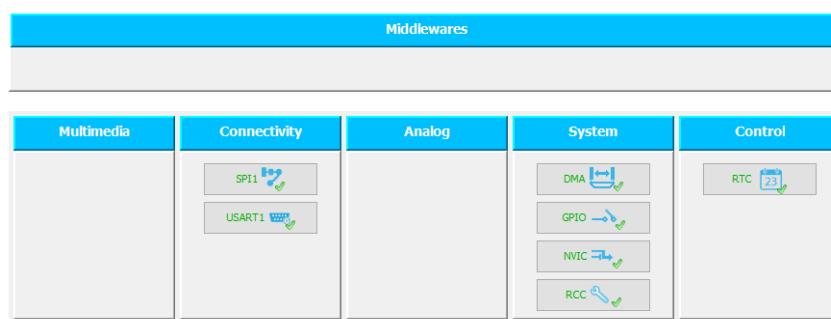
2.6 Một số công cụ hỗ trợ khác

2.6.1 Phần mềm hỗ trợ cấu hình vi điều khiển

Để cấu hình chức năng phần cứng cho các dòng vi điều khiển, ST đưa ra bộ công cụ STM32CubeMX [10]. Phần mềm này hỗ trợ cấu hình các chức năng cho một vi điều khiển, phục vụ cho một dự án mới một cách trực quan, nhanh chóng. Ngoài ra, STM32CubeMX tự động download các driver mới nhất của ST dành cho các dòng chip của mình. Vì vậy đảm bảo các API của hãng sẽ luôn được cập nhật. Công cụ này còn hỗ trợ tự động sinh project mới với tùy chọn các IDE thông dụng. Giao diện phần mềm và giao diện cấu hình cho từng ngoại vi được thể hiện trong Hình 2.19 và Hình 2.20.



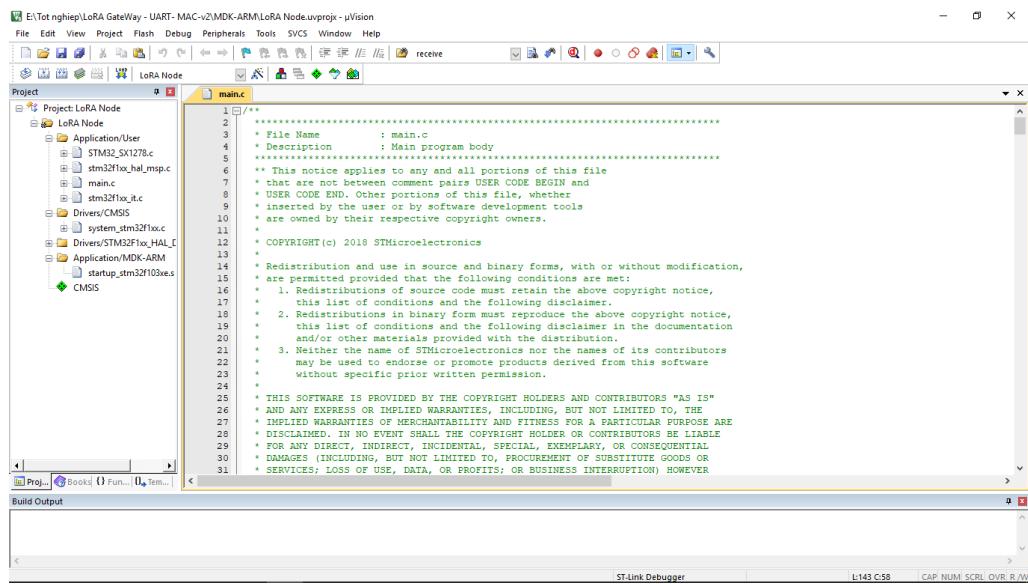
Hình 2.19: Giao diện phần mềm STM32CubeMX hỗ trợ cấu hình



Hình 2.20: Giao diện cấu hình chi tiết cho từng ngoại vi của STM32CubeMX

2.6.2 Phần mềm lập trình và biên dịch firmware Keil C

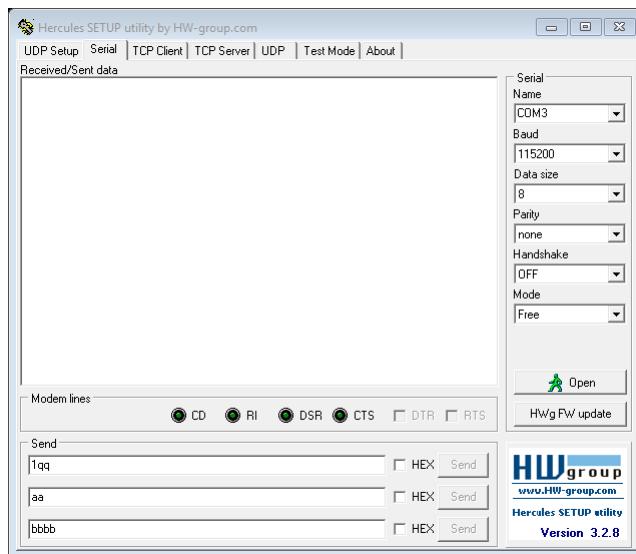
Keil C μ Vision (Hình 2.21) là công cụ chuyên dụng hỗ trợ lập trình nhúng rất nổi tiếng. Keil C hỗ trợ hầu hết các loại chip vi xử lý trên thị trường và hỗ trợ 2 loại ngôn ngữ C/C++, Assembly. Nó được coi là IDE (môi trường phát triển) khá toàn diện, bao gồm khả năng biên dịch với trình biên dịch thương mại được tối ưu hóa, hỗ trợ nạp code và gỡ lỗi chuyên nghiệp.



Hình 2.21: Giao diện phần mềm lập trình firmware Keil C μ Vision

2.6.3 Phần mềm hỗ trợ giao tiếp UART Hercules

Trong quá trình lập trình, ngoài tính năng debug do phần mềm lập trình Keil C hỗ trợ, em còn sử dụng phần mềm Hercules (Hình 2.22) để hỗ trợ cho công việc kiểm tra và sửa lỗi. Hercules là một phần mềm hỗ trợ giao tiếp UART giúp hiển thị thông báo từ vi điều khiển lên máy tính, sử dụng đơn giản và thân thiện với người dùng. Sau khi kit kết nối với máy tính qua cổng COM, ta chỉ cần chọn đúng cổng COM được kết nối, chọn baud rate trùng với baud rate được cài đặt ở vi điều khiển, thế là có thể xem được thông báo từ vi điều khiển.



Hình 2.22: Giao diện phần mềm HERCULES

2.7 Kết luận

Từ những kiến thức em đã tìm hiểu và do những điểm phù hợp của ALOHA đối với hệ thống BKRES, em quyết định dựa trên giao thức ALOHA để xây dựng giao thức đa truy nhập và tích hợp giao thức đa truy nhập này vào thiết bị BKRES. Các bước tiến hành xây dựng giao thức sẽ được trình bày trong chương sau.

Chương 3

Xây dựng thuật toán đa truy nhập

Từ những cơ sở lý thuyết thu thập ở chương 2, chương này em sẽ trình bày quá trình thiết kế thuật toán đa truy nhập để ứng dụng thuật toán vào hệ thống BKRES-LoRa.

3.1 Phân tích yêu cầu bài toán đa truy nhập

Từ việc phân tích chi tiết yêu cầu và chức năng của thuật toán đa truy nhập cần phải đáp ứng, em đã xác định được yêu cầu chức năng và phi chức năng của hệ thống như sau:

- Yêu cầu chức năng:
 - Có khả năng đa truy nhập ($n - 1$), tức là nhiều nút sẽ gửi dữ liệu đến 1 Gateway,
 - Có khả năng xử lý bài toán join/disjoin mạng,
 - Hoạt động được với số lượng nút lớn,
- Yêu cầu phi chức năng:
 - Các nút truyền dữ liệu chính xác,
 - Nút tiêu tốn ít năng lượng,
 - Thiết bị hoạt động ổn định với thuật toán,

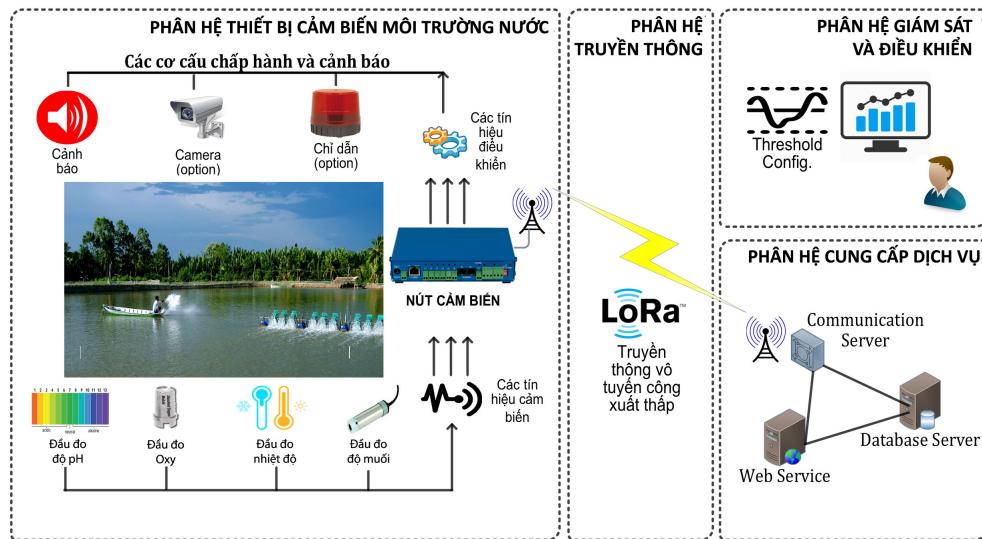
3.2 Hệ thống BKRES-LoRa

3.2.1 Kiến trúc tổng thể hệ thống

Hệ thống giám sát tham số môi trường nước BKRES sau khi được tích hợp thêm module truyền thông LoRa thì dữ liệu thu thập được từ các nút sẽ được gửi đến Gateway thông qua module truyền thông LoRa, còn Gateway sẽ giao tiếp với server thông qua một số giao thức như TCP/IP, GSM,...Hình 3.1 sẽ mô tả rõ hơn kiến trúc gồm 4 phân hệ chính của hệ thống BKRES-LoRa.

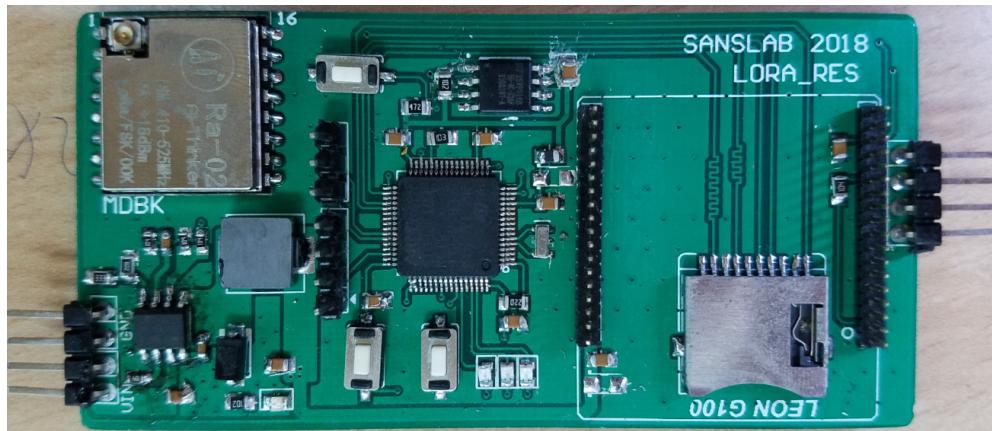
- Phân hệ cảm biến và xử lý dữ liệu: Ở mỗi nút mạng được trang bị các bộ cảm biến ghi đo 4 tham số (hàm lượng Oxy, nhiệt độ, độ pH và nồng độ muối) để thu thập dữ liệu tham số môi trường. Dữ liệu cảm biến được gửi đến bộ điều khiển trung tâm (sử dụng vi điều khiển STM32) để phân tích, xử lý, đóng gói và gửi đến module truyền thông LoRa.
- Phân hệ truyền thông sử dụng LoRa: Hoạt động theo mô hình đơn chặng, có nhiệm vụ gửi dữ liệu từ phân hệ cảm biến đến Gateway và từ đó dữ liệu được gửi đến server trung tâm.
- Phân hệ cung cấp dịch vụ: Sau khi nhận dữ liệu từ phân hệ truyền thông, phân hệ này có nhiệm vụ xử lý dữ liệu và cung cấp các dịch vụ cho phân hệ giám sát và người dùng.
- Phân hệ giám sát và điều khiển: có nhiệm vụ hiển thị dữ liệu cảm biến một cách trực quan thông qua ứng dụng di động, ứng dụng web. Người dùng có thể cấu hình hệ thống, đặt các mức ngưỡng cảnh báo,... trên ứng dụng.

Module LoRa SX1278 được lựa chọn để tích hợp vào thiết bị cảm biến như thể hiện trên Hình 3.2. Nút cảm biến được thiết kế và chế tạo tài phòng nghiên cứu SANSLAB, gồm các thành phần phần cứng chính như: phân hệ cảm biến các tham số môi trường nước, phân hệ truyền thông, phân hệ vi xử lý và điều khiển, phân hệ cấp nguồn và phân hệ hiển thị, cảnh tại chỗ. Phần mềm trên nút cảm biến gồm firmware và drivers điều khiển hoạt động các phân hệ và ngoại vi, thuật toán đa truy nhập cũng được nhúng trong firmware của thiết bị.



Hình 3.1: Kiến trúc hệ thống

3.2.2 Nút mạng cảm biến tích hợp module SX1278



Hình 3.2: Nút cảm biến tích hợp chip LoRa SX1278

Nút cảm biến được thiết kế gồm các khối có nhiệm vụ như sau:

- Khối nguồn: có nhiệm vụ cung cấp nguồn cho toàn bộ mạch. Mức điện áp 3,3 V cung cấp cho vi điều khiển, thiết bị ngoại vi cùng với cảm biến. Nguồn cung cấp cho mạch là pin, nhỏ gọn và có thể sạc được nên rất phù hợp với các thiết bị trong mạng WSN,
- Khối xử lý trung tâm: có nhiệm vụ lấy dữ liệu từ các thiết bị cảm biến, xử lý rồi chuyển qua khối truyền thông, cũng như xử lý các thông tin nhận được từ khối truyền thông. Khối xử lý trung tâm sử dụng vi điều khiển

STM32F1 là một loại vi điều khiển lõi Cortex M3 có tốc độ xử lý cao, cấu hình mạnh, tiết kiệm năng lượng với kích thước nhỏ gọn và giá thành rẻ,

- Khối truyền thông: có nhiệm vụ truyền nhận dữ liệu giữa các thiết bị trong mạng. Khối truyền thông sử dụng module LoRa Ai Thinker SX1278 được kết nối với khối xử lý trung tâm qua giao tiếp SPI,
- Khối lưu trữ dữ liệu: được dùng để lưu trữ cấu hình hệ thống cùng với nhật ký cảm biến. Khối này gồm IC flash NAND SST25VF016B và thẻ nhớ. IC flash có tốc độ xử lý cao, dung lượng bộ nhớ nhỏ thích hợp với việc lưu trữ phần mềm cùng với cấu hình của hệ thống. Thẻ nhớ được sử dụng để lưu trữ nhật ký cảm biến, dễ dàng cho người sử dụng có thể kiểm tra và sao lưu dữ liệu.

3.2.3 Mô hình hoạt động của hệ thống BKRES-LoRa

Thuật toán đa truy nhập đề xuất được phát triển và thử nghiệm trên mô hình gồm 3 nút cảm biến và một Gateway như Hình 3.3. Để đánh giá khả năng truyền dữ liệu, độ ổn định, khả năng tùy biến và tự cấu hình của các nút trong mạng, các thông số cơ bản của các nút sẽ được thiết lập theo như Bảng 3.1.

Bảng 3.1: Bảng thông số cấu hình của các nút trong mạng

Thông số	Giá trị
Channel	11 (433 MHz)
BW	125 kHz
CR	4/5
SF	7
Header	ON
CRC	ON

Hoạt động của mô hình được mô tả như sau. Ban đầu, mỗi nút mới (sau khi tự động cấu hình) sẽ gửi yêu cầu tham gia mạng đến Gateway, sau khi được chấp nhận thì nút bắt đầu quá trình gửi dữ liệu đến Gateway. Trong quá trình này, mỗi nút sẽ nhận dữ liệu đã được xử lý từ vi điều khiển, rồi gửi dữ liệu đến Gateway theo một chu kỳ định sẵn. Trong trường hợp xảy ra lỗi với bộ truyền

thông LoRa, nút sẽ gửi dữ liệu trực tiếp đến server trung tâm qua mạng GSM (có sẵn trong BKRES phiên bản cũ), đến khi hết xảy ra lỗi với bộ truyền thông, nút lại bắt đầu lại quá trình như bình thường.



Hình 3.3: Mô hình hoạt động của mạng

3.3 Xây dựng thuật toán đa truy nhập

Do số lượng nút trong mạng ít, nên em quyết định sử dụng phương pháp đa truy nhập ngẫu nhiên, tức là khi nút có dữ liệu nó sẽ gửi luôn bản tin cho Gateway.

3.3.1 Thuật toán đa truy nhập của Gateway

Trong thuật toán đa truy nhập, Gateway có các nhiệm vụ chính sau:

- Nhận dữ liệu từ các nút,
- Quản lý nút trong mạng.

Sau khi đã tự cấu hình, trong vòng lặp vô hạn, Gateway liên tục mở kênh truyền để nhận bản tin. Nếu bản tin đó là do một nút mới gửi đến (nút chưa tham gia vào mạng) thì gateway sẽ tiến hành quá trình xác thực nút, rồi gửi bản tin phản hồi cho nút. Còn nếu bản tin là bản tin chứa dữ liệu, Gateway sẽ nhận và xử lý dữ liệu và tiếp tục quá trình mở kênh truyền để nhận dữ liệu. Algorithm 1 thể hiện thuật toán xử lý của Gateway.

Algorithm 1 Thuật toán của Gateway

```

1: pass  $\leftarrow$  @Rsanslab
2: Gateway tự động cấu hình  $\triangleright$  Gateway sẽ tự động cài một số thông số như
   ID, SF, CR, tần số, ...
3: while 1 do
4:   if Nhận được gói tin then
5:     packet  $\leftarrow$  packet_receive  $\triangleright$  gói tin nhận được
6:     if packet là gói tin gửi dữ liệu then
7:       data  $\leftarrow$  packet.data
8:       Xử lý dữ liệu  $\triangleright$  Gateway có thể gửi dữ liệu lên Server, hiển thị
   hoặc lưu dữ liệu
9:     else if packet là gói tin yêu cầu tham gia mạng then
10:    ID  $\leftarrow$  packet.src  $\triangleright$  Lấy ID của nút gửi đến
11:    authentication  $\leftarrow$  packet.data  $\triangleright$  Lấy dữ liệu bản tin xác thực
12:    response  $\leftarrow$  true
13:    if authentication  $\neq$  pass then
14:      response  $\leftarrow$  false
15:    end if
16:    gửi bản tin phản hồi response cho nút ID
17:  end if
18: end if
19: end while

```

3.3.2 Thuật toán đa truy nhập của nút

Chức năng chính của nút trong thuật toán đa truy nhập:

- Khi nút chưa tham gia mạng, nút sẽ gửi yêu cầu để tham gia mạng. Khi đó xảy ra một số trường hợp sau:
 - Quá trình yêu cầu không thành công, nút sẽ gửi lại,
 - Nếu bản tin xác thực sai, nút sẽ không hoạt động nữa,
 - Nếu quá trình yêu cầu tham gia mạng thành công, nút sẽ gửi dữ liệu.
- Sau khi tham gia mạng thành công, nút sẽ gửi dữ liệu theo chu kỳ.

Algorithm 2 Thuật toán của Nút

```

1: msgAuthentication ← @Rsanslab
2: Nút tự động cấu hình      ▷ Nút sẽ tự động cài một số thông số như ID, SF,
   CR, tần số, ...
3: idGateway ← Địa chỉ Gateway
4: response ← 2           ▷ response = 0 → bản tin xác thực chính xác; response
   = 1 → bản tin xác thực không chính xác; response = 2 → nút chưa tham
   gia mạng.
5: RequestJoinNetworks(idGateway, msgAuthentication)
6: while 1 do
7:   if response == 0 then
8:     data ← lấy dữ liệu từ cảm biến
9:     cnt = 0
10:    repeat
11:      state ← sendPacket(idGateway, data)
12:      if state > 0 then
13:        Delay()
14:      end if
15:      cnt ← cnt + 1
16:    until state > 0 and cnt < threshold      ▷ threshold là số lần gửi tối đa
17:    else if response == 1 then
18:      Sleep()
19:    else
20:      RequestJoinNetworks(idGateway, msgAuthentication)
21:    end if
22: end while

```

Sau khi được cấu hình, nút sẽ gửi bản tin xác thực cho Gateway (ID của Gateway được thiết lập sẵn). Nút sẽ thực hiện quá trình yêu cầu tham gia mạng (quá trình yêu cầu tham gia mạng được mô tả bởi Algorithm 3) đến khi thành công (nếu bản tin xác thực sai, nút sẽ dừng lại). Sau khi nút nhận được bản tin chấp nhận tham gia vào mạng của Gateway, nút sẽ tiến hành quá trình gửi dữ liệu. Quá trình xử lý của nút được thể hiện bởi Algorithm 2.

Trong quá trình yêu cầu tham gia vào mạng, nút sẽ gửi bản tin xác thực

đến khi nào nhận lại được bản tin phản hồi của Gateway. Khi đó nút sẽ biết được mình đã được chấp thuận tham gia mạng hay chưa. Còn về phần Gateway, sau khi nhận được bản tin xác thực được gửi từ nút, Gateway sẽ kiểm tra bản tin xác thực đó có chính xác không. Kết quả của quá trình kiểm tra bản tin xác thực sẽ được Gateway gửi lại cho nút.

Algorithm 3 RequestJoinNetworks

```

1: procedure REQUESTJOINNETWORKS(idGateway, msgAuthentication)
2:   repeat
3:     state  $\leftarrow$  sendPacket(idGateway, msgAuthentication)     $\triangleright$  state là trạng
   thái của quá trình gửi tin; state = 0 gửi bản tin thành công; state > 0 gửi
   bản tin chưa thành công
4:     if state > 0 then
5:       Delay()
6:     else if Nhận được bản tin hồi đáp từ Gateway then
7:       response  $\leftarrow$  packet_receive.data
8:       if response không chính xác then
9:         state  $\leftarrow$  1
10:      end if
11:    end if
12:  until state > 0
13: end procedure
  
```

3.4 Xây dựng chi tiết và đánh giá

3.4.1 Bản tin yêu cầu tham gia mạng

Bản tin yêu cầu tham gia mạng sẽ được gửi khi nút đã tự cấu hình. Để giảm số lượng bản tin phải trao đổi giữa nút và Gateway để tham gia mạng, thì bản tin yêu cầu tham gia mạng sẽ chứa luôn bản tin xác thực. Trong quá trình thử nghiệm, chuỗi xác thực sẽ là "@Rsanslab" và được cấu hình sẵn trong firmware của cả nút và Gateway. Gateway sau khi nhận được bản tin yêu cầu tham gia mạng của nút, sẽ tách phần dữ liệu (chứa chuỗi xác thực) để so sánh với chuỗi xác thực của nó rồi tiến hành phản hồi cho nút. Cấu trúc gói yêu cầu tham gia

mạng được mô tả trong bảng 3.2. Trong thí nghiệm, bản tin yêu cầu tham gia

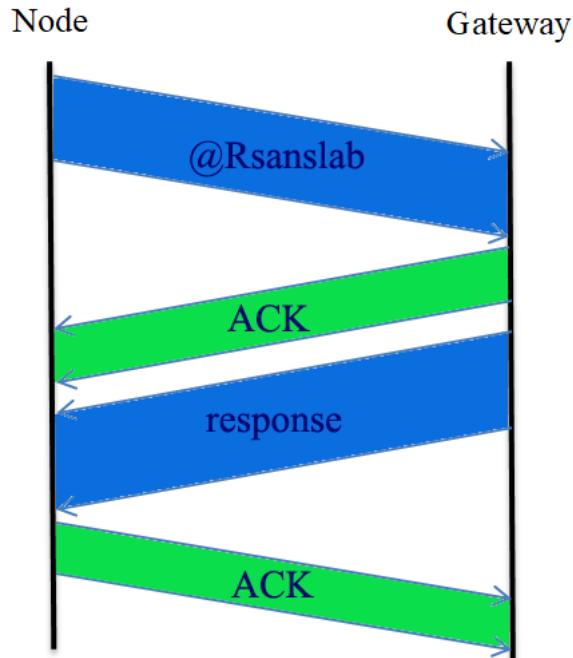
Bảng 3.2: Cấu trúc bản tin yêu cầu tham gia mạng

Header	Data
2 byte	max. 255 byte

mạng tương đối đơn giản mục đích để giảm thời gian xác thực của Gateway. Cấu trúc cụ thể bản tin:

- Header: @R,
- Data: sanslab.

Luồng trao đổi bản tin giữa nút và Gateway sẽ được mô tả trong Hình 3.4. Dữ liệu của bản tin phản hồi chỉ là "1" nếu bản tin xác thực chính xác hoặc là "0" nếu bản tin xác thực không chính xác.



Hình 3.4: Luồng trao đổi dữ liệu trong quá trình yêu cầu tham gia mạng

Thuật toán yêu cầu tham gia mạng sẽ xảy ra 2 trường hợp:

- Trường hợp tốt nhất: Nút gửi yêu cầu tham gia mạng đến Gateway và

nhận luôn được bản tin phản hồi của Gateway. Khi đó độ phức tạp của thuật toán là $O(c)$.

- Trường hợp xấu nhất: Nút gửi yêu cầu tham gia mạng đến Gateway nhưng không nhận được bản tin phản hồi của Gateway (hoặc gửi bản tin không thành công), nút sẽ gửi lại bản tin yêu cầu tham gia mạng. Khi đó độ phức tạp của thuật toán là $O(n)$.

3.4.2 Bản tin gửi dữ liệu

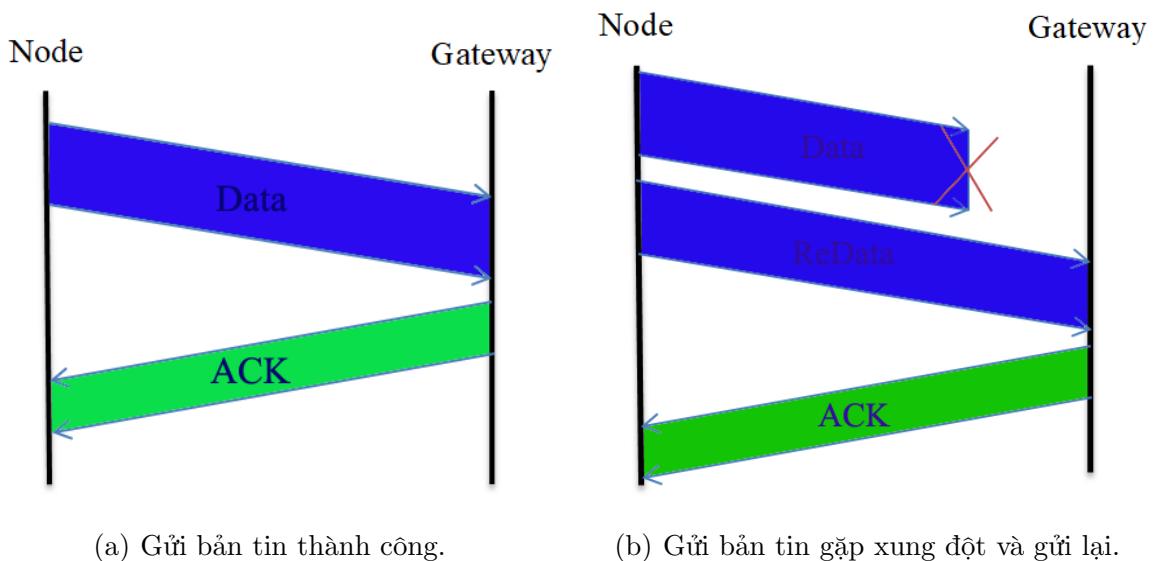
Sau khi đã tham gia mạng thành công, theo chu kỳ, nút bắt đầu gửi bản tin chứa dữ liệu đến Gateway. Do số lượng nút nhỏ (3 đến 4 nút) nên khả năng xảy ra va chạm dẫn đến mất gói là rất ít. Do đó, nguyên nhân chính khiến nút gửi bản tin không thành công là do nút gửi bản tin không đúng vào thời điểm Gateway mở kênh truyền để nhận bản tin. Khi không gửi được bản tin thì nút sẽ trễ một khoảng thời gian, sau đó sẽ gửi lại bản tin. Cũng như bản tin yêu cầu tham gia mạng, bản tin chứa dữ liệu có phần header để nhận diện, ngoài ra bản tin còn chứa các thông số khác. Bản 3.3 mô tả cấu trúc bản tin gửi dữ liệu.

Bảng 3.3: Cấu trúc bản tin dữ liệu

Header	IMEI	Command_ID	Timestamp	Data	End
7 byte	15 byte	1 byte	7 byte	max. 255 byte	2 byte

Luồng trao đổi bản tin dữ nút và Gateway được mô tả trong Hình 3.4.2.

- Header: chứa dấu hiệu nhận diện bản tin chứa dữ liệu ("@DN") và kích thước của bản tin,
- IMEI: là mã nhận dạng giữa các thiết bị với nhau, mỗi thiết bị có một IMEI khác nhau được cấu hình sẵn,
- Command_ID: là loại bản tin,
- Timestamp: chứa thông tin về thời gian của dữ liệu (gồm: ngày, tháng, năm, giờ, phút, giây) được lấy từ vi điều khiển,



Hình 3.5: Luồng dữ liệu trong hai trường hợp

- Data: chứa dữ liệu các thông số môi trường nước, gồm lần lượt các trường sau: pH, Oxy, Salt, Temp, NH_3 , H_2S , NO_2 ,
- End: báo hiệu kết thúc bản tin, là ký tự "\$".

Trong cả hai trường hợp gửi bản tin gặp xung đột và không gặp xung đột độ phức tạp của thuật toán là $O(c)$ do số lần gửi lại của nút không được quá số lần quy định để đảm bảo tính thời gian thực của dữ liệu.

3.5 Kết luận

Sau khi đã xây dựng và đánh giá thuật toán đa truy nhập một cách kỹ lưỡng, thuật toán sẽ được triển khai dưới dạng code và nhúng vào những thiết bị trong hệ thống BKRES-LoRa. Quá trình thử nghiệm cùng với kết quả thử nghiệm sẽ được mô tả trong Chương 4.

Chương 4

Kết quả và đánh giá

Sau khi đã xây dựng thành công thuật toán đã trình bày trong 3, em tiến hành thực nghiệm để phát hiện và điều chỉnh để thuật toán đa truy nhập tốt hơn. Chương này, em xin trình bày về những kết quả đã đạt được và hướng phát triển của đề tài.

4.1 Các kết quả đạt được

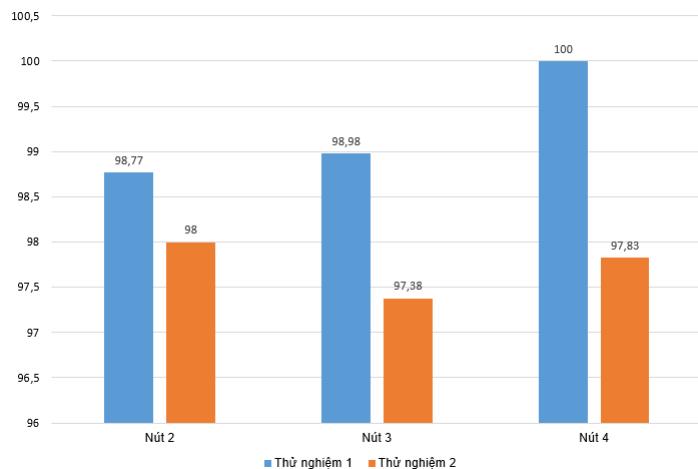
4.1.1 Phân tích kết quả các thử nghiệm

Sau khi tự động cấu hình, nó sẽ gửi yêu cầu tham gia mạng đến Gateway và bắt đầu gửi tin (nếu được Gateway chấp thuận). Hình 4.1 mô tả quá trình kiểm tra yêu cầu tham gia mạng của nút và nhận gói tin từ nút.

```
Received/Sent data
Start configuring LoRa module
LoRa successfully configured

Receive msg
Receive Data form node 9: @DM028|05:13:35 24/05/2018|ID: 354725065508149|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Send accepted Msg to Node 9
Gateway: 1
Receive msg
Receive Data form node 9: @DM028|05:10:02 24/05/2018|ID: 354725065508131|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:15:08 24/05/2018|ID: 354725065508149|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:10:14 24/05/2018|ID: 354725065508131|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:15:20 24/05/2018|ID: 354725065508149|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:10:27 24/05/2018|ID: 354725065508131|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:15:33 24/05/2018|ID: 354725065508149|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:10:39 24/05/2018|ID: 354725065508131|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
Receive msg
Receive Data form node 9: @DM028|05:15:46 24/05/2018|ID: 354725065508149|Temp: 28.0[oC]|Oxy:0.67[mg/1]|PH:6.8|NH3:0.066[mg/1]|H2S:0.065[mg/1]
|Salt:0.069[]|44
```

Hình 4.1: Quá trình xử lý của Gateway



Hình 4.2: Kết quả thí nghiệm đánh giá độ mất gói

Thử nghiệm 1: Đánh giá mất gói trong điều kiện thuận lợi (PTN)

Thí nghiệm này được triển khai tại phòng nghiên cứu Sanslab, thí nghiệm thực hiện gồm có 3 nút và 1 gateway. Các thông số kỹ thuật được cấu hình như trong Bảng 3.1, các nút nằm trong bán kính 10 m của Gateway. Ngoài ra có hai số thông số được thiết lập trước là ID của nút và ID của Gateway. Kết quả của thực nghiệm được mô tả trong Bảng 4.1 và biểu đồ Hình 4.2.

Bảng 4.1: Bảng kết quả độ mất gói trong điều kiện thuận lợi

	Nút 2	Nút 3	Nút 4
Gửi gói tin thành công	98,77,02 %	98,98 %	100 %
Hủy gói tin	1,23 %	1,03 %	0 %

Thử nghiệm 2: Đánh giá độ mất gói trong điều kiện thách thức

Thí nghiệm 2 được thực hiện tại khuôn viên Đại học Bách Khoa Hà Nội, các thông số kỹ thuật của thử nghiệm 2 được thiết lập giống với thử nghiệm 1. Trong thử nghiệm này, Gateway được đặt tại tầng 6 thư viện Tạ Quang Bửu, nút đặt rải rác trong khuôn viên trường với môi trường có vật cản. Kết quả thử nghiệm 2 được ghi lại trong Bảng 4.2 và được biểu diễn trên Hình 4.2.

Bảng 4.2: Bảng quả độ mất gói trong điều kiện thách thức

	Nút 2	Nút 3	Nút 4
Gửi bản tin thành công	98,0 %	97,38 %	97,83 %
Hủy bản tin	2,0 %	2,62 %	2,17 %

Đánh giá kết quả thử nghiệm

Trong cả 2 thử nghiệm, độ mất gói là rất nhỏ (nhỏ hơn 3%) chứng tỏ thuật toán đa truy nhập hoạt động tốt với số lượng nút nhỏ (3 nút). Bên cạnh đó, tỷ lệ mất gói của thử nghiệm 2 cao hơn thử nghiệm 1, cho thấy được khoảng cách và vật cản có ảnh hưởng đến độ mất gói trong quá trình truyền dữ liệu.

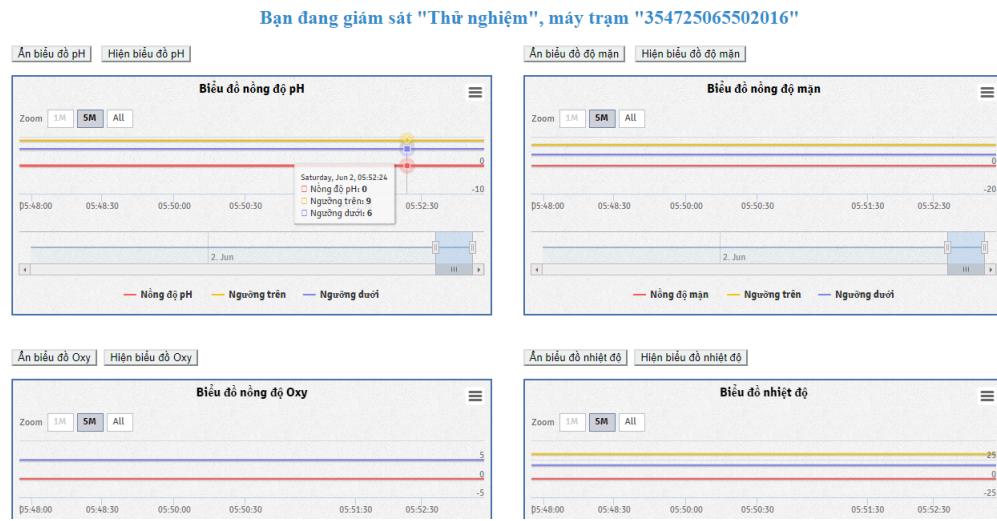
4.1.2 Hệ thống BKRES-LoRa

Công nghệ LoRa đã được phòng nghiên cứu Sanslab áp dụng vào hệ thống BKRES thành công. Module truyền thông được tích hợp vào kit điều khiển (hình 4.3) với kích thước gọn nhẹ, tiêu tốn ít năng lượng (module sử dụng Pin để cung cấp năng lượng).

Những bản tin Gateway nhận được sẽ được gửi đến server và những dữ liệu đó được hiển thị các ứng dụng (hình 4.4).



Hình 4.3: Kit điều khiển BKRES-LoRa



Hình 4.4: Hiển thị dữ liệu trên ứng dụng web

4.2 Đánh giá thuật toán và định hướng phát triển

Thuật toán đa truy nhập có những ưu điểm và nhược điểm sau:

- **Ưu điểm:**
 - Thuật toán hoạt động tốt,
 - Hiệu năng truyền gói thành công cao hơn nhiều so với các giao thức truy nhập ngẫu nhiên khác,
 - Các thiết bị sử dụng thuật toán hoạt động ổn định, tiêu thụ ít năng lượng (so với sử dụng module sim).
- **Nhược điểm:**
 - Chưa xác định được độ ảnh hưởng của số lượng thiết bị đến thuật toán,
 - Chưa xác định được các thời gian tiêu tốn để thực hiện các quá trình (tham gia mạng, gửi dữ liệu) để điều chỉnh trễ hợp lý,

- Quá trình yêu cầu tham gia mạng phức tạp, ảnh hưởng đến quá trình gửi tin của nút khác,
- Năng lượng tiêu thụ của Gateway lớn (vì Gateway không có khoảng thời gian để nghỉ),
- Hầu như chưa có sự quản lý của Gateway đối với các nút trong mạng.

Thuật toán cần được phát triển hơn nữa để có thể tăng hiệu năng, độ ổn định của những thiết bị trong mạng và áp dụng cho mạng có số lượng nút lớn. Ngoài ra để tăng khoảng cách truyền, cần phát triển giao thức áp dụng cho mô hình mạng đa chặng để giảm số lượng Gateway trong mạng.

4.3 Kết luận

Qua kết quả của những thử nghiệm đã được tiến hành, chứng tỏ thuật toán đa truy nhập hoạt động tốt với số lượng nút nhỏ (3 nút), do điều kiện về số lượng thiết bị giới hạn nên em chưa xác định được kết quả hoạt động của thuật toán đa truy nhập trong mạng có số lượng nút lớn hơn. Ngoài ra, thuật toán đa truy nhập đã được thích hợp vào hệ thống BKRES-LoRa thành công.

Kết luận

Sản phẩm của đồ án "**Thiết kế, phát triển giao thức đa truy nhập LoRa, tích hợp vào mạng WSN giám sát môi trường**" đã được thử nghiệm và tích hợp vào hệ thống BKRES-LoRa. Hệ thống BKRES-LoRa chuẩn bị được thực nghiệm trong môi trường thực tế. Nhờ đó, thuật toán đa truy nhập có thể phát hiện những hạn chế còn tồn tại và sẽ được sửa chữa, cải thiện thêm. Trong tương lai, em tin hệ thống BKRES-LoRa sẽ được cải tiến ngày một tốt hơn để ngoài phục vụ cho lĩnh vực nghiên cứu, nó có thể trở thành một sản phẩm thương mại. Từ đó, hỗ trợ con người trong việc giám sát các thông số môi trường nói chung, cũng như giúp người nông dân trong quá trình nuôi tôm nói riêng.

Trong quá trình nghiên cứu, xây dựng và thiết kế sản phẩm, em đã vấp phải nhiều khó khăn. Nhưng nhờ có sự hướng dẫn tận tình của TS. Trần Quang Vinh cũng như sự giúp đỡ của các thành viên trong phòng nghiên cứu Sanslab đã giúp em hoàn thiện đồ án này. Qua đó, giúp em củng cố thêm những kiến thức về phần cứng, lập trình; những kỹ năng làm việc nhóm, tìm hiểu nghiên cứu những vấn đề mới; cũng như có được những kiến thức mới và trải nghiệm thực tế.

Tài liệu tham khảo

- [1] <https://www.i-scoop.eu/internet-of-things-guide/iot-network-lora-lorawan/>
truy nhập cuối cùng ngày 12/03/2018.
- [2] O. Khutsoane, B. Isong và A., "IoT Devieces and Appications based on LoRa/LoRaWAN", Industrial Electronics Society, IECON 2017 – 43rd Annual Conference of the IEEE, Beijing, China, 29 Oct.-1 Nov. 2017.
- [3] Guillaume Ferré, "Collision and Packet Loss Analysis in a LoRaWAN Network", 25th European Signal Processing Conference (EUSIPCO), 2017.
- [4] Alexandru Lavric, Valentin Popa, "Internet of Things and LoRaTM Low-Power Wide-Area Networks: A Survey", Signal, Cricuits and Systems (ISSCS), 2017 International Symposium on, Iasi, Romania, 13–14 July, 2017.
- [5] U. Noreen, A. Bounceur và L. Clavier, "A Study of LoRa Low Power and Wide Area Network Technology", 3rd International Conference on Advanced Technologies for Signal and Image Processing - ATSIP, Fez, Morroco, May 22-24, 2017.
- [6] Andrew S. Tanenbaum, David J. Wetherall, "*Computer Network*", Prentice Hall, 2010.
- [7] Module LoRa Ra-02 SX1278 datasheet (<https://www.en.ai-thinker.com>).
- [8] SX1276/77/78 datasheet (<https://www.semtech.com>).
- [9] STM32F1xC, STM32F1xD, STM32F1xE datasheet (<https://www.st.com>).
- [10] STM32Cube initialization code generator.

Phụ lục

Hàm main của Gateway

```
int main(void)
{
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();
    /* Configure the system clock */
    SystemClock_Config();
    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_SPI3_Init();
    MX_USART1_UART_Init();
    /* USER CODE BEGIN 2 */
    SX1278_Config();
    /* USER CODE END 2 */
    while (1)
    {
        uint8_t e = receivePacketTimeoutACK (&hspi3, 10000);
        if (!e){
            UART_print ("\nReceive msg");
            char msg[40] = {0};
            for (uint8_t i = 0; i < packet_received.length; i++){
                msg[i] = (char)packet_received.data[i];
            }
            int ID = packet_received.src;
            if (msg[1] == 'D'){
                char title[100];
                sprintf(title, "\nReceive Data form node %d: ", ID);
                UART_print(title);
                UART_print(msg);
            }
            else checkNode(ID, msg);
        }
    }
}
```

```
    }
    else UART_print ("\nDon't have msg");
}
}
```

Hàm kiểm tra bản tin xác thực của Gateway

```

void checkNode (int nodeAddr, char msg[]){
    char title[100];
    sprintf(title, "\nSend accepted Msg to Node %d", nodeAddr);
    char *response;
    uint8_t e;
    UART_print(title);
    if (msg[2] == 's' && msg[3] == 'a' && msg[4] == 'n' && msg[5] == 's' &&
        msg[6] == 'l' && msg[7] == 'a' && msg[8] == 'b'){
        response = "1";
        UART_print ("\nGateway: 1");
        arrNodeAddr[numberofNode++] = nodeAddr;
    }
    else{
        response = "0";
        UART_print ("\nGateway: 0");
    }
    uint8_t cnt;
    do{
        e = sendPacketTimeoutACKRetries(&hspi3, nodeAddr, response);
        cnt++;
        HAL_Delay (5);
    }
    while (e > 0 && cnt < 3);
}

```

Hàm main của nút

```
int main(void)
{
    /* MCU Configuration-----*/
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_SPI3_Init();
    MX_USART1_UART_Init();
    /* USER CODE BEGIN 2 */

    SX1278_Config();
    /* USER CODE END 2 */

    delay = NODE_ADDRESS * 10;
    requestJoinNetwork(GWAY_ADDRESS);

    while (1)
    {
        /* USER CODE END WHILE */

        if (joinedNetwork == 1)
            sendData(GWAY_ADDRESS);
        HAL_Delay(15000);
        /* USER CODE BEGIN 3 */
    }
}
```

Hàm yêu cầu tham gia mạng của nút

```

void requestJoinNetwork (uint8_t GWaddr)
{
    char msgRQ[20] = "@Rsanslab";
    uint8_t e;
    do{
        e = sendPacketTimeoutACK (&hspi3, GWaddr , msgRQ);
        if (e == 0){
            UART_print ("\nSend msg_request successful, waiting
msg_accept");
            uint8_t state = receivePacketTimeoutACK (&hspi3, 10000);
            char msgAccept[10] = {0};
            if (state == 0 && packet_received.src == GWAY_ADDRESS){
                for (uint8_t i = 0; i < packet_received.length; i++){
                    msgAccept[i] = (char) packet_received.data[i];
                }
                if (msgAccept[0] == '1'){
                    UART_print ("\nGateway accepted");
                    joinedNetwork = 1;
                }
                else{
                    UART_print ("\nGateway didn't accept");
                    joinedNetwork = 2;
                }
                //delay = NODE_ADDRESS * 10;
            }
        }
        else{
            UART_print ("\nSend msg_request nonsuccessful");
            //int x = rand() % 10;
            HAL_Delay (1000);
            //delay *= 2;
        }
    }
    while (e);
}

```

Hàm gửi dữ liệu của nút

```

void sendData (uint8_t GWAddr){
    char msgData[100] =
        "@DN045<~1Gq!J1,UaD2)R'IOf(0='UY5E$cWE>!!!!d!!!!g!!!!f!!!!c!!!!b!!!!>";
    //sprintf(msgData, "", counter);
    delay = 10 * NODE_ADDRESS;
    uint8_t e;
    uint8_t cnt = 0;
    do{
        e = sendPacketTimeoutACK(&hspi3, GWAddr, msgData);
        cnt_sent++;
        cnt++;
        if (e > 0){
            //int x = rand() % 10;
            HAL_Delay (delay);
            if (delay *2 < 2000) delay *= 2;
        }
        else
            counter++;
    }
    while(e && cnt < 5);
    char tmp[100];
    if (e == 0){
        sprintf(tmp, "\nNode %d :received %d, retries %d, deleted %d, sent
%d", NODE_ADDRESS, counter, cnt, cnt_huy_goi, cnt_sent);
        UART_print(tmp);
    }
    else
        cnt_huy_goi++;
}

```
