

ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP

ĐÁNH GIÁ MỘT SỐ THUẬT TOÁN ĐỊNH TUYẾN RPL CHO MẠNG TỒN HAO VÀ NĂNG LƯỢNG THẤP

PHẠM CÔNG MINH

minh.pc200688@sis.hust.edu.vn

Ngành KT Điều khiển & Tự động hóa

Chuyên ngành Điều khiển tự động

Giảng viên hướng dẫn: PGS. TS. Nguyễn Quốc Cường

Chữ ký của GVHD

Khoa:

Tự động hóa

Trường:

Điện - Điện tử

HÀ NỘI, 6/2024

**NHIỆM VỤ
ĐỒ ÁN TỐT NGHIỆP**

Họ và tên sinh viên: Phạm Công Minh

Mã số sinh viên: 20200688

Khóa: 65

Trường: Điện - Điện tử

Ngành: KT ĐK & TĐH

1. Tên đề tài:

Đánh giá một số thuật toán định tuyến RPL cho mạng tổn hao và năng lượng thấp.

2. Nội dung đề tài:

Đánh giá một số thuật toán định tuyến cải tiến dựa trên thuật toán RPL cho mạng cảm biến không dây năng lượng thấp và tổn hao về mặt tiết kiệm năng lượng, truyền tin hiệu quả, Các thuật toán được đánh giá bằng mô phỏng và thực nghiệm. Đề tài cũng giới thiệu mạng 6LowPan, hệ điều hành mã nguồn mở Contiki OS, triển khai trên kit phát triển LaunchPad CC2650 của hãng TI.

3. Cán bộ hướng dẫn: PGS. TS. Nguyễn Quốc Cường

4. Thời gian giao đề tài: 8/3/2024

5. Thời gian hoàn thành: 26/6/2024

Ngày..... tháng năm 2024

LÃNH ĐẠO BỘ MÔN

CÁN BỘ HƯỚNG DẪN

SINH VIÊN THỰC HIỆN

(Ký và ghi rõ họ tên)

Lời cảm ơn

Lời đầu tiên, em xin bày tỏ lòng biết ơn đến cha mẹ em, những người đã tạo điều kiện để em được học tập và trưởng thành. Trong quá trình học tập tại Đại học Bách Khoa Hà Nội, em nhận được rất nhiều sự quan tâm, giúp đỡ, chia sẻ từ thầy cô, gia đình và bạn bè. Em xin chân thành cảm ơn thầy PGS.TS Nguyễn Quốc Cường là người đã trực tiếp định hướng, hướng dẫn em thực hiện đề tài này. Dưới sự hướng dẫn nhiệt tình của thầy trong suốt thời gian học tập và làm đồ án, em đã học hỏi được nhiều kinh nghiệm trong cả công việc, cuộc sống, các hướng phát triển trong tương lai, cũng như hoàn thành tốt đồ án tốt nghiệp. Trong quá trình thực hiện đồ án còn nhiều hạn chế và sai sót, em mong nhận được ý kiến đóng góp của quý Thầy, Cô để kiến thức của em được hoàn thiện hơn.

Tóm tắt nội dung đồ án

Các mạng cảm biến không dây IoT đã và đang được triển khai rộng rãi như giám sát môi trường, giám sát từ xa các thiết bị công nghiệp, đèn giao thông thông minh, hệ thống cảm biến trong nhà thông minh, Các module cảm biến ngày càng trở nên rẻ, nhỏ gọn và tiêu thụ năng lượng thấp. Vì vậy việc tối ưu hoạt động của mạng cảm biến không dây, đặc biệt là vấn đề định tuyến cần được chú ý và nghiên cứu. Vì vậy em xin chọn đề tài: “ **Đánh giá một số thuật toán định tuyến RPL cho mạng tổn hao và năng lượng thấp**” làm đề tài tốt nghiệp. Đề tài này đặt ra yêu cầu: Xây dựng các mạng cảm biến trong phần mềm mô phỏng và thử nghiệm lại trên phần cứng để đánh giá hiệu quả của một số thuật toán định tuyến cải tiến được phát triển từ thuật toán định tuyến RPL; các nút cảm biến được lập trình bằng hệ điều hành mã nguồn mở Contiki OS, ứng dụng công nghệ mạng 6LowPan, sử dụng chuẩn không dây IEEE 192.15.4. Mô phỏng được thực hiện trên phần mềm Cooja. Phần cứng sử dụng là kit phát triển LaunchPad CC2650 của hãng TI.

Các kết quả đồ án cơ bản đáp ứng được mục tiêu đề ra, các kết quả mô phỏng và thực nghiệm đã cho thấy các thuật toán được đánh giá mang lại cải thiện đáng kể về mặt sử dụng năng lượng hiệu quả, hiệu suất truyền tin so với thuật toán tiêu chuẩn. Tuy nhiên do hạn chế về mặt phần cứng nên mạng cảm biến được xây dựng trong thực nghiệm còn nhỏ, các thử nghiệm thực tế còn tương đối ít. Các kết quả thu được từ đề tài có thể được dùng như một nguồn tham khảo để người đọc có thể xây dựng mạng cảm biến không dây của mình, hoặc phát triển các thuật toán định tuyến mới.

Các kiến thức và kỹ năng đã đạt được:

- Hoạt động của thuật toán định tuyến RPL
- Lập trình mạng cảm biến ứng dụng hệ điều hành mã nguồn mở Contiki OS
- Xây dựng mạng cảm biến ứng dụng mạng 6LowPan, chuẩn truyền thông không dây IEEE 192.15.4.

Sinh viên thực hiện

Ký và ghi rõ họ tên

LỜI MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, Internet of Things (IoT) đang dần khẳng định vai trò quan trọng của mình trong việc thay đổi cách chúng ta sống và làm việc. Các mạng cảm biến không dây (Wireless Sensor Networks - WSN) là một thành phần thiết yếu trong hệ sinh thái IoT, được triển khai rộng rãi trong nhiều lĩnh vực như giám sát môi trường, quản lý năng lượng, giám sát từ xa các thiết bị công nghiệp, đèn giao thông thông minh, và các hệ thống nhà thông minh. Sự phát triển mạnh mẽ của công nghệ cảm biến đã giúp cho các module cảm biến ngày càng trở nên rẻ hơn, nhỏ gọn hơn và tiêu thụ năng lượng thấp hơn, từ đó mở ra nhiều cơ hội ứng dụng mới.

Quy mô mạng cảm biến ngày càng mở rộng và đa dạng, vì vậy yêu cầu nghiên cứu tối ưu hoạt động của mạng trở nên thiết yếu. Trong một mạng cảm biến không dây, các nút cảm biến thường hoạt động bằng pin và có thể được đặt ở những vị trí khó tiếp cận, nên vấn đề tiết kiệm năng lượng và tối ưu hóa hoạt động của mạng là vô cùng quan trọng. Một trong những thách thức lớn nhất của mạng cảm biến không dây là việc định tuyến dữ liệu một cách hiệu quả. Do đó, các nghiên cứu về thuật toán định tuyến luôn nhận được sự quan tâm đặc biệt từ cộng đồng khoa học và kỹ thuật.

Trong bối cảnh đó, em đã chọn đề tài **“Đánh giá một số thuật toán định tuyến RPL cho mạng tổn hao và năng lượng thấp”** làm đề tài tốt nghiệp của mình. Đề tài này hướng đến việc xây dựng các mạng cảm biến trong phần mềm mô phỏng và thử nghiệm lại trên phần cứng nhằm đánh giá hiệu quả của một số thuật toán định tuyến cải tiến được phát triển từ thuật toán định tuyến RPL (Routing Protocol for Low-Power and Lossy Networks). Cụ thể, các nút cảm biến sẽ được lập trình bằng hệ điều hành mã nguồn mở Contiki OS, sử dụng công nghệ mạng 6LowPan và chuẩn không dây IEEE 802.15.4. Quá trình mô phỏng sẽ được thực hiện trên phần mềm Cooja, và phần cứng sử dụng là kit phát triển LaunchPad CC2650 của hãng Texas Instruments (TI).

Yêu cầu đề tài đặt ra các nhiệm vụ bao gồm việc phân tích, thiết kế và thực hiện các thuật toán định tuyến, sau đó đánh giá hiệu quả của chúng thông qua các tiêu chí như độ tin cậy, hiệu suất sử dụng năng lượng. Mục tiêu cuối cùng là đánh giá các giải pháp định tuyến tối ưu, giúp nâng cao hiệu quả hoạt động của mạng cảm biến không dây trong điều kiện thực tế, đặc biệt là trong môi trường có nhiều tổn hao và mất mát. Thông qua đề tài này, em hy vọng có thể đóng góp vào việc tối ưu hóa các mạng cảm biến không dây, giúp chúng hoạt động hiệu quả hơn, từ đó thúc đẩy sự phát triển của các ứng dụng IoT trong thực tiễn. Việc nghiên cứu và đánh giá các thuật toán định tuyến cho mạng cảm biến không dây sẽ đóng góp vào sự phát triển và tối ưu hóa các hệ thống IoT, từ đó cải thiện hiệu suất và độ tin cậy của các ứng dụng IoT trong thực tế. Các kết quả của đề tài có thể được dùng làm tài liệu tham khảo để người đọc có thể lựa chọn công nghệ để xây dựng mạng cảm biến không dây của mình, hoặc làm nền tảng cho các nghiên cứu tiếp theo, góp phần thúc đẩy việc cải tiến và phát triển các thuật toán định tuyến mới, phù hợp hơn với các yêu cầu thực tiễn ngày càng cao của mạng cảm biến không dây.

Với mục tiêu đề ra, bản báo cáo đồ án được trình bày theo cấu trúc như sau:

Chương 1 cung cấp những kiến thức tổng quan về mạng cảm biến, bao gồm khái niệm, cấu trúc, đặc điểm, ứng dụng. Đồng thời chương 1 cũng đặt ra bài toán về vấn đề định tuyến trong mạng cảm biến không dây, đặc biệt là mạng tổn hao và năng lượng thấp.

Xuất phát từ những yêu cầu của bài toán định tuyến trong chương 1, chương 2 giới thiệu đến người đọc một thuật toán định tuyến hiệu quả được sử dụng phổ biến trong các mạng cảm biến không dây hiện đại. Qua chương này, em hi vọng người đọc sẽ có được hiểu biết về cách thuật toán RPL hoạt động, ưu điểm cũng như nhược điểm còn tồn tại của nó. Một số nghiên cứu cải tiến thuật toán RPL cũng được trình bày.

Chương 3 trình bày về 3 thuật toán định tuyến cải tiến dựa trên thuật toán RPL sẽ được đánh giá trong đồ án này. Qua chương này, người đọc sẽ hiểu được cách hoạt động của thuật toán định tuyến và những mục tiêu mà mỗi thuật toán nhắm đến.

Chương 4 trình bày về phương pháp được sử dụng để đánh giá các thuật toán định tuyến, các tiêu chí đánh giá được đề ra, đồng thời giới thiệu các công cụ phần cứng và phần mềm được sử dụng để xây dựng các mô phỏng và thử nghiệm trong đồ án.

Chương 5 và chương 6 làm rõ quá trình thử nghiệm thuật toán bằng mô phỏng và thực nghiệm, sau đó trình bày các kết quả đạt được cùng những đánh giá từ kết quả thu được.

Cuối cùng, phần Kết luận tổng kết lại các kết quả thu được, so sánh kết quả đồ án với mục tiêu đề ra, chỉ ra các hạn chế còn tồn tại của đồ án và hướng phát triển trong tương lai.

Em xin chân thành cảm ơn các thầy cô và các bạn đã hỗ trợ, đồng hành cùng em trong suốt quá trình nghiên cứu và thực hiện đề tài này. Hy vọng rằng những kết quả đạt được sẽ mang lại những giá trị thiết thực và có ý nghĩa cho cộng đồng nghiên cứu cũng như ứng dụng trong thực tế.

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỊNH TUYẾN TRONG MẠNG CẢM BIẾN KHÔNG DÂY TỔN HAO VÀ NĂNG LƯỢNG THẤP 1

1.1	Tổng quan về mạng cảm biến không dây	1
1.1.1	Mạng cảm biến không dây là gì?	1
1.1.2	Cấu trúc mạng cảm biến không dây	1
1.1.3	Đặc điểm mạng cảm biến không dây	2
1.1.4	Ứng dụng của mạng cảm biến không dây	3
1.2	Định tuyến trong mạng cảm biến không dây tổn hao và năng lượng thấp	4
1.2.1	Định tuyến là gì?	4
1.2.2	Mạng cảm biến không dây tổn hao và năng lượng thấp	4
1.2.3	Các thách thức với quá trình định tuyến cho mạng cảm biến không dây tổn hao và năng lượng thấp	5
1.3	Kết luận chương	6

CHƯƠNG 2. THUẬT TOÁN ĐỊNH TUYẾN RPL 7

2.1	Giới thiệu thuật toán định tuyến RPL	7
2.2	Các khái niệm quan trọng	7
2.3	Các loại bản tin điều khiển.....	9
2.3.1	Bản tin DIS.....	10
2.3.2	Bản tin DIO	11
2.3.3	Bản tin DAO.....	12
2.3.4	Bản tin DAO-ACK.....	13
2.4	Hàm mục tiêu	14
2.5	Hoạt động định tuyến sử dụng thuật toán RPL	16
2.5.1	Xây dựng DODAG.....	16
2.5.2	Định tuyến hướng lên.....	18
2.5.3	Định tuyến hướng xuống.....	19
2.6	Tại sao RPL phù hợp với mạng tổn hao và năng lượng thấp.....	20
2.7	Các hướng cải tiến.....	21
2.7.1	Cải tiến hàm mục tiêu (OF).....	21
2.7.2	Cơ chế downward.....	23
2.7.3	Tính bảo mật.....	24
2.8	Kết luận chương	24

CHƯƠNG 3. MỘT SỐ THUẬT TOÁN CẢI TIẾN DỰA TRÊN RPL	25
3.1 Thuật toán RPL cải tiến dựa trên năng lượng còn dư	25
3.1.1 Ước tính năng lượng tiêu thụ của nút.....	25
3.1.2 Chi phí đường đi.....	26
3.2 Thuật toán RPL cải tiến dựa trên Radio Duty Cycle	26
3.2.1 Ước tính trạng thái tải dựa trên radio duty cycle	27
3.2.2 Chi phí đường đi.....	27
3.3 Thuật toán RPL cải tiến dựa trên tỉ lệ sử dụng hàng đợi tin nhắn	27
3.3.1 Hàng đợi tin nhắn	27
3.3.2 Chi phí đường đi.....	28
3.4 Kết luận chương	29
CHƯƠNG 4. PHƯƠNG ÁN THỰC HIỆN ĐỀ TÀI.....	30
4.1 Phạm vi đề tài.....	30
4.2 Phương án thực hiện.....	30
4.2.1 Sơ đồ tổng quan hệ thống.....	30
4.2.2 Ngăn xếp mạng (network stack)	31
4.2.3 Các thông số đánh giá	33
4.3 Công cụ sử dụng.....	33
4.3.1 Phần mềm	34
4.3.2 Phần cứng	39
4.4 Hạn chế.....	41
4.5 Kết luận chương	41
CHƯƠNG 5. MÔ PHỎNG MỘT SỐ THUẬT TOÁN ĐỊNH TUYẾN.....	42
5.1 Kịch bản mô phỏng	42
5.1.1 Cấu trúc mạng	42
5.1.2 Các kịch bản mô phỏng.....	43
5.1.3 Tham số mô phỏng.....	43
5.2 Kết quả mô phỏng	44
5.2.1 Kịch bản lưu lượng thấp.....	44
5.2.2 Kịch bản lưu lượng cao	48
5.3 Kết luận chương	53
CHƯƠNG 6. THỬ NGHIỆM THỰC TẾ	54
6.1 Tham số thử nghiệm.....	54
6.2 Kết quả thử nghiệm.....	54

6.3	Kết luận chương	58
	KẾT LUẬN	59
1.	Kết luận	59
2.	Hướng phát triển của đồ án trong tương lai	59
	TÀI LIỆU THAM KHẢO	60

DANH MỤC HÌNH VẼ

Hình 1.1: Cấu trúc mạng cảm biến không dây điển hình.....	1
Hình 1.2: Ứng dụng mạng cảm biến không dây [1].....	4
Hình 1.3: Quá trình truyền tin qua nhiều nút trung gian.....	4
Hình 2.1: Cấu trúc DODAG.....	8
Hình 2.2: Bản tin điều khiển RPL [2]	10
Hình 2.3: Cấu trúc bản tin DIS[2]	10
Hình 2.4: Cấu trúc bản tin DIO [2]	11
Hình 2.5: Cấu trúc bản tin DAO[2].....	12
Hình 2.6: Cấu trúc bản tin DAO-ACK	13
Hình 2.7: Xây dựng DODAG	16
Hình 2.8: Hoạt động bản tin DIO.....	17
Hình 2.9: Bản tin DIS.....	18
Hình 2.10: Định tuyến upward.....	18
Hình 2.11: Bảng định tuyến chế độ lưu trữ.....	19
Hình 2.12: Bảng định tuyến chế độ không lưu trữ.....	20
Hình 3.1: Hoạt động của hàng đợi tin nhắn	28
Hình 4.1: Sơ đồ hệ thống	31
Hình 4.2: Logo contiki-ng.....	34
Hình 4.3: Giao diện Cooja Simulator.....	36
Hình 4.4: Cửa sổ network Cooja.....	37
Hình 4.5: Cửa sổ Timeline	37
Hình 4.6: Cửa sổ Mote output của Cooja.....	37
Hình 4.7: Tab Serial Data.....	38
Hình 4.8: Tab năng lượng còn lại	39
Hình 4.9: Kit phát triển LAUNCHPAD-CC2650.....	39
Hình 4.10: Mặt trước của kit LAUNCHPAD-CC2650	40
Hình 4.11: Kit cc2650 cùng với pin.....	41
Hình 5.1 Cấu trúc mạng mô phỏng	42
Hình 5.2: Sơ đồ bố trí cảm biến	43
Hình 5.3: Phân bố năng lượng khi chỉ sử dụng ETX.....	44
Hình 5.4 Phân bố năng lượng còn lại khi sử dụng ETX kết hợp năng lượng còn dư	44
Hình 5.5 Phân bố năng lượng còn lại khi sử dụng ETX-RDC	45
Hình 5.6 Phân bố năng lượng còn lại khi sử dụng ETX-QU	45
Hình 5.7: Tổng năng lượng còn lại	47
Hình 5.8: Tỷ lệ truyền gói tin thành công	47

Hình 5.9 Số bước nhảy trung bình	48
Hình 5.10 Độ trễ trung bình	48
Hình 5.11 Phân bố năng lượng còn lại khi chỉ sử dụng ETX với lưu lượng lớn .	49
Hình 5.12 Phân bố năng lượng còn lại khi sử dụng ETX kết hợp năng lượng còn dư với lưu lượng lớn	49
Hình 5.13 Phân bố năng lượng còn lại khi sử dụng ETX-RDC khi lưu lượng lớn	50
Hình 5.14 Phân bố năng lượng còn lại khi sử dụng ETX-QU khi lưu lượng lớn	50
Hình 5.15: Tổng năng lượng còn lại	51
Hình 5.16: Tỷ lệ truyền gói tin thành công	51
Hình 5.17: Số gói tin mất do tràn queue	52
Hình 5.18: Số bước nhảy trung bình	52
Hình 5.19: Độ trễ trung bình	53
Hình 6.1: Giao diện theo dõi	54
Hình 6.2: Theo dõi năng lượng theo thời gian thực	55
Hình 6.3: Phân bố năng lượng còn lại của RPL-ETX thực nghiệm	55
Hình 6.4: Phân bố năng lượng còn lại của ETX-EE	56
Hình 6.5: Phân bố năng lượng còn lại của ETX-RDC	56
Hình 6.6: Phân bố năng lượng còn lại của ETX-QU	57
Hình 6.7: Tổng năng lượng còn lại thực nghiệm	57
Hình 6.8: Tỷ lệ truyền gói tin thành công thực nghiệm	58

DANH MỤC BẢNG BIỂU

Bảng 4.1: Ngăn xếp mạng được sử dụng	31
Bảng 4.2: Các giao thức hỗ trợ bởi Contiki OS	35
Bảng 4.3: Thông số kỹ thuật kit LAUNCHPAD-CC2650	40
Bảng 5.1: Bảng tham số mô phỏng	43
Bảng 5.2: Số bản tin được trung chuyển qua từng nút.....	46
Bảng 6.1: Dòng tiêu thụ trong từng chế độ của cc2650.....	54

DANH MỤC CÁC TỪ VIẾT TẮT

Chữ viết tắt	Nguyên nghĩa
WSNs	Wireless sensor networks
LLNs	Lowpower and lossy networks
RPL	Routing protocol for Low-power and Lossy Networks
DODAG	Destination Oriented Directed Acyclic Graph
DIS	DODAG Information Solicitation
DAO	Destination Advertisement Object
DIO	DODAG Information Object
DAO-ACK	Destination Advertisement Object Acknowledgment
OF	Objective Function
ETX	Expected Transmission Count
UDP	User Datagram Protocol
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks

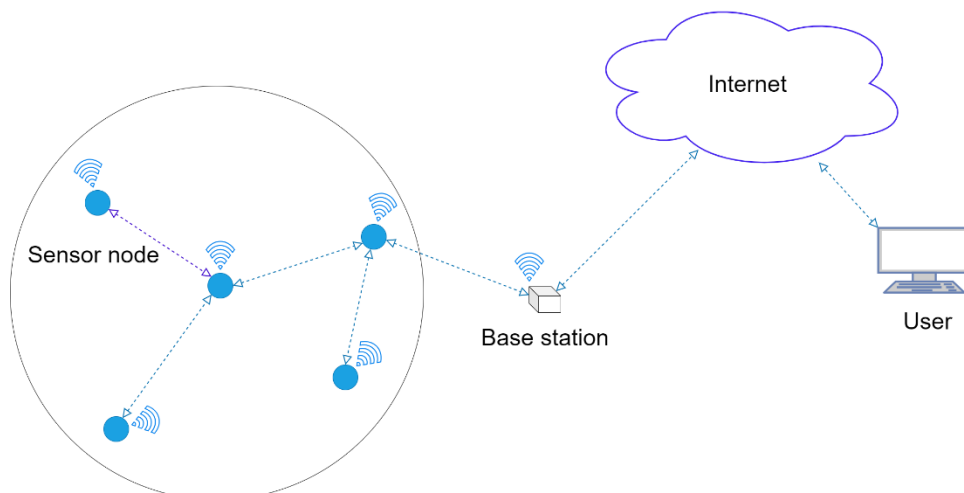
CHƯƠNG 1. TỔNG QUAN VỀ ĐỊNH TUYẾN TRONG MẠNG CẢM BIẾN KHÔNG DÂY TỒN HAO VÀ NĂNG LƯỢNG THẤP

Chương 1 sẽ cung cấp những kiến thức tổng quan về mạng cảm biến, bao gồm khái niệm, cấu trúc, đặc điểm, ứng dụng của mạng cảm biến không dây trong hệ thống IoT hiện đại. Đồng thời chương này cũng đặt ra bài toán về vấn đề định tuyến trong mạng cảm biến không dây, đặc biệt là mạng tồn hao và năng lượng thấp.

1.1 Tổng quan về mạng cảm biến không dây

1.1.1 Mạng cảm biến không dây là gì?

Mạng cảm biến không dây (Wireless Sensor Networks) là một hệ thống mạng gồm các thiết bị cảm biến được phân tán trong không gian để thu thập và truyền dữ liệu từ môi trường xung quanh một cách tự động. Các thiết bị cảm biến này giao tiếp không dây với nhau và thường có khả năng tự tổ chức thành một mạng lưới để chuyển tiếp dữ liệu đến các điểm thu thập và xử lý thông tin.



Hình 1.1: Cấu trúc mạng cảm biến không dây điển hình

1.1.2 Cấu trúc mạng cảm biến không dây

Cấu trúc của một mạng cảm biến không dây thường bao gồm các thành phần cơ bản sau:

- **Nút Cảm Biến (Sensor Node):** Là thành phần cơ bản của mạng, thường được triển khai hàng loạt trên một khu vực để thu thập dữ liệu từ môi trường xung quanh. Mỗi nút cảm biến được trang bị các cảm biến để đo lường các thông số như nhiệt độ, độ ẩm, ánh sáng, độ rung, v.v. Nút cảm biến thường có khả năng xử lý cơ bản và gửi dữ liệu về nút xử lý hoặc trung tâm.
- **Nút Xử Lý (Processing Node):** Nút này thường nằm ở các vị trí chiến lược trong mạng để tiếp nhận dữ liệu từ các nút cảm biến. Có khả năng xử lý dữ liệu phức tạp hơn so với các nút cảm biến, thường được trang bị các tài nguyên tính toán và bộ nhớ lớn hơn. Có thể thực hiện các tính toán, lọc dữ liệu, và tổng hợp thông tin trước khi gửi dữ liệu đến trung tâm.

- *Nút trung Tâm (Base Station hoặc Sink Node)*: Là nút cuối cùng trong mạng, thường là một trạm cơ sở cố định hoặc di động. Trung tâm thu thập dữ liệu từ các nút cảm biến và nút xử lý, và có thể gửi dữ liệu đến các hệ thống ngoại vi như máy chủ hoặc hệ thống giám sát. Thường có nguồn năng lượng ổn định và có kết nối mạng liên tục.
- *Kết Nối Không Dây*: Các nút trong mạng kết nối với nhau và với trung tâm thông qua kết nối không dây như Wi-Fi, Bluetooth, Zigbee, hoặc các tiêu chuẩn kết nối không dây khác. Tùy thuộc vào ứng dụng cụ thể, kết nối có thể được thiết kế để tiết kiệm năng lượng hoặc tối ưu hóa hiệu suất truyền thông.
- *Giao Thức Mạng*: Cung cấp các quy tắc và quy trình cho việc truyền thông và quản lý mạng, bao gồm định tuyến dữ liệu, quản lý năng lượng, phát hiện lỗi, và điều chỉnh mạng. Các giao thức phổ biến bao gồm LEACH (Low-Energy Adaptive Clustering Hierarchy), Zigbee, và IEEE 802.15.4.
- *Nguồn Năng Lượng*: Đặc điểm quan trọng của mạng cảm biến không dây là các nút thường hoạt động trên pin hoặc nguồn năng lượng hạn chế. Việc quản lý năng lượng là một yếu tố quan trọng để đảm bảo tuổi thọ và hiệu suất của mạng.

Cấu trúc của mạng cảm biến không dây thường phản ánh mục tiêu ứng dụng cụ thể và yêu cầu về hiệu suất, tiết kiệm năng lượng, và khả năng mở rộng của hệ thống.

1.1.3 Đặc điểm mạng cảm biến không dây

Mạng cảm biến không dây có những đặc điểm riêng biệt so với các loại mạng khác, điều này phản ánh vào mục tiêu sử dụng và yêu cầu kỹ thuật của chúng. Dưới đây là một số đặc điểm chính của mạng cảm biến không dây:

- *Phân tán và đa nút*: WSN thường bao gồm một lượng lớn các nút cảm biến phân tán trên một khu vực rộng. Mỗi nút cảm biến thường nhỏ gọn, đơn giản, và các nút có khả năng tự tổ chức thành một mạng.
- *Môi trường động và không đồng nhất*: Các nút cảm biến thường được triển khai trong môi trường đa dạng, từ môi trường ngoại trời đến môi trường trong nhà. Môi trường có thể thay đổi đột ngột và không đồng nhất, đòi hỏi sự linh hoạt và khả năng thích ứng của mạng.
- *Năng lượng thấp*: Một trong những đặc điểm quan trọng nhất của WSN là năng lượng hạn chế. Các nút thường hoạt động trên pin hoặc nguồn năng lượng tái tạo công suất thấp (ví dụ pin mặt trời). Do đó, việc tiết kiệm năng lượng là một ưu tiên cao, thông qua việc sử dụng kỹ thuật sleep mode, điều chỉnh công suất truyền, và quản lý năng lượng thông minh.
- *Tính tổn hao*: Do đặc điểm phân tán trên một không gian rộng lớn, khoảng cách giữa các nút lớn, các nút có công suất phát tương đối nhỏ, cộng với các tác động của môi trường, khiến mạng cảm biến không dây có tỉ lệ mất gói tin cao.

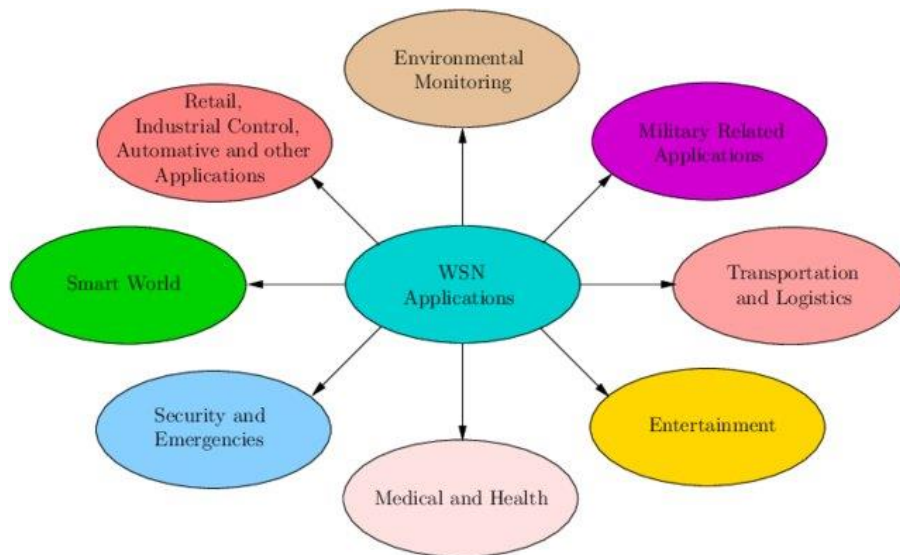
- *Tính linh hoạt và mở rộng*: Mạng cảm biến không dây thường có khả năng mở rộng linh hoạt, cho phép thêm hoặc loại bỏ các nút dễ dàng. Nó cũng cần có khả năng tự hồi phục khi có sự cố xảy ra trong mạng.
- *Bảo mật và quyền riêng tư*: Vì dữ liệu thu thập từ mạng cảm biến thường là nhạy cảm, bảo mật và quyền riêng tư là một vấn đề quan trọng. Cần triển khai các biện pháp bảo mật như mã hóa dữ liệu, xác thực, và cơ chế phát hiện xâm nhập.

Những đặc điểm này định hình cách thức thiết kế, triển khai và quản lý mạng cảm biến không dây để đáp ứng được yêu cầu cụ thể của từng ứng dụng và môi trường sử dụng.

1.1.4 Ứng dụng của mạng cảm biến không dây

Mạng cảm biến không dây có rất nhiều ứng dụng trong nhiều lĩnh vực khác nhau nhờ vào khả năng thu thập dữ liệu từ môi trường xung quanh một cách liên tục và phân tán. Dưới đây là một số ứng dụng phổ biến của WSN:

- *Y tế và chăm sóc sức khỏe*: WSN có thể theo dõi các dấu hiệu y tế như nhịp tim, huyết áp, nồng độ glucose trong máu, và đưa ra cảnh báo khi có sự biến đổi. WSN giúp theo dõi vị trí và trạng thái của bệnh nhân, thiết bị y tế, và nhân viên y tế trong các cơ sở y tế.
- *Nông nghiệp*: Mạng lưới cảm biến như cảm biến đo độ ẩm đất, đo ánh sáng được sử dụng để hỗ trợ người nông dân giám sát, quản lý môi trường nông trại của mình.
- *Môi trường và giám sát tự nhiên*: Ứng dụng giám sát chất lượng không khí: WSN có thể đo lường các chất lượng không khí như ô nhiễm không khí, nồng độ khí CO₂, khí ozone, v.v. Đồng thời, mạng cảm biến không dây có thể ứng dụng để theo dõi nước và môi trường nước: WSN có thể theo dõi chất lượng nước, mực nước, và sự di chuyển của các hồ, sông, và đại dương. WSN có thể giúp phát hiện cháy rừng, theo dõi sự di chuyển của động vật hoặc đo lường độ ẩm đất đai.
- *Quản lý năng lượng và tài nguyên*: WSN giúp theo dõi và tối ưu hóa sử dụng năng lượng trong các nhà máy, tòa nhà, và hệ thống điện.
- *Công nghiệp và tự động hóa*: Mạng cảm biến được sử dụng để giám sát các máy móc, dây chuyền sản xuất, và điều khiển quy trình sản xuất trong các nhà máy. WSN có thể theo dõi điều kiện làm việc như nhiệt độ, độ ẩm, và khí độc trong môi trường công nghiệp, đảm bảo điều kiện an toàn lao động.
- *Đô thị thông minh và giao thông*: WSN có thể giúp giám sát tình trạng giao thông, đo lường mật độ xe cộ, và cung cấp thông tin giao thông thời gian thực cho người dùng thông qua hệ thống camera thông minh. WSN cũng có thể được sử dụng để giám sát và quản lý các dịch vụ công cộng như đèn đường, hệ thống thoát nước, và quản lý rác thải.



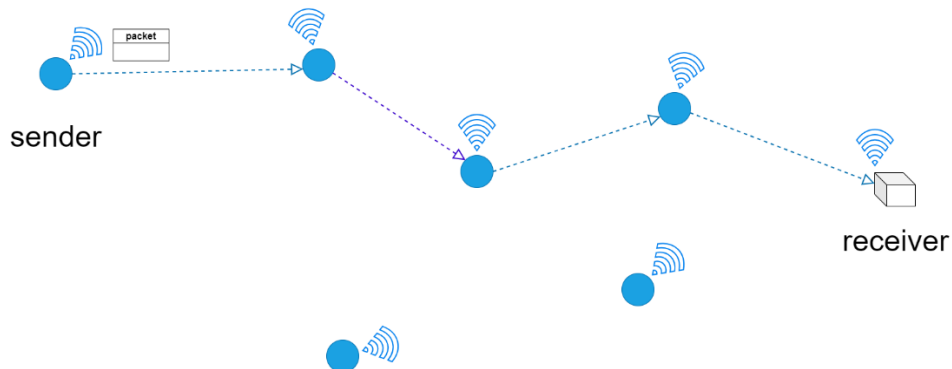
Hình 1.2: Ứng dụng mạng cảm biến không dây [1]

Tùy thuộc vào mục đích sử dụng cụ thể, mạng cảm biến không dây có thể được tùy chỉnh và triển khai để đáp ứng nhu cầu đặc biệt của từng ứng dụng.

1.2 Định tuyến trong mạng cảm biến không dây tổn hao và năng lượng thấp

1.2.1 Định tuyến là gì?

Trong mạng cảm biến không dây, các nút thường được phân bố trên diện tích rộng, truyền thông qua giao tiếp không dây. Do các nút thường hoạt động trên nguồn điện là pin nên có công suất phát sóng điện tử nhỏ, tầm hoạt động ngắn. Vì vậy dữ liệu không thể truyền trực tiếp từ các nút tới nút trung tâm mà phải thông qua các nút trung gian.



Hình 1.3: Quá trình truyền tin qua nhiều nút trung gian

Trong mạng cảm biến không dây, định tuyến (Routing) là quá trình xác định và chọn lựa đường dẫn cho dữ liệu từ các nút cảm biến đến đích, thường là trạm cơ sở (base station) hoặc nút xử lý. Mục tiêu của định tuyến là đảm bảo dữ liệu được truyền đến đúng đích mà không gây lãng phí năng lượng và tối ưu hóa hiệu suất mạng.

1.2.2 Mạng cảm biến không dây tổn hao và năng lượng thấp

Mạng cảm biến không dây tổn hao và năng lượng thấp (Low-Power and Lossy Networks) là một loại mạng cảm biến không dây được thiết kế để hoạt động trong các điều kiện với tài nguyên hạn chế và môi trường truyền dẫn dễ bị ảnh hưởng

bởi nhiều. LLNs thường bao gồm các thiết bị cảm biến có khả năng giao tiếp không dây, nhưng có hạn chế về năng lượng (thường chạy bằng pin) và khả năng xử lý. Các đặc điểm chính của mạng cảm biến không dây tổn hao và năng lượng thấp bao gồm:

- Tiêu thụ năng lượng thấp: Các thiết bị trong mạng LLNs được thiết kế để tiêu thụ năng lượng thấp nhằm kéo dài tuổi thọ pin, thường sử dụng các cơ chế tiết kiệm năng lượng như chế độ ngủ.
- Môi trường tổn hao: LLNs hoạt động trong môi trường có thể gây ra mất mát dữ liệu do nhiễu sóng, sự thay đổi điều kiện môi trường, hoặc các vật cản vật lý. Do đó, các giao thức truyền thông phải linh hoạt và có khả năng xử lý lỗi cao.
- Băng thông thấp: Để tiết kiệm năng lượng và tài nguyên, các thiết bị LLNs thường truyền dữ liệu ở tốc độ thấp. Điều này yêu cầu các giao thức phải hiệu quả về mặt băng thông.
- Tài nguyên hạn chế: Các thiết bị trong LLNs thường có khả năng xử lý, bộ nhớ và dung lượng lưu trữ hạn chế. Giao thức và thuật toán phải được thiết kế để hoạt động hiệu quả trong các điều kiện này.
- Ứng dụng đa dạng: LLNs được sử dụng trong nhiều lĩnh vực khác nhau như giám sát môi trường, hệ thống nhà thông minh, theo dõi sức khỏe, nông nghiệp thông minh, và các ứng dụng công nghiệp.

1.2.3 Các thách thức với quá trình định tuyến cho mạng cảm biến không dây tổn hao và năng lượng thấp

Do tính chất phân bố rộng và tài nguyên cũng như năng lượng hạn chế của các nút cảm biến mà quá trình định tuyến gói tin truyền trong mạng là một phần không thể thiếu và phải đối mặt với các thách thức sau:

Định tuyến trong mạng cảm biến không dây đối mặt với nhiều thách thức đặc biệt do tính đặc thù của mạng. Dưới đây là một số thách thức chính mà các thuật toán định tuyến cần phải đối mặt trong WSN:

- *Hạn chế về năng lượng*: Các nút cảm biến thường hoạt động trên nguồn năng lượng có hạn như pin hoặc năng lượng mặt trời. Việc truyền dẫn dữ liệu tiêu tốn rất nhiều năng lượng, và việc chọn đường dẫn tối ưu để giảm tiêu thụ năng lượng là rất quan trọng. Để khắc phục điều này, các nhà phát triển đã đề xuất thuật toán tiết kiệm năng lượng, tối ưu hóa định tuyến để giảm số lần truyền dữ liệu, sử dụng kỹ thuật sleep mode để tiết kiệm năng lượng khi không hoạt động.
- *Cấu trúc mạng thay đổi*: Môi trường triển khai của WSN thường thay đổi do vị trí của các nút, sự di chuyển của nút, và điều kiện môi trường. Điều này tạo ra thách thức trong việc duy trì đường dẫn ổn định và hiệu quả.
- *Băng thông hạn chế*: WSN thường có băng thông hạn chế do sự giới hạn về tần số truyền thông và nhu cầu chia sẻ băng thông giữa các nút. Hạn chế này yêu cầu thuật toán định tuyến phải có ít bản tin điều khiển, bản tin có kích thước nhỏ, tần suất gửi thấp để không làm tắc nghẽn mạng.

- *Độ trễ và độ ổn định*: Độ trễ trong việc truyền dẫn dữ liệu và sự không ổn định trong kết nối không dây có thể ảnh hưởng đến hiệu suất mạng.
- *Quản lý tài nguyên*: Sử dụng hiệu quả tài nguyên như bộ nhớ, xử lý, và băng thông để tối ưu hóa hoạt động của mạng.
- *Tính linh hoạt và khả năng mở rộng*: Đây là yêu cầu quan trọng trong việc xây dựng WSN. Mạng cảm biến cần có khả năng linh hoạt và mở rộng để thích ứng với sự thay đổi trong môi trường và nhu cầu mở rộng. Các nút có thể tham gia vào mạng mà không cần thay đổi toàn bộ mạng hay cần sự can thiệp của con người.
- *Bảo mật và quyền riêng tư*: Dữ liệu trong mạng cảm biến có thể là nhạy cảm và cần được bảo vệ khỏi việc đánh cắp hoặc tấn công. Mạng cần cân bằng giữa tính linh hoạt (các nút mới có thể tham gia dễ dàng) và tính bảo mật (ngăn chặn các nút giả dạng tham gia mạng để đánh cắp thông tin, phá hoại).

Đối mặt với những thách thức này, các nhà nghiên cứu và kỹ sư đã phát triển và tối ưu hóa các thuật toán định tuyến trong mạng cảm biến không dây để đáp ứng được yêu cầu đa dạng và đòi hỏi cao về hiệu suất của các ứng dụng WSN.

1.3 Kết luận chương

Qua chương 1, em đã giới thiệu tổng quan về mạng cảm biến không dây, cấu trúc, đặc điểm và ứng dụng của mạng. Các thách thức về định tuyến trong mạng cũng được trình bày. Để đáp ứng các yêu cầu về một thuật toán định tuyến hiệu quả cho mạng cảm biến không dây, các nhà nghiên cứu và kỹ sư đã phát triển thuật toán định tuyến RPL, sẽ được em trình bày chi tiết trong chương sau.

CHƯƠNG 2. THUẬT TOÁN ĐỊNH TUYẾN RPL

Xuất phát từ những yêu cầu của bài toán định tuyến trong chương 1, chương 2 giới thiệu đến người đọc một thuật toán định tuyến hiệu quả được sử dụng phổ biến trong các mạng cảm biến không dây hiện đại. Qua chương này, người đọc sẽ có được hiểu biết về cách thuật toán RPL hoạt động, ưu điểm cũng như nhược điểm còn tồn tại của nó. Một số nghiên cứu cải tiến thuật toán RPL cũng được trình bày.

2.1 Giới thiệu thuật toán định tuyến RPL

Để đáp ứng nhu cầu về một thuật toán định tuyến hiệu quả cho mạng cảm biến không dây tốn hao và năng lượng thấp, nhóm công tác ROLL của IETF đã phát triển thuật toán định tuyến RPL. RPL được thiết kế đặc biệt cho các mạng cảm biến không dây có năng lượng thấp và tốn hao (LLNs). Được chuẩn hóa bởi IETF trong RFC 6550[2], RPL giải quyết các thách thức của việc định tuyến trong môi trường hạn chế về năng lượng, băng thông và có nhiều sự cố mất mát dữ liệu.

RPL là một giao thức đầy triển vọng với nhiều ưu điểm cho các thiết bị nhỏ bé. Một trong những điểm mạnh chính của nó là sự linh hoạt trong việc xử lý và quản lý các thay đổi trong cấu trúc mạng và các điều chỉnh triển khai. Hơn nữa, nó được sử dụng để đáp ứng các yêu cầu của LLNs như cân bằng tải, tiêu thụ năng lượng không đồng đều và lưu lượng mạng.

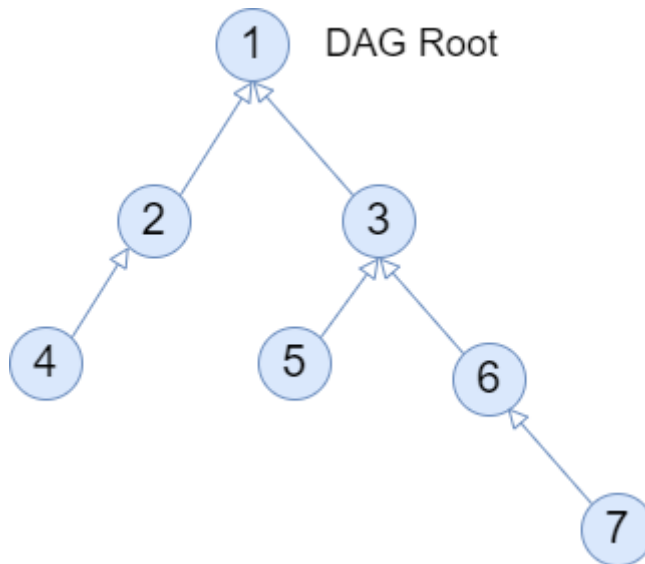
Ưu điểm:

- Tiết Kiệm Năng Lượng: RPL được thiết kế để hoạt động trong môi trường mạng cảm biến không dây, nơi mà việc tiết kiệm năng lượng là rất quan trọng. Thuật toán này cung cấp cơ chế định tuyến tối ưu để giảm thiểu việc sử dụng năng lượng của các nút mạng.
- Độ Tin Cậy Cao: RPL cung cấp khả năng chịu lỗi và tự phục hồi trong mạng. Khi có lỗi xảy ra hoặc đường dẫn gặp vấn đề, RPL có thể tìm ra đường dẫn thay thế một cách nhanh chóng và tự động.
- Hỗ Trợ Mô Hình DAG: RPL sử dụng cấu trúc Directed Acyclic Graph (DAG) để tổ chức cấu trúc mạng. Điều này giúp cho việc định tuyến trở nên linh hoạt hơn, đồng thời cũng giảm thiểu độ phức tạp trong việc xử lý thông tin định tuyến.
- Hỗ Trợ IPv6: RPL hoạt động trên nền tảng giao thức IPv6, giúp tận dụng các tính năng và lợi ích mà IPv6 mang lại cho mạng cảm biến không dây.

2.2 Các khái niệm quan trọng

Để hiểu được nguyên lý hoạt động của thuật toán RPL, chúng ta cần nắm được một số khái niệm quan trọng trong cách RPL tổ chức và định tuyến gói tin. Các khái niệm quan trọng trong RPL bao gồm:

- a. Mô hình DODAG (Destination Oriented Directed Acyclic Graph): là cấu trúc mạng cơ bản được sử dụng trong giao thức định tuyến RPL. Đây là một cấu trúc đồ thị có hướng không chu kỳ, được tổ chức để định tuyến dữ liệu từ các nút cảm biến đến một hoặc nhiều điểm đích trong mạng.



Hình 2.1: Cấu trúc DODAG

Hình 2.1 minh họa một cấu trúc DODAG đơn giản, trong đó hướng các vòng tròn nhỏ đại diện các nút cảm biến, chiều mũi tên chỉ chiều từ nút con đến nút cha ưu tiên của nó, cũng chính là hướng mà bản tin sẽ được chuyển tiếp để lên tới nút gốc.

DODAG được tổ chức sao cho tất cả các cạnh được hướng theo cách sao cho không có vòng lặp tồn tại. Tất cả các cạnh đều nằm trong các đường đi hướng về và kết thúc tại một hoặc nhiều nút gốc. Điều này đảm bảo rằng dữ liệu truyền từ nút này đến nút khác không quay trở lại điểm xuất phát, tránh các vòng lặp định tuyến.

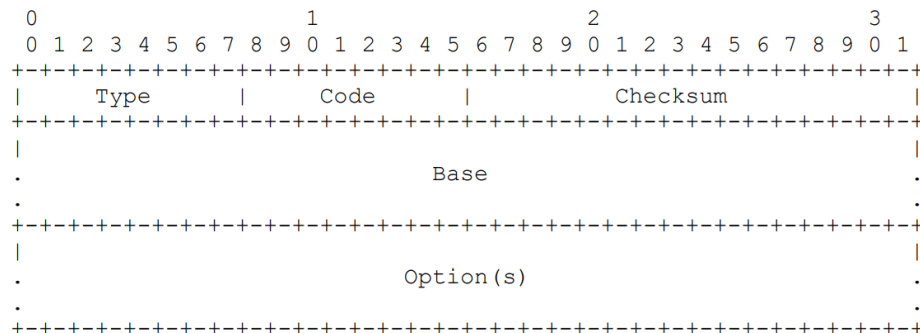
DODAG được xây dựng với một đích đến (sink node hay root node). Điểm đích này thường có năng lực xử lý và lưu trữ tốt hơn so với các nút khác.

- b. Root node hay sink node (nút gốc): Root node là nút gốc hoặc nút đích trong DODAG, đóng vai trò là điểm tập trung mà tất cả các dữ liệu trong mạng được định tuyến tới. Nó thường có các đặc điểm như năng lực xử lý mạnh, dung lượng lưu trữ lớn và kết nối mạng ổn định hơn so với các nút cảm biến thông thường. Vai trò của nút gốc bao gồm:
 - Khởi tạo DODAG: Root node khởi tạo quá trình xây dựng DODAG bằng cách phát thông điệp DIO (DODAG Information Object). Thông điệp này chứa thông tin cần thiết để các nút khác trong mạng có thể tham gia và xác định thứ hạng của mình.
 - Duy trì cấu trúc mạng: Root node thường xuyên phát thông điệp DIO để duy trì cấu trúc DODAG và cập nhật thông tin định tuyến cho các nút khác. Điều này giúp DODAG duy trì tính ổn định và thích ứng với các thay đổi trong mạng.
 - Điểm tập trung dữ liệu: Root node là điểm tập trung mà tất cả các dữ liệu từ các nút cảm biến trong DODAG được định tuyến tới. Sau khi nhận dữ liệu, root node có thể xử lý, lưu trữ hoặc chuyển tiếp dữ liệu tới các hệ thống khác bên ngoài mạng cảm biến.
- c. Up: Thuật ngữ Up đề cập đến hướng từ các nút lá về phía DODAG root, theo các cạnh DODAG. Điều này tuân theo thuật ngữ phổ biến được sử dụng

- dụng trong đồ thị và tìm kiếm theo chiều sâu, trong đó các đỉnh xa gốc hơn là "sâu hơn" hoặc "down" và các đỉnh gần gốc hơn là "nông hơn" hoặc "up".
- d. Down: Thuật ngữ Down đề cập đến hướng từ gốc DODAG tới các nút lá, theo hướng ngược lại của các cạnh DODAG.
 - e. Rank của nút: Là giá trị được gán cho mỗi nút nhằm xác định vị trí riêng của nút đó so với các nút khác đối với gốc DODAG. Thứ hạng tăng mạnh theo hướng Xuống và giảm mạnh theo hướng Lên. Cách tính chính xác Rank tùy thuộc vào Hàm mục tiêu (OF) của DAG. Rank có thể được tính toán dựa trên khoảng cách đến root node, các yếu tố về độ trễ hay chất lượng kết nối, số bước nhảy, Nhìn chung, rank trong RPL giúp xác định sự ưu tiên và chất lượng của các nút trong việc chọn lựa nút cha để xây dựng cây routing trong mạng low-power và lossy.
 - f. Parent node(Nút cha): Mỗi nút trong mạng RPL được phân bổ một hoặc nhiều nút cha (parent nodes). Nút cha là nút mà nút hiện tại chọn làm điểm đích để gửi gói tin dữ liệu đến. Khi một nút tham gia vào mạng hoặc khi đang hoạt động trong mạng, nó sẽ cần phải chọn một nút cha trong cây đường dẫn. Quá trình này thường dựa trên các tiêu chí như độ trễ, năng lượng, hoặc chất lượng kết nối. Nút cha có thể được chọn dựa trên rank của nó, một chỉ số được sử dụng để đánh giá ưu tiên của mỗi nút trong việc trở thành nút cha.
 - g. Prefer parent node (Nút cha ưu tiên): Là nút cha có chỉ số tốt nhất được chọn để nút con gửi bản tin của mình đến.
 - h. Objective Function (Hàm mục tiêu): OF xác định cách sử dụng số liệu định tuyến, mục tiêu tối ưu hóa và các chức năng liên quan để tính Rank. Hơn nữa, OF quyết định cách chọn parent node trong DODAG và do đó, hình thành DODAG.
 - i. RPL Instance: RPL Instance là một tập hợp gồm một hoặc nhiều DODAG có chung RPLInstanceID. Nhiều nhất, một nút RPL có thể thuộc về một DODAG trong RPL Instance. Mỗi instance RPL hoạt động độc lập với các instance RPL khác.
 - j. DODAG ID: DODAGID là mã định danh của DODAG root. DODAGID là duy nhất trong phạm vi của RPL Instance trong LLN. Bộ dữ liệu (RPLInstanceID, DODAGID) xác định duy nhất một DODAG.
 - k. DODAG Version: DODAG Version là một “phiên bản” tại một lần tái cấu trúc cụ thể của một DODAG gắn với một DODAGID.
 - l. Goal: Trong RPL, "goal" thường ám chỉ đến một mục tiêu cụ thể mà một hoặc nhiều nút trong mạng cố gắng đạt được. Mục tiêu này có thể bao gồm việc tối ưu hóa năng lượng, tối ưu hóa độ trễ, tối ưu hóa băng thông, hoặc mục tiêu khác liên quan đến cấu trúc và hoạt động của mạng. RPL có thể được cấu hình với mục tiêu nhất định để đáp ứng yêu cầu cụ thể của mạng.
 - m. Grounded: DODAG là grounded khi DODAG root có thể đạt được Goal.
 - n. Floating DODAG: Ngược lại với grounded

2.3 Các loại bản tin điều khiển

Cấu trúc phần thân bản tin điều khiển như sau:

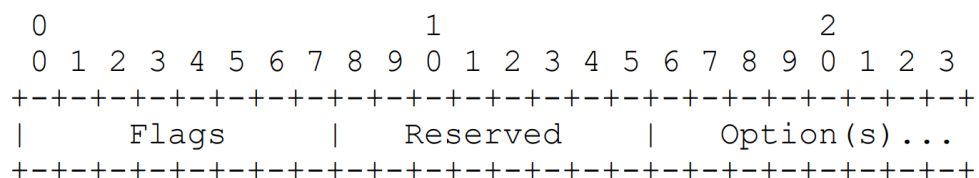


Trường code cho biết loại tin nhắn điều khiển RPL được sử dụng. Quy ước như sau:

- 0x00: DODAG Information Solicitation (DIS)
- 0x01: DODAG Information Object (DIO)
- 0x02: Destination Advertisement Object (DAO)
- 0x03: Destination Advertisement Object Acknowledgment(DAO-ACK)

Một bản tin DIS được sử dụng để yêu cầu bản tin DIO từ một node RPL.

Cấu trúc bản tin DIS như sau:



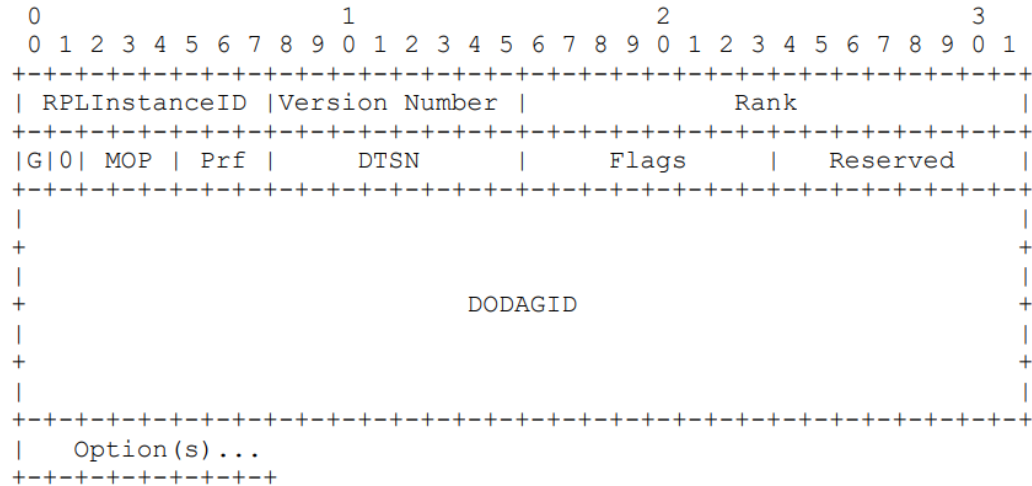
Chức năng các trường:

- 10

2.3.2 Bản tin DIO

Bản tin DIO(DODAG Information Object) được các nút cha gửi đi, mang thông tin cho phép nút khám phá RPL Instance , tìm hiểu các tham số cấu hình của nó, chọn tập DODAG parent và duy trì hoạt động của DODAG.

Cấu trúc bản tin DIO như sau:



Hình 2.4: Cấu trúc bản tin DIO [2]

- Grounded (G): Cờ 'G' cho biết liệu DODAG được quảng bá có thể đáp ứng mục tiêu do ứng dụng xác định hay không. Nếu cờ được đặt, DODAG là grounded. Nếu cờ bị xóa, DODAG là floating.
- Mode of Operation (MOP): Trường Chế độ hoạt động (MOP) xác định chế độ hoạt động của Phiên bản RPL. Quy ước như sau:

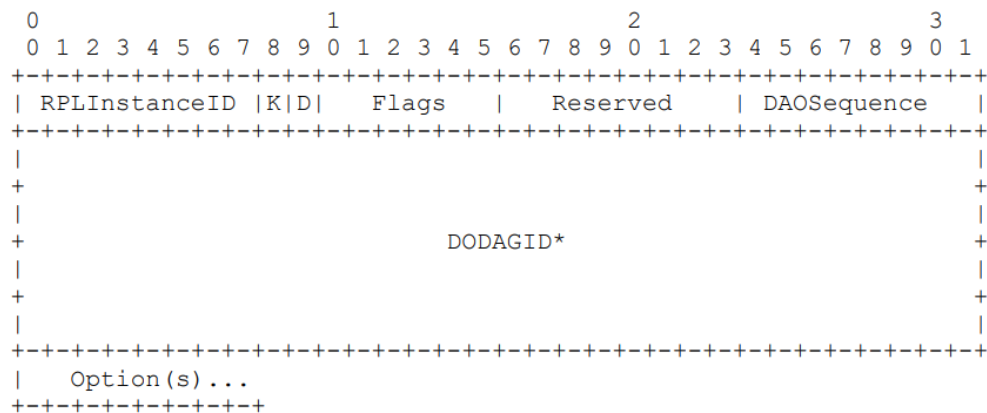
MOP	Mô tả
0	Không có tuyến đường downward nào được RPL duy trì
1	Non-Storing Mode
2	Storing Mode không hỗ trợ multicast
3	Storing Mode có hỗ trợ multicast

- DODAGPreference (Prf): Số nguyên không dấu 3 bit xác định mức độ ưu tiên của gốc của DODAG này so với các gốc DODAG khác trong Instance. DAGPreference nằm trong khoảng từ 0x00 (ít ưu tiên nhất) đến 0x07 (ưu tiên nhất). Giá trị mặc định là 0 (ít được ưu tiên nhất).
- Version number: Số nguyên không dấu 8 bit được đặt bởi gốc DODAG.
- Rank: Số nguyên không dấu 16 bit cho biết DODAG rank của nút gửi tin nhắn DIO.
- RPLInstanceID: Trường 8 bit được thiết lập bởi gốc DODAG cho biết RPL Instance nào mà DODAG này là thành viên.
- Destination Advertisement Trigger Sequence Number (DTSN): Số nguyên không dấu 8 bit được thiết lập bởi nút phát ra thông báo DIO. Cờ DTSN được sử dụng như một phần của quy trình định tuyến downward.

- Flag: Trường 8 bit không sử dụng dành riêng cho flag. Trường PHẢI được nút gửi khởi tạo về 0 và PHẢI bị nút nhận bỏ qua.
- Reserved: Trường 8 bit không sử dụng. Trường PHẢI được nút gửi khởi tạo về 0 và PHẢI bị nút nhận bỏ qua.
- DODAGID: Địa chỉ IPv6 128 bit được đặt bởi gốc DODAG để xác định duy nhất một DODAG. DODAGID PHẢI là địa chỉ IPv6 có thể định tuyến thuộc gốc DODAG.
- DIO options: Bản tin DIO có thể mang các options sau:
 - 0x00 Pad1
 - 0x01 PadN
 - 0x02 DAG Metric Container
 - 0x03 Routing Information
 - 0x04 DODAG Configuration
 - 0x08 Prefix Information

2.3.3 Bản tin DAO

Bản tin DAO được sử dụng bởi các nút con để thông báo cho các nút cha (parent nodes) về các đích mà chúng có thể đáp ứng được trong mạng. Khi các nút cha nhận được bản tin DAO từ các nút lá, chúng sẽ sử dụng thông tin trong bản tin để cập nhật bảng định tuyến của mình. Thông tin về các địa chỉ tiền tố và các thuộc tính liên quan được sử dụng để xác định các đường dẫn tới các nút lá thông qua các nút cha.



Hình 2.5: Cấu trúc bản tin DAO[2]

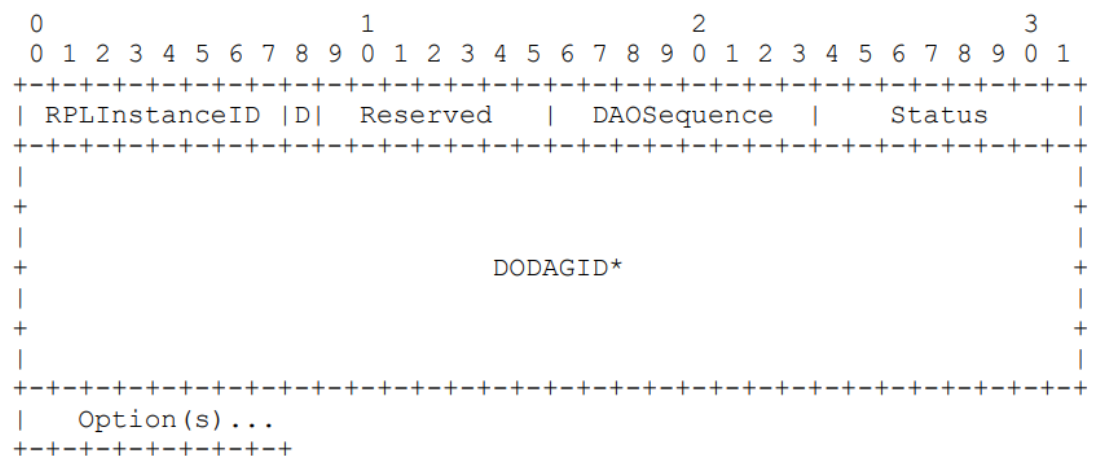
- RPLInstanceID: Trường 8 bit chỉ định phiên bản cấu trúc mạng liên quan với DODAG, giống như bản tin DIO.
- K: Cờ 'K' chỉ định rằng nút nhận được mong đợi gửi một bản tin DAO-ACK trả lại.
- D: Cờ 'D' chỉ định rằng trường DODAGID được hiện diện. Cờ này PHẢI được thiết lập khi sử dụng một RPLInstanceID cục bộ.
- Flags: 6 bit còn lại chưa sử dụng trong trường Flags được dành cho các cờ. Trường PHẢI được khởi tạo bằng không bởi bên gửi và PHẢI bị bỏ qua bởi bên nhận.

- Reserved: Trường 8 bit chưa sử dụng. Trường PHẢI được khởi tạo bằng không bởi bên gửi và PHẢI bị bỏ qua bởi bên nhận.
- DAOSequence: Được tăng lên tại mỗi bản tin DAO duy nhất từ một nút và được phản ánh trong bản tin DAO-ACK.
- DODAGID (tùy chọn): Số nguyên không dấu 128 bit được thiết lập bởi một nút gốc DODAG để định danh duy nhất một DODAG. Trường này chỉ hiện diện khi cờ 'D' được thiết lập. Trường này thường chỉ hiện diện khi sử dụng một RPLInstanceID cục bộ, để xác định DODAGID liên kết với RPLInstanceID. Khi sử dụng một RPLInstanceID toàn cục, trường này không cần phải có mặt.
- Options: Bản tin DAO có thể mang các options sau:
 - 0x00 Pad1
 - 0x01 PadN
 - 0x05 RPL Target
 - 0x06 Transit Information
 - 0x09 RPL Target Descriptor

2.3.4 Bản tin DAO-ACK

Bản tin DAO-ACK được sử dụng để xác nhận và thông báo lại cho nút gửi rằng bản tin DAO đã được nhận và xử lý.

Nút gửi bản tin DAO sẽ nhận được bản tin DAO-ACK từ các nút cha (parent nodes) để biết rằng thông tin về đường dẫn đã được nhận và xác nhận.



Hình 2.6: Cấu trúc bản tin DAO-ACK

- RPLInstanceID: Trường 8 bit chỉ định phiên bản cấu trúc mạng liên quan đến DODAG, như đã học từ bản tin DIO.
- D: Cờ 'D' chỉ định rằng trường DODAGID hiện diện. Thông thường, điều này chỉ được thiết lập khi sử dụng một RPLInstanceID cục bộ.
- Reserved: Trường 7 bit, dành cho các cờ.
- DAOSequence: Được tăng lên tại mỗi bản tin DAO từ một nút, và được phản ánh trong bản tin DAO-ACK bởi người nhận. DAOSequence được sử dụng để tương quan giữa một bản tin DAO và một bản tin DAO ACK và

không nên bị nhầm lẫn với tùy chọn Transit Information Path Sequence liên kết với một Mục Tiêu xuống DODAG cụ thể.

- Status: Chỉ định trạng thái hoàn thành. Status 0 được định nghĩa là chấp nhận không điều kiện trong tài liệu này. Các giá trị trạng thái còn lại được dành cho các mã từ chối. Không có mã trạng thái từ chối được định nghĩa trong tài liệu này, tuy nhiên mã trạng thái NÊN được cấp phát theo các hướng dẫn sau trong các tài liệu tương lai:
 - 0: Chấp nhận không điều kiện (tức là, nút nhận được bản tin DAO-ACK không bị từ chối).
 - 1-127: Không phải là từ chối trực tiếp; nút gửi bản tin DAO-ACK sẵn lòng làm cha mẹ, nhưng nút nhận được được đề xuất để tìm và sử dụng một cha mẹ thay thế thay vì vậy.
 - 127-255: Từ chối; nút gửi bản tin DAO-ACK không sẵn lòng làm cha mẹ.
- DODAGID (tùy chọn): Số nguyên không dấu 128 bit được thiết lập bởi một nút gốc DODAG để định danh duy nhất một DODAG. Trường này chỉ hiện diện khi cờ 'D' được thiết lập. Trường này thường chỉ hiện diện khi sử dụng một RPLInstanceID cục bộ, để xác định DODAGID liên kết với RPLInstanceID. Khi sử dụng một RPLInstanceID toàn cục, trường này không cần phải có mặt.

2.4 Hàm mục tiêu

Theo mô tả trong tiêu chuẩn RFC6550 [2], Hàm mục tiêu (Objective Function) xác định cách các nút RPL chọn và tối ưu hóa các tuyến đường trong phiên bản RPL. OF xác định cách các nút dịch một hoặc nhiều số liệu và ràng buộc, được xác định trong RFC6551 [3], thành một giá trị được gọi là Rank, giá trị này gần đúng với khoảng cách của nút từ gốc DODAG. OF cũng xác định cách các nút chọn nút cha. Trong thuật toán định tuyến RPL, hàm mục tiêu là một thành phần quan trọng quyết định cách nút trong mạng cảm biến không dây chọn đường đi tốt nhất đến nút gốc (root node). Các nút con sử dụng hàm mục tiêu để lựa chọn một nút cha làm nút cha ưu tiên. Hàm mục tiêu này đo lường các yếu tố khác nhau như độ trễ, độ tin cậy, và tiêu thụ năng lượng để các nút con quyết định nút cha của mình. Theo khuyến cáo trong tài liệu RFC6551 [3], các chỉ số có thể được sử dụng bao gồm:

- Chỉ số nút:
 - Năng lượng của nút
 - Hop count
- Chỉ số kết nối:
 - Thông lượng
 - Độ trễ
 - Độ tin cậy kết nối(Link Quality Level- LQL, Expected Transmission Count- ETX)

Hai hàm mục tiêu phổ biến nhất được sử dụng trong các mạng cảm biến không dây bao gồm:

a. Objective function zero(OFO):

OFO xem xét số lượng bước nhảy (hop count) như một chỉ số định tuyến để chọn ra các nút cha tốt nhất từ các nút hàng xóm ứng cử. Trong quá trình xây dựng DODAG, các nút xem xét con đường ngắn nhất dựa trên số lượng bước nhảy đến gốc grounded root. Rank của nút này nên gần với gốc. Chỉ số rank tăng dần từ nút gốc xuống đến các nút ứng cử. OFO được thiết kế để phù hợp với môi trường mạng thấp công suất và mất mát, nhưng việc sử dụng chỉ số của nút có thể cung cấp chất lượng kết nối kém. Các nút chọn con đường có số lượng bước nhảy tối thiểu ngay cả khi chúng không đáng tin cậy và dẫn đến nhiều lần gửi lại, gây ra mất gói dữ liệu nhiều hơn. Ngoài ra, việc chọn cùng một con đường cung cấp số lượng bước nhảy ngắn nhất gây ra thêm nhiều lỗi nút, ảnh hưởng đến thời gian sống của mạng.

b. Minimum rank hysteresis objective function (MRHOF):

OFO sử dụng một chỉ số tĩnh là số bước nhảy từ nút gốc để tính toán nút cha tốt nhất, khiến cho OFO chỉ xem xét nút cha cung cấp ít bước nhảy hơn, ngay cả khi có thể có kết nối kém. Vì lý do này, việc sử dụng MRHOF (hàm mục tiêu Minimum Rank with Hysteresis) dựa trên một chỉ số động của liên kết như số lượng gói truyền dự kiến (ETX) để khôi phục tính ổn định của rank. MRHOF được đề xuất bởi IETF để giải quyết các vấn đề về chỉ số tĩnh. Nó được thiết kế để chọn con đường cung cấp chi phí đường truyền thấp nhất trong khi tránh tràn dữ liệu trong mạng. MRHOF áp dụng hai cơ chế để chọn con đường tốt nhất. Thứ nhất, nó chọn con đường có rank thấp. Thứ hai, nó sử dụng cơ chế histerezis giúp giảm số lần chuyển đổi đường dẫn không cần thiết trong mạng. Cụ thể, nút sẽ giữ nguyên đường dẫn và chỉ chuyển đổi đường dẫn nếu có đường dẫn có chi phí thấp hơn sau những khoảng thời gian xác định trước.

ETX (Expected Transmission Count) là một số liệu được sử dụng trong các giao thức định tuyến mạng không dây để đánh giá chất lượng liên kết giữa các nút. Nó đo lường số lần truyền trung bình cần thiết để một gói tin được truyền thành công từ một nút nguồn đến một nút đích và ngược lại. Giá trị ETX càng thấp thì liên kết càng tốt.

Cách tính ETX: Trên lý thuyết, giá trị ETX được tính dựa trên tỉ lệ truyền thành công (delivery ratio) của các gói tin theo cả hai chiều:

- Tỷ lệ truyền thành công theo chiều đi (forward delivery ratio): d_f là tỷ lệ gói tin được truyền thành công từ nút nguồn đến nút đích.
- Tỷ lệ truyền thành công theo chiều ngược lại (reverse delivery ratio): d_r là tỷ lệ gói tin được truyền thành công từ nút đích trở lại nút nguồn.
- Công thức tính ETX là :

$$ETX = \frac{1}{d_f d_r}$$

Trong thực tế, do không đo được tỉ lệ truyền thành công theo chiều ngược, nên người ta lấy đơn giản ETX bằng số lần truyền (cả gửi lại) để truyền thành công một gói tin.

Ý nghĩa của ETX:

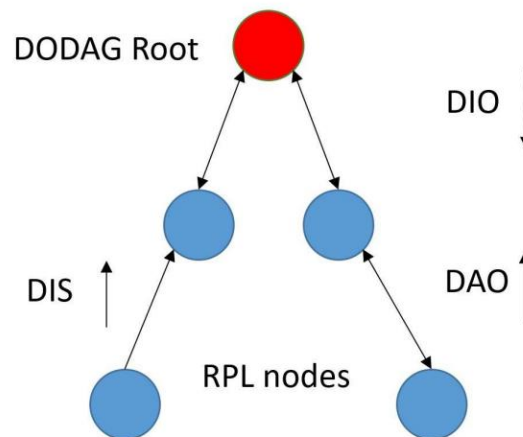
- $ETX = 1$: Liên kết hoàn hảo, chỉ cần một lần truyền để gói tin đến đích và nhận được xác nhận.
- $ETX > 1$: Liên kết kém hơn, cần nhiều lần truyền hơn để gói tin đến đích và nhận được xác nhận. Ví dụ, $ETX = 2$ có nghĩa là trung bình cần hai lần truyền để gói tin đến đích thành công.
- ETX càng cao: Chất lượng liên kết càng kém, dẫn đến tốn nhiều năng lượng và băng thông hơn.

2.5 Hoạt động định tuyến sử dụng thuật toán RPL

2.5.1 Xây dựng DODAG

Thuật toán RPL sử dụng 3 bản tin chính để xây dựng DODAG và định tuyến:

- DIS
- DIO
- DAO



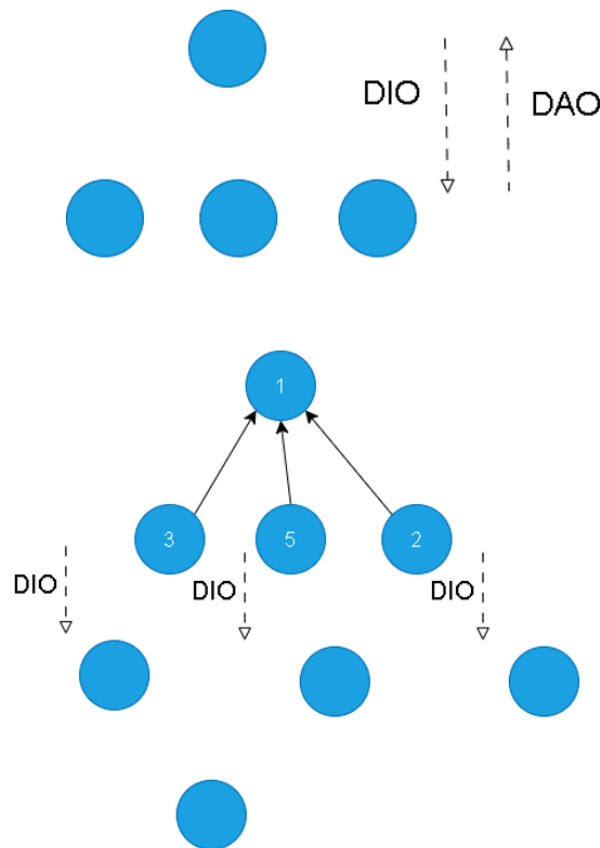
Hình 2.7: Xây dựng DODAG

Hình 2.7 minh họa hoạt động của 3 bản tin điều khiển trong thuật toán RPL. Bản tin DIO có mục tiêu hướng tới các nút ở xa nút gốc hơn (có rank cao hơn) nhằm cung cấp thông tin về rank của nút cha và các thông tin cấu hình của DODAG). Các bản tin DIS, DAO có chiều hướng lên trên thể hiện mục tiêu là để các nút con gửi đến các nút cha nhằm tìm kiếm DODAG để gia nhập hoặc gửi các thông tin về khả năng tiếp cận các nút khác của mình.

Quá trình xây dựng một DODAG được diễn ra như sau:

- Khi root node muốn xây dựng 1 DODAG, nó multicast bản tin DIO đến các node lân cận của nó.
- Bản tin DIO chứa các thông tin về tham số cấu hình của mạng, Rank của nút gửi.
- Khi các node nhận được bản tin DIO, nó sẽ tính rank của mình dựa trên hàm mục tiêu OF. Nếu nó muốn gia nhập DODAG, node sẽ gửi lại cho node đã phát bản tin DIO một bản tin DAO và được xác nhận lại bằng bản tin DAO-ACK.

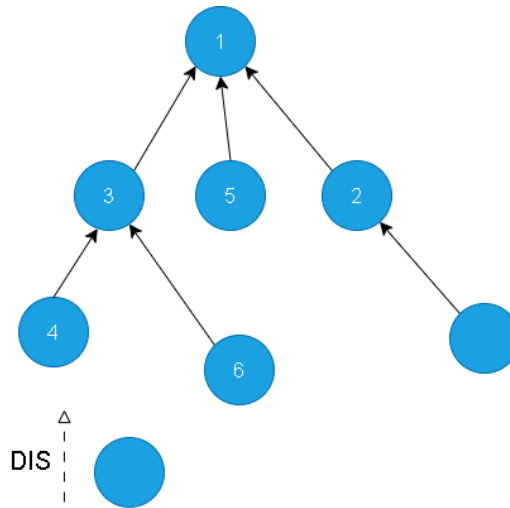
- Sau khi nhận được bản tin DIO, mỗi node có thể phát bản tin DIO của chính mình cho các node lân cận nếu nó là node router. Hoặc node có thể chỉ gửi lại bản tin DAO đến node parent mà nó chọn và im lặng nếu nó là node lá.
- Khi một node nhận được nhiều bản tin DIO, nó sẽ xem xét rank của node gửi để lựa chọn node nào sẽ trở thành parent của nó.



Hình 2.8: Hoạt động bản tin DIO

Hình 2.8 minh họa cách hoạt động của bản tin DIO. Ban đầu các nút chưa có rank xác định (minh hoạt bởi 3 nút trống ở phía trên). Sau đó nút gốc (nút trên cùng gửi bản tin DIO đến các nút xung quanh). Các nút này nhận được và dựa vào hàm OF để tính toán rank của mình, sau đó gửi bản tin DAO đến nút gốc để xin gia nhập DODAG. Sau qua trình này, các nút có rank 2, 3, 5 và chúng sẽ phát bản tin DIO của riêng mình để các xa hơn nhận được. Quá trình lặp lại cho đến khi tất cả các nút được gán một giá trị rank xác định.

Tìm kiếm DODAG: Khi một node muốn tìm kiếm 1 DODAG để gia nhập, nó có thể chủ động gửi bản tin DIS đến lân cận để yêu cầu các node khác gửi bản tin DIO.



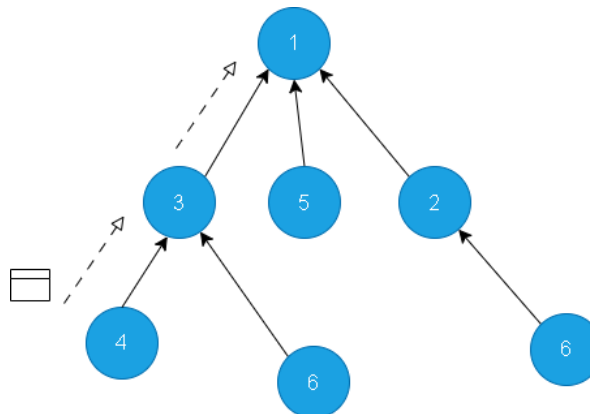
Hình 2.9: Bản tin DIS

Hình 2.9 minh họa hoạt động của bản tin DIS. Các nút có đánh số là các nút đã tham gia DODAG, chiều mũi tên chỉ chiều từ nút con đến nút cha ưu tiên của nó. Khi một nút chưa tham gia DODAG nào (nút chưa được đánh số), chúng sẽ gửi bản tin DIS tới các nút xung quanh. Khi các nút đã tham gia vào DODAG nhận được bản tin DIS, chúng sẽ phát bản tin DIO chứa thông tin về rank của mình và cấu hình DODAG đến các nút xung quanh.

2.5.2 Định tuyến hướng lên

Định tuyến hướng lên là hoạt động cơ bản của một DODAF. Chiều hướng lên là chiều từ các node lá dọc theo cạnh của DODAG để đến được Root node.

Khi một node nhận được 1 bản tin Upward mà địa chỉ đích không phải của nó, nó đơn giản sẽ chuyển tiếp bản tin lên trên parent của mình. Bằng cách này, bản tin sẽ được chuyển tiếp cho đến khi lên được nút gốc.



Hình 2.10: Định tuyến upward

Hình 2.10 minh họa đường đi của một bản tin theo chiều upward. Các nút được biểu diễn bằng hình tròn nhỏ, chiều mũi tên nét liền là chiều từ nút con đến nút cha ưu tiên của nó. Mũi tên nét đứt chỉ chiều di chuyển của bản tin. Bản tin xuất phát từ nút 4, được gửi đến nút cha ưu tiên của nó là nút 3. Tại đây, bản tin được nút 3 chuyển tiếp đến nút cha ưu tiên của nó, nút 1, cũng chính là nút gốc.

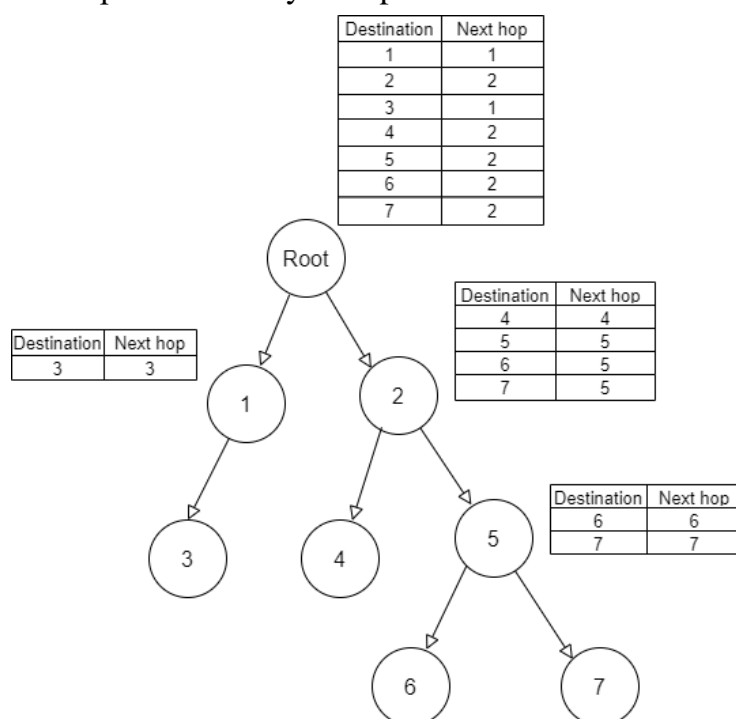
2.5.3 Định tuyến hướng xuống

Chiều định tuyến hướng xuống là chiều từ nút gốc tới các nút lá. Các nút sẽ gửi bản tin DAO chứa thông tin về các children node của mình theo hướng upward. Nút gốc và các nút đóng vai trò là router sẽ sử dụng thông tin này để xây dựng bảng định tuyến. Có hai chế độ hoạt động của định tuyến downward bao gồm chế độ lưu trữ và chế độ không lưu trữ.

a. Chế độ lưu trữ

Ở chế độ này, các nút không phải nút lá đều sẽ lưu trữ bảng định tuyến của riêng mình. Khi nút nhận được tin nhắn DAO, nó sẽ sử dụng thông tin này để cập nhập bảng định tuyến.

Khi nút nhận được gói tin theo chiều xuống, nó sẽ dựa vào bảng định tuyến của mình để lựa chọn nút tiếp theo để chuyển tiếp.



Hình 2.11: Bảng định tuyến chế độ lưu trữ

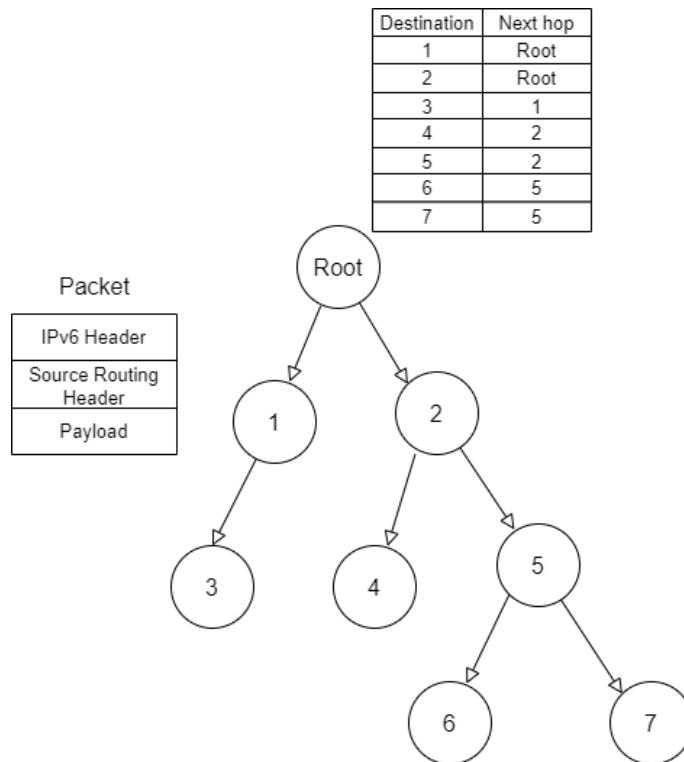
Hình 2.11 minh họa ví dụ về bảng định tuyến trong chế độ storing mode. Các nút được biểu diễn bằng các vòng tròn nhỏ, chiều mũi tên thể hiện chiều từ nút cha đến các nút con của nó. Bảng định tuyến chứa 2 giá trị, destination là địa chỉ của đích đến của bản tin, next hop là địa chỉ của nút cần chuyển tiếp bản tin để bản tin có thể đến được địa chỉ đích.

Theo Hình 2.11, next hop sẽ là nút con có thể tiếp cận đến nút đích của nút hiện tại.

b. Chế độ không lưu trữ

Ở chế độ này, chỉ có nút gốc lưu trữ bảng định tuyến của toàn bộ DODAG.

Để định tuyến gói tin, root node sử dụng bảng định tuyến, xác định đường đi và thêm thông tin đường đi (danh sách các node cần đi qua) vào trong routing header của gói tin ipv6.



Hình 2.12: Bảng định tuyến chế độ không lưu trữ

Hình 2.12 minh họa ví dụ về bảng định tuyến trong chế độ storing mode. Các nút được biểu diễn bằng các vòng tròn nhỏ, chiều mũi tên thể hiện chiều từ nút cha đến các nút con của nó. Bảng định tuyến chứa 2 giá trị, destination là địa chỉ của đích đến của bản tin, next hop là địa chỉ của nút cần chuyển tiếp bản tin để bản tin có thể đến được địa chỉ đích.

Theo Hình 2.12, next hop sẽ là nút cha ưu tiên của nút đích.

2.6 Tại sao RPL phù hợp với mạng tổn hao và năng lượng thấp

Thuật toán RPL (Routing Protocol for Low-Power and Lossy Networks) được thiết kế đặc biệt để phù hợp với các mạng tổn hao và năng lượng thấp (LLNs - Low-Power and Lossy Networks) nhờ vào những đặc điểm sau:

- Tiêu thụ năng lượng thấp: RPL sử dụng các thuật toán định tuyến và cấu trúc mạng nhằm tối ưu hóa việc sử dụng năng lượng. Các thiết bị trong LLNs thường có nguồn năng lượng hạn chế, chẳng hạn như pin, nên RPL giúp kéo dài tuổi thọ của các nút mạng bằng cách giảm thiểu số lần truyền dữ liệu không cần thiết và cho phép các nút chuyển sang chế độ ngủ khi không hoạt động.
- Khả Năng Chịu Lỗi và Linh Hoạt:
 - Chịu lỗi tốt: RPL hoạt động hiệu quả trong môi trường có nhiều nhiễu và mất mát dữ liệu. Nó xây dựng và duy trì các đồ thị có hướng không chu kỳ (DODAGs) để định tuyến, giúp nhanh chóng tìm được các tuyến đường thay thế khi xảy ra lỗi.
 - Linh hoạt: RPL có khả năng thích ứng với các thay đổi trong cấu trúc mạng và các điều kiện môi trường, đảm bảo mạng luôn hoạt động ổn định và hiệu quả.

- **Cấu Trúc DODAG:** Định tuyến động: RPL sử dụng cấu trúc DODAG, trong đó các nút cảm biến được tổ chức theo một cây đồ thị có hướng không chu kỳ. Cấu trúc này giúp tối ưu hóa đường đi từ các nút cảm biến đến các điểm đích, giảm thiểu số bước nhảy và tiêu thụ năng lượng.
- **Objective Function (OF):** Lựa chọn tuyến đường tối ưu: OF trong RPL cho phép mạng lựa chọn tuyến đường dựa trên các tiêu chí cụ thể như năng lượng, độ trễ, độ tin cậy, và số bước nhảy. Điều này giúp tối ưu hóa việc sử dụng tài nguyên và cải thiện hiệu suất tổng thể của mạng.
- **Hỗ Trợ Đa Tuyến:** Truyền thông hai chiều: RPL hỗ trợ cả truyền thông hướng lên (từ các nút cảm biến đến cổng mạng) và truyền thông hướng xuống (từ cổng mạng đến các nút cảm biến). Điều này quan trọng đối với các ứng dụng yêu cầu giám sát và điều khiển từ xa, đảm bảo tính linh hoạt và độ tin cậy của mạng.
- **Quản Lý và Bảo Trì Mạng:** Tự khắc phục và bảo trì mạng: RPL có cơ chế tự phát hiện và khắc phục lỗi định tuyến. Khi phát hiện sự cố, các nút có thể tái cấu trúc DODAG hoặc tìm các tuyến đường thay thế, đảm bảo truyền thông liên tục và đáng tin cậy.
- **Giảm gói tin điều khiển:** RPL sử dụng các gói tin điều khiển như DIO, DAO và DIS một cách hiệu quả để cập nhật và duy trì cấu trúc DODAG, giảm thiểu số lượng gói tin điều khiển cần thiết và tiết kiệm năng lượng.
- **Quản Lý Tài Nguyên Hiệu Quả:** Các thiết bị trong LLNs thường có khả năng xử lý, bộ nhớ và dung lượng lưu trữ hạn chế. RPL được thiết kế để hoạt động hiệu quả trong các điều kiện này, sử dụng ít tài nguyên nhưng vẫn đảm bảo hiệu suất cao.
- **Khả Năng Mở Rộng:** RPL có thể được cấu hình để hỗ trợ nhiều loại ứng dụng và cấu trúc mạng khác nhau, từ mạng cây, mạng lưới đến các mô hình lai. Điều này cho phép mạng cảm biến không dây LLNs dễ dàng mở rộng và tích hợp vào nhiều lĩnh vực khác nhau.

RPL phù hợp với các mạng tồn hao và năng lượng thấp nhờ vào các đặc tính như hiệu quả năng lượng, khả năng chịu lỗi, cấu trúc DODAG, Objective Function linh hoạt, hỗ trợ đa tuyến, và khả năng quản lý tài nguyên hiệu quả. Những yếu tố này giúp RPL duy trì kết nối ổn định và hiệu suất cao trong môi trường mạng có nhiều thách thức, đáp ứng được nhu cầu của nhiều ứng dụng mạng cảm biến khác nhau.

2.7 Các hướng cải tiến

Thuật toán RPL hoạt động hiệu quả với mạng cảm biến không dây tồn hao và năng lượng thấp, tuy nhiên nó vẫn còn một số hạn chế. Một số hướng cải thiện các hạn chế của thuật toán định tuyến RPL của các nghiên cứu liên quan được trình bày dưới đây.

2.7.1 Cải tiến hàm mục tiêu (OF)

Hàm mục tiêu quyết định trực tiếp đến hiệu suất của mạng cảm biến không dây. Việc lựa chọn hàm mục tiêu phù hợp sẽ giúp tiết kiệm năng lượng, tránh tắc nghẽn mạng, định tuyến gói tin hiệu quả. Vì vậy nhiều nghiên cứu tập trung vào cải tiến hàm mục tiêu cho thuật toán RPL. Có 2 cách tiếp cận chính được đề xuất:

a. Đề xuất chỉ số mới để tính OF.

Các chỉ số được sử dụng để tính toán OF dựa trên đặc tính của đường truyền (Link metrics) hoặc đặc tính của nút (Nút metrics). Các chỉ số thường được sử dụng gồm:

Link metrics:

- RSSI (Received Signal Strength Indicator) và LQI (Link Quality Indicator): Tầng vật lý cho phép xác định các đặc tính mạng như tín hiệu, tần số, điện áp, vv. Hai chỉ số ước lượng liên kết radio phổ biến nhất là Received Signal Strength Indicator (RSSI) và Link Quality Indicator (LQI). RSSI là một tầng vật lý phần cứng, hoạt động như một bộ thu phát radio được tải để kiểm tra sự có sẵn của tín hiệu tần số nhận được trước khi gửi dữ liệu. Trong khi đó, LQI đo độ tin cậy của liên kết sử dụng một phạm vi từ 0 đến 7 để chỉ ra mức độ chất lượng của liên kết.
- ETX (Expected Transmission Count): Số Lượng Gói Truyền Dự Kiến là một chỉ số đánh giá độ tin cậy của mạng. Nó thể hiện số lần truyền gói cần thiết để nhận được một phản hồi từ đích đến. Con đường cung cấp giá trị ETX thấp nhất được chọn là con đường tối ưu để đến gốc.

Nút metrics:

- Năng lượng: Năng lượng là một chỉ số của nút, thể hiện lượng năng lượng mà các nút tiêu tốn trong quá trình hoạt động mạng của chúng. Do sự phân bố của các nút và khoảng cách của chúng đến điểm thu, một số nút có thể tiêu tốn năng lượng nhanh hơn. Những nút này tạo ra một nút thắt cổ chai gây tắc nghẽn mạng. Vì lý do này, cần phải xem xét năng lượng còn lại của các nút khi thiết kế một hàm mục tiêu mới[4]. Trong nghiên cứu [5], tác giả nhận thấy các nút có rank thấp hơn (gần root nút) hơn sẽ phải vận chuyển nhiều tin nhắn do các nút con gửi hơn. Do đó tác giả đề xuất thêm hệ số điều chỉnh dựa trên rank để cân bằng năng lượng trong mạng.
- Trạng thái tải: Sự phân bố không hợp lý của cấu trúc mạng có thể khiến một số nút trung gian bị quá tải vì có quá nhiều nút con dưới nó. Nghiên cứu [6] chứng minh rằng trạng thái tắc nghẽn của mạng có thể thấy được qua dung lượng còn lại của hàng đợi tin nhắn của các nút. Vì vậy nghiên cứu đề xuất đưa hệ số sử dụng hàng đợi (queue utilization), kết hợp với giá trị ETX để làm hàm mục tiêu.

b. Kết hợp nhiều chỉ số (Metrics):

Các chỉ số định tuyến được sử dụng đơn lẻ có những ưu và nhược điểm riêng. Việc kết hợp các chỉ số lại với nhau có thể mang lại hiệu quả định tuyến cao hơn. Có 2 cách kết hợp:

- Kiểu thứ bậc: Hai chỉ số được sử dụng. Nút sẽ quyết định chọn nút cha theo từng chỉ số. Nếu giá trị của chỉ số đầu tiên bằng nhau, nút sẽ kiểm tra chỉ số thứ hai để xác định nút cha.
- Đánh trọng số: Các chỉ số được đánh trọng số để tính ra một chỉ số chung dùng để quyết định nút cha[7]. Ví dụ số bước nhảy (Hop count- HP) và Số lượng gói truyền dự kiến (Expected transmission count – ETX) được sử dụng. Chỉ số chung:

$$w = (a_1 * HP) + (a_2 * ETX)$$

ở đó a_1 và a_2 thỏa mãn điều kiện: $0 \leq a_1, a_2 \leq 1$ và $a_1 + a_2 = 1$

- Sử dụng Fuzzy logic[8]: Logic mờ đã được ứng dụng thành công trong nhiều lĩnh vực nơi mà đầu ra phụ thuộc không rõ ràng vào các đầu vào, và các đầu vào có sự tương quan lớn với nhau. Trong nghiên cứu [8], tác giả đề xuất sử dụng logic mờ để kết hợp chỉ số ETX và tiêu thụ năng lượng. Kết quả cho thấy phương pháp cân bằng phân phối tiêu thụ năng lượng tốt hơn và hiệu suất tốt hơn.

2.7.2 Cơ chế downward

Cơ chế downward của thuật toán RPL có hai chế độ: chế độ lưu trữ (storing mode) và chế độ không lưu trữ (non-storing mode). Cả hai chế độ này đều tồn tại những nhược điểm:

- Trong chế độ không lưu trữ, một định tuyến nguồn được sử dụng cho truyền gói tin, nhưng định tuyến nguồn có thể gây ra độ trễ và mất gói tin. Để thực hiện định tuyến nguồn, chế độ không lưu trữ lưu trữ tất cả thông tin đường đi định tuyến trong một DODAG root. DODAG root tạo ra các phần bổ sung của một header định tuyến bao gồm thông tin cho định tuyến nguồn. Do đó, chế độ không lưu trữ sẽ có lợi ích cho các nút bị hạn chế bộ nhớ. Tuy nhiên, chế độ không lưu trữ có thể gây ra việc chia nhỏ gói tin khi chiều dài của con đường đến một điểm đích trở nên dài hơn vì tất cả các đường đều phải được lưu trữ trong header định tuyến. Việc chia nhỏ gói tin dẫn đến việc sử dụng nhiều gói tin hơn để mang một lượng thông tin tương đương. Do đó, độ trễ và mất gói tin được gây ra bởi việc chia nhỏ gói tin.
- Chế độ lưu trữ cũng có một điểm yếu. Trong chế độ này, thông tin định tuyến hướng xuống được lưu trữ riêng biệt giữa các nút định tuyến, và việc truyền gói tin được thực hiện bằng cách sử dụng thông tin đã lưu trữ này. Việc chuyển tiếp gói tin có thể được thực hiện mà không cần bất kỳ thông tin bổ sung nào, nhưng có một hạn chế là các nút định tuyến gần với một DODAG root cần nhiều không gian lưu trữ hơn vì chúng phải đảm bảo giữ thông tin định tuyến liên quan đến toàn bộ các DODAG con. Điều này khiến cho tính mở rộng của mạng với các nút có bộ nhớ hạn chế bị hạn chế. Điểm yếu tự nhiên của chế độ lưu trữ có thể trở nên trầm trọng trong tình huống có thể xảy ra khi các nút di động xuất hiện trong mạng DODAG. Các nút di động được phép di chuyển tự do trên mạng DODAG mà không cần dọn dẹp các bảng định tuyến đã được lưu trữ trong các nút đã tham gia trước đó. Vì lý do này, trong chế độ lưu trữ, các bộ định tuyến có thể cần thêm không gian lưu trữ để lưu trữ thông tin định tuyến của các nút di động.

Nghiên cứu [9] đề xuất chế độ kết hợp giữa storing mode và non-storing mode (hybrid mode) để cải thiện nhược điểm của chế độ downward. Một phần tiêu đề mở rộng Ipv6 mới được tác giả đề xuất để mang thêm thông tin hỗ trợ định tuyến.

2.7.3 Tính bảo mật

Mặc dù RPL được thiết kế để hoạt động hiệu quả trong các mạng tổn hao và năng lượng thấp, nó vẫn dễ bị ảnh hưởng bởi nhiều loại tấn công bảo mật. Dưới đây là một số tấn công bảo mật phổ biến có thể gặp phải đối với RPL:

- Tấn công rank: Trong tấn công này, một nút độc hại báo cáo thứ hạng thấp hơn thực tế để thu hút lưu lượng từ các nút khác. Điều này có thể dẫn đến mất mát dữ liệu, tạo ra các vòng lặp định tuyến, và gây ra tình trạng tắc nghẽn ở nút độc hại. Để phòng ngừa, chúng ta có thể sử dụng các cơ chế xác thực thứ hạng, theo dõi và giám sát hành vi của nút để phát hiện và ngăn chặn các hành vi giả mạo.
- Tấn công blackhole: Một nút độc hại nhận tất cả các gói tin định tuyến và sau đó bỏ tất cả chúng. Tấn công này gây mất mát dữ liệu và làm giảm hiệu suất của mạng. Cách khắc phục: Sử dụng các cơ chế xác thực và giám sát mạng để phát hiện và loại bỏ các nút độc hại.
- Tấn công Dos: Các tấn công DoS nhằm làm quá tải mạng bằng cách gửi một lượng lớn gói tin giả mạo hoặc gây tắc nghẽn tại các nút quan trọng. Hậu quả làm gián đoạn dịch vụ, giảm hiệu suất mạng và tiêu tốn năng lượng của các nút. Cách khắc phục là sử dụng các cơ chế bảo vệ như lọc gói tin, giới hạn tốc độ và giám sát lưu lượng để phát hiện và ngăn chặn các hành vi tấn công.

Các nhà nghiên cứu đã và đang nỗ lực nhằm cải tiến thuật toán định tuyến RPL chống lại các cuộc tấn công bảo mật trên.

2.8 Kết luận chương

Mặc dù còn một số hạn chế, nhưng thuật toán RPL vẫn là một thuật toán định tuyến hiệu quả và phổ biến cho mạng cảm biến không dây. Các kỹ sư và nhà khoa học đã và đang nỗ lực nghiên cứu nhằm khắc phục các nhược điểm còn tồn tại của thuật toán RPL. Đây cũng là một chủ đề nghiên cứu đáng chú ý dành cho các nhà nghiên cứu quan tâm đến mạng cảm biến không dây và những ứng dụng của nó.

CHƯƠNG 3. MỘT SỐ THUẬT TOÁN CẢI TIẾN DỰA TRÊN RPL

Chương 3 trình bày về 3 thuật toán định tuyến cải tiến dựa trên thuật toán RPL sẽ được đánh giá trong đồ án này. Qua chương này, người đọc sẽ hiểu được cách hoạt động của thuật toán định tuyến và những mục tiêu và mỗi thuật toán nhắm đến.

Thuật toán RPL tiêu chuẩn được mô tả bởi IETF trong [10] cung cấp 2 hàm mục tiêu cơ bản là OF0 và MRHOF. Giá trị Expected Transmit Count (ETX) thường được sử dụng để như chỉ số đánh giá chất lượng đường truyền từ nút đến nút gốc. Chúng có thể hoạt động tốt trong mạng đơn giản với lưu lượng thấp. Tuy nhiên, những thuật toán này mới chỉ xem xét đến chất lượng đường truyền làm tiêu chí chọn nút cha ưu tiên, có thể dẫn đến mất cân bằng về mặt năng lượng khi một nhóm các nút cùng chọn một nút cha có chỉ số ETX nhỏ nhất để làm nút cha ưu tiên. Nút cha này sẽ nhanh chóng cạn kiệt năng lượng và ngừng hoạt động. Để khắc phục vấn đề này, một số nghiên cứu đã đề xuất các OF mới để giải quyết vấn đề cân bằng năng lượng trong mạng tổn hao và năng lượng thấp. Dưới đây em sẽ giới thiệu một số thuật toán cải tiến. Đặc điểm chung của các thuật toán này là đều sử dụng chỉ số ETX để đánh giá chất lượng đường truyền, kết hợp với một chỉ số đề xuất để cân bằng năng lượng và phát hiện tắc nghẽn trong mạng.

3.1 Thuật toán RPL cải tiến dựa trên năng lượng còn dư

Do đặc điểm của mạng cảm biến không dây năng lượng thấp là các nút thường được hoạt động trên các nguồn năng lượng hạn chế như pin, hoặc nguồn năng lượng tái tạo công suất thấp như năng lượng mặt trời. Vì vậy lượng năng lượng còn lại có thể được sử dụng của nút cần được đưa vào để xem xét như một tiêu chí để định tuyến. Nghiên cứu [11] sử dụng tỉ số năng lượng còn lại của nút trên tổng năng lượng của nút kết hợp với giá trị ETX để làm hàm mục tiêu cho thuật toán RPL.

3.1.1 Ước tính năng lượng tiêu thụ của nút

Để xác định được tỉ lệ năng lượng còn lại của nút, chúng ta cần tính được lượng năng lượng nút đã sử dụng. Việc này có thể được thực hiện bằng cách lắp đặt các mạch điện phân cứng cho phép đo công suất, dòng điện, điện áp tiêu thụ của từng nút. Tuy nhiên phần cứng triển khai thêm làm tăng chi phí, kích thước của sản phẩm, khó có thể triển khai trong thực tế. Vì vậy phương pháp ước tính năng lượng tiêu thụ bằng phần mềm được sử dụng. Phương pháp này dựa trên quan sát rằng một nút cảm biến truyền tin bằng sóng radio có 4 trạng thái hoạt động ứng với 4 mức độ tiêu thụ năng lượng khác nhau:

- CPU: Trạng thái cpu hoạt động
- LPM: Trạng thái low power mode, khi cpu vào chế độ ngủ để tiết kiệm năng lượng
- TRANSMIT: Bộ radio hoạt động để gửi dữ liệu
- LISTEN: Bộ radio lắng nghe dữ liệu gửi từ nút khác

Năng lượng tiêu thụ trong khoảng thời gian t được tính theo công thức:

$$EC = U(i_{cpu}t_{cpu} + i_{lpm}t_{lpm} + i_{tx}t_{tx} + i_{rx}t_{rx}) \quad (3.1)$$

Trong đó: EC là năng lượng tiêu thụ của nút trong thời gian t
 $i_{cpu}, i_{lpm}, i_{tx}, i_{rx}$ là dòng tiêu thụ trong từng chế độ
 $t_{cpu}, t_{lpm}, t_{tx}, t_{rx}$ là thời gian nút hoạt động trong từng chế độ
 Năng lượng còn lại của mỗi nút:

$$EE = Max\ Energy - EC \quad (3.2)$$

Trong đó: EE là năng lượng còn lại của nút
 $Max\ Energy$ là năng lượng ban đầu của nút
 EC là năng lượng đã tiêu thụ của nút tính theo (3.1)

Các giá trị $t_{cpu}, t_{lpm}, t_{tx}, t_{rx}$ là có thể đo được bằng phần mềm qua các module đo thời gian được tích hợp sẵn trong các hệ điều hành mã nguồn mở phổ biến cho mạng cảm biến không dây như Contiki OS, Tiny OS.

3.1.2 Chi phí đường đi

Theo RFC6551 [3], chi phí đường đi phải được ghi lại dọc theo đường dẫn đó. Công thức xác định chi phí đường đi (path cost) qua một nút cha nào đó như sau:

$$\begin{aligned} pathcost(N_p, S) & \quad (3.3) \\ &= ETX(N, P) + K * \left(1 - \frac{EE}{Max\ Energy}\right) \\ &+ pathcost(P, S) \end{aligned}$$

Trong đó: $pathcost(N_p, S)$ là chi phí đường đi từ nút hiện tại đến nút gốc
 $pathcost(P, S)$ là chi phí đường đi từ nút cha đến nút gốc
 $ETX(N, P)$ là chỉ số ETX từ nút con đến nút cha đang xét
 K là hệ số điều chỉnh

Rank của nút con được cập nhập theo công thức:

$$Rank(N) = MAX(Rank(P) + MinHopRankInc, pathcost(N_p, S)) \quad (3.4)$$

Bằng cách điều chỉnh hệ số K , nút con sẽ cân bằng giữa việc lựa chọn đường đi có tổng ETX nhỏ nhất, tương ứng với chất lượng tốt nhất và đường đi có tổng năng lượng còn lại lớn nhất nhất.

Phương pháp này có ưu điểm là đơn giản, có hiệu quả cao trong việc cân bằng mức độ sử dụng năng lượng của nút. Tuy nhiên các thông số $i_{cpu}, i_{lpm}, i_{tx}, i_{rx}$ là thay đổi theo phần cứng được sử dụng cho nút, vì vậy thay đổi theo từng nút, khó để tiêu chuẩn hóa.

3.2 Thuật toán RPL cải tiến dựa trên Radio Duty Cycle

3.2.1 Ước tính trạng thái tải dựa trên radio duty cycle

Đối với mạng cảm biến không dây sử dụng năng lượng từ pin, lớp MAC (Medium Access Control) thường sử dụng giao thức RDC (Radio Duty Cycle) để tiết kiệm năng lượng. RDC chịu trách nhiệm cho việc giữ bộ radio ở chế độ ngủ trong hầu hết thời gian, thường là từ 90-99% của chu kỳ hoạt động và chỉ thức dậy để nhận và gửi dữ liệu khi cần thiết. Năng lượng được tiêu thụ chủ yếu trong quá trình thức dậy để nhận và gửi dữ liệu, và tăng khi thức dậy để gửi gói tin của nó hoặc chuyển tiếp gói tin của nút khác, trong khi đó năng lượng tiêu thụ trong quá trình CPU hay LPM thường không đổi. Vì vậy, nghiên cứu [12] đề xuất sử dụng tỉ số giữa thời gian radio hoạt động và thời gian cpu hoạt động để ước tính mức độ sử dụng năng lượng của một nút. Ước tính mức độ sử dụng năng lượng của nút được tính theo công thức:

$$EE = \frac{Radio\ time}{cpu\ time} 100\% = \frac{t_{tx}+t_{rx}}{t_{cpu}} 100\% \quad (3.5)$$

Ước tính này phản ánh trạng thái tải của nút và không phụ thuộc vào phần cứng nút sử dụng.

3.2.2 Chi phí đường đi

Tương tự như phương pháp sử dụng năng lượng còn dư, công thức xác định chi phí đường đi (path cost) qua một nút cha nào đó như sau:

$$\begin{aligned} pathcost(N_p, S) \\ &= ETX(N, P) + K_r * EE \\ &+ pathcost(P, S) \end{aligned} \quad (3.6)$$

Trong đó: $pathcost(N_p, S)$ là chi phí đường đi từ nút hiện tại đến nút gốc
 $pathcost(P, S)$ là chi phí đường đi từ nút cha đến nút gốc
 $ETX(N, P)$ là chỉ số ETX từ nút con đến nút cha đang xét
 K_r là hệ số điều chỉnh

Rank của nút con được cập nhật theo công thức:

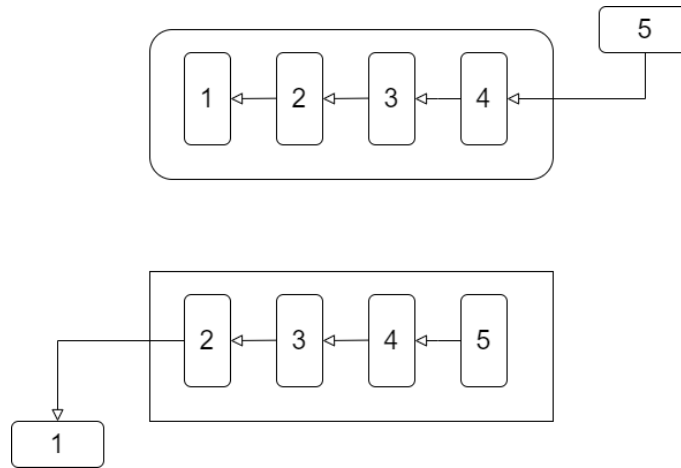
$$Rank(N) = MAX(Rank(P) + MinHopRankInc, pathcost(N_p, S)) \quad (3.7)$$

3.3 Thuật toán RPL cải tiến dựa trên tỉ lệ sử dụng hàng đợi tin nhắn

3.3.1 Hàng đợi tin nhắn

Một trong những nguyên nhân gây mất gói tin trong mạng mất mát và năng lượng thấp đó là việc gói tin bị bỏ hàng chờ tin nhắn của nút trung gian bị đầy do có quá nhiều nút chuyển tiếp đến cùng lúc.

Hàng đợi tin nhắn của nút trong mạng cảm biến hoạt động như sau:



Hình 3.1: Hoạt động của hàng đợi tin nhắn

Hình 3.1 thể hiện hoạt động của hàng đợi tin nhắn. Các tin nhắn được đánh số theo thứ tự nhận được bản tin, chiều mũi tên thể hiện chiều di chuyển của tin nhắn.

Hàng đợi tin nhắn (queue) là một cấu trúc dữ liệu hoạt động theo cơ chế Vào trước-ra trước (First in- first out hay FIFO). Khi có gói tin mới được gửi đến, gói tin được đẩy vào cuối của hàng đợi và chờ ở đó. Gói tin sẽ được lấy ra từ đầu hàng đợi để xử lý hoặc chuyển tiếp đến cho nút khác. Tuy nhiên sức chứa của hàng đợi là có giới hạn, tùy thuộc vào giới hạn phần cứng của nút. Các nút trong mạng cảm biến không dây năng lượng thấp thường được xây dựng trên phần cứng hạn chế, vì vậy kích thước hàng đợi thường là nhỏ. Trong điều kiện thông thường, lưu lượng truyền tin nhỏ, hàng đợi hiếm khi bị đầy. Tuy nhiên trong điều kiện tắc nghẽn, số lượng gói tin truyền đến nút lớn, vượt quá sức chứa của hàng đợi, buộc nút phải hủy bỏ các gói tin mới nhận được. Việc này dẫn đến mất gói tin, giảm tỉ lệ truyền thành công trong mạng.

Nghiên cứu [6] chỉ ra rằng mất gói tin do đầy hàng đợi là nguyên nhân chính gây ra mất mát gói tin trong điều kiện lưu lượng tải lớn. Vì vậy tác giả đề xuất sử dụng tỉ lệ sử dụng hàng đợi, được tính bằng tỉ số số gói tin đang trong hàng đợi chia trên kích thước của hàng đợi, như một chỉ số để tính hàm mục tiêu trong thuật toán RPL. Tác giả kết hợp giữa chỉ số ETX và tỉ lệ sử dụng hàng đợi tin nhắn- QU, trong đó ETX thể hiện chất lượng đường truyền còn QU dùng để phát hiện tình trạng tắc nghẽn tại 1 nút.

$$QU = \frac{\text{số gói tin trong hàng đợi}}{\text{kích thước hàng đợi}} 100\% \quad (3.8)$$

3.3.2 Chi phí đường đi

Công thức xác định chi phí đường đi (path cost) qua một nút cha nào đó như sau:

$$\begin{aligned} pathcost(N_p, S) &= ETX(N, P) + K_q * QU \\ &+ pathcost(P, S) \end{aligned} \quad (3.9)$$

Trong đó: $pathcost(N_p, S)$ là chi phí đường đi từ nút hiện tại đến nút gốc

$pathcost(P, S)$ là chi phí đường đi từ nút cha đến nút gốc
 $ETX(N, P)$ là chỉ số ETX từ nút con đến nút cha đang xét
 K_q là hệ số điều chỉnh

Các kết quả cho thấy thuật toán sử dụng tỉ lệ sử dụng hàng đợi hoạt động tốt trong điều kiện lưu lượng lớn, giúp cân bằng tải, tăng tỉ lệ truyền tin thành công và kéo dài thời gian sống của mạng.

3.4 Kết luận chương

Chương 3 đã trình bày về cách hoạt động của một số thuật toán định tuyến cải tiến dựa trên thuật toán RPL. Những đánh giá chi tiết về các thuật toán này được thực hiện trên thử nghiệm sẽ được trình bày ở phần sau.

CHƯƠNG 4. PHƯƠNG ÁN THỰC HIỆN ĐỀ TÀI

Chương 4 trình bày về phương pháp được sử dụng để đánh giá các thuật toán định tuyến, các tiêu chí đánh giá được đề ra để so sánh giữa các thuật toán, đồng thời giới thiệu các công cụ phần cứng và phần mềm được sử dụng để xây dựng các mô phỏng và thử nghiệm trong đề án.

4.1 Phạm vi đề tài

Đề tài “Đánh giá một số thuật toán định tuyến RPL cho mạng tổn hao và năng lượng thấp” được thực hiện nhằm thử nghiệm lại một số thuật toán định tuyến được phát triển dựa trên thuật toán RPL, từ đó đưa ra các phân tích, đánh giá điểm mạnh và điểm yếu của từng thuật toán. Kết quả đề tài có thể được sử dụng như một yếu tố tham khảo để người đọc có thể lựa chọn và xây dựng mạng cảm biến không dây của mình.

Trong đề tài này, em thực hiện đánh giá trên 4 thuật toán đã được trình bày trong chương 3, bao gồm:

- Thuật toán RPL tiêu chuẩn với hàm mục tiêu chỉ sử dụng ETX [3] (viết tắt RPL-ETX).
- Thuật toán RPL cải tiến với hàm mục tiêu sử dụng ETX kết hợp với năng lượng còn dư [11] (viết tắt ETX-EE).
- Thuật toán RPL cải tiến với hàm mục tiêu sử dụng ETX kết hợp Radio Duty Cycle [12] (viết tắt ETX-RDC).
- Thuật toán RPL cải tiến với hàm mục tiêu sử dụng ETX kết hợp tỉ lệ sử dụng hàng đợi [6] (viết tắt ETX-QU).

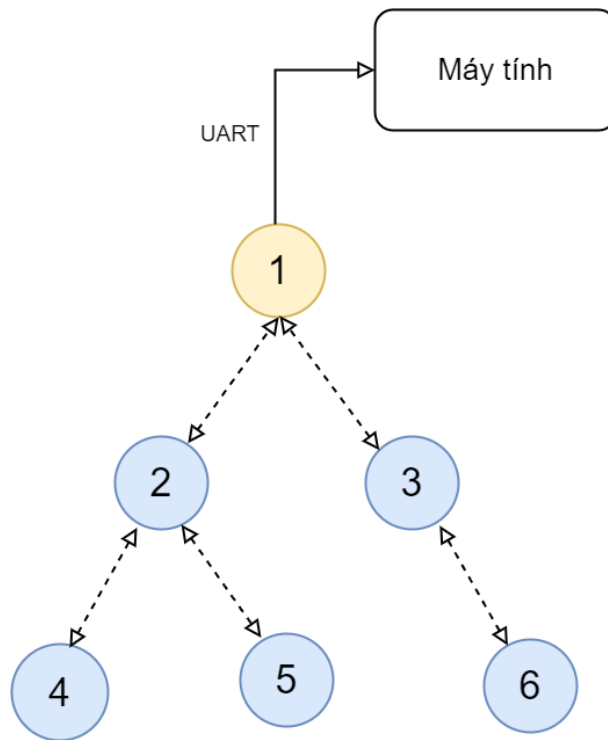
Các thuật toán trên có mục tiêu chính là cân bằng phân bố mức độ sử dụng năng lượng trong mạng, giảm tắc nghẽn, tăng tỉ lệ truyền tin thành công. Các thuật toán được đánh giá bằng mô phỏng và thử nghiệm trên thiết bị thật.

Do trong mạng cảm biến không dây thực tế, các nút thường chỉ hoạt động ở chế độ upward - tức là các nút cảm biến thu thập dữ liệu từ môi trường, sau đó gửi lên nút gốc. Nút gốc ít khi phải gửi dữ liệu đến các nút phía dưới. Vì vậy các thử nghiệm trong đề án chỉ đánh giá hoạt động của chế độ upward.

4.2 Phương án thực hiện

4.2.1 Sơ đồ tổng quan hệ thống

Để đánh giá các thuật toán, em xây dựng một mạng cảm biến không dây, với một nút vừa đóng vai trò là nút gốc trong mô hình DODAG, vừa tiếp nhận dữ liệu từ các nút khác. Các nút khác sẽ gửi thông tin về trạng thái của mình như năng lượng còn lại, số tin nhắn đã gửi, ... đến cho nút gốc. Dữ liệu được thu thập sẽ được nút gốc gửi lên máy tính qua cổng serial. Dữ liệu sẽ được lưu trữ vào một file văn bản và file sẽ được xử lý để đưa ra các thông số về năng lượng còn lại, tỉ lệ truyền thành công, độ trễ trung bình, ...



Hình 4.1: Sơ đồ hệ thống

4.2.2 Ngăn xếp mạng (network stack)

Ngăn xếp mạng (network stack hay protocol stack), là tập hợp các giao thức mạng được sử dụng, hoạt động cùng nhau để đảm bảo truyền gói tin từ nguồn đến đích. Trong mô phỏng và thực nghiệm, em sử dụng một ngăn xếp mạng phổ biến thường dùng triển khai cho các mạng cảm biến IoT hiện đại.

Bảng 4.1: Ngăn xếp mạng được sử dụng

Tầng giao vận (transport layer)	UDP
Tầng mạng	Ipv6 + RPL
Tầng thích ứng	6LowPan
Tầng liên kết	CSMA, ContikiMAC, RDC
Tầng vật lý	IEEE 802.15.4 Channel 25

Đây là stack mạng điển hình sử dụng mạng 6LowPan trên nền IEEE 802.15.4. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) là một tiêu chuẩn mạng được thiết kế để cho phép các thiết bị có khả năng truyền thông tin qua các mạng không dây cá nhân với mức tiêu thụ năng lượng thấp, đặc biệt là trong các ứng dụng Internet of Things (IoT). Các đặc điểm chính của mạng 6LowPan bao gồm:

- *Sử dụng IPv6*: 6LoWPAN cho phép các thiết bị sử dụng giao thức IPv6, cung cấp một không gian địa chỉ rộng lớn và các lợi ích khác của IPv6.
- *Năng lượng thấp*: Được thiết kế cho các thiết bị có giới hạn về năng lượng, như cảm biến và thiết bị điều khiển trong mạng IoT.

- *Nén Header*: 6LoWPAN sử dụng kỹ thuật nén header để giảm kích thước gói tin, giúp tiết kiệm băng thông và năng lượng.
- *Mạng không dây cá nhân*: Được sử dụng trong các mạng không dây với phạm vi ngắn và tốc độ truyền dữ liệu thấp, như IEEE 802.15.4.
- *Mạng lưới*: Hỗ trợ cấu trúc mạng lưới (mesh network), cho phép các thiết bị giao tiếp với nhau qua nhiều bước nhảy (hops), tăng khả năng mở rộng và độ tin cậy của mạng.

Theo Bảng 4.1, các tầng của mạng có chức năng cụ thể như sau:

- Tầng giao vận**: Cho phép nhiều ứng dụng trên mỗi thiết bị có kênh liên lạc riêng. Ở đây tầng giao vận sử dụng giao thức UDP- User Datagram Protocol. Đây là giao thức được thiết kế để cung cấp một phương tiện truyền tải dữ liệu đơn giản và hiệu quả, không đòi hỏi các dịch vụ bổ sung như kiểm soát lỗi, kiểm soát luồng, không phân đoạn và tái lắp ghép. UDP phù hợp với các ứng dụng cần tốc độ cao và độ trễ thấp. Đặc biệt UDP được sử dụng nhiều trong mạng cảm biến trong các ứng dụng IoT bởi vì tính chất đơn giản, phù hợp với các thiết bị có tài nguyên hạn chế, các gói tin thường nhỏ và yêu cầu độ tin cậy thấp.

Trong mô phỏng và thử nghiệm, em sử dụng gói tin UDP có kích thước và các trường như sau:

- Tầng mạng**: Lớp mạng thực hiện đánh địa chỉ và định tuyến gói tin trong mạng. Ipv6 cung cấp không gian địa chỉ 128 bit lớn, giải quyết vấn đề cạn kiệt địa chỉ mà ipv4 gặp phải. Thuật toán định tuyến RPL tối ưu cho mạng cảm biến không dây năng lượng thấp và mất mát.
- Tầng thích ứng**: Tầng thích ứng (Adaptation Layer) trong 6LoWPAN là một thành phần quan trọng, đóng vai trò trung gian giữa giao thức IPv6 và giao thức mạng không dây IEEE 802.15.4. Tầng này chịu trách nhiệm đảm bảo rằng các gói IPv6 có thể được truyền qua các mạng không dây năng lượng thấp một cách hiệu quả và đáng tin cậy. Tầng thích ứng có chức năng nén header ipv6 nhằm tối ưu băng thông và tiết kiệm năng lượng, phân đoạn và tái lắp ghép gói tin ipv6 để phù hợp với kích thước gói tin tối đa là 127 octets của chuẩn IEEE 192.15.4, và một số chức năng khác.
- Tầng liên kết**: Có vai trò cung cấp một đường truyền dữ liệu đáng tin cậy và hiệu quả giữa 2 nút được kết nối trực tiếp với nhau. Tầng liên kết dữ liệu điều khiển cách các thiết bị truy cập và sử dụng phương tiện truyền thông (media). Điều này bao gồm các cơ chế để tránh hoặc xử lý xung đột dữ liệu khi nhiều thiết bị cố gắng truyền dữ liệu đồng thời.

CSMA là phương pháp điều khiển truy cập đa truy cập được sử dụng trong tầng liên kết. CSMA yêu cầu các thiết bị trong mạng đầu tiên phải kiểm tra đường truyền có đang được sử dụng hay không trước khi bắt đầu truyền dữ liệu. Nếu đường truyền đang bận, thiết bị sẽ phải chờ cho đến khi đường truyền rảnh.

Giao thức MAC được sử dụng là ContikiMAC, được cung cấp mặc định bởi hệ điều hành Contiki OS, hỗ trợ Radio Duty Cycling. Đây là phương thức cho phép các nút ở trong chế độ ngủ trong hầu hết thời gian hoạt

động(khoảng 90-99%), chỉ thức dậy để truyền nhận dữ liệu khi cần thiết. Mục đích nhằm tiết kiệm năng lượng tiêu thụ, đồng thời đảm bảo mạng không dây hoạt động hiệu quả nhờ giảm thiểu xung đột tín hiệu.

- e. Tầng vật lý: Chuẩn IEEE 802.15.4 là một chuẩn kỹ thuật cho mạng không dây cá nhân (WPAN - Wireless Personal Area Network), được phát triển để hỗ trợ các ứng dụng có yêu cầu về mức độ tiêu thụ năng lượng thấp, chi phí thấp và khoảng cách truyền tải ngắn. Đây là một trong những chuẩn quan trọng trong các mạng cảm biến không dây và IoT (Internet of Things). Chuẩn này hoạt động ở các tần số vô tuyến thấp, bao gồm 868 MHz, 915 MHz và 2.4 GHz, tùy thuộc vào khu vực địa lý và yêu cầu cụ thể của ứng dụng. Ở đây em sử dụng tần số 24 GHz, channel 25. IEEE 802.15.4 còn cung cấp các cơ chế để tiết kiệm năng lượng cho các thiết bị, bao gồm các chế độ ngủ (sleep modes) và cơ chế tự động đánh thức (wake-up mechanisms). Mạng 6LoWPAN thường được triển khai chủ yếu trên giao thức IEEE 802.15.4.

4.2.3 Các thông số đánh giá

Các thông số được đánh giá bao gồm:

- a. Lượng năng lượng còn lại của các nút: Năng lượng là yếu tố hàng đầu cần cân nhắc trong mạng cảm biến không dây. Với cùng một lượng năng lượng ban đầu, lượng năng lượng còn lại của từng nút quyết định thời gian sống của mạng. Khi xảy ra tắc nghẽn trong mạng, một số nút phải trở thành nút chuyển tiếp cho rất nhiều nút phía dưới. Các nút này sẽ hết năng lượng rất nhanh, trong khi các nút khác vẫn còn nhiều năng lượng. Thuật toán định tuyến tốt phải có khả năng phân bổ lưu lượng gói tin qua các nút để cân bằng lượng năng lượng còn lại giữa các nút, đảm bảo sử dụng năng lượng một cách hiệu quả.
- b. Tổng lượng năng lượng còn lại: Bảng tổng năng lượng còn lại của tất cả các nút, thể hiện tổng chi phí năng lượng cần để truyền tin.
- c. Tỷ lệ truyền nhận gói tin thành công: Được tính bằng số gói tin truyền thành công trên số gói tin đã gửi.
- d. Tổng số gói tin mất do hàng đợi đầy: Tiêu chí này được đánh giá trong thử nghiệm với mạng có lưu lượng lớn. Số gói tin mất càng nhỏ thể hiện phân bổ lưu lượng tốt hơn, sử dụng hàng đợi hiệu quả hơn.
- e. Độ trễ trung bình của các nút xa nhất: Là thời gian trung bình để gói tin đi từ các nút xa nhất đến nút gốc. Độ trễ có thể được đo dễ dàng trong mô phỏng, nhưng trong thực nghiệm rất khó do yêu cầu tính đồng bộ cao về thời gian giữa các nút. Vì vậy độ trễ chỉ được tính trong thử nghiệm bằng mô phỏng.
- f. Số bước nhảy trung bình: Trung bình số nút trong gian mà gói tin cần đi qua để đi từ nút nguồn đến nút đích.

4.3 Công cụ sử dụng

4.3.1 Phần mềm

a. Contiki OS

Contiki là một hệ điều hành nhúng mã nguồn mở nhẹ, linh hoạt và hiệu quả được thiết kế đặc biệt cho mạng cảm biến không dây (Wireless Sensor Networks - WSNs) và thiết bị Internet of Things (IoT). Được phát triển bởi Adam Dunkels từ năm 2002 và hiện được duy trì bởi một nhóm lập trình viên và cộng đồng sáng tạo, Contiki đã trở thành một trong những hệ điều hành nhúng phổ biến và được sử dụng rộng rãi trong các ứng dụng như mạng cảm biến, IoT, và các thiết bị thông minh.



Hình 4.2: Logo contiki-ng

Contiki OS được sử dụng rộng rãi nhờ những ưu điểm sau:

- *Kích thước nhỏ và tiết kiệm năng lượng:* Contiki được thiết kế để có kích thước nhỏ và tối ưu với yêu cầu về tài nguyên hạn chế của các thiết bị nhúng và mạng cảm biến. Kích thước mã nguồn cơ bản của Contiki chỉ khoảng vài KB đến vài chục KB, tùy thuộc vào cấu hình và chức năng được kích hoạt. Contiki có thể chạy trên các vi xử lý có cấu hình thấp như 8-bit AVR với RAM chỉ từ vài KB đến vài chục KB. Trong môi trường mạng cảm biến không dây, Contiki đã được kiểm tra và chứng minh có thể hoạt động trên các nút cảm biến với tiêu thụ năng lượng chỉ khoảng vài miliwatt trong trạng thái hoạt động và vài micro-watt trong trạng thái sleep.
- *Hỗ trợ đa dạng các giao thức mạng:* Contiki hỗ trợ giao thức IPv6 và 6LoWPAN, giúp các thiết bị IoT kết nối trực tiếp với Internet thông qua IPv6. Contiki cũng hỗ trợ các giao thức mạng cảm biến như RPL (Routing Protocol for Low-Power and Lossy Networks), CoAP (Constrained Application Protocol), và MQTT (Message Queuing Telemetry Transport). Contiki cung cấp thư viện và hỗ trợ cho các giao thức này, giúp ứng dụng IoT truyền thông và tương tác một cách hiệu quả.
- *Đa nền tảng:* Contiki có thể chạy trên nhiều loại vi xử lý và nền tảng phần cứng khác nhau như: vi xử lý ARM Cortex-M (Phổ biến trong các ứng dụng IoT có hiệu suất cao), vi xử lý AVR (Thích hợp cho các ứng dụng đòi hỏi tiết kiệm năng lượng và kích thước nhỏ), MSP430 (Dòng vi xử lý tiết kiệm năng lượng, thích hợp cho các thiết bị cảm biến không dây). Điều này tạo ra sự linh hoạt trong việc triển khai và phát triển các ứng dụng trên nhiều nền tảng khác nhau.

- *Khả năng đa nhiệm:* Contiki hỗ trợ việc chạy nhiều tiến trình đồng thời thông qua cơ chế ngắt (interrupt-driven). Mỗi tiến trình có thể chạy trong một ngữ cảnh bảo mật riêng, giúp người phát triển quản lý tác vụ và tài nguyên một cách linh hoạt. Tiến trình trong Contiki thường chỉ chiếm ít bộ nhớ (ví dụ: chỉ vài KB cho mỗi tiến trình), giúp giảm tải cho hệ thống.
- *Cộng đồng phát triển lớn:* Contiki có hơn 450 người dùng đăng ký trong nhóm người dùng chính thức trên GitHub và hàng trăm dự án, phát triển và ứng dụng được xây dựng trên nền tảng này. Cộng đồng Contiki liên tục cập nhật, sửa lỗi và cải tiến mã nguồn, đồng thời cung cấp hỗ trợ cho người dùng thông qua diễn đàn trực tuyến và tài liệu hướng dẫn.
- *Tiện ích phát triển mạnh mẽ:* Contiki cung cấp các công cụ mô phỏng như Cooja, một môi trường mô phỏng mạng cảm biến không dây. Trình biên dịch GCC, makefile và các công cụ phát triển phổ biến khác được tích hợp sẵn trong Contiki. Các thư viện chuẩn hóa như ContikiMAC, ContikiRPL, và ContikiNet giúp đơn giản hóa quá trình phát triển.

Contiki được sử dụng trong nhiều ứng dụng khác nhau, bao gồm hệ thống giám sát môi trường, các ứng dụng y tế, quản lý năng lượng, quản lý đèn đường thông minh, và nhiều ứng dụng IoT khác. Với tính linh hoạt, hiệu quả và khả năng mở rộng, Contiki là một lựa chọn phổ biến cho việc phát triển các ứng dụng nhúng và IoT đa dạng và tiện lợi.

Contiki OS cung cấp các thư viện để triển khai các giao thức truyền thông không dây phổ biến trong mạng IoT hiện đại. Các giao thức được hỗ trợ như sau:

Bảng 4.2: Các giao thức hỗ trợ bởi Contiki OS

Tầng	Giao thức
Application	MQTT, COAP, HTTP, SNMP, DTLS
Transport	UDP, TCP
Network	IP, IPv6, RPL
Adaptation	6LowPan
MAC	TSCH, CSMA, NullMAC
Duty Cycling	ContikiMAC, NullRDC
Physic	CC2650, CC2652, CC1310, CC1312, Cooja, CC2538, JN516x, NullRadio

Một số phần cứng được hỗ trợ bởi Contiki OS:

- AVR: kit Micaz- sản xuất bởi Crossbow technology, atmega128 + CC2420
- RL78 + ADF7023 sub-GHz platform - eval-adf7023- board phát triển từ Analog Devices.
- 8051-Based-Platforms (TI cc2530 DK and Sensinode/cc2430)
- Zolertia-z1-motes (MSP430x2 + CC2420 based)
- Zolertia Zoul platforms (RE-Mote and Firefly) (CC2538 + CC1200 based, dual radio)

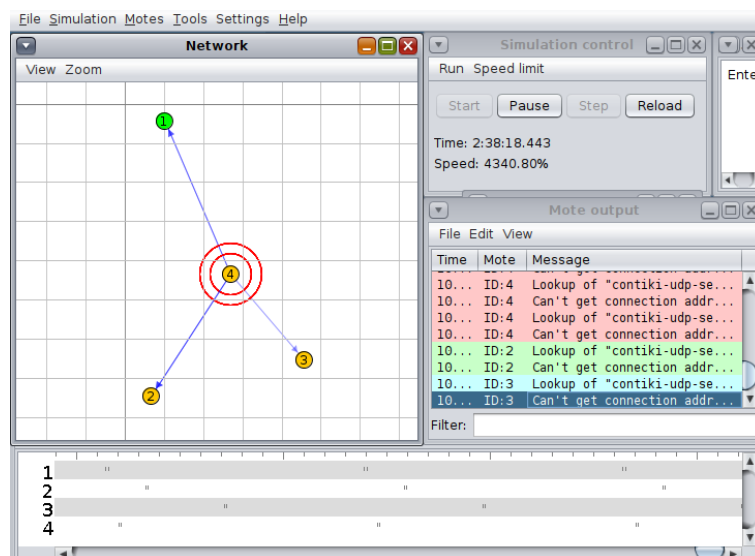
- Kit phát triển TI CC2538
- TI srf06-cc26xx: Hỗ trợ cho TI's CC2630 và CC2650 CPUs. Hai board phát triển được hỗ trợ: The CC2650 SensorTag và CC2630/CC2650 EM gắn trên SmartRF 06 EB.

Người dùng tự có thể triển khai Contiki OS trên nền tảng của riêng mình bằng cách thêm các driver cần thiết cho phần cứng, chỉnh sửa makefile. Contiki OS hỗ trợ việc này và có hướng dẫn tương đối chi tiết [13].

b. Phần mềm mô phỏng Cooja

Cooja (Co-simulation of Objects in Java) là phần mềm mô phỏng mạng cảm biến không dây. Nó được phát triển như một phần của dự án Contiki OS. Cooja cho phép các nhà nghiên cứu và nhà phát triển kiểm tra và mô phỏng các mạng cảm biến không dây trong môi trường ảo trước khi triển khai thực tế. Điều này giúp tiết kiệm thời gian và chi phí cũng như giảm thiểu rủi ro.

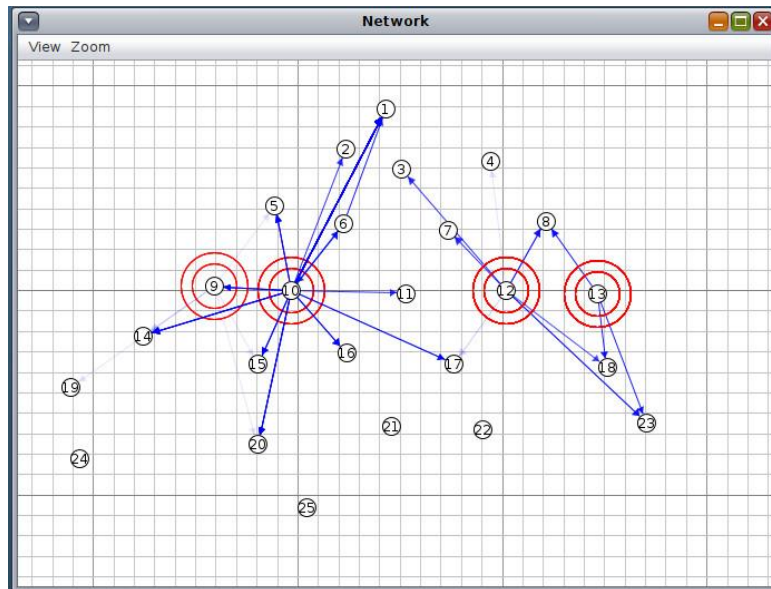
Cooja cung cấp giao diện người dùng trực quan, dễ dàng theo dõi, cùng nhiều tiện ích hữu dụng.



Hình 4.3: Giao diện Cooja Simulator

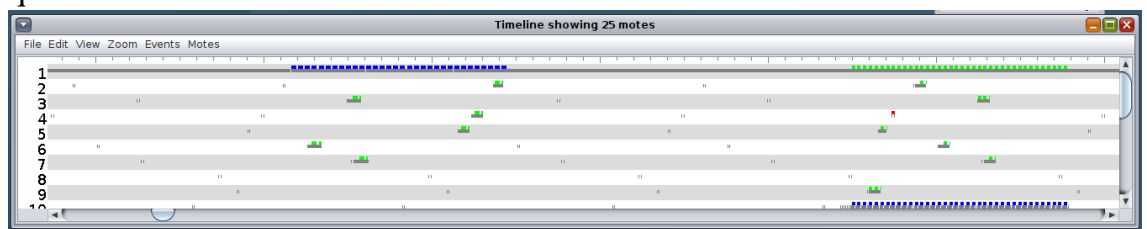
Một số tính năng của Cooja:

- Cửa sổ Network: Quan sát vị trí tương đối của các nút trong mạng, cung cấp cái nhìn tổng quan về cấu trúc mạng, quản lý các thành phần trong mạng.



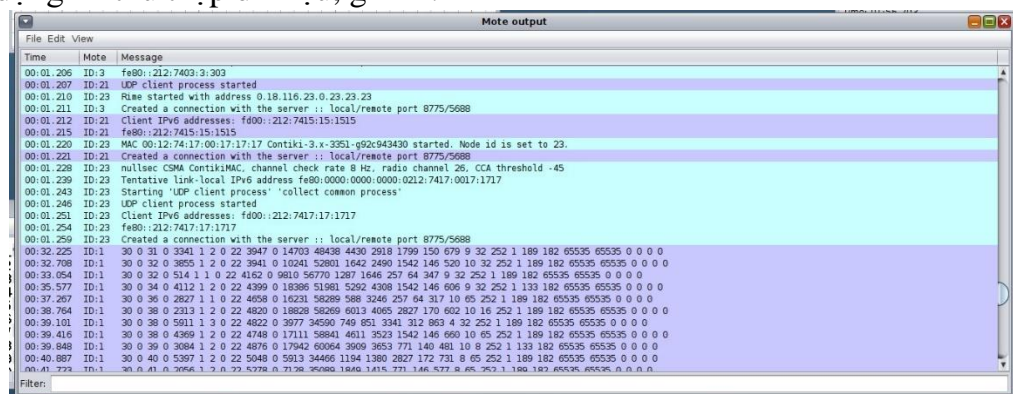
Hình 4.4: Cửa sổ network Cooja

- Cửa sổ Timeline: Hiển thị các sự kiện và hoạt động của các node trong mô phỏng. Cửa sổ Timeline trong Cooja cho phép phân tích các sự kiện và hoạt động diễn ra trong mạng mô phỏng theo thời gian một cách thuận tiện và trực quan.



Hình 4.5: Cửa sổ Timeline

- Mote output: Hiển thị dữ liệu được các nút gửi ra qua giao thức UART. Sử dụng để thu thập dữ liệu, gỡ lỗi.



Hình 4.6: Cửa sổ Mote output của Cooja

Cooja là một công cụ mạnh mẽ và linh hoạt cho việc mô phỏng mạng cảm biến không dây, hỗ trợ hiệu quả cho nghiên cứu, phát triển và giáo dục trong lĩnh vực IoT và mạng nhúng. Được phát triển cùng với Contiki OS, vì vậy Cooja được tối ưu cho mô phỏng nút cảm biến xây dựng bằng Contiki OS.

c. Ngôn ngữ Python

Sau khi mô phỏng hoặc thực nghiệm, dữ liệu thu được dưới dạng file văn bản. Để thu được các kết quả mong muốn, file dữ liệu cần được đọc, phân tích và xử lý. Em sử dụng ngôn ngữ lập trình Python để viết code đọc và xử lý file kết quả, tính toán và vẽ đồ thị minh họa kết quả thử nghiệm.

- Dễ đọc và dễ học, cú pháp đơn giản
- Mạnh mẽ: Cung cấp nhiều thư viện như Pandas, NumPy, SciPy, Matplotlib, và Scikit-Learn hỗ trợ mạnh mẽ cho việc phân tích và trực quan hóa dữ liệu.
- Đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux, và nhiều hệ điều hành khác, giúp ứng dụng dễ dàng di chuyển và triển khai.
- Cộng đồng mạnh mẽ: Python có một cộng đồng người dùng và nhà phát triển rộng lớn và sôi động, cung cấp tài liệu phong phú, diễn đàn thảo luận, và hỗ trợ kỹ thuật.

Log from node

Serial Data Bar Chart

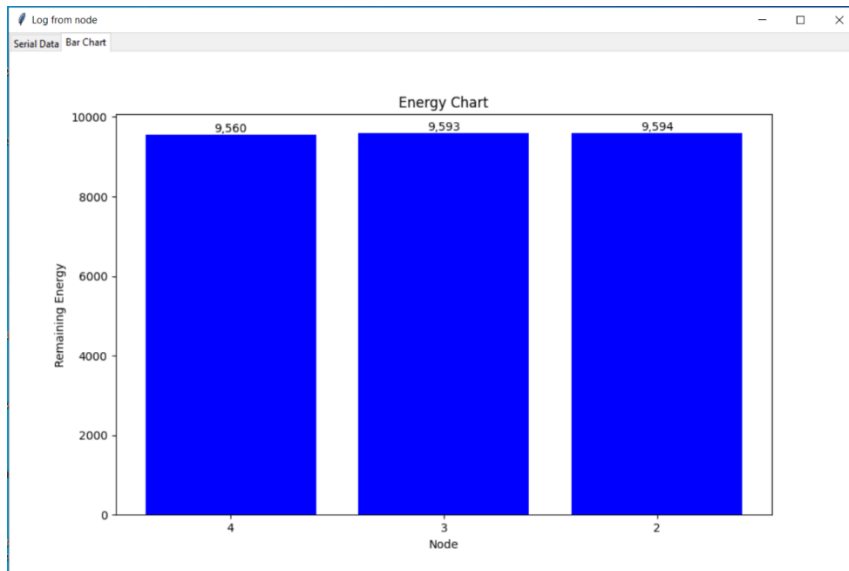
```

[00:13:42.238] TCP, 1, 39158, 3, 87, 1, 0, 33490, 9633, 0, ac03
[00:13:42.611] TCP, 1, 39206, 4, 90, 1, 0, 34691, 9603, 0, ac03
[00:13:42.907] TCP, 1, 39254, 2, 84, 1, 0, 32528, 9637, 0, ac03
[00:13:45.361] TCP, 1, 39558, 3, 88, 1, 0, 33891, 9629, 0, ac03
[00:13:46.486] TCP, 1, 39702, 2, 85, 1, 0, 32978, 9630, 0, ac03
[00:13:47.422] TCP, 1, 39822, 4, 91, 1, 0, 35311, 9596, 0, ac03
[00:13:47.505] TCP, 1, 39830, 2, 86, 1, 0, 33105, 9628, 0, ac03
[00:13:47.994] TCP, 1, 39894, 3, 89, 1, 0, 34228, 9626, 0, ac03
[00:13:50.612] TCP, 1, 40230, 4, 92, 1, 0, 38719, 9593, 0, ac03
[00:13:51.049] TCP, 1, 40286, 4, 93, 1, 0, 35722, 9592, 0, ac03
[00:13:51.790] TCP, 1, 40382, 2, 87, 1, 0, 33658, 9623, 0, ac03
[00:13:52.290] TCP, 1, 40446, 3, 90, 1, 0, 34777, 9622, 0, ac03
[00:13:53.924] TCP, 1, 40654, 2, 88, 1, 0, 33928, 9620, 0, ac03
[00:13:54.862] TCP, 1, 40774, 4, 94, 1, 0, 36258, 9598, 0, ac03
[00:13:55.522] TCP, 1, 40910, 3, 91, 1, 0, 35433, 9617, 0, ac03
[00:13:56.234] TCP, 1, 40950, 2, 89, 1, 0, 34225, 9616, 0, ac03
[00:13:58.547] TCP, 1, 41246, 4, 95, 1, 0, 36734, 9583, 0, ac03
[00:13:59.612] TCP, 1, 41382, 3, 92, 1, 0, 35710, 9614, 0, ac03
[00:14:00.172] TCP, 1, 41454, 2, 90, 1, 0, 34730, 9611, 0, ac03
[00:14:00.421] TCP, 1, 41486, 3, 93, 1, 0, 35814, 9612, 0, ac03
[00:14:01.799] TCP, 1, 41662, 4, 96, 1, 0, 37146, 9577, 0, ac03
[00:14:02.861] TCP, 1, 41788, 2, 91, 1, 0, 35075, 9608, 0, ac03
[00:14:05.100] TCP, 1, 42086, 3, 94, 1, 0, 36416, 9606, 0, ac03
[00:14:05.423] TCP, 1, 42126, 4, 97, 1, 0, 37617, 9572, 0, ac03
[00:14:06.361] TCP, 1, 42246, 2, 92, 1, 0, 35523, 9604, 0, ac03
[00:14:07.252] TCP, 1, 42310, 3, 95, 1, 0, 36645, 9604, 0, ac03
[00:14:07.379] TCP, 1, 42342, 4, 98, 1, 0, 37826, 9570, 0, ac03
[00:14:09.796] TCP, 1, 42696, 4, 99, 1, 0, 38174, 9567, 0, ac03
[00:14:10.173] TCP, 1, 42734, 2, 93, 1, 0, 36013, 9600, 0, ac03
[00:14:10.736] TCP, 1, 42806, 3, 96, 1, 0, 37141, 9600, 0, ac03
[00:14:12.732] TCP, 1, 43062, 3, 97, 1, 0, 37392, 9598, 0, ac03
[00:14:13.235] TCP, 1, 43126, 2, 94, 1, 0, 36406, 9597, 0, ac03
[00:14:13.485] TCP, 1, 43158, 4, 100, 1, 0, 39647, 9562, 0, ac03

```

Command

Chương trình có 2 tab: Tab thứ nhất hiển thị dữ liệu được gửi lên cổng Serial của máy tính. Tab thứ hai hiển thị mức năng lượng còn lại của mỗi nút

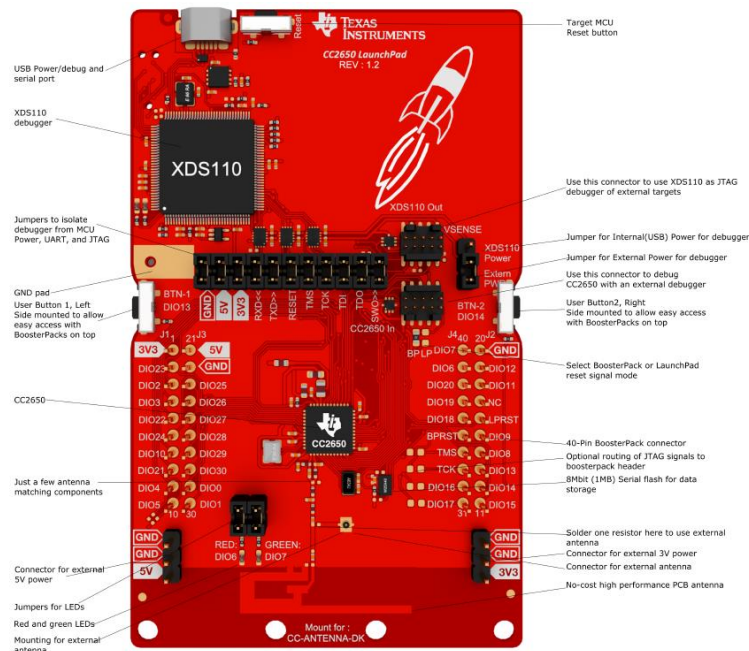


Hình 4.8: Tab năng lượng còn lại

4.3.2 Phần cứng

Kit phát triển LaunchPad CC2650

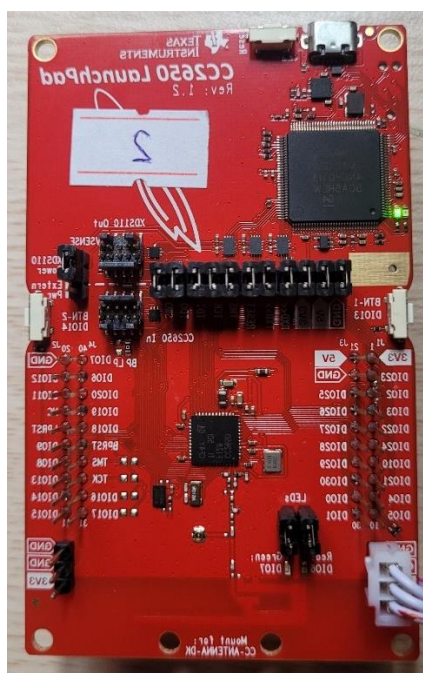
- a. Tổng quan: LAUNCHXL-CC2650 là bộ công cụ phát triển của Texas Instrument dành cho CC2650 SoC MCU. Đây là một SoC (System on Chip) chứa bộ xử lý ARM® Cortex®-M3 32 bit chạy ở tốc độ 48 MHz làm bộ vi điều khiển chính và nhiều bộ ngoại vi phong phú. CC2650 tích hợp bộ thu phát radio 2.4-GHz tương thích với tiêu chuẩn Bluetooth Low Energy(BLE) 4.2 và IEEE 802.15.4.



Bảng 4.3: Thông số kỹ thuật kit LAUNCHPAD-CC2650

Thông số	Giá trị
Vi điều khiển	ARM Cortex-M3
Tần số hoạt động tối đa	48MHz
Bộ nhớ	8KB SRAM bộ nhớ đệm 20KB SRAM dùng chung
Ngoại vi	4 bộ đếm 32-bit, ADC 12-bit, Analog Comparator, 32 GPIOs
Giao tiếp	UART, SPI, I2C, I2S
Công suất phát radio	Tối 5 dBm
Độ nhạy	-97 dBm cho BLE và -100 dBm cho 802.15.4
Điện áp hoạt động	1.8- 3.8V
Dòng tiêu thụ	Active-Mode RX: 5.9 mA Active-Mode TX at 0 dBm: 6.1 mA Active-Mode TX at +5 dBm: 9.1 mA Active-Mode MCU: 61 μ A/MHz

Kit phát triển LAUNCHPAD-CC2650 được hỗ trợ bởi Contiki OS, do đó có thể dễ dàng phát triển phần mềm cho thiết bị. Ngoài ra nhà sản xuất cũng cung cấp nhiều công cụ phần mềm để hỗ trợ lập trình cho kit như: Sensor Controller Studio, SmartRF™ Studio, SmartRF Flash Programmer 2, Code Composer Studio™ . Để có thể hoạt động độc lập, bộ kit cần được cấp năng lượng từ pin, vì vậy em đã sử dụng một viên pin lithium 18650 để cấp nguồn cho cảm biến hoạt động.



Hình 4.10: Mặt trước của kit LAUNCHPAD-CC2650



Hình 4.11: Kit cc2650 cùng với pin

4.4 Hạn chế

Do hạn chế về số lượng phần cứng (chỉ có 4 kit LAUNCHPAD-CC2650) nên quy mô thực nghiệm thuật toán còn nhỏ. Tuy nhiên các kết quả cũng đã chứng minh được một phần kết quả lý thuyết và mô phỏng.

4.5 Kết luận chương

Như vậy qua chương 4 em đã trình bày phương án mô phỏng và thử nghiệm các thuật toán định tuyến, cụ thể là lập trình bằng hệ điều hành Contiki OS, mô phỏng trên phần mềm Cooja, thực nghiệm trên kit phát triển CC2650. Các kết quả thu được sẽ được trình bày trong chương 5 và 6.

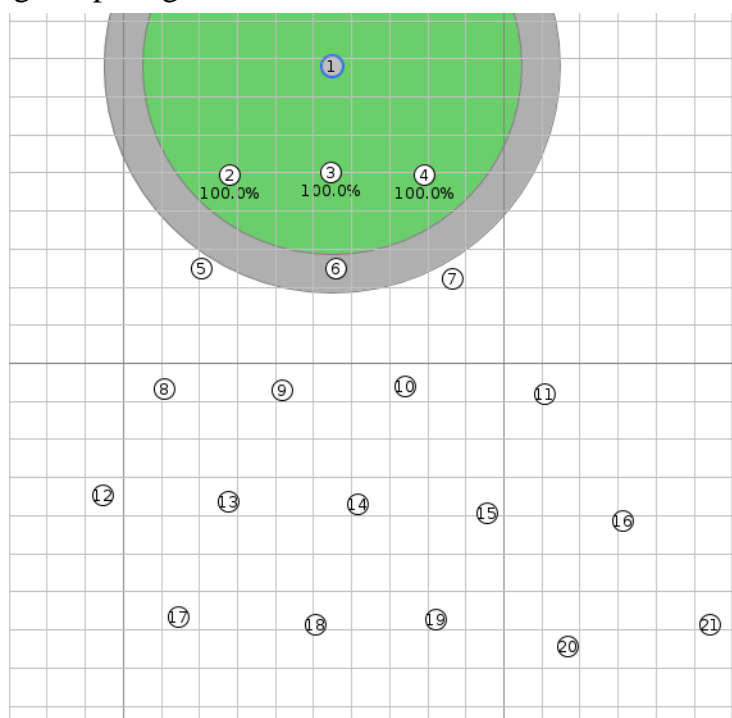
CHƯƠNG 5. MÔ PHỎNG MỘT SỐ THUẬT TOÁN ĐỊNH TUYẾN

Trong thiết kế hệ thống WSN, mô phỏng là bước quan trọng. Nó giúp đánh giá hiệu suất, các quá trình mạng và tối ưu thiết kế mạng. Đồng thời, mô phỏng cũng hỗ trợ kiểm tra độ tin cậy, ước lượng tài nguyên và đánh giá tính bảo mật. Bằng cách này, người thiết kế có thể phát hiện và giải quyết vấn đề trước khi triển khai, giảm thiểu rủi ro và chi phí trong quá trình triển khai thực tế. Trong chương này, em mô phỏng mạng theo những kịch bản khác nhau, từ đó đưa ra các đánh giá về hiệu năng của từng giao thức định tuyến RPL cải tiến trong các điều kiện khác nhau. Các kết quả thu được sẽ được minh họa và đánh giá.

5.1 Kịch bản mô phỏng

5.1.1 Cấu trúc mạng

a. Mạng mô phỏng

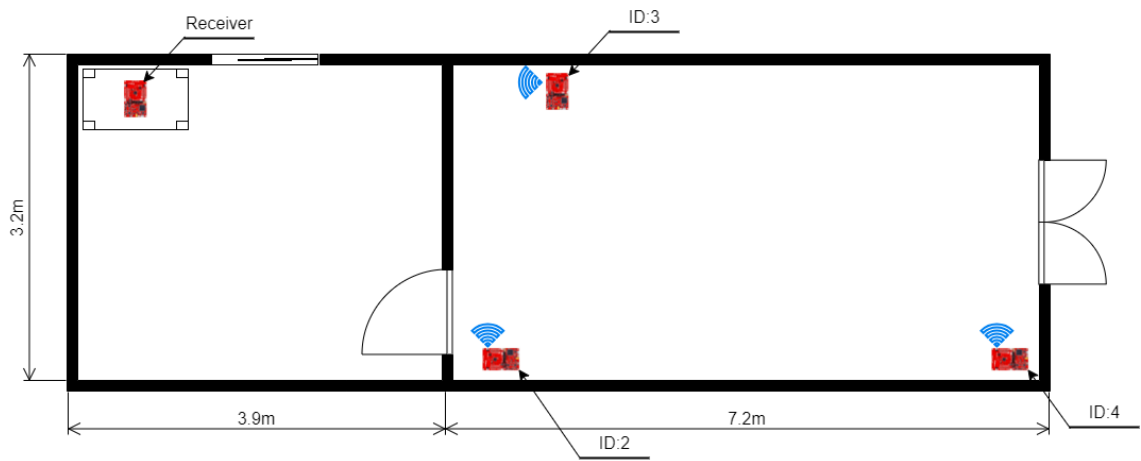


Hình 5.1 Cấu trúc mạng mô phỏng

Vị trí các nút được sắp xếp như trên hình Hình 5.1 . Có tổng cộng 21 nút, sắp xếp theo sơ đồ hình cây với nút gốc được đặt ở đỉnh cây. Mục đích của việc bố trí như vậy là để tạo ra sự phân tầng giữa các nút, tăng số bước nhảy các nút cần để gửi gói tin đến nút gốc. Các nút được bố trí lệch sang phải một chút để tạo sự mất cân bằng trong mạng.

b. Mạng thực nghiệm

Có 4 module cảm biến được bố trí như hình vẽ:



Hình 5.2: Sơ đồ bố trí cảm biến

5.1.2 Các kịch bản mô phỏng

a. Mô phỏng

Hai kịch bản mô phỏng được thực hiện. Thứ nhất, các nút gửi tin đều đặn về nút gốc với chu kỳ mỗi 30s một tin nhắn. Kịch bản này mô phỏng hoạt động bình thường của mạng với lưu lượng gói tin thấp.

Kịch bản thứ hai cho các nút gửi tin nhắn với tần suất lớn, mỗi 5s một tin nhắn. Kịch bản này thử nghiệm hoạt động của mạng trong điều kiện lưu lượng lớn, độ tắc nghẽn cao.

b. Thử nghiệm thực tế

Các nút gửi sẽ gửi tin nhắn chứa thông tin trạng thái của mình về cho nút nhận với chu kỳ 3s một lần. Việc bố trí như trên nhằm mục đích buộc nút 4 (ID=4) chọn giữa nút 2 và 3 làm nút cha ưu tiên. Nút được chọn làm nút cha ưu tiên sẽ có mức tiêu thụ cao hơn và nhanh hết pin hơn. Do đó thuật toán định tuyến cần phải có khả năng chuyển đổi qua lại giữa 2 nút cha để cân bằng năng lượng tiêu thụ trên các nút.

5.1.3 Tham số mô phỏng

Các tham số được sử dụng trong mô phỏng như trong bảng dưới đây. Các hệ số điều chỉnh được chọn bằng cách chạy nhiều mô phỏng để chọn ra kết quả tốt nhất cho mỗi thuật toán.

Bảng 5.1: Bảng tham số mô phỏng

Tham số	Giá trị
Phần mềm mô phỏng	Cooja, Contiki 3.0
Số nút	21 nút, 1 nút nhận, 20 nút gửi
Chu kỳ gửi	30s, 5s
Thời gian mô phỏng	2000s, 360s
Transmision Range	50m
Interference Range	60m
K	17

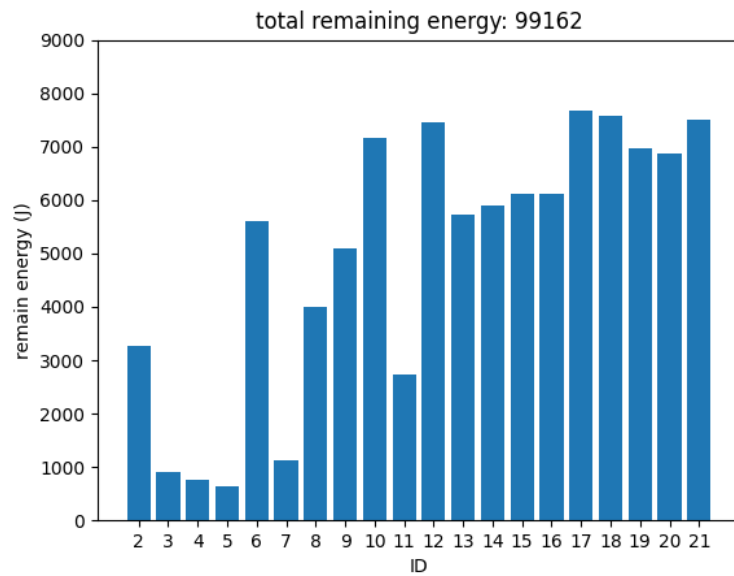
Kr	220
Kq	100

5.2 Kết quả mô phỏng

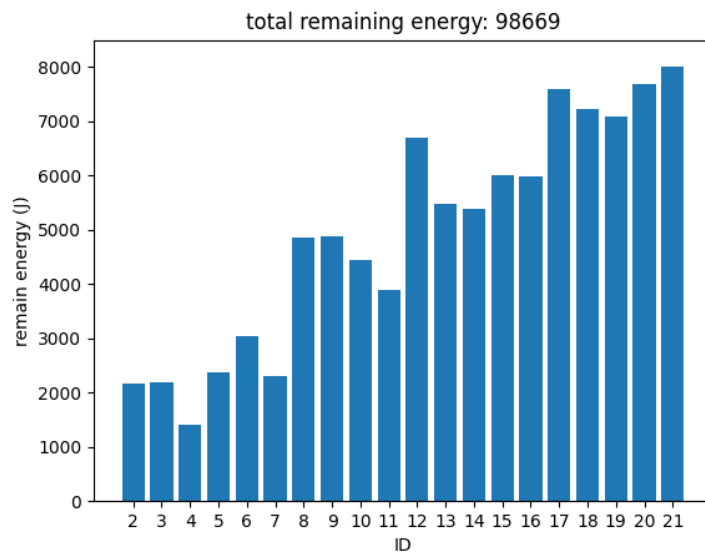
5.2.1 Kịch bản lưu lượng thấp

Mô phỏng được thực hiện với tần suất gửi tin 30s/ 1 tin nhắn trong vòng 2000s.

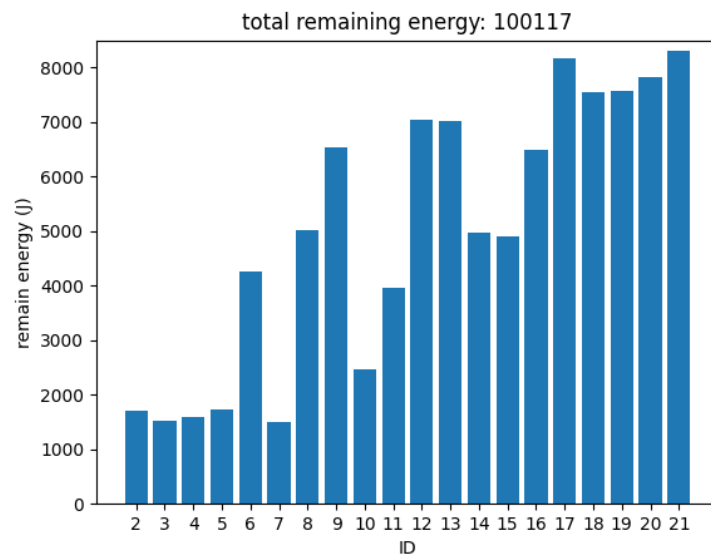
a. Phân bố năng lượng còn dư sau 2000s mô phỏng



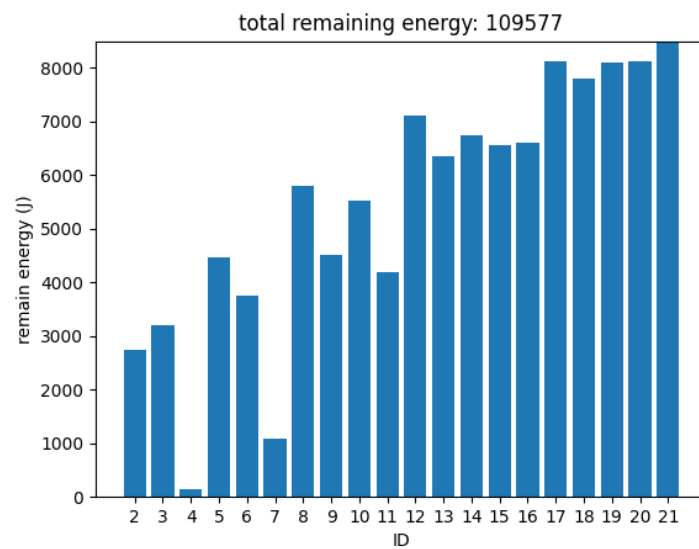
Hình 5.3: Phân bố năng lượng khi chỉ sử dụng ETX



Hình 5.4 Phân bố năng lượng còn lại khi sử dụng ETX kết hợp năng lượng còn dư



Hình 5.5 Phân bố năng lượng còn lại khi sử dụng ETX-RDC



Hình 5.6 Phân bố năng lượng còn lại khi sử dụng ETX-QU

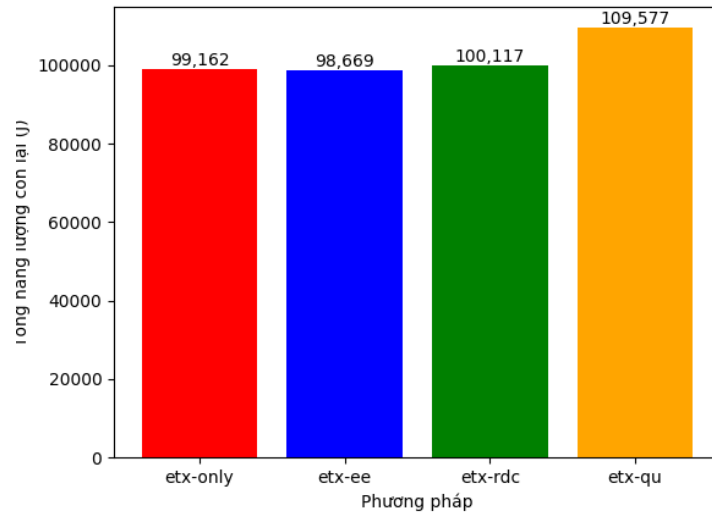
Bảng 5.2: Số bản tin được trung chuyển qua từng nút

Nút	RPL-ETX	ETX-EE	ETX-RDC	ETX-QU
2	153	310	310	234
3	367	389	330	203
4	350	423	389	420
5	387	290	360	109
6	211	269	164	129
7	483	330	386	387
8	137	182	178	125
9	132	153	35	136
10	114	105	256	102
11	235	152	133	147
12	12	22	34	35
13	117	101	37	41
14	98	71	93	39
15	104	74	105	51
16	84	70	71	47
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0

Nhận xét: Từ kết quả mô phỏng ta thấy rằng phân bố năng lượng còn lại trong trường hợp chỉ sử dụng ETX có sự mất cân bằng lớn giữa các nút cùng cấp. Nút 3, 4, 5 hết năng lượng rất nhanh trong khi nút 2 vẫn còn rất nhiều năng lượng. Sự mất cân bằng cũng xảy ra đối với các lớp tầng dưới như nút 7 và 11 do các nút này phải làm nút cha của nhiều nút hơn. Điều này có thể thấy được trong Bảng 5.2, số bản tin trung chuyển qua nút số 7 và 11 có sự chênh lệch lớn so với các nút cùng cấp.

Trong khi đó với thử nghiệm sử dụng ETX kết hợp năng lượng còn dư và ETX kết hợp RDC, phân bố năng lượng được cải thiện đáng kể. Còn đối với ETX kết hợp với tỉ lệ sử dụng hàng đợi, phân bố năng lượng hầu như không được cải thiện, với nút 4 hết năng lượng rất nhanh trong khi nút 2, 3 vẫn còn nhiều. Có thể giải thích điều này bởi trong điều kiện lưu lượng thấp, các tin nhắn được chuyển tiếp rất nhanh, vì vậy chỉ số tỉ lệ sử dụng hàng đợi rất nhỏ. Vì vậy thuật toán sử dụng ETX kết hợp tỉ lệ sử dụng hàng đợi không hoạt động hiệu quả trong điều kiện tải thấp.

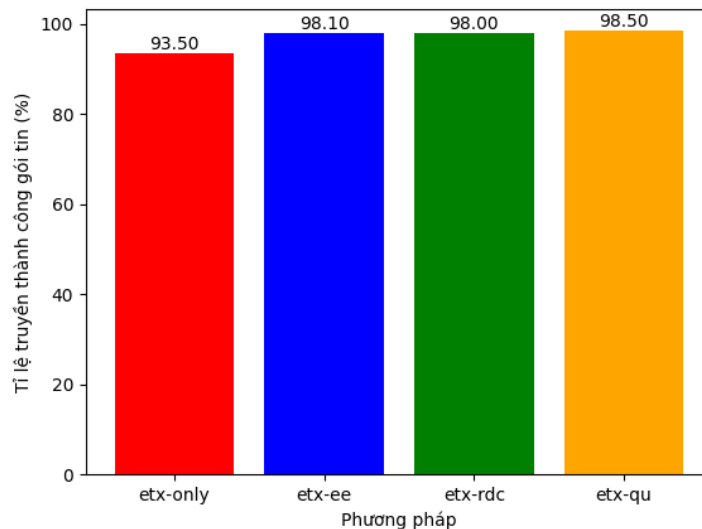
b. Tổng năng lượng còn dư



Hình 5.7: Tổng năng lượng còn lại

Tổng năng lượng còn lại của tất cả các nút trong các thuật toán là gần như nhau. Do các thuật toán định tuyến chỉ thay đổi đường đi của tin nhắn nên không ảnh hưởng đến tổng chi phí năng lượng cần sử dụng.

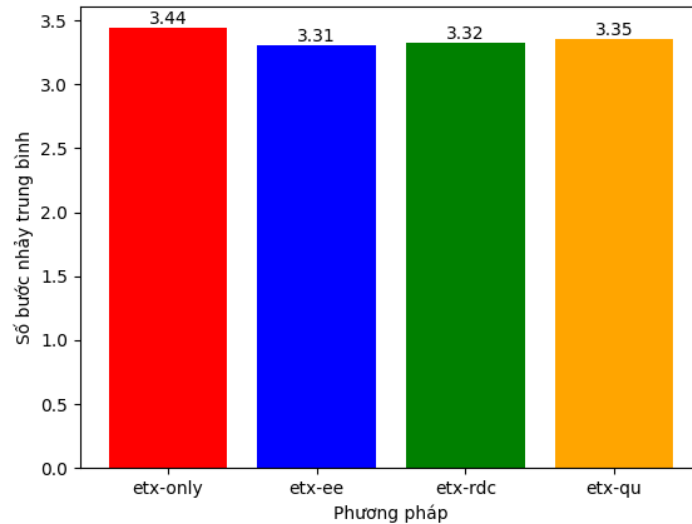
c. Tỷ lệ truyền gói tin thành công



Hình 5.8: Tỷ lệ truyền gói tin thành công

Kết quả cho thấy tỷ lệ truyền tin thành công trong các thuật toán cải tiến tăng nhẹ so với thuật toán RPL-ETX tiêu chuẩn. Điều này có được là nhờ các thuật toán cải tiến làm giảm số lượng tin nhắn bị mất do tắc nghẽn đường truyền.

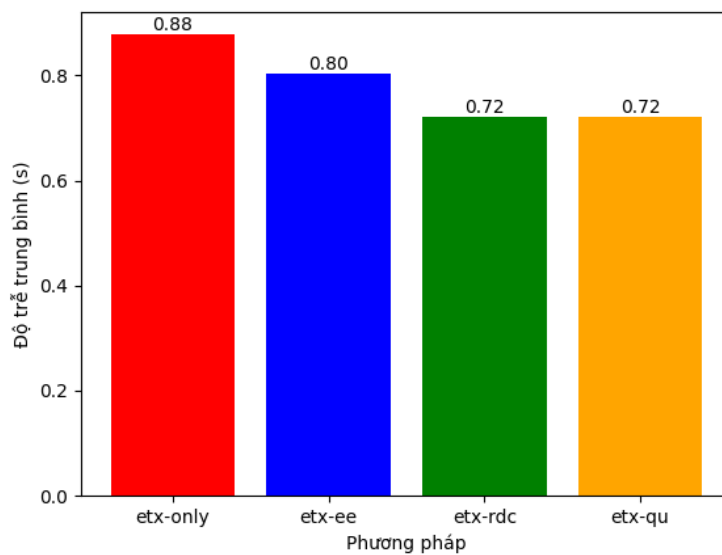
d. Số bước nhảy trung bình



Hình 5.9 Số bước nhảy trung bình

Số bước nhảy trung bình không thay đổi nhiều giữa các thuật toán do cấu trúc mạng hình cây phân lớp, vì vậy khi đường đi thay đổi nhưng số nút cần đi qua vẫn không đổi (bằng số lớp).

e. Độ trễ trung bình của các nút từ 17 đến 21



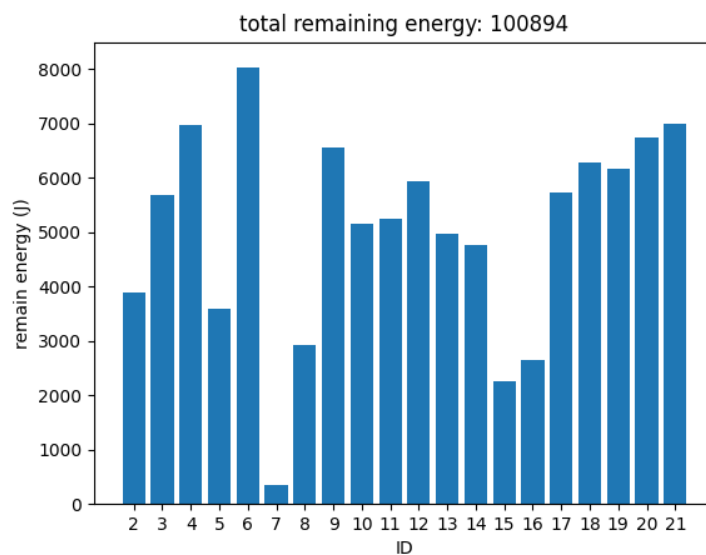
Hình 5.10 Độ trễ trung bình

Độ trễ trong các thuật toán cải tiến thấp hơn so với thuật toán tiêu chuẩn nhờ giải quyết được vấn đề tắc nghẽn tại các nút trung gian, thời gian chờ xử lý tại các nút trung gian thấp hơn.

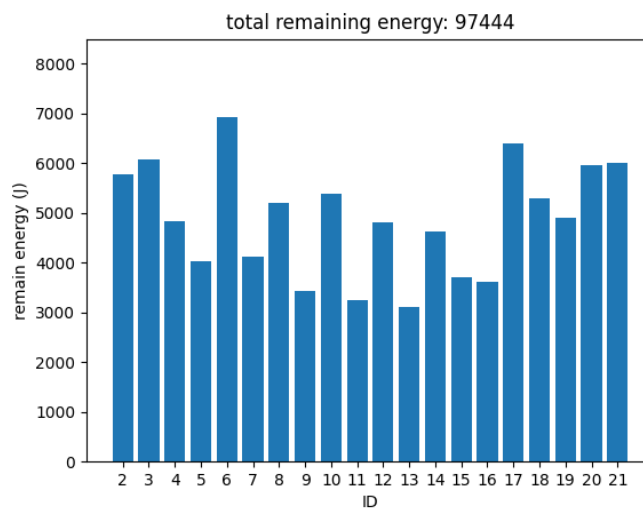
5.2.2 Kịch bản lưu lượng cao

Mô phỏng được thực hiện với tần suất gửi tin 5s/ 1 tin nhắn trong vòng 360s.

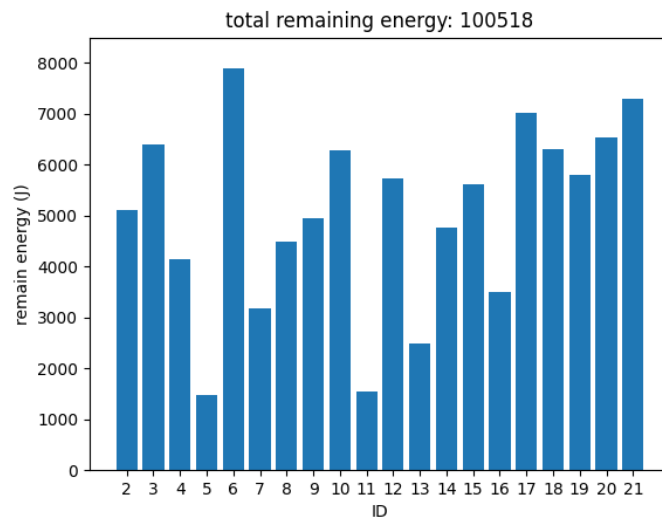
a. Phân bố năng lượng còn dư sau 360s mô phỏng



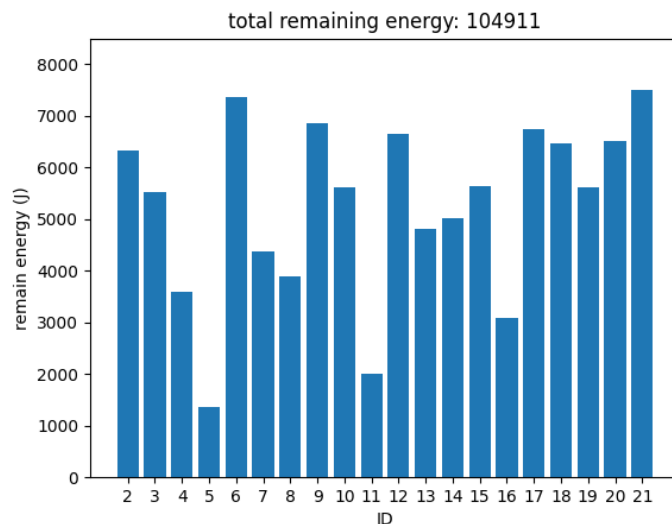
Hình 5.11 Phân bố năng lượng còn lại khi chỉ sử dụng ETX với lưu lượng lớn



Hình 5.12 Phân bố năng lượng còn lại khi sử dụng ETX kết hợp năng lượng còn dư với lưu lượng lớn



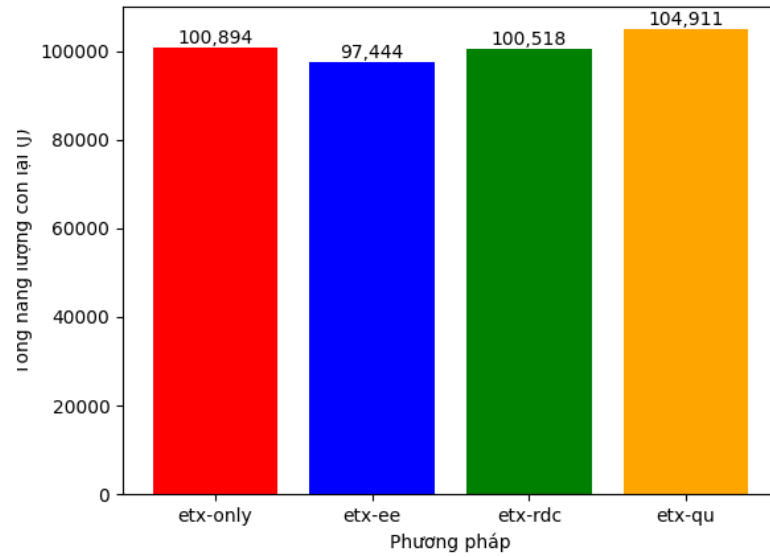
Hình 5.13 Phân bố năng lượng còn lại khi sử dụng ETX-RDC khi lưu lượng lớn



Hình 5.14 Phân bố năng lượng còn lại khi sử dụng ETX-QU khi lưu lượng lớn

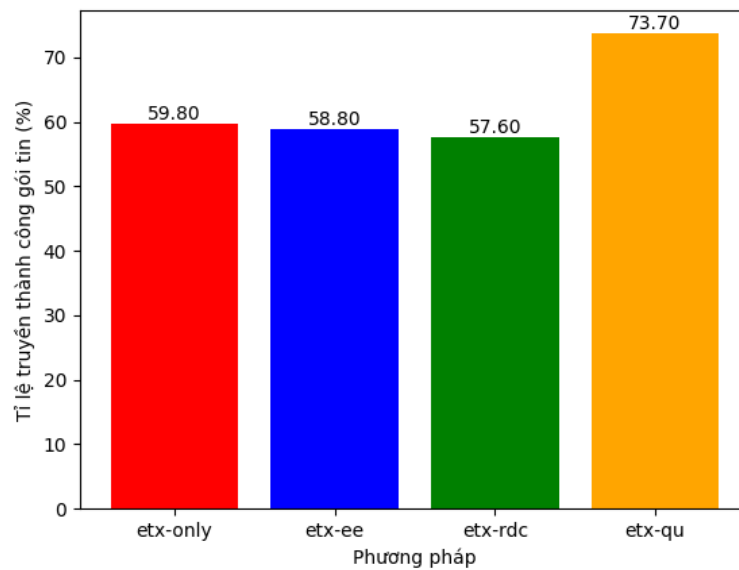
Nhận xét: Trong kịch bản lưu lượng lớn, các thuật toán rpl cải tiến vẫn có khả năng cải thiện phân bố năng lượng đáng kể so với chỉ sử dụng ETX tiêu chuẩn. Thuật toán ETX kết hợp năng lượng còn dư vẫn hoạt động tốt nhất.

b. Tổng năng lượng còn lại



Hình 5.15: Tổng năng lượng còn lại

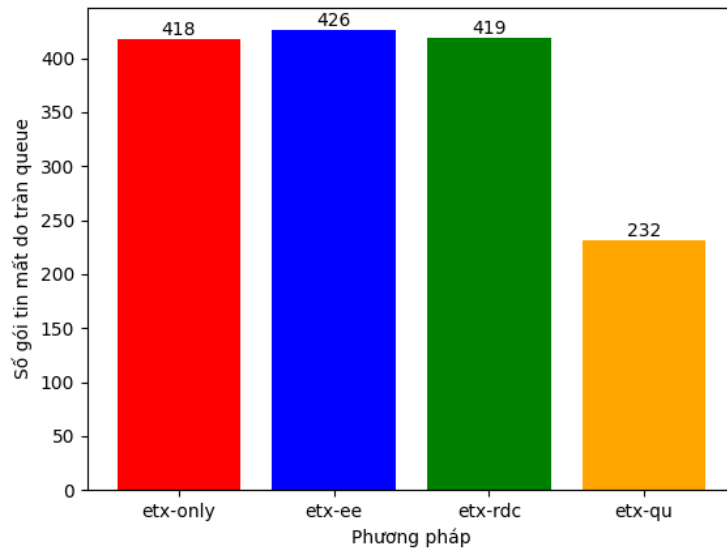
c. Tỷ lệ truyền gói tin thành công



Hình 5.16: Tỷ lệ truyền gói tin thành công

Nhận xét: Tỷ lệ truyền gói tin thành công trong các thuật toán đều giảm mạnh so với kịch bản lưu lượng thấp. Ba thuật toán ETX tiêu chuẩn, ETX kết hợp năng lượng còn dư, ETX kết hợp rdc là như nhau trong khi thuật toán ETX-QU cho tỷ lệ truyền tin thành công cao hơn khoảng 23%. Điều này chứng tỏ vấn đề tràn hàng đợi là một trong những nguyên nhân chính gây mất bản tin và thuật toán ETX-QU hoạt động tốt trong điều kiện lưu lượng tải lớn, tắc nghẽn mạng.

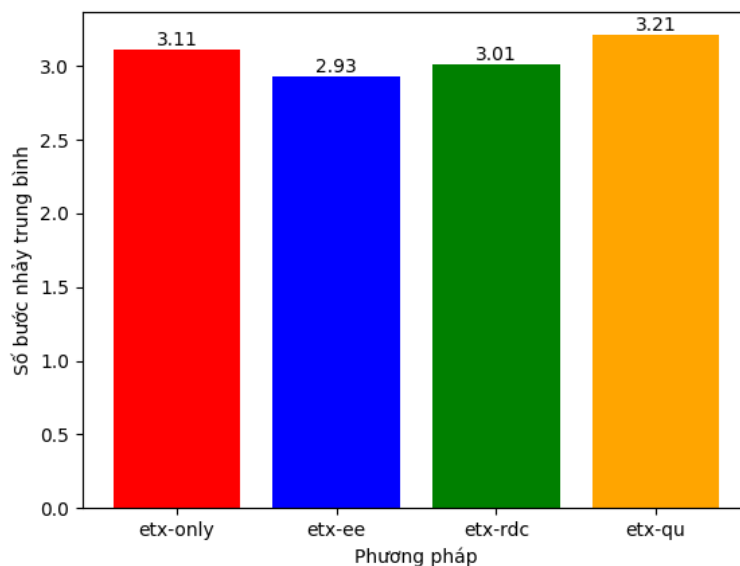
d. Số gói tin mất do tràn queue



Hình 5.17: Số gói tin mất do tràn queue

Nhận xét: Nhờ việc xem xét tỉ lệ sử dụng queue như một chỉ số để chọn nút cha ưu tiên, thuật toán ETX-QU cho thấy hiệu quả trong việc phân bố tải, làm giảm tỉ lệ mất gói tin do tràn hàng đợi lên tới 45% so với các thuật toán định tuyến dựa trên RPL khác. Đối với thuật toán ETX-EE và ETX-RDC, số gói tin mất không đổi do không giải quyết được vấn đề chính là tràn hàng đợi tin nhắn.

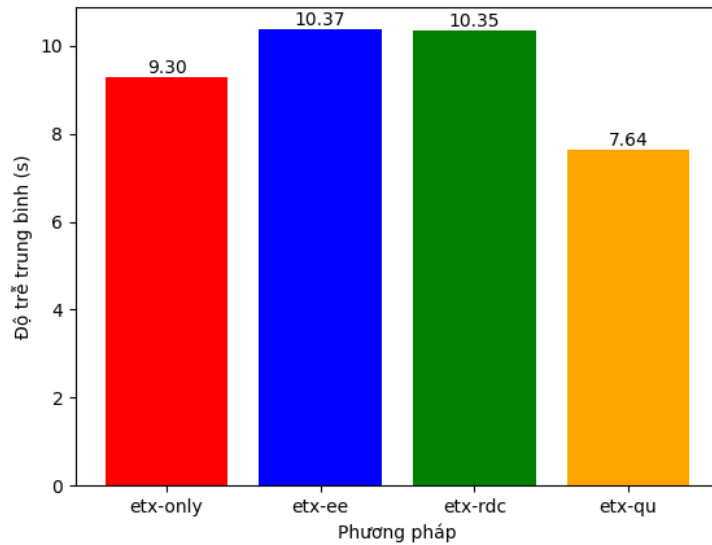
e. Số bước nhảy trung bình



Hình 5.18: Số bước nhảy trung bình

Số bước nhảy trung bình không thay đổi nhiều giữa các thuật toán do cấu trúc mạng hình cây phân lớp, vì vậy khi đường đi thay đổi nhưng số nút cần đi qua vẫn không đổi (bằng số lớp).

f. Độ trễ trung bình của các nút từ 17 đến 21



Hình 5.19: Độ trễ trung bình

Nhận xét: Độ trễ truyền tin trong thuật toán ETX-QU cũng thấp hơn so với các thuật toán khác (thấp hơn 18% so với thuật toán etx tiêu chuẩn) do giải quyết được vấn đề tắc nghẽn tại các nút trung gian, các tin nhắn được xử lý nhanh hơn tại nút trung gian.

5.3 Kết luận chương

Như vậy, từ các kết quả mô phỏng, chúng ta có thể thấy các thuật toán định tuyến cải tiến dựa trên RPL có khả năng cải thiện các nhược điểm mà thuật toán RPL-ETX gặp phải như vấn đề phân bố năng lượng, mất bản tin, ... Trong đó, thuật toán định tuyến sử dụng etx kết hợp năng lượng còn dư hoạt động tốt về mặt phân bố năng lượng trong cả 2 kịch bản. Trong khi đó thuật toán sử dụng ETX kết hợp tỉ lệ sử dụng hàng đợi hoạt động tốt với mạng gặp vấn đề tắc nghẽn nặng, lưu lượng lớn. Thuật toán ETX-QU giúp tăng tỉ lệ truyền thành công, giảm số bản tin mất do tràn hàng đợi. Còn thuật toán ETX kết hợp với rdc cho hiệu quả kém hơn, tuy nhiên có ưu điểm là hoạt động không phụ thuộc vào loại phần cứng của nút (ETX-EE cần biết giá trị dòng điện hoạt động trong từng chế độ). Việc lựa chọn triển khai thuật toán nào là tùy thuộc vào yêu cầu của từng bài toán mà người dùng yêu cầu.

CHƯƠNG 6. THỬ NGHIỆM THỰC TẾ

Thử nghiệm thực tế được thực hiện nhằm kiểm tra xem các kết quả mô phỏng có còn phù hợp trong môi trường thật hay không. Trong chương này, em sẽ trình bày về môi trường thử nghiệm, kết quả thử nghiệm và các đánh giá thu được.

6.1 Tham số thử nghiệm

Năng lượng tiêu thụ của mỗi nút được xác định bằng ước lượng phần mềm, đã được trình bày trong 3.1.1. Để xác định năng lượng tiêu thụ của mỗi nút, chúng ta cần biết dòng tiêu thụ trong từng chế độ, sau đó áp dụng công thức (3.1). Cụ thể, các chế độ hoạt động của nút bao gồm:

- Cpu: Trạng thái cpu hoạt động
- Lpm: Trạng thái low power mode, khi cpu vào chế độ ngủ để tiết kiệm năng lượng
- Transmit: Bộ radio hoạt động để gửi dữ liệu
- Listen: Bộ radio lắng nghe dữ liệu gửi từ nút khác

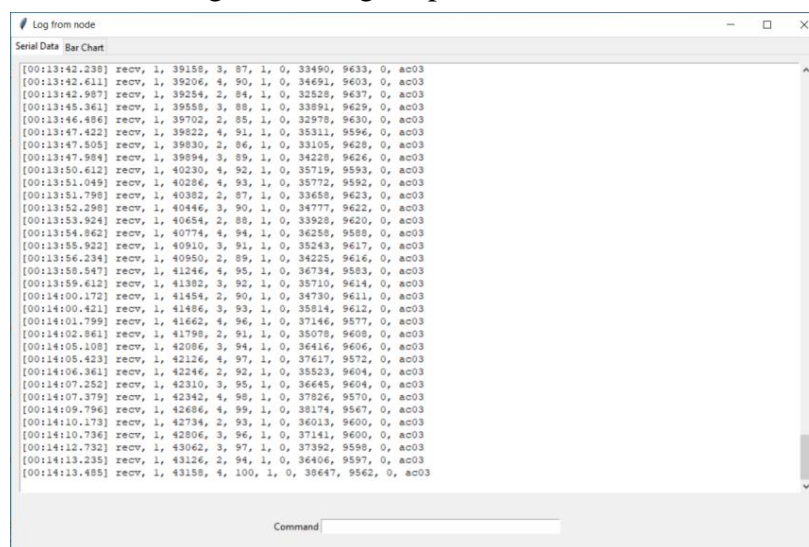
Từ Bảng 4.3 dòng tiêu thụ trong các chế độ của cảm biến như sau:

Bảng 6.1: Dòng tiêu thụ trong từng chế độ của cc2650

Chế độ	Dòng tiêu thụ
Cpu	2.94mA
Lpm	1.2uA
Tx	6.1mA
Rx	5.9mA

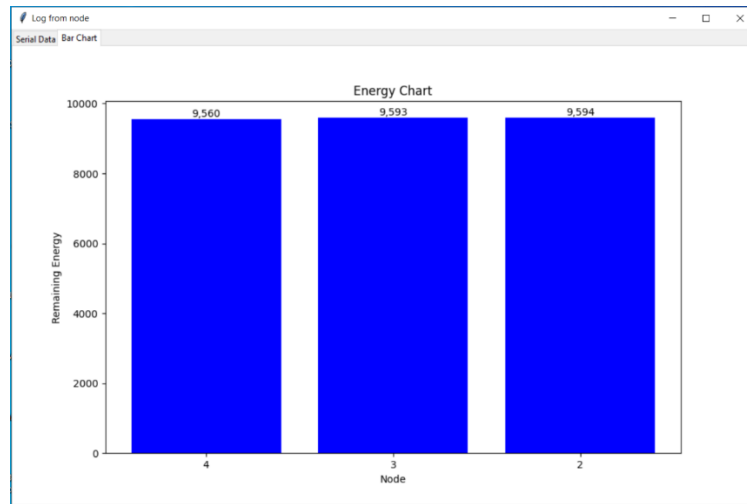
6.2 Kết quả thử nghiệm

Thực hiện tiến hành thử nghiệm trong 60 phút.



Hình 6.1: Giao diện theo dõi

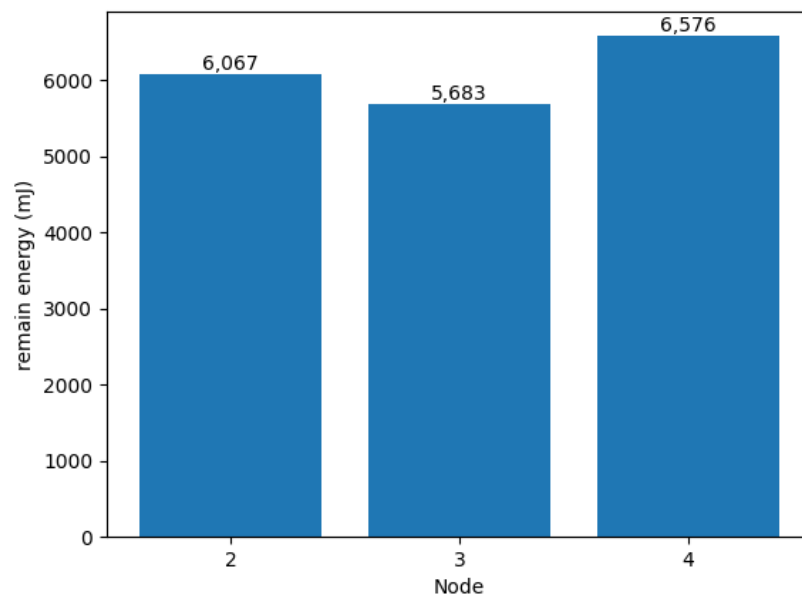
Lượng năng lượng còn lại của mỗi nút được theo dõi theo thời gian thực:



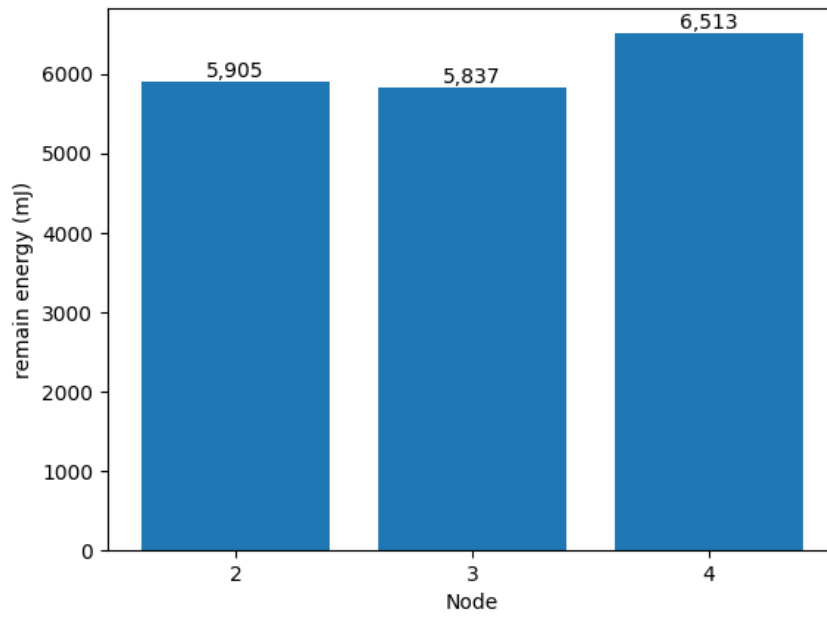
Hình 6.2: Theo dõi năng lượng theo thời gian thực

Các kết quả thu được như sau:

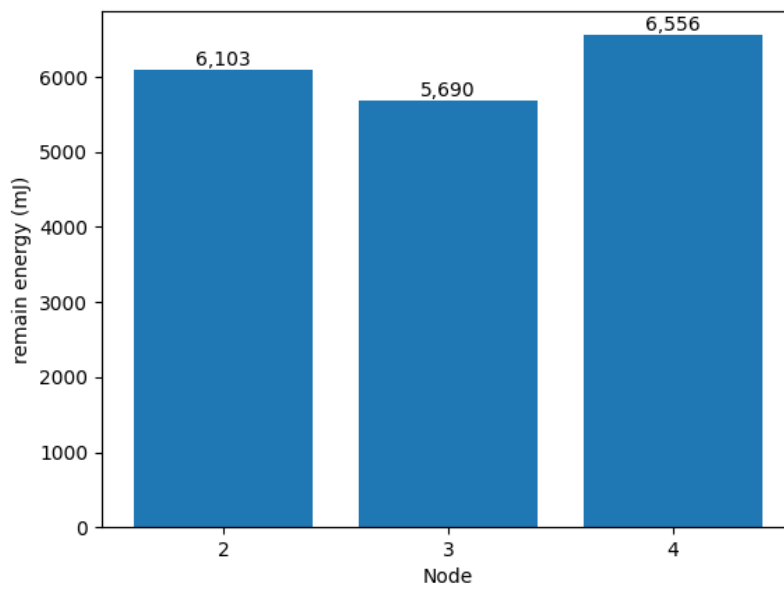
a. Phân bố năng lượng còn lại



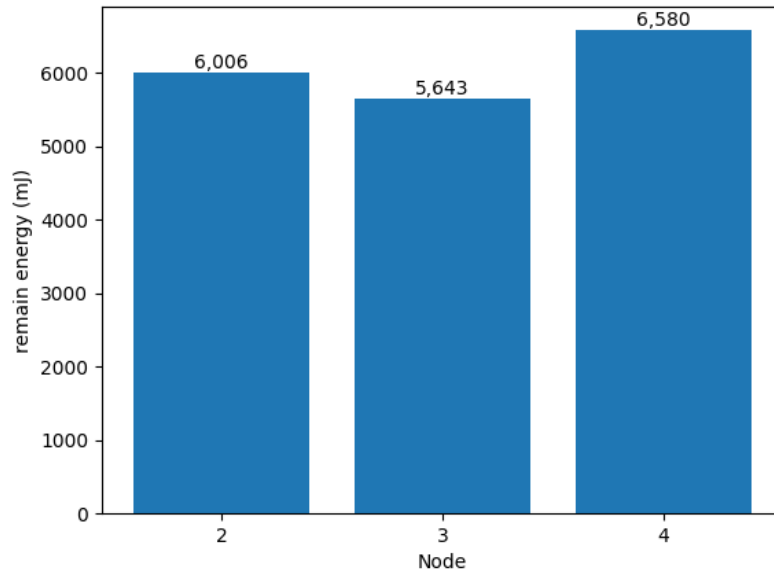
Hình 6.3: Phân bố năng lượng còn lại của RPL-ETX thực nghiệm



Hình 6.4: Phân bố năng lượng còn lại của ETX-EE

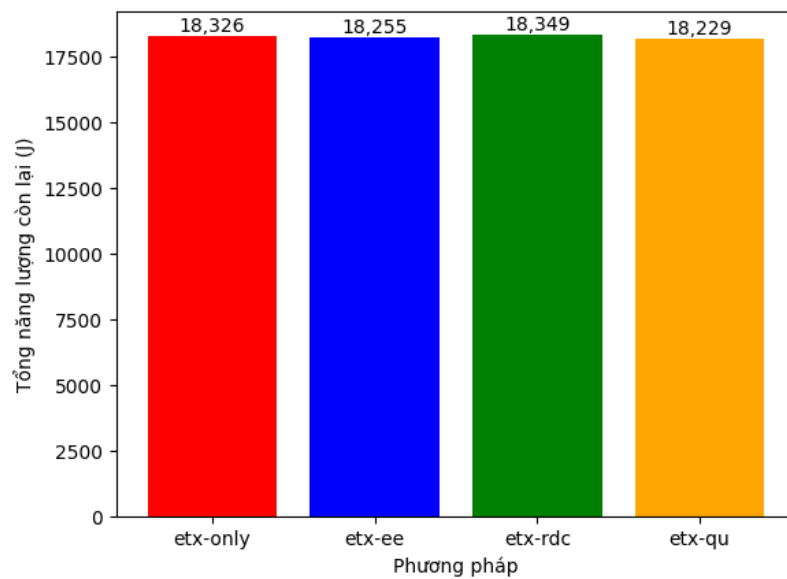


Hình 6.5: Phân bố năng lượng còn lại của ETX-RDC



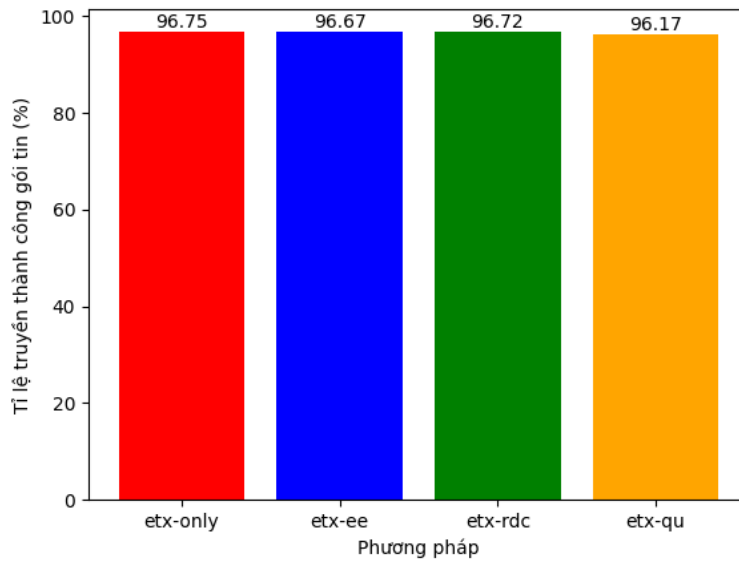
Hình 6.6: Phân bố năng lượng còn lại của ETX-QU

b. Tổng năng lượng còn lại



Hình 6.7: Tổng năng lượng còn lại thực nghiệm

c. Tỷ lệ truyền gói tin thành công



Hình 6.8: Tỉ lệ truyền gói tin thành công thực nghiệm

Nhận xét: Các kết quả thực nghiệm cho thấy thuật toán ETX-EE cân bằng năng lượng tốt hơn các thuật toán còn lại. Trong khi đó thuật toán ETX-RDC và ETX-QU cho kết quả không đổi so với thuật toán ETX tiêu chuẩn. Về tỉ lệ truyền tin thành công các thuật toán cho kết quả tương đương.

Kết quả này là phù hợp với kết quả mô phỏng thu được trong chương 5. Thuật toán ETX-EE sử dụng chỉ số năng lượng còn lại của nút, là một chỉ số tích lũy theo thời gian, sau một khoảng thời gian, chênh lệch năng lượng còn lại giữa các nút đủ lớn sẽ kích hoạt quá trình sửa chữa và các nút sẽ chọn lại nút cha ưu tiên của mình. Trong khi đó với thuật toán ETX-RDC và ETX-QU, các chỉ số được sử dụng là tức thời, do đó trong điều kiện mạng nhỏ và lưu lượng thấp, sự chênh lệch của các chỉ số này nhỏ, so đó không thể làm các nút chọn lại nút cha của mình. Kết quả này là do hạn chế về quy mô thử nghiệm còn nhỏ. Các thuật toán vẫn đảm bảo được tỉ lệ truyền tin tốt.

6.3 Kết luận chương

Mặc dù hạn chế về quy mô thử nghiệm còn nhỏ, nhưng các kết quả chứng tỏ được phần nào lý thuyết và mô phỏng. Thực nghiệm cho thấy thuật toán ETX-EE cân bằng năng lượng tốt trong hầu hết tình huống, trong khi vẫn giữ tỉ lệ thành công gói tin không đổi.

KẾT LUẬN

1. Kết luận

Đồ án đã cơ bản đáp ứng được các yêu cầu đề ra. Đồ án đã giới thiệu thuật toán định tuyến RPL cho mạng cảm biến không dây tốn hao và năng lượng thấp và ứng dụng hệ điều hành mã nguồn mở Contiki OS để xây dựng mạng cảm biến.

Các kết quả mô phỏng đã chứng minh được hiệu quả hoạt động của các thuật toán cải tiến về mặt sử dụng năng lượng, hiệu quả truyền tin. Mô phỏng cũng cho thấy thuật toán dựa trên năng lượng còn dư hoạt động tốt trong hầu hết các kịch bản, về mặt cân bằng phân bố năng lượng tiêu thụ. Trong khi đó, thuật toán sử dụng hàng đợi tin nhắn giúp tăng tỉ lệ truyền thành công, giảm độ trễ và giảm số gói tin mất trong điều kiện lưu lượng lớn.

Tuy nhiên do hạn chế về phần cứng, kết quả thử nghiệm bằng cảm biến thật chưa tốt.

Qua quá trình thực hiện đồ án, em đã có hiểu biết sâu hơn về hoạt động của mạng cảm biến không dây và thuật toán định tuyến RPL. Em cũng học được cách triển khai một hệ điều hành mã nguồn mở cho thiết bị nhúng. Các kỹ năng tìm kiếm tài liệu, tổng hợp thông tin, diễn giải và trình bày cũng được cải thiện trong quá trình làm đồ án.

2. Hướng phát triển của đồ án trong tương lai

Trong tương lai, đề tài sẽ có các hướng phát triển:

- Thử nghiệm trên mạng cảm biến thật với nhiều nút hơn.
- Thử nghiệm với nhiều thuật toán khác.
- Đánh giá các thuật toán với nhiều kịch bản khác(xem xét ảnh hưởng của môi trường biến động, các nút di động, ...)
- Đánh giá các thuật toán với chế độ downward

TÀI LIỆU THAM KHẢO

- [1] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, ‘A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks’, *IEEE Commun. Surv. Tutor.*, vol. 19, no. 2, pp. 828–854, 2017, doi: 10.1109/COMST.2017.2650979.
- [2] R. Alexander *et al.*, ‘RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks’, Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012. doi: 10.17487/RFC6550.
- [3] JP. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, ‘Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks’, RFC Editor, RFC6551, Mar. 2012. doi: 10.17487/rfc6551.
- [4] A. K. Idrees and A. J. H. Witwit, ‘Energy-efficient load-balanced RPL routing protocol for internet of things networks’, *Int. J. Internet Technol. Secur. Trans.*, vol. 11, no. 3, p. 286, 2021, doi: 10.1504/IJTST.2021.114930.
- [5] B. Khemapataphan, A. Kheaksong, K. Thakulsukanant, and W. Lee, ‘Weight ranking mechanism of energy balancing routing metric for RPL protocol in smart grid communications’, in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket: IEEE, Jun. 2017, pp. 640–644. doi: 10.1109/ECTICon.2017.8096319.
- [6] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, ‘Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks’, *IEEE Trans. Mob. Comput.*, vol. 16, no. 4, pp. 964–979, Apr. 2017, doi: 10.1109/TMC.2016.2585107.
- [7] H. Lamaazi and N. Benamar, ‘A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function’, *Ad Hoc Netw.*, vol. 96, p. 102001, Jan. 2020, doi: 10.1016/j.adhoc.2019.102001.
- [8] H. Lamaazi and N. Benamar, ‘RPL enhancement using a new objective function based on combined metrics’, in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain: IEEE, Jun. 2017, pp. 1459–1464. doi: 10.1109/IWCMC.2017.7986499.
- [9] S. Oh, D. Hwang, K. Kim, and K.-H. Kim, ‘A hybrid mode to enhance the downward route performance in routing protocol for low power and lossy networks’, *Int. J. Distrib. Sens. Netw.*, vol. 14, p. 155014771877253, Apr. 2018, doi: 10.1177/1550147718772533.
- [10] R. Alexander *et al.*, ‘RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks’, Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012. doi: 10.17487/RFC6550.
- [11] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao, ‘Energy-Efficient Oriented Routing Algorithm in Wireless Sensor Networks’, in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester: IEEE, Oct. 2013, pp. 3813–3818. doi: 10.1109/SMC.2013.651.
- [12] M. Banh, N. Nguyen, K.-H. Phung, L. Nguyen, N. H. Thanh, and K. Steenhaut, ‘Energy balancing RPL-based routing for Internet of Things’, in

- 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, Jul. 2016, pp. 125–130. doi: 10.1109/CCE.2016.7562624.
- [13] ‘Porting Contiki-NG to new platforms — Contiki-NG documentation’. Accessed: Jun. 24, 2024. [Online]. Available: <https://docs.contiki-ng.org/en/master/doc/programming/Porting-Contiki-NG-to-new-platforms.html>
- [14] ‘LAUNCHXL-CC2650 Development kit | TI.com’. Accessed: Jul. 08, 2024. [Online]. Available: <https://www.ti.com/tool/LAUNCHXL-CC2650>