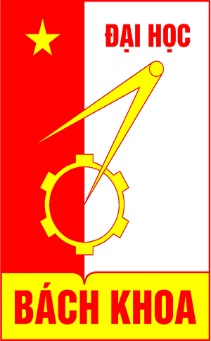
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP**

**THUẬT TOÁN ĐỊNH TUYẾN HIỆU QUẢ CHO MẠNG CẢM BIẾN KHÔNG DÂY**

**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 |
| **Bộ môn:** | Điều khiển tự động |
| **Viện:** | Điện |

**HÀ NỘI, 4/2024**

[Mục tiêu đồ án: 4](#_Toc164004228)

[Kế hoạch thực hiện: 4](#_Toc164004229)

[CHƯƠNG 1. TỔNG QUAN 5](#_Toc164004230)

[1.1. Giới thiệu về mạng cảm biến không dây 5](#_Toc164004231)

[1.1.1. Khái niệm 5](#_Toc164004232)

[1.1.2. Cấu trúc 5](#_Toc164004233)

[1.1.1. Đặc điểm 6](#_Toc164004234)

[1.1.2. Ứng dụng 6](#_Toc164004235)

[1.2. Định tuyến trong mạng cảm biến không dây 7](#_Toc164004236)

[1.2.1. Định tuyến là gì? 7](#_Toc164004237)

[1.2.2. Các thách thức đối với quá trình định tuyến 8](#_Toc164004238)

[1.3. Hệ điều hành Contiki 9](#_Toc164004239)

[1.3.1. Giới thiệu 9](#_Toc164004240)

[1.4. Phần mềm mô phỏng Cooja 10](#_Toc164004241)

[CHƯƠNG 2. THUẬT TOÁN ĐỊNH TUYẾN RPL 10](#_Toc164004242)

[2.1. Giới thiệu thuật toán định tuyến RPL 10](#_Toc164004243)

[2.2. Các khái niệm quan trọng 11](#_Toc164004244)

[2.3. Các loại bản tin điều khiển 12](#_Toc164004245)

[2.3.1. Bản tin DIS 13](#_Toc164004246)

[2.3.2. Bản tin DIO 13](#_Toc164004247)

[2.3.3. Bản tin DAO 15](#_Toc164004248)

[2.3.4. Bản tin DAO-ACK 16](#_Toc164004249)

[2.4. Hoạt động định tuyến sử dụng RPL 17](#_Toc164004250)

[2.4.1. Xây dựng DODAG 17](#_Toc164004251)

[2.4.2. Định tuyến upward 19](#_Toc164004252)

[2.4.3. Định tuyến downward 19](#_Toc164004253)

[2.5. Hàm mục tiêu (Objective Function- OF) 20](#_Toc164004254)

[2.6. Các nghiên cứu cải tiến 21](#_Toc164004255)

[2.6.1. Hàm mục tiêu (OF) 21](#_Toc164004256)

[2.6.2. Cơ chế downward 22](#_Toc164004257)

[2.6.3. Tính bảo mật 23](#_Toc164004258)

[3. Phần mềm mô phỏng cooja 23](#_Toc164004259)

[3.1. Tổng quan: 23](#_Toc164004260)

## Mục tiêu đồ án:

* Đề xuất cải tiến thuật toán định tuyến RPL để nâng cao khả năng tiết kiệm năng lượng.
* So sánh hiệu suất các thuật toán bằng mô phỏng.
* Xây dựng mạch cảm biến không dây thật để đánh giá hiệu suất các thuật toán.

## Kế hoạch thực hiện:

|  |  |  |  |
| --- | --- | --- | --- |
| Tuần | Công việc | Kết quả | Ghi chú |
| 5 | Tìm hiểu tổng quan, lập kế hoạch | Báo cáo tổng quan, bảng kế hoạch |  |
| 6-8, | Tìm Hiểu Về Các Thuật Toán Định Tuyến Phổ Biến (RPL protocol) Hiểu cách hoạt động và ưu/nhược điểm của mỗi thuật toán | Báo cáo tìm hiểu |  |
| Nghiên Cứu Contiki OS và Cách Triển Khai Trên Mạng Cảm Biến | Báo cáo tìm hiểu |  |
| 9-10, | Mô phỏng lại các thuật toán trên Contiki OS Đánh giá, so sánh hiệu năng | Báo cáo so sánh với các thuật toán và đánh giá hiệu suất |  |
| 10-12, | Phân tích kết quả, tìm điểm yếu và cải thiện hiệu năng Tối Ưu Hóa và Cải Tiến | Thuật toán cải tiến |  |
| 13 | Thiết kế sơ đồ hệ thống demo Thử nghiệm lại bằng mô phỏng | Sơ đồ chức năng hệ thống File mô phỏng, báo cáo kết quả mô phỏng |  |
| 14 | Thiết kế mạch cảm biến | Sơ đồ nguyên lý, mạch cảm biến |  |
| Lập trình mạch cảm biến | Lưu đồ thuật toán, code mạch cảm biến |  |
|  |
| 15 | Thử nghiệm mô phỏng | Phần cứng hoạt động ổn định |  |  |
| Kiểm Tra, Sửa Lỗi và Hoàn Thiện | Hệ thống thử nghiệm hoạt động hiệu quả |  |  |
| 16 | Tổng hợp báo cáo | Báo cáo word, power point |  |  |
| Hoàn thiện báo cáo | Báo cáo word, power point |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |

# TỔNG QUAN

## Giới thiệu về mạng cảm biến không dây

### Khái niệm

Mạng cảm biến không dây (Wireless Sensor Network - WSN) là một loại mạng không dây được hình thành bởi một số lượng lớn các nút cảm biến có khả năng thu thập thông tin từ môi trường xung quanh và truyền dữ liệu đến một trạm cơ sở (base station) hoặc điểm thu thập dữ liệu (sink node). Mục tiêu chính của mạng cảm biến không dây là thu thập dữ liệu môi trường một cách hiệu quả và chính xác.

A diagram of a cloud with a cloud connected to a sensor

Description automatically generated

Hình 1: Cấu trúc mạng cảm biến không dây điển hình[1]

### Cấu trúc

Cấu trúc của một mạng cảm biến không dây (WSN) 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.

### Đặc điểm

Mạng cảm biến không dây (Wireless Sensor Network - WSN) 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 có khả năng tự tổ chức thành một mạng.
* *Hạn chế về Năng lượng:* 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 có hạn. 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.
* *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.
* *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.

### Ứng dụng

Mạng cảm biến không dây (Wireless Sensor Network - WSN) 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 thế 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.

A diagram of a company's application

Description automatically generated

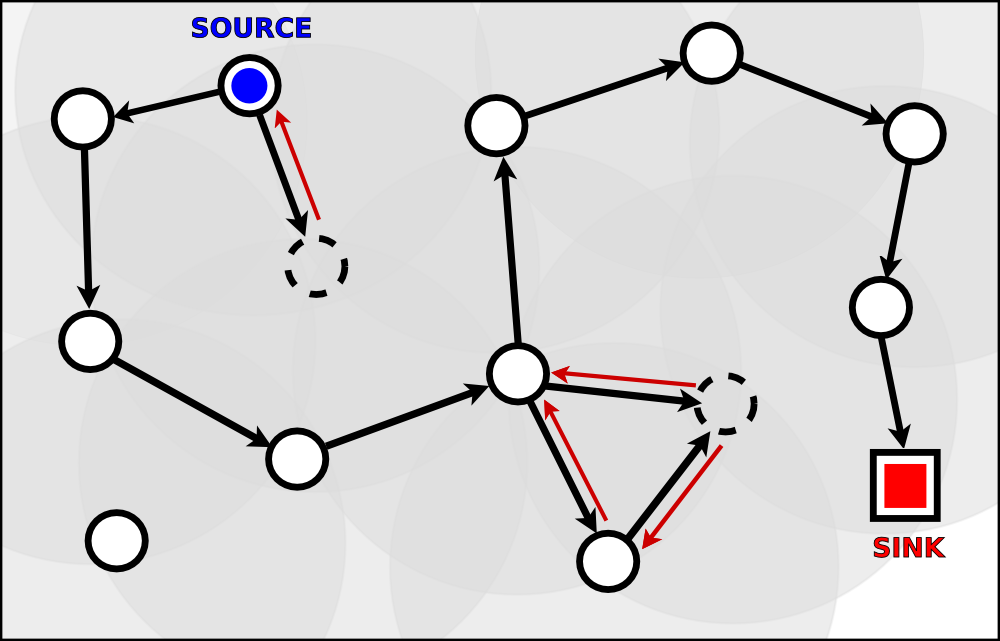
Hình 2: Ứng dụng của mạng cảm biến không dây [2]

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.

## Định tuyến trong mạng cảm biến không dây

### 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 3: Quá trình truyền tin qua nhiều nút trung gian

Trong mạng cảm biến không dây (Wireless Sensor Network - WSN), 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.

### Các thách thức đối với quá trình định tuyến

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 (Wireless Sensor Network - WSN) 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.
* *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.
* *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.

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.

## Hệ điều hành Contiki

### Giới thiệu

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: 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 động 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 mạng 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.

## Phần mềm mô phỏng Cooja

# THUẬT TOÁN ĐỊNH TUYẾN RPL

## Giới thiệu thuật toán định tuyến RPL

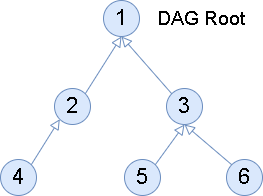
RPL (Routing Protocol for Low-Power and Lossy Networks) là một thuật toán định tuyến được thiết kế đặc biệt cho mạng cảm biến không dây và mạng IoT (Internet of Things) có tài nguyên hạn chế về năng lượng, băng thông và có thể xảy ra mất mát dữ liệu. Thuật toán này giúp xác định đường dẫn hiệu quả giữa các nút trong mạng để truyền thông dữ liệu một cách hiệu quả nhất, tiết kiệm năng lượng và đảm bảo độ tin cậy.

*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.

## Các khái niệm quan trọng

* **DAG:** Directed Acyclic Graph. 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.
* **Root:** DAG root là một nút trong DAG không có cạnh đi ra. Bởi vì biểu đồ có tính chất không tuần hoàn nên theo định nghĩa, tất cả các DAG phải có ít nhất một gốc DAG và tất cả các đường dẫn đều kết thúc tại gốc DAG.

****

Hình 5: DAG

* **DODAG(** Distance Oriented Directed Acyclic Graphs**):** Một DAG root từ một đích duy nhất, tức là tại một DAG root duy nhất (DODAG root) không có cạnh đi ra.
* **DODAG Root:** DODAG Root là DAG root của DODAG. DODAG Root có thể hoạt động như một bộ định tuyến biên cho DODAG; đặc biệt, nó có thể tổng hợp các tuyến đường trong DODAG và có thể phân phối lại các tuyến đường DODAG vào các giao thức định tuyến khác.
* **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 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".
* **Down:** Thuật ngữDown đề cập đến hướng từ gốc DODAG tới các nút lá, theo hướng ngược lại của các cạnh DODAG.
* **Rank:** Rank của nút 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.
* **Objective Function:** 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.
* **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.
* **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.
* **DODAG Version**: DODAG Version là một “ phiên bản” tại một lần tái cấu trúc cụ thể của một DODAG gắn với một DODAGID.
* **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.
* **Grounded:** DODAG là grounded khi DODAG root có thể đạt được Goal.
* **Floating DODAG:** Ngược lại với grounded
* **Parent node:** Mỗi nút trong mạng RPL được phân bố 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.

## Các loại bản tin điều khiển

Cấu trúc bản tin điều khiển của thuật toán RPL bao gồm ICMPv6 header của gói tin 6LoWPAN, theo sau bởi phần thân tin nhắn. Phần thân tin nhắn gồm một số trường cố định và thêm một số options.  
Cấu trúc thân bản tin điều khiển như sau:

A diagram of a code

Description automatically generated with medium confidence

Hình 6: RPL Control message[3]

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)

### Bản tin DIS

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

Cấu trúc bản tin DIS như sau:

A close-up of a reserved sign

Description automatically generated

Hình 7: Cấu trúc bản tin DIS[3]

* + Flag: Trường 8 bit không sử dụng dành riêng cho flag. Trường này PHẢI được node gửi khởi tạo về 0 và PHẢI bị node nhận bỏ qua.
  + Reserved: Trường 8 bit không sử dụng. Trường PHẢI được node gửi khởi tạo về 0 và PHẢI bị người node bỏ qua.
  + DIS Options: Các options trong bản tin DIS có thể một trong các loại sau:

0x00 Pad1

0x01 PadN

0x07 Solicited Information

### Bản tin DIO

Bản tin DIO( DODAG Information Object) 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:

A white paper with numbers and letters

Description automatically generated with medium confidence

Hình 8: Bản tin DIO[3]

* + 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

### Bản tin DAO

Bản tin DAO được sử dụng bởi các nút lá (leaf nodes) 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.

A white background with black text and numbers

Description automatically generated

Hình 9: Bản tin DAO[3]

* + 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

### 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.

A close-up of a number

Description automatically generated

Hình 10: Bản tin DAO-ACK[3]

* 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.

## Hoạt động định tuyến sử dụng RPL

### Xây dựng DODAG

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

* DIS
* DIO
* DAO

A diagram of a diagram of a root system

Description automatically generated

Hình 11: 3 bản tin điều khiển

Bản tin DIO

- 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.

A diagram of a diagram

Description automatically generated

Hình 12: Hoạt động của bản tin DIO và DAO

- 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.

Bản tin DIS:

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.

A diagram of a diagram

Description automatically generated

Hình 13: Bản tin DIS

### Định tuyến upward

* Chiều Up 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, nó sẽ chuyển tiếp bản tin lên trên parent của mình, cho đến khi đến được root node.

A diagram of a diagram

Description automatically generated

Hình 14: Upward

### Định tuyến downward

* Chiều down là chiều từ root tới các node lá.
* Các node sẽ gửi bản tin DAO chứa thông tin về các children node của mình theo hướng upward. Root node và các node đóng vai trò là router sẽ sử dụng thông tin này để xây dựng bảng định tuyến.

Storing mode:

* Ở chế độ này, các node không phải node lá đều sẽ lưu trữ bảng định tuyến của riêng mình. Khi node nhận được tin nhắn DAO, nó sẽ sử dụng thông tin này để cập nhập bảng định tuyến.
* Khi node nhận được gói tin theo chiều xuống, nó sẽ dựa vào bảng định tuyến của mình để lựa chọn node tiếp theo để chuyển tiếp

A diagram of a rooting table

Description automatically generated

Hình 15: Routing table[4]

Non-storing mode:

* Ở chế độ này, chỉ có Root node 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.

A diagram of a root

Description automatically generated

Hình 16: Non-stoting mode[4]

## Hàm mục tiêu (Objective Function- OF)

Trong thuật toán định tuyến RPL, hàm mục tiêu (Objective Function) 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). 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.

Trong Contiki OS hỗ trợ 2 hàm mục tiêu cơ bản:

* *Objective function zero( OFO):* OF0 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. OF0 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.
* *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 OF0 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 chuyến đổi đường dẫn nếu có đường dẫn có chi phí thấp hơn sau những khoảng thời gian xác định trước.

## Các nghiên cứu 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:

1. 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 node( Node 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.\

Node 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 lí do này, cần phải xem xét năng lượng còn lại của các nút khi thiết kế một hàm mục tiêu mới[5]. Trong nghiên cứu [6], tác giả nhận thấy các nút có rank thấp hơn( gần root node) 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 [7] 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.

1. 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[8]. 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:

ở đó và thỏa mãn điều kiện:

* Sử dụng Fuzzy logic[9]: 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 [9], 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.

### 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 kxhông 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 lí 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 [4] 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.

### Tính bảo mật

## Phần mềm mô phỏng cooja

### Tổng quan:

* Co-simulation of Objects in Java (COOJA) là một phần mềm mô phỏng cho mạng cảm biến không dây. Nó được phát triển như một phần của dự án Contiki OS, một hệ điều hành nhúng phổ biến cho các mạng cảm biến không dây và IoT.
* COOJA được thiết kế để mô phỏng mạng cảm biến không dây, giúp nhà phát triển và nghiên cứu kiểm tra, hiểu và phát triển ứng dụng cho các mạng như ZigBee, 6LoWPAN, và các mạng khác dựa trên giao thức IEEE 802.15.4.

**Ưu điểm:**

- Hỗ trợ Contiki OS: COOJA được tích hợp sâu vào Contiki OS, giúp cho việc phát triển ứng dụng và thử nghiệm trở nên dễ dàng hơn.

- Mô phỏng đa nút: Bạn có thể tạo và chạy một số lượng lớn các nút mạng cảm biến khác nhau trong một mạng mô phỏng, từ đó kiểm tra và đánh giá hiệu suất của các ứng dụng mạng.

- Môi trường mô phỏng linh hoạt: COOJA cung cấp các công cụ để mô phỏng môi trường, giúp người dùng tạo ra các điều kiện mô phỏng thực tế như độ trễ, mất mát gói tin, và điều kiện mạng khác.

- Giao diện dễ sử dụng: Giao diện người dùng của COOJA được thiết kế rõ ràng và dễ sử dụng, cho phép người dùng tùy chỉnh và điều khiển mạng mô phỏng một cách dễ dàng.

**Giao diện:**

**A computer screen shot of a network

Description automatically generated**

Hình 17: Giao diện phần mềm mô phỏng Cooja

* 1. Mô phỏng mạng sử dụng định tuyến RPL

Thực hiện chạy mô phỏng ví dụ tại đường dẫn: “contiki/examples/ipv6/rpl-collect/collect-tree-dense-noloss.csc”

Mô phỏng tạo 25 nút cảm biến không dây để xây dựng DADOG bằng định tuyến RPL.

A grid with many circles and numbers

Description automatically generated

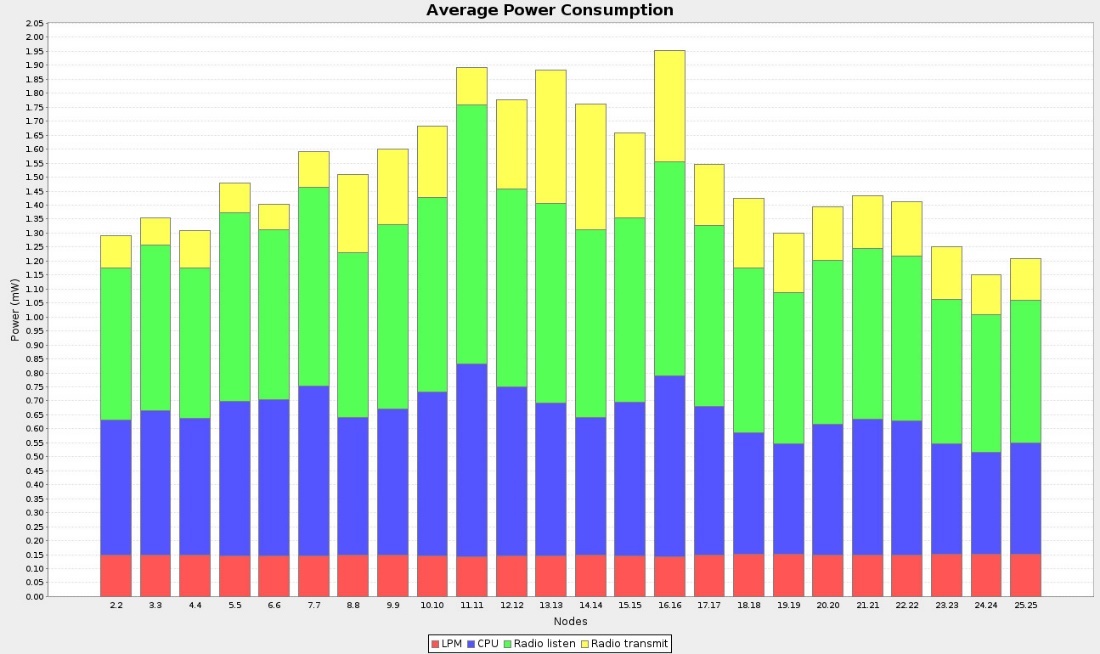
Hình 18: Bản đồ các node

Kết quả mô phỏng:

A screenshot of a computer

Description automatically generated

Hình 19: Đồ thị mạng RPL



Hình 20: Đồ thị năng lượng của các node

A screenshot of a computer

Description automatically generated

Hình 21: Thông tin của các node

TÀI LIỆU THAM KHẢO

[1] S. Manikandan, N. Poongavanam, V. Vivekanandhan, and T. A. Mohanaprakash, ‘Performance Comparison of Various Wireless Sensor Network Dataset using Deep Learning Classifications’, in *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, Karnataka, India: IEEE, Dec. 2022, pp. 1–4. doi: 10.1109/ICMNWC56175.2022.10032015.

[2] 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.

[3] 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.

[4] 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.

[5] 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/IJITST.2021.114930.

[6] 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.

[7] 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.

[8] 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.

[9] 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.