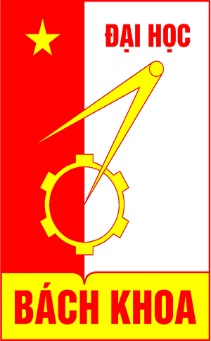
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**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:

* Đề xuất thuật toán mới hoặc cải tiến thuật toán định tuyến đã có để nâng cao khả năng tiết kiệm năng lượng.
* So sánh hiệu suất các thuật toán bằng mô phỏng.
* Xây dựng mạng 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 |  |  |  |  |

BÁO CÁO TUẦN 7

Nội dung:

* Thuật toán định tuyến RPL
* Cách sử dụng cooja để mô phỏng mạng cảm biến không dây

## Thuật toán định tuyến RPL

### Tổng quan

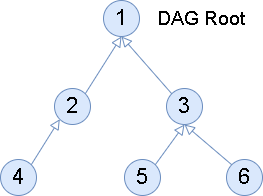
RPL (Routing Protocol for Low-Power and Lossy Networks) là một thuật toán định tuyến được thiết kế đặc biệt cho mạng cảm biến không dây và mạng IoT (Internet of Things) có tài nguyên hạn chế về năng lượng, băng thông và có thể xảy ra mất mát dữ liệu. Thuật toán này giúp xác định đường dẫn hiệu quả giữa 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 1: 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 theo sau bởi 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 2: RPL Control message[1]

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 3: Cấu trúc bản tin DIS[1]

* + 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 4: Bản tin DIO[1]

* + 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 5: Bản tin DAO[1]

* + 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 6: Bản tin DAO-ACK[1]

* 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 7: 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.

A diagram of a diagram

Description automatically generated

Hình 8: 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 9: 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 10: 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 11: Routing table[2]

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 12: Non-stoting mode[2]

## 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**

**Dữ liệu thu từ mô phỏng**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**