**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỆN TỬ**

---------------o0o---------------

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**TIÊU ĐỀ LUẬN VĂN**

**GVHD:**

**SVTH:**

**MSSV:**

**TP. HỒ CHÍ MINH, THÁNG NĂM 20**

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC BÁCH KHOA Độc lập – Tự do – Hạnh phúc.

-----✩----- -----✩-----

Số: \_\_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Điện Tử**

N**HIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. HỌ VÀ TÊN : MSSV:

1. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP :
2. Đề tài:
3. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):
4. Ngày giao nhiệm vụ luận văn: ...............................
5. Ngày hoàn thành nhiệm vụ: ...................................
6. Họ và tên người hướng dẫn: Phần hướng dẫn

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

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

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

*Tp.HCM, ngày…... tháng….. năm 20*

**CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH**

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ):.......................

Đơn vị:......................................................

Ngày bảo vệ : ...........................................

Điểm tổng kết: .........................................

Nơi lưu trữ luận văn: ...............................

***LỜI CẢM ƠN***

*Tp. Hồ Chí Minh, ngày tháng năm .*

**Sinh viên**

**TÓM TẮT LUẬN VĂN**

Luận văn này trình bày về …

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc310380267)

[1.1 Tổng quan 1](#_Toc310380268)

[1.2 Tình hình nghiên cứu trong và ngoài nước 1](#_Toc310380269)

[1.3 Nhiệm vụ luận văn 1](#_Toc310380270)

[2. LÝ THUYẾT 1](#_Toc310380271)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 1](#_Toc310380272)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM (NẾU CÓ) 2](#_Toc310380273)

[5. KẾT QUẢ THỰC HIỆN 2](#_Toc310380274)

[6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 4](#_Toc310380275)

[6.1 Kết luận 4](#_Toc310380276)

[6.2 Hướng phát triển 4](#_Toc310380277)

[7. TÀI LIỆU THAM KHẢO 4](#_Toc310380278)

[8. PHỤ LỤC 4](#_Toc310380279)

DANH SÁCH HÌNH MINH HỌA

[Hình 5‑1 Kết quả thi công 3](#_Toc310380287)

[Hình 5‑2 Kết quả mô phỏng 3](#_Toc310380288)

**DANH SÁCH BẢNG SỐ LIỆU**

[Bảng 1 Thông số hệ thống 3](#_Toc310380293)

# GIỚI THIỆU

## Tổng quan

Ngày nay, các vi điều khiển đã có một bước phát triển mạnh với mật độ tích hợp cao, khả năng xử lý mạnh, tiêu thụ năng lượng ít và giá thành thấp. Khi được nạp phần mềm nhúng, các vi điều khiển này sẽ hoạt động độc lập trong các loại môi trường và những vị trí địa lý khác nhau. Mỗi vi điều khiển khi được tích hợp với bộ thu phát sóng vô tuyến và bộ cảm biến sẽ tạo thành một nút mạng, tập hợp các nút mạng đó trong một phạm vi nhất định, được gọi là mạng cảm biến không dây (WSN - Wireless Senser Network). Với sự phát triển mạnh mẽ của các công nghệ viễn thông như 5G và dự kiến là 6G, các mô hình mạng cảm biến không dây (WSN - Wireless Sensor Networks) cũng đang có những bước tiến đáng kể. Các công nghệ viễn thông mới không chỉ cải thiện hiệu năng của WSN mà còn mở rộng khả năng ứng dụng, làm cho mạng cảm biến trở nên thông minh và linh hoạt hơn. Điều này đồng nghĩa với việc các mô hình WSN sẽ ngày càng được tích hợp vào nhiều khía cạnh của cuộc sống và công nghiệp hơn, đóng góp vào sự phát triển của xã hội số. Mạng cảm biến không dây (WSN - Wireless Senser Network) đã trở thành một thành phần cốt lõi trong các hệ thống Internet vạn vật (IoT), cung cấp các giải pháp tiên tiến cho việc giám sát và kiểm soát trong nhiều lĩnh vực như nông nghiệp thông minh, thành phố thông minh, chăm sóc sức khỏe, và các ứng dụng công nghiệp. Để giải quyết nhu cầu thực tế của xã hội về các giải pháp tiết kiệm năng lượng và hiệu quả trong WSN rất cao. Cụ thể, các ứng dụng như nông nghiệp thông minh, thành phố thông minh và chăm sóc sức khỏe đòi hỏi các hệ thống WSN hoạt động liên tục và hiệu quả năng lượng. Mặc dù có nhiều nghiên cứu và phát triển trong lĩnh vực này, nhưng việc tối ưu hóa các giao thức truyền thông để đạt được hiệu quả năng lượng và độ tin cậy vẫn là một thách thức lớn, cần được làm rõ và thực nghiệm kỹ càng trước khi đưa vào đời sống sản xuất của con người.

Trong khi mạng cảm biến không dây bao gồm một số lượng lớn các nút cảm biến, các nút cảm biến có giới hạn và ràng buộc về tài nguyên. Các thiết bị không dây công suất thấp phải duy trì ngân sách năng lượng nghiêm ngặt để đạt được tuổi thọ nhiều năm. Một số giải pháp đã được đề xuất để giải quyết các yêu cầu và hạn chế của Mạng cảm biến không dây (Wireless Sensor Network - WSN) như băng thông hạn chế và mức tiêu thụ năng lượng thấp. Đặc biệt, rất nhiều nghiên cứu đã được thực hiện để tránh các tình huống trong đó các nút duy trì hoạt động vô tuyến của chúng một cách không cần thiết trong khi không có gói đến nào để nhận. Tình trạng như vậy, được gọi là tình trạng nghe lúc rảnh rỗi, đã được giải quyết bằng cách thiết kế chu kỳ nhiệm vụ (radio duty cycle - RDC) ở lớp liên kết. Các giao thức này buộc mọi nút phải chuyển đổi định kỳ bộ thu phát vô tuyến của nó giữa các khoảng thời gian hoạt động (listen - nghe) ngắn và không hoạt động trong thời gian dài (sleep - ngủ). Chu kỳ nhiệm vụ (radio duty cycle - RDC) đó biểu thị tỷ lệ thời gian mà một thiết bị phát sóng vô tuyến (radio) hoạt động so với tổng thời gian trong một khoảng thời gian xác định. Cụ thể, chu kỳ nhiệm vụ được tính bằng cách chia thời gian hoạt động của radio cho tổng thời gian và thường được biểu diễn dưới dạng phần trăm hoặc tỷ lệ phần trăm.. Có hai loại giao thức chu kỳ nhiệm vụ vô tuyến chính: giao thức không đồng bộ và giao thức đồng bộ. Các giao thức chu trình nhiệm vụ đồng bộ buộc các nút phải đồng bộ hóa thời gian thức dậy của chúng trong khi các giao thức không đồng bộ cho phép mọi nút tự do xác định lịch trình ngủ độc lập của riêng mình. Bằng cách loại bỏ nhu cầu tạo, chia sẻ và duy trì khoảng thời gian hoạt động chung, việc triển khai các giao thức không đồng bộ sẽ ít phức tạp hơn và dẫn đến chi phí giao thức thấp hơn. Do đó, các giao thức không đồng bộ rất phù hợp đến các mạng băng thông thấp nơi chúng đặc biệt tiết kiệm năng lượng.

Một số giao thức không đồng bộ như LPL, B-MAC , WiseMAC , X- MAC, Contiki MAC và gần đây hơn là CSL (Coordinated Sampled Listening). Trong đó thì giao thức Contiki MAC và giao thức CSL là một trong những loại phổ biến nhất do tính chất của chúng tính khả dụng trong Contiki NG. Giao thức Contiki MAC và CSL khác nhau về quy trình truyền tải. Công dụng của Contiki MAC một cách tiếp cận khác: nó liên tục truyền gói dữ liệu đầy đủ cho đến khi được xác nhận bởi người nhận. Việc gửi các gói dữ liệu đầy đủ có thể có vẻ kém hiệu quả về mặt năng lượng hơn, nhưng đáng ngạc nhiên là giao thức Contiki MAC đã được báo cáo là hoạt động tốt hơn giao thức CSL. Các nghiên cứu khác cho thấy Contiki MAC có thể giảm đáng kể mức tiêu thụ năng lượng. Nhưng chúng ta sẽ quan sát cả về sự thành công khi nhận được gói tin của giao thức CSL.

## Nhiệm vụ luận văn

Mục đích của nghiên cứu này là tìm hiểu các giao thức truyền thông hiện có trong WSN, đặc biệt là giao thức thuộc lớp bất đồng bộ MAC như

CSL (Coordinated Sampled Listening). Góp phần vào sự phát triển của lý thuyết và mô hình về giao thức MAC bất đồng bộ cho mạng cảm biến không dây. Phân tích ưu và nhược điểm của các giao thức này về mặt tiêu thụ năng lượng và độ tin cậy của giao thức CSL (Coordinated Sampled Listening) so với các giao thức bất đồng bộ khác (cụ thể là Coniki MAC) trong bối cảnh 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) bằng cách tối ưu hóa cơ chế nghe mẫu để giảm thiểu thời gian hoạt động không cần thiết của radio. Tăng cường độ tin cậy để đảm bảo dữ liệu được truyền tải và nhận một cách chính xác và kịp thời, giảm thiểu mất gói tin. Đánh giá hiệu quả của các giao thức truyền thông hiện có trong các ứng dụng này. Cung cấp dữ liệu cho thực nghiệm tạo ra cơ sở lý thuyết và dữ liệu cho thực nghiệm sau này trong lĩnh vực.

## 1.3 Nội dung nghiên cứu

**Nội dung 1: Tìm hiểu lý thuyết mạng cảm biến không dây** (WSN - Wireless Senser Network), tìm hiểu mạng 6loWPAN, IPv6 và ứng dụng của nó.

**Nội dung 2: Tìm hiểu về những khái niệm của lớp MAC**

**Nội dung 3: Đánh giá phân tích về năng lượng và độ tin cậy** của truyền của lớp MAC trong mạng 6loWPAN thông qua các giao thức Contiki MAC

**Nội dung 4: Sử dụng mô phỏng Cooja trên hệ điều hành nhúng Contiki-NG của mạng cảm biến để đánh giá** 6LoWPAN IPv6 tìm hiểu về giao thức CSL ứng dụng cho đề tài.

## 1.4 Kế hoạch triển khai

|  |  |
| --- | --- |
| **Nội dung, công việc chủ yếu**  (các mốc đánh giá chủ yếu) | **Thời gian** (bắt đầu, kết thúc) |
| Nội dung 1 | Tháng 12 |
| Nội dung 2,3 | Tháng 1 - 4 |
| Nội dung 4 | Tháng 5 |

# LÝ THUYẾT

## Giới thiệu chung về giao thức IP

Giao thức IP (Internet Protocol) là một giao thức hướng dữ liệu được sử dụng bởi các máy chủ nguồn và đích để truyền dữ liệu trong một liên mạng chuyển mạch gói. Dữ liệu trong một liên mạng IP được gửi theo các gói (packet hoặc datagram). Cụ thể, IP không cần thiết lập các đường truyền trước khi một máy chủ gửi các gói tin cho một máy khác mà trước đó nó chưa từng liên lạc đến. Giao thức IP cung cấp một dịch vụ gửi dữ liệu không đảm bảo (còn gọi là cố gắng cao nhất), nghĩa là nó hầu như không đảm bảo gì về gói dữ liệu. Gói dữ liệu có thể đến nơi mà không còn nguyên vẹn, nó có thể đến không theo thứ tự (so với các gói khác được gửi giữa hai máy nguồn và đích đó), nó có thể bị trùng lặp hoặc bị mất hoàn toàn. Nếu một phần mềm ứng dụng cần được bảo đảm, nó có thể được cung cấp từ nơi khác, thường từ các giao thức giao vận nằm phía trên IP.

Cho đến nay đã có hai phiên bản của giao thức IP đó là: IP Version 4 (IPv4) và IP Version 6 (IPv6). Internet hiện nay (và nhiều mạng IP độc lập) chủ yếu dựa trên IPv4 và sử dụng địa chỉ 32-bit, giới hạn không gian địa chỉ chỉ đến 4,294,967,296 địa chỉ duy nhất. Sự cạn kiệt địa chỉ IPv4 đã xảy ra vào ngày 3 tháng 2 năm 2011, mặc dù đã được trì hoãn đáng kể bởi các thay đổi địa chỉ như chuyển đổi địa chỉ mạng (NAT). Giới hạn này của IPv4 đã thúc đẩy sự phát triển của IPv6 từ những năm 1990, và đã được triển khai thương mại từ năm 2006. IPv6 bao gồm một không gian địa chỉ của 2128 và 3.4\*1038 địa chỉ duy nhất. Điều này nên đủ để Internet mở rộng trong những thập kỷ tới - thậm chí với lời hứa của Internet of Things, theo ước tính có thể bao gồm 50 tỷ thiết bị kết nối vào năm 2020. Để nhận ra sự gia tăng trong băng thông, IPv6 tăng kích thước đơn vị truyền tải tối thiểu (MTU) từ 576 lên 1280 byte. IPv6 cũng phản ánh các thay đổi và tiến bộ trong các công nghệ lớp liên kết mà Internet sử dụng. Ethernet là công nghệ liên kết phổ biến và tốc độ truyền dữ liệu của nó đã tăng qua các năm. Wi-Fi phản ánh khả năng của Ethernet với việc hỗ trợ kích thước MTU tương tự và tốc độ truyền dữ liệu rất cao. Cả Ethernet và Wi-Fi hoạt động trong ngữ cảnh của nguồn năng lượng đầy đủ và các thiết bị có khả năng cao. Tuy nhiên, IEEE 802.15.4 được thiết kế để phục vụ một thị trường khác; các ứng dụng lâu dài yêu cầu số lượng lớn thiết bị chi phí thấp, siêu thấp năng lượng. Tốc độ truyền dữ liệu dưới tiêu chuẩn này bị giới hạn ở 250 kbps, và độ dài khung bị giới hạn ở 127 byte để đảm bảo tỷ lệ lỗi gói và lỗi bit thấp trong môi trường RF không đáng tin cậy. Ngoài ra, **IEEE 802.15.4** sử dụng hai địa chỉ: một địa chỉ ngắn 16- bit và một địa chỉ mở rộng EUI-64. Các địa chỉ này giảm thiểu đầu trang và yêu cầu bộ nhớ tối thiểu.

IEEE 802.15.4 là công nghệ truy cập không dây tốc độ dữ liệu thấp, chi phí thấp dành cho các thiết bị hoạt động bằng pin. Phần này mô tả cách hoạt động của mạng khu vực cá nhân không dây tốc độ thấp (LR-WPAN). IEEE 802.15.4e là một phần mở rộng của tiêu chuẩn IEEE 802.15.4, được thiết kế để cải thiện và tối ưu hóa hiệu suất cho các ứng dụng công nghiệp, tự động hóa, và các hệ thống thời gian thực.

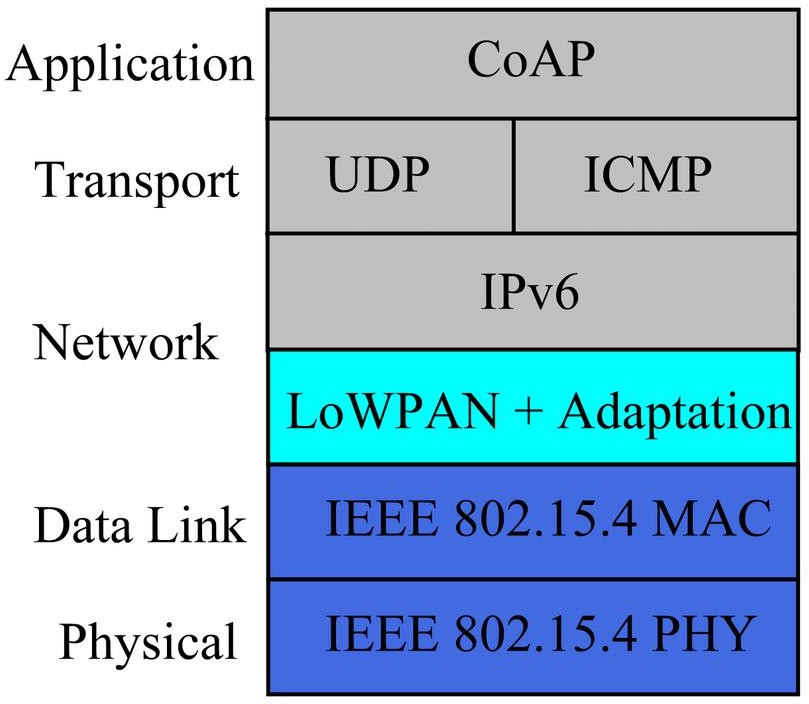
## Tổng Quan về 6LoWPAN

“6LoWPAN” là tên viết tắt của IPv6 protocol over low-power wireless PANs (có nghĩa : sử dụng giao thức IPv6 trong các mạng PAN không dây công suất thấp). 6LoWPAN được phát triển bởi hiệp hội đặc trách kỹ thuật Internet IETF ( Internet Engineering Task Foce), cho phép truyền dữ liệu qua các giao thức IPv6 và IPv4 trong các mạng không dây công suất thấp với các cấu trúc mạng điểm - điểm ( P2P: point to point ) và dạng lưới ( mesh). Tiêu chuẩn được đặt ra để quy định các đặc điểm của 6LoWPAN - cho phép sử dụng rộng rãi trong các ứng dụng IoT. Điểm khác của 6LoWPAN so với Zigbee, Bluetooth là: Zigbee hay bluetooth là các giao thức ứng dụng, còn 6LoWPAN là giao thức mạng, cho phép quy định cơ chế đóng gói bản tin và nén header. Đặc biệt, IPv6 là sự kế thừa của IPv4 và cung cấp khoảng 5 x 1028 địa chỉ cho tất cả mọi đối tượng trên thế giới, cho phép mỗi đối tượng là một địa chỉ IP xác định để kết nối với Internet. Được thiết kế để gửi các bản tin IPv6 qua mạng IEEE802.15.4 và các tiêu chuẩn IP mở rộng như: TCP, UDP, HTTP, COAP, MQTT và Websocket, là các tiêu chuẩn cung cấp nodes end-to-end, cho phép các router kết nối mạng tới các IP.

* Standard: RFC6282.
* Frequency: (adapted and used over a variety of other networking media including Bluetooth Smart (2.4GHz) or ZigBee or low-power RF (sub- 1GHz).
* Range: N/A.
* Data Rates: N/A

Mục đích: 6LoWPAN được thiết kế để cho phép các thiết bị trong mạng không dây cá nhân (Wireless Personal Area Networks - WPANs) sử dụng giao thức IPv6, mở rộng khả năng của các mạng cảm biến và thiết bị IoT.

Lớp 6LoWPAN nằm giữa lớp vận chuyển và liên kết dữ liệu như ở hình sau:



Hình 1: Lớp 6LoWPAN nằm giữa lớp vận chuyển và liên kết dữ liệu

6LoWPAN được thiết kế để hoạt động giữa lớp liên kết dữ liệu (p) và lớp mạng (IPv6). Đây là lý do mà nó được xem là một lớp 2.5. Kiến trúc này bao gồm các chức năng chính như nén, đóng gói và phân mảnh, hỗ trợ truyền dữ liệu IPv6 qua mạng không dây công suất thấp.

Tính năng chính của 6LoWPAN bao gồm các cơ chế để đóng gói và phân mảnh các gói dữ liệu IPv6 nhằm phù hợp với kích thước khung dữ liệu của IEEE 802.15.4, thường nhỏ hơn khung IPv6 tiêu chuẩn. Để giảm kích thước gói tin, 6LoWPAN sử dụng các kỹ thuật nén tiêu đề IPv6, cho phép truyền dữ liệu hiệu quả hơn trong các mạng công suất thấp. Ngoài ra, 6LoWPAN hỗ trợ các giao thức định tuyến như RPL (Routing Protocol for Low-Power and Lossy Networks), giúp định tuyến dữ liệu hiệu quả trong mạng lưới cảm biến.

6LoWPAN hiện nay vẫn còn đang được định chuẩn bởi IETF. Nhóm làm việc của IETF vẫn tiếp tục phát triển và duy trì các tiêu chuẩn và tài liệu liên quan đến 6LoWPAN để hỗ trợ các ứng dụng IoT và các mạng không dây công suất thấp.

Hiện nay, từ các nghiên cứu cũng như thực tiễn cho thấy rằng: mạng cảm biến không dây (Wireless Sensor Networks - WSN) đang ngày càng quan tâm đến việc tích hợp giao thức IP (Internet Protocol) trong các hệ thống. Nhưng tại sao lại là IP? Ở mạng Zigbee (cùng thuộc khung dữ liệu của IEEE 802.15.4) đến nay vẫn không dùng IP, WirelessHART có xài đâu.Đó là vì sử dụng IP cho phép các thiết bị WSN dễ dàng giao tiếp với các thiết bị khác trong mạng IP, bao gồm cả Internet với TCP/IP có tuổi thọ hơn 30 năm. Khả năng mở rộng vì mạng IP hỗ trợ mạng lớn và phức tạp hơn, cho phép mở rộng số lượng nút trong WSN mà không gặp phải các vấn đề về quản lý địa chỉ. Cuối cùng sự kết hợp giữa WSN và IP là nền tảng cho sự phát triển của Internet of Things (IoT), nơi mà các cảm biến và thiết bị nhỏ gọn có thể kết nối và trao đổi dữ liệu trên quy mô toàn cầu. Với không gian địa chỉ rộng lớn hơn, cũng được kỳ vọng sẽ đóng vai trò quan trọng trong quá trình này, đặc biệt là khi số lượng thiết bị kết nối tăng lên đáng kể trong mạng WSN.

Khả năng tương thích của lớp Network 6LoWPAN thì lại hoạt động tốt với tiêu chuẩn IP mở bao gồm TCP, UDP, HTTP, COAP, MATT và web-socket. Nó cung cấp các nút địa chỉ IP end-to-end. Không cần gateway, chỉ cần bộ định tuyến có thể kết nối mạng 6LoWPAN với IP. Nó hỗ trợ định tuyến mesh tự phục hồi, mạnh mẽ và có thể mở rộng. Cung cấp định tuyến one-to-many và many-to-one. Các bộ định tuyến mesh 6LoWPAN có thể định tuyến dữ liệu đến các nút khác trong mạng. Trong mạng 6LowPAN, các nút có thể ngủ trong một khoảng thời gian dài. Nó cũng cung cấp hỗ trợ triệt để cho lớp PHY, cho phép tự do dải tần và lớp vật lý, có thể được sử dụng trên nhiều nền tảng truyền thông như Ethernet, Wi-Fi, 802.15.4 hoặc ISM Sub-1GHz với khả năng tương tác ở cấp IP.

## 2.3 MAC Layer

Mạng cảm biến không dây (WSN) có nhiều hạn chế về tài nguyên - năng lượng, nên việc bảo toàn năng lượng là thực sự là một điều thiết yếu cũng như là một thách thức trong quá trình thực hiện nghiên cứu. Quá trình thu và phát của nút cảm biến trong WSN tiêu thụ một lượng lớn năng lượng. Có rất nhiều nghiên cứu đã được thực hiện cho các thiết bị tiêu thụ năng lượng thấp nhằm tạo ra những nốt cảm biến tối ưu nhất. Nhưng vì những giới hạn về phần cứng, hiệu quả năng lượng có thể đạt được hơn nữa. Trong đó những phương pháp nghiên cứu, thông qua việc thiết kế các giao thức truyền thông tiết kiệm năng lượng là một trong những cách tối ưu nhất. **Medium access control** (MAC) Lớp Điều khiển Truy cập Phương tiện là một kỹ thuật quan trọng đảm bảo hoạt động thành công của mạng.Một trong những chức năng chính của giao thức MAC là tránh các va chạm từ các nút gây nhiễu. Giao thức MAC IEEE 802.11 cổ điển cho mạng cục bộ không dây tiêu tốn nhiều năng lượng vì nghe lén không tải. Thiết kế giao thức MAC tiết kiệm năng lượng là một trong những cách kéo dài tuổi thọ của mạng.

Lớp Điều khiển Truy cập Phương tiện (MAC - Medium Access Control) chịu trách nhiệm quản lý việc liên lạc giữa các nút trong mạng. Như được minh họa trong *Hình 1*, lớp MAC là một phần của tầng liên kết dữ liệu (Data Link Layer) trong mô hình OSI (Open Systems Interconnection). Tầng liên kết dữ liệu được chia thành hai lớp chính: Lớp MAC (Medium Access Control): Lớp này có nhiệm vụ điều phối truy cập vào kênh truyền thông vật lý. Nó kiểm soát cách các nút trong mạng chia sẻ và truy cập kênh truyền thông chung, giúp ngăn ngừa xung đột và đảm bảo truyền thông hiệu quả giữa các nút. Lớp Điều khiển Liên kết Logic (LLC - Logical Link Control): Lớp LLC chịu trách nhiệm quản lý giao tiếp logic giữa các thiết bị, đảm bảo dữ liệu được truyền tải một cách tin cậy. LLC cũng thực hiện việc đánh dấu, xác định và điều khiển các gói dữ liệu trong khi chúng được truyền dưới dạng tín hiệu điện hoặc sóng vô tuyến. Sự phân chia này giúp đảm bảo rằng các nhiệm vụ liên quan đến truy cập kênh và quản lý liên kết được thực hiện hiệu quả, tối ưu hóa hiệu suất và độ tin cậy của hệ thống mạng.

## Vai Trò và Chức Năng của Lớp Điều Khiển Truy Cập Phương Tiện (MAC) trong Hệ Thống Mạng

Lớp Điều khiển Truy cập Phương tiện (MAC) đóng một vai trò quan trọng trong hệ thống mạng, đảm nhiệm việc quản lý truy cập và chia sẻ kênh truyền thông giữa các nút. Lớp MAC thực hiện các chức năng như điều phối truy cập kênh, giúp ngăn ngừa xung đột khi nhiều nút cố gắng truyền dữ liệu đồng thời. Trong môi trường có tranh chấp, lớp MAC đảm bảo sự công bằng và hiệu quả bằng cách quyết định nút nào được phép truyền dữ liệu tại những thời điểm cụ thể tuỳ thuộc vào loại giao thức. Ngoài ra, lớp MAC sử dụng các cơ chế như tránh xung đột (CSMA/CA) và phát hiện xung đột (CSMA/CD) để giảm thiểu khả năng xung đột, từ đó tối ưu hóa hiệu suất mạng. Đặc biệt, trong các mạng cảm biến không dây, lớp MAC đóng vai trò quan trọng trong việc **quản lý năng lượng**, bằng cách tắt hoặc đưa các nút vào chế độ chờ khi không có dữ liệu cần truyền hoặc nhận. Bên cạnh đó, lớp MAC cũng giúp giảm độ trễ truyền dữ liệu thông qua việc tối ưu hóa thời gian truy cập vào kênh, điều này rất quan trọng đối với các ứng dụng yêu cầu phản hồi thời gian thực. Cuối cùng, trong các mạng có tính di động cao, lớp MAC cần phải thích ứng với sự thay đổi vị trí của các nút, đảm bảo duy trì kết nối liên lạc ổn định giữa các nút di động và nút tĩnh. Nhưng hầu hết các giao thức MAC hiện nay được thiết kế cho mạng tĩnh hoặc có tính di động hạn chế. Những giao thức này có thể xử lý một phần vấn đề di động, nhưng chưa thể đáp ứng đầy đủ các yêu cầu của môi trường có di động cao và lưu lượng biến đổi. Như vậy, lớp MAC là một thành phần không thể thiếu trong hệ thống mạng, đảm bảo sự hiệu quả, ổn định, và tin cậy của các hoạt động mạng.

## Những Thách Thức Trong Thiết Kế Giao Thức MAC

Việc thiết kế một giao thức Điều khiển Truy cập Phương tiện (MAC) hiệu quả cho mạng cảm biến không dây (WSN) liên quan đến nhiều yếu tố quan trọng.

Đầu tiên và là quan trọng nhất, hiệu quả năng lượng là yếu tố then chốt trong thiết kế giao thức điều khiển truy cập phương tiện (MAC) cho mạng cảm biến không dây (WSNs) vì các nút cảm biến thường được cấp nguồn bằng pin, và việc thay thế hoặc sạc pin có thể rất khó khăn và tốn kém. Điều này đặc biệt quan trọng trong các ứng dụng mà các nút cảm biến được triển khai ở những vị trí xa xôi, khó tiếp cận, như trong lòng đất để theo dõi hoạt động địa chấn hoặc trong các khu vực nguy hiểm như rừng rậm hoặc môi trường độc hại. Ví dụ, trong một mạng cảm biến dưới lòng đất dùng để theo dõi hoạt động địa chấn, việc tiếp cận các nút cảm biến để thay thế hoặc sạc pin sẽ rất phức tạp, yêu cầu trang thiết bị đặc biệt và gây ra chi phí lớn. Trong các môi trường khắc nghiệt khác như rừng sâu hay khu vực nhiễm độc, việc bảo trì nguồn pin cho các nút cảm biến thậm chí còn khó khăn hơn. Giao thức MAC phải tối ưu hóa việc sử dụng năng lượng, không chỉ để giảm thiểu lãng phí mà còn để đảm bảo rằng mạng cảm biến có thể hoạt động trong thời gian dài mà không cần bảo trì. Điều này không chỉ giúp giảm chi phí mà còn đảm bảo rằng mạng có thể thực hiện nhiệm vụ giám sát và báo cáo liên tục, đặc biệt là trong các môi trường khó tiếp cận và nguy hiểm.

Thứ hai, Latency - độ trễ trong giao thức điều khiển truy cập phương tiện (MAC) là một thách thức quan trọng vì nó ảnh hưởng trực tiếp đến khả năng phản hồi kịp thời của hệ thống. Trong các ứng dụng mạng cảm biến không dây (WSN), khi các sự kiện quan trọng được phát hiện, việc truyền thông tin nhanh chóng đến nút đích (sink node) là điều cần thiết để đảm bảo hành động thích hợp được thực hiện ngay lập tức. Độ trễ trong mạng cảm biến thường phát sinh từ nhiều yếu tố như thời gian xử lý dữ liệu tại nút cảm biến, thời gian truyền gói tin qua môi trường không dây, và thời gian truy cập kênh truyền trong các giao thức MAC. Vậy nên giao thức MAC phải được thiết kế để giảm thiểu các yếu tố này, đảm bảo rằng các gói tin có thể tiếp cận nút đích một cách nhanh chóng. Để đạt được độ trễ thấp, giao thức MAC cần phải cân bằng giữa việc tiết kiệm năng lượng và tốc độ truyền tin. Việc tiết kiệm năng lượng bằng cách đưa các nút vào trạng thái nghỉ có thể gây ra độ trễ khi các nút cần phải được đánh thức để truyền dữ liệu. Do đó, việc quản lý tài nguyên một cách hiệu quả mà không ảnh hưởng đến độ trễ là một thách thức lớn. Xung đột trên kênh truyền là một yếu tố khác gây ra độ trễ. Khi nhiều nút cảm biến cố gắng truyền dữ liệu cùng lúc, các xung đột có thể xảy ra, yêu cầu các nút phải đợi hoặc truyền lại, dẫn đến tăng độ trễ. Giao thức MAC cần có cơ chế xử lý xung đột hiệu quả để giảm thiểu độ trễ này.

Với mạng cảm biến không dây (WSNs) thì tính mở rộng **(Scalability)** là một yếu tố quan trọng trong thiết kế giao thức điều khiển truy cập phương tiện (MAC). Tính mở rộng đề cập đến khả năng của mạng trong việc duy trì hiệu suất ổn định khi quy mô mạng tăng lên, tức là khi số lượng nút trong mạng tăng đáng kể. Mạng cảm biến không dây thường bao gồm một số lượng lớn các nút cảm biến, có thể lên đến hàng nghìn hoặc hàng triệu nút. Giao thức MAC cần phải xử lý hiệu quả số lượng nút thay đổi này mà không làm giảm hiệu suất của mạng. Điều này bao gồm việc đảm bảo rằng các gói dữ liệu được truyền tải một cách hiệu quả và rằng các nút có thể truy cập kênh truyền một cách công bằng và hợp lý. Khi số lượng nút trong mạng tăng lên, nguy cơ xảy ra xung đột truyền thông, nghẽn mạng, và tiêu thụ năng lượng không cân đối cũng tăng theo. Giao thức MAC cần phải thích ứng với những thay đổi này để duy trì thông lượng cao và độ trễ thấp, đồng thời tiết kiệm năng lượng.

## Các Nguồn Lãng Phí Năng Lượng Chính

Trong mạng cảm biến không dây (WSN), năng lượng là một tài nguyên cực kỳ quý giá, vì các nút cảm biến thường được cấp nguồn bằng pin và việc thay thế hoặc sạc lại pin thường gặp nhiều khó khăn. Có 4 nguồn lãng phí năng lượng chính trong WSN mà bất kỳ giao thức Điều khiển Truy cập Phương tiện (MAC) nào cũng cần phải giải quyết để tối ưu hóa hiệu suất năng lượng đó là:

· **Va chạm (Collision)**:

Va chạm xảy ra khi hai hoặc nhiều nút cảm biến cùng gửi dữ liệu vào cùng một thời điểm và trên cùng một kênh truyền thông. Kết quả là các gói tin bị trộn lẫn và trở nên không thể đọc được, dẫn đến phải loại bỏ các gói tin này. Để khắc phục, các nút cảm biến phải truyền lại dữ liệu, điều này tiêu tốn thêm năng lượng. Không chỉ vậy, va chạm còn làm tăng độ trễ, gây ảnh hưởng xấu đến hiệu suất tổng thể của mạng, đặc biệt là trong các ứng dụng yêu cầu thời gian thực.

· **Nghe thừa (Overhearing)**:

Nghe thừa xảy ra khi một nút cảm biến nhận và xử lý các gói tin không được gửi cho nó. Trong trường hợp này, năng lượng bị lãng phí vì nút cảm biến sử dụng năng lượng để xử lý thông tin không cần thiết, gây ra sự suy giảm trong hiệu suất năng lượng tổng thể của mạng. Hiện tượng này thường xuyên xảy ra trong các mạng cảm biến không dây, nơi mà các nút cảm biến hoạt động gần nhau và thường xuyên giao tiếp.

· **Quá tải gói điều khiển (Packet Overhead)**:

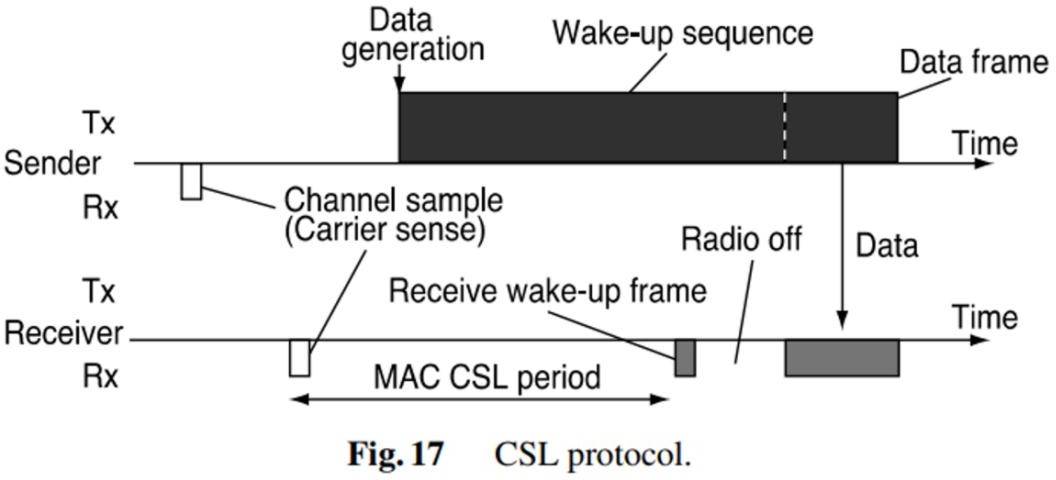
Trong bất kỳ hệ thống truyền thông nào, ngoài việc truyền tải dữ liệu thực, các gói tin điều khiển cũng cần được gửi để duy trì hoạt động của mạng. Tuy nhiên, việc truyền và nhận các gói tin điều khiển này tiêu tốn năng lượng, làm giảm số lượng gói tin dữ liệu thực sự có thể được truyền đi. Điều này dẫn đến giảm hiệu suất của mạng và gây ra sự lãng phí năng lượng đáng kể.

· **Lắng nghe khi không có dữ liệu (Idle Listening)**:

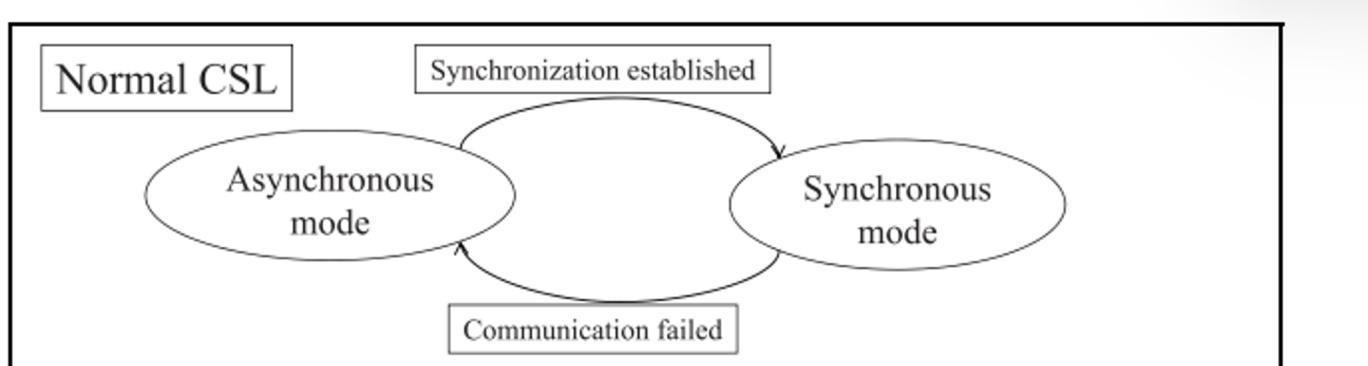
Lắng nghe khi không có dữ liệu xảy ra khi một nút cảm biến giữ trạng thái lắng nghe để chờ nhận dữ liệu, mặc dù không có gói tin nào được gửi đến nó. Hiện tượng này rất phổ biến trong các mạng cảm biến không dây, đặc biệt là khi các nút cảm biến hoạt động trong các khu vực mà dữ liệu chỉ được gửi đi một cách không thường xuyên. Việc duy trì trạng thái lắng nghe trong thời gian dài mà không có dữ liệu làm tăng đáng kể mức tiêu thụ năng lượng của nút cảm biến.

## Giao thức MAC: Đồng bộ và Bất đồng bộ

Giao thức MAC (Medium Access Control) có thể được phân loại thành hai nhóm chính: đồng bộ và bất đồng bộ, mỗi loại có những ưu điểm và thách thức riêng tuỳ vào mỗi mô hình thì có thể ứng dụng phù hợp. Các giao thức MAC đồng bộ yêu cầu tất cả các nút trong mạng phải tuân thủ một lịch trình chung về thời gian thức dậy và ngủ. Điều này có nghĩa là tất cả các nút trong mạng sẽ đồng thời thức dậy và sẵn sàng để trao đổi dữ liệu theo một lịch trình đã định sẵn. Việc đồng bộ lịch trình thức và ngủ của các nút dễ dàng đảm bảo rằng tất cả các nút có thể trao đổi dữ liệu một cách có tổ chức và đồng bộ, giảm thiểu khả năng xung đột và tăng cường hiệu suất truyền tải dữ liệu. Tuy nhiên, việc yêu cầu tất cả các nút phải đồng bộ về thời gian có thể gây ra vấn đề khi có sự thay đổi trong cấu hình mạng hoặc khi các nút mới gia nhập vào mạng. Thêm vào đó, việc đồng bộ hóa đồng hồ giữa các nút có thể phức tạp và tiêu tốn tài nguyên, nhất là trong các mạng lớn hoặc có nhiều nút di động. Một ví dụ nổi bật trong nhóm giao thức đồng bộ là giao thức CSL MAC, được thiết kế để duy trì đồng bộ hóa liên tục giữa các nút trong mạng cảm biến không dây. Giao thức CSL cải thiện khả năng đồng bộ hóa thời gian và giảm độ trễ trong truyền dữ liệu, nhưng cũng đòi hỏi quản lý tài nguyên phức tạp hơn.

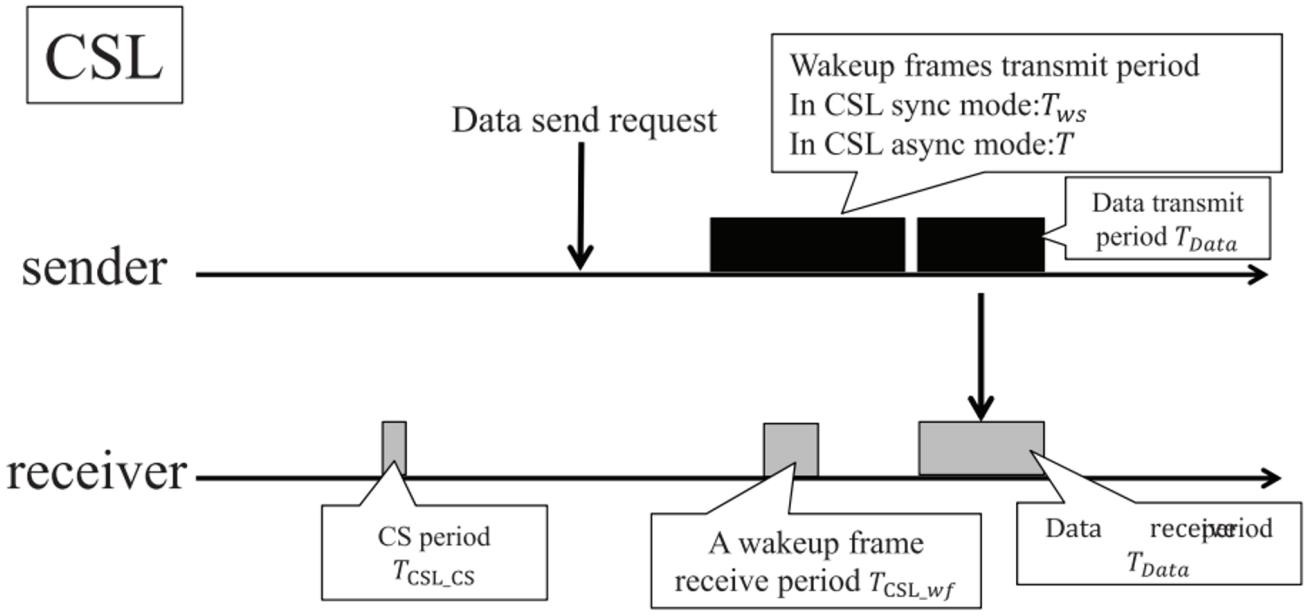


Việc triển khai này bao gồm giao thức chu kỳ nhiệm vụ vô tuyến 802.15.4 của giao thức CSL tiêu chuẩn cho mạng cảm biến không dây trên Contiki- OS. Hình sau đây cho thấy trạng thái không đồng bộ và không đồng bộ của giao thức.



Nghiên cứu cách thức hoạt động của các giao thức MAC tuần hoàn nhiệm vụ và sự khác biệt giữa các giao thức này Giao thức MAC không đồng bộ. Một giao thức MAC không đồng bộ được tiêu chuẩn hóa được CSL (Continuous Synchronized Listening) là một giao thức mạng không yêu cầu đồng bộ thời gian trên toàn mạng. Để cải thiện độ tin cậy, hiệu suất năng lượng và thông lượng, CSL hỗ trợ chuyển kênh, truyền dữ liệu đồng bộ hóa và chuyển tiếp dữ liệu nhanh chóng. Ngoài ra, CSL cho phép các thiết bị được cấp nguồn từ lưới điện tắt chế độ đợi vòng lặp để giảm độ trễ, nhưng tính năng đó vẫn chưa được triển khai. Giao thức CSL cũng là một tính năng MAC của IEEE 802.15.4e nhắm mục tiêu vào các nút năng lượng thấp. Các nút trong kịch bản WSN đang được điều tra chạy IEEE 802.15.4e giao thức CSL các giao thức trên hệ điều hành thời gian thực Contiki NG. Trong giao thức CSL, nút phát (TX) gửi khung đánh thức trước khi một gói dữ liệu được gửi đi. Thông tin về thời điểm mà gói dữ liệu sẽ được gửi sẽ được bao gồm trong các lần đánh thức này khung. Thay vì ở chế độ nhận tiêu tốn năng lượng, giao thức CSL thức dậy định kỳ để quét kênh để tìm cái gọi là **“wake-up frame”**. Các khung đánh thức được truyền bởi giao thức CSL trong một chuỗi liên tục trước khi truyền một **“payload frame”** thực tế frame chứa thời gian khi quá trình truyền khung tải trọng đó bắt đầu. Nếu nút nhận (RX) thức khi có khung đánh thức đang được truyền đi và nó có thể lấy lại thành công thời gian thông tin khi gói dữ liệu sẽ được gửi đi, nó sẽ đi đến ngủ cho đến khi gói dữ liệu được gửi đi. Tuy nhiên, nếu người nhận không thức và không nhận được khung đánh thức, người nhận đi ngủ. Người nhận thức dậy định kỳ để kiểm tra các khung đánh thức. Nếu người nhận thành công trong nhận gói dữ liệu từ máy phát (sau khi nhận được khung đánh thức), nó có thể gửi một khung xác nhận “acknowledgement frame” được phản hồi cũng chứa tính chu kỳ mà người nhận thức dậy. Thông tin này có thể được sử dụng tại máy phát để ước tính lần sau người nhận sẽ thức. Tại thời điểm này, sau đó máy phát có thể quyết định gửi các khung đánh thức tại thời gian dự kiến để người nhận thức. Việc thiếu khung xác nhận khiến giao thức CSL để truyền lại sau một khoảng thời gian chờ ngẫu nhiên, trừ khi có tối đa đã đạt được số lần truyền lại. Tiếp theo, chúng tôi mô tả thiết lập mô phỏng được sử dụng để có được kết quả. Ngoài ra, giao thức CSL bao gồm hai tối ưu hóa quan trọng. Sự tối ưu hóa đầu tiên liên quan đến độ dài của các lần đánh thức. Độ dài này mặc định kéo dài suốt khoảng thời gian thức dậy, nhưng được rút ngắn theo tiêu chuẩn vô tuyến IEEE 802.15.4 gọi là “truyền đồng bộ”. Là một phần của việc tối ưu hóa này, giao thức CSL hỗ trợ giai đoạn hiện tại của nó, tức là thời gian cho đến giai đoạn tiếp theo đánh thức, trên các khung xác nhận, cũng như, tùy chọn, trên khung tải trọng. Dữ liệu được hỗ trợ này cho phép giao thức CSL tìm hiểu và theo dõi thời gian thức dậy của các nút lân cận. Sau đó, khi truyền một khung unicast đến một máy thu có đã biết thời gian đánh thức, giao thức CSL chỉ truyền các khung đánh thức cho sự không chắc chắn dựa trên sự trôi dạt của thời gian về mong đợi của người nhận thời gian thức dậy và bắt đầu trình tự đánh thức này ngay trước khi người nhận có thể thức dậy sớm nhất.

|  |
| --- |
| **#define NBR\_TABLE\_CONF\_WITH\_LOCKING 1**: Định nghĩa một hằng số để chỉ ra rằng bảng lưu trữ địa chỉ hàng xóm (neighbor table) sẽ được sử dụng với cơ chế khóa để đảm bảo sự an toàn trong việc truy cập từ nhiều tiến trình cùng một lúc |

Tối ưu hóa thứ hai liên quan đến thông lượng của giao thức CSL. Nếu giao thức CSL có nhiều khung tải trọng đi cho cùng một đích, giao thức CSL hỗ trợ truyền những khung hình này theo từng đợt. Nghĩa là, không gửi một chuỗi đánh thức khác, giao thức CSL sẽ truyền phát đơn hướng hoặc phát sóng đang chờ xử lý đó các khung ngay sau lần xác nhận cuối cùng được nhận hoặc gửi khung phát sóng tương ứng. Người nhận được thông báo liệu nhiều khung tải trọng hơn sẽ đi qua cờ Đang chờ khung của tiêu đề khung IEEE 802.15.4.

## Contiki MAC

Ngược lại, các giao thức MAC bất đồng bộ cho phép các nút trong mạng có lịch trình thức dậy và ngủ độc lập, tạo điều kiện linh hoạt hơn cho hoạt động của mạng. Các nút không cần phải đồng bộ hóa với nhau về thời gian thức dậy, và có thể hoạt động một cách độc lập, điều này đặc biệt hữu ích trong các mạng yêu cầu tiết kiệm năng lượng cao hoặc khi các nút di động thường xuyên thay đổi vị trí. Nhưng sự linh hoạt này cũng đi kèm với những thách thức riêng, đặc biệt là trong việc đồng bộ hóa truyền dữ liệu. Khi các nút không có lịch trình chung, việc đảm bảo rằng các gói dữ liệu được truyền đi và nhận đúng cách trở nên phức tạp hơn. Điều này có thể dẫn đến việc tăng cường độ trễ và khả năng mất dữ liệu, đồng thời yêu cầu các phương pháp bổ sung để giải quyết các vấn đề liên quan đến đồng bộ hóa và khả năng tương thích giữa các nút.

ContikiMAC là một giao thức mạnh mẽ cho việc truyền thông không dây tiết kiệm năng lượng, đặc biệt trong các bối cảnh lấy mẫu tiền đề. Giao thức này sử dụng kỹ thuật lấy mẫu tiền đề để giảm thiểu việc lắng nghe khi rảnh, một vấn đề phổ biến trong các mạng cảm biến không dây (WSN) vì radio thường tiêu tốn nhiều năng lượng hơn khi hoạt động so với khi ngủ. Các nút trong mạng sẽ định kỳ thức dậy để kiểm tra các gói dữ liệu đến, qua đó giảm thiểu thời gian mà radio hoạt động không cần thiết. ContikiMAC cung cấp ba chế độ truyền tải chính: **unicast**, **broadcast** và **burst**. Trong chế độ unicast, nút gửi sẽ lặp đi lặp lại việc truyền gói dữ liệu chứa payload và địa chỉ đích cho đến khi nhận được xác nhận từ nút nhận. Nút nhận sẽ định kỳ thức dậy để kiểm tra tín hiệu từ các nút lân cận và giữ radio hoạt động để nhận gói dữ liệu tiếp theo khi phát hiện truyền tải. Sau khi nhận gói dữ liệu thành công, nút nhận sẽ phản hồi bằng gói xác nhận (ACK). Trong chế độ broadcast, nút gửi truyền liên tục gói dữ liệu trong toàn bộ thời gian tiền đề để đảm bảo rằng tất cả các nút lân cận đều nhận được gói mà không yêu cầu phản hồi xác nhận từ các nút nhận. Chế độ burst được thiết kế để xử lý các giai đoạn truyền tải cao, khi nút gửi dự đoán có nhiều gói dữ liệu cần truyền liên tiếp. Nút gửi sẽ đánh dấu tiêu đề của mỗi gói dữ liệu (ngoại trừ gói cuối cùng) để thông báo cho nút nhận rằng có gói tiếp theo. Nút nhận giữ radio hoạt động để nhận các gói tiếp theo và chuyển sang chế độ Carrier Sense Multiple Access (CSMA) cho đến khi nhận được gói dữ liệu không được đánh dấu với cờ burst.

Các nghiên cứu đã chỉ ra rằng ContikiMAC đạt hiệu suất tốt hơn về độ trễ và tiêu tốn năng lượng so với các giao thức khác như X-MAC, nhờ vào các tính năng cải tiến của nó. Bằng cách kết hợp các đặc điểm của các giao thức preamble-sampling khác như WiseMAC và BoX-MAX, ContikiMAC không chỉ tối ưu hóa việc sử dụng năng lượng mà còn giảm thiểu độ trễ trong mạng. Do đó, ContikiMAC là một lựa chọn hàng đầu trong các giao thức preamble-sampling, và chúng tôi có thể phát triển cơ chế của mình dựa trên nền tảng này vì nó cung cấp một sự cân bằng tốt giữa hiệu suất và tiết kiệm năng lượng.

Một ví dụ về giao thức bất đồng bộ là ContikiMAC, giao thức MAC mặc định trong hệ điều hành Contiki. ContikiMAC được thiết kế để cải thiện hiệu suất trong các mạng cảm biến không dây bằng cách sử dụng kỹ thuật "**preamble-sampling**" để giảm thiểu thời gian các nút phải hoạt động. Giao thức này hỗ trợ các chế độ truyền dữ liệu khác nhau, bao gồm unicast, broadcast và burst, nhằm tối ưu hóa hiệu suất truyền tải dữ liệu và tiết kiệm năng lượng. ContikiMAC cũng tích hợp các phương pháp xử lý truyền tải bursty, giúp nâng cao hiệu suất trong các tình huống có lưu lượng cao. Tuy nhiên, việc sử dụng các kỹ thuật này trong môi trường mạng có các nút di động hoặc trong các tình huống yêu cầu đồng bộ hóa chính xác vẫn cần được cân nhắc để tối ưu hóa hiệu quả và giảm thiểu các vấn đề liên quan đến đồng bộ hóa và độ trễ.

Về mặt tổng thể, ContikiMAC nhắm đến việc cung cấp truyền thông đáng tin cậy trong khi giảm thiểu tiêu thụ năng lượng, điều này rất quan trọng trong môi trường mạng cảm biến không dây có tài nguyên hạn chế.

Mặc dù giao thức ContikiMAC được thiết kế tốt cho các mạng tĩnh, nhưng nó không hoạt động hiệu quả trong các môi trường có sự kết hợp giữa các nút tĩnh và di động. Thực tế, vì các nút di động không tham gia vào việc xây dựng cây định tuyến, chúng không biết địa chỉ của nút kế tiếp và điều này thực sự ngăn cản chúng sử dụng các truyền tải unicast. Với các chức năng truyền tải unicast và broadcast mặc định của ContikiMAC, một nút di động sẽ phải truyền các gói của nó bằng cách sử dụng broadcast. Tuy nhiên, việc broadcast là một phương pháp tốn kém, khiến các nút di động không thể tiếp cận kênh truyền thông với các nút tĩnh một cách hiệu quả (dẫn đến hiệu suất tiêu tốn năng lượng thấp và độ trễ cao). Trong trường hợp truyền tải burst, một nút di động có thể truyền tất cả các gói dữ liệu n bằng broadcast, hoặc nó truyền gói dữ liệu đầu tiên bằng broadcast để tìm một nút cha tạm thời và sau đó chuyển sang chế độ unicast để xử lý các gói trong bộ đệm của nó. Kết quả là, các chế độ truyền tải mặc định của ContikiMAC gây ra một số hạn chế trong mạng khi liên quan đến các nút di động.

Cơ chế đánh thức ContikiMAC và CSL hoạt động như một protothread được lên lịch theo thời gian thực định kỳ hẹn giờ. Protothread này thực hiện việc đánh thức định kỳ và thực hiện tối ưu hóa giấc ngủ nhanh. Việc truyền tải được điều khiển bởi quy trình Contiki NG thông thường. Nếu việc đánh thức được lên kế hoạch diễn ra khi radio đang hoạt động bận trong quá trình truyền, bộ hẹn giờ đánh thức sẽ lên lịch đánh thức mới sau khoảng thời gian đánh thức khác mà không thực hiện việc đánh thức. Giảm độ dài của việc truyền tải do đó làm giảm tắc nghẽn vô tuyến. Nếu một người hàng xóm có pha được biết đã khởi động lại hoặc nếu đồng hồ của nó đã trôi đủ xa so với trước đó giai đoạn đánh thức, việc truyền tới hàng xóm sẽ thất bại. Để bảo vệ khỏi điều này, duy trì số lượng truyền thất bại cho mỗi người hàng xóm đã biết. Sau một số lần truyền không thành công đã cố định, người hàng xóm bị xóa khỏi danh sách những người hàng xóm đã biết. Tương tự như vậy, nếu không nhận được xác nhận lớp liên kết trong một khoảng thời gian cố định, bất kể về số lần truyền, người hàng xóm bị xóa ra khỏi danh sách.

Tóm lại, lựa chọn giữa giao thức MAC đồng bộ và bất đồng bộ phụ thuộc vào yêu cầu cụ thể của ứng dụng và môi trường mạng. Giao thức đồng bộ có thể phù hợp hơn với các mạng có cấu trúc cố định và yêu cầu cao về hiệu suất truyền tải dữ liệu, trong khi giao thức bất đồng bộ thường được ưa chuộng trong các môi trường thay đổi liên tục và yêu cầu tiết kiệm năng lượng cao.

## Chu kỳ Hoạt Động

Trong các giao thức lớp MAC (Medium Access Control), tần số lấy mẫu kênh (Channel Sampling Rate) và tần số thức dậy (Wake-up Interval) là hai khái niệm có vai trò quan trọng, với sự phân biệt rõ ràng giữa các giao thức đồng bộ đại diện cho CSL và bất đồng bộ đại diện cho Contiki MAC. Hai yếu tố này không chỉ ảnh hưởng trực tiếp đến hiệu suất truyền thông mà còn quyết định mức độ tiêu thụ năng lượng của các nút trong mạng cảm biến không dây và IoT.

Trong các giao thức MAC đồng bộ như CSL,Tần Số Lấy Mẫu Kênh (Channel Sampling Rate) đóng vai trò quan trọng trong việc xác định khoảng thời gian mà các nút trong mạng sẽ kiểm tra kênh truyền thông theo một lịch trình đã được đồng bộ hóa. Nút sẽ "nghe" kênh vào các thời điểm cụ thể, thường được phối hợp với các nút khác trong mạng để đảm bảo tất cả các nút đều kiểm tra kênh vào cùng một thời điểm hoặc theo một chuỗi thời gian đã định. Vì các nút đều tuân theo một lịch trình đã được đồng bộ hóa, tần số lấy mẫu kênh giúp giảm thiểu xung đột và tăng hiệu quả trong việc truyền nhận dữ liệu, đồng thời giảm tiêu thụ năng lượng do chỉ cần "nghe" kênh tại các thời điểm đã biết trước. Ví dụ như một mạng cảm biến không dây sử dụng giao thức CSL (Coordinated Sampled Listening) để đồng bộ hoạt động của các nút. Trong trường hợp này, mỗi nút trong mạng sẽ kiểm tra kênh truyền thông 10 lần mỗi giây, tức là mỗi 100ms một lần. Điều này có nghĩa là các nút sẽ "nghe" kênh vào những khoảng thời gian đã được định trước để kiểm tra xem có tín hiệu nào không. Vì đây là một giao thức đồng bộ, tất cả các nút sẽ thực hiện việc này đồng thời, giúp giảm thiểu va chạm và tối ưu hóa việc truyền nhận dữ liệu.

Trong các giao thức MAC bất đồng bộ như ContikiMAC,Tần Số Thức Dậy (Wake-up Interval) là yếu tố chính để quyết định khoảng thời gian mà một nút trong mạng sẽ thức dậy để kiểm tra kênh truyền thông. Không có sự đồng bộ chặt chẽ giữa các nút, do đó mỗi nút tự thức dậy theo lịch trình riêng của mình để kiểm tra kênh, nhận hoặc truyền dữ liệu. Điều này cho phép mạng hoạt động linh hoạt hơn mà không cần phải đồng bộ hóa giữa các nút. Các nút không cần phải đồng bộ với nhau, tần số thức dậy trong giao thức MAC bất đồng bộ giúp mạng có khả năng mở rộng linh hoạt, nhưng cũng có thể dẫn đến độ trễ cao hơn và mức tiêu thụ năng lượng không đều. Ví dụ với tần số thức dậy là 4 Hz, mỗi nút ContikiMAC sẽ thức dậy từ trạng thái ngủ mỗi 250ms (1 giây / 4 = 0.25 giây) để kiểm tra kênh, nhận hoặc truyền dữ liệu, sau đó quay trở lại trạng thái ngủ nếu không có hoạt động nào cần thiết.

## Khả Năng Nhận Gói Tin

**Khả Năng Nhận Gói Tin (Packet Reception Rate - PRR)** là một chỉ số đo lường trong các mạng truyền thông, đặc biệt là mạng cảm biến không dây và mạng IoT, dùng để đánh giá tỷ lệ phần trăm của các gói tin dữ liệu được nhận thành công bởi một nút trong mạng so với tổng số gói tin đã được gửi đến nút đó.

Công Thức Tính Khả Năng Nhận Gói Tin (Packet Reception Rate - PRR) được tính bằng tỷ lệ giữa số gói tin được nhận thành công và tổng số gói tin đã được gửi đi. Công thức cụ thể như sau:

=

PRR là yếu tố quan trọng trong việc thiết kế và tối ưu hóa các giao thức MAC (Medium Access Control). Ví dụ, các giao thức MAC với khả năng điều chỉnh tần số thức dậy hoặc tần số lấy mẫu kênh có thể cải thiện PRR bằng cách giảm thiểu va chạm gói tin và nâng cao khả năng nhận dữ liệu. Công thức tính PRR giúp đánh giá khả năng nhận gói tin thành công trong mạng truyền thông, cung cấp thông tin quan trọng về hiệu suất, độ tin cậy, và chất lượng của mạng. Bằng cách phân tích và tối ưu hóa PRR, các nhà thiết kế và quản trị mạng có thể cải thiện hiệu suất mạng, tiết kiệm năng lượng, và đảm bảo truyền thông dữ liệu hiệu quả.

## Khả Năng Chịu Tải

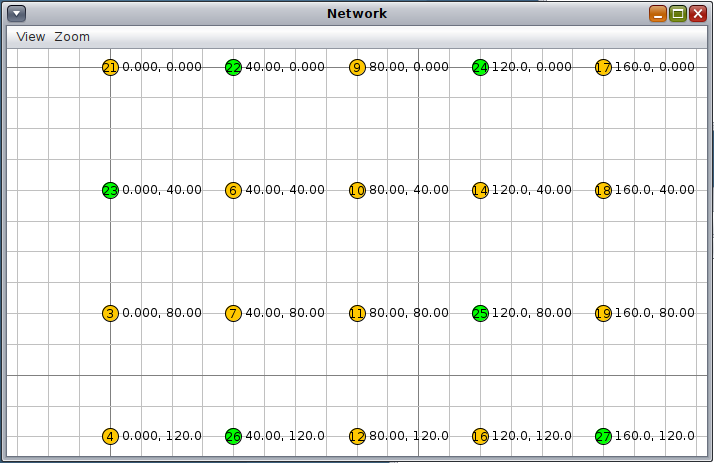
Khả Năng Chịu Tải, hay còn gọi là "load tolerance," là khả năng của giao thức lớp MAC trong việc duy trì hiệu quả truyền thông khi số lượng nút, lượng dữ liệu và tần suất truyền thông tăng lên trong mạng cảm biến không dây (WSN). Nói cách khác, nó phản ánh khả năng của giao thức trong việc xử lý một lượng lớn dữ liệu và quản lý nhiều nút hoạt động đồng thời mà không làm giảm hiệu suất tổng thể của mạng.

Khi đánh giá hiệu suất của các giao thức MAC, Khả Năng Chịu Tải là một yếu tố quan trọng để đảm bảo rằng mạng có thể hoạt động ổn định và hiệu quả ngay cả khi đối mặt với sự gia tăng về số lượng nút hoặc lượng dữ liệu cần truyền tải. Điều này bao gồm việc quản lý độ trễ, tỷ lệ nhận gói, và mức tiêu thụ năng lượng trong những tình huống tải cao.

CSL là một giao thức MAC được thiết kế với cơ chế đồng bộ hóa giữa các nút cảm biến, giúp giảm thiểu tình trạng trùng lặp và va chạm gói tin, vốn là vấn đề phổ biến trong mạng cảm biến không dây khi nhiều nút cố gắng truyền dữ liệu cùng lúc trên cùng một kênh truyền thông. Nhờ vào tính năng này, CSL trở nên hiệu quả hơn trong môi trường tải cao, nơi mà số lượng nút và lưu lượng dữ liệu đều lớn. Giao thức CSL quản lý và xử lý lượng dữ liệu lớn mà không làm tăng đáng kể độ trễ hoặc tỷ lệ va chạm, nhờ vào việc các nút cảm biến được thiết lập để thức dậy và lắng nghe trong các khoảng thời gian đồng bộ hóa định trước. Điều này tối ưu hóa quá trình truyền nhận dữ liệu và giảm thiểu xung đột trên kênh truyền. Sự đồng bộ hóa giữa các nút giúp giảm va chạm giữa các gói dữ liệu khi nhiều nút cùng truy cập vào kênh, điều này rất quan trọng trong các mạng có số lượng nút cảm biến lớn và lưu lượng dữ liệu cao. Tuy nhiên, cơ chế đồng bộ hóa này cũng đi kèm với một chi phí cao. Khi tải mạng tăng lên, việc duy trì sự đồng bộ hóa trở nên phức tạp và đòi hỏi nhiều tài nguyên hơn, bao gồm năng lượng và băng thông.

Còn ContikiMAC là một giao thức MAC không đồng bộ, nơi các nút cảm biến thức dậy và lắng nghe kênh truyền theo chu kỳ ngẫu nhiên, không theo lịch trình cố định. Điều này mang lại sự linh hoạt trong việc quản lý truyền thông, đặc biệt trong các mạng cảm biến có sự không đồng nhất về thời gian hoạt động giữa các nút. Ưu điểm của ContikiMAC là khả năng dễ dàng thích ứng với các thay đổi trong tải mạng mà không cần đồng bộ hóa, giúp giảm độ phức tạp và tối ưu hóa năng lượng tiêu thụ. Tuy nhiên, trong môi trường tải cao, việc lắng nghe ngẫu nhiên có thể dẫn đến hiện tượng nghe thừa (overhearing), làm lãng phí năng lượng và giảm hiệu suất mạng.

Có thể thấy trong quá trình thực hiện mô phỏng thì số nút mạng có thể sắp xếp tối đa cho giao thức Contiki MAC là 6 nút TX trong 20 nút mạng. Con số này nhỏ hơn nhiều so với có thể sắp xếp tối đa 9 nút mạng của giao thức CSL, trong khi vẫn cho chỉ số PRR trung bình > 95%.



# THỰC HIỆN TRÊN CONTIKI NG

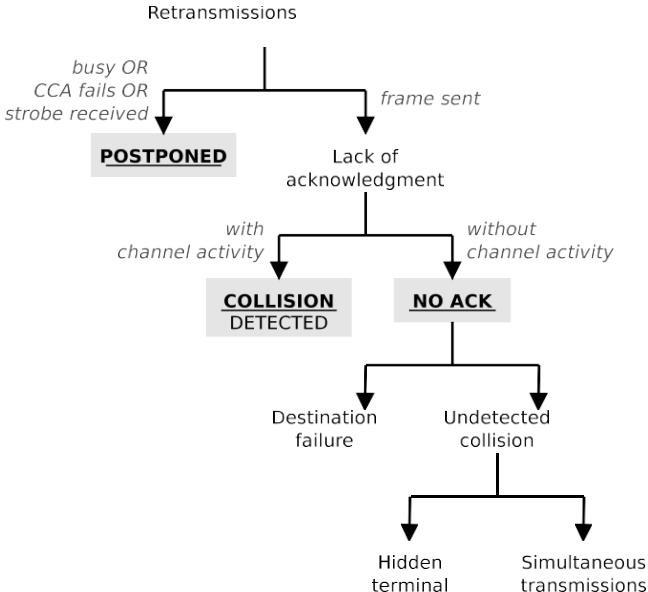
Thực hiện mô phỏng bằng Contiki NG COOJA Simulator. COOJA là một trình mô phỏng sự kiện rời rạc dựa trên Java. Nó cho phép thực hiện mô phỏng trên mạng và các cấp độ phần sụn. Ưu điểm chính là các mô phỏng có thể được chạy với hầu hết mã giống như được triển khai trên nền tảng thực. Một số triển khai giao thức như vì ContikiMAC, CSL và RPL có sẵn trong mã nguồn Contiki. Các thử nghiệm dựa trên nền tảng Zolertia Z1 được hỗ trợ tốt bởi COOJA. Mô hình lan truyền được sử dụng trong mô phỏng là Unit Disk Graph Medium của COOJA (UDGM) với mức lỗ liên tục. Mô hình này sử dụng hai đĩa đồng tâm có bán kính khác nhau. Bán kính thứ nhất là phạm vi truyền trong khi bán kính thứ hai, lớn hơn là phạm vi giao thoa. Bất kỳ nút nào trong phạm vi truyền tải đều nhận được các gói được truyền với xác suất 100% (giá trị mặc định). Bất kỳ nút nào nằm trong phạm vi nhiễu nhưng nằm ngoài phạm vi trong phạm vi truyền không thể nhận bất kỳ gói nào nhưng nhìn thấy đường truyền của chính nó ảnh hưởng. Cuối cùng, các nút nằm ngoài phạm vi nhiễu không thể nhận gói tin và không bị ảnh hưởng bởi các gói được truyền đi.

Việc triển khai ContikiMAC và CSL trong Contiki NG COOJA sử dụng Bộ hẹn giờ thời gian thực Contiki (rtimer) để lên lịch định kỳ đánh thức để đảm bảo hoạt động ổn định ngay cả khi nhiều tiến trình cơ bản đang chạy. Bộ hẹn giờ thời gian thực ưu tiên bất kỳ quy trình Contiki NG nào tại thời điểm chính xác mà chúng lên kế hoạch.

Dựa trên thông tin được cung cấp: Khoảng thời gian đánh thức và chu kỳ hoạt động: Cả ContikiMAC và CSL đều sử dụng khoảng thời gian đánh thức là 125 mili giây, đây là giá trị mặc định trong Contiki. Cơ chế tránh va chạm: Số lượng Đánh giá kênh thông suốt (CCA) được cơ chế tránh va chạm của ContikiMAC sử dụng đã được thay đổi từ 6 thành 2. Việc điều chỉnh này đảm bảo thỏa mãn ràng buộc Ti < Tc. Sử dụng nhiều hơn hai CCA sẽ làm tăng độ trễ mà không có bất kỳ lợi ích đáng chú ý nào. Việc điều chỉnh này cho phép chỉ tập trung vào các đường truyền tiêu chuẩn và bỏ qua các trường hợp đích đã hoạt động sau khi nhận được gói trước đó trong cùng một luồng dữ liệu. Truyền lại: ContikiMAC không truyền được khung, lớp trên sẽ lên lịch truyền lại. Lỗi truyền xảy ra do thiếu dữ liệu-ACK (đối với cả CSL và ContikiMAC). Số lần truyền lại được giới hạn ở 2. Những sửa đổi và cấu hình này nhằm mục đích tối ưu hóa hiệu suất và hiệu quả của cả giao thức ContikiMAC và CSL về mức tiêu thụ năng lượng, độ trễ và độ tin cậy trong môi trường mạng được chỉ định.

|  |  |  |
| --- | --- | --- |
|  | ContikiMAC | CSL |
| Wake-up interval | 8Hz | 8Hz |
| Duty-cycle | dynamic | dynamic |
| Streaming / burst mode | disabled | disabled |
| Retransmissions | 2 | 2 |

Phần này trình bày các kết quả thu được với cấu trúc liên kết sao và định tuyến tĩnh. Kết quả đã được tính trung bình trên 9 lần chạy.



# CẤU HÌNH THÍ NGHIỆM

Trong phần này, chúng tôi mô tả chi tiết về các cấu hình mạng được sử dụng trong các kịch bản khác nhau để so sánh các giao thức.

**Mạng hình chữ nhật (line) và hình lưới (gird) với định tuyến RPL**

Đối với kịch bản này, chúng tôi sử dụng các cấu hình mạng hình sao nhỏ với các đặc điểm sau:

**Kích thước có thể điều chỉnh**: Số lượng nút có thể được thay đổi để kiểm tra các tình huống khác nhau.

**Ứng dụng kiểu thu thập**: Mỗi nút trong mạng đang chạy một ứng dụng thu thập dữ liệu, trong đó các nút phát được chọn ngẫu nhiên gửi gói tin đến nút đích.

**Định tuyến động bằng RPL**: Các tuyến đường tới nút đích (sink) được tính toán động bằng giao thức RPL, giúp phù hợp hơn với các ứng dụng thực tế nơi cấu trúc mạng có thể thay đổi theo thời gian.

Trong các cấu hình này, chúng tôi có thể mô phỏng các tình huống thực tế hơn, nơi các nút di chuyển hoặc thay đổi trạng thái, và lưu lượng được tạo ra bởi các ứng dụng thực sự đang chạy trên mạng. Môi trường phức tạp và thực tế: Cung cấp cái nhìn sâu hơn về hiệu suất của các giao thức trong các tình huống thực tế hơn, giúp xác định những điểm mạnh và yếu của mỗi giao thức khi áp dụng vào các ứng dụng cụ thể.

## Mô hình mạng thực tế — Định tuyến RPL LITE

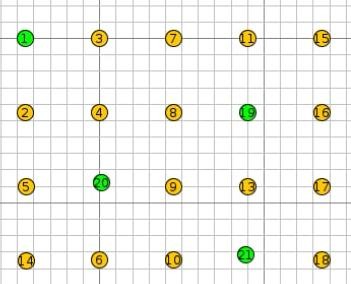
Chúng tôi thực hiện mô phỏng trong một thiết lập thực tế hơn. Trong các kịch bản này, các nút đang chạy một ứng dụng thu thập, trong đó mỗi nút phát được chọn ngẫu nhiên gửi các gói tin đến nút đích với khoảng cách 15 giây. Các tuyến đường được tính toán động bằng giao thức RPL. Cấu hình RPL của Contiki NG được sử dụng. Mỗi thí nghiệm kéo dài 2 giờ mô phỏng. Năm mươi lần chạy được thực hiện. Các cấu trúc mạng được tạo bằng hình một lưới vuông(gird) và có dạng hình chữ nhật( line). Các nút được bố trí trên một lưới vuông, tọa độ của chúng được cách nhau một khoảng cách là 50 và cách đều tất cả các nút lân cận. Mỗi cấu trúc mạng chứa 20 nút, 4 trong số đó là các nguồn và một nút còn lại là nút đích. Hình 7 cho thấy một ví dụ về một trong các cấu trúc mạng được sử dụng.

## Thiết lập thí nghiệm

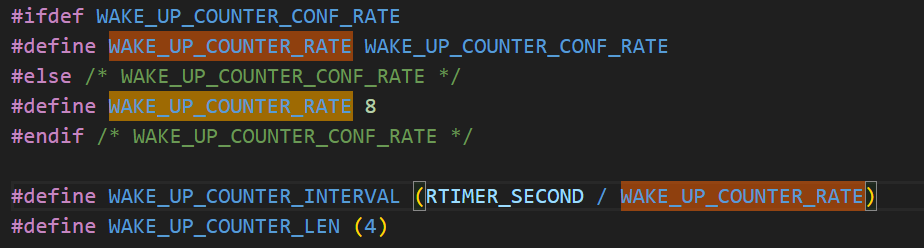
* Cấu trúc mạng: Mạng được bố trí theo cấu trúc lưới (grid topology) với tổng cộng 20 nút, trong đó có 4 nút truyền (transmitter) và 16 nút nhận (receiver).
* Tần số đánh thức: Các nút hoạt động với tần số đánh thức (wake-up interval) là 8Hz, nghĩa là các nút thức dậy để kiểm tra kênh và truyền dữ liệu mỗi 125ms (1 giây / 8 = 0.125 giây).
* Tần số lấy mẫu kênh Khi các nút có tần số lấy mẫu kênh là 8Hz, nghĩa là chúng sẽ kiểm tra kênh truyền thông mỗi 125ms (1 giây / 8 = 0.125 giây).
* Mô phỏng: Thí nghiệm được thực hiện để khảo sát chu kỳ hoạt động trung bình của radio (RDC) và tỷ lệ nhận gói (PRR).



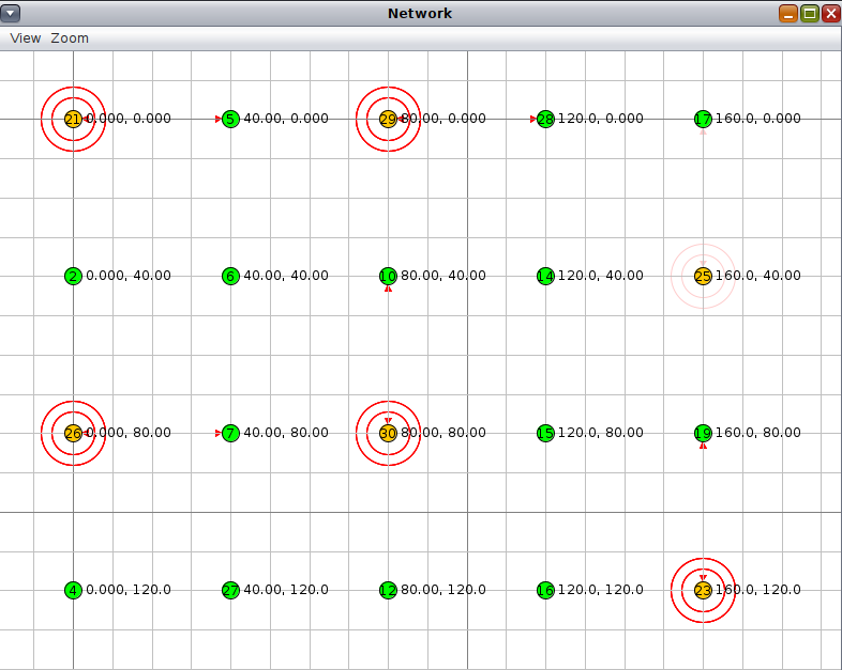
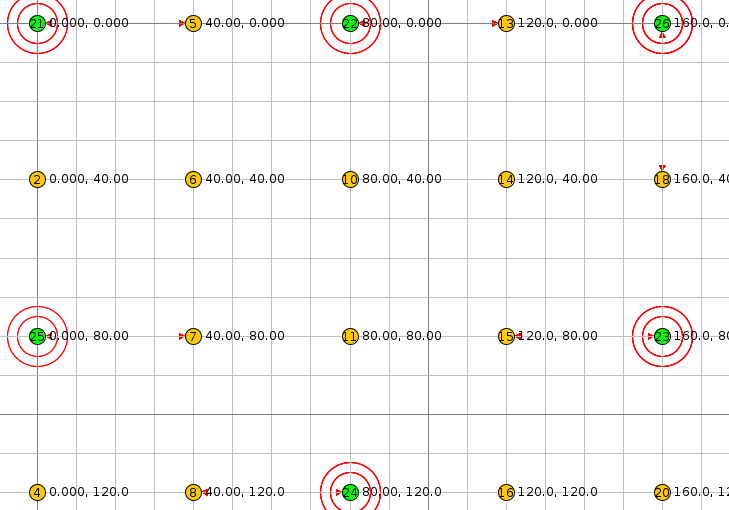
Hình 11 Line topology có 20 node với 4 node truyền và 16 node nhận.



Hình 12 Gird topology có 20 node với 4 node truyền và 16 node nhận.



WAKE\_UP\_COUNTER\_RATE



**Hoạt Động Của Nút Trong Mạng**

Khi nút thức dậy, nó sẽ thực hiện các bước cơ bản sau:

1. **Kiểm Tra Kênh:** Nút sẽ kiểm tra xem có dữ liệu cần nhận hoặc truyền không.
2. **Truyền/Nhận Dữ Liệu:** Nếu có dữ liệu, nút sẽ thực hiện các thao tác truyền hoặc nhận dữ liệu.
3. **Ngủ:** Nếu không có hoạt động nào cần thiết, nút sẽ quay lại trạng thái ngủ để tiết kiệm năng lượng.

Khi phân tích CSL (Coordinated Synchronized Listening) và ContikiMAC, ngoài \*\*khả năng chịu tải\*\* và \*\*khả năng di động\*\*, bạn nên xem xét các yếu tố sau để có một đánh giá toàn diện:

### 1. \*\*Hiệu quả năng lượng (Energy Efficiency)\*\*

- \*\*Cách thức tiết kiệm năng lượng\*\*: So sánh cách hai giao thức quản lý thời gian ngủ và thức dậy của các nút để tiết kiệm năng lượng. CSL sử dụng đồng bộ hóa để tối ưu hóa việc thức dậy, trong khi ContikiMAC dựa vào các chu kỳ nghe ngẫu nhiên.

- \*\*Mức độ tiêu thụ năng lượng\*\*: Đánh giá mức tiêu thụ năng lượng trong cả hai giao thức trong các tình huống mạng có tải khác nhau.

### 2. \*\*Độ trễ (Latency)\*\*

- \*\*Thời gian truyền thông điệp\*\*: Đo lường và so sánh độ trễ từ khi một gói tin được gửi đến khi nó được nhận tại đích. CSL có thể giảm độ trễ do đồng bộ hóa, trong khi ContikiMAC có thể gặp khó khăn hơn trong việc giữ độ trễ thấp trong môi trường tải cao.

- \*\*Khả năng đáp ứng thời gian thực\*\*: Đánh giá khả năng của cả hai giao thức trong việc đáp ứng các yêu cầu thời gian thực của ứng dụng, đặc biệt trong các tình huống yêu cầu thông tin phải được truyền nhanh chóng.

### 3. \*\*Khả năng mở rộng (Scalability)\*\*

- \*\*Khả năng xử lý số lượng lớn nút\*\*: Xem xét xem giao thức có thể mở rộng để hỗ trợ một số lượng lớn nút trong mạng mà không làm suy giảm hiệu suất.

- \*\*Tính linh hoạt trong môi trường thay đổi\*\*: Đánh giá cách hai giao thức thích ứng với sự thay đổi trong số lượng nút, mật độ mạng, và cấu trúc mạng.

### 4. \*\*Quản lý xung đột (Collision Management)\*\*

- \*\*Khả năng giảm thiểu xung đột\*\*: So sánh các cơ chế giảm thiểu xung đột của hai giao thức, như cơ chế phát hiện va chạm hoặc tránh va chạm.

- \*\*Hiệu quả sau va chạm\*\*: Đánh giá cách mà hai giao thức xử lý và phục hồi sau khi xảy ra xung đột, và ảnh hưởng của điều này đến hiệu suất mạng.

### 5. \*\*Khả năng mở rộng mạng (Network Expansion)\*\*

- \*\*Dễ dàng trong việc thêm các nút mới\*\*: So sánh khả năng của hai giao thức trong việc tích hợp các nút mới vào mạng mà không cần cấu hình lại hoặc gây gián đoạn cho mạng hiện có.

- \*\*Khả năng tự tổ chức\*\*: Đánh giá khả năng của hai giao thức trong việc tự tổ chức và tự quản lý mà không cần can thiệp từ bên ngoài.

### 6. \*\*Khả năng bảo mật (Security)\*\*

- \*\*Khả năng chống lại các cuộc tấn công\*\*: Xem xét cách mà cả hai giao thức có thể bảo vệ mạng khỏi các cuộc tấn công như nghe trộm, giả mạo gói tin, hoặc tấn công từ chối dịch vụ (DoS).

- \*\*Bảo mật khi đồng bộ hóa\*\*: Đánh giá các rủi ro bảo mật liên quan đến quá trình đồng bộ hóa trong CSL và cách ContikiMAC xử lý các vấn đề bảo mật trong môi trường không đồng bộ.

### 7. \*\*Khả năng thích nghi (Adaptability)\*\*

- \*\*Thích ứng với điều kiện mạng thay đổi\*\*: Đánh giá khả năng của hai giao thức trong việc tự động điều chỉnh theo các thay đổi trong điều kiện mạng như nhiễu sóng, suy giảm tín hiệu, hoặc sự thay đổi trong lưu lượng mạng.

- \*\*Thích ứng với sự thay đổi yêu cầu ứng dụng\*\*: Xem xét cách mà hai giao thức có thể được điều chỉnh để phù hợp với các yêu cầu ứng dụng khác nhau, như độ trễ, năng lượng, và độ tin cậy.

### 8. \*\*Khả năng triển khai (Deployment Feasibility)\*\*

- \*\*Độ phức tạp khi triển khai\*\*: So sánh mức độ phức tạp trong việc triển khai hai giao thức này trong các ứng dụng thực tế, bao gồm chi phí phần cứng, yêu cầu phần mềm, và độ khó trong cấu hình.

- \*\*Khả năng bảo trì\*\*: Đánh giá độ dễ dàng trong việc bảo trì và cập nhật các nút trong mạng sử dụng hai giao thức này, đặc biệt là trong các môi trường khó khăn.

Bằng cách phân tích các yếu tố này, bạn sẽ có được cái nhìn toàn diện hơn về ưu và nhược điểm của CSL và ContikiMAC, từ đó giúp đưa ra quyết định chọn giao thức phù hợp nhất cho ứng dụng cụ thể của mình.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

* Yêu cầu thiết kế
  + Liệt kê các yêu cầu đặt ra
  + Ghi cụ thể (có tính định lượng) các yêu cầu, chi tiết kỹ thuật.
* **Phân tích**
  + Phân tích rõ cách thức dẫn đến phương pháp thiết kế từ yêu cầu đã đặt ra
  + Nêu rõ ưu điểm và khuyết điểm của từng phương pháp, từ đó lựa chọn phương pháp phù hợp
* Vẽ sơ đồ khối tổng quát và **giải thích** (nếu mạch đơn giản thì lược bỏ phần này)
  + Phải giải thích rõ nhiệm vụ, chức năng từng khối
* Vẽ sơ đồ khối chi tiết và **giải thích**
  + Phải giải thích rõ nhiệm vụ, chức năng từng khối
* Tính toán và vẽ sơ đồ mạch chi tiết
  + Thiết kế, vẽ sơ đồ mạch chi tiết và tính toán từng khối đã nêu trong phần trên

# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM (NẾU CÓ)

* Yêu cầu đặt ra cho phần mềm
  + Liệt kê các yêu cầu đặt ra
  + Ghi cụ thể (có tính định lượng) các yêu cầu, chi tiết kỹ thuật.
* **Phân tích**
  + Phân tích các yêu cầu để đưa ra phương pháp thực hiện chương trình
* Vẽ lưu đồ giải thuật tổng quát và **giải thích** (nếu giải thuật đơn giản thì lược bỏ phần này)
  + Phải giải thích rõ nhiệm vụ, chức năng từng phần
* Vẽ lưu đồ giải thuật chi tiết và **giải thích**
  + Phải giải thích rõ nhiệm vụ, chức năng từng phần

# KẾT QUẢ THỰC HIỆN

Trong phần này, sinh viên mô tả:

* Trình bày **cách thức đo đạc, thử nghiệm** 
  + Ghi rõ các thiết bị sử dụng và sơ đồ kết nối trong việc thử nghiệm
  + Ghi rõ các phần mềm sử dụng trong việc viết và thực thi chương trình
  + Ghi rõ cách bước tiến hành thử nghiệm (phần cứng và phần mềm)
* Trình bày số liệu đo đạc
  + Thực hiện thu thập số liệu trong nhiều trường hợp
  + Ghi rõ số liệu đo đạc thu được dưới hình thức bảng biểu, đồ thị …
* **Giải thích và phân tích về kết quả thu được**
  + Cần giải thích rõ ràng số liệu thu được trên các bảng biểu, đồ thị, dạng sóng …
  + Phân tích các số liệu để biết kết quả đã thực hiện là phù hợp, đạt yêu cầu

Nếu những bảng số liệu và kết quả mô phỏng quá nhiều, sinh viên có thể trình bày đưa vào phần Phụ Lục.

Ví dụ về hình minh họa: (dùng chức năng **Insert Caption** để tạo liên kết cho Danh sách hình minh họa)



Hình 5‑1 Kết quả thi công



Hình 5‑2 Kết quả mô phỏng

Ví dụ về Bảng số liệu

Bảng 1 Thông số hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số 1 | Thông số 2 | Thông số 3 | Thông số 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Sinh viên tóm tắt những điều rút ra được từ kết quả đề tài, những kinh nghiệm có được sau khi thực hiện đề tài. **Ưu và khuyết điểm** của kết quả nghiên cứu đề tài cũng được trình bày trong mục này. Sinh viên cần so sánh với mục tiêu đặt ra trong chương 1.

## Hướng phát triển

Sinh viên trình bày hướng phát triển và khả năng ứng dụng của đề tài

# TÀI LIỆU THAM KHẢO

Trong mục này, sinh viên liệt kê những tài liệu đã tham khảo khi thực hiện đề tài luận văn. Những nội dung trình bày ở mục trên có tham khảo tài liệu thì sinh viên cần ghi chú bằng chỉ số (ví dụ [1], [2]). Chỉ số này cần tương ứng danh mục tài liệu tham khảo. Sinh viên xem thêm hướng dẫn cách viết trích dẫn kiểu IEEE.

Ví dụ:

1. Tống Văn On, “Thiết kế mạch số với VHDL & Verilog”, Nhà xuất bản Lao động Xã Hội, 2007.
2. Altera Corp., “SDRAM Controller for Altera’s DE2/ DE1 boards”, [www.altera.com](http://www.altera.com)

# PHỤ LỤC

Trong phần này, sinh viên có thể trình bày:

* Những kết quả nghiên cứu bổ sung mà trong phần Kết quả luận văn chưa trình bày hết.
* Phần mã nguồn chương trình, sinh viên cũng có thể trình bày trong mục này. Để ngắn gọn, sinh viên chỉ đưa những mã nguồn chính vào phần Phụ lục.
* Sơ đồ toàn mạch chi tiết