

LỜI NÓI ĐẦU

Ngày nay nhờ có những tiến bộ nhanh chóng trong khoa học và công nghệ sự phát triển của những mạng bao gồm các cảm biến giá thành rẻ, tiêu thụ ít năng lượng và đa chức năng đã nhận được những sự chú ý đáng kể. Hiện nay người ta đang tập trung triển khai các mạng cảm biến để áp dụng vào trong cuộc sống hàng ngày. Đó là các lĩnh vực về y tế, quân sự, môi trường, giao thông... Trong một tương lai không xa, các ứng dụng của mạng cảm biến sẽ trở thành một phần không thể thiếu trong cuộc sống con người nếu chúng ta phát huy được hết các điểm mạnh mà không phải mạng nào cũng có được như mạng cảm biến.

Tuy nhiên mạng cảm ứng đang phải đối mặt với rất nhiều thách thức, một trong những thách thức lớn nhất đó là nguồn năng lượng bị giới hạn khả năng xử lý thấp, giá thành thấp, giải thông bé, tín hiệu yếu và hoạt động dưới tần số chia sẻ. Hiện nay rất nhiều nhà nghiên cứu đang tập trung vào việc cải thiện khả năng sử dụng hiệu quả năng lượng của mạng cảm biến trong từng lĩnh vực khác nhau.

Trong quá trình tìm hiểu và nghiên cứu về mạng cảm biến, em đã lựa chọn và tìm hiểu về việc nâng cấp hiệu năng mạng để khai thác hiệu quả thông qua việc lựa chọn các phương pháp xâm nhập môi trường MAC phù hợp kết hợp lựa chọn phương pháp định tuyến để được phương pháp tối ưu nhất, và em quyết định lựa chọn đề tài này làm đồ án tốt nghiệp

Hải Phòng tháng 6 năm 2010

Sinh viên

Vũ Văn Hưng

MỤC LỤC

LỜI NÓI ĐẦU	1
GIỚI THIỆU.....	4
CHƯƠNG 1 : TỔNG QUAN VỀ MẠNG CẢM BIẾN KHÔNG DÂY.....	7
(WIRELESS SENSOR NETWORK - WSN)	7
1. Định nghĩa.....	7
2. Động lực phát triển	7
3. Cấu trúc của WSN	8
3.1. Node cảm biến	8
3.2. Mạng cảm nhận.....	8
4. Những thách thức của WSN	11
5. Sự khác nhau giữa WSN và mạng truyền thống.....	12
6. Ứng dụng của WSN.....	12
CHƯƠNG 2 : MỘT SỐ GIAO THỨC MAC VÀ ĐỊNH TUYẾN	
TRONG MẠNG CẢM BIẾN KHÔNG DÂY	17
2.1. Giao Thức Mac	17
2.1.1 Yêu cầu thiết kế giao thức MAC cho mạng cảm biến không dây	18
2.1.2. Các nguyên nhân gây nên lãng phí năng lượng.....	21
2.1.3. Các giao thức MAC trong mạng cảm nhận không dây	22
2.2. Định tuyến trong mạng cảm biến	42
2.2.1. Giới thiệu	42
2.2.2 Thách thức trong vấn đề định tuyến	42
2.2.3. Các vấn đề về thiết kế giao thức định tuyến	43
2.2.4. Phân loại và so sánh các giao thức định tuyến	45
2.2.5. Giao thức trung tâm dữ liệu	47
2.2.6. Giao thức phân cấp.....	53
2.2.7. Giao thức dựa trên vị trí.....	57
2.2.8. Kết luận.....	61
CHƯƠNG 3: KẾT HỢP GIỮA MAC VÀ GIAO THỨC ĐỊNH TUYẾN	
MERLIN.....	62
3.1 Giới Thiệu.....	62

3.2 Thiết kế	64
3.3 Thiết Lập.....	72
3.4 Kết Luận.....	76
CHƯƠNG 4: MÔ PHỎNG VÀ ĐÁNH GIÁ	77
4.1 :Mô Phỏng giao thức MERLIN được viết bằng công cụ prowlser trên môi trường Matlab	77
4.2 Đánh giá về giao thức Mac và giao thức định tuyến	78
Kết Luận.....	85
<i>Tài liệu tham khảo</i>	86

GIỚI THIỆU

Ngày nay nhờ tiến bộ vượt bậc trong khoa học và công nghệ, mạng cảm biến đã trở thành đề tài nghiên cứu nóng bỏng và nhận được sự tiến bộ đáng kể trong vài năm qua. Mạng cảm biến là mạng vô tuyến bao gồm các thiết bị cảm biến được phân bố một cách ngẫu nhiên trong không gian, nhằm quan sát các hiện tượng vật lý, hay điều kiện môi trường như nhiệt độ, âm thanh, sự chấn động, áp suất, sự chuyển động, ô nhiễm ở các vị trí khác nhau.

Sự phát triển của mạng cảm biến mở đầu là các ứng dụng trong quân đội ví dụ như giám sát chiến trường. Tuy nhiên bây giờ mạng cảm biến còn được sử dụng trong nhiều lĩnh vực dân dụng bao gồm: quan sát môi trường sống, chăm sóc sức khỏe, nhà tự động hay điều khiển giao thông.

Các con cảm biến là các thiết bị điện tử nhỏ, thông thường được trang bị bộ thu phát vô tuyến hoặc các thiết bị không dây khác, một bộ vi xử lý nhỏ và một nguồn năng lượng. Các con cảm biến này có khả năng thu thập, xử lý và truyền thông tin đến các nút khác và ra thế giới bên ngoài.

Mạng cảm biến là một lĩnh vực rất sâu rộng, đồ án này sẽ giới thiệu một cách khái quát nhất về các đặc điểm của mạng cảm biến, các giao thức Mac và định tuyến. Sau đó phần cuối sẽ nghiên cứu và đưa ra giải thuật về việc nâng cấp hiệu năng mạng để khai thác hiệu quả thông qua lựa chọn các phương pháp xâm nhập môi trường Mac phù hợp, kết hợp lựa chọn phương pháp định tuyến để được phương pháp tối ưu nhất

Đồ án này gồm có 4 chương:

Chương 1: Tổng quan về mạng cảm biến. Chương này trình bày những khái niệm chung nhất về WSNs và đưa ra cấu trúc của mạng cảm biến. Đồng thời cũng nêu ra các ứng dụng cụ thể trong nhiều lĩnh vực cuộc sống.

Chương 2: Các giao thức đặc trưng của giao Mac và định tuyến trong mạng cảm biến không dây. Chương này trình bày về các giao thức Mac, định tuyến trong mạng cảm biến và những nguyên nhân gây lãng phí năng lượng khi xâm nhập môi trường.

Chương 3: Kết hợp giữa giao thức Mac và giao thức định tuyến < MERLIN>.

Chương này giới thiệu về MERLIN ,nêu ra những phương pháp kết hợp giữa giao thức Mac và giao thức định tuyến

Chương 4: Mô phỏng và đánh giá thủ tục đơn giản, thông qua chương trình Prowler mô phỏng các giao thức trong mạng cảm nhận không dây trên matlab. Và cuối cùng là phần **kết luận** trình bày tóm tắt lại những kết quả đã đạt được và đưa ra hướng phát triển trong tương lai.

LỜI CẢM ƠN

Để có thể hoàn thành được đồ án tốt nghiệp này, em đã được học hỏi những kiến thức quý báu từ các thầy, cô giáo của Trường Đại Học Dân Lập Hải Phòng trong suốt bốn năm đại học. Em vô cùng biết ơn sự dạy dỗ, chỉ bảo tận tình của các thầy, các cô trong thời gian học tập này.

Em xin bày tỏ lòng biết ơn tới thầy Nguyễn Trọng Thể - Khoa công nghệ thông tin – Trường Đại Học Dân Lập Hải Phòng đã tận tình chỉ bảo và định hướng cho em nghiên cứu đề tài này. Thầy đã cho em những lời khuyên quan trọng trong suốt quá trình hoàn thành đồ án. Cuối cùng, em xin cảm ơn gia đình và bạn bè luôn tạo điều kiện thuận lợi, động viên và giúp đỡ em trong suốt thời gian học tập, cũng như quá trình nghiên cứu, hoàn thành đồ án này.

Do hạn chế về thời gian thực tập, tài liệu và trình độ bản thân, bài đồ án của em không thể tránh khỏi những thiếu sót, rất mong các thầy cô góp ý và sửa chữa để bài đồ án tốt nghiệp của em được hoàn thiện hơn. Em xin chân thành cảm ơn!

CHƯƠNG 1 :

TỔNG QUAN VỀ MẠNG CẢM BIẾN KHÔNG DÂY

(WIRELESS SENSOR NETWORK - WSN)

1. Định nghĩa

Mạng cảm nhận không dây (WSN) có thể hiểu đơn giản là mạng liên kết các node với nhau bằng kết nối sóng vô tuyến, trong đó các node mạng thường là các thiết bị đơn giản, nhỏ gọn, giá thành thấp ... và có số lượng lớn, được phân bố một cách không có hệ thống trên một diện tích rộng (phạm vi hoạt động rộng), sử dụng nguồn năng lượng hạn chế và có thể hoạt động trong môi trường khắc nghiệt (chất độc, ô nhiễm, nhiệt độ cao...).

2. Động lực phát triển

Trong những năm gần đây, rất nhiều mạng cảm nhận không dây đã và đang được phát triển và triển khai cho nhiều các ứng dụng khác nhau: theo dõi sự thay đổi của môi trường, khí hậu, theo dõi và điều khiển giao thông, các phương tiện xe cộ,...

Hơn nữa, với sự tiến bộ của công nghệ và sự hội tụ của hệ thống các công nghệ như: kỹ thuật vi điện tử, giao tiếp không dây, công nghệ mạch tích hợp, vi mạch phần cảm ứng, xử lý và tính toán tín hiệu,... làm nền tảng thúc đẩy, tạo ra những node cảm biến có kích thước nhỏ, đa chức năng, giá thành thấp, công suất tiêu thụ thấp, làm tăng khả năng ứng dụng rộng rãi của mạng cảm nhận không dây.

Khi nghiên cứu về mạng cảm nhận không dây, một trong những đặc điểm quan trọng và then chốt đó là thời gian sống của node cảm biến hay chính là sự giới hạn về năng lượng của chúng. Các node cảm biến này yêu cầu tiêu thụ công suất thấp. Các node cảm biến hoạt động có giới hạn và nói chung là không thể thay thế được nguồn cung cấp. Do đó, trong khi mạng truyền thông tập trung vào đạt được các dịch vụ chất lượng cao, thì các giao thức mạng cảm nhận phải tập trung vào vấn đề tiết kiệm năng lượng.

3. Cấu trúc của WSN

3.1. Node cảm biến

Một node cảm biến được cấu tạo bởi 3 thành phần cơ bản sau: vi điều khiển, sensor, bộ phát radio. Ngoài ra, còn có các cổng kết nối với máy tính.

3.1.1.1. Vi điều khiển

- Bao gồm: CPU; bộ nhớ ROM, RAM; bộ phận chuyển đổi tín hiệu tương tự thành tín hiệu số và ngược lại

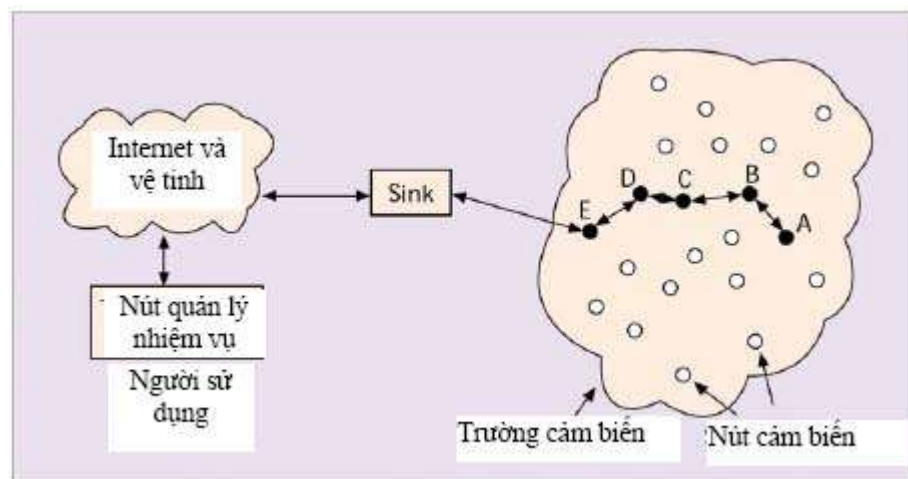
3.1.1.2. Sensor

- Chức năng: cảm nhận thế giới bên ngoài, sau đó chuyển dữ liệu qua bộ phận chuyển đổi để xử lý.

3.1.1.3. Bộ phát radio

Bởi vì node cảm biến là thành phần quan trọng nhất trong WSN, do vậy việc thiết kế các node cảm biến sao cho có thể tiết kiệm được tối đa nguồn năng lượng là vấn đề quan trọng hàng đầu.

3.2. Mạng cảm nhận



Hình 1.3.1. Phân bố node cảm biến trong trường cảm biến

Như hình 1.3.1, chúng ta thấy, mạng cảm nhận bao gồm rất nhiều các node cảm biến được phân bố trong một trường cảm biến. Các node này có khả năng thu thập dữ liệu thực tế, sau đó chọn đường (thường là theo phương pháp đa bước nhảy) để chuyển những dữ liệu thu thập này về *node gốc*. Node gốc liên

lạc với *node quản lý nhiệm vụ* thông qua Internet hoặc vệ tinh. Việc thiết kế mạng cảm nhận như mô hình trong Hình 1.3.1 phụ thuộc vào nhiều yếu tố như:

- **Khả năng chịu lỗi:** Một số các node cảm biến có thể không hoạt động nữa do thiếu năng lượng, do những hư hỏng vật lý hoặc do ảnh hưởng của môi trường. Khả năng chịu lỗi thể hiện ở việc mạng vẫn hoạt động bình thường, duy trì những chức năng của nó ngay cả khi một số node mạng không hoạt động.

- **Khả năng mở rộng:** Khi nghiên cứu một hiện tượng, số lượng các node cảm biến được triển khai có thể đến hàng trăm nghìn node, phụ thuộc vào từng ứng dụng mà con số này có thể vượt quá hàng trăm nghìn node. Do đó cấu trúc mạng phải có khả năng mở rộng để phù hợp với từng ứng dụng cụ thể.

- **Giá thành sản xuất:** Vì mạng cảm nhận bao gồm một số lượng lớn các node cảm biến nên chi phí mỗi node là rất quan trọng trong việc điều chỉnh chi phí mạng. Do vậy chi phí cho mỗi node cảm biến phải giữ ở mức thấp.

- **Tích hợp phần cứng:** Vì số lượng node cảm biến trong mạng là nhiều nên node cảm biến cần phải có các ràng buộc phần cứng sau: kích thước nhỏ, tiêu thụ năng lượng ít, chi phí sản xuất thấp, thích ứng với môi trường, có khả năng tự cấu hình và hoạt động không cần sự giám sát.

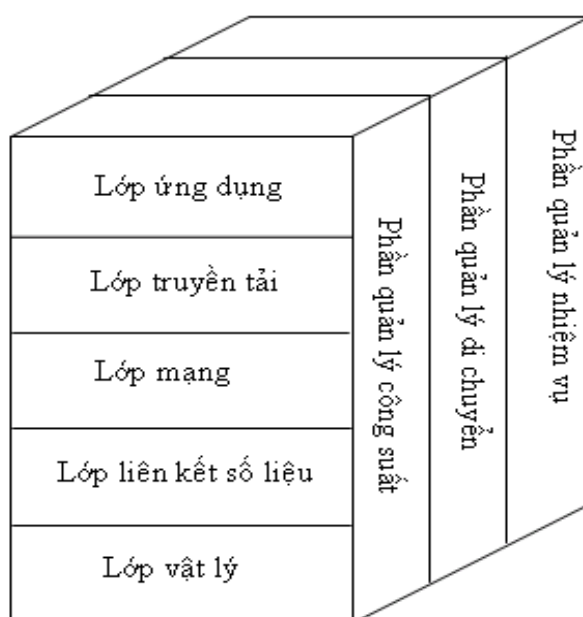
- **Môi trường hoạt động:** Các node cảm biến thường là khá dày đặc và phân bố trực tiếp trong môi trường (kể cả môi trường ô nhiễm, độc hại hay dưới nước,...) => node cảm biến phải thích ứng với nhiều loại môi trường và sự thay đổi của môi trường.

- **Các phương tiện truyền dẫn:** Ở mạng cảm nhận, các node được kết nối với nhau trong môi trường không dây, môi trường truyền dẫn có thể là sóng vô tuyến, hồng ngoại hoặc những phương tiện quang học. Để thiết lập được sự hoạt động thống nhất chung cho các mạng này thì các phương tiện truyền dẫn phải được chọn phù hợp trên toàn thế giới.

- **Cấu hình mạng cảm nhận:** Mạng cảm nhận bao gồm một số lượng lớn các node cảm biến, do đó phải thiết lập một cấu hình ổn định.

- **Sự tiêu thụ năng lượng:** Mỗi node cảm biến được trang bị nguồn năng lượng giới hạn. Trong một số ứng dụng, việc bổ sung nguồn năng lượng là không thể thực hiện. Vì vậy thời gian sống của mạng phụ thuộc vào thời gian sống của node cảm biến, thời gian sống của node cảm biến lại phụ thuộc vào thời gian sống của phin. Do vậy, hiện nay các nhà khoa học đang nỗ lực tìm ra các giải thuật và giao thức thiết kế cho node mạng nhằm tiết kiệm nguồn năng lượng hạn chế này.

*** Kiến trúc giao thức mạng cảm nhận**



Hình 1.3.2. Kiến trúc giao thức của mạng cảm biến

Kiến trúc giao thức áp dụng cho mạng cảm nhận được trình bày trong hình 1.3.2. Kiến trúc này bao gồm các lớp và các mặt phẳng quản lý. Các mặt phẳng quản lý này làm cho các node có thể làm việc cùng nhau theo cách có hiệu quả nhất, định tuyến dữ liệu trong mạng cảm nhận di động và chia sẻ tài nguyên giữa các node cảm biến.

+ **Lớp vật lý:** có nhiệm vụ lựa chọn tần số, tạo ra tần số sóng mang, phát hiện tín hiệu, điều chế và mã hóa tín hiệu.

+ **Lớp liên kết số liệu:** có nhiệm vụ ghép các luồng dữ liệu, phát hiện các khung dữ liệu, cách truy cập đường truyền và điều khiển lỗi. Vì môi trường có tạp âm và các node cảm biến có thể di động, giao thức điều khiển truy nhập môi

trường (MAC) phải xét đến vấn đề công suất và phải có khả năng tối thiểu hóa việc va chạm với thông tin quảng bá của các node lân cận.

+ **Lớp mạng:** quan tâm đến việc chọn đường số liệu được cung cấp bởi *lớp truyền tải*

+ **Lớp truyền tải:** giúp duy trì luồng số liệu nếu ứng dụng mạng cảm nhận yêu cầu. Lớp truyền tải chỉ cần thiết khi hệ thống có kế hoạch được truy cập thông qua mạng Internet hoặc các mạng bên ngoài khác.

+ **Lớp ứng dụng:** tùy theo nhiệm vụ cảm biến, các loại phần mềm ứng dụng khác nhau có thể được xây dựng và sử dụng ở ớp ứng dụng.

+ **Mặt phẳng quản lý công suất:** điều khiển việc sử dụng công suất của node cảm biến. Ví dụ:

- node cảm biến có thể tắt bộ thu sau khi nó nhận được một bản tin để tránh tạo ra các bản tin giống nhau.

- Khi mức công suất của node cảm biến thấp, nó sẽ phát quảng bá sang các node cảm biến bên cạnh thông báo rằng mức năng lượng của nó thấp và nó không thể tham gia vào quá trình định tuyến. Công suất còn lại được giành cho nhiệm vụ cảm biến.

+ **Mặt phẳng quản lý di chuyển:** có nhiệm vụ phát hiện và đăng ký sự chuyển động của các node. Từ đó có thể xác định xem ai là node hàng xóm của mình.

+ **Mặt phẳng quản lý nhiệm vụ:** có nhiệm vụ cân bằng và sắp xếp nhiệm vụ cảm biến giữa các node trong một vùng quan tâm. Tuy nhiên không phải tất cả các node trong vùng đó đều thực hiện nhiệm vụ cảm biến tại cùng một thời điểm.

4. Những thách thức của WSN

Để WSN thực sự trở nên rộng khắp trong các ứng dụng, một số thách thức và trở ngại chính cần vượt qua:

- Vấn đề về năng lượng
- Năng lực xử lý, tính toán
- Bộ nhớ lưu trữ

- Thích ứng tốt với môi trường
- Ngoài ra, còn có một số thách thức và trở ngại thứ yếu như: vấn đề mở rộng mạng, giá thành các node, quyền sở hữu,...

5. Sự khác nhau giữa WSN và mạng truyền thống

Dựa vào sự trình bày ở trên, ta dễ dàng nhận thấy sự khác nhau giữa WSN và các mạng truyền thống:

- Số lượng node cảm biến trong một mạng cảm nhận lớn hơn nhiều lần so với những node trong các mạng truyền thống.
- Các node cảm biến thường được triển khai với mật độ dày hơn.
- Những node cảm biến dễ hỏng, ngừng hoạt động hơn.
- Cấu trúc mạng cảm nhận thay đổi khá thường xuyên.
- Mạng cảm nhận chủ yếu sử dụng truyền thông quảng bá, trong khi đó đa số các mạng truyền thống là điểm - điểm.
- Những node cảm biến có giới hạn về năng lượng, khả năng tính toán và bộ nhớ.
- Những node cảm biến có thể không có số định dạng toàn cầu (global identification) (ID).
- Truyền năng lượng hiệu quả qua các phương tiện không dây
- Chia sẻ nhiệm vụ giữa các node lân cận

6. Ứng dụng của WSN

WSN bao gồm các node cảm biến nhỏ gọn, thích ứng được với môi trường khắc nghiệt. Những node cảm biến này, cảm nhận môi trường xung quanh, sau đó gửi những thông tin thu được đến trung tâm để xử lý theo ứng dụng. Các node không những có thể liên lạc với các node xung quanh nó, mà còn có thể xử lý dữ liệu thu được trước khi gửi đến các node khác. WSN cung cấp rất nhiều những ứng dụng hữu ích ở nhiều lĩnh vực trong cuộc sống.

* Các ứng dụng trong bảo vệ môi trường

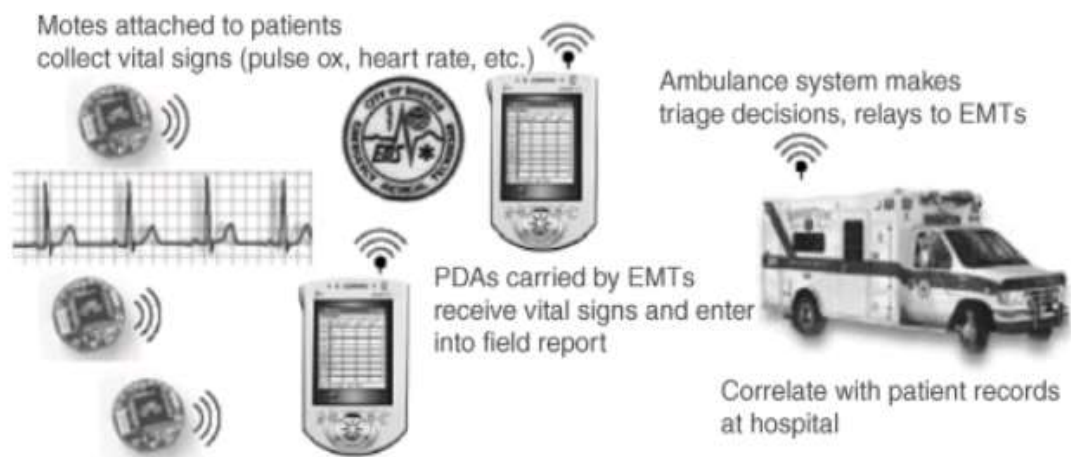
- Phát hiện mìn, chất độc trong môi trường
- Giám sát lũ lụt, bão, gió, mưa,...
- Phát hiện ô nhiễm, chất thải,...
- Phát hiện hoạt động núi lửa
- Phát hiện động đất
- Giám sát cháy rừng

.....

* Các ứng dụng trong y tế

- Định vị theo dõi bệnh nhân
- Hệ thống báo động khẩn cấp
- Cảm biến gắn trực tiếp lên cơ thể con người
- Phân tích nồng độ các chất
- Chăm sóc sức khỏe
- Hỗ trợ chăm sóc bệnh nhân

.....

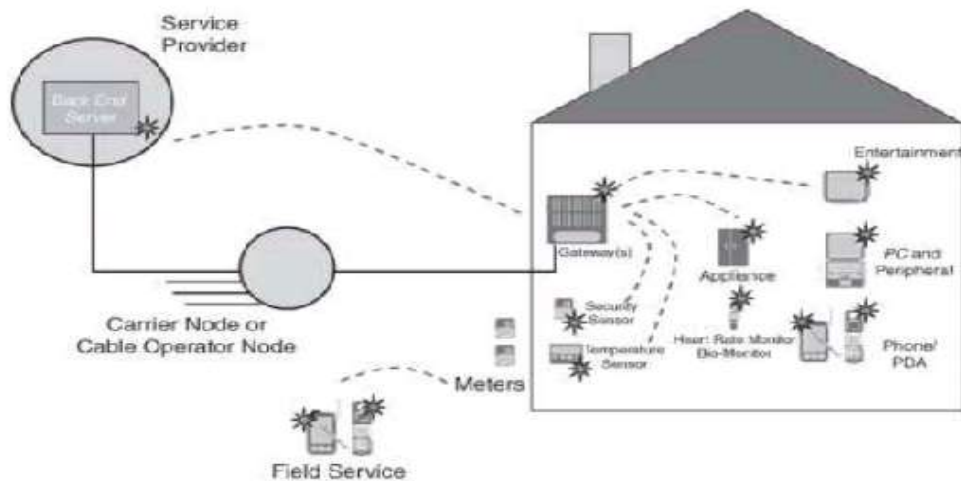


Hình 1.4.1. Ứng dụng trong y tế

* Các ứng dụng trong gia đình

- Hệ thống giao tiếp và điều khiển từ xa các thiết bị
- Hệ thống cảnh báo an ninh,...

.....

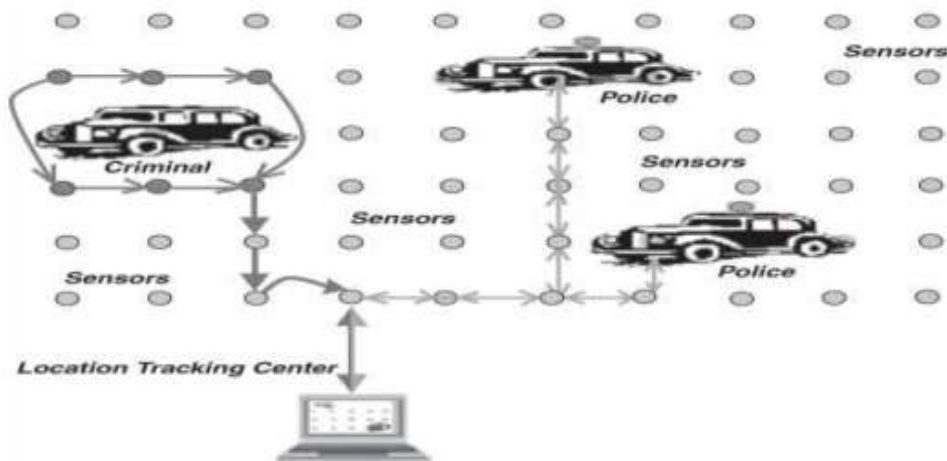


Hình 1.4.2. Ứng dụng điều khiển trong gia đình

Như hình 1.4.2, các node cảm biến được lắp trên các thiết bị giải trí, đo nhiệt độ trong ngôi nhà hoặc cảnh báo an ninh,... ở vị trí cần thiết, sau đó kết nối thành mạng, truyền dữ liệu về nơi cung cấp dịch vụ => cho phép chủ nhà có thể có thể quản lý từ xa các thiết bị đồ dùng, đảm bảo sự an toàn của ngôi nhà,... một cách thuận tiện, dễ dàng.

*** Hệ thống giao thông thông minh**

- Giao tiếp giữa biển báo và phương tiện giao thông
- Hệ thống điều tiết lưu thông công cộng
- Hệ thống báo hiệu tai nạn, kẹt xe,...
- Hệ thống định vị phương, trợ giúp điều khiển tự động phương tiện giao thông



Hình 1.4.3. Ứng dụng định vị phương tiện giao thông

*** Ứng dụng trong quân sự, an ninh**

- Định vị, theo dõi sự di chuyển của các thiết bị quân sự
- Điều khiển tự động các thiết bị, robot,...
- Kích hoạt thiết bị, vũ khí quân sự
- Theo dõi biên giới kết hợp với vệ tinh
-

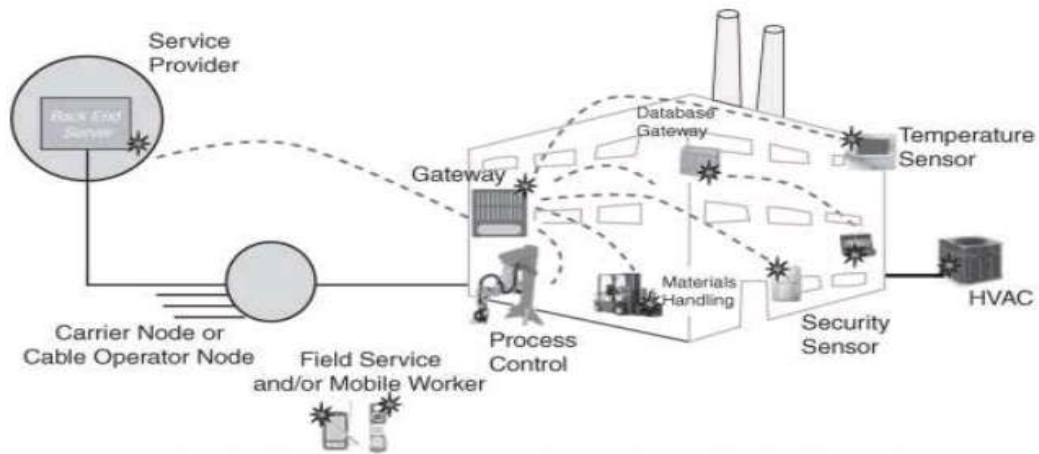


Hình 1.4.4. Ứng dụng cảm biến trong quân sự

Các ứng dụng trong quân sự và an ninh, đặc biệt giúp ích cho con người trong việc tránh xa các vùng nguy hiểm (đặc biệt các vùng có chiến tranh) mà vẫn giám sát được các hoạt động chiến trường.

*** Ứng dụng trong thương mại**

- Quản lý kiến trúc và xây dựng
- Quản lý sản xuất
- Hệ thống xử lý vật liệu
- Quản lý tải trong tiêu thụ điện năng
- Điều khiển nhiệt độ
- Hệ thống tự động
- Thu thập dữ liệu thời gian thực



Hình 1.4.5. Các ứng dụng trong công nghiệp

Hình 1.4.5, các node cảm biến kết nối thành mạng lưới gửi dữ liệu đến node trung tâm. Các ứng dụng trong sản xuất công nghiệp bao gồm điều khiển, quản lý, hiệu suất và an toàn. Các cảm biến đặt trong môi trường làm việc giám sát quá trình sản xuất, chất lượng sản phẩm, kiểm soát môi trường làm việc, quản lý nhân viên,... dữ liệu đưa về trung tâm để người quản lý có thể đưa ra các quyết định kịp thời.

CHƯƠNG 2 :

MỘT SỐ GIAO THỨC MAC VÀ ĐỊNH TUYẾN TRONG MẠNG CẢM BIẾN KHÔNG DÂY

2.1. Giao Thức Mac

Mạng cảm biến không dây là loại mạng đặc biệt với số lượng lớn nút cảm biến được trang bị bộ vi xử lý, thành phần cảm biến và thành phần quản lý sóng vô tuyến. Các nút cảm biến cộng tác với nhau để hoàn thành một nhiệm vụ chung. Trong nhiều ứng dụng, các nút cảm biến sẽ được triển khai phi cấu trúc như mạng ad hoc. Chúng phải tự tổ chức để hình thành một mạng không dây đa bước nhảy. Thách thức chung trong mạng không dây là vấn đề xung đột do hai nút gửi dữ liệu cùng lúc trên cùng kênh truyền.

Giao thức điều khiển truy nhập đường truyền (MAC) đã được phát triển để giúp đỡ mỗi nút quyết định khi nào và làm sao để truy nhập kênh. Vấn đề này cũng được biết như sự định vị kênh hoặc đa truy nhập. Lớp MAC được xem xét bình thường như một lớp con của lớp liên kết dữ liệu trong giao thức mạng. Những giao thức MAC đã nghiên cứu rộng rãi trên những lĩnh vực truyền thống của truyền thông tiếng nói và dữ liệu không dây. Đa truy nhập phân chia theo thời gian (Time Division multiple Access - TDMA), Đa truy nhập phân chia theo tần số (Frequency Division Multiple Access - FDMA) và đa truy nhập phân chia theo mã (Code Division Multiple Access - CDMA) là những giao thức MAC được sử dụng rộng rãi trong những hệ thống truyền thông tế bào hiện đại.

Ý tưởng cơ bản của các phương pháp trên là sẽ tránh xung đột bởi việc chia nhỏ kênh truyền thành những kênh truyền con, các nút sẽ phân chia truy nhập các kênh truyền con đó. Việc phân chia kênh được thực hiện theo thời gian, tần số hoặc theo mã. Những kênh truyền con này không ảnh hưởng lẫn nhau, những giao thức MAC này được phân vào nhóm phi xung đột (*collision-free*). Lớp giao thức MAC khác dựa trên sự cạnh tranh dành quyền truy nhập trên một kênh dung chung, kết quả trong sự phối hợp xác suất có điều kiện,

không cần cấp phát sẵn kênh truyền. Xung đột có thể xảy ra trong thời gian thủ tục cạnh tranh trong những hệ thống như vậy.

Mạng cảm biến khác với mạng dữ liệu không dây truyền thống trên một vài khía cạnh. Trước hết, đa số các nút trong những mạng cảm biến hoạt động dựa trên nguồn điện pin, và rất khó để nạp điện cho những nguồn pin của tất cả các nút. Thứ hai, những nút thường được triển khai trong một kiểu cách đặc biệt phi cấu trúc; chúng phải tự tổ chức hình thành một mạng truyền thông. Ba là, nhiều ứng dụng cần phải sử dụng số lượng lớn những nút, và mật độ nút sẽ thay đổi tại những địa điểm và thời gian khác nhau, với cả những mạng mật độ thưa lẫn những nút với nhiều lân cận. Cuối cùng, đa số các lưu thông trong mạng được thúc đẩy bởi những sự kiện cảm ứng, phân bố không đều và rất co cụm. Tất cả những đặc trưng này cho thấy những giao thức MAC truyền thống không thích hợp cho những mạng cảm biến không dây nếu không có những sự cải biến.

2.1.1 Yêu cầu thiết kế giao thức MAC cho mạng cảm biến không dây

Tránh xung đột

Tính tránh xung đột (Collision Avoidance) là một yêu cầu cơ bản của tất cả các giao thức MAC, nó xác định khi nào một nút có thể truy nhập đường truyền và thực hiện trao đổi dữ liệu.

Hiệu quả năng lượng

Tính hiệu năng (Energy Efficiency) là một trong những thuộc tính quan trọng nhất những giao thức MAC mạng cảm biến. Như đã đề cập ở trên, đa số các nút cảm biến hoạt động bằng pin, rất khó để thay đổi hoặc nạp điện lại cho pin của những nút này. Thực tế, nhiều mục đích thiết kế của những mạng cảm biến được xây dựng bằng những nút đủ rẻ để vứt bỏ hơn là nạp lại. Trong tất cả các trường hợp, việc kéo dài cả cuộc đời của mỗi nút là một vấn đề then chốt. Dù với nền tảng phần cứng nào, năng lượng cho thu phát sóng vô tuyến là nguồn tiêu thụ năng lượng chính. Lớp MAC trực tiếp điều khiển hoạt động thu phát

sóng vô tuyến, và sự tiêu thụ năng lượng của nó như thế nào ảnh hưởng đáng kể tới cả cuộc đời của nút.

Khả năng thích ứng và biến đổi được

Tính biến đổi được và khả năng thích ứng (Scalability and Adaptivity) là những thuộc tính liên quan của một giao thức MAC điều tiết những sự thay đổi trong kích thước mạng, mật độ và topo mạng. Nhiều nút có thể không hoặc ngừng hoạt động trong thời gian dài; vài nút mới có thể tham gia về sau; một vài nút khác có thể di chuyển tới những vị trí khác. Một giao thức MAC tốt cần phải điều tiết những sự thay đổi như vậy một cách hợp lý. Tính biến đổi được và khả năng thích ứng để thay đổi trong kích thước, mật độ và topo mạng là những thuộc tính quan trọng, bởi vì những mạng cảm biến được triển khai phi cấu trúc và thường hoạt động trong những môi trường không chắc chắn.

Khả năng sử dụng kênh

Sự sử dụng kênh (Channel utilization) phản chiếu toàn bộ băng thông của kênh được dùng trong truyền thông ra sao, nó cũng được đề cập như sự sử dụng băng thông hoặc dung lượng kênh truyền. Đó là một vấn đề quan trọng đối với hệ thống điện thoại tế bào hoặc mạng cục bộ không dây (WLANs), khi băng thông là tài nguyên quý giá nhất trong những hệ thống như vậy và các nhà cung cấp dịch vụ đều muốn càng nhiều người dùng càng tốt. Mặt khác, số những nút hoạt động trong mạng cảm biến chủ yếu về được xác định bởi loại ứng dụng. Sự sử dụng kênh thường là một mục tiêu thứ nhì trong những mạng cảm biến.

Độ trễ

Độ trễ (Latency) đó là sự trì hoãn một nút gửi có một gói tin để gửi cho đến khi gói tin được nhận thành công bởi nút nhận. Trong mạng cảm biến, sự quan trọng của độ trễ phụ thuộc vào ứng dụng. Trong những ứng dụng như giám sát hoặc theo dõi, các nút cảm biến không hoạt động phần lớn thời gian cho đến khi một sự kiện nào đó được phát hiện. Những ứng dụng này có thể thường bỏ qua sự trễ thông điệp bổ sung nào đó, bởi vì tốc độ mạng nhanh hơn tốc độ của

một đối tượng vật lý. Tốc độ cảm biến đối tượng đặt một ranh giới trên về tốc độ phản ứng mà mạng phải đạt được. Trong khoảng thời gian không có sự kiện cảm ứng, có rất ít dữ liệu trao đổi trong mạng. Sự trễ ở mức nhỏ hơn một giây cho một khởi tạo một thông báo sau thời kỳ nhàn rỗi thì không quan trọng bằng sự tiết kiệm năng lượng và thời gian hoạt động của thiết bị. Nhưng ngược lại, sau khi cảm biến xác định được sự kiện, hoạt động với độ trễ thấp thành quan trọng tiêu quan trọng.

Thông lượng

Thông lượng (Throughput) đề cập tới số lượng của dữ liệu chuyển thành công từ một nơi gửi đến một nơi nhận trong một khoảng thời gian cho trước. Nhiều nhân tố ảnh hưởng đến thông lượng, bao gồm hiệu quả của sự tránh xung đột, sự sử dụng kênh, độ trễ, và xử lý thông tin điều khiển. Giống với độ trễ, sự quan trọng của thông lượng phụ thuộc vào loại ứng dụng. Những ứng dụng cảm biến mà yêu cầu vòng đời lâu thường chấp nhận độ trễ nhiều hơn và thông lượng thấp hơn.

Công bằng

Fairness thể hiện khả năng những người dùng, những nút hoặc những ứng dụng khác nhau cùng nhau chia sẻ kênh truyền một cách công bằng. Nó là một thuộc tính quan trọng trong mạng tiếng nói hoặc những mạng dữ liệu truyền thống, một khi mỗi người dùng mong muốn một cơ hội như nhau để gửi hoặc nhận dữ liệu cho những ứng dụng của chính mình. Tuy nhiên, trong những mạng cảm biến, tất cả các nút hợp tác cho một nhiệm vụ chung đơn lẻ. Ở tại thời điểm đặc biệt, một nút có thể có nhiều dữ liệu hơn để gửi so với các nút khác, như vậy, hơn là đối xử với mỗi nút công bằng, thành công được đo bởi sự thực hiện của ứng dụng, Và độ *fairness* đối với từng nút hoặc từng người dùng trở nên ít quan trọng hơn.

Tóm lại, các vấn đề nêu ở trên là những thuộc tính thể hiện những đặc trưng của một giao thức MAC. Đối với mạng cảm biến không dây, những yếu tố

quan trọng nhất là sự tránh xung đột có hiệu quả, hiệu quả năng lượng, tính biến đổi và thích ứng được với mật độ và số lượng nút.

2.1.2. Các nguyên nhân gây nên lãng phí năng lượng

Xung đột

Sự xung đột (Collision) là nguyên nhân đầu tiên gây tiêu phí năng lượng. Khi hai gói được truyền cùng thời điểm sẽ xảy ra xung đột, chúng bị hỏng và phải được loại bỏ. Yêu cầu truyền lại gói tin sau đó sẽ làm phát sinh sự tiêu hao năng lượng. Do đó tất cả các giao thức MAC cố gắng tránh xung đột bằng mọi cách.

Nghe khi rỗi

Nguyên nhân thứ hai gây tiêu hao năng lượng là vấn đề *nghe khi rỗi (Idle Listening)*. Nó xảy ra khi thành phần sóng vô tuyến thực hiện “nghe” kênh xem có dữ liệu không để nhận. Sự tiêu hao này đặc biệt cao trong những ứng dụng mạng cảm biến, nơi không có dữ liệu trao đổi trong thời gian không có sự kiện được cảm biến.

Nhiều giao thức MAC (như CSMA và CDMA) luôn luôn nghe kênh khi hoạt động dù không có dữ liệu để gửi. Chi phí chính xác của vấn đề nghe khi rỗi phụ thuộc vào phần cứng và chế độ hoạt động thành phần sóng vô tuyến. Đa số các mạng cảm biến được thiết kế để hoạt động trong thời gian dài và các nút cảm biến cũng sẽ trong ở trạng thái nghe khi rỗi một thời gian dài. Trong những trường hợp như vậy, nghe khi rỗi là một yếu tố chính trong vấn đề tiêu thụ năng lượng của thành phần sóng vô tuyến.

Nghe thừa

Nguyên nhân thứ ba là vấn đề *nghe thừa (overhearing)* xuất hiện khi một nút nhận được những gói tin mà được dành cho những nút khác. Phải nghe thừa những lưu thông không cần thiết, không giành cho mình có thể là một nhân tố chính gây tiêu hao năng lượng khi lưu lượng, tải truyền tăng và mật độ phân bố nút cao.

Nguyên nhân cuối cùng mà chúng ta xem xét là sự xử lý gói tin điều khiển. Sự gửi, nhận, và nghe những gói tin điều khiển cũng tiêu thụ năng lượng. Khi những gói điều khiển không trực tiếp chuyên chở dữ liệu, chúng cũng làm giảm *goodput*.

Một giao thức MAC thiết kế cho mạng cảm biến phải đạt được yêu cầu tiết kiệm năng lượng bởi việc điều khiển thành phần sóng vô tuyến để tránh hoặc giảm bớt tiêu phí năng lượng do những nguyên nhân trên. Việc tắt thành phần sóng vô tuyến khi nó chưa được cần đến là một chiến lược quan trọng cho việc tiết kiệm năng lượng. Một lược đồ quản lý năng lượng đầy đủ phải xem xét tất cả các nguồn làm tiêu phí năng lượng, không phải là chỉ là thành phần sóng vô tuyến.

2.1.3. Các giao thức MAC trong mạng cảm nhận không dây

CSMA

Các giao thức mà trong đó các trạm làm việc lắng nghe đường truyền trước khi đưa ra quyết định mình phải làm gì tương ứng với trạng thái đường truyền đó được gọi là các giao thức có “cảm nhận” đường truyền (carrier sense protocol). Cách thức hoạt động của CSMA như sau: lắng nghe kênh truyền, nếu thấy kênh truyền rỗi thì bắt đầu truyền khung, nếu thấy đường truyền bận thì trì hoãn lại việc gửi khung.

Thế nhưng trì hoãn việc gửi khung cho đến khi nào?

Có ba giải pháp:

- Theo dõi không kiên trì (Non-persistent CSMA): Nếu đường truyền bận, đợi trong một khoảng thời gian ngẫu nhiên rồi tiếp tục nghe lại đường truyền.
- Theo dõi kiên trì (persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng 1.
- Theo dõi kiên trì với xác suất p (P-persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng p .

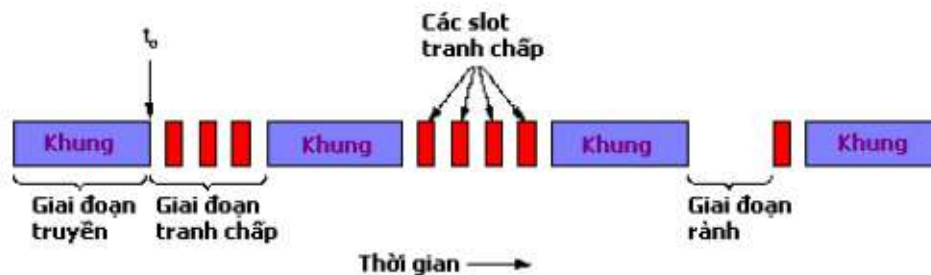
Dễ thấy rằng giao thức CSMA cho dù là theo dõi đường truyền kiên trì hay không kiên trì thì khả năng tránh xung đột vẫn tốt hơn là ALOHA. Tuy thế, xung đột vẫn có thể xảy ra trong CSMA.

Tình huống phát sinh như sau: khi một trạm vừa phát xong thì một trạm khác cũng phát sinh yêu cầu phát khung và bắt đầu nghe đường truyền. Nếu tín hiệu của trạm thứ nhất chưa đến trạm thứ hai, trạm thứ hai sẽ cho rằng đường truyền đang rảnh và bắt đầu phát khung. Như vậy xung đột sẽ xảy ra.

Hậu quả của xung đột là: khung bị mất và toàn bộ thời gian từ lúc xung đột xảy ra cho đến khi phát xong khung là lãng phí.

Bây giờ phát sinh vấn đề mới: các trạm có quan tâm theo dõi xem có xung đột xảy ra không và khi xung đột xảy ra thì các trạm sẽ làm gì?

CSMA/CD (CSMA với cơ chế theo dõi xung đột) về cơ bản là giống như CSMA: lắng nghe trước khi truyền. Tuy nhiên CSMA/CD có hai cải tiến quan trọng là: phát hiện xung đột và làm lại sau xung đột.



Hình 2.1. CSMA/CD có thể ở một trong ba trạng thái:

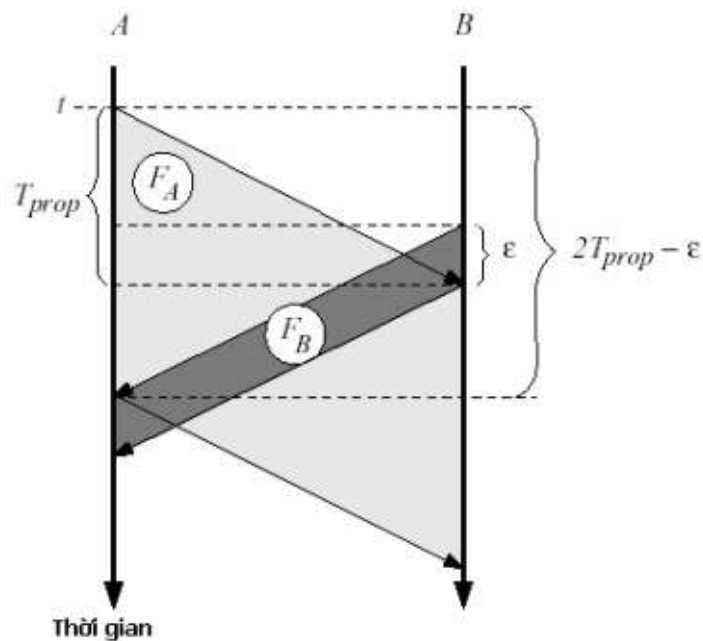
Tranh chấp, truyền, rảnh

Phát hiện xung đột: Trạm vừa truyền vừa tiếp tục dò xét đường truyền. Ngay sau khi xung đột được phát hiện thì trạm ngưng truyền, phát thêm một dãy nhồi (dãy nhồi này có tác dụng làm tăng cường thêm sự va chạm tín hiệu, giúp cho tất cả các trạm khác trong mạng thấy được sự xung đột), và bắt đầu làm lại sau xung đột.

CSMA/CD, cũng giống như các giao thức trong LAN khác, sử dụng mô hình quan niệm như trong hình sau:

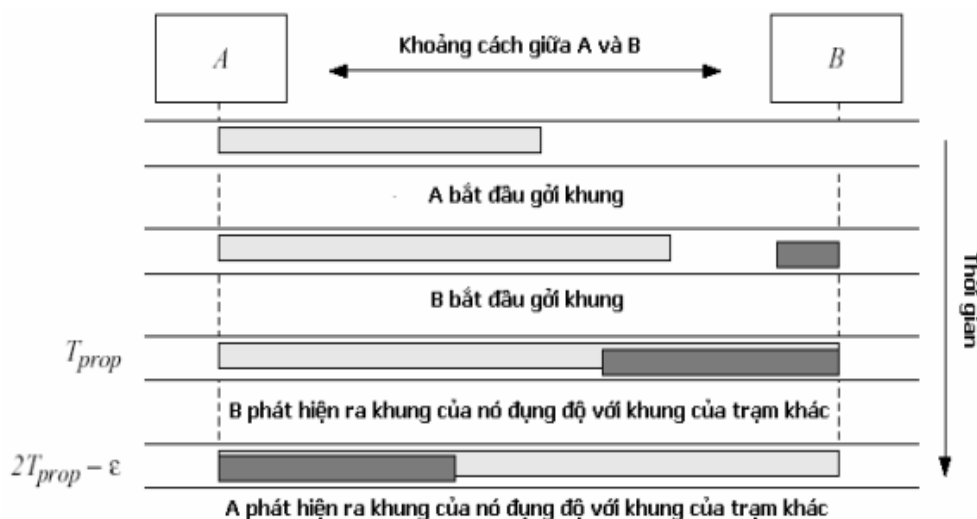
Tại thời điểm t_0 , một trạm đã phát xong khung của nó. Bất kỳ trạm nào khác có khung cần truyền bây giờ có thể cố truyền thử. Nếu hai hoặc nhiều hơn

các trạm làm như vậy cùng một lúc thì sẽ xảy ra xung đột. Xung đột có thể được phát hiện bằng cách theo dõi năng lượng hay độ rộng của xung của tín hiệu nhận được và đem so sánh với độ rộng của xung vừa truyền đi.

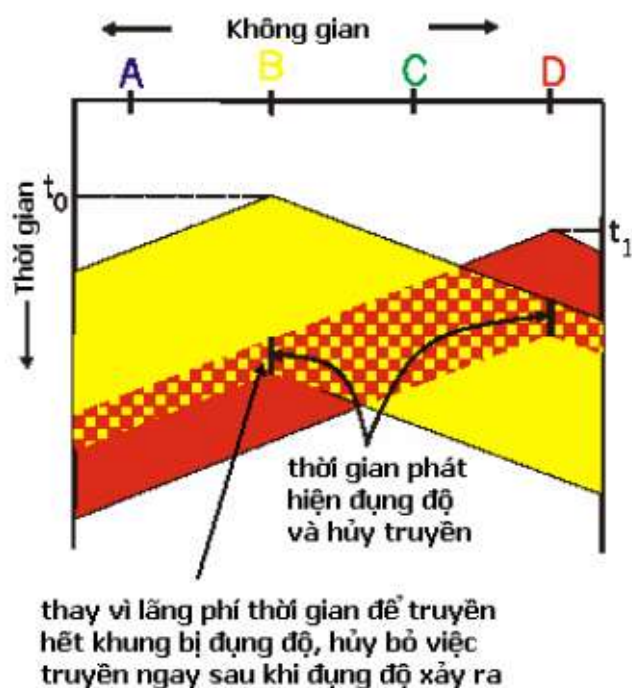


Hình 2.2. Thời gian cần thiết để truyền một khung

Bây giờ ta đặt ra câu hỏi: Sau khi truyền xong khung (hết giai đoạn truyền), trạm sẽ bỏ ra thời gian tối đa là bao lâu để biết được là khung của nó đã bị xung đột hoặc nó đã truyền thành công?



Hình 2.3. Phát hiện xung đột khi truyền tin



Hình 2.4. Xử lý khung xung đột

Hình 2.2 sẽ mô phỏng chi tiết về thời gian phát khung giữa hai trạm A và B ở hai đầu mút xa nhất trên đường truyền tải.

Việc hủy bỏ truyền khung ngay khi phát hiện có xung đột giúp tiết kiệm thời gian và băng thông, vì nếu cứ tiếp tục truyền khung đi nữa, khung đó vẫn hư và vẫn phải bị hủy bỏ.

Làm lại sau khi xung đột: Sau khi bị xung đột, trạm sẽ chạy một thuật toán gọi là back-off dùng để tính toán lại lượng thời gian nó phải chờ trước khi gửi lại khung. Lượng thời gian này phải là ngẫu nhiên để các trạm sau khi quay lại không bị xung đột với nhau nữa.

a. Sensor-MAC

S-MAC được giới thiệu bởi các tác giả: Wei Ye, Jonh Heidermann, Deborah Estrin tại Hội nghị INFOCOM lần thứ 21, năm 2002. Được xây dựng trên nền tảng của các giao thức cạnh tranh như 802.11, S-Mac cố gắng kế thừa sự linh hoạt, tính khả biến của giao thức trên nền cạnh tranh trong khi cải tiến tính hiệu quả sử dụng năng lượng trong mạng đa bước nhảy. S-MAC cố gắng giảm bớt tiêu thụ năng lượng từ tất cả các nguồn được xác định là nguyên nhân

gây tiêu hao năng lượng, đó là: nghe khi rỗi (*idle listening*), xung đột (*collision*), nghe thừa (*overhearing*) và xử lý thông tin điều khiển (*overhead*). Để đạt được mục đích như thiết kế, S-MAC được thiết kế gồm có ba vấn đề chính: thực hiện chu kỳ thức - ngủ; tránh xung đột và nghe thừa; xử lý thông điệp.

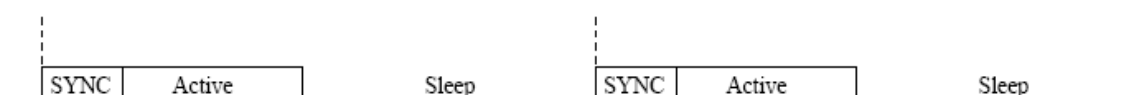
Thực hiện chu kỳ thức/ngủ

Trong những ứng dụng của mạng cảm biến, nút cảm biến thường ở trạng thái nhàn rỗi trong phần lớn thời gian nếu không xuất hiện sự kiện cảm biến. Thực tế tốc độ trao đổi dữ liệu rất thấp do vậy không cần thiết để các nút cảm biến ở trạng thái thức trong tất cả thời gian. S-MAC được thiết kế để giảm bớt thời gian thức bằng cách để cho nút cảm biến định kỳ chuyển sang trạng thái ngủ. Ví dụ, trong chu kỳ một giây, nút cảm biến ở trạng thái ngủ nửa giây và ở trạng thái nghe ở nửa giây còn lại thì chu trình hoạt động giảm bớt tới 50%. Như vậy có thể tiết kiệm được 50% năng lượng.

a, Lược đồ cơ bản

Mỗi nút cảm biến chuyển vào trạng thái “ngủ” trong một khoảng thời gian, sau đó tỉnh dậy và nghe xem liệu có nút nào muốn “nói chuyện” với nó. Trong thời gian ngủ, nút cảm biến tắt bộ phận thu phát vô tuyến và đặt thời gian để quay về trạng thái thức.

Khoảng thời gian cho việc thức và ngủ có thể được lựa chọn theo những ứng dụng khác nhau.



Hình 2.5. Lược đồ S-MAC

Lược đồ trên yêu cầu có định kỳ sự đồng bộ giữa các nút cảm biến trong vùng tránh sai lệch thời gian. Tất cả các nút cảm biến đều tự do lập lịch cho mình chu kỳ thức/ngủ. Tuy nhiên, để giảm bớt phải xử lý những gói tin điều khiển, tốt hơn là để cho các nút trong vùng đồng bộ cùng nhau. Có nghĩa là

chúng thức cùng lúc và chuyển sang trạng thái ngủ cùng lúc. Nhưng cũng cần chú ý trong một mạng đa bước nhảy không phải tất cả các nút lân cận có thể đồng bộ hóa cùng nhau. Hai nút lân cận A và B có thể có lịch khác nhau vì chúng tiến hành đồng bộ với những nút khác nhau, C và D (Hình 2.6).



Hình 2.6. Đồng bộ giữa các nút. Hai nút lân cận A, B có lịch khác nhau vì A đồng bộ với C, B đồng bộ với D

Các nút cảm biến trao đổi với nhau thông tin lịch làm việc của chúng bằng cách phát quảng bá cho tất cả các nút lân cận hiện thời. Điều này bảo đảm rằng tất cả các nút trong vùng vẫn có thể nói chuyện được với nhau dù chúng có lịch làm việc khác nhau. Ví dụ trong Hình 2.6, nếu nút A muốn nói chuyện với nút B, nó chỉ cần đợi cho đến khi B ở trạng thái thức. Nếu có nhiều nút trong vùng lân cận muốn nói chuyện với một nút, thì chúng cần tiến hành cạnh tranh chiếm đường truyền khi nút nhận ở trạng thái thức, sử dụng gói tin RTS (*Request to Send*) và CTS (*Clear to Send*). Nút nào gửi gói tin RTS ra trước sẽ giành quyền truy nhập và nút nhận sẽ trả lời với một gói CTS. Sau đó chúng bắt đầu sự truyền dữ liệu, lúc này chúng không tuân theo lịch làm việc trước đó của chúng cho đến khi chúng kết thúc truyền dữ liệu.

Mặt trái của lược đồ là sự gia tăng độ trễ do duy trì chu kỳ ngủ (*sleep*) của mỗi nút. Hơn nữa, độ trễ có thể tích lũy qua mỗi chặng (*hop*), nên yêu cầu giới hạn độ trễ của ứng dụng tạo ra giới hạn thời gian ngủ trong chu kỳ làm việc của các nút cảm biến.

b, Tiến trình lựa chọn và duy trì lịch làm việc

Trước khi mỗi nút bắt đầu chu kỳ thức/ngủ, nó cần phải chọn một lịch biểu làm việc (khi nào thức, khi nào ngủ) và trao đổi lịch này với các nút lân cận. Mỗi nút duy trì một bảng lưu giữ tất cả các thời gian biểu của các nút lân cận mà nó biết.

Rất hiếm khi xảy ra các nút phải duy trì nhiều thời gian biểu. Các nút sẽ cố gắng chọn một thời gian biểu đã tồn tại trước khi tự chọn cho mình một thời gian biểu độc lập. Mặt khác, xảy ra trường hợp các nút lân cận thất bại trong việc khám phá, phát hiện ra nhau tại thời điểm ban đầu do xung đột khi quảng bá thời gian biểu, thì chúng vẫn có thể tìm thấy nhau trong chu kỳ kế tiếp.

Một tùy chọn khác để cho những nút trên vùng biên chấp nhận duy nhất một thời gian biểu là chấp nhận cái đến trước tiên. Khi nó biết thời gian biểu khác mà một số nút lân cận của nó theo, nó có thể vẫn còn nói chuyện với chúng. Tuy nhiên, với những gói quảng bá, nó cần gửi hai lần với hai thời gian biểu khác nhau. Ưu điểm của phương pháp này là các nút nằm trong vùng biên sẽ có cùng chu kỳ nghe ngủ với những nút khác.

c, Thực hiện đồng bộ

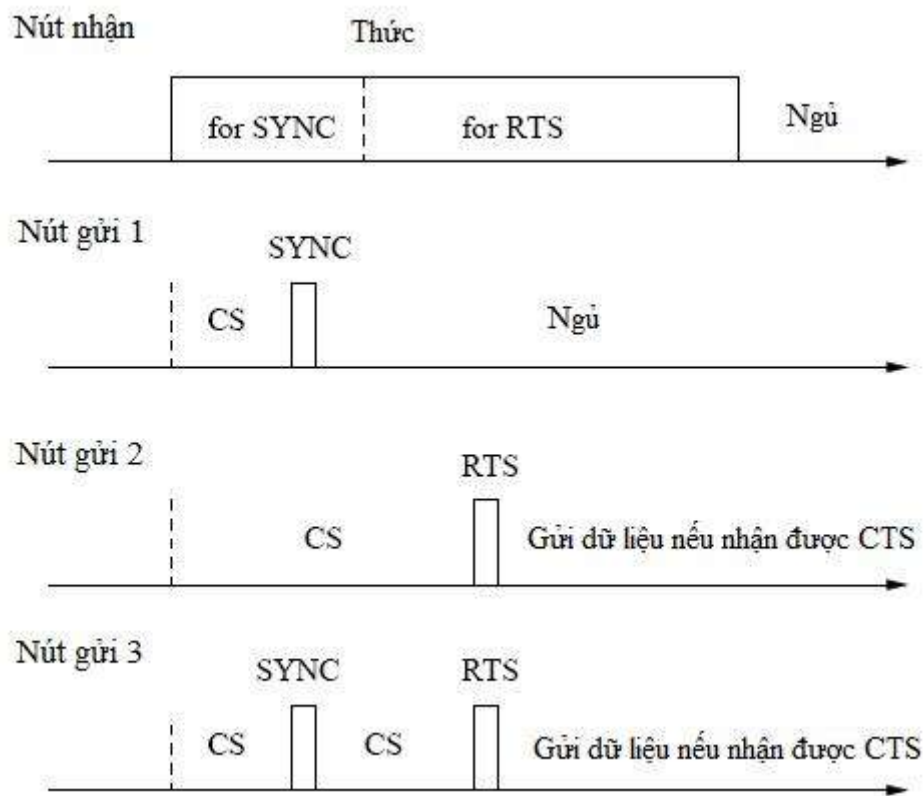
Lược đồ thức/ngủ yêu cầu sự đồng bộ giữa những nút trong vùng lân cận. Việc các nút trong vùng lân cận định kỳ cập nhật lẫn nhau thời gian biểu của chúng là cần thiết để ngăn ngừa sự sai lệch thời điểm của chu kỳ nghe/ngủ.

Việc cập nhật thời gian biểu được thực hiện bằng trao đổi gói tin đồng bộ SYNC. Gói tin SYNC rất ngắn, và bao gồm địa chỉ của nút gửi và thời điểm chuyển sang trạng thái ngủ tiếp theo của nó.

Để một nút nhận được cả những gói đồng bộ lẫn những gói dữ liệu, chúng ta chia khoảng thức (*active time*) của nó thành hai phần. Phần đầu tiên để nhận những gói tin đồng bộ, phần hai để nhận những gói RTS (Hình 2.7). Mỗi phần được chia tiếp thành nhiều khe thời gian cho những nút gửi để thực hiện cảm nhận sóng mang. Ví dụ, nếu một nút gửi muốn gửi một gói tin đồng bộ thì nó khởi động cảm nhận sóng mang khi nút nhận bắt đầu nghe. Nó ngẫu nhiên lựa chọn một khe thời gian để kết thúc cảm nhận sóng mang. Nếu nó không phát hiện ra bất kỳ sự truyền nào vào khoảng cuối khe, thì nó chiếm được đường truyền và bắt đầu gửi gói tin đồng bộ của nó ở tại thời điểm ấy. Việc thực hiện truyền gói dữ liệu cũng được thực hiện tương tự.

Hình 2.7 cũng thể hiện mối quan hệ định thời của ba trường hợp có thể khi một nút gửi thực hiện truyền tới một nút nhận. CS là cảm ứng sóng mang. Trong lược đồ, Nút gửi 1 chỉ gửi một gói tin đồng bộ SYNC. Nút gửi 2 chỉ muốn gửi dữ liệu. Nút gửi 3 gửi một gói tin đồng bộ và một gói tin RTS.

Mỗi nút định kỳ quảng bá những gói tin đồng bộ tới các lân cận của nó kể cả khi nó không có nút đồng bộ theo. Điều này cho phép một nút mới gia nhập nhóm lân cận đã hình thành trước đó. Nút mới thực hiện thủ tục để chọn một thời gian biểu có sẵn làm thời gian biểu của nó. Quãng thời gian nghe đủ dài để nó có khả năng học và theo một thời gian biểu có sẵn trước khi nó tự chọn cho mình một thời gian biểu độc lập.



Hình 2.7. Quan hệ định thời giữa nút nhận và các nút gửi.

. Tránh xung đột và nghe thừa

Tránh xung đột là một nhiệm vụ cơ bản của giao thức MAC. S-MAC sử dụng một lược đồ tránh xung đột trên nền cạnh tranh. Khi một nút phát đi một gói tin, gói tin đó được thu bởi tất cả các nút lân cận của nó mặc dù chỉ một

trong số chúng là nút nhận, đó chính là nghe thừa. Phải nghe thừa làm cho giao thức MAC trên nền cạnh tranh kém hiệu quả về tiết kiệm năng lượng hơn so với những giao thức TDMA, vậy nên nó cần phải tránh.

a, Tránh xung đột

Khi nhiều nút có nhu cầu gửi số liệu vào cùng một thời điểm, chúng cần cạnh tranh để quyết định một nút được quyền gửi (chiếm đường truyền). Trong số những giao thức cạnh tranh, 802.11 thực hiện rất tốt việc tránh xung đột. S-MAC sử dụng các kỹ thuật như chuẩn 802.11, bao gồm cảm nhận sóng mang vật lý, cảm nhận sóng mang ảo lần thực hiện trao đổi RTS/CTS.

Có một trường độ dài phát (*duration field*) trong mỗi gói tin được truyền đi để chỉ rằng việc truyền này sẽ duy trì trong thời gian bao lâu. Như vậy nếu một nút nhận được một gói tin dành cho nút khác, thì nó biết việc nó phải giữ yên lặng bao lâu. Nút ghi giá trị này trong một biến gọi là vector thời gian chiếm giữ mạng (Network allocation Vector - NAV) và đặt một đồng hồ tính giờ cho nó. Vào mọi thời điểm khi đồng hồ NAV hoạt động, nút cảm biến tuần tự giảm giá trị của NAV cho đến khi nó về giá trị 0. Khi một nút có dữ liệu để gửi, đầu tiên nó kiểm tra đồng hồ NAV. Nếu giá trị của NAV khác 0, thì nút xác định rằng đường truyền bận và sẽ không thực hiện phát dữ liệu. Kỹ thuật này được gọi là cảm nhận sóng mang ảo (*Virtual Carrier Sense*).

Cảm nhận sóng mang vật lý được thực hiện ở tại lớp vật lý bằng cách thực hiện nghe kênh để truyền. Thời gian ngẫu nhiên cho việc cảm nhận sóng mang rất quan trọng cho việc tránh xung đột. Đường truyền chỉ được xác định là rỗi nếu cả cảm nhận sóng mang vật lý lẫn cảm nhận sóng mang ảo đều xác định đường truyền rỗi.

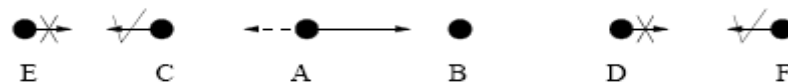
Tất cả các nút gửi thực hiện cảm nhận sóng mang trước khi bắt đầu phát dữ liệu. Nếu một nút thất bại trong việc thăm dò đường truyền, thì nó chuyển sang trạng thái ngủ và thức giấc tại thời điểm nút nhận ở trạng thái nghe và đường truyền rỗi trở lại. Những gói tin quảng bá được gửi mà không sử dụng kỹ

thuật RTS/CTS. Những gói tin Unicast sẽ theo tuần tự RTS/CTS/Data/ACK giữa nút gửi và nút nhận.

b, Tránh nghe thừa

Ở chuẩn 802.11, mỗi nút duy trì trạng thái nghe cho việc truyền tới tất cả các nút lân cận của nó để thực hiện có hiệu quả việc cảm nhận sóng mang ảo. Kết quả là mỗi nút phải nghe thừa nhiều gói không gửi cho nó. Đây là một trong những nguyên nhân chính cho việc tiêu phí năng lượng không cần thiết, đặc biệt khi mật độ nút lớn và lưu lượng mạng tăng.

S-MAC được thiết kế với mục tiêu cố gắng tránh nghe thừa bằng cách để cho những nút có khả năng gây nhiễu không tham gia vào quá trình truyền phát dữ liệu, chuyển sang trạng thái ngủ sau khi chúng nhận được một gói RTS hoặc CTS. Khi những gói dữ liệu luôn dài hơn gói tin điều khiển, cách tiếp cận là ngăn cản các nút lân cận nghe thừa những gói dữ liệu dài và sử dụng gói tin ACK theo sau. Phần tiếp theo sẽ mô tả cách truyền có hiệu quả một gói tin dài kết hợp tránh nghe thừa.



Hình 2.8. Thực hiện tránh nghe thừa. Nút nào nên chuyển tới trạng thái ngủ.

Trong Hình 2.8, nút A, B, C, D, E, Và F hình thành một mạng đa bước nhảy mà từng nút chỉ có thể nghe thông tin truyền từ lân cận hiện thời của nó. Giả thiết nút A đang truyền một gói dữ liệu tới nút B. Câu hỏi đặt ra những nút nào phải chuyển sang trạng thái ngủ.

Xung đột dễ xảy ra ở nút nhận, nút D cần phải ngủ vì sự truyền của nó ảnh hưởng tới sự tiếp nhận tín hiệu của B. Cũng dễ để nhận ra nút E và nút F không phát sinh nhiễu, vì vậy chúng không cần phải ngủ. Nút C có nên đi ngủ hay không? C là cách hai bước tới B, và sự truyền của nó không gây nhiễu tới sự tiếp nhận của B, như vậy nó tự do được phép truyền tới lân cận của nó, ví dụ như E. Tuy nhiên, C không thể nhận bất kỳ sự trả lời nào từ E, vì sự truyền của

E xung đột với sự truyền của A tại nút C. Như vậy sự truyền của C đơn giản là một sự tiêu phí năng lượng. Tóm lại, tất cả lân cận tức thời của cả nút gửi và nút nhận cần phải chuyển trạng thái ngủ khi chúng nghe thấy gói RTS hoặc CTS cho đến khi sự truyền hiện thời kết thúc.

Mỗi nút duy trì NAV để chỉ báo hoạt động trong khu lân cận của nó. Khi một nút nhận một gói dành cho tới những nút khác, nó cập nhật NAV của nó tại trường *duration* trong định dạng gói tin. Một giá trị NAV lớn hơn 0 chỉ báo rằng có một nút đang gửi số liệu trong khu vực lân cận của nó. Giá trị NAV giảm dần theo thời gian. Như vậy một nút cần phải ở trạng thái ngủ để tránh nghe thừa khi giá trị NAV của nó khác 0.

Xử lý thông điệp

Truyền dữ liệu dài trong một gói tin thì chi phí cho việc truyền lại khi chỉ có một vài bit lỗi trong lần truyền đầu tiên là rất cao. Tuy nhiên, nếu chúng ta chia nhỏ thông điệp vào trong nhiều gói nhỏ độc lập, chúng ta phải xử lý quá nhiều gói tin điều khiển do vậy độ trễ truyền sẽ tăng.

S-MAC xử lý vấn đề trên bằng cách chia nhỏ thông điệp dài thành nhiều phân đoạn nhỏ và truyền chúng trong một cụm (burst) nhưng chỉ sử dụng một gói tin RTS và một gói tin CTS. Chúng chiếm dụng đường truyền truyền tất cả các đoạn. Mỗi lần một đoạn dữ liệu được truyền, nơi gửi đợi một xác nhận ACK từ nơi nhận. Nếu nó không nhận được ACK, nó sẽ mở rộng thời gian chiếm dụng đường truyền cho đủ một phân đoạn nữa, và truyền lại ngay phân đoạn dữ liệu hiện thời.

Như đã biết, tất cả các gói tin đều có trường thời gian, bây giờ nó là thời gian cần cho sự phát tất cả các phân đoạn dữ liệu còn lại và những gói ACK. Nếu một nút trong vùng lân cận nhận được một gói RTS hoặc CTS, thì nó sẽ chuyển sang trạng thái ngủ trong khoảng thời gian truyền tất cả các phân đoạn.

Mục đích của việc sử dụng ACK sau mỗi phân đoạn dữ liệu nhằm ngăn ngừa vấn đề nút ẩn (hidden terminal). Có thể một nút lân cận thức dậy hoặc một

nút mới gia nhập vùng lân cận trong quá trình truyền. Nếu nút chỉ là lân cận của nút nhận nhưng không phải nút gửi, thì nó sẽ không nghe thấy các phân đoạn dữ liệu đang được phát từ nút gửi. Nếu nút nhận không gửi ACK thường xuyên, thì nút mới có thể gây nhiễu vì cảm nhận sóng mang trong việc thăm dò đường truyền sẽ thông báo đường truyền rồi. Nếu nó khởi động tiến trình phát, thì quá trình truyền hiện thời sẽ bị hỏng ở tại nút nhận.

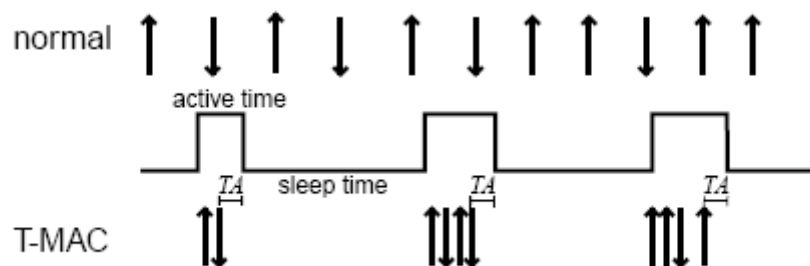
Mỗi phân đoạn dữ liệu và gói tin ACK cũng có trường *thời gian*. Bằng cách này, nếu một nút tỉnh dậy hoặc một nút mới gia nhập trong quá trình truyền, thì nó chuyển sang trạng thái ngủ bất kể nó là lân cận của nút gửi hay nút nhận. Ví dụ, giả sử một nút lân cận nhận được một RTS của nút gửi hoặc một CTS từ nút nhận, nó sẽ chuyển trạng thái ngủ trong toàn bộ thời gian được cung cấp trong thông điệp. Nếu bên gửi mở rộng thời gian truyền do mất phân đoạn dữ liệu hoặc do lỗi, vì ngủ nên các nút lân cận không ý thức được sự mở rộng này ngay lập tức. Tuy nhiên, các nút sẽ biết được điều này từ những phân đoạn mở rộng hoặc những gói tin ACK khi nó tỉnh dậy.

b. Time out-MAC

Mặc dù thực hiện giảm tiêu hao năng lượng bằng việc giảm thời gian chờ nghe qua giải pháp thực hiện chu trình *thức/ngủ* cố định, nhưng giải pháp này của S-MAC chưa đạt hiệu quả tối ưu. S-MAC có hai tham số quan trọng: độ lớn của khung thời gian (*frame time*) và độ dài thời gian thức (*active time*). Độ lớn khung thời gian bị giới hạn bởi yêu cầu về độ trễ cho phép và độ lớn bộ đệm. Độ lớn thời gian thức phụ thuộc chủ yếu trên tốc độ phát sinh thông điệp: nó phải đủ lớn để nút cảm biến có thể phát đi tất cả các thông điệp của nó trong khoảng thời gian thức. Trong khi yêu cầu độ trễ và không gian bộ đệm nói chung là cố định thì tốc độ phát sinh thông điệp thường thay đổi. Để đảm bảo tất cả các thông điệp được phát như mong muốn, nút cảm biến phải được cài đặt một thời gian thức sao cho có thể xử lý ở mức thông lượng cao nhất. Nhưng khi thông lượng xuống thấp thì thời gian thức sẽ không được sử dụng tối ưu và do đó năng lượng sẽ bị lãng phí do vẫn đề nghe khi rồi (*idle listening*).

Giao thức điều khiển truy nhập T-MAC (Timeout-MAC) do hai tác giả Tijs van Dam và Koen Langendoen, khoa Công nghệ thông tin và các hệ thống, Trường đại học công nghệ Delft, Hà Lan, giới thiệu tại Hội nghị quốc tế về các hệ thống mạng cảm biến nhúng lần thứ nhất tại Los Angeles, Mỹ, năm 2003 (Sensys'03), là sự cải tiến S-MAC để khắc phục nhược điểm trên. Ý tưởng mới của giao thức T-MAC là giảm bớt thời gian nghe khi rỗi bằng việc truyền tất cả các thông điệp trong những cụm (*burst*) có độ dài thay đổi tùy theo, và thực hiện ngủ giữa các cụm, xác định một cách mềm dẻo độ dài tối ưu thời gian thức theo sự thay đổi của lưu lượng đường truyền.

Những vấn đề cơ bản



Hình 2.9. Lược đồ cơ bản T-MAC với thời gian thức thay đổi

Hình 2.9 cho thấy lược đồ cơ bản của giao thức T-MAC. Mỗi nút định kỳ tỉnh dậy liên lạc các nút lân cận, sau đó ngủ tiếp cho đến khi khung tiếp theo. Trong lúc đó, những thông điệp mới được đưa vào hàng đợi. T-MAC cũng sử dụng kỹ thuật *RTS*, *CTS*, *Data*, *ACK* để tránh xung đột và truyền số liệu tin cậy.

Một nút sẽ được đặt ở chế độ nghe và sẵn sàng thực hiện truyền số liệu khi nó đang ở trong trạng thái thức. Trạng thái thức sẽ kết thúc khi không có một sự kiện kích hoạt (*activation event*) nào xuất hiện một khoảng thời gian TA. Một sự kiện kích hoạt là:

- + Sự kết thúc một khung thời gian theo định kỳ.
- + Sự tiếp nhận bất kỳ dữ liệu nào trên sóng vô tuyến.
- + Sự xuất hiện sự kiện cảm biến được phát hiện qua thành phần vô tuyến.

+ Sự kết thúc truyền dữ liệu của một nút có sở hữu gói dữ liệu hoặc sự biên nhận ACK;

+ thông tin về sự kết thúc trao đổi dữ liệu của các nút lân cận qua nhận được các gói RTS, CTS.

Thông số TA xác định thời gian tối thiểu cho việc thức chờ nghe trên một khung thời gian.

Lược đồ timeout chuyển tất cả các giao dịch vào một cụm tại điểm bắt đầu của khung. Khi đó những thông điệp giữa các thời gian hoạt động phải được đưa vào bộ đệm. Độ lớn của bộ đệm xác định cận trên của độ lớn khung thời gian cực đại.

Phân nhóm và đồng bộ

Đồng bộ khung thời gian được thực hiện qua sự hình thành phân nhóm ảo như được mô tả trong giao thức S-MAC. Khi một nút cảm biến bắt đầu quá trình hoạt động của mình, nó bắt đầu bằng việc đợi và nghe. Nếu nó không nghe thấy gì trong một khoảng thời gian nhất định, thì nó tự chọn cho mình một lịch làm việc và truyền một gói tin đồng bộ SYNC chứa đựng thời gian khởi tạo của khung tiếp theo trong lịch làm việc. Nếu nút cảm biến trong thời gian khởi động nghe thấy một gói tin đồng bộ từ nút khác, thì nó sẽ theo lịch làm việc trong gói tin đồng bộ đó và quảng bá gói tin đồng bộ tương ứng của chính mình.

Các nút cảm biến thực hiện phát lại ngay gói tin đồng bộ của chúng. Chúng thực hiện nghe đầy đủ một khung một cách không thường xuyên, vì vậy chúng có thể phát hiện ra sự tồn tại của những thời gian biểu khác nhau. Điều này này cho phép các nút mới hoặc các nút di động có thể được chấp nhận gia nhập nhóm đã tồn tại trước đó.

Nếu một nút đã có một thời gian biểu nhưng lại nghe được từ gói tin đồng bộ một thời gian biểu khác (từ nút khác), thì nó chấp nhận cả hai và thực hiện phát một gói tin đồng bộ chứa thời gian biểu của mình để cho các nút khác biết có sự tồn tại thời

gian biểu đó. Việc chấp nhận cả hai thời gian biểu làm việc có nghĩa rằng nút sẽ có những sự kiện kích hoạt ở tại thời điểm bắt đầu của cả hai khung.

Muốn truyền dữ liệu, các nút cảm biến phải khởi động tại điểm bắt đầu khoảng thời gian thức quy định trong lịch biểu của chúng. Tại thời điểm đó, cả các nút lân cận có cùng thời gian biểu, và các lân cận mà đã chấp nhận thời gian biểu như sự bổ sung đều ở trạng thái thức. Nếu nó thực hiện ở điểm bắt đầu của một khung của nút lân cận, thì có thể nó phát trong khi lân cận của nó vẫn đang trong trạng thái ngủ. Với lược đồ này làm có thể thực hiện quảng bá mà chỉ cần phát một lần duy nhất.

Lược đồ đồng bộ được mô tả ở trên được gọi là quá trình phân nhóm ảo, thúc đẩy các nút hình thành nhóm với cùng thời gian biểu mà không bắt buộc thời gian biểu này áp dụng tới tất cả các nút trong mạng. Nó cho phép thực hiện quảng bá có hiệu quả, và tránh sự duy trì thông tin các nút lân cận.

Thực hiện gửi RTS và chọn TA trong T-MAC

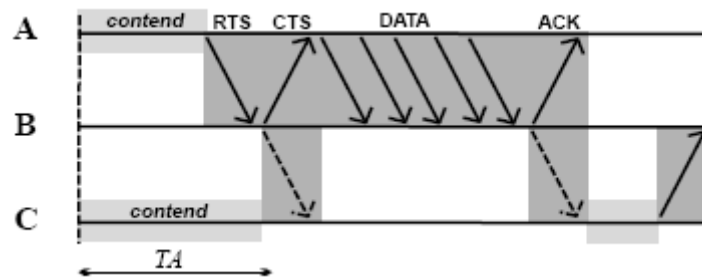
T-MAC cần bổ sung một số đặc tính so với S-MAC để thực hiện sự điều chỉnh tối ưu thời gian thức.

a, Khoảng cạnh tranh cố định (Fixed contention interval)

Trong những giao thức trên nền cạnh tranh, như IEEE 802.11, các nút đợi ngẫu nhiên một khoảng thời gian nhất định, gọi là khoảng thời gian cạnh tranh, sau khi phát hiện có xung đột. Chỉ khi đường truyền rỗi trong thời gian ấy chúng mới khởi động lại sự truyền. Thông thường, một lược đồ back-off được sử dụng: khoảng thời gian cạnh tranh tăng thêm khi lưu lượng đường truyền tăng. Lược đồ back-off giảm bớt xác suất xảy ra xung đột khi tải tăng cao, trong khi tối thiểu độ trễ khi tải thấp.

Trong giao thức T-MAC, mỗi nút truyền các thông điệp trong hàng đợi của nó vào một cụm tại điểm bắt đầu của khung. Trong thời gian truyền cụm này, đường truyền là bão hòa: những thông điệp được truyền ở tốc độ cực đại. Mọi nút đều muốn giành quyền truy nhập đường truyền mỗi khi nó gửi một gói

tin RTS. Khoảng cạnh tranh ngày càng tăng thì lại không có ích khi tải phần lớn đã cao và không thay đổi. Bởi vậy, sự truyền RTS trong T-MAC bắt đầu bởi việc đợi và nghe một khoảng thời gian ngẫu nhiên trong phạm vi một khoảng cạnh tranh cố định. Khoảng này được điều chỉnh phù hợp với tải cực đại. Khoảng thời gian cạnh tranh luôn luôn được sử dụng dù không có xung đột.



Hình 2.10. Lược đồ trao đổi dữ liệu cơ bản. Nút C nghe được CTS từ nút B và sẽ không làm phiên giao tiếp giữa A và B. TA phải đủ dài để C có thể nghe được phần đầu CTS

b, Thử lại phát lại RTS

Khi một nút phát một gói tin RTS, nhưng không nhận được trở lại một CTS, có thể một trong ba trường hợp xảy ra:

- + Nút nhận không nghe được RTS vì xung đột, hoặc
- + Nút nhận bị ngăn cản trả lời vì nghe được RTS hoặc CTS, hoặc
- + Nút nhận đang ngủ.

Khi nút gửi không nhận câu trả lời trong khoảng TA, nó có thể chuyển sang trạng thái ngủ. Tuy nhiên, điều đó có thể không hợp lý trong những trường hợp 1 và 2: sẽ xảy ra hiện tượng nút muốn gửi chuyển sang trạng thái ngủ trong khi nút nhận vẫn thức. Khi trường hợp này xảy ra thậm chí ở ngay tại thông báo đầu tiên của khung, thông lượng giảm đáng kể. Bởi vậy, một nút cần phải cố gắng gửi lại RTS nếu nó không nhận được câu trả lời. Nếu không có còn sự trả lời sau khi thử lại, nó cần phải từ bỏ ý định truyền và sang trạng thái ngủ.

c, Xác định khoảng TA

Một nút không nên chuyển sang trạng thái ngủ trong khi các nút lân cận của nó vẫn còn trao đổi số liệu, một khi nút lân cận đó có thể là nút nhận của một thông báo kế tiếp. Khi bắt đầu nhận được gói tin RTS hoặc CTS của một nút lân cận cũng đủ thực hiện một tác vụ kích hoạt khởi tạo khoảng TA.

Không nằm trong vùng lân cận, nên một nút sẽ không nhận được thông điệp RTS từ một nút mà khởi tạo truyền thông với lân cận của nó. Khoảng TA phải đủ dài để nhận ít nhất bắt đầu của gói CTS (Hình 2.10). Sự quan sát này cho chúng ta một cận dưới của độ dài khoảng TA:

$$TA > C + R + T$$

Ở đây C là chiều dài khoảng cạnh tranh, R là độ dài một gói RTS, và T là thời gian turn-around (khoảng thời gian ngắn giữa kết thúc của gói RTS và sự bắt đầu của gói CTS). Chọn thời gian TA lớn sẽ làm tăng sự tiêu phí năng lượng.

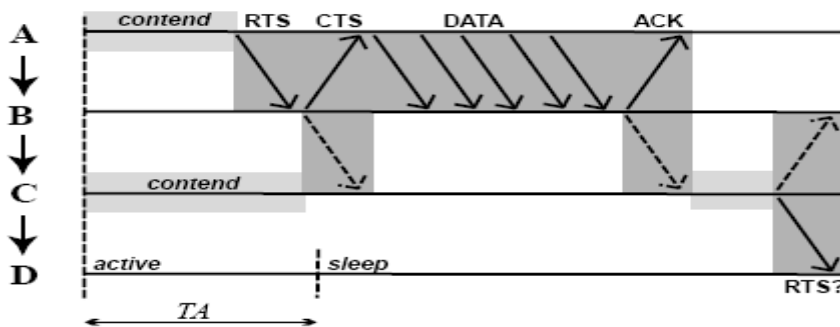
Tránh nghe thừa

Giao thức S-MAC đưa ra ý tưởng nút sẽ sang trạng thái ngủ sau khi nghe được một gói tin RTS hoặc CTS dành cho cho nút khác. Khi đó nút bị ngăn cản việc gửi dữ liệu trong thời gian đó, nó không thể tham gia bất kỳ truyền thông nào và tốt nhất là tắt bộ phận thu phát vô tuyến của nó để tiết kiệm năng lượng.

Tránh nghe thừa là một tùy chọn trong giao thức T-MAC để giảm năng lượng tiêu thụ. Tuy nhiên, chúng sẽ làm xung đột do thông tin điều khiển (*overhead collision*) cao hơn: một nút có thể không nhận được gói tin RTS và CTS trong khi ngủ và làm phiên giao tiếp nào đó khi nó tỉnh dậy trở lại. Do vậy, lưu lượng cực đại giảm bớt. Mặc dầu việc tránh nghe thừa sẽ tiết kiệm điện năng nhưng nó không được sử dụng khi muốn đạt băng thông cực đại.

Truyền thông bất đối xứng

Do lưu lượng trên mạng cảm biến phần lớn là đẳng hướng, như dạng truyền thông từ nhiều nút tới nút gốc (Notes-to-Sink), nên T-MAC xuất hiện hiện tượng làm giảm thông lượng cực đại của mạng. Hiện tượng này được mô tả như sau (Hình 2.11): Các nút từ A đến D hình thành một tế bào với các lân cận của nó. Các thông điệp di chuyển từ trên xuống dưới, như vậy nút A chỉ phát tới B, B chỉ phát tới C, và C chỉ phát tới D.



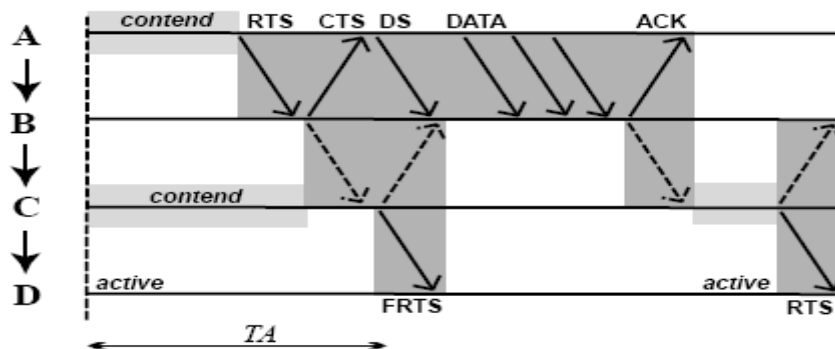
Hình 2.11. Hiện tượng ngủ sớm. D đi ngủ trước khi C gửi một RTS cho nó

Khi nút C muốn phát dữ liệu tới nút D, nó phải tiến hành cạnh tranh, thăm dò đường truyền để giành quyền phát. Việc thăm dò có thể không tiến hành được vì trước đó nó nhận một thông điệp RTS từ nút B, hoặc nghe được thông điệp CTS từ nút B trả lời tới nút A.

Khi C không tiến hành được việc thăm dò đường truyền do nhận được thông điệp RTS từ nút B, nó sẽ trả lời B một thông điệp CTS, D sẽ nghe được thông điệp này và đặt lịch chuyển sang trạng thái thức khi truyền thông giữa C và B kết thúc. Tuy nhiên, nếu C không tiến hành được là do nghe được thông điệp CTS từ B trả lời A (Hình 2.11), nó phải giữ im lặng. Ở trường hợp này, do D không biết truyền thông giữa A và B, không nhận được thông điệp muốn truyền dữ liệu từ C, nó sẽ chuyển sang trạng thái ngủ khi thời gian thức theo lịch kết. Chỉ ở tại điểm bắt đầu của khung tiếp theo, nút C mới có cơ hội để thực hiện thăm dò và tiến hành trao đổi dữ liệu với nút D.

Gửi sớm RTS (Future request to send)

Ý tưởng của giải pháp gửi sớm RTS là sẽ để cho nút nhận tiềm năng (nút D) biết được có một nút muốn gửi dữ liệu cho nó, nhưng đang trong tình trạng phải giữ im lặng để không làm ảnh hưởng đến giao tiếp khác. Khi một nút nghe được một thông điệp CTS dành cho cho nút khác, nó ngay lập tức gửi một gói FRTS (nút C trong Hình 2.12). Gói FRTS chứa thông tin về độ dài của khối dữ liệu truyền thông lấy được trong thông điệp CTS.



Một nút nhận được gói tin FRTS thì nó biết rằng trong khoảng thời gian t tiếp theo nó sẽ nhận được một thông điệp RTS, do vậy phải lập lịch thức trước thời gian ấy. Thông tin thời gian t được lấy trong thông điệp FRTS.

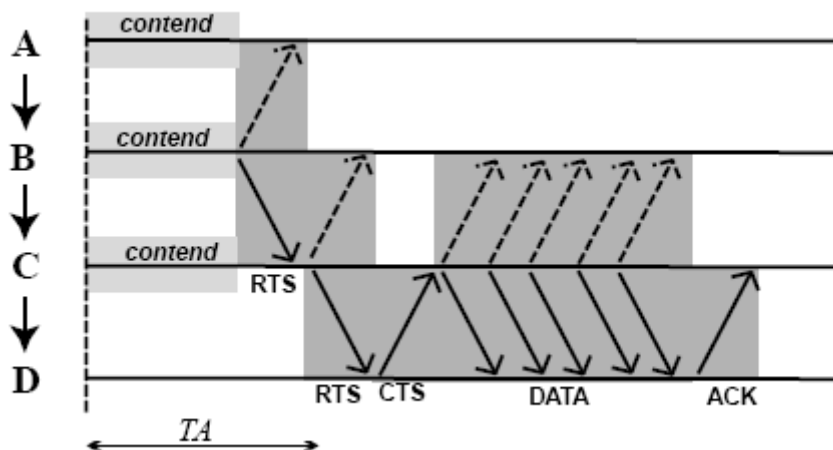
Để thông điệp FRTS (do C phát) không gây nhiễu dữ liệu trao đổi (giữa A và B) theo sau thông điệp CTS, dữ liệu phải được hoãn lại khoảng thời gian truyền FRTS. Để không mất kênh truyền, nút khởi tạo RTS ban đầu (Nút A) truyền một gói tin DS (Data-Send). Sau gói DS, nó ngay lập tức gửi dữ liệu bình thường. FRTS có cùng kích thước với DS, nó sẽ chỉ xung đột với gói DS mà

không phải với gói dữ liệu. Gói DS bị mất, nhưng không có vấn đề gì vì nó không chứa đựng thông tin.

Với giải pháp FRTS, độ dài của khoảng thời gian TA phải được tăng thêm một khoảng bằng độ dài thông điệp CTS. Việc thực hiện giải pháp gửi sớm RTS sẽ làm tăng thông lượng cực đại trong truyền thông đẳng hướng. Tuy nhiên, vì có DS và FRTS, mức tiêu thụ năng lượng cũng tăng thêm. Cũng có thể sử dụng kỹ thuật FRTS trong các dạng truyền thông khác nhưng chỉ khi muốn tăng thông lượng một cách chính đáng. Vì khi tải ở mức thấp thì tốc độ trao đổi dữ liệu cũng thấp do phải gia tăng xử lý thông tin điều khiển.

Thực hiện ưu tiên gửi khi bộ đệm đầy (taking priority on full buffers)

Khi nào bộ đệm truyền/định tuyến của một nút gần đây, thì việc gửi sẽ hợp lý hơn là tiếp tục nhận. Khi một nút nhận được RTS dành cho nó, ngay lập tức nó gửi gói RTS của chính mình cho nút khác, thay vì việc trả lời với một CTS như bình thường.



Hình 2.13. Thực hiện ưu tiên gửi khi bộ đệm đầy

Giải pháp này có hai hiệu quả, trước hết khi nút C khi trả lời B bằng thông điệp RTS khi bộ đệm của nó đầy, một mặt nó trả lời B rằng nó không muốn nhận, mặt khác đồng thời nó cũng thông báo cho D là nó muốn gửi dữ liệu. Như vậy xác suất mà vấn đề ngủ sớm xảy ra sẽ thấp hơn. Hai là, *thực hiện ưu tiên gửi khi bộ đệm đầy* hình thành một giới hạn điều khiển luồng trong mạng có lợi cho

những dạng truyền thông từ nút tới nút gốc. Trong Hình 2.13, nút B bị ngăn gửi cho đến khi nút C có đủ không gian bộ đệm.

2.2. Định tuyến trong mạng cảm biến

2.2.1. Giới thiệu

Mặc dù mạng cảm biến có khá nhiều điểm tương đồng so với các mạng adhoc có dây và không dây nhưng chúng cũng biểu lộ một số các đặc tính duy nhất mà tạo cho chúng tồn tại thành mạng riêng. Chính những đặc tính này làm cho tập trung mũi nhọn vào yêu cầu thiết kế các giao thức định tuyến mới mà khác xa so với các giao thức định tuyến trong các mạng adhoc có dây và không dây. Việc nhằm vào đặc tính này đã đưa ra một tập các thách thức lớn và riêng đối với WSN. Chương này sẽ trình bày ba loại giao thức định tuyến chính hay được dùng trong mạng cảm biến, đó là định tuyến trung tâm dữ liệu (data – centric protocol), định tuyến phân cấp (hierarchical protocol) và định tuyến dựa vào vị trí (location – based protocol).

2.2.2 Thách thức trong vấn đề định tuyến

Chính vì những đặc điểm riêng biệt của mạng cảm biến mà việc định tuyến trong mạng cảm biến phải đối mặt với rất nhiều thách thức sau:

Mạng cảm biến có một số lượng lớn các nút, cho nên ta không thể xây dựng được sơ đồ địa chỉ toàn cầu cho việc triển khai số lượng lớn các nút đó vì lượng mã đầu để duy trì ID quá cao.

Dữ liệu trong mạng cảm biến yêu cầu cảm nhận từ nhiều nguồn khác nhau và truyền đến sink.

Các nút cảm biến bị ràng buộc khá chặt chẽ về mặt năng lượng, tốc độ xử lý, lưu trữ.

Hầu hết trong các ứng dụng mạng cảm biến các nút nói chung là tĩnh sau khi được triển khai ngoại trừ một vài nút có thể di động.

Mạng cảm biến là những ứng dụng riêng biệt.

Việc nhận biết vị trí là vấn đề rất quan trọng vì tập hợp dữ liệu thông thường đưa lên vị trí.

Khả năng dư thừa dữ liệu rất cao vì các nút cảm biến thu lượm dữ liệu dự trên hiện tượng chung.

2.2.3. Các vấn đề về thiết kế giao thức định tuyến

Mục đích chính của mạng cảm biến là truyền thông dữ liệu trong mạng trong khi cố gắng kéo dài thời gian sống của mạng và ngăn chặn việc giảm các kết nối bằng cách đưa ra những kỹ thuật quản lý năng lượng linh hoạt. Trong khi thiết kế các giao thức định tuyến, chúng ta thường gặp phải các vấn đề sau.

Đặc tính thay đổi thời gian và trật tự sắp xếp của mạng

Các nút cảm biến hoạt động với sự giới hạn về khả năng tính toán, lưu trữ và truyền dẫn, dưới ràng buộc về năng lượng khắt khe. Tùy thuộc vào ứng dụng mật độ các nút cảm biến trong mạng có thể từ thưa thớt đến rất dày. Hơn nữa trong nhiều ứng dụng số lượng các nút cảm biến có thể lên đến hàng trăm, thậm chí hàng ngàn nút được triển khai tùy ý và thông thường không bị giám sát bao phủ một vùng rộng lớn.

Trong mạng này, đặc tính của các con cảm biến là có tính thích nghi động và cao, như là nhu cầu tự tổ chức và bảo toàn năng lượng buộc các nút cảm biến phải điều chỉnh liên tục để thích ứng hoạt động hiện tại.

Ràng buộc về tài nguyên

Các nút cảm biến được thiết kế với độ phức tạp nhỏ nhất cho triển khai trong phạm vi lớn để giảm chi phí toàn mạng. Năng lượng là mối quan tâm chính trong mạng cảm biến không dây, làm thế nào để đạt được thời gian sống kéo dài trong khi các nút hoạt động với sự giới hạn về năng lượng dự trữ. Việc truyền gói mutilhop chính là nguồn tiêu thụ năng lượng chính trong mạng. Để giảm việc tiêu thụ năng lượng có thể đạt được bằng cách điều khiển tự động chu kỳ công suất của mạng cảm biến. Tuy nhiên vấn đề quản lý năng lượng đã trở thành một thách thức chiến lược trong nhiều ứng dụng quan trọng.

Mô hình dữ liệu trong mạng cảm biến

Mô hình dữ liệu mô tả luồng thông tin giữa các nút cảm biến và các sink. Mô hình này phụ thuộc nhiều vào bản chất của ứng dụng trong đó cái cách dữ liệu được yêu cầu và sử dụng. Một vài mô hình dữ liệu được đề xuất nhằm tập trung vào yêu cầu tương tác và nhu cầu tập hợp dữ liệu của đa dạng các ứng dụng.

Một loại các ứng dụng của mạng cảm biến yêu cầu mô hình thu thập dữ liệu mà dựa trên việc lấy mẫu theo chu kỳ hay sự xảy ra của sự kiện trong môi trường quan sát.

Trong các ứng dụng khác dữ liệu có thể được chụp và lưu trữ hoặc có thể được xử lý, tập hợp tại một nút trước khi chuyển tiếp dữ liệu đến sink. Một loại thứ 3 đó là mô hình dữ liệu tương tác hai chiều giữa các nút cảm biến và sink.

Nhu cầu hỗ trợ đa dạng các mô hình dữ liệu làm tăng tính phức tạp của vấn đề thiết kế giao thức định tuyến.

Cách truyền dữ liệu

Cái cách mà các truy vấn và dữ liệu được truyền giữa các trạm cơ sở và các vị trí quan sát hiện tượng là một khía cạnh quan trọng trong mạng cảm biến không dây.

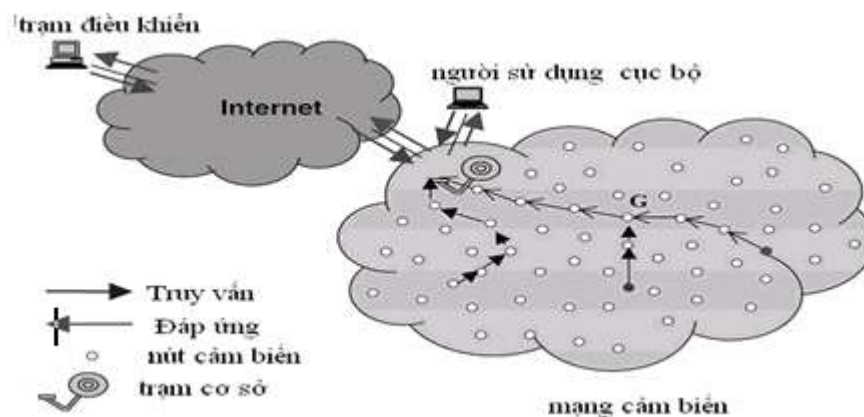
Một phương pháp cơ bản để thực hiện việc này là mỗi nút cảm biến có thể truyền dữ liệu trực tiếp đến trạm cơ sở. Tuy nhiên phương pháp dựa trên bước nhảy đơn (single-hop) có chi phí rất đắt và các nút mà xa trạm cơ sở thì sẽ nhanh chóng bị tiêu hao năng lượng và do đó làm giảm thời gian sống của mạng.

Nhằm giảm thiểu lỗi của phương pháp này thì dữ liệu trao đổi giữa các nút cảm biến và trạm cơ sở có thể được thực hiện bằng việc sử dụng truyền gói đa bước nhảy (multihop) qua phạm vi truyền ngắn. Phương pháp này tiết kiệm năng lượng đáng kể và cũng giảm đáng kể sự giao thoa truyền dẫn giữa các nút khi cạnh tranh nhau để truy cập kênh, đặc biệt là trong mạng cảm biến không dây mật độ cao. Dữ liệu được truyền giữa các nút cảm biến và các sink được minh họa như hình vẽ (3.1).

Để đáp ứng các truy vấn từ các sink hoặc các sự kiện đặc biệt xảy ra tại môi trường thì dữ liệu thu thập được sẽ được truyền đến các trạm cơ sở thông qua nhiều đường dẫn multihop.

Trong định tuyến multihop của mạng cảm biến không dây, các nút trung gian đóng vai trò chuyển tiếp dữ liệu giữa nguồn và đích. Việc xác định xem

tập hợp các nút nào tạo thành đường dẫn chuyển tiếp dữ liệu giữa nguồn và đích là một nhiệm vụ quan trọng trong thuật toán định tuyến. Nói chung việc định tuyến trong mạng kích thước lớn vốn đã là một vấn đề khó khăn, các thuật toán phải nhằm vào nhiều yêu cầu thiết kế thách thức bao gồm sự chính xác, ổn định, tối ưu hóa và chú ý đến sự thay đổi của các thông số.



Hình 2.14: Mô hình truyền dữ liệu giữa sink và các nút

Với đặc tính bên trong của mạng cảm biến bao gồm sự ràng buộc về dải thông và năng lượng đã tạo thêm thách thức cho các giao thức định tuyến là phải nhằm vào việc thỏa mãn yêu cầu về lưu lượng trong khi vẫn mở rộng được thời gian sống của mạng.

2.2.4. Phân loại và so sánh các giao thức định tuyến

Vấn đề định tuyến trong mạng cảm biến là một thách thức khó khăn đòi hỏi phải cân bằng giữa sự đáp ứng nhanh của mạng và hiệu quả. Sự cân bằng này yêu cầu sự cần thiết thích hợp khả năng tính toán và truyền dẫn của các nút cảm biến ngược với mào đầu yêu cầu thích ứng với điều kiện này. Trong mạng cảm biến không dây, mào đầu được đo chính là lượng băng thông được sử dụng, tiêu thụ công suất và yêu cầu xử lý của các nút di động. Việc tìm ra chiến lược cân bằng giữa sự cạnh tranh này cần thiết tạo ra một nền tảng chiến lược định tuyến.

Việc thiết kế các giao thức định tuyến trong mạng cảm biến không dây phải xem xét giới hạn về công suất và tài nguyên của mỗi nút mạng, chất lượng

thay đổi theo thời gian của các kênh vô tuyến và khả năng mất gói và trễ. Nhằm vào các yêu cầu thiết kế này một số các chiến lược định tuyến trong mạng cảm biến được đưa ra.

Bảng(3.1) đưa ra sự phân loại một số giao thức dựa trên nhiều tiêu chí khác nhau. Một loại giao thức định tuyến thông qua kiến trúc phẳng trong đó các nút có vai trò như nhau. Kiến trúc phẳng có một vài lợi ích bao gồm số lượng mã đầu tối thiểu để duy trì cơ sở hạ tầng, và có khả năng khám phá ra nhiều đường giữa các nút truyền dẫn để chống lại lỗi.

Loại thứ 2 là phân cấp theo cụm, lợi dụng cấu trúc của mạng để đạt được hiệu quả về năng lượng, sự ổn định, sự mở rộng. Trong loại giao thức này các nút mạng tự tổ chức thành các cụm trong đó một nút có mức năng lượng cao hơn các nút khác và đóng vai trò là nút chủ. Nút chủ thực hiện phối hợp hoạt động trong cụm và chuyển tiếp thông tin giữa các cụm với nhau. Việc tạo thành các cụm có khả năng làm giảm tiêu thụ năng lượng và mở rộng thời gian sống của mạng.

Loại giao thức định tuyến thứ 3 là sử dụng phương pháp trung tâm dữ liệu để phân bố sự quan tâm (interest) bên trong mạng. Phương pháp này sử dụng thuộc tính dựa trên tên do đó một nút nguồn truy vấn một thuộc tính của hiện tượng hơn là một nút riêng lẻ.

Phân phối quan tâm trong toàn mạng đạt được bằng việc gán nhiệm vụ cho các con cảm biến và nhấn mạnh vào các câu hỏi mà liên quan đến các thuộc tính riêng.

Một giao thức khác có thể truyền quan tâm tới các nút bao gồm quảng bá, các thuộc tính dựa trên multilcasting, geo-casting.

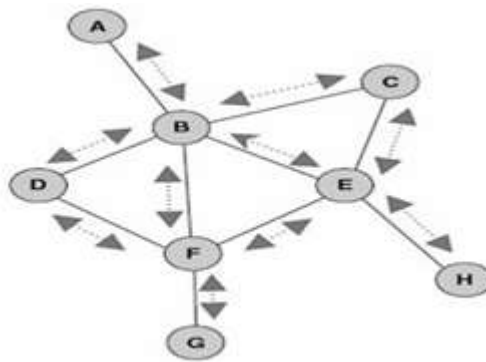
Loại giao thức thứ 4 là dựa vào vị trí để đánh địa chỉ cho các nút cảm biến. lại giao thức này rất có ích cho những ứng dụng nơi mà vị trí của các nút cảm biến trong vùng địa lý được bao phủ bởi mạng liên quan đến truy vấn được đưa ra bởi nút nguồn.

2.2.5. Giao thức trung tâm dữ liệu

Flooding và Gossiping

Flooding là kỹ thuật chung thường được sử dụng để tìm ra đường và truyền thông tin trong mạng adhoc vô tuyến và hữu tuyến.

Chiến lược định tuyến này rất đơn giản và không phụ thuộc vào cấu hình mạng và các giải thuật định tuyến phức tạp. Flood sử dụng phương pháp reactive nhờ đó mỗi nút nhận dữ liệu hoặc điều khiển dữ liệu để gửi các gói tới các nút lân cận. Sau khi truyền, một gói sẽ được truyền trên tất cả các đường có thể. Trừ khi mạng bị ngắt không thì các gói sẽ truyền đến đích (hình 3.2)



Hình 2.15: Truyền gói trong Flooding

Hơn nữa khi cấu hình mạng thay đổi các gói sẽ truyền theo những tuyến mới giải thuật này sẽ tạo ra vô hạn các bản sao của mỗi gói khi đi qua các nút. Giải thuật này có 3 nhược điểm lớn như sau: thứ nhất là hiện tượng bản tin kép. Tức là các 2 gói dữ liệu giống nhau được gửi đến cùng nút.

Thứ hai là hiện tượng chồng chéo, tức là các nút cùng cảm nhận một vùng không gian và do đó tạo ra các gói tương tự nhau gửi đến các nút lân cận.

Và thứ ba đó là thuật toán này không hề quan tâm đến vấn đề năng lượng của các nút, các nút sẽ nhanh chóng tiêu hao năng lượng và làm giảm thời gian sống của mạng.

Một sự cải tiến của giao thức này là Gossiping, thuật toán này cải tiến ở chỗ mỗi nút sẽ ngẫu nhiên gửi gói mà nó nhận được đến một trong các nút lân cận của nó. Thuật toán này làm giảm số lượng các gói lan truyền trong mạng, tránh hiện tượng bản tin kép tuy nhiên có nhược điểm là có thể gói sẽ không bao giờ đến được đích

SPIN

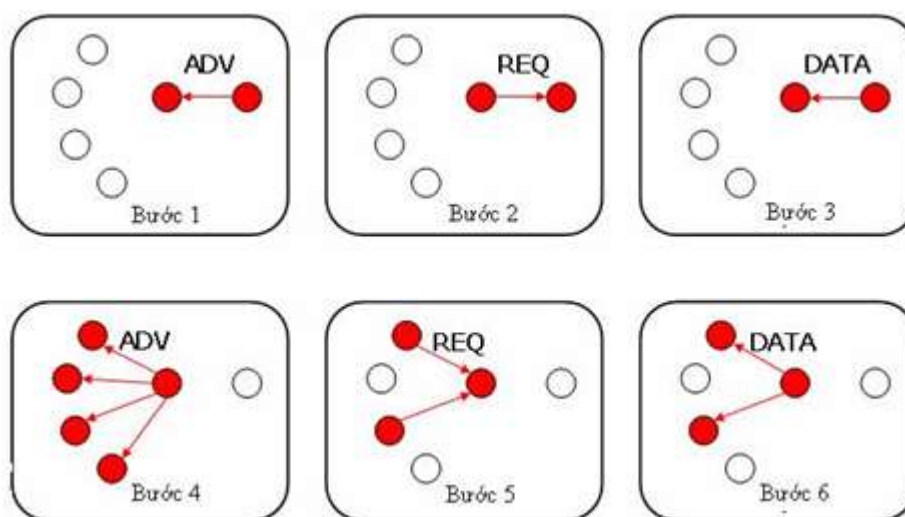
SPIN (Sensor Protocol for Information via Negotiation) là giao thức định tuyến thông tin dựa trên sự đàm xếp dữ liệu. Mục tiêu chính của giao thức này đó là tập trung việc quan sát môi trường có hiệu quả bằng một số các nút cảm biến riêng biệt trong toàn bộ mạng. Nguyên lý của giao thức này đó là sự thích ứng về tài nguyên và sắp xếp dữ liệu. Ý nghĩa của việc đàm xếp dữ liệu (data negotiation) này là các nút trong SPIN sẽ biết về nội dung của dữ liệu trước khi bất kỳ dữ liệu nào được truyền trong mạng .

SPIN khai thác tên dữ liệu nhờ đó mà các nút sẽ kết hợp miêu tả dữ liệu (metadata) với dữ liệu mà chúng tạo ra và sử dụng sự miêu tả này để thực hiện việc giàn xếp dữ liệu trước khi truyền dữ liệu thực tế. Nơi nhận dữ liệu có thể bày tỏ mối quan tâm đến nội dung dữ liệu bằng cách gửi yêu cầu để lấy được dữ liệu quảng bá.

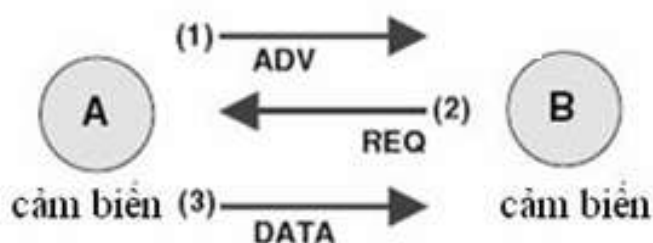
Điều này tạo ra sự sắp xếp dữ liệu để đảm bảo rằng dữ liệu chỉ được truyền đến nút quan tâm đến loại dữ liệu này. Do đó mà loại trừ khả năng bản tin kép và giảm thiểu đáng kể việc truyền dữ liệu dư thừa qua mạng. Hơn nữa việc sử dụng bộ miêu tả dữ liệu cũng loại trừ khả năng chồng lấn vì các nút có thể chỉ giới hạn về tên loại dữ liệu mà chúng quan tâm đến.

Việc thích ứng tài nguyên cho phép các nút cảm biến chạy SPIN có thể thích ứng với trạng thái hiện tại của tài nguyên năng lượng. Mỗi nút có thể dò tìm tới bộ quản lý để theo dõi mức tiêu thụ năng lượng của mình trước khi truyền hoặc xử lý dữ liệu. Khi mức năng lượng còn lại thấp các nút này có thể giảm hoặc loại bỏ một số hoạt động như là truyền miêu tả dữ liệu hoặc các gói. Chính việc thích nghi với tài nguyên làm tăng thời gian sống của mạng.

Để thực hiện truyền và sắp xếp dữ liệu các nút sử dụng giao thức này sử dụng ba loại bản tin (hình 3.3).



Hình 2.16 Ba tín hiệu bắt tay của SPIN



Hình 2.17 Hoạt động của SPIN

Hoạt động của SPIN gồm 6 bước như hình (3.4).

Bước 1: ADV để thông báo dữ liệu mới tới các nút.

Bước 2: REQ để yêu cầu dữ liệu cần quan tâm. Sau khi nhận được ADV các nút quan tâm đến dữ liệu này sẽ gửi REQ để yêu cầu lấy dữ liệu.

Bước 3: bản in DATA bản tin này thực sự chứa dữ liệu được cảm biến và kèm theo mã đầu miêu tả dữ liệu.

Bước 4, sau khi nút này nhận dữ liệu nó sẽ chia sẻ dữ liệu của nó cho các nút còn lại trong mạng bằng việc phát bản tin ADV chứa miêu tả dữ liệu (metadata).

Bước 5: sau đó các nút xung quanh lại gửi bản tin REQ yêu cầu dữ liệu

Bước 6: DATA lại được truyền đến các nút mà yêu cầu dữ liệu này. Tuy nhiên giao thức SPIN cũng có hạn chế khi mà nút trung gian không quan tâm đến dữ liệu nào đó, khi đó dữ liệu không thể đến được đích

Directed Diffusion

Đây là giao thức trung tâm dữ liệu đối với việc truyền và phân bổ thông tin trong mạng cảm biến không dây. Mục tiêu chính của phương pháp này là tiết kiệm năng lượng để tăng thời gian sống của mạng để đạt được mục tiêu này, giao thức này giữ tương tác giữa các nút cảm biến, dựa vào việc trao đổi các bản tin, định vị trong vùng lân cận mạng. Sử dụng sự tương tác về vị trí nhận thấy có tập hợp tối thiểu các đường truyền dẫn. Đặc điểm duy nhất của giao thức này là sự kết hợp với khả năng của nút để có thể tập trung dữ liệu đáp ứng truy vấn của sink để tiết kiệm năng lượng.

Thành phần chính của giao thức này bao gồm 4 thành phần: interest (các mối quan tâm của mạng), data message (các bản tin dữ liệu), gradient, reinforcements.

Directed diffusion sử dụng mô hình publish- and subscribe trong đó một người kiểm tra (tại sink) sẽ miêu tả mối quan tâm (interest) bằng một cặp thuộc tính-giá trị.

Bảng (3.2) miêu tả cặp thuộc tính-giá trị, các nút cảm biến có khả năng đáp ứng interest này sẽ trả lời kèm theo dữ liệu tương ứng.

Hoạt động của Directed Diffusion như hình (3.5). Với mỗi nhiệm vụ cảm biến tích cực, sink sẽ gửi quảng bá bản tin interest theo chu kỳ cho các nút lân cận.

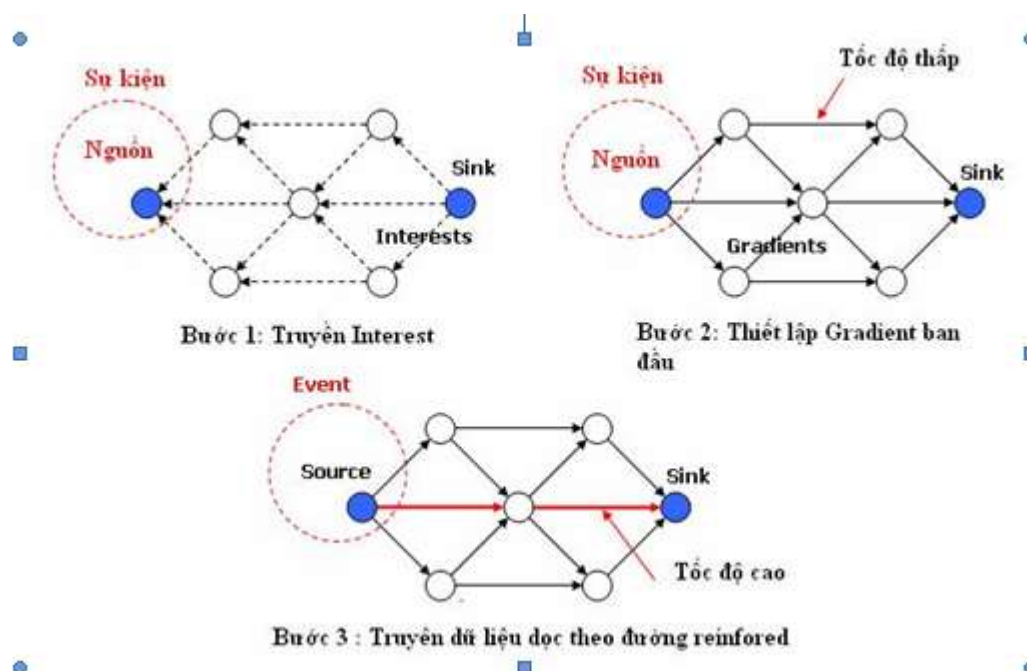
Cặp thuộc tính-giá trị	Miêu tả
Type = chim ruồi	Phát hiện vị trí của chim ruồi
Interval=20ms	Báo cáo sự kiện chu kỳ 20ms
Duration=10s	Thời gian sống của Interest
Field=[(x ₁ ,x ₂),(y ₁ ,y ₂)]	Báo cáo từ các con cảm biến trong vùng

Bảng 2.18: Miêu tả interest sử dụng cặp thuộc tính-giá trị

Bản tin này sẽ truyền qua tất cả các nút trong mạng như là một sự quan tâm đến một dữ liệu nào đó. Mục đích chính của việc thăm dò này là để xem xét xem có nút cảm biến nào đó có thể tìm kiếm dữ liệu tương ứng với interest. Tất cả các nút đều duy trì một interest cache để lưu trữ các interest entry khác nhau. Mỗi một mục (entry) trong interest cache sẽ lưu trữ một interest khác nhau. Các entry cache này sẽ lưu trữ một số trường sau: Một nhãn thời gian (timestamp), nhiều trường gradient cho mỗi nút lân cận và trường duration. Nhãn thời gian sẽ lưu trữ nhãn thời gian của interest nhận được sau cùng. Mỗi gradient sẽ lưu trữ cả tốc độ dữ liệu và chiều mà dữ liệu được gửi đi. Giá trị của tốc độ dữ liệu nhận được từ thuộc tính khoảng thời gian trong bản tin interest. Trường duration sẽ xác định khoảng thời gian tồn tại của interest.

Một gradient có thể coi như là một liên kết phản hồi của nút lân cận khi mà nhận được bản tin interest. Việc truyền bản tin interest trong toàn mạng cùng với việc thiết lập các gradient tại mỗi nút cho phép việc tìm ra và thiết lập các đường dẫn giữa sink mà đưa ra yêu cầu về dữ liệu quan tâm và các nút mà đáp ứng mỗi quan tâm đó.

Khi một nút phát hiện một sự kiện nó sẽ tìm kiếm trong cache xem có interest nào phù hợp không, nếu có nó sẽ tính toán tốc độ sự kiện cao nhất cho tất cả các gradient lỗi ra. Sau đó nó thiết lập một phân hệ cảm biến để lấy mẫu các sự kiện ở mức tốc độ cao này. Các nút sẽ gửi ra ngoài miêu tả về sự kiện cho các nút lân cận có gradient. Các nút lân cận này nhận dữ liệu và sẽ kiểm tra trong cache xem có entry nào phù hợp không, nếu không nó sẽ loại bỏ dữ liệu còn nếu phù hợp nó sẽ nhận dữ liệu các nút này sẽ thêm bản tin vào cache dữ liệu và sau đó gửi bản tin dữ liệu cho các nút lân cận.



Hình 2.19: Hoạt động cơ bản của Directed Diffusion

Khi nhận được một interest các nút tìm kiếm trong interest cache của nó xem có entry nào phù hợp không, nếu không nút sẽ tạo một cache entry mới. Các nút sẽ sử dụng các thông tin chứa trong interest để tạo ra các thông số interest trong entry. Các entry này là một tập hợp chứa các trường gradient với tốc độ và chiều tương ứng với nút lân cận mà interest được nhận. Nếu như interest nhận được có trong cache thì nút sẽ cập nhật nhãn thời gian và trường duration cho phù hợp với entry. Một trường gradient sẽ được remove khỏi entry nếu quá hạn.

Trong pha thiết lập gradient thì các sink sẽ thiết lập một tập hợp các đường dẫn. Sink có thể sử dụng đường dẫn này với sự kiện chất lượng cao để làm tăng tốc độ dữ liệu. Điều này đạt được thông qua một đường dẫn được hỗ trợ xử lý (path reinforcement process). Các sink này có thể sử dụng sự hỗ trợ của một số các nút lân cận. Để làm được điều này sink có thể gửi lại bản tin interest nguồn ở tốc độ cao thông qua các đường dẫn được chọn, nhờ việc tăng cường các nút nguồn trên đường dẫn để gửi dữ liệu thường xuyên hơn. Directed diffusion có ưu điểm nếu một đường dẫn nào đó giữa sink và một nút bị lỗi, một đường dẫn có tốc độ dữ liệu thấp hơn được thay thế.

Kỹ thuật định tuyến này ổn định dưới phạm vi mạng động. Loại giao thức định tuyến này tiết kiệm năng lượng đáng kể.

2.2.6. Giao thức phân cấp

LEACH

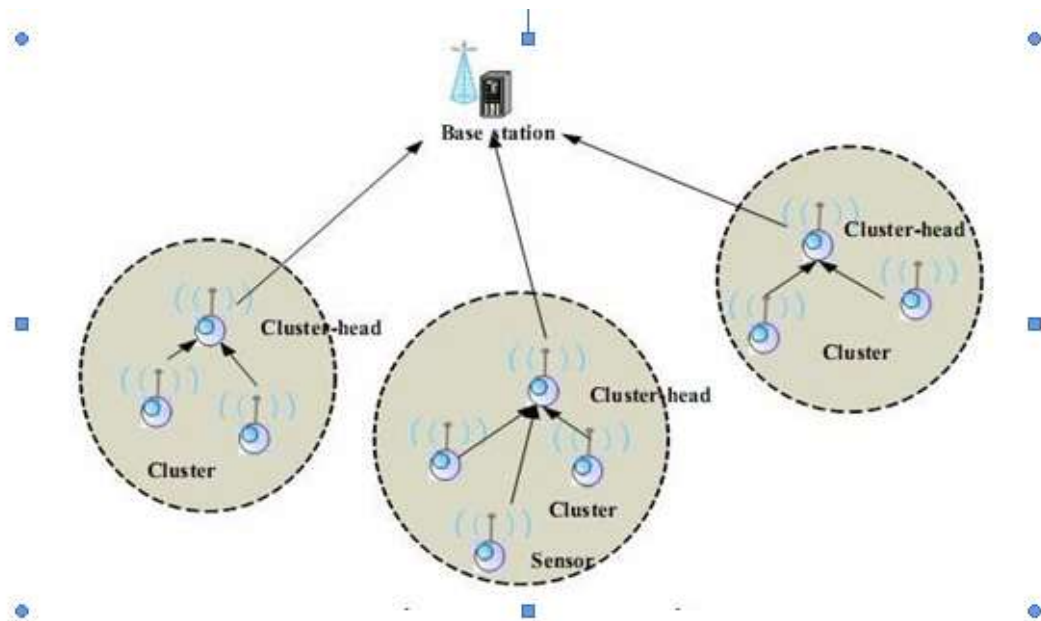
LEACH (Low Energy Adaptive Clustering Hierarchy) là giao thức phân cấp theo cụm thích ứng năng lượng thấp. Đây là giao thức thu lượm và phân phát dữ liệu tới các sink đặc biệt là các trạm cơ sở. Mục tiêu chính của LEACH là:

- Mở rộng thời gian sống của mạng
- Giảm sự tiêu thụ năng lượng bởi mỗi nút mạng
- Sử dụng tập trung dữ liệu để giảm bản tin truyền dẫn trong mạng

Để đạt được những mục tiêu này LEACH đã thông qua mô hình phân cấp để tổ chức mạng thành các cụm, mỗi cụm được quản lý bởi nút chủ. Nút chủ gánh lấy trọng trách thực hiện nhiều tác vụ. Đầu tiên là thu lượm dữ liệu theo chu kỳ từ các nút thành viên, trong quá trình tập trung dữ liệu nút chủ sẽ cố gắng tập hợp dữ liệu để giảm dư thừa về những dữ liệu tương quan nhau.

Nhiệm vụ thứ hai đó là nút chủ sẽ trực tiếp truyền dữ liệu đã được tập hợp lại đến các trạm cơ sở. Việc truyền này có thể thực hiện theo kiểu single hop. Nhiệm vụ thứ ba là LEACH sẽ tạo ra một mô hình ghép kênh theo thời gian TDMA, mỗi nút trong cụm sẽ được gán một khe thời gian mà có thể sử dụng để truyền tin.

Mô hình LEACH như hình vẽ (3.6). Các nút chủ sẽ quảng bá mô hình TDMA cho các nút thành viên trong cụm của nó. Để giảm thiểu khả năng xung đột giữa các nút cảm biến trong và ngoài cụm, LEACH sử dụng mô hình truy cập đa phân chia theo mã CDMA. Quá trình hoạt động của LEACH được chia thành hai pha là pha thiết lập và pha ổn định. Pha thiết lập bao gồm hai bước là lựa chọn nút chủ và thông tin về cụm. Pha ổn định trạng thái gồm thu lượm dữ liệu, tập trung dữ liệu và truyền dữ liệu đến các trạm cơ sở. Thời gian của bước ổn định kéo dài hơn so với thời gian của bước thiết lập để giảm thiểu mâu thuẫn



Hình 2.20 : Mô hình mạng LEACH

Trong bước thiết lập, một nút cảm biến lựa chọn một số ngẫu nhiên giữa 0 và 1.

Nếu số này nhỏ hơn ngưỡng $T(n)$ thì nút cảm biến là nút chủ. $T(n)$ được tính như sau:

$$T(n) = \frac{p}{1 - p * (r \bmod 1/p)} \text{ nếu } n \in G$$

$T(n)=0$ còn lại

Trong đó:

p : tỉ lệ phần trăm nút chủ

r : chu kì hiện tại

G : tập hợp các nút không được lựa chọn làm nút chủ trong $1/p$ chu kì cuối.

Sau khi được chọn làm nút chủ, các nút chủ sẽ quảng bá vai trò mới của chúng cho các nút còn lại trong mạng. Các nút còn lại trong mạng dựa vào bản tin đó và cường độ tín hiệu nhận được hoặc một số tiêu chuẩn nào đó để quyết định xem có tham gia vào cụm đó hay không. Và sau đó các nút này sẽ thông báo cho nút chủ biết là mình có mong muốn trở thành thành viên của cụm do nút chủ đó đảm nhận.

Trong quá trình tạo cụm các nút chủ sẽ tạo và phân phát mô hình TDMA cho các nút thành viên trong cụm. Mỗi nút chủ cũng chọn lựa một mã CDMA mà sau đó sẽ thông báo tới tất cả các thành viên trong cụm biết. Sau khi pha thiết lập hoàn thành báo hiệu sự bắt đầu của pha ổn định trạng thái và các nút trong cụm sẽ thu lượm dữ liệu và sử dụng các khe thời gian để truyền dữ liệu đến nút chủ. Dữ liệu được thu lượm theo chu kỳ.

Việc mô phỏng cho thấy LEACH tiết kiệm đáng kể năng lượng. Và sự tiết kiệm này phụ thuộc chủ yếu vào hệ số tập trung dữ liệu các nút chủ của cụm.

Tuy nhiên LEACH cũng có một số khuyết điểm sau:

Việc giả sử rằng tất cả các nút chủ trong mạng đều truyền đến trạm cơ sở thông qua một bước nhảy là không thực tế, và vì dự trữ năng lượng và khả năng của các nút hay đổi theo thời gian từ nút này đến nút khác. Hơn nữa khoảng chu kỳ ổn định trạng thái là vấn đề then chốt để đạt được giảm năng lượng cần thiết để bù đắp lượng mào đầu gây ra bởi xử lý lựa chọn cụm. Chu kỳ ngắn sẽ làm tăng lượng mào đầu, chu kỳ dài sẽ nhanh chóng làm tiêu hao năng lượng của nút chủ.

LEACH có đặc tính giúp tiết kiệm năng lượng, yêu cầu về năng lượng trong LEACH được phân bổ cho tất cả các nút trong mạng vì chúng ta giả sử rằng vai trò nút chủ được luân chuyển vòng tròn dựa trên năng lượng còn lại trên mỗi nút. LEACH là thuật toán phân tán hoàn toàn và không yêu cầu sự điều khiển bởi trạm cơ sở. Việc quản lý cụm là cục bộ và không cần sự hiểu biết về mạng toàn cục. Hơn nữa việc tập trung dữ liệu theo cụm cũng tiết kiệm năng lượng đáng kể vì các nút không yêu cầu gửi trực tiếp dữ liệu đến sink.

PEGASIS

PEGASIS (Power-Efficient Gathering in Sensor Information Systems), PEGASIS phân cấp là một họ các giao thức định tuyến và tập trung thông tin trong mạng cảm biến.

Giao thức này đầu tiên hỗ trợ việc kéo dài thời gian sống của mạng nhờ đạt được việc tiêu thụ năng lượng đồng nhất và hiệu suất năng lượng cao qua tất cả các nút trong mạng, thứ hai làm giảm trễ truyền dữ liệu đến sink.

Giao thức này xem xét mô hình mạng bao gồm tập hợp các nút đồng nhất được triển khai qua một vùng địa lý. Các nút này có sự hiểu biết về vị trí các nút khác trong toàn mạng và chúng còn có khả năng điều khiển công suất và bao phủ một vùng tùy ý. Các nút này cũng được trang bị bộ thu phát sóng hỗ trợ CDMA. Trách nhiệm của các nút này là thu lượm và truyền dữ liệu đến các sink, thông thường là các trạm cơ sở.

Mục đích để phát triển một cấu trúc định tuyến và một sơ đồ tập trung dữ liệu để giảm thiểu sự tiêu thụ công suất và truyền dữ liệu được tập trung đến trạm cơ sở với trễ truyền dẫn nhỏ nhất trong khi vẫn cân bằng sự tiêu thụ công suất giữa các nút trong mạng.

Giải thuật này sử dụng mô hình cấu trúc dạng chuỗi.

Dựa trên mô hình này các nút sẽ giao tiếp với nút hàng xóm gần nó nhất. Cấu trúc chuỗi bắt đầu với nút xa sink nhất, các nút mạng được thêm dần vào chuỗi làm chuỗi lớn dần lên, bắt đầu từ nút hàng xóm gần nút cuối nhất. Các nút sẽ được gán vào chuỗi theo cách greedy từ nút lân cận gần nhất cho tới các nút còn lại trong mạng. Để xác định được nút lân cận gần nhất mỗi nút sẽ sử dụng cường độ tín hiệu để đo khoảng cách tới các nút lân cận của nó. Sử dụng dữ kiện này các nút sẽ điều chỉnh cường độ tín hiệu sao cho chỉ có nút lân cận gần nhất nghe được.

Một nút trong chuỗi sẽ được chọn làm nút chủ, trách nhiệm của nút chủ là truyền dữ liệu tập hợp được tới trạm cơ sở. Vai trò nút chủ sẽ bị dịch chuyển vị trí trong chuỗi sau mỗi vòng chu kỳ. Chu kỳ này được quản lý bởi sink và việc chuyển trạng thái từ vòng này đến vòng tiếp theo có thể được khởi tạo bởi việc đưa ra dấu hiệu công suất cao bởi sink. Việc quay vòng nút chủ trong chuỗi nhằm đảm bảo công bằng trong tiêu thụ năng lượng giữa các nút trong mạng. Tuy nhiên cũng cần chú ý rằng việc thay đổi có khi dẫn đến nút chủ rời xa trạm cơ sở, sink, khi đó nút này lại cần yêu cầu công suất cao để truyền đến trạm cơ sở.

Việc tập trung dữ liệu trong mạng dọc theo chuỗi. Đầu tiên chain leader sẽ gửi một thẻ bài tới nút cuối cùng bên phải cuối chuỗi. Trong khi nhận được tín hiệu này nút cuối sẽ gửi dữ liệu nó thu lượm được đến nút lân cận theo chiều

xuôi trong chuỗi, sau đó nút này tập trung dữ liệu và lại tiếp tục gửi đến nút lân cận gần nó nhất, cứ như vậy cho đến khi gửi đến nút chủ. Sau đó nút chủ sẽ lại tập trung dữ liệu và gửi đến sink.

Mặc dù đơn giản nhưng mô hình tập trung dạng chuỗi dễ gây ra trễ trước khi dữ liệu tập trung được truyền đến sink. Một phương pháp để giảm độ trễ này là tập trung dữ liệu song song dọc theo chuỗi, và sẽ càng giảm nhiều hơn nếu các nút được trang bị bộ thu phát sử dụng CDMA.

Dùng PEGASIS sẽ giải quyết được vấn đề về mào đầu gây ra bởi việc hình thành các cụm động trong LEACH và giảm được số lần truyền và nhận bằng việc tập hợp dữ liệu. Tuy nhiên PEGASIS lại có độ trễ đường truyền lớn đối với các nút ở xa trong chuỗi. Hơn nữa ở nút chính có thể xảy ra hiện tượng thất cổ chai.

2.2.7. Giao thức dựa trên vị trí

Mục tiêu chính của giải thuật định tuyến này là dựa vào các thông tin về vị trí của các nút cảm biến để tìm một đường đi hiệu quả đến đích. Loại định tuyến này rất phù hợp với mạng cảm biến nơi mà việc tập trung dữ liệu là kỹ thuật hữu ích để giảm thiểu việc truyền bản tin đến trạm cơ sở bằng cách loại bỏ sự dư thừa giữa các gói đến từ các nguồn khác nhau. Loại định tuyến này còn yêu cầu sự tính toán và lượng mào đầu truyền dẫn thấp.

Ta sẽ xem xét một số giao thức định tuyến dựa trên vị trí như sau:

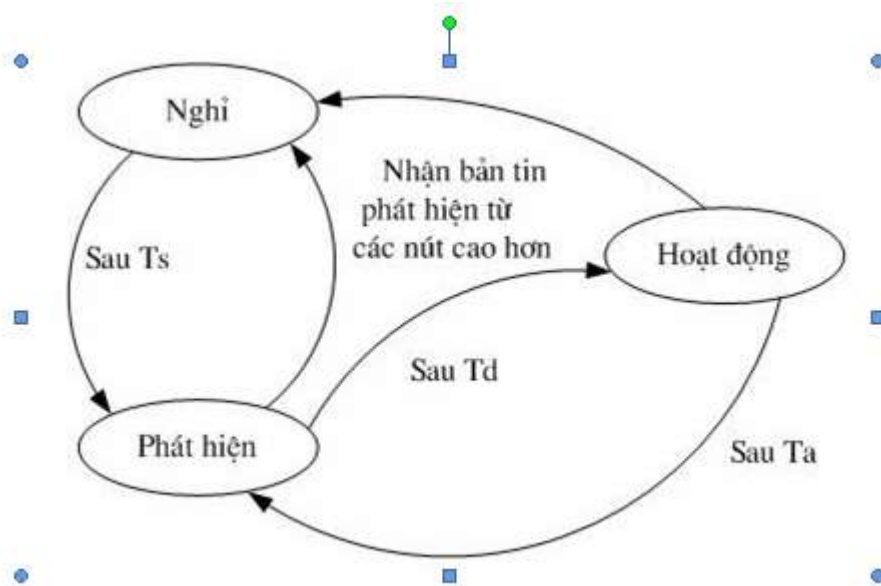
GAF

Giải thuật chính xác theo địa lý (GAF) dựa trên vị trí có hiệu quả về mặt năng lượng được thiết kế chủ yếu cho các mạng ad hoc di động, nhưng cũng có thể áp dụng cho mạng cảm biến. GAF khai thác việc dư thừa dữ liệu trong mạng bằng cách coi một tập hợp các nút con trong mạng là tương đương nhau khi nhìn từ giao thức lớp trên.

GAF chia vùng quan sát thành các hình vuông đủ nhỏ, bất kỳ các nút nào trong hình vuông cũng đều có thể giao tiếp vô tuyến với bất kỳ nút nào nằm trong hình vuông bên cạnh. GAF dự trữ năng lượng bằng cách tắt các nút không

cần thiết trong mạng mà không ảnh hưởng đến mức độ chính xác của định tuyến. Nó tạo ra một lưới ảo cho vùng bao phủ. Mỗi nút dùng GPS của nó – vị trí xác định để kết hợp với cùng một điểm trên lưới mà được coi là tương đương khi tính đến giá của việc định tuyến gói. Sự tương đương như vậy được tận dụng để giữ các nút định vị trong vùng lưới xác định trong trạng thái nghỉ để tiết kiệm năng lượng. Vì vậy GAF có thể tăng đáng kể thời gian sống của mạng cảm biến khi mà số lượng các nút tăng lên. Một ví dụ cụ thể được đưa ra ở hình (3.7). Trong hình vẽ này, nút 1 có thể truyền đến bất kì nút nào trong số các nút 2, 3 và 4 và các nút 2, 3, 4 có thể truyền tới nút 5. Do đó các nút 2, 3, và 4 là tương đương và 2 trong số 3 nút đó có thể ở trạng thái nghỉ.

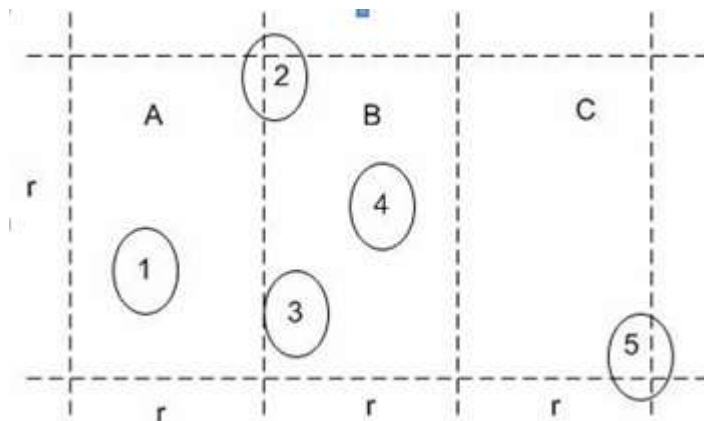
Các nút chuyển trạng thái từ nghỉ sang hoạt động lần lượt để cho các tải được cân bằng. Có ba trạng thái được định nghĩa trong GAF, đó là phát hiện (discovery), để xác định các nút lân cận trong lưới, hoạt động (active), thể hiện sự tham gia vào quá trình định tuyến và nghỉ (sleep) khi sóng được tắt đi. Sự chuyển trạng thái trong GAF được miêu tả ở hình (3.8). Nút nào nghỉ trong bao lâu liên quan đến các thông số được điều chỉnh trong quá trình định tuyến. Để điều khiển độ di động, mỗi nút trong lưới ước đoán thời gian rời khỏi lưới của nó và gửi thông tin này đến nút lân cận.



Hình 2.21: Sự chuyển trạng thái trong GAF

Các nút đang không hoạt động điều chỉnh thời gian nghỉ của chúng phù hợp các

thông tin nhận được từ các nút lân cận đó để giữ cho việc định tuyến được chính xác. Trước khi thời gian rời khỏi lưới của các nút đang hoạt động quá hạn, các nút đang nghỉ thoát khỏi trạng thái đó và một trong số các nút đó trở nên hoạt động. GAF được triển khai cho cả những mạng bao gồm các nút không di động (GAF cơ bản) và mạng bao gồm các nút di động (GAF thích ứng di động).



Hình 2.22: Ví dụ về lưới ảo trong GAF

GAF cố gắng giữ mạng hoạt động bằng cách giữ cho các nút đại diện luôn ở chế độ hoạt động trong mỗi vùng ở lưới ảo của nó. Các kết quả mô phỏng đã chỉ ra rằng GAF thực hiện tối thiểu sẽ được như giao thức định tuyến trong mạng ad hoc thông thường khi nói đến tổn thất gói và làm tăng thời gian sống của mạng bằng cách tiết kiệm năng lượng. Mặc dù GAF là một giao thức dựa trên vị trí, nó cũng có thể được coi là như một giao thức phân cấp khi mà các cụm dựa trên vị trí địa lý. Đối với mỗi vùng lưới xác định, mỗi nút đại diện hoạt động như một nút chủ để truyền dữ liệu đến các nút khác. Tuy nhiên nút chủ này không thực hiện bất cứ một nhiệm vụ hợp nhất hay tập trung dữ liệu nào như trong các giao thức phân cấp thông thường.

GEAR

Yu et al. đã đưa ra việc sử dụng thông tin về địa lý trong khi phổ biến các yêu cầu đến các vùng thích hợp vì các yêu cầu dữ liệu thường bao gồm các thuộc tính địa lý. Giao thức GEAR (Geographic and Energy-Aware Routing) dùng sự

nhận biết về năng lượng và các phương pháp thông báo thông tin về địa lý tới các nút lân cận. Việc định tuyến thông tin theo vùng địa lý rất có ích trong các hệ thống xác định vị trí, và đặc biệt là trong mạng cảm biến. Ý tưởng này hạn chế số lượng các yêu cầu ở Directed Diffusion bằng cách quan tâm đến một vùng xác định hơn là gửi các yêu cầu tới toàn mạng. GEAR cải tiến hơn Directed Diffusion ở điểm này và vì thế dự trữ được nhiều năng lượng hơn.

Trong giao thức GEAR, mỗi một nút giữ một estimated cost và một learned cost trong quá trình đến đích qua các nút lân cận. Estimated cost là sự kết hợp của năng lượng còn dư và khoảng cách đến đích. Learned cost là sự cải tiến của estimated cost giải thích cho việc định tuyến xung quanh các hốc trong mạng. Hốc xảy ra khi mà một nút không có bất kì một nút lân cận nào gần hơn so với vùng đích hơn là chính nó. Trong trường hợp không có một hốc nào thì estimated cost bằng với learned cost. Learned cost được truyền ngược lại 1 hop mỗi lần một gói đến đích làm cho việc thiết lập đường cho gói tiếp theo được điều chỉnh.

Có 2 pha trong giải thuật này

Chuyển tiếp gói đến vùng đích: GEAR dùng cách tự chọn nút lân cận dựa trên sự nhận biết về năng lượng và vị trí địa lý để định tuyến gói đến vùng đích.

Có 2 trường hợp cần quan tâm:

- +, Khi tồn tại nhiều hơn một nút lân cận gần hơn so với đích: GEAR sẽ chọn hop tiếp theo trong số tất cả các nút lân cận gần đích hơn.

- +, Khi mà tất cả các nút đều xa hơn: trong trường hợp này sẽ có một lỗ hổng. GEAR chọn hop tiếp theo mà làm tối thiểu giá chi phí của nút lân cận này. Trong trường hợp này, một trong số các nút lân cận được chọn để chuyển tiếp gói dựa trên learned cost. Lựa chọn này có thể được cập nhật sau theo sự hội tụ của learned cost trong suốt quá trình truyền gói.

Chuyển tiếp gói trong vùng: Nếu gói được chuyển đến vùng, nó có thể truyền dữ liệu trong vùng đó có thể bằng cách chuyển tiếp địa lý đệ quy hoặc flooding có giới hạn. Flooding có giới hạn áp dụng tốt trong trường hợp các sensor triển khai không dày đặc. Ở những mạng có mật độ sensor cao, flooding

địa lý đệ quy lại hiệu quả về mặt năng lượng hơn là flooding có giới hạn. Trong trường hợp đó, người ta chia vùng thành 4 vùng nhỏ và tạo ra 4 bản copy của gói đó. Việc chia nhỏ này và quá trình chuyển tiếp tiếp tục cho đến khi trong vùng chỉ còn 1 nút, ví dụ như hình (3.9).

Để thỏa mãn các điều kiện chúng ta dùng giải thuật chuyển tiếp địa lý đệ quy để truyền gói trong vùng này. Tuy nhiên, với những vùng mật độ thấp, chuyển tiếp địa lý đệ quy đôi khi không hoàn thành, định tuyến vô tác dụng trong một vùng đích rộng trước khi số hop gói đi qua vượt quá giới hạn. Trong trường hợp này chúng ta dùng flooding có giới hạn.

2.2.8. Kết luận

Chương này đã tổng kết và đưa ra khá nhiều các giao thức định tuyến. Mỗi giao thức đều có những ưu và nhược điểm riêng. Hiện nay, đã có rất nhiều các cải tiến của các loại giao thức này được đưa ra, và cho kết quả rất khả quan. Việc lựa chọn loại giao thức nào hoàn toàn phụ thuộc vào ứng dụng mà chúng ta triển khai. Mặc dù sự hoạt động của các giải thuật định tuyến này đầy hứa hẹn trong vấn đề sử dụng hiệu quả năng lượng, các nghiên cứu sau này cần phải xác định rõ các vấn đề như chất lượng dịch vụ của các ứng dụng của các cảm biến hình ảnh và các ứng dụng thời gian thực.

CHƯƠNG 3:

KẾT HỢP GIỮA MAC VÀ GIAO THỨC ĐỊNH TUYẾN MERLIN

3.1 Giới Thiệu

Việc kết hợp giao thức Mac và định tuyến nhằm kiểm soát truy cập và cung cấp dữ liệu đa chặng theo hai hướng chuyển tiếp đến gateway và ngược lại. Trước khi triển khai, thì việc phân tích hết sức quan trọng. Sau đó, mô tả giao thức MAC và định tuyến. Có hai vấn đề cơ bản cần quan tâm thông tin về tuyến (lập lịch, đồng bộ hóa thời gian) và duy trì tuyến.

Đặt vấn đề

Trước khi miêu tả các chi tiết của giao thức MERLIN, cần quan đến một số yếu tố triển khai trong mạng cảm biến. Nhất thiết phải có node sink (node cơ sở), gọi là Gateway, tại đó dữ liệu các cảm biến hội tụ về. Như vậy, gateway cần có khả năng xử lý cao hơn node cảm biến. Trường hợp nếu quy mô mạng lớn, thì cần phải có nhiều gateway.

Trong trường hợp nhiều gateway, giao thức kết hợp MERLIN đòi hỏi phải đồng bộ thời gian, có thể bởi phương tiện của một hệ thống truyền thông thay thế như qua vệ tinh, qua WLAN hoặc WMAN. Gateway đồng bộ cho phép quá trình thiết lập như mô tả trong mục 3.9. Các loại truyền thông trong giao thức kết hợp này bao gồm (1) truyền dữ từ các node về gateway gần nhất, (2) từ một gateway đến tất cả các node con hoặc một node một cụ thể nào đó, (3) truyền thông từ một node đến láng giềng.

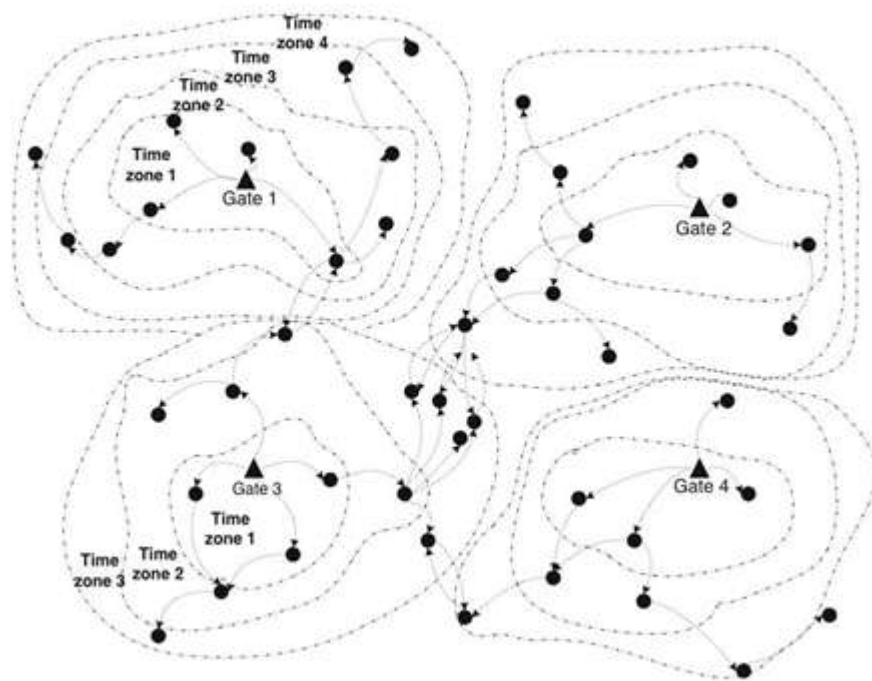
Tổng quan Merlin

Những điều cơ bản trong giao thức MERLIN là miền thời gian mà trong đó nó chia nhỏ vùng thời gian trong toàn mạng, như minh họa trong hình 3.1. Miền thời gian được giới thiệu trong pha thiết lập với thông điệp quảng bá SYNC từ gateway tới các nút láng giềng và ngược lại sẽ đồng bộ hóa đồng hồ nội bộ của chúng, sẽ thiết

lập miền thời gian tăng biến đếm miền thời gian cho gói dữ liệu SYNC trước khi chuyển tiếp nó tới các nút xung quanh. Khi kết thúc phải thiết lập tất cả các node sẽ được tổ chức thành các miền thời gian tương ứng.

Ngược lại với các chương trình trao đổi tin nhắn thông qua bởi hầu hết các giao thức mạng khác, các nút trong vòng MERLIN không đề cử một định tuyến chính cụ thể. Thay vào đó, giao thức MERLIN dụng phát đa truyền Upstream và phát đa truyền Downstream để chuyển tiếp thông tin đến gateway và đi ra từ nó. Trao đổi các gói tin đồng bộ hóa và cập nhật vùng thời gian thông qua việc phát quảng bá cục bộ theo pha đồng bộ burst ACK và NACK cho biết thành công và các lỗi.

Điều khiển đa luồng là sử dụng như một phương tiện dự phòng giảm gói rớt và nghe lỏm(overheard). Trong MERLIN, vị trí mỗi node trong bảng lập lịch mà quy định hoạt động định kỳ của nó.



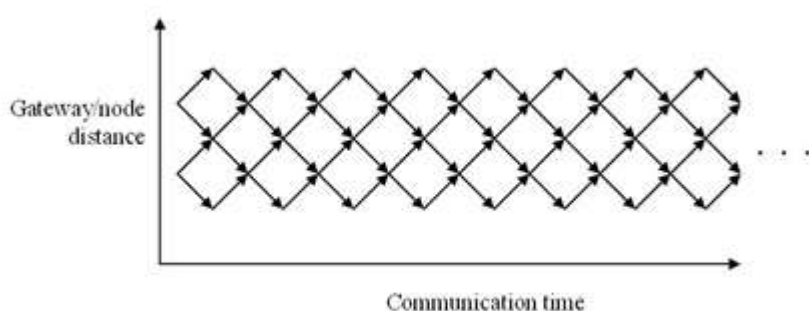
Hình 3.1 Bộ phận của mạng trong miền thời gian sau mạng đồng thời tràn bởi các cổng của một gói tin SYNC. Mỗi node bộ khu của nó và chuyển tiếp gói để thêm các nút xa

3.2 Thiết kế

Truyền dữ liệu

MERLIN sử dụng cách tiếp cận phát đa truyền trong giao tiếp các gói tin đến biên của miền thời gian. Theo tập mẫu truyền được sinh ra từ các ứng dụng, MERLIN giải quyết việc truyền giữa các node với các gateway và ngược lại. Đặc biệt MERLIN hỗ trợ hai loại phát đa truyền (1) phát đa truyền Upstream đến gateway; (2) phát đa truyền Downstream từ gateway. Hơn nữa, MERLIN cung cấp phát quảng bá cục bộ để truyền dữ liệu, thông tin giữa các node láng giềng (ví dụ như chỉ số RSSI, bảng của láng giềng, pin cấp, quảng bá thông tin, vv.)

Trong trường hợp phát đa truyền Downstream và phát quảng bá cục bộ, được dành cho tất cả các nút, lưu ý nếu có ít nhất một gói tin truyền không chính xác thì dẫn đến việc tự sửa lỗi đơn bit hoặc lập kế hoạch truyền lại.



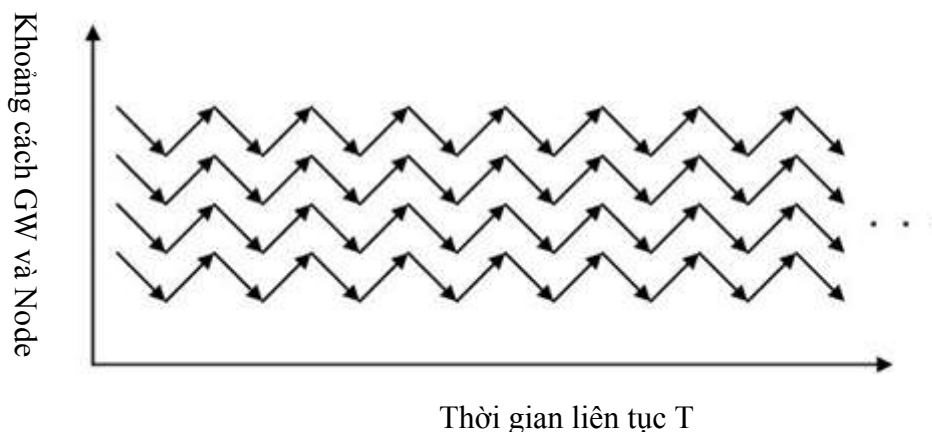
Hình 3.2 Mô hình dữ liệu hai chiều liên tục các mẫu của nút / gateway truyền thông trong các ứng dụng của giám sát môi trường và công nghiệp dựa trên WSN

Lập lịch

Mục đích của bảng lập lịch là để cho phép các khe thời gian phân bổ cho các nút mạng để giao các thời kỳ của các node hoạt động và không hoạt động. Lập kế hoạch cho phép đồng bộ hóa của các nút láng giềng để truyền và nhận. Các ứng dụng theo dõi công nghiệp và trong môi trường. Trong mục này nhấn mạnh đến các chu kỳ lập lịch hoặc sự kiện dựa trên hai hướng truyền dữ liệu Upstream và truyền dữ liệu Downstream. Mô hình dữ liệu tiêu biểu có thể được phân loại liên tục và không liên tục.

Mô hình dữ liệu liên tục có thể chuyển tiếp một gói tin mà không bị gián đoạn từ nguồn tới đích còn mô hình dữ liệu không liên tục có một số độ trễ trong quá trình chuyển tiếp. Độ trễ này có thể sử dụng để tránh va chạm hoặc tiết kiệm năng lượng. Trong hình 5.5 các mũi tên truyền lại nguồn tới đích. Gói tin được truyền ở mô hình dữ liệu liên tục có thể đạt đến đích mà không bị gián đoạn.

VD: Một nút nhận được một gói tin và chuyển tiếp nó đi ngay lập tức



Hình 3.3 Mô hình dữ liệu không liên tục hai hướng mô hình của nodet /gateway trong các ứng dụng theo dõi môi trường và công nghiệp dựa trên WSN.

Hình 3.3 Trình bày mô hình dữ liệu không liên tục, mà trong đó một gói tin được chuyển tiếp đi sau độ trễ nhất cho đến đích. Chính sách lập lịch được phát triển theo như dự kiến mô hình truyền. MERLIN sử dụng lập lịch truyền dựa trên miền thời gian do đó các nút trong cùng một miền thời gian sử dụng cùng một khe cắm để truyền.

Bảng lập lịch luôn được truyền bởi gateway trong pha khởi động, cho phép lập lịch các nút hoạt động theo chu kỳ. Hình 3.4 và 3.5 phân biệt 2 Bảng lập lịch quan sát trong MELIN thấy rằng bảng X và bảng V. Đây là những bảng lấy ra trực tiếp từ mẫu hình trong hình 3.2 và 3.3. Trong thực tế, mẫu node / gateway được nêu ở trên có thể xây dựng được những bảng phụ thêm cho khung thời gian mà miền thời gian về sau. Trong mỗi lớp, khác nhau về chiều dài khe hoặc chiều dài khung nguyên nhân chậm trễ khác nhau trong gói lịch. Tổng chiều dài của một bảng bằng chiều dài của một khung duy nhất trong khi mỗi nhỏ hình chữ nhật đại diện cho một khe thời gian. Trong bảng V, truyền Upstream và Downstream là xen kẽ còn trong bảng X truyền Upstream và Downstream là kết hợp. Trong cả hai bảng các nút trong cùng một miền thời gian chạy theo kênh để truyền, các khu vực lân cận sở hữu

khe cấm để tiếp nhận và nút trong miền thời gian đều ở chế độ ngủ.

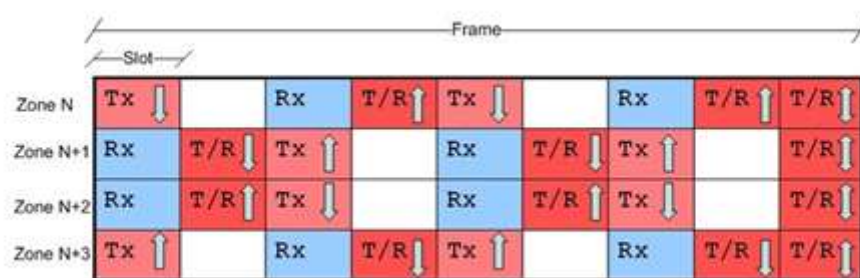
Bảng X và bảng V trình bày hai mô hình lập lịch cung cấp các ưu tiên khác nhau cho việc tiêu thụ năng lượng và độ trễ của gói tin. Thực chất nó là một mảng 2 chiều (ma trận). Vì vậy khe thời gian của node trong ma trận tại thời gian t được xác định bởi các hàng và cột như sau:

$$\text{Row} = [\text{Mod}(\text{myZone}, \text{NZONE})]$$

$$\text{Column} = [\text{Mod}((t/\text{SLOTTIME}), \text{NSLOT})]$$

myZone là miền thời gian của node, $\text{NZONE} = 4$ là số miền trong bảng, $\text{NSLOT} = 9$ là số khe trong bảng và SLOTTIME là độ dài của một khe. Nói cách khác, việc lập lịch của những miền thời gian sau này được lấy ra từ 1 bảng phụ tương tự. Việc lập lịch cho các khung được lấy ra từ bảng phụ. Như vậy lập lịch cho khung được lấy ra từ thủ tục vòng lặp trên cùng một bảng. Sau đây là lý do đằng sau hai phương pháp tiếp cận lập kế hoạch được thông qua:

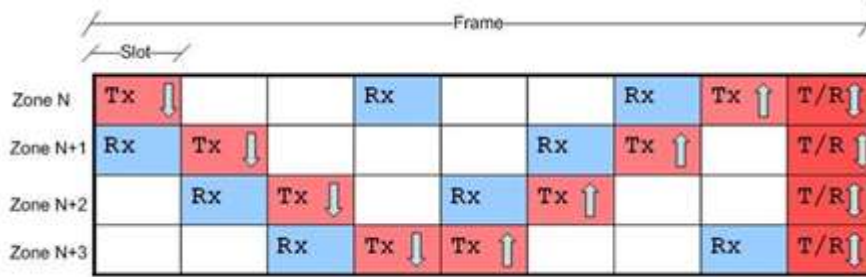
Trong lập lịch V, Downstream và Upstream truyền dữ liệu không giao nhau và có khe cấm thêm các nút không hoạt động hơn trong bảng X. 4 vùng trong bảng V có khả năng cho phép truyền song song giữa các nút được xác định ở 4 vùng khác nhau. Vì vậy sẽ tốt hơn trong việc tránh va chạm và tiêu thụ năng lượng hơn so với bảng X. Bảng V thực hiện Upstream, Downstream truyền không giao nhau bằng cách chuyển tiếp một gói tin tới 4 miền thời gian hướng theo bộ phối hợp PAN. Nói chung, việc phân bổ Upstream, Downstream, và phát quảng bá cục bộ trong hệ thống mạng có N miền thì yêu cầu có $N \times 2 + 1$ khe thời gian trên một khung. Bảng V thực hiện truyền Upstream và Downstream nhanh bằng cách chuyển tiếp một gói tin tới bốn miền thời gian đối với các điều phối PAN hoặc theo hướng ngược lại. Bảng lập lịch phụ cho việc truyền miền cho thấy tăng của số lượng các miền một bảng sẽ dẫn đến việc giảm thời gian song song. Vì vậy một bảng có ít miền thời gian hơn sử dụng hiệu quả đường truyền hơn. Tuy nhiên, một bảng với chỉ 2 khu, và 5 khe thời gian, sẽ gây ra va chạm gói tin liên tục khi bảng được thêm vào. Hơn nữa, như là kết quả mô phỏng có hiển thị, một bảng có ba miền thời gian sẽ vẫn sinh ra nhiều xung đột tại miền giữa việc truyền song song khi được gán thêm bảng, xung đột gây ra bởi vị trí của nút ngẫu nhiên vào khoảng truyền bất quy tắc.



Hình 3.4 Bảng lập lịch X với danh thời gian trong một khe cho đến chu kỳ phát quảng bá cục bộ.

Trong lịch-X, trọng tâm là về việc sử dụng kênh cao hơn, do đó các nút có nhiều hoạt động, và chúng giữ khe truyền đồng thời giữa miền thời gian gần . Như trong bảng V, lên lịch các khu tiếp thu được bằng cách phụ thêm cùng một bảng. Kết quả là, các bảng X thực hiện Upstream và Downstream đồng thời truyền bằng một gói tin tới 8 miền thời gian chuyển tiếp về (gateway) hoặc chiều ngược lại trong cùng một khung thời gian. Nói cách khác, bảng X hỗ trợ chồng chéo khu liên lạc để giao 8 khe thời gian cho Upstream, 8 khe thời gian cho Downstream và 1 khe thời gian cho truyền quảng bá. Nói chung, việc phân bổ phát sóng truyền Downstream và Upstream và phát quảng bá cục bộ trong một mạng lưới đối xứng của các khu N yêu cầu $N + 1$ khe thời gian cho mỗi khung. Trái ngược với bảng V, bảng X trình bày một chu kỳ nhiệm vụ cao hơn và do đó thích hợp hơn cho các ứng dụng với dữ liệu tốc độ cao. Tuy nhiên, bảng X có thể gây va chạm nhiều hơn hoặc tại trì hoãn việc sau đó truyền đến khe do thất bại trong việc đánh giá kênh chống lại khu thời gian liền kề. Mặt khác, bảng X có thể dẫn đến hiệu suất tốt hơn so với bảng V về độ trễ của gói tin và thông lượng cho các kịch bản nhất định.

Cột cuối cùng của khe thời gian trong bảng V và bảng X là dành riêng cho truyền quảng bá các gói tin. Tuy nhiên, việc phát quảng bá đồng thời các nút lân cận hoặc trong miền thời gian hoặc 2 miền thời gian xa nhau kết quả trong va chạm gói.



Hình 3.5: Bảng lập lịch V với danh thời gian trong một khe cho đến chu kỳ phát quảng bá cục bộ..

Việc phát quảng bá đồng thời có thể có ít nhất 3 miền ngoài (chương trình phát quảng bá tại khu 1 và tiếp theo là khu 4, 2 và 5 và vv). Để tiếp tục đảm bảo tránh va chạm giữa các vùng, các bảng V đồng thời triển khai phát quảng bá cục tách 4 miền thời gian. Nodes được phép phát quảng bá nếu áp dụng công thức sau:

$$\text{Mod}(f \text{ Ramen}, \text{NZONE}) == \text{Mod}(\text{myZone}, \text{NZONE})$$

f Ramen là nơi khung truy cập. Do đó, các nút trong cùng miền thời gian có thể cạnh tranh khe cấm cho truyền quảng bá chỉ một lần mỗi 4 khung hình. Khi một nút cạnh tranh thành công các kênh quảng bá cho, các nút trong cùng vùng và lân cận những nút lân cận vào chế độ lắng nghe. Lưu ý rằng mặc dù CC2420 cho 802.15.4 không hỗ trợ việc phát đa truyền, MERLIN vẫn có thể thực hiện nó bằng cách phát sóng Downstream và Upstream, trong khi các nút hoặc trong các miền thời gian liên kế cao hơn hoặc thấp hơn là trong chế độ ngủ tương ứng. Điều này tương đương với việc phát đa truyền thực hiện để các nút gần hoặc xa hơn từ điều phối viên PAN.

Định dạng gói

Trong MERLIN gói một đại diện cho một tập hợp các thông điệp được lắp ráp khi nút được yêu cầu để chuyển tiếp một số tin nhắn nhận được từ các nút khác. Một nút tạo ra một tin nhắn có chứa thông tin giao thức như sourceID, destID, forwardID, e orwardZone, msgstr, msgType, và, tất nhiên, các DATApayload. Tất cả các thông điệp được xác định duy nhất bởi các msgstr, mà là một sự kết hợp của sourceID.

Trong quá trình chuyển tiếp đến gateway, một nút có thể nhận được gói tin từ một số nút đó sẽ được chuyển tiếp đến các nút trên miền thời gian thấp hơn. Kể từ khi dữ liệu cảm biến là thường chỉ có vài bytes, ví dụ, nhiệt độ, áp suất, hoá dữ liệu và như vậy MERLIN tiếp xúc các tin nhắn và sau đó truyền cho chúng như là một gói duy nhất.

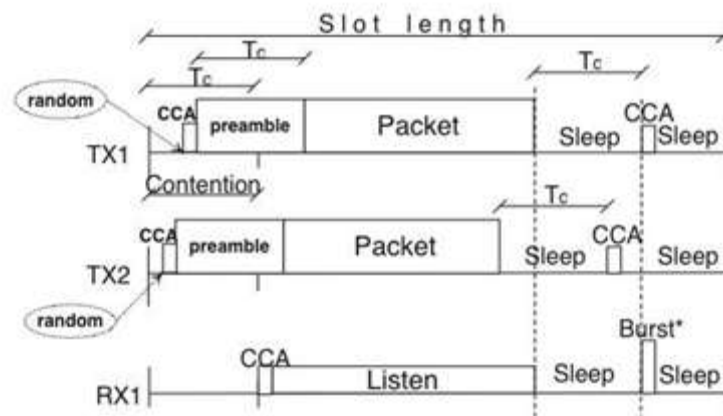
Do đó, một nút có nhiều hơn một tin nhắn để gửi, tập hợp và gửi tin nhắn trong một gói dữ liệu trong cùng một khe thời gian, mà cả hai tiết kiệm năng lượng. Một số yếu tố như loại ứng dụng, tỷ lệ thu phát dữ liệu, và kênh điều kiện chi thị số lượng tối đa của tin nhắn trong một gói. Ví dụ, gói kích thước nhỏ được thuận lợi trong các kênh tổn hao không dây. Trong tiêu chuẩn MERLIN cấu hình, kích thước gói tin không vượt quá 100 byte (tối đa chiều dài gói tin) mặc dù các mô phỏng xem xét một phạm vi lớn hơn kích thước gói tin đã được xem xét.

Để tạo thuận lợi cho quá trình nối, MERLIN cung cấp ba bộ đệm nhỏ cho truyền quảng bá, phát đa truyền Upstream, Downstream. Vùng đệm này được sử dụng để tạm thời lưu trữ tin nhắn của riêng của một node cũng như các tin nhắn nhận được từ khác nút đó sẽ được chuyển tiếp. Các quá trình truyền dẫn cho các loại gói tin khác nhau được quy định trong bảng lịch trình. Trong một truyền node, các nút trong khu vực lân cận hoặc là thời gian dự kiến tiếp nhận hoặc chúng đang ngủ. Vì vậy, chỉ có các nút lân cận trong cùng một thời gian phát của khu vực có thể nghe trộm gói tin được gửi đi. Nếu một nút nghe lén sau đó nhận được một gói tin, nó sẽ so sánh từng msgstr chứa trong các gói tin với tất cả các msgIDs trong bộ đệm thích hợp của nó. Bản thông điệp này sau đó đã bị xóa từ đệm như đã được xử lý bởi một nút lân cận. Cuối cùng, các gói tin để truyền được hình thành bởi tin nhắn theo nguyên tắc FIFO.

Cơ chế truyền động

Nodes nhận: MERLIN thông qua kênh đánh giá rõ ràng (CCA) cơ chế thông qua Low Power Listening (LPL), một cách tiếp cận đó là hiệu quả sử dụng trong BMAC và triển khai trong các hệ điều hành TinyOS. Với kỹ thuật LPL, các raido đánh thức và kênh truyền mẫu cho một khoảng thời gian ngắn 4ms gọi là khoảng thời gian CCA

Trong MERLIN, Node được lập lịch sẽ tiếp nhận dữ liệu kích hoạt radio của chúng trong một thời gian CCA để nghe kênh. Nếu không phát hiện được kênh có hoạt động thì nút đi vào chế độ ngủ và ngược lại nó nhận được gói tin truyền. Việc áp dụng của cơ chế CCA làm giảm đáng kể hiện tượng nghe ngửi ở bên nhận.



Hình 3.6: Cơ chế truyền động tránh va chạm của MERLIN

Nodes truyền: Một nút mà muốn truyền những lựa chọn khởi tạo ở thời gian ngẫu nhiên trong khoảng thời gian cạnh tranh và thức dậy tại thời gian đó nghe kênh lúc CCA. Nếu không tìm thấy thì kênh được xem là rỗi và nó truyền ngay gói preamble, trong đó thời gian T_c bằng tổng chiều dài CP. Điều này đảm bảo rằng việc truyền preamble đạt đến đích CP. Gói tin được truyền ngay sau preamble.

Việc khởi động thời gian ngẫu nhiên cho phép truyền không đồng bộ tới bên nhận thông qua gói burstACK, gọi là BACK. Điều này nhấn mạnh một thực tế là gói tin vỡ không thể mang mã thông tin còn các gói tin vỡ chồng lên nhau được xác định ở gói nhận như là một cụm duy nhất. Đối với phát đa truyền Upstream thì gói tin vỡ burstACK sẽ thông báo cho bên nhận tiếp nhận chính xác của ít nhất một nút trong miền gần gateway hơn. Trong thực tế, khi đang phát đa truyền Upstream thì bộ truyền không cần biết tất cả các nút tham gia trong quá trình chuyển tiếp. Thay vào đó, các thông báo xác nhận ít nhất một gói tin đã được chuyển tiếp. Bảng lập lịch phải đảm bảo các gói dữ liệu có được truyền đi theo hướng đúng. Trong trường hợp phát đa truyền Upstream chỉ có các nút trong miền thời gian lắng giềng gần gateway bật chế độ nghe cho cơ chế làm việc. Điều này không áp dụng cho cả việc phát quảng bá cục bộ và phát đa truyền Downstream các gói tin được thực hiện thông tin cho tất cả các node lắng giềng hoặc từ gateway vào mạng tương ứng.

Trong trường hợp phát đa truyền Downstream, phát quảng bá cục bộ, trong đó xảy ra trong khe cấm dành riêng trong bảng lập lịch. Điều này được gọi là burstACK âm (BNACK). Bởi vì chương trình phát quảng bá được dành cho tất cả các nút lân

cận, truyền các gói tin trong trường hợp nhận được một BNACK. Một thiếu sót của cơ chế BNACK là một nút đó là giấy lát có thể một truyền gói tin không được nhận thức của do đó nó không gửi lại một BNACK. Vấn đề làm giảm nhẹ bằng cách xác định các gói tin thông qua một sự kết hợp của sourceID, là ID của nút đó lắp ráp các gói tin, và một packetID, mà là một lưu trữ truy cập tại mỗi nút và nó được tăng lên mỗi lần một gói tin mới được. SourceID và packetID đều có trong mỗi gói tin. Nếu thêm gói tin truyền từ cùng một nút xảy ra, người nhận có thể xác định các gói tin không phù hợp mất tích của một trong số packetID gia tăng, và sau đó yêu cầu một bản sao khác của nó. Nên gateway nhận được hai gói tin liên tiếp từ cùng một nguồn với packetID bằng n và $n + 2$, nút là có thể xác định các gói tin bị mất $n + 1$ và yêu cầu của nó truyền lại. Đối với các nút này nguồn lý do nên giữ một bản sao lưu của vài cuối tạo ra các gói tin. Tuy nhiên, thủ tục thu hồi gói mô tả các kết quả trong một tăng chậm trễ truyền dẫn và số lượng sao lưu gói tin bị giới hạn bởi bộ nhớ hạn chế của thiết bị.

Trong bản chất, không có sự khác biệt trong các tín hiệu được truyền bởi các t burst BACK và BNACK có thể chỉ cần xác định bằng phương tiện của khe trong đó xuyên. Nhiệm vụ xảy ra. Lưng được truyền đi trong cả hai khe cấm dành riêng cho phát đa truyền Upstream trong khi BNACKs được truyền đi trong phát đa truyền Downstream và phát quảng bá cục bộ. Trong cả hai phát đa truyền và phát quảng bá cục bộ nếu một lỗi được phát hiện, các gói tin sẽ được đòi lại sau khi một số mũ-off ngẫu nhiên trở lại thủ tục. Những burst thường cung cấp trong radio cảm biến chuẩn stack. Để không làm ảnh hưởng đến truyền liên tục khác, việc truyền tải burst bị trì hoãn một cách chính xác T_c sau khi kết thúc gói tối đa chiều dài cho phép. Một hệ quả chính của việc này là việc loại bỏ khả năng của vô tình làm hư truyền đi gần đó. Ngoài ra, việc truyền tải burst là không đồng bộ, đảm bảo rằng các máy phát có thể trở lại trạng thái ngủ sau khi truyền tải hoàn tất, và có thể thức tỉnh sau khi T_c để phát hiện hoặc một BACK hay BNACK để đáp ứng với gói tin truyền qua đường của nó.

3.3 Thiết Lập

Nodes bắt đầu giai đoạn pha khởi động mạng bằng cách lắng nghe cho một gói SYNC, chứa thông tin thời gian, ID của người gửi và miền thời gian của người gửi. Gateways, được đồng bộ hóa tới miền thời gian tham khảo như nhau, bắt đầu khởi tạo mạng bằng việc phát quảng bá các gói tin SYNC. Gateways thiết lập miền thời gian của mình là 0. các nút cảm biến trong vùng lân cận của gateway nhận được gói SYNC dùng nó để đồng bộ hóa đồng hồ nội bộ của chúng. Khi chúng ở ngoài 1 hop của gateway thì các nút thiết lập giá trị của miền thời gian là 1. Theo miêu tả trong Hình 3.1 miền thời gian node 1 sẽ chuyển tiếp gói SYNC tới các nút xa hơn. Cơ chế truyền tránh va chạm được mô tả trong phần 3.8 để đảm bảo hoạt động chuyển tiếp đúng. Tất cả các nút nhận được gói tin từ SYNC 1 nút trong múi giờ của mình để thiết lập miền thời gian 2. Thủ tục này được lặp đi lặp lại cho đến khi tất cả các nút đã thiết lập miền thời gian của chúng. Trong trường hợp nhiều cổng, các cổng bắt đầu loang mạng cùng một lúc bằng cách gửi một gói tin SYNC đến các nút lân cận. Trên SYNC nhận các gói tin từ hai cổng khác nhau, các nút có thể tính toán của họ thời gian khu để mỗi cổng và chọn miền thời gian của nó đối với các gateway gần nhất.

Kết thúc pha khởi động, số miền thời gian của một nút bằng với số lượng tối thiểu hops một gói tin cần phải đạt được các gateway gần nhất. Trong pha khởi động, có thể một nút được phân lớp trong miền thời gian cao hơn. Ví dụ, có một nút trong miền thời gian lỗi N truy cập vào các kênh, hoặc gói tin va chạm thì nó phải áp dụng lại thủ tục cạnh tranh cho node để khung chuyển kế tiếp kênh thành công. Trong khi đó, bất kỳ nút nào chờ đợi trong pha khởi động có thể nhận được một gói SYNC từ bên thứ ba, việc này gây chọn một miền thời gian cao hơn. Như miền thời gian cao thoáng qua không ngăn cản các nút từ giao tiếp với các nút lân cận, mặc dù điều này có thể dẫn đến một con đường dài hơn để gateway. Trong trường hợp của các nút được cách đều từ hai cổng nút sẽ lựa chọn một múi giờ theo đầu tiên SYNC nhận được. Pha khởi động thời gian tổng số phụ thuộc vào Bảng lập lịch sử dụng. Đặc biệt, trước đây các nghiên cứu về hiệu suất của các lịch trình của MERLIN

chứng minh rằng bảng V pha khởi động của 14 miền thời gian trong một mạng lưới trong khoảng 9 giây.

Đồng bộ hóa

Sự tinh vi phần cứng nghèo như độ chính xác đồng hồ hạn chế các nút cảm biến đòi hỏi phải đồng bộ hóa gateway lặp đi lặp lại. Ngoài việc đồng bộ hóa bắt đầu thời điểm (hiệu số) của các khe giữa các nút, thu liên tục phải bồi thường các skew tần số đồng hồ cá nhân của mỗi nút. Cả hai vấn đề được giải quyết bằng cách bao gồm thời gian trong mỗi gói tin truyền đi. Tất cả các nút nhận sau đó có thể ước tính khi bắt đầu khe theo người gửi, và đồng bộ hóa đồng hồ của mình bù đắp. Hơn nữa, một nút có thể tính toán nghiêng đồng hồ của mình bằng cách so sánh theo thời gian các quan sát sự khác biệt giữa một chiếc đồng hồ của người gửi và đồng hồ của riêng mình. Trong MERLIN, các nút cập nhật của họ đồng hồ đồng bộ hóa từ các nút thuộc miền thời gian thấp hơn được gần gũi hơn với gateway. Trong thực tế, các nút như tổ chức một đồng bộ hóa tốt hơn, thông qua lý luận tương tự để các tầng được sử dụng trong NTP.

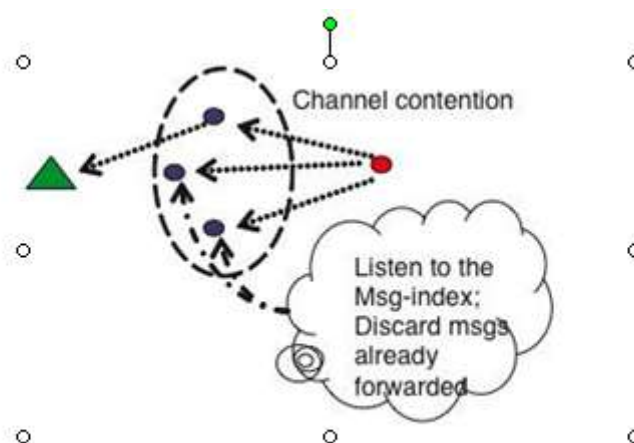
Một node mới có thể tham gia mạng bằng cách đơn giản lắng nghe gói tin bất kỳ. Các gói tin chứa thông tin, chẳng hạn như người gửi miền thời gian, thời gian truyền gói dữ liệu cho phép các node mới ước tính miền thời gian và đồng bộ hóa của nó bằng cách điều chỉnh đồng hồ của mình bù đắp theo thời gian nhận được node sau đó ở một vị trí để gia nhập mạng và làm theo các hoạt động theo bảng lập lịch đã định. Mạng này đơn giản thủ tục gia nhập làm tăng khả năng mở rộng của giao thức để đối phó với các mạng có quy mô lớn. Để ngăn chặn sự thiếu chính xác do nghiêng đồng hồ biến động, MERLIN sử dụng hai phương pháp: (1) Mỗi nút truyền các gói tin đồng bộ hóa với các miền định kỳ với một tốc độ dữ liệu thấp; (2) Tiếp nhận các nút 30 μ sec thức dậy sớm hơn thời gian dự kiến là nó đã được thực nghiệm chứng minh giải quyết phù hợp đồng hồ nghiêng gây ra bằng cách sử dụng đồng bộ hóa thời gian .

Nếu một gateway mới gia nhập mạng, nó lần đầu tiên tham gia như là một nút mới. Sau đó, nó thông báo sự hiện diện của nó thông qua việc quảng bá của một gói tin SYNC. Nodes so sánh thời gian cũ của nó. Miền với các số mới và chuyển tiếp các gói tin SYNC. Trong trường hợp gần gateway mới, nút một thay đổi múi giờ của nó tại đầu của khung kế tiếp chỉ sau khi việc truyền tải các SYNC để có được thành

công. Điều này tránh tạm thời mạng lưới gián đoạn gây ra bởi các nút lân cận đồng bộ khác nhau.

Đặc điểm định tuyến

Việc phân chia trong múi giờ cùng với lịch trình và loại hình truyền dữ liệu cho phép gói tin được chuyển đến và đi từ đến gateway gần nhất. Nhớ lại rằng MERLIN không không cụ thể địa chỉ một nút chuyển tiếp. Điều này có thể gây ra trùng lặp của các gói tin trong chuyển tiếp hoạt động. Tuy nhiên, thế hệ gói được điều khiển thông qua một cơ chế nghe lỏm ,lắng nghe trên các tin nhắn được truyền qua một node lắng giềng để xác định chúng và sau đó xóa các bản sao các gói tin từ riêng .



Hình 3.7: Cơ chế nghe lỏm đa đường kiểm soát thông qua các nút lân cận
Điều khiển đa luồng

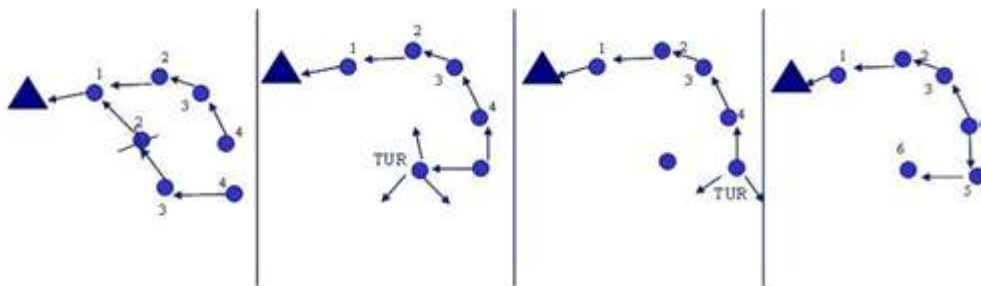
Thông điệp có thể được nối để tạo thành một gói dữ liệu. Khi một gói được thành lập, một chỉ số msg được tạo ra. Chỉ số msg có tất cả msgIDs thu trong gói tin. Chỉ số msg, hoặc các msgstr trong trường hợp của một tin nhắn đơn, là nằm ở đầu của gói tin, do đó cho phép các nút lân cận của cùng một miền thời gian chọt nghe tin nhắn được truyền đi. Một phân cách khung bắt đầu có thể cho phép xác định sự khởi đầu của chỉ số msg. Về các tin nhắn xác định trong bộ đệm của nó mà đã được truyền bởi một nút lân cận trong vùng Đồng thời, một nút ngay lập tức xóa các tin nhắn từ bộ đệm như trong hình 3.7. Điều này cơ chế được sử dụng cho truyền thông multicast Upstream như nó chỉ là cần thiết mà một trong những dự của thông điệp đến gateway. Downstream truyền không áp dụng cơ chế này, như là xóa một tin gây

ra một số các nút không nhận được thông tin liên lạc từ các gateway, chẳng hạn như cập nhật định kỳ mạng

Duy trì miền thời gian

Nếu một nút trong khu N không nhận được bản cập nhật định kỳ miền thời gian từ bất kỳ nút lân cận trong khu vực N-1, nó phát đi một yêu cầu cập nhật miền thời gian (TUR) thông qua phát đa truyền thượng nguồn như trong hình 5,8. Trong trường hợp trả lời không, các nút sẽ cho rằng sự kết nối với khu N-1 đã thất bại. Node sau đó cố gắng thiết lập lại kết nối với bất kỳ node trong cùng một miền thời gian qua việc quảng bá của TUR. Nếu node nhận được ít nhất một burstACK, nó thay đổi miền thời gian của mình cho N +1. Nếu không, nút các giả định một kết nối không thành công cũng với các nút trong cùng một khu. Kết quả là, nó sẽ cố gắng thiết lập lại một kết nối thông qua hop phát đa truyền Downstream của TUR. Ở giai đoạn này, một nhận burstACK một phương tiện một xác định các kết nối với các nút trong khu N 1 mà sẽ đòi hỏi một sự thay đổi của miền thời gian để N 2. Trong trường hợp không có sự thừa nhận từ khu N +1, nút tạm thời bỏ qua lập kế hoạch và bắt đầu nghe kênh cho đến khi nó phát hiện bất kỳ gói tin từ các node láng giềng.

Các miền thời gian chứa trong các gói tin sẽ cho phép các node để thu được một thời gian mới khu vực, thiết lập lại kết nối, và thực hiện theo lập lịch một lần nữa. Đối với tất cả các trường hợp, các nút lặp đi lặp lại một truyền TUR hai lần trước khi giả định kết nối tuyến.



Hình 3.8 :Khu thời gian bảo trì cơ chế MERLIN

3.4 Kết Luận

Phần này tập trung vào thiết kế giao thức MERLIN tích hợp MAC và chức năng định tuyến vào trong cùng một kiến trúc. MERLIN được tối ưu hóa cho giao tiếp giữa các nút và các gateway, hỗ trợ phía Upstream , Downstream và phát quảng bá cục bộ trong việc truyền dữ liệu. Các gói tin lưu chuyển là đạt được thông qua một bộ phận của mạng trong miền thời gian và thông qua việc sử dụng lập lịch truyền thích hợp. Trong đó, giao thức đề xuất 2 bảng lịch trình, cụ thể là bảng V và bảng X, để sử dụng trong mạng khác nhau.

Việc tích hợp loang theo kiểu đồng bộ hóa thủ tục cho phép thiết lập miền thời gian và đồng bộ hóa đồng hồ của mỗi nút. Truyền trong MERLIN không địa chỉ một nút cụ thể trong khi chuyển tiếp được truyền đi thành công được xác định thông qua gói ACK và gói NACK

Việc điều khiển đa luồng làm giảm đáng kể sao chép một gói tin trong cơ chế duy trì miền thời gian đảm bảo kết nối giữa các nút. Trước khi đánh giá hiệu năng của MERLIN chống lại cách tiếp cận khác, sau đây là một nghiên cứu sơ bộ về tiêu thụ năng lượng trong các phương thức hoạt động khác nhau của việc truyền và nhận

CHƯƠNG 4:

MÔ PHỎNG VÀ ĐÁNH GIÁ

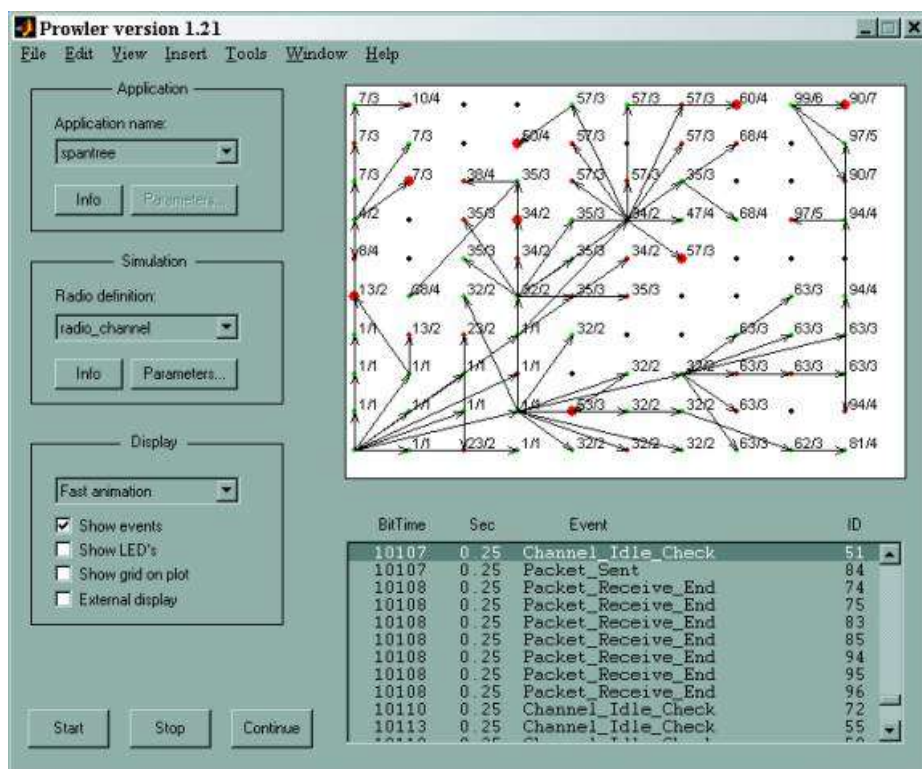
4.1 :Mô Phỏng giao thức MERLIN được viết bằng công cụ prowler trên môi trường Matlab

Prowler là gì?

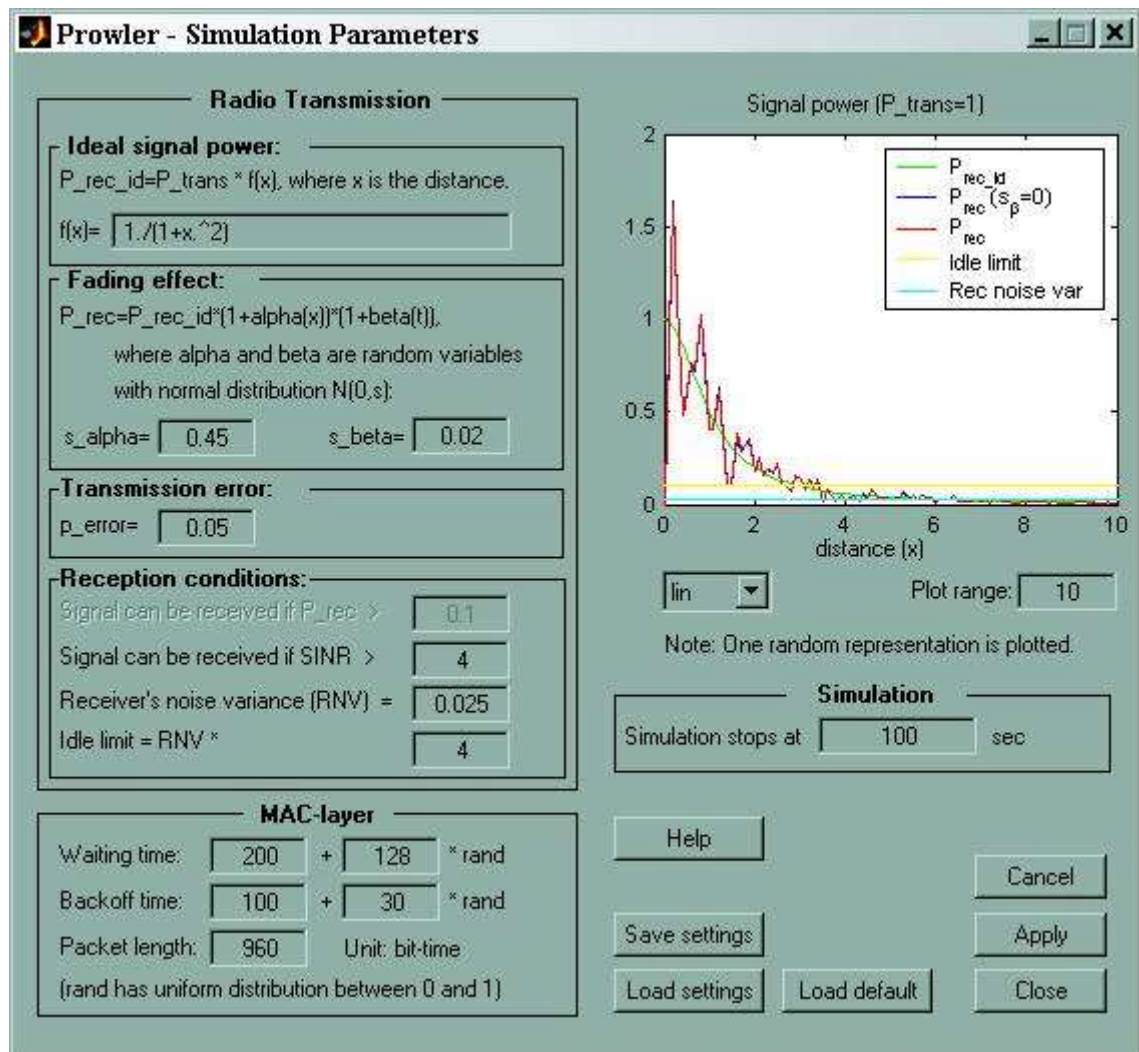
Prowler(Probabilistic Wireless Network Simulator):

Các hệ thống mạng nhúng là các hệ thống phân tán với quy mô rộng lớn và nguồn năng lực xử lý các node hạn chế được gắn với các tiện ích xử lý vật lý thông qua các cảm biến và thiết bị truyền. Các ứng dụng chạy trên nền tảng này được phân phối rộng lớn và có ảnh hưởng bởi các kênh truyền thông. Các mô phỏng có mô phỏng các hành vi của thiết bị nhưng không mô phỏng những hiệu ứng trong các kênh truyền tuy nhiên trong các kênh truyền không hoàn thiện thường ảnh hưởng tới công suất của các ứng dụng vì vậy rất cần các tương tác trong mô phỏng cho kết quả thực tế.

Prowler là hệ thống mô phỏng những hệ thống mạng không dây từ lớp ứng dụng cho tới lớp vật lý, Prowler chạy trên môi trường matlab lên để thiết lập những khả năng cho trực quan



Hình 4.1 Mô phỏng Prowler chạy trên môi trường Matlab



Hình 4.2 Mô phỏng Prowler chạy trên môi trường Matlab

4.2 Đánh giá về giao thức Mac và giao thức định tuyến

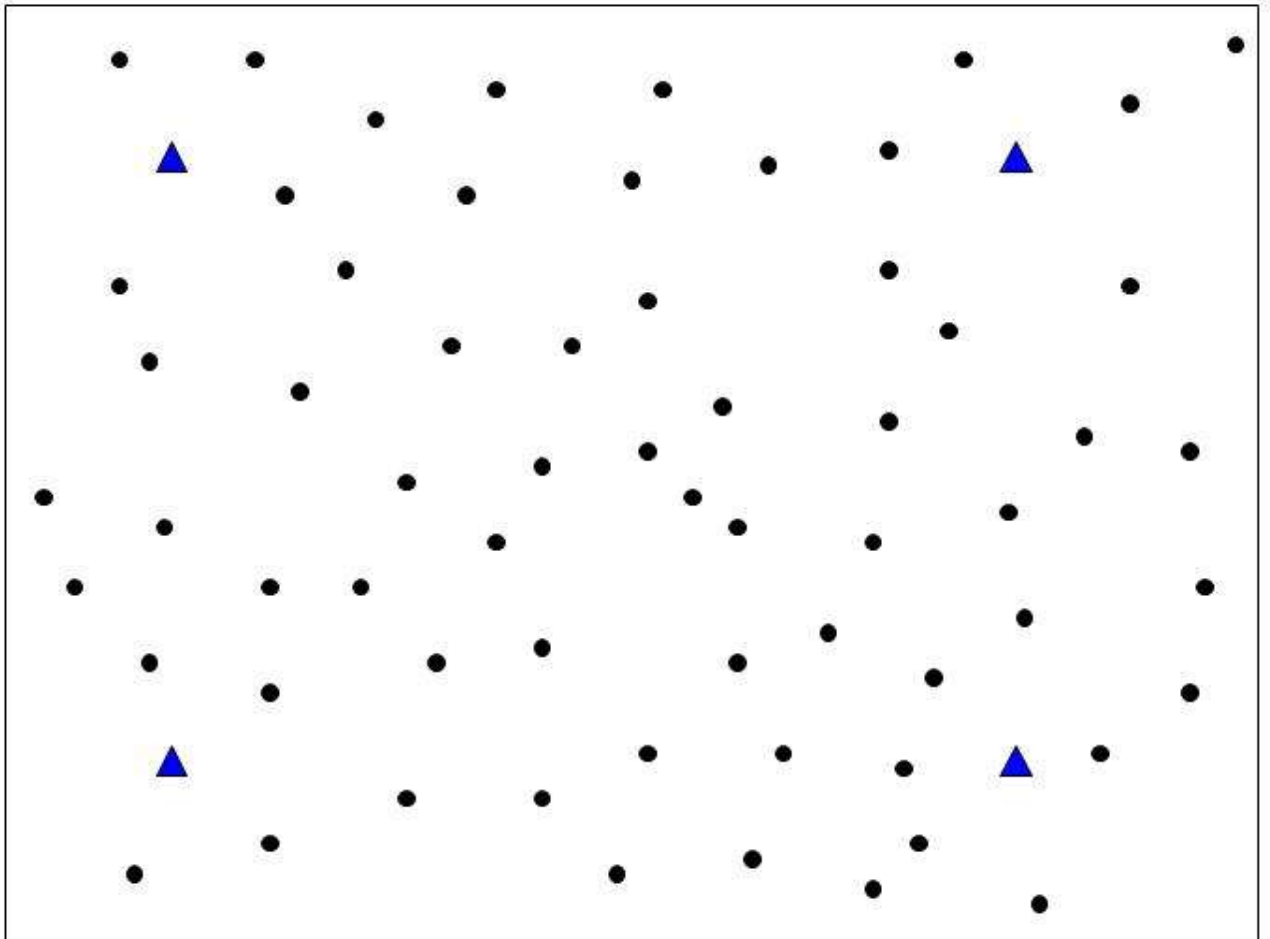
Mạng cảm biến khác với mạng dữ liệu không dây truyền thống trên một vài khía cạnh. Trước hết, đa số các nút trong những mạng cảm biến hoạt động dựa trên nguồn điện pin, và rất khó để nạp điện cho những nguồn pin của tất cả các nút. Thứ hai, những nút thường được triển khai trong một kiểu cách đặc biệt phi cấu trúc; chúng phải tự tổ chức hình thành một mạng truyền thông. Ba là, nhiều ứng dụng cần phải sử dụng số lượng lớn những nút, và mật độ nút sẽ thay đổi tại những địa điểm và thời gian khác nhau, với cả những mạng mật độ thưa lẫn những nút với nhiều lân cận. Cuối cùng, đa số các lưu thông trong mạng được thúc đẩy bởi những sự kiện cảm ứng, phân bố không đều và rất co cụm.

Tất cả những đặc trưng này cho thấy những giao thức MAC truyền thống không thích hợp cho những mạng cảm biến không dây nếu không có những sự cải biến. Chính vì những đặc điểm riêng biệt của mạng cảm biến mà việc định tuyến trong mạng cảm biến phải đối mặt với rất nhiều thách thức sau:

Mạng cảm biến có một số lượng lớn các nút, cho nên ta không thể xây dựng được sơ đồ địa chỉ toàn cầu cho việc triển khai số lượng lớn các nút đó vì lượng mào đầu để duy trì ID quá cao. Dữ liệu trong mạng cảm biến yêu cầu cảm nhận từ nhiều nguồn khác nhau và truyền đến sink. Các nút cảm biến bị ràng buộc khá chặt chẽ về mặt năng lượng, tốc độ xử lý, lưu trữ. Hầu hết trong các ứng dụng mạng cảm biến các nút nói chung là tĩnh sau khi được triển khai ngoại trừ một vài nút có thể di động. Mạng cảm biến là những ứng dụng riêng biệt. Việc nhận biết vị trí là vấn đề rất quan trọng vì tập hợp dữ liệu thông thường đưa lên vị trí. Khả năng dư thừa dữ liệu rất cao vì các nút cảm biến thu lượm dữ liệu dự trên hiện tượng chung.

Vì vậy việc kết hợp giữa Mac và định tuyến trong mạng cảm biến không đây là rất cần thiết nó giúp tránh xung đột giữa các node, giảm chi phí đường truyền, tránh xung đột giữa các gói tin... Việc kết hợp giao thức Mac và định tuyến nhằm kiểm soát truy cập và cung cấp dữ liệu đa chặng theo hai hướng chuyển tiếp đến gateway và ngược lại nhằm mục đích nâng cao hiệu năng mạng cảm nhận

Sau đây em xin chỉ ra những hình minh họa

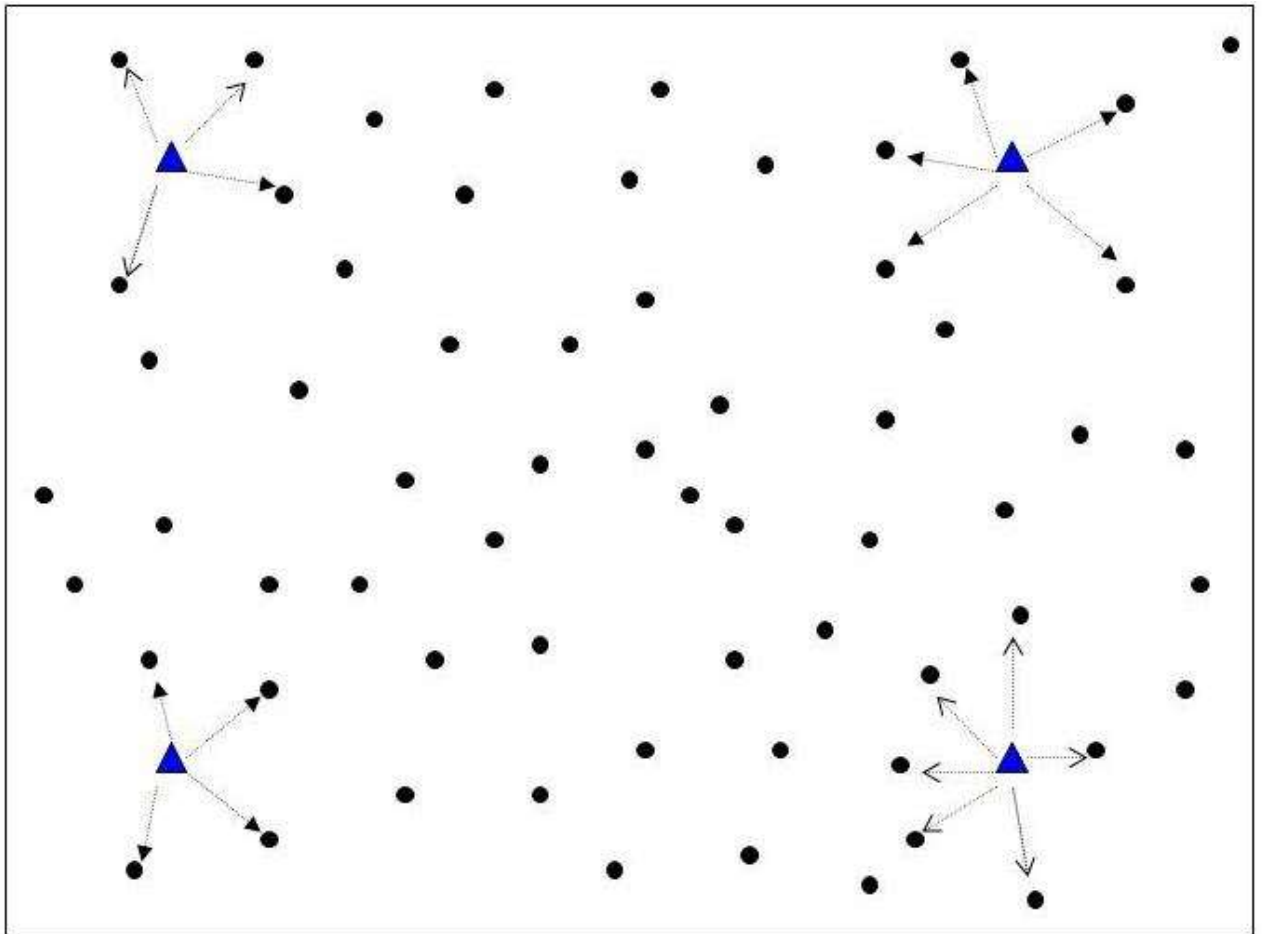


Hình 4.3: Mô phỏng các node và các gateway trong mạng cảm biến

▲ : Gateway

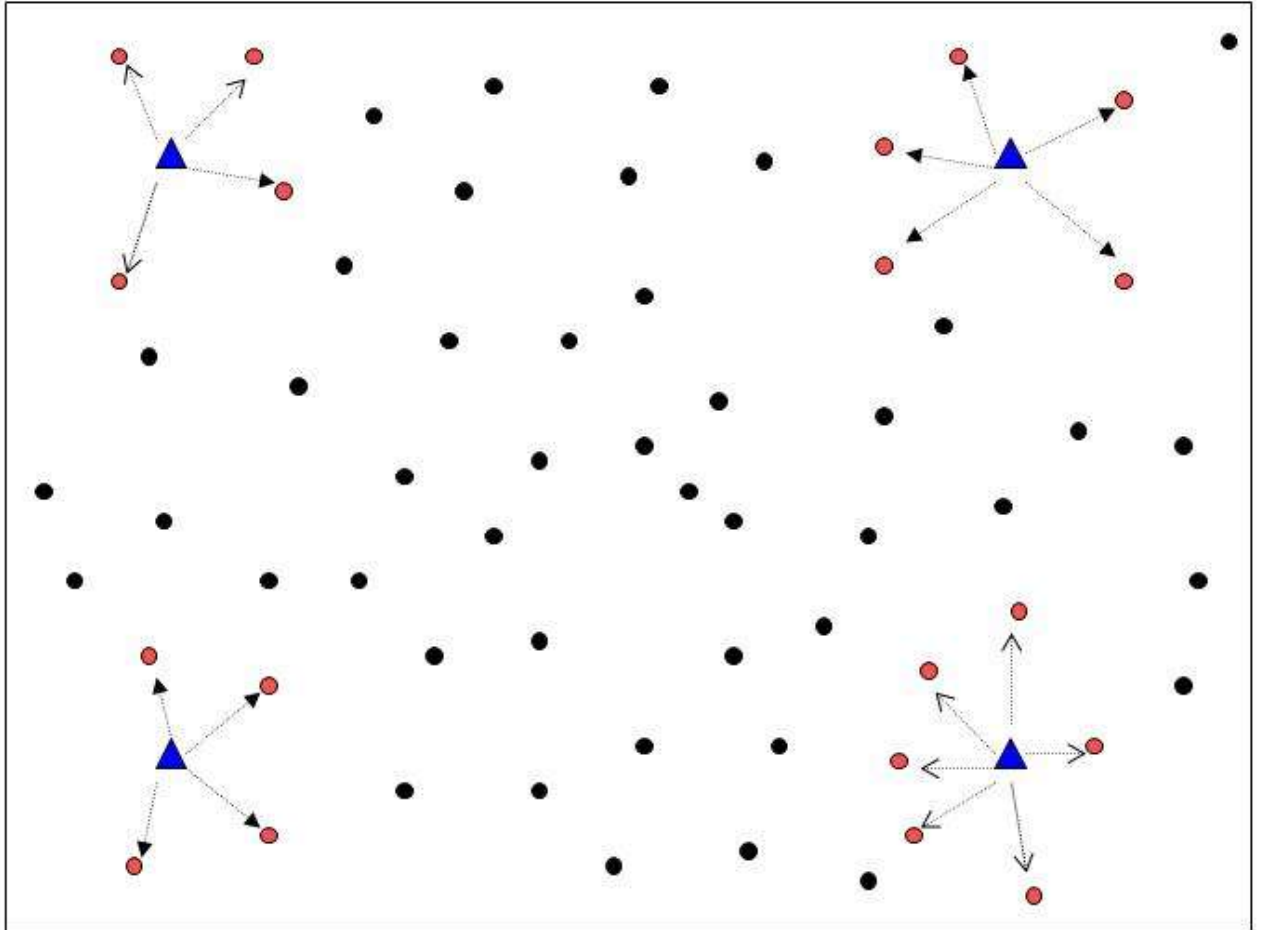
● : node

Trong giao thức MERLIN toàn mạng được chia thành những miền thời gian nhỏ. Miền thời gian được giới thiệu trong pha thiết lập với thông điệp phát quảng bá SYNC từ gateway tới các nút lắng giềng và ngược lại sẽ đồng bộ hóa đồng hồ nội bộ của chúng, sẽ thiết lập miền thời gian tăng biến đếm miền thời gian cho gói dữ liệu SYNC trước khi chuyển tiếp nó tới các nút xung quanh. Khi kết thúc phải thiết lập tất cả các node sẽ được tổ chức thành các miền thời gian tương ứng.



Hình 4.4: Các gói tin SYNC được truyền từ gateways đến các nút láng giềng

Nodes bắt đầu giai đoạn pha khởi động mạng bằng cách lắng nghe cho một gói SYNC, chứa thông tin thời gian, ID của người gửi và miền thời gian của người gửi. Gateways, được đồng bộ hóa tới miền thời gian tham khảo như nhau, bắt đầu khởi tạo mạng bằng việc phát quảng bá các gói tin SYNC. Gateways thiết lập miền thời gian của mình là 0. các nút cảm biến trong vùng lân cận của gateway nhận được gói SYNC dùng nó để đồng bộ hóa đồng hồ nội bộ của chúng. Khi chúng ở ngoài 1 hop của gateway thì các nút thiết lập giá trị của miền thời gian là 1

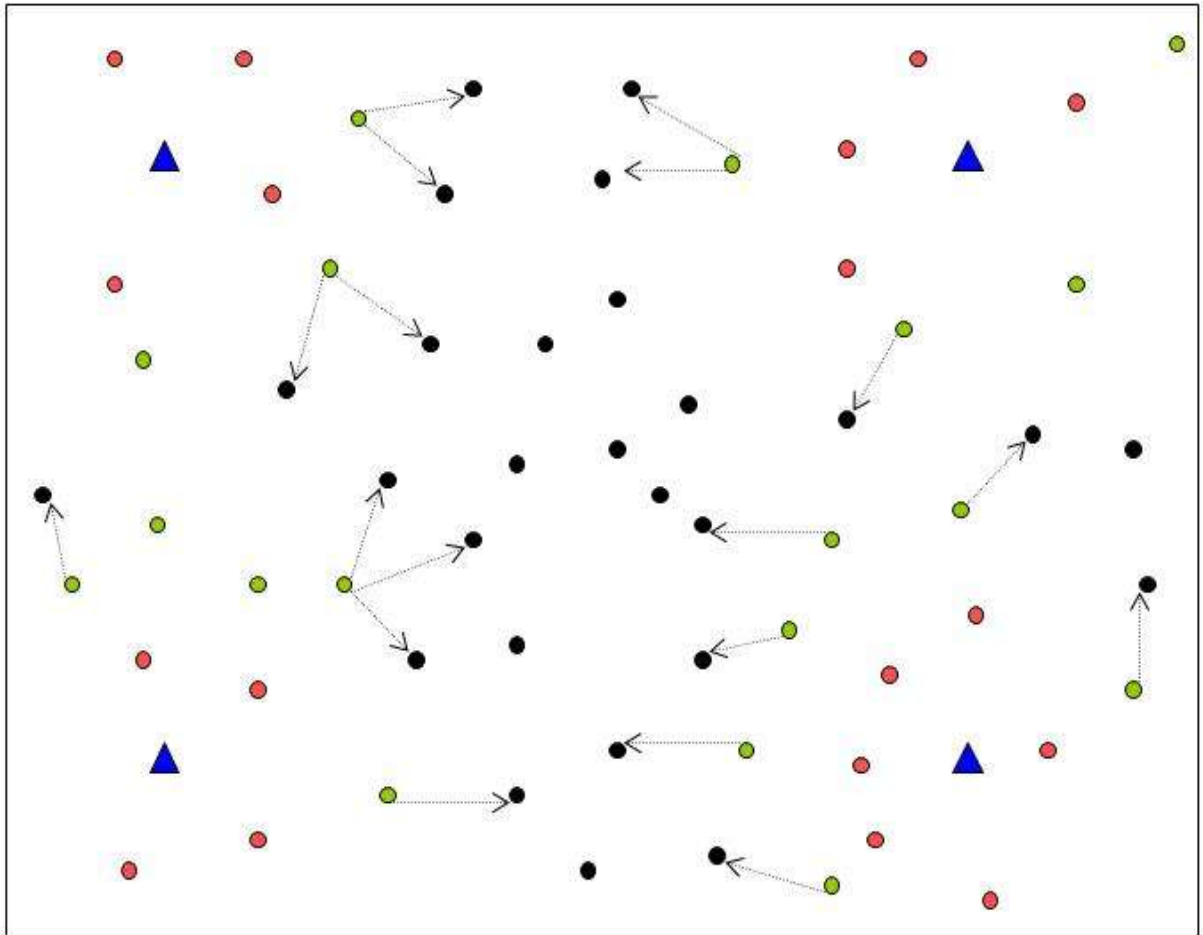


Hình 4.5: Mô hình truyền dữ liệu từ gateway đến các node trong cùng một miền thời gian

Các node cùng màu thì nằm trên cùng một miền thời gian

Các nút trong cùng một miền thời gian thì có chu kỳ cùng thức hoặc cùng ngủ. Việc cùng thức hoặc cùng ngủ đó sẽ tiết kiệm năng lượng cho các node.

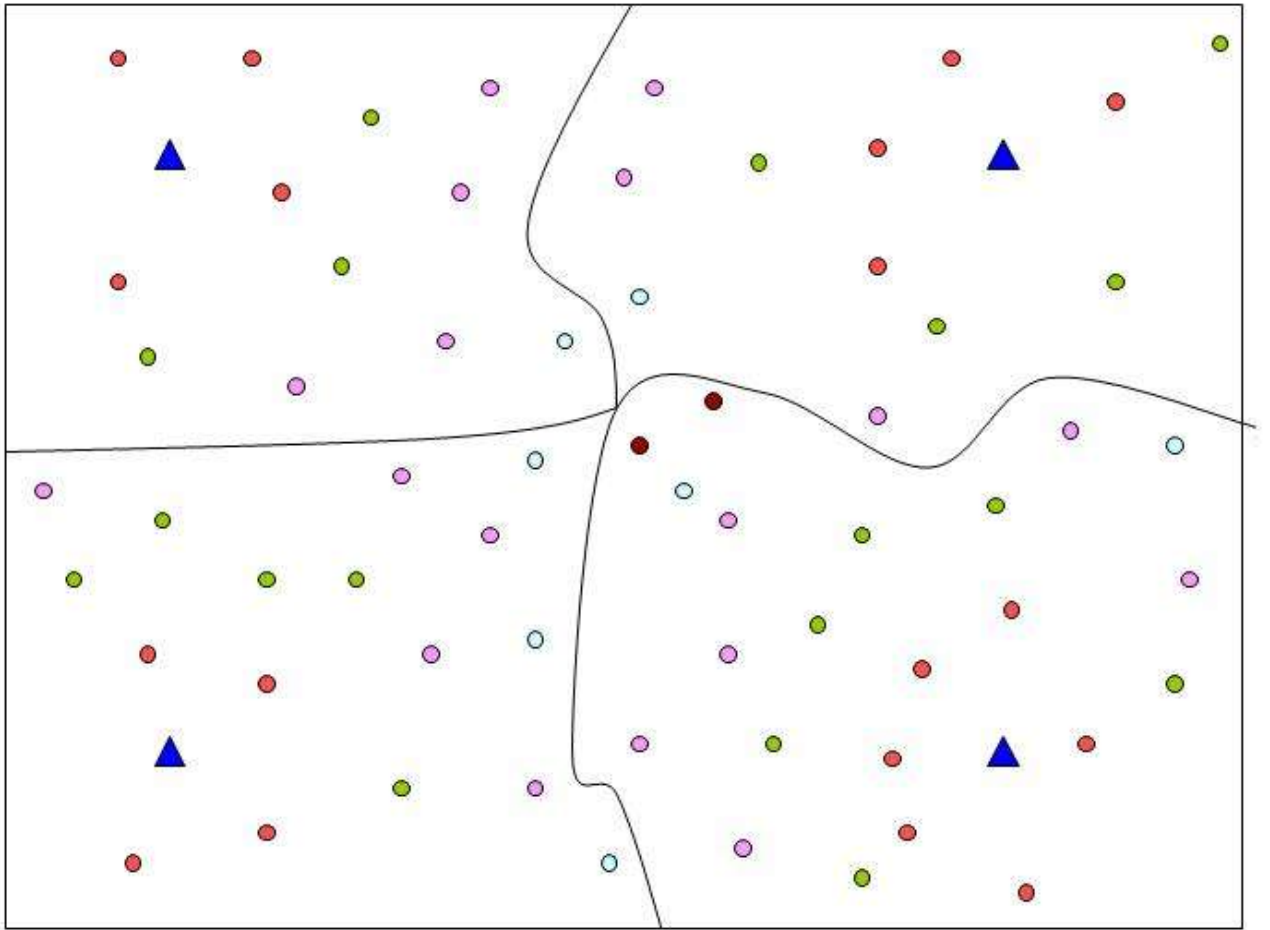
Miền thời gian node 1 sẽ chuyển tiếp gói SYNC tới các nút xa hơn. Cơ chế truyền tránh va chạm được mô tả trong phần 3.8 để đảm bảo hoạt động chuyển tiếp đúng. Tất cả các nút nhận được gói tin từ SYNC 1 nút trong múi giờ của mình để thiết lập múi giờ 2. Thủ tục này được lặp đi lặp lại cho đến khi tất cả các nút đã thiết lập miền thời gian của họ. Trong trường hợp nhiều cổng, các cổng bắt đầu loang mạng cùng một lúc bằng cách gửi một gói tin SYNC đến các nút lân cận. Trên SYNC nhận các gói tin từ hai cổng khác nhau, các nút có thể tính toán của họ thời gian khu để mỗi cổng và chọn miền thời gian của nó đối với các gateway gần nhất.



Hình 4.4: Mô hình chuyển tiếp các gói tin từ node đến node

Trong quá trình chuyển tiếp đến gateway, một nút có thể nhận được gói tin từ một số nút đó sẽ được chuyển tiếp đến các nút trên miền thời gian thấp hơn. Kể từ khi dữ liệu cảm biến là thường chỉ có vài bytes, ví dụ, nhiệt độ, áp suất, hoá dữ liệu và như vậy MERLIN tiếp xúc các tin nhắn và sau đó truyền cho chúng như là một gói duy nhất.

Do đó, một nút có thể gửi nhiều hơn một tin nhắn, tập hợp và gửi tin nhắn trong một gói liệu trong cùng một khe thời gian, mà cả hai tiết kiệm năng lượng. MERLIN không cụ thể địa chỉ một nút chuyển tiếp. Điều này có thể gây ra trùng lặp của các gói tin trong chuyển tiếp hoạt động. Tuy nhiên, thế hệ gói được điều khiển thông qua một cơ chế nghe lỏm, lắng nghe trên các tin nhắn được truyền qua một node lắng giềng để xác định chúng và sau đó xóa các bản sao các gói tin từ riêng.



Hình 4.5: Mô phỏng các miền thời gian trong mạng cảm nhận

Các nút trong cùng một tập hợp thì thuộc về cùng một gateway

Kết Luận

Ngày nay nhờ có những tiến bộ nhanh chóng trong khoa học và công nghệ sự phát triển của những mạng bao gồm các cảm biến giá thành rẻ, tiêu thụ ít năng lượng và đa chức năng đã nhận được những sự chú ý đáng kể. Hiện nay người ta đang tập trung triển khai các mạng cảm biến để áp dụng vào trong cuộc sống hàng ngày.

Đồ án này em đã tìm hiểu về mạng cảm biến không dây, cùng các ứng dụng của nó trong đời sống hiện nay. Nghiên cứu một số giao thức Mac như CSMA, S-MAC, TMAC... Giao thức định tuyến trong mạng cảm biến và những nguyên nhân gây lãng phí năng lượng khi xâm nhập môi trường. nêu ra những phương pháp kết hợp giữa giao thức Mac(CSMA) và giao thức định tuyến. Mô phỏng và đánh giá thủ tục đơn giản, thông qua chương trình Prowler mô phỏng các giao thức trong mạng cảm nhận không dây trên matlab.

Tuy nhiên việc kết hợp giao thức Mac và định tuyến trong mạng cảm biến mới chỉ dừng lại ở mức mô phỏng, chỉ phục vụ cho việc nghiên cứu và học tập chứ chưa xây dựng được mạng cảm biến thực tế .

Do hạn chế về thời gian thực tập, tài liệu và trình độ bản thân, bài đồ án của em không thể tránh khỏi những thiếu sót, rất mong các thầy cô góp ý và sửa chữa để bài đồ án tốt nghiệp của em được hoàn thiện hơn. Em xin chân thành cảm ơn!

Tài liệu tham khảo

1. Antonio G. Ruzzelli, *Media Access Controll and Routing*, Thesis 2008.
2. *WIRELESS SENSOR NETWORK – WSN*, Kỹ thuật, giao thức và ứng dụng- Vương Phát, Đỗ Duy Tân- Bộ môn Điện tử viễn thông trường Đh Bách khoa TPHCM
3. J. Alonso, A. Dunkels, , and T. Voigt. *Bounds on the energy consumption of routings in wireless sensor networks. In Proceedings of the 2ndWiOpt, Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 100–109, Mar 2004.
4. Chipcon AS. CC1010 datasheet. *Technical report, Chipcon AS, Oslo, Norway*, May 2005.
5. C. Barrett, A. Marathe, M.V. Marathe, and M. Drozda. *Characterizing the interaction between routing and mac protocols in ad-hoc networks. In Proceedings of MobiHoc '02, the 3rd ACM International symposium on Mobile ad hoc networking & computing*, pages 92–103, 2002.
6. R. Barry. Freertos. In *Proceedings of LCN'04 the 29th Annual IEEE International Conference on Local Computer Networks*, 2004
7. T. Falck, J. Espina, J. P. Ebert, and D. Dietterle. *Basuma - the sixth sense for chronically ill patients. In Proceedings of BSN 2006, International Workshop on Wearable and Implantable Body Sensor Networks*, pages 57–60, 2006.
8. <http://www.isis.vanderbilt.edu/Projects/nest/prowler/>