

MỤC LỤC

	Trang
MỤC LỤC	1
DANH MỤC HÌNH VẼ	3
DANH MỤC BẢNG BIỂU	4
LỜI NÓI ĐẦU	5
TÓM TẮT ĐỒ ÁN	6
CHƯƠNG 1: TỔNG QUAN VỀ MẠNG CẢM NHẬN KHÔNG DÂY	7
1.1 Khái niệm	7
1.2 Các ứng dụng của mạng cảm nhận không dây	7
1.2.1 Ngôi nhà thông minh.....	8
1.2.2 Giám sát các hoạt động công nghiệp	8
1.2.3 Ứng dụng trong chăm sóc sức khỏe	9
1.2.4 Giám sát an ninh trong quân đội và an toàn công nghiệp	10
1.2.5 Ứng dụng trong môi trường	11
1.3 Các chỉ tiêu của nút mạng cảm nhận không dây	12
1.3.1 Năng lượng.....	12
1.3.2 Kích thước và chi phí	12
1.3.3 Tính mềm dẻo	13
1.3.4 Sức mạnh.....	13
1.3.5 Bảo mật	14
1.3.6 Truyền thông	14
1.3.7 Tính toán	15
1.3.8 Đồng bộ thời gian.....	15
1.4 Kiến trúc của mạng WSN	15
1.4.1 Kiến trúc nút mạng.....	16
1.4.2 Kiến trúc mạng	17
CHƯƠNG 2: ĐỊNH TUYẾN TRONG MẠNG CẢM NHẬN KHÔNG DÂY ..	21
2.1 Giới thiệu	21
2.2 Thách thức trong vấn đề định tuyến	21
2.3 Các vấn đề về thiết kế giao thức định tuyến	22
2.3.1 Đặc tính thay đổi thời gian và trật tự sắp xếp của mạng.....	22

2.3.2 Ràng buộc về tài nguyên	22
2.3.3 Mô hình dữ liệu trong mạng cảm biến	22
2.3.4 Cách truyền dữ liệu	23
2.4 Phân loại và so sánh các giao thức định tuyến	24
2.5 Giao thức trung tâm dữ liệu.....	26
2.5.1 Flooding và Gossiping	26
2.5.2 SPIN	27
2.5.3 Directed Diffusion.....	28
2.6 Giao thức phân cấp	31
2.6.1 LEACH	31
2.6.2 PEGASIS.....	33
2.7 Giao thức dựa trên vị trí.....	34
2.7.1 GAF.....	35
2.7.2 GEAR	37
2.8 Kết luận	38
CHƯƠNG 3: MÔ PHỎNG PEGASIS BẰNG MOBILITY FRAMEWORK	
CỦA OMNeT++	39
3.1 Giới thiệu về OMNeT++ và Mobility Framework.....	39
3.1.1 Giới thiệu về OMNeT++.....	39
3.1.2 Giới thiệu về Mobility.....	42
3.2 Giới thiệu về PEGASIS.....	48
3.2.1 PEGASIS cơ bản.....	49
3.2.2 PEGASIS cải tiến.....	50
3.3 Mô phỏng	52
3.3.1 Mô hình năng lượng.....	52
3.3.2 Giả thiết và thiết lập thông số ban đầu cho quá trình mô phỏng	57
3.3.3 Kết quả mô phỏng.....	63
3.4 Kết luận và hướng nghiên cứu tiếp theo	65
KẾT LUẬN	66

DANH MỤC HÌNH VẼ

STT	Tên hình vẽ	Trang
1	Hình 1.1: Cấu trúc mạng cảm nhận không dây	7
2	Hình 1.2: Kiến trúc mạng đơn	18
3	Hình 1.3: Kiến trúc mạng liên kết bước	18
4	Hình 1.4: Kiến trúc mạng liên kết bó	19
5	Hình 2.1: Mô hình truyền dữ liệu giữa sink và các nút	24
6	Hình 2.2: Truyền gói trong Flooding	26
7	Hình 2.3: Ba tín hiệu bắt tay của Spin	27
8	Hình 2.4: Hoạt động của Spin	28
9	Hình 2.5: Hoạt động cơ bản của Directed Diffusion	30
10	Hình 2.6: Mô tả mạng Leach	32
11	Hình 2.7: Mạng lưới ảo trong Gaf	36
12	Hình 2.8: Sự chuyển trạng thái trong Gaf	36
13	Hình 2.9: Chuyển tiếp địa lý đệ quy trong Gaf	38
14	Hình 3.1: Cấu trúc phân cấp module trong OMNeT++	40
15	Hình 3.2: Các kết nối trong OMNeT++	41
16	Hình 3.3: Cấu trúc của host di động	43
17	Hình 3.4: Cấu trúc kế thừa module trong MF	45
18	Hình 3.5: Xây dựng chuỗi sử dụng thuật toán Greedy	49
19	Hình 3.6: Xử lý lỗi khi một nút trong chuỗi chết	50
20	Hình 3.7: Khắc phục của Pegasus	52
21	Hình 3.8: Mô hình năng lượng đơn giản	55
22	Hình 3.9: Trạm BS gửi broadcast đến cho các nút trong mạng	59
23	Hình 3.10: Trạm BS gửi bản tin Max Distance đến các nút xa nhất	60
24	Hình 3.11: Nút xa nhất chuỗi gửi bản tin Invite mời nút gần nhất vào chuỗi	61
25	Hình 3.12: Các nút kết nối vào nhau tạo thành chuỗi	61
26	Hình 3.13: Chuỗi sau khi thiết lập xong	62
27	Hình 3.14: Kết quả mô phỏng mạng có kích thước (50m,50m) với năng lượng ban đầu của nút là 0.25J	64
28	Hình 3.15: Kết quả mô phỏng khi kích thước mạng là (100m,100m) với năng lượng ban đầu của nút là 0.5J	65

DANH MỤC BẢNG BIỂU

STT	Tên bảng biểu	Trang
1	Bảng 2.1: Phân loại và so sánh các giao thức chọn đường trong WSN	25
2	Bảng 2.2: Miêu tả internet sử dụng cặp thuộc tính - giá trị	29
3	Bảng 3.1: Các loại bảng tin tương ứng của các lớp	46
4	Bảng 3.2: Số vòng khi 1%, 20%, 50% và 100% nút chết	64

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 và không thể nạp lại. 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 giao thức định tuyến PEGASIS. Giao thức này cải thiện đáng kể thời gian sống của mạng cảm biến, và em quyết định chọn đề tài này làm đề án tốt nghiệp.

Để có thể hoàn thành được đề án tốt nghiệp này, em đã được học hỏi những kiến thức quý báu từ các thầy, cô giáo của Trường Đại học DL 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 chân thành cảm ơn ThS. Nguyễn Văn Thế 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 đề án.

TÓM TẮT ĐỒ ÁN

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.

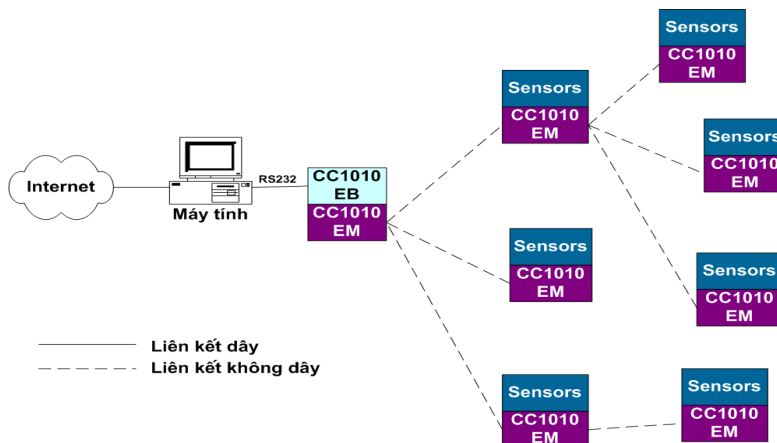
Mạng cảm biến là một lĩnh vực rất sâu rộng, đồ án này sẽ giới thiệu một cách khái quát nhất về các đặc điểm của mạng cảm biến. Sau đó phần cuối sẽ nghiên cứu và đưa ra giải thuật định tuyến PEGASIS nhằm cải thiện đáng kể thời gian sống của mạng.

CHƯƠNG 1: TỔNG QUAN VỀ MẠNG CẢM NHẬN KHÔNG DÂY

1.1 Khái niệm

Mạng cảm nhận không dây (Wireless Sensor Network – WSN) là một mạng không dây mà các nút của nó sử dụng các vi điều khiển, cảm biến, bộ truyền RF... với các đặc trưng: kích thước nhỏ, tiêu thụ điện năng thấp, tự tổ chức, tự bảo trì, giá thành thấp và chu trình tác vụ thấp dùng để đo các dữ liệu và truyền không dây giữa các nút.

Các mạng cảm nhận không dây bao gồm một tập hợp các nút mạng kết nối không dây (bằng sóng điện từ), mà mỗi nút trong đó được trang bị với một hoặc nhiều đầu cảm nhận, các hệ thống truyền thông, lưu trữ và xử lý tài nguyên,... Các đầu cảm nhận trong các nút có thể quan sát các hiện tượng như là nhiệt, quang, âm thanh, địa chấn và các sự kiện gia tốc, để xử lý, phân tích dữ liệu thô và trả lời các yêu cầu cụ thể của người dùng. Với sự tiến bộ trong công nghệ gần đây đã mở đường cho việc thiết kế và thi hành của các thể hệ nút mạng cảm nhận mới, có kích thước rất nhỏ và giá thành thấp với các nhân tố có khả năng giao tiếp không dây và tính toán hết sức tinh vi. Mặc dù mới được phát triển, các mạng cảm nhận không dây cho thấy sẽ đem lại một tiềm năng lớn cho nhiều ứng dụng trong tất cả các lĩnh vực của đời sống.



Hình 1.1: Cấu trúc mạng cảm nhận không dây

1.2 Các ứng dụng của mạng cảm nhận không dây

Ngày nay với sự phát triển của công nghệ cao, các mạng cảm nhận không dây không ngừng được phát triển và được ứng dụng rất nhiều vào cuộc sống. Các ứng dụng của mạng cảm nhận không dây mà chúng ta có thể thấy rõ nhất như là:

1.2.1 Ngôi nhà thông minh

Nhà là môi trường ứng dụng rất lớn cho những mạng cảm nhận không dây. Nhiều ứng dụng công nghiệp được xây dựng trong nhà. Nhiều tiện nghi khác trong nhà có thể thực hiện như điều khiển từ xa, thiết bị số thiết bị trợ giúp cá nhân (PDA) có thể điều khiển tivi, đầu DVD, giàn âm thanh và các thiết bị điện tử khác ở trong nhà (đèn điện, rèm che và khoá) cũng có thể điều khiển bằng một mạng cảm nhận không dây. Với điều khiển từ xa của một thiết bị điều khiển từ xa như vậy có thể điều khiển toàn bộ tiện nghi trong nhà trong khi vẫn ngồi trên ghế bành. Tuy nhiên, tiềm năng hấp dẫn nhất của mạng cảm nhận không dây là sự kết hợp nhiều dịch vụ như việc cho phép những rèm cửa đóng tự động khi truyền hình được bật hoặc có thể tự động tắt tiếng tivi, hệ thống giải trí ở nhà khi nhận điện thoại hoặc có chuông cửa.

Việc sử dụng mạng cảm nhận không dây trong nhà được kỳ vọng là việc ghép nối các thiết bị ngoại vi với máy tính cá nhân như bàn phím và con chuột không dây. Những ứng dụng này có lợi thế là giá thành thấp và tiêu thụ điện năng ít là điều kiện thiết yếu của mạng cảm nhận không dây.

Những đồ chơi hiện tại là thị trường rộng lớn khác cho ứng dụng mạng cảm nhận không dây. Danh sách đồ chơi được hỗ trợ bằng hoặc điều khiển bởi mạng cảm nhận không dây càng lớn và tính năng điều khiển ô tô và tàu thuyền bằng sóng vô tuyến truyền thống đến những trò chơi máy tính dùng cần điều khiển và thiết bị điều khiển không dây càng tăng.

Một ứng dụng quan trọng khác trong nhà chính là khoá không có chìa điều khiển từ xa (RKE - Remote Keyless). Đặc tính truy cập khoá không chìa (RKE) ứng dụng trên xe ô tô, cửa và cửa sổ, đèn trong nhà bằng những cảm nhận điều khiển không dây, chủ nhà có thể có một thiết bị như một chìa khoá với duy nhất một nút bấm. Khi nút này được nhấn thiết bị khoá tất cả các cửa ra vào, cửa sổ trong nhà, tắt đèn trong nhà (để một vài đèn ngủ), bật đèn bảo vệ an toàn ở bên ngoài và đặt hệ thống HVAC điều khiển nhà ở chế độ ngủ hoặc đóng, mở cửa xe ô tô...

1.2.2 Giám sát các hoạt động công nghiệp

Phòng điều khiển bao gồm các chi dẫn và các màn hình mô tả trạng thái của dây chuyền (như trạng thái của những van, tình trạng của các thiết bị, nhiệt độ và áp suất của các vật liệu,) cũng như các thiết bị nhập vật liệu, điều khiển toàn bộ kế hoạch (đóng mở van, bếp lò...) và giám sát trạng thái dây chuyền. Những cảm biến mô tả trạng thái vật lý của dây chuyền, những màn hình trong phòng điều khiển việc

nhập vào những thiết bị, nguyên liệu. Thông tin được truyền thông là thông tin trạng thái thường thay đổi chậm. Như vậy, trong thao tác bình thường thông lượng dữ liệu của mạng chỉ cần tốc độ thấp, đòi hỏi độ tin cậy của mạng càng cao. Một mạng cảm nhận không dây có nhiều nút cung cấp nhiều kênh truyền thông thông báo tới các nút.

Ví dụ trong điều khiển chiếu sáng thương mại chi phí trong sự sắp đặt hệ thống ánh sáng trong một tòa nhà liên quan đến việc điều khiển ánh sáng – nơi đặt các công tắc bật hoặc tắt ánh sáng hoặc làm giảm cường độ chiếu sáng... Một hệ thống không dây linh hoạt điều khiển bằng một chương trình có thể điều khiển một số lớn đèn bằng nhiều cách trong khi vẫn đảm bảo sự an toàn của hệ thống chiếu sáng thương mại.

Vì mạng không dây sử dụng các giải thuật phân tán có nhiều kênh và có thể tự sửa chữa và bảo trì nên phù hợp với sự thay đổi (tăng, giảm) của các dây chuyền công nghiệp, cung cấp thông tin chính xác về tình trạng dây chuyền trong các điều kiện khó khăn.

Mạng không dây phù hợp cho việc giám sát và điều khiển vận hành hoặc chuyển động của máy móc trong một không gian nhất định. Trong những ứng dụng như vậy các cảm biến và cần điều khiển rất quan trọng để theo dõi nhiệt độ, rung động, sự bôi trơn ... những thành phần quay của máy để tối ưu hóa thời gian bảo trì định kỳ.

1.2.3 Ứng dụng trong chăm sóc sức khỏe

Hai dạng ứng dụng theo dõi sức khỏe của mạng cảm nhận không dây. Một là theo dõi thể lực: quần áo mặc có thể theo dõi xung nhịp, hơi thở qua những cảm nhận và gửi thông tin tới một máy tính cá nhân để phân tích. Dạng khác là theo dõi sức khỏe tại nhà: quản lý cân nặng, trọng lượng của bệnh nhân qua mạng không dây được gửi cho một máy tính cá nhân hay theo dõi lượng đường trong máu để theo dõi bệnh tiểu đường.

Sử dụng những mạng cảm nhận không dây trong theo dõi sức khỏe đang tăng tốc vì sự phát triển của những cảm nhận sinh vật thích hợp với công nghệ mạch tích hợp CMOS truyền thống. Những cảm nhận này có thể phát hiện ra những men, axit nucleic và nguyên liệu sinh vật quan trọng khác mà kích thước rất nhỏ và không đất dẫn tới nhiều ứng dụng trong dược học và chăm sóc y học.

Theo dõi bác sĩ và bệnh nhân trong bệnh viện : mỗi bệnh nhân được gắn một nút mạng cảm biến có kích thước nhỏ gọn, mỗi nút cảm biến này có nhiệm vụ riêng.

Ví dụ cảm biến xác định nhịp tim trong khi con cảm biến khác xác định áp suất máu, các bác sĩ cũng có thể mang nút cảm biến để xác định được vị trí của họ trong bệnh viện.

1.2.4 Giám sát an ninh trong quân đội và an toàn công nghiệp

Một trong những lợi ích to lớn của việc sử dụng mạng cảm nhận không dây là chúng có thể thay thế cho nhân viên bảo vệ, những người lính ở những khu vực bảo vệ, canh gác đảm bảo sự an toàn cho họ. Mạng cảm nhận không dây có thể sử dụng trong các mô nguy hiểm thay cho việc phải sử dụng con người trong các công việc mạo hiểm trong thời gian khai thác. Ngoài những ứng dụng bảo vệ mạng cảm nhận không dây có thể sử dụng để định vị và xác định những mục đích tấn công tiềm ẩn và hỗ trợ tấn công. Mạng có thể được trang bị bằng những mic, cảm biến thu rung động địa chấn, cảm biến từ tính, radar băng tần rộng và những cảm biến khác. Mạng cảm nhận không dây có thể rất nhỏ, đơn giản và được ngụy trang như các viên đá, cây hoặc rác thải ven đường. Do có đặc tính giống tự nhiên, phù hợp với địa hình tự nhiên nên những mạng không dây (không cần cơ sở hạ tầng) được sử dụng rất nhiều. Những mạng này sử dụng những giải thuật xử lý phân tán và lộ trình đường đi (không có một nút bị hỏng, vì khi hỏng sẽ tự tìm một đường đi khác), một đặc tính làm cho mạng khó bị phát hiện và phá hủy. Sự sử dụng kỹ thuật trải phổ rộng, kết hợp với nhiều khuôn dạng truyền tới nhiều mạng cảm nhận không dây khác (để tối ưu hóa pin nguồn) làm cho chúng khó bị phát hiện bởi một thiết bị điện tử khác.

Những khả năng xác định vị trí của mạng cảm nhận không dây có thể cho phép những nút mạng sẽ được sử dụng như những phần tử của một mảng tự định hướng và điều khiển theo sự bức xạ phân tán ngẫu nhiên của những phần tử; một mảng như vậy có thể sử dụng để cung cấp bộ lọc dữ liệu của mạng cảm nhận không dây.

Thông tin định vị tương đối được sử dụng để chỉnh và liên kết pha của tín hiệu được truyền bởi mỗi nút; với thông tin này dữ liệu lọc được truyền không phải là duy nhất theo hướng tín hiệu đến nhưng có thể đến bất kỳ hướng nào cần đến. Kỹ thuật Beamforming được áp dụng cho các cảm nhận để tăng cường tính linh hoạt của chúng và cải thiện xác suất phát hiện.

Giám sát chiến trường: địa hình hiểm trở, các tuyến đường, đường mòn và các chỗ eo hẹp có thể nhanh chóng được bao phủ bởi mạng cảm biến và gần như có thể theo dõi được các hoạt động của quân địch.

Phát hiện thăm dò các cuộc tấn công bằng hóa học, sinh học, hạt nhân : trong các cuộc chiến tranh hóa học và sinh học đang gần kề, một điều rất quan trọng là phát hiện đúng lúc và chính xác các tác nhân đó. Mạng cảm biến triển khai ở những vùng mà được sử dụng như là hệ thống cảnh báo sinh học và hóa học có thể cung cấp các thông tin mang ý nghĩa quan trọng đúng lúc nhằm tránh thương vong nghiêm trọng.

Hệ thống bảo vệ an ninh không dây mô tả ở ứng dụng trong nhà có thể được sử dụng trong những ứng dụng an toàn công nghiệp, hỗ trợ nhiều cảm biến phù hợp trong an ninh công nghiệp như cửa ra vào kiểm soát bằng hồng ngoại, cửa mở bằng từ và những cảm biến báo kính vỡ, những cảm biến phát hiện sự can thiệp trực tiếp đến con người.

1.2.5 Ứng dụng trong môi trường

Cung cấp thông tin về mưa, độ ẩm, nhiệt độ : một vùng canh tác rộng lớn và những trại chăn nuôi có thể bao trùm vài dặm vuông và chúng có thể nhận mưa rời rạc và chỉ trên vài phần của nông trại. Một ứng dụng như vậy thì lý tưởng cho những mạng cảm nhận không dây: lượng dữ liệu thấp “có mưa hay không?” được gửi từng phút trong mạng chi phí thấp và tiêu thụ điện thấp trong mùa gieo trồng.

Mạng cảm nhận không dây phù hợp với với một sự đa dạng gần như vô hạn của những cảm biến sinh vật và hóa học. Dữ liệu do một mạng như vậy có khả năng cung cấp nông dân độ ẩm của đất, nhiệt độ, nhu cầu hóa chất diệt côn trùng (thuốc sát trùng), thuốc diệt cỏ và phân bón, mức độ nhận nắng và nhiều số liệu khác.

Trong chăn nuôi gia súc: Những chủ trại có những trại chăn nuôi rộng mênh mông có thể sử dụng những mạng cảm nhận không dây trong việc xác định vị trí của những động vật và với những cảm nhận đặt trên mỗi động vật xác định nhu cầu cho những nghiên cứu ngăn ngừa các sinh vật ký sinh, bệnh tật.

Phát hiện cháy rừng: vì các nút mạng cảm biến có thể được triển khai một cách ngẫu nhiên nên phù hợp với mọi địa hình trong rừng, núi. Các nút mạng sẽ dò tìm nguồn gốc của lửa để thông báo về trung tâm biết trước khi lửa lan rộng không kiểm soát được. Hàng triệu các nút mạng cảm biến có thể được triển khai và tích hợp sử dụng trong tần số không dây hoặc quang học. Chúng có thể được trang bị cách thức sử dụng công suất có hiệu quả như là pin mặt trời vì các nút cảm biến bị bỏ lại hàng tháng, hàng năm.

1.3 Các chỉ tiêu của nút mạng cảm nhận không dây

Sau đây là những chỉ tiêu để đánh giá một nút mạng cảm nhận không dây. Qua các tiêu chí đánh giá đó ta có thể lựa chọn loại vi điều khiển thích hợp cũng như để xây dựng hệ thống hiệu quả.

1.3.1 Năng lượng

Để đạt được yêu cầu duy trì năng lượng hoạt động trong nhiều năm thì các nút mạng cần phải tiêu thụ năng lượng rất thấp. Việc tiêu thụ năng lượng thấp chỉ đạt được bằng cách kết hợp các thành phần phần cứng năng lượng thấp và chu trình hoạt động ngắn. Trong thời gian hoạt động truyền thông radio sẽ tiêu thụ một phần năng lượng đáng kể trong tổng mức tiêu thụ năng lượng của nút mạng. Các thuật toán và các giao thức cần được phát triển để giảm hoạt động truyền nhận radio.

Những ứng dụng mạng cảm nhận không dây yêu cầu những thành phần tiêu thụ điện trung bình thấp hơn mạng không dây hiện thời đã có như Bluetooth. Ví dụ những thiết bị cho những kiểu cảm nhận y học và công nghiệp nhất định... yêu cầu năng lượng từ những nguồn pin nhỏ cần phải kéo dài từ vài tháng đến nhiều năm.

Những ứng dụng liên quan đến kiểm tra và điều khiển thiết bị công nghiệp yêu cầu tuổi thọ nguồn pin dài đặc biệt để duy trì lịch trình hiện tại của thiết bị theo dõi có thể chưa xác định. Những ứng dụng khác như theo dõi môi trường những vùng lớn yêu cầu rất nhiều thiết bị làm cho việc thay thế nguồn pin thường xuyên là không thực tế.

1.3.2 Kích thước và chi phí

Kích thước vật lý và giá thành của mỗi nút riêng lẻ có ảnh hưởng tới sự dễ dàng và chi phí khi triển khai. Tổng chi phí vật tư và chi phí triển khai ban đầu là hai yếu tố chủ chốt dẫn đến việc chấp nhận công nghệ WSN. Với một ngân sách cố định việc giảm giá thành trên mỗi nút mạng sẽ làm cho có khả năng mua thêm nhiều nút, triển khai một mạng thu thập với mật độ cao hơn, thu thập được nhiều dữ liệu hơn.

Với mục tiêu này thiết kế giao thức và truyền thông mạng phải tránh nhu cầu những thành phần giá cao, rời rạc mà việc sử dụng mạng nên có thể ở mọi nơi bằng việc tối giản yêu cầu giao thức phức tạp và cần nhiều bộ nhớ.

Ngoài ra một trong những yếu tố làm giá thành của nhiều mạng lớn là quản trị và bảo trì hệ thống nên mạng cảm nhận không dây cần phải thiết kế đặc biệt và có khả năng tự cấu hình và tự bảo trì.

Một mạng “đặc biệt” trong ngữ cảnh này là một mạng không có một phân phối vật lý hoặc địa thế logic định trước của các nút. “Tự cấu hình” là khả năng của nút mạng phát hiện ra sự có mặt nút khác và tổ chức vào trong một mạng có cấu trúc, có chức năng mạng mà không có sự can thiệp con người. “Tự bảo trì” được định nghĩa là khả năng mạng phát hiện ra và phục hồi những lỗi xuất hiện trong những nút mạng hoặc liên kết truyền thông mà không có sự can thiệp con người.

Để dễ dàng sản xuất theo những khả năng mong muốn của những hệ thống và những thiết bị như vậy đáp ứng việc tối giản giá của các thành phần mạng không dây thì sự phát triển một tiêu chuẩn hóa nghi thức truyền thông là rất cần thiết.

1.3.3 Tính mềm dẻo

Các nút mạng phải có khả năng thích nghi cao để thích hợp với các ngữ cảnh khác nhau. Mỗi một ứng dụng sẽ yêu cầu về thời gian sống, tốc độ lấy mẫu, thời gian đáp ứng và xử lý nội mạng khác nhau. Một kiến trúc WSN cần phải đủ mềm dẻo để cung cấp một dải rộng các ứng dụng. Thêm vào đó vì lý do chi phí mỗi thiết bị sẽ chỉ có phần cứng và phần mềm cho một ứng dụng cụ thể. Kiến trúc cần phải đơn giản để kết hợp giữa phần cứng và phần mềm. Vì vậy những thiết bị này đòi hỏi một mức độ cao về tính modul của phần cứng và phần mềm trong khi vẫn giữ được tính hiệu quả.

1.3.4 Sức mạnh

Để hỗ trợ các yêu cầu về thời gian sống, mỗi nút cần phải càng mạnh càng tốt. Trong thực tế hàng trăm nút mạng sẽ hoạt động trong nhiều năm. Để đạt được điều này hệ thống cần được xây dựng để vẫn có thể hoạt động khi một nút có lỗi. Module hóa hệ thống là một công cụ mạnh để phát triển hệ thống. Bằng cách chia chức năng hệ thống thành các thành phần con độc lập, mỗi chức năng có thể được kiểm tra đầy đủ trước khi kết hợp chúng thành một ứng dụng hoàn chỉnh. Để làm điều này, các thành phần hệ thống phải độc lập đến mức có thể và có giao tiếp chặt chẽ để ngăn chặn các tương tác không mong muốn. Để tăng sức mạnh của hệ thống khi nút bị lỗi, một WSN cũng cần có khả năng đối phó với nhiễu ngoài. Các mạng thường cùng tồn tại với các hệ thống không dây khác, chúng cần phải có khả năng để thích nghi theo các hành động khác nhau. Nó cũng phải có khả năng hoạt động trong môi trường đã có các thiết bị không dây khác hoạt động với một hay nhiều tần số. Khả năng tránh tắc nghẽn tần số là điều cốt yếu để đảm bảo một sự triển khai thành công.

1.3.5 Bảo mật

Để đạt được mức độ bảo mật mà các ứng dụng yêu cầu, các nút riêng lẻ cần có khả năng thực hiện sự mã hóa phức tạp và thuật toán xác thực. Truyền dữ liệu không dây rất dễ bị chặn. Chỉ có một cách bảo mật dữ liệu là mã hóa toàn bộ dữ liệu truyền, CPU cần có khả năng tự thực hiện các thao tác mật mã. Để bảo mật toàn bộ các dữ liệu truyền, các nút cần tự bảo mật dữ liệu của chúng.

Mạng thật sự an toàn như thế nào và việc mạng được lĩnh hội cho những người dùng đặc biệt là những người dùng tiềm năng. Sự nhận thức về sự an toàn rất quan trọng bởi vì người dùng cần biết dữ liệu của họ được truyền qua không gian (sóng) cho người nhận (khả năng bị người khác lấy là rất cao).

Thường một ứng dụng sử dụng mạng cảm nhận không dây thay thế mạng có dây-nơi mà người dùng có thể nhìn thấy dây hoặc cáp mang thông tin của họ và biết rằng với sự chắc chắn hợp lý không ai khác người đang nhận thông tin của họ có thể sửa chữa thông tin hoặc nhận dữ liệu. Nên sự an toàn thông tin chỉ bằng mã hóa thông báo tuy nhiên, trong nhiều ứng dụng sự mã hoá thì không phải là mục đích an toàn quan trọng của những mạng cảm nhận không dây mà mục đích an toàn quan trọng nhất là bảo đảm cho bất kỳ người nhận được thông báo từ người gửi không sửa đổi thông tin bên trong bằng bất kỳ cách nào.

Việc này yêu cầu một sự kiểm tra an toàn, chứng thực thông báo và toàn vẹn được thực hiện bởi việc thêm vào một mã toàn vẹn thông báo phụ thuộc (MIC) cho việc truyền thông báo (trong lĩnh vực an toàn MIC thường được gọi là mã chứng thực thông báo (MAC), nhưng MIC được sử dụng trong văn bản này để tránh sự lẫn lộn có thể với lớp điều khiển truy nhập truyền thông của giao thức MAC trong OSI) của người nhận và người gửi chia sẻ một khóa sử dụng bởi người gửi để phát sinh MIC cũng như bởi người nhận để xác nhận sự toàn vẹn của thông báo và sự nhận biết người gửi.

Để tránh “những sự tấn công” trong đó một người nghe trộm ghi một thông báo và gửi lại thì một máy đếm thông báo hoặc thiết bị bấm giờ trong tính toán của MIC. Với cách này không có hai thông báo xác thực nào chứa đựng cùng dữ liệu, nếu có chúng sẽ bị đồng nhất.

1.3.6 Truyền thông

Một chỉ tiêu đánh giá cho bất kì WSN nào là tốc độ truyền dữ liệu, năng lượng tiêu thụ và khoảng cách. Trong khi độ bao phủ của mạng không dây bị giới hạn bởi khoảng cách truyền của các nút riêng biệt, khoảng cách truyền có một ảnh

hưởng quan trọng tới mật độ tối thiểu có thể chấp nhận được. Nếu các nút được đặt rất xa nó không thể tạo được kết nối với mạng liên kết hoặc một nút dự trữ để có được độ tin cậy cao. Nếu khoảng cách truyền radio thoả mã mật độ nút cao, các nút thêm vào sẽ làm tăng mật độ hệ thống tới một mức độ nào đó cho phép. Tốc độ truyền cũng có ảnh hưởng lớn đến hiệu suất của mạng. Tốc độ truyền cao hơn làm cho khả năng lấy mẫu hiệu quả hơn và năng lượng tiêu thụ của mạng ít hơn. Khi tốc độ tăng, việc truyền mất ít thời gian hơn do đó đòi hỏi ít năng lượng hơn. Tuy nhiên khi tăng tốc độ cũng thường làm tăng năng lượng tiêu thụ radio.

Tuy những mục đích thiết kế tốc độ dữ liệu cực đại có thể là 512b/s (64bytes/s) mặc dù tốc độ dữ liệu thường dưới tốc độ này ở khoảng 1b/s hoặc thấp hơn trong một vài ứng dụng.

1.3.7 Tính toán

Hai việc tính toán cho nút mạng tập trung chủ yếu vào xử lý dữ liệu nội mạng và quản lý các giao thức truyền thông không dây mức thấp. Có những yêu cầu giới hạn về mặt thời gian thực đối với truyền thông và cảm biến. Khi dữ liệu tới trên mạng CPU cần điều khiển đồng thời radio và ghi lại/giải mã dữ liệu tới. Tốc độ truyền cao hơn đòi hỏi tính toán nhanh hơn. Điều tương tự cũng đúng với xử lý dữ liệu cảm biến. Các cảm biến tương tự có thể phát ra hàng ngàn mẫu trong một giây. Các thao tác xử lý cảm biến nói chung bao gồm : lọc số, trung bình hóa, nhận biết ngưỡng, phân tích phổ...Để tăng khả năng xử lý cục bộ các nút láng giềng có thể kết hợp dữ liệu với nhau trước khi truyền đi trên mạng. Các kết quả từ nhiều nút mạng có thể được tổng hợp cùng nhau. Xử lý nội mạng này đòi hỏi thêm tài nguyên tính toán. Ngoài ra, ứng dụng xử lý dữ liệu có thể tiêu thụ một lượng tính toán phụ thuộc vào các phép toán được thực hiện.

1.3.8 Đồng bộ thời gian

Để hỗ trợ sự tương quan thời gian đọc cảm biến và chu trình hoạt động ngắn của ứng dụng thu thập dữ liệu, các nút cần duy trì đồng bộ thời gian chính xác với các thành viên khác trong mạng. Các nút cần ngủ và thức dậy cùng nhau để chúng có thể định kỳ truyền thông cho nhau. Các lỗi trong cơ chế tính toán thời gian sẽ tạo lên sự không hiệu quả dẫn đến làm tăng chu trình làm việc. Trong các hệ phân tán, việc trôi clock theo thời gian là do có chế tính toán thời gian không chính xác, phụ thuộc vào nhiệt độ, điện áp, độ ẩm, thời gian dựa theo dao động sẽ không như nhau. Cần có cơ chế đồng bộ hóa cao để bù lại những sự không chính xác như vậy.

1.4 Kiến trúc của mạng WSN

Trong một mạng phẳng hoàn toàn không có cấu trúc lôgic, tất cả các nút phải hợp tác để điều khiển mạng - xác định việc tạo thành và mất mối liên kết, hợp và tách nút là giá của truyền thông vì những nút chỉ trực tiếp ý thức được môi trường mạng trong vùng lân cận tức thời của chúng.

1.4.1 Kiến trúc nút mạng

Tương tự như kiến trúc của máy tính thông thường, các thành phần chính của kiến trúc vật lý của các nút mạng cảm nhận không dây có thể được phân loại vào 4 nhóm chính: Bộ vi xử lý, bộ lưu trữ, bộ truyền thông, và bộ cảm nhận, bộ khởi động

1.4.1.1 Bộ vi xử lý

Có 2 điều ràng buộc cho các thành phần xử lý là năng lượng và giá. Thực chất tất cả các bộ xử lý WSN hiện thời đều đã được sử dụng phổ biến nhờ vào sự phát triển về mặt công nghệ. Một khi sự xử lý trong một nút phải hướng vào một sự đa dạng của các tác vụ khác nhau, nhiều nút có vài kiểu của bộ xử lý.

1.4.1.2 Bộ lưu trữ

Hiện nay, các nút cảm nhận có những thành phần lưu trữ tương đối nhỏ. Chúng thường sử dụng bộ nhớ DRAM và Flash. Khi việc truyền thông chính là thành phần tiêu thụ năng lượng chính của mạng cảm nhận không dây, chúng ta mong rằng số lượng lưu trữ tại mỗi nút mạng sẽ tiếp tục tăng lên

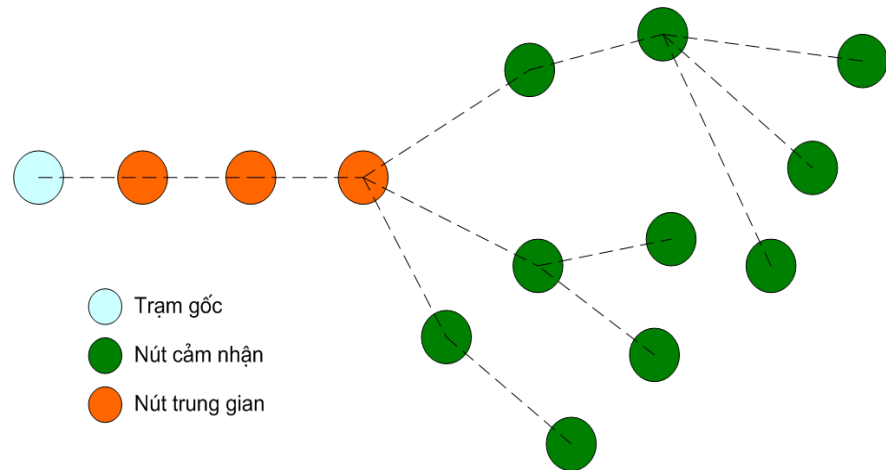
1.4.1.3 Bộ truyền thông

Mô hình truyền thông thường được đề cập cho các thế hệ mạng cảm nhận không dây hiện thời là việc truyền thông đa bước. Một vài kết quả hiện thời chỉ ra rằng việc truyền thông đa bước có khả năng co dẫn rất tốt và có thể làm giảm đáng kể năng lượng tiêu thụ trong các mạng cảm nhận lớn. Điều cần đặc biệt chú ý tại đây chính là việc lắng nghe thường yêu cầu năng lượng ngang bằng với việc truyền thông.

1.4.1.4 Bộ cảm nhận, bộ khởi động

Có thể hình dung các bộ cảm nhận như là đôi mắt của mạng cảm nhận, và bộ khởi động như là cơ bắp của nó. Công nghệ MEMS đã giúp việc xử lý ổn định, và các kết quả thu được rất đáng được quan tâm.

Có 3 loại nút mạng cần chú ý trong mạng cảm nhận không dây đa bước:



Hình 1.2: Cấu trúc mạng cảm nhận không dây

- Trạm gốc
- Nút cảm nhận
- Nút chuyển tiếp

Trong đó các nút cảm nhận có gắn các đầu đo vừa trực tiếp đo số liệu và truyền về trạm gốc, vừa đóng vai trò nút chuyển tiếp dữ liệu nhận được từ các nút con khác gửi về nút cha.

Để thiết kế hiệu quả nút mạng nhằm xây dựng một mạng cảm nhận không dây đa bước, chúng ta sẽ đề xuất việc sử dụng chip CC1010 trong vai trò là nút mạng trong mạng cảm nhận không dây.

1.4.2 Kiến trúc mạng

Nói rõ hơn về điều này, chúng ta cùng xem xét kỹ hơn về các dạng kiến trúc mạng được đề cập cho mạng cảm nhận không dây:

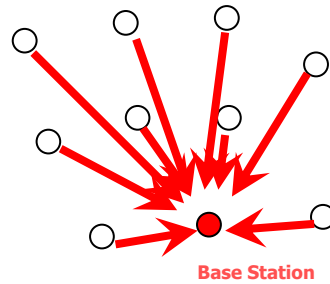
- Mạng đơn: Nơi tất cả các nút liên kết trực tiếp với trạm gốc.
- Mạng liên kết bước: Các nút truyền thông gián tiếp tới trạm gốc qua các nút trung gian, nhằm giảm năng lượng tải cho các nút.
- Mạng liên kết bó: Nhóm các nút vào trong các bó, tập hợp dữ liệu và đánh dấu một nút giữ vai trò truyền thông với trạm gốc.

1.4.2.1 Mạng đơn

* Đặc điểm

Đây là kiến trúc mạng đơn giản nhất, mà trong đó tất cả các nút cảm nhận trong mạng truyền thông trực tiếp tới trạm gốc. Với phạm vi truyền thông có hạn của các nút mạng cảm ứng thường là hàng chục đến trăm mét thì kiến trúc mạng

đơn khó có thể mở rộng được, do đó ta có thể áp dụng kiến trúc mạng đơn đối với các mạng làm việc trong phạm vi nhỏ cần khả năng truyền thông nhanh.



Hình 1.2: Kiến trúc mạng đơn

*** Ưu điểm**

- Đơn giản, dễ cấu hình và thực hiện.
- Tốc độ thực hiện nhanh.

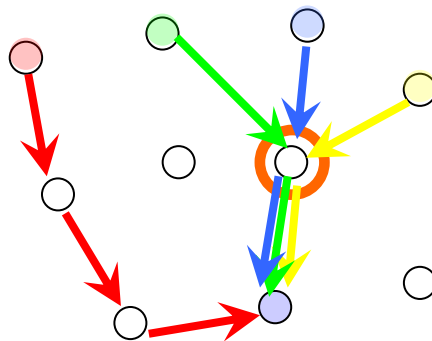
*** Nhược điểm**

- Khó xác định nút truyền thông kế tiếp.
- Nhanh tiêu hao năng lượng tại các nút (đặc biệt là các nút xa trạm gốc).

1.4.2.2 Mạng liên kết bước

*** Đặc điểm**

Các nút mạng trong kiến trúc mạng liên kết bước vừa đóng vai trò nút cảm nhận – thu thập thông tin môi trường, vừa đóng vai trò nút mạng – có thể chuyển tiếp dữ liệu thu được từ các nút mạng khác. Trong suốt giai đoạn thiết lập, mỗi nút tìm cho mình một lộ trình tốt nhất đi qua các chặng tới trạm gốc qua một hoặc nhiều trạm trung gian. Các nút sử dụng giao thức CSMA để tránh xung đột, các gói dữ liệu sẽ được truyền sau mỗi khoảng $t_{\text{delay}}(s)$. Khi một nút hết năng lượng, quá trình thiết lập lộ trình tự động làm việc để duy trì kết nối tới trạm gốc.



Hình 1.3 Kiến trúc mạng liên kết bước

*** Ưu điểm**

- Tiết kiệm năng lượng truyền tải từ các nút tới trạm gốc
- Nhanh chóng, tiện dụng
- Dễ dàng mở rộng kích thước mạng

*** Nhược điểm**

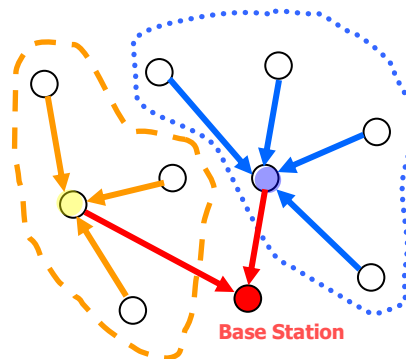
- Dữ liệu truyền qua nhiều trạm dẫn đến tiềm ẩn an ninh cao
- Dữ liệu tới nút trung gian liên tục khiến năng lượng nút trung gian tiêu hao nhanh chóng so với các nút thường

1.4.2.3 Mạng liên kết bó

*** Đặc điểm**

Trong mạng tổ chức thành một tập hợp của những bó các nút, mỗi nút thuộc về ít nhất một bó. Mỗi bó có tiêu đề bó hành động như một điều khiển cục bộ cho những nút bên trong bó. Những cổng vào của các nút cung cấp truyền thông giữa các bó. Quá trình xếp nhóm được áp dụng để hình thành một sự phân cấp của các bó.

Những nút trong bó thực hiện một giải thuật để chọn nút đầu bó và các nút thành viên khác. Tất cả các nút thành viên trong bó truyền dữ liệu của chúng tới nút đầu bó, sử dụng giao thức TDMA điều khiển truy cập. Trong khi đó nút đầu bó nhận dữ liệu từ tất cả các thành viên trong bó, thực hiện xử lý các chức năng trên dữ liệu (ví dụ tập hợp dữ liệu), và truyền dữ liệu tới trạm gốc.



Hình 1.4: Kiến trúc mạng liên kết bó

*** Ưu điểm**

- Tiết kiệm năng lượng truyền tải từ các nút tới trạm gốc
- Nhanh chóng, tiện dụng
- Dễ dàng mở rộng kích thước mạng

*** Nhược điểm**

Năng lượng tại các nút đầu bó tiêu hao nhanh chóng, có thể dẫn tới hoạt động của bó tạm dừng.

1.4.2.4 Đánh giá

Ta có thể nhận xét về các dạng kiến trúc mạng qua bảng sau:

Đặc điểm	Mạng đơn	Mạng LK bước	Mạng LK bó
Tốc độ tiêu thụ năng lượng	Rất nhanh	Nhanh	Chậm hơn (nhưng nút đầu bó ngược lại)
Điều khiển truy cập	Không đề cập	CSMA	TDMA
Tác động khi 1 nút ngừng hoạt động	Một nút ngừng hoạt động	Một nút ngừng (sau khi xác lập lại lộ trình)	Toàn bộ bó ngừng hoạt động (nếu là đầu bó)
Tiềm ẩn	One hop	Multiple hops	Two hops
Phạm vi làm việc	Nhỏ	Rộng hơn	Rộng nhất

Qua 3 dạng kiến trúc trên thì dạng kiến trúc bó tiêu thụ năng lượng thấp nhất, nhưng đi kèm với nó là những nguy cơ về sự tiềm ẩn lẫn sự tác động cao. Đối với các mạng cảm nhận không dây thì yêu cầu về thời gian hoạt động của mạng đóng vai trò hết sức quan trọng, nên việc xây dựng một kiến trúc mạng có khả năng làm việc hiệu quả lẫn tiêu hao năng lượng thấp là rất cần thiết. Do đó chúng ta có thể phát triển dựa trên ý tưởng của mạng liên kết bó xây dựng một kiến trúc mạng phù hợp hơn. Trước hết ta cần quan tâm xem xét kỹ hơn về kiến trúc mạng liên kết bó.

CHƯƠNG 2: ĐỊNH TUYẾN TRONG MẠNG CẢM NHẬN KHÔNG DÂY

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 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 mà 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 việc tập hợp dữ liệu thông thường dựa trên vị trí.
- Khả năng dư thừa dữ liệu rất cao vì các nút cảm biến thu lượm dữ liệu dựa trên hiện tượng chung.

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:

2.3.1 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 khắc 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.

2.3.2 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 multihop 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.

2.3.3 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.

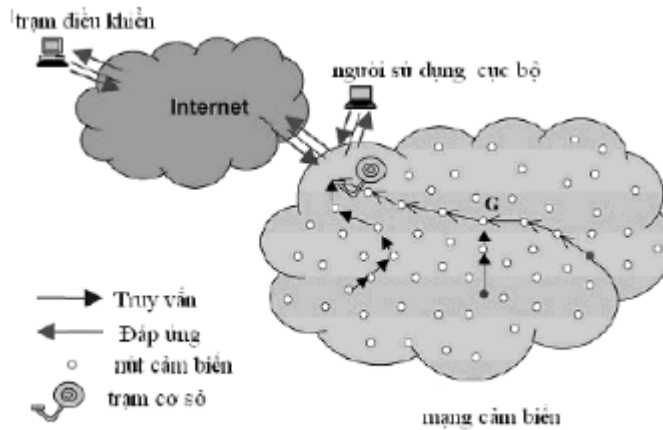
2.3.4 Cách truyền dữ liệu

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 2.1 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.

2.4 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 mào 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.

Bảng 2.1 Phân loại và so sánh các giao thức chọn đường trong WSN

Giao thức chọn đường	Giao thức trung tâm dữ liệu	Giao thức phân cấp	Giao thức dựa trên vị trí	Phân loại	Di chuyển	Dựa vào hình ảnh	Kết hợp số liệu	Xác định vị trí	QoS	Độ phức tạp của trạng thái	Khả năng định cỡ	Đa đường	Dựa vào yêu cầu
SPIN	x			Ngang hàng	Có thể	Có	Có	Không	Không	Thấp	Hạn chế	Có	Có
Directed Diffusion	x			Ngang hàng	Hạn chế	Có	Có	Có	Không	Thấp	Hạn chế	Có	Có
Rumor	x			Ngang hàng	Rất hạn chế	Không	Có	Không	Không	Thấp	Tốt	Không	Có
GBR	x			Ngang hàng	Hạn chế	Không	Có	Không	Không	Thấp	Tốt	Không	Có
CADR	x			Ngang hàng	Không	Không	Có	Không	Không	Thấp	Hạn chế	Không	Không
COUGAR	x			Ngang hàng	Không	Không	Có	Không	Không	Thấp	Hạn chế	Không	Có
ACQUIRE	x			Ngang hàng	Hạn chế	Không	Có	Không	Không	Thấp	Hạn chế	Không	Có
LEACH		X		Phân cấp	Nút gốc cố định	Không	Có	Có	Không	Nút chủ nhóm	Tốt	Không	Không
TBBN& APTBBN	x	X		Phân cấp	Nút gốc cố định	Không	Có	Có	Không	Nút chủ nhóm	Tốt	Không	Không
PEGASIS		X		Phân cấp	Nút gốc cố định	Không	Không	Có	Không	Nút chủ nhóm	Tốt	Không	Không
MBCN& SMBCN		X		Phân cấp	Không	Không	Không	Không	Không	Thấp	Thấp	Không	Không
GAF		X	x	Dựa theo vị trí	Dựa theo vị trí	Không	Không	Không	Không	Thấp	Tốt	Không	Không
GBAR			x	Dựa theo vị trí	Không	Không	Không	Không	Không	Thấp	Hạn chế	Không	Không
SAR			x	Dựa theo vị trí	Không	Có	Có	Không	Có	Trung bình	Hạn chế	Không	Có
SPBBD			x	Dựa theo QoS	Không	Không	Không	Không	Có	Trung bình	Hạn chế	Không	Có

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 multicaasting, geo-casting

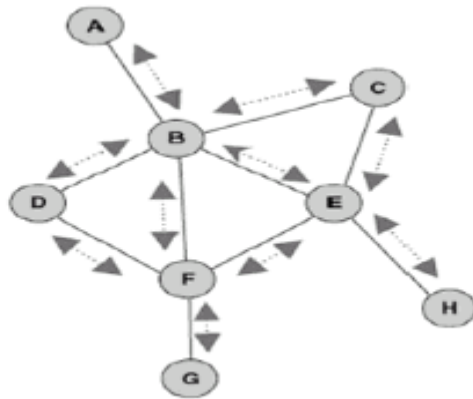
Loại giao thức thứ 4 là dựa vào vị trí để đánh địa chỉ cho các nút cảm biến. lại giao thức này rất có ích cho những ứng dụng nơi mà vị trí của các nút cảm biến trong vùng địa lý được bao phủ bởi mạng liên quan đến truy vấn được đưa ra bởi nút nguồn.

2.5 Giao thức trung tâm dữ liệu

2.5.1 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 2.2 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 3 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

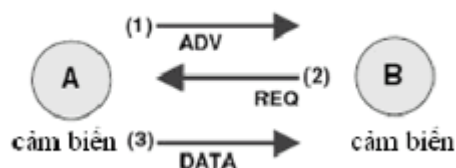
tượng bản tin kép tuy nhiên có nhược điểm là có thể gói sẽ không bao giờ đến được đích.

2.5.2 SPIN

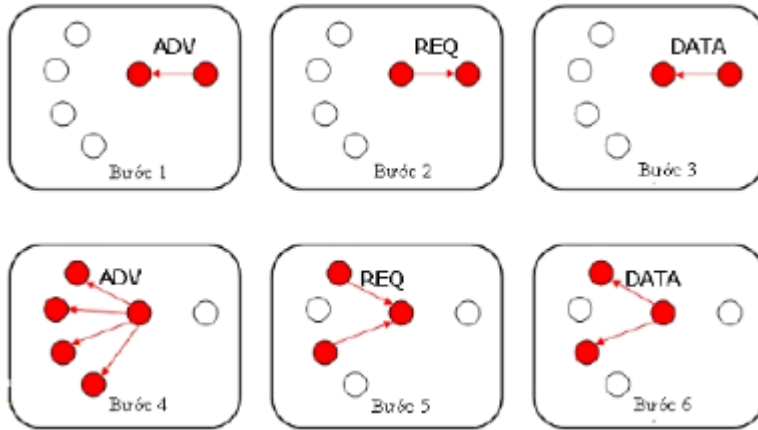
SPIN (Sensor Protocol for Information via Negotiation) là giao thức định tuyến thông tin dựa trên sự đàm phán dữ liệu. Mục tiêu chính của giao thức này đó là tập trung việc quan sát môi trường có hiệu quả bằng một số các nút cảm biến riêng biệt trong toàn bộ mạng. Nguyên lý của giao thức này đó là sự thích ứng về tài nguyên và sắp xếp dữ liệu. Ý nghĩa của việc đàm phán 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 2.3 Ba tín hiệu bắt tay của SPIN



Hình 2.4 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 tin 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, và bước 6 là 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.

2.5.3 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 mỗi quan tâm (interest) bằng một cặp thuộc tính-giá trị.

Bảng (2.2) 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 (2.5). Với mỗi nhiệm vụ cảm biến tích cực, sink sẽ gửi quảng bá bản tin interest theo chu kỳ cho các nút lân cận. 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.

Bảng 2.2 Miêu tả interest sử dụng cặp thuộc tính-giá trị

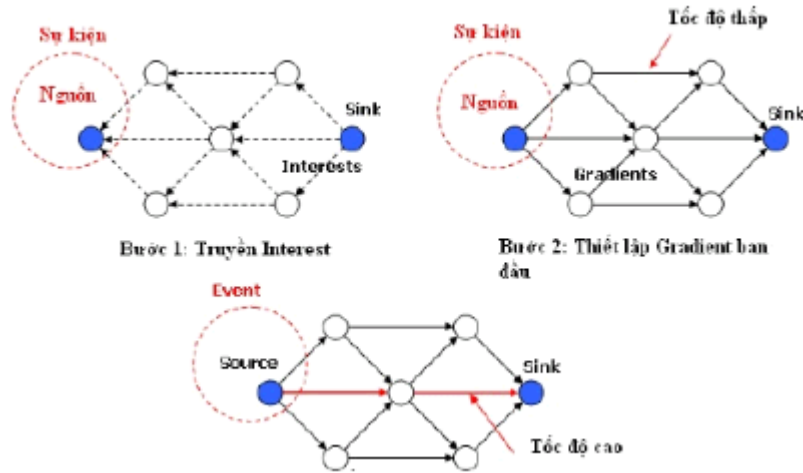
Cặp thuộc tính-giá trị	Miêu tả
Type = chim ruồi	Phát hiện vị trí của chim ruồi
Interval=20ms	Báo cáo sự kiện chu kỳ 20ms
Duration=10s	Thời gian sống của Interest
Field=[(x ₁ ,x ₂),(y ₁ ,y ₂)]	Báo cáo từ các con cảm biến trong vùng

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 lối 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 2.5 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.

2.6 Giao thức phân cấp

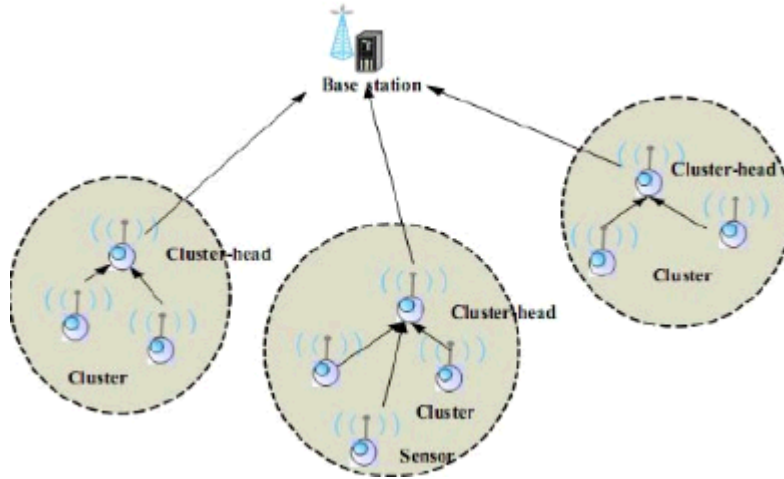
2.6.1 LEACH

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 2.6 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:

$$T(n) = \frac{p}{1 - p * (r \bmod l / p)} \quad \text{nếu } n \in G$$
$$T(n) = 0 \quad \text{còn lại}$$

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 thay 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.

2.6.2 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 xa trong chuỗi. Hơn nữa ở nút chính có thể xảy ra hiện tượng thất cổ chai.

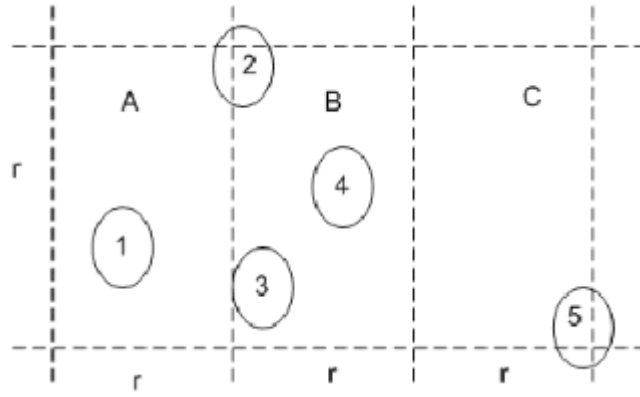
2.7 Giao thức dựa trên vị trí

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:

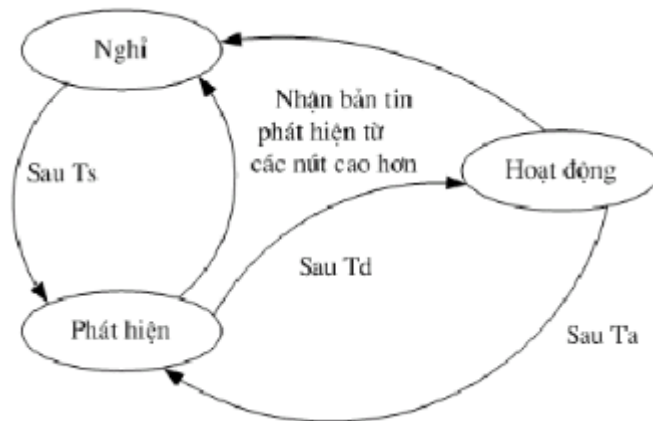
2.7.1 GAF

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 (2.7). Trong hình vẽ này, nút 1 có thể truyền đến bất kỳ 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 song được tắt đi. Sự chuyển trạng thái trong GAF được miêu tả ở hình (2.8). 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 2.7. Ví dụ về lưới ảo 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 2.8 Sự chuyển trạng thái trong GAF

GAF cố gắng giữ mạng hoạt động 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.

2.7.2 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:

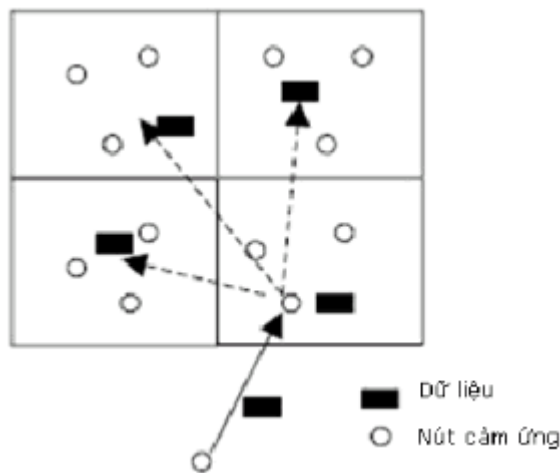
Chuyển tiếp gói đến vùng đích: GEAR dùng cách tự chọn nút lân cận dựa trên sự nhận biết về năng lượng và vị trí địa lý để định tuyến gói đến vùng đích. Có 2 trường hợp cần quan tâm: khi tồn tại nhiều hơn một nút lân cận gần hơn so với đích: GEAR sẽ chọn hop tiếp theo trong số tất cả các nút lân cận gần 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 bảncopy 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 (2.9).

Để thỏa mãn các điều kiện chúng ta dùng giải thuật chuyển tiếp địa lý để qui để truyền gói trong vùng này. Tuy nhiên, với những vùng mật độ thấp, chuyển tiếp địa lý để 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.



Hình 2.9. Chuyển tiếp địa lý để quy trong GEAR

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: MÔ PHỎNG PEGASIS BẰNG MOBILITY FRAMEWORK CỦA OMNeT++

3.1 Giới thiệu về OMNeT++ và Mobility Framework

3.1.1 Giới thiệu về OMNeT++

OMNeT++ là viết tắt của cụm từ Objective Modular Network Testbed in C++. OMNeT++ là một ứng dụng cung cấp cho người sử dụng môi trường để tiến hành mô phỏng hoạt động của mạng. Mục đích chính của ứng dụng là mô phỏng hoạt động mạng thông tin, tuy nhiên do tính phổ cập và linh hoạt của nó, OMNeT++ còn được sử dụng trong nhiều lĩnh vực khác như mô phỏng các hệ thống thông tin phức tạp, các mạng kiểu hàng đợi (queueing networks) hay các kiến trúc phần cứng... OMNeT++ cung cấp sẵn các thành phần tương ứng với các mô hình thực tế. Các thành phần này (còn được gọi là các module) được lập trình theo ngôn ngữ C++, sau đó được tập hợp lại thành những thành phần hay những mô hình lớn hơn bằng một ngôn ngữ bậc cao (NED). OMNeT++ hỗ trợ giao diện đồ họa, tương ứng với các mô hình cấu trúc của nó đồng thời phân nhân mô phỏng (simulation kernel) và các module của OMNeT++ cũng rất dễ dàng nhúng vào trong các ứng dụng khác.

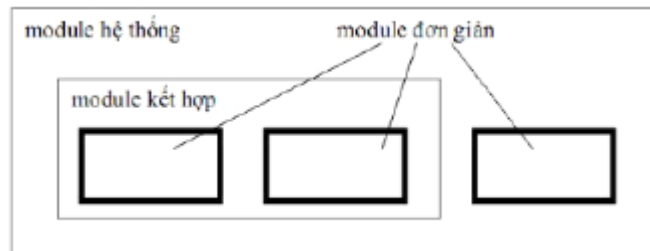
Một mô hình trong OMNeT++ bao gồm các module lồng nhau có cấu trúc phân cấp. Độ sâu của các module lồng nhau là không giới hạn, điều này cho phép người sử dụng có thể biểu diễn các cấu trúc logic của các hệ thống trong thực tế bằng các cấu trúc mô hình. Các module trao đổi thông tin với nhau thông qua việc gửi các message (message). Các message này có thể có cấu trúc phức tạp tùy ý. Các module có thể gửi các message này theo hai cách, một là gửi trực tiếp tới địa chỉ nhận, hai là gửi đi theo một đường dẫn được định sẵn, thông qua các cổng và các kết nối.

Các module có thể có các tham số của riêng nó. Các tham số này có thể được sử dụng để chỉnh sửa các thuộc tính của module và để biểu diễn cho topology của mô hình. Các module ở mức thấp nhất trong cấu trúc phân cấp đóng gói các thuộc tính.

Các module này được coi là các module đơn giản, và chúng được lập trình trong ngôn ngữ C++ bằng cách sử dụng các thư viện mô phỏng.

Cấu trúc phân cấp của các module

Một mô hình trong OMNeT++ chứa các module lồng nhau có cấu trúc phân cấp, trao đổi thông tin với nhau bằng cách gửi các message. Cấu trúc của mô hình có thể được mô tả bằng ngôn ngữ NED của OMNeT++.



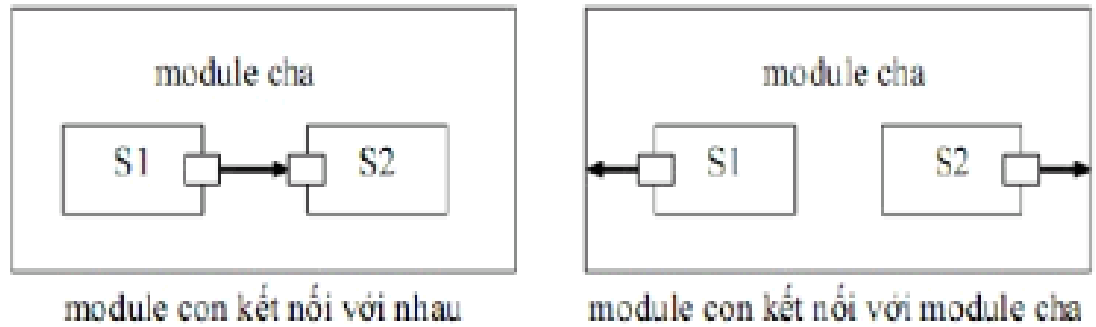
Hình 3.1 Cấu trúc phân cấp module trong OMNeT++

Các module có thể chứa nhiều module con và được gọi là module kết hợp. Các module đơn giản là các module có cấp thấp nhất trong cấu trúc phân cấp. Các module đơn giản chứa các thuật toán của mô hình. Người sử dụng triển khai các module đơn giản bằng ngôn ngữ C++, sử dụng các thư viện mô phỏng của OMNeT++. Message, cổng, liên kết

Các module trao đổi thông tin bằng việc gửi các message. Trong thực tế, message có dạng khung (frame) hoặc là các gói tin (packet) được truyền đi trong mạng. Các message có thể có cấu trúc phức tạp tùy ý. Các module đơn giản có thể gửi các message đi một cách trực tiếp đến vị trí nhận hoặc gửi đi theo một đường dẫn định sẵn thông qua các cổng và các liên kết.

“Thời gian mô phỏng địa phương” (local simulation time) của một module tăng lên khi module nhận được một message. Message có thể đến từ một module khác hoặc đến từ cùng một module (message của chính bản thân module - self-message được dùng để thực hiện bộ định thời). Cổng (gate) là các giao tiếp vào ra của module

Message được gửi đi qua các cổng ra và được nhận vào thông qua các cổng vào. Mỗi kết nối (connection) hay còn gọi là liên kết (link) được tạo bên trong một mức đơn trong cấu trúc phân cấp của các module: bên trong một module kết hợp, một kết nối có thể được tạo ra giữa các cổng tương ứng của hai module con, hoặc giữa cổng của module con với cổng của module kết hợp.



Hình 3.2 Các kết nối trong OMNeT++

Mô hình truyền gói tin:

Một kết nối có thể có ba tham số đặc trưng. Những tham số này rất thuận tiện cho các mô hình mô phỏng mạng thông tin nhưng không hữu dụng lắm cho các kiểu mô hình khác. Ba tham số này bao gồm:

- Độ trễ đường truyền (propagation delay) tính bằng s - giây.
- Tỉ số lỗi bit, được tính bằng số lỗi/bit.
- Tỉ số dữ liệu, được tính bằng số bit/s.

Các tham số này là tùy chọn. Giá trị của các tham số này là khác nhau trên từng kết nối, phụ thuộc vào kiểu của liên kết (hay còn gọi là kiểu của kênh truyền - channel type).

Độ trễ đường truyền là tổng thời gian đến của message bị trễ đi khi truyền qua kênh. Tỉ số lỗi bit ảnh hưởng đến quá trình truyền message qua kênh. Tỉ số này là xác suất các bit bị truyền sai. Do đó xác suất để một message độ dài n bit truyền đi chính xác là: $P(\text{message gửi đi được nhận chính xác}) = (1 - \text{ber})^n$

Trong đó ber là tỉ số lỗi bit và n là số bit của message.

Các message truyền đi đều có một cờ lỗi, cờ này sẽ được thiết lập khi việc truyền message có lỗi. Tỉ số dữ liệu được tính theo đơn vị bit/s, và nó được sử dụng để tính thời gian để truyền một gói tin. Khi tỉ số này được sử dụng, quá trình gửi message đi trong mô hình sẽ tương ứng với việc truyền bit đầu tiên và message được tính là đến nơi sau khi bên nhận đã nhận được bit cuối cùng.

Xây dựng và chạy thử các mô hình mô phỏng

Một mô hình OMNeT++ bao gồm những phần sau:

- Ngôn ngữ mô tả topology - NED (file có phần mở rộng .ned): mô tả cấu trúc của module với các tham số, các cổng... Các file .ned có thể được viết bằng bất kỳ bộ soạn thảo hoặc sử dụng chương trình GNED có trong OMNeT++.

- Định nghĩa cấu trúc của các message (các file có phần mở rộng .msg): Người sử dụng có thể định nghĩa rất nhiều kiểu message và thêm các trường dữ liệu cho chúng. OMNeT++ sẽ dịch những định nghĩa này sang các lớp C++ đầy đủ.

- Mã nguồn của các module đơn giản. Đây là các file C++ với phần mở rộng là .h hoặc .cc.

Quá trình tiếp theo giống như biên dịch mã nguồn C/C++:

- Trong Linux: các file .cc liên kết thành file .o.

- Trong Windows: các file .cpp liên kết thành file .obj.

Sau đó, tất cả các file trên sẽ được liên kết (link) với các thư viện cần thiết để tạo thành file.exe.

Chạy các ứng dụng mô phỏng bằng OMNeT++:

Ta thực hiện đánh hai lệnh sau: `opp_nmake` để tạo ra `Makefile.vc`. Để tạo file chạy ta gõ: `nmake -f Makefile.vc`.

3.1.2 Giới thiệu về Mobility

Mobility là frame work được xây dựng trên OMNeT++ để nhằm hỗ trợ mô phỏng mạng vô tuyến và di động. Phần lõi của frame work hỗ trợ di động của các nút, quản lý kết nối động và mô hình kênh vô tuyến. Thêm vào đó, phần lõi này còn cung cấp các module cơ bản mà ta có thể dựa vào đó mà xây dựng module của mình. Với khái niệm này người lập trình dễ dàng mô phỏng được giao thức của mình dựa trên Mobility frame work.

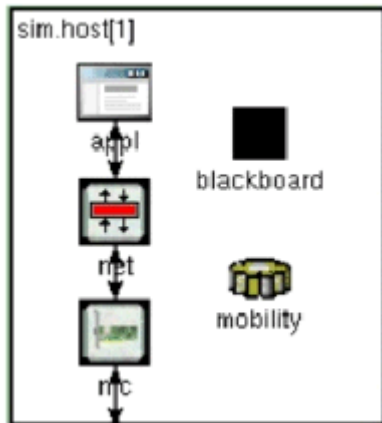
Frame work này chủ yếu được dùng để mô phỏng:

- Mạng vô tuyến cố định.
- Mạng vô tuyến di động.
- Mạng tập trung và phân bố.
- Mạng cảm biến.
- Mạng vô tuyến đa kênh.
- Nhiều sự mô phỏng khác mà cần hỗ trợ di động và giao diện vô tuyến.

Các khái niệm cơ bản của Mobility FW:

Channel control: Hỗ trợ di động và quản lý kết nối động. Điều khiển và duy trì các kết nối tiềm ẩn giữa các host trong mạng.

Host: Cấu trúc của một host di động được miêu tả như hình vẽ dưới đây. Host là module kết hợp gồm có các lớp (chính là các module) cơ bản sau: lớp ứng dụng appl, lớp mạng Net, lớp Nic. Trong đó module nic bao gồm các module con là: Mac, decider, snrEval. Tương ứng với các lớp là các module được tạo ra trong thư mục / template.



Hình 3.3 Cấu trúc của host di động

Cấu trúc này cũng dựa trên một phần của tiêu chuẩn ISO/OSI. Tuy nhiên host còn có thêm hai module nữa là: blackboard và mobility. Mobility cung cấp thông tin về vị trí địa lý và điều khiển sự di chuyển của host. Blackboard được sử dụng qua các lớp giao tiếp. Nó cung cấp thông tin qua một hay nhiều lớp và rất có ích khi ta đánh giá giao thức.

Cài đặt Mobility FW:

Ban đầu bạn down Mobility FW trên trang web của OMNeT++ và để chúng vào thư mục cài đặt OMNeT++. Sau đó bạn làm theo các bước sau:

Sửa biến OMNETPP_ROOT trong file mkmk.cmd chỉ đến thư mục bạn cài OMNeT++.

Chạy mkmk.cmd

Sau đó bạn cần liên kết các thư viện của Mobility bằng cách vào cmd chỉ đến thư mục Mobility chạy lệnh sau: `nmake /f Makefile.vc core_dir && nmake /f Makefile.vc contrib_dir && nmake /f Makefile.vc networks_dir`

Đến đây thì bạn có thể sử dụng được Mobility và sẵn sàng xây dựng một chương trình của chính mình. Mobility FM cung cấp các module cơ bản tương ứng

với các lớp của host. Module cung cấp các khai báo hàm cần thiết, các hướng dẫn cơ bản để tạo module của chính bạn. Hoặc bạn có thể tham khảo các module trong thư mục /core hoặc /contric. Bạn cũng có thể kế thừa các Module này.

Sau khi xây dựng các module cần thiết cho mô phỏng, bạn cần copy file Makefile.gen.vc vào thư mục Yournetwork mà bạn tạo ra, thay đổi biến MOBFW chỉ đến đường dẫn của mobility. Và cuối cùng bạn vẫn phải tạo file Omnetpp.ini để thiết lập các tham số đầu vào cho mô phỏng.

Mobility chỉ cung cấp các module cần thiết cho quá trình mô phỏng. Tuy nhiên nếu bạn cũng có thể tạo ra một module mới mà phù hợp với mục đích của bạn. Module mới được tạo ra phải bao gồm 3 file: a.cc, a.h, a.ned.

Module cơ bản

Module cơ bản sử dụng khởi tạo nhiều giai đoạn của OMNeT++ cho nên các module của MF khởi tạo bằng hai giai đoạn. Trong trường hợp bạn sử dụng Blackboard thì bạn phải public ở giai đoạn 0 và subscribe ở giai đoạn 1. Điều này để tránh trường hợp việc miêu tả một tham số trước khi tham số ấy được public

Trong các file ở phần template có chứa hàm initialize(int) của module đơn giản. Bạn không được xóa dòng này vì nó được coi như là điều bắt buộc của giai đoạn khởi tạo module cơ bản.

Khái niệm địa chỉ

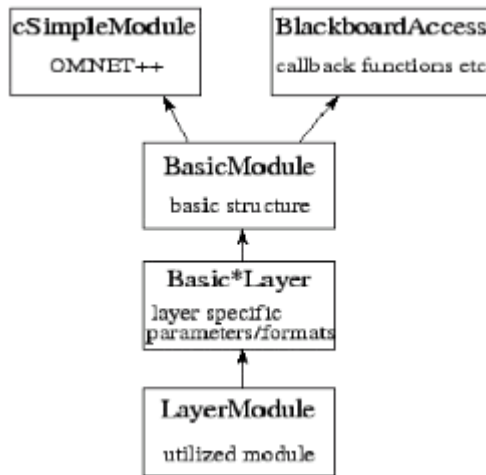
Chúng ta sử dụng id() của module để đánh địa chỉ trong MF.Id() của lớp Nic được dùng để đánh địa chỉ lớp Mac và id() của lớp Net được dùng để đánh địa chỉ lớp mạng và lớp ứng dụng. Để lấy được địa chỉ của các module này MF cung cấp các hàm cơ bản để lấy địa chỉ: myAppAddr(), myNetwAddr() và myMacAddr() đối với các lớp BasicAppLayer, BasicNetwLayer và BasicMacLayer tương ứng. Hàm netw2mac() của module Channel Control cung cấp giao thức ARP đơn giản để chuyển đổi địa chỉ giữa lớp mạng và lớp Mac. Tuy nhiên bạn cũng có thể định nghĩa lại hàm getMacAddr() của lớp BasicNetwLayer để thực hiện chức năng này.

Quy ước đặt tên: Có một số các quy ước đặt tên trong các module file.ned và bạn phải tuân theo.

Tất cả các module của host phải chứa từ Host hoặc host trong tên của module. Ví dụ: BaseStationHost hoặc MobileHost. Ngoài ra còn có một số module ned bạn không thể thay đổi tên được như channelcontrol, blackboard, net, phy, snrEval.

Module *Basic

MF cung cấp cho mỗi lớp một module tương ứng lấy từ lớp module cơ bản. Bạn cũng có thể mở rộng các module này. Cấu trúc kế thừa module của MF được thể hiện như hình vẽ.



Hình 3.4 Cấu trúc kế thừa module trong MF

NetwPkt	LỚP MẠNG	
	<i>destAddr</i>	Địa chỉ đích lớp mạng
	<i>srcAddr</i>	Địa chỉ nguồn lớp mạng
	<i>seqNum</i>	Chuỗi số
	<i>ttl</i>	Thời gian sống
	<i>MacControlInfo</i>	Lớp thông tin điều khiển để nói cho lớp Mac biết địa chỉ của next hop tiếp theo

AppIPkt	LỚP ỨNG DỤNG	
	<i>destAddr</i>	Địa chỉ đích lớp ứng dụng
	<i>srcAddr</i>	Địa chỉ nguồn lớp ứng dụng

Khái niệm bản tin: Để cung cấp các chức năng cơ bản như là đóng gói (encapsulate) và dỡ gói (decapsulate) trong module*Basic chúng ta cần thiết phải có những header cho mỗi lớp tương ứng. Những bản tin này có những trường bắt buộc và quan trọng cần thiết cho mô phỏng. Bạn chỉ có thể mở rộng các loại bản tin này. Sau đây là một số loại bản tin tương ứng với các lớp:

Bảng 3.1 Các loại bản tin tương ứng của các lớp

<i>AirFrame</i>	LỚP VẬT LÝ	
	<i>pSend</i>	Công suất gửi
	<i>channelId</i>	Kênh mà bản tin được truyền
	<i>id</i>	id của bộ khởi tạo để lấy vị trí
	<i>duration</i>	Thời gian cần thiết để gửi bản tin trong kênh
	<i>SrcControlInfo</i>	Lớp thông tin điều khiển được sử dụng để truyền thông tin Sn()r đến Decider

MacPkt	LỚP MAC	
	<i>destAddr</i>	Địa chỉ đích Mac
	<i>srcAddr</i>	Địa chỉ nguồn Mac
	<i>channelID</i>	Kênh mà bản tin được gửi

Blackboard

Khi bạn muốn đánh giá hoạt động một giao thức, bạn cần thiết phải biết thông tin về trạng thái bên trong một giao thức. Khi đó bạn cần sử dụng Blackboard.

Sự thay đổi trạng thái được publish trên Blackboard, và đơn giản bạn chỉ cần đăng ký tham số đó với Blackboard và sau đó truy cập đến tham số này.

Ngoài ra Blackboard còn cho phép bạn trao đổi thông tin giữa các lớp mà không cần sử dụng biến con trỏ để liên lạc vòng quanh giữa các module. Ví dụ lớp vật lý có thể cảm nhận xem kênh bạn hay rồi. Nếu giao thức lớp mạng dựa trên cảm nhận song mang thì nó cần thông tin này từ lớp vật lý. Blackboard là một module mà các thông tin tương ứng có thể publish trên nó và sau đó có thể truy cập bởi bất kỳ module nào mà đăng ký với nó. BasicModule cung cấp mọi thứ cần thiết để tương tác với lớp Blackboard

Nếu bạn muốn subscribe() một tham số, bạn bắt buộc phải làm điều này ở giai đoạn 0 khi khởi tạo module:

```
void YourClass::initialize(int stage) {  
    BaseClass::initialize(stage);  
    if (stage == 0){  
        SomeBBItem item;  
        catItem = bb->subscribe(this, &item, -1);  
        ...  
    }else if(stage == 1) {  
        ...  
    }  
}
```

Biến item của lớp SomeBBItem thể hiện cho tham số mà bạn muốn đăng ký. Ví dụ bạn muốn đăng ký thông tin về vị trí của host bạn có thể khai báo biến: SomeBBItem catHostMove. Tiếp theo đó bạn cần phải thực hiện đăng ký biến này với module Blackboard như sau: catItem = bb->subscribe(this, &item, -1);

CatItem trả về kiểu số nguyên, this là con trỏ đến module mà bạn sau này bạn muốn lấy thông tin về sự thay đổi của tham số item, -1 là phạm vi của tham số. Hủy đăng ký: unsubscribe()

Muốn hủy đăng ký bạn làm như sau:

```
bb->unsubscribe(this, catItem);
```

Lấy thông tin đăng ký: bạn sử dụng hàm receiveBBItem:

```
void YourClass::receiveBBItem(int category, \  
const BBItem *details, int scopeModuleId) {
```

```
// in case you want to handle messages here
Enter_Method_Silent();
// in case not you but your base class subscribed:
BaseClass::receiveBBItem(category, details,
scopeModuleId);
// react on the stuff that you subscribed
if(category == catItem) {
someBBItemPtr =
static_cast<const SomeBBItem *>(details);
// do something }
```

Tham số đầu tiên của hàm này là `category` là số nguyên mà hàm đăng ký trả lại, tham số thứ hai là đối tượng của lớp mà bạn đăng ký và tham số thứ ba là phạm vi của tham số này. Bạn cũng có thể đặt hai `macroEnter_Method` or `Enter_Method_Silent` ở đầu. Điều này cho phép bạn sắp xếp hoặc hủy bản tin. Bạn cũng nên thông báo cho lớp cơ sở. Bây giờ bất kỳ sự thay đổi nào của tham số này, lớp cơ sở sẽ thông báo cho bạn.

Phần này chỉ trình bày tóm tắt về Mobility, bạn có thể tham khảo thêm cách sử dụng các hàm API trong thư mục/doc sau khi cài đặt Mobility.

3.2 Giới thiệu về PEGASIS

Mạng cảm biến bao gồm những nút giới hạn năng lượng và truyền thông vô tuyến để thu lượm thông tin hữu ích trong trường cảm biến. Việc hợp nhất dữ liệu cảm nhận được theo một cách có hiệu quả năng lượng là vấn đề then chốt trong hoạt động kéo dài thời gian sống của mạng. Trong bài toán tập hợp dữ liệu, trong mỗi vòng truyền thông mỗi nút sẽ có một gói dữ liệu để truyền đến trạm cơ sở. Nếu mỗi nút truyền trực tiếp dữ liệu đến trạm cơ sở thì nó sẽ nhanh chóng bị tiêu hao năng lượng. Giao thức LEACH khắc phục bằng cách tạo ra các cụm, trong mỗi cụm sẽ bầu chọn một nút chủ để tập hợp dữ liệu từ các nút thành viên và truyền đến trạm cơ sở. Bằng cách ngẫu nhiên lựa chọn nút chủ, LEACH đạt được sự cải thiện 8 lần so với cách truyền trực tiếp khi đo tỷ lệ nút chết. Mặc dù đạt được những cải thiện đáng kể nhưng LEACH vẫn còn một số giới hạn sau:

- Tiêu tốn lượng năng lượng đầu để thiết lập các cụm.
- Truyền dữ liệu trực tiếp từ mỗi cụm chủ đến trạm cơ sở ở xa.

- Số lượng các lần truyền ở khoảng cách xa sẽ tăng khi kích thước mạng tăng và tiêu hao tổng năng lượng của mạng.

Giao thức PEGASIS em đưa ra ở đây có sự cải thiện đáng kể hơn so với LEACH. PEGASIS (Power-Efficient Gathering in Sensor Information Systems) là giao thức dựa trên xây dựng chuỗi gần tối ưu. Tức là mỗi nút chỉ giao tiếp với một nút lân cận gần nó nhất, và lần lượt truyền dữ liệu đến trạm cơ sở, do đó nó làm giảm tối thiểu năng lượng sử dụng trong một vòng.

3.2.1 PEGASIS cơ bản

PEGASIS hỗ trợ tối thiểu hóa khoảng cách truyền trong mạng, tối thiểu hóa lượng mã đầu quảng bá, tối thiểu hóa khối lượng bản tin truyền đến trạm cơ sở và phân bố năng lượng đồng đều giữa các nút trong mạng.

Ý tưởng của PEGASIS là tạo một chuỗi các nút cảm biến để mỗi nút có thể nhận và truyền dữ liệu tới nút bên cạnh, việc truyền dữ liệu từ nút đến nút, tập hợp lại và sau cùng truyền đến trạm cơ sở. Các nút này sẽ thay nhau truyền dữ liệu đến trạm cơ sở, để năng lượng trung bình được sử dụng bởi mỗi nút được giảm ở mỗi vòng.

Để thực hiện thuật toán chúng ta giả sử tất cả các nút cảm biến đều có hiểu biết toàn cục về mạng và đều có thể sử dụng thuật toán Greedy. Thuật toán greedy thực hiện rất tốt và việc xây dựng chuỗi được thực hiện trước khi một vòng truyền dữ liệu bắt đầu.

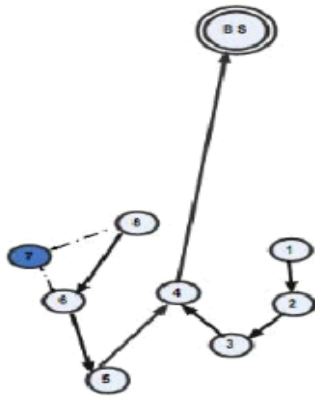
Để xây dựng một chuỗi chúng ta bắt đầu từ nút xa trạm BS nhất. Chúng ta làm điều này để đảm bảo các nút ở xa BS đều có nút lân cận gần nó, vì trong thuật toán greedy khoảng cách giữa các nút sẽ tăng dần vì các nút nằm trong chuỗi sẽ không được thăm lại. Hình (3.5) chỉ ra thứ tự liên kết, nút 0 nối với nút 3, nút 3 nối với nút 1, nút 1 nối với nút 2. Khi một nút chết, các nút sẽ phải xây dựng lại chuỗi và bỏ qua nút chết ấy.



Hình 3.5 Xây dựng chuỗi sử dụng thuật toán Greedy

Để tập hợp dữ liệu mỗi vòng, mỗi nút sẽ nhận dữ liệu từ nút hàng xóm và hợp nhất với dữ liệu nó cảm nhận được và truyền đến nút hàng xóm tiếp theo trong chuỗi.

Sau khi chuỗi được thành lập, bước tiếp theo là chọn nút chủ. Nút chủ được chọn bằng cách sau: ở vòng thứ i thì nút thứ $i \bmod N$ (N là số nút trong mạng) sẽ làm chủ. Như vậy năng lượng sẽ được san sẻ giữa các nút. Khi một nút chết, chuỗi sẽ được cập nhật lại bằng cách bỏ qua nút đó trong chuỗi. Như hình 4.6 khi nút 7 chết, nút 8 sẽ cố gắng liên lạc với nút 6



Hình 3.6 Xử lý lỗi khi một nút trong chuỗi chết

3.2.2 PEGASIS cải tiến

Trong giải thuật PEGASIS cơ bản, chúng ta thấy rằng mặc dù năng lượng đã được chia sẻ cho các nút nhưng các nút ở xa trạm BS sẽ bị tiêu thụ năng lượng nhiều hơn và do đó nhanh chóng chết đi. Như vậy sẽ ảnh hưởng đến thời gian sống của toàn mạng. Sau đây, ta đưa ra một cải tiến trong quá trình chọn nút chủ để làm tăng thời gian sống của mạng. Chúng ta sẽ không cho các nút ở xa trạm BS và có năng lượng thấp làm nút chủ. Chúng ta chọn nút chủ như sau:

- Tất cả các nút sẽ tính toán tỉ số R_i như sau:

$$R_i = P_{a,i} / P_{T,i}$$

Trong đó: $P_{a,i}$: năng lượng của nút i tại thời điểm hiện tại.
 $P_{T,i}$: năng lượng cần thiết để nút i truyền đến trạm cơ sở.

- Nút cuối chuỗi sẽ bắt đầu gửi gói chứa giá trị R_i của nó về phía nút hàng xóm trong chuỗi. Mỗi nút nhận gói này sẽ so sánh giá trị hiện tại trong gói với giá trị R của nó. Nếu cao hơn, nó đơn giản sẽ chuyển tiếp gói, còn nếu nhỏ hơn, nó

sẽ biến đổi gói với giá trị hiện tại của nó và chuyển tiếp đến nút cạnh nó trong chuỗi.

- Nút có giá trị R cao nhất sẽ là nút chủ. Nút chủ sẽ thông báo cho cách thành viên trong chuỗi biết.

- Việc bầu chọn nút chủ được thực hiện theo một số vòng nào đó.

Số vòng để lựa chọn nút chủ thay đổi thích ứng theo mức năng lượng hiện tại của mỗi nút. Tại thời điểm bắt đầu, mức năng lượng của mỗi nút khác nhau tương đối nhỏ và các nút vẫn có mức năng lượng rất cao. Một khi được lựa chọn làm nút chủ, nút sẽ giữ vai trò này trong một số vòng. Sau đó nó khởi tạo chu kỳ lựa chọn nút chủ khác và do đó làm giảm số mào đầu liên kết với nút chủ. Khi mức năng lượng của các nút giảm thì số vòng để chọn lại nút chủ cũng giảm và do đó tránh được một nút tiêu thụ năng lượng quá nhiều khi làm nút chủ. Khi mức năng lượng của nút trở nên quá thấp, việc chọn nút chủ sẽ diễn ra thường xuyên ở mỗi vòng. Kỹ thuật này đảm bảo các nút có năng lượng cao và gần trạm BS sẽ có nhiều cơ hội làm nút chủ hơn. Việc chọn nút gần trạm BS làm nút chủ sẽ giảm tổng chi phí truyền trong mạng.

Sau khi chọn nút chủ. Nút chủ sẽ truyền thẻ bài dọc theo chuỗi đến nút cuối chuỗi. Nút này bắt đầu cảm nhận dữ liệu và truyền đến nút bên cạnh nó trong chuỗi. Nút này sẽ tập hợp dữ liệu của nó và dữ liệu nhận được trong một gói và truyền đến nút bên cạnh nó trong chuỗi. Cứ như thế, dữ liệu được truyền đến trạm cơ sở

Như vậy về mặt thuật toán chúng ta thấy rằng PEGASIS có những cải tiến đáng kể hơn so với LEACH về thời gian sống. PEGASIS tiết kiệm năng lượng ở một số giai đoạn. Cụ thể như sau:

- Đầu tiên, việc tập hợp dữ liệu cục bộ, khoảng cách mà hầu như các nút trong mạng truyền dữ liệu nhỏ hơn nhiều so với việc truyền dữ liệu của các nút thành viên đến nút chủ trong cụm của LEACH.

- Thứ hai, khối lượng dữ liệu nút chủ trong PEGASIS nhận được nhiều nhất là hai bản tin trong khi đó của LEACH là 20 (nếu mạng có 100 nút), nhiều hơn rất nhiều.

- Thứ ba, chỉ có một nút trong mạng truyền dữ liệu đến trạm cơ sở trong khi đó ở LEACH có 5% số nút truyền đến trạm cơ sở.

Mặc dù có những cải tiến đáng kể so với LEACH, nhưng PEGASIS vẫn tồn tại một số hạn chế như sau:

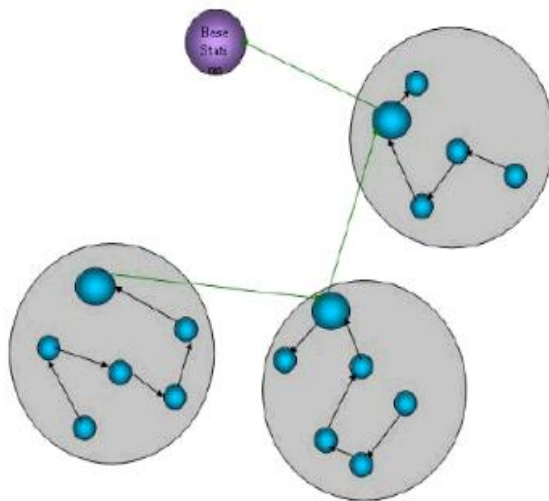
- Trễ trong mạng khá lớn, đặc biệt là nếu kích thước mạng lớn thì chuỗi sẽ rất dài và số lượng bước nhảy rất cao khi truyền dữ liệu từ cuối chuỗi đến trạm cơ sở.

- Thêm vào đó, các nút trong chuỗi phải biết cấu hình mạng và điều này không phải luôn luôn dễ dàng đối với mạng cảm biến.

- Xảy ra hiện tượng tắc cổ chai tại nút chủ. Tức là dữ liệu tập hợp được đến nút chủ thì nút chủ không còn đủ năng lượng truyền đến trạm BS nữa.

Khắc phục:

Để khắc phục trễ chúng ta có thể chia mạng ra thành nhiều khu vực con, mỗi khu vực con này sẽ thiết lập nên một chuỗi. Tương ứng với mỗi chuỗi con sẽ có một nút chủ. Các nút chủ này lại có thể liên kết với nhau tạo thành chuỗi cấp cao hơn và chuỗi này sẽ lại chọn nút chủ để truyền đến trạm BS. Mô tả như hình (4.7) sau



Hình 3.7 Khắc phục của PEGASIS

3.3 Mô phỏng

3.3.1 Mô hình năng lượng

Đặc điểm của kênh vô tuyến

Trong kênh vô tuyến, quá trình truyền sóng điện từ có thể được mô hình bằng quy luật hàm giảm công suất ở khoảng cách giữa bộ phát và bộ thu. Thêm vào đó, nếu giữa bộ phát và bộ thu không truyền theo đường truyền thẳng mà bị cản trở bởi chướng ngại vật thì khi đó sóng điện từ sẽ đến bộ thu bằng các đường khác nhau tại thời điểm khác nhau. Điều này gây ra hiện tượng phading đa đường. Vấn đề là ta sử dụng mô hình nào (mô hình đường truyền thẳng hay mô hình phading đa

đường) trong mạng cảm biến. Ta biết công suất thu được tại bộ thu sẽ giảm khi khoảng cách giữa bên phát và bên nhận tăng lên. Theo “Wendi” thì cả hai mô hình này đều được sử dụng tùy thuộc vào khoảng cách giữa bên truyền và bên phát. Nếu khoảng cách giữa bên phát và bên thu nhỏ hơn khoảng cách dcross-over thì mô hình không gian được sử dụng (suy hao d2) và nếu ngược lại thì mô hình 2 đường dẫn được sử dụng (suy hao d4)

Điểm cắt (cross-over) được định nghĩa như sau

$$d_{\text{cross-over}} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda}$$

Trong đó:

L1

≥: hệ số suy hao hệ thống không liên quan đến quá trình truyền

hr: chiều cao của anten bên nhận so với mặt đất

ht: là chiều cao của anten phát so với mặt đất

λ: là bước sóng của tín hiệu sóng mang

Nếu khoảng cách truyền nhỏ hơn dcross-over thì công suất truyền suy hao được tính như sau:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L}$$

Trong đó:

Pr(d): công suất nhận được ở khoảng cách d

Pt: công suất bên truyền

Gt: hệ số khuếch đại của anten bên truyền

Gr: hệ số khuếch đại của anten bên nhận

λ: là bước sóng của tín hiệu sóng mang

L1

\geq : hệ số suy hao hệ thống không liên quan đến quá trình truyền

d: khoảng cách giữa bên truyền và bên nhận

Phương trình này mô hình sự suy hao khi bên phát và bên thu có sự thông tin tầm nhìn thẳng (truyền thẳng không có chướng ngại vật), điều này chỉ xảy ra nếu như bên phát và bên thu gần nhau ($d < d_{\text{cross-over}}$). Nếu khoảng cách $d > d_{\text{cross-over}}$ thì ta có công thức sau:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4}$$

Trong đó:

$P_r(d)$: công suất nhận được ở khoảng cách d

P_t : công suất bên truyền

h_r : chiều cao của anten bên nhận so với mặt đất

h_t : là chiều cao của anten phát so với mặt đất

d: khoảng cách giữa bên truyền và bên nhận

G_t : hệ số khuếch đại của anten bên truyền

G_r : hệ số khuếch đại của anten bên nhận

Trong trường hợp này tín hiệu nhận được theo cả hai hướng, hướng trực tiếp và hướng phản xạ. Vì có một hay nhiều đường truyền mà tín hiệu đến, nên tín hiệu sẽ suy giảm theo d^4 .

Ví dụ:

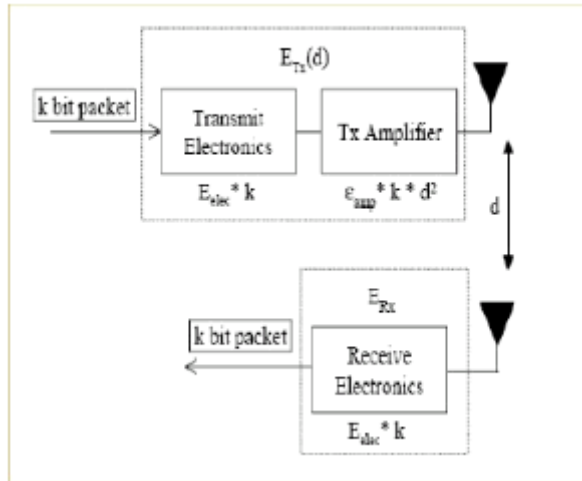
Cho $G_t = G_r = 1$, $h_t = h_r = 1.5\text{m}$, hệ thống không suy hao $L = 1$, $f = 914\text{ MHz}$

Suy ra: $\lambda = 3 \times 10^8 / 914 \times 10^6 = 0.328\text{m}$, thay giá trị này vào hai biểu thức trên

Ta có:

$$P_r = \begin{cases} 6.82 \times 10^{-4} \frac{P_t}{d^4} & : d < 86.2\text{m} \\ 2.25 \frac{P_t}{d^4} & : d \geq 86.2\text{m} \end{cases}$$

Trong phần này, chúng ta sẽ xem xét một mô hình đơn giản sẽ áp dụng trong phần mô phỏng.



Hình 3.8 Mô hình năng lượng đơn giản

Như thảo luận ở trên sự suy giảm công suất trong quá trình truyền phụ thuộc vào khoảng cách giữa bên phát và bên thu, nếu khoảng cách tương đối ngắn, ta có thể áp dụng mô hình tỉ lệ nghịch với d^2 , và ngược lại nếu khoảng cách dài ta áp dụng mô hình tỉ lệ với d^4 . Bộ điều khiển công suất có thể đảo ngược sự suy hao này bằng cách thiết lập khuếch đại công suất để đảm bảo mức công suất nào đó tại bên nhận, do đó để truyền một bản tin dài l bit, ở khoảng cách d ta có:

$$E_{Tx}(l, d) = E_{Tx-dcc}(l) + E_{Tx-amp}(l, d)$$

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs-amp} d^2 & : d < d_{max-cr} \\ lE_{elec} + l\epsilon_{fs-ray-tramp} d^4 & : d \geq d_{max-cr} \end{cases}$$

Bên nhận:

$$E_{Rx}(l) = E_{Rx-dcc}(l)$$

$$E_{Rx}(l) = lE_{elec}$$

Trong đó năng lượng điện tử, E_{elec} phụ thuộc vào các hệ số như: mã hóa số, điều chế, lọc tín hiệu trước khi được gửi đến bộ khuếch đại. Thêm vào đó việc sử dụng kỹ thuật trải phổ năng lượng điện tử phải tính đến cả năng lượng trải

phổ tín hiệu khi truyền và tương quan dữ liệu với mã trải phổ khi nhận. Các nhà nghiên cứu đã thiết kế chip thu phát baseband hỗ trợ thông tin trải phổ đa người dùng và hoạt động ở 165mW ở chế độ truyền và 46.5 mW ở chế độ nhận, theo “windy”, người ta đã tập hợp năng lượng tiêu thụ trên 1 bit dữ liệu ở bộ thu phát là $E_{elec} = 50 \text{ nJ/bit}$ đối với bộ thu phát tốc độ 1Mbps. Điều này có nghĩa là phần điện tử sẽ tiêu tán 50mW khi hoạt động (thu hoặc phát dữ liệu).

Hai tham số: $\epsilon_{\text{friss-amp}}$ và $\epsilon_{\text{two-ray-amp}}$ tùy thuộc vào độ nhạy máy thu yêu cầu, và nhiễu tạp âm máy thu, do đó công suất truyền cần phải điều chỉnh để công suất máy thu lớn hơn một mức ngưỡng PR-thresh. Chúng ta có thể làm ngược lại từ ngưỡng của công suất máy thu để tính toán công suất truyền tối thiểu. Nếu tốc độ truyền là R_b thì công suất truyền P_t sẽ bằng năng lượng truyền trên bit $E_{Tx-amp}(1, d)$ nhân với tốc độ:

$$P_t = E_{Tx-amp}(1, d) * R_b$$

Khi đó ta có:

$$P_t = \begin{cases} \epsilon_{\text{friss-amp}} R_b d^2 & : d < d_{\text{crossover}} \\ \epsilon_{\text{two-ray-amp}} R_b d^4 & : d \geq d_{\text{crossover}} \end{cases}$$

Sử dụng mô hình kênh truyền ở trên chúng ta sẽ có:

$$P_r = \begin{cases} \epsilon_{\text{friss-amp}} R_b G_t G_r \lambda^2 & : d < d_{\text{crossover}} \\ \epsilon_{\text{two-ray-amp}} R_b G_t G_r h_t^2 h_r^2 & : d \geq d_{\text{crossover}} \end{cases}$$

Và

$$\epsilon_{\text{friss-amp}} = \frac{P_{r-\text{thresh}} (4\pi)^2}{R_b G_t G_r \lambda^2}$$

$$\epsilon_{\text{two-ray-amp}} = \frac{P_{r-\text{thresh}}}{R_b G_t G_r h_t^2 h_r^2}$$

Do đó công suất truyền P_t sẽ là hàm của ngưỡng công suất bên thu và khoảng cách d

$$P_t = \begin{cases} \alpha_1 P_{r-\text{thresh}} d^2 & : d < d_{\text{crossover}} \\ \alpha_2 P_{r-\text{thresh}} d^4 & : d \geq d_{\text{crossover}} \end{cases}$$

Với

$$\alpha_1 = \frac{(4\pi)^2}{G_t G_r \lambda^2} ; \alpha_2 = \frac{1}{G_t G_r h_t^2 h_r^2}$$

Chúng ta có thể xác định mức ngưỡng ở máy thu sử dụng việc đánh giá nhiễu ở máy thu. Nếu nhiễu sàn nhiễu là 99dBm và tạp âm nhiễu máy thu là 17dB2 và chúng ta yêu cầu tỉ số tín hiệu trên nhiễu ít nhất là 30dB để nhận tín hiệu k nhiễu, thì công suất nhận tối thiểu là

$$P_{R-ibmsb} > 30 + (-82) = -52 \text{ dBm}$$

Do đó công suất nhận được ít nhất phải là -52dBm hay 6.3 nW để nhận thành công gói. Thay các giá trị vào ($G_t=G_r=1$, $h_t=h_r=1.5\text{m}$, $L=1$, $f=914$ MHz, $\lambda=0.328\text{m}$, $R_b=1\text{Mbps}$). Ta có:

$$e_{\text{fix-amp}} = 10 \text{ pJ/bit/m}^2$$

$$e_{\text{fix-no-amp}} = 0.0013 \text{ pJ/bit/m}^4$$

Chúng ta sẽ sử dụng mô hình vô tuyến đơn giản kiểu thứ nhất như sau: Truyền bản tin k bit ở khoảng cách d sử dụng mô hình vô tuyến.

ETx-elec: năng lượng/bit truyền

ERx-elec: năng lượng /bit nhận

ϵ_{amp} : hệ số của bộ khuếch đại

Phương trình bên truyền:

$$E_{T_x}(k, d) = E_{T_x\text{-elec}}(k) + E_{T_x\text{-amp}}(k, d)$$

$$E_{T_x}(k, d) = E_{\text{elec}} * k + \epsilon_{\text{amp}} * k * d^2$$

Phương trình bên nhận:

$$E_{R_x}(k) = E_{R_x\text{-elec}}(k)$$

$$E_{R_x}(k) = E_{\text{elec}} * k$$

Việc nhận bản tin cũng tiêu tốn năng lượng khá cao cho nên chúng ta cũng cần tối thiểu số lần truyền và nhận ở mỗi nút.

3.3.2 Giả thiết và thiết lập thông số ban đầu cho quá trình mô phỏng

Chúng ta có một số các giả thiết ban đầu như sau:

- Các nút trong mạng đều biết về topology của mạng

- Tất cả các nút trong toàn mạng đều có thể truyền dữ liệu trực tiếp đến trạm cơ sở (Sink).

Số nút trong mạng $N=100$ nút.

Phạm vi mô phỏng mạng (50m x 50m), trạm BS đặt ở vị trí (25m,150m).

Nếu phạm vi mô phỏng mạng là 100m x 100m trạm BS đặt ở vị trí (50m,300m).

Năng lượng ban đầu $iniPower = 0.25J; 0.5J$

Năng lượng tiêu tốn khi xử lý một bit: $E_{elec} = 50nJ/bit$

$$c_{fix-amp} = 10 pJ/bit/m^2$$

Hệ số khuếch đại

Chúng ta sẽ mô phỏng trên Mobility Framework. Các nút trong mạng được phân bố vị trí một cách ngẫu nhiên. Quá trình mô phỏng gồm các bước như sau:

- Bước 1: Tìm nút xa trạm BS nhất
- Bước 2: Thiết lập chuỗi
- Bước 3: Chọn nút chủ
- Bước 4: Truyền dữ liệu và xử lý lỗi khi nút chết

Chúng ta sẽ đi lần lượt từng bước như trên.

Bước 1: Tìm nút xa nhất

Ban đầu BS sẽ đưa ra lệnh xây dựng mạng thông qua bản tin INITIATE_CONFIGURE_NETWORK. Sau đó BS broadcast bản tin BROADCASTING_POSITION cho toàn bộ nút trong mạng để tính toán khoảng cách từ các nút đó đến BS. Bản tin BROADCASTING_POSITION:

```
cplusplus {{#include "NetwPkt_m.h"}};
class NetwPkt;
message BSP extends NetwPkt {
fields:
int posX;
int posY;
}
```

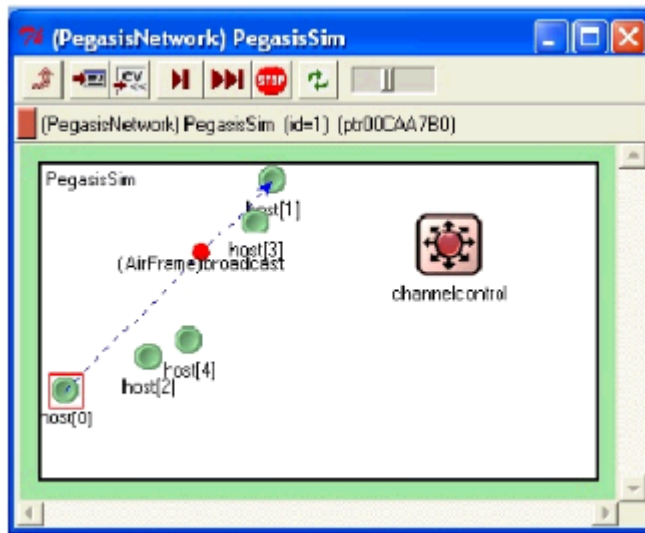
Tham số pos X và pos Y chứa tọa độ của trạm BS. Chúng ta có thể lấy được vị trí hiện tại của BS thông qua việc đăng ký biến HostCoor với Blackboard

Pos X = (int) (HostCoor.x)

Pos Y = (int) (HostCoor.y)

Hình (4.9) là giao diện của Mobility khi mô phỏng, miêu tả quá trình broadcast của trạm cơ sở (BS) đến các nút trong mạng, host [0] trong hình chính là trạm cơ sở.

Đường nét đứt là đường truyền vô tuyến của bản tin. Channel control điều khiển việc di động của các host và tính toán khoảng cách giao thoa giữa các nút. Trong phần mô phỏng này ta giả sử các nút cảm biến không di động.



Hình 3.9. Trạm BS gửi broadcast đến cho các nút trong mạng

Các nút nhận được bản tin này sẽ tính toán khoảng cách đến BS nhờ thông số tọa chứa trong bản tin. Và sau đó gửi reply lại bằng bản tin REPLY_BROADCASTING_POSITION:

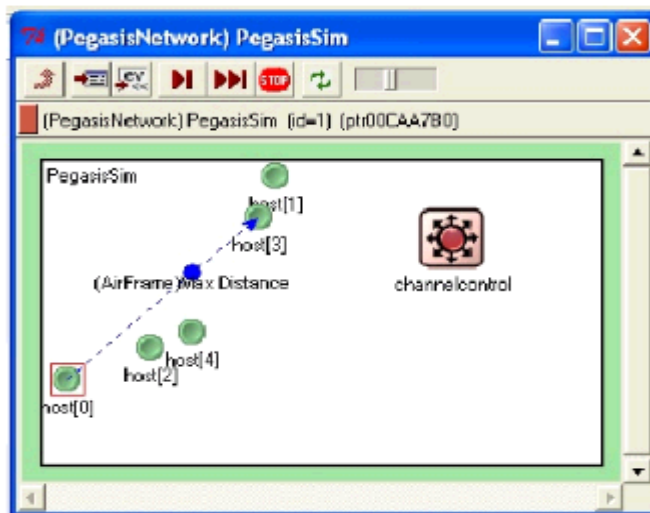
```
cplusplus {{#include "NetwPkt_m.h"}};  
class NetwPkt;  
message RBSP extends NetwPkt {  
fields:  
double distance;  
};
```

Trong đó: Tham số distance sẽ chứa khoảng cách của các nút đến BS. Sau khi BS nhận được bản tin này, BS sẽ so sánh các giá trị distance và tìm ra MaxDistance, tức là sẽ tìm ra nút xa nhất so với BS.

Bước 2: Thiết lập chuỗi

Sau khi tìm được nút xa nhất, BS gửi thông báo đến nút đó thông qua bản tin MAX_DISTANCE (hình 4.10).

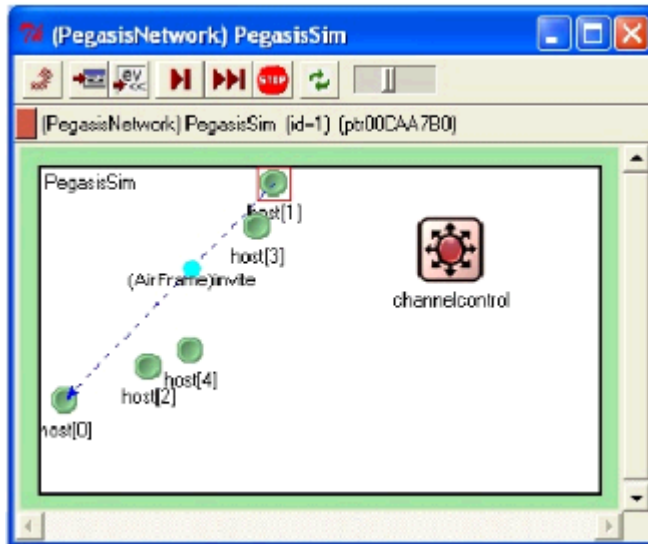
Nút xa nhất này chính là nút gốc của chuỗi. Nút này nhận được bản tin MAX_DISTANCE sẽ bắt đầu tìm nút khác gần nó nhất và cho vào chuỗi. Nó sử dụng bản tin FIND_NODE_INTO_CHAIN có các trường giống như bản tin BROADCASTING_POSITION của BS .



Hình 3.10. Trạm BS gửi bản tin Max Distance đến nút xa nhất

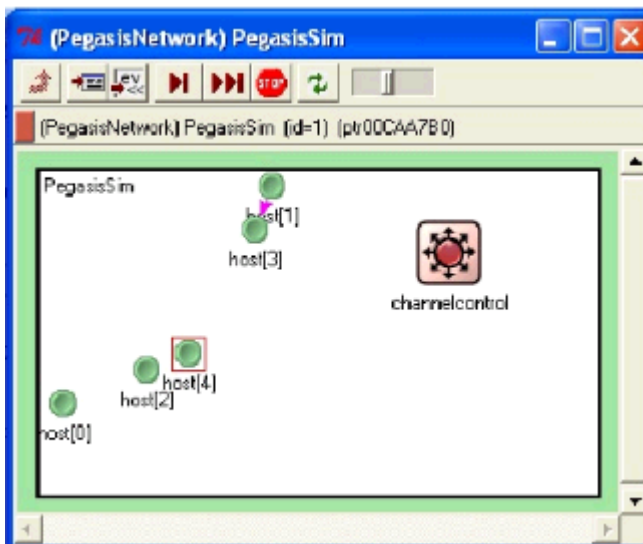
Các nút xung quanh nhận được bản tin này và gửi bản tin đáp trả lại REPLY_FIND_NODE_INTO_CHAIN. Nút này căn cứ vào các giá trị distance trong các bản tin để lựa chọn ra nút gần mình nhất có địa chỉ MinAddr và khoảng cách Min. Sau đó nó gửi bản tin mời gọi vào chuỗi INVITE_INTO_CHAIN như sau:

```
cplusplus {{#include "NetwPkt_m.h"}};  
class NetwPkt;  
message InvitePkt extends NetwPkt {  
fields:  
int vitri;  
int index;  
};
```



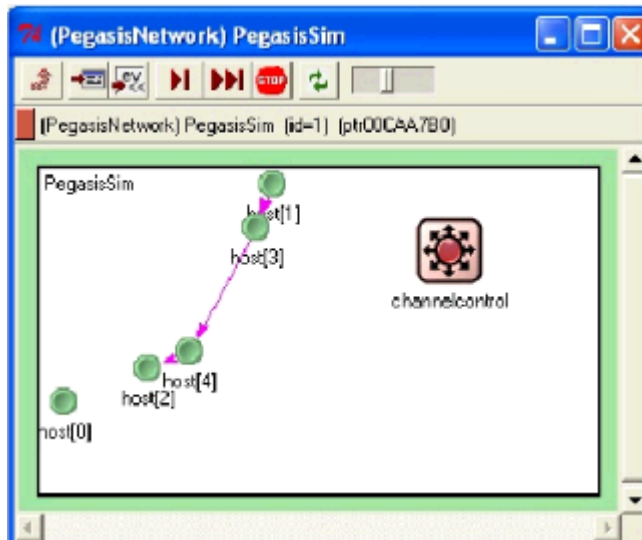
Hình 3.11. Nút xa nhất chuỗi gửi bản tin Invite mời nút gần nhất vào chuỗi

Cứ như vậy, sau khi nút này vào chuỗi thì lại tiếp tục mời gọi các nút còn lại trong chuỗi vào chuỗi.



Hình 3.12. Các nút kết nối vào nhau thành chuỗi

Chú ý: Ta nên khai báo thêm một biến vào Chuoi kiểu bool để đánh dấu khi một nút đã vào chuỗi rồi. Điều này sẽ rất thuận lợi và các nút một khi đã vào chuỗi rồi sẽ không tính toán khoảng cách khi nhận được bản tin FIND_NODE_INTO_CHAIN.



Hình 3.13 Chuỗi sau khi thiết lập xong.

Sau khi tất cả các nút đều đã vào chuỗi, nút cuối cùng vào chuỗi sẽ gửi bản tin REQUEST_CHOSING_HEADER đến cho trạm BS. BS sẽ bắt đầu khởi tạo quá trình chọn nút chủ bằng cách gửi đến nút gốc của chuỗi, chính là nút có khoảng cách xa BS nhất.

Bước 3: Chọn nút chủ

Nút xa nhất bắt đầu tính toán tỉ lệ: $\text{Ratio} = \text{curPower} / \text{distance}$

Với curPower là năng lượng hiện tại của nút và cho vào bản tin truyền dọc theo chuỗi, tại các nút: khi nhận được bản tin cũng tính toán giá trị này và sau đó gửi so sánh giá trị Ratio của nó và của bản tin nhận được. Nếu nhỏ hơn nó đơn giản sẽ forward đi còn ngược lại sẽ thay thế bằng Ratio của mình và lại truyền đi. Nút có giá trị Ratio cao nhất sẽ được chọn làm nút chủ. Nút chủ sẽ thông báo cho các nút khác biết vị trí của các nút và nó là nút chủ.

Bước 4: Truyền dữ liệu và xử lý lỗi khi một nút chết

Nút chủ bắt đầu gửi TOKEN đến nút gốc chuỗi để bắt đầu một vòng truyền dữ liệu, sau đó như thuật toán đã nêu ở trên, các nút sẽ lần lượt tích hợp dữ liệu của nó và truyền đến nút chủ. Sau đó nút chủ sẽ tập hợp dữ liệu của nó và hai bản tin từ hai phía truyền về và truyền đến Sink. Mỗi bản tin có kích thước $k=2000$ bit. Tại các nút mỗi khi nhận được bản tin sẽ tính toán năng lượng nhận và truyền theo công thức đã đề cập ở mô hình mô phỏng.

Phương trình tính toán năng lượng khi truyền bản tin:

$$E_{Tx}(k, d) = E_{Tx-timeout}(k) + E_{Tx-amp}(k, d)$$

$$E_{Tx}(k, d) = E_{ctc} * k + c_{amp} * k * d^2$$

Phương trình tính toán năng lượng khi nhận bản tin:

$$E_{Rx}(k) = E_{Rx-timeout}(k)$$

$$E_{Rx}(k) = E_{ctc} * k$$

Sau mỗi lần nhận gói tin, các nút sẽ kiểm tra xem còn đủ năng lượng để truyền và nhận không? Nếu không đủ năng lượng thì nó sẽ không truyền gói đi và cũng sẽ không nhận gói tin. Lúc này một nút coi như là đã chết, các nút khác dựa vào thời gian timeout, không thấy nút đó gửi dữ liệu đến sẽ thông báo đến nút chủ để cập nhật lại chuỗi. Chuỗi mới sẽ bỏ qua nút chết. Sau đó nút chủ lại gửi TOKEN để bắt đầu thu thập dữ liệu. Khi nút chết, nút chủ có nhiệm vụ gửi thông báo đến BS, BS đếm số nút chết và sau đó BS đưa ra kết quả.

3.3.3 Kết quả mô phỏng

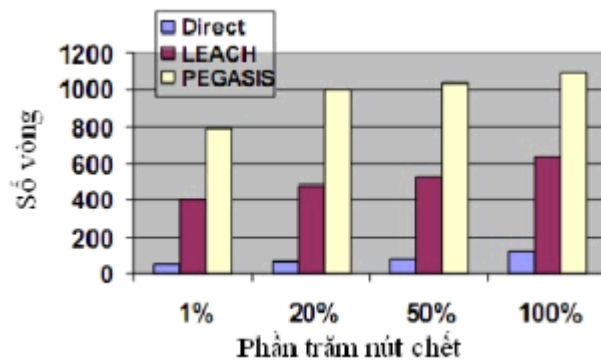
Sau đây ta sẽ đưa ra kết quả mô phỏng của ba giao thức: truyền tin trực tiếp, LEACH, PEGASIS. Trong bảng (3.10) là kết quả khi mô phỏng, phần đậm ứng với kích thước mạng là (50m, 50m), phần còn lại là kết quả mô phỏng ứng với mạng có kích thước (100m, 100m).

Trong bảng (3.2) ta thấy, các nút bắt đầu chết đồng loạt sau khi 20% nút chết. Bởi vì khoảng cách giữa các nút lúc này lớn hơn rất nhiều, việc lựa chọn nút chủ diễn ra thường xuyên mỗi vòng và các nút cũng nhanh chóng tiêu hao năng lượng hơn.

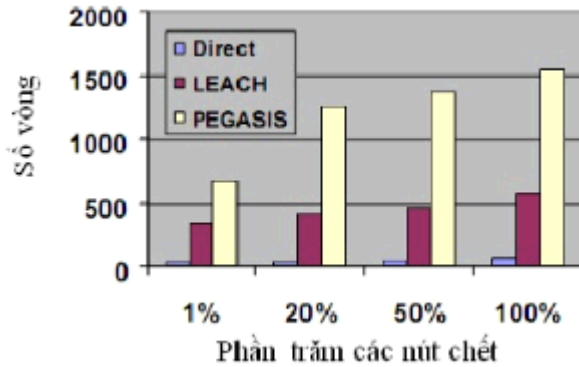
Bảng 3.2 Số vòng khi 1%, 20%, 50%, và 100% nút chết

năng lượng J/Node	Giao thức	1%	20%	50%	100%
.25	Direct	54	62	76	117
	LEACH	402	480	523	635
	PEGASIS	788	1004	1041	1096
.5	Direct	108	124	152	235
	LEACH	803	962	1036	1208
	PEGASIS	1578	2011	2082	2192
1.0	Direct	215	248	304	471
	LEACH	1610	1921	2055	2351
	PEGASIS	3159	4023	4165	4379
.25	Direct	14	16	20	30
	LEACH	166	204	232	308
	PEGASIS	335	624	684	779
.5	Direct	28	32	40	61
	LEACH	339	408	461	576
	PEGASIS	675	1250	1362	1544
1.0	Direct	56	64	80	122
	LEACH	690	812	911	1077
	PEGASIS	1346	2497	2720	3076

Hình (3.14) và (3.15) chỉ ra số vòng khi 1% , 20%, 50% và 100% nút chết với kích thước mô phỏng mạng là (50m,50m) và (100m,100m). Từ kết quả ta nhận thấy rằng, số vòng PEGASIS đạt được gấp 2 lần so với LEACH với kích thước mạng là (50m,50m).



Hình 3.14 Kết quả khi mô phỏng mạng có kích thước (50m, 50m) với năng lượng ban đầu của nút là 0.25 J



Hình 3.15 Kết quả mô phỏng khi kích thước mạng là (100m.100m) với năng lượng của nút ban đầu là 0.5J

Năng lượng ban đầu của nút trong hình (3.9) là 0.25J và hình (3.10) là 0.5J. Mức năng lượng tăng gấp đôi dẫn đến số vòng cũng tăng gấp đôi. Với kích thước mạng (100m, 100m) thì số vòng khi thực hiện PEGASIS gấp ba so với LEACH.

3.4 Kết luận và hướng nghiên cứu tiếp theo

Trong chương này đã đưa ra kết quả mô phỏng của PEGASIS. Kết quả đã cho thấy PEGASIS khắc phục được nhược điểm của LEACH bằng cách loại bỏ lượng mào đầu của thông tin các cụm động, tối thiểu hóa khoảng cách truyền và nhận giữa các nút trong mạng, và chỉ sử dụng một lần truyền dữ liệu hợp nhất trên mỗi vòng đến trạm cơ sở. Các nút thay nhau truyền dữ liệu hợp nhất đến trạm cơ sở làm cân bằng năng lượng tiêu tán trong mạng và tăng khả năng chống lại lỗi khi các nút chết ở vị trí ngẫu nhiên.

Việc phân bố năng lượng trong mạng tải làm tăng thời gian sống và chất lượng của mạng. Việc mô phỏng cho thấy giao thức PEGASIS tốt hơn LEACH và thậm chí cải thiện hơn khi kích thước mạng tăng.

Tuy nhiên, một vấn đề nổi trội trong PEGASIS là trễ truyền, nút chủ phải đợi nhận được bản tin dữ liệu hợp nhất của các nút sau đó mới truyền đến trạm cơ sở. Hơn nữa thường xảy ra hiện tượng nút cổ chai tại nút chủ. Hướng nghiên cứu tiếp theo cần khắc phục được các nhược điểm này.

KẾT LUẬN

Các khái niệm và các vấn đề liên quan đến mạng cảm biến vẫn còn là vấn đề khá mới với nhiều người. Trong đồ án này em đã trình bày tổng quan về mạng cảm nhận không dây. Với các tính năng ưu việt cùng với các ứng dụng đa dạng nó có thể làm việc trong các điều kiện khắc nghiệt mà không phải mạng nào cũng có. Vì vậy mà trong tương lai không xa thì mạng cảm nhận không dây sẽ phát triển nhanh chóng. Em hy vọng rằng đồ án này sẽ đóng góp một phần nhỏ vào việc nghiên cứu về lĩnh vực còn tương đối mới mẻ ở Việt Nam.

Trong phạm vi của đồ án này em đã nghiên cứu được khái quát về mạng cảm nhận không dây và tìm hiểu về nguyên lý định vị các phương pháp định vị và giải thuật định vị nút mạng. Và đã biết được cách xác định vị trí nút mạng và biết được cách tính toán xác định vị trí của nút mạng thông qua một số các bài toán. Do đây là vấn đề mới mẻ cùng với kiến thức còn hạn chế và thời gian nghiên ngắn lên đồ án của em không tránh khỏi những thiếu sót, em rất mong nhận được sự phê bình, của các thầy cô để đồ án của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn thầy giáo Th.S Nguyễn Trọng Thế, Khoa Công Nghệ Thông Tin DHDL Hải Phòng đã nhiệt tình giúp đỡ em trong thời gian vừa qua.

Hải Phòng, tháng 7 năm 2011

Sinh viên thực hiện

Vũ Xuân Tình

TÀI LIỆU THAM KHẢO

[1]. Holger Karl Andreas Willig, Protocols and Architectures for Wireless Sensor Networks, Wiley, 2005.

[2]. S. Lindsay, PEGASIS: power-Efficient Gathering in Sensor Information Systems, Computer Systems Research Department The Aerospace Corporation P.O. Box 92957, Los Angeles, CA 90009-2957.

[3]. Jamal N. Al-Karaki Ahmed E. Kamal, Routing Techniques in Wireless Sensor Networks, Dept. of Electrical and Computer Engineering Iowa State University, Ames, Iowa 50011.

[4]. Armin Veichtlbauer, Peter Dorfinger Salzburg, Modeling of Energy Efficient Wireless Communication, Research Forschungsgesellschaft mbH, Advanced Networking Center Salzburg, Austria.

[5]. I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey”, Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, Received 12 December 2001; accepted 20 December 2001

[6]. <http://ceng.usc.edu/~anrg/SensorNetBib.html>