

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN QUANG HUY

**THIẾT KẾ MẠNG VÀ PHÂN TÍCH HIỆU SUẤT CỦA MẠNG
CẢM BIẾN KHÔNG DÂY CHO GIÁM SÁT SỨC KHỎE**

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2016

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN QUANG HUY

**THIẾT KẾ MẠNG VÀ PHÂN TÍCH HIỆU SUẤT CỦA MẠNG
CẢM BIẾN KHÔNG DÂY CHO GIÁM SÁT SỨC KHỎE**

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: TRUYỀN DỮ LIỆU VÀ MẠNG MÁY TÍNH

MÃ SỐ:

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS. TS. NGUYỄN ĐÌNH VIỆT

Hà Nội – 2016

LỜI CAM ĐOAN

Tôi xin cam đoan nội dung của luận văn “***Thiết kế mạng và phân tích hiệu suất mạng cảm biến không dây cho giám sát sức khỏe***” là sản phẩm do tôi thực hiện dưới sự hướng dẫn của **PGS. TS. Nguyễn Đình Việt**. Trong toàn bộ nội dung của luận văn, những điều được trình bày là do tôi nghiên cứu được từ các tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin chịu trách nhiệm cho lời cam đoan của mình.

Hà Nội, ngày 21 tháng 11 năm 2016

Người cam đoan

Trần Quang Huy

LỜI CẢM ƠN

Tôi xin bày tỏ lòng biết ơn sâu sắc đến thầy hướng dẫn của tôi, PGS.TS. Nguyễn Đình Việt. Thầy đã giúp tôi có những cơ hội để có thể theo đuổi nghiên cứu lĩnh vực mình yêu thích. Trong suốt quá trình thực hiện luận văn, thầy đã tận tình hướng dẫn cho tôi, góp ý cho tôi về đường lối, đồng thời đưa ra những lời khuyên bổ ích để tôi có thể hoàn thành luận văn của mình.

Tiếp đến, tôi xin chân thành cảm ơn các thầy cô giáo trong Khoa Công nghệ Thông tin, Đại học Công nghệ - Đại học Quốc gia Hà Nội đã truyền đạt cho tôi những kiến thức và kinh nghiệm vô cùng quý báu trong quá trình học tập và nghiên cứu.

Tôi cũng muốn cảm ơn các bạn cùng lớp và các đồng nghiệp đã cho tôi những lời động viên, những hỗ trợ và góp ý về mặt chuyên môn.

Cuối cùng, tôi xin cảm ơn gia đình, bạn bè, những người đã luôn bên cạnh ủng hộ và động viên tôi.

Hà Nội, tháng 11 năm 2016

Trần Quang Huy

MỤC LỤC

LỜI CAM ĐOAN	1
LỜI CẢM ƠN	2
MỤC LỤC.....	3
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	5
DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ	6
DANH MỤC BẢNG BIỂU	7
CHƯƠNG 1 MỞ ĐẦU	8
1.1 Lý do chọn đề tài	8
1.2 Mục tiêu của đề tài.....	8
1.3 Tổ chức của luận văn.....	8
CHƯƠNG 2 MẠNG CẢM BIẾN KHÔNG DÂY VÀ ỨNG DỤNG TRONG CÁC HỆ THỐNG THEO DÕI SỨC KHỎE	10
2.1 Giới thiệu về WSN [1].....	10
2.2 Các thành phần trong một hệ thống WSN	11
2.2.1 Cấu tạo của nút cảm biến – Sensor node	11
2.2.2 Hệ điều hành của các nút cảm biến.....	13
2.2.3 Nút Sink	13
2.3 Giới thiệu các công nghệ không dây áp dụng cho WSN.....	14
2.3.1 WLAN và chuẩn 802.11 [3]	14
2.3.2 802.15.4 và Zigbee [4] [5]	16
2.3.3 Bluetooth [4]	18
2.4 WSN trong các hệ thống theo dõi sức khỏe.....	19
2.4.1 Yêu cầu của một hệ thống theo dõi sức khỏe	19
2.4.2 Các tham số sức khỏe được theo dõi [7]	20
2.4.3 Các Sensor cảm biến hiện có trên thị trường.....	20
CHƯƠNG 3 THIẾT KẾ MẠNG CẢM BIẾN KHÔNG DÂY CHO HỆ THỐNG THEO DÕI SỨC KHỎE.....	23
3.1 Kiến trúc chung của hệ thống mạng WSN	23
3.1.1 Kiến trúc chung của WSN	23
3.1.2 Mô hình giao thức của WSN	25
3.2 Giao thức MAC trong các chuẩn truyền thông không dây	27
3.2.1 Giao thức MAC trong chuẩn 802.11	27

3.2.2 Giao thức MAC trong chuẩn 802.15.4 [5] [4]	30
3.3 Giao thức MAC trong WSN	32
3.3.1 Contention-based MAC [10]	33
3.3.2 Schedule-based MAC [10].....	36
3.4 Các giao thức định tuyến trong WSN.....	41
3.5 Hệ thống mạng WSN trong theo dõi sức khỏe	44
3.5.1 Mô hình mạng WSN trong theo dõi sức khỏe [13].....	44
3.5.2 Khảo sát một số hệ thống giám sát sức khỏe. [14]	45
CHƯƠNG 4 MÔ PHỎNG VÀ ĐÁNH GIÁ HIỆU NĂNG MẠNG	48
4.1 Giới thiệu các công cụ mô phỏng.....	48
4.1.1 OMNet++	48
4.1.2 TOSSIM.....	48
4.1.3 NS2 (Network Simulator 2).....	48
4.2 Vấn đề nguồn sinh lưu lượng trong WSN cho giám sát sức khỏe [16].....	50
4.3 Thiết lập kịch bản mô phỏng.....	51
4.3.1 Các tham số sinh học.....	51
4.3.2 Thiết lập topo mạng và các tham số mô phỏng	53
4.3.3 Các kịch bản mô phỏng	55
4.3.4 Thực hiện mô phỏng.....	56
4.4 Tổng kết việc mô phỏng	62
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TRONG TƯƠNG LAI	64
5.1 Kết luận.....	64
5.2 Hướng nghiên cứu trong tương lai	64
TÀI LIỆU THAM KHẢO.....	65
PHỤ LỤC.....	67

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

ACK	Acknowledgment
ADC	Analog to Digital Converter
APS	Application Provided Sublayer
BMAC	Berkeley MAC
CAP	Contention Access Period
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CID	Cluster ID
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CTS	Clear to Send
CH	Cluster Head
DARPA	Defense Advanced Research Projects Agency
DCF	Distributed Coordinator Function
DSN	Distributed Sensor Network
FFD	Full Function Device
GTS	Guaranteed Time Slot
IEEE	Institute of Electrical and Electronics Engineers
LEACH	Low-Energy Adaptive Clustering Hierachy
NAM	Network Animator
NAV	Network Allocation Vector
PAN	Personal Area Network
PCF	Point Coordinator Function
PEDAMACS	Power-Efficient and Delay-Aware Medium Access Protocol
PRIMA	Priority-Based MAC
RFD	Reduced-function Device
RTS	Request to Send
SMAC	Sensor MAC
SPIN	Sensor Protocols for Information via Negotiation
SQDDP	Sensor Query and Data Dissemination Protocol
TADAP	Task Assignment and Data Advertisement Protocol
TDMA	Time Division Multiple Access
ZC	Zigbee Coordinator
ZR	Zigbee Router

DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ

Hình 2-1 Cấu tạo của một nút cảm biến	12
Hình 2-2 Các giao thức truyền dẫn không dây phổ biến	14
Hình 2-3 Mô hình mạng WLAN có Access Point	15
Hình 2-4 Chồng giao thức ZigBee	17
Hình 2-5 Thí dụ một topo mạng ZigBee	18
Hình 2-6 Thí dụ một mạng cảm biến không dây mang được trên người	19
Hình 2-7 Thí dụ về một cảm biến đo oxy	21
Hình 2-8 Cảm biến nhịp tim EKG.....	22
Hình 3-1 Kiến trúc chung của WSN.....	23
Hình 3-2 Hai kiến trúc đặc trưng của WSN.....	24
Hình 3-3 Mô hình giao thức của WSN.....	26
Hình 3-4 Khuôn dạng gói tin RTS	28
Hình 3-5 Khuôn dạng gói tin CTS	28
Hình 3-6 Vấn đề hidden terminal và exposed terminal	29
Hình 3-7 Cơ chế DCF trong chuẩn IEEE 802.11.....	30
Hình 3-8 Cấu trúc siêu khung của chuẩn 802.15.4	31
Hình 3-9 Chu kỳ thức-ngủ của Sensor MAC.....	33
Hình 3-10 Quá trình đồng bộ của Sensor MAC.....	34
Hình 3-11 Giao thức B-MAC.....	36
Hình 3-12 Quá trình thiết lập cụm.....	39
Hình 3-13 Vòng phân chia thời gian của LEACH	39
Hình 3-14 Giao thức SPIN.....	42
Hình 3-15 Giao thức LEACH	43
Hình 3-16 Sơ đồ thời gian hoạt động của LEACH	43
Hình 3-17 Hệ thống WSN giám sát sức khỏe.....	44
Hình 4-1 Kiến trúc của NS-2	49
Hình 4-2 Phân phối chuẩn.....	50
Hình 4-3 Thống kê nhịp tim.....	51
Hình 4-4 Mô hình WSN cho giám sát sức khỏe	54
Hình 4-5 Tệp vết mô phỏng cảm biến gửi dữ liệu	57
Hình 4-6 Cấu trúc của tệp vết	57
Hình 4-7 Đồ thị thể hiện các dữ liệu cảm biến được theo kịch bản 1	58
Hình 4-8 Thời gian trễ truyền tin kịch bản 1	58
Hình 4-9 Biểu đồ thể hiện độ trễ truyền dữ liệu kịch bản 1	59
Hình 4-10 Độ suy hao năng lượng của nút cảm biến	59
Hình 4-11 Thời gian trễ truyền tin kịch bản 2.....	59
Hình 4-12 Thời gian trễ truyền dữ liệu, kịch bản 2.....	60
Hình 4-13 Mức suy hao năng lượng của kịch bản 2.	60
Hình 4-14 Các dữ liệu cảm biến vượt ngưỡng theo kịch bản 3	61
Hình 4-15 Độ trễ truyền tin theo kịch bản 3	61
Hình 4-16 Biểu đồ thể hiện độ trễ truyền dữ liệu kịch bản 3	61
Hình 4-17 Độ suy hao năng lượng của các nút mạng trong kịch bản 3	62
Hình 4-18 Biểu đồ sự suy hao năng lượng của nút mạng trong kịch bản 3	62

DANH MỤC BẢNG BIỂU

Bảng 2-1 Tóm tắt về các chuẩn trong công nghệ 802.11 [3].....	16
Bảng 2-2 Các tham số sức khỏe được theo dõi.....	20
Bảng 3-1 Khảo sát một số hệ thống giám sát sức khỏe [14]	47
Bảng 4-1 So sánh đặc điểm của Mannasim với một số công cụ mô phỏng khác [15] .	50
Bảng 4-2 Các tham số trung bình của nhịp tim.....	52
Bảng 4-3 Giá trị trung bình huyết áp.....	53
Bảng 4-4 Giá trị nhiệt độ trung bình	53
Bảng 4-5 Các tham số mô phỏng	54
Bảng 4-6 Kịch bản 1	55
Bảng 4-7 Kịch bản 2	56
Bảng 4-8 Kịch bản 3	56

CHƯƠNG 1 MỞ ĐẦU

1.1 Lý do chọn đề tài

Mạng cảm biến không dây hay Wireless Sensor Network là một tập hợp các nút cảm biến có khả năng tự trị, có thể cố định hoặc di động và thực hiện các nhiệm vụ khác nhau. Các nút mạng này có khả năng hoạt động độc lập, sử dụng cảm biến để thu thập các thông tin xung quanh nó và gửi dữ liệu đến các trạm xử lý xung quanh. Với đặc điểm linh động, dễ triển khai, giá thành rẻ, hoạt động tiêu thụ ít năng lượng, mạng cảm biến không dây hiện đang được ứng dụng trong rất nhiều lĩnh vực khác nhau. Cụ thể như: Quân sự, giám sát an ninh, giám sát môi trường, cảnh báo thiên tai, giám sát sức khỏe..v.v. Trong lĩnh vực y tế, mạng cảm biến không dây là lựa chọn không thể thay thế đối với việc giám sát, theo dõi sức khỏe bệnh nhân từ xa mà không bị giới hạn bởi các kết nối có dây. Với những ưu điểm như trên WSN đang có được sự phát triển mạnh mẽ và trở thành đề tài nghiên cứu của nhiều trường Đại Học trên thế giới. Chính vì vậy tôi đã quyết định chọn đề tài: Thiết kế và phân tích hiệu suất của mạng cảm biến không dây cho giám sát sức khỏe.

1.2 Mục tiêu của đề tài

Trong khuôn khổ luận văn này, tôi tập trung nghiên cứu về mạng cảm biến không dây, kiến trúc của mạng cảm biến không dây, ứng dụng thiết kế và phân tích hiệu suất của mạng cảm biến không dây phục vụ việc giám sát sức khỏe.

Với bài toán đặt ra là giám sát các thông số sức khỏe cơ bản của bệnh nhân thông qua các sensor gắn trên cơ thể. Dữ liệu sẽ được tổng hợp và gửi về bộ xử lý trung tâm để bác sĩ theo dõi, hoặc đưa ra các cảnh báo đến người bệnh. Trong mô hình này các thông số sức khỏe như: Nhiệt độ cơ thể, nhịp tim, Huyết áp v.v. sẽ được theo dõi thông qua các node cảm biến. Thông tin sẽ được hiển thị tại Terminal của máy tính xử lý trung tâm giúp người sử dụng có thể dễ dàng theo dõi và nhận biết được các cảnh báo cần thiết.

Trong bài toán này, các yếu tố như độ chính xác của dữ liệu, thời gian trễ, tần suất gửi báo cáo dữ liệu được quan tâm hàng đầu. Bằng sự trợ giúp của các công cụ mô phỏng tôi sẽ đưa ra các mô hình mạng đề xuất, đồng thời phân tích và đánh giá hiệu suất của hệ thống mạng thông qua 4 tiêu chí: 1) Độ tin cậy của dữ liệu, 2) Thời gian trễ truyền gói tin, 3) Khả năng xử lý dữ liệu lớn, 4) Sự thay đổi vị trí của các node cảm biến với các mô hình mạng khác nhau. Qua đó có thể đưa ra các đánh giá cũng như đề xuất các giải pháp làm tăng hiệu năng cho mạng cảm biến không dây phục vụ giám sát sức khỏe.

1.3 Tổ chức của luận văn

Luận văn này sẽ được trình bày trong 5 chương như sau:

Chương 1: Mở đầu. Giới thiệu về đề tài, mục tiêu và tổ chức của đề tài.

Chương 2: Mạng cảm biến không dây và ứng dụng trong các hệ thống giám sát sức khỏe. Trình bày tổng quan về mạng cảm biến không dây. Các chuẩn áp dụng cho mạng cảm biến không dây. Tổng quan về các ứng dụng của mạng cảm biến không dây trong hệ thống giám sát sức khỏe.

Chương 3: Thiết kế mạng cảm biến không dây cho hệ thống giám sát sức khỏe. Trình bày về kiến trúc, mô hình mạng cảm biến không dây cho hệ thống giám sát sức khỏe. Vai trò của các nút cảm biến. Các vấn đề về kết nối mạng.

Chương 4: Mô phỏng và phân tích hiệu suất. Trình bày về các công cụ mô phỏng, xây dựng chương trình mô phỏng hệ thống giám sát sức khỏe. Thực hiện mô phỏng, đưa ra kết quả và phân tích hiệu suất.

Chương 5: Kết luận và hướng nghiên cứu trong tương lai.

CHƯƠNG 2 MẠNG CẢM BIẾN KHÔNG DÂY VÀ ỨNG DỤNG TRONG CÁC HỆ THỐNG THEO DÕI SỨC KHỎE

Chương này bao gồm các thông tin giới thiệu về mạng cảm biến không dây – Wireless Sensor Network, các công nghệ không dây và một số ứng dụng của WSN trong các hệ thống theo dõi sức khỏe.

2.1 Giới thiệu về WSN [1]

Mạng cảm biến không dây có tên tiếng anh là Wireless Sensor Network. Nó bắt đầu được nghiên cứu và phát triển vào khoảng những năm 1980. Dự án được phát triển bởi DARPA – cơ quan các dự án phòng thủ tiên tiến của bộ quốc phòng Hoa Kỳ. Ban đầu dự án có tên là Distributed Sensor Networks (DSN). DSNs ban đầu được giả định là một mạng có nhiều nút cảm biến phân tán, có khả năng kết nối với nhau nhưng hoạt động độc lập, thông tin trong các nút cảm biến này có thể được định tuyến tới bất kỳ nút mạng nào trong phạm vi. Các thành phần của DSN được giới thiệu lần đầu vào năm 1978 bao gồm bộ phận cảm biến, mô đun xử lý dữ liệu và kết nối, phần mềm phân tán. Ứng dụng đầu tiên của DSN là một hệ thống theo dõi máy bay trực thăng được phát triển bởi học viện công nghệ Massachusetts.

Mặc dù bước đầu được phát triển từ rất sớm, cùng với đó là các mục tiêu cụ thể được đặt ra từ các nhà nghiên cứu, nhưng tại thời điểm đó DSN chưa sẵn sàng để hiện thực hóa các mục tiêu đề ra. Cụ thể hơn là bởi các nhược điểm của kích thước và giá thành bộ cảm biến, ít ứng dụng và đặc biệt DSN chưa hoàn toàn tương thích với các kết nối không dây.

Sự phát triển vượt bậc của ngành công nghiệp máy tính, truyền thông và công nghệ vi điện tử đã tạo nên một bước tiến mới cho DSN đến gần hơn với các mục tiêu đề ra ban đầu. Các nghiên cứu về DSN bắt đầu được chú ý trở lại vào năm 1998 và càng ngày càng thu hút nhiều nhà khoa học tham gia cũng như sự quan tâm của quốc tế. Các nghiên cứu mới về DSN lúc này chủ yếu tập trung vào các công nghệ mạng, các thuật toán xử lý dữ liệu và các phương pháp tiết kiệm tài nguyên cho nút cảm biến. Các nút cảm biến đã có kích thước nhỏ gọn và giá thành rẻ hơn trước. DSN lúc này đã phù hợp để triển khai các ứng dụng như theo dõi môi trường, mạng cảm biến giao thông, và hệ thống WSN theo dõi sức khỏe.

Định nghĩa WSN

WSN là một hệ thống mạng bao gồm các node cảm biến có kết nối không dây và khả năng tính toán.

Các node cảm biến thường là các sensor bao gồm các thành phần như: bộ vi xử lý, bộ phận cảm biến, bộ phận thu phát không dây, nguồn. Kích thước của các node

cảm biến thay đổi tùy thuộc vào loại ứng dụng, nhưng chúng có chung những đặc điểm sau:

- Sử dụng phương pháp phát thông tin quảng bá trong phạm vi hẹp và định tuyến đa chặng.
- Kích thước vật lý nhỏ, giá thành rẻ, chủ yếu sử dụng pin. Do vậy các node cảm biến bị hạn chế về khả năng xử lý cũng như dung lượng nhớ.
- Vị trí các node mạng cảm biến có thể được phân bố ngẫu nhiên không cần xác định trước.
- Có thể xử lý dữ liệu đơn giản.
- Hoạt động ổn định và đáng tin cậy.

WSN có rất nhiều ứng dụng trong đời sống hiện nay của con người. Cụ thể là [2]:

- Y tế: Mạng cảm biến có thể được sử dụng để theo dõi các thông số sinh lý của bệnh nhân như nhịp tim, huyết áp, nhiệt độ cơ thể từ xa. Và gửi báo cáo đến bệnh viện khi cần thiết.
- Nông nghiệp: Mạng cảm biến có thể được sử dụng để theo dõi điều kiện khí hậu của một diện tích canh tác lớn. Tính toán nhu cầu nước, hóa chất của cây trồng và đưa ra thông báo.
- Quân sự: Mạng cảm biến không dây có thể được sử dụng để giám sát các khu vực nguy hiểm mà không cần sự có mặt của con người. Phát hiện chuyển động hay vị trí của đối phương, phát hiện các cuộc tấn công hóa sinh và dò tìm bom mìn.
- Môi trường: Mạng cảm biến có thể được sử dụng để theo dõi các biến đổi khí hậu, cảnh báo thiên tai, .v.v

Ngoài ra còn rất nhiều ứng dụng khác có thể được triển khai dựa vào những ưu điểm của hệ thống mạng cảm biến không dây mang lại tiện ích cho con người. Chính vì vậy việc nghiên cứu và phát triển mạng cảm biến không dây đang là đề tài nóng trong những năm gần đây.

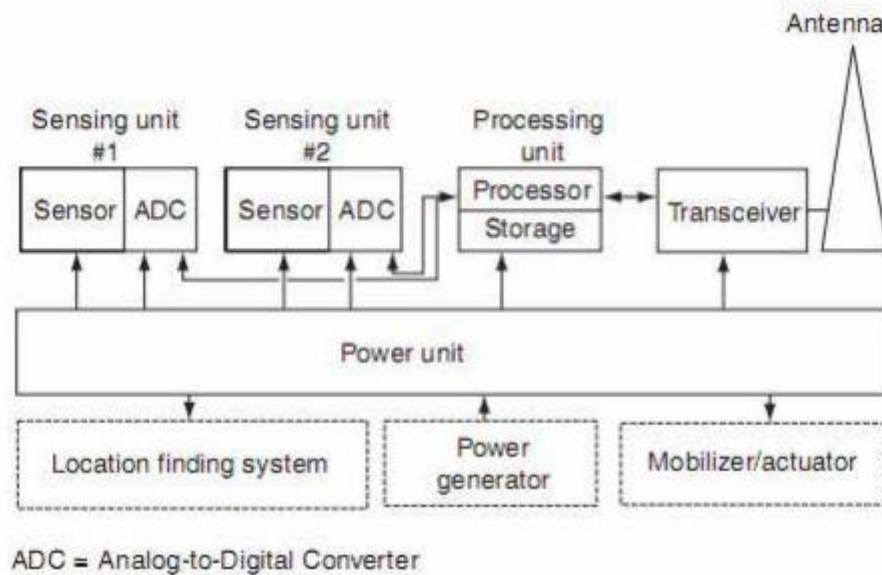
2.2 Các thành phần trong một hệ thống WSN

Hệ thống WSN bao gồm 3 thành phần cơ bản: Source hay Sensor Node là các nút làm nhiệm vụ cảm biến, kết hợp và gửi dữ liệu. Sink Node là các nút chỉ làm nhiệm vụ thu thập dữ liệu từ source và chuyển cho AP. Sink Node cũng có thể làm nhiệm vụ trung gian giữa WSNs và các hệ thống mạng khác. AP làm nhiệm vụ trung gian trung chuyển dữ liệu, đôi khi vai trò của AP không rõ ràng.

2.2.1 Cấu tạo của nút cảm biến – Sensor node

Một hệ thống WSN là một tập hợp của rất nhiều nút cảm biến phân tán trong một khoảng diện tích rộng. Mỗi nút cảm biến đều có khả năng thực hiện một số tác vụ cảm biến và xử lý dữ liệu độc lập. Sau đó nó sẽ thực hiện kết nối với các nút lân cận để gửi dữ liệu cảm biến về bộ xử lý trung tâm. Thông thường một nút cảm biến sẽ bao

gồm những thành phần như: Bộ phận thu phát sóng, bộ xử lý, bộ nhớ, một hoặc nhiều đơn vị cảm biến và bộ nguồn. [1]



Hình 2-1 Cấu tạo của một nút cảm biến

- Bộ xử lý: Có nhiệm vụ lập lịch cho các tác vụ, xử lý dữ liệu và điều khiển các thành phần khác. Các kiểu bộ xử lý có thể được sử dụng trong WSN: Microcontroller, Digital Signal Processor, FPGA (Field Program Gate Array), .v.v. Trong đó Microcontroller được sử dụng nhiều hơn cả. Ví dụ như cảm biến Mica2 Mote sử dụng bộ xử lý ATmega 128L microcontroller.
- Bộ thu phát tín hiệu: Chịu trách nhiệm cho việc thu phát tín hiệu không dây. Có thể sử dụng sóng Radio, Laser hoặc hồng ngoại. Sóng Radio được sử dụng phổ biến hơn cả vì nó phù hợp với yêu cầu của hầu hết ứng dụng trong WSN. Bộ thu phát tín hiệu có các trạng thái như: Transmit, Receive, Idle và Sleep.
- Bộ nhớ: Bao gồm các chip nhớ flash và RAM của bộ vi xử lý. Có thể có cả các thẻ nhớ ngoài. Ví dụ chip ATmega 128L microcontroller của Mica 2 Mote sử dụng 128 Kbyte bộ nhớ trong và 4 Kbyte RAM.
- Bộ nguồn: Trong một nút cảm biến, các hoạt động như cảm ứng, xử lý và truyền dữ liệu đều tiêu thụ năng lượng. Trong đó việc truyền dữ liệu tiêu thụ nhiều năng lượng hơn so với các hoạt động khác. Pin là nguồn năng lượng chủ yếu được sử dụng trong các bộ nguồn của cảm biến. Ví dụ cảm biến Mica2 Mote sử dụng 2 viên pin AA. Vì nguồn năng lượng bị giới hạn việc tiết kiệm năng lượng cho các nút cảm biến đang là mục tiêu quan tâm hàng đầu trong hoạt động của các WSN. Đã có rất nhiều nghiên cứu và đề xuất cải tiến nhằm tăng cường thời lượng pin của các

nút cảm biến. Người ta cũng tính toán đến việc sử dụng các nguồn năng lượng tự nhiên như ánh sáng mặt trời cho các nút cảm biến.

- Nút cảm biến: Là một thiết bị phần cứng có nhiệm vụ đo các thông số vật lý ở môi trường xung quanh nó như nhiệt độ, áp suất, độ ẩm, v.v. Dữ liệu tương tự được cảm ứng bởi các cảm biến và được số hóa bằng bộ ADC sau đó được gửi cho bộ xử lý để thực hiện các tác vụ tiếp theo. Các cảm biến là các thiết bị vi điện tử vì vậy chúng có kích thước nhỏ và tiêu thụ rất ít năng lượng. Có thể có một hoặc nhiều loại cảm biến gắn trên một nút cảm biến.

2.2.2 Hệ điều hành của các nút cảm biến

Hệ điều hành là một môi trường để các chương trình ứng dụng thực hiện công việc của mình một cách đơn giản và hiệu quả mà không phải giao tiếp với phần cứng. Trong WSN vai trò của hệ điều hành đơn giản hơn so với các hệ điều hành thông thường. Lí do là bởi các giới hạn về mặt phần cứng của thiết bị cũng như các yêu cầu ít phức tạp của các ứng dụng.

TinyOS là hệ điều hành đầu tiên được thiết kế dành cho hệ thống WSNs hay cụ thể hơn là dành cho các nút mạng. Các thư viện bên trong TinyOS bao gồm: các bộ giao thức mạng, các dịch vụ phân tán, trình điều khiển thiết bị và công cụ thu thập dữ liệu. Không giống như hầu hết các hệ điều hành khác có khả năng chạy đa nhiệm TinyOS được phát triển theo mô hình lập trình hướng sự kiện. Cả TinyOS và các chương trình ứng dụng được viết cho TinyOS đều sử dụng một loại ngôn ngữ lập trình đặc biệt có tên gọi là nesC – một phiên bản mở rộng của ngôn ngữ lập trình C. Ngày nay TinyOS đã được biết đến và sử dụng rộng rãi trong rất nhiều nền tảng cảm biến khác nhau. Bởi đặc tính mã nguồn mở của nó, cộng đồng sử dụng TinyOS cũng phát triển mạnh mẽ với hàng trăm nhóm nghiên cứu lớn nhỏ.

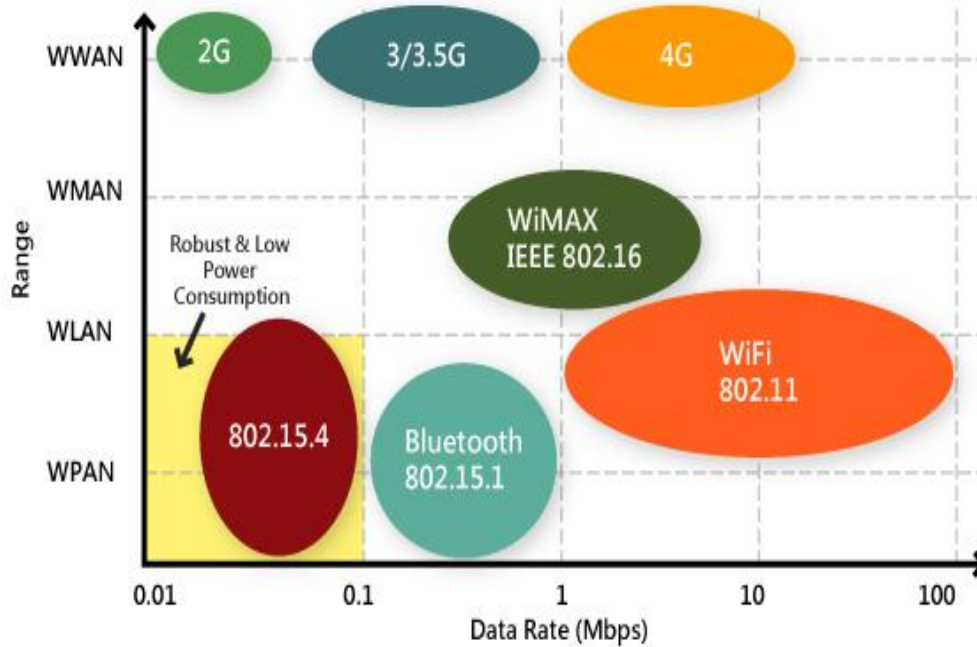
Một đại diện khác của các hệ điều hành dành cho WSN có tên gọi là Contiki. Contiki cũng sử dụng cấu trúc kernel hướng sự kiện như TinyOS tuy nhiên hệ điều hành này cũng hỗ trợ đa nhiệm với một số ứng dụng cơ bản. Ngoài ra Contiki còn hỗ trợ cả IPv4 và IPv6. Rất nhiều các cơ chế và cải tiến của Contiki được áp dụng rộng rãi.

2.2.3 Nút Sink

Là nút chịu trách nhiệm tương tác với các nút cảm biến. Các dữ liệu từ các nút cảm biến được xử lý và chuyển đến các Base Station hoặc đến các AP. Nút Sink dựa trên bộ vi xử lý có khả năng xử lý tín hiệu kỹ thuật số, bao gồm các mô-đun khác nhau như ăng ten, bộ thu phát, bộ xử lý, bộ nhớ flash, khối đầu vào, màn hình hiển thị và pin với mạch sạc. Các dữ liệu nhận là có điều kiện và tiếp tục được xử lý để loại bỏ các nhiễu, và tính toán các thông số. Dữ liệu y tế có đánh mốc thời gian, được lưu trữ trong bộ nhớ và truyền đến một trạm giám sát từ xa để phân tích thêm. Nút Sink thường được trang bị một pin sạc cho thời gian hoạt động tối thiểu 8-12h.

2.3 Giới thiệu các công nghệ không dây áp dụng cho WSN

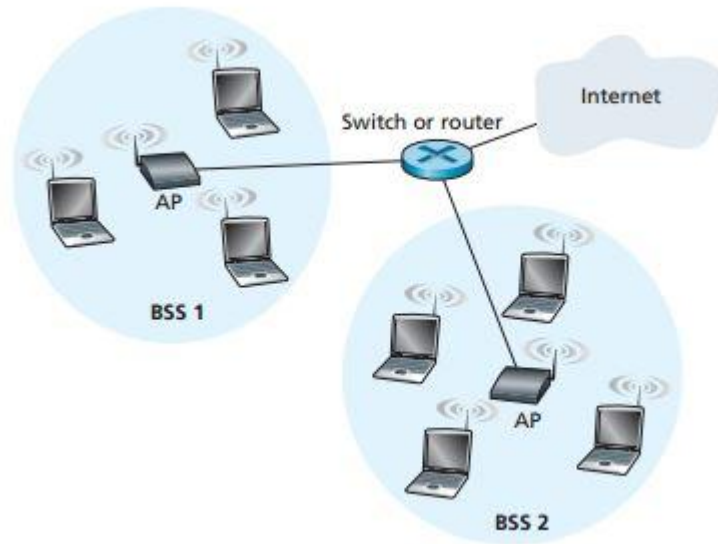
WSN có thể sử dụng một số công nghệ truyền dẫn không dây được thiết kế sẵn như Bluetooth, Zigbee, 802.15.4, Wireless LAN 802.11, Broadband Wireless Access/WiMax và 3G. Trong phần này tôi sẽ trình bày về 3 công nghệ không dây thông dụng và được áp dụng nhiều trong các hệ thống WSN.



Hình 2-2 Các giao thức truyền dẫn không dây phổ biến

2.3.1 WLAN và chuẩn 802.11 [3]

WLAN hay wireless local area network là mạng LAN không dây sử dụng công nghệ 802.11 còn có tên gọi khác là Wi-Fi. WLAN được thiết kế dựa trên kiến trúc mạng tế bào (cellular), trong đó hệ thống được chia thành các cells, mỗi cells được điều khiển bởi một Base Station hay còn gọi là Access Point. Mặc dù wireless LAN có thể được hình thành chỉ với một cells và một AP, hoặc thậm chí không cần AP. Tuy nhiên đa số các hệ thống WLAN đều được triển khai với nhiều cells trong đó AP kết nối với các hệ thống phân tán khác thông qua cáp Ethernet hoặc cũng có thể là kết nối không dây.



Hình 2-3 Mô hình mạng WLAN có Access Point

Các hệ thống WLAN hoạt động dựa trên tập hợp các chuẩn IEEE 802.11 được công bố vào năm 1997. Chuẩn 802.11 cũng giống với các chuẩn khác trong họ 802, nó đặc tả hoạt động mạng tại 2 tầng Physical và Data Link trong mô hình tham chiếu OSI. Do đó tất cả hệ thống mạng thiết kế theo chuẩn 802 đều có 02 thành phần chính là MAC và PHY, trong đó MAC là một tập hợp các luật định nghĩa việc truy xuất và gửi dữ liệu còn PHY phụ trách việc truyền dẫn và thu nhận dữ liệu. Khi mới được công bố vào năm 1997, công nghệ 802.11 có 02 phiên bản là 802.11a hoạt động tại dải tần 5GHz và 802.11b hoạt động tại dải tần 2.4GHz. Sau đó với sự phát triển của khoa học công nghệ các phiên bản khác của 802.11 lần lượt được ra đời với những cải tiến và tính năng khác nhau. Bảng 2-1 Tóm tắt về các chuẩn trong công nghệ 802.11 [3] sẽ chỉ ra một số đặc điểm chính của từng phiên bản.

Chuẩn	Năm phát triển	Một số đặc điểm chính
IEEE 802.11	1997	Hoạt động lại layer 2 trong mô hình OSI
		Công nghệ trải phổ FHSS sử dụng tần số 2.4GHz tốc độ 1-2Mbps
		Công nghệ trải phổ DSSS sử dụng tần số 2.4GHz tốc độ 1-2Mbps
IEEE 802.11a	1999	Sử dụng phương pháp điều chế OFDM. Hoạt động ở dải tần 5GHz, có thể đạt tốc độ từ 6 ~ 54Mbps
IEEE 802.11b	1999	Công nghệ trải phổ DSSS. Sử dụng tần số 2.4GHz tốc độ 5.5 ~ 11Mbps
IEEE 802.11c	2003	Chế độ Bridge của 802.11 MAC layer
IEEE 802.11d	2001	Physical: Mở rộng hoạt động của 802.11 WLANs với các miền quy định.
IEEE 802.11e	Đang phát triển	MAC: Mục tiêu cải tiến QoS và các cơ chế bảo mật
IEEE 802.11f	Đang phát triển	Đề xuất khả năng tương tác giữa các AP

IEEE 802.11g	2003	Mở rộng 802.11b với tốc độ > 20Mbps
IEEE 802.11h	Đang phát triển	Physical/MAC: Cải tiến 802.11a thêm tính năng lựa chọn kênh indoor hoặc outdoor, tính năng quản lý công suất hoạt động
IEEE 802.11i	Đang phát triển	MAC: Cải tiến bảo mật và cơ chế xác thực
IEEE 802.11j	Đang phát triển	Physical: Cải tiến 802.11a theo chuẩn của Nhật Bản
IEEE 802.11k	Đang phát triển	Chuẩn đánh giá tài nguyên Radio
IEEE 802.11m	Đang phát triển	Bảo trì chuẩn IEEE 802.11-1999 chỉnh sửa các vấn đề về công nghệ và biên tập.
IEEE 802.11n	Đang phát triển	Physical/MAC: Cải tiến cho phép tăng thông lượng.

Bảng 2-1 Tóm tắt về các chuẩn trong công nghệ 802.11 [3]

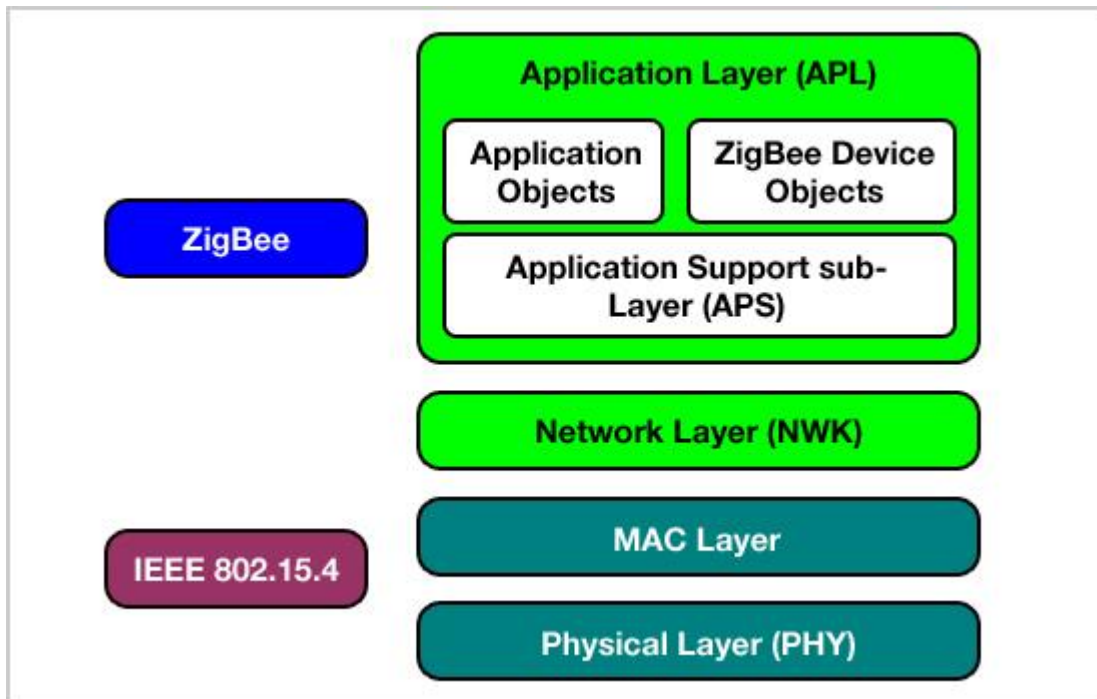
Với các ưu điểm về tính di động, tính mềm dẻo, dễ dàng triển khai và mở rộng, tiết kiệm chi phí. WLAN đang ngày càng trở nên phổ biến hơn trong đời sống. Đặc biệt trong môi trường mạng gia đình và mạng doanh nghiệp. Tuy nhiên với các hạn chế về bảo mật, tốc độ truy cập hay phạm vi phủ sóng cũng đã ảnh hưởng không nhỏ đến hiệu năng của các hệ thống mạng. Vì vậy cần có thêm những nghiên cứu, cải tiến để có thể khắc phục được những vấn đề này.

2.3.2 802.15.4 và Zigbee [4] [5]

Zigbee là một công nghệ mạng không dây cá nhân được thiết kế theo chuẩn IEEE 802.15.4. Trong đó chuẩn 802.15.4 là chuẩn truyền thông có tốc độ truyền dữ liệu thấp, tiêu hao ít năng lượng, dễ dàng di chuyển.

Không như các tiêu chuẩn mạng không dây khác chủ yếu phục vụ các hệ thống mạng cần tốc độ dữ liệu cao và tiêu thụ nhiều năng lượng đồng thời chi phí sản xuất cũng như lắp đặt rất lớn. Có rất nhiều hệ thống theo dõi giám sát chỉ ưu tiên vào việc tiêu thụ ít năng lượng, hoạt động đơn giản chứ không cần tốc độ truyền dữ liệu cao. Ví dụ như các hệ thống giám sát theo dõi nhiệt độ, theo dõi sức khỏe v.v.

ZigBee được phát triển bởi ZigBee Alliance là một tổ chức các công ty viễn thông độc lập không phụ thuộc vào IEEE. Trên thực tế ZigBee được xây dựng trên nền tảng của chuẩn IEEE 802.15.4. Cụ thể, giống như các nhóm phát triển 802 khác, 802.15.4 chỉ định nghĩa lớp MAC và lớp PHY. Dựa trên nền tảng đó ZigBee xây dựng thêm lớp Network và lớp Application đồng thời bổ sung thêm thông tin vào 2 lớp đã có. Tạo thành một công nghệ mạng LR-WPANs của riêng mình.



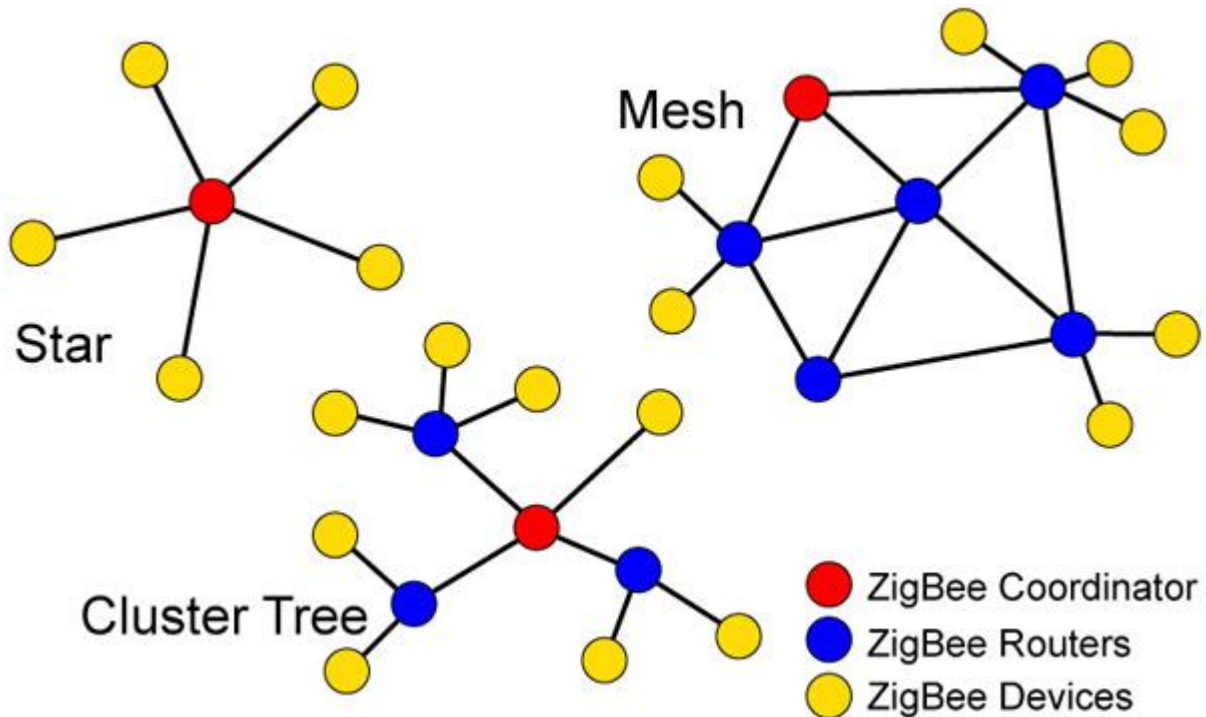
Hình 2-4 Chồng giao thức ZigBee

ZigBee có đầy đủ các đặc điểm của một hệ thống LR-WPANs đó là: dễ dàng triển khai, độ tin cậy cao, hoạt động tiêu thụ năng lượng thấp giúp kéo dài thời lượng pin. Đồng thời bộ giao thức hoạt động đơn giản và linh hoạt giúp dễ dàng kết hợp với các ứng dụng ở tầng trên.

Một hệ thống ZigBee gồm nhiều thành phần tạo nên. Bao gồm các thiết bị như: ZigBee Coordinator(ZC), ZigBee Router(ZR), và ZigBee End Device. Trong đó, thiết bị coordinator và router còn được gọi là full-function device(FFD). Thiết bị này đảm nhận tất cả các chức năng trong mạng và hoạt động như một bộ điều phối mạng. Các thiết bị ZigBee end device với các chức năng hạn chế hơn còn có tên gọi khác là reduced-function device (RFD). Một mạng ZigBee tối thiểu phải có một thiết bị FFD.

FFD có thể hoạt động trong 03 trạng thái: Điều phối toàn mạng, điều phối mạng con, hoặc hoạt động như một RFD. Một FFD có thể làm việc với nhiều RFD hoặc với FFD khác. RFD chỉ làm việc được với một FFD quản lý nó.

Các nút mạng trong một mạng ZigBee có thể liên kết với nhau theo cấu trúc mạng hình sao (Star), hình cây(Tree), và mắt lưới(Mesh).



Hình 2-5 Thí dụ một topo mạng ZigBee

- Star topology: Đối với mô hình này, tất cả các nút trong mạng đều kết nối đến một trung tâm điều khiển, nó được gọi là bộ điều phối mạng PAN. Một PAN là do một FFD hình thành, sau đó các FFD và RFD khác đều có thể kết nối đến nó. Mỗi mạng hình sao đều phải có một chỉ số nhận dạng cá nhân được gọi là PAN.
- Mesh Topology: Mô hình này cũng sử dụng một bộ điều phối mạng PAN. Tuy nhiên các nút mạng FFD trong mạng đều có thể kết nối với các FFD khác trong tầm phủ sóng của nó. Thực chất đây là mô hình kết hợp giữa mạng hình sao và mạng ngang hàng. Mô hình này có thể ứng dụng trong các mạng cảm biến không dây theo dõi và cảnh báo.
- Cluster-Tree: Mô hình này là một dạng đặc biệt của cấu trúc mắt lưới. Trong đó bộ điều phối mạng PAN tự bầu ra nút mạng quản lý mạng, và gán cho nút mạng này một chỉ số đặc biệt là CID-0, nó phát thông tin quảng bá đến các thiết bị lân cận. Thiết bị nào nhận được khung tin này có thể yêu cầu kết nối vào mạng với Cluster Head (CH). Nếu bộ điều phối đồng ý nó sẽ ghi tên thiết bị đó vào danh sách. Sau đó thiết bị mới này lại trở thành CH của nhánh cây mới. Các RFD có thể kết nối vào cluster như một nốt rời rạc ở điểm cuối của nhánh.

2.3.3 Bluetooth [4]

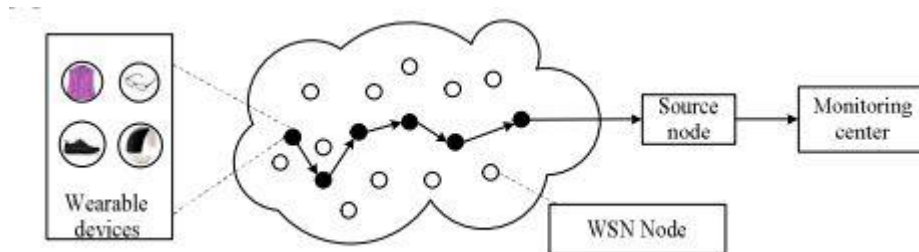
Bluetooth là công nghệ không dây cho phép các thiết bị điện, điện tử như máy tính xách tay, điện thoại di động, các thiết bị ngoại vi có thể giao tiếp với nhau trong khoảng cách ngắn, bằng sóng vô tuyến qua băng tần chung. Bluetooth hoạt động ở dải tần 2.4GHz ISM đây là dải tần không cần đăng ký để dùng riêng cho các thiết bị

không dây dùng trong công nghiệp, y tế, khoa học. Ban đầu bluetooth được thiết kế để thay thế cáp kết nối giữa máy tính và các thiết bị điện tử một cách nhanh chóng và tiện lợi. Khi được kích hoạt Bluetooth có thể tự động định vị những thiết bị khác có sử dụng chung công nghệ ở xung quanh.

Đặc điểm của Bluetooth là công suất tiêu thụ thấp, giá thành thấp, tốc độ truyền dữ liệu có thể đạt tối đa 25Mbps với phiên bản Bluetooth 4.0 và còn đang tiếp tục được nâng cao, tính tương thích cao hỗ trợ nhiều loại phần cứng hoặc phần mềm, dễ dàng phát triển ứng dụng.

2.4 WSN trong các hệ thống theo dõi sức khỏe

Hệ thống mạng cảm biến không dây giám sát sức khỏe bao gồm một loạt các cảm biến được gắn trên quần áo hoặc trên cơ thể bệnh nhân để liên tục theo dõi các thông số sức khỏe và truyền dữ liệu đến một trạm giám sát từ xa. Hình 2-6 Thí dụ một mạng cảm biến không dây mang được trên người mô tả một hệ thống theo dõi sức khỏe dựa trên mạng cảm biến không dây [6].



Hình 2-6 Thí dụ một mạng cảm biến không dây mang được trên người

2.4.1 Yêu cầu của một hệ thống theo dõi sức khỏe

Để thực hiện theo dõi sức khỏe của con người, việc sử dụng một hệ thống mạng cảm biến không dây là một lựa chọn tối ưu. Bởi các kết nối cáp sẽ giới hạn chuyển động của con người ảnh hưởng đến các tham số sinh lý được theo dõi. Một hệ thống mạng không dây theo dõi sức khỏe cần đảm bảo được các yếu tố sau:

a, Độ tin cậy trong truyền dữ liệu

Một trong những mục tiêu chính của hệ thống theo dõi sức khỏe là có được những cảnh báo sớm và chính xác các thông số về sinh lý trên cơ thể người để từ đó có những chẩn đoán và điều trị sớm. Do vậy khi thiết kế hệ thống mạng chúng ta cần quan tâm đến tính chính xác của cảm biến, chất lượng đường liên kết và độ tin cậy trong các giao thức định tuyến để tránh tình trạng mất mát hoặc lỗi gói tin gây sai lệch dữ liệu.

b, Dữ liệu thời gian thực

Bao gồm 03 khía cạnh: 1) thu thập dữ liệu thời gian thực, độ nhạy cảm của cảm biến và phạm vi cảm biến. 2) truyền dữ liệu thời gian thực, chủ yếu tập trung vào các giao thức định tuyến. 3) xử lý dữ liệu thời gian thực, tập trung vào xử lý dữ liệu kịp thời.

c, Xử lý dữ liệu lớn

Thông thường để theo dõi sức khỏe thì cần ít nhất 04 bộ cảm biến sinh học trên cơ thể. So với hệ thống cảm biến môi trường thì mạng cảm biến phục vụ theo dõi sức khỏe có nhu cầu về băng thông cao hơn. Đặc biệt hệ thống thực hiện thu thập dữ liệu theo thời gian thực nên tạo ra lượng lớn dữ liệu đòi hỏi mạng phải có băng thông cao để chịu được tải.

d, Thay đổi topology

Hệ thống mạng cảm biến không dây theo dõi sức khỏe phải có khả năng thích ứng với những thay đổi thường xuyên của topology, sự chuyển động của các node mạng rời mạng, tham gia vào mạng,.v.v.

2.4.2 Các tham số sức khỏe được theo dõi [7]

Các thông số sinh lý thông thường cần theo dõi là điện tâm đồ (ECG), nhịp tim có nguồn gốc từ các tín hiệu ECG bằng cách xác định các khoảng R-R, huyết áp, nhiệt độ cơ thể, phản xạ trên da (GSR), độ bão hòa oxy trong máu (SaO₂), tỷ lệ hô hấp, điện đồ cơ (EMG), đo điện não đồ (EEG) và chuyển động trực ba của đối tượng. Bảng 2-2 Các tham số sức khỏe được theo dõi liệt kê một số tham số sức khỏe của con người được theo dõi trong một hệ thống WSN.

Physical Parameter	Specifications
Electrocardiogram (ECG)	Frequency: 0.5Hz – 100 Hz Amplitude: 0.25 – 1mV
Electromyogram (EMG)	Frequency: 10Hz - 3KHz Amplitude: 50 μ V – 1mV
Electroencephalogram (EEG)	Frequency: 0.5Hz - 100Hz Amplitude: 1 μ V – 100 μ V
Blood Pressure (BP)	Systolic: 60 - 200mmHg Diastolic: 50 – 110mmHg
Body Temperature	32°C – 40°C
Galvanic Skin Response (GSR)	0 – 100 K Ω
Respiratory Rate (RR)	2 –50 breaths/min Frequency 0.1 – 10Hz
Oxygen Saturation in Blood (SaO ₂)	0-100%
Heart Rate (HR)	40 – 220 Beats per minute

Bảng 2-2 Các tham số sức khỏe được theo dõi

2.4.3 Các Sensor cảm biến hiện có trên thị trường

a, Cảm biến đo ô-xy kiểu xung (Pulse oximeter) [8]

Bộ đo oxy kiểu xung đã được sử dụng như một kỹ thuật chẩn đoán y tế kể từ khi nó được phát minh ra vào đầu năm 1970. Công nghệ không xâm nhập cơ thể này được sử dụng để đánh giá đáng tin cậy hai số liệu sức khỏe bệnh nhân quan trọng: nhịp tim - HR (Heart Rate) và độ bão hòa oxy trong máu - SpO₂. Các thông số mang thông

tin quan trọng, đặc biệt trong trường hợp khẩn cấp khi một sự thay đổi đột ngột nhịp tim, giảm oxy trong máu có thể cho thấy cần có sự can thiệp của y tế khẩn cấp. Xung oxy có thể cung cấp cảnh báo trước khi bắt đầu thiếu oxy thậm chí trước khi bệnh nhân biểu hiện triệu chứng thể chất.



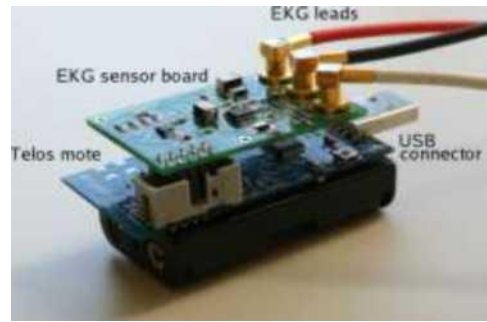
Hình 2-7 Thí dụ về một cảm biến đo oxy

Công nghệ đo oxy kiểu xung thực hiện việc chiếu tia hồng ngoại và cận hồng ngoại qua các mạch máu gần da. Các xung oxy đặt trong hộp nhựa kẹp ngón tay trở hoặc dải tai. Những hộp nhựa có chứa một loạt các đèn LED cùng một bề mặt bên trong và một bộ cảm biến quang điện tử đối diện. Bằng cách đo lượng ánh sáng được hấp thu bởi huyết sắc tố trong máu ở hai bước sóng khác nhau (thường là 650nm và 805nm), mức độ bão hòa oxy có thể xác định được. Thêm vào đó, nhịp tim có thể được xác định từ mô hình hấp thu ánh sáng theo thời gian. Tính toán về HR và SpO₂ từ các dạng sóng truyền ánh sáng có thể được thực hiện sử dụng kỹ thuật xử lý tín hiệu kỹ thuật số tiêu chuẩn (DSP). Một số thuật toán phức tạp đã được phát triển để giảm thiểu lỗi do vật chuyển động. Ngày nay công nghệ này đã được tích hợp trên một số thiết bị khác như Smart Phone hay vòng đeo tay thông minh.

b, Điện tim (EKG) [8]

Hai loại khác nhau của điện tâm đồ (EKG hay ECG) thường được sử dụng trong y tế để đo hoạt động của tim. Thiết bị này ghi lại một mẫu ngắn (không quá 30 giây) hoạt động của điện tim giữa các cặp điện cực khác nhau. Mỗi cặp điện cực cung cấp một hình ảnh duy nhất và chi tiết về nhịp tim, tiếng vọng riêng lẻ của xung điện của tim khi chúng được thực hiện thông qua mô xung quanh. Một bác sĩ tim mạch có kinh nghiệm có thể nhanh chóng giải thích một điện tâm đồ tiêu chuẩn để chẩn đoán một loạt các rối loạn nhịp tim, cũng như thiếu máu cục bộ cơ tim cấp và nhồi máu.

Mote-based EKG là loại cảm biến được phát triển dựa trên nền tảng Mica2/MicaZ và Telos cung cấp theo dõi điện tâm đồ liên tục bằng cách đo sự khác biệt trên một cặp điện cực. Hình 2-8 là ví dụ của một cảm biến nhịp tim EKG được thiết kế mạch tích hợp của Texas Instruments INA321 CMOS.



Hình 2-8 Cảm biến nhịp tim EKG

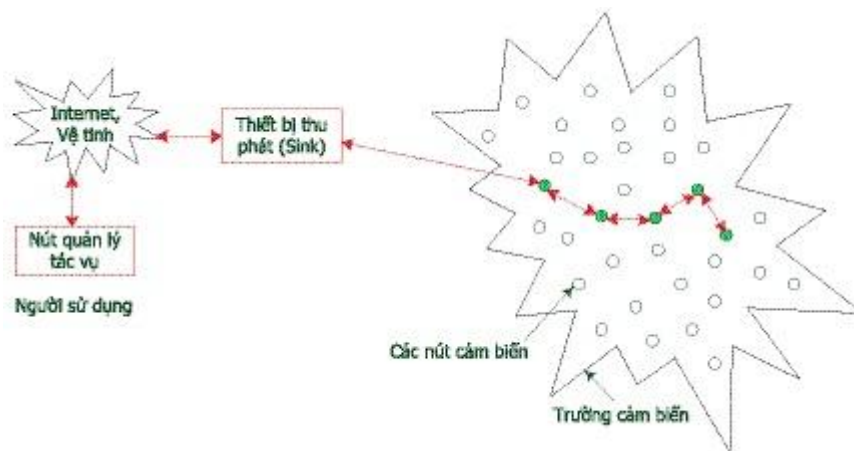
Có 3 điện cực được gắn vào trên và dưới ngực của bệnh nhân, một được đặt trên da của bệnh nhân trong khi hai điện cực khác được sử dụng để đo hoạt động của tim. Bộ khuếch đại và lọc tín hiệu INA321 cho phép lọc ra gần như tất cả các nhiễu thông thường. Các tín hiệu sau đó được khuếch đại thêm và được chuyển đến một cổng đầu vào của ADC.

CHƯƠNG 3 THIẾT KẾ MẠNG CẢM BIẾN KHÔNG DÂY CHO HỆ THỐNG THEO DÕI SỨC KHỎE

3.1 Kiến trúc chung của hệ thống mạng WSN

3.1.1 Kiến trúc chung của WSN

Các node cảm biến được phân bố trong một trường sensor được minh họa trên Hình 3-1 Kiến trúc chung của WSN. Mỗi node cảm biến có khả năng thu thập dữ liệu và định tuyến đến các sink. Dữ liệu được định tuyến đến các sink bởi một cấu trúc đa điểm như hình dưới. các sink có thể giao tiếp với các node quản lý nhiệm vụ (task manager node) qua mạng Internet hoặc vệ tinh.

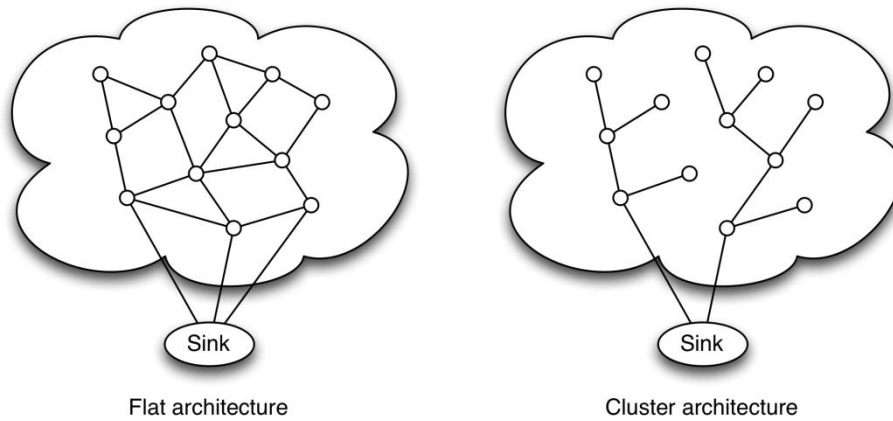


Hình 3-1 Kiến trúc chung của WSN

Do đặc điểm và ứng dụng của mạng WSN nên Tô pô của mạng cảm biến không dây thường được sử dụng với 2 dạng đó là: dạng ngang hàng và dạng hình sao, phân cụm.

- Dạng ngang hàng: ít được sử dụng hơn vì đặc điểm chính của mạng sensor thu thập dữ liệu nên vai trò của các nút trong WSN thường không giống nhau.

- Dạng hình sao: Đây là dạng phổ biến trong WSN, đối với mạng WSN đơn giản nhất là các nút cảm biến dữ liệu và chuyển về một nút trung tâm bằng phương pháp trực tiếp hoặc gián tiếp qua một số nút trung gian. Mở rộng của dạng hình sao là dạng cụm hay là cây phân cấp, các nút sensor được phân thành từng cụm và theo phân cấp thành từng mức khác nhau.



Hình 3-2 Hai kiến trúc đặc trưng của WSN

a, Kiến trúc phẳng:

Trong cấu trúc phẳng (flat architectute – Hình 3-2 Hai kiến trúc đặc trưng của WSN): tất cả các node đều ngang hàng và đồng nhất trong hình dạng và chức năng. Các node giao tiếp với sink qua multihop sử dụng các node ngang hàng làm bộ tiếp sóng. Với phạm vi truyền cố định, các node gần sink hơn sẽ đảm bảo vai trò của bộ tiếp sóng đối với một số lượng lớn nguồn. Giả thiết rằng tất cả các nguồn đều dùng cùng một tần số để truyền dữ liệu, vì vậy có thể chia sẻ đường truyền theo thời gian. Tuy nhiên cách này chỉ có hiệu quả với điều kiện là có nguồn tài nguyên chia sẻ đơn lẻ, ví dụ như thời gian, tần số.

b, Kiến trúc tầng:

Trong Cấu trúc tầng (Cluster architecture - Hình 3-2 Hai kiến trúc đặc trưng của WSN): các cụm được tạo ra giúp các nút nguồn trong cùng một cụm gửi dữ liệu theo phương thức single hop hay multihop (tùy thuộc vào kích cỡ của cụm) đến một node định sẵn, thường gọi là node cụm trưởng (cluster head). Trong cấu trúc này các node tạo thành một hệ thống cấp bậc mà ở đó mỗi node ở một mức xác định thực hiện các nhiệm vụ đã định sẵn.

Trong cấu trúc tầng thì chức năng cảm nhận, tính toán và phân phối dữ liệu không đồng đều giữa các node. Những chức năng này có thể phân theo cấp, cấp thấp nhất thực hiện nhiệm vụ cảm nhận, cấp giữa thực hiện tính toán, và cấp trên cùng thực hiện phân phối dữ liệu

Mạng cảm biến xây dựng theo cấu trúc tầng hoạt động hiệu quả hơn so với cấu trúc phẳng, vì một số lý do sau:

- Cấu trúc tầng có thể giảm chi phí cho mạng cảm biến bằng việc định vị các tài nguyên ở vị trí mà chúng hoạt động hiệu quả nhất. Rõ ràng là nếu triển khai các phần cứng thông nhất. Mỗi node chỉ cần một lượng tài nguyên tối thiểu để thực hiện tất cả các nhiệm vụ.

- Mạng cấu trúc tầng có tuổi thọ cao hơn cấu trúc mạng phẳng. Khi cần phải tính toán nhiều thì một bộ xử lý nhanh sẽ hiệu quả hơn, phụ thuộc vào thời gian yêu cầu thực hiện tính toán. Tuy nhiên, với các nhiệm vụ cảm nhận cần hoạt động trong khoảng thời gian dài, các node tiêu thụ ít năng lượng phù hợp với yêu cầu xử lý tối thiểu sẽ hoạt động hiệu quả hơn. Do vậy, với cấu trúc tầng mà các chức năng mạng phân chia giữa các phần cứng đã được thiết kế riêng cho từng chức năng sẽ làm tăng tuổi thọ của mạng.

- Về độ tin cậy: mỗi mạng cảm biến phải phù hợp với số lượng các node yêu cầu thỏa mãn điều kiện về thời gian sống và băng thông. Với mạng cấu trúc phẳng, khi kích cỡ mạng tăng lên thì thông lượng của mỗi node sẽ giảm về 0. Việc nghiên cứu các mạng cấu trúc tầng đem lại nhiều triển vọng để khắc phục vấn đề này. Một cách tiếp cận là dùng một kênh đơn lẻ trong cấu trúc phân cấp, trong đó các node ở cấp thấp hơn tạo thành một cụm xung quanh trạm gốc. Trong trường hợp này, dung lượng của mỗi lớp trong cấu trúc tầng và dung lượng của mỗi cụm trong mỗi lớp xác định là độc lập với nhau.

3.1.2 Mô hình giao thức của WSN

Trên cơ sở kiến trúc phân tầng trong mô hình tham chiếu OSI, ta có thể phân loại giao thức theo từng tầng của nó, thường có 3 loại giao thức thường được nghiên cứu và sử dụng trong WSN:

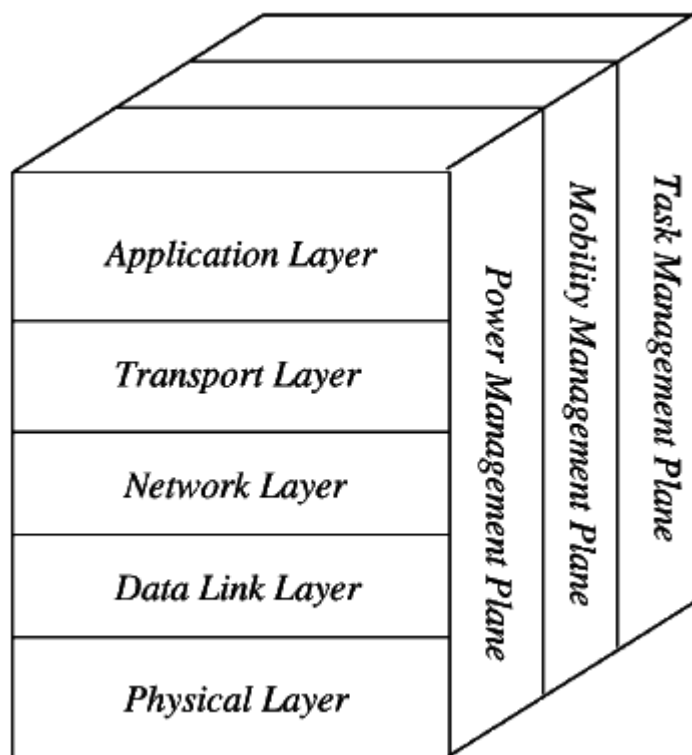
- Giao thức tầng ứng dụng: Tùy theo nhiệm vụ cảm biến, các loại phần mềm ứng dụng khác nhau có thể được xây dựng và sử dụng ở lớp ứng dụng. Hiện nay có ba giao thức lớp ứng dụng quan trọng là giao thức quản lý cảm biến SMP (Sensor Management Protocol), giao thức phân nhiệm vụ và quảng cáo số liệu TADAP (Task Assignment and Data Advertisement Protocol), giao thức truy vấn cảm biến và phổ biến số liệu SQDDP (Sensor Query and Data Dissemination Protocol). Tất cả các giao thức lớp ứng dụng này đều là những vấn đề nghiên cứu có tính mở.

- Giao thức tầng giao vận: TCP và UDP. kết nối TCP được sử dụng để liên lạc giữa nút quản lý của người sử dụng và nút thu nhận (Sink) và một giao thức lớp giao vận phù hợp với môi trường mạng cảm biến được sử dụng cho truyền thông giữa nút thu nhận và các nút cảm biến. Kết quả là truyền thông giữa nút người sử dụng và nút thu nhận có thể sử dụng giao UDP hoặc TCP thông qua Internet hoặc qua vệ tinh. Mặt khác, việc truyền thông giữa nút thu nhận và các nút cảm biến chỉ sử dụng hoàn toàn các giao thức kiểu như UDP, bởi vì các nút cảm biến có bộ nhớ hạn chế.

- Giao thức tầng mạng: Bao gồm các giao thức định tuyến như SPIN, Rumor, LEACH, PEGASIS,... Có thể chia các giao thức định tuyến ra làm 03 nhóm [9]: Data-Centric, Hierarchical, Location-based. Các giao thức định tuyến ở tầng mạng trong WSN đều được thiết kế khác với các giao thức định tuyến truyền thống. Với mục tiêu tối ưu hiệu quả hoạt động của mạng, tiết kiệm tối đa năng lượng, không cản trở sự cộng tác của các nút mạng.

- Giao thức tầng MAC: MAC 802.11, MAC 802.15.4, SMAC, BMAC, TMAC, ZMAC... Có thể phân loại giao thức tầng MAC thành 3 loại đó là: Contention-based MAC, Scheduled-based MAC và Two-Radio-Based MAC. Tầng này có nhiệm vụ ghép các luồng dữ liệu, phát hiện các khung dữ liệu, điều khiển truy nhập đường truyền và sửa lỗi. Nó đảm bảo sự tin cậy cho kết nối điểm - điểm (Point to Point) và điểm - đa điểm (Point to Multi-point) trong mạng truyền thông.

Mô hình giao thức được áp dụng cho mạng cảm biến được trình bày trong Hình 3-3 Mô hình giao thức của WSN. Kiến trúc này bao gồm các tầng và các khối quản lý. Các khối quản lý này làm cho các nút mạng có thể làm việc cùng nhau theo cách có hiệu quả nhất, định tuyến dữ liệu trong mạng và chia sẻ tài nguyên giữa các nút cảm biến.



Hình 3-3 Mô hình giao thức của WSN

Khối quản lý năng lượng: Quản lý cách cảm biến sử dụng nguồn năng lượng của nó. Nút cảm biến có thể tắt bộ phận nhận sau khi nhận một bản tin từ một trong các nút lân cận. Ngoài ra, khi mức năng lượng của nút cảm biến thấp, nút cảm biến sẽ thông báo tới tất cả các nút lân cận rằng mức năng lượng của nó đã thấp nên nó không thể tham gia vào việc định tuyến và chuyển tiếp cho các bản tin. Năng lượng còn lại được dự trữ cho việc cảm biến và gửi thông điệp.

Khối quản lý di động: Có nhiệm vụ phát hiện và đăng ký sự chuyển động của các node. Các node duy trì việc theo dõi xem ai là node lân cận của chúng. Với việc nhận biết được các nút cảm biến lân cận, nút cảm biến có thể cân bằng giữa nhiệm vụ và năng lượng sử dụng.

Khối quản lý nhiệm vụ: Cân bằng và sắp xếp nhiệm vụ cảm biến giữa các node trong một vùng quan tâm. Không phải tất cả các node cảm biến đều thực hiện nhiệm vụ cảm nhận ở cùng một thời điểm. Một vài nút cảm biến thực hiện nhiệm vụ nhiều hơn các nút khác tùy theo mức năng lượng của chúng. Những khối quản lý này rất cần thiết, như vậy, các nút cảm biến có thể làm việc cùng với nhau để có hiệu quả về năng lượng, có thể định tuyến số liệu trong một mạng cảm biến di động và chia sẻ tài nguyên giữa các nút cảm biến. Nếu không, mỗi nút cảm biến sẽ chỉ làm việc một cách đơn lẻ.

3.2 Giao thức MAC trong các chuẩn truyền thông không dây

3.2.1 Giao thức MAC trong chuẩn 802.11

Chuẩn IEEE 802.11 là một họ các chuẩn dành cho truyền thông không dây với các định nghĩa ở lớp Physical và lớp DataLink. Chuẩn này chủ yếu được áp dụng trong mạng LAN không dây hay còn gọi là WLAN. WLAN là một loại mạng cục bộ sử dụng kết nối không dây giữa các thành phần, môi trường truyền thông của các thành phần trong mạng là không khí. Mạng LAN nói chung sử dụng cơ chế điều khiển truy cập kênh CSMA với các công đoạn: Lắng nghe đường truyền, nếu rỗi sẽ gửi dữ liệu, nếu bận sẽ hoãn gửi dữ liệu theo 3 cách non-persistent CSMA, persistent csma và P-persistent CSMA. Tuy nhiên với giao thức CSMA thông thường nếu có nhiều hơn một nút muốn gửi dữ liệu và thấy đường truyền rỗi thì đụng độ sẽ xảy ra, các gói tin đụng độ sẽ bị hủy và nút gửi phải gửi lại.

Để khắc phục hiện tượng xung đột tín hiệu trên đường truyền trong mạng thông thường thì CSMA được cải tiến thành 2 phiên bản: CSMA/CD và CSMA/CA.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

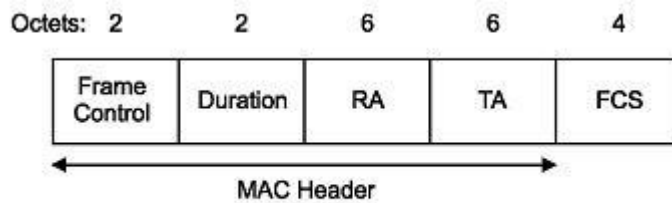
Đa truy cập nhận biết sóng mang có phát hiện xung đột. Phương pháp này thực hiện những cải tiến sau: 1) Giao thức CSMA/CD sẽ ngừng ngay việc gửi dữ liệu khi phát hiện có xung đột. 2) Thực thể giao thức MAC sẽ gửi quảng bá tín hiệu báo xung đột (jam signal) để các nút khác trong mạng biết và không gửi dữ liệu vào thời điểm đó. 3) Các nút gửi dữ liệu quảng bá sẽ khởi tạo một bộ đếm lùi thời gian ngẫu nhiên, khi bộ đếm này hết chúng sẽ thực hiện gửi lại dữ liệu bị xung đột.

Tuy nhiên cơ chế này không thể áp dụng được cho các mạng WLAN. Nguyên nhân là do để cài đặt được các kỹ thuật phát hiện xung đột đòi hỏi phải sử dụng các thiết bị có khả năng truyền radio song công, điều này có thể làm tăng đáng kể chi phí thiết bị mạng. Một nguyên nhân khác là do trong môi trường không dây, chúng ta không thể đảm bảo rằng tất cả các trạm đều nghe được toàn bộ môi trường truyền một cách đúng đắn. Tức là việc cảm nhận môi trường truyền đang rỗi có thể không chính xác. Hiện tượng này sẽ được trình bày trong phần tiếp theo.

CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance):

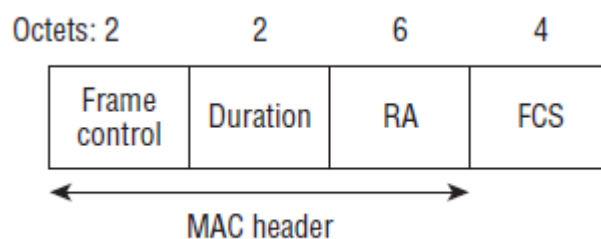
Đa truy cập nhận biết sóng mang và tránh xung đột, đây là giao thức được sử dụng trong lớp MAC của chuẩn 802.11. Giao thức này bao gồm các chức năng chính sau:

- Chức năng cảm nhận sóng mang: Gồm hai cơ chế: Cảm nhận sóng mang vật lý và cảm nhận sóng mang ảo. Mục đích chính của việc cảm nhận sóng mang là xác định xem trên đường truyền có sóng mang hay không từ đó xác định được đường truyền là rỗi hay bận.
- Chức năng tránh xung đột: Khi một nút muốn gửi gói tin, nó sẽ gửi quảng bá gói tin yêu cầu gửi RTS, gói tin này có gắn một trường chỉ rõ khoảng thời gian cần để truyền gói tin đó. Hình 3-4 Khuôn dạng gói tin RTS



Hình 3-4 Khuôn dạng gói tin RTS

Trên Hình 3-4 Khuôn dạng gói tin RTS, trường Duration sẽ giúp cho các nút lân cận biết được đường truyền sẽ bận trong bao lâu, sau khi nhận được gói tin này các nút sẽ ghi giá trị này trong một biến vector thời gian NAV. Giá trị NAV này sẽ giúp các nút khác xác định trạng thái đường truyền đang là bận do đó sẽ không tham gia việc truy cập môi trường truyền. Sau khi nút nhận gửi gói tin CTS thì nút gửi mới tiến hành gửi dữ liệu.

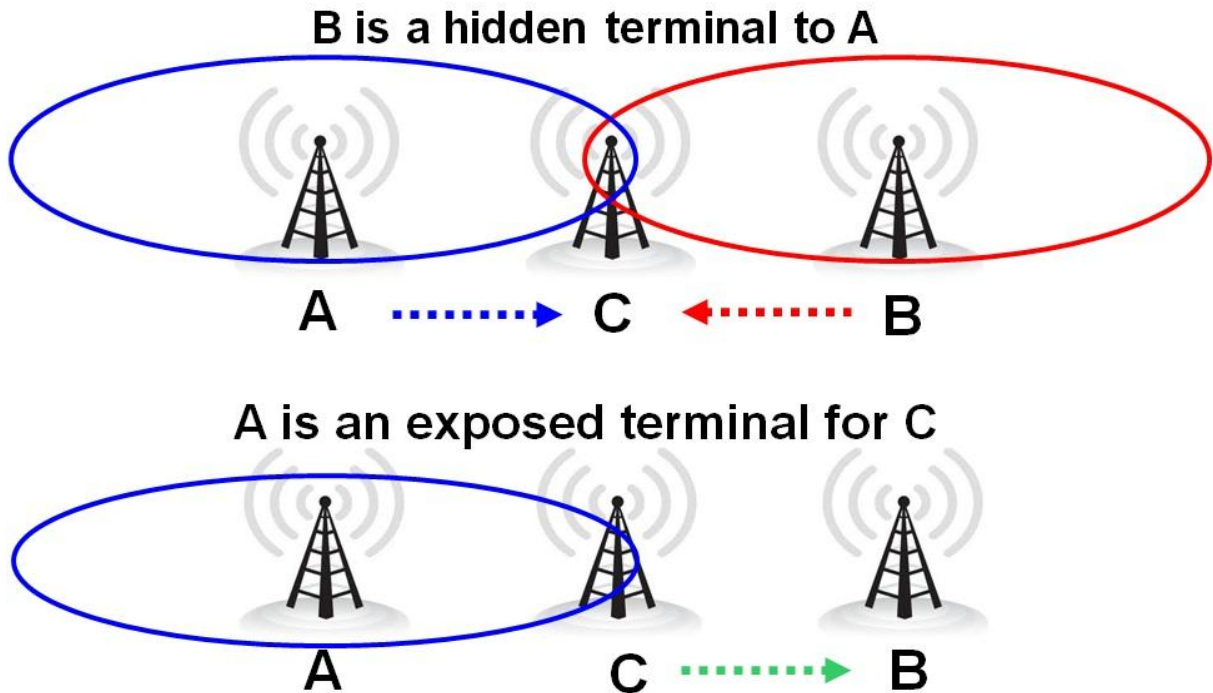


Hình 3-5 Khuôn dạng gói tin CTS

Như vậy với việc sử dụng cặp gói tin RTS/CTS thì giao thức CSMA/CA đã giải quyết được hai vấn đề chính trong mạng không dây đó là:

- Vấn đề Hidden Terminal Hình 3-6 Vấn đề hidden terminal và exposed terminal: Hai nút A B không nằm trong vùng phủ sóng của nhau, nút C nằm trong vùng phủ sóng của cả A và B, do đó khi A và B muốn gửi dữ liệu 2 nút này sẽ không “nhìn” thấy nút còn lại và cùng gửi dữ liệu cho C dẫn đến tình trạng xung đột. Cơ chế RTS/CTS sẽ giải quyết việc này khi C nhận được RTS của A hoặc B. C sẽ gửi quảng

bá CTS đến tất cả các nút trong phạm vi của nó. Do vậy các nút khác sẽ xác định được trạng thái đường truyền bận và không tiếp tục gửi gói tin. Nút gửi RTS sẽ nhận được CTS và bắt đầu truyền tin.

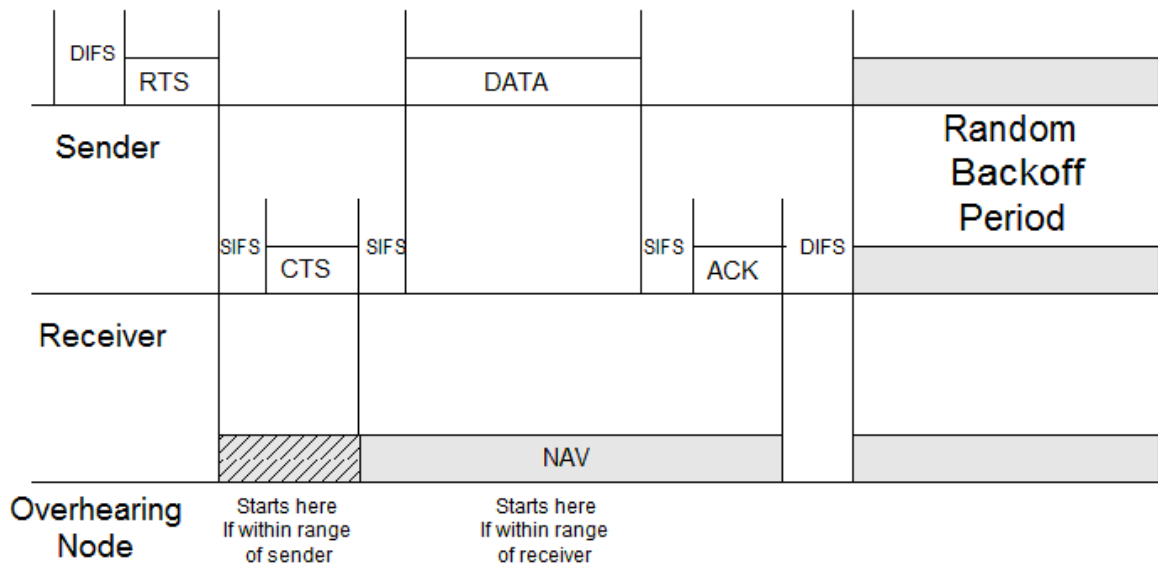


Hình 3-6 Vấn đề hidden terminal và exposed terminal

- Vấn đề Exposed Terminal Hình 3-6 Vấn đề hidden terminal và exposed terminal: Nút C đang truyền dữ liệu cho nút B. Khi đó nếu A muốn truyền dữ liệu cho nút khác nằm ngoài vùng phủ sóng của C và B, nhưng do nó nằm trong phạm vi của C nên nó nghĩ đường truyền đang bận và không thực hiện việc truyền dữ liệu. Cơ chế RTS/CTS sẽ giải quyết vấn đề này bằng cách khi C quảng bá RTS đến A và B. B sẽ trả lời CTS, nhưng vì A và B không nằm trong phạm vi của nhau nên A sẽ không nhận được CTS của B. Do vậy A sẽ nghĩ đường truyền rỗi và thực hiện truyền dữ liệu đến nút khác bình thường.

Mô hình cộng tác phân tán – DCF [10] [3]: Đây là một trong 2 mô hình điều khiển truy cập trong chuẩn IEEE 802.11. DCF sử dụng hoàn toàn cơ chế truy cập đường truyền CSMA/CA. Hoạt động với việc không có một thiết bị trung tâm điều khiển truyền thông. Đối với cơ chế này, khi một nút mạng muốn truyền dữ liệu, nó sẽ phải đợi một khoảng thời gian xác định DIFS sau khi đường truyền rỗi. Sau khi hết DIFS các trạm có thể truyền dữ liệu. Để tránh tình huống xung đột, DCF sử dụng một thuật toán Random Back-Off sẽ chọn ngẫu nhiên một giá trị từ 0 đến giá trị Contention Window (CW). Theo mặc định, giá trị CW có thể khác nhau tùy nhà sản xuất và nó được lưu trữ trong NIC (Network Interface Card) của máy trạm. Để có được giá trị Random Back-Off, ta lấy một số ngẫu nhiên đã chọn ở trên nhân với Slot Time (Random Back-Off chính là số lần Slot Time). Giá trị của Random Back-Off là

khoảng thời gian các trạm phải chờ thêm sau khi đường truyền rồi và DIFS đã trôi qua, sau đó trạm nào có Random Back-Off ngắn nhất sẽ giành được quyền truyền frame.



Hình 3-7 Cơ chế DCF trong chuẩn IEEE 802.11

Mô hình cộng tác điểm – PCF [10] [3]: Đây cũng là một cơ chế điều khiển truy cập trong chuẩn 802.11. PCF được phát triển dựa trên DCF, cơ chế chủ đạo của PCF là việc cho phép các nút được truyền tin mà không phải thông qua quá trình tranh chấp như DCF bằng cách thực hiện quá trình bầu cử (polling). Việc bầu cử trong cơ chế này được thực hiện bởi AP, do vậy những mạng ad-hoc sẽ không hoạt động được với mô hình này. Ưu điểm của mô hình này là đảm bảo độ trễ được xác định trước do đó phù hợp với các ứng dụng đòi hỏi QoS như truyền âm thanh, hình ảnh.

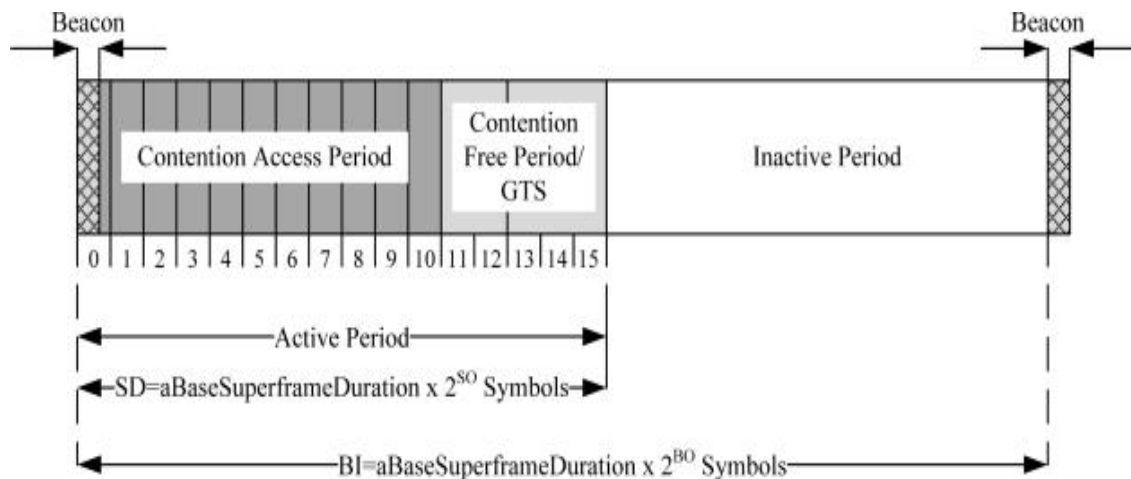
3.2.2 Giao thức MAC trong chuẩn 802.15.4 [5] [4]

IEEE 802.15.4 là tiêu chuẩn được định nghĩa bởi tổ chức IEEE dành cho các đặc tính ở tầng Physical và tầng MAC phục vụ truyền thông giữa những thiết bị sử dụng ít năng lượng, giá thành rẻ và tốc độ truyền dữ liệu thấp. Chuẩn 802.15.4 được kế thừa và phát triển bởi ZigBee Alliance (Một hiệp hội các công ty viễn thông) thành công nghệ ZigBee. Sử dụng chuẩn 802.15.4 cho các giao thức ở các tầng thấp, và thiết kế thêm các ứng dụng, giao thức ở tầng cao hơn.

Lớp MAC trong chuẩn 802.15.4 thực hiện các nhiệm vụ như: Xử lý truy cập vào kênh, sinh ra các mốc báo hiệu (Beacon) nếu một thiết bị trở thành điều phối, đồng bộ hóa các Beacons, thực hiện cơ chế CSMA/CA, cơ chế GTS, cung cấp liên kết giữa các thực thể.

Cấu trúc siêu khung: Tiêu chuẩn IEEE 802.15.4 định nghĩa một cấu trúc siêu khung theo nhu cầu. Định dạng của siêu khung do PAN coordinator qui định. Siêu khung bắt đầu bằng beacon (mốc báo hiệu) và được chia thành 16 khe như nhau (Hình 3-8 Cấu trúc siêu khung của chuẩn 802.15.4. Khe thời gian đầu tiên của siêu khung

dùng để phát beacon. Mục đích chính của beacon là để đồng bộ các thiết bị tham gia vào mạng, xác nhận PAN, mô tả cấu trúc siêu khung. Các khe còn lại được dùng bởi các thiết bị tranh chấp để trao đổi thông tin trong thời gian tranh chấp truy cập (Contention Access Period - CAP). Các thiết bị dùng giao thức CSMA/CA phân khe thời gian để tranh chấp truy cập với các thiết bị khác. Việc thông tin giữa các thiết bị phải hoàn tất trước khi kết thúc CAP của siêu khung hiện tại và bắt đầu phát mốc Beacon của siêu khung kế tiếp. Để đáp ứng yêu cầu về độ trễ và băng thông của ứng dụng, PAN coordinator đưa ra nhóm khe tích cực cho các ứng dụng. Các khe này được gọi là khe thời gian đảm bảo (Guaranteed Time Slots - GTSs). Số GTS không vượt quá 7. Tuy nhiên một khe GTS có thể kéo dài hơn một khe thời gian trong siêu khung trong vùng không có tranh chấp - CFP (Contention Free Period). Như trên Hình 3-8 Cấu trúc siêu khung của chuẩn 802.15.4 CFP luôn xuất hiện ngay sau CAP và trước khi kết thúc siêu khung. Các khe CAP dành cho tranh chấp truy cập giữa các thiết bị mạng và các thiết bị mới muốn tham gia vào mạng. Tất cả việc thông tin dùng tranh chấp truy cập hay tranh chấp tự do phải hoàn tất trước khi kết thúc khoảng thời gian tương ứng CAP và CFP trong siêu khung.



Hình 3-8 Cấu trúc siêu khung của chuẩn 802.15.4

Các thiết bị mạng cần phân chia GTS, sẽ gọi yêu cầu trong khoảng thời gian CAP để có một số khe thời gian mong muốn. Các khe được yêu cầu có thể là dùng cho mục đích thu hoặc phát. Các khe thu được dùng để các thiết bị lấy dữ liệu về từ bộ coordinator, còn khe phát dùng để gửi dữ liệu lên cho coordinator. Các thiết bị không có dữ liệu để trao đổi với PAN coordinator có thể chuyển sang chế độ ngủ để tiết kiệm năng lượng. Mặt khác, để giảm công suất tiêu thụ, bộ coordinator có thể chia siêu khung thành khoảng thời gian tích cực và thời gian nghỉ như trên Hình 3-8 Cấu trúc siêu khung của chuẩn 802.15.4. Khoảng thời gian tích cực gồm 16 khe thời gian chứa Beacon, CAP và CFP. Khoảng thời gian nghỉ tất cả các thiết bị gồm cả PAN coordinator có thể đi vào chế độ ngủ và đặt thời gian timer wake-up để bắt đầu siêu khung mới. Tùy theo yêu cầu ứng dụng, chiều dài khoảng tích cực và khoảng nghỉ, chiều dài khe thời gian và số GTSs được chọn phù hợp hoạt động của mạng.

Các chế độ hoạt động trong 802.15.4: Chuẩn IEEE 802.15.4 định nghĩa 3 dạng dữ liệu: periodic data (dữ liệu theo chu kỳ), intermittent data (dữ liệu gián đoạn), và repetitive low-latency data (dữ liệu lặp lại có độ trễ thấp). Periodic data trong các ứng dụng cảm biến không dây, các cảm biến thay đổi lần lượt giữa hai chế độ tích cực hoặc nghỉ. Intermittent data được tạo ra bởi một kích thích bên ngoài. Repetitive low-latency data dùng trong các ứng dụng quan trọng như hệ thống giám sát an ninh. Dạng này đòi hỏi sự phân chia khe thời gian để đảm bảo truy cập kênh truyền với độ trễ cho phép. Để thích ứng với 3 dạng dữ liệu này, IEEE 802.15.4 phân ra hai chế độ hoạt động : Beacon (có mốc báo hiệu) và NonBeacon (không có mốc báo hiệu).

- **Chế độ Beacon:** Chế độ này cho phép mở rộng mạng trong các cấu hình mesh hay cluster tree để đồng bộ hoạt động và phối hợp bộ điều phối với các thiết bị khác. để điều tiết truy cập kênh truyền, bộ điều phối dùng cấu trúc siêu khung như đã đề cập ở mục trước. Siêu khung được chia thành 16 khe như nhau. Khe đầu tiên dành để phát beacon. Thiết bị mạng có thể tranh chấp truy cập trong khoảng CAP với cơ chế CSMA/CA. Đối với các ứng dụng đòi hỏi độ trễ thấp có thể phải yêu cầu PAN coordinator cấp khe GTS trong khoảng tranh chấp tự do CFP. Sự phân chia GTSS được thực hiện trước khi bắt đầu CAP, các hoạt động dựa trên GTS diễn ra trong các khe thời được phân và hoàn tất trước khi kết thúc CFP.

- **Chế độ NonBeacon:** Trong chế độ không có beacon, bộ điều phối mạng không gởi khung beacon cho các thiết bị khác. Hơn nữa, chế độ này không cung cấp các khe thời gian đảm bảo cho các ứng dụng yêu cầu độ trễ thấp. Thay vào đó các thiết bị phải tranh chấp truy cập kênh truyền dùng giao thức CSMA/CA không phân khe thời gian.

3.3 Giao thức MAC trong WSN

Các chuẩn IEEE 802 trong mạng LAN định nghĩa lớp điều khiển truy cập MAC là một lớp con của tầng liên kết dữ liệu trong mô hình tham chiếu OSI. Các chức năng chính của lớp MAC là thực hiện tiếp nhận, kiểm tra lỗi và chuyển tiếp dữ liệu từ các lớp trên. Thực hiện các cơ chế đánh địa chỉ và điều khiển truy cập kênh. Lớp MAC trong WSNs phải đảm bảo được các yêu cầu về tiết kiệm năng lượng cho các nút mạng, ngoài ra nó phải thích ứng được với các thay đổi của mô hình mạng và nhiễu trên các kênh truyền không dây.

Nhu cầu duy trì năng lượng là vấn đề quan trọng nhất trong thiết kế các giao thức MAC cho WSNs. Có nhiều yếu tố có thể gây lãng phí năng lượng, ví dụ như việc phải liên tục lắng nghe đựng độ từ các trạm. Việc điều khiển truy cập đòi hỏi trao đổi thông tin đồng bộ giữa các node tranh chấp. Sự trao đổi thông tin này cũng tăng công suất tiêu thụ. Việc liên tục xảy ra đựng độ cũng như phát lại các gói bị đựng độ cũng tiêu tốn năng lượng của node. Tương tự, việc các trạm nghe các gói tin có đích đến không phải là trạm đó, hiện tượng nghe thừa cũng gây lãng phí các tài nguyên không cần thiết. Chính vì lý do đó các giao thức MAC được thiết kế tập trung chính vào việc giảm năng lượng hao phí do đựng độ, lắng nghe thời gian dài, nghe thừa. Các giao

thức MAC được chia ra làm 02 nhóm [10]: - Giao thức MAC dựa trên kế hoạch (Schedule – based) và giao thức MAC dựa trên tranh chấp (Contention – based). Trong phần này tôi sẽ trình bày về các giao thức MAC được sử dụng trong các chuẩn mạng không dây 802.11 và 802.15.4, cùng với các giao thức MAC được thiết kế dành cho WSNs.

3.3.1 Contention-based MAC [10]

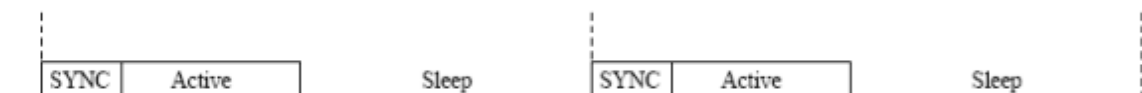
a, Sensor MAC [11]

Là giao thức được cải tiến từ giao thức MAC truyền thông của chuẩn 802.11. Ý tưởng của S-MAC là giúp tiết kiệm năng lượng các nút mạng bằng cách chia khung thời gian thành 2 phần: Phần hoạt động và Phần ngủ.

- Phần hoạt động (Active): Trong thời gian hoạt động thì thiết bị thu phát sóng ra-đio sẽ được bật và cung cấp năng lượng phục vụ truyền tin.
- Phần ngủ (Sleep): Nút cảm biến tắt thiết bị thu phát để bảo tồn năng lượng.

S-MAC (Sensor MAC) được giới thiệu bởi các tác giả: Wei Ye, John Heidemann, Deborah Estrin tại Hội nghị INFOCOM lần thứ 21, năm 2002 [11]. Được xây dựng trên nền tảng của các giao thức kiểu cạnh tranh như 802.11, Sensor Mac đã kế thừa sự linh hoạt, tính khả biến của giao thức MAC trên nền cạnh tranh, trong khi cải tiến tính hiệu quả sử dụng năng lượng trong mạng đa chặng (Multi-hop). S-MAC cũng đã đạt mục tiêu giảm bớt tiêu thụ năng lượng do các nguyên nhân gây tiêu hao năng lượng, như: nghe khi rỗi (idle listening), xung đột (collision), nghe thừa (overhearing) và xử lý thông tin điều khiển (overhead). Giao thức S-MAC đã được thiết kế gồm có ba vấn đề chính: chu kỳ thức - ngủ; tránh xung đột và nghe thừa; xử lý thông điệp.

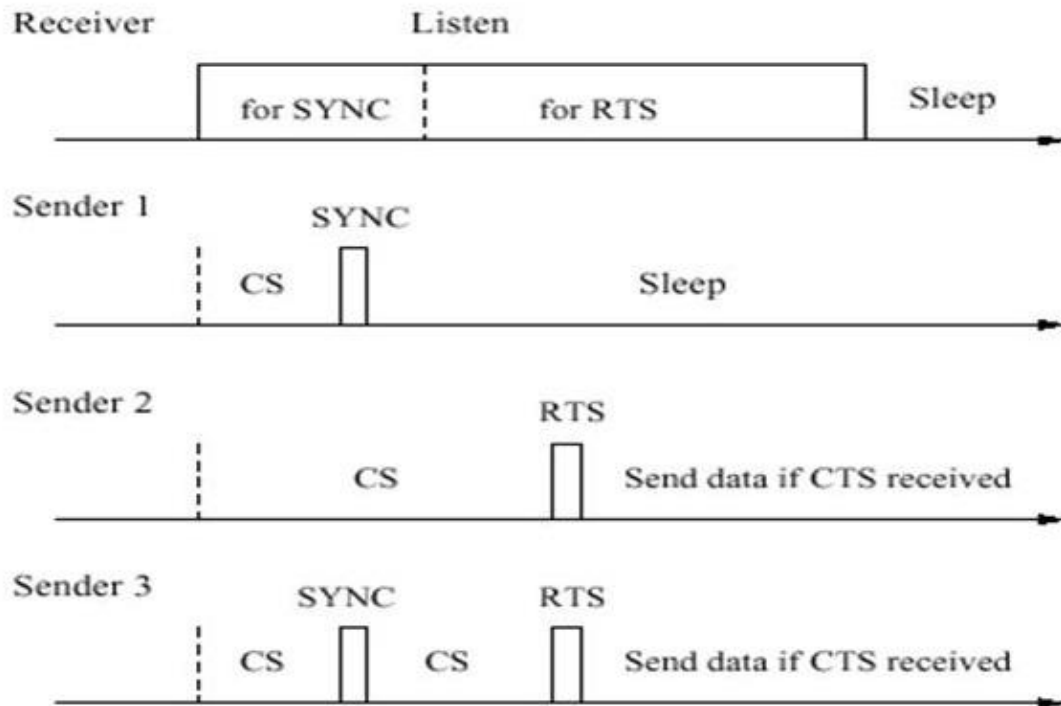
- Chu kỳ thức – ngủ: Một trong các mục tiêu khi thiết kế S-MAC là làm giảm năng lượng tiêu thụ do lắng nghe. Do vậy cải tiến quan trọng nhất của S-MAC là việc các nút mạng hoạt động với chu kỳ thức – ngủ, với độ dài của một chu kỳ phụ thuộc vào các ứng dụng cụ thể của mạng cảm biến không dây mà người thiết kế có thể thiết lập.



Hình 3-9 Chu kỳ thức-ngủ của Sensor MAC

- Đồng bộ thời gian giữa các nút trong mạng: Việc đồng bộ thời gian giữa các nút cảm biến trong mạng là vấn đề cần thiết để xác định thời gian thức, ngủ của mỗi nút. Việc cập nhật thời gian biểu được thực hiện bằng trao đổi gói tin đồng bộ SYNC (Hình 3-9 Chu kỳ thức-ngủ của Sensor MAC). Gói tin SYNC rất ngắn và bao

gồm địa chỉ của nút gửi và thời điểm chuyển sang trạng thái ngủ tiếp theo của nó. Thời điểm ngủ tiếp theo liên quan đến thời điểm mà nơi gửi kết thúc truyền gói tin đồng bộ SYNC, cũng xấp xỉ khi nút nhận nhận được gói tin (khi độ trễ ngắn). Những nút nhận sẽ điều chỉnh đồng hồ của chúng ngay sau khi nhận được gói tin đồng bộ. Nút cảm biến sẽ chuyển sang trạng thái ngủ khi đồng hồ của nó kết thúc tính giờ, báo đến thời điểm ngủ.



Hình 3-10 Quá trình đồng bộ của Sensor MAC

Để một nút nhận được cả những gói đồng bộ lẫn những gói dữ liệu, chúng được chia khoảng thời gian thức (active time) thành hai phần. Phần đầu tiên để nhận những gói tin đồng bộ, phần thứ hai để trao đổi dữ liệu (Hình 3-10 Quá trình đồng bộ của Sensor MAC). Mỗi phần được chia tiếp thành nhiều khe thời gian cho những nút gửi để thực hiện cảm nhận sóng mang (Carrier Sense). Ví dụ, nếu một nút gửi muốn gửi một gói tin đồng bộ thì nó khởi động cảm nhận sóng mang khi nút nhận bắt đầu nghe. Nó lựa chọn ngẫu nhiên một khe thời gian để kết thúc cảm nhận sóng mang. Nếu nó không phát hiện ra bất kỳ sự truyền nào vào khoảng cuối khe, thì nó chiếm đường truyền và bắt đầu gửi gói tin đồng bộ của nó ở tại thời điểm ấy. Việc thực hiện truyền gói dữ liệu cũng được thực hiện tương tự, nút muốn gửi dữ liệu chúng sẽ sử dụng gói tin RTS sau đó nút nào nhận được gói tin CTS sớm nhất sẽ được quyền gửi dữ liệu.

- Vấn đề tránh xung đột: Do kế thừa giao thức CSMA/CA của bộ chuẩn 802.11 nên S-MAC có đầy đủ các tính năng tránh xung đột gồm có cảm nhận sóng mang vật lý, cảm nhận sóng mang ảo, có sử dụng các gói tin yêu cầu truy nhập đường truyền và gói tin sẵn sàng nhận RTS/CTS, như đã trình bày trong mục 3.2.1; đồng thời

do phát triển việc tránh xung đột trên nền cạnh tranh theo chuẩn 802.11 nên S-MAC cũng giải quyết được vấn đề nút ẩn “Hidden Terminal” và nút hiện “Exposed Terminal” như giao thức CSMA/CA. Tuy nhiên để đảm bảo rằng việc tránh xung đột hiệu quả hơn thì S-MAC kết hợp việc cảm nhận sóng mang vật lý (do lớp vật lý tiến hành) với cảm nhận sóng mang ảo, có nghĩa là nếu một nút xác định không có cảm biến sóng mang vật lý đồng thời giá trị NAV của nó bằng không thì mới xác định đường truyền là rỗi.

- Vấn đề nghe thừa: Vấn đề này diễn ra khi có nhiều nút ở chu kỳ thức (Active) nhưng lại có một vài nút gửi dữ liệu, với đặc điểm lan truyền của sóng radio, do đó các nút nằm trong vùng phủ sóng của nút tham gia truyền thông đều có thể nghe thấy các thông tin không phải gửi cho nó. Sensor-MAC giải quyết vấn đề bằng cách cho phép các nút chuyển sang trạng thái ngủ ngay khi chúng nghe thấy các gói RTS hay CTS giữa các nút khác. Trong RTS và CTS có trường NAV chưa thời gian quy định tối đa cho truyền dữ liệu, các nút không tham gia quá trình truyền thông sẽ chuyển sang chế độ ngủ với thời gian bằng với giá trị của trường NAV. Hoạt động này sẽ giúp tiết kiệm đáng kể năng lượng.

- Vấn đề xử lý thông điệp: Để tối ưu hiệu quả xử lý dữ liệu, S-MAC chia các gói tin dài thành các gói nhỏ hơn sao cho phù hợp với thông lượng của đường truyền. Sau đó sử dụng kiểu phát Burst đối với toàn bộ các gói. Việc sử dụng kiểu phát Burst này chỉ cần sử dụng một cặp RTS/CTS, ngoài ra bên nhận cần gửi lại ACK để báo hiệu thành công và giữ kết nối. Đồng thời cũng tạo ra tín hiệu cho các gói lân cận biết việc truyền thông chưa hoàn thành và tiếp tục giữ trạng thái ngủ.

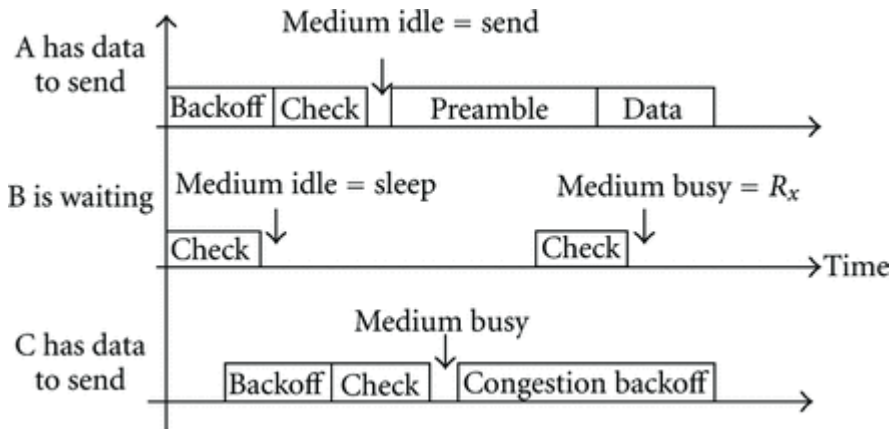
S-MAC có một số ưu điểm trong việc giảm tiêu thụ năng lượng, giao thức dễ dàng thích nghi với những thay đổi trong cấu trúc mạng, không cần một thực thể quản lý. Tuy nhiên nó vẫn còn tồn tại những nhược điểm về vấn đề đồng bộ hóa, hiện tượng nghe thừa trong một số trường hợp vẫn xảy ra. S-MAC đã được nghiên cứu rộng rãi và một số giao thức kế thừa đã được đề xuất. Ví dụ như T-MAC và DS-MAC. Trong phần tiếp theo của luận văn tôi sẽ trình bày giao thức B-MAC một hướng tiếp cận khác được nghiên cứu ứng dụng trong WSN.

b, Berkeley MAC [10]

B-MAC được thiết kế cho một mạng Ad-hoc với N nút gửi và 1 nút nhận. Ý tưởng cơ bản của B-MAC là đưa ra một cải biến cho các khung khởi đầu nhằm giảm thời gian chờ nghe (idle listening) từ đó tiết kiệm được năng lượng. Khi một gói tin muốn gửi dữ liệu, nó phải chờ một khoảng thời gian backoff rồi mới bắt đầu kiểm tra đường truyền. Nếu đường truyền rỗi nó sẽ bắt đầu truyền gói tin. Nếu đường truyền bận nó sẽ bắt đầu một khoảng thời gian backoff lần 2. Mỗi nút phải kiểm tra đường truyền một cách định kỳ sử dụng cơ chế Low Power Listening, nếu đường truyền đang bận và nút không có nhu cầu truyền dữ liệu thì nó sẽ chuyển sang trạng thái ngủ.

B-MAC điều chỉnh khoảng thời gian kiểm tra đường truyền sao cho bằng với kích thước của các khung khởi đầu. Ví dụ nếu đường truyền được kiểm tra mỗi 100ms

thì khung khởi đầu cũng phải có kích thước tối thiểu tương ứng với thời gian truyền là 100ms, để đảm bảo bên nhận phát hiện được gói tin. Các lớp trên của bộ giao thức có thể thay đổi khoảng thời gian của khung mở đầu này tùy vào ứng dụng, tuy nhiên phải đảm bảo được yêu cầu trên.



Hình 3-11 Giao thức B-MAC

Ưu điểm của việc sử dụng B-MAC là nó mặc định không sử dụng RTS, CTS hay ACK trong bất kỳ khung điều khiển nào, nhưng có thể được thêm vào khi cần. Hơn nữa B-MAC không yêu cầu đồng bộ hóa, hiệu năng của giao thức có thể được điều chỉnh bởi các lớp cao hơn để phù hợp với nhiều loại ứng dụng khác nhau. B-MAC đã được triển khai và kiểm thử trên các hệ thống phần cứng thật. Nhược điểm của B-MAC là nó tạo ra thông tin điều khiển quá lớn. Ví dụ cần sử dụng tới 271 bytes trong khung khởi đầu cho việc gửi 36 bytes dữ liệu.

3.3.2 Schedule-based MAC [10]

Là giao thức điều khiển truy cập dựa trên việc sắp đặt trước tài nguyên để tránh xung đột. Hầu hết các giao thức này đều dùng mô hình tương tự TDMA – giao thức đa truy cập phân chia theo thời gian. Trong đó các kênh truyền được chia ra thành các khe thời gian. Bao gồm một dãy N các khe nằm kế nhau tạo thành các khung thời gian. Các khung này lặp lại tuần hoàn. Trong mỗi khung các node được phân một số khe thời gian. Các khe này chính là lịch trình hoạt động của node. Lịch trình này có thể cố định, hoặc xây dựng theo nhu cầu. Các node hoạt động ở 2 trạng thái: hoạt động và nghỉ. Khi đến khung thời gian được định trước, các node chuyển sang trạng thái hoạt động và gửi dữ liệu. Sau đó sẽ chuyển về trạng thái nghỉ khi hết khung thời gian nhằm bảo toàn năng lượng. Có thể kể đến một số giao thức sử dụng kiểu thiết kế này như:

a, LEACH:

Low-Energy Adaptive Clustering Hierachy (LEACH) là một tập hợp các giao thức bao gồm các ứng dụng, các giao thức định tuyến, giao thức điều khiển truy cập và các đặc tính vật lý cho việc truyền thông trong WSNs. Giao thức điều khiển truy cập trong LEACH hoạt động theo vòng. Các node được tổ chức thành các cụm và bầu ra cụm trưởng (Cluster Head) sau đó gửi thông tin cho cụm trưởng. Mỗi cụm sử dụng

phương pháp trải phổ thành dãy trực tiếp - DSSS với các mã khác nhau để tránh xung đột. Việc truy cập đường truyền trong LEACH được chọn sao cho giảm được sự tiêu hao năng lượng cho các nút không phải là CH (Cluster-Head). Khi các nút chủ biết được tất cả các nút trong cụm của nó, nó sẽ gửi bản tin định thời TDMA để thông báo cho mỗi nút chính xác khi nào thì truyền dữ liệu đến nó. Điều này cho phép các nút có thể duy trì trong trạng thái ngủ đông (sleep state), chỉ khi đến thời điểm nó gửi dữ liệu thì nó mới thức dậy. Hơn nữa, dùng bản tin TDMA cho việc truyền dữ liệu còn giúp tránh được hiện tượng đụng độ (collision) xảy ra trong cụm. Hoạt động của LEACH được chia thành các vòng (round), mỗi vòng bắt đầu với pha thiết lập khi mà các cụm được hình thành, sau đó đến pha ổn định khi mà các khung dữ liệu được gửi tới các nút chủ và gửi tới base station. Tất cả các nút phải đồng bộ về mặt thời gian để bắt đầu pha thiết lập tại thời điểm giống nhau. Pha ổn định thường dài hơn rất nhiều so với pha thiết lập.

- Self – Configuring Cluster Formation

LEACH thực hiện phân nhóm (cụm) bằng việc sử dụng giải thuật phân tán, các nút tự quyết định mà không cần bất cứ sự điều khiển tập trung nào. Ưu điểm của phương pháp này là không yêu cầu việc giao tiếp với BS, do đó tránh được việc tiêu hao năng lượng nếu các nút ở xa BS. Đồng thời việc hình thành các cụm phân tán mà không cần biết chính xác vị trí của các nút trong mạng. Và nó không yêu cầu sự liên lạc toàn cục trong pha thiết lập cụm, và không có giả thiết nào về trạng thái hiện tại của các nút khác trong quá trình hình thành.

- Determining Cluster- Head Nodes

Khi các cụm được tạo ra, mỗi nút n tự động quyết định nó có là nút chủ cho vòng tiếp theo hay không. Quá trình chọn lựa diễn ra như sau: mỗi nút cảm biến chọn một số ngẫu nhiên giữa 0 và 1. Nếu con số này nhỏ hơn ngưỡng $T(n)$ thì nút đó trở thành nút chủ. $T(n)$ được xác định theo phương trình sau:

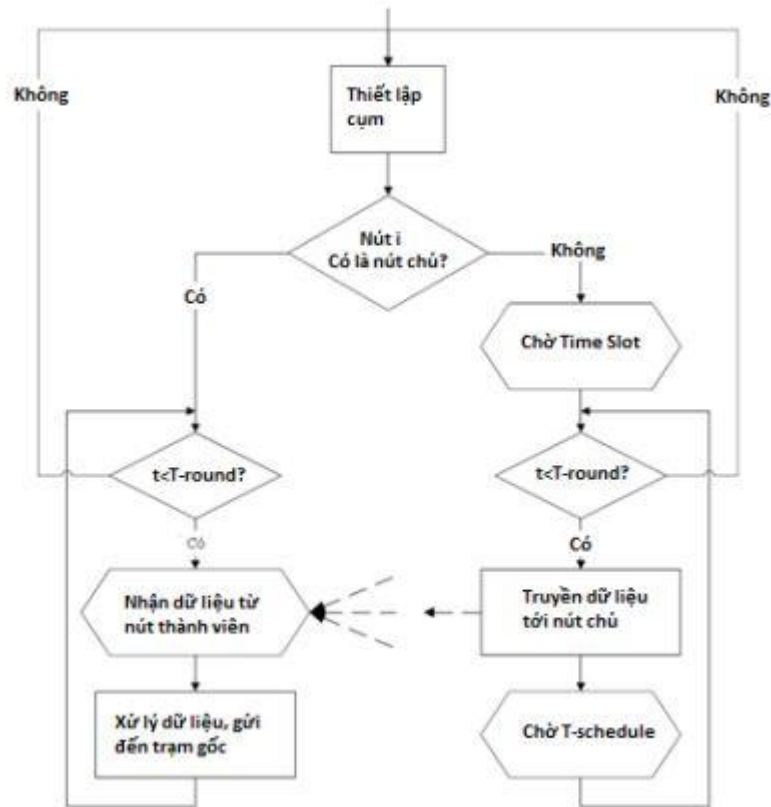
$$T(n) = \frac{P}{1 - P \times (r \bmod \frac{1}{P})} \quad (3-1)$$

Trong đó P là tỉ lệ cần thiết số nút làm chủ, r là số vòng hiện tại. Với những nút mà không được làm nút chủ trong $1/P$ vòng cuối thì ngược lại $T1(n) = 0$. Dùng thuật toán này thì mỗi nút sẽ là nút chủ đúng một lần trong vòng $1/P$ vòng. Chú ý: sau $1/P-1$ vòng, $T1(n) = 1$ với tất cả các nút chưa được làm nút chủ. Khi có một nút làm nút chủ, nó thông báo điều này tới tất cả các nút khác. Các nút không phải nút chủ dùng những bản tin này từ các nút chủ để chọn cụm mà chúng muốn tham gia dựa trên cường độ tín hiệu nhận được bản tin này. Sau khi các nút chủ đã được hình thành, nút chủ sau đó sẽ quyết định mô hình TDMA cho các nút tùy thuộc từng cụm, quảng bá mô hình và sau đó pha trạng thái tĩnh bắt đầu.

- Set-up Phase: Mỗi một nút sẽ tự nó chọn làm nút chủ cụm nếu xác suất của nó thỏa mãn phương trình (3-1) trên. Nút chủ phải thông báo cho các nút khác trong mạng biết rằng nó được chọn làm nút chủ ở vòng hiện tại. Để thực hiện điều này, mỗi nút chủ sẽ phát bản tin quảng bá (ADV – Advertise) dùng thuật toán CSMA. Bản

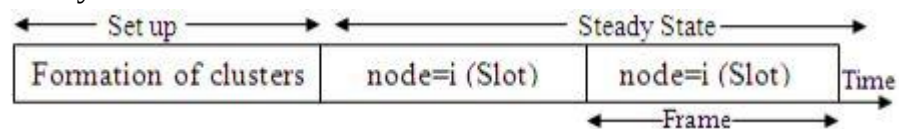
tin này là một bản tin nhỏ bao gồm ID của nút và header để phân biệt bản tin này là bản tin quảng bá. Tuy nhiên, bản tin này phải được phát quảng bá tới tất cả các nút trong mạng. Thứ nhất là để đảm bảo tất cả các nút lắng nghe bản tin quảng bá để tránh xảy ra đụng độ khi CSMA được dùng. Thứ hai là không có cơ chế để đảm bảo rằng các nút mà được chọn là nút chủ cụm sẽ được phân bố đều trên toàn mạng. Nếu công suất phát bản tin quảng bá bị giảm đi, một số nút ở biên có thể sẽ không nhận được thông báo và do đó có thể sẽ không còn ở trong vòng này. Bản tin quảng bá có kích thước rất nhỏ, do đó việc tăng công suất phát bản tin này để nó đến được tất cả các nút trong mạng không phải là một trở ngại lớn. Bởi vậy công suất phát sẽ được thiết lập ở mức cao vừa đủ để tất cả các nút trong mạng có thể lắng nghe được bản tin ADV này. Các nút chủ trong LEACH hoạt động như khối điều khiển trung tâm cục bộ để liên kết các dữ liệu trong cụm mà nó làm nút chủ. Nút chủ thiết lập bản tin định thời TDMA và truyền tới các nút trong cụm. Điều này đảm bảo sẽ không có đụng độ xảy ra và cho phép bộ phát sóng radio của các nút không phải nút chủ sẽ ở trạng thái ngủ (sleep state). Nó chỉ thức dậy tại thời điểm mà nó truyền dữ liệu. Như vậy sẽ tiết kiệm được năng lượng cho các nút. Sau khi bản tin TDMA được truyền đến tất cả các nút trong cụm, pha thiết lập đã hoàn thành và bắt đầu pha ổn định (steady state phase).

- **Steady- state Phase:** Hoạt động của pha ổn định được chia ra thành các frame. Mỗi nút sẽ gửi dữ liệu của nó tới Cluster-Head một lần trên một frame trong khe định vị của nó. Mỗi nút sẽ có một khe thời gian cố định, cứ đến khe thời gian đó thì nút truyền dữ liệu tới Cluster-Head. Số khe thời gian cho một khung dữ liệu phụ thuộc vào số lượng nút ở trong cụm, tức là có bao nhiêu nút trong cụm (trừ nút chủ) thì sẽ có bấy nhiêu khe thời gian. Trong khi giải thuật phân tán để xác định nút chủ, với số cụm trong mỗi vòng là k , nhưng nó lại không có cơ chế đảm bảo rằng sẽ có k cụm trong mỗi vòng. Thêm vào đó giao thức trong pha thiết lập không đảm bảo các nút sẽ phân bố đều cho mỗi nút chủ. Do đó, số nút trong một cụm là khác nhau và tổng dữ liệu mà mỗi nút gửi đến nút chủ phụ thuộc vào số nút trong cụm. Để giảm sự tiêu thụ năng lượng, mỗi nút không phải là nút chủ sẽ điều khiển công suất phát dựa trên cường độ của bản tin quảng bá nhận được từ nút chủ. Và kênh phát sóng của nút sẽ ở trạng thái ngủ cho đến khe thời gian phát sóng của nó. Các nút chủ sẽ phải giữ lại các dữ liệu mà các nút trong cụm gửi đến nó. Khi đã nhận được hết dữ liệu từ tất cả các nút, nó tiến hành xử lý dữ liệu cục bộ như nén, tổng hợp dữ liệu.... Dữ liệu đã được tổng hợp sau đó được gửi tới BS. Khoảng cách từ nút chủ tới BS có thể xa và kích cỡ bản tin dữ liệu thường là lớn, do đó mà năng lượng tiêu thụ do quá trình truyền này thường là cao.



Hình 3-12 Quá trình thiết lập cụm

Time-line trong một vòng của LEACH, từ khi các cụm được hình thành trong pha thiết lập, quá hoạt động của pha ổn định khi dữ liệu được truyền từ các nút tới Cluster-Head rồi truyền đến BS.



Hình 3-13 Vòng phân chia thời gian của LEACH

Để mô tả về việc trao đổi thông tin trong phạm vi một cluster. Giao thức MAC và giao thức định tuyến được thiết kế để đảm bảo cho các nút tiêu thụ năng lượng thấp và không xảy ra xung đột trong cụm. Tuy nhiên, kênh phát sóng không dây vốn là môi trường truyền kiểu quảng bá, do đó sự phát sóng của một cụm cũng sẽ ảnh hưởng đến các cụm gần nó.

Để giảm thiểu sự ảnh hưởng này, mỗi cluster trong LEACH sẽ trao đổi thông tin dùng cơ chế trải phổ thành dãy trực tiếp DSSS (directed-sequence spread spectrum) hay CDMA (Code Division Multiple Access). Mỗi một cụm sẽ dùng một mã trải phổ (spread code) duy nhất, tất cả các nút trong cụm phát dữ liệu của chúng tới nút chủ sẽ dùng mã trải phổ này và nút chủ sẽ lọc tất cả các nút có mã trải phổ này. Chú ý rằng mỗi nút chủ chỉ cần một mã trải phổ đơn để lọc cho tất cả các tín hiệu đến nó mà sử dụng mã trải phổ giống nhau. Điều này cũng hơi khác với cơ chế CDMA mà mỗi nút

sẽ có một mã trải phổ duy nhất. Dữ liệu từ các nút chủ được gửi tới BS sẽ dùng một mã trải phổ cố định, và cũng dùng cơ chế CSMA để tránh xảy ra đụng độ với các nút chủ khác. Tuy là kênh truyền vô tuyến, nhưng khi một nút chủ có dữ liệu để gửi tới BS, nó sẽ phải lắng nghe xem có nút chủ nào phát dữ liệu không. Nếu không có nút nào phát thì nó sẽ phát dữ liệu tới BS, còn nếu có nút đang phát dữ liệu thì nó sẽ đợi để phát dữ liệu.

b, Giao thức PEDAMACS .

Giao thức PEDAMACS (Power-Efficient and Delay-Aware Medium Access Protocol) giả định một AP có khả năng giao tiếp với tất cả các nút cảm biến trong một hop. Tuy nhiên các nút cảm biến có thể sử dụng nhiều hơn một hop để gửi dữ liệu đến AP. Có ba mức công suất truyền tải được dùng để xác định ba khoảng cách: Tối đa P_l , trung bình P_m , và tối thiểu P_s . Giao thức hoạt động với 4 pha.

Pha 1 Topology Learning: AP sẽ quảng bá một gói tin với công suất tối đa P_l , sau đó AP tiếp tục gửi một gói tin với công suất trung bình P_m , gói tin này sẽ được các node lan truyền đi trên toàn bộ mạng. Sử dụng mức độ công suất và các tín hiệu nhận được, các node có thể xác định các node hàng xóm (các node có khả năng nhận các gói tin truyền với công suất tối thiểu), các node gây nhiễu (các node không có khả năng nhận được các gói tin với công suất tối thiểu, nhưng nhận được các mức tín hiệu đủ cao để gây nhiễu với các tín hiệu khác), và các node cha (node luân chuyển gói tin đến AP). Quá trình này được thực hiện bằng một giao thức gần giống với 802.11 với RTS và CTS.

Pha 2 Topology Collection: Các node gửi thông tin topology của nó đến AP với mức công suất tối thiểu P_s , dữ liệu có thể đi qua nhiều chặng để đến được AP. Quá trình này sử dụng CSMA.

Pha 3 Scheduling : AP tính toán và quảng bá khung thời gian, các node sẽ tự điều chỉnh đồng hồ và biết được khoảng thời gian nào được phép để truyền và nhận tín hiệu. Ngoài khoảng thời gian đó các node ở trạng thái ngừng hoạt động.

Ưu điểm của PEDAMACS: sử dụng những đặc trưng của lớp vật lý và lớp mạng. PEDAMACS thích hợp cho các cảm biến định kỳ hoặc các cảm biến hướng sự kiện. Gán khe thời gian chỉ khi có sự kiện xảy ra, nếu không các node tiếp tục ở trạng thái ngừng hoạt động. Giao thức này có khả năng mở rộng để sử dụng nhiều hơn một AP xử lý các node ở ngoài phạm vi của AP chính.

Nhược điểm: Việc sử dụng một AP giao tiếp với tất cả các node đòi hỏi AP này phải có nguồn năng lượng duy trì liên tục. Điều này là không khả thi với WSNs. Ngoài ra việc sử dụng các mức công suất truyền tải thấp để tiết kiệm năng lượng sẽ ảnh hưởng đến phạm vi phủ sóng. Ví dụ cảm biến Mica2 motes chỉ có phạm vi 25cm với mức độ truyền tải tối thiểu – 20dBm.

c, Giao thức PRIMA

PRIMA (Priority-Based MAC Protocol for WSNs) sử dụng phương pháp tạo các Cluster giống với giao thức LEACH. Mỗi cluster đều có cluster head để điều khiển việc truyền thông và giữ đồng bộ giữa các nút bên trong cluster. Cluster Head được bầu lại mỗi 15 phút. PRIMA định nghĩa 4 mức ưu tiên để đánh dấu các loại ứng dụng khác nhau, các mức ưu tiên này sử dụng 2 bits để biểu diễn và 2 bits này được thêm vào cuối mỗi gói. Lớp MAC trong giao thức PRIMA sử dụng 2 giao thức khác nhau: Classifier MAC thực hiện đưa các gói tin vào 4 hàng đợi tùy theo độ ưu tiên của chúng. Giao thức còn lại là Channel Access MAC (CA-MAC) sử dụng CSMA/CA và TDMA. Các khe truy cập ngẫu nhiên dùng cho việc yêu cầu khe thời gian của các nút và CH quảng bá lịch trình. Các nút sẽ gửi dữ liệu theo lịch trình định sẵn sử dụng giao thức TDMA không có xung đột. CH cũng sử dụng giao thức tương tự khi muốn truyền dữ liệu đến Base Station.

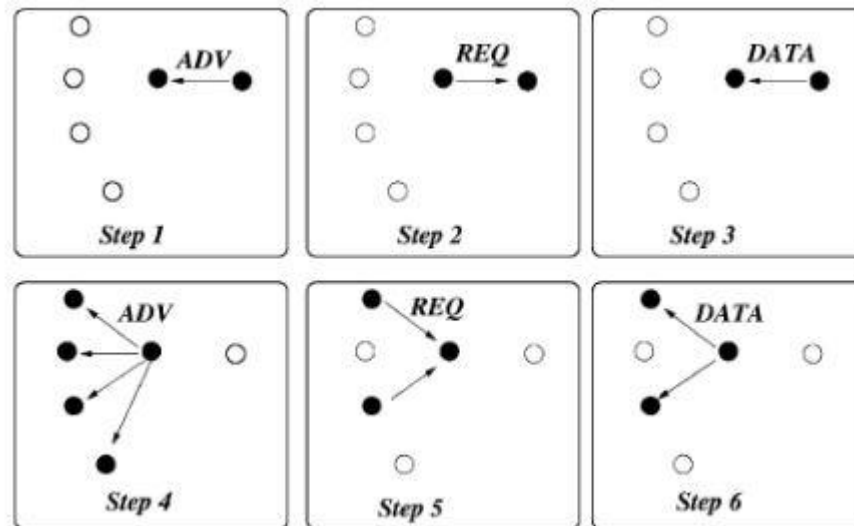
Ưu điểm chính của PRIMA là việc nó giảm trễ truyền tin theo yêu cầu. Ngoài ra nó cũng thừa hưởng những ưu điểm của LEACH trong việc bầu chọn CH làm tăng thời gian hoạt động. Tuy nhiên nhược điểm chính của PRIMA là khi nút CH ngừng hoạt động thì các nút khác cũng ngừng hoạt động theo cho đến khi CH mới được bầu lên. Các gói tin Overhead cũng làm tăng mức tiêu thụ năng lượng.

3.4 Các giao thức định tuyến trong WSN

a, Giao thức SPIN [9]

SPIN (Sensor Protocols for Information via Negotiation) là một họ các giao thức có khả năng thích ứng được thiết kế để khắc phục nhược điểm của giao thức phát tràn cổ điển bằng thỏa thuận và khả năng thích ứng tài nguyên. Họ giao thức SPIN được thiết kế dựa trên hai ý tưởng cơ bản:

- Các nút Sensor hoạt động hiệu quả hơn và có thể bảo tồn năng lượng bằng cách chỉ gửi các số liệu mô tả về số liệu cảm biến thay vì gửi toàn bộ số liệu.
- Các nút Sensor phải giám sát sự thay đổi trong tài nguyên năng lượng của chúng. SPIN có ba kiểu bản tin là ADV (quảng cáo), REQ (yêu cầu) và DATA (số liệu). Trước khi gửi bản tin DATA, Sensor phát quảng bá một bản tin ADV gồm một bản mô tả (meta-data) của DATA chỉ ra trong bước 1 trên hình 2.8. Nếu một nút Sensor lân cận quan tâm đến số liệu này, nó sẽ gửi một bản tin REQ về số liệu và bản tin DATA sẽ được gửi tới nút Sensor này như các bước 2 và 3. Nút Sensor lân cận này lại lặp lại tiến trình trên như chỉ ra ở bước 4, 5, 6. Kết quả là tất cả các nút Sensor trong toàn mạng Sensor có nhu cầu về số liệu này sẽ nhận được bản sao của số liệu.



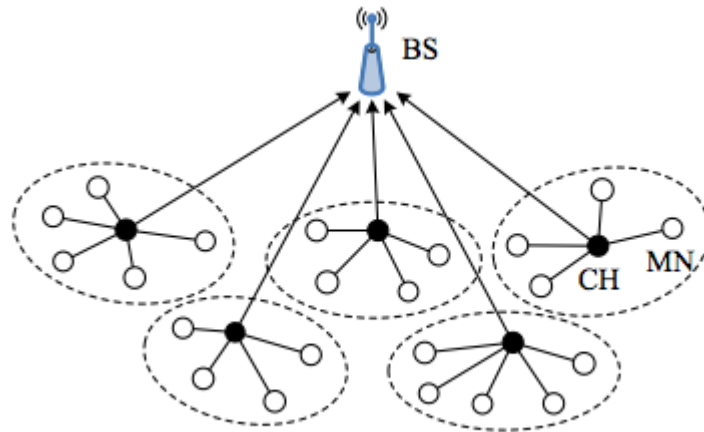
Hình 3-14 Giao thức SPIN

b, Giao thức LEACH [9]

Vấn đề mà mạng WSN phải đối mặt là giới hạn về mặt năng lượng. LEACH (Low-Energy Adaptive Clustering Hierachy) là giao thức phân cấp theo cụm thích ứng năng lượng thấp. Nó dựa trên thuật toán phân nhóm và có những đặc trưng sau:

- Các nút có thể phân bố ngẫu nhiên, và tự hình thành cụm. Nút chủ cụm sẽ điều khiển các nút khác trong cụm gửi dữ liệu đến nó. Quá trình xử lý dữ liệu là nút chủ cụm sẽ tổng hợp dữ liệu từ các nút gửi đến rồi gửi tới BS (Base Station).

- Trong mạng WSN dữ liệu của các nút gửi về thường có mối tương quan với nhau, người dùng cuối không cần yêu cầu tất cả dữ liệu hay chỉ cần các thông tin ở mức cao của dữ liệu mô tả về các sự kiện xuất hiện trong môi trường mà các nút cảm biến được. Vì các tín hiệu dữ liệu được gửi từ các nút đặt gần nhau có sự tương quan rất lớn, do đó chúng ta chọn sử dụng kiến trúc phân nhóm cho LEACH. Điều này cho phép tất cả dữ liệu từ các nút trong phạm vi cụm sẽ được xử lý cục bộ, giảm được lượng dữ liệu truyền tới người dùng cuối. Do đó tiết kiệm được năng lượng của các nút. Trong giao thức LEACH, các nút tự tổ chức thành các cụm, trong đó một nút sẽ đóng vai trò là nút chủ cụm. Tất cả các nút không phải là nút chủ sẽ phải truyền dữ liệu của nó tới nút chủ, nút chủ cụm phải nhận dữ liệu từ tất cả các nút thành viên trong cụm, thực hiện xử lý dữ liệu cục bộ (tổng hợp dữ liệu - aggregation), rồi truyền tới BS. Bởi vậy, việc trở thành nút chủ sẽ tiêu hao nhiều năng lượng hơn các nút không được chọn là nút chủ. Mà năng lượng của các nút là giới hạn, nếu nút chủ được chọn cố định trong suốt thời gian sống của mạng, như trong giải thuật phân nhóm tĩnh (static clustering), thì các nút chủ sẽ hết năng lượng rất nhanh.



Hình 3-15 Giao thức LEACH

Khi nút chủ chết, tất cả các nút trong cụm sẽ không có khả năng trao đổi thông tin nữa. Vì vậy, LEACH thực hiện ngẫu nhiên việc quay vòng vai trò nút chủ giữa các nút có năng lượng cao trong số tất cả các nút để tránh sự tiêu hao năng lượng trên một nút cụ thể trong mạng. Với cách này, tiêu hao năng lượng liên quan đến việc trở thành nút chủ sẽ được phân bố đều cho tất cả các nút.



Hình 3-16 Sơ đồ thời gian hoạt động của LEACH

c, Giao thức PEGASIS [9] [12]

Là một sự cải tiến của giao thức LEACH. Thay vì hình thành nhiều cụm, PEGASIS hình thành các dây chuyền từ các nút cảm biến mà mỗi nút truyền và nhận từ 1 nút hàng xóm và chỉ một nút được chọn từ chuỗi đó để truyền đến trạm cơ sở (sink). Thu thập dữ liệu chuyển từ nút tới nút, được tổng hợp và cuối cùng gửi đến các trạm cơ sở. Sự khác biệt với LEACH là sử dụng định tuyến nhiều chặng (multi hops) bằng cách hình thành chuỗi và lựa chọn chỉ một nút để truyền đến trạm cơ sở, thay vì sử dụng nhiều nút. PEGASIS đã được thể hiện tốt hơn LEACH về mặt hiệu suất khoảng 100-300% [12], đối với các kích cỡ và cấu trúc mạng khác nhau. Như vậy hiệu suất đạt được thông qua việc loại bỏ các chi phí gây ra bởi sự hình thành cụm năng động trong LEACH và thông qua giảm số lần truyền và nhận dữ liệu bằng cách dùng tổng hợp dữ liệu. Tuy nhiên, PEGASIS có thể gây ra sự chậm trễ quá lớn đối với các nút xa trên chuỗi đó. Thêm vào đó trường nhóm duy nhất có thể trở thành nên nút thất nút cổ chai. PEGASIS phân cấp là một mở rộng của PEGASIS, với mục tiêu giảm phát sinh chậm trễ cho các gói tin trong việc truyền đến trạm cơ sở và đề xuất một giải pháp cho vấn đề thu thập dữ liệu bằng cách xem xét số liệu độ trễ và năng lượng. Để giảm sự chậm trễ trong PEGASIS, cần truyền đồng thời các thông điệp dữ liệu. Các giao thức dựa trên dây chuyền với các nút CDMA có khả năng xây dựng một chuỗi các nút,

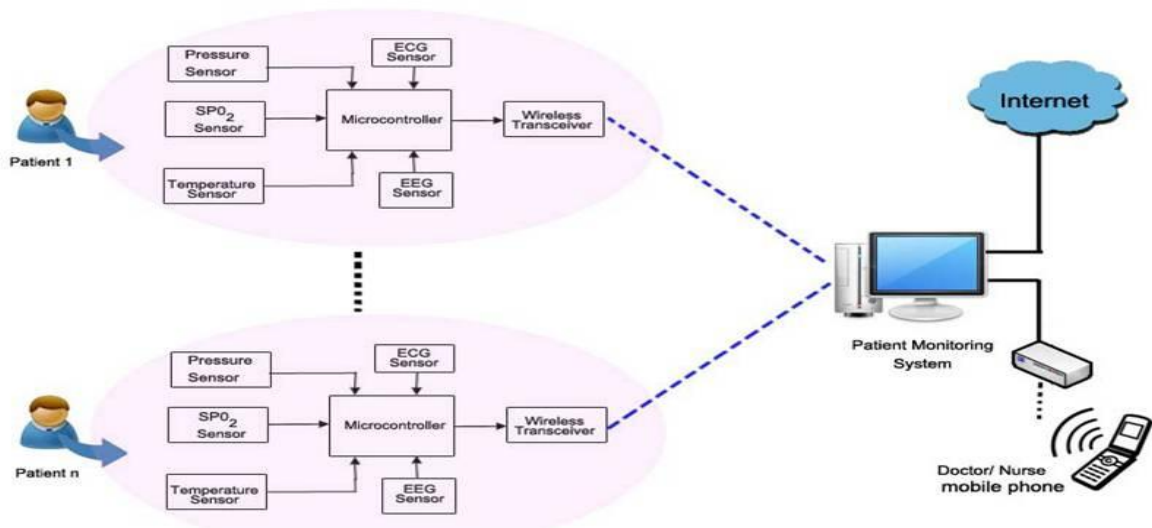
có hình thức giống như một cây phân cấp, và mỗi nút được chọn trong một mức cụ thể truyền dữ liệu tới nút ở mức cao trong hệ thống phân cấp đó. Phương pháp này đảm bảo truyền dữ liệu song song và làm giảm sự chậm trễ đáng kể.

3.5 Hệ thống mạng WSN trong theo dõi sức khỏe

Hiện nay các hệ thống WSN được ứng dụng rất nhiều trong các hệ thống theo dõi chăm sóc sức khỏe, đặc biệt là đối với các bệnh viện. WSN đã chứng tỏ ưu điểm trong việc theo dõi sức khỏe các bệnh nhân nhờ việc không bị giới hạn bởi các kết nối có dây. Bệnh nhân có thể đeo cảm biến trên người và đi lại trong một khu vực nhỏ, miễn là các cảm biến vẫn nằm trong phạm vi kết nối của trạm giám sát thì các tín hiệu cảm biến được trên cơ thể vẫn liên tục được báo về cho bác sĩ hoặc người nhà theo dõi thông qua các hệ thống liên lạc. Đặc biệt với chức năng cảm biến hướng sự kiện bác sĩ sẽ dễ dàng phát hiện các tình huống cấp cứu và đưa ra các biện pháp xử lý kịp thời. Trong phần này tôi sẽ trình bày một số mô hình WSN đang được triển khai trong thực tế và khảo sát một số cảm biến dùng trong giám sát sức khỏe con người.

3.5.1 Mô hình mạng WSN trong theo dõi sức khỏe [13]

Hệ thống WSN giám sát sức khỏe sử dụng các cảm biến để liên tục theo dõi các thông số và truyền dữ liệu đến trạm giám sát từ xa. Có 3 kiểu cảm biến thường được sử dụng: 1) cảm biến gắn trên quần áo. 2) cảm biến trên các thiết bị đeo trên người. 3) cảm biến dán trực tiếp lên cơ thể.



Hình 3-17 Hệ thống WSN giám sát sức khỏe

Một hệ thống WSN thường bao gồm các thành phần như sau, Hình 3-17 Hệ thống WSN giám sát sức khỏe:

- Nút cảm biến: Tùy vào hệ thống mạng mà nút cảm biến thực hiện các công việc cảm ứng khác nhau. Ví dụ: Đối với việc theo dõi phụ nữ mang thai, các nút cảm biến cần có khả năng theo dõi không chỉ các thông số cơ thể của người mẹ mà còn cần có khả năng nhận diện các thông số của bào thai phát

ra. Các nút cảm biến trong mạng WSN chủ yếu sử dụng các kết nối không dây, kiểu cảm biến sẽ được cấu hình bởi các bác sĩ. Thông thường có 4 kiểu cảm biến: Cảm biến định kỳ, cảm biến liên tục, cảm biến theo yêu cầu và cảm biến sự kiện.

- Nút điều khiển trung tâm: Thường được gọi là nút Sink, nút này có vai trò tập hợp và đóng gói dữ liệu gửi đến từ các nút cảm biến sau đó gửi cho trạm giám sát.
- Hệ thống định vị GPS: Một số hệ thống WSN theo dõi sức khỏe còn trang bị GPS lên các nút cảm biến nhằm nắm được một cách chính xác vị trí của các bệnh nhân cần theo dõi.

Một số hệ thống WSN hiện nay đã được đưa vào thực tế triển khai trong các bệnh viện. Tuy nhiên trong quá trình triển khai nó vẫn còn bộc lộ nhiều hạn chế như sau:

- Các nút cảm biến gắn trên quần áo có thể bị nhiễu do tĩnh điện từ vải quần áo. Cảm biến dán trên người trở nên không hiệu quả trong thời gian dài sử dụng do các chất phụ gia hữu cơ (gel) được sử dụng khô dần theo thời gian. Một số thiết bị đeo trên người còn cồng kềnh gây cản trở vận động .v.v
- Các nút cảm biến hoạt động độc lập nên hạn chế về nguồn năng lượng.
- Các hạn chế trong truyền thông không dây. Ví dụ vấn đề định tuyến đa chặng trong truyền thông không dây rất phức tạp gây ảnh hưởng đến sự chậm trễ của dữ liệu. Vấn đề chất lượng kết nối giữa các nút.
- Vai trò của nút Sink trong mạng là rất quan trọng cần phải có những biện pháp xử lý trong trường hợp nút Sink ngừng hoạt động đột ngột,v.v.

Chính vì những lý do trên việc lựa chọn công nghệ, thiết kế, triển khai và phân tích hiệu suất của mạng cảm biến không dây phục vụ giám sát sức khỏe trở nên rất quan trọng và hiện đang là đề tài nghiên cứu được nhiều sự quan tâm trên thế giới.

3.5.2 Khảo sát một số hệ thống giám sát sức khỏe. [14]

Trong phần này, tôi sẽ trình bày một số tham khảo về các dự án phát triển hệ thống mạng cảm biến cho giám sát sức khỏe.

Nhìn chung các hệ thống WSN cho giám sát sức khỏe được phát triển theo 3 kiểu cảm biến chính đó là: Cảm biến trên quần áo, cảm biến trên các thiết bị đeo trên người, và cảm biến dán trực tiếp lên cơ thể. Tuy nhiên các dự án phát triển hệ thống cảm biến cho giám sát sức khỏe thường chia thành 5 loại:

Loại 1: Hệ thống sử dụng bảng vi điều khiển (microcontroller board) làm nền tảng để cảm nhận các dữ liệu sinh học, sử dụng truyền dẫn có dây để truyền các tin hiệu đó về bảng xử lý. Một đại diện tiêu biểu của hệ thống này là dự án LiveNet của phòng thí nghiệm MIT. Hệ thống này sử dụng một thiết bị di động trên nền tảng Linux, kết hợp với một mô đun cảm ứng để thu thập dữ liệu và một bảng tích hợp các

thông tin sinh học để tính toán và xử lý dữ liệu. Hệ thống này đã được triển khai phục vụ theo dõi bệnh nhân Parkinson, bệnh nhân động kinh và gắn trên người các quân nhân hoạt động trong điều kiện môi trường khắc nghiệt. Nhìn chung mục tiêu của LiveNet là đưa ra các cảnh báo thời gian thực, phân loại các trường hợp bệnh lý và cung cấp một hệ thống thông tin phản hồi, theo dõi sức khỏe khép kín.

Loại 2: Hệ thống giám sát sức khỏe này dựa trên những cảm biến được tích hợp vào quần áo giúp theo dõi, phân tích các hoạt động, chỉ số sinh học trên cơ thể bệnh nhân. Qua đó đưa ra các cảnh báo và chuẩn đoán sớm tình trạng bệnh lý.

Loại 3: Hệ thống giám sát sức khỏe dựa trên các nút cảm biến gắn trên cơ thể. Hệ thống này còn có tên gọi là WWBAN (wearable wireless body area network). Hệ thống này bao gồm các nút cảm biến có khả năng tự trị, sử dụng các kết nối không dây để trao đổi dữ liệu với trạm điều phối hoặc nút trung tâm. Đại diện tiêu biểu của loại này là dự án Code Blue của các nhà nghiên cứu đến từ đại học Havard.

Loại 4: Hệ thống giám sát sức khỏe sử dụng cảm biến kết nối bluetooth và điện thoại di động. Dự án có HealthGear của tập đoàn MicroSoft. Hệ thống này bao gồm một nút cảm biến có khả năng thu thập các thông tin sức khỏe như nhiệt độ cơ thể, nhịp tim, độ bão hòa oxy trong máu. Nút cảm biến này được tích hợp một mô đun sử dụng bluetooth và sử dụng pin. Dữ liệu sẽ được gửi từ nút cảm biến qua điện thoại di động của người dùng.

Loại 5: Ngoài 4 loại trên thì các nhà nghiên cứu cũng đưa ra những cách tiếp cận khác cho từng mục đích khác nhau trong giám sát sức khỏe. Cụ thể như với dự án AUBADE được phát triển bởi đại học Ioannina Hy Lạp, các nhà nghiên cứu đã đưa ra đề xuất sử dụng một chiếc mặt nạ có gắn cảm biến ở bên trong để giúp đánh giá các trạng thái cảm xúc của con người. Hay một số các hệ thống nhận dạng chuyển động sử dụng các găng tay có gắn cảm biến hoặc các vòng đeo ở ngực để đo nhịp tim.

Trên đây là 5 loại hệ thống cảm biến đeo trên người được sử dụng để giám sát sức khỏe. Và các dự án đang được phát triển. Bảng 3-1 Khảo sát một số hệ thống giám sát sức khỏe dưới đây sẽ đưa ra một số chi tiết và so sánh về các dự án đang được phát triển. Các thông tin so sánh bao gồm: Kiểu triển khai phần cứng, các mô đun kết nối, các loại tín hiệu cảm biến được, và một số ứng dụng trong y tế.

Các loại tín hiệu cảm biến được bao gồm: ECG – electrocardiogram (điện tâm đồ), HR – heart rate (nhịp tim) , EMG - Electromyogram (điện cơ), BP – blood pressure (huyết áp), T – temperature (nhiệt độ cơ thể), A – activity (hoạt động) v.v

Tên dự án/tổ chức thực hiện	Kiểu phần cứng	Kiểu kết nối	Loại tín hiệu cảm nhận được	Ứng dụng trong y học
LiveNet/MIT	Microcontroller board	Có dây, radio, GPRS	ECG, BP, T, EMG, v.v	Theo dõi bệnh phân parkinson, bệnh nhân động kinh.
WEALTHY/EU IST FP6	PDA, cảm biến gắn trên quần áo	Bluetooth, GPRS	ECG, A, T	Theo dõi sức khỏe người cao tuổi
Code Blue/Harvard University	Sensor mote	ZigBee	ECG, SpO2, A	Theo dõi các trạng thái sinh học theo thời gian thực
Body area network/Malta Univ.	Zigbee-based mote & Zigbee-based custom base device	ZigBee, Wi-Fi, GPRS	ECG, BP, R	Phát hiện và dự đoán các trạng thái sinh học
HealthGear/Microsoft	Sensor and Cell-phone	Bluetooth, GPRS	HR, SpO2	Theo dõi giấc ngủ
AUDABE/Dept . of Medical Physics, Ioannina, Greece	Mặt nạ, Găng tay, Vòng cảm biến	Kết nối có dây, Bluetooth, Wi-fi	ECG, R, EMG, A	Theo dõi trạng thái, hoạt động

Bảng 3-1 Khảo sát một số hệ thống giám sát sức khỏe [14]

CHƯƠNG 4 MÔ PHỎNG VÀ ĐÁNH GIÁ HIỆU NĂNG MẠNG

4.1 Giới thiệu các công cụ mô phỏng

Hiện nay với sự phát triển của khoa học công nghệ, có khá nhiều công cụ mô phỏng mạng nói chung và mạng cảm biến nói riêng đã được phát triển. Các công cụ mô phỏng này có vai trò quan trọng trong việc đánh giá, phân tích hiệu suất của các hệ thống mạng mà lại giúp tiết kiệm chi phí khi không phải triển khai thực tế. Trong phần này tôi sẽ giới thiệu 3 công cụ mô phỏng được sử dụng nhiều hiện nay là: OMNet++, Ns-2 và TOSSIM.

4.1.1 OMNet++

OMNet++ là công cụ mô phỏng mã nguồn mở, được sử dụng rất nhiều trong việc mô phỏng mạng. OMNet++ được viết bằng ngôn ngữ C++ và có giao diện GUI. OMNet++ hỗ trợ khá đầy đủ các giao thức và mô hình truyền thông của mạng Internet như TCP, UDP, Ipv4, Ipv6, các giao thức và chuẩn dành cho mạng không dây như 802.11, MANET, các mô hình mạng thể hệ mới DiffServ, MPLS v.v. Trong đó có mô đun mở rộng Castalia được phát triển phục vụ mục đích mô phỏng mạng WSN và mạng BAN khá đầy đủ.

4.1.2 TOSSIM

Là một bộ công cụ mô phỏng WSN khá điển hình, cụ thể TOSSIM mô phỏng toàn bộ các ứng dụng trên hệ điều hành TinyOS – hệ điều hành chuyên biệt cho các nút cảm biến được nêu trong mục 2.2.2. TOSSIM mô phỏng mạng WSN bằng cách thay thế các thành phần vật lý bằng các mô đun mô phỏng. Mô đun ứng dụng TinyOS và TOSSIM được biên dịch và liên kết thành một thư viện phần mềm. Trong TOSSIM Python được sử dụng làm trình thông dịch và chạy mô phỏng. Một số thành phần viết bằng C++ cũng có thể được sử dụng. Tuy nhiên với các nhược điểm về việc thiếu các mô hình tiêu thụ năng lượng, các tài liệu hướng dẫn chi tiết, TOSSIM đã không đáp ứng đủ yêu cầu của một phần mềm mô phỏng sử dụng để phân tích hiệu suất của WSN.

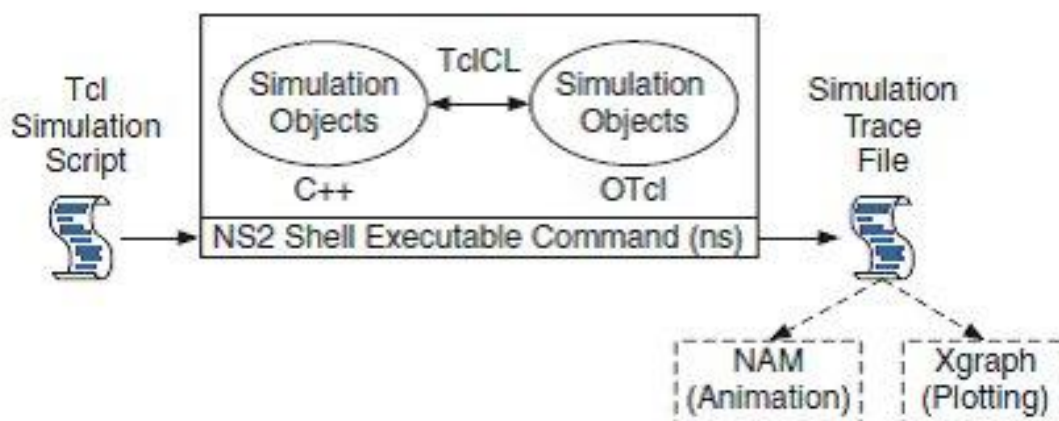
4.1.3 NS2 (Network Simulator 2)

Là phần mềm mô phỏng mạng được phát triển bởi trường đại học UC Berkeley, được cung cấp dưới dạng mã nguồn mở và có thể chạy được cả trên môi trường hệ điều hành Windows lẫn Linux.

NS-2 là một chương trình biên dịch mã Otcl được liên kết tới các thư viện viết bằng C++. Các đối tượng cơ sở trong Ns-2 như TCP, UDP, các giao thức định tuyến v.v. được xây dựng bằng ngôn ngữ C++. Đầu vào của Ns-2 là tập lệnh Otcl được viết bằng ngôn ngữ TCL là ngôn ngữ kịch bản. Kết quả mô phỏng của Ns-2 có thể được kết xuất ra file text dạng .nam để có thể quan sát bằng hình ảnh mô tả trực quan với ứng dụng NAM (Network Animator), hoặc có thể được kết xuất ra file vết .tr có dạng

text để có thể phân tích, kết xuất và vẽ đồ thị theo yêu cầu của người phân tích với các công cụ như Xgraph và GnuPlot.

Ns-2 có nhiều phiên bản chạy trên các hệ điều hành khác nhau và được phát triển bởi cộng đồng người sử dụng đồng đảo trên toàn thế giới.



Basic architecture of NS.

Hình 4-1 Kiến trúc của NS-2

Để có thể thực hiện mô phỏng WSN trong luận văn này tôi có sử dụng thêm mô đun MANNASIM [14] là một công cụ mở rộng cho Ns-2 phục vụ cho việc mô phỏng hoạt động của WSN. MANNASIM được sử dụng để hỗ trợ Ns-2 cũng như người sử dụng trong việc cấu hình các kịch bản mô phỏng, điều chỉnh các thông số trong mạng như: số lượng node, vị trí phân bố các node, diện tích khu vực thực hiện mô phỏng, kiểu đường liên kết v.v. Sau khi thực hiện mô phỏng mannasim có thể sinh ra kết quả mô phỏng bằng các tệp vết trong đó có đầy đủ thông tin về tình trạng của hệ thống mạng. Ví dụ như tỉ lệ mất gói tin, độ trễ, mức năng lượng tiêu thụ, v.v giúp cho những người nghiên cứu có thể dễ dàng kết xuất đồ thị và đánh giá. Bảng 4-1 So sánh đặc điểm của Mannasim với một số công cụ mô phỏng khác [15].

Framework	Đặc điểm	Mannasim
SENSE	Thiết kế phục vụ: người dùng cao cấp, người xây dựng mạng lưới và thiết kế. Tuy nhiên vẫn thiếu các mô hình tổng quát phục vụ cho việc cấu hình	Cung cấp đầy đủ các thông số cấu hình tổng quát cho các thành phần mạng phục vụ người sử dụng, xây dựng và thiết kế mạng
SensorSim	Được xây dựng từ Ns-2 nhưng không còn tiếp tục phát triển	Tối ưu hơn cho WSN và vẫn đang được cộng đồng tiếp tục phát triển

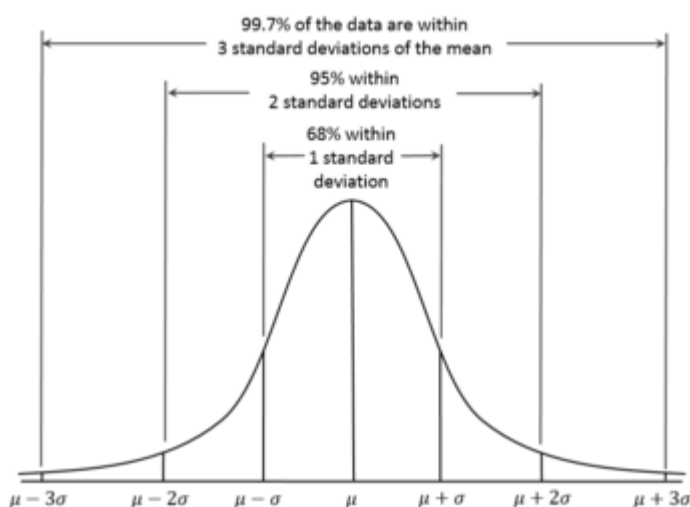
Avrora	Mô phỏng hành vi của các nút cảm biến ở mức độ chi thị	Mô tả rõ ràng các sự kiện truyền thông của các nút cho người sử dụng
NetTopo	Mục tiêu là đánh giá sự ảnh hưởng của các thuật toán khác nhau trong WSN	Mục tiêu là đánh giá sự ảnh hưởng của các thành phần mạng khác nhau, ví dụ như số nút hay cách tổ chức mạng
Castalia	Cung cấp môi trường để kiểm tra thuật toán và giao thức trong mô hình thực tế	Đề xuất công cụ tối ưu để đánh giá hiệu năng của WSN

Bảng 4-1 So sánh đặc điểm của Mannasim với một số công cụ mô phỏng khác [15]

4.2 Vấn đề nguồn sinh lưu lượng trong WSN cho giám sát sức khỏe [16]

Trong luận văn này, tôi xây dựng một mô hình WSN cho giám sát sức khỏe con người với các tham số sức khỏe chính là: Nhịp tim, nhiệt độ cơ thể, và huyết áp.

Để đảm bảo tính chính xác cũng như độ tin cậy của kết quả mô phỏng, công việc quan trọng nhất là phải tạo ra được một nguồn sinh lưu lượng có phân bố xác suất giống với các dữ liệu trong thực tế. Đối với các tham số sinh học trên cơ thể người, các dữ liệu thu được có phân bố khá giống với phân bố chuẩn (normal distribution) trong xác suất thống kê [16]. Cụ thể như sau.



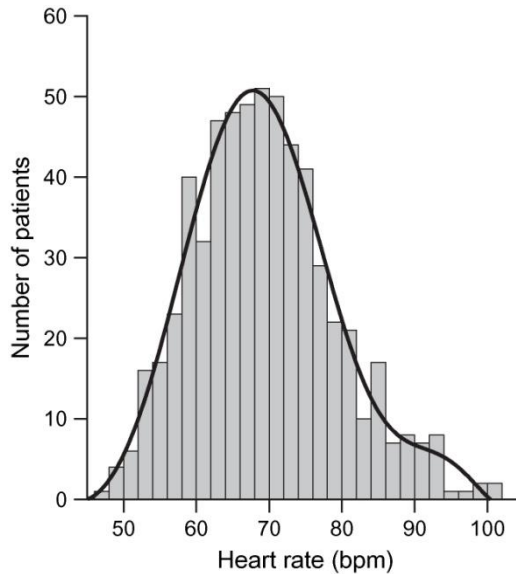
Hình 4-2 Phân phối chuẩn

Phân bố chuẩn (còn gọi là normal distribution hay phân bố gauss) trên $\mathbf{R}$ với giá trị trung bình (kí hiệu là μ) và độ lệch chuẩn (kí hiệu là σ) là phân bố liên tục với hàm mật độ: $\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$.

Đồ thị của hàm mật độ phân bố chuẩn có hình cái chuông, trong đó trung điểm là $x = \mu$ và độ cao chuông chính bằng $\frac{1}{\sigma\sqrt{2\pi}}$. Hình 4-2 Phân phối chuẩn cho thấy hầu hết xác suất của một phân bố chuẩn nằm trong đoạn $[\mu - 3\sigma, \mu + 3\sigma]$, chỉ có 0.3%

nằm ngoài đoạn đó. Vì vậy nếu X là một biến ngẫu nhiên có phân bố xác suất là phân bố chuẩn với các tham số $[\mu, \sigma]$ thì có 99,7% khả năng giá trị của X nằm trong đoạn $[\mu - 3\sigma, \mu + 3\sigma]$.

Đây là một phân bố rất quan trọng trong xác suất thống kê. Vì nhiều phân bố xác suất gặp trong thực tế có dáng điệu rất giống với phân bố chuẩn, ví dụ như phân bố chiều cao của đàn ông, phân bố chỉ số IQ, giá chứng khoán. .v.v. So sánh Hình 4-3 Thống kê nhịp tim và Hình 4-2 Phân phối chuẩn cũng cho thấy sự tương đồng về mặt đồ thị. [16]



Hình 4-3 Thống kê nhịp tim

Chính vì vậy trong luận văn này, tôi sử dụng các hàm sinh lưu lượng tuân theo phân bố chuẩn được tích hợp trong công cụ Ns-2 và mô đun Mannasim để mô phỏng việc cảm biến các thông số về nhiệt độ, nhịp tim và huyết áp trên cơ thể người. Trường hợp cần mô phỏng các tham số không tuân theo phân bố chuẩn mà tuân theo phân bố hàm mũ hoặc phân bố Poisson sẽ là hướng phát triển tiếp theo của đề tài luận văn này.

Phần tiếp theo của luận văn sẽ trình bày về việc lựa chọn giá trị trung bình μ và độ lệch chuẩn σ cho các kịch bản mô phỏng dựa vào các thống kê về số liệu y sinh.

4.3 Thiết lập kịch bản mô phỏng

Sau khi xây dựng xong nguồn sinh dữ liệu cho các tham số sức khỏe con người. Tôi sẽ thiết lập các kịch bản mô phỏng dựa trên các số liệu sức khỏe thu thập được từ các thống kê về y sinh học. Cụ thể như sau.

4.3.1 Các tham số sinh học

a, Nhịp tim [17]

Theo thống kê về số liệu nhịp tim từ trung tâm kiểm soát và phòng ngừa dịch bệnh hoa kỳ CDC. Nhịp tim trung bình của một người nam giới trưởng thành là 71

nhịp/phút và nữ giới là 74 nhịp/phút [16]. Các thông số về nhịp tim thường phụ thuộc vào rất nhiều yếu tố như độ tuổi, giới tính, màu da, khu vực sinh sống, trạng thái hoạt động .v.v Tuy nhiên nhìn chung nhịp tim được cho là bình thường rơi vào khoảng 60~100 nhịp/phút. Tùy vào điều kiện thực tế và thể trạng của bệnh nhân mà nhịp tim có thể tăng hoặc giảm trong một thời gian ngắn. Nhịp tim lớn hơn 100 nhịp/phút được coi là nhanh và ở mức lớn hơn 120 nhịp/phút khi người bệnh trong trạng thái bình thường thì được coi là nguy hiểm. Tuy nhiên đối với những người đang chơi thể thao nhịp tim có thể tăng lên tới đa tới 190 nhịp/phút ở thanh niên. Bảng 4-2 Các tham số trung bình của nhịp tim sẽ tổng hợp các tham số cần thiết để mô phỏng đại lượng này.

Tham số	Giá trị
Loại phân bố	Phân bố chuẩn (Normal Distribution)
Giá trị trung bình (Mean)	74 nhịp/phút
Giá trị tối đa	190 nhịp/phút
Độ lệch chuẩn (Standard Deviation)	5.0 ~ 15.0

Bảng 4-2 Các tham số trung bình của nhịp tim

Việc phát hiện sự thay đổi giá trị trung bình và đưa ra các cảnh báo là rất quan trọng trong các mô hình giám sát sức khỏe. Các giá trị trên có thể thay đổi trong một khoảng thời gian rất ngắn tùy thuộc vào nhiều tình huống. Ví dụ nhịp tim của một người bình thường là 75 nhịp/phút, nhưng trong trường hợp bị ngoại cảnh tác động nhịp tim có thể tăng lên 90~120 nhịp/phút trong khoảng thời gian ngắn. Nếu không có các phát hiện và cảnh báo sớm sẽ đưa bệnh nhân đến tình trạng nguy kịch. Do đó cần đặc biệt lưu ý và có cơ chế cảnh báo khi nhịp tim của bệnh nhân vượt ngưỡng cho phép.

b, Huyết áp [18] [19]

Là chỉ số thể hiện áp lực máu lên thành động mạch, được đo bằng đơn vị là mil-li-mét thủy ngân (mm Hg). Huyết áp được xác định bằng hai chỉ số, thường được viết dưới dạng một tỷ số. Chỉ số trên là “tâm thu” là mức huyết áp cao nhất trong mạch máu xảy ra khi tim co bóp. Chỉ số dưới là “tâm trương” là mức huyết áp thấp nhất trong mạch máu xảy ra khi tim ngưng co bóp, thả lỏng.

Huyết áp ở người trưởng thành bình thường được xác định là có chỉ số tâm thu là 120 mm Hg, tâm trương là 80 mm Hg [17]. Tuy nhiên mức huyết áp ổn định và tốt nhất cho tim mạch là ở mức 105 mm Hg tâm thu và dưới 65 mm Hg tâm trương. Bệnh tăng huyết áp là một căn bệnh rất phổ biến, nó có thể tiến triển thầm lặng mà người bệnh không hề hay biết. Tăng huyết áp được xác định là khi có huyết áp tâm thu từ 140 mm Hg trở lên và/hoặc huyết áp tâm trương từ 90 mm Hg trở lên [18]. Các mức huyết áp tâm thu và tâm trương bình thường có ý nghĩa đặc biệt quan trọng đối với chức năng hiệu quả của các cơ quan sinh tồn như tim, não và thận, và đối với sức khỏe nói chung và trạng thái khỏe mạnh. Vì vậy việc theo dõi các chỉ số huyết áp là đặc biệt quan trọng.

Tham số	Giá trị (MEAN $\pm$ Standard Deviation)	
	Tâm thu	Tâm trương
Giá trị trung bình	120 $\pm$ 10 mm Hg	80 $\pm$ 10 mm Hg
Bệnh nhân cao huyết áp [18]	136.7 $\pm$ 16.4 mm Hg	76.5 $\pm$ 6.2 mm Hg
Bệnh nhân huyết áp thấp	< 90 mm Hg	

Bảng 4-3 Giá trị trung bình huyết áp

c, Nhiệt độ cơ thể [20]

Giá trị trung bình của nhiệt độ cơ thể người là 98,6 °F hay 37°C. Nhiệt độ trên cơ thể con người còn thay đổi tùy thuộc vào nhiều yếu tố. Bao gồm cả vị trí đặt nhiệt kế. Tuy nhiên các giá trị trung bình nhiệt độ có thể phân chia theo bảng sau.

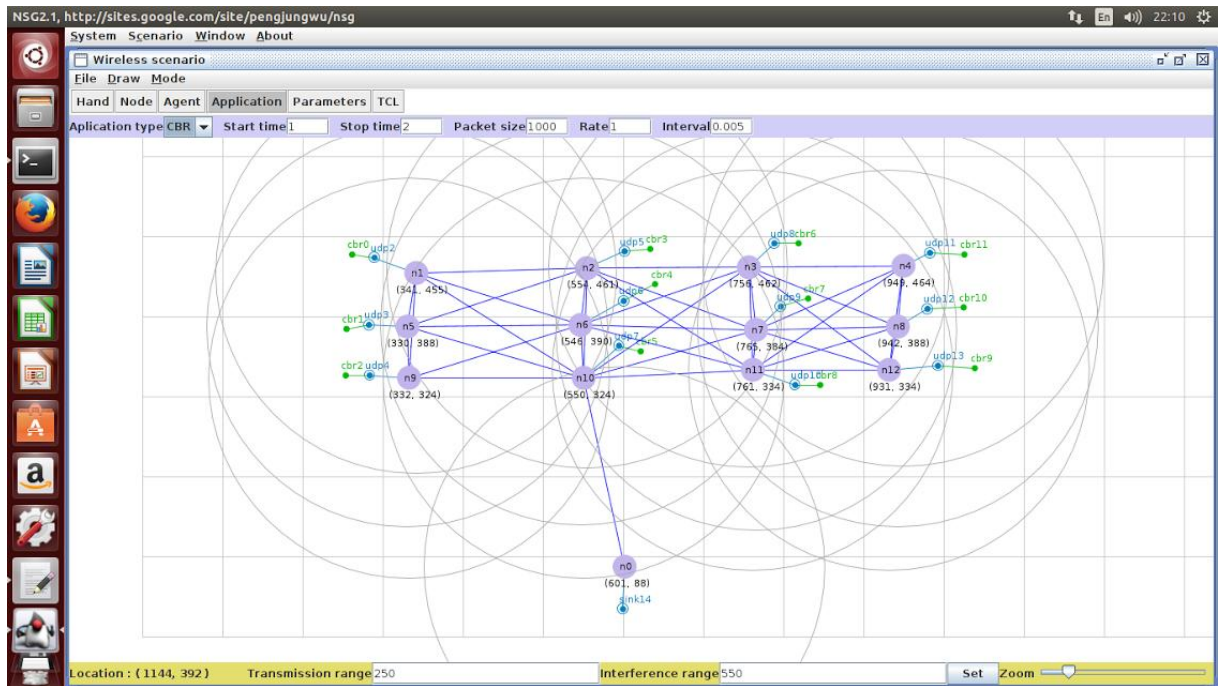
Trạng thái	Giá trị
Mất thân nhiệt	< 35°C
Bình thường	36.5 ~ 37.5 °C
Sốt	>37.5 °C hoặc 38.3 °C
Thân nhiệt cao	>37.5 °C hoặc 38.3 °C
Thân nhiệt cao nguy hiểm	>40.5 °C hoặc 41.5 °C

Bảng 4-4 Giá trị nhiệt độ trung bình

Các trạng thái giá trị trên của nhiệt độ có thể thay đổi tùy vào cơ chế đánh giá thân nhiệt khác nhau. Giá trị nhiệt độ của từng người tại cùng một thời điểm cũng khác nhau tùy thuộc vào trạng thái cơ thể họ. Các thống kê cũng chỉ ra rằng các dữ liệu về thân nhiệt cũng tuân theo phân bố chuẩn.

4.3.2 Thiết lập topo mạng và các tham số mô phỏng

Trong phần này tôi sẽ trình bày về việc thiết kế topo mạng cũng như lựa chọn các tham số, chuẩn công nghệ sẽ triển khai cho mô hình mạng cảm biến không dây phục vụ giám sát sức khỏe. Hình 4-4 Mô hình WSN cho giám sát sức khỏe đây mô tả topo WSN giám sát sức khỏe cho 4 bệnh nhân trong một phòng bệnh. Các nút n1 n2 n3 n4 đóng vai trò cảm biến nhiệt độ cơ thể, nút n5 n6 n7 n8 đóng vai trò cảm biến nhịp tim, nút n9 n10 n11 n12 đóng vai trò cảm biến huyết áp. Nút n0 đóng vai trò là nút sink và access point.



Hình 4-4 Mô hình WSN cho giám sát sức khỏe

Các công nghệ được lựa chọn cho mô hình mạng này sẽ tuân theo Bảng 4-5 Các tham số mô phỏng.

Tham số	Giá trị
Diện tích	2m x 2m
Tổng số nút	12
Số nút sink	01
Số nút access point	01
Giao thức MAC	Mac/802_11
Kiểu hàng đợi	Queue/DropTail/PriQueue
Độ dài hàng đợi	200
Giao thức định tuyến	DumbAgent
Chế độ năng lượng	Battery

Bảng 4-5 Các tham số mô phỏng

Giải thích việc lựa chọn các tham số mô phỏng:

Diện tích mô phỏng được lựa chọn 2m x 2m là tương đương với diện tích cơ thể của một người trưởng thành. Với mỗi tham số sức khỏe đã nêu có thể có từ 1 đến 3 nút theo dõi trạng thái. Các nút kết nối trực tiếp đến nút Access Point. Nút Access Point sẽ làm nhiệm vụ tổng hợp và gửi dữ liệu đến nút Sink.

Giao thức MAC theo chuẩn 802.11 được sử dụng để hạn chế các tranh chấp trong quá trình truy cập đường truyền của các nút. Cũng như các hiện tượng nút ẩn và nút hiện. Trường hợp này tất cả các nút cảm biến đều có thể kết nối trực tiếp đến Access Point vì vậy giao thức định tuyến DumbAgent được sử dụng, để đánh giá sự hiệu quả của các giao thức tầng MAC.

Chế độ năng lượng Battery đây là một mô hình năng lượng giả lập được cung cấp bởi mô đun mannasim. Mô hình này mô phỏng lại hoạt động của các viên pin trong thực tế. Độ tiêu thụ pin được tính bằng công suất cảm biến nhân với thời gian cảm biến. Độ đo này sẽ giúp đánh giá mức độ tiêu thụ năng lượng của các nút cảm biến trong mạng.

4.3.3 Các kịch bản mô phỏng

Mannasim đã cung cấp 4 phương thức cảm biến là cảm biến theo định kỳ, cảm biến liên tục (continuous), cảm biến sự kiện (event driven) và cảm biến theo yêu cầu (on demand). Trong luận văn này tôi sẽ mô phỏng sử dụng 03 phương thức cảm biến là 0-periodic, 1-continuous, 3-event driven. Với các tham số mô phỏng tuân theo các kịch bản sau.

a, Kịch bản 1: Sử dụng phương thức cảm biến định kỳ.

Mục đích của tôi trong kịch bản này là mô phỏng một hệ thống giám sát sức khỏe theo định kỳ. Với các hàm sinh dữ liệu theo phân bố chuẩn. Sử dụng các giá trị trung bình và độ lệch chuẩn của các thông số sức khỏe của người bình thường. Cụ thể các giá trị tuân theo Bảng 4-6 Kịch bản 1 Thời gian mô phỏng là 120s. Kết quả mô phỏng sẽ được sử dụng để vẽ đồ thị báo cáo trạng thái sức khỏe giúp các bác sĩ dễ dàng hơn trong việc theo dõi và chẩn đoán.

Tham số	Sensing_interval	Sensing_type	Giá trị trung bình (Avg)	Độ lệch chuẩn (Std)
Nhiệt độ cơ thể (Body Temperature)	3.0	0	37.5	1.0
Nhịp tim (Heart Rate)	3.0	0	74.0	5.0
Huyết áp (Blood Pressure)	3.0	0	120.0	10.0

Bảng 4-6 Kịch bản 1

b, Kịch bản 2: Sử dụng phương thức cảm biến liên tục.

Kịch bản này tôi sẽ giữ nguyên các tham số giá trị trung bình và độ lệch chuẩn. Thực hiện thay đổi phương thức cảm biến từ định kỳ sang liên tục - Bảng 4-7 Kịch bản 2. Với khoảng cảm biến là 0.01s, đây là giá trị mặc định. Mỗi nút cảm biến sẽ liên tục gửi về nút Access Point các giá trị mà nó cảm nhận được. Do vậy access point sẽ phải tiếp nhận và xử lý khối lượng dữ liệu rất lớn. Tôi sẽ sử dụng tệp vết để đánh giá và phân tích hiệu suất của mạng trong trường hợp thử nghiệm này.

Tham số	Sensing_interval	Sensing_type	Giá trị trung bình (Avg)	Độ lệch chuẩn (Std)
Nhiệt độ cơ thể	0.01	1	37.5	1.0

(Body Temperature)				
Nhịp tim (Heart Rate)	0.01	1	74.0	5.0
Huyết áp (Blood Pressure)	0.01	1	120.0	10.0

Bảng 4-7 Kịch bản 2

c, Kịch bản 3: Sử dụng phương thức cảm biến hướng sự kiện

Theo đặc điểm của mạng WSN giám sát sức khỏe. Yếu tố nhận biết các thay đổi tham số theo thời gian thực là rất quan trọng. Vì vậy, tôi thử nghiệm một số thay đổi về giá trị trung bình và độ lệch chuẩn của tín hiệu theo các trường hợp đặc biệt. Các giá trị trung bình và độ lệch chuẩn thay đổi là các giá trị được trích ra từ các số liệu thống kê trong mục 4.3.1. Ngoài ra tôi cũng đưa thêm vào các giá trị ngưỡng cảnh báo của dữ liệu. Tôi sẽ đánh giá khả năng phản ứng của mạng cảm biến trong việc cảm nhận các sự kiện thăng giáng vượt ngưỡng.

Tham số	Sensing_type	Giá trị trung bình (Avg)	Độ lệch chuẩn (Std)	Ngưỡng cảnh báo (Max Allowed)
Nhiệt độ cơ thể (Body Temperature)	3	37.5	5.0	40.0
Nhịp tim (Heart Rate)	3	74.0	15.0	90.0
Huyết áp (Blood Pressure)	3	136.7	16.4	146.0

Bảng 4-8 Kịch bản 3

4.3.4 Thực hiện mô phỏng

Trong phần này tôi sẽ thực hiện chạy mô phỏng bằng Ns-2. Sau đó kết xuất thông tin để vẽ đồ thị bằng GnuPlot, sử dụng ngôn ngữ lập trình Perl để phân tích tệp vết và đưa ra đánh giá, phân tích hiệu năng. Các script đánh giá hiệu năng được viết dựa trên các script đánh giá hiệu năng mạng trong tài liệu [21] [22].

Mã nguồn của các kịch bản mô phỏng và script đánh giá hiệu năng có trong phần phụ lục của luận văn này.

a, Kết quả mô phỏng kịch bản 1

Thực hiện chạy Script mô phỏng cho kịch bản 1 với lệnh: ns wsn-healthcare-mode-0 > script1-output.tr. Ta được kết quả như Hình 4-5 Tập vết mô phỏng cảm biến gửi dữ liệu.

```

1 Starting Simulation...
2 num_nodes is set 12
3 Node 03 - Temperature Data - 37.482932 - Time - 3.183757
4 Node 08 - BloodPresure Data - 89.781189 - Time - 3.185813
5 Node 04 - Temperature Data - 37.986656 - Time - 3.225103
6 Node 02 - Temperature Data - 39.669478 - Time - 3.234400
7 Node 07 - HeartRate Data - 69.514730 - Time - 3.294750
8 Node 06 - HeartRate Data - 81.969978 - Time - 3.435744
9 Node 10 - BloodPresure Data - 89.427171 - Time - 3.438858
10 Node 05 - HeartRate Data - 65.260125 - Time - 3.692068
11 Node 09 - BloodPresure Data - 91.208883 - Time - 3.700968
12 Common Node 3 - Disseminating data - Time 5.184 - Destination node 1
13 channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
14 highestAntennaZ_ = 1.5, distCST_ = 129.1
15 Common Node 8 - Disseminating data - Time 5.186 - Destination node 1
16 Access point - Received a message with 1 elements
17 Message received from node 3 with 2.006025 delay :: 3.183757 5.189782
18 Access point - Received a message with 1 elements
19 Message received from node 8 with 2.008017 delay :: 3.185813 5.193830
20 Common Node 4 - Disseminating data - Time 5.225 - Destination node 1
21 Access point - Received a message with 1 elements
22 Message received from node 4 with 2.004527 delay :: 3.225103 5.229630
23 Common Node 2 - Disseminating data - Time 5.234 - Destination node 1
24 Access point - Received a message with 1 elements
25 Message received from node 2 with 2.004587 delay :: 3.234400 5.238987
26 Common Node 7 - Disseminating data - Time 5.295 - Destination node 1
27 Access point - Received a message with 1 elements
28 Message received from node 7 with 2.004847 delay :: 3.294750 5.299597

```

Hình 4-5 Tập vết mô phỏng cảm biến gửi dữ liệu

Hình ảnh tập vết được sinh ra sau khi chạy mô phỏng.

```

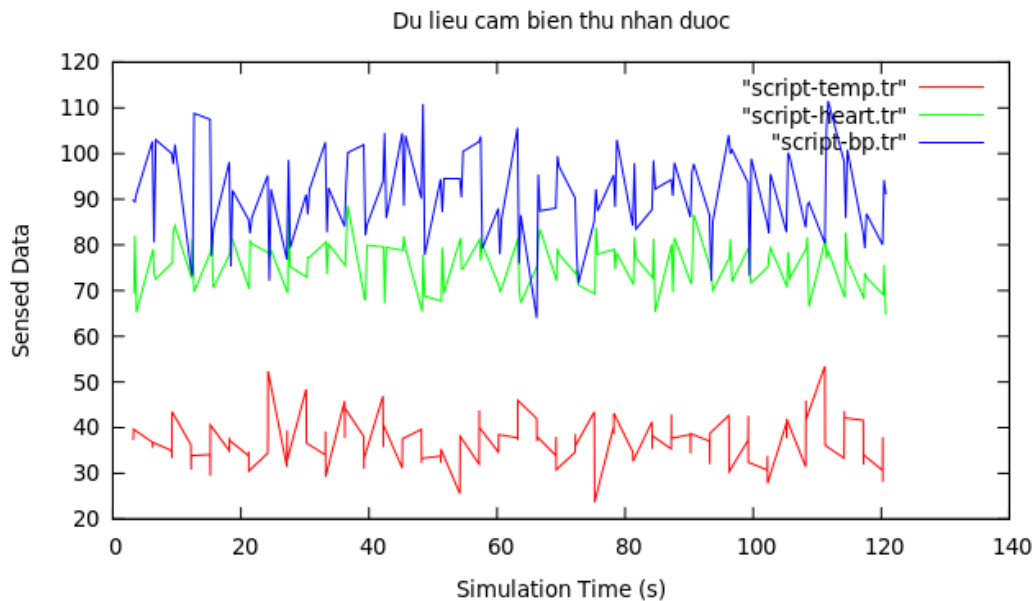
1 s -t 5.183757119 -Hs 3 -Hd -2 -Ni 3 -Nx 1.17 -Ny 1.33 -Nz 0.00 -Ne 9.999837 -
  Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.2020 -Id 1.2020 -It udp -Il 32 -
  If 0 -Ii 0 -Iv 32
2 r -t 5.183757119 -Hs 3 -Hd -2 -Ni 3 -Nx 1.17 -Ny 1.33 -Nz 0.00 -Ne 9.999837 -
  Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.2020 -Id 1.2020 -It udp -Il 32 -
  If 0 -Ii 0 -Iv 32
3 s -t 5.183757119 -Hs 3 -Hd -2 -Ni 3 -Nx 1.17 -Ny 1.33 -Nz 0.00 -Ne 9.999837 -
  Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.2020 -Id 1.2020 -It udp -Il 32 -
  If 0 -Ii 0 -Iv 32
4 N -t 5.184272 -n 9 -e 9.999821
5 N -t 5.184272 -n 10 -e 9.999821
6 N -t 5.184272 -n 2 -e 9.994982
7 N -t 5.184272 -n 8 -e 9.999821
8 N -t 5.184272 -n 4 -e 9.999821
9 N -t 5.184272 -n 6 -e 9.999821
10 N -t 5.184272 -n 7 -e 9.999821
11 N -t 5.184272 -n 5 -e 9.999821
12 N -t 5.184272 -n 0 -e 99.999656
13 N -t 5.184272 -n 1 -e 99.999656
14 N -t 5.185575 -n 3 -e 9.999804
15 N -t 5.185575 -n 10 -e 9.999813
16 N -t 5.185575 -n 8 -e 9.999813
17 N -t 5.185575 -n 9 -e 9.999813
18 N -t 5.185575 -n 2 -e 9.994974
19 N -t 5.185575 -n 7 -e 9.999813
20 N -t 5.185575 -n 6 -e 9.999813
21 N -t 5.185575 -n 4 -e 9.999813

```

Hình 4-6 Cấu trúc của tập vết

Các dữ liệu cảm biến được gửi về node Sink trong quá trình mô phỏng sẽ được hiển thị trực tiếp lên Terminal như trên Hình 4-5 Tập vết mô phỏng cảm biến gửi dữ

liệu, dữ liệu đó cũng có thể được ghi vào file dạng text để xử lý và vẽ thành đồ thị từ file text đó. Kết quả ta được các đồ thị biểu diễn dữ liệu cảm biến thu nhận được như sau.



Hình 4-7 Đồ thị thể hiện các dữ liệu cảm biến được theo kịch bản 1

Nhìn trên đồ thị Hình 4-7 Đồ thị thể hiện các dữ liệu cảm biến được theo kịch bản 1 ta có thể nhận thấy được các dữ liệu cảm biến đều có cập nhật đầy đủ. Sau khi chạy các kịch bản có sự thay đổi độ lệch chuẩn. Tín hiệu thu được cũng thể hiện sự thăng giáng khác nhau.

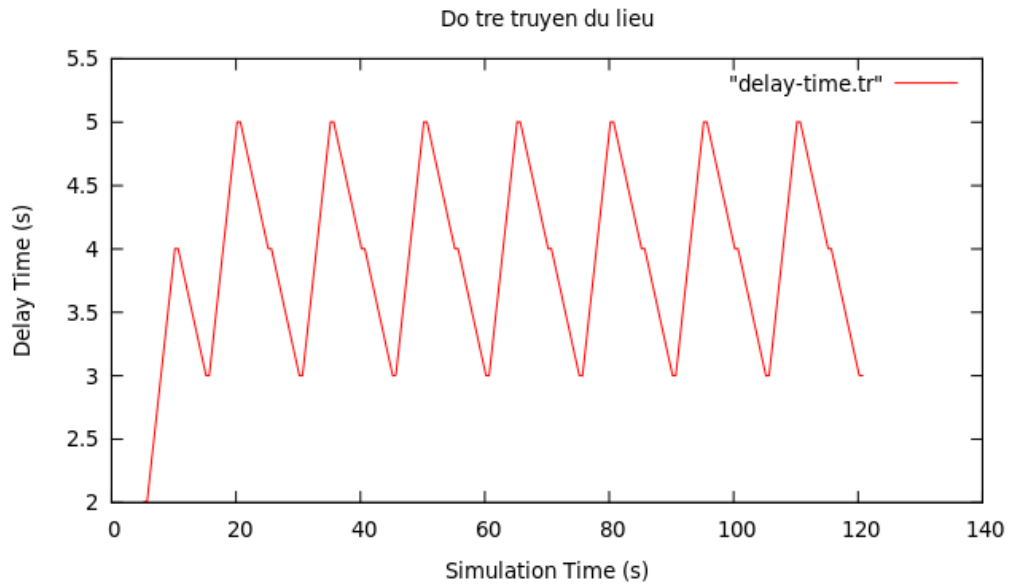
Tôi sử dụng 02 perl scripts để tính toán và trích xuất dữ liệu vẽ đồ thị cho 2 độ đo là độ trễ và mức độ suy hao năng lượng. Đối với độ đo thời gian trễ trung bình. Kết quả được thể hiện trong Hình 4-8 Thời gian trễ truyền tin kịch bản 1 và Hình 4-9 Biểu đồ thể hiện độ trễ truyền dữ liệu.

```

huytran@huytran-virtual-machine: ~/Desktop/kichban1
huytran@huytran-virtual-machine:~/Desktop/kichban1$ perl xulyfile.pl script1-output.tr
Avg Delay: 4.10445127350427 (s)
huytran@huytran-virtual-machine:~/Desktop/kichban1$

```

Hình 4-8 Thời gian trễ truyền tin kịch bản 1



Hình 4-9 Biểu đồ thể hiện độ trễ truyền dữ liệu kịch bản 1

Đối với độ đo sự suy hao năng lượng trong thời gian mô phỏng. Kết quả được thể hiện ở Hình 4-10 Độ suy hao năng lượng của nút cảm biến.

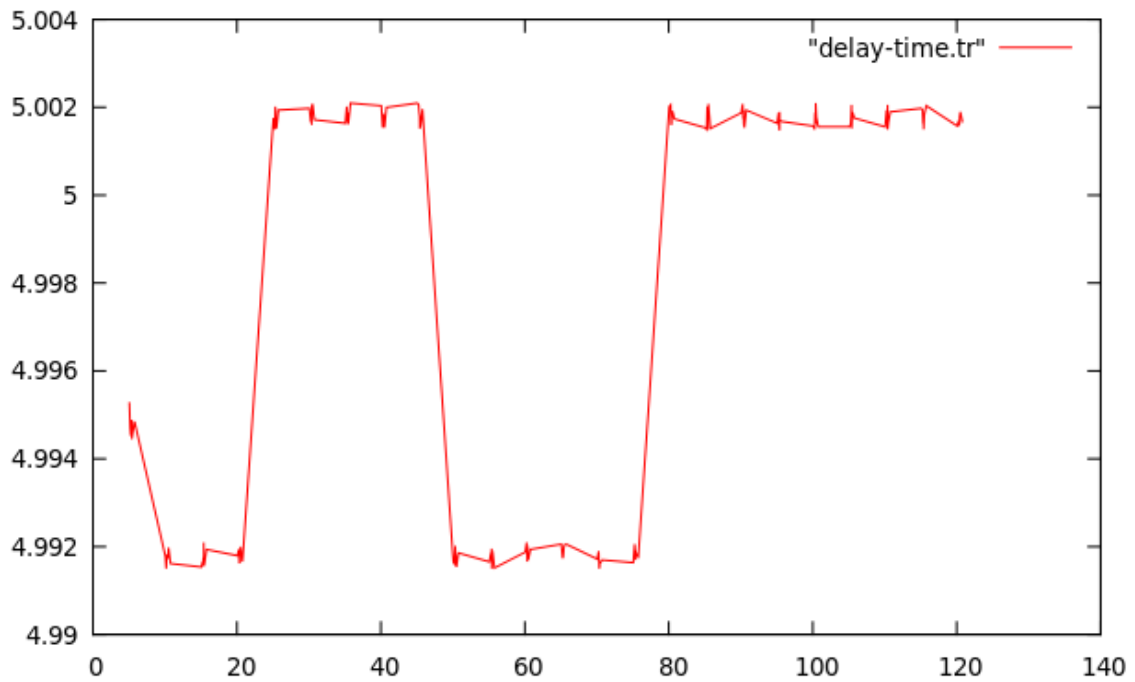
```
huytran@huytran-virtual-machine: ~/Desktop/kichban1
huytran@huytran-virtual-machine:~/Desktop/kichban1$ perl energy.pl script-1.tr 1
Max: 100 , Min: 99.808200
Do suy hao pin của node 1 la: 0.1918000000000001 %
huytran@huytran-virtual-machine:~/Desktop/kichban1$
```

Hình 4-10 Độ suy hao năng lượng của nút cảm biến

b, Kết quả mô phỏng kịch bản 2

```
huytran@huytran-virtual-machine: ~/Desktop/kichban2
huytran@huytran-virtual-machine:~/Desktop/kichban1$ cd ../kichban2
huytran@huytran-virtual-machine:~/Desktop/kichban2$ perl xulyfile.pl script-2-ou
tput.tr
Avg Delay: 4.99764491794841
huytran@huytran-virtual-machine:~/Desktop/kichban2$
```

Hình 4-11 Thời gian trễ truyền tin kịch bản 2



Hình 4-12 Thời gian trễ truyền dữ liệu, kịch bản 2

Mức suy hao năng lượng của nút cảm biến trong kịch bản 2 được thể hiện ở Hình 4-13 Mức suy hao năng lượng của kịch bản 2.. Dễ nhận thấy với việc sử dụng phương thức cảm biến liên tục với khoảng cảm biến 0.01s trong thời gian 120s. Nút cảm biến đã tiêu thụ gần 20% điện năng. Tức là gấp 100 lần so với cảm biến định kỳ 3.0s ở kịch bản 1.

```

huytran@huytran-virtual-machine: ~/Desktop/kichban2
huytran@huytran-virtual-machine:~/Desktop/kichban2$ perl energy.pl script-1.tr 5
Max: 10 , Min: 8.034321
Do suy hao pin của node 5 là: 19.65679 %
huytran@huytran-virtual-machine:~/Desktop/kichban2$

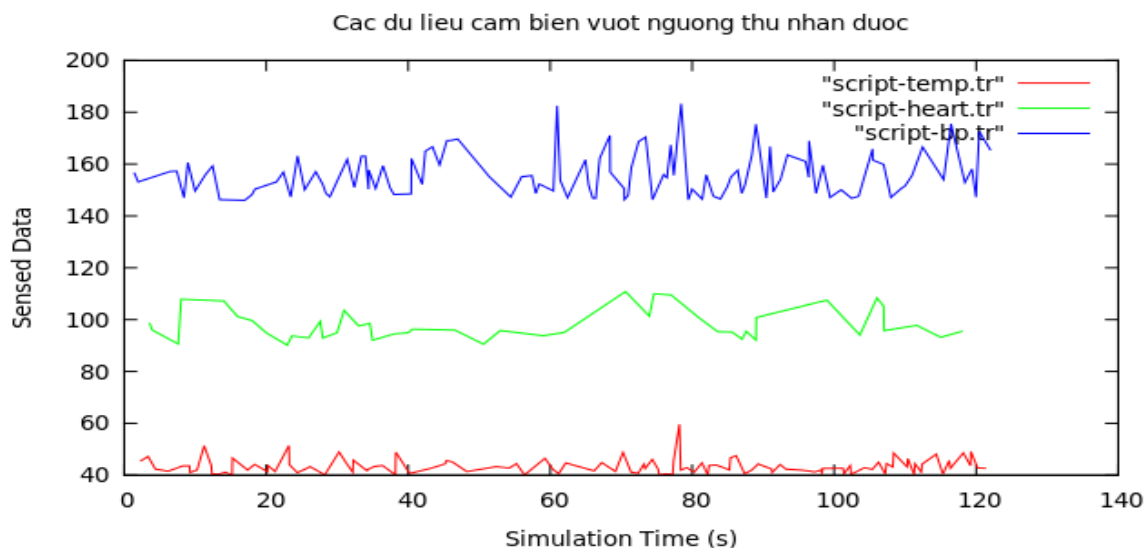
```

Hình 4-13 Mức suy hao năng lượng của kịch bản 2.

c, Kết quả mô phỏng kịch bản 3

Trong kịch bản này hệ thống sẽ kiểm tra liên tục các số liệu cảm biến thu nhận được và so sánh với ngưỡng đưa vào từ kịch bản. Chỉ những số liệu nào có giá trị vượt ngưỡng thì mới có cảnh báo gửi về Access Point. Do vậy số lượng gói tin gửi đến Access Point giảm đi đáng kể. Điều này giúp giảm độ trễ truyền tin và mức tiêu hao năng lượng của các nút mạng.

Trường hợp này tôi chỉ kết xuất và vẽ đồ thị biểu diễn những giá trị cảm biến vượt ngưỡng.



Hình 4-14 Các dữ liệu cảm biến vượt ngưỡng theo kịch bản 3

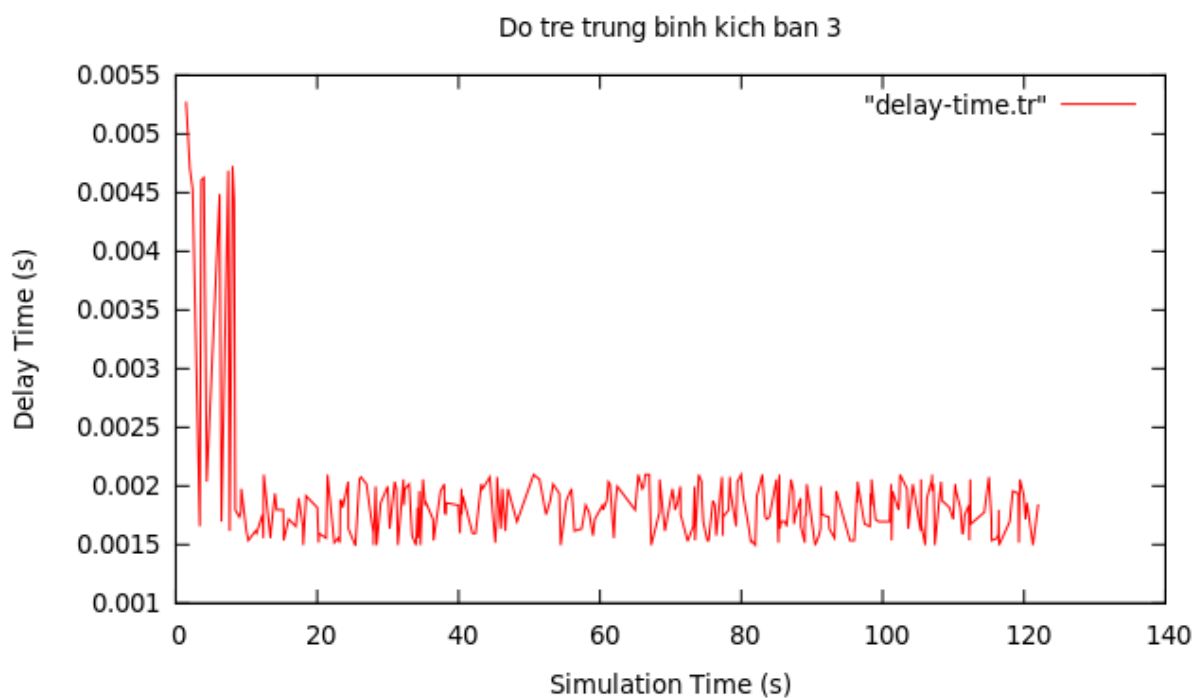
Sử dụng các perl script như trên ta có được các kết quả về độ trễ và mức tiêu hao năng lượng như sau (Hình 4-14).

```

huytran@huytran-virtual-machine: ~/Desktop/kichban3
huytran@huytran-virtual-machine:~/Desktop/kichban3$ perl xulyfile.pl script-3-output.tr
Sum Delay= 0.522731 , Num = 277 , Avg Delay: 0.0018871155234657
huytran@huytran-virtual-machine:~/Desktop/kichban3$

```

Hình 4-15 Độ trễ truyền tin theo kịch bản 3



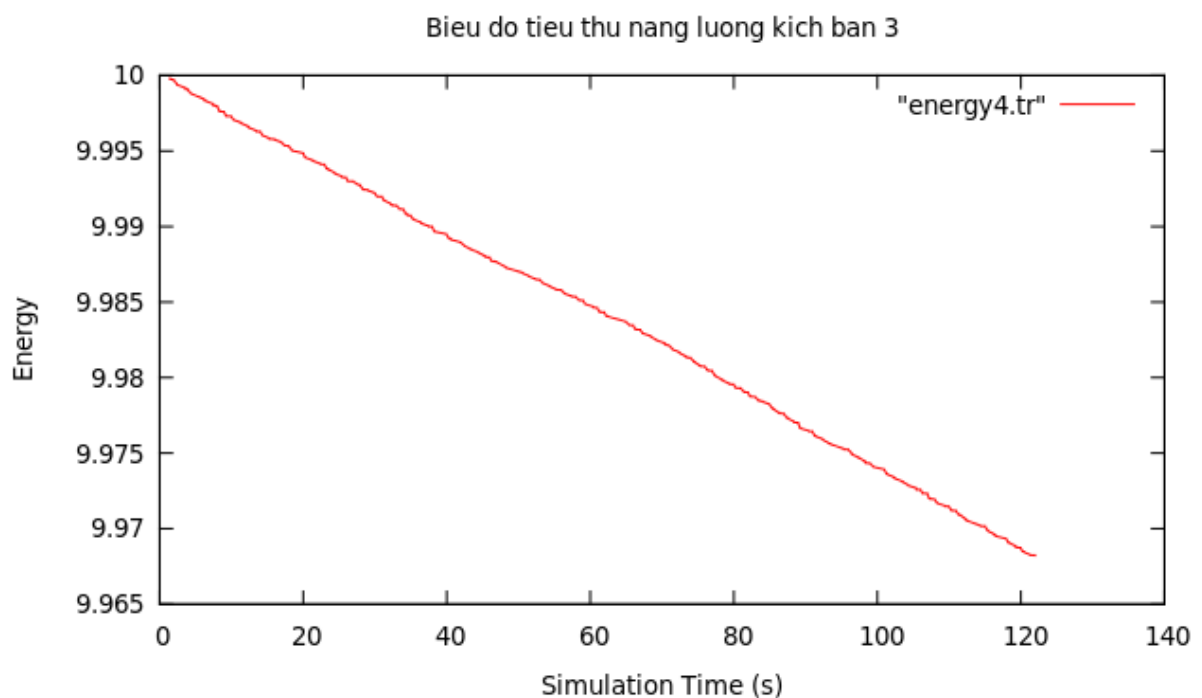
Hình 4-16 Biểu đồ thể hiện độ trễ truyền dữ liệu kịch bản 3

```

huytran@huytran-virtual-machine: ~/Desktop/kichban3
huytran@huytran-virtual-machine:~/Desktop/kichban3$ perl energy.pl script-3.tr 4
Max: 10 , Min: 9.968211
Do suy hao pin của node 4 là: 0.317889999999998 %
huytran@huytran-virtual-machine:~/Desktop/kichban3$ perl energy.pl script-3.tr 1
Max: 100 , Min: 99.756960
Do suy hao pin của node 1 là: 0.2430399999999994 %
huytran@huytran-virtual-machine:~/Desktop/kichban3$

```

Hình 4-17 Độ suy hao năng lượng của các nút mạng trong kịch bản 3



Hình 4-18 Biểu đồ sự suy hao năng lượng của nút mạng trong kịch bản 3

4.4 Tổng kết việc mô phỏng

Nhìn vào kết quả dữ liệu của tệp vết và đồ thị nhận được tôi đưa ra các nhận xét sau:

1/ Các thông số cần được theo dõi đều có báo cáo cập nhật đầy đủ. Những sự kiện thăng giáng vượt ngưỡng đều được báo cáo về với độ trễ nhỏ.

2/ Trong trường hợp sử dụng cảm biến ở chế độ liên tục, với số lượng 12 node cảm biến đã gửi khoảng 42000 bản tin cập nhật đến nút sink cho mỗi loại cảm biến. Tuy nhiên độ trễ truyền tin vẫn nằm trong mức chấp nhận được ở khoảng 4.5s. Như vậy theo quan điểm cá nhân, tôi cho là hệ thống đã đáp ứng được nhu cầu trong mô hình giám sát sức khỏe với số lượng 09 nút cảm biến.

3/ Phương thức mô phỏng hướng sự kiện đã cho thấy sự tối ưu trong việc sử dụng WSN giám sát sức khỏe cho con người. Với điều kiện lí tưởng độ trễ truyền tin trong khoảng 2ms, mức tiêu thụ năng lượng chỉ khoảng 0.3% dung lượng pin / 1 phút cảm biến. WSN với phương thức cảm biến hướng sự kiện rất thích hợp cho việc theo dõi, giám sát và cảnh báo kịp thời các biến động về sức khỏe của con người, vừa tạo sự thuận tiện, thoải mái.

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TRONG TƯƠNG LAI

5.1 Kết luận

Về kiến thức chung, tôi đã đạt được các kết quả như sau:

- Nghiên cứu tổng quan về WSN, các đặc điểm của WSN và ứng dụng WSN trong thực tế.
- Nghiên cứu các chuẩn truyền thông không dây như 802.11, 802.15.4 và ứng dụng các chuẩn này trong WSN.
- Tìm hiểu các hệ thống WSN phục vụ việc chăm sóc sức khỏe trên thực tế.
- Phân tích, thiết kế hệ thống WSN phục vụ giám sát sức khỏe. Đánh giá các yếu tố ảnh hưởng đến hiệu năng của WSN trong mô hình giám sát sức khỏe.

Về thực nghiệm, tôi đã đạt được các kết quả như sau:

- Nghiên cứu, tìm hiểu, cài đặt các công cụ mô phỏng. Cài đặt bộ mô phỏng ns-2 và mô đun manasim để mô phỏng hệ thống WSN.
- Nghiên cứu nguồn sinh lưu lượng và phân bố xác suất của nguồn sinh lưu lượng. Thiết kế hàm mô phỏng nút cảm biến sử dụng nguồn sinh lưu lượng đã phân tích.
- Tiến hành thử nghiệm các kịch bản mô phỏng. Viết các Script phân tích hiệu suất của hệ thống mạng và đưa ra kết luận.

5.2 Hướng nghiên cứu trong tương lai

Có thể triển khai các mô hình WSN trong thực tế dựa trên những kết quả đánh giá, phân tích hiệu suất để đưa ra các cải tiến. Bộ công cụ mô phỏng có thể được sử dụng để tiếp tục nghiên cứu, mô phỏng về mạng cảm biến không dây.

Xây dựng thêm các hàm mô phỏng nút cảm biến của nhiều loại tham số sức khỏe khác. Xây dựng thêm các kịch bản mô phỏng và phân tích hiệu suất. Nghiên cứu các vấn đề an toàn bảo mật

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

- [16] GS. Đỗ Đức Thái, GS. Nguyễn Tiến Dũng, Nhập môn hiện đại XÁC SUẤT và THỐNG KÊ, 2015: TỦ SÁCH SPUTNIK.
- [21] PGS. TS. Nguyễn Đình Việt, Bài Giảng: Mạng và Truyền số liệu nâng cao, 2010.
- [22] PGS. TS. Nguyễn Đình Việt, Bài giảng: Đánh giá hiệu năng mạng, 2016.

TIẾNG ANH

- [1] I. B. Qinghua Wang, "Wireless Sensor Networks: An Introduction," in *Wireless Sensor Networks: Application - Centric Design*, Yen Kheng Tan (Ed.), InTech, DOI: 10.5772/13225., 2010, p. 1.
- [2] P. P. Paolo Baronti, "WSN Applications," *Wireless sensor networks: A survey on the state of the art and the 802.15.4 and Zigbee standards*, p. 1657, 2006.
- [3] W. Stalling, "WiFi and the IEEE 802.11 Wireless Standard," in *Wireless Communications and Networks - Second Edition*, 2005, p. 421.
- [4] K. W. R. James F. Kurose, "Personal Area Network: Bluetooth and Zigbee," in *Computer Networking: A top-down approach - Six Edition*, 2013, p. 544.
- [5] I. S. f. I. Technology, "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," in *IEEE Standard for Local and metropolitan area networks*, 2011.
- [6] W. Z. Lincong Zhang, *Network Design and Performance Analysis of Wearable Wireless Sensor*, p. 411, 2015.
- [7] P. S. Pandia, "Wireless Sensor Network for Wearable Physiological Monitoring," *JOURNAL OF NETWORKS.*, Vols. 3, No 5, p. 21, 2008.
- [8] B.-r. C. K. L. Victor Shnayder, "Sensor Networks for Medical Care," *Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University*, 2005.
- [9] D. K. S. Shio Kumar Singh M P Singh, "Routing Protocols in Wireless Sensor Networks –A Survey," *International Journal of Computer Science & Engineering Survey (IJCSSES)*, Vols. 1, No.2, p. 63, 2010.
- [10] Joseph Kabara, Maria Calle, "MAC Protocols Used by Wireless Sensor Networks and a General Method of Performance Evaluation," *International Journal of*

Distributed Sensor Networks, vol. 8, Jan 2012.

- [11] J. H. D. E. Wei Ye, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE Infocom New York*, 2002.
- [12] S. L. a. C. S. Raghavendra, PEGASIS: Power-Efficient GATHERing in Sensor Information Systems, Computer Systems Research Department- The Aerospace Corporation.
- [13] H. R. N. Media Aminian, "A Hospital Healthcare Monitoring System Using Wireless Sensor Networks," *Health & Medical Informatics*, 2013.
- [14] N. G. B. F. I. Alexandros Pantelopoulos, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," in *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, IEEE, 2010.
- [15] L. B. R. Rodolfo Miranda Pereira, "MannaSim: A NS-2 extension to simulate Wireless Sensor Network," *ICN 2015 : The Fourteenth International Conference on Networks*, 2015.
- [17] P. R. Yechiam Ostchega and M. M. Kathryn S. Porter, "Resting Pulse Rate Reference Data for Children, Adolescents, and Adults: United States, 1999–2008," U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES, 08/2011.
- [18] D. Jacqueline D. Wright and M. Jeffery P. Hughes, "Mean Systolic and Diastolic Blood Pressure in Adults Aged 18 and Over in the United States, 2001–2008," U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES , 2011.
- [19] H. M. L. a. S. M. C. Yook Chin Chia, "Long-Term Visit-to-Visit Blood Pressure Variability and Renal Function Decline in Patients With Hypertension Over 15 Years," *Journal of the American Heart Association.*, 2016.
- [20] E. b. G. Elert, "Temperature of a Healthy Human (Body Temperature)," in *The Physics Factbook*.

PHỤ LỤC

1. Wsn-healthcare

```
#=====

#Script-1

#Number node:09 common node - 1 sink node - 1 AP

#Type of sensor: Temperature, Heart rate, Blood Presure

#Sensor mode: 0

#Time: 120s

#=====

# Procedure to create a common node application

#=====

proc      create_common_app      {destination_id      disseminating_type
disseminating_interval} {

    set app_ [new Application/SensorBaseApp/CommonNodeApp]

    $app_ set destination_id_ $destination_id

    $app_ set disseminating_type_ $disseminating_type

    $app_ set disseminating_interval_ $disseminating_interval

    return $app_

}

# =====

# Procedure to create a cluster head node application

# =====

proc      create_cluster_head_app      {destination_id      disseminating_type
disseminating_interval} {

    set app_ [new Application/SensorBaseApp/ClusterHeadApp]

    $app_ set destination_id_ $destination_id

    $app_ set disseminating_type_ $disseminating_type

    $app_ set disseminating_interval_ $disseminating_interval

    return $app_

}
```

```

}

# =====

# Procedure to create a access point node application.
# =====

proc create_access_point_app {destination_id} {
    set app_ [new Application/AccessPointApp]
    $app_ set destination_id_ $destination_id
    return $app_
}

# =====

# Procedure to create a Temperature Data Generator
# =====

proc create_temp_data_generator {sensing_interval sensing_type
    avg_measure std_deviation} {
    set temp_gen_ [new DataGenerator/TemperatureDataGenerator]
    $temp_gen_ set sensing_interval_ $sensing_interval
    $temp_gen_ set sensing_type_ $sensing_type
    $temp_gen_ set avg_measure $avg_measure
    $temp_gen_ set std_deviation $std_deviation
    return $temp_gen_
}

# =====

# Procedure to create a Heart Rate (HR) Generator
# =====

proc create_heart_data_generator {sensing_interval sensing_type
    avg_measure std_deviation} {
    set heart_gen_ [new DataGenerator/HeartRateDataGenerator]
    $heart_gen_ set sensing_interval_ $sensing_interval
    $heart_gen_ set sensing_type_ $sensing_type
    $heart_gen_ set avg_measure $avg_measure

```

```

    $heart_gen_ set std_deviation $std_deviation

    return $heart_gen_
}

# =====

# Procedure to create a Blood Pressure (BP) Data Generator
# =====

proc   create_carbon_data_generator   {sensing_interval   sensing_type
avg_measure std_deviation} {

    set carbon_gen_ [new DataGenerator/CarbonMonoxideDataGenerator]

    $carbon_gen_ set sensing_interval_ $sensing_interval

    $carbon_gen_ set sensing_type_ $sensing_type

    $carbon_gen_ set avg_measure $avg_measure

    $carbon_gen_ set std_deviation $std_deviation

    return $carbon_gen_
}

# =====

# Antenna Settings
# =====

Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# =====

# Wireless Phy Settings
# =====

Phy/WirelessPhy set CPTresh_ 10.0
Phy/WirelessPhy set CSTresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6

```

```

Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0
set contador_nodos 0

# =====
# Simulation parameters
# =====

set val(pt_common) 8.564879510890936E-4
set val(pt_cluster_head) 0.0
set val(chan) Channel/WirelessChannel ;# channel
set val(prop) Propagation/TwoRayGround ;# propagation
set val(netif) Phy/WirelessPhy; # phy
set val(mac) Mac/802_11 ; # mac
set val(ifq) Queue/DropTail/PriQueue ; # queue
set val(ll) LL ; # link layer
set val(ant) Antenna/OmniAntenna ; # antenna
set val(ifqlen) 200 ; # queue length
set val(rp) DumbAgent ; # routing protocol
set val(en) EnergyModel/Battery ; # energy model
set val(nn) 12 ; # number of nodes
set val(n_pas) 1 ; # number os access points
set val(n_sinks) 1 ; # number of sink
set val(n_cluster) 0 ; # number of cluster heads
set val(n_common) 11 ; # number of common nodes
set val(x) 2.0 ; # x lenght of scenario
set val(y) 2.0 ; # y lenght of scenario
set val(disseminating_type) 0 ; # common node disseminating type
set val(ch_disseminating_type) 0 ; # cluster heard disseminating
type

```



```

set val(disseminating_interval) 5.0 ; # common node disseminating
interval

set val(cluster_head_disseminating_interval) 0.0 ; # cluster head
disseminating interval

set val(start) 0.0 ; # simulation start time

set val(stop) 122.0 ; # simulation stop time

set val(father_addr) 1 ; # sink address

set val(port) 2020 ; # default port

# =====

# Global variables

# =====

set ns_ [new Simulator]

set traceFile [open script-1.tr w]

$ns_ trace-all $traceFile

$ns_ use-newtrace

set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

set rng [new RNG]

$rng seed 0

# =====

# Procedure to create a common Temp node

# =====

proc create_common_temp_node {} {
global val ns_ node_ topo udp_ app_ gen_ contador_nodos rng
Phy/WirelessPhy set Pt_ $val(pt_common)

    $ns_ node-config -sensorNode ON \
    -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \

```

```

-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-energyModel $val(en) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.024 \
-txPower 0.036 \
-initialEnergy 10.0 \
-movementTrace OFF

set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0

set x [$rng uniform 0.0 $val(x)]
set y [$rng uniform 0.0 $val(y)]
$node_($contador_nodos) set X_ $x
$node_($contador_nodos) set Y_ $y
$node_($contador_nodos) set Z_ 0.0

set interval [$rng uniform 0.0 1.0]

Node/MobileNode/SensorNode set sensingPower_ 0.015
Node/MobileNode/SensorNode set processingPower 0.024
Node/MobileNode/SensorNode set instructionsPerSecond_ 8000000

Phy/WirelessPhy set bandwidth_ 288000.0

set udp_($contador_nodos) [new Agent/UDP]

set distance 10000000

set initial [expr $val(n_pas) + $val(n_sinks)]

```

```

for {set j $initial} {$j < [expr $initial + $val(n_cluster)]} {incr
j} {

    set x_father [$node_($j) set X_]
    set y_father [$node_($j) set Y_]
    set x_son [$node_($contador_nodos) set X_]
    set y_son [$node_($contador_nodos) set Y_]
    set x_temp [expr pow([expr $x_father-$x_son],2)]
    set y_temp [expr pow([expr $y_father-$y_son],2)]
    set temp_distance [expr sqrt([expr $x_temp + $y_temp])]
    if {$temp_distance < $distance} {
        set distance $temp_distance
        set val(father_addr) [$node_($j) node-addr]
    }
}

set app_($contador_nodos) [create_common_app $val(father_addr)
$val(disseminating_type) $val(disseminating_interval)]
$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)
$node_($contador_nodos) add-app $app_($contador_nodos)

set processing_($contador_nodos) [new Processing/AggregateProcessing]

$app_($contador_nodos) node $node_($contador_nodos)
$app_($contador_nodos) attach-agent $udp_($contador_nodos)
$app_($contador_nodos) attach-processing $processing_($contador_nodos)
$processing_($contador_nodos) node $node_($contador_nodos)

$ns_ at [expr $val(start) + 0 + $interval] "$app_($contador_nodos)
start"

$ns_ at $val(stop) "$app_($contador_nodos) stop"

set gen_($contador_nodos) [create_temp_data_generator 3.0 0 37.0
5.0]

$app_($contador_nodos) attach_data_generator $gen_($contador_nodos)

incr contador_nodos

```

```

}

# =====

# Procedure to create a common HR node

# =====

proc create_heart_rate_node {} {
    global val ns_ node_ topo udp_ app_ gen_ contador_nodos rng

Phy/WirelessPhy set Pt_ $val(pt_common)

    $ns_ node-config -sensorNode ON \
    -adhocRouting $val(rp) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -energyModel $val(en) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -rxPower 0.024 \
    -txPower 0.036 \
    -initialEnergy 10.0 \
    -movementTrace OFF

    set node_($contador_nodos) [$ns_ node]
    $node_($contador_nodos) random-motion 0
    set x [$rng uniform 0.0 $val(x)]
    set y [$rng uniform 0.0 $val(y)]
    $node_($contador_nodos) set X_ $x

```

```

$node_($contador_nodos) set Y_ $y
$node_($contador_nodos) set Z_ 0.0
set interval [$rng uniform 0.0 1.0]

Node/MobileNode/SensorNode set sensingPower_ 0.015
Node/MobileNode/SensorNode set processingPower 0.024
Node/MobileNode/SensorNode set instructionsPerSecond_ 8000000
Phy/WirelessPhy set bandwidth_ 288000.0
set udp_($contador_nodos) [new Agent/UDP]
set distance 10000000
set initial [expr $val(n_pas) + $val(n_sinks)]
for {set j $initial} {$j < [expr $initial + $val(n_cluster)]} {incr j} {
    set x_father [$node_($j) set X_]
    set y_father [$node_($j) set Y_]
    set x_son [$node_($contador_nodos) set X_]
    set y_son [$node_($contador_nodos) set Y_]
    set x_temp [expr pow([expr $x_father-$x_son],2)]
    set y_temp [expr pow([expr $y_father-$y_son],2)]
    set temp_distance [expr sqrt([expr $x_temp + $y_temp])]
    if {$temp_distance < $distance} {
        set distance $temp_distance
        set val(father_addr) [$node_($j) node-addr]
    }
}

set app_($contador_nodos) [create_common_app $val(father_addr)
$val(disseminating_type) $val(disseminating_interval)]
$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)
$node_($contador_nodos) add-app $app_($contador_nodos)

set processing_($contador_nodos) [new
Processing/AggregateProcessing]

```

```

$app_($contador_nodos) node $node_($contador_nodos)

$app_($contador_nodos) attach-agent $udp_($contador_nodos)

$app_($contador_nodos)                                attach-processing
$processing_($contador_nodos)

$processing_($contador_nodos) node $node_($contador_nodos)

$ns_ at [expr $val(start) + 0 + $interval] "$app_($contador_nodos)
start"

$ns_ at $val(stop) "$app_($contador_nodos) stop"

set gen_($contador_nodos) [create_heart_data_generator 3.0 0 75.0
5.0]

$app_($contador_nodos) attach_data_generator $gen_($contador_nodos)

incr contador_nodos
}

# =====

# Procedure to create a common blood pressure node

# =====

proc create_blood_pressure_node {} {
    global val ns_ node_ topo udp_ app_ gen_ contador_nodos rng
Phy/WirelessPhy set Pt_ $val(pt_common)

    $ns_ node-config -sensorNode ON \
    -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -energyModel $val(en) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \

```

```

-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.024 \
-txPower 0.036 \
-initialEnergy 10.0 \
-movementTrace OFF

set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0

set x [$rng uniform 0.0 $val(x)]
set y [$rng uniform 0.0 $val(y)]

$node_($contador_nodos) set X_ $x
$node_($contador_nodos) set Y_ $y
$node_($contador_nodos) set Z_ 0.0

set interval [$rng uniform 0.0 1.0]

Node/MobileNode/SensorNode set sensingPower_ 0.015
Node/MobileNode/SensorNode set processingPower 0.024
Node/MobileNode/SensorNode set instructionsPerSecond_ 8000000
Phy/WirelessPhy set bandwidth_ 288000.0

set udp_($contador_nodos) [new Agent/UDP]

set distance 10000000

set initial [expr $val(n_pas) + $val(n_sinks)]

for {set j $initial} {$j < [expr $initial + $val(n_cluster)]} {incr j} {

    set x_father [$node_($j) set X_]
    set y_father [$node_($j) set Y_]

    set x_son [$node_($contador_nodos) set X_]
    set y_son [$node_($contador_nodos) set Y_]

    set x_temp [expr pow([expr $x_father-$x_son],2)]
    set y_temp [expr pow([expr $y_father-$y_son],2)]

```

```

    set temp_distance [expr sqrt([expr $x_temp + $y_temp])]

    if {$temp_distance < $distance} {

        set distance $temp_distance

        set val(father_addr) [$node_($j) node-addr]

    }

}

set app_($contador_nodos) [create_common_app $val(father_addr)
$val(disseminating_type) $val(disseminating_interval)]

$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)

$node_($contador_nodos) add-app $app_($contador_nodos)

set processing_($contador_nodos) [new
Processing/AggregateProcessing]

$app_($contador_nodos) node $node_($contador_nodos)

$app_($contador_nodos) attach-agent $udp_($contador_nodos)

$app_($contador_nodos) attach-processing
$processing_($contador_nodos)

$processing_($contador_nodos) node $node_($contador_nodos)

$ns_ at [expr $val(start) + 0 + $interval] "$app_($contador_nodos)
start"

$ns_ at $val(stop) "$app_($contador_nodos) stop"

set gen_($contador_nodos) [create_carbon_data_generator 3.0 0 90.0
10.0]

$app_($contador_nodos) attach_data_generator $gen_($contador_nodos)

incr contador_nodos

}

# =====

# Procedure to create a cluster head node

# =====

proc create_cluster_head_node {} {

global val ns_ node_ topo contador_nodos rng

Phy/WirelessPhy set Pt_ $val(pt_cluster_head)

$ns_ node-config -sensorNode ON \

```



```

-adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-energyModel $val(en) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.0 \
-txPower 0.0 \
-initialEnergy 0.0 \
-movementTrace ON

set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0

set x [$rng uniform 0.0 $val(x)]
set y [$rng uniform 0.0 $val(y)]
$node_($contador_nodos) set X_ $x
$node_($contador_nodos) set Y_ $y
$node_($contador_nodos) set Z_ 0.0

set interval [$rng uniform 0.0 1.0]

Node/MobileNode/SensorNode set processingPower 0.0
Node/MobileNode/SensorNode set instructionsPerSecond_ 0
Phy/WirelessPhy set bandwidth_ 0.0

set udp_($contador_nodos) [new Agent/UDP]

```

```

set app_($contador_nodos) [create_cluster_head_app [$node_(1) node-
addr                                     $val(disseminating_type)
$val(cluster_head_disseminating_interval)]

$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)

$node_($contador_nodos) add-app $app_($contador_nodos)

set processing_($contador_nodos) [new ]

$app_($contador_nodos) node $node_($contador_nodos)

$app_($contador_nodos) attach-agent $udp_($contador_nodos)

$app_($contador_nodos)                                     attach-processing
$processing_($contador_nodos)

$processing_($contador_nodos) node $node_($contador_nodos)

$ns_ at [expr $val(start) + 1 + $interval] "$app_($contador_nodos)
start"

$ns_ at $val(stop) "$app_($contador_nodos) stop"

incr contador_nodos

}

# =====

# Procedure to create a sink node

# =====

proc create_sink {} {

    global ns_ val node_ sink_ contador_nodos topo

    Phy/WirelessPhy set Pt_ 0.2818

    $ns_ node-config -sensorNode ON \

    -adhocRouting $val(rp) \

    -llType $val(ll) \

    -macType $val(mac) \

    -ifqType $val(ifq) \

    -ifqLen $val(ifqlen) \

    -antType $val(ant) \

    -propType $val(prop) \

```

```

-energyModel $val(en) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.5 \
-txPower 0.5 \
-initialEnergy 100.0 \
-movementTrace OFF

set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0
set sink_(0) [new Agent/LossMonitor]
$node_($contador_nodos) attach $sink_(0) $val(port)
$node_($contador_nodos) set X_ 0.0
$node_($contador_nodos) set Y_ 0.0
$node_($contador_nodos) set Z_ 0.0
incr contador_nodos
}

# =====
# Procedure to create a access point node
# =====

proc create_access_point {} {
global ns_ val node_ app_ udp_ contador_nodos topo
Phy/WirelessPhy set Pt_ 0.2818
$ns_ node-config -sensorNode ON \
-adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \

```

```

-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-energyModel $val(en) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-rxPower 0.5 \
-txPower 0.5 \
-initialEnergy 100.0 \
-movementTrace OFF

set node_($contador_nodos) [$ns_ node]
$node_($contador_nodos) random-motion 0

set udp_($contador_nodos) [new Agent/UDP]

set app_($contador_nodos) [create_access_point_app [$node_(0) node-
addr]]

$node_($contador_nodos) attach $udp_($contador_nodos) $val(port)
$app_($contador_nodos) attach-agent $udp_($contador_nodos)

$node_($contador_nodos) set X_ 5.0
$node_($contador_nodos) set Y_ 5.0
$node_($contador_nodos) set Z_ 0.0

$ns_ at [expr $val(stop)+1] "$app_($contador_nodos) stop"

incr contador_nodos
}

# =====
# Procedures to control common node and cluster head node creation

```

```

# =====

create_sink

create_access_point

for {set j 0} {$j < $val(n_cluster)} {incr j} {

    create_cluster_head_node

}

#create_temperature_node

for {set i 2} {$i < 5} {incr i} {

    create_common_temp_node

}

#create_heart_rate_node

for {set i 5} {$i < 8} {incr i} {

    create_heart_rate_node

}

#create_blood_pressure_node

for {set i 8} {$i < $val(n_common)} {incr i} {

    create_blood_pressure_node

}

# =====

# Simulation

# =====

$ns_ at [expr $val(stop)+2.0] "finish"

$ns_ at [expr $val(stop)+2.0] "puts \"NS EXITING...\" ; $ns_ halt"

$ns_ at [expr $val(stop)+2.0] "$ns_ nam-end-wireless $val(stop)"

proc finish {} {

    global ns_ traceFile

    $ns_ flush-trace

    close $traceFile

}

puts "Starting Simulation..."

```

```
$ns_ run
```

2. Perl Script xulyfile.pl

```
#Type: Chuong trinh xu ly output wsn
```

```
$input = $ARGV[0];
```

```
my $output1 = 'script-temp.tr';
```

```
my $output2 = 'script-heart.tr';
```

```
my $output3 = 'script-bp.tr';
```

```
my $output4 = 'delay-time.tr';
```

```
$sum = 0;
```

```
$num = 0;
```

```
open (my $write1, '>', $output1);
```

```
open (my $write2, '>', $output2);
```

```
open (my $write3, '>', $output3);
```

```
open (my $write4, '>', $output4);
```

```
open (DATA,"<$input") || die "Can't open $input $!";
```

```
while (<DATA>) {
```

```
    @x = split(' ');
```

```
    if ($x[0] eq 'Node' && $x[3] eq 'Temperature')
```

```
        { print $write1 $x[10].' '.$x[6]."\n"; }
```

```
    if ($x[0] eq 'Node' && $x[3] eq 'HeartRate')
```

```
        { print $write2 $x[10].' '.$x[6]."\n"; }
```

```
    if ($x[0] eq 'Node' && $x[3] eq 'BloodPresure')
```

```
        { print $write3 $x[10].' '.$x[6]."\n"; }
```

```
    if ($x[0] eq 'Message')
```

```
    {
```

```
        $sum = $sum+$x[6];
```

```
        $num ++;
```

```
        print $write4 $x[10].' '.$x[6]."\n";
```

```
    }
```

```

}

$avg_delay = $sum / $num;

print ("Avg Delay: $avg_delay \n");

close DATA;

exit(0);

```

3. Energy.pl

```

#Tinh do suy hao nang luong

#Type: Chuong trinh xu ly output wsn

$input = $ARGV[0];

$node_id = $ARGV[1];

my $output1 = 'energy'.$node_id.'.tr';

if ($node_id eq 1)
{
    $max_energy = 100;
}

elsif ($node_id eq 0)
{
    $max_energy = 100;
}

else {
    $max_energy = 10;
}

$min_energy = $max_energy;

open (my $write, '>', $output1);

open (DATA,"<$input") || die "Can't open $input !";

while (<DATA>) {
    @x = split(' ');

    if ($x[0] eq 'N' && $x[4] eq $node_id)
    {

```

```

print $write $x[2].' '$x[6]."\n";

if ($x[6] < $min_energy)
{
    $min_energy = $x[6];
}

}

close DATA;

$energy = ($max_energy - $min_energy);
$suyhao = ($energy / $max_energy) * 100;
print ("Max: $max_energy , Min: $min_energy \n");
print ("Do suy hao pin cua node $node_id la: $suyhao %\n");
exit(0);

```