

## LỜI CAM ĐOAN

Tôi xin cam đoan tất cả các kết quả được trình bày trong luận án: ***“Nghiên cứu xây dựng thuật toán định tuyến trên cơ sở nhận thức về năng lượng cho mạng cảm biến không dây”*** là công trình nghiên cứu của riêng tôi, không sao chép từ bất kỳ một công trình nào khác.

Các số liệu trong luận án được sử dụng là trung thực, một phần đã được công bố trên các tạp chí khoa học chuyên ngành với sự đồng ý và cho phép của các đồng tác giả.

*Hà Nội, ngày 12 tháng 12 năm 2014*

Tác giả

**Vũ Chiến Thắng**

## LỜI CẢM ƠN

Tôi xin bày tỏ lòng biết ơn sâu sắc đến PGS.TS Nguyễn Chấn Hùng, Trung tâm Công nghệ xanh, Viện nghiên cứu Điện tử - Tin học - Tự động hóa và PGS.TS Lê Nhật Thăng, Bộ môn Kỹ thuật chuyển mạch, Khoa Viễn thông 1, Học viện Công nghệ Bưu chính Viễn thông đã tận tình hướng dẫn, tạo mọi điều kiện thuận lợi, giúp tôi thực hiện và hoàn thành luận án này.

Tôi xin trân trọng cảm ơn Viện trưởng Viện nghiên cứu Điện tử - Tin học - Tự động hóa, cùng các cán bộ trong Viện nghiên cứu Điện tử - Tin học - Tự động hóa đã tạo điều kiện cho tôi thực hiện bản luận án này.

Xin được gửi lời cảm ơn chân thành nhất tới Ban lãnh đạo Trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên, các anh chị, bạn bè và đồng nghiệp Khoa Công nghệ Điện tử và Truyền thông, Trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên đã chia sẻ và động viên giúp tôi vượt qua mọi khó khăn để hoàn thành tốt công việc nghiên cứu của mình.

Tôi biết ơn những người thân trong gia đình đã luôn bên tôi, quan tâm, động viên, tạo điều kiện thuận lợi nhất để tôi có thể hoàn thành bản luận án.

*Hà Nội, ngày 12 tháng 12 năm 2014*

Tác giả

**Vũ Chiến Thắng**

# MỤC LỤC

LỜI CAM ĐOAN .....	1
LỜI CẢM ƠN .....	2
MỤC LỤC.....	3
DANH MỤC CÁC TỪ VIẾT TẮT .....	7
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ.....	9
DANH MỤC CÁC BẢNG BIỂU .....	12
MỞ ĐẦU.....	13
1. Tính khoa học và tính cấp thiết của luận án .....	13
2. Đối tượng và phạm vi nghiên cứu.....	15
3. Mục tiêu của luận án.....	15
4. Phương pháp luận nghiên cứu .....	16
5. Nội dung của luận án.....	16
6. Đóng góp của luận án.....	17
Chương 1. BÀI TOÁN ĐỊNH TUYẾN TRONG MẠNG CẢM BIẾN	
KHÔNG DÂY .....	19
1.1. Vấn đề năng lượng tiêu thụ trong mạng cảm biến không dây .....	19
1.2. Tổng quan về vấn đề định tuyến trong mạng cảm biến không dây.....	20
1.2.1. Khái niệm và phân loại các giao thức định tuyến cho mạng cảm biến không dây.....	20
1.2.2. Thước đo định tuyến.....	21
1.2.3. Đặc điểm của mạng cảm biến không dây ảnh hưởng đến vấn đề định tuyến .....	22
1.2.4. Những thách thức đối với vấn đề định tuyến trong mạng cảm biến không dây.....	23
1.2.5. Một số phương pháp thiết kế giao thức định tuyến cho mạng cảm biến không dây.....	25
1.3. Một số giao thức định tuyến hiệu quả về mặt năng lượng cho mạng cảm biến không dây.....	29
1.3.1. Giao thức định tuyến trung tâm dữ liệu.....	29
1.3.2. Giao thức định tuyến dựa trên sự phân cụm.....	33
1.3.3. Giao thức định tuyến dựa trên vị trí.....	37
1.4. Giao thức cây thu thập dữ liệu CTP .....	38

1.4.1. Thước đo định tuyến được sử dụng trong giao thức CTP .....	39
1.4.2. Cấu trúc các bản tin trong giao thức CTP.....	40
1.4.3. Các thành phần chính của giao thức CTP.....	42
1.4.4. Điểm yếu của giao thức CTP .....	43
1.5. Bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng ....	44
1.6. Hiện trạng nghiên cứu và phương pháp tiếp cận bài toán định tuyến có sự nhận thức về năng lượng trong các nghiên cứu trước đây .....	48
1.6.1. Định tuyến với tổng năng lượng tối thiểu MTPR (Minimal Total Power Routing) .....	48
1.6.2. Định tuyến với chi phí nguồn pin nhỏ nhất (Minimum Battery Cost Routing) .....	48
1.6.3. Giao thức định tuyến nhận thức về năng lượng EAR (Energy Aware Routing) .....	49
1.6.4. Giao thức định tuyến E-Span (Energy-Aware Spanning Tree Algorithm) .....	49
1.6.5. Giao thức định tuyến có sự nhận thức về năng lượng và cân bằng tải ....	50
1.6.6. Giao thức định tuyến BRE (Bursty Routing Extensions).....	50
1.6.7. Giao thức định tuyến BCTP (Balanced Collection Tree Protocol) .....	50
1.6.8. Giao thức định tuyến ICTP (Improved Collection Tree Protocol).....	51
1.6.9. Giao thức định tuyến EQLR (Energy and Link Quality Based Routing Tree).....	52
1.6.10. Giao thức định tuyến ELR (Energy Aware and Link Quality Based Routing Protocol).....	52
1.6.11. Giao thức định tuyến EARBB (Energy Aware Routing Based on Beaconing) .....	53
1.7. Giải pháp tiếp cận bài toán trong luận án .....	54
1.8. Kết luận chương 1 .....	55
<b>Chương 2. GIAO THỨC CÂY THU THẬP DỮ LIỆU CÓ SỰ NHẬN THỨC VỀ NĂNG LƯỢNG .....</b>	<b>56</b>
2.1. Đề xuất mô hình toán học cho bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng.....	56
2.1.1. Mô hình kết nối giữa các nút mạng cảm biến không dây dựa trên lý thuyết đồ thị .....	56
2.1.2. Các vấn đề về nhiễu trong mạng cảm biến không dây .....	57

2.1.3. Mô hình bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng dựa trên lý thuyết đồ thị.....	59
2.2. Đề xuất giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng EACTP .....	61
2.2.1. Mục tiêu và những thách thức trong việc đề xuất giao thức EACTP.....	61
2.2.2. Những đề xuất cải tiến trong giao thức EACTP.....	62
2.3. Thực thi giao thức EACTP.....	71
2.3.1. Lựa chọn môi trường phát triển phần mềm.....	71
2.3.2. Thực thi giao thức EACTP trên hệ điều hành Contiki.....	73
2.4. Đánh giá giao thức EACTP dựa trên mô phỏng .....	81
2.4.1. Các tham số đánh giá.....	82
2.4.2. Mô hình đánh giá mô phỏng 1.....	83
2.4.3. Mô hình đánh giá mô phỏng 2.....	86
2.4.4. Nhận xét.....	88
2.5. Đánh giá giao thức EACTP dựa trên cơ sở phân tích lý thuyết.....	89
2.6. Kết luận chương 2 .....	95
Chương 3. TRIỂN KHAI ĐÁNH GIÁ THỰC NGHIỆM.....	96
3.1. Phần cứng cho mạng cảm biến không dây.....	96
3.1.1. Cấu trúc phần cứng của một nút mạng cảm biến không dây.....	96
3.1.2. Khảo sát một số nền tảng phần cứng cho mạng cảm biến không dây.....	99
3.2. Đề xuất cấu trúc phần cứng TUmote.....	101
3.2.1. Bộ vi điều khiển MSP430.....	101
3.2.2. Bộ thu phát vô tuyến CC2420EM .....	102
3.2.3. Cảm biến SHT11 .....	102
3.2.4. Bộ nhớ ngoài.....	103
3.2.5. Nguồn cung cấp .....	103
3.3. Các thông số kỹ thuật chính của phần cứng TUmote.....	103
3.4. Triển khai mô hình đánh giá thực nghiệm với giao thức EACTP sử dụng phần cứng TUmote .....	105
3.4.1. Mô hình đánh giá thực nghiệm 1.....	105
3.4.2. Mô hình đánh giá thực nghiệm 2.....	108
3.4.3. Nhận xét.....	110
3.5. Kết luận chương 3 .....	111
KẾT LUẬN.....	113

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ .....	115
TÀI LIỆU THAM KHẢO.....	116
PHỤ LỤC .....	121
Phụ lục 1. Bản thiết kế sơ đồ nguyên lý và sơ đồ mạch in phần cứng TUmote..	121
Phụ lục 2. Ngăn xếp truyền thông RIME trong hệ điều hành Contiki .....	124
Phụ lục 3. Giao thức ContikiMAC trong hệ điều hành Contiki .....	126
Phụ lục 4. Đánh giá mô phỏng với Cooja. ....	127
Phụ lục 5. Đánh giá thực nghiệm với TUmote.....	132

## DANH MỤC CÁC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Từ gốc</b>	<b>Nghĩa tiếng Việt</b>
ACK	Acknowledgement	Bản tin xác nhận
ADV	ADvertisement	Bản tin thông báo
APIs	Application Programming Interfaces	Các giao diện lập trình ứng dụng
ANR	Alive Node Ratio	Tỷ lệ nút sống
BCTP	Balanced Collection Tree Protocol	Giao thức cây thu thập dữ liệu cân bằng tải
BRE	Bursty Routing Extensions	Định tuyến bùng phát
CTP	Collection Tree Protocol	Giao thức cây thu thập dữ liệu
DDR	Data Delivery Ratio	Tỷ lệ chuyển phát dữ liệu
ETX	Expected Transmission	Số lần truyền kỳ vọng
EACTP	Energy Aware Collection Tree Protocol	Giao thức cây thu thập dữ liệu nhận thức về năng lượng
EI	Energy Indicator	Chỉ số năng lượng
EIB	Energy Indicator Balance	Mức cân bằng chỉ số năng lượng
ES	Energy State	Trạng thái năng lượng
EAR	Energy Aware Routing	Định tuyến nhận thức năng lượng
GAF	Geographic Adaptive Fidelity	Định tuyến thích ứng vị trí
GEAR	Geographic and Energy Aware Routing	Định tuyến nhận thức năng lượng và vị trí
HT	Hard Threshold	Ngưỡng cứng
ICTP	Improved Collection Tree Protocol	Giao thức cây thu thập dữ liệu cải tiến
LEACH	Low Energy Adaptive Clustering Hierarchy	Kiến trúc phân cụm thích ứng năng lượng thấp
LQI	Link Quality Indicator	Chỉ số chất lượng liên kết

MAC	Media Access Control	Điều khiển truy nhập kênh truyền
MANET	Mobile Ad-hoc Network	Mạng tùy biến di động
MECN	Minimum Energy Communication Network	Mạng truyền thông năng lượng thấp
PEGASIS	Power-Efficient Gathering in Sensor Information System	Hệ thống thu thập thông tin cảm biến hiệu quả năng lượng
PHY	Physical layer	Lớp vật lý
RAM	Random Access Memory	Bộ nhớ truy nhập ngẫu nhiên
ROM	Read Only Memory	Bộ nhớ chỉ đọc
SPIN	Sensor Protocols for Information via Negotiation	Các giao thức định tuyến thông qua thương lượng
SPIN-PP	Sensor Protocols for Information via Negotiation – Point to Point	Giao thức định tuyến thông qua thương lượng – Điểm tới Điểm
SPIN-EC	Sensor Protocols for Information via Negotiation – Energy Conservation	Giao thức định tuyến thông qua thương lượng – Bảo tồn năng lượng
SPIN-RL	Sensor Protocols for Information via Negotiation – Reliable	Giao thức định tuyến thông qua thương lượng – Tin cậy
SPIN-BC	Sensor Protocols for Information via Negotiation – BroadCast	Giao thức định tuyến thông qua thương lượng – Quảng bá
ST	Soft Threshold	Ngưỡng mềm
SINR	Signal-to-Interference Plus Noise	Tín hiệu trên nhiễu thêm tạp âm
TTL	Time To Live	Thời gian sống
TEEN	Threshold sensitive Energy Efficient sensor Network	Mạng cảm biến hiệu quả về năng lượng dựa trên các ngưỡng
UDG	Unit Disk Graph	Đồ thị vòng tròn đơn vị
UDI	UDG with Distance Interference	Mô hình UDG với nhiễu khoảng cách
WSN	Wireless Sensor Network	Mạng cảm biến không dây



## DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

<b>Hình 1.1:</b> Mô hình mạng cảm biến không dây với các nút cảm biến phân bố rải rác trong trường cảm biến. ....	20
<b>Hình 1.2:</b> Phân loại các giao thức định tuyến cho mạng cảm biến không dây. ....	21
<b>Hình 1.3:</b> Giao thức SPIN. ....	29
<b>Hình 1.4:</b> Hoạt động của giao thức Directed Diffusion. ....	31
<b>Hình 1.5:</b> Kiến trúc phân cấp dựa trên các cụm. ....	33
<b>Hình 1.6:</b> Cấu trúc chuỗi của PEGASIS. ....	34
<b>Hình 1.7:</b> Kiến trúc phân cấp trong giao thức TEEN. ....	36
<b>Hình 1.8:</b> Cấu trúc liên kết mạng được xây dựng theo giao thức CTP. ....	38
<b>Hình 1.9:</b> $ETX_{link}$ của một liên kết. ....	39
<b>Hình 1.10:</b> Cấu trúc bản tin dữ liệu. ....	40
<b>Hình 1.11:</b> Cấu trúc bản tin điều khiển. ....	41
<b>Hình 1.12:</b> Các thành phần chính của giao thức CTP. ....	42
<b>Hình 1.13:</b> Cấu trúc liên kết mạng được xét đến trong bài toán định tuyến EACTP. ....	45
<b>Hình 1.14:</b> Ví dụ minh họa cấu trúc hình học của bài toán định tuyến EACTP. ....	46
<b>Hình 2.1:</b> Mô hình UDG. ....	57
<b>Hình 2.2:</b> Mô hình UDI. ....	59
<b>Hình 2.3:</b> Mô hình bài toán định tuyến EACTP dựa trên lý thuyết đồ thị. ....	60
<b>Hình 2.4:</b> Cấu trúc khung bản tin điều khiển trong giao thức EACTP. ....	68
<b>Hình 2.5:</b> Cấu trúc cây định tuyến EACTP. ....	69
<b>Hình 2.6:</b> Các thành phần chính của giao thức EACTP. ....	74
<b>Hình 2.7:</b> Quá trình xử lý sự kiện lớp ứng dụng gửi một bản tin dữ liệu. ....	76
<b>Hình 2.8:</b> Quá trình xử lý sự kiện nút nhận một bản tin dữ liệu. ....	77
<b>Hình 2.9:</b> Quá trình xử lý sự kiện nút nhận một bản tin ACK/Timeout. ....	78
<b>Hình 2.10:</b> Quá trình xử lý sự kiện nút nhận một bản tin điều khiển. ....	79
<b>Hình 2.11:</b> Lưu đồ thuật toán Thêm/Cập nhật (rtmetric, ES) của nút lân cận. ....	80
<b>Hình 2.12:</b> Khoảng thời gian gửi bản tin điều khiển định kỳ. ....	81
<b>Hình 2.13:</b> Mô hình mô phỏng một cụm gồm 30 nút. ....	84
<b>Hình 2.14:</b> So sánh tỷ lệ các nút còn sống trong mạng. ....	85
<b>Hình 2.15:</b> So sánh tỷ lệ chuyển phát bản tin dữ liệu. ....	85

<b>Hình 2.16:</b> So sánh sự cân bằng năng lượng giữa các nút mạng. ....	85
<b>Hình 2.17:</b> Mô hình mô phỏng một cụm nhỏ gồm 10 nút. ....	86
<b>Hình 2.18:</b> So sánh tỷ lệ các nút còn sống trong mạng. ....	87
<b>Hình 2.19:</b> So sánh tỷ lệ chuyển phát bản tin dữ liệu. ....	87
<b>Hình 2.20:</b> So sánh sự cân bằng năng lượng giữa các nút mạng. ....	88
<b>Hình 3.1:</b> Các thành phần chính trong kiến trúc phần cứng của một nút cảm biến không dây. ....	96
<b>Hình 3.2:</b> Bo mạch MicaZ của hãng Crossbow Technology. ....	98
<b>Hình 3.3:</b> Mốc thời gian ra đời các nền tảng phần cứng . ....	99
<b>Hình 3.4:</b> Sơ đồ khối cấu trúc phần cứng TUmote. ....	102
<b>Hình 3.5:</b> Hình ảnh sản phẩm phần cứng TUmote. ....	103
<b>Hình 3.6:</b> Mô hình triển khai thực nghiệm một cụm gồm 10 nút. ....	105
<b>Hình 3.7:</b> Sơ đồ bố trí các nút cảm biến TUmote. ....	106
<b>Hình 3.8:</b> So sánh tỷ lệ các nút còn sống trong mạng. ....	107
<b>Hình 3.9:</b> So sánh tỷ lệ chuyển phát bản tin dữ liệu. ....	107
<b>Hình 3.10:</b> So sánh sự cân bằng năng lượng giữa các nút mạng. ....	107
<b>Hình 3.11:</b> Khuôn viên khu vực tiến hành đo đạc thực nghiệm. ....	108
<b>Hình 3.12:</b> Triển khai các nút cảm biến tại khu vực thực nghiệm. ....	108
<b>Hình 3.13:</b> So sánh tỷ lệ các nút còn sống trong mạng. ....	109
<b>Hình 3.14:</b> So sánh tỷ lệ chuyển phát bản tin dữ liệu. ....	110
<b>Hình 3.15:</b> So sánh sự cân bằng năng lượng giữa các nút mạng. ....	110
<b>Hình 1:</b> Sơ đồ nguyên lý khối điều khiển của sản phẩm phần cứng TUmote. ....	121
<b>Hình 2:</b> Sơ đồ nguyên lý khối cảm biến và khối giao tiếp với mô đun vô tuyến... của sản phẩm phần cứng TUmote. ....	122
<b>Hình 3:</b> Sơ đồ nguyên lý khối giao tiếp người dùng, khối kết nối mở rộng ....	122
<b>Hình 4:</b> Sơ đồ mạch in sản phẩm phần cứng TUmote. ....	123
<b>Hình 5:</b> Giao thức CTP được xây dựng trên ..... ngăn xếp truyền thông RIME trong Contiki. ....	124
<b>Hình 6:</b> Cơ chế hoạt động của giao thức ContikiMAC. ....	126
<b>Hình 7:</b> Tạo mô phỏng mới với Cooja. ....	127
<b>Hình 8:</b> Sử dụng nút cảm biến giả lập trong Cooja. ....	127
<b>Hình 9:</b> Nạp mã nguồn chương trình cho nút cảm biến giả lập trong Cooja. ....	127
<b>Hình 10:</b> Cửa sổ giao diện tạo các nút cảm biến giả lập. ....	128
<b>Hình 11:</b> Phân bố của các nút cảm biến. ....	128

<b>Hình 12:</b> Mở giao diện Collect View của nút 30. ....	128
<b>Hình 13:</b> Cửa sổ giao diện Collect View của nút 30.....	129
<b>Hình 14:</b> Cửa sổ giao diện Contiki Test Editor.....	129
<b>Hình 15:</b> Cửa sổ giao diện Control Panel.....	129
<b>Hình 16:</b> Cửa sổ giao diện Log Listener. ....	130
<b>Hình 17:</b> Giao diện Lost (Over Time).....	130
<b>Hình 18:</b> Kết nối TUmote với mạch nạp.....	132
<b>Hình 19:</b> Chọn File mã nguồn cần nạp.....	133
<b>Hình 20:</b> Chọn AUTO PROG. để nạp chương trình cho TUmote .....	133
<b>Hình 21:</b> Triển khai các nút cảm biến TUmote.....	134
<b>Hình 22:</b> Giao diện chương trình thu thập dữ liệu trên máy tính.....	134

## DANH MỤC CÁC BẢNG BIỂU

<b>Bảng 2.1:</b> Mô hình năng lượng của TUmote.....	63
<b>Bảng 2.2:</b> Các trạng thái năng lượng của nút cảm biến.....	66
<b>Bảng 2.3:</b> Bảng mã hóa các trạng thái năng lượng còn lại của nút cảm biến. ....	67
<b>Bảng 2.4:</b> So sánh giữa các hệ điều hành TinyOS, Contiki, LiteOS.....	72
<b>Bảng 2.5:</b> Kịch bản đánh giá mô phỏng 1. ....	84
<b>Bảng 2.6:</b> Kịch bản đánh giá mô phỏng 2. ....	86
<b>Bảng 2.7:</b> Kết quả phân tích, đánh giá so sánh giữa giao thức EACTP với một số giao thức định tuyến có sự nhận thức về năng lượng khác.....	90
<b>Bảng 3.1:</b> So sánh một số nền tảng phần cứng mức thấp.....	99
<b>Bảng 3.2:</b> So sánh một số nền tảng phần cứng mức cao. ....	100
<b>Bảng 3.3:</b> Các thông số kỹ thuật của TUmote.....	104
<b>Bảng 3.4:</b> Kịch bản đánh giá thực nghiệm 1. ....	106
<b>Bảng 3.5:</b> Kịch bản đánh giá thực nghiệm 2. ....	109

# MỞ ĐẦU

## 1. Tính khoa học và tính cấp thiết của luận án

Mạng cảm biến không dây (Wireless Sensor Network) là một kết cấu hạ tầng bao gồm các thành phần cảm nhận (đo lường), tính toán và truyền thông nhằm cung cấp cho người quản trị khả năng đo đạc, quan sát và tác động lại với các sự kiện, hiện tượng trong một môi trường xác định. Các ứng dụng điển hình của mạng cảm biến không dây bao gồm các ứng dụng thu thập dữ liệu, theo dõi, giám sát, y học ...

Một mạng cảm biến không dây có thể bao gồm hàng trăm, hàng nghìn nút mạng. Các nút mạng thường là các thiết bị đơn giản, nhỏ gọn, giá thành thấp, sử dụng nguồn năng lượng hạn chế (thường dùng pin), có thời gian hoạt động lâu dài (từ vài tháng đến vài năm) và có thể hoạt động trong môi trường khắc nghiệt (như trong môi trường độc hại, ô nhiễm, nhiệt độ cao...).

Các nút cảm biến được phân bố rải rác trong trường cảm biến. Mỗi nút cảm biến có khả năng thu thập và định tuyến dữ liệu đến một điểm thu thập (Sink/Gateway) và người dùng cuối. Các nút giao tiếp với nhau qua mạng vô tuyến ad-hoc và truyền dữ liệu về nút gốc bằng kỹ thuật truyền đa chặng. Người dùng cuối/người quản lý có thể truyền thông với điểm thu thập thông qua Internet hay bất kỳ mạng không dây nào ví dụ như mạng di động, WiFi, WiMAX... hoặc người dùng cuối cũng có thể truyền thông trực tiếp với điểm thu thập.

Trong mạng cảm biến không dây, các nút cảm biến thực hiện đồng thời cả hai chức năng đó là:

- **Chức năng sinh dữ liệu:** Các nút cảm biến thu thập thông tin về các sự kiện trong trường cảm biến và thực hiện việc truyền thông để gửi dữ liệu của chúng về điểm thu thập.
- **Chức năng định tuyến dữ liệu:** Các nút cảm biến cũng tham gia vào quá trình chuyển tiếp các bản tin nhận được từ các nút lân cận trong tuyến đường đa chặng đến điểm thu thập.

Hầu hết các ứng dụng chính của mạng cảm biến không dây là thu thập thông tin cảm nhận được trong trường cảm biến nên các giao thức thu thập dữ liệu nhận được nhiều sự quan tâm nghiên cứu trong cộng đồng mạng cảm biến không dây.

Giao thức cây thu thập dữ liệu - CTP (Collection Tree Protocol) thực thi cơ chế thu thập dữ liệu tin cậy từng bước nhảy (hop-by-hop). Các nút tự tổ chức thành một cấu trúc dạng cây và dữ liệu luôn được gửi về nút cha cho tới khi đến được đỉnh của cây (nút gốc). Nút gốc được gán là đỉnh của cây và tất cả các nút khác được khởi tạo là các nút lá. Các nút lá sẽ cập nhật vị trí của nó trong cấu trúc cây và quá trình này được mở rộng trong toàn mạng với điểm xuất phát ban đầu là từ nút gốc. Dữ liệu được gửi qua một cấu trúc cây để đến được nút gốc.

Trong giao thức CTP, thước đo định tuyến được sử dụng là số lần truyền kỳ vọng - *ETX* (*Expected Transmission*) [1]. Thước đo chất lượng liên kết của một tuyến đường - *rtmetric* (*route metric*) được xác định bằng tổng *ETX* của tất cả các liên kết trên toàn tuyến đường đó. Vị trí của các nút trong cây được xác định bởi thước đo tuyến đường *rtmetric*. Tuyến đường có giá trị *rtmetric* càng lớn thì chất lượng các liên kết thuộc tuyến đường càng thấp. Tuyến đường tốt nhất là tuyến đường có *rtmetric* nhỏ nhất. Đây là tuyến đường có tổng số lần truyền kỳ vọng *ETX* đến nút gốc là nhỏ nhất và cũng là tuyến đường hiệu quả về mặt năng lượng nhất.

Trong thời gian gần đây, giao thức CTP cũng thu hút được nhiều sự quan tâm nghiên cứu trong cộng đồng mạng cảm biến [2], [3], [4], [5], [6], [7], [8]. Giao thức CTP đã được chứng minh là một giao thức thu thập dữ liệu đạt hiệu quả cao về mặt năng lượng tiêu thụ cũng như tỷ lệ chuyển phát thành công bản tin dữ liệu trong mạng. Tuy nhiên, giao thức CTP hiện tại chỉ dựa vào thước đo định tuyến *ETX* để lựa chọn tuyến đường tối ưu. Thước đo định tuyến *ETX* không giải quyết được vấn đề cân bằng năng lượng giữa các nút mạng. Các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt phải thực hiện nhiều việc truyền dẫn trong mạng. Các nút mạng này sẽ hết năng lượng nhanh hơn các nút mạng khác và tạo thành các lỗ hổng trong mạng, làm giảm hiệu năng của toàn bộ hệ thống mạng. Đây là một trong những thách thức quan trọng đối với các mạng cảm biến không dây hoạt động bằng pin. Các bài báo [9], [10], [11], [12] đã chỉ ra rằng, thời gian sống của mạng sẽ được kéo dài nếu năng lượng tiêu thụ trên toàn mạng đồng đều.

Một số kết quả đánh giá giao thức CTP hiện tại dựa trên công cụ mô phỏng Cooja [13] và thực nghiệm với phần cứng TUMote cũng cho các kết quả tương tự. Trong luận án này, tác giả đề xuất, thực thi và đánh giá một giao thức định tuyến mới EACTP (Energy Aware Collection Tree Protocol) có sự nhận thức về năng lượng nhằm giải quyết điểm yếu này của giao thức CTP. Các kết quả đánh giá dựa trên mô

phòng và thực nghiệm đã cho thấy giao thức EACTP đảm bảo được sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và thời gian sống của các nút mạng được cải thiện tốt hơn so với giao thức CTP ban đầu.

## **2. Đối tượng và phạm vi nghiên cứu**

*Đối tượng nghiên cứu* của luận án là bài toán định tuyến có sự nhận thức về năng lượng áp dụng cho giao thức cây thu thập dữ liệu (bài toán định tuyến EACTP) trên mạng cảm biến không dây nhằm đảm bảo được sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng.

*Phạm vi nghiên cứu* của luận án là tác giả tập trung vào các phương pháp định tuyến dựa trên sự nhận thức về năng lượng trong mạng cảm biến không dây.

## **3. Mục tiêu của luận án**

*Mục tiêu của luận án* là: Đề xuất một giao thức định tuyến mới EACTP (Energy Aware Collection Tree Protocol) có sự nhận thức về năng lượng nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Từ đó, thực thi và đánh giá hiệu năng của giao thức này dựa trên mô phỏng và thực nghiệm.

*Mục tiêu cụ thể của luận án* là:

- Nghiên cứu về các giao thức định tuyến có sự nhận thức về năng lượng cho mạng cảm biến không dây, phân tích đánh giá và so sánh các loại giao thức định tuyến này.
- Nghiên cứu giao thức định tuyến cây thu thập dữ liệu - CTP (Collection Tree Protocol) cho mạng cảm biến không dây.
- Đề xuất một giao thức định tuyến mới có sự nhận thức về năng lượng EACTP dựa trên giao thức CTP đã có.
- Xây dựng mô hình và phân tích mô hình toán cho giao thức EACTP dựa trên lý thuyết đồ thị.
- Thực thi và phân tích đánh giá hiệu năng của giao thức EACTP dựa trên mô phỏng.
- Xây dựng một hệ thống triển khai thực nghiệm cho phép tùy biến, tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của các giao thức này trong điều kiện thực tế.

#### 4. Phương pháp luận nghiên cứu

Phương pháp nghiên cứu trong luận án được kết hợp giữa nghiên cứu lý thuyết với nghiên cứu mô phỏng và thực nghiệm. Về nghiên cứu lý thuyết, tác giả nghiên cứu khảo sát các giao thức định tuyến có sự nhận thức về năng lượng cho mạng cảm biến không dây dựa vào các kiến thức cơ bản và các kết quả nghiên cứu lý thuyết đã được công bố. Về nghiên cứu mô phỏng và thực nghiệm, tác giả thực hiện cài đặt giao thức định tuyến mới, chạy thử nghiệm giao thức định tuyến mới dựa trên mô phỏng và thực nghiệm trên các nút cảm biến thật, từ đó tác giả đánh giá các kết quả mô phỏng, thực nghiệm và kết luận về tính ưu việt của giao thức định tuyến mới.

#### 5. Nội dung của luận án

Luận án được trình bày thành 03 chương như sau:

- **Chương 1: Bài toán định tuyến trong mạng cảm biến không dây.** Chương này trình bày những kết quả nghiên cứu khảo sát và đánh giá về các giao thức định tuyến dựa trên sự nhận thức về năng lượng cho mạng cảm biến không dây, phát biểu bài toán định tuyến có sự nhận thức về năng lượng với giao thức cây thu thập dữ liệu và lựa chọn phương pháp tiếp cận bài toán của tác giả trong luận án.
- **Chương 2: Giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng.** Chương này đề xuất một giao thức định tuyến mới EACTP có sự nhận thức về năng lượng. Giao thức EACTP được xây dựng nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Một số kết quả thực thi và đánh giá giao thức EACTP dựa trên công cụ mô phỏng Cooja cũng được trình bày trong chương này.
- **Chương 3: Triển khai đánh giá thực nghiệm.** Trong chương này, tác giả xây dựng một hệ thống triển khai thực nghiệm dựa trên 10 nút cảm biến phần cứng TUmote (Thainguyen University mote). Hệ thống triển khai thực nghiệm này cho phép tùy biến, tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của các giao thức này trong điều kiện thực tế. Một số kết quả đánh giá so sánh hiệu năng giữa giao thức EACTP và giao thức CTP thông qua một số mô hình triển khai thực nghiệm cũng được trình bày trong chương này.

Cuối cùng, tác giả đưa ra một số kết luận về kết quả thực hiện luận án và một số kiến nghị đề xuất cho luận án.



Ngoài các nội dung chính được trình bày trong các chương kể trên, những nội dung tham khảo mở rộng được trình bày trong năm phụ lục:

- Phụ lục 1: Bản thiết kế sơ đồ nguyên lý và sơ đồ mạch in phần cứng TUmote.
- Phụ lục 2: Ngăn xếp truyền thông RIME trong hệ điều hành Contiki.
- Phụ lục 3: Giao thức ContikiMAC trong hệ điều hành Contiki.
- Phụ lục 4: Đánh giá mô phỏng với Cooja.
- Phụ lục 5: Đánh giá thực nghiệm với TUmote.

## 6. Đóng góp của luận án

Trong quá trình hoạt động của mạng cảm biến không dây đòi hỏi phải có các giải pháp mới để đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Trong đó, vấn đề định tuyến có sự nhận thức về năng lượng được coi là giải pháp quan trọng. Trên cơ sở phân tích mô hình toán học dựa trên lý thuyết đồ thị cho giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng, tác giả đã có một số đóng góp mới trong luận án như sau:

- Đề xuất một giao thức định tuyến mới đó là giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng - EACTP (Energy Aware Collection Tree Protocol). Trong giao thức EACTP, tác giả đã thực hiện một số cải tiến: Thứ nhất, tác giả đã bổ sung thêm thành phần ước lượng năng lượng còn lại trên mỗi nút cảm biến; Thứ hai, tác giả đã đề xuất một thước đo định tuyến mới đó là trạng thái năng lượng còn lại *ES (Energy State)* để xác định tuyến đường tối ưu trong mạng; Thứ ba, tác giả đã đề xuất một thuật toán lựa chọn tuyến đường tối ưu mới dựa trên sự kết hợp giữa hai thước đo định tuyến là chất lượng liên kết của tuyến đường và trạng thái năng lượng còn lại trên nút chuyên tiếp.
- Mô phỏng và đánh giá hiệu năng giao thức EACTP thông qua một số mô hình mạng. Tác giả đã xác định các thước đo đánh giá phù hợp cho bài toán định tuyến EACTP và đưa ra một số kết quả đánh giá so sánh hiệu năng giữa giao thức EACTP và giao thức CTP. Các kết quả đánh giá mô phỏng cho thấy thời gian sống của mạng khi hoạt động theo EACTP mới đề xuất được cải thiện tốt hơn so với giao thức CTP ban đầu.

- Xây dựng một môi trường thực nghiệm dựa trên 10 nút cảm biến phần cứng TUmote (Thainguyen University mote), cho phép tùy biến, tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của các giao thức CTP, EACTP trong điều kiện thực tế. Môi trường thực nghiệm này hoàn toàn có thể sử dụng lại được cho các nghiên cứu thực nghiệm khác trong lĩnh vực mạng cảm biến không dây và có thể rút ngắn thời gian để đưa các mô hình nghiên cứu lý thuyết sang các mô hình thực nghiệm có tính ứng dụng. Kết quả đánh giá thực nghiệm trên 10 nút cảm biến phần cứng TUmote đã kiểm chứng lại tính đúng đắn của các kết quả mô phỏng cũng như những lập luận về tính hiệu quả của giao thức EACTP mới đề xuất.

# **Chương 1. BÀI TOÁN ĐỊNH TUYẾN TRONG MẠNG CẢM BIẾN KHÔNG DÂY**

Các mạng cảm biến không dây được triển khai cho các ứng dụng giám sát và tương tác với môi trường vật lý. Mặc dù mạng cảm biến không dây bao gồm các nút cảm biến đơn giản với tài nguyên hạn chế nhưng mạng cảm biến có thể phải thực hiện nhiều nhiệm vụ phức tạp, trong đó cần có sự tương tác giữa các nút mạng. Các nút cảm biến không dây có nguồn năng lượng và phạm vi truyền sóng hạn chế. Do đó, để truyền được dữ liệu về điểm thu thập thì các nút cảm biến phải thực hiện việc chuyển tiếp các bản tin thông qua các nút lân cận của chúng. Vì vậy, các giao thức định tuyến đóng một vai trò hết sức quan trọng trong mạng cảm biến không dây. Chương này sẽ mô tả những thách thức đối với vấn đề định tuyến trong mạng cảm biến không dây và các giải pháp định tuyến dựa trên sự nhận thức về năng lượng đã được nghiên cứu, đề xuất cho mạng cảm biến không dây.

## **1.1. Vấn đề năng lượng tiêu thụ trong mạng cảm biến không dây**

Vấn đề năng lượng tiêu thụ của nút cảm biến rất quan trọng bởi vì nhiều nút cảm biến bị hạn chế về nguồn năng lượng. Các nút cảm biến không dây có thể hoạt động bằng pin hoặc cũng có thể từ các nguồn năng lượng khác được tích trữ từ môi trường. Trong cả hai trường hợp, năng lượng đều là một nguồn tài nguyên hạn chế. Để kéo dài thời gian sống của mạng thì phần mềm trên các nút cảm biến cần phải quản lý năng lượng tiêu thụ một cách hiệu quả.

Hạn chế về tài nguyên năng lượng đã ảnh hưởng nhiều đến việc thiết kế phần cứng, phần mềm, giao thức mạng và cả kiến trúc mạng. Các nhà thiết kế phần cứng bắt buộc phải lựa chọn các linh kiện phần cứng có công suất thấp cũng như hỗ trợ chế độ ngủ hiệu quả về mặt năng lượng. Phần mềm chạy trên các nút cảm biến không dây cần phải tắt các thành phần phần cứng không sử dụng và đặt các thành phần phần cứng ở chế độ ngủ càng nhiều càng tốt. Nhờ sự hỗ trợ của các nhà phát triển phần mềm, các nút mạng cảm biến có thể chạy hệ điều hành hỗ trợ các cơ chế hoạt động công suất thấp giúp tiết kiệm năng lượng.

Vấn đề hiệu quả năng lượng ảnh hưởng nhiều đến kiến trúc mạng cũng như việc thiết kế các giao thức mạng. Trong mạng, quá trình truyền thông tiêu tốn nhiều năng lượng. Do vậy, điều quan trọng là cần xây dựng được các giao thức truyền

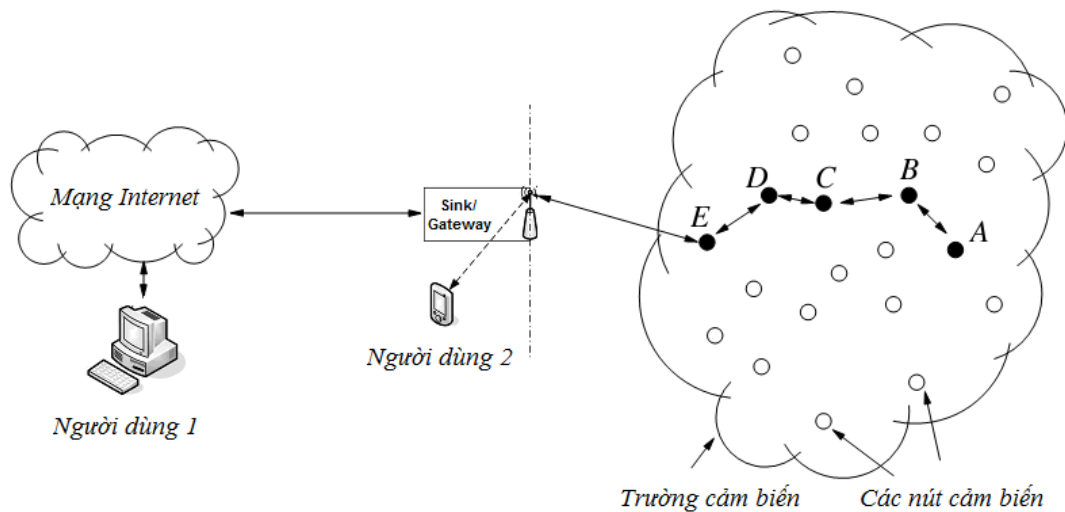
thông sao cho các nút cảm biến có thể sử dụng hiệu quả nguồn tài nguyên sẵn có. Do đó, phần cứng và phần mềm cần xác định được sự tiêu hao về năng lượng và cung cấp thông tin này đến tầng mạng để phục vụ cho việc định tuyến dữ liệu.

## 1.2. Tổng quan về vấn đề định tuyến trong mạng cảm biến không dây

### 1.2.1. Khái niệm và phân loại các giao thức định tuyến cho mạng cảm biến không dây

Định tuyến là một trong những giao thức quan trọng trong mạng cảm biến không dây. Nhiệm vụ của các giao thức định tuyến là tìm ra được tuyến đường tốt nhất từ nguồn đến đích.

Một mạng cảm biến không dây có thể bao gồm hàng trăm, hàng nghìn nút mạng. Các nút mạng thường là các thiết bị đơn giản, nhỏ gọn, giá thành thấp, sử dụng nguồn năng lượng hạn chế (thường dùng pin), có thời gian hoạt động lâu dài (từ vài tháng đến vài năm) và có thể hoạt động trong môi trường khắc nghiệt (như trong môi trường độc hại, ô nhiễm, nhiệt độ cao...).



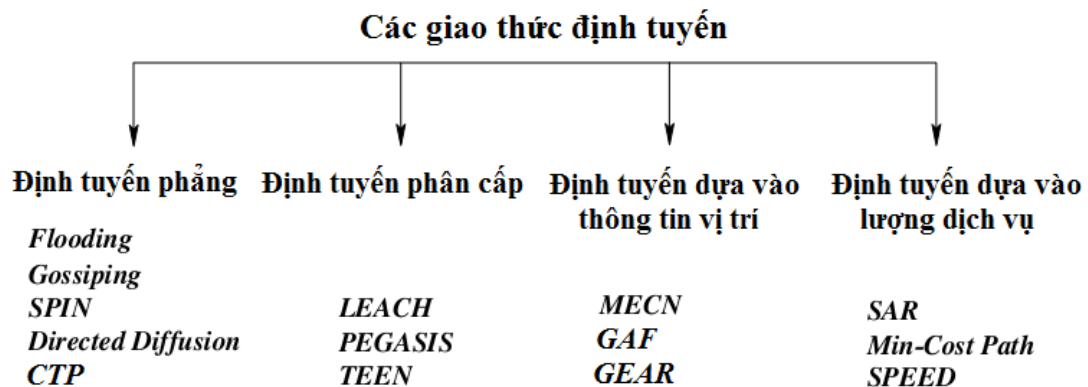
**Hình 1.1:** Mô hình mạng cảm biến không dây với các nút cảm biến phân bố rải rác trong trường cảm biến.

Các nút cảm biến được phân bố rải rác trong trường cảm biến như được minh họa ở hình 1.1. Mỗi nút cảm biến có khả năng thu thập và định tuyến dữ liệu đến một điểm thu thập (Sink/Gateway) và người dùng cuối. Trong mạng cảm biến không dây, các nút cảm biến thực hiện đồng thời cả hai chức năng đó là:

- **Chức năng sinh dữ liệu:** Các nút cảm biến thu thập thông tin về các sự kiện trong trường cảm biến và thực hiện việc truyền thông để gửi dữ liệu của chúng về điểm thu thập.

- **Chức năng định tuyến dữ liệu:** Các nút cảm biến cũng tham gia vào quá trình chuyển tiếp các bản tin nhận được từ các nút lân cận trong tuyến đường đa chặng đến điểm thu thập.

Trong thời gian qua, đã có nhiều giao thức định tuyến khác nhau được đề xuất cho mạng cảm biến không dây. Các giao thức định tuyến này có thể được phân loại thành bốn nhóm sau: Định tuyến phẳng, định tuyến phân cấp, định tuyến dựa vào thông tin vị trí và định tuyến dựa vào chất lượng dịch vụ [14]. Hình 1.2 giới thiệu một số giao thức định tuyến thuộc bốn nhóm này.



**Hình 1.2:** Phân loại các giao thức định tuyến cho mạng cảm biến không dây.

### 1.2.2. Thước đo định tuyến

Trong các giao thức định tuyến, thước đo định tuyến là thành phần quan trọng của chiến lược định tuyến. Giao thức định tuyến sẽ tính toán đường đi ngắn nhất dựa trên các thước đo định tuyến này. Một số thước đo định tuyến có thể được sử dụng trong mạng cảm biến không dây như:

- **Độ tin cậy của liên kết:** Rất nhiều bài báo nghiên cứu đã khảo sát các thước đo đánh giá độ tin cậy cho các liên kết tổn hao cũng như các liên kết công suất thấp. Thước đo đánh giá độ tin cậy được sử dụng phổ biến nhất là số lần truyền kỳ vọng *ETX* (*Expected Transmission*). Số lần truyền kỳ vọng *ETX* đặc trưng cho số lần truyền dẫn bản tin trung bình cần thiết để truyền thành công một bản tin.
- **Độ trễ:** Được sử dụng để thông báo trễ tuyến đường. Độ trễ có thể được sử dụng như một thước đo hoặc một ràng buộc. Khi được sử dụng như một thước đo thì đối tượng trễ biểu diễn tổng lượng trễ và trễ lớn nhất hay nhỏ nhất dọc theo tuyến đường. Khi được sử dụng như một ràng buộc thì độ trễ có thể được sử dụng để loại bỏ các liên kết có độ trễ lớn hơn một giá trị ngưỡng cho trước.

- **Số bước nhảy:** Xác định số bước nhảy dọc theo tuyến đường từ nút nguồn đến nút đích.
- **Năng lượng của nút:** Đây là một thước đo quan trọng trong mạng cảm biến không dây. Một vài thông số để mô tả năng lượng của một nút đã được đề xuất bao gồm: Chế độ nguồn năng lượng tiêu thụ của nút, năng lượng còn lại của một nút.

### 1.2.3. Đặc điểm của mạng cảm biến không dây ảnh hưởng đến vấn đề định tuyến

Các mạng cảm biến không dây phải thực hiện việc chuyển tiếp các bản tin đến nút gốc nên các mạng này có nhiều đặc điểm gần giống với những mạng khác. Mạng cảm biến không dây gần giống với mạng MANET ở một số phương diện như mạng tự tổ chức, đa chặng, phân tán và kết cấu hạ tầng mạng thường xuyên thay đổi. Những đặc điểm tương đồng này kết hợp với những đặc điểm riêng của mạng cảm biến không dây đã ảnh hưởng nhiều đến việc thiết kế các giao thức định tuyến. Những ảnh hưởng này bao gồm:

- Một mạng cảm biến không dây thường bao gồm một số lượng lớn các nút cảm biến được phân bố ngẫu nhiên với mật độ cao trong trường cảm biến.
- Trong một số trường hợp ứng dụng, các nút cảm biến thường được triển khai ngẫu nhiên và chúng ít được bảo trì. Do đó, các mạng cảm biến không dây phải tự cấu hình, tự phục hồi và vận hành theo cách phân tán.
- Các nút cảm biến bị hạn chế về nguồn năng lượng, khả năng xử lý, phạm vi truyền thông và dung lượng bộ nhớ. Ngoài ra, các mạng cảm biến không dây khác hẳn với các mạng MANET bởi vì nguồn pin của nút cảm biến chỉ được triển khai một lần và không được nạp lại.
- Mặc dù khả năng di động của các nút bị hạn chế nhưng cấu trúc liên kết mạng lại luôn thay đổi bởi vì các nguyên nhân sau: Các nút cảm biến có thể phải liên tục tiếp xúc với một môi trường khắc nghiệt dẫn đến việc hư hỏng các nút cảm biến, các nút cảm biến có thể hết năng lượng, bộ thu phát vô tuyến của nút cảm biến có thể đang ở chế độ ngủ, chất lượng liên kết vô tuyến không ổn định do ảnh hưởng bởi các mạng không dây khác sử dụng cùng một dải tần số.
- Các mạng cảm biến không dây chủ yếu được thiết kế để thu thập thông tin với các dòng dữ liệu chỉ theo một chiều từ các nút cảm biến đến nút gốc. Đặc điểm này có thể phân biệt mạng cảm biến không dây với các mạng khác (đặc biệt là mạng MANET).

Những đặc điểm này dẫn đến nhiều thách thức khi thiết kế giao thức định tuyến cho mạng cảm biến không dây.

#### **1.2.4. Những thách thức đối với vấn đề định tuyến trong mạng cảm biến không dây**

Các đặc điểm của mạng cảm biến không dây (đã được đề cập trong mục 1.2.3) đã làm cho việc phát triển mô hình định tuyến cho các mạng này gặp nhiều khó khăn. Sau đây là một số thách thức cần phải giải quyết khi phát triển các giao thức định tuyến cho mạng cảm biến không dây:

- **Thời gian sống dự kiến** của một mạng cảm biến không dây có thể kéo dài từ 1 đến 10 năm tùy thuộc vào từng ứng dụng. Nguồn năng lượng được tích trữ phụ thuộc vào dung lượng của pin. Các nút cảm biến không dây có kích thước rất nhỏ nên nguồn năng lượng của chúng cũng bị hạn chế. Điều này dẫn đến những ràng buộc khắt khe cho mọi hoạt động của các nút cảm biến. Trong cấu trúc phần cứng của nút cảm biến không dây, bộ thu phát vô tuyến là một trong những thành phần tiêu thụ năng lượng nhiều nhất. Do đó, giao thức định tuyến sẽ có ảnh hưởng nhiều đến thời gian sống của toàn mạng. Ngoài ra, mỗi nút cảm biến thực hiện đồng thời cả hai chức năng đó là: Chức năng khởi tạo dữ liệu và chức năng định tuyến dữ liệu. Một số nút bị ngừng hoạt động có thể gây ra những thay đổi về cấu trúc liên kết mạng và có thể cần phải tổ chức lại mạng. Để giảm năng lượng tiêu thụ thì các thuật toán định tuyến được đề xuất cho mạng cảm biến không dây sử dụng chiến thuật định tuyến hiệu quả về năng lượng với một số cách tiếp cận khác nhau như các phương thức phân nhóm, phân công vai trò riêng cho các nút trong mạng, tập hợp dữ liệu.
- **Khả năng mở rộng** là một vấn đề quan trọng trong mạng cảm biến không dây. Giao thức định tuyến cần phải hoạt động hiệu quả trong các mạng lớn bao gồm hàng ngàn các nút cảm biến. Việc định tuyến trong các trường hợp này gặp nhiều khó khăn bởi vì các nút cảm biến có khả năng xử lý và bộ nhớ lưu trữ rất hạn chế.
- **Khả năng lưu trữ và tính toán của các nút cảm biến** đã làm hạn chế nhiều đến các giao thức định tuyến. Do đó, các thuật toán định tuyến đơn giản, gọn nhẹ cần phải được nghiên cứu và phát triển cho các mạng cảm biến không dây. Thách thức này có thể được giải quyết với một chi phí thấp bằng cách sử dụng một số nút cảm biến có khả năng lưu trữ lớn hơn và tốc độ tính toán nhanh

hơn. Những mạng không đồng nhất như vậy cần phải được xem xét đến khi thiết kế giao thức định tuyến.

- **Các nút không đồng nhất** cần phải được tính đến khi thiết kế giao thức định tuyến cho mạng cảm biến không dây. Có hai nguyên nhân chính dẫn đến việc các nút trong mạng không đồng nhất. Thứ nhất là hoàn toàn có thể tăng được hiệu năng của mạng thông qua việc triển khai một số nút mạng có năng lượng, có khả năng lưu trữ và tính toán tốt hơn các nút còn lại trong mạng. Các nút này đóng vai trò là các nút chủ cụm để chuyển tiếp lưu lượng của các nút khác đến nút gốc. Thứ hai là sự khác biệt giữa các nút cảm biến có thể phát sinh trong quá trình hoạt động của mạng. Một số nút cảm biến có thể phải thực hiện nhiều nhiệm vụ hơn dẫn đến việc chúng bị hết năng lượng nhanh hơn các nút khác. Do đó, giao thức định tuyến cần phải tránh việc chuyển tiếp các bản tin thông qua các nút mạng có trạng thái nguồn năng lượng còn lại ở mức thấp để bù lại sự không đồng đều về năng lượng giữa các nút trong mạng.
- **Sự triển khai các nút mạng** trong mạng cảm biến không dây phụ thuộc vào ứng dụng. Quá trình triển khai các nút cảm biến có thể là ngẫu nhiên hoặc cũng có thể được xác định trước vị trí của từng nút trong mạng. Trong trường hợp triển khai ngẫu nhiên, các nút cảm biến bị phân tán ngẫu nhiên và các tuyến đường cần phải được xác định theo cách phân tán. Trong trường hợp còn lại, các nút cảm biến được triển khai thủ công và các bản tin có thể được chuyển tiếp thông qua các tuyến đường đã được xác định trước. Trong trường hợp mạng có kích thước lớn thì việc xác định tuyến đường sẽ được phân cấp.
- **Khả năng chịu lỗi** cũng cần phải được quan tâm khi định tuyến các bản tin. Tuy nhiên, khi một nút bị lỗi thì nó sẽ ảnh hưởng đến toàn bộ các hoạt động chung của mạng. Các thuật toán định tuyến cần phải có các tuyến đường dự phòng hoặc xây dựng kịp thời một tuyến đường khác trong trường hợp liên kết mạng bị lỗi.
- **Phạm vi truyền thông** có ảnh hưởng nhiều đến hoạt động của mạng. Các nút cảm biến có thể thay đổi công suất phát để tăng phạm vi truyền nhưng cùng với đó là sự tiêu hao nhiều hơn về nguồn năng lượng. Việc gửi các bản tin với công suất phát hạn chế, qua một khoảng cách ngắn có thể kéo dài thời gian sống của một nút mạng nhưng cũng làm tăng trễ truyền bản tin. Ngược lại, khi phạm vi truyền thông được mở rộng thì tổng năng lượng được sử dụng cho



việc xử lý các bản tin tại các nút trung gian sẽ giảm nhưng nhiều trong mạng cũng có thể xuất hiện nhiều hơn.

- **Chất lượng dịch vụ (QoS)** đặc trưng cho các yêu cầu dịch vụ cần được đáp ứng khi vận chuyển một luồng bản tin từ nguồn đến đích. Tuy nhiên, những yêu cầu ràng buộc về chất lượng dịch vụ trong các ứng dụng mạng cảm biến không dây có thể rất khác so với các mạng truyền thống. Các tham số chất lượng dịch vụ trong các mạng truyền thống có thể chưa đủ để mô tả chúng. Ngoài các tham số cơ bản như độ trễ thì còn có một số tham số khác như: Lỗi phân loại sự kiện (các sự kiện không chỉ được phát hiện mà còn được phân loại và xác suất lỗi phân loại sự kiện cần phải thấp), trễ phát hiện sự kiện (là độ trễ được tính từ thời điểm phát hiện một sự kiện và thời điểm gửi thông báo đến nút gốc), độ chính xác của việc bám mục tiêu (trong trường hợp ứng dụng bám mục tiêu thì vị trí của đối tượng cần phải được thông báo gần với vị trí của đối tượng trong thế giới thực nhất có thể).
- **Vấn đề di động** có thể gặp phải trong một số ứng dụng của mạng cảm biến không dây. Các nút cảm biến có thể cố định hoặc di động. Do đó, việc định tuyến các bản tin trở nên phức tạp hơn. Ngoài ra, trong một số trường hợp các nút gốc có thể di chuyển và điều này cũng cần phải tính đến khi thiết kế các mô hình định tuyến.

### **1.2.5. Một số phương pháp thiết kế giao thức định tuyến cho mạng cảm biến không dây**

Có nhiều phương pháp khác nhau được sử dụng trong việc thiết kế các giao thức định tuyến cho mạng cảm biến không dây. Các phương pháp này có thể đã được biết đến trong các mạng truyền thống và cũng có nhiều phương pháp mới được áp dụng cho các mạng cảm biến không dây. Sau đây là những ý tưởng định tuyến khác nhau đã được sử dụng.

#### ***1.2.5.1. Kỹ thuật Flooding, Gossiping và ước lượng tuyến đường***

Cách đơn giản nhất khi gửi một bản tin từ một nút cảm biến qua nhiều bước nhảy để đến được nút gốc đó là sử dụng kỹ thuật Flooding. Kỹ thuật Flooding thực hiện gửi bản tin đến tất cả các nút lân cận. Để hạn chế việc truyền lặp lại một bản tin trong mạng thì mỗi nút sẽ chỉ chuyển tiếp các bản tin mà nó chưa nhận được. Kỹ thuật Flooding đảm bảo rằng tất cả các bản tin sẽ được gửi tới nút gốc nếu nút nguồn gửi bản tin và nút gốc không ở trong các khu vực bị ngắt kết nối. Tuy nhiên, kỹ thuật

này cũng có nhiều nhược điểm ví dụ như: Sự bùng nổ về lưu lượng trong mạng gây ra bởi nhiều bản tin bị nhân bản được gửi đến cùng một nút, sự trùng lặp khi nhiều nút cảm nhận cùng một khu vực và gửi các bản tin giống nhau đến cùng một nút lân cận. Kỹ thuật Flooding cũng yêu cầu nhiều nguồn tài nguyên trong mạng. Điều này có thể khắc phục được bằng cách chỉ chuyển tiếp bản tin đến một nút lân cận bất kỳ thay vì gửi cho tất cả các nút lân cận. Kỹ thuật định tuyến này được gọi là Gossiping. Kỹ thuật này làm tăng trễ truyền bản tin và không có gì đảm bảo rằng bản tin sẽ được gửi đến nút gốc. Cả hai kỹ thuật định tuyến này đều đơn giản nhưng hiệu quả về trễ và thông lượng đều rất hạn chế. Nguyên nhân chính là do không có sự dự đoán trước tuyến đường và có quá nhiều sự truyền dẫn bản tin trong mạng. Do đó, cần phải sử dụng các kỹ thuật để xác định nút lân cận tốt nhất để chuyển tiếp bản tin. Nút lân cận tốt nhất được xác định bởi chi phí để gửi một bản tin đến nút gốc. Hàm chi phí này có thể được biểu diễn bằng các thước đo định tuyến khác nhau ví dụ như số bước nhảy đến nút gốc hoặc năng lượng cần thiết để gửi một bản tin đến nút gốc. Mỗi nút lưu trữ các thước đo định tuyến này trong bảng định tuyến. Sau đó, mỗi nút sẽ thực hiện chuyển tiếp các bản tin thông qua các nút lân cận có chi phí là nhỏ nhất.

#### ***1.2.5.2. Truyền thông trực tiếp, định tuyến phẳng và định tuyến phân cấp***

Chúng ta có thể phân loại các giao thức định tuyến theo cấu trúc mạng thành ba loại sau: Truyền thông trực tiếp, định tuyến phẳng và định tuyến phân cấp. Trong thuật toán truyền thông trực tiếp, các nút gửi trực tiếp bản tin đến nút gốc. Đối với các mạng cảm biến không dây có quy mô lớn thì những giao thức này không phù hợp bởi vì chúng không hiệu quả về năng lượng và có thể làm tăng trễ truy nhập kênh vô tuyến. Ngoài ra, sự xung đột các bản tin cũng làm hạn chế khả năng của kỹ thuật định tuyến này.

Đối với các giao thức định tuyến phẳng thì tất cả các nút ngang hàng với nhau. Khi một nút muốn gửi một bản tin thì nó gửi bản tin đó tới một trong các nút lân cận gần nút gốc nhất. Trễ truyền bản tin trong các giao thức định tuyến loại này tỷ lệ với khoảng cách giữa nút nguồn gửi bản tin và nút gốc. Các nút gần nút gốc có thể hết năng lượng nhanh hơn các nút ở xa nút gốc bởi vì bên cạnh việc phải gửi các bản tin của chính các nút đó thì chúng còn phải chuyển tiếp các bản tin đến từ các nút khác.

Các giao thức định tuyến phân cấp xác định vai trò khác nhau cho các nút. Các nút cảm biến được nhóm thành các cụm với một nút chủ cụm có trách nhiệm chuyển tiếp các bản tin từ cụm đến nút gốc. Hơn nữa, các nút chủ cụm có thể phối hợp với

nhau tạo thành các cụm lớp cao hơn. Các bản tin được gửi từ cụm lớp thấp đến cụm lớp cao hơn cho đến khi chúng đến được điểm thu thập. Việc chuyển tiếp dữ liệu giữa các lớp sẽ giảm trễ truyền bản tin bởi vì nó giảm số bước nhảy cần thiết để gửi bản tin đến nút gốc. Một số nút có thể có mức năng lượng và khả năng tính toán tốt hơn các nút khác trong mạng. Các nút này sẽ đóng vai trò là các nút chủ cụm và chúng có thể thực hiện các nhiệm vụ khác như chuyển tiếp dữ liệu, tập hợp dữ liệu và quản lý cụm. Các nút khác chỉ chịu trách nhiệm cảm nhận và gửi dữ liệu của chúng đến nút chủ cụm. Mô hình định tuyến phân cấp hiệu quả hơn và có khả năng mở rộng mạng tốt hơn hai mô hình định tuyến phẳng.

#### ***1.2.5.3. Quảng bá, đa phát và đơn phát***

Các giao thức định tuyến cho mạng cảm biến không dây có thể được phân loại thành ba nhóm: Quảng bá, đa phát và đơn phát. Các bản tin có thể được gửi đến một nút duy nhất (đơn phát) hoặc đến tất cả các nút trong mạng (quảng bá) hoặc đến một nhóm các nút (đa phát). So với định tuyến đơn phát thì định tuyến quảng bá và đa phát có thể thực hiện mà không cần các chi phí bổ sung bởi vì một sự truyền dẫn có thể lan truyền bản tin đến nhiều nút lân cận. Tuy nhiên, nếu có sự tiếp nhận nhiều bản tin thì cũng sẽ làm tăng sự tiêu hao năng lượng của nút cảm biến.

#### ***1.2.5.4. Các thuật toán tự tổ chức và các thuật toán tĩnh***

Cũng tương tự như các mạng có dây khác, chúng ta có thể phân chia các giao thức định tuyến trong mạng cảm biến không dây thành hai loại: Các thuật toán tự tổ chức và các thuật toán tĩnh. Trong các mạng có kích thước nhỏ thì các thuật toán định tuyến tĩnh dễ dàng hơn trong việc thiết kế, điều khiển và yêu cầu ít bộ nhớ hơn. Tuy nhiên, các thuật toán định tuyến tĩnh lại không phù hợp với các mạng cảm biến không dây, đặc biệt là các mạng có kích thước lớn bởi vì: Thứ nhất, các nút cảm biến không dây dễ bị lỗi nên cấu trúc liên kết ban đầu của mạng sẽ không phù hợp; Thứ hai, một số nút trong mạng cũng có thể di chuyển nên các thuật toán tĩnh không thể thích ứng được với cấu trúc mới của mạng; Thứ ba là việc tính toán tất cả các tuyến đường và lưu trữ các tuyến đường đó trên các nút cảm biến là không khả thi bởi sự hạn chế về khả năng lưu trữ và sự phức tạp của thuật toán khi hoạt động.

#### ***1.2.5.5. Các giao thức trung tâm dữ liệu và trung tâm nút***

Việc định tuyến trong các mạng truyền thông thường dựa trên cách tiếp cận trung tâm nút. Theo cách tiếp cận này thì một số nút nhất định được xác định địa chỉ bởi các nút nguồn và các bản tin sẽ được gửi đến các nút này. Một cách tiếp cận khác

đó là các giao thức trung tâm dữ liệu. Trong các giao thức trung tâm dữ liệu, nút gốc gửi yêu cầu dữ liệu đến các nút khác và nhận sự phản hồi từ các nút này. Ví dụ như nút gốc có thể yêu cầu sự phản hồi của các nút khác trong mạng khi cảm nhận được một sự kiện nào đó.

Định tuyến trung tâm dữ liệu cho phép xây dựng một kiến trúc mạng khác nhiều so với các mạng truyền thống dựa trên cách tiếp cận trung tâm nút. Thứ nhất, nó cho phép thực hiện việc tổng hợp dữ liệu. Thứ hai, nó cho phép biểu diễn các mô hình truyền thông trong mạng một cách đơn giản mà không cần phân biệt giữa các mô hình truyền thông nhiều - nhiều, nhiều - một, một - nhiều hoặc một - một.

#### ***1.2.5.6. Nhận thức về vị trí***

Đối với các giao thức định tuyến dựa trên vị trí thì các nút cảm biến biết được thông tin về vị trí của chúng. Trong hầu hết các ứng dụng của mạng cảm biến không dây, các nút cảm biến cần biết tọa độ của chúng. Do đó, việc định tuyến có thể dựa vào các thông tin vị trí có sẵn này. Các thuật toán định tuyến dựa trên vị trí có thể giúp tiết kiệm đáng kể năng lượng tiêu thụ trong mạng.

#### ***1.2.5.7. Tối thiểu năng lượng tiêu thụ cho mỗi bản tin và tối đa thời gian sống của mạng***

Thách thức chính trong việc thiết kế các mạng cảm biến không dây đó là đạt được sự hiệu quả về năng lượng. Có một số cách tiếp cận để đạt được sự hiệu quả về năng lượng. Chúng ta có thể phân biệt hai cách tiếp cận chính để xây dựng một thuật toán định tuyến có sự nhận thức về năng lượng đó là: Giảm thiểu năng lượng tiêu thụ cho mỗi bản tin và tối đa thời gian sống của mạng. Cách tiếp cận đầu tiên có liên quan đến việc giảm thiểu tổng năng lượng cần thiết để truyền một bản tin qua một tuyến đường đa chặng từ nguồn đến đích. Cách tiếp cận thứ hai là kéo dài thời gian sống của mạng. Thời gian sống của mạng có thể được xác định bằng một trong ba cách sau: Đó là thời gian cho đến khi nút đầu tiên trong mạng hết năng lượng hoặc thời gian cho đến khi có sự phân vùng mạng (khi có hai nút không thể giao tiếp được với nhau).

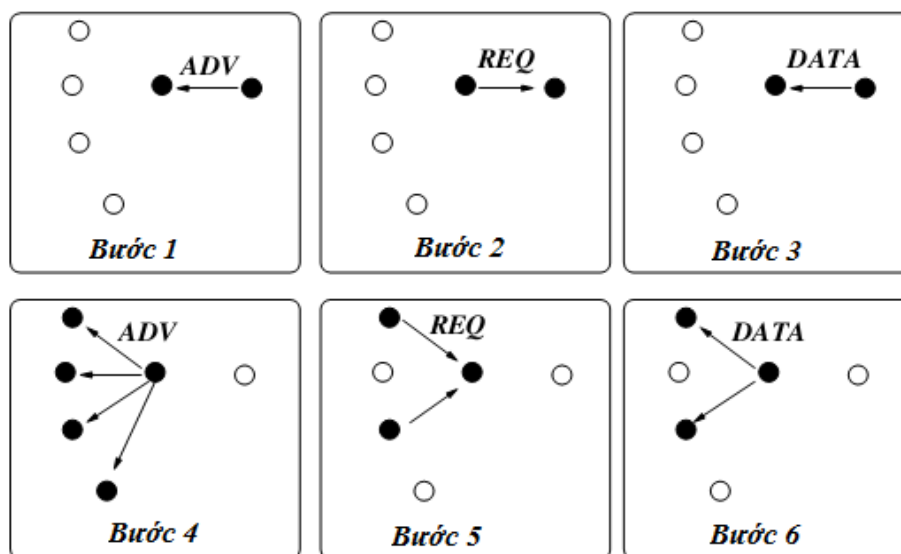
### 1.3. Một số giao thức định tuyến hiệu quả về mặt năng lượng cho mạng cảm biến không dây

#### 1.3.1. Giao thức định tuyến trung tâm dữ liệu

##### 1.3.1.1. Giao thức định tuyến đối với thông tin thông qua đàm phán (SPIN)

SPIN (Sensor Protocols for Information via Negotiation) là một họ các giao thức định tuyến được thiết kế để giải quyết các nhược điểm của giao thức Flooding thông qua sự dàn xếp và thích ứng tài nguyên. Để đạt được mục đích này, có hai cách tiếp cận chính được sử dụng như sau:

- Thứ nhất, thay vì gửi tất cả các bản tin dữ liệu thì các nút cảm biến sẽ dàn xếp với nhau thông qua các bản tin mô tả dữ liệu. Bởi vậy, thông tin quan sát được chỉ gửi đến các nút cảm biến quan tâm đến dữ liệu đó.
- Thứ hai, mỗi nút cảm biến sẽ giám sát tài nguyên năng lượng của nó. Tài nguyên năng lượng này sẽ được sử dụng để thực hiện các quyết định có sự nhận thức về năng lượng.



**Hình 1.3:** Giao thức SPIN [14].

Hình 1.3 là một ví dụ minh họa giao thức SPIN. Giao thức SPIN được thực hiện thông qua ba loại bản tin là bản tin thông báo (ADV), bản tin yêu cầu (REQ), bản tin dữ liệu (DATA) như được minh họa ở hình 1.3. Trước khi gửi một bản tin dữ liệu thì một nút sẽ thông báo ý định của nó thông qua việc gửi quảng bá một bản tin ADV (Bước 1). Bản tin ADV chứa một mô tả về bản tin DATA được gửi và bản tin ADV có kích thước nhỏ hơn so với bản tin DATA. Sau đó, nếu một nút lân cận quan tâm đến bản tin ADV thì nó sẽ trả lời lại bằng một bản tin REQ (bước 2). Cuối cùng,

bản tin DATA sẽ được gửi tới nút yêu cầu nó (bước 3). Quá trình truyền bản tin dữ liệu trong mạng được thực hiện theo cơ chế này tại mỗi bước nhảy. Chúng ta có thể thấy ở các bước 4, 5 và 6 của hình 1.3, nhiều nút có thể gửi bản tin REQ tới cùng một nút và nút đó sẽ gửi lại bản tin DATA tới từng nút cho đến khi tất cả các nút gửi yêu cầu đều nhận được.

Nguyên lý hoạt động được minh họa trong hình 1.3 liên quan đến giao thức SPIN điểm - điểm (SPIN-PP). Ngoài giao thức SPIN-PP còn có một số dạng khác đã được đề xuất để khắc phục nhược điểm của giao thức SPIN-PP ví dụ như giao thức SPIN có sự nhận thức về năng lượng tiêu thụ (SPIN-EC), giao thức SPIN cho các mạng quảng bá (SPIN-BC) và giao thức SPIN đảm bảo độ tin cậy (SPIN-RL).

Giao thức SPIN-PP chưa khắc phục được nhược điểm về khả năng nhận thức được năng lượng của các giao thức Flooding hay Gossiping. Giao thức SPIN-EC đã khắc phục nhược điểm này thông qua một cơ chế bảo tồn năng lượng đơn giản. Cơ chế này xác định một ngưỡng năng lượng. Khi năng lượng còn lại của nút thấp hơn ngưỡng thì nút đó sẽ không tham gia vào quá trình hoạt động của giao thức. Ví dụ như nút đó sẽ không gửi bản tin REQ nếu nó không đủ năng lượng để gửi bản tin REQ và nhận bản tin DATA. Bởi vì các nút tham gia vào hoạt động giao thức phụ thuộc vào năng lượng còn lại nên nếu một nút còn nhiều năng lượng thì hoạt động của giao thức SPIN-EC cũng giống như giao thức SPIN-PP.

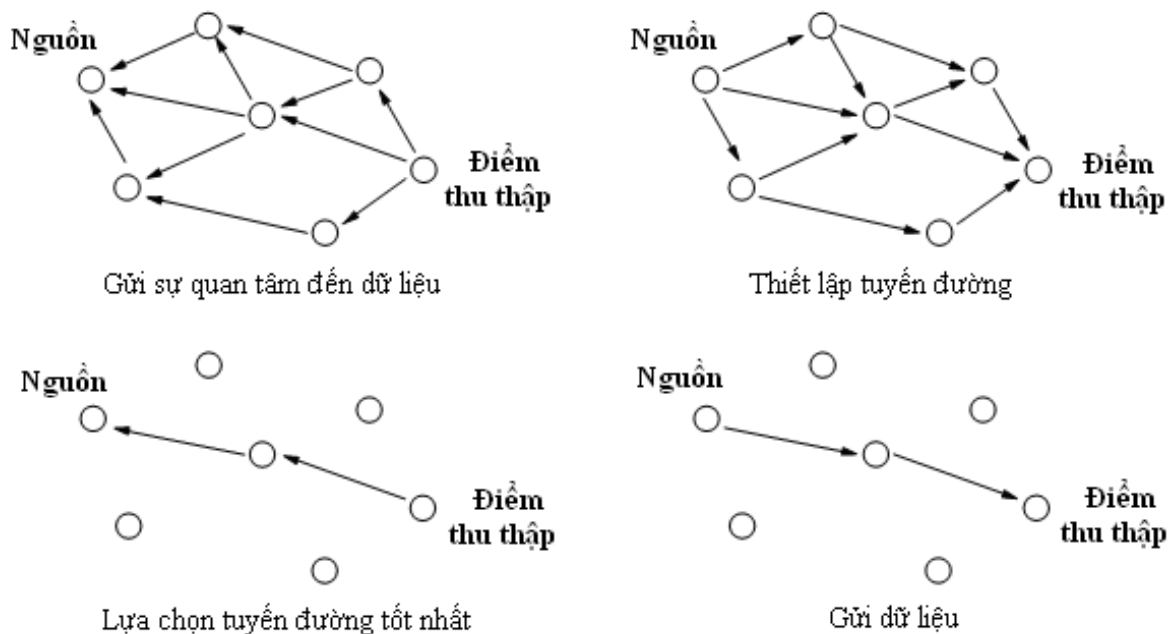
Một nhược điểm khác của giao thức SPIN-PP đó là khi có nhiều nút gửi bản tin REQ thì bản tin DATA sẽ được gửi đến từng nút riêng lẻ. Cách tiếp cận này sẽ gây lãng phí năng lượng. Hơn nữa, giao thức SPIN-PP không hỗ trợ bất kỳ một cơ chế tránh xung đột nào khi có nhiều bản tin REQ cùng được gửi. Để khắc phục nhược điểm này, giao thức SPIN-BC được đề xuất cho các mạng quảng bá. Trong giao thức SPIN-BC, nếu một nút quan tâm đến một bản tin dữ liệu nhưng nó lại lắng nghe được bản tin REQ có liên quan đến dữ liệu đó thì nút đó sẽ hủy bản tin REQ và chờ đợi để nhận bản tin DATA. Khi nhận được một bản tin REQ thì nút gửi sẽ phát quảng bá bản tin DATA và tất cả các nút lân cận quan tâm đến bản tin DATA đều có thể nhận được. Như vậy, giao thức SPIN-BC giảm được chi phí về năng lượng tiêu thụ xảy ra khi có nhiều nút lân cận cùng quan tâm đến dữ liệu.

SPIN là giao thức định tuyến trung tâm dữ liệu mà ở đó các nút cảm biến phát quảng bá các thông báo đối với các dữ liệu sẵn có và chờ đợi yêu cầu từ các nút gốc quan tâm đến dữ liệu đó. So với giao thức Flooding thì giao thức SPIN-PP giảm năng

lượng tiêu thụ lên tới 70% bởi vì đã loại bỏ được những truyền dẫn dư thừa trong mạng [14]. Tuy nhiên khi so sánh với giao thức Flooding thì độ trễ trong giao thức SPIN cao hơn bởi vì cần thêm chi phí về thời gian cho cơ chế bắt tay giữa các nút.

### 1.3.1.2. Giao thức định tuyến lan truyền trực tiếp (Directed Diffusion)

Giao thức SPIN cung cấp các cơ chế hiệu quả để các nút cảm biến gửi dữ liệu mà chúng cảm nhận được đến các nút quan tâm đến dữ liệu đó. Vì vậy, lưu lượng dữ liệu trong giao thức SPIN được bắt nguồn từ các nút cảm biến và thường kết thúc tại các nút gốc. Tuy nhiên, kiểu lưu lượng dữ liệu này có thể không phù hợp khi người dùng yêu cầu thông tin cụ thể từ các nút cảm biến. Giao thức định tuyến Directed Diffusion [15] đã được đề xuất để giải quyết yêu cầu này.



**Hình 1.4:** Hoạt động của giao thức Directed Diffusion [14].

Đối với giao thức Directed Diffusion, nút gốc yêu cầu dữ liệu từ các nút cảm biến bằng cách gửi quảng bá một bản tin yêu cầu (ví dụ “gửi cho tôi nhiệt độ tại một khu vực A”) và yêu cầu này được lan truyền trong mạng từ nút này đến nút khác theo mô hình đa chặng. Do vậy, tất cả mọi nút trong trường cảm biến sẽ biết được yêu cầu. Khi yêu cầu này được lan truyền trong mạng thì mỗi nút sẽ duy trì một hướng liên kết với nhau. Sau khi dữ liệu được cảm nhận thì nó có thể gửi ngược trở lại nút gốc. Tuy nhiên, cũng có thể xảy ra trường hợp một nút nhận được dữ liệu giống nhau từ nhiều nút lân cận và nút đó sẽ có nhiều tuyến đường khác nhau để gửi dữ liệu về nút gốc. Trong trường hợp đó, nút có thể chọn tuyến đường tốt nhất dựa trên một số thước đo

định tuyến ví dụ như độ trễ của tuyến đường. Để tiết kiệm năng lượng, các nút chuyển tiếp có thể thực hiện một vài xử lý trong mạng chẳng hạn như tập hợp dữ liệu dựa vào tên và giá trị các thuộc tính. Tuy nhiên để thực hiện các xử lý này thì cũng đòi hỏi năng lượng xử lý nhiều hơn.

### **1.3.1.3. Đánh giá**

Các giao thức định tuyến trung tâm dữ liệu có một số ưu điểm đó là:

- Thứ nhất, các giao thức trung tâm dữ liệu cung cấp các tuyến đường tùy thuộc vào ứng dụng dựa trên sự quan tâm đến dữ liệu của người dùng. Mỗi khi bộ cảm biến đo được sự thay đổi thì các tuyến đường trong mạng cũng thích ứng với những sự thay đổi này để đáp ứng được các yêu cầu của người dùng. Giao thức định tuyến trung tâm dữ liệu cũng không yêu cầu một cơ chế đánh địa chỉ toàn cầu cho các nút cảm biến. Điều này sẽ mang lại hiệu quả về năng lượng bởi vì các tuyến đường được xây dựng khi có sự quan tâm đến dữ liệu của người dùng và cũng không cần thiết phải duy trì một cấu trúc liên kết mạng.
- Thứ hai, các giao thức trung tâm dữ liệu cũng giảm được năng lượng tiêu thụ trong mạng thông qua việc xác định rõ các tuyến đường được lựa chọn cho từng sự quan tâm đến dữ liệu cụ thể của người dùng. Theo đó, chỉ những nút có thông tin phù hợp mới tham gia vào việc cung cấp các thông tin.

Nhược điểm chính của các giao thức trung tâm dữ liệu đó là:

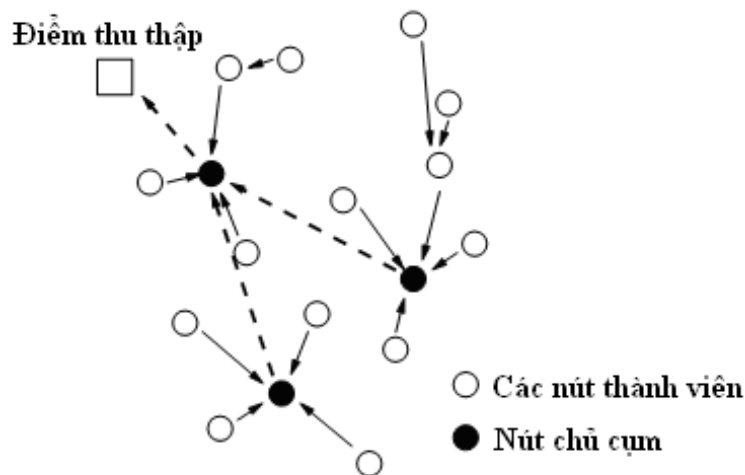
- Thứ nhất, chúng thường dựa trên cấu trúc phẳng. Điều này dẫn đến việc hạn chế khả năng mở rộng mạng và làm tăng hiện tượng tắc nghẽn đối với các nút ở gần nút gốc. Vì vậy, cần thiết phải có các cơ chế tổng hợp dữ liệu để giảm lưu lượng thông tin tại mỗi phần của mạng.
- Thứ hai, các giao thức định tuyến chẳng hạn như giao thức Directed Diffusion chỉ có thể áp dụng cho một vài ứng dụng của mạng cảm biến không dây. Trong các giao thức này, quá trình truyền thông được thiết lập bởi các câu truy vấn được tạo ra bởi nút gốc. Điều này làm cho giao thức Directed Diffusion không phải là một sự lựa chọn tốt cho các ứng dụng đòi hỏi cần phải cung cấp thông tin liên tục.
- Thứ ba, quá trình xử lý kết hợp giữa dữ liệu và các câu truy vấn cũng làm phát sinh một số chi phí về tài nguyên tại các nút cảm biến.



### 1.3.2. Giao thức định tuyến dựa trên sự phân cụm

Đối với các giao thức trung tâm dữ liệu và dựa trên kiến trúc phẳng, phần lớn thông tin tạo ra được tập trung ở các nút cảm biến gần với nút gốc. Điều này dẫn đến việc quá tải tại các nút ở gần nút gốc khi mật độ mạng tăng. Các nút ở gần nút gốc sẽ phải định tuyến nhiều thông tin hơn so với các nút khác ở trong mạng. Các nút này cũng sẽ hết năng lượng nhanh hơn các nút khác và tạo thành lỗ hổng trong mạng. Do đó, các giao thức dựa trên kiến trúc phẳng có một nhược điểm lớn đó là sự mất cân bằng năng lượng trong toàn mạng và hạn chế khả năng mở rộng của mạng.

Nhược điểm này của kiến trúc phẳng có thể được khắc phục bởi kiến trúc mạng phân cấp mà ở đó các nút được nhóm thành các cụm. Sự tương tác giữa các nút trong cụm được điều khiển bởi một nút chủ cụm như hình 1.5. Dựa theo kiến trúc phân cấp này, một số giao thức định tuyến phân cấp đã được nghiên cứu và phát triển để giải quyết những thách thức về vấn đề năng lượng tiêu thụ và khả năng mở rộng mạng. Một số giao thức được thiết kế dựa trên cách tiếp cận phân cụm như: LEACH (Low Energy Adaptive Clustering Hierarchy) [16], PEGASIS (Power-Efficient Gathering in Sensor Information System) [17], TEEN (Threshold sensitive Energy Efficient sensor Network) [18].



**Hình 1.5:** Kiến trúc phân cấp dựa trên các cụm [14].

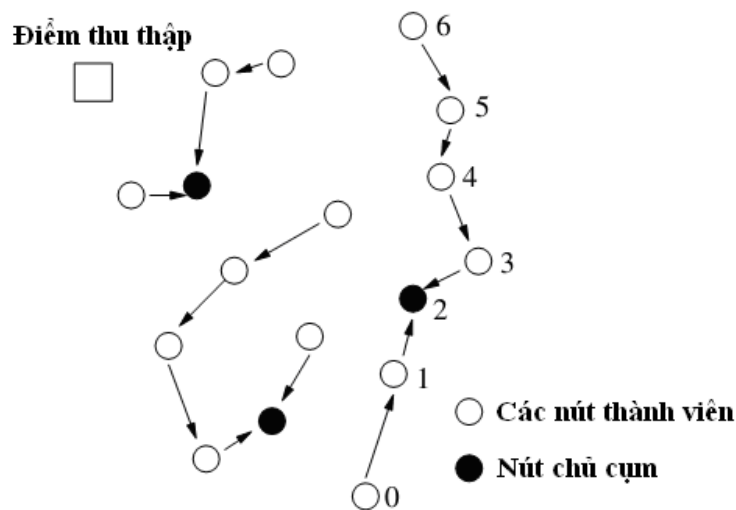
#### 1.3.2.1. Giao thức LEACH

Giao thức LEACH là một trong những giao thức định tuyến phân cụm đầu tiên. Các cụm được hình thành theo cách tự tổ chức. Nút chủ cụm thu thập và kết hợp dữ liệu thu thập được từ các nút trong cụm của nó và gửi dữ liệu đó tới điểm thu thập. Giao thức LEACH sử dụng quá trình ngẫu nhiên để lựa chọn các nút chủ cụm. Giao

thức này tạo sự cân bằng năng lượng giữa các nút và kéo dài thời gian sống của mạng.

### 1.3.2.2. Giao thức PEGASIS

Giao thức PEGASIS là giao thức dựa trên chuỗi gần như tối ưu và là một sự cải tiến hơn so với giao thức LEACH. Giao thức PEGASIS giải quyết vấn đề chi phí phát sinh trong quá trình hình thành cụm của giao thức LEACH thông qua việc xây dựng các chuỗi nút như được minh họa trong hình 1.6. Quá trình xây dựng chuỗi được thực hiện theo thuật toán tham lam. Mỗi nút sẽ lựa chọn nút lân cận gần nhất làm bước nhảy tiếp theo trong cấu trúc chuỗi. Sau khi chuỗi hình thành, thay vì phải duy trì thông tin về cụm và các thành viên trong cụm thì mỗi nút chỉ cần theo dõi các nút lân cận tiếp theo và phía trước nó trong chuỗi. Quá trình truyền thông trong chuỗi được thực hiện tuần tự. Mỗi nút trong chuỗi sẽ tập hợp dữ liệu từ các nút lân cận của nó cho đến khi tất cả dữ liệu được tập hợp tại nút đầu chuỗi. Nút đầu chuỗi điều khiển thứ tự quá trình truyền thông giữa các nút trong chuỗi.



**Hình 1.6:** Cấu trúc chuỗi của PEGASIS [14].

Một ví dụ về quá trình truyền thông trong chuỗi được thể hiện trong hình 1.6. Nút đầu chuỗi trong ví dụ này là nút số 2. Đầu tiên nút 2 gửi thông báo đến nút 0 để khởi tạo quá trình truyền thông. Nút 0 gửi dữ liệu của nó đến nút 1. Nút 1 sẽ tổng hợp các dữ liệu này với dữ liệu của bản thân nó để tạo ra một bản tin dữ liệu có cùng một độ dài. Bản tin này được truyền đến nút 2. Sau khi nút 2 nhận được bản tin dữ liệu từ nút 1, nút 2 sẽ gửi thông báo đến đầu kia của chuỗi (nút 6). Thông tin từ các nút 6, 5, 4, 3 cũng được tổng hợp và được gửi đến nút 2. Sau khi nhận được thông tin tổng hợp

trong chuỗi thì nút 2 sẽ gửi thông tin tổng hợp này đến điểm thu thập thông qua một bước nhảy duy nhất.

Giao thức PEGASIS đem lại hiệu quả về mặt năng lượng hơn so với giao thức LEACH. Điều này có được là nhờ giảm chi phí về năng lượng truyền thông trong chuỗi. Tuy nhiên giao thức PEGASIS lại có độ trễ lớn hơn so với giao thức LEACH bởi vì bản tin dữ liệu phải được truyền tuần tự trong chuỗi và nút đầu chuỗi phải đợi cho đến khi tất cả các bản tin nhận được trước khi truyền thông với điểm thu thập. Ngoài ra, giao thức PEGASIS yêu cầu tất cả thông tin trong chuỗi cần phải được tổng hợp vào trong một bản tin duy nhất. Điều này có thể gây ra sự thiếu chính xác trong thông tin được gửi đến điểm thu thập.

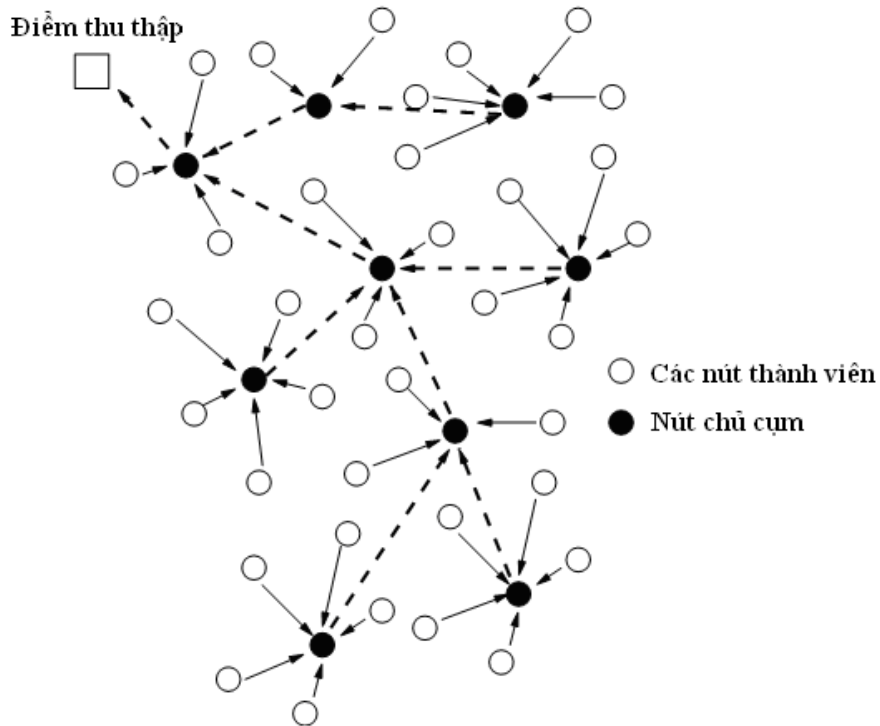
### **1.3.2.3. Giao thức TEEN**

Các giao thức LEACH và PEGASIS hỗ trợ cho các ứng dụng mà ở đó thông tin từ các nút cảm biến được gửi định kỳ tới điểm thu thập. Do đó, nội dung thông tin từ nhiều nút mạng được giảm đi đáng kể nhờ các kỹ thuật tổng hợp dữ liệu. Tuy nhiên, các giao thức này có thể không phù hợp với các ứng dụng dựa trên sự kiện. Trong các ứng dụng này, thông tin chỉ được tạo ra khi xuất hiện các sự kiện. Giao thức TEEN [18] được thiết kế nhằm mục đích hỗ trợ khả năng chuyển phát dữ liệu dựa trên các sự kiện ở trong mạng. Trong giao thức TEEN, các tuyến đường đa chặng được tạo ra phụ thuộc vào một ngưỡng. Ngưỡng này có liên quan với dữ liệu cảm nhận được và được thiết lập bởi ứng dụng.

Giao thức TEEN tổ chức các nút cảm biến thành nhiều mức phân cấp như hình 1.7. Trong kiến trúc phân cấp này, đầu tiên dữ liệu từ các nút cảm biến được gửi tới các nút chủ cụm. Các nút chủ cụm thu thập, tổng hợp và truyền những dữ liệu này tới nút chủ cụm ở mức cao hơn cho đến khi tới được điểm thu thập. Để đảm bảo sự cân bằng năng lượng trong mạng thì các nút chủ cụm được thay đổi định kỳ trong cụm.

Dựa theo cấu trúc mạng phân cấp, giao thức TEEN hỗ trợ khả năng truyền thông dựa trên sự kiện thông qua hai ngưỡng: Ngưỡng cứng (*HT - Hard Threshold*) và ngưỡng mềm (*ST - Soft Threshold*). Các nút cảm biến được lập trình để đáp ứng lại với những thay đổi của dữ liệu cảm nhận được (ví dụ như nhiệt độ, độ ẩm...) thông qua việc so sánh giá trị đo được với một ngưỡng cứng. Nếu giá trị mới cảm nhận được vượt quá ngưỡng cứng thì các nút cảm biến sẽ gửi dữ liệu của nó đến nút chủ cụm. Nếu sự kiện có thể kéo dài trong một thời gian dài thì quá trình truyền dữ liệu trong mạng sẽ diễn ra thường xuyên hơn. Để giảm sự dư thừa dữ liệu trong

trường hợp này, một ngưỡng mềm sẽ được sử dụng. Bất cứ khi nào giá trị dữ liệu mới nhận được vượt quá giá trị ngưỡng cứng thì các nút cảm biến cũng sẽ kiểm tra ngưỡng mềm cho các giá trị dữ liệu mới tiếp theo. Nếu sự sai khác giữa các giá trị dữ liệu mới liên tiếp nhau không vượt quá ngưỡng mềm thì nút cảm biến sẽ không gửi đi thông tin này. Nút cảm biến sẽ thông báo tới nút chủ cụm về sự giống nhau giữa các giá trị dữ liệu mới cảm nhận được. Giá trị dữ liệu mới cảm nhận được chỉ được gửi đi khi vượt quá ngưỡng mềm.



**Hình 1.7:** Kiến trúc phân cấp trong giao thức TEEN [14].

Bởi vì giao thức TEEN dựa trên các giá trị ngưỡng cố định nên nó không phù hợp với các ứng dụng đòi hỏi dữ liệu cần phải được gửi về điểm thu thập một cách định kỳ.

#### 1.3.2.4. **Đánh giá**

Các giao thức định tuyến phân cấp có một số ưu điểm đó là:

- Thứ nhất, chúng hỗ trợ tốt khả năng mở rộng mạng thông qua việc phân chia mạng thành các cụm.
- Thứ hai, cơ chế phân cụm động cũng đạt được hiệu quả về mặt năng lượng tốt hơn so với các giao thức dựa trên kiến trúc phẳng. Trong các mạng cảm biến không dây dựa trên sự kiện thì các nút cảm biến chủ yếu ở chế độ ngủ. Một

lượng nhỏ các nút chủ cụm thực hiện các nhiệm vụ xử lý và tổng hợp dữ liệu phức tạp hơn so với các nút còn lại trong mạng.

Tuy nhiên, các giao thức định tuyến dựa trên sự phân cụm cũng tồn tại một số nhược điểm đó là:

- Thứ nhất, chúng dựa nhiều vào các nút chủ cụm. Khi các nút chủ cụm bị lỗi thì cả cụm sẽ không thể thực hiện được việc truyền thông với điểm thu thập.
- Thứ hai là quá trình xây dựng cụm đòi hỏi bổ sung thông tin báo hiệu trong mạng và làm tăng chi phí về năng lượng trong mạng.
- Thứ ba, các giao thức định tuyến này được thiết kế cho các mạng tĩnh và không thể xử lý đối với vấn đề di động trong mạng. Trong các kịch bản di động, các nút chủ cụm cũng sẽ di chuyển. Do vậy, nó cần phải thường xuyên tính toán nút chủ cụm khi chúng di chuyển ra ngoài phạm vi truyền thông của nhóm.
- Các giao thức này cũng không thích hợp với các ứng dụng đòi hỏi độ chính xác về thời gian bởi vì cần phải mất thời gian để tính toán các nút chủ cụm.

### **1.3.3. Giao thức định tuyến dựa trên vị trí**

Trong các giao thức định tuyến dựa trên thông tin vị trí, thông tin về vị trí của các nút cảm biến được sử dụng để thực hiện quá trình định tuyến. Có rất nhiều giao thức đã được đề xuất dựa trên cách tiếp cận này. Một số giao thức định tuyến được biết đến nhiều nhất theo cách tiếp cận này đó là: MECN (Minimum Energy Communication Network) [19], GAF (Geographic Adaptive Fidelity) [20], GEAR (Geographic and Energy Aware Routing) [21]. Một ứng dụng mạng cảm biến không dây có thể quan tâm đến việc biết thông tin tại một khu vực X tại thời điểm Y. Cách thức hiệu quả nhất là dựa vào các thông tin về vị trí trong câu truy vấn để gửi trực tiếp câu truy vấn đến các khu vực cần quan tâm thay vì gửi câu truy vấn đến tất cả các nút trong mạng.

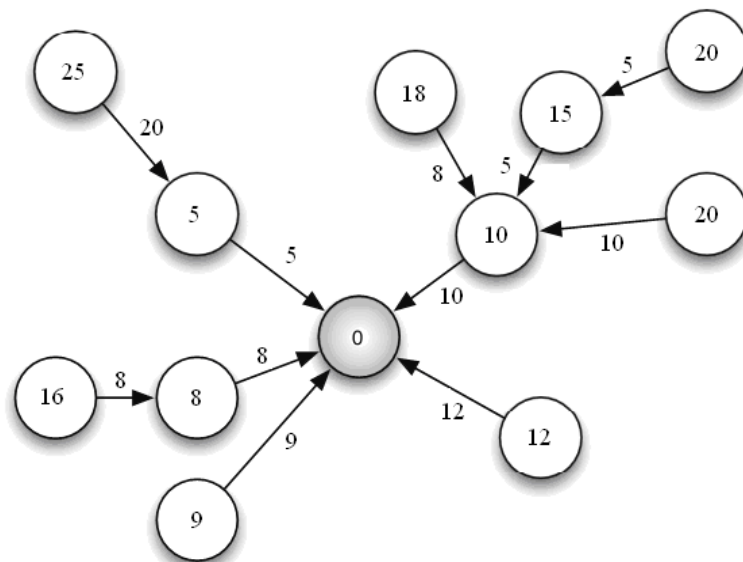
Các giao thức định tuyến dựa trên vị trí sử dụng thông tin về vị trí cục bộ để xác định tuyến đường. Hiệu quả của các giao thức này phụ thuộc vào độ chính xác trong việc xác định vị trí của các nút mạng. Nếu việc xác định vị trí không chính xác thì nó có thể gây ra lỗi trong quá trình định tuyến. Ví dụ, các nút cảm biến có thể dễ bị hư hỏng khi được triển khai trong các môi trường khắc nghiệt. Nếu thiết bị định vị bị hỏng thì các nút cảm biến hoạt động dựa trên các thiết bị này sẽ trở nên vô dụng. Ngoài ra, chi phí đầu tư cho các thiết bị định vị cũng là khá cao. Bên cạnh đó, vấn đề

năng lượng tiêu thụ và kích thước của thiết bị định vị có thể không phù hợp đối với các nút cảm biến hoạt động bằng pin và được triển khai với số lượng lớn lên tới hàng ngàn nút.

#### 1.4. Giao thức cây thu thập dữ liệu CTP

Hầu hết các ứng dụng chính của mạng cảm biến không dây là thu thập thông tin cảm nhận được trong trường cảm biến nên các giao thức thu thập dữ liệu nhận được nhiều sự quan tâm nghiên cứu trong cộng đồng mạng cảm biến không dây.

Một trong những giao thức thu thập dữ liệu hiệu quả nhất cho mạng cảm biến không dây đó là giao thức cây thu thập dữ liệu - CTP (Collection Tree Protocol). Giao thức CTP được xây dựng dựa trên phương pháp tiếp cận định tuyến phẳng, tự tổ chức. Giao thức CTP cho phép truyền thông hiệu quả từ các nút mạng trong trường cảm biến đến một trong các nút gốc.



**Hình 1.8:** Cấu trúc liên kết mạng được xây dựng theo giao thức CTP.

Giao thức cây thu thập dữ liệu CTP thực thi cơ chế thu thập dữ liệu tin cậy từng bước nhảy (hop-by-hop). Các nút tự tổ chức thành một cấu trúc dạng cây và dữ liệu luôn được gửi về nút cha cho tới khi đến được đỉnh của cây (nút gốc). Nút gốc được gán là đỉnh của cây và tất cả các nút khác được khởi tạo là các nút lá. Các nút sẽ cập nhật vị trí của nó trong cây và quá trình này được mở rộng dần ra với điểm xuất phát ban đầu là từ nút gốc. Dữ liệu được gửi qua một cấu trúc cây đến nút gốc. Hình 1.8 minh họa một cấu trúc liên kết mạng được xây dựng theo giao thức CTP.

Trong quá trình xây dựng và duy trì cấu trúc cây định tuyến, các nút cần phải xác định thước đo định tuyến để lựa chọn nút lân cận tốt nhất (nút *cha*). Hiện tại, giao thức CTP sử dụng thước đo định tuyến chất lượng liên kết *ETX* để xây dựng cấu trúc cây định tuyến. Các nút cần phải thu thập thông tin về chất lượng liên kết của các nút lân cận và dựa vào thông tin đó để tính toán và lựa chọn nút *cha*. Để thực hiện điều này, các nút định kỳ trao đổi các bản tin điều khiển. Bản tin điều khiển mang thông tin về chất lượng tuyến đường (*rtmetric*) từ nút đó đến nút gốc.

#### 1.4.1. Thước đo định tuyến được sử dụng trong giao thức CTP

Để xây dựng và duy trì cấu trúc cây định tuyến thì cần phải xác định một thước đo định tuyến để các nút có thể lựa chọn tuyến đường tối ưu. Một số thước đo định tuyến có thể được sử dụng như số bước nhảy đến nút gốc, thuộc tính năng lượng của nút hoặc chất lượng liên kết...

Trong giao thức CTP hiện tại, thước đo định tuyến được sử dụng là số lần truyền kỳ vọng *ETX* (*Expected Transmission*) [1]. *ETX* của một liên kết là số lần truyền cần thiết để gửi thành công một bản tin từ nguồn đến đích qua liên kết đó bao gồm cả việc truyền lại. Hình 1.9 minh họa cách tính thước đo  $ETX_{link}$  của một liên kết.



**Hình 1.9:**  $ETX_{link}$  của một liên kết.

Thước đo  $ETX_{link}$  của một liên kết được xác định theo công thức sau:

$$ETX_{link} = \frac{1}{D_f \cdot D_b} \quad (1.1)$$

Trong đó:

- $D_f$ : Tỷ lệ chuyển phát bản tin theo chiều từ nút A đến nút B.
- $D_b$ : Tỷ lệ chuyển phát bản tin theo chiều ngược lại từ nút B đến A.

Thước đo *ETX* của một tuyến đường *rtmetric* (*route metric*) được xác định bằng tổng  $ETX_{link}$  của tất cả các liên kết trên toàn tuyến đường đó.

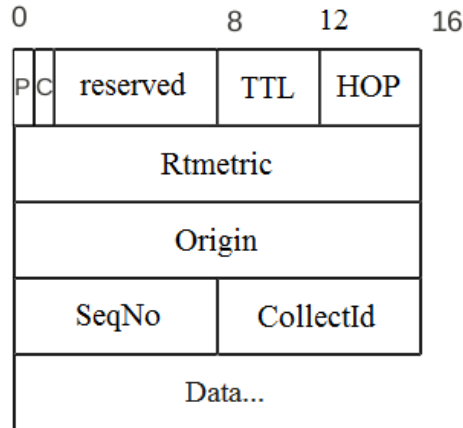
Thước đo *rtmetric* của mỗi nút được gửi quảng bá cho các nút lân cận thông qua việc trao đổi các bản tin điều khiển.

## 1.4.2. Cấu trúc các bản tin trong giao thức CTP

Giao thức CTP sử dụng 3 loại bản tin để xây dựng và duy trì hoạt động của cấu trúc liên kết mạng đó là: Bản tin điều khiển, bản tin dữ liệu và bản tin xác nhận.

### 1.4.2.1. Bản tin dữ liệu

Cấu trúc khung của bản tin dữ liệu (bản tin data) được minh họa ở hình 1.10.



**Hình 1.10:** Cấu trúc bản tin dữ liệu.

Các trường trong bản tin dữ liệu bao gồm:

- **Trường P - 1bit (routing Pull):** Bit P cho phép các nút yêu cầu thông tin định tuyến từ các nút khác. Nếu một nút có địa chỉ unicast trong cấu trúc khung của bản tin dữ liệu lắng nghe được một bản tin có bit P được thiết lập thì nó sẽ gửi đi một bản tin điều khiển trong khoảng thời gian gần nhất.
- **Trường C - 1bit (Congestion notification):** Bit thông báo tắc nghẽn. Nếu một nút loại bỏ một bản tin dữ liệu thì nó cần phải thiết lập trường C trong cấu trúc khung của bản tin dữ liệu kế tiếp mà nó gửi đi.
- **Trường reserved - 6bit:** Dự trữ.
- **Trường TTL - 4bit (Time To Live):** Trường này thể hiện số bước nhảy tối đa mà một bản tin dữ liệu có thể đi qua. Tại mỗi nút chuyển tiếp, trường TTL giảm đi một. Nếu trường TTL  $\leq 1$  thì bản tin dữ liệu bị loại bỏ.
- **Trường HOP - 4bit:** Đây là trường bộ đếm số bước nhảy. Khi một nút tạo ra một bản tin dữ liệu thì trường này được thiết lập bằng 0. Khi bản tin dữ liệu được chuyển tiếp bởi một nút trung gian thì trường này được tăng lên một. Trường này thể hiện số bước nhảy mà một bản tin dữ liệu đã đi qua trước khi đến được nút hiện tại.

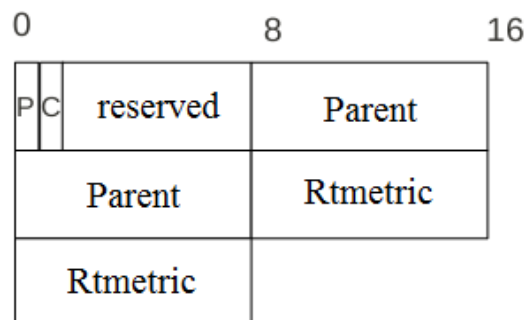


- **Trường Rtmetric - 16bit (route metric):** Là thước đo định tuyến *rtmetric* của nút gửi đơn chặng. Khi một nút gửi một bản tin dữ liệu thì nó phải gửi kèm theo giá trị *rtmetric* của nó được xác định thông qua điểm đến đơn chặng hiện tại. Nếu một nút nhận được một bản tin có *rtmetric* nhỏ hơn *rtmetric* của nó thì nó sẽ lập lịch một bản tin điều khiển trong một khoảng thời gian gần nhất. Đây cũng chính là cơ chế tránh vòng lặp định tuyến trong giao thức CTP.
- **Trường Origin - 16bit:** Trường này mang địa chỉ nút khởi nguồn của bản tin dữ liệu. Nút chuyển tiếp bản tin dữ liệu không được phép thay đổi trường này.
- **Trường Seqno - 8bit (sequence number):** Đây là trường số thứ tự bản tin dữ liệu. Nút khởi nguồn sẽ thiết lập trường này và nút thực hiện chuyển tiếp bản tin dữ liệu không được phép thay đổi trường này. Sự kết hợp hai trường (origin, seqno) sẽ xác định một bản tin dữ liệu duy nhất trong mạng.
- **Trường CollectId - 8bit:** Trường nhận dạng giao thức lớp cao hơn. Nút khởi nguồn thiết lập trường này và nút thực hiện chuyển tiếp bản tin dữ liệu không được thay đổi trường này.
- **Trường Data:** Đây là trường tải dữ liệu. Các nút chuyển tiếp bản tin dữ liệu không được phép chỉnh sửa trường này.

#### 1.4.2.2. Bản tin điều khiển

Cấu trúc khung của bản tin điều khiển được minh họa ở hình 1.11. Các trường trong bản tin điều khiển bao gồm:

- **Trường P - 1bit:** Tương tự như trường P trong cấu trúc khung của bản tin dữ liệu nhưng có một điểm khác đó là bản tin điều khiển được phát quảng bá. Do đó, sẽ có nhiều nút lân cận trả lời lại bit P trong cấu trúc khung của bản tin điều khiển.



**Hình 1.11:** Cấu trúc bản tin điều khiển.

- **Trường C - 1 bit (Congestion notification):** Thông báo tắc nghẽn. Nếu một nút loại bỏ một bản tin dữ liệu thì nó sẽ thiết lập trường C trong cấu trúc khung của bản tin điều khiển tiếp theo mà nút sẽ gửi đi.
- **Trường reserved - 6bit:** Dự trữ. Các bit dự trữ này sẽ được sử dụng trong giao thức EACTP.
- **Trường Parent - 16bit:** Địa chỉ nút cha hiện tại.
- **Trường rtmetric - 16bit:** Giá trị thước đo định tuyến hiện tại của nút.

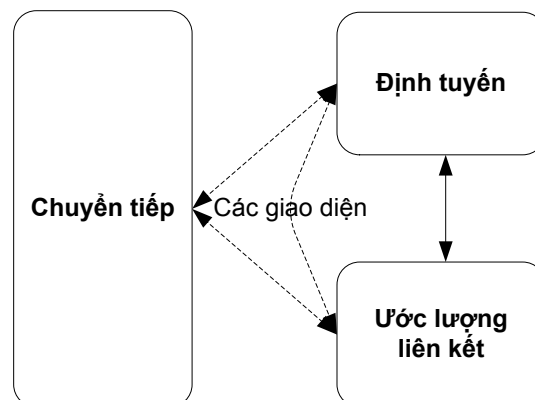
Khi một nút lắng nghe được một bản tin điều khiển thì nó sẽ cập nhật lại bảng định tuyến. Nếu giá trị *rtmetric* của một nút thay đổi thì nó sẽ gửi quảng bá một bản tin điều khiển để thông báo cho các nút lân cận biết. Trường Parent (nút cha) cũng chính là trường đích đến đơn chặng của bản tin dữ liệu. Khi nút cha lắng nghe được một bản tin điều khiển từ nút con có giá trị *rtmetric* nhỏ hơn giá trị *rtmetric* của nó thì nút cha sẽ lập lịch để gửi đi một bản tin điều khiển trong khoảng thời gian gần nhất.

#### 1.4.2.3. Bản tin xác nhận

Bản tin xác nhận ACK được sử dụng để thông báo sự tiếp nhận thành công một bản tin dữ liệu. Bởi vì giao thức CTP sử dụng cơ chế xác nhận tại lớp liên kết dữ liệu nên các bản tin ACK chỉ chứa các thông tin lớp vật lý (PHY) và lớp điều khiển truy nhập kênh truyền (MAC).

#### 1.4.3. Các thành phần chính của giao thức CTP

Hình 1.12 biểu diễn các thành phần chính của giao thức CTP. Giao thức CTP bao gồm ba thành phần chính đó là:



**Hình 1.12:** Các thành phần chính của giao thức CTP.

- **Phần định tuyến:** Có nhiệm vụ gửi và nhận các bản tin điều khiển cũng như tạo và cập nhật bảng định tuyến. Bảng định tuyến của mỗi nút lưu các nút lân

cận có thể được lựa chọn làm nút cha của nó. Bảng này được xây dựng dựa vào các thông tin lấy từ các bản tin điều khiển. Bảng này lưu thông tin về địa chỉ của các nút lân cận cũng như các thông tin khác như thước đo định tuyến của mỗi nút.

- **Phần chuyển tiếp:** Có nhiệm vụ chuyển tiếp các bản tin dữ liệu đến từ lớp ứng dụng cũng như các bản tin dữ liệu đến từ các nút lân cận. Phần chuyển tiếp cũng có nhiệm vụ phát hiện và sửa chữa các vòng lặp định tuyến cũng như ngăn chặn sự xung đột của các bản tin.
- **Phần ước lượng liên kết:** Có nhiệm vụ xác định chất lượng liên kết ( $ETX_{link}$ ). Phần ước lượng liên kết tính toán  $ETX_{link}$  dựa vào sự thống kê các bản tin điều khiển nhận được và các bản tin dữ liệu được truyền thành công. Dựa vào sự thống kê này, phần ước lượng liên kết sẽ tính toán  $ETX_{link}$  theo chiều đến được xác định bằng tổng số bản tin điều khiển được gửi bởi nút lân cận chia cho tổng số bản tin điều khiển nhận được. Tương tự như vậy, giá trị  $ETX_{link}$  theo chiều đi được xác định bằng tổng số bản tin dữ liệu được gửi đi (bao gồm cả việc truyền lại) chia cho tổng số bản tin dữ liệu được xác nhận bởi bản tin xác nhận ACK.

#### 1.4.4. Điểm yếu của giao thức CTP

Trong thời gian gần đây, giao thức CTP cũng thu hút được nhiều sự quan tâm nghiên cứu trong cộng đồng mạng cảm biến [2], [3], [4], [5], [6], [7], [8]. Giao thức CTP đã được chứng minh là một giao thức thu thập dữ liệu đạt hiệu quả cao về mặt năng lượng tiêu thụ cũng như tỷ lệ chuyển phát thành công bản tin dữ liệu trong mạng ở mức cao. Trong bài báo [5], các tác giả đã đưa ra một số kết quả đánh giá hiệu năng mạng đối với giao thức CTP. Các kết quả đánh giá cho thấy giao thức CTP đạt được bốn mục tiêu chính đó là: Độ tin cậy cao, khả năng chống lỗi tốt, hiệu quả về năng lượng và độc lập với nhiều kiến trúc phần cứng khác nhau.

Tuy nhiên, giao thức CTP hiện tại không có sự nhận thức về mức năng lượng còn lại trên các nút mạng. Giao thức CTP chỉ dựa vào thước đo định tuyến  $ETX$  để lựa chọn tuyến đường tối ưu. Thước đo định tuyến  $ETX$  không giải quyết được vấn đề cân bằng năng lượng giữa các nút mạng. Bởi vậy, giao thức CTP dễ bị mất cân bằng năng lượng. Các nút mạng thuộc tuyến đường tối ưu phải thực hiện nhiều việc truyền dẫn hơn các nút khác. Chúng sẽ hết năng lượng nhanh hơn các nút khác và tạo thành các lỗ hổng trong mạng, làm giảm hiệu năng của toàn bộ hệ thống mạng. Đây là một

trong những thách thức quan trọng trong các mạng cảm biến không dây hoạt động bằng pin. Các bài báo [9], [10], [11], [12] đã chỉ ra rằng thời gian sống của mạng sẽ được kéo dài nếu năng lượng tiêu thụ trên toàn mạng đồng đều.

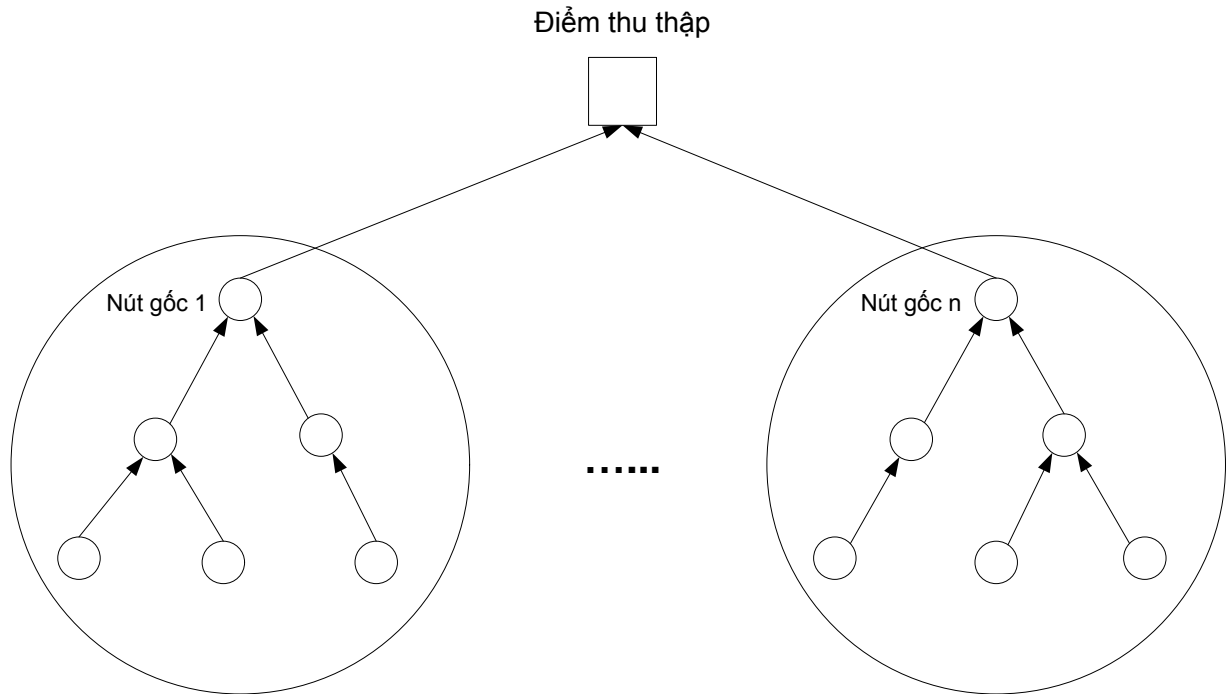
Một số kết quả đánh giá giao thức CTP hiện tại dựa trên công cụ mô phỏng Cooja [13] và thực nghiệm với phần cứng TUMote cũng cho các kết quả tương tự. Trong luận án này, tác giả đề xuất một giao thức định tuyến mới EACTP (Energy Aware Collection Tree Protocol) có sự nhận thức về năng lượng nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng.

### **1.5. Bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng**

Bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng (bài toán định tuyến EACTP) được đặt trong các giả thiết sau:

- Các nút mạng không đồng nhất và trong mạng có hai loại nút đó là các nút cảm biến và các nút gốc. Các nút cảm biến sử dụng nguồn năng lượng bằng pin và có khả năng xử lý cũng như có bộ nhớ hạn chế. Các nút gốc có nguồn năng lượng, khả năng lưu trữ và tính toán tốt hơn các nút cảm biến. Các nút này đóng vai trò là các nút chủ cụm để chuyển tiếp lưu lượng từ các nút cảm biến đến điểm thu thập.
- Các nút cảm biến đọc và gửi dữ liệu về các nút gốc bằng kỹ thuật truyền đa chặng thông qua các nút cảm biến trung gian khác. Các nút gốc có nhiệm vụ thu thập dữ liệu và gửi trực tiếp dữ liệu về điểm thu thập. Quá trình này được lặp đi lặp lại cho đến khi các nút hết năng lượng, làm hỏng các kết nối mạng và làm cho mạng không thể hoạt động được nữa.
- Trong suốt toàn bộ quá trình hoạt động của mạng, các nút truyền cùng một mức công suất không đổi. Không có sự tổng hợp dữ liệu (data aggregation) nào được thực hiện trong mạng. Tất cả các dữ liệu thu thập bởi các nút gốc đều được gửi tới điểm thu thập.
- Mạng cảm biến không dây bao gồm nhiều nút mạng được phân bố trên một vùng triển khai được xem là phẳng (mạng 2D).
- Mạng được chia thành nhiều cụm nhỏ khác nhau.

- Các nút cảm biến phát sóng đẳng hướng. Các liên kết được giả thiết là đối xứng. Trong thực tế, các nút cảm biến có thể được trang bị loại anten đẳng hướng.
- Các nút cảm biến là cố định. Mạng được xem là tĩnh.



**Hình 1.13:** Cấu trúc liên kết mạng được xét đến trong bài toán định tuyến EACTP.

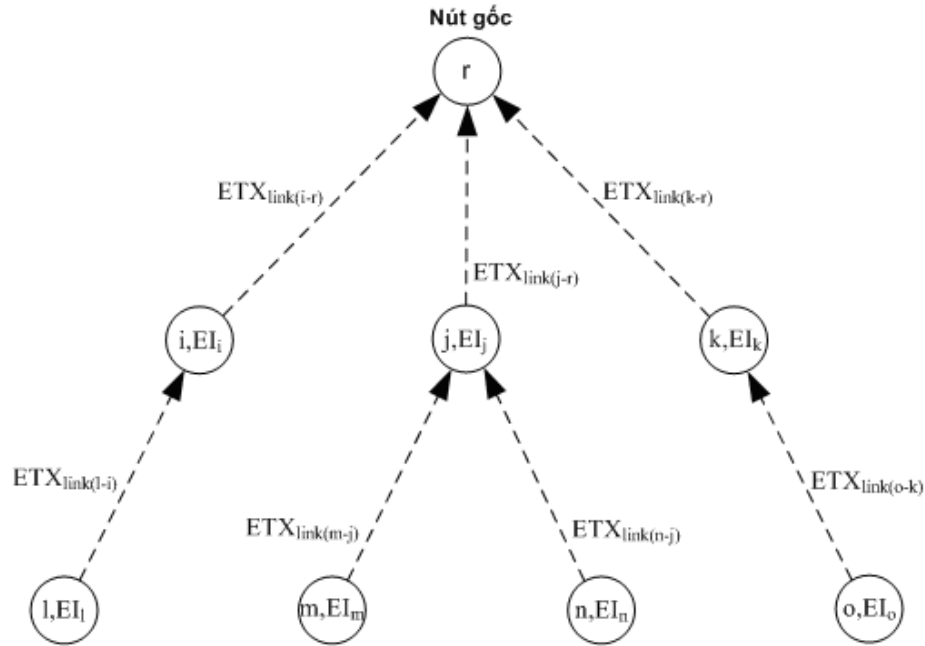
Hình 1.13 minh họa mô hình cấu trúc liên kết mạng được tác giả xét đến trong luận án. Mạng được chia thành nhiều cụm nhỏ khác nhau. Do thuật toán để xây dựng cấu trúc cây trong các cụm nhỏ này là hoàn toàn giống nhau nên tác giả chỉ xét đến việc tối ưu hóa thời gian sống của một cụm trong mạng.

**Bài toán định tuyến EACTP** được phát biểu như sau:

Cho một cấu trúc mạng  $G = (V, E)$  là một đồ thị, trong đó  $V$  là số đỉnh,  $E$  là số cạnh. Bài toán định tuyến EACTP đặt ra đó là tìm được một cấu trúc cây tối ưu dựa trên hai tham số là chất lượng liên kết giữa các nút mạng và trạng thái năng lượng còn lại của các nút mạng.

Hình 1.14 là ví dụ minh họa một cấu trúc hình học của bài toán định tuyến EACTP. Mỗi đỉnh thuộc cây biểu diễn một nút mạng có cả chức năng sinh dữ liệu và chức năng định tuyến. Mỗi nút mạng xác định được chỉ số năng lượng còn lại  $EI$  (Energy Indicator) của nó.

Mỗi cạnh thuộc cây là một tuyến (link) kết nối giữa hai nút mạng  $(u, v)$  bất kỳ được biểu diễn bởi chất lượng liên kết giữa hai nút mạng theo tham số chất lượng liên kết ( $ETX_{link}$ ).



**Hình 1.14:** Ví dụ minh họa cấu trúc hình học của bài toán định tuyến EACTP.

Gọi độ dài đường đi P nối từ một điểm bất kỳ trên  $G = (V, E)$  đến nút gốc được xác định bằng tổng  $ETX_{link}$  của tất cả các tuyến kết nối thuộc tuyến đường đó. Mỗi nút được gán một nhãn ( $rtmetric$ ) chính là độ dài đường đi tốt nhất từ nút đó đến nút gốc được xác định theo thước đo định tuyến chất lượng liên kết  $ETX$ . Bài toán định tuyến EACTP được phát biểu như sau: Tìm nút  $n$  thuộc tập hợp các nút lân cận  $N$  của nút nguồn  $s$  trên cấu trúc  $G = (V, E)$  sao cho nút  $n$  thỏa mãn các điều kiện sau:

$$\begin{cases} rtmetric_s = \arg \min_{n \in N} (rtmetric_n + ETX_{link(s-n)}) \\ n.EI \geq Threshold \end{cases} \quad (1.2)$$

Trong đó: -  $rtmetric_n$  là nhãn của nút  $n$ .

- $rtmetric_s$  là nhãn của nút  $s$  được xác định theo nút  $n$ .
- $ETX_{link(s-n)}$  là chất lượng liên kết giữa nút  $s$  và nút  $n$ .
- $n.EI$  là chỉ số năng lượng còn lại của nút  $n$ .
- $Threshold$  là ngưỡng chỉ số năng lượng còn lại.

Bài toán định tuyến EACTP được xuất phát từ đề tài thực tế đang được triển khai tại Viện nghiên cứu Điện tử, Tin học và Tự động hóa đó là:

- Nhiệm vụ khoa học công nghệ cấp nhà nước về “*Nghiên cứu thiết kế, chế tạo hệ thống quan trắc ô nhiễm nước tự động, lưu động*”, mã số 07.12/CNMT: Tại Việt Nam tình trạng ô nhiễm môi trường do nước thải công nghiệp gây ra đang ở mức báo động. Môi trường nước ở nhiều đô thị, khu công nghiệp và làng nghề ngày càng bị ô nhiễm. Tình trạng ô nhiễm nước có thể thấy rõ không chỉ ở Hà Nội, thành phố Hồ Chí Minh mà ở các thành phố khác cũng có ô nhiễm nước thải công nghiệp nặng nề. Khu công nghiệp Thái Nguyên thải nước biến Sông Cầu thành màu đen, mặt nước sủi bọt trên chiều dài hàng chục cây số. Khu công nghiệp Việt Trì xả mỗi ngày hàng ngàn mét khối nước thải của nhà máy hoá chất, thuốc trừ sâu, giấy, dệt... xuống Sông Hồng làm nước bị nhiễm bẩn đáng kể. Khảo sát một số làng nghề sắt thép, đúc đồng, nhôm, chì, giấy, dệt nhuộm ở Bắc Ninh cho thấy có lượng nước thải hàng ngàn m<sup>3</sup>/ngày không qua xử lý, gây ô nhiễm nguồn nước và môi trường trong khu vực. Hệ thống quan trắc ô nhiễm nước tự động cho phép giám sát thường xuyên tình trạng ô nhiễm nước thải từ các khu công nghiệp, phát hiện và cảnh báo kịp thời cho các cơ quan chức năng để có biện pháp xử lý ngăn ngừa ô nhiễm.
- Nhiệm vụ khoa học công nghệ cấp nhà nước về “*Nghiên cứu thiết kế, chế tạo hệ thống quan trắc lưu lượng dòng chảy và lượng mưa hỗ trợ điều tiết an toàn hệ thống hồ chứa nước Sông Đà*”, mã số ĐTĐL.2011-G/48: Hệ thống hồ nước Sông Đà là nơi điều tiết, chống lũ cho lưu vực sông Hồng nói chung và thủ đô Hà Nội nói riêng, hơn nữa nó còn có các mục tiêu khác là phát điện tại các nhà máy thủy điện trên Sông Đà, cấp nước phục vụ cho nông nghiệp và các nhu cầu khác phục vụ phát triển kinh tế xã hội. Hai chức năng chống lũ và phát điện của công trình thủy điện có mối liên quan mật thiết với nhau. Để điều tiết an toàn hồ chứa nước của các công trình thủy điện, người ta cần theo dõi/dự báo một số thông số chính như: Mức nước, lưu lượng dòng chảy, lượng mưa, độ thấm thấu và độ biến dạng của đập... Các thông số này cần được cung cấp một cách chính xác, kịp thời để phục vụ công tác điều tiết an toàn hồ chứa nước Sông Đà.

Trong hai đề tài này, các nút cảm biến thường được triển khai ở những vị trí xa nguồn điện lưới hoặc ở những vị trí gặp nhiều khó khăn trong việc kéo đường

dây điện từ nguồn điện lưới. Vì vậy, các nút mạng cảm biến thường sử dụng những nguồn năng lượng tích trữ (VD: Pin, Ắc quy). Vì vậy, một trong những nhiệm vụ cấp thiết đặt ra trong các đề tài này đó là: *Cần phải đề xuất một giao thức truyền thông thu thập dữ liệu có sự nhận thức về năng lượng nhằm tăng thời gian sống của các nút mạng sử dụng những dạng nguồn năng lượng tích trữ này.*

## **1.6. Hiện trạng nghiên cứu và phương pháp tiếp cận bài toán định tuyến có sự nhận thức về năng lượng trong các nghiên cứu trước đây**

Từ trước đến nay, có nhiều giao thức định tuyến dựa trên sự nhận thức về năng lượng đã được đề xuất. Trong phần này, tác giả sẽ giới thiệu về một số công trình nghiên cứu có liên quan.

### **1.6.1. Định tuyến với tổng năng lượng tối thiểu MTPR (Minimal Total Power Routing)**

Một trong những đề xuất ban đầu cho các thước đo định tuyến liên quan đến năng lượng đó là tối thiểu năng lượng tiêu thụ trên mỗi bản tin. Singh và cộng sự [34] đã thực hiện ý tưởng này như sau: Gọi  $e_{i,j}$  biểu thị năng lượng tiêu thụ để truyền một bản tin từ nút  $i$  đến một nút  $j$  lân cận. Nếu bản tin phải đi qua một tuyến đường  $P$  bao gồm các nút  $n_1, \dots, n_k$  thì tổng năng lượng  $E$  cần thiết để truyền bản tin là:

$$E = \sum_{i=1}^{k-1} e_{n_i, n_{i+1}} \quad (1.3)$$

Trong một tập  $P$  các tuyến đường có thể, tuyến đường được lựa chọn là tuyến đường có tổng năng lượng tiêu thụ là nhỏ nhất.

Một nhược điểm của thước đo định tuyến MTPR đó là nó không cung cấp thông tin về thời gian sống còn lại của pin. Tuyến đường tìm được có năng lượng tiêu thụ trên mỗi bản tin là nhỏ nhất nhưng lại có thể dẫn đến việc một số nút mạng sẽ hết năng lượng nhanh hơn các nút mạng khác [49].

### **1.6.2. Định tuyến với chi phí nguồn pin nhỏ nhất (Minimum Battery Cost Routing)**

Để đảm bảo sự cân bằng năng lượng tiêu thụ trên tất cả các nút trong mạng và giải quyết nhược điểm của thước đo định tuyến MTPR thì dung lượng pin còn lại trên mỗi nút cảm biến được quan tâm đến trong thước đo định tuyến MBCR [35]. Thước đo MBCR dựa vào dung lượng pin còn lại trên mỗi nút cảm biến. Mỗi nút cảm biến



được gán một chi phí. Chi phí này được tính bằng nghịch đảo dung lượng pin còn lại trên nút cảm biến. Trong một tập P các tuyến đường có thể, tuyến đường được lựa chọn là tuyến đường có tổng chi phí là nhỏ nhất.

Nhược điểm của thước đo định tuyến MBCR đó là tuyến đường được lựa chọn có thể bao gồm một số nút mạng có dung lượng pin còn lại ở mức thấp. Những nút mạng này sẽ hết năng lượng nhanh hơn các nút mạng khác [49].

### **1.6.3. Giao thức định tuyến nhận thức về năng lượng EAR (Energy Aware Routing)**

Giao thức định tuyến có sự nhận thức về năng lượng EAR [36] được đề xuất bởi Rahul C. Shah và các cộng sự. Giao thức này duy trì một tập các tuyến đường tốt thay vì chỉ lựa chọn một tuyến đường tối ưu. Giao thức EAR sử dụng thước đo năng lượng để xác định các tuyến đường tốt. Thước đo năng lượng này được xác định bởi cả chi phí chuyển phát một bản tin và năng lượng còn lại của các nút chuyển tiếp. Tuy nhiên giao thức EAR tồn tại hai nhược điểm chính đó là: Thứ nhất, giao thức EAR dựa vào năng lượng còn lại của cả tuyến đường mà bỏ qua sự khác nhau về năng lượng của từng nút riêng lẻ trên tuyến đường. Một tuyến đường với năng lượng còn lại nhiều không có nghĩa là tất cả các nút trên tuyến đường đó còn nhiều năng lượng. Thứ hai, giao thức EAR xác định thước đo chi phí năng lượng dựa trên sự hỗ trợ về phần cứng trên các nút cảm biến. Tuy nhiên, không phải mọi phần cứng đều hỗ trợ khả năng này [7].

### **1.6.4. Giao thức định tuyến E-Span (Energy-Aware Spanning Tree Algorithm)**

Trong bài báo [37], các tác giả đã đề xuất giao thức định tuyến E-Span. Trong đó, nút có năng lượng còn lại ở mức cao nhất sẽ được chọn làm nút gốc. Các nút khác sẽ lựa chọn nút cha trong số các nút lân cận dựa trên mức năng lượng còn lại và số bước nhảy đến nút gốc.

Nhược điểm chính của giao thức định tuyến này đó là tuyến đường tối ưu được lựa chọn dựa vào thước đo định tuyến số bước nhảy. Thước đo định tuyến này không xét đến hiệu suất của từng liên kết trên tuyến đường từ nguồn đến đích. Mạng cảm biến không dây bao gồm nhiều liên kết tổn hao (lossy). Vì vậy, các tuyến đường với số bước nhảy tối thiểu hoàn toàn có thể bao gồm những liên kết tổn hao và điều này làm giảm thông lượng của mạng. Bản tin có thể phải truyền lại nhiều lần gây lãng phí về năng lượng và làm tăng độ trễ truyền bản tin [49].

### **1.6.5. Giao thức định tuyến có sự nhận thức về năng lượng và cân bằng tải**

Trong bài báo [38], các tác giả đã đề xuất giao thức định tuyến có sự nhận thức về năng lượng và đảm bảo sự cân bằng tải trong mạng. Các tác giả đề xuất cơ chế ước lượng năng lượng tiêu thụ dựa vào các hoạt động (truyền, nhận) của bộ thu phát vô tuyến. Bước nhảy kế tiếp được lựa chọn là nút có năng lượng còn lại ở mức cao nhất. Một nút mạng được xác định là quá tải dựa trên số lượng các bản tin truyền, nhận của bộ thu phát vô tuyến.

Nhược điểm của giao thức định tuyến này đó là tuyến đường tối ưu được lựa chọn dựa vào năng lượng còn lại trên nút chuyển tiếp. Thước đo định tuyến này cũng có nhược điểm giống như thước đo định tuyến số bước nhảy khi không xét đến chất lượng liên kết giữa các nút mạng. Điều này có thể dẫn đến việc truyền lại bản tin nhiều lần, gây ra sự lãng phí năng lượng và sự quá tải của các nút trong mạng. Ngoài ra, mô hình năng lượng tiêu thụ của nút cảm biến được sử dụng trong giao thức này cũng chỉ xét đến năng lượng tiêu thụ của bộ thu phát vô tuyến mà chưa kể đến các thành phần tiêu thụ năng lượng khác.

### **1.6.6. Giao thức định tuyến BRE (Bursty Routing Extensions)**

Trong bài báo [23], các tác giả quan sát thấy rằng các giao thức định tuyến như giao thức CTP có khuynh hướng sử dụng lâu dài các liên kết ổn định chứ không phải là các liên kết không ổn định. Các tác giả cho rằng các liên kết có chất lượng thay đổi nhanh chóng theo thời gian cần phải được sử dụng trong quá trình định tuyến. Các tác giả cũng chỉ ra rằng vấn đề cân bằng tải giữa các nút trong mạng rất quan trọng. Tất cả các giao thức định tuyến như BRE, CTP đều tập trung vào việc giảm số lần truyền dẫn. Sự mất cân bằng tải có thể dẫn đến việc một số nút ngừng hoạt động nhanh hơn mong muốn và làm giảm hiệu năng của toàn mạng.

### **1.6.7. Giao thức định tuyến BCTP (Balanced Collection Tree Protocol)**

Trong bài báo [7], các tác giả đã xác định một điểm yếu cần khắc phục đối với giao thức CTP hiện tại đó là sự mất cân bằng tải giữa các nút trong mạng. Vấn đề cân bằng tải là rất quan trọng đối với mạng cảm biến không dây bởi vì sự mất cân bằng tải có thể tạo ra một số nút điểm nóng. Các nút này sẽ hết năng lượng nhanh hơn dự kiến.

Các tác giả đã đề xuất giao thức BCTP (Balanced Collection Tree Protocol) cải tiến. Giao thức BCTP được cải tiến dựa trên giao thức CTP ở hai điểm: Thứ nhất, BCTP thêm một thành phần ghi lại sự truyền dẫn để xác định một nút có phải là quá

tải hay không; Thứ hai, BCTP thay đổi tiêu chí định tuyến đối với các nút con của một nút đã quá tải nhằm giảm tải cho nút đó. Giao thức BCTP cho phép giảm tải cho các nút điểm nóng (nút có lưu lượng dữ liệu lớn). Giao thức BCTP sử dụng tốc độ truyền dẫn trung bình như là một thước đo để xác định tải lưu lượng của một nút. Mỗi khi một nút được xác định là quá tải thì giao thức BCTP sử dụng một chiến lược định tuyến ngẫu nhiên để đảm bảo sự cân bằng tải. Giao thức BCTP đã được thử nghiệm với 9 nút TelosB. Kết quả thực nghiệm cho thấy giao thức BCTP có thể giảm tải của nút điểm nóng tới 61,9% trong một mạng được triển khai với mật độ dày.

#### **1.6.8. Giao thức định tuyến ICTP (Improved Collection Tree Protocol)**

Trong bài báo [8], các tác giả đã chỉ ra rằng giao thức CTP hiện tại chỉ quan tâm đến việc lựa chọn tuyến đường tối ưu theo thước đo *ETX* và điều này dẫn đến ba nhược điểm chính đối với giao thức CTP hiện tại đó là: Thứ nhất, đây là nguyên nhân dẫn đến hiện tượng rung pha (jitter) trong mạng. Hiện tượng này được lý giải là nếu tại một thời điểm nào đó tuyến đường bị tắc nghẽn thì mạng sẽ lựa chọn một tuyến đường khác để tiếp tục gửi bản tin cho đến khi xảy ra sự tắc nghẽn đối với tuyến đường mới. Điều này sẽ gây ra hiện tượng jitter trong mạng; Nhược điểm thứ hai đó là các nút mạng trong tuyến đường tối ưu sẽ bị mất nhiều năng lượng hơn các nút khác trong mạng. Điều này không chỉ dẫn đến việc mất cân bằng năng lượng giữa các nút trong mạng mà còn dẫn đến việc giảm thời gian sống của mạng; Nhược điểm thứ ba đó là mặc dù thước đo *ETX* được sử dụng trong giao thức CTP có thể phát hiện được hiện tượng tắc nghẽn trong mạng để từ đó lựa chọn một tuyến đường mới nhưng hiện tượng tắc nghẽn chắc chắn sẽ gặp phải trong mạng và điều này có thể sẽ làm giảm hiệu năng của mạng. Các tác giả đã đề xuất giao thức ICTP (Improved Collection Tree Protocol) cải tiến. Trong giao thức ICTP, các tác giả sử dụng hàm xác suất để xác định nút cha.

Giao thức ICTP tồn tại ba nhược điểm chính: Thứ nhất, việc lựa chọn tuyến đường dựa trên xác suất có thể dẫn đến việc lựa chọn tuyến đường có chất lượng xấu và điều này dẫn đến việc gửi lại bản tin nhiều lần và làm cho năng lượng tiêu thụ trong toàn mạng tăng lên; Thứ hai, giao thức ICTP làm phát sinh thêm chi phí năng lượng trong việc gửi thông tin về năng lượng còn lại trên các nút cảm biến; Thứ ba, cơ chế ước lượng năng lượng tiêu thụ trong giao thức ICTP còn đơn giản bởi vì các tác giả mới chỉ xét đến thành phần tiêu thụ năng lượng chính là bộ thu phát vô tuyến. Các tác giả đã giả định rằng: Năng lượng ban đầu của mỗi nút cảm biến là 0xFFFF;

Năng lượng tiêu thụ khi gửi một bản tin điều khiển là 20; Năng lượng tiêu thụ khi gửi một bản tin dữ liệu là 30. Mô hình năng lượng này không phù hợp khi thực thi trên các nút cảm biến thực tế. Các kết quả đánh giá hiệu năng giao thức ICTP mới chỉ dựa trên mô phỏng.

#### **1.6.9. Giao thức định tuyến EQLR (Energy and Link Quality Based Routing Tree)**

Trong bài báo [50], các tác giả đã xác định một nhược điểm của giao thức CTP hiện tại đó là các nút có chất lượng liên kết tốt thường được lựa chọn làm nút cha. Do vậy, các nút này thường hết năng lượng nhanh hơn so với các nút mạng khác. Các tác giả đã đề xuất giao thức định tuyến EQLR. Giao thức EQLR xây dựng cấu trúc cây thu thập dữ liệu dựa trên hai thước đo định tuyến là chất lượng liên kết và năng lượng pin còn lại của mỗi nút cảm biến. Cơ chế ước lượng năng lượng tiêu thụ trong giao thức EQLR dựa trên đề xuất của Adam Dunkels [29]. Các tác giả đã so sánh đánh giá giao thức EQLR và giao thức CTP dựa trên mô phỏng và thực nghiệm với 9 nút cảm biến. Kết quả mô phỏng và thực nghiệm cho thấy giao thức EQLR đạt được sự cân bằng tải và thời gian sống của mạng tốt hơn so với giao thức CTP.

Tuy nhiên, giao thức EQLR vẫn còn tồn tại một số nhược điểm: Thứ nhất, bản tin điều khiển cần phải bổ sung thêm trường năng lượng còn lại để mang thông tin về mức năng lượng còn lại của pin. Điều này sẽ làm phát sinh thêm chi phí về năng lượng trong việc gửi bản tin điều khiển; Thứ hai, ngưỡng năng lượng để xác định một nút hết năng lượng là một giá trị cố định. Để xác định một ngưỡng năng lượng cố định tối ưu là rất khó; Thứ ba, thuật toán lựa chọn tuyến đường dựa trên ngưỡng *ETX*. Trong bài báo, các tác giả chọn ngưỡng *ETX* thay đổi từ 50 - 500. Ngưỡng này được xác định dựa trên thực nghiệm với 9 nút và giá trị lớn nhất đo được của *ETX* là 500. Tuy nhiên, trong triển khai thực tế với số lượng nút lớn thì việc xác định ngưỡng *ETX* dựa trên thực nghiệm sẽ gặp nhiều khó khăn.

#### **1.6.10. Giao thức định tuyến ELR (Energy Aware and Link Quality Based Routing Protocol)**

Trong bài báo [51], các tác giả đã đề xuất giao thức định tuyến ELR. Giao thức ELR xây dựng cấu trúc cây định tuyến dựa trên thước đo về chất lượng liên kết và phần trăm năng lượng còn lại của các nút cảm biến. Cơ chế ước lượng năng lượng tiêu thụ trong giao thức EQLR dựa trên đề xuất của Adam Dunkels [29]. Các tác giả đã so sánh đánh giá giao thức ELR với giao thức CTP. Kết quả đánh giá mô phỏng

cho thấy, thời gian sống của mạng khi hoạt động theo giao thức ELR tốt hơn so với thời gian sống của mạng khi hoạt động theo giao thức CTP ban đầu.

Tuy nhiên, giao thức ELR vẫn còn tồn tại một số nhược điểm: Thứ nhất, bản tin điều khiển được mở rộng thêm 16 bit để mang thông tin về mức năng lượng còn lại của nút cảm biến. Điều này sẽ làm phát sinh thêm chi phí về năng lượng trong việc gửi bản tin điều khiển; Thứ hai, ngưỡng năng lượng được thiết lập là một giá trị cố định 10%. Để xác định một ngưỡng năng lượng tối ưu là rất khó. Các tác giả cũng chưa phân tích rõ về ngưỡng năng lượng này; Thứ ba, ngưỡng  $ETXdiffTh$  là một giá trị cố định và được xác định bằng 10. Các tác giả cũng chưa phân tích rõ giá trị ngưỡng cố định này. Trong triển khai thực tế, việc xác định ngưỡng  $ETXdiffTh$  là một khó khăn cần được giải quyết.

#### **1.6.11. Giao thức định tuyến EARBB (Energy Aware Routing Based on Beaconing)**

Trong bài báo [52], các tác giả đã chỉ ra rằng giao thức CTP hiện tại là một giao thức thu thập dữ liệu hiệu quả về năng lượng và đáng tin cậy. Tuy nhiên, các tác giả cũng chỉ ra nhược điểm của giao thức CTP hiện tại đó là giao thức này không quan tâm đến sự cân bằng năng lượng của các nút trong mạng và không hỗ trợ lưu lượng theo chiều hướng xuống (downstream). Các tác giả đã đề xuất một giao thức định tuyến EARBB mới. Ngoài cơ chế định tuyến từ các nút đến nút gốc (node-to-sink) thì giao thức EARBB còn hỗ trợ cả cơ chế định tuyến từ nút đến nút (node-to-node). Các kết quả mô phỏng cho thấy giao thức EARBB thiết lập được cấu trúc liên kết mạng tin cậy và có thể nhanh chóng khắc phục lỗi. Các tác giả đã đề xuất một thước đo định tuyến ETHER mới. Thước đo định tuyến ETHER cho một bước nhảy được xác định như sau:

$$ETHER = E^\alpha \cdot L \quad (1.4)$$

Trong đó: E là năng lượng còn lại của nút cảm biến;  $\alpha$  được chọn bằng 1,5; L là tỷ lệ mất mát bản tin của kênh truyền. Thước đo định tuyến ETHER của tuyến đường bằng tổng thước đo định tuyến ETHER của các liên kết thuộc tuyến đường đó. Thước đo định tuyến ETHER có tính đến cả chất lượng liên kết và năng lượng còn lại của nút cảm biến. Do vậy, tuyến đường với năng lượng còn lại sẽ không được lựa chọn làm tuyến đường tối ưu.

Tuy nhiên, giao thức EARBB được đề xuất trong bài báo này vẫn còn tồn tại một số nhược điểm cần khắc phục: Thứ nhất, các tác giả chưa đề cập đến cách xác

định năng lượng còn lại của nút cảm biến; Thứ hai, mô hình năng lượng được sử dụng trong mô phỏng còn đơn giản khi các tác giả sử dụng số lượng bản tin gửi và nhận làm thước đo năng lượng; Thứ ba, các tác giả chọn hằng số  $\alpha = 1,5$  nhưng không có sự giải thích đầy đủ cho việc lựa chọn giá trị này; Thứ tư, tuyến đường tối ưu được lựa chọn dựa trên tổng thước đo định tuyến ETHER của các liên kết trên toàn tuyến đường. Như vậy, giao thức này hoàn toàn có thể lựa chọn phải tuyến đường bao gồm một số nút mạng có dung lượng pin còn lại ở mức thấp.

### 1.7. Giải pháp tiếp cận bài toán trong luận án

Trong luận án này, tác giả đề xuất giải pháp định tuyến EACTP nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Giao thức EACTP được cải tiến dựa trên giao thức CTP ở ba điểm:

- Thứ nhất, giao thức EACTP **bổ sung thêm thành phần ước lượng năng lượng** tiêu thụ của nút cảm biến. Cơ chế ước lượng năng lượng này hoàn toàn dựa trên phần mềm mà không cần sự bổ sung thêm bất kỳ một chi phí nào về phần cứng. Trong mô hình năng lượng đề xuất, tác giả có xét đến các thành phần tiêu thụ năng lượng chính trong cấu trúc phần cứng của một nút mạng cảm biến không dây.
- Thứ hai, giao thức EACTP bổ sung thêm một thước đo định tuyến mới đó là **trạng thái năng lượng còn lại ES (Energy State)** của mỗi nút cảm biến. Tác giả đã sử dụng ngưỡng năng lượng còn lại thích ứng và mỗi nút sẽ có một số trạng thái năng lượng còn lại. Các trạng thái năng lượng này được mã hóa bởi hai bit và được phát quảng bá cho các nút lân cận sử dụng hai bit dự trữ trong bản tin điều khiển. Vì vậy, giải pháp này không phát sinh thêm các chi phí mới về năng lượng trong việc gửi các bản tin điều khiển.
- Thứ ba, giao thức EACTP **bổ sung tiêu chí định tuyến** nhằm đảm bảo sự cân bằng năng lượng tiêu thụ giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt. Sự kết hợp giữa hai thước đo định tuyến là chất lượng liên kết (ETX) và trạng thái năng lượng còn lại (ES) trên từng nút chuyển tiếp cho phép lựa chọn tuyến đường tối ưu nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng.

## 1.8. Kết luận chương 1

Trong chương này, tác giả đã trình bày một số đặc điểm và thách thức đặt ra đối với vấn đề định tuyến cho mạng cảm biến không dây. Một số phương pháp tiếp cận trong việc thiết kế giao thức định tuyến cho mạng cảm biến không dây cũng được trình bày ở chương này.

Hầu hết các ứng dụng chính của mạng cảm biến không dây là thu thập thông tin cảm nhận được trong trường cảm biến nên các giao thức thu thập dữ liệu nhận được nhiều sự quan tâm nghiên cứu trong cộng đồng mạng cảm biến không dây. Một trong những giao thức thu thập dữ liệu hiệu quả cho mạng cảm biến không dây đó là giao thức cây thu thập dữ liệu.

Thông qua một số kết quả nghiên cứu khảo sát và đánh giá thực nghiệm với giao thức CTP, tác giả đã xác định các điểm yếu cần khắc phục đối với giao thức CTP. Một trong những điểm yếu mà tác giả sẽ tập trung giải quyết trong luận án này đó là sự mất cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt. Các nút thuộc tuyến đường có chất lượng liên kết tốt sẽ hết năng lượng nhanh hơn các nút còn lại trong mạng, làm giảm hiệu năng của toàn bộ hệ thống mạng. Để khắc phục nhược điểm này, tác giả đề xuất bài toán định tuyến có sự nhận thức về năng lượng (bài toán định tuyến EACTP). Dựa trên những khảo sát, phân tích và đánh giá một số công trình nghiên cứu có liên quan đến bài toán định tuyến có sự nhận thức về năng lượng, tác giả đã đề xuất giải pháp để tiếp cận bài toán định tuyến EACTP.

Một phần kết quả nghiên cứu của chương này đã được tác giả công bố trong bài báo số [2].

## **Chương 2. GIAO THỨC CÂY THU THẬP DỮ LIỆU CÓ SỰ NHẬN THỨC VỀ NĂNG LƯỢNG**

Chương 2 đề xuất một giao thức định tuyến mới EACTP (Energy Aware Collection Tree Protocol) được mở rộng từ giao thức CTP (Collection Tree Protocol). Tác giả cũng đề xuất mô hình toán học cho bài toán định tuyến EACTP dựa trên cơ sở lý thuyết đồ thị. Một số kết quả thực thi và đánh giá giao thức EACTP dựa trên công cụ mô phỏng Cooja cũng được trình bày ở phần cuối của chương này.

### **2.1. Đề xuất mô hình toán học cho bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng**

#### **2.1.1. Mô hình kết nối giữa các nút mạng cảm biến không dây dựa trên lý thuyết đồ thị**

Để phát triển các thuật toán và đưa ra những đánh giá về hiệu năng của mạng cảm biến không dây thì cần phải xây dựng mô hình toán cho mạng cảm biến không dây. Hiện tại, nhiều mô hình cho mạng cảm biến không dây có nguồn gốc từ lý thuyết khoa học máy tính và toán ứng dụng đã được đề xuất. Một mô hình toán được sử dụng rộng rãi đối với mạng cảm biến không dây đó là mô hình toán được xây dựng dựa trên lý thuyết đồ thị. Do cấu trúc liên kết của mạng cảm biến không dây có thể được coi như một đồ thị nên các thuật toán cho mạng cảm biến không dây thường sử dụng mô hình toán từ lý thuyết đồ thị. Trong mô hình lý thuyết đồ thị, các nút cảm biến đại diện cho các đỉnh và các liên kết vô tuyến đại diện cho các cạnh của đồ thị.

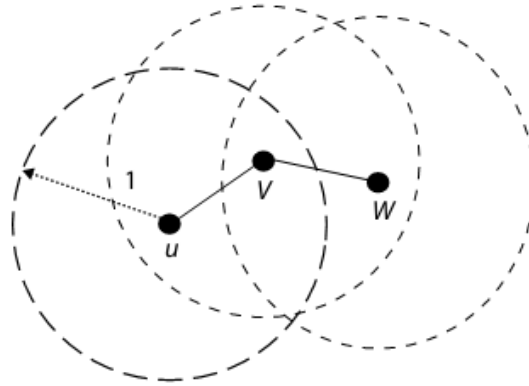
Một vấn đề rất quan trọng liên quan đến kết nối giữa các nút cảm biến đó là: Cho một tập hợp các nút cảm biến phân tán trong trường cảm biến, chúng ta cần phải xác định những nút nào có thể nhận được sự truyền dẫn từ một nút khác. Trong mô hình toán sử dụng lý thuyết đồ thị, nếu một nút  $u$  ở trong phạm vi truyền dẫn của nút  $v$  thì chúng ta nói rằng nút  $u$  liền kề với nút  $v$  hoặc nút  $u$  là nút lân cận của nút  $v$ . Trong trường hợp không có sự can thiệp của nhiễu thì mối quan hệ này thường là đối xứng (hoặc vô hướng). Điều này có nghĩa là nếu nút  $u$  có thể lắng nghe được nút  $v$  thì nút  $v$  cũng có thể lắng nghe được nút  $u$ .

Các kết nối của một mạng cảm biến không dây được mô tả bằng một đồ thị  $G = (V, E)$ , trong đó  $V$  là một tập hợp các nút cảm biến và  $E$  mô tả quan hệ liên kề giữa các nút. Với hai nút  $(u, v) \in V$ , liên kết  $(u, v) \in E$  nếu nút  $v$  liền kề với nút  $u$ . Trong



một đồ thị không định hướng, nếu  $(u, v) \in E$  thì  $(v, u)$  cũng  $\in E$ . Do đó, các cạnh của đồ thị có thể được biểu diễn bởi cặp  $\{u, v\} \in E$ .

Mô hình kết nối kinh điển được sử dụng trong mạng cảm biến không dây là mô hình đồ thị UDG (Unit Disk Graph). Trong mô hình này, các nút đều sử dụng các anten đẳng hướng có độ lợi như nhau theo mọi hướng và các nút được giả định triển khai trong một mặt phẳng có môi trường không bị cản trở. Hai nút được gọi là liên kề với nhau khi và chỉ khi chúng nằm trong phạm vi truyền dẫn của nhau.



**Hình 2.1:** Mô hình UDG [22].

**Mô hình UDG (Unit Disk Graph):** Cho  $V \subset R^2$  là tập hợp các nút trong mặt phẳng Euclide hai chiều. Đồ thị Euclide  $G = (V, E)$  được gọi là đồ thị UDG nếu nó có hai nút liên kề khi và chỉ khi khoảng cách Euclide của chúng lớn nhất bằng 1. Điều này có nghĩa là nếu  $u, v \in V$  thì  $\{u, v\} \in E$  khi và chỉ khi  $|u, v| \leq 1$ . Hình 2.1 mô tả một ví dụ của mô hình UDG. Nút  $u$  là liên kề với nút  $v$  (khoảng cách nhỏ hơn 1) nhưng không liên kề với nút  $w$  (khoảng cách lớn hơn 1).

### 2.1.2. Các vấn đề về nhiễu trong mạng cảm biến không dây

Trong các mạng không dây, kênh truyền thông vô tuyến được chia sẻ và các truyền dẫn trong mạng chịu ảnh hưởng bởi nhiễu. Một nút  $u$  có thể không nhận được chính xác một bản tin được gửi từ một nút liên kề  $v$  do có một truyền dẫn đồng thời khác ở gần đó.

Mô hình nhiễu sẽ diễn tả các truyền dẫn đồng thời trong mạng ảnh hưởng đến nhau như thế nào. Nhiễu là một hiện tượng phức tạp với nhiều đặc điểm khó nắm bắt. Ví dụ, một tín hiệu có thể ảnh hưởng đến chính nó do có hiện tượng truyền sóng vô tuyến đa đường. Mô hình nhiễu được chấp nhận và được sử dụng rộng rãi bởi các nhà nghiên cứu về lý thuyết thông tin là mô hình vật lý hay mô hình SINR (Signal-to-Interference Plus Noise). Trong mô hình này, tỷ lệ tiếp nhận thành công một bản tin

phụ thuộc vào cường độ tín hiệu nhận được, mức độ tạp âm xung quanh và nhiễu gây ra bởi sự truyền dẫn đồng thời của các nút mạng.

**Mô hình SINR:** Gọi  $P_r$  là công suất tín hiệu nhận được bởi nút  $v_r$  và  $I_r$  biểu thị nhiễu sinh ra bởi các nút khác,  $N$  là mức công suất tạp âm của môi trường xung quanh. Khi đó, nút  $v_r$  nhận được một truyền dẫn khi và chỉ khi:

$$\frac{P_r}{N + I_r} \geq \beta \quad (2.1)$$

Trong đó:  $\beta$  là độ nhạy thu (phụ thuộc vào phần cứng) biểu thị tỷ lệ tín hiệu trên nhiễu nhỏ nhất để phía thu có thể nhận thành công một bản tin. Giá trị công suất tín hiệu nhận được  $P_r$  là một hàm giảm theo khoảng cách  $d(v_s, v_r)$  giữa nút gửi  $v_s$  và nút nhận  $v_r$ . Cụ thể hơn, công suất tín hiệu nhận được có thể được mô hình hóa suy hao theo khoảng cách  $d(v_s, v_r)$  là  $\frac{1}{d(v_s, v_r)^\alpha}$ . Trong đó,  $\alpha$  là hằng số mũ suy hao đường truyền có giá trị từ 2 đến 6 phụ thuộc vào điều kiện môi trường truyền sóng cũng như khoảng cách chính xác giữa nút gửi và nút nhận. Gọi  $P_i$  là mức công suất truyền của nút  $v_i$ . Một bản tin được truyền từ nút  $v_s \in V$  sẽ được tiếp nhận thành công bởi nút  $v_r$  nếu:

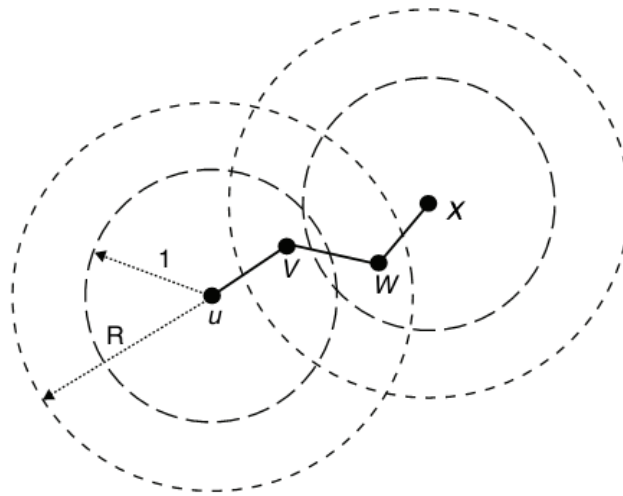
$$\frac{\frac{P_s}{d(v_s, v_r)^\alpha}}{N + \sum_{v_i \in V \setminus \{v_s\}} \frac{P_i}{d(v_i, v_r)^\alpha}} \geq \beta \quad (2.2)$$

Trong mô hình SINR, nút nhận sẽ nhận được chính xác một truyền dẫn nếu công suất tín hiệu nhận được (phụ thuộc vào công suất phát và khoảng cách giữa nút gửi và nút nhận) đủ lớn so với công suất tín hiệu của các truyền dẫn đồng thời khác và mức độ tạp âm xung quanh.

Mặc dù mô hình SINR kết hợp đầy đủ nhiều tính chất vật lý quan trọng của môi trường nhưng nó không được sử dụng nhiều trong cộng đồng nghiên cứu thuật toán cho mạng cảm biến không dây. Nguyên nhân chính là do mô hình SINR quá phức tạp. Ví dụ như có rất nhiều truyền dẫn khác nhau được tổng hợp và có thể từng cặp nút gửi - nhận gần nhau có ảnh hưởng đến nhau. Trong thực tế, những truyền dẫn khác nhau này thường chỉ tạo thêm tạp âm trong môi trường xung quanh và không cần thiết phải tính riêng như vậy.

Một mô hình đơn giản hơn được sử dụng phổ biến đó là mô hình UDI (UDG with Distance Interference). Mô hình UDI là một dạng của mô hình UDG có xét đến tác động của nhiễu. Mô hình UDI sẽ được sử dụng trong mô phỏng và đánh giá hiệu năng mạng ở mục 2.4 của luận án này.

**Mô hình UDG với nhiễu khoảng cách (UDI):** Các nút được phân bố tùy ý trong mặt phẳng. Hai nút có thể truyền thông trực tiếp với nhau khi và chỉ khi khoảng cách Euclide lớn nhất bằng 1 và nếu như phía thu không bị ảnh hưởng bởi một nút thứ ba với khoảng cách Euclide nhỏ hơn hoặc bằng một hằng số  $R \geq 1$ .



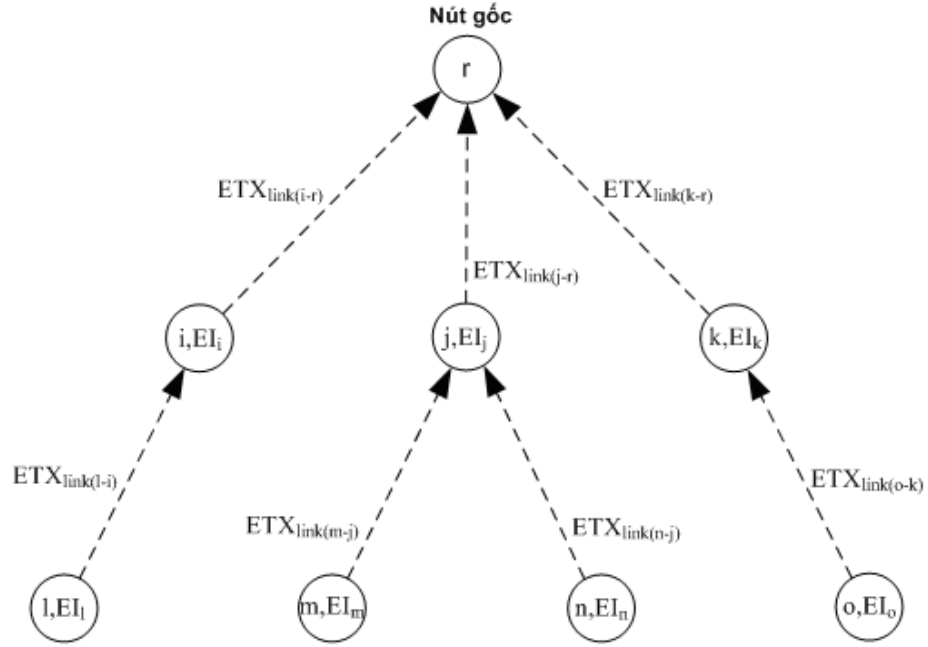
**Hình 2.2:** Mô hình UDI [22].

Hình 2.2 mô tả một ví dụ mô hình UDI với hai bán kính: Một bán kính truyền dẫn (bằng 1) và một bán kính nhiễu ( $R \geq 1$ ). Trong ví dụ này, nút  $v$  không thể nhận được một truyền dẫn từ nút  $u$  nếu nút  $x$  truyền dữ liệu đồng thời đến nút  $w$  mặc dù nút  $v$  không liền kề với nút  $x$ .

### 2.1.3. Mô hình bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng dựa trên lý thuyết đồ thị

Bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng (bài toán định tuyến EACTP) được mô hình hóa dựa trên lý thuyết đồ thị như sau:

Cho một cấu trúc mạng  $G = (V, E)$  là một đồ thị, trong đó  $V$  là số đỉnh,  $E$  là số cạnh. Bài toán định tuyến EACTP đặt ra đó là tìm được một cấu trúc cây tối ưu dựa trên hai tham số là chất lượng liên kết giữa các nút mạng và trạng thái năng lượng còn lại của các nút mạng. Hình 2.3 là ví dụ minh họa mô hình bài toán định tuyến EACTP dựa trên lý thuyết đồ thị.



**Hình 2.3:** Mô hình bài toán định tuyến EACTP dựa trên lý thuyết đồ thị.

Mỗi đỉnh thuộc cây biểu diễn một nút mạng có cả chức năng sinh dữ liệu và chức năng định tuyến. Mỗi nút mạng xác định được chỉ số năng lượng còn lại  $EI$  (Energy Indicator) của nó.

Mỗi cạnh thuộc cây là một tuyến (link) kết nối giữa hai nút mạng  $(u, v)$  bất kỳ được biểu diễn bởi chất lượng liên kết giữa hai nút mạng theo tham số chất lượng liên kết ( $ETX_{link}$ ).

Gọi độ dài đường đi  $P$  nối từ một điểm bất kỳ trên  $G = (V, E)$  đến nút gốc được xác định bằng tổng  $ETX_{link}$  của tất cả các tuyến kết nối thuộc tuyến đường đó. Mỗi nút được gán một nhãn ( $rtmetric$ ) chính là độ dài đường đi tốt nhất từ nút đó đến nút gốc được xác định theo thước đo định tuyến chất lượng liên kết  $ETX$ . Bài toán định tuyến EACTP được mô hình hóa như sau: Tìm nút  $n$  thuộc tập hợp các nút lân cận  $N$  của nút nguồn  $s$  trên cấu trúc  $G = (V, E)$  sao cho nút  $n$  thỏa mãn các điều kiện sau:

$$\begin{cases} rtmetric_s = \arg \min_{n \in N} (rtmetric_n + ETX_{link(s-n)}) \\ n.EI \geq Threshold \end{cases} \quad (2.3)$$

Trong đó: -  $rtmetric_n$  là nhãn của nút  $n$ .

-  $rtmetric_s$  là nhãn của nút  $s$  được xác định theo nút  $n$ .

-  $ETX_{link(s-n)}$  là chất lượng liên kết giữa nút  $s$  và nút  $n$ .

- $n.EI$  là chỉ số năng lượng còn lại của nút  $n$ .
- *Threshold* là ngưỡng chỉ số năng lượng còn lại.

## 2.2. Đề xuất giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng EACTP

### 2.2.1. Mục tiêu và những thách thức trong việc đề xuất giao thức EACTP

Trong giao thức CTP hiện tại, các nút cảm biến thuộc tuyến đường có chất lượng liên kết tốt sẽ được chọn làm nút cha trong phần lớn thời gian và các nút đó tham gia nhiều vào quá trình truyền thông trong mạng. Điều này dẫn đến việc các nút thuộc những tuyến đường có chất lượng liên kết tốt sẽ bị hết năng lượng nhanh hơn các nút khác và làm giảm thời gian sống của toàn mạng.

Để giải quyết vấn đề này, tác giả đề xuất một giao thức định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng (giao thức EACTP) nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc các tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Giao thức EACTP sử dụng kết hợp thước đo chất lượng liên kết *ETX* và trạng thái năng lượng còn lại trên mỗi nút cảm biến để xác định tuyến đường tối ưu. Thông tin về trạng thái năng lượng còn lại trên mỗi nút cảm biến được bổ sung trong bảng định tuyến và được trao đổi qua lại giữa các nút lân cận nhau thông qua các bản tin điều khiển. Tuyến đường trong giao thức EACTP là tuyến đường dự trữ. Tuyến đường dự trữ này được lựa chọn dựa trên thông tin về chất lượng liên kết của tuyến đường và thông tin về trạng thái năng lượng còn lại của nút chuyển tiếp.

Tuy nhiên, điều này có thể dẫn đến việc lựa chọn tuyến đường không phải là tối ưu nhất theo thước đo *ETX* và có thể làm tăng tỷ lệ mất gói tin và năng lượng tiêu thụ trên mỗi bước nhảy. Để giải quyết vấn đề này, trước tiên giao thức EACTP quan tâm đến việc lựa chọn tuyến đường tối ưu theo thước đo chất lượng liên kết *ETX* và khi trạng thái năng lượng còn lại trên nút cha hiện tại thấp hơn một ngưỡng nhất định thì thuật toán sẽ lựa chọn một tuyến đường dự trữ để đảm bảo rằng nút cha mới có mức năng lượng còn lại lớn hơn một ngưỡng cho trước.

Một số thách thức đặt ra khi xây dựng giao thức EACTP đó là:

- Thứ nhất, cần phải xác định được năng lượng còn lại trên mỗi nút cảm biến. Cách xác định năng lượng còn lại trên mỗi nút cảm biến cần thực hiện được

trên nhiều kiến trúc phần cứng khác nhau và không làm phát sinh thêm bất kỳ một chi phí về phần cứng nào.

- Thứ hai, cần phải đưa ra một thước đo định tuyến phù hợp dựa trên thông tin về năng lượng còn lại trên mỗi nút cảm biến. Thước đo định tuyến mới đề xuất cần phản ánh được các trạng thái năng lượng còn lại trên mỗi nút cảm biến. Thước đo định tuyến này được phát quảng bá cho các nút lân cận thông qua các bản tin điều khiển và được cập nhật thường xuyên trong bảng định tuyến của các nút lân cận. Điều này sẽ làm phát sinh thêm chi phí mới về năng lượng trong việc gửi các bản tin điều khiển.
- Thứ ba, tuyến đường tối ưu được lựa chọn trong giao thức EACTP phải thỏa mãn các tiêu chí của việc định tuyến trong mạng cảm biến không dây đó là tuyến đường lựa chọn phải có tỷ lệ chuyển phát bản tin dữ liệu thành công đến nút gốc ở mức cao và đây cũng là tuyến đường hiệu quả về mặt năng lượng.

## **2.2.2. Những đề xuất cải tiến trong giao thức EACTP**

### ***2.2.2.1. Ước lượng năng lượng tiêu thụ trên nút cảm biến***

Năng lượng là một vấn đề rất quan trọng đối với các mạng cảm biến không dây. Nếu năng lượng tiêu thụ trên các nút cảm biến có thể ước lượng được thì các giao thức định tuyến có thể lựa chọn các tuyến đường tối ưu dựa vào thông tin năng lượng còn lại trên mỗi nút cảm biến nhằm tăng thời gian sống của mạng. Kết quả nghiên cứu trong bài báo [26] đã chỉ ra rằng việc lựa chọn tuyến đường dựa trên thông tin về năng lượng tiêu thụ trên các nút cảm biến có thể giúp kéo dài thời gian sống của mạng.

Một số nền tảng phần cứng cho các mạng cảm biến không dây hiện nay như TUmote, IRIS, MicaZ [27]... đều không hỗ trợ cơ chế đo năng lượng tiêu thụ của nút cảm biến dựa trên phần cứng. Hơn nữa, các ứng dụng của mạng cảm biến không dây với những đặc điểm riêng cũng làm cho việc đo năng lượng tiêu thụ dựa trên phần cứng gặp nhiều khó khăn [28]. Các cơ chế đo năng lượng dựa trên phần cứng thường gặp nhiều khó khăn khi được áp dụng vào các thiết kế phần cứng hiện tại bởi vì các cơ chế này thường yêu cầu khá nhiều sự điều chỉnh trên các thiết kế phần cứng hiện tại [28]. Chi phí để đo năng lượng dựa trên phần cứng có thể tăng lên tới 100% [28]. Trong khi đó, cơ chế ước lượng năng lượng tiêu thụ dựa trên phần mềm được áp dụng trên các thiết kế phần cứng và các thiết kế phần mềm hiện có mà không yêu cầu thêm bất kỳ một chi phí bổ sung nào.

Trong luận án này, tác giả sử dụng cơ chế ước lượng năng lượng tiêu thụ dựa trên phần mềm cho các nút cảm biến không dây. Cơ chế này sử dụng mô hình năng lượng tuyến tính đơn giản và dễ dàng thực thi trên các hệ điều hành cho mạng cảm biến không dây hiện nay. Trong bài báo [39], các tác giả đã công bố một số kết quả đánh giá về độ chính xác của cơ chế này. Các kết quả đánh giá so sánh giữa cơ chế ước lượng năng lượng dựa trên phần mềm và cơ chế ước lượng năng lượng dựa trên phần cứng đã cho thấy cơ chế ước lượng năng lượng dựa trên phần mềm có thể đạt độ chính xác đến 96%.

Cơ chế ước lượng năng lượng dựa trên phần mềm sử dụng mô hình tiêu thụ năng lượng tuyến tính. Tổng năng lượng  $E$  được tính toán như sau [29]:

$$E = U(I_a t_a + I_l t_l + I_t t_t + I_r t_r + \sum_i I_{ci} t_{ci}) \quad (2.4)$$

Trong đó:

- $U$ : Là điện áp nguồn cung cấp (điện áp pin).
- $I_a, t_a$ : Là dòng tiêu thụ và thời gian mà bộ vi xử lý khi hoạt động ở chế độ tích cực (*active mode*).
- $I_l, t_l$ : Là dòng tiêu thụ và thời gian mà bộ vi xử lý khi hoạt động ở chế độ công suất thấp (*low power mode*).
- $I_t, t_t$ : Là dòng tiêu thụ và thời gian bộ thu phát vô tuyến ở chế độ truyền (*transmit*).
- $I_r, t_r$ : Là dòng tiêu thụ và thời gian bộ thu phát vô tuyến ở chế độ nhận (*receive*).
- $I_{ci}, t_{ci}$ : Là dòng tiêu thụ và thời gian hoạt động của các bộ phận khác như cảm biến, LED...

**Bảng 2.1:** Mô hình năng lượng của TUmote.

Thành phần	Trạng thái	Dòng tiêu thụ
MSP430 F1611	Tích cực	1,95 mA
	Công suất thấp	0,0026 mA
CC2420	Truyền (0 dBm)	17,4 mA
	Truyền (-15 dBm)	9,9 mA
	Nhận	18,8 mA
SHT11	Tích cực	0,55 mA

Bảng 2.1 là mô hình năng lượng của phần cứng TUmote. Trong mô hình năng lượng của TUmote, tác giả chỉ quan tâm đến các thành phần tiêu thụ năng lượng chính và bỏ qua các thành phần tiêu thụ năng lượng nhỏ khác. Các thông số về dòng tiêu thụ của các linh kiện trong TUmote được xác định từ tài liệu mô tả kỹ thuật của nhà cung cấp. Cụ thể như sau:

- Bộ điều khiển MSP430 F1611 ở chế độ tích cực có tần số làm việc của CPU được thiết lập trong File *platform-config.h*:  $F\_CPU = 3,9 \text{ MHz}$ . Theo công thức tính toán dòng tiêu thụ của nhà cung cấp [30] thì dòng tiêu thụ được tính như sau:

$$I(AM) = I(AM) [1\text{MHz}] \times F [\text{MHz}] = 500 \times 3,9 = 1,95 \text{ mA}.$$

- Bộ vi điều khiển MSP430 F1611 ở chế độ công suất thấp có dòng tiêu thụ là  $2,6 \mu\text{A} = 0.0026 \text{ mA}$  [30].
- Bộ thu phát vô tuyến CC2420 ở chế độ phát công suất -15dBm là 9,9 mA; 0dBm là 17,4 mA; nhận là 18,8 mA [30].
- SHT 11 có dòng tiêu thụ là 0,55 mA [31].

Các biến thời gian ( $t_a, t_l, t_t, t_r, t_{ci}$ ) trong công thức (2.4) được xác định dựa trên nhận xét sau: Để tiết kiệm năng lượng thì các nút cảm biến thường xuyên bật hoặc tắt các thiết bị của chúng (ví dụ như thiết bị truyền thông, các cảm biến, các LED) hoặc chuyển đổi qua lại giữa chế độ tích cực và chế độ công suất thấp (ví dụ với bộ vi điều khiển). Cơ chế ước lượng năng lượng sử dụng mô hình tuyến tính sẽ được gọi đến mỗi khi một thiết bị phần cứng bật hoặc tắt hoặc chuyển chế độ. Khi một thiết bị phần cứng được bật thì mô đun ước lượng năng lượng sẽ lưu một dấu thời gian. Khi thiết bị phần cứng này được tắt thì sự sai khác về thời gian giữa hai thời điểm sẽ được tính toán và được cộng vào tổng thời gian bật của thiết bị.

Thuật toán 2.1 diễn giải cách xác định các biến thời gian đối với phần cứng TUmote ở các chế hoạt động khác nhau.

---

**Thuật toán 2.1: Xác định thời gian các thiết bị phần cứng ở các chế độ hoạt động khác nhau**

---

**1. Định nghĩa các chế độ hoạt động khác nhau của các thiết bị phần cứng**

*enum energest\_type {*

*ENERGEST\_TYPE\_CPU, //Chế độ xử lý của bộ vi điều khiển*

*ENERGEST\_TYPE\_LPM, //Chế độ công suất thấp của bộ vi điều khiển*



```

ENERGEST_TYPE_TRANSMIT, //Chế độ truyền của bộ thu phát vô tuyến
ENERGEST_TYPE_LISTEN, //Chế độ nhận của bộ thu phát vô tuyến
ENERGEST_TYPE_SENSOR //Cảm biến
} type;
2. Khi kích hoạt chế độ hoạt động của thiết bị phần cứng
ENERGEST_ON(type) {
    energest_current_time[type] = RTIMER_NOW(); //Luu một dấu thời gian
}
3. Khi tắt chế độ hoạt động của thiết bị phần cứng
ENERGEST_OFF(type){
    energest_total_time[type].current += (rtimer_clock_t)(RTIMER_NOW() -
    energest_current_time[type]);
    //Tính tổng thời gian kích hoạt chế độ hoạt động của thiết bị phần cứng
}
4. Xác định thời gian các thiết bị phần cứng ở các chế độ hoạt động khác nhau
energest_type_time(type){
    return energest_total_time[type].current;
}

```

#### 2.2.2.2. Thước đo chỉ số năng lượng còn lại thích ứng

Có một số giải pháp để tính toán thước đo chỉ số năng lượng còn lại *EI* (*Energy Indicator*) trên mỗi nút cảm biến.

Trong tài liệu [32], các tác giả đã đề xuất công thức tính chỉ số năng lượng còn lại *EI* như sau:

$$EI(\%) = \frac{P_{\max}}{P_{\text{now}}} \cdot 100\% \quad (2.5)$$

Trong đó : -  $P_{\text{now}}$ : Là công suất tiêu thụ trung bình hiện tại.

-  $P_{\max}$ : Là công suất tiêu thụ lớn nhất mong muốn.

$P_{\max}$  có thể được xác định theo công thức sau:

$$P_{\max} = \frac{E_0}{T} \quad (2.6)$$

Trong đó : -  $E_0$ : Là năng lượng ban đầu của nút cảm biến.

-  $T$ : Là thời gian sống dự kiến của một nút cảm biến.

Chỉ số năng lượng còn lại *EI* được mã hóa bởi 8 bit. Khi lựa chọn giải pháp này, bản tin điều khiển cần phải được mở rộng thêm 8 bit để mang thông tin về chỉ số

năng lượng còn lại  $EI$ . Do đó, giải pháp này sẽ phát sinh thêm các chi phí mới về năng lượng trong việc gửi các bản tin điều khiển.

Một giải pháp khác đơn giản hơn và cũng đạt được hiệu quả mong muốn mà không làm phát sinh thêm các chi phí mới về năng lượng đã được tác giả sử dụng trong giao thức EACTP.

Giao thức EACTP đề xuất ý tưởng ngưỡng chỉ số năng lượng còn lại thích ứng. Mỗi nút có bốn trạng thái năng lượng còn lại khác nhau. Các trạng thái năng lượng này khác nhau tương ứng với các mức chỉ số năng lượng còn lại khác nhau. Giá trị của ngưỡng năng lượng được điều chỉnh thích ứng theo bốn trạng thái năng lượng còn lại của nút cảm biến. Các thuật toán thu thập dữ liệu hiện tại lựa chọn nút chuyển tiếp chỉ dựa vào một ngưỡng cụ thể và là một giá trị cố định. Điều này dẫn đến việc làm giảm hiệu năng của mạng. Bảng 2.2 minh họa bốn trạng thái năng lượng của nút. Bốn trạng thái năng lượng còn lại trên nút cảm biến có thể được mã hóa bởi 2 bit dự trữ trong bản tin điều khiển. Vì vậy, giải pháp này không làm phát sinh các chi phí mới về năng lượng trong việc gửi các bản tin điều khiển.

**Bảng 2.2:** Các trạng thái năng lượng của nút cảm biến.

Trạng thái năng lượng	Chỉ số năng lượng còn lại EI	Mô tả	Ngưỡng chỉ số năng lượng EI
0	30%-100%	Nhiều năng lượng	30%
1	10%-30%	Thiếu hụt năng lượng	10%
2	5%-10%	Thiếu nhiều năng lượng	5%
3	0-5%	Hết năng lượng	0

Giả sử rằng các nút ban đầu được tích trữ đầy năng lượng và chỉ số năng lượng còn lại  $EI$  của mỗi nút đều bằng 100%. Chỉ số năng lượng còn lại trên mỗi nút cảm biến tại một thời điểm  $t$  bất kỳ có thể được xác định theo công thức sau:

$$EI(\%) = \frac{residual\_energy}{E_0} \cdot 100\% \quad (2.7)$$

Trong đó:

- ***residual\_energy***: Là năng lượng còn lại trên nút cảm biến. Năng lượng còn lại tại thời điểm  $t$  bất kỳ của nút cảm biến được xác định theo công thức:

$$residual\_energy = E_0 - consumption\_energy \quad (2.8)$$

Trong đó: *consumption\_energy* là năng lượng tiêu thụ trên nút cảm biến tại thời điểm *t*. Năng lượng tiêu thụ trên nút cảm biến được xác định theo mô hình năng lượng tuyến tính (mục 2.2.2.1).

- $E_0$ : Là năng lượng ban đầu của nút cảm biến.

Nếu TUmote được cung cấp năng lượng bằng 2 pin tiêu AA có dung lượng 2200mAh. Mỗi pin tiêu AA có điện áp bằng 1,5V. Như vậy, năng lượng ban đầu của TUmote có thể được xác định theo công thức:

$$E_0 = 2 \times 1,5 \times 2,2 \times 3600 = 23760 \text{ J}$$

### 2.2.2.3. Bổ sung thông tin về trạng thái năng lượng còn lại của nút cảm biến trong cấu trúc bản tin điều khiển

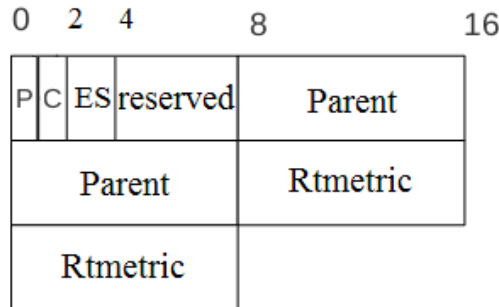
Giao thức EACTP đề xuất ngưỡng năng lượng còn lại thích ứng. Mỗi nút cảm biến có bốn trạng thái năng lượng khác nhau tương ứng với các mức ngưỡng năng lượng khác nhau. Bốn trạng thái năng lượng này có thể được mã hóa bởi hai bit dự trữ trong bản tin điều khiển. Vì vậy, giao thức EACTP không làm phát sinh thêm về chi phí năng lượng trong việc gửi các bản tin điều khiển. Bảng 2.3 là bảng mã hóa bốn trạng thái năng lượng còn lại trên mỗi nút cảm biến.

**Bảng 2.3:** Bảng mã hóa các trạng thái năng lượng còn lại của nút cảm biến.

Trạng thái năng lượng	ES
0	00
1	01
2	10
3	11

Trong giao thức EACTP, chất lượng liên kết *ETX* và trạng thái năng lượng còn lại *ES* (*Energy States*) được sử dụng làm thước đo định tuyến. Nút cha được lựa chọn để chuyển tiếp bản tin dữ liệu phải có trạng thái năng lượng còn lại ở mức cao (chỉ số năng lượng còn lại *EI* lớn hơn hoặc bằng một ngưỡng cho trước). Trạng thái năng lượng còn lại *ES* của mỗi nút cần phải được gửi cho các nút lân cận để các nút lân cận biết và cập nhật lại trạng thái năng lượng *ES* trong bảng định tuyến. Giao thức CTP hiện tại sử dụng bản tin điều khiển (mục 1.4.2.2 chương 1) để gửi quảng bá thước đo định tuyến *rtmetric* của mỗi nút cho các nút lân cận. Cấu trúc khung của bản tin điều khiển đã được trình bày chi tiết trong mục 1.4.2.2 thuộc chương 1 của luận án.

Hình 2.4 minh họa cấu trúc khung của bản tin điều khiển trong giao thức EACTP. Bản tin này được bổ sung thêm trường trạng thái năng lượng còn lại *ES* gồm hai bit *ES[1]* và *ES[0]* để mã hóa bốn trạng thái năng lượng khác nhau của nút cảm biến (theo bảng 2.2).



**Hình 2.4:** Cấu trúc khung bản tin điều khiển trong giao thức EACTP.

#### 2.2.2.4. Thuật toán lựa chọn tuyến đường dựa trên sự kết hợp giữa chất lượng liên kết của tuyến đường và trạng thái năng lượng còn lại của nút chuyển tiếp

Giao thức CTP hiện tại sử dụng thước đo định tuyến *ETX* để xây dựng cấu trúc cây định tuyến. Vị trí của các nút trong cây được xác định bởi thước đo tuyến đường *rtmetric* (*route metric*). Thước đo *rtmetric* của một nút được xác định bằng tổng thước đo *ETX* trên các liên kết thuộc tuyến đường từ nút đó đến nút gốc. Nút gốc ở đỉnh cây có giá trị *rtmetric* = 0. Các nút lá sẽ có giá trị *rtmetric* càng lớn khi càng xa nút gốc. Tuyến đường có *rtmetric* càng lớn thì chất lượng các liên kết thuộc tuyến đường càng thấp. Tuyến đường tốt nhất là tuyến đường có *rtmetric* là nhỏ nhất. Đây là tuyến đường có tổng số lần truyền đến nút gốc là ít nhất và cũng là tuyến đường hiệu quả về mặt năng lượng nhất. Thuật toán 2.2 minh họa quá trình lựa chọn tuyến đường của giao thức CTP.

---

#### **Thuật toán 2.2: Thuật toán lựa chọn tuyến đường với giao thức CTP**

---

##### **1. Đầu vào:**

*RoutingTable*; //Bảng định tuyến của nút nguồn *s*

##### **2. Lựa chọn nút lân cận tốt nhất**

*best* = NULL; // nút lân cận tốt nhất trong bảng định tuyến

*rtmetric* = RTMETRIC\_MAX;

//RTMETRIC\_MAX = COLLECT\_MAX\_DEPTH = 511

**for** *n* = *RoutingTable*[*i*]

**if** (*n.rtmetric* +  $ETX_{link(s-n)} < rtmetric$ ) **then**

*rtmetric* = *n.rtmetric* +  $ETX_{link(s-n)}$ ;

*best* = *n*;

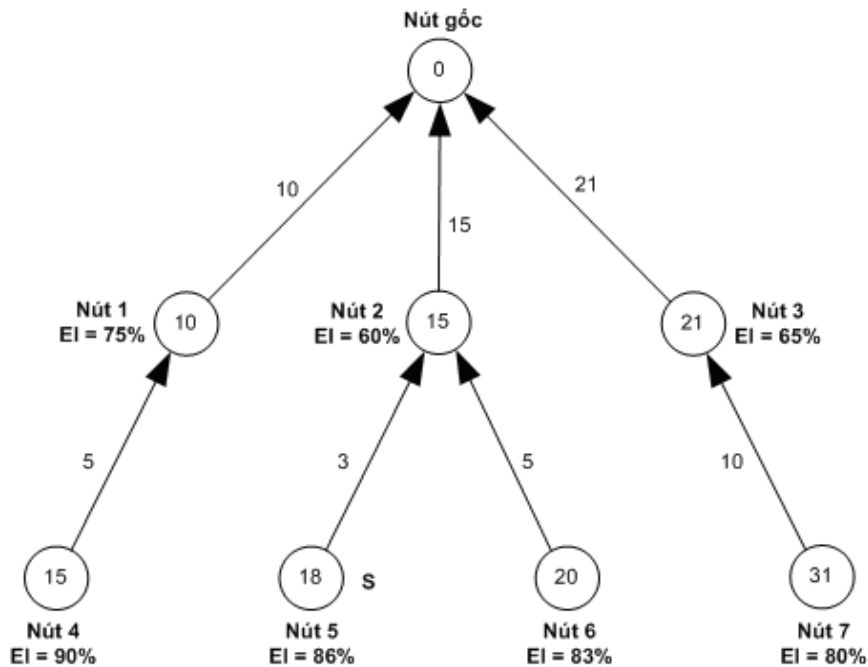
```

end
return best;
3. Cập nhật nút cha
if best != NULL then
    parent = best;
else
    s.rtmetric = RTMETRIC_MAX;
    broadcast(s.rtmetric);
    //Yêu cầu thông tin định tuyến từ các nút lân cận
end

```

Tuy nhiên, giao thức CTP hiện tại chỉ dựa vào thước đo định tuyến *ETX* để lựa chọn tuyến đường tối ưu. Thước đo định tuyến *ETX* không giải quyết được vấn đề cân bằng năng lượng giữa các nút mạng. Các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt phải thực hiện nhiều việc truyền dẫn trong mạng. Các nút mạng này sẽ hết năng lượng nhanh hơn các nút mạng khác và tạo thành các lỗ hổng trong mạng, làm giảm hiệu năng của toàn bộ hệ thống mạng. Để giải quyết vấn đề này, tác giả đề xuất một thuật toán lựa chọn tuyến đường mới cho giao thức EACTP.

Thuật toán lựa chọn tuyến đường mới dựa trên thông tin về chất lượng liên kết *ETX* và trạng thái năng lượng còn lại *ES*. Hình 2.5 minh họa một cấu trúc cây định tuyến EACTP. Các nút tự tổ chức thành một cấu trúc dạng cây và dữ liệu luôn được gửi về nút cha cho đến khi tới được đỉnh của cây (nút gốc).



**Hình 2.5:** Cấu trúc cây định tuyến EACTP.

Thuật toán 2.3 minh họa quá trình lựa chọn tuyến đường của giao thức EACTP.

---

**Thuật toán 2.3: Thuật toán lựa chọn tuyến đường với giao thức EACTP**

---

**1. Đầu vào:**

*RoutingTable*; //Bảng định tuyến của nút nguồn *s*

**2. Lựa chọn nút lân cận tốt nhất**

*best* = NULL; // nút lân cận tốt nhất trong bảng định tuyến

*rtmetric* = *RTMETRIC\_AVERAGE*; // giá trị *RTMETRIC* trung bình của các //tuyến đường có thể, đi qua các nút lân cận trong bảng định tuyến

**for** *n* = *RoutingTable*[*i*]

**if** (*n.ES* == 00) && (*n.rtmetric* +  $ETX_{link(s-n)} \leq rtmetric$ ) **then**

*rtmetric* = *n.rtmetric* +  $ETX_{link(s-n)}$ ;

*best* = *n*;

**end**

**if** *best* == NULL

**for** *n* = *RoutingTable*[*i*]

**if** (*n.ES* == 01) && (*n.rtmetric* +  $ETX_{link(s-n)} \leq rtmetric$ ) **then**

*rtmetric* = *n.rtmetric* +  $ETX_{link(s-n)}$ ;

*best* = *n*;

**end**

**end**

**if** *best* == NULL

**for** *n* = *RoutingTable*[*i*]

**if** (*n.ES* == 10) && (*n.rtmetric* +  $ETX_{link(s-n)} \leq rtmetric$ ) **then**

*rtmetric* = *n.rtmetric* +  $ETX_{link(s-n)}$ ;

*best* = *n*;

**end**

**end**

**return** *best*;

**3. Cập nhật nút cha**

**if** *best* != NULL **then**

*parent* = *best*;

**else**

*s.rtmetric* = *RTMETRIC\_MAX*;

*broadcast*(*s.rtmetric*);

    //Yêu cầu thông tin định tuyến từ các nút lân cận

**end**

Quá trình lựa chọn tuyến đường được thực hiện lần lượt qua từng trạng thái năng lượng còn lại trên các nút cảm biến. Thuật toán lựa chọn tuyến đường này đảm bảo lựa chọn được tuyến đường có chất lượng liên kết tốt (nhỏ hơn hoặc bằng mức *RTMETRIC* trung bình) và vừa đảm bảo nút chuyển tiếp có trạng thái năng lượng còn

lại ở mức cao. Vì vậy, giao thức EACTP tránh được việc lựa chọn nút lân cận có mức năng lượng còn lại ở mức thấp làm nút cha.

Với việc lựa chọn ngưỡng năng lượng thích ứng, giao thức EACTP đã phân loại các nút cảm biến theo một số trạng thái năng lượng khác nhau. Điều này tốt hơn so với việc lựa chọn một ngưỡng năng lượng cố định bởi vì quá trình lựa chọn tuyến đường dự trữ được điều chỉnh trong nhiều khoảng ngưỡng năng lượng khác nhau. Các nút có chỉ số năng lượng còn lại ở mức thấp sẽ không được lựa chọn làm nút cha ngay trong giai đoạn đầu của quá trình suy giảm năng lượng.

Trong trường hợp không tìm được một nút lân cận nào thỏa mãn thì một bản tin điều khiển có trường  $rtmetric = RTMETRIC\_MAX$  sẽ được gửi đi cho các nút lân cận. Các nút lân cận sau khi nhận được bản tin điều khiển có trường  $rtmetric = RTMETRIC\_MAX$  sẽ lập lịch để gửi quảng bá trạng thái năng lượng còn lại  $ES$  và thước đo  $rtmetric$  của nó trong khoảng thời gian gần nhất. Mục đích của việc gửi bản tin điều khiển có trường  $rtmetric = RTMETRIC\_MAX$  là để dò tìm các nút lân cận ứng viên khác ở trong vùng lân cận. Bản tin điều khiển này thông báo cho các nút lân cận biết rằng nút đó hiện tại chưa xác định được nút cha để chuyển tiếp dữ liệu. Các nút lân cận sau khi nhận được bản tin điều khiển có trường  $rtmetric = RTMETRIC\_MAX$  sẽ lập lịch để gửi quảng bá thông tin định tuyến của nó trong khoảng thời gian gần nhất.

## **2.3. Thực thi giao thức EACTP**

### **2.3.1. Lựa chọn môi trường phát triển phần mềm**

#### **2.3.1.1. Hệ điều hành cho mạng cảm biến không dây**

Chức năng cơ bản của hệ điều hành bao gồm việc trừu tượng hóa tài nguyên cho các thiết bị phần cứng khác nhau, quản lý ngắt và lập lịch cho các nhiệm vụ, điều khiển đồng thời và hỗ trợ mạng. Dựa trên các dịch vụ được cung cấp bởi hệ điều hành, người lập trình ứng dụng có thể thuận tiện sử dụng các giao diện lập trình ứng dụng mức cao (APIs) độc lập với phần cứng lớp dưới.

Hệ điều hành cho các nút cảm biến được thiết kế với các yêu cầu đặc trưng riêng của nút cảm biến ví dụ như những hạn chế về tài nguyên bộ nhớ và sức mạnh xử lý của phần cứng. Hệ điều hành cho nút cảm biến không có một giao diện người dùng như các hệ điều hành cho máy tính bởi vì người dùng không tương tác trực tiếp với hệ điều hành trên nút cảm biến. Do đó, hệ điều hành ẩn sâu bên trong bộ vi điều

khuyến và chỉ những người lập trình cho nút cảm biến mới tiếp xúc trực tiếp với hệ điều hành.

Trong thời gian qua đã có nhiều hệ điều hành khác nhau được nghiên cứu và phát triển cho mạng cảm biến không dây ví dụ như TinyOS, Contiki, LiteOS, RTOS, Mantis... Trong đó, TinyOS là hệ điều hành ra đời sớm nhất, còn LiteOS là hệ điều hành mới được nghiên cứu và phát triển trong thời gian gần đây. Hiện nay, TinyOS và Contiki là hai hệ điều hành phổ biến nhất trong cộng đồng nghiên cứu về mạng cảm biến không dây. Trong bài báo công bố số 4, tác giả đã trình bày một nghiên cứu so sánh về ba hệ điều hành này. Bảng 2.4 so sánh giữa ba hệ điều hành này dựa trên một số đặc điểm chính.

**Bảng 2.4:** So sánh giữa các hệ điều hành TinyOS, Contiki, LiteOS.

<b>Đặc điểm</b>	<b>TinyOS</b>	<b>ContikiOS</b>	<b>LiteOS</b>
Năm công bố	ASPLOS (2000)	EmNets (2004)	IPSN (2008)
Website	www.tinyos.net	<a href="http://www.contiki-os.org/">http://www.contiki-os.org/</a>	www.liteos.net
Hệ thống	Tĩnh, nguyên khối	Động, Mô đun	Động, Mô đun
Hỗ trợ mạng	Active Message	uIP, uIPv6, Rime	File-Assisted
Ngôn ngữ lập trình	nesC	C	LiteC++
Lập trình dựa trên sự kiện	Có	Có	Có
Hỗ trợ đa luồng	Có	Có	Có
Lập trình lại không dây	Có	Có	Có
File hệ thống	ELF, Matchbox	Coffee	Hierarchical Unix-like
Hỗ trợ các Nền tảng phần cứng	Mica, Mica2, MicaZ, TelosB, Tmote, XYZ, IRIS, Tinynode, Eyes, Shimmer	Tmote, TelosB, ESB, AVR MCU, MSP430 MCU	MicaZ, IRIS, AVR MCU
Công cụ mô phỏng	TOSSIM, Power Tossim	Cooja, MSPSim, Netsim	Through AVRORA

### 2.3.1.2. Hệ điều hành Contiki

Contiki là một hệ điều hành mã nguồn mở được thiết kế cho các hệ thống mạng nhúng nói chung và mạng cảm biến không dây nói riêng. Hệ điều hành Contiki được phát triển bởi Adam Dunkels và phiên bản đầu tiên được phát hành vào năm 2003. Nhóm phát triển Contiki gồm nhiều thành viên đến từ SICS, CISCO, cùng



nhiều tổ chức và các trường đại học khác trên thế giới. Hệ điều hành Contiki được thiết kế cho các vi điều khiển có bộ nhớ nhỏ, với thông số 2KB RAM và 40KB ROM. Nhờ đó, Contiki có thể được sử dụng cho các hệ thống nhúng và các ứng dụng trong mạng cảm biến không dây. Trong thực tế, Contiki đã được ứng dụng trong nhiều dự án như giám sát đường hầm xe lửa, theo dõi nước trong biển Baltic...

Hệ điều hành Contiki cùng đi kèm với công cụ mô phỏng Cooja/MSPSim. Công cụ mô phỏng này cho phép các đoạn mã chương trình viết trên nền hệ điều hành Contiki có thể được mô phỏng, thử nghiệm trước khi triển khai.

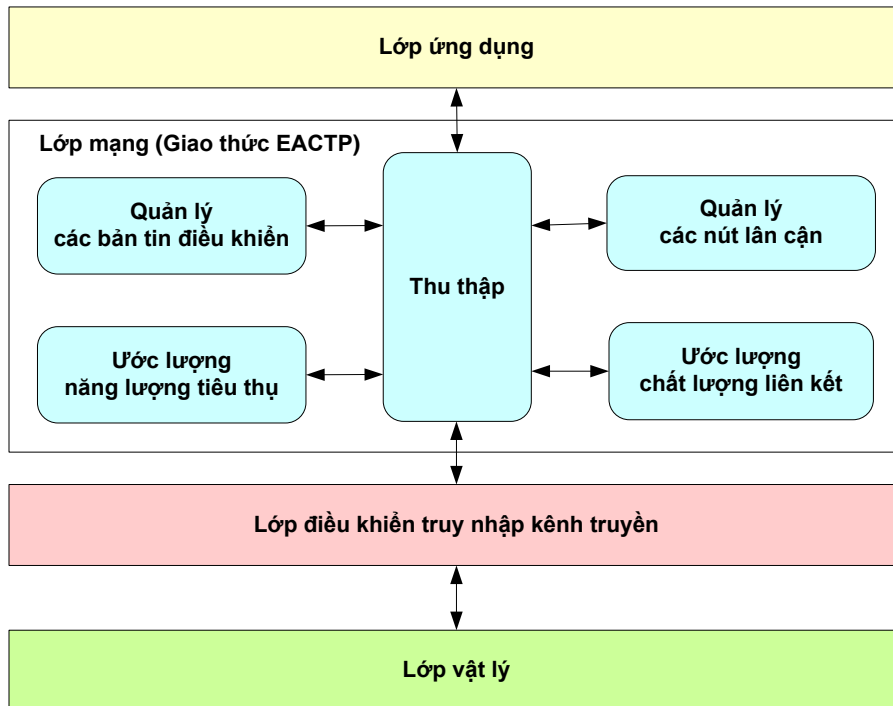
Trong luận án này, tác giả lựa chọn hệ điều hành Contiki để phục vụ cho việc nghiên cứu với mạng cảm biến không dây. Tác giả lựa chọn hệ điều hành Contiki bởi vì có ba lý do chính sau: Thứ nhất, hệ điều hành Contiki là một trong những hệ điều hành phổ biến nhất trong cộng đồng nghiên cứu về mạng cảm biến không dây hiện nay; Thứ hai, hệ điều hành Contiki được đánh giá là rất phù hợp với các mạng cảm biến không dây nói riêng và mạng các thiết bị nhúng nói chung. Vì vậy, hệ điều hành này đã được triển khai nhiều trong thực tế; Thứ ba là ngôn ngữ lập trình được sử dụng trong hệ điều hành Contiki là ngôn ngữ lập trình C. Đây là ngôn ngữ lập trình rất phổ biến và tác giả cũng đã có một số kinh nghiệm với ngôn ngữ lập trình này.

### **2.3.2. Thực thi giao thức EACTP trên hệ điều hành Contiki**

#### **2.3.2.1. Các thành phần của giao thức EACTP**

Tác giả thực thi giao thức EACTP trên hệ điều hành Contiki. Giao thức EACTP được xây dựng trên ngăn xếp truyền thông RIME trong hệ điều hành Contiki. Hình 2.6 minh họa các thành phần chính của giao thức EACTP.

- **Khối ước lượng chất lượng liên kết (*Collect-link-estimate.c, Collect-link-estimate.h*):** Khối ước lượng chất lượng liên kết thực hiện việc tính toán chất lượng liên kết giữa hai nút lân cận. Khối này tính toán chất lượng liên kết giữa hai nút mạng dựa trên sự thống kê các bản tin điều khiển nhận được và số bản tin dữ liệu được truyền thành công giữa hai nút mạng. Khối này là sự kế thừa của khối ước lượng chất lượng liên kết của giao thức CTP ban đầu.
- **Khối quản lý các bản tin điều khiển (*Announcement.h, Announcement.c*):** Khối quản lý các bản tin điều khiển có nhiệm vụ phát quảng bá thông tin về thước đo định tuyến của mỗi nút. Để thực hiện nhiệm vụ này, một bộ định thời được sử dụng để định kỳ phát quảng bá các bản tin điều khiển. Bản tin điều khiển mang thông tin về địa chỉ và các thước đo định tuyến của nút gửi.



**Hình 2.6:** Các thành phần chính của giao thức EACTP.

Khối này có sự mở rộng so với khối quản lý các bản tin điều khiển của giao thức CTP ban đầu đó là: Bổ sung thêm 2 bit mang thông tin về trạng thái năng lượng *ES* trong trường dự trữ của cấu trúc bản tin điều khiển.

- **Khối quản lý các nút lân cận (*Collect-neighbor.h*, *Collect-neighbor.c*):** Khối quản lý các nút lân cận lưu trữ thông tin của các nút lân cận trong bảng định tuyến. Các thông tin này bao gồm địa chỉ của nút lân cận, thước đo định tuyến chất lượng liên kết *ETX* và trạng thái năng lượng còn lại *ES* của các nút lân cận. Khối này chứa một bộ định thời để định kỳ xóa các nút lân cận trong bảng định tuyến khi các số liệu về thước đo định tuyến của nút đó không được cập nhật trong một khoảng thời gian dài. Ngoài ra, một thuật toán lựa chọn tuyến đường tối ưu cũng được thực thi trong khối này.

Khối này có sự mở rộng so với giao thức CTP ban đầu đó là: Bổ sung thêm trường lưu thông tin về trạng thái năng lượng của các nút lân cận trong bảng định tuyến; Thay đổi thuật toán lựa chọn tuyến đường mới có sự kết hợp giữa chất lượng liên kết của tuyến đường và trạng thái năng lượng của nút chuyển tiếp.

- **Khối ước lượng năng lượng tiêu thụ (*Energest.h*, *Energest.c*):** Khối này có nhiệm vụ ước lượng năng lượng tiêu thụ trên nút cảm biến.

Khối này được bổ sung thêm mới so với giao thức CTP ban đầu.

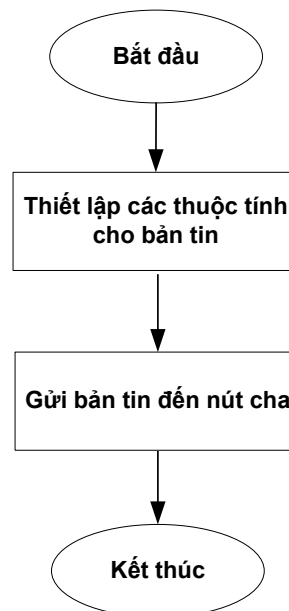
- **Khối thu thập (Collect.h, Collect.c):** Đây là khối trung tâm của giao thức EACTP. Khối này cung cấp các giao tiếp cho các lớp trên và lớp dưới. Khối này cũng thực hiện việc khởi tạo cho các khối còn lại. Nó thực hiện việc gửi và nhận các bản tin dữ liệu cũng như các bản tin xác nhận. Nếu có sự thay đổi về vị trí của nút hiện tại trong cấu trúc cây định tuyến thì khối này sẽ tác động đến khối quản lý các bản tin điều khiển để phát quảng bá các bản tin điều khiển nhằm thông báo cho các nút lân cận biết về những thay đổi này. Để tránh xảy ra hiện tượng trùng lặp các bản tin dữ liệu thì mỗi nút sẽ kiểm tra bản tin dữ liệu cần được chuyển tiếp với các bản tin dữ liệu vừa mới được chuyển tiếp gần nhất. Nếu bản tin này đã được chuyển tiếp thì nó sẽ bị loại bỏ. Khối này có sự sửa đổi so với khối thu thập của giao thức CTP ban đầu để gọi các hàm xử lý trong các khối quản lý bản tin điều khiển, quản lý nút lân cận và khối ước lượng năng lượng tiêu thụ.

#### 2.3.2.2. Hoạt động của giao thức EACTP

- **Khởi tạo:** Khi giao thức cây thu thập được khởi tạo thì nút gốc (Sink) có  $rtmetric = 0$ , các nút còn lại có giá trị  $rtmetric$  bằng  $RTMETRIC\_MAX = 511$ , năng lượng ban đầu  $E_0$  bằng dung lượng ban đầu của pin, chỉ số năng lượng ban đầu  $EI = 100\%$  ( $ES = 00$ ), chất lượng liên kết giữa các nút ban đầu được khởi tạo bằng  $INITIAL\_LINK\_ESTIMATE = 16$ . Việc khám phá các nút lân cận được bắt đầu (bằng việc đăng ký với mô đun phát bản tin điều khiển) trên một kênh riêng. Nút gốc sẽ phát quảng bá thước đo định tuyến ( $rtmetric, ES$ ) của nó cho các nút xung quanh. Các nút trong vùng phủ sóng của nút gốc sẽ lưu nút gốc vào trong bảng định tuyến và chọn nút gốc làm nút cha để gửi bản tin dữ liệu. Sau khi cập nhật lại thước đo định tuyến  $rtmetric$  và trạng thái năng lượng  $ES$  thì các nút trong vùng phủ sóng của nút gốc lại phát quảng bá bản tin điều khiển để gửi thước đo định tuyến ( $rtmetric, ES$ ) của chúng cho các nút lân cận. Quá trình cứ thế tiếp diễn cho đến khi mỗi nút đều nhận biết được thông tin về địa chỉ và thước đo định tuyến của các nút lân cận. Mỗi nút sẽ lựa chọn một nút lân cận tốt nhất trong bảng định tuyến làm nút cha theo thuật toán 2.3.
- **Gửi các bản tin:** Hình 2.7 là lưu đồ thuật toán xử lý sự kiện lớp ứng dụng gửi một bản tin dữ liệu.

Thuật toán gửi một bản tin đến nút gốc được hoạt động như sau:

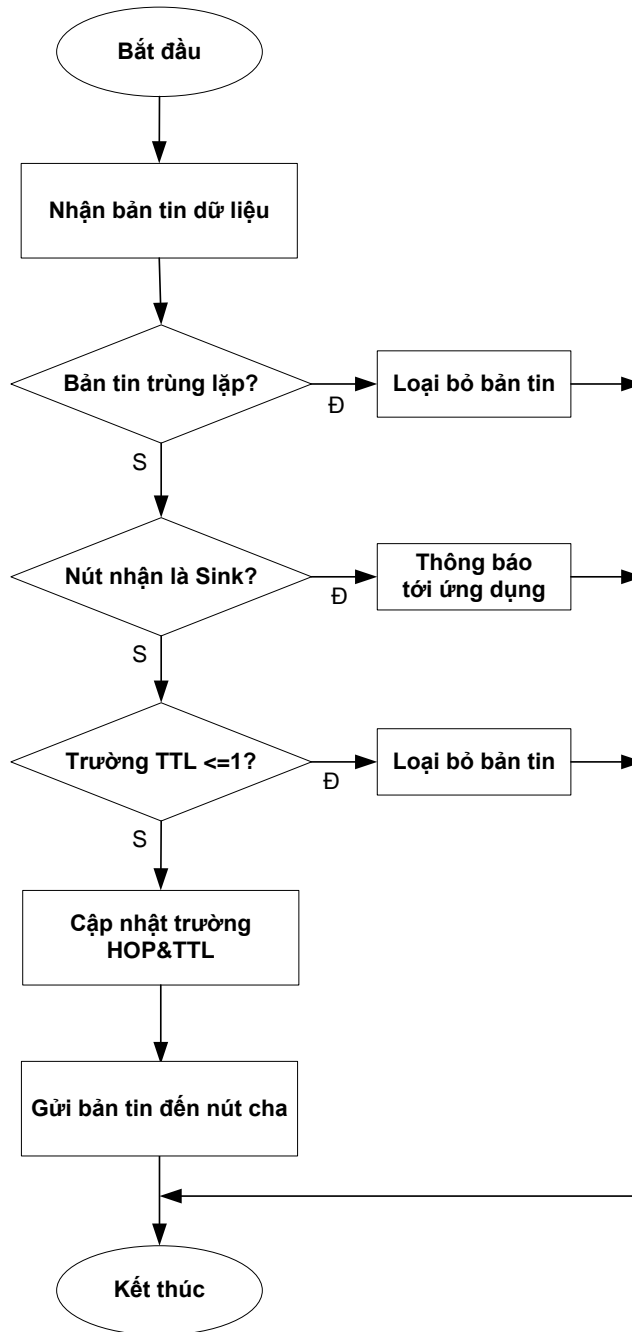
- ✓ Đầu tiên tất cả các thuộc tính của giao thức cây thu thập được thiết lập.
- ✓ Sau đó, mỗi nút gửi bản tin dữ liệu đến nút cha sử dụng mô đun reliable unicast trong ngăn xếp truyền thông RIME. Những mô tả chi tiết về ngăn xếp truyền thông RIME được trình bày ở phụ lục 2. Mô đun reliable unicast sẽ gửi một bản tin dữ liệu đến nút lân cận. Mô đun này sẽ thực hiện gửi một bản tin đến nút lân cận thông qua việc truyền lại một số lần lớn nhất có thể cho đến khi nhận được bản tin xác nhận ACK từ nút nhận. Nếu truyền bản tin thành công thì nó sẽ thông báo cho lớp trên (trong trường hợp này là giao thức thu thập). Ngược lại, nếu không thành công (khi bản tin không nhận được ACK sau một số lần truyền) tức là bản tin dữ liệu bị quá hạn (Timeout) và sự quá hạn này được thông báo cho lớp trên.



**Hình 2.7:** Quá trình xử lý sự kiện lớp ứng dụng gửi một bản tin dữ liệu.

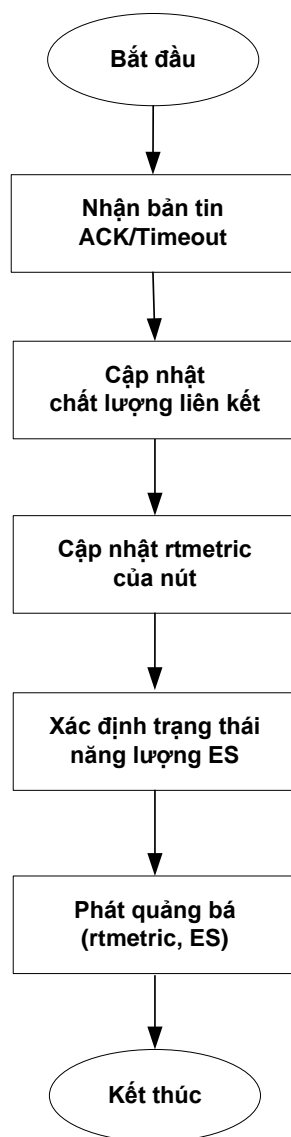
- ✓ Khi bản tin gửi đi được xác nhận bởi một bản tin ACK từ nút cha hoặc nút gửi nhận được một thông báo quá hạn thì giá trị *rtmetric* của nút gửi cùng với chất lượng liên kết *ETX* giữa nút gửi và nút nhận sẽ được cập nhật lại.
- **Nhận bản tin dữ liệu:** Hình 2.8 là lưu đồ thuật toán xử lý sự kiện nút nhận được một bản tin dữ liệu. Khi một nút nhận được một bản tin dữ liệu, đầu tiên nút nhận sẽ thực hiện thủ tục kiểm tra xem bản tin dữ liệu nhận được có bị

trùng lặp hay không. Bản tin dữ liệu sẽ được kiểm tra với các bản tin dữ liệu vừa mới được chuyển tiếp. Nếu nhận dạng bản tin và địa chỉ nút khởi nguồn giống với một bản tin dữ liệu vừa mới được chuyển tiếp gần đây thì bản tin dữ liệu đó sẽ được loại bỏ. Nếu bản tin không bị trùng lặp thì các thuộc tính của bản tin này sẽ được thêm vào bảng kiểm tra sự trùng lặp bản tin. Bảng này lưu một số thuộc tính của các bản tin vừa mới được chuyển tiếp gần nhất.



**Hình 2.8:** Quá trình xử lý sự kiện nút nhận một bản tin dữ liệu.

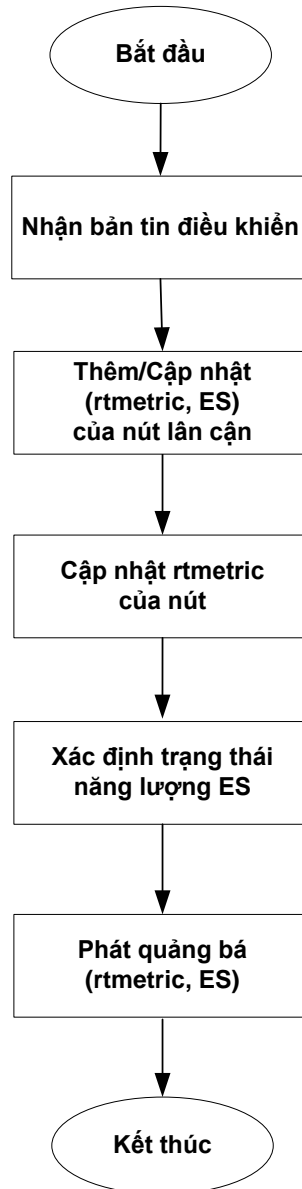
- ✓ Nếu nút nhận được bản tin là nút gốc thì ứng dụng sử dụng giao thức cây thu thập sẽ được thông báo về sự tiếp nhận một bản tin.
  - ✓ Nếu nút nhận bản tin không phải là nút gốc thì bản tin sẽ được chuyển tiếp.
    - Nếu trường TTL nhỏ hơn hoặc bằng 1 thì bản tin được loại bỏ.
    - Trường HOP được tăng lên một và trường TTL giảm đi một.
    - Bản tin được chuyển tiếp đến nút cha của nó sử dụng mô đun truyền thông reliable unicast trong ngăn xếp truyền thông RIME.
- **Nhận bản tin ACK/Timeout:** Hình 2.9 là lưu đồ thuật toán xử lý sự kiện nút nhận được một bản tin ACK hoặc Timeout.



**Hình 2.9:** Quá trình xử lý sự kiện nút nhận một bản tin ACK/Timeout.

Khi nút nhận được bản tin ACK/Timeout thì chất lượng liên kết giữa nút gửi và nút nhận sẽ được cập nhật. Nút nhận cập nhật lại thước đo chất lượng tuyến đường ( $rtmetric$ ) và xác định lại trạng thái năng lượng còn lại ( $ES$ ). Sau đó, nút nhận sử dụng bản tin điều khiển để phát quảng bá giá trị hai thước đo định tuyến ( $rtmetric$ ,  $ES$ ) mới cho các nút lân cận.

- **Nhận bản tin điều khiển:** Hình 2.10 là lưu đồ thuật toán xử lý sự kiện một nút nhận được một bản tin điều khiển từ một nút lân cận.

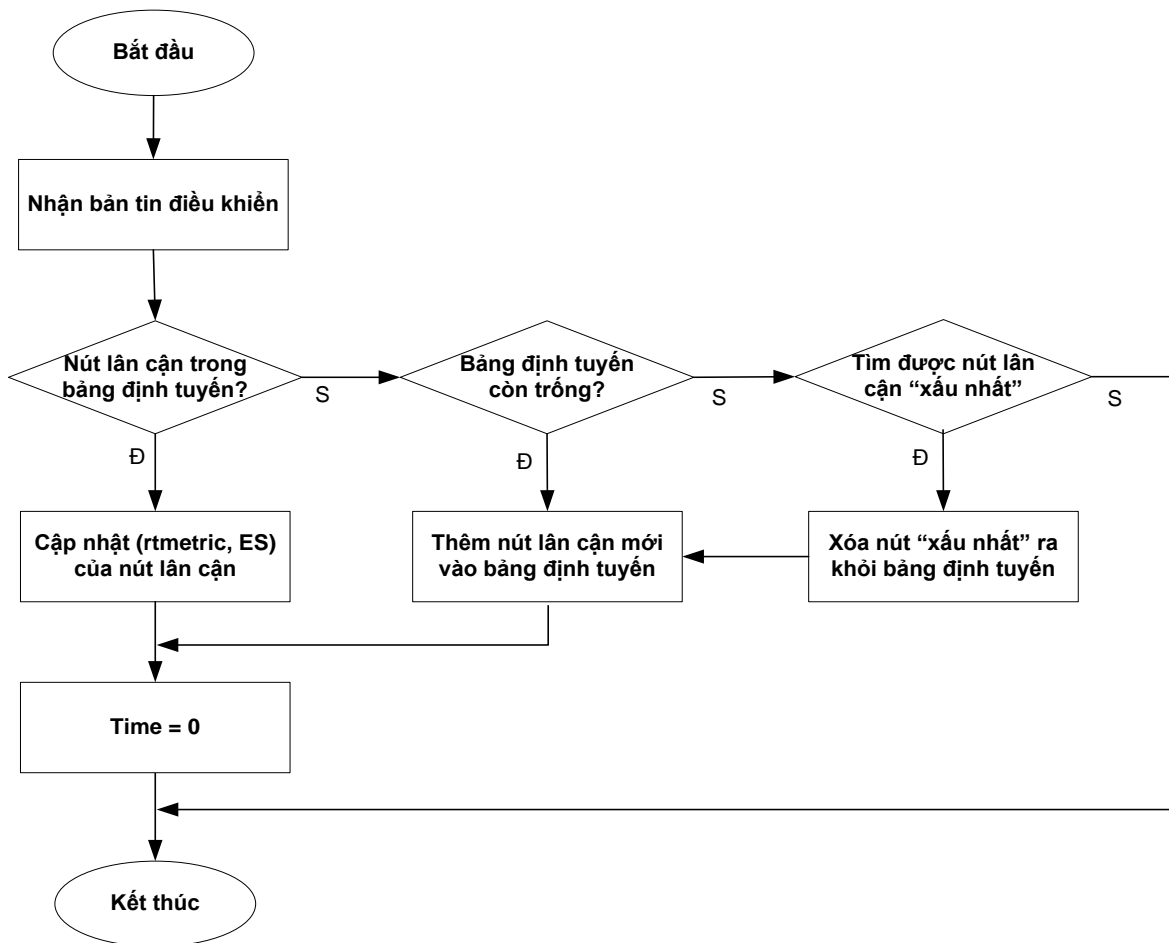


**Hình 2.10:** Quá trình xử lý sự kiện nút nhận một bản tin điều khiển.

- Nút nhận sẽ kiểm tra thông tin của nút lân cận trong bảng định tuyến. Hình 2.11 là lưu đồ thuật toán Thêm/Cập nhật thước đo định tuyến ( $rtmetric$ ,  $ES$ )

của nút lân cận. Nếu nút lân cận chưa có trong bảng định tuyến thì nó sẽ được thêm vào bảng định tuyến. Nếu nút lân cận đã có trong bảng định tuyến thì hai thước đo định tuyến ( $rtmetric$ ,  $ES$ ) của nó sẽ được cập nhật trong bảng định tuyến. Nếu bảng định tuyến đầy thì nút lân cận “xấu nhất” trong bảng định tuyến bị loại bỏ và thay thế vào đó là một nút lân cận mới. Nút lân cận “xấu nhất” được xác định là nút có thước đo định tuyến ( $rtmetric$ ) đến nút gốc là lớn nhất và lớn hơn  $rtmetric$  của nút mới được thêm vào.

- Nút nhận bản tin điều khiển cũng sẽ cập nhật lại thước đo định tuyến  $rtmetric$  và xác định lại trạng thái năng lượng còn lại ( $ES$ ). Sau đó, nút nhận sử dụng bản tin điều khiển để phát quảng bá giá trị hai thước đo định tuyến ( $rtmetric$ ,  $ES$ ) mới của nó cho các nút lân cận.



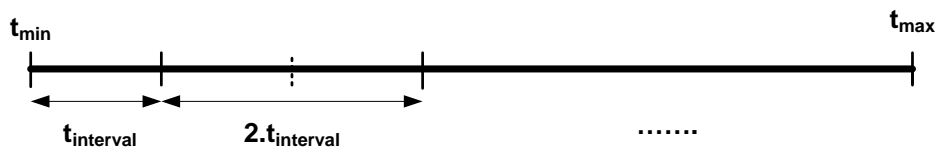
**Hình 2.11:** Lưu đồ thuật toán Thêm/Cập nhật ( $rtmetric$ ,  $ES$ ) của nút lân cận.

- Trong trường hợp có sự cập nhật ( $rtmetric$ ,  $ES$ ) của nút lân cận trong bảng định tuyến hoặc có một nút lân cận mới được thêm vào thì biến Time ứng với nút được cập nhật/nút mới được thêm vào được thiết lập lại bằng 0. Biến



Time được sử dụng để xác định một nút tồn tại quá lâu trong bảng định tuyến và không được dùng đến (không có sự cập nhật thước đo định tuyến *rtmetric*, *ES*). Sau một khoảng thời gian  $\text{Time} = 3600\text{s}$ , những nút không được sử dụng đến sẽ được tự động loại bỏ ra khỏi bảng định tuyến.

- **Gửi bản tin điều khiển:** Cũng giống như giao thức CTP ban đầu, các bản tin điều khiển được cập nhật và được gửi đi trong các trường hợp sau:
  - ✓ Có sự cập nhật (*rtmetric*, *ES*) của nút.
  - ✓ Khi một nút không xác định được nút cha ( $\text{parent} = \text{NULL}$ ) thì nút đó sẽ gửi đi một bản tin điều khiển có  $\text{rtmetric} = \text{RTMETRIC\_MAX}$  để yêu cầu thông tin định tuyến của các nút lân cận.
  - ✓ Bản tin điều khiển được gửi định kỳ trong khoảng thời gian từ  $t_{\min} = 60\text{s}$  đến  $t_{\max} = 3600\text{s}$ . Khoảng thời gian giữa hai lần gửi được thiết lập ban đầu  $t_{\text{interval}} = 8\text{s}$ . Sau mỗi lần gửi bản tin điều khiển, khoảng thời gian này sẽ được tăng lên gấp đôi cho đến giá trị giới hạn là  $t_{\max}$ . Như vậy, trong khoảng thời gian đầu mới tham gia vào mạng, bản tin điều khiển được gửi với tần suất cao để các nút trong mạng có thể nhận biết được nhau. Sau khi mạng ổn định thì tần suất gửi bản tin điều khiển sẽ được giảm đi để tiết kiệm năng lượng. Hình 2.12 minh họa cơ chế này.



**Hình 2.12:** Khoảng thời gian gửi bản tin điều khiển định kỳ.

- **Quá trình định tuyến lại:** Quá trình định tuyến lại tức là quá trình thực hiện lại thuật toán 2.3. Cũng giống như giao thức CTP ban đầu, quá trình định tuyến lại được thực hiện khi có sự cập nhật lại (*rtmetric*, *ES*) của một nút trong bảng định tuyến.

## 2.4. Đánh giá giao thức EACTP dựa trên mô phỏng

Phần này của luận án sẽ trình bày về một số kết quả mô phỏng đánh giá và so sánh hiệu năng giữa giao thức EACTP mới đề xuất và giao thức CTP ban đầu. Để mô phỏng và đánh giá giao thức EACTP, tác giả sử dụng công cụ mô phỏng Cooja. Cooja là một công cụ mô phỏng được viết bằng ngôn ngữ Java. Đây là công cụ được phát triển để mô phỏng các nút cảm biến hoạt động với hệ điều hành Contiki. Cooja

cũng là một công cụ mô phỏng rất gần với các nút cảm biến được triển khai trong thực tế. Mỗi nút cảm biến trong mạng mô phỏng không chỉ khác nhau về phần mềm mà còn khác nhau về cả phần cứng. Công cụ Cooja có thể mô phỏng được nhiều cấp độ khác nhau như cấp độ mạng, cấp độ hệ điều hành và cấp độ tập lệnh mã máy.

#### 2.4.1. Các tham số đánh giá

Tác giả đánh giá và so sánh hiệu năng giữa giao thức EACTP và giao thức CTP thông qua một số thước đo đánh giá sau.

##### 2.4.1.1. Tỷ lệ các nút còn sống trong mạng

Tỷ lệ các nút còn sống trong mạng *ANR (Alive Node Ratio)* được xác định bằng tỷ số giữa số nút còn sống trong mạng và tổng số nút ban đầu trong mạng.

$$ANR(\%) = \frac{N_{alive\_nodes}}{N} \cdot 100\% \quad (2.9)$$

Trong đó:

- $N_{alive\_nodes}$ : Là tổng số nút còn sống trong mạng.
- $N$ : Là tổng số nút ban đầu trong mạng.

##### 2.4.1.2. Tỷ lệ chuyển phát bản tin dữ liệu

Tỷ lệ chuyển phát bản tin dữ liệu *DDR (Data Delivery Ratio)* được xác định bằng tỷ số giữa số bản tin dữ liệu nhận được tại nút gốc và tổng số bản tin dữ liệu được gửi đi bởi tất cả các nút trong mạng.

$$DDR(\%) = \frac{N_{received}}{N_{data}} \cdot 100\% \quad (2.10)$$

Trong đó:

- $N_{received}$ : Là tổng số bản tin dữ liệu nhận được tại nút gốc.
- $N_{data}$ : Là tổng số bản tin dữ liệu được gửi bởi tất cả các nút trong mạng.

Tỷ lệ chuyển phát bản tin dữ liệu (DDR) càng cao phản ánh hiệu quả truyền thông trong mạng càng tốt và tỷ lệ các nút còn sống trong mạng càng cao.

##### 2.4.1.3. Sự cân bằng năng lượng giữa các nút trong mạng

Để đánh giá sự cân bằng năng lượng giữa các nút mạng, tác giả dựa vào chỉ số năng lượng còn lại *EI* trên các nút mạng. Thước đo đánh giá sự cân bằng năng lượng *EIB (Energy Indicator Balance)* giữa các nút trong mạng được xác định theo công thức sau:

$$EIB(\%) = \sqrt{\sum_{i=1}^N (\overline{EI} - EI_i)^2} (\%) \quad (2.11)$$

Trong đó:  $\overline{EI}$  là chỉ số năng lượng còn lại trung bình trên các nút mạng.

#### **2.4.1.4. Thời gian sống của mạng**

Thời gian sống của mạng có thể được định nghĩa là khoảng thời gian bắt đầu một truyền dẫn đầu tiên ở trong mạng và kết thúc khi tỷ lệ phần trăm các nút hết năng lượng dưới một ngưỡng cho trước. Giá trị ngưỡng được thiết lập tùy thuộc vào từng ứng dụng (có thể là 100% hoặc thấp hơn).

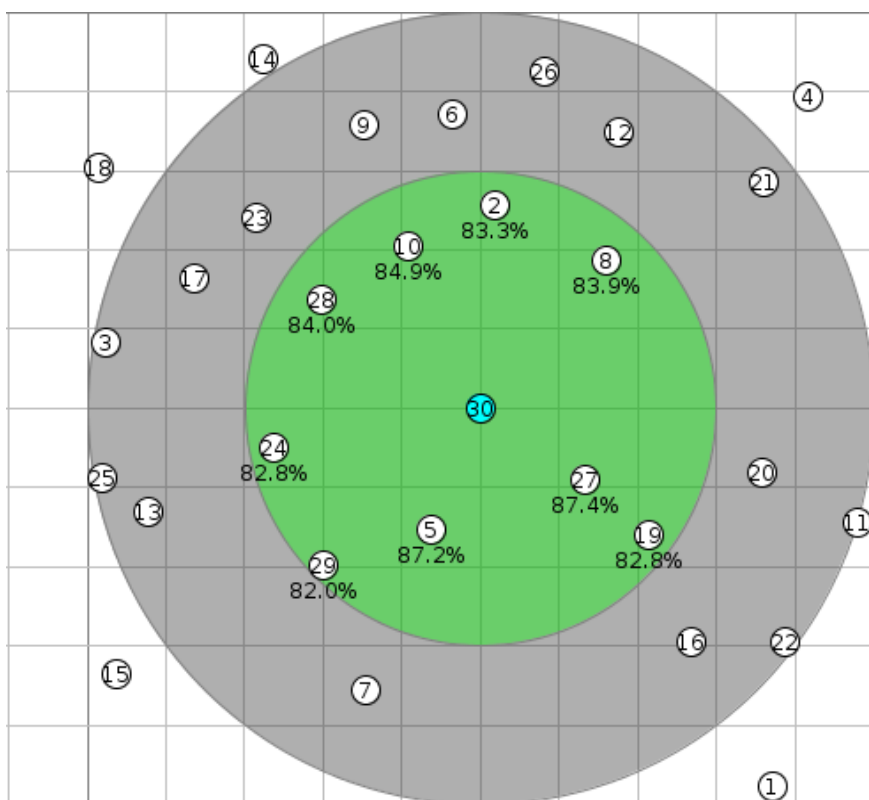
Định nghĩa này có liên quan đến thời gian sống của một nút mạng và không xét đến vai trò cụ thể của các nút mạng bị hết năng lượng. Nếu tỷ lệ phần trăm được thiết lập là 100% thì thời điểm nút đầu tiên trong mạng hết năng lượng cũng là thời điểm kết thúc thời gian sống của mạng [33].

### **2.4.2. Mô hình đánh giá mô phỏng 1**

#### **2.4.2.1. Kịch bản đánh giá**

Với các giả thiết đã được đặt ra trong mục 1.5 của luận án, một mô hình cụm gồm 30 nút cảm biến được phân bố ngẫu nhiên trong mặt phẳng có kích thước 100m x 100m. Các nút mạng định kỳ gửi bản tin dữ liệu về nút gốc là nút số 30. Hình 2.13 minh họa mô hình cụm này.

Các tham số được sử dụng trong suốt thời gian đánh giá mô phỏng được tóm tắt ở bảng 2.5. Mô hình truyền thông vô tuyến được sử dụng trong mô phỏng là mô hình truyền thông UDI, trong đó phạm vi truyền thông hiệu quả là 30m và phạm vi ảnh hưởng của nhiễu là 50m. Các nút số 2, 5, 8, 10, 19, 24, 27, 28, 29 có thể truyền thông trực tiếp đến nút gốc (nút số 30). Các nút còn lại phải thực hiện truyền thông theo mô hình đa chặng đến nút gốc. Giao thức lớp MAC được sử dụng trong kịch bản mô phỏng là giao thức ContikiMAC. Phụ lục 3 giới thiệu về cơ chế hoạt động của giao thức này. Các bước thực hiện đánh giá mô phỏng được tác giả trình bày chi tiết ở Phụ lục 4 của luận án.



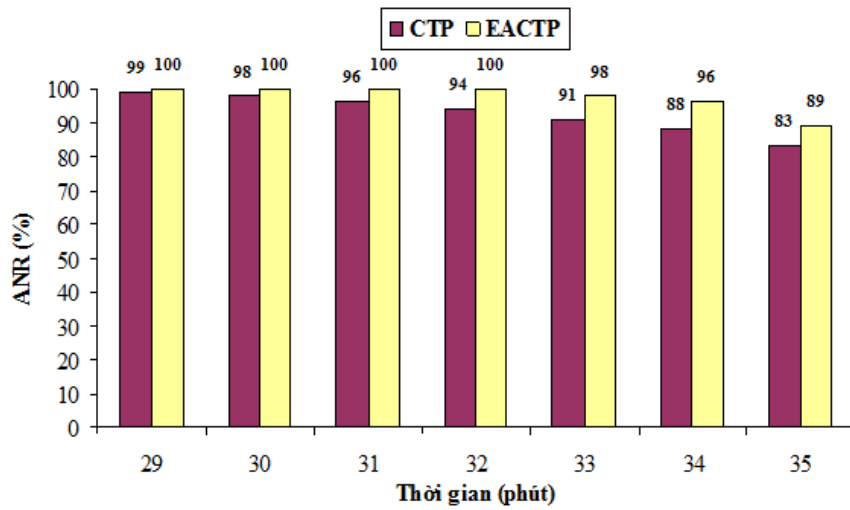
**Hình 2.13:** Mô hình mô phỏng một cụm gồm 30 nút.

**Bảng 2.5:** Kịch bản đánh giá mô phỏng 1.

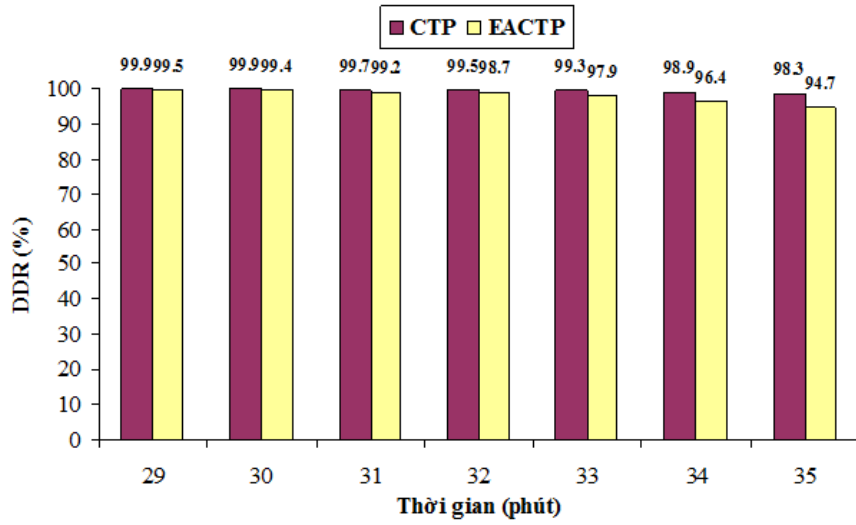
Các tham số	Giá trị
Mô hình truyền thông vô tuyến	UDI (mục 2.1.2)
Số nút mạng (nút)	30
Kích thước mạng (m x m)	100 x 100
Năng lượng ban đầu của mỗi nút	10J
Công suất phát (dBm)	0
Phạm vi phủ sóng của nút (m)	Phạm vi truyền hiệu quả: 30 Phạm vi ảnh hưởng của nhiễu: 50
Chu kỳ gửi bản tin dữ liệu (giây)	20
Nguồn gửi bản tin dữ liệu	Tất cả các nút trong mạng (ngoài nút gốc)
Giao thức lớp MAC	CSMA/ContikiMAC

#### 2.4.2.2. Kết quả đánh giá

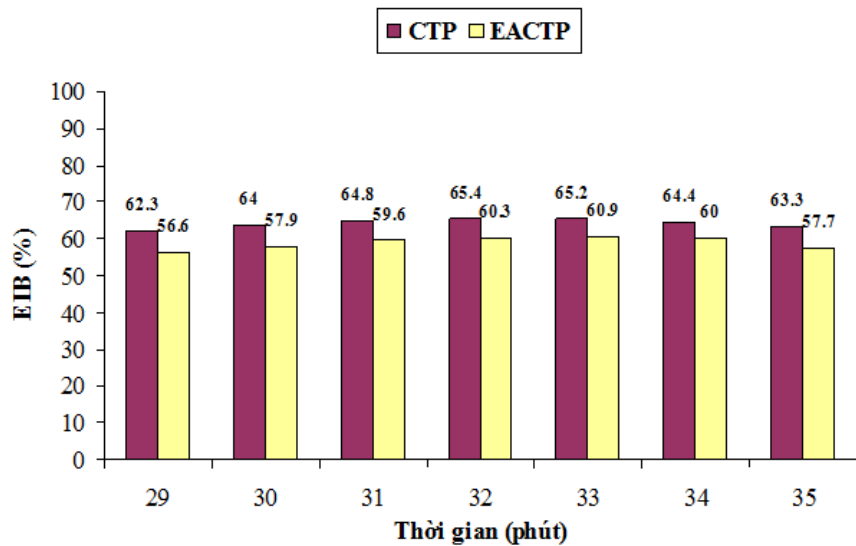
Hình 2.14, 2.15, 2.16 lần lượt là kết quả mô phỏng đánh giá so sánh tỷ lệ các nút còn sống trong mạng, tỷ lệ chuyển phát bản tin dữ liệu và sự cân bằng năng lượng giữa các nút mạng đối với giao thức CTP và giao thức EACTP trong kịch bản mô phỏng 1.



**Hình 2.14:** So sánh tỷ lệ các nút còn sống trong mạng.



**Hình 2.15:** So sánh tỷ lệ chuyển phát bản tin dữ liệu.

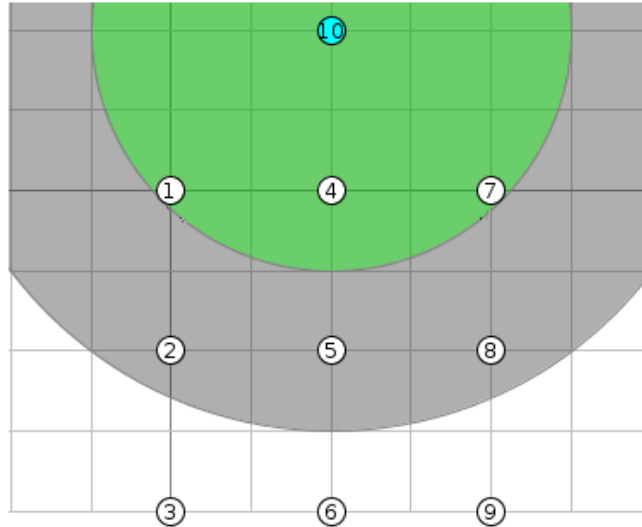


**Hình 2.16:** So sánh sự cân bằng năng lượng giữa các nút mạng.

### 2.4.3. Mô hình đánh giá mô phỏng 2

#### 2.4.3.1. Kích bản đánh giá

Trong kịch bản đánh giá mô phỏng 2, một cụm thu nhỏ gồm 10 nút mạng được xét đến (gần giống với kịch bản đánh giá thực nghiệm). Hình 2.17 là mô hình mô phỏng mạng gồm 10 nút. Các nút mạng định kỳ gửi bản tin dữ liệu về nút gốc là nút số 10. Các nút số 1, 4, 7 có thể truyền thông trực tiếp với nút gốc. Các nút còn lại phải thực hiện truyền thông theo mô hình đa chặng đến nút gốc.



**Hình 2.17:** Mô hình mô phỏng một cụm nhỏ gồm 10 nút.

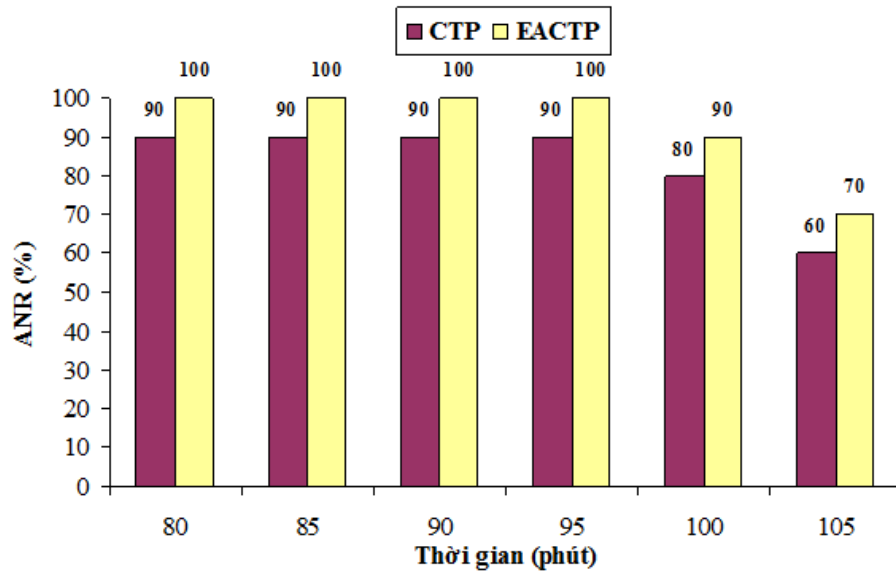
Các tham số được sử dụng trong suốt thời gian đánh giá mô phỏng được tóm tắt ở bảng 2.6. Các bước thực hiện đánh giá mô phỏng được tác giả trình bày chi tiết ở Phụ lục 4 của luận án.

**Bảng 2.6:** Kích bản đánh giá mô phỏng 2.

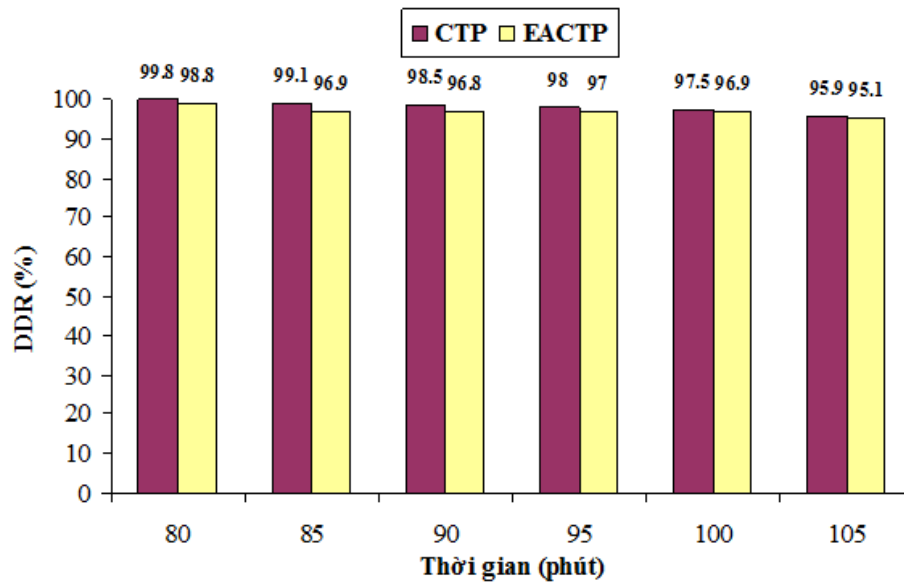
Các tham số	Giá trị
Mô hình truyền thông vô tuyến	UDI (mục 2.1.2)
Số nút mạng (nút)	10
Kích thước mạng (m x m)	60 x 80
Năng lượng ban đầu của mỗi nút	50J
Công suất phát (dBm)	0
Phạm vi phủ sóng của nút (m)	Phạm vi truyền hiệu quả: 30 Phạm vi ảnh hưởng của nhiễu: 50
Chu kỳ gửi bản tin dữ liệu (giây)	10
Nguồn gửi bản tin dữ liệu	Tất cả các nút trong mạng (ngoài nút gốc)
Giao thức lớp MAC	CSMA/ContikiMAC

### 2.4.3.2. Kết quả đánh giá

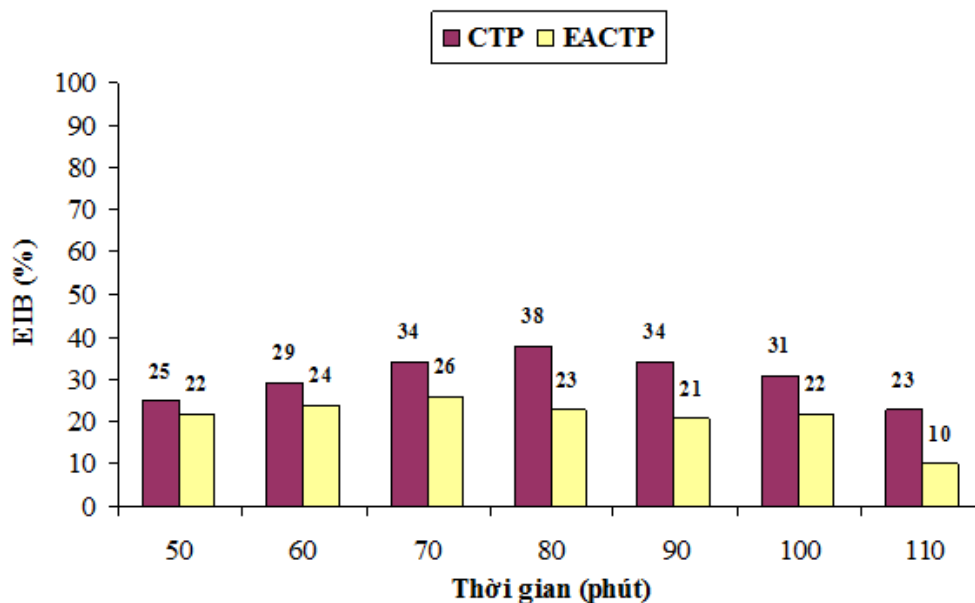
Hình 2.18, 2.19, 2.20 lần lượt là kết quả mô phỏng đánh giá so sánh tỷ lệ các nút còn sống trong mạng, tỷ lệ chuyển phát bản tin dữ liệu và sự cân bằng năng lượng giữa các nút mạng đối với giao thức CTP và giao thức EACTP trong kịch bản mô phỏng 2.



Hình 2.18: So sánh tỷ lệ các nút còn sống trong mạng.



Hình 2.19: So sánh tỷ lệ chuyển phát bản tin dữ liệu.



**Hình 2.20:** So sánh sự cân bằng năng lượng giữa các nút mạng.

#### 2.4.4. Nhận xét

Từ các kết quả đánh giá mô phỏng:

- Hình 2.14, 2.15, 2.18, 2.19 cho thấy tỷ lệ các nút còn sống trong mạng của giao thức EACTP cao hơn so với giao thức CTP. Tuy nhiên, giao thức EACTP vẫn đảm bảo được tỷ lệ chuyển phát bản tin dữ liệu ở mức chấp nhận được so với giao thức CTP ban đầu.
- Nếu giá trị ngưỡng được thiết lập để xác định thời gian sống của mạng là 100% thì hình 2.14, 2.18 cũng cho thấy thời gian sống của mạng hoạt động theo giao thức EACTP được cải thiện đáng kể so với giao thức CTP: Trong kịch bản mô phỏng 1, thời điểm nút hết năng lượng đầu tiên đối với giao thức CTP là phút thứ 29 và đối với giao thức EACTP là phút thứ 33 (như vậy, thời gian sống của mạng tăng 14%); Trong kịch bản mô phỏng 2, thời điểm nút hết năng lượng đầu tiên đối với giao thức CTP là phút thứ 80 và đối với giao thức EACTP là phút thứ 100 (thời gian sống của mạng tăng 25%).
- Hình 2.16, 2.10 là kết quả mô phỏng so sánh sự cân bằng năng lượng giữa các nút mạng. Kết quả mô phỏng cho thấy giao thức EACTP đảm bảo được sự cân bằng năng lượng giữa các nút mạng tốt hơn so với giao thức CTP ban đầu. Điều này được thể hiện bởi chỉ số EIB của giao thức EACTP thấp hơn so với chỉ số EIB của giao thức CTP.



- Kết quả mô phỏng cũng cho thấy: Khi tỷ lệ các nút còn sống trong mạng giảm thì tỷ lệ chuyển phát thành công bản tin dữ liệu (DDR) và chỉ số đánh giá sự cân bằng năng lượng (EIB) của cả hai giao thức CTP và giao thức EACTP cũng giảm theo. Kết quả mô phỏng hoàn toàn phù hợp với những lập luận, phân tích lý thuyết.

## **2.5. Đánh giá giao thức EACTP dựa trên cơ sở phân tích lý thuyết**

Khi so sánh giao thức EACTP với các giao thức định tuyến có sự nhận thức về năng lượng khác (đã được trình bày ở mục 1.6 của luận án) thì giao thức EACTP có một số ưu điểm sau:

- Thứ nhất, giao thức EACTP không làm phát sinh thêm chi phí về năng lượng trong việc gửi các bản tin điều khiển bởi vì giao thức này chỉ sử dụng 2 bit dự trữ trong cấu trúc bản tin điều khiển.
- Thứ hai, mô hình năng lượng được xét đến trong giao thức EACTP đầy đủ hơn khi xét đến các thành phần tiêu thụ năng lượng chính trong cấu trúc phần cứng TU mote.
- Thứ ba, cơ chế ước lượng năng lượng tiêu thụ dựa trên phần mềm cho phép thực thi giao thức EACTP trên mọi nền tảng phần cứng mà không làm phát sinh thêm bất kỳ một chi phí mới nào về phần cứng.
- Thứ tư, tuyến đường tối ưu được lựa chọn dựa trên sự kết hợp giữa hai thước đo định tuyến là chất lượng liên kết và trạng thái năng lượng còn lại của nút chuyển tiếp. Tuyến đường tối ưu vừa đảm bảo được số lần truyền cần thiết là ít nhất giúp giảm nhiều phát sinh trong mạng, tiết kiệm được năng lượng của các nút mạng và cũng vừa đảm bảo sự cân bằng năng lượng giữa các nút chuyển tiếp thuộc những tuyến đường có chất lượng liên kết tốt giúp tăng thời gian sống của mạng.

Các kết quả phân tích, đánh giá so sánh giữa giao thức EACTP với một số giao thức định tuyến có sự nhận thức về năng lượng khác được tóm tắt ở bảng 2.7.

**Bảng 2.7:** Kết quả phân tích, đánh giá so sánh giữa giao thức EACTP với một số giao thức định tuyến có sự nhận thức về năng lượng khác.

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
EACTP	<p>Chỉ sử dụng 2 bit dự trữ trong bản tin điều khiển.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Không làm phát sinh thêm chi phí năng lượng.</li> </ul>	<p>Cơ chế ước lượng năng lượng dựa trên phần mềm và có xét đến các thành phần tiêu thụ năng lượng của: Bộ thu phát vô tuyến, bộ vi điều khiển, cảm biến.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Có thể thực thi giao thức trên mọi nền tảng phần cứng [29].</li> <li>- Có thể đạt độ chính xác cao đến 96% [39].</li> </ul>	<p>Dựa trên <b>chất lượng liên kết</b> toàn tuyến đường và <b>trạng thái năng lượng còn lại</b> của nút chuyển tiếp.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Tuyến đường tối ưu đảm bảo có chất lượng liên kết tốt, tiết kiệm được năng lượng khi gửi bản tin dữ liệu [49] và đảm bảo được sự cân bằng năng lượng giữa các nút chuyển tiếp thuộc những tuyến đường có chất lượng liên kết tốt.</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Tỷ lệ chuyển phát bản tin dữ liệu thấp hơn so với giao thức CTP ban đầu.</li> </ul>
MTPR [34]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng khi gửi thông tin về năng lượng tiêu thụ để truyền một bản tin giữa hai nút lân cận.</li> </ul>		<p>Dựa trên năng lượng tiêu thụ để truyền một bản tin giữa hai nút lân cận. Tuyến đường tối ưu là tuyến đường có <b>tổng năng lượng tiêu thụ trên toàn tuyến đường</b> để truyền bản tin từ nguồn đến đích là nhỏ nhất.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Tối thiểu năng lượng tiêu thụ trên mỗi bản tin.</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Không cung cấp thông tin về dung lượng còn</li> </ul>

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
			<p>lại của pin [49].</p> <ul style="list-style-type: none"> <li>- Một số nút mạng hết năng lượng nhanh hơn các nút mạng khác [49].</li> </ul>
MBCR [35]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng để gửi thông tin về dung lượng còn lại của pin trên mỗi nút cảm biến.</li> </ul>		<p>Chi phí được xác định là <b>ngịch đảo dung lượng còn lại của pin</b> trên mỗi nút cảm biến. Tuyến đường tối ưu là tuyến đường có tổng chi phí của các nút chuyển tiếp trên toàn tuyến đường là nhỏ nhất.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Dung lượng còn lại của pin trên mỗi nút cảm biến được tính đến [49].</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Tuyến đường được lựa chọn có thể bao gồm một số nút mạng với dung lượng pin còn lại ở mức thấp [49]. Điều này dẫn đến việc một số nút mạng sẽ hết năng lượng nhanh hơn các nút mạng khác.</li> </ul>
EAR [36]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng để gửi thông tin về năng lượng còn lại của các nút cảm biến.</li> </ul>	<p>Thước đo về chi phí năng lượng dựa trên giả thiết rằng năng lượng còn lại và chi phí về năng lượng để truyền bản tin có thể xác định được chính xác dựa trên sự hỗ trợ về phần cứng [7].</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Không phát sinh chi phí về năng lượng để xử lý và tính toán năng</li> </ul>	<p>Dựa trên <b>chi phí chuyển tiếp một bản tin</b> và <b>tổng năng lượng còn lại</b> của các nút chuyển tiếp thuộc tuyến đường.</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Không xét đến sự khác nhau về năng lượng của từng nút riêng lẻ trên tuyến đường [7].</li> </ul>

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
		<p>lượng còn lại trên mỗi nút cảm biến.</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Không phải mọi nền tảng phân cứng đều hỗ trợ khả năng này.</li> <li>- Phát sinh thêm chi phí về giá thành phần cứng thiết bị.</li> </ul>	
E-Span [37]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng để gửi thông tin về mức năng lượng còn lại của các nút cảm biến.</li> </ul>	<p>Mô hình năng lượng chỉ xét đến năng lượng tiêu thụ của bộ thu phát vô tuyến [37].</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Mô hình năng lượng đơn giản, không gần với thực tế.</li> </ul>	<p>Dựa trên <b>mức năng lượng còn lại</b> của các nút lân cận và <b>số bước nhảy</b> đến nút gốc.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Thước đo số bước nhảy đơn giản, được sử dụng rộng rãi trong các mạng có dây [49].</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Thước đo số bước nhảy không phù hợp với các mạng không dây bởi vì trong môi trường vô tuyến, tuyến đường có số bước nhảy tối thiểu hoàn toàn có thể bao gồm các liên kết có chất lượng kém. Điều này gây lãng phí về năng lượng để truyền lại bản tin và làm tăng độ trễ truyền bản tin [49].</li> </ul>
LBEAC [38]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng để gửi thông tin về năng lượng còn lại của các nút cảm biến.</li> </ul>	<p>Cơ chế ước lượng năng lượng tiêu thụ dựa trên các hoạt động (truyền, nhận) của bộ thu phát vô tuyến [38].</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Mô hình năng lượng đơn giản, không gần với thực tế.</li> </ul>	<p>Dựa trên <b>năng lượng còn lại</b> của các nút lân cận và <b>trạng thái quá tải</b> của các nút lân cận [38].</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Có xét đến năng lượng còn lại của các nút mạng.</li> </ul>

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
			<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Không xét đến chất lượng liên kết của tuyến đường. Tuyến đường có thể bao gồm các liên kết có chất lượng kém, bản tin có thể phải truyền lại nhiều lần. Điều này gây lãng phí về năng lượng và tăng độ trễ truyền bản tin.</li> </ul>
ICTP [8]	<p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh chi phí về năng lượng để gửi thông tin về năng lượng còn lại của các nút cảm biến.</li> </ul>	<p>Mô hình năng lượng tiêu thụ đơn giản khi giả thiết năng lượng ban đầu của mỗi nút cảm biến là 0xFFFF, năng lượng tiêu thụ khi gửi một bản tin dữ liệu là 30, năng lượng tiêu thụ khi gửi một bản tin điều khiển là 20 [8].</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Mô hình năng lượng đơn giản, dễ thực thi trong mô phỏng.</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Mô hình năng lượng không gần với thực tế.</li> </ul>	<p>Dựa trên <b>năng lượng còn lại</b> của các nút lân cận và <b>chất lượng liên kết</b> toàn tuyến đường.</p>
EQLR [50]	<p>Bổ sung thêm trường năng lượng còn lại để mang thông tin về mức năng lượng còn lại của pin.</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh thêm chi phí năng lượng trong việc gửi bản tin điều khiển.</li> </ul>	<p>Cơ chế ước lượng năng lượng dựa trên phần mềm được đề xuất bởi Adam Dunkels.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Có thể thực thi giao thức trên mọi nền tảng phần cứng [29].</li> <li>- Có thể đạt độ chính xác cao đến 96% [39].</li> </ul>	<p>Dựa trên <b>chất lượng liên kết</b> toàn tuyến đường và <b>mức năng lượng còn lại</b> của nút chuyển tiếp.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Tuyến đường tối ưu đảm bảo có chất lượng liên kết tốt và đảm bảo được sự cân bằng năng lượng giữa các nút chuyển tiếp thuộc những</li> </ul>

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
			<p>tuyến đường có chất lượng liên kết tốt.</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Ngưỡng năng lượng là một giá trị cố định và không có gì đảm bảo rằng đó là một giá trị tối ưu.</li> <li>- Ngưỡng <i>ETX</i> được xác định dựa trên thực nghiệm với 9 nút. Trong triển khai thực tế với số lượng nút lớn thì việc xác định ngưỡng <i>ETX</i> dựa trên thực nghiệm sẽ gặp nhiều khó khăn.</li> </ul>
ELR[51]	<p>Bổ sung thêm trường 16bit để mang thông tin về mức năng lượng còn lại.</p> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Phát sinh thêm chi phí năng lượng trong việc gửi bản tin điều khiển.</li> </ul>	<p>Cơ chế ước lượng năng lượng dựa trên phần mềm được đề xuất bởi Adam Dunkels.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Có thể thực thi giao thức trên mọi nền tảng phần cứng [29].</li> <li>- Có thể đạt độ chính xác cao đến 96% [39].</li> </ul>	<p>Dựa trên <b>chất lượng liên kết</b> toàn tuyến đường và <b>mức năng lượng còn lại (tính theo %)</b> của nút chuyển tiếp.</p> <p><b>Ưu điểm:</b></p> <ul style="list-style-type: none"> <li>- Tuyến đường tối ưu đảm bảo có chất lượng liên kết tốt và đảm bảo được sự cân bằng năng lượng giữa các nút chuyển tiếp thuộc những tuyến đường có chất lượng liên kết tốt.</li> </ul> <p><b>Nhược điểm:</b></p> <ul style="list-style-type: none"> <li>- Ngưỡng năng lượng là một giá trị cố định (10%) và không có gì đảm bảo rằng đó là một giá trị tối ưu.</li> <li>- Ngưỡng <i>ETXdiffTh</i> là một giá trị cố định (bằng 10). Trong triển khai thực tế, việc xác định ngưỡng <i>ETXdiffTh</i> là</li> </ul>

Giao thức	Đặc điểm		
	Chi phí năng lượng gửi bản tin điều khiển	Mô hình năng lượng	Thước đo định tuyến
			một khó khăn cần giải quyết.

## 2.6. Kết luận chương 2

Trong chương này, tác giả đã có 2 đề xuất mới đó là:

- Đề xuất mô hình toán học cho bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng dựa trên lý thuyết đồ thị.
- Đề xuất và thực thi một giao thức định tuyến mới đó là giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng EACTP.

Các kết quả đánh giá so sánh dựa trên phân tích và mô phỏng giữa giao thức EACTP mới đề xuất và giao thức CTP ban đầu cho thấy giao thức EACTP đạt được một số tiêu chí quan trọng đó là: Tăng được thời gian sống của các nút trong mạng nhưng vẫn đảm bảo được tỷ lệ chuyển phát bản tin dữ liệu trong mạng ở mức cao, đảm bảo được sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt, không làm phát sinh thêm chi phí mới về năng lượng trong việc gửi các bản tin điều khiển. Một phần kết quả của chương này đã được tác giả công bố trong các bài báo [1], [4].

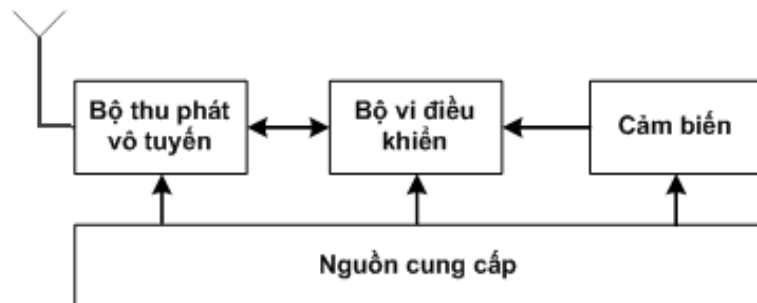
## Chương 3. TRIỂN KHAI ĐÁNH GIÁ THỰC NGHIỆM

Trong chương này, tác giả xây dựng một hệ thống triển khai thực nghiệm dựa trên 10 nút cảm biến phần cứng TUmote (Thainguyen University mote). Hệ thống triển khai thực nghiệm này cho phép tùy biến, tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của các giao thức này trong điều kiện thực tế. Một số kết quả đánh giá so sánh hiệu năng giữa giao thức EACTP và giao thức CTP thông qua một số mô hình triển khai thực nghiệm cũng được trình bày trong chương này.

### 3.1. Phần cứng cho mạng cảm biến không dây

#### 3.1.1. Cấu trúc phần cứng của một nút mạng cảm biến không dây

Hiện nay, trên thị trường có rất nhiều kiến trúc phần cứng khác nhau đã được nghiên cứu và phát triển cho mạng cảm biến không dây. Trong bài báo công bố số 5, tác giả đã trình bày về một nghiên cứu so sánh về các kiến trúc phần cứng này. Các thành phần chính trong kiến trúc phần cứng của một nút mạng cảm biến không dây được minh họa ở hình 3.1. Các thành phần chính bao gồm:



**Hình 3.1:** Các thành phần chính trong kiến trúc phần cứng của một nút cảm biến không dây.

##### 3.1.1.1. Thiết bị truyền thông

Thiết bị truyền thông cho phép các nút cảm biến có thể truyền thông với nhau. Thiết bị truyền thông thường là một bộ thu phát vô tuyến. Một bộ thu phát vô tuyến có cả hai chức năng đó là vừa gửi và vừa nhận bản tin vô tuyến.

Các loại bộ thu phát vô tuyến khác nhau thì có khả năng xử lý khác nhau. Một số bộ thu phát vô tuyến đơn giản chỉ làm nhiệm vụ gửi và nhận các bit thông tin, trong khi nhiều bộ thu phát vô tuyến có thể đóng gói thông tin thành các gói tin, định dạng phần tiêu đề và thậm chí cả mã hóa và giải mã dữ liệu bằng cách sử dụng phương pháp mã hóa bảo mật.



Trong số các thành phần phân cứng của một nút cảm biến không dây thì bộ thu phát vô tuyến là một trong những thành phần tiêu tốn năng lượng nhiều nhất. So sánh với mức tiêu thụ năng lượng của bộ vi điều khiển thì bộ thu phát vô tuyến thường sử dụng năng lượng lớn hơn nhiều lần. Nguyên nhân là do yêu cầu về xử lý điều chế và giải điều chế tín hiệu vô tuyến. Vì vậy, bộ thu phát vô tuyến cần phải được tắt để tiết kiệm năng lượng. Tuy nhiên, khi bộ thu phát vô tuyến được tắt thì nó không thể nhận được bất kỳ dữ liệu nào từ các nút lân cận. Để hình thành mạng đa chặng thì các bộ thu phát vô tuyến của tất cả các nút trong mạng phải có cách đồng bộ nào đó để chúng có thể nhận được dữ liệu trong khi vẫn bảo tồn được năng lượng.

#### **3.1.1.2. Bộ vi điều khiển**

Bộ vi điều khiển chạy các chương trình phần mềm của nút cảm biến. Vi điều khiển là một bộ vi xử lý có bộ nhớ trong, bộ định thời và phân cứng để kết nối với các thiết bị bên ngoài như các cảm biến và bộ thu phát vô tuyến.

Do hạn chế về giá thành và năng lượng nên các vi điều khiển được sử dụng trong các nút cảm biến đơn giản hơn nhiều so với các bộ vi xử lý được sử dụng trong các máy tính cá nhân. Thông thường, một bộ vi điều khiển được sử dụng trong nút cảm biến có vài chục KB bộ nhớ trong và hoạt động ở tốc độ xung nhịp một vài MHz. Trong khi đó, máy tính hiện đại có đến hàng Gbytes bộ nhớ và chạy ở xung nhịp một vài Ghz.

#### **3.1.1.3. Cảm biến**

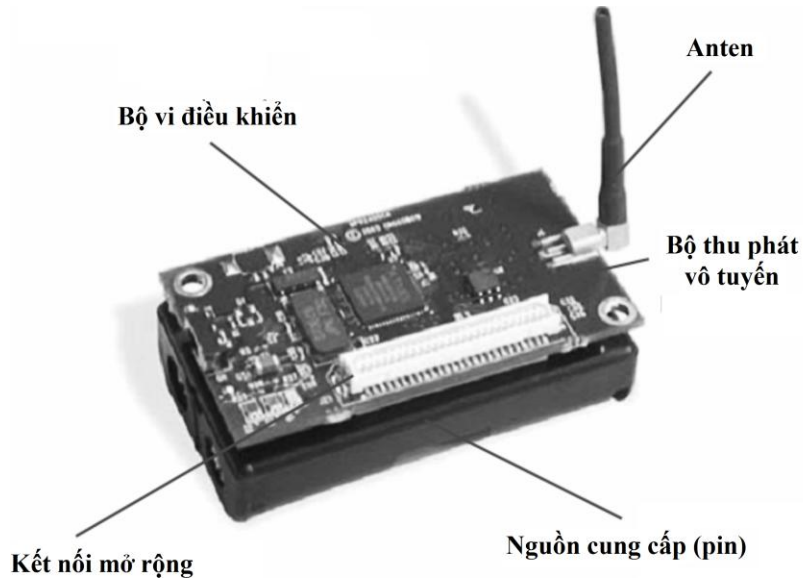
Các nút cảm biến không dây tương tác với môi trường vật lý nhờ các cảm biến. Các cảm biến được gắn trên nút cảm biến không dây có thể từ rất đơn giản đến rất phức tạp. Ví dụ: Các cảm biến nhiệt độ có thể là một biến trở có trở kháng thay đổi theo nhiệt độ môi trường là một loại cảm biến đơn giản; Trong khi đó, cảm biến nhiệt độ, độ ẩm SHT11 có dữ liệu đầu ra số lại là một ví dụ khác về loại cảm biến phức tạp với nhiều tính năng hơn.

#### **3.1.1.4. Nguồn cung cấp**

Một nút cảm biến không dây bao gồm nhiều thiết bị điện tử và các thiết bị điện tử cần năng lượng. Vì vậy, mỗi nút cảm biến cần một nguồn cung cấp năng lượng. Ngày nay, nguồn năng lượng phổ biến nhất là pin nhưng trong một số trường hợp sử dụng các tế bào năng lượng mặt trời, áp điện và các hình thức tích trữ năng lượng khác.

Pin là nguồn năng lượng phổ biến nhất cho các nút cảm biến không dây. Pin có nhiều hình dạng và kiểu dáng khác nhau. Khi lựa chọn pin cho các nút cảm biến, chúng ta cần quan tâm đến kích thước cũng như khả năng tích trữ năng lượng của pin. Các pin Lithium hiện nay là phổ biến nhất.

Bằng việc sử dụng các thiết bị điện tử tiêu thụ ít năng lượng và phần mềm tiết kiệm năng lượng, một nút cảm biến có thể có thời gian tồn tại nhiều năm trên một pin kích thước tiêu chuẩn AA.



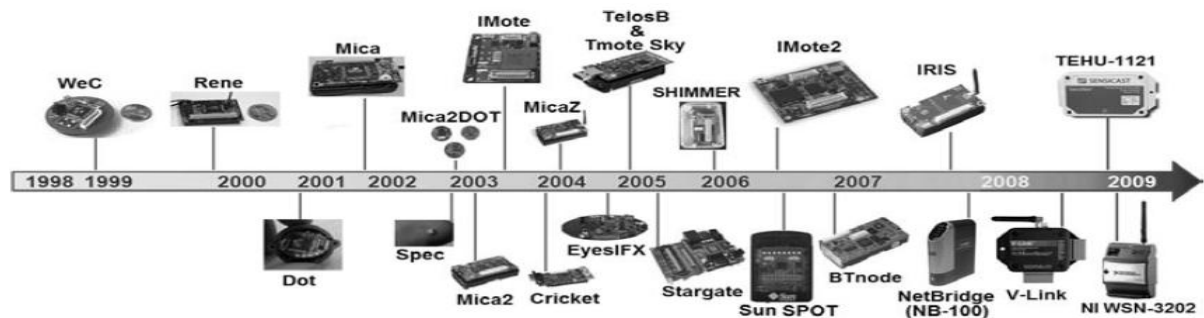
**Hình 3.2:** Bo mạch MicaZ của hãng Crossbow Technology [27].

Hình 3.2 minh họa các thành phần cơ bản của một nền tảng phần cứng điển hình, đó là bo mạch mẫu thử nghiệm MicaZ của hãng Crossbow Technology. Nó bao gồm bộ vi điều khiển, nguồn điện, bộ thu phát vô tuyến và các kết nối mở rộng để kết nối với các cảm biến. Nguồn năng lượng là một bộ pin gồm hai tế bào pin AA. Bộ thu phát vô tuyến được gắn ở phía bên trong của bo mạch. Hệ thống sử dụng một Anten bên ngoài gắn liền với bo mạch. Bo mạch không chứa bất kỳ một cảm biến nào. Các cảm biến có thể được gắn với bo mạch thông qua khối kết nối mở rộng. Vì vậy, chúng ta có thể sử dụng bo mạch như một hệ thống tạo mẫu cho nhiều ứng dụng khác nhau.

Phần cứng MicaZ là một bo mạch mẫu được sử dụng khi thử nghiệm với mạng cảm biến không dây. Đối với những sản phẩm ứng dụng cuối cùng thì các thành phần phần cứng của nút cảm biến thường được tích hợp trên cùng một sản phẩm có kích thước nhỏ gọn hơn.

### 3.1.2. Khảo sát một số nền tảng phần cứng cho mạng cảm biến không dây

Trong thời gian qua, đã có nhiều nền tảng phần cứng khác nhau được nghiên cứu và phát triển cho mạng cảm biến không dây. Các mốc thời gian ra đời của các nền tảng phần cứng này có thể được thấy rõ ở hình 3.3.



**Hình 3.3:** Mốc thời gian ra đời các nền tảng phần cứng [14].

Các nền tảng phần cứng có thể được chia thành hai loại cơ bản sau:

- **Các nền tảng phần cứng mức thấp:** Các nền tảng phần cứng này được đặc trưng bởi sự hạn chế về khả năng xử lý, bộ nhớ và khả năng truyền thông. Các nền tảng phần cứng này thường được triển khai với số lượng lớn để thực hiện nhiệm vụ cảm nhận cũng như tạo sự kết nối giữa các nút mạng.
- **Các nền tảng phần cứng mức cao:** Ngoài nhiệm vụ cảm nhận thì các nền tảng phần cứng này còn thực hiện nhiệm vụ tổng hợp, xử lý dữ liệu, kết nối mạng cảm biến không dây với các mạng khác bên ngoài. Các nền tảng phần cứng này đòi hỏi bộ vi xử lý có khả năng xử lý mạnh hơn, dung lượng bộ nhớ nhiều hơn.

Bảng 3.1, 3.2 tổng hợp các đặc điểm chính của một số nền tảng phần cứng.

**Bảng 3.1:** So sánh một số nền tảng phần cứng mức thấp.

Phần cứng	Bộ vi điều khiển	Bộ thu phát vô tuyến	Tần số trung tâm	Hệ điều hành
Mica2	ATMega 128L, 8bit, 128KB program memory, 4KB RAM	Chipcon CC1000	315/433/868/916 MHz	TinyOS, SOS, Mantis
Mica2 Dot	ATMega 128L, 8bit, 128KB program memory, 4KB RAM	Chipcon CC1000	315/433/868/916 MHz	TinyOS, SOS, Mantis
MicaZ	ATMega 128L, 8bit, 128KB program memory, 4KB RAM	Chipcon CC2420	2.4 GHz	TinyOS, SOS, Mantis, LiteOS, RETOS

<b>Phần cứng</b>	<b>Bộ vi điều khiển</b>	<b>Bộ thu phát vô tuyến</b>	<b>Tần số trung tâm</b>	<b>Hệ điều hành</b>
IRIS	ATmega 1281, 8bit, 128KB program memory, 8KB RAM	Atmel AT86RF230	2.4 GHz	TinyOS, LiteOS
Cricket	ATmega 128L, 8bit, 128KB program memory, 4KB RAM	Chipcon CC1000	433MHz	TinyOS
Eyes	MSP430F149, 16bit, 60KB program memory, 2KB RAM	RFM TR1001	868 MHz	TinyOS, PEEROS
EyesIFX v1	MSP430F149, 16bit, 60KB program memory, 2KB RAM	Infineon TDA5250	868 MHz	TinyOS
EyesIFX v2	MSP430F1611, 16bit, 48KB program memory, 10KB RAM	Infineon TDA5250	868 MHz	TinyOS
Btnode	ATmega 128L, 8bit, 128KB program memory, 4KB RAM	Chipcon CC1000 and Bluetooth radio	433 - 915 MHz, 2.4GHz	TinyOS
PowWow	MSP430F1612, 16bit, 55KB program memory, 5KB RAM	CC2420	2.4 GHz	Contiki, TinyOS, LiteOS
WiSMote	MSP430F5X, 16bit, 128/192/256KB program memory, 16KB RAM	CC2520	2.4 GHz	TinyOS
Epic Mote	MSP430F1611, 16bit, 48KB program memory, 10KB RAM	CC2420	2.4 GHz	TinyOS
CM5000	MSP430F1611, 16bit, 48KB program memory, 10KB RAM	CC2420	2.4 GHz	TinyOS, Contiki

**Bảng 3.2:** So sánh một số nền tảng phần cứng mức cao.

<b>Phần cứng</b>	<b>Bộ xử lý</b>	<b>Bộ nhớ</b>	<b>Các kết nối trên bo mạch</b>	<b>Hệ điều hành</b>
Stargate	Intel PXA255 Processor, 400MHz	64MB SDRAM, 32MB Flash	PCMCIA and compact flash connector, 51-pin expansion Connector for Mica2 Motes; Ethernet, RS232 Serial,	Embedded Linux

Phần cứng	Bộ xử lý	Bộ nhớ	Các kết nối trên bo mạch	Hệ điều hành
			JTAG, USB Connector via 51-pin Daughter Card Interface	
Netbridge NB-100	Intel IXP420 Xscale Processor, 266MHz	32MB RAM, 8MB Flash, 2GB USB Flash Disk	Mica2, MicaZ, IRIS, Telos Connector Ethernet, USB connector	Debian Linux
Imote	ARM7 processor, 12MHz	64KB SRAM, 512KB Flash	I2C, UART, USB, JTAG connector	TinyOS
Imote2	Marvell PXA271 Xscale Processor, 13 - 416MHz	256KB SRAM, 32MB SDRAM, 32MB Flash Memory	Integrated 802.15.4 radio, support for external radio through SDIO, and UART; USB client and host, 2xSPI, 3xUART, Camera, I2C, I2S, GPIO, AC97 connector	TinyOS, Linux, SOS

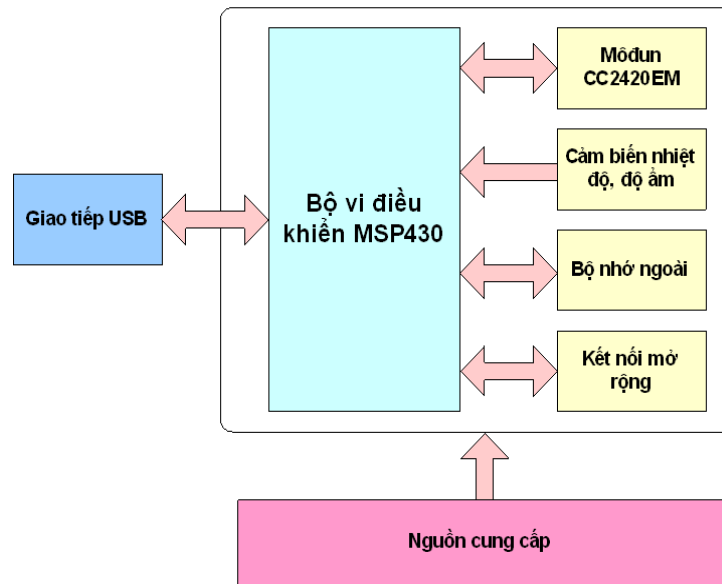
### 3.2. Đề xuất cấu trúc phần cứng TUmote

Dựa trên những nghiên cứu khảo sát về các cấu trúc phần cứng cho mạng cảm biến không dây, tác giả đã đề xuất một cấu trúc phần cứng riêng để phục vụ cho việc nghiên cứu và đánh giá thực nghiệm với giao thức EACTP.

Phần cứng này hoàn toàn có thể sử dụng lại được cho các nghiên cứu thực nghiệm khác trong lĩnh vực mạng cảm biến không dây và có thể rút ngắn thời gian để đưa các mô hình nghiên cứu lý thuyết sang các mô hình thực nghiệm có tính ứng dụng. Hình 3.4 là sơ đồ khối của cấu trúc phần cứng TUmote.

#### 3.2.1. Bộ vi điều khiển MSP430

Trong cấu trúc phần cứng TUmote, tác giả lựa chọn bộ vi điều khiển 16 bit MSP430 F1611. Đây là bộ vi điều khiển rất tiết kiệm năng lượng của hãng Texas Instruments. Bộ vi điều khiển MSP430 F1611 có 10KB bộ nhớ RAM, 48KB bộ nhớ Flash. Bộ vi điều khiển MSP430 F1611 được thiết kế cho những ứng dụng tiêu thụ năng lượng thấp và có thể hoạt động được ở nhiều chế độ tiết kiệm năng lượng khác nhau.



**Hình 3.4:** Sơ đồ khối cấu trúc phần cứng TUmote.

Trong cấu trúc phần cứng đề xuất, tác giả lựa chọn bộ vi điều khiển MSP430 F1611 bởi vì đây là dòng vi điều khiển rất tiết kiệm năng lượng và nhiều hệ điều hành cho mạng cảm biến không dây đã hỗ trợ dòng vi điều khiển này. Do vậy, TUmote hoàn toàn tương thích với các hệ điều hành phổ biến hiện nay như TinyOS, Contiki. Việc cài đặt hệ điều hành trên phần cứng TUmote sẽ cho phép tác giả lập trình phần mềm ứng dụng được thuận tiện và các tài nguyên phần cứng trên nút cảm biến cũng được quản lý một cách hiệu quả để tiết kiệm năng lượng.

### 3.2.2. Bộ thu phát vô tuyến CC2420EM

Trong cấu trúc phần cứng TUmote, tác giả sử dụng chip CC2420 của hãng Texas Instruments. Chip CC2420 là một chip thu phát vô tuyến công suất thấp theo chuẩn IEEE 802.15.4. Chip này hoạt động ở băng tần 2,4 GHz phù hợp với quy hoạch tần số ở Việt Nam. Tuy nhiên, việc chế tạo mạch in ở tần số 2,4 GHz là một thách thức đối với công nghệ chế tạo mạch in trong nước. Để giải quyết vấn đề này, tác giả lựa chọn module thu phát vô tuyến CC2424EM của hãng Texas Instruments. Trên module này đã được tích hợp sẵn chip CC2420.

### 3.2.3. Cảm biến SHT11

Tùy theo từng ứng dụng, phần cứng TUmote hoàn toàn có thể kết nối được với nhiều loại cảm biến khác ở bên ngoài thông qua khối kết nối mở rộng. Tuy nhiên, để phục vụ cho việc đo đạc thực nghiệm thì một cảm biến nhiệt độ và độ ẩm (SHT11) đã được tích hợp sẵn trên cấu trúc phần cứng TUmote.

Cảm biến SHT11 được sản xuất bởi hãng Sensirion AG. Cảm biến này cung cấp dữ liệu đầu ra số. Cảm biến được chế tạo theo quy trình CMOS và được kết hợp với bộ chuyển đổi A/D 14-bit. Đây là loại cảm biến tiêu thụ năng lượng thấp và có kích thước nhỏ gọn. Cảm biến này có thể được sử dụng cho nhiều ứng dụng giám sát môi trường.

#### **3.2.4. Bộ nhớ ngoài**

Bộ nhớ ngoài cho phép TUmote tăng khả năng lưu trữ ngoài bộ nhớ ROM và RAM trên vi điều khiển MSP430 F1611. TUmote sử dụng bộ nhớ ST M25P80. Bộ nhớ này chứa 1024kB cho dữ liệu và được chia thành 16 segments, mỗi segment có kích thước là 64kB.

#### **3.2.5. Nguồn cung cấp**

TUmote sử dụng hai pin AA có điện áp một chiều nằm trong dải từ 2,7V đến 3,6V. Nguồn nuôi cũng có thể lấy trực tiếp từ máy tính khi TUmote được kết nối với máy tính.

### **3.3. Các thông số kỹ thuật chính của phần cứng TUmote**

Hình 3.5 là hình ảnh minh họa cấu trúc phần cứng TUmote. Một vài thông số kỹ thuật chính của phần cứng TUmote được tóm tắt trong bảng 3.3.



**Hình 3.5:** Hình ảnh sản phẩm phần cứng TUmote.

Bảng 3.1 và bảng 3.3 cho chúng ta thấy có sự khác biệt về bộ vi điều khiển được sử dụng trong giải pháp thiết kế phần cứng TUmote với các bộ vi điều khiển được sử dụng trong các phần cứng MicaZ, Mica2 và IRIS. TUmote sử dụng bộ vi điều khiển MSP430 F1611 16 bit của hãng Texas Instruments. Đây là bộ vi điều khiển rất tiết kiệm năng lượng cả trong chế độ tích cực và chế độ ngủ. Đặc điểm này cho phép TUmote đạt được hiệu quả về năng lượng cũng như hiệu quả xử lý cao hơn so với các phần cứng khác chỉ sử dụng bộ vi điều khiển 8 bit. Ngoài ra, TUmote sử dụng bộ thu phát vô tuyến CC2420 của Texas Instruments. Đây là bộ thu phát vô tuyến hoạt động ở tần số 2,4 GHz phù hợp với băng tần không cấp phép ở Việt Nam. So với các bộ thu phát vô tuyến khác như CC1000 thì bộ thu phát vô tuyến CC2420 có tốc độ dữ liệu cao hơn lên đến 250 kbps.

**Bảng 3.3:** Các thông số kỹ thuật của TUmote.

Bộ vi điều khiển	MSP430 F1611, 8MHz, 16bit, 10KB RAM, 48KB Flash
Bộ thu phát vô tuyến	CC2420, 250kbps, 2.4GHz, tương thích với các thiết bị theo chuẩn IEEE 802.15.4
Phạm vi phủ sóng ở công suất phát 0 dBm	30m trong nhà, 100m ngoài trời
Cảm biến	SHT 11 (nhiệt độ, độ ẩm) Có khả năng mở rộng thêm các cảm biến khác
Giao diện lập trình	USB, JTAG
Hệ điều hành tương thích	Contiki, TinyOS

Các phần cứng PowWow, WiSMote, Epic Mote, CM5000 được thiết kế để phục vụ cho mục đích nghiên cứu và thử nghiệm với mạng cảm biến không dây. Khi so sánh với các phần cứng này, chúng ta thấy TUmote sử dụng bộ vi điều khiển và bộ thu phát vô tuyến có tính năng tương đương về khả năng xử lý, tính toán cũng như khả năng tiết kiệm năng lượng. Tuy nhiên, trong bản thiết kế phần cứng TUmote dành cho các nút cảm biến không cần kết nối với máy tính, khối giao tiếp máy tính đã được loại bỏ. Điều này giúp giảm giá thành sản phẩm, đồng thời cho phép TUmote có thể kết nối thêm được với nhiều cảm biến và các thiết bị ngoại vi khác. Do vậy, chúng ta có thể sử dụng phần cứng TUmote cho nhiều ứng dụng khác nhau trong thực tế. Hiện tại, phần cứng TUmote đang được ứng dụng trong sản phẩm Green Meter của đề tài khoa học công nghệ Cấp Bộ Công thương “*Nghiên cứu phát triển thiết bị đo và chấp hành thông minh nối mạng Smart Metter dựa trên công nghệ*”

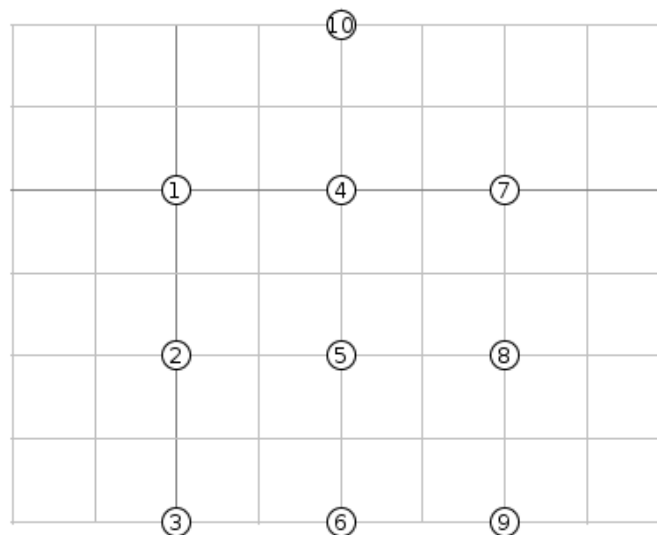


mạng cảm biến không dây phục vụ cho việc quản lý và tiết kiệm năng lượng điện” đang được thực hiện tại Viện nghiên cứu Điện tử, Tin học, Tự động hóa.

### 3.4. Triển khai mô hình đánh giá thực nghiệm với giao thức EACTP sử dụng phần cứng TUmote

Để đánh giá hiệu năng giao thức EACTP bằng phương pháp thực nghiệm, tác giả sử dụng hệ thống nghiên cứu thực nghiệm với mạng cảm biến không dây. Hệ thống này bao gồm phần cứng TUmote và công cụ phần mềm TU2C. Trong bài báo công bố số 3, tác giả có trình bày về hệ thống nghiên cứu thực nghiệm này.

Hình 3.6 là mô hình triển khai mạng gồm 10 nút. Các nút mạng định kỳ gửi bản tin dữ liệu về nút gốc là nút số 10. Các nút cảm biến có nhiệm vụ thu thập dữ liệu nhiệt độ và độ ẩm của môi trường và gửi về một nút gốc được kết nối với máy tính. Các tham số đánh giá được sử dụng giống như mô phỏng (mục 2.4.1).



**Hình 3.6:** Mô hình triển khai thực nghiệm một cụm gồm 10 nút.

#### 3.4.1. Mô hình đánh giá thực nghiệm 1

##### 3.4.1.1. Kịch bản đánh giá

Các nút cảm biến được bố trí cách đều nhau 1,5m trong tầm nhìn thẳng. Hình 3.7 minh họa sơ đồ bố trí các nút cảm biến TUmote.

Các tham số được sử dụng trong kịch bản đánh giá được tóm tắt ở bảng 3.4. Trong kịch bản đánh giá này, tác giả giảm công suất phát của các nút cảm biến xuống còn -15dBm. Các bước thực hiện đánh giá thực nghiệm được tác giả trình bày chi tiết ở Phụ lục 5 của luận án.



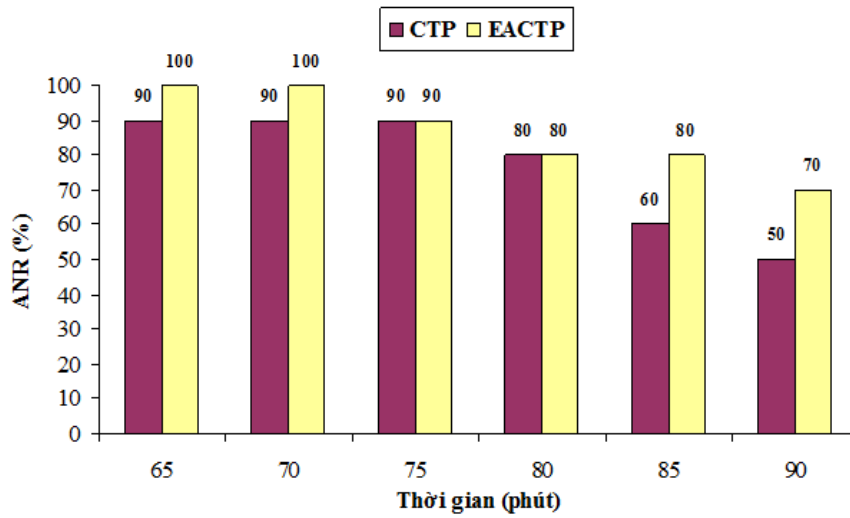
**Hình 3.7:** Sơ đồ bố trí các nút cảm biến TUmote.

**Bảng 3.4:** Kịch bản đánh giá thực nghiệm 1.

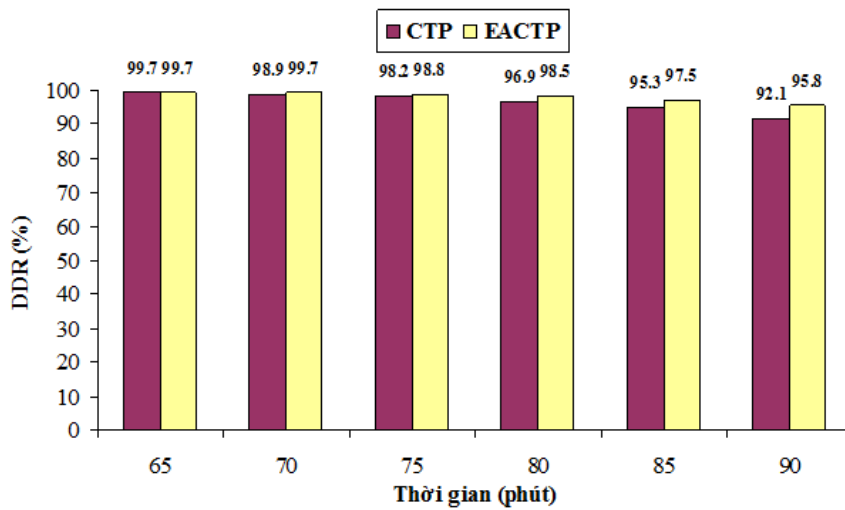
<b>Các tham số</b>	<b>Giá trị</b>
Môi trường truyền sóng	Trong tầm nhìn thẳng
Số nút mạng (nút)	10
Khoảng cách giữa các nút mạng (m x m)	1,5 x 1,5
Năng lượng ban đầu của mỗi nút	10J
Công suất phát (dBm)	-15
Chu kỳ gửi bản tin dữ liệu (giây)	60
Nguồn gửi bản tin dữ liệu	Tất cả các nút trong mạng (ngoài nút gốc)
Giao thức lớp MAC	CSMA/ContikiMAC

#### **3.4.1.2. Kết quả đánh giá**

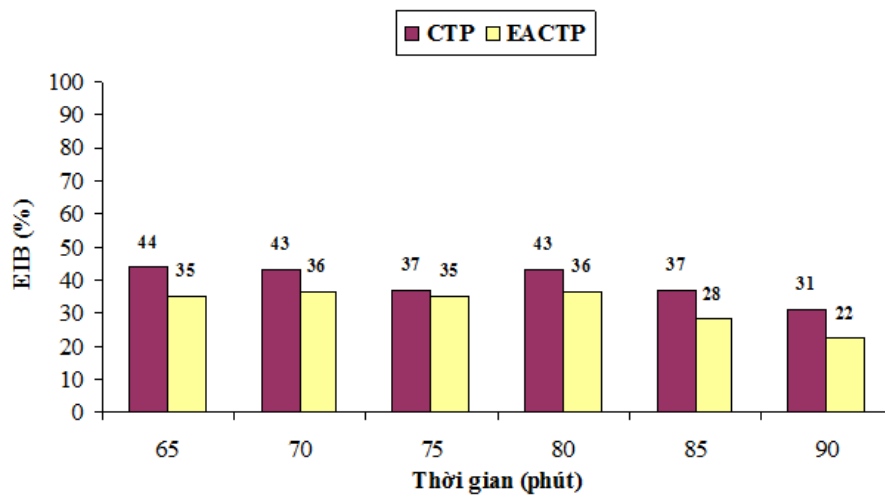
Hình 3.8, 3.9, 3.10 lần lượt là kết quả thực nghiệm đánh giá so sánh tỷ lệ các nút còn sống trong mạng, tỷ lệ chuyển phát bản tin dữ liệu và sự cân bằng năng lượng giữa các nút mạng đối với giao thức CTP và giao thức EACTP.



Hình 3.8: So sánh tỷ lệ các nút còn sống trong mạng.



Hình 3.9: So sánh tỷ lệ chuyển phát bản tin dữ liệu.



Hình 3.10: So sánh sự cân bằng năng lượng giữa các nút mạng.

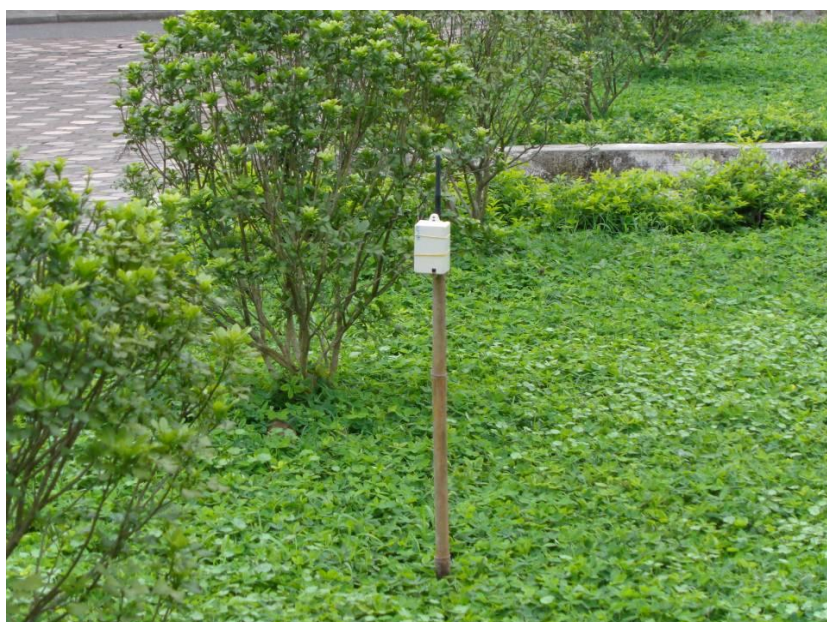
## 3.4.2. Mô hình đánh giá thực nghiệm 2

### 3.4.2.1. Kịch bản đánh giá

Các nút cảm biến được bố trí cách đều nhau 20m trong khuôn viên khu ký túc xá trường Đại học Công nghệ thông tin và truyền thông. Hình 3.11 minh họa khuôn viên khu vực tiến hành đo đạc thực nghiệm.



**Hình 3.11:** Khuôn viên khu vực tiến hành đo đạc thực nghiệm.



**Hình 3.12:** Triển khai các nút cảm biến tại khu vực thực nghiệm.

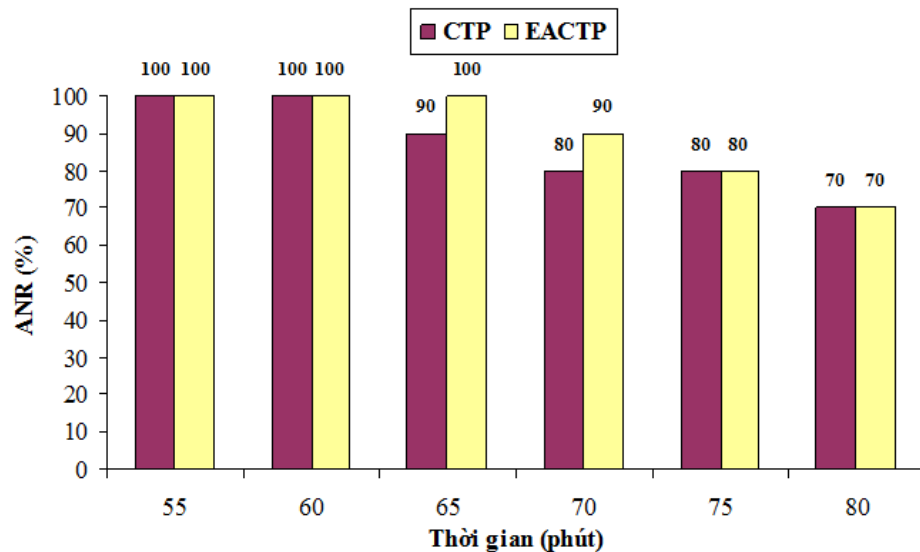
Các tham số được sử dụng trong kịch bản đánh giá thực nghiệm 2 được tóm tắt ở bảng 3.5. Trong kịch bản đánh giá này, tác giả cấu hình công suất phát của các nút cảm biến bằng 0 dBm. Các bước thực hiện đánh giá thực nghiệm được tác giả trình bày chi tiết ở Phụ lục 5 của luận án.

**Bảng 3.5:** Kịch bản đánh giá thực nghiệm 2.

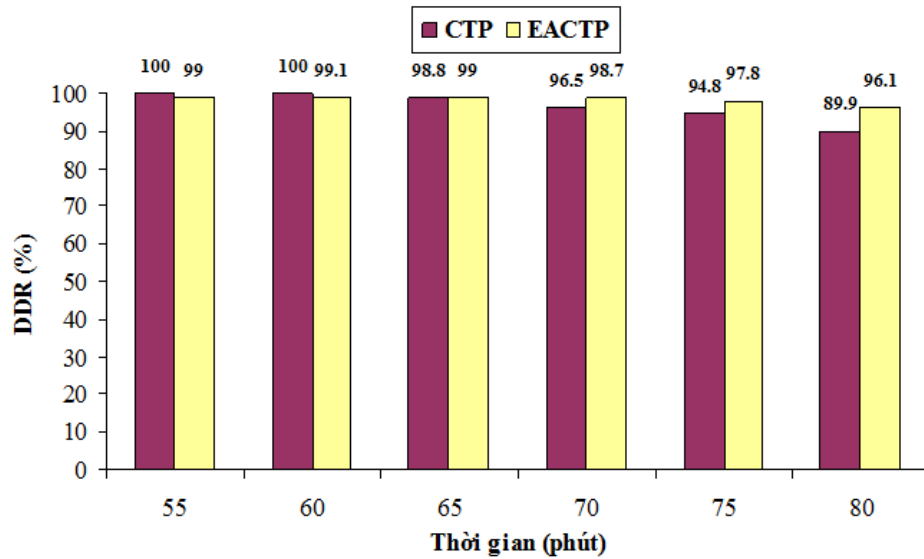
Các tham số	Giá trị
Môi trường truyền sóng	Không trong tầm nhìn thẳng, bị che chắn bởi cây cối, vật cản tầm thấp
Số nút mạng (nút)	10
Khoảng cách giữa các nút mạng (m x m)	20 x 20
Năng lượng ban đầu của mỗi nút	10J
Công suất phát (dBm)	0
Chu kỳ gửi bản tin dữ liệu (giây)	60
Nguồn gửi bản tin dữ liệu	Tất cả các nút trong mạng (ngoài nút gốc)
Giao thức lớp MAC	CSMA/ContikiMAC

### 3.4.2.2. Kết quả đánh giá

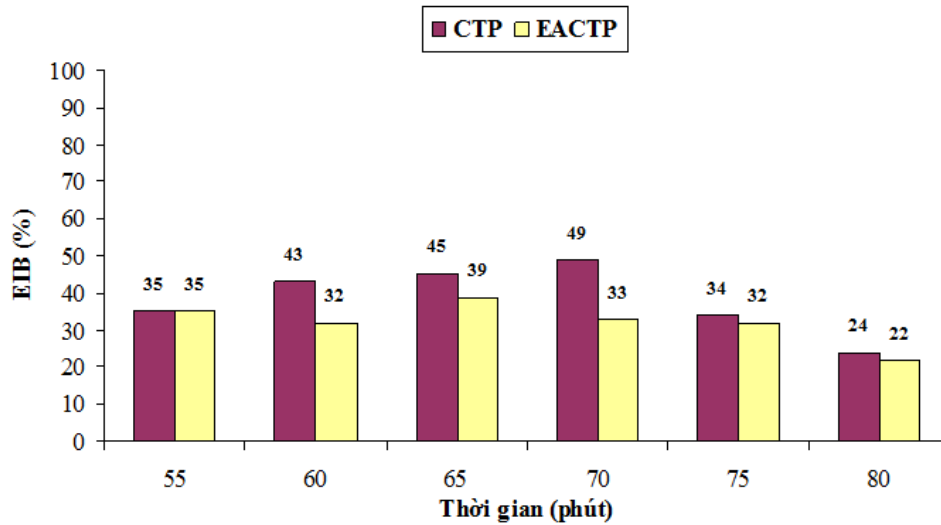
Hình 3.13, 3.14, 3.15 lần lượt là kết quả thực nghiệm đánh giá tỷ lệ các nút còn sống trong mạng, tỷ lệ chuyển phát bản tin dữ liệu và so sánh sự cân bằng năng lượng giữa các nút mạng đối với giao thức CTP và giao thức EACTP.



**Hình 3.13:** So sánh tỷ lệ các nút còn sống trong mạng.



**Hình 3.14:** So sánh tỷ lệ chuyển phát bản tin dữ liệu.



**Hình 3.15:** So sánh sự cân bằng năng lượng giữa các nút mạng.

### 3.4.3. Nhận xét

Kết quả đánh giá thực nghiệm cho thấy:

- Nếu giá trị ngưỡng được thiết lập để xác định thời gian sống của mạng là 100% thì hình 3.8, 3.13 cho thấy thời gian sống của mạng khi hoạt động theo giao thức EACTP được cải thiện hơn so với giao thức CTP: Tăng 10 phút (tương ứng với 15,4% ở hình 3.8) và tăng 5 phút (tương ứng với 7,7% ở hình 3.13). Như vậy, cả mô phỏng và thực nghiệm đều cho kết quả giống nhau đó là giao thức EACTP có thời gian sống của mạng tốt hơn so với giao thức CTP ban đầu.

- Hình 3.8, 3.9, 3.13, 3.14 cũng cho thấy khi số lượng các nút trong mạng hết năng lượng tăng lên thì số lượng bản tin dữ liệu được gửi về nút gốc giảm. Do vậy, tỷ lệ chuyển phát bản tin dữ liệu trong mạng cũng giảm theo. Qua đó, chúng ta nhận thấy rằng kết quả đánh giá bằng thực nghiệm hoàn toàn phù hợp với phân tích lý thuyết.
- Khi so sánh tỷ lệ chuyển phát bản tin dữ liệu ở hình 2.19, hình 3.9 và hình 3.14, chúng ta thấy kết quả đánh giá thực nghiệm khác so với kết quả đánh giá mô phỏng. Các kết quả mô phỏng cho thấy: Ở giai đoạn cuối, tỷ lệ chuyển phát bản tin dữ liệu (DDR) của giao thức EACTP thấp hơn so với giao thức CTP. Tuy nhiên, các kết quả đánh giá thực nghiệm lại trái ngược so với các kết quả đánh giá mô phỏng. Nguyên nhân của sự khác biệt này là do trong kịch bản mô phỏng, phạm vi truyền sóng của nút gốc đã bị thu hẹp lại sao cho chỉ có ba nút 1, 4, 7 là có thể truyền thông trực tiếp được với nút gốc. Khi các nút 1, 4, 7 hết năng lượng thì các nút còn lại trong mạng không còn tuyến đường nào khác để gửi dữ liệu về nút gốc. Trong khi đó, ở kịch bản đánh giá thực nghiệm, nút gốc có thể truyền thông trực tiếp với các nút khác trong mạng. Khi các nút 1, 4, 7 hết năng lượng thì các nút còn lại trong mạng vẫn có thể tìm được một tuyến đường khác để đến được nút gốc. Vì vậy, ở giai đoạn cuối của quá trình đánh giá thực nghiệm, lượng bản tin dữ liệu được gửi về nút gốc và tỷ lệ chuyển phát bản tin dữ liệu của giao thức EACTP cao hơn so với giao thức CTP ban đầu.
- Giao thức EACTP đảm bảo được sự cân bằng năng lượng giữa các nút mạng tốt hơn so với giao thức CTP. Điều này được thể hiện bởi chỉ số EIB của giao thức EACTP thấp hơn so với chỉ số EIB của giao thức CTP. Như vậy, các kết quả đánh giá thực nghiệm cũng tương đồng với các kết quả đánh giá mô phỏng.

### **3.5. Kết luận chương 3**

Trong chương 3, tác giả đã xây dựng một hệ thống triển khai thực nghiệm dựa trên 10 nút cảm biến phần cứng TUmote. Dựa trên hệ thống này, tác giả đã tiến hành triển khai và đánh giá hiệu năng giao thức EACTP mới đề xuất và giao thức CTP ban đầu. Các kết quả đánh giá thực nghiệm tương đồng với những phân tích lý thuyết và các kết quả đánh giá mô phỏng. Kết quả đánh giá thực nghiệm trên 10 nút cảm biến phần cứng TUmote đã kiểm chứng lại tính đúng đắn của những lập luận về tính hiệu

quả của giao thức truyền thông EACTP đó là: Thời gian sống của mạng khi hoạt động theo EACTP được cải thiện tốt hơn so với giao thức CTP ban đầu. Một phần kết quả của chương này đã được tác giả công bố trong các bài báo [3], [5], [6].



## KẾT LUẬN

Trong luận án này, tác giả tập trung giải quyết bài toán định tuyến có sự nhận thức về năng lượng áp dụng cho giao thức cây thu thập dữ liệu (bài toán định tuyến EACTP) trên mạng cảm biến không dây. Mục tiêu chính của luận án là: Đề xuất một giao thức định tuyến mới EACTP (Energy Aware Collection Tree Protocol) có sự nhận thức về năng lượng nhằm đảm bảo sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt và nâng cao thời gian sống của các nút mạng. Từ đó, thực thi và đánh giá hiệu năng của giao thức này bằng phương pháp mô phỏng, thực nghiệm.

Trên cơ sở phân tích mô hình toán học dựa trên lý thuyết đồ thị cho bài toán định tuyến cây thu thập dữ liệu có sự nhận thức về năng lượng EACTP, tác giả đã có một số đóng góp mới trong luận án như sau:

- Đề xuất một giao thức định tuyến mới đó là giao thức cây thu thập dữ liệu có sự nhận thức về năng lượng - EACTP (Energy Aware Collection Tree Protocol). Trong giao thức EACTP, tác giả đã thực hiện một số cải tiến: Thứ nhất, tác giả đã bổ sung thêm thành phần ước lượng năng lượng còn lại trên mỗi nút cảm biến; Thứ hai, tác giả đã đề xuất một thước đo định tuyến mới đó là trạng thái năng lượng còn lại *ES (Energy State)* để xác định tuyến đường tối ưu trong mạng; Thứ ba, tác giả đã đề xuất một thuật toán lựa chọn tuyến đường tối ưu mới dựa trên sự kết hợp giữa hai thước đo định tuyến là chất lượng liên kết của tuyến đường và trạng thái năng lượng còn lại trên nút chuyển tiếp.
- Mô phỏng và đánh giá hiệu năng giao thức EACTP thông qua một số mô hình mạng. Tác giả đã xác định các thước đo đánh giá phù hợp cho bài toán định tuyến EACTP và đưa ra một số kết quả đánh giá so sánh hiệu năng giữa giao thức EACTP và giao thức CTP. Các kết quả đánh giá so sánh dựa trên mô phỏng cho thấy giao thức EACTP đạt được một số tiêu chí quan trọng đó là: Tăng được thời gian sống của các nút mạng, đạt được tỷ lệ chuyển phát bản tin dữ liệu trong mạng ở mức cao, đảm bảo được sự cân bằng năng lượng giữa các nút mạng thuộc những tuyến đường có chất lượng liên kết tốt, không làm phát sinh thêm các chi phí mới về năng lượng trong việc gửi các bản tin điều khiển.

- Xây dựng một môi trường thực nghiệm dựa trên 10 nút cảm biến phần cứng TUmote (Thainguyen University mote), cho phép tùy biến, tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của các giao thức CTP, EACTP trong điều kiện thực tế. Môi trường thực nghiệm này hoàn toàn có thể sử dụng lại được cho các nghiên cứu thực nghiệm khác trong lĩnh vực mạng cảm biến không dây và có thể rút ngắn thời gian để đưa các mô hình nghiên cứu lý thuyết sang các mô hình thực nghiệm có tính ứng dụng. Kết quả đánh giá thực nghiệm trên 10 nút cảm biến phần cứng TUmote đã kiểm chứng lại tính đúng đắn của các kết quả mô phỏng cũng như những lập luận về tính hiệu quả của giao thức EACTP mới đề xuất đó là: Thời gian sống của mạng khi hoạt động theo EACTP được cải thiện tốt hơn so với giao thức CTP ban đầu.

**Hướng phát triển của đề tài:** Ứng dụng phần cứng TUmote và giao thức EACTP cho hệ thống quan trắc nhanh môi trường đang được nghiên cứu và triển khai tại Viện nghiên cứu Điện tử, Tin học, Tự động hóa. Hiện tại, phần cứng TUmote cũng đang được nghiên cứu phát triển cho ứng dụng giám sát năng lượng tiêu thụ thiết bị điện. Đây là một trong những ứng dụng thuộc đề tài cấp Bộ Công thương “*Nghiên cứu phát triển thiết bị đo và chấp hành thông minh nối mạng Smart Meter dựa trên công nghệ mạng cảm biến không dây phục vụ cho việc quản lý và tiết kiệm năng lượng điện*” đang được thực hiện tại Viện nghiên cứu Điện tử, Tin học, Tự động hóa.

## DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

1. Vũ Chiến Thắng, Nguyễn Chấn Hùng, Lê Nhật Thăng, “*EACTP: Giao thức cây thu thập dữ liệu cải tiến cho mạng cảm biến không dây,*” Chuyên san Công nghệ thông tin và Truyền thông, Tạp chí Khoa học và Kỹ thuật - Học viện Kỹ thuật quân sự, số 157, 10/2013, ISSN 1859-0209, trang 65-79.
2. Vũ Chiến Thắng, Nguyễn Chấn Hùng, Lê Nhật Thăng, “*Một nghiên cứu so sánh hiệu năng giao thức cây thu thập dữ liệu với các giao thức MAC khác nhau,*” Tạp chí Khoa học và Công nghệ, Đại học Đà Nẵng, số 5(66), 2013, trang 62-67.
3. Vũ Chiến Thắng, Nguyễn Chấn Hùng, Lê Nhật Thăng, “*Về một hệ thống nghiên cứu thực nghiệm cho mạng cảm biến không dây,*” Tạp chí Khoa học và Công nghệ, Đại học Đà Nẵng, số 3(64), 2013, ISSN 1859-1531, trang 103-109.
4. Thang Vu Chien, Hung Nguyen Chan, Thanh Nguyen Huu, “*Operating System for Wireless Sensor Networks and an Experiment of Porting ContikiOS to MSP430 Microcontroller,*” Journal of Computer Science and Information, Vol 5, Issue 1, February 2012, ISSN: 2088-7051, pp. 50-56.
5. Thang Vu Chien, Hung Nguyen Chan, Thanh Nguyen Huu, “*A Comparative Study on Hardware Platforms for Wireless Sensor Networks,*” International Journal on Advanced Science Engineering Information Technology, 2012, ISSN: 2088-5334, Vol 2, No. 1, pp. 70-74.
6. Vũ Chiến Thắng, Nguyễn Chấn Hùng, Lê Nhật Thăng, “*EACTP: Giao thức cây thu thập dữ liệu với thông lượng cao và đảm bảo sự cân bằng năng lượng,*” Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông, Tạp chí Công nghệ thông tin & Truyền thông, Chuyên san Kỳ 3, Tập V-1, 2014, ISSN: 1859-3526, trang 41-50.

## TÀI LIỆU THAM KHẢO

- [1]. De Couto D, Aguayo D, Bicket J, Morris R , “A high-throughput path metric for multi-hop wireless routing”. In: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, New York, 2003.
- [2]. Joakim Flathagen, Erlend Larsen, Paal E. Engelstad, “O-CTP: Hybrid Opportunistic Collection Tree Protocol for Wireless Sensor Networks,” in Proceedings of 7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications, SensrApp 2012, Clearwater, Florida.
- [3]. Ugo Colesanti, Silvia Santini, “The Collection Tree Protocol for the Castalia Wireless Sensor Networks Simulator,” Technical Report Nr. 729, Department of Computer Science, ETH Zurich, June, 2011.
- [4]. Dixit Sharma, Ramiro Liscano, and Shahram Shah Heydari, “Collector Tree Protocol Performance in Mobile Wireless Sensor Networks,” IEEE CCECE 2011.
- [5]. Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis, “Collection Tree Protocol,” In Proceedings of the 7th ACM Conference on Embedded Net-worked Sensor Systems (SenSys 2009), Berkeley, CA, USA, November 2009.
- [6]. Jiliang Wang, Yunhao Liu, MoLi, Wei Dong, Yuan He, “QoF: Towards Comprehensive Path Quality Measurement in Wireless Sensor Networks,” INFOCOM, 2011 Proceedings IEEE, Shanghai, pp. 775-783.
- [7]. Jing Zhao, Lei Wang, Wenlong Yue, Zhengquan Qin, Ming Zhu, “Load Migrating for the Hot Spots in Wireless Sensor Networks using CTP,” 2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks.
- [8]. Yongjun Li, Hu Chen, Rongchuan He, Rong Xie, Shaocong Zou, “ICTP: An Improved Data Collection Protocol Based On CTP,” in Proceedings of the International Conference on Wireless Communications and Signal, 2010, Suzhou, pp. 1-5.
- [9]. Rahul C. Shah , Jan M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” in proceedings of IEEE Wireless Communications and Networking Conference (WCNC’02), Orlando, FL, USA, March 2002, pp. 350-355.

- [10]. K. Daabaj, M. Dixon, T. Koziniec, "Avoiding Routing Holes in Wireless Sensor Networks", in Proceedings of the World Congress of Engineering and Computer Science (WCECS'09), San Francisco, CA, USA, October 2009, pp. 356-361.
- [11]. K. Daabaj, M. Dixon, T. Koziniec, "LBR: Load Balancing Routing for Wireless Sensor Networks," IAENG Transactions on Engineering Technologies Vol. 4, American Institute of Physics (AIP) Conference Proceedings, 1247, June 2010, pp. 71-85.
- [12]. Ramesh Rajagopalan and Pramod K.Varshney, "Data Aggregation Techniques in Sensor Networks: A Survey," IEEE Communications Surveys & Tutorials, 4th Quarter 2006.
- [13]. Fredrik Österlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt, "Cross-level sensor network simulation with cooja," In Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006), Tampa, Florida, USA, November 2006.
- [14]. Ian F. Akyildiz, Mehmet Can Vuran, "Wireless Sensor Networks", Wiley, 2010.
- [15]. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in MobiCom: Proceedings of the 6th annual international conference on Mobile computing and networking, New York, NY, USA, 2000, ACM, pp. 56–67.
- [16]. A. Chandrakasan, and H. Balakrishnan W. Heinzelman, "Energy efficient Communication Protocol for Wireless Micro Sensor Networks," in 33rd Annual Hawaii International Conf. on System Sciences, Washington, 2000.
- [17]. S. Lindsey and C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," in Aerospace Conference Proceedings, 2002, vol. 3, 2002, pp. 3-1125 - 3-1130.
- [18]. Dharma P. Agrawal Arati Manjeshwar, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," in 15th International Parallel and Distributed Processing Symposium (IPDPS'01) Workshops, vol. 3, 2001, p. pp.30189a.
- [19]. V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," IEEE J. Select. Areas Commun., 17 (1999), pp. 1333 - 1344.

- [20]. Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for Ad Hoc routing,” in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA, 2001, ACM, pp. 70–84.
- [21]. Paolo Santi, “Topology Control in Wireless Ad Hoc and Sensor Networks,” John Wiley & Sons Inc., ISBN-10 0-470-09453-2, 2006.
- [22]. Azzedine Boukerche, “Algorithms and Protocols for Wireless Sensor Networks,” John Wiley & Sons Inc., ISBN: 9780470396360, 2008.
- [23]. M. H. Alizai, O. Landsiedel, J. A. B. Link, S. Gotz, and K. Wehrle, “Bursty traffic over bursty links,” in *SenSys*, 2009.
- [24]. Trương Mỹ Dung, “Lý thuyết đồ thị,” Trường Đại học Khoa học Tự nhiên, Đại học Quốc Gia Thành phố Hồ Chí Minh, 2005.
- [25]. Nguyễn Đức Nghĩa, Nguyễn Tô Thành, “Toán rời rạc,” NXB Đại học Quốc Gia Hà Nội, 2003.
- [26]. C. Park, K. Lahiri, and A. Raghunathan, “Battery discharge characteristics of wireless sensor nodes: An experimental analysis,” In *Proceedings of the IEEE Conf. on Sensor and Ad-hoc Communications and Networks (SECON)*, Santa Clara, California, USA, September 2005.
- [27]. Crossbow technology. [Online]. Available: <http://xbow.com>
- [28]. X. Jiang, P. Dutta, D. Culler, and I. Stoica, “Micropower meter for energy monitoring of wireless sensor networks at scale,” In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 186–195, Cambridge, Massachusetts, USA, 2007.
- [29]. Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, Zhitao He, “Software-based Online Energy Estimation for Sensor Nodes,” *Proceedings of the 4th workshop on Embedded networked sensors*, 2007.
- [30]. <http://ti.com>, [Online].
- [31]. <http://www.sensirion.com>, [Online].
- [32]. JP. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, “Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks,” RFC 6551, March 2012.

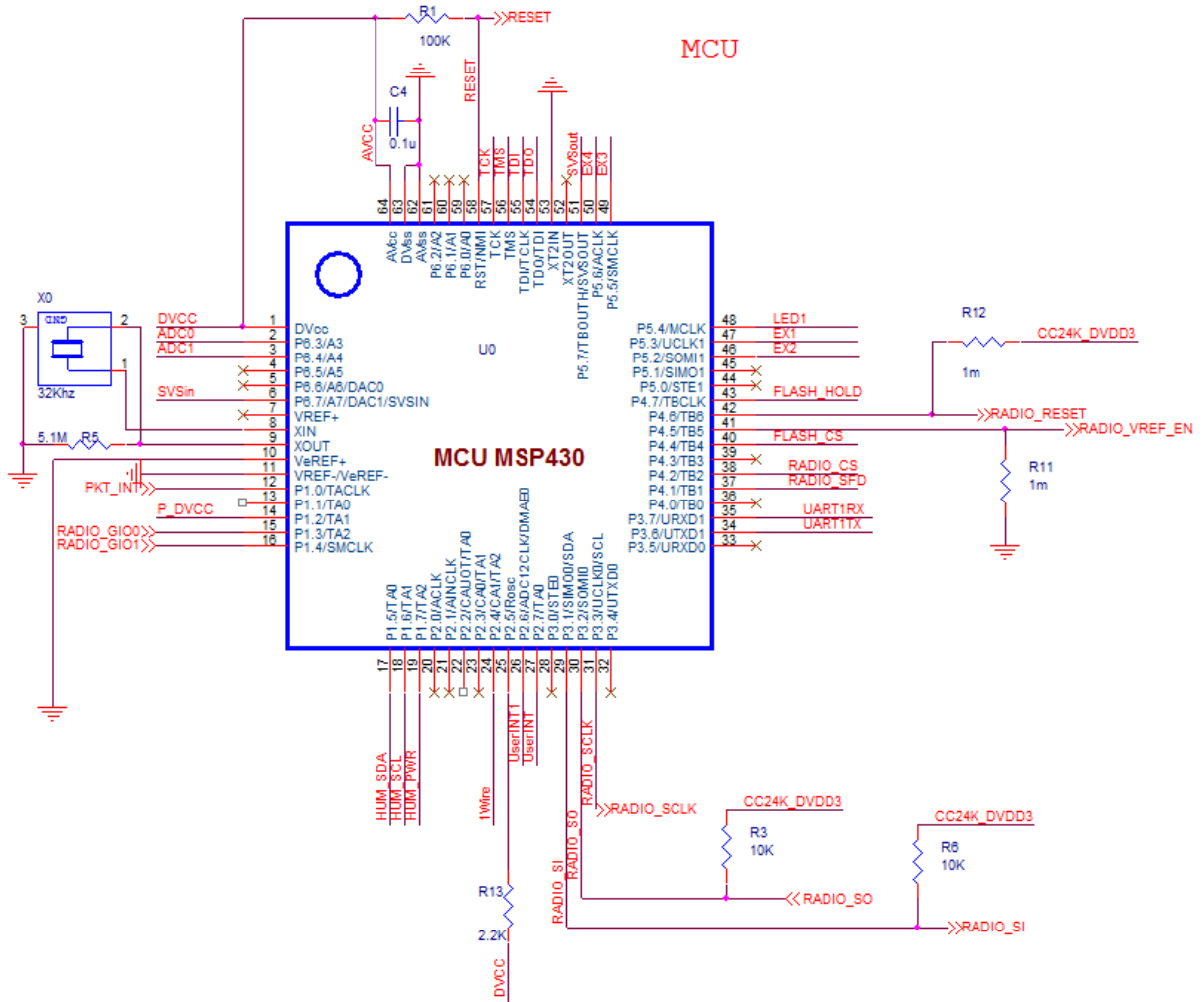
- [33]. Roberto Verdone, Davide Dardari, Gianluca Mazzini, Andrea Conti, “Wireless Sensor and Actuator Networks: Technologies, Analysis and Design,” Academic Press, ISBN-10: 0123725399, 2008.
- [34]. S. Singh, M. Woo, and C. Raghavendra, “Power-aware routing in mobile ad hoc networks,” In The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 181-190, 1998.
- [35]. J-P. Sheu, C-T. Hu, and C-M. Chao, “The Handbook of Ad Hoc Wireless Networks,” Chapter Energy-Conserving Grid Routing Protocol in Mobile Ad Hoc Networks. RCR Press LLC, 2003.
- [36]. Rahul C. Shah , Jan M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” in proceedings of IEEE Wireless Communications and Networking Conference (WCNC’02), Orlando, FL, USA, March 2002, pp. 350-355.
- [37]. Marc Lee and Vincent W.S. Wong, “An Energy-Aware Spanning Tree Algorithm for Data Aggregation in Wireless Sensor Networks,” In Proceedings of Communications, Computers and signal Processing, pp. 300-303, 2005.
- [38]. Xiaoyan Hong, Mario Gerla and Hanbiao Wang, “Load Balanced, Energy-Aware Communications for Mars Sensor Networks,” In Proceedings of Aerospace Conference, Vol 3, pp.3-1109 – 3-1115, 2002.
- [39]. Adam Dunkels, Joakim Eriksson, Nicolas Finne, Nicolas Tsiftes, “Powertrace: Network-level Power Profiling for Low-power Wireless Networks,” SICS Report, 2011.
- [40]. H. Cha, S. Choi, I. Jung, H. Kim, and H. Shin, “RETOS: Resilient, expandable, and threaded operating system for wireless sensor networks,” in Proc. ACM/IEEE IPSN, 2007.
- [41]. Q. Cao, T. F. Adbelzaher, and J. A. Stankovic, “The LiteOS operating system: towards Unix-like abstractions for wireless sensor networks,” in Proc. ACM/IEEE IPSN, 2008.
- [42]. TinyOS. [Online]. Available: <http://tinycos.net/>
- [43]. The EyesIFX platform. [Online]. Available: <http://www.tinycos.net/ttx-02-2005/platforms/ttx2005-eyesIFX.ppt>
- [44]. National Instruments. [Online]. Available: <http://www.ni.com/wsn/>

- [45]. Imote2. [Online]. Available:  
[http://wsn.cse.wustl.edu/images/e/e3/Imote2\\_Datasheet.pdf](http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf)
- [46]. Stargate. [Online]. Available: [http://www.willow.co.uk/Stargate\\_Datasheet.pdf](http://www.willow.co.uk/Stargate_Datasheet.pdf)
- [47]. BTnode. [Online]. Available:  
<http://www.btnode.ethz.ch/Documentation/BTnodeRev3HardwareReference>
- [48]. Cricket. [Online]. Available: <http://cricket.csail.mit.edu/>
- [49]. Georgios Parissidis, Merkourios Karaliopoulos, Rainer Baumann, Thrasyvoulos Spyropoulos, "*Guide to Wireless Mesh Networks*," Computer Communications and Networks, Springer-Verlag London Limited 2009, Online ISBN: 978-1-84800-909-7, pp 199-230.
- [50]. Sivagami , K. Pavai, D. Sridharan and S.A.V. Satya Murty, "*Energy and Link Quality Based Routing for Data Gathering Tree in Wireless Sensor Networks Under TinyOS 2x*," International Journal of Wireless & Mobile Networks, Vol.2, No.2, May 2010.
- [51]. Dhaval Patel, Bijal Chawla, Chandresh Parekh, "*Energy Aware and Link Quality Based Routing in Wireless Sensor Networks under TinyOS-2.x*," International Journal of Modern Engineering Research, Vol.3, Issue.3, 2013.
- [52]. Li Ya; Wang Pengjun; Luo Rong; Yang Huazhong; Liu Wei, "*Reliable energy-aware routing protocol for heterogeneous WSN based on beaconing*," in Proc. Advanced Communication Technology (ICACT), February 2014.
- [53]. Epic Mote. [Online]. Available:  
<http://www.eecs.berkeley.edu/~prabal/projects/epic/>
- [54]. PowWow. [Online]. Available: <http://powwow.gforge.inria.fr/>
- [55]. CM5000. [Online]. Available: <http://www.advanticsys.com/>
- [56]. WiSMote. [Online]. Available:  
<http://www.aragosystems.com/en/wisnet-item/wisnet-wismote-item.html>

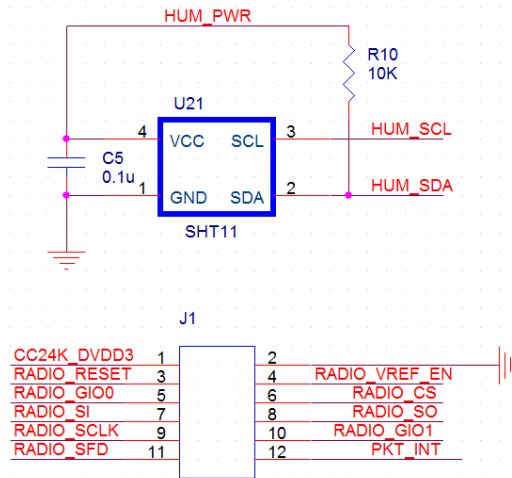


# PHỤ LỤC

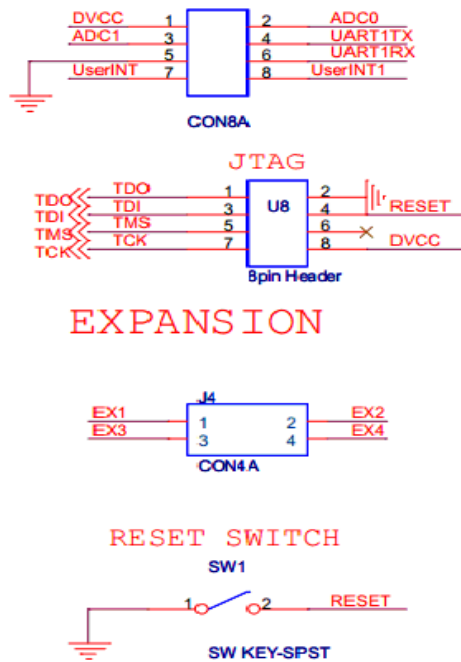
## Phụ lục 1. Bản thiết kế sơ đồ nguyên lý và sơ đồ mạch in phần cứng TUmate



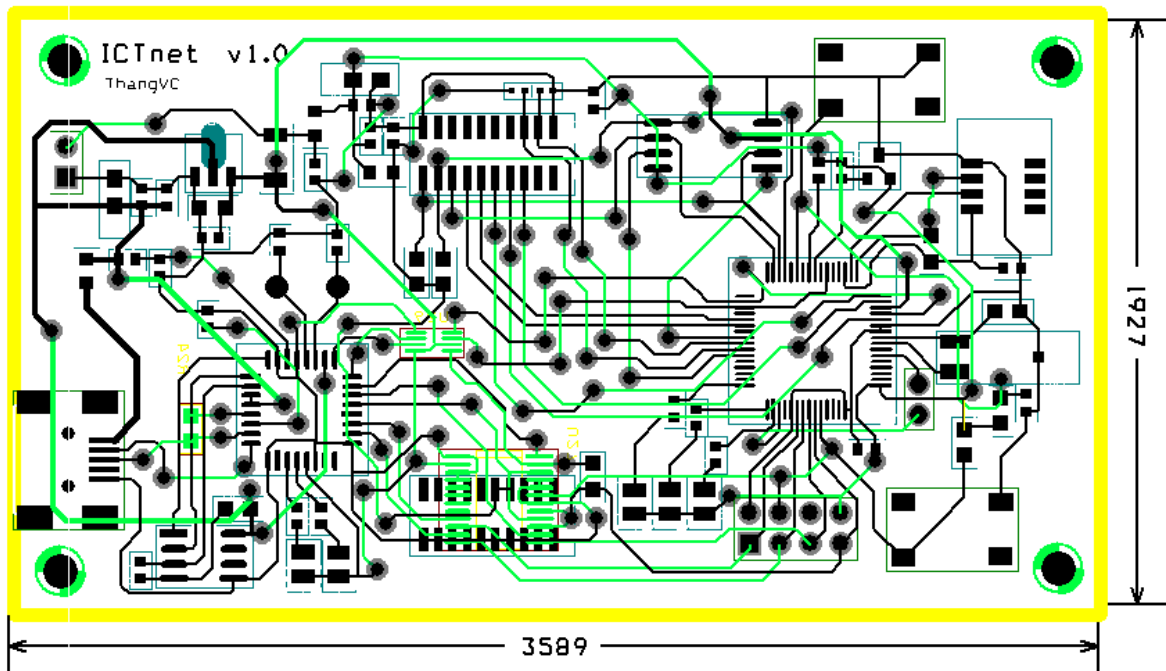
Hình 1: Sơ đồ nguyên lý khối điều khiển của sản phẩm phần cứng TUmate.



**Hình 2:** Sơ đồ nguyên lý khối cảm biến và khối giao tiếp với mô đun vô tuyến của sản phẩm phần cứng TUmote.



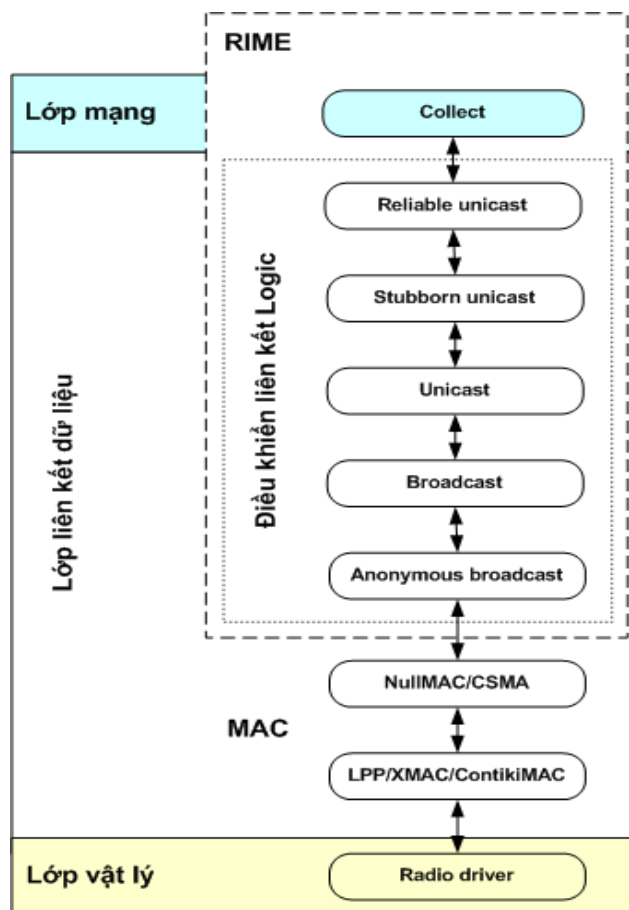
**Hình 3:** Sơ đồ nguyên lý khối giao tiếp người dùng, khối kết nối mở rộng của sản phẩm phần cứng TUmote.



**Hình 4:** Sơ đồ mạch in sản phẩm phần cứng TUmote.

## Phụ lục 2. Ngăn xếp truyền thông RIME trong hệ điều hành Contiki

Giao thức EACTP được xây dựng trên ngăn xếp truyền thông RIME trong hệ điều hành Contiki.



**Hình 5:** Giao thức CTP được xây dựng trên ngăn xếp truyền thông RIME trong Contiki.

RIME là một ngăn xếp truyền thông trong hệ điều hành Contiki được thiết kế cho các mạng không dây công suất thấp. RIME cung cấp nhiều cơ chế truyền thông cơ bản như phát quảng bá cục bộ, truyền thông unicast đơn chặng, truyền thông unicast đa chặng. Ngăn xếp truyền thông RIME cũng được thiết kế để phù hợp với nhiều giao thức phổ biến trong mạng cảm biến không dây như giao thức định tuyến mạng lưới (mesh), giao thức cây thu thập dữ liệu. Các mô đun truyền thông trong RIME được sử dụng trong giao thức CTP bao gồm:

- **Mô đun runicast (reliable unicast):** Là mô đun truyền thông unicast tin cậy giữa 2 nút. Runicast thực hiện việc truyền lại và sử dụng các bản tin xác nhận ACK để đảm bảo rằng nút lân cận nhận thành công một bản tin. Mô đun runicast sử dụng mô đun stubborn unicast để thực hiện việc truyền lại. Do vậy

mô đun runicast không phải quản lý chi tiết việc thiết lập các bộ định thời và thực hiện việc truyền lại mà nó chỉ tập trung vào việc giải quyết vấn đề xác nhận bản tin.

Mô đun runicast thêm vào bản tin hai thuộc tính là: Kiểu bản tin đơn chặng và nhận dạng (ID) bản tin đơn chặng. Mô đun runicast sử dụng thuộc tính nhận dạng bản tin như là số thứ tự để kết hợp các bản tin xác nhận với các bản tin dữ liệu tương ứng.

Ứng dụng hoặc giao thức sử dụng mô đun runicast có thể xác định số lần truyền tối đa mà mô đun runicast sẽ thử trước khi thông báo bản tin bị quá hạn (timeout).

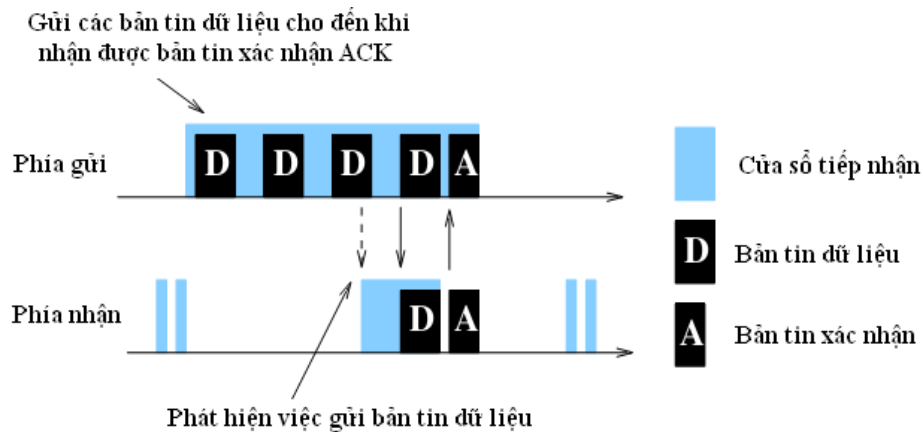
- **Mô đun stunicast (stubborn unicast):** Mô đun stunicast gửi lặp lại các bản tin đến một nút lân cận sử dụng mô đun unicast. Mô đun stunicast thực hiện việc gửi và gửi lại các bản tin cho đến khi lớp trên hoặc giao thức lớp trên hủy bỏ việc truyền dẫn.

Trước khi mô đun stunicast gửi một bản tin thì nó khởi tạo một bộ đệm hàng đợi để sao chép các thuộc tính bản tin và dữ liệu lớp ứng dụng đồng thời bộ định thời được thiết lập. Khi bộ định thời hết hạn thì mô đun stunicast sao chép bộ đệm hàng đợi vào bộ đệm ngăn xếp truyền thông RIME để gửi bản tin đi sử dụng mô đun unicast. Mô đun stunicast thiết lập số lần truyền lại cho bản tin như là một thuộc tính của các bản tin gửi đi.

- **Mô đun unicast:** Mô đun unicast gửi một bản tin đơn chặng đến một nút lân cận cho trước. Mô đun unicast sử dụng mô đun broadcast và thêm vào thuộc tính địa chỉ của nút nhận cho các bản tin gửi đi. Với các bản tin đến, mô đun unicast kiểm tra thuộc tính địa chỉ nút nhận và loại bỏ bản tin đến nếu địa chỉ nút nhận không khớp với địa chỉ của nó.
- **Mô đun broadcast:** Mô đun này gửi các bản tin đến tất cả các nút lân cận trong phạm vi phủ sóng. Mô đun này thêm địa chỉ nút gửi như là một thuộc tính của các bản tin gửi đi. Tất cả các mô đun trong RIME cần xác định nút gửi trong các bản tin được gửi đi thông qua mô đun broadcast hoặc là trực tiếp hoặc là gián tiếp thông qua bất kỳ một mô đun truyền thông nào khác dựa trên mô đun broadcast.
- **Mô đun anonymous broadcast (abc):** Mô đun này gửi các bản tin đến tất cả các nút lân cận. Mô đun abc thêm tiêu đề cho các bản tin được gửi đi.

### Phụ lục 3. Giao thức ContikiMAC trong hệ điều hành Contiki

Một trong những mục tiêu cần quan tâm khi thiết kế giao thức lớp MAC cho mạng cảm biến không dây là chia sẻ hiệu quả tài nguyên truyền thông giữa các nút cảm biến. Các tài nguyên này bao gồm thời gian, năng lượng và tần số. Với bất kỳ một giao thức lớp MAC nào thì vấn đề hiệu quả năng lượng là vô cùng quan trọng. Giao thức lớp MAC cần phải hỗ trợ các chế độ hoạt động tiết kiệm năng lượng cho nút cảm biến. Việc bảo toàn năng lượng hiệu quả nhất là tắt bộ vô tuyến khi không cần thiết. Hiện tại, đã có một số giao thức lớp MAC như giao thức LPL, XMAC và ContikiMAC đã được thực thi trong hệ điều hành Contiki. Trong bài báo công bố số 2, tác giả trình bày một số kết quả mô phỏng đánh giá hiệu năng của giao thức CTP khi sử dụng các giao thức lớp MAC khác nhau. Hiệu năng của mạng được đánh giá qua hai tham số chính đó là: Tỷ lệ chuyển phát bản tin dữ liệu và công suất tiêu thụ trung bình trong toàn mạng. Các kết quả mô phỏng cho thấy, giao thức CTP đạt được hiệu năng cao nhất khi sử dụng kết hợp giữa giao thức CSMA và giao thức ContikiMAC ở lớp MAC.



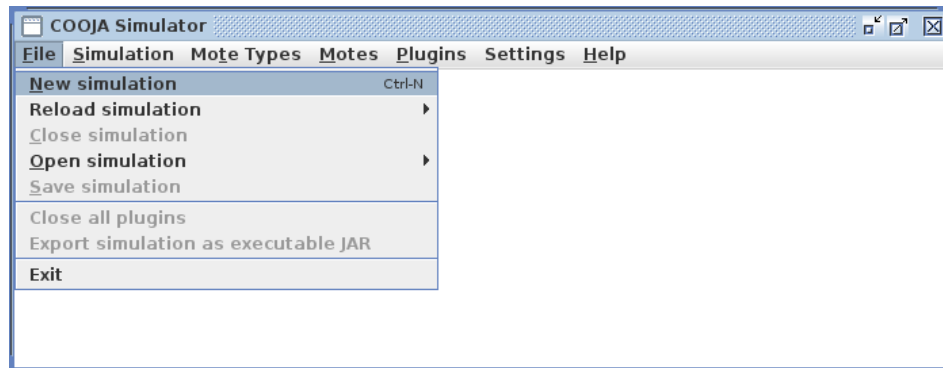
**Hình 6:** Cơ chế hoạt động của giao thức ContikiMAC.

Cơ chế hoạt động của giao thức ContikiMAC như sau: Để gửi một bản tin dữ liệu, phía gửi sẽ gửi lặp đi lặp lại cùng một bản tin cho đến khi nhận được một bản tin xác nhận. Các nút trong mạng tắt bộ vô tuyến trong phần lớn thời gian và định kỳ bật bộ vô tuyến để kiểm tra kênh truyền. Nếu phát hiện một bản tin dữ liệu đang được truyền trên kênh truyền thì phía thu sẽ luôn bật bộ vô tuyến để nhận bản tin dữ liệu. Nút nhận sẽ kiểm tra bản tin dữ liệu, nếu bản tin dữ liệu gửi cho nó thì nó sẽ xác nhận lại phía gửi bằng một bản tin xác nhận ACK. Như vậy, giao thức ContikiMAC được thiết kế đơn giản, dễ thực thi và không cần phải sử dụng các bản tin báo hiệu gửi cũng như các tiêu đề bổ sung.

#### Phụ lục 4. Đánh giá mô phỏng với Cooja.

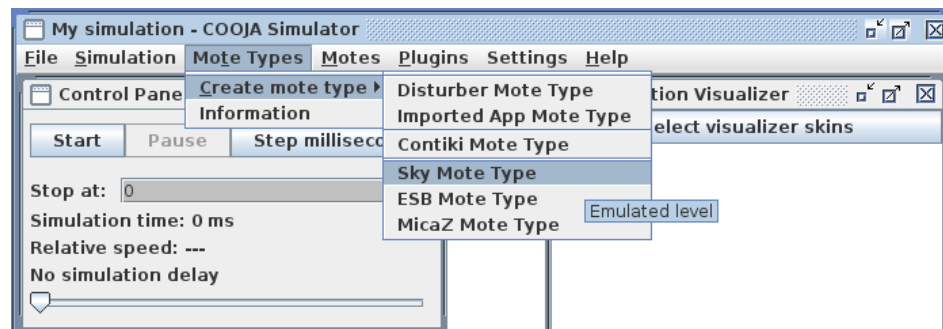
**Bước 1:** Chuẩn bị mã nguồn chương trình cho giao thức EACTP (EACTP.c), giao thức CTP (CTP.c).

**Bước 2:** Chạy công cụ mô phỏng Cooja.

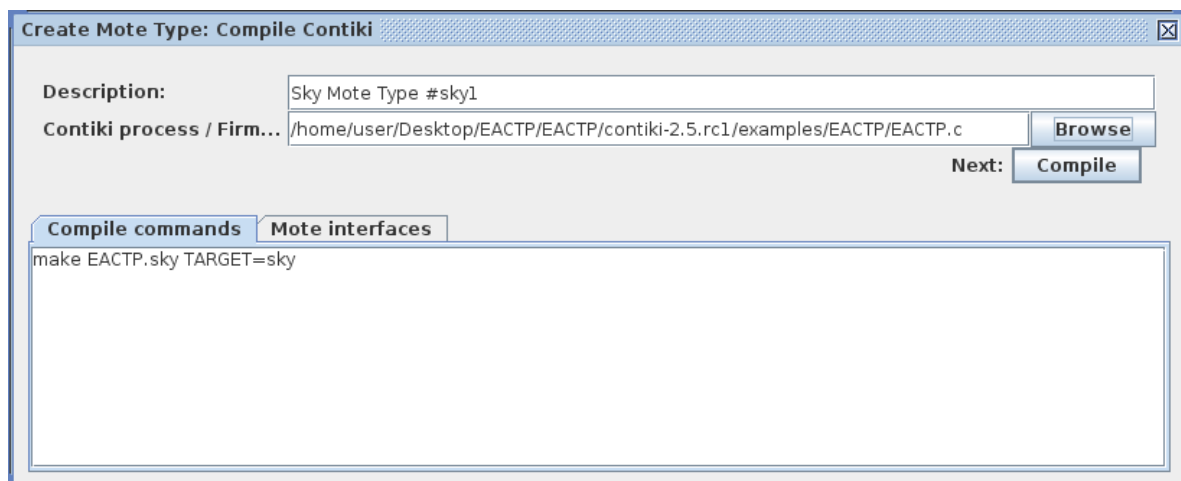


**Hình 7:** Tạo mô phỏng mới với Cooja.

**Bước 3:** Sử dụng nút cảm biến giả lập có sẵn trong Cooja và nạp mã nguồn chương trình EACTP cho nút cảm biến giả lập (trong trường hợp mô phỏng với giao thức CTP thì chọn đường dẫn đến mã nguồn CTP.c).

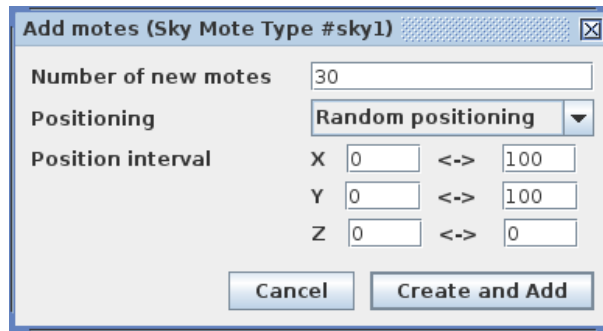


**Hình 8:** Sử dụng nút cảm biến giả lập trong Cooja.



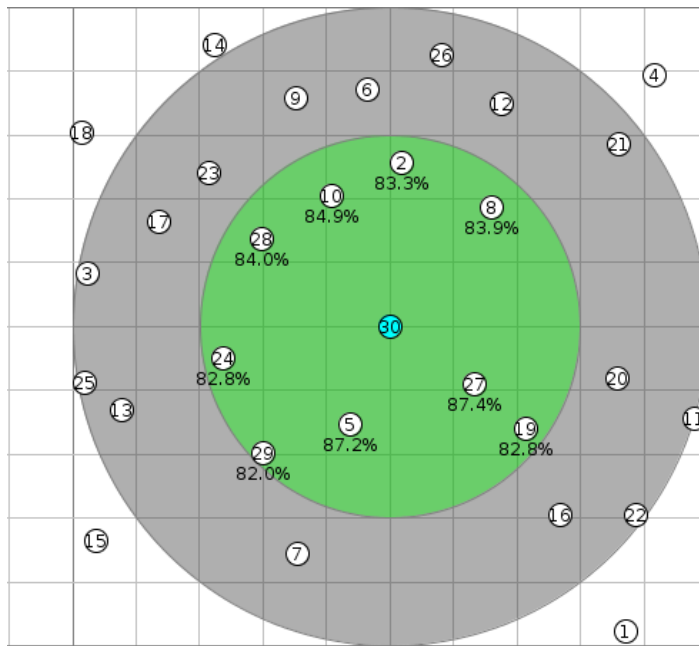
**Hình 9:** Nạp mã nguồn chương trình cho nút cảm biến giả lập trong Cooja.

**Bước 4:** Tạo 30 nút cảm biến giả lập vào chương trình mô phỏng.



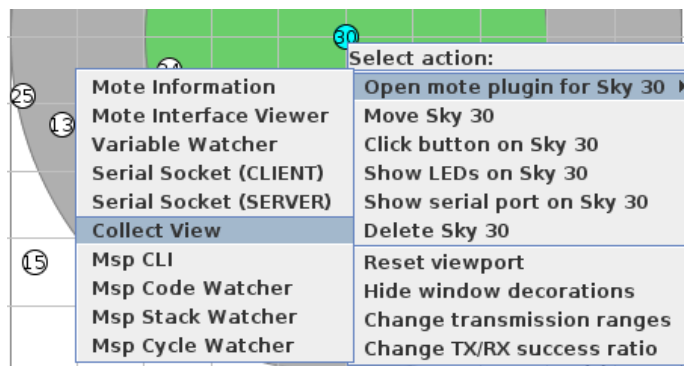
**Hình 10:** Cửa sổ giao diện tạo các nút cảm biến giả lập.

**Bước 5:** Di chuyển nút Sink (nút số 30) về vị trí trung tâm.



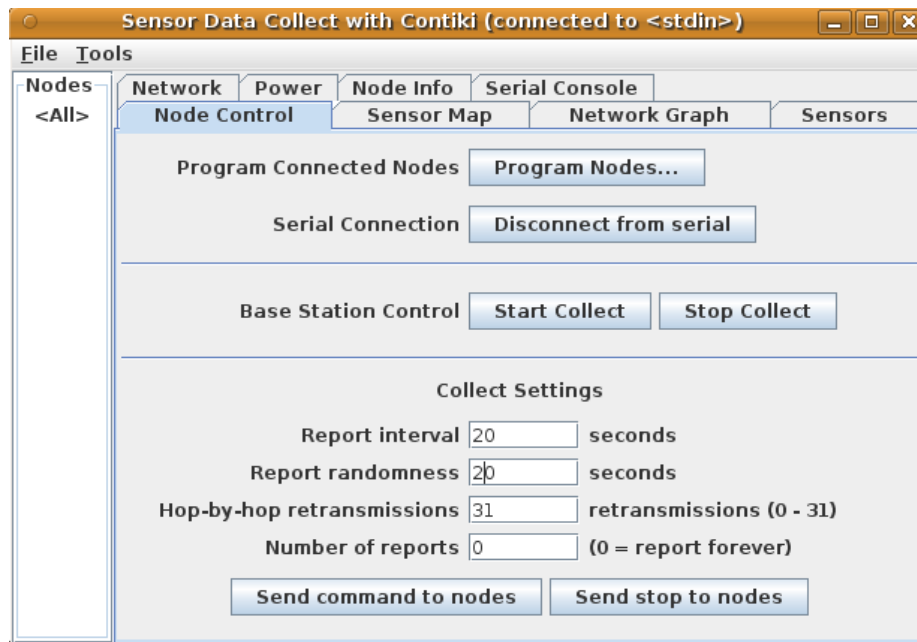
**Hình 11:** Phân bố của các nút cảm biến.

**Bước 6:** Mở giao diện Collect View của nút số 30 và thiết lập các tham số mô phỏng.



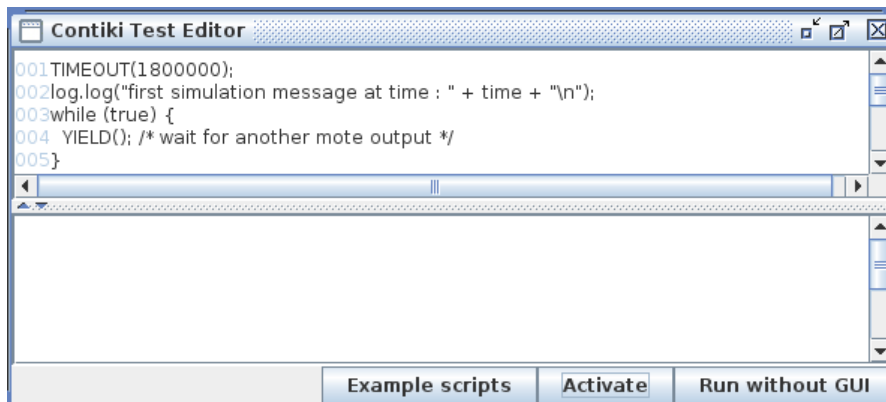
**Hình 12:** Mở giao diện Collect View của nút 30.





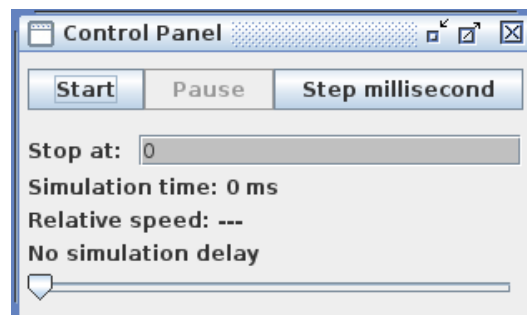
**Hình 13:** Cửa sổ giao diện Collect View của nút 30.

**Bước 7:** Thiết lập thời gian dừng mô phỏng tại các thời điểm mong muốn để thu thập dữ liệu và vẽ đồ thị. Hình 14 minh họa thiết lập thời gian chạy mô phỏng là 30 phút.



**Hình 14:** Cửa sổ giao diện Contiki Test Editor.

**Bước 8:** Chạy mô phỏng: Kích chọn **Start**



**Hình 15:** Cửa sổ giao diện Control Panel.

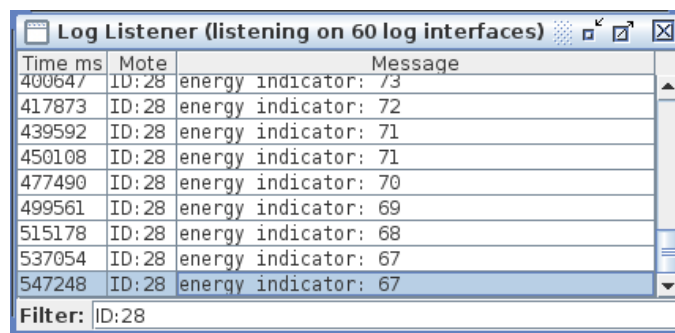
**Bước 9:** Từ giao diện Collect View của nút 30 (Hình 13):

- Gửi các tham số cấu hình (chu kỳ gửi bản tin dữ liệu, số lần truyền lại) tới tất cả các nút trong mạng: Kích chọn **Send command to nodes**.

- Thiết lập nút 30 là nút Sink để bắt đầu thu thập dữ liệu, đồng thời gửi thông báo tới tất cả các nút trong mạng: Kích chọn **Start Collect**.

**Bước 10:** Thu thập dữ liệu tại các thời điểm được xác định trước để vẽ đồ thị.

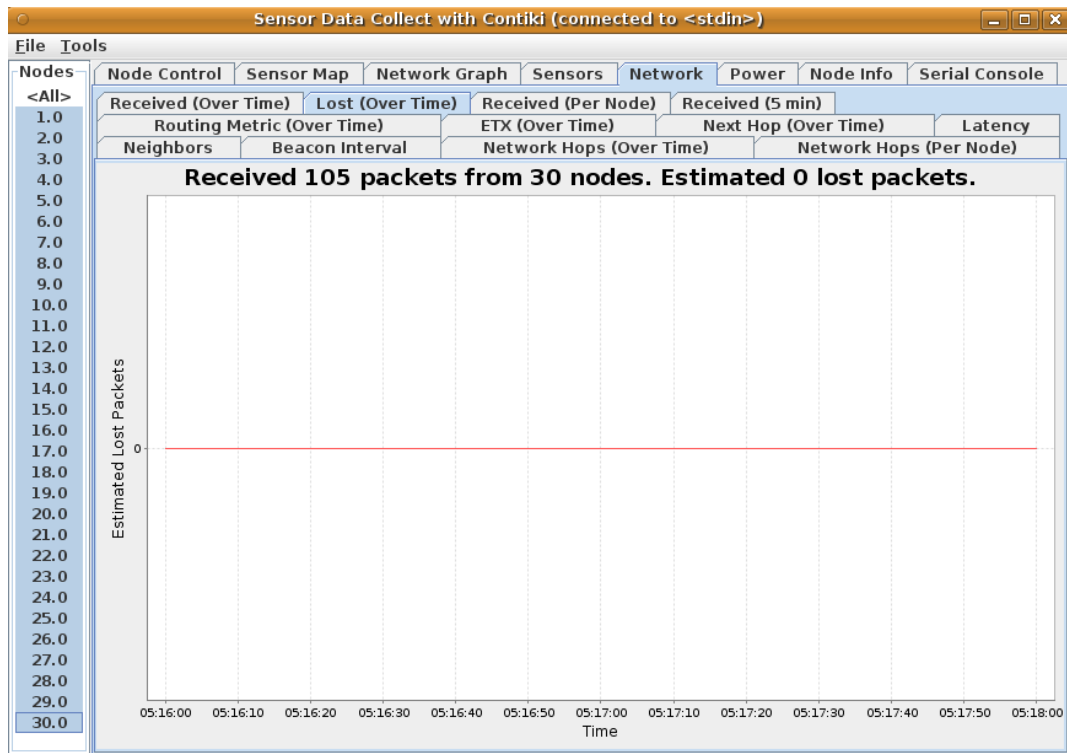
- Thu thập dữ liệu về sự thay đổi chỉ số năng lượng EI của các nút trong mạng tại cửa sổ Log Listener. Hình 16 minh họa sự thay đổi chỉ số năng lượng EI của nút 28 theo thời gian.



Time ms	Note	Message
400647	ID:28	energy indicator: 73
417873	ID:28	energy indicator: 72
439592	ID:28	energy indicator: 71
450108	ID:28	energy indicator: 71
477490	ID:28	energy indicator: 70
499561	ID:28	energy indicator: 69
515178	ID:28	energy indicator: 68
537054	ID:28	energy indicator: 67
547248	ID:28	energy indicator: 67

Filter: ID:28

**Hình 16:** Cửa sổ giao diện Log Listener.



**Hình 17:** Giao diện Lost (Over Time).

- Thu thập dữ liệu về số bản tin nhận được, số bản tin mất trong **Tab Network > Lost (Over time)** tại giao diện Collect View của nút Sink.

- Sao chép toàn bộ dữ liệu tổng hợp được vào một File Excel để xử lý.

**Bước 11:** Quay lại bước 6 để thu thập dữ liệu tại các thời điểm tiếp theo. Quá trình này được lặp lại cho đến thời điểm cuối cùng được vẽ trên đồ thị.

**Bước 12:** Từ bảng dữ liệu tổng hợp được trong File Excel:

- Xác định các tham số ANR, EIB, số bản tin nhận được, số bản tin mất theo thời gian dựa vào các công thức trong mục 2.4.1.

- Vẽ đồ thị biến thiên các tham số này theo thời gian.

## **Phụ lục 5. Đánh giá thực nghiệm với TUmote.**

**Bước 1:** Chuẩn bị mã nguồn chương trình cho giao thức EACTP (EACTP.c), giao thức CTP (CTP.c).

- Biên dịch mã nguồn chương trình EACTP và CTP thành File EACTP.hex và CTP.hex.

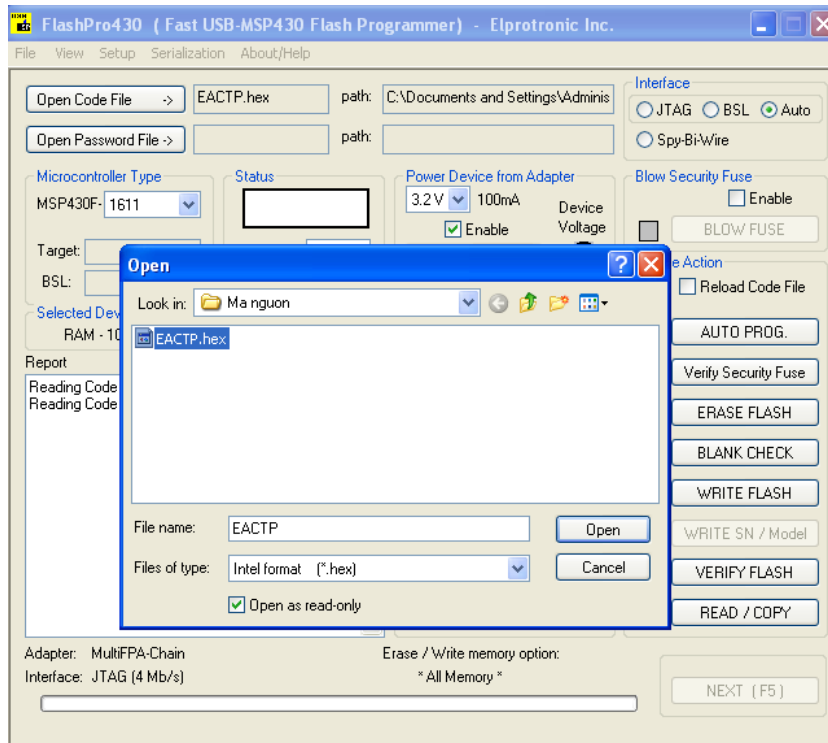
**Bước 2:** Nạp chương trình cho các nút cảm biến qua mạch nạp.

- Nạp mã nguồn chương trình EACTP.hex cho các nút cảm biến thông qua mạch nạp khi đánh giá giao thức EACTP.

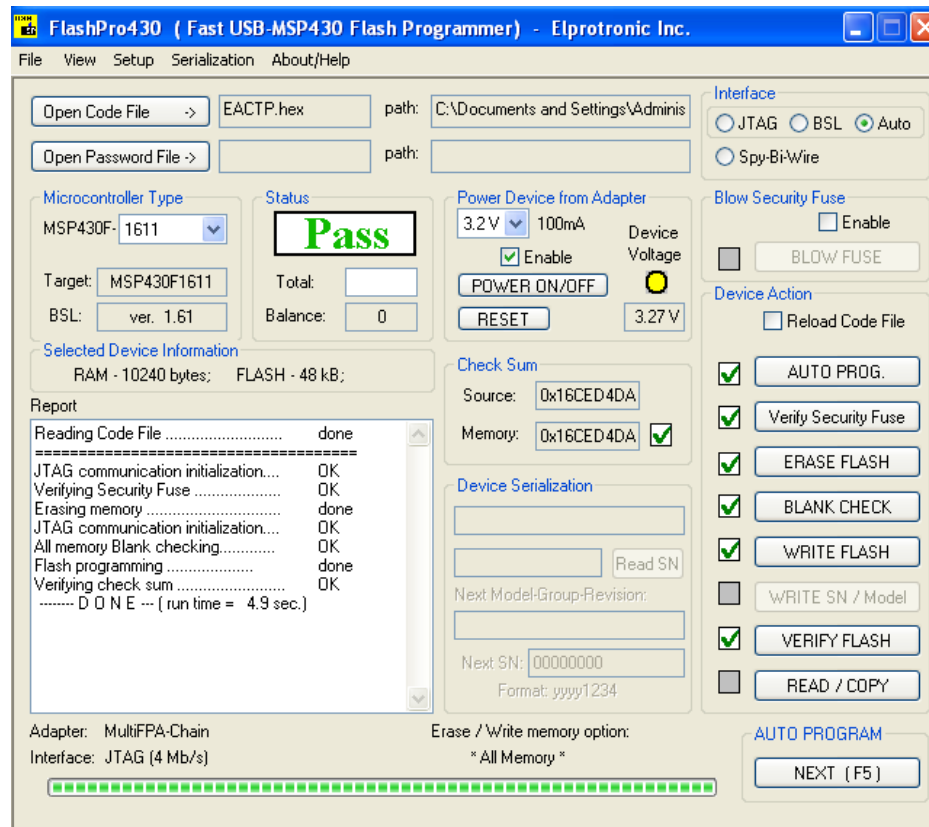
- Nạp mã nguồn chương trình CTP.hex cho các nút cảm biến thông qua mạch nạp khi đánh giá giao thức CTP.



**Hình 18:** Kết nối TUmote với mạch nạp.



Hình 19: Chọn File mã nguồn cần nạp.



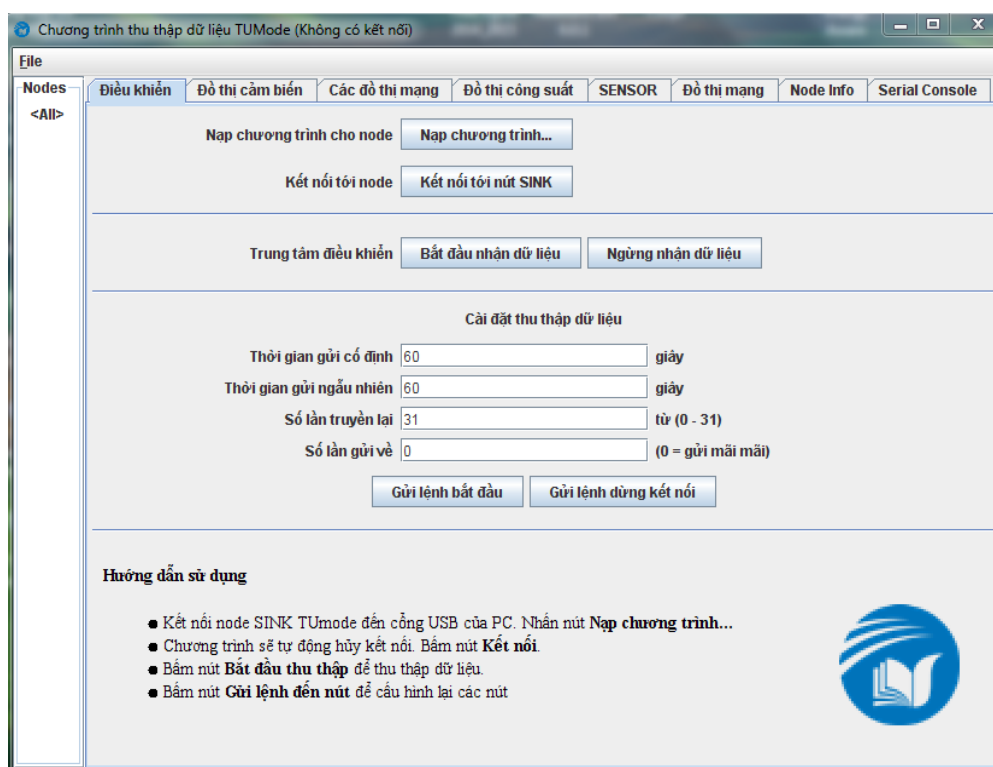
Hình 20: Chọn AUTO PROG. để nạp chương trình cho TUMote .

**Bước 3:** Lắp pin và triển khai các nút cảm biến theo kịch bản đánh giá. Nút Sink được kết nối với máy tính qua cổng USB.



**Hình 21:** Triển khai các nút cảm biến TUmode.

**Bước 4:** Mở chương trình thu thập dữ liệu trên máy tính để **Kết nối tới nút Sink**.



**Hình 22:** Giao diện chương trình thu thập dữ liệu trên máy tính.

**Bước 5:** Thực hiện Reset các thiết bị phần cứng TUmote bằng cách nhấn nút Reset trên thiết bị phần cứng. Khi đó, các thiết bị phần cứng TUmote bắt đầu khởi tạo lại giao thức định tuyến.

**Bước 6:** Từ giao diện chương trình phần mềm thu thập dữ liệu trên máy tính (Hình 22):

- Gửi các tham số cấu hình (chu kỳ gửi bản tin dữ liệu, số lần truyền lại) tới tất cả các nút trong mạng: Kích chọn **Gửi lệnh bắt đầu**.

- Thiết lập nút đang kết nối với máy tính là nút Sink để bắt đầu thu thập dữ liệu, đồng thời gửi thông báo tới tất cả các nút trong mạng: Kích chọn **Bắt đầu nhận dữ liệu**.

**Bước 7:** Thu thập dữ liệu trên máy tính.

- Các nút phần cứng TUmote định kỳ gửi các bản tin dữ liệu về nút Sink cho đến khi chúng hết năng lượng ( $EI = 0$ ).

- Chỉ số năng lượng EI của các nút được cho vào một trường trong phần tải tin của bản tin dữ liệu để gửi về chương trình thu thập dữ liệu trên máy tính.

- Chương trình thu thập trên máy tính dựa vào (trường địa chỉ nút nguồn và trường số thứ tự của bản tin) để xác định số bản tin nhận được và số bản tin mất của từng nút.

- Dữ liệu về sự thay đổi chỉ số năng lượng EI của các nút trong mạng, số bản tin nhận được, số bản tin mất được cập nhật tự động vào cơ sở dữ liệu Access trên máy tính.

**Bước 8:** Sao chép toàn bộ dữ liệu sang File Excel để xử lý. Từ bảng dữ liệu tổng hợp được trong File Excel:

- Xác định các tham số ANR, EIB, số bản tin nhận được, số bản tin mất theo thời gian dựa vào các công thức trong mục 2.4.1.

- Vẽ đồ thị biến thiên các tham số này theo thời gian.