

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

---

**NGUYỄN DUY TÂN**

**NGHIÊN CỨU GIAO THỨC ĐỊNH TUYẾN**  
**TIẾT KIỆM NĂNG LƯỢNG CHO MẠNG SENSOR**

**LUẬN ÁN TIẾN SỸ CÔNG NGHỆ THÔNG TIN**

**Hà Nội – 2017**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**NGUYỄN DUY TÂN**

**NGHIÊN CỨU GIAO THỨC ĐỊNH TUYẾN**  
**TIẾT KIỆM NĂNG LƯỢNG CHO MẠNG SENSOR**

Chuyên ngành: Truyền Dữ liệu và Mạng Máy tính  
Mã số:

**LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN**

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

**Hà Nội – 2017**

## **LỜI CẢM ƠN**

Nghiên cứu sinh Nguyễn Duy Tân xin được bày tỏ lòng biết ơn sâu sắc đến thầy hướng dẫn khoa học của mình là PGS. TS. Nguyễn Đình Việt, những chỉ dẫn tận tình và sự động viên, khích lệ của thầy đã giúp nghiên cứu sinh hoàn thành luận án này.

Nghiên cứu sinh xin chân thành cảm ơn ban lãnh đạo Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội đã tạo môi trường thuận lợi và điều kiện nghiên cứu tốt cho nghiên cứu sinh trong suốt quá trình làm nghiên cứu. Đồng thời, nghiên cứu sinh cũng xin được cảm ơn các thầy, cô Bộ môn Truyền thông và Mạng máy tính, phòng thí nghiệm Hệ thống nhúng, các thầy, cô Khoa Công nghệ Thông tin Trường Đại học Công nghệ đã hỗ trợ nghiên cứu sinh trong suốt quá trình học tập, nghiên cứu và bảo vệ luận án, các NCS, HVCH và sinh viên đã tham gia seminar của phòng thí nghiệm Hệ thống nhúng trong những năm qua.

## **LỜI CAM ĐOAN**

Tôi xin cam đoan luận án “Nghiên cứu giao thức định tuyến tiết kiệm năng lượng cho mạng sensor” là do tôi thực hiện dưới sự hướng dẫn tận tình của PGS. TS. Nguyễn Đình Việt, và không chứa bất kỳ nội dung nào được sao chép từ các công trình đã được người khác công bố. Các tài liệu trích dẫn là trung thực và được chỉ rõ nguồn gốc.

Tôi xin hoàn toàn chịu trách nhiệm về lời cam đoan trên.

*Hà Nội, ngày    tháng    năm 2017*

## MỤC LỤC

<b>DANH MỤC CÁC THUẬT NGỮ .....</b>	<b>vi</b>
<b>BẢNG CÁC KÝ HIỆU .....</b>	<b>ix</b>
<b>DANH MỤC CÁC BẢNG .....</b>	<b>xii</b>
<b>DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ .....</b>	<b>xiii</b>
<b>Chương 1: MỞ ĐẦU .....</b>	<b>1</b>
1.1. Mạng cảm biến không dây .....	1
1.1.1. Sự ra đời của mạng cảm biến không dây .....	3
1.1.2. Các ứng dụng điển hình của mạng cảm biến không dây .....	3
1.1.2.1. Các ứng dụng đã được áp dụng trong thực tế .....	3
1.1.2.2. Các ứng dụng trong tương lai và các yêu cầu kèm theo .....	5
1.1.3. Các vấn đề phải nghiên cứu, giải quyết .....	6
1.2. Tình hình nghiên cứu trên thế giới .....	8
1.3. Tình hình nghiên cứu ở Việt Nam .....	10
1.4. Mục tiêu nghiên cứu của luận án và các vấn đề được giải quyết .....	12
1.4.1. Các giả thiết .....	12
1.4.2. Các mục tiêu cụ thể .....	13
1.5. Nội dung luận án .....	13
1.6. Đóng góp của luận án .....	15
<b>Chương 2: ĐỊNH TUYẾN VÀ ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG TRONG MẠNG CẢM BIẾN KHÔNG DÂY .....</b>	<b>17</b>
2.1. Giải pháp tiết kiệm năng lượng trong mạng cảm biến không dây .....	17
2.1.1. Giải pháp tiết kiệm năng lượng trong kiến trúc nút cảm biến .....	17
2.1.2. Giải pháp tiết kiệm năng lượng trong điều khiển truy nhập môi trường truyền dẫn không dây .....	18
2.1.3. Giải pháp tổng hợp dữ liệu .....	19
2.2. Định tuyến trong mạng cảm biến không dây .....	20
2.2.1. Phân loại các giao thức định tuyến trong mạng cảm biến không dây .....	21
2.2.2. Các giao thức kiến trúc phẳng .....	22
2.2.3. Các giao thức định tuyến theo thông tin địa lý .....	25

2.2.4. Các giao thức dựa trên chất lượng dịch vụ.....	27
2.2.5. Các giao thức có thứ bậc .....	28
2.2.5.1. Phân cụm hiệu quả năng lượng dựa trên xác suất .....	29
2.2.5.2. Định tuyến phân cụm tập trung .....	33
2.2.5.3. Phân cụm hiệu quả năng lượng dựa trên chuỗi .....	35
2.2.5.4. Phân cụm hiệu quả năng lượng dựa trên cây tối thiểu.....	40
2.2.5.5. Giao thức ngưỡng nhảy cảm năng lượng thấp .....	43
2.3. Phân cụm tổng hợp dữ liệu.....	45
2.4. Tổng kết chương .....	47
<b>Chương 3: ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG DỰA TRÊN PHÂN CỤM .....</b>	<b>50</b>
3.1. Kỹ thuật định tuyến phân cụm phân tán.....	50
3.2. Đề xuất cải tiến giao thức LEACH.....	53
3.3. Mô phỏng để đánh giá hiệu quả của đề xuất cải tiến giao thức LEACH.....	60
3.4. Phân tích và so sánh với các thuật toán cùng hướng khác .....	66
3.5. Tổng kết chương .....	67
<b>Chương 4: ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG DỰA TRÊN CHUỖI .....</b>	<b>69</b>
4.1. Đặt vấn đề.....	69
4.2. Phân tích tổng hợp dữ liệu.....	70
4.3. Đề xuất cải tiến thuật toán xây dựng chuỗi dài .....	73
4.3.1. Giai đoạn chọn nút cụm trưởng (CH) .....	73
4.3.2. Giai đoạn xây dựng chuỗi.....	74
4.3.3. Giai đoạn tổng hợp dữ liệu trong chuỗi .....	75
4.3.4. Giai đoạn truyền dữ liệu.....	78
4.4. Mô phỏng để đánh giá hiệu quả của đề xuất cải tiến giao thức DFCB .....	78
4.5. Phân tích và so sánh với các thuật toán cùng hướng khác .....	82
4.6. Đề xuất cải tiến lược đồ xây dựng cụm chuỗi .....	82
4.6.1. Giai đoạn thiết lập cụm .....	85
4.6.2. Giai đoạn cảm biến và truyền dữ liệu .....	94
4.7. Mô phỏng để đánh giá hiệu quả của giao thức đề xuất SCBC.....	94

4.7.1. Phân tích, đánh giá và so sánh thời gian sống của mạng khi áp dụng các giao thức PEGASIS, IEEPB và SCBC .....	95
4.7.2. Phân tích, đánh giá và so sánh năng lượng tiêu thụ của mạng khi áp dụng các giao thức PEGASIS, IEEPB và SCBC.....	96
4.8. Phân tích và so sánh với các thuật toán cùng hướng khác .....	98
4.9. Tổng kết chương.....	98
<b>Chương 5: ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG DỰA TRÊN CÂY TỐI THIỂU .....</b>	<b>100</b>
5.1. Lược đồ định tuyến kết hợp với tổng hợp dữ liệu trên cây .....	100
5.2. Đề xuất cải tiến thuật toán xây dựng cụm cây .....	102
5.2.1. Giai đoạn thiết lập cụm cây .....	102
5.2.2. Giai đoạn truyền dữ liệu .....	107
5.3. Kết hợp với lập lịch ngủ .....	108
5.3.1. Đặt vấn đề .....	108
5.3.2. Phân tích tiêu thụ năng lượng trong giao thức SSTBC.....	108
5.3.3. Thuật toán lập lịch ngủ .....	113
5.4. Phân tích, đánh giá và so sánh thời gian sống của mạng khi áp dụng các giao thức PEGASIS, STDC, DFTBC và SSTBC.....	117
5.4.1. Các tham số mô phỏng .....	117
5.4.2. Kết quả mô phỏng DFTBC.....	117
5.4.3. Kết quả mô phỏng giao thức SSTBC .....	121
5.5. Phân tích và so sánh với các thuật toán cùng hướng khác .....	123
5.6. Tổng kết chương.....	124
<b>Chương 6. KẾT LUẬN.....</b>	<b>125</b>
<b>DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN .....</b>	<b>128</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>129</b>
<b>PHỤ LỤC.....</b>	<b>142</b>
Phụ lục 1. Lý thuyết Dempster-Shafer .....	142
Phụ lục 2. Mã nguồn phân tán (Distributed source coding).....	147

## DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng Anh	Diễn giải
1	APTEEN	Adaptive Threshold sensitive Energy Efficient Sensor Network Protocol	Giao thức mạng cảm biến hiệu quả năng lượng ngưỡng nhạy cảm thích ứng
2	BS	Base Station	Trạm cơ sở
3	CEEC	Centralized Energy Efficient Clustering	Phân cụm tập trung cho hiệu quả năng lượng
4	CH	Cluster Head	Cụm trưởng
5	CO <sub>2</sub>	Carbon Dioxide	Đi-ô-xit cac bon
6	DB-LEACH	Distance-based LEACH	LEACH dựa trên khoảng cách
7	DD	Directed Diffusion	Truyền thông trực tiếp
8	DFCB	Data Fusion and Chain-Based Clustering	Tổng hợp dữ liệu và phân cụm dựa trên chuỗi
9	DFTBC	Data Fusion and Tree-Based Clustering	Tổng hợp dữ liệu và phân cụm dựa trên cây
10	DKF	Distributed Kalman Filter	Lọc Kalman phân tán
11	DSC	Distributed source coding	Mã nguồn phân tán
12	EAP	Energy-Aware Routing Protocol	Giao thức định tuyến nhận biết năng lượng
13	EECS	Energy Efficient Clustering Scheme	Lược đồ phân cụm cho hiệu quả năng lượng
14	EE-TLDC	Energy Efficient Two Level Distributed Clustering	Phân cụm phân tán hai mức cho hiệu quả năng lượng
15	EEUCA	Energy-efficient Uniform Clustering Algorithm	Thuật toán phân cụm đồng dạng hiệu quả năng lượng
16	EPLEACH	Enhancing the Performance of LEACH	Nâng cao hiệu năng cho LEACH
17	GA	Greedy Algorithm	Thuật toán tham lam
18	GAF	Geographical Adaptive Fidelity	Sự chính xác thích ứng theo địa lý
19	GEAR	Geographical Energy Aware	Định tuyến cảm nhận năng



		Routing	lượng và địa lý
20	GEM	Graph Embedding for Routing	Định tuyến nhúng đồ thị
21	GPS	Global Positioning System	Hệ thống định vị toàn cầu
22	GSSC	Geography-Informed Sleep Scheduling and Chaining Based Routing	Lập lịch ngủ có báo thông tin địa lý và định tuyến dựa trên chuỗi
23	GSTEB	General Self-Organized Tree-Based Energy-Balance	Cân bằng năng lượng dựa trên cây tự tổ chức tổng quát
24	HEED	Hybrid, Energy-Efficient, Distributed Clustering	Phân cụm phân tán, hiệu quả năng lượng, lai ghép
25	ID	Identification	Định danh
26	IEEE	Institute of Electrical and Electronics Engineers	Viện kỹ nghệ Điện và Điện tử
27	IIEPB	Improved Energy-Efficient PEGASIS-Based protocol	Giao thức dựa trên PEGASIS hiệu quả về năng lượng được cải tiến
28	LBEERA	Load Balance and Energy Efficient Routing Algorithm	Thuật toán định tuyến hiệu quả năng lượng và cân bằng tải
29	LEACH	Low-Energy Adaptive Clustering Hierarchy	Phân cấp phân cụm thích ứng năng lượng thấp
30	LEACH-C	LEACH - Centralized	LEACH - Tập trung
31	LEECF	Low-Energy Event Centric Fusion	Tổng hợp tập trung theo sự kiện, năng lượng thấp
32	MAC	Medium Access Control	Điều khiển truy nhập đường truyền
33	MAMC	Multiple Aggregator Multiple Chain	Đa tổng hợp đa chuỗi
34	MANET	Mobile Ad hoc NETwork	Mạng tùy biến di động
35	MEMS	Micro-Electro-Mechanical System	Hệ thống vi cơ điện tử
36	MERIG	Minimum Energy Reliable Information Gathering	Tập hợp thông tin tin cậy năng lượng tối thiểu

37	NS2	Network Simulator Version 2.0	Công cụ mô phỏng mạng phiên bản 2
38	OMLEACH	An Optimal Mechanism of LEACH	Một kỹ thuật tối ưu cho LEACH
39	PEGASIS	Power-Efficient Gathering in Sensor Information Systems	Tập hợp hiệu quả năng lượng trong hệ thống thông tin cảm biến
40	RSSI	Received Signal Strength Indication	Chỉ thị độ mạnh tín hiệu nhận được
41	SAR	Sequential Assignment Routing	Định tuyến phân chia tuần tự
42	SCBC	Sector-Chain Based Clustering	Phân cụm dựa trên cung chuỗi
43	SCH	Secondary Cluster Head	Cụm trưởng thứ hai
44	SNGF	Stateless Non-deterministic Geographic Forwarding	Chuyển tiếp địa lý không xác định phi trạng thái
45	SPEED	Stateless Protocol for Real-Time Communication	Giao thức phi trạng thái truyền thông thời gian thực
46	SPIN	Sensor Protocols for Information via Negotiation	Giao thức cảm biến cho thông tin thông qua đàm phán
47	SSTBC	Sleep Scheduled and Tree-Based Clustering	Phân cụm dựa trên cây và lập lịch ngủ
48	TBC	Tree-Based Clustering	Phân cụm dựa trên cây
49	TCDGP	Tree-Clustered Data Gathering Protocol	Giao thức thu thập dữ liệu cây phân cụm
50	TEEN	Threshold sensitive Energy Efficient Sensor Network Protocol	Giao thức mạng cảm biến hiệu quả năng lượng nhạy với ngưỡng
51	TTL	Time To Live	Thời gian sống
52	U-LEACH	Universal - LEACH	LEACH - phổ biến
53	VLSI	Very-large-scale integration	Tích hợp phạm vi rất lớn
54	WSN	Wireless Sensor Network	Mạng cảm biến không dây

## BẢNG CÁC KÝ HIỆU

Ký hiệu	Diễn giải
$\arctan(X)$	Giá trị hàm ác tang của X
$bel(A)$	Hàm chỉ độ đo “ <i>niềm tin</i> ” ( <b>belief</b> ) của A
$caf()$	Hàm tổng hợp dữ liệu
$cf()$	Hàm nén dữ liệu theo mã nguồn phân tán (DSC)
$CH_i$	Nút cụm trưởng (CH) thứ i
$CH_{prob}$	Giá trị xác suất để trở thành CH
$cost(j,i)$	Hàm cost được tính ở nút thứ j cho nút CH thứ i
$d(x,y)$	Khoảng cách từ nút x đến nút y
$D_{avg}$	Khoảng cách trung bình từ các nút CH đến BS
$dc_2$	Hệ số phụ thuộc vào khoảng cách
$d_{crossover}$	Khoảng cách chỉ định vùng phủ sóng của nút cảm biến
$d_{f\_max}$	Khoảng cách lớn nhất của hàm f()
$dH(X, Y)$	Khoảng cách Hamming giữa hai nguồn rời rạc X và Y
$D_{Max}$	Giá trị đường kính mạng, được tính sau khi triển khai mạng
$d_{toBS}$	Khoảng cách từ mạng đến BS
$E_{amp}$	Năng lượng yêu cầu cho bộ khuếch đại để truyền theo mô hình hai tia mặt đất
$E_{average}$	Năng lượng trung bình của các nút cảm biến còn sống trong vòng hiện tại
$E_{avg}$	Năng lượng tiêu thụ trung bình của mạng
$E_{brc}$	Năng lượng tiêu thụ của một nút quảng bá gói tin đến BS
$ec_1$	Hệ số phụ thuộc vào năng lượng còn lại
$E_{CH}$	Năng lượng tiêu thụ của nút CH
$E_{cluster}$	Tổng năng lượng tiêu thụ trong mỗi cụm (cây)
$E_{control}$	Năng lượng tiêu thụ bởi các nút trong một cụm (chuỗi) khi chúng trao đổi thông điệp với BS trong một vòng
$E_{elec}$	Năng lượng yêu cầu để chạy mạch điện cho bộ thu phát sóng vô tuyến

$E_{DA}(q)$	Năng lượng tiêu thụ cho việc tổng hợp q bit dữ liệu
$E_{friis}$	Năng lượng yêu cầu cho bộ khuếch đại để truyền theo mô hình không gian trống
$E_G$	Năng lượng tiêu thụ cho việc sinh ra gói tin
$E_{init}$	Năng lượng khởi tạo pin dùng cho mô phỏng
$E_{non-CH}$	Năng lượng tiêu thụ của nút không phải CH
$E_{residual}(i)$	Năng lượng còn lại của nút i ở vòng hiện tại
$E_{rm}$	Năng lượng tiêu thụ của nút trong hoạt động, nhận thông điệp từ BS
$E_{round}$	Tổng năng lượng tiêu thụ cho một vòng
$E_{RX}(q)$	Năng lượng tiêu thụ cho nhận q bit dữ liệu
$E_S$	Năng lượng tiêu thụ cho việc cảm biến, đo
$E_{SCH}$	Năng lượng tiêu thụ của nút cụm trưởng thứ cấp
$E_{threshold}$	Giá trị chỉ ngưỡng năng lượng
$E_{total}$	Tổng năng lượng tiêu thụ của mạng
$E_{TX}(q, d)$	Năng lượng tiêu thụ khi truyền q bit dữ liệu qua khoảng cách d
$f_{criterion}$	Hàm tiêu chuẩn chọn nút cụm trưởng để gia nhập nhóm
$\gamma$	Hằng số phản ánh đơn vị năng lượng nhỏ nhất và được thay đổi tùy theo yêu cầu
$G$	Tập hợp các nút không được chọn làm CH trong (1/k) vòng cuối
$GF(2)$	Trường Galois cho 2 phần tử
$h$	Số nút hàng xóm của nút thứ i
$H(X Y)$	Hàm Entropi có điều kiện của nguồn X
$HopDelay_i^j$	Độ trễ ước tính giữa nút i và j
$h_r$	Chiều cao của ăng ten thu
$h_t$	Chiều cao của ăng ten phát
$k, k_{opt}$	Tỷ lệ phần trăm nút cụm trưởng, số cụm tối ưu trong mạng
$\lambda$	Bước sóng
$l$	Giá trị suy giảm của hệ thống truyền không dây
$m$	Số gói tin được truyền trong giai đoạn ổn định truyền dữ liệu của một nút

$m(A)$	Hàm " <i>khối lượng</i> " tương quan và tính hiển nhiên sẵn có của A
$m_x$	Giá trị trung bình
$\sigma$	Độ lệch chuẩn
$\xi$	Tỉ lệ độ lệch chuẩn
$n$	Tổng số nút còn sống ở vòng hiện tại
$N$	Tổng số nút trong mạng
$nn$	Tổng số nút thành viên trong cụm còn sống ở vòng hiện tại
$N_R$	Tổng số vòng đã hoạt động sau khi nút đầu tiên trong mạng chết
$pl(A)$	Hàm chỉ định độ đo về sự " <i>đáng tin</i> "
$p_{min}$	Giá trị ngưỡng để đảm bảo $CH_{prob}$ không nhỏ hơn $p_{min}$
$Pr(X)$	Xác suất xuất hiện của X
$Q$	Thông lượng mạng
$\Theta$	Không gian mẫu
$r$	Vòng hiện tại
$r_i$	Số vòng liên tiếp mà nút i không được làm nút CH
$SPEED_i^j(D)$	Tốc độ chuyển tiếp dữ liệu giữa nút i và nút j đến đích D
$T(i)$	Giá trị ngưỡng của nút i
$t_{dt}$	Khoảng thời gian trong giai đoạn ổn định truyền dữ liệu
$T_{frame}$	Khoảng thời gian đủ để một gói tin được truyền
$V(i)$	Giá trị tính ở nút i để xem xét chọn làm nút cụm trưởng ở vòng r
$X_{BS}$	Tọa độ theo trục X của nút trung chuyển BS
$Y_{BS}$	Tọa độ theo trục Y của nút trung chuyển BS
$\psi$	Khoảng thời gian tồn tại trong giai đoạn thiết lập cụm
$\{w_n, v\}_{n \times v}$	Ma trận niềm tin của n nút cảm biến, quan sát v trọng số xác định

## DANH MỤC CÁC BẢNG

Bảng 2.1: So sánh các thuật toán phân cụm.....	34
Bảng 2.2: So sánh các thuật toán phân cụm dựa trên chuỗi.....	40
Bảng 2.3: So sánh các thuật toán phân cụm dựa trên cây .....	43
Bảng 3.1: Các tham số môi trường mô phỏng.....	59
Bảng 4.1: Tỷ lệ gói tin nhận được ở BS và tỷ lệ nút chết khi thay đổi $t_{\text{round}}$ .....	90
Bảng P1.1: Kết quả quan sát của các phần tử trong tập $\Theta$ .....	143
Bảng P1.2: Kết quả tính toán “ <i>niềm tin</i> ” và sự “ <i>đáng tin</i> ” của A và B .....	146

## DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1: Một mạng cảm biến không dây .....	2
Hình 2.1: Tổng hợp dữ liệu theo mô hình phân cụm đơn chặng .....	19
Hình 2.2: Tổng hợp dữ liệu theo mô hình phân cụm đa chặng .....	20
Hình 2.3: Lược đồ thu thập và tập hợp dữ liệu song song trên chuỗi trong một vòng .....	37
Hình 2.4: Lược đồ phân chia mức và xây dựng cây [54] .....	42
Hình 3.1: Tô-pô cơ bản của giao thức LEACH trong một vòng .....	51
Hình 3.2: Hoạt động hai giai đoạn trong một vòng của LEACH .....	53
Hình 3.3: Mô hình phân cụm của LEACH .....	54
Hình 3.4: Sơ đồ hoạt động của giao thức LEACH-DE .....	56
Hình 3.5: Sơ đồ mạng cảm biến không dây gồm 7 nút .....	58
Hình 3.6: Tỷ lệ nút còn sống giảm theo thời gian; vị trí BS ở (49,175) .....	63
Hình 3.7: Tỷ lệ nút còn sống giảm theo thời gian; vị trí BS ở (49,225) .....	64
Hình 3.8: Tổng năng lượng mạng tiêu thụ áp dụng với ba giao thức; vị trí BS ở (49,175) .....	64
Hình 3.9: Tổng năng lượng mạng tiêu thụ áp dụng với ba giao thức; vị trí BS ở (49,225) .....	64
Hình 3.10: Tỷ lệ (phần trăm) nút chết theo thời gian .....	65
Hình 3.11: Tỷ lệ (phần trăm) gói tin nhận được ở BS .....	66
Hình 4.1: Mô hình truyền dữ liệu trong chuỗi, (a) không tổng hợp dữ liệu và (b) có tổng hợp dữ liệu .....	70
Hình 4.2: Tỷ lệ nút còn sống trong suốt thời gian mô phỏng .....	80
Hình 4.3: Tổng năng lượng mạng tiêu thụ khi áp dụng ba giao thức .....	80
Hình 4.4: Tỷ lệ các nút chết theo thời gian mô phỏng .....	81
Hình 4.5: Tỷ lệ gói dữ liệu nhận được ở BS .....	81
Hình 4.6: Phân chia mạng thành các cung (cụm) .....	85

Hình 4.7: Một ví dụ xây dựng chuỗi giữa các thuật toán: (a) PEGASIS, (b) IEEPB và (c) SCBC .....	91
Hình 4.8: Kết quả phân cụm mạng sử dụng giao thức SCBC trong một vòng.....	92
Hình 4.9: Tỷ lệ nút còn sống trong suốt thời gian mô phỏng .....	96
Hình 4.10: Năng lượng tiêu thụ của mạng theo thời gian.....	97
Hình 4.11: Tỷ lệ gói tin nhận được ở BS khi vị trí của BS thay đổi .....	98
Hình 5.1: Ví dụ minh họa tổng hợp dữ liệu dựa trên cụm cây phân cấp.....	101
Hình 5.2: (a) Minh họa phân cụm mạng gồm 100-nút trong vùng 100m×100m, (b) một ví dụ về cây khung nhỏ nhất.....	102
Hình 5.3: Hoạt động hai giai đoạn trong một vòng của SSTBC.....	109
Hình 5.4: Năng lượng tiêu thụ trung bình trong mỗi vòng với số cụm khác nhau	112
Hình 5.5: Thông lượng trung bình Q nhận được ở BS trong mỗi vòng với số cụm khác nhau.....	112
Hình 5.6: Ví dụ về kích thước của các ô lưới ảo nhỏ hơn hoặc bằng 20m × 20m.	114
Hình 5.7: (a) Minh họa phân cụm mạng gồm 100 nút trong vùng 100m×100m với kích thước lưới ô ảo 5m ×5m, (b) một ví dụ về cây khung tối thiểu.....	116
Hình 5.8: Tỷ lệ nút mạng còn sống trong suốt thời gian mô phỏng.....	117
Hình 5.9: Tỷ lệ (phần trăm) nút chết khi vị trí BS ở (49,175) .....	118
Hình 5.10: Năng lượng tiêu thụ của mạng khi áp dụng các giao thức .....	119
Hình 5.11: Tỷ lệ gói tin nhận được ở BS khi vị trí thay đổi .....	120
Hình 5.12: Tỷ lệ nút còn sống trong suốt thời gian mô phỏng .....	121
Hình 5.13: Trung bình năng lượng tiêu thụ áp dụng bởi ba giao thức .....	122
Hình 5. 14: Tỷ lệ (phần trăm) gói tin nhận được ở BS khi vị trí BS thay đổi .....	123
Hình P1.1: Khoảng không chắc chắn giữa “ <i>niềm tin</i> ”, sự “ <i>đáng tin</i> ” .....	144
Hình P2.1: Mã hóa độc lập và giải mã đồng thời hai nguồn dữ liệu tương quan X và Y .....	147
Hình P2.2: Triển khai mã nguồn phân tán trong mạng cảm biến không dây .....	147
Hình P2.3: Biểu diễn giá trị tập con $Z_{xxx}$ với $u=7$ bit.....	150



## Chương 1: MỞ ĐẦU

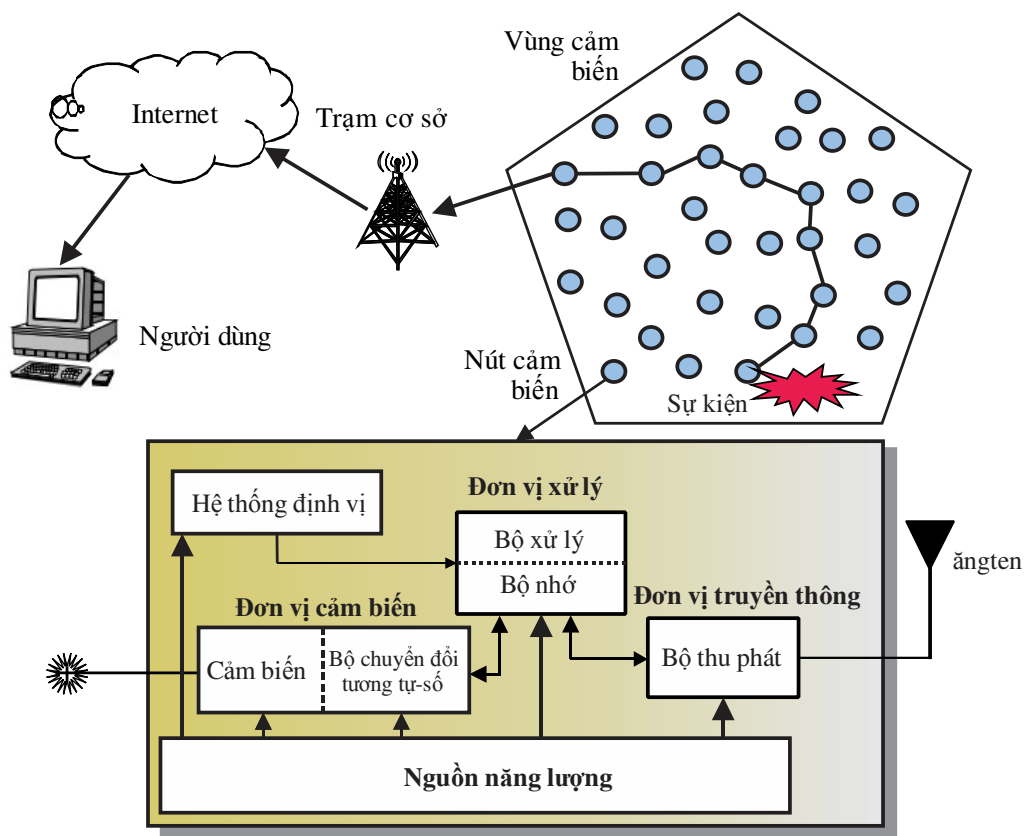
### 1.1. Mạng cảm biến không dây

Các nút cảm biến không dây (wireless sensors) có khả năng cảm biến, thu thập thông tin từ thế giới vật lý rồi chuyển đổi thành dạng tín hiệu số. Nó có thể lưu trữ, truyền tải và xử lý thông qua máy tính và mạng Internet đến người dùng. Khi các thiết bị cảm biến được tích hợp vào các hệ thống khác nhau, chúng đem lại nhiều lợi ích cho đời sống con người như giám sát môi trường, ngôi nhà thông minh, hệ thống giao thông thông minh, hệ thống tự động hóa, chăm sóc sức khỏe, các hệ thống cảnh báo, v.v. Cùng với sự phát triển trong công nghệ MEMS kết hợp với kiến trúc vi xử lý VLSI, hệ thống công nghệ nhúng và truyền thông không dây đã sản xuất ra các nút cảm biến không dây có kích thước nhỏ gọn, giá thành thấp và tiêu thụ năng lượng ít. Thông thường một thiết bị cảm biến chỉ có thể cảm biến được một loại thông tin vật lý như nhiệt độ, độ ẩm, độ rung động, áp suất, nồng độ khí hy-đrô-rô hay khí CO<sub>2</sub>, v.v. [4, 122].

Mạng cảm biến không dây WSN bao gồm nhiều nút cảm biến thường được triển khai dày đặc, ngẫu nhiên trên một vùng rộng lớn (vùng rừng rậm, đồi núi). Chúng có thể di chuyển hoặc đứng yên sau khi được triển khai. Mỗi nút cảm biến không dây được trang bị các thành phần như bộ vi xử lý, bộ nhớ, bộ phận thu/phát sóng vô tuyến và nguồn năng lượng (pin) và có thể có cả hệ thống định vị GPS. Các nút cảm biến có nhiệm vụ thu nhận các tín hiệu vật lý từ môi trường xung quanh, các tín hiệu được chuyển đổi từ dạng tín hiệu tương tự (có thể là hình ảnh, âm thanh, v.v) sang tín hiệu số rồi chuyển tới bộ vi xử lý để tạo khả năng quan sát, phân tích, phản ứng lại với các sự kiện và hiện tượng cụ thể. Bộ vi xử lý và bộ nhớ trong nút cảm biến có dung lượng và khả năng tính toán hạn chế. Do đó, các bộ phận này thường tiêu thụ nguồn năng lượng ít hơn nhiều so với bộ phận truyền/nhận. Bộ phận truyền/nhận tín hiệu không dây có nhiệm vụ điều chế và truyền tín hiệu dưới dạng sóng vô tuyến, đồng thời thu nhận và giải điều chế tín hiệu để vi xử lý có thể hiểu được. Bộ phận thu/phát tiêu thụ năng lượng nhiều hơn các bộ phận khác. Đặc biệt,

khi nó truyền tín hiệu với khoảng cách xa [81, 96]. Hệ thống định vị GPS để cho các nút trong mạng sau khi được triển khai biết vị trí (tọa độ) của nút cảm biến trong mạng. Bộ phận nguồn nuôi (thường là Pin AA có dung lượng khoảng 3000mAh, 1,5V [81], tùy từng hãng sản xuất) có nhiệm vụ cung cấp điện cho nút hoạt động, có kích thước nhỏ và thường không được sạc điện bổ sung hoặc thay mới trong suốt thời gian nút hoạt động. Do đó, sau khi tiêu thụ hết pin, nút cảm biến sẽ ngừng hoạt động.

Các tiêu chuẩn công nghệ được sử dụng phổ biến cho mạng cảm biến không dây bao gồm chuẩn IEEE 802.15.4 và ZigBee dùng cho truyền thông năng lượng thấp trên băng tần 868 MHz, 915 MHz, và 2,4 GHz [90, 94, 127, 128]. Tất cả các bộ phận trên cấu thành một vi máy tính siêu nhỏ có khả năng đo đạc, thu nhận, xử lý, và truyền tải dữ liệu qua mạng Internet đến người dùng sau khi WSN được triển khai trên một khu vực và đi vào hoạt động.



**Hình 1.1:** Một mạng cảm biến không dây

### **1.1.1. Sự ra đời của mạng cảm biến không dây**

Vào đầu những năm 1950, một hệ thống cảm biến âm thanh vùng xa dưới nước, đầu tiên được gọi là hệ thống giám sát âm thanh (Sound Surveillance System - SOSUS), đã được triển khai ở các lưu vực nước sâu của Đại Tây Dương và Thái Bình Dương cho việc giám sát tàu ngầm. Chúng kết hợp các mảng được sử dụng để dò tìm, phát hiện và xác định vị trí các mối nguy hại dưới nước. Gần đây, SOSUS đã được thay thế bởi hệ thống giám sát dưới nước phức tạp hơn [19].

Tiếp theo là một trong những mạng cảm biến không dây ra đời sớm nhất trên thế giới được triển khai trong thời gian chiến tranh Việt Nam năm 1967 [81, 130]. Chúng được gọi là "Igloo White" bao gồm khoảng 30000 nút cảm biến địa chấn và âm thanh được rải dọc theo đường mòn Hồ Chí Minh từ bắc Việt Nam, qua Lào và vào miền nam Việt Nam bằng máy bay trực thăng và máy bay chiến đấu. Các nút cảm biến có thể tự định hình với một ăng-ten ra-đi-ô, chúng được thiết kế giống như những tán lá cây xung quanh. Các nút sử dụng nguồn pin Lithium với thời gian sống khoảng 30 ngày. Các nút thực hiện một ngưỡng cục bộ đơn giản và kích hoạt báo động khi mức độ hoạt động được phát hiện thông qua mạch điện tử. Các nút truyền tín hiệu với công suất 2W trên băng tần UHF theo quỹ đạo của máy bay EC-121R và truyền đến 360 máy tính IBM ở Nakhon Phanom, Thái Lan để phân tích và xử lý. Từ đó, các nhà phân tích sẽ đưa ra các quyết định không kích chống lại các mục tiêu dựa vào tín hiệu nhận được từ các nút cảm biến.

### **1.1.2. Các ứng dụng điển hình của mạng cảm biến không dây**

#### **1.1.2.1. Các ứng dụng đã được áp dụng trong thực tế**

##### **Ứng dụng trong giám sát môi trường:**

Giám sát môi trường được xem là một hướng ứng dụng điển hình cho mạng cảm biến không dây, chúng bao gồm việc theo dõi, đo đạc và tính toán để đưa ra dự báo sớm về các thảm họa thiên tai như động đất, lũ lụt, sự hoạt động của núi lửa, dự báo cháy rừng, theo dõi sự di chuyển, sinh hoạt của các loài động vật hoang dã, nghiên cứu môi trường sống, nghiên cứu thời tiết hoặc địa vật lý, theo dõi các mối nguy

hiểm khác nhau cho xã hội loài người như nghiên cứu sự ô nhiễm của nguồn nước, đo hàm lượng khí CO<sub>2</sub> trong một khu vực dân cư, đo độ ẩm, áp suất không khí, tốc độ gió, v.v. Trong [26, 79, 107] các tác giả đã cho chúng ta thấy khá toàn diện về ứng dụng của mạng cảm biến không dây trong lĩnh vực này.

### **Ứng dụng trong an ninh, quốc phòng:**

Mạng cảm biến không dây với các đặc tính có thể triển khai nhanh, chi phí thấp và rất thích hợp trên khu vực đồi núi, vùng rừng rậm nguy hiểm cùng với mục đích ứng dụng trong quân sự và an ninh quốc phòng. Do đó, các ứng dụng của mạng cảm biến không dây trong lĩnh vực này là rất phong phú, ví dụ như: Giám sát lực lượng, trang thiết bị và đạn dược, giám sát chiến trường, theo dõi trận địa, đo thám lực lượng và trận địa đối phương, xác định mục tiêu; đánh giá tổn thất sau trận chiến, thăm dò, phát hiện và cảnh báo các cuộc tấn công bằng hóa học, sinh học và hạt nhân [4, 96].

### **Ứng dụng trong y tế và chăm sóc sức khỏe:**

Các nút cảm biến không dây có thể được gắn vào vải quần áo của người mặc nó để liên tục đo, theo dõi các thông số sức khỏe như nhiệt độ cơ thể, huyết áp, nhịp tim, điện tâm đồ, điện não đồ, độ bão hòa oxy trong máu v.v. sau đó truyền không dây đến một trạm giám sát từ xa để theo dõi, nghiên cứu tình trạng sức khỏe tổng thể của người mặc [80, 90].

Đặc biệt hơn, các thiết bị cảm biến tin sinh siêu nhỏ có thể được cấy trực tiếp vào cơ thể bệnh nhân để theo dõi các cơn đau tim, các trận hen suyễn, ức chế thần kinh, sự phát triển của ung thư, nồng độ đường trong máu, tỷ lệ hô hấp v.v [83]. Các thông số này được thu thập và gửi đến máy tính cá nhân của các bác sỹ thông qua mạng Internet. Các bác sỹ sẽ dựa vào giá trị của các thông số này mà có thể theo dõi được tình trạng sức khỏe của từng bệnh nhân từ đó đưa ra các phương pháp điều trị phù hợp mà không cần phải đến trực tiếp phòng bệnh nhân.

Cùng với sự phát triển của công nghệ thông tin, sự tăng lên của nhu cầu trong đời sống con người, mạng cảm biến không dây ngày càng được ứng dụng nhiều hơn

trong đời sống con người tạo nên ngôi nhà thông minh, hệ thống giao thông thông minh, v.v. [96].

#### **1.1.2.2. Các ứng dụng trong tương lai và các yêu cầu kèm theo**

**Ngôi nhà thông minh (Smart home):** Cùng với sự phát triển mạng mẽ của công nghệ mạng không dây như hiện nay, mạng cảm biến không dây trong tương lai sẽ đóng vai trò quan trọng trong việc hình thành ngôi nhà thông minh, nơi mà các đồ dùng quen thuộc đều được gắn các nút cảm biến, nó có thể thực hiện các công việc hằng ngày một cách tự động hoặc thông qua sự điều khiển của con người. Ví dụ máy điều hòa nhiệt độ có thể tự điều chỉnh nhiệt độ trong nhà sao cho phù hợp với con người hoặc hệ thống kiểm soát cường độ chiếu sáng của hệ thống đèn trong nhà sao cho phù hợp với khung cảnh và giúp tiết kiệm điện, hệ thống camera giám sát đảm bảo an ninh, hệ thống khóa cửa tự động trong nhà, v.v.

**Cảm biến thông minh (Cognitive sensor):** Các nút cảm biến không dây có thể cảm nhận phổ tần số thông minh trên băng tần được cấp phép hoặc không cấp phép, từ đó truy cập phổ theo cơ hội (opportunistic spectrum) mà không làm ảnh hưởng đến người dùng chính.

**Thành phố thông minh (Smart Cities):** Các nút cảm biến thông minh sẽ đóng vai trò quan trọng trong việc hình thành thành phố thông minh, nơi mà công nghệ thông tin và truyền thông kết nối các nút cảm biến thông minh, mạng không dây tốc độ cao, xử lý dữ liệu lớn được ứng dụng để nâng cao chất lượng cuộc sống nơi đô thị. Ví dụ: Sử dụng cảm biến thông minh để điều khiển hệ thống đèn giao thông hoạt động tiết kiệm năng lượng, điều khiển luồng trong thành phố để tránh tắc nghẽn, chăm sóc sức khỏe tại nhà, giám sát mức độ ô nhiễm trong không khí để kịp thời cung cấp thông tin cảnh báo cho người dân, ...

Để đáp ứng được các ứng dụng trên, yêu cầu các nút cảm biến cần được trang bị nhiều bộ thu phát sóng vô tuyến, có kích thước nhỏ, dung lượng pin lớn và khả năng xử lý, tính toán mạnh hơn nữa. Các thuật toán, phần mềm mạng cài đặt vào các nút cảm biến cần có độ phức tạp tính toán thấp, chạy tiết kiệm năng lượng.

### 1.1.3. Các vấn đề phải nghiên cứu, giải quyết

**Vấn đề định tuyến:** Do các đặc điểm khác biệt của mạng cảm biến không dây như: Các nút cảm biến có kích thước vật lý nhỏ, dung lượng bộ nhớ bé, khả năng tính toán của bộ vi xử lý vừa phải. Đặc biệt là nguồn năng lượng pin hạn chế và sau khi đã triển khai, chúng rất khó có thể được sạc pin hay thay thế nguồn pin mới. Bởi vậy, chúng ta không thể áp dụng các thuật toán định tuyến truyền thống trong mạng MANET vào mạng cảm biến không dây được. Khi thiết kế các giao thức định tuyến, các nhà khoa học cần thiết kế các giao thức định tuyến đơn giản, sử dụng ít tài nguyên và có độ phức tạp tính toán vừa phải. Hơn nữa, các thuật toán này phải giải quyết tốt vấn đề cân bằng tiêu thụ năng lượng để kéo dài tối đa thời gian hoạt động của mạng và càng có nhiều nút hết năng lượng cùng lúc càng tốt.

**Vấn đề tiết kiệm năng lượng:** Giới hạn về tài nguyên của nút cảm biến là kết quả trong việc hạn chế thời gian sống của mạng mà đa số các ứng dụng mạng cảm biến đều mong muốn mạng có thời gian sống càng lâu càng tốt. Thời gian sống của mạng được định nghĩa như là khoảng thời gian từ khi mạng bắt đầu hoạt động cho đến khi trong mạng có 50% nút đầu tiên cạn kiệt năng lượng, hoặc nút cuối cùng trong mạng tiêu hao hết năng lượng [7, 29, 46, 78]. Ngay khi nút cuối cùng trong mạng chết, mạng cảm biến sẽ ngừng hoạt động vì BS đã mất hết tất cả kết nối đến các nút cảm biến. Trong trường hợp lý tưởng, người ta mong muốn tất cả các nút mạng cùng hết năng lượng ở cùng một thời điểm. Do đó, tiết kiệm năng lượng là một trong những vấn đề quan trọng nhất và cũng là thách thức lớn nhất trong mạng cảm biến không dây.

**An ninh mạng cảm biến không dây:** Trong mạng không dây có nhiều nguy cơ tiềm ẩn tấn công hơn trong mạng có dây, vì thế, các ứng dụng yêu cầu cung cấp cơ chế bảo mật thông tin đảm bảo an toàn cho việc trao đổi dữ liệu để chống lại mã độc tấn công. Mã hóa dữ liệu và chứng thực là các kỹ thuật chung được sử dụng trong mạng cảm biến không dây [19, 89]. Tuy nhiên, giải pháp này sẽ tiêu tốn nhiều năng lượng của các nút mạng do các nút cảm biến phải truyền thông và xử lý dữ liệu nhiều hơn. Có nhiều phương pháp tấn công vào mạng cảm biến không dây. Các phương pháp tấn công thường được chia ra thành phương pháp tấn công chủ động hoặc bị động,

phương pháp tấn công bên trong hoặc bên ngoài như: Tấn công từ chối dịch vụ, tấn công vào lưu lượng dữ liệu (tấn công gây nhiễu - Jamming attack, tấn công lỗ sâu - worm hole attack, tấn công phù thủy - Sybil attack, v.v), hay tấn công vào các giao thức định tuyến, v.v. [82, 98].

Trong các phương pháp tấn công lưu lượng dữ liệu, điển hình là phương pháp tấn công gây nhiễu (Jamming Attack) [82]. Theo phương pháp này, nút độc hại sẽ lợi dụng giao thức truy nhập đường truyền ở tầng MAC (Medium Access Control) như giao thức cảm nhận sóng mang tránh đụng độ CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), giao thức SMAC (Sensor MAC), v.v. Từ đó, nút độc hại sẽ sử dụng một thiết bị phần cứng đặc biệt để phát ra tín hiệu gây nhiễu đến tất cả các nút trong mạng. Khi đó tất cả các nút trong mạng sẽ cho rằng kênh truyền đang bận và không kết nối để truyền dữ liệu đến nút khác. Các giải pháp ngăn chặn phương pháp tấn công gây nhiễu như: Sử dụng các kỹ thuật phát hiện tấn công, các kỹ thuật mã hóa gói dữ liệu ở lớp liên kết để ngăn chặn thiết bị độc hại thu thập thông tin về hệ thống, kỹ thuật trải phổ nhảy tần - FHSS (Frequency Hopping Spread Spectrum), trải phổ trực tiếp - DSSS (Direct Sequence Spread Spectrum), hoặc thay đổi ngẫu nhiên khoảng thời gian truyền dữ liệu trong các giao thức MAC [82, 59].

Trong phương pháp tấn công vào giao thức định tuyến, điển hình là phương pháp tấn công lỗ đen (Black hole Attack). Tư tưởng của phương pháp tấn công lỗ đen là giả mạo các gói tin điều khiển được giao thức định tuyến sử dụng trong quá trình thiết lập tuyến, từ đó thu hút lưu lượng mạng gửi về phía nút tấn công. Ví dụ, trong giao thức AODV, nút độc hại sẽ chờ đợi thông điệp yêu cầu tuyến RREQ từ các nút hàng xóm trong mạng gửi đến nó để bắt đầu cuộc tấn công. Khi nút độc hại nhận được thông điệp RREQ, nó sẽ gửi lại thông điệp RREP giả mạo có tuyến đường đến đích tốt nhất (tuyến có số tuần tự cao nhất và số chặng nhỏ nhất) ngược hướng đến nút nguồn để các nút nguồn chọn tuyến đi qua nút độc hại truyền dữ liệu. Hay trong giao thức LEACH [44, 96], nút độc hại sẽ giả mạo nó là nút cụm trưởng (CH - cluster head) và quảng bá gói tin HEAD\_Adv\_Msg đến các nút khác trong mạng với cường độ mạnh, từ đó thu hút được nhiều nút thành viên chọn nó làm nút



chuyển tiếp dữ liệu đến BS, do đó gây lãng phí năng lượng. Hơn nữa, tác hại của tấn công lỗ đen sẽ tăng lên rất nhiều nếu trong mạng tồn tại nhiều nút độc hại cùng tấn công lỗ đen. Để khắc phục phương pháp tấn công này, Anfeng Liu [66] đã đề xuất phương pháp định tuyến đa đường an toàn hiệu quả năng lượng trong mạng cảm biến không dây (Secure and Energy-Efficient Disjoint Multi-Path Routing for WSNs). Theo đề xuất, nút nguồn sẽ chia toàn bộ gói dữ liệu ra thành M mảnh nhỏ, mỗi mảnh gửi trên một tuyến đường khác nhau đến BS, do đó giảm xác suất chặn được toàn bộ gói tin của nút độc hại. Nhóm tác giả cũng chứng minh được xác suất nút độc hại có thể thu được tất cả các mảnh của một gói tin là rất thấp vì vùng phủ sóng của nút độc hại là rất nhỏ so với vùng mạng.

**Vấn đề đảm bảo chất lượng dịch vụ (QoS):** Trong một số ứng dụng, mạng cảm biến được yêu cầu hệ thống đáp ứng chất lượng dịch vụ tốt như: Thông lượng cao, Độ trễ truyền thông và tỉ lệ lỗi thấp, ví dụ như hệ thống mạng cảm biến được sử dụng để giám sát hiện trường, có truyền thông đa phương tiện (âm thanh, hình ảnh động v.v.). Bên cạnh đó, nhiều ứng dụng đòi hỏi hệ thống cần giữ gìn nguồn năng lượng vì chúng liên quan trực tiếp đến thời gian sống của mạng. Vì thế, thiết kế các giao thức định tuyến nhận biết về năng lượng là vấn đề quan trọng hơn trong hệ thống thông tin cảm biến [19, 81].

## 1.2. Tình hình nghiên cứu trên thế giới

Ngày nay, các hướng nghiên cứu chính về mạng cảm biến không dây chủ yếu tập trung vào bài toán "tiết kiệm năng lượng" để kéo dài thời gian sống cho mạng vì các ứng dụng mạng cảm biến thường được triển khai ở vùng đồi núi, khu vực rừng rậm, núi lửa hoặc dưới nước biển sâu. Các nút mạng sau khi đã được triển khai thường rất khó để sạc lại nguồn pin hoặc thay thế pin mới. Do vậy, bài toán "tiết kiệm năng lượng" hoặc sử dụng hiệu quả nguồn năng lượng pin hạn chế của mạng cảm biến là hướng nghiên cứu chủ yếu và trở thành thách thức lớn nhất cho các nhà khoa học trên thế giới [81]. Những năm qua có nhiều công bố trên thế giới về hướng nghiên cứu này. Điển hình là nhóm tác giả W. Heinzelman đề xuất phương pháp định tuyến phân cụm tiết kiệm năng lượng cho mạng cảm biến không dây [44]. Các nghiên cứu



cải tiến từ tác giả có rất nhiều, điển hình là Joydeep Banerjee cùng các cộng sự đề xuất cải tiến phương pháp chọn nút CH dựa vào khoảng cách [14], hoặc nhóm tác giả Tri Gia Nguyen đề xuất cải tiến phương pháp chọn nút CH dựa vào khoảng cách và năng lượng còn lại [77] hoặc [57, 95].

Thêm nữa, hướng nghiên cứu về "Tổng hợp dữ liệu" (Data fusion) cũng là một trong các hướng đóng vai trò quan trọng trong việc triển khai các ứng dụng mạng cảm biến không dây vì các nút mạng cảm biến thường được triển khai với mật độ dày đặc để đo, cảm biến, giám sát môi trường, v.v. các nút sau khi được triển khai thường đưa ra nhiều kết quả giống nhau. Do đó, tổng hợp dữ liệu sẽ giúp các nút giảm chi phí truyền thông dữ liệu trùng lặp, đưa ra dữ liệu chính xác hơn từ các nút cảm biến khác nhau gửi đến BS. Các công trình công bố về lĩnh vực này như của tác giả S. Sandeep Pradhan và các cộng sự đề xuất sử dụng mã nén phân tán để thực hiện nén hai nguồn dữ liệu tương quan với nhau [84]. Các đề xuất tiếp theo là nhóm tác giả Zhibin Sun cùng các cộng sự [99] hoặc nhóm J.E. Barceló-Lladó cùng các cộng sự nghiên cứu sử dụng mã nguồn phân tán để xóa bỏ thông tin tương quan dư thừa, trùng lặp trong mạng cảm biến không dây được triển khai dày đặc [16, 24].

Nhiều tạp chí nổi tiếng thế giới chuyên công bố các kết quả nghiên cứu về mạng cảm biến không dây, thí dụ tạp chí "IEEE Sensors Journal" công bố bài báo của tác giả Chi-Tsun Cheng cùng các cộng sự trình bày về "A Clustering Algorithm for Wireless Sensor Networks Based on Social Insect Colonies". Nội dung bài báo đề xuất thuật toán phân cụm phân tán dựa trên cấu trúc của đàn côn trùng để mở rộng thời gian sống mạng cảm biến [25]. Tạp chí "Sensors Open Access Journal" công bố bài báo của nhóm tác giả Ming Liu đề xuất giao thức định tuyến mới nhận thức về năng lượng "An Energy-Aware Routing Protocol in Wireless Sensor Networks - EAP" [67]. EAP đạt được hiệu quả năng lượng bằng cách giảm tối thiểu năng lượng tiêu thụ và cân bằng tải tiêu thụ năng lượng giữa các nút dựa trên kỹ thuật phân cụm.

Gần đây, Khamiss, A.A cùng các cộng sự [52] đề xuất cải tiến thuật toán định tuyến phân cụm tập trung LEACH-C để kéo dài thời gian sống cho mạng. Theo đề xuất này, toàn bộ mạng được chia thành các cụm dựa vào hàm mật độ phân bố của nút

trên vùng cảm biến để giảm khoảng cách truyền thông giữa các nút trong cụm và chọn CH có xem xét đến năng lượng còn lại và khoảng cách truyền thông của nút ứng viên đến BS. Thêm nữa, Hnin Yu Shwe và các cộng sự [95] đề xuất giao thức định tuyến tiết kiệm năng lượng dựa trên phân cụm ứng dụng cho tòa nhà thông minh gọi là "Cluster-based WSN Routing Protocol for Smart Buildings". Đầu tiên, mỗi nút trong mạng được gán một số định danh ID. Tiếp theo, các nút quảng bá thông điệp chứa số định danh này tới các nút hàng xóm. Để quyết định nút có là CH hay không, nút so sánh số định danh của nó với số định danh của tất cả các nút hàng xóm gửi đến. Nếu ID của nó lớn hơn tất cả các ID khác, nút sẽ trở thành CH và quảng bá thông điệp CH đến các nút khác. Các nút khác chọn CH và gửi thông điệp gia nhập cụm dựa vào chỉ thị độ mạnh của tín hiệu nhận được RSSI [5, 61].

Ngoài ra cũng có nhiều dự án nghiên cứu nâng cao hiệu quả sử dụng năng lượng mạng cảm biến không dây để mở rộng thời gian sống của mạng thông qua các lược đồ định tuyến khác nhau [81]. Điển hình là các đề án cải tiến dựa vào lược đồ phân cụm như LEACH, cải tiến chuỗi dài hoặc cải tiến dựa trên mô hình cây, v.v.

### 1.3. Tình hình nghiên cứu ở Việt Nam

Việc nghiên cứu và triển khai ứng dụng hệ thống mạng cảm biến không dây phục vụ cuộc sống phát triển kinh tế ở Việt Nam trong những năm qua còn nhiều khiếm tốn do các yếu tố như: Thiếu trang thiết bị, phòng thí nghiệm để nghiên cứu; thiếu các giáo sư đầu ngành hướng dẫn có trình độ chuyên môn cao; các đề tài nghiên cứu được nghiệm thu xong nhưng không có đủ kinh phí để triển khai ứng dụng trên quy mô lớn. Tuy nhiên, ở Việt Nam cũng đã có một số nghiên cứu viết về mạng cảm biến không dây điển hình như:

- Đại học Công nghệ, ĐHQGHN, PGS.TS. Vương Đạo Vy và Nguyễn Quốc Đạt (2007) "Định tuyến theo giá trị tối thiểu trên nút mạng cảm biến không dây và một số thử nghiệm", Tạp chí Tự động hóa ngày nay, 11(87), pp. 41-43. Đề xuất giao thức định tuyến dựa trên khoảng cách tối thiểu trong mạng cảm biến không dây cho hiệu năng tốt khi so sánh với thuật toán đã có. Tuy

nhiên đề xuất mới dừng lại ở việc tính toán thử nghiệm, chưa thực nghiệm trên các công cụ mô phỏng tin cậy.

- Phòng thí nghiệm những bộ môn Truyền dữ liệu và Mạng máy tính, khoa CNTT trường Đại học Công nghệ, Đại học Quốc gia Hà nội, Vũ Tiến Thái, Nguyễn Đình Việt, Nguyễn Minh Hiền, (2013). "An Energy-Aware Routing Protocol for Wireless Sensor Networks based on K-Means Clustering", *Recent Advances in Electrical Engineering and Related Sciences (AETA)*, pp 297-306. Đề xuất cải tiến giao thức định tuyến phân cụm LEACH [44] dựa trên giải thuật K-Means để kéo dài thời gian sống cho mạng cảm biến.

Luận án tiến sĩ:

- Lê Đình Thanh, (2014). "Hỗ Trợ định vị và nâng cao hiệu năng định tuyến dựa trên thông tin vị trí cho các mạng cảm biến không dây", ĐHCN-ĐHQG HN, người hướng dẫn khoa học: PGS.TS. Hồ Thuần, TS. Nguyễn Đại Thọ. Tác giả luận án đã có những đề xuất cải tiến hiệu năng định tuyến dựa trên thông tin vị trí như: Đề xuất một thuật toán tối ưu hóa đường đi bằng cách tạo các đường tắt và áp dụng chuyển tiếp tham lam nhằm rút ngắn các đường đi và tránh cực tiểu địa phương ở các nút biên. Đề xuất thuật toán định tuyến dựa trên thông tin vị trí sử dụng hình thức cạnh tranh kết hợp. Các đề xuất được công bố trên các tạp chí, hội thảo trong nước và quốc tế.
- Nguyễn Trung Dũng, (2014), "Nghiên cứu phát triển định tuyến tiết kiệm năng lượng cho mạng cảm biến không dây", ĐHBK HN, người hướng dẫn khoa học: PGS.TS. Nguyễn Văn Đức. Luận án này cũng đạt được các kết quả nhất định như: Đề xuất giao thức định tuyến tiết kiệm năng lượng bằng cách giảm gói tin điều khiển trong quá trình khám phá tuyến dựa trên AODV, đề xuất cải tiến giao thức định tuyến dựa trên thông số năng lượng nhằm chọn tuyến có mức năng lượng cao từ đó cân bằng được năng lượng trong mạng. Các đề xuất cũng được công bố trên các tạp trí và hội thảo quốc tế nhưng tác giả Nguyễn Trung Dũng không phải là tác giả chính.
- Vũ Chiến Thắng, (2014), "Nghiên cứu xây dựng thuật toán định tuyến trên cơ sở nhận thức về năng lượng cho mạng cảm biến không dây", Viện nghiên

cứ Điện tử-Tin học, người hướng dẫn khoa học: PGS.TS. Nguyễn Chấn Hùng, PGS.TS. Lê Nhật Thăng. Luận án có đề xuất cải tiến giao thức định tuyến cây thu thập dữ liệu có nhận thức về năng lượng dựa trên hai thông số là năng lượng còn lại của mỗi nút và trạng thái ổn định tuyến. Triển khai thực nghiệm cho 10 nút cảm biến tích hợp các chức năng định tuyến mới và đánh giá hiệu năng của chúng trong điều kiện thực tế.

Các luận văn thạc sĩ:

- Hồ Đức Ái, (2011). "Nghiên cứu kiến trúc hệ thống tiêu thụ ít năng lượng cho mạng cảm biến".
- Đào Trọng Biên, (2013), "Giao thức MAC sử dụng năng lượng hiệu quả và thích ứng cho mạng cảm biến không dây".
- Nguyễn Thị Loan, (2014). "Đánh giá hiệu suất của các giao thức định tuyến mạng cảm biến không dây".
- Nguyễn Hoài Nam (2015). "Mạng cảm biến không dây ứng dụng chăm sóc sức khỏe con người", người hướng dẫn khoa học: PGS. TS. Nguyễn Đình Việt.

Các luận văn thạc sĩ trên mới dừng lại ở việc phân tích, mô phỏng để đánh giá các công bố đã có trên thế giới, chưa có được các đóng góp riêng cho các vấn đề mà tác giả đã nghiên cứu.

#### **1.4. Mục tiêu nghiên cứu của luận án và các vấn đề được giải quyết**

##### **1.4.1. Các giả thiết**

Không mất tính tổng quát, các giả thiết sau đây nên được thỏa mãn để xác định rõ yêu cầu bài toán:

- Mạng cảm biến không dây với số lượng lớn nút cảm biến được triển khai ngẫu nhiên, dày đặc trên vùng rộng lớn, phẳng được gọi là 2D.
- Các kịch bản triển khai ứng dụng thực tế cho giả thiết này là mạng cảm biến không dây được triển khai để đo nhiệt độ, độ ẩm, theo dõi cháy rừng, điều khiển tưới tiêu tự động, hay theo dõi sự xâm nhập trái phép của quân địch trên một bãi chiến trường, v.v.

- Các nút phát sóng ra-đi-ô đẳng hướng, các liên kết được giả thiết là hai chiều sử dụng cùng loại với ăng-ten thu phát.
- Bộ phát sóng có khả năng điều chỉnh công suất phát để có thể truyền thông với các nút ở xa.
- Các nút mạng có khả năng xử lý và có dung lượng bộ nhớ như nhau nhưng dung lượng pin có thể khác nhau và sau khi được triển khai, chúng rất khó có thể được sạc hoặc thay thế mới.
- Nút trung chuyển BS hoặc sink được đặt ở gần hoặc xa mạng và các nút mạng sau khi được triển khai ít hoặc không di chuyển hay mạng được xem là tĩnh. Giả thiết này hoàn toàn hợp lý vì hầu hết các mạng cảm biến sau khi được triển khai trong thực tế đều ở trạng thái tĩnh.

#### **1.4.2. Các mục tiêu cụ thể**

Trong luận án này, các nội dung nghiên cứu thuộc bài toán định tuyến phân cụm tiết kiệm năng lượng nêu trên được quan tâm giải quyết:

1. Tổng hợp, hệ thống hóa các kết quả nghiên cứu liên quan và phân tích các ưu điểm và nhược điểm của chúng.
2. Nghiên cứu, đề xuất cải tiến một thuật toán định tuyến phân cụm truyền đơn chặng dựa trên tính toán mức năng lượng còn lại trung bình của các nút còn sống và xem xét đến khoảng cách từ nút ứng viên đến BS cho hiệu quả sử dụng năng lượng tốt.
3. Nghiên cứu, đề xuất cải tiến thuật toán định tuyến dựa trên chuỗi, kết hợp với việc tổng hợp, nén dữ liệu ở các nút dọc theo chuỗi cho hiệu quả sử dụng năng lượng và có thể ứng dụng trên các mạng cảm biến không đồng nhất, nơi mà các nút có dung lượng pin không giống nhau.
4. Nghiên cứu, đề xuất cải tiến giao thức định tuyến phân cụm dựa trên cây tối thiểu kết hợp với tổng hợp, nén dữ liệu theo mô hình cây và lập lịch ngủ cho hiệu quả sử dụng năng lượng cao.

#### **1.5. Nội dung luận án**

Luận án được trình bày trong năm chương với các nội dung chính như sau:

- *Chương 1* đặt vấn đề; phát biểu bài toán và mục tiêu của luận án, tóm tắt nội dung và những đóng góp chính của luận án.

- *Chương 2* trình bày kết quả nghiên cứu tổng quan về định tuyến có thứ bậc và định tuyến phân cụm tiết kiệm năng lượng. Các bài toán định tuyến phẳng, định tuyến theo thông tin vị trí địa lý hay định tuyến theo chất lượng dịch vụ cũng được khảo sát một cách tóm tắt ở chương này. Chương 2 kết thúc với một số nhận xét về các giải pháp đã được đề xuất.

- *Chương 3* trình bày đề xuất thuật toán cải tiến giao thức định tuyến phân cụm phân tán dựa trên việc xem xét đến năng lượng còn lại trung bình và khoảng cách từ nút ứng viên đến nút BS trước khi chọn làm CH. Nội dung chương gồm, phân tích các hạn chế của giao thức đã có, đề xuất thuật toán cải tiến của chúng tôi, đánh giá và so sánh hiệu quả năng lượng của thuật toán được đề xuất với những thuật toán đã có, cùng với hướng phát triển được trình bày.

- *Chương 4* trình bày hai thuật toán do chúng tôi đề xuất cải tiến nhằm nâng cao hiệu quả tiêu thụ năng lượng trong định tuyến phân cụm dựa trên xây dựng chuỗi kết hợp với tổng hợp, nén dữ liệu. Kỹ thuật này giúp nâng cao hiệu quả sử dụng năng lượng của thuật toán định tuyến chuỗi một cách rõ rệt và được kiểm chứng qua mô phỏng. Mục đích thiết kế thuật toán, nội dung thuật toán, đánh giá hiệu năng của thuật toán và hướng nghiên cứu cải tiến thuật toán lần lượt được trình bày.

- *Chương 5* trình bày hai thuật toán định tuyến dựa trên xây dựng cây tối thiểu được chúng tôi đề xuất cải tiến nhằm nâng cao hiệu quả sử dụng năng lượng, kết hợp với tổng hợp dữ liệu hay lập lịch ngủ cho các nút có hiệu quả rõ rệt thông qua các kết quả mô phỏng. Mục đích thiết kế thuật toán, nội dung thuật toán, đánh giá hiệu năng của thuật toán, hướng nghiên cứu cải tiến thuật toán lần lượt được trình bày.

- Kết thúc luận án là phần kết luận, tổng kết các kết quả đạt được của luận án và giới thiệu một số hướng nghiên cứu mở rộng tiếp theo.

Ngoài nội dung chính được trình bày trong nội dung các chương kể trên, phần tài liệu tham khảo hoặc cơ sở toán học được sử dụng trong luận án được trình bày trong hai phụ lục:

- *Phụ lục 1* trình bày lý thuyết hiển nhiên Dempster-Shafer miêu tả sự không chắc chắn về tính ngẫu nhiên và không chính xác, thông qua độ đo niềm tin và sự hợp lý.
- *Phụ lục 2* trình bày cơ sở toán học về mã nguồn phân tán DSC.

## **1.6. Đóng góp của luận án**

### **Những đóng góp chính của luận án bao gồm:**

1. Đề xuất cải tiến giao thức định tuyến phân cụm phân tán dựa trên tiêu chuẩn chọn nút cụm trưởng và các nút thành viên gia nhập cụm, thuật toán có thể làm việc tốt trên các mạng cảm biến đồng nhất. Theo thuật toán, mỗi nút ứng viên trước khi được chọn làm nút cụm trưởng ở vòng thứ  $i$  phải có năng lượng còn lại lớn hơn mức năng lượng trung bình của tất cả các nút còn sống và xem xét đến khoảng cách từ nó đến nút BS. Các nút khác sẽ chọn nút CH để gia nhập cụm dựa vào khoảng cách từ nó đến CH và từ CH đến BS, theo biểu thức (3.4). Các kết quả mô phỏng cho thấy thuật toán được đề xuất có thể kéo dài thời gian sống của mạng thêm khoảng 20%.
2. Đề xuất cải tiến thuật toán xây dựng chuỗi kết hợp với tổng hợp, nén dữ liệu cho hiệu quả năng lượng tốt, được gọi là DFCB (Data Fusion and Chain-based Routing Protocol). Các nút trong mạng được kết nối với nhau thành một chuỗi dài để giảm khoảng cách truyền thông giữa các nút, tiếp theo đó, dữ liệu trước khi truyền được các nút tổng hợp, nén không mất dọc theo chuỗi dựa vào mã nguồn phân tán để giảm chi phí truyền thông. Các kết quả mô phỏng của chúng tôi cho thấy DFCB có thể kéo dài thời gian sống của mạng thêm khoảng 40% và 10% khi so sánh với LEACH và PEGASIS.
3. Đề xuất cải tiến một lược đồ xây dựng cung chuỗi có tên SCBC (Sector-Chain Based Clustering Routing Protocol) cho mạng cảm biến không dây. Theo lược đồ này, toàn bộ vùng mạng (vùng cảm biến mạng bao phủ) sẽ được phân chia thành các cung (cụm) lô-gíc với số nút được phân bố tương đối đều nhau dựa vào góc tạo bởi nút và BS cùng với hệ trục tọa độ OXY. Sau khi chia cụm, các cụm sẽ chọn nút CH hoặc nút cụm trưởng thứ hai (SCH), nút ở gần BS, chịu trách nhiệm chuyển tiếp dữ liệu đến BS. Do đó,

SCBC có thể sử dụng hiệu quả nguồn năng lượng quý hiếm ở các nút cảm biến không dây và kéo dài thời gian hoạt động mạng. Các kết quả mô phỏng của chúng tôi cho thấy SCBC có thể kéo dài thời gian sống của mạng thêm khoảng 20% và 15% khi so sánh với PEGASIS và IEEPB.

4. Đề xuất cải tiến thuật toán định tuyến hiệu quả năng lượng dựa trên xây dựng cây khung nhỏ nhất kết hợp với tổng hợp, nén dữ liệu, có tên là DFTBC (Data Fusion and Tree-Based Clustering Routing Protocol). Tư tưởng của thuật toán là chia toàn bộ vùng cảm biến mạng thành các cụm, (giả thiết số cụm là 5%), sau đó trong mỗi vòng, coi các nút còn sống của mỗi cụm như một đồ thị vô hướng  $G(V, E)$ , xây dựng cây khung có khoảng cách của các nút còn sống trong cụm là nhỏ nhất kết hợp với tổng hợp, nén dữ liệu trên cây. Các kết quả mô phỏng cho thấy DFTBC cho hiệu quả sử dụng năng lượng tốt hơn các giao thức LEACH-C, PEGASIS và STDC đã có tương ứng khoảng 60%, 20% và 8%.
5. Đề xuất cải tiến thuật toán định tuyến hiệu quả năng lượng dựa trên xây dựng cây khung nhỏ nhất kết hợp với lập lịch ngủ có tên SSTBC (Sleep Scheduled and Tree-Based Clustering Routing Protocol) cho mạng cảm biến không dây. Tư tưởng của thuật toán là chia toàn bộ vùng cảm biến mạng thành các cụm, mỗi cụm lại được chia thành các ô lưới ảo, mỗi ô lưới ảo có nhiều nút, kích thước của lưới do người dùng định nghĩa. Mỗi ô chỉ có một nút có mức năng lượng cao, ở chế độ thức (làm việc), các nút khác sẽ đi vào chế độ ngủ để tiết kiệm năng lượng, sau mỗi vòng lại đổi vai trò cho nhau. Các kết quả mô phỏng cho thấy SSTBC giúp kéo dài thời gian sống của mạng khoảng 65% so với LEACH-C, khoảng 20% so với PEGASIS và khoảng 12% so với STDC.



## **Chương 2: ĐỊNH TUYẾN VÀ ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG TRONG MẠNG CẢM BIẾN KHÔNG DÂY**

Chương này trình bày tổng quan về tiết kiệm năng lượng và định tuyến tiết kiệm năng lượng dựa trên phân cụm trong mạng cảm biến không dây, tóm tắt nội dung và đánh giá các giao thức đã có nhằm làm cơ sở cho các đề xuất được trình bày ở các chương tiếp theo. Những nội dung chính sau được trình bày trong chương này:

- Các giải pháp tiết kiệm năng lượng
- Định tuyến trong mạng cảm biến không dây
- Định tuyến tiết kiệm năng lượng
- Nhận xét về các giao thức đã có

### **2.1. Giải pháp tiết kiệm năng lượng trong mạng cảm biến không dây**

#### **2.1.1. Giải pháp tiết kiệm năng lượng trong kiến trúc nút cảm biến**

Tư tưởng của giải pháp này là thay đổi cấu trúc thiết kế phần cứng cho nút cảm biến để tiết kiệm năng lượng bằng cách sử dụng các công nghệ tiết kiệm năng lượng, tối ưu hóa việc sử dụng năng lượng, thiết kế nhiều chế độ hoạt động với mức năng lượng tiêu tốn khác nhau. Đây là giải pháp trực tiếp để tiết kiệm năng lượng ngay trên từng nút cảm biến không dây, hoặc chúng ta cũng có thể lấy năng lượng trực tiếp từ môi trường bên ngoài để có thể đảm bảo cho việc hoạt động của các nút mạng cảm biến, năng lượng này gọi là năng lượng sạch và bao gồm quang năng, động năng, năng lượng mặt trời và các nguồn năng lượng khác. Nút mạng có thể tận dụng nguồn năng lượng này để thay thế hoặc kết hợp với nguồn năng lượng pin được sử dụng bởi chính nó. Điển hình của giải pháp này là đề xuất của Hempstead [45] một kiến trúc hệ thống năng lượng thấp cho các ứng dụng mạng cảm biến. Theo đề xuất này, kiến trúc vi điều khiển của nút cảm biến được chia thành hai phần gồm: Bộ vi điều khiển (Controller) dùng để xử lý các sự kiện tổng thể như khởi động hệ thống và lập trình lại. Bộ xử lý sự kiện (Event Processor) xử lý các sự kiện thông thường được ví như máy trạng thái để giảm tối đa nguồn năng lượng cung

cấp cho Bộ vi điều khiển hoạt động, từ đó tiết kiệm năng lượng. Thêm nữa, kiến trúc hệ thống thực hiện tắt các thành phần mạch điện không cần thiết trong quá trình hoạt động để giảm tiêu thụ năng lượng, ví dụ như tắt mạch điện của các băng nhớ không lưu dữ liệu, hay tắt bộ thu/phát sóng ra-đi-ô sau khi đã hoàn thành công việc, v.v.

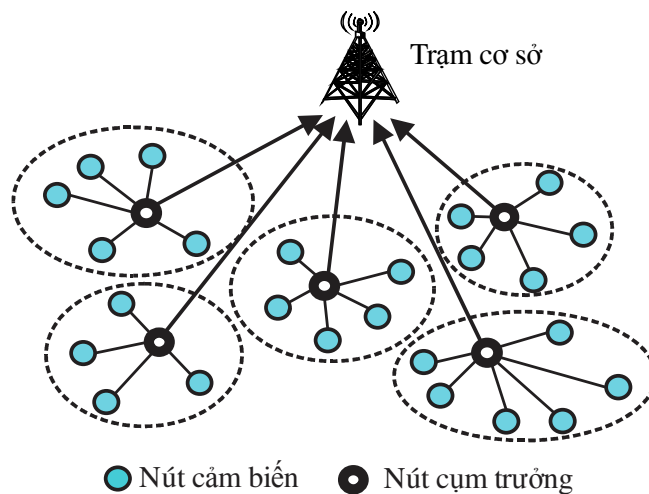
### **2.1.2. Giải pháp tiết kiệm năng lượng trong điều khiển truy nhập môi trường truyền dẫn không dây**

Theo kiến trúc phân tầng, ta có thể thay đổi các giao thức ở tầng liên kết dữ liệu hay tầng mạng cho mạng cảm biến và có chiến lược bảo tồn nguồn năng lượng. Ta có thể duy trì nguồn năng lượng bằng cách sử dụng các chế độ hoạt động tiết kiệm năng lượng và dùng giới hạn thời gian trong các bản tin báo nhận. Ví dụ, các thiết bị không dây nên được tắt đi trong suốt thời gian nhàn rỗi. Ta có thể cải tiến các giao thức điều khiển truy nhập môi trường (Medium Access Control) như CSMA/CA, S-MAC (Sensor MAC), B-MAC (Berkeley-MAC) hoặc WiseMAC để nâng cao sử dụng nguồn năng lượng một cách hiệu quả, kéo dài thời gian sống cho mạng [4, 96]. Điển hình là giao thức T-MAC (Time out MAC) [28], một cải tiến từ giao thức S-MAC. T-MAC kế thừa được các ưu điểm từ S-MAC và T-MAC đạt được hiệu quả sử dụng năng lượng tốt hơn S-MAC bằng cách giảm thời gian nghe kênh truyền rỗi (Idle Listening) và xác định mềm dẻo khoảng thời gian hoạt động trong mỗi chu kỳ thức/ngủ (Active/Sleep). Tuy nhiên, T-MAC khó tránh khỏi việc lãng phí năng lượng do nghe thừa khi muốn mạng đạt đến thông lượng cực đại. Để khắc phục nhược điểm này, Zhenzhou Tang và các cộng sự [103] đã đề xuất giao thức ALLEE-MAC (An Adaptive Low Latency and Energy Efficient MAC Protocol) giao thức MAC hiệu quả năng lượng và độ trễ thấp thích ứng. ALLEE-MAC được thiết kế với hai lược đồ chính là lược đồ truyền dữ liệu liên tục và lược đồ ngủ sớm. Lược đồ truyền dữ liệu liên tục cho phép các nút gửi gói dữ liệu liên tục (gửi nhanh) khi hàng đợi tại giao diện đủ lớn, trong khi đó, lược đồ ngủ sớm giảm năng lượng tiêu thụ của nút bằng cách cho nút vào chế độ ngủ ngay sau khi hoàn thành việc truyền dữ liệu hay nó nhận thấy không có dữ liệu để truyền và không có nút nào truyền dữ liệu đến nó. Thêm nữa, Lei Tang cùng các cộng sự [102] đã đề xuất giao

thức dự đoán thức giấc cho hiệu quả năng lượng PW-MAC (An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks) dựa trên lược đồ không đồng bộ chu kỳ ngủ/hoạt động. Theo lược đồ này, thời gian hoạt động của nút được chia ra thành các chu kỳ không đồng bộ, mỗi chu kỳ chứa đựng hai khoảng thời gian là khoảng thời gian ngủ và khoảng thời gian hoạt động. PW-MAC tối thiểu thời gian hoạt động của nút dựa trên việc dự đoán chính xác thời điểm nút nhận hoạt động để truyền dữ liệu tin cậy. Sau khi hoàn thành truyền dữ liệu, cả hai nút sẽ nhanh chóng đi vào chế độ ngủ để tiết kiệm năng lượng, kéo dài thời gian sống của mạng.

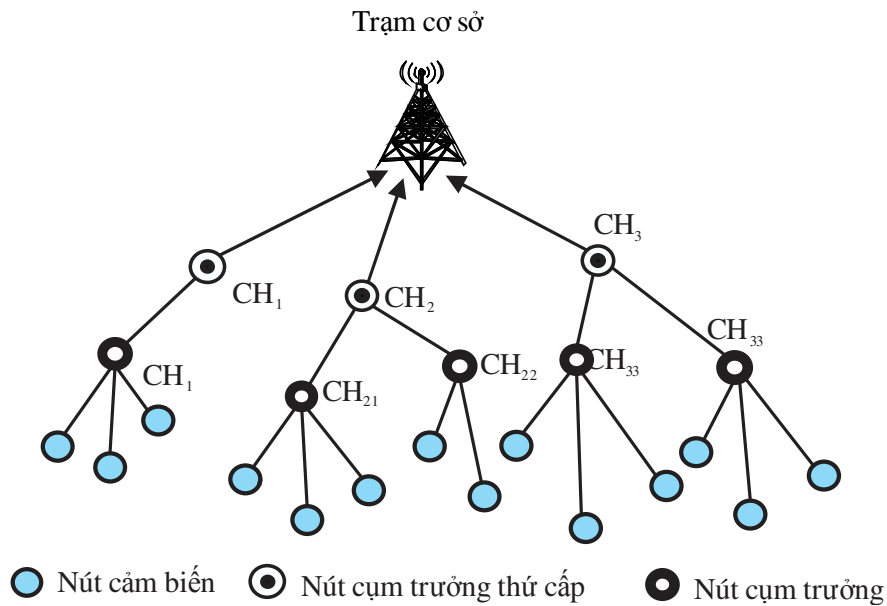
### 2.1.3. Giải pháp tổng hợp dữ liệu

Trong mạng cảm biến không dây, các nút cảm biến quan sát môi trường và định kỳ gửi gói dữ liệu quan sát được tới BS, BS được kết nối với người dùng thông qua Internet. Tổng hợp dữ liệu là quá trình xử lý dữ liệu từ nhiều nguồn khác nhau để loại bỏ dữ liệu dư thừa (trùng lặp), nâng cao chất lượng nguồn dữ liệu đo được, giảm số bit cần truyền, và do đó giảm năng lượng tiêu thụ, mở rộng thời gian sống mạng WSN. Tổng hợp dữ liệu có thể được thực hiện ở các nút BS, CH hoặc ở ngay các nút cảm biến trước khi chúng gửi dữ liệu ra kênh truyền theo mô hình mạng có cấu trúc phân bậc, ví dụ như mô hình phân cụm (Hình 2.1), hoặc theo cấu trúc cây (Hình 2.2) hay theo mô hình chuỗi (Hình 2.3 trang 38) [69].



**Hình 2.1:** Tổng hợp dữ liệu theo mô hình phân cụm đơn chặng

Hình 2.1 minh họa quá trình tổng hợp dữ liệu ở nút CH trong các cụm. Theo lược đồ này, toàn bộ mạng cảm biến được chia thành các cụm khác nhau, mỗi cụm bầu ra một nút cụm trưởng gọi là CH. CH chịu trách nhiệm thu nhận thông tin cảm biến được từ các nút thành viên trong cụm sau đó thực hiện tổng hợp chúng với dữ liệu nó cảm biến được rồi chuyển tiếp đơn chặng (trực tiếp) đến BS, các nút khác (nút thành viên cụm) không gửi dữ liệu đến BS mà gửi trực tiếp tới nút CH của nó tương ứng [69, 106].



**Hình 2.2:** Tổng hợp dữ liệu theo mô hình phân cụm đa chặng

Hình 2.2 minh họa quá trình tổng hợp dữ liệu theo cấu trúc hình cây nhiều tầng, nhiều CH, các nút cảm biến gửi dữ liệu đến nút cụm trưởng (cha) trực tiếp của nó, các nút CH tổng hợp dữ liệu rồi lại gửi đến nút cụm trưởng ở mức cao hơn, v.v. cứ như vậy cho đến khi BS nhận được dữ liệu của toàn bộ các nút trong mạng.

## 2.2. Định tuyến trong mạng cảm biến không dây

Định tuyến là quá trình tìm kiếm đường đi cho các gói tin từ nút nguồn đến nút đích, định tuyến trong mạng cảm biến không dây là quá trình tìm kiếm đường đi cho tất cả các gói tin, chúng được gửi từ tất cả các nút cảm biến trong mạng theo định kỳ đến nút BS (nút nguồn là tất cả các nút cảm biến trong mạng, nút đích là BS hoặc sink). Một giao thức định tuyến tốt phải có khả năng cân bằng giữa sự đáp ứng

về thông lượng, các yêu cầu chất lượng dịch vụ của mạng với hiệu quả sử dụng năng lượng. Để đạt được sự cân bằng này thì khả năng tính toán và truyền thông phải được xem xét sao cho phù hợp. Thông thường, mỗi giao thức định tuyến hoạt động dựa trên các nguyên tắc khác nhau và nó chỉ hiệu quả với một số hoàn cảnh, điều kiện, tô-pô hay tiêu chuẩn nhất định [96].

### **2.2.1. Phân loại các giao thức định tuyến trong mạng cảm biến không dây**

Có nhiều cách để phân loại các giao thức định tuyến trong mạng cảm biến không dây, một trong các cách phân loại được nhiều người sử dụng phân chia chúng thành các loại như sau: Các giao thức *kiến trúc phẳng (Flat-Architecture)*, các giao thức *dựa trên thông tin vị trí (Location-based Protocols)*, các giao thức dựa trên *chất lượng dịch vụ (QoS-Based Protocols)* và các giao thức *có thứ bậc (Hierarchical Protocol)* [4, 81].

Trong loại thứ nhất - các giao thức kiến trúc phẳng, tất cả các nút trong mạng có nhiệm vụ và chức năng ngang nhau. Các nút tham gia truyền thông đa chặng (truyền thông qua nút trung gian) sử dụng phương pháp phát tràn để tìm đường như Flooding hoặc gửi ngẫu nhiên gói tin tới một nút hàng xóm lân cận để tìm đường như Gossiping [4, 96], hay trao đổi thông điệp trong phạm vi lân cận gọi là truyền thông trực tiếp DD, hoặc giao thức truyền thông tin thông qua đàm phán SPIN [4, 96], v.v.

Trong loại thứ hai - các giao thức dựa trên thông tin vị trí sử dụng các ưu điểm thông tin vị trí, tọa độ của các nút để chuyển tiếp dữ liệu nhận được trong vùng cục bộ, không phải toàn mạng. Các giao thức thuộc lớp này có thể tìm kiếm tuyến từ nguồn tới đích và tối thiểu năng lượng sử dụng của các nút cảm biến. Chúng có hạn chế trong trường hợp này là các nút phải biết vị trí của các nút khác trong mạng nhờ sử dụng GPS, hoặc thông điệp từ BS gửi như GEAR [37, 81], GEM [81], v.v.

Trong loại thứ ba - các giao thức dựa trên chất lượng dịch vụ quan tâm nhiều đến chất lượng dịch vụ mạng hơn là nâng cao hiệu quả sử dụng năng lượng. Cơ chế định tuyến của các giao thức thuộc loại này được xác định dựa trên một số hiểu biết về tài nguyên trong mạng cũng như yêu cầu về chất lượng dịch vụ (thông lượng, độ trễ,

độ tin cậy, mức độ tiêu thụ tài nguyên, ...) của luồng dữ liệu, ví dụ: Giao thức SAR xây dựng nhiều tuyến đường từ nút nguồn tới đích để tránh quá tải hoặc lỗi tuyến đường từ đó giảm tỉ lệ lỗi gói tin và tăng khả năng đáp ứng tuyến [81], SPEED [43, 56, 96] duy trì tốc độ chuyển phát kết hợp với điều khiển phản hồi và chuyển tiếp địa lý không xác định để giảm độ trễ và tránh nghẽn mạng, từ đó nâng cao chất lượng dịch vụ.

Cuối cùng, nhóm giao thức trong lược đồ có thứ bậc, các giao thức định tuyến đạt được hiệu quả sử dụng năng lượng, sự ổn định và khả năng mở rộng cao. Theo lược đồ này, các nút mạng được tổ chức thành các cụm (Clustering), trong mỗi cụm có một nút được bầu làm nhiệm vụ quản lý, điều hành hoạt động cụm như một cụm trưởng. Nút cụm trưởng chịu trách nhiệm cho các hoạt động bên trong cụm và chuyển tiếp thông tin đến nút cơ sở, gọi là nút trung truyền (BS hoặc sink), kỹ thuật này giảm năng lượng tiêu thụ và kéo dài thời gian sống cho mạng. Chúng có tỉ lệ chuyển phát và sự ổn định cao, và cân bằng về năng lượng tiêu thụ giữa các nút trong mạng, ví dụ giao thức: LEACH [6, 22, 39, 49], LEACH-C [53, 69, 75, 110], HEED [69, 115], PEGASIS [64, 93], TEEN [74], APTEEN [73], v.v.

### 2.2.2. Các giao thức kiến trúc phẳng

Đặc điểm của mạng WSN khác với mạng MANET là số lượng nút cảm biến của mạng sau khi được triển khai thường rất lớn và rất khó có thể đánh số định danh (ID) cụ thể cho từng nút cảm biến. Do vậy các giao thức định tuyến dựa trên địa chỉ định danh là không phù hợp cho mạng WSN. Để khắc phục được điều này, các giao thức định tuyến tập trung vào các thuộc tính đặc trưng cho dữ liệu của mỗi loại ứng dụng cụ thể.

Giao thức phát tràn (Flooding) [4, 81, 96] là kỹ thuật được sử dụng phổ biến nhất để tìm đường và phát tán thông tin trong mạng MANET không dây và có dây. Hoạt động định tuyến này đơn giản, không đòi hỏi chi phí cấu hình mạng tốn kém, không cần các giao thức tìm đường phức tạp v.v. Theo đó khi một nút nhận được gói điều khiển hay gói dữ liệu, nó sẽ gửi đến tất cả các nút lân cận khác trừ nút đã gửi dữ liệu cho nó. Công việc này được tiếp tục cho đến khi tất cả các nút trong mạng đều

nhận được thông điệp và đảm bảo thông điệp sẽ đến được đích của mình. Để ngăn chặn gói tin có thể lưu hành lòng vòng vô thời hạn trong mạng, mỗi gói tin được gắn với một trường "hop count" hoặc thời gian sống (TTL). Nó quy định rõ khoảng thời gian mà một gói tin tồn tại trên mạng, sau khoảng thời gian này, gói tin sẽ bị hủy bỏ. Flooding có ưu điểm là hoạt động đơn giản nhưng khi áp dụng cho mạng WSN nó gặp phải các hạn chế như hiện tượng “bùng nổ lưu lượng” (traffic implosion), hiện tượng này xảy ra do có nhiều bản sao của cùng một gói tin gửi tới cùng một nút mạng. Vấn đề tiếp theo là hiện tượng “chồng chéo dữ liệu” (data overlap), đó là hiện tượng hai hay nhiều nút cảm biến cùng cảm nhận được một sự kiện và các nút cảm biến này cùng gửi thông tin nó cảm nhận được về một nút khác. Cuối cùng là hiện tượng “mù tài nguyên” (resource blindness). Đây là vấn đề nguy hiểm nhất trong phát tràn, giao thức mà phát tràn sử dụng thì không hề quan tâm đến năng lượng của mỗi nút cảm biến, do đó các nút sẽ nhanh chóng bị tiêu hao năng lượng và giảm thời gian sống của mạng. Một cải tiến của Flooding là giao thức Gossiping [4, 81, 96], Gossiping yêu cầu nút nhận được thông điệp gửi nó tới một nút ngẫu nhiên lân cận của nó. Quá trình này tiếp tục cho tới khi gói tin tới được đích hay khi trường "hop count" bằng 0. Điều này tránh được vấn đề “bùng nổ lưu lượng” do giảm số gói tin truyền đi trên mạng. Tuy nhiên, trong nhiều trường hợp thì độ trễ có thể rất cao hoặc thậm chí gói tin có thể không đến được đích.

Giao thức cho thông tin cảm biến thông qua đàm phán (SPIN) [4, 81, 96], SPIN khắc phục các vấn đề mà Flooding gặp phải thông qua việc đàm phán với các nút khác sử dụng các gói tin mô tả dữ liệu trước khi gửi dữ liệu và sự thích ứng của tài nguyên. Đó là, trước khi gửi toàn bộ dữ liệu, các nút sẽ thực hiện vấn đề bắt tay sử dụng ba kiểu thông điệp; kiểu thứ nhất, thông điệp ADV (Advertisement) chứa mô tả dữ liệu dùng để quảng bá gói dữ liệu mà nút muốn phát; kiểu thứ hai, thông điệp REQ (Request) dùng để yêu cầu nút nguồn gửi gói dữ liệu thực đã được quảng bá trước đó; Kiểu thứ ba là thông điệp DATA chứa dữ liệu thực, thông điệp DATA có kích thước lớn hơn gói ADV và REQ. Khi một nút nhận được thông điệp mô tả dữ liệu và muốn nhận gói dữ liệu thực, nó sẽ đưa ra yêu cầu với nút nguồn gửi gói tin mô tả đó bằng cách gửi thông điệp REQ. Do vậy, các thông tin quan sát được sẽ chỉ



gửi đến các nút cảm biến quan tâm như là kết quả của cuộc đàm phán này. Thứ hai, mỗi nút cảm biến tự kiểm tra quá trình tiêu thụ năng lượng của nó trước khi phát hay xử lý dữ liệu. Nếu mức năng lượng xuống thấp, nó sẽ giảm hay ngừng hoàn toàn các hoạt động xử lý hoặc chuyển tiếp thông điệp. Công việc này sẽ được nhường lại cho nút cảm biến khác có mức năng lượng cao hơn. Do đó, SPIN có thể cân bằng tiêu thụ năng lượng và kéo dài thời gian sống cho các nút mạng.

Một cải tiến của SPIN là SPIN-EC kết hợp kỹ thuật đàm phán với kiểm tra nguồn năng lượng còn lại dựa trên mức ngưỡng. Nút mạng chỉ có thể tham gia vào các hoạt động của giao thức khi nó hoàn thành tất cả các công việc mà không làm giảm năng lượng xuống dưới mức cho phép. Do đó, khi nút nhận được gói tin ADV, nó sẽ không gửi gói tin REQ nếu nguồn năng lượng còn lại không đủ để gửi gói tin REQ và nhận gói tin DATA.

Điểm yếu của SPIN là sự đàm phán chỉ mang tính cục bộ (trong một vùng mạng), do đó nó không đảm bảo các nút có nhu cầu đều nhận được thông điệp quảng bá và gói dữ liệu mong muốn. Tình trạng này có thể xảy ra nếu các nút trung gian không quan tâm tới gói dữ liệu và bỏ qua thông điệp ADV tương ứng. Điều này làm hạn chế việc áp dụng SPIN cho các ứng dụng mạng cảm biến không dây, đặc biệt các ứng dụng giám sát xâm nhập, bảo vệ các cơ sở hạ tầng quan trọng.

Giao thức khuếch tán trực tiếp (DD) [4, 81, 96] là giao thức định tuyến tiết kiệm năng lượng bằng cách tạo sự tương tác giữa các nút thông qua trao đổi thông điệp trong phạm vi lân cận. Các thành phần chính gồm thông tin yêu cầu "interest", thông điệp dữ liệu "data message", độ chênh lệch "gradients" và thông điệp củng cố "reinforcement". Thông điệp "interest" được nút nguồn (thường là BS hoặc sink) phát quảng bá (theo chu kỳ) để yêu cầu nút ở một vùng nào đó gửi dữ liệu cảm biến đến nó. Nút nguồn nhận được thông điệp "interest" từ BS gửi đến, nó sẽ thiết lập tuyến "gradients" theo hướng ngược lại (tuyến đảo) về phía BS và gửi dữ liệu theo tuyến này. Có thể có nhiều tuyến đảo "gradient" cho cùng một gói tin "interest" từ nút đích về nút nguồn, khi đó nút nguồn sẽ chọn một tuyến có chất lượng tốt nhất như tốc độ, liên kết ổn định, v.v. để truyền dữ liệu bằng cách gửi lại thông điệp củng cố "reinforcement". Trong quá trình quảng bá gói tin "interest" trong mạng thì



mỗi nút sẽ duy trì một hướng liên kết với nhau và lưu vào một "interest cache". Nó được dùng như bộ nhớ đệm bao gồm trường "timestamp" chứa thời gian của gói "interest", trường "multiple gradient" chứa tốc độ và hướng dữ liệu nhận được, và trường "duration" chỉ ra thời gian sống của thông điệp "interest". Các nút khi nhận được thông điệp "interest", nó sẽ kiểm tra xem có trùng với dữ liệu có trong "interest cache" của nó không. Nếu có, nút sẽ tính tốc độ cao nhất trong số các đường "gradient" liên kết với các nút lân cận, sau đó, nút gửi mô tả dữ liệu cho các nút có liên kết, ngược lại, nút bỏ qua thông điệp đó. Thông thường, khách tán trực tiếp cung cấp một tuyến đường đơn để truyền dữ liệu từ nút có dữ liệu tới nút nguồn cần. Tuy nhiên, nút nguồn cũng có thể lựa chọn nhiều con đường trong giai đoạn củng cố để truyền nhận dữ liệu đa đường. Bởi vậy, giao thức này phù hợp với các mạng thông tin cảm biến có nhiều nút nguồn (BS), tùy theo sự quảng bá thông điệp thăm dò "interest" và các kỹ thuật thiết lập tuyến "gradient".

### **2.2.3. Các giao thức định tuyến theo thông tin địa lý**

Khác với các giao thức kiến trúc phẳng, các giao thức định tuyến theo địa lý là sử dụng thông tin về vị trí địa lý để xây dựng tuyến đường hiệu quả giữa nguồn và đích. Nó không đòi hỏi kiến thức về toàn bộ kiến trúc mạng cũng như không cần thiết phải cập nhập các thông tin định tuyến theo định kỳ mà chúng chỉ yêu cầu thông tin thông qua một chặng duy nhất để đưa ra các quyết định chuyển tiếp dữ liệu, chẳng hạn như vị trí của các nút lân cận. Trong đó, các nút cảm biến trong cùng một mắt lưới (một ô vuông) đều có thể giao tiếp không dây với bất kỳ nút cảm biến nào ở trong mắt lưới bên cạnh.

Sự chính xác thích ứng theo địa lý (GAF) [37] giảm năng lượng tiêu thụ bằng cách tắt các nút không cần thiết trong mạng mà không ảnh hưởng đến mức độ chính xác của thông tin trong quá trình định tuyến. Để thực hiện điều này, GAF chia vùng mạng cảm biến thành các ô lưới ảo và sử dụng công nghệ GPS hoặc các hệ thống định vị khác để có thông tin vị trí của tất cả các nút mạng. Kích thước của các ô lưới ảo được xác định dựa trên phạm vi cường độ sóng của các nút cảm biến. Kết hợp giữa thông tin về vị trí và kích thước của lưới, mỗi nút cảm biến sẽ xác định được

mất lưới mà nó thuộc vào. GAF định nghĩa ba trạng thái cho các nút cảm biến, đó là trạng thái khám phá "discover", trạng thái hoạt động "active" và trạng thái ngủ "sleep" dựa vào các ngưỡng thời gian  $T_d$ ,  $T_a$  và  $T_s$  tương ứng. Mỗi nút cảm biến sẽ được chuyển đổi giữa các trạng thái để đảm bảo cân bằng tải năng lượng tiêu thụ mà không ảnh hưởng đến hoạt động định tuyến mạng.

Định tuyến cảm nhận năng lượng và địa lý (GEAR) [37, 81] sử dụng thông tin về năng lượng và vị trí địa lý để định tuyến gói tin đến vùng đích thay vì gửi thông tin quảng bá đến toàn mạng. Bằng cách này, GEAR hạn chế được số thông điệp cần gửi, do đó, đạt được hiệu quả sử dụng năng lượng và kéo dài thời gian sống cho mạng. GEAR duy trì ở mỗi nút hai giá trị là chi phí ước tính "estimated cost" và chi phí học "learning cost" trên con đường tới đích thông qua các nút hàng xóm của chúng. Chi phí ước tính là sự kết hợp của mức năng lượng còn lại và khoảng cách để đi tới đích trong khi chi phí học được xác định dựa trên việc định tuyến vòng quanh các vùng trống trong mạng. Vùng trống trong mạng xảy ra khi một nút không có bất kỳ nút hàng xóm lân cận nào mà gần vùng đích hơn nó, nếu trong mạng không xảy ra vùng trống thì chi phí ước tính bằng chi phí học. Giao thức GEAR gồm hai giai đoạn: Thứ nhất chuyển tiếp các gói tin tới vùng đích. Ở giai đoạn này, khi một nút nhận được một gói tin, nó sẽ kiểm tra các nút hàng xóm của nó để xem xét xem có nút nào gần vùng đích hơn nó hay không, nếu có nó sẽ chọn nút đó là chặng kế tiếp để định tuyến đến đích. Ngược lại, nếu tất cả các nút hàng xóm lân cận đều ở xa vùng đích hơn thì có nghĩa là sẽ có một vùng trống trong mạng. Trong trường hợp này, một trong những nút hàng xóm sẽ được lựa chọn trong tương lai để chuyển tiếp gói tin dựa trên hàm chi phí học và tránh vùng trống trong mạng. Sự lựa chọn này có thể được cập nhật sau đó dựa vào sự hội tụ của chi phí học trong suốt thời gian chuyển phát gói tin. Giai đoạn tiếp theo là chuyển tiếp các gói tin trong vùng đích. Khi gói dữ liệu đã được chuyển tới vùng đích, nó có thể được khuếch tán trong vùng bằng cách hoặc là chuyển tiếp địa lý đệ quy hoặc là phát tràn có giới hạn. Theo phương pháp phát tràn, mỗi nút nhận được gói dữ liệu, nó sẽ gửi đến tất cả các nút hàng xóm trừ nút đã gửi dữ liệu cho nó. Việc gửi này được lặp lại cho đến khi tất cả các nút trong mạng đều nhận được gói tin, do đó nó đảm bảo gói tin

sẽ đến được nút đích. Với phương pháp chuyển tiếp địa lý đệ quy, toàn bộ vùng đích sẽ được chia ra thành 4 vùng con và chuyển tiếp 4 bản sao của gói tin đến 4 vùng tương ứng. Quá trình chuyển tiếp được tiếp diễn cho đến khi mỗi vùng đó chỉ còn lại 1 nút mạng. Nếu các nút cảm biến được triển khai với mật độ dày đặc thì phương pháp chuyển tiếp địa lý đệ quy sẽ mang lại hiệu quả năng lượng tốt hơn phương pháp phát tràn vì giảm được số gói tin chuyển tiếp đến đích.

#### **2.2.4. Các giao thức dựa trên chất lượng dịch vụ**

Các giao thức thuộc loại này chủ yếu được thiết kế cho các ứng dụng đòi hỏi về chất lượng dịch vụ, ví dụ như đáp ứng nâng cao thông lượng, giảm độ trễ, giảm tỉ lệ lỗi gói tin, v.v. khi chuyển phát gói tin tới đích (BS). Trên thực tế, nâng cao hiệu quả tiêu thụ năng lượng cũng chỉ là một trong số các tiêu chí hiệu suất quan trọng của mạng cảm biến và nó không phải là yếu tố quan trọng nhất áp dụng cho mọi trường hợp.

Định tuyến phân chia tuần tự (SAR) được đề xuất trong [4, 81], đây là một trong những giao thức định tuyến đầu tiên được thiết kế cho WSNs, nó quan tâm đến chất lượng dịch vụ trong khi quyết định tuyến. SAR tạo nhiều tuyến đường từ nút nguồn tới đích để tránh quá tải hoặc lỗi trên một tuyến đường bằng cách xây dựng cây với nút nguồn bắt đầu là nút gốc cây (BS) nối tiếp với các nút hàng xóm của nó. Các tuyến trên cây tiếp tục được mở rộng khắp mạng và tránh các nút có mức năng lượng thấp hoặc chất lượng dịch vụ không được đảm bảo. SAR đạt được hiệu quả sử dụng năng lượng và kháng lỗi thông qua việc tính trọng số chất lượng dịch vụ cũng như độ đo định tuyến. Tuyến đường có chất lượng dịch vụ cao sẽ được sử dụng để chuyển tiếp các gói tin có độ ưu tiên cao.

Khác với SAR, giao thức SPEED [43, 56, 81] cung cấp phương pháp truyền thông thời gian thực cho mạng cảm biến thông qua việc duy trì tốc độ chuyển phát mong muốn, sự kết hợp của điều khiển phản hồi và chuyển tiếp địa lý không xác định. SPEED yêu cầu mỗi nút duy trì thông tin về các nút hàng xóm của nó sử dụng bảng gồm mã định danh, vị trí, độ trễ và thời hạn (*NeighborID*, *Position*, *SendToDelay*, *ExpireTime*), và sử dụng phương pháp chuyển tiếp địa lý không xác định phi trạng

thái-SNGF (Stateless Non-deterministic Geographic Forwarding) để tìm kiếm tuyến. Thêm nữa, các nút trong SPEED định kỳ trao đổi thông điệp báo hiệu "beacon" tới các nút hàng xóm của nó để thu thập thông tin về các nút và vị trí của chúng. Mỗi nút ước lượng độ trễ hành trình vòng đơn chặng bằng cách tính thời gian từ khi gói tin bắt đầu đi vào mạng tới nút đích cho đến khi nhận được thông điệp ACK của nút nhận. SPEED tính tốc độ chuyển tiếp giữa nút  $i$  và nút  $j$  hướng đến đích  $D$  được tính như (2.1) để chọn nút chuyển tiếp gói tin trong thuật toán chuyển tiếp địa lý không xác định phi trạng thái (SNGF).

$$\text{Speed}_i^j(D) = \frac{d(i, D) - d(j, D)}{\text{HopDelay}_i^j} \quad (2.1)$$

Trong đó,  $d(i, D)$ ,  $d(j, D)$  và  $\text{HopDelay}_i^j$  tương ứng là các khoảng cách để đến được đích  $D$  từ nút  $i$  và nút  $j$  và độ trễ ước tính giữa nút  $i$  và  $j$ .

Khi so sánh với các giao thức DSR và AODV, SPEED thực hiện tốt hơn về độ trễ đầu cuối, tỉ lệ lỗi gói tin và tổng năng lượng truyền thông cũng như chi phí định tuyến nhỏ [43]. Tuy nhiên, SPEED không xem xét đến độ đo về năng lượng của các nút trong quá trình quyết định tuyến, do đó, không có hiểu biết về năng lượng tiêu thụ của SPEED so với các giao thức nhận biết về năng lượng khác.

### 2.2.5. Các giao thức có thứ bậc

Khác với các lược đồ định tuyến phẳng, lược đồ địa lý và lược đồ dựa trên chất lượng dịch vụ, các giao thức trong lược đồ phân cấp, nơi mà các nút mạng được tổ chức phân cấp bằng cách gom lại thành các cụm (cluster) riêng biệt. Mỗi cụm sẽ bầu ra một nút đứng đầu cụm gọi là nút "cụm trưởng" (CH), nó chịu trách nhiệm quản lý các hoạt động của cụm, thu nhận, tổng hợp dữ liệu, và chuyển tiếp đến BS. Các nút khác (gọi là các nút thành viên cụm) sẽ tương tác cục bộ với nút cụm trưởng và chịu sự điều khiển của nút cụm trưởng. Việc phân cấp thứ bậc có thể được phân thành nhiều tầng (lớp) khác nhau như cấu trúc "cây". Dựa trên kiến trúc này, một số giao thức định tuyến phân cấp đã được phát triển để giải quyết bài toán định tuyến đáp ứng được các vấn đề trong việc tiêu thụ năng lượng và mở rộng thời gian hoạt động cho mạng cảm biến không dây.

### 2.2.5.1. Phân cụm hiệu quả năng lượng dựa trên xác suất

Các phương pháp định tuyến có thứ bậc hiệu quả năng lượng dựa trên mô hình xác suất làm tiêu chuẩn để chọn nút ứng viên làm cụm trưởng đã được đề xuất như LEACH [4, 44, 69, 112]. Với mục tiêu là giảm năng lượng tiêu thụ ở mỗi nút mạng, kéo dài thời gian sống của mạng, sử dụng tổng hợp dữ liệu ở nút cụm trưởng để giảm số lượng dữ liệu truyền đi đến BS. LEACH chọn nút CH một cách ngẫu nhiên từ một tập nút cảm biến sau một khoảng thời gian cho trước sẽ quay vòng vai trò của nút cụm trưởng cho các nút khác nhằm phân bố đều tải tiêu thụ năng lượng cho các nút cảm biến khác trong mạng từ đó kéo dài thời gian sống toàn mạng.

Theo kết quả mô phỏng cho thấy giao thức LEACH đã góp phần đáng kể vào việc tiết kiệm năng lượng tiêu thụ ở các nút mạng nhờ vào việc tổng hợp dữ liệu ở các nút cụm trưởng và lựa chọn quay vòng các nút cụm trưởng trong mạng. Tuy nhiên, ở LEACH vẫn còn một số điểm hạn chế như: Chưa xác định cụ thể số lượng tối ưu các nút CH trong mạng, khi mà các mạng có cấu hình và số lượng nút khác nhau, chưa có gợi ý khi nào thì việc tái xác định lại các nút cụm trưởng cần được thực hiện. Hơn nữa, khoảng cách từ nút CH đến trạm cơ sở vẫn còn xa do các nút CH được chọn ở vị trí ngẫu nhiên mà nút CH phải làm việc nhiều hơn, do đó nó sẽ nhanh hết năng lượng hơn các nút khác trong nhóm. Một vấn đề nữa là khoảng thời gian tồn tại ở pha ổn định trạng thái trong mỗi vòng. Nó có ảnh hưởng lớn đến năng lượng tiêu thụ của toàn mạng vì nếu khoảng thời gian này là ngắn thì sẽ làm tăng tổng chi phí trong giai đoạn thiết lập cụm, nếu dài thì các nút cụm trưởng sẽ hết năng lượng nhanh chóng (mà chúng ta mong muốn kéo dài thời gian hoạt động của toàn mạng).

Từ đó đã có nhiều phiên bản cải tiến từ giao thức LEACH, một thí dụ là giao thức XLEACH (Extended LEACH), nó xem xét đến mức năng lượng của các nút trong khi chọn nút đó làm nút cụm trưởng, khi đó mức ngưỡng  $T(i)$  được xác định bởi biểu thức (2.2) [96]:

$$T(i) = \frac{k}{1 - k(r \bmod (1/k))} \left[ \frac{E_{\text{residual}}(i)}{E_{\text{init}}} + \left( r_i \bmod \frac{1}{k} \right) \left( 1 - \frac{E_{\text{residual}}(i)}{E_{\text{init}}} \right) \right] \quad (2.2)$$

Trong đó: -  $E_{\text{residual}}(i)$  là năng lượng còn lại ở vòng hiện tại của nút  $i$

-  $E_{\text{init}}$  là năng lượng khởi tạo của nút cảm biến thứ  $i$

-  $k$  là tỉ lệ phần trăm nút cụm trưởng trong mạng

-  $r_i$  chỉ định số vòng liên tiếp mà nút  $i$  không được làm nút cụm trưởng.

Khi giá trị của  $r_i$  tiến sát  $1/k$ , giá trị ngưỡng  $T(i)$  được khởi tạo lại giá trị nó có trước khi xem xét đến năng lượng còn lại trong biểu thức  $T(i)$ . Thêm nữa,  $r_i$  được thiết lập về 0 khi một nút trở thành CH.

Giao thức ALEACH [6] chọn nút CH có xem xét đến xác suất tổng thể của nút ứng viên (được tính giống như LEACH) và xác suất trạng thái hiện tại được tính dựa vào năng lượng còn lại và năng lượng khởi tạo của nút ứng viên. Một giao thức khác là MS-LEACH [86], kết hợp giữa truyền đa chặng và đơn chặng trong giai đoạn truyền dữ liệu để tránh các liên kết đơn chặng cách xa nhau để giảm năng lượng tiêu thụ, mở rộng thời gian sống mạng. MIN-RC [34] điều khiển thời gian hoạt động trong mỗi vòng trong giai đoạn truyền dữ liệu thay vì cố định như LEACH và LEACH-C [44, 75] và xem xét đến số lượng cụm cho mỗi vòng để nâng cao hiệu quả sử dụng năng lượng mạng. Trong các đề xuất khác nữa bao gồm LEACH-G [22], E-LEACH [112], OMLEACH [113], LEACH-HPR [40], [44], [62], việc tính toán tối ưu giá trị số cụm được phân chia sao cho phù hợp với tô-pô mạng trong giao thức LEACH để kéo dài thời gian sống cho mạng. Trong các đề xuất này, các tác giả cũng đánh giá hiệu năng của giao thức đề xuất bằng cách sử dụng mô hình toán học kết hợp với mô phỏng trên máy tính. Các kết quả cho thấy nếu tô-pô mạng phẳng thì giá trị tối ưu cho số cụm nên được phân chia phù hợp cho mỗi vòng là từ 3% đến 5% tổng số nút mạng thì giao thức LEACH sẽ cho hiệu quả sử dụng năng lượng tốt nhất. Cụ thể là với đề xuất [22], số CH là 3% sẽ cho số lượng vòng mô phỏng cao nhất; với đề xuất [113] số CH là 4% và đề xuất [44], số CH bằng 5% thì năng lượng tiêu thụ trung bình trên mỗi vòng sẽ nhỏ nhất.

Gần đây, các đề xuất cải tiến kỹ thuật phân cụm trong DB-LEACH [77], hoặc trong DIST\_LEACH [15], [14] đã cải tiến hiệu quả sử dụng năng lượng của LEACH khi bầu chọn nút làm CH xem xét đến khoảng cách và năng lượng còn lại của nút ứng

viên. Chúng phải lớn hơn mức năng lượng trung bình của các nút còn sống trong mạng ở vòng hiện tại.

Lược đồ phân cụm hiệu quả năng lượng EECS [114] là lược đồ phân cụm phân tán giống như LEACH. Tuy nhiên, để chọn nút cụm trưởng, các nút sẽ chọn một giá trị ngẫu nhiên  $\mu$  trong khoảng từ 0 đến 1 sau đó so sánh với giá trị xác suất  $T$ , (giá trị  $T$  nên được thiết lập sao cho đảm bảo được hiệu quả hoạt động mạng và giảm tổng chi phí thiết lập cụm vì nếu  $T$  lớn, sẽ có nhiều nút trở thành CH và ngược lại). Nếu giá trị  $\mu$  nhỏ hơn  $T$  thì nó được chọn là nút ứng viên và quảng bá thông điệp "COMPETE\_HEAD\_MSGs", nó chứa số định danh và năng lượng còn lại của nút. Tiếp theo, thuật toán lặp cho đến khi ngưỡng thời gian "Timer" (chúng được thiết lập bởi người dùng) chưa hết hạn thì thực hiện, kiểm tra số nhận dạng và năng lượng còn lại của nó với các nút khác để chọn ra nút CH có mức năng lượng còn lại cao. Sau khi chọn được CHs, các nút CHs quảng bá thông điệp "HEAD\_AD\_MSG", nó chứa số định danh và khoảng cách từ nó đến BS. Các nút khác nhận thông điệp và tính toán giá trị "*cost*" để thực hiện gia nhập cụm với CH có giá trị "*cost*" nhỏ nhất.

$$\text{cost}(j, i) = w \times f(d(P_j, CH_i)) + (1 - w) \times g(d(CH_i, BS)) \quad (2.3)$$

trong đó,  $P_j$  là nút thành viên thứ  $j$  trong cụm,  $CH_i$  là nút CH thứ  $i$  trong mạng, và các hàm  $f()$ ,  $g()$ ,  $w()$  được tính như sau:

$$f(P_j, CH_i) = \frac{d(P_j, CH_i)}{d_{f\_max}} \quad (2.4)$$

$$g(CH_i) = \frac{d(CH_i, BS) - d_{g\_min}}{d_{g\_max} - d_{g\_min}} \quad (2.5)$$

các giá trị  $d_{f\_max} = \exp(\max\{d(P_j, CH_i)\})$ ,  $d_{g\_max} = \max\{d(CH_i, BS)\}$ ,

$d_{g\_min} = \min\{d(CH_i, BS)\}$  và  $d(P_j, CH_i)$  là khoảng cách địa lý từ nút thành viên  $P_j$  đến nút  $CH_i$ .

$$w(P_j) = c + (1 - c) \sqrt{\frac{d(P_j, BS)}{(d_{g\_max} - d_{g\_min})}} \quad (2.6)$$

Hàm  $f()$  ở công thức (2.4) đảm bảo rằng nút thành viên chọn CH gần nhất để tối thiểu chi phí truyền thông trong cụm, trong khi hàm  $g()$  giúp nút thành viên gia



nhập cụm có khoảng cách từ CH đến BS là nhỏ. Hàm con  $w()$  là yếu tố cân bằng giữa  $f()$  và  $g()$ . Hơn nữa, hằng số  $c$  thể hiện đặc tính của mạng.

Khác với LEACH, giao thức phân cụm phân tán lai ghép hiệu quả năng lượng - HEED [4, 69, 115] được đề xuất bởi Younis và Fahmy, là giao thức định tuyến phân cụm hiệu quả năng lượng bằng cách truyền đa chặng bên trong cụm giữa CH và các nút thành viên cụm trong suốt giai đoạn truyền dữ liệu. Hơn nữa, HEED định kỳ chọn nút cụm trưởng dựa vào hàm  $CH_{prob}$ , chúng kết hợp giữa năng lượng còn lại của nút ứng viên và chi phí truyền thông của các nút thành viên trong cụm với nút ứng viên. Khi khởi tạo cụm, các nút tự sinh ra một giá trị ngẫu nhiên, nếu giá trị ngẫu nhiên này nhỏ hơn giá trị  $CH_{prob}$  (giá trị xác suất để trở thành CH) thì nút trở thành nút ứng viên CH, giá trị  $CH_{prob}$  được tính như (2.7) dưới đây:

$$CH_{prob} = \max \left\{ k_{opt} \frac{E_{residual}}{E_{init}}, p_{min} \right\} \quad (2.7)$$

trong đó,  $E_{residual}$  là năng lượng còn lại của nút ở vòng hiện tại,  $E_{init}$  là năng lượng tham chiếu lớn nhất (khi pin được sạc đầy),  $k_{opt}$  là tỉ lệ tối ưu giới hạn số CH ban đầu (chúng thường được chọn bằng 5% [44, 113]) và  $p_{min}$  là giá trị ngưỡng để đảm bảo  $CH_{prob}$  không nhỏ hơn  $p_{min}$  (ví dụ  $10^{-4}$  [115]). Sau đó, các nút ứng viên gửi thông điệp chứa mã định danh của nó và giá trị  $CH_{prob}$  đến các nút hàng xóm bên trong vùng của nó. Có hai kiểu thông điệp mà một nút cảm biến có thể gửi đến các nút hàng xóm của nó là "*tentative\_CH*", chỉ định nút chưa chính thức là CH và "*final\_CH*" chỉ định nút đã chính thức là CH. Nếu có nút không nghe thấy các thông điệp này thì nó sẽ tự gấp đôi giá trị  $CH_{prob}$  của nó cho vòng lặp tiếp theo để gần đến 1, nếu giá trị này bằng 1 thì nó sẽ tự trở thành CH và gọi là nút cô lập. Các nút khác chọn CH để gia nhập cụm dựa vào tham số năng lượng còn lại của CH và chi phí truyền thông trong nhóm. HEED đạt được các ưu điểm sau [115]: Chọn CH xem xét đến năng lượng còn lại của nút, do đó, chọn được các nút có mức năng lượng cao làm CH; truyền thông đa chặng giữa CH và BS, do đó, ổn định và tiết kiệm nguồn năng lượng hơn giao thức truyền trực tiếp và xa như LEACH. Tuy nhiên, hạn chế của HEED gồm: Thứ nhất, sử dụng thông điệp "*tentative\_CH*" đối với các nút không phải CH cuối cùng sẽ dẫn đến lãng phí tài nguyên mạng. Thứ hai, hoạt động



xây dựng lại cụm ở mỗi vòng sẽ tiêu tốn nhiều năng lượng cho tổng chi phí định tuyến. Cuối cùng, các nút CHs ở gần BS có thể sẽ hết năng lượng sớm hơn các nút khác vì phải truyền dữ liệu nhiều hơn các nút khác.

Gần đây, các nhà nghiên cứu đã tập trung xem xét đến hệ thống mạng hỗn hợp, nơi mà các nút cảm biến không dây có các chức năng, khả năng tính toán, truyền thông và mức năng lượng khởi tạo không giống nhau, điều này giống với thực tế hơn nhằm mục đích nâng cao hiệu quả sử dụng năng lượng, mở rộng thời gian sống cho mạng [42, 51, 124]. Các nút được cấy thêm vào mạng thường có khả năng xử lý mạnh và mức năng lượng cao hơn nút khác. Do đó, nếu các nút này được chọn làm CH, mạng sẽ hoạt động hiệu quả hơn vì nó rất phù hợp với nút phải xử lý nhiều công việc và tiêu tốn nhiều năng lượng. Ví dụ như: ATCEEC [48], CEEC [13], LEACH-HPR [40], MAMC [42], EE-TLDC [51], U-LEACH [57], [85, 87, 124]. Các đề xuất [13, 51], mở rộng giao thức xây dựng cụm LEACH trong mạng hỗn hợp, chọn nút ứng viên trở thành CH xem xét đến năng lượng khởi tạo, năng lượng còn lại của nút ứng viên và khoảng cách từ nó đến BS. Các tham số xem xét thường có lợi cho nút được cấy thêm vào mạng để nâng cao hiệu quả tổ chức cụm và duy trì hoạt động cụm tốt hơn, từ đó, nâng cao hiệu quả sử dụng năng lượng. Hơn nữa, các giao thức này sử dụng phương pháp truyền thông đa chặng từ CH đến BS thay vì đơn chặng như LEACH để tránh tiêu tốn năng lượng cho các nút CH ở xa BS, do đó, đạt hiệu quả năng lượng tốt hơn LEACH.

#### **2.2.5.2. Định tuyến phân cụm tập trung**

Trong kỹ thuật định tuyến phân cụm tập trung, việc chọn nút ứng viên làm CH và phân chia cụm được thực hiện bởi một nút trung tâm, nút này có khả năng tính toán lớn và có nguồn năng lượng không bị hạn chế (sink hoặc BS), do đó mạng sẽ không tiêu tốn nhiều năng lượng cho việc xây dựng cụm như LEACH-C (LEACH-Centralized) [44, 75]. LEACH-C, là một cải tiến của LEACH ở giai đoạn thiết lập; các nút phải gửi toàn bộ thông tin về năng lượng còn lại, số định danh, vị trí địa lý của nó trong mạng tới BS ở khoảng cách xa. Dựa vào thông tin này, BS sẽ tính toán mức năng lượng trung bình và lựa chọn số nút làm cụm trưởng sao cho phân bổ tải

tiêu thụ năng lượng đều cho các nút trên toàn mạng. Đây là một bài toán khó và trở thành một thách thức lớn cho các nhà nghiên cứu để tìm ra số cụm tối ưu cho toàn bộ mạng, cân bằng số nút cho các cụm và khoảng cách phân bố đều cho các nút thành viên tham gia nhóm, còn phương pháp chọn CH là giống với LEACH. Sau đó thông tin cụ thể về các cụm được gửi quảng bá đến các nút trong mạng. Các nút trong mạng nhận được thông điệp quảng bá sẽ biết được rằng nó ở cụm nào và CH ở đâu để gửi dữ liệu. Với kỹ thuật này, ưu điểm là sử dụng nút sink, nút không bị giàng buộc về năng lượng, bộ nhớ và khả năng tính toán để tính toán, phân chia nhóm.

**Bảng 2.1:** So sánh các thuật toán phân cụm

Thuật toán phân cụm	Phân tán	Tập trung	Đặc tính truyền thông	Chọn CH dựa trên	Hiệu quả năng lượng	Độ phức tạp tính toán	Độ phức tạp thông báo
Dựa trên LEACH [6, 40, 49, 77, 86]	✓		Đơn chặng	Xác suất	Cao	$O(n)$	$O(1)$
LEACH-C [75]		✓	Đơn chặng	Xác suất	Cao	$O(n)$	$O(1)$
EECS [114]	✓		Đơn chặng	Xác suất	Cao	$O(n)$	$O(n)$
HEED [115]	✓		Đa chặng	Lắp	Cao	$O(1)$	$O(n)$
CEEC [13]		✓	Đơn chặng	Xác suất	Cao	$O(n)$	$O(1)$
SDWSN [109]	✓		Đơn chặng	Khoảng cách và năng lượng	Cao	$O(n)$	$O(n)$

Tuy nhiên, nếu BS ở xa, các nút sẽ tiêu tốn năng lượng cho giai đoạn đầu của mỗi vòng cho việc truyền thông tin với BS. Giao thức điển hình cho việc phân cụm tập trung là LEACH-C [75, 110] được đề xuất bởi Siva D. Muruganathan và các cộng sự, nhóm tác giả cũng tính toán và đưa ra được số cụm thiết lập cho mỗi vòng tối ưu là 5% số nút cho mỗi mạng.

Bảng 2.1 tóm tắt những điểm khác nhau chính giữa các thuật toán phân cụm, truyền thông đơn chặng được khảo sát. Các tiêu chuẩn đánh giá gồm độ phức tạp tính toán, độ phức tạp thông báo, nó được ước lượng bằng tổng số gói tin điều khiển và gói dữ liệu được gửi ở một nút mạng, ở đây, chúng tôi giả thiết rằng kích thước các gói tin là như nhau. Trong thực tế, độ phức tạp thông báo thông thường được đánh giá bằng tổng kích thước của các gói tin (tính bằng số "bít") được sử dụng trong giai đoạn thiết lập cụm vì chúng có ảnh hưởng trực tiếp đến mức tiêu thụ năng lượng của các nút.

#### **2.2.5.3. Phân cụm hiệu quả năng lượng dựa trên chuỗi**

Thay vì các nút trong cụm kết nối, truyền thông trực tiếp một chặng với nút cụm trưởng, các giao thức định tuyến dựa trên chuỗi giảm khoảng cách truyền giữa các nút trong cụm bằng cách kết nối và truyền thông với nút hàng xóm gần nhất, do đó tăng hiệu quả sử dụng năng lượng của mạng cảm biến không dây.

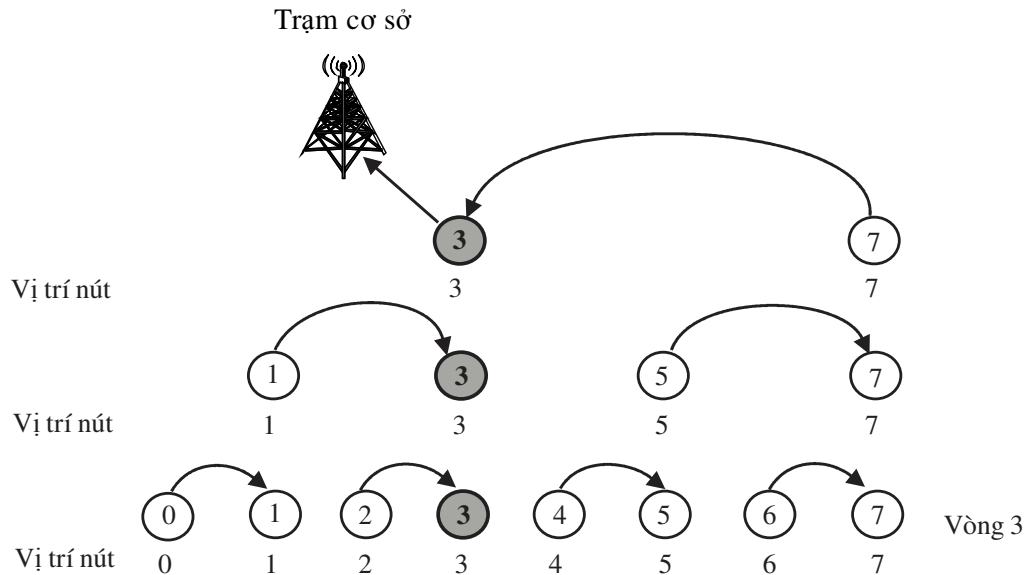
Giao thức tập hợp hiệu quả năng lượng trong hệ thống thông tin cảm biến (PEGASIS) [38, 64, 93] thuộc họ giao thức định tuyến phân cụm dựa trên chuỗi trong mạng cảm biến không dây, PEGASIS thực hiện hai nhiệm vụ chính:

- Kéo dài thời gian sống cho mạng bằng cách nâng cao hiệu quả tiêu thụ năng lượng và cân bằng tiêu thụ năng lượng cho tất cả các nút
- Giảm thời gian trễ truyền từ các nút đến BS

PEGASIS xem xét mô hình mạng bao gồm tập hợp các nút mạng có cấu trúc đồng nhất (các nút có dung lượng bộ nhớ, nguồn pin và bộ vi xử lý như nhau) được triển khai trên một vùng địa lý, các nút này có sự hiểu biết về vị trí các nút khác trong mạng. Hơn nữa các nút còn có khả năng điều chỉnh công suất và được trang bị bộ thu phát sóng vô tuyến dùng kỹ thuật CDMA. Mục tiêu của giao thức định tuyến

này là giảm năng lượng tiêu thụ và truyền dữ liệu thu thập được từ các nút đến trạm gốc với độ trễ thấp. Khác với các giao thức khác dựa trên cấu trúc mạng hình cây hay tổ chức mạng phân chia theo cụm, PEGASIS sử dụng cấu trúc chuỗi, các nút chỉ truyền thông với nút hàng xóm gần nó nhất. Việc xây dựng chuỗi bắt đầu từ nút xa nút BS nhất và các nút mới được thêm vào để chuỗi tăng lên bắt đầu từ nút hàng xóm gần nhất tới nút cuối cùng. Để xác định các hàng xóm gần nhất, một nút đo độ mạnh của tín hiệu nhận được từ tất cả các nút hàng xóm bằng đơn vị chức năng RSSI [5, 61]. Sử dụng thông tin này các nút cảm biến điều chỉnh cường độ tín hiệu phát đủ để chỉ các nút gần nhất nghe thấy. Sau đó, PEGASIS chọn một nút có vị trí ngẫu nhiên bên trong chuỗi làm nút CH, nút này có nhiệm vụ tập hợp dữ liệu và truyền dữ liệu tập hợp được tới trạm cơ sở. Vai trò nút cụm trưởng sẽ được chuyển cho nút khác sau mỗi vòng (khoảng thời gian cố định). Việc quay vòng nút CH trong chuỗi nhằm đảm bảo cân bằng tải tiêu thụ năng lượng giữa các nút trong mạng. Tuy nhiên, điều đáng chú ý là việc thay đổi vai trò nút cụm trưởng trong chuỗi có thể dẫn đến nút cụm trưởng cách xa nút trung chuyển BS, khi đó nút này lại cần công suất cao để truyền đến nút trung chuyển. Việc tổng hợp dữ liệu dọc theo chuỗi có thể được thực hiện tuần tự. Đầu tiên nút cụm trưởng sẽ gửi một thẻ bài tới nút cuối cùng bên phải chuỗi. Khi nhận thẻ bài, nút cuối sẽ gửi dữ liệu cảm biến được đến nút lân cận theo chiều hướng tới nút CH, tiếp đó nút này lại tổng hợp dữ liệu và tiếp tục gửi đến nút lân cận kế tiếp. Khi nhận được dữ liệu từ nút bên phải chuỗi, nút CH lại gửi thẻ bài tới nút cuối bên trái chuỗi. Cứ như vậy cho đến khi nút CH nhận được dữ liệu từ cả hai bên chuỗi, khi đó, nó thực hiện tổng hợp dữ liệu và gửi đến BS. Mặc dù kỹ thuật này là đơn giản nhưng mô hình tổng hợp tuần tự này dễ gây ra trễ trước khi dữ liệu tập trung được gửi đến BS. Do đó, một phương pháp khác được nêu ra để giảm độ trễ là dùng mô hình tổng hợp dữ liệu song song dọc theo chuỗi, và nó sẽ giảm nhiều hơn nếu các nút mạng được trang bị bộ thu phát sử dụng CDMA. Hình 2.3 minh họa giải thuật thu thập và tổng hợp dữ liệu song song dựa trên chuỗi trong một vòng của PEGASIS [65, 96]. Trong lược đồ Hình 2.3, giả thiết rằng tất cả các nút đều biết được các thông tin chung về mạng và được liên kết thành chuỗi. Hơn nữa, các nút luân phiên nhau tổng hợp, truyền dữ

liệu tới trạm cơ sở dựa theo vòng. Tại vòng thứ  $i$  thì nút thứ  $(i \bmod N)$  với  $N$  là số nút tham gia vào chuỗi sẽ chịu trách nhiệm truyền dữ liệu tới nút cơ sở. Dựa vào phương pháp này thì nút 3 ở vị trí số 3 là nút cụm trưởng trong vòng đang xét (vòng 3).



**Hình 2.3:** Lược đồ thu thập và tập hợp dữ liệu song song trên chuỗi trong một vòng

Mức đầu tiên các nút được đánh số chẵn sẽ phát dữ liệu tập hợp được cho nút bên phải của mình. Mức kế tiếp, các nút đang có dữ liệu tổng hợp nó được đánh số lại và các nút đánh số chẵn sẽ phát dữ liệu cho nút ngay bên phải của nó. Ở mức cuối cùng, nút 7 được đánh số 1 là nút chứa dữ liệu lân cận cuối cùng của nút cụm trưởng 3. Nút 3 tổng hợp dữ liệu nhận được của các nút gửi tới trước đó cùng với dữ liệu nó cảm biến được và chuyển tiếp tới BS. Lược đồ đảm bảo rằng sau  $\log_2 N$  bước, dữ liệu tổng hợp sẽ được gửi đến nút CH [96].

Các kết quả mô phỏng cho thấy cấu trúc mở rộng của PEGASIS cũng như khả năng cải tiến của giao thức LEACH đã cải thiện được đáng kể các nhược điểm của các giao thức định tuyến truyền thống [65]. Tuy nhiên, Yongchang Yu cùng các cộng sự [117] đã cho thấy thuật toán xây dựng chuỗi như PEGASIS khó tránh khỏi các liên kết dài - "long chain". Từ đó, nhóm tác giả đã đưa ra giải pháp cải tiến bằng cách sử dụng một giá trị ngưỡng khoảng cách  $d_{\text{threshold}}$ . Trước khi nút  $i+1$  gia nhập chuỗi,

nút  $i+1$  sẽ kiểm tra khoảng cách  $d_i$  của nó với ngưỡng  $d_{\text{threshold}}$ , nếu  $d_i$  nhỏ hơn  $d_{\text{threshold}}$  thì nút được gia nhập chuỗi. Ngược lại, nút  $i+1$  sẽ được kết nối với một nút khác gần chuỗi nhất. Thêm nữa, Kyu Sung Ahn cùng các cộng sự đã đề xuất lược đồ định tuyến chuỗi cho hiệu quả sử dụng năng lượng BCBRP [3], nơi mà toàn bộ vùng cảm biến mạng được chia thành các cụm (hoặc mạng con) có kích thước bằng nhau. Mỗi mạng con xây dựng một chuỗi sau đó chúng được kết nối với nhau thông qua các nút bắc cầu (bridge nodes). Với BCBRP, khoảng cách lớn nhất giữa các nút trong chuỗi sẽ giảm bằng cách giảm kích thước các mạng con, do đó, tiết kiệm được năng lượng và kéo dài thời gian sống cho mạng.

Gần đây, một công trình của Feng Sen cùng các cộng sự [93] đã đề xuất phương án cải tiến PEGASES gọi là IEEPB. Để giảm các liên kết dài "long link" giữa các nút trong chuỗi, IEEPB thực hiện chọn nút gia nhập chuỗi không chỉ so sánh với nút cuối chuỗi "end node" mà còn so sánh với các nút khác trong chuỗi để tìm ra nút kết nối gần nhất trong chuỗi. Hơn nữa, IEEPB cũng xem xét đến năng lượng còn lại của các nút ứng viên và khoảng cách từ nó đến BS để chọn nút lãnh đạo chuỗi (CH). Do vậy, IEEPB cho hiệu quả sử dụng năng lượng tốt hơn PEGASIS.

Một đề xuất xây dựng chuỗi khác được Yongchang Yu và Gang Wei đưa ra là LBEERA [118], nơi mà toàn bộ vùng cảm biến mạng được chia thành các vùng, cụm cố định khác nhau. Các nút trong cụm được xây dựng thành các chuỗi riêng biệt, chúng gồm nút cụm trưởng và các nút thành viên trong cụm. Dữ liệu được truyền định kỳ từ nút xa nhất trong chuỗi đến nút hàng xóm kế tiếp và đến CH, các CH được chia ra thành CH và siêu CH dựa vào khoảng cách từ chúng tới BS. Các nút siêu CH sẽ ở gần BS hơn và chịu trách nhiệm gửi dữ liệu đến BS còn các nút CH ở xa sẽ gửi dữ liệu đến các siêu CH. Nhờ đó, LBEERA đạt được tỉ lệ nút chết thấp hơn và có độ trễ tuyến nhỏ hơn khi so sánh với LEACH và PEGASIS [118].

Wenjing Guo và các cộng sự [38] đã đề xuất phương án cải tiến PEGASIS, gọi là PEG-ant, sử dụng thuật toán đàn kiến để xây dựng chuỗi, tạo ra tổng khoảng cách toàn cục là nhỏ nhất thay vì sử dụng thuật toán tham lam. Thêm nữa, trong quá trình chọn nút lãnh đạo chuỗi, PEG-ant xem xét đến năng lượng còn lại của nút ứng viên để chọn làm nút lãnh đạo chuỗi để tăng thời gian sống cho mạng.

Khác với các đề xuất trên, trong các đề xuất [23, 32, 50] thực hiện chia toàn bộ vùng cảm biến ra thành các cụm, cung nhỏ gọi là "track-sector", các cung này có tâm là nút đích (BS). Các nút trong mỗi cung được chia ra thành các mức khác nhau, chúng xa dần tính từ nút BS. Việc chia này kết hợp với việc kết nối các nút thành chuỗi nhỏ sử dụng thuật toán tham lam đã giúp giảm dữ liệu dư thừa và khoảng cách truyền giữa CH và BS, do đó, nâng cao hiệu quả sử dụng năng lượng và kéo dài thời gian sống cho mạng.

Trong đề xuất giao thức lập lịch ngủ thông tin địa lý và định tuyến hiệu quả năng lượng dựa trên chuỗi GSSC [70], các tác giả chia vùng cảm biến thành các cụm, mỗi cụm gồm nhiều ô lưới ảo dựa trên thông tin vị trí của các nút. Trong mỗi ô lưới nhỏ, chỉ có một nút có năng lượng còn lại lớn nhất ở trạng thái hoạt động tích cực, các nút khác đi vào chế độ ngủ để tiết kiệm năng lượng và loại bỏ dữ liệu trùng lặp. Hơn nữa, GSSC xây dựng chuỗi cho mỗi cụm bắt đầu từ nút ở ô lưới xa nhất để giảm khoảng cách truyền thông giữa các nút trong cụm, định tuyến dữ liệu cảm biến được từ các nút hoạt động tích cực tới BS.

Ying Tian cùng cộng sự đã đề xuất giao thức định tuyến cụm chuỗi hiệu quả năng lượng (ECR) [104]. ECR kết hợp giữa chia toàn bộ vùng cảm biến thành các vùng nhỏ có kích thước bằng nhau, mỗi vùng tương ứng với một cụm, mỗi cụm xây dựng một chuỗi tương ứng giống như PEGASIS. Ở vòng thứ  $r$ , BS chọn một nút đóng vai trò như CH dựa vào ID của nút (mỗi nút trong mạng được gán một số gọi là ID) và mức năng lượng được chèn vào trong gói dữ liệu của các nút, chúng được truyền tới BS ở vòng thứ  $r - 1$  trước đó. Ưu điểm của đề xuất này là dữ liệu truyền trong mạng theo chiều dài chuỗi trong nhiều cụm khác nhau, do đó, giảm được độ trễ truyền thông và tránh tắc nghẽn mạng. Tuy nhiên, việc gửi ID của nút và năng lượng còn lại của nó trong các gói dữ liệu dẫn đến tăng chi phí truyền thông mà truyền dữ liệu là một trong những hoạt động tiêu tốn năng lượng nhiều nhất trong WSN. Bảng 2.2 tóm tắt những điểm khác nhau chính giữa các thuật toán phân cụm dựa trên chuỗi được khảo sát.

**Bảng 2.2:** So sánh các thuật toán phân cụm dựa trên chuỗi

Thuật toán phân cụm	Phân tán	Tập Trung	Độ trễ truyền thông	Chọn CH dựa trên	Hiệu quả năng lượng	Độ phức tạp tính toán	Độ phức tạp thông báo
Dựa trên chuỗi dài [64, 116]		✓	Cao	Xác suất	Cao	$O(n.\log(n))$	$O(1)$
Dựa trên chuỗi dài [38, 93, 117]		✓	Cao	Khoảng cách và năng lượng	Cao	$O(n.\log(n))$	$O(1)$
Dựa trên cụm chuỗi [32, 104]		✓	Thấp	Ngẫu nhiên	Cao	$O(n.\log(n))$	$O(1)$
Dựa trên cụm chuỗi [23, 70, 101, 118]		✓	Thấp	Khoảng cách và năng lượng	Cao	$O(n.\log(n))$	$O(1)$
Dựa trên cụm chuỗi [7]	✓		Thấp	Xác suất	Cao	$O(n.\log(n))$	$O(1)$

#### 2.2.5.4. Phân cụm hiệu quả năng lượng dựa trên cây tối thiểu

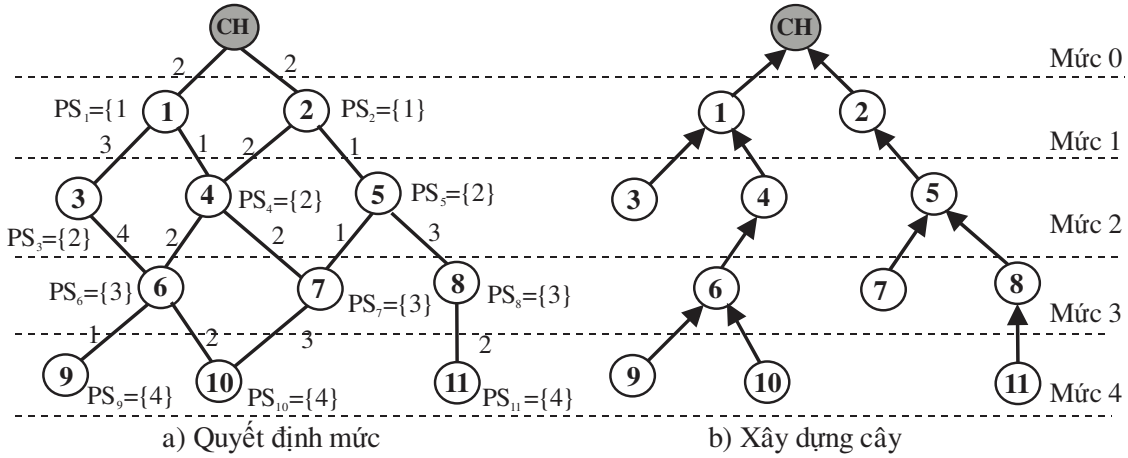
Khác với lược đồ định tuyến dựa trên chuỗi, các giao thức định tuyến theo lược đồ phân cụm dựa trên cây cho hiệu quả năng lượng tốt bằng cách kết hợp giữa phân cụm và xây dựng cây tối thiểu sử dụng thuật toán Prim để giảm khoảng cách truyền giữa các nút trong mạng. Điển hình là giao thức tập hợp dữ liệu cây được phân cụm TCDGP [46], nơi mà toàn bộ các nút mạng trong vùng cảm biến được chia ra thành các cụm, các nút trong cụm kết nối với nhau thành cây tối thiểu với nút gốc của cây



là CH. Để chọn nút ứng viên làm CH, TCDGP tính toán giá trị ngưỡng năng lượng truyền dựa vào tham số năng lượng còn lại và khoảng cách truyền từ nút ứng viên đến BS. Các kết quả mô phỏng cho thấy TCDGP cho hiệu quả năng lượng tốt khi di chuyển BS ra xa vùng cảm biến. Tuy nhiên, TCDGP không đảm bảo cân bằng số nút trong các cụm vì các nút chọn CH có khoảng cách nhỏ để gia nhập cụm, hơn nữa, TCDGP phân chia số cụm trong mạng chưa tối ưu (1%, 2% và 3% tổng số nút mạng), điều này có thể dẫn đến việc nút CH sẽ hết năng lượng nhanh do phải nhận và truyền nhiều gói dữ liệu đến BS. Một đề xuất khác của lược đồ này như trong [54, 55] gọi là phân cụm dựa trên cây TBC. Trong TBC, các nút trong một vùng cảm biến được chia ra thành các cụm khác nhau, mỗi cụm xây dựng một cây với nút gốc là nút CH, vị trí của giả thiết BS ở tâm vùng cảm biến. Nút CH được đánh số ở mức 0 (mức gốc), nút kết nối trực tiếp với CH được đánh số là mức một. Quá trình đánh số mức cho cây cứ như vậy cho đến nút cuối cùng (nút lá trên cây); như Hình 2.4 dưới đây, nút có số mức cao nhất là các nút lá trên cây.

Hình 2.4 (a) minh họa quá trình phân chia mức trên cây với nút gốc là CH có mức bằng 0,  $PS_i$  là tập các nút cha ứng viên của nút  $i$ , nó bao gồm các nút hàng xóm của nút  $i$  có mức bằng mức của nút  $i$  trừ một.

Hình 2.4 (b) minh họa quá trình xây dựng cây cho mỗi cụm. Sau khi xây dựng cây, thuật toán bước sang giai đoạn thu thập và truyền dữ liệu, ở giai đoạn này, bắt đầu từ các nút có số mức lớn nhất (nút lá) sẽ cảm biến, truyền dữ liệu cho nút ở mức nhỏ hơn (nút cha của nó). Các nút nhận dữ liệu từ các nút con của nó, tổng hợp với dữ liệu nó cảm biến được để giảm số lượng gói dữ liệu và chuyển tiếp đến nút có số mức nhỏ hơn nữa. Khi nào nút gốc nhận được dữ liệu từ tất cả các nút thành viên trong cụm, nó sẽ tổng hợp dữ liệu và gửi về BS. Thông qua phân tích các kết quả mô phỏng cho thấy, TBC đạt hiệu quả năng lượng tốt hơn các giao thức truyền thống như LEACH, PEGASIS và TREEPSI [91]. Tuy nhiên, các nút gốc của cây tiêu tốn nhiều năng lượng vì việc chọn CH như nút gốc vẫn dựa vào xác suất ngẫu nhiên, do đó, có thể các nút gốc được chọn sẽ ở xa BS.



**Hình 2.4:** Lược đồ phân chia mức và xây dựng cây [54]

Thêm nữa, để giải quyết các vấn đề với mạng hỗn hợp bao gồm nhiều hơn hai kiểu nút cảm biến không dây được triển khai trên vùng mạng, Li Han đã đề xuất LEACH-HPR [40], đó là một giao thức lựa chọn CH hiệu quả và sử dụng thuật toán cây khung nhỏ nhất để xây dựng giao thức định tuyến liên cung. Trong đó, nút CH được chọn thông qua việc tính giá trị hàm xác suất giống với LEACH nhưng nó tương ứng cho từng kiểu nút cảm biến. Các nút khác tham gia nhập cụm dựa vào tham số cạnh tranh  $C_{joint}$ , chúng gồm năng lượng còn lại của nút ứng viên và góc tạo bởi nó và nút CH muốn ra nhập. Gần đây, đề xuất giao thức định tuyến cân bằng năng lượng dựa trên cây tự tổ chức tổng quát GSTEB [41] xây dựng cây định tuyến xử lý cho tiến trình, nơi mà ở mỗi vòng, BS sẽ tính toán, chọn nút gốc cây dựa vào mức năng lượng còn lại của nút ứng viên  $EL$  (energy level) mà các nút gửi tới,  $EL$  được tính như công thức (2.8). Sau đó BS quảng bá lựa chọn CH đến tất cả các nút khác trong mạng để cân bằng năng lượng tiêu thụ giữa các nút mạng. Các nút khác lựa chọn kết nối với nút cha của nó trên cây dựa vào  $EL$  và khoảng cách từ nó đến nút hàng xóm gần nhất:

$$EL(i) = \lceil E_{\text{residual}}(i) / \gamma \rceil \quad (2.8)$$

trong đó,  $\gamma$  là một hằng số, nó phản ánh đơn vị năng lượng nhỏ nhất và được thay đổi tùy theo yêu cầu.

**Bảng 2.3:** So sánh các thuật toán phân cụm dựa trên cây

Thuật toán xây dựng cây	Phân tán	Tập trung	Đặc tính truyền thông	Chọn CH dựa trên	Hiệu quả năng lượng	Độ phức tạp tính toán	Độ phức tạp thông báo
Phân cụm cây [54, 55]	✓		Đa chặng	Xác suất	Cao	$O(n)$	$O(1)$
Xây dựng cây [46]		✓	Đa chặng	Năng lượng và khoảng cách	Cao	$O(n \log(n))$	$O(1)$
Xây dựng cây [91]		✓	Đa chặng	Ngẫu nhiên	Cao	$O(n \log(n))$	$O(1)$
GSTEB [41]	✓		Đa chặng	Năng lượng và khoảng cách	Cao	$O(n)$	$O(1)$

Bảng 2.2 tóm tắt những điểm khác nhau chính giữa các thuật toán phân cụm dựa trên cây khung tối thiểu được khảo sát.

#### 2.2.5.5. Giao thức ngưỡng nhạy cảm năng lượng thấp

Các giao thức LEACH và PEGASIS đều hỗ trợ các ứng dụng mang tính giám sát; nghĩa là thông tin cảm biến được gửi về BS theo định kỳ. Trong nhiều ứng dụng, hệ thống giám sát chỉ cần đưa ra các cảnh báo khi có sự kiện xảy ra, ví dụ hệ thống giám sát cháy rừng, giám sát sự xâm nhập trái phép v.v. Để giảm thiểu năng lượng tiêu thụ, nút đứng đầu sẽ phải xử lý nội bộ dữ liệu trước khi gửi về BS nhằm loại bỏ

phần dư thừa, trùng lặp. Tuy nhiên, các giao thức này không đáp ứng được cho các ứng dụng dựa trên sự kiện, khi mà dữ liệu chỉ được truyền về BS nếu có một sự kiện cụ thể nào đó xảy ra. Giao thức ngưỡng nhảy cảm mạng cảm biến năng lượng hiệu quả TEEN [4, 74, 81] được xây dựng nhằm cung cấp một cơ chế truyền nhận dữ liệu dựa trên sự kiện trong mạng, nơi mà các nút liên tục quan sát, cảm biến môi trường nhưng gửi dữ liệu về BS chỉ khi sự kiện xảy ra. TEEN đạt được hiệu quả sử dụng năng lượng cao nhờ vào sự điều khiển hoạt động thông qua các giá trị ngưỡng và tổ chức các nút cảm biến thành một hệ thống phân cấp với nhiều cấp độ khác nhau giống như cấu trúc cây phân cấp. Trong kiến trúc phân cấp, dữ liệu được truyền đi từ các nút cảm biến tới nút CH. Nút CH thu thập và tổng hợp dữ liệu từ các nút thành viên trong cụm gửi tới cùng với dữ liệu của chính nó, sau đó nó gửi gói dữ liệu đó tới nút đứng đầu cấp cao hơn. Quá trình cứ tiếp tục như vậy cho tới khi dữ liệu tới được BS.

Dựa trên kiến trúc mạng phân cấp như vậy, TEEN cung cấp cơ chế truyền nhận dữ liệu thông qua hai ngưỡng là: Ngưỡng cứng (Hard Threshold- $H_T$ ) và ngưỡng mềm (Soft Threshold- $S_T$ ), chúng có thể được điều khiển và thiết lập cố định bởi người dùng. Dữ liệu từ các nút cảm biến chỉ được gửi về BS nếu giá trị mà nó cảm nhận được vượt qua ngưỡng cứng ( $H_T$ ). Tuy nhiên, nếu giá trị cảm biến này có thể đạt được trong một khoảng thời gian dài, thì dữ liệu sẽ được truyền đi một cách liên tục, điều này dẫn đến lãng phí về năng lượng. Do đó, để giảm thiểu sự truyền tải một cách dư thừa không cần thiết thì ngưỡng mềm ( $S_T$ ) được sử dụng. Bất cứ khi nào ngưỡng cứng ( $H_T$ ) bị vượt qua, các nút cảm biến sẽ kiểm tra ngưỡng mềm ( $S_T$ ) cho các lần quan sát tiếp theo. Khi đó thông tin chỉ được gửi đi nếu sự chênh lệch vượt qua ngưỡng mềm ( $S_T$ ). Giao thức TEEN đạt được hiệu quả sử dụng năng lượng do các ngưỡng được thiết lập cố định. Tuy nhiên, TEEN không phù hợp cho các ứng dụng mà dữ liệu cần thu thập theo định kỳ vì người dùng sẽ không có dữ liệu gì nếu thông tin cảm biến dưới ngưỡng. Hơn nữa, BS có thể không nhận ra các nút đã chết từ các nút còn sống vì chỉ khi nào dữ liệu cảm biến từ các nút mạng vượt quá ngưỡng cứng và có biến đổi cao hơn ngưỡng mềm thì nó mới gửi về BS. Cuối cùng là dữ liệu có thể không đến được BS do CH đã hết năng lượng. Để khắc phục các

hạn chế này, Manjeshwar và Agrawal đã đề xuất APTEEN [69, 73, 81] đây được coi là một cải tiến của TEEN. Trong APTEEN kỹ thuật TDMA được sử dụng để truyền dữ liệu trong mỗi cụm, nút CH thực hiện việc gán cho mỗi nút một khe để tránh đụng độ trong pha truyền dữ liệu trong nhóm. Mỗi nút cảm biến trong mạng sẽ gửi định kỳ thông tin mà nó cảm biến được về nút CH thông qua khe thời gian này. Ở đây, các giá trị ngưỡng cứng ( $H_T$ ) và ngưỡng mềm ( $S_T$ ) sẽ quyết định thời điểm và mức độ thường xuyên để gửi dữ liệu. APTEEN là sự kết hợp của LEACH và TEEN, do đó nó có thể hỗ trợ cho cả những ứng dụng truyền dữ liệu theo sự kiện và cả những ứng dụng truyền dữ liệu theo chu kỳ. Tuy nhiên, APTEEN không thể tránh khỏi các vấn đề mà LEACH gặp phải như truyền thông đơn chặng giữa các nút thành viên trong cụm và CH với khoảng cách xa, không đảm bảo cân bằng số nút trong các cụm. Thêm nữa, hoạt động gửi dữ liệu trong APTEEN chỉ xảy ra khi giá trị cảm biến vượt ngưỡng nên cả TEEN và APTEEN có thêm hai điểm yếu mới: Thứ nhất, khe thời gian của mỗi nút là lãng phí nếu như nó không có dữ liệu để gửi, trong khi các nút khác phải đợi đến khe thời gian của mình. Thứ hai, các nút không có cơ chế để phân biệt nút chết với nút mà giá trị cảm nhận của nó không vượt ngưỡng trong một thời gian dài.

### **2.3. Phân cụm tổng hợp dữ liệu**

Nếu như các thuật toán phân cụm dựa trên chuỗi và cây cho hiệu quả sử dụng năng lượng bằng cách giảm khoảng cách truyền thông giữa các nút trong mạng thì tổng hợp dữ liệu sẽ loại bỏ dữ liệu cảm biến dư thừa từ các nút cảm biến khác nhau trong các ứng dụng mạng cảm biến để lấy về thông tin quan sát chính xác hơn [2, 12, 60]. Thêm nữa, nén dữ liệu cũng là giải pháp tốt cho việc tiết kiệm nguồn năng lượng pin quý hiếm của các nút cảm biến không dây bằng cách giảm số bit dữ liệu mà các nút CH phải truyền đến BS [68, 71, 99]. Tuy nhiên, các phương pháp nén như Huffman; Lempel-Ziv [92] là không phù hợp với mạng cảm biến do nó yêu cầu không gian bộ nhớ lớn và thuật toán phức tạp mà các nút cảm biến không dây thông thường bị giới hạn về băng thông, bộ nhớ và khả năng xử lý, tính toán. Do đó, việc thiết kế các thuật toán tổng hợp dữ liệu đơn giản, có độ phức tạp tính toán nhỏ, sử

dụng ít bộ nhớ mà vẫn đạt được yêu cầu đặt ra là một thách thức lớn đối với các nhà nghiên cứu. Các tác giả trong [12, 17, 30] đã đề xuất phương pháp ứng dụng lý thuyết lập luận hiển nhiên Dempster-Shafer vào tổng hợp dữ liệu nhiều cảm biến bằng cách sử dụng tập luật kết hợp [20, 60]. Phương pháp này cho thấy hiệu quả cao trong việc đưa ra các kết quả dự đoán từ tập dữ liệu thô mà các nút cảm biến được. Tuy nhiên, phương pháp Dempster-Shafer tiêu tốn nhiều thời gian cho việc tính toán để đạt được kết quả khi số lượng nút cảm biến trong mạng tăng lên. Để giải quyết vấn đề này, Bin Zeng và các cộng sự [119] đã đề xuất phương pháp tổng hợp dữ liệu sử dụng phân tích ma trận gọi là LEECF. LEECF không chỉ tối ưu về chi phí năng lượng mà còn tăng tốc độ tổng hợp, tính toán cho các nút cảm biến bằng cách phân tích ma trận niềm tin. Đầu tiên, toàn bộ thông tin cảm biến từ các nút trong cụm được trình bày trong một ma trận kích thước  $n \times m$ , với  $n$  là số nút cảm biến,  $m$  là xác suất dữ liệu của nút cảm biến đo được, sau đó thực hiện tính toán trong ma trận để lấy về kết quả tổng hợp. Các tác giả cũng chứng minh được các kết quả trả về có độ tin cậy tương đương với việc áp dụng lý thuyết hiển nhiên Dempster-Shafer nhưng cho độ phức tạp tính toán nhỏ hơn.

Phương pháp mã nguồn phân tán (DSC - Distributed Source Coding) [16, 58, 84, 111] là kỹ thuật nén hiệu quả và phù hợp nhất đối với mạng cảm biến không dây, nó được đề xuất bởi Slepian-Wolf, thực hiện nén không mất dữ liệu hai nguồn dữ liệu tương quan sử dụng thông tin phụ "*side information*" [33, 97, 99]. Nó có thể là thông tin cảm biến được của nút cảm biến đã xuất hiện trong quá khứ hoặc là dữ liệu của nút hàng xóm cảm biến được. Trong [68] các tác giả đề xuất thuật toán nén dữ liệu cảm biến cho ứng dụng chăm sóc sức khỏe sử dụng DSC, thuật toán đạt được hiệu quả nén lên tới 80% ~ 100% mà không mất dữ liệu. Trong [71], Hong Luo cùng các cộng sự đã nghiên cứu tập hợp thông tin tin cậy năng lượng tối thiểu trên các tô-pô mạng khác nhau như tô-pô hình sao, tô-pô chuỗi và cây. Các tác giả cung cấp các giải pháp tối ưu để tính toán số lượng gói dữ liệu cần truyền cho mỗi nút trên các tô-pô sử dụng phương pháp Lagrange. Các chứng minh giải tích và các kết quả mô phỏng cho thấy giải pháp mà nhóm tác giả đề xuất đảm bảo thông tin tổng hợp tin cậy cho hiệu quả năng lượng khi truyền trong môi trường không dây

không tin cậy [71, 84, 106]. Các đề xuất [16], [58], đề xuất phương pháp tổng hợp dữ liệu áp dụng mã nguồn phân tán (DSC) tiết kiệm năng lượng sử dụng thông tin phụ ở bộ giải mã để loại bỏ các thông tin dư thừa (trùng lặp) trong mạng cảm biến không dây.

Gần đây, các đề xuất tổng hợp dữ liệu gồm: Phương pháp tổng hợp dữ liệu để giảm dữ liệu dư thừa từ các nút cảm biến sử dụng luật Simpson [31]. Mục đích chính của đề xuất này là quản lý hiệu quả nguồn năng lượng và đưa ra dự báo chính xác thông tin cảm biến được. Phương pháp lọc phân tán (DKF) để tổng hợp dữ liệu cảm biến trong mạng, nơi mà mỗi nút cảm biến sẽ ước lượng giá trị cảm biến và đưa vào bộ lọc phân tích để lấy về thông tin tổng hợp [1]; đề xuất tập trung nghiên cứu các kiến trúc, thuật toán tổng hợp dữ liệu đa cảm biến, chúng cho phép tiết kiệm năng lượng kéo dài thời gian sống cho mạng WSN. Hơn nữa, các dữ liệu khoa học dựa trên tập luật thông kê theo lô-gíc mờ để tổng hợp dữ liệu đa cảm biến cũng được Xiaojun Zhai cùng các cộng sự sử dụng trong dự án SWIPE (Space Wireless Sensor Networks for Planetary Exploration) khám phá bề mặt mặt trăng. Thuật toán tổng hợp dữ liệu có thể giảm dữ liệu thô trùng lặp ở nút cảm biến và tăng độ chính xác thông tin cảm biến và giảm tiêu thụ năng lượng [120].

## **2.4. Tổng kết chương**

Các giao thức định tuyến trong lược đồ phân cụm đều tổ chức mạng dưới dạng phân cấp, nhằm hạn chế tối đa các truyền thông trực tiếp với BS ở xa. Do đó, lưu lượng truyền thông với BS bị giới hạn bởi các nút đứng đầu nhóm. Điều này cho phép các mạng có quy mô lớn được triển khai mà không gặp phải tình trạng quá tải, độ trễ truyền thông ở một số điểm trọng yếu. Mặt khác, thời gian sống của toàn mạng được cải thiện rõ rệt do hoạt động mạng được tổ chức chặt chẽ.

Các giao thức phân cụm dựa theo chuỗi và cây tối thiểu cho hiệu quả sử dụng năng lượng tốt nhờ giảm tối đa khoảng cách truyền thông giữa các nút trong mạng bằng cách xây dựng liên kết các nút thành chuỗi hoặc cây khung nhiều cấp sử dụng thuật toán tham lam (GA-Greedy Algorithms) để xây dựng chuỗi [3, 42, 63, 70, 72, 116] hoặc cây [46, 54, 55]. Hơn nữa, các cơ chế phân cụm được thực thi một cách tự

động theo vòng, điều này khiến cho mức năng lượng tiêu thụ ít hơn hẳn so với các hoạt động trong các giao thức kiến trúc phẳng. Thêm nữa, hầu hết dữ liệu được tổng hợp ở các nút đứng đầu cụm thông qua các thuật toán tổng hợp dữ liệu, cho nên hạn chế được nhiều dữ liệu dư thừa gửi về BS. Ngoài ra, trong các mạng WSNs dựa trên sự kiện, khi các nút mạng không hoạt động chúng sẽ ở trạng thái ngủ dưới sự giám sát của các nút đứng đầu cụm, điều này hạn chế được tối đa năng lượng bị tiêu hao không cần thiết trên các nút này.

Tuy nhiên, nhiều thuật toán phân cụm dựa trên chuỗi và cây trên chọn nút CH ở vị trí ngẫu nhiên hay dựa vào xác suất, không xem xét đến năng lượng còn lại của nút và khoảng cách từ nó đến BS, do đó, hiệu quả sử dụng nguồn năng lượng quý hiếm chưa cao [6, 49, 51, 86, 112]. Thêm nữa, việc cố định thời gian ổn định hoạt động mạng sau khi thiết lập cụm của mỗi vòng chưa đạt hiệu quả vì nếu thời gian ổn định ngắn thì sẽ lãng phí năng lượng cho tổng chi phí xây dựng lại nhóm. Ngược lại, nếu thời gian ổn định dài rất có thể có nút CH sẽ hết năng lượng sớm, cụm đó không hoạt động. Hơn nữa, thuật toán định tuyến đề xuất xây dựng lược đồ cây mở rộng chưa kết hợp được giữa định tuyến với tổng hợp, nén dữ liệu theo mô hình cây để giảm gói dữ liệu dư thừa lấy về dữ liệu chính xác hơn gửi về BS [41, 46, 54, 55, 121].

Từ các kết quả khảo sát, phân tích và so sánh các thuật toán định tuyến phân cụm như đã được trình bày ở trên, chúng tôi đưa ra hướng tiếp cận nhằm hoàn thiện giải pháp được đề xuất với các vấn đề sau sẽ được nghiên cứu giải quyết (ở các chương tiếp theo):

- Nâng cao hiệu năng, khắc phục những hạn chế còn tồn tại của phương pháp định tuyến phân cụm truyền đơn chặng bằng cách thay đổi phương pháp phân cụm và tiêu chí chọn nút cụm trưởng.
- Phân chia cân bằng tổng số nút còn sống trong mạng vào 5% số cụm (số nút trong các cụm cân bằng nhau sẽ cho hiệu quả năng lượng tốt hơn không cân bằng)
- Giảm chi phí năng lượng trong giai đoạn thiết lập cụm bằng cách tính toán tối ưu khoảng thời gian ổn định truyền dữ liệu.



- Giảm khoảng cách truyền thông trong cụm bằng cách cải tiến thuật toán kết nối các nút trong cụm thành chuỗi và thuật toán xây dựng cụm cây.
- Giảm số bit dữ liệu truyền trong mạng dựa trên giải pháp tổng hợp, nén dữ liệu theo mô hình chuỗi hoặc cây.

### **Chương 3: ĐỊNH TUYẾN TIẾT KIỂM NĂNG LƯỢNG DỰA TRÊN PHÂN CỤM**

Định tuyến tiết kiệm năng lượng dựa trên phân cụm, sử dụng giao thức LEACH phân tán là phương pháp định tuyến hiệu quả và khả thi đối với cả mạng đồng nhất và mạng không đồng nhất. Tuy nhiên, việc lựa chọn CH dựa vào xác suất không đảm bảo được nút CH lúc nào cũng có mức năng lượng còn lại cao và có vị trí gần BS để duy trì hoạt động cụm trong khoảng thời gian dài [44, 87].

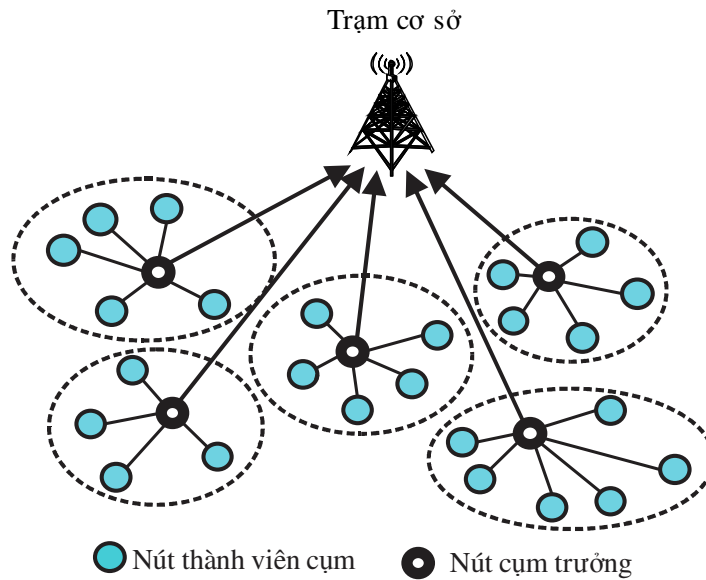
Trong chương này, một thuật toán cải tiến LEACH được chúng tôi đề xuất nhằm nâng cao hiệu quả sử dụng năng lượng và mở rộng thời gian sống của mạng cảm biến. Thuật toán này có độ phức tạp truyền thông báo và tính toán thấp cũng như có thể làm việc tốt trên cả các mạng cảm biến có mật độ nút dày đặc trên vùng quan sát. Theo đề xuất này, mỗi nút sẽ kiểm tra mức năng lượng còn lại của nó ở thời điểm hiện tại và khoảng cách từ đó đến BS. Nếu mức năng lượng còn lại của nút lớn hơn mức năng lượng còn lại trung bình của tất cả các nút còn sống trong mạng và ở gần BS thì nó có xác suất lớn để trở thành CH ở vòng hiện tại.

Phần đầu của chương này phân tích chi tiết các bước thực hiện của LEACH cũng như các ưu điểm và các vấn đề hạn chế của nó. Phần tiếp theo trình bày thuật toán được chúng tôi đề xuất, sau đó hiệu năng của thuật toán được phân tích, đánh giá và so sánh với các thuật toán đã có dựa vào các kết quả mô phỏng. Cuối cùng, một số thảo luận và hướng phát triển tiếp theo của thuật toán cũng được đề cập ở cuối chương này.

#### **3.1. Kỹ thuật định tuyến phân cụm phân tán**

Trong kỹ thuật định tuyến phân cụm phân tán như giao thức LEACH, các hoạt động phân cụm mạng như bầu nút làm cụm trưởng, chọn nút CH để gia nhập cụm cùng với hoạt động truyền dữ liệu được thực hiện thông qua thông điệp điều khiển ở giai đoạn đầu của mỗi vòng. Sau đó các cụm đi vào giai đoạn ổn định, việc truyền dữ liệu được thực hiện định kỳ. Ưu điểm của phương pháp này là ở mỗi vòng thông

điện đều được gửi trong khoảng cách gần (các nút hàng xóm), tải tiêu thụ năng lượng được chia đều cho các nút trong mạng và chỉ có nút CH gửi dữ liệu đến BS, do đó hiệu quả năng lượng tốt hơn các giao thức theo lược đồ phẳng. Hoạt động của LEACH được chia ra thành các vòng, mỗi vòng bao gồm hai giai đoạn: Giai đoạn thiết lập và giai đoạn ổn định truyền dữ liệu như được mô tả dưới đây:



**Hình 3.1:** Tô-pô cơ bản của giao thức LEACH trong một vòng

Hình 3.1 minh họa tô-pô cơ bản của giao thức LEACH sau khi thực hiện phân cụm và chọn nút CH trong một vòng.

### **Giai đoạn thiết lập:**

#### **Bước 1: Chọn nút cụm trưởng**

Các nút trong mạng sẽ tự mình quyết định có trở thành nút đứng đầu cụm hay không theo quy tắc như sau: Mỗi nút chọn một số ngẫu nhiên  $P$  nằm trong khoảng từ 0 tới 1. Nếu số ngẫu nhiên này nhỏ hơn giá trị ngưỡng  $T(i)$  thì nút đó sẽ trở thành nút cụm trưởng ở vòng hiện tại. Giá trị ngưỡng được tính toán dựa trên biểu thức toán học sau [44, 113]:

$$T(i) = \begin{cases} \frac{k}{1 - k(r \bmod \frac{1}{k})} & , \text{if } i \in G \\ 0 & \text{otherwise } (i \notin G) \end{cases} \quad (3.1)$$

Trong đó:

- k là tỉ lệ phần trăm nút cụm trưởng trong mạng (k=5% [44])
- r là vòng hiện tại
- G là tập hợp các nút không được lựa chọn làm CH trong (1/k) vòng cuối.

### **Bước 2: Xây dựng cụm**

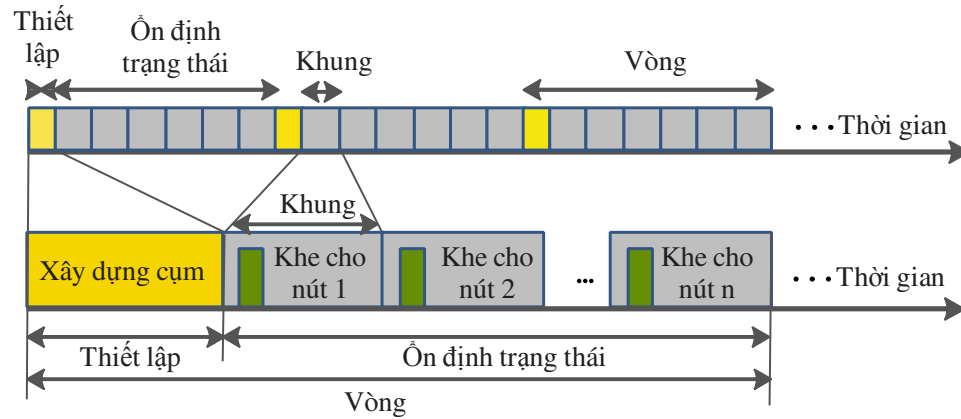
Sau khi chọn được nút CH, các nút CH sẽ quảng bá vai trò mới của mình cho các nút khác trong mạng. Các nút khác (không phải là CH) nhận thông điệp quảng bá và sẽ quyết định gia nhập một cụm nào đó dựa vào việc tính độ mạnh tín hiệu RSSI nhận được từ các thông điệp quảng bá mà các CH đã gửi, và gửi thông điệp gia nhập cụm tới CH tương ứng [5, 61].

### **Bước 3: Tạo bộ lập lịch TDMA và CDMA**

Sau khi mạng được tổ chức thành các cụm, các CHs tạo khe thời gian TDMA cho các nút thành viên trong cụm để truyền dữ liệu trong cụm và chọn một mã CDMA để nó truyền dữ liệu tới BS, việc làm này giúp tránh được đụng độ xảy ra trong giai đoạn truyền dữ liệu bên trong cụm và liên cụm.

### **Sang giai đoạn ổn định trạng thái**

Các nút mạng bắt đầu cảm biến và truyền dữ liệu về nút CH của cụm mình thông qua khe thời gian TDMA đã được cấp phát. Việc thu thập dữ liệu diễn ra theo chu kỳ. Các nút CH ngoài việc thu nhận dữ liệu từ các nút thành viên gửi tới, nó còn phải tổng hợp dữ liệu để giảm thiểu số lượng gói dữ liệu trước khi gửi tới BS. Sau một khoảng thời gian nhất định nào đó được xác định trước, mạng sẽ quay trở lại giai đoạn thiết lập và bắt đầu một vòng mới bằng cách lựa chọn lại các nút CH và xây dựng lại cụm. Hình 3.2 minh họa hoạt động trong giai đoạn thiết lập cụm và giai đoạn ổn định trạng thái truyền dữ liệu trong một vòng của thuật toán LEACH.



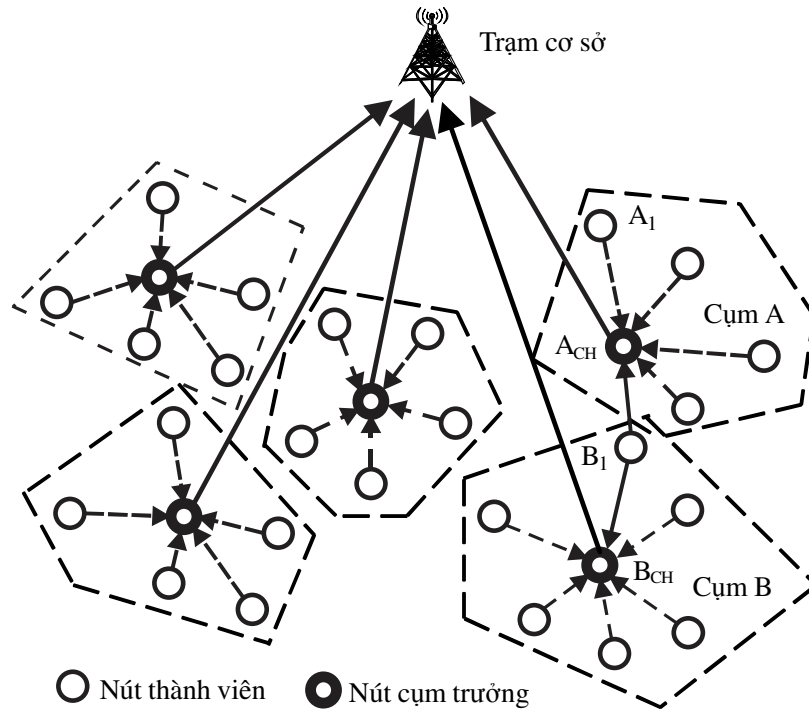
**Hình 3.2:** Hoạt động hai giai đoạn trong một vòng của LEACH

### 3.2. Đề xuất cải tiến giao thức LEACH

#### *Các hạn chế của LEACH*

Mặc dù giao thức định tuyến LEACH có nhiều ưu điểm cho mạng WSNs trong việc tự tổ chức hoạt động như chọn nút CH, xây dựng cụm phân tán. Tuy nhiên, việc chọn nút CH chỉ xem xét gián tiếp đến năng lượng còn lại của nút ứng viên có thể dẫn đến nút CH được chọn có mức năng lượng còn lại thấp. Thêm nữa, vị trí địa lý của nút cũng ảnh hưởng đáng kể đến hoạt động cụm vì CH phải chuyển tiếp gói tin đến BS ở xa. Do đó, cụm sẽ không có tác dụng nếu nút CH hết năng lượng sớm. Hình 3.3 minh họa hoạt động mạng sau khi được tổ chức thành các cụm, nơi mà các nút CH ở xa BS. Do vậy, trong một số cụm, truyền thông từ các nút thành viên tới các CHs xảy ra ngược chiều tới BS. Trong LEACH, để tham gia nhập cụm, các nút dựa vào độ mạnh tín hiệu để quyết định việc tham gia. Ví dụ trong Hình 3.3, nút  $B_1$  sẽ gia nhập nút cụm trưởng  $B_{CH}$  để truyền dữ liệu. Tuy nhiên, nếu nút  $B_1$  gia nhập cụm có nút cụm trưởng là  $A_{CH}$  thì hiệu quả năng lượng cho nút  $B_{CH}$  và cho tổng thể mạng sẽ cao hơn gia nhập cụm  $B_{CH}$  vì nút  $B_1$  truyền ngược hướng với nút  $B_{CH}$  tới BS (nút  $A_{CH}$  sẽ truyền nối tiếp theo hướng từ  $B_1$  tới BS).

Để giải quyết vấn đề được đưa ra ở trên, trong phần nội dung tiếp theo, luận án giới thiệu đề xuất LEACH-DE của chúng tôi như dưới đây:



**Hình 3.3:** Mô hình phân cụm của LEACH

#### ***Bước chọn nút làm cụm trưởng***

Các nút tính giá trị ngưỡng  $V(i)$  theo công thức (3.2) thay vì  $T(i)$  theo công thức (3.1) như LEACH ở vòng hiện tại để xem xét trở thành nút cụm trưởng. Giá trị  $V(i)$  tỉ lệ thuận với năng lượng còn lại và tỉ lệ nghịch với khoảng cách từ nút  $i$  đến BS để chọn ra nút có mức năng lượng còn lại lớn và ở gần BS làm CH, do đó cho hiệu quả năng lượng tốt.

$$V(i) = c \times \frac{E_{\text{residual}}(i)}{E_{\text{average}}} \times \frac{D_{\text{Max}}}{D(i, \text{BS})} \quad (3.2)$$

Trong đó: -  $E_{\text{residual}}(i)$  là mức năng lượng còn lại của nút  $i$  ở vòng hiện tại

-  $D(i, \text{BS})$  là khoảng cách địa lý từ nút  $i$  đến BS

-  $c$  là một hằng số dùng để giới hạn giá trị  $V(i)$  lớn hơn 0 và nhỏ hơn 1 (với kịch bản trong mô phỏng như Bảng 3.1, tác giả dùng  $c=0.09$ )

-  $D_{\text{Max}}$  là giá trị đường kính mạng, được tính sau khi triển khai mạng

-  $E_{\text{average}}$  chỉ định mức năng lượng trung bình của các nút cảm biến còn sống trong vòng hiện tại, nó được tính như dưới đây:

$$E_{\text{average}} = \frac{1}{n} \sum_{i=1}^n E_{\text{residual}}(i) \quad (3.3)$$

Trong đó,  $n$  là tổng số nút còn sống ở vòng hiện tại. Ở vòng đầu tiên, năng lượng còn lại chính là năng lượng khởi tạo của các nút và công thức (3.2) sẽ chọn nút có khoảng cách gần BS nhất. Từ vòng tiếp theo trở đi, ở chu kỳ truyền dữ liệu cuối của mỗi vòng, các nút sẽ gửi mức năng lượng còn lại của mình đến BS theo tuyến đường truyền dữ liệu. BS nhận thông điệp chứa năng lượng còn lại của các nút còn sống, tính mức năng lượng trung bình và quảng bá tới các nút mạng ở giai đoạn đầu của vòng tiếp theo.

### ***Bước xây dựng nhóm***

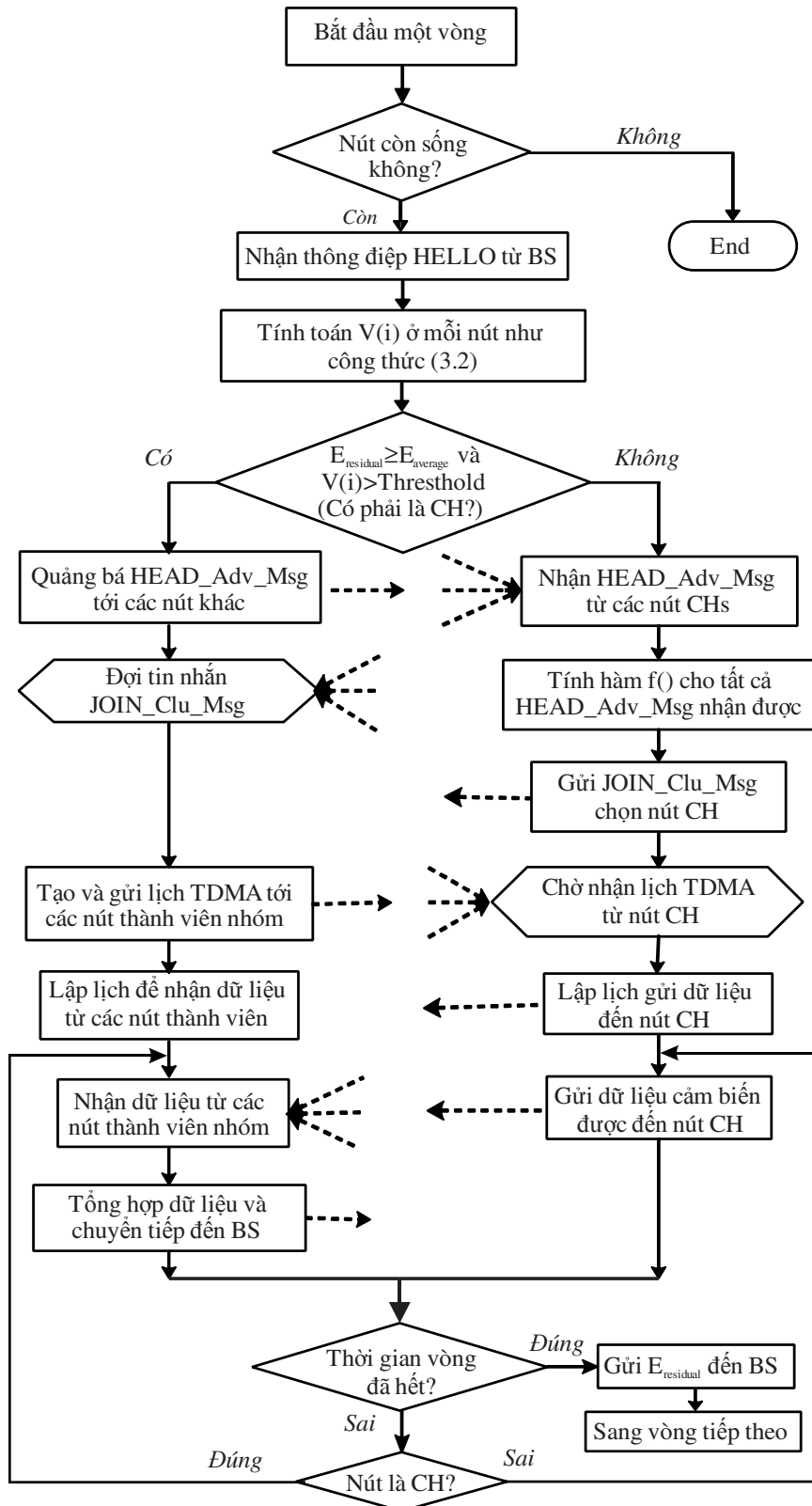
Sau khi chọn được các nút CH, các CH gửi quảng bá thông điệp HEAD\_Adv\_Msg đến các nút khác trong mạng. Các nút không phải CHs nhận thông điệp HEAD\_Adv\_Msg từ tất cả các CHs và ước lượng khoảng cách, năng lượng còn lại của các nút CHs, chọn CH có hàm  $f_{\text{criterion}}(i, CH_j)$  lớn nhất theo công thức (3.4) và gửi thông điệp gia nhập cụm JOIN\_Clu\_Msg đến nút CH tương ứng.

$$f_{\text{criterion}}(i, CH_j) = \text{Max}_i \left\{ \frac{E_{\text{residual}}(j)}{d(\text{BS}, CH_j) + d(i, CH_j)} \right\} \quad (3.4)$$

Trong đó,  $d(\text{BS}, CH_j)$  và  $d(i, CH_j)$  là khoảng cách từ BS đến  $CH_j$  và khoảng cách từ nút  $i$  đến  $CH_j$ . Khoảng cách địa lý giữa hai nút  $a$  và  $b$  được tính toán như dưới đây:

$$|d(a, b)| = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (3.5)$$

Quan sát công thức (3.4) ta thấy, giá trị hàm  $f_{\text{criterion}}$  tiến đến cực đại khi giá trị  $E_{\text{residual}}$  cực đại và  $d(CH_j, \text{BS}) + d(i, CH_j)$  cực tiểu, điều này cho phép chọn nút có mức năng lượng còn lại cao nhất và khoảng cách nhỏ nhất làm cụm trưởng. Các CH tạo và gửi lịch TDMA tới các nút thành viên trong cụm để nhận dữ liệu, các nút thành viên chờ nhận lịch TDMA từ nút CH để gửi dữ liệu. Còn các CH sử dụng mã CDMA để gửi dữ liệu đến BS, từ đó tránh đụng độ giữa truyền thông trong cụm và truyền thông với BS. Sơ đồ thuật toán của LEACH-DE được minh họa trong Hình 3.4 dưới đây.



**Hình 3.4:** Sơ đồ hoạt động của giao thức LEACH-DE



### Mô hình năng lượng

Theo mô hình năng lượng được thảo luận như trong [44, 75, 96, 109] cho tiêu thụ năng lượng truyền thông sóng vô tuyến, để truyền  $q$  bit dữ liệu giữa hai nút ở khoảng cách  $d(a, b)$ , năng lượng tiêu thụ được tính như dưới đây:

$$E_{TX}(q, d) = \begin{cases} q \times E_{elec} + q \times E_{friis} \times d^2 & , \text{if } d < d_{crossover} \\ q \times E_{elec} + q \times E_{amp} \times d^4 & , \text{if } d \geq d_{crossover} \end{cases} \quad (3.6)$$

Ở đây:

- $E_{elec}$  là năng lượng yêu cầu để chạy mạch điện cho bộ thu phát sóng vô tuyến
- $E_{friis}$  và  $E_{amp}$  là đơn vị năng lượng yêu cầu cho bộ khuếch đại trong mô hình truyền thẳng (free space) và hai tia mặt đất (two-ray ground), nó phụ thuộc vào khoảng cách và mô hình truyền thông không dây.
- $d_{crossover}$  là đường kính vùng phủ sóng của nút cảm biến, được tính trong NS2 cho các kịch bản mô phỏng như dưới đây [11, 47, 75]:

$$d_{crossover} = \sqrt{\frac{(4\pi)^2 \times l \times h_t^2 \times h_r^2}{\lambda^2}} \approx \sqrt{\frac{E_{friis}}{E_{tworay}}} \quad (3.7)$$

Trong đó,  $\lambda$  là bước sóng của sóng mang;  $l$  là hệ số suy giảm của hệ thống,  $h_t$  và  $h_r$  là chiều cao của ăng ten phát và ăng ten thu. Các tham số giá trị cho năng lượng được sử dụng cho các kịch bản mô phỏng được đề xuất như trong Bảng 3.1 và  $l = 1$ ;  $h_t = h_r = 1,5(m)$ ;  $\lambda = 0,328227(m)$ ;  $d_{crossover} = 86,1424 (m)$  [75, 125, 126].

Để nhận  $q$ -bit dữ liệu, năng lượng tiêu thụ là:

$$E_{RX}(q) = q \times E_{elec} \quad (3.8)$$

Do đó, năng lượng tiêu thụ của nút cụm trưởng (CH) thứ  $i$  cho một vòng là:

$$E_{CH} = \sum_{l=1}^{nn-1} E_{RX}(q) + E_{DA}(q) + E_S(q) + E_{TX}(q, d(CH, BS)) \quad (3.9)$$

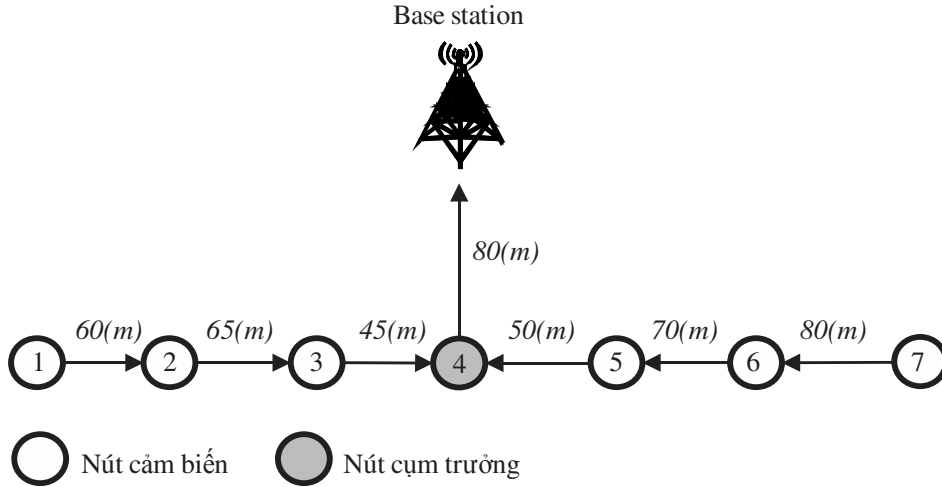
Trong đó,  $nn$  là số nút thành viên cụm của CH,  $E_S(q)$  và  $E_{DA}(q)$  là năng lượng tiêu thụ cho việc cảm biến, đo và tổng hợp  $q$  bit dữ liệu.

Năng lượng tiêu thụ của một nút thành viên thứ  $j$  trong cụm  $i$  cho một vòng là:

$$E_{Enon-CH} = E_S(q) + E_{TX}(q, d(j, CH_i)) \quad (3.10)$$

**Ví dụ 3.1. Minh họa tiêu thụ năng lượng truyền dữ liệu theo mô hình (3.6)**

Hình 3.5 minh họa năng lượng tiêu thụ khi truyền 1 bit dữ liệu từ nút cảm biến 1 đến BS theo Công thức (3.6) và (3.8) với các tham số được cho trong Bảng 3.1 cho hai trường hợp truyền đơn chặng và truyền đa chặng.



**Hình 3.5:** Sơ đồ mạng cảm biến không dây gồm 7 nút

- **Trường hợp 1:** Truyền đơn chặng (nút 1 sẽ truyền dữ liệu trực tiếp đến BS), ta giả sử khoảng cách từ nút cảm biến 1 và BS là 250 (m)  $d > d_{\text{crossover}} = 86(\text{m})$ . Theo (3.6) ta có:

$$E_{\text{TX}} = 1*50000 + 1*0,013*250*250*250*250 = \mathbf{50.831.250 \text{ (pJ)}}.$$

- **Trường hợp 2:** Truyền đa chặng theo cụm. Giả sử nút cảm biến 4 là nút cụm trưởng (nút 1 truyền dữ liệu trực tiếp đến nút 4 sau đó nút 4 truyền đến BS), khoảng cách từ nút 1 đến nút 4 là  $d = 170 \text{ (m)}$ , khoảng cách từ nút 4 đến BS là 80 (m).

$$E_{\text{TX}} = 2*50000 + 0.013*(170*170*170*170) + 10*(80*80) = \mathbf{11.021.730 \text{ (pJ)}}.$$

- **Trường hợp 3:** Truyền đa chặng theo chuỗi (nút 1 sẽ truyền dữ liệu qua các nút hàng xóm 2,3,4 đến BS), khoảng cách giữa các chặng  $d < d_{\text{crossover}} = 86(\text{m})$ .

Theo (3.6) ta có:

$$E_{\text{TX}} = 4*50000 + 10*(60*60) + 10*(65*65) + 10*(45*45) + 10*(80*80), \text{ và cuối cùng, ta có: } E_{\text{TX}} = \mathbf{362.500 \text{ (pJ)}}.$$

Như vậy, việc truyền dữ liệu đa chặng có phân cụm giữa hai nút cảm biến không dây dựa vào mô hình tiêu thụ năng lượng như (3.6) cho hiệu quả sử dụng năng lượng cao hơn truyền trực tiếp. Hơn nữa, các kết quả mô phỏng như trong các đề xuất [44, 65, 116] cho ta thấy rõ hơn vấn đề này.

### ***Các tham số mô phỏng***

Giao thức LEACH-DE đã được chúng tôi cài đặt và mô phỏng trên bộ công cụ mã nguồn mở NS2 để đánh giá hiệu năng và thời gian sống của các nút mạng với thuật toán được đề xuất [125, 126]. Các tham số mô phỏng được chúng tôi thiết lập sử dụng trong các kịch bản mô phỏng với các tham số như Bảng 3.1, chúng cũng được sử dụng trong các nghiên cứu [34, 65, 76, 110].

**Bảng 3.1:** Các tham số môi trường mô phỏng

<b>Thứ tự</b>	<b>Mô tả các tham số</b>	<b>Giá trị</b>
1	Vùng cảm biến mô phỏng	100m × 100m
2	Số nút mạng	100 (nút)
3	$E_{amp}$	0,013 pJ/bit/m <sup>4</sup>
4	$E_{init}$ (Năng lượng khởi tạo)	2 J
5	$E_{friis}$	10 pJ/bit/m <sup>2</sup>
6	$E_{elec}$	50 nJ/bit
7	$E_{DA}$	5 nJ/bit/message
8	$E_S$	0 J/bit
9	Mô hình năng lượng	Battery
10	Kích thước gói dữ liệu	1024 bytes
11	Thời gian mô phỏng	3600s
12	Vị trí của trạm cơ sở	(49,175)
13	Kiểu kênh truyền	Channel/wireless channel
14	Mô hình ăng ten	Antenna/omniantenna
15	Số lượng cụm	k = 5
16	Băng thông kênh truyền	1(Mpbs)

**Các độ đo hiệu năng được sử dụng để đánh giá hiệu quả tiêu thụ năng lượng**

**Tỷ lệ nút cảm biến bị chết theo thời gian:** Phần trăm số nút mạng cảm biến tiêu hao hết năng lượng và ngừng hoạt động theo thời gian mô phỏng.

**Tỷ lệ nút cảm biến còn sống theo thời gian:** Phần trăm số nút mạng cảm biến chưa tiêu hao hết năng lượng vẫn còn hoạt động tính theo thời gian mô phỏng (dựa vào tỉ lệ nút cảm biến bị chết theo thời gian, lấy 100% trừ đi tỉ lệ đó).

**Thời gian sống của mạng cảm biến:** Là khoảng thời gian từ khi mạng bắt đầu hoạt động cho đến khi trong mạng có 50% nút đầu tiên cạn kiệt năng lượng, hoặc nút cuối cùng trong mạng tiêu hao hết năng lượng [7, 29, 46, 78].

**Tổng năng lượng tiêu thụ của mạng cảm biến theo thời gian:** Tổng số năng lượng của tất cả các nút cảm biến trong mạng đã tiêu hao tính đến thời điểm hiện tại.

**Tỉ lệ gói tin nhận được ở BS:** Phần trăm số gói tin nhận được ở nút trung chuyển BS trên khoảng thời gian mạng hoạt động.

### **3.3. Mô phỏng để đánh giá hiệu quả của đề xuất cải tiến giao thức LEACH**

Về độ phức tạp tính toán của LEACH-DE, mỗi nút chỉ cần tính giá trị hàm  $V(i)$  và  $f()$  theo công thức (3.2) và (3.4), do đó độ phức tạp tính toán là  $O(n)$ . Độ phức tạp thông báo, mỗi nút phải thu nhận  $k$  thông điệp từ  $k$  nút cụm trưởng để tính hàm  $f()$  và gửi, nhận thêm hai thông điệp từ BS và CH, độ phức tạp thông báo là  $O(k)$ . Do đó LEACH-DE có độ phức tạp tính toán và thông báo là  $O(n)$ .

Về vấn đề thực nghiệm để kiểm chứng tính đúng của thuật toán, chúng tôi đã tiến hành chạy mô phỏng các giao thức LEACH, LEACH-C và LEACH-DE trên nhiều kịch bản khác nhau với các tham số mô phỏng như trong Bảng 3.1. Tham số khác nhau của nhiều kịch bản này là vị trí của 100 nút cảm biến được sinh ngẫu nhiên trong vùng mô phỏng  $100m \times 100m$ . Chúng tôi đã sử dụng lệnh "setdest" trong ns2 để sinh ngẫu nhiên nhiều lần cho nhiều kịch bản khác nhau với số nút  $N=100$ . Để xác định số kịch bản ( $n_{sc}$ ) cần phải chạy mô phỏng bao nhiêu là đủ, chúng tôi thực hiện theo các bước sau:

**Bước 1:** Khởi tạo, sinh ngẫu nhiên 100 lần cho 100 kịch bản khác nhau với số nút cảm biến  $N=100$  nút trên vùng mô phỏng  $100m \times 100m$ , 100 kịch bản này được bỏ đi phần di chuyển vì chúng tôi giả thiết mạng cảm biến sau khi được triển khai ở trạng thái tĩnh.

**Bước 2:** Chạy mô phỏng các giao thức (ở Chương 3 là giao thức LEACH-C, LEACH và LEACH-DE) trên kịch bản đầu tiên (kịch bản 1), các kết quả thu được sau khi mô phỏng trình bày trên bảng chỉ phần trăm nút chết, tổng năng lượng tiêu thụ và tỷ lệ gói tin nhận được ở BS.

**Bước 3:** Chọn một độ đo hiệu năng, ở đây chúng tôi chọn độ đo “Tỉ lệ nút còn sống”, để đánh giá.

**Bước 4:** Chạy kịch bản tiếp theo (kịch bản  $i$ ,  $i \geq 2$ ), trình bày các kết quả thu được sau khi mô phỏng trên bảng chỉ phần trăm nút chết, năng lượng tiêu thụ và tỷ lệ gói tin nhận được ở BS.

**Bước 5:** Tính giá trị trung bình  $m_x$ , độ lệch chuẩn  $\sigma$  và tỉ lệ độ lệch chuẩn  $\xi$  theo các công thức (3.11), (3.12) và (3.13) trên bảng các kết quả thu được sau khi chạy từ kịch bản thứ hai trở đi.

**Bước 6:** So sánh các kết quả thu được với kết quả của lần trước đó. Nếu chúng ta có được tỷ lệ độ lệch chuẩn không quá  $\xi\%$  thì dừng mô phỏng và chuyển sang Bước 7, vì nếu có chạy mô phỏng thêm nữa thì tỷ lệ độ lệch chuẩn ( $\xi$ ) cũng không thay đổi. Ngược lại, quay lên Bước 4 để chạy kịch bản thứ  $(i+1)$  tiếp theo.

**Bước 7:** Vẽ đồ thị biểu diễn các kết quả mô phỏng dựa trên bảng giá trị trung bình và độ lệch chuẩn của  $n_{sc}$  kịch bản để đánh giá hiệu năng của giao thức (LEACH, LEACH-C và LEACH-DE) theo các độ đo: Tỉ lệ (phần trăm) nút còn sống theo thời gian mô phỏng; năng lượng tiêu thụ theo thời gian mô phỏng và tỉ lệ gói tin nhận được ở BS khi vị trí BS thay đổi [6, 75, 76, 88].

Ở đây, chúng tôi chọn giá trị  $\xi$  với giao thức LEACH bằng 7,7%, LEACH-C là 4,8% và LEACH-DE là 6,6% tại thời điểm mạng có 95% nút chết thì số kịch bản mô phỏng như sau: Với giao thức LEACH, số kịch bản đã chạy  $n_{sc} = 11$  kịch bản,

với giao thức LEACH-C số kích bản đã chạy  $n_{sc}= 8$  và LEADH-DE số kích bản đã chạy  $n_{sc}= 34$ .

Công thức tính giá trị trung bình  $m_X$ , độ lệch chuẩn  $\sigma$  và tỉ lệ độ lệch chuẩn  $\xi$  như dưới đây:

$$m_X = \frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} x_i \quad (3.11)$$

$$\sigma_i = \sqrt{\frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} (m_X - x_i)^2} \quad (3.12)$$

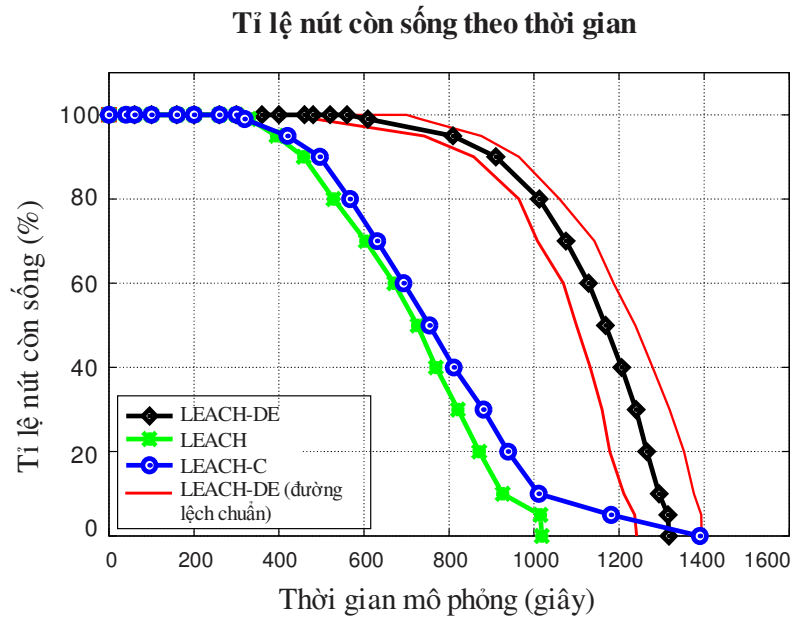
$$\xi = \sigma_i / \max(x_i) \quad \text{với } i=1 \text{ đến } n_{sc} \quad (3.13)$$

Trong đó: -  $n_{sc}$  là số kích bản khác nhau đã chạy mô phỏng.

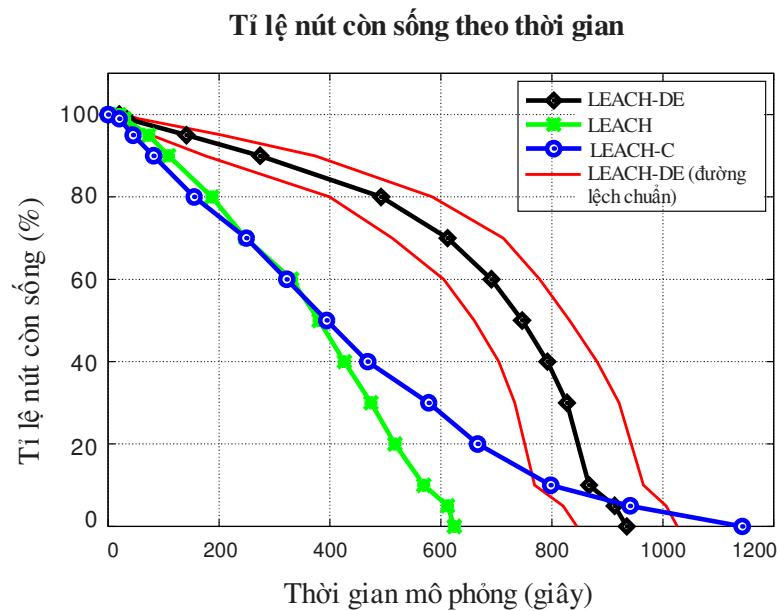
-  $x_i$  là giá trị cần tính trung bình trong kích bản thứ  $i$ . Ở đây chúng tôi tính trung bình thời gian và độ lệch chuẩn của  $n_{sc}$  kích bản tại thời điểm mà trong mạng có 1%, 10%, 20%, ... 90%, 95% và 100% nút mạng chết với các giao thức LEACH, LEACH-C và LEACH-DE. Tiếp theo, chúng tôi cũng tính giá trị trung bình và độ lệch chuẩn cho tổng năng lượng tiêu thụ và tỉ lệ gói tin nhận được ở BS. Các số liệu thu được từ mô phỏng sau khi tính giá trị trung bình được chúng tôi trình bày bằng đồ thị trên Hình 3.6; các nút trong mạng cảm biến chạy theo giao thức LEACH-DE có thời gian sống lâu hơn LEACH và LEACH-C. Trong Hình 3.6, trục Y biểu diễn tỉ lệ nút cảm biến còn sống (theo phần trăm) và trục X biểu diễn thời gian mô phỏng (giờ). Dựa vào kết quả mô phỏng trong Hình 3.6 cho thấy tỉ lệ (phần trăm) nút còn sống theo thời gian của LEACH-DE tăng lên khoảng 20% và tỉ lệ nút chết theo thời gian giảm xuống khoảng 20% so với giao thức LEACH đã có. Các đường màu đỏ biểu diễn độ lệch chuẩn của giá trị trung bình theo giao thức LEACH-DE có  $\xi$  bằng 6,6% tương ứng với số kích bản  $n_{sc}= 34$ , điều này có nghĩa là nếu chúng ta có tăng thêm mô phỏng thì kết quả thu được cũng không vượt ra ngoài đường lệch chuẩn.

Trong Hình 3.6, trục Y biểu diễn tỉ lệ nút cảm biến còn sống (theo phần trăm) và trục X biểu diễn thời gian sống của các nút trong mạng cảm biến theo thời gian mô phỏng (giờ). Theo Hình 3.7, khi vị trí của BS là ở vị trí (49,175) giao thức LEACH-C là tốt hơn LEACH về thời gian sống của mạng, nhưng khi di chuyển BS

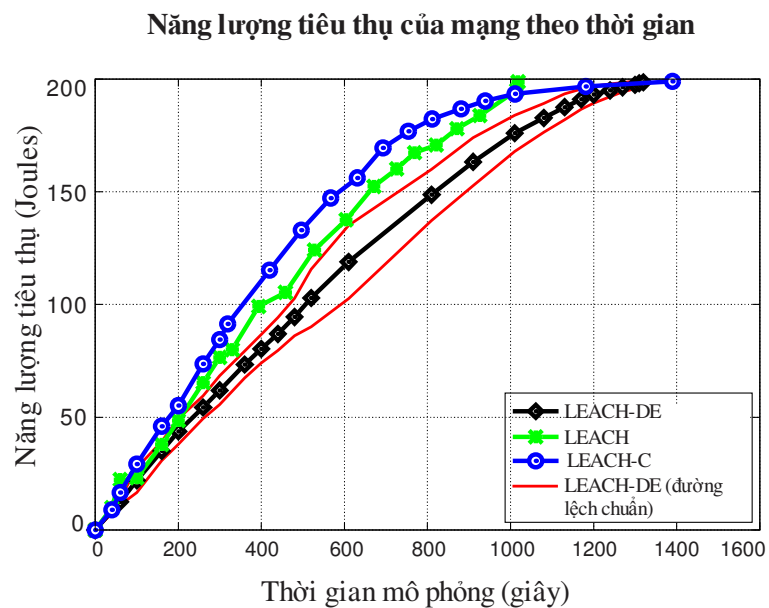
đến vị trí (49, 225) hoặc xa hơn, giao thức LEACH cân bằng năng lượng tiêu thụ tốt hơn LEACH-C [110]. Khi BS ở xa, các nút CH trong mạng phải gửi gói dữ liệu đến BS với khoảng cách xa mà theo Công thức (3.6), năng lượng tiêu thụ tăng thêm phần năng lượng khuếch đại nhân với khoảng cách mũ bốn vì khi đó  $d > d_{\text{crossover}}$ , do đó, thời gian sống của mạng giảm đáng kể.



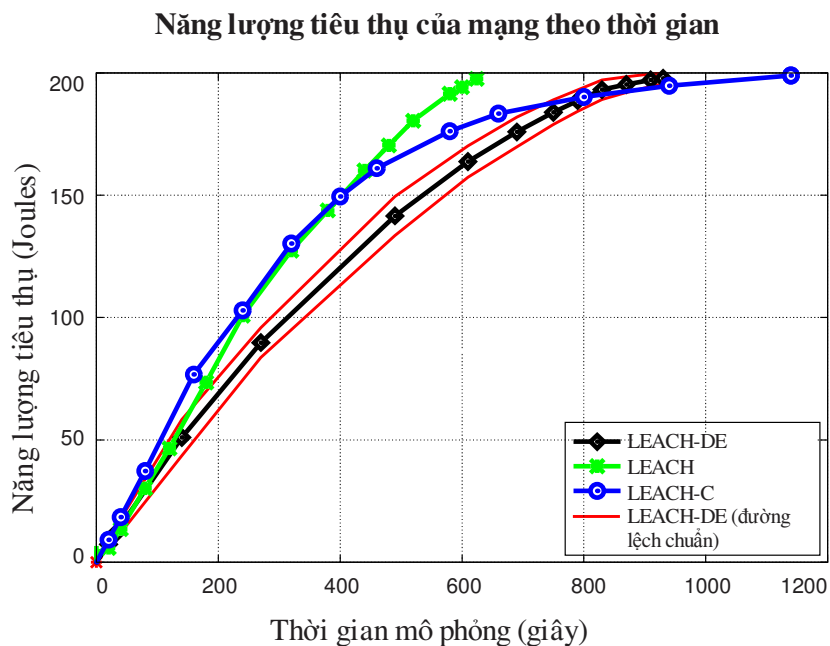
**Hình 3.6:** Tỉ lệ nút còn sống giảm theo thời gian; vị trí BS ở (49,175)



**Hình 3.7:** Tỷ lệ nút còn sống giảm theo thời gian; vị trí BS ở (49,225)



**Hình 3.8:** Tổng năng lượng mạng tiêu thụ áp dụng với ba giao thức; vị trí BS ở (49,175)

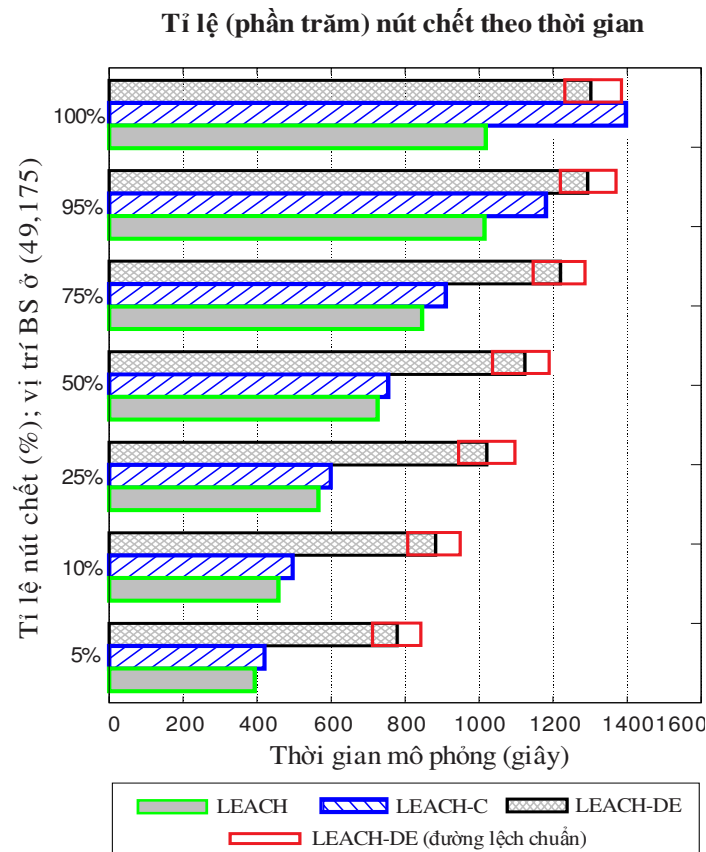


**Hình 3.9:** Tổng năng lượng mạng tiêu thụ áp dụng với ba giao thức; vị trí BS ở (49,225)



Hình 3.8 và 3.9 so sánh tổng năng lượng mạng tiêu thụ khi chạy ba giao thức trong suốt thời gian mô phỏng. Năng lượng tiêu thụ của các nút dựa trên các hoạt động như cảm biến, xử lý dữ liệu và truyền thông. Qua các hình vẽ biểu diễn kết quả, có thể thấy rõ ràng, giao thức LEACH-DE tiêu thụ năng lượng ít hơn so với giao thức LEACH hoặc LEACH-C.

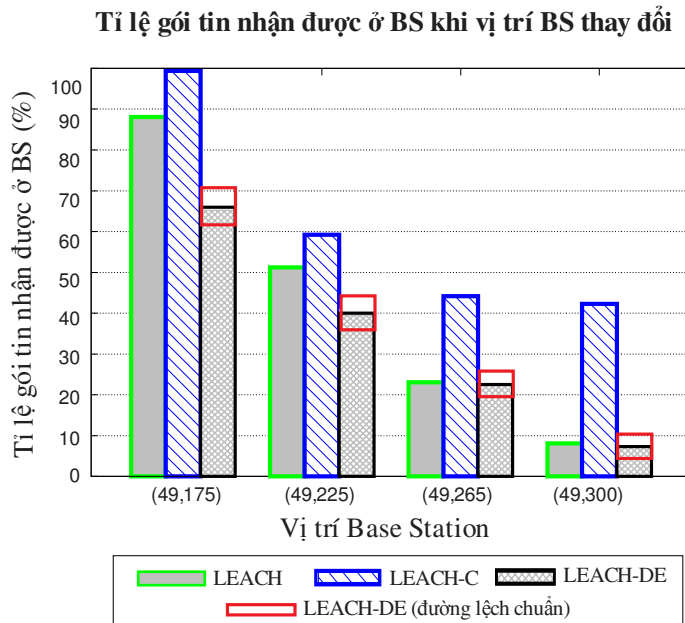
Hình 3.10 thể hiện tỉ lệ (phần trăm) nút chết trong suốt thời gian mô phỏng (ngược lại với Hình 3.6). Qua các hình vẽ biểu diễn kết quả, có thể thấy rõ ràng tỉ lệ nút chết tăng dần theo thời gian với cả ba giao thức. Tuy nhiên, LEACH-DE vẫn đạt được hiệu quả năng lượng tốt hơn và thời gian sống của mạng cảm biến dài hơn LEACH và LEACH-C.



**Hình 3.10:** Tỉ lệ (phần trăm) nút chết theo thời gian

Hình 3.11 hiển thị giá trị trung bình tỉ lệ gói tin nhận được tại BS khi chúng tôi di chuyển BS từ vị trí gần nhất (49, 175) đến vị trí (49, 300) cách xa vùng cảm biến

100m × 100m. Qua các hình vẽ biểu diễn kết quả, có thể thấy rõ ràng, có mối quan hệ giảm dần về tỉ lệ gói tin nhận được ở BS giữa LEACH-DE, LEACH-C và LEACH khi chúng tôi di chuyển BS ra xa vùng cảm biến hơn vì khi BS ở xa, chi phí truyền thông (gửi gói tin đến BS) ở khoảng cách xa cao hơn nhiều so với khoảng cách gần, do đó, các nút mạng sẽ hết năng lượng nhanh hơn. Tuy nhiên, với giao thức LEACH-C, số lượng thông điệp nhận được vẫn cao hơn giao thức LEACH và LEACH-DE. Điều này cho thấy giao thức định tuyến phân cụm tập trung, việc phân cụm được thực hiện bởi nút không bị ràng buộc về năng lượng BS, cho tỉ lệ gói tin nhận được ở BS tốt hơn giao thức định tuyến phân tán.



**Hình 3.11:** Tỉ lệ (phần trăm) gói tin nhận được ở BS

### 3.4. Phân tích và so sánh với các thuật toán cùng hướng khác

Các thuật toán phân cụm trong [6, 40, 75, 112, 114, 115], chọn CH theo xác suất kết hợp với năng lượng của nút ứng viên, không xem xét đến khoảng cách từ nó đến BS, do đó hiệu quả sử dụng năng lượng chưa cao vì CH không đảm bảo được lúc nào cũng ở gần BS. Với đề xuất LEACH-DE của chúng tôi, mỗi nút ứng viên được chọn làm CH ở vòng thứ  $i$  phải có năng lượng còn lại lớn hơn mức năng lượng trung bình của tất cả các nút còn sống trong mạng và xem xét đến khoảng cách từ

nó đến BS. Hơn nữa ở giai đoạn xây dựng cụm, trong các đề xuất trên, các nút chủ yếu dựa vào độ mạnh tín hiệu nhận được từ các CHs quảng bá đến để quyết định cụm tham gia. Do đó, khó tránh khỏi các nhược điểm mà LEACH gặp phải như trình bày trong Mục 3.2. Trang 53. Với LEACH-DE, các nút quyết định tham gia cụm dựa vào hàm ước lượng khoảng cách và năng lượng còn lại của các CHs quảng bá tới do đó nâng cao hiệu quả sử dụng năng lượng.

### 3.5. Tổng kết chương

Trong các kỹ thuật định tuyến, phân cụm là cách tiếp cận tốt cho mạng cảm biến không dây do điều kiện các nút cảm biến hạn chế về nguồn tài nguyên. Trong các thuật toán định tuyến phân cụm đã được đề xuất như LEACH [112, 113], LEACH-C [53, 75], EECS [114], HEED [105, 115], CEEC [13] đều cho hiệu quả và khả thi về việc sử dụng năng lượng. Tuy nhiên, định tuyến theo phương pháp này vẫn tồn tại ba nhược điểm chính. Thứ nhất: Việc phân bố các nút trong mạng vào các cụm là không đồng đều do các nút gia nhập mạng chỉ dựa vào độ mạnh của tín hiệu nhận được. Thứ hai: Khoảng thời gian ổn định truyền dữ liệu tồn tại trong một vòng sau giai đoạn thiết lập cụm là không được xác định rõ ràng. Thứ ba: Khoảng cách truyền thông trong mỗi cụm và truyền thông từ các CH đến BS vẫn còn xa do thuật toán chọn CH và phân cụm chưa tối ưu. Điều này ảnh hưởng lớn đến hiệu quả hoạt động và sử dụng năng lượng của các nút cảm biến để kéo dài thời gian sống mạng. Giải pháp đề xuất của chúng tôi LEACH-DE nâng cao hiệu quả sử dụng năng lượng bằng cách chọn nút CH và xây dựng cụm có tính đến mức năng lượng còn lại trung bình của các nút còn sống và xem xét đến khoảng cách từ nút ứng viên đến BS. Kết quả mô phỏng khẳng định thuật toán LEACH-DE cho hiệu quả sử dụng năng lượng tốt hơn thuật toán phân cụm LEACH và LEACH-C truyền thống nhưng tỉ lệ chuyển phát gói tin vẫn thấp hơn LEACH-C. Cụ thể là tỉ lệ nút còn sống theo thời gian của LEACH-DE tăng lên khoảng 20% và tỉ lệ (phần trăm) nút chết theo thời gian giảm xuống khoảng 20% so với LEACH. Trong khi đó, tỉ lệ gói tin nhận được ở BS khi vị trí của BS ở (49, 175) và (49, 225) thì LEACH-DE giảm 10% so với LEACH,

còn khi BS ở (49, 265) và (49, 300) thì LEACH-DE là tương đương với LEACH. Các kết quả nghiên cứu trong Chương 3 tương ứng với công trình 1 đã công bố.

Để nâng cao hơn nữa giải pháp được đề xuất, các vấn đề sau đây sẽ được nghiên cứu giải quyết (ở các Chương 4, 5 tiếp theo):

- Phân chia cân bằng tổng số nút còn sống trong mạng vào 5% số cụm
- Tính toán tối ưu khoảng thời gian ổn định truyền dữ liệu để giảm chi phí năng lượng trong giai đoạn thiết lập cụm.
- Giảm khoảng cách truyền thông trong cụm bằng cách kết nối các nút trong mỗi cụm thành chuỗi hoặc xây dựng cây tối thiểu.
- Giảm số bit dữ liệu truyền trong mạng dựa trên giải pháp tổng hợp, nén dữ liệu trong chuỗi hoặc cây.

## **Chương 4: ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG DỰA TRÊN CHUỖI**

Chương này trình bày thuật toán xây dựng chuỗi với hai đề xuất cải tiến của chúng tôi, đề xuất thứ nhất có tên là DFCB (Data Fusion and Chain-Based Clustering for Energy-Efficient in Wireless Sensor Networks). Theo đề xuất này, các nút cảm biến sẽ được kết nối với nút hàng xóm gần nhất để tạo thành một chuỗi dài bằng cách sử dụng thuật toán tham lam. Ngoài ra, việc xây dựng chuỗi còn kết hợp với tổng hợp dữ liệu và nén dữ liệu sử dụng mã nguồn phân tán. Các kết quả mô phỏng của chúng tôi cho thấy hiệu quả năng lượng DFCB tốt hơn khoảng 40% so với LEACH và khoảng 10% so với PEGASIS.

Đề xuất thứ hai, được gọi là SCBC (Sector-Chain Based Clustering for Energy Efficiency In Heterogeneous Wireless Sensor Network), đề xuất lược đồ định tuyến phân cụm dựa trên chuỗi, nâng cao hiệu quả sử dụng năng lượng bằng việc cân bằng số nút cảm biến trong mỗi cụm và tối ưu thời gian hoạt động của các cụm trong giai đoạn truyền dữ liệu. Các kết quả mô phỏng của chúng tôi đối với đề xuất này cho thấy SCBC có thể cải tiến khoảng 20% và 15% tương ứng khi so sánh với giao thức PEGASIS và IEEPB.

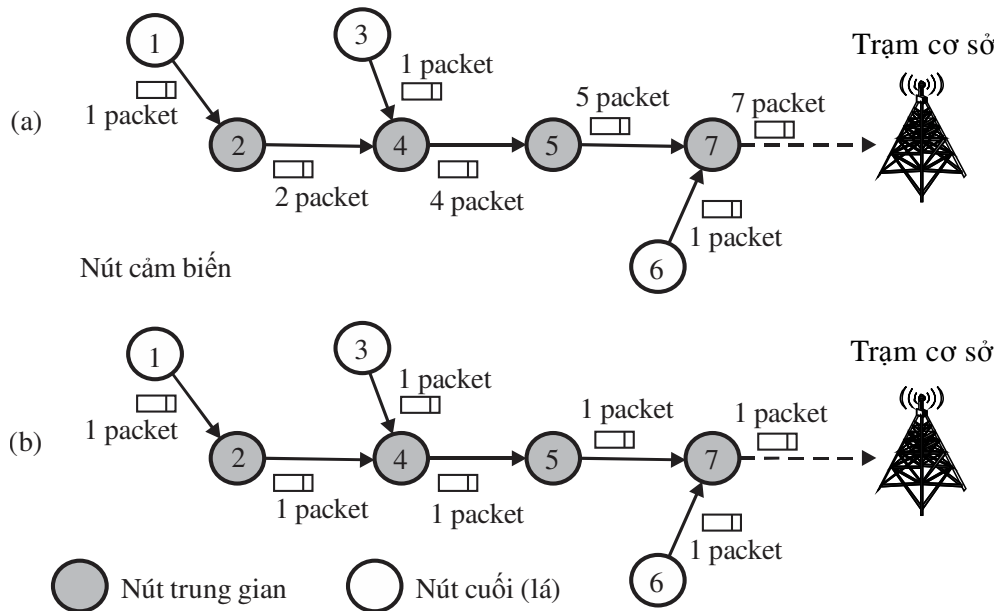
### **4.1. Đặt vấn đề**

Như đã được thảo luận trong Chương 2, định tuyến phân cụm cho hiệu quả năng lượng tốt bằng cách tổ chức mạng thành các cụm nhỏ, việc truyền dữ liệu trong mạng được chia thành truyền trong cụm (*intra-cluster*) và truyền đến BS (*inter-cluster*). Trong mỗi cụm việc truyền dữ liệu từ các nút thành viên cụm đến CH là đơn chặng và chỉ có một nút CH truyền dữ liệu đến BS, do đó, nó giảm được xác suất đụng độ gói tin và tiết kiệm năng lượng cho các nút thành viên trong cụm. Tuy nhiên, kỹ thuật này không thể tránh được việc có các liên kết giữa một số cặp nút thành viên trong cụm với CH, có khoảng cách xa nhau, dẫn đến hiệu quả sử dụng năng lượng không cao. Để khắc phục vấn đề này, chúng tôi đề xuất giải pháp xây

dựng chuỗi dài kết hợp với tổng hợp dữ liệu nhằm giảm lượng dữ liệu cần truyền, giúp nâng cao hiệu quả sử dụng năng lượng trong các hệ thống thông tin cảm biến.

#### 4.2. Phân tích tổng hợp dữ liệu

Trong mạng WSN, các nút cảm biến quan sát môi trường và định kỳ gửi gói dữ liệu tương quan tới BS, BS được kết nối với người dùng thông qua Internet. Hai nguồn dữ liệu X và Y được gọi là tương quan khi khoảng cách Hamming giữa chúng  $dH(X, Y) \leq 1$ , hay xác suất xuất hiện của các bit trong nguồn là  $\Pr(X_i = 0) = \Pr(X_i = 1) = \Pr(Y_i = 0) = \Pr(Y_i = 1) = 0.5$ , với mọi bit  $i=0$  đến hết nguồn [111]. Để tiết kiệm năng lượng của các nút cảm biến, chúng ta có thể sử dụng các kỹ thuật nén dữ liệu để giảm số bit cần truyền, thí dụ kỹ thuật nén DSC (Chi tiết được trình bày ở Phụ lục 2 Trang 144). Thêm nữa, nếu dữ liệu cảm biến từ tất cả các nút đều được gửi đến BS thì các nút sẽ lãng phí nhiều năng lượng cho việc truyền dữ liệu trùng lặp trong mạng cảm biến. Do đó, bên cạnh kỹ thuật nén thì kỹ thuật tổng hợp dữ liệu cũng là một kỹ thuật tốt để đạt được hiệu quả năng lượng sử dụng lý thuyết Dempster-Shafer (Chi tiết được trình bày trong Phụ lục 1 Trang 139) [17, 20, 30].



**Hình 4.1:** Mô hình truyền dữ liệu trong chuỗi, (a) không tổng hợp dữ liệu và (b) có tổng hợp dữ liệu

Trong phần này, chúng tôi phân tích việc tổng hợp các gói dữ liệu thông qua mô hình chuỗi được minh họa trong Hình 4.1; Trong đó hình (a) là trường hợp không thực hiện tổng hợp dữ liệu, hình (b) là có tổng hợp dữ liệu. Các nút trung gian và CH trong chuỗi sẽ tổng hợp một hoặc nhiều gói dữ liệu từ các nút cảm biến khác nhau để sinh ra một gói dữ liệu duy nhất gửi đến BS. Do đó, số lượng gói dữ liệu được truyền trong chuỗi và BS giảm đáng kể. Cụ thể, tổng năng lượng tiêu thụ có thể được tính cho một chu kỳ (một vòng gồm nhiều chu kỳ) truyền dữ liệu cảm biến trong chuỗi như dưới đây:

$$E_a = 14E_{RX}(q) + 21E_{TX}(q, d(i, j)) + 7E_G(q) + 7E_S(q) \quad (4.1)$$

$$E_b = 6E_{RX}(q) + 7E_{TX}(q, d(i, j)) + 4E_{DF}(q) + 7E_G(q) + 7E_S(q) \quad (4.2)$$

trong đó,  $E_a$  và  $E_b$  tương ứng là tổng năng lượng tiêu thụ đối với trường hợp (a) không tổng hợp dữ liệu và (b) có tổng hợp dữ liệu. Quan sát Hình 4.1 chúng ta có thể thấy, số nút trong chuỗi còn sống bằng bảy, do đó mạng sẽ tiêu tốn  $7E_G(q)$  và  $7E_S(q)$ , trong khi nếu có tổng hợp dữ liệu thì sẽ mất  $7E_{TX}(q, d(i, j))$ , nếu không có tổng hợp dữ liệu thì sẽ mất  $14E_{TX}(q, d(i, j))$  vì các nút phải chuyển tiếp gói dữ liệu của cả nút hàng xóm của nó đến BS.

Để cho việc tính toán được đơn giản nhưng cũng không làm mất tính tổng quát, chúng tôi giả sử rằng kích thước gói tin của mỗi nút là  $q = 1$  bit và khoảng cách  $d$  giữa hai nút là bằng nhau tương đương và  $d = 10m$ . Kết hợp với (3.6), (3.8), (4.1), và (4.2), ta có:

$$E_a = 35E_{elec} + 21E_{fris}d^2 + 7E_G + 7E_S \quad (4.3)$$

$$E_b = 13E_{elec} + 7E_{fris}d^2 + 4E_{DF} + 7E_G + 7E_S \quad (4.4)$$

So sánh (4.3) và (4.4) có thể thấy rõ ràng rằng, năng lượng tiêu thụ khi không có tổng hợp dữ liệu lớn hơn nhiều so với truyền dữ liệu có tổng hợp dữ liệu. Thế nhưng, làm thế nào ta có thể tổng hợp từ nhiều gói dữ liệu tương quan để đưa ra được một gói dữ liệu duy nhất như Hình 4.1? Để giải quyết vấn đề này, chúng ta có thể kết hợp các luật của lý thuyết Dempster-Shafer. Tuy nhiên, độ phức tạp về thời gian tính toán tập luật theo lý thuyết Dempster-Shafer tăng lên theo hàm mũ khi số lượng nút cảm biến trong mạng tăng lên.

Vì thế, Zeng và các cộng sự [119] đã đề xuất thuật toán trung tâm tổng hợp sự kiện năng lượng thấp - LEECF (Low-Energy Event Centric Fusion) dựa trên phân tích ma trận. Nhóm tác giả đã chứng minh được rằng kết quả đạt được từ LEECF giống với Dempster-Shafer nhưng thời gian thực hiện nhỏ hơn.

Đặt  $W = \{w\}_{n \times v}$ , biểu diễn ma trận niềm tin của  $n$  nút cảm biến có dữ liệu cần tổng hợp (mỗi nút cảm biến trên một dòng trong  $W$ ), mỗi nút quan sát môi trường đích với  $v$  giá trị có thể xảy ra tại một thời điểm ( $W$  là ma trận vuông có  $n$  hàng và  $v$  cột),  $w_{ij}$  là niềm tin của nút cảm biến thứ  $i$  quan sát giá trị xác suất đích thứ  $j$ , và tổng niềm tin (tổng các khả năng có thể quan sát được tại một thời điểm) của một nút cảm biến nên bằng 1.

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & \dots & w_{1v} \\ w_{21} & w_{22} & \dots & \dots & w_{2v} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & \dots & w_{nv} \end{bmatrix}$$

Từ đó chúng ta có  $w_{i1} + w_{i2} + \dots + w_{iv} = 1$  ( $i=1, \dots, n$ ). Dựa vào tập quy tắc (luật) của lý thuyết Dempster-Shafer và các phép tính toán trên ma trận. Ta có thể chuyển vị một hàng (hàng  $i$ ) của ma trận  $W$  và nhân với một hàng khác (hàng  $j$ ) như  $W_i^T \times W_j = [w_{i1}, w_{i2}, \dots, w_{iv}]^T \times [w_{j1}, w_{j2}, \dots, w_{jv}]$ , ký hiệu  $W^T$  là ma trận chuyển vị, chúng ta có thể lấy được một ma trận mới  $U = \{W_i^T \times W_j\}_{n \times v}$  như sau:

$$U = \begin{bmatrix} w_{i1} \times w_{j1} & w_{i1} \times w_{j2} & \dots & \dots & w_{i1} \times w_{jv} \\ w_{i2} \times w_{j1} & w_{i2} \times w_{j2} & \dots & \dots & w_{i2} \times w_{jv} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ w_{iv} \times w_{j1} & w_{iv} \times w_{j2} & \dots & \dots & w_{iv} \times w_{jv} \end{bmatrix}$$

Trong đó, đường chéo chính của  $U$  là kết quả niềm tin từ  $n$  nút cảm biến và tổng của các phần tử, chúng khác với đường chéo chính là yếu tố không chắc chắn  $F$  [119].

$$F = F + \sum_{p \neq q} w_{ip} \times w_{jq} \quad (p, q = 1, 2, \dots, v) \quad (4.5)$$

và ta có:

$$M_j = \frac{u_{ij}}{1 - F} \quad (\forall j = 1, \dots, v) \quad (4.6)$$



trong đó,  $M_j$  là kết quả tổng hợp dữ liệu ở đầu ra của thuật toán tương ứng với khả năng thứ  $j$  có thể xảy ra mà các nút cảm biến được.

### 4.3. Đề xuất cải tiến thuật toán xây dựng chuỗi dài

#### 4.3.1. Giai đoạn chọn nút cụm trưởng (CH)

Trong mỗi vòng của DFCB, tất cả các nút còn sống sẽ gửi thông điệp HELLO tới BS, chúng chứa mã định danh của nút, thông tin về vị trí và năng lượng còn lại. BS sẽ lựa chọn nút lãnh đạo chuỗi. Chúng có mức năng lượng còn lại ( $E_{\text{residual}}$ ) lớn hơn mức năng lượng trung bình ( $E_{\text{average}}$ ) của các nút còn sống và có hàm  $\text{cost}()$  đạt giá trị lớn nhất.  $E_{\text{average}}$  và hàm  $\text{cost}()$  được tính tương ứng theo công thức (3.3) và (4.7) dưới đây:

$$\text{cost}(i) = \text{Max}_i \left\{ \frac{ec_1}{dc_2} \times \frac{E_{\text{residual}}(i)}{d(i, \text{BS})} \right\} \quad (4.7)$$

Trong đó:

- $d(i, \text{BS})$  là khoảng cách địa lý từ nút ứng viên thứ  $i$  đến BS, nó được tính dựa vào công thức (3.5).
- Hằng số  $ec_1$  và  $dc_2$  được thiết lập để dung hòa tỷ lệ giữa năng lượng còn lại của nút và khoảng cách từ nút  $i$  đến BS. Khi  $ec_1$  là lớn hơn  $dc_2$ , điều đó có nghĩa là năng lượng còn lại quan trọng hơn khoảng cách giữa nút ứng viên và BS khi xem xét chọn nút CH lãnh đạo chuỗi. Ở đây, trong các kịch bản mô phỏng với các tham số mô phỏng như Bảng 3.1, chúng tôi chọn giá trị  $ec_1=2/J$  và  $dc_2 = 5/m$ .
- Hàm  $\text{cost}()$  trả về giá trị hằng số, dùng làm tiêu chuẩn chọn nút CH, nút nào có giá trị  $\text{cost}()$  lớn nhất trong cụm sẽ được chọn ở vòng hiện tại.

Thuật toán 4.1 đặc tả các bước thực hiện việc chọn nút CH lãnh đạo chuỗi.

---

#### Thuật toán Chọn nút CH lãnh đạo chuỗi (Thuật toán 4.1)

---

**Input:** Các nút mạng, biến  $CH=\emptyset$ .

**Output:** Biến  $CH$  chứa nút lãnh đạo chuỗi.

---

- 
- 1: **for each**  $p$  in {Danh sách nút còn sống} **do**
  - 2:   Gửi thông điệp HELLO chứa năng lượng còn lại và vị trí địa lý tới BS
  - end for**
  - 4: BS tính mức năng lượng còn lại trung bình như công thức (3.3)
  - 5: BS chọn nút CH, nút có hàm  $\text{cost}()$  đạt giá trị lớn nhất như trong (4.7)
  - 6: BS quảng bá thông điệp HELLO tới tất cả các nút trong mạng
  - 7: Các nút nhận thông điệp HELLO
  - 8: Chuyển sang thực hiện **Thuật toán Xây dựng chuỗi dài (Thuật toán 4.2)**
- 

#### 4.3.2. Giai đoạn xây dựng chuỗi

Sau khi chọn nút CH, BS xây dựng chuỗi như Thuật toán 4.2 dưới đây dựa trên thuật toán tham lam (GA-Greedy Algorithms). Đầu tiên, nút cụm trưởng sẽ gia nhập vào chuỗi và được coi như nút cuối chuỗi. Sau đó các nút khác được chọn thông qua vòng lặp tìm kiếm nút bên ngoài chuỗi có khoảng cách từ nó đến chuỗi gần nhất. Để giảm các liên kết dài, DFCB thực hiện so sánh không chỉ với nút cuối chuỗi mà còn so sánh với tất cả các nút khác đã có trong chuỗi để tìm nút hàng xóm gần chuỗi nhất và nút vừa thêm vào lại được coi là nút cuối chuỗi. Cứ như vậy cho đến khi kết thúc vòng lặp, chuỗi mới đã được xây dựng, Thuật toán 4.2 đặc tả các bước thực hiện xây dựng chuỗi bắt đầu từ nút CH [64, 93, 118].

---

#### Thuật toán Xây dựng chuỗi dài (Thuật toán 4.2)

---

**Input:** Các nút mạng, biến  $CHAIN = \emptyset$ .

**Output:** Biến  $CHAIN$  chứa chuỗi.

- 1:  $endNode \leftarrow CH$ ;
  - 2:  $CHAIN \leftarrow endNode$ ; // Chuỗi bắt đầu với nút là nút CH
  - 3: **while** ({Danh sách nút còn sống}  $\neq \emptyset$ ) **do**
  - 4:   Tìm kiếm nút  $j$  trong {Danh sách nút còn sống}, nút có  $d(endNode, j)$  là nhỏ nhất;
-

---

```

5:   Tìm kiếm nút  $k$  trong  $\{CHAIN\}$ , nút có  $d(j, k)$  là nhỏ nhất;
6:   if  $d(k, j) < d(endNode, j)$  and  $(j \neq endNode)$  then
7:       Thêm nút  $j$  vào trong  $\{CHAIN\}$  bằng cách kết nối với nút  $k$ ;
8:   else
9:       Thêm nút  $j$  vào trong  $\{CHAIN\}$  bằng cách kết nối với nút  $endNode$ ;
10:  end if
12:   $endNode \leftarrow node\ j$ ;
12:  Xóa nút  $j$  trong  $\{Danh\ sách\ nút\ còn\ sống\}$ ;
13: end while
14:  Tạo lịch TDMA cho tất cả các nút trong chuỗi để truyền dữ liệu;
15:  Quảng bá thông tin chuỗi (CHAIN) và TDMA tới các nút trong mạng
16:  return CHAIN;

```

---

#### 4.3.3. Giai đoạn tổng hợp dữ liệu trong chuỗi

Sau khi xây dựng chuỗi, BS quảng bá thông tin về CH, chuỗi, lịch TDMA và mã CDMA tới các nút trong mạng, các nút cảm biến không dây bắt đầu vào giai đoạn thu thập, tổng hợp và nén dữ liệu trước khi truyền đến nút hàng xóm dọc theo chuỗi và truyền đến BS. Thuật toán 4.3 đặc tả các bước thực hiện tổng hợp dữ liệu ở các nút trung gian hoặc nút CH trong chuỗi. Ở đây chúng tôi giả thiết dữ liệu cảm biến thoả mãn điều kiện tương quan để có thể sử dụng mã nguồn phân tán DSC (xem Phụ lục 2, trường hợp 1). Thuật toán gồm ba công việc chính: (1) giải nén gói dữ liệu nhận được từ nút hàng xóm sử dụng DSC (từ dòng 1 đến dòng 9); tiếp theo, (2) đoạn mã từ dòng 10 đến dòng 19 thực hiện tổng hợp dữ liệu nhận được từ nút hàng xóm với dữ liệu nó cảm biến được dựa trên phân tích ma trận như [119], (3) đoạn mã từ dòng 20 đến (27) nén gói dữ liệu không mất theo DSC. Chi tiết thuật toán được minh họa như dưới đây:

---

**Thuật toán Tổng hợp dữ liệu trong chuỗi (Thuật toán 4.3)**

---

**Ký hiệu:**

- CHAIN: thông tin về chuỗi dài,
- $G$  là ma trận sinh  $b$  hàng  $u$  cột (xem Phụ lục 2, trường hợp 1)
- $Z$  chứa  $2^3$  tập con được sử dụng để xác định dữ liệu ở bộ giải mã
- $X, Y, S, F, U, A$  là các biến sử dụng trong chương trình
- $Z_s$  là phần tử có chỉ số nhị phân  $s$  trong tập  $Z$  (ví dụ  $s=011$ ,  $Z_s = (0011, 1100)$ ), (xem Phụ lục 2, trường hợp 1, trang 145)

**Input:** Gói tin nén, thông tin cảm biến, CHAIN,  $G, Z$ .

**Output:** Gói tin đã tổng hợp và nén

// Đoạn mã sau giải nén gói dữ liệu nhận được từ nút hàng xóm trong chuỗi

- 1: **if** (nút  $i$  trong {CHAIN} là nút trung gian hoặc CH) **then**
- 2:     **if** (gói dữ liệu đã được nén) **then**
- 3:         **for each** (7 bit trong gói dữ liệu được nén ở bộ giải mã) **do**
  - Y  $\leftarrow$  Lấy 4 bit cao như "side information"
  - s  $\leftarrow$  Lấy 3 bit thấp
- 6:         Tìm kiếm giá trị  $X$  trong tập con  $Z_s$ , cái mà có khoảng cách Hamming với  $Y$  là nhỏ hơn hoặc bằng 1;
- 7:         Thêm  $Y$  và  $X$  vào trong gói tin giải nén;
- end for**
- 9:     **end if**

//Đoạn mã tổng hợp dữ liệu giống với công thức (P1.3) và  $c$  (xem Phụ lục 1)

- 10:     Biểu diễn dữ liệu cảm biến của  $n$  nút trong ma trận  $W = \{w\}_{n \times v}$
- 11:      $A \leftarrow$  Chuyển vị hàng 1 của  $W$
- 12:     **for each** ( $i \leftarrow$  hàng từ 2 đến  $n$  trong  $W$ ) **do**

//Nhân ma trận cột  $A$  với ma trận  $W_i$ , ta được ma trận  $U$

- 13:         Tính  $U \leftarrow A * W_i$ ;
  - for each** ( $j \leftarrow$  hàng từ 1 đến  $v$  trong  $U$ ) **do**
-

---

```

       $A_j \leftarrow$  Lấy các phần tử trên đường chéo chính của  $U$ ,  $(u_{jj})$ ;
16:      Tính  $F \leftarrow F + \sum_{p \neq q} u_{pq} \quad (p, q = 1, 2, 3, \dots, v)$ ;
17:      end for
      //Đoạn mã tổng hợp dữ liệu giống với công thức (P1.3) và (P1.5) (xem Phụ lục 1).
18:      for each  $(j \leftarrow$  từ 1 đến  $m)$  do
19:           $M_j = A_j / (1 - F)$ ;
      // Đoạn mã sau nén gói dữ liệu không mất theo DSC
20:      if (dữ liệu cần được nén ở bộ mã hóa) then
21:          for each 8 bit trong gói dữ liệu do
               $Y \leftarrow$  Lấy 4 bit cao như thông tin phụ "side information"
               $X \leftarrow$  Lấy 4 bit thấp như dữ liệu gốc
24:          Tính toán giá trị  $s \leftarrow X * G^T$ 
25:          Thêm  $Y$  và  $s$  vào trong gói dữ liệu mới được nén
26:          end for
27:      end if

```

---

Các lệnh từ dòng 3 đến dòng 8 sẽ giải nén gói tin để nhận được thông tin gốc bằng cách đọc một khối 7 bit, lấy 4 bit đầu đặt ở biến  $Y$  như "thông tin phụ", sau đó gán 3 bit còn lại vào biến  $s$ , dòng lệnh 6 tìm  $X$  trong tập con  $Z_s$  (có 2 phần tử), chọn một phần tử mà có khoảng cách Hamming với  $Y$  là nhỏ hơn hoặc bằng 1 và ghi  $Y$ ,  $X$  vào gói dữ liệu giải nén. Việc tổng hợp dữ liệu bắt đầu từ dòng 10; biểu diễn dữ liệu cảm biến thu được trên một ma trận  $W$  như [119] sau đó chuyển vị một hàng thành ma trận cột và nhân với một hàng còn lại, ta sẽ được ma trận  $U$  thông qua dòng lệnh 13. Dòng lệnh 14 và 15 sẽ lấy các phần tử trên đường chéo chính của  $U$  gán vào ma trận cột  $A$ , tiếp theo dòng lệnh 16 tính tổng các phần tử khác đường chéo chính trong ma trận  $U$  đặt vào biến  $F$ . Cuối cùng, dòng lệnh 19 sẽ tính biểu thức (4.6) để được biểu thức giống với công thức (P1.3) và (P1.5) (xem Phụ lục 1).

Để nén dữ liệu theo DSC, ta sử dụng các dòng lệnh từ 20 đến 27. Dòng lệnh 20 đọc mỗi khối dữ liệu 8 bit sau đó tách ra thành 2 dãy, mỗi dãy 4 bit đặt ở các biến  $X$  và

Y. Theo DSC, chúng ta không gửi cả 8 bit này mà gửi Y như "thông tin phụ" và gửi  $s = 3$  bit nhờ sử dụng phép nhân  $\mathbf{X} * \mathbf{G}^T$  ( $s$  chính là kết quả của phép nhân  $\mathbf{X}$  với ma trận sinh  $\mathbf{G}$  và  $\mathbf{X}$  có trong tập con  $Z_S$ , xem Phụ lục 2, trường hợp 1). Câu lệnh dòng 25 sẽ ghi Y và  $s$  vào gói tin nén ( $Y+s = 7$  bit thay vì gửi  $Y + X = 8$  bit, nhờ đó giảm được 1 bit).

Về độ phức tạp tính toán của DFCB, các Thuật toán 4.1 và 4.2 mỗi nút chỉ thực hiện truyền và nhận một thông điệp đến BS, do đó, độ phức tạp là hằng số, còn các thao tác tính toán được thực hiện bởi BS. Thuật toán 4.1 thực hiện tính hàm  $\text{cost}()$  theo (4.7) để chọn nút CH có độ phức tạp tính toán là  $O(n)$ . Công việc xây dựng chuỗi dài theo Thuật toán 4.2 sử dụng thuật toán tham lam có độ phức tạp tính toán là  $O(n \log(n))$ . Độ phức tạp tính toán của DFCB chủ yếu ở Thuật toán 4.3 tổng hợp dữ liệu từ  $n$  nút cảm biến, mỗi nút quan sát môi trường đích với  $v$  giá trị, có độ phức tạp là  $O(ch.v^2)$ . Do đó, DFCB có độ phức tạp tính toán là  $O(ch.v^2)$  và độ phức tạp thông báo là hằng số.

#### **4.3.4. Giai đoạn truyền dữ liệu**

Ở giai đoạn này, các nút cảm biến định kỳ thu thập dữ liệu từ môi trường bên ngoài, tổng hợp và gửi gói dữ liệu nó quan sát được đến các nút kế tiếp trong khe thời gian được gán bởi kỹ thuật TDMA dọc theo chuỗi. Đầu tiên, các nút ở xa BS nhất trong chuỗi sẽ bắt đầu nén dữ liệu và truyền tới nút hàng xóm của nó. Các nút giữa nhận dữ liệu, giải nén, tổng hợp với dữ liệu nó quan sát được và nén lại rồi mới gửi đến CH. Bất cứ khi nào nút CH nhận được tất cả các gói dữ liệu, nó sẽ chuyển tiếp đến BS theo mã CDMA sau khi thực hiện nén và tổng hợp theo cách tương tự. Sau một khoảng thời gian, vòng mới sẽ được bắt đầu bằng việc chọn lại CH và xây dựng lại chuỗi để chuyển đổi vai trò làm CH cho các nút khác để nâng cao hiệu quả sử dụng năng lượng.

#### **4.4. Mô phỏng để đánh giá hiệu quả của đề xuất cải tiến giao thức DFCB**

Chúng tôi đã cài đặt giao thức DFCB trong công cụ mô phỏng mạng mã nguồn mở ns-2.34 và chạy mô phỏng nhiều lần trên nhiều kịch bản khác nhau (sử dụng 100

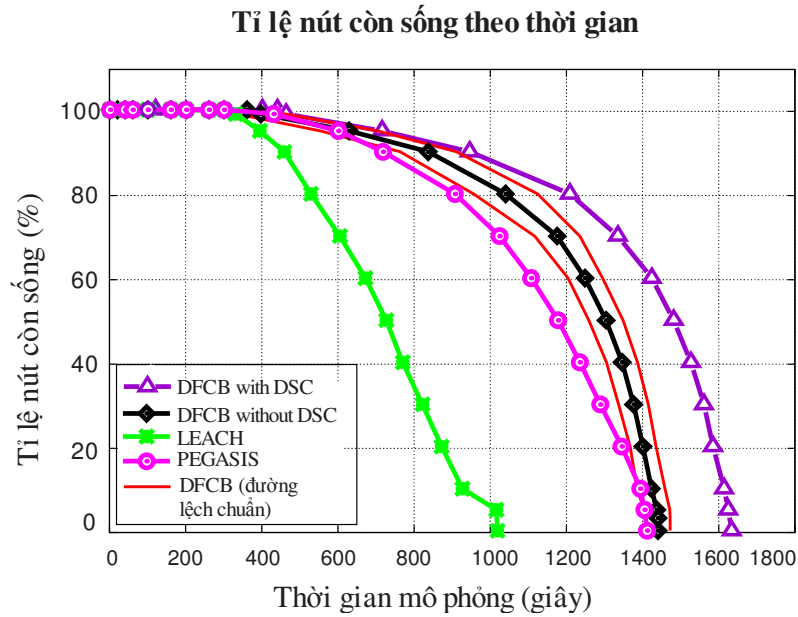
kịch bản đã sinh ra ở Chương 3, Mục 3.3). Các kịch bản này có tô-pô mạng gồm 100 nút cảm biến được sinh ngẫu nhiên nhiều lần trong vùng cảm biến có diện tích  $100\text{m} \times 100\text{m}$ , các tham số mô phỏng giống như Bảng 3.1 [29, 44, 75, 126]. Giả thiết các nút cảm biến và BS là không di chuyển sau khi đã được triển khai.

Để đánh giá hiệu năng của giao thức DFCB, chúng tôi đã thực hiện các bước giống với các bước ở Chương 3, Mục 3.3 (từ Bước 2 đến Bước 7) để xác định số kịch bản ( $n_{sc}$ ) cần phải chạy mô phỏng bao nhiêu là đủ? Dựa vào bảng số liệu trình bày giá trị trung bình  $m_x$ , độ lệch chuẩn  $\sigma$  và tỉ lệ độ lệch chuẩn  $\xi$  theo các công thức (3.11), (3.12) và (3.13), chúng tôi chọn giá trị  $\xi$  với giao thức PEGASIS bằng 1,5%, và DFCB là 2,3% tại thời điểm mạng có 95% nút chết thì số kịch bản mô phỏng như sau: Với giao thức PEGASIS, số kịch bản đã chạy  $n_{sc}= 11$  kịch bản thì dừng và với giao thức DFCB số kịch bản đã chạy là  $n_{sc}= 11$ . Chúng tôi cũng đã chạy mô phỏng thêm cho giao thức DFCB thì thấy giá trị trung bình và tỷ lệ độ lệch chuẩn ( $\xi=2,3\%$ ) lặp lại ở lần chạy trên kịch bản thứ 16, 17, 20 và 23, do đó chúng tôi dừng lại và chọn số kịch bản cần chạy mô phỏng  $n_{sc}= 11$  kịch bản.

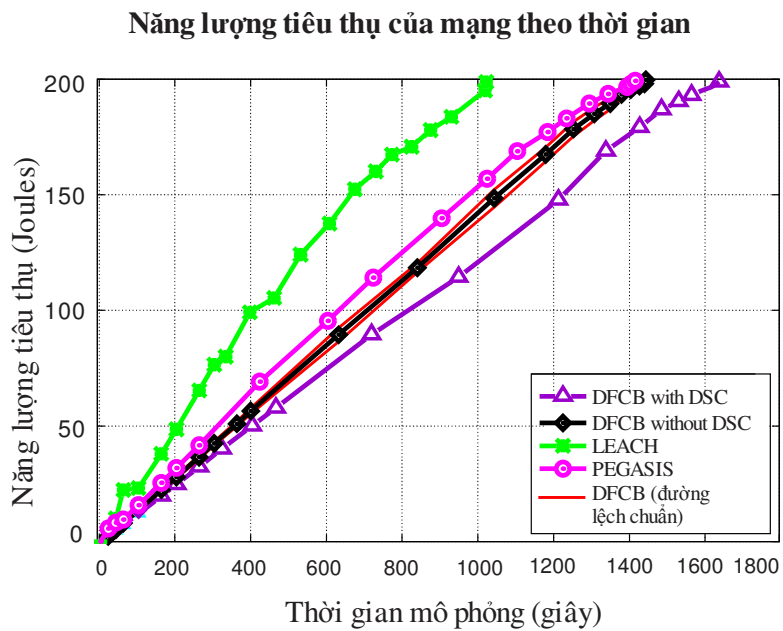
Giá trị trung bình nhận được từ mô phỏng được chúng tôi xử lý và trình bày bằng đồ thị, trên Hình 4.2 biểu diễn các kết quả mô phỏng để đánh giá và so sánh tỉ lệ nút cảm biến còn sống theo thời gian đối với các giao thức LEACH, PEGASIS và DFCB. Rõ ràng, nếu áp dụng DFCB thì mạng có thời gian sống lâu hơn khoảng 40% và 10% khi so với LEACH và PEGASIS trong trường hợp có tổng hợp, không nén. Ngoài ra, Hình 4.2 cũng biểu diễn sự so sánh giữa DFCB trong trường hợp không nén và có nén sử dụng mã nguồn phân tán (DSC). Các đường màu đỏ biểu diễn độ lệch chuẩn của giao thức DFCB có  $\xi$  bằng 2,3% so với giá trị trung bình tương ứng với số kịch bản  $n_{sc}= 11$ , điều này có nghĩa là nếu chúng ta có tăng thêm mô phỏng thì kết quả thu được cũng không vượt ra ngoài đường lệch chuẩn.

Hình 4.3 biểu diễn kết quả mô phỏng để đánh giá và so sánh năng lượng tiêu thụ theo thời gian của ba giao thức. Trục X biểu diễn thời gian mô phỏng (tính theo giây) và trục Y biểu diễn mức năng lượng tiêu thụ (tính theo Joules). Dựa vào các kết quả trên đồ thị ta thấy rằng, giao thức được chúng tôi đề xuất có mức năng

lượng tiêu thụ nhỏ hơn LEACH và PEGASIS. Các đường trong đồ thị càng gần trục X cho hiệu quả năng lượng càng tốt.

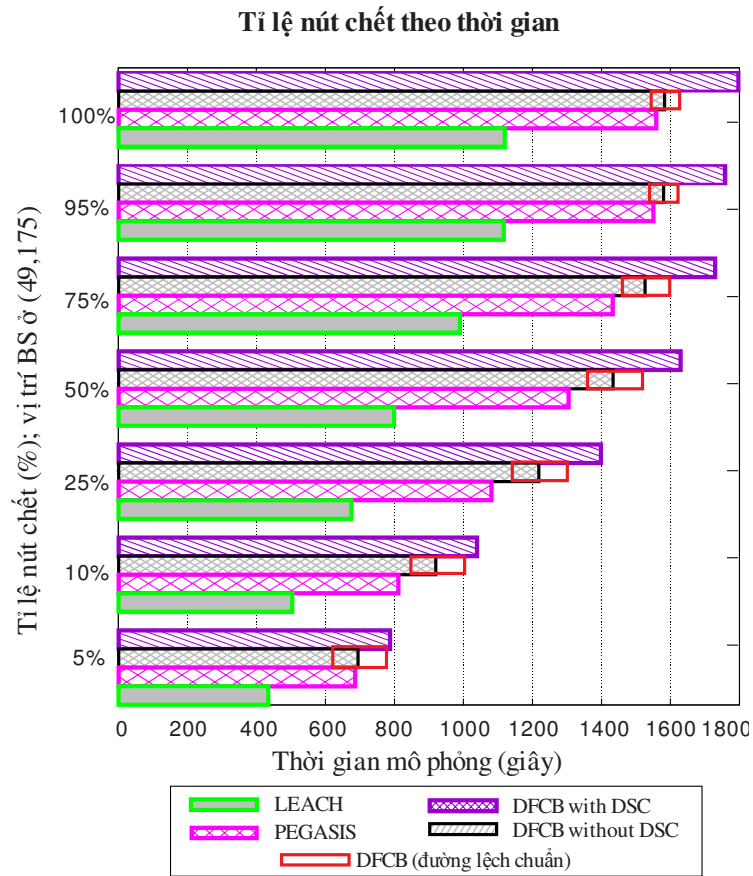


**Hình 4.2:** Tỉ lệ nút còn sống trong suốt thời gian mô phỏng

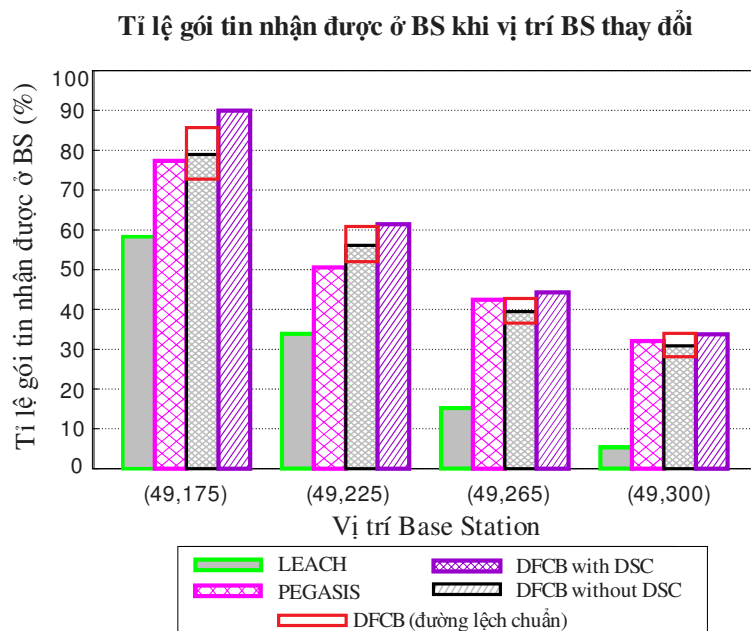


**Hình 4.3:** Tổng năng lượng mạng tiêu thụ khi áp dụng ba giao thức





**Hình 4.4:** Tỉ lệ các nút chết theo thời gian mô phỏng



**Hình 4.5:** Tỉ lệ gói dữ liệu nhận được ở BS

Có thể thấy rõ trên Hình 4.4, chúng tôi biểu diễn bằng đồ thị cột, tỉ lệ (phần trăm) các nút chết của ba giao thức được phân tích theo thời gian mô phỏng (vẽ ngược lại với Hình 4.2). Chúng ta có thể quan sát thấy rằng DFCB với mức tiêu thụ năng lượng ít đã đạt được hiệu quả năng lượng tốt và kéo dài thời gian sống lâu hơn LEACH và PEGASIS.

Hình 4.5 biểu diễn bằng đồ thị tỉ lệ gói dữ liệu nhận được ở các vị trí khác nhau của BS. Rõ ràng, chúng có quan hệ giảm về tỉ lệ gói dữ liệu nhận được ở BS khi chúng tôi di chuyển BS ra vị trí xa dần trong vùng mô phỏng. Tỉ lệ gói tin nhận được của DFCB là tốt hơn LEACH và tương đương với PEGASIS.

#### **4.5. Phân tích và so sánh với các thuật toán cùng hướng khác**

Các thuật toán xây dựng chuỗi dài trong [38, 64, 116] chưa xem xét kết hợp giữa năng còn lại của nút ứng viên và khoảng cách từ nó đến BS trong giai đoạn chọn nút CH, hơn nữa các đề xuất trên cũng chưa kết hợp được giữa định tuyến và tổng hợp, nén dữ liệu cho nên hiệu quả đạt được chưa cao. Đề xuất DFCB kết hợp giữa định tuyến và tổng hợp, nén dữ liệu sử dụng mã nén phân tán để giảm số bit dữ liệu cần truyền từ đó nâng cao hiệu quả sử dụng năng lượng.

#### **4.6. Đề xuất cải tiến lược đồ xây dựng cụm chuỗi**

Đề xuất cải tiến thứ hai của chúng tôi trong lược đồ xây dựng cụm chuỗi được gọi là SCBC (Sector-Chain Based Clustering) nhằm cải tiến lược đồ xây dựng chuỗi đã từng được đề xuất bởi Stephanie Lindsey [64], trong đó các nút mạng cảm biến được cân bằng về số lượng các nút trong mỗi cụm và khoảng thời gian tồn tại trong giai đoạn truyền dữ liệu của mỗi vòng được tính dựa vào các nút CH trong mỗi cụm. Trước hết, chúng tôi phân tích năng lượng tiêu thụ của các nút trong SCBC.

##### **Năng lượng tiêu thụ của các nút trong giai đoạn thiết lập**

Giả sử mạng gồm  $N$  nút cảm biến được triển khai ngẫu nhiên trong vùng  $M \times M$ , BS có tọa độ là  $(x_{BS}, y_{BS})$ , ở mỗi vòng các nút mạng được chia thành  $k$  chuỗi (cung). Người ta đã xác định được giá trị tối ưu của  $k$  là xấp xỉ 5% [44, 113]; Do đó, có trung bình  $N/k$  nút cảm biến trong mỗi cụm, chúng bao gồm một nút CH và  $((N/k)-1)$  nút thành viên.

Ký hiệu  $E_{\text{control}}$  chỉ năng lượng tiêu thụ của các nút trong một cụm (chuỗi) khi chúng trao đổi thông điệp với BS trong giai đoạn thiết lập cụm ở vòng thứ  $r$ , có thể xây dựng biểu thức tính  $E_{\text{control}}$  như sau:

$$E_{\text{control}}(r) = \frac{N}{k} (qE_{\text{elec}} + qE_{\text{tworay}} d_{\text{toBS}}^4) + \frac{N}{k} qE_{\text{elec}} \quad (4.8)$$

Ở đây:

- $E_{\text{elec}}$  là năng lượng tiêu thụ của mạch điện bộ thu phát sóng vô tuyến
- $E_{\text{tworay}}$  là đơn vị năng lượng tiêu thụ của bộ khuếch đại trong mô hình truyền sóng hai tia mặt đất
- $d_{\text{toBS}}^4$  là khoảng cách từ nút mạng đến BS
- và  $q$  là số bit dữ liệu cần truyền giữa hai nút (xem thêm ở Công thức 3.6).

Biểu thức thành phần đầu tiên của công thức (4.8),  $\frac{N}{k} (qE_{\text{elec}} + qE_{\text{tworay}} d_{\text{toBS}}^4)$  tính năng lượng tiêu thụ của các nút trong cụm cho việc quảng bá thông điệp đến BS; biểu thức thành phần sau là năng lượng tiêu thụ cho việc nhận thông điệp từ BS. Các nút mạng quảng bá thông điệp đến BS, mà khoảng cách từ nút đến BS lớn hơn khoảng cách  $d_{\text{crossover}}$ , do đó, theo (3.6) ta có (4.8).

#### **Năng lượng tiêu thụ của các nút trong giai đoạn ổn định truyền dữ liệu**

(1).  $E_{\text{CH}}$ : Năng lượng tiêu thụ của CH, chúng gồm chi phí năng lượng nhận gói dữ liệu từ các nút hàng xóm, tổng hợp và chuyển tiếp gói dữ liệu đến SCH hoặc BS, được tính như dưới đây:

$$E_{\text{CH}} = pqE_{\text{elec}} + (p+1)qE_{\text{DF}} + qE_{\text{elec}} + qE_{\text{friis}} d_{\text{toSCH}}^2 \quad (4.9)$$

trong đó,  $p$  là số nút được kết nối tới CH. Để đơn giản cho việc tính toán mà không mất tính tổng quát, chúng tôi giả sử mỗi nút trong chuỗi chỉ kết nối với một nút hàng xóm ( $p=1$ ) khi đó, (4.9) trở thành (4.10)

$$E_{\text{CH}}(i) = 2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{friis}} d_{\text{toSCH}}^2 \quad (4.10)$$

Nếu CH là SCH thì công thức (4.10) được thay bằng (4.11) như dưới đây:

$$E_{\text{SCH}}(i) = 2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{tworay}} d_{\text{toBS}}^4 \quad (4.11)$$

(2).  $E_{\text{non-CH}}$ : Năng lượng tiêu thụ của nút không phải CH, chúng gồm tổng năng lượng tiêu thụ cho việc nhận, tổng hợp dữ liệu và truyền trong chuỗi cho một gói tin được mô tả bằng biểu thức dưới đây:

$$E_{\text{non-CH}} = \left(\frac{N}{k} - 2\right)qE_{\text{elec}} + \left(\frac{N}{k} - 1\right)qE_{\text{DA}} + \left(\frac{N}{k} - 1\right)(qE_{\text{elec}} + qE_{\text{friis}}d_{\text{ToNB}}^2) \quad (4.12)$$

Trong biểu thức ở vế phải (4.12):

- $d_{\text{toNB}}$  biểu diễn khoảng cách từ nút thành viên đến nút hàng xóm gần nhất trong cụm
- $E_{\text{elec}}$  là năng lượng yêu cầu để chạy mạch điện cho bộ thu phát sóng vô tuyến
- $E_{\text{friis}}$  biểu diễn đơn vị năng lượng yêu cầu cho bộ khuếch đại trong mô hình truyền không gian trống (free space)
- $E_{\text{DA}}$  biểu diễn chi phí năng lượng cho việc tổng hợp dữ liệu.

(3).  $E_{\text{round}}$ : Tổng năng lượng tiêu thụ cho một vòng trong một cụm được tính như:

$$E_{\text{round}} = E_{\text{control}} + m(E_{\text{SCH}} + E_{\text{non-CH}}) \quad (4.13)$$

trong đó,  $m$  là số gói tin được truyền trong giai đoạn truyền dữ liệu của một nút.

Thay (4.8), (4.11), (4.12), vào (4.13), ta nhận được:

$$E_{\text{round}} = \frac{N}{k}(qE_{\text{elec}} + qE_{\text{tworay}}d_{\text{toBS}}^4) + \frac{N}{k}qE_{\text{elec}} + m(2qE_{\text{elec}} + 2qE_{\text{DA}} + qE_{\text{tworay}}d_{\text{toBS}}^4) + m\left(\left(\frac{N}{k} - 2\right)qE_{\text{elec}} + \left(\frac{N}{k} - 1\right)qE_{\text{DA}} + \left(\frac{N}{k} - 1\right)(qE_{\text{elec}} + qE_{\text{friis}}d_{\text{ToNB}}^2)\right) \quad (4.14)$$

$$E_{\text{round}} = q\left(\left(\frac{2N(2m+1)}{k} - m\right)E_{\text{elec}} + \left(\frac{N}{k} + 1\right)mE_{\text{DA}} + m\left(\frac{N}{k} - 1\right)E_{\text{friis}}d_{\text{toNB}}^2 + (2m+1)E_{\text{tworay}}d_{\text{toBS}}^4\right) \quad (4.15)$$

Vì vậy, thời gian sống của mạng có thể được tính theo tổng số vòng là:

$$N_R = \frac{E_{\text{total}}}{kE_{\text{round}}} \quad (4.16)$$

$$E_{\text{total}} = \sum_{i=1}^N E_{\text{init}}(i) \quad (4.17)$$

Trong đó:

- $N$  là số nút cảm biến còn sống trong mạng
- $E_{\text{init}}(i)$  là năng lượng khởi tạo ban đầu của nút cảm biến thứ  $i$ .

Còn năng lượng tiêu thụ trung bình của vòng thứ  $r$  có thể được ước lượng như dưới đây [51, 124]:

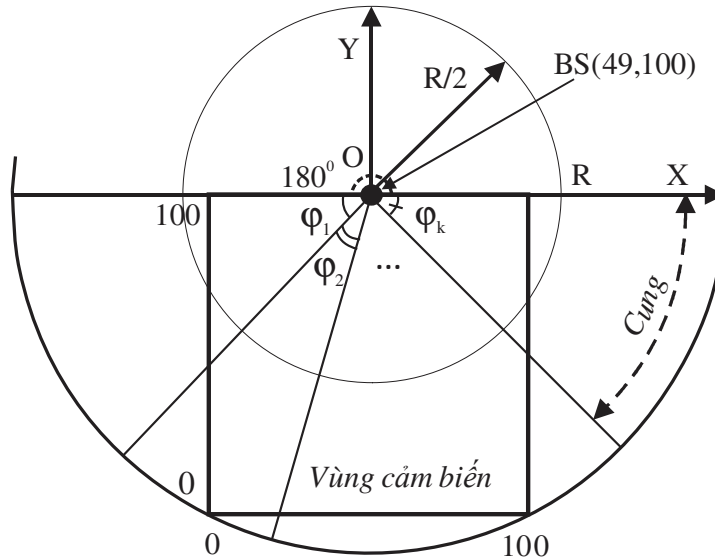
$$E_{\text{avg}}(r) = \frac{E_{\text{total}}}{N_R} \left(1 - \frac{r}{N_R}\right) \quad (4.18)$$

#### 4.6.1. Giai đoạn thiết lập cụm

##### Bước 1: Chia cụm (cung) cân bằng số nút

Đầu tiên, BS sẽ lấy thông tin vị trí, số định danh và năng lượng còn lại của tất cả các nút còn sống trong mạng bằng cách trao đổi thông điệp giữa BS và các nút cảm biến còn sống. Sau đó, BS phân chia toàn bộ vùng cảm biến thành  $k$  cung lô-gíc (tương đương  $k$  cụm), chúng cân bằng về số nút cảm biến cho mỗi cung như được minh họa trên Hình 4.6, trong đó mạng cảm biến được triển khai trong vùng có kích thước  $100\text{m} \times 100\text{m}$  và BS ở tọa độ  $(49,100)$ .

Để việc tính toán được đơn giản, chúng tôi dời gốc tọa độ OXY đến vị trí của BS ở vị trí  $(49,100)$  như Hình 4.6. Chúng tôi dựa vào số nút mạng còn sống để chia góc  $\varphi_i$  ( $i=1..k$ ) sao cho các cụm cân bằng nhau về số nút. Góc của mỗi nút cảm biến được tạo bởi vị trí địa lý của nút với trục OX có thể tính toán theo công thức dưới đây [36, 108]:



**Hình 4.6:** Phân chia mạng thành các cung (cụm)

$$\omega = \arctan\left(\frac{Y}{X}\right) \times \frac{180}{\pi}, \quad \text{với } Y = |Y - Y_{BS}|, X = |X - X_{BS}| \quad (4.19)$$

Ở đây, X và Y là tọa độ của nút trong hệ tọa độ với gốc là O, nếu tọa độ này là ở trong góc phần tư thứ ba, góc  $\varphi = 180 + \omega$ , nếu ở góc phần tư thứ tư thì  $\varphi = 360 + \omega$  tại vì ở tọa độ đó, giá trị hàm tang là âm.

---

#### **Thuật toán Chia cung (Thuật toán 4.4)**

---

**Input:** Các nút mạng, biến  $Cluster1 = \emptyset, Cluster2 = \emptyset, \dots, Clusterk = \emptyset$ , biến  $k=5$ .

**Output:** Các biến  $Cluster1, Cluster2, \dots, Clusterk$  chứa  $N/k$  nút mạng.

- 1:  $N \leftarrow$  Số lượng nút còn sống trong mạng
  - 2: **for each** node  $i$  in {Danh sách nút còn sống} **do**
  - 3: Gửi thông điệp HELLO tới BS, thông điệp này chứa ID, năng lượng còn lại và vị trí của nó.
  - end for**
  - 5: Tính toán góc cho tất cả các nút còn sống như biểu thức (4.19)
  - 6: Sắp xếp các nút theo thứ tự không giảm của góc
  - 7: Chia toàn bộ mạng thành  $k$  cung lấy BS làm gốc, mỗi cung gồm  $N/k$  nút cảm biến dựa vào góc của chúng như Hình 4.6
  - 8: Chuyển sang **Thuật toán Chọn nút CH lãnh đạo chuỗi (Thuật toán 4.5)**
- 

#### **Bước 2: Chọn nút lãnh đạo chuỗi (CH)**

Tại vòng thứ  $r$  của thuật toán SCBC, BS sẽ lựa chọn nút CH cho mỗi cụm, chúng có mức năng lượng lớn hơn mức năng lượng trung bình của mỗi cụm  $E_{avg}()$  và có giá trị hàm  $cost()$  đạt giá trị cực đại.  $E_{avg}$  và hàm  $cost()$  được tính như (4.20) và (4.21) dưới đây:

$$E_{avg}(cl) = \frac{1}{nn} \sum_{j=1}^{nn} E_{residual}(j) \quad (4.20)$$

$$cost(cl) = \text{Max}_i \left\{ ec_1 * E_{residual}(i) + \frac{dc_2}{d(i, BS) + \sum_{j=1}^h d(i, j)} \right\} \quad (4.21)$$

Trong đó:

- $nn$  là tổng số nút còn sống trong mỗi cụm
- $E_{\text{residual}}(i)$  là năng lượng còn lại của nút cảm biến thứ  $i$  trong cụm  $cl$  tương ứng.
- $h$  là số nút hàng xóm của nút thứ  $i$
- $d(i, j)$  là khoảng cách địa lý từ nút ứng viên thứ  $i$  tới nút thứ  $j$ , nó được tính như (3.5). Hơn nữa, các giá trị  $ec_1 (=100/J)$  và  $dc_2 (=0.3m)$  là giống với (4.7).

### **Bước 3: Chọn nút CH thứ cấp (SCH)**

Trong thuật toán SCBC, chỉ có một vài nút cụm trưởng thứ cấp (SCH) chuyển tiếp các gói dữ liệu đến BS để tiết kiệm năng lượng cho các nút cụm trưởng (CH) khác ở xa. Vì hoạt động truyền dữ liệu ở xa là một trong các hoạt động tiêu thụ năng lượng nhiều nhất trong WSNs. Do đó, để chọn được các nút SCHs có khoảng cách từ nó đến BS càng gần thì càng tốt, chúng tôi kiểm tra nếu khoảng cách từ CH đến BS là nhỏ hơn khoảng cách trung bình  $D_{\text{avg}}$  giữa các CHs và BS và nhỏ hơn  $R/2$  như Hình 4.6 thì nút CH được chọn làm SCH. Giá trị  $D_{\text{avg}}$  có thể được tính như (4.22) dưới đây:

$$D_{\text{avg}} = \frac{1}{k} \sum_{i=1}^k d(\text{CH}_i, \text{BS}) \quad (4.22)$$

Trong đó,  $k$  là số cụm (cung),  $d(\text{CH}_i, \text{BS})$  là khoảng cách từ nút  $\text{CH}_i$  đến BS và số nút SCHs được chọn là nhỏ hơn nửa số lượng CHs để cân bằng tiêu thụ năng lượng. Thuật toán 4.5 chọn nút cụm trưởng cho mỗi cụm với ý tưởng như sau: Giai đoạn 1: Ứng với mỗi nút trong cụm, tính năng lượng trung bình như biểu thức (4.20), sau đó tính toán, so sánh để chọn nút có giá trị hàm  $\text{cost}()$  lớn nhất như trong công thức (4.21). Giai đoạn 2: Ứng với mỗi nút  $\text{CH}_i$  trong danh sách CHs, kiểm tra nếu  $d(\text{CH}_i, \text{BS}) < D_{\text{avg}}$  và  $d(\text{CH}_i, \text{BS}) < R/2$  và (số lượng SCHs)  $< k/2$  thì thêm nút CH vào trong danh sách các nút SCH. Kết thúc, thuật toán trả về nút cụm trưởng CH và SCH trong các biến đầu ra.





Chúng ta thấy tỉ lệ gói tin nhận được ở BS hay thông lượng mạng  $Q$  trong khoảng thời gian mạng hoạt động sẽ tăng nếu ta giảm số lần thiết lập cụm (hay giảm số vòng  $N_R$ ) và tăng chiều dài thời gian ( $t_{\text{round}}$ ) trong giai đoạn truyền dữ liệu của mỗi vòng. Tuy nhiên, nếu chúng ta tăng chiều dài thời gian của mỗi vòng quá dài, năng lượng của nút SCHs sẽ hết nhanh hơn các nút khác, vì chúng phải xử lý nhiều và chuyển tiếp gói dữ liệu đến BS xa hơn các nút khác. Do đó, hàm mục tiêu với biến  $N_R$  có thể được diễn tả như:

$$f = \underset{N_R}{\operatorname{argmax}}(Q) = \underset{N_R}{\operatorname{argmax}}(N_R \times m) \quad (4.25)$$

Trong đó,  $\operatorname{arg max}$  (arguments of the maximum) là làm cho  $Q$  đạt giá trị cực đại hay tìm các giá trị của  $N_R$  để  $Q$  đạt cực đại [62]. Theo (4.23) và (4.24),  $Q$  đạt cực đại khi và chỉ khi  $E_{\text{total}}$  cực đại hay  $k \cdot (E_{\text{control}} + m(E_{\text{SCH}} + E_{\text{non-CH}}))$  cực tiểu, mà  $E_{\text{total}}$  và  $k$  là cố định, hơn nữa, ta cũng mong muốn  $m$  (số gói tin một nút mạng được truyền trong giai đoạn truyền dữ liệu) cực đại. Do đó, ta có thể làm cực tiểu  $E_{\text{control}}$  hay giảm năng lượng tiêu thụ dành cho giai đoạn khởi tạo cụm ( $E_{\text{elec}}$ ,  $E_{\text{tworay}}$ ,  $d_{\text{toBS}}^4$ ) và tăng thời gian ổn định truyền dữ liệu tại mỗi vòng ( $m$ ,  $E_{\text{CH}}$ ,  $E_{\text{non-CH}}$ ), ta sẽ được thông lượng lớn nhất.

Theo công thức (3.6) để tính năng lượng tiêu thụ cho việc truyền  $q$  bit giữa 2 nút ở khoảng cách  $d$ , công thức (3.8) để tính năng lượng tiêu thụ cho việc nhận  $q$  bit và các số liệu của Bảng 3.1 về chi phí năng lượng để chạy mạch điện cho bộ thu phát sóng vô tuyến ..., chúng ta có thể tính toán chiều dài thời gian cho giai đoạn ổn định truyền dữ liệu với  $m$  gói tin.

$$E_{\text{SCH}}(r) - m(E_{\text{tx}}(q, d) + E_{\text{rx}}(q, d) + E_{\text{DF}}) \geq E_{\text{threshold}} \quad (4.26)$$

Trong đó,  $E_{\text{threshold}}$  là ngưỡng năng lượng, nó được cố định bởi người dùng tùy theo ứng dụng mạng và lớn hơn không để đảm bảo sau khi kết thúc vòng, nút SCH là vẫn còn sống. Đặt  $t_{\text{round}}$  chỉ định chiều dài thời gian của mỗi vòng,

$$t_{\text{round}} = \psi + m \cdot T_{\text{frame}} \quad (4.27)$$

ở đây,  $\psi$  là khoảng thời gian trong giai đoạn thiết lập cụm,  $T_{\text{frame}}$  là khoảng thời gian một gói tin được truyền. Để thấy được ảnh hưởng của  $t_{\text{round}}$  tới hiệu năng của các giao thức định tuyến, chúng tôi đã tiến hành mô phỏng các giao thức sử dụng 100

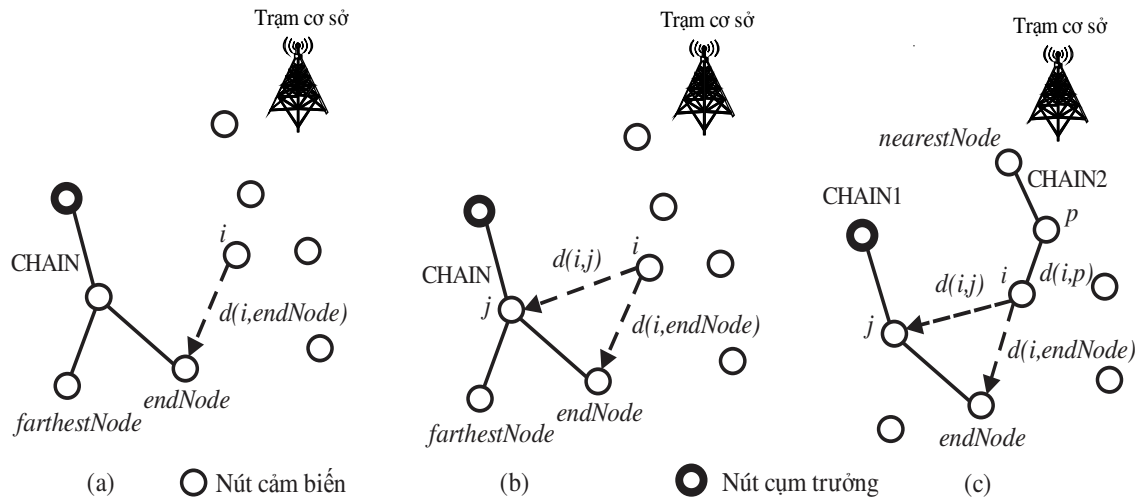
kịch bản đã tạo như ở Chương 3, Mục 3.3 với việc thay đổi thời gian trong giai đoạn ổn định truyền dữ liệu. Trên Bảng 4.1 chúng tôi trình bày kết quả đánh giá và so sánh giá trị trung bình tỉ lệ nút chết và tỉ lệ gói tin nhận được ở BS với các giá trị  $t_{\text{round}}$  được thay đổi từ 10 đến 700 (giây). Sau khi chúng tôi thực hiện mô phỏng từ Bước 2 đến Bước 7 giống với Chương 3, Mục 3.3 để tính toán giá trị trung bình của các kết quả sau khi chạy 21 kịch bản áp dụng với giao thức SCBC tỉ lệ độ lệch chuẩn ( $\xi=4,4\%$ ), với giao thức IEEPB số kịch bản đã chạy là  $n_{\text{sc}}=15$  tỉ lệ độ lệch chuẩn ( $\xi=3,4\%$ ), với EE-TLDC số kịch bản đã chạy là  $n_{\text{sc}}=15$  tỉ lệ độ lệch chuẩn ( $\xi=6,4\%$ ) và giao thức LEACH có số kịch bản đã chạy là  $n_{\text{sc}}=10$  tỉ lệ độ lệch chuẩn ( $\xi=7,7\%$ ). Các kết quả cho thấy rõ ràng rằng, nếu  $t_{\text{round}}$  tăng thì tỉ lệ gói tin nhận được ở BS cũng sẽ tăng lên nhưng nút đầu tiên chết sẽ sớm hơn. Theo kết quả mô phỏng của chúng tôi trong Bảng 4.1, rõ ràng giá trị trung bình của SCBC là tốt nhất với nút đầu tiên chết ở 522 (giây) khi  $t_{\text{round}}$  bằng 50 (giây) trong mạng gồm 100 nút cảm biến và vị trí BS ở (49,175).

**Bảng 4.1:** Tỉ lệ gói tin nhận được ở BS và tỉ lệ nút chết khi thay đổi  $t_{\text{round}}$

Giao thức	$t_{\text{round}}$ (giây)	Tỉ lệ gói tin nhận được ở BS (%)	Phần trăm các nút chết					
			1%	10%	25%	50%	75%	100%
LEACH	10	38,18	238	348	438	526	633	1607
	20	38,00	139	294	396	489	579	1019
	50	39,03	33	142	293	444	553	824
	100	61,87	37	283	473	740	918	1206
	200	83,86	31	343	773	1075	1265	1606
EE-TLDC	10	49,29	417	604	714	864	999	1311
	20	61,46	283	511	688	874	1039	1256
	50	67,75	57	426	615	977	1227	1401
	150	89,40	47	379	862	1329	1630	1950
PEGASIS	10	56,56	456	598	702	796	847	999
	50	73,43	96	616	878	937	1097	1372
	100	79,97	91	796	955	993	1218	1564
	300	85,43	79	974	1015	1068	1445	1942

	500	87,95	86	1006	1034	1059	1603	2354
IEEPB	10	60,61	162	376	596	816	1021	1470
	50	77,57	157	460	754	963	1205	1725
	100	82,51	134	478	793	1000	1273	1693
	500	95,84	142	528	958	1262	1681	2418
	700	96,21	123	888	945	1322	1733	2460
SCBC	10	81,04	435	648	765	914	1178	2476
	50	91,99	522	732	868	1015	1237	2266
	200	96,15	277	689	890	1099	1437	2174
	500	98,94	308	613	909	1184	1557	2418
	700	100	189	621	897	1243	1626	2681

### Bước 5: Xây dựng cụm chuỗi

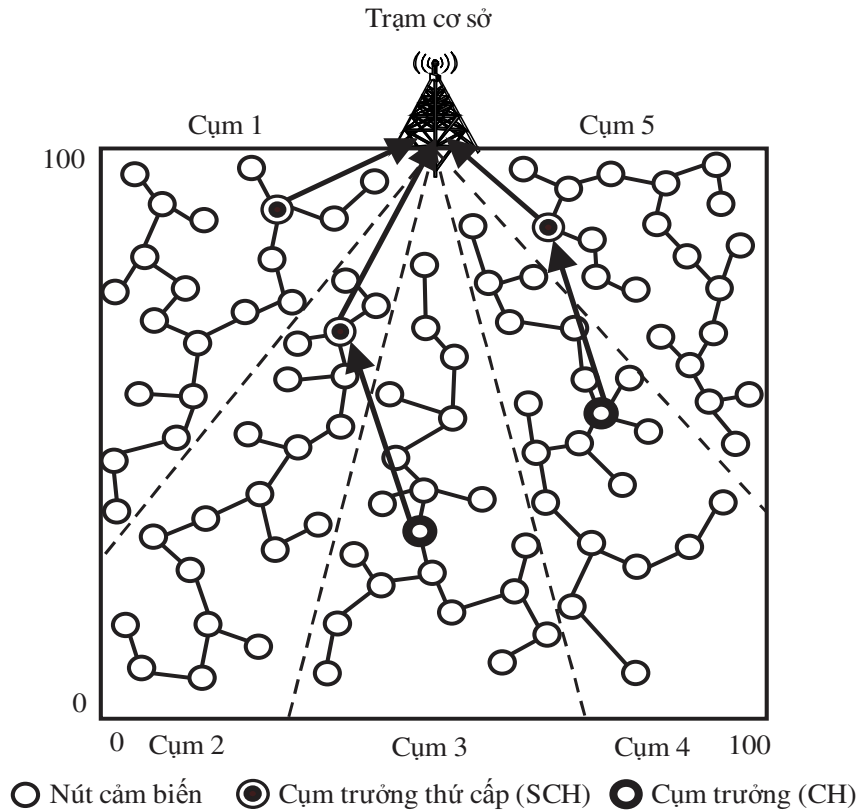


**Hình 4.7:** Một ví dụ xây dựng chuỗi giữa các thuật toán: (a) PEGASIS, (b) IEEPB và (c) SCBC

Hình 4.7 đưa ra ví dụ minh họa xây dựng cụm chuỗi theo các thuật toán PEGASIS, IEEPB và thuật toán SCBC do chúng tôi đề xuất, trong đó PEGASIS và IEEPB bắt đầu xây dựng chuỗi với nút xa BS nhất, trong khi SCBC bắt đầu với nút CH cho chuỗi 1 và nút gần BS nhất cho chuỗi 2.

Hình 4.8 là một thí dụ minh họa kết quả phân cụm mạng gồm 100 nút cảm biến sau khi xây dựng 5 chuỗi (cụm) theo thuật toán SCBC trong một vòng. Như quan sát

được, ở đây, giao thức được chúng tôi đề xuất không chỉ có liên kết truyền thông ngắn mà còn cân bằng về số lượng nút (20 nút) cho mỗi chuỗi trong mạng. Các mũi tên hướng từ CH đến SCH chỉ hướng truyền dữ liệu từ CH đến SCH, hơn nữa, Hình 4.8 cũng hiển thị kết quả của việc chọn CHs, SCHs và xây dựng chuỗi như trong Thuật toán 4.5 dựa trên thuật toán tham lam (GA-Greedy Algorithms).



**Hình 4.8:** Kết quả phân cụm mạng sử dụng giao thức SCBC trong một vòng

Ý tưởng của thuật toán là: Sử dụng hai biến CHAIN1 bắt đầu bởi nút CH, CHAIN2 bắt đầu bởi nút xa nhất trong cụm. Để kết nối một nút  $i$  trong cụm vào chuỗi, nút  $i$  sẽ chọn kết nối đến nút  $j$  (nút  $j$  thuộc CHAIN1 hoặc CHAIN2) sao cho khoảng cách từ nút  $i$  đến  $j$  là nhỏ nhất. Lặp lại việc này cho đến khi kết nạp hết tất cả các nút vào chuỗi. Cuối cùng là kết nối hai chuỗi CHAIN1 và CHAIN2 thành một chuỗi bằng cách kết nối phần tử  $u$  trong CHAIN1 và  $nearestNode$  trong CHAIN2 sao cho khoảng cách từ  $u$  đến  $nearestNode$  là nhỏ nhất, nội dung Thuật toán 4.6 như dưới đây:

---

**Thuật toán Xây dựng cụm chuỗi dựa trên GA (Thuật toán 4.6)**

---

**Input:** Biến  $Cluster(c)$ ,  $CH$  với  $(c=1..k)$  chứa  $N/k$  nút mạng.

**Output:** Chuỗi lưu trong biến  $CHAIN$ .

```
1:  $endNode \leftarrow CH$ ;           //  $endNode$  bắt đầu từ nút cụm trưởng thứ  $i$ 
2:  $CHAIN1 \leftarrow endNode$ ;      // chuỗi  $CHAIN1$  bắt đầu bằng nút cụm trưởng
    $CHAIN2 \leftarrow nearestNode$ ; // chuỗi  $CHAIN2$  bắt đầu bởi nút gần BS nhất
4: while ({Danh sách nút còn sống trong cụm}  $\neq \emptyset$ ) do
5:   Tìm kiếm nút  $i$  trong {Danh sách nút còn sống}, nút có  $d(i, endNode)$  là nhỏ
                                     nhất;
6:   Tìm kiếm nút  $j$  trong { $CHAIN1$ }, nút có  $d(i, j)$  là nhỏ nhất;
   Tìm kiếm nút  $p$  trong { $CHAIN2$ }, nút có  $d(i, p)$  là nhỏ nhất;
8:    $minijp \leftarrow \min\{d(i, endNode), d(i, j), d(i, p)\}$ 
   if  $minijp = d(i, p)$  then
10:     Thêm nút  $i$  vào trong { $CHAIN2$ } bằng cách kết nối nút  $i$  với nút  $p$ ;
   else
12:     if  $d(i, j) < d(i, endNode)$  and  $(j \neq endNode)$  then
13:       Thêm nút  $i$  vào trong { $CHAIN1$ } bằng cách kết nối nút  $i$  với nút  $j$ ;
     else
15:       Thêm nút  $i$  vào trong { $CHAIN1$ } bằng cách kết nối nút  $i$  với  $endNode$ ;
16:     end if
17:      $endNode \leftarrow node\ j$ ;
18:   Xóa nút  $i$  trong {Danh sách nút còn sống};
   end while
20:  Tìm kiếm nút  $u$  trong { $CHAIN1$ }, nút có  $d(u, nearestNode)$  là nhỏ nhất;
   Kết nối  $CHAIN \leftarrow CHAIN1$  và  $CHAIN2$  bằng cách nối  $nearestNode$  với  $u$ 
   Tạo lịch TDMA cho tất cả các nút trong mỗi cụm
23:  Quảng bá lịch TDMA và  $CHAIN$  tới các nút trong mạng
24: return  $CHAIN$ ;
```

---

#### 4.6.2. Giai đoạn cảm biến và truyền dữ liệu

Sau khi hoàn thành giai đoạn thiết lập cụm và xây dựng cụm chuỗi ở trên, các gói dữ liệu bắt đầu được truyền, giống như PEGASIS, mỗi nút sẽ truyền dữ liệu nó cảm biến được tới nút kế tiếp về phía CH trong khe thời gian của nó. Các khe thời gian này được xác định theo kỹ thuật TDMA và được BS gán cho các nút mạng dọc theo chuỗi. Đầu tiên, nút xa nhất trong chuỗi truyền gói dữ liệu của nó đến nút hàng xóm gần nhất. Nút trung gian trong chuỗi nhận gói dữ liệu, tổng hợp một hoặc nhiều gói nhận được với gói dữ liệu nó cảm biến thành một gói duy nhất và chuyển tiếp cho nút hàng xóm kế tiếp hoặc nút CH hoặc SCH dọc theo chuỗi. Khi nào nút SCH nhận được tất cả các gói dữ liệu, nó sẽ chuyển tiếp đến BS theo mã CDMA sau khi đã tổng hợp theo cách tương tự. Sau khoảng thời gian  $t_{round}$ , vòng kế tiếp sẽ được bắt đầu bằng cách phân chia lại cung, chọn lại CH, SCH cũng như xây dựng lại chuỗi cho vòng mới.

#### 4.7. Mô phỏng để đánh giá hiệu quả của giao thức đề xuất SCBC

Về độ phức tạp tính toán của SCBC, các Thuật toán 4.4, 4.5 và 4.6 mỗi nút chỉ thực hiện truyền và nhận một thông điệp đến BS, do đó, độ phức tạp thông báo ở mỗi nút cảm biến là hằng số. Các thao tác tính toán được thực hiện bởi BS, Thuật toán 4.5 thực hiện công việc tính hàm  $cost()$  theo (4.21) để chọn nút CH có độ phức tạp tính toán là  $O(n/k)$ . Thuật toán 4.6 xây dựng cụm chuỗi sử dụng thuật toán tham lam có độ phức tạp tính toán là  $O[(n/k)\log(n/k)]$ . Do đó SCBC có độ phức tạp tính toán là  $O(n\log(n))$  và độ phức tạp thông báo là hằng số.

Để kiểm chứng và đánh giá hiệu năng của SCBC, chúng tôi đã tiến hành cài đặt mô phỏng SCBC, EE-TLDC và IEEPB bằng bộ công cụ mô phỏng mạng mã nguồn mở ns-2 (v.2.34), sử dụng kịch bản với các hằng số  $M_1=0,5$ ,  $M_2=0,4$ ,  $\alpha=0,5$ ,  $\beta=2$  [51]. Trong các giao thức này, chúng tôi giả sử rằng mô hình mạng cảm biến hỗn hợp gồm  $N$  nút cảm biến, chúng được dùng với ba kiểu nút khác nhau: Nút thông thường (normal), nút nâng cao (advanced) và siêu nút (super sensor node). Ba kiểu nút này chỉ khác nhau ở dung lượng pin khởi tạo. Nếu gọi  $E_{init}$  là năng lượng khởi tạo của tất cả các nút mạng thì:  $E_{init} = 1J$ ,  $E_{init}*(1+\alpha)$  và  $E_{init}*(1+\beta)$  là năng lượng

khởi tạo của nút nâng cao và siêu nút [51]. Do đó, ta có tổng năng lượng khởi tạo ban đầu cho toàn mạng là:

$$E_{\text{Total}} = E_0(N_N + N_A(1 + \alpha) + N_S(1 + \beta)) \quad (4.28)$$

$$N_S = N \times M_1 \times M_2, \quad N_A = N \times M_1(1 - M_2), \quad N_N = N(1 - M_1) \quad (4.29)$$

và tổng số nút mạng là:  $N = N_N + N_A + N_S \quad (4.30)$

- Trong đó:
- $N_N$  là tổng số nút thông thường trong mạng
  - $N_A$  là tổng số nút nâng cao trong mạng
  - $N_S$  là tổng số siêu nút trong mạng
  - $\alpha, \beta$  là hệ số năng lượng tăng thêm cho nút nâng cao và siêu nút
  - $M_1, M_2$  chỉ định hệ số nút nâng cao và siêu nút trong mạng.

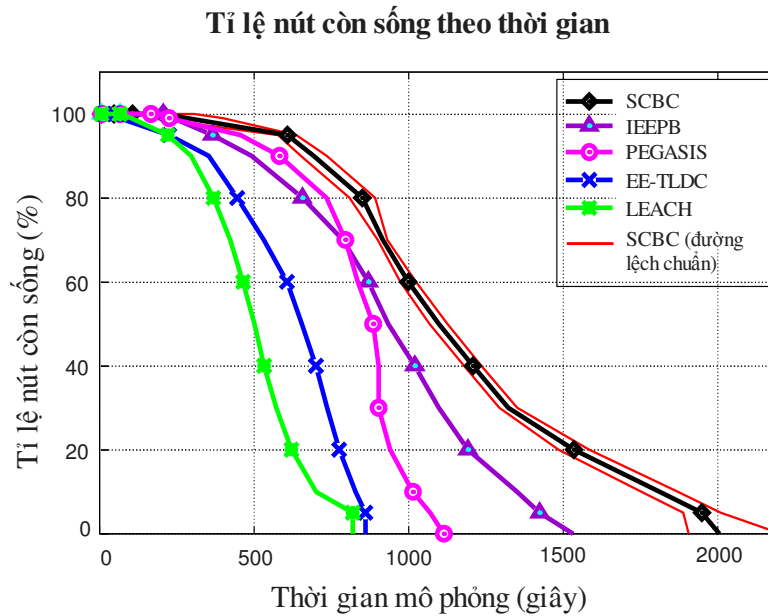
Chúng tôi đã sử dụng 100 kịch bản được sinh ra ở Chương 3, Mục 3.3 để mô phỏng, đánh giá hiệu năng của các giao thức LEACH, EE-TLDC, IEEPB và SCBC. Các kịch bản này có tô-pô mạng gồm 100 nút cảm biến được sinh ngẫu nhiên nhiều lần trong vùng cảm biến có diện tích 100m×100m. Dựa vào các công thức (4.28), (4.29) và (4.30), chúng tôi thiết lập năng lượng khởi tạo của siêu nút ( $N_S$ ) có năng lượng khởi tạo = 3J ( $\beta=2$ ), nút nâng cao ( $N_A$ ) có mức năng lượng khởi tạo = 1,5J ( $\alpha=0,5$ ) và nút thông thường ( $N_N$ ) có mức năng lượng = 1J. Tổng số nút trong kịch bản gồm  $N=100$ , ( $N_N=50$ ,  $N_A=30$ ,  $N_S=20$ ) và tổng năng lượng khởi tạo bằng 155J và các tham số mô phỏng khác giống với Bảng 3.1 [6, 34, 110].

#### 4.7.1. Phân tích, đánh giá và so sánh thời gian sống của mạng khi áp dụng các giao thức PEGASIS, IEEPB và SCBC

Để đánh giá hiệu năng của giao thức SCBC, chúng tôi đã thực hiện các bước giống với các bước ở Chương 3, Mục 3.3 (từ Bước 2 đến Bước 7) để xác định số kịch bản ( $n_{sc}$ ) cần phải chạy mô phỏng bao nhiêu là đủ? Dựa vào bảng số liệu trình bày giá trị trung bình  $m_x$ , độ lệch chuẩn  $\sigma$  và tỉ lệ độ lệch chuẩn  $\xi$  theo các công thức (3.11), (3.12) và (3.13), chúng tôi chọn giá trị  $\xi$  với giao thức PEGASIS bằng 3,1%, EE-TLDC bằng 6,4%, IEEPB bằng 3,4% và SCBC là 2,3% tại thời điểm mạng có 95% nút chết thì số kịch bản mô phỏng như sau: Với giao thức PEGASIS, số kịch bản đã chạy  $n_{sc}=14$  kịch bản thì dừng, với giao thức EE-TLDC số kịch bản đã chạy là  $n_{sc}=$

15, với giao thức IEEPB số kịch bản đã chạy là  $n_{sc}=15$  và với giao thức SCBC số kịch bản đã chạy là  $n_{sc}=21$  kịch bản thì dùng mô phỏng.

Giá trị trung bình nhận được từ mô phỏng được chúng tôi xử lý và trình bày bằng đồ thị, trên các hình từ Hình 4.9 đến Hình 4.11 dưới đây:



**Hình 4.9:** Tỉ lệ nút còn sống trong suốt thời gian mô phỏng

Hình 4.9 hiển thị kết quả mô phỏng các giao thức PEGASIS, IEEPB và SCBC với tỉ lệ nút còn sống theo thời gian mô phỏng. Đồ thị biểu diễn kết quả cho thấy rõ ràng giao thức SCBC, nhờ kết hợp linh hoạt giữa cân bằng số lượng nút trong mỗi chuỗi và khoảng thời gian ổn định truyền dữ liệu của mỗi vòng, làm thời gian sống của mạng tăng thêm khoảng 20% và 15% khi so sánh với giao thức PEGASIS và IEEPB.

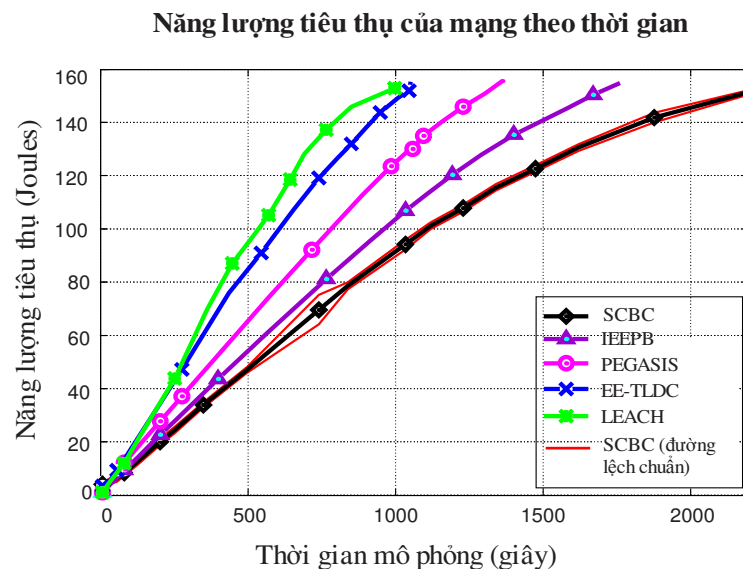
#### 4.7.2. Phân tích, đánh giá và so sánh năng lượng tiêu thụ của mạng khi áp dụng các giao thức PEGASIS, IEEPB và SCBC

Giá trị trung bình nhận được từ mô phỏng trên nhiều kịch bản khác nhau tương ứng với tỉ lệ độ lệch chuẩn của giao thức PEGASIS bằng 3,1%, EE-TLDC bằng 6,4%, IEEPB bằng 3,4% và SCBC là 2,3% được chúng tôi xử lý và trình bày bằng đồ thị, trên Hình 4.10. Dựa trên kết quả mô phỏng trong Hình 4.10 cho thấy giao thức

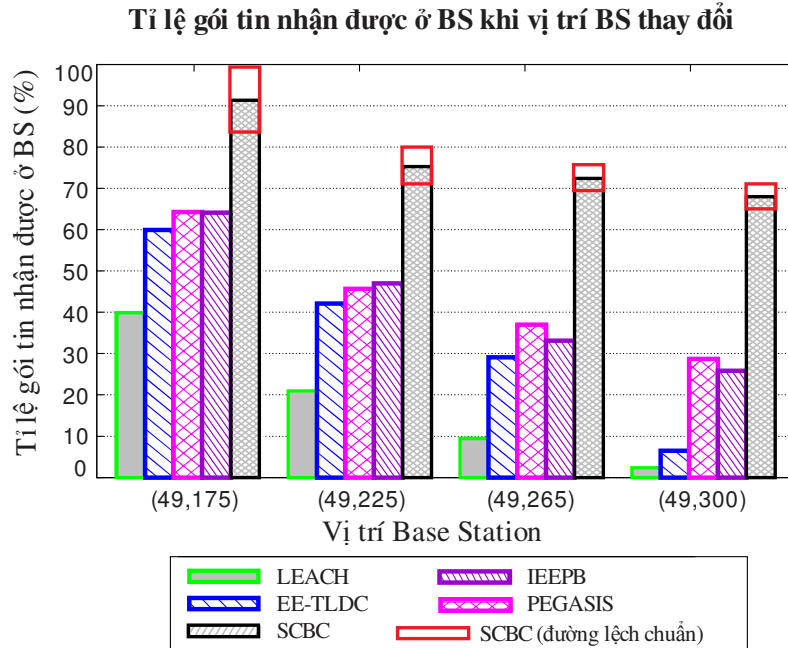


SCBC với thuật toán xây dựng cụm chuỗi kết hợp với chọn nút SCH để giảm khoảng cách truyền dữ liệu trong mạng, do đó, giảm năng lượng tiêu thụ cho các nút. Các đường đồ thị trong Hình 4.10 càng gần trục X, năng lượng tiêu thụ càng ít, càng tốt. Do đó, SCBC giúp đạt được hiệu quả sử dụng năng lượng tốt hơn và nâng cao thời gian sống mạng cảm biến khi so sánh với PEGASIS và IEEPB.

Hình 4.11 biểu diễn bằng đồ thị tỉ lệ gói tin nhận được ở BS khi vị trí địa lý của BS được thay đổi từ vị trí (49,100) tới vị trí xa nhất (49, 265) so với vùng mạng cảm biến. Rõ ràng, chúng có mối quan hệ giảm khi di chuyển vị trí địa lý của BS ra xa hơn vì các nút mạng phải truyền dữ liệu xa, tiêu tốn nhiều năng lượng cho bộ khuếch đại trong mô hình truyền thông vô tuyến. Tuy nhiên, tỉ lệ gói tin nhận được ở BS đối với SCBC vẫn tốt hơn PEGASIS và IEEPB. Các đường màu đỏ trên các hình biểu diễn độ lệch chuẩn so với giá trị trung bình khi áp dụng giao thức SCBC có  $\xi$  bằng 2,3% tương ứng với số kích bản đã chạy  $n_{sc}=21$ , điều này có nghĩa là nếu chúng ta có tăng thêm mô phỏng thì kết quả thu được cũng không vượt ra ngoài đường lệch chuẩn.



**Hình 4.10:** Năng lượng tiêu thụ của mạng theo thời gian



**Hình 4.11:** Tỉ lệ gói tin nhận được ở BS khi vị trí của BS thay đổi

#### 4.8. Phân tích và so sánh với các thuật toán cùng hướng khác

Các thuật toán xây dựng cụm chuỗi trong [7, 94, 101, 104, 117, 118] chưa xem xét đến phân chia đều số nút cảm biến vào các cụm và khoảng thời gian tồn tại trong giai đoạn truyền dữ liệu trong mỗi vòng, do đó hiệu quả sử dụng năng lượng chưa cao. Đề xuất SCBC của chúng tôi phân chia tương đối đều số nút cảm biến vào các cụm và tính toán phù hợp khoảng thời gian tồn tại trong giai đoạn truyền dữ liệu của mỗi vòng, hơn nữa, thuật toán xây dựng cụm chuỗi SCBC giảm liên kết dài trong chuỗi bằng cách kết hợp hai chuỗi với nhau, chúng được tạo ra bằng cách kết nối các nút có khoảng cách gần nhất trong cụm, nhờ đó SCBC cho hiệu quả sử dụng năng lượng tốt hơn hẳn các giao thức đã có.

#### 4.9. Tổng kết chương

DFCB, một lược đồ xây dựng chuỗi dài kết hợp với tổng hợp dữ liệu đã được chúng tôi đề xuất. DFCB giảm đáng kể năng lượng sử dụng trong truyền dữ liệu giữa các nút trong chuỗi dựa trên sự kết nối giữa các nút hàng xóm gần nhất với nhau và tổng hợp dữ liệu xảy ra ở các nút trong chuỗi. Chúng tôi đã thực hiện các mô phỏng

và việc phân tích các kết quả mô phỏng cho thấy, DFCB cho hiệu quả sử dụng năng lượng tốt hơn thuật toán phân cụm LEACH và PEGAGIS truyền thống. Tuy nhiên, việc áp dụng DFCB có thể dẫn đến việc xảy ra hiện tượng thắt nút cổ chai do tất cả các nút trong mạng kết nối và truyền thông trong một chuỗi dài. Do đó, DFCB chỉ phù hợp với các ứng dụng mạng không đòi hỏi độ trễ truyền thông thấp. Để khắc phục điểm yếu này, chúng tôi đề xuất tiếp giao thức SCBC nhằm giảm độ trễ và nâng cao hiệu quả sử dụng nguồn năng lượng dựa trên sự cân bằng số nút trong mỗi cụm và tăng khoảng thời gian truyền dữ liệu trong mỗi vòng mà chúng vẫn giữ được hoạt động ổn định của mạng. Việc tối ưu hai tham số này trong lược đồ định tuyến phân cụm (cung) dựa trên chuỗi dẫn đến thuật toán định tuyến mới đạt hiệu quả tốt hơn. Thông qua các kết quả mô phỏng được thể hiện bằng đồ thị trên các Hình 4.9, 4.10 và 4.11 cho thấy hiệu quả tiêu thụ năng lượng của giao thức được chúng tôi đề xuất cải tiến tốt hơn các giao thức đã có, làm tăng thêm khoảng 20% và 15% thời gian sống cho mạng khi so sánh với giao thức PEGASIS và IEEPB. Các kết quả nghiên cứu trong Chương 4 tương ứng với công trình 4 và 5 đã công bố.

## **Chương 5: ĐỊNH TUYẾN TIẾT KIỆM NĂNG LƯỢNG DỰA TRÊN CÂY TỐI THIỂU**

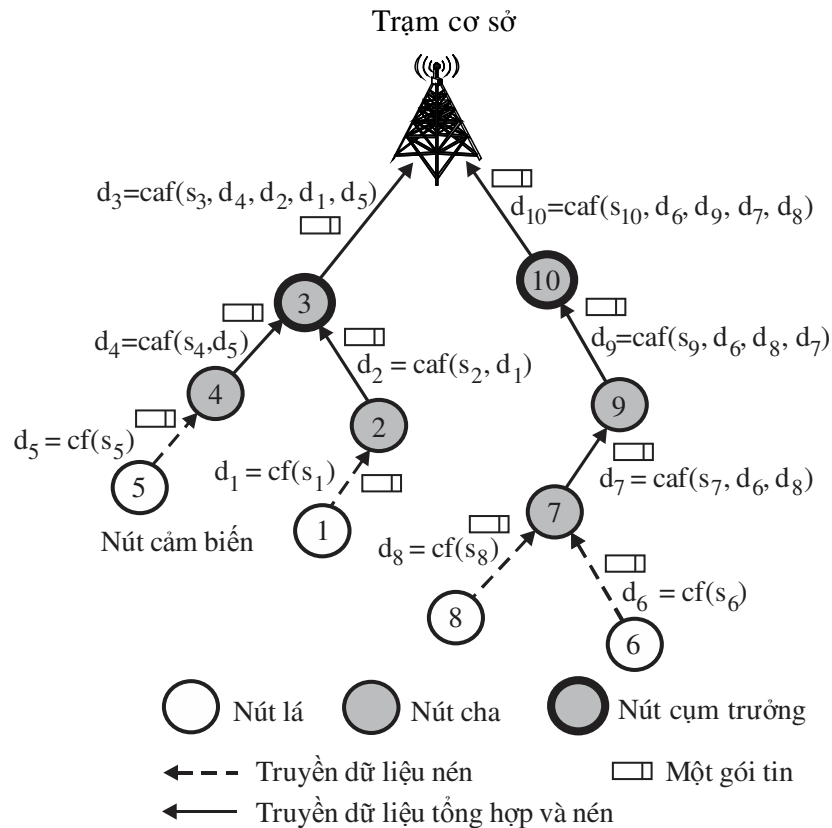
Trong chương này, chúng tôi trình bày hai đề xuất cải tiến giao thức định tuyến tiết kiệm năng lượng dựa trên cây khung tối thiểu; đề xuất thứ nhất có tên viết tắt là DFTBC (Data Fusion and Tree-Based Clustering Routing Protocol). DFTBC kế thừa ưu điểm của các đề xuất [70, 104, 118] là chia toàn bộ vùng cảm biến thành 5 cụm cố định tương ứng với 5% nút cảm biến cho mỗi cụm. Mỗi cụm chọn một nút làm CH có xem xét đến khoảng cách và năng lượng còn lại của nó. Quá trình truyền dữ liệu được thực hiện bởi các nút bên trong cụm với CH và nút CH với BS, do đó giảm độ trễ truyền thông, kéo dài thời gian sống cho mạng. Tuy nhiên, thuật toán xây dựng chuỗi cho mỗi cụm trong các đề xuất khó tránh khỏi có các liên kết dài trong chuỗi, do đó đạt hiệu quả về năng lượng chưa cao. Để khắc phục điểm yếu này, chúng tôi sử dụng thuật toán Prim [27] để kết nối các nút cảm biến với nhau thành cây với nút CH chính là nút gốc, nhờ đó giảm được khoảng cách truyền thông tổng thể trong cụm, nâng cao hiệu quả sử dụng năng lượng.

Đề xuất thứ hai của chúng tôi, được gọi tắt là SSTBC (Sleep Scheduled and Tree-Based Clustering) dựa trên [70], theo đề xuất này lược đồ định tuyến phân cụm dựa trên cây kết hợp với lập lịch ngủ để nâng cao hiệu quả sử dụng năng lượng. Theo đề xuất này, toàn bộ vùng cảm biến mạng được chia ảo (tưởng tượng) thành các ô lưới nhỏ, kích thước các ô lưới ảo được thiết lập bởi người dùng thông qua giá trị ngưỡng sao cho phù hợp với yêu cầu ứng dụng và tiết kiệm năng lượng mạng cảm biến. Trong mỗi ô lưới nhỏ, chỉ có một nút ở chế độ hoạt động trong vòng hiện tại, trong khi đó các nút khác sẽ đi vào chế độ ngủ để tiết kiệm năng lượng.

### **5.1. Lược đồ định tuyến kết hợp với tổng hợp dữ liệu trên cây**

Như chúng tôi đã phân tích trong Mục 4.2 Chương 4, để tiết kiệm năng lượng cho các nút mạng cảm biến trong khi truyền dữ liệu, ta có thể sử dụng mã nguồn phân tán DSC để nén hai nguồn dữ liệu tương quan và để giảm số lượng gói dữ liệu dư

thừa chuyển tiếp về BS và lấy về thông tin quan sát chính xác bằng cách thực hiện tổng hợp dữ liệu ở các nút CH đối với mô hình truyền đơn chặng trong cụm hoặc ở các nút trung gian đối với mô hình chuỗi. Việc tổng hợp dữ liệu cũng có thể được thực hiện trên mô hình cây phân cấp như được minh họa trên Hình 5.1 dưới đây [2, 71, 106]. Theo mô hình này, các nút lá trên cây thực hiện quan sát, sinh ra gói dữ liệu, nén nó bằng DSC và gửi đến nút cha. Các nút cha thực hiện thu nhận các gói tin quan sát được từ các nút con của nó, giải nén, tổng hợp với gói dữ liệu do chính nó quan sát được để sinh ra một gói tin duy nhất, nén dữ liệu rồi chuyển tiếp gói dữ liệu cho nút cha ở cấp cao hơn.



**Hình 5.1:** Ví dụ minh họa tổng hợp dữ liệu dựa trên cụm cây phân cấp

Bất cứ khi nào nút gốc của cây (CH) nhận được thông tin từ tất cả các nút thành viên trong cụm, nó sẽ chuyển tiếp đến BS sau khi thực hiện giải nén, tổng hợp và nén theo cách tương tự.

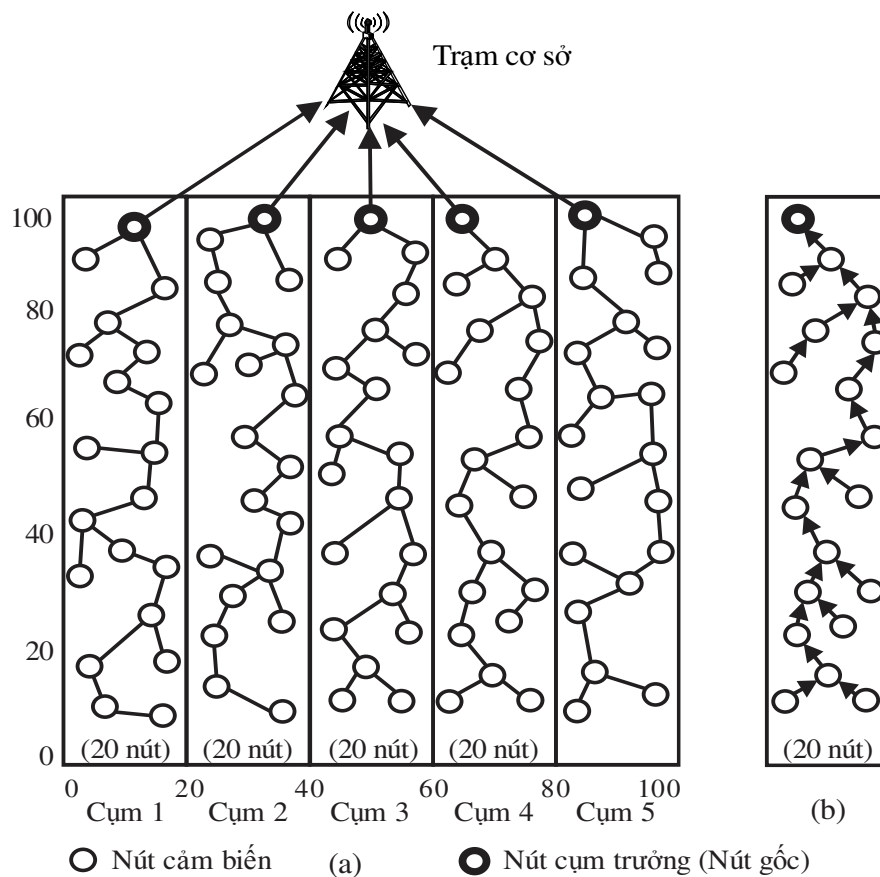
## 5.2. Đề xuất cải tiến thuật toán xây dựng cụm cây

### 5.2.1. Giai đoạn thiết lập cụm cây

Nội dung của giao thức DFTBC bao gồm bốn giai đoạn: (1) phân chia vùng và lựa chọn CH cho mỗi vùng, (2) xây dựng cụm dựa trên cây, (3) tổng hợp dữ liệu và (4) giai đoạn ổn định truyền dữ liệu.

#### 1. Phân chia cụm và chọn nút CH cho mỗi cây

Đầu tiên, BS sẽ lấy thông tin vị trí địa lý, số định danh và năng lượng còn lại của các nút còn sống trong mạng bằng cách trao đổi thông điệp giữa BS và các nút mạng. Sau đó, BS phân chia toàn bộ vùng cảm biến thành  $k$  cụm ( $k=5\%$  tổng số nút mạng) cân bằng nhau về số nút trong mỗi cụm như được minh họa trên Hình 5.2 [70, 118].



**Hình 5.2:** (a) Minh họa phân cụm mạng gồm 100-nút trong vùng 100m×100m, (b) một ví dụ về cây khung nhỏ nhất

Trong thí dụ minh họa này chúng tôi giả sử mạng gồm 100 nút cảm biến được triển khai trong vùng diện tích  $100\text{m} \times 100\text{m}$ .

Hình 5.2(a) minh họa tô-pô mạng hoạt động trong một vòng, các nút trong mỗi cụm được kết nối thành một cây tối thiểu và được truyền dữ liệu như Hình 5.2(b).

Trong mỗi vòng của DFTBC, BS sẽ chọn nút lãnh đạo (CH) cho mỗi cây trong mỗi cụm, chúng có năng lượng còn lại lớn hơn mức năng lượng trung bình  $E_{\text{avg}}$  được tính theo (4.20) và hàm  $\text{cost}()$  được tính theo (4.7) với các hằng số  $ec_1 (=100/J)$  và  $dc_2 (=5/m)$ .

---

#### **Thuật toán Phân chia cụm và chọn nút CH (Thuật toán 5.1)**

---

**Input:** Các nút mạng, biến  $k=5$ .

**Output:** Các biến  $Cluster1, \dots, Clusterk$  chứa  $N/k$  nút mạng và  $CH1, \dots, CHk$  chứa nút lãnh đạo cụm tương ứng.

//Các nút còn sống gửi thông tin về vị trí và năng lượng còn lại đến BS

1: **for each** nút  $i$  trong {Danh sách nút còn sống} **do**

2:     Gửi thông điệp HELLO có chứa năng lượng còn lại và vị trí địa lý tới BS

**end for**

//BS nhận thông tin từ các nút, phân chia cụm và chọn nút cụm trưởng như sau:

4: **if** (round là vòng đầu tiên) **then**

      Chia toàn bộ vùng cảm biến thành năm cụm với số nút cân bằng nhau như Hình 5.2

**end if**

7: **for each** mỗi cụm (mạng con) trong mạng **do**

8:     Tính mức năng lượng trung bình của mỗi cụm như trong (4.20)

9:     Chọn CH, nút có giá trị hàm  $\text{cost}$  là lớn nhất như trong (4.7)

10: **end for**

11: Chuyển sang **Thuật toán xây dựng cụm dựa trên cây (Thuật toán 5.2)**

---

## 2. Giai đoạn xây dựng cụm dựa trên cây

Trong phần này cây khung tối thiểu sẽ được xây dựng cho năm cụm dựa trên thuật toán Prim [27] để giải quyết vấn đề đồ thị vô hướng với giá được lựa chọn là khoảng cách giữa các nút để lấy về tổng khoảng cách giữa các nút trong cụm là nhỏ nhất từ đó tiết kiệm được nguồn năng lượng pin trong khi truyền. Ứng với mỗi cụm, đầu tiên ta sẽ kết nạp nút gốc (CH) vào cây, sau đó ta lần lượt kết nạp từng nút ngoài cây có khoảng cách nhỏ nhất từ nó đến một nút trên cây mà không tạo thành chu trình. Thuật toán sẽ kết thúc khi kết nạp được  $N/k$  nút vào cây.

---

### Thuật toán xây dựng cụm dựa trên cây (Thuật toán 5.2)

---

**Input:** Các nút mạng trong cụm (biến *Clusteri* chứa  $N/k$  nút)

**Output:** Cây *TREEi*.

```
1: label[CH]  $\leftarrow$  0; //bắt đầu từ nút CH (nút gốc) của cây
2: label[others node]  $\leftarrow$  MAX; //đặt nhãn khoảng cách cho các nút khác = MAX
3: TREE[CH]  $\leftarrow$  BS; // nút CH sẽ liên kết với BS để truyền dữ liệu
//Duyệt tất cả các nút còn sống để chọn nút gia nhập cây
4: while ({Danh sách nút còn sống trong cụm}  $\neq \emptyset$ ) do
5:     Tìm kiếm nút i trong {Danh sách nút còn sống} nút có label[i] là nhỏ nhất;
6:     endNode  $\leftarrow$  node i; //chọn nút i có khoảng cách từ i đến nút CH nhỏ nhất
//Tính lại các nhãn khoảng cách cho các nút khác lưu vào biến label
7:     for each nút j trong {Danh sách nút còn sống} do
8:         if label[j] >  $d(j, endNode)$  and (j  $\neq$  endNode) then
9:             label[j] =  $d(j, endNode)$ ; //lưu lại khoảng cách mới nhỏ hơn
10:            TREE[j] = endNode; //nút j sẽ truyền dữ liệu tới nút i trong cây
            end if
        end for
13:        Xóa nút endNode trong {Danh sách nút còn sống trong cụm } ;
    end while
```

---



---

14: Lập lịch TDMA, mã CDMA cho tất cả các nút trong cụm để truyền dữ liệu;  
15: Quảng bá lịch TDMA, mã CDMA và thông tin cây tới các nút còn sống trong mạng;  
16: **return**;

---

### 3. Giai đoạn tổng hợp dữ liệu

Sau khi BS chọn CH và xây dựng cây khung nhỏ nhất, lập lịch TDMA, mã CDMA và phân bố lịch TDMA tới các nút không phải CH, mã CDMA tới tất cả các nút CH trong mạng. Các nút cảm biến nhận thông tin cây, lịch TDMA mã CDMA và bắt đầu thu thập, tổng hợp và nén dữ liệu như Thuật toán 5.3. Trong Thuật toán 5.3, chúng tôi có sử dụng một số ký hiệu và kiến thức về mã nén phân tán, chúng được trình bày chi tiết trong Phụ lục 2 (Trang 144), chi tiết thuật toán gồm các bước như dưới đây:

---

#### Thuật toán Tổng hợp dữ liệu trên cây (Thuật toán 5.3)

---

##### **Ký hiệu:**

- *TREE*: thông tin về cây,
- *H* là ma trận kiểm tra chẵn lẻ nhị phân (xem Phụ lục 2, trường hợp 2)
- *Z* chứa  $2^3$  tập con được sử dụng để xác định dữ liệu ở bộ giải mã
- *X, Y, S, V, A, F* là các biến sử dụng trong chương trình
- $Z_s$  là phần tử có chỉ số nhị phân *s* trong tập *Z* (ví dụ,  $Z_{000} = (0000, 1111)$ : 000,  $Z_{001} = (0111, 1000)$ : 001), (xem Phụ lục 2, trường hợp 2, trang 146)

**Input:** Gói tin nén, thông tin cảm biến, *TREE*, *H*, *Z*.

**Output:** Gói tin đã tổng hợp và nén

// Giải nén sau khi nhận được gói dữ liệu từ các nút con trong cây

```
1: if (nút i trong {TREE} là nút cha) then  
2:   if (gói dữ liệu đã được nén) then  
3:     for each (10 bit trong gói dữ liệu được nén) do  
4:       Lấy Y  $\leftarrow$  7 bit cao như thông tin phụ
```

---

---

```

5:      Lấy  $s_X \leftarrow 3$  bit thấp còn lại
6:      Tính toán  $s_Y \leftarrow H * Y^T$ ;  $V \leftarrow [s_X \oplus s_Y]^T$ ;
7:      Tìm kiếm giá trị X trong tập  $Z_V$ , cái mà có  $Z_V \oplus Y = X$ ;
8:      Thêm Y và X vào gói dữ liệu giải nén;

      end for

10:  end if

//Tổng hợp dữ liệu từ các nút con với dữ liệu cảm biến của chính nó
11:  Biểu diễn dữ liệu cảm biến của n nút trong ma trận  $W = \{w\}_{n \times v}$ 
12:   $A \leftarrow$  Chuyển vị hàng 1 của W
13:  for each (i  $\leftarrow$  hàng từ 2 đến n trong W) do
//Nhân ma trận cột A với ma trận  $W_i$ , ta được ma trận U
14:      Tính  $U \leftarrow A * W_i$ ;
15:      for each (j  $\leftarrow$  hàng từ 1 đến v trong U) do
16:           $A_j \leftarrow$  Lấy các phần tử trên đường chéo chính của U, ( $u_{jj}$ );
17:          Tính  $F \leftarrow F + \sum_{p \neq q} u_{pq} \quad (p, q = 1, 2, 3, \dots, v)$ ;
18:      end for
// Tổng hợp dữ liệu giống với công thức (P1.3) và (P1.5) (xem Phụ lục 1).
19:      for each (j  $\leftarrow$  từ 1 đến m) do
20:           $M_j = A_j / (1 - F)$ ;
21:      end if
// Nén dữ liệu không mất sử dụng DSC
22:  if gói dữ liệu cần được nén trong bộ mã hóa then
23:      for each (X = 14 bit trong gói dữ liệu) do
24:          Đặt Y  $\leftarrow$  7 bit cao từ X như thông tin phụ
25:          Tính toán  $s_X \leftarrow H * X^T$ 
26:          Thêm Y và  $s_X$  vào trong gói dữ liệu mới;
      end fo
28:  end if

```

---

Hoạt động trong Thuật toán 5.3 được chia thành ba công việc chính đó là: (1) giải nén gói dữ liệu nhận được từ các nút con; (2) tổng hợp dữ liệu nhận được từ các nút con thành một gói tin duy nhất và (3) nén gói tin sử dụng DSC. Công việc (1) và (2) được thực hiện bởi các nút cha hoặc CH trong khi công việc (3) được thực hiện bởi tất cả các nút.

Các câu lệnh từ dòng 3 đến dòng 10 sẽ giải nén gói tin để nhận được thông tin gốc bằng cách đọc một khối 10 bit, lấy 7 bit đầu đặt vào biến Y như "thông tin phụ", sau đó gán 3 bit còn lại vào biến  $s_X$ , dòng lệnh 6 tính  $\mathbf{H} * \mathbf{Y}^T$  đặt vào  $s_Y$  và tính  $[s_X \oplus s_Y]^T$  đặt vào biến V (ký hiệu  $\mathbf{Y}^T$  là ma trận chuyển vị của ma trận Y). Dòng lệnh 7 tính  $Z_V \oplus Y$  đặt vào X và ghi Y, X vào gói dữ liệu giải nén ( $Z_V$  là phần tử có chỉ số V trong tập Z).

Các dòng lệnh tổng hợp dữ liệu từ dòng 11 đến dòng 21 được thực hiện giống với Thuật toán 4.3 từ dòng 10 đến dòng 19.

Các dòng lệnh từ dòng 22 đến 28 thực hiện nén dữ liệu theo DSC, dòng 23 đọc mỗi khối dữ liệu 14 bit sau đó tách ra thành 2 dãy, mỗi dãy 7 bit đặt ở các biến X và Y. Theo DSC, chúng ta không gửi cả 14 bit này mà chỉ gửi Y như "thông tin phụ" và gửi  $s_X = 3$  bit nhờ sử dụng phép nhân  $\mathbf{H} * \mathbf{X}^T$  ( $s_X$  chính là chỉ số của tập con  $Z_{s_X}$ , xem Phụ lục 2, trường hợp 2). Câu lệnh dòng 26 sẽ ghi Y và  $s_X$  vào gói tin nén ( $Y + s_X = 10$  bit thay vì  $Y + X = 14$  bit, nhờ đó giảm được 4 bit).

### 5.2.2. Giai đoạn truyền dữ liệu

Sau khi hoàn thành chia vùng, xây dựng cụm dựa trên cây, các gói dữ liệu bắt đầu được truyền từ các nút lá. Đầu tiên, các nút lá từ vị trí xa nhất trong mỗi cây sẽ bắt đầu truyền dữ liệu đến nút cha của nó trong khe TDMA. Các nút cha sẽ nhận dữ liệu, giải nén và tổng hợp với gói dữ liệu mà nó quan sát được, nén rồi chuyển tiếp đến nút cha ở mức cao hơn như được minh họa trên Hình 5.2(b). Bất cứ khi nào nút gốc (CH) nhận được tất cả dữ liệu từ các nút con của nó, nó sẽ chuyển tiếp đến BS theo mã CDMA sau khi tổng hợp và nén theo cách tương tự. Sau một khoảng thời gian, vòng mới sẽ được bắt đầu bằng việc chọn lại nút gốc cũng như xây dựng lại cây khung tối thiểu mới cho mỗi nhóm.

### 5.3. Kết hợp với lập lịch ngủ

#### 5.3.1. Đặt vấn đề

Đề xuất thứ hai của chúng tôi trong lược đồ định tuyến theo mô hình cây khung nhỏ nhất là lược đồ SSTBC. Trong lược đồ này, toàn bộ vùng cảm biến mạng được chia ra thành các cụm, mỗi cụm được chia thành các ô lưới ảo nhỏ, kích thước lưới ảo nhỏ hơn hoặc bằng một giá trị ngưỡng (Threshold), chúng được thiết lập bởi người dùng. Để tiết kiệm năng lượng, SSTBC chọn những nút có mức năng lượng còn lại lớn nhất trong số các nút thuộc cùng một ô lưới ảo ở chế độ thức, các nút khác được lập lịch tắt các bộ thu phát sóng vô tuyến không cần thiết của chúng (vào chế độ ngủ) vì các nút ở trong cùng một ô lưới ảo gần nhau dù nó cùng thức thì cũng chỉ quan sát được các thông tin gần như giống nhau. Hơn nữa, SSTBC còn giảm năng lượng tiêu thụ bằng cách giảm khoảng cách truyền đa chặng của các nút trong cụm dựa trên việc xây dựng cây khung nhỏ nhất.

#### 5.3.2. Phân tích tiêu thụ năng lượng trong giao thức SSTBC

Đặt  $\psi$  biểu thị khoảng thời gian của giai đoạn thiết lập "set-up", và  $t_{dt}$  (data transmission) là khoảng thời gian của giai đoạn truyền dữ liệu, do vậy, khoảng thời gian cho mỗi vòng là  $t_{round} = (\psi + t_{dt})$ . Giả sử chúng ta coi khoảng thời gian từ khi mạng bắt đầu hoạt động cho đến khi nút đầu tiên trong mạng chết là thời gian sống của mạng cảm biến và ký hiệu là  $t_{fnd}$  (dying time of the first node). Do đó, chúng ta có quan hệ giữa thời gian sống của mạng  $t_{fnd}$  và  $t$  như được hiển thị trong Hình 5.3 dưới đây [62]:

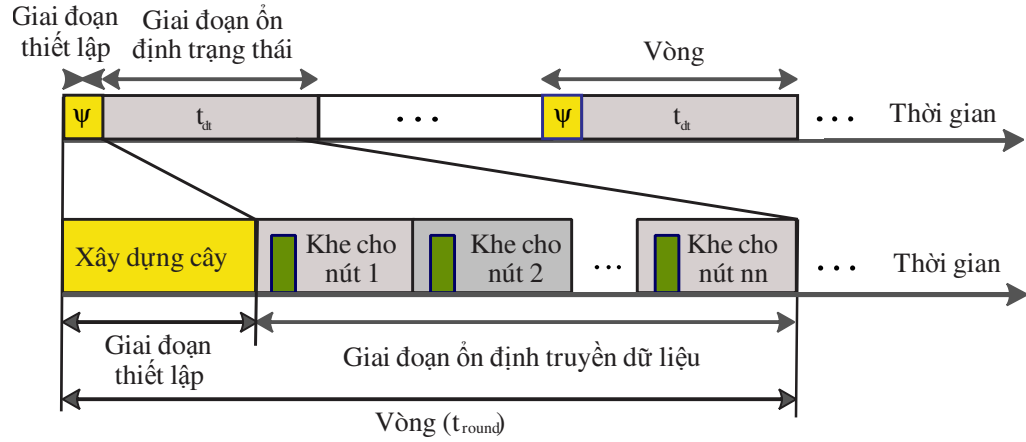
$$t_{fnd} = N_R * t_{round} = u * (\psi + t_{dt}) \quad (5.1)$$

trong đó,  $N_R$  là số vòng mà mạng đã hoạt động cho đến khi nút đầu tiên chết. Theo giao thức SSTBC, có  $m$  gói dữ liệu được truyền trong khoảng thời gian  $t_{dt}$ , với:

$$t_{round} = (\psi + m.T_{frame}) \quad (5.2)$$

ở đây,  $T_{frame}$  là thời gian truyền của mỗi gói. Do đó, ta có thời gian sống của mạng là:

$$t_{fnd} = N_R(\psi + m.T_{frame}) \quad (5.3)$$



**Hình 5.3:** Hoạt động hai giai đoạn trong một vòng của SSTBC

### Năng lượng tiêu thụ của một nút CH và một nút thành viên trong pha khởi tạo

Giả sử có  $N$  nút cảm biến, chúng gồm  $k$  cụm (cây) được phân bố đều trong vùng có kích thước  $M \times M$ , tọa độ của BS là  $(X_{BS}, Y_{BS})$ , và năng lượng ban đầu của nút là  $E_{init}$ . Do đó, có trung bình  $(N/k - N_S)$  nút cảm biến trong cụm ở chế độ hoạt động. Nó chứa một nút CH và  $(N/k - N_S - 1)$  nút thành viên, ở đây,  $N_S$  là số nút thành viên trong cụm đang ở chế độ ngủ. Không mất tính tổng quát, chúng tôi giả sử rằng tất cả các nút cảm biến đều ở chế độ thức (hoạt động,  $N_S = 0$ ).

Ký hiệu  $E_{brc}$ : Năng lượng tiêu thụ của một nút hoạt động, nó quảng bá mã định danh, năng lượng còn lại và vị trí địa lý đến BS. Do đó:

$$E_{brc}(q, d_{ToBS}) = qE_{elec} + qE_{tworay} d_{ToBS}^4 \quad (5.4)$$

$E_{rm}$ : Năng lượng tiêu thụ của một nút hoạt động, nhận thông điệp từ BS, chúng chứa đựng thông tin về cụm dựa trên cây, CH và lịch TDMA.

$$E_{rm}(q) = qE_{elec} \quad (5.5)$$

### Năng lượng tiêu thụ trong giai đoạn truyền dữ liệu

Giai đoạn ổn định truyền dữ liệu là một phần của khoảng thời gian trong mỗi vòng, trên Hình 5.3, đó là khoảng thời gian được ký hiệu là  $t_{round}$ . Đặt  $E_{CH}$  là năng lượng tiêu thụ của một nút CH cho việc truyền một gói tin, năng lượng này bao gồm năng lượng dùng cho việc nhận các gói dữ liệu từ các nút con của nó, năng lượng dùng

cho việc tổng hợp và chuyển tiếp một gói dữ liệu tới BS và có thể được tính theo công thức sau:

$$E_{CH} = chqE_{elec} + (ch + 1)qE_{DF} + qE_{elec} + qE_{tworay} d_{ToBS}^4 \quad (5.6)$$

trong đó,  $ch$  là số nút con kết nối với nút CH, giả sử mỗi nút trên cây chỉ kết nối với một nút con ( $ch=1$ ) thì (5.6) trở thành:

$$E_{CH} = 2qE_{elec} + 2qE_{DF} + qE_{tworay} d_{ToBS}^4 \quad (5.7)$$

$E_{non-CH}$  là năng lượng tiêu thụ của các nút thành viên (không phải CH) cho việc nhận, tổng hợp và truyền gói dữ liệu trong một cụm (cây) được tính tương tự như biểu thức (4.12) ở Chương 4.

Do đó, tổng năng lượng tiêu thụ trong mỗi cụm (cây) (ký hiệu  $E_{cluster}$ ) cho việc truyền một gói tin đến BS là [22, 39]:

$$E_{Cluster} = E_{CH} + \left(\frac{N}{k} - 1\right)E_{non-CH} \approx E_{CH} + \frac{N}{k}E_{non-CH} \quad (5.8)$$

Và tổng năng lượng tiêu thụ của toàn bộ mạng (ký hiệu  $E_{Total}$ ) để mỗi nút CH truyền một gói tin đến BS là [10]:

$$E_{Total} = kE_{Cluster} \quad (5.9)$$

Nếu các nút cảm biến được triển khai đều trong vùng diện tích  $M \times M$ , chúng ta có thể đưa ra tọa độ của nút cảm biến  $p$  theo chỉ số hàng ( $i$ ) và chỉ số cột ( $j$ ), như dưới đây [62]:

$$p_{i,j} = \left( \frac{M}{\sqrt{N}}(0.5 + j), \frac{M}{\sqrt{N}}(0.5 + i) \right) \quad (5.10)$$

Do đó, khoảng cách của nút  $p_{i,j}$  tới nút hàng xóm gần nhất (ví dụ nút  $p_{i+1,j}$ ) là:

$$d_{ToNB} = \frac{M}{\sqrt{N}} \quad (5.11)$$

Từ đó ta có, năng lượng tiêu thụ của mỗi cụm trong một vòng là:

$$E_R = \frac{N}{k}E_{brc} + \frac{N}{k}E_{rm} + mE_{CH} + m\left(\frac{N}{k} - 1\right)E_{non-CH} \quad (5.12)$$

Và số vòng mà toàn mạng hoạt động, trước khi có nút đầu tiên chết (hết năng lượng), có thể được tính như sau:

$$N_R = \frac{\sum_{i=1}^N E_{\text{init}}(i)}{k.E_R} \quad (5.13)$$

Bây giờ, chúng ta có thể tính giá trị tối ưu của số lượng cây (cụm) dựa vào phép tính cực trị của hàm  $E_{\text{Total}}$  theo  $k$  như sau [44, 113]:

$$\frac{dE_{\text{Total}}}{dk} = 0, \quad \text{mà} \quad E_{\text{Total}} = k \left( E_{\text{CH}} + \left( \frac{N}{k} - 1 \right) E_{\text{non-CH}} \right) \quad (5.14)$$

Thật vậy, thay (5.7), (4.12) và (5.8) vào (5.14), ta có:

$$E_{\text{Total}} = k \left( 2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{tworay}} d_{\text{ToBS}}^4 \right) + N \left[ \left( \frac{N}{k} - 1 \right) \left( qE_{\text{elec}} + qE_{\text{DF}} + qE_{\text{elec}} + qE_{\text{friis}} d_{\text{toNB}}^2 \right) - qE_{\text{elec}} \right] \quad (5.15)$$

$$= k \left( 2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{tworay}} d_{\text{ToBS}}^4 \right) + N \left[ \frac{N}{k} \left( 2qE_{\text{elec}} + qE_{\text{DF}} + qE_{\text{friis}} d_{\text{toNB}}^2 \right) \right] \quad (5.16)$$

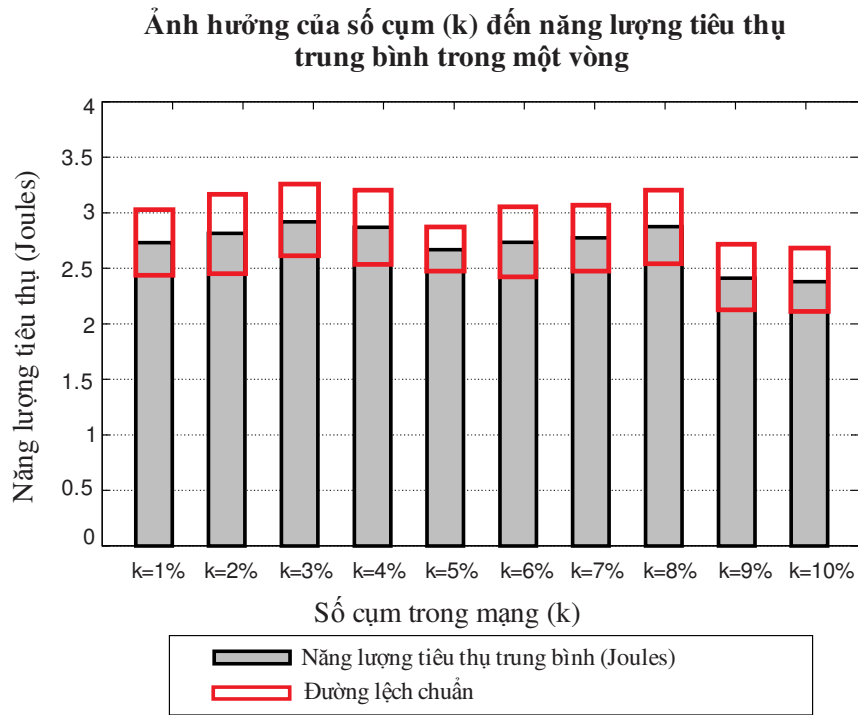
Từ đó, ta có:

$$\frac{dE_{\text{Total}}}{dk} = 2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{tworay}} d_{\text{ToBS}}^4 - \frac{N^2}{k^2} \left( 2qE_{\text{elec}} + qE_{\text{DF}} + qE_{\text{friis}} d_{\text{toNB}}^2 \right) = 0 \quad (5.17)$$

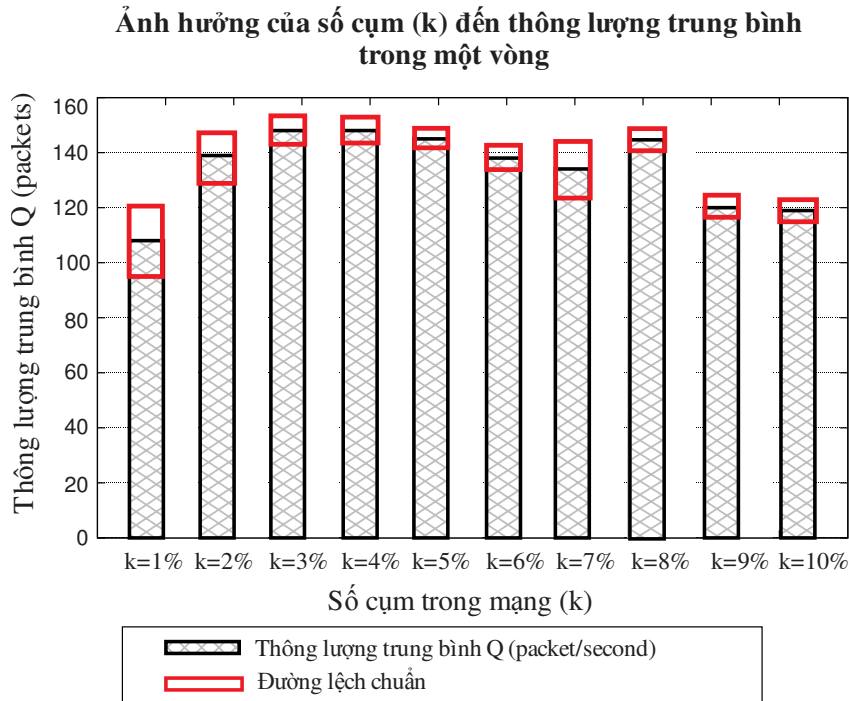
$$k^2 = \frac{N^2 \left( 2qE_{\text{elec}} + qE_{\text{DF}} + qE_{\text{friis}} d_{\text{toNB}}^2 \right)}{2qE_{\text{elec}} + 2qE_{\text{DF}} + qE_{\text{tworay}} d_{\text{ToBS}}^4} \quad (5.18)$$

$$\text{và cuối cùng, ta có: } k_{\text{Opt}} = \sqrt{\frac{2N^2 E_{\text{elec}} + N^2 E_{\text{DF}} + E_{\text{friis}} N M^2}{2E_{\text{elec}} + 2E_{\text{DF}} + E_{\text{tworay}} d_{\text{toBS}}^4}} \quad (5.19)$$

Để xác định số cụm tối ưu cần chia trong mạng khi áp dụng giao thức SSTBC, chúng tôi đã sử dụng 100 kịch bản mô phỏng đã được sử dụng ở Chương 3, Mục 3.3 và Bảng 3.1 gồm 100 nút cảm biến được phân bố ngẫu nhiên trong vùng 100m×100m, vị trí BS ở (49, 175). Toàn bộ vùng cảm biến mạng được chia thành  $k$  cụm khác nhau với  $k=1\%$ ,  $2\%$ , ... và  $10\%$  tổng số nút mạng. Sau khi thực hiện các bước chạy mô phỏng (từ Bước 2 đến Bước 7, Chương 3, Mục 3.3), giá trị trung bình tổng năng lượng mạng tiêu thụ, thông lượng trung bình của toàn mạng trong một vòng của 21 kịch bản được chúng tôi vẽ đồ thị trên Hình 5.4 và Hình 5.5 dưới đây:



**Hình 5.4:** Năng lượng tiêu thụ trung bình trong mỗi vòng với số cụm khác nhau



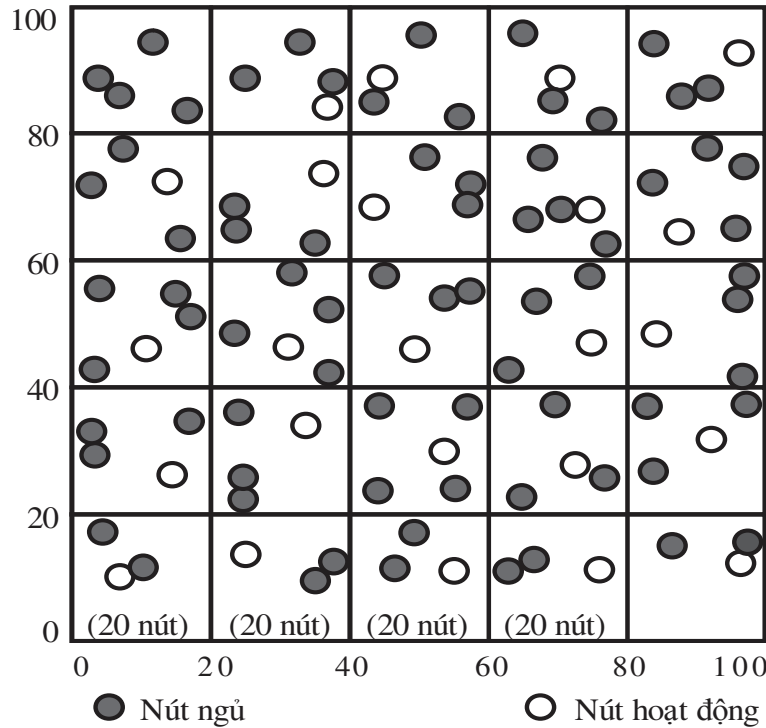
**Hình 5.5:** Thông lượng trung bình Q nhận được ở BS trong mỗi vòng với số cụm khác nhau



Hình 5.4 và 5.5 hiển thị năng lượng tiêu thụ trung bình và thông lượng trung bình Q nhận được ở BS trên một vòng với phần trăm cụm khác nhau được chia trong mạng. Trục tung của đồ thị Hình 5.4 hiển thị giá trị năng lượng tiêu thụ trung bình của mạng (Joules) và trục tung của đồ thị Hình 5.5 hiển thị thông lượng trung bình Q (gói tin/giây) của mạng trong một vòng. Rõ ràng, giao thức SSTBC là tốt nhất khi phần trăm cụm phân chia trong mạng là  $k_{Opt}=5\%$ . Giá trị này là cũng tương đương với kết quả nghiên cứu của các công trình [44] và [113].

### 5.3.3. Thuật toán lập lịch ngủ

**Bước 1: Lập lịch ngủ:** Trong hầu hết các trường hợp, các nút cảm biến không dây thường được triển khai dày đặc (có thể lên đến 300 nút trên vùng  $25 \text{ m}^2$  [35]) bao phủ toàn bộ vùng cảm biến và liên tục gửi thông tin thu thập được tương tự nhau về BS [70, 100]. Các nút cảm biến có thông tin giống nhau nếu gửi sẽ là trùng lặp, dư thừa, gây lãng phí nhiều năng lượng vì hoạt động gửi dữ liệu là một trong các hoạt động tiêu hao năng lượng nhiều nhất trong nút cảm biến [2, 9, 21]. Do đó, để hạn chế việc gửi thông tin dư thừa trong mạng, chúng ta có thể giới hạn trong một vùng nhỏ chỉ cho một nút hoạt động còn các nút khác vào chế độ ngủ để tiết kiệm năng lượng mà vẫn đảm bảo thông tin về đối tượng cảm biến được liên tục truyền về BS. Cụ thể là, chúng ta sẽ chia toàn bộ vùng cảm biến thành 5 cụm khác nhau, mỗi cụm lại chia nhỏ thành các ô lưới ảo, chúng có kích thước nhỏ hơn hoặc bằng một giá trị gọi là giá trị ngưỡng. Giá trị ngưỡng càng lớn thì càng có ít nút ở chế độ hoạt động, mạng tiêu thụ năng lượng càng ít nhưng chất lượng thông tin cảm biến sẽ không tốt. Do đó, tùy từng ứng dụng mạng mà chúng ta có thể chọn giá trị ngưỡng sao cho phù hợp. Trong Hình 5.6, chúng tôi minh họa vùng cảm biến mạng được chia thành các ô lưới ảo nhỏ với kích thước nhỏ hơn hoặc bằng kích thước ngưỡng ( $20\text{m} \times 20\text{m}$ ). Sau đó, trong mỗi ô thuộc một ô lưới ảo chúng ta sẽ chọn một nút có mức năng lượng còn lại lớn nhất trong tập nút cùng thuộc một ô lưới ảo ở chế độ hoạt động còn các nút khác sẽ đi vào chế độ ngủ để tiết kiệm năng lượng. Sau mỗi vòng hoạt động, nút ở chế độ hoạt động trong ô sẽ được chọn lại để đảm bảo cân bằng tiêu thụ năng lượng.



**Hình 5.6:** Ví dụ về kích thước của các ô lưới ảo nhỏ hơn hoặc bằng  $20m \times 20m$

---

#### Thuật toán Lập lịch ngủ (Thuật toán 5.4)

---

**Input:** Các nút mạng, biến  $k=5$ , biến khoảng cách ngưỡng  $Threshold=5$ .

**Output:** Các biến  $Cluster1$ ,  $Cluster2$ ,  $Cluster3$ ,  $Cluster4$ ,  $Cluster5$  chứa các nút mạng ở chế độ hoạt động.

1: **for each** {Tất cả các nút còn sống trong mạng} **do**

2:     Gửi thông điệp HELLO, chúng chứa ID, năng lượng còn lại và vị trí địa lý của nút, đến BS

**end for**

4: Chia vùng cảm biến mạng thành các ô lưới ảo kích thước nhỏ hơn khoảng cách ngưỡng  $Threshold$  như trong Hình 5.6.

//Từ vòng thứ hai trở đi, thuật toán không chia lại vùng mà chỉ đổi vai trò nút ở chế độ hoạt động

5: **for each** {các nút cảm biến trong ô} **do**

6:     Chọn nút trong chế độ hoạt động, nút có mức năng lượng còn lại lớn nhất

---

---

7: Thiết lập các nút còn lại trong cùng lưới vào chế độ ngủ

8: **end for**

9: Chuyển sang **Thuật toán Chọn nút CH (Thuật toán 5.5)**

---

## **Bước 2: Chia vùng và chọn nút CH**

BS sẽ chia vùng cảm biến thành  $k$  cụm khác nhau ( $k=5$ ), chúng bằng nhau về kích thước mạng con như được minh họa trên Hình 5.7, nơi mà toàn bộ vùng mạng được đề xuất gồm 100 nút cảm biến triển khai trong vùng  $100m \times 100m$ . Trong mạng này, các nút cảm biến ở chế độ hoạt động trong mỗi cụm được tổ chức thành một cây khung tối thiểu với nút gốc là CH.

Trong mỗi vòng của lược đồ SSTBC, BS sẽ chọn nút CH cho mỗi cụm, chúng có năng lượng còn lại lớn hơn mức năng lượng trung bình của các nút ở chế độ hoạt động và có giá trị hàm  $cost()$  lớn nhất như (4.20) và (4.7) với các hằng số  $ec_1 (=100/J)$  và  $dc_2 (=5/m)$ .

---

### **Thuật toán Chọn nút CH (Thuật toán 5.5)**

---

**Input:** Các nút mạng trong biến  $Cluster1, Cluster2, \dots, Clusterk$ , biến  $k=5$ .

**Output:** Nút lãnh đạo trong biến  $CH1, CH2, \dots, CHk$  tương ứng.

1: **if** (số vòng =1) **then**

Chia vùng cảm biến mạng thành  $k$  cụm như trong Hình 5.7.

**end if**

4: **for each** (cụm trong mạng) **do**

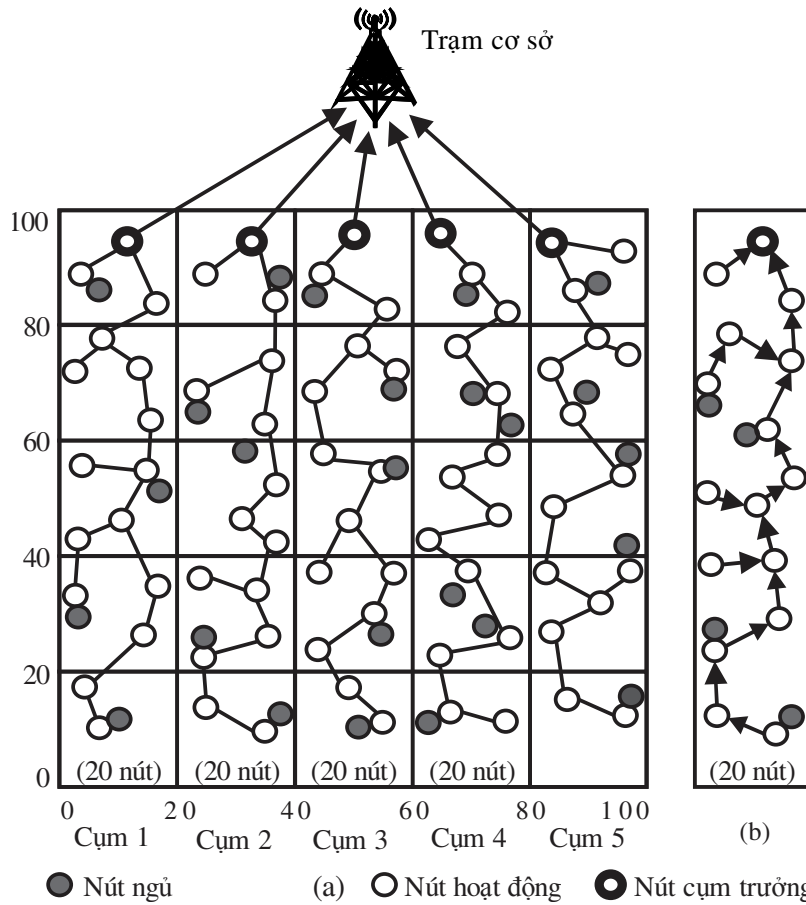
5: Tính mức năng lượng trung bình như trong (4.20)

6: Chọn nút CH, chúng có giá trị hàm  $cost()$  lớn nhất như (4.7)

7: **end for**

8: Chuyển sang **Thuật toán Xây dựng cụm dựa trên cây (Thuật toán 5.2)**

---



**Hình 5.7:** (a) Minh họa phân cụm mạng gồm 100 nút trong vùng 100m×100m với kích thước lưới ô ảo 5m×5m, (b) một ví dụ về cây khung tối thiểu

Về độ phức tạp tính toán của DFTBC và SSTBC, các Thuật toán 5.1, 5.2, 5.4 và 5.5 mỗi nút chỉ thực hiện truyền và nhận một thông điệp đến BS, do đó, độ phức tạp thông báo ở mỗi nút cảm biến là hằng số. Các thao tác tính toán được thực hiện bởi BS, Thuật toán 5.1 thực hiện công việc tính mức năng lượng trung bình, hàm  $\text{cost}()$  theo (4.7) để chọn nút CH có độ phức tạp tính toán là  $O(n)$ . Thuật toán 5.2 xây dựng cụm cây sử dụng thuật toán Prim có độ phức tạp tính toán là  $O[(n/k)\log(n/k)]$ . Do đó SSTBC có độ phức tạp tính toán là  $O(n\log(n))$  và độ phức tạp thông báo là hằng số. Thuật toán 5.3 tổng hợp dữ liệu từ ch nút cảm biến (nút con trên cây), mỗi nút quan sát môi trường đích với  $v$  giá trị, có độ phức tạp là  $O(ch \times v^2)$ . Do đó, DFTBC có độ phức tạp tính toán là  $O(ch \times v^2)$  và độ phức tạp thông báo là hằng số.

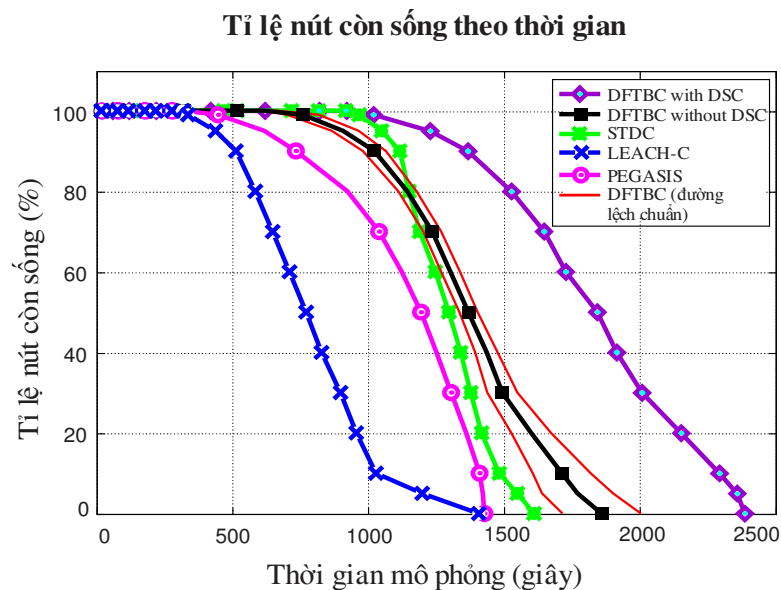
## 5.4. Phân tích, đánh giá và so sánh thời gian sống của mạng khi áp dụng các giao thức PEGASIS, STDC, DFTBC và SSTBC

### 5.4.1. Các tham số mô phỏng

Chúng tôi đã cài đặt và mô phỏng giao thức DFTBC và SSTBC trên bộ công cụ mã nguồn mở NS2 để phân tích, đánh giá hiệu năng và thời gian sống của các nút mạng cảm biến dựa trên 100 kịch bản đã được sinh ra ở Chương 3, Mục 3.3. Thêm nữa, chúng tôi cũng so sánh giao thức được chúng tôi đề xuất, cải tiến với các giao thức đã có như LEACH-C [75], PEGASIS [64] và STDC [55]. Các tham số mô phỏng được chúng tôi thiết lập sử dụng trong các kịch bản mô phỏng với các tham số như Bảng 3.1, chúng cũng được sử dụng trong các nghiên cứu [6, 76, 78, 110].

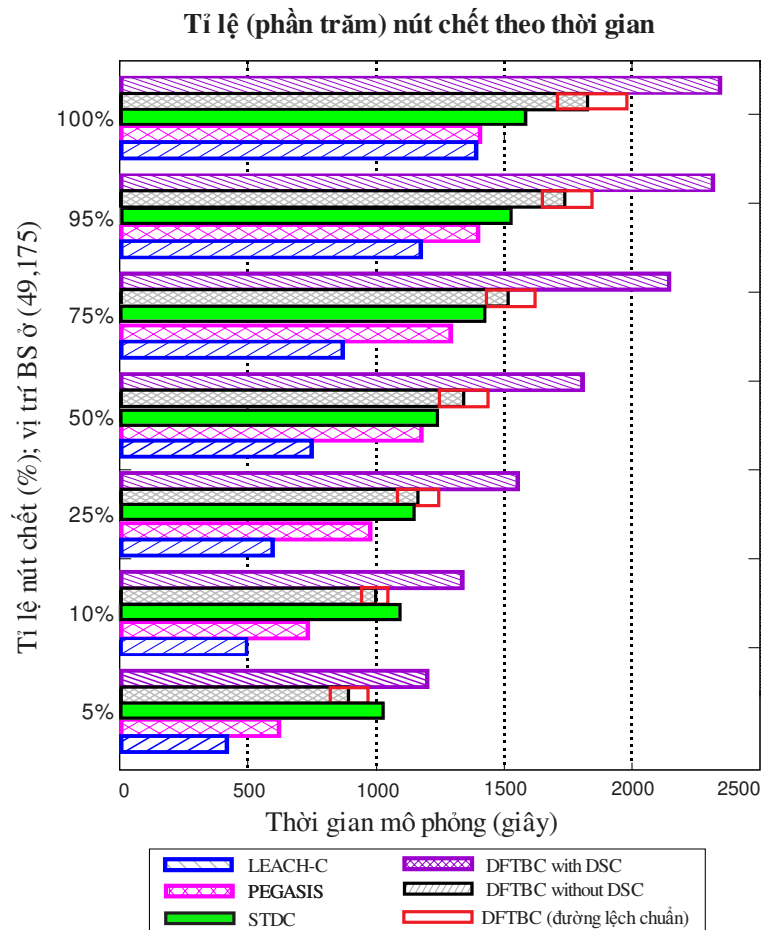
### 5.4.2. Kết quả mô phỏng DFTBC

Sau khi cài đặt, chạy mô phỏng theo các bước từ Bước 2 đến Bước 7 như ở Chương 3, Mục 3.3, chúng tôi chọn tỉ lệ độ lệch chuẩn  $\xi$  tính theo các công thức (3.11), (3.12) và (3.13), với giao thức DFTBC bằng 6,2% số kịch bản đã chạy  $n_{sc}=19$  kịch bản, với giao thức STDC bằng 9,3% số kịch bản  $n_{sc}=24$  thì dừng. Giá trị trung bình nhận được sau khi chạy mô phỏng được chúng tôi xử lý và trình bày bằng đồ thị, trên các hình từ Hình 5.8 đến Hình 5.11 dưới đây:



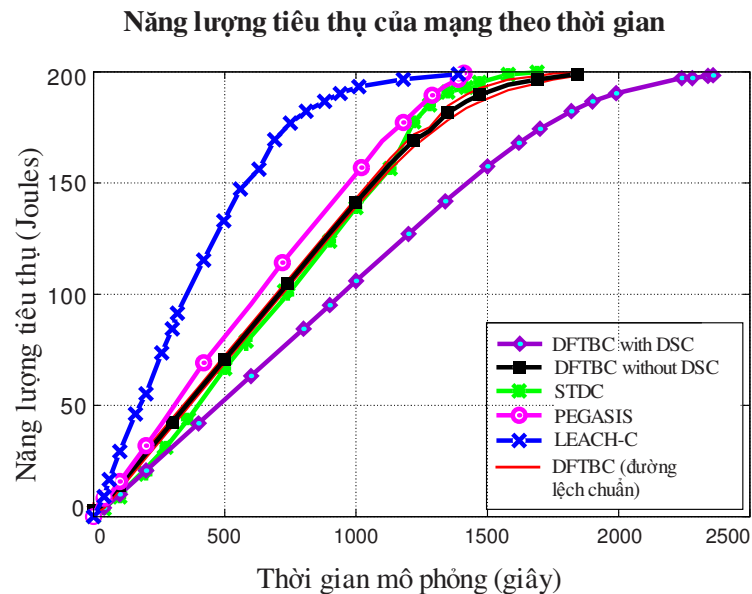
**Hình 5.8:** Tỉ lệ nút mạng còn sống trong suốt thời gian mô phỏng

Hình 5.8 trình bày trung bình các kết quả mô phỏng của  $n_{sc}$  ( $n_{sc}=24$ ,  $n_{sc}=19$ ) kịch bản khác nhau khi áp dụng các giao thức LEACH-C, PEGASIS, STDC và DFTBC với độ đo hiệu năng là tỉ lệ nút mạng còn sống theo thời gian. Trên hình 5.8, trục Y biểu diễn tỉ lệ nút mạng còn sống, trục X biểu diễn thời gian sống của các nút biểu thị theo giây. Đồ thị biểu diễn kết quả cho thấy rõ ràng giao thức DFTBC, nhờ kết hợp linh hoạt giữa chọn nút CH xem xét đến năng lượng còn lại và khoảng cách từ nút ứng viên đến BS và giảm khoảng cách truyền thông trong cụm nhờ xây dựng cây tối thiểu, làm thời gian sống của mạng tăng thêm khoảng 60%, 20% khi so sánh với giao thức LEACH-C, PEGASIS và DFTBC có tỉ lệ thời gian ở 50% nút mạng chết, 100% nút mạng chết dài hơn khoảng 8% khi so sánh với giao thức STDC trong trường hợp có tổng hợp, không nén dữ liệu.



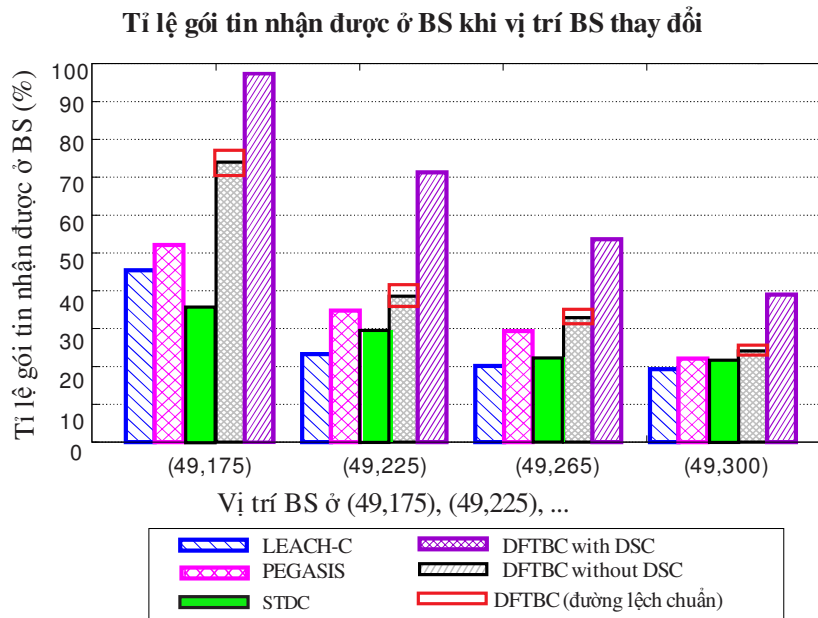
**Hình 5.9:** Tỉ lệ (phần trăm) nút chết khi vị trí BS ở (49,175)

Như có thể thấy trên Hình 5.9, một cách biểu diễn ngược lại với Hình 5.8, tỉ lệ các nút mạng chết khi áp dụng các giao thức khác nhau được phân tích trong suốt thời gian mô phỏng tương ứng với 5%, 10%, 25%, 50%, 75%, 95%, và 100% nút mạng chết. Chúng ta có thể thấy giao thức xây dựng cụm cây mở rộng STDC cho hiệu quả sử dụng năng lượng tốt hơn DFTBC ở khoảng 20% nút đầu tiên chết nhưng DFTBC với tiêu thụ năng lượng ít đạt được tỉ lệ từ 10% đến 20% nút chết thấp hơn các giao thức STDC và PEGASIS theo thời gian mô phỏng [75, 78].



**Hình 5.10:** Năng lượng tiêu thụ của mạng khi áp dụng các giao thức

Hình 5.10 cho thấy giao thức DFTBC với thuật toán xây dựng cụm cây dựa trên thuật toán cây khung với khoảng cách nhỏ nhất, liên kết các nút với nhau thành tuyến truyền dữ liệu, do đó, giảm năng lượng tiêu thụ cho các nút. Các đường đồ thị trong Hình 5.10 nằm càng thấp hơn (gần trục X hơn), năng lượng tiêu thụ càng ít, hiệu quả sử dụng năng lượng càng cao. Do đó, DFTBC giúp đạt được hiệu quả sử dụng năng lượng tốt và nâng cao thời gian sống cho mạng cảm biến khi so sánh với LEACH-C và PEGASIS. Giao thức STDC cho hiệu quả sử dụng năng lượng tốt ở khoảng 1000 giây đầu tiên, sau đó năng lượng tiêu thụ hết nhanh hơn DFTBC.



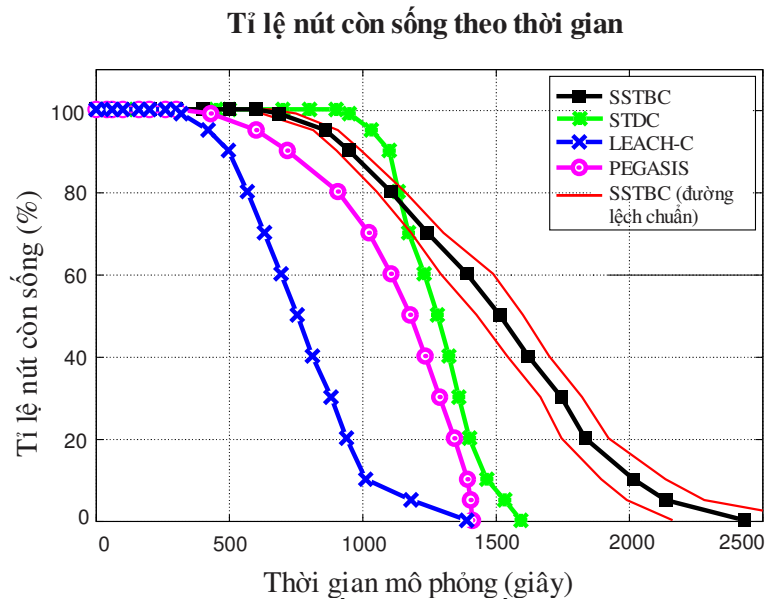
**Hình 5.11:** Tỉ lệ gói tin nhận được ở BS khi vị trí thay đổi

Hình 5.11 biểu diễn bằng đồ thị tỉ lệ gói tin nhận được ở BS khi vị trí địa lý của BS được thay đổi từ vị trí gần nhất (49,175) tới vị trí xa nhất (49, 300) so với vùng mạng cảm biến. Rõ ràng, chúng có mối quan hệ giảm khi di chuyển vị trí địa lý của BS ra xa hơn vì các nút mạng phải truyền dữ liệu xa, tiêu tốn nhiều năng lượng cho bộ khuếch đại trong mô hình truyền thông không dây. Tuy nhiên, tỷ lệ gói tin nhận được ở BS đối với DFTBC vẫn tốt hơn LEACH-C, PEGASIS và STDC. Giao thức STDC tiêu thụ nhiều năng lượng cho việc phân cụm, xây dựng cây mở rộng phân tán (sử dụng các gói tin chào hỏi, truyền quảng bá, ...) do đó tỉ lệ gói tin nhận được ở BS thấp hơn các giao thức khác. Hơn nữa, các Hình 5.8, 5.9, 5.10, và 5.11 cũng cho thấy hiệu quả tiêu thụ năng lượng của giao thức được chúng tôi đề xuất khi áp dụng phương pháp nén phân tán DSC để cải tiến tốt hơn, làm tăng thêm khoảng 30% thời gian sống cho mạng khi so sánh với cùng giao thức DFTBC mà không nén. Các đường màu đỏ biểu diễn độ lệch chuẩn của giá trị trung bình theo giao thức DFTBC có tỉ lệ độ lệch chuẩn  $\xi$  bằng 6,2% tương ứng với số kịch bản  $n_{sc}=19$ .



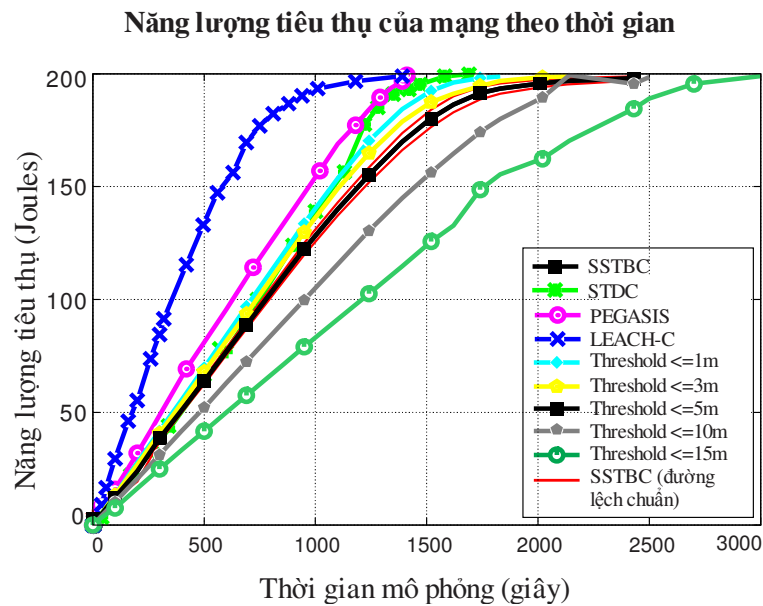
### 5.4.3. Kết quả mô phỏng giao thức SSTBC

SSTBC đã được chúng tôi cài đặt và tiến hành mô phỏng với công cụ mô phỏng mạng mã nguồn mở NS2. Các kịch bản mô phỏng và các tham số mô phỏng là giống với các kịch bản được sử dụng ở Chương 3, Mục 3 và Bảng 3.1. Hiệu năng của SSTBC được chúng tôi đánh giá bằng các độ đo: (1) Tỷ lệ (phần trăm) nút còn sống trong suốt thời gian mô phỏng; (2) năng lượng tiêu thụ theo vòng và (3) tỉ lệ gói tin nhận được ở BS khi vị trí BS thay đổi. Các độ đo này có tác động nhiều nhất đến hiệu quả sử dụng năng lượng của các giao thức. Các số liệu thu được sau khi chạy mô phỏng theo các bước từ Bước 2 đến Bước 7 như ở Chương 3, Mục 3.3, chúng tôi chọn tỉ lệ độ lệch chuẩn tính theo các công thức (3.11), (3.12) và (3.13), với giao thức STDC bằng 9,3% số kịch bản đã chạy  $n_{sc}=24$  kịch bản thì dừng và với giao thức SSTBC bằng 5,8% số kịch bản đã chạy  $n_{sc}=16$  kịch bản thì dừng. Giao thức STDC có tỉ lệ độ lệch chuẩn cao là do việc chọn CH theo xác suất và việc xây dựng cụm phân tán trong khi đó SSTBC xây dựng cụm tập trung. Giá trị trung bình nhận được sau khi chạy 16 kịch bản mô phỏng được chúng tôi xử lý, kết xuất và biểu diễn trên các đồ thị dạng đường và dạng cột khác nhau, trên các Hình 5.12, 5.13 và 5.14.



**Hình 5.12:** Tỉ lệ nút còn sống trong suốt thời gian mô phỏng

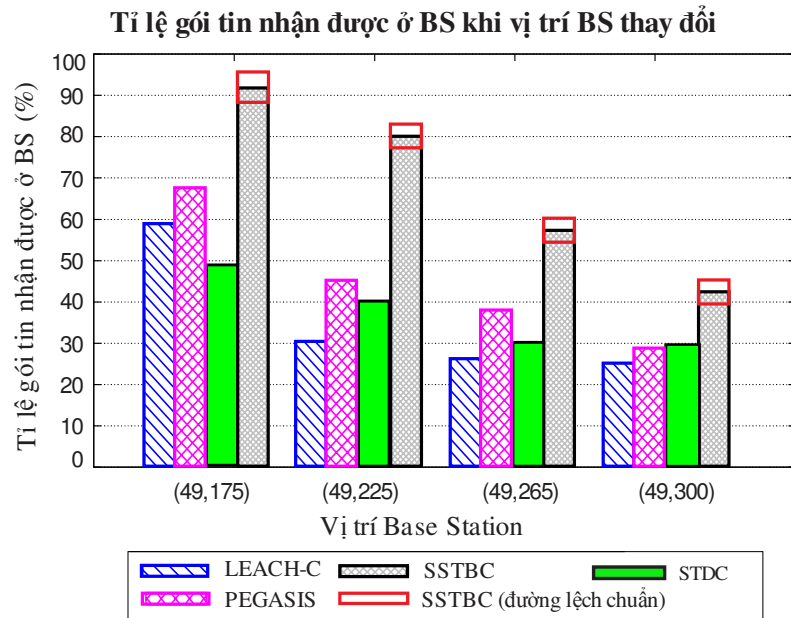
Hình 5.12 cho thấy giá trị trung bình của 16 kịch bản mô phỏng khác nhau áp dụng giao thức SSTBC với tỉ lệ nút còn sống trong mạng trong suốt thời gian mô phỏng (tính theo giây). Rõ ràng, giao thức SSTBC với việc phân chia vùng cảm biến thành lưới ô vuông ảo và hạn chế nút ở chế độ hoạt động, có thời gian sống dài hơn LEACH-C, PEGASIS khoảng 65%, 25% và có tỉ lệ nút mạng còn sống từ 1% đến 75% nút cao hơn giao thức STDC khoảng 12% theo thời gian mô phỏng [75, 78].



**Hình 5.13:** Trung bình năng lượng tiêu thụ áp dụng bởi ba giao thức

Hình 5.13 cho thấy giao thức SSTBC với việc kết hợp giữa chọn nút CH hợp lý và thuật toán xây dựng cụm cây dựa trên thuật toán cây khung với khoảng cách nhỏ nhất, liên kết các nút với nhau thành tuyến truyền dữ liệu, do đó, giảm năng lượng tiêu thụ cho các nút. Các đường đồ thị trong Hình 5.13 nằm càng thấp hơn (gần trục X hơn), năng lượng tiêu thụ càng ít, hiệu quả sử dụng năng lượng càng cao. Do đó, SSTBC giúp đạt được hiệu quả sử dụng năng lượng tốt hơn khoảng 65% và 20% khi so sánh với LEACH-C và PEGASIS. Giao thức STDC cho hiệu quả sử dụng năng lượng tốt ở khoảng thời gian 900 giây đầu tiên, sau đó năng lượng tiêu thụ hết nhanh hơn SSTBC.

Hình 5.13 cũng hiển thị năng lượng tiêu thụ của SSTBC với giá trị ngưỡng (kích thước các ô lưới ảo) khác nhau được chia trong mạng. Quan sát thấy nếu kích thước các ô lưới ảo càng lớn, có càng nhiều nút vào chế độ ngủ thì mạng tiêu thụ năng lượng càng ít.



**Hình 5. 14:** Tỉ lệ (phần trăm) gói tin nhận được ở BS khi vị trí BS thay đổi

Như minh họa trong Hình 5.14, trung bình tỉ lệ gói tin nhận được ở BS sau khi chạy 16 kịch bản mô phỏng khác nhau với vị trí địa lý của BS được thay đổi từ vị trí (49,175) tới vị trí xa nhất (49, 300) cách xa vùng mạng cảm biến khác nhau. Nó dường như có mối quan hệ giảm khi tỉ lệ thông điệp nhận được ở BS với cả ba giao thức khi vị trí BS di chuyển ra xa dần khu vực mạng cảm biến vì khi đó các nút mạng phải chuyển tiếp các gói tin đi xa hơn. Tuy nhiên, giao thức SSTBC với việc hạn chế số nút hoạt động và giảm khoảng cách truyền thông vẫn tốt hơn các giao thức LEACH-C, PEGASIS và STDC đã có.

### 5.5. Phân tích và so sánh với các thuật toán cùng hướng khác

Các thuật toán phân cụm dựa trên cây trong [41, 54, 55, 91] có các nút gốc của cây tiêu tốn nhiều năng lượng trong giai đoạn truyền dữ liệu đến BS vì việc chọn nút

CH như nút gốc vẫn dựa vào xác suất ngẫu nhiên, không xem xét đến khoảng cách từ nó đến BS. Hai đề xuất của chúng tôi DFTBC và SSTBC chọn nút ứng viên CH như nút gốc cây có mức năng lượng còn lại lớn hơn mức năng lượng còn lại trung bình của các nút còn sống trong cụm và xem xét kết hợp giữa năng lượng còn lại của nút ứng viên và khoảng cách từ nó đến BS. Hơn nữa, đề xuất DFTBC còn kết hợp giữa định tuyến và tổng hợp, nén dữ liệu sử dụng mã nén phân tán để giảm số bit dữ liệu cần truyền từ đó nâng cao hiệu quả sử dụng năng lượng.

## **5.6. Tổng kết chương**

Trong chương này, chúng tôi đã trình bày đề xuất giao thức DFTBC, một cải tiến của lược đồ định tuyến phân cụm dựa trên việc xây dựng cây khung tối thiểu để tạo liên kết truyền dữ liệu giữa các nút trong mỗi cụm. Hơn nữa, DFTBC còn kết hợp giữa định tuyến và tổng hợp, nén dữ liệu trên cây để giảm chi phí truyền thông trong mạng, nâng cao hiệu quả sử dụng năng lượng. Để đánh giá hiệu năng của giao thức do chúng tôi đề xuất, chúng tôi đã cài đặt, mô phỏng trên NS2 để phân tích, đánh giá hiệu năng của giao thức DFTBC. Thông qua các kết quả mô phỏng được trình bày trên các đồ thị cho thấy DFTBC kéo dài thời gian sống của mạng tăng thêm khoảng 60%, 20% và 8% khi so sánh với giao thức LEACH-C, PEGASIS và STDC. Hơn nữa, giao thức DFTBC cũng cho thấy hiệu quả của việc nén dữ liệu bằng DSC để giảm năng lượng tiêu thụ, mở rộng thời gian sống của các nút mạng thêm khoảng 30% khi so sánh với cùng giao thức DFTBC mà không nén. Lược đồ định tuyến phân cụm dựa trên cây kết hợp với lập lịch ngủ cũng được chúng tôi trình bày trong chương này. Thông qua các Hình 5.12, 5.13 và 5.14 cho thấy lược đồ SSTBC với việc hạn chế số nút hoạt động ở mỗi ô lưới ảo cho hiệu quả sử dụng năng lượng cao hơn LEACH-C, PEGASIS và STDC khoảng 65%, 25% và 12%. Các kết quả nghiên cứu trong Chương 5 tương ứng với công trình 4 và 5 đã công bố.

## Chương 6. KẾT LUẬN

Chương này, chúng tôi tổng kết các kết quả đạt được của luận án, các đóng góp chính của luận án và giới thiệu một số hướng nghiên cứu mở rộng tiếp theo. Đầu tiên, chúng tôi hệ thống hóa các vấn đề cơ bản và chuyên sâu về các giao thức định tuyến tiết kiệm năng lượng trong mạng cảm biến không dây. (Chủ yếu được trình bày trong Chương 2).

Đề xuất giao thức LEACH-DE, một cải tiến của giao thức LEACH, nâng cao hiệu quả sử dụng năng lượng bằng cách chọn nút CH và xây dựng cụm phân tán có xem xét đến khoảng cách và năng lượng còn lại trung bình của các nút. Đề xuất này giúp giảm rõ rệt tỉ lệ nút mạng bị chết do hết năng lượng sớm so với LEACH và LEACH-C. Nội dung chi tiết của đề xuất được trình bày trong Chương 3.

Đề xuất cải tiến một thuật toán xây dựng chuỗi dài, có kết hợp với tổng hợp và nén dữ liệu trong chuỗi DFCB (Data Fusion and Chain-Based Clustering for Energy-Efficient in Wireless Sensor Networks). DFCB giúp nâng cao hiệu quả sử dụng năng lượng khoảng 40% so với LEACH và khoảng 10% so với PEGASIS. (trình bày trong Chương 4). Chương 4 cũng trình bày đề xuất SCBC (Sector-Chain Based Clustering for Energy Efficiency In Heterogeneous Wireless Sensor Network) - Một lược đồ định tuyến phân cụm (cung) dựa trên chuỗi, thực hiện việc cân bằng số nút cảm biến trong mỗi cụm và tối ưu thời gian hoạt động của các cụm trong giai đoạn truyền dữ liệu. Đề xuất này giúp nâng cao hiệu quả sử dụng năng lượng khoảng 20% so với PEGASIS và khoảng 15% so với IEEPB.

Đề xuất DFTBC (Data Fusion and Tree-Based Clustering Routing Protocol) - Thực hiện cải tiến giao thức định tuyến phân cụm tiết kiệm năng lượng dựa trên cây khung tối thiểu, cụ thể là áp dụng thuật toán Prim để kết nối các nút cảm biến trong cụm với nhau thành cây với nút CH chính là nút gốc, nhờ đó giảm được khoảng cách truyền thông tổng thể trong cụm, nâng cao hiệu quả sử dụng năng lượng. Đề xuất này giúp nâng cao hiệu quả về thời gian sống của mạng khoảng 60% so với LEACH-C, khoảng 20% so với PEGASIS và khoảng 8% so với STDC. (trình bày

trong Chương 5). Đề xuất SSTBC (Sleep Scheduled and Tree-Based Clustering), cải tiến một thuật toán phân cụm dựa trên cây kết hợp với lập lịch ngủ đã có [70], bằng cách chia ảo toàn bộ vùng cảm biến thành các ô, trong mỗi ô chỉ duy trì một nút mạng “thức” còn các nút khác vào chế độ “ngủ” để tiết kiệm năng lượng. Các kết quả mô phỏng của chúng tôi cho thấy SSTBC giúp kéo dài thời gian sống của mạng khoảng 65% so với LEACH-C, khoảng 20% so với PEGASIS và khoảng 12% so với STDC (Chương 5).

Việc áp dụng các đề xuất cải tiến vào thực tế là tùy vào từng ứng dụng cụ thể, tùy vào mật độ nút cảm biến được triển khai, khoảng cách truyền giữa các nút hoặc tần suất lấy mẫu mà các đề xuất sẽ cho hiệu quả năng lượng khác nhau. Ví dụ với những ứng dụng có mật độ nút dày đặc, rõ ràng đề xuất cải tiến SSTBC sẽ cho hiệu quả năng lượng tốt nhờ vào kỹ thuật lập lịch ngủ cho nhiều nút trong cùng một ô lưới trong khi các đề xuất khác sẽ cho nhiều dữ liệu trùng lặp hơn nếu các nút cảm biến nhiệt độ hay độ ẩm, v.v. Ngoài ra, vị trí của BS cũng có ảnh hưởng tương đối đến hiệu quả năng lượng của WSN mà kỹ thuật định tuyến phân cụm mang lại. Nếu BS ở gần mạng WSN, khi đó kỹ thuật phân cụm tập trung rõ ràng cho hiệu quả tiêu thụ năng lượng tốt hơn kỹ thuật phân cụm phân tán vì trong giai đoạn thiết lập cụm, BS đã thực hiện tính toán phân cụm, chọn CH, v.v. Do đó, các nút mạng cảm biến không phải tiêu hao năng lượng cho việc truyền/nhận các gói tin chào hỏi, quảng bá CH, ...đến mạng. Trong khi đó, với các đề xuất cải tiến có tổng hợp và có nén dữ liệu sẽ cho hiệu quả năng lượng tốt hơn hẳn trong cả kỹ thuật định tuyến tập trung và phân tán.

Trên cơ sở các vấn đề chưa giải quyết và các hướng nghiên cứu mở rộng với bài toán định tuyến tiết kiệm năng lượng trong mạng cảm biến nói riêng và mạng không dây nói chung, chúng tôi sẽ tiếp tục tìm hiểu nghiên cứu sâu hơn theo các định hướng sau: Nghiên cứu giao thức định tuyến tiết kiệm năng lượng trong mạng cảm biến không dây khi triển khai mạng trên mặt biển và dưới nước biển. Thêm nữa, nghiên cứu giao thức định tuyến tiết kiệm năng lượng trong mạng MANET hoặc trong mạng cảm biến không dây thông minh (Cognitive Radio Sensor Network).

Bên cạnh các kết quả đạt được, luận án chắc chắn khó tránh khỏi những thiếu sót, hạn chế đó là: (1) luận án trình bày bằng tiếng Việt, nếu có thể trình bày bằng tiếng

Anh thì sẽ tốt hơn; (2) danh mục công trình nghiên cứu của tác giả chưa phong phú, chưa có nhiều công bố ở các tạp chí nổi tiếng thế giới.

Nghiên cứu sinh rất mong muốn nhận được thêm nhiều ý kiến đóng góp hữu ích của các thầy, cô và bạn đọc.

## DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. **Nguyen Duy Tan**, Ho Duc Ai, Duong Viet Huy and Nguyen Dinh Viet (2013), "An Improved LEACH Routing Protocol for Energy-Efficiency of Wireless Sensor Networks", *Proceeding of Fundamental and Applied Information Technology Research (FAIR)*, pp. 33-40.
2. **Nguyen Duy Tan** and Nguyen Dinh Viet (2014), "DFCB: Data Fusion and Chain-Based Clustering Routing Protocol for Energy-Efficient in WSNs", *Proceeding of Fundamental and Applied Information Technology Research (FAIR)*, pp. 102-111.
3. Duong Viet Huy, **Nguyen Duy Tan**, Ho Duc Ai, and Nguyen Dinh Viet (2014), "Tiếp cận phương pháp tổng hợp dữ liệu nhiều cảm biến trong mạng cảm biến không dây bằng lý thuyết tập thô", *Proceeding of Fundamental and Applied Information Technology Research (FAIR)*, pp. 668-677.
4. **Nguyen Duy Tan** and Nguyen Dinh Viet (2014), "DFTBC: Data Fusion and Tree-Based Clustering Routing Protocol for Energy-Efficient in Wireless Sensor Networks", *Knowledge and Systems Engineering (KSE)*, pp. 61-77.
5. **Nguyen Duy Tan** and Nguyen Dinh Viet (2015), "SSTBC: Sleep Scheduled and Tree-Based Clustering Routing Protocol for Energy-Efficient in Wireless Sensor Networks", *International Conference on Computing and Communication Technologies (IEEE-RIVF)*, pp. 180-185.
6. **Nguyen Duy Tan** and Nguyen Dinh Viet (2015), "SCBC: Sector-Chain Based Clustering Routing Protocol for Energy Efficiency in Heterogeneous Wireless Sensor Network", *Advanced Technologies for Communications (ATC)*, pp. 314-319.
7. **Nguyen Duy Tan** and Nguyen Dinh Viet (2015), "CEER-Channel and Energy-Efficient Routing Protocol Based on Interference Avoidance for Cognitive Radio Ad Hoc Networks", *Proceeding of Fundamental and Applied Information Technology Research (FAIR)*, pp. 61-71.

Danh mục này bao gồm 07 công trình.



## TÀI LIỆU THAM KHẢO

- [1]. Abdelgawad, A. (2014). "Distributed Data Fusion Algorithm for Wireless Sensor Network". In *IEEE 11th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 334-337.
- [2]. Abdelgawad, A., & Bayoumi, M. (2012). "Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks". *Louisiana, USA: Lecture Notes in Electrical Engineering*.
- [3]. Ahn, K. S., Kim, D. G., Sim, B. S., Youn, H. Y., & Song, O. (2011). "Balanced Chain-Based Routing Protocol (BCBRP) for Energy Efficient Wireless Sensor Networks". In *Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW)*, pp. 227-231.
- [4]. Akyildiz, I. F., & Vuran, M. C. (2010). "*Wireless Sensor Networks*". USA: John Wiley & Sons.
- [5]. Alawi, R. A. (2011). "RSSI Based Location Estimation in Wireless Sensors Networks". *Proceedings of the 17th IEEE International Conference on Networks (ICON)*, pp. 118-122.
- [6]. Ali, M. S., Dey, T., & Biswas, R. (2008). "ALEACH Advanced LEACH Routing Protocol for Wireless Microsensor Networks". *International Conference on Electrical and Computer Engineering (ICECE)*, pp. 909-914.
- [7]. Aliouat, Z., & Aliouat, M. (2012). "Effective Energy Management in Routing Protocol for Wireless Sensor Networks". In *5th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-5.
- [8]. Aliouat, Z., & Harous, S. (2012). "An Efficient Clustering Protocol Increasing Wireless Sensor Networks Life Time". *International Conference on Innovations in Information Technology (IIT)*, pp. 194-199.
- [9]. Almasri, M. M., & Elleithy, K. M. (2014). "Data Fusion Models in WSNs: Comparison and Analysis". *Conference of the American Society for Engineering Education (ASEE Zone 1)*, pp. 1-6.
- [10]. Alqamzi, H., & Li, T. J. (2010). "A Distributed Topology Management Protocol for Wireless Sensor Networks". *IEEE International Conference on Communications (ICC)*, pp. 1-5.
- [11]. Alshudukhi, J., Ou, S., & Ball, P. (2014). "A Ground Level Radio Propagation Model for Road-based Wireless Sensor Networks". In *the 9th International*

*Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 146-151.

- [12]. Arif, M., Brouard, T., & Vincent, N. (2006). "A fusion methodology based on Dempster-Shafer evidence theory for two biometric applications". *The 18th International Conference on Pattern Recognition (ICPR)*, pp. 590-593.
- [13]. Aslam, M., Shah, T., Javaid, N., Rahim, A., Rahman, Z., & Khan, Z. A. (2012). "CEEC: Centralized Energy Efficient Clustering A New Routing Protocol for WSNs". *In the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 103-105.
- [14]. Banerjee, J., Naskar, M. K., Biswas, U., Alfandi, O., & Hogrefe, D. (2014). "Leader Selection in Wireless Sensor Networks - An Energy Efficient Approach". *International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, pp. 508-512.
- [15]. Barad, J., & Kadhiwala, B. (2014). "DIST-LEACH - A deterministic key management scheme for securing cluster-based sensor networks". *Proceedings of the International Conference on Advances in Engineering and Technology Research (ICAETR)*, pp. 1-5.
- [16]. Barceló-Lladó, J., Morell, A., & Granados, S. G. (2010). "Distributed Source Coding in Large Wireless Sensor Networks". *In Forty-Fourth Asilomar Conference on Signals, Systems & Computers*, pp. 1535-1539.
- [17]. Basir, O., Karray, F., & Zhu, H. (2005). "Connectionist-Based Dempster-Shafer Evidential Reasoning for Data Fusion". *IEEE Transactions on Neural Networks*, 16 (6), 1513-1530.
- [18]. Brar, G. S., Rani, S., Chopra, V., Malhotra, R., Song, H., & Ahmed, S. H. (2016). "Energy Efficient Direction-Based PDORP Routing Protocol for WSN". *IEEE Access Journal*, 4, 3182-3194.
- [19]. Butenko, M. V., Nazarenko, A., Sarian, V., Sushchenko, N., & Lutokhin, A. (2014). "Applications of Wireless Sensor Networks in Next Generation Networks". *International Telecommunication Union*.
- [20]. Challa, S., & Koks, D. (2004). "Bayesian and Dempster-Shafer fusion". *SADHANA-Academy Proceedings in Engineering Sciences*, 29 (2), 145-174.
- [21]. Chang, B., & Zhang, X. (2010). "An Energy-Efficient Cluster-based Data Gathering Protocol for Wireless Sensor Networks". *In 6th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-5.

- [22]. Chen, H., Zhang, C., Zong, X., & Wang, C. (2013). "LEACH-G: an Optimal Cluster-heads Selection Algorithm based on LEACH". *Journal of Software*, 8 (10), 2660-2667.
- [23]. Chen, K.-H., Huang, J.-M., & Hsiao, C.-C. (2009). "CHIRON: An Energy-Efficient Chain-Based Hierarchical Routing Protocol in Wireless Sensor Networks". *Wireless Telecommunications Symposium (WTS)*, pp. 1-5.
- [24]. Chen, W.-Q., & Hu, B.-J. (2010). "A Data Fusion Algorithm in Leach Protocol Using Gauss Membership Function". In *6th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4.
- [25]. Cheng, C.-T., Tse, C. K., & Lau, F. C. (2011). "A Clustering Algorithm for Wireless Sensor Networks Based on Social Insect Colonies". *IEEE Sensors Journal*, 11 (3), 711-721.
- [26]. Corke, P., Wark, T., Jurdak, R., Hu, W., Valencia, P., & Moore, D. (2010). "Environmental Wireless Sensor Networks". *Proceedings of the IEEE*, 98 (11), 1903-1917.
- [27]. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). "Introduction to Algorithms". *Cambridge, Massachusetts London, England: The MIT Press, Third Edition*.
- [28]. Dam, T. V., & Langendoen, K. (2003). "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks". In *The First ACM Conference on Embedded Networked Sensor Systems (Sensys'03)*, pp. 171-180.
- [29]. Deosarkar, B. P., Yadav, N. S., & R.P.Yadav. (2010). "Distributed Clustering with Restricted Number of Clusterheads for Energy Efficient Data Gathering in Wireless Sensor Networks". *International Journal of Engineering and Technology*, 2 (1), 7-16.
- [30]. Fontani, M., Bianchi, T., Rosa, A. D., Piva, A., & Barni, M. (2011). "A Dempster-Shafer Framework for Decision Fusion in Image Forensics". *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1-6.
- [31]. G.Rajesh, B.Vinayagasundaram, & Moorthy, G. (2014). "Data Fusion in Wireless Sensor Network Using Simpson's 3/8 Rule". *International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 1-5.

- [32]. Gautam, N., Lee, W.-I., & Pyun, J.-Y. (2009). "Track-Sector Clustering for Energy Efficient Routing in Wireless Sensor Networks". In *Ninth IEEE International Conference on Computer and Information Technology*, pp. 116-121.
- [33]. Gehrig, N., & Dragotti, P. L. (2005). "Symmetric and Asymmetric Slepian-Wolf codes with systematic and non-systematic linear codes". *IEEE Communications Letters*, 9 (1), 61-63.
- [34]. Ghneimat, A., Mellor, J., & Jiang, P. (2011). "Adaptive, Cluster Based, Sensor Network Routing Protocol". *Proceedings UKSim 13th International Conference on Modelling and Simulation*, pp. 472-476.
- [35]. Glisic, S. G (2016). "Advanced Wireless Networks: Technology and Business Models". USA: John Wiley & Sons.
- [36]. Gou, H., Yo o, Y., & Zeng, H. (2009). "A Partition-Based LEACH Algorithm for Wireless Sensor Networks". In *Ninth IEEE International Conference on Computer and Information Technology*, pp. 40-45. Xiamen, China.
- [37]. Grover, J., Shikha, & Sharma, M. (2014). "Location Based Protocols in Wireless Sensor Network - A Review". *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-5.
- [38]. Guo, W., Zhang, W., & Lu, G. (2010). "PEGASIS Protocol in Wireless Sensor Network Based on an Improved Ant Colony Algorithm". *Proceedings the Second International Workshop on Education Technology and Computer Science (ETCS)*, pp. 64-67.
- [39]. Guo, Z., Dou, J., Wang, P., & Zhang, G. (2008). "Energy-efficient Uniform Clustering Algorithm for Wireless Sensor Networks". *International Conference on High Performance Switching and Routing (HSPR)*, pp. 257-262.
- [40]. Han, L. (2010). "LEACH-HPR: An Energy Efficient Routing Algorithm for Heterogeneous WSN". *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, pp. 507-511.
- [41]. Han, Z., Wu, J., Zhang, J., Liu, L., & Tian, K. (2014). "A General Self-Organized Tree-Based Energy-Balance Routing Protocol for Wireless Sensor Network". *IEEE Transactions on Nuclear Science*, 61 (2), 732-740.
- [42]. Harichandan, P., Jaiswal, A., & Kumar, S. (2013). "Multiple Aggregator Multiple Chain Routing Protocol for Heterogeneous Wireless Sensor Networks". *International Conference on Signal Processing and Communication (ICSC)*, pp. 127-121.

- [43]. Hea, T., Stankovica, J. A., Lub, C., & Abdelzaher, T. (2003). "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks". *Proceedings 23rd International Conference on Distributed Computing Systems*, pp. 46-55.
- [44]. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). "An Application-Specific Protocol Architecture for Wireless Microsensor Networks". *IEEE Transactions on wireless communication*, 1 (4), 660-670.
- [45]. Hempstead, M., Tripathi, N., Mauro, P., Wei, G.-Y., & Brooks, D. (2005). "An Ultra Low Power System Architecture for Sensor Network Applications". In *Proceedings of the 32nd International Symposium on Computer Architecture (ISCA'05)*, pp. 208-219.
- [46]. Huang, K.-C., Yen, Y.-S., & Chao, H.-C. (2007). "Tree-Clustered Data Gathering Protocol (TCDGP) for Wireless Sensor Networks". *Future Generation Communication and Networking (FGCN)*, pp. 31-36.
- [47]. Issariyakul, T., & Hossain, E. (2009). *"Introduction to Network Simulator NS2"*. NewYork, USA: Springer US.
- [48]. Javaid, N., Aslam, M., K.Djouani, Khan, Z. A., & Alghamdi, T. A. (2014). "ATCEEC: A New Energy Efficient Routing Protocol for Wireless Sensor Network". *IEEE International Conference on Communications (ICC)*, pp. 263-268.
- [49]. Jun, L., Hua, Q., & Yan, L. (2012). "A Modified LEACH algorithm In Wireless Sensor Network Based On NS2". *International Conference on Computer Science and Information Processing (CSIP)*, pp. 604-606.
- [50]. Jung, S.-M., Han, Y.-J., & Chung, T.-M. (2007). "The Concentric Clustering Scheme for Efficient Energy Consumption in the PEGASIS". *The 9th International Conference on Advanced Communication Technology*, pp. 260-265.
- [51]. Kaur, M., Jain, A., & Goel, A. K. (2014). "Energy Efficient Two Level Distributed Clustering Scheme to Prolong Stability Period of Wireless Sensor Network". *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 68-73.
- [52]. Khamiss. A. A., Senchun, C., Baihai, Z., Jingye, L., & Qiao, L. (2014). "An Energy-Efficient and Density-Aware Clustering for WSNs". *Proceedings of the 33rd Chinese Control Conference (CCC)*, pp. 377-382.
- [53]. Khamiss, A., Senchun, C., Baihai, Z., & Lingguo, C. (2015). "Combined metrics-clustering algorithm based on LEACH-C". *The 27th Chinese Control and Decision Conference (CCDC)*, pp. 5252-5257.

- [54]. Kim, K. T., Lyu, C. H., Moon, S. S., & Youn, H. Y. (2010). "Tree-Based Clustering (TBC) for Energy Efficient Wireless Sensor Networks". *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 680-685.
- [55]. Kim, K. T., Park, J. C., Kim, M., Kim, U. M., & Youn, H. Y. (2014). "A Spanning Tree based Data Collection for Real-Time Streaming Sensor Data". *IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, pp. 202-207.
- [56]. Kordafshari, M. S., Pourkabirian, A., Faez, K., & Rahimabadi, A. M. (2009). "Energy-Efficient SPEED Routing Protocol for Wireless Sensor Networks". *Fifth Advanced International Conference on Telecommunications AICT*, pp. 267-271.
- [57]. Kumar, N., Sandeep, Bhutani, P., & Mishra, P. (2012). "U-LEACH: A Novel Routing Protocol for Heterogeneous Wireless Sensor Networks". *International Conference on Communication, Information and Computing Technology (ICCICT)*, pp. 1-4.
- [58]. Kumar, P., & Gupta, V. (2011). "Data Compression using Distributed Source Coding in Wireless Sensor Network". *International Journal of Electronics Communication and Computer Technology*, 2 (1), 10-14.
- [59]. Law, Y. W., Palaniswami, M., Hoesel, L. V., Doumen, J., Hartel, P., & Havinga, P. (2009). "Energy-Efficient Link-Layer Jamming Attacks against Wireless Sensor Network MAC Protocols". *ACM Transactions on Sensor Networks*, 5 (1), 1-38.
- [60]. Li, J., Luo, S., & Jin, J. S. (2010). "Sensor Data Fusion for Accurate Cloud Presence Prediction Using Dempster-Shafer Evidence Theory". *Sensors Open Access Journal*, 10 (10), 9384-9396.
- [61]. Li, X., Shi, H., & Shang, Y. (2005). "A Sorted RSSI Quantization Based Algorithm for Sensor Network Localization". *Proceedings 11th International Conference on Parallel and Distributed Systems*, pp. 557-563.
- [62]. Li, Y., Yu, N., Zhang, W., Zhao, W., You, X., & Daneshmand, M. (2011). "Enhancing the Performance of LEACH Protocol in Wireless Sensor Networks". *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 223-228.
- [63]. Lin, Y.-C., & Zhong, J.-H. (2012). "Hilbert-Chain Topology for Energy Conservation in Large-Scale Wireless Sensor Networks". *The 9th International*

*Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UICATC)*, pp. 225-232.

- [64]. Lindsey, S., & Raghavendra, C. S. (2002). "PEGASIS: Power-Efficient GATHERing in Sensor Information System". *IEEE Aerospace Conference Proceedings*, pp. 1125-1130.
- [65]. Lindsey, S., Raghavendra, C., & Sivalingam, K. M. (2002). "Data Gathering Algorithms in Sensor Networks Using Energy Metrics". *IEEE Transactions on Parallel and Distributed Systems*, 13 (9), 924-935.
- [66]. Liu, A., Zheng, Z., Zhang, C., Chen, Z., & Shen, X. (2012). "Secure and Energy-Efficient Disjoint Multi-Path Routing for WSNs". *IEEE Transactions on Vehicular Technology*, 61 (7), 3255-3265.
- [67]. Liu, M., Cao, J., Chen, G., & Wang, X. (2009). "An Energy-Aware Routing Protocol in Wireless Sensor Networks". *Sensors Open Access Journal*, 9 (1), 445-462.
- [68]. Liu, S., & Cheng, L. (2006). "Efficient Data Compression in Wireless Sensor Networks for Civil Infrastructure Health Monitoring". *The 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, pp. 823-829.
- [69]. Liu, X. (2012). "A Survey on Clustering Routing Protocols in Wireless Sensor Networks". *Sensors Open Access Journal*, 11113-11153.
- [70]. Lohan, P., & Chauhan, R. (2012). "Geography-Informed Sleep Scheduled and Chaining Based Energy Efficient Data Routing in WSN". *IEEE Students Conference on Electrical, Electronics and Computer Science*, pp. 1-4.
- [71]. Luo, H., Tao, H., Ma, H., & Das, S. K. (2011). "Data Fusion with Desired Reliability in Wireless Sensor Networks". *IEEE Transactions on Parallel and Distributed Systems*, 22 (3), 501-513.
- [72]. Majumder, K., & Ray, S. (2010). "A Novel Energy Efficient Chain Based Hierarchical Routing Protocol for Wireless Sensor Networks". *International Conference Emerging Trends in Robotics and Communication Technologies (INTERACT)*, pp. 339-344.
- [73]. Manjeshwar, A., & Agrawal, D. P. (2002). "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks". *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS*, pp. 1-8.

- [74]. Manjeshwar, A., & Agrawal, D. P. (2000). "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks". *Proceedings 15th International Parallel and Distributed Processing Symposium*, pp. 2009-2015.
- [75]. Muruganthan, S. D., Ma, D. C., Rollyi, Bhasin, & Fapojuwo, A. (2005). "A Centralized energy-efficient routing protocol for wireless sensor networks". *IEEE Radio Communications*, 43 (3), 8-13.
- [76]. Nguyen, L. T., Defago, X., Beuran, R., & Shinoda, Y. (2008). "An Energy Efficient Routing Scheme for Mobile Wireless Sensor Networks". *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pp. 568-572.
- [77]. Nguyen, T. G., So-In, C., & Nguyen, N. G. (2014). "Two Energy-Efficient Cluster Head Selection Techniques Based on Distance for Wireless Sensor Networks". *International Computer Science and Engineering Conference (ICSEC)*, pp. 33-38.
- [78]. Oladimeji, M. O., Turkey, M., & Dudley, S. (2017). "HACH: Heuristic Algorithm for Clustering Hierarchy protocol in wireless sensor networks". *Applied Soft Computing*, 55. 452-461.
- [79]. Oliveira, L.s. M., & Rodrigues, J. J. (2011). "Wireless Sensor Networks: a Survey on Environmental Monitoring". *Journal of Communication*, 6 (2), 143-151.
- [80]. Pandian, P. S., Safeer, K. P., Gupta, P., Shakunthala, D. T., Sundershesu, B. S., & Padaki, V. C. (2008). "Wireless Sensor Network for Wearable Physiological Monitoring". *Journal of Networks*, 3 (5), 21-29.
- [81]. Pantazis, N. A., Nikolidakis, S. A., & Vergados, D. D. (2012). "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey". *IEEE Communication Survey and Tutorials, Accepted for Publication*, 15 (2), 551-591.
- [82]. Patel, M. M., & Aggarwal, A. (2013). "Security Attacks in Wireless Sensor Networks: A Survey". In *International Conference on Intelligent Systems and Signal Processing (ISSP)*, pp. 329-333.
- [83]. Paulo Neves, M. S. (2008). "Application of Wireless Sensor Networks to Healthcare Promotion". *Journal of Communications Software and Systems (JCOMSS)*, 4 (3), 181-190.
- [84]. Pradhan, S. S., & Ramchandran, K. (2003). "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction". *IEEE Transactions on Information Theory*, 49 (3), 626-643.



- [85]. Priyankara, S., Kinoshita, K., Tode, H., & Murakami, K. (2010). "A Clustering/Multi-hop Hybrid Routing Method for Wireless Sensor Networks with Heterogeneous Node Types". *IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 207-212.
- [86]. Qiang, T., Bingwen, W., & Zhicheng, D. (2009). "MS-LEACH: A Routing Protocol Combining Multi-hop Transmissions and Single-hop Transmissions". *Pacific-Asia Conference on Circuits, Communications and Systems (PACCS)*, pp. 107-110.
- [87]. Qing, L., Zhu, Q., & Wang, M. (2006). "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks". *Computer Communications*, 29 (12), 2230-2237.
- [88]. Ren, F., Zhang, J., Tao He, C. L., & Das, S. K. (2011). "EBRP Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks". *IEEE Transactions on Parallel and Distributed Systems*, 22 (12), 2108-2125.
- [89]. Roopashree, H., & Kanavalli, A. (2015). "STREE: A Secured Tree based Routing with Energy Efficiency in Wireless Sensor Network". *International Conference on Computing and Communications Technologies (ICCCCT)*, pp. 25-30.
- [90]. Saleem, S., Ullah, S., & Kwak, K. S. (2011). "A Study of IEEE 802.15.4 Security Framework for Wireless Body Area Networks". *Sensors Open Access Journal*, 1383-1395.
- [91]. Satapathy, S. S., & Sarma, N. (2006). "TREEPSI: Tree Based Energy Efficient Protocol for Sensor Information". *International Conference on Wireless and Optical Communications Networks (IFIP)*, pp. 1-4.
- [92]. Sayood, K. (2003). "Lossless Compression Handbook". *California, USA: Academic Press*.
- [93]. Sen, F., Bing, Q., & Liangrui, T. (2011). "An Improved Energy-Efficient PEGASIS-Based Protocol in Wireless Sensor Networks". *Eighth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2230-2233.
- [94]. Shuaib, A. H., & Aghvami, A. H. (2009). "A Routing Scheme for the IEEE-802.15.4-Enabled Wireless Sensor Networks". *IEEE Transactions on Vehicular Technology*, 58 (9), 5135-5151.

- [95]. Shwe, H. Y., & Chong, P. H. (2015). "Cluster-based WSN Routing Protocol for Smart Buildings". *IEEE The 81<sup>st</sup> Vehicular Technology Conference (VTC Spring)*, pp. 1-5.
- [96]. Sohraby, K., Minoli, D., & Znati, T. (2007). "Wireless Sensor Network Technology, Protocols, and Applications". USA: John Wiley & Sons.
- [97]. Stankovic, V., Liveris, A. D., Xiong, Z., & Georgiades, C. N. (2006). "On code design for the Slepian-Wolf problem and lossless multiterminal networks". *IEEE Transactions on Information Theory*, 52 (4), 1495-1507.
- [98]. Sun, F., Zhao, Z., Zhen Fang, L. D., Xu, Z., & Chen, D. (2014). "A Review of Attacks and Security Protocols for Wireless Sensor Networks". *Journal of Network*, 9 (5), 1103-1113.
- [99]. Sun, Z., Tian, C., & Wong, K. M. (2010). "LDPC Code Design for Asynchronous Slepian-Wolf Coding". *IEEE Transactions on Communications*, 58 (2), 511-520.
- [100]. Swain, A. R., Hansdah, R. C., & Chouhan, V. K. (2010). "An Energy Aware Routing Protocol with Sleep Scheduling for Wireless Sensor Networks". In *24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 933-940.
- [101]. Taghikhaki, Z., Meratnia, N., & Havinga, P. J. (2013). "A Reliable and Energy-efficient Chain-cluster Based Routing Protocol for Wireless Sensor Networks". *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 248-253.
- [102]. Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011). "PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks". In *Proceedings IEEE INFOCOM*, pp. 1305-1313.
- [103]. Tang, Z., & Hu, Q. (2010). "ALLEE-MAC: An Adaptive Low Latency and Energy Efficient MAC Protocol for Wireless Sensor Networks". In *Sixth Advanced International Conference on Telecommunications (AICT)*, pp. 269-274.
- [104]. Tian, Y., Wang, Y., & Zhang, S. F. (2007). "A Novel Chain-Cluster Based Routing Protocol for Wireless Sensor Networks". *Proceedings The Second International Workshop on Education Technology and Computer Science*, pp. 2456-2459.
- [105]. Toyoda, S.-N., & Sato, F. (2012). "Energy-Effective Clustering Algorithm Based on Adjacent Nodes and Residual Electric Power in Wireless Sensor Networks". In

*26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 601-606.

- [106]. Wang, Y. C. (2012). "Data Compression Techniques in Wireless Sensor Networks". *Pervasive Computing*, New York: Nova Science Publishers, Inc.
- [107]. Werner-Allen, G., Lorincz, K., Welsh, M., Marcillo, O., Johnson, J., Ruiz, M., et al. (2006). "Deploying a wireless sensor network on an active volcano". *IEEE Internet Computing*, 10 (2), 18-25.
- [108]. Xinhua, W., & Jianbing, C. (2011). "A Partition-Based Hybrid Clustering Routing Protocol for WSN". *International Conference on Internet Technology and Applications (ITAP)*, pp. 1-4.
- [109]. Xiang, W., Wang, N., & Zhou, Y. (2016). "An Energy-Efficient Routing Algorithm for Software-Defined Wireless Sensor Networks". *IEEE Sensors Journal*, 16 (20), 7393-7400.
- [110]. Xinhua, W., & Sheng, W. (2010). "Performance Comparison of LEACH and LEACH-C Protocols by NS2". *Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES)*, pp. 254-258.
- [111]. Xiong, Z., Liveris, A. D., & Cheng, S. (2004). "Distributed Source Coding for Sensor Networks". *IEEE Signal Processing Magazine*, 21 (5), 80-94.
- [112]. Xu, J., Jin, N., Lou, X., Peng, T., Zhou, Q., & Chen, Y. (2012). "Improvement of LEACH protocol for WSN". *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2174-2177.
- [113]. Yang, Z., Liu, J., & Chen, X. (2009). "An Optimal Mechanism of LEACH Protocol for Wireless Sensor Networks". *The Second ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM)*, pp. 254-257.
- [114]. Ye, M., Li, C., Chen, G., & Wu, J. (2005). "EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks". *The 24th IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 535-540.
- [115]. Younis, O., & Fahmy, S. (2004). "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks". *IEEE Transactions on Mobile Computing*, 3 (4), 366-379.
- [116]. Yu, J. D., Kim, K. T., Jung, B. Y., & Youn, H. Y. (2009). "An Energy Efficient Chain-Based Clustering Routing Protocol for Wireless Sensor Networks". *The*

- IEEE 23rd International Conference on Advanced Information Networking and Applications Workshops*, pp. 383-388.
- [117]. Yu, Y., & Song, Y. (2010). "An Energy-Efficient Chain-Based Routing Protocol in Wireless Sensor Network". *International Conference on Computer Application and System Modeling (ICCASM)*, pp. 486-489.
  - [118]. Yu, Y., & We, G. (2007). "Energy Aware Routing Algorithm Based on Layered Chain in Wireless Sensor Network". *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2701-2704.
  - [119]. Zeng, B., Wei, J., & Hu, T. (2007). "An Energy-Efficient Data Fusion Protocol for Wireless Sensor Network". *The 10th International Conference on Information Fusion*, pp. 1-7.
  - [120]. Zhai, X., Jing, H., & Vladimirova, T. (2014). "Multi-Sensor Data Fusion in Wireless Sensor Networks for Planetary Exploration". *NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 188-195.
  - [121]. Zhang, M., Lu, Y., & Gong, C. (2008). "Energy-Efficient Routing Protocol based on Clustering and Least Spanning Tree in Wireless Sensor Networks". *Proceedings of the International Conference on Computer Science and Software Engineering*, pp. 361-364.
  - [122]. Zhou, R., Leibo, L., Yin, S., Luo, A., Chen, X., & Wei, S. (2010). "A VLSI design of sensor node for wireless image sensor network". *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 149-152.
  - [123]. Zonouz, A. E., Xing, L., Vokkarane, V. M., & Sun, Y. L. (2014). "Reliability-Oriented Single-Path Routing Protocols in Wireless Sensor Networks". *IEEE Sensors Journal*, 14 (11), 4059-4068.
  - [124]. Zytoune, O., Aboutajdine, D., & Tazi, M. (2010). "Energy Balanced Clustering Algorithm for Routing in heterogeneous Wireless Sensor Networks". *The 5th International Symposium on I/V Communications and Mobile Network (ISVC)*, pp. 1-4.
  - [125]. Project, 1997. "The Network Simulator - NS2". *The Network Simulator - NS2*. Available: <http://www.isi.edu/nsnam/ns> [Accessed 24 June 2017].
  - [126]. Heinzelman, W. (2004). MIT uAMPS LEACH ns Extensions. Available: <http://www.ece.rochester.edu/research/wcng/code/index.htm> [Accessed 24 June 2017].

- [127]. [http://en.wikipedia.org/wiki/IEEE\\_802.15.4](http://en.wikipedia.org/wiki/IEEE_802.15.4) [Accessed 24 June 2017].
- [128]. <http://www.zigbee.org> [Accessed 24 June 2017].
- [129]. Johnson, S. J. (2015). "Introducing Low-Density Parity-Check Codes", [https://www.researchgate.net/publication/228977165\\_Introducing\\_Low-Density\\_Parity-Check\\_Codes](https://www.researchgate.net/publication/228977165_Introducing_Low-Density_Parity-Check_Codes) [Accessed 24 June 2017].
- [130]. Welsh, B. M. (1967). "Sensor networks, circa 1967", <http://matt-welsh.blogspot.com/2010/01/sensor-networks-circa-1967.html> [Accessed 24 June 2017].
- [131]. <https://www.nsnam.org/docs/tutorial/html/index.html> [Accessed 24 June 2017].
- [132]. <https://www.isi.edu/nsnam/ns/CHANGES.html> [Accessed 10 July 2017].
- [133]. <https://www.nsnam.org/docs/release/3.24/tutorial/singlehtml/> [Accessed 10 July 2017].

## PHỤ LỤC

### Phụ lục 1. Lý thuyết Dempster-Shafer

Lý thuyết hiển nhiên đã được Dempster giới thiệu vào cuối những năm 1960, và sau đó được mở rộng bởi Shafer [2, 17, 20, 30, 60]. Lý thuyết hiển nhiên Dempster-Shafer miêu tả sự không chắc chắn về tính ngẫu nhiên và không chính xác, thông qua niềm tin và sự hợp lý. Chúng bắt nguồn từ một tập dữ liệu gọi là "khối lượng" (ví dụ, xác suất xuất hiện của một sự kiện, v.v.). Lý thuyết Dempster-Shafer cũng cung cấp các phương pháp biểu diễn, lập luận, các quy tắc (luật) cho phép tính toán, đối chiếu, tổng hợp, trộn dữ liệu ở các nguồn trạng thái khác nhau dựa trên các dữ liệu cả cũ lẫn mới.

Giả sử ta có một tập hữu hạn các khả năng có thể xảy ra của các sự kiện loại trừ lẫn nhau là  $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ , gọi là không gian mẫu. Mỗi phần tử  $\theta_i$  trong tập có xác suất giữa 0 và 1, do đó, có ít nhất một phần tử là đúng (xác suất = 1). Số lượng các tập con có thể lấy từ tập  $\Theta$  là  $2^n$  tập con.

#### Ví dụ P1.1:

Tập  $\Theta = \{B, Y, R\}$  khi ta quan sát đèn giao thông tương ứng với các tín hiệu (B: đèn xanh, Y: đèn vàng và R: đèn đỏ) thì, các tập con có thể lấy ra từ  $\Theta$  là  $(\{\emptyset\}, \{B\}, \{Y\}, \{R\}, \{B,Y\}, \{B,R\}, \{Y,R\}, \{B,Y,R\})$ , ở đây tập rỗng  $\{\emptyset\}$  có xác suất bằng 0.

Gọi  $m(A)$  là hàm "khối lượng" (mass) diễn tả sự cân xứng của các tương quan và tính hiển nhiên sẵn có của  $A$  ( $A$  là tập con thành viên của  $\Theta$ ). Ta có:  $m(A) \rightarrow [0,1]$ ,  $m(\emptyset) = 0$ ,  $m(\Theta) = 1$ ,  $0 \leq m(A) \leq 1$  cho tất cả các tập con  $A \subseteq \Theta$ , và  $\sum_{A \subseteq \Theta} m(A) = 1$ .

Giả sử, sau khi quan sát, ta có trọng số của các phần tử trong tập  $\Theta$  tương ứng trong Bảng P1.1 sau:

**Bảng P1.1:** Kết quả quan sát của các phần tử trong tập  $\Theta$

Sự kiện	Khối lượng $m(A)$	$bel(A)$	$pl(A)$
Không có	0	0	0
B	0.1	0.1	0.4
Y	0.2	0.2	0.7
R	0.1	0.1	0.6
B, Y	0.1	0.1	0.9
B, R	0.1	0.1	0.8
Y, R	0.3	0.3	0.9
B, Y, R	0.1	0.1	0.1

Hàm chỉ độ đo “*niềm tin*” (**belief**) của A trong tập toàn bộ là tổng của các hàm “*khối lượng*” trong tập con của A, chúng là tập con của A (bao gồm cả chính nó).

$$bel(A) = \sum_{B \subseteq A; B \neq \emptyset} m(B) \quad (P1.1)$$

Với ví dụ trên thì, ta có:

$$bel(\{B\}) = m(\{B\}) = 0.1$$

$$bel(\{B, Y\}) = m(\{B\}) + m(\{Y\}) + m(\{B, Y\}) = 0.1 + 0.2 + 0.1 = 0.4, \text{ và}$$

$$bel(\Theta) = m(\{B\}) + m(\{Y\}) + m(\{R\}) + m(\{B, Y\}) + m(\{B, R\}) + m(\{Y, R\}) + m(\{B, Y, R\})$$

$$bel(\Theta) = 0.1 + 0.2 + 0.1 + 0.1 + 0.1 + 0.3 + 0.1 = 1$$

Hàm đối chiếu của hàm niềm tin là độ đo về sự “*đáng tin*” (plausibility)

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad (P1.2)$$

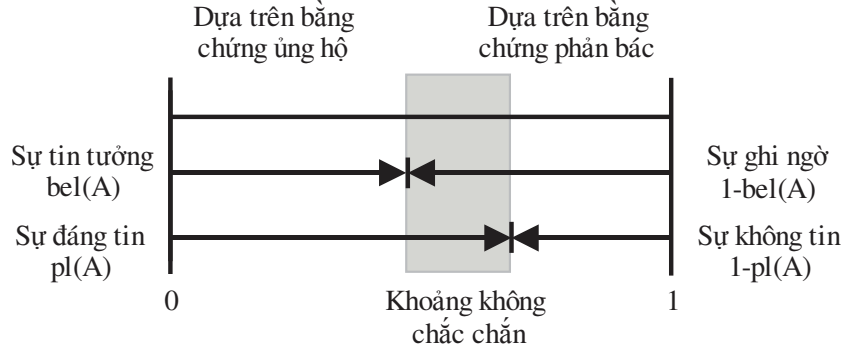
**Ví dụ P1.2:**

$$\begin{aligned} pl(\{B, Y\}) &= m(\{B\}) + m(\{Y\}) + m(\{B, Y\}) + m(\{B, R\}) + m(\{Y, R\}) + m(\{B, Y, R\}) \\ &= 0.1 + 0.2 + 0.1 + 0.1 + 0.3 = 0.9 \end{aligned}$$

Độ chắc chắn liên kết với một tập con của A cho ta khoảng tin cậy được định nghĩa bởi: **[bel(A) pl(A)]**, ví dụ khoảng tin cậy của {B, Y} là: **[0.1 0.7]**

Hình P1.1 minh họa về các độ đo được định nghĩa ở trên (**belief** and **plausibility**)

$bel(A) < pl(A)$ :



**Hình P1.1:** Khoảng không chắc chắn giữa “niềm tin”, sự “đáng tin”

Đối lập với “niềm tin”, sự “đáng tin” đó là sự nghi ngờ và không chắc chắn:

$$pl(\bar{A}) = 1 - bel(A), \quad bel(\bar{A}) = 1 - pl(A), \quad \text{với } bel(\bar{A}) \leq pl(\bar{A})$$

Đối lập với sự tin cậy “niềm tin” là sự không tin cậy (Disbelief) hay là sự ghi ngờ (Doubt) của A, ký hiệu là  $pl(\bar{A})$  và sự không “đáng tin” ký hiệu là: **disbelief(A)** hay là  $bel(\bar{A})$ , nó được tính bằng tổng trọng số các tập không chứa A trong tập toàn bộ  $\Theta$ .

**Luật kết hợp:** Giả sử nếu có thông tin từ hai nguồn niềm tin  $bel_1$  và  $bel_2$  thuộc tập  $\Theta$  với các hàm trọng số của nó tương ứng là  $m_1(Y)$  và  $m_2(R)$ . Dempster-Shafer đưa ra một luật kết hợp để tính toán độ đo niềm tin B cho các hàm không rỗng từ hai nguồn như sau [20, 30, 60]:

$$[bel_1 \oplus bel_2](B) = \frac{\sum_{Y \cap R = B \neq \emptyset} m_1(Y) \times m_2(R)}{1 - K} \quad (P1.3)$$

$$K = \sum_{Y \cap R = \emptyset} m_1(Y) \times m_2(R) \quad (P1.4)$$

trong đó, K là mức độ của cuộc xung đột ở hai nguồn niềm tin  $bel_1$  và  $bel_2$ ,  $\oplus$  chỉ định hoạt động kết hợp (phép cộng theo modulo-2) và  $m_i$  là hàm trọng số được cung cấp bởi nguồn i. Phần mẫu số (1-K) là giúp cho việc tổng hợp dữ liệu hoàn thành bằng cách bỏ qua các bằng chứng xung đột.



### **Ví dụ P1.3:**

Giả sử có hai nút cảm biến không dây cùng quan sát môi trường để đưa ra dự báo về thời tiết. Các giá trị mà nút cảm biến 1 quan sát được có thể là A (có mưa 30%), hoặc C (có mây 30%) hoặc {A, C} (mưa và có mây 30%) và  $\theta$  (trường hợp khác 10%). Nút cảm biến 2 quan sát được: A (có mưa 60%), hoặc {A, B} (mưa và nắng 30%) và  $\theta$  (trường hợp khác 10%) [20].

Ta có tập:  $\Theta = \{\emptyset, A, B, C, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

Từ giá trị quan sát được, ta có:

$$m_1(A) = 0.3, m_1(C) = 0.3, m_1(A, C) = 0.3, m_1(\theta) = 0.1;$$

$m_2(A) = 0.6, m_2(A, B) = 0.3, m_2(\theta) = 0.1$ ; các trường hợp khác có giá trị bằng không như:  $m_1(B) = 0, m_1(B, C) = 0, m_2(A, C) = 0, \dots$

Hàm chỉ độ đo “niềm tin” (**belief**) của A, B như sau:

$$bel_1(A) = m_1(A) = 0.3, bel_1(C) = 0.3, bel_1(A, C) = m_1(A, C) = 0.3,$$

$$bel_1(\Theta) = m_1(A) + m_1(C) + m_1(A, C) + m_1(\theta) = 0.3 + 0.3 + 0.3 + 0.1 = 1.0;$$

$$bel_2(A) = m_2(A) = 0.6, bel_2(C) = 0.3, bel_2(A, B) = m_2(A, B) = 0.3,$$

$$bel_2(\Theta) = m_2(A) + m_2(A, B) + m_2(\theta) = 0.6 + 0.3 + 0.1 = 1.0;$$

Hàm chỉ độ đo về sự “đáng tin” (plausibility) của A, B như sau:

$$\{A\} \cap \{A\} = \{A\}, \neq \emptyset \text{ do đó } m_1(A) = 0.3,$$

$$\{C\} \cap \{A\} = \{\emptyset\}; \{A, C\} \cap \{A\} = \{A\}, \neq \emptyset; \{\theta\} \cap \{A\} = \{A\}, \neq \emptyset$$

$$\{A, B\} \cap \{A\} = \{A\}, \neq \emptyset; \{A, B\} \cap \{\emptyset\} = \{\emptyset\}; \{A, B\} \cap \{A, C\} = \{A\}, \neq \{\emptyset\};$$

$$\{A, B\} \cap \{\theta\} = \{A, B\}, \neq \{\emptyset\};$$

$$\{\theta\} \cap \{A\} = \{A\}, \neq \emptyset; \{\theta\} \cap \{C\} = \{C\}, \neq \emptyset; \{\theta\} \cap \{A, C\} = \{C\}, \neq \emptyset;$$

$$\{\theta\} \cap \{\theta\} = \{\theta\}, \neq \emptyset;$$

Từ đó ta có:

$$m_{12}(A) = m_1(A)m_2(A) + m_1(A)m_2(B) + m_1(A)m_2(\theta) \\ + m_1(A, C)m_2(A) + m_1(A, C)m_2(A, B) + m_1(\theta)m_2(A)$$

$$m_{12}(C) = m_1(A)m_2(A) + m_1(A)m_2(B) + m_1(A)m_2(\theta) \\ + m_1(A, C)m_2(A) + m_1(A, C)m_2(A, B) + m_1(\theta)m_2(A)$$

Do đó, ta có bảng dữ liệu sau:

**Bảng P1.2:** Kết quả tính toán “niềm tin” và sự “đáng tin” của A và B

			Cảm biến 1				
			$m_1(A)$	$m_1(C)$	$m_1(A, C)$	$m_1(\theta)$	
			0.3	0.3	0.3	0.1	1
Cảm biến 2	$m_2(A)$	0.6	$m_1(A)m_2(A)$ =0.18	$m_1(A)m_2(C)$ = $\emptyset$	$m_1(A,C) m_2(A)$ =0.18	$m_1(\theta)m_2(A)$ =0.06	
	$m_2(A,B)$	0.3	$m_1(A)m_2(A,B)$ =0.09	$m_1(A,B)m_2(C)$ = $\emptyset$	$m_1(A,C)m_2(A,B)$ =0.09	$m_1(\theta)m_2(A,B)$ =0.03	
	$m_2(\theta)$	0.1	$m_1(A)m_2(\theta)$ =0.03	$m_1(C) m_2(\theta)$ =0.03	$m_1(A,C) m_2(\theta)$ =0.03	$m_1(\theta)m_2(\theta)$ =0.03	
		1					

Sử dụng luật kết hợp theo (P1.3) và (P1.4) cho hai cảm biến ta có:

$$K = 0.6*0.3+0.3*0.3=0.27,$$

$$[bel_1 \oplus bel_2](A) = \frac{0.18 + 0.09 + 0.03 + 0.18 + 0.09 + 0.06}{1 - 0.27} = 0.863$$

$$[bel_1 \oplus bel_2](\theta) = \frac{0.01}{1 - 0.27} = 0.0137$$

$$[bel_1 \oplus bel_2](A, C) = \frac{0.03}{1 - 0.27} = 0.0411$$

$$[bel_1 \oplus bel_2](C) = \frac{0.03}{1 - 0.27} = 0.0411$$

$$[bel_1 \oplus bel_2](A, B) = \frac{0.03}{1 - 0.27} = 0.0411$$

Dựa vào kết quả kết hợp luật của D-S, ta có thể thấy có tới 86,3% thời tiết là mưa vào ngày mai (theo dữ liệu quan sát được từ hai nút cảm biến).

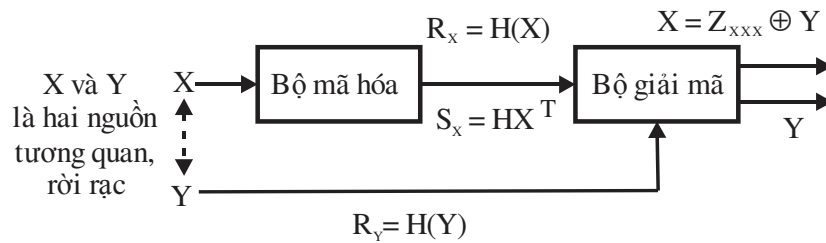
Ta có trường hợp tổng quát, sau với n nguồn quan sát [30, 60]:

$$\bigoplus_{i=1}^n bel_i(B) = \frac{\sum_{B_1 \cap \dots \cap B_n = B \neq \emptyset} \prod_{i=1}^n m_i(B_i)}{1 - K} \quad (P1.5)$$

$$K = \sum_{B_1 \cap \dots \cap B_n = \emptyset} \prod_{i=1}^n m_i(B_i) \quad (P1.6)$$

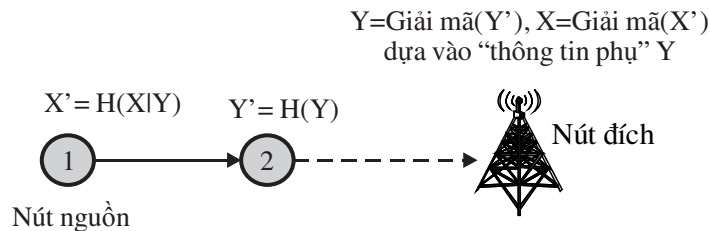
## Phụ lục 2. Mã nguồn phân tán (Distributed source coding)

Một trong những kỹ thuật của mã nguồn phân tán dựa trên lý thuyết Slepian-Wolf, chúng được sử dụng để nén hai nguồn dữ liệu tương quan không mất dữ liệu và rất phù hợp cho việc tiết kiệm năng lượng trong mạng cảm biến không dây [33, 97, 111]. Để đạt được việc nén không mất dữ liệu của các nguồn tương quan từ nhiều nút cảm biến gửi về với tốc độ nén lớn hơn hoặc bằng Entropi đồng thời. Giả sử chúng ta có hai nguồn  $X$  và  $Y$ , mỗi nguồn sinh ra  $u$  bit dữ liệu nhị phân, như được biểu diễn ở Hình P2.1 [84, 106].



**Hình P2.1:** Mã hóa độc lập và giải mã đồng thời hai nguồn dữ liệu tương quan  $X$  và  $Y$

Giả sử hai nguồn dữ liệu  $X$  và  $Y$  là có tính tương quan cao, hay khoảng cách Hamming giữa hai nguồn  $dH(X, Y) \leq 1$ , hay xác suất xuất hiện của các giá trị trong nguồn là  $\Pr(X_i = 0) = \Pr(X_i = 1) = \Pr(Y_i = 0) = \Pr(Y_i = 1) = 0.5$ , với mọi  $i=0$  đến  $u$ . Nếu giá trị nguồn  $Y$  đã có ở cả bộ mã hóa và bộ giải mã như "thông tin phụ", chúng gồm  $u$  bit dữ liệu, thì chúng ta có thể biểu diễn và truyền  $X \oplus Y$  trong  $H(X|Y)$  bit trên mẫu thay vì truyền nguyên giá trị nguồn  $X$ , ở đây  $H(X|Y)$  và  $\Pr(X)$  chỉ định hàm Entropi có điều kiện và hàm mật độ phân bố xác suất của nguồn  $X$  tương ứng [68].



**Hình P2.2:** Triển khai mã nguồn phân tán trong mạng cảm biến không dây

Ở bộ giải mã, giá trị nguồn X có thể được khôi phục bằng cách tính XOR nhị phân ( $\oplus$ ) với hai trường hợp sau:

**Trường hợp 1:** Không mất tính tổng quát, chúng tôi giả sử rằng mã Slepian-Wolf của hai nguồn X và Y là được phân bố đồng nhất với  $2^u$  mẫu ( $u = 4$  bit nhị phân cho mỗi mẫu) [33, 111].

Đặt  $a = u - b$ , với  $a$  là một số nguyên và  $2^a - 1 \geq dH(X, Y)$ , thì  $H(X) = H(Y) = 4$  bit trên mẫu,  $H(X|Y) = b = 3$  bit, và  $H(X, Y) = 4+3 = 7$  bit trên cặp mẫu cho giải mã đồng thời, với  $H(X|Y)$  chỉ định hàm Entropi có điều kiện và  $H(X, Y)$  và hàm Entropi đồng thời của X và Y. Do đó, chúng ta có thể nhóm  $2^4$  mẫu như  $2^3 = 8$  tập con, trong mỗi tập con, khoảng cách Hamming giữa hai phần tử là lớn hơn hoặc bằng 4 và gán 8 chỉ số nhị phân khác nhau bởi  $Z_{xxx}$ , tương ứng như sau:  $\{Z_{000} = (0000, 1111): 000, Z_{001} = (0111, 1000): 001, Z_{010} = (0100, 1011): 010, Z_{011} = (0011, 1100): 011, Z_{100} = (0010, 1101): 100, Z_{101} = (0101, 1010): 101, Z_{110} = (0110, 1001): 110, Z_{111} = (0001, 1110): 111\}$  [84, 68]. Chúng tôi xem xét truyền dữ liệu nguồn Y, dữ liệu của nó được sử dụng như "thông tin phụ", ví dụ  $\{0100\}$  và chỉ số của X là  $\{110\}$ . Khi bộ giải mã nhận được chỉ số là  $\{110\}$ , giá trị của X thuộc tập con  $\{0110, 1001\}$  trong  $Z_{110}$ . Bộ giải mã có thể lấy được giá trị chính xác của X là  $\{0110\}$  vì khoảng cách Hamming giữa X và Y là nhỏ hơn hoặc bằng một ( $dH(X, Y) = 1$ ).

Thực tế, "thông tin phụ" Y là hoàn toàn sẵn có ở bộ giải mã nhưng việc mã hóa đồng thời (ở đây là tại các nút cảm biến, nút sinh ra X) là khó để triển khai trong các nút cảm biến, như được minh họa ở Hình P2.1 và P2.2. Giá trị nguồn Y chỉ có ở bộ giải mã (ví dụ BS hoặc sink), do vậy, làm thế nào chúng ta có thể nén hai nguồn X và Y trong tổng  $(u + b)$  bit giải mã không mất như ví dụ trên? Chúng ta có thể sử dụng mô hình mã khối tuyến tính (Linear Block Code -LBC) trong (Galois Field- $GF(2)$ ) [129] với mẫu đại diện khối  $(u, b)$  LBC, chúng được cho bởi  $G = [I_b: P]$  và  $H = [P^T: I_{u-b}]_{(u-b) \times u}$  với  $GH^T = 0$ , ở đây  $G$  là ma trận sinh  $b$  hàng  $u$  cột và  $P^T$  là ma trận chuyển vị  $b$  hàng  $(u - b)$  cột, tương ứng.  $H$  là ma trận kiểm tra chẵn lẻ được sử dụng ở bộ giải mã và  $G$  được sử dụng ở bộ mã hóa. Với ví dụ trên, chúng ta có ma trận  $H$  và  $G$  như sau:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad H = [1 \quad 1 \quad 1 \quad 1]$$

Do vậy, bộ mã hóa sẽ tính toán giá trị nguồn  $s = XG^T$ , (kết hợp với tập con  $Z$ ,  $X \in Z_s$ ), phần tử mà có chỉ số của  $X$  sử dụng  $H(X|Y) = b$  bit thay vì  $u$  bit dữ liệu gốc, ta sẽ đạt được mã Slepian-Wolf với tỉ lệ nén  $u:b$  [111].

**Ví dụ P2.1:** với  $X = (0101)$  và  $Y = (0100)$

$$s = X \times G^T = [0 \quad 1 \quad 0 \quad 1] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = [1 \quad 0 \quad 1]$$

Giá trị sau khi nén sẽ là  $(101) = 3$  bit.

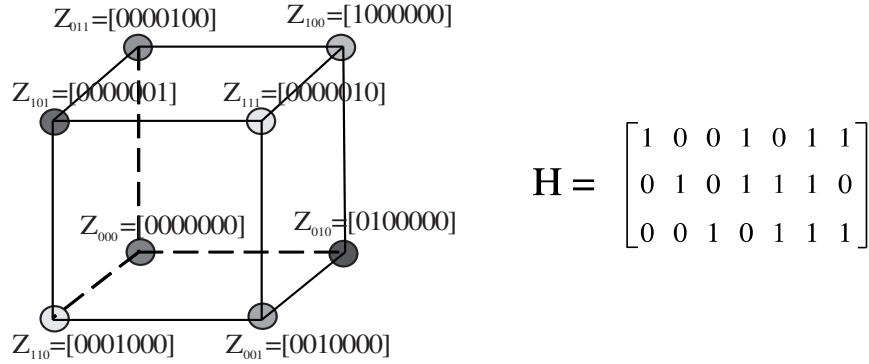
Để giải nén, chúng ta sử dụng  $Y = (0100)$  kết hợp với tập  $Z$  có chỉ số  $= (101)$ , tìm giá trị của  $X$  trong tập  $Z_{101} = (0101, 1010)$ , giá trị mà có khoảng cách Hamming với  $Y$  bằng một, ta có  $X = (0101)$ .

**Trường hợp 2:** Giả sử rằng mã Slepian-Wolf [33, 97] của hai nguồn  $X$  và  $Y$  là được phân bố đồng nhất với  $2^u$  mẫu ( $u = 7$  bit nhị phân cho mỗi mẫu).

Đặt  $b = u - a$ , với  $a$  là một số nguyên, thì  $H(X) = H(Y) = 7$  bit trên mẫu và giá trị  $X \oplus Y$  sẽ thuộc vào một trong các phần tử sau:  $\{(0000000), (0000001), \dots, (1000000)\}$ , chúng ta chỉ thực hiện  $8=2^3$  giá trị khác nhau. Do đó,  $H(X|Y) = b = 3$  bit, và  $H(X, Y) = 7+3=10$  bit trên cặp mẫu cho việc giải mã đồng thời, với  $H(\cdot)$  là hàm Entropi đồng thời. Chúng ta có thể gán  $2^3$  giá trị khác nhau tương ứng với  $8=2^3$  tập con, chúng được đánh số  $Z_{xxx}$ , tương ứng như sau:  $Z_{000} = (0000000)$ ,  $Z_{001} = (0010000)$ ,  $Z_{010} = (0100000)$ ,  $Z_{011} = (0000100)$ ,  $Z_{100} = (1000000)$ ,  $Z_{101} = (0000001)$ ,  $Z_{110} = (0001000)$ ,  $Z_{111} = (0000010)$  vì khoảng cách Hamming giữa  $X$  và  $Y$  là nhỏ hơn hoặc bằng 1 ( $dH(X, Y) \leq 1$ ). Do đó, chúng ta xem xét truyền nguồn  $Y$  như  $(1001110)$ ,  $X$  như  $(1001100)$  và  $X \oplus Y = (0000010)$ , tương ứng với  $Z_{111}$ .

Để giải quyết vấn đề mã hóa độc lập, giải mã đồng thời, chúng ta có thể sử dụng ma trận kiểm tra chẵn lẻ nhị phân  $H$   $a$  hàng  $b$  cột, chúng được đề xuất bởi  $\{[I_b : P^T]\}_{b \times u}$

in GF(2) (Galois Field) [129], với  $P^T$  ma trận chuyển vị P của (u - b) hàng b cột như trên.



**Hình P2.3:** Biểu diễn giá trị tập con  $Z_{XXX}$  với  $u=7$  bit

Bộ mã hóa đơn giản tính toán  $s_X = H.X^T$  trong  $Z_{XXX}$  của nguồn X. Sau khi nhận được  $s_X$ , bộ giải mã sẽ tính toán  $s_Y = H.Y^T$ .

Kết hợp với các tập con của  $Z_{XXX}$ , chúng ta sẽ có  $[s_X \oplus s_Y]^T = Z_{XXX}$ .

Như ví dụ trên,  $s_X = [001]^T$ ,  $s_Y = [110]^T$ ,  $[s_X \oplus s_Y]^T = [111]$ .

Và do đó, ta có  $X = Z_{111} \oplus Y = [1001100]$  như mong muốn.

**Ví dụ P2.2:** Với  $X = [0011011]$  và  $Y = [0111011]$

$$s_X = HX^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$s_Y = HY^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\text{và tính: } [s_x \oplus s_y]^T = \left[ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right]^T = [0 \ 1 \ 0]$$

$Z_{010} = [0100000]$ , và,  $X = Y \oplus Z_{010} = [0111011] \oplus [0100000] = [0011011]$ , đây chính là giá trị của X cần tìm.