

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----o0o-----

**NGHIÊN CỨU MẠNG CẢM BIẾN KHÔNG DÂY – WSN VÀ NHỮNG**  
**ĐẶC ĐIỂM SỬA LỖI TRUYỀN DỮ LIỆU**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Sinh viên thực hiện:        Hồ Anh Tuấn**

**Giáo viên hướng dẫn:      PGS.TS.Vương Đạo Vy**

**Mã số sinh viên:            110786**

---

---

## MỤC LỤC

MỤC LỤC.....	2
DANH MỤC HÌNH VẼ.....	4
DANH MỤC TỪ VIẾT TẮT.....	5
LỜI CẢM ƠN.....	6
MỞ ĐẦU.....	7
CHƯƠNG 1: TỔNG QUAN MẠNG CẢM BIẾN KHÔNG DÂY.....	8
1.1 Cái nhìn ban đầu về WSN.....	8
1.2 Nút cảm biến không dây.....	8
1.2.1 Phần cứng của nút cảm biến không dây.....	9
1.2.2 Chuẩn cho nút cảm biến không dây.....	10
1.2.3 Phần mềm cho nút cảm biến không dây.....	11
1.2.4 Một số loại nút cảm biến không dây.....	12
1.3 Kiến trúc và giao thức.....	13
1.3.1 Kiến trúc mạng của WSN.....	13
1.3.2 Giao thức Stack.....	14
1.4 Ứng dụng của WSN.....	15
1.5 Các yếu tố ảnh hưởng đến thiết kế.....	17
1.5.1 Hạn chế phần cứng.....	17
1.5.2 Khả năng chịu lỗi.....	17
1.5.3 Khả năng mở rộng.....	18
1.5.4 Chi phí sản xuất.....	18
1.5.5 Cấu trúc liên kết.....	18
1.5.6 Phương tiện truyền thông.....	18
1.5.7 Năng lượng tiêu thụ.....	19
CHƯƠNG 2: KIỂM SOÁT LỖI TRONG WSN.....	26
2.1 Tổng quan về vấn đề kiểm soát lỗi trong WSN.....	26
2.2 Các phương án kiểm soát Lỗi trong WSN.....	27
2.2.1 Kiểm soát năng lượng.....	27
2.2.2 Tự động phát lại (ARQ).....	28
2.2.3 Sửa lỗi trước khi truyền (FEC).....	28
2.2.4 ARQ lai ghép (HARQ).....	35
2.2 Lợi ích của tăng khả năng phục hồi lỗi.....	36
2.3 Phân tích mô hình lớp chéo.....	38

---

---

2.3.1	Mô hình tham chiếu của WSN.....	38
2.3.2	Khoảng cách bước nhảy dự kiến. ....	39
2.3.3	Phân tích năng lượng tiêu thụ dự kiến. ....	41
2.3.4	Phân tích độ trễ dự kiến. ....	45
2.3.5	Phân tích BER và PER.....	46
2.4	So sánh các phương án kiểm soát Lỗi trong WSN.....	47
<b>CHƯƠNG 3: BÀI TOÁN KIỂM SOÁT LỖI TRONG WSN .....</b>		<b>52</b>
3.1	Vấn đề và giải pháp trong WSN.....	52
3.2	Bài toán so sánh giữa sửa lỗi và phát lại trong WSN.....	53
3.2.1	Phát biểu bài toán.....	53
3.2.2	Nhận định từ bài toán.....	53
3.2.3	Giải quyết bài toán.....	54
<b>KẾT LUẬN .....</b>		<b>57</b>
<b>TÀI LIỆU THAM KHẢO.....</b>		<b>58</b>

---

---

## DANH MỤC HÌNH VẼ

<b>Hình 1.1.</b> Sơ đồ cấu tạo chung của một nút cảm biến không dây. ....	9
<b>Hình 1.2.</b> Thành phần kiến trúc dạng chuẩn WirelessHART.....	11
<b>Hình 1.3.</b> Kiến trúc mạng WSN. ....	14
<b>Hình 1.4.</b> Giao thức Stack của WSN. ....	14
<b>Hình 1.5.</b> Mô hình một số dự án ứng dụng của WSN. ....	16
<b>Hình 1.6.</b> Mức độ tiêu thụ năng lượng của nút cảm biến MicaZ. ....	20
<b>Hình 1.7.</b> Đơn giản hóa năng lượng tiêu thụ cho một cặp nút. ....	24
<b>Hình 2.1.</b> Tổng quan lớp liên kết dữ liệu.....	27
<b>Hình 2.2.</b> Mô hình xử lý trong WSN.....	29
<b>Hình 2.3.</b> Sơ đồ mã hóa byte dữ liệu. ....	29
<b>Hình 2.4.</b> Biểu đồ tần suất lỗi bit của một nút cảm biến khi phát 10000 gói và tỷ lệ lỗi gói tin mở đầu theo khoảng cách. ....	30
<b>Hình 2.5.</b> Sơ đồ quy trình giải mã - truyền - mã hóa.....	32
<b>Hình 2.6.</b> Trung bình tỉ lệ mục tiêu nhận gói tin, đồ thị (a) ARQ (N = 7) và (b) FEC (BCH (128, 78, 7)) cho nút MicaZ.....	36
<b>Hình 2.7.</b> So sánh hiệu quả của FEC với ARQ. ....	37
<b>Hình 2.8.</b> Mô hình tham chiếu của WSN. ....	38
<b>Hình 2.9.</b> Trung bình khoảng cách bước nhảy (MicaZ).....	48
<b>Hình 2.10.</b> Năng lượng tiêu thụ của một lưu lượng với $\psi_{Th}$ (MicaZ).....	48
<b>Hình 2.11.</b> PER Tổng thể với $\psi_{Th}$ (MicaZ). ....	49
<b>Hình 2.12.</b> Độ trễ trung bình tổng thể với $\psi_{Th}$ (MicaZ).....	50
<b>Hình 2.13.</b> (a) Năng lượng tiêu thụ trung bình của một luồng. (b) Độ trễ tổng thể trung bình với ngưỡng ( $\psi_{Th}$ ) cho những Pt khác nhau.....	51

## DANH MỤC TỪ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Nghĩa tiếng anh</b>	<b>Nghĩa tiếng việt</b>
ARQ	Automatic Repeat Request	Tự động phát lại.
ADC	Analog-to-Digital Converter	Bộ chuyển đổi tương tự sang số.
BER	Bit Error Rate	Tỷ lệ lỗi bit.
DC	Drection Current	Dòng điện một chiều.
DSP	Digital Signal Processing	Xử lý tín hiệu số.
ECC	Error Correct Code	Mã sửa lỗi.
FEC	Forward Error Correction	Sửa lỗi trước khi truyền.
HARQ	Hybrid Automatic Repeat Request	Tự động phát lại lai ghép FEC.
PLL	Phase Locked Loop	Vòng khóa pha.
PER	Packet Error Rate	Tỷ lệ lỗi gói tin.
SNR	Signal to Noise Ratio	Tỷ lệ nhiễu tín hiệu.
VCO	Voltage Controlled Oscillator	Bộ điều khiển điện áp dao động
WSN	Wireless sensor networks	Mạng cảm biến không dây.

## LỜI CẢM ƠN

Cám ơn các thầy cô giáo trường Đại học Dân lập Hải Phòng, đã dạy dỗ chúng em trong nhiều năm qua. Cám ơn thầy Trần Hữu Nghị đã cho em một mái trường để cho chúng em có cơ hội học được những kiến thức bổ ích để có thể trở thành một công dân có ích cho xã hội. Xin chân thành cảm ơn thầy cô bộ môn Tin học đã truyền đạt kiến thức về công nghệ thông tin, một môn học bổ ích, là hành trang vững chắc để em tự tin trong công việc sau này.

Cám ơn thầy Vương Đạo Vy, trường đại học công nghệ, Đại học Quốc gia Hà Nội đã giúp đỡ em trong quá trình viết đề án cũng như quá trình học tập trên ghế nhà trường. Để em có thể đem kiến thức mình đã học được trên ghế nhà trường áp dụng vào thực tiễn để em có thể nhận thấy mình đã trang bị được những gì còn thiếu những gì trong hành trang của mình.

Cám ơn gia đình và người thân, đã tận tình giúp đỡ, chu cấp tài chính, động viên em trong suốt thời gian học tập tại trường.

Xin cảm ơn các bạn bè trong lớp và các bạn trong khoa cũng như sinh viên cả trường đã giúp đỡ tôi trong thời gian học tập cũng như trong thời gian làm thực tập tốt nghiệp.

Hải Phòng, tháng 7 năm 2011.

Sinh viên

Hồ Anh Tuấn

## MỞ ĐẦU

Wireless Sensor Networks (WSN) hay mạng cảm biến không dây, một xu hướng phát triển của thời đại ngày nay. Với khả năng cảm nhận, cung cấp các thông tin thực tế và triển khai, mở rộng phạm vi dễ dàng nhờ triển khai trên mô hình mạng truyền thông không dây.

WSN gồm tập hợp các nút cảm biến rất nhỏ, hoạt động độc lập nguồn nuôi và thông qua hàng loạt các nút cảm biến để nắm bắt thông tin dữ liệu. Với bộ xử lý riêng, các nút cảm biến có thể được lập trình để hoàn thành nhiệm vụ phức tạp hơn ngoài việc xử lý đơn giản như thu, phát, chuyển tiếp dữ liệu. Tuy nhiên, Một thách thức sống còn của WSN là vấn đề năng lượng, nó tạo nên một thách thức lớn với WSN chính là giảm thiểu tối đa tiêu thụ năng lượng nhằm giữ hoạt động lâu dài cho các nút mạng.

Mục đích của đồ án này là cung cấp một cái nhìn tổng quan về WSN. Chương 1 là sự giới thiệu toàn diện về WSN, bao gồm các nút cảm biến và kiến trúc mạng, cung cấp cái nhìn toàn diện về các đặc điểm, các yếu tố thiết kế quan trọng, và khó khăn của WSN, đồng thời đưa ra một số ứng dụng hiện có của WSN trong quân sự, y tế, công nghiệp và ứng dụng tại nhà. Chương 2 tập trung đi sâu vào các kỹ thuật kiểm soát lỗi trong WSN cũng như tác động của nó lên truyền thông, tiết kiệm năng lượng. Cuối cùng, chương 3 giải quyết vấn đề cụ thể trong một bài toán để từ đó thấy được tầm quan trọng của sửa lỗi truyền dữ liệu trong WSN.

---

---

## CHƯƠNG 1: TỔNG QUAN MẠNG CẢM BIẾN KHÔNG DÂY

### 1.1 Cái nhìn ban đầu về WSN.

Hiện nay, công nghệ vi mạch, truyền thông không dây và điện tử kỹ thuật số phát triển ngày càng mạnh mẽ. Nhờ đó những thiết kế và quá trình triển khai với giá cả ngày càng thấp, năng lượng tiêu thụ được giảm thiểu đã tạo điều kiện cho những nút cảm biến đa chức năng có kích thước nhỏ và giao tiếp trong khoảng cách ngắn trở nên khả thi. Khả năng của các nút cảm biến ngày càng tăng trong đó bao gồm: cảm biến, xử lý dữ liệu và giao tiếp với một lượng lớn các nút cảm biến.

WSN cấu thành từ một lượng lớn các nút cảm biến, truyền thông multi-hop là chủ yếu. Do đó, nó có khả năng triển khai với quy mô lớn, tương tác nhanh chóng và đáng tin cậy nhờ sự tổng hợp thông tin hiệu quả giữa các nút. Hơn nữa, không chỉ truyền thông tin thô, các nút cảm biến còn có khả năng tự xử lý tính toán trước khi truyền đi. Về triển khai, mạng này cho phép triển khai một cách ngẫu nhiên do đó thích hợp với cả những vùng thiên tai và những địa hình phức tạp.

Tuy nhiên, WSN vẫn còn gặp một số thách thức nhất định để trở thành một mạng hoàn thiện. Thông thường các giao thức truyền thông yêu cầu cung cấp năng lượng với hiệu suất cao do đó chỉ tập trung vào thông lượng và độ trễ còn với WSN với nguồn cung cấp năng lượng hạn chế mà thách thức chính của nó là tìm cách để giảm năng lượng tiêu thụ. Qua đó mà việc phát triển WSN phải tập trung vào giảm thiểu tối đa năng lượng nguồn cấp. Trong chương 2 sau này, yếu tố sửa lỗi trong mạng được nghiên cứu khá kỹ nhằm giải quyết vấn đề này.

Những phần tiếp sau, chúng ta đi sâu vào những yếu tố quan trọng tạo lên WSN. Đó là, nút cảm biến không dây: góp phần tạo nên cơ sở hạ tầng vật lý cho WSN, kiến trúc và giao thức: yếu tố không thể thiếu trong bất kỳ một mạng lưới nào. ứng dụng của WSN trong thực tế từ đó rút ra một số yếu tố ảnh hưởng và cách giải quyết trong quá trình thiết kế và triển khai WSN.

### 1.2 Nút cảm biến không dây.

WSN bao gồm các hệ thống nhúng có khả năng: tương tác với môi trường thông qua các cảm biến khác nhau, xử lý thông tin tại chỗ, giao tiếp không dây giữa các nút. Một nút cảm biến là tập hợp của phần cứng, phần mềm và chuẩn.



### 1.2.1 Phần cứng của nút cảm biến không dây.

*Mô-đun không dây:* (hay còn gọi là motes) Motes là thành phần chính của nút cảm biến. Có khả năng thu phát vô tuyến.

Một motes bao gồm:

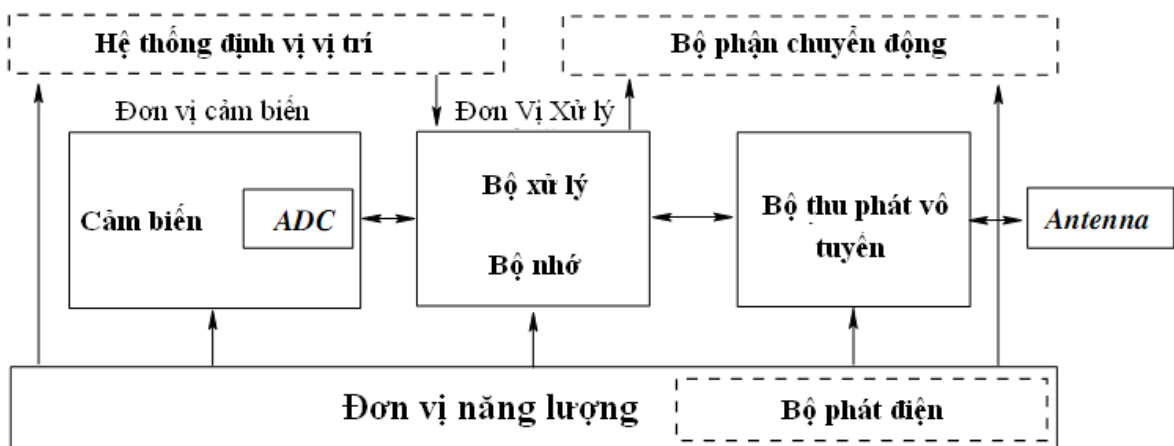
- ✓ Một vi điều khiển.
- ✓ Bộ phận thu phát.
- ✓ Nguồn điện (thường là PIN).
- ✓ Đơn vị bộ nhớ.
- ✓ Đầu đo cảm biến.

Ngoài ra còn một số thành phần như:

*Bảng mạch cảm biến* (sensor board): được gắn trên motes, chứa một vùng thử cho khách hàng nối các loại đầu đo cảm biến khác nhau của họ vào motes.

*Bảng mạch lập trình* (programming board): còn được gọi là bảng công, cung cấp giao tiếp Ethernet, Wi-Fi, USB. Với mục đích thu thập thông tin, nhúng chương trình và tải các ứng dụng cho motes.

Sơ đồ cấu tạo nút cảm biến: Cụ thể hơn, nút cảm biến không dây gồm bốn thành phần chính: đơn vị cảm biến, đơn vị xử lý, đơn vị thu phát và đơn vị năng lượng. thể hiện trong sơ đồ (hình 1.1).



**Hình 1.1.** Sơ đồ cấu tạo chung của một nút cảm biến không dây.

*Đơn vị cảm biến:* thành phần chính trong nút, gồm một số cảm biến, một bộ truyền đổi tín hiệu tương tự sang tín hiệu số (ADC). Nhiệm vụ, thu thập thông tin môi trường, chuyển thành tín hiệu số rồi gửi về đơn vị xử lý.

---

---

*Đơn vị xử lý:* gồm vi xử lý và một bộ nhớ trên bo mạch. Nhiệm vụ, quản lý hoạt động của nút, chạy các thuật toán liên quan, phối hợp với các nút khác thông qua mạng tuyến thông không dây.

*Bộ thu phát vô tuyến:* chuyển đổi các bit thông tin để truyền thông qua một tần số vô tuyến (RF) và phục hồi ở đầu kia.

*Đơn vị năng lượng:* thành phần quan trọng nhất quyết định tuổi thọ của WSN, gồm pin hoặc có thể thêm bộ phát điện (tùy ứng dụng). Nhiệm vụ, điều phối năng lượng tiêu thụ của các thành phần khác kéo dài.

*Hệ thống định vị vị trí:* gồm một mô-đun tính toán phân tán giúp xác định vị trí hoặc một GPS (nút cao cấp). Nhiệm vụ, xác định vị trí hiện tại của nút nhằm cung cấp các thông tin chính xác cho mạng.

*Bộ phận chuyển động:* Giúp nút di chuyển dưới sự chỉ đạo của bộ vi xử lý.

*Máy phát điện:* gồm một thành phần thu năng lượng mặt trời hoặc thu năng lượng nhờ rung động. Nhiệm vụ, cung cấp phụ trợ cho pin trong các ứng dụng đòi hỏi thời gian hoạt động lâu.

*Đặc điểm:* kích cỡ bằng một bao diêm hoặc rất nhỏ tùy vào từng ứng dụng cụ thể, trọng lượng nhẹ đến rất nhẹ (có thể bay lơ lửng trong không khí). Do đó Tiêu thụ rất ít năng lượng, hoạt động ở mật độ cao, chi phí sản xuất thấp, tự trị, hoạt động không cần giám sát và thích nghi với môi trường tốt.

### **1.2.2 Chuẩn cho nút cảm biến không dây.**

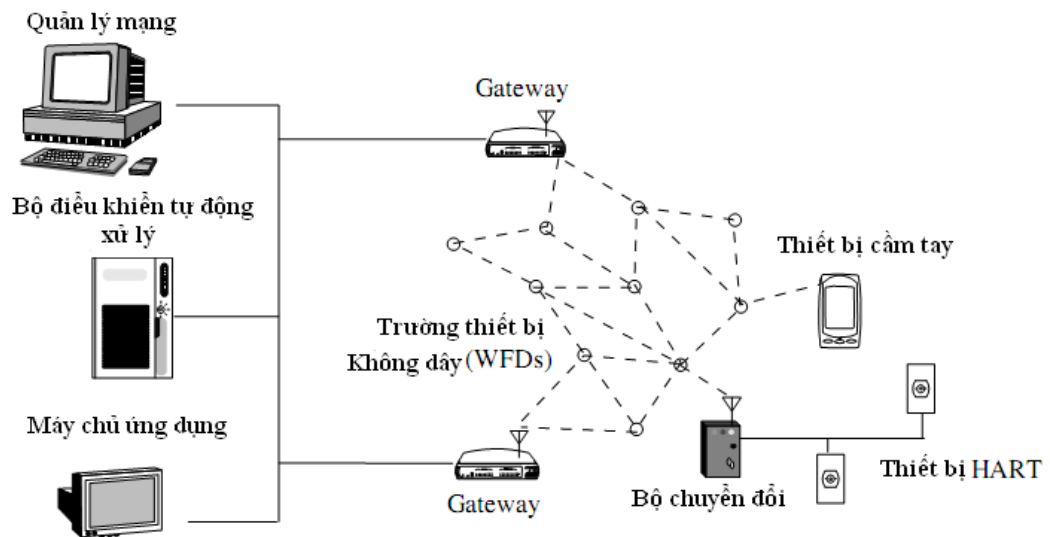
*Mục đích:* do sự không đồng nhất của các loại nút cảm dẫn đến không tương thích giữa các mạng và các ứng dụng khác nhau.

*Phân loại chuẩn:*

*IEEE 802.15.4:* Truyền thông với 3 băng tần: toàn cầu (2.4GHz), Châu Mỹ (915MHz) và Châu Âu (868MHz). Tầng Vật lý, sử dụng phương án điều chế binary phase shift keying (BPSK) trong dải tần 868/915MHz và offset quadrature phase shift keying (O-QPSK) trong băng tần 2.4GHz. Tầng MAC cung cấp cấu trúc liên kết dạng sao, lưới hoặc cây. Phạm vi truyền của các nút trong 10 đến 100m với tốc độ dữ liệu là 20-250 kbps.

*ZigBee:* nổi bật ở việc gán địa chỉ cho thiết bị khi nó tham gia mạng (địa chỉ này ngắn hơn ID của thiết bị) do vậy mà nâng cao hiệu suất truyền thông. Sử dụng kiến trúc dạng cây cho định tuyến.

*WirelessHART*: nổi bật với việc tạo và xác minh liên tục nhiều đường dự phòng trong quá trình thiết lập mạng dẫn đến khi một đường truyền từ thiết bị đến cổng bị hỏng thì sẽ được thay thế ngay, hỗ trợ quản lý băng thông động cho thiết bị bằng cách chỉ định con số thích hợp của các khe đến các thiết bị. Kiến trúc mạng: gồm 5 phần như trong hình 1.2.



**Hình 1.2.** Thành phần kiến trúc dạng chuẩn WirelessHART

*6LoWPAN*: nhằm tích hợp WSN với Internet thông qua IPv6, khi tích hợp IPv6 với WSN, do kích thước của Ipv6 khá lớn (40byte). Một số giải pháp đưa ra đó là thay vì một tiêu đề đơn khối duy nhất, bốn loại tiêu đề được sử dụng theo từng kiểu gói tin được gửi đi kết hợp sử dụng kĩ thuật nén stateless để giảm kích thước.

### 1.2.3 Phần mềm cho nút cảm biến không dây.

Ngoài phần cứng và chuẩn, phần mềm cũng được phát triển riêng cho WSN. Trong số này tiêu biểu nhất là TinyOS.

*Đặc điểm*: là hệ điều hành mã nguồn mở cho WSN. Có kích cỡ mã nhỏ, linh hoạt trong giao thức truyền thông. Hoạt động dựa trên xử lý hướng sự kiện.

*Tiện ích*: TinyOS mote simulator đơn giản hóa việc phát triển các giao thức mạng cảm biến và các ứng dụng, tạo môi trường mô phỏng rộng và biên dịch trực tiếp từ mã TinyOS. Cung cấp một công cụ giao diện đồ họa người dùng (TinyViz) giúp hình dung và tương tác với các hoạt động mô phỏng.

Ngoài ra, còn có một số hệ điều hành khác như:

- LiteOS: là một hệ điều hành đa luồng như Unix.
- Contiki: là hệ điều hành đa nhiệm (nguồn mở)
- Squawk: sử dụng một máy ảo JAVA, thực hiện trực tiếp ở bộ nhớ flash.

#### 1.2.4 Một số loại nút cảm biến không dây.

Hiện nay, WSN đang phát triển rất mạnh mẽ trong các lĩnh vực từ quan trọng như quân sự, công nghiệp hay y tế cho đến sử dụng tại gia đình. Do đó có rất nhiều loại nút cảm biến không dây khác nhau ra đời và có thể chia làm hai loại: loại nút thể hệ thấp và loại nút cao cấp.

Nút cảm biến thể hệ thấp: đặc trưng bởi khả năng xử lý, bộ nhớ và khả năng truyền thông thấp. Tuy nhiên, được sử dụng phổ biến trong các ứng dụng WSN. Một số loại nút như: Họ MICA (Mica, Mica2, MicaZ), IRIS, Telos / Tmote, EYES.

Đặc điểm chung: sử dụng băng tần (ISM). Ngoài ra gần đây được bổ sung các máy thu phát CC2420 trong băng tần 2.4 GHz và tương thích IEEE 802.15.4.

Nút cảm biến cao cấp: đặc trưng bởi khả năng xử lý cục bộ cao hơn, bộ nhớ nhiều hơn. Áp dụng cho những nút trung tâm xử lý - lưu trữ. Một số loại nút như: Stargate, Stargate NetBridge, Imote, Imote2. Bảng 1.1, tổng hợp cấu hình một số loại nút cảm biến hiện nay.

**Bảng 1.1.** Cấu hình một số loại nút cảm biến không dây.

Loại nút	Tốc độ CPU (MHz)	Bộ nhớ (kB)	RAM (kB)	Tần số (MHz)	Tốc độ truyền (kbps)
<i>Berkeley</i>					
WeC	8	8	0.5	916	10
rene	8	8	0.5	916	10
rene2	8	16	1	916	10
dot	8	16	1	916	10
mica	6	128	4	868	10/40
mica2	16	128	4	433/868/916	38.4kbaud
micaz	16	128	4	2.4GHz	250
Cricket	16	128	4	433	38.4kbaud
EyesIFX	8	60	2	868	115
TelosB/Tmote	16	48	10	2.4 GHz	250

SHIMMER	8	48	10	BT/2.4 GHz <sup>a</sup>	250
Sun SPOT	16–60	2 MB	256	2.4 GHz	250
BTnode	8	128	64	BT/433–915 <sup>a</sup>	Varies
IRIS	16	128	8	2.4 GHz	250
V-Link	N/A	N/A	N/A	2.4 GHz	250
TEHU-1121	N/A	N/A	N/A	0.9/2.4 GHz	N/A
NI WSN-3202	N/A	N/A	N/A	2.4 GHz	250
Imote	12	512	64	2.4 GHz (BT)	100
Imote2	13–416	32 MB	256	2.4 GHz	250
Stargate	400	32 MB	64 MB SD	2.4 GHz	Varies <sup>b</sup>
Netbridge NB-100	266	8 MB	32 MB	Varies <sup>b</sup>	Varies <sup>b</sup>

(a) Trang bị hai bộ thu phát: Bluetooth và một radio năng lượng thấp.

(b) Phụ thuộc vào các thiết bị truyền thông được kết nối.

### 1.3 Kiến trúc và giao thức.

WSN hình thành bởi tập hợp rất lớn các nút cảm biến, và hoạt động trên sự kết hợp của những nút này. Do đặc điểm của ứng dụng mà các nút trong mạng phân tán trong một vùng tương đối rộng. Trong mục này, làm rõ kiến trúc và giao thức sử dụng trong WSN.

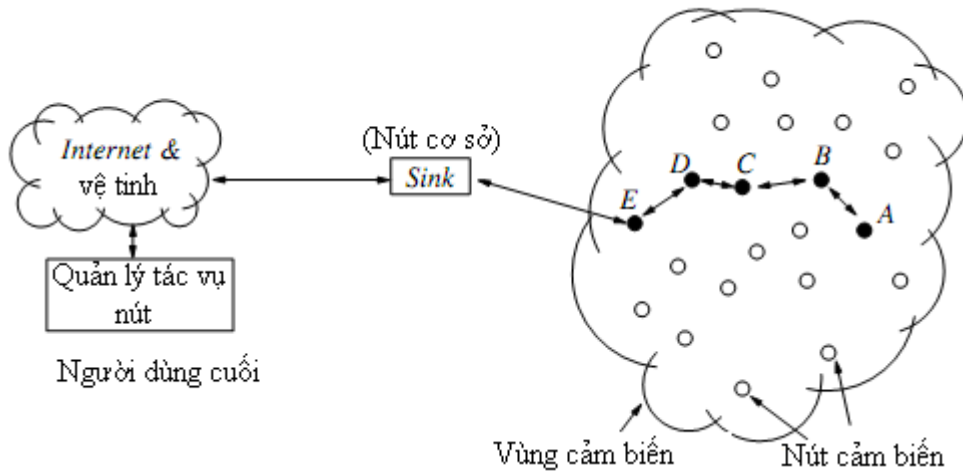
#### 1.3.1 Kiến trúc mạng của WSN.

Các nút cảm biến được phân tán trong một vùng cảm biến như hình 1.3, mỗi nút đều có khả năng thu thập và truyền dữ liệu về nút cơ sở (hay còn gọi là sink) và đến người dùng cuối. Dữ liệu được truyền thông qua kiến trúc multi-hop.

Sink có thể giao tiếp với tác vụ quản lý/người dùng cuối thông qua Internet, vệ tinh hoặc bất kỳ loại mạng Wifi, mesh networks, cellular systems, WiMAX. Có thể có nhiều sink/cổng và người dùng cuối.

Các nút thành viên: (với 2 chức năng)

- ✓ chức năng nguồn dữ liệu: nút thực hiện truyền tải gói dữ liệu về sink.
- ✓ chức năng định tuyến: tham gia vào chuyển tiếp gói tin nhận được từ các nút khác đến nút tiếp theo trong đường dẫn multi-hop để tới sink.

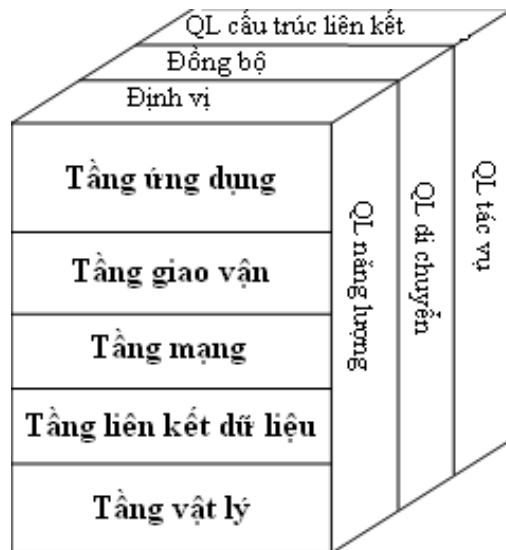


**Hình 1.3.** Kiến trúc mạng WSN.

### 1.3.2 Giao thức Stack.

Được sử dụng ở sink và tất cả các nút cảm biến nhằm kết hợp: năng lượng với định tuyến, tích hợp dữ liệu với giao thức mạng, truyền thông hiệu quả với phương tiện không dây và giao tiếp giữa các nút.

Thành phần của giao thức: (thể hiện trong hình 1.4)



**Hình 1.4.** Giao thức Stack của WSN.

✓ *Tầng vật lý:* giải quyết nhu cầu về kỹ thuật điều chế, truyền và nhận.

- 
- 
- ✓ *Tầng liên kết dữ liệu*: bảo đảm thông tin liên lạc đáng tin cậy, sử dụng kỹ thuật kiểm soát lỗi sẽ được nêu ra trong chương 2 và quản lý truy cập kênh giảm thiểu va chạm.
  - ✓ *Tầng mạng*: quan tâm đến việc định tuyến gói dữ liệu.
  - ✓ *Tầng vận chuyển*: duy trì lưu thông dữ liệu trong mạng với UDP.
  - ✓ *Tầng ứng dụng*: quản lý các ứng dụng, quản lý mạng, xử lý truy vấn, chứa các mã ứng dụng.

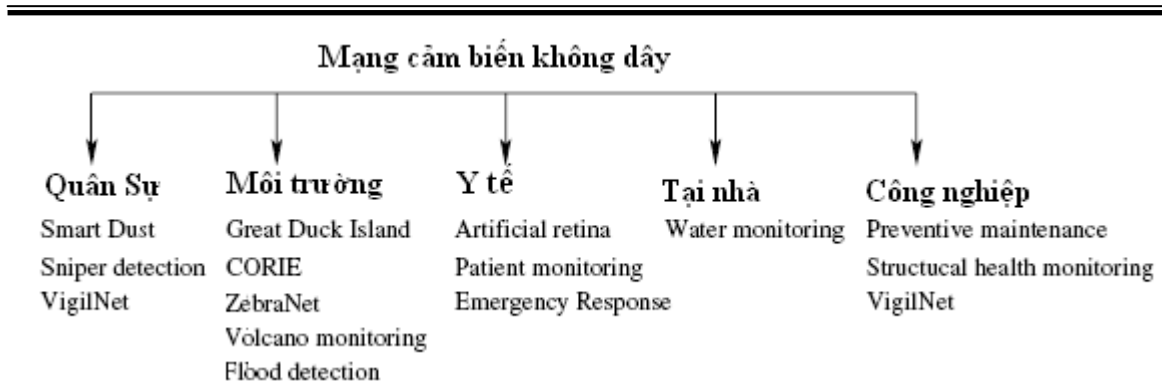
Các chức năng quản lý, định vị, đồng bộ đều nhằm mục tiêu: gắn kết các nút cảm biến trong mạng lại với nhau, sử dụng năng lượng hiệu quả, chia sẻ tài nguyên mạng và giúp mở rộng mạng cảm biến.

Giải pháp lớp chéo trong WSN: nhằm tích hợp chặt chẽ các giao thức trong lớp stack. Bằng cách loại bỏ giới hạn giữa các lớp cũng như các giao diện liên quan giữa chúng, nhằm tăng hiệu quả trong không gian và điều khiển. Nó nêu lên:

- Mỗi nút được trang bị một thiết bị đo xung nhịp cục bộ.
- Cảm biến ,xử lý, truyền thông được liên kết, kiểm soát bởi xung nhịp cục bộ.
- Thời gian truyền thông dữ liệu ở từng nút được nhất quán.
- Thông tin thu phát phải kết hợp với vị trí của nút tạo ra.
- Giao thức truyền thông phải có thông tin vị trí.

#### **1.4 Ứng dụng của WSN.**

Các ứng dụng của WSN ngày càng phát triển bởi những khả năng đặc biệt của nó, kích thước các nút mạng rất nhỏ có thể triển khai trong bất kì môi trường hay địa hình phức tạp nào, hoạt động tự do không cần nguồn cấp cố định do được tích hợp sẵn nguồn nuôi, tự xử lý cục bộ giúp cung cấp thông tin và thao tác nhanh chính xác với các hiện tượng mà nó theo dõi, đáp ứng được những yêu cầu phức tạp của các ứng dụng nhờ khả năng tích hợp được rất nhiều loại cảm biến như nhiệt độ, ánh sáng, gia tốc, âm thanh,v.v. Một số lĩnh vực đã triển khai ứng dụng của mạng cảm biến không dây: quân sự, y tế, công nghiệp, môi trường và tại nhà. Một số ứng dụng được thể hiện trong hình 1.5.



**Hình 1.5.** Mô hình một số dự án ứng dụng của WSN.

Trong quân sự: nhằm mục tiêu tiếp cận chiến trường tốt hơn giảm thiệt hại cho con người và trang thiết bị. Một số ứng dụng của nó như: theo dõi các lực lượng thân thiện, thiết bị và đạn dược, giám sát chiến trường, trinh sát địa hình của lực lượng thù địch, phát hiện và tấn công trinh sát, mục tiêu, đánh giá thiệt hại trận chiến do vũ khí sinh học, hóa chất, hạt nhân.

Trong y tế: nhằm mục đích hỗ trợ người bệnh, bệnh viện hay nghiên cứu y sinh như cung cấp giao diện cho người tàn tật, theo dõi tổng hợp bệnh nhân, chẩn đoán, quản lý thuốc tại các bệnh viện, giám sát các dữ liệu sinh lý con người, theo dõi chuyển động của côn trùng hay vi sinh vật.

Trong công nghiệp: nhằm tăng khả năng tự động hóa trong công việc như quản lý hàng tồn kho, giám sát chất lượng sản phẩm, giám sát giao thông vận tải, kiểm soát quy trình sản xuất và tự động hóa, phát hiện và theo dõi hành vi trộm cắp xe; theo dõi và phát hiện xe, văn phòng thông minh, điều khiển robot, hướng dẫn trong các môi trường sản xuất tự động, bảo tàng tương tác.

Trong môi trường: theo dõi chuyển động của các loài chim, động vật nhỏ, côn trùng, theo dõi môi trường ảnh hưởng đến cây trồng, chăn nuôi, thủy lợi, theo dõi tình trạng trái đất, thăm dò các hành tinh, giám sát môi trường đất, biển, và khí quyển, phát hiện cháy rừng, khí tượng hoặc nghiên cứu địa vật lý, phát hiện lũ, lập bản đồ môi trường và nghiên cứu ô nhiễm.

Tại nhà: Các nút cảm biến không dây thông minh có thể gắn bên trong các thiết bị như máy hút bụi, lò vi sóng, tủ lạnh, và đầu DVD cũng như hệ thống nước kết nối với mạng bên ngoài thông qua Internet hoặc vệ tinh. Cho phép người dùng dễ dàng hơn trong quản lý các thiết bị trong nhà từ xa.



---

---

## 1.5 Các yếu tố ảnh hưởng đến thiết kế.

Như phần trước đã đề cập, WSN hiện nay đang phát triển khá mạnh mẽ trên mọi lĩnh vực. Nhưng chính vì thế mà những nhà phát triển ứng dụng của mạng này lại phải đối mặt với những thách thức không hề nhỏ. Phần này, đi sâu vào những yếu tố ảnh hưởng tạo nên những thách thức trong thiết kế WSN. Các yếu tố đó là: hạn chế phần cứng, khả năng chịu lỗi, khả năng mở rộng, chi phí sản xuất, cấu trúc liên kết bộ cảm biến mạng, truyền thông và tiêu thụ điện năng. Trong đó, yếu tố tiêu thụ năng lượng quyết định đến sự sống còn của WSN.

### 1.5.1 Hạn chế phần cứng.

Thách thức:

- Kích cỡ quá nhỏ mà yêu cầu ứng dụng cao.
- Nguồn năng lượng pin quá ít.
- Điện năng nuôi ăng-ten và bộ thu phát là rất lớn.
- Truyền thông qua tần số vô tuyến khả năng mất mát, hỏng thông tin.
- Bộ nhớ và bộ xử lý cấu hình vẫn còn thấp.

Giải pháp: bổ sung máy phát điện cho mỗi nút trong mạng, thiết kế bộ thu phát với khả năng bật tắt lúc cần thiết, sử dụng thiết bị truyền thông cao cấp hơn, sử dụng phần mềm dung lượng thấp và thuật toán xử lý tối ưu nhất.

### 1.5.2 Khả năng chịu lỗi.

Yếu tố lỗi mạng sinh ra do những hạn chế của phần cứng và môi trường ứng dụng. Một số lý do: thiếu điện, hỏng hóc do tác động của môi trường và lỗi phần mềm. Quan trọng nhất là do các nút được cung cấp nguồn nuôi chủ yếu là pin dẫn đến năng lượng không đủ làm các nút ngưng hoạt động. Khả năng chịu lỗi hiệu là khả năng mạng vẫn hoạt động bình thường ngay khi có một vài lỗi nào đó xảy ra.

Thách thức: do pin với năng lượng ít, vi xử lý với khả năng chưa cao, bộ nhớ thấp dẫn đến lỗi khi chạy phần mềm và làm cho nút ngưng hoạt động. Ngoài còn tác động của nhiều loại môi trường.

Giải pháp: xây dựng mạng với mật độ các nút cao. Tức là nhiều nút trong một phạm vi phát sóng. Một nút bị hỏng thì có nút khác phát thay thế ngay.

---

---

### 1.5.3 Khả năng mở rộng.

Do nâng cao khả năng chịu lỗi của mạng, giải pháp tăng khả năng chịu lỗi được đưa ra ở trên lại tạo ra một thách thức khác cho thiết kế WSN là với mật độ nút cao trong một phạm vi dẫn đến khi mở rộng mạng cần một lượng rất lớn các nút cảm biến (hàng trăm đến hàng nghìn nút). chính vấn đề này đã tạo ra một thách thức nan giải cho những nhà thiết kế, đó là làm sao để kiểm soát và điều khiển một lượng lớn các nút này hoạt động cách hiệu quả.

### 1.5.4 Chi phí sản xuất.

WSN là tập hợp của một lượng lớn các nút cảm biến, do vậy chi phí để tạo ra một nút mạng là rất quan trọng để đưa ra giá cả tổng thể cho toàn mạng. nếu chi phí này đắt hơn chi phí cho những thiết bị cảm biến truyền thông thì nó sẽ không được sử dụng. Hiện nay giá cả của một nút cảm biến vẫn khá cao (hơn 10\$) trong khi yêu cầu mạng sẽ chỉ triển khai thực tế được với giá một nút là 1\$. Do đó đây cũng là một thách thức khó khăn cho những nhà thiết kế WSN.

### 1.5.5 Cấu trúc liên kết.

Số lượng lớn các nút cảm biến không thể truy cập, giám sát và thường xuyên bị lỗi làm việc duy trì cấu trúc liên kết là một công việc đầy thử thách. Quá trình nghiên cứu cấu trúc liên kết của WSN thông qua ba giai đoạn: giai đoạn trước và trong triển khai, giai đoạn hậu triển khai và giai đoạn tái triển khai.

Trước và trong triển khai: phân bố hàng loạt bằng máy bay, tên lửa, hoặc đặt từng nút một bằng người hoặc robot. Cần một kế hoạch thiết kế cẩn thận nhằm : giảm chi phí lắp đặt ban đầu, tăng sự linh hoạt, tự tổ chức và chịu lỗi của mạng.

Giai đoạn hậu triển khai: cấu trúc liên kết mạng thay đổi do nhiều nguyên nhân, do đó cần nhưng giao thức mạng linh động có khả năng thay đổi ngắn hạn, định kỳ, và dài hạn trong các cấu trúc liên kết.

Giai đoạn tái triển khai: bổ sung, thay thế các nút hỏng, thay đổi hoạt động theo yêu cầu ứng dụng.

### 1.5.6 Phương tiện truyền thông.

Hoạt động thành công của WSN phụ thuộc vào sự truyền thông tin cậy giữa các nút mạng. Chúng có thể truyền thông với nhau thông qua: radio, hồng ngoại, quang học, âm thanh hoặc cảm ứng từ. Nhưng để thành một mạng lưới rộng khắp nó cần một phương tiện có sẵn cho cả thế giới và băng tần ISM được chọn.

Đặc điểm: ISM được cấp giấy phép miễn phí, có sẵn ở khắp thế giới, không ràng buộc tiêu chuẩn cụ thể, tự do sử dụng các giao thức. Theo đó băng tần hiện nay sử dụng là ISM 2.4GHz trên toàn thế giới, ngoài ra băng tần ISM 433MHz ở châu Âu và băng tần ISM 915MHz ở Bắc Mỹ. Tần ISM trong bảng 1.2.

**Bảng 1.2.** Tần số ISM.

<b>Khoảng tần số</b>	<b>Tần số trung tâm</b>
6765–6795 kHz	6780 kHz
13 553–13 567 kHz	13 560 kHz
26 957–27 283 kHz	27 120 kHz
40.66–40.70 MHz	40.68 MHz
433.05–434.79 MHz	433.92 MHz
902–928 MHz	915 MHz
2400–2500 MHz	2450 MHz
5725–5875 MHz	5800 MHz
24–24.25 GHz	24.125 GHz
61–61.5 GHz	61.25 GHz
122–123 GHz	122.5 GHz
244–246 GHz	245 GHz

Hạn chế: có thể nhiễu tín hiệu từ các ứng dụng khác, do không được quy định dùng riêng cho WSN nên các mạng không dây khác cũng có can thiệp vào nó.

Ngoài ra, Hồng ngoại cũng được sử dụng do được cấp miễn phí, chi phí rẻ và xây dựng dễ dàng. Tuy nhiên khoảng cách truyền thông ngắn thích hợp với các ứng dụng trong môi trường khắc nghiệt. Âm thanh, sử dụng trong ứng dụng hàng hải.

### **1.5.7 Năng lượng tiêu thụ.**

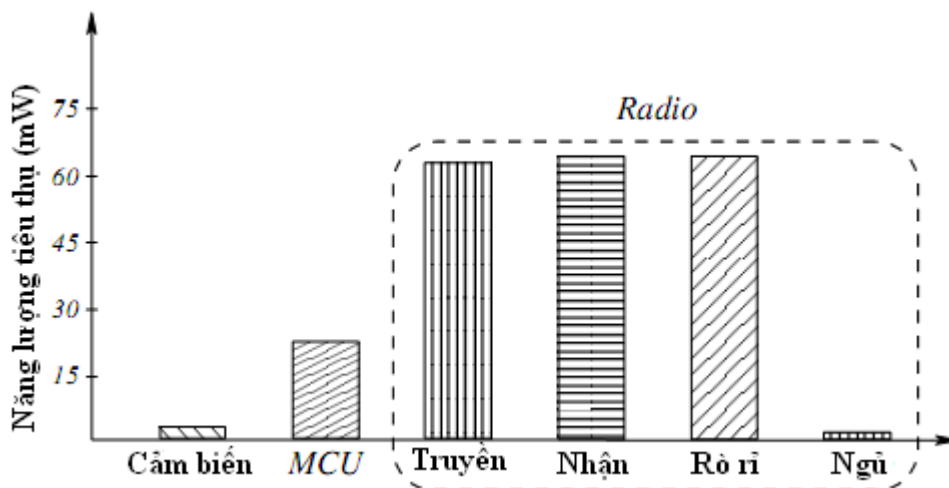
Trong WSN, năng lượng tiêu thụ quyết định đến sự sống còn của mạng. Trong mục này, giới thiệu công thức tính toán để xác định mức năng lượng tiêu thụ trong mạng WSN. Một nút cảm biến không dây thông thường chỉ được trang bị nguồn nuôi là Pin (0.5Ah <, 1.2V), mặt khác với hầu hết các ứng dụng của WSN thì việc bổ sung nguồn nuôi là không thể thực hiện được. Vì vậy, cần phân tích hoạt động tiêu thụ năng lượng của các nút trong WSN để sử dụng hiệu quả nó hiệu quả.

Các hoạt động của một nút cảm biến:

- Khởi tạo dữ liệu: nút thu thập thông tin, xử lý chúng tạo dữ liệu khởi đầu.
- Định tuyến dữ liệu: các nút chuyển dữ liệu khởi đầu đến các nút lân cận để đưa đến sink (theo quy định định tuyến).

Công việc tiêu thụ năng lượng: (rút ra từ hai hoạt động trên)

- ✓ Cảm biến.
- ✓ Truyền thông. (tiêu tốn năng lượng nhất thể hiện trong hình 1.6)
- ✓ Xử lý dữ liệu.



**Hình 1.6.** Mức độ tiêu thụ năng lượng của nút cảm biến MicaZ.

### 1.5.7.1 Năng lượng tiêu thụ cho cảm biến.

Mức độ tiêu thụ năng lượng cho cảm biến tùy thuộc ứng dụng cụ thể như: một ứng dụng thu thập thông tin từng đợt mất ít năng lượng hơn thu thập thường xuyên, sự kiện cảm biến phức tạp sẽ tốn năng lượng nhiều hơn sự kiện đơn giản hay nhiều nhiều sẽ làm nút tiêu thụ năng lượng nhiều hơn. Nhưng nhìn chung, trong mỗi đơn vị cảm biến có một hệ thống con gồm: một thiết bị khuếch đại tạp âm thấp, một bộ lọc khử răng cưa, một ADC và một bộ xử lý tín hiệu số (DSP).

Có thể tính được năng lượng tiêu thụ do cảm biến thông qua công thức 1.1.

$$P \propto F_s \cdot 2^{ENOB} \quad (1.1)$$

Trong đó,  $F_s$  là tỷ lệ lấy mẫu, ENOB là số bit hiệu quả (hay độ phân giải).

Tính chính xác của dữ liệu quyết định bởi tỷ lệ lấy mẫu. Tăng tỷ lệ lấy mẫu cũng sẽ cung cấp độ phân giải tốt hơn.

*Ví dụ:* với một ứng dụng cảm biến nhiệt độ:

Tỷ lệ lấy mẫu = 1ms không phù hợp khi nhiệt độ thay đổi theo phút hay giờ. Độ phân giải tăng lên 8bit - 10bit, ADC phải tinh vi hơn và gia tăng hoạt động cho một đơn cảm biến. Theo đó, năng lượng tiêu thụ tăng lên.

### 1.5.7.2 Năng lượng tiêu thụ cho xử lý.

Theo tính toán, chi phí năng lượng của truyền một gói 1 (KB) trong khoảng cách 100 (m) là xấp xỉ bằng thực hiện 3.000.000 (chỉ thị) bởi một vi xử lý. Do đó, việc xử lý tại chỗ được áp dụng trong WSN. Với vi xử lý với công nghệ bán dẫn (CMOS) rất thích hợp với các nút cảm biến yêu cầu nhỏ và rẻ của WSN. Công thức 1.2 (xác định mức năng lượng tiêu thụ cho xử lý  $E_p$ ).

$$E_p = N \cdot C \cdot V_{dd}^2 + V_{dd}(I_0 e^{V_{dd}/n \cdot V_T})(N/f) \quad (1.2)$$

Trong đó,  $N \cdot C \cdot V_{dd}^2$ : năng lượng tiêu hao của thiết bị chuyển mạch bán dẫn. (N: số xung nhịp đồng hồ, C: tổng điện dung chuyển đổi,  $V_{dd}$ : điện áp cung cấp),

$V_{dd}(I_0 e^{V_{dd}/n \cdot V_T})(N/f)$ : năng lượng tiêu hao do rò rỉ với mặt đất ( $I_0$ : sự rò rỉ hiện thời, n: là hằng số liên quan tới phần cứng vi xử lý,  $V_T$ : ngưỡng điện áp,  $f$ : tần số xung nhịp). Từ đó, ta có thể kiểm soát được  $V_{dd}$  và  $f$ .

Thành phần  $f$  có thể được tính toán qua độ trễ cổng  $T_g$  (gate delay) trong công thức 1.3. Độ trễ cổng, phụ thuộc vào điện áp cung cấp, giảm điện áp cung cấp làm tăng độ trễ cổng có thể giảm thời gian nhàn rỗi cho bộ vi xử lý.

$$T_g = \frac{V_{dd}}{K(V_{dd} - V_{th})^a} \quad (1.3)$$

(K và a: biến phụ thuộc vào bộ vi xử lý ( $a \sim 2$ ),  $V_{th}$ : điện áp tối thiểu)

Nếu bộ vi xử lý hoạt động ở một ( $f$ ) tương ứng với thời gian chuyển tiếp cổng  $T_0=1/f$  trong đó bộ vi xử lý có nhiệm vụ duy nhất là xử lý. Khi  $T_g < T_0$ , bộ xử lý được nhàn rỗi khi hoàn tất một công việc cho đến khi công việc tiếp theo. Từ đó,  $f$  được tính bằng công thức 1.4.

$$f \leq \frac{K(V_{dd} - V_{th})^a}{V_{dd}} \quad (1.4)$$

Giải pháp tiết kiệm năng lượng: từ những công thức trên.

- Giảm  $V_{th}$  có thể giảm được  $V_{dd}$  mà không ảnh hưởng đến xử lý.
- Giảm  $f$  trong suốt thời gian hoạt động, chỉ tổn thất điện năng do dòng rò rỉ.
- Giảm  $V_{dd}$  làm giảm sự rò rỉ tối đa.
- Tìm mức năng lượng tiêu thụ cao nhất để kiểm soát điện áp hoạt động.

### 1.5.7.2 Năng lượng tiêu thụ cho truyền thông.

Một nút cảm biến không dây bỏ ra phần lớn năng lượng cho truyền thông. Theo phân tích, năng lượng mất đi nhiều nhất cho quá trình thu, phát và rò rỉ, mất khác ở trạng thái ngủ nó lại tiết kiệm đến 99.99%. Thành phần tiêu thụ năng lượng chính trong truyền thông là bộ thu phát vô tuyến do đó cần xem xét thành phần này.

Bộ thu phát: gồm bộ trộn, bộ tổng hợp tần số, bộ điều khiển điện áp dao động (VCO), vòng khóa pha (PLL), bộ giải điều chế và bộ khuếch đại năng lượng.

Mô hình đơn giản: năng lượng tiêu thụ trong truyền thông ( $P_c$ ), là cung cấp cho truyền và nhận thông tin, dữ liệu. thể hiện trong công thức 1.5 dưới đây.

$$P_c = P_o + P_{tx} + P_{rx} \quad (1.5)$$

( $P_o$ : năng lượng đầu ra bộ khuếch đại,  $P_{rx}$  và  $P_{tx}$ : năng lượng cho thu và phát).

Theo công thức thì:

- $P_o + P_{tx}$  : là năng lượng tiêu thụ của nút phát.
- $P_{rx}$  : là năng lượng tiêu thụ của nút thu.

Mô hình tổng thể: ngoài việc thu và phát, nút cảm biến còn tiêu thụ năng lượng trong chuyển trạng thái thức-ngủ và khởi động phần cứng. đồng thời cung cấp năng lượng cho thiết bị chuyển mạch thực hiện. Thể hiện ở công thức 1.6.

$$E_c = E_{st} + E_{rx} + E_{sw} + E_{tx} \quad (1.6)$$

Trong đó,  $E_c$ : năng lượng tiêu thụ tổng thể;  $E_{st}$ : năng lượng tiêu thụ cho khởi động;  $E_{sw}$ : năng lượng tiêu thụ cho thiết bị chuyển mạch;  $E_{rx}$  và  $E_{tx}$ : năng lượng tiêu thụ cho thu và phát.

Năng lượng tiêu thụ cho khởi động:  $E_{st} = P_{LO} \cdot t_{st} \quad (1.7)$

Trong đó,  $P_{LO}$ : năng lượng tiêu thụ của bộ tổng hợp tần số và VCO;  $t_{st}$ : thời gian cần để khởi động nút.

---

---

Năng lượng tiêu thụ cho thiết bị chuyển mạch:  $E_{sw} = P_{LO} \cdot t_{sw}$  (1.8)

Trong đó, ( $t_{sw}$ ): thời gian chuyển đổi.

Năng lượng tiêu thụ cho thu nhận thông tin:  $E_{rx} = (P_{LO} + P_{RX})t_{rx}$  (1.9)

Trong đó,  $P_{RX}$ : năng lượng tiêu thụ của bộ khuếch đại, bộ trộn, và bộ giải điều chế; ( $t_{rx}$ ): thời gian thu một gói tin.

Năng lượng tiêu thụ cho phát thông tin.  $E_{tx} = (P_{LO} + P_{PA})t_{tx}$  (1.10)

Trong đó,  $P_{PA}$ : năng lượng tiêu thụ cho bộ khuếch đại (không đáng kể) và bộ khuếch đại điện năng; ( $t_{tx}$ ): thời gian phát một gói tin.

Năng lượng tiêu thụ cho bộ khuếch đại:  $P_{PA} = \frac{1}{\eta} P_{out} = \frac{1}{\eta} \cdot \gamma_{PA} \cdot r \cdot d^n$  (1.11)

Trong đó,  $\eta$ : năng lượng hiệu quả của các bộ khuếch đại;  $P_{out}$ : năng lượng tạo ra từ bộ khuếch đại;  $\gamma_{PA}$ : là một yếu tố phụ thuộc vào ăng ten;  $d$ : khoảng cách truyền thông;  $r$ : tốc độ dữ liệu;  $n$ : số mũ trong công thức thể hiện sự mất mát.

Yếu tố phụ thuộc ăng ten: là Bước sóng, nhiệt độ làm nảy sinh tạp nhiễu cho thiết bị, tỷ lệ nhiễu tín hiệu (SNR).

Chu trình giao tiếp: khi mà một nút truyền một gói tin đến một nút lân cận và nhận một phản ứng trở lại, bao gồm khởi động của bộ thu phát vô tuyến, truyền tải gói dữ liệu, chuyển đổi từ chế độ phát sang chế độ thu, và tiếp nhận gói tin. Kết quả là, mức tiêu thụ năng lượng tổng thể theo công thức 1.12.

Thay các công thức 1.7, 1.8, 1.9, 1.10, 1.11, vào 1.6 ta được:

$$\begin{aligned} E_c &= E_{st} + E_{rx} + E_{sw} + E_{tx} \\ &= P_{LO} \cdot t_{st} + (P_{LO} + P_{RX})t_{rx} + P_{LO} \cdot t_{sw} + (P_{LO} + P_{PA})t_{tx}. \end{aligned} \quad (1.12)$$

Giả sử  $t_{RX} = t_{TX} = l_{PKT} / r$ , với  $l_{PKT}$  là độ dài gói tin, ta được mức tiêu thụ năng lượng tổng thể là:

$$E_c = P_{LO}(t_{st} + t_{sw}) + (2P_{LO} + P_{RX}) \frac{l_{PKT}}{r} + \frac{1}{\eta} \cdot \gamma_{PA} \cdot d^n t_{tx} \cdot l_{PKT} \quad (1.13)$$

Từ công thức 1.13, thấy rằng việc tiêu thụ năng lượng cho truyền thông có ba thành phần chính:  $P_{LO}(t_{st} + t_{sw})$ : không đổi và phụ thuộc vào các mạch thu phát.

$(2P_{LO} + P_{RX}) \frac{l_{PKT}}{r}$ : độc lập với khoảng cách truyền thông ( $d$ ) nhưng có thể được

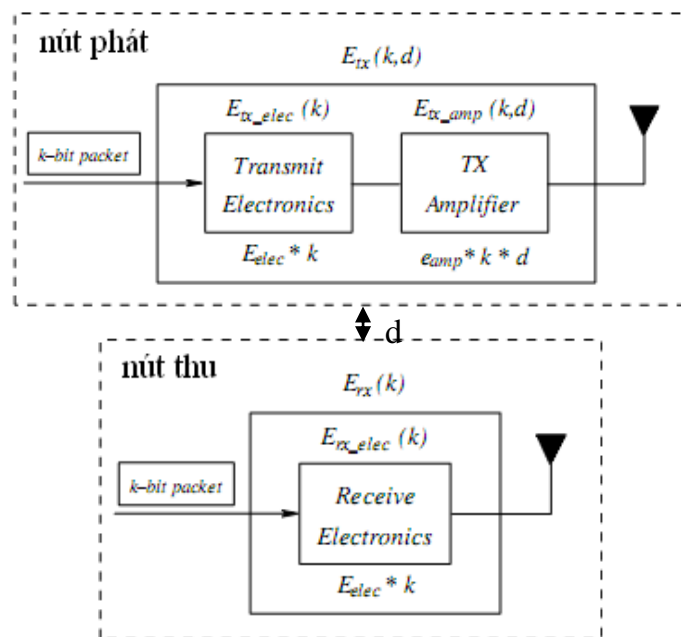
điều khiển thông qua các kích thước gói hoặc tốc độ truyền. (Hai thành phần này là các thành phần độc lập với năng lượng tiêu thụ)

$$\frac{1}{\eta} \cdot \gamma_{PA} \cdot d^n t_{tx} \cdot l_{PKT}$$

: phụ thuộc vào khoảng cách truyền thông cũng như chiều dài gói và có thể được điều khiển thông qua giao thức lớp cao hơn như là MAC và các giao thức định tuyến.

Tính toán đơn giản cho tiêu thụ năng lượng trong truyền thông:

Được đơn giản hóa cho một cặp nút thu-phát với khoảng cách truyền thông d. thể hiện trong hình 1.7.



**Hình 1.7.** Đơn giản hóa năng lượng tiêu thụ cho một cặp nút.

Theo mô hình, năng lượng tiêu thụ do phát phụ thuộc vào thiết bị truyền điện tử và bộ khuếch đại truyền. Năng lượng tiêu thụ do thu chỉ phụ thuộc vào thiết bị thu điện tử. Từ đó ta có thể mô hình hóa nó dưới dạng công thức 1.14.

$$E_c = E_{tx}(k, d) + E_{rx}(k) \quad (1.14)$$

Trong đó,  $E_{tx}(k, d)$  và  $E_{rx}(k)$ : năng lượng tiêu thụ của nút phát và nút thu.  $E_{tx}(k, d) = E_{tx-elec} \cdot k + e_{amp} \cdot k \cdot d^n$  và  $E_{rx}(k) = E_{rx-elec} \cdot k$ . Với  $E_{TX-elec}$  và  $E_{RX-elec}$  là năng lượng tiêu thụ trên mỗi bit cho nút phát và thu;  $e_{amp}$ : năng lượng tiêu thụ trên mỗi bit trên bộ khuếch đại điện năng (phụ thuộc d).



Thông qua những phân tích và công thức tính toán, có thể thấy năng lượng dành cho truyền thông trong WSN là rất lớn, trên thực tế việc truyền nhận dữ liệu không phải lúc nào cũng thành công. Hơn nữa, do đặc điểm phân cứng của WSN cũng gây ra lỗi dữ liệu truyền. Do đó một phương án kiểm soát lỗi trong mạng WSN là vô cùng quan trọng nó sẽ được trình bày trong chương 2 của đồ án.

---

---

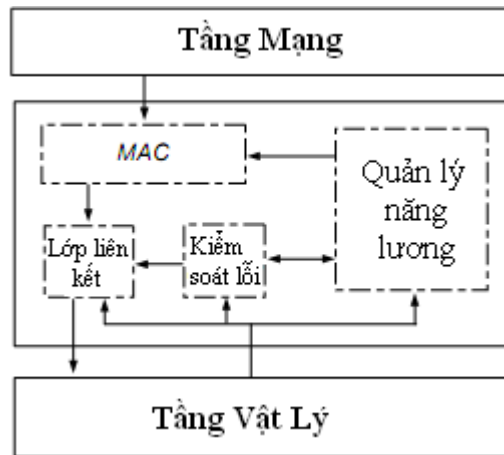
## CHƯƠNG 2: KIỂM SOÁT LỖI TRONG WSN

### 2.1 Tổng quan về vấn đề kiểm soát lỗi trong WSN.

WSN dựa vào sự kết hợp và phản hồi của nhiều cảm biến tương tác với môi trường. Do đó, lớp liên kết dữ liệu trong mạng thực sự rất quan trọng, với nhiệm vụ ghép kênh / giải mã kênh dữ liệu, phát khung dữ liệu, truy cập môi trường và kiểm soát lỗi. Theo đó, để có độ tin cậy và hiệu quả dữ liệu trong mạng công việc kiểm soát lỗi là không thể thiếu. Trái ngược với những mạng truyền thông, WSN được đặc trưng bởi năng lượng thấp và sự hợp tác chặt chẽ của các nút cảm biến. Do vậy, các giải pháp truyền thông thường được đưa ra đều nhằm mục đích để làm giảm sự dư thừa trong lưu thông bằng cách lọc dữ liệu liên quan hoặc tắt các nút dư thừa.

Mục đích chính của kiểm soát lỗi là cung cấp thông tin đáng tin cậy trong các kênh không dây bởi lẽ những kênh này rất hay gặp sự cố như: nhiễu, yếu hay mất bit đồng bộ. Điều này dẫn đến lỗi kênh và tác động đến sự toàn vẹn của các gói tin được gửi đi bởi các nút cảm biến. Các lỗi này còn gia tăng hơn với WSN vì truyền thông điện năng thấp. Kết quả là, ngoài các kỹ thuật ở lớp vật lý cung cấp sự tin cậy ở cấp độ bit, các phương án kiểm soát lỗi được sử dụng tại các lớp liên kết cung cấp độ tin cậy ở mức gói dữ liệu.

Nhìn chung, các cơ chế kiểm soát lỗi trong WSN có thể được phân thành bốn phương pháp chính: kiểm soát năng lượng, tự động phát lại (ARQ), sửa lỗi trước truyền (FEC), và ARQ lai ghép (HARQ). Bốn phương án này sẽ được là rõ tại chương này, đồng thời trong chương phân tích các phương án kiểm soát lỗi bằng cách sử dụng kỹ thuật phân tích qua lớp chéo để để xem xét những tác động trên kênh không dây, truy cập môi trường, và định tuyến multi-hop. Dưới đây, tổng quan về các thành phần của lớp liên kết dữ liệu thông qua hình 2.1.



**Hình 2.1.** Tổng quan lớp liên kết dữ liệu.

## 2.2 Các phương án kiểm soát Lỗi trong WSN.

WSN đặc trưng bởi nguồn năng lượng thấp và khả năng liên kết chặt chẽ giữa các nút cảm biến trong mạng. Do đó, vấn đề đặt ra là làm sao để giữ cho mạng hoạt động ổn định và hiệu quả với nguồn năng lượng thấp đó. Các phương án đã được đưa ra: kiểm soát năng lượng và kiểm soát lỗi trong truyền thông dữ liệu.

### 2.2.1 Kiểm soát năng lượng.

Trong nghiên cứu và triển khai nhưng nhà nghiên cứu rút ra, khi kiểm soát nguồn năng lượng tốt thì lỗi mạng sẽ được giảm thiểu. Có 2 phương án kiểm soát nguồn năng lượng để giảm tỉ lệ lỗi.

*Thứ nhất*, Kiểm soát năng lượng tiêu thụ từ ăng-ten có thể giúp kiểm soát phạm vi truyền thông hiệu quả của một nút.

*Thứ hai*, Kiểm soát năng lượng truyền tải. Truyền tải năng lượng cao làm giảm tỷ lệ lỗi gói tin bởi vì nó cải thiện được tỷ lệ nhiễu tín hiệu. Tuy nhiên, năng lượng tiêu thụ vì thế mà tăng đồng thời có thể gia tăng nhiễu với các nút lân cận.

Kiểm soát năng lượng yêu cầu các giao thức phức tạp, do vậy đòi hỏi bộ nhớ cao để thực hiện làm cho phương án này dường như không phù hợp với tất cả các dự án của WSN. Mặc dù vậy, nó rất phù hợp khi sử dụng trong các dự án hoạt động trong khoảng thời gian gian dài.

Năng lượng truyền tải của mỗi gói tin chiếm chi phí đáng kể trong nguồn năng lượng của WSN. Do vậy cần một phương án hiệu quả hơn và đó là kiểm soát lỗi trong truyền thông. Sau đây là một số phương án kiểm soát lỗi trong truyền thông được đưa ra để xem xét đối với WSN.

---

---

### 2.1.2 Tự động phát lại (ARQ).

Một phương án kiểm soát lỗi khá phổ biến trong các mạng truyền thông. Sau đây là nội dung và cơ chế hoạt động của nó trong quá trình kiểm soát lỗi.

Nội dung: ARQ dựa chủ yếu dựa vào việc phát lại các gói dữ liệu bị mất hoặc bị hỏng để duy trì sự tin cậy.

Cơ chế hoạt động:

Bên gửi, sử dụng một phương án phát hiện lỗi tạo các bit dư thừa rồi gắn vào gói dữ liệu truyền đi rồi đợi phản hồi từ bên nhận. Bên gửi được trang bị một bộ định thời gian, nếu sau khoảng thời gian quy định (time out) mà không nhận được phản hồi của bên nhận thì coi như đã có lỗi xảy ra và sẽ truyền lại dữ liệu.

Bên nhận, sau khi nhận được gói dữ liệu, sẽ sử dụng phương án phát hiện lỗi tương ứng kiểm tra gói dữ liệu. Nếu không có lỗi sẽ phản hồi lại bên gửi một ACK cho mỗi gói toàn vẹn.

Các loại ARQ:

- *Go-back-N ARQ* là cho phép bên gửi gửi N gói tin. Nếu có một gói tin không được ACK, thì tất cả các gói tin bị truyền lại.
- *Selective-repeat ARQ* là gửi lại các gói tin chưa được ACK.
- *Stop-and-wait ARQ* là cho phép một gói tin đơn chưa hoàn thành qua và truyền các gói tin sau sau khi các gói tin này đã hoàn thành nghĩa là ACK.

Trong các chiến lược trên stop-and-wait ARQ được sử dụng phổ biến với WSN. ARQ dựa trên cơ chế kiểm soát lỗi phải chịu chi phí đáng kể để bổ sung và phát lại. Tuy nhiên, ARQ thích hợp với ứng dụng bị hạn chế về khả năng xử lý của các nút cảm biến.

### 2.1.3 Sửa lỗi trước khi truyền (FEC).

Trong WSN năng lượng tiêu thụ là yếu tố vô cùng quan trọng, do vậy mà phương án kiểm soát lỗi ARQ thực sự không phù hợp với đa số ứng dụng của mạng này. Một phương án kiểm soát lỗi khác được đưa ra, để tránh việc truyền lại khi gói dữ liệu bị lỗi thì việc sửa lỗi ngay tại nút nhận là giải pháp hợp lý nhất. Hơn nữa, các nút cảm biến với khả năng tự xử lý làm phương án này trở lên khả thi hơn. Sau đây là những luận điểm và chứng minh phương án mới này thực sự khả thi với WSN. Phương án này mang tên “Sửa lỗi trước khi truyền”.

### 2.1.3.1 Cở sở lý thuyết của FEC.

*Sơ bộ hoạt động:* (theo hình 2.2) Theo đó, tại lớp Application dữ liệu tồn tại ở dạng gói và khi tới lớp radio chip nó tồn tại dạng từng bit và được truyền đi. Lớp radio chip có thể là chipcon radio ( loại mới nhất hiện nay) hoặc RFM radio.

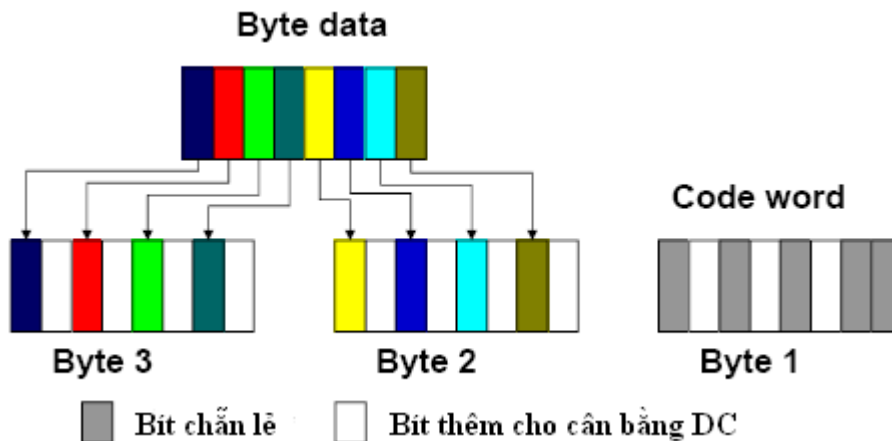
*Bên gửi,* các gói dữ liệu được phân mảnh thành từng byte được mã hóa rồi gửi đi. Trước đó một chuỗi byte đặc biệt gọi là “preamble” hay phân mở đầu được gửi đi nhằm đồng phát hiện và đồng bộ hóa gói dữ liệu.

*Bên nhận,* sau khi nhận được gói mở đầu, nó bắt đầu giải mã các gói sau đó, và MAC thực hiện lắp ráp các gói đó lại và thông báo sự tồn tại gói tin.



**Hình 2.2.** Mô hình xử lý trong WSN.

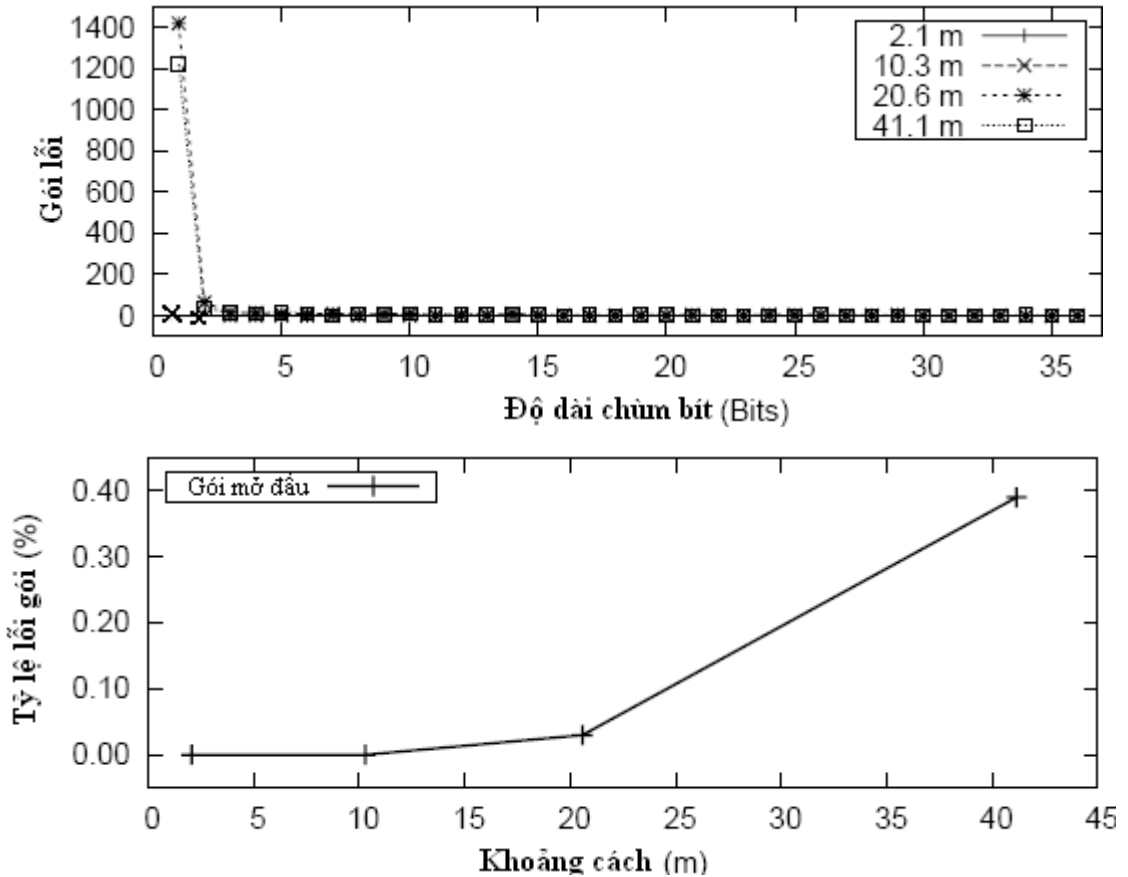
Trong quá trình phân mảnh (hình 2.3), các bit dữ liệu được mã hóa với mã sửa lỗi (ECC) và được ánh xạ vào trong các byte dữ liệu gốc. Nếu có lỗi xảy ra, các bit này sẽ được bên nhận sử dụng để sửa lỗi phục hồi dữ liệu gốc.



**Hình 2.3.** Sơ đồ mã hóa byte dữ liệu.

#### Đặc điểm lỗi trong WSN:

Một thực nghiệm được thực hiện, nút cảm biến với chipcon radio gửi liên tục 10000 gói tin không mã hóa đến một máy PC. Kết quả nhận được cho bởi hình 2.4.



**Hình 2.4.** Biểu đồ tần suất lỗi bit của một nút cảm biến khi phát 10000 gói và tỷ lệ lỗi gói tin mở đầu theo khoảng cách.

Từ biểu đồ hình 2.4 ta thấy đa số lỗi bit trong thực nghiệm là lỗi đơn bit và lỗi 2 bit, lỗi đa bit và lỗi gói khởi đầu có nhưng rất ít. Vì vậy, đối với đặc điểm này mà phương án FEC tập trung vào sửa lỗi đơn bit và hai bit chứ không tập trung sửa lỗi đa bit. Sau đây, trình bày khả năng sửa lỗi trong mã khối tuyến tính vì nó đơn giản và hiệu quả trong WSN.

### 2.1.3.2 Mã sửa lỗi sử dụng trong FEC.

Mã khối tuyến tính:

Phát biểu: // Một mã khối có chiều dài  $n$  gồm  $2^k$  từ mã được gọi là mã tuyến tính  $C(n,k)$  nếu và chỉ nếu  $2^k$  từ mã hình thành một không gian vector con  $k$  chiều của không gian vector  $n$  chiều gồm tất cả các vector  $n$  thành phần trên trường  $GF(2)$ .

Trường  $GF(2)$  (Galois Field 2) là trường nhị phân đồng thời phép cộng là phép cộng modul 2 (cộng tuyệt đối), và phép và (AND). //

Cụ thể: Mã tuyến tính  $C(n, k)$  có mục đích mã hoá những khối tin  $k$  bit thành những từ mã  $n$  bit. Hay nói cách khác trong  $n$  bit của từ mã có chứa  $k$  bit thông tin.

*Cách mã hóa:*  $\mathbf{u} = [a_0, a_1, a_2, \dots, a_{k-1}]$  là thông tin cần được mã hoá thì từ mã  $\mathbf{v}$  tương ứng với  $\mathbf{u}$  sẽ có được bằng cách lấy  $\mathbf{u}$  nhân với ma trận sinh  $\mathbf{G}$ .

$$\text{ta có : } \mathbf{v} = \mathbf{u} \cdot \mathbf{G} \text{ hay } \mathbf{v} = \mathbf{a}_0 \mathbf{g}_0 + \mathbf{a}_1 \mathbf{g}_1 + \dots + \mathbf{a}_{k-1} \mathbf{g}_{k-1}$$

*Cách giải mã:* có  $\mathbf{u} = [a_0, a_1, a_2, \dots, a_{k-1}]$ , từ mã nhận được tại bộ thu  $\mathbf{v} = [b_0, b_1, b_2, \dots, b_{k-1}, b_k, \dots, b_{n-1}]$ , giả sử quá trình truyền không lỗi (tức là  $\mathbf{v}$  thu được là đúng), từ  $\mathbf{v}$  ta tìm lại  $\mathbf{u}$  như sau:

$$\text{Từ phương trình : } \mathbf{v} = \mathbf{u} \cdot \mathbf{G} \text{ hay } \mathbf{v} = \mathbf{a}_0 \mathbf{g}_0 + \mathbf{a}_1 \mathbf{g}_1 + \dots + \mathbf{a}_{k-1} \mathbf{g}_{k-1}$$

ta giải hệ phương trình trên để tìm các  $a_i \Rightarrow$  ta tìm được  $\mathbf{u}$ .

$$\text{ví dụ: } \mathbf{u} = [a_0, a_1, a_2, \dots, a_{k-1}], \mathbf{v} = [b_0, b_1, b_2, \dots, b_{k-1}, b_k, \dots, b_{n-1}], C(7,4)$$

$$G_{4 \times 7} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Từ:  $\mathbf{v} = \mathbf{u} \mathbf{G} \Rightarrow$  hệ phương trình dưới

$$b_0 = a_0 + a_1 + a_3 \quad b_1 = a_0 + a_2 \quad b_2 = a_1 + a_3 \quad b_3 = a_0 + a_1$$

$$b_4 = a_1 \quad b_5 = a_2 \quad b_6 = a_2 + a_3$$

$$\text{với } \mathbf{v} = [1100101] \text{ thay vào hệ trên } \Rightarrow \mathbf{u} = [1101]$$

*Mã hoá khối tuyến tính hệ thống:* Một mã tuyến tính  $C(n,k)$  được gọi là mã tuyến tính hệ thống nếu mỗi từ mã có một trong hai dạng sau:

<b>k bit thông tin</b>	<b>n – k bit kiểm tra</b>
------------------------	---------------------------

Hoặc

<b>n-k bit kiểm tra</b>	<b>k bit thông tin</b>
-------------------------	------------------------

Khi đó ta sẽ xác định được ma trận sinh  $\mathbf{G}$  hệ thống tương ứng.

$$\text{Với dạng 1 thì : } \mathbf{G} = [\mathbf{I}_k, \mathbf{P}]$$

$$\text{Với dạng 2 thì : } \mathbf{G} = [\mathbf{P}, \mathbf{I}_k]$$

Trong đó,  $\mathbf{P}$  là ma trận nhị phân,  $\mathbf{I}_k$  là ma trận đơn vị kích thước  $k \times k$ .

khi đó, thông tin gốc có thể lấy được từ  $k$ -bit đầu hoặc  $k$ -bit cuối của từ mã.





---

---

*Cách phát hiện lỗi:* một ma trận kiểm tra ( $H$ ) của bộ mã có ma trận sinh  $G_{k \times n}$  có kích thước  $(n-k) \times n$  sao cho :  $G \times H^T = \mathbf{0}$ ; Như vậy: Nếu ma trận sinh hệ thống của một mã tuyến tính hệ thống có dạng : ( $H^T$  và  $P^T$  là 2 ma trận chuyển vị)

$$G = [I_k : P] \Rightarrow H = [P^T : I_r] \text{ hoặc } G = [P : I_k] \Rightarrow H = [I_r : P^T]$$

Ta có:  $G \times H^T = [I_k : P][P^T : I_r]^T = P + P = 0$  (xét trong trường  $GF(2)$ ). Nếu gọi vector lỗi là  $e$  thì chúng ta có  $s$  - syndrome:

$$s = r.H^T = (v + e).H^T = uGH^T + eH^T = eH^T$$

$s \neq 0$  nghĩa là có lỗi; sau đó tùy thuộc vào khả năng của mã sửa lỗi,  $s$  được so sánh với từng hàng hay tổng của các hàng trong  $H^T$  để xác định lỗi dựa vào tập hợp các mẫu lỗi ( $T$ ):

- Tồn tại duy nhất một  $t_i \in T$  sao cho  $t_i = s$  thì xác định được vị trí lỗi.
- Tồn tại hơn một  $t_i = s$  ( giả sử  $s = t_i = t_j \in T$  ) thì chỉ xác định được lỗi mà không sửa được.
- Không xác định được  $t_i = s$  hay  $s \notin T$  thì lỗi không xác định được hoặc gói tin bị mất một đoạn nào đó.

Từ đó, ta tìm được vị trí lỗi

- Lỗi đơn bit: có  $s$  trùng với 1 cột trong ma trận  $H$ , vị trí cột là vị trí của lỗi
- Lỗi 2 bit: có  $s$  trùng với phép hoặc tuyệt đối của 2 cột trong ma trận  $H$ , vị trí của các cột trùng với vị trí của lỗi.

*Cách sửa lỗi:* Phía thu sau đó có thể giải mã các từ mã đã được sửa lỗi bằng việc giải phương trình  $v = u.G$  (như đã trình bày ở trên). Một cách đặc biệt, nếu  $G$  là một ma trận sinh hệ thống thì thông tin gốc có thể lấy được từ  $k$ -bit đầu hoặc  $k$ -bit cuối của từ mã nhận được.

Phương pháp mã cột trọng số lẻ (Odd-weight-column): khả năng sửa được lỗi đơn bit và phát hiện lỗi 2bit.

Với đầu vào là  $k$  bit dữ liệu thì  $H$  với số cột  $\geq r+k$  ( $r$  số lượng bit chẵn lẻ). Ví dụ,  $r = 5$  bit chẵn lẻ là cần thiết để phục hồi  $k = 8$  bit dữ liệu, và  $r = 6$  bit chẵn lẻ là cần thiết để phục hồi  $k = 24$  bit dữ liệu.

Xác định các cột của  $H$  như sau:

- $r$  cột cuối của  $H$  là một ma trận đơn vị  $I_r$ .

- k cột đầu của H được lấy từ r hàng vector cột trọng số lẻ ngoại trừ những cái sử dụng trong  $I_r$ .

Ví dụ: cho  $u = [0100\ 0010]$ ,  $k=8$ ,  $r=5$

$$G = [I_8 : C] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad H = [C^T : I_5] = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Từ  $u$  và  $G$  ta có:  $v = uG = [0100\ 0010\ 10111]$

Giả sử rằng các bit thứ hai của thông điệp mã hóa bị đảo ngược do nhiễu. Bên nhận, nhận được thông điệp  $v' = [0000\ 0010\ 10111]$ . Sau đó ta tính được:  $s = v'H^T = [0\ 1\ 0\ 1\ 1]$  từ đây ta thấy rằng  $s$  giống với hàng thứ 2 của ma trận  $H^T$ . Do đó, bằng cách đảo giá trị tại bit thứ 2 của  $v'$  thì ta sẽ được  $v$ .

Sau khi nhận được  $v'$  không lỗi hoặc đã sửa lỗi, với ma trận sinh  $G$  theo một mã có hệ thống thì,  $u$  sẽ là  $k$  bit đầu tiên của  $v'$ . tức là :  $u = [0100\ 0010]$ .

Phương án “Double error correction codes”: khả năng sửa lỗi hai bit và phát hiện lỗi 3 bit. Phương án này chỉ sửa được lỗi đơn bit và 2 bit, lỗi 3 bit chỉ được phát hiện và thông báo cho các lớp trên.

Một mã *Double-Error-Correcting* và *Triple-Error-Detecting* (DEC-TED) (16,8) được đề xuất bởi Gulliver và Bhargave với ma trận nhị phân  $C$ .

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Ma trận sinh  $G$  được tạo ra với dạng  $G=[I_8, C]$  và ma trận chẵn lẻ  $H$  có dạng  $H = [C^T, I_8]$ . Nếu 1 hay 2 bit lỗi xuất hiện, syndrome  $s$  sẽ hoặc là giống với 1 cột trong  $H$  hoặc là bằng phép cộng tuyệt đối của 2 cột trong  $H$ ; chỉ số của các cột này chính là vị trí của bit lỗi.

Ví dụ: cho  $u=[0100\ 0010]$ ,

Lưu ý: rằng khoảng cách Hamming tối thiểu giữa hai hàng bất kì là 5.

$$G = [I_8 : C] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad H = [C^T : I_8] = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Từ đó,  $v = uG = [0100 \ 0010 \ 1001 \ 1100]$

Giả sử, bit thứ 2 và 3 bị đảo ngược do nhiễu. Bên nhận, nhận được thông điệp  $v' = [0010 \ 0010 \ 1001 \ 1100]$ . Từ đó, tính được:  $s = v'H^T = [1010 \ 1111]$ , khi đó ta đem từng cặp hàng trong  $H^T$  XOR với nhau, thì thấy hàng thứ 2 và 3 XOR với nhau bằng s. Vậy lỗi nằm ở bit thứ 2 và 3 của  $v'$ . Từ đó, đảo giá trị bit ta sẽ được v.

Sau khi nhận được  $v'$  không lỗi hoặc đã sửa lỗi, với ma trận sinh G theo một mã có hệ thống thì, u sẽ là 8 bit đầu tiên của  $v'$ , tức là,  $u = [0100 \ 0010]$ .

Trên đây, là những lý thuyết cơ bản của phương án FEC và cách sửa lỗi đơn bit và 2 bit của nó. Đối với mạng WSN, *FEC thực sự thích hợp được chọn làm phương án kiểm soát lỗi hiệu quả nhất.*

#### 2.1.4 ARQ lai ghép (HARQ).

Phương án ARQ và FEC duy trì độ tin cậy trong các kênh thông qua các kỹ thuật khác nhau. ARQ thường có hiệu quả với kênh truyền tốt vì không có bit dư thừa được gửi đi. Tuy nhiên, ARQ trở nên không hiệu quả khi gặp lỗi do phải gửi lại toàn bộ gói tin, mặc dù chỉ có một số bit là bị hỏng. FEC thì ngược lại, với hiệu quả kênh truyền không tốt vì phải truyền thêm nhưng bit dư thừa nhưng lại tốt trong điều kiện gặp lỗi vì nó có khả năng phục hồi lỗi. một phương án mới được tính đến đó là Hybrid ARQ hay ARQ lai ghép.

Nội dung: ARQ lai ghép (HARQ) kết hợp phương án FEC và ARQ bằng cách nâng cao khả năng phục hồi lỗi trong một gói tin thông qua truyền lại.

Cơ chế hoạt động: một gói uncoded hoặc mã hóa thấp được gửi lần đầu tiên. Trong trường hợp lỗi kênh, HARQ thực hiện phát lại bằng cách sử dụng khả năng của FEC. Với 2 cách xử lý khác nhau trong 2 loại HARQ.

##### Các loại HARQ:

- *Kỹ thuật HARQ-I:* Nếu gói tin nhận bị lỗi, bên nhận sẽ gửi một NACK cho bên gửi. Sau đó, bên gửi, sẽ gửi một gói với dữ liệu và mã FEC.
- *Kỹ thuật HARQ-II:* nếu gặp lỗi, chỉ có các bit mã FEC được gửi đi.

Trong khi loại II giảm sử dụng băng thông, thì loại I không yêu cầu các gói dữ liệu đã gửi trước đó được lưu trữ ở phía bên nhận.

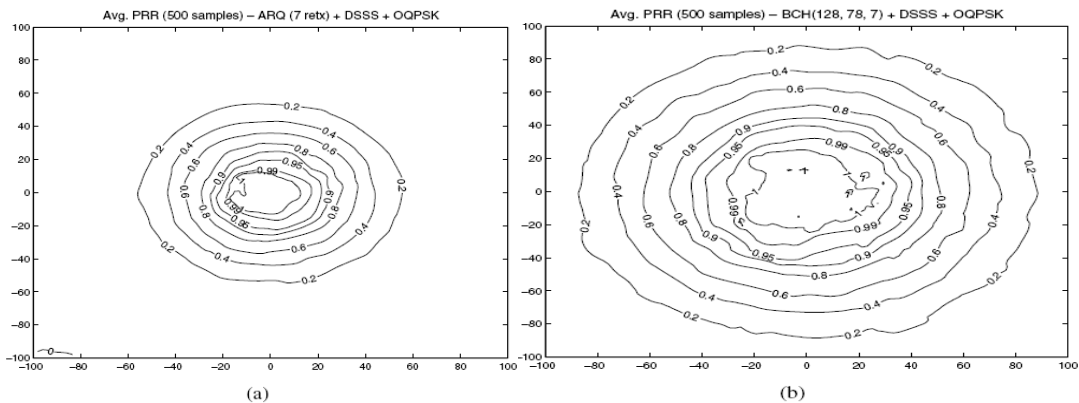
## 2.2 Lợi ích của tăng khả năng phục hồi lỗi.

Đối với hai phương án FEC và HARQ với khả năng phục hồi lỗi đã làm năng lượng tiêu thụ trong truyền thông đã được kiểm soát rất tốt, tăng phạm vi truyền hiệu quả ở tỉ lệ lỗi gói tin (PER) giảm. Nhìn chung lợi ích này được tạo ra bởi hai khả năng chính mà FEC và HARQ cung cấp đó là, mở rộng độ dài bước nhảy và kiểm soát năng lượng truyền tải.

### Khả năng 1: Mở rộng độ dài bước nhảy.

Trong các mạng multi-hop, khả năng phục hồi lỗi của FEC và HARQ đã nâng cao phạm vi truyền thông hiệu quả của một nút (hình 2.6). Trong hình những đường nét thể hiện PER xung quanh một nút phát. Các đường nét hiển thị các vị trí có PER tương ứng với giá trị cụ thể.

So sánh Hình 2.6 (a) và Hình 2.6 (b) đối với tỷ lệ mục tiêu nhận gói tin là 90%, phạm vi truyền tải trung bình của một nút là 29m với ARQ và 42m với FEC ở điện năng truyền tải là -5 dBm. Nó tăng 45% phạm vi hiệu quả, rõ ràng mã FEC có thể tăng phạm vi truyền dẫn hiệu quả của một nút so với ARQ. Từ đó, có thể thấy, việc tăng độ dài bước nhảy trong WSN làm tăng phạm vi truyền dẫn hiệu quả.



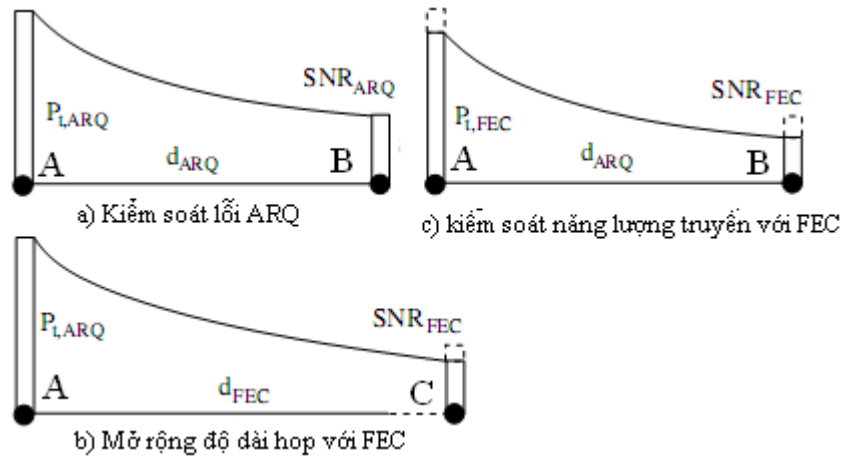
**Hình 2.6.** Trung bình tỉ lệ mục tiêu nhận gói tin, đồ thị (a) ARQ (N = 7) và (b) FEC (BCH (128, 78, 7)) cho nút MicaZ.

### Phương án 2: Kiểm soát năng lượng truyền tải.

Thông qua khả năng phục hồi lỗi của mã FEC cũng có thể giảm năng lượng truyền tải mà vẫn đảm bảo chất lượng mạng. Kỹ thuật này được gọi là kiểm soát

năng lượng truyền tải và cải thiện hiệu quả mạng bằng cách giảm nhiễu, giảm một phần năng lượng tiêu thụ.

Dẫn chứng cho việc nâng cao hiệu quả mạng với FEC: xem xét FEC để so sánh với ARQ thông qua PER trong mạng (Hình 2.7).



**Hình 2.7.** So sánh hiệu quả của FEC với ARQ.

Trong Hình 2.7(a), truyền thông giữa nút A và B với ARQ. Nút A sử dụng truyền tải điện năng là  $P_{t,ARQ}$  để gửi một gói tin đến nút B, cách xa một khoảng  $d_{ARQ}$ . Thu được  $SNR_{ARQ}$  tại nút B. Theo đó, xác định một  $PER_{ARQ}^*$ .

Bây giờ thay ARQ bằng FEC: có 2 trường hợp xảy ra.

Trường hợp mở rộng độ dài bước nhảy: Hình 2.7 (b), nút A lựa chọn nút C là bước nhảy tiếp theo xa hơn B. Do đó,  $d_{ARQ} > d_{FEC}$  và vẫn là  $P_{t,ARQ}$ ,  $SNR_{FEC}$  nhỏ hơn  $SNR_{ARQ}$ . Do đó, vẫn đạt được giá trị PER tương tự ARQ.

Trường hợp kiểm soát năng lượng truyền: Hình 2.7 (c), nút A vẫn sử dụng nút B là bước nhảy tiếp theo tức là  $d_{FEC} = d_{ARQ}$ . Tuy nhiên, sử dụng năng lượng truyền tải nhỏ hơn, tức là,  $P_{t,FEC} < P_{t,ARQ}$ ,  $SNR$  vẫn nhỏ hơn. Do đó, cũng có được PER tương tự với ARQ.

Ngoài những lợi thế, phương án FEC và HARQ phải bỏ chi phí năng lượng và độ trễ cho mã hóa/giải mã, mà làm truyền thông lâu hơn. Do đó, các phương án kiểm soát lỗi đòi hỏi phân tích kỹ lưỡng. Phần tiếp theo, trình bày một hướng phân tích “mô hình lớp chéo” để thấy ưu điểm và nhược điểm của các phương án này.

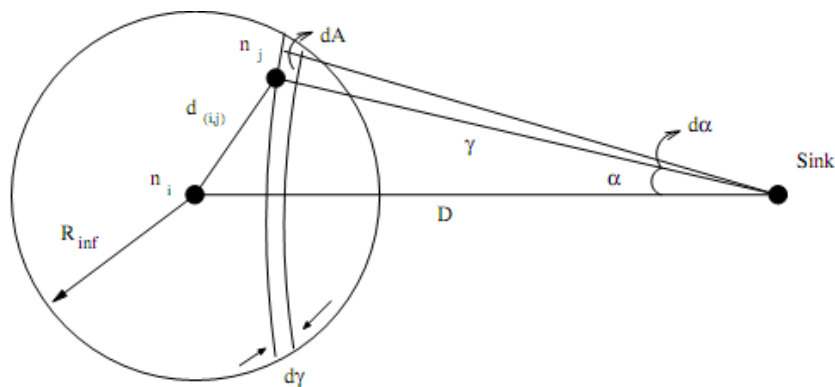
## 2.3 Phân tích mô hình lớp chéo.

Cung cấp một cái nhìn cụ thể về các phương án ARQ, FEC, và HARQ. Mục tiêu của mô hình này là để phân tích và xác định các yếu tố sau đây: Khoảng cách bước nhảy dự kiến, năng lượng tiêu thụ, độ trễ và PER tổng thể. Sau đó giải thích mô hình và mô tả cách tính toán các số liệu này.

### 2.3.1 Mô hình tham chiếu của WSN.

*Mô tả mô hình:* (hình 2.8)

- Các nút cảm biến phân phối hai chiều với mật độ ( $\rho$ ).
- Một nút hoạt động trong khoảng thời gian ( $\delta$ ) và những nút khác thì ngủ.
- Là ứng dụng theo dõi với tỷ lệ báo cáo của cảm biến là thấp, nhưng các thông điệp được truyền đáng tin cậy với PER tổng thể.
- Bước nhảy tiếp theo được xác định theo SNR nhận của gói tin được gửi từ một nút  $i$  cách sink một khoảng  $D$ .
- Trong số những nút lân cận của  $i$ , có nút  $j$  gần sink hơn và có SNR cao hơn SNR ngưỡng ( $\psi_{Th}$ ), tức là ( $\psi_j > \psi_{Th}$ ), được chọn là bước nhảy tiếp theo.
- Truy cập môi trường được thực hiện thông qua trao đổi bắt tay:
  - FEC là RTS-CTS-DATA.
  - ARQ là RTS-CTS-DATA-ACK.
  - HARQ là RTS-CTS-DATA-NACK.
- Khi một nút gửi ( $n_i$ ) truyền một gói tin đến sink thông qua một đường mà nút ( $n_j$ ) là bước nhảy kế tiếp. Với phạm vi truyền của một nút ( $R_{inf}$ ), là khoảng cách mà tại đó gói tin có thể nhận thành công.



**Hình 2.8.** Mô hình tham chiếu của WSN.

*Lưu ý:* Việc đặt khoảng cách bước nhảy và các thông số liên quan tại mỗi bước nhảy là chu kỳ nhiệm vụ hoạt động độc lập. Kết quả là, trạng thái của mạng sẽ thay đổi tại mỗi bước nhảy do các nút sẽ thức dậy vào thời điểm khác nhau. Mô hình phân tích chi phí cho quá trình truyền một gói dữ liệu từ nút nguồn dữ liệu cách sink một khoảng  $D$ . Được tính toán bằng những công thức cụ thể dưới đây.

Tổng năng lượng tiêu thụ của một luồng từ nút nguồn cách sink 1 khoảng  $D$ .

$$E_{flow}(D) = E[E_h] E[n_h(D)] \quad (2.1)$$

Trong đó,  $E[E_h]$  là năng lượng tiêu thụ dự kiến tại mỗi bước nhảy và  $E[n_h(D)]$  là số lượng bước nhảy dự kiến từ nút nguồn cách sink một khoảng  $D$ .

Tương tự, độ trễ tổng thể của nó được cho bởi công thức 2.2.

$$T_{flow}(D) = E[T_h] E[n_h(D)] \quad (2.2)$$

Trong đó,  $E[T_h]$  là độ trễ dự kiến tại mỗi bước nhảy. Số lượng bước nhảy dự kiến tính bằng công thức 2.3.

$$E[n_h(D)] \simeq \frac{D - R_{inf}}{E[d_h]} + 1 \quad (2.3)$$

Trong đó,  $D$  là khoảng cách tổng thể,  $R_{inf}$  là phạm vi truyền của nút,  $E[d_h]$  là khoảng cách bước nhảy dự kiến. Trong các mục 2.3.2, 2.3.3, và 2.3.4 tiếp theo, sẽ làm rõ với khoảng cách bước nhảy dự kiến  $E[d_h]$ , mức tiêu thụ năng lượng dự kiến tại mỗi bước nhảy  $E[E_h]$ , và độ trễ dự kiến tại mỗi bước nhảy  $E[T_h]$ .

### 2.3.2 Khoảng cách bước nhảy dự kiến.

Trong hình 2.8, nơi mà một nút  $j$  nằm trong khu vực rất nhỏ  $dA = d\gamma da$  tại tọa độ  $(\gamma, \alpha)$  đối với sink. Khoảng cách từ nút  $i$  đến nút  $j$  là:

$$d_{(i,j)} = d(D, \gamma, \alpha) = \sqrt{\gamma^2 + D^2 - 2\gamma D \cos \alpha} \quad (2.4)$$

Trong đó,  $D$  là khoảng cách từ nút nguồn  $i$  đến sink,  $(\gamma)$  là khoảng cách giữa sink và nút kế tiếp tiếp  $j$ , và góc  $\alpha$  như trong hình 2.8. Theo đó, khoảng cách bước nhảy dự kiến  $E[d_h]$ , được tính bằng công thức 2.5.

$$E[d_h] = \int_{\gamma_{min}}^D \int_{-\alpha_\gamma}^{\alpha_\gamma} d_{(i,j)} dP\{\mathcal{N}_i = j\} \quad (2.5)$$

Trong đó,  $\gamma_{min}=D-R_{inf}$  là khoảng cách tối thiểu giữa một nút bước nhảy tiềm năng tiếp theo  $j$  đến sink. Hơn nữa,  $d_{(i,j)}$  là khoảng cách giữa nút  $i$  và  $j$  được cho bởi

công thức (2.4) và  $dP\{N_i = j\}$  là xác suất cho rằng bước kế tiếp  $N_i$  được chọn là nút  $j$ . Cuối cùng,  $\alpha_\gamma = a \cdot \cos[(\gamma^2 + D^2 - R_{inf}^2) / (2\gamma D)]$  là giới hạn của khoảng cách.

Điều kiện  $j$  được chọn là bước nhảy tiếp theo:

- Nút  $j$  tồn tại trong khu vực  $dA$ .
- Các SNR nhận của nút  $j$  phải đáp ứng ( $\psi_j > \psi_{Th}$ ).
- Các SNR nhận của mỗi nút  $k$ , gần với sink hơn so với nút  $j$ , phải thấp hơn ngưỡng SNR, tức là ( $\psi_k \leq \psi_{Th}$ ).

Do đó, xác suất mà nút  $j$  được chọn như là bước nhảy kế tiếp được tính:

$$dP\{N_i = j\} = P\{N_A(d\gamma) = 1\}P\{\psi_j > \psi_{Th}\}P\{d_{(j,s)} \geq \gamma\} \quad (2.6)$$

Trong đó,  $N_A(d\gamma)$  là số lượng nút trong vùng  $dA$  tại khoảng cách  $\gamma$  từ sink;  $P\{N_A(d\gamma) = 1\}$  là xác suất có một nút trong khu vực  $A(d\gamma)$ ;  $P\{\psi_j > \psi_{Th}\}$  là xác suất mà SNR nhận của nút  $j$  trên SNR ngưỡng ( $\psi_{Th}$ );  $P\{d_{(j,s)} \geq \gamma\}$  là xác suất có những nút gần sink hơn  $j$  có SNR dưới ngưỡng có thể tìm thấy tại khoảng cách  $\gamma$  từ sink.

Thành phần thứ nhất,  $P\{N_A(d\gamma) = 1\}$ , trong (2.6) có thể được xấp xỉ bằng:

$$\begin{aligned} P\{N_A(d\gamma) = 1\} &\simeq 1 - e^{-\rho\delta\gamma d\gamma d\alpha} \quad \text{as } d\gamma \rightarrow 0 \\ &\simeq \rho\delta\gamma d\gamma d\alpha \end{aligned} \quad (2.7)$$

Trong đó,  $\rho$  là mật độ nút trong mạng và  $\delta$  là tham số chu kỳ nhiệm vụ. Thành phần  $e^{-x} \simeq 1 - x$  được dùng khi  $(\rho\delta\gamma d\gamma d\alpha) \rightarrow 0$  cũng như  $d\gamma \rightarrow 0, d\alpha \rightarrow 0$ .

Để tính toán  $P\{\psi_j > \psi_{Th}\}$  và  $P\{d_{(j,s)} \geq \gamma\}$  trong (2.6). Cần xác định năng lượng thu nhận ở một khoảng cách  $d$  từ nút phát  $P_r(d)$  được cho bởi công thức 2.8.

$$P_r(d) = P_t - PL(d_0) - 10\eta \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad (2.8)$$

Trong đó,  $P_t$  là công suất truyền tải tính bằng (dBm),  $PL(d_0)$  là tổn hao năng lượng ở khoảng cách lý thuyết  $d_0$  tính bằng (dBm),  $\eta$  là số mũ tổn hao và  $X_\sigma$  là thành phần bóng mờ, với  $X_\sigma \sim N(0, \sigma)$ .

Các SNR nhận cho bởi:  $\psi(\mathbf{d}) = \mathbf{Pr}(\mathbf{d}) - \mathbf{Pn}$  tính bằng (dBm), trong đó  $\mathbf{Pn}$  là năng lượng nhiễu (dBm).

Thành phần thứ hai,  $P\{\psi_j > \psi_{Th}\}$ , trong (2.6) tính bằng công thức 2.9.

$$\begin{aligned} P\{\psi_j > \psi_{Th}\} &= P\{X_\sigma > \beta(d_{(i,j)}, \psi_{Th})\} \\ &= Q\left(\frac{\beta(d_{(i,j)}, \psi_{Th})}{\sigma}\right) \end{aligned} \quad (2.9)$$



$$\beta(d_{(i,j)}, \psi_{Th}) = \psi_{Th} + P_n - P_t + PL(d_0) + 10\eta \log_{10} \left( \frac{d_{(i,j)}}{d_0} \right) \quad (2.10)$$

Trong đó,  $Q(x) = 1/\sqrt{2\pi} \int_x^\infty e^{-(t^2/2)} dt$ .

Thành phần thứ ba,  $P\{d(j, s) \geq \gamma\}$ , trong (2.6). Khu vực bao gồm các nút gần với sink hơn so với nút  $j$  là  $A(\gamma)$ . Thì:

$$\begin{aligned} P\{d_{(j,s)} \geq \gamma\} &= \sum_{i=0}^{\infty} P\{N_{A(\gamma)} = i\} p_k(\gamma)^i \\ &= \sum_{i=0}^{\infty} \frac{e^{-M(\gamma)} M(\gamma)^i}{i!} p_k(\gamma)^i \\ &= e^{-M(\gamma)(1-p_k(\gamma))} \end{aligned} \quad (2.11)$$

Trong đó  $A(\gamma)$  là vùng giao nhau của hai vòng tròn với tâm phân cách bởi  $D$  và với bán kính  $R_{inf}$  và  $\gamma$  tương ứng.

Do đó,  $N_{A(\gamma)}$  là số nút trong  $A(\gamma)$  và  $M(\gamma) = \rho\delta A(\gamma)$  là số lượng trung bình của các nút trong khu vực này.  $P_k(\gamma) = P\{\psi_k \leq \psi_{Th}, k \in A(\gamma)\}$  là khả năng mà một nút  $k$  trong  $A(\gamma)$ , với SNR nhận thấp hơn SNR ngưỡng tức là  $\psi_k \leq \psi_{Th}$ . Điều này có thể tìm thấy trong công thức 2.12.

$$p_k(\gamma) = \int_{\gamma_{min}}^{\gamma} \int_{-\alpha\gamma}^{\alpha\gamma} \left[ 1 - Q\left(\frac{\beta}{\sigma}\right) \right] \frac{1}{A(\gamma)} d\alpha d\gamma \quad (2.12)$$

(Trong đó  $\gamma_{min} = D - R_{inf}$ )

Sử dụng (2.6), (2.7), (2.9), (2.11), và (2.12) trong (2.5), khoảng cách bước nhảy dự kiến có thể được tìm ra bởi công thức 2.13.

$$E[d_h] = \rho\delta \int_{\gamma_{min}}^D \int_{-\alpha\gamma}^{\alpha\gamma} \gamma d_{(i,j)} Q\left(\frac{\beta}{\sigma}\right) e^{-M(\gamma)(1-p_k(\gamma))} d\alpha d\gamma, \quad (2.13)$$

### 2.3.3 Phân tích năng lượng tiêu thụ dự kiến.

Năng lượng tiêu thụ dự kiến được tính toán bằng cách xem xét một nút  $j$  như trong hình 2.8. Việc tiêu thụ năng lượng dự kiến giữa  $i$  nút và nút  $j$  được ký hiệu là  $E[E_j]$ , là một hàm của  $\gamma$  và  $\alpha$ . Dựa vào mục 2.3.2 và thay thế  $d_{(i,j)}$  bởi  $E[E_j]$  trong (2.13), mức tiêu thụ năng lượng dự kiến mỗi bước nhảy được tính bằng công thức:

$$E[E_h] = \rho\delta \int_{\gamma_{min}}^D \int_{-\alpha\gamma}^{\alpha\gamma} \gamma E[E_j] Q\left(\frac{\beta}{\sigma}\right) e^{-M(\gamma)(1-p_k(\gamma))} d\alpha d\gamma \quad (2.14)$$

Một nút có thể trở thành một bước nhảy tiếp theo nếu có giá trị SNR nhận lớn hơn ngưỡng, mức năng lượng tiêu thụ dự kiến  $E[E_j]$ , tính bằng công thức 2.15.

$$E[E_j] = \int_{\psi_{Th}}^{\infty} E_{comm}(d_{(i,j)}, \psi) f_{\psi}(d_{(i,j)}, \psi) d\psi \quad (2.15)$$

Trong đó,  $E_{comm}(d_{(i,j)}, \psi)$  là năng lượng tiêu thụ cho truyền một gói tin giữa các nút  $i$  và  $j$  ở khoảng cách  $d_{(i,j)}$  và  $\psi$  là giá trị SNR tại nút  $j$ . Thành phần thứ hai,  $f_{\psi}(d_{(i,j)}, \psi)$  là hàm mật độ xác suất của SNR, là một hàm của các  $d_{(i,j)}$  hiệu quả.

Vì  $P(\psi \leq \beta) = P(X_{\sigma} \leq \beta(d_{(i,j)}, \psi))$ ,  $f_{\psi}(d_{(i,j)}, \psi)$  được tính:

$$f_{\psi}(d_{(i,j)}, \psi) = f_{X_{\sigma}}(\beta(d_{(i,j)}, \psi)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\beta(d_{(i,j)}, \psi))^2/2\sigma^2} \quad (2.16)$$

Trong đó  $\beta(d_{(i,j)}, \psi)$  đã có trong công thức 2.10.

Năng lượng tiêu thụ cho một giao tiếp trong WSN: gồm ba thành phần

- ✓ Năng lượng tiêu thụ của nút truyền
- ✓ Năng lượng tiêu thụ của nút nhận
- ✓ Năng lượng tiêu thụ cho nút lân cận.

Theo đó, ta có công thức:

$$E_{comm} = E_{TX} + E_{RX} + E_{neigh} \quad (2.17)$$

Trong đó  $E_{TX}$  là năng lượng tiêu thụ bởi nút gửi (nút  $i$ ),  $E_{RX}$  là năng lượng tiêu thụ bởi nút nhận (nút  $j$ ), và  $E_{neigh}$  là năng lượng tiêu thụ bởi các nút lân cận.

Ba giá trị tiêu thụ năng lượng phụ thuộc vào MAC và sơ đồ điều khiển lỗi được triển khai trên mỗi nút. Theo đó, để truyền tải các gói dữ liệu thành công, nút một cần hoàn thành bốn bước bắt tay: RTS-CTS-DATA-ACK của ARQ, ba bước bắt tay: RTS-CTS-DATA của FEC, và RTS-CTS-DATA-NACK của HARQ.

Trước khi truyền một gói tin RTS, của truy cập môi trường, một nút thực hiện cơ chế cảm nhận sóng mang để đánh giá sự sẵn sàng của một kênh truyền và truyền tải một gói tin sau đó.

Một cấp phát thành công của kênh phụ thuộc vào cả cảm nhận sóng mang thành công và truyền tải không có va chạm. Xác suất cảm nhận sóng mang thành công ( $P_{cs}$ ), có thể được tính như sau:

$$p_{cs} = 1 - (1 - p_{cf})^{K+1} \quad (2.18)$$

Trong đó  $K$  là số lần “cảm nhận lại” cho phép đối với một truyền tải và  $(P_{cf})$  là xác suất cảm nhận được kênh tự do, được tính như sau

$$p_{cf} = e^{-\lambda_{net}(\tau_{cs} + T_{comm})} \quad (2.19)$$

Trong đó  $\tau_{cs}$  là thời gian truyền cảm nhận và  $T_{comm}$  là thời gian truyền gói tin. Sau một cảm nhận truyền thành công, một va chạm chỉ có thể xảy ra nếu một nút truyền trong khoảng thời gian dễ bị đụng độ của  $\tau_{cs}$ . kết quả là, xác suất không có va chạm  $P_{noColl}$  tính bởi công thức:

$$P_{noColl} = e^{-\lambda_{net}\tau_{cs}} \quad (2.20)$$

Kí hiệu  $\lambda_{net}$  trong (2.19) và (2.20) là lưu lượng tổng thể được tạo ra bởi tất cả các nút trong phạm vi truyền dẫn của một nút, được cho bởi công thức 2.21.

$$\lambda_{net} = \lambda \frac{P_{cs}}{P_{comm}} (1 - (1 - p_{comm})^{L+1}) \quad (2.21)$$

Trong đó  $\lambda$  là tổng lưu lượng được tạo trong phạm vi truyền dẫn của một nút và  $P_{comm}$  là xác suất truyền thành công. Theo đó, xác suất một nút có thể nhận thành công trên kênh được cho bởi  $P_{cs}$  và  $P_{noColl}$ , có thể được tìm thấy bằng cách giải hệ phương trình (2.18) và (2.21).

Giá trị ( $\lambda$ ) phụ thuộc vào *tỷ lệ lưu lượng được tạo* và *kích thước của gói*. Giả định rằng các nút cảm biến có tỷ lệ lấy mẫu trung bình là  $(b)$  bit/s. Biểu diễn chiều dài tải trọng gói tin là  $(L_D)$ , tỷ lệ trung bình tạo gói của một nút  $i$  là  $(\lambda_{ii} = b/L_D)$  pkts/s.

Trong WSN, từ một nút cũng có thể chuyển tiếp các gói tin thông qua các nút khác đến sink, tỉ lệ truyền gói tin của một nút cao hơn giá trị truyền kiểu này. Nếu một phương án định tuyến bằng cách chia sẻ tải nguyên mạng giữa các nút thì tỉ lệ trung bình truyền tải gói tin của nút là  $(\lambda_i = c_i \lambda_{ii})$  với  $c_i > 1$ . Do đó,  $\lambda$  tại (2.21)

được cho bởi công thức:  $\lambda = \sum_{i=1}^{M_n} \lambda_i$ ; Khi đó số lượng các nút có trong phạm vi truyền dẫn của một nút là  $(M_n - 1)$ .

Khi truy cập kênh truyền, mức tiêu thụ năng lượng phụ thuộc vào xác suất nhận thành công gói dữ liệu  $p_s^D(\psi)$  và gói kiểm soát  $p_s^C(\psi)$  ở khoảng cách  $d_{(i,j)}$  với giá trị SNR  $\psi$ . Cách tính  $p_s^D(\psi)$  và  $p_s^C(\psi)$  sẽ được giải thích trong mục 2.3.5.

Ngoài ra, một gói tin truyền (RTS) chỉ thành công nếu một nút có thể nhận thành công trên kênh (cho bởi  $P_{cs}$  và  $P_{noColl}$ ).

Năng lượng tiêu thụ của nút truyền: Theo đó, năng lượng tiêu thụ cho nút truyền  $E_{TX}$  được cho bởi công thức tương ứng với ARQ, FEC và HARQ.

$$E_{TX}^{ARQ} = n_{ret}^{ARQ} \{ E_{sense} + p_{cs}(E_{tx}^C + p_{noColl} p_s^C E_{rx}^C + p_{noColl}(p_s^C)^2 E_{tx}^D + p_{noColl}(p_s^C)^2 p_s^D E_{rx}^C + (1 - p_{noColl} p_s^C) E_{t/o}^C + p_{noColl}(p_s^C)^2 (1 - p_s^D) E_{t/o}^D \} \quad (2.22)$$

$$E_{TX}^{FEC} = n_{ret}^{FEC} \{ E_{sense} + p_{cs}(E_{tx}^C + p_{coll} p_s^C (E_{rx}^C + E_{dec}^C) + (1 - p_{coll} p_s^C) E_{t/o}^C) \} + p_{cs} p_{coll} (p_s^C)^2 E_{tx}^D \quad (2.23)$$

$$E_{TX}^{HARQ} = n_{ret}^{HARQ} \{ E_{sense} + p_{cs}(E_{tx}^R + E_{rx}^C + E_{dec}^C + E_{tx}^{D1}) \} + (p_s^C)^2 (1 - p_s^{D1}) (E_{rx}^N + E_{dec}^C + E_{tx}^{D2}) \quad (2.24)$$

Với  $n_{ret}$  của ARQ, FEC, và HARQ, tương ứng khi đó là:

$$n_{ret}^{ARQ} = (1 - p_{cs} + p_{cs} p_{noColl} (p_s^C)^3 p_s^D)^{-1} \quad (2.25)$$

$$n_{ret}^{FEC} = n_{ret}^{HARQ} = (1 - p_{cs} + p_{cs} p_{coll} (p_s^C)^2)^{-1} \quad (2.26)$$

Các giá trị  $n_{ret}$  là số lượng dự kiến truyền lại với ARQ, FEC và HARQ, trong đó lại được giới hạn bởi số lượng tối đa cho phép của truyền lại;  $E_{tx}^x$  và  $E_{rx}^x$  là năng lượng truyền và nhận gói tin, trong đó: R, C, D và A tương ứng với RTS, CTS, DATA, và ACK;  $E_{sense}$  là mức tiêu thụ năng lượng cho cảm nhận khu vực. Hai thành phần cuối cùng trong dấu ngoặc đơn của 3 công thức thể hiện năng lượng tiêu thụ với độ trễ cho CTS và ACK trong trường hợp lỗi gói tin.

Ngoài ra,  $E_{dec}^x$  là năng lượng giải mã, D1 và D2 trong (2.24) tương ứng với gói dữ liệu truyền cho lần đầu tiên và lần thứ hai trong HARQ.

Hoạt động truyền một gói tin: Nếu các kênh cảm biến ở trạng thái tốt, nút truyền một gói tin RTS và nhận được một gói tin CTS nếu không có sự va chạm và các gói tin RTS được nhận thành công tại đích đến. Tương tự, một gói dữ liệu được gửi nếu các gói tin CTS được nhận thành công, tiếp đó là tiếp nhận gói tin ACK.

Năng lượng tiêu thụ của nút nhận: Sử dụng phương pháp tương tự, mức năng lượng tiêu thụ của nút nhận,  $E_{RX}$  tại (2.17), được tính như sau:

$$E_{RX}^{ARQ} = n_{ret}^{ARQ} \{ E_{rx}^R + E_{tx}^C + E_{rx}^D + E_{tx}^A \} \quad (2.27)$$

$$E_{RX}^{FEC} = n_{ret}^{FEC} \{ E_{rx}^R + E_{dec}^R + E_{tx}^C + E_{rx}^D + E_{dec}^D \} \quad (2.28)$$

$$E_{RX}^{HARQ} = n_{ret}^{HARQ} \{ E_{rx}^R + E_{dec}^R + E_{tx}^C \} + E_{rx}^{D1} + E_{dec}^{D1} + (p_s^C)^2 (1 - p_s^{D1}) (E_{tx}^N + E_{rx}^{D2} + E_{dec}^{D2}) \quad (2.29)$$

Năng lượng tiêu thụ cho nút lân cận: Trong các mạng không dây, những nút lân cận của nút truyền và nút nhận cũng bị ảnh hưởng bởi các thông tin liên lạc do tính chất quảng bá của kênh không dây. Do đó, kí hiệu cuối cùng trong (2.17),  $E_{neigh}$ , là năng lượng tiêu thụ bởi những nút láng giềng của các nút truyền và nhận. Được tính bằng những công thức sau:

$$E_{neigh}^{ARQ} = n_{ret}^{ARQ} \{ (\rho\delta\pi R_{inf}^2 - 2)E_{rx}^R + [\rho\delta(\pi R_{inf}^2 - A(D, R_{inf}, D)) - 2]E_{rx}^C \} \quad (2.30)$$

$$E_{neigh}^{FEC} = E_{neigh}^{HARQ} = n_{ret}^{FEC} (\rho\delta\pi R_{inf}^2 - 2)E_{rx}^R + [\rho\delta(\pi R_{inf}^2 - A(D, R_{inf}, D)) - 2]E_{rx}^C \quad (2.31)$$

### 2.3.4 Phân tích độ trễ dự kiến.

Các biểu thức xác định độ trễ tổng thể của luồng phát sử dụng phương pháp tương tự như trên. Độ trễ mỗi bước nhảy được cho bởi công thức 2.32.

$$E[T_h] = \rho\delta \int_{\gamma_{min}}^D \int_{-\alpha_\gamma}^{\alpha_\gamma} \gamma E[T_j] Q\left(\frac{\beta}{\sigma}\right) e^{-M(1-p_k)} d\alpha d\gamma \quad (2.32)$$

$$E[T_j] = \int_{\psi_{Th}}^{\infty} T_{comm}(\psi, d_{(i,j)}) f_\psi(\psi, d_{(i,j)}) d\psi \quad (2.33)$$

Trong đó,  $T_{comm}$  là độ trễ để truyền thông với ARQ, FEC và HARQ, được tính bởi công thức:

$$\begin{aligned} T_{comm}^{ARQ} = & n_{ret}^{ARQ} \{ T_{sense} + 2p_{cs}p_{noColl}(p_s^C)^2 T^{Ctrl} \\ & + p_{cs}(1 - p_{noColl}(p_s^C)^2) T_{t/o}^C + p_{cs}p_{noColl}(p_s^C)^3 p_s^D (T^D + T^{Ctrl}) \\ & + p_{cs}p_{noColl}(p_s^C)^3 (1 - p_s^C p_s^D) T_{t/o}^A \} \end{aligned} \quad (2.34)$$

$$T_{comm}^{FEC} = n_{ret}^{FEC} \{ T_{sense} + 2T^{Ctrl} + 2T_{dec}^C + T^D + T_{dec}^D \} \quad (2.35)$$

$$\begin{aligned} T_{comm}^{HARQ} = & n_{ret}^{HARQ} \{ T_{sense} + 2T^{Ctrl} + 2T_{dec}^C + T^{D1} + T_{dec}^{D1} \} \\ & + (p_s^C)^2 (1 - p_s^{D1}) (T^{Ctrl} + T_{dec}^C + T^{D2} + T_{dec}^{D2}) \end{aligned} \quad (2.36)$$

Trong đó,  $T_{sense}$  là thời gian dành cho cảm biến,  $T^{Ctrl}$  và  $T^D$  là thời gian truyền gói kiểm soát và gói dữ liệu,  $T_{t/o}$  là thời gian chờ,  $T_{dec}^{Ctrl}$  và  $T_{dec}^D$  là độ trễ giải mã cho các gói kiểm soát và gói dữ liệu.

Độ trễ giải mã và năng lượng: Một trong những chi phí quan trọng của mã FEC ngoài các truyền và nhận các bit dư thừa là tiêu thụ năng lượng để mã hóa và giải mã các gói dữ liệu cũng như độ trễ liên quan đến nó. Như đã biết rằng *năng lượng mã hóa* cho mã khối là không đáng kể. Do đó, *năng lượng giải mã và độ trễ*

giải mã nên được xem xét trong phân tích này. Các độ trễ của giải mã cho một mã khối (n, k, t) được cho bởi công thức:

$$T_{dec}^{BL} = (2nt + 2t^2)(T_{add} + T_{mult}) \quad (2.37)$$

Trong đó,  $T_{add}$  và  $T_{mult}$  là độ trễ do cộng và nhân tương ứng, trong trường  $GF(2^m)$ , với  $m = \lceil \log_2 n + 1 \rceil$ . Cả hai nút Mica2 và MicaZ được thực hiện với vi điều khiển 8-bit, nó có thể thực hiện cộng và nhân của 8 bit trong chu kỳ một và hai, tương ứng. Kết quả là:

$$T_{add} + T_{mult} = 3 \left\lceil \frac{m}{8} \right\rceil t_{cycle} \quad (2.38)$$

Trong đó,  $t_{cycle}$  là một chu kỳ thời gian, thường bằng 250 ns. Do đó, việc tiêu thụ năng lượng cho giải mã là  $E_{dec}^{BL} = I_{proc} V T_{dec}^{BL}$  trong đó  $I_{proc}$  là cường độ dòng cho bộ xử lý,  $V$  là điện áp cung cấp và  $T_{dec}^{BL}$  độ trễ cho giải mã.

### 2.3.5 Phân tích BER và PER.

Việc tiêu thụ năng lượng và độ trễ này phụ thuộc vào PER trong truyền thông. Trong phần này, cung cấp các biểu thức cho BER và PER cho các nút Mica2 và MicaZ. Bởi những phương án điều chế được sử dụng trong mỗi nút này rất khác nhau, do đó nó là cần thiết để nghiên cứu tác động của FEC và HARQ vào các nút này một cách riêng biệt.

#### Tính BER:

Với nút Mica2: được thực hiện với một phương án điều chế không nhất quán FSK. Các BER của phương án này được cho bởi công thức:

$$p_b^{FSK} = \frac{1}{2} e^{-(E_b/N_0)/2}, \quad E_b/N_0 = \psi \frac{B_N}{R} \quad (2.39)$$

Trong đó  $\psi$  là SNR nhận,  $B_N$  là băng thông nhiễu và  $R$  là tốc độ dữ liệu.

Với nút MicaZ: được điều chế bởi phương án khóa dịch pha cần phương lệch (O-QPSK) với trải phổ tuần tự trực tiếp (DSSS). Các BER của phương án này được cho bởi công thức:

$$p_b^{OQPSK} = Q(\sqrt{(E_b/N_0)_{DS}})$$

$$(E_b/N_0)_{DS} = \frac{2N \times E_b/N_0}{N + 4(E_b/N_0)(K - 1)/3} \quad (2.40)$$

Trong đó,  $N$  là số lượng chip/bit, và  $K$  là số nút sử dụng đồng thời truyền tải.

Tính PER:

Căn cứ vào BER ( $P_b$ ) đã tính ở trên, các PER cho các phương án kiểm soát lỗi có thể được tính như sau. Đối với ARQ, cơ chế phát hiện lỗi CRC-16 được triển khai ở cả hai loại nút Mica trên. Giả sử tất cả các lỗi trong một gói có thể được phát hiện, các PER của đơn truyền một gói tin với trọng tải  $l$  bit được cho bởi công thức:

$$PER^{CRC}(l) = 1 - (1 - p_b)^l \quad (2.41)$$

Đối với các mã BCH, giả sử sự đan xen hoàn thiện ở bộ thu phát, tỷ lệ lỗi khối (BLER) được cho bởi công thức:

$$BLER(n, k, t) = \sum_{i=t+1}^n \binom{n}{i} p_b^i (1 - p_b)^{n-i} \quad (2.42)$$

Qua thuật toán Berlekamp-Massey được thực hiện và mô phỏng để tìm các mối quan hệ giữa các BLER và BER. Khi gói tin lớn hơn độ dài khối  $n$ , đặc biệt là trong đó độ dài khối nhỏ được sử dụng, các PER cho FEC được cho bởi công thức:

$$PER^{FEC}(l, n, k, t) = 1 - (1 - BLER(n, k, t))^{[l/k]} \quad (2.43)$$

Trong đó  $[l/k]$  là số khối cần thiết để gửi  $l$  bit và  $[.]$  là Hàm mức trần. Sử dụng (2.41), (2.42), và (2.43), các PER với các phương án HARQ cũng được tìm thấy. Xác suất thành công của gói tin,  $p_s^C$  và  $p_s^D$  được sử dụng trong mục 2.3.3, mục 2.3.4 với các gói kiểm soát và gói dữ liệu sau đó có thể được tìm thấy bằng cách sử dụng  $l=l_C$  và  $l=l_D$ , tương ứng.

**2.4 So sánh các phương án kiểm soát Lỗi trong WSN.**

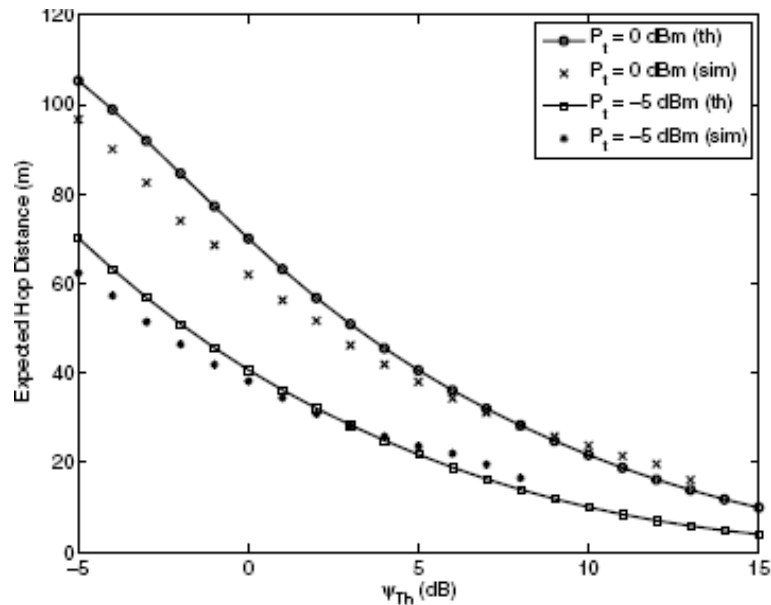
Trong phần này, minh họa tác động của các phương án FEC và HARQ với PER, năng lượng tiêu thụ và độ trễ tổng thể trong một mạng multi-hop thông qua các đánh giá bằng số liệu trong MATLAB và mô phỏng. Để thực hiện so sánh này, hai loại nút cảm biến được sử dụng là Mica2 và MicaZ ngoài ra có một số trường khác, các thông số trong Bảng 2.1, được sử dụng cho các số liệu kết quả.

**Bảng 2.1.** Các tham số.

				Mica2	MicaZ	
$D$	300 m	$l_C$	8 bytes	$e_{rx}$	21 mJ	59.1 mJ
$P_t$	0, -5, -15 dBm	$l_D$	38 bytes	$e_{tx} (P_t = 0)$	24 mJ	52.2 mJ
$PL_{d0}$	55 dB	$t_{cycle}$	250 ns	$e_{tx} (P_t = -5)$	21.3 mJ	42 mJ
$P_n$	-105 dBm	$I_{proc}$	8 mA	$e_{tx} (P_t = -15)$	16.2 mJ	29.7 mJ
$\eta$	3	$V$	3 V	$t_{bit} = 1/R$	62.4 $\mu$ s	4 $\mu$ s
$\sigma$	3.8			$N$	N/A	16 chips
				$K$	N/A	2

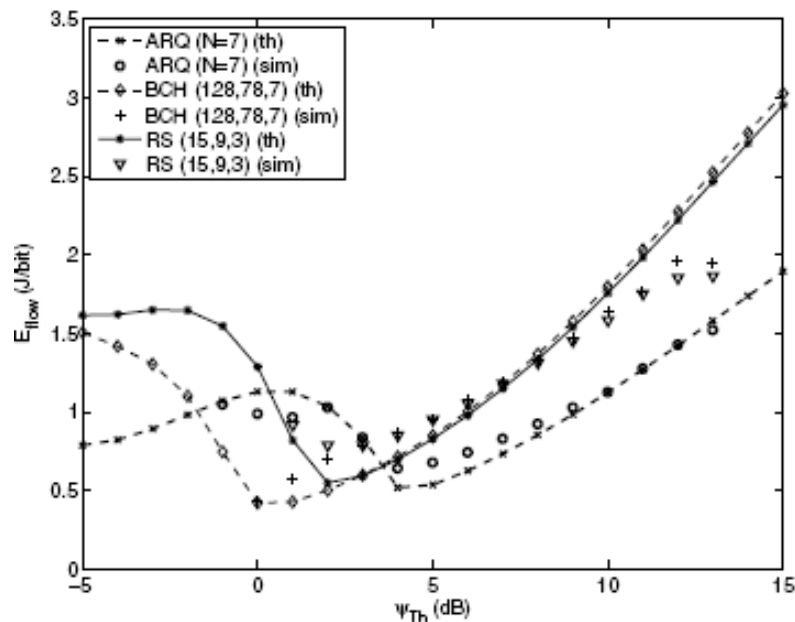
Các kí hiệu sử dụng, khoảng cách bước nhảy dự kiến là  $d_{hop}$ , được xác định tại (2.5), SNR nhận ( $\psi_{Th}$ ), năng lượng truyền ( $P_t$ ).

*Khả năng tăng khoảng cách bước nhảy ở FEC và HARQ:*



**Hình 2.9.** Trung bình khoảng cách bước nhảy (MicaZ).

Từ hình 2.9, nhận thấy ở  $\psi_{Th}$  nhỏ độ dài bước nhảy lớn là số lượng bước nhảy từ một nút tới sink giảm. Hơn nữa, khi truyền tải điện năng của nút giảm, khoảng cách bước nhảy dự kiến sẽ giảm.



**Hình 2.10.** Năng lượng tiêu thụ của một lưu lượng với  $\psi_{Th}$  (MicaZ).



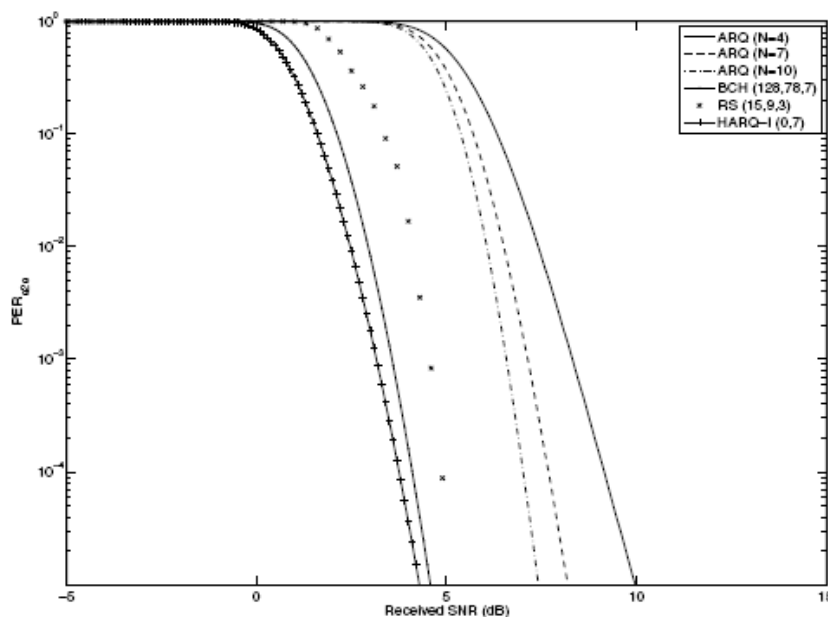
Trong hình 2.10, năng lượng tiêu thụ của luồng giảm khi giá trị  $\psi_{Th}$  giảm từ 15dB. Điều này do khoảng cách bước nhảy dự kiến tăng trong hình 2.9.

Tuy nhiên, mức tiêu thụ năng lượng cho ARQ tăng đáng kể khi  $\psi_{Th}$  giảm nhỏ hơn một giá trị cụ thể, ví dụ, 5 dB. Vì chất lượng kênh thấp dẫn đến truyền lại xảy ra, làm tăng tiêu thụ năng lượng trên mỗi bước nhảy.

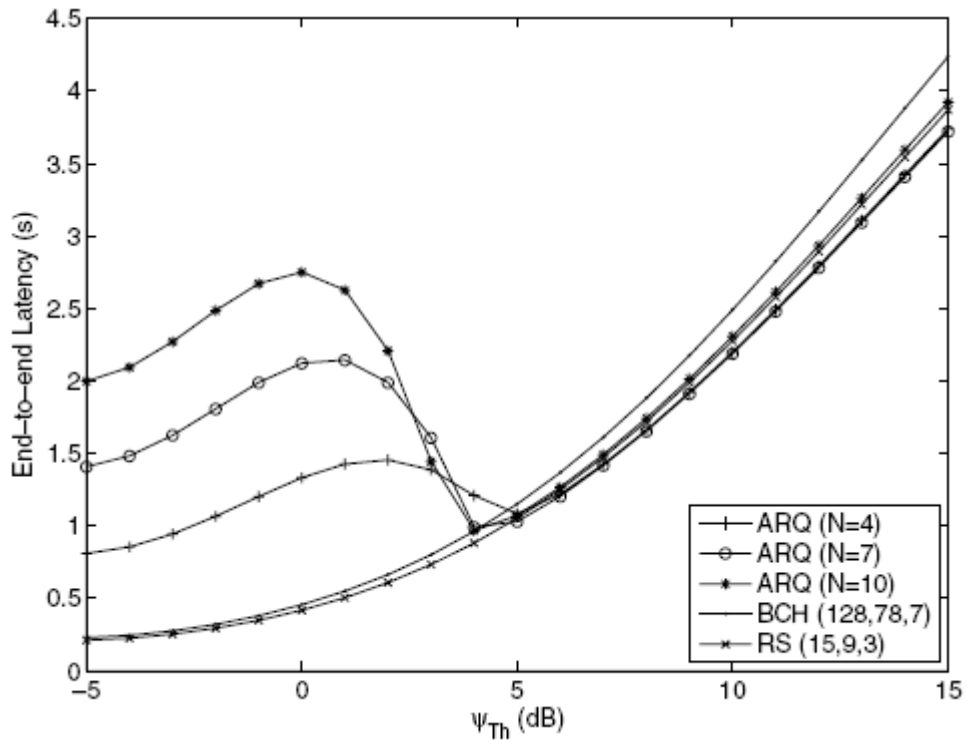
*Lưu ý:* đối với ARQ, các đường cong tiêu thụ năng lượng đạt đến một đỉnh cao và giảm khi  $\psi_{Th}$  giảm. Điểm này tương ứng với trường hợp truyền lại với số lượng tối đa, không còn đủ để truyền thông tin cậy.

Trong hình 2.10, với các mã của FEC, khi chất lượng kênh truyền tốt thì năng lượng tiêu thụ nhiều hơn với ARQ bởi phải truyền thêm bit dư thừa và chi phí giải mã, nhưng hiệu quả năng lượng hơn với  $\psi_{Th} \sim 2\text{dB}$  vì khả năng phục hồi lỗi.

Theo hình 2.11, nhận thấy PER của các mã FEC đã giảm khi giá trị SNR nhận vẫn còn thấp, từ đó có thể thấy khả năng phục hồi lỗi của FEC có thể giúp mạng hoạt động tốt trên một kênh truyền có chất lượng kênh không tốt, mà vẫn đảm bảo độ tin cậy cho gói tin. Trong hình 2.12, lại thấy với SNR thấp, các mã FEC hoạt động với độ trễ tốt hơn ARQ tuy nhiên ở những SNR cao hơn thì độ trễ của nó tăng rất nhanh cao hơn ca ARQ. Nguyên nhân, chính là do việc giải mã và các bit dư thừa tạo ra. Tuy nhiên, xét về tổng thể nó vẫn tốt hơn với ARQ vì ở SNR thấp, độ trễ của FEC ít hơn rất nhiều so với ARQ.



**Hình 2.11.** PER Tổng thể với  $\psi_{Th}$  (MicaZ).



**Hình 2.12.** Độ trễ trung bình tổng thể với  $\psi_{Th}$  (MicaZ).

Ví dụ, nếu PER mục tiêu của một ứng dụng là  $10^{-2}$ , giá trị tối thiểu cho  $\psi_{Th}$  tương ứng với 6.1dB cho ARQ, 3dB cho BCH(128, 78, 7), 4.8dB cho RS(15, 9, 3), và 2.5dB cho HARQ-I. Kết quả là, trong hình 2.10 có BCH(128, 78, 7) là tiết kiệm năng lượng hơn chút ít so với ARQ. Mặt khác, mã RS(15,9,3) kết quả năng lượng tiêu thụ cao hơn so với ARQ. Rõ ràng rằng năng lượng được tiêu thụ trên bước nhảy với mã FEC gồm cả truyền bit dư thừa và giải mã. Tuy nhiên, vì khả năng phục hồi lỗi được cải thiện với mã FEC, các giá trị SNR thấp có thể được hỗ trợ. Kết quả là khoảng cách bước nhảy dài hơn còn có thể đạt đến năng lượng tiêu thụ tổng thể thấp hơn.

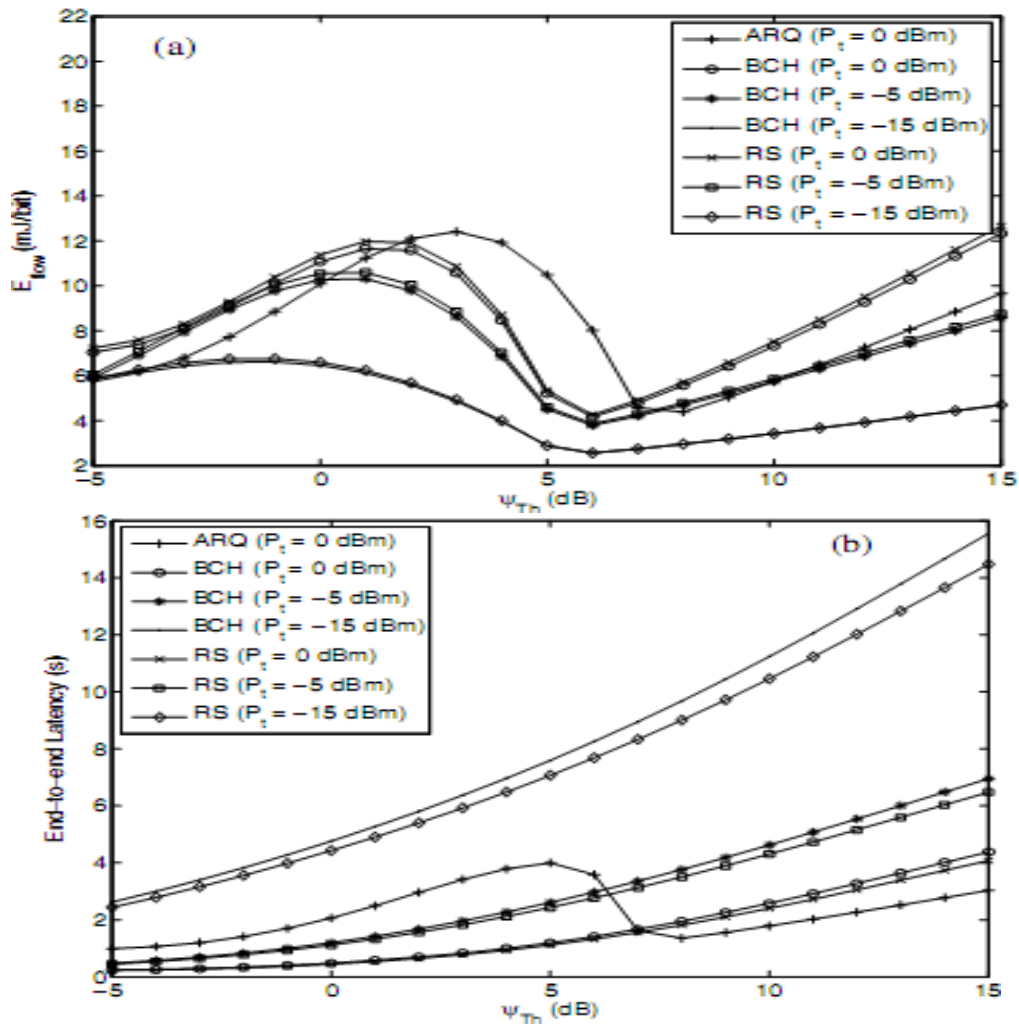
Khả năng kiểm soát năng lượng truyền tải ở FEC và HARQ:

Để kiểm chứng ảnh hưởng của năng lượng truyền (Pt) đến phương án kiểm soát lỗi FEC hãy xem xét ba cấp điện áp: 0,-5 và -15 dBm. Khi giảm năng lượng truyền có thể nâng cao hiệu quả năng lượng cho phương án FEC (hình 2.13 a) vì:

- Tiêu thụ ít năng lượng hơn cho việc truyền gói dữ liệu mã hóa dài hơn.
- Phạm vi giao thoa của một nút giảm, số lượng các nút láng giềng có thể tiêu thụ năng lượng nhân rồi cũng giảm.

- Làm tăng số lượng bước nhảy.

Thể hiện trong hình 2.13(b), nơi mà độ trễ tổng thể được thể hiện. Trái ngược với mở rộng khoảng cách bước nhảy, do kiểm soát năng lượng truyền không ảnh hưởng đến thời gian cần thiết để truyền một gói tin, độ trễ tổng thể phụ thuộc vào số lượng bước nhảy, dẫn đến tăng đáng kể trong độ trễ, đặc biệt ảnh hưởng rất lớn với mã BCH của FEC.



**Hình 2.13.** (a) Năng lượng tiêu thụ trung bình của một luồng. (b) Độ trễ tổng thể trung bình với ngưỡng ( $\psi_{Th}$ ) cho những  $P_t$  khác nhau.

Kết luận từ việc so sánh: các phương án kiểm soát lỗi được sử dụng trong WSN có những ưu nhược điểm khác nhau. Tuy nhiên, trong WSN phương án kiểm soát lỗi FEC vẫn được coi trọng hơn cả. Ngoài ra, phương án HARQ cũng được xem xét kỹ càng trong từng ứng dụng khác nhau.

---

---

## CHƯƠNG 3: BÀI TOÁN KIỂM SOÁT LỖI TRONG WSN

### 3.1 Vấn đề và giải pháp trong WSN.

Trong chương 1 của đề án đã đề cập đến rất nhiều nội dung xung quanh WSN, từ cấu tạo của một nút cảm biến không dây cho đến những ứng dụng, những yếu tố ảnh hưởng đến thiết kế. Nhưng có một điểm chung đó là xoay xung quanh vấn đề năng lượng tiêu thụ. Có thể nói, về phần cứng với khả năng hoạt động độc lập của các nút cảm biến không dây làm WSN hầu như thích hợp với mọi ứng dụng, mọi địa hình. Tuy nhiên, chính do sự độc lập đó đã tạo ra một yếu tố sống còn của WSN, đó là với một nguồn nuôi vô cùng hạn chế từ Pin khiến nó có thể ngưng hoạt động bất cứ khi nào một nút mạng bị “hết điện”.

Các giải pháp đã được đưa ra, để sao cho một nút tiêu thụ năng lượng là thấp nhất mà vẫn phải đảm bảo độ tin cậy về dữ liệu cho mạng. Cuối cùng, phương án giảm thiểu truyền thông giữa các nút mạng được lựa chọn hơn cả, bởi theo những thực nghiệm, các nhà nghiên cứu rút ra một kết luận đó là *năng lượng tiêu thụ cho truyền thông tốn hơn rất nhiều so với xử lý*. (được nói cụ thể trong mục 1.5.7). Khi đó, hướng tập trung đã đẩy vào vấn đề kiểm soát lỗi, vì kiểm soát lỗi tốt sẽ giúp mạng hoạt động hiệu quả hơn dẫn đến sử dụng năng lượng tiết kiệm hơn.

Trong kiểm soát lỗi lại chia hai hướng chính, đó là ARQ và FEC đã được trình bày chi tiết trong mục 2.1.2 và 2.1.3. Cụ thể có thể rút gọn nội dung là, với phương án ARQ, khi gặp lỗi tại nơi nhận thì sẽ thực hiện phát lại, còn với FEC thì khi gặp lỗi sẽ phục hồi lỗi ngay tại nơi nhận và truyền tiếp. Chính vì đặc điểm này mà trong WSN, FEC là một lựa chọn hàng đầu vì không tốn chi phí cho phát đi phát lại nhiều lần rất thích hợp với WSN. Một vấn đề được nghĩ tới, vậy các thuật toán sửa lỗi rất phức tạp, đặc biệt với lỗi đa bit. Do đó, một số nghiên cứu đã được thực hiện và kết luận, *trong WSN lỗi thường gặp là lỗi đơn bit và hai bit lỗi đa bit cũng có nhưng rất ít*, trình bày trong mục 2.1.3. Vậy là, FEC hoàn toàn có thể được sử dụng một cách hiệu quả trong WSN.

Sau đây, là một bài toán cụ thể nhằm củng cố những nhận định về lý thuyết giữa hai phương án kiểm soát lỗi ARQ và FEC trong WSN.

---

---

## 3.2 Bài toán so sánh giữa sửa lỗi và phát lại trong WSN.

### 3.2.1 Phát biểu bài toán.

Bài toán (a):

Đối với một kênh truyền dẫn có trung bình là 1 lỗi trong 4000 bit. Những lỗi bit đơn lẻ này được thống kê độc lập. Một gói cơ sở bao gồm 128 byte. Nó có thể được truyền: đầy đủ, chính xác hoặc không. Nút nhận có thể phát hiện được lỗi một gói tin truyền mà không mất chi phí bổ sung. Nếu một lỗi xuất hiện nút nhận yêu cầu nút gửi phát lại duy nhất một lần. Các yêu cầu phát lại được coi như một gói thông thường là 128 byte. Nếu lỗi xảy ra trong một yêu cầu nó thì coi như không có yêu cầu được gửi đi.

Làm thế nào có được tỷ lệ dữ liệu tổng thể cao trong trường hợp này (theo phần trăm tỷ lệ dữ liệu về mặt lý thuyết có thể đạt được, nếu không có lỗi)?

Bài toán (b):

Để cho dễ dàng hơn chúng ta giả định rằng một bit lỗi xảy ra một lần trên mỗi gói. Thay vì yêu cầu nút gửi phát lại chúng ta chọn phương pháp sửa lỗi trước khi truyền (FEC) với mã Hamming (mã này là không quan trọng trong bài tập này).

Làm thế nào có được tỷ lệ dữ liệu thực tế cao trong trường hợp này (theo phần trăm) so với tính toán theo mặt lý thuyết không sửa lỗi trước và không có lỗi?

### 3.2.2 Nhận định từ bài toán.

Bài toán được chia hai phần, với hai phương án kiểm soát lỗi thông dụng trong WSN. Mục tiêu chính của hai phần là hướng vào tốc độ dữ liệu.

Trong trường hợp (a), tuy không nói trực tiếp nhưng dễ dàng nhận thấy phương án kiểm soát lỗi được sử dụng là ARQ do nếu có lỗi xuất hiện trên một gói thì sẽ thực hiện truyền lại gói. Còn trong trường hợp (b), đã nêu trực tiếp phương án sửa lỗi là FEC.

Trong trường hợp (b), thực chất là việc so sánh tốc độ dữ liệu đạt được khi sử dụng phương án FEC với ARQ. Để qua đó, có một cái nhìn chính xác về hai phương án này. Bổ sung thêm khẳng định về mặt lý thuyết nêu ở chương 2.

### 3.2.3 Giải quyết bài toán.

#### Giải quyết bài toán (a):

Để giải quyết bài toán này, ta phải tập trung vào một số dữ kiện mà đề bài cho dưới đây.

(1) Dữ kiện thứ nhất: “Đối với một kênh truyền dẫn có trung bình là 1 lỗi trong 4000 bit”. Từ đó ta có thể suy ra:

- Tỷ lệ lỗi bit (BER) là:  $p_b = 1 / 4000$ .
- Tương ứng một bit được truyền chính xác với một xác suất = 3999/4000.

(2) Dữ kiện thứ hai: “Một gói cơ sở bao gồm 128 byte”.

Ta có thể chuyển sang tải trọng bit một lần truyền là ( $l$ ):  $l = 128 \times 8 = 1024$  bit.

Nhữ đã biết ở chương 2, ARQ chủ yếu sử dụng phương án CRC-16 để phát hiện lỗi kết hợp với hai dữ kiện 1 và dữ kiện 2 ta có thể xác định tỷ lệ lỗi gói (PER). Theo đó, có 2 trường hợp đặt ra:

- Trong môi trường lý thuyết, tức là không có lỗi nào xảy ra thì:

$$PER^{CRC}(l) = (1 - p_b)^l$$

Từ đó, có thể tính ra tỷ lệ lỗi gói lý thuyết =  $(1 - 1/4000)^{1024} \approx 77,4 \%$

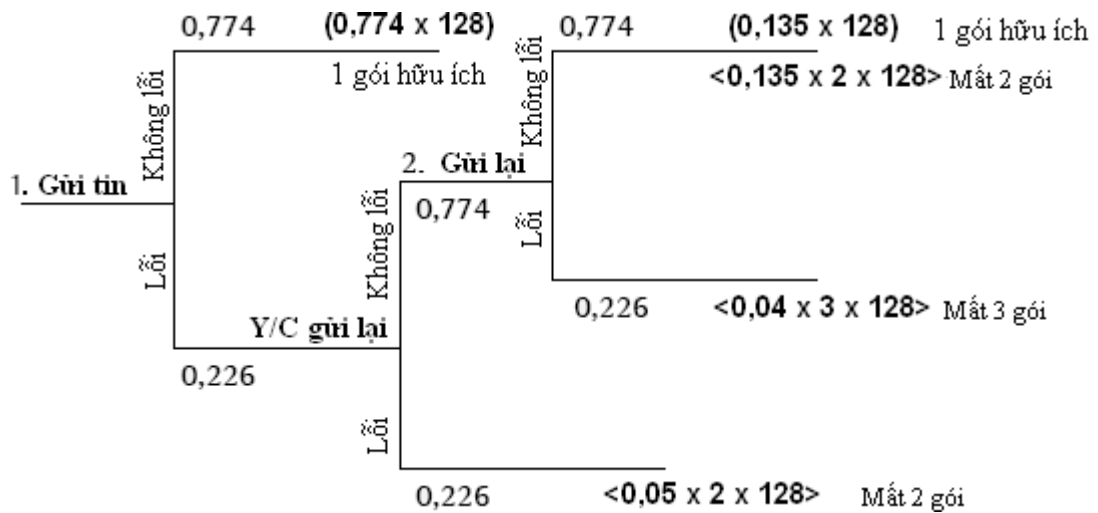
- Khi có lỗi xảy ra do bất kì trường hợp nào. Thì sử dụng công thức 2.41.

$$PER^{CRC}(l) = 1 - (1 - p_b)^l$$

Ta có thể tính ra tỷ lệ lỗi gói =  $1 - (1 - 1/4000)^{1024} \approx 22,6 \%$

(3) Dữ kiện thứ 3: “Gói có thể được truyền đầy đủ, chính xác hoặc không. Nếu một lỗi xuất hiện nút nhận yêu cầu nút gửi phát lại duy nhất một lần. Các yêu cầu phát lại được coi như một gói thông thường là 128 byte. Nếu lỗi xảy ra trong một yêu cầu nó thì coi như không có yêu cầu được gửi đi”

Từ dữ kiện 3 này, ta có thể lập một sơ đồ trường hợp xảy ra khi thực hiện một chu kì gửi gói tin và kiểm soát gói tin với phương án ARQ như sau:



Quá trình thực hiện như sau:

Ban đầu gửi một gói tin, sẽ có 2 trường hợp xảy ra là có lỗi và không có lỗi. Nếu không có lỗi coi như truyền thành công. Nếu có lỗi xảy ra, thì một yêu cầu phát lại được truyền đi. Đối với yêu cầu phát lại, nếu lỗi xảy ra với yêu cầu phát lại thì sẽ dừng không phát nữa, ngược lại nếu yêu cầu này thành công thì bên gửi sẽ thực hiện phát lại một lần duy nhất (theo đề bài). Với gói tin phát lại tương ứng xảy ra hai trường hợp như lần đầu gửi. tuy nhiên nếu lại có lỗi xảy ra thì ngừng phát.

Từ quá trình hoạt động, gửi một gói tin và những trường hợp xảy ra với một gói trong quá trình gửi đó. Với tỷ lệ lỗi gói tin trong hai trường hợp không lỗi (0.774) và có lỗi (0.226). Kết quả, ta tìm được dữ liệu hữu ích trong dấu “( )” và số lượng byte mất đi do truyền thông trong dấu “<>”.

Từ sơ đồ hoạt động và kết quả nhận được ta xác định được:

- Số lượng byte hiệu quả (payload) dự kiến là:

$$0,774 \times 128 + 0,135 \times 128 = 116 \text{ (byte)}.$$

- Số lượng byte mất mát dự kiến là:

$$0,135 \times 2 \times 128 + 0,04 \times 3 \times 128 + 0,05 \times 2 \times 128 = 63 \text{ (byte)}.$$

Khi đó, tận dụng khả năng tối đa của kênh ta có được tỷ lệ dữ liệu tổng thể:

$$\text{Tỷ lệ dữ liệu tổng thể} = 116 / (116 + 63) \approx 64,8\%$$

---

---

Giải quyết bài toán (b):

Như đã biết FEC là một phương pháp chèn thêm các bit dư thừa vào gói dữ liệu để sau này khi có lỗi xảy ra bên nhận sẽ sử dụng các bit dư thừa này để phục hồi dữ liệu nguyên gốc. Cụ thể về cách phục hồi lỗi trong FEC đã được trình bày trong mục 2.1.3. Ở đây chỉ đi vào những yếu tố của FEC ảnh hưởng đến tỷ lệ dữ liệu tổng thể.

Với dữ kiện ở bài toán (b) “*giả định rằng một bit lỗi xảy ra một lần trên mỗi gói*”, tức là ở bài toán này chỉ có lỗi đơn bit xảy ra. Bởi vậy nằm trong khả năng sửa lỗi của FEC trình bày trong mục 2.1.3. Do việc truyền lại của FEC là không có do vậy dữ liệu chỉ có duy nhất một trường hợp là gói luôn đến nút nhận. Tuy nhiên, như đã nói ở trên, FEC cần chèn thêm các bit dư thừa vào dữ liệu, vì vậy tỷ lệ dữ liệu hữu ích phải loại bỏ thành phần này.

Để tính toán ra số lượng bit dư thừa, bằng cách: từ dữ kiện 2 ở bài toán (a), ta có tải trọng gói tin là: 1024 bit. Qua đó, để bảo vệ 1024 bit dữ liệu chống lại 1 lỗi bit thì phải thêm số lượng bit dư thừa (r) theo bất đẳng thức sau đây:

$$1024 + r + 1 \leq 2^r$$

Từ đó, có thể thấy r=11 là thỏa mãn với bất đẳng thức trên. Vì :

$$1024 + 11 + 1 \leq 2^{11} = 2048.$$

Qua đó có thể xác định tỷ lệ dữ liệu tổng thể =  $1024 / (1024 + 11) \approx 98,9\%$

Một hướng giải quyết khác của bài toán (b):

Nếu như, trong gói tin (128 byte) đã có chứa sẵn các bit dư thừa rồi thì: tỷ lệ dữ liệu tổng thể =  $(1024 - 11) / 1024 \approx 98,9\%$ .

Nếu các bit kiểm lỗi được sắp xếp theo byte, thì tính toán sẽ là

$(1024-16)/1024=98.4\%$  , ở đây con số 16 là do 11 bit xếp theo byte là 2 byte.

Kết luận từ bài toán:

Bằng những số liệu chính xác từ bài toán nêu trên, ta thấy phương án kiểm soát lỗi FEC và ARQ với cùng một tải trọng gói tin, cùng điều kiện truyền tuy nhiên với FEC với khả năng phục hồi lỗi trực tiếp tại nơi nhận hiệu quả hơn rất nhiều so với ARQ. Bằng chứng là thông qua tỷ lệ dữ liệu tổng thể đối với ARQ chỉ là 64,8% còn với FEC là 98,9%. Qua đó có thể khẳng định, trong WSN phương án FEC hoàn toàn chiếm ưu thế so với phương án ARQ về mặt hiệu quả dữ liệu.



---

---

## KẾT LUẬN

### **Những vấn đề đã nghiên cứu được:**

Nắm được tổng quan về mạng cảm biến không dây (WSN), về thành phần của một nút cảm biến không dây (phần cứng, phần mềm và chuẩn nút), đồng thời cung cấp thông tin phân loại một số loại nút cảm biến không dây hiện nay đang được sử dụng phổ biến.

Giới thiệu một số ứng dụng của mạng cảm biến không dây trong các lĩnh vực như: quân sự, y tế, công nghiệp, môi trường và sử dụng tại nhà. Từ đó, đưa ra các yếu tố ảnh hưởng đến thiết kế mạng và chỉ ra yếu tố sống còn trong thiết kế mạng cảm biến không dây.

Như một giải pháp, củng cố khả năng hoạt động hiệu quả của mạng truyền thông nói chung, và đặc biệt quan trọng với mạng cảm biến không dây nói riêng, đó là vấn đề kiểm soát lỗi trong truyền thông. Tìm hiểu được *đặc điểm lỗi chính của WSN gặp phải là lỗi đơn bit và hai bit*, Lần lượt đưa ra các phương án kiểm soát lỗi từ đơn giản đến phức tạp, bằng những số liệu chính xác từ thực nghiệm hay mô hình lớp chéo, đưa ra những so sánh công bằng giữa các phương án kiểm soát lỗi.

Để củng cố về mặt lý thuyết nêu ra, một bài toán cụ thể đã được giải quyết với mục đích tăng tính thuyết phục cho việc so sánh hai phương án kiểm soát lỗi chính trong mạng cảm biến không dây ARQ và FEC. Để cuối cùng đưa ra khẳng định *Phương án FEC hoàn toàn chiếm ưu thế so với phương án ARQ trong mạng cảm biến không dây*.

### **Những vấn đề tồn tại cần được giải quyết:**

Trong các phương án kiểm soát lỗi trong mạng cảm biến không dây, HARQ hay ARQ lai ghép thực sự hiệu quả hơn FEC trên một số thí nghiệm. Điều này cho thấy, phương án HARQ có thể là một phương án rất tốt khi áp dụng trong WSN. Tuy nhiên, nó chưa được nghiên cứu kỹ càng trong đề án.

Đã củng cố lý thuyết về vấn đề kiểm soát lỗi. Tuy nhiên, chỉ với một bài toán so sánh ARQ và FEC thông qua tỉ lệ dữ liệu tổng thể, chưa thể chứng minh hoàn toàn việc chiếm ưu thế của FEC trong WSN.

## TÀI LIỆU THAM KHẢO

- [1]. Ian F. Akyildiz and Mehmet Can Vuran, *Wireless Sensor Networks*, 2010.
- [2]., Jaein Jeong and Cheng Tien Ee, Department of Electrical Engineering and Computer Science, *Forward Error Correction in Sensor Networks*, University of California, Berkeley 2003.
- [3]. A. J. Goldsmith, *Wireless Communications*, 2005.
- [4]. Thomas Haenselmann, *Sensor Network*, 2008.
- [5]. S.Lin and D.J.Costello Jr, *Error Control Coding: Fundamentals and Applications*, 1983.
- [6]. M. C. Vuran and I. F. Akyildiz, *Error control in wireless sensor networks: a cross layer analysis*. IEEE/ACM Transactions on Networking, 2009.
- [7]. L. C. Zhong and J. M. Rabaey, *An integrated data-link energy model for wireless sensor networks*. In Proceedings of IEEE ICC'04, 2004.