Lời mở đầu

Ưu điểm của mạng máy tính đã được thể hiện khá rõ trong mọi lĩnh vực của cuộc sống. Đó chính là sự trao đổi, chia sẻ, lưu trữ và bảo vệ thông tin. Bên cạnh nền tảng mạng máy tính hữu tuyến, mạng máy tính không dây ngay từ khi ra đời đã thể hiện nhiều ưu điểm nổi bật về độ linh hoạt, tính giản đơn, khả năng tiện dụng. Trước đây, do chi phí còn cao nên mạng không dây còn chưa phổ biến, ngày nay khi mà giá thành thiết bị phần cứng ngày một hạ, khả năng xử lý ngày càng tăng thì mạng không dây đã được triển khai rộng rãi, ở một số nơi đã thay thế được mạng máy tính có dây khó triển khai.

Do đặc điểm trao đổi thông tin trong không gian truyền sóng nên khả năng truyền dữ liệu trong mạng không dây phải là ưu tiên hàng đầu. Vì vậy có thể nói việc truyền dữ liệu không dây được gửi đi và để nơi nhận nhận chính xác dữ liệu đã gửi là một yêu cầu cơ bản của mạng máy tính không dây. Chính vì vậy em đã quyết định chọn đề tài **"Tăng khả năng thành công truyền dữ liệu trong mạng không dây bằng phương pháp mã hóa dữ liệu"** làm đề tài tốt nghiệp, với mong muốn có thể tìm hiểu, nghiên cứu, hiểu biết thêm đề tài này.

Em xin chân thành cảm ơn **PGS.TS. Vương Đạo Vy** đã giúp đỡ em nhiệt tình trong suốt quá trình làm đồ án cũng như xin được cảm ơn bạn bè đã góp ý, giúp đỡ em hoàn thành đồ án này. Vì đây là đề tài khá mới, nguồn tài liệu chủ yếu là Tiếng Anh nên đồ án này chắc chắn sẽ không tránh được những sai sót, em rất mong nhận được những ý kiến đóng góp của thầy cô và các bạn.

Hải Phòng, tháng 07/2010

Nguyễn Thị Hồng Hạnh

Mục lục

CHƯƠNG 1: TRUYỀN DỮ LIỆU

1. TỔNG QUAN VỀ MẠNG KHÔNG DÂY:

1.1 Thế nào là mạng máy tính không dây ?

1.1.1 Giới thiệu:

Thuật ngữ "mạng máy tính không dây" nói đến công nghệ cho phép hai hay nhiều máy tính giao tiếp với nhau dùng những giao thức mạng chuẩn nhưng không cần dây cáp mạng. Nó là một hệ thống mạng dữ liệu linh hoạt được thực hiện như một sự mở rộng hoặc một sự lựa chọn mới cho mạng máy tính hữu tuyến (hay còn gọi là mạng có dây). Các mạng máy tính không dây sử dụng các sóng điện từ không gian (sóng vô tuyến hoặc sóng ánh sáng) thu, phát dữ liệu qua không khí, giảm thiểu nhu cầu về kết nối bằng dây. Vì vậy, các mạng máy tính không dây kết hợp liên kết dữ liệu với tính di động của người sử dụng.

Công nghệ này bắt nguồn từ một số chuẩn công nghiệp như là IEEE 802.11 đã tạo ra một số các giải pháp không dây có tính khả thi trong kinh doanh, công nghệ chế tạo, các trường đại học… khi mà ở đó mạng hữu tuyến là không thể thực hiện được. Ngày nay, các mạng máy tính không dây càng trở nên quen thuộc hơn, được công nhận như một sự lựa chọn kết nối đa năng cho một phạm vi lớn các khách hàng kinh doanh.

1.1.2 Ưu điểm của mạng máy tính không dây:

Mạng máy tính không dây đang nhanh chóng trở thành một mạng cốt lõi trong các mạng máy tính và đang phát triển vượt trội. Với công nghệ này, những người sử dụng có thể truy cập thông tin dùng chung mà không phải tìm kiếm chỗ để nối dây mạng, chúng ta có thể mở rộng phạm vi mạng mà không cần lắp đặt hoặc di chuyển dây. Các mạng máy tính không dây có ưu điểm về hiệu suất, sự thuận lợi, cụ thể như sau:

- Tính di động : những người sử dụng mạng máy tính không dây có thể truy nhập nguồn thông tin ở bất kỳ nơi nào. Tính di động này sẽ tăng năng suất và tính kịp thời thỏa mãn nhu cầu về thông tin mà các mạng hữu tuyến không thể có được.

- Tính đơn giản : lắp đặt, thiết lập, kết nối một mạng máy tính không dây là rất dễ dàng, đơn giản và có thể tránh được việc kéo cáp qua các bức tường và trần nhà.

- Tính linh hoạt : có thể triển khai ở những nơi mà mạng hữu tuyến không thể triển khai được.

- Tiết kiệm chi phí lâu dài : Trong khi đầu tư cần thiết ban đầu đối với phần cứng của một mạng máy tính không dây có thể cao hơn chi phí phần cứng của một mạng hữu tuyến nhưng toàn bộ phí tổn lắp đặt và các chi phí về thời gian tồn tại có thể thấp hơn đáng kể. Chi phí dài hạn có lợi nhất trong các môi trường động cần phải di chuyển và thay đổi thường xuyên.

- Khả năng vô hướng : các mạng máy tính không dây có thể được cấu hình theo các topo khác nhau để đáp ứng các nhu cầu ứng dụng và lắp đặt cụ thể. Các cấu hình dễ dàng thay đổi từ các mạng ngang hàng thích hợp cho một số lượng nhỏ người sử dụng đến các mạng có cơ sở hạ tầng đầy đủ dành cho hàng nghìn người sử dụng mà có khả năng di chuyển trên một vùng rộng.

1.1.3 Hoạt động của mạng máy tính không dây:

Các mạng máy tính không dây sử dụng các sóng điện từ không gian (vô tuyến hoặc ánh sáng) để truyền thông tin từ một điểm tới điểm khác. Các sóng vô tuyến thường được xem như các sóng mang vô tuyến do chúng chỉ thực hiện chức năng cung cấp năng lượng cho một máy thu ở xa. Dữ liệu đang được phát được điều chế trên sóng mang vô tuyến (thường được gọi là điều chế sóng mang nhờ thông tin đang được phát) sao cho có thể được khôi phục chính xác tại máy thu.

Nhiễu sóng mang vô tuyến có thể tồn tại trong cùng không gian, tại cùng thời điểm mà không can nhiễu lẫn nhau nếu các sóng vô tuyến được phát trên các tần số vô tuyến khác nhau. Để nhận lại dữ liệu, máy thu vô tuyến sẽ thu trên tần số vô tuyến của máy phát tương ứng.

Trong một cấu hình mạng máy tính không dây tiêu chuẩn, một thiết bị thu/phát (bộ thu/phát) được gọi là một điểm truy cập, nối với mạng hữu tuyến từ một vị trí cố định sử dụng cáp tiêu chuẩn. Chức năng tối thiểu của điểm truy cập là thu, làm đệm, và phát dữ liệu giữa mạng máy tính không dây và cơ sở hạ tầng mạng hữu tuyến. Một điểm truy cập đơn có thể hỗ trợ một nhóm nhỏ người sử dụng và có thể thực hiện chức

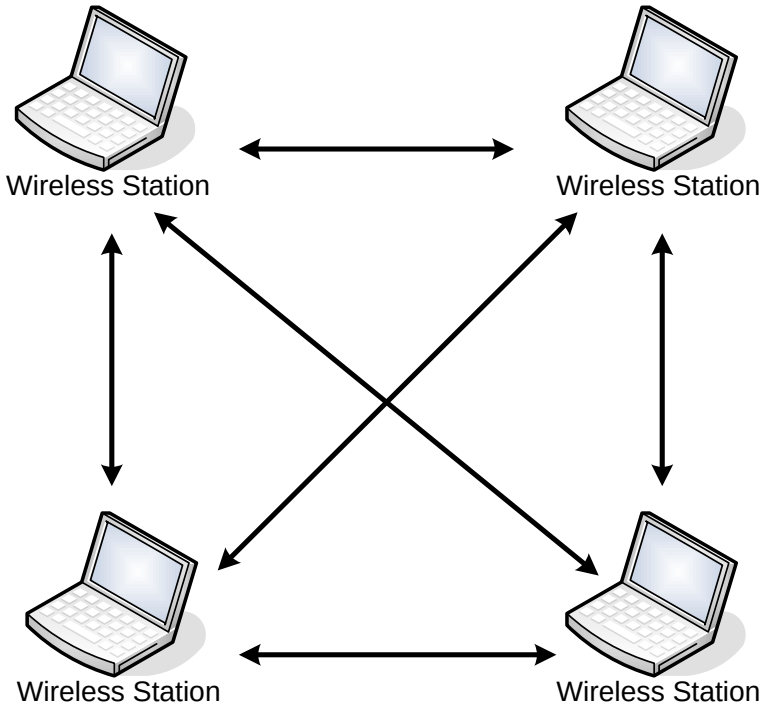
năng trong một phạm vi từ một trăm đến vài trăm feet. Điểm truy cập (hoặc anten được gắn vào điểm truy cập) thường được đặt cao nhưng về cơ bản có thể được đặt ở bất kỳ chỗ nào miễn là đạt được vùng phủ sóng mong muốn.

Những người sử dụng truy cập vào mạng máy tính không dây thông qua các bộ thích ứng máy tính không dây như các Card mạng không dây trong các vi máy tính, các máy Palm, PDA. Các bộ thích ứng máy tính không dây cung cấp một giao diện giữa hệ thống điều hành mạng (NOS – Network Operation System) của máy khách và các sóng không gian qua một anten. Bản chất của kết nối không dây là trong suốt đối với hệ điều hành mạng.

1.1.4 Các mô hình của mạng máy tính không dây cơ bản:

- ***Kiểu Ad – hoc:***

Mỗi máy tính trong mạng giao tiếp trực tiếp với nhau thông qua các thiết bị card mạng không dây mà không dùng đến các thiết bị định tuyến hay thu phát không dây.



Hình I.1: Mô hình mạng Ad – hoc (hay mạng ngang hàng)

- ***Kiểu Infrastructure***

Các máy tính trong hệ thống mạng sử dụng một hoặc nhiều các thiết bị định tuyến hay thiết bị thu phát để thực hiện các hoạt động trao đổi dữ liệu với nhau và các hoạt động khác.

1.1.5 Cự ly truyền sóng, tốc độ truyền dữ liệu:

Truyền sóng điện từ trong không gian sẽ gặp hiện tượng suy hao. Vì thế đối với kết nối không dây nói chung, khoảng cách càng xa thì khả năng thu tín hiệu càng kém, tỷ lệ lỗi sẽ tăng lên, dẫn đến tốc độ truyền dữ liệu sẽ phải giảm xuống.

Các tốc độ của chuẩn không dây như 11 Mbps hay 54 Mbps không liên quan đến tốc độ kết nối hay tốc độ download, vì những tốc độ này được quyết định bởi nhà cung cấp dịch vụ Internet.

Với một hệ thống mạng không dây, dữ liệu được giử qua sóng radio nên tốc độ có thể bị ảnh hưởng bởi các tác nhân gây nhiễu hoặc các vật thể lớn. Thiết bị định tuyến không dây sẽ tự động điều chỉnh xuống các mức tốc độ thấp hơn. (Ví dụ như là từ 11 Mbps sẽ giảm xuống còn 5,5 Mbps và 2 Mbps hoặc thậm chí là 1 Mbps).

1.2 Ứng dụng:

Trong những năm gần đây, các nghiên cứu về WSN đã đạt được bước phát triển mạnh mẽ, các bước tiến từ các nghiên cứu hứa hẹn tác động lớn đến các ứng dụng rộng rãi trong các lĩnh vực an ninh quốc gia, chăm sóc sức khỏe, môi trường, năng lượng, an toàn thực phẩm và sản xuất...

Các ứng dụng của mạng WSN thực sự chỉ bị giới hạn bởi sự tưởng tượng của con người. Sau đây là các ứng dụng phổ biến nhất của WSN:

1.2.1 Ứng dụng quân sự, an ninh và thiên nhiên:

Trong phản ứng với dịch bệnh, thảm họa thiên nhiên lượng lớn các cảm biến được thả từ trên không, mạng lưới các cảm biến sẽ cho biết vị trí người sống sót, vùng nguy hiểm, giúp cho người giám sát có các thông tin chính xác đảm bảo hiệu quả và an toàn cho các hoạt động tìm kiếm.

Sử dụng mạng WSN hạn chế sự có mặt trực tiếp của con người trong môi trường nguy hiểm . Ứng dụng an ninh bao gồm phát hiện xâm nhập và truy bắt tội phạm.

Mạng cảm biến quân sự phát hiện và có được thông tin về sự di chuyển của đối phương, chất nổ và các thông tin khác.

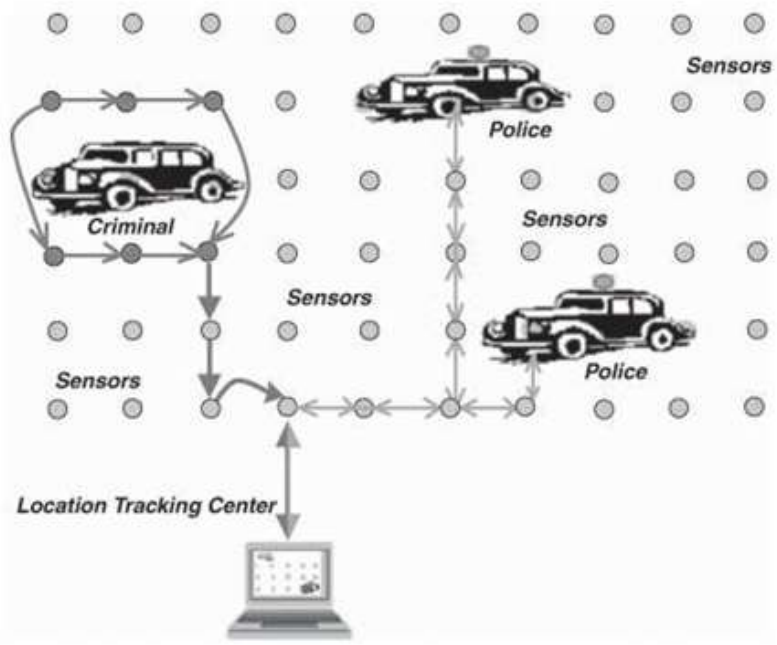
Phát hiện và phân loại các chất hóa chất, sinh hóa, sóng vô tuyến, phóng xạ hạt nhân, chất nổ…

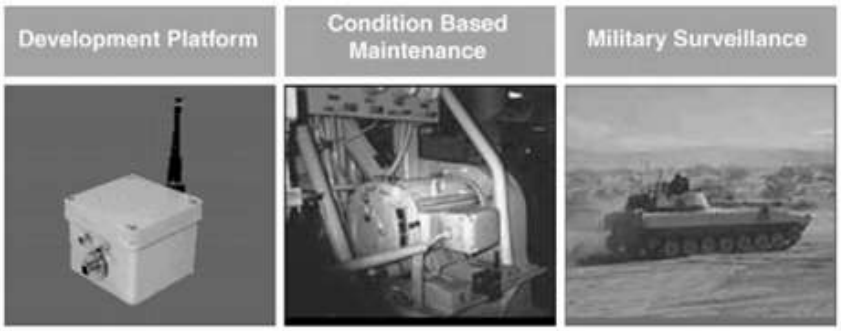
Giám sát sự thay đổi khí hậu, rừng, biển….

Giám sát xe cộ trên đường.

Giám sát an ninh trong các khu vực dân cư, thương mại…

Theo dõi biên giới kết hợp với vệ tinh…





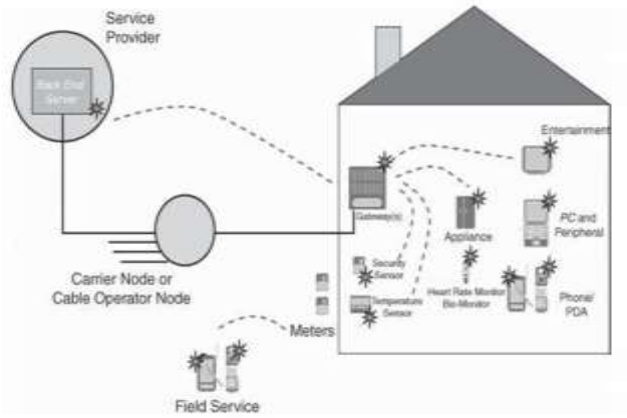
Hình I.2: Ứng dụng WSN trong an ninh quốc gia và luật pháp

1.2.2 Ứng dụng trong giám sát xe cộ và thông tin liên quan:

Mục tiêu của các hệ thống này là thu thập thông tin thông qua các mạng cảm biến, xử lý và lưu trữ dữ liệu tại trung tâm, sử dụng dữ liệu đó cho các ứng dụng cần thiết. Hệ thống được lắp đặt dọc theo các đường chính, mạng cảm biến số tập hợp dữ liệu về tốc độ lưu thông, mật độ xe, số lượng xe trên đường. Dữ liệu sau đó được truyền đến trung tâm dữ liệu để xử lý. Mạng theo dõi liên tục, cung cấp thông tin cập nhật thường xuyên theo thời gian thực. Các thông tin thu được dùng để giám sát lưu lượng , điều phối giao thông hoặc cho các mục đích khác.

1.2.3 Điều khiển các thiết bị trong nhà:

Điều khiển các thiết bị trong nhà:



Hình I.3: Các ứng dụng điều khiển

Các node cảm biến được lắp trên các thiết bị , vị trí cần thiết, sau đó kết nối thành mạng truyền dữ liệu về node trung tâm. Một khả năng có thể phát triển là các cảm biến theo dõi y tế được gắn trực tiếp lên cơ thể người bệnh để đo đạc thường xuyên các thông số về huyết áp, nhịp tim,…

1.2.4 Các tòa nhà tự động:

Ứng dụng cung cấp khả năng điều khiển, quản lý, tạo sự tiện lợi trong kiểm soát, an ninh…Quản lý nhiều hệ thống cùng lúc, hệ thống chiếu sáng, nhiệt độ, an ninh, giám sát nhân viên, quản lý hiệu quả tiêu thụ năng lượng trong tòa nhà, gắn các chip lên hàng hóa,giảm được thời gian kiểm tra…có thể dễ dàng được thực hiện bằng

C2WSNs và công nghệ ZigBee. đặc điểm nổi bật là dùng các công nghệ microsensor tiêu thụ rất ít công suất, thu phát vô tuyến , kỹ thuật liên lạc vả cảm biến không dây đa chức năng.

1.2.5 Quản lý quá trình tự động trong công nghiệp:

Các ứng dụng trong sản xuất công nghiệp gồm điều khiển, quản lý, hiệu suất và an toàn. Các cảm biến đặt trong môi trường làm việc giám sát quá trình sản xuất, chất lượng sản phẩm, kiểm soát môi trường làm việc, quản lý nhân viên,…dữ liệu được đưa về trung tâm để người quản lý có thể đưa ra các quyết định kịp thời.

1.2.6 Các ứng dụng trong y học:

Một số bệnh viện và trung tâm y tế đang ứng dụng công nghệ WSNs vào tiền chẩn đoán, chăm sóc sức khỏe, đối phó với các dịch bệnh và phục hồi chức năng cho người bệnh. WSNs cho phép theo dõi tình trạng của các bệnh nhân kinh niên ngay tại nhà, làm cho việc phân tích và điều trị thuận tiện hơn, rút ngắn thời gian điều trị tại bệnh viện. WSNs còn cho phép thu thập thông tin y tế qua thời gian dài thành các cơ sở dữ liệu quan trọng, các biện pháp can thiệp hiệu quả.

2. TRUYỀN DỮ LIỆU GIỮA CÁC NÚT MẠNG, NGUYÊN TẮC:

Một nút cảm nhận không dây giao tiếp với các nút cảm nhận khác bằng việc sử dụng net work - stack.

- Ở lớp ứng dụng (Application), dữ liệu có thể được gửi ở dạng gói, do chip radio ở lớp phần cứng có thể truyền và nhận dữ liệu dạng byte nên các gói tin này cần được phân đoạn trước khi chúng được gửi và phải được ghép lại sau khi chúng được nhận.

Lớp điều khiển đa truy cập Media - Access - Control (MAC) chuyển tiếp lớp ứng dụng với radio chip:

+ Khi 1 gói tin được gửi đi nhờ 1 ứng dụng, gói tin được phân nhỏ thành các byte. 1 chuỗi liên tiếp các byte đặc biệt được gọi là Preamble (phần mở đầu) được gửi đi trước các byte dữ liệu do đó phía thu có thể đồng bộ và xác định phần bắt đầu của 1 gói tin.

+ Sau khi nhận 1 byte, radio chip mã hóa bit dữ liệu và truyền chúng. Ở phía nhận, radio chip báo hiệu việc đến của các byte sau khi xác định và giải mã cácbit dữ liệu.

+ Sau đó lớp MAC hợp lại gói tin bắt đầu từ gói đầu tiên sau gói preamble. Tiếp theo lớp MAC báo hiệu việc đến của gói tin cho lớp trên.

Các byte dữ liệu có thể được mã hóa một cách tùy ý sau khi được phân đoạn cùng với mã sửa lỗi (ECC). Mã sửa lỗi ECC được dùng để khôi phục các bit dữ liệu trong trường hợp có số lượng nhỏ bit lỗi. Khi các byte dữ liệu đó tới bộ thu, tại đây nó được giải mã để trở về dạng các byte dữ liệu ban đầu.

3. TIẾT KIỆM VÀ TIÊU THỤ NĂNG LƯỢNG:

Năng lượng tiêu thụ phụ thuộc vào nhiều yếu tố sử dụng khác nhau như: Kiến trúc giao thức mạng, Giao thức chọn đường, Hoạt động truyền nhận không dây.

3.1 Kiến trúc giao thức mạng:

Kiến trúc giao thức này kết hợp giữa năng lượng và chọn đường, kết hợp số liệu với các giao thức mạng, sử dụng năng lượng hiệu quả. Kiến trúc giao thức bao gồm: lớp vật lý, lớp liên kết số liệu, lớp mạng, lớp truyền tải, lớp ứng dụng, phần quản lý công suất, phần quản lý di động và phần quản lý nhiệm vụ.

Lớp vật lý cung cấp các kỹ thuật điều chế, phát và thu. Vì môi trường có tạp âm và các nút cảm biến có thể di động, giao thức điều khiển truy cập môi trường (MAC) phải xét đến vấn đề công suất và phải có khả năng tối thiểu hoá việc va chạm với thông tin quản bá của các nút lân cận.

Lớp mạng quan tâm đến việc chọn đường số liệu được cung cấp bởi lớp truyền tải.

Lớp truyền tải giúp duy trì luồng số liệu nếu ứng dụng mạng cảm biến yêu cầu. Tuỳ theo nhiệm vụ cảm biến, các loại phần mềm khác nhau có thể được xây dựng và sử dụng ở lớp ứng dụng. Ngoài ra, các phần quản lý công suất, di chuyển và nhiệm vụ sẽ giám sát việc sử dụng công suất, sự di chuyển và thực hiện nhiệm vụ giữa các nút cảm biến. Những phần này giúp các nút cảm biến phối hợp nhiệm vụ cảm biến và tiêu thụ năng lượng tổng thể thấp hơn.

Phần quản lý năng lượng điều khiển việc sử dụng năng lượng của nút mạng. Ví dụ, nút mạng có thể tắt khối thu của nó sau khi thu được một bản tin từ một nút lân cận. Điều này giúp tránh tạo ra các bản tin giống nhau. Cũng vậy, khi mức năng lượng của nút mạng thấp, nó sẽ phát quảng bá tới các nút lân cận để thông báo nó có mức

năng lượng thấp và không thể tham gia vào các bản tin chọn đường. Phần năng lượng còn lại sẽ dành riêng cho nhiệm vụ cảm biến.

Phần quản lý di chuyển phát hiện và ghi lại sự di chuyển của các nút cảm biến để duy trì tuyến tới người sử dụng và các nút cảm biến có thể lưu vết của các nút cảm biến lân cận. Nhờ xác định được các nút cảm biến lân cận, các nút cảm biến có thể cân bằng giữa cân bằng giữa công suất của nó và nhiệm vụ thực hiện.

Phần quản lý nhiệm vụ dùng để làm cân bằng và lên kế hoạch các nhiệm vụ cảm biến trong một vùng xác định. Không phải tất cả các nút cảm biến trong vùng đó đều phải thực hiện nhiệm vụ cảm biến tại cùng một thời điểm. Kết quả là một số nút cảm biến thực hiện nhiều hơn các nút khác tuỳ theo mức công suất của nó. Những phần quản lý này là cần thiết để các nút cảm biến có thể làm việc cùng nhau theo một cách thức sử dụng hiệu quả công suất, chọn đường số liệu trong mạng cảm biến di động và phân chia tài nguyên giữa các nút cảm biến.

Kiến trúc mạng như trên góp phần quản lý năng lượng của các nút mạng đồng thời duy trì hoạt động của toàn mạng trong thời gian dài hơn.

3.2 Giao thức chọn đường:

3.2.1 Khó khăn trong giao thức chọn đường:

Mặc dù các ứng dụng của mạng WSN là rất lớn, tuy nhiên những mạng này có một số hạn chế như: giới hạn về nguồn công suất, khả năng tính toán và độ rộng băng thông. Một số giao thức chọn đường, quản lý công suất và trao đổi số liệu đã được thiết kế cho WSN với yêu cầu quan trọng nhất là tiết kiệm được năng lượng.

Một trong những mục tiêu thiết kế chính của WSN là kéo dài thời gian sống của mạng và tránh suy giảm kết nối nhờ các kỹ thuật quản lý năng lượng. Vấn đề này cần được giải quyết triệt để thì mới đạt được hiệu quả truyền tin trong WSN. Các giao thức chọn đường trong WSN có thể khác nhau tuỳ theo ứng dụng và cấu trúc mạng. Tuy nhiên, việc chọn đường gặp phải những khó khăn như: Phân bố nút, tiêu thụ năng lượng không được làm mất độ chính xác, tính không đồng nhất của nút mạng, tính động của mạng, khả năng định cỡ, khả năng chống lỗi, chất lượng dịch vụ.

Phân bố nút: Việc phân bố nút trong WSN phụ thuộc vào ứng dụng và có thể được thực hiện bằng tay hoặc phân bố ngẫu nhiên. Khi phân bố bằng tay, số liệu được

chọn đường thông qua các đường đã định trước. Tuy nhiên, khi phân bố các nút ngẫu nhiên sẽ tạo ra một cấu trúc chọn đường đặc biệt (Ad-hoc).

Tiêu thụ năng lượng không được làm mất độ chính xác: Các nút cảm biến có thể sử dụng quá các giới hạn về công suất để thực hiện tính toán và truyền tin trong môi trường vô tuyến. Thời gian sống của nút mạng cảm biến phụ thuộc rất nhiều vào thời gian sử dụng của pin. Trong WSN đa bước nhảy, mỗi nút đóng hai vai trò là truyền số liệu và chọn đường. Một số nút cảm biến hoạt động sai chức năng do lỗi nguồn công suất có thể gây ra sự thay đổi cấu hình mạng nghiêm trọng và phải chọn đường lại các gói hoặc phải tổ chức lại mạng.

Tính không đồng nhất của nút mạng: Nghĩa là các nút mạng có khả năng tính toán, khả năng truyền tin và có công suất khác nhau. Khi đó sẽ gây khó khăn cho kỹ thuật chọn đường.

Khả năng chống lỗi: Một số nút cảm biến có thể bị lỗi hoặc bị ngắt do thiếu công suất, hỏng phần cứng hoặc bị nhiễu môi trường. Sự cố của các nút cảm biến không được ảnh hưởng tới nhiệm vụ của toàn mạng cảm biến. Nếu có nhiều nút bị lỗi, các giao thức chọn đường và điều khiển truy cập môi trường (MAC) phải thành lập các tuyến mới tới nút gốc. Việc này có thể cần thiết phải điều chỉnh công suất phát và tốc độ tín hiệu trên các tuyến hiện tại để giảm sự tiêu thụ năng lượng hoặc là các gói phải chọn đường lại qua các các vùng mạng có công suất khả dụng lớn hơn.

Khả năng định cỡ: Số lượng nút mạng có thể là hàng trăm, hàng nghìn hoặc nhiều hơn. Bất kỳ phương pháp chọn đường nào cũng phải có khả năng làm việc với một số lượng lớn các nút cảm biến như vậy.

Tính động của mạng: Trong nhiều nghiên cứu, các nút cảm biến được coi là cố định. Tuy nhiên, trong một số ứng dụng, cả nút gốc và các nút cảm biến có thể di chuyển. Khi đó, các bản tin chọn đường từ hoặc tới các nút di chuyển sẽ gặp phải các vấn đề như: đường liên lạc, cấu hình mạng, năng lượng, độ rộng băng,…

Chất lượng dịch vụ: Khi năng lượng gần hết, các nút mạng có thể yêu cầu giảm chất lượng các kết quả để giảm mức tiêu thụ năng lượng của nút và kéo dài thời gian sống của toàn mạng.

Mặc dù việc chọn đường gặp nhiều khó khăn nhưng khi có một giao thức chọn đường tốt sẽ tiết kiệm thời gian truyền nhận thông tin, giảm năng lượng tiêu hao lãng phí do tắc nghẽn, lỗi đường truyền,..Vì vậy, người ta đã đưa ra một số giao thức chọn đường mà dưới đây sẽ trình bầy.

3.2.2 Các giao thức chọn đường:

Phần này sẽ trình bày về phương pháp chọn đường trong WSN. Nói chung, các giao thức chọn đường được chia làm 3 loại dựa vào cấu trúc mạng: ngang hàng, phân cấp hoặc dựa vào vị trí.

Trong chọn đường ngang hàng, tất cả các nút thường có vai trò hoặc chức năng như nhau như: cảm nhận, truyền,... Chúng sẽ thực hiện việc cảm nhận thông tin cần thiết và truyền số liệu về nút gốc.

Trong chọn đường phân cấp, các nút sẽ đóng vai trò khác nhau trong mạng, sẽ có nút chủ đại diện cho từng nhóm các nút cảm biến để nhận số liệu từ các nút cảm biến truyền tới. Các nút chủ cũng tập hợp số liệu từ các nút trong nhóm của nó trước khi gửi số liệu đến nút gốc. Nút chủ sẽ thay đổi khi bắt đầu chu kỳ làm việc mới và sẽ thay bằng nút khác có khả năng đảm nhận chức năng này.

Trong chọn đường dựa theo vị trí thì vị trí của các nút cảm biến sẽ được dùng để chọn đường số liệu. Giao thức chọn đường còn có thể được phân loại dựa theo đa đường, yêu cầu, hỏi/đáp, QoS và liên kết tuỳ thuộc vào chế độ hoạt động. Ngoài ra, cũng có thể chia thành 3 loại: chủ động, tương tác hoặc lai ghép tuỳ thuộc vào cách thức mà nguồn tìm đường tời đích. Một giao thức chọn đường được coi là thích ứng nếu các tham số của hệ thống có thể điều khiển được để phù hợp với các trạng thái mạng hiện tại và các mức năng lượng khả dụng. Một số giao thức chọn đường đã phát huy hiệu quả tiết kiệm tiêu thụ năng lượng như: LEACH, TEEN&APTEEN, MECN&SVECN, PEGASIS thuộc chọn đường phân cấp; SPIN, Directed Diffusion, CADR, CUGAR thuộc chọn đường ngang hàng. Trong đó đáng quan tâm nhất là giao thức chọn đường LEACH. LEACH (Low Energy Adaptive Clustering Hierarchy) - phân cấp nhóm thích ứng công suất thấp, giao thức này cho phép tiết kiệm năng lượng trong mạng WSN. Mục đích của LEACH là lựa chọn ngẫu nhiên các nút cảm biến làm các nút chủ, do đó, việc tiêu hao năng lượng khi liên lạc với nút gốc được trải đều cho tất cả các nút cảm biến trong mạng. Quá trình hoạt động

của LEACH được chia thành 2 bước: bước thiết lập và bước ổn định. Thời gian bước ổn định kéo dài hơn so với thời gian của bước thiết lập để giảm thiểu phần điều khiển. Tại bước thiết lập sẽ xác định nút mạng nào sẽ làm chủ. Tại bước ổn định, các nút cảm biến bắt đầu cảm nhận và truyền số liệu về nút chủ, nút chủ cũng tập hợp số liệu từ các nút trong nhóm trước khi gửi các số liệu này về nút gốc. Theo thử nghiệm mô phỏng giao thức LEACH của mạng WSN có 160 nút, phân bố đều, công suất ban đầu của nút là 3.0. Kết quả thu được là: khi truyền trực tiếp tới nút gốc, sau khoảng 470 chu kỳ thời gian sẽ kết thúc truyền tin; khi sử dụng giao thức LEACH, sau khoảng 685 chu kỳ thời gian mới kết thúc truyền tin. Kết quả này cho thấy đây là một phương pháp chọn đường phân cấp có khả năng tiết kiệm được năng lượng và kéo dài thời gian sống của mạng.

Tuy nhiên, cơ chế hoạt động của LEACH là lựa chọn số liệu được tập trung và thực hiện theo chu kỳ. Do đó, giao thức này chỉ thích hợp với yêu cầu giám sát liên tục bởi mạng cảm biến. Với ứng dụng mà người sử dụng không cần tất cả các số liệu ngay lập tức thì việc truyền số liệu theo chu kỳ là không cần thiết và có thể làm tiêu tốn năng lượng vô ích. Giao thức LEACH cần tiếp tục được cải tiến để khắc phục hạn chế.

3.3 Hoạt động truyền nhận không dây:

Đối với mạng cảm nhận không dây thì năng lượng trở thành một vấn đề được chú ý để duy trì hoạt động của hệ thống mạng vì các nút mạng độc lập, chỉ được cung cấp năng lượng từ một nguồn cố định (pin). Việc tiêu thụ năng lượng thấp là một nhu cầu quan trọng cho các hệ thống hoạt động bằng pin - không được cung cấp năng lượng thường xuyên. Khi có một dòng tiêu thụ thấp thì thời gian sống của chúng sẽ kéo dài thêm. Hệ thống tiêu thụ năng lượng thấp là mục tiêu cần đạt của mạng cảm nhận không dây sử dụng CC1010. Các vấn đề đưa ra trong phần này là cơ sở cho việc viết phần mềm tiết kiệm tiêu thụ năng lượng cho nút mạng cảm nhận không dây.

CHƯƠNG 2: MÃ SỬA LỖI

Mạng phải có khả năng truyền chính xác dữ liệu từ thiết bị này tới thiết bị khác. Một hệ thống không bảo đảm dữ liệu nhận giống như dữ liệu truyền là hệ thống không sử dụng được. Có nhiều nhân tố, như tạp nhiễu đường dây có thể làm hỏng một hoặc nhiều bit của khối dữ liệu đang truyền. Một hệ thống thực sự dùng được phải có cơ cấu để phát hiện và sửa các lỗi đó.

Việc phát hiện và sửa lỗi được trang bị ở lớp liên kết dữ liệu hoặc lớp giao vận tương ứng trong mô hình OSI.

1. CÁC CƠ CHẾ TRUYỀN DỮ LIỆU:

1.1. Truyền dữ liệu song song:

• Mỗi bit dùng một đường truyền riêng. Nếu có 8 bits được truyền đồng thời sẽ yêu cầu 8 đường truyền độc lập.

• Để truyền dữ liệu trên một đường truyền song song, một kênh truyền riêng được dùng để thông báo cho bên nhận biết khi nào dữ liệu có sẵn (clock signal).

• Cần thêm một kênh truyền khác để bên nhận báo cho bên gởi biết là đã sẵn sàng để nhận dữ liệu kế tiếp.

1.2. Truyền dữ liệu tuần tự:

• Tất cả các bit đều được truyền trên cùng một đường truyền, bit này tiếp theo sau bit kia.

• Không cần các đường truyền riêng cho tín hiệu đồng bộ và tín hiệu bắt tay (các tín hiệu này được mã hóa vào dữ liệu truyền đi).

• Vấn đề định thời (timing) đòi hỏi phải có cơ chế đồng bộ giữa bên truyền và bên nhận.

• 2 cách giải quyết:

- Bất đồng bộ: mỗi ký tự được đồng bộ bởi start và stop bit.

- Đồng bộ: mỗi khối ký tự được đồng bộ dùng cờ.

1.3. Truyền bất đồng bộ:

• Dữ liệu được truyền theo từng ký tự để tránh việc mất đồng bộ khi nhận được chuỗi bit quá dài:

- 5 đến 8 bits.

- Chỉ cần giữ đồng bộ trong một ký tự

- Tái đồng bộ cho mỗi ký tự mới

• Hành vi:

- Đối với dòng dữ liệu đều, khoảng cách giữa các ký tự là đồng nhất (bằng chiều dài của phần tử stop).

- Ở trạng thái rỗi, bộ thu phát hiện sự chuyển 1 đến 0.

- Lấy mẫu 7 khoảng kế tiếp (chiều dài ký tự).

- Đợi việc chuyển 1 đến 0 cho ký tự kế tiếp.

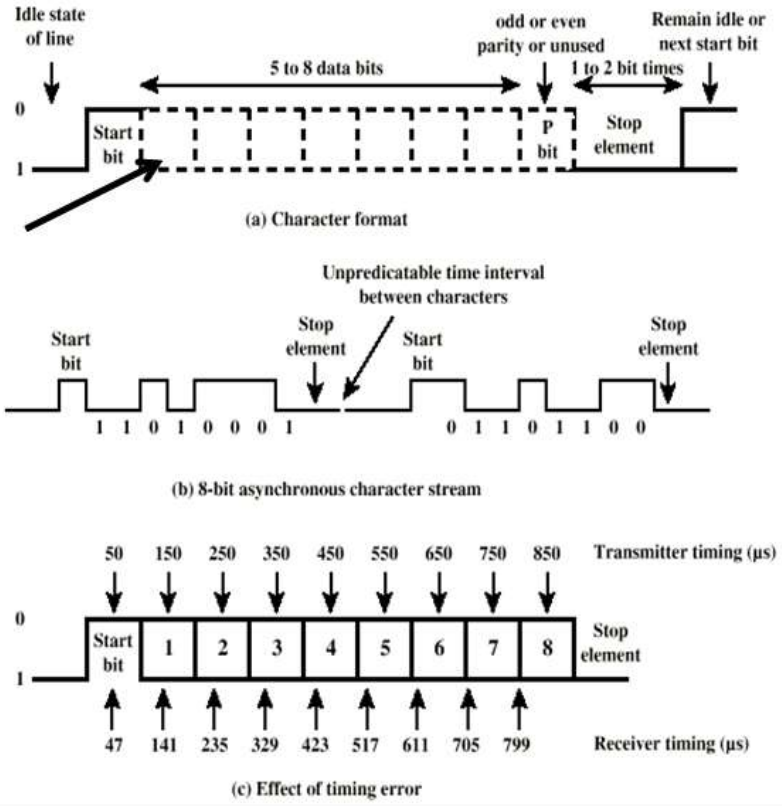
• Hiệu suất:

- Đơn giản

- Rẻ

- Phí tổn 2 hoặc 3 bit cho một ký tự (~20%)

- Thích hợp cho dữ liệu với khoảng trống giữa các ký tự lớn (dữ liệu nhập từ bàn phím).



• Đồng bộ khung (frame synchronization): dùng các ký tự điều khiển (STX, ETX, DLE).

1.4. Truyền đồng bộ:

- **Truyền không cần start/stop**
- **Phải có tín hiệu đồng bộ**
- **Đồng bộ bit (bit synchronization): sử dụng các phương pháp sau:**

- Tích hợp xung clock vào dữ liệu truyền đi:

+ Tích hợp thông tin đồng bộ (clock) vào trong dữ liệu truyền.

+ Đầu nhận sẽ tách thông tin đồng bộ dựa vào dữ liệu nhận được.

+ Manchester, differential Manchester, tần số sóng mang (analog).

- Sử dụng đường clock riêng:

+ Dùng một đường tín hiệu đồng bộ riêng biệt.

+ Một bên (phát hoặc nhận) tạo ra các xung clock đồng bộ với các bit truyền đi trên đường clock riêng.

+ Bên còn lại dùng tín hiệu trên đường clock riêng để làm clock.

+ Thích hợp khi truyền trong khoảng cách ngắn.

+ Tín hiệu đồng bộ dễ bị suy giảm trên đường truyền.

- **Đồng bộ frame:**

- Mỗi block dữ liệu được bắt đầu bằng một cờ gọi là preamble, kết thúc bằng một cờ gọi là postamble.

- Preamble và postamble là một mẫu bit (bit pattern) được quy định sẵn:

+ Một chuỗi các ký tự SYN (16h trong bảng mã ASCII)

+ Mẫu bit 11111110

- Frame: dữ liệu + preamble + postamble + thông tin điều khiển

- Hiệu quả hơn so với truyền bất đồng bộ (phí tổn thấp hơn cho các bit điều khiển):

+ HDLC: 48 bit điều khiển cho mỗi block 1000 ký tự (8000 bit)

2. VẤN ĐỀ XỬ LÝ LỖI

1.1 Các lỗi xẩy ra trên đường truyền:

• Môi trường truyền dẫn bị nhiễu (điện, từ, …) dẫn đến dữ liệu nhận có lỗi (các bit bị thay đổi).

• 2 cách khắc phục khi phát hiện có lỗi:

- Forward error control: thông tin sửa sai được thêm vào các ký tự hoặc các frame truyền đi, để bên nhận có thể phát hiện khi nào có lỗi va lỗi nằm ở đâu để sửa (có khả năng sửa lỗi).

- Feedback (backward) error control: thông tin sửa sai được thêm vào các ký tự hoặc các frame truyền đi chỉ đủ để phát hiện khi nào có lỗi (không có khả năng sửa lỗi). Cơ chế yêu cầu truyền lại ký tự/frame sai được dùng trong trường hợp này.

• Phân loại lỗi

- Lỗi 1 bit:

+ Chỉ 1 bit bị lỗi, không ảnh hưởng các bit xung quanh

+ Thường xảy ra do nhiễu trắng

- Lỗi chùm (busrt error):

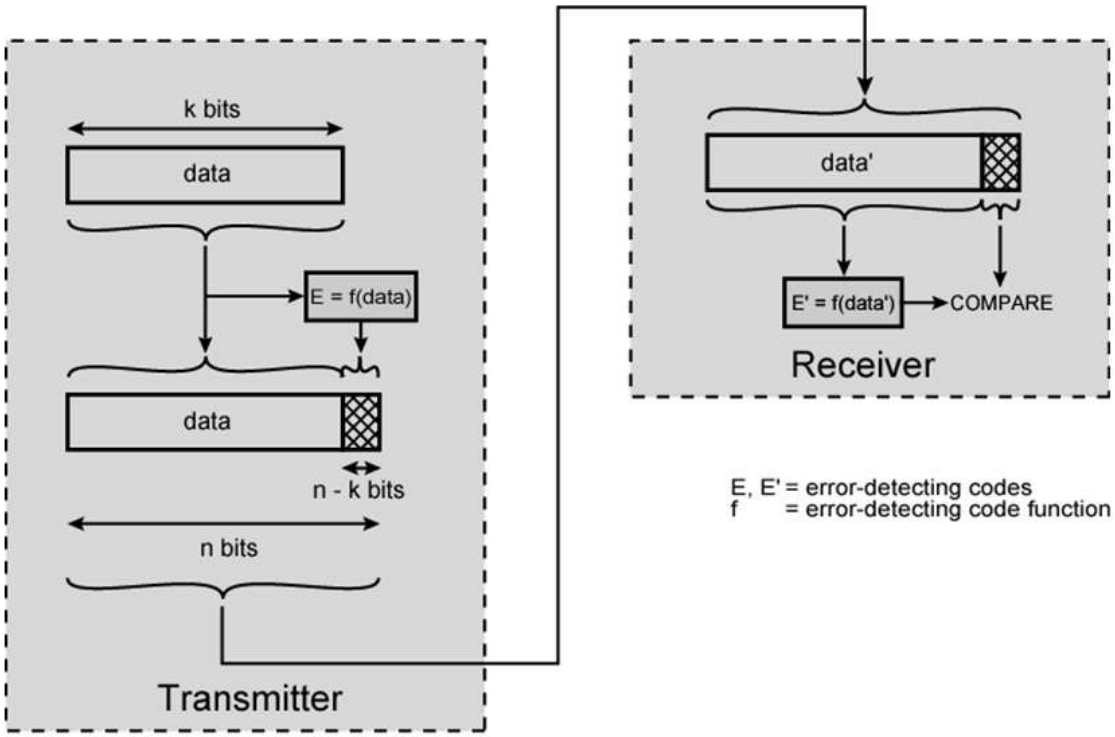
+ Một chuỗi liên tục B bit trong đó bit đầu, bit cuối và các bit bất kì nằm giữa chuỗi đều bị lỗi.

+ Thường xảy ra do nhiễu xung

+ Ảnh hưởng càng lớn đối với tốc độ truyền cao

• Bit error rate (BER): xác suất một bit nhận được bị lỗi

1.2 Cơ chế phát hiện lỗi:



1.3 Phát hiện lỗi bằng bit parity:

• 1 bit parity được thêm vào 1 khối dữ liệu cần truyền đi.

• Bit parity

- Parity chẵn: tổng số bit 1 có trong khối dữ liệu, kể cả bit parity, là số chẵn.

- Parity lẻ: tổng số bit 1 có trong khối dữ liệu, kể cả bit parity, là số lẻ.

Có đặc điểm:

- Chỉ dò được lỗi sai một số lẻ bit, không dò được lỗi sai một số chẵn bit

- Không sửa được lỗi

- Ít được dùng trong truyền dữ liệu đi xa, đặc biệt ở tốc độ cao.

1.4 Bộ mã phát hiện lỗi

Khi truyền tải một chuỗi các bit, các lỗi có thể phát sinh ra, bit 1 có thể biến thành bit 0 hay ngược lại.

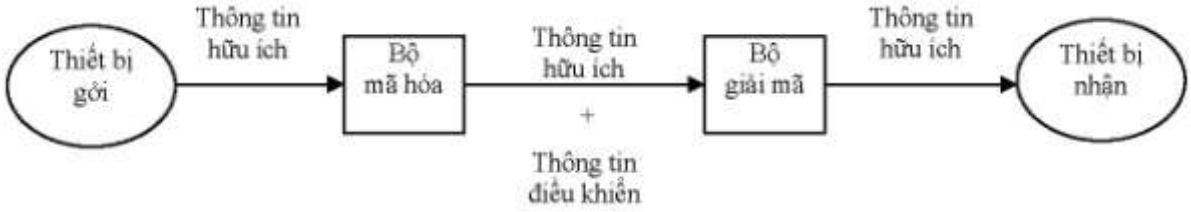
Ta định nghĩa tỷ lệ lỗi bởi tỷ số sau:

τ = Số bít bị lỗi / Tổng số bít được truyền

Tỷ lệ lỗi này có giá trị từ 10-5 đến 10-8. Tùy thuộc vào từng loại ứng dụng, một lỗi có mức độ nghiêm trọng khác nhau, chính vì thế cần có các cơ chế cho phép phát hiện lỗi cũng như sửa lỗi.

Các thống kê cho thấy rằng 88% các lỗi xẩy ra do sai lệch một bit và 10% các lỗi xảy ra do sự sai lệch 2 bit kề nhau. Chính vì thế ta ưu tiên cho vấn đề phát hiện các lỗi trên một bit và sửa đổi chúng một cách tự động.

Với ý tưởng như thế, ta sử dụng các mã phát hiện lỗi: bên cạnh các thông tin hữu ích cần truyền đi, ta thêm vào các thông tin điều khiển. Bên nhận thực hiện việc giải mã các thông tin điều khiển này để phân tích xem thông tin nhận được là chính xác hay có lỗi.



Hình 2.1: Mô hình xử lý lỗi trong truyền dữ liệu

Thông tin điều khiển được đưa vào có thể theo 2 chiến lược. Chiến lược thứ nhất gọi là bộ mã sửa lỗi (Error-correcting codes) và chiến lược thứ hai gọi là bộ mã phát hiện lỗi (Error-detecting codes). Bộ mã sửa lỗi cho phép bên nhận có thể tính toán và suy ra được các thông tin bị lỗi (sửa dữ liệu bị lỗi). Trong khi bộ mã phát hiện lỗi chỉ cho phép bên nhận phát hiện ra dữ liệu có lỗi hay không. Nếu có lỗi bên nhận sẽ yêu cầu bên gởi gởi lại thông tin. Với tốc độ của đường truyền ngày càng cao, người ta thấy rằng việc gởi lại một khung thông tin bị lỗi sẽ ít tốn kém hơn so với việc tính toán để suy ra giá trị ban đầu của các dữ liệu bị lỗi. Chính vì thế đa số các hệ thống mạng ngày nay đều chọn bộ mã phát hiện lỗi.

1.5 Những bộ mã phát hiện lỗi (Error-Detecting Codes)

Có nhiều sơ đồ phát hiện lỗi, trong đó có các sơ đồ thông dụng là:

- Kiểm tra chẵn lẻ (Parity checks)

- Kiểm tra thêm theo chiều dọc (Longitudinal reduncy check)

- Kiểm tra phần dư tuần hoàn (Cyclic redundancy check)

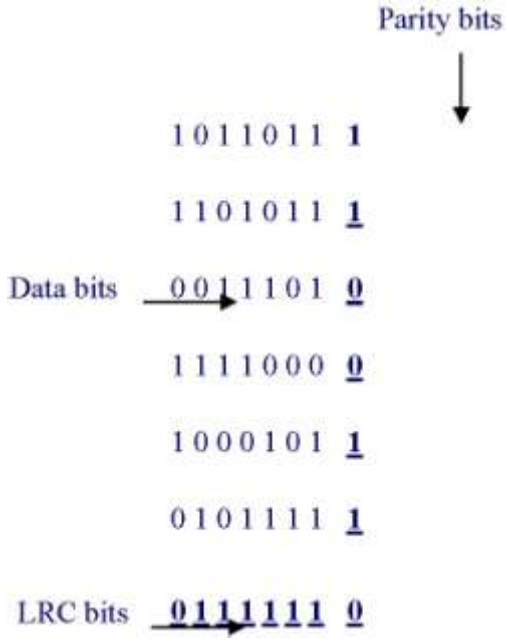
1.2.1 Kiểm tra chẵn lẻ (Parity Check):

Sơ đồ phát hiện bit lỗi đơn giản nhất là nối một bit chẵn-lẻ vào cuối của mỗi từ trong khung. Một ví dụ tiêu biểu là việc truyền các ký tự ASCII, mà trong đó một bit chẵn-lẻ được nối vào mỗi ký tự ASCII 7 bit. Giá trị của bit này được lựa chọn sao cho có một số chẵn của bit 1, với kiểm tra chẵn (even parity) hoặc một số lẻ của bit 1, với kiểm tra lẻ (odd parity).

Ví dụ, nếu bên gởi đang truyền một ký tự ASCII G (mã ASCII là1110001) và đang dùng phương pháp kiểm tra lẽ, nó sẽ nối một bit 1 và truyền đi 11100011. Bên nhận sẽ kiểm tra ký tự nhận được và nếu tổng của các bit 1 là lẻ, nó xem như không có lỗi. Nếu một bit hoặc một số lẻ bất kỳ các bit bị lỗi đảo ngược thì rõ ràng bên nhận sẽ phát hiện được lỗi. Tuy nhiên, nếu hai hay một số chẵn bất kỳ các bit bị lỗi đảo ngược thì nó sẽ không phát hiện được lỗi. Trên thực tế những xung nhiễu lại thường đủ dài để có thể phá hủy hơn một bit, đặc biệt là với tốc độ dữ liệu cao. Do đó, cần phải có một phương pháp cải thiện trường hợp này.

1.2.2 Kiểm tra thêm theo chiều dọc (Longitudinal Redundancy Check or Checksum)

Có thể cải thiện sơ đồ trên bằng cách dùng phương pháp LRC. Trong phương pháp này, khung được xem như một khối nhiều ký tự được sắp xếp theo dạng hai chiều, và việc kiểm tra được thực hiện cả chiều ngang lẫn chiều dọc.



Theo chiều ngang, mỗi ký tự được thêm vào một bit kiểm tra chẵn lẽ như trường hợp trên, và được gọi là bit Kiểm tra chiều ngang VRC (Vertical Redundancy Check).

Theo chiều dọc, cung cấp thêm một ký tự kiểm tra, được gọi là ký tự Kiểm tra chiều dọc LRC (Longitudinal Redundancy Check) hay Checksum. Trong đó, bít thứ i của ký tự này chính là bit kiểm tra cho tất cả các bit thứ i của tất cả các ký tự trong khối.

Các phép đo chỉ ra rằng việc dùng cả hai VRC và LRC giảm đi tỷ lệ lỗi không phát hiện được hai đến bốn bậc so với dùng chỉ VRC. Hãy xem trường hợp bit 1 và 3 trong ký tự 1 đang bị lỗi. Khi bên nhận tính toán được bit VRC cho ký tự 1, nó sẽ kiểm tra với bit VRC đã nhận, và sẽ không phát hiện được lỗi. Tuy nhiên, khi nó tính toán được ký tự LRC, bit 1 và 3 của ký tự này sẽ khác với những bit đó trong ký tự LRC nhận được, và sẽ phát hiện được lỗi.

Tuy nhiên, ngay sơ đồ này cũng không phải là thật sự tốt. Bây giờ, nếu giả sử bit 1 và 3 của ký tự 5 cũng bị lỗi, phương pháp này sẽ không phát hiện được điểm sai.

1.2.3 Kiểm tra phần dư tuần hoàn (Cyclic Redundancy Check)

Để cải tiến hơn nữa các nhà thiết kế đã dùng kỹ thuật mới dễ dàng và hiệu quả được gọi là kiểm tra phần dư tuần hoàn, trong đó có thể sử dụng một số phương pháp cài đặt khác nhau như: modulo 2, đa thức, thanh ghi dịch và các cổng Exclusive-or.

Các thủ tục với modulo 2 diễn ra như sau. Với một thông điệp M có k bit cần gởi đi, bên gởi sẽ nối vào cuối thông điệp một chuỗi F có r bit, được gọi là Chuỗi kiểm tra khung (FCS: Frame Check Sequence). Chuỗi kiểm tra khung sẽ tính toán sao cho khung kết quả T được hình thành từ việc nối M với F (gồm k + r bit) có thể chia hết bởi số P nào đó được định trước. Bên gởi sẽ gởi T đi. Khi bên nhận nhận được T, nó sẽ thực hiện phép chia modulo T cho P. Nếu phép chia không hết, tức có số dư, bên nhận xác định rằng khung T đã bị lỗi, ngược lại là không có lỗi. Nếu khung không có lỗi, bên nhận sẽ tách thông điệp M từ T, là k bits trọng số cao của T.

Phương pháp này dùng phép chia modulo 2 trong việc chia T cho P, phép toán modulo 2 dùng một phép cộng nhị phân không nhớ và đó cũng chính là phép toán Exclusive-or. Ví dụ sau mô tả phép toán cộng và nhân modulo 2:

```
      1          1
111         1001
      1          1
010         1
-------   -------
      0          1
101         1001

                11001
               -------
                    1
               01011
```

Giả sử ta có:

o M: Thông điệp k bit cần được gởi sang bên nhận.

o F : Chuỗi kiểm tra khung FCS gồm r bit là thông tin điều khiển được gởi theo M để giúp bên nhận có thể phát hiện được lỗi.

o T =MF là khung (k + r) bit, được hình thành bằng cách nối M và F lại với nhau. T sẽ được truyền sang bên nhận, với r < k

Với M (k bit) , P (r+1 bit), F (r bit), T (k+r bit), thủ tục tiến hành để xác định checksum F và tạo khung truyền như sau:

o Nối r bit 0 vào cuối M, hay thực hiện phép nhân M với 2r

o Dùng phép chia modulo 2 chia chuỗi bit M*2r cho P.

o Phần dư của phép chia sẽ được cộng với M*2r tạo thành khung T truyền đi.

o Trong đó P được chọn dài hơn F một bit, và cả hai bit cao nhất và thấp nhất phải là 1.

Ngoài ra người ta còn có thể sử dụng phương pháp đa thức để biểu diễn phương pháp kiểm tra phần dư tuần hòan. Trong phương pháp này người ta biểu diễn các chuỗi nhị phân dưới dạng những đa thức của biến x với các hệ số nhị phân. Các hệ số tương ứng với các bit trong chuỗi nhị phân cần biểu diễn.

Giả sử ta có M=110011và P = 11001, khi đó M và P sẽ được biểu diễn lại bằng 2 đa thức sau:

$M(x) = x^5 + x^4 + x + 1$

$P(x) = x^4 + x^3 + 1$

Những phép toán trên đa thức vẫn là modulo 2. Quá trình tính CRC được mô tả dưới dạng các biểu thức sau:

1. $\frac{X^n M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$

2. $T(X) = X^n M(X) + R(X)$

Các version thường được sử dụng của P là :

CRC-12 $= X^{12} + X^{11} + X^3 + X^2 + X + 1$

CRC-16 $= X^{16} + X^{15} + X^2 + 1$

CRC-CCITT $= X^{16} + X^{12} + X^5 + 1$

CRC-32 $= X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10}$

$+ X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Ví dụ:

Cho: M=1010001101, P=110101

Ta có: r=5

$M(x)= x^9+x^7+x^3+x^2+1$

$x^5M(x)= x^{14}+x^{12}+x^8+x^7+x^5$

$P(x) = x^5+x^4+x^2+1$

Thực hiện phép toán:

$$\frac{x^r M(x)}{P(x)} = Q(x) + \frac{R(x)}{P(x)}$$

=> $Q(x)= x^9+ x^8+ x^6+x^4+x^2 +x^1$

$F(x)= x^3+ x^2+ x^1$ <> 01110

Khung cần truyền đi là T= 101000110101110

CHƯƠNG 3: PHƯƠNG PHÁP TẠO RA TỪ MÃ

Truyền dữ liệu trong mạng không dây là do mạng không dây nuôi bằng nguồn nuôi độc lập cho nên tiết kiệm được năng lượng. Khi nghiên cứu một nút mạng người ta nhận thấy khi truyền đi là tốn nhiều năng lượng nhất ít hơn tí là nhận năng lượng, ít hơn nữa là quá trình xử lý và cuối cùng nghỉ là tốn ít năng lượng nhất.

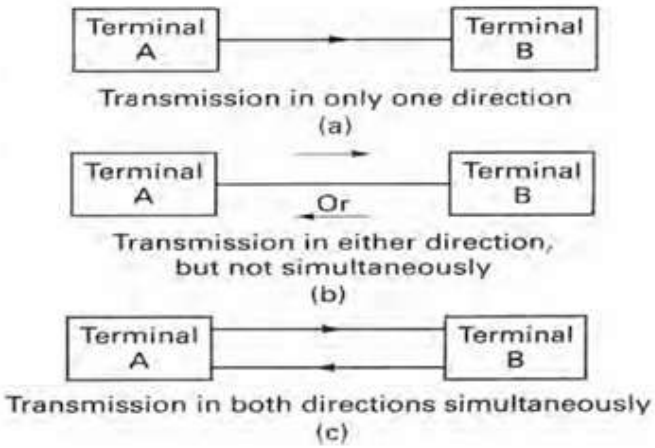
Truyền mạng WSN sai số lớn nhất khi truyền cáp quang là sai số ít cáp đồng, không dây là lỗi truyền dữ liệu là nhiều nhất. Vấn đề đặt ra khi bị lỗi thì nơi gửi phải truyền lại dữ liệu người ta đã nghĩ ra nếu nhận biết có lỗi khi truyền nên phải tự động sửa lỗi thì không tốn năng lượng.

Trong các lỗi có lỗi đơn bít, đa bit, đảo bít... tuy nhiên trong truyền không dây thì đa bit và đảo bit xuất hiện ít, chỉ xuất hiện đơn bit là nhiều. Trong nội dung nghiên cứu của chương này ta tập trung vào sửa lỗi đơn bit.

Muốn sửa lỗi thì nơi gửi phải tạo ra được các từ mã (từ mã gồm dữ liệu và bit dư thừa).

3.1 Phát hiện lỗi và truyền lại::

- Phát hiện lỗi và truyền lại (error detection & retransmission)
- Phát hiện lỗi và sửa lỗi FEC (Forward Error correction)



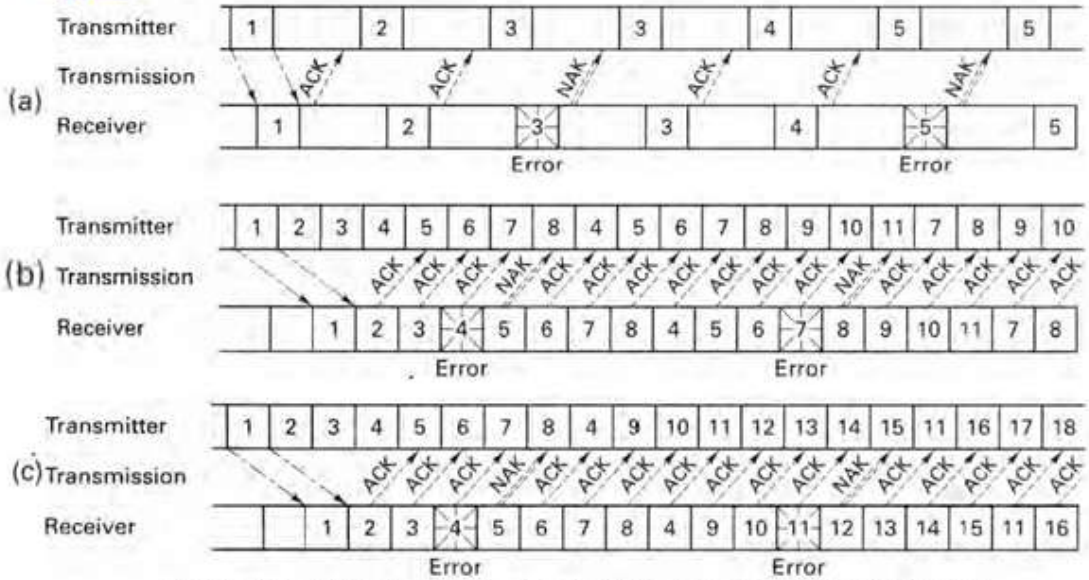


Figure 6.7 Automatic repeat request (ARQ). (a) Stop-and-wait ARQ (half-duplex). (b) Continuous ARQ with pullback (full-duplex). (c) Continuous ARQ with selective repeat (full-duplex).

3.2 Các phương pháp tạo ra từ mã:

Trong bất kỳ mạng nào đều có 2 phương pháp cơ bản để khôi phục lại các gói tin bị lỗi: Một là Yêu cầu trả lời tự động ARQ (Automatic Repeat Request) và một là sửa lỗi hướng trước FEC (Forward Erro Correction). Trong mạng cảm biến nguồn năng lượng là khan hiếm và chủ yếu được tiêu thụ bởi các quá trình truyền dẫn và thu nhận không dây mặt khác đặc tính lỗi của mạng WSN là các lỗi đơn bit và lỗi 2 bit nên sử dụng sửa lỗi hướng trước FEC là hiệu quả hơn cả.

FEC có khả năng sửa các lỗi chính: 1 loại mã có khả năng sửa lỗi đơn bit, một có khả năng sửa lỗi kép và 1 mã Reed Solomen, tất cả đều thuộc loại mã khối - block codes. Vì vậy ta sẽ tập trung nghiên cứu mã hóa khối tuyến tính.

3.1.1. Mã hóa khối tuyến tính:

Một mã khối có chiều dài n gồm nk từ mã được gọi là mã tuyến tính C(n,k) nếu và chỉ nếu 2k từ mã hình thành một không gian vectơ con k chiều của không gian vectơ n chiều gồm tất cả các vectơ n thành phần trên trường nhị phân GF (2).

Với trường GF(2) là trường nhị phân đồng thời phép cộng và phép cộng modul 2(cộng tuyệt đối) và phép và (AND).

1+1=0	1 and 1 = 1
1+0=1	1and 0 = 0
0+1=1	0 and 1 = 0
0+0=0	0 and 0 = 0

Mã tuyến tính C (n,k) có mục đích mã hóa những khối tin k bit thành những từ mã n bit.

- ***Cách mã hóa:***

Gọi u = [a_0, a_1, a_2,...., a_{k-1}] là thông tin cần được mã hóa thì từ mã v tương ứng với u sẽ có được bằng cách lấy u nhân với ma trận G.

Công thức: v = u.G hoặc v= a_0g_0+ a_1g_1+...+ $a_{k-1}g_{k-1}$

Vì các từ mã tương ứng với các thông báo được sinh ra bởi G theo cách trên nên G được gọi là ma trận sinh của bộ mã.

Xét ví dụ sau: Ta có một ma trận sinh của bộ mã tuyến tính C(7,4):

$$G_{4x7} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Nếu u = [1101] là thông tin cần mã hóa thì từ mã tương ứng là:

$$\mathbf{v = 1g_0 + 1g_1 + 0g_2 + 1g_3 = [1100101]}$$

Tóm tắt như sau:

- Bất kỳ k từ mã độc lập tuyến tính nào cũng có thể được dùng để làm ma trận sinh cho bộ mã.

- Một bộ mã tuyến tính có thể có nhiều ma trận sinh khác nhau cùng biểu diễn.

- Mỗi ma trận sinh tương ứng với một cách mã hóa khác nhau.

- ***Cách giải mã:***

Ta có thông tin u = [a_0, a_1, a_2,...., a_{k-1}].
Từ mã nhận được tại bộ thu v = [b_0, b_1, b_2,....,b_{k-1}, b_k,..., b_{n-1}]
Giả sử quá trình truyền tín hiệu không lỗi tức là v thu đúng thông tin u. Từ v ta tìm lại u như sau:
ta giải phương trình v = u.G hoặc v= $a_0g_0 + a_1g_1 + ... + a_{k-1}g_{k-1}$ để tìm a_i từ đó tìm được u.
Ta có một ma trận sinh của bộ mã tuyến tính C(7,4):

$$G_{4x7} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

u = [a_0, a_1, a_2,a_3]

v = [b_0, b_1, b_2, b_3, b_4, b_5, b_6]

v = u x G = hệ phương trình:

$b_0 = a_0 + a_1 + a_3$

$b_1 = a_0 + a2$

$b_2 = a_1 + a_3$

$b_3 = a_0 + a_1$

$b_4 = a_1$

$b_5 = a_2$

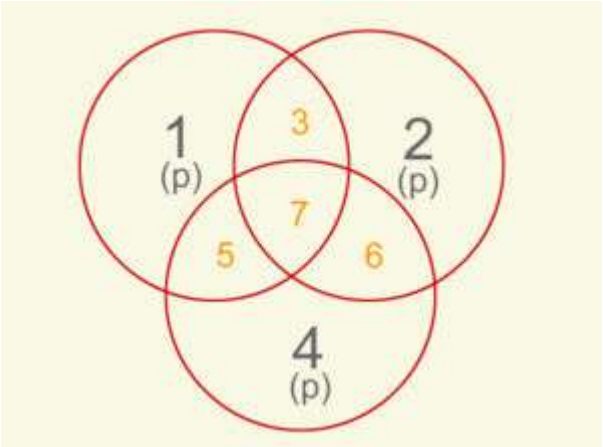
$b_6 = a_2 + a_3$

với v = [1100101] ta giải được u = [1101]

Thực tế dữ liệu không chỉ được kiểm tra lỗi theo từng byte mà còn được kiểm tra lỗi theo dạng khối; sử dụng ma trận sinh trong việc kiểm tra lỗi hướng trước FEC mang lại hiệu quả cao.

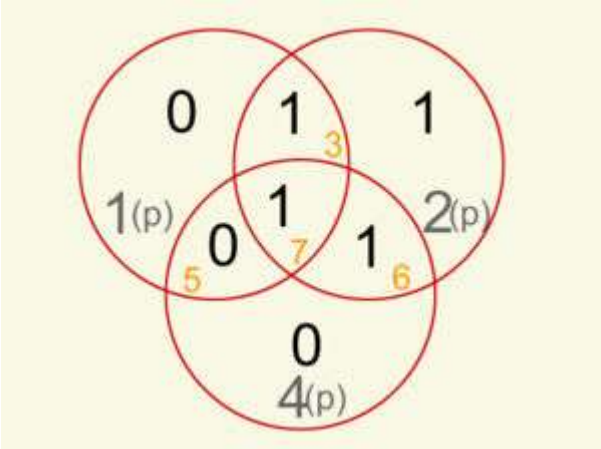
3.1.2. Mã Hamming:

Mã Hamming là một mã sửa lỗi tuyến tính (linear error-correcting code), được đặt tên theo tên của người phát minh ra nó, Richard Hamming. Mã Hamming có thể phát hiện một bit hoặc hai bit bị lỗi (single and double-bit errors). Mã Hamming còn có thể sửa các lỗi do một bit bị sai gây ra. Ngược lại mã chẵn lẻ (parity code) đơn giản vừa không có khả năng phát hiện các lỗi khi 2 bit cùng một lúc bị hoán vị (0 thành 1 và ngược lại), vừa không thể giúp để sửa được các lỗi mà nó phát hiện thấy.



Mô hình của một mã 7-bit, bao gồm 4 bit dữ liệu (3,5,6,7) và 3 bit chẵn lẻ (1,2,4). Sự liên quan của các bit dữ liệu với bit chẵn lẻ được biểu hiện bằng các phần của hình

tròn gối lên nhau. Bit thứ 1 kiểm tra bit thứ (3, 5, 7), trong khi bit 2 kiểm tra bit (3, 6, 7). Lưu ý, các vị trị (1,2,4 v.v.) thực ra là vị trí 2^0, 2^1, 2^2 v.v.



Các bit dữ liệu và bit chẵn lẻ trong mối quan hệ chồng gối với nhau. Các bit chẵn lẻ được tính dùng quy luật "số chẵn". Giá trị của nhóm dữ liệu là 1100110 - các bit chẵn lẻ được in đậm, và đọc từ phải sang trái.

Càng nhiều bit sửa lỗi thêm vào trong thông điệp, và các bit ấy được bố trí theo một cách là mỗi bố trí của nhóm các bit bị lỗi tạo nên một hình thái lỗi riêng biệt, thì chúng ta có thể xác định được những bit bị sai. Trong một thông điệp dài 7-bit, chúng ta có 7 khả năng một bit có thể bị lỗi, như vậy, chỉ cần 3 bit kiểm tra ($2^3 = 8$) là chúng ta có thể, không những chỉ xác định được là lỗi trong truyền thông có xảy ra hay không, mà còn có thể xác định được bit nào là bit bị lỗi.

Hamming nghiên cứu các kế hoạch mã hóa hiện có, bao gồm cả mã hai-trong-năm, rồi tổng quát hóa khái niệm của chúng. Khởi đầu, ông xây dựng một danh mục (nomenclature) để diễn tả hệ thống máy, bao gồm cả số lượng bit dùng cho dữ liệu và các bit sửa lỗi trong một khối. Chẳng hạn, bit chẵn lẻ phải thêm 1 bit vào trong mỗi từ dữ liệu (DATA word). Hamming diễn tả phương pháp này là mã (8,7). Nó có nghĩa là một từ dữ liệu có tổng số bit là 8 bit, trong đó chỉ có 7 bit là các bit của dữ liệu mà thôi. Theo phương pháp suy nghĩ này, mã tái diễn (nhắc lại) ở trên phải được gọi là mã (3,1). Tỷ lệ thông tin là tỷ lệ được tính bằng việc lấy con số thứ hai chia cho con số thứ nhất. Như vậy với mã tái diễn (3,1) ở trên, tỷ lệ thông tin của nó là $\frac{1}{3}$.

Hamming còn phát hiện ra vấn đề với việc đảo giá trị của hai hoặc hơn hai bit nữa, và miêu tả nó là "khoảng cách" (distance) (hiện nay nó được gọi là khoảng cách Hamming (Hamming distance) - theo cái tên của ông). Mã chẵn lẻ có khoảng cách bằng 2, vì nếu có 2 bit bị đảo ngược thì lỗi trong truyền thông trở nên vô hình, không phát hiện được. Mã tái diễn (3,1) có khoảng cách là 3, vì 3 bit, trong cùng một bộ ba, phải bị đổi ngược trước khi chúng ta được một từ mã khác. Mã tái diễn (4,1) (mỗi bit được nhắc lại 4 lần) có khoảng cách bằng 4, nên nếu 2 bit trong cùng một nhóm bị đảo ngược thì lỗi đảo ngược này sẽ đi thoát mà không bị phát hiện.

Cùng một lúc, Hamming quan tâm đến hai vấn đề; tăng khoảng cách và đồng thời tăng tỷ lệ thông tin lên, càng nhiều càng tốt. Trong những năm thuộc niên kỷ 1940, ông đã xây dựng môt số kế hoạch mã hóa. Những kế hoạch này đều dựa trên những mã hiện tồn tại song được nâng cấp và tiến bộ một cách sâu sắc. Bí quyết chìa khóa cho tất cả các hệ thống của ông là việc cho các bit chẵn lẻ gối lên nhau (overlap), sao cho chúng có khả năng tự kiểm tra lẫn nhau trong khi cùng kiểm tra được dữ liệu nữa.

Thuật toán cho việc sử dụng bit chẵn lẻ trong 'mã Hamming' thông thường cũng tương đối đơn giản:

1. Tất cả các bit ở vị trí là các số mũ của 2 (powers of two) được dùng làm bit chẵn lẻ. (các vị trí như 1, 2, 4, 8, 16, 32, 64 v.v. hay nói cách khác 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6 v.v.)

2. Tất cả các vị trí bit khác được dùng cho dữ liệu sẽ được mã hóa. (các vị trí 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)

3. Mỗi bit chẵn lẻ tính giá trị chẵn lẻ cho một số bit trong từ mã (code word). Vị trí của bit chẵn lẻ quyết định chuỗi các bit mà nó luân phiên kiểm tra và bỏ qua (skips).

- Vị trí 1 (n=1): bỏ qua 0 bit(n-1), kiểm 1 bit(n), bỏ qua 1 bit(n), kiểm 1 bit(n), bỏ qua 1 bit(n), v.v.
- Vị trí 2(n=2): bỏ qua 1 bit(n-1), kiểm 2 bit(n), bỏ qua 2 bit(n), kiểm 2 bit(n), bỏ qua 2 bit(n), v.v.
- Vị trí 4(n=4): bỏ qua 3 bit(n-1), kiểm 4 bit(n), bỏ qua 4 bit(n), kiểm 4 bit(n), bỏ qua 4 bit(n), v.v.

- Vị trí 8(n=8): bỏ qua 7 bit(n-1), kiểm 8 bit(n), bỏ qua 8 bit(n), kiểm 8 bit(n), bỏ qua 8 bit(n), v.v.
- Vị trí 16(n=16): bỏ qua 15 bit(n-1), kiểm 16 bit(n), bỏ qua 16 bit(n), kiểm 16 bit(n), bỏ qua 16 bit(n), v.v.
- Vị trí 32(n=32): bỏ qua 31 bit(n-1), kiểm 32 bit(n), bỏ qua 32 bit(n), kiểm 32 bit(n), bỏ qua 32 bit(n), v.v.
- và tiếp tục như trên.

Nói cách khác, bit chẵn lẻ tại vị trí 2^k kiểm các bit ở các bit ở vị trí t có giá trị logic của phép toán AND giữa k và t là khác 0

a) Ví dụ dùng (11,7) mã Hamming:

Lấy ví dụ chúng ta có một từ dữ liệu dài 7 bit với giá trị là "0110101". Để chứng minh phương pháp các mã Hamming được tính toán và được sử dụng để kiểm tra lỗi, xin xem bảng liệt kê dưới đây. Chữ d (DỮ LIỆU) được dùng để biểu thị các bit dữ liệu và chữ p (parity) để biểu thị các bit chẵn lẻ (parity bits).

Đầu tiên, các bit của dữ liệu được đặt vào vị trí tương thích của chúng, sau đó các bit chẵn lẻ cho mỗi trường hợp được tính toán dùng quy luật bit chẵn lẻ số chẵn[1].

Thứ tự bit	1	2	3	4	5	6	7	8	9	10	11
Vị trí bit chẵn lẻ và các bit dữ liệu	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7
Nhóm dữ liệu (không có bit chẵn lẻ):			0		1	1	0		1	0	1
p_1	1		0		1		0		1		1
p_2		0	0			1	0			0	1
p_3				0	1	1	0				
p_4								0	1	0	1
Nhóm dữ liệu (với bit chẵn lẻ):	1	0	0	0	1	1	0	0	1	0	1

Cách tính các bit chẵn lẻ trong mã Hamming (từ trái sang phải)

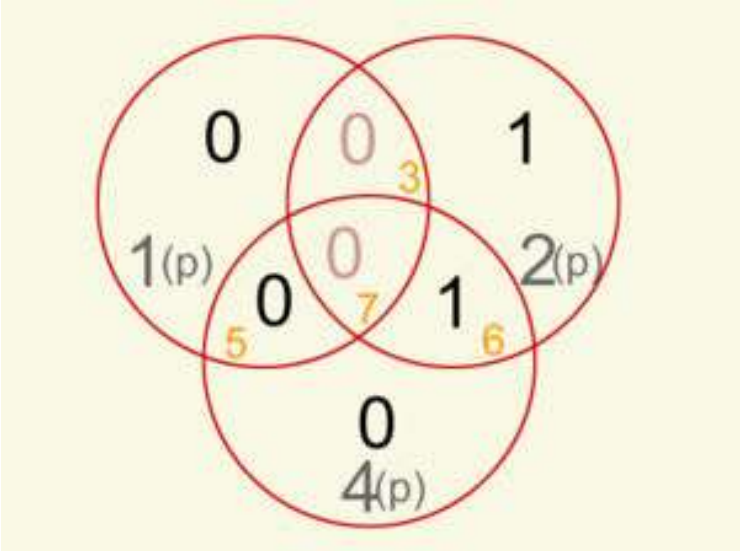
Nhóm dữ liệu mới (new DỮ LIỆU word) - bao gồm các bit chẵn lẻ - bây giờ là "10001100101". Nếu chúng ta thử cho rằng bit cuối cùng bị thoái hóa (gets corrupted) và bị lộn ngược từ 1 sang 0. Nhóm dữ liệu mới sẽ là "10001100100"; Dưới đây, chúng ta sẽ phân tích quy luật kiến tạo mã Hamming bằng cách cho bit chẵn lẻ giá trị 1 khi kết quả kiểm tra dùng quy luật số chẵn bị sai.

Thứ tự bit	1	2	3	4	5	6	7	8	9	10	11		
Vị trí bit chẵn lẻ và các bit dữ liệu	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	Kiểm chẵn lẻ	Bit chẵn lẻ
Nhóm dữ liệu nhận :	1	0	0	0	1	1	0	0	1	0	0	1	
p1	1		0		1		0		1		0	Sai	1
p2		0	0			1	0			0	0	Sai	1
p3				0	1	1	0					Đúng	0
p4								0	1	0	0	Sai	1

Kiểm tra các bit chẵn lẻ (bit bị đảo lộn có nền thẫm)

Bước cuối cùng là định giá trị của các bit chẵn lẻ (nên nhớ bit nằm dưới cùng được viết về bên phải - viết ngược lại từ dưới lên trên). Giá trị số nguyên của các bit chẵn lẻ là $11_{(10)}$, và như vậy có nghĩa là bit thứ 11 trong nhóm dữ liệu (DỮ LIỆU word) - bao gồm cả các bit chẵn lẻ - là bit có giá trị không đúng, và bit này cần phải đổi ngược lại.

	p4	p3	p2	p1	
Nhị phân	1	0	1	1	
Thập phân	8		2	1	Σ = 11



Khi hai bit dữ liệu (3,7) có cùng bit chẵn lẻ kiểm tra tại vi trí 2^k - ví dụ (1,2) - biến đổi giá trị (lỗi trong truyền thông) thì giá trị của bit chẵn lẻ vẫn đúng như giá trị gốc (0,1)

Việc đổi ngược giá trị của bit thứ 11 làm cho nhóm

10001100100

trở lại thành

10001100101.

Bằng việc bỏ đi phần mã Hamming, chúng ta lấy được phần dữ liệu gốc với giá trị là: 0110101.

Lưu ý, các bit chẵn lẻ không kiểm tra được lẫn nhau, nếu chỉ một bit chẵn lẻ bị sai thôi, trong khi tất cả các bit khác là đúng, thì chỉ có bit chẵn lẻ nói đến là sai mà thôi và không phải là các bit nó kiểm tra (not any bit it checks).

Cuối cùng, giả sử có hai bit biến đổi, tại vị trí x và y. Nếu x và y có cùng một bit tại vị trí 2^k trong đại diện nhị phân của chúng, thì bit chẵn lẻ tương ứng với vị trí đấy kiểm tra cả hai bit, và do đó sẽ giữ nguyên giá trị, không thay đổi. Song một số bit chẵn lẻ nào đấy nhất định phải bị thay đổi, vì $x \neq y$, và do đó hai bit tương ứng nào đó có giá trị x và y khác nhau. Do vậy, mã Hamming phát hiện tất cả các lỗi do hai bit bị thay đổi — song nó không phân biệt được chúng với các lỗi do 1 bit bị thay đổi.

b) Mã Hamming (7,4):

Với mỗi nhóm 4 bit dữ liệu, mã Hamming thêm 3 bit kiểm tra. Thuật toán (7,4) của Hamming có thể sửa chữa bất cứ một bit lỗi nào, và phát hiện tất cả lỗi của 1 bit, và các lỗi của 2 bit gây ra. Điều này có nghĩa là đối với tất cả các phương tiện truyền thông không có chùm lỗi đột phát (burst errors) xảy ra, mã (7,4) của Hamming rất có hiệu quả (trừ phi phương tiện truyền thông có độ nhiễu rất cao thì nó mới có thể gây cho 2 bit trong số 7 bit truyền bị đảo lộn).

c) Ví dụ về cách dùng các ma trận thông qua GF(2) [2]

Nguyên lý của mã Hamming bắt nguồn từ việc khai triển và mở rộng quan điểm chẵn lẻ. Việc khai triển này bắt đầu bằng việc nhân các ma trận, được gọi là Ma trận Hamming (Hamming matrices), với nhau. Đối với mã Hamming (7,4), chúng ta sử dụng hai mã trận có liên quan gần gũi, và đặt tên cho chúng là:

$$H_e := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

và

$$H_d := \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Các cột vectơ trong H_e là nên tảng hạch của H_d và phần trên của H_e (4 hàng đầu) là một ma trận đơn vị (identity matrix). Ma trận đơn vị cho phép vectơ dữ liệu đi qua trong khi làm tính nhân, và như vậy, các bit dữ liệu sẽ nằm ở 4 vị trí trên cùng (sau khi nhân). Sau khi phép nhân hoàn thành, khác với cách giải thích ở phần trước (các bit chẵn lẻ nằm ở vị trí 2^k), trật tự của các bit trong từ mã (codewords) ở đây khác với cách bố trí đã nói (các bit dữ liệu nằm ở trên, các bit kiểm chẵn lẻ nằm ở dưới).

Chúng ta dùng một nhóm 4 bit dữ liệu (số 4 trong cái tên của mã là vì vậy) chủ chốt, và cộng thêm vào đó 3 bit dữ liệu thừa (vì 4+3=7 nên mới có số 7 trong cái tên

của mã). Để truyền gửi dữ liệu, chúng ta hãy nhóm các bit dữ liệu mà mình muốn gửi thành một vectơ. Lấy ví dụ, nếu dữ liệu là "1011" thì vectơ của nó là:

$$\mathbf{p} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Giả sử, chúng ta muốn truyền gửi dữ liệu trên. Chúng ta tìm tích của H_e và p, với các giá trị môđulô 2 [3]:

$$H_e\mathbf{p} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \mathbf{r}$$

Máy thu sẽ nhân H_d với r, để kiểm tra xem có lỗi xảy ra hay không. Thi hành tính nhân này, máy thu được (một lần nữa, các giá trị đồng dư môđulô 2):

$$H_d\mathbf{r} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Vì chúng ta được một vectơ toàn số không cho nên máy thu có thể kết luận là không có lỗi xảy ra

Sở dĩ một vectơ toàn số không có nghĩa là không có lỗi, bởi vì khi H_e được nhân với vectơ dữ liệu, một sự thay đổi trong nền tảng xảy ra đối với không gian bên trong vectơ (vector subspace), tức là hạch của H_d. Nếu không có vấn đề gì xảy ra trong khi truyền thông, r sẽ nằm nguyên trong hạch của H_d và phép nhân sẽ cho kết quả một vectơ toàn số không.

Trong một trường hợp khác, nếu chúng ta giả sử là lỗi một bit đã xảy ra. Trong toán học, chúng ta có thể viết:

$$\mathbf{r} + \mathbf{e}_i$$

môđulô 2, trong đó e_i là vectơ đơn vị đứng thứ i (ith unit vector), có nghĩa là, một vectơ số 0 có một giá trị 1 trong vị trí i (tính từ 1 tính đi). Biểu thức trên nói cho chúng ta biết rằng có một bit bị lỗi tại vị trí i.

Nếu bây giờ chúng ta nhân H_d với cả hai vectơ này:

$$H_d(\mathbf{r} + \mathbf{e}_i) = H_d\mathbf{r} + H_d\mathbf{e}_i$$

Vì r là dữ liệu thu nhận được không có lỗi, cho nên tích của H_d và r bằng 0. Do đó

$$H_d\mathbf{r} + H_d\mathbf{e}_i = 0 + H_d\mathbf{e}_i = H_d\mathbf{e}_i$$

Vậy, tích của H_d với vectơ nền chuẩn tại cột thứ i (the ith standard basis vector) làm lộ ra cột ở trong H_d, vì thế mà chúng ta biết rằng lỗi đã xảy ra tại vị trí cột này trong H_d. Vì chúng ta đã kiến tạo H_d dưới một hình thức nhất định, cho nên chúng ta có thể hiểu giá trị của cột này như một số nhị phân - ví dụ, (1,0,1) là một cột trong H_d, tương đồng giá trị với cột thứ 5, do đó chúng ta biết lỗi xảy ra ở đâu và có thể sửa được nó.

Lấy ví dụ, giả sử chúng ta có:

$$\mathbf{s} = \mathbf{r} + \mathbf{e}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Nếu thi hành phép nhân:

$$H_d\mathbf{s} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Tích của phép nhân cho chúng ta một kết quả tương đương với cột thứ 2 ("010" tương đương với giá trị 2 trong số thập phân), và do đó, chúng ta biết rằng lỗi đã xảy ra ở vị trí thứ 2 trong hàng dữ liệu, và vì vậy có thể sửa được lỗi.

Chúng ta có thể dễ dàng thấy rằng, việc sửa lỗi do 1 bit bị đảo lộn gây ra, dùng phương pháp trên là một việc thực hiện được. Bên cạnh đó, mã Hamming còn có thể phát hiện lỗi do 1 bit hoặc 2 bit bị đảo lộn gây ra, dùng tích của H_d khi tích này không cho một vectơ số không. Tuy thế, song mã Hamming không thể hoàn thành cả hai việc.

d) Mã Hamming và bit chẵn lẻ bổ sung:

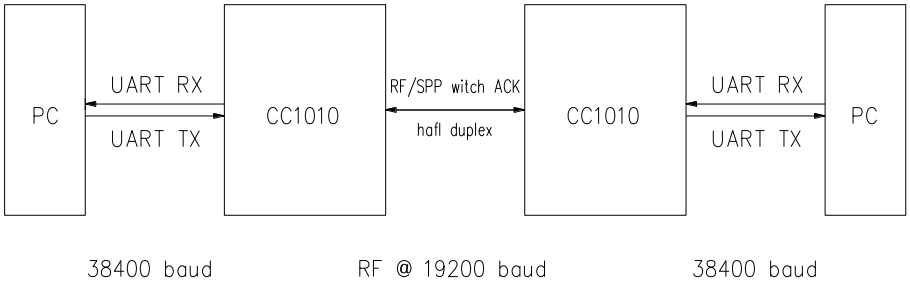
Nếu chúng ta bổ sung thêm một bit vào mã Hamming, thì mã này có thể dùng để phát hiện những lỗi gây ra do 2 bit bị lỗi, và đồng thời nó không cản trở việc sửa các lỗi do một bit gây ra. [1] Nếu không bổ sung một bit vào thêm, thì mã này có thể phát hiện các lỗi do một bit, hai bit, ba bit gây ra, song nó sẽ cản trở việc sửa các lỗi do một bit bị đảo lộn. Bit bổ sung là bit được áp dụng cho tất cả các bit sau khi tất cả các bit kiểm của mã Hamming đã được thêm vào.

Khi sử dụng tính sửa lỗi của mã, nếu lỗi ở một bit chẵn lẻ bị phát hiện và mã Hamming báo hiệu là có lỗi xảy ra thì chúng ta có thể sửa lỗi này, song nếu chúng ta không phát hiện được lỗi trong bit chẵn lẻ, nhưng mã Hamming báo hiệu là có lỗi xảy ra, thì chúng ta có thể cho rằng lỗi này là do 2 bit bị đổi cùng một lúc. Tuy chúng ta phát hiện được nó, nhưng không thể sửa lỗi được.

7 bit dữ liệu	byte có bit chẵn lẻ	
	Quy luật số chẵn	Quy luật số lẻ
0000000	00000000	00000001
1010001	10100011	10100010
1101001	11010010	11010011
1111111	11111111	11111110

Cách tính bit chẵn lẻ trong nhóm các bit dữ liệu, dùng quy luật số chẵn như sau: nếu số lượng bit có giá trị bằng 1 (the bit is set) là một số lẻ, thì bit chẵn lẻ bằng $1_{(2)}$ (và do việc cộng thêm một bit có giá trị $1_{(2)}$ này vào dữ liệu, tổng số bit có giá trị $1_{(2)}$ sẽ là một số chẵn - bao gồm cả bit chẵn lẻ, còn không thì bit chẵn lẻ sẽ có giá trị $0_{(2)}$. Ngược lại, bit chẵn lẻ dùng quy luật số lẻ sẽ có giá trị $1_{(2)}$ nếu số lượng các bit có giá trị bằng $1_{(2)}$ là một số chẵn - do việc thêm bit chẵn lẻ có giá trị bằng $1_{(2)}$ vào nhóm dữ liệu, tổng số bit có giá trị $1_{(2)}$ là một số lẻ - và bằng 0 nếu ngược lại.

3.3 Một vài ví dụ về hình thành từ mã và sửa lỗi:



SƠ ĐỒ MINH HỌA QUA TRÌNH TRUYỀN TIN GIỮA 2 NÚT MẠNG SỬ DỤNG GIAO THỨC SPP CÓ ACK

Nguyên lý truyền dẫn giữa 2 nút mạng được mô tả như sau:

Với giao thức SPP (simple packet protocol) có ACK thì đường truyền giữa 2 nút là hlf duplex, chỉ có một bên được truyền. Quá trình truyền được diễn ra ngay sau khi phía truyền nhận được gói ACK từ phía nhận.

Quá trình truyền:

Nếu bộ thu phát ở trạng thái rỗi (sẵn sàng truyền) thì thành phần phát sẽ được kích hoạt. Quá trình truyền được thực hiện trên RF ISR và nút truyền chờ nhận ack (nếu được yêu cầu).

Nếu tại bộ phát (nút truyền) được cấp yêu cầu truyền lại với sppSetting.TXAttempts=n) thì gói tin sẽ được truyền lại (n-1) lần cho đến khi nhận được gói tin phản hồi ack.

Quá trình truyền nhận diễn ra liên tục ngay cả khi UART vẫn đang trong quá trình nhận, trong suốt quá truyền nó được đặt vào chế độ TX_Mode (chế độ đang truyền) hoặc chờ nhận ACK (TXACK_MODE). Kết thúc hoạt động truyền trạng thái của đường truyền sppStatus ( ) sẽ trở lại trạng thái rỗi (spp_IDLE_MODE).

Quá trình nhận:

Nếu bộ thu phát rỗi (ở trạng thái sẵn sàng truyền và nhận (spp_IDLE_MODE); khi đó nếu nhận được yêu cầu nhận, thì thành phần nhận trong nút mạng sẽ được kích hoạt. Quá trình nhận được thực hiện nhờ RF ISR (radio-Frequency Interrupt-Service-Routine) và thực hiện truyền ACK (nếu được yêu cầu).

Kết thúc mỗi lần nhận. Bộ nhận sẽ được tắt nguồn. Nhưng chức năng nhận sẽ được kích hoạt trở lại ngay khi ứng dụng được yêu cầu tiếp theo đó. Trong suốt quá trình nhận bộ nhận luôn ở trạng thái chờ nhận gói tin (RX-MODE) hoặc đang truyền ACK (RXACK_MODE).

Kết thúc hoạt động truyền, trạng thái của đường truyền spp Status ( ) sẽ trở lại trạng thái rỗi ban đầu (spp_IDLE_MODE).

Giữa hai nút mạng cảm biến thì quá trình truyền và nhận luôn được chuyển đổi qua lại. Do đường truyền là bán song công nên trong cùng một thời điểm chỉ có một bên được phép truyền, kết thúc quá trình truyền nhận thường bộ thu phát sẽ thiết lập lại các giá trị: Hệ thống truyền dẫn RF, thời gian time out vv.... Dưới đây là minh họa về cấu trúc chương trình truyền nhận trên RF được thực hiện trong vòng lặp While:

```
While (TRUE) {
//Receive a packet - Nếu nhận tin
if (sppReceive(&RXI) ==SPP_RX_STARTED) {
```

//thực hiện truyền khi bên nhận đã sẵn sàng. Nhưng cần kiểm tra nếu đường truyền RF đang có gói tin được truyền thì cần chờ cho nó truyền xong rồi mới thực hiện. Vòng lặp dưới đây được thực hiện khi SPP_Status ( ) không rỗi.

```
do {
if (txRequest && (TXI.status == SPP_TX_FINISHED) && (RXI.status == SPP_RX_WAITING)) {
sppReset ();
}
} while (SPP_STATUS()!=SPP_IDLE_MODE
```

```
// gói tin sau khi được nhận hoàn chỉnh sẽ được đưa vào UART
//Check the results (ignore retransmitted messages)
if (RXI.status == SPP_RX_FINISHED)  {
if (SPP_SEQUENCE_BIT & (lastRXflags ^RXI.flags)) {
while (uartRXPos !=rxDateLen[uartRXBuffer]);
if ((rxDataLen[rfRXBuffer] = RXI.dataLen) !=0 dau
//Switch buffers and initiate UART0 transmission
GLED=~GLED;
SWITCH(uartRXBuffer, rfRXBuffer);
uartRXPos=0;
UART0_SEND(pRXBuffer[uartRXBuffer][uartRXPos]);
      }
   }
lastRXflags = RXI.flags;
 }
}
```

//Transmitpacket, if requested --- truyền tin nếu nhận được yêu cầu

```
if (txRequest)  {                    //thực hiện khi có yêu cầu
if (sppSend(&TXI) === SPP_TX_STARTED)  {
```

//kiểm tra xem trên đường truyền có rỗi không, nếu không rỗi thì chờ cho đến khi quá trình truyền kết thúc mới thực hiện,

```
Wait for the transmission to finish
YLED = LED_ON;                    //đèn vàng bật <-> đang truyền
}while (SPP_STATUS() !=SPP_IDLE_MODE
```

//nếu đường truyền rỗi thì ta thực hiện truyền

```
YLED=LED_OFF;
```

```
//Check out the results
if(TXI.status==SPP_TX_FINISHED)  {
sppSetting.rxTimeout = NORMAL_TIMEOUT;
RLED = LED_OFF;
txRequest = TX_REQUEST_OFF;
} else {
RLED = LED_ON
sppSetings.rxTimeout = RETRY_TX_TIMEOUT;          }
                }
}
```

//Recalibrate the transceiver if requested to ---xác lập lại tại bộ thu phát nếu có lệnh yêu cầu

```
if (recalibRequest)  {
     sppSetup RF(&RF_SETTINGS, &RF_CALDATA, TRUE);
      }
    }
}//main
```

KẾT LUẬN

Đề tài đã xem xét các đặc trưng chủ yếu của mạng cảm nhận không dây. Nghiên cứu chuyên sâu về việc: Tăng khả năng thành công truyền dữ liệu trong mạng không dây bằng phương pháp mã hóa dữ liệu. Trên cơ sở:

- Xây dựng từ mã cho việc tự động sửa lỗi.

- Đánh giá khoảng cách Hamming qua các từ mã.

- Đánh giá khả năng phát hiện lỗi và sửa lỗi qua tính toán khoảng cách Hamming.

- Đưa ra phương pháp phát hiện và sửa lỗi tại nơi nhận qua tập hợp từ mã nhận được.

Để giải quyết các vấn đề trên ta sử dụng phương pháp sửa lỗi FEC (Forward Error Correction) - Sửa lỗi hướng trước. Với mã sửa lỗi FEC, dữ liệu mất sẽ được khôi phục ở phía thu mà không cần yêu cầu phát lại ở phía phát. Ưu điểm của nó là có thể tự khôi phục lại các gói bị mất nhờ vào các gói đã thu được và phần gói chẵn lẻ dư thừa được thêm vào tại nơi phát trong quá trình truyền như vậy lỗi đơn bit và các đoạn dữ liệu còn thiếu tại bộ thu sẽ được tự động sửa lại nhờ vào FEC.

Trên thực tế sửa lỗi hướng trước FEC cho thấy hiệu quả trong việc giảm đi các bit lỗi khi hầu hết là các lỗi đơn bít với số lượng các bit lỗi trong gói tin là nhỏ (đơn bit, 2 bit hay 3 bit) còn với trường hợp lỗi chùm bit hay nhiều lỗi nó không phát huy được nhiều tác dụng. Mặc dù lỗi đa bit hoặc chùm bit là hiếm nhưng nếu gặp phải đôi khi cũng gây nên những ảnh hưởng lớn và tuy ARQ đã được đưa ra để giải quyết những trường hợp này nhưng dễ thấy đó không phải là phương pháp tối ưu, hơn nữa cũng không có gì thú vị nếu trong cùng một nút mạng mà sử dụng hai thuật toán liền để sửa lỗi. Do vậy sẽ thật sự có ý nghĩa nều tìm được phương phát hiệu quả mà có thể sửa được đồng thời cả lỗi đơn bit, đa bit và chùm bit mà vẫn đảm bảo các yêu cầu về thuật toán: như đơn giản, tốn ít dung lượng bộ nhớ, tiêu thụ năng lượng thấp vv...

Tài liệu tham khảo

[1] "Mạng truyền số liệu" - Vương Đạo Vy - Nhà xuất bản ĐHQG - Năm 2006.

[2] Mạng máy tính và các hệ thống mở (Nguyễn Thúc Hải)

[3] Bài giảng mạng máy tính (Phạm Thế Quế)

[4] Wireless Local Area Netwozzrks *(Pierfranco Issa 1999)*

[5] Designing A Wireless Network *(Syngress Publishing 2001)*

[6] Building A Cisco Wireless LAN *(Syngress Publishing 2002)*

[7] Building Wireless Community Networks *(O'Reilly 2002)*

[8] Configuring the Cisco Wireless Security Suite *(Cisco System 2002)*

[9] Wireless Security and Privacy: Best Practices and Design Techniques *(Addison Wesley 9/2002)*

[10] Building Secure Wireless Networks with 802.11 *(Wiley Publishing 2003)*

[11] Wireless Security: Critical Issues and Solutions *(Craig J. Mathias 2003)*