

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Khoa Kỹ thuật điện tử I

Ths. Vũ Anh Đào - Ths. Trần Thị Thục Linh

Bài giảng:

MẠNG CẢM BIẾN KHÔNG DÂY

Hà nội, 7/2010

MỤC LỤC

MỤC LỤC	2
LỜI NÓI ĐẦU.....	6
DANH MỤC CÁC HÌNH VẼ	8
DANH MỤC BẢNG BIỂU	11
CÁC THUẬT NGỮ VIẾT TẮT.....	12
CHƯƠNG 1. GIỚI THIỆU CHUNG	17
1.1 Cơ sở hạ tầng cho mạng cảm biến không dây	17
1.2 Mạng cảm biến không dây và ứng dụng.....	19
1.3 Những khó khăn của mạng cảm biến không dây.....	26
1.3.1 Các yêu cầu kỹ thuật.....	26
1.3.2 Các kỹ thuật yêu cầu.....	28
1.4 Tại sao mạng cảm biến lại khác với các mạng khác?	29
1.4.1 Mạng Ad hoc di động và WSN	29
1.4.2 Fieldbus and WSN.....	32
1.5 Các kỹ thuật cho WSN	33
CÂU HỎI ÔN TẬP CHƯƠNG 1	34
CHƯƠNG 2. KIẾN TRÚC NÚT ĐƠN.....	35
2.1 Các bộ phận phần cứng	35
2.1.1 Tổng quan phần cứng của nút cảm biến	35
2.1.2 Bộ điều khiển.....	36
2.1.3 Bộ nhớ	38
2.1.4 Thiết bị liên lạc	38
2.1.5 Các cảm biến và cơ cấu chấp hành	46
2.1.6 Nguồn cung cấp của các nút cảm biến	47
2.2 Tiêu thụ năng lượng của các nút cảm biến	48
2.2.1 Các trạng thái hoạt động với mức tiêu thụ công suất khác nhau	48
2.2.2 Tiêu thụ năng lượng bộ vi xử lý	50
2.2.3 Bộ nhớ	52
2.2.4 Các bộ thu phát vô tuyến	52
2.2.5 Mối quan hệ giữa tính toán và liên lạc	54
2.2.6 Tiêu thụ năng lượng của các cảm biến và cơ cấu chấp hành.....	55
2.3 Các hệ điều hành và môi trường thực hiện.....	55
2.3.1 Các hệ điều hành nhúng.....	55
2.3.2 Mô hình lập trình và các giao diện lập trình ứng dụng.....	56
2.3.3 Cấu trúc hệ điều hành và ngăn xếp giao thức.....	58
2.3.4 Năng lượng động và quản lý năng lượng	59
2.4 Một số ví dụ của nút cảm biến.....	60
2.4.1 Họ Mica Mote.....	60
2.4.2 Các nút EYES.....	60
2.4.3 Btnode.....	61
2.5 Kết luận	61
CÂU HỎI ÔN TẬP CHƯƠNG 2	63
CHƯƠNG 3. KIẾN TRÚC MẠNG	64
3.1 Mạng cảm biến	64
3.1.1 Nguồn và bộ chứa đơn.....	64
3.1.2 Mạng một bước nhảy và đa bước nhảy	65
3.1.3 Đa bộ chứa và nguồn	66
3.1.4 Ba loại chuyển động	66

3.2 Các mục tiêu tối ưu và hệ số chất lượng.....	68
3.2.1 Chất lượng dịch vụ	68
3.2.2 Hiệu quả năng lượng	69
3.2.3 Khả năng mở rộng	70
3.2.4 Tính bền vững.....	70
3.3 Nguyên tắc thiết kế WSN	70
3.3.1 Tổ chức phân bố	70
3.3.2 Xử lý trong mạng.....	71
3.3.3 Tính chính xác và độ tin cậy.....	73
3.3.4 Tập trung dữ liệu	73
3.3.5 Khai thác thông tin vị trí.....	74
3.3.6 Khai thác các mô hình hoạt động	75
3.3.7 Giao thức dựa trên phân tử và tối ưu lớp giao nhau	75
3.4 Giao diện dịch vụ của WSN	75
3.4.1 Cấu trúc của giao diện ngăn xếp giao thức/ứng dụng	75
3.4.2 Các yêu cầu cho giao diện dịch vụ của WSN.....	76
3.5 Khái niệm cổng vào/ra.....	77
3.5.1 Sự cần thiết của các cổng vào/ra	77
3.5.2 Giao tiếp WSN với Internet.....	78
3.5.3 Giao tiếp Internet-WSN.....	79
3.5.4 Giao tiếp giữa các WSN.	80
3.6 Kết luận	80
CÂU HỎI ÔN TẬP CHƯƠNG 3	82
CHƯƠNG 4. LỚP VẬT LÝ.....	83
4.1 Giới thiệu.....	83
4.2 Kênh vô tuyến và cơ sở truyền tin.....	83
4.2.1 Cáp phát tần số	83
4.2.2 Điều chế và giải điều chế.....	85
4.2.3 Các ảnh hưởng của truyền sóng và nhiễu	88
4.2.4 Các mô hình kênh	93
4.2.5 Công nghệ trải phổ	94
4.2.6 Đồng bộ và truyền gói tin.....	95
4.2.7 Chất lượng kênh vô tuyến và các phép đo độ cải thiện	97
4.3 Lớp vật lý và thiết kế bộ thu phát trong WSN.....	98
4.3.1 Sử dụng năng lượng.....	99
4.3.2 Lựa chọn sơ đồ điều chế	99
4.3.3 Antenna.....	100
CÂU HỎI ÔN TẬP CHƯƠNG 4	101
CHƯƠNG 5. CÁC GIAO THỨC MAC	102
5.1. Các nguyên tắc cơ bản của các giao thức MAC (không dây)	102
5.1.1 Các yêu cầu và ràng buộc thiết kế cho các giao thức MAC không dây.	102
5.1.2 Phân loại các giao thức MAC.....	104
5.1.3 Các giao thức MAC cho WSN	109
5.2 Các giao thức hệ số lấp đầy thấp và khái niệm thức dậy (wakeup).....	110
5.2.1 Cấu hình mỏng và vấn đề quản lý năng lượng	111
5.2.2 S-MAC	113
5.2.3 Giao thức thiết bị trung gian.....	116
5.3 Các giao thức theo kiểu cạnh tranh	117
5.3.1 Các giao thức CSMA.....	118
5.3.2 PAMAS	118
5.4 Các giao thức dựa trên lịch trình	120
5.4.1 LEACH.....	120

5.4.2 SMACS.....	122
5.5 Giao thức MAC của IEEE 802.15.4	124
5.5.1 Kiến trúc mạng, loại và vai trò của nút	126
5.5.2 Kiến trúc của siêu khung	127
5.5.3 Giao thức Slotted CSMA-CA.....	128
5.6 Tổng kết.....	129
CÂU HỎI ÔN TẬP CHƯƠNG 5	130
CHƯƠNG 6. CÁC GIAO THỨC LỚP LIÊN KẾT	131
6.1 Các nhiệm vụ và yêu cầu cơ bản	131
6.2 Giám sát lỗi.....	132
6.2.1 Nguyên nhân và các đặc tính của sai số truyền dẫn	132
6.2.2 Các kỹ thuật ARQ	133
6.2.3 Các kỹ thuật FEC.....	138
6.2.4 Các sơ đồ lai	143
6.2.5 Điều khiển công suất	145
6.2.6 Điều khiển lỗi	145
6.3 Định dạng khung	146
6.3.1 Các sơ đồ tương thích.....	149
6.3.2 Các sơ đồ kiểm tra tổng trung gian.....	150
6.3.3 Kết hợp tối ưu kích thước gói tin và FEC	152
6.4 Quản lý tuyến.	153
6.4.1 Các đặc tính của chất lượng tuyến.....	153
6.4.2 Ước lượng chất lượng tuyến.....	155
6.5 Tổng kết.....	157
CÂU HỎI ÔN TẬP CHƯƠNG 6	158
CHƯƠNG 7. ĐẶT TÊN VÀ ĐÁNH ĐỊA CHỈ.....	159
7.1 Các nguyên tắc cơ bản.....	159
7.1.1 Sử dụng các địa chỉ và các tên trong các mạng (cảm biến).....	159
7.1.2 Nhiệm vụ quản lý địa chỉ.....	160
7.1.3 Sự duy nhất của các địa chỉ	161
7.1.4 Phân phối và ấn định địa chỉ.....	162
7.1.5 Đánh địa chỉ phần tiếp đầu	163
7.2 Quản lý tên và địa chỉ trong các mạng cảm biến không dây	164
7.3 Ấn định các địa chỉ MAC.....	165
7.4. Đánh địa chỉ theo địa lý và dựa trên nội dung.....	167
7.4.1 Đánh địa chỉ dựa trên nội dung	168
7.4.2 Đánh địa chỉ theo địa lý.....	171
7.5 Tổng kết.....	172
CÂU HỎI ÔN TẬP CHƯƠNG 7	173
CHƯƠNG 8. ĐỒNG BỘ THỜI GIAN	174
8.1 Giới thiệu về vấn đề đồng bộ thời gian	174
8.1.1 Sự cần thiết của việc đồng bộ thời gian trong các mạng cảm biến không dây.....	174
8.1.2 Các khóa nút và vấn đề về độ chính xác.....	175
8.1.3 Các đặc điểm và cấu trúc của các thuật toán đồng bộ thời gian.....	177
8.1.4 Đồng bộ thời gian trong các mạng cảm biến không dây	179
8.2 Các giao thức dựa trên đồng bộ máy phát/ máy thu	180
8.2.1 Giao thức đồng bộ thời gian hạng nhẹ - Lightweight time synchronization protocol (LTS)	180
8.2.2 Làm thế nào để tăng độ chính xác và ước lượng độ lệch	185
8.2.3 Giao thức đồng bộ thời gian cho các mạng cảm biến (TPSN)	188
8.3 Các giao thức dựa trên sự đồng bộ máy thu/máy thu	191
8.3.1 Sự đồng bộ quảng bá tham khảo (RBS - Reference broadcast synchronization)..	191

8.3.2 Phân tầng tham chiếu đồng bộ thời gian (HRTS - Hierarchy referencing time synchronization)	197
CÂU HỎI ÔN TẬP CHƯƠNG 8	201
CHƯƠNG 9. ĐỊNH VỊ VÀ XÁC ĐỊNH VỊ TRÍ	202
9.1 Các đặc điểm của quy trình định vị và xác định vị trí	202
9.2 Các phương pháp có thể	204
9.2.1 Phương pháp dựa trên sự gần gũi	204
9.2.2 Phép đặc tam giác (triangulation) và phép đo ba cạnh tam giác (Trilateration)....	204
9.2.3 Phân tích cạnh.....	208
9.3 Cơ sở toán học cho các vấn đề cạnh	209
9.3.1 Giải pháp với ba nút neo và sửa lỗi các giá trị khoảng cách	209
9.3.2 Giải quyết vấn đề các lỗi khoảng cách	210
9.4 Xác định vị trí trong các môi trường đa bước nhảy.....	211
9.4.1 Sự kết nối trong một mạng đa bước nhảy.....	211
9.4.2 Việc ước lượng phạm vi đa bước nhảy.....	212
9.4.3 Đa bên cộng tác và lặp.....	213
9.4.4 Mô tả và truyền lan việc xác định vị trí theo xác suất.....	215
9.5 Tổng kết.....	216
CÂU HỎI ÔN TẬP CHƯƠNG 9	217
CHƯƠNG 10: ĐIỀU KHIỂN CẤU TRÚC LIÊN KẾT MẠNG (TOPOLOGY)	218
10.1.1 Các tùy chọn cho việc điều khiển cấu trúc liên kết mạng	219
10.1.2 Các khía cạnh của các thuật toán điều khiển cấu trúc liên kết mạng	221
10.2 Điều khiển cấu trúc liên kết mạng trong các mạng phẳng – Điều khiển công suất....	223
10.2.1 Một số các kết quả phức tạp	224
10.2.2 Các giới hạn của các tham số chủ chốt.....	225
10.2.3 Vài cấu trúc và giao thức ví dụ.....	227
10.3 Các mạng phân tầng – xương sống.....	229
10.3.1 Động cơ và định nghĩa.....	229
10.3.2 Một số xấp xỉ phân tán	234
10.4 Mạng phân tầng – phân nhóm (clustering).....	237
10.4.1 Định nghĩa các phân nhóm	237
10.4.2 Một ý tưởng cơ bản để xây dựng các tập hợp độc lập.....	240
10.4.4 Kết nối các phân nhóm	241
10.4.5 Quay các chỉ huy phân nhóm	242
CÂU HỎI ÔN TẬP CHƯƠNG 10	243
TÀI LIỆU THAM KHẢO	244

LỜI NÓI ĐẦU

Những tiến bộ gần đây trong thông tin vô tuyến và điện tử đã cho phép phát triển các mạng cảm biến giá thành thấp. Mạng cảm biến có thể được sử dụng trong các ứng dụng khác nhau như chăm sóc sức khỏe, trong quân sự hoặc sử dụng trong gia đình. Mạng cảm biến không dây (WSN) bao gồm các nút nhỏ có khả năng cảm biến, tính toán và trao đổi thông tin vô tuyến. Một số giao thức chọn đường, quản lý năng lượng và trao đổi dữ liệu đã được thiết kế cho WSN với yêu cầu quan trọng nhất là tiết kiệm được năng lượng. Các giao thức chọn đường trong WSN có thể khác nhau tùy theo ứng dụng và cấu trúc mạng. Nói chung, các giao thức chọn đường được chia thành 3 loại dựa vào cấu trúc mạng: ngang hàng, phân cấp hoặc dựa vào vị trí. Ngoài ra, những giao thức này có thể được phân loại dựa theo đa đường, yêu cầu, hỏi/đáp, QoS và liên kết tùy thuộc vào chế độ hoạt động.

Mạng cảm biến không dây (WSN) có thể hiểu đơn giản là mạng liên kết các nút với nhau bằng kết nối sóng vô tuyến, trong đó các nút mạng thường là các (thiết bị) đơn giản, nhỏ gọn, giá thành thấp ... và có số lượng lớn, được phân bố một cách không có hệ thống trên một diện tích rộng, sử dụng nguồn năng lượng hạn chế (pin), có thời gian hoạt động lâu dài (vài tháng đến vài năm) và có thể hoạt động trong môi trường khắc nghiệt (chất độc, ô nhiễm, nhiệt độ ...).

Các nút mạng thường có chức năng cảm ứng, quan sát môi trường xung quanh, theo dõi hay định vị các mục tiêu cố định hoặc di động ... Các nút giao tiếp ad-hoc với nhau và truyền dữ liệu về trung tâm một cách gián tiếp bằng kỹ thuật đa bước nhảy.

Lưu lượng dữ liệu lưu thông trong WSN thường thấp và không liên tục. Do vậy để tiết kiệm năng lượng, các nút cảm biến thường có nhiều trạng thái hoạt động và trạng thái nghỉ khác nhau. Thông thường thời gian một nút ở trạng thái nghỉ lớn hơn ở trạng thái hoạt động rất nhiều.

Như vậy, đặc trưng cơ bản nhất để phân biệt mạng cảm biến và mạng không dây khác chính là giá thành, mật độ nút mạng, phạm vi hoạt động, cấu hình mạng, lưu lượng dữ liệu, năng lượng tiêu thụ và thời gian ở trạng thái hoạt động.

Bài giảng “Mạng cảm biến không dây” gồm 10 chương: 6 chương đầu do Ths. Vũ Anh Đào biên soạn và 4 chương sau do Ths. Trần Thị Thục Linh biên soạn. Cuốn sách giới thiệu cho bạn đọc những kiến thức cơ bản về WSN như các tiêu chí khi xây dựng một WSN và các đặc điểm của WSN như lớp MAC, lớp liên kết dữ liệu, cách đặt tên và đánh địa chỉ..., cụ thể:

Chương 1: Giới thiệu chung

Chương 2: Kiến trúc nút đơn

Chương 3: Kiến trúc mạng

Chương 4: Lớp vật lý

Chương 5: Các giao thức MAC

Chương 6: Các giao thức lớp liên kết

Chương 7: Đặt tên và đánh địa chỉ

Chương 8: Đồng bộ thời gian

Chương 9: Định vị và xác định vị trí

Chương 10: Điều khiển cấu trúc liên kết mạng

Tuy nhiên, do thời gian biên soạn có hạn nên cuốn bài giảng còn có thể có những thiếu sót, rất mong được bạn đọc góp ý. Các ý kiến phản hồi xin gửi về Bộ môn Kỹ thuật điện tử - Khoa Kỹ thuật điện tử I, Học viện Công nghệ Bưu chính Viễn thông.

Xin trân trọng giới thiệu!

Nhóm tác giả

DANH MỤC CÁC HÌNH VẼ

Hình 1.1 Mạng điểm – điểm.....	21
Hình 1.2 Mạng điểm – đa điểm.....	21
Hình 1.3 Mạng đa điểm – đa điểm.....	22
Hình 1.4 Sơ đồ nguyên lý mạng chia sẻ kết nối dạng ad hoc.....	30
Hình 2.1 Cấu trúc phần cứng của nút cảm biến.....	36
Hình 2.2 Đầu cuối RF.....	43
Hình 2.3 Tiết kiệm năng lượng và chi phí cho các chế độ ngủ.....	49
Hình 2.4 Năng lượng/hoạt động với dải điện áp động trên Intel StrongARM-1100.....	51
Hình 2.5 Hai mô hình không tương thích với hệ điều hành WSN.....	56
Hình 2.6 Mô hình lập trình dựa trên sự kiện.....	57
Hình 2.7. Nút Mica.....	60
Hình 2.8 Nút cảm biến EYES.....	61
Hình 2.9 Btnode.....	61
Hình 3.1 Ba loại bộ chứa trong một mạng cảm biến một bước nhảy đơn giản.....	64
Hình 3.2 Mạng đa bước nhảy.....	65
Hình 3.3 Nhiều nguồn/nhiều bộ chứa.....	67
Hình 3.4 Bộ chứa di động dịch chuyển qua mạng cảm biến.....	67
Hình 3.5 Vùng các cảm biến tìm kiếm sự kiện.....	68
Hình 3.6. Ví dụ tập hợp.....	72
Hình 3.7 Hai lựa chọn giao diện dịch vụ với ngăn xếp giao diện.....	76
Hình 3.8 WSN với nút cổng vào ra, có thể truy cập đến đối tượng ở xa qua Internet.....	78
Hình 3.9 Giao tiếp WSN – Internet.....	79
Hình 3.10 Giao tiếp Internet-WSN.....	79
Hình 3.11 Kết nối giữa hai WSN qua Internet.....	80
Hình 4.1 Phổ điện từ - tần số vô tuyến.....	84
Hình 4.2 Ví dụ về ASK.....	86
Hình 4.3 Ví dụ về PSK.....	87
Hình 4.4 Ví dụ về FSK.....	87
Hình 4.5 Mô tả hiện tượng truyền sóng.....	88
Hình 4.6 Truyền sóng đa đường.....	89
Hình 4.7 Tốc độ lỗi bit của PSK và FSK nhị phân được tách sóng nhất quán.....	92
Hình 4.8 Ví dụ về trải phổ dây trực tiếp.....	95
Hình 4.9 Ví dụ về lấy mẫu và đồng bộ.....	97
Hình 5.1 Tình huống đầu cuối ẩn (các vòng tròn là dải truyền dẫn và nhiễu).....	103
Hình 5.2 Bắt tay RTS/CTS trong IEEE 802.11.....	108
Hình 5.3 Hai vấn đề trong bắt tay RTS/CTS.....	108
Hình 5.4 Sơ đồ thức dậy định kỳ.....	111
Hình 5.5 Chu trình làm việc STEM của một nút đơn.....	112
Hình 5.6 Nguyên lý S-MAC.....	113
Hình 5.7 Khung S-MAC và đặt NAV.....	115
Hình 5.8 Giao thức thiết bị trung gian với các bộ hoà giải không ràng buộc.....	116
Hình 5.9 Nhiễu giữa các nhóm.....	121
Hình 5.10 Tổ chức của chu kỳ LEACH.....	122
Hình 5.11 SMACS: cài đặt tuyến cho hai nút đơn.....	123
Hình 5.12 Các ứng dụng của Zigbee.....	125
Hình 5.13 Các kiểu mạng của ZigBee.....	126
Hình 5.14 Cấu trúc siêu khung của IEEE 802.15.4.....	127
Hình 5.15 Bắt tay giữa bộ điều phối và thiết bị khi thiết bị tìm kiếm gói tin.....	129
Hình 6.1 Format của một khung lớp vật lý IEEE 802.11/802.11b.....	133

Hình 6.2 Định dạng gói tin ở DLL	134
Hình 6.3 Vị trí của FEC trong chuỗi gửi/nhận; thực hiện mã hoá và giải mã FEC.....	138
Hình 6.4 Chi phí năng lượng của mã Reed-Solomon với kích thước khối và tỉ lệ mã khác nhau.....	140
Hình 6.5 Hoạt động của mã chập	141
Hình 6.6 Tiêu thụ năng lượng trong khi truyền 10 kB dữ liệu qua tuyến với suy hao đường truyền 70 dB theo các sơ đồ mã hoá khác nhau	142
Hình 6.7 Năng lượng sử dụng để truyền thành công 1 bit dữ liệu với các t khác nhau	144
Hình 6.8 Năng lượng/bit hữu dụng cho dữ liệu có kích thước $u = 100$ và $u = 500$ với tỉ lệ lỗi bit khác nhau.....	148
Hình 6.9 Năng lượng/bit hữu dụng với tỉ lệ lỗi bit p cố định và kích thước dữ liệu u thay đổi.....	148
Hình 6.10 Định khung truyền thông với header/dữ liệu/kiểm tra tổng so với định khung kiểu kiểm tra tổng trung gian	151
Hình 6.11 Tiêu thụ năng lượng và trễ mong đợi của các gói thông tin có kích thước khác nhau L (theo byte) và các tỉ lệ lỗi trạng thái tốt khác nhau.	153
Hình 6.12 Đường cong biểu thị xác suất nhận tin cho các gói tin được tạo ra từ nút trung tâm với hai công suất khác nhau.....	154
Hình 6.13 Tỉ lệ nhận tin ở các bộ thu có khoảng cách khác nhau	155
Hình 6.14 Mô tả vấn đề ước lượng.....	156
Hình 7.1 Ví dụ cho sự phân chia của mạng.....	161
Hình 7.2 Ví dụ cho sự phân chia của mạng.....	163
Hình 7.3 “Xác suất sinh nhật” mà k ngoài $n = 2^{14}$ trạm lựa chọn các địa chỉ ngẫu nhiên không có các xung đột.....	166
Hình 8.1 Xác định góc tới của một sự kiện âm thanh ở xa bằng một mạng các cảm biến âm thanh.	174
Hình 8.2 Một phần sơ đồ của hoạt động trong đồng bộ máy phát/máy thu	182
Hình 8.3 LTS đa bước nhảy phân tán.....	185
Hình 8.4 Sự đồng bộ máy phát/máy thu trong TPSN.....	190
Hình 8.5 Ví dụ RBS: hai nút i và j và một máy phát R	192
Hình 8.6 Nhiều miền quảng bá.....	196
Hình 8.7 Tích hợp gói tin, chuyển tiếp và biến đổi nhãn thời gian	196
Hình 8.8 Đồng bộ thời gian: đồng bộ trong một miền quảng bá đơn lẻ.....	199
Hình 9.1 Xác định vị trí của các nút cảm biến với sự hỗ trợ từ một số các điểm neo; không phải tất cả các nút là cần thiết trong giao tiếp với tất cả các nút neo.	203
Hình 9.2 Phép đặc tam giác bởi ba vòng tròn giao nhau.....	205
Hình 9.3 Bản chất ngẫu nhiên của việc ánh xạ các giá trị RSSI tới các khoảng cách [25]. ...	206
Hình 9.4 Góc tạo bởi hai nút neo.....	208
Hình 9.5 Ước lượng khoảng cách Euclidean trong sự vắng mặt của kết nối trực tiếp [22] ...	213
Hình 9.6 Đa bên lập: các nút A, B, và C có thể xác định các vị trí của nó trong vài phép lập (các mũi tên đậm chỉ các đường nối mà phạm vi của chúng được sử dụng trong một phép lập đã cho)	214
Hình 9.7 Các kịch bản còn phải bàn cho đa bên lập.....	215
Hình 9.8 Sự xử lý theo xác suất các vị trí của nút [25]	216
Hình 10.1 Cấu trúc liên kết mạng trong một mạng cảm biến không dây được triển khai với mật độ cao.....	219
Hình 10.2 Cấu trúc liên kết mạng thưa hơn sau khi giảm công suất phát.	220
Hình 10.3 Giới hạn cấu trúc liên kết mạng bằng cách sử dụng một xương sống.....	220
Hình 10.4 Sử dụng các phân nhóm để chia một sơ đồ thành các phần.	221
Hình 10.5 Mục tiêu tối ưu hóa điều khiển cấu trúc cấu trúc liên kết mạng: xác suất của mạng được kết nối so với kích thước trung bình của thành phần cấu thành lớn nhất.	223

Hình 10.6 Cấu trúc của RNG: vùng tối phải không chứa các nút nào khác cho hai nút được kết nối.	227
Hình 10.7 Minh họa vùng tiếp sóng của nút i với nút r (vùng tối chỉ thị các điểm nơi mà tiếp sóng có hiệu suất cao hơn thay vì truyền thông trực tiếp).....	228
Hình 10.8 Hai kịch bản làm cách nào xây dựng các nút lân cận cho nút i ngoài các vùng tiếp sóng với các nút khác.	229
Hình 10.9 Minh họa một thuật toán tập hợp chi phối đơn giản dựa trên việc phát triển một cây (các đường nét đậm chỉ các cạnh trong cây, các nút màu đen tạo thành tập hợp chi phối.	231
Hình 10.10 Sơ đồ ví dụ ở đó một lựa chọn khám phá tham vọng dựa trên lợi ích một bước bị thất bại	232
Hình 10.11 Cấu trúc tập hợp chi phối tham vọng với và không với việc nhìn về phía trước.	233
Hình 10.12 Tính đến sự cần thiết để kết nối hai nút màu đen phá vỡ mối quan hệ giữa hai nút A và B.	234
Hình 10.13 Sơ đồ ví dụ cho việc đánh dấu các nút với các nút lân cận không được kết nối.	236
Hình 10.14 Một sơ đồ ví dụ với một tập hợp độc lập cực đại	238
Hình 10.15 Tập hợp độc lập cực đại tạo ra các phân nhóm chồng nhau hoặc không chồng nhau	239
Hình 10.16 Hai phân nhóm được nối bởi hai cổng chính phân tán	239
Hình 10.17 Thuật toán cơ bản để xác định các tập hợp độc lập, sử dụng các số nhận dạng nút như xếp hạng (các nút màu trắng là chưa được xác định, các nút màu đen là các chỉ huy phân nhóm, các nút màu xám là các thành viên phân nhóm.	241

DANH MỤC BẢNG BIỂU

Bảng 2.1 Mật độ năng lượng của các loại pin chính và phụ khác nhau.	47
Bảng 2.2 Một số tham số về tiêu thụ năng lượng của bộ thu phát (N/A : không cho phép)	54
Bảng 2.3 Ví dụ về một số đặc tính của các cảm biến	55
Bảng 4.1 Một số dải ISM	84
Bảng 6.1 Số lượng dữ liệu k tương ứng với số lượng bit có thể hiệu chỉnh t khác nhau trong khối $n=1023$ bit đối với mã BCH	144
Bảng 7.1 Tập hợp các toán tử	169

CÁC THUẬT NGỮ VIẾT TẮT

Chữ viết tắt	Tiếng Anh	Tiếng Việt
ADC	Analog/Digital Convert	Bộ biến đổi tương tự/số
ANDA	Ad hoc Network Design Algorithm	Thuật toán thiết kế mạng Ad hoc
API	Application Programming Interface	Giao diện lập trình ứng dụng
ARQ	Automatic Repeat Request	Yêu cầu lặp lại tự động
ASIC	Application-Specific Integrated Circuit	Mạch tích hợp chuyên dụng
ASK	Amplitude Shift Keying	Khoá dịch biên
AWGN	Additive White Gaussian Noise	Tạp âm Gaussian cộng sinh
BCH	Bose – Chaudhuri – Hocquenghem	Mã BCH
BER	Bit-Error Rate	Tỉ số lỗi bit
BPSK	Binary Phase Shift Keying	Khoá dịch pha nhị phân
BSC	Binary Symmetric Channel	Kênh đối xứng nhị phân
CCA	Clear Channel Assessment	Kiểm tra kênh truyền có rỗi không
CCK	Complementary Code Keying	Điều chế khoá mã bù
CDMA	Code Division Multiple Access	Đa truy cập phân chia theo mã
CIR	Carrier to Interference Ratio	Tỉ số tín hiệu/nhiều
CPU	Central Processing Unit	Khối xử lý trung tâm
CRC	Cyclic Redundancy Check	Kiểm tra tính dư tuần hoàn
CSMA	Carrier Sense Multiple Access	Đa truy cập cảm nhận sóng mang
CTS	Clear To Send	Chắc chắn để gửi
DAC	Digital/Analog Converter	Bộ biến đổi số/tương tự
DAD	Duplicate Address Detection	Phát hiện địa chỉ dup
DBPSK	Differential Binary Phase Shift Keying	Khoá dịch pha nhị phân vi sai
DCF	Distributed Coordination Function	Hàm tọa độ phân bố

DHT	Distributed Hash Table	Bảng băm phân bố
DLL	Data Link Layer	Lớp liên kết dữ liệu
DPM	Dynamic Power Managemen	Quản lý năng lượng động
DQPSK	Differential Quaternary Phase Shift Keying	Khoá dịch pha tứ phân vi sai
DSP	Digital Signal Processor	Bộ xử lý tín hiệu số
DSSS	Direct Sequence Spread Spectrum	Trải phổ dây trực tiếp
DVS	Dynamic Voltage Scaling	Dải điện áp động
EEPROM	Electrically Erasable Programmable Read-Only Memory	Bộ nhớ chỉ đọc lập trình được xoá bằng điện
EHF	Extremely High Frequency	Tần số cực cao
FDMA	Frequency Division Multiple Access	Đa truy cập phân chia theo tần số
FEC	Forward Error Correction	Hiệu chỉnh phòng lỗi
FFD	Full Function Device	Thiết bị chức năng đầy đủ
FFT	Fast Fourier Transform	Biến đổi Fourier nhanh
FHSS	Frequency Hopping Spread Spectrum	Trải phổ nhảy tần
FIFO	First In First Out	Vào trước – Ra trước
FPFA	Field-Programmable Gate Array	Mảng cổng logic khả trình
FSK	Frequency Shift Keying	Khoá dịch tần
GPS	Global Positioning System	Hệ thống định vị toàn cầu
GTS	Guaranteed Time Slot	Khe thời gian đảm bảo
IEEE	Institute of Electrical and Electronics Engineers	Viện công nghệ kỹ thuật điện -điện tử
IF	Intermediate Frequency	Tần số trung gian
ISI	InterSymbol Interference	Nhiễu liên ký hiệu
ISM	Industrial, Scientific, and Medical	Công nghiệp, khoa học và y tế

LEACH	Low-Energy Adaptive Clustering Hierarchy	Phân cấp nhóm thích ứng công suất thấp
LED	Light-Emitting Diode	Diode phát quang
LNA	Low Noise Amplifier	Bộ khuếch đại tạp âm thấp
LOS	Line Of Sight	Đường nhìn thẳng
MAC	Medium Access Control	Điều khiển truy cập môi trường
MANET	Mobile Ad Hoc Network	Mạng ad hoc di động
MCDS	Minimum Connected Dominating Set	Tập hợp chi phối được nhỏ nhất
MDS	MultiDimensional Scaling	Chia tỷ lệ đa chiều
MEMS	MicroElectroMechanical System	Hệ thống vi cơ điện tử
MLE	Maximum Likelihood Estimation	Cực đại hoá khả năng để ước lượng tham số
MPDU	MAC-layer Protocol Data Unit	Khối dữ liệu giao thức lớp MAC
NAT	Network Address Translation	Dịch địa chỉ mạng
NAV	Network Allocation Vector	Vector định vị mạng
NLOS	Non Line Of Sight	Không có đường nhìn thẳng
OOK	On-Off-Keying	Khoá đóng – mở
PAN	Personal Area Network	Mạng cá nhân
PA	Power Amplifier	Bộ khuếch đại công suất
PDA	Personal Digital Assistant	Thiết bị hỗ trợ cá nhân kỹ thuật số
PHY	Physical Layer	Lớp vật lý
PPDU	Physical-layer Protocol Data Unit	Khối dữ liệu giao thức lớp vật lý
PPM	Pulse Position Modulation	Điều chế vị trí xung
PSD	Power Spectral Density	Mật độ phổ công suất
PSK	Phase Shift Keying	Khoá dịch pha
QAM	Quadrature Amplitude Modulation	Điều chế biên độ vuông góc
QPSK	Quaternary Phase Shift Keying	Khoá dịch pha tứ phân

QoS	Quality of Service	Chất lượng của dịch vụ
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RFD	Reduced Function Device	Thiết bị giới hạn chức năng
RFID	Radio Frequency Identifier	Bộ nhận dạng tần số vô tuyến
RF	Radio Frequency	Tần số vô tuyến
GRT	Geography RouTing protocol	Giao thức định tuyến địa lý
RISC	Reduced Instruction Set Computer	Máy tính có bộ lệnh rút gọn
ROM	Read-Only Memory	Bộ nhớ chỉ đọc
RSSI	Received Signal Strength Indicator	Bộ chỉ thị cường độ tín hiệu nhận
RS	Reed–Solomon	Mã Reed–Solomon
RTS	Request To Send	Yêu cầu để gửi
SDMA	Space Division Multiple Access	Đa truy cập phân chia không gian
SFD	Start Frame Delimiter	Bắt đầu dấu tách khung
SINR	Signal to Interference and Noise Ratio	Tỉ số tín hiệu/nhiều và tạp âm
SMACS	Self-Organizing Medium Access Control for Sensor Networks	Điều khiển truy cập môi trường tự tổ chức của các mạng cảm biến
SNR	Signal-to-Noise Ratio	Tỉ số tín hiệu/tạp âm
STEM	Sparse Topology and Energy Management	Giao thức cấu hình mỏng và vấn đề quản lý năng lượng
TCP	Transmission Control Protocol	Giao thức điều khiển truyền dẫn
TDMA	Time Division Multiple Access	Đa truy cập phân chia thời gian
TRAMA	Traffic-Adaptive Medium Access	Truy cập môi trường lưu lượng thích ứng
UTM	Universal Transverse Mercator	Các vị trí trong các tọa độ
UWB	UltraWideBand	Băng siêu rộng
VLF	Very Low Frequency	Tần số rất thấp
WLAN	Wireless Local Area Network	Mạng không dây cục bộ

WPAN	Wireless Personal Area Network	Mạng không dây cá nhân
WRP	Wireless Routing Protocol	Giao thức định tuyến không dây
WSDL	Web Service Description Language	Ngôn ngữ mô tả dịch vụ Web
WSN	Wireless Sensor Network	Mạng cảm biến không dây

CHƯƠNG 1. GIỚI THIỆU CHUNG

Mục đích của chương

Các ứng dụng phải được định hình và xây dựng dựa trên công nghệ mà nó sử dụng. Điều này đặc biệt đúng với mạng cảm biến không dây vì nó có thể được xây dựng theo các cách khác nhau. Chương này sẽ trình bày khái niệm của mạng cảm biến không dây và một số ứng dụng của nó. Từ các ví dụ đã trình bày sẽ phân thành các loại hoặc các lớp ứng dụng. Tiếp theo là những thách thức đối với các ứng dụng này và đưa ra lí do tại sao công nghệ hiện nay không thể thoả mãn những thách thức này.

Phần cuối của chương là những đánh giá về các loại ứng dụng này đối với mạng cảm biến không dây và đưa ra những gợi ý về giải pháp công nghệ cho chúng, cả phần cứng và kỹ thuật mạng.

1.1 Cơ sở hạ tầng cho mạng cảm biến không dây

Một mạng cảm biến bao gồm số lượng lớn các nút cảm biến được phân bố cả bên trong và bên cạnh hiện tượng. Vị trí của các nút cảm biến không cần phải thiết kế hoặc xác định trước. Điều này cho phép phân bố ngẫu nhiên trong các địa hình phức tạp hoặc các hoạt động trợ giúp thảm họa. Điều này cũng có nghĩa là các giao thức của mạng cảm biến và các thuật toán phải có khả năng tự tổ chức. Một đặc điểm quan trọng khác của các mạng cảm biến là khả năng phối hợp giữa các nút cảm biến. Các nút cảm biến được gắn một bộ xử lý bên trong. Thay vì gửi đi số liệu thô tới nút đích, chúng sử dụng khả năng xử lý để thực hiện các tính toán đơn giản và chỉ truyền số liệu đã được xử lý theo yêu cầu.

Chọn đường trong WSN rất khó khăn do các đặc tính riêng phân biệt của mạng này với các mạng vô tuyến khác như các mạng ad-hoc hoặc các mạng tế bào. *Trước hết*, do số lượng nút cảm biến là khá lớn nên không thể xây dựng một quy tắc cho địa chỉ toàn cục khi triển khai vì phần điều khiển cho việc thiết lập ID là cao. Vì vậy, các giao thức dựa trên IP truyền thống có thể không áp dụng được cho WSN. *Thứ hai*, khác với các mạng thông tin nói chung, hầu hết các ứng dụng của mạng cảm biến yêu cầu truyền số liệu cảm biến từ nhiều nguồn tới một nút gốc. *Thứ ba*, các nút cảm biến bị hạn chế về công suất, khả năng xử lý và dung lượng nhớ. *Thứ tư*, trong hầu hết các ứng dụng, các nút trong WSN thường có vị trí cố định. Tuy nhiên, trong một số ứng dụng, các nút cảm biến có thể được phép di chuyển và thay đổi vị trí (mặc dù với độ di chuyển rất thấp). *Thứ năm*, các mạng cảm biến thường phụ thuộc vào ứng dụng. *Thứ sáu*, vị trí của các nút cảm biến đóng vai trò quan trọng vì việc lựa chọn số liệu thường dựa vào vị trí. Hiện nay chưa thích hợp cho việc sử dụng các phần cứng của hệ thống định vị toàn cầu (GPS) cho mục đích này. Các phương pháp xác định vị trí của các nút cảm biến thường dựa vào cường độ tín hiệu từ một số điểm xác định. *Cuối cùng*, số liệu được lựa chọn bởi các nút cảm biến trong WSN thường dựa vào hiện tượng chung, do đó sẽ có độ dư thừa. Các giao thức chọn đường phải khắc phục được độ dư thừa này để sử dụng hiệu quả băng thông. Tóm lại, phương pháp chọn đường trong WSN cần phải quan tâm đến các đặc tính riêng của WSN cùng với các yêu cầu về ứng dụng và cấu trúc.

Nhìn chung, các thiết bị tính toán đều phải xử lý thông tin từ cách thức đơn giản đến

phức tạp. Trong nhiều ứng dụng, ví dụ như các ứng dụng văn phòng, các thiết bị tính toán này thường được sử dụng để xử lý thông tin liên quan đến con người, như vậy đối tượng ở đây quan hệ gián tiếp với môi trường vật lý.

Trong một lớp ứng dụng khác, môi trường vật lý lại tập trung vào đối tượng. Tính toán được sử dụng để điều khiển đối tượng thông qua các quá trình vật lý. Ví dụ như điều khiển khiến các quá trình hoá học trong nhà máy để điều chỉnh nhiệt độ và áp suất. Ở đây, quá trình tính toán sẽ được kết hợp với điều khiển và nó được gọi là *nhúng* trong môi trường. Không giống như các loại hệ thống trước kia, các *hệ thống nhúng* này thường không dựa vào sự tương tác với con người và vì thế, nó có một số yêu cầu về mặt kỹ thuật để làm được việc đó. Trong các hệ thống lớn thì nó có quan hệ mật thiết với nhiệm vụ điều khiển.

Các hệ thống nhúng đã được biết đến và sử dụng trong một thời gian dài (trên thực tế, trên 98% các thiết bị tính toán sử dụng nhúng). Chúng đã xuất hiện hàng ngày, hàng giờ quanh cuộc sống của chúng ta và phát triển một cách nhanh chóng. Hầu hết các thiết bị gia dụng đều sử dụng nhúng, ví dụ như máy giặt, máy quay phim, điện thoại... Trong các ứng dụng này, hệ thống nhúng dựa trên sự tương tác với con người.

Xu hướng công nghệ này không chỉ xuất hiện trong các thiết bị lớn như máy giặt mà nó còn được thực hiện trong các thiết bị nhỏ hơn, thậm chí trong cả hàng tạp hoá. Nhiều thiết bị khác nhau sẽ kết hợp với thông tin xử lý để điều khiển các quá trình vật lý và tương tác với người sử dụng. Công nghệ này được đưa ra bởi Marc Weiser. Bằng cách tích hợp tính toán và điều khiển trong một môi trường vật lý, mô hình tương tác được biết đến ở đây là người – người, người – máy và máy – máy và cuối cùng là người – môi trường vật lý. Tương tác với môi trường vật lý sẽ quan trọng hơn là chỉ làm việc với các số liệu.

Để thực hiện điều này, yêu cầu ngoài việc tính toán và điều khiển là thông tin. Tất cả các nguồn tin này có thể chuyển thông tin thành vị trí của cơ cấu chấp hành hoặc người sử dụng và nó có thể kết hợp lại với nhau để tạo nên một bức tranh chính xác về môi trường thực tế khi có yêu cầu. Đối với một số ứng dụng, như các mạng cảm biến và cơ cấu chấp hành, rất dễ xây dựng hệ thống với các công nghệ mạng hữu tuyến đã có. Tuy nhiên, với một số ứng dụng, có rất nhiều trở ngại để thực hiện điều này như giá thành cao (lên đến 200 USD/ 1 cảm biến). Đặc biệt là trong một hệ thống có số thiết bị rất lớn, chúng ta có thể hình dung ra được sự phức tạp của nó, vấn đề bảo dưỡng đường dây, dây dẫn cản trở sự di chuyển của đối tượng, ngăn không cho cảm biến/cơ cấu chấp hành lại gần đối tượng điều khiển. Do đó, thông tin vô tuyến giữa các thiết bị này, trong nhiều ứng dụng, là một yêu cầu tất yếu.

Một lớp mạng mới đã xuất hiện trong những năm gần đây, đó là mạng cảm biến không dây (WSN - Wireless Sensor Network). Mạng này bao gồm những nút đơn có thể tương tác với môi trường bằng cách cảm biến và điều khiển các tham số vật lý. Những nút này cộng tác hoàn toàn với nhau để hoàn thành nhiệm vụ của nó vì một nút đơn không thể làm được gì và chúng sử dụng thông tin vô tuyến để thực hiện điều này. Các nút thuộc mạng này ít nhất cũng chứa một số phép tính, giao tiếp vô tuyến và có chức năng cảm biến hoặc điều khiển. Mặc dù thực chất mạng này bao gồm các cơ cấu chấp hành nhưng thuật ngữ mạng cảm biến không dây lại là tên gọi phổ biến. Đôi khi nó còn được gọi là **mạng cảm biến và cơ cấu chấp hành không dây** (wireless sensor and actuator network).

WSN hữu dụng khi nó dùng để hỗ trợ cho nhiều ứng dụng thực tế khác nhau. Mạng này tồn tại nhiều thách thức trong quá trình nghiên cứu và nhiều vấn đề về mặt công nghệ vì nó rất phức tạp. Do đó không thể có cách duy nhất để phân loại một cách rõ ràng các WSN và cũng không thể tồn tại một giải pháp để thực hiện các yêu cầu thiết kế. Ví dụ, trong rất nhiều ứng dụng của WSN, các nút đơn trong mạng không thể dễ dàng kết nối với nguồn cung cấp mà phải dựa vào nguồn pin gắn trên nó. Trong các ứng dụng này, việc sử dụng năng lượng có hiệu quả là nhiệm vụ rất quan trọng, sao cho nó có thể hoạt động trong thời gian lâu nhất có thể. Trong một số ứng dụng không có nguồn cung cấp thì các tham số khác lại đóng vai trò quan trọng hơn, ví dụ như độ chính xác của kết quả nhận được. Kích thước và giá thành của một nút đơn cũng được xem xét đến trong một số ứng dụng. Về mặt kích thước, nó phải đủ nhỏ để nằm trên nguồn pin còn về mặt giá thành, nó phụ thuộc trực tiếp vào chất lượng cảm biến của nút. Thông thường, độ chính xác của kết quả có thể nhận được từ các nút đơn. Số lượng, giá thành và độ chính xác của các nút đơn có liên quan đến nhau khi so sánh một hệ thống có nhiều nút cảm biến với một mạng tập trung. Nút đắt tiền sẽ cho độ chính xác cao hơn. Sử dụng nhiều cảm biến đơn giản và gần với hiện tượng cần nghiên cứu sẽ tạo ra một kiến trúc hệ thống vừa đơn giản, vừa sử dụng năng lượng có hiệu quả so với mạng phân bố. Cách hiểu sâu hơn mạng cảm biến không dây là hiện thực hoá nó.

1.2 Mạng cảm biến không dây và ứng dụng

Đặc trưng và cấu hình mạng cảm biến

Một nút trong mạng WSN thông thường bao gồm 2 phần: phần cảm biến (sensor) hoặc điều khiển và phần giao tiếp vô tuyến (RF transceiver). Do số lượng nút trong WSN là lớn và không cần các hoạt động bảo trì, nên yêu cầu thông thường đối với một nút mạng là giá thành thấp (10 - 50 usd) và kích thước nhỏ gọn (diện tích bề mặt vài đến vài chục cm^2).

Do giới hạn về nguồn năng lượng cung cấp (pin...), giá thành và yêu cầu hoạt động trong một thời gian dài, nên vấn đề tiêu thụ năng lượng là tiêu chí thiết kế quan trọng nhất trong mạng cảm biến:

- **Lớp vật lý** (physical layer) (sẽ được trình bày trong chương 4) tương đối đơn giản, gọn nhẹ do ràng buộc về kích thước và khả năng tính toán của nút. Kỹ thuật điều chế tín hiệu số như QPSK, FSK để cải thiện hiệu suất bộ khuếch đại công suất. Các kỹ thuật mã hóa sửa sai phức tạp như Turbo Codes không được sử dụng, kỹ thuật trải phổ được sử dụng để cải thiện SNR ở thiết bị thu và giảm tác động của fading của kênh truyền...

- **Lớp MAC** (sẽ được trình bày trong chương 5): kỹ thuật đa truy cập TDMA hoặc CSMA-CA hiệu chỉnh với mục đích giảm năng lượng tiêu thụ.

WSN thường được triển khai trên một phạm vi rộng, số lượng nút mạng lớn và được phân bố một cách tương đối ngẫu nhiên, các nút mạng có thể di chuyển làm thay đổi sơ đồ mạng... do vậy WSN đòi hỏi một sơ đồ mạng (topology) linh động (ad-hoc, mesh, star ...) và các nút mạng có khả năng tự điều chỉnh, tự cấu hình (auto-reconfigurable).

Trong một số WSN thông dụng (giám sát, cảm biến, môi trường ...) địa chỉ ID các nút chính là vị trí địa lý và giải thuật định tuyến dựa vào vị trí địa lý này gọi là **giao thức định tuyến địa lý** (GRT - Geography Routing protocol). Đối với mạng có số lượng nút lớn, sơ đồ mạng không ổn định ... GRT giúp đơn giản hóa giải thuật tìm đường, giảm dữ liệu bảng định tuyến (routing table) lưu trữ tại các nút. GRT phù hợp với các WSN cố định, tuy nhiên đối với

các nút di động (địa chỉ ID node thay đổi) giao thức định tuyến trở nên phức tạp và không ổn định.

- **Phân nhóm:** phân chia mạng diện rộng (hàng trăm, hàng ngàn nút) thành các nhóm để ổn định cấu hình mạng, đơn giản hóa giải thuật định tuyến, giảm xung đột (collision) khi truy cập vào kênh truyền nên giảm được năng lượng tiêu thụ, đơn giản hóa việc quản lý mạng và cấp phát địa chỉ cho từng nút mạng (theo nhóm).

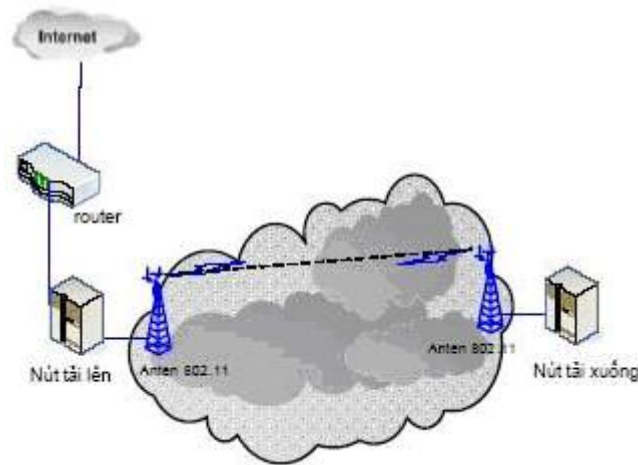
Do giới hạn khả năng tính toán của từng nút mạng cũng như để tiết kiệm năng lượng, WSN thường sử dụng các phương pháp tính toán và xử lý tín hiệu phi tập trung (giảm tải cho nút gần hết năng lượng) hoặc gửi dữ liệu cần tính toán cho các trạm gốc (có khả năng xử lý tín hiệu mạnh và ít ràng buộc về tiêu thụ năng lượng).

Khi cảm biến không dây trở thành thương phẩm quen thuộc trên thị trường thì việc lựa chọn giữa các loại cảm biến và kiểu mạng mới và cũ làm đau đầu các chuyên gia và làm cho họ phải xem xét lại những chiến lược xây dựng kiến trúc mạng mà trước đây đã từng bị gạt sang một bên. Sau đây, chúng ta cùng tìm hiểu về 3 kiểu mạng cơ bản sử dụng cho cảm biến là kiểu mạng điểm - điểm (point to point), điểm - đa điểm và mạng đa điểm - đa điểm.

Cấu trúc mạng điểm - điểm

Là kiểu kết nối đơn giản nhất, hai nút truyền thông qua hai anten thu phát công suất cao hướng trực tiếp với nhau. Trên lý thuyết, kiểu mạng này là an toàn nhất vì chỉ có 1 điểm trong mạng có khả năng xảy ra sự cố, đó là điểm mạng chủ hay còn gọi là host. Để nâng cao khả năng của hệ thống mạng chỉ cần bổ sung thêm 1 host dự phòng. Tuy nhiên, việc nối dây 2 host lại là vấn đề đáng bàn. Một số cấu trúc mạng này hỗ trợ **tín hiệu điều biến tần** cho dây dẫn để truyền tải tín hiệu của nhiều cảm biến trên các kênh FM riêng biệt. Một số chuẩn (như HART bus) hỗ trợ đường truyền tín hiệu số song công cho những đường truyền tín hiệu tương tự sẵn có trong một số nhà máy công nghệ cũ. Kiểu kiến trúc như vậy xóa nhòa danh giới khác biệt giữa cấu trúc mạng điểm - điểm và cấu trúc mạng đa điểm.

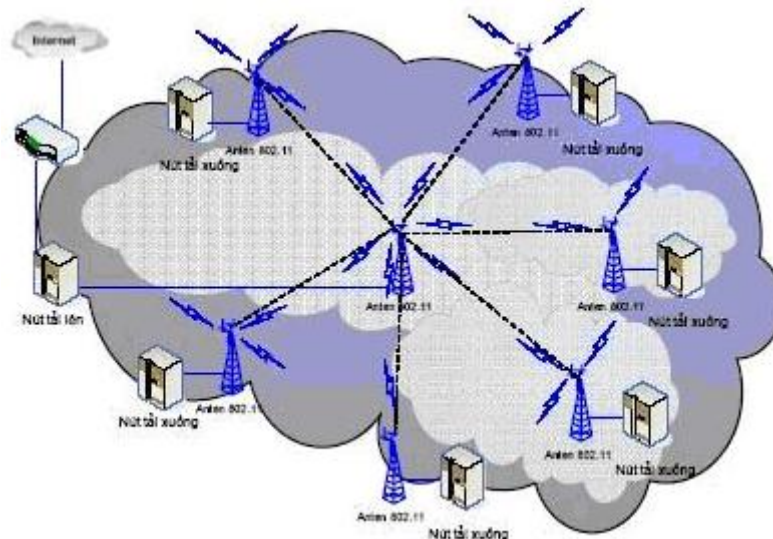
Những kiểu mạng không dây đầu tiên sử dụng tín hiệu tần số radio (RF - Radio Frequency) đơn giản cho cấu trúc mạng điểm - điểm. Cấu trúc mạng này sử dụng modem RF ở 2 đầu mạng. Một cái đặt tại đầu phát làm nhiệm vụ chuyển tín hiệu RS-232 thành tín hiệu radio và cái còn lại đặt tại đầu thu làm nhiệm vụ chuyển tín hiệu về trạng thái vốn dĩ của nó. Fluke (Everett, Washington) đã phát triển một loại von kế số có thể nhận tín hiệu điện áp và truyền đi qua một kênh tần số radio chuyên biệt. Tuy nhiên, cách này bị nghi ngờ về độ tin cậy vì thiết bị này được thiết kế cho những tín hiệu FM mã hóa đơn giản. Còn trong môi trường nhà máy, các yếu tố gây nhiễu có thể làm giảm đáng kể chất lượng của tín hiệu, do vậy mạng trở nên không đáng tin cậy, trừ khi nó được chăm sóc rất chu đáo. Kiểu mạng LAN không dây cũng được sử dụng trong cấu trúc mạng này nhưng nó chỉ tỏ ra hợp với môi trường văn phòng hơn là nhà máy. Những nhà thiết kế sử dụng hệ thống thu thập dữ liệu từ xa kết hợp với cấu trúc mạng này bằng cách dùng bộ gom dữ liệu và đẩy dữ liệu này tới bộ truyền tín hiệu radio để truyền tới host, tại đây, tín hiệu sẽ được đưa trở về hình ảnh ban đầu vốn có của nó.



Hình 1.1 Mạng điểm – điểm

Điểm – Đa điểm (Point-to-Multipoints)

Kết nối được chia sẻ giữa **nút tải lên** (Nút kết nối tới mạng Internet thông qua đường truyền hữu tuyến để cung cấp kết nối Internet cho toàn mạng) dùng Anten đa hướng với các **nút tải xuống** (Nút kết nối tới mạng và có khả năng phục vụ cả kết nối hữu tuyến và vô tuyến cho mạng) hoặc **nút lặp** (Nút kết nối vào mạng và không dùng để phục vụ các client chỉ đóng vai trò là nút trung gian lặp tín hiệu) với Anten thu công suất cao. Cấu hình mạng này dễ triển khai hơn cấu hình điểm– điểm vì khi thêm một thuê bao mới chỉ cần lắp đặt thêm thiết bị tại khu vực thuê bao chứ không phải lắp tại nút tải lên. Tuy vậy, các trạm thu phải nằm trong phạm vi phủ sóng và có đường nhìn thẳng với trạm phát sóng gốc. Các vật cản như cây cối, nhà cửa, đồi núi, ... sẽ góp phần làm cấu hình mạng lưới điểm – đa điểm hoạt động không hiệu quả.



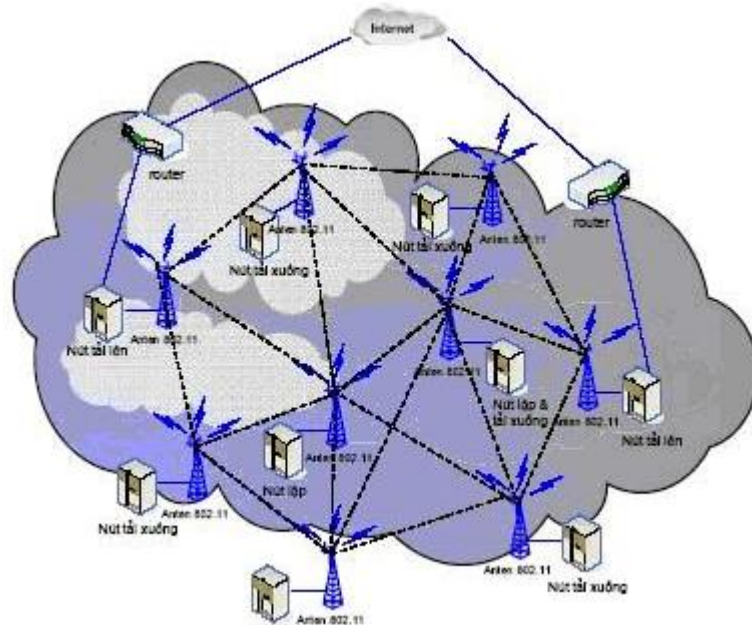
Hình 1.2 Mạng điểm – đa điểm

Đa điểm – Đa điểm (Multipoints to Multipoints)

Mỗi nút có vai trò không chỉ là điểm truy nhập cho các trạm mà còn làm nhiệm vụ chuyển tiếp dữ liệu.

Cấu hình này có độ tin cậy mạng cao nhất do các nút có sự liên thông với nhau, một nút

chỉ cần có kết nối với một nút bất kỳ mà không cần phải có kết nối trực tiếp với nút tải lên như trong cấu hình điểm – đa điểm, là có thể kết nối với toàn mạng. Tuy nhiên, đổi lại giao thức tìm đường của mạng sẽ có độ phức tạp cao hơn.



Hình 1.3 Mạng đa điểm – đa điểm

Yêu cầu của mạng cảm biến không dây là công nghệ của nó phải thuận tiện với các ứng dụng hiện có và thật sự mang lại hiệu quả. Ngoài sự cần thiết là phải xây dựng một mạng giá rẻ, đơn giản trong lập trình và mạng, các nút cảm biến có tuổi thọ dài, thành phần bắt buộc và chủ yếu để phát triển các ứng dụng trên thực tế là khả năng cảm nhận và thực hiện của các cảm biến trong mạng. Đối với nhiều tham số vật lý, công nghệ cảm biến thích hợp đã tồn tại có thể được tích hợp trong một nút của WSN. Các tham số đó là nhiệt độ, độ ẩm, ánh sáng hồng ngoại và ánh sáng nhìn thấy (kể cả ánh sáng phát ra từ camera), âm thanh, rung động (ví dụ như phát hiện rối loạn địa chấn), áp suất, các cảm biến hoá học... Thậm chí nó có thể cảm nhận được các đối tượng phức tạp hơn như đồ chơi trong vườn trẻ có thể có cảm giác.

Các cơ cấu chấp hành không thể được điều khiển một cách trọn vẹn bằng một nút của mạng cảm biến không dây. Thông thường, các cơ cấu chấp hành kiểm soát một thiết bị cơ khí như một ổ đĩa servo, hoặc có thể bật/tắt một số thiết bị điện bằng role điện như bóng đèn, micro hoặc thiết bị tương tự. Trên cơ sở đó các nút có cảm biến và /hoặc cơ cấu chấp hành sẽ kết hợp với khả năng tính toán và giao tiếp để tạo ra nhiều ứng dụng khác nhau với nhiều kiểu nút khác.

Các ứng dụng cứu trợ thiên tai: Đây là một trong những ứng dụng được chú ý nhất của WSN. Hiện tượng thường gặp nhất là phát hiện cháy rừng: các nút cảm biến được trang bị nhiệt kế và có thể xác định vị trí của nó (theo hệ tọa độ tuyệt đối hoặc tương đối, tức vị trí của nó so với các nút khác). Các cảm biến này được thả vào một đám cháy, ví dụ như một khu rừng và từ một chiếc máy bay. Chúng tạo ra một “bản đồ nhiệt độ” chung của khu vực hoặc xác định chu vi của khu vực có nhiệt độ cao để có thể tiếp cận từ bên ngoài. Ví dụ tương tự là kiểm soát tai nạn tại nhà máy hóa chất.

Một số các ứng dụng cứu trợ thiên tai có điểm chung với các ứng dụng quân sự, khi các cảm biến được dùng để phát hiện, ví dụ, quân địch chứ không phải là cháy rừng. Trong ứng dụng như vậy, cảm biến nên có giá thành đủ rẻ để có thể dùng một lần vì cần phải dùng với số lượng rất lớn và yêu cầu thời gian tồn tại không quá cao.

Giám sát môi trường và lập bản đồ đa dạng sinh học: Các WSN có thể được dùng để giám sát môi trường, ví dụ như các chất ô nhiễm hóa học tại vị trí đổ rác. Liên quan mật thiết với việc giám sát môi trường là sử dụng các WSN để biết số lượng các loài động thực vật sống trong một môi trường.

Ưu điểm chính của các WSN ở đây là các cảm biến không phải giám sát, hoạt động trong thời gian dài, giao tiếp vô tuyến có thể đặt gần đối tượng được quan sát. Do các cảm biến có thể được cấu tạo đủ nhỏ để không bị lộ nên nó chỉ hơi làm phiền các động thực vật được quan sát mà thôi. Thông thường, yêu cầu về số lượng cảm biến cao hơn yêu cầu về thời gian hoạt động của nó.

Các tòa nhà thông minh: Sự lãng phí năng lượng của các tòa nhà gồm một lượng lớn năng lượng do sử dụng độ ẩm, thông gió, điều hòa nhiệt độ (HVAC) không hiệu quả. Sẽ tốt hơn nếu giám sát theo thời gian thực, có độ phân giải cao nhiệt độ, luồng không khí, độ ẩm và các thông số vật lý khác trong tòa nhà bằng WSN. Điều này có thể làm tăng một cách đáng kể mức độ thoải mái của người dân và giảm lượng tiêu thụ năng lượng. Cải thiện hiệu suất năng lượng và tăng mức độ thuận tiện là những mục đích của "tòa nhà thông minh" hướng tới. Do đó, các hệ thống hữu tuyến hiện nay như BACnet, LonWorks, hoặc KNX đã hoặc đang được triển khai. Các tiêu chuẩn này cũng bao gồm việc phát triển thành phần không dây hoặc đã kết hợp chúng trong tiêu chuẩn.

Ngoài ra, các nút cảm biến như vậy có thể được sử dụng để theo dõi mức chịu đựng cơ học của các tòa nhà trong khu hoạt động địa chấn mạnh. Bằng cách đo các thông số cơ học như tải uốn của dầm để có thể nhanh chóng xác định thông qua WSN là liệu tòa nhà có đủ an toàn để vào sau trận động đất mặc dù nó sắp sập? Đây là một lợi thế đáng kể cho nhân viên cứu hộ. Các hệ thống tương tự cũng có thể được áp dụng cho các cây cầu. Loại cảm biến khác được dùng để tìm người trong một tòa nhà bị sập và chuyển thông tin đó cho một đội cứu hộ.

Ưu điểm chính ở đây là lập bản đồ cộng tác của các thông số vật lý. Tùy thuộc vào ứng dụng đặc biệt, cảm biến có thể được trang bị thêm vào các tòa nhà hiện có (cho các ứng dụng loại HVAC) hoặc phải được đưa vào tòa nhà sắp xây dựng.

Quản lý thiết bị: Để quản lý được nhiều hơn các thiết bị trong một tòa nhà, các WSN cũng có một loạt các ứng dụng có thể. Một ví dụ đơn giản là vào không cần chìa khoá ở những nơi mọi người đeo phù hiệu, nó cho phép WSN kiểm tra những người được phép vào các khu vực đó. Ví dụ này có thể được mở rộng để phát hiện kẻ xâm nhập, ví dụ như khi phương tiện của họ ra khỏi cơ quan trong giờ làm việc. Một WSN diện rộng có thể theo dõi vị trí của một phương tiện nào đó và báo cho nhân viên an ninh. Ứng dụng này có nhiều điểm chung với các ứng dụng trong quân sự. Theo cách khác, một WSN có thể được sử dụng trong một nhà máy hóa chất để kiểm tra rò rỉ hóa chất. Đi kèm các ứng dụng này là các yêu cầu đầy thách thức như số lượng cảm biến có thể lớn, phải có tính cộng tác (như trong ví dụ theo dõi) và có thể hoạt động bằng pin trong một thời gian dài.

Bảo dưỡng và giám sát máy: Một ý tưởng là cố định các nút cảm biến ở những nơi đặt máy móc nhưng khó tiếp cận, nơi nó có thể phát hiện các chấn động và nó cho thấy sự cần thiết phải bảo trì. Trường hợp này có thể là các robot hoặc các trục xe của đoàn tàu. Các ứng dụng khác trong sản xuất như máy đặt ở những nơi có phóng xạ, chứa nhiều chất độc hại, hoặc nằm sâu dưới lòng đất... Ưu điểm chính của các WSN ở đây là hoạt động không cần dây nối nên không cần phải bảo dưỡng, các cảm biến có giá thành thấp và có thể thường xuyên được cài đặt lại. Nguồn cấp bằng dây dẫn thì lúc có, lúc không phụ thuộc vào từng ứng dụng. Nếu không có nguồn nuôi thì việc cảm biến phải hoạt động trong một thời gian dài bằng nguồn pin có năng lượng hạn chế là không thực tế và tốn kém. Thế nhưng kích thước và giá thành của các nút thường không phải là vấn đề quá quan trọng.

Trong nông nghiệp: Áp dụng WSN vào nông nghiệp làm cho việc thụ phấn và tưới tiêu được chính xác bằng cách đặt các cảm biến đo độ ẩm/ thành phần đất vào môi trường. Chỉ cần một số lượng tương đối cảm biến nhỏ là đủ, khoảng một cảm biến cho mỗi vùng có diện tích $100\text{ m} \times 100\text{ m}$. Tương tự, giám sát sâu bọ có thể được thực hiện bằng một bộ theo dõi có độ phân giải cao. Ngoài ra, chăn nuôi có thể tăng lợi nhuận từ việc gắn cảm biến cho mỗi con lợn hoặc bò để kiểm soát tình trạng sức khỏe của động vật (bằng cách kiểm tra nhiệt độ cơ thể, đếm bước...) và báo động nếu vượt quá ngưỡng cho phép.

Chăm sóc sức khỏe và y tế: Cùng cách làm tương tự, sử dụng WSN trong các ứng dụng chăm sóc sức khỏe đem lại rất nhiều lợi nhuận nhưng cũng gây tranh cãi về mặt đạo đức. Phạm vi áp dụng là từ chăm sóc sau phẫu thuật và chuyên sâu, nơi mà cảm biến trực tiếp gắn vào với bệnh nhân – thể hiện lợi thế so với hữu tuyến - để giám sát bệnh nhân trong thời gian dài (thường là người cao tuổi) và để quản lý thuốc tự động (cảm biến nhúng vào bao bì của thuốc, bật báo động khi sử dụng sai cho bệnh nhân). Do đó, các hệ thống theo dõi bệnh nhân và bác sĩ trong bệnh viện giúp tiết kiệm thời gian.

Thông tin: Các cảm biến nhúng vào trong các đường phố hay lề đường để có thể thu thập thông tin về điều kiện giao thông để từ đó đưa ra những giải pháp tốt hơn. Các cảm biến này có thể tương tác với xe đi trên đường để cảnh báo nguy hiểm về các điều kiện đường hay ách tắc giao thông phía trước.

Bên cạnh đó, các ứng dụng khác của WSN hỗ trợ cất cánh cho máy bay, trong các nhà máy xử lý nước thải, thiết bị đo đặc của phòng xử lý chất bán dẫn và các đường hầm gió, trong "nhà trẻ thông minh" (nơi có sự tương tác giữa đồ chơi và trẻ em), phát hiện của lũ lụt, giám sát môi trường sống của chim trên một hòn đảo, và cảm biến cấy vào cơ thể người (để theo dõi lượng đường hoặc võng mạc giả).

Hầu hết các ứng dụng này đều có thể được thực hiện theo một cách nào đó bằng công nghệ hiện nay mà không cần các mạng cảm biến không dây. Nhưng, các ứng dụng sẽ làm việc tốt hơn khi có nhiều thông tin hơn về giải pháp không gian và thời gian của đối tượng cần quan tâm so với những gì mà công nghệ cảm biến truyền thống có thể có được. Các mạng cảm biến không dây có phạm vi cung cấp thông tin lớn với độ chính xác yêu cầu tại cùng thời điểm và có lượng tiêu thụ năng lượng là ít nhất có thể.

Mạng cảm biến có thể sử dụng trong rất nhiều lĩnh vực. Nếu phân loại dựa vào chức năng của mạng thì có thể hình dung ứng dụng của các mạng cảm biến như sau:

- *Cảm biến môi trường* (quân sự: phát hiện mìn, chất độc, dịch chuyển quân địch...; công nghiệp: hệ thống chiếu sáng, độ ẩm, phòng cháy, chống rò rỉ...; dân dụng: hệ thống điều hòa nhiệt độ, chiếu sáng...).

- *Điều khiển* (quân sự: kích hoạt thiết bị, vũ khí quân sự ...; công nghiệp: điều khiển tự động các thiết bị, robot ...).

- *Theo dõi, giám sát, định vị* (quân sự: định vị, theo dõi sự dịch chuyển thiết bị, quân đội...).

- *Môi trường* (giám sát lũ lụt, bão, gió, mưa ... phát hiện ô nhiễm, chất thải...).

- *Y tế* (định vị, theo dõi bệnh nhân, hệ thống báo động khẩn cấp...).

- *Hệ thống giao thông thông minh*: giao tiếp giữa biển báo và phương tiện giao thông, hệ thống điều tiết lưu thông công cộng, hệ thống báo hiệu tai nạn, kẹt xe ... hệ thống định vị phương, trợ giúp điều khiển tự động phương tiện giao thông...

- *Gia đình* (nhà thông minh: hệ thống cảm biến, giao tiếp và điều khiển các thiết bị thông minh ...)

Hầu hết các ứng dụng này có đặc tính cơ bản giống nhau. Khác biệt rõ ràng giữa các nguồn dữ liệu (nút cảm biến) là mối liên hệ giữa dữ liệu đo được – bộ chứa - và các nút (nơi dữ liệu được gửi đi). Bộ chứa có khi là một phần của chính mạng cảm biến, nhưng có khi nó là các hệ thống ở ngoài mạng. Do đó, số nguồn thường nhiều hơn bộ chứa (nhưng không phải lúc nào cũng vậy) và bộ chứa không biết hoặc không chú ý đến việc nhận dạng nguồn. Ở đây dữ liệu có vai trò quan trọng hơn nhiều.

Mô hình tương tác giữa nguồn và bộ chứa được thể hiện theo một số cách sau:

Phát hiện sự kiện: Các nút cảm biến phải báo cáo với bộ chứa khi nó phát hiện sự xuất hiện của sự kiện. Những sự kiện đơn giản nhất có thể được phát hiện bởi một nút cảm biến đơn (ví dụ như vượt quá ngưỡng nhiệt độ; loại sự kiện phức tạp hơn yêu cầu phải có sự cộng tác của các bộ cảm biến lân cận hoặc thậm chí là các cảm biến ở xa để quyết định xem liệu sự kiện đã xảy ra chưa (ví dụ như đường biển thiên nhiệt độ trở nên quá dốc). Nếu nhiều sự kiện khác nhau xảy ra thì phải thực hiện phân loại sự kiện.

Các phép đo định kỳ: Các cảm biến có thể được giao nhiệm vụ thu thập các số liệu đo có tính định kỳ. Thông thường, dữ liệu có được khi một sự kiện phát hiện được kích hoạt; chu kỳ của dữ liệu phụ thuộc vào ứng dụng.

Chức năng xấp xỉ và phát hiện biên: Cách giá trị vật lý (ví dụ như nhiệt độ) thay đổi từ nơi này sang nơi khác có thể được coi là chức năng của địa điểm. WSN có thể được sử dụng để xấp xỉ chức năng chưa biết này (chỉ xét các đặc tính không gian của nó), bằng cách sử dụng một số giới hạn các mẫu lấy tại mỗi nút cảm biến riêng biệt. Bản đồ xấp xỉ này phải được thực hiện bộ chứa. Làm thế nào và khi nào cập nhật bản đồ này, độ chính xác của chức năng xấp xỉ và mức tiêu thụ năng lượng phụ thuộc vào nhu cầu của các ứng dụng.

Tương tự như vậy, vấn đề liên quan là có thể được tìm thấy các khu vực hoặc các điểm có cùng giá trị thu được. Ví dụ, để tìm những điểm thuộc đường đẳng nhiệt khi cháy rừng để

phát hiện biên giới của đám cháy thực tế. Điều này có thể được tổng quát để tìm ra "biên" trong các chức năng như vậy hoặc để gửi tin nhắn dọc theo ranh giới của các mẫu trong cả không gian và /hoặc thời gian.

Theo dõi: Nguồn của một sự kiện có thể di chuyển (ví dụ khi theo dõi trộm đột nhập). WSN có thể được sử dụng để cập nhật vị trí, đánh giá tốc độ và hướng của nguồn sự kiện đến bộ chứa. Để làm như vậy, thông thường các nút cảm biến phải hợp tác với nhau trước khi cập nhật dữ liệu cho bộ chứa.

Các nút cảm biến tương tác với nhau cả về thời gian và không gian (báo cáo sự kiện chỉ thực hiện trong một khoảng thời gian nhất định và chỉ từ các khu vực nhất định). Những yêu cầu này cũng có thể thay đổi một cách linh hoạt theo thời gian; bộ chứa phải thông báo cho các bộ cảm ứng yêu cầu của chúng trong quá trình hoạt động. Tương tác có thể diễn ra chỉ nhằm phục vụ một yêu cầu cụ thể của bộ chứa, hoặc phục vụ cho mối quan hệ lâu dài giữa các bộ cảm biến và bộ chứa.

Các ví dụ đã cho thấy sự đa dạng trong các **lựa chọn ứng dụng**. Có thể cố định vị trí đặt các nút cảm biến (như trong các ứng dụng bảo trì máy móc thiết bị) hoặc có thể đặt các nút cảm biến một cách ngẫu nhiên bằng cách thả một số lượng lớn các nút từ máy bay vào một đám cháy rừng. Ngoài ra, các nút cảm biến có thể tự di chuyển để bù đắp thiếu sót trong quá trình triển khai và đến vị trí mới mà tại đó, nhiệm vụ cảm biến của nó thực hiện được tốt hơn. Nó có thể di chuyển để giám sát thêm các đối tượng khác (trong các ứng dụng hậu cần) và để mạng thích ứng với nó.

Các ứng dụng cũng ảnh hưởng đến việc lựa chọn bảo dưỡng: Có nhất thiết phải bảo dưỡng các cảm biến và các máy móc thiết bị có liên quan? Bảo dưỡng là không phù hợp vì các mạng này chỉ được triển khai trong mạng ad hoc, theo cách thức ngắn hạn, bị giới hạn bởi thời gian thực hiện tối đa (như trong các hoạt động khắc phục thảm họa)? Các cảm biến này có chức năng không giám sát trong một thời gian dài nên không có khả năng để bảo trì?

Trong một số ứng dụng, nguồn cấp hữu tuyến có thể thực hiện được và ở đây không có vấn đề gì về cung cấp năng lượng. Đối với các nút cảm biến tự hoạt động, tùy thuộc vào yêu cầu về thời gian thực hiện mà năng lượng cung cấp có thể không đáng kể (với những ứng dụng yêu cầu thời gian thực hiện là một vài ngày) hoặc sẽ là vấn đề đối với các ứng dụng không thể bảo dưỡng mà yêu cầu thời gian thực hiện là vài năm. Rõ ràng là giá cả và kích thước mỗi nút đóng một vai trò quan trọng trong việc thiết kế cung cấp năng lượng.

1.3 Những khó khăn của mạng cảm biến không dây

Xử lý một loạt các loại ứng dụng như trên sẽ gây khó khăn cho mạng cảm biến. Tuy nhiên, vẫn có đặc điểm chung cho các ứng dụng như vậy, đặc biệt là trên khía cạnh đặc tính và các kỹ thuật yêu cầu. Hiện thực hoá các đặc tính này bằng kỹ thuật mới là khó khăn chính đối với mạng cảm biến không dây.

1.3.1 Các yêu cầu kỹ thuật

Các đặc tính sau được yêu cầu trong hầu hết các ứng dụng của mạng cảm biến không dây:

Loại hình dịch vụ: Chức năng của mạng thông tin thông thường hiển nhiên là di chuyển các bit từ nơi này đến nơi khác. Đối với một WSN, di chuyển các bit chỉ là bước kết thúc, nhưng đó không phải là mục đích thực tế. Hơn thế nữa, một WSN được kỳ vọng là cung cấp thông tin có ý nghĩa và/ hoặc hành động theo nhiệm vụ đã cho. Ngoài ra, các khái niệm phạm vi tương tác, khu vực địa lý đặc trưng hay khoảng thời gian sẽ trở nên quan trọng. Do đó yêu cầu phải có một mô hình mạng mới với các giao diện mới và cách nghĩ mới về dịch vụ mạng.

Chất lượng dịch vụ: Liên quan chặt chẽ đến các loại hình dịch vụ mạng là chất lượng dịch vụ. Thông thường, chất lượng của dịch vụ thể hiện trong các ứng dụng thuộc loại hình đa phương tiện. Trễ bị chặn hoặc băng thông tối thiểu không thích hợp cho các ứng dụng có trễ hoặc khi băng thông của dữ liệu truyền đi rất nhỏ. Trong một số ít trường hợp, việc cung cấp các gói dữ liệu có thể thừa, trong những trường hợp khác, yêu cầu độ tin cậy rất cao. Những trường hợp như vậy sẽ làm tăng vai trò của trễ khi các cơ cấu chấp hành được điều khiển theo thời gian thực bởi mạng cảm biến. Chất lượng dịch vụ chỉ căn cứ vào hệ số cung cấp gói tin là chưa đầy đủ vì nó còn liên quan đến số lượng và chất lượng thông tin về đối tượng hoặc khu vực được quan sát.

Mức chịu lỗi: Do các nút có thể hết năng lượng hoặc bị hỏng, hoặc thông tin vô tuyến giữa hai nút bị gián đoạn nên WSN có thể có lỗi. Trong những trường hợp như vậy, để hạn chế lỗi thì WSN nên có số nút nhiều hơn mức cần thiết.

Thời gian sử dụng: Trong nhiều trường hợp, các nút phải dựa vào nguồn năng lượng hạn chế (sử dụng pin). Việc thay thế nguồn pin này là không thực tế và không thể thực hiện tức thời. WSN phải sử dụng được ít nhất là trong thời gian thực hiện nhiệm vụ được giao. Do đó, tuổi thọ của WSN rất quan trọng và hiển nhiên, cách thức để WSN sử dụng năng lượng một cách có hiệu quả thực sự cần thiết.

Để thay đổi hoặc bổ sung thêm năng lượng cho WSN, nguồn có công suất hạn chế (ví dụ như pin mặt trời) cũng có thể được sử dụng cho các nút cảm biến. Thông thường, các nguồn này không đủ mạnh để đảm bảo cho hoạt động được diễn ra liên tục nhưng nó có thể dùng để sạc pin. Với cách làm như vậy thì tuổi thọ của WSN, theo lý thuyết, sẽ là vô hạn.

Tuổi thọ của WSN tỉ lệ nghịch với chất lượng dịch vụ: sử dụng nhiều năng lượng sẽ làm tăng chất lượng dịch vụ nhưng như vậy sẽ làm giảm tuổi thọ của mạng. Chính vì vậy, khi thiết kế một WSN phải cân bằng hai yếu tố này.

Định nghĩa chính xác tuổi thọ phụ thuộc vào ứng dụng cụ thể của mạng. Đó là khoảng thời gian từ khi mạng bắt đầu sử dụng cho đến khi nút đầu tiên bị hỏng (hoặc hết năng lượng); hoặc nó bao gồm khoảng thời gian từ khi mạng bắt đầu sử dụng cho đến khi mất kết nối giữa hai hoặc nhiều nút và khoảng thời gian cho đến khi 50% nút bị hỏng; hoặc đó là khoảng thời gian khi lần đầu tiên một điểm nằm trong vùng quan sát không còn chịu sự giám sát của bất kỳ nút cảm biến nào (trong nhiều trường hợp, WSN sử dụng số nút cảm biến nhiều hơn yêu cầu nên một điểm có thể chịu sự giám sát của nhiều nút).

Khả năng mở rộng: Từ một WSN gồm một số lượng lớn các nút, kiến trúc và giao thức của mạng làm việc có khả năng mở rộng.

Dải rộng của mật độ: Trong một WSN, số lượng nút trên mỗi đơn vị diện tích - mật độ của mạng - có thể khác nhau đáng kể. Các ứng dụng khác nhau sẽ có mật độ nút rất khác nhau. Ngay cả trong một ứng dụng nhất định, mật độ có thể khác nhau theo thời gian và không gian do các nút di chuyển hoặc bị hỏng; mật độ cũng không phải đồng nhất trong toàn bộ mạng (do vị trí đặt nút không hoàn hảo) và mạng nên thích ứng với các biến đổi đó.

Khả trình: không chỉ cần để cho các nút để xử lý thông tin mà còn nó còn dùng để các nút phản ứng linh hoạt hơn khi thay đổi nhiệm vụ. Các nút cần được lập trình, và chương trình đó phải được thay đổi trong quá trình hoạt động, khi nhiệm vụ mới trở nên quan trọng. Xử lý lý thông tin theo một cách định sẵn không thể đáp ứng được yêu cầu.

Có khả năng bảo dưỡng: Khi cả môi trường của WSN và chính bản thân WSN thay đổi (hết pin, nút bị hỏng, khi có nhiệm vụ mới) thì hệ thống phải thích nghi với điều đó. Nó có thể theo dõi trạng thái của chính mình để thay đổi các thông số hoặc để lựa chọn tiêu chí hoạt động (ví dụ như chất lượng dịch vụ sẽ bị giảm sút khi nguồn năng lượng bị yếu đi). Trong trường hợp này, mạng phải tự bảo dưỡng, nó cũng có thể tương tác với các cơ chế bảo dưỡng bên ngoài để quá trình hoạt động đảm bảo chất lượng đã đặt ra.

1.3.2 Các kỹ thuật yêu cầu

Để thực hiện các yêu cầu này, kỹ thuật đổi mới mạng thông tin, kiến trúc và các khái niệm giao thức mới đã được thực hiện. Thách thức ở đây là cần phải tìm công nghệ đáp ứng được các yêu cầu về chất lượng dịch vụ, tuổi thọ và các yêu cầu về khả năng bảo dưỡng. Mặt khác, công nghệ này phải sử dụng được trong nhiều ứng dụng cụ thể và phải cho thấy được tính cần thiết của WSN trong các ứng dụng này. Sau đây là một số công nghệ thường được dùng trong các WSN:

Giao tiếp vô tuyến đa bước nhảy: Khi giao tiếp vô tuyến là kỹ thuật cốt lõi, thông tin trực tiếp giữa người gửi và người nhận sẽ phải đối mặt với một số hạn chế. Đặc biệt, giao tiếp đường dài chỉ thực hiện được khi sử dụng công suất phát rất cao. Việc sử dụng các nút trung gian như các trệ có thể làm giảm tổng năng lượng yêu cầu. Do đó, với nhiều loại WSN, cái gọi là giao tiếp đa bước nhảy là một thành phần cần thiết.

Sử dụng năng lượng có hiệu quả: để kéo dài tuổi thọ thì việc sử dụng năng lượng một cách có hiệu quả là kỹ thuật trọng điểm. Điều này có thể có được bằng cách sử dụng hiệu quả năng lượng truyền dữ liệu giữa hai nút (đo bằng bit /J) hoặc quan trọng hơn là xác định hiệu quả năng lượng của thông tin yêu cầu. Ngoài ra, tiêu thụ năng lượng không đồng đều – tạo nên các "điểm nóng - hotspot" cũng là một vấn đề.

Tự cấu hình: WSN sẽ phải tự cấu hình các thông số hoạt động của nó, việc này độc lập với cấu hình bên ngoài – Số nút càng ít càng tốt và bố trí đơn giản là yêu cầu trong hầu hết các ứng dụng. Ví dụ, các nút có thể xác định vị trí địa lý của nó chỉ bằng cách sử dụng các nút khác trong mạng. Nó còn được gọi là "tự định vị - self-location". Ngoài ra, mạng có thể có các nút hỏng (ví dụ như do hết pin) hoặc tích hợp các nút mới (ví dụ như để bổ sung thêm nút khi mạng chưa đáp ứng được yêu cầu đặt ra).

Sự cộng tác và xử lý trong mạng: Trong một số trường hợp, một cảm biến không thể quyết định được việc liệu sự kiện có xảy ra hay không nhưng nhiều cảm biến cộng tác với

nhau sẽ cung cấp đủ thông tin để làm điều đó. Thông tin được xử lý trong mạng theo các cách thức khác nhau để có được sự cộng tác này. Điều này ngược với việc mỗi nút truyền dữ liệu đến một mạng bên ngoài để xử lý.

Ví dụ, để xác định nhiệt độ cao nhất và nhiệt độ trung bình trong một khu vực và đưa giá trị đến bộ chứa. Để giải quyết bài toán này một cách có hiệu quả, các kết quả đọc từ các cảm biến riêng lẻ được tập hợp lại và truyền qua mạng. Việc này sẽ làm giảm số dữ liệu truyền trên mạng và do đó, việc sử dụng năng lượng sẽ hiệu quả hơn.

Trung tâm dữ liệu: Mạng thông tin truyền thống thường tập trung vào việc chuyển giao dữ liệu giữa hai thiết bị cụ thể, mỗi thiết bị yêu cầu có ít nhất một địa chỉ mạng. Mạng có kiến trúc như vậy hoạt động theo trung tâm địa chỉ. Trong WSN, nơi mà các nút thường được bố trí dư để dự phòng cho việc nút bị hỏng hoặc để tăng chất lượng của thiết bị chỉ có một nút cảm biến. Việc cung cấp dữ liệu giống nhau cho cùng một nút mạng là không cần thiết. Vấn đề quan trọng ở đây là bản thân dữ liệu nhận được chứ không phải là nút đã cung cấp dữ liệu đó. Do đó, xu hướng hiện nay là chuyển từ mô hình trung tâm địa chỉ sang mô hình trung tâm dữ liệu trong việc thiết kế kiến trúc và giao thức truyền thông.

Ví dụ, mô hình tương tác của trung tâm dữ liệu như vậy sẽ được sử dụng khi cần có nhiệt độ trung bình trong một khu vực đã cho, nó ngược với yêu cầu nhiệt độ đọc từ các nút riêng lẻ. Mô hình trung tâm dữ liệu như vậy cũng có thể được dùng để đặt điều kiện cho các cảnh báo hoặc các sự kiện ("bật báo động nếu nhiệt độ vượt quá ngưỡng").

1.4 Tại sao mạng cảm biến lại khác với các mạng khác?

Trên cơ sở của một số ứng dụng và những thách thức chính, mạng cảm biến không dây có thể trở thành mạng Ad hoc di động (MANET – Mobile Ad hoc Network) và fieldbus.

1..4.1 Mạng Ad hoc di động và WSN

Mạng Ad hoc là một mạng cài đặt, theo nghĩa đen, cho một mục đích cụ thể, để đáp ứng nhu cầu thông tin xuất hiện tức thời. Ví dụ đơn giản của một mạng Ad hoc là một tập hợp các máy tính kết nối với nhau qua cáp để tạo thành một mạng nhỏ, như một số máy tính xách tay trong một phòng họp. Trong ví dụ này, khía cạnh tự cấu hình đóng vai trò quan trọng - mạng dự kiến làm việc mà không cần quản lý hoặc cấu hình bằng tay.

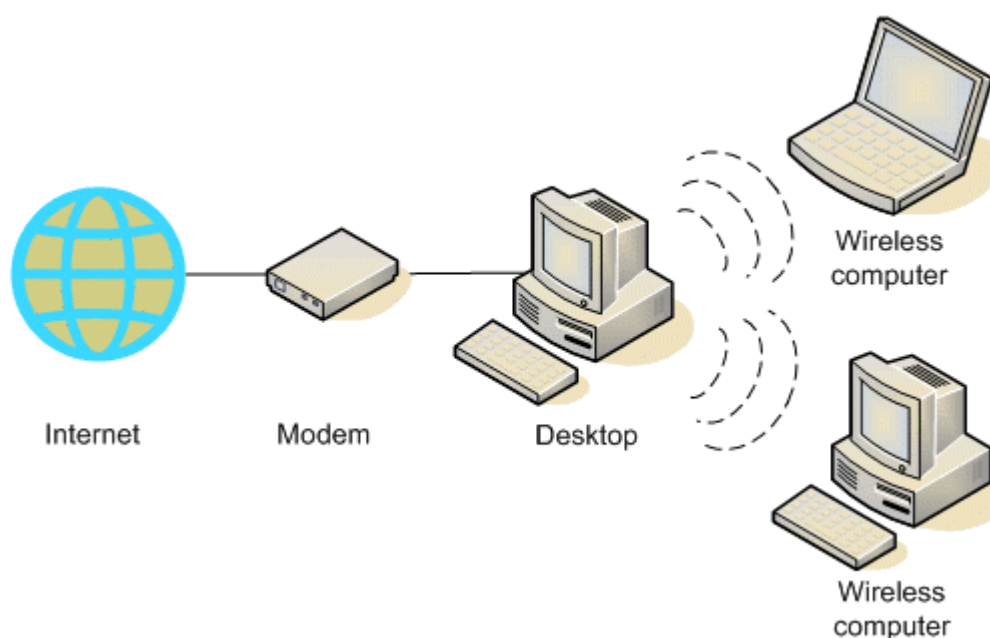
Mạng ad hoc

Ad hoc networks là điểm biên cuối cùng của thông tin không dây (thông tin vô tuyến). Công nghệ này cho phép các nút (điểm nối) mạng truyền trực tiếp với nhau sử dụng bộ thu phát không dây (wireless transceiver) mà không cần bất cứ một cơ sở hạ tầng cố định nào. Đây là một đặc tính riêng biệt của mạng ad hoc so với các mạng không dây truyền thống như các mạng chia ô (cellular networks) và mạng WLAN, trong đó các nút (ví dụ như các thuê bao điện thoại di động) giao tiếp với nhau thông qua các trạm vô tuyến cơ sở (wired radio antenna).

Ad hoc networks được mong đợi sẽ làm cách mạng hóa thông tin không dây trong một vài năm tới bằng việc bổ sung thêm vào các mô hình mạng truyền thống (như Internet, cellular networks, truyền thông vệ tinh). Mạng Ad hoc cũng có thể được xem như những bản

sao công nghệ của những khái niệm máy tính thường gặp. Bằng việc khám phá công nghệ mạng không dây Ad hoc, những thiết bị cầm tay đủ chủng loại (như điện thoại di động, PDAs, máy tính xách tay, máy nhắn tin “pager”.....) và các thiết bị cố định (như các trạm vô tuyến cơ sở, các điểm truy cập Internet không dây, ...) có thể được kết nối với nhau, tạo thành mạng toàn cầu, khắp mọi nơi.

Trong tương lai, công nghệ mạng Ad hoc có thể sẽ là lựa chọn rất hữu ích. Ví dụ, hãy xem tình huống sau. Một cơn động đất khủng khiếp đã tàn phá thành phố của chúng ta, trong đó có hầu hết các cơ sở hạ tầng viễn thông (như các đường điện thoại, trạm vô tuyến cơ sở...). Nhiều đội cứu hộ (như lính cứu hỏa, cảnh sát, bác sĩ, các tình nguyện viên ...) đang nỗ lực để cứu mọi người khỏi cơn động đất và chữa trị cho những người bị thương. Để hỗ trợ tốt hơn cho đội cứu hộ, các hoạt động cứu hộ của họ phải được hợp tác với nhau. Rõ ràng là 1 hoạt động hợp tác như thế chỉ đạt được thành quả khi đội cứu hộ có thể giao tiếp, thông tin với nhau, cả với đồng nghiệp của mình (ví dụ 1 cảnh sát với 1 cảnh sát khác) và cả với thành viên của đội cứu hộ khác (ví dụ 1 lính cứu hỏa yêu cầu sự trợ giúp từ 1 bác sĩ).



Hình 1.4 Sơ đồ nguyên lý mạng chia sẻ kết nối dạng ad hoc

Với những công nghệ hiện có, những nỗ lực của đội cứu hộ sẽ rất khó thành công khi những cơ sở hạ tầng viễn thông cố định bị tàn phá nặng nề. Thậm chí những thành viên của đội cứu hộ này được trang bị PDA hay các thiết bị tương tự khác trong trường hợp không thể truy cập được với các điểm cố định, chỉ những kết nối giữa các thành viên của đội cứu hộ đứng gần nhau mới thực hiện được. Vì vậy, một trong những ưu tiên trong việc quản lý và không chế thảm họa ngày nay là cài đặt lại các cơ sở hạ tầng viễn thông nhanh nhất có thể, bằng cách sửa chữa các thiết bị, kết cấu hư hỏng hay triển khai các thiết bị viễn thông tạm thời.

Khó khăn này có thể được giải quyết đáng kể nếu chúng ta áp dụng những công nghệ dựa vào mạng Ad hoc: sử dụng các giao tiếp không dây phân tán giữa nhiều điểm truy cập khác nhau, thậm chí các đội cứu hộ ở cách xa nhau cũng có thể liên lạc với nhau hay liên lạc với các thành viên đội cứu hộ khác ở khoảng giữa như hoạt động của một trạm chuyển tiếp.

Vì khu vực xảy ra thảm họa sẽ tập trung nhiều đội cứu hộ, nên các liên lạc trong phạm vi thành phố (hay thậm chí là phạm vi cả nước) có thể thực hiện được, cho phép các nỗ lực cứu hộ được hợp tác thành công mà không cần thiết lập lại các cơ sở viễn thông cố định.

Thông thường, khái niệm về MANET liên quan đến truyền thông vô tuyến và đặc biệt là truyền thông đa bước nhảy vô tuyến. Sự di chuyển của các nút là một thành phần của mạng. Các mạng này được sử dụng cho hoạt động cứu trợ thiên tai - nhân viên cứu hỏa giao tiếp với nhau - hoặc dùng trong các địa điểm khó khăn như công trình xây dựng lớn, nơi có thể việc triển khai cơ sở hạ tầng không dây (ví dụ như điểm truy cập) trong khi thực hiện hữu tuyến không phải là một lựa chọn khả thi. Trong mạng đó, các nút riêng lẻ kết hợp với nhau tạo thành một mạng để chuyển tiếp các gói tin giữa các nút, do đó nó mở rộng phạm vi hoạt động của một nút đơn, mở rộng khu vực địa lý giữa người gửi và người nhận. Hai thách thức cơ bản của MANET là tổ chức lại mạng khi các nút di chuyển và xử lý các vấn đề về hạn chế của truyền thông vô tuyến.

Đây là những tồn tại chung của cả mạng MANET và WSN. Tuy nhiên vẫn tồn tại một số khác biệt cơ bản giữa hai mạng này để đảm bảo sự phân biệt giữa chúng.

Thiết bị và các ứng dụng: Ứng dụng và thiết bị của MANET khác WSN. Trong MANET, đầu cuối có thể khá mạnh (một máy tính xách tay hoặc một PDA (Personal Digital Assistant - Thiết bị hỗ trợ cá nhân kỹ thuật số)) có nguồn pin tương đối lớn. Cần có thiết bị này vì trong các ứng dụng MANET điển hình thường có sự xuất hiện của con người: MANET được sử dụng cho truyền thông thoại giữa hai đồng nghiệp ở xa, hoặc được sử dụng để truy cập vào một cơ sở hạ tầng từ xa, giống như một máy chủ Web. Vì vậy, thiết bị phải được đủ mạnh để hỗ trợ các ứng dụng này.

Nhờ sử dụng một lượng lớn các nút cảm biến, sử dụng công nghệ tính toán và truyền thông mà WSN có thể thực hiện được rất nhiều ứng dụng khác nhau. Điều đó không có nghĩa là chỉ sử dụng một giải pháp mà giải quyết được mọi yêu cầu đặt ra. Ví dụ, WSN có thể hiểu được rất nhiều mật độ mạng khác nhau, từ thưa thớt dày đặc, vì vậy mà nó sẽ yêu cầu nhiều giao thức khác nhau hoặc ít nhất phải là các giao thức phù hợp. Sự đa dạng về mật độ không phải là điều phổ biến trong các MANET.

Tương tác với môi trường: Do các WSN phải có tương tác với môi trường nên đặc tính lưu lượng của nó có thể rất khác nhau. Thông thường, các WSN có tốc độ dữ liệu rất thấp một thời gian dài nhưng nó lại có lưu lượng rất cao khi có hiện tượng xảy ra (ví dụ, đối với một hệ thống thời gian thực là mưa hoặc bão). Trạng thái nghỉ trong một thời gian dài có thể thay thế bằng trạng thái hoạt động rất cao của mạng trong một thời gian ngắn, nâng công suất của mạng lên giới hạn. Trong khi đó, MANET được sử dụng để hỗ trợ nhiều ứng dụng thông thường hơn (Web, giọng nói...) với các đặc tính lưu lượng không có sự thay đổi đột biến.

Quy mô: Các WSN thường dùng để giám sát số lượng thực thể lớn hơn nhiều (hàng nghìn, thậm chí hàng trăm nghìn) so với mạng ad hoc.

Năng lượng: Trong cả hai mạng WSN và MANET, năng lượng là vấn đề được quan tâm. Các WSN có yêu cầu chặt chẽ hơn về thời gian sử dụng, khả năng sạc pin hoặc thay thế nguồn pin của nút so với MANET. Do đó, WSN quan tâm nhiều đến việc tiêu thụ năng lượng

hơn MANET.

Tự cấu hình: Tương tự như mạng ad hoc, các WSN yêu cầu tự cấu hình thành mạng kết nối, nhưng sự khác biệt trong lưu lượng, sử dụng năng lượng... có thể đòi hỏi giải pháp khác nhau. Tuy nhiên, về mặt này thì MANET và WSN tương tự nhau.

Độ tin cậy và chất lượng dịch vụ: Các yêu cầu liên quan đến độ tin cậy và chất lượng dịch vụ là khá khác nhau. Trong một MANET, mỗi nút riêng rẽ đều phải có độ tin cậy khá cao, còn trong một WSN, một nút riêng rẽ không đóng vai trò quan trọng. Chất lượng dịch vụ của Manet được quyết định bởi các ứng dụng truyền thống; đối với WSN, khái niệm chất lượng dịch vụ là yêu cầu hoàn toàn mới, và đương nhiên, mạng sẽ phải sử dụng năng lượng để thực hiện yêu cầu này.

Trung tâm dữ liệu: Việc triển khai dự phòng các nút sẽ làm tăng kích thước của các giao thức trung tâm dữ liệu trong WSN. Các giao thức trung tâm dữ liệu không liên quan đến MANET.

Dịch chuyển: Vấn đề dịch chuyển trong MANET bao gồm sự dịch chuyển các nút di chuyển, thay đổi tuyến đường đa bước nhảy trong mạng và điều này phải được xử lý. Trong một WSN, vấn đề này cũng có thể tồn tại nếu các nút cảm biến di chuyển. Trong WSN, có hai vấn đề về dịch chuyển cần được quan tâm.

Thứ nhất, mạng cảm biến có thể được dùng để phát hiện và quan sát một hiện tượng vật lý (ví dụ như trong các chương trình phát hiện xâm nhập). Hiện tượng này là nguyên nhân của sự kiện xảy ra trong mạng (bất báo động) và cũng có thể tạo ra một số xử lý cục bộ (ví dụ, xác định xem có thực sự đó có phải là một kẻ xâm nhập?). Điều gì xảy ra nếu đối tượng di chuyển? Về mặt lý tưởng, dữ liệu tại một địa điểm được tập hợp lại và có thể sử dụng cho địa điểm gần đó. Ngoài ra, trong các ứng dụng theo dõi, đây là nhiệm vụ đương nhiên của mạng để đảm bảo một số hình thức hoạt động xảy ra trong các nút xung quanh hiện tượng quan sát.

Thứ hai, các bộ chứa thông tin trong mạng (các nút mà thông tin phải được gửi đến) cũng có thể di chuyển. Về nguyên tắc, điều này là không khác với việc di chuyển của các nút trong MANET thông thường, nhưng có thể làm giảm hiệu quả hoạt động của các giao thức so với trạng thái tĩnh.

Hơn nữa, trong MANET và WSN, tính di động có thể tương quan với nhau, tức là có một nhóm các nút cùng di chuyển. Ví dụ như trong MANET là một nhóm người đi du lịch cùng nhau. Trong WSN, sự chuyển động của các nút có thể bão cuốn đi.

Tóm lại, tuy có những điểm chung nhưng trên thực tế, WSN hỗ trợ cho các ứng dụng rất khác nhau, chúng tương tác với môi trường vật lý và phải xem xét thật cẩn thận mức ưu tiên của các thông số. Điều này thực sự làm nên sự khác biệt giữa WSN và MANET.

1.4.2 Fieldbus and WSN

Fieldbus là mạng được thiết kế đặc biệt cho các hoạt động theo thời gian thực và thường với dung sai lỗi định sẵn. Nó được sử dụng chủ yếu trong các ứng dụng điều khiển được xem như là một phần của vòng điều khiển. Ví dụ như Profibus và mạng Token Bus IEEE 802.4 cho tự động hóa nhà máy hoặc bus CAN cho các mạng onboard trong xe hơi... Do các yêu

cầu về thời gian thực rất nghiêm ngặt nên các mạng này thường là mạng hữu tuyến và chỉ có các lớp một (lớp vật lý), hai (lớp liên kết), và bảy (lớp ứng dụng) của mô hình chuẩn OSI được sử dụng. Tránh giao tiếp qua nhiều bước và trễ khi đi qua các nút trung gian. Mặc dù vậy, một số nghiên cứu vẫn liệt kê fieldbus vào loại truyền thông vô tuyến.

Do fieldbus cũng quan tâm đến môi trường vật lý mà nó thu thập dữ liệu cảm biến để từ đó đưa ra các quyết định điều khiển nên với nghĩa này, nó rất giống với các WSN. Theo một cách hiểu nào đó thì WSN có thể được xem như fieldbus vô tuyến. Tuy nhiên vẫn còn một số khác biệt: WSN không chú ý đến vấn đề thời gian thực mà nó tập trung nhiều hơn vào các ứng dụng có thể chịu được trễ. Ngoài ra, việc lựa chọn các đặc tính một cách phù hợp là điều mà WSN muốn thực hiện (ví dụ như tính chính xác đối với hiệu quả năng lượng), và điều này lại không thường hiện diện trong fieldbus. Fieldbuses không nỗ lực để bảo tồn năng lượng, và các giao thức của nó không được chuẩn bị để làm điều đó.

1.5 Các kỹ thuật cho WSN

Xây dựng các mạng cảm biến không dây chỉ trở thành hiện thực với một số tiến bộ cơ bản trong công nghệ cho phép. Trước hết, đó là công nghệ tối thiểu hoá phần cứng. Kích thước của chip càng nhỏ thì mức tiêu thụ điện năng càng ít, thậm chí nằm trong khả năng tính toán của WSN. Điều này liên quan đặc biệt đến vi điều khiển và chip bộ nhớ, nhưng các modem vô tuyến, chịu trách nhiệm về phần giao tiếp vô tuyến, sẽ làm cho việc sử dụng năng lượng có hiệu quả hơn. Giảm kích thước của chip và hiệu quả năng lượng cải thiện đi kèm với chi phí giảm, điều này là cần thiết để thực hiện triển khai dự phòng các nút một cách hợp lý.

Sau việc thu thập và xử lý thông tin thì thiết bị cảm nhận thực tế là công nghệ quan tâm thứ ba. Rất khó để bao quát vấn đề này vì có rất nhiều loại cảm biến khác nhau. Chương 2 sẽ trình bày chi tiết hơn vấn đề này.

Ba phần cơ bản này của nút cảm biến phải đi cùng với nguồn cấp. Các yêu cầu này phụ thuộc vào ứng dụng, pin có dung lượng cao thì tuổi thọ dài. Một cách lý tưởng, nút cảm biến cũng có thiết bị để tìm năng lượng, nạp pin bằng năng lượng lấy từ môi trường - các tế bào năng lượng mặt trời là một lựa chọn.

Tương ứng với các công nghệ phần cứng cơ bản là phần mềm. Câu hỏi đầu tiên để trả lời ở đây là sự phân chia nhiệm vụ và chức năng trong một nút đơn - kiến trúc của hệ điều hành hay môi trường hoạt động. Môi trường này phải hỗ trợ việc thực hiện lại nhiệm vụ đơn giản, trao đổi thông tin liên lớp và có module cho phép thực hiện các bảo dưỡng đơn giản. Kiến trúc phần mềm trên nút đơn phải được mở rộng đến kiến trúc của mạng, nơi thực hiện việc phân chia nhiệm vụ giữa các nút.

CÂU HỎI ÔN TẬP CHƯƠNG 1

1. Nêu bảy đặc điểm cơ bản của WSN
2. Các yêu cầu kỹ thuật của WSN
3. Các kỹ thuật yêu cầu của WSN
4. So sánh mạng Ad hoc và WSN
4. So sánh Fieldbus và WSN
5. Nêu một số ứng dụng của WSN trong cuộc sống hàng ngày
7. Nêu đặc trưng và cấu hình WSN

CHƯƠNG 2. KIẾN TRÚC NÚT ĐƠN

Mục đích của chương

Chương này giải thích phần tử cơ bản của WSN là **nút**, các nhiệm vụ cơ bản của nó như tính toán, lưu trữ, giao tiếp, cảm nhận/vận hành và các bộ phận được yêu cầu để thực hiện các nhiệm vụ này. Khi đó, tiêu thụ năng lượng của các bộ phận này cũng được đề cập đến: năng lượng có thể lưu trữ, thu thập từ môi trường và lưu giữ nó bằng cách điều khiển chế độ hoạt động của các bộ phận của nút một cách thông minh. Việc điều khiển này do một hệ điều hành giống như môi trường thực hiện quyết định.

Cuối chương, bạn đọc sẽ có những hiểu biết nhất định về khả năng và những hạn chế của các nút trong mạng cảm biến. Nó là nền tảng của các chương sau, trong đó tập trung vào các nguyên tắc lựa chọn nút cảm biến riêng rẽ có thể kết nối thành một WSN.

Để xây dựng một WSN, yêu cầu đầu tiên là các nút hợp thành phải được triển khai và có thể sử dụng. Các nút này phải có các yêu cầu từ các yêu cầu của ứng dụng đã cho: nhỏ, rẻ hoặc hiệu quả năng lượng, được trang bị các cảm biến tốt, có khả năng tính toán và tài nguyên bộ nhớ cần thiết, phải thích hợp với các phương tiện truyền thông. Các bộ phận phần cứng này và cấu tạo của chúng trong nút chức năng được đề cập trong phần 2.1; công suất tiêu thụ của các bộ phận này và các thỏa hiệp được nói đến trong phần 2.2. Trong khi chương này chỉ tập trung và nút riêng lẻ thì việc lựa chọn kỹ thuật giao tiếp đặc biệt cho kiến trúc của mạng cảm biến không dây lại được mô tả trong chương 3.

Ngoài phần cứng của các nút cảm biến, hệ điều hành và mô hình lập trình cũng là các mối quan tâm. Phần 2.3 mô tả nhiệm vụ của hệ điều hành, một số ví dụ và các giao diện lập trình phù hợp.

2.1 Các bộ phận phần cứng

2.1.1 Tổng quan phần cứng của nút cảm biến

Khi chọn các bộ phận phần cứng cho nút cảm biến không dây, rõ ràng là các yêu cầu của ứng dụng đóng vai trò quyết định với mối quan tâm chủ yếu về kích thước, giá thành và tiêu thụ năng lượng của các nút – giao tiếp và các phương tiện tính toán thường được quan tâm để có chất lượng chấp nhận được, nhưng thỏa hiệp giữa các đặc điểm và chi phí phải là cốt yếu. Trong một số trường hợp đặc biệt, toàn bộ nút cảm biến phải nhỏ hơn 1 cc, nhẹ hơn 100 g, rẻ hơn 1 USD và sử dụng ít hơn 100 μ W, thậm chí có những trường hợp kích thước thu gọn chỉ bằng hạt bụi. Trong nhiều ứng dụng thực tế, kích thước của nút nhỏ không quan trọng mà sự thuận tiện, nguồn cấp đơn giản và chi phí đóng vai trò quan trọng hơn.

Tuy đa dạng như vậy nhưng xu hướng chung vẫn là tập trung vào nền tảng phần cứng của các nút cảm biến không dây. Trong khi đương nhiên là không tồn tại sẵn một chuẩn để sử dụng cũng như không tồn tại chuẩn có thể hỗ trợ cho tất cả các loại ứng dụng nên phần này sẽ đưa ra cái nhìn tổng quát về các kiến trúc nút cảm biến điển hình. Ngoài ra còn có một số nghiên cứu tập trung vào việc rút ngắn các bộ phận về mặt kích thước, tiêu thụ năng lượng hay giá thành.

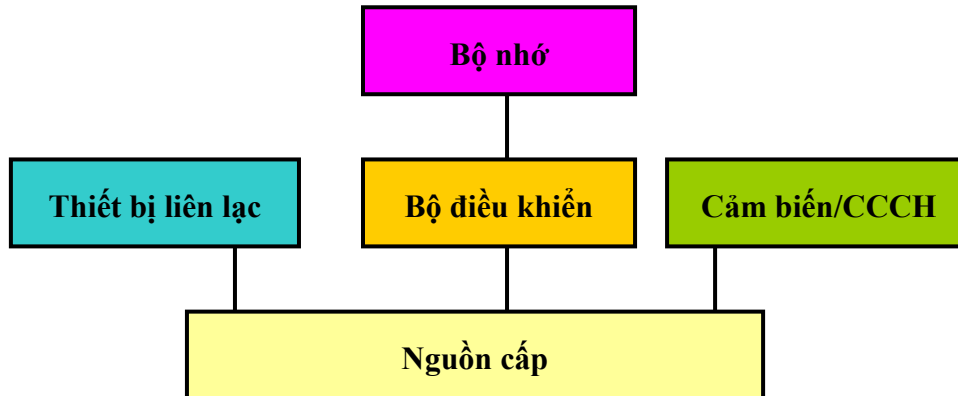
Một nút cảm biến cơ bản gồm 5 bộ phận chính (hình 2.1):

Bộ điều khiển: Bộ điều khiển xử lý tất cả dữ liệu có liên quan theo mã nhị phân.

Bộ nhớ: Thông thường, bộ nhớ được dùng để lưu giữ chương trình và dữ liệu.

Cảm biến và cơ cấu chấp hành: Đây là giao diện với môi trường vật lý. Các thiết bị này có thể quan sát hoặc giám sát các thông số vật lý của môi trường.

Thiết bị liên lạc: Các nút trong mạng đòi hỏi một thiết bị để gửi và nhận thông tin qua kênh vô tuyến.



Hình 2.1 Cấu trúc phần cứng của nút cảm biến

Nguồn cấp: Thông thường, không có sẵn nguồn cấp điện nên cần phải có pin để cung cấp năng lượng, thậm chí cả pin có thể sạc từ môi trường (pin mặt trời).

Mỗi bộ phận này phải hoạt động sao cho đảm bảo lượng tiêu thụ năng lượng là ít nhất có thể, đồng thời phải hoàn thành tất cả các nhiệm vụ đặt ra. Ví dụ, cả hai thiết bị liên lạc và bộ điều khiển nên được tắt càng lâu càng tốt. Muốn thức dậy (wake up), bộ điều khiển có thể sử dụng một bộ đếm thời gian được lập trình trước để kích hoạt lại. Ngoài ra, các cảm biến có thể được lập trình để tạo ngắt khi một sự kiện được xảy ra – ví dụ như khi giá trị nhiệt độ vượt quá một ngưỡng nhất định hoặc thiết bị giao tiếp phát hiện có thông tin đến.

Để hỗ trợ chức năng cảnh báo như vậy, yêu cầu phải có kết nối phù hợp giữa các bộ phận riêng rẽ. Hơn nữa, cả thông tin dữ liệu và điều khiển đều phải được trao đổi qua các kết nối này. Kết nối này có thể rất đơn giản, ví dụ, một cảm biến có thể gửi giá trị tương tự đến bộ điều khiển; hoặc phức tạp hơn là tiền xử lý dữ liệu cảm biến và chỉ đánh thức bộ điều khiển chính khi một sự kiện được phát hiện - ví dụ, phát hiện ngưỡng qua cho một cảm biến nhiệt độ đơn giản. Tiền xử lý của nút cảm biến phải đảm bảo tính đơn giản và hiệu quả năng lượng được cải thiện.

2.1.2 Bộ điều khiển

Bộ vi điều khiển/ vi xử lý, FPGA và ASIC

Bộ điều khiển là cốt lõi của một nút cảm biến không dây. Nó thu thập dữ liệu từ các cảm biến, xử lý dữ liệu này, quyết định thời gian và địa điểm gửi dữ liệu đến, nhận dữ liệu từ

các nút cảm biến khác và quyết định hành vi của các cơ cấu chấp hành. Ngoài ra, nó còn thực hiện các chương trình khác, sắp xếp các giao thức giao tiếp và xử lý tín hiệu theo thời gian tới hạn và các chương trình ứng dụng. Nó là **khối xử lý trung tâm** (CPU) của nút.

Sự đa dạng trong nhiệm vụ xử lý tạo nên các kiến trúc khác nhau của bộ điều khiển dựa trên sự thỏa hiệp giữa độ linh hoạt, đặc tính, hiệu quả năng lượng và giá thành.

Một giải pháp là sử dụng bộ vi xử lý cho mục đích chung, như máy tính để bàn. Đây là những bộ vi xử lý rất mạnh nhưng tiêu thụ năng lượng quá lớn. Các bộ xử lý đơn giản hơn có thể tạo ra bằng cách sử dụng các hệ thống nhúng. Các bộ xử lý này thường được gọi là các **bộ vi điều khiển**. Đặc điểm chính khiến các vi điều khiển đặc biệt phù hợp với các hệ thống nhúng là sự linh hoạt trong việc kết nối với các thiết bị khác (như bộ cảm biến), tập lệnh được thiết lập để xử lý tín hiệu theo thời gian tới hạn, và năng lượng tiêu thụ thấp. Một điểm thuận tiện nữa là nó thường có bộ nhớ được xây dựng bên trong. Ngoài ra, chúng có thể lập trình được nên rất linh hoạt. Các vi điều khiển cũng thích hợp cho WSN vì nó có thể giảm năng lượng tiêu thụ bằng các **trạng thái ngủ**, tức là chỉ có một số phần của bộ điều khiển hoạt động. Một trong những khác biệt chính với các hệ thống đa dụng là hệ vi điều khiển thường không có đơn vị quản lý bộ nhớ nên nó phần nào hạn chế chức năng của bộ nhớ.

Trường hợp riêng của bộ vi xử lý có thể lập trình được là bộ **xử lý tín hiệu số** (DSP - Digital Signal Processors). Đây là một thiết bị chuyên dụng, đặc trưng bởi cấu trúc và tập lệnh, dùng để xử lý một lượng lớn dữ liệu trong các ứng dụng xử lý tín hiệu. Trong nút cảm biến không dây, DSP như vậy có thể được sử dụng để xử lý dữ liệu từ một thiết bị liên lạc vô tuyến tương tự đơn giản để tạo ra dữ liệu số. Trong giao tiếp không dây băng thông rộng, DSP thực sự phù hợp và có hiệu quả. Nhưng trong các mạng cảm biến không dây, các yêu cầu về truyền thông không dây thường khiêm tốn hơn nhiều (ví dụ như đơn giản, dễ điều chế, sử dụng hiệu quả trong phần cứng của thiết bị liên lạc) và các nhiệm vụ xử lý tín hiệu liên quan đến việc cảm nhận thực tế dữ liệu cũng không quá phức tạp. Do đó, những lợi thế của DSP thường không cần thiết đối với nút WSN và nó thường không được sử dụng.

Một lựa chọn khác cho bộ điều khiển xuất phát từ yêu cầu cao về tính linh hoạt của bộ vi điều khiển là sử dụng **mảng cổng logic lập trình được** (FPGA – Field Programmable Gate Arrays) hoặc **mạch tích hợp chuyên dụng** (ASIC - Application- Specific Integrated Circuits). FPGA có thể tái lập trình (hay tái cấu hình) theo yêu cầu. Tuy nhiên, điều này có thể mất thời gian và năng lượng. ASIC là một bộ xử lý chuyên dụng và người sử dụng có thể thiết kế để phù hợp với ứng dụng đã cho, ví dụ như các chuyển mạch và định tuyến tốc độ cao. Cân nhắc trong quá trình thiết kế ở đây là phải đảm bảo thỏa hiệp giữa độ linh hoạt và hiệu quả năng lượng cũng như các đặc tính của thiết bị. Mặt khác, khi bộ vi điều khiển đòi hỏi phải phát triển phần mềm, ASIC sẽ cung cấp chức năng tương tự trong phần cứng, điều này sẽ làm tăng giá thành của phần cứng.

Với một ứng dụng WSN chuyên dụng, nơi mà nhiệm vụ các nút cảm biến không thay đổi trong thời gian tồn tại và số lượng của chúng đủ lớn để đảm bảo sự phát triển trong ASIC thì chúng có thể có giải pháp mạnh hơn. Tuy nhiên, với công nghệ WSN hiện nay, các bộ vi điều khiển linh hoạt hơn và sử dụng đơn giản hơn đã tạo ra giải pháp được ưa thích hơn.

Trong phần còn lại của cuốn sách này, giả sử WSN có kiến trúc dựa trên bộ vi điều

khẩn.

Một số ví dụ về các bộ vi điều khiển

Các bộ vi điều khiển sử dụng trong một số mẫu nút cảm biến không dây là bộ xử lý Atmel hoặc MSP 430 của Texas Instrument. Trong các mẫu cũ hơn, bộ xử lý Intel StrongARM cũng được dùng nhưng từ lâu nó không còn là lựa chọn trên thực tế. Tuy nhiên, các tính chất lý thuyết của các bộ xử lý và điều khiển này là tương tự nhau, các kết luận từ các nghiên cứu gần nhất vẫn có giá trị.

Intel StrongARM

Intel StrongARM trong WSN là một bộ xử lý high-end khá tốt, nó như là một thiết bị cầm tay, giống các PDA. Model SA-1100 có lõi là **máy tính có bộ lệnh rút gọn** 32 bit (RISC - Reduced Instruction Set Computer) và tốc độ lên đến 206 MHz.

Texas Instruments MSP 430

Texas Instrument cung cấp một họ các bộ vi điều khiển dưới tên gọi MSP 430. Không giống như StrongARM, nó rất rõ ràng là dùng cho các ứng dụng nhúng. Do đó, nó chạy trên lõi RISC 16 bit ở các tần số clock thấp hơn nhiều (đến 4 MHz) nhưng kết hợp với một loạt các khả năng kết nối và một tập lệnh đi kèm để dễ xử lý với các kiểu ngoại vi khác nhau. Nó mô tả những nét nổi bật của một lượng các RAM trên chip khác nhau (kích thước 2-10 kB), một số bộ chuyển đổi tương tự/số 12 bit và đồng hồ thời gian thực. Hiển nhiên là nó đủ mạnh để xử lý các nhiệm vụ tính toán của các nút cảm biến không dây thông thường.

Atmel ATmega

Atmel ATmega 128L là một bộ vi điều khiển 8 bit, cũng thường dùng cho các ứng dụng nhúng, được trang bị các giao diện bên ngoài để dùng cho các thiết bị ngoại thông thường.

2.1.3 Bộ nhớ

Bộ nhớ thường khá đơn giản. Rõ ràng là có nhu cầu cho **bộ nhớ truy cập ngẫu nhiên** (RAM - Random Access Memory) để chứa dữ liệu đọc từ cảm biến trung gian, các gói tin từ các nút khác... Trong khi RAM có ưu điểm là tốc độ nhanh thì nhược điểm chính của nó là bị mất dữ liệu nếu nguồn nuôi bị ngắt. Mã chương trình có thể được lưu trong **bộ nhớ chỉ đọc** (ROM - Read-Only Memory) hay loại thông dụng hơn là **bộ nhớ chỉ đọc lập trình được xóa bằng điện** (EEPROM - Electrically Erasable Programmable Read-Only Memory) hoặc **bộ nhớ FLASH** (loại mới nhất tương tự như EEPROM nhưng cho phép dữ liệu được xóa hoặc viết trong các khối thay vì chỉ theo byte tại một thời điểm). Bộ nhớ FLASH cũng có thể được dùng để lưu dữ liệu trung gian trong trường hợp RAM không đủ hoặc khi nguồn cấp của RAM tạm ngừng một thời gian. Trễ truy cập đọc và viết của bộ nhớ FLASH lớn sẽ phải đưa vào báo cáo và nó cũng yêu cầu năng lượng cao.

Kích thước bộ nhớ, đặc biệt là RAM, có thể quyết định giá thành sản xuất và mức tiêu thụ năng lượng. Tuy nhiên, các yêu cầu của bộ nhớ phụ thuộc rất nhiều vào ứng dụng có liên quan.

2.1.4 Thiết bị liên lạc

Lựa chọn môi trường truyền dẫn

Thiết bị liên lạc được sử dụng để trao đổi dữ liệu giữa các nút riêng rẽ. Trong một số trường hợp, liên lạc hữu tuyến là một lựa chọn và thường áp dụng trong việc cài đặt mạng cảm biến (sử dụng các bus trường như Profibus, LON, CAN ...). Các thiết bị liên lạc cho các mạng này thường là các bộ phận làm sẵn theo ý khách hàng.

Trường hợp giao tiếp vô tuyến được chú ý nhiều hơn. Lựa chọn đầu tiên là môi trường truyền dẫn – các lựa chọn thường bao gồm tần số vô tuyến, thông tin quang và sóng siêu âm; môi trường khác như độ tự cảm từ chỉ sử dụng trong các trường hợp rất đặc biệt. Các lựa chọn khác như giao tiếp dựa vào tần số vô tuyến (RF - Radio Frequency) là thích hợp nhất, nó cũng phù hợp với hầu hết yêu cầu của các ứng dụng WSN: nó tạo ra các tốc độ dữ liệu cao và có dải tương đối rộng, tỉ lệ lỗi chấp nhận được ở mức tiêu thụ năng lượng hợp lý và không yêu cầu đường nhìn thẳng giữa người gửi và người nhận.

Với hệ thống dựa vào RF vô tuyến thực tế, tần số sóng mạng phải được lựa chọn một cách cẩn thận. Chương 4 sẽ nói rõ hơn về vấn đề này. Trong phần này chỉ đề cập đến các mạng cảm biến không dây điển hình sử dụng các tần số liên lạc nằm trong khoảng từ 433 MHz đến 2.4 GHz.

Các bộ thu phát

Hiện nay, liên lạc yêu cầu cả bộ phát và bộ thu trong một nút cảm biến. Nhiệm vụ cốt yếu là biến đổi luồng bit đến từ bộ vi điều khiển (hoặc dãy byte, hoặc khung) thành/từ các sóng vô tuyến. Đối với các mục đích thực tế, rất thuận tiện để sử dụng thiết bị kết hợp cả hai nhiệm vụ này trong một thực thể duy nhất. Thiết bị kết hợp này gọi là **bộ thu phát**. Thông thường, hoạt động bán song công này được thực hiện khi phát và thu được thực hiện ở cùng thời điểm trong môi trường vô tuyến, điều này là không thực tế trong hầu hết các trường hợp (bộ thu chỉ thường lắng nghe bộ phát của nó). Loại bộ thu phát có giá thành thấp hữu dụng về phương diện thương mại sẽ được kết hợp trong tất cả các mạch được yêu cầu cho việc phát và thu – điều chế, giải điều chế, các bộ khuếch đại, bộ lọc, bộ trộn... Để lựa chọn đúng, phải hiểu được nhiệm vụ và các đặc tính chính của bộ thu phát.

• *Nhiệm vụ và các đặc tính của bộ thu phát*

Để lựa chọn các bộ thu phát phù hợp, số lượng các đặc tính của nó phải được liệt kê. Các đặc tính quan trọng nhất là:

• *Phục vụ lớp cao hơn*

Bộ thu phải hỏi các dịch vụ đến các lớp cao hơn, thường là lớp **điều khiển truy cập môi trường** (MAC - Medium Access Control). Dịch vụ này có thể coi là sự **định hướng gói tin**, đôi khi bộ thu phát chỉ cung cấp **giao diện byte** hoặc thậm chí chỉ là **giao diện bit** đến bộ vi điều khiển.

Trong một số trường hợp, bộ thu phát phải cung cấp giao diện để cho phép lớp MAC bắt đầu truyền dẫn khung và truyền gói tin đến bộ nhớ chính của nút cảm biến trong bộ thu phát (hoặc chùm byte hoặc bit mà không có yêu cầu xử lý thêm trong bộ vi điều khiển). Theo cách khác, các gói tin đến phải thành luồng trong các bộ đệm mà có thể truy cập bằng giao

thức MAC.

- **Công suất tiêu thụ và hiệu quả năng lượng**

Giải thích đơn giản nhất của hiệu quả năng lượng là năng lượng yêu cầu để truyền và nhận 1 bit. Do vậy, để phù hợp cho việc sử dụng trong các WSN, bộ thu phát phải có thể chuyển giữa các trạng thái khác nhau, ví dụ như trạng thái hoạt động và ngủ. Công suất tiêu thụ nhân rồi ở mỗi trạng thái này và trong quá trình chuyển trạng thái giữa chúng là rất quan trọng. Điều này sẽ được giới thiệu cụ thể trong phần 2.2.

- **Tần số sóng mang và đa kênh**

Các bộ thu phát có các tần số sóng mang khác nhau, đương nhiên nó phải phù hợp với các yêu cầu ứng dụng và các giới hạn điều chỉnh. Thông thường, bộ thu phát cung cấp một số tần số sóng mang (các kênh) để lựa chọn nhằm giảm vấn đề tắc nghẽn trong mạng mật độ dày đặc. Các kênh (hay các băng con) có liên quan đến nhau, ví dụ như trong các giao thức MAC (FDMA hay kỹ thuật đa kênh CSMA/ALOHA ở chương 5).

- **Thời gian thay đổi trạng thái và năng lượng**

Bộ thu phát có thể hoạt động ở các chế độ khác nhau: gửi hoặc nhận, sử dụng các kênh khác nhau hoặc trong các trạng thái an toàn nguồn khác nhau. Trong một số trường hợp, thời gian và năng lượng yêu cầu để chuyển giữa hai trạng thái này là các hệ số phẩm chất quan trọng. Ví dụ như thời gian giữa gửi và nhận quan trọng đối với các giao thức truy cập môi trường khác nhau (chương 5).

- **Tốc độ dữ liệu**

Tần số sóng mang và băng thông sử dụng kết hợp với việc điều chế và mã hoá xác định tốc độ dữ liệu. Thông thường, các giá trị khoảng vài chục kB/s, nhỏ hơn rất nhiều trong liên lạc vô tuyến, nhưng lại đủ cho các WSN. Có thể có được các tốc độ dữ liệu khác nhau, ví dụ như bằng cách sử dụng các sơ đồ điều chế khác nhau hay thay đổi tốc độ ký hiệu.

- **Điều chế**

Các bộ thu phát thường hỗ trợ khoá đóng – mở (on/off-keying), ASK, FSK hoặc các sơ đồ điều chế tương tự. Nếu có thể dùng được một số sơ đồ điều chế thì sẽ rất thuận tiện cho các cuộc thử nghiệm.

- **Mã hoá**

Một số bộ thu phát cho phép lựa chọn các sơ đồ mã hoá khác nhau.

- **Điều khiển công suất truyền dẫn**

Một số bộ thu phát có thể đưa ra điều khiển một cách trực tiếp qua công suất truyền dẫn mà nó sử dụng. Một số khác lại yêu cầu mạch phụ cho mục đích này. Thông thường, chỉ có một số rời rạc các mức điện năng là hữu dụng để từ đó có thể lựa chọn công suất truyền dẫn. Công suất đầu ra cực đại thường được xác định bằng cách điều chỉnh.

- **Hệ số tạp âm**

Hệ số tạp âm (NF - **noise figure**) được định nghĩa như là tỉ số giữa **tỉ số tín hiệu/tạp**

âm (SNR - Signal-to-Noise Ratio) đầu vào SNR_I và tỉ số tín hiệu/tạp âm đầu ra SNR_O của phân tử:

$$NF = \frac{SNR_I}{SNR_O} \quad (0.1)$$

Nó mô tả sự suy biến của SNR do hoạt động của phân tử và thường được tính theo dB:

$$NF(dB) = SNR_I(dB) - SNR_O(dB) \quad (0.2)$$

• **Hệ số khuếch đại**

Hệ số khuếch đại là tỉ số giữa công suất tín hiệu ra và công suất tín hiệu vào, được tính theo dB. Bộ khuếch đại có hệ số khuếch đại cao là mong muốn để đạt được hiệu quả năng lượng tốt.

• **Hiệu suất**

Hiệu suất là tỉ số giữa công suất phát và tổng công suất tiêu thụ. Với một bộ khuếch đại công suất, hiệu suất là tỉ số giữa công suất tín hiệu ra và công suất tiêu thụ của toàn bộ khuếch đại.

• **Độ nhạy bộ thu**

Độ nhạy bộ thu (theo dBm) là công suất tín hiệu tối thiểu ở bộ thu cần để có được E_b/N_0 định mức hay tỉ số lỗi bit/gói tin định mức. Độ nhạy tốt sẽ mở rộng dải cho phép của hệ thống.

• **Dải làm việc**

Trong khi băng trực giác thì dải của bộ phát đã rõ nhưng về mặt hình thức thì định nghĩa nó cần một số chú ý. Dải được xét đến khi không có nhiễu, nó hiển nhiên phụ thuộc vào công suất truyền dẫn cực đại, đặc tính của các antenna và sự suy giảm do môi trường. Khi điều chỉnh thì nó phụ thuộc vào tần số sóng mang, sơ đồ điều chế/ mã hoá mà nó sử dụng và tỉ lệ lỗi bit chấp nhận được ở bộ thu. Nó cũng phụ thuộc vào chất lượng của bộ thu. Các giá trị thông thường rất khó đưa ra ở đây nhưng nó thường nằm trong khoảng từ vài mét đến vài trăm mét.

• **Năng suất khối**

Năng suất khối của bộ thu là tỉ lệ lỗi bit đạt được của nó khi có nhiễu. Chính xác hơn, ở mức công suất nào nhiễu (ở khoảng cách cố định) có thể được gửi một độ dịch tần đã cho từ tần số sóng mang mà mục tiêu BER vẫn có thể được đáp ứng? Nhiễu ở độ dịch tần càng cao thì phải chịu mức công suất càng lớn. Hiển nhiên, năng suất khối có thể được cải thiện bằng cách đặt bộ lọc giữa antenna và bộ thu phát.

Một trường hợp quan trọng là nhiễu kênh liên kề truyền trên các tần số lân cận. Chặn kênh liên kề thể hiện khả năng của bộ thu phát để lọc ra các tín hiệu từ các băng tần liên kề (và do đó, giảm được nhiễu kênh liên kề), nó tác động trực tiếp đến **tỉ số tín hiệu trên nhiễu và tạp âm** (SINR - Signal to Interference and Noise Ratio).

• *Phát xạ ngoài băng*

Ngược lại với chặn kênh liền kề là phát xạ ngoài băng của bộ phát. Để hạn chế sự nhiễu loạn của các hệ thống khác hay của bản thân WSN trong cài đặt đa kênh, bộ phát phải tạo ra ít nhất có thể công suất truyền dẫn ngoài băng thông quy định của nó, chỉ tập trung quanh tần số sóng mang.

• *Cảm nhận sóng mang và RSSI*

Trong nhiều giao thức điều khiển truy cập môi trường, việc cảm nhận kênh vô tuyến, sóng mang đang bận (nút khác đang truyền) là một thông tin cần thiết. Bộ thu phải cung cấp được thông tin này. Độ chính xác của cảm nhận sóng mang này phụ thuộc vào cách thực hiện. Ví dụ, chuẩn IEEE 802.15.4 phân biệt:

- Năng lượng nhận trên ngưỡng, tuy nhiên tín hiệu bên dưới không cần tuân theo điều chế và các đặc tính phổ.

- Sóng mang phải được phát hiện, đó là một số tín hiệu tuân theo điều chế.

- Sóng mang phát hiện và năng lượng phải tồn tại.

Cường độ tín hiệu của gói tin nhận được có thể là thông tin có ích (ví dụ để ước lượng thô về khoảng cách từ bộ phát với giả thiết công suất truyền dẫn đã biết). Bộ thu phải cung cấp thông tin này trong **bộ chỉ thị cường độ tín hiệu nhận được** (RSSI - Received Signal Strength Indicator).

• *Độ ổn định tần số*

Độ ổn định tần số biểu thị mức độ biến đổi từ tần số trung tâm danh định khi các điều kiện môi trường của bộ tạo dao động như nhiệt độ, áp suất thay đổi. Trong các trường hợp đặc biệt, độ ổn định tần số kém có thể phá hỏng các tuyến thông tin, ví dụ như khi một nút được đặt dưới ánh nắng mặt trời trong khi lân cận nó lại đang trong bóng râm.

• *Dải điện áp*

Các bộ thu phát phải hoạt động tin cậy trong dải điện áp cung cấp. Nói cách khác, cần phải có mạch ổn định điện áp.

Nhiều nhà sản xuất chế tạo các bộ thu phát phù hợp với WSN. Thông thường, có một họ các thiết bị được chọn. Sản phẩm phổ biến hiện nay bao gồm RFM TR 1001, Chipcon CC 1000 và CC 2420 (như là một trong các model đầu tiên của IEEE 802.15.4) và họ Infineon TDA525x.

Một nét riêng biệt quan trọng và là sự khác nhau cơ bản so với các thiết bị liên lạc khác, đó là các bộ thu phát đơn giản thường thiếu thiết bị nhận dạng duy nhất. Ví dụ, mỗi thiết bị Ethernet có một địa chỉ mức MAC duy nhất để nhận dạng thiết bị này. Với các bộ thu phát đơn giản, chi phí thêm vào bộ nhận dạng này tương đối lớn so với tổng chi phí của thiết bị, do đó bộ nhận dạng duy nhất không thể có mặt trên tất cả các thiết bị. Tính hữu dụng của bộ nhận dạng thiết bị này được ghi nhận trong nhiều giao thức liên lạc và sự vắng mặt của nó sẽ có hậu quả lớn trong thiết kế giao thức.

Cải thiện các thiết kế thương mại này để tạo ra đặc tính tốt hơn ở mức tiêu thụ năng

lượng thấp hơn và giá thành giảm là các mục tiêu cần hướng tới. Những khó khăn phải đối mặt là độ truyền dẫn điện trở thấp hay các hạn chế của các bộ phận RF thụ động tích hợp.

Kiến trúc bộ thu phát

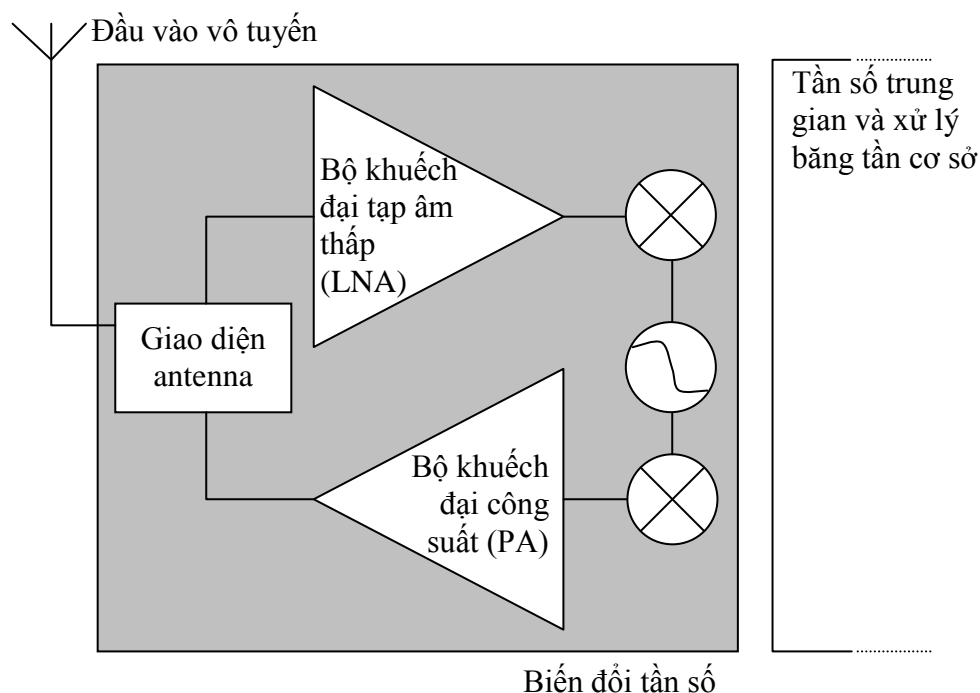
Kiến trúc chung của các bộ thu phát là đầu vào tần số vô tuyến (RF - Radio Frequency) và phân băng gốc:

- **Đầu vào tần số vô tuyến** thực hiện xử lý tín hiệu tương tự trong dải tần số vô tuyến hiện tại.

- **Bộ xử lý băng gốc** thực hiện xử lý tất cả các tín hiệu trong miền số và liên lạc với bộ xử lý của nút cảm biến hay mạch số khác.

Giữa hai phần này có một sự chuyển đổi tần số trực tiếp hoặc thông qua một/một vài tần số trung gian (IF - Intermediate Frequency). Ranh giới giữa miền tương tự và số được tạo bởi các bộ biến đổi tương tự/số (ADC - Analog/Digital Converter) và số/tương tự (DAC - Digital/Analog Converter).

Đầu vào RF thực hiện xử lý tín hiệu tương tự trong dải tần số vô tuyến hiện tại, ví dụ như dải 2.4 GHz ISM (công nghiệp, khoa học và y tế - Industrial, Scientific, and Medical). Đây là giai đoạn đầu tiên của liên lạc giữa các sóng điện từ và xử lý tín hiệu số của bộ thu phát. Một số thành phần quan trọng trong kiến trúc đầu vào RF được phác thảo trên hình 2.2.



Hình 2.2 Đầu cuối RF

- Bộ khuếch đại công suất (PA - Power Amplifier) nhận các tín hiệu đã chuyển đổi từ IF hoặc phân băng tần cơ sở và khuếch đại chúng để truyền qua antenna.

- Bộ khuếch đại tạp âm thấp (LNA - Low Noise Amplifier) khuếch đại tín hiệu đến lên các mức phù hợp với xử lý tiếp theo mà không làm giảm SNR đáng kể. Dải công suất của tín hiệu đến thay đổi từ tín hiệu rất yếu của các nút gần ranh giới nhận đến các tín hiệu rất mạnh của các nút ở vị trí gần. Dải này có thể lên đến 100 dB. Không có các hành

động điều khiển, LNA được kích hoạt tại mọi thời điểm và có thể tiêu thụ một tỉ lệ năng lượng đáng kể của bộ thu phát.

- Các thành phần như bộ dao động hay bộ dao động điều khiển bằng điện áp và bộ trộn được sử dụng để biến đổi tần số từ phổ RF thành các tần số trung gian hoặc băng tần cơ sở. Trong bộ trộn, tín hiệu đến ở tần số RF (f_{RF}) được nhân với tín hiệu có tần số cố định từ bộ dao động cục bộ (f_{LO}). Tín hiệu tần số trung gian nhận được có tần số $f_{LO}-f_{RF}$. Phụ thuộc kiến trúc đầu vào RF mà các thành phần khác như các bộ lọc cũng có mặt.

Các trạng thái hoạt động của bộ thu phát

Rất nhiều bộ thu phát có thể phân biệt được 4 trạng thái hoạt động sau:

Truyền: Ở **trạng thái truyền**, phần truyền của bộ thu phát được kích hoạt và antenna bức xạ năng lượng.

Nhận: Trong trạng thái nhận, phần nhận được kích hoạt.

Nghỉ: Bộ thu phát đã sẵn sàng nhận nhưng hiện thời không nhận được gì thì nói rằng nó đang ở **trạng thái nghỉ**. Ở trạng thái này, nhiều phần của mạch thu được kích hoạt và một số khác thì có thể tắt. Ví dụ, trong mạch đồng bộ, một số thành phần liên quan đến việc thu thập dữ liệu được kích hoạt trong khi các phần liên quan đến việc theo dõi có thể tắt và chỉ được kích hoạt khi quá trình thu thập có kết quả. Myers đã giới thiệu kỹ thuật tắt các phần trong mạch thu thập dữ liệu trong IEEE 802.11. Nhân tố chính của việc tiêu thụ năng lượng là sự **rò rỉ**.

Ngủ: Trong **trạng thái ngủ**, các phần quan trọng của bộ thu phát được tắt đi. Các bộ thu phát thường có các trạng thái ngủ khác nhau. Các trạng thái này khác nhau ở mạch tắt và liên quan đến **thời gian hồi phục** và **năng lượng khởi động**.

Ngăn xếp giao thức của nút cảm biến và phần mềm hoạt động phải quyết định xem trạng thái nào của bộ thu phát được bật dựa trên yêu cầu của các liên lạc hiện tại và dự đoán. Một vấn đề làm phức tạp quyết định này là hoạt động của sự thay đổi trạng thái cũng làm lãng phí năng lượng. Ví dụ, bộ thu phát thức dậy từ trạng thái ngủ để chuyển sang chế độ truyền yêu cầu thời gian và năng lượng khởi động. Trong thời gian khởi động, dữ liệu không thể truyền hoặc nhận. Vấn đề lịch trình trạng thái của các nút (tương đương với việc tắt/bật các bộ phận của nút/bộ thu phát) cũng như việc tối thiểu năng lượng tiêu thụ trung bình (còn được gọi là **điều khiển năng lượng**) sẽ khá phức tạp.

Các khái niệm vô tuyến nâng cao

Ngoài các khái niệm cơ bản về bộ thu phát ở trên, một số khái niệm nâng cao về liên lạc vô tuyến cũng là đối tượng của các công trình nghiên cứu hiện nay.

• *Thức dậy vô tuyến*

Quan sát khái niệm bộ thu phát được giới thiệu ở trên, một trong những hoạt động cần nhiều năng lượng nhất là đợi để được truyền và sẵn sàng để nhận tin. Trong thời gian này, mạch của bộ thu phải bật, vì vậy kênh vô tuyến có thể quan sát được, sử dụng năng lượng mà không cần bước trung gian. Trong khi vấn đề tắt yếu là bộ thu phải sử dụng

năng lượng cho quá trình đợi tin thì điều mong muốn lại là nó không mất năng lượng khi nút chỉ đợi gói tin đến. Kiến trúc của bộ thu cần phải không yêu cầu năng lượng nhưng vẫn có thể phát hiện khi gói tin bắt đầu đến. Để có được mục đích này, nó phải đủ mạnh để phát động một sự kiện nhằm thông báo đến các bộ phận khác về việc gói tin đến. Dựa trên sự kiện này, bộ thu chính có thể được bật và thực hiện nhận gói tin hiện tại.

Các khái niệm bộ thu như vậy được gọi là các **bộ thu thức**. Mục đích của nó chỉ là thức bộ thu chính mà không yêu cầu (một lượng đáng kể) năng lượng. Trong trường hợp đơn giản nhất, việc thức này sẽ xảy ra với mọi gói tin, còn cách phức tạp hơn là đưa ra quyết định bằng cách sử dụng thông tin địa chỉ phù hợp ở thời điểm bắt đầu của mỗi gói, nếu gói tin đến là đích thực sự của nút đó thì bộ thu chính mới thức dậy.

Các bộ thu phát phổ rộng

Các khái niệm bộ thu phát đơn giản dựa trên các sơ đồ điều chế như khoá dịch biên (ASK - Amplitude Shift Keying) hay khoá dịch tần (FSK - Frequency Shift Keying) có thể kém chất lượng do đặc tính bị hạn chế, đặc biệt trong trường hợp có nhiễu nhiều. Để khắc phục hạn chế này, người ta sử dụng các bộ thu phát phổ rộng. Tuy nhiên, nhược điểm của các bộ thu phát này là phần cứng phức tạp và giá thành cao, chính điều này đã ngăn cản nó trở thành bộ phận chủ đạo trong các WSN.

Liên lạc băng siêu rộng

Liên lạc băng siêu rộng là sự thay đổi khá cơ bản từ liên lạc không dây thông thường như đã đề cập ở trên. Thay vì điều chế, tín hiệu số trong tần số sóng mang, băng thông rất rộng được dùng để truyền trực tiếp dãy số như là những xung rất ngắn (để tạo các xung gần vuông góc thì yêu cầu phải có băng thông đáng kể do khái niệm này không thường sử dụng). Do đó, các xung này sử dụng phổ lớn, bắt đầu từ vài Hz lên đến vài GHz. Khó khăn ở đây là đồng bộ một cách có hiệu quả giữa người gửi và người nhận (để độ chính xác lên đến 1 phần triệu tỷ giây), vì vậy xung có thể được phát hiện một cách chính xác.

Bằng cách sử dụng băng tần rộng như vậy, liên lạc băng siêu rộng chòng chéo với phổ của hệ thống vô tuyến thông thường. Nhưng do sự trải rộng của tín hiệu nên công suất truyền rất nhỏ cũng đủ đáp ứng. Công suất này có thể đủ nhỏ, vì vậy nó biến mất trong nhiễu nền từ các hệ vô tuyến truyền thống.

Thông tin quang

Kahn và các nhà khoa học khác quan tâm đến việc sử dụng tuyến quang giữa các nút cảm biến. Ưu điểm chính là năng lượng/bit được yêu cầu cho cả việc tạo và phát hiện ánh sáng quang rất nhỏ. Diode phát quang (LED - Light-Emitting Diode) đơn giản là một ví dụ cho máy phát hiệu suất cao. Mạch yêu cầu cho bộ thu phát quang cũng đơn giản hơn và thiết bị cũng nhỏ hơn so với thiết bị tần số vô tuyến tương đương. Ngoài ra, liên lạc có thể đồng thời loại bỏ được nhiễu. Tuy nhiên, nhược điểm ở đây là giao tiếp ngang hàng cần phải có kết nối LOS và thông tin quang chịu ảnh hưởng lớn bởi các điều kiện thời tiết.

Siêu âm

Cả tần số vô tuyến và thông tin quang đều phù hợp với các môi trường ngoài trời. Tuy nhiên, trong một số ứng dụng, các nút cảm biến chỉ sử dụng trong môi trường mà sóng vô tuyến hay thông tin quang không thể dùng do các sóng này không xuyên qua môi trường được. Nước là một trong những môi trường như vậy và ứng dụng là giám sát sự ăn mòn của đáy biển để giúp xây dựng những cánh đồng gió ở ngoài khơi. Các cảm biến phải được triển khai dưới đáy biển và phải có sự liên lạc với nhau. Trong môi trường dưới nước như vậy, siêu âm là một môi trường liên lạc lý tưởng, nó có thể truyền qua khoảng cách tương đối dài với năng lượng tương đối nhỏ.

2.1.5 Các cảm biến và cơ cấu chấp hành

Đây chỉ là những quan điểm thô về các cảm biến và cơ cấu chấp hành sử dụng trong WSN.

Cảm biến: Cảm biến có thể được chia ra làm ba loại:

- **Cảm biến đơn hướng, thụ động:** Các cảm biến này có thể đo đại lượng vật lý ở nút cảm biến mà không cần điều khiển môi trường bằng cách tìm kiếm hoạt động, theo hướng này nó được gọi là thụ động. Hơn nữa, một số các cảm biến này còn có nguồn năng lượng riêng, tức chúng có thể có năng lượng cần thiết từ môi trường – năng lượng chỉ cần để khuếch đại tín hiệu tương tự. Không cần quan tâm đến “phương hướng” trong các phép đo này. Ví dụ điển hình cho loại cảm biến này là nhiệt kế, cảm biến ánh sáng, dao động, micro, độ ẩm...

- **Các cảm biến chùm hẹp, thụ động:** Các cảm biến này cũng là thụ động nhưng phải xác định được phương hướng của phép đo. Ví dụ điển hình là camera, nó có thể “thực hiện các phép đo” theo hướng đã cho, nhưng nó cũng có thể quay nếu có yêu cầu.

- **Các cảm biến chủ động:** Nhóm cảm biến cuối cùng tìm kiếm môi trường một cách chủ động, ví dụ như cảm biến siêu âm hoặc radar và một số loại cảm biến địa chấn, nó tạo ra các sóng xung kích bằng các tiếng nổ nhỏ.

Trên thực tế, các cảm biến loại này có thể sử dụng theo nhiều dạng khác nhau với các đặc trưng riêng. Hiển nhiên, các thỏa hiệp ở đây bao gồm độ chính xác, độ tin cậy, tiêu thụ năng lượng, chi phí, kích thước... Mỗi nút cảm biến đều có một **vùng bao phủ** nào đó mà ở đó, nó có thể có báo cáo tin cậy và chính xác liên quan đến đại lượng mà nó quan sát. Cụ thể hơn, mô hình phát hiện cảm biến được sử dụng, nó liên quan đến khoảng cách giữa cảm biến và sự kiện hay đối tượng phải phát hiện để có được xác suất phát hiện đối tượng.

Các cơ cấu chấp hành

Các cơ cấu chấp hành cũng đa dạng như cảm biến và dùng để thiết kế WSN, chúng thường đơn giản hơn về cách mô tả. Về cơ bản, tất cả các nút cảm biến đều có thể đóng, mở một chuyển mạch hay rơ le hoặc dùng để đặt giá trị theo một số cách. Liệu chúng có điều khiển được động cơ, bóng đèn hay các đối tượng vật lý khác mà không cần quan tâm tới các giao thức liên lạc được thiết kế không? Cuốn sách này sẽ xem xét các cơ cấu chấp hành một cách khá sơ lược và không có sự phân biệt giữa các loại khác nhau.

Tuy nhiên trong mạng thực tế, ta phải quan tâm đến đặc điểm riêng của các loại cơ cấu chấp hành khác nhau. Ngoài ra, để có được một thiết kế tốt trong các ứng dụng hệ thống

nhúng thì cần phải kết hợp cơ cấu chấp hành với cảm biến điều khiển.

2.1.6 Nguồn cung cấp của các nút cảm biến

Với các nút cảm biến, nguồn cung cấp là một bộ phận không thể thiếu. Nó có hai khía cạnh cơ bản: *thứ nhất* là năng lượng dự trữ và nguồn cung cấp theo các dạng yêu cầu; *thứ hai* là cố gắng bổ sung nguồn năng lượng đã tiêu thụ bằng cách “tìm kiếm” nó từ nguồn cung cấp bên ngoài nút theo thời gian.

Năng lượng dự trữ thường dùng pin. Pin AA thường chứa khoảng 2.2 – 2.5 Ah ở 1.5 V.

Năng lượng dự trữ: Pin

- **Pin truyền thống**

Nguồn cung cấp của nút cảm biến là pin, nó có thể không sạc lại được (pin chính), hoặc sạc lại được (pin phụ) nếu thiết bị tìm kiếm năng lượng ở trên nút.

Ở một số dạng khác, pin dự trữ năng lượng dạng điện hoá – các chất hoá học là nhân tố xác định chính trong kỹ thuật của pin. Pin có các yêu cầu sau:

- **Dung lượng:** Chúng phải có dung lượng cao, trọng lượng nhỏ, thể tích nhỏ, giá thành thấp. Đơn vị chính là năng lượng /thể tích (J/cm^3). Bảng 2.1 chỉ ra một số giá trị tiêu biểu của mật độ năng lượng sử dụng các công nghệ pin có quy mô lớn, truyền thống.

Pin chính			
Năng lượng hoá học (J/cm^3)	Pin kẽm không khí 3780	Lithium 2880	Alkaline 1200
Pin phụ			
Năng lượng hoá học (J/cm^3)	Lithium 1080	NiMHd 860	NiCd 650

Bảng 2.1 Mật độ năng lượng của các loại pin chính và phụ khác nhau.

- **Dung lượng dưới tải:** Pin được sử dụng trong các mô hình khác nhau vì nút cảm biến có thể tiêu thụ các mức năng lượng rất khác nhau theo thời gian và thậm chí sử dụng dòng lớn trong một số chế độ hoạt động nào đó.

Năng lượng tiêu thụ của các nút WSN thay đổi (xem chi tiết ở phần 2.2) và do đó, rất khó để đưa ra các nguyên tắc chính xác. Nhưng với hầu hết công nghệ, pin càng lớn thì có thể cung cấp càng nhiều năng lượng ngay lập tức. Hơn nữa, dung lượng pin được chỉ rõ bởi nhà sản xuất chỉ có giá trị khi dòng phóng điện cực đại không bị vượt quá vì dung lượng có thể bị giảm hoặc pin bị hỏng trước thời hạn.

- **Độ tự phóng điện:** Độ tự phóng điện của chúng phải thấp, phải sử dụng được trong thời gian dài (bằng cách sử dụng các công nghệ nào đó, pin chỉ hoạt động trong vài tháng).

Ví dụ, các pin kẽm không khí (Zinc-air batteries) có tuổi thọ rất ngắn (tính theo tuần) nhưng bù lại, nó có mật độ năng lượng cao.

- **Xạc pin:** Xạc phải có hiệu quả, thậm chí khi nguồn sạc thấp và không liên tục.

Một số kỹ thuật tìm kiếm năng lượng mô tả dưới đây chỉ có thể tạo ra dòng cỡ μA (nhưng có thể duy trì liên tục) tại mức điện áp chỉ vài vôn. Công nghệ pin hiện nay về cơ bản không xạc ở các giá trị này.

• **Sự hồi pin:** Hiệu ứng hồi pin - giống như tự xạc lại pin hết hoặc gần hết khi pin không còn dòng, dựa trên các quá trình khuếch tán hoá học bên trong cell. Tuổi thọ và dung lượng của pin có thể mở rộng một cách đáng kể nếu lấy hiệu ứng này làm đòn bẩy. Như trong một ví dụ, có thể dùng nhiều pin mắc song song và lịch trình xạc pin phụ thuộc vào các tính chất hồi pin và các yêu cầu năng lượng của các hoạt động được hỗ trợ.

Biến đổi DC - DC

Không may, pin (hay dạng dự trữ năng lượng khác) không chấp nhận nguồn cấp trực tiếp cho nút cảm biến. Một vấn đề thông thường là điện áp của pin giảm khi dung lượng của nó giảm. Do đó, nếu dùng nguồn yếu cung cấp cho các mạch của nút cảm biến thì nó sẽ ảnh hưởng trực tiếp đến các tần số bộ dao động và công suất truyền dẫn – nút có pin yếu sẽ có dải truyền dẫn nhỏ hơn nút có pin khoẻ.

Một bộ biến đổi DC – DC có thể sử dụng để vượt qua vấn đề này bằng cách điều chỉnh điện áp cung cấp cho các mạch của nút. Để đảm bảo điện áp không đổi mặc dù điện áp cung cấp cho pin có thể giảm, bộ biến đổi DC – DC phải nâng dòng từ pin cao lên khi pin bị yếu đi. Mặc dù bộ biến đổi DC – DC không tiêu thụ năng lượng cho các hoạt động của nó nhưng nó làm giảm hiệu quả của pin. Nhìn chung, bộ biến đổi này có ưu điểm nhiều hơn nhược điểm.

Tìm kiếm năng lượng: Một số cách dự trữ năng lượng giới thiệu ở trên trái với thông lệ - các pin nhiên liệu, động cơ nhiệt cực nhỏ, tính phóng xạ - biến đổi năng lượng được dự trữ thành điện để dễ sử dụng hơn năng lượng từ pin. Toàn bộ nguồn năng lượng được dự trữ trên bản thân nút – một khi nguồn cung cấp nhiên liệu hết thì nút sẽ hỏng. Để đảm bảo các nút và WSN tồn tại trong một thời gian dài thì nguồn năng lượng bị hạn chế là không chấp nhận được. Hơn nữa, năng lượng từ môi trường của nút phải được lấy và làm cho phù hợp với nút – **tìm kiếm năng lượng** phải được thực hiện.

2.2 Tiêu thụ năng lượng của các nút cảm biến

2.2.1 Các trạng thái hoạt động với mức tiêu thụ công suất khác nhau.

Như đã nói ở phần trước, nguồn năng lượng cho nút cảm biến là pin có dung lượng nhỏ, có thể xạc lại bằng cách tìm kiếm năng lượng rất phức tạp và không ổn định. Do đó, tiêu thụ năng lượng của nút cảm biến phải được giám sát một cách chặt chẽ. Thành phần tiêu thụ năng lượng chính là bộ điều khiển, đầu vào vô tuyến đến một số mức độ bộ nhớ và phụ thuộc vào loại cảm biến.

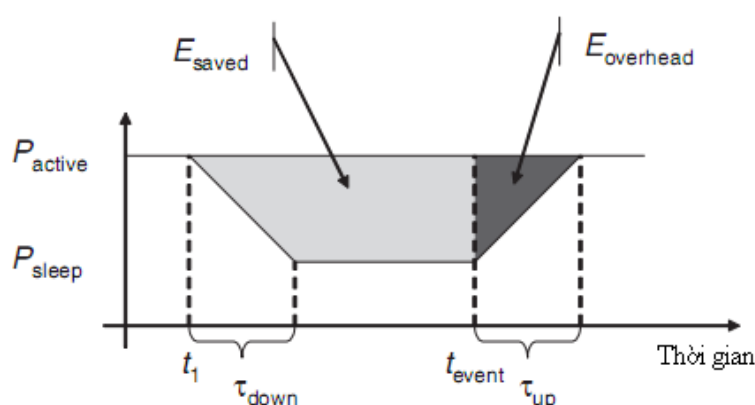
Một đóng góp quan trọng để làm giảm tiêu thụ công suất của các thành phần này là thiết kế các chip công suất thấp. Nhưng điều này mới chiếm một nửa thành công vì một số ưu điểm của các thiết kế này dễ bị phí phạm khi các thành phần hoạt động không đúng cách.

Quan sát nhận thấy rằng phần lớn thời gian nút cảm biến không dây không làm gì. Do

đó, tốt nhất là nên tắt nó đi. Đương nhiên, nó có thể thức trở lại trên cơ sở của các tác nhân bên ngoài hay trên cơ sở thời gian. Do vậy, tắt hoàn toàn một nút là không thể nhưng trạng thái hoạt động của nó có thể phù hợp với các nhiệm vụ. Giới thiệu và sử dụng nhiều trạng thái hoạt động với mức tiêu thụ năng lượng giảm bằng cách giảm chức năng là công nghệ cốt lõi cho nút cảm biến không dây có hiệu quả năng lượng. Sự thật là phương pháp này đã được biết đến từ phần cứng máy tính cá nhân chuẩn, ví dụ như **giao diện nguồn và cấu hình nâng cao** (ACPI - Advanced Configuration and Power Interface) đưa ra một trạng thái trình bày toàn bộ bộ máy hoạt động và bốn trạng thái ngủ của chức năng phân loại/tiêu thụ năng lượng/thời gian thức (thời gian cần thiết để quay trở lại trạng thái hoạt động hoàn toàn).

Các chế độ này có thể đưa vào tất cả các thành phần của nút cảm biến, cụ thể là bộ điều khiển, đầu vào vô tuyến, bộ nhớ và các cảm biến. Các mô hình khác nhau thường hỗ trợ cho số lượng các trạng thái ngủ khác nhau với các đặc tính khác nhau. Với bộ điều khiển, các trạng thái điển hình là “hoạt động”, “ngủ” và “ngủ”; Một modem vô tuyến có thể tắt/bật bộ phát, bộ thu hoặc cả hai. Các cảm biến và bộ nhớ cũng có thể tắt/bật.

Trong khi mô hình trạng thái ngủ có chọn lọc đủ dễ hiểu thì sự phức tạp lại nằm ở việc truyền dẫn giữa các trạng thái theo thời gian và năng lượng. Giả thiết thông thường là trạng thái ngủ càng sâu thì thời gian và năng lượng nó cần để dậy và quay trở lại trạng thái hoạt động hoàn toàn càng lớn. Do đó, về phương diện tiêu thụ năng lượng thì nên duy trì trạng thái nghỉ thay vì chuyển đến các trạng thái ngủ sâu hơn.



Hình 2.3 Tiết kiệm năng lượng và chi phí cho các chế độ ngủ

Hình 2.3 mô tả ý kiến này dựa trên mô hình thường sử dụng. Ở thời điểm t_1 , quyết định liệu thiết bị (bộ vi điều khiển) có chuyển sang trạng thái ngủ để giảm năng lượng tiêu thụ từ P_{active} sang P_{sleep} không? Nếu nó vẫn hoạt động và sự kiện tiếp theo xảy ra tại thời điểm t_{event} thì năng lượng tổng $E_{\text{active}} = P_{\text{active}}(t_{\text{event}} - t_1)$ đã được sử dụng một cách vô ích. Đặt thiết bị ở trạng thái ngủ thì nó yêu cầu thời gian τ_{down} cho đến khi đạt được trạng thái ngủ. Đơn giản hoá, giả sử rằng tiêu thụ năng lượng trung bình trong giai đoạn này là $(P_{\text{active}} + P_{\text{sleep}})/2$. Khi đó P_{sleep} là năng lượng được tiêu thụ cho đến thời điểm t_{event} . Gộp lại, năng lượng $\tau_{\text{down}} (P_{\text{active}} + P_{\text{sleep}})/2 + (t_{\text{event}} - t_1 - \tau_{\text{down}})P_{\text{sleep}}$ được yêu cầu trong trạng thái ngủ, ngược lại là năng lượng $(t_{\text{event}} - t_1)P_{\text{active}}$ khi vẫn hoạt động. Do đó, sự tiết kiệm năng lượng ở đây là :

$$E_{\text{saved}} = (t_{\text{event}} - t_1) P_{\text{active}} - \left(\tau_{\text{down}} (P_{\text{active}} + P_{\text{sleep}}) / 2 + (t_{\text{event}} - t_1 - \tau_{\text{down}}) P_{\text{sleep}} \right) \quad (0.3)$$

Một khi sự kiện xảy ra:

$$E_{\text{overhead}} = \tau_{\text{up}} (P_{\text{active}} + P_{\text{sleep}}) / 2 \quad (0.4)$$

là năng lượng bổ sung để quay trở lại trạng thái hoạt động trước khi sự kiện có thể xử lý, với giả thiết đơn giản về tiêu thụ năng lượng trung bình trong thời gian bổ sung. Năng lượng này là chi phí chung do không nhận được hoạt động hữu dụng nào trong thời gian này. Rõ ràng, chuyển sang chế độ ngủ chỉ có lợi nếu $E_{\text{overhead}} < E_{\text{saved}}$, hay tương tự, nếu thời gian để sự kiện tiếp theo xảy ra là đủ lớn :

$$(t_{\text{event}} - t_1) > \frac{1}{2} \left(\tau_{\text{down}} + \frac{P_{\text{active}} + P_{\text{sleep}}}{P_{\text{active}} - P_{\text{sleep}}} \tau_{\text{up}} \right) \quad (0.5)$$

2.2.2 Tiêu thụ năng lượng bộ vi xử lý

Tiêu thụ công suất cơ bản trong các trạng thái hoạt động rời rạc.

Các bộ điều khiển nhúng thường thực hiện nhiều trạng thái hoạt động như đã liệt kê ở trên. Nó cũng tương đối dễ điều khiển. Một số ví dụ sau là cách giải thích tốt nhất cho ý kiến này.

Intel StrongARM

The Intel StrongARM cung cấp ba chế độ ngủ:

- Ở chế độ bình thường, tất cả các phần của bộ xử lý được cấp nguồn đầy đủ. Năng lượng tiêu thụ lên đến 40 mW.
- Ở chế độ nghỉ, clock của CPU bị dừng, clock gắn với thiết bị ngoại vi được kích hoạt. Một số ngắt sẽ làm cho nó quay trở về chế độ bình thường. Công suất tiêu thụ lên đến 100 mW.
- Ở chế độ ngủ, chỉ clock thời gian thực là vẫn hoạt động. Thức xảy ra sau khi bộ định thời ngắt 160 ms. Công suất tiêu thụ lên đến 50 μ s.

Texas Instruments MSP 430

Họ MSP430 mô tả một dải rộng các chế độ hoạt động: một chế độ hoạt động đầy đủ tiêu thụ khoảng 1.2 mW (tất cả giá trị năng lượng được cho ở 1 MHz và 3V). Tổng cộng có tất cả 4 chế độ ngủ. Chế độ ngủ sâu nhất, LPM4, chỉ tiêu thụ 0.3 μ W nhưng bộ điều khiển chỉ thức từ các ngắt bên ngoài. Chế độ cao hơn tiếp theo, LMP3, clock vẫn chạy và có thể được sử dụng để lên lịch trình quá trình thức, nó tiêu thụ năng lượng khoảng 6 μ W.

Atmel ATmega

Atmel ATmega 128L có sáu chế độ tiêu thụ năng lượng, về nguyên tắc cơ bản nó giống như MSP 430 nhưng khác nhau ở một số chi tiết. Tiêu thụ công suất thay đổi trong

khoảng 6 mW đến 15 mW ở các trạng thái nghỉ và thức, khoảng 75 μ W ở các chế độ tắt nguồn.

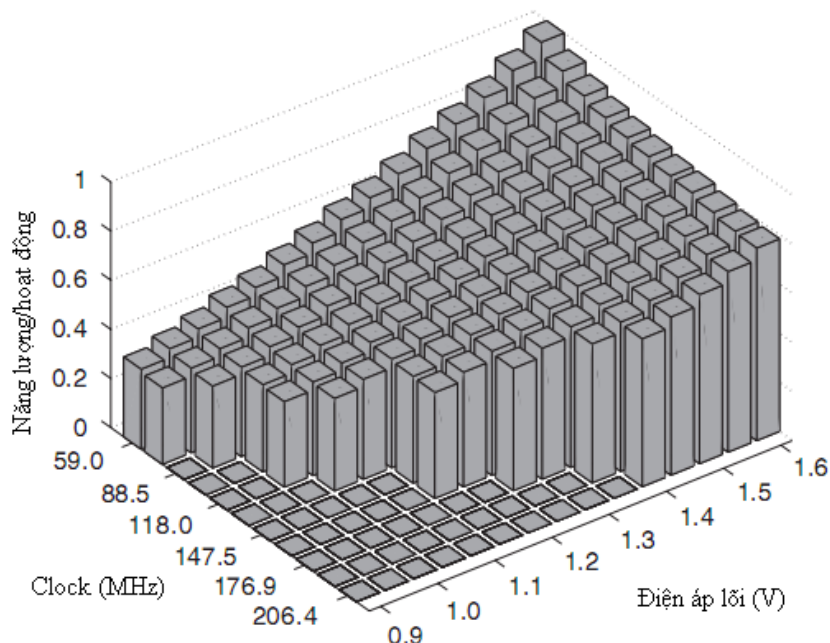
Thang điện áp động

Có thể phức tạp hơn các trạng thái hoạt động rời rạc là sử dụng khái niệm liên tục của liên lạc chức năng/nguồn bằng cách thích ứng tốc độ với các hoạt động của bộ điều khiển. Ý tưởng này chọn tốc độ tốt nhất có thể để tính toán nhiệm vụ phải hoàn thành theo các giới hạn đã cho. Một giải pháp hiển nhiên là bật bộ điều khiển ở chế độ hoạt động hoàn toàn, tính toán nhiệm vụ ở tốc độ cao nhất và quay trở về trạng thái ngủ nhanh nhất có thể.

Phương pháp khác là tính toán nhiệm vụ chỉ ở tốc độ yêu cầu để kết thúc nó trước thời hạn. Lý do cơ bản là bộ điều khiển chạy ở tốc độ thấp hơn thì tốc độ clock cũng thấp hơn, tiêu thụ ít năng lượng hơn tốc độ cực đại. Điều này là do nguồn cấp có thể giảm ở các tốc độ clock thấp hơn nhưng vẫn đảm bảo hoạt động đúng. Kỹ thuật này gọi là **thang điện áp động** (DVS - Dynamic Voltage Scaling).

Kỹ thuật này có lợi đối với các chip CMOS: khi công suất tiêu thụ thực tế P tỉ lệ với bình phương điện áp cung cấp V_{DD} thì giảm điện áp là cách hiệu quả để giảm công suất tiêu thụ. Công suất tiêu thụ cũng phụ thuộc vào tần số f , do đó $P \sim f \cdot V_{DD}^2$.

Thang điện áp động cũng giảm tiêu thụ năng lượng. Ví dụ, bộ xử lý Transmeta Crusoe có thang điện áp từ 700 MHz ở 1.65 V xuống đến 200 MHz ở 1.1 V. Công suất tiêu thụ giảm theo thừa số $700 \cdot (1.65)^2 / 200 \cdot (1.1)^2 = 7.875$, nhưng tốc độ chỉ giảm theo thừa số $700/200 = 3.5$. Do đó năng lượng yêu cầu/lệnh giảm $3.5/7.875 \approx 44\%$. Các bộ xử lý và vi điều khiển khác cũng tương tự như vậy. Hình 2.4 là ví dụ về StrongARM SA-1100. Lý do cuối cùng của sự cải tiến này là hình dáng lồi của hàm công suất theo tốc độ khi thay đổi nguồn cung cấp.



Hình 2.4 Năng lượng/hoạt động với dải điện áp động trên Intel StrongARM-1100

Khi áp dụng dải điện áp động ta phải chú ý để bộ điều khiển hoạt động theo các đặc điểm kỹ thuật của nó. Tồn tại tốc độ clock cực đại và cực tiểu cho mỗi thiết bị và tại mỗi tốc

độ lại có ngưỡng cực đại và cực tiểu phải tuân theo. Do đó, khi không có gì để xử lý thì các chế độ ngủ cũng vẫn là một lựa chọn. Muốn sử dụng điện áp tùy ý thì phải có bộ biến đổi DC – DC thực sự hiệu quả.

2.2.3 Bộ nhớ

Trên góc nhìn năng lượng, loại thích hợp nhất là bộ nhớ trên chip của bộ vi điều khiển hay bộ nhớ FLASH, RAM ngoài chip rất ít khi sử dụng. Thực tế là công suất cần cho bộ nhớ trên chip thường nằm trong lượng công suất tiêu thụ đã cho của các bộ điều khiển.

Do đó, phù hợp nhất vẫn là bộ nhớ FLASH, cấu trúc và các sử dụng của bộ nhớ FLASH có thể ảnh hưởng nghiêm trọng tới tuổi thọ của nút. Các thông số có liên quan là thời gian đọc, viết và năng lượng tiêu thụ.

Tất cả thông tin này đều có thể biết được từ dải dữ liệu của nhà sản xuất và thay đổi phụ thuộc vào một số yếu tố. Thời gian đọc và năng lượng tiêu thụ cho việc đọc thường tương đương nhau giữa các loại bộ nhớ FLASH khác nhau. Viết lại phức tạp hơn một chút, nó phụ thuộc vào loại dữ liệu mà nó truy cập (các byte riêng lẻ hay các trang có kích thước khác nhau). Một phương tiện dùng để so sánh là nhìn vào số lượng viết đè lên trên toàn chip. Sự khác nhau đáng kể là năng lượng tiêu thụ cho việc xóa và viết, tỉ lệ lên đến 900:1 giữa các loại bộ nhớ khác nhau.

Do đó, viết trong bộ nhớ FLASH có thể là nhiệm vụ tiêu tốn năng lượng và thời gian và tốt nhất là nên tránh, nếu có thể.

2.2.4 Các bộ thu phát vô tuyến

Bộ thu phát vô tuyến có hai nhiệm vụ cần thiết: truyền và nhận dữ liệu giữa các cặp nút. Tương tự như các bộ vi điều khiển, các bộ thu phát vô tuyến có thể hoạt động ở các chế độ khác nhau, chế độ đơn giản nhất là tắt/bật. Để có được mức tổng năng lượng tiêu thụ thấp cần thiết, nó phải được tắt trong hầu hết thời gian và chỉ được kích hoạt khi cần thiết – nó làm việc **với hệ số lấp đầy** (duty cycle – là tỉ lệ thời gian hoạt động trên thời gian một chu kỳ) thấp. Nhưng điều này sẽ làm tăng độ phức tạp, thời gian và năng lượng.

Mô hình tiêu thụ năng lượng trong khi truyền

Về mặt nguyên tắc, năng lượng dùng cho bộ phát lấy từ hai nguồn. *Phần thứ nhất* là do việc tạo tín hiệu RF, nó phụ thuộc chủ yếu vào sơ đồ điều chế và khoảng cách mục tiêu và do đó, công suất truyền dẫn P_{tx} là công suất phát xạ bởi antenna. *Phần thứ hai* là do các thiết bị điện cần để tổng hợp tần số, biến đổi tần số, các bộ lọc... Các giá trị này về cơ bản là không đổi.

Một trong những quyết định quan trọng nhất là khi nào truyền gói tin phụ thuộc vào việc chọn P_{tx} . Giả sử công suất truyền mong muốn P_{tx} đã biết, phụ thuộc vào các thông số của hệ thống như năng lượng/nhiều và tạp âm E_b/N_0 , hệ số băng thông η_{BW} , khoảng cách d và hệ số suy hao đường truyền γ .

Công suất truyền được tạo bởi bộ khuếch đại của máy phát. Công suất tiêu thụ P_{amp} của nó phụ thuộc vào kiến trúc bộ khuếch đại nhưng trên tất cả, nó phụ thuộc vào công suất mà nó tạo ra. Trong mô hình đơn giản nhất, hai giá trị này tỉ lệ với nhau, nhưng đây là trường

hợp quá đơn giản. Với mô hình phức tạp hơn, giả sử rằng mức công suất không đổi nào đó luôn được yêu cầu, không phụ thuộc vào công suất phát xạ, cộng thêm độ lệch tỉ lệ:

$$P_{\text{amp}} = \alpha_{\text{amp}} + \beta_{\text{amp}} \cdot P_{\text{tx}} \quad (0.6)$$

Trong đó α_{amp} , β_{amp} là các hằng số phụ thuộc vào công nghệ xử lý và kiến trúc bộ khuếch đại.

Ví dụ, theo Min và Chandrakasan, với các nút $\mu\text{AMPS-1}$, $\alpha_{\text{amp}} = 174 \text{ mW}$ và $\beta_{\text{amp}} = 5.0$. Do đó, **hiệu suất của bộ khuếch đại công suất** η_{PA} với công suất phát xạ $P_{\text{tx}} = 0 \text{ dBm} = 1 \text{ mW}$ là:

$$\mu_{\text{PA}} = \frac{P_{\text{tx}}}{P_{\text{amp}}} = \frac{1 \text{ mW}}{174 + 5.0 \cdot 1 \text{ mW}} \approx 0.55\% \quad (0.7)$$

Mô hình này cho thấy hiệu suất của bộ khuếch đại $P_{\text{tx}}/P_{\text{amp}}$ là tốt nhất khi công suất đầu ra đạt cực đại. Tuy nhiên, công suất cực đại không cần thiết trong các trường hợp thông thường nên thiết kế này không phải là cái có lợi nhất. Ví dụ, trong các hệ thống tế bào, bộ khuếch đại thường không hoạt động ở công suất đầu ra cực đại. Trong khi không hiểu tại sao lại đưa điều này vào các WSN nhưng nó đã đem lại kết quả thật bất ngờ đối với các mạng dày đặc khi sử dụng các bộ khuếch đại có các đặc tính hiệu suất khác nhau.

Sau đây ta sẽ xem xét mô hình trong phương trình (0.6). Ngoài bộ khuếch đại, mạch khác cũng sẽ được cấp nguồn trong quá trình truyền dẫn, ví dụ như các bộ xử lý băng tần cơ bản. Nguồn này được gọi là P_{txElec} .

Năng lượng để truyền gói tin có độ dài n bit (bao gồm tất cả các header) khi đó phụ thuộc vào thời gian gửi gói tin, được xác định bởi tốc độ bit danh định R , tốc độ mã hoá R_{code} và tổng năng lượng tiêu thụ trong quá trình truyền dẫn. Nếu bộ thu phát được bật trước khi truyền thì chi phí khởi động cũng phải tính đến (hầu hết dùng để sắp xếp các bộ dao động điều khiển bằng điện áp, các vòng lặp khoá pha). Phương trình (2.8) tổng kết tất các hoạt động này :

$$E_{\text{tx}}(n, R_{\text{code}}, P_{\text{amp}}) = T_{\text{start}} \cdot P_{\text{start}} + \frac{n}{R \cdot R_{\text{code}}} (P_{\text{txElec}} + P_{\text{amp}}) \quad (0.8)$$

Trong đó T_{start} , P_{start} là thời gian và năng lượng để khởi động. Ta thấy rằng phương trình này không phụ thuộc vào sơ đồ điều chế được chọn cho quá trình truyền dẫn. Các phép đo dựa trên phần cứng IEEE 802.11 đã chỉ ra rằng có chút phụ thuộc vào sơ đồ điều chế, nhưng sự khác nhau giữa 1 Mbit/s và 11 Mbit/s nhỏ hơn 10% cho tất cả các giá trị công suất truyền dẫn được tính đến. Vì vậy, đây là một sự đơn giản hoá có thể chấp nhận được.

Tuy nhiên, với giả thiết là chi phí mã hoá chỉ phụ thuộc vào tốc độ mã thì đây là một giả thiết có thể chấp nhận được. Trong mô hình này, không xét đến hiệu quả của antenna, tức ta đã giả thiết rằng có một antenna hoàn hảo. Nếu không sẽ có sự suy giảm công suất giữa đầu ra của PA và công suất phát xạ.

Mô hình này có thể dễ dàng được nâng cấp bằng các tác dụng của mã hoá **hiệu chỉnh phòng lỗi** (FFC - Forward Error Correction). FFC sẽ tăng số lượng bit được hiệu chỉnh mà

năng lượng cho việc mã hoá lại không đáng kể.

Mô hình tiêu thụ năng lượng trong khi nhận

Tương tự như bộ phát, bộ thu cũng có thể tắt hoặc bật. Khi bật, nó có thể chủ động nhận tin hoặc nghỉ, quan sát kênh truyền và sẵn sàng nhận. Hiển nhiên, tiêu thụ năng lượng trong khi tắt là không đáng kể. Sự khác nhau giữa trạng thái nhận và nghỉ là rất nhỏ, thậm chí trong hầu hết mục đích, nó được giả thiết bằng 0 (tức không có sự khác nhau).

Để giải thích, năng lượng P_{rcvd} yêu cầu để nhận tin có thành phần khởi động $T_{start}P_{start}$ tương tự như trường hợp truyền khi bộ thu tắt (ở đây thời gian khởi động bằng với việc truyền và nhận). Nó cũng là thành phần tỉ lệ với thời gian gói tin $n/R.R_{code}$. Trong thời gian nhận thực tế này, mạch bộ thu được nối với nguồn, ví dụ nó yêu cầu nguồn P_{rxElec} để điều khiển LNA trong đầu vào RF. Thành phần cuối cùng là chi phí giải mã E_{decBit} , nó có ở mọi bit, chi phí giải mã này có thể phụ thuộc đáng kể vào FEC mà nó sử dụng. Phương trình (2.9) sẽ tổng kết các thành phần này :

$$E_{rcvd} = T_{start}P_{start} + \frac{n}{R.R_{code}}P_{rxElec} + nE_{decBit} \quad (0.9)$$

Năng lượng giải mã tương đối phức tạp đối với mô hình, nó phụ thuộc vào các tham số của phần cứng và hệ thống (ví dụ như nó có thể được thực hiện trong phần cứng chuyên dụng như bộ giải mã Viterbi chuyên dụng cho các mã chập), nguồn cung cấp, thời gian giải mã/bit (phụ thuộc vào tốc độ xử lý, bị ảnh hưởng bởi các kỹ thuật như DVS), độ dài ràng buộc K của mã đang sử dụng và các tham số khác.

Ký hiệu	Mô tả	Ví dụ về bộ thu phát		
		μ AMPS-1	WINS	DEDUSA-II
α_{amp}	Phương trình (0.6)	174 mW	N/A	N/A
β_{amp}	Phương trình (0.6)	5.0	8.9	7.43
P_{amp}	Bộ khuếch đại công suất	179-674 mW	N/A	N/A
P_{rxElec}	Năng lượng nhận	279 mW	368.3 mW	12.48 mW
P_{rxIdle}	Nghỉ để nhận	N/A	344.2 mW	12.34 mW
P_{Start}	Năng lượng khởi động	58.7 mW	N/A	N/A
P_{txElec}	Năng lượng truyền	151 mW	\approx 386 mW	11.61 mW
R	Tốc độ truyền	1 Mbps	100 kbps	OOK 30 kpbs ASK 115.2 kbps
T_{Start}	Thời gian khởi động	466 μ s	N/A	N/A

Bảng 2.2 Một số tham số về tiêu thụ năng lượng của bộ thu phát (N/A : không cho phép)

2.2.5 Mối quan hệ giữa tính toán và liên lạc

Nhìn vào tổng số tiêu thụ năng lượng của các bộ vi điều khiển và bộ thu phát vô tuyến, một câu hỏi đặt ra là cách tốt nhất để sử dụng các nguồn năng lượng quý báu của nút cảm biến

là gì? Gửi dữ liệu đi hay tính toán nó sẽ tốt hơn? Mối quan hệ về tiêu thụ năng lượng giữa việc gửi dữ liệu và tính toán là gì?

Một lần nữa, mối quan hệ này phụ thuộc phần lớn và phần cứng riêng biệt đang sử dụng. Việc tính một lệnh đơn trên bộ vi điều khiển yêu cầu khoảng 1 nJ. 1 nJ đủ để giữ một mẫu đơn trong bộ thu phát vô tuyến. Các bộ thu phát bluetooth cho rằng yêu cầu 100 nJ để truyền một bit đơn. Đối với phần cứng khác, tỉ lệ tiêu thụ năng lượng để gửi một bit so với việc tính một lệnh đơn khoảng từ 1500 đến 2700 đối với các nút Rockwell, từ 220 đến 2900 đối với các nút MEDUSA II và khoảng 1400 đối với các nút WINS NG 2.0.

Rõ ràng là liên lạc tốn nhiều năng lượng hơn tính toán nhưng năng lượng dùng để tính toán không thể dễ dàng bỏ qua.

2.2.6 Tiêu thụ năng lượng của các cảm biến và cơ cấu chấp hành

Đưa ra nguyên tắc về tiêu thụ năng lượng của các cảm biến và cơ cấu chấp hành thực tế là điều không thể vì chúng rất đa dạng, ví dụ như các cảm biến nhiệt độ hay ánh sáng thụ động thì năng lượng tiêu thụ có thể bỏ qua so với các thiết bị khác trên nút vô tuyến. Với các thiết bị tích cực, ví dụ như radar siêu âm, thì năng lượng tiêu thụ lại đáng kể và phải xét đến kích thước của nguồn cung cấp trên nút cảm biến, không được hoạt động quá mức của pin.

Hơn nữa, tốc độ mẫu thực sự quan trọng, không chỉ mẫu có tần số càng lớn thì yêu cầu năng lượng từ các cảm biến càng nhiều mà dữ liệu phải xử lý cũng tăng lên và có thể phải liên lạc đến một nơi nào đó nữa.

Cảm biến	Độ chính xác	Tính hoán đổi	Tốc độ mẫu (Hz)	Khởi động (ms)	Dòng (mA)
Điện trở quang	N/A	10 %	2000	10	1.235
Nhiệt độ I2C	1 K	0.2 K	2	500	0.15
Khí áp	1.5 mbar	0.5 %	10	500	0.01
Bar. áp suất. nhiệt độ	0.8 K	0.24 K	10	500	0.01
Độ ẩm	2 %	3 %	500	500-3000	0.775
Pin nhiệt điện	3K	5 %	2000	200	0.17
Nhiệt kế	5 K	10 %	2000	10	0.126

Bảng 2.3 Ví dụ về một số đặc tính của các cảm biến

2.3 Các hệ điều hành và môi trường thực hiện

2.3.1 Các hệ điều hành nhúng

Nhiệm vụ thông thường của một hệ điều hành là điều khiển và bảo vệ truy cập tới các tài nguyên (bao gồm sự hỗ trợ cho đầu vào/đầu ra), quản lý sự cấp phát đến các người dùng khác nhau và hỗ trợ việc thực hiện cùng lúc của một số quá trình và liên lạc giữa các quá trình

này. Tuy nhiên, các nhiệm vụ này chỉ được yêu cầu **một phần** trong hệ thống nhưng vì **việc thực hiện mã lệnh** có nhiều hạn chế và thường phù hợp hơn trong các hệ thống đa dụng.

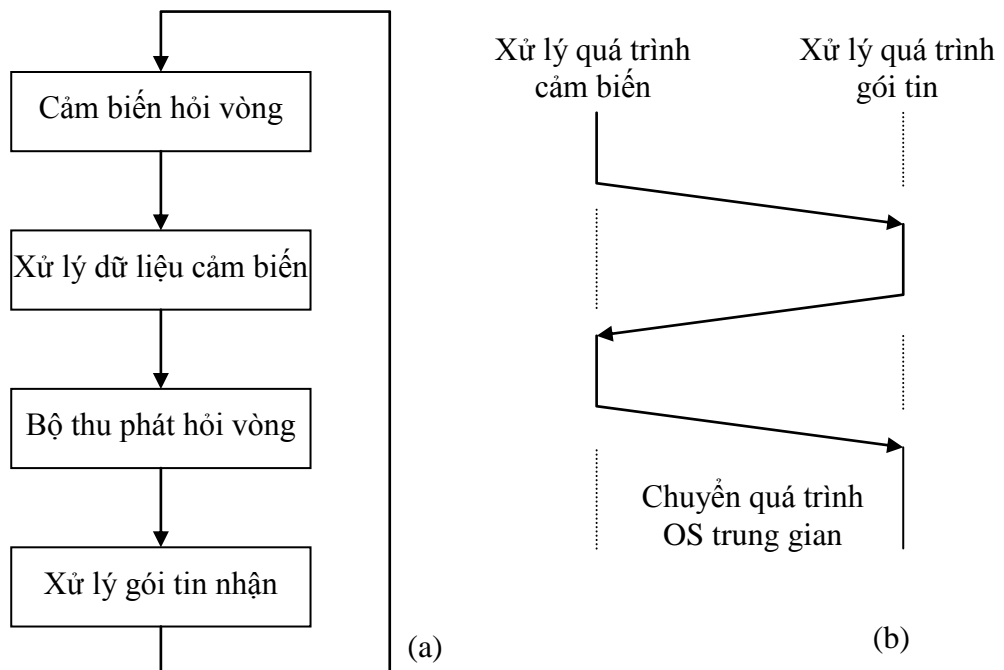
Đúng hơn, một hệ điều hành hay môi trường thực hiện càng đơn giản thì càng phù hợp. WSN sẽ hỗ trợ các nhu cầu chuyên dụng của chúng. Đặc biệt, nhu cầu về hiệu quả năng lượng sẽ yêu cầu sự hỗ trợ để quản lý năng lượng, ví dụ như điều khiển tắt của các thành phần riêng rẽ hay kỹ thuật **thang điện áp động** (DVS). Các thành phần bên ngoài như các cảm biến, modem vô tuyến hay các bộ định thời phải được xử lý một cách dễ dàng và có hiệu quả, phải xử lý được thông tin không đồng bộ (tại thời điểm bất kỳ).

Tất cả điều này yêu cầu một mô hình lập trình phù hợp, một phương pháp rõ ràng để xây dựng ngăn xếp giao thức và hỗ trợ rõ ràng cho việc quản lý năng lượng.

2.3.2 Mô hình lập trình và các giao diện lập trình ứng dụng

Lập trình điều khiển

Câu hỏi đầu tiên cho mô hình lập trình là nó hỗ trợ điều khiển như thế nào? Hỗ trợ thực hiện điều khiển là điều cốt yếu của các nút WSN vì chúng sử dụng dữ liệu đến từ các nguồn khác nhau (từ nhiều cảm biến hay bộ thu phát vô tuyến) tại các thời điểm bất kỳ. Ví dụ, hệ thống có thể thăm dò cảm biến để quyết định xem dữ liệu có hữu dụng và được xử lý đúng cách hay không, sau đó thăm dò bộ thu phát để kiểm tra xem liệu gói tin có hữu dụng không rồi xử lý gói tin ngay lập tức... (hình 2.5). Mô hình tuần tự đơn giản như vậy (hình 2.5(a)) sẽ hoạt động với nguy cơ bị mất gói tin trong khi gói tin được xử lý hoặc khi thông tin từ cảm biến được xử lý. Nguy cơ này sẽ cao nếu quá trình xử lý dữ liệu từ cảm biến hay các gói tin đến chiếm một lượng lớn thời gian. Do đó, mô hình lập trình tuần tự, đơn giản rõ ràng là không hiệu quả.



Mô hình lập trình tuần tự

Mô hình lập trình dựa trên quá trình

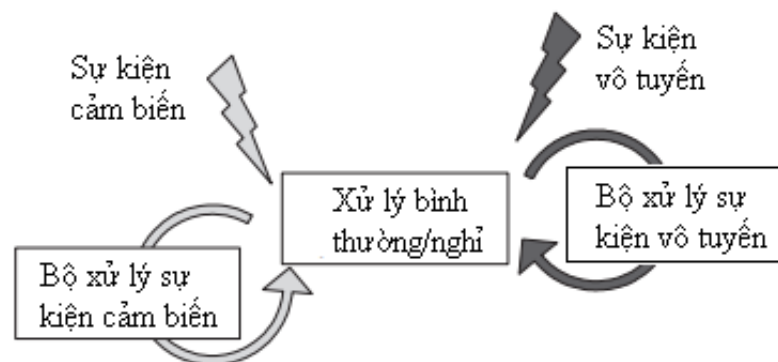
Hình 2.5 Hai mô hình không tương thích với hệ điều hành WSN

Điều khiển dựa trên quá trình

Các hệ điều hành đa dụng hiện đại nhất hỗ trợ thực hiện điều khiển nhiều quá trình trên CPU đơn lẻ. Do đó, phương pháp dựa trên quá trình là lựa chọn đầu tiên để hỗ trợ việc điều khiển trong nút cảm biến, nó được mô tả trong phần (b) hình 2.5. Trong khi phương pháp này làm việc theo nguyên tắc thì việc sắp xếp mô hình thực hiện các quá trình điều khiển đến nút cảm biến lại có một số điểm không phù hợp: các chức năng/lớp giao thức riêng rẽ tương đương với các quá trình riêng rẽ sẽ đòi hỏi chi phí cao khi chuyển từ quá trình này sang quá trình khác. Vấn đề này đặc biệt nghiêm trọng nếu các nhiệm vụ thực hiện nhỏ so với chi phí dùng để chuyển giữa các nhiệm vụ mà điều này thường xảy ra trong các mạng cảm biến. Mỗi quá trình yêu cầu một không gian ngăn xếp trong bộ nhớ, điều này không phù hợp với ràng buộc nghiêm ngặt về bộ nhớ của các nút cảm biến.

Lập trình dựa trên sự kiện

Với các lý do này, mô hình lập trình hơi khác một chút dường như thích hợp hơn. Ý tưởng này dùng đặc tính tác động trở lại của nút WSN và tích hợp nó trong thiết kế hệ điều hành. Hệ thống cần đợi sự kiện xảy ra trong khi sự kiện có thể là dữ liệu hữu dụng từ cảm biến, là gói tin đến hoặc sự kết thúc của bộ định thời. Sự kiện như vậy sau đó được xử lý bởi một chuỗi lệnh ngắn để chỉ lưu giữ sự thật là sự kiện đã xảy ra và lưu giữ thông tin cần thiết, ví dụ như gói tin đến hoặc giá trị của cảm biến. Quá trình xử lý thực tế thông tin này không được thực hiện trong các thủ tục bộ xử lý sự kiện này mà nó được tách từ sự xuất hiện thực tế của các sự kiện. Mô hình **lập trình dựa trên sự kiện** được biểu diễn trên hình 2.6.



Hình 2.6 Mô hình lập trình dựa trên sự kiện

Bộ xử lý sự kiện như vậy có thể ngắt quá trình xử lý của bất kỳ mã thông thường nào nếu nó rất đơn giản và ngắn, nó có thể được yêu cầu để kết thúc nhanh trong tất cả các tình huống mà không làm xáo trộn mã khác một cách quá đáng. Các bộ xử lý sự kiện không thể ngắt lẫn nhau nhưng nó dễ dàng thực hiện liên tiếp nhau.

Mô hình lập trình dựa trên sự kiện phân biệt hai tình huống khác nhau: một cho các bộ xử lý sự kiện thời gian tới hạn, nơi mà quá trình thực hiện không thể ngắt và một cho quá trình xử lý mã thông thường, nó chỉ được khởi động bởi các bộ xử lý sự kiện.

Li so sánh đặc tính của mô hình lập trình dựa trên quá trình và dựa trên sự kiện trên cùng một phần cứng và nhận thấy rằng đặc tính được cải thiện theo thừa số 8, các yêu cầu bộ nhớ lệnh/dữ liệu giảm theo các thừa số 2 và 30, công suất tiêu thụ giảm theo thừa số 12.

Các giao diện với hệ điều hành

Nếu mô hình lập trình quy định không được chấp nhận trên thực tế do hệ điều hành thì cần phải chỉ rõ một số giao diện để trạng thái bên trong của hệ thống trao đổi với bên ngoài.

Khi phân biệt rạch ròi giữa ngăn xếp giao thức và các chương trình ứng dụng trong các WSN thì giao diện như vậy phải sử dụng được từ các hệ thống xử lý giao thức và nó phải cho phép các hệ thống này truy cập lẫn nhau. Giao diện này cũng liên kết một cách chặt chẽ với kiến trúc của các ngăn xếp giao thức được giới thiệu trong phần trước.

Một **giao diện lập trình ứng dụng** (API - Application Programming Interface) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của hệ điều hành để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn

Thông thường, giao diện lập trình ứng dụng như vậy bao gồm giao diện chức năng, sự trừu tượng hoá đối tượng và hiện thực chi tiết. Trừu tượng là các tuyến vô tuyến, các nút...; các chức năng có thể bao gồm truy vấn và thao tác trạng thái, gửi và nhận dữ liệu, truy cập phần cứng (các cảm biến, cơ cấu chấp hành, bộ thu phát), cài đặt các điều khoản như các thoả hiệp về năng lượng/ chất lượng.

2.3.3 Cấu trúc hệ điều hành và ngăn xếp giao thức

Phương pháp truyền thống để xây dựng giao thức liên lạc là sử dụng lớp: các giao thức riêng rẽ được xếp lên nhau, mỗi lớp chỉ sử dụng các chức năng của lớp ngay bên dưới. Phương pháp xếp lớp có hiệu quả cao khi quản lý được bộ ngăn xếp giao thức. Tuy nhiên đối với WSN lại không rõ ràng là phương pháp xếp lớp có đáp ứng được yêu cầu hay không.

Ví dụ: xem xét việc sử dụng thông tin về cường độ tín hiệu nhận được từ đối tác liên lạc. Thông tin lớp vật lý có thể dùng để trợ giúp các giao thức trong mạng ra quyết định về thay đổi tuyến (tín hiệu yếu hơn nếu nút dịch chuyển ra xa và có lẽ không được sử dụng trong bước tiếp theo), tính toán thông tin vị trí bằng cách ước lượng khoảng cách từ cường độ tín hiệu hoặc để trợ giúp giao thức lớp liên kết trong các kênh thích nghi hoặc các sơ đồ FEC/ARQ lai. Do đó, một nguồn thông tin đơn có thể được dùng để kết hợp một cách gián tiếp ưu điểm của các giao thức khác nhau với nguồn thông tin này.

Sự trao đổi thông tin liên lớp (cross-layer information exchange) là cách làm giảm bớt những quy định cứng nhắc của phương pháp xếp lớp. Trong các mạng truyền thông, các mối quan tâm về hiệu quả, nhu cầu hỗ trợ các giao thức mạng hữu tuyến trong các hệ thống vô

tuyến (ví dụ TCP qua vô tuyến), nhu cầu di chuyển chức năng vào đường trục mà không phụ thuộc vào các mô tả của mô hình điểm tới điểm của Internet... rất được chú ý.

Tách khỏi kiến trúc phân lớp, xu hướng thông dụng hiện nay là sử dụng mô hình thành phần. Các lớp đơn khối tương đối lớn được chia thành những thành phần, module nhỏ, độc lập. Các thành phần này chỉ đáp ứng một chức năng xác định, ví dụ như tính toán **kiểm tra tính dư tuần hoàn** (CRC - Cyclic Redundancy Check), và tương tác với nhau qua các giao diện dễ hiểu. Sự khác nhau cơ bản so với kiến trúc phân lớp là các tương tác này không bị hạn chế với các lân cận trung gian trong mối quan hệ trên/dưới nhưng có thể đối với các thành phần khác.

Mô hình thành phần này không chỉ giải quyết một số vấn đề kiến trúc với các ngăn xếp giao thức mà nó còn phù hợp một cách tự nhiên với phương pháp dựa trên sự kiện để lập trình các nút cảm biến vô tuyến.

2.3.4 Năng lượng động và quản lý năng lượng

Chuyển các thành phần riêng rẽ sang các trạng thái ngủ khác nhau hay giảm hiệu suất của chúng bằng cách giảm tần số và nguồn cấp, lựa chọn điều chế và các mã chuyên dụng là các ví dụ nổi bật được giới thiệu trong phần 2.2 để cải thiện hiệu quả năng lượng. Để điều khiển các khả năng này, các quyết định phải được thực hiện bằng hệ điều hành, ngăn xếp giao thức hoặc bằng ứng dụng khi chuyển giữa các trạng thái này. **Quản lý năng lượng động** (DPM - Dynamic Power Management) trên mức hệ thống là vấn đề tiếp theo. Một trong các yếu tố phức tạp của DPM là năng lượng và thời gian yêu cầu cho việc chuyển giữa hai trạng thái của thành phần. Nếu các yếu tố này được bỏ qua thì rõ ràng đây là cách tối ưu để luôn luôn và ngay lập tức chuyển chế độ với mức tiêu thụ công suất nhỏ nhất có thể. Nếu không nằm trong trường hợp này thì yêu cầu các thuật toán tiên tiến hơn, tốc độ cập nhật các quyết định quản lý năng lượng, phân bổ xác suất của thời gian cho đến khi các sự kiện tiếp theo xảy ra, các tính chất của thuật toán sử dụng đều được đưa vào chi phí.

Thoả hiệp giữa độ chính xác và mức tiêu thụ năng lượng

Hầu hết các công việc được mô tả trên DVS điều khiển đều giả thiết giới hạn cứng cho mỗi nhiệm vụ (nhiệm vụ phải được hoàn thành trong thời gian đã cho, nếu không kết quả sẽ không được sử dụng). Trong các WSN, giả thiết như vậy thường không phù hợp. Hơn nữa, các nhiệm vụ đều có thể tính được độ chính xác cao hay thấp. Độ chính xác đạt được bằng các nhiệm vụ như vậy sẽ được dùng để thoả hiệp với các tiêu chí khác. Trong WSN, thoả hiệp đương nhiên là năng lượng được yêu cầu để tính toán nhiệm vụ. Câu hỏi được đặt ra ở đây là làm thế nào để đầu tư tốt nhất lượng năng lượng sẵn có cho nhiệm vụ đã cho. *Đánh đổi kết quả không chính xác để có mức tiêu thụ năng lượng thấp hơn là đặc điểm đặc trưng của các WSN.*

Một số phương pháp khai thác các thoả hiệp này đã được đưa ra nhưng hầu hết đều trong các hệ thống đa phương tiện. Sinhra đã giới thiệu thoả hiệp năng lượng – chất lượng để thiết kế thuật toán, đặc biệt là cho các mục đích xử lý tín hiệu (lọc, các biến đổi trong miền tần số, sự phân loại). Ý tưởng để biến đổi thuật toán là xấp xỉ kết quả cuối cùng một cách nhanh chóng và giữ tính toán miễn là năng lượng đủ để dùng.

Một ví dụ đơn giản là tính đa thức:

$$f(x) = \sum_{i=0}^N k_i x^i \quad (0.10)$$

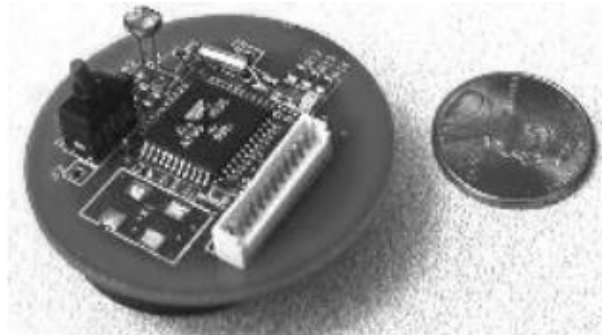
Phụ thuộc vào liệu $x < 1$ hay $x \geq 1$ mà các tính toán sẽ bắt đầu với các số hạng bậc thấp hay bậc cao để có xấp xỉ có thể tốt nhất trong trường hợp phải ngừng tính vì tốn quá nhiều năng lượng cho nó. Đặc trưng của các thuật toán này (cơ bản hay đã bị biến đổi) được nghiên cứu bằng cách sử dụng thông số năng lượng – chất lượng, nó biểu thị qua kết quả là chất lượng đạt được (đã được chuẩn hoá) ứng với bao nhiêu năng lượng (đã được chuẩn hoá).

2.4 Một số ví dụ của nút cảm biến

Có khá nhiều nút cảm biến trên thực tế có thể sử dụng cho việc nghiên cứu và phát triển mạng cảm biến không dây. Phụ thuộc vào từng ứng dụng, nó sẽ thực hiện các yêu cầu tương đối khác nhau về tuổi thọ của pin, độ bền vững cơ học của vỏ bọc của nút, kích thước... Một số ví dụ sẽ chỉ ra đặc điểm nổi bật của các phương pháp và tổng quan về sự phát triển hiện nay.

2.4.1 Họ Mica Mote

Bắt đầu từ cuối những năm 90, toàn bộ họ nút được lấy ra từ các dự án nghiên cứu của trường đại học California ở Berkeley, một phần cộng tác với Intel trong nhiều năm. Chúng thường được gọi là Mica Mote với các phiên bản khác nhau (Mica, Mica2, Mica2Dot) đã được thiết kế. Hình 2.7 là một ví dụ gần đây về sơ đồ của nút như vậy.



Hình 2.7. Nút Mica

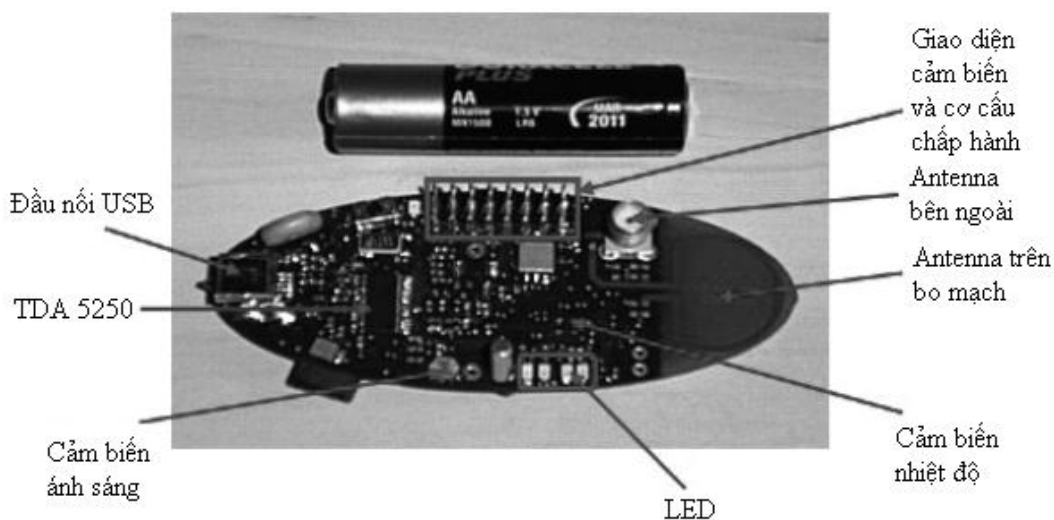
Bảng mạch được đặc trưng bởi bộ vi điều khiển thuộc họ Atmel, modem vô tuyến đơn giản (thường là TR 1000 từ RFM) và các kết nối khác nhau với bên ngoài. Thêm vào đó, nó có thể kết nối với các bảng mạch cảm biến, ví dụ như các cảm biến độ ẩm hay khí áp, nó cho phép một dải lớn các ứng dụng và thực nghiệm. Ngoài ra còn có các hàng rào bao quanh chuyên dụng để sử dụng trong các môi trường khó khăn như theo dõi thói quen của loài chim. Các cảm biến được nối với bộ điều khiển qua bus I2C hay SPI, phụ thuộc vào từng phiên bản.

Các nút MEDUSA – II chia sẻ các thành phần cơ bản và khá giống nhau trong thiết kế.

2.4.2 Các nút EYES

Các nút được phát triển bởi Infineon ở Liên minh Châu Âu – dự án “Các mạng cảm biến có hiệu quả năng lượng” EYES (Energy- efficient Sensor Networks) – là một ví dụ khác về các nút cảm biến điển hình (hình 2.8)

Nó được trang bị bộ vi điều khiển MSP 430 của Texas Instrument, modem vô tuyến Infineon TDA 5250 với bộ lọc SAW và điều khiển công suất truyền. Modem vô tuyến cũng báo cáo cường độ tín hiệu đo được đến bộ điều khiển. Nút có giao diện USB đến PC và có khả năng mắc thêm các cảm biến/cơ cấu chấp hành.



Hình 2.8 Nút cảm biến EYES

2.4.3 Btnode

Btnode được phát triển từ một số dự án nghiên cứu ở ETH Zürich (hình 2.9). Chúng được đặc trưng bởi bộ vi điều khiển 128L của Atmel ATmega, RAM 64 ÷ 180 kB và bộ nhớ FLASH 128 kB. Không giống như hầu hết các nút cảm biến khác (nhưng tương tự như một số nút được phát triển bởi Intel), chúng sử dụng Bluetooth như là kỹ thuật vô tuyến khi kết hợp với Chipcon CC1000, hoạt động trong khoảng tần số từ 433 đến 915 MHz.



Hình 2.9 Btnode

2.5 Kết luận

Chương này giới thiệu các điều kiện cần thiết của phần cứng để xây dựng các mạng cảm biến không dây – nút. Chương đã trình bày các cách chủ yếu để xây dựng các nút đó,

một số đặc trưng và tiêu thụ năng lượng của các thành phần chính (chủ yếu là bộ điều khiển, thiết bị liên lạc và các cảm biến). Điều đó có được dựa trên giả thiết là các nút cảm biến không dây có hai thành phần riêng biệt: một phần chú ý đến tính liên tục, nó có thể phát hiện và thu thập sự kiện, tiêu thụ một mức năng lượng nhỏ, thậm chí là bỏ qua. Phần thứ hai thực hiện xử lý và liên lạc thực tế, nó có mức tiêu thụ năng lượng cao hơn và không thể bỏ qua, vì vậy nó hoạt động với hệ số lấp đầy thấp. Việc chia chức năng sẽ hợp lý hơn nếu dựa trên các tính chất của phần cứng và được hỗ trợ bởi hệ điều hành.

Nhìn vào một lượng lớn các thành phần đã được chọn, chúng đều có ưu điểm và nhược điểm đặc trưng. Sẽ không ngạc nhiên khi không có nút cảm biến không dây hoàn hảo. Các yêu cầu ứng dụng khác nhau sẽ yêu cầu các thoả hiệp khác nhau, và do đó, các kiến trúc khác nhau sẽ được sử dụng. Có mạng cảm biến sử dụng sự kết hợp giữa các loại nút khác nhau để thực hiện nhiệm vụ của nó, ví dụ như các nút có khả năng tính toán lớn hơn/nhỏ hơn, các loại liên lạc vô tuyến khác nhau hay các kích thước pin khác nhau. Vấn đề đặt ra ở đây là thiết kế mạng cảm biến không dây như thế nào bằng cách khai thác sự kết hợp này trong phần cứng để ấn định các nhiệm vụ khác nhau và để cho các nút có sự hoà hợp tốt nhất.

Trong khi rất nhiều mạng mô tả ở đây vẫn đang được nghiên cứu hoặc nó vẫn ở trạng thái nguyên mẫu thì khả năng nổi bật của các nút cảm biến trong tương lai về các thoả hiệp giữa liên lạc, tính toán, lưu giữ và tiêu thụ năng lượng đã khá rõ ràng. Các con số xác thực vẫn tiếp tục thay đổi nhưng không giống như những thoả hiệp đang có. Ví dụ như giữa năng lượng được yêu cầu để liên lạc và tính toán vẫn sẽ thay đổi một cách đột ngột trong thời gian tới. Các thoả hiệp cơ bản này tạo nền tảng cho việc xây dựng các chức năng của mạng, hướng tới các yêu cầu chuyên dụng của các ứng dụng mạng cảm biến không dây.

CÂU HỎI ÔN TẬP CHƯƠNG 2

1. Các yêu cầu phân cứng trong kiến trúc nút đơn
2. Đặc điểm các bộ điều khiển trong nút cảm biến
3. Các đặc điểm bộ thu phát trong nút cảm biến
4. Kiến trúc bộ thu phát trong nút cảm biến
5. Nêu các đặc điểm các loại cảm biến thường dùng trong nút
6. Đặc điểm nguồn cung cấp trong nút cảm biến
7. Tiêu thụ năng lượng trong bộ thu phát của nút cảm biến?
8. Mô hình lập trình trong nút cảm biến

CHƯƠNG 3. KIẾN TRÚC MẠNG

Mục đích của chương

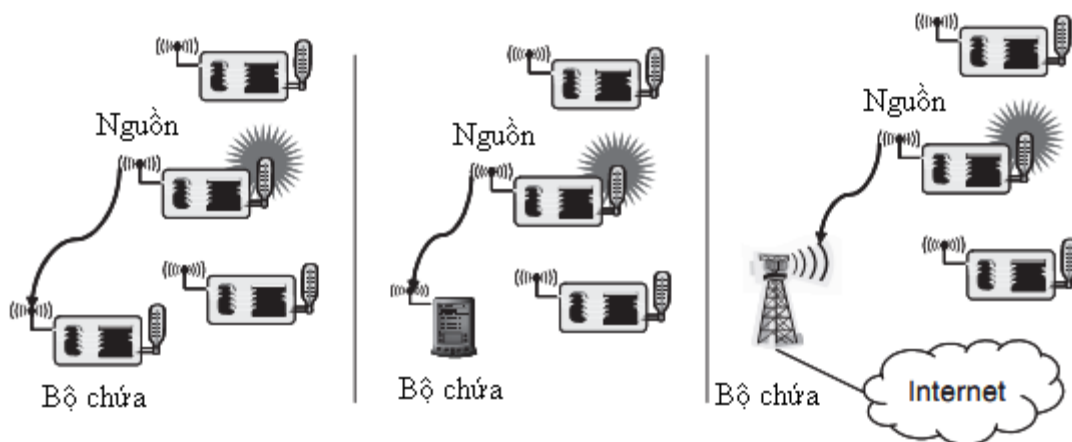
Chương này giới thiệu các nguyên tắc cơ bản của các nút cảm biến riêng biệt trong mạng cảm biến không dây. Trên cơ sở các ứng dụng được giới thiệu trong chương 1, chương này sẽ trình bày cụ thể hơn và các phương pháp tối ưu hóa kết quả. Trên cơ sở đó, một số nguyên tắc thiết kế các giao thức mạng trong các mạng không dây sẽ được giới thiệu đến bạn đọc. Những nguyên tắc này và các cơ chế giao thức tạo nên sự khác biệt cốt lõi của WSN so với các loại mạng khác. Để sử dụng các tính năng của WSN cần phải có một giao diện dịch vụ phù hợp, như là sự tích hợp của các WSN thành một mạng lớn hơn.

Kiến trúc của mạng cảm biến không dây có được theo nhiều cách khác nhau. Trước đây, rất nhiều việc liên quan đến WSN đã được thực hiện trong mạng ad hoc di động tự tổ chức. Khi các mạng này sử dụng cho nhiều mục đích khác nhau, chúng sẽ phải chia quyền ưu tiên.

3.1 Mạng cảm biến

3.1.1 Nguồn và bộ chứa đơn

Phần 1.3 giới thiệu một số mô hình tương tác điển hình của WSN - phát hiện sự kiện, đo định kỳ, chức năng xấp xỉ và phát hiện sườn, hoặc theo dõi - nó cũng được biết đến dưới định nghĩa "nguồn" và "bộ chứa".



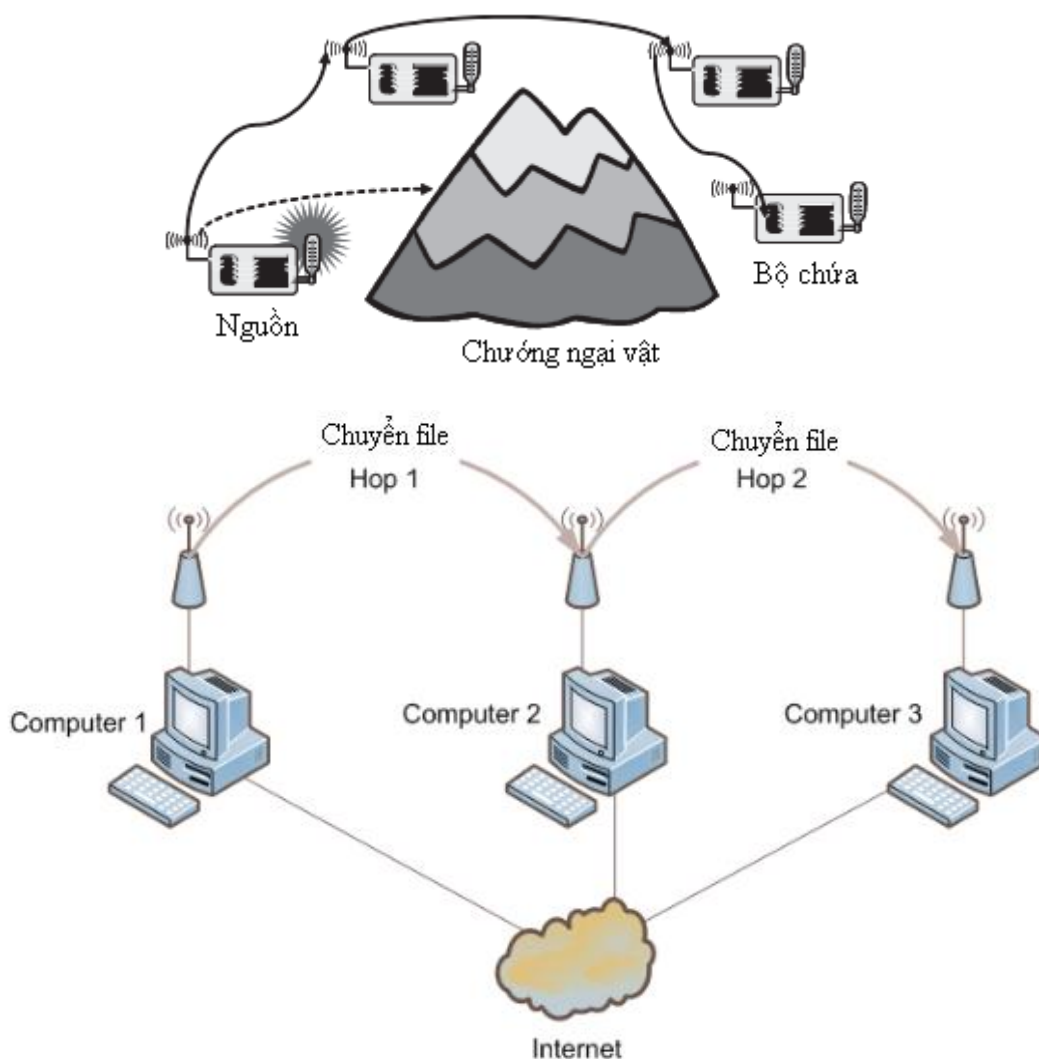
Hình 3.1 Ba loại bộ chứa trong một mạng cảm biến một bước nhảy đơn giản

Nguồn là nơi cung cấp thông tin trong mạng và thường là một nút cảm biến, nó cũng có thể là một nút chấp hành cung cấp thông tin phản hồi về trạng thái hoạt động.

Bộ chứa lưu thông tin được yêu cầu. Có ba lựa chọn cơ bản cho một bộ chứa: nó có thể thuộc mạng cảm biến; thuộc nút cảm biến/ chấp hành hoặc nó không thuộc mạng. Đối với trường hợp thứ hai, bộ chứa có thể là một thiết bị thực, ví dụ, nó là một thiết bị cầm tay hay PDA được sử dụng để tương tác với mạng cảm biến; nó cũng có thể chỉ đơn thuần là cổng vào/ ra cho mạng lớn hơn, như Internet, nơi mà yêu cầu về thông tin đến từ một số nút ở xa và chỉ gián tiếp kết nối đến mạng cảm biến. Các loại bộ chứa chính được minh họa trên hình 3.1, ở đây nguồn và bộ chứa giao tiếp trực tiếp.

3.1.2 Mạng một bước nhảy và đa bước nhảy

Trên cơ sở của thông tin vô tuyến, hạn chế về công suất của thông tin vô tuyến kéo theo hạn chế về khoảng cách giữa người gửi và người nhận. Do khoảng cách này hạn chế nên giao tiếp trực tiếp, đơn giản giữa nguồn và bộ chứa không phải lúc nào cũng thực hiện được, đặc biệt đối với WSN vì nó hoạt động trong môi trường tồn tại nhiều chướng ngại vật hoặc môi trường sóng vô tuyến có sự suy giảm mạnh (ví dụ như trong các tòa nhà).



Hình 3.2 Mạng đa bước nhảy

Để khắc phục hạn chế về khoảng cách người ta sử dụng các trạm chuyển tiếp. Bằng cách đó, các gói dữ liệu được đưa qua nhiều trạm từ nguồn đến bộ chứa. Khái niệm về mạng đa bước nhảy (minh họa trong hình 3.2) sử dụng rất hiệu quả cho các WSN, trong đó bản thân mỗi nút cảm biến có thể được coi như các nút chuyển tiếp, do đó cần thêm thiết bị bổ sung cho mạng. Tùy thuộc vào ứng dụng cụ thể, khả năng có nút cảm biến trung gian ở đúng vị trí cần thiết là khá cao. Ví dụ, trong một khu vực nhất định được trang bị các nút cảm biến, không phải lúc nào cũng tồn tại tuyến đường đa bước nhảy từ nguồn tới bộ chứa, và cũng không thể đảm bảo tuyến đường đó là đủ ngắn.

Trong khi đa bước nhảy là một giải pháp hiển nhiên để khắc phục vấn đề về khoảng cách lớn hoặc chướng ngại vật thì nó cũng được yêu cầu để nâng cao hiệu quả năng lượng của

thông tin. Trong hầu hết các môi trường, sự suy giảm các tín hiệu vô tuyến có bậc ít nhất là 2 (và thường là lớn hơn), nó tiêu thụ ít năng lượng hơn để sử dụng chuyển tiếp thay vì truyền thông trực tiếp. Với mục tiêu có SNR không đổi ở tất cả các máy thu (giả sử tỉ lệ sai số không đáng kể tại SNR này), năng lượng bức xạ cần thiết để giao tiếp trực tiếp qua khoảng cách d là cd^α (c là hằng số, $\alpha \geq 2$ là hệ số suy hao); sử dụng một chuyển tiếp ở khoảng cách $d/2$ làm giảm năng lượng này xuống $2c(d/2)^\alpha$.

Tính toán này chỉ quan tâm tới năng lượng bức xạ mà không chú ý đến năng lượng tiêu thụ, thực tế là có sự tiêu thụ năng lượng tại các nút chuyển tiếp trung gian. Nếu giả sử rằng các nút chuyển tiếp này thuộc WSN thì khi tính tổng năng lượng tiêu thụ vẫn phải đưa chúng vào phép tính. Rất dễ nhận thấy là sẽ phí năng lượng khi các nút chuyển tiếp trung gian sử dụng cho các khoảng cách d ngắn. Chỉ với khoảng cách d lớn, năng lượng bức xạ mới chiếm ưu thế trong chi phí năng lượng cố định tiêu thụ của bộ phát và bộ thu. Đây là khoảng cách cụ thể, khi mà giao tiếp trực tiếp và đa bước nhảy cân bằng nhau, nó phụ thuộc rất nhiều vào các tham số của thiết bị và môi trường cụ thể. Tuy nhiên, mối quan hệ này thường không được xem xét.

Ở đây, ta chỉ quan tâm đến các mạng đa bước nhảy hoạt động ở trạng thái **lưu giữ** và **chuyển tiếp**. Trong mạng như vậy, nút sẽ nhận gói tin một cách trực tiếp trước khi nó được chuyển đến một nơi nào đó. Một phương pháp mới được sử dụng để khai thác gói tin nhận được ngay cả khi nó bị sai. Ví dụ, khi nhiều nút gửi cùng một gói tin và mỗi chuyển tiếp riêng lẻ không thể nhận được, nhưng tập hợp lại, một nút có thể tái tạo lại các gói dữ liệu một cách đầy đủ. Kỹ thuật **hợp tác chuyển tiếp** như vậy không được xem xét ở đây.

3.1.3 Đa bộ chứa và nguồn

Cho đến nay chỉ minh họa được các mạng có bộ chứa và nguồn đơn. Trong nhiều trường hợp, có thể có nhiều nguồn và /hoặc nhiều bộ chứa. Khó khăn nhất lớn nhất là các nguồn phải gửi tin đến nhiều bộ chứa, khi đó tất cả hoặc một số thông tin sẽ đến được tất cả hoặc một số các bộ chứa. Hình 3.3 minh họa những kết hợp này.

3.1.4 Ba loại chuyển động

Trong các ví dụ trên, các đối tượng đều đứng yên. Nhưng ưu điểm chính của giao tiếp vô tuyến là khả năng hỗ trợ cho các ứng dụng có đối tượng di chuyển. Trong các mạng cảm biến không dây, di chuyển có thể xuất hiện dưới ba hình thức chính: nút di động, bộ chứa di chuyển và đối tượng di chuyển.

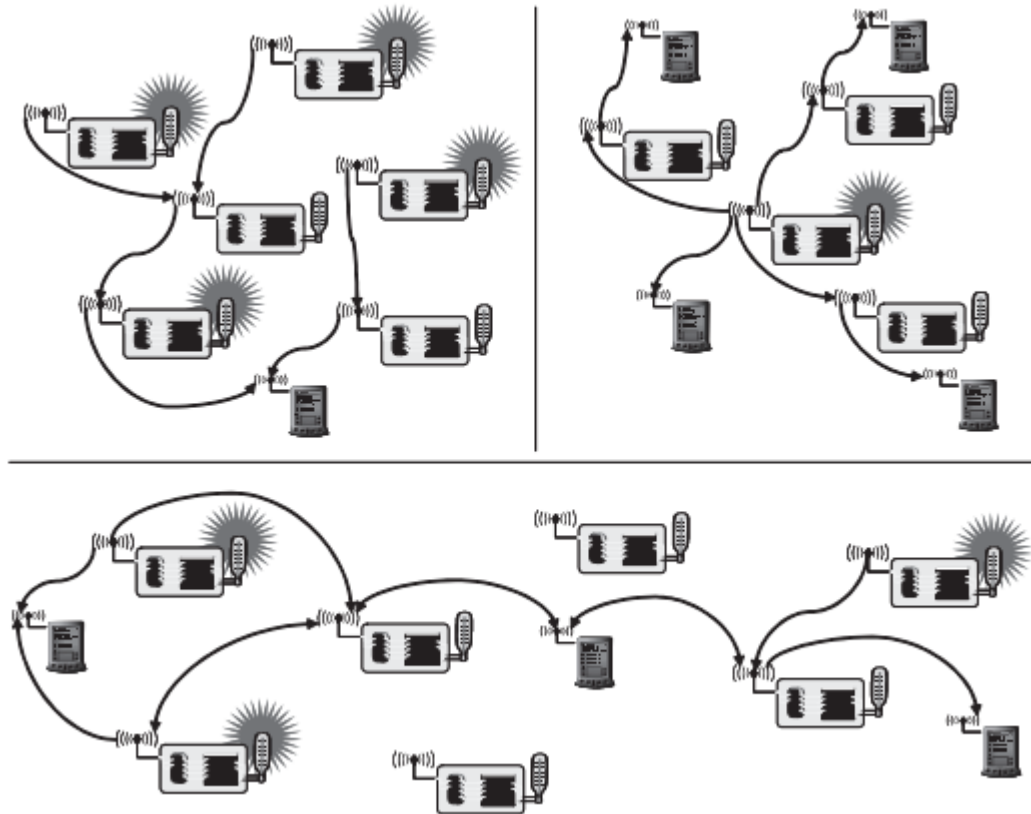
Nút di chuyển: Bản thân các nút cảm biến không dây có thể di chuyển được. Ý nghĩa của tính di chuyển này sẽ khác nhau trong từng ứng dụng cụ thể. Để giám sát môi trường thì không nên dùng các nút di chuyển; trong giám sát chăn nuôi (các nút cảm biến gắn liền với gia súc) thì đây lại là quy luật phổ biến.

Khi sử dụng nút di chuyển, các mạng phải thường xuyên tự tổ chức lại để nó hoạt động đúng chức năng. Rõ ràng là có sự thỏa hiệp giữa tần số và tốc độ của việc di chuyển nút cũng như năng lượng cần để duy trì chức năng mong muốn của mạng.

Bộ chứa di chuyển: Bộ chứa thông tin có thể di chuyển (hình 3.4). Đây có thể được coi là một trường hợp đặc biệt của nút di chuyển. Điểm khác biệt là bộ chứa thông tin không phải là một phần của của mạng cảm biến. Ví dụ, người sử dụng yêu cầu thông tin qua PDA khi đi

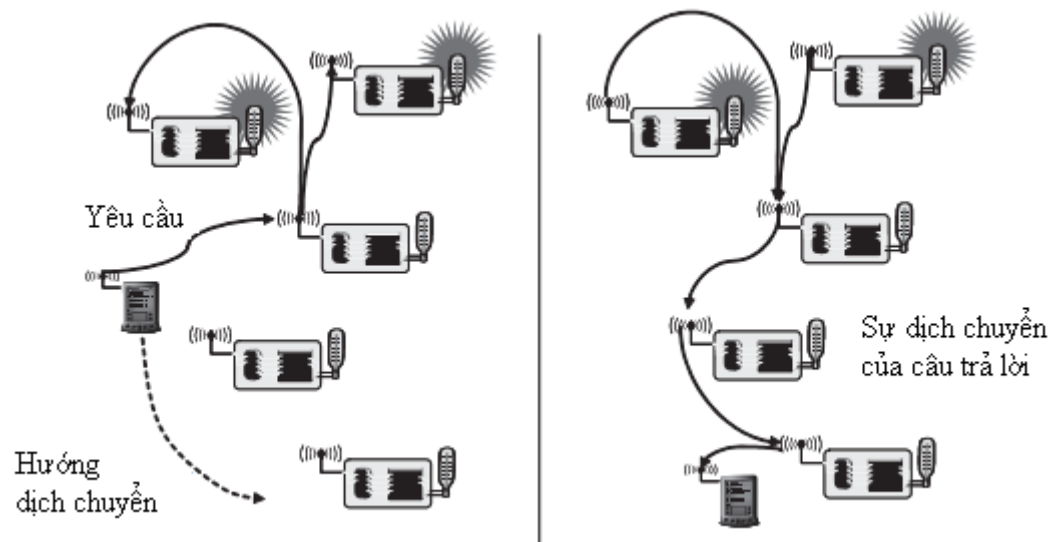
bộ trong một tòa nhà thông minh.

Trong trường hợp đơn giản, người yêu cầu có thể tương tác với WSN tại một thời điểm và hoàn thành các tương tác trước khi di chuyển. Người yêu cầu được phép tương tác với nút bất kỳ hoặc chỉ với các nút cụ thể là lựa chọn thiết kế cho các lớp giao thức thích hợp.



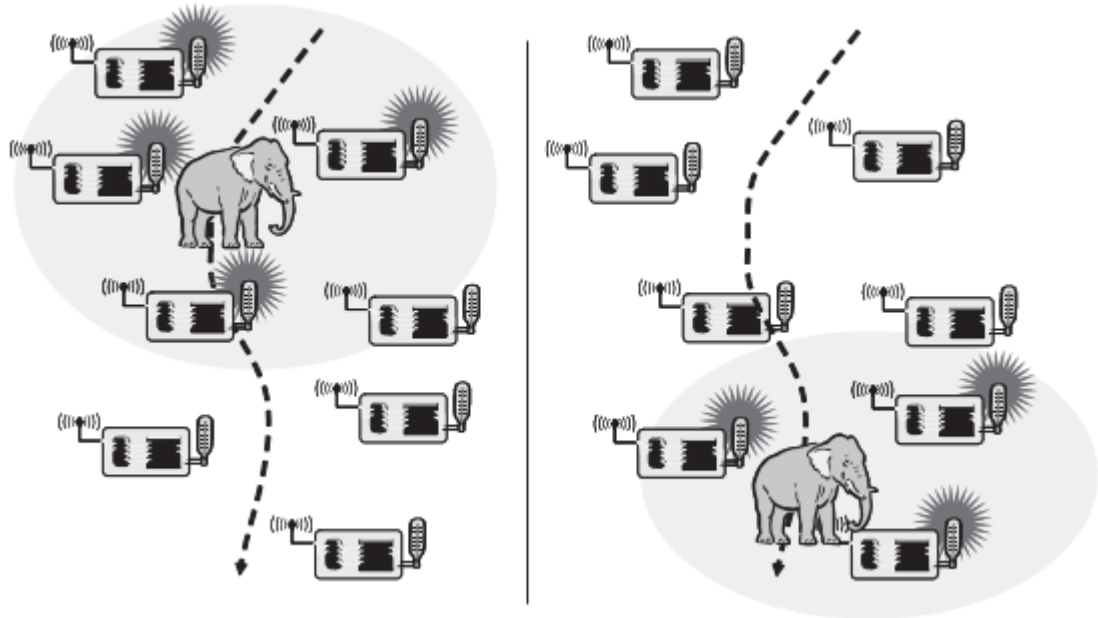
Hình 3.3 Nhiều nguồn/nhiều bộ chứa

Đối tượng di chuyển: Trong các ứng dụng phát hiện sự kiện hoặc các ứng dụng theo dõi, sự kiện hoặc đối tượng được theo dõi có thể di chuyển. Trong trường hợp này, tại một thời điểm, nó thường bị giám sát bởi một số cảm biến.



Hình 3.4 Bộ chứa di động dịch chuyển qua mạng cảm biến

Do đó, cảm biến sẽ “thức” quanh đối tượng, giữ mức hoạt động cao hơn để quan sát đối tượng hiện tại, và sau đó trở về trạng thái “ngủ”. Khi nguồn di chuyển trong mạng, nó cũng di chuyển trong vùng hoạt động của mạng – đây được gọi là mô hình frisbee. Khái niệm này được mô tả trên hình 3.5, nhiệm vụ của nó là phát hiện một con voi đang di chuyển và quan sát khi nó di chuyển xung quanh. Các nút không chủ động phát hiện sự kiện thường có trạng thái ngủ ngắn hơn, trừ khi nó được yêu cầu để truyền thông tin từ vùng hoạt động đến một số bộ chứa ở xa (điều này không được hiển thị trong hình 3.5).



Hình 3.5 Vùng các cảm biến tìm kiếm sự kiện

Các giao thức thông tin của WSN sẽ phải thực hiện hỗ trợ thích hợp cho các hình thức di chuyển. Sự kiện di động là khá phổ biến so với các hình thức trước đây của mạng di động hoặc mạng không dây.

3.2 Các mục tiêu tối ưu và hệ số chất lượng

Thông thường, đối với các ứng dụng khác nhau sẽ có các giải pháp mạng khác nhau. Câu hỏi đặt ra là tối ưu mạng bằng cách nào, làm thế nào để so sánh các giải pháp, làm thế nào để quyết định phương pháp tiếp cận hỗ trợ tốt hơn ứng dụng đã cho, và làm thế nào để biến các mục tiêu tối ưu thành các hệ số chất lượng đo được?

3.2.1 Chất lượng dịch vụ

Mạng cảm biến khác với các mạng thông tin truyền thống chủ yếu là ở loại dịch vụ mà nó cung cấp. Các mạng này chỉ có nhiệm vụ dịch chuyển các bit từ nơi này đến nơi khác. Các yêu cầu sẽ được thêm vào để có được chất lượng dịch vụ (QoS) mong muốn, đặc biệt là trong các ứng dụng đa phương tiện. Giống như các mạng truyền thống, QoS của WSN phụ thuộc nhiều vào ứng dụng. Một số đặc điểm chung xuất hiện ở hầu hết các ứng dụng là:

Khả năng phát hiện/báo cáo sự kiện.

Lỗi phân loại sự kiện: Nếu các sự kiện không chỉ được phát hiện mà còn được phân loại thì lỗi phân loại sự kiện phải rất nhỏ.

Trễ phát hiện sự kiện: Đó là trễ từ lúc phát hiện sự kiện cho đến khi báo cáo nó với một số hoặc tất cả các bộ chứa có liên quan.

Báo cáo lỗi: Trong các ứng dụng yêu cầu phải có báo cáo định kỳ, khả năng các báo cáo chưa được đọc phải rất nhỏ.

Độ chính xác của phép gần đúng: Trong các ứng dụng có chức năng tính gần đúng (ví dụ, tính gần đúng nhiệt độ là hàm theo vị trí của một địa điểm đã cho), sai số tuyệt đối hoặc tương đối trung bình/ lớn nhất phải tương ứng với một hàm.

Độ chính xác của theo dõi: Trong các ứng dụng theo dõi, phải không được để lạc mất đối tượng được theo dõi. Vị trí báo cáo phải gần nhất có thể vị trí thực và lỗi phải nhỏ.

3.2.2 Hiệu quả năng lượng

Phần lớn các cuộc thảo luận đã chỉ ra rằng năng lượng là một nguồn tài nguyên quý giá trong các mạng cảm biến không dây và do đó, hiệu quả năng lượng hiển nhiên là mục tiêu tối ưu. Với một lượng năng lượng tùy ý, hầu hết các số liệu QoS được định nghĩa ở trên có thể được tăng theo ý muốn (độ chính xác theo dõi và xấp xỉ phụ thuộc vào mật độ mạng).

Thuật ngữ **hiệu quả năng lượng** biểu diễn nhiều mặt khác nhau của một hệ thống và cần phân biệt rõ ràng với các hệ số chất lượng đo được trên thực tế. Các khía cạnh thường được quan tâm nhất là:

Năng lượng cho mỗi bit nhận được: Tính trung bình, sử dụng bao nhiêu năng lượng để truyền một bit thông tin từ nguồn đến đích? (đếm tất cả các nguồn tiêu thụ năng lượng ở tất cả các trạm trung gian có thể). Đây là thước đo cho các ứng dụng giám sát định kỳ.

Năng lượng cho mỗi sự kiện (duy nhất): Tương tự như vậy, năng lượng trung bình được sử dụng để báo cáo một sự kiện? Do một sự kiện đôi khi được báo cáo từ các nguồn khác nhau, nên thường chuẩn hóa số liệu này chỉ đối với các sự kiện có tính duy nhất (thông tin dự phòng về một sự kiện đã biết sẽ không cung cấp thêm thông tin).

Các thỏa hiệp về độ trễ/ năng lượng: Một số ứng dụng quan tâm đến các sự kiện có tính "khẩn cấp" nên cần phải cung cấp thêm năng lượng để báo cáo ngay các hiện tượng như vậy. Ở đây, sự cân nhắc giữa độ trễ và tổng chi phí năng lượng là vấn đề rất được quan tâm.

Tuổi thọ của mạng: đó là thời gian mạng hoạt động hay thời gian để mạng hoàn thành nhiệm vụ của nó (ứng với mức năng lượng mà nó lưu trữ). Có thể định nghĩa tuổi thọ của mạng theo các cách khác nhau:

Thời gian đến khi nút đầu tiên chết: Khi nút đầu tiên trong mạng hết năng lượng, hỏng hay ngừng hoạt động.

Nửa tuổi thọ của mạng: Khi 50% số nút hết năng lượng và dừng hoạt động.

Thời gian phân chia: Khi lần đầu tiên mạng được chia thành hai phần không nối với nhau. Điều này xảy ra ngay khi nút đầu tiên chết (nếu nút nằm ở vị trí chủ chốt) hoặc rất lâu sau khi điều này xảy ra nếu mạng có cấu hình bền vững.

Thời gian mất vùng bao phủ: Thông thường, với các mạng triển khai dư thì các cảm biến có thể quan sát một vùng thay vì chỉ quan sát một điểm tại vị trí đặt nút. Như vậy, mỗi

nút trong vùng đó được quan sát bởi nhiều nút cảm biến. Hệ số chất lượng cần quan tâm ở đây sẽ là thời điểm khi lần đầu tiên một vị trí bất kỳ trong khu vực triển khai không còn nằm trong vùng quan sát của một nút nào. Nếu có k nút cùng quan sát một vị trí (ví dụ như trong các ứng dụng theo dõi) thì định nghĩa thời gian mất vùng bao phủ sẽ là thời điểm lần đầu tiên một nút bất kỳ trong vùng triển khai không còn bao phủ bởi các nút cảm biến k khác nhau.

Tuổi thọ của mạng mô phỏng thực sự là một vấn đề khó thống kê. Rõ ràng là thời gian này càng dài càng tốt. Tất cả những số liệu trên chỉ có thể được đánh giá một cách rõ ràng trên giả định về các đặc tính tiêu thụ năng lượng của nút đã cho, về tải "thực tế" của mạng và cả về hành vi của kênh vô tuyến.

3.2.3 Khả năng mở rộng

Khả năng duy trì các tính năng kỹ thuật mà không phụ thuộc vào kích thước của mạng được gọi là khả năng mở rộng. Với một WSN gồm hàng ngàn nút, khả năng mở rộng là một yêu cầu rõ ràng không thể thiếu. Khả năng mở rộng phục vụ cho các cấu trúc đòi hỏi tính nhất quán, ví dụ như địa chỉ hay bảng định tuyến. Do đó, cần phải hạn chế thực thi thông tin này, kèm theo đó là hạn chế tài nguyên của các nút cảm biến, đặc biệt chú ý đến bộ nhớ.

Khả năng mở rộng có ảnh hưởng trực tiếp đến việc thiết kế giao thức. Các kiến trúc và giao thức nên hỗ trợ khả năng mở rộng hơn là cố gắng để thực hiện nó. Khả năng mở rộng có hiệu quả cao hơn với các ứng dụng chỉ có vài chục nút cảm biến thay vì có hàng ngàn nút.

3.2.4 Tính bền vững

Liên quan đến QoS và các yêu cầu về khả năng mở rộng, mạng cảm biến không dây cũng phải có tính bền vững thích hợp. Mạng không thể bị lỗi chỉ vì một số nút năng lượng, hoặc vì thay đổi môi trường, hoặc vì đứt đường kết nối vô tuyến giữa hai nút (lỗi này có thể được khắc phục bằng cách tìm các tuyến đường khác). Đánh giá chính xác tính bền vững của mạng trên thực tế là một vấn đề khó và nó chủ yếu phụ thuộc vào các mô hình lỗi cho cả hai nút và các tuyến giao tiếp.

3.3 Nguyên tắc thiết kế WSN

QoS, hiệu quả năng lượng và khả năng mở rộng là những tiêu chí quan trọng khi thiết kế mạng cảm biến không dây, nhưng bản thân những mục tiêu này không cung cấp nhiều gợi ý về cách cấu trúc một mạng cảm biến. Phần tiếp theo sẽ giới thiệu một số nguyên tắc cơ bản có thể có ích khi thiết kế giao thức mạng.

3.3.1 Tổ chức phân bố

Khả năng mở rộng và tính bền vững là hai mục tiêu tối ưu, và ngoài ra để thỏa mãn một số mục đích khác nữa thì mạng phải được tổ chức dưới dạng phân bố. Điều đó có nghĩa là không có thực thể tập trung phụ trách, ví dụ như giám sát truy cập môi trường hay ra các quyết định định tuyến, nó tương tự như nhiệm vụ thực hiện bởi một trạm gốc trong mạng di động. Nhược điểm của phương pháp tập trung là dễ thất bại và khó thực hiện trong mạng vô tuyến, nơi mà các đối tượng chỉ có một dải giao tiếp hạn chế. Thay vào đó, các nút của WSN phải hợp tác với nhau trong mạng, sử dụng các giao thức và thuật toán phân bố. Tự tổ chức là một thuật ngữ thường được dùng cho nguyên tắc này.

Khi tổ chức một mạng theo kiểu phân bố, cần phải nhận thức được các hạn chế của phương pháp này. Trong nhiều trường hợp, phương pháp tập trung có thể cho các giải pháp tốt hơn hoặc yêu cầu ít tài nguyên hơn (đặc biệt là năng lượng).

3.3.2 Xử lý trong mạng

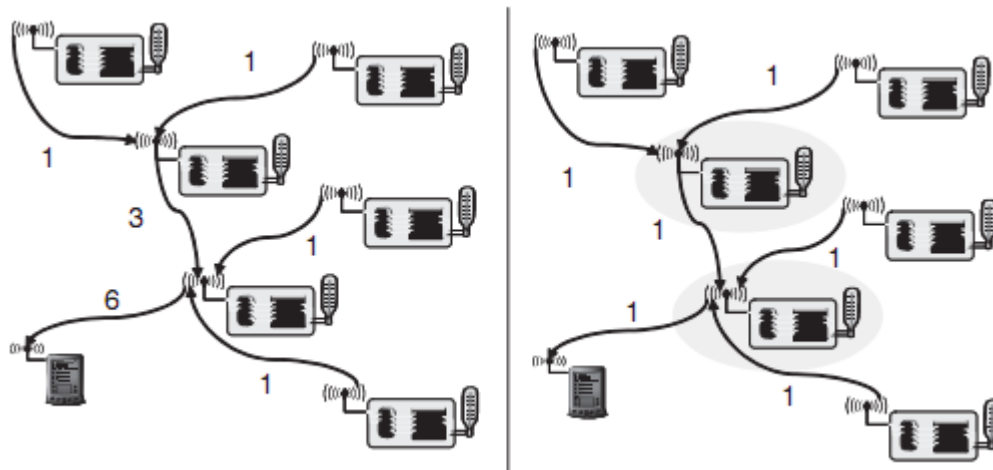
Khi tổ chức mạng theo kiểu phân bố, các nút trong mạng không chỉ truyền các gói tin mà còn phải kích hoạt các chương trình ứng dụng, chúng cũng thực hiện việc ra quyết định sự hoạt động của mạng. Đây là một hình thức cụ thể của xử lý thông tin xảy ra trong mạng, nhưng bị giới hạn thông tin về chính bản thân mạng. Có thể mở rộng khái niệm này bằng cách lấy dữ liệu cụ thể sẽ truyền qua mạng đưa vào quá trình xử lý thông tin này, thực hiện **xử lý trong mạng** là bước đầu tiên trong nguyên tắc thiết kế.

Tập hợp.

Có lẽ tập hợp là kỹ thuật xử lý trong mạng đơn giản nhất. Giả sử một bộ chứa có số liệu từ các phép đo định kỳ của tất cả các cảm biến, nhưng nó chỉ thích hợp để kiểm tra xem giá trị trung bình có thay đổi không, hoặc sự chênh lệch giữa giá trị lớn nhất và nhỏ nhất có quá lớn không? Trong trường hợp này, rõ ràng là không cần phải chuyển tất cả dữ liệu từ các cảm biến đến bộ chứa, cụ thể là nó chỉ cần gửi giá trị trung bình, lớn nhất và nhỏ nhất đến bộ chứa. Trong phần 2.3 đã cho ta thấy việc truyền dữ liệu sẽ đắt hơn việc tính toán dữ liệu, do đó nó cho thấy hiệu quả năng lượng to lớn của phương pháp này. Tên gọi **tập hợp** bắt nguồn từ thực tế là tại các nút trung gian giữa nguồn và bộ chứa, thông tin được tập hợp lại thành một dạng cô đặc, tách biệt với thông tin có được từ các nút ở xa bộ chứa.

Rõ ràng, chức năng tập hợp áp dụng trong các nút trung gian phải đáp ứng một số điều kiện để kết quả có ý nghĩa. Quan trọng nhất, chức năng phải **có thể cấu hình**. Các chức năng như lấy trung bình, đếm, hoặc tối thiểu sẽ có nhiều lợi thế nếu dùng tập hợp. Các chức năng có tính toàn diện như vẽ đường trung bình không hoàn toàn tuân theo tập hợp.

Hình 3.6 minh họa các ý tưởng của tập hợp. Trong nửa trái, một số cảm biến chuyển giá trị đến bộ chứa qua giao tiếp đa bước nhảy. Tổng cộng có 13 tin nhắn được yêu cầu (các số trong hình cho thấy số tin nhắn truyền qua một kết nối đã cho). Khi các nút được đánh dấu để tạo tập hợp, ví dụ, bằng cách tính toán giá trị trung bình (hiển thị trong nửa bên phải của hình), chỉ có 6 tin nhắn là cần thiết. Những khó khăn trong trường hợp này là làm thế nào để xác định vị trí tạo tập hợp và tạo từ những nút nào, bao lâu thì có kết quả và xác định tác động của các gói tin bị mất.



Hình 3.6. Ví dụ tập hợp

Mã hoá nguồn phân bố và nén phân bố.

Thông tin được tập hợp về các giá trị đo được không chỉ để truyền tất cả các bit dữ liệu từ tất cả các nguồn về bộ chứa. Câu hỏi đặt ra là có thể giảm số lượng bit truyền (so với việc chỉ đơn giản là truyền tất cả các bit) nhưng vẫn có thông tin đầy đủ về cảm biến ở bộ chứa?

Câu trả lời của câu hỏi này có liên quan đến vấn đề mã hóa và nén trong mạng thông thường, nơi mà vấn đề mã hóa rất được coi trọng. So với mạng thông thường, mạng cảm biến có một số khác biệt. WSN quan tâm đến việc mã hóa thông tin được cung cấp bởi một số cảm biến. Sơ đồ mã hóa truyền thống thường tập trung vào việc mã hóa, mà điều này tạo ra những quá trình tính toán phức tạp cho các nút cảm biến đơn giản.

Làm thế nào để thông tin được cung cấp bởi nhiều bộ cảm biến giúp đỡ cho quá trình mã hóa? Nếu các cảm biến được kết nối và có thể trao đổi dữ liệu, có thể thực hiện điều này bằng cách sử dụng thuật toán nén tiêu chuẩn, nhưng kết quả không khả quan. Do đó, bắt buộc phải có một số liên kết ngầm giữa hai cảm biến. Trong WSN, các cảm biến này được nhúng trong một môi trường vật lý nên dữ liệu thu được từ các cảm biến lân cận sẽ tương tự nhau, được gọi là tương quan. **Mối tương quan** này có thể được sử dụng để không phải tất cả các dữ liệu thu được tại các cảm biến được truyền đi. Cơ sở lý thuyết cho việc này là định lý của Slepian và Wolf.

Định lý Slepian-Wolf dựa trên ví dụ về khai thác mối tương quan không gian, thường có trong dữ liệu của cảm biến, khi mạng đủ dày đặc. Nó được so sánh với vi phân của hàm quan sát và cấp của sự tương quan giữa dữ liệu ở hai nơi. Tương tự như vậy, **mối tương quan tức thời** có thể được sử dụng trong các giao thức mạng cảm biến.

Xử lý tín hiệu hợp tác và phân bố.

Các phương pháp xử lý trong mạng không thể được dùng để xử lý trong các nút cảm biến, nếu dùng thì cũng chỉ trong các hoạt động đơn giản như tính trung bình hoặc tìm giá trị cực đại. Khi thực hiện tính toán phức tạp đối với một lượng dữ liệu nhất định, vẫn yêu cầu hiệu quả năng lượng cao để tính những hàm này trên các nút cảm biến mặc dù bị hạn chế về công suất xử lý thì số lượng dữ liệu được truyền có thể giảm xuống.

Một ví dụ cho khái niệm này là biến đổi Fourier nhanh (Fast Fourier Transform -FFT).

Phụ thuộc vào địa điểm nhận dữ liệu vào mà có các thuật toán khác nhau để tính toán FFT theo luật phân bố, với các thoả hiệp khác nhau giữa độ phức tạp của tính toán và yêu cầu của thông tin. Về nguyên tắc, đây là thuật toán tương tự như thiết kế cho các máy tính song song. Tuy nhiên, độ trễ của giao tiếp, năng lượng tiêu thụ của giao tiếp và độ phức tạp trong tính toán là các tham số quyết định việc lựa chọn thuật toán.

3.3.3 Tính chính xác và độ tin cậy

Như phần 2.3.4 đã nói, trong trường hợp chỉ có một nút duy nhất, điều làm nên độ tin cậy của kết quả tính toán chính là năng lượng dùng cho nó. Khái niệm này có thể mở rộng từ một nút duy nhất ra cho toàn mạng.

Ví dụ, hãy xem xét một ứng dụng có chức năng xấp xỉ. Rõ ràng, càng nhiều cảm biến tham gia vào quá trình xấp xỉ thì việc lấy mẫu được thực hiện ở càng nhiều điểm và kết quả sẽ tốt hơn. Để có được điều này thì năng lượng được sử dụng càng nhiều. Tương tự là các ứng dụng theo dõi và phát hiện sự kiện của WSN.

Điều đó dẫn đến ứng dụng là bằng cách nào đó xác định mức độ chính xác của kết quả (giả định rằng nó có thể tồn tại với các kết quả gần đúng và không chính xác) và đây là nhiệm vụ của các giao thức giao tiếp để cố gắng ít nhất là đạt được độ chính xác mà vẫn đảm bảo hiệu quả năng lượng. Hơn nữa, nó phải phù hợp với các yêu cầu ứng với trạng thái hiện tại của mạng - có bao nhiêu nút đã bị hỏng, có bao nhiêu năng lượng có thể tìm thấy từ môi trường, các điều kiện hoạt động là gì (các sự kiện quan trọng đã xảy ra gần đây)...

Tuy nhiên, như đã thảo luận trong phần QoS của WSN, sự đa dạng của các ứng dụng WSN chính là thách thức để có một giao diện thống nhất cho các yêu cầu, hãy để các giao thức truyền thông thực hiện việc ra các quyết định này. Đây vẫn là một trong những vấn đề nghiên cứu cốt lõi của WSN.

3.3.4 Tập trung dữ liệu

Dữ liệu địa chỉ.

Trong các mạng thông tin truyền thống, trọng tâm của mối quan hệ giao tiếp thường là cặp đối tượng ngang hàng - người gửi và người nhận dữ liệu. Tuy nhiên, trong mạng cảm biến không dây, ứng dụng không chú trọng quá nhiều vào việc nhận dạng nút cảm biến đặc biệt mà tập trung vào thông tin thực tế về môi trường vật lý, đặc biệt trong trường hợp khi WSN triển khai dư, tức là một hiện tượng có thể được quan sát bởi nhiều nút cảm biến. Ứng dụng cũng không quan tâm một cách chính xác là nút nào đang cung cấp dữ liệu từ môi trường. Thực tế là không cần nhận dạng nút nhưng dữ liệu của các nút lại là trung tâm của sự chú ý. Chính vì vậy, đây được gọi là hệ thống tập trung dữ liệu. Đối với các ứng dụng, điều này có nghĩa là giao diện được đưa ra bởi mạng, nơi mà dữ liệu, chứ không phải các nút, được đánh địa chỉ theo yêu cầu. Tập các nút tham gia vào việc đánh địa chỉ của trung tâm dữ liệu như vậy được xác định bởi lượng dữ liệu mà nó đóng góp vào địa chỉ.

Thực hiện tùy chọn cho trung tâm dữ liệu mạng: Có thể có một số cách để làm cho khái niệm trừu tượng này của mạng trung tâm dữ liệu được rõ ràng hơn. Một trong số đó là tập hợp một số giao tiếp mà có thể sử dụng được bởi một ứng dụng. Sau đây là ba phần quan trọng nhất:

- **Mạng xếp chồng và bảng băm phân bố:** Có một số điểm tương đồng giữa các ứng dụng ngang hàng (peer-to-peer) và WSN: Trong cả hai trường hợp, người sử dụng/ người yêu cầu là chỉ quan tâm đến tìm kiếm và thu thập dữ liệu không có trong mã nguồn của nó; các yêu cầu về dữ liệu và độ khả dụng của nó có thể được tách theo thời gian; cả hai mạng nên có quy mô lớn.

Trong mạng ngang hàng, giải pháp để tìm kiếm dữ liệu một cách có hiệu quả từ một nguồn không rõ thường tạo ra mạng xếp chồng, thực hiện một **bảng băm phân bố** (Distributed Hash Table - DHT). Dữ liệu cần tìm có thể xác định qua một khoá đã cho (hash) và DHT sẽ cung cấp một (hoặc nhiều) nguồn dữ liệu liên quan đến khoá này. Điểm quan trọng là để việc tra cứu nguồn dữ liệu này có thể được thực hiện có hiệu quả đòi hỏi phải thực hiện $O(\log n)$ bước trong đó n là số nút, thậm chí với cả thông tin về nơi cất giữ dữ liệu trong mạng ngang hàng.

Mặc dù có nhiều điểm tương đồng, nhưng ở đây cũng tồn tại một số điểm khác nhau. *Trước hết*, không có sự tương ứng rõ ràng giữa khoá tĩnh của DHT và các yêu cầu động, tham số hoá trong mạng WSN. *Thứ hai*, và quan trọng hơn, các DHT, được tạo ra từ mạng IP, có xu hướng bỏ qua khoảng cách/số bước nhảy giữa hai nút và xem xét các nút lân cận chỉ trên cơ sở thông tin về các khoá lưu trữ trong đó.

- **Công bố / Đăng ký:** Phân biệt điều này được yêu cầu theo thời gian và việc nhận dạng bộ chứa của nút, nó cần hoạt động của việc cung cấp tin chứ không phải là đặc tính đồng bộ của giao thức yêu cầu/ trả lời. Vậy yêu cầu nào có ý nghĩa để thể hiện sự cần thiết của dữ liệu và việc cung cấp chúng?

Hành vi này được thực hiện bởi phương pháp công bố/ đăng ký. Bất kỳ nút nào quan tâm đến loại dữ liệu đã cho đều có thể đăng ký, và bất kỳ nút nào cũng có thể công bố dữ liệu. Sau khi công bố, tất cả các thuê bao dữ liệu này được thông báo về dữ liệu mới. Nếu thuê bao không còn quan tâm, nó có thể bỏ đăng ký từ bất kỳ loại sự kiện nào và sẽ không còn được thông báo về sự kiện đó.

Việc công bố/ đăng ký thông tin có thể được thực hiện theo nhiều cách khác nhau. Một khả năng là sử dụng đối tượng trung tâm, nơi mà việc công bố/ đăng ký xuất hiện với nhau, nhưng điều này rõ ràng không thích hợp cho các WSN. Giải pháp phân phối có hiệu quả nhưng phức tạp hơn.

- **Cơ sở dữ liệu:** Một cách nhìn khác về WSN là xem chúng như là các cơ sở dữ liệu. Quan điểm này rất phù hợp với ý tưởng sử dụng tổ chức trung tâm dữ liệu của các giao thức mạng. Điều này dựa trên đặc điểm của môi trường vật lý là nó được khảo sát bởi một WSN, tương đương với việc tạo ra các câu hỏi cho cơ sở dữ liệu. Để đưa các mạng cảm biến vào trong khuôn khổ của cơ sở dữ liệu có liên quan, thông thường coi các cảm biến như là một bảng ảo cập nhật các hoạt động.

3.3.5 Khai thác thông tin vị trí

Một kỹ thuật hữu dụng khác là khai thác vị trí thông tin trong các giao thức truyền thông mỗi khi thông tin xuất hiện. Từ vị trí của sự kiện, là thông tin quan trọng cho nhiều ứng dụng, phải có các cơ chế xác định vị trí của các nút cảm biến (và có thể cả sự kiện được quan sát).

Thông tin đó có thể đơn giản hóa việc thiết kế và hoạt động của giao thức truyền thông và có thể cải thiện hiệu quả năng lượng đáng kể.

3.3.6 Khai thác các mô hình hoạt động

Các mô hình hoạt động trong mạng cảm biến không dây có xu hướng khác so với các mạng truyền thống. Trong khi sự thật là tốc độ dữ liệu được lấy trung bình trong một thời gian dài rất nhỏ khi có rất ít sự kiện được báo cáo, điều này có thể thay đổi đáng kể khi một sự kiện xảy ra. Khi một sự kiện xảy ra, nó có thể được phát hiện bởi một số lượng lớn các cảm biến, tham gia vào hoạt động của mạng và gây ảnh hưởng đến các sự kiện đã biết. Do đó, giao thức được thiết kế phải xử lý được các lưu lượng chùng bằng việc chuyển đổi giữa các chế độ giữa nghỉ và tích cực cao của mạng.

3.3.7 Giao thức dựa trên phần tử và tối ưu lớp giao nhau

Cuối cùng, ta xét đến những khía cạnh thực hiện các giao thức truyền thông trong WSN. Phần 2.3.3 đã giới thiệu các mô hình dựa trên phần tử, nó đối lập với mô hình dựa trên lớp của giao thức trong WSN. Nhiệm vụ còn lại là việc lựa chọn phần tử sao cho không phải tất cả đều luôn sẵn sàng trên tất cả các nút cảm biến, nhưng nó có thể tạo thành một toolbox cơ bản của các giao thức và thuật toán.

Tất cả các cảm biến không dây đều yêu cầu các giao thức vật lý, MAC và lớp liên kết (ngay cả đối với mạng đơn giản). Một số mạng yêu cầu thêm chức năng định tuyến và lớp truyền tải. Các module trợ giúp như đồng bộ thời gian, giám sát cấu hình mạng hay cục bộ hoá sẽ rất hữu dụng trong quá trình sử dụng. Phía trên các phần tử cơ bản này xây dựng các chức năng trừu tượng. Tiếp theo là tập các phần tử kích hoạt nút cảm biến, thay đổi từ ứng dụng này sang ứng dụng khác.

3.4 Giao diện dịch vụ của WSN

3.4.1 Cấu trúc của giao diện ngăn xếp giao thức/ứng dụng

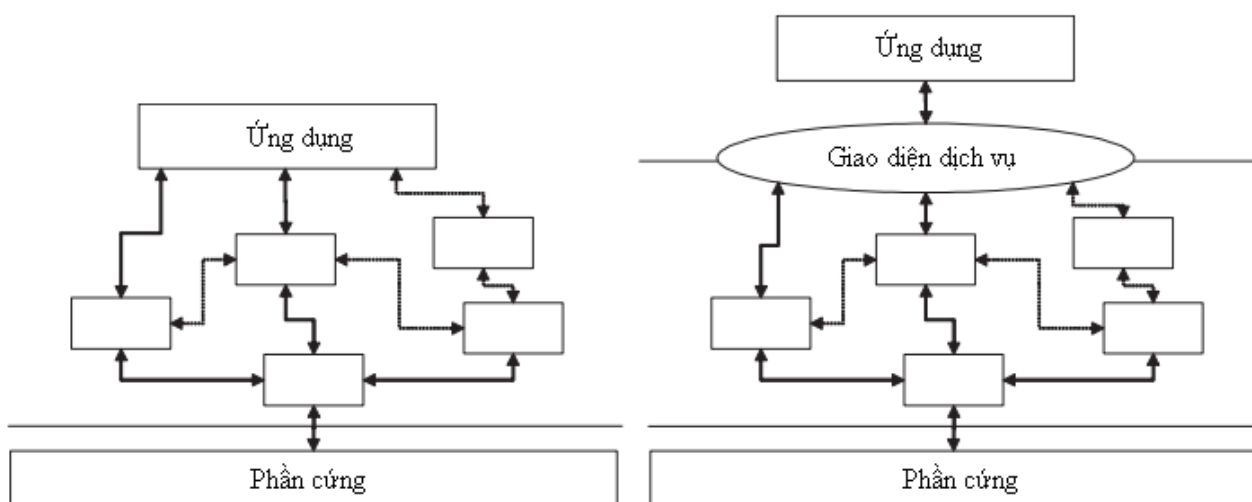
Phần 2.3 đã giới thiệu hệ thống hoạt động dựa trên phần tử và ngăn xếp giao thức chuẩn bị cho một ứng dụng: các phần tử tương tác trực tiếp với nhau thông qua giao diện tồn tại giữa chúng (ví dụ như cấu trúc lệnh/ sự kiện của TinyOS). Ứng dụng có thể sử dụng một số phần tử, được tích hợp ở những nơi khác nhau trong ngăn xếp giao thức. Cách tiếp cận này có một số lợi thế: tinh gọn với cấu trúc giao thức tổng thể, dễ dàng giới thiệu mã ứng dụng cụ thể vào WSN ở các mức khác nhau và không yêu cầu giao diện dịch vụ chuyên dụng.

Nhưng, tính tổng quát và linh hoạt cũng chính là nhược điểm của phương pháp này. Việc cho phép các lập trình viên ứng dụng làm lộn xộn ngăn xếp giao thức nên hệ điều hành phải được giám sát cẩn thận. Trong các mạng truyền thống như Internet, các lập trình viên ứng dụng có thể truy cập vào các dịch vụ của mạng thông qua một giao diện thông thường là ổ cắm. Giao diện này quy định rõ ràng về cách xử lý các kết nối, làm thế nào để gửi và nhận các gói tin, và làm thế nào để hỏi thông tin về trạng thái của mạng. Đây là nhiệm vụ hiển nhiên của các giao diện - trao đổi các gói tin với nhau.

Như vậy, có hai cách là thiết kế ứng dụng như là một phần tử hoặc thiết kế giao diện dịch vụ cho tất cả các phần tử. Hai lựa chọn này được chỉ ra trên hình 3.7. Giao diện dịch vụ

sẽ cho phép nâng cao mức độ trừu tượng hoá cho ứng dụng có thể tương tác với WSN - thay vì phải xác định giá trị để đọc từ các cảm biến riêng lẻ, người ta mong muốn cung cấp một ứng dụng với khả năng thể hiện cảm nhận nhiệm vụ gắn với ý nghĩa của ứng dụng.

Rõ ràng, sự tích hợp chặt chẽ của ứng dụng trong ngăn xếp giao thức sẽ tạo ra tính tối ưu cho nhà lập trình ứng dụng. Mặt khác, nó đòi hỏi phải có nhiều kinh nghiệm hơn khi sử dụng một giao diện dịch vụ chuẩn. Câu hỏi đặt ra ở đây là giá thành, sự suy giảm các đặc tính quan trọng và độ phức tạp của giao diện dịch vụ. Trong thực tế, so với mạng Internet, tính đa dạng và phức tạp về mẫu thông tin trong WSN lớn hơn nhiều. Do đó, giao diện dịch vụ thực sự có ý nghĩa và sẽ phức tạp hơn.



Hình 3.7 Hai lựa chọn giao diện dịch vụ với ngăn xếp giao diện

3.4.2 Các yêu cầu cho giao diện dịch vụ của WSN.

Các chức năng quan trọng nhất mà một giao diện dịch vụ nên có là:

- **Hỗ trợ các yêu cầu/ đáp ứng đơn giản:** lấy giá trị đo từ một số cảm biến hoặc đặt tham số cho một số nút. Đây là một mô hình tương tác đồng bộ và kết quả sẽ có ngay lập tức. Ngoài ra, các đáp ứng có thể yêu cầu được cung cấp định kỳ, hỗ trợ các ứng dụng đo lường định kỳ.

- **Hỗ trợ cho các thông báo sự kiện không đồng bộ:** một nút có thể yêu cầu mạng thông báo khi có một sự kiện xảy ra. Đây là mô hình không đồng bộ trên khía cạnh không tồn tại mối quan hệ giữa thời gian yêu cầu được thực hiện và thời gian thông tin được cung cấp. Hình thức này yêu cầu không đồng bộ nên được đi kèm với khả năng hủy bỏ yêu cầu thông tin.

Mối quan tâm lớn hơn là câu hỏi làm thế nào để tập trung dữ liệu? Một lựa chọn, liên quan chặt chẽ đến phương pháp công bố/ đăng ký được đưa ra trong phần 3.3.4, nó được định nghĩa bởi một số hàm thành viên của một nhóm các nút như:

- **Vị trí** - tất cả các nút đều thuộc một vùng xác định phụ thuộc vào nhóm
- **Giá trị quan sát được** - tất cả các nút có giá trị quan sát được phù hợp với một thuộc

tính đã cho sẽ thuộc về một nhóm. Ví dụ, yêu cầu nhiệt độ đo được lớn hơn 20°C.

Do các yêu cầu của giao diện dịch vụ tương ứng với khái niệm công bố/ đăng ký nên phương pháp này khá tự nhiên, nhưng không vì thế mà nó chỉ phù hợp với các WSN.

- Chức năng xử lý trong mạng được sử dụng. Đối với hoạt động truy cập toàn bộ một nhóm các nút, đặc biệt là khi đọc các giá trị từ nhóm này (hoặc đồng bộ hoặc không đồng bộ), nó cần phải xác định loại xử lý trong mạng nào áp dụng được cho nó. Đặc biệt, khi xử lý thay đổi bản chất của kết quả (ví dụ như hợp nhất dữ liệu) thì nó phải được sự cho phép của ứng dụng đã yêu cầu.

- Liên quan đến chức năng tập hợp là một đặc điểm kỹ thuật về độ chính xác yêu cầu của kết quả. Điều này có thể có được bằng cách chỉ định giới hạn về số thành viên trong nhóm để tạo ra kết quả. Đi kèm với độ chính xác yêu cầu là chi phí năng lượng chấp nhận được.

- Các yêu cầu về tính kịp thời khi cung cấp dữ liệu là một khía cạnh tương tự. Ví dụ, nó có thể cung cấp kết quả một cách nhanh chóng nhưng với chi phí năng lượng cao hoặc chậm nhưng với chi phí năng lượng thấp.

Nói chung, bất kỳ sự thỏa hiệp nào về việc tiêu thụ năng lượng của sự trao đổi các gói dữ liệu cần được thể hiện càng rõ càng tốt.

- Yêu cầu để truy cập vào địa điểm, thời gian, hoặc thông tin trạng thái mạng (ví dụ như dự trữ năng lượng có sẵn trong các nút) thông qua giao diện dịch vụ.

Nó cũng có thể được sử dụng nhằm kết khối thông tin vị trí vào cấp trừu tượng cao hơn để có thể nói về các đối tượng tương ứng với cách nhìn của con người về sự vật.

- Bảo mật cũng là một đặc tính thường xuyên được sử dụng

- Hỗ trợ kết nối giữa các nút khác nhau, toàn bộ mạng lưới hoặc truy cập đơn giản vào các dịch vụ trong một mạng “chưa biết”

3.5 Khái niệm cổng vào/ra

3.5.1 Sự cần thiết của các cổng vào/ra

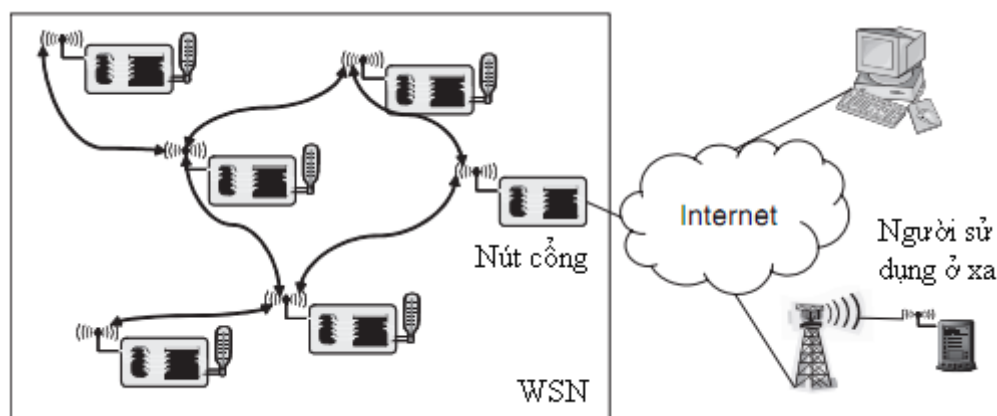
Trên thực tế, mạng cảm biến chỉ quan tâm tới bản thân nó thôi là không đủ, mạng có thể tương tác với các thiết bị thông tin khác, ví dụ. Ví dụ, một người sử dụng được trang bị PDA di chuyển trong vùng phủ sóng của mạng hoặc với một người dùng từ xa, cố gắng để tương tác với các mạng cảm biến qua Internet (ví dụ chuẩn là đọc cảm biến nhiệt độ trong nhà của một người khi đi du lịch và truy cập Internet thông qua kết nối vô tuyến). Hình 3.8 mô tả trường hợp này.

Để thực hiện điều này, trước hết WSN phải có thể trao đổi dữ liệu với một thiết bị di động hoặc với một số loại cổng vào/ra tạo kết nối với Internet. Về bản chất, việc này tương đối đơn giản. Lớp MAC và lớp liên kết - hoặc các thiết bị di động/ cổng vào ra được trang bị một bộ thu phát vô tuyến như được sử dụng trong các WSN, hoặc một số (không phải tất cả) các nút trong WSN hỗ trợ các công nghệ giao tiếp vô tuyến chuẩn như IEEE 802.11. Các tùy chọn có thể gặp thuận lợi, phụ thuộc vào từng ứng dụng và từng trường hợp cụ thể. Thỏa hiệp có thể bao gồm tỷ lệ các nút cảm biến được yêu cầu để phục vụ người dùng di chuyển so với chi

phí để các bộ thu phát của WSN phù hợp với các thiết bị di động như PDA.

Cách đơn giản để thiết kế cổng vào ra là xem nó như một định tuyến đơn giản giữa Internet và mạng cảm biến. Điều này yêu cầu phải sử dụng các giao thức Internet trong mạng cảm biến. Các WSN thường yêu cầu rất nhiều giao thức tối ưu nên một bộ định tuyến đơn giản sẽ không đủ như một cổng vào ra.

Khả năng còn lại là thiết kế nó như là cổng vào ra cấp ứng dụng: trên cơ sở của thông tin cấp ứng dụng, cổng sẽ phải quyết định hành động của mình.

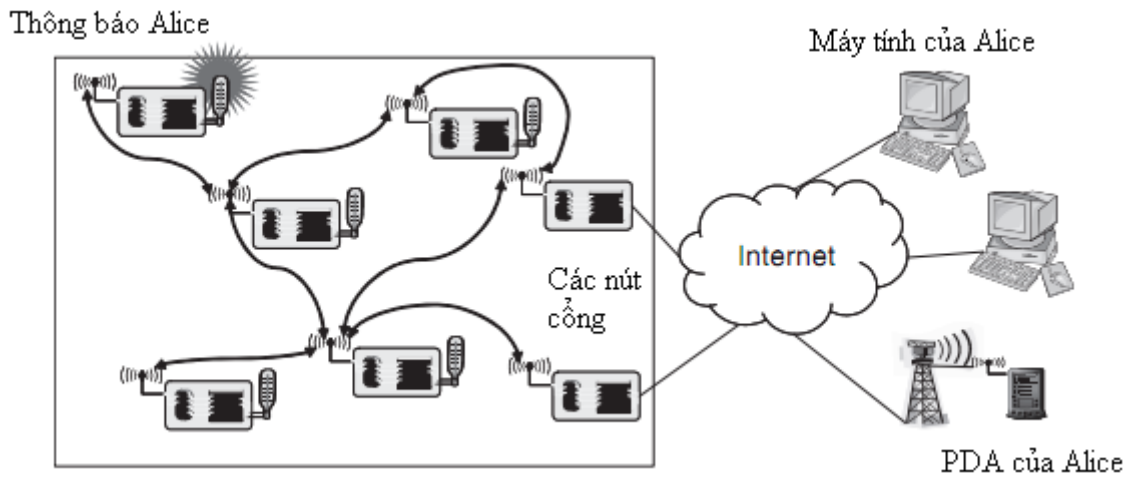


Hình 3.8 WSN với nút cổng vào ra, có thể truy cập đến đối tượng ở xa qua Internet

3.5.2 Giao tiếp WSN với Internet

Giả sử khởi xướng của giao tiếp WSN-Internet nằm trong WSN (Hình 3.9). Ví dụ, nút cảm biến muốn chuyển một tin báo động cho một số máy chủ Internet. Vấn đề đầu tiên để giải quyết là giống như các mạng ad hoc, làm thế nào để tìm thấy cổng vào ra từ bên trong mạng. Về cơ bản, định tuyến đến một nút sẽ được dịch vụ chuyên dụng phát hiện. Nếu một số cổng có thể làm được điều đó thì làm thế nào để lựa chọn giữa chúng? Đặc biệt, nếu không phải tất cả các host đều có thể truy cập Internet thông qua mỗi cổng hoặc một cổng bất kỳ chỉ phù hợp với một máy chủ đích nhất định? Làm thế nào để xử lý cổng, IP có khả năng kết nối mạng, và giao tiếp giữa họ? Lựa chọn ở đây là xây dựng một mạng IP bao phủ ở trên mạng cảm biến.

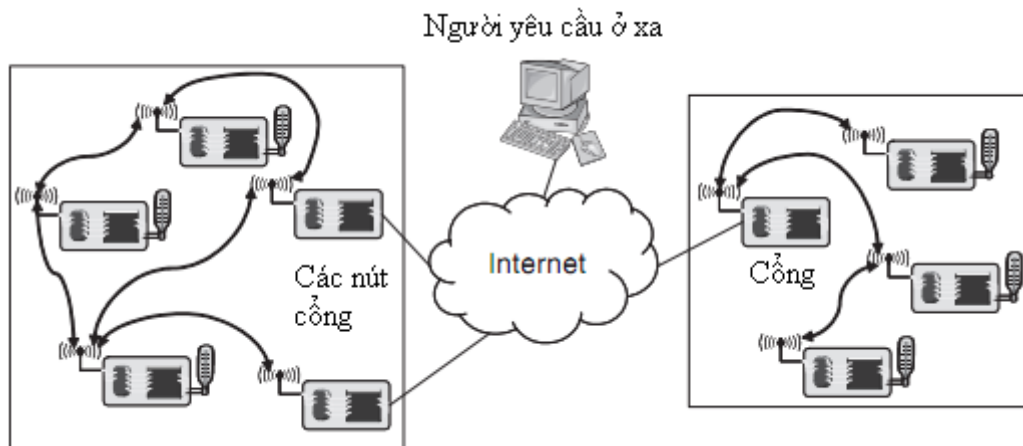
Làm thế nào nút cảm biến phát hiện địa chỉ của máy chủ Internet cần gửi tin, tức là làm thế nào để gửi tin "Thông báo Alice" đến một địa chỉ IP cụ thể? Ngay cả khi các nút cảm biến không cần phải xử lý giao thức IP thì nó cũng phải có đầy đủ thông tin cần thiết (ví dụ như địa chỉ IP và số cổng) trong các gói tin riêng của mình; Cổng vào ra sẽ chuyển chuyển thông tin này vào các gói tin IP. Một câu hỏi tiếp theo là nguồn địa chỉ sử dụng ở đây – cổng vào ra phải có cảm giác về các nhiệm vụ phải thực hiện tương tự như một thiết bị dịch địa chỉ mạng (NAT - Network Address Translation).



Hình 3.9 Giao tiếp WSN – Internet

3.5.3 Giao tiếp Internet-WSN

Điều này xảy ra khi một đối tượng trên Internet đang cố truy cập dịch vụ của WSN (hình 3.10). Điều này là khá đơn giản nếu thiết bị đầu cuối yêu cầu này có thể trực tiếp liên lạc với các WSN, ví dụ, một thiết bị yêu cầu di động được trang bị một bộ thu phát WSN. Trong trường hợp này, thiết bị yêu cầu đầu cuối có thể là một phần trực tiếp của WSN.



Hình 3.10 Giao tiếp Internet-WSN

Trường hợp tổng quát hơn là, thiết bị đầu cuối yêu cầu dịch vụ và không thể ngay lập tức giao tiếp được với nút cảm biến, do đó nó đòi hỏi phải có sự hỗ trợ của cổng vào ra. Câu hỏi đặt ra ở đây là làm thế nào để tìm được một mạng cảm biến ở vị trí mong muốn, và làm thế nào để biết được sự tồn tại của một cổng vào ra? Một khi các thiết bị đầu cuối có được yêu cầu thông tin này, làm thế nào để truy cập vào các dịch vụ thực tế? Giải pháp được đưa ra ở đây là thiết bị đầu cuối thay vì gửi định dạng đúng yêu cầu của cổng này thì nó lại hoạt động như một cổng vào ra ở mức ứng dụng hoặc proxy cho một/một số bộ cảm biến có thể trả lời yêu cầu này; cổng sẽ chuyển yêu cầu này vào các tương tác giao thức mạng phù hợp. Giả sử tồn tại giao thức ở cấp ứng dụng để yêu cầu từ xa và cổng có thể sử dụng và nó phù hợp cho giao tiếp trên Internet hơn các giao thức mạng cảm biến thực tế và nó cũng thuận tiện hơn cho các thiết bị đầu cuối từ xa sử dụng.

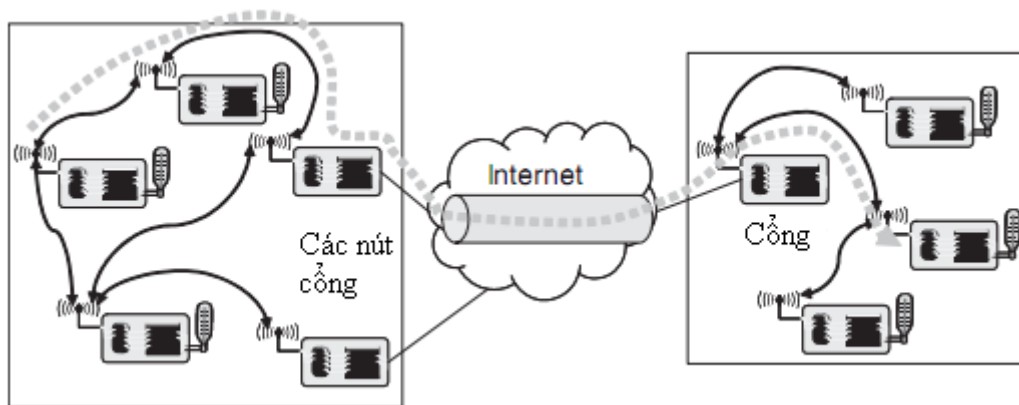
Điều đó không có nghĩa là giao thức mức ứng dụng có thể mô tả một cách đơn giản từ các giao thức mạng cảm biến thực tế đến thiết bị đầu cuối ở xa. Ví dụ, nó không cần cho các thiết bị đầu cuối ở xa để quan tâm với việc duy trì các định tuyến đa bước nhảy trong mạng hay nó cũng không thể được xem như "một bước nhảy khác" khi các đặc tính của kết nối Internet rất khác so với bước nhảy không dây.

Ngoài ra, giao thức mức ứng dụng còn có một số đặc điểm tương đương với các giao thức dịch vụ web (Web Service Protocols), đó là nó có thể mô tả một cách rõ ràng các dịch vụ và cách nó có thể được truy cập. Đặc biệt, ngôn ngữ mô tả dịch vụ web (WSDL - Web Service Description Language) có thể sẽ là một khởi đầu đầy hứa hẹn cho việc mở rộng yêu cầu đối với việc truy cập dịch vụ WSN. Ví dụ, yêu cầu độ chính xác, sự thỏa hiệp năng lượng, hay mô tả dịch vụ tập trung dữ liệu. Câu hỏi đặt ra ở đây là làm thế nào để tích hợp WSN với các kiến trúc middleware tổng quát hay làm thế nào để có thể truy cập các dịch vụ WSN từ một trình duyệt Web tiêu chuẩn (thực hiện tự động bằng cách sử dụng WSDL và các chuẩn liên quan trong công vào ra).

3.5.4 Giao tiếp giữa các WSN.

Ngoài trường hợp giao tiếp giữa WSN và các đầu cuối Internet, các cổng vào ra còn có thể hoạt động như một phần mở rộng đơn giản của giao tiếp giữa các WSN với nhau. Ý tưởng ở đây là để xây dựng một WSN “ảo” lớn hơn các WSN riêng rẽ. Tất cả các bản tin giao thức giữa hai WSN sẽ được trao đổi với nhau bằng cách sử dụng Internet như là một mạng vận chuyển. Để thực hiện được điều này thì phải không có sự nhầm lẫn giữa các liên kết ảo giữa hai nút cổng với liên kết thực. Nếu không, các giao thức dựa vào tính chất vật lý của liên kết truyền thông có thể gặp vấn đề không mong muốn (ví dụ như đồng bộ hóa thời gian).

Kết nối giữa hai mạng không đòi hỏi là các kết nối cố định, ngay cả các nút di động cũng có thể được coi là phương tiện để kết nối trung gian của WSN.



Hình 3.11 Kết nối giữa hai WSN qua Internet

3.6 Kết luận

Chương này cho ta biết mạng cảm biến không dây và kiến trúc của nó có nhiều kiểu khác nhau. Đối với nhiều ứng dụng (nhưng không phải tất cả), giao tiếp đa bước nhảy là công nghệ quan trọng và hầu hết các nghiên cứu WSN cũng như phần sau của cuốn sách này tập trung vào dạng đặc biệt này của mạng vô tuyến.

Bốn mục tiêu tối ưu chính của WSN là chất lượng dịch vụ, hiệu quả năng, khả năng mở rộng, và tính bền vững bao phủ các yêu cầu của các WSN và chúng phải được coi trọng như nhau. Muốn vậy thì việc thiết kế các WSN phải dựa vào mạng truyền thống, và do đó nó tạo ra một số quy tắc thiết kế. Điều quan trọng nhất là tổ chức phân phối mạng, sử dụng xử lý trong mạng, tập trung dữ liệu. Các kết quả phải trung thực và thực sự phù hợp với từng hoàn cảnh nhất định. Đây là kỹ thuật quan trọng được xem xét để sử dụng WSN.

Sự đa dạng của WSN làm cho việc thiết kế mạng đồng đều và có giao diện dịch vụ chung rất khó thực. Do đó, không có giải pháp cuối cùng để giải quyết vấn đề. Tương tự, sự tích hợp của các WSN để tạo ra một mạng lớn hơn, ví dụ như để cho phép các máy chủ dựa vào Internet truy cập vào các dịch vụ của WSN, vẫn là một vấn đề mở.

CÂU HỎI ÔN TẬP CHƯƠNG 3

1. Phân biệt mạng một bước nhảy và mạng đa bước nhảy
2. Các loại di chuyển thường gặp trong WSN?
3. Các mục tiêu tối ưu và chất lượng dịch vụ WSN?
4. Các nguyên tắc thiết kế WSN?
5. Giao diện dịch vụ của WSN?
6. Khái niệm và chức năng của cổng vào/ra?

CHƯƠNG 4. LỚP VẬT LÝ

Mục đích của chương

Chương này được dành cho lớp vật lý bao gồm các chức năng của nó, các thành phần của một nút cảm biến trung gian giữa việc truyền và nhận vô tuyến, xử lý tín hiệu số trong nút, bao gồm cả xử lý giao thức lớp cao hơn.

Không thể phủ nhận rằng các tính chất của kênh truyền và lớp vật lý là các phần quan trọng của ngăn xếp giao thức. Mục tiêu đầu tiên của chương này, do đó, là cung cấp cho người đọc những hiểu biết về một số khái niệm cơ bản liên quan đến thông tin số qua các kênh vô tuyến.

Mục tiêu quan trọng thứ hai là giải thích những bắt buộc của một WSN (ví dụ như năng lượng và giá thành của nút) để từ đó tạo ra sơ đồ điều chế và các bộ thu phát. Người đọc sẽ có hiểu biết về một số thỏa hiệp cơ bản liên quan đến sức mạnh của việc truyền tin và mức năng lượng tiêu thụ, những ảnh hưởng của chúng đến các tính chất tiêu thụ năng lượng của các thành phần trong bộ thu phát.

4.1 Giới thiệu

Lớp vật lý chủ yếu quan tâm đến điều chế và giải điều chế dữ liệu số. Nhiệm vụ này được thực hiện bởi các bộ thu phát. Trong các mạng cảm biến, thách thức là tìm sơ đồ điều chế và kiến trúc bộ thu phát phải đơn giản, giá rẻ, nhưng vẫn đủ mạnh để cung cấp dịch vụ mong muốn.

Phần đầu của chương này giải thích các khái niệm quan trọng nhất liên quan đến các kênh vô tuyến và truyền thông số (qua các kênh vô tuyến). Mục đích chính của nó là cung cấp các khái niệm thích hợp và có một cái nhìn sâu sắc về các nhiệm vụ liên quan đến việc truyền và nhận qua các kênh vô tuyến. Phần thứ hai thảo luận các yêu cầu cụ thể của mạng cảm biến không dây, đặc biệt chú ý đến vấn đề năng lượng, việc thiết kế các bộ thu phát và sơ đồ truyền tin.

4.2 Kênh vô tuyến và cơ sở truyền tin

Phần này cung cấp nền tảng cơ bản về các kênh vô tuyến và truyền thông số qua chúng, sẽ không đi sâu tìm hiểu vấn đề này mà chỉ dừng lại ở việc cung cấp cơ sở và các khái niệm quan trọng nhất để hiểu các khía cạnh liên quan đến năng lượng.

Trong các kênh vô tuyến, sóng điện từ lan truyền gần như tự do trong không gian giữa bộ phát và bộ thu. Vì vậy, kênh vô tuyến là một môi trường không điều khiển, nghĩa là tín hiệu truyền đi không bị hạn chế đến các vị trí đã được xác định.

4.2.1 Cấp phát tần số

Đối với một mạng vô tuyến thực tế, hệ thống RF, tần số sóng mang phải được chọn cẩn thận. Tần số sóng mang này xác định các đặc tính truyền sóng và năng lực của kênh truyền. Với các mục đích truyền thông thì một phần hữu hạn của phổ điện từ, được gọi là **băng tần**, được sử dụng. Trong giao tiếp tần số vô tuyến (RF - Radio-Frequency), dải tần có thể sử dụng thường bắt đầu ở tần số thấp rất thấp (VLF - Very Low Frequency) và kết thúc ở tần số cực cao (EHF - Extremely High Frequency) (Hình 4.1). Đây cũng là lựa chọn của **giao tiếp hồng**

ngoại và thông tin quang. Phổ hồng ngoại có bước sóng từ 1 mm (tương ứng với 300 GHz) đến 2,5 μm (120 THz), trong khi phổ quang kết thúc ở 780 nm (≈ 385 THz).

3kHz 30kHz 300kHz 3MHz 30MHz 300MHz 3GHz 30GHz 300GHz

VLF	LF	MF	HF	VHF	UHF	SHF	EHF	
100km	10km	1km	100m	10m	1m	10cm	1cm	1mm

VLF = Very low frequency (tần số rất thấp)

LF = Low frequency (tần số thấp)

MF = Medium frequency (tần số trung bình)

HF = High frequency (tần số cao)

VHF = Very high frequency (tần số rất cao)

UHF = Ultrahigh frequency SHF = Super high frequency (tần số siêu cao)

EHF = Extremely high frequency (tần số cực cao)

Hình 4.1 Phổ điện từ - tần số vô tuyến

Tần số	Ghi chú
13.553 – 13.567 MHz	
26.957 – 27.283 MHz	
40.66 – 40.70 MHz	
433 – 464 MHz	Châu Âu
902 – 928 MHz	Chỉ dùng ở Mỹ
2.4 – 2.5 GHz	Chỉ sử dụng công nghệ WLAN/WPAN
5.725 – 5.875 GHz	Chỉ sử dụng công nghệ WLAN
24 – 24.25 GHz	

Bảng 4.1 Một số dải ISM

Việc lựa chọn dải tần làm việc là một yếu tố quan trọng trong thiết kế hệ thống. Ngoại trừ các công nghệ băng tần siêu rộng (UWB – Ultra Wide Band) (xem Phần 2.1.4), hầu hết các hệ thống RF hiện nay làm việc ở tần số dưới 6 GHz. Dải tần vô tuyến phải được hiệu chỉnh để tránh nhiễu không mong muốn, phù hợp với các hệ thống và người sử dụng khác nhau. Một số hệ thống phải được cấp phép đặc biệt cho các băng tần riêng biệt. Ví dụ, ở châu Âu, hệ thống GSM có thể sử dụng các băng tần GSM 900 (880-915 MHz) và GSM 1800 (1710-1785 MHz). Cũng có những băng tần được cấp miễn phí như các băng tần công nghiệp, khoa học và y tế (ISM). Bảng 4.1 liệt kê một số băng tần ISM. Làm việc ở dải tần không được cấp phép, có nghĩa là nó có thể làm việc mà không cần bất kỳ sự cho phép nào từ chính phủ hay từ nơi cấp phát tần số. Không ngạc nhiên khi những băng tần này khá phổ biến, không chỉ cho các mạng cảm biến mà còn cho/ trong các công nghệ vô tuyến khác. Ví dụ, băng tần 2.4 GHz ISM được sử dụng cho IEEE 802.11, Bluetooth, và IEEE 802.15.4. Sau đây là một số

chú ý khi lựa chọn tần số làm việc:

- Trong các băng tần ISM công cộng, hệ thống luôn tồn tại nhiều được tạo bởi các hệ thống khác (sử dụng các công nghệ giống và khác nhau) trong cùng một băng tần, chỉ đơn giản vì nó không bị hạn chế sử dụng. Ví dụ, có nhiều hệ thống cùng chia sẻ băng tần 2.4 GHz ISM, bao gồm IEEE 802.11b, Bluetooth và IEEE 802.15.4 WPAN – chúng **cùng tồn tại** với nhau trong cùng một băng tần. Vì vậy, tất cả các hệ thống trong những băng tần này phải đủ mạnh để chống lại nhiễu từ các hệ thống khác. Khái niệm cùng tồn tại cần phải được tiếp cận cả lớp vật lý và lớp MAC.
- Một thông số quan trọng trong hệ thống truyền dẫn là hiệu suất antenna, được định nghĩa là tỷ lệ giữa công suất phát và tổng công suất đầu vào của antenna. Công suất còn lại bị tiêu tán do nhiệt. Do các nút cảm biến có kích thước nhỏ nên nó chỉ sử dụng được các antenna nhỏ. Ví dụ, sóng vô tuyến ở 2.4 GHz có bước sóng 12.5 cm, dài hơn nhiều so với kích thước dự kiến của các nút cảm biến. Nhìn chung, rất khó để xây dựng được một antenna có hiệu quả khi giảm kích thước của antenna để giảm bước sóng. Nếu làm được điều này thì năng lượng sử dụng sẽ tăng lên để có cùng một công suất phát.

4.2.2 Điều chế và giải điều chế

Khi máy tính số giao tiếp, nó trao đổi **dữ liệu số**, mà chủ yếu là các **ký hiệu** (symbol). Mỗi ký hiệu được lấy từ bảng chữ cái, **channel alphabet**. Trong quá trình điều chế, các ký hiệu từ **channel alphabet** được ánh xạ tới một trong các dạng sóng có cùng chiều dài hữu hạn. Chiều dài này được gọi là **bề rộng ký hiệu** (symbol duration). Với hai dạng sóng khác nhau tạo ra **điều chế nhị phân** (binary modulation). Nếu kích thước là $m \in \mathbb{N}$, $m > 2$, ta có điều chế m -ary. Các trường hợp phổ biến là số nhị phân (gồm hai ký hiệu là 0, 1) và số lưỡng cực (gồm hai ký hiệu là 1, -1).

Khi đề cập đến các “tốc độ” truyền/ điều chế dữ liệu, ta cần phân biệt giữa các tham số sau:

Tốc độ ký hiệu (Symbol rate): Là nghịch đảo của bề rộng ký hiệu. Trong điều chế nhị phân, nó còn được gọi là tốc độ bit.

Tốc độ dữ liệu (Data rate): Là số bit trong một giây mà bộ điều chế có thể chấp nhận được để truyền. Do đó, tốc độ này được tính khi dùng dữ liệu nhị phân. Đối với điều chế nhị phân, tốc độ bit và tốc độ dữ liệu giống nhau và thường dùng tốc độ bit để biểu thị tốc độ dữ liệu.

Đối với điều chế m -ary, tốc độ dữ liệu là tích giữa tốc độ ký hiệu và số bit mã hóa trong một sóng đơn. Ví dụ, nếu sử dụng điều chế 8-ary, chúng ta có thể kết hợp mỗi dạng sóng với một trong tám nhóm ba bit và do đó, tốc độ bit gấp ba lần tốc độ ký hiệu.

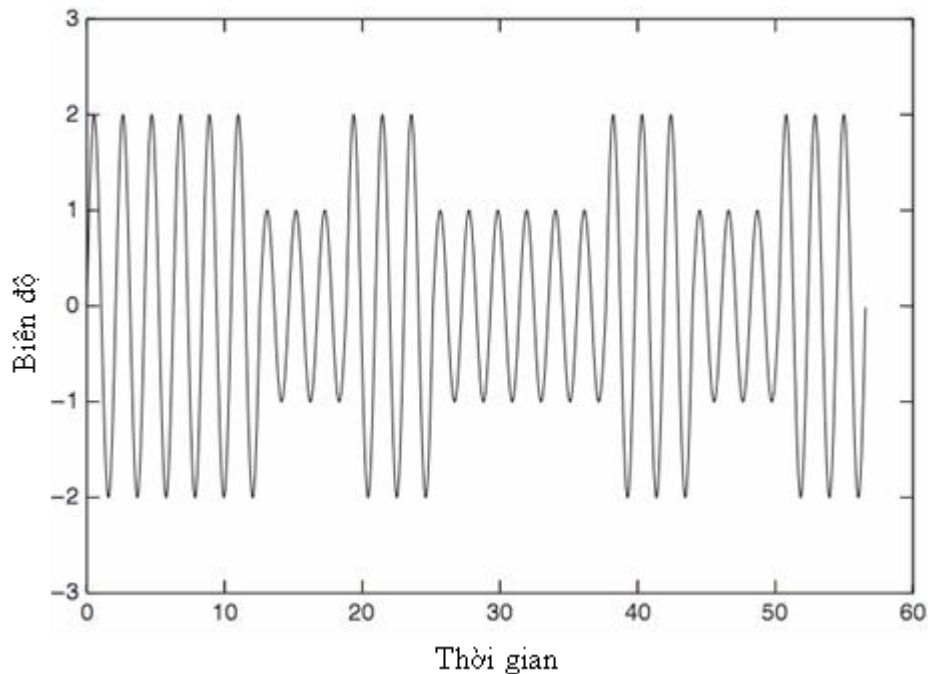
Điều chế được thực hiện ở máy phát. Người nhận cuối cùng muốn khôi phục lại dữ liệu đã phát từ dạng sóng nhận được. Quá trình biến đổi từ sóng nhận được để tạo ra các ký hiệu được gọi là **giải điều chế** (demodulation). Vì tạp âm, sự suy giảm, hoặc nhiễu nên các dạng sóng nhận được thường bị méo (so với sóng đã phát đi), do đó bộ thu không thể xác định một cách chắc chắn dữ liệu đã phát. Thay vào đó, bộ thu quyết định ký hiệu lỗi với một xác suất nào đó, được gọi là **tỉ số lỗi ký hiệu** (symbol error rate). Đối với dữ liệu số được biểu diễn theo bit, khái niệm này rất quan trọng và được gọi là **tỉ số lỗi bit** (BER - Bit Error Rate), nó mô tả xác suất một bit chuyển tới lớp cao hơn bị sai. Nếu sử dụng điều chế nhị phân, xác suất lỗi bit và xác suất lỗi ký hiệu là giống nhau. Trong trường hợp điều chế m -ary, chúng có thể khác nhau. Ngay cả khi tín hiệu được giải điều chế không chính xác, nhóm bit chuyển đi vẫn

có thể đúng tại một số vị trí (miễn là SNR không quá thấp). Tất cả các lớp cao hơn chủ yếu quan tâm đến xác suất lỗi bit.

Hình thức điều chế phổ biến nhất là **điều chế băng thông**, khi đó tín hiệu được điều chế trên các sóng mang tuần hoàn ở dải tần cao. Phổ được sử dụng bởi sơ đồ điều chế băng thông thường được mô tả theo **tần số trung tâm** f_c và **băng thông** B . Phần lớn năng lượng của tín hiệu có thể được tìm thấy trong các dải tần số $(f_c - B/2, f_c + B/2)$. Sóng mang thường được biểu diễn dưới dạng sóng cosin, nó là duy nhất khi biết biên độ, tần số và góc dịch pha. Theo đó, tín hiệu điều chế $s(t)$ thường được biểu diễn dưới dạng:

$$s(t) = A(t) \cdot \cos(\omega(t) + \phi(t))$$

trong đó $A(t)$ là biên độ phụ thuộc thời gian, $\omega(t)$ là tần số phụ thuộc thời gian, và $\phi(t)$ là góc dịch pha. Theo đó, có ba loại điều chế cơ bản: khoá dịch biên (ASK - Amplitude Shift Keying), khoá dịch pha (PSK - Phase Shift Keying) và khoá dịch tần (FSK - Frequency Shift Keying). Chúng có thể sử dụng độc lập hoặc kết hợp với nhau.



Hình 4.2 Ví dụ về ASK

Trong ASK, $s_i(\cdot)$ cho các ký hiệu khác nhau là:

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos(\omega_0(t) + \phi)$$

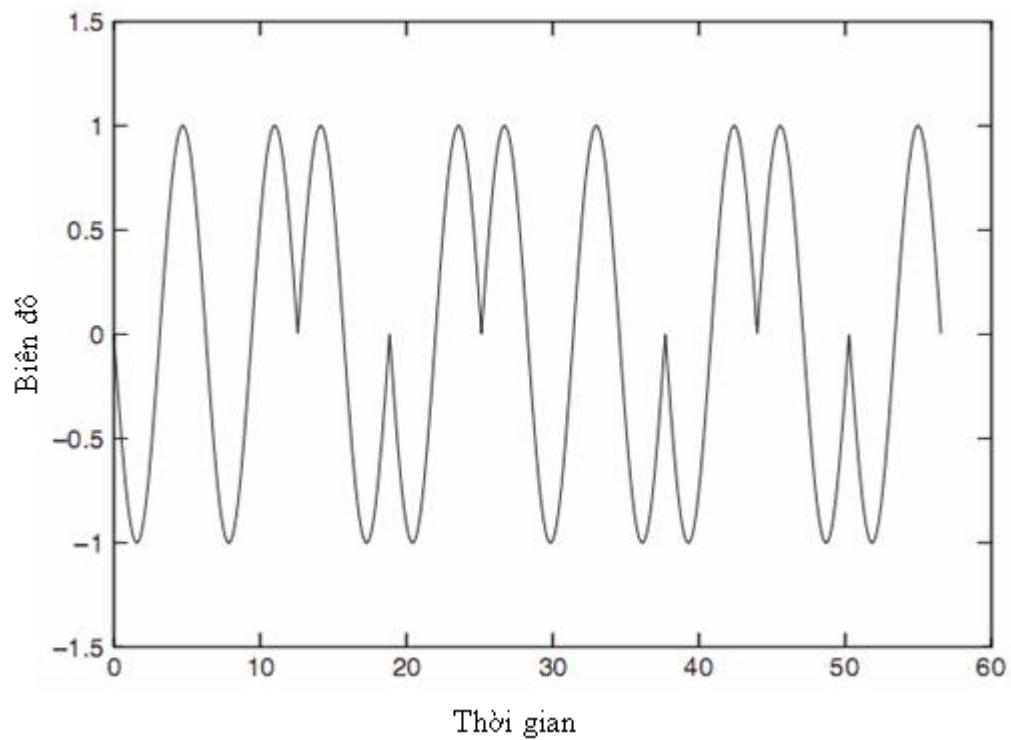
Với ω_0 là tần số trung tâm, ϕ là góc pha ban đầu, $E_i(t)$ không đổi trong một chu kỳ $[0, T]$. Quy ước $\sqrt{\frac{2E_i(t)}{T}}$ là biên độ, nó biểu thị **năng lượng ký hiệu** E . Một ví dụ về ASK

được chỉ ra trên hình 4.2, trong đó chuỗi nhị phân 110100101 được điều chế, sử dụng $E_0(t) = 1$ và $E_1(t) = 2$ cho tất cả t biểu diễn giá trị logic 0 và 1. Trường hợp đặc biệt của điều chế ASK là sơ đồ kênh nhị phân trong đó các điểm 0 biểu thị trường hợp không có tín hiệu và các điểm 1 cho những trường hợp còn lại. Do vậy, nó tương đương với một chuyển mạch

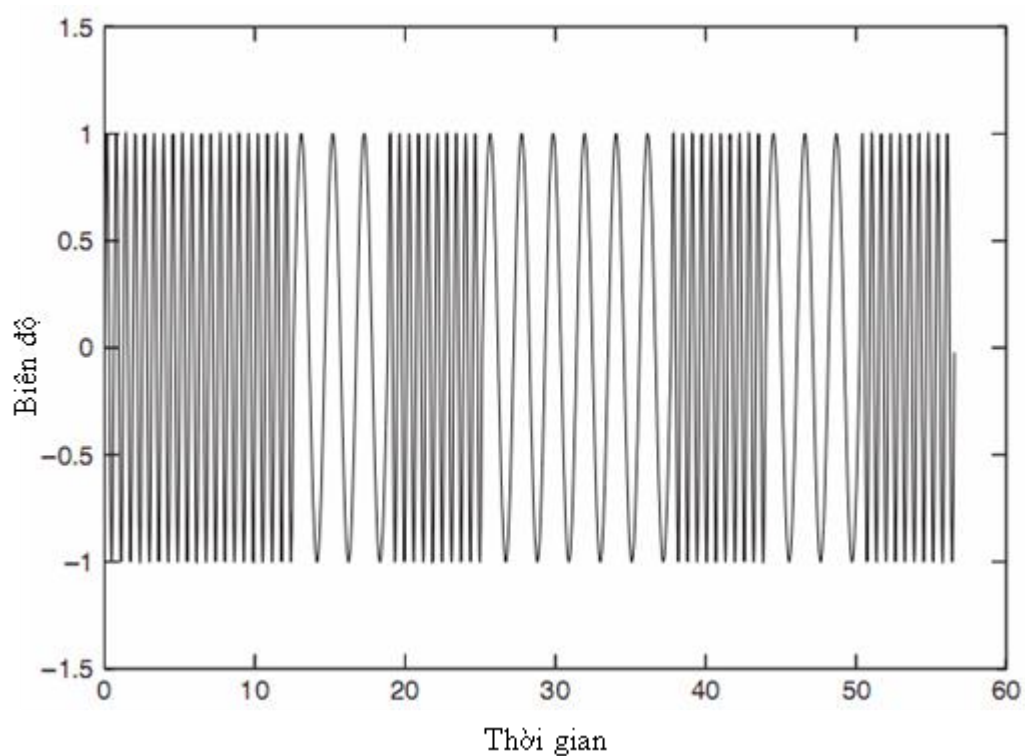
của bộ phát nên nó còn được gọi là khoá đóng – mở (OOK – On Off Keying). Trong PSK ta có:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_0(t) + \phi_i(t))$$

Với ω_0 là tần số trung tâm, E là năng lượng ký hiệu và $\phi_i(t)$ là một trong m giá trị hằng số khác nhau biểu thị góc dịch pha.



Hình 4.3 Ví dụ về PSK



Hình 4.4 Ví dụ về FSK

Hình 4.3 biểu thị PSK cũng với dữ liệu nhị phân như trong ví dụ ASK. Hai sơ đồ PSK phổ biến là BPSK (Binary Phase Shift Keying – khoá dịch pha nhị phân) và QPSK (Quaternary Phase Shift Keying – khoá dịch pha tứ phân). Ví dụ, chúng được sử dụng cho các điều chế 1-Mbps và 2-Mbps trong IEEE 802.11. Trong BPSK sử dụng dịch pha của 0 và π , còn trong QPSK sử dụng dịch pha của 0, $\pi/2$, π và $3\pi/2$.

Trong FSK ta có:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i(t).t + \phi)$$

Với $\omega_i(t)$ là một trong n tần số khác nhau, E là năng lượng ký hiệu và ϕ là góc pha ban đầu. Hình 4.4 vẫn là ví dụ trên nhưng sử dụng điều chế FSK.

Rõ ràng là ba loại điều chế cơ bản này có thể kết hợp với nhau. Ví dụ, **điều chế biên độ vuông góc** (QAM – Quadrature Amplitude Modulation) là sự kết hợp giữa điều chế biên độ và điều chế pha bằng cách sử dụng hai biên độ khác nhau và hai góc pha khác nhau để biểu thị hai bit trong một ký hiệu.

4.2.3 Các ảnh hưởng của truyền sóng và nhiễu

Sóng truyền qua kênh vô tuyến bị ảnh hưởng của một số hiện tượng vật lý gây *méo* sóng nhận được ở máy thu. Méo là hiện tượng không mong muốn, nó tạo ra các bit lỗi ở máy thu.

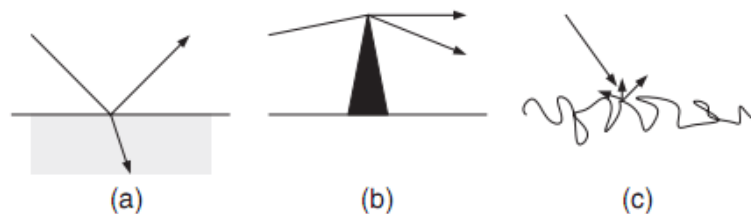
Phản xạ, khúc xạ, tán xạ và fading Doppler

Các hiện tượng truyền sóng cơ bản là:

Phản xạ: Khi sóng đang truyền trong môi trường A gặp một môi trường B khác, giả sử lớp tiếp giáp giữa chúng là nhẵn thì một phần sóng sẽ được phản xạ trở lại môi trường A , một phần được truyền vào môi trường B và được hấp thụ (hình 4.5(a)). Năng lượng phản xạ/truyền/hấp thụ phụ thuộc vào môi trường và tần số truyền.

Khúc xạ: Theo nguyên lý của Huygen, tất cả các điểm trên mặt sóng đều có thể được xem như là nguồn của một mặt sóng mới. Nếu một mặt sóng gặp một đỉnh nhọn, nó có thể làm cho sóng bị chuyển vào vùng che khuất (hình 4.5(b)).

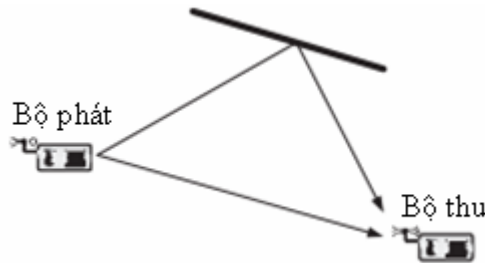
Tán xạ: Khi sóng tiếp xúc với bề mặt gồ ghề, nó có thể bị phản xạ nhiều lần và bị khuếch tán theo nhiều hướng (hình 4.5(c)).



Hình 4.5 Mô tả hiện tượng truyền sóng

Fading Doppler: Khi bộ phát và bộ thu chuyển động tương đối với nhau, sóng thu được sẽ bị dịch tần, tạo nên hiệu ứng Doppler. Sự dịch tần quá nhiều sẽ làm cho bộ thu nhận tín hiệu ở sai tần số. Tần số tăng lên khi bộ phát ở gần bộ thu và giảm khi chúng xa nhau. Như vậy trong hiệu ứng Doppler, bản thân tần số của nguồn sóng không bị thay đổi mà chính khoảng cách thay đổi giữa bộ phát và bộ thu gây nên sự thay đổi tần số.

Antenna vô tuyến phát tín hiệu theo tất cả các hướng với cùng một cường độ, hoặc nó có một hướng ưu tiên đặc trưng bởi một chùm tín hiệu. Trong trường hợp đầu tiên, ta có antenna **mọi hướng** (omnidirectional antenna), và trong lần thứ hai, ta có antenna **một hướng** (directed antenna). Trong cả hai trường hợp, không chỉ có một mà có rất nhiều bản sao của tín hiệu đến được bộ thu từ các đường có độ dài và độ suy giảm khác nhau. Hình 4.6 minh họa điều đó, nó gồm đường trực tiếp hay **đường nhìn thẳng** (LOS - Line Of Sight) và đường phản xạ hay **đường nhìn không thẳng** (NLOS - Non line Of Sight).



Hình 4.6 Truyền sóng đa đường

Vì vậy, tín hiệu ở máy thu là sự xếp chồng của nhiều bản sao bị trễ của cùng một tín hiệu. Tín hiệu thực sự chiếm một phổ nhất định, được biểu diễn bằng công thức Fourier. Các bản sao khác nhau của tín hiệu có độ trễ khác nhau, đó là do sự khác nhau về tần số của tín hiệu gây ra sự dịch pha ở máy thu. Sự dịch pha tương đối của các thành phần tín hiệu tạo ra nhiều tích cực và tiêu cực. Nếu kênh xử lý tất cả các thành phần tần số của tín hiệu theo cùng một cách (có nghĩa là biên độ ở máy thu tương đối mạnh), ta có **fading không chọn tần** (frequency- nonselective fading), còn được gọi là **fading phẳng** (flat fading), nếu không ta sẽ có **kênh chọn tần** (frequency-selective channel). Tính lựa chọn (hay không lựa chọn) tần số của một kênh là liên quan chặt chẽ đến sự **phân tán thời gian** (time dispersion) hay **độ rộng trễ** (delay spread), hay chính xác hơn là liên quan đến giá trị độ rộng trễ RMS. **Giải thông phù hợp** (coherence bandwidth) của một môi trường truyền sóng là dải tần mà qua nó, kênh truyền được coi như phẳng. Nó được định nghĩa là nghịch đảo của tích giữa độ rộng trễ RMS và một hệ số không đổi. Kênh là kênh fading phẳng nếu toàn băng thông tín hiệu nhỏ hơn giải thông phù hợp.

Đối với các mạng cảm biến không dây, phạm vi truyền nhỏ (làm cho độ rộng trễ RMS nhỏ) và tốc độ ký hiệu tương đối thấp. Đây chính là lý do để giả sử nó là các kênh fading phẳng.

Khi máy phát và máy thu di chuyển tương đối với nhau, số lượng và độ lệch pha tương đối của nhiều đường truyền sẽ thay đổi theo thời gian và cường độ tín hiệu nhận được có thể dao động trong khoảng 30 - 40 dB trong thời gian ngắn. Đó được gọi là **fading nhanh** (fast fading) hay **fading đa đường** (multipath fading). Phụ thuộc vào tốc độ chuyển động tương đối, các dao động xảy ra trong khoảng thời gian từ hàng chục đến hàng trăm ms.

Tầm quan trọng của fading chính là tác động của nó đến máy thu. Do bất kỳ một máy thu nào cũng muốn cường độ tín hiệu là nhỏ nhất để có thể chọn được phương pháp giải điều chế phù. Khi cường độ tín hiệu thấp hơn ngưỡng của nó do fading nhanh gây ra, được gọi là **suy giảm sâu** (deep fade). Khi xét các kênh fading nhanh, tốc độ mà tại đó tín hiệu nằm dưới mức ngưỡng (**tốc độ vượt mức** - level-crossing rate) và khoảng thời gian xảy ra suy giảm sâu là các tham số rất quan trọng. Về mặt chất lượng, kênh fading có xu hướng có lỗi chùm,

đó là lỗi ký hiệu thường xảy ra trong trong các cụm, được ngăn cách bằng những khoảng không lỗi.

Nguồn lỗi khác (chủ yếu) do truyền sóng đa đường gây ra là **hiệu nhiễu liên ký hiệu** (ISI – InterSymbol Interference): Khi máy phát truyền các ký hiệu đầu lung, sự xuất hiện của nhiễu đa đường với độ trễ khác nhau có thể dẫn đến tình huống là dạng sóng phụ thuộc vào một số ký hiệu s_t và đến máy thu theo đường nhìn thẳng (LOS), nó trùng với các bản sao bị trễ của các ký hiệu đã được gửi trước đó s_{t-1}, s_{t-2}, \dots . Mức độ nghiêm trọng của ISI phụ thuộc vào mối quan hệ giữa khoảng cách của các ký hiệu và độ rộng trễ RMS.

Suy hao đường truyền và sự suy giảm

Sóng vô tuyến được truyền tự do trong môi trường là đối tượng của suy hao công suất phụ thuộc khoảng cách, còn được gọi là **suy hao đường truyền** (path loss). Công suất nhận được ở khoảng cách $d \geq d_0$ giữa máy phát và máy thu được biểu diễn bằng **phương trình không gian tự do Friis** (Friis free-space equation):

$$\begin{aligned} P_{rcvd}(d) &= \frac{P_{tx} \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2 \cdot L} \\ &= \frac{P_{tx} \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d_0^2 \cdot L} \left(\frac{d_0}{d} \right)^2 = P_{rcvd}(d_0) \left(\frac{d_0}{d} \right)^2 \end{aligned} \quad (0.11)$$

Trong đó P_{tx} là **công suất phát**; G_t và G_r là **độ khuếch đại** của antenna của máy phát và máy thu; d_0 được gọi là **khoảng cách trường xa** (far-field distance), nó phụ thuộc vào công nghệ của antenna, $d \geq d_0$ là khoảng cách giữa máy phát và máy thu; λ là **bước sóng** và $L \geq 1$ là **tổng suy hao** qua hệ thống phát/thu. Chú ý rằng phương trình này chỉ có giá trị khi $d \geq d_0$. Với môi trường khác, phương trình có dạng:

$$P_{rcvd}(d) = P_{rcvd}(d_0) \left(\frac{d_0}{d} \right)^\gamma \quad (0.12)$$

Trong đó γ là **số mũ của suy hao đường truyền**, nó thường thay đổi giữa 2 (suy hao trong không gian tự do) và 5 đến 6 (các vùng bị che khuất và ở trong nhà). Thậm chí nếu có nhiều tích cực thì $\gamma < 2$. Suy hao đường truyền được định nghĩa là tỉ số giữa công suất phát và công suất nhận được $P_{tx} / P_{rcvd}(d)$. Từ (0.12) ta có:

$$PL(d)[dB] = PL(d_0)[dB] + 10\gamma \cdot \log_{10} \left(\frac{d}{d_0} \right) \quad (0.13)$$

Đây được gọi là mô hình suy hao theo **khoảng cách logarithm**, $PL(d_0)[dB]$ là suy hao đường truyền ở khoảng cách chuẩn.

Chúng ta có thể rút ra một số kết luận đầu tiên từ phương trình này. *Thứ nhất*, công suất thu được phụ thuộc vào tần số: tần số càng cao thì công suất thu được càng lớn. *Thứ hai*, công suất thu được phụ thuộc khoảng cách theo luật công suất. Ví dụ, giả sử số mũ suy hao đường truyền là 2, nút ở khoảng cách $2d$ đến bộ thu bất kỳ sử dụng gấp 4 lần năng lượng so với nút

ở khoảng cách d vẫn đến bộ thu đó để cùng có được công suất thu như nhau là P_{rcvd} . Do đó, tốc độ lỗi bit/ ký hiệu ở máy thu là hàm đơn điệu của công suất thu P_{rcvd} , tần số càng cao hoặc khoảng cách càng lớn thì phải tăng công suất phát cho phù hợp để duy trì giá trị P_{rcvd} quy định.

Tạp âm và nhiễu

Thông thường, **nhiễu** (interference) là sự hiện diện của các tín hiệu không mong muốn từ bên ngoài làm che hoặc giấu tín hiệu. Nhiễu thường nói tới các nhân tố ngoài môi trường tác động lên kênh truyền. Các tín hiệu này có thể đến từ các máy phát thông tin khác trong cùng một băng thông tại cùng một thời điểm (**nhiễu đa truy cập**) hoặc từ các thiết bị khác như lò vi sóng phát xạ tại cùng một dải tần số. Trong **nhiễu đồng kênh**, các nguồn nhiễu phát xạ tại cùng một dải tần hoặc có các dải tần chồng chéo nhau. Trong **nhiễu kênh lân cận**, nhiễu hoạt động ở dải lân cận.

Hiện tượng quan trọng khác là tạp âm nhiệt, hay còn gọi là tạp âm, thường ám chỉ đến những tác động bên trong hệ thống. Nó được tạo ra do chuyển động nhiệt của các điện tích trong thiết bị dẫn điện như bộ khuếch đại, mạch phát và mạch thu. Ở các bộ thu số, tạp âm thường được đo bằng mật độ phổ công suất (PSD - Power Spectral Density) N_0 :

$$N_0 = K.T \left[\frac{W}{Hz} \right]$$

Trong đó K là hằng số Boltzmanns ($\approx 1.38 \cdot 10^{-23}$ J/K) và T là nhiệt độ của hệ thống (theo độ Kelvin). Tạp âm nhiệt sẽ được cộng thêm vào tín hiệu, nghĩa là tín hiệu thu $r(t)$ là tổng của tín hiệu phát $s(t)$ (đến bộ thu sau khi bị suy hao đường truyền, bị suy giảm, bị tán xạ...) và tín hiệu tạp âm $n(t)$:

$$r(t) = s(t) + n(t) \quad (0.14)$$

Và đây được gọi là tạp âm **Gaussian**, trong đó $n(t)$ là phân bố chuẩn/Gausse có giá trị trung bình bằng 0 và phương sai hữu hạn σ^2 với mọi t . Một tính chất rất quan trọng của tạp âm Gaussian là PSD của nó là một hằng số (có giá trị bằng $N_0/2$ ở mọi tần số). Hiện tượng có PSD là một hằng số còn được gọi là **tạp âm trắng** (white noise). Chính vì vậy, nhiễu nhiệt còn được gọi là **tạp âm Gaussian cộng sinh** (AWGN - Additive White Gaussian Noise).

Sai số ký hiệu và bit

Xác suất lỗi ký hiệu/bit phụ thuộc vào sơ đồ điều chế thực tế, là tỉ số giữa công suất tín hiệu thu được (P_{rcvd}) và công suất của tạp âm và nhiễu. Khi xét AWGN, tỉ số này được gọi là tỉ số tín hiệu/ tạp âm (SNR - Signal-to-Noise Ratio) và được biểu diễn theo công thức sau:

$$SNR = 10 \log_{10} \frac{P_{rcvd}}{N_0}$$

Trong đó N_0 là công suất tạp âm và P_{rcvd} là công suất trung bình của tín hiệu thu. Khi các nguồn nhiễu khác cũng được quan tâm, tỉ số tín hiệu/nhiễu + tạp âm (SINR - Signal to Interference and Noise Ratio) lại đóng vai trò quan trọng:

$$\text{SINR} = 10 \log_{10} \left(\frac{P_{rcvd}}{N_0 + \sum_{i=1}^k I_i} \right)$$

Trong đó N_0 là công suất tạp âm và I_i công suất thu được từ nhiễu thứ i .

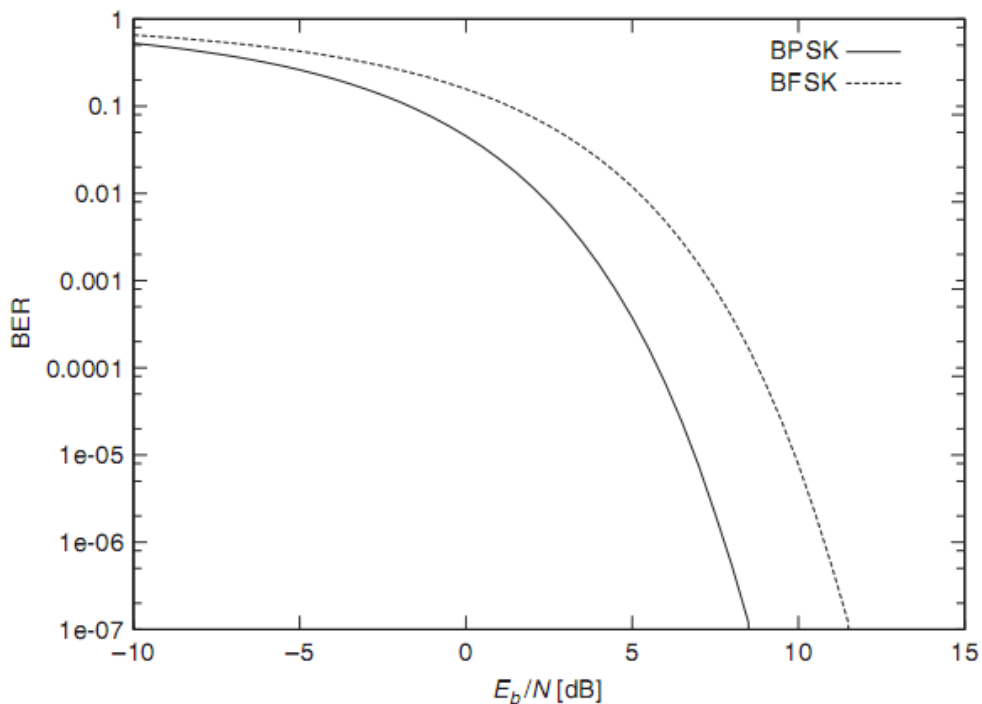
SINR biểu diễn công suất đến các bộ thu và do đó, nó liên quan đến các ký hiệu gửi qua kênh truyền dẫn. Để giải điều chế và giải mã chính xác một bit nhận được, năng lượng trên một bit E_b phải tương quan với năng lượng tạp âm N_0 . Tỉ số E_b/N_0 gần giống SNR (hay SINR khi nhiễu được xem như là một tạp âm).

$$\frac{E_b}{N_0} = \text{SNR} \cdot \frac{1}{R} = \frac{P_{rcvd}}{N_0} \frac{1}{R} \quad (0.15)$$

Trong đó R là tốc độ bit. Thông số này sẽ được sử dụng trong phần sau khi xét đến băng thông W được sử dụng bởi tín hiệu điều chế và sử dụng **hệ số băng thông** (bandwidth efficiency) $\eta_{BW}=R/W$ (bit/Hz) như là một thước đo đánh giá hiệu quả của sơ đồ điều chế. Khi đó, (0.15) trở thành:

$$\frac{E_b}{N_0} = \frac{P_{rcvd}}{N_0} \frac{1}{\eta_{BW} \cdot W} \quad (0.16)$$

Một khác biệt quan trọng không trực tiếp liên quan đến điều chế nhưng liên quan đến các máy thu là **sự tách sóng nhất quán** (coherent detection) và **sự tách sóng không nhất quán** (noncoherent detection). Trong sự tách sóng nhất quán, thông tin về tần số và pha là đầy đủ. Nhìn chung, các bộ thu nhất quán phức tạp hơn các bộ thu không nhất quán rất nhiều nhưng nó yêu cầu tỉ số tín hiệu/ tạp âm nhỏ hơn để đạt được cùng tốc độ lỗi bit (BER).



Hình 4.7 Tốc độ lỗi bit của PSK và FSK nhị phân được tách sóng nhất quán

Muốn có BER đạt giá trị tối đa như mong muốn, ta có thể có nhiều sơ đồ điều chế, từ đó xác định SNR tối thiểu cần để đạt được hệ số BER này trên kênh AWGN. Điều này chỉ ra trên hình 4.7, nó được thực hiện theo hai phương pháp PSK và FSK nhị phân tách sóng nhất quán. Đánh giá về mặt định tính, giá trị của BER theo E_b/N_0 là như nhau đối với các loại điều chế thông thường. Ví dụ, với BPSK, tỉ số E_b/N_0 phải lớn hơn 4 dB để có được BER đạt giá trị nhỏ nhất là 10^{-3} . Công suất tạp âm là cố định, vì vậy ta phải điều chỉnh công suất thu P_{revd} để có được SNR như mong muốn. Với các antenna đã cho, điều này chỉ có thể có được bằng cách tăng công suất phát ở máy thu P_{tx} .

4.2.4 Các mô hình kênh

Theo nghiên cứu về các sơ đồ điều chế và điều chỉnh lỗi, cần thiết phải sử dụng mô hình cho các kênh vô tuyến. Do các kênh vô tuyến thực tế rất phức tạp nên chủ yếu sử dụng các mô hình ngẫu nhiên, trong đó thay thế mô hình phức tạp trong môi trường truyền sóng bằng các biến ngẫu nhiên. Cách đơn giản nhất là các mô hình này làm việc ở cấp độ dạng sóng để mô tả các tín hiệu nhận được. Cách phức tạp hơn, các mô hình mô tả các số liệu thống kê về ký hiệu, bit lỗi hoặc thậm chí là các gói bit. Các mô hình này phải tuân theo đặc điểm của các giao thức mạng, nơi thường xuyên có hàng nghìn hoặc hàng triệu gói tin được truyền đi.

Mô hình tín hiệu

Giả sử ta có một mô hình dạng sóng, mô hình AWGN với SNR là hằng số. Mô hình này biểu diễn tín hiệu thu $r(t)$ ở dạng:

$$r(t) = s(t) + n(t)$$

Trong đó $s(t)$ là tín hiệu phát và $n(t)$ là tạp âm trắng Gaussian. Một tính chất quan trọng của mô hình này là SNR luôn là hằng số. Mô hình này dễ dàng phân tích về mặt lý thuyết nhưng nó lại không phù hợp với mô hình có các kênh biến đổi theo thời gian, ví dụ như các kênh fading.

Hiện có các mô hình phổ biến khác, đặc biệt là cho các kênh fading không chọn tần. Các mô hình này giả thiết các SNR là một biến ngẫu nhiên, dao động từ ký hiệu này sang ký hiệu khác hoặc từ khối này sang khối khác. Trong mô hình **fading Rayleigh**, nó được giả định rằng không có đường LOS. Thay vào đó, một số lượng lớn các bản sao tín hiệu có biên độ độc lập một cách ngẫu nhiên nhưng có giá trị trung bình giống nhau và chồng lên nhau ở máy thu. Theo định lý giới hạn trung tâm, biên độ của tín hiệu thu được có phân bố Rayleigh, trong đó pha được phân bố chuẩn trong khoảng $[0, 2\pi]$. Mô hình phổ biến thứ hai là mô hình **fading Rice**, nó có cùng giả thiết như mô hình fading Rayleigh nhưng thành phần LOS phải lớn. Kênh fading như vậy kết hợp với AWGN có thể được biểu diễn:

$$r(t) = R.e^{i\theta}.s(t) + n(t)$$

Trong đó $n(t)$ là tạp âm Gaussian trắng và $R.e^{i\theta}$ là biến ngẫu nhiên Gaussian với R là hàm mật độ xác suất Rice hoặc Rayleigh.

Mô hình số

Trong kênh AWGN, mỗi ký hiệu được truyền đi sẽ bị lỗi với một xác suất lỗi cố định và lỗi của các ký hiệu gần nhau độc lập với nhau. Nếu hai điều kiện này là đúng và xác suất lỗi không phụ thuộc vào giá trị ký tự thì ta có **kênh đối xứng nhị phân** (BSC - Binary Symmetric Channel).

Đã có một số kết quả khi tìm các mô hình ngẫu nhiên tốt cho các kênh fading (Rayleigh) ở mức độ bit/ký tự. Những mô hình này cố gắng nắm bắt được xu hướng của kênh fading để có các lỗi chùm. Thông thường, các kênh này được mô hình thành các **chuỗi Markov** với các trạng thái của chuỗi tương ứng với “các mức chất lượng” của các kênh khác nhau. Ví dụ, **mô hình Gilbert - Elliot** hai trạng thái mô tả sự thay đổi giữa trạng thái tốt và sự suy giảm sâu của kênh fading.

Các mô hình kênh WSN cụ thể

Một trong những hạn chế trong thiết kế WSN là sử dụng công suất truyền nhỏ (do vậy mà công suất phát nhỏ), cỡ 1dBm trên một bước nhảy để tiết kiệm năng lượng nhưng nó làm ảnh hưởng đến giao tiếp đa bước nhảy. Lựa chọn công suất phát nhỏ làm ảnh hưởng đến các đặc tính của kênh truyền, cụ thể là:

- Theo phương trình Friis (phương trình 4.1), công suất phát nhỏ có nghĩa là phạm vi truyền nhỏ.
- Phạm vi truyền nhỏ có nghĩa là độ rộng trễ RMS vào khoảng vài ns, nhỏ so với độ rộng ký hiệu (vài ms hoặc μ s). Do đó, tốc độ dữ liệu ở mức vừa phải, đó là lý do tại sao sử dụng các kênh fading có tạp âm không chọn tần và ISI thấp hoặc không đáng kể. Vậy nên không có yêu cầu đặc biệt nào để ISI giống các bộ lọc tử hoá.

4.2.5 Công nghệ trải phổ

Trải phổ là một kỹ thuật truyền thông được đặc trưng bởi băng thông rộng và công suất thấp. Trong các hệ thống trải phổ, băng thông của sóng phát lớn hơn yêu cầu thực tế để truyền dữ liệu của người sử dụng. Tín hiệu được *trải rộng* ở máy phát và *thu gọn* ở máy thu. Sử dụng tín hiệu băng rộng sẽ giảm được tạp âm/nhiều băng hẹp. Các hệ thống trải phổ thường có tính bền vững cao để chống lại các ảnh hưởng đa đường nhưng vì thế mà máy thu đắt và phức tạp hơn để có được sơ đồ điều chế phù hợp.

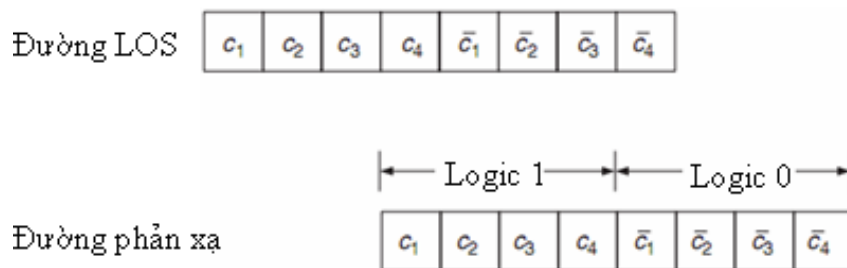
Hai loại công nghệ trải phổ phổ biến nhất là trải phổ dãy trực tiếp (DSSS - Direct Sequence Spread Spectrum) và trải phổ nhảy tần (FHSS - Frequency Hopping Spread Spectrum).

Trải phổ dãy trực tiếp

Trong **trải phổ dãy trực tiếp** (DSSS), việc truyền bit dữ liệu trong khoảng thời gian t_b được thay bằng việc truyền dãy chip hữu hạn $c = c_1c_2...c_n$ với $c_i \in \{0, 1\}$ nếu bit có giá trị logic bằng 1, hoặc $c = \bar{c}_1\bar{c}_2...\bar{c}_n$ nếu bit có giá trị logic bằng 0 (\bar{c}_i là đảo của c_i). Mỗi chip c_i có độ rộng $t_c = t_b/n$, trong đó n là hệ số khuếch đại hay hệ số trải. Mỗi chip sau đó sẽ được điều chế theo sơ đồ điều chế số như BPSK hay QPSK. Do phổ của tín hiệu số là nghịch đảo của độ rộng ký hiệu, phổ của chuỗi chip phải rộng hơn phổ tín hiệu số nên trong trường hợp này yêu cầu điều chế trực tiếp. Với mục đích là độ rộng chip phải nhỏ hơn giá trị trung bình hoặc giá trị RMS nên kênh truyền phải là kênh chọn tần. Vì vậy, khi có hiện tượng fading đa đường, dãy chip c có được từ LOS và bản sao bị trễ của nó lệch nhau lớn hơn một độ rộng chip (được gọi là **lag**). Điều này sẽ được sử dụng để tạo ra thiết kế ở mức cao hơn dãy chip: **dãy giả ngẫu nhiên**.

Ví dụ về trải phổ dãy trực tiếp được chỉ ra trên hình 4.8. Trong trường hợp này tồn tại cả đường LOS và đường phản xạ, lag tương ứng với ba độ rộng chip. Dãy chip LOS trực tiếp là

$c = c_1 c_2 \dots c_n$ nên ta có $c^{-1} = \bar{c}_1 \bar{c}_2 \dots \bar{c}_n$, trong khi dãy chip từ đường phản xạ bắt đầu trễ mất ba độ rộng chip. Như vậy, máy thu sẽ hoạt động như sau: giả sử máy thu đồng bộ với đường trực tiếp. Nó so sánh dãy chip đến với dãy chuẩn đã biết c bằng cách tính tích nội. Nếu dãy nhận được giống c thì hoạt động này tạo ra giá trị n , nếu dãy chip nhận được là c^{-1} thì kết quả là $-n$. Các bản sao trễ làm méo tín hiệu trực tiếp giống cách của mô hình AWGN. Do đó, DSSS tăng tính bền vững để chống lại các hiệu ứng đa đường. Tuy nhiên, nó có một số bất buộc sau. *Thứ nhất* là các máy thu phải đồng bộ hoàn toàn với các máy phát. *Thứ hai* là phải có sự quản lý dãy chip. Trong các hệ thống như IEEE 802.11 với lớp vật lý DSSS (PHY) hoặc IEEE 802.15.4, chỉ có dãy chip đơn là được sử dụng bởi tất cả các nút. Thực hiện các phép đo ở lớp MAC để tránh xung đột. Có thể ấn định các mã hoặc dãy chip khác nhau đến những nơi khác nhau bằng cách truyền song song và chỉ tạo ra các méo không quan trọng. Phương pháp này được gọi là **đa truy cập phân chia theo mã** (CDMA - Code Division Multiple Access). Tuy nhiên, câu hỏi được đặt ra ở đây là có thể có bao nhiêu mã được ấn định đến các nút?



Hình 4.8 Ví dụ về trải phổ dãy trực tiếp

Trải phổ nhảy tần

Trong **trải phổ nhảy tần** (FHSS), các hệ thống như Bluetooth, phiên bản FHSS của IEEE 802.11, phổ được chia nhỏ thành các kênh hoặc băng con có kích thước bằng nhau (không lẫn với các kênh vật lý đã nói ở phần trên); Bluetooth và IEEE 802.11 chia phổ của chúng trong dải 2.4 GHz thành 78 băng con có độ rộng 1 MHz. Dữ liệu luôn được truyền trong một kênh tại một thời điểm; do đó, băng thông của nó cũng bị hạn chế. Tất cả các nút trong mạng nhảy một cách đồng bộ giữa các kênh theo lịch trình đã định trước. Theo cách đó, nếu kênh hiện tại đang bị suy giảm sâu ở một số điểm tại một thời điểm nào đó thì các nút sẽ nhảy qua kênh khác tốt hơn. Các mạng khác nhau có thể cùng chia sẻ một khu vực địa lý bằng cách sử dụng lịch trình nhảy không chồng chéo nhau.

Ví dụ, phiên bản FHSS của IEEE 802.11 nhảy với 2.5 Hz và nhiều gói tin có thể truyền đi trước khi có bước nhảy tiếp theo. Trong Bluetooth, tần số nhảy là 1.6 kHz và chỉ có một gói tin được truyền đi trước khi có bước nhảy tiếp theo. Các gói tin có độ dài tương ứng với một, ba hoặc năm bước nhảy. Khi gói tin dài hơn, các bước nhảy sẽ bị chặn lại và gói tin sẽ được truyền ở cùng một tần số. Khi gói tin đã được truyền đi, hệ thống sẽ tiếp tục ở tần số nó đang có nếu gói tin dài không xuất hiện.

4.2.6 Đồng bộ và truyền gói tin

Lớp MAC ở trên lớp vật lý sử dụng các gói hoặc khung như là các đơn vị cơ bản cho việc truyền tin.

Từ cách nhìn của lớp MAC, khung như vậy có kiến trúc. Tuy nhiên, đối với máy thu thì nó chỉ là một khối các bit. Các bộ thu phát thường thực hiện các chức năng điều chế và giải điều chế trong suốt quá trình xử lý ở tần số trung và cao ở phần cứng để tạo ra giao diện với lớp vật lý.

Bộ thu phải biết các tính chất của sóng đến để có thể cảm nhận nó và tìm kiếm khung. Nó bao gồm tần số, pha, các ký hiệu/bit bắt đầu và kết thúc, các khung bắt đầu và kết thúc. Nguồn gốc của vấn đề đồng bộ là gì? Việc tạo các sóng mang hình sin và clock cục bộ (tương ứng với các thời điểm ký hiệu được mô tả) cần phải có các **máy tạo dao động** (oscillator) của một **tần số danh định** (nominal frequency) nào đó. Tuy nhiên, do sản phẩm không chính xác, do sự chênh lệch nhiệt độ và một số lý do khác, tần số thực tế của các máy tạo dao động bị lệch so với tần số danh định. Độ lệch này thường có giá trị khoảng **vài phần triệu** (ppm - parts per million) và số các dao động clock thêm vào hoặc mất đi tạo ra khoảng thời gian cần cho một triệu dao động ở tốc độ danh định. Máy tạo dao động càng rẻ thì độ lệch càng lớn.

Để bù độ lệch này, máy thu phải biết tần số và cơ sở thời gian của máy phát. Máy thu phải có thông tin đồng bộ từ dạng sóng nhận được. Vấn đề thường thấy của phương pháp này là sự phân biệt giữa **pha huấn luyện** (training) hay **theo dõi** (acquisition) và **tìm kiếm** (tracking). Các khung được trang bị một dãy huấn luyện biết trước để cho phép máy thu biết các tham số chi tiết của máy phát, ví dụ như tốc độ clock. Dãy huấn luyện này thường được đặt ở đầu khung (ví dụ như trong chuẩn IEEE 802.11 hoặc IEEE 802.15.4), nhưng đôi khi nó được đặt ở giữa (ví dụ như trong GSM). Trường hợp thứ nhất, ta gọi nó là **phần mở đầu** (preamble), còn trong trường hợp thứ hai là **phần ở giữa**. Trong cả hai trường hợp, dãy huấn luyện đều có phần mở đầu. Ví dụ như trong IEEE 802.15.4, phần mở đầu bao gồm 32 bit 0 (zero).

Sau khi máy thu nhận thành công tín hiệu đồng bộ ban đầu từ dãy huấn luyện, nó sẽ chuyển sang chế độ tìm kiếm, tiếp tục điều chỉnh lại máy tạo dao động cục bộ của nó.

Vấn đề đồng bộ bao gồm:

- **Đồng bộ thiết bị truyền tải:**

Máy thu phải biết được tần số, các sơ đồ tìm kiếm và pha của tín hiệu. Độ lệch tần có thể do các máy tạo dao động hoặc do dịch tần Doppler trong trường hợp các nút di chuyển. Có một cách để có được sự đồng bộ tần số, đó là cho máy phát gửi các gói tin dao động theo dạng phổ đã biết và cho máy thu quét một số vị trí phổ quanh dải tần danh định của dạng phổ này. Thông thường, pha biến đổi nhanh hơn tần số, do đó, đồng bộ pha phải được thực hiện thường xuyên hơn đồng bộ tần số. Trong các sơ đồ tìm kiếm không nhất quán có thể không cần thực hiện đồng bộ pha nhưng cái giá để có được BER cao hơn khi so với cùng một công suất phát.

- **Đồng bộ bit/ký hiệu**

Khi có yêu cầu đồng bộ thiết bị truyền tải, máy thu phải xác định được cả độ rộng ký hiệu, các ký hiệu bắt đầu và kết thúc để quá trình giải điều chế thành công. Quá trình hiệu chỉnh lại tiếp tục ở pha tìm kiếm yêu cầu phải có “kích thước” phù hợp để biểu thị giới hạn của ký hiệu. Điều này có thể giải thích bằng ví dụ của OOK, trong đó các giá trị 0 được điều chế như là sự vắng mặt của một sóng mang nào đó. Nếu có nhiều số 0 liên tục, clock của máy thu không có sự điều chỉnh thì có thể nó sẽ bị lệch so với clock máy phát, điều này có thể gây ra hiện tượng thêm các ký hiệu giả hoặc nhảy ký hiệu. Ví dụ, với bộ thu phát RF Monolithics

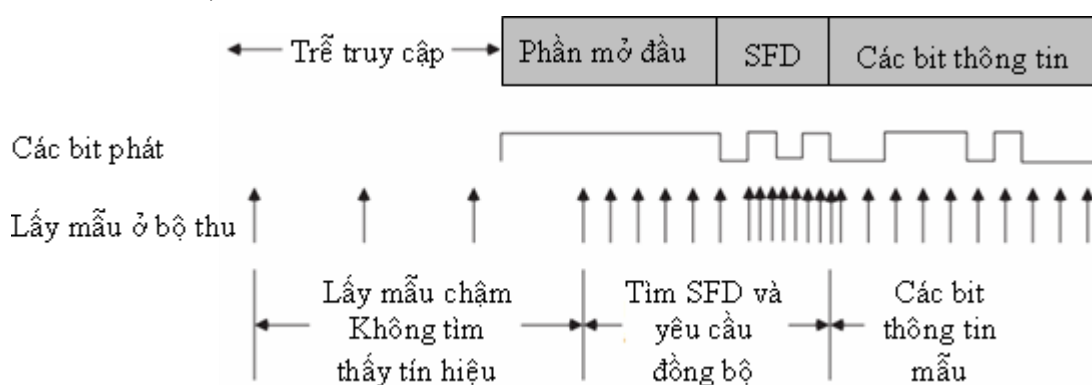
TR1000 sử dụng trong các mote Mica, phải tránh hiện tượng có nhiều hơn 4 bit 0 hoặc bit 1 liên tiếp. Có thể tránh tình huống này bằng cách chọn sơ đồ mã hoá có số bit 1 phù hợp như kỹ thuật chèn hoặc xáo trộn bit, trong đó luồng dữ liệu sẽ được dịch qua một thanh ghi dịch phản hồi tuyến tính. Kỹ thuật xáo trộn bit có thể được dùng trong IEEE 802.11 và không có ký hiệu thêm nào được gửi đi.

• **Đồng bộ khung:**

Máy thu phải tìm kiếm từ khi khung bắt đầu cho đến khi nó kết thúc (biên giới của khung). Đồng bộ khung thừa nhận rằng đồng bộ bit/ ký hiệu luôn được yêu cầu. Có một số kỹ thuật cho đồng bộ khung như khe trống thời gian, các trường độ dài, sử dụng các dãy flag đặc biệt trong kỹ thuật chèn bit để tránh xung đột của các dãy này trong gói dữ liệu hoặc giữa các gói dữ liệu với nhau.

Kỹ thuật đánh dấu điểm bắt đầu của khung là phương pháp của IEEE 802.15.4, trong đó phần mở đầu ngay lập tức thực hiện theo “**bắt đầu dấu tách khung**” (SFD - Start Frame Delimiter). SFD là một phần trong phần đầu của lớp vật lý, nó không thuộc phần dữ liệu.

Hình 4.9 là một ví dụ về lấy mẫu và đồng bộ. Trong các mote Mica, một trong những lựa chọn cho điều chế là OOK. Trong đó, các bit biểu diễn hai mức của công suất truyền tải: mức công suất 0 tương ứng với giá trị logic 0 và mức công suất khác 0 tương ứng với giá trị logic 1 (bỏ qua nhiễu nền). Một gói tin sẽ bao gồm một phần mở đầu, SFD và phần dữ liệu. Một khoảng thời gian nghỉ ở đó sẽ được hiểu như là ranh giới giữa các gói tin. Trong khoảng thời gian đó, máy thu chỉ cần thỉnh thoảng lấy mẫu. Thời gian giữa các mẫu phải nhỏ hơn độ dài phần mở đầu để không bị bỏ qua nó nhưng phải đủ lớn để không tốn quá nhiều năng lượng cho việc lấy mẫu. Khi mẫu cho thấy hoạt động trong kênh, tần số của nó phải tăng lên để tìm điểm kết thúc của phần mở đầu và xác định độ dài của một bit truyền từ SFD. Một khi thông tin này được xác định, các mẫu sẽ nằm giữa các bit dữ liệu. Để tránh sự có mặt của thời gian rỗi trong phần dữ liệu và hiểu sai nó như là ranh giới của gói, số bit 0 (và 1) liên tiếp phải bị hạn chế (ví dụ như chỉ dưới 4 bit).



Hình 4.9 Ví dụ về lấy mẫu và đồng bộ

4.2.7 Chất lượng kênh vô tuyến và các phép đo độ cải thiện

Ngược với các kênh hữu tuyến, các kênh vô tuyến thường có tốc độ lỗi bit/ ký hiệu kém hơn. Chất lượng kênh thực tế phụ thuộc vào nhiều yếu tố như tần số, khoảng cách giữa bộ phát và bộ thu, tốc độ chuyển động tương đối giữa chúng, môi trường truyền sóng... Có nhiều cách để cải thiện chất lượng kênh vô tuyến làm việc trong môi trường truyền sóng và ở các lớp cao

hơn:

- **Tối ưu hoá các tham số truyền dẫn:**

Lựa chọn cơ chế điều chế cũng như công suất phát có thể cải thiện BER một cách đáng kể. Mẫu chốt giám sát quan trọng khác là lựa chọn kích thước và cấu trúc của gói tin.

- **Cơ chế phân tập:**

Tất cả các kỹ thuật phân tập hình thành và sử dụng độc lập với nhau đối với cùng một tín hiệu. Nói một cách đơn giản, người ta hy vọng rằng, nếu một mô hình đang ở trạng thái suy giảm sâu và có lỗi ký hiệu thì có thể được thay bằng một mô hình khác tốt hơn. Bộ thu sẽ cố gắng lựa chọn tất cả các mô hình tốt nhất. Trong các cơ chế phân tập hiện nay, rất nhiều bản sao được tạo bởi máy phát bằng cách gửi cùng một gói tin ở các tần số khác nhau qua các khe thời gian khác nhau hoặc gửi nó theo các hướng khác nhau trong không gian. Trong các cơ chế phân tập trước đây, tín hiệu chỉ được gửi một lần nhưng nhiều bản copy được tạo ra trong kênh truyền do truyền sóng đa đường. Trong cả hai trường hợp, máy thu cần phải có cơ chế để lựa chọn ưu điểm của các bản copy.

- **Sự hiệu chỉnh:**

Kỹ thuật hiệu chỉnh được sử dụng để chống lại nhiễu liên ký hiệu (ISI). Hiệu chỉnh được thực hiện như sau: máy phát gửi dạng sóng/ mẫu ký hiệu đã biết, còn được gọi là **dãy huấn luyện**. Bộ hiệu chỉnh ở máy thu làm việc ở hai chế độ là **huấn luyện** và **tìm kiếm**. Ở pha huấn luyện, bộ hiệu chỉnh phân tích bản thu được của mẫu đã có để biết cơ chế của méo, tính algorithm cho “ngịch đảo” của méo. Ở pha tìm kiếm, gói tin còn lại được phân tích bằng cách áp dụng thuật toán ngược lại với nó để bộ điều chỉnh tiếp tục điều chỉnh lại thuật toán ngược. Sự hiệu chỉnh yêu cầu một số xử lý tín hiệu ở bộ thu và giả sử kênh đứng yên trong thời gian gói tin được truyền. Dây huấn luyện còn có thể được sử dụng để **đồng bộ bit**.

- **Hiệu chỉnh phòng lỗi (FEC - Forward error correction):**

Bộ phát chấp nhận một luồng, khối dữ liệu hoặc nguồn bit, thêm vào các bit dư phù hợp và truyền chúng tới bộ thu. Phụ thuộc vào số lượng và cấu trúc của bit dư mà bộ thu có thể hiệu chỉnh một số lỗi bit/ ký hiệu. Ta đã biết rằng các kênh AWGN có công suất cao hơn các kênh fading Rayleigh và nhiễu sơ đồ mã hoá có đặc tính BER tốt hơn các kênh fading có lỗi chùm. Hoạt động của **kỹ thuật đan xen** áp dụng thuật toán hoán vị trong các khối bit để hy vọng sự phân bố các lỗi chùm đều hơn và xem các kênh truyền như là các kênh AWGN.

ARQ: Ý tưởng cơ bản của các giao thức ARQ được mô tả như sau: Bộ phát sử dụng một header và gắn một kiểm tra tổng và khối dữ liệu. Gói tin này sau đó sẽ được truyền đi. Bộ thu kiểm tra tính toàn vẹn của gói tin với sự trợ giúp của kiểm tra tổng và cung cấp một số thông tin phản hồi cho bộ phát về sự thành công của việc truyền dẫn gói tin. Trên đường phản hồi âm nhận được, bộ phát thực hiện việc tái truyền dẫn.

4.3 Lớp vật lý và thiết kế bộ thu phát trong WSN.

Một số điểm quyết định ảnh hưởng đến thiết kế PHY trong WSN là:

- Tiêu thụ năng lượng thấp.
- Công suất phát nhỏ sẽ tạo ra dải truyền dẫn nhỏ. Điều này sẽ tạo ra một chu trình hoạt động có công suất thấp. Đa số phần cứng phải tắt hoặc hoạt động ở chế độ standby công suất thấp trong hầu hết thời gian sử dụng.

- Tốc độ dữ liệu tương đối thấp, yêu cầu khoảng vài chục đến vài trăm kbit/s.
- Giá thành và tính phức tạp trong quá trình thực hiện thấp.
- Mức độ thay đổi thấp...

Thông thường, trong các mạng cảm biến, rất khó để tìm ra sơ đồ điều chế và cấu trúc bộ thu phát đơn giản, giá rẻ nhưng vẫn đủ mạnh để tạo cung cấp dịch vụ mong muốn.

4.3.1 Sử dụng năng lượng

Lựa chọn công suất phát thấp để tiết kiệm năng lượng trong mạng cảm biến không dây khác với các thiết bị vô tuyến khác.

Thứ nhất là, công suất phát nhỏ, thường có bậc bằng 0 dBm (tương ứng với 1 mW). Mặt khác, tổng thu phát (đầu cuối RF và phần băng gốc) tiêu thụ năng lượng nhiều hơn công suất phát thực tế. Wang et al. ước tính bộ thu phát làm việc ở tần số vượt quá 1 GHz cần công suất 10 đến 100 mW để tạo ra 1 mW. Số liệu tương tự cũng được cho bởi các bộ thu phát CMOS 2.4 GHz: Đối với công suất phát 0 dBm, bộ phát thực tế sử dụng 32 mW trong khi bộ thu sử dụng nhiều hơn, 38 mW. Đối với các mote Mica, chế độ phát sử dụng 21 mW và chế độ thu sử dụng 15 mW.

Thứ hai là, công suất phát nhỏ thì các chế độ thu phát tiêu thụ nhiều hoặc ít năng lượng như nhau, thậm chí quá trình thu sử dụng nhiều năng lượng hơn quá trình phát. Phụ thuộc vào cấu trúc bộ thu phát mà tiêu thụ năng lượng ở chế độ nghỉ có thể ít hơn hoặc bằng công suất thu. Để giảm năng lượng tiêu thụ trung bình trong WSN lưu lượng thấp, phải giữ bộ thu phát ở chế độ nghỉ trong tất cả thời gian. Vì vậy, điều quan trọng là phải đặt bộ thu phát ở trạng thái ngủ thay vì chỉ ở trạng thái nghỉ.

Tuy nhiên, vấn đề đặt ra ở đây là **thời gian/ năng lượng khởi động** mà bộ thu phát sử dụng để chuyển từ trạng thái ngủ sang trạng thái hoạt động (thức). Trong thời gian khởi động, không có việc truyền và nhận dữ liệu. Ví dụ bộ thu phát μ AMPS-1 cần thời gian khởi động là 466 μ s và tiêu thụ công suất là 58 mW. Vì vậy, chuyển sang trạng thái ngủ sẽ không có lợi khi trạng thái hoạt động sắp xảy ra. Phụ thuộc vào lưu lượng và hành vi của các giao thức MAC để tạo ra kế hoạch hoạt động cho bộ thu phát. Nếu có thể, không chỉ một mà nhiều gói tin được gửi đi ở chế độ thức để chia giá thành khởi động cho nhiều gói tin.

Thứ ba là, giá thành của việc truyền dẫn so với tính toán trên một nút cảm biến. Rõ ràng, giá thành này phụ thuộc vào phần giao tiếp dựa trên yêu cầu của BER, dải hoạt động, loại bộ thu phát... Phần tính toán lại phụ thuộc vào loại bộ xử lý, số câu lệnh... Ví dụ, với các nút Rockwell's WIN, có 1500 đến 2700 câu lệnh được thực hiện trên một bit truyền; với các nút MEDUSA II, tỉ số này nằm trong khoảng từ 220:1 đến 2900:1; với các nút WINS NG, tỉ số này vào khoảng 1400:1. Như vậy, tính toán rẻ hơn truyền dẫn.

4.3.2 Lựa chọn sơ đồ điều chế

Các nhân tố sau phải được coi trọng như nhau khi lựa chọn sơ đồ điều chế: tốc độ dữ liệu/ tốc độ ký hiệu yêu cầu và mong muốn, độ phức tạp thực hiện, mối quan hệ giữa công suất phát và BER và các đặc tính mong đợi của kênh truyền. Thời gian bộ thu phát ở chế độ ngủ càng nhiều thì thời gian truyền càng ít. Tốc độ dữ liệu của bộ thu phát/ điều chế càng cao thì thời gian yêu cầu để truyền dữ liệu đã cho càng thấp, do đó năng

lượng tiêu thụ càng ít. Điểm quan trọng thứ hai là tiêu thụ năng lượng của sơ đồ điều chế phụ thuộc vào tốc độ ký hiệu nhiều hơn tốc độ dữ liệu. Ví dụ, các phép đo năng lượng tiêu thụ của WLAN (Wireless Local Area Network) IEEE 802.11b chỉ ra rằng năng lượng tiêu thụ phụ thuộc vào sơ đồ điều chế, **điều chế khoá mã bù** (CCK - Complementary Code Keying) càng nhanh thì tiêu thụ năng lượng càng nhiều so với DBPSK và DQPSK. Tuy nhiên, sự chênh lệch sẽ dưới 10% và tất cả các sơ đồ này có cùng tốc độ ký hiệu.

4.3.3 Antenna

Một đặc điểm cần chú ý của tất cả các nút cảm biến là hạn chế về kích thước và số lượng antenna. Nếu antenne nhỏ hơn rất nhiều độ dài sóng mang thì rất khó đạt được hiệu suất cao, do đó, những antenna không phù hợp kích thước phải sử dụng nhiều năng lượng truyền hơn để có được cùng năng lượng bức xạ.

Thứ hai là, với trường hợp nút cảm biến nhỏ, rất khó để đặt hai antenna có khoảng cách phù hợp để có được phân tập thu. Trong phần 2.4.7 đã nói, các antenna phải cách nhau ít nhất là 40-50% độ dài bước sóng để có hiệu quả tốt từ phân tập. Ở dải 2.4 GHz, khoảng cách này sẽ nằm trong khoảng 5 đến 6 cm.

Thêm vào đó, sóng vô tuyến phát ra từ các antenna gần đất, trong một số ứng dụng, sẽ có các hệ số suy hao đường truyền cao hơn, giá trị thường thấy là $\alpha=2$ cho truyền thông trong không gian. Hệ số đó trong các môi trường có chướng ngại vật (như các toà nhà, tường...) là 4.

Hơn nữa, phụ thuộc vào ứng dụng mà antenna không được nhô lên khỏi vỏ bọc của nút để tránh bị hỏng. Thông thường, hạn chế này sẽ làm giảm chất lượng và các đặc tính của antenna trong các nút cảm biến không dây.

Ví dụ, các nút được phân bố rải rác trên mặt đất sẽ nằm ở các vị trí ngẫu nhiên, antenna của chúng sẽ tiếp xúc với đất hoặc các vật cản khác. Điều này sẽ làm cho việc truyền sóng vô tuyến **không đẳng hướng**, tức cường độ tín hiệu phát theo các hướng khác nhau là khác nhau. Điều này cũng có thể do thiết kế của antenna, tạo ra các khác biệt lớn trong các đặc tính truyền sóng không gian (còn được gọi là các vấu của antenna).

CÂU HỎI ÔN TẬP CHƯƠNG 4

1. Dải tần làm việc của kênh vô tuyến
2. Các đặc điểm của điều chế trong bộ thu phát của WSN
3. Các ảnh hưởng của truyền sóng
4. Suy hao đường truyền và sự suy giảm trong WSN
5. Các loại tạp âm và nhiễu trong WSN?
6. Các mô hình kênh của WSN
7. Phân biệt trải phổ dải trực tiếp (DSSS) và trải phổ nhảy tần (FHSS)
8. Vấn đề đồng bộ trong truyền tin của lớp vật lý.
9. Các phép đo cải thiện chất lượng kênh vô tuyến
10. Các lưu ý khi thiết kế bộ thu phát trong WSN.

CHƯƠNG 5. CÁC GIAO THỨC MAC

Mục đích của chương

Giao thức MAC (Medium Access Control – Điều khiển truy nhập môi trường) giải quyết một nhiệm vụ đơn giản là sắp xếp thời gian khi có một số nút cùng truy nhập vào một mạng thông tin môi trường có chia sẻ. Chương này giới thiệu cơ sở giao thức MAC, giải thích một số yêu cầu cụ thể và vấn đề mà MAC gặp trong mạng cảm biến không dây. Yêu cầu quan trọng nhất là hiệu quả năng lượng và việc sử dụng các nguồn MAC cụ thể cho các việc khác nhau. Chương này còn thảo luận một số cơ chế đánh địa chỉ, phương pháp quan trọng để bật bộ phát không dây ở chế độ ngủ. Do đó, cần phải cân nhắc giữa việc sử dụng năng lượng của mạng cảm biến và một số đặc tính thông thường như trễ và công suất khi sử dụng mạng cảm biến không dây.

Giao thức MAC là lớp giao thức đầu tiên của lớp vật lý (Physical Layer – PHY) do đó giao thức MAC có ảnh hưởng lớn nhờ các tính chất của nó. Nhiệm vụ cơ bản của giao thức MAC là điều chỉnh việc truy nhập của các nút đến mạng môi trường có chia sẻ theo cách là phải thoả mãn được các yêu cầu về đặc tính trong từng ứng dụng cụ thể. Các thông số như độ trễ, công suất... thường được coi trọng nhưng ngược lại, trong mạng cảm biến không dây (WSN – Wireless Sensor Network) thì sự bảo toàn năng lượng lại đóng vai trò quan trọng.

Trong mô hình OSI chuẩn thì MAC được xem là một phần của lớp liên kết dữ liệu (DLL – Data Link Layer) nhưng có sự phân chia rõ ràng về công việc giữa MAC và phần còn lại của DLL. Giao thức MAC xác định nút tại một thời điểm xác định khi truy nhập vào môi trường để truyền dữ liệu, điều khiển hay quản lý đến nút khác (truyền thông đơn hướng) hay để đặt các nút (trong truyền thông đa hướng). Hai thông số quan trọng trong phần còn lại của DLL là điều khiển sai số và điều khiển lưu lượng. Điều khiển sai số được sử dụng để đảm bảo độ tin cậy trong quá trình truyền và để có hành động phù hợp khi có sai số truyền. Điều khiển lưu lượng làm nhiệm vụ điều chỉnh tốc độ truyền để bảo vệ máy thu tốc độ chậm không bị tràn dữ liệu. Lớp liên kết sẽ được trình bày cụ thể trong chương 6.

5.1. Các nguyên tắc cơ bản của các giao thức MAC (không dây)

Phần này giới thiệu một số nguyên tắc cơ bản và một số ví dụ quan trọng của giao thức MAC không dây do các giao thức được sử dụng trong WSN kế thừa nhiều đặc điểm đã tồn tại trong các mạng thông thường.

5.1.1 Các yêu cầu và ràng buộc thiết kế cho các giao thức MAC không dây.

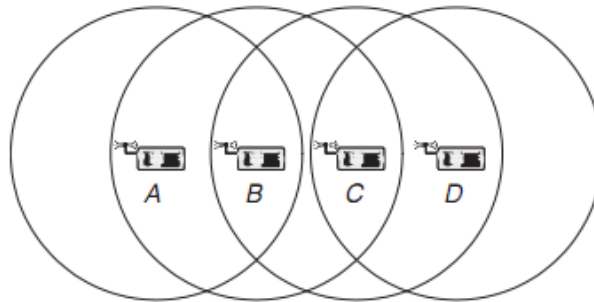
Thông thường, yêu cầu kỹ thuật quan trọng nhất của các giao thức MAC là hiệu suất thông lượng, độ ổn định, tính công bằng, trễ truy nhập thấp (khoảng thời gian khi gói thông tin đến và lần đầu truyền nó), trễ truyền dẫn thấp (khoảng thời gian khi gói thông tin đến và truyền nó thành công) và tổng chi phí thấp. Tổng chi phí trong các giao thức MAC có thể được tính trên tổng chi phí cho mỗi gói (từ phần đầu đến phần cuối của MAC), các xung đột hay sự trao đổi của các gói điều khiển bên ngoài. Các xung đột có thể xảy ra nếu giao thức cho phép hai hay nhiều nút gửi gói tin ở cùng một thời điểm. Các xung đột có thể khiến cho máy thu không có khả năng giải mã gói tin một cách đúng đắn nên các lớp trên cùng phải thực hiện việc truyền lại. Với các ứng dụng đòi hỏi chặt chẽ về thời gian thì điều quan trọng là đảm bảo tính tiên định hoặc ngẫu nhiên về thời gian phân phối hay tốc độ dữ liệu tối thiểu sẵn có.

Đôi khi cần thiết phải đối xử ưu tiên với các gói tin quan trọng hơn các gói tin không quan trọng nên ở đây có khái niệm về *tính ưu tiên*.

Hoạt động và đặc tính của các giao thức MAC chịu ảnh hưởng lớn bởi các tính chất của lớp vật lý bên dưới. Do WSN sử dụng môi trường vô tuyến nên nó kế thừa tất cả các vấn đề đã biết của truyền sóng vô tuyến. Một trong những vấn đề đó là biến đổi theo thời gian và đôi khi, tốc độ lỗi thực sự cao. Đó là do các hiện tượng vật lý như fading chậm và nhanh, suy hao đường truyền, suy giảm hay tạp âm nhiệt. Phụ thuộc vào sơ đồ điều chế, tần số, khoảng cách giữa bộ phát và bộ thu, môi trường truyền sóng mà có thể dễ dàng quan sát được tốc độ lỗi bit tức thời ở dải $10^{-3} \dots 10^{-2}$.

Như đã giải thích trong chương 4, công suất thu P_{rcvd} giảm theo khoảng cách giữa nút phát và thu. Suy hao đường truyền này sẽ kết hợp với việc bộ thu phát cần cường độ tín hiệu nhỏ nhất để giải điều chế các tín hiệu thành công sẽ tạo ra một dải lớn nhất mà nút cảm biến có thể có được đối với một công suất phát đã cho. Nếu hai nút nằm ở ngoài dải này thì chúng không thể nghe thấy nhau. Đây chính là nguồn gốc của các vấn đề về đầu cuối ẩn/ đầu cuối hiện đã biết. **Vấn đề đầu cuối ẩn** (hidden-terminal problem) thường xảy ra đối với lớp các giao thức **đa truy cập cảm nhận sóng mang** (CSMA - Carrier Sense Multiple Access), ở đó nút cảm nhận môi trường trước khi nó truyền dữ liệu. Nếu môi trường bận, nút sẽ hoãn việc truyền lại để tránh xung đột và sẽ truyền lại vào lúc khác.

Xem xét ví dụ trên hình 5.1. Ở đây ta có ba nút A , B và C được sắp xếp như sau: A và B cùng dải, B và C cùng dải nhưng A và C không thể nghe lẫn nhau. Giả sử A bắt đầu truyền tin đến B và một lúc sau, C cũng quyết định truyền tin. Hoạt động cảm nhận sóng mang ở C cho biết môi trường rỗi vì C không thể nghe các tín hiệu từ A . Khi C bắt đầu truyền tin thì ở B xảy ra sự xung đột của hai gói tin và cả hai gói tin đều không sử dụng được. Bằng cách sử dụng CSMA đơn giản ở đầu cuối ẩn sẽ tránh được các xung đột không mong muốn này.



Hình 5.1 Tình huống đầu cuối ẩn (các vòng tròn là dải truyền dẫn và nhiễu)

Trong tình huống đầu cuối hiện, B truyền tin cho A và một lúc sau, C muốn truyền tin đến D . Mặc dù điều này về mặt lý thuyết là có thể do cả A và D sẽ nhận các gói tin không có méo, hoạt động cảm nhận sóng mang được thực hiện bởi C sẽ ngăn cản việc truyền của C và dải thông sẽ bị bỏ phí. Sử dụng CSMA đơn giản trong đầu cuối hiện sẽ tránh được việc đợi chờ không cần thiết.

Hai giải pháp cho các vấn đề của đầu cuối ẩn và đầu cuối hiện là giải pháp âm báo bận trong các giao thức MACA/ MACAW và bắt tay RTS/CTS được sử dụng trong chuẩn IEEE 802.11 WLAN.

Trong truyền thông hữu tuyến, CSMA được sử dụng cho **bộ phát** để phát hiện xung đột tức thời ở **bộ thu** và để hủy việc truyền tin. Đặc điểm này còn được gọi là **phát hiện xung đột** (CD - Collision Detection) và nó được sử dụng trong giao thức CSMA/CD của Ethernet để tăng hiệu suất truyền dẫn. Việc phát hiện xung đột thực hiện được là vì suy giảm trong môi trường dây dẫn thấp, tạo nên các SNR tương tự nhau ở bộ phát và bộ thu. Do đó, khi bộ phát đọc ngược lại tín hiệu kênh trong khi truyền dẫn và quan sát xung đột, nó có thể luận ra rằng cũng có xung đột ở bộ thu. Điều quan trọng hơn, nếu không có xung đột ở bộ phát thì có thể kết luận rằng không có xung đột ở bộ thu khi truyền tin. Trong môi trường vô tuyến, nhiều ở bộ phát không cho biết nhiều thông tin về nhiễu ở bộ thu. Hơn nữa, các bộ thu phát vô tuyến đơn giản chỉ làm việc ở chế độ bán song công. Có nghĩa là tại một thời điểm nhất định, chỉ có mạch phát hoặc mạch thu được kích hoạt. Vì vậy, các giao thức phát hiện xung đột thường không được sử dụng đối với các phương tiện vô tuyến.

Vấn đề quan trọng khác nảy sinh khi không có băng tần chuyên dụng cho WSN và WSN phải chia sẻ phổ của nó với các hệ thống khác. Do được hoạt động tự do nên nhiều hệ thống vô tuyến sử dụng băng ISM, như trong ví dụ là 2.4 GHz. Một số hệ thống sử dụng băng chuyên dụng này, ví dụ như IEEE 802.11/IEEE 802.11b WLANs, Bluetooth, và IEEE 802.15.4 WPAN, vì vậy xuất hiện khái niệm cùng tồn tại của các hệ thống này.

Cuối cùng, thiết kế các giao thức MAC phụ thuộc vào mô hình tải. Nếu WSN được sử dụng để quan sát liên tục một hiện tượng vật lý, ví dụ như phân bố nhiệt độ theo thời gian trong một khu rừng thì người ta quan tâm đến tính tuần hoàn của phép đo. Nếu với mục đích là chờ đợi sự xuất hiện của một hiện tượng quan trọng và khi sự kiện đó xảy ra sẽ ghi lại dữ liệu càng nhiều càng tốt thì mạng sẽ làm việc ở chế độ nghỉ trong phần lớn thời gian hoạt động và sau đó, một lượng lớn gói tin sẽ được chuyển đi một cách nhanh chóng.

5.1.2 Phân loại các giao thức MAC

Việc phân loại các giao thức MAC (không dây) xuất hiện cách đây hơn 30 năm. Chúng có thể được phân thành các loại sau: các giao thức phân chia cố định, các giao thức phân chia theo yêu cầu và các giao thức truy cập ngẫu nhiên.

Các giao thức phân chia cố định

Trong lớp các giao thức này, các tài nguyên có sẵn được chia cho các nút sao cho việc phân chia tài nguyên là dài hạn, mỗi nút có thể sử dụng riêng tài nguyên của nó và do đó không có nguy cơ xung đột. Sử dụng trong thời gian dài có nghĩa là trong khoảng vài phút, vài giờ hoặc lâu hơn, ngược lại với sử dụng trong thời gian ngắn như phạm vi của burst dữ liệu, tương ứng với khoảng thời gian vài chục ms. Để đếm sự thay đổi trong cấu hình mạng, ví dụ như do một số nút hỏng hoặc một số mới được đưa vào sử dụng, nút di chuyển hoặc thay đổi trong mô hình tải, sơ đồ tín hiệu cần phải sử dụng các giao thức phân chia cố định để có thể bố trí lại nguồn tài nguyên cho các nút. Câu hỏi đặt ra ở đây là khả năng mở rộng của các giao thức này.

Các giao thức tiêu biểu của lớp này là TDMA, FDMA, CDMA và SDMA. **Đa truy cập phân chia theo thời gian** (TDMA - Time Division Multiple Access) chia nhỏ trục thời gian thành những siêu khung có độ dài cố định, mỗi siêu khung đó một lần nữa lại được chia nhỏ thành các khe thời gian. Mỗi khe thời gian này được gán riêng cho các nút, do đó nút có thể truyền dữ liệu trong khe thời gian này một cách tuần tự trong mỗi siêu khung. TDMA yêu

cầu đồng bộ về mặt thời gian một cách nghiêm ngặt giữa các nút để tránh hiện tượng chồng chéo tín hiệu trong các khe thời gian lân cận.

Trong **đa truy cập phân chia theo tần số** (FDMA - Frequency Division Multiple Access), dải tần hoạt động được chia thành các kênh con và chúng được gán cho các nút. Nút có thể truyền dữ liệu trên kênh của chúng. Sơ đồ này yêu cầu đồng bộ về mặt tần số, các bộ lọc có dải tương đối hẹp và bộ thu có thể điều chỉnh theo kênh được bộ phát sử dụng. Do đó, bộ thu FDMA thường phức tạp hơn bộ thu TDMA.

Trong **đa truy cập phân chia theo mã** (CDMA - Code Division Multiple Access), các nút trải tín hiệu của chúng trong một dải tần lớn hơn yêu cầu, sử dụng các mã khác nhau để phân chia việc truyền dẫn của chúng. Bộ thu biết mã được sử dụng bởi bộ phát, tất cả các truyền dẫn song song sử dụng các mã khác xuất hiện như tạp âm. Điều quan trọng trong CDMA là việc quản lý mã.

Cuối cùng là **đa truy cập phân chia không gian** (SDMA - Space Division Multiple Access), sự phân chia không gian của các nút được sử dụng để phân chia việc truyền dẫn của chúng. SDMA yêu cầu các mảng antenna, kỹ thuật xử lý tín hiệu phức tạp và đây không thể coi là một công nghệ cho các WSN.

Các giao thức phân chia theo yêu cầu

Trong giao thức phân chia theo yêu cầu, sự cấp phát riêng các tài nguyên cho nút được xây dựng trên cơ sở ngắn hạn, thường là khoảng thời gian của dữ liệu chum. Lớp các giao thức này có thể được chia thành các giao thức tập trung và phân bố. Trong giao thức điều khiển tập trung (như HIPERLAN/2, DQRUMA hay MASCARA...), các nút gửi yêu cầu về việc cấp phát băng thông đến nút trung tâm. Nút trung tâm có thể chấp nhận hoặc từ chối yêu cầu này. Khi yêu cầu được chấp nhận, xác nhận được truyền trở lại nút đã yêu cầu kèm theo mô tả về tài nguyên được cấp phát, ví dụ như số lượng và vị trí của các khe thời gian được ấn định trong hệ thống TDMA và khoảng thời gian cấp phát đó. Nút có thể sử dụng riêng những tài nguyên này. Việc đệ trình yêu cầu của các nút đến trạm trung tâm thường dựa trên cơ sở tranh chấp, sử dụng giao thức truy cập ngẫu nhiên trên kênh tín hiệu chuyên dụng. Các nút thường mang theo các yêu cầu trong các gói dữ liệu được truyền trong các khe thời gian riêng của chúng, do đó nó tránh được việc truyền riêng các gói tin mang thông tin yêu cầu.

Nút trung tâm phải luôn ở trạng thái bật và chịu trách nhiệm về việc phân bố tài nguyên. Việc phân bố lại tài nguyên được thực hiện như sau: khi nút không sử dụng khe thời gian của nó nữa, nút trung tâm có thể phân bố nó cho các nút khác. Do đó, các nút không cần phải gửi thêm các gói tin hủy bỏ phân bố. Nói tóm lại, nút trung tâm thực hiện rất nhiều hoạt động, nó phải luôn ở trạng thái thức và do đó, nó sử dụng nhiều năng lượng. Lớp giao thức này là lựa chọn tốt nếu có đủ các nút không bị giới hạn về mức năng lượng tiêu thụ và nhiệm vụ của nút trung tâm có thể đưa đến các nút này.

Một ví dụ về các giao thức phân chia theo yêu cầu *phân bố* là các giao thức truyền Token như IEEE 802.4 Token Bus. Yêu cầu để bắt đầu một quá trình quyền dẫn là việc nhận được một khung Token nhỏ đặc biệt. Đây là một phương pháp khác cũng nhằm truy cập vào môi trường chia sẻ, là phương pháp dùng Token điều khiển. Token này được chuyển từ nút này sang nút khác phụ thuộc vào quy tắc được định nghĩa và được tất cả các nút nối vào môi trường hiểu và tuân theo. Một nút bất kỳ chỉ có thể truyền dữ liệu khi đang nắm giữ thẻ

Token, sau khi đã truyền xong khung, nó chuyển Token sang nút kế tiếp để cho phép nút khác truy cập vào môi trường truyền. Tuần tự hoạt động như sau:

- + Đầu tiên, một **vòng logic** (logical ring) được thiết lập để liên kết tất cả các nút nối vào môi trường vật lý này, và một Token điều khiển được tạo ra.

- + Token điều khiển được chuyển từ nút này qua nút khác xung quanh vòng logic cho tới khi một nút đang đợi truyền lấy được quyền điều khiển.

- + Sau đó, nút đang đợi truyền sẽ truyền các khung của nó qua môi trường vật lý này. Sau khi kết thúc việc truyền, nó sẽ chuyển Token điều khiển đến một nút khác trong vòng logic.

Ngoài ra, kể từ khi bắt đầu thời điểm lưu thông Token, nút phải luôn nhận được Token để không phá vỡ vòng logic. Do đó, các bộ thu phát của nút phải được bật trong hầu hết thời gian hoạt động. Thêm vào đó, duy trì vòng logic sẽ đối mặt với việc thường xuyên thay đổi cấu hình mạng, đây không phải là một nhiệm vụ dễ dàng.

Các giao thức truy cập ngẫu nhiên

Trong trường hợp này, các nút không phối hợp với nhau, các giao thức hoạt động theo phương thức phân phối hoàn toàn. Giao thức truy cập ngẫu nhiên thường kết hợp với một yếu tố ngẫu nhiên, ví như như thời điểm nhận gói tin ngẫu nhiên, đặt bộ định thời tới các giá trị ngẫu nhiên... Giao thức truy cập ngẫu nhiên đầu tiên và quan trọng nhất là giao thức ALOHA hay slotted ALOHA, được phát triển ở trường đại học Hawaii. Ở giao thức ALOHA, khi một nút muốn truyền một gói tin mới thì sẽ truyền nó ngay lập tức. Không có sự phối hợp với các nút khác nên thường xảy ra xung đột ở bộ thu. Để phát hiện xung đột này, bộ thu được yêu cầu gửi thông tin tức thời về gói tin đã nhận được. Nếu không có thông tin phản hồi này, bộ phát sẽ hiểu là đã có xung đột, sau một khoảng thời gian chờ ngẫu nhiên nào đó nó sẽ bắt đầu quá trình truyền tiếp theo. ALOHA tạo ra truy cập ngắn và trễ truyền dẫn đối với các tải nhẹ. Đối với các tải nặng hơn, số xung đột sẽ tăng lên, điều này sẽ làm giảm hiệu suất và tăng thời gian trễ truyền dẫn. Ở slotted ALOHA, thời gian được chia thành các khe và nút chỉ được cho phép bắt đầu truyền tin khi bắt đầu một khe. Khe phải đủ lớn để có thể chứa được gói tin có độ dài cực đại. Theo đó, chỉ khi các bộ phát khác cũng bắt đầu truyền tin trong cùng khe thời gian mới có thể làm hỏng gói tin của nút. Nếu một nút nào đó muốn bắt đầu muộn hơn thì nó có thể đợi đến thời điểm bắt đầu khe thời gian tiếp theo và khi đó, không còn nguy cơ phá hủy gói tin của nút. Trong một thời điểm nào đó, sự đồng bộ sẽ làm giảm xác suất xung đột và slotted ALOHA có hiệu suất cao hơn ALOHA.

Trong lớp **giao thức CSMA**, nút truyền sẽ cố gắng để quá trình truyền dẫn được diễn ra liên tục. Trước hết, nút được yêu cầu nghe ngóng về môi trường, đây được gọi là **cảm nhận sóng mang** (carrier sensing). Nếu môi trường rỗi thì nút sẽ bắt đầu truyền. Nếu môi trường bận, nó sẽ hoãn việc truyền lại sau một khoảng thời gian nào đó. Ví dụ, trong **CSMA không liên tục (nonpersistent CSMA)**, nút sẽ đưa ra thời gian đợi ngẫu nhiên sau đó môi trường được cảm nhận lại. Trước khoảng thời gian này, nút không quan tâm đến trạng thái của môi trường. Trong các dạng **CSMA liên tục (persistent CSMA)** khác, sau khi cảm nhận được là môi trường bận, nút sẽ đợi đến khi kết thúc quá trình truyền dẫn và sau đó nó thực hiện theo **thuật toán chờ để truyền** (backoff algorithm). Trong nhiều thuật toán backoff, thời gian sau khi kết thúc khung trước được chia thành các khe thời gian. Với CSMA liên tục p , nút bắt đầu

truyền trong khe thời gian với xác suất p nào đó và nó đợi khe thời gian khác với xác suất $1-p$. Nếu một số nút khác cũng bắt đầu truyền tại thời điểm đó thì nút sẽ hoãn việc truyền và lặp lại quá trình trên cho đến khi kết thúc một khung mới. Một giá trị nhỏ của p làm cho không xảy ra xung đột nhưng lại làm tăng giá thành và trễ truy cập, p lớn có ý nghĩa ngược lại.

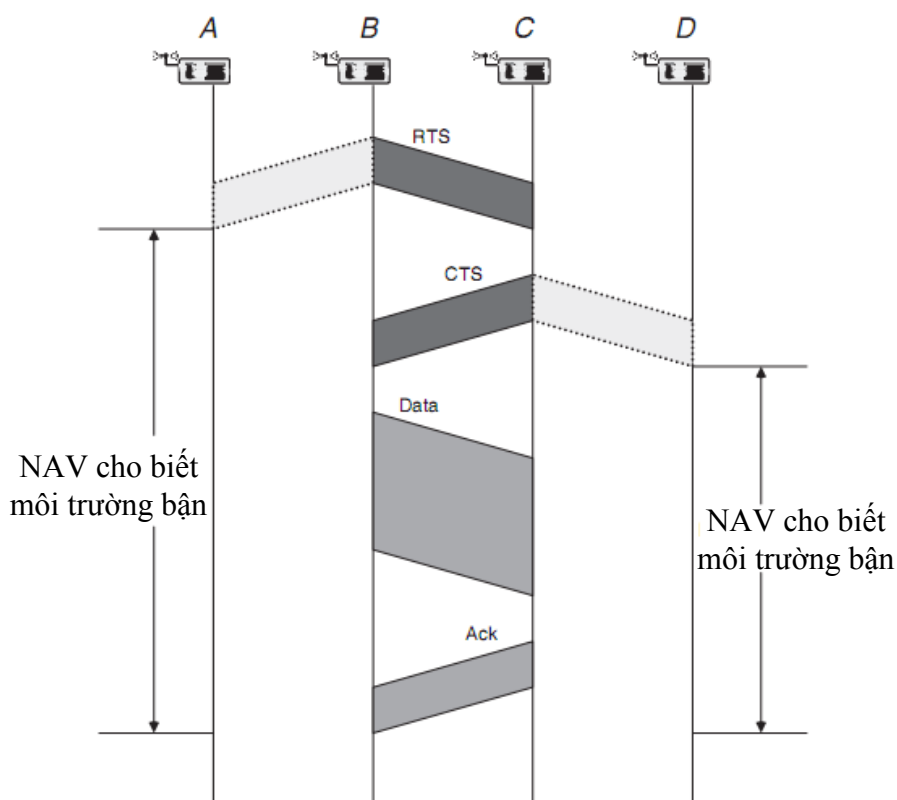
Trong thuật toán backoff được thực hiện bởi **hàm tọa độ phân bố** (DCE -) của IEEE 802.11, một nút truyền khung mới sẽ chọn một giá trị ngẫu nhiên từ **cửa sổ cạnh tranh** hiện tại và bắt đầu bộ định thời bằng giá trị này. Bộ định thời giảm sau mỗi khe. Nếu nút khác bắt đầu trong lúc này thì bộ định thời sẽ bị treo và được lấy lại sau khi kết thúc khung kế tiếp và quá trình cạnh tranh lại tiếp tục. Nếu bộ định thời giảm đến không (zero) thì nút sẽ truyền khung của nó. Khi xảy ra lỗi truyền dẫn (ví dụ như thiếu khung phản hồi) thì kích thước của cửa sổ cạnh tranh tăng theo thủ tục backoff với số mũ nhị phân thay đổi. Mặc dù các giao thức CSMA dễ bị ảnh hưởng bởi các xung đột nhưng nó lại có hệ số hiệu suất thông lượng cao hơn các giao thức ALOHA do các gói tin đang truyền sẽ không bị phá hủy khi các nút đối thủ nghe được chúng trên môi trường truyền dẫn.

Như đã nói ở trên, các giao thức cảm nhận sóng mang dễ bị ảnh hưởng bởi vấn đề đầu cuối ẩn do nhiễu ở bộ thu không thể phát hiện bởi bộ phát. Vấn đề này có thể gây ra xung đột các gói tin. Năng lượng sử dụng trong các gói tin bị xung đột là lãng phí và chúng phải được truyền lại. Đã có một số giải pháp được đưa ra nhằm giải quyết hoặc ít nhất cũng làm giảm vấn đề đầu cuối ẩn đó là giải pháp âm báo bận và bắt tay RTS/CTS.

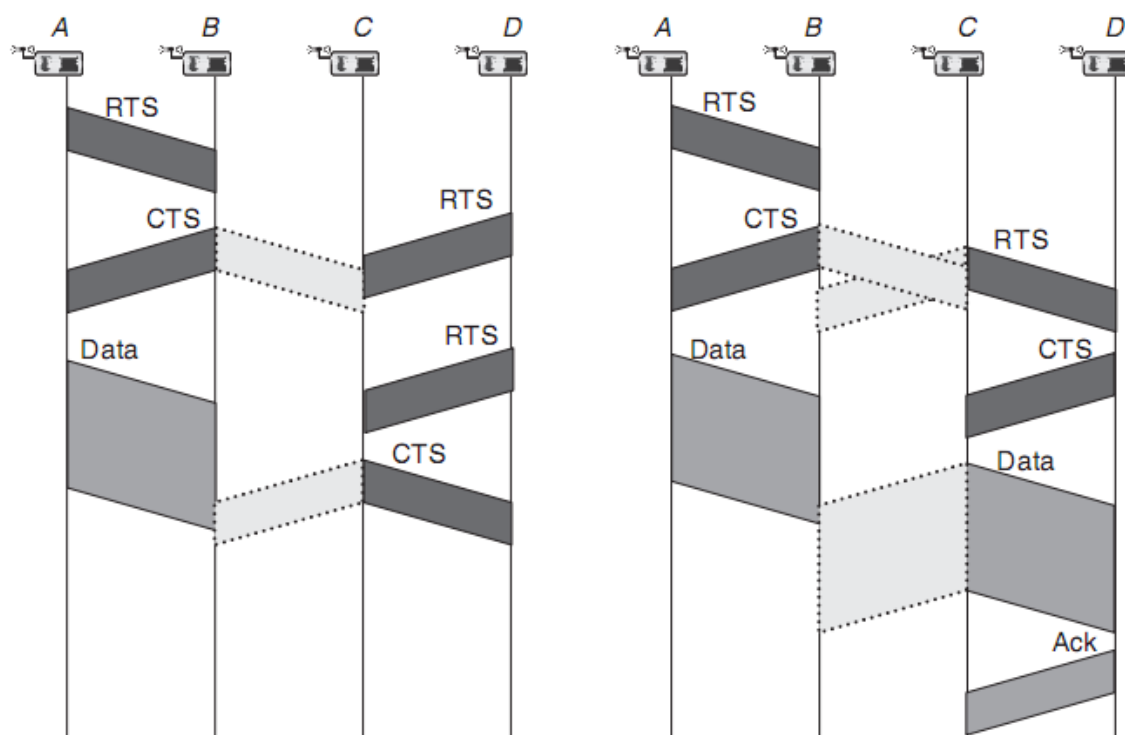
Trong giải pháp **âm báo bận**, hai kênh tần số khác nhau được sử dụng, một cho các gói dữ liệu và một cho kênh điều khiển. Ngay khi nút bắt đầu nhận gói tin được dự định truyền cho nó, nó phát ra một sóng không được điều chế lên kênh điều khiển và kết thúc điều này khi việc nhận gói tin được hoàn tất. Một nút muốn truyền tin thì trước hết nó phải cảm nhận kênh điều khiển xem có âm báo bận không. Nếu nghe được điều gì đó thì nút sẽ chờ để truyền theo một số thuật toán, ví dụ như tương tự với CSMA không liên tục. Nếu không nghe thấy gì thì nút bắt đầu truyền trên kênh dữ liệu. Giao thức này giải quyết cả vấn đề đầu cuối ẩn và đầu cuối hiện bằng cách dùng tín hiệu âm báo bận. Nếu âm báo bận quá yếu, nút trong dải vô tuyến của bộ thu có thể bắt đầu truyền dữ liệu và làm hỏng tín hiệu của bộ thu. Nếu âm báo bận quá mạnh, sẽ có nhiều nút hơn cần thiết bị cấm truyền. Kênh điều khiển không cần nhiều băng thông nhưng kênh băng thông hẹp yêu cầu phải đồng bộ tần số tốt. Giải pháp với hai âm báo bận, một sẽ được gửi bởi bộ thu và một được gửi bởi nút phát. Phần 5.3.2 giới thiệu các phương pháp âm báo bận khác nhau được sử dụng bởi PAMAS.

Bắt tay RTS/CTS được sử dụng trong IEEE802.11 dựa trên giao thức MACAW và được mô tả như trong hình 5.2. Nó chỉ sử dụng một kênh và hai gói điều khiển đặc biệt. Giả sử nút B muốn truyền dữ liệu đến nút C . Sau khi B truy nhập được kênh truyền (ví dụ như sau khi cảm nhận được kênh đang rỗi), nó sẽ gửi gói **yêu cầu để gửi** (RTS - Request To Send) đến C gồm khoảng thời gian còn lại của toàn bộ giao dịch (có nghĩa là cho đến khi B nhận phản hồi về gói dữ liệu của nó). Nếu C nhận đúng gói RTS, nó sẽ gửi lại gói **chắc chắn để gửi** (CTS - Clear To Send). Khi B nhận gói CTS, nó bắt đầu truyền dữ liệu và cuối cùng, C trả lời bằng gói phản hồi. Phản hồi được sử dụng để nói với B về sự thành công của quá trình truyền dẫn. Thiếu phản hồi thì sẽ được hiểu là có xung đột. Nút A và D nghe được gói RTS hoặc CTS hoặc gói dữ liệu hoặc gói phản hồi sẽ thiết lập một bộ định thời nội bộ, còn được gọi là **vector định vị mạng** (NAV - Network Allocation Vector) để khoảng thời

gian còn lại được biểu thị trong khung tương ứng và tránh việc gửi tin cho đến khi bộ định thời này chưa kết hết hiệu lực. Đặc biệt, nút A và D không gửi gói trả lời CTS ngay cả khi nó nhận được đúng gói RTS. Do đó, quá trình truyền dẫn đang thực hiện không bị sai.



Hình 5.2 Bắt tay RTS/CTS trong IEEE 802.11



Hình 5.3 Hai vấn đề trong bắt tay RTS/CTS

Cách này có loại trừ hoàn toàn được xung đột không? Câu trả lời là không, vẫn xảy ra một số xung đột. Trước hết, như mô tả ở trên, nút *A* và *C* có thể đồng thời phát ra gói RTS đến *B*. Tuy nhiên, trong trường hợp này, các gói RTS bị mất và không có khung dữ liệu dài nào được truyền. Hai vấn đề khác được mô tả trong hình 5.3: trong phần bên trái của hình, các nút *A* và *B* chạy chuỗi RTS-CTS-dữ liệu-phản hồi và gói CTS của *B* cũng đến được *C*. Tuy nhiên, ở cùng thời điểm đó, nút *D* cũng gửi gói RTS đến *C* và xảy ra xung đột giữa nó với gói CTS ở *B* (tại nút *C*). Do đó, *C* không có cơ hội để giải mã trường khoảng thời gian của gói CTS và đặt biến NAV. Sau khi gói RTS bị lỗi, *D* gửi lại gói RTS đến *C* và *C* trả lời bằng gói CTS. *C* làm vậy là do nó không biết gì về quá trình truyền dẫn đang diễn ra ở *A* và không có sự nhập vào NAV một cách đúng đắn. Gói CTS của *C* và gói dữ liệu của *A* xung đột ở *B*. Trong phần bên phải của hình vẽ, vấn đề nảy sinh khi *C* bắt đầu truyền gói RTS của nó đến *D* ngay trước khi nó cảm nhận được gói CTS của *B* và do đó, *C* không thể giải mã một cách đúng đắn được. Một giải pháp được đưa ra, đó là đảm bảo rằng các gói CTS dài hơn các gói RTS. Giải thích điều này bằng cách quan sát phần bên phải hình 5.3. Ở đây thậm chí nếu gói CTS của *B* đến *C* ngay sau khi *C* bắt đầu RTS của nó thì nó cũng đủ dài để *C* có cơ hội điều chỉnh bộ thu phát của nó về chế độ thu và để cảm nhận tín hiệu của *B*. Một luật giao thức nữa là nút *C* hoãn bất cứ quá trình truyền dẫn tiếp theo nào trong một thời gian đủ dài để tạo một gói dữ liệu có độ dài cực đại. Do đó, gói dữ liệu giữa *A* và *B* có thể được truyền mà không bị méo. Không khó để nhận ra rằng vấn đề trong phần bên trái của hình 5.3 cũng đã được loại bỏ.

5.1.3 Các giao thức MAC cho WSN

Phần này sẽ đưa ra các yêu cầu đặc biệt và những mối quan tâm khi thiết kế các giao thức MAC cho WSN.

Cân bằng các yêu cầu

Đối với WSN, cân bằng các yêu cầu khó thực hiện hơn các mạng (không dây) truyền thống. Yêu cầu đầu tiên và quan trọng nhất là bảo tồn năng lượng. Sự quan trọng của hiệu quả năng lượng trong thiết kế các giao thức MAC là vấn đề tương đối mới và một số các giao thức thông thường như ALOHA và CSMA chưa đề cập đến. Các đặc tính thông thường khác như tính công bằng, công suất hoặc trễ đóng vai trò thứ yếu trong các mạng cảm biến. Tính công bằng không quan trọng vì các nút trong WSN không có sự cạnh tranh về băng thông nhưng chúng kết hợp với nhau để có được mục đích chung này. Đặc tính trễ truy cập/ truyền dẫn và công suất thường tỉ lệ nghịch với việc tiêu hao năng lượng.

Yêu cầu quan trọng khác đối với các giao thức MAC khả năng mở rộng và tính bền vững để chống lại những thay đổi về cấu hình mạng. Yêu cầu về khả năng mở rộng là hiển nhiên khi xem xét những mạng cảm biến có mật độ rất cao với hàng chục hoặc hàng trăm nút có quan hệ với nhau.

Vấn đề năng lượng trong lớp MAC

Như đã thảo luận trong chương 2 và 4, bộ thu phát của các nút tiêu thụ một lượng năng lượng đáng kể. Nó có thể có bốn trạng thái chính là truyền tin, nhận tin, nghỉ và ngủ. Phần 2.1.4 đã thảo luận các vấn đề về tiêu thụ năng lượng của một số bộ thu phát trong các trạng thái hoạt động khác nhau. Tóm lại là việc truyền tin tốn nhiều năng lượng, nhận tin cũng tốn

năng lượng như truyền tin, trạng thái nghỉ có thể mất ít năng lượng hơn nhưng cũng tương đương như việc nhận tin, trạng thái ngủ không mất năng lượng nhưng tạo ra nút “điếc”. Áp dụng phần này vào hoạt động của giao thức MAC ta có các **vấn đề năng lượng** và mục đích thiết kế như sau:

Xung đột: Xung đột là không cần thiết cho chi phí nhận tin tại nút đích, không cần thiết cho chi phí truyền tin tại nút nguồn và phải cung cấp thêm năng lượng cho việc truyền tin lại. Do đó, phải tránh xung đột hoặc bằng thiết kế (các giao thức phân chia cố định/ TDMA hoặc các giao thức phân chia theo yêu cầu) hoặc bằng cách tránh xung đột thích hợp/ đầu cuối ẩn như trong các giao thức CSMA. Tuy nhiên, nếu chắc chắn rằng ứng dụng mạng cảm biến luôn có tải đủ nhỏ thì xung đột không là vấn đề được quan tâm.

Nghe lén: Các khung truyền thông đơn hướng có một nguồn và một nút đích. Tuy nhiên, môi trường vô tuyến là môi trường quảng bá nên tất cả các nguồn có trạng thái nhận ở cạnh nhau đều có thể nghe tin dù không muốn. Những nút này được gọi là nghe lén tin. Các nghiên cứu chỉ ra càng tránh được việc nghe lén càng nhiều thì càng tiết kiệm được nhiều năng lượng. Mặt khác, nghe lén đôi khi lại là mong muốn, ví dụ như khi lựa chọn thông tin lân cận nhau hoặc đánh giá lưu lượng tải hiện tại cho mục đích quản lý.

Tầng trên giao thức: Tầng trên giao thức tạo bởi các khung điều khiển liên quan đến MAC là các gói tin RTS và CTS hoặc các gói yêu cầu trong các giao thức phân chia theo yêu cầu.

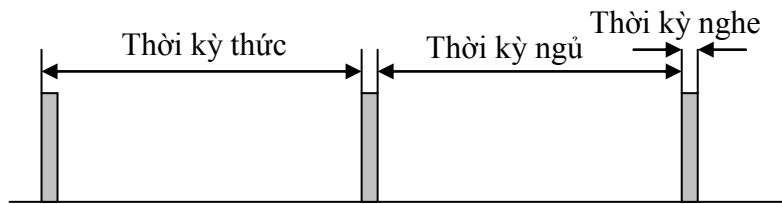
Nghe rồi: Nút ở trạng thái rồi sẵn sàng nhận tin nhưng nó vẫn chưa nhận được gì. Sự sẵn sàng này gây tốn kém và không cần thiết trong trường hợp mạng có tải thấp. Đối với nhiều modem vô tuyến, trạng thái rồi vẫn tiêu tốn năng lượng đáng kể. Tất bộ thu phát là một giải pháp, tuy nhiên, do thay đổi chế độ hoạt động cũng tiêu tốn nhiều năng lượng nên tần suất phải giữ ở mức hợp lý. Các giao thức dựa trên TDMA đưa ra giải pháp ẩn cho vấn đề này do nút được ẩn định vào khe thời gian và nó chỉ thay đổi trạng thái hoạt động (phát/ thu) nếu các khe thời gian này tắt bộ thu phát một cách an toàn trong tất cả các khe thời gian khác.

Hầu hết các giao thức MAC sử dụng cho WSN đều sử dụng ít nhất một trong các vấn đề này để giảm năng lượng tiêu thụ.

Một thiết kế quan tâm đến mức tiêu thụ năng lượng thường yêu cầu **hoạt động ít phức tạp** (low complexity operation). Các nút cảm biến thường đơn giản, rẻ tiền, không có khả năng lớn về xử lý, bộ nhớ và năng lượng. Vì vậy, các hoạt động tính toán phức tạp thường tránh sử dụng. Mong muốn ở đây là dùng phần cứng cho nút có giá rẻ, bao gồm các thiết bị như bộ tạo dao động và clock. Một chú ý khi thiết kế các giao thức MAC đó là yêu cầu chặt chẽ về đồng bộ theo thời gian (cũng như yêu cầu đối với TDMA là các khe thời gian nhỏ) sẽ kéo theo yêu cầu phải thường xuyên đồng bộ lại các nút lân cận, mà điều này có thể tiêu thụ năng lượng đáng kể.

5.2 Các giao thức hệ số lấp đầy thấp và khái niệm thức dậy (wake-up)

Các **giao thức hệ số lấp đầy thấp** (Low duty cycle protocol) (là tỉ lệ thời gian hoạt động trên thời gian một chu kỳ) tránh việc sử dụng quá nhiều thời gian ở trạng thái nghỉ (rời) và giảm các hoạt động giao tiếp của nút cảm biến đến mức tối thiểu. Trong trường hợp lý tưởng, chỉ thoát khỏi trạng thái ngủ khi nút truyền và nhận tin. Tuy nhiên, không thể xây dựng được các hệ thống như vậy và người ta phải tìm các phương pháp khác.



Hình 5.4 Sơ đồ thức dậy định kỳ

Trong một số giao thức, cơ chế **thức dậy định kỳ** (periodic wakeup) được sử dụng. Cơ chế này tồn tại dưới một số cách khác nhau. Thứ nhất là phương pháp **bộ thu chu kỳ** (cycled receiver). Trong phương pháp này, các nút sử dụng phần lớn thời gian ở trạng thái ngủ và được thức dậy có tính chu kỳ để nhận tin từ các nút khác. Đặc biệt, nút *A* nghe trên kênh truyền chỉ trong **thời kỳ nghe** (listen period) của nó và trở lại trạng thái ngủ khi không còn nút nào gửi tin đến *A*. Bộ phát *B* yêu cầu phải biết về thời kỳ nghe của *A* để gửi tin tại đúng thời điểm đó, công việc này tương đương như một cuộc hẹn. Tuy nhiên, trong trường hợp khác, nút *A* chỉ nhận tin trong thời kỳ nghe của nó. Nếu bản thân nút *A* muốn truyền tin thì nó phải có được thời kỳ nghe của nơi đến. Một chu trình đầy đủ bao gồm thời kỳ ngủ và thời kỳ nghe được gọi là chu trình hoạt động của nút. Nó có một số đặc điểm quan trọng sau:

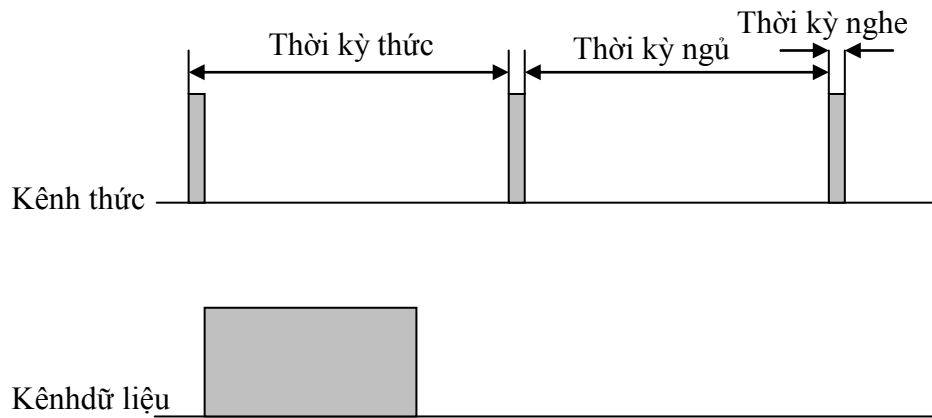
- Bằng cách chọn hệ số lấp đầy thấp, bộ thu phát hầu như ở trạng thái ngủ nên nó sẽ tránh được trạng thái nghe rồi và tiết kiệm năng lượng tiêu thụ.
- Bằng cách chọn hệ số lấp đầy thấp, thông tin trực tiếp từ các nút lân cận đến nút đã cho tập trung trong một khoảng thời gian nhỏ (thời kỳ nghe) nên có thể gây ra hiện tượng quá tải.
- Bằng cách chọn thời kỳ ngủ dài nên nó tạo ra **thời gian chờ trên bước nhảy** (per-hop latency) rất hiệu quả do nút truyền sắp tới thường phải chờ trung bình là nửa thời kỳ ngủ trước khi bộ thu nhận tin. Trong trường hợp đa bước nhảy, các khoảng chờ trên bước nhảy tăng và tạo ra các khoảng chờ từ điểm đầu đến điểm cuối rất hiệu quả.
- Thời gian ngủ không được quá ngắn vì quá trình khởi động mất rất nhiều năng lượng.

Trong các giao thức khác, ví dụ như S-MAC, cũng có cơ chế thức dậy định kỳ nhưng nút có thể vừa phát vừa thu trong thời gian thức của nó. Khi các nút thức tại cùng một thời điểm, nút không phải thoát khỏi trạng thái này để hẹn với bộ thu khi muốn truyền tin.

5.2.1 Cấu hình mở và vấn đề quản lý năng lượng

Giao thức cấu hình mở và vấn đề quản lý năng lượng (STEM - The Sparse Topology and Energy Management) không bao hàm tất cả các khía cạnh của giao thức MAC nhưng nó đưa ra giải pháp cho vấn đề nghe rồi. STEM tạo ra các mạng được sử dụng để đợi và ghi nhận hành vi của một hiện tượng nào đó, ví dụ như nghiên cứu đường đi của các con voi trong môi trường sống. Đối với cảm biến đơn lẻ, hầu hết thời gian là không có voi nên cảm biến không ghi nhận được gì. Tuy nhiên, một khi voi xuất hiện, cảm biến ghi lại và đọc nó theo định kỳ. Trừu tượng hơn là mạng có trạng thái giám sát, ở đó các nút rồi không làm gì kể cả ở trạng thái truyền dẫn, khi mà các nút có cảm nhận về môi trường và có hoạt động liên lạc. STEM cố gắng loại bỏ trạng thái nghe rồi trong trạng thái giám sát và nhanh chóng chuyển sang trạng thái truyền dẫn khi có yêu cầu. Ở trạng thái truyền dẫn, các giao thức MAC

khác có thể được sử dụng. Thuật ngữ “cấu hình mạng” trong STEM xuất phát từ quan sát nhận thấy rằng các nút vào và thoát khỏi trạng thái ngủ sẽ thay đổi cấu hình mạng. Một yêu cầu quan trọng cho cơ chế quản lý cấu hình mạng là mạng phải ở trạng thái kết nối, có thể có một số nút ở trạng thái ngủ.



Hình 5.5 Chu trình làm việc STEM của một nút đơn

Hình 5.5 giải thích hoạt động của STEM. Hai kênh khác nhau được sử dụng và yêu cầu hai bộ thu phát cho mỗi nút: kênh thức và kênh dữ liệu. Ngoài trạng thái truyền và nhận tin thì kênh dữ liệu luôn ở trạng thái ngủ. Ở kênh thức, thời gian được chia thành các thời kỳ thức có độ dài cố định T . Thời kỳ thức lại được chia thành các thời kỳ nghe có độ dài $T_{Rx} \ll T$ và thời kỳ ngủ, khi mà các bộ thu phát của kênh thức cũng ở trạng thái ngủ. Nếu nút ở thời kỳ nghe, rất đơn giản để bật bộ thu của kênh thức để đợi tín hiệu đến. Nếu không nhận được gì trong khoảng thời gian T_{Rx} , nút sẽ quay trở lại trạng thái ngủ. Nếu có, bộ phát và bộ thu sẽ bắt đầu truyền và nhận tin trên kênh dữ liệu. Có hai điểm khác nhau đối với bộ phát để yêu cầu sự chú ý của bộ thu:

- Trong STEM-B (Busy), bộ phát được gọi là **các báo hiệu** trên kênh thức một cách định kỳ mà không có cảm nhận sóng mang. Mỗi báo hiệu như vậy biểu thị các địa chỉ MAC của bộ phát và bộ thu. Ngay khi bộ thu có được báo hiệu, nó sẽ gửi tin báo đã nhận được phản hồi lên kênh thức (để kênh phát dừng việc truyền báo hiệu), bật bộ thu phát cho kênh dữ liệu và cả hai nút có thể bắt đầu thực hiện giao thức MAC trên kênh dữ liệu. Một nút nào đó có thể nhận báo hiệu trên kênh thức và nhận ra rằng gói tin không phải dành cho nó thì nó sẽ quay trở lại trạng thái ngủ. Bộ phát gửi các báo hiệu này ít nhất là trong thời gian thức để đảm bảo gặp được thời kỳ nghe của các bộ thu.

- Trong STEM-T, bộ phát gửi một âm báo bận đơn giản lên kênh giám sát (T trong Stem-T có nghĩa là tone- âm) trong một thời gian đủ dài để gặp được thời kỳ nghe của các bộ thu. Do âm báo bận không mang thông tin về địa chỉ nên tất cả các nút lân cận của bộ thu (gồm bộ thu và các nút khác) sẽ cảm nhận âm báo bận và bật kênh dữ liệu của nó nhưng không gửi tin phản hồi về hoạt động này. Các nút khác quay trở lại trạng thái ngủ khi chúng nhận ra rằng gói tin trên kênh dữ liệu không liên quan đến việc truyền dữ liệu. Bộ thu phát có thể tạo và cảm nhận các âm báo bận thường rẻ tiền và tiêu thụ ít năng lượng hơn bộ thu phát được sử dụng để truyền dẫn dữ liệu nhưng yêu cầu đồng bộ về mặt tần số.

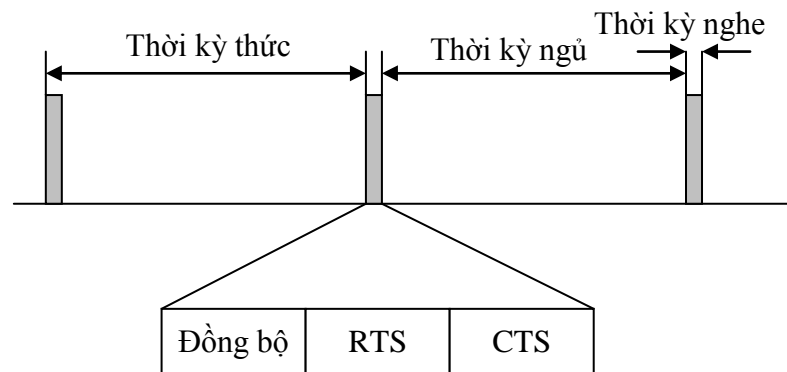
Trong STEM-B, một số bộ phát có thể truyền tức thời các báo hiệu của nó, điều này

dẫn đến xung đột giữa các báo hiệu. Một nút ở trạng thái thức và chỉ nhận một phần thông tin trên kênh thức nên không thể giải mã nó một cách chính xác như trong STEM-T: Nó không gửi thông tin phản hồi, bật kênh dữ liệu và đợi xem chuyện gì xảy ra. Trong trường hợp này, bộ phát sẽ gửi báo hiệu trong thời gian tối đa (từ khi nó biết không có thông tin phản hồi), sau đó nó chuyển sang kênh dữ liệu và cố gắng liên lạc với nút của bộ thu.

Cần lưu ý rằng trong STEM, nút ở thời kỳ nghe vẫn yên lặng, khi đó không có sự truyền tin. Có thể lựa chọn cách tiếp cận ngược lại, ví dụ như trong giao thức thiết bị hoà giải đưa ra trong phần 5.2.3 ở hệ thống Piconet, các nút chỉ thức để thông báo sẵn sàng nhận tin bằng cách truyền một gói tin báo hiệu. Trong phương pháp đưa ra bởi STEM, bộ phát phải gửi các báo hiệu hoặc âm báo bận trong khoảng thời gian trung bình là $\approx T/2$ và trong trường hợp xấu nhất là trong khoảng thời gian tối đa $\approx T$. Nếu gói tin truyền đi là một sự kiện hiếm gặp thì không nên (và không cần thiết) thường xuyên hỏi các báo hiệu, như thế sẽ làm tăng thêm gánh nặng trên đường truyền dẫn. Vì vậy, trong trường hợp tải thấp thường hay sử dụng STEM-T hơn STEM-B.

5.2.2 S-MAC

Giao thức S-MAC (Sensor-MAC) tạo ra các cơ chế để loại bỏ nghe rồi, các xung đột và nghe lén. Ngược với STEM, nó không yêu cầu phải có hai kênh truyền khác nhau.



Hình 5.6 Nguyên lý S-MAC

S-MAC chấp nhận cơ chế thức định kỳ, đó là nút sẽ xen kẽ thời kỳ nghe và thời kỳ ngủ có độ dài cố định theo lịch trình của nó, như hình 5.6. Tuy nhiên, ngược với STEM, thời kỳ nghe của S-MAC có thể được dùng để truyền và nhận tin. S-MAC cố gắng kết hợp lịch trình của các nút lân cận để các thời kỳ nghe của chúng bắt đầu cùng một lúc. Thời kỳ nghe của nút x được chia thành ba giai đoạn khác nhau:

- **Giai đoạn đầu tiên (giai đoạn đồng bộ)**, nút x nhận các gói tin đồng bộ từ các nút lân cận của nó. Trong các gói tin đó, các nút lân cận mô tả lịch trình của nó và nút x chứa lịch trình của các nút lân cận trong bảng lịch trình (schedule table). Giai đoạn đồng bộ của nút x được chia thành các khe thời gian và các nút lân cận theo cơ chế CSMA, đó là mỗi nút lân cận y mong muốn truyền gói đồng bộ trong một khe thời gian ngẫu nhiên nào đó và bắt đầu truyền nếu không nhận được tín hiệu trong bất kỳ khe thời gian nào trước đó. Trong trường hợp khác, y quay trở lại trạng thái ngủ và đợi chu kỳ thức tiếp theo của x .

Theo một phương diện khác, do x biết lộ trình của nút lân cận y nên x có thể thức tại đúng thời điểm phù hợp và gửi gói tin đồng bộ đến y (ở chế độ quảng bá). Điều này không

yêu cầu x phải thông báo lộ trình của nó đến tất cả các thời kỳ thức của y . Tuy nhiên, vì các lý do đồng bộ về mặt thời gian và để cho các nút mới biết cấu hình mạng cục bộ của chúng mà x phải gửi các gói đồng bộ một cách định kỳ. Do đó, thời kỳ này được gọi là **thời kỳ đồng bộ hoá**.

- Trong giai đoạn thứ hai (**giai đoạn RTS**), x lắng nghe các gói RTS từ các nút lân cận. Trong S-MAC, chế độ bắt tay RTS/CTS mô tả trong phần 5.1.2 được sử dụng để giảm xung đột giữa các gói dữ liệu do các vị trí đầu cuối ẩn.

- Trong giai đoạn thứ ba (**giai đoạn CTS**), nút x truyền gói CTS nếu nhận được gói RTS ở giai đoạn thứ hai. Sau đó, việc trao đổi gói tin tiếp tục, mở rộng thành thời gian ngủ danh định của nút x .

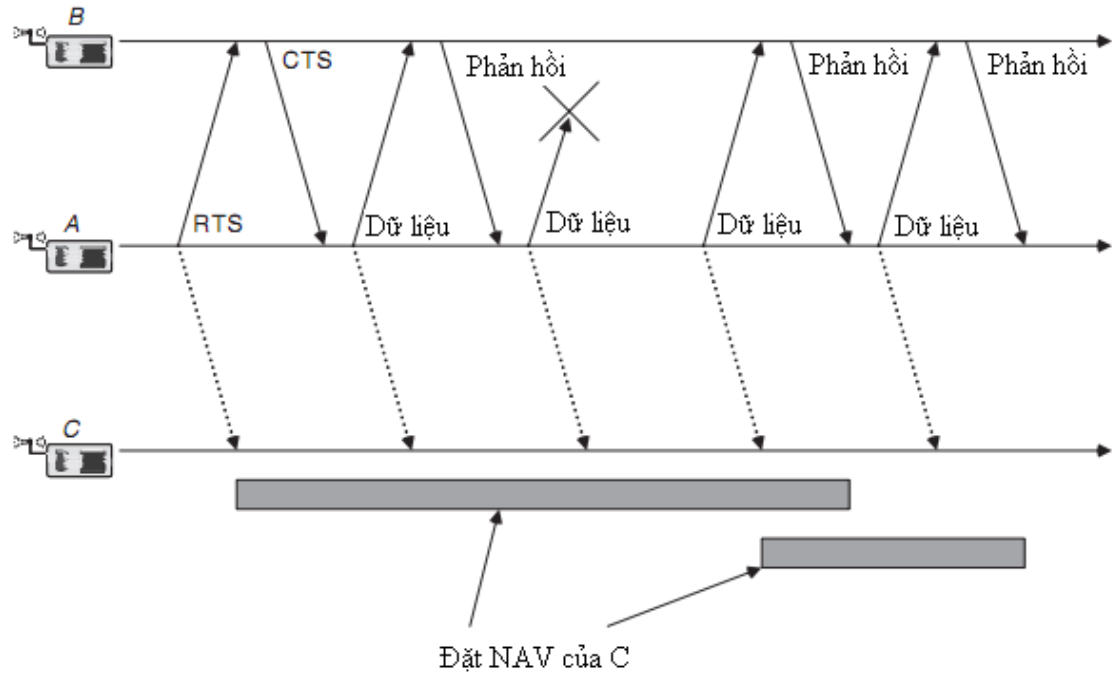
Thông thường, khi cạnh tranh trong môi trường, các nút sử dụng chế độ bắt tay RTS/CTS, bao gồm cơ chế cảm nhận sóng mang ảo, do đó nút duy trì biến NAV. Cơ chế NAV được dùng để tắt nút khi nó đang trong giai đoạn truyền dẫn để tránh bị nghe lén. Khi truyền trong chế độ quảng bá (ví dụ như các gói tin đồng bộ), các gói tin RTS và CTS bị bỏ đi và các nút sử dụng CSMA với backoff.

Nếu có thể đồng bộ được lịch trình của x và các nút lân cận của nó thì chúng sẽ thức tại cùng một thời điểm và x có thể liên lạc được với các nút lân cận đó chỉ bằng một gói tin đồng bộ. Giao thức S-MAC cho phép các nút lân cận sử dụng một lịch trình để tạo thành các **nhóm ảo**. Cấu trúc nhóm chỉ được dùng để trao đổi lịch trình, việc truyền dữ liệu không bị ảnh hưởng bởi nhóm ảo.

Giao thức S-MAC được mô tả như sau theo các nhóm ảo: Nút x mới được bật đang trong giai đoạn lắng nghe, ít nhất là ở thời kỳ đồng bộ. Nếu x nhận bất kỳ gói đồng bộ nào từ nút lân cận, nó sẽ chấp nhận lịch trình đã công bố và quảng bá ở một trong các thời kỳ nghe tiếp theo của các nút lân cận. Trong trường hợp khác, nút x lựa chọn lịch trình và quảng bá nó. Nếu nút x nhận lịch trình của nút khác trong giai đoạn cạnh tranh gói tin quảng bá, nó sẽ bỏ lịch trình của mình và thực hiện theo lịch trình đã nhận được. Cũng có thể xảy ra trường hợp nút x nhận được lịch trình khác khi nó đã chọn sẵn một lịch trình khác, ví dụ như các lỗi bit làm hỏng các gói tin đồng bộ trước đó. Nếu nút x đã biết về sự tồn tại của các nút lân cận và chấp nhận lịch trình với chúng, nó sẽ giữ lịch trình này và sau đó phải truyền gói dữ liệu và đồng bộ theo cả hai lịch trình. Mặt khác, nếu x không có nút lân cận để chia sẻ lịch trình thì nó sẽ loại bỏ lịch trình của nó và sử dụng một lịch trình khác. Do gói đồng bộ nhận được luôn có lỗi nên nút x phải lắng nghe một cách định kỳ trong cả thời kỳ đồng bộ để cập nhật các nút lân cận của nó, điều này tạo nên một nhóm ảo.

Sơ đồ thức định kỳ được sử dụng bởi S-MAC cho phép các nút dùng rất nhiều thời gian cho chế độ ngủ, nhưng điều này cũng trả giá bằng trễ (latency). Không có nhiều thay đổi, thời gian đợi trên một bước nhảy của S-MAC tương đương với thời kỳ ngủ trung bình khi tất cả các nút cùng theo một lịch trình. YE et al. đã mô tả một cơ chế **nghe thích nghi**. Xem xét tình huống sau: trong thời kỳ nghe, nút x nhận một gói tin RTS hoặc CTS phụ thuộc vào sự thay đổi gói tin từ nút lân cận y đến z . Trong thời gian tồn tại của các gói tin này, x có thể tạo ra thời gian t_0 khi kết thúc sự thay đổi gói tin. Do có thể xảy ra trường hợp x là bước nhảy tiếp theo của gói tin từ z nên nút x có thời gian nghe thêm xung quanh giá trị t_0 và z có gửi thêm gói RTS tại thời điểm t_0 mà không cần để ý đến thời kỳ thức mặc định của x . Trong trường hợp lý tưởng, x thức khi z gửi RTS và gói tin có thể chuyển đến bước nhảy tiếp theo

rất nhanh.



Hình 5.7 Khung S-MAC và đặt NAV

S-MAC cũng có thể sử dụng phương pháp chuyển thông báo (như hình 5.7), ở đây thông báo là dữ liệu có ý nghĩa lớn hơn đối với ứng dụng. Xử lý trong mạng thường yêu cầu tập hợp nút để nhận thông báo một cách đầy đủ. Mặt khác, ở thiết bị không dây, nó được sử dụng để ngắt một gói tin dài thành các gói tin ngắn hơn. Cơ chế phân đoạn của S-MAC được thực hiện như sau: các đoạn chỉ được truyền theo một RTS/CTS giữa nút phát A và nút nhận B. Sau mỗi đoạn, B trả lời về gói tin đã nhận. Tất cả các gói tin (dữ liệu, tin tức, RTS, CTS) đều có khoảng thời gian và nút lân cận C được yêu cầu để đặt NAV một cách phù hợp. Trong S-MAC, khoảng thời gian của tất cả các gói tin vẫn có độ dài là một bước chuyển, bao gồm tất cả các đoạn và thông tin phản hồi của chúng. Vì vậy, tất cả thông báo phải được chuyển đi một lần. Nếu một đoạn nào đó cần được truyền lại thì khoảng thời gian trên sẽ tăng lên một lượng bằng độ dài của gói dữ liệu cộng với tin phản hồi của nó, và môi trường cũng phải dành thêm khoảng thời gian phát sinh này.

Cơ chế này có một số điểm tương đồng so với cơ chế phân đoạn được sử dụng trong IEEE 802.11 nhưng có một số khác biệt quan trọng. Trong IEEE 802.11, khung RTS và CTS chỉ đặt trước môi trường trong thời gian của đoạn đầu tiên, đoạn bất kỳ chỉ có thể đặt trước cho đoạn tiếp theo. Nếu một gói tin cần truyền lại, nút đầu tiên sẽ phải từ bỏ kênh truyền, coi nó như một gói tin mới và lặp lại quá trình trên. Phương pháp đưa ra bởi S-MAC giảm trễ của các gói tin bằng cách bỏ các truyền dẫn của các gói tin khác nhau bị quấn vào nhau. Vì vậy, xét về một mặt nào đó, giao thức này là không đúng vì các nút đơn có thể ngăn cản môi trường trong một thời gian dài. Tuy nhiên, như giải thích trong phần 5.1.3, yêu cầu về tính công bằng trong WSN khác với mạng dữ liệu, nơi mà người sử dụng mong muốn có một môi trường truy cập công bằng.

Hạn chế chính của S-MAC là rất khó thích nghi độ dài của thời kỳ thức với các trạng thái khác nhau của tải do độ dài này và độ dài thời kỳ nghe thường cố định.

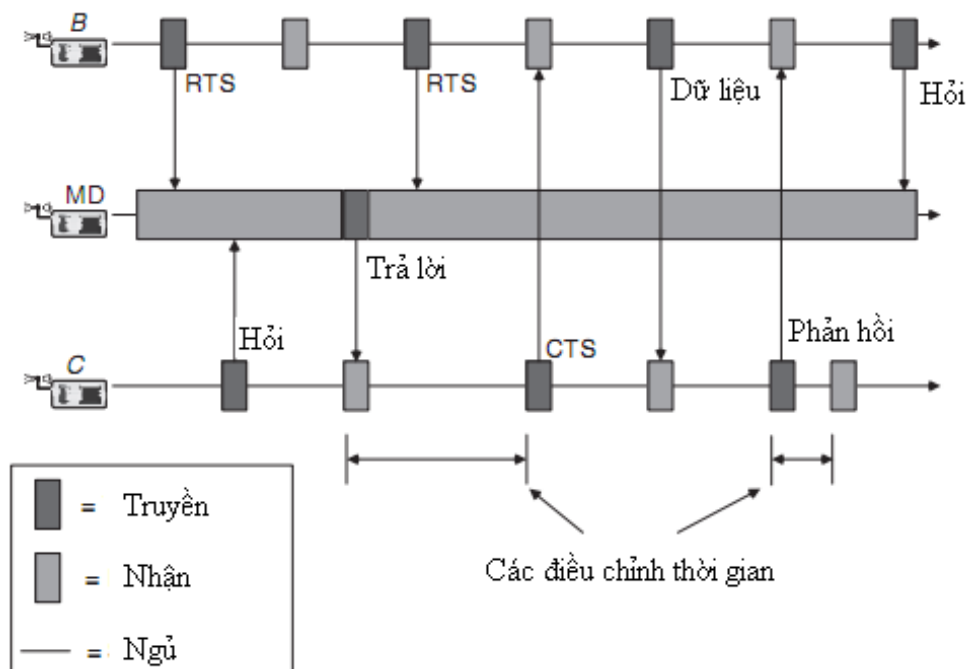
Giao thức T-MAC được đưa ra bởi van Dam và Langendoen, nó tương tự như S-MAC nhưng đã rút ngắn thời kỳ nghe một cách phù hợp. Nếu nút x cảm thấy không có hoạt động nào ở môi trường trong một khoảng thời gian nào đó, nó sẽ chuyển ngay sang trạng thái ngủ. Vì vậy, nếu không có nút nào muốn truyền đến x , thời kỳ nghe có thể kết thúc một cách nhanh chóng trong khi ở S-MAC, độ dài thời kỳ nghe là cố định.

5.2.3 Giao thức thiết bị trung gian

Giao thức thiết bị trung gian (mediation device protocol) phù hợp với chế độ giao tiếp ngang hàng (peer-to-peer) chuẩn tốc độ thấp WPAN của IEEE 802.15.4. Nó cho phép mỗi nút trong WSN có trạng thái ngủ một cách định kỳ và chỉ thức dậy trong một thời gian ngắn để nhận tin từ các nút lân cận. Không có chung mối quan hệ thời gian nào, mỗi nút có lịch trình ngủ riêng và không quan tâm đến lịch trình ngủ của nút lân cận.

Mỗi lần thức dậy, nút sẽ truyền một **báo hiệu ngắn** mang thông tin địa chỉ của nút và cho biết nó đã sẵn sàng nhận tin từ các nút khác. Nút ở trạng thái thức trong một thời gian ngắn sau khi truyền báo hiệu, mở cửa sổ cho gói tin đến. Nếu không nhận được gói tin nào trong cửa sổ này, nút sẽ quay trở lại trạng thái ngủ.

Khi nút muốn truyền tin tới nút lân cận thì nó phải đồng bộ với nút đó. Điều đó có nghĩa là bên gửi sẽ phải đợi trong thời gian gửi báo hiệu và sẽ phí phạm năng lượng chỉ cho mục đích đồng bộ. Phương pháp **đồng bộ động** (dynamic synchronization) thực hiện đồng bộ này mà không yêu cầu bên phát phải thức cố định để tìm kiếm báo hiệu ở bên nhận. Muốn vậy, **thiết bị trung gian** (MD- Mediation Device) được sử dụng. Trường hợp MD không có ràng buộc về năng lượng và hoạt động ở mọi lúc được mô tả trên hình 5.8. Ở một chu kỳ hoạt động đầy đủ, MD có thể nhận báo hiệu từ tất cả các nút trong vùng lân cận của nó và biết được các thời kỳ thức của chúng.



Hình 5.8 Giao thức thiết bị trung gian với các bộ hoà giải không ràng buộc

Giả sử nút A muốn truyền tin đến nút B. Nút A thông báo điều này tới MD bằng cách gửi một cách định kỳ các gói tin **yêu cầu được gửi** (RTS - request to send). Nút A gửi các

gói RTS của nó thay vì các báo hiệu vào do đó, nó có cùng độ dài. Tồn tại một cửa sổ trả lời ngắn sau khi các gói RTS được gửi để A nghe câu trả lời. Sau khi MD nhận được gói RTS của A, nó đợi báo hiệu tiếp theo của B. MD trả lời điều này bằng gói tin trả lời mang thông tin địa chỉ của A và độ lệch thời gian, là thời gian để B biết khi gửi gói trả lời **chắc chắn để gửi** (CTS - clear to send) đến A, tức là khi gói CTS hiện trên cửa sổ trả lời sau khi có gói RTS tiếp theo của A. Do đó, B biết được các thời kỳ của A. Sau khi A nhận gói CTS, nó có thể gửi dữ liệu và đợi phản hồi từ B. Sau khi giao dịch kết thúc, A khôi phục chu kỳ thức định kỳ và phát lại các báo hiệu. Nút B cũng khôi phục chu kỳ định kỳ của nó và tách khỏi chu kỳ của A.

Giao thức này có một số ưu điểm. Trước hết là nó không yêu cầu đồng bộ về mặt thời gian giữa các nút, chỉ cần MD biết chu trình của các nút. Thứ hai, giao thức là không đối xứng theo nghĩa là hầu hết năng lượng sử dụng đã được chuyển vào MD. Các nút khác có thể hầu như ở trạng thái ngủ và chỉ sử dụng năng lượng cho việc báo hiệu định kỳ. Thậm chí khi bộ phát muốn đồng bộ với bộ thu nó cũng không phải đợi báo hiệu nhưng nó có thể quay trở lại trạng thái ngủ và đợi MD để đồng bộ mạng. Mạng hệ số lấp đầy thấp có thể hỗ trợ phương pháp này. Giao thức này có một số hạn chế sau: nút truyền báo hiệu mà không kiểm tra các quá trình truyền dẫn đang diễn ra và do đó, báo hiệu của các nút khác nhau có thể xung đột nhiều lần khi các nút có cùng chu kỳ và các giai đoạn thức của chúng bị chồng chéo nhau. Nếu các thời kỳ thức là ngẫu nhiên và mạng có mật độ đủ thấp thì xác suất xảy ra xung đột cũng thấp. Tuy nhiên, trong trường hợp mạng có mật độ cao hơn hoặc có sự đồng bộ không mong muốn giữa các nút thì xung đột có thể xảy ra nhiều. Giải pháp cho vấn đề này là: khi có xung đột giữa các thanh ghi MD, nó sẽ lịch trình lại khung điều khiển. Tất cả các nút xung đột có thể biết khung này và MD sẽ lặp lại nó một cách thường xuyên.

Nhược điểm chính của giao thức này là:

i) MD không hạn chế năng lượng nên nó không phù hợp với yêu cầu về năng lượng của WSN.

ii) Phải có đủ MD để quan sát được tất cả các nút. Giao thức thiết bị trung gian phân bố (distributed mediation device protocol) xử lý các vấn đề trên theo phương thức xác suất. Nó để các nút thức dậy một cách ngẫu nhiên và phục vụ nó như là MD trong một khoảng thời gian nào đó và sau đó đưa chúng quay trở lại trạng thái thức định kỳ thông thường. Thời gian phục vụ ít nhất cũng dài bằng chu kỳ cực đại của tất cả các nút cộng với độ dài báo hiệu. Tuy nhiên, theo các giả định này, ta không mong rằng MD tạm thời biết được chu kỳ và thời gian thức của tất cả các nút. Nếu ta giả thiết rằng tất cả các nút có cùng chu kỳ, thế cũng đủ để MD tạm thời nghe được RTS từ nút phát A và báo hiệu từ nút nhận B để tính độ lệch thời gian của chúng và chỉ dẫn nút B theo MD. Vấn đề ở đây là các nút A và B có hai hoặc nhiều hơn MD trong vùng lân cận của nó, điều này sẽ tạo ra xung đột ở một số báo hiệu. Bằng cách phân bố thời gian ngẫu nhiên một cách phù hợp khi các nút quyết định phục vụ MD, xác suất của xung đột có thể được giữ ở mức thấp.

5.3 Các giao thức theo kiểu cạnh tranh

Trong các giao thức theo kiểu cạnh tranh, cơ hội để truyền tới nút nhận đã biết có thể phụ thuộc vào các nút lân cận của nó. Nếu chỉ có một nút thì gói dữ liệu sẽ được truyền trên kênh truyền. Nếu số nút lớn hơn hoặc bằng hai thì sẽ có sự cạnh tranh với nhau và trong trường hợp không may, ví dụ như do các giải pháp đầu cuối ẩn, xung đột có thể xảy ra và

như vậy sẽ tổn năng lượng cho cả bộ phát và bộ thu. Phần 5.1.2 đã giới thiệu khái quát hai giao thức theo kiểu cạnh tranh quan trọng là (slotted) ALOHA và CSMA, cùng các cơ chế để giải quyết vấn đề đầu cuối ẩn. Phần này sẽ nêu sự khác nhau giữa các giao thức này với mục đích bảo tồn năng lượng.

Ngược với các giao thức theo kiểu cạnh tranh có cơ chế thức định kỳ (ví dụ như S-MAC trong phần 5.2.2), các giao thức mô tả trong phần này không tránh trạng thái nghe rồi và không có hạn chế khi nút nhận tin.

5.3.1 Các giao thức CSMA

Woo và Culler đã đưa ra một số giao thức CSMA khác nhau dựa trên giá thành năng lượng vốn có và tính công bằng của chúng mà không có quy định và các phép đo để tránh trạng thái nghe rồi hoặc tránh việc nghe lén. “Giá thành vốn có” chủ yếu là năng lượng sử dụng cho việc truyền và nhận tin.

Tác giả đã quan tâm tới mạng đa bước nhảy với một hoặc một số bộ chứa và có lưu lượng giống như STEM (trong phần 5.2.1): Mạng sẽ chạy rồi trong một thời gian dài và sẽ trở lại hoạt động khi được kích bởi một hiện tượng quan trọng từ bên ngoài. Dựa vào hiện tượng kích, tất cả các nút mong muốn sẽ được truyền ngay lập tức nên sẽ tạo ra rất nhiều xung đột. Trong trường hợp các nút muốn chuyển gói tin một cách định kỳ, nguy hiểm của các xung đột sẽ lặp lại nếu không thực hiện các phép đo đặc biệt.

Sau khi nút có được gói tin mới cho truyền dẫn từ các lớp cao hơn của nó, nó sẽ bắt đầu với một trễ ngẫu nhiên cho bộ đếm thử nghiệm `num_retries` giá trị khởi tạo bằng không (zero). Mục đích của trễ ngẫu nhiên là ngừng đồng bộ các nút đã được đồng bộ ban đầu bởi hiện tượng kích bên ngoài. Do trễ ngẫu nhiên này mà bộ thu phát của các nút có thể đưa vào trạng thái ngủ. Sau thời kỳ nghe, nút sẽ thực hiện cảm nhận sóng mang. Nếu môi trường bận hoặc số thử nghiệm nhỏ hơn rất nhiều giá trị cực đại thì nút sẽ chuyển đến trạng thái backoff. Ở trạng thái backoff, nút sẽ đợi trong thời gian ngẫu nhiên này, nó phụ thuộc vào số thử nghiệm và khoảng thời gian mà nút có thể ngủ (do đó giao thức này là sự biến đổi CSMA không liên tục). Chế độ backoff cũng có thể dùng cho lớp ứng dụng để khởi động “sự thay đổi pha”, tạo ra lưu lượng tuần hoàn. Sự thay đổi pha nhằm mục đích ngừng đồng bộ lưu lượng tuần hoàn hoặc tương quan của các nút khác nhau. Sau khi chế độ backoff kết thúc, nút sẽ lại lắng nghe. Nếu môi trường bận và nút đã quá mệt mỏi với số thử nghiệm tối đa, gói tin sẽ bị rút. Nếu môi trường rỗi, nút sẽ truyền một gói RTS và chuyển vào trạng thái đợi CTS (bước này có thể được bỏ qua nếu nút biết rằng mạng hiện trong tình trạng tải thấp). Trong trường hợp không có gói CTS nào đến hoặc nhận được gói CTS cho giao dịch khác thì nút hoặc ở trạng thái backoff hoặc bị rút gói tin, điều này phụ thuộc vào giá trị của `num_retries`. Nếu gói CTS đến, nút sẽ gửi gói dữ liệu của nó và đợi phản hồi. Phản hồi này là một gói riêng biệt hoặc nút gốc sẽ gắn phản hồi lên gói tin và truyền đi tiếp. Tuy nhiên, khi phản hồi nằm trên nút gốc, không dễ để xác định thời gian đợi phù hợp cho đến khi phản hồi đến được đích.

5.3.2 PAMAS

Giao thức **đa truy cập cảm nhận nguồn với báo hiệu PAMAS** (Power Aware Multiaccess with Signaling) được giới thiệu bởi Raghavendra và Singh là thiết kế đầu tiên của các mạng ad hoc. Nó đưa ra cơ chế tránh nghe lén một cách chi tiết trong khi nó không

quan tâm tới vấn đề nghe rồi. Giao thức kết hợp giải pháp âm báo bận và bắt tay RTS/CTS tương tự như giao thức MACA (MACA không có gói phản hồi cuối cùng). Đặc điểm phân biệt PAMAS là nó sử dụng hai kênh là **kênh dữ liệu** và **kênh điều khiển**. Tất cả các gói báo hiệu (RTS, CTS, âm báo bận) được truyền trên kênh điều khiển trong khi kênh dữ liệu được dành riêng cho các gói dữ liệu.

Xem xét một nút rồi x gửi một gói tin mới tới nút lân cận y . Trước hết x gửi gói RTS lên kênh điều khiển mà không cảm nhận sóng mang. Gói tin này mang địa chỉ MAC của cả x và y . Khi y nhận gói tin này, nó sẽ trả lời bằng gói CTS nếu như y không biết gì về quá trình truyền dẫn đang diễn ra trong vùng lân cận của nó. Khi nhận được CTS, x bắt đầu truyền tin đến y trên kênh dữ liệu. Khi y bắt đầu nhận tin, nó sẽ gửi gói **âm báo bận** lên kênh điều khiển. Nếu x không nhận được gói CTS trong cửa sổ thời gian, nó sẽ trở lại trạng thái backoff, nơi mà sơ đồ backoff số mũ nhị phân được sử dụng.

Giờ ta sẽ xem xét nút nhận gói RTS của x trên kênh điều khiển. Đó là nút nhận như mong muốn y và còn có một số nút khác nữa. Gọi z là một trong số chúng. Nếu z nhận gói tin đó, nó sẽ phản ứng lại bằng cách gửi đi một gói âm báo bận chồng lên gói CTS của nút y ở nút x và sẽ làm mất hiệu lực gói CTS. Vì vậy, x không thể bắt đầu truyền và việc nhận tin của z không gây xáo trộn. Do gói âm báo bận dài hơn gói CTS nên ta có thể đảm bảo rằng gói CTS thực sự mất hiệu lực. Tiếp theo, ta sẽ xem xét nút nhận mong muốn y . Nếu y biết về quá trình truyền dẫn đang diễn ra trong vùng lân cận, nó sẽ bỏ gói CTS nên x sẽ về trạng thái backoff. Nút y có thể có được phản hồi này bằng cách hoặc cảm nhận kênh dữ liệu, hoặc kiểm tra xem liệu có tạp âm trên kênh điều khiển sau khi nhận gói RTS không? Tạp âm này có thể là RTS hoặc CTS của một nút khác xung đột ở y . Trong trường hợp khác, y trả lời bằng gói CTS và bắt đầu gửi gói âm báo bận ngay khi việc truyền dẫn của x bắt đầu. Hơn nữa, y gửi âm báo bận mỗi khi nó nhận được tạp âm hoặc gói tin có giá trị trên kênh điều khiển để ngăn cản hoạt động của các nút lân cận.

Nút nhận gói RTS trong khi ở trạng thái backoff sẽ bắt đầu thủ tục nhận tin của nó, đó là kiểm tra các điều kiện để gửi CTS.

Khi nào nút có thể đặt bộ thu phát của nó (điều khiển và dữ liệu) vào trạng thái ngủ? Nói một cách khái quát, bất cứ thời điểm nào nút biết là không thể truyền và nhận tin từ các nút lân cận khác thì nó sẽ làm vậy. Tuy nhiên, quyết định có quay trở lại trạng thái ngủ hay không sẽ nảy sinh câu hỏi là khi nào sẽ trở lại trạng thái thức? Quyết định này sẽ dễ thực hiện nếu nút x biết độ dài của quá trình truyền dẫn đang diễn ra, ví dụ như từ nghe lén gói tin RTS hay CTS hoặc phần đầu của gói dữ liệu trên kênh dữ liệu. Tuy nhiên x thường không biết độ dài này, ví dụ như các gói tin bị sai lệch hoặc chu kỳ truyền dẫn bắt đầu khi x vừa ngủ...

Giả sử x thức và nhận thấy rằng kênh dữ liệu bận. Có hai trường hợp có thể xảy ra: hoặc x không có tin cần chuyển hoặc x muốn truyền tin. Trong trường hợp đầu tiên, x muốn quay trở lại trạng thái ngủ và thức dậy đúng lúc kết thúc quá trình truyền dẫn đang diễn ra để có thể ngay lập tức nhận gói tin tiếp theo. Thức dậy vào thời điểm sớm nhất có thể có ưu điểm là tránh được trễ không mong muốn. Tuy nhiên, do x không nghe lén RTS, CTS hoặc phần đầu gói dữ liệu phụ thuộc vào quá trình truyền dẫn đang diễn ra nên nó hoạt động theo **giao thức tìm kiếm** trên kênh dữ liệu để có thể biết được độ dài gói tin đang truyền. Giao thức tìm kiếm làm việc tương tự như thuật toán tìm kiếm nhị phân.

5.4 Các giao thức dựa trên lịch trình

Phần này sẽ giới thiệu một số **giao thức dựa trên lịch trình** (schedule-based protocol), ở đây không cần phải tránh trạng thái nghe rỗi, ví dụ như sử dụng các sơ đồ TDMA thì việc truyền dẫn được ấn định một cách rõ ràng, cơ hội nhận tin của các nút và đưa chúng về trạng thái ngủ có thể thực hiện ở mọi thời điểm khác nhau. Ưu điểm thứ hai của các giao thức dựa trên lịch trình là lịch trình truyền dẫn được tính toán sao cho không có xung đột xảy ra ở các bộ thu và do đó, không yêu cầu các sơ đồ đặc biệt để tránh các giải pháp đầu cuối ản.

Tuy nhiên, các sơ đồ này cũng có một số rủi ro. *Thứ nhất* là việc cài đặt và duy trì lịch trình bao gồm lưu lượng tín hiệu, đặc biệt khi đối mặt với mạng có cấu hình thay đổi. *Thứ hai* là nếu dùng biến thể của TDMA, thời gian được chia thành các khe tương đối nhỏ, cả bộ phát và bộ thu đều đồng ý với việc chia khe thời gian đó để có thể thực sự gặp nhau và tránh được việc chồng lên các khe thời gian khác, mà đây là lý do tạo ra các xung đột. Tuy nhiên, để duy trì đồng bộ thời gian bao gồm cả trường hợp lưu lượng tín hiệu bị tăng lên là một việc khó. *Thứ ba* là lịch trình đó không dễ thích nghi với các tải khác nhau trong một khoảng thời gian nhỏ. Đặc biệt, trong TDMA rất khó nhường các khe thời gian không sử dụng cho các nút lân cận nó. Nhược điểm tiếp theo là lịch trình của nút (có thể là cả của các nút lân cận nó) có thể yêu cầu nhiều mà bộ nhớ mà đây lại là tài nguyên khan hiếm trong một số thiết kế nút cảm biến. Cuối cùng và việc ấn định phân bố các sơ đồ TDMA không có xung đột thực sự là một vấn đề khó

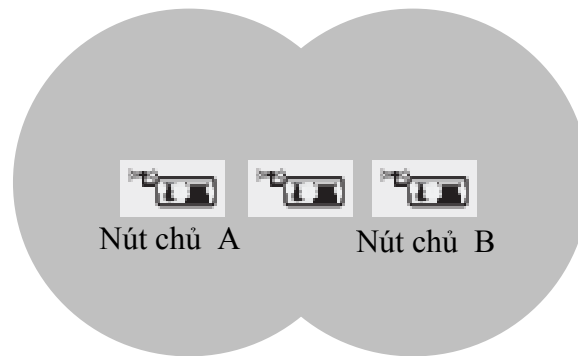
5.4.1 LEACH

Giao thức **phân cấp nhóm thích ứng công suất thấp** LEACH (Low-energy Adaptive Clustering Hierarchy) được giới thiệu bởi **Heinzelman** dựa trên giả thiết là mạng cảm biến thuần nhất mật độ cao, các nút không bị ràng buộc năng lượng và sẽ chuyển dữ liệu đến bộ chứa của nút. Trong LEACH, giao thức MAC dựa trên TDMA tích hợp với giao thức định tuyến đơn giản và phân nhóm.

LEACH phân chia nút thành các **nhóm** (cluster) và trong mỗi nhóm, **nút chủ** (clusterhead) được dùng để tạo và duy trì lịch trình TDMA. Tất cả các nút khác của nhóm được gọi là **nút thành viên** (member node). Với tất cả các nút thành viên, các khe TDMA được ấn định và có thể được dùng để trao đổi dữ liệu giữa các thành viên và nút chủ. Ở đây không có giao tiếp ngang hàng. Ngoài các khe thời gian của chúng, các thành viên có thể ở trạng thái ngủ. Nút chủ kết hợp dữ liệu của các thành viên và truyền chúng đến bộ chứa hoặc đến các nút khác. Do bộ chứa thường ở xa nên nút chủ phải sử dụng năng lượng đáng kể cho việc truyền này. Đối với nút thành viên, việc truyền dữ liệu đến nút chủ thường rẻ hơn rất nhiều so với chuyển trực tiếp đến bộ chứa. Các nút chủ tiêu tốn nhiều năng lượng do nó luôn phải bật và phải có trách nhiệm trong các truyền dẫn khoảng cách lớn. Nếu một nút cố định có vai trò này thì nó sẽ tiêu thụ năng lượng rất nhanh và sau đó, nó sẽ chết, tất cả các thành viên của nó sẽ như “rắn mất đầu” và do đó, chúng trở nên vô dụng. Vì vậy, vai trò chủ đạo này sẽ được thay đổi giữa các nút. Đặc biệt, mỗi nút sẽ quyết định độc lập với các nút khác rằng nó có trở thành nút chủ không nên lưu lượng tín hiệu không phụ thuộc vào việc chọn nút chủ (mặc dù lưu lượng tín hiệu được yêu cầu để kết hợp các nút sau đó). Quyết định này được đưa vào báo cáo khi nút được xem như là nút chủ lần cuối cùng. Giao thức sẽ được xâu

dựng dựa trên cơ sở này, đó là tất cả các nút quyết định xem liệu có trở thành nút chủ tại cùng một thời điểm không và các nút không trở thành nút chủ sẽ phải liên kết với nhau để sau đó trở thành nút chủ. Các nút không trở thành nút chủ chọn nút chủ cho chúng trên cơ sở cường độ tín hiệu thu được. Mạng được chia thành các nhóm sẽ biến đổi theo thời gian và giao thức phải đồng bộ thời gian hoàn toàn.

Sau khi tạo thành các nhóm, mỗi nút chủ sẽ chọn một mã CDMA ngẫu nhiên cho nhóm mình và sẽ thông báo tới các nút thành viên. Điều này sẽ tránh tình trạng khi nút nằm ở biên giới thuộc nút chủ A nhưng do méo trong quá trình truyền lại thuộc nhóm B như hình 5.9.



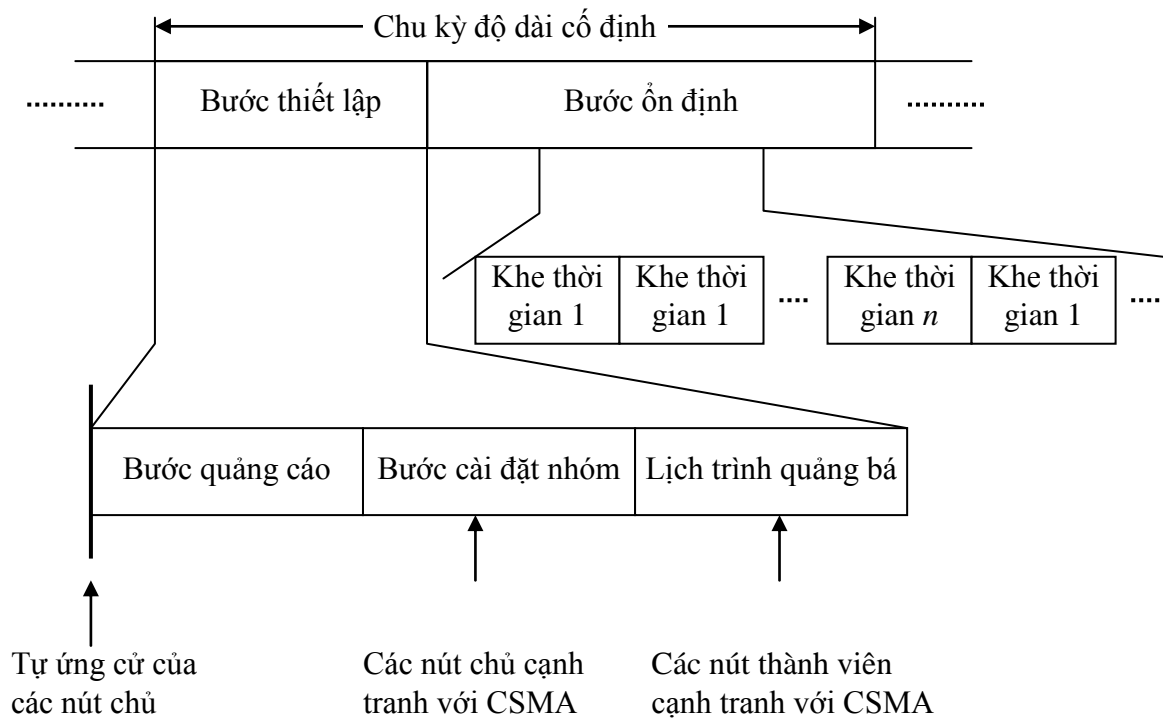
Hình 5.9 Nhiễu giữa các nhóm

Tham số mạng tối hạn là phần trăm nút làm nút chủ. Nếu chỉ có một số ít làm nút chủ thì khoảng cách mong muốn giữa nút thành viên và nút chủ trở nên lớn hơn, do đó nút thành viên phải sử dụng nhiều năng lượng để đến được nút chủ của chúng khi đảm bảo đảm bảo tỉ số BER đã cho. Nói cách khác, nếu có nhiều nút chủ thì sẽ phải sử dụng nhiều năng lượng hơn cho việc truyền dẫn từ các nút chủ đến bộ chứa. Vì vậy, tồn tại tỉ lệ phần trăm tối ưu nút chủ là $\approx 5\%$. Nếu giá trị tối ưu này được chọn thì LEACH có thể giảm được bảy đến tám lần năng lượng sử dụng so với trường hợp mỗi nút truyền trực tiếp dữ liệu của mình đến bộ chứa, và sẽ là bốn đến tám lần so với trường hợp các gói tin chuyển tiếp trong đa bước nhảy. Do LEACH phân bổ vai trò của nút chủ một cách công bằng đối với tất cả các nút nên chúng thường chết cùng lúc.

Giao thức được tổ chức thành các chu kỳ và mỗi chu kỳ được chia thành bước thiết lập và bước ổn định (hình 5.10). **Bước thiết lập** bắt đầu bằng việc tự ứng cử của các nút làm nút chủ. Trong **bước thông báo**, các nút chủ báo cáo với các nút lân cận qua gói tin thông báo. Các nút chủ cạnh tranh đường truyền bằng cách sử dụng giao thức CSMA để tránh vấn đề đầu cuối ẫn. Các nút không phải là nút chủ chọn gói thông báo nào có cường độ tín hiệu thu được mạnh nhất. Trong bước thiết lập nhóm tiếp theo, các thành viên thông báo cho nút chủ biết “việc gia nhập” và lại sử dụng giao thức CSMA. Sau bước thiết lập nhóm, nút chủ biết số lượng thành viên và cách nhận dạng chúng. Nó xây dựng một lịch trình TDMA, chọn mã CDMA một cách ngẫu nhiên và thông báo thông tin này trong bước con thông báo lịch trình. Sau đó sẽ bắt đầu bước ổn định.

Do xung đột giữa các gói tin thông báo hoặc nút chủ nên giao thức không thể đảm bảo rằng mỗi nút không phải là nút chủ sẽ thuộc nhóm. Tuy nhiên, có thể đảm bảo rằng các nút sẽ thuộc nhiều nhất là một nhóm.

Nút chủ được bật trong cả chu kỳ và các nút thành viên phải bật trong bước cài đặt và đôi khi trong bước ổn định, phụ thuộc vào vị trí của chúng trong lịch trình TDMA của nhóm.



Hình 5.10 Tổ chức của chu kỳ LEACH

Với các giao thức đã được mô tả ở trên, LEACH không thể bao phủ một vùng địa lý rộng lớn khoảng vài dặm vuông hoặc lớn hơn vì nút chủ chỉ được cách bộ chứa khoảng hai dặm là đã không đủ năng lượng để tới bộ chứa, chưa kể đến việc tỉ số BER lúc đó sẽ thấp. Nếu có thể sắp xếp sao cho nút chủ này có thể sử dụng các nút chủ khác để truyền tiếp thì hạn chế này có thể khắc phục phần nào.

5.4.2 SMACS

Giao thức **điều khiển truy cập môi trường tự tổ chức cho các mạng cảm biến** (SMACS - Self-Organizing Medium Access Control for Sensor Networks) được mô tả bởi Sohrabi và Pottie là một phần của giao thức WSN phù hợp với MAC địa chỉ, phát hiện vùng lân cận, gắn liền với các nút di chuyển, giao thức định tuyến đa bước nhảy và giao thức định tuyến cục bộ cho mục đích xử lý tín hiệu hợp tác.

SMAC về cơ bản kết hợp việc phát hiện vùng lân cận và sự ổn định của các lịch trình TDMA của các nút. SMACS dựa trên các giả thiết sau:

- Phổ được chia thành một số kênh và mỗi nút có thể điều chỉnh bộ thu phát của mình đến một giá trị bất kỳ, giả thiết rằng các mã CDMA được sử dụng.

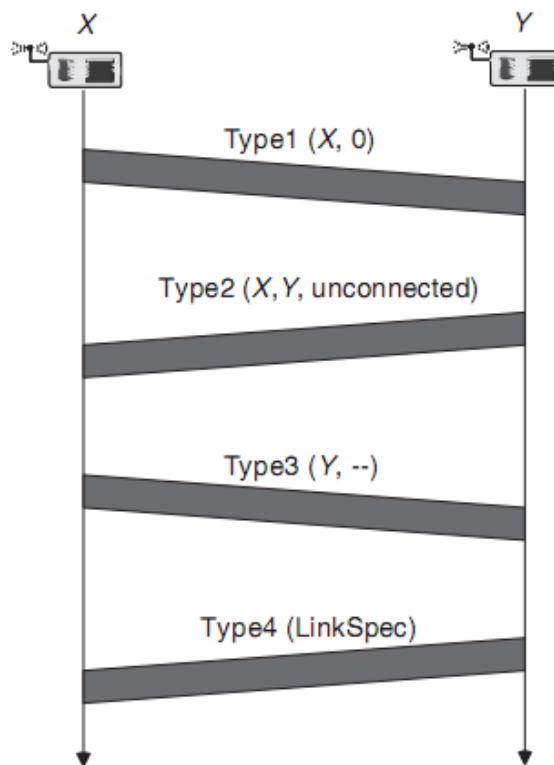
- Hầu hết các nút trong mạng cảm biến là đứng yên và được ấn định một giá trị trong thời gian khá dài.

- Mỗi nút chia thời gian của nó thành các siêu khung có độ dài cố định (có độ rộng T_{frame} giây) nhưng không cần thiết phải cùng pha. Tuy nhiên, tất cả các nút phải có cùng độ dài siêu khung và yêu cầu đồng bộ về mặt thời gian. Các siêu khung cũng được chia thành các khe thời gian nhưng điều này không chính xác vì các truyền dẫn không bị giới hạn là chỉ xảy ra trong một khe thời gian.

Mục tiêu của SMACS là phát hiện các nút lân cận và cài đặt các **tuyến** và các **kênh truyền** dẫn đến chúng. Tuyến được xác định theo tất cả các gói tin đã cho được truyền theo một hướng. Hơn nữa, tuyến được sử dụng bởi khe TDMA ở một trong hai điểm đầu cuối. Khi hai nút muốn hoạt động theo hình thức hai chiều thì yêu cầu phải có hai tuyến. Một nút sẽ có một **khe nhận** và một **khe truyền** tới các nút khác. Ấn định các tuyến để đảm bảo không có xung đột xảy ra ở các bộ thu. Để có được điều này, SMACS quan tâm rằng với nút đơn, các khe thời gian của các tuyến khác nhau sẽ không chồng nhau. Với mỗi tuyến ngẫu nhiên, một lượng lớn kênh tần số/ mã CDMA sẽ được lựa chọn và sử dụng. Nó không yêu cầu nút và các nút lân cận truyền ở các thời điểm hoàn toàn khác nhau. Tuy nhiên, trong trường hợp này, chúng phải truyền đến các bộ thu khác nhau và phải sử dụng các mã/ tần số khác nhau. Sau khi cài đặt tuyến, các nút phải thức dậy định kỳ (một lần trên siêu khung) trong các khe thời gian nhận tương ứng với bộ thu được điều chỉnh để phù hợp với tần số hoặc mã CDMA. Khe thời gian truyền chỉ được sử dụng khi có yêu cầu.

Nếu sử dụng sơ đồ cục bộ thay vì ấn định toàn bộ thì nên tránh nhiệm vụ truyền tin từ vùng lân cận đến nút trung tâm phản hồi kết quả tính toán.

Đối với vùng bao phủ lân cận và cài đặt tuyến, ta sẽ xét bốn trường hợp. Giả sử nút x và y muốn cài đặt tuyến và x được bật trước.



Hình 5.11 SMACS: cài đặt tuyến cho hai nút đơn

Trường hợp đầu tiên, ta giả sử hoặc x hoặc y không có nút lân cận như hình 5.11. Nút x nghe ở dải tần số cố định trong một khoảng thời gian ngẫu nhiên. Nếu không nhận được gì trong khoảng thời gian này thì x sẽ gửi một thông báo mời là $TYPE1(x, unattached)$ nhằm cho biết nhận dạng nút và số nút lân cận của nó. Khi nút lân cận z bất kỳ của x nhận được thông báo này, nó sẽ đợi một cách ngẫu nhiên nhưng nằm trong giới hạn thời gian và trả lời với thông báo $TYPE2(x, z, n)$ nhằm cho biết địa chỉ của nó, địa chỉ của x và số nút lân

cận n . Giả sử rằng nút y trả lời không chính xác trước bằng $\text{TYPE2}(x, y, \text{unattached})$ và x nhận đúng thông báo này. Do y gửi câu trả lời trước nên x mời y tạo tuyến bằng cách gửi thông báo $\text{TYPE3}(y, -)$, mang nhận dạng của nút “chiến thắng” y và các tham số của nó. Thông báo này được gửi khi kết thúc thời kỳ cạnh tranh của thông báo trả lời $\text{TYPE2}(x, z, n)$. Giờ y đã biết rằng nó đã được lựa chọn (i) và (ii) nó có thể lựa chọn khe thời gian bất kỳ mà nó muốn do x hoặc y không có tuyến được chỉ định.

Nút y trả lời x bằng tuyến đã được chỉ định, đó là hai khe thời gian và tần số/ mã bằng cách sử dụng thông báo $\text{TYPE2}(x, y, \text{LinkSpec})$. Các khe thời gian chỉ định có mỗi thời gian chung do nút y chấp nhận pha siêu khung của x dựa trên việc nhận thông báo $\text{TYPE3}(y, -)$. Do đó, các nút lân cận được nhận ra trong trường hợp đầu tiên sẽ cùng pha (và cùng chu kỳ).

Nút z sẽ quay trở lại trạng thái ngủ và cố đợi thêm một thời gian nữa. Các nút định kỳ lặp lại lời mời của chúng bằng các thông báo $\text{TYPE1}(x, \text{unattached})$.

Trường hợp thứ hai, nút x có sẵn các nút lân cận nhưng nút chiến thắng y ở gần. Vì vậy, x gửi thông báo $\text{TYPE1}(x, \text{unattached})$ và y cố gắng trả lời trước tiên bằng thông báo $\text{TYPE2}(x, y, \text{unattached})$ của nó. Sau đó, nút x biết rằng nó có thể tự do sắp xếp tuyến đến y do nó ở gần x . Nút x chọn hai khe thời gian và tần số phù hợp, gửi thông báo $\text{TYPE3}(y, \text{LinkSpec})$ đến y . Nó nhận pha siêu khung của x và trả lời bằng thông báo $\text{TYPE2}(x, y, -)$ mang chỉ định của tuyến rỗng (có nghĩa là chỉ định tuyến của x đã được chấp nhận).

Trường hợp thứ ba, nút x thực sự không có nút lân cận, vì vậy y trả lời thông báo $\text{TYPE1}(x, \text{unattached})$ của x bằng $\text{TYPE2}(x, y, \text{unattached})$. Nút x tiếp tục bằng cách gửi thông báo $\text{TYPE3}(y, -)$ mà không chỉ định tuyến đến y và y sẽ điều chỉnh để lựa chọn các khe thời gian và tần số. Sau đó, y gửi trả lại thông báo $\text{TYPE2}(x, y, \text{unattached})$ đến x .

Trường hợp thứ tư, cả x và y đều có tuyến với các nút khác và các siêu khung của nó chúng thường không thẳng hàng. Vì vậy, x gửi thông báo $\text{TYPE1}(x, \text{unattached})$ và y trả lời bằng $\text{TYPE2}(x, y, \text{unattached})$. Nút x trả lời với thông báo $\text{TYPE3}(y, \text{Schedule})$ bao gồm toàn bộ lịch trình và thông tin thời gian cho phép y xác định sự dịch pha giữa các siêu khung của x và y . Sau khi nhận được thông tin này, nút y xác định các khe thời gian còn trống giữa hai lịch trình, chúng không nhất thiết phải thẳng hàng với biên giới các khe thời gian ở một trong hai lịch trình.

Giao thức này cho phép cài đặt các tuyến tĩnh giữa các nút cố định. Do quá trình xác định vùng bao phủ được lặp lại theo thời gian nên giao thức này phù hợp để thay đổi cấu hình mạng.

5.5 Giao thức MAC của IEEE 802.15.4

Viện Công nghệ kỹ thuật Điện và Điện tử (IEEE - Institute of Electrical and Electronics Engineers) đã hoàn thành chuẩn IEEE 802.15.4 vào tháng 10 năm 2003. Chuẩn bao trùm lớp vật lý và lớp MAC của **mạng không dây cá nhân** (WPAN - Wireless Personal Area Network) tốc độ thấp. Tuy nhiên, rất nhiều người bị nhầm lẫn giữa IEEE 802.15.4 và ZigBee, một chuẩn nổi bật của khối liên minh ZigBee. ZigBee sử dụng các dịch

vụ của IEEE 802.15.4 và thêm vào cấu trúc mạng (các mạng hình sao, mạng ngang hàng/mesh, mạng hình cây), tính bảo mật, các dịch vụ ứng dụng...

Zigbee là gì?

Zigbee là một tiêu chuẩn được định nghĩa là tập hợp các giao thức giao tiếp mạng không dây khoảng cách ngắn có tốc độ truyền dữ liệu thấp. Các thiết bị không dây dựa trên chuẩn Zigbee hoạt động trên 3 dải tần số là 868MHz, 915 MHz và 2.4GHz. với những đặc điểm chính :

- Tốc độ truyền dữ liệu thấp 20-250Kbps
- Sử dụng công suất thấp, ít tiêu hao điện năng
- Thời gian sử dụng pin rất dài
- Cài đặt, bảo trì dễ dàng
- Độ tin cậy cao
- Có thể mở rộng đến 65000 node
- Chi phí đầu tư thấp

Zigbee chủ yếu được nhắm tới cho các ứng dụng chạy pin có tốc độ dữ liệu thấp, chi phí nhỏ, và thời gian sử dụng pin dài. Trong nhiều ứng dụng của Zigbee, tổng thời gian mà thiết bị không dây thực sự hoạt động rất ít; thiết bị sử dụng hầu hết thời gian của nó trong chế độ tiết kiệm năng lượng, hay chế độ ngủ (sleep mode). Kết quả là, Zigbee cho phép các thiết bị có khả năng hoạt động trong nhiều năm trước khi cần phải nạp lại pin hoặc thay pin mới.

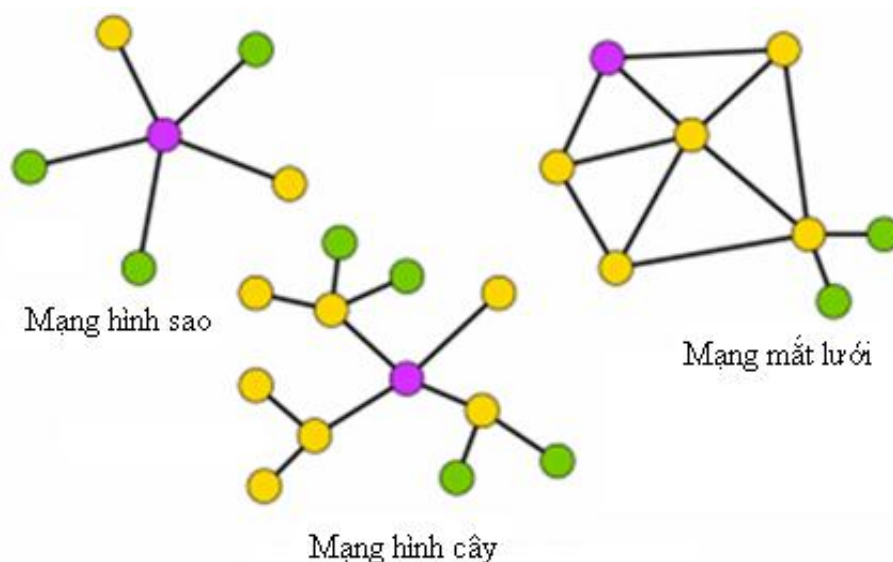


Hình 5.12 Các ứng dụng của Zigbee

Một ứng dụng của Zigbee là theo dõi bệnh nhân tại nhà. Ví dụ, huyết áp và nhịp tim của một bệnh nhân được đo bởi các thiết bị đeo trên người. Bệnh nhân mang một thiết bị Zigbee tập hợp các thông tin liên quan đến sức khỏe như huyết áp và nhịp tim. Sau đó dữ liệu được truyền không dây đến một máy chủ địa phương, có thể là một máy tính cá nhân đặt trong nhà bệnh nhân, nơi mà việc phân tích ban đầu được thực hiện. Cuối cùng, thông tin quan trọng được chuyển tới y tá của bệnh nhân hay nhân viên vật lý trị liệu thông qua Internet để phân tích sâu hơn.

Khái niệm sử dụng giao tiếp không dây để tập hợp thông tin hay thực hiện những tác vụ điều khiển nhất định trong nhà hay trong nhà máy thì không mới. Có vài chuẩn giao tiếp bao gồm IEEE 802.11 Wireless Local Area Network (WLAN) và Bluetooth. Mỗi trong số

các chuẩn này đều có ưu điểm của nó trong những ứng dụng đặc biệt. Chuẩn Zigbee được phát triển đặc biệt để nhắm tới chi phí cài đặt thấp của các mạng không dây có tốc độ truyền dữ liệu thấp với mức tiêu thụ năng lượng rất nhỏ. Chuẩn Zigbee giúp giảm chi phí cài đặt bằng cách đơn giản hóa các giao thức liên lạc và giảm tốc độ truyền dữ liệu. Yêu cầu tối thiểu để sử dụng các đặc tính kỹ thuật của Zigbee và IEEE 802.15.4 thì khá nhẹ khi so với các chuẩn khác như IEEE 802.11, giảm độ phức tạp và chi phí lắp đặt các bộ thu phát theo chuẩn Zigbee.



Hình 5. 13 Các kiểu mạng của ZigBee

IEEE 802.15.4 được dùng trong WSN, tự động hoá ngôi nhà, mạng gia đình, các thiết bị tuyến đến PC... Hầu hết các ứng dụng này chỉ yêu cầu tốc độ bit từ thấp đến trung bình (lên đến hàng trăm kbps), trễ trung bình ở mức vừa phải mà không có yêu cầu quá chặt chẽ và với các nút đã biết thì cần phải giảm năng lượng tiêu thụ đến mức tối thiểu. Lớp vật lý yêu cầu tốc độ bit là 20 kbps (kênh đơn ở dải tần 868 – 868.6 MHz), 40 kbps (10 kênh ở dải tần từ 905 đến 928 MHz) và 250 kbps (16 kênh trong dải ISM 2.4 GHz giữa 2.4 và 2.485 GHz với khoảng cách 5 MHz giữa các tần số trung tâm). Có tổng cộng 27 kênh có thể dùng được nhưng giao thức MAC chỉ sử dụng một trong số các kênh này tại một thời điểm. Đây không phải là giao thức đa kênh.

Giao thức MAC kết hợp cả sơ đồ dựa trên lịch trình và sơ đồ dựa trên sự cạnh tranh. Đây là giao thức không đối xứng đối với các loại nút khác nhau và các luật khác nhau được sử dụng.

5.5.1 Kiến trúc mạng, loại và vai trò của nút.

Lớp MAC phân ra làm hai loại nút:

- FFD (Full Function Device – thiết bị đầy đủ chức năng) có thể hoạt động theo ba vai trò: nó có thể là một **bộ điều phối mạng cá nhân** (PAN coordinator, PAN - Personal Area Network), một **bộ điều phối** (coordinator) đơn giản hoặc một **thiết bị** (device). Đây là những thiết bị hỗ trợ đầy đủ các chức năng theo chuẩn của IEEE 802.15.4 và có thể đảm nhận bất kỳ vai trò nào trong hệ thống.
- RFD (Reduced Function Device - thiết bị giới hạn chức năng) chỉ có thể hoạt động

như một thiết bị. Đây là những thiết bị giới hạn một số chức năng (chỉ giao tiếp được với các FFD, áp dụng cho các ứng dụng đơn giản như tắt mở đèn).

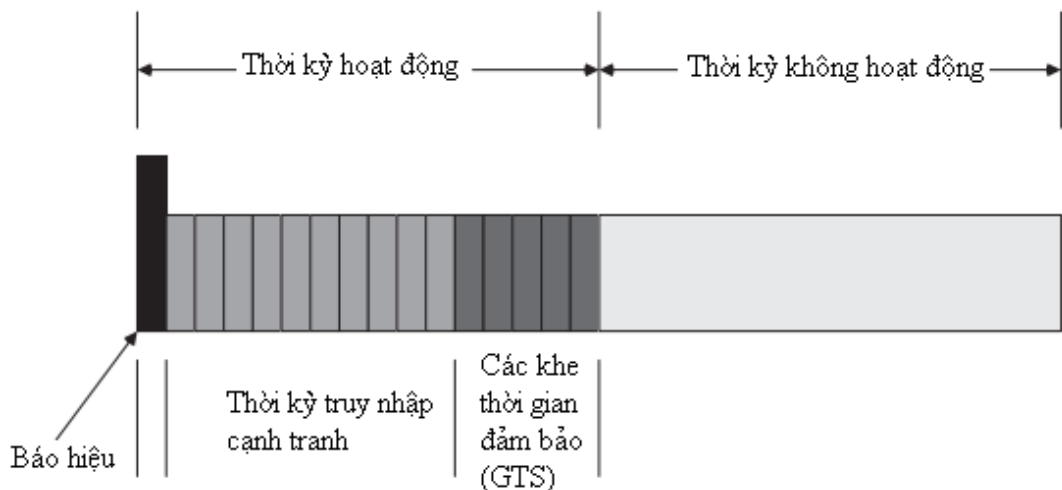
Thiết bị phải kết hợp với nút điều phối (nó phải là một FFD) và chỉ giao tiếp với nó, cách này tạo nên **mạng hình sao** (star network). Các bộ điều phối có thể hoạt động ở dạng ngang hàng và nhiều bộ điều phối có thể tạo thành một PAN. PAN được xác định bởi **bộ định dạng PAN** (PAN Identifier) 16 bit và một trong số chúng phải được thiết kế như một bộ điều phối PAN.

Bộ điều phối có các nhiệm vụ sau:

- Nó cố gắng liệt kê các thiết bị liên quan. Các thiết bị được yêu cầu phải liên kết hoặc không liên kết một cách rõ ràng với bộ điều phối bằng cách sử dụng các gói tín hiệu nào đó.
- Nó ấn định các địa chỉ ngắn cho các thiết bị đó. Tất cả các nút IEEE 802.15.4 có 64 bit địa chỉ thiết bị. Khi thiết bị kết hợp với bộ điều phối, nó có thể có yêu cầu ấn định địa chỉ ngắn 16 bit cho tất cả các giao tiếp sau đó giữa thiết bị và bộ điều phối. Địa chỉ được ấn định cho biết gói đáp ứng liên quan được đưa ra bởi bộ điều phối.
- Trong chế độ báo hiệu của IEEE 802.15.4, nó truyền các gói **báo hiệu khung** một cách thường xuyên nhằm thông báo bộ định dạng PAN, danh sách các khung còn tồn tại và các tham số khác. Hơn nữa, bộ điều phối có thể chấp nhận và xử lý yêu cầu để giữ khác khe thời gian cố định cho các nút và các chỉ định phải được thể hiện trong báo hiệu.
- Nó thay đổi gói dữ liệu bằng các thiết bị và các bộ điều phối.

5.5.2 Kiến trúc của siêu khung

Bộ điều phối của mạng hình sao hoạt động ở chế độ báo hiệu, nó thiết lập cách truy cập kênh truyền và truyền dẫn dữ liệu với sự trợ giúp của kiến trúc siêu khung như hình 5.14.



Hình 5.14 Cấu trúc siêu khung của IEEE 802.15.4

Tất cả các siêu khung đều có cùng độ dài. Bộ điều phối bắt đầu mỗi siêu khung bằng cách gửi một gói báo hiệu khung. Báo hiệu khung bao gồm **đặc điểm của siêu khung** mô tả độ dài của các thành phần khác nhau sau của siêu khung:

- Siêu khung được chia thành thời kỳ hoạt động (active) và thời kỳ không hoạt động

(inactive). Ở thời kỳ không hoạt động tất cả các nút bao gồm bộ điều phối có thể tắt bộ thu phát của nó và chuyển sang trạng thái ngủ. Các nút sẽ thức dậy ngay trước khi kết thúc thời kỳ không hoạt động và nhận báo hiệu tiếp theo. Có thể tránh thời kỳ này.

- Giai đoạn hoạt động được chia thành 16 khe thời gian. Khe thời gian đầu tiên được sử dụng bởi khung báo hiệu và các khe thời gian còn lại được chia ra trong **thời kỳ truy nhập cạnh tranh** (CAP - Contention Access Period) theo số **khe thời gian đảm bảo** (GTSs - Guaranteed Time Slots) có giá trị lớn nhất bằng bảy.

Độ dài của thời kỳ hoạt động và không hoạt động bằng bộ dài của một khe thời gian và sử dụng các khe GTS có thể cấu hình được.

Bộ điều phối hoạt động trong toàn bộ thời kỳ hoạt động. Các thiết bị liên quan trong pha GTS chỉ hoạt động trong các khe thời gian được ấn định cho chúng. Trong tất cả các khe GTS khác, chúng có thể chuyển sang trạng thái ngủ. Trong CAP, một thiết bị có thể tắt bộ thu phát không có dữ liệu để truyền hoặc không có dữ liệu để mang về từ bộ điều phối.

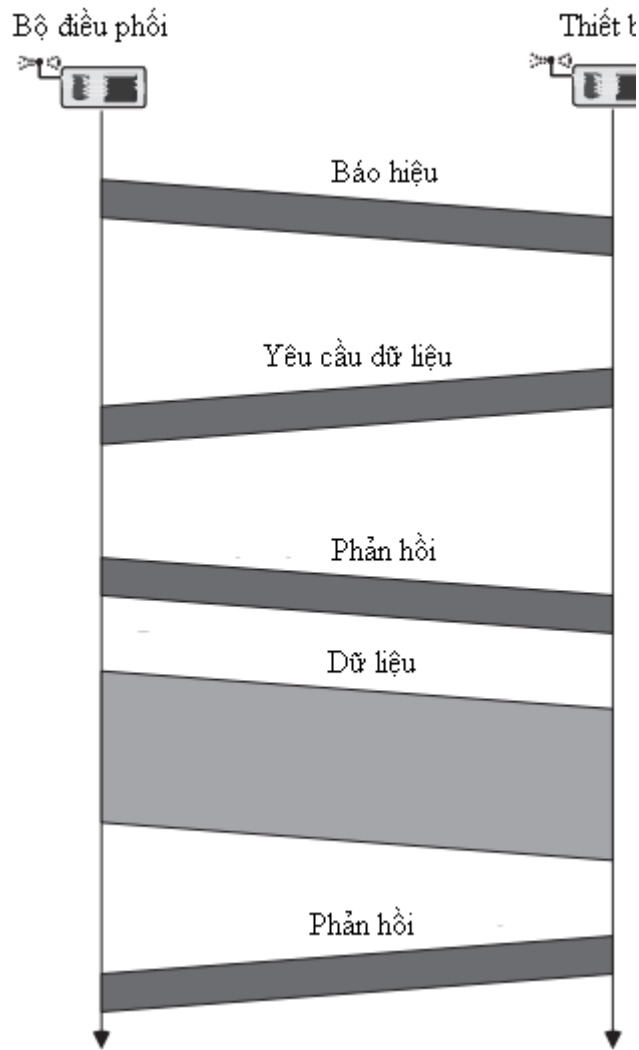
Như vậy, bộ điều phối làm việc ít hơn các thiết bị và bản thân giao thức là không đối xứng. Giao thức được tối ưu trong các trường hợp khi mà các cảm biến bị ràng buộc năng lượng được gắn với các nút không bị ràng buộc năng lượng.

5.5.3 Giao thức Slotted CSMA-CA

Khi các nút gửi dữ liệu hay các gói quản lý/ điều khiển trong CAP, chúng sử dụng giao thức Slotted CSMA-CA. Giao thức không chuẩn bị cho các tình huống đầu cuối ẩn, ví dụ như không có bắt tay RTS/ CTS. Để giảm xác suất xảy ra xung đột, giao thức sử dụng các trễ ngẫu nhiên, đó là giao thức CSMA-CA (CSMA with Collision Avoidance - CSMA tránh xung đột). Sử dụng các trễ ngẫu nhiên như vậy cũng là một phần của các giao thức được mô tả trong phần 5.3.1. Để mô tả hoạt động của nó một cách chi tiết hơn, ta hãy xem hình 5.15.

Các khe thời gian tạo thành CAP được chia thành các khe thời gian nhỏ hơn, được gọi là **các thời kỳ backoff**. Một thời kỳ backoff có độ dài tương đương với 20 thời gian ký hiệu kênh và các khe được xem xét bởi giao thức Slotted CSMA-CA chỉ là các thời kỳ backoff này.

Thiết bị duy trì ba biến là NB , CW và BE . Biến NB tính số backoff, CW biểu thị cỡ cửa sổ sổ nghẽn mạng hiện thời và BE là số mũ backoff hiện thời. Dựa trên việc đến của gói tin mới được truyền, các biến này có giá trị khởi tạo là $NB = 0$, $CW = 2$ và $BE = macMinBE$ (với $macMinBE$ là tham số của giao thức). Thiết bị đợi thời kỳ backoff tiếp theo và chọn một số nguyên ngẫu nhiên γ nằm trong khoảng $[0, 2^{BE} - 1]$. Thiết bị đợi trong γ thời kỳ backoff và thực hiện hoạt động cảm nhận sóng mang (nó có nghĩa như là **kiểm tra kênh truyền có rỗi không** (CCA - Clear Channel Assessment)). Nếu môi trường rỗi, thiết bị giảm CW , đợi ranh giới của thời kỳ backoff tiếp theo và cảm nhận kênh truyền một lần nữa. Nếu kênh vẫn rỗi thì thiết bị cho rằng nó đã thắng và bắt đầu truyền dữ liệu. Nếu các hoạt động CCA cho biết môi trường bận thì số backoff NB và số mũ backoff BE tăng và CW được đặt trở lại $CW = 2$. Nếu NB vượt quá ngưỡng thì thiết bị sẽ giảm khung và thông báo lỗi. Nói cách khác, thiết bị lại chọn một số nguyên γ nằm trong khoảng $[0, 2^{BE} - 1]$ và đợi trong γ khe backoff. Tất cả các bước tiếp theo sẽ được lặp lại như trên.



Hình 5.15 Bắt tay giữa bộ điều phối và thiết bị khi thiết bị tìm kiếm gói tin

5.6 Tổng kết

Một số giao thức MAC khác nhau cho WSN đã được thảo luận ở chương này. Tất cả các đặc tính quan trọng nhất đều được thiết kế với mục đích chính là tiết kiệm năng lượng, ngoài ra còn cần phải có đến độ trễ nhỏ, công suất cao. Không có giao thức MAC tốt nhất cho mọi trường hợp. Lựa chọn được giao thức phù hợp phụ thuộc vào ứng dụng, tải, cách triển khai (mạng cảm biến mật độ thưa thớt hay dày đặc) và hình thức tiêu thụ năng lượng của phần cứng ở phía dưới, ví dụ như chi phí cho việc truyền tin, nhận tin, chuyển giữa các trạng thái, thời điểm thức, năng lượng thức dậy từ trạng thái ngủ...

CÂU HỎI ÔN TẬP CHƯƠNG 5

1. Các yêu cầu ràng buộc thiết kế cho các giao thức MAC không dây?
2. Phân loại các giao thức MAC?
3. Vấn đề bắt tay RTS/CTS trong MAC?
4. Mối quan tâm khi thiết kế MAC cho WSN?
5. Phân biệt STEM-B và STEM-T?
6. Giới thiệu về S-MAC?
7. Chức năng và đặc điểm của giao thức thiết bị trung gian?
8. Giới thiệu khái quát các giao thức theo kiểu cạnh tranh?
9. Chức năng và đặc điểm của LEACH?
10. Chức năng và đặc điểm của SMACS?
11. Đặc điểm mạng Zigbee?
12. Giao thức MAC của IEEE 802.15.4?

CHƯƠNG 6. CÁC GIAO THỨC LỚP LIÊN KẾT

Mục đích của chương

Mục đích quan trọng nhất của lớp liên kết là hình thành và duy trì các tuyến thông tin trực tiếp giữa các nút cảm biến lân cận, đảm bảo việc truyền tin một cách hiệu quả và tin cậy qua các tuyến này. Độ tin cậy phải được đảm bảo mặc dù có tình trạng lỗi biến thiên theo thời gian trên tuyến không dây, nhiều cơ chế với đặc điểm và đặc tính tiêu thụ năng lượng khác nhau phải được chú trọng. Ta sẽ thảo luận các cơ chế này và chỉ ra mối quan hệ đơn giản giữa độ tin cậy, năng lượng và tốc độ lỗi: tốc độ lỗi tăng hay độ tin cậy tăng sẽ làm cho mức tiêu thụ năng lượng tăng? Vì vậy, độ tin phải được xem xét một cách cẩn thận dựa trên từng yêu cầu của ứng dụng và các tình trạng lỗi.

Lớp liên kết dữ liệu (DLL – Data Link Layer), hay lớp liên kết, có nhiệm vụ đảm bảo giao tiếp **tin cậy** giữa các nút cảm biến lân cận ở mạng (cảm biến) không dây trong dải vô tuyến. Vì vậy, **giám sát lỗi** là một trong những nhiệm vụ quan trọng nhất. Kỹ thuật giám sát lỗi được thiết kế với mục đích có được độ tin cậy nào đó trong quá trình truyền các gói tin, kể cả khi có lỗi trên kênh truyền. Trong WSN, ta có thêm yêu cầu để có được độ tin cậy này, đó là mức tiêu thụ năng lượng phải là nhỏ nhất. Mấu chốt của việc điều khiển và cơ chế hiệu chỉnh quan trọng nhất mà cơ chế giám sát lỗi thường dùng là độ dư thừa, việc truyền lại, lựa chọn các tham số truyền như kích thước gói tin và công suất phát. Một đặc điểm quan trọng của các cơ cấu này là nó không làm việc giống nhau ở tất cả các kênh có lỗi. Kênh bị ảnh hưởng bởi mô hình BSC đơn giản (lỗi bit độc lập với hằng số xác suất lỗi p) sẽ thực sự khác biệt so với các kênh có lỗi chùm, đặc tính và mức tiêu thụ năng lượng phụ thuộc vào loại kênh.

Tuy nhiên, lớp liên kết có nhiều nhiệm vụ hơn là chỉ đảm bảo độ tin cậy cho việc truyền dẫn. Chương này sẽ giới thiệu tổng quát các chức năng khác nhau của lớp liên kết và thảo luận các khía cạnh quan trọng của WSN.

6.1 Các nhiệm vụ và yêu cầu cơ bản

DLL nằm trên dịch vụ truyền và nhận gói tin được cung cấp bởi lớp MAC và đưa các dịch vụ này đến lớp mạng và các lớp cao hơn khác. Đặc biệt, lớp mạng có thể sử dụng dịch vụ lớp liên kết để phân phát gói tin, hỗ trợ định tuyến và các hoạt động giám sát cấu hình mạng.

Một trong những nhiệm vụ quan trọng nhất của lớp liên kết là tạo ra tuyến thông tin tin cậy cho việc truyền tin giữa các nút cảm biến lân cận cùng nằm trong dải vô tuyến. Nó có thể được xem xét dựa trên các đặc điểm sau:

- **Đóng khung:** Dữ liệu được sắp xếp và định hình thành các **gói** hoặc **khung**, nó bao gồm dữ liệu và các thông tin liên quan đến giao thức của lớp liên kết (và các lớp dưới lớp MAC). Hình dạng và kích thước của gói tin dựa vào các thông số là công suất và mức tiêu thụ năng lượng.

- **Giám sát lỗi:** Chức năng này cần thiết cho tất cả các phương tiện truyền dẫn, đặc biệt là các phương tiện vô tuyến bị méo dạng sóng truyền khiến các gói tin đã phát không thể sử dụng được. Các cơ chế giám sát lỗi có thể bù ảnh hưởng của lỗi. Hiệu quả và mức tiêu thụ năng lượng của các cơ chế lỗi khác nhau phụ thuộc vào loại lỗi trên đường truyền.

- **Giám sát lưu lượng:** Bộ thu của các gói tin có thể tạm thời không muốn nhận chúng, ví dụ như do thiếu không gian bộ đệm hoặc khả năng xử lý. Các cơ chế giám sát lưu lượng có thể tạo ra tín hiệu bất kỳ khiến bộ phát làm chậm lại quá trình truyền dẫn. Do nhiều nút cảm biến chỉ được thiết kế có tốc độ bit thấp nên đây là một lý do để cho rằng giám sát lưu lượng không được đề cập đến trong các mạng cảm biến. Điều này thực sự đúng khi đề cập đến khả năng bộ chứa của các nút. Do đó, giám sát lưu lượng không được nghiên cứu một cách rõ ràng trong WSN và dường như với các kỹ thuật đã có, một số được tích hợp trong các giao thức giám sát lỗi (sơ đồ của sơ trượt) lại có hiệu quả.

- **Quản lý liên kết:** Kỹ thuật này bao gồm phát hiện, cài đặt, duy trì và kết thúc liên kết với các nút lân cận. Ví dụ như do thiếu không gian bộ đệm hoặc khả năng xử lý. Một phần quan trọng của quá trình duy trì liên kết là đánh giá chất lượng của liên kết để dùng cho các giao thức lớp cao hơn như cho các quyết định định tuyến hay cho mục đích giám sát cấu hình mạng. Đánh giá chất lượng liên kết được thảo luận trong phần 6.4.2. Một số thủ tục cho việc cài đặt và kết thúc liên kết đã được giới thiệu trong các giao thức MAC, ví dụ như trong SMACS ở chương 5.

6.2 Giám sát lỗi

Trong chương này ta sẽ thảo luận về một số phương pháp giám sát lỗi, chúng xử lý lỗi truyền dẫn trong các tuyến vô tuyến để dùng cho một dịch vụ nào đó. Dịch vụ truyền dữ liệu được cung cấp bởi lớp liên kết có thể được đặc trưng bởi các thuộc tính sau:

- **Không lỗi:** Thông tin mà lớp liên kết của nút nhận đưa cho người sử dụng phải không có lỗi, tức là các bit truyền phải được khôi phục một cách chính xác.

- **Sắp xếp theo chuỗi:** Nếu người sử dụng DLL của bộ phát chuyển giao hai gói tin A và B theo chuỗi thì DLL của bộ thu không bao giờ chuyển B đến người sử dụng trước A , tất cả các kết quả đều phải là A trước B và chỉ có A và B mà thôi, không một gói tin nào được phép gửi nữa.

- **Không có bản sao:** Người sử dụng DLL của bộ thu phải có cùng một thông tin tại tất cả các thời điểm.

- **Không mất tin:** Người sử dụng DLL của bộ thu phải có tin ít nhất một lần.

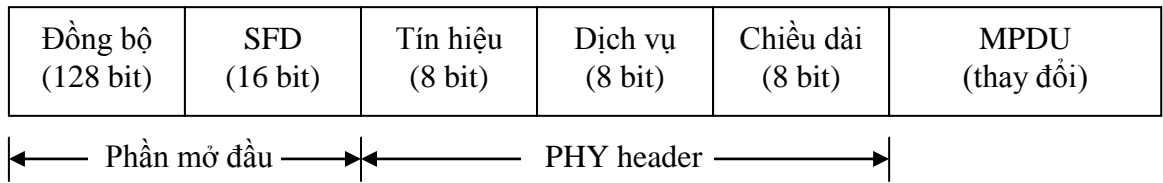
Ngoài các thuộc tính trên là các hạn chế về độ trễ và năng lượng, có nghĩa là trễ trong DLL và các lớp thấp hơn hoặc năng lượng sử dụng bởi DLL và các lớp thấp hơn bị giới hạn bởi một giá trị đã cho.

Các kỹ thuật giám sát lỗi quan trọng nhất là **sự hiệu chỉnh phòng lỗi** (FEC - Forward Error Correction) và **yêu cầu lặp tự động** (ARQ - Automatic Repeat Request), sự phối hợp của chúng sẽ được giới thiệu trong phần 6.2.2 và 6.2.3. Các giao thức ARQ đánh địa chỉ tất cả các dịch vụ mong muốn (không lỗi, sắp xếp theo chuỗi, không có bản sao và không mất tin) trong khi phương pháp FEC lại chủ yếu tập trung vào việc làm thế nào để không có lỗi truyền dẫn.

6.2.1 Nguyên nhân và các đặc tính của sai số truyền dẫn

Chúng ta cũng biết rằng truyền dẫn trên các kênh vô tuyến có nhiều lỗi hơn là truyền trong các kênh hữu tuyến. Các hiện tượng vật lý như phản xạ, khúc xạ và tán xạ cùng với sự di chuyển của các nút, sự dịch chuyển trong môi trường tạo nên **fading nhanh** và **nhiều liên**

ký hiệu. Suy hao đường truyền, sự suy giảm và sự có mặt của các chương ngại vật tạo nên **fading chậm**. Thêm vào đó là tạp âm và nhiễu từ các hệ thống hoặc nút khác làm việc ở cùng dải tần hoặc ở các dải tần lân cận. Điều này đã được đề cập đến trong chương 4.



Hình 6.1 Format của một khung lớp vật lý IEEE 802.11/802.11b

Méo dạng sóng gây ra các lỗi bit và suy hao tín hiệu. Hình 6.1 là format của **khối dữ liệu giao thức lớp vật lý** (PPDU - Physical-layer Protocol Data Unit) trong chuẩn WLAN của IEEE 802.11 với lớp vật lý DSSS. PPDU được chia thành phần mở đầu, PHY header và phần dữ liệu **khối dữ liệu giao thức lớp MAC** (MPDU - the MAC-layer Protocol Data Unit). Phần tiếp theo là gói tin MAC. Phần mở đầu là mẫu bit không đổi, được dùng cho các mục đích cân bằng và cho phép bộ thu có được sự đồng bộ bit và khung. PHY header mô tả độ dài và sơ đồ điều chế được dùng trong phần dữ liệu, được bảo vệ bởi trường kiểm tra tổng (kiểm tra tổng) của nó. Kết thúc phần PHY header và bắt đầu MPDU được biểu thị bởi một SFD cố định. **Mất tín hiệu** xảy ra nếu (i) bộ thu không đồng bộ bit/ khung được, (ii) SFD lỗi hoặc (iii) các lỗi bit còn lại trong PHY header làm cho kiểm tra tổng chúng không đúng. Kết quả của việc mất tín hiệu là các giai đoạn sau đó của bộ thu như bộ giải mã FEC hay giao thức MAC không có dữ liệu. Khi quá trình đồng bộ và PHY header kết thúc hoàn toàn, các bit tạo nên MPDU có thể được xử lý bởi FEC hoặc MAC. Nếu một số bit không giống như đã phát thì ta nói đã có **lỗi bit**. Các giao thức ARQ cung cấp các kiểm tra tổng để phát hiện các lỗi bit và việc mất tín hiệu.

Trong sơ đồ khung nói riêng này, việc áp dụng FEC có thể hiệu chỉnh các lỗi bit trong phần dữ liệu/ MPDU nhưng PHY header thì không và do đó, việc mất tín hiệu vẫn có thể xảy ra. Các phép đo với một bộ thu phát vô tuyến phù hợp với IEEE 802.11 được thực hiện trong môi trường công nghiệp không có đường nhìn thẳng (NLOS - Non Line Of Sight) cho thấy xảy ra cả hai loại lỗi với một tỉ lệ lớn. Cấu trúc khung như hình 6.1 là dạng chung và việc mất tín hiệu do đó cũng có thể xảy ra ở một số hệ thống khác.

Các thống kê về lỗi bit và mất tín hiệu phụ thuộc vào nhiều yếu tố như tần số, sơ đồ điều chế, khoảng cách, môi trường truyền sóng (số đường truyền, loại vật liệu) và sự có mặt của nhiễu. Một số nghiên cứu về các thống kê này chỉ ra các tính chất sau:

- + Cả lỗi bit và mất tín hiệu là chòm, tức chúng xảy ra theo nhóm với các thời kỳ không lỗi giữa các nhóm. Theo kinh nghiệm, các phân bố nhóm và độ dài thời kỳ không lỗi thường có hệ số thay đổi lớn.
- + Lỗi của các bộ phát và bộ thu cố định thay đổi theo thời gian và tỉ lệ lỗi bit tức thời đôi khi rất cao ($10^{-4} \dots 10^{-2}$). Tỉ lệ mất tín hiệu cũng xảy ra tương tự như vậy, có thể vượt quá 50%.

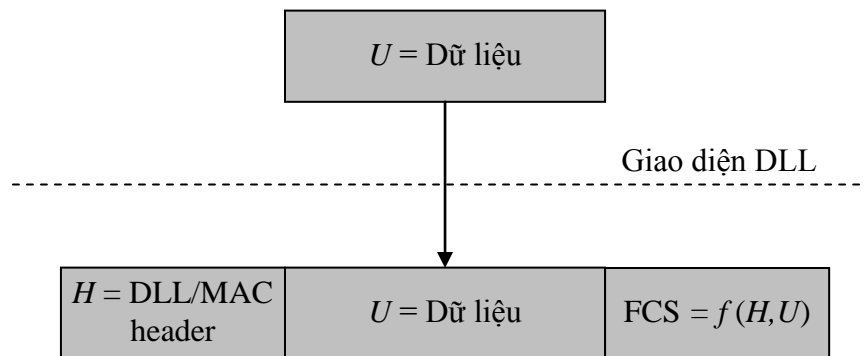
6.2.2 Các kỹ thuật ARQ

Ý tưởng cơ bản của các giao thức ARQ có thể được mô tả như sau. Lớp liên kết của

nút phát chấp nhận gói tin tạo thành gói tin lớp liên kết với header và kiểm tra tổng, và truyền gói này tới bộ thu. Bộ thu kiểm tra tính toàn vẹn của gói tin với sự trợ giúp của kiểm tra tổng và tạo ra **tín hiệu phản hồi** (feedback) đến bộ phát về sự thành công của quá trình truyền dẫn. Nếu bộ thu không có tín hiệu phản hồi thì bộ phát sẽ truyền lại.

Thành phần chính của các giao thức ARQ

- **Định dạng gói tin:** DLL ở bộ phát chấp nhận dữ liệu U từ các lớp cao hơn, thường là các đối tượng bị hạn chế về mặt kích thước (hình 6.2). Hạn chế này có thể bị áp đặt, ví dụ như bằng các tham số kỹ thuật hoặc các hạn chế của lớp vật lý như sóng mang hay sơ đồ đồng bộ bit có ít thuật toán tìm kiếm (xem phần 4.2.6). DLL tạo **header** H tới gói tin chứa thông tin về sự giám sát và địa chỉ. Thông tin về địa chỉ là cần thiết vì phương tiện vô tuyến là phương tiện quảng bá và về mặt lý thuyết, gói tin có thể được nhận từ các nút lân cận ở cùng dải vô tuyến. Địa chỉ sẽ giúp phân biệt bộ thu phát mong muốn với các bộ thu phát khác. Thông tin giám sát có thể bao gồm số chuỗi hoặc cờ phụ thuộc vào từng loại giao thức ARQ.



Hình 6.2 Định dạng gói tin ở DLL

- **Kiểm tra tổng:** Kiểm tra tổng (thường được gọi là **chuỗi kiểm tra khung**) được nối vào gói tin sau quá trình định dạng chúng. Thông thường, nó là hàm của cả dữ liệu U và header H . Lớp phổ biến của nó là các giá trị CRC có độ rộng 8, 16 hoặc 32 bit. Việc tính toán các giá trị của CRC tương đối dễ vì nó có thể được thực hiện bằng các thanh ghi dịch phản hồi tuyến tính. Bộ thu lặp lại việc tính toán kiểm tra tổng cho header nhận được H' và dữ liệu nhận được U' . Nếu kiểm tra tổng mới tính giống với kiểm tra tổng kèm theo gói tin thì bộ thu chấp nhận gói tin là đúng. Thông thường, kích thước dữ liệu U và header H lớn hơn độ rộng của kiểm tra tổng và việc tính toán kiểm tra tổng không phải là một – thành – một. Do đó, một lỗi nào đó biến đổi dữ liệu U thành dữ liệu U' , chúng có cùng một kiểm tra tổng thì gói tin cũng sẽ được chấp nhận mặc dù có lỗi (không phát hiện ra). Vì vậy, luôn có **tỉ lệ lỗi dư thừa** (residual error rate) và độ dư thừa lớn khi kiểm tra tổng nhỏ, nó còn phụ thuộc vào các đặc tính thống kê của cả các bit và lỗi. Kiểm tra tổng nhỏ là điều mong muốn trong các mạng cảm biến để giữ cho số bit truyền thấp.

- **Tạo tín hiệu phản hồi** (feedback generation): Bước **tạo tín hiệu phản hồi** cung cấp cho bộ phát thông tin về kết quả của việc truyền dẫn. Hai cơ cấu thường dùng để có được phản hồi là các bộ định thời (ở bộ phát) và các gói phản hồi. Phản hồi có thể xác nhận (positive) hoặc phủ nhận (negative). **Phản hồi xác nhận** tức là bộ thu xác nhận đã nhận được

gói tin. Phản hồi có thể gửi cho mỗi gói hoặc một gói phản hồi mang thông tin của một số gói dữ liệu. Phương pháp thứ hai rõ ràng là hiệu quả hơn về mặt tiết kiệm năng lượng. **Phản hồi phủ nhận** tức là bộ thu phát hiện việc nhận tin bị lỗi. Tuy nhiên, trong môi trường có các nút có thể nghe lẫn nhau thì phản hồi phủ nhận khả thi nếu (i) header MAC/DLL có kiểm tra tổng riêng, việc nhận tin thực sự đúng và do đó, bộ thu biết nút nào đã truyền tin, hoặc (ii) bộ thu đã phát hiện việc truyền tin bị lỗi. Sau đây là cách phát hiện lỗi nhận tin của các gói tin tiếp theo. Do cả gói dữ liệu và phản hồi đều có thể bị suy hao nên bộ phát cần phải sử dụng các **bộ định thời** (timer). Bộ phát đặt bộ định thời khi bit cuối cùng của gói dữ liệu được gửi đi và tự báo thức khi thời gian cho việc nhận tin phản hồi kết thúc. Timeout (giá trị quá thời gian) phải đủ lớn để bao gồm cho cả thời gian xử lý ở bộ thu, thời gian cho gói phản hồi và trễ truyền dẫn. Trong các WSN, các nút nằm rất gần nhau nên trễ truyền dẫn không đáng kể. Đây cũng là một lựa chọn cho bộ thu, đó là thông tin phản hồi sẽ được đính kèm với gói dữ liệu khi truyền ngược lại. Tuy nhiên, để tránh trễ lớn cho thông tin phản hồi lại làm tăng lưu lượng một cách đáng kể. Trong trường hợp lưu lượng thấp như vẫn thường thấy trong các ứng dụng của WSN, cơ chế này thường gây lỗi và phải gửi thêm các gói phản hồi.

- **Truyền lại** (Retransmissions): Dựa trên gói tin phản hồi phủ nhận, bộ phát sẽ thực hiện việc truyền lại. Hệ quả đầu tiên của việc làm này là bộ phát phải làm bộ đệm cho gói tin. Thứ hai là phải quyết định *khi nào* phải truyền lại và truyền lại *như thế nào*?

Các giao thức ARQ chuẩn

Các giao thức ARQ chuẩn khác nhau ở cách truyền lại và các yêu cầu về bộ đệm.

- **Bit xen kẽ**: Các bộ phát đệm một gói tin, gửi nó và đặt bộ định thời. Bộ thu nhận gói tin và gửi phản hồi xác nhận, nếu không nhận được gì thì bộ thu yên lặng hoặc gửi phản hồi phủ nhận. Nếu bộ phát nhận được phản hồi xác nhận, bộ đệm sẽ được giải phóng và gói tin tiếp theo có thể được truyền. Nếu không, bộ phát truyền lại gói tin. Bộ phát đánh dấu mỗi gói tin mới bằng một chuỗi (sequence) các số 0 và 1 xen kẽ nhau. Các bản tin truyền lại là bản sao của gói tin gốc và do đó, nó có cùng chuỗi số. Các chuỗi số giúp bộ thu phát hiện các bản sao, nó là đúng nếu có phản hồi xác nhận, không có dữ liệu thì gói tin đã bị mất. Bit xen kẽ sẽ tránh được hiện tượng bị mất tin, có bản sao, sắp xếp dữ liệu đã cho theo chuỗi, thời gian khứ hồi (RTT - round-trip time) được giữ trong giới hạn và timeout của bộ phát lớn hơn RTT.

- **Goback N (làm lại)**: Bit xen kẽ sẽ không hiệu quả trong trường hợp các tuyến có thể có nhiều gói tin truyền trong thời gian khứ hồi. Goback N là giao thức dùng cho bộ phát có nhiều vấn đề chưa được giải quyết, đó là các khung không có phản hồi. Bộ phát giữ bộ đệm cho N gói tin, nó còn được gọi là **cửa sổ** (window). Mỗi gói tin trong cửa sổ có một bộ định thời riêng và được khởi động dựa trên việc truyền dẫn tin. Bộ thu chỉ chấp nhận các khung theo chuỗi và bỏ khung khi nhận đúng gói tin nhưng nó không có dãy số mong muốn (thường là do mất khung nào đó ở phía trước). Do đó, bộ thu chỉ cần không gian bộ đệm cho một khung. Chiến lược chung cho các gói phản hồi là để bộ thu luôn phản hồi gói tin cuối cùng nhận được theo chuỗi. Nếu ở bộ phát, bộ định thời cho khung cuối cùng hết hạn do gói phản hồi tương ứng không nhận được thì khung này và tất cả các khung khác trong cửa sổ phải truyền lại.

- **Loại bỏ có lựa chọn/Lặp lại có lựa chọn**: Lặp lại có lựa chọn tương tự như Goback N. Tuy nhiên, điểm khác nhau ở đây là trong lặp lại có lựa chọn, bộ thu cũng có N bộ đệm và

dùng chúng để các khung đệm đến không theo chuỗi. Để sắp xếp dữ liệu theo chuỗi, bộ thu phải giữ các gói không theo chuỗi trong bộ đệm cho đến khi các gói tin lỗi đến. Bộ thu có thể sử dụng cả gói tin phản hồi xác nhận hoặc phủ nhận. Bộ phát chỉ truyền lại khi các gói tin này trong trường hợp không nhận được phản hồi trong thời kỳ timeout.

Đợi – và – gửi, lặp lại có lựa chọn có đặc điểm quan trọng là chỉ các gói lỗi được truyền lại trong khi Goback N truyền lại cả gói tin nhận đúng, làm như vậy sẽ bị lãng phí năng lượng. Trên thực tế, số lần gói tin truyền lại bị giới hạn để tránh việc sử dụng quá nhiều năng lượng trong nhiều trường hợp không mong muốn. Trong trường hợp này, dịch vụ không mất tin không được đảm bảo và giao thức này còn được gọi là **bán tin cậy** (semireliable).

Sử dụng các gói phản hồi như thế nào?

Các gói phản hồi được thảo luận kỹ trong chương này là các phản hồi lớp liên kết và ý nghĩa của chúng là bộ thu (i) nhận được đúng gói tin, (ii) có không gian bộ đệm đủ để xử lý nó và (iii) thực sự chấp nhận gói tin đó. Nói cách khác, một số giao thức MAC sử dụng các phản hồi của lớp MAC như giao thức thiết bị trung gian (phần 5.2.3) hay S-MAC (phần 5.2.2).

Rõ ràng là ta có thể tiết kiệm năng lượng bằng cách giảm số gói tin phản hồi, do đó ta nên dùng một gói phản hồi để thông báo cho nhiều gói dữ liệu. Sơ đồ này yêu cầu bộ phát, và cả bộ thu trong trường hợp lặp lại có lựa chọn, tạo ra một số bộ đệm và đây chính là vấn đề khi các nút cảm biến chỉ có tài nguyên nhớ hạn chế.

Hai sơ đồ này được đề cập đến trong các giao thức Goback N và lặp lại có lựa chọn. Chúng làm việc tương tự như giao thức AIRMAIL lớp liên kết. Ta sẽ tập trung vào trường hợp lặp lại có lựa chọn vì nó tránh được trường hợp phải truyền lại các khung đã nhận đúng. Trong sơ đồ đầu tiên, được gọi là **phản hồi cửa sổ với lặp lại có lựa chọn**, bộ thu gửi gói phản hồi sau khi nhận được một số cố định W gói tin (còn gọi là *kích thước cửa sổ*), sau timeout hoặc sau khi nhận được bản sao của gói tin. Phản hồi chứa bitmap cho biết quan điểm của bộ thu dựa vào trạng thái gói tin từ khi gửi đi gói phản hồi cuối cùng.

Trong **sơ đồ phản hồi tức thời với lặp lại có lựa chọn**, bộ thu gửi phản hồi (phủ nhận) khi nhận được một gói tin không theo trật tự. Hành động của bộ phát rất đơn giản. Nó có thể gửi ngay các gói tin mới phụ thuộc vào cửa sổ hiện thời cỡ W . Nếu kết thúc cửa sổ mà vẫn chưa nhận được phản hồi nào, bộ phát sẽ lặp lại khung gần nhất để kích việc tạo phản hồi ở bộ thu. Khi bộ phát nhận được phản hồi, nó có thể giải thoát các bộ đệm của các gói tin nhận đúng và các gói tin lỗi sẽ được truyền lại.

Sau đây là các thông số quan trọng đặc trưng cho các sơ đồ này. Với các hệ thống chia khe có các gói tin có kích thước cố định, hiệu quả năng lượng đo được là tỉ số trung bình của các khe mang khung phản hồi. Cách này chỉ đặc trưng cho việc tiết kiệm các khung phản hồi. Hiệu suất được ước lượng theo tỉ lệ chấp nhận trung bình của các gói tin mới trong cửa sổ bộ phát trên một khe thời gian đã cho. Trễ là khoảng thời gian từ khi gói tin đến bộ phát cho đến khi nhận được phản hồi và bộ đệm được giải phóng.

- Cả fading nhanh và chậm có tỉ lệ lỗi tin trung bình là 30%, tăng kích thước cửa sổ sẽ làm tăng hiệu suất và giảm mức tiêu thụ năng lượng trung bình cho cả hai sơ đồ, sơ đồ phản hồi cửa sổ mang lại hiệu quả năng lượng tốt hơn nếu ở cùng mức công suất.

- Khi xem xét mối quan hệ giữa mức tiêu thụ năng lượng và trễ, tăng kích thước cửa sổ, đối với sơ đồ phản hồi tức thời sẽ làm giảm năng lượng tiêu thụ và các trễ thay đổi không nhiều. Với sơ đồ phản hồi cửa sổ, tăng kích thước cửa sổ cũng sẽ làm giảm năng lượng tiêu thụ nhưng các trễ lại tăng rất nhiều.

Tóm lại, khi mức tiêu thụ năng lượng là cố định thì việc lựa chọn sơ đồ phụ thuộc vào tham số thứ hai: nếu mục tiêu là trễ thì sơ đồ phản hồi tức thời thích hợp hơn nhưng nếu mục tiêu là công suất thì sơ đồ phản hồi cửa sổ lại tốt hơn.

Khi nào thì truyền lại? Trong các giao thức ARQ, không chỉ có truyền đi mà truyền lại cũng phải quyết định khi nào làm việc này. Thời điểm có tạo nên sự khác nhau không, điều này phụ thuộc và các đặc tính lỗi sau của kênh truyền:

- Trong kênh BSC tĩnh, thời điểm không có ý nghĩa. Không thể tăng cơ hội thành công của quá trình truyền dẫn bằng cách đợi.

- Trong các kênh fading thì thời điểm truyền lại có sự khác nhau. Nếu gói tin gặp fading sâu và nếu độ dài của nó ngắn so với khoảng fading trung bình thì quyết định truyền lại ngay lập tức sẽ giống như việc gặp fading sâu và do đó, sẽ phí năng lượng. Nếu giao thức là bán tin cậy thì số lần truyền lại cực đại có thể được dùng hết trước khi kênh quay trở lại trạng thái tốt.

Vì vậy, các kênh fading có thể **hoãn việc truyền lại**. Hoãn việc truyền lại và các gói tin sẽ đến các nút khác, trong lúc đó có thể tăng công suất của một cách đáng kể do đó không có băng thông (và năng lượng) được sử dụng một cách vô ích cho việc truyền lại ngay lập tức. Vấn đề ở đây là phải đợi trong bao lâu?

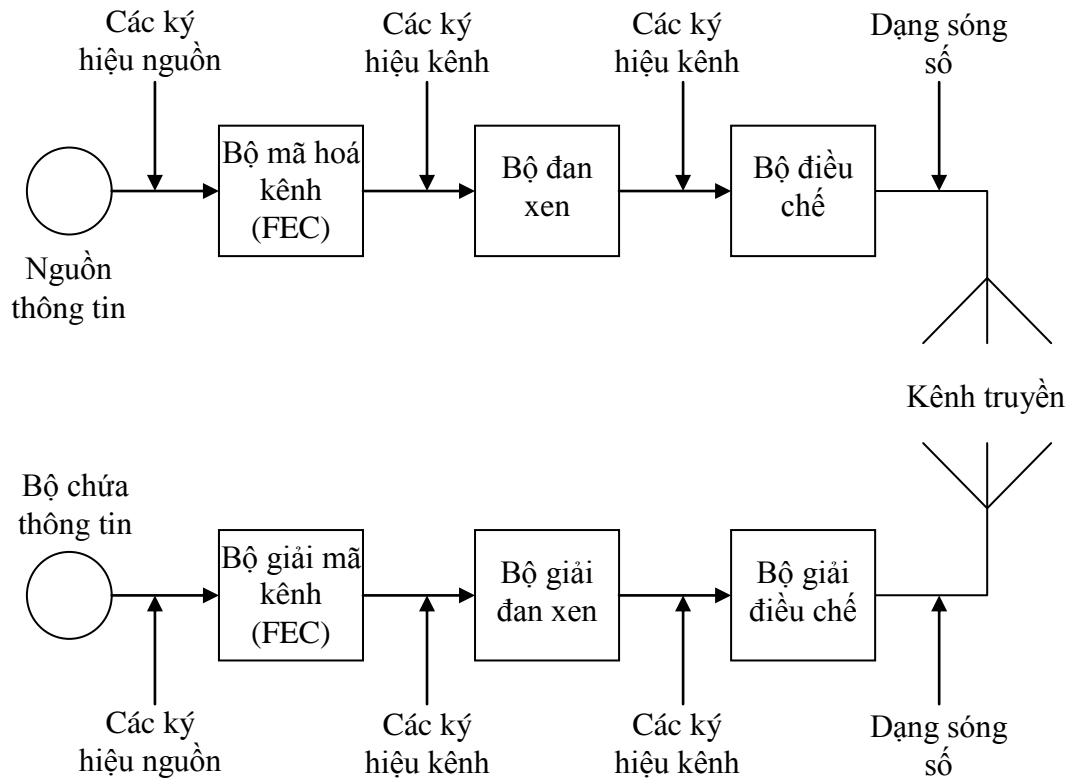
Giao thức tìm kiếm (probing protocol) được giới thiệu bởi Zorzi và Rao phân biệt hai “chế độ kênh” khác nhau là **chế độ thường** và **chế độ tìm kiếm**. Trong chế độ thường, nút truyền gửi tin theo giao thức ARQ, ví dụ như Goback N hoặc lặp lại có lựa chọn. Nếu bộ phát nhận được phản hồi phủ nhận (ví dụ như thiếu phản hồi và hỏng bộ định thời) thì nó sẽ chuyển sang chế độ tìm kiếm. Trong chế độ này, bộ phát gửi định kỳ **các gói tin tìm kiếm nhỏ** và chúng được phản hồi bởi các bộ thu. Dựa trên việc nhận thông tin phản hồi tìm kiếm, bộ phát cho rằng cả kênh tiến (forward - từ bộ phát đến bộ thu) và kênh phản hồi (feedback - từ bộ thu đến bộ phát) đều tốt và tiếp tục ở chế độ bình thường.

Đã có hai phiên bản của giao thức tìm kiếm được giới thiệu tương ứng với Goback N và lặp lại có lựa chọn. Trong trường hợp đầu tiên, khi chuyển sang chế độ thường, bộ phát sẽ truyền các gói tin bị lỗi và tất cả các gói tin sau đó lại đến bộ thu. Trong giao thức tìm kiếm lựa chọn, bộ thu có các bộ đệm cho các gói tin sau khi nó bị lỗi hoặc bị sửa lại. Trong phản hồi gói tìm kiếm, bộ thu cho biết các khung đệm và bộ phát chỉ truyền lại các gói bị mất. Phương pháp này sẽ hữu dụng trong trường hợp tuyến dài, còn một vài khung chưa được dùng đến và lỗi chùng đủ ngắn để chỉ cần sử dụng một khung hoặc trong trường hợp kênh tiến tốt nhưng kênh phản hồi lại thực sự kém. Các giao thức tìm kiếm làm giảm hiệu suất do kênh sau khi chuyển sang trạng thái tốt sẽ cần thời gian để bộ phát nhận biết chúng và chuyển sang chế độ bình thường. Thời gian này liên quan đến chu kỳ của các gói tìm kiếm. Hiệu quả năng lượng của các giao thức tìm kiếm và giao thức Goback N kinh điển được so sánh với số thử nghiệm trung bình cần để truyền một gói dữ liệu. Trong các điều kiện về kênh truyền xấu (các lỗi chùng dài) và các kênh có thời gian khử hồi ngắn thì giao thức tìm

kiểm cố giữ con số này gần với một. Số gói tin tìm kiếm trung bình được yêu cầu tăng khi kênh truyền có điều kiện xấu/ các lỗi chùm dài hơn.

6.2.3 Các kỹ thuật FEC

Trong tất cả các cơ chế FEC, bộ phát chấp nhận chùm dữ liệu, khối bit dữ liệu hoặc nguồn bit, độ dư thừa phù hợp và truyền tất cả chúng về phía bộ thu. Vị trí của FEC trong hệ thống truyền dẫn được chỉ ra trên hình 6.3.



Hình 6.3 Vị trí của FEC trong chuỗi gửi/nhận; thực hiện mã hoá và giải mã FEC

Phụ thuộc và số lượng và cấu trúc của độ dư thừa, bộ thu có thể hiệu chỉnh một số lỗi bit. FEC có thể sử dụng như một **kỹ thuật vòng hở** (open loop technique), nghĩa là không có thông tin phản hồi từ bộ thu. Do đó, bộ phát sử dụng một phương pháp mã hoá trong toàn bộ thời gian. Đây là một đặc điểm đáng chú ý khi xét đến khía cạnh năng lượng. Điều này yêu cầu nút truyền phải chuyển bộ thu phát của nó sang trạng thái nhận và đợi phản hồi, tức là mạng phải tốn năng lượng để nhận tin (đối với gói phản hồi) và điều chỉnh bộ thu. Hơn nữa, thông thường các gói dữ liệu trong WSN có kích thước nhỏ nên các gói phản hồi chiếm một phần năng lượng đáng kể để truyền chúng.

Hai sơ đồ mã hoá được sử dụng rộng rãi là **mã khối** và **mã chập**. Thời gian gần đây người ta hay sử dụng các **mã turbo** để thoả mãn định lý thứ hai của Shannon về khả năng của kênh truyền dẫn nhưng thực hiện nó rất phức tạp, chính vì vậy đây không phải là một lựa chọn đúng đắn cho WSN.

FEC mã khối

Một bộ mã hoá FEC khối lấy một **khối** (block) hoặc một **từ** (word) k **ký hiệu nguồn** p -ary (source symbol) và tạo ra một khối có chứa n **ký hiệu kênh** q -ary. Thông thường ta có

$p = q = 2$, $n \geq k$ và các ký hiệu tương ứng với các bit. Các khối nguồn khác nhau được mã hoá *độc lập* với nhau. Ánh xạ của 2^k word khác nhau vào 2^n bit kênh khác nhau là ánh xạ (injective) và phạm vi của ánh xạ này là tập **các ký hiệu kênh hợp lệ** (valid channel symbols). Ánh xạ này còn được gọi là **mã hoá**, và tỉ số k/n được gọi là **tỉ lệ mã hoá** (tỉ lệ mã hoá nhỏ tương đương với độ dư thừa lớn và tỉ lệ dữ liệu hữu dụng nhỏ). Số t bit có thể hiệu chỉnh một cách tin cậy trong khối kênh có độ dài n bit phụ thuộc vào sơ đồ mã hoá. Tuy nhiên, giới hạn trên bị áp đặt theo **giới hạn Hamming** (Hamming bound), tức là các trạng thái mà mã hoá khối với k bit dữ liệu ánh xạ tới n bit kênh có thể hiệu chỉnh tới t lỗi bit chỉ nếu mối quan hệ sau được thoả mãn:

$$2^{n-k} \geq \sum_{i=0}^t \binom{n}{i} \quad (0.17)$$

Trên thực tế, ba giá trị (n, k, t) thoả mãn mối quan hệ này không có nghĩa là mã với các tính chất này thực sự tồn tại.

Một tham số quan trọng của mã hoá khối là **khoảng cách Hamming** (Hamming distance): khoảng cách Hamming của hai mã ký hợp lệ w_1 và w_2 được định nghĩa là số bit khác nhau và khoảng cách Hamming tối thiểu d_{\min} là số bit khác nhau tối thiểu giữa các cặp mã kênh hợp lệ để máy thu có thể phát hiện được lỗi.

Trong lý thuyết thông tin, khoảng cách Hamming giữa hai dãy ký tự (strings) có chiều dài bằng nhau là số các ký hiệu ở vị trí tương đương có giá trị khác nhau. Nói một cách khác, khoảng cách Hamming đo số lượng thay thế cần phải có để đổi giá trị của một dãy ký tự sang một dãy ký tự khác, hay số lượng lỗi xảy ra biến đổi một dãy ký tự sang một dãy ký tự khác.

Ví dụ 6.1:

Khoảng cách Hamming giữa 1011101 và 1001001 là 2.

Khoảng cách Hamming giữa 2143896 và 2233796 là 3.

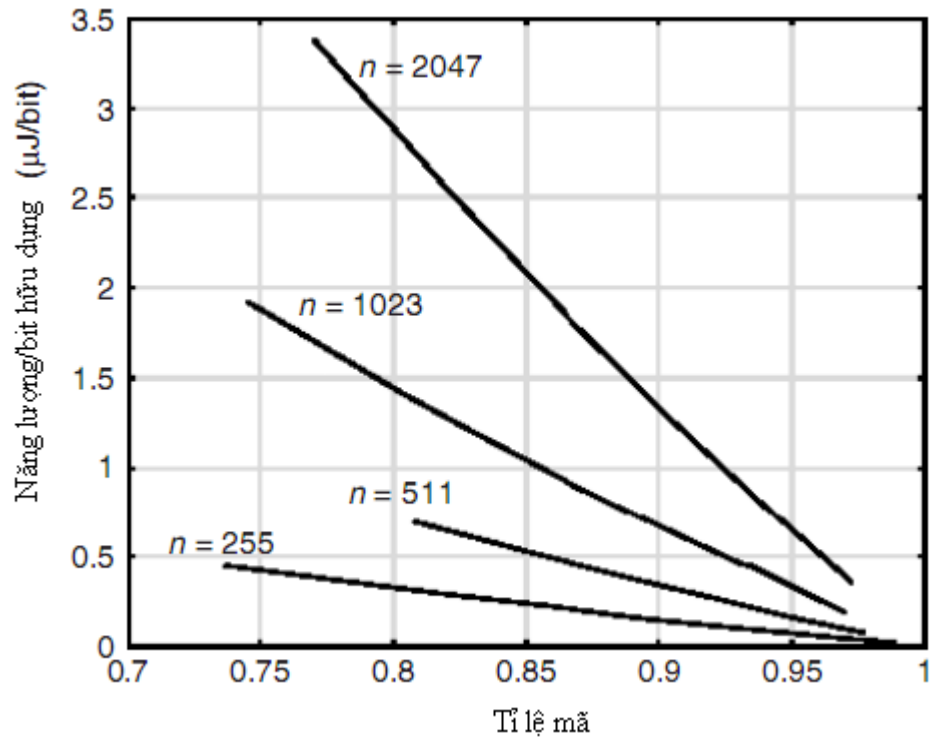
Khoảng cách Hamming giữa "toned" và "roses" là 3.

Một mã bất kỳ với khoảng cách Hamming d_{\min} có thể phát hiện được $(d_{\min} - 1)$ lỗi bit và hiệu chỉnh $(d_{\min} - 1)/2$ lỗi bit. Trong các ứng dụng thực tế, tập các mã kênh hợp lệ có thể được tạo ra sao cho quá trình mã hoá và giải mã dễ thực hiện. Ví dụ, đối với các mã khối tuyến tính, việc xây dựng không gian vector dựa vào 2^n mã ký và tập con 2^k mã ký hợp lệ là không gian con. Giải mã có thể được xem như là phép chiếu trực giao của mã ký nhận được trong không gian con của các mã ký hợp lệ. Ứng dụng phổ biến của mã khối là mã Reed – Solomon (RS) và mã BCH (Bose – Chaudhuri – Hocquenghem).

Nếu mã khối FEC được áp dụng như là một kỹ thuật vòng hở, mào đầu (overhead) hằng số sẽ được kèm theo mỗi gói tin. Trên nút truyền, mào đầu này gồm một số bit phụ và làm các tính toán cần thiết cho việc mã hoá. Với các mã nhị phân BCH, quá trình mã hoá sử dụng thanh ghi dịch phản hồi tuyến tính và có thể có chi phí năng lượng không đáng kể. Với giải mã, một số thuật toán có hiệu quả đã được triển khai, ví dụ như thuật toán Berlekamp-Massey. Chi phí năng lượng của thuật toán này phụ thuộc vào độ dài khối n và số lượng/ tỉ lệ mã của các bit có thể hiệu chỉnh. Đối với khối có độ dài n , các thuật toán này cho thấy giá thành năng lượng tăng tuyến tính. Năng lượng giải mã phụ thuộc vào độ dài khối n và số lượng các bit có thể hiệu chỉnh t :

$$E_{\text{dec}} = (2nt + 2t^2) \cdot (E_{\text{add}} + E_{\text{mult}}) \quad (0.18)$$

Trong đó E_{add} và E_{mult} là năng lượng cần để thực hiện phép cộng và nhân trong trường Galois $GF(2^m)$ với $m = \log_2 n + 1$. Với mã Reed – Solomon, mối quan hệ cũng tương tự như vậy do chúng là lớp con của mã BCH nhị phân. Chi phí năng lượng còn phụ thuộc vào tốc độ mã, chúng tỉ lệ nghịch với nhau (hình 6.4).



Hình 6.4 Chi phí năng lượng của mã Reed-Solomon với kích thước khối và tỉ lệ mã khác nhau

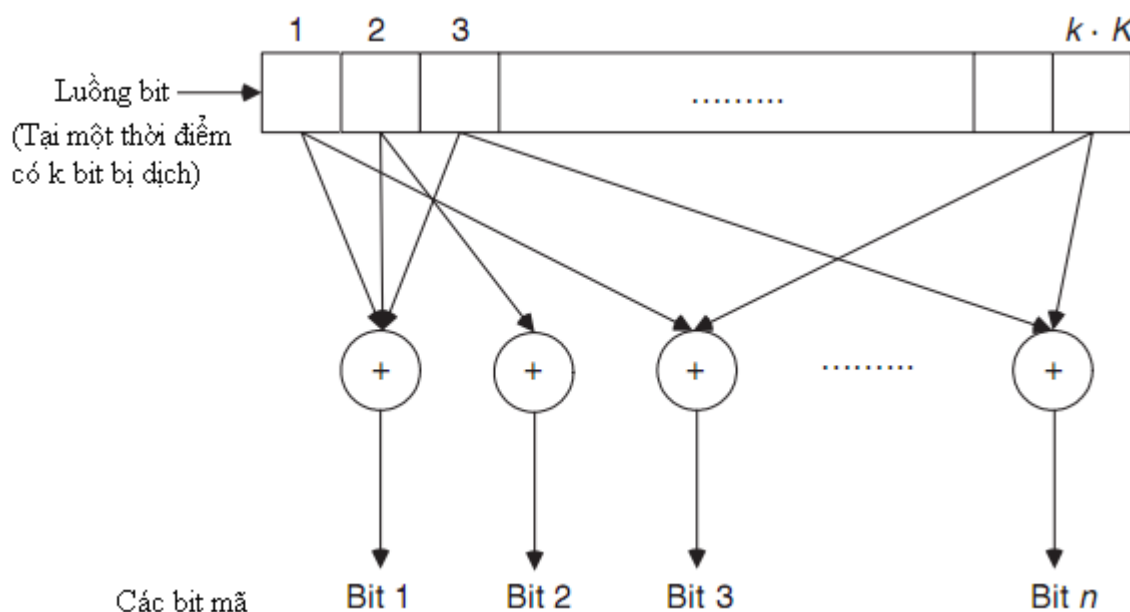
Nếu các điều kiện lỗi trên kênh truyền thay đổi theo thời gian thì **tỉ lệ lỗi sót** (residual error rate – tỉ lệ của các lỗi không thể hiệu chỉnh) của các kỹ thuật FEC cũng thay đổi. Nếu tại mọi thời điểm, tỉ lệ lỗi bit rất nhỏ hoặc ít nhất cũng trong thời kỳ kênh truyền tốt trong kênh fading, mã đầu FEC lớn sẽ không hợp lý. Ngược lại, trong trường hợp tỉ lệ lỗi cực kỳ cao, ví dụ như thời kỳ fading sâu trong mô hình kênh fading Rayleigh (phần 4.2.4) hoặc nếu cường độ tín hiệu nhận của nút bộ phát ở nút bộ thu gần ngưỡng nhận, tỉ lệ mã rất thấp sẽ là cần thiết để đảm bảo truyền không có lỗi. Như đã nói ở trên, các mã giải mã với tỉ lệ mã rất thấp sẽ tốn nhiều năng lượng. Vì vậy, sẽ thích hợp hơn để đợi kết thúc fading và tiếp tục với tỉ lệ mã vừa phải trong thời kỳ kênh tốt.

Mã chập

Trong mã chập, cũng k bit dữ liệu ánh xạ đến n ký hiệu kênh. Tuy nhiên, mã hoá của hai khối k bit thành công phụ thuộc vào nhau. Hình 6.5 giải thích hoạt động của bộ mã chập.

Thủ tục mã hoá được thực hiện theo **từng bước**. Trong mỗi bước, k bit dữ liệu được dịch chuyển trong thanh ghi dịch. Thanh ghi dịch có độ dài $K.k$ bit, trong đó K được gọi là **độ dài hạn chế** của mã. K bit dữ liệu được dịch chuyển trong thanh ghi dịch theo một hướng, khối k bit ngoài cùng bị chuyển khỏi thanh ghi dịch. Hơn nữa, có n bộ cộng modulo-2, mỗi bộ cộng sẽ tính tổng một tập con cụ thể của $k.K$ thanh ghi. Đầu ra của n bộ cộng

được truyền một lần trong mỗi bước. Sơ đồ mã hoá này có tính chất là k bit có mặt trong thanh ghi dịch sau K bước và do đó, mã hoá k bit không chỉ phụ thuộc vào chúng mà còn phụ thuộc vào $(K-1) \cdot k$ bit trước đó nữa. Tỉ số n/k được gọi là tỉ lệ textitcode. Độ dài hạn chế K giám sát số lượng dư thừa trong các bit mã. Nếu tăng K sẽ tăng độ khuếch đại mã hoá, do đó giảm được năng lượng truyền cần để đạt được tỉ số BER đã cho hoặc giảm BER với công suất truyền cố định.



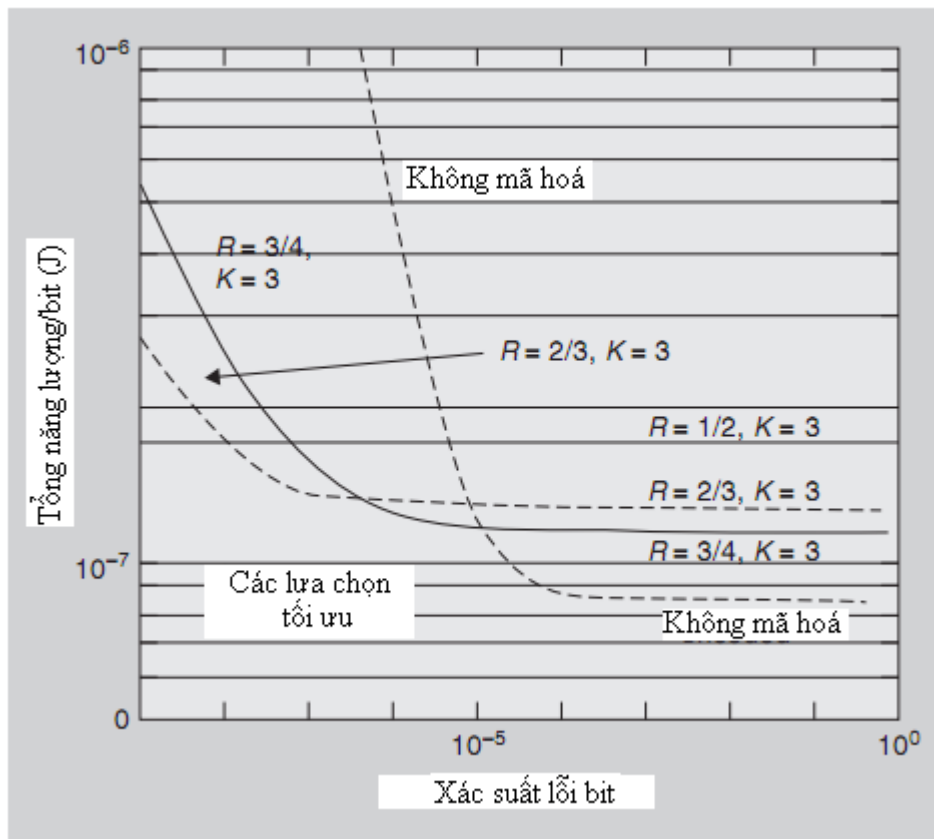
Hình 6.5 Hoạt động của mã chập

Tương tự như mã khối, quá trình giải mã của mã chập tốn ít năng lượng. Để giải mã mã chập, người ta thường sử dụng thuật toán Viterbi. Với thuật toán này, các yêu cầu về bộ nhớ của bộ thu phụ thuộc theo hàm mũ của độ dài hạn chế K . Mô tả phần cứng và phần mềm thuật toán Viterbi cho thấy, mức tiêu thụ năng lượng cũng tăng theo hàm mũ của K .

Sau đây ta sẽ làm phép so sánh hiệu quả năng lượng của mã khối và mã chập. Nếu chọn tiêu chí hiệu quả năng lượng và chi phí năng lượng trên một gói tin là thước đo khả năng thành công của nó, bao gồm chi phí cho bộ thu phát và quá trình giải mã. Với BSC có tỉ lệ lỗi bit cố định $p = 10^{-3}$ và chọn độ dài của gói tin là tối ưu, các sơ đồ mã hoá khối có được hiệu quả năng lượng tốt hơn các sơ đồ giải mã chập. Mã chập tỉ lệ cao cho thấy khả năng thành công của gói tin tương đối nhỏ, trong khi các mã tỉ lệ thấp lại có độ tin cậy cao nhưng chi phí năng lượng lại rất đáng kể. Tuy nhiên, các mã có tỉ lệ trung bình ($3/4$ và $5/6$) và độ dài hạn chế cực đại $K = 9$ lại làm tốt hơn mã BCH khi nói đến hiệu quả năng lượng.

Min so sánh một số mã chập về hiệu quả năng lượng với mục đích là đạt được xác suất lỗi dư thừa có thể chấp nhận được nào đó trên một tuyến vô tuyến cố định với suy hao đường truyền là 70 dB và đặc tính tiêu thụ năng lượng các nút MITs μ AMPS-1 (phần 2.2.4). Các đặc tính này bao gồm năng lượng tiêu thụ bởi bộ thu phát cho việc mã hoá và giải mã cũng như các chi phí xử lý khác. Tất cả các mã được chọn có độ dài hạn chế $K = 3$ nhưng có tỉ lệ mã khác nhau: $1/2$, $2/3$ và $3/4$. Với xác suất lỗi dư thừa mong muốn rất nhỏ ($\approx 10^{-9}$ và nhỏ hơn), mã có tỉ lệ $1/2$ có hiệu quả năng lượng tốt nhất trong khi trong dải $\approx 10^{-9} \dots 10^{-7}$ thì mã tỉ lệ $2/3$ lại là mã tốt nhất. Cuối cùng, nếu các yêu cầu tỉ lệ lỗi bit dư thừa trung bình thì điều chế giải mã là phương pháp được lựa chọn. Hình 6.6 cho thấy bằng cách nối lỏng các yêu cầu về độ

tin cậy thì các mã sẽ có tỉ lệ mã cao hơn và do đó, mã đầu nhỏ hơn có thể sử dụng nên tiết kiệm năng lượng truyền dẫn.



Hình 6.6 Tiêu thụ năng lượng trong khi truyền 10 kB dữ liệu qua tuyến với suy hao đường truyền 70 dB theo các sơ đồ mã hoá khác nhau

Kỹ thuật đan xen

Nhiều sơ đồ mã hoá, đặc biệt là các sơ đồ mã chập, không có được đặc tính tối ưu khi chúng đối mặt với môi trường có các đặc tính lỗi chùm. Các thông thường để giải quyết vấn đề này (và làm giảm năng lượng trong mã hoá FEC) là sử dụng **kỹ thuật đan xen** (interleaving). Ở đây, một **bộ đan xen** (interleaver) trong nút phát chấp nhận gói dữ liệu có độ dài cố định được tạo bởi bộ giải mã FEC (như hình 6.3), nó thường bao gồm nhiều khối mã. Các bit trong khối này được hoán đổi vị trí cho nhau trước khi truyền đi. **Bộ giải đan xen** (deinterleaver) ở nút thu sẽ nghịch chuyển hoán vị trước khi gói tin được chuyển đến bộ giải mã FEC. Khi lỗi chùm gặp gói tin bị hoán đổi vị trí như vậy, kỹ thuật giải đan xen sẽ trải lỗi chùm tập trung ra toàn bộ chiều dài gói tin. Do đó, các lỗi chùm sẽ được trải ra nhiều khối mã hoá thay vì tập trung vào một hay một vài khối. Điều này làm tăng cơ hội để mỗi khối đều có thể được giải mã thành công.

Chế độ hoạt động thông thường của bộ đan xen là:

- Bộ đan xen đợi m từ mã w_i ($i = 1, \dots, m$) trong mỗi n bit ($w_i = w_{i,1}, w_{i,2}, \dots, w_{i,n}$)
- Chúng được xếp thành ma trận $m \times n$

Ví dụ với $n = 6$, $m = 4$, các từ mã thành công được xếp thành các hàng sau:

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$	$w_{1,5}$	$w_{1,6}$
$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$	$w_{2,5}$	$w_{2,6}$

$$\begin{array}{cccccc} w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} & w_{3,5} & w_{3,6} \\ w_{4,1} & w_{4,2} & w_{4,3} & w_{4,4} & w_{4,5} & w_{4,6} \end{array}$$

Các ký hiệu này được truyền đi theo thứ tự $w_{1,1} w_{2,1} w_{3,1} w_{4,1} w_{1,2} w_{2,2} \dots w_{3,6} w_{4,6}$.

Trong ví dụ này, tất cả các lỗi chùm có độ dài $\leq m = 4$ được phân bố bởi hoạt động giải đan xen một cách công bằng trên tất cả các từ mã w_i , do đó mỗi từ mã chỉ có một lỗi bit. Như vậy, kỹ thuật đan xen có hiệu quả nhất khi m ký hiệu kênh có thời gian tồn tại ít nhất là bằng thời gian fading trung bình. Chú ý là kỹ thuật đan xen không làm giảm tỉ lệ lỗi bit của bộ giải mã tiền FEC nhưng nó sắp xếp các lỗi cho bộ giải mã FEC "đẹp hơn". Kỹ thuật đan xen có nhược điểm là nút phát phải đợi truyền dẫn cho đến khi $m.n$ ký hiệu dữ liệu được lựa chọn, điều này sẽ tạo ra trễ tương đối lớn.

FEC đa bước nhảy

Phần trên mới chỉ giới thiệu FEC trong một bước. Zorzi và Rao xem xét đến trường hợp đa bước nhảy và đưa ra ba sơ đồ khác nhau cho việc tiết kiệm năng lượng đối với hạn chế đã cho dựa vào công suất nhận ở đích cuối cùng: (i) truyền dẫn FEC mã hoá trực tiếp từ nguồn đến đích, (ii) truyền dẫn đa bước nhảy với các nút trung gian thực hiện mã hoá và giải mã FEC, (iii) truyền dẫn đa bước nhảy không có nút trung gian thực hiện giải mã FEC nhưng dùng chúng để chuyển các gói tin đi. Ưu điểm của ba sơ đồ này phụ thuộc vào khoảng cách giữa nguồn và đích, sơ đồ đa bước nhảy sẽ phù hợp hơn với các khoảng cách lớn. Chỉ thực hiện giải mã FEC ở các nút trung gian khi khoảng cách giữa nguồn và đích quá lớn vì nên tránh việc tích lũy quá nhiều lỗi.

6.2.4 Các sơ đồ lại

Từ thảo luận ở trên, rõ ràng là không có một chiến lược điều khiển lỗi đơn giản và cố định để có được hiệu quả năng lượng tối ưu tại mọi thời điểm. Ta sẽ mô tả các thoả hiệp thông qua một ví dụ.

Ví dụ 6.2: Hiệu quả năng lượng của FEC và ARQ

Xem xét các nút bộ phát và bộ thu được nối với nhau qua kênh vô tuyến. Kênh truyền là BSC với tỉ lệ lỗi bit cố định p . Với FEC, ta xem xét trường hợp đặc biệt của mã BCH nhị phân. Các mã này có tính chất (xem phần 4.3) là với mọi số nguyên dương m và t , tồn tại mã BCH nhị phân có độ dài khối mã $n = 2^m - 1$ ở $t.m$ bit mã đầu và có khả năng hiệu chỉnh lên đến t lỗi. Thêm vào đó, giả sử các lỗi không thể hiệu chỉnh phát hiện được là ít nhất, giao thức bit xen kẽ đơn giản có số lần truyền lại giữa hai nút không bị hạn chế. Cố định độ dài khối n ở $n = 1023$. Theo bảng 4.6, khả năng hiệu chỉnh lỗi của $t = 0, 2, 4, 6, 8, 10$ bit nằm trong khoảng $n = 1023$ bit, số lượng k bit dữ liệu có thể chuyển được cho trong bảng 6.1.

Với mọi t , xác suất gói tin $n = 1023$ được truyền đi thành công (có nghĩa số bit lỗi $\leq t$) là:

$$P(n, t, p) = \sum_{i=0}^t \binom{n}{i} (1-p)^{n-i} p^i \quad (0.19)$$

Với giao thức bit xen kẽ qua BSC, số lần thử nghiệm X cần để truyền thành công gói tin là một biến ngẫu nhiên với kỳ vọng:

$$E[X] = \frac{1}{P(n, t, p)} \quad (0.20)$$

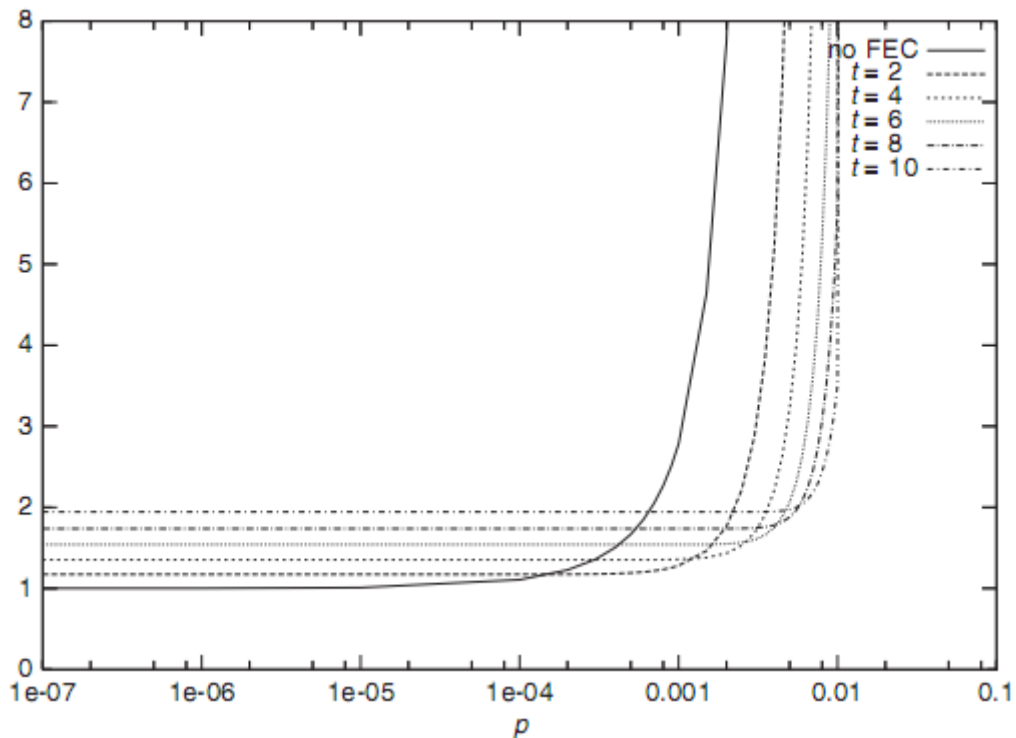
Giả sử chi phí để truyền và nhận một bit đơn ($E_t = 1$) và nếu chi phí giải mã E_{dec} theo phương trình (0.18) (với $E_{\text{add}} = 3,3 \cdot 10^{-4}$ và $E_{\text{mult}} = 3,7 \cdot 10^{-2}$ đơn vị năng lượng, những số liệu này chỉ được chọn cho mục đích mô phỏng), ta có thể biểu diễn năng lượng mong muốn sử dụng cho k bit là:

$$E[Y] = \frac{n \cdot E_t + E_{\text{dec}}}{k \cdot P(n, t, p)} \quad (0.21)$$

Hình 6.7 biểu diễn giá trị mong muốn này với các t khác nhau.

t	k
0	1023
2	1003
4	983
6	963
8	943
10	923

Bảng 6.1 Số lượng dữ liệu k tương ứng với số lượng bit có thể hiệu chỉnh t khác nhau trong khối $n=1023$ bit đối với mã BCH



Hình 6.7 Năng lượng sử dụng để truyền thành công 1 bit dữ liệu với các t khác nhau

Có thể nhận thấy rằng với tỉ lệ lỗi bit cực thấp, mã hoá FEC sẽ làm tăng chi phí lên nhiều so với không có FEC, trên giá trị $p \approx 10^{-4}$ thì dùng FEC sẽ phù hợp hơn. Tuy nhiên, với các kênh truyền có phẩm chất kém, ($p \approx 10^{-2}$ hoặc cao hơn), tất cả các sơ đồ đều phải cần một lượng lớn năng lượng và trong trường hợp đó, các cơ chế khác sẽ được sử dụng, ví dụ như chọn kích thước gói tin phù hợp hơn (phần 6.3).

Mặc dù một số hệ thống trên thực tế khác với lựa chọn đã mô tả ở trên nhưng các kết luận vẫn được giữ nguyên giá trị.

6.2.5 Điều khiển công suất

Mấu chốt điều khiển khác để làm tăng độ tin cậy của quá trình truyền dẫn là công suất truyền hay nói một cách chính xác là công suất đầu ra của bộ phát. Tăng công suất này sẽ làm tăng năng lượng/bit (E_b/N_0) và do đó làm giảm tỉ lệ lỗi bit và các yêu cầu truyền lại. Ebert và Wolisz chỉ ra rằng mạng ad hoc một bước có công suất truyền tối ưu (tương đương với mục tiêu BER) cân bằng với năng lượng phát và các yêu cầu truyền lại đối với chiều dài gói tin đã cho.

Tuy nhiên, trong các mạng lớn hơn với các giao tiếp đa bước nhảy có một chút khác biệt. Nếu một nút tăng công suất truyền của nó thì nó cũng tăng nhiễu đối với các nút khác và do đó, nó sẽ ảnh hưởng đến các tỉ lệ lỗi bit. Trong trường hợp này, vấn đề này lại phụ thuộc vào tình trạng tải.

Narendran đã mô tả một cách đầy đủ sơ đồ phân bố mà trong đó, hai nút cùng nhau chia sẻ việc điều khiển công suất phát và tỉ lệ mã. Bộ phát phù hợp với công suất và mã dựa trên các phép đo tỉ lệ từ lỗi, nhiễu và công suất nhận lấy ở bộ thu và phản hồi về bộ phát. Thuật toán vẫn phù hợp mặc dù nó được triển khai và ước lượng trong hệ thống tế bào trong khi tất cả các thông tin trên lại có hiệu lực ở các trạm gốc. Tuy nhiên, thuật toán này cũng thực sự có hiệu quả khi áp dụng cho các mạng cảm biến.

Phương pháp này dễ giải thích nhất cho các sơ đồ mã hoá khối FEC. Giả sử rằng bộ giải mã FEC ở bộ thu không chỉ phân phối các khối bit mà còn cho biết khối có lỗi hiệu chỉnh được hay không? Qua thông tin này, bộ thu có thể tính **tỉ lệ lỗi từ** (WER - Word Error Rate). Thuật toán này được thực hiện lặp đi lặp lại. Cuối một vòng lặp, bộ thu tính WER trên tất cả các từ thu được trong vòng lặp này và kiểm tra xem liệu nó có nằm trong khoảng giữa ngưỡng thấp và ngưỡng cao không? Nếu nó nằm trong giới hạn này thì gọi đó là **tỉ lệ lỗi từ mong muốn** (DWER - Desired Word Error Rate). Khi đó, bộ thu cung cấp cho bộ phát tỉ lệ lỗi từ công suất nhận và mức nhiễu có thể quan sát được. Bộ thu có dải $[P_{min}, P_{max}]$ hợp lệ mà trong đó, nó có thể chọn công suất đầu ra và tập $\{c_1, \dots, c_N\}$ các sơ đồ mã hoá. Trong mỗi sơ đồ mã hoá có thể tính tỉ số tín hiệu/nhiều (CIR - Carrier to Interference Ratio) yêu cầu để có được DWER, cho nên với mỗi sơ đồ mã hoá c_i , có thể tính mức công suất đầu ra p_i yêu cầu để có được CIR cụ thể này ở bộ thu. Cuối cùng, bộ phát chọn cặp mức công suất/mã có thể đạt được mục đích tổng chi phí năng lượng là nhỏ nhất. Nếu tỉ lệ từ lỗi vượt quá ngưỡng trên ta có **vòng lặp hồng**.

6.2.6 Điều khiển lỗi

Trong phần này ta sẽ xem xét một số phương pháp điều khiển lỗi và các mặt tiêu thụ năng lượng của chúng. Điều khiển lỗi được quan tâm để cải thiện độ tin cậy của quá trình

truyền dẫn kênh truyền, để giải quyết thỏa hiệp vì tăng độ tin cậy hay tăng tỉ lệ lỗi kênh đều yêu cầu nhiều năng lượng hơn. Vì vậy, trước khi bắt đầu thiết kế các sơ đồ điều khiển lỗi, có một số điểm rất quan trọng để ước lượng độ tin cậy yêu cầu. Trong WSN, một điều may mắn là thường các hiện tượng vật lý có mặt trong nhiều nút có thông tin giống nhau hoặc ít nhất cũng liên quan đến nhau và do đó, không cần từng nút đơn phải có độ tin cậy rất cao để truyền được gói tin một cách tốt nhất.

Các sơ đồ mã hoá FEC có lợi thế để có được lợi ích thực tế về hiệu quả năng lượng với độ tin cậy đã cho trong trường hợp môi trường không quá chậm và tỉ lệ lỗi bit không quá cao. Với BER rất thấp và rất cao, phần mào đầu của mã hoá sẽ bị lãng phí do các sơ đồ mã hoá thực tế không có khả năng hiệu chỉnh lỗi hiệu quả để điều chỉnh năng lượng. Với cả hai sơ đồ mã hoá khối và chập, năng lượng yêu cầu cho hoạt động mã hoá không đáng kể nhưng năng lượng giải mã lại rất đáng kể và vượt quá năng lượng yêu cầu. Các sơ đồ mã hoá khối thường có hiệu quả năng lượng cao hơn mã chập nhưng mã chập lại có các khả năng hiệu chỉnh lỗi tốt hơn. Nếu mã hoá và giải mã có thể thực hiện được ở phần cứng thì các sơ đồ FEC sẽ trở nên hấp dẫn.

Mặt khác, các sơ đồ ARQ có thể hoà hợp phần mào đầu của nó với các điều kiện của kênh truyền dẫn. Với các kênh rất tốt, phần mào đầu chỉ là các khung phản hồi và có thể giảm số lượng của nó bằng cách lựa chọn các sơ đồ phản hồi phù hợp. Nói cách khác, trong trường hợp các kênh bị lỗi, các giao thức ARQ chuẩn phải truyền lại cả gói tin, thậm chí cả khi chỉ có vài bit bị sai.

Có ít nhất hai cách đã được đưa ra. Cách thứ nhất là các sơ đồ lai, đó là kết hợp FEC và ARQ theo các cách thích hợp, ví dụ như bằng cách áp dụng sơ đồ FEC đã được tính toán sao cho tất cả các gói tin có giá không quá đắt và đặt giao thức ARQ hiệu chỉnh các lỗi còn lại. Cách thứ hai là cố gắng tìm các **sơ đồ thích hợp** mà sự thay đổi chiến lược điều khiển lỗi của chúng phụ thuộc vào các điều kiện kênh truyền dẫn, ví dụ như chuyển từ Goback- N_1 sang Goback- N_2 ... Tuy nhiên, điều này phụ thuộc vào từng ứng dụng, phần cứng của nút và các mô hình lỗi trong khi độ phức tạp lại gia tăng cho hoạt động tương thích (ước lượng trạng thái của kênh, tín hiệu hoá các hoạt động điều khiển thích nghi), chi phí cho năng lượng...

6.3 Định dạng khung

Trong quá trình định dạng khung, lớp liên kết tạo khung sau đó sẽ truyền đi. Một số chú ý về việc định dạng khung đã được thảo luận trong phần 6.2.2 và phần đánh địa chỉ sẽ được giới thiệu trong chương 7. Trong phần này chỉ trình bày cách **lựa chọn kích thước gói tin**. Ví dụ sau sẽ chứng minh tại sao vấn đề này lại quan trọng.

Ví dụ 6.3: Hiệu quả năng lượng với các gói tin có kích thước khác nhau.

Xét hai nút được nối với nhau qua BSC với xác suất lỗi bit $p \in (0,1)$. Mỗi gói tin có phần mào đầu cố định o bit (header, trailer) và một số thay đổi u bit. Theo đó, xác suất để truyền gói tin đó thành công là:

$$P(o, u, p) = (1 - p)^{o+u} \quad (0.22)$$

Và xác suất của gói bị lỗi (gói có ít nhất một lỗi bit) là $Q(o, u, p) = 1 - P(o, u, p)$. Trong giao thức ARQ bit luân phiên đơn giản với một số hữu hạn lần truyền lại, số lần thử

nghiệm cần thiết để truyền gói tin, X , là một biến ngẫu nhiên với xác suất thành công $P(o, u, p)$ và kỳ vọng:

$$E[X] = \frac{1}{P(o, u, p)} = \frac{1}{(1-p)^{o+u}} \quad (0.23)$$

Các gói tin phản hồi xác nhận chỉ bao gồm mào đầu và cũng có độ dài o bit. Ta sẽ tập trung vào bộ phát. Giả sử rằng năng lượng sử dụng cho việc truyền/ nhận tin có kích thước l bit là:

$$\begin{aligned} e_t(l) &= e_{t,0} + e_t.l \\ e_r(l) &= e_{r,0} + e_r.l \end{aligned} \quad (0.24)$$

Trong đó $e_{t,0}$ và $e_{r,0}$ là chi phí năng lượng cố định được sử dụng cho mỗi lần truyền/nhận tin. Các tham số e_t và e_r là năng lượng sử dụng để truyền/nhận một bit. Các nguồn tiêu thụ năng lượng khác không quan tâm. Năng lượng trung bình được sử dụng để truyền thành công một gói tin là tích số giữa số năng lượng sử dụng trên mỗi gói nhân với số thử nghiệm trung bình cần thiết để truyền thành công gói tin đó. Bộ phát sử dụng năng lượng trên mỗi gói tin là $e_t(o+u) + e_r(o)$ trong đó số hạng đầu tiên dùng để truyền gói dữ liệu, còn số hạng thứ hai dùng cho thử nghiệm để nhận phản hồi (giả sử chế độ rồi và nhận có cùng mức chi phí). Tổng năng lượng mong muốn để dùng cho một gói tin là:

$$\frac{e_t(o+u) + e_r(o)}{(1-p)^{o+u}}$$

Và năng lượng sử dụng cho mỗi bit là:

$$h(o, u, p) = \frac{e_t(o+u) + e_r(o)}{u.(1-p)^{o+u}}$$

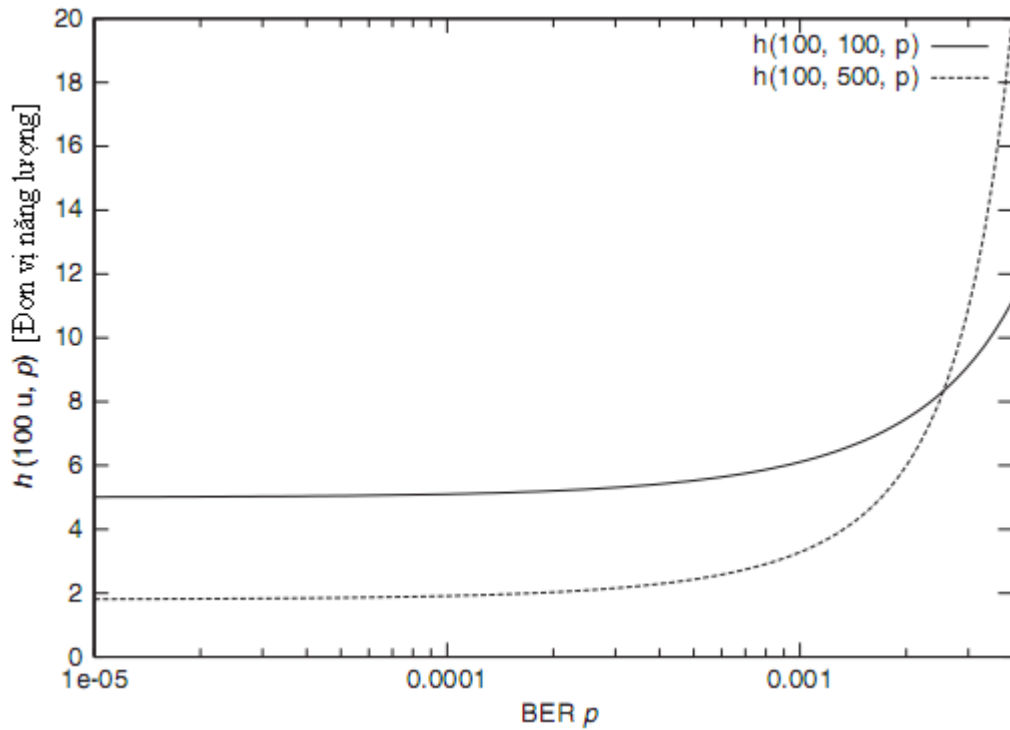
Hàm này được vẽ trên hình 6.8 với tỉ lệ lỗi bit p khác nhau và các giá trị cố định là $e_{r,0} = e_{t,0} = 100$, $e_r = e_t = 1$ và $o = 100$ với hai kích thước dữ liệu khác nhau ($u = 100$ bit và $u = 500$ bit).

Có hai điểm cần chú ý: Sự khác nhau có thể đáng kể, nó phụ thuộc vào tỉ lệ lỗi bit p và các tham số $e_{r,0}$, $e_{t,0}$, e_r , e_t , còn lựa chọn u làm tăng hiệu quả năng lượng đối với giá trị p đã cho. Thông thường, có thể tiết kiệm rất nhiều năng lượng nếu kích thước gói tin được chọn một cách hợp lý. Các kết quả thử nghiệm trên WLAN đã xác nhận điều đó.

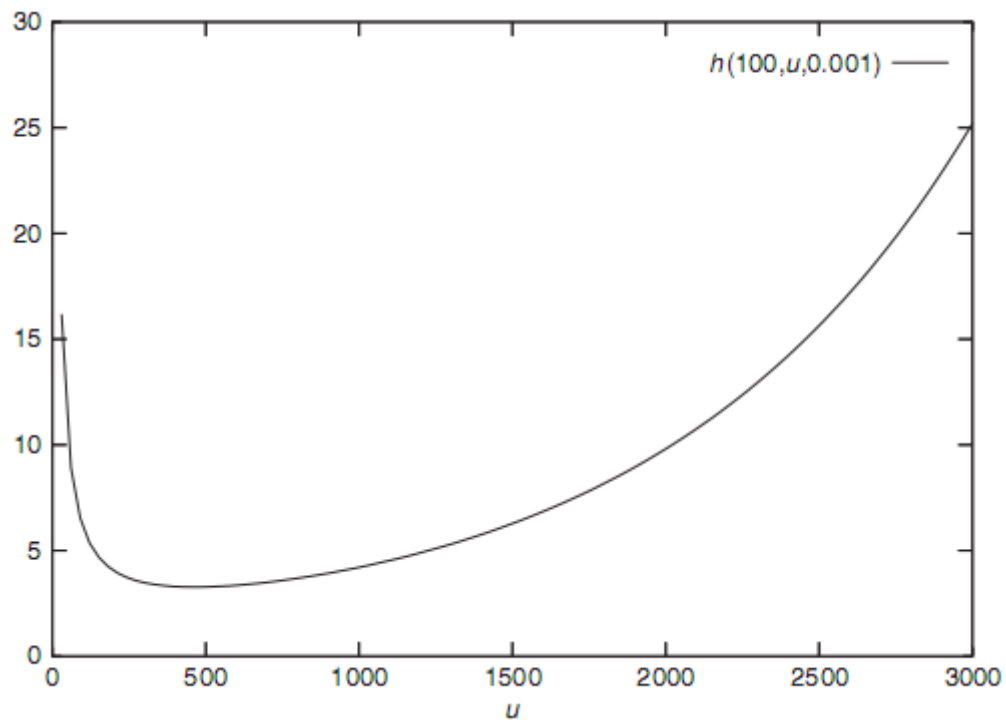
Rõ ràng là mào đầu của khung lớn sẽ phù hợp với các gói tin có kích thước lớn để có được một cách hợp lý hiệu quả năng lượng trên mỗi bit. Nói cách khác, các gói tin lớn sẽ nhạy cảm hơn với các lỗi bit nếu không áp dụng các cơ cấu như FEC. Vì vậy, có thể thấy rằng với tỉ lệ lỗi bit tức thời p đã cho sẽ tồn tại **gói tin có kích thước tối ưu**, khi đó nó sẽ rất dễ

phân tích (giải $\frac{\partial h(o, u, p)}{\partial u} = 0$ và kiểm tra tính tối thiểu). Với tập giá trị đã cho trong ví dụ trên, hình 6.9 biểu diễn năng lượng trên mỗi bit với kích thước gói dữ liệu u khác nhau và tỉ lệ

lỗi bit cố định $p = 0.001$. Hình này cho giá trị cực tiểu tại $u \approx 463$ bit. Tuy nhiên, cũng cần chú ý rằng nếu chọn giá trị u nhỏ hơn thì mức tiêu thụ năng lượng sẽ tăng theo đường dốc và do đó, bộ phát phải được điều khiển kích thước gói tin của nó một cách cẩn thận. Kích thước gói tin có được từ các tham số kỹ thuật và ước lượng xác suất lỗi bit p .



Hình 6.8 Năng lượng/bit hữu dụng cho dữ liệu có kích thước $u = 100$ và $u = 500$ với tỉ lệ lỗi bit khác nhau



Hình 6.9 Năng lượng/bit hữu dụng với tỉ lệ lỗi bit p cố định và kích thước dữ liệu u thay đổi

Các giả thiết về BSC thường đơn giản và khác xa thực tế. Nếu các lỗi kênh bị ảnh hưởng bởi fading thì sẽ không có gói tin nào được truyền qua kênh do tình trạng kênh xấu, đó là chưa kể đến kích thước của nó. Nếu tình trạng kênh tốt thì các gói tin có kích thước lớn sẽ có hiệu suất và hiệu quả năng lượng tốt hơn. Siew and Goodman đã có gói tin có kích thước tối ưu khi công suất của kênh fading Rayleigh thay đổi giữa các fading sâu, ở đó các gói tin giả sử bị lỗi, các chu kỳ kênh tốt và các gói tin qua kênh một cách nguyên vẹn. Để các gói tin được truyền một cách đúng đắn thì nó phải nằm hoàn toàn trong giai đoạn tốt. Kích thước càng dài thì nó càng dễ gặp fading. Đôi khi tồn tại một số gói tin có kích thước tối ưu.

6.3.1 Các sơ đồ tương thích

Kênh vô tuyến không có các điều kiện lỗi tĩnh mà bị ảnh hưởng cả trong khoảng thời gian ngắn (fading nhanh) và dài (fading chậm). Môi trường biến thiên theo thời gian dường như rất phù hợp để cho phép các nút bộ thu và bộ phát **ước lượng các điều kiện kênh** một cách liên tục và do đó, nó sẽ tương thích với kích thước của gói tin.

Tuy nhiên, ý tưởng này có một hạn chế. Ứng dụng hay giao thức lớp cao hơn sẽ hoạt động như thế nào trong nút cảm biến có kích thước gói tin thay đổi? Nếu lớp liên kết chuyển các thay đổi đến lớp ứng dụng hay các lớp cao hơn thì sau đó phải quan tâm đến việc phân đoạn dữ liệu một cách phù hợp để truyền dẫn và để đưa vào các ứng dụng. Nói cách khác, nếu các gói tin có kích thước khác nhau được đưa vào ứng dụng thì công việc **phân đoạn và ghép lại** sẽ được thực hiện ở lớp liên kết, tức là độ phức tạp sẽ tăng lên. Việc này cũng bao gồm phân loại gói tin và mã hoá FEC ở vị trí phù hợp để gói tin có thể dùng cho các ứng dụng thực sự khác nhau. Một cách lý tưởng, lớp MAC sẽ hỗ trợ việc truyền dẫn từng đoạn của gói tin, ví dụ như giao thức S-MAC (phần 5.2.2).

Làm cách nào để ước lượng các điều kiện kênh truyền dẫn một cách tức thời, đặc biệt là tỉ lệ lỗi bit tức thời p ? Một sơ đồ chuẩn đã được sử dụng được giới thiệu bởi Lettieri và Srivastava, nó cho phép bộ thu lựa chọn thông tin có chất lượng của kênh và truyền nó về bộ phát bằng các khung phản hồi để sau đó điều chỉnh kích thước khung. Bộ thu có thể sử dụng các thông số sau:

- Thông tin thành công có thể đạt được bằng sơ đồ ARQ không lai (ví dụ như số lần truyền lại trên mỗi gói tin quan sát được) có thể dùng để ước lượng tỉ lệ lỗi bit tức thời.
- Nếu sử dụng FEC thì bộ giải mã FEC có thể cung cấp thông tin về số lượng lỗi được hiệu chỉnh và nó sẽ được bộ thu phản hồi về bộ phát.

Phương pháp đầu tiên có ưu điểm là đưa ra ít giả thuyết nhất về các khả năng của bộ thu phát. Sau đây ta sẽ thảo luận kỹ hơn vấn đề này.

Giả sử kênh có thể được làm theo từ BSC cho thời kỳ quan sát nào đó và độ dài gói tin không đổi trong thời kỳ này, bộ thu/ bộ phát có thể ước lượng tỉ lệ thành công của gói tin hiện tại $P(o, u, p)$, ví dụ như bằng cách đếm toàn bộ số thử nghiệm T được dùng để truyền M gói tin (nếu bộ thu lựa chọn thông tin này thì nó phải trả lời cho bộ phát). Số T/M là kỳ vọng của biến ngẫu nhiên với tham số thành công $P(o, u, p)$ và giá trị này được sử dụng khi giải phương trình (0.22) theo p . Phương pháp tương tự cũng được giới thiệu, trong đó các kích thước gói tin được lựa chọn từ một số kích thước gói tin đã cho. Modiano đã đề xuất hai phương pháp khác nhau. Trong phương pháp đầu tiên, bảng được tính trước, nó liên quan đến số lần truyền lại R được yêu cầu cho M gói tin đã truyền đi với kích thước gói tin đã cho k để

chọn ra được kích thước gói tin tối ưu từ một tập các gói cố định. Trong phương pháp thứ hai, nguyên tắc **cực đại hoá khả năng để ước lượng tham số** (MLE - Maximum Likelihood Estimation) đã được đưa ra. Mô tả hoạt động của phương pháp này thông qua ví dụ sau.

Ví dụ 6.4: Phương pháp MLE để thích ứng với các kích thước của gói tin

BSC với xác suất lỗi bit p và các gói tin có mào đầu o bit, u bit dữ liệu, xác suất lỗi gói tin là:

$$Q(o, u, p) = 1 - (1 - p)^{o+u} \quad (0.25)$$

Và xác suất để M gói tin cuối cùng truyền lại R lần là:

$$\Pr[R|p] = \binom{M}{R} Q(o, u, p)^R \cdot (1 - Q(o, u, p))^{M-R} \quad (0.26)$$

Giả sử rằng o và u đã biết, còn R và M đang quan sát. Giá trị của p thường phù hợp vì đây là trạng thái được yêu cầu. Giá trị này của p làm $\Pr[R|p]$ đạt giá trị cực đại. Nó có thể được xác định bằng cách lấy đạo hàm $d\Pr[R|p]/dp$ và giải $d\Pr[R|p]/dp = 0$ theo p . Kết quả là:

$$\mu = 1 - \left(\frac{M - R}{M} \right)^{\frac{1}{u}} \quad (0.27)$$

Sau đó rất dễ để tìm được kích thước gói tin u để cho hiệu quả năng lượng tốt nhất. Vấn đề của bộ ước lượng này xảy ra khi tỉ lệ lỗi rất thấp hoặc rất cao. Trong trường hợp tỉ lệ lỗi rất cao, ta thường có $M = R$ và bộ ước lượng cho $\mu = 1$. Ngược lại, nếu tỉ lệ lỗi bit cực thấp thì $\mu = 0$, điều này làm cho độ dài gói tin là vô hạn. Vì vậy, độ dài gói tin cực đại và cực tiểu hợp lý sẽ được chọn như là các giá trị giới hạn.

Vấn đề có tính chất quyết định ở đây là lựa chọn **độ dài thời kỳ quan sát**. Nếu thời kỳ này quá ngắn sẽ không đủ dữ liệu để có được ước lượng chính xác xác suất lỗi bit p . Mặt khác, nếu thời kỳ này quá dài thì thuật toán có thể chiếm quá nhiều thời gian để có thể thích ứng với các thay đổi điều kiện kênh truyền và gói tin có kích thước gần tối ưu sẽ sử dụng trong thời gian quá dài.

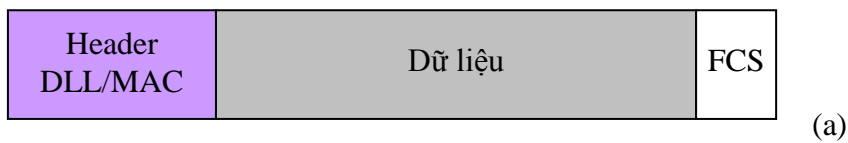
Tất cả các phương pháp dùng để ước lượng các điều kiện kênh truyền tức thời dùng chúng như là dự đoán ngắn hạn chất lượng kênh. Nếu nút muốn gửi một gói tin mới sau một thời kỳ im lặng dài thì tất cả thông tin về kênh truyền sẵn có sẽ là cũ và có thể trở nên vô dụng.

6.3.2 Các sơ đồ kiểm tra tổng trung gian

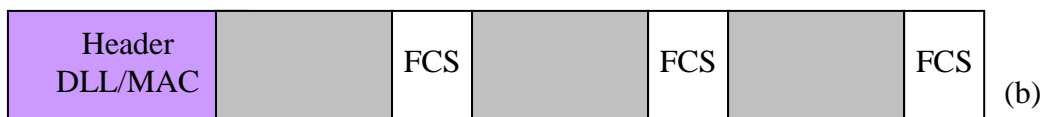
Lettieri, Srivastava và Willig thảo luận sơ đồ cho phép sử dụng các gói tin dài mà không yêu cầu sơ đồ phân đoạn/ ghép lại. Phương pháp này có ưu thế trong các trường hợp chỉ một vài bit trong gói tin bị lỗi và nó sẽ cứu được nhiều nhất các bit đúng. Việc truyền lại bị hạn chế, chỉ khi các phần trên của gói tin thực sự xảy ra lỗi. Sơ đồ kiểm tra tổng trung gian có thể được tích hợp với các giao thức ARQ chuẩn.

Quan sát hình 6.10. Nếu có u bit dữ liệu, các giao thức với sơ đồ định khung header /dữ liệu/trailer thông thường (hình 6.10a) đặt header o bit trước dữ liệu và trailer h bit sau dữ liệu. Header thường mang địa chỉ của nguồn và đích, thông tin độ dài khung và thông tin điều khiển trong khi trailer chứa kiểm tra tổng của khung. Do đó, toàn bộ khung có kích thước $o + s + h$. Trong trường hợp phải truyền lại thì cả $o + s + h$ bit sẽ phải truyền lại. Ngược lại, trong **sơ đồ kiểm tra tổng trung gian** (hình 6.10b), u bit dữ liệu được chia thành L **đoạn**, mỗi đoạn có c bit chưa xử lý với kiểm tra tổng h' bit gắn vào. Đoạn cuối cùng phải có ít hơn c bit. Khung được tạo ra bằng cách gắn tất cả các đoạn để header khung có kích thước $o' > o$ bit và toàn bộ khung có kích thước $o' + L(c + h')$ bit. Header khung trong sơ đồ kiểm tra tổng trung gian còn bao gồm việc phân chia kiểm tra tổng header và thông tin về kích thước/ số lượng đoạn.

Định khung truyền thông:



Định khung kiểu kiểm tra tổng trung gian:



Hình 6.10 Định khung truyền thông với header/dữ liệu/kiểm tra tổng so với định khung kiểu kiểm tra tổng trung gian

Bộ thu hành động như sau. Nếu nó phát hiện được lỗi trong header khung thì toàn bộ khung sẽ bị loại bỏ và bộ phát sẽ phải truyền lại toàn bộ khung cuối cùng. Nếu header đúng, bộ thu sẽ kiểm tra từng đoạn riêng rẽ và đệm đúng các đoạn. Nếu tất cả các đoạn là đúng, bộ thu sẽ phân phát khung đến các lớp cao hơn của nó và gửi **phản hồi cuối cùng**. Nếu có một số đoạn không đúng, các đoạn lỗi sẽ ra dấu cho bộ phát bằng một **phản hồi không hoàn chỉnh**. Bộ phát *chỉ truyền lại các khung bị lỗi*. Ví dụ, nếu khung đầu tiên có $L = 8$ đoạn và bộ thu nhận 5 trong số 8 đoạn đó, tức là có 3 khung bị lỗi. Phương pháp này có ưu điểm là số khung truyền lại nhỏ hơn rất nhiều, năng lượng tiêu thụ giảm, tạo ra ít nhiễu, ít lỗi và đến bộ thu với trễ nhỏ hơn. Nó cũng có nhược điểm là các kiểm tra tổng trung gian có mào đầu lớn hơn, điều này có thể tránh với một số hệ số khuếch đại ở đầu vào tốt và giảm năng lượng đối với các tỉ lệ lỗi bit nhỏ. Tuy nhiên, câu hỏi đặt ra ở đây là kích thước đoạn tối ưu có tăng không? Willig đã chỉ ra rằng, BSC với tỉ lệ lỗi bit từ trung bình đến cao ($\geq 10^{-4}$), sơ đồ kiểm tra tổng trung gian có thể có hiệu suất cao hơn và cần phải giảm số khung so với sơ đồ khung thông thường, thậm chí cả khi kích thước khung được chọn là tối ưu. Kích thước đoạn phải phù hợp với sơ đồ đơn giản, khi mà bộ phát đếm số đoạn yêu cầu cho việc truyền lại, tổng số đoạn được truyền và sử dụng dữ liệu này để ước lượng tỉ lệ lỗi bit tức thời.

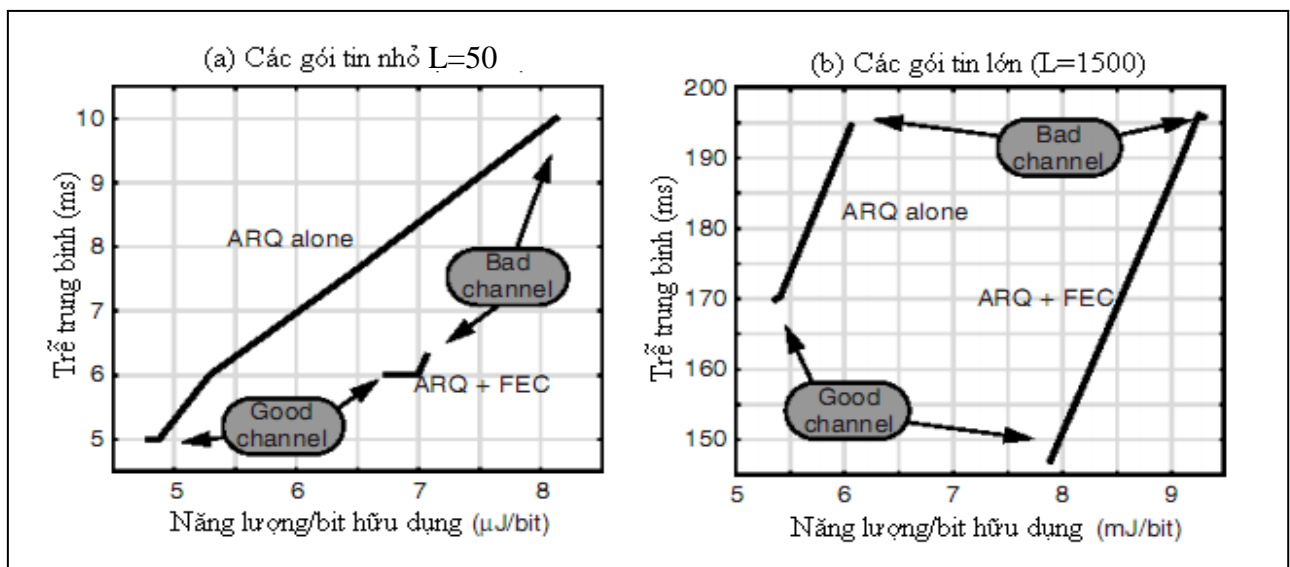
Nhược điểm của sơ đồ này là việc tính toán CRC không dễ. Trong sơ đồ truyền thông, CRC có thể có được bằng cách dịch header và dữ liệu qua thanh ghi dịch phản hồi tuyến tính,

tất cả nội dung được nối với nhau một cách đơn giản. Sơ đồ kiểm tra tổng trung gian yêu cầu nhiều quá trình tính toán CRC phức tạp hơn.

6.3.3 Kết hợp tối ưu kích thước gói tin và FEC

Tối ưu kích thước gói tin có thể kết hợp với FEC. Sankarasubramaniam đã nghiên cứu mã BCH và mã chập về mặt hiệu quả năng lượng. Với kênh truyền loại BSC và với tỉ lệ lỗi bit trung bình $p = 0.001$, chiều dài gói tin tối ưu (bao gồm dữ liệu và mào đầu) để việc tiêu thụ năng lượng có hiệu quả tăng khi sử dụng mã BCH so với trường hợp không có hiệu chỉnh lỗi. Với khả năng hiệu chỉnh 6 lỗi, kích thước gói tin tối ưu sẽ là 2047 bit, mối quan hệ giữa kích thước khối $n = 2047$, kích thước dữ liệu k và khả năng hiệu chỉnh lỗi $t = 6$ được giới thiệu trong phần 6.2.4, gói tin có thể có ít nhất $k \geq 2047 - 6.11 = 1981$ bit dữ liệu và và khung đó sẽ phân phát thành công với xác suất $P(2047, 6, 0.001) \approx 0.995$ (sử dụng phương trình (0.19)). Hara mã các gói tin có kích thước khác với mã Reed-Solomon và hiệu suất cho giao thức bit xen kẽ cũng được đánh giá. Với tỉ số tín hiệu/ nhiễu rất tốt (tỉ lệ lỗi bit thấp), giao thức không mã hoá sẽ tốt hơn giao thức mã hoá, SNR cao hơn sẽ có mã hoá tốt hơn.

Tối ưu kích thước gói tin, sơ đồ điều khiển lỗi kết hợp FEC (các mã Reed-Solomon và mã chập tỉ lệ $1/2$), lặp lại có lựa chọn ARQ đã được ứng dụng cho các kênh fading Rayleigh. Kênh truyền được mô hình hoá như một chuỗi Markov hai trạng thái với các trạng thái "tốt" và "xấu" và nó hoạt động theo mô hình BSC ở cả hai trạng thái. Ở trạng thái xấu, tỉ lệ lỗi bit được giả sử là 0.5 trong khi BER ở trạng thái tốt thay đổi. Xác suất truyền giữa các trạng thái có được từ mô hình vật lý và chỉ dựa vào các tham số vật lý đơn giản. Khoảng thời gian giữa các trạng thái fading/ xấu của kênh và trạng thái tốt phụ thuộc vào tốc độ của trạm di động (thông qua tần số Doppler của nó). Ví dụ, trạm (nút) di chuyển nhanh thì nhiều fading xảy ra trong một gói tin và làm cho kênh giống mô hình BSC, nếu trạm di chuyển rất chậm thì một fading có thể trải qua nhiều gói tin. Ảnh hưởng của sự thay đổi kích thước gói tin, FEC và ARQ được nghiên cứu trong ba ứng dụng điển hình là (i) gói thông tin (datagram) dữ liệu đơn giản được tạo ra từ một nguồn dữ liệu bão hoà, (ii) dữ liệu thoại tuần hoàn với hạn chế trễ (gói tin không được chuyển đi thành công cho đến khi hết thời hạn) và (iii) lưu lượng TCP. Tỉ lệ lỗi bit ở trạng thái tốt là 10^{-2} hoặc 10^{-8} tương ứng với hai kênh khác nhau là *tốt* và *xấu*. Một số điểm cần lưu ý:



Hình 6.11 Tiêu thụ năng lượng và trễ mong đợi của các gói thông tin có kích thước khác nhau L (theo byte) và các tỉ lệ lỗi trạng thái tốt khác nhau.

- Trong trường hợp gói thông tin, với kích thước gói tin là nhỏ (50 byte), kết quả được chỉ ra trên hình 6.11. Ở đây có sự kết hợp giữa FEC và ARQ cho cả kênh tốt và xấu. Các giá trị gần như giống nhau (năng lượng/ bit hữu dụng, trễ gói tin mong muốn). Sơ đồ chỉ với ARQ có trễ nhỏ, tiêu thụ năng lượng ít cho kênh tốt và trễ lớn, tiêu thụ năng lượng lớn cho kênh xấu. Với gói tin có kích thước 1500 byte, ARQ + FEC thường có trễ tốt hơn ở mức chi phí năng lượng cao hơn cho kênh tốt, trong khi ở kênh xấu thì cả hai sơ đồ có cùng trễ trung bình, với ARQ + FEC yêu cầu nhiều năng lượng hơn một cách đáng kể.

- Với nguồn thoại, nó cho thấy tỉ lệ lỗi bit thấp trong thời kỳ kênh tốt, sơ đồ ARQ yêu cầu năng lượng ít hơn ARQ + FEC, cả hai sơ đồ đều không bị rớt gói tin do vi phạm thời hạn. Với các tỉ lệ lỗi bit cao hơn, sơ đồ chỉ có ARQ yêu cầu nhiều năng lượng hơn và ở cùng thời điểm thì bị rớt nhiều gói tin hơn sơ đồ ARQ+FEC.

6.4 Quản lý tuyến.

Các lớp cao hơn, đặc biệt là giao thức định tuyến, cần phải biết về **các nút lân cận khả dụng** và **chất lượng tuyến** của các chúng. Thông tin về chất lượng có thể được dùng để đưa ra các quyết định định tuyến bằng cách tránh các tuyến xấu với khả năng suy hao tin cao. Một số chú ý:

- Chất lượng tuyến không phải là giá trị nhị phân, tức nó có nhiều trạng thái chứ không phải chỉ có "tốt" và "xấu". Một tham số đặc trưng cho chất lượng tuyến là xác suất suy hao tin qua tuyến này.

- Chất lượng tuyến thay đổi theo thời gian, ví dụ như do nút di chuyển hay chương ngại vật di chuyển giữa hai nút.

- Chất lượng phải ước lượng được, bằng cách gửi các gói thăm dò và đánh giá đáp ứng hoặc bằng cách nghe lén và phân xử việc truyền dẫn của các nút lân cận. Cả hai phương pháp này đều tiêu tốn năng lượng và trong một số trường hợp, nó đã có sẵn để dùng bằng cách dùng giao thức MAC như là một phần của việc khám phá vùng lân cận, ví dụ như TRAMA (Traffic-Adaptive Medium Access - Truy cập môi trường lưu lượng thích ứng) (phần 5.4.3) hoặc S-MAC (phần 5.2.2).

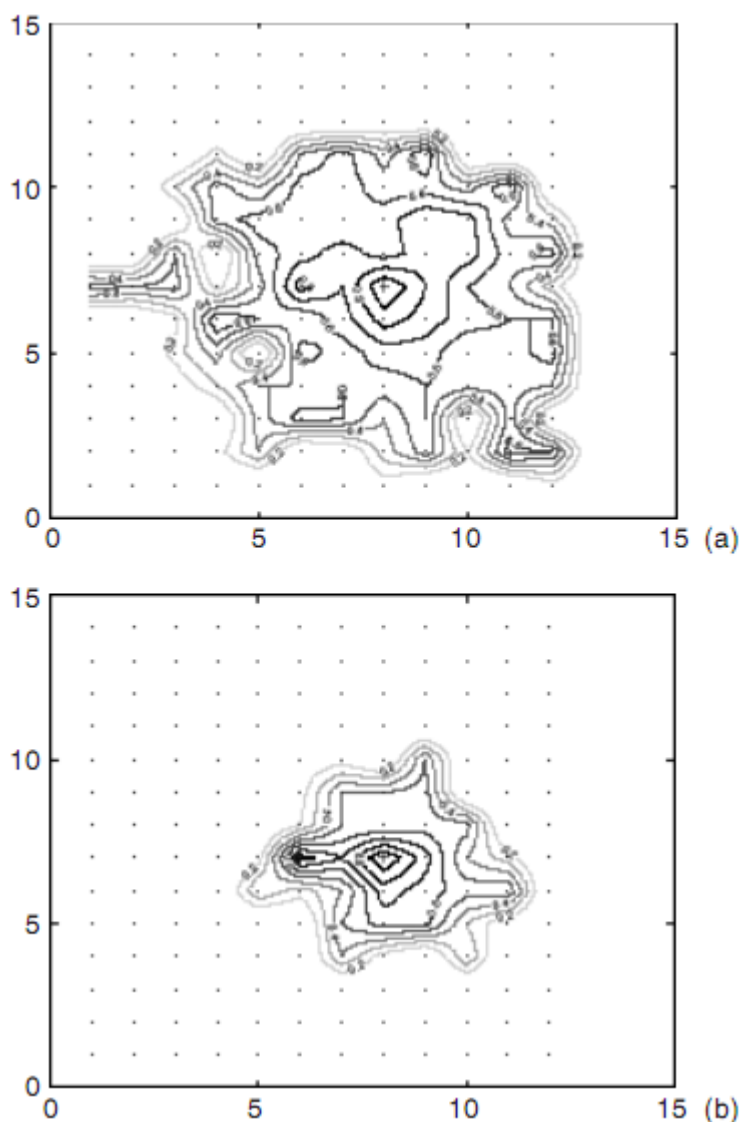
Các nút lân cận và chất lượng tuyến liên quan của nó thường được lưu trong **bảng vùng lân cận**, có thể truy cập bằng các lớp cao hơn. Trong trường hợp mạng có mật độ nút cảm biến dày đặc, các nút rẻ và bị hạn chế bộ nhớ sẽ xảy ra trường hợp không đủ bộ nhớ để chứa tất cả các lân cận có thể. Khi đó phải lựa chọn các lân cận có chất lượng tuyến tốt nhất. Phần tiếp theo sẽ trình bày kỹ hơn về các đặc tính của chất lượng tuyến, các yêu cầu và phương pháp để đánh giá nó.

Quá trình phát hiện vùng lân cận thường được thực hiện bằng một phần của các giao thức MAC (ví dụ TRAMA hay S-MAC) hay các giao thức cấp phát địa chỉ (chương 7). Nó được lặp lại lần này qua lần khác để điều chỉnh theo sự thay đổi cấu hình mạng.

6.4.1 Các đặc tính của chất lượng tuyến

Woo và Ganesan biểu diễn chất lượng tuyến qua tỉ lệ suy hao tin. Đặc biệt, Ganesan biểu thị các phép đo chất lượng tuyến bằng một lưới 13×13 , tức 169 điểm đặt ở vị trí hỏ và cách nhau 2 ft. Trong môi trường này, một nút đang ở thời điểm truyền tin còn các nút khác đang cố nhận nó. Kết quả quan trọng nhất của môi trường này là:

- Với công suất phát đã cho, không có mối quan hệ tiên định nào giữa khoảng cách và chất lượng tuyến. Các nút ở cùng khoảng cách với bộ phát có thể có tỉ lệ suy hao tin rất khác nhau. Trong các trường hợp giới hạn, vùng lân cận không thể nghe gói tin của nút nhưng các nút ở xa đôi khi lại có thể.



Hình 6.12 Đường cong biểu thị xác suất nhận tin cho các gói tin được tạo ra từ nút trung tâm với hai công suất khác nhau

- Vùng xung quanh nút có cùng một tỉ lệ suy hao tin không có dạng hình tròn, nó có dạng không đều. Điều này được mô tả trên hình 6.12, nó là đường viền của xác suất nhận tin khi một nút trung tâm truyền tin. Các đường này là đẳng trị, tức các điểm trên đường có cùng xác suất nhận tin.

- Có một lượng đáng kể các tuyến không đối xứng. Trong một tuyến không đối xứng, các gói tin được gửi từ nút A đến nút B và được nhận bởi nút B với một chút suy hao nhưng

ngược lại, A nhận các gói tin từ B với xác suất suy hao tin cao hơn. Tỷ lệ các tuyến không đối xứng tăng theo khoảng cách, nó chiếm từ 5 đến 15% tất cả tuyến truyền tin.

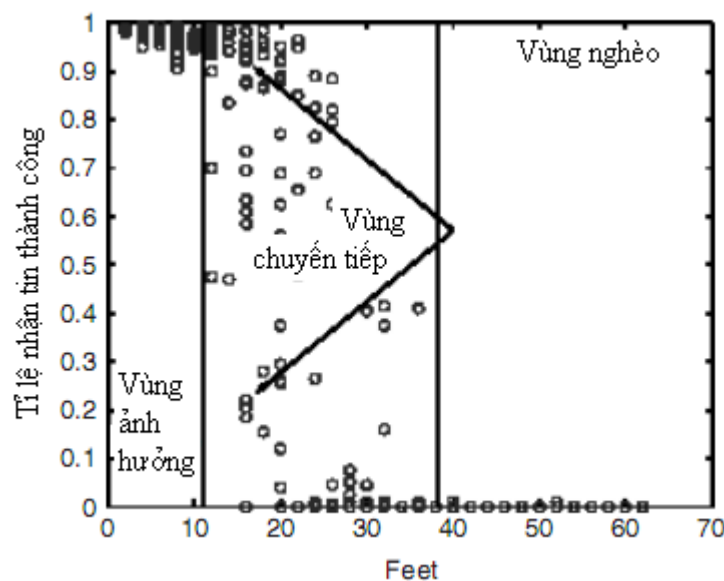
- Tỷ lệ suy hao tin thay đổi theo thời gian, thậm chí ngay cả khi các nút lân cận đứng yên. Mặc dù tỷ lệ suy hao tin trung bình với khoảng cách đã cho theo thời gian là không đổi nhưng nó có sự thay đổi nhất định trong thời gian ngắn.

Woo đã giới thiệu kết quả phép đo đối với mạng tuyến tính. Các nút được sắp xếp theo hàng với khoảng cách 2 ft. Mỗi nút truyền 200 gói tin và tất cả các nút khác đều cố gắng nhận được chúng, nhưng tại một thời điểm chỉ có một nút được truyền. Theo cách này, sẽ có thể có nhiều phép đo khác nhau ở cùng một khoảng cách giữa bộ phát và các bộ thu. Mỗi quan hệ giữa chất lượng – khoảng cách (chất lượng được đánh giá theo tỷ lệ nhận tin) được đặc trưng bởi ba **vùng** khác nhau như hình 6.13.

Trong **vùng ảnh hưởng**, nút thu có khoảng cách ít nhất 10 ft đến bộ phát và có hơn 90% gói tin được nhận bởi nút nằm trong vùng này.

Vùng nghèo bắt đầu bằng khoảng cách 40 ft giữa bộ phát và các bộ thu, các nút có tỷ lệ suy hao trên 90%.

Trong **vùng chuyển tiếp** ở giữa, thay đổi tỷ lệ suy hao của các nút ở cùng khoảng cách là đáng kể.



Hình 6.13 Tỷ lệ nhận tin ở các bộ thu có khoảng cách khác nhau

Mỗi hình tròn nhỏ biểu diễn chất lượng tuyến của đường trực tiếp trong cấu hình mạng.

Mỗi đường có khoảng cách như nhau nhưng có độ tin cậy rất khác nhau.

Tất cả những vấn đề trên xảy ra ở các công suất truyền khác nhau, mặc dù ở các phạm vi khác nhau. Tóm lại, *chất lượng tuyến phải hiểu theo hướng thống kê và thay đổi theo thời gian*.

6.4.2 Ước lượng chất lượng tuyến

Nếu nút muốn ước lượng chất lượng tuyến truyền đến nút lân cận, nó phải làm điều đó bằng các gói nhận từ nút lân cận và phán đoán chất lượng của chúng hay tính toán tỷ lệ suy hao. Do có sự thay đổi lớn về tỷ lệ suy hao đối với cùng một khoảng cách nên nó sẽ không đủ

để có được thông tin về suy hao gói tin từ khoảng cách đã biết đến nút tiếp theo. Sau đây là một số tính chất mong muốn của **bộ ước lượng**:

Tính chính xác: Nó phải tập hợp đủ kết quả và đưa ra các thống kê có giá trị.

Độ nhạy: Nó phải phát hiện được các thay đổi về điều kiện tuyến truyền đủ nhanh, ví dụ như do sự di chuyển của các nút.

Độ ổn định: Ước lượng phải không có các dao động ngắn/ tạm thời trong chất lượng tuyến.

Tính hiệu quả: Để tránh phải nghe quá nhiều việc truyền dẫn của các nút khác do điều này sẽ phải sử dụng nguồn năng lượng quý giá. Hơn nữa, sự phức tạp của tính toán và số lượng bộ nhớ cần để giữ các thống kê tuyến sẽ làm cho các nút cảm biến không dây luôn ở trong tình trạng nằm ở giới hạn cho phép. Đây là một cản trở để thực hiện các thuật toán lọc tiên tiến hơn như các bộ lọc Wiener hay Kalman.

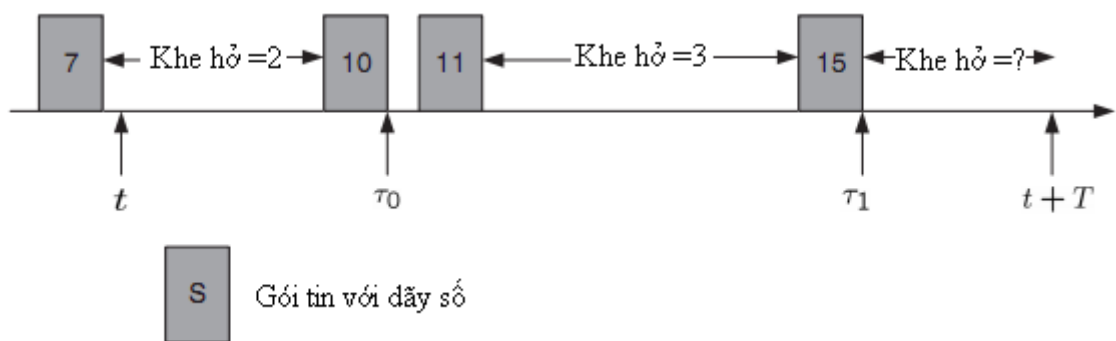
Có hai loại bộ ước lượng là **chủ động** và **bị động**.

Bộ ước lượng chủ động: Trong bộ đánh giá chủ động, nút gửi đi các gói tin đo chuyên dụng và lựa chọn phản hồi từ các nút lân cận. Bằng cách lặp lại điều này, sẽ không cần phải lưu các thống kê tuyến.

Bộ ước lượng bị động: Trong bộ đánh giá bị động, nút nghe lén quá trình truyền dẫn từ các nút lân cận và ước lượng tỉ lệ suy hao từ quan sát số lượng dây của chúng. Suy hao gói tin được phát hiện từ các khe hở trong mỗi lần thu.

Ước lượng bị động thực sự khả thi nếu các nút lân cận tạo ra lưu lượng đáng kể trong một khoảng thời gian nào đó. Nó rất phù hợp khi các quá trình truyền dẫn sử dụng nhiều năng lượng hơn trạng thái nhận và nghỉ.

Điều chỉnh ước lượng bị động được mô tả trong hình 6.14. Sự kiện đầu vào bộ ước lượng là các gói tin đến; gói tin suy hao có thể được suy luận gián tiếp bằng cách sử dụng dãy số này. Bộ ước lượng phải quan sát một số sự kiện trước khi đưa ra ước lượng để có được các kết quả thống kê có ý nghĩa.



Hình 6.14 Mô tả vấn đề ước lượng

Giả sử bộ ước lượng tại thời điểm $t+T$ đưa ra ước lượng tỉ lệ suy hao gói tin trong phạm vi $(t, t+T]$, trong đó T là thời kỳ quan sát (ví dụ như $T = 30s$). Giả sử vì một lý do nào đó bộ ước lượng xác định được dãy cuối cùng mà nó có thể nhìn thấy ngay trước thời điểm t là số 7. Dựa vào gói tin đến số 10 ở thời điểm τ_0 nó biết rằng có hai gói tin đã bị lỡ trong khoảng thời gian $(t, \tau_0]$. Dựa vào gói tin đến số 15 ở thời điểm τ_1 nó biết rằng chỉ 3 trong 8

gói tin nhận được trong khoảng thời gian $(t, \tau_1]$. Vậy giá trị nào sẽ được đưa ra tại thời điểm $t+T$? Để đưa ra được ước lượng đáng tin cậy, ta cần số lượng gói tin bị mất trong khe hở cuối cùng. Nếu gói tin được tạo ra định kỳ thì quyết định này dường như dễ thực hiện. Lớp MAC và ứng dụng có thể đưa ra một số biến động ngẫu nhiên và khi đó, quyết định chỉ có thể đưa ra dự đoán hợp lý về kích thước khe hở.

Giả sử số lượng này là khả dụng thì chúng có thể được sử dụng theo các cách khác nhau trong các bộ ước lượng. Woo và Culler đã khám phá ra một số bộ ước lượng bị động, bao gồm **trung bình động trọng số theo số mũ** (EWMA - Exponentially Weighted Moving Average) (các hệ số được giảm dần theo hàm số mũ), trung bình động thuần túy, trung bình động thời gian trọng số và giá trị trung bình cửa sổ với EWMA (WMEWMA - window mean with EWMA). Gần đây đã ra được sự thỏa hiệp tốt nhất giữa tính ổn định, độ nhạy và công việc như sau: Bộ ước lượng đưa ra các dự đoán $\hat{\mu}_n$ chỉ tại các thời điểm $t_n = t + n.T$, ký hiệu là $\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3 \dots$, đây là dãy ước lượng tại các thời điểm này. Bộ ước lượng có hai tham số hiệu chỉnh là $\alpha \in (0, 1)$ và $T \in \mathbb{N}$, được biểu diễn trong các khoảng thời gian của các gói tin có kích thước cố định. Để cập nhật tức thời t_n , đặt r_n là số gói tin nhận được trong $(t_{n-1}, t_n]$ và f_n là số gói tin nhận dạng được khi bị mất trong khoảng $(t_{n-1}, t_n]$. Khi đó:

$$\mu_n = \frac{r_n}{r_n + f_n} \quad (0.28)$$

$$\hat{\mu}_n = \alpha \cdot \hat{\mu}_{n-1} + (1 - \alpha) \cdot \mu_n \quad (0.29)$$

Như vậy, để biết được lịch sử của bộ đánh giá, ta cần biết $\hat{\mu}_{n-1}$, ngoài ra còn cần cả f_n và r_n nữa.

6.5 Tổng kết

Thiết kế lớp liên kết phải dựa trên hiệu quả năng lượng và vì vậy nó phụ thuộc nhiều vào các đặc tính tiêu thụ năng lượng của lớp vật lý ở dưới và các đặc tính của tải. Có một số mâu chốt điều khiển có thể sử dụng để tiết kiệm năng lượng, ví dụ như FEC hay tối ưu kích thước gói tin. Nếu ứng dụng mạng cảm biến được đặc trưng bởi quá trình truyền dẫn dữ liệu tuần hoàn thì các cơ chế thích ứng khác nhau rất được coi trọng vì năng lượng được xả một cách liên tục và ta có thể làm giảm tốc độ xả.

Một nhiệm vụ rất được quan tâm và có vai trò quan trọng là tìm các mô hình phù hợp cho chi phí năng lượng của lớp liên kết, có thể chú ý đến vai trò của lớp MAC và lớp vật lý.

CÂU HỎI ÔN TẬP CHƯƠNG 6

1. Nhiệm vụ và các yêu cầu cơ bản của giao thức lớp liên kết?
2. Các đặc tính cơ bản của việc giám sát lỗi trong các giao thức lớp liên kết?
3. Nguyên nhân và các đặc tính của sai số truyền dẫn?
4. Các thành phần chính của giao thức ARQ?
5. Các giao thức ARQ chuẩn?
6. Kỹ thuật FEC?
7. Cách thức điều khiển công suất và lỗi trong các giao thức lớp liên kết?
8. Sơ đồ kiểm tra tổng trung gian?
9. Kết hợp FEC và tối ưu kích thước gói tin?
10. Nêu một số đặc điểm của chất lượng tuyến trong các giao thức lớp liên kết?
11. Cách thức ước lượng chất lượng tuyến?

CHƯƠNG 7. ĐẶT TÊN VÀ ĐÁNH ĐỊA CHỈ

Mục đích của chương

Các kế hoạch đặt tên và đánh địa chỉ được sử dụng để biểu thị và tìm kiếm các vật. Trong mạng, các tên và địa chỉ thường liên quan đến cá nhân các nút cũng như liên quan đến các dữ liệu lưu trong các nút. Các địa chỉ/các tên luôn có quan hệ chặt chẽ với một sự biểu diễn, sự biểu diễn này có một chiều dài nào đó khi nó được xem như một chuỗi bit. Không giống như các loại mạng khác, trong các mạng cảm biến không dây thì kích thước biểu diễn là một vấn đề cốt yếu, bởi vì các địa chỉ hiện diện trong hầu hết các gói. Tuy nhiên, việc kết hợp giữa các nút là cần thiết để chỉ định các địa chỉ ngắn một cách hợp lý. Một khía cạnh chủ chốt thứ hai là đánh địa chỉ dựa trên nội dung, ở đó không phải nút hay mạng giao tiếp mà chính dữ liệu sẽ được đánh địa chỉ. Việc đánh địa chỉ dựa trên nội dung có thể được tích hợp với việc định tuyến dữ liệu trung tâm và nó cũng là một khả năng chủ chốt của việc xử lý trong mạng.

Đặt tên và đánh địa chỉ là hai vấn đề cơ bản trong mạng. Có thể nói một cách mộc mạc rằng **các tên** được sử dụng để biểu thị các vật (ví dụ: các nút, dữ liệu, sự chuyển giao) trong đó **các địa chỉ** cung cấp thông tin cần thiết để tìm các vật này; ví dụ: chúng giúp cho việc định tuyến trong một mạng đa bước nhảy (multihop). Sự khác biệt này là không lớn; đôi khi các địa chỉ cũng được sử dụng để biểu thị các vật – một địa chỉ IP chứa thông tin để vừa tìm kiếm một nút (phần mạng của một địa chỉ) và để nhận dạng nút – một cách chính xác hơn: một giao tiếp mạng trong một nút – trong một mạng con đơn lẻ (phần máy chủ). Trong các mạng truyền thống như mạng Internet hoặc các mạng Ad hoc, **các nút** hay **các trạm** thường xuyên độc lập cũng như dữ liệu được đăng ký bởi các mạng đó được đặt tên và đánh địa chỉ. Điều này phù hợp với việc sử dụng như dự định của các mạng này: chúng kết nối nhiều người sử dụng và để những người sử dụng này trao đổi dữ liệu hoặc truy cập vào các máy chủ. Phạm vi của các loại dữ liệu người dùng có thể sử dụng là rất rộng và mạng có thể hỗ trợ các nhiệm vụ này tốt nhất bằng cách thực hiện các giả thuyết yếu nhất về dữ liệu – tất cả dữ liệu chỉ là một đồng bit được di chuyển từ một nút này đến một nút khác. Trong các mạng không dây, các nút là không độc lập nhưng chúng phối hợp với nhau để giải quyết một nhiệm vụ đã cho và để cung cấp cho người dùng một giao diện với thế giới bên ngoài. Do đó, nó có thể phù hợp với việc thay chuyển quan điểm từ đặt tên các nút sang đặt tên các khía cạnh của thế giới vật lý hoặc **đặt tên dữ liệu**.

Vấn đề đặt tên và đánh địa chỉ thường được tích hợp chặt chẽ với các phần của một ngăn xếp giao thức sử dụng chúng. Ví dụ: các giao thức giải pháp địa chỉ hay định tuyến. Trong chương này chúng ta tập trung vào các khía cạnh như sự phân phối địa chỉ, biểu diễn địa chỉ, và sử dụng chính xác các kiểu đặt tên/đánh địa chỉ khác nhau trong các mạng cảm biến không dây.

7.1 Các nguyên tắc cơ bản

7.1.1 Sử dụng các địa chỉ và các tên trong các mạng (cảm biến)

Trong phần lớn các mạng máy tính và mạng cảm biến, các loại tên, địa chỉ, và nhận dạng sau đây có thể được tìm thấy:

Nhận dạng nút duy nhất: một bộ nhận dạng nút duy nhất (UID - unique node identifier) là một dạng dữ liệu liên tục duy nhất cho tất cả các nút. Ví dụ: một UID có thể là một sự kết hợp của tên nhà cung cấp, tên sản phẩm, và số seri, được xác định vào thời điểm sản xuất. Một UID như vậy có thể hoặc không thể có chức năng nào trong ngăn xếp giao thức.

Địa chỉ MAC: một địa chỉ MAC được dùng để phân biệt giữa các nút lân cận cách một bước nhảy của một nút. Điều này là rất quan trọng trong các mạng cảm biến không dây sử dụng các giao thức MAC dựa trên sự tranh chấp, bởi vì bằng cách bao gồm một địa chỉ MAC trong các gói MAC đơn tính một nút có thể xác định gói tin nào không được nhắm tới đích là nó và nút đi vào chế độ ngủ trong khi gói tin đó đi qua. **Việc tránh nghe lén** là một phương pháp quan trọng để bảo tồn năng lượng tại lớp MAC (Chương 5).

Địa chỉ mạng: Một **địa chỉ mạng** được dùng để tìm kiếm và biểu diễn một nút qua nhiều bước nhảy (multi hops) và do đó các địa chỉ mạng thường được kết nối để định tuyến.

Các nhận dạng mạng: Trong các mạng (cảm biến) không dây có chồng phủ về mặt địa lý cùng một loại và làm việc ở cùng một băng tần, thì **các nhận dạng** (identifiers) về mạng là cái rất quan trọng để phân biệt giữa các mạng. Ví dụ: trong các mạng cảm biến ở lĩnh vực y tế thì các bệnh nhân trong cùng một phòng phải được phân biệt với nhau để ngăn ngừa sự nhầm lẫn các dữ liệu cảm biến thuộc về các bệnh nhân khác nhau.

Các nhận dạng tài nguyên: một cái tên hay nhận dạng tài nguyên được biểu diễn dưới dạng các thuật ngữ mà người sử dụng có thể hiểu được theo cách “có ý nghĩa nào đó” đối với người sử dụng. Ví dụ: dựa trên việc đọc tên www.xemacs.org, một người sử dụng có kinh nghiệm sẽ biết rằng (i) cái mà tên gọi này đề cập đến giống như một máy chủ web và (ii) người sử dụng có thể tìm thông tin về một trình soạn thảo văn bản lớn. Ngược lại, khi xem địa chỉ IP 199.184.165.136, hầu như không người sử dụng nào có thể rút ra được kết luận gì. Các tên có thể chỉ dẫn tới các nút, các nhóm nút, các mục dữ liệu, hoặc các khái niệm trừu tượng tương đương.

Một nút đơn lẻ có thể có nhiều tên và nhiều địa chỉ. Ví dụ: máy chủ WWW www.xemacs.org có tên www.xemacs.org, nó có địa chỉ IP 199.184.165.136 và giả thiết rằng máy chủ được gắn với một mạng Ethernet, nó có một địa chỉ IEEE MAC 48 bit. Việc ánh xạ giữa các tên người sử dụng thân thiện giống như www.xemacs.org và các địa chỉ liên quan đến hoạt động của mạng được tiến hành bởi **các dịch vụ ràng buộc** (binding services). Việc ánh xạ này thường được đề cập đến như là **giải pháp đặt tên**. Trong ví dụ của chúng ta, dịch vụ tên miền (DNS - domain name service) cung cấp việc ánh xạ từ tên tới địa chỉ IP trong khi giao thức giải pháp địa chỉ (ARP - address resolution protocol) ánh xạ địa chỉ IP tới một địa chỉ MAC.

7.1.2 Nhiệm vụ quản lý địa chỉ

Chúng ta tóm tắt các nhiệm vụ cơ bản của việc quản lý địa chỉ. Các nhiệm vụ cơ bản này là độc lập với các loại địa chỉ:

Phân phối địa chỉ: nói chung, việc phân phối địa chỉ ấn định một địa chỉ cho một thực thể từ một kho địa chỉ.

Không phân phối địa chỉ: các kiểu đánh địa chỉ theo yêu cầu, không gian địa chỉ thường có kích thước từ nhỏ tới trung bình. Số lượng của nút trong các mạng cảm biến thực chất là không cố định, có cả các nút chết hoặc di chuyển ra ngoài và các nút mới được thêm

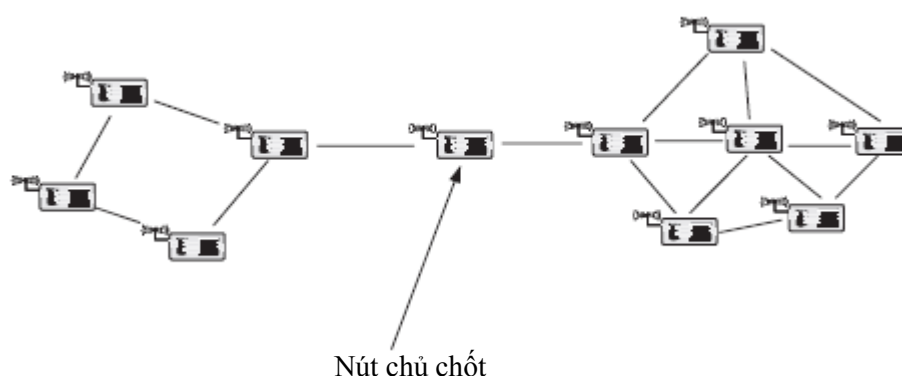
vào mạng. Nếu các địa chỉ của các nút chuyển đi không được gửi trả lại kho địa chỉ để **tái sử dụng** thì kho địa chỉ cuối cùng sẽ bị cạn kiệt và không một địa chỉ nào có thể được phân phối cho các nút mới. Việc không phân phối địa chỉ có thể hoặc là có chủ ý hoặc đột ngột. Trong việc **không phân phối địa chỉ có chủ ý**, một nút gửi đi các gói tin điều khiển một cách dứt khoát (rõ ràng) để từ bỏ địa chỉ của nó. Trong việc **không phân phối địa chỉ đột ngột**, nút này biến mất hoặc lên vào mạng và do đó không gửi đi các gói tin điều khiển phù hợp, để lại trách nhiệm phát hiện và không phân phối địa chỉ của nút cho mạng. Khi các không gian địa chỉ rất rộng được sử dụng, chẳng hạn như các địa chỉ MAC IEEE 802.3 có chiều dài 48 bit, thì việc không phân phối địa chỉ không phải là một vấn đề.

Biểu diễn địa chỉ: một định dạng cho việc biểu diễn các địa chỉ cần phải được dàn xếp và thực hiện.

Phát hiện/ giải quyết xung đột: các xung đột địa chỉ có thể xảy ra trong các mạng với nhiệm vụ phân phối các địa chỉ theo yêu cầu hoặc trong trường hợp liên kết các mạng khác nhau. Nếu xung đột không thể chịu đựng được, người ta buộc phải giải quyết chúng.

Ràng buộc: nếu vài lớp địa chỉ được sử dụng, cần phải cung cấp một ánh xạ giữa các lớp khác nhau. Ví dụ: trong các mạng IP, một địa chỉ IP phải được ánh xạ đến một địa chỉ MAC sử dụng giao thức ARP.

Bất cứ kiểu quản lý địa chỉ nào cho các mạng cảm biến và ad hoc đôi khi phải đối mặt với các sự kiện **phân chia của mạng** và các sự kiện **liên kết mạng**. Chẳng hạn như mạng được cho ở hình 7.1. Nếu một nút chủ chốt (critical node) bị hết năng lượng, thì mạng được chia thành hai phần mà không có kết nối nào với nhau nữa. Vấn đề chủ chốt ở đây là việc không phân phối địa chỉ. Cả hai mạng con sẽ phát hiện ra rằng có vài nút mạng mà chúng không thể vươn tới được nữa và các địa chỉ của chúng sẽ được phục hồi. Mặt khác, việc tái phân phối các địa chỉ được phục hồi sẽ không xảy ra quá nhanh. Nếu vì lý do nào đó mạng được liên kết lại thì việc phân phối cùng địa chỉ như trước khi việc chia mạng xảy ra và đảm bảo không có xung đột địa chỉ nào là những vấn đề cần được giải quyết.



Hình 7.1 Ví dụ cho sự phân chia của mạng

7.1.3 Sự duy nhất của các địa chỉ

Chúng ta có thể phân biệt các yêu cầu duy nhất sau đây đối với các tên và các địa chỉ mạng.

Duy nhất toàn cầu: một địa chỉ hoặc nhận dạng duy nhất toàn cầu được cho là xuất hiện gần như là một lần trên toàn thế giới. Ví dụ: các địa chỉ MAC IEEE 48 bit dùng trong

mạng Ethernet và Token Ring. Biểu diễn nhị phân của các địa chỉ này phải đủ lớn để cung cấp cho tất cả các thiết bị toàn trên thế giới.

Duy nhất toàn mạng: một **địa chỉ duy nhất toàn mạng** được cho là duy nhất trong một mạng đã cho, nhưng địa chỉ tương tự có thể được sử dụng trong các mạng khác. Bằng việc có các mạng khác nhau A và B, chúng ta ngụ ý rằng không có cặp nút nào thỏa mãn $a \in A$ và $b \in B$ mà có thể truyền thông được với nhau.

Duy nhất ở địa phương: một **địa chỉ duy nhất ở địa phương** có thể xuất hiện vài lần trong cùng một mạng, nhưng nó phải là duy nhất trong một **vùng lân cận** được xác định phù hợp. Để minh họa điều này:

- Đối với các địa chỉ MAC, nó là hợp lý để yêu cầu rằng chúng là duy nhất chỉ trong vùng lân cận hai bước nhảy. Vấn đề xảy ra liên quan đến yêu cầu này được trình bày trong hình 7.2: nút C là một nút lân cận một bước nhảy của nút B và là một nút lân cận hai bước nhảy của nút A. Nếu nút A và nút C có cùng các địa chỉ MAC, thì nút B không thể luận ra một gói tin đang tới là từ máy phát nào, hoặc nút B cũng không thể truyền trực tiếp các gói của nó tới một máy thu dự kiến duy nhất.
- Một ví dụ khác được cho bởi một mạng cảm biến với các loại cảm biến khác nhau như các cảm biến nhiệt độ, độ ẩm, và ánh sáng. Chúng ta có thể yêu cầu rằng không có 2 cảm biến nhiệt độ nào có cùng địa chỉ nhưng một cảm biến nhiệt độ và một cảm biến độ ẩm thì có thể. Trong trường hợp này, vùng lân cận được tạo thành bởi các cảm biến của cùng một loại.

7.1.4 Phân phối và ấn định địa chỉ

Việc ấn định địa chỉ có thể nảy sinh một **ưu tiên**, (ví dụ: trong quá trình sản xuất hoặc trước khi triển khai mạng) hoặc **theo yêu cầu**, bằng cách sử dụng một giao thức ấn định địa chỉ. Giao thức ấn định địa chỉ theo yêu cầu này có thể hoặc là **tập trung** hoặc là **phân tán**. Trong một giải pháp tập trung, có một nút/người có quyền quản lý (các phần của) kho địa chỉ, trong khi đó ở các giải pháp phân tán, không có nút nào thể hiện như vậy. Thay vào đó, tất cả các nút tiềm năng thực hiện cùng một vai trò trong việc ấn định địa chỉ. Việc giải phóng/không phân phối địa chỉ thực hiện một vai trò quan trọng khi các địa chỉ duy nhất toàn mạng hoặc cục bộ được ấn định theo yêu cầu.

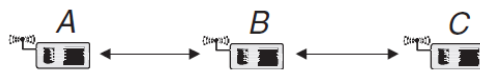
Trong việc ấn định địa chỉ phân tán, không thể luôn luôn cho phép việc đảm bảo là duy nhất trong toàn mạng tại tất cả các thời điểm. Một người có thể hoặc là quyết định đơn giản bằng việc sống chung với vài xung đột địa chỉ hoặc là phát hiện và **giải quyết** chúng. Trong trường hợp thứ hai, người ta sự khác biệt giữa Việc phát hiện địa chỉ đúp (DAD: Duplicate Address Detection) mạnh và yếu:

- Ở DAD mạnh, người ta yêu cầu rằng nếu địa chỉ x đã được phân phối cho nút A tại thời điểm t_0 và sau đó lại được cấp phát cho nút B tại thời điểm t_1 , thì sau đó việc cấp phát địa chỉ đúp này phải được phát hiện muộn nhất là tại thời điểm t_1+T trong đó T là một giới hạn thời gian cố định nào đó.
- Trong DAD yếu, các địa chỉ đúp được chịu đựng miễn là chúng không gây nhiễu cho các phiên tiếp theo. Ví dụ: nếu hai mạng A và B kết hợp với nhau và một địa

chỉ x được cấp phát trong cả hai mạng, không có hoạt động nào xảy ra chừng nào mà tất cả các gói tin từ các nút của mạng cũ A được nhắm tới x , đi tới nút trong mạng A với địa chỉ x và không phải là nút với cùng một địa chỉ trong mạng khác.

Một ví dụ cho kiểu cấp phát địa chỉ tập trung là giao thức DHCP (Dynamic Host Configuration Protocol) được biết đến từ thế giới Internet. Tuy nhiên, có một số vấn đề với các kiểu tập trung này:

- Các giải pháp tập trung không phù hợp tốt với các mạng cảm biến. Số lượng tuyệt đối của các nút tạo ra lưu lượng đáng kể, lưu lượng này được hướng trực tiếp tới một hoặc vài máy chủ (server) địa chỉ và khu vực xung quanh máy chủ trở thành một điểm nóng. Điều này có thể được lợi dụng cho một số mở rộng bằng cách triển khai các kỹ thuật dựa trên các phân nhóm, trong các kỹ thuật đó việc chia sẻ tách rời của không gian địa chỉ được phân phối tới các chỉ huy phân nhóm, các chỉ huy phân nhóm này lại lần lượt cấp phát các địa chỉ này cho các thành viên của phân nhóm.
- Nếu mạng đã bị chia tách trước khi một nút mới gia nhập vào, thì máy chủ địa chỉ trung tâm có thể không tiếp cận được.
- Giao thức DHCP yêu cầu các nút làm mới các địa chỉ của chúng theo chu kỳ để phát hiện sự không phân phối địa chỉ đột ngột.



Hình 7.2 Ví dụ cho sự phân chia của mạng

7.1.5 Đánh địa chỉ phân tiếp đầu

Một trong các khía cạnh quan trọng nhất của các địa chỉ là số lượng các bit cần thiết cho việc biểu diễn chúng hoặc phân tiếp đầu (overhead) của chúng. Phân tiếp đầu này và do đó năng lượng cần thiết để truyền thông tin địa chỉ liên quan đến hai yếu tố: (i) Tần số mà với tần số này các địa chỉ được sử dụng và (ii) kích thước của việc biểu diễn chúng. Chúng ta xem ví dụ về các địa chỉ MAC. Có một số giao thức MAC như TRAMA (phần 5.3.3) hoặc SMACS (phần 5.3.2) thiết lập các đường kết nối dành riêng giữa hai nút cạnh nhau bằng cách ấn định các khe thời gian hoặc các tần số không có xung đột. Nếu một đường kết nối như vậy được sử dụng cho một gói dữ liệu, thì không cần thiết phải mang thông tin địa chỉ trong các gói này bởi vì các nút nguồn và nút đích là hoàn toàn đã cho.

Ngược lại, trong các giao thức MAC dựa trên sự tranh chấp (ganh đua), tại một thời điểm đã cho bất cứ nút nào cũng có thể truyền tới các nút khác và thông tin đánh địa chỉ do đó là yếu tố sống còn để nhận dạng nguồn và đích và để tránh nghe lén. Vì vậy, càng ít bit dùng cho một địa chỉ thì càng tốt. Chúng ta hãy nhìn vào sự không dung hòa (trade-offs) liên quan ở đây:

- Chẳng hạn, chúng ta chọn một địa chỉ duy nhất toàn cầu được ấn định ưu tiên như trong Ethernet IEEE 802.3. Ở đó, 48 bit được sử dụng để cung cấp số lượng thiết bị hiện tại và đã biết trước. Trong các mạng Ethernet các khung tương đối lớn có kích thước vài trăm byte được sử dụng, nên các địa chỉ 6-byte là một phân tiếp đầu

không đáng kể, và một cấp phát địa chỉ ưu tiên sẽ loại bỏ việc cần đến một giao thức cấp phát địa chỉ. Mặt khác, trong các mạng cảm biến không dây sẽ có nhiều gói dữ liệu nhỏ và vì thế một địa chỉ đơn 48 bit có thể sẽ lớn hơn dữ liệu.

- Một địa chỉ duy nhất toàn mạng phải có một số lượng các bit đủ để cung cấp cho tất cả các nút trong mạng. Trong một mạng cảm biến với 10.000 nút, một địa chỉ gồm 14 bit là đủ. Tuy nhiên, để tối thiểu hóa số lượng của các bit địa chỉ, kích thước của mạng phải được biết trước. Một số dự trữ an toàn độ rộng trường địa chỉ là quan trọng nếu xảy ra việc triển khai đa pha, nghĩa là nếu các nút thêm vào mạng được triển khai lâu sau khi mạng đã sẵn sàng cho việc sử dụng lần đầu tiên, ví dụ: để thay thế các nút có dự trữ năng lượng đã cạn kiệt.
- Một địa chỉ duy nhất cục bộ phải là duy nhất trong một vùng lân cận nhất định, vùng này thường nhỏ hơn rất nhiều so với toàn mạng. Ví dụ: các địa chỉ MAC phải là duy nhất trong một vùng hai bước nhảy, vùng này có thể chứa vài tá nút, phụ thuộc vào mật độ nút. Do đó, các địa chỉ có thể sử dụng ít bit hơn đáng kể nó phải cần cho các địa chỉ duy nhất toàn mạng. Nói cách khác, bởi vì cấu trúc liên kết mạng (topology) chính xác rất ít khi được biết trước, nên một giao thức cấp phát địa chỉ là cần thiết. Việc có ít bit địa chỉ hơn là điều quan trọng nếu số lượng của các bit dữ liệu trong một gói là nhỏ.

Có một sự không dung hòa (trade-off) ở đây, đó là việc sử dụng các địa chỉ MAC cục bộ ngắn hơn có thể tiết kiệm năng lượng đáng kể trong trường hợp các gói dữ liệu nhỏ, nhưng yêu cầu phần tiếp đầu là ở dưới dạng các giao thức cấp phát/dàn xếp địa chỉ. Trong các mạng cảm biến với các nút phần lớn là không chuyển động, một giao thức như vậy là cần thiết để chạy tại thời điểm bắt đầu và đôi khi muộn hơn để xử lý các nút mới và các nút đã bị xóa. Trong trường hợp này, những lợi ích từ việc tiết kiệm các bit địa chỉ trong mỗi gói dữ liệu có thể có nhiều tác dụng hơn đối với giá thành của giao thức dàn xếp. Mặt khác, trong các mạng ad hoc và mạng cảm biến có khả năng di động cao, giao thức dàn xếp cần phải chạy thường xuyên nhằm mục đích tiết kiệm năng lượng.

7.2 Quản lý tên và địa chỉ trong các mạng cảm biến không dây

Chúng ta thấy rằng các địa chỉ MAC là không thể thiếu được nếu giao thức MAC sử dụng việc tránh nghe lén và chuyển sang chế độ ngủ (sleep mode) một cách thường xuyên có thể. Tuy nhiên, các địa chỉ MAC có cần phải duy nhất toàn mạng hay duy nhất toàn cầu không? Câu trả lời là “Không”, bởi vì phạm vi của một giao thức MAC là truyền thông giữa các nút gần nhau (xem phần 7.1.3). Yêu cầu này đảm bảo rằng không có hai nút gần nhau nào của một nút đã được lựa chọn có cùng địa chỉ MAC. Như đã thảo luận ở trên, các địa chỉ duy nhất cục bộ có khả năng là ngắn nhưng cần một giao thức cấp phát địa chỉ. Vấn đề này được đề cập đến trong phần 7.3. Làm cách nào mà các địa chỉ lớp cao, đặc biệt là các địa chỉ lớp mạng, mà dùng cho các giao thức định tuyến phải là duy nhất toàn mạng hoặc duy nhất toàn cầu? Chúng ta sẽ thảo luận ngắn gọn rằng việc thỏa mãn yêu cầu này là một nhiệm vụ cực khó. Chúng ta cũng tranh luận rằng yêu cầu này là không thực sự cần thiết trong các mạng cảm biến không dây bởi vì cuối cùng thì toàn mạng không phải là một bộ sưu tập của các nút thuộc những người sử dụng cá nhân mà các nút kết hợp với nhau để xử lý các tín hiệu và các sự kiện từ môi trường vật lý. Sự tranh cãi chính đó là người sử dụng rất cực quan tâm đến dữ

liệu chứ không quan tâm đến cá nhân các nút hay các nhóm nút mang dữ liệu đến. Đi xa hơn về vấn đề này, thì dữ liệu cũng có khả năng ảnh hưởng đến hoạt động của các giao thức, mà các giao thức lại là cốt lõi của mạng dữ liệu tập trung. Các kiểu đánh địa chỉ dựa trên nội dung hoặc đánh địa chỉ tập trung vì thế rất quan trọng và sẽ được thảo luận trong phần 7.4.

7.3 Ấn định các địa chỉ MAC

Phần này thảo luận về các phương pháp ấn định địa chỉ cho các địa chỉ MAC. Như đã được thảo luận trong phần 7.1.5, việc ấn định của các địa chỉ MAC duy nhất toàn cầu là không thể mong đợi trong các mạng cảm biến với các gói tin phân lớn là nhỏ.

Một sự ấn định ưu tiên của các địa chỉ duy nhất toàn mạng là khả thi chỉ khi nó có thể được thực hiện với nỗ lực phù hợp. Nhưng vẫn có vấn đề là phân tiếp đầu để biểu diễn các địa chỉ có thể là đáng kể mặc dầu nó không dài như trong các địa chỉ duy nhất toàn cầu. Ví dụ: lên đến 16.384 nút có thể được đánh địa chỉ với 14 bit và con số này tiện lợi hơn nhiều so với 48 bit được sử dụng cho các địa chỉ IEEE duy nhất toàn cầu.

Do đó, chúng ta tập trung vào sự ấn định động và phân tán cho các địa chỉ cục bộ và các địa chỉ toàn mạng. Các giao thức được thảo luận trong phần này khác về số lượng và phạm vi của sự cộng tác so với các nút khác.

Việc ấn định phân tán của các địa chỉ toàn mạng

Một cách tiếp cận đơn giản: một nút chọn ngẫu nhiên một địa chỉ từ một phạm vi địa chỉ đã cho và hy vọng là địa chỉ đó là duy nhất. Để đơn giản việc giải thích, chúng ta giả thiết rằng phạm vi địa chỉ này được cho bởi các số nguyên trong phạm vi 0 đến $2m - 1$ và một địa chỉ do đó có thể được biểu diễn với m bit. Không gian địa chỉ có một kích thước là $n = 2m$ địa chỉ.

Một nút lựa chọn địa chỉ của nó không cần đến bất kỳ thông tin ưu tiên nào, trong trường hợp này tốt nhất là sử dụng một phân bố đồng đều trên phạm vi địa chỉ bởi vì sự phân bố này có entropy cực đại. Tuy nhiên, cách tiếp cận này không tránh khỏi các vấn đề xảy ra, như được trình bày trong ví dụ dưới đây.

Ví dụ 7.1 (Việc ấn định địa chỉ ngẫu nhiên) Giả thiết rằng chúng ta có k nút và mỗi nút trong số các nút này chọn một địa chỉ ngẫu nhiên một cách độc lập và cùng kiểu từ 0 đến $2m - 1$. Xác suất các nút này chọn một sự ấn định không có xung đột là bằng bao nhiêu? Một vấn đề tương tự được biết đến với tên gọi “vấn đề sinh nhật” (Trong một phòng có n người, xác suất không có hai người nào trong số họ có cùng ngày sinh nhật là bằng bao nhiêu?) và có thể được trả lời bởi các đối số tổ hợp.

Đối với $k = 1$, xác suất này là 1. Đối với $k = 2$, nút thứ hai chọn với xác suất $\frac{n-1}{n}$ một địa chỉ

khác với lựa chọn của nút đầu tiên. Đối với $k = 3$, nút thứ ba chọn với xác suất $\frac{(n-1)(n-2)}{n^2}$

một địa chỉ khác với lựa chọn của hai nút đầu tiên và cứ tiếp tục như vậy. Suy ra, chúng ta có xác suất $P(n, k)$ để tìm một sự ấn định không có xung đột:

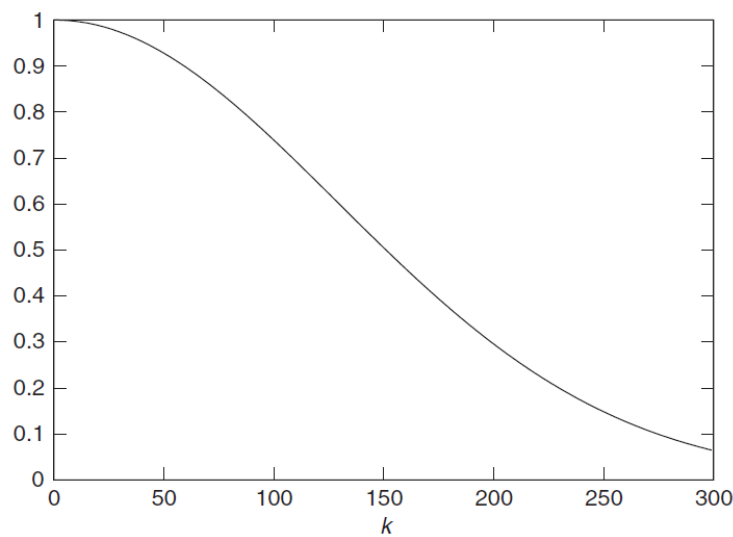
$$P(n, k) = 1 \cdot \frac{n-1}{n} \cdots \frac{n-k+1}{n} = \frac{1}{n^k} \cdot \frac{n!}{(n-k)!} = \frac{k!}{n^k} \binom{n}{k}$$

mà theo xấp xỉ Stirlings ($n! \approx \sqrt{2\pi} \cdot n^{n+1/2} \cdot e^{-n}$ [255, Chương II]), được cho xấp xỉ bằng:

$$P(n, k) = e^{-k} \cdot \left(\frac{n}{n-k} \right)^{(n-k)+1/2}$$

Đối với một trường địa chỉ có kích thước $m = 14$ bit, tương đương với $n = 2^{14} = 16.384$ địa chỉ khác nhau, chúng ta chỉ ra trong hình 7.3 rằng xác suất $P(n, k)$ cho các giá trị k khác nhau. Đối với các giá trị k khá nhỏ, xác suất của các xung đột trở nên gần bằng 1. Ví dụ: với $k = 275$ xác suất của các xung đột là lớn hơn 90% nhưng chỉ $\approx 1,7\%$ không gian địa chỉ là được sử dụng.

Do đó, phương pháp ấn định địa chỉ ngẫu nhiên này nhanh chóng dẫn đến các xung đột địa chỉ. Để bảo vệ sự duy nhất toàn mạng, hoặc là cần một giao thức giải quyết xung đột hoặc cần lựa chọn các phương pháp ấn định địa chỉ thông minh hơn.



Hình 7.3 “Xác suất sinh nhật” mà k ngoài $n = 2^{14}$ trạm lựa chọn các địa chỉ ngẫu nhiên không có các xung đột

Có cách nào tốt hơn không? Một nút có thể cố gắng thu nhận thông tin về các địa chỉ đã được phân phối bằng cách nghe lén các gói tin trong vùng phụ cận của nó và cố gắng tránh các địa chỉ này. Trong nhiều ứng dụng của mạng cảm biến, ở đó các nút truyền dữ liệu cảm biến của chúng tới một điều phối viên cục bộ tập hợp và xử lý dữ liệu, việc nghe lén có thể tránh được nhiều xung đột với các nút cục bộ khác đang truyền tin tới cùng một điều phối viên.

Với việc ấn định địa chỉ ngẫu nhiên chúng ta đối mặt với các xung đột địa chỉ với xác suất cao. Chúng ta giải quyết vấn đề này như thế nào? Giải pháp đầu tiên đơn giản là chấp nhận chúng và chẳng làm gì cả. Các kỹ thuật khác được nghiên cứu trong tình huống của sự ấn định địa chỉ IP trong các mạng MANET (Mobile Ad Hoc Network):

- Một giao thức tự động cấu hình địa chỉ phù hợp cho các mạng MANET được biểu diễn như sau: Một nút bắt đầu bằng cách lựa chọn một cách ngẫu nhiên một địa chỉ tạm thời và một địa chỉ cố định được đề xuất và gửi đi một gói tin điều khiển **yêu cầu địa chỉ**, mang các địa chỉ cố định và địa chỉ tạm thời đã chọn. Các địa chỉ tạm thời được cấp phát từ một kho địa chỉ dành riêng, nó tách biệt khỏi kho các địa chỉ thực sự của nút. Giao thức định tuyến cơ bản cố gắng tìm ra một đường tới một nút

có cùng địa chỉ cố định. Nếu tồn tại một nút như vậy (và một đường tới nút đó) thì một gói tin **trả lời địa chỉ** được tạo ra và gửi tới địa chỉ tạm thời. Khi nhận được trả lời này, nút biết rằng địa chỉ cố định nó chọn đã được cấp phát và cố tìm địa chỉ khác. Nếu không nhận được gói tin trả lời địa chỉ trong một thời gian nhất định thì nút lặp lại gói tin yêu cầu địa chỉ một số lần để bù lại các gói tin trả lời địa chỉ có thể bị mất. Nếu vẫn không nhận được gói tin trả lời địa chỉ sau tất cả các cố gắng hết sức, nút chấp nhận địa chỉ IP đã chọn. Người ta chứng minh rằng giao thức này không dùng được nếu các trễ không được giới hạn, ví dụ: sau khi phân chia mạng. Nếu trong các mạng cảm biến, phương pháp này được áp dụng cho các địa chỉ MAC thay vì các địa mạng, thì sau đó các nút khác không có bất cứ thông tin định tuyến nào, và các gói tin yêu cầu/trả lời địa chỉ phải được tràn vào trong mạng.

- Nesargi và Prakash [21] quan tâm đến vấn đề ấn định địa chỉ như một vấn đề thỏa thuận phân tán, một vấn đề nổi tiếng trong các hệ thống phân tán. Một nút yêu cầu (**kẻ yêu cầu**) liên hệ với một nút lân cận mà đã có một địa chỉ - **kẻ khởi đầu**. Kẻ khởi đầu giữ một bảng tất cả các ấn định địa chỉ đã biết và chọn một địa chỉ chưa sử dụng. Kẻ khởi đầu sau đó phổ biến địa chỉ mới được đề xuất cho tất cả các nút trong mạng và thu thập các câu trả lời. Tất cả các nút đặt địa chỉ đề xuất vào trong một danh sách chứa các địa chỉ ứng cử. Nếu một nút tìm địa chỉ hoặc là trong danh sách ứng cử hoặc trong danh sách cục bộ của nó về các ấn định đã biết, nó trả lời với một gói tin từ chối, nếu không thì nó trả lời với một gói tin chấp nhận. Nếu tất cả các nút đã biết trả lời bằng một gói tin chấp nhận, kẻ khởi đầu ấn định địa chỉ cho kẻ yêu cầu và thông báo cho tất cả các nút khác trong mạng rằng việc ấn định bây giờ là cố định. Nếu không thì kẻ khởi đầu lựa chọn địa chỉ khác và thử lại lần nữa. Phương pháp này tương tự với một giao thức cam kết hai pha và rõ ràng là tạo ra quá nhiều phần tiếp đầu (overhead) trong các gói tin được truyền đi và các yêu cầu về không gian bộ đệm phải là khả thi trong các mạng cảm biến không dây.
- Một thuật toán tự động cấu hình địa chỉ phân tầng đối với các địa chỉ IPv6 dự kiến cho các mạng MANET được trình bày như sau: Một số nút trong mạng trở thành các nút lãnh đạo và lựa chọn một ID mạng con một cách ngẫu nhiên. Một DAD được thực hiện giữa các nút lãnh đạo để đảm bảo sự duy nhất của các ID mạng con. Các nút khác tạo ra các địa chỉ của chúng từ ID mạng con của nút lãnh đạo của chúng và một địa chỉ cục bộ (ví dụ: dựa trên địa chỉ MAC của nút). Một nút lãnh đạo có thể được bầu cho mỗi phần của mạng, ấn định các địa chỉ cho các nút mới đến. Các liên kết của các mạng được phát hiện bởi việc giới thiệu các nhận dạng mạng tách biệt.

Giá thành dưới dạng tổng chi phí truyền thông sẽ được trả dựa trên một thử nghiệm ấn định địa chỉ, ví dụ: khi mạng phải được làm tràn bởi vì nút yêu cầu không có địa chỉ có thể định tuyến được. Như một sự lựa chọn, nếu một giao thức định tuyến tiên phong được sử dụng, các nút sở hữu các bảng chứa các địa chỉ được sử dụng, các địa chỉ mà có thể được tra cứu một cách nhanh chóng hoặc có thể suy ra sự biểu diễn của các nút đúp bởi vì các bản tin định tuyến giả nhận được mang địa chỉ nguồn của nút. Tuy nhiên, các giao thức định tuyến theo yêu cầu là phổ biến hơn trong các mạng cảm biến so với các giao thức định tuyến thực hiện tiên phong là do tổng chi phí của chúng.

7.4. Đánh địa chỉ theo địa lý và dựa trên nội dung

Các mạng ad hoc và mạng cố định truyền thống cung cấp các dịch vụ và các giao thức, chúng cho phép một số lượng lớn những người sử dụng độc lập trao đổi dữ liệu với nhau và với phần còn lại của thế giới. Mặt khác, trong các mạng cảm biến không dây, các nút tương tác với môi trường vật lý, và chúng kết hợp với nhau, nghĩa là chúng không độc lập với nhau. Một người sử dụng của một mạng cảm biến không dây rất cục muốn biết đôi điều về môi trường vật lý mà mạng tương tác với, nhưng anh ta thường không quan tâm đến các nút cảm biến cá nhân. Ví dụ: một người sử dụng muốn hỏi “hãy cho tôi biết nhiệt độ trong phòng 205 trong tòa nhà A3” thay vì “thu thập các giá trị nhiệt độ của nút cảm biến 13, 47, 2225, 14592, và 14593 và cho tôi giá trị trung bình của chúng”. Người ta thường thích cho phép người sử dụng đặt tên cho dữ liệu họ quan tâm đến chứ không phải là nút (hay tập hợp các nút) tạo ra dữ liệu. Trong các mạng dựa trên IP truyền thống, yêu cầu này tương ứng với việc giới thiệu một hệ thống tên gọi nằm trên đỉnh của các địa chỉ IP và để giới thiệu các dịch vụ ràng buộc phù hợp như DNS hoặc các dịch vụ thư mục khác, cung cấp một ánh xạ từ các tên (có ý nghĩa đối với người sử dụng) tới các địa chỉ IP (có ý nghĩa đối với giao thức định tuyến). Tuy nhiên, trong các mạng cảm biến các mức không thư mục này có thể bị loại bỏ và các thuộc tính định rõ người sử dụng có thể được dùng một cách trực tiếp để tìm kiếm (các nhóm của) các nút. Ý tưởng này thường được gọi là **đánh địa chỉ tập trung - data-centric addressing**. Phương pháp này làm cho các dữ liệu ứng dụng có ý nghĩa đối với hoạt động của các giao thức mạng (đặc biệt là định tuyến), đó cũng là một sự cho phép chủ yếu đối với các kỹ thuật xử lý trong mạng. Việc đánh địa chỉ địa lý có thể xem như một trường hợp đặc biệt của việc đánh địa chỉ dựa trên nội dung. Ở đây, một số các thuộc tính định rõ người sử dụng chỉ đến các tọa độ không gian. Đánh địa chỉ địa lý giả thuyết rằng mỗi nút biết vị trí của bản thân nó đối với một hệ thống tọa độ đã được thỏa thuận. Do đó, các kỹ thuật định vị (chương 9) là thiết yếu để làm việc với các địa chỉ địa lý. Mặt khác, các địa chỉ địa lý có thể giúp cho việc định tuyến. Cả hai cách đánh địa chỉ theo nội dung và theo địa lý đều không thay thế được các giao thức MAC nhưng chúng có thể sử dụng, ví dụ như trong lớp mạng để giúp đỡ các quyết định định tuyến.

7.4.1 Đánh địa chỉ dựa trên nội dung

Vài hệ thống đặt tên dựa theo nội dung hoặc dựa theo thuộc tính xuất hiện trong các tài liệu. Trước khi chuyển đến một thảo luận chung hơn, chúng ta biểu diễn một ví dụ về phương pháp đặt tên mà được phát triển trong khái niệm của các mạng cảm biến không dây.

Ví dụ 7.2 (Một cơ chế đặt tên mức thấp) trong phương pháp này, việc đánh địa chỉ dựa trên nội dung được tích hợp với định tuyến **truyền tin trực tiếp**. Trong một phạm vi nhỏ, trong truyền tin trực tiếp một nút thu phát ra một **bản tin lợi ích**, chỉ rõ một tập hợp các thuộc tính dùng để mô tả dữ liệu mong muốn. Bản tin này được phổ biến trong mạng. Các nút mà có thể tạo ra dữ liệu cảm biến phù hợp với lợi ích đó được gọi là **các nút nguồn**. Một gói dữ liệu được tạo ra bởi một nút nguồn sẽ di chuyển qua các nút trung gian để tới nút thu. Một nút trung gian lưu lại lợi ích này cùng với một (một tập) các nút lân cận theo hướng ngược lại có thể trong **cache lợi ích (interest cache)**. Trong lúc nhận một gói dữ liệu, nút trung gian tìm kiếm trong cache của nó một lợi ích phù hợp với dữ liệu và gửi tiếp gói dữ liệu này tới nút lân cận có liên kết theo hướng ngược lại.

Cả các lợi ích và các gói dữ liệu được biểu diễn như một tập hợp các bộ AVO (Attribute Value Operation – Toán tử giá trị thuộc tính). Tập hợp các thuộc tính được xác định trước và mỗi thuộc tính xử lý một khóa duy nhất, đã được biết rõ cũng như hiểu rõ loại dữ liệu của giá

trị tương ứng. Đối với các mạng cảm biến, việc biểu diễn của tất cả các thành phần cấu thành bộ AVO phải càng ngắn càng tốt để giảm số lượng các bit cần thiết. Các toán tử khác nhau và ý nghĩa của chúng được cho ở bảng 7.1. Toán tử IS chỉ ra rằng thuộc tính tương ứng thực sự có giá trị biểu thị và thường được nguồn dữ liệu tạo ra; IS cũng được gọi là một **toán tử thực (actual operator)**. Tất cả các toán tử khác được gọi là các toán tử chính thức (**formal operators**), và chúng được dùng để chỉ rõ các lợi ích đối lập với nó là các giá trị thực tế được tạo ra bởi nguồn phù hợp.

Các nút trung gian sử dụng một toán tử gọi là **phù hợp một chiều (one-way match)** (trình bày ở danh sách 7.1 trong chú giải mã giả (pseudocode)) để xác định một gói dữ liệu nhận được phù hợp với lợi ích nào. Cơ bản là đối với một lợi ích đã cho, nút trung gian đi qua tất cả các thuộc tính chính thức và kiểm tra xem liệu (i) gói dữ liệu xử lý toàn bộ một thuộc tính tương ứng và (ii) liệu giá trị thực tế mà thuộc tính của gói dữ liệu mang trong nó có phù hợp với toán tử chính thức trong lợi ích.

Chúng ta hãy xét một ví dụ sau đây. Giả thuyết rằng người sử dụng tại nút thu muốn biết khi nào nhiệt độ trong một vùng địa lý nào đó vượt quá một ngưỡng nào đó. Anh ta kết hợp vài thuộc tính trong một bản tin lợi ích như trình bày ở danh sách 7.2. Trong ví dụ này, loại thuộc tính chỉ rõ loại cảm biến mà lợi ích hướng tới (ở đây, đó là các cảm biến nhiệt độ). Thuộc tính tiếp theo là ngưỡng dưới chỉ rõ rằng máy thu được quan tâm trong các trường hợp tại đó ngưỡng 20⁰C bị vượt quá từ phía dưới. Khu vực nằm trong sự quan sát là một hình vuông giữa (0,0) và (20,20) (biểu diễn dưới dạng mét). Nếu nhiệt độ vượt quá ngưỡng này, một cảm biến phù hợp sẽ ghi lại sự kiện này cứ 0,05 s một lần trong một thời hạn là 10 s. Lớp thuộc tính cuối cùng biểu diễn rằng AVO tuple hiện thời là một lợi ích chứ không phải là một bản tin dữ liệu.

Tên toán tử	Ý nghĩa
EQ	Xảy ra nếu giá trị thực sự bằng với <i>giá trị</i>
NE	Xảy ra nếu giá trị thực sự không bằng với <i>giá trị</i>
LT	Xảy ra nếu giá trị thực sự nhỏ hơn <i>giá trị</i>
GT	Xảy ra nếu giá trị thực sự lớn hơn <i>giá trị</i>
LE	Xảy ra nếu giá trị thực sự nhỏ hơn hoặc bằng với <i>giá trị</i>
GE	Xảy ra nếu giá trị thực sự lớn hơn hoặc bằng với <i>giá trị</i>
EQ_ANY	Ứng với bất kể cái gì, <i>giá trị</i> là vô nghĩa
IS	Chỉ rõ một thuộc tính theo nghĩa đen

Bảng 7.1 Tập hợp các toán tử

Giả thuyết rằng chúng ta có các loại cảm biến khác nhau, bao gồm cả các cảm biến nhiệt độ. Mỗi cảm biến xử lý một tập hợp các bộ AVO tự mô tả. Một sự tự mô tả của một cảm biến nhiệt độ có thể giống như cái được trình bày ở danh sách 7.3. Nếu bản tin lợi ích trình bày ở hình 7.2 đi đến cảm biến nhiệt độ, việc kiểm tra một chiều cho thấy rằng việc tự mô tả

của cảm biến phù hợp với các thuộc tính trong lợi ích. Theo đó, cảm biến (nói chính xác hơn: tất cả các cảm biến phù hợp với lợi ích này và thực tế là sẵn sàng phục vụ cho các lợi ích) bắt đầu quan sát môi trường của chúng và tạo ra các bản tin dữ liệu khi mà sự kiện của lợi ích thực sự xảy ra. Một ví dụ của một bản tin dữ liệu này được trình bày trong danh sách 7.4. Như đã được giải thích ở trên, các cảm biến không phù hợp lưu lợi ích trong cache lợi ích của chúng.

Người ta buộc phải ghi chú rằng tập hợp các thuộc tính được sử dụng trong các danh sách thực sự chỉ là các ví dụ; nói chung, các thuộc tính và các giá trị của chúng phụ thuộc vào ứng dụng.

```

parameters : attribute sets A and B
               // A corresponds to the interest , B to the data message
foreach attribute a in A where a.op is formal {
    matched = false
    foreach attribute b in B where
        a.key == b.key and b.op is actual {
        if b.val satisfies condition
            expressed by a.key and a.val then {
            matched = true
        }
    }
    if (not matched ) then {
        return false
    }
}
return true; // matching successful !

```

Danh sách 7.1 Thuật toán phù hợp một chiều

```

<type ,temperature ,EQ >
<threshold -from -below ,20,IS >
<x-coordinate ,20,LE >
<x-coordinate ,0,GE >
<y-coordinate ,20,LE >
<y-coordinate ,0,GE >
<interval ,0.05 ,IS >
<duration ,10,IS >
<class ,interest ,IS >

```

Danh sách 7.2 Ví dụ bản tin lợi ích (interest message)

```

<type ,temperature ,IS >
<x-coordinate ,10,IS >
<y-coordinate ,10,IS >

```

Danh sách 7.3 Ví dụ cảm biến nhiệt độ

```

<type ,temperature ,IS >
<x-coordinate ,10,IS >
<y-coordinate ,10,IS >
<temperature ,20.01 ,IS >
<class ,data ,IS >

```

Danh sách 7.4 Ví dụ bản tin dữ liệu

Một đặc điểm quan trọng của bất kỳ hệ thống đặt tên nào là **tính diễn cảm** (expressiveness) của nó. Chú thích này rất khó để định nghĩa; nó bao gồm cả một phép đo của nhiều thứ mà có thể thực sự được đặt tên bằng *hệ thống đặt tên* (the naming system) cũng như

nỗ lực cần thiết để biểu diễn/ viết tên. Một đặc điểm quan trọng thứ hai đó là nỗ lực tính toán (*computational effort*), nỗ lực này là cần thiết để kiểm tra liệu một thực thể (trong ví dụ tiếp theo: gói dữ liệu với thuộc tính thực sự của nó) có phù hợp với một cái tên hay không (ở đây: một lợi ích). Đối với ví dụ này, một người có thể thấy rằng (i) các thuộc tính là đơn giản để đánh giá và (ii) tiến trình phù hợp một chiều chỉ cho phép một chế độ kết hợp thuộc tính – tất cả các thuộc tính đề cử phải phù hợp một cách đồng thời, tương ứng với một phép toán logic AND giữa chúng. Bằng cách giới hạn việc kết hợp thuộc tính bằng AND, gần như là có bao nhiêu thuộc tính chính thức thì có bấy nhiêu việc kiểm tra cần phải thực hiện. Do đó, nỗ lực tính toán là tuyến tính theo số lượng các thuộc tính chính thức. Việc cho phép các hàm Boolean tổng quát trên một thuộc tính đơn phù hợp có khả năng yêu cầu thời gian tính toán nhiều hơn rất nhiều.

Hệ thống đặt tên của Heidemann [15] được thực hiện (cùng với sự truyền tin trực tiếp), và hai ví dụ ứng dụng cụ thể được thảo luận: sự kết hợp trong mạng và một khả năng để chỉ rõ các câu hỏi lồng vào nhau (**nested queries**). Các câu hỏi lồng vào nhau là một cách rõ ràng để phát hiện các sự kiện phức hợp. Trong một sự kiện phức hợp, sự xuất hiện của một sự kiện kích hoạt lợi ích trong một sự kiện khác. Điểm chủ chốt trong các cuộc thử nghiệm này đó là bằng cách sử dụng một hệ thống đặt tên dựa trên nội dung và bằng cách triển khai cái gọi là **các bộ lọc** vào trong mạng. Khi đó mạng có thể xử lý dữ liệu được đặt tên, một khối lượng tính toán tốt có thể được di chuyển vào mạng, gần với các nguồn sự kiện. Sự kết hợp cục bộ và việc xử lý có thể tiết kiệm băng thông đáng kể. Giao thức SPIN giới thiệu cái gọi là **metadata**, nghĩa là một mô tả dữ liệu mà một nút có thể xử lý. Metadata được giả thiết là ngắn hơn dữ liệu thực tế. Một nút nguồn tạo ra dữ liệu phổ biến một quảng cáo chứa metadata qua mạng. Các nút khác quan tâm đến dữ liệu được quảng cáo gửi trả lại một gói tin yêu cầu. Dựa vào việc nhận một yêu cầu, nút nguồn gửi gói dữ liệu đầy đủ. Rõ ràng là có nhiều kiểu đặt tên phức tạp hơn và cung cấp thêm các giao thức giải pháp/ràng buộc cho phép sự phức tạp hơn nhiều. Người ta có thể tích hợp việc đánh địa chỉ dựa trên nội dung với phần sụn (middleware) công khai/thuê bao cho mạng không dây dựa trên IP và mạng ad hoc mà không chỉ rõ một kiểu đặt tên cụ thể nào.

Các hệ thống đặt tên và các giao thức giải pháp cũng đóng một vai trò quan trọng trong việc phát hiện dịch vụ, các mạng chia sẻ file ngang hàng - peer to peer (ví dụ: các mạng đánh địa chỉ dựa trên nội dung), và hệ thống thư mục X.500. Tuy nhiên, các hệ thống này không thể áp dụng trực tiếp vào các mạng cảm biến bởi vì kiến trúc và các tập hợp yêu cầu khác nhau của chúng. Hiệu suất năng lượng không đóng vai trò gì trong thiết kế của các hệ thống này.

7.4.2 Đánh địa chỉ theo địa lý

Thường thì dễ thuận tiện hơn cho những người sử dụng trong việc biểu diễn các yêu cầu của họ với một mạng cảm biến dưới dạng không chỉ là *loại (type)* và phương thức của dữ liệu mà họ muốn nhận được mà còn là vùng (*region*) hoặc vị trí (*location*) nơi mà dữ liệu bắt nguồn từ đó. Bằng cùng một mã thông báo (token), giống như đối với các địa chỉ dựa trên nội dung, người sử dụng không muốn chỉ rõ một cách tách biệt mỗi nút thuộc vào khu vực lợi ích nào nhưng họ thích **chỉ rõ một vùng** và để mạng chỉ ra các cảm biến nào là phù hợp. Hơn nữa, nếu đã biết vị trí của một nút cảm biến, các **kiểu định tuyến địa lý** có thể được áp dụng. Có nhiều cách khác nhau để chỉ rõ một vùng, ví dụ:

- Chỉ rõ một điểm đơn
- Chỉ rõ một vòng tròn hoặc một mặt cầu bằng cách cho tâm điểm và bán kính.
- Chỉ rõ một hình chữ nhật hoặc một hình hộp bằng cách cho hai hoặc ba điểm góc.
- Chỉ rõ một đa giác (hai chiều) hoặc một polytope (ba chiều) bằng cách cho một danh sách các điểm.

Việc đánh địa chỉ địa lý yêu cầu rằng một cảm biến có thể kiểm tra xem liệu vị trí của nó có nằm trong một vùng đã cho hay không. Việc kiểm tra này có thể phức tạp nếu các đa giác/ polytope chung được sử dụng để chỉ rõ vùng lợi ích. Việc kiểm tra một điểm trong đa giác như vậy là một nhiệm vụ tiêu chuẩn trong hình học dùng máy điện toán. Việc sử dụng các hình dạng phức tạp hơn cũng có nhược điểm là nhiều điểm và do đó cần nhiều bit dữ liệu để chỉ rõ chúng.

7.5 Tổng kết

Trong bất cứ mạng nào bao gồm cả các mạng cảm biến, đều có các mức khác nhau của các địa chỉ và các tên, ví dụ các địa chỉ MAC và các địa chỉ lớp mạng. Các địa chỉ MAC được sử dụng để phân biệt giữa các nút lân cận sát cạnh nhau và các địa chỉ lớp mạng được dùng để nhận dạng (nhóm của) các nút trong một mạng đa bước nhảy (multihop). Điều quan tâm chủ yếu đối với việc đặt tên và đánh địa chỉ trong các mạng cảm biến đó là phần tiếp đầu (overhead) và việc tiêu thụ năng lượng mà các phương pháp này phải gánh chịu. Ví dụ, năng lượng có thể bị tiêu hao khi biểu diễn địa chỉ không hiệu quả, khi chi phí cho địa chỉ cao và các giao thức không định rõ vị trí, hoặc khi yêu cầu vài giao thức giải pháp ràng buộc/địa chỉ.

Ở mức thấp nhất là các địa chỉ MAC, các địa chỉ này trong các giao thức MAC dựa trên sự ganh đua là không thể thiếu được để nhận biết các sự tiết kiệm năng lượng từ việc tránh nghe lén. Nếu được yêu cầu bởi giao thức MAC, chúng cần thiết được biểu diễn trong tất cả các gói dữ liệu và có thể đem lại phần tiếp đầu có ý nghĩa, đặc biệt là nếu dữ liệu người sử dụng là nhỏ. Đối lập với các giao thức MAC dựa trên chương trình (schedule-based MAC protocols), các địa chỉ MAC luôn là cần thiết trong các giao thức MAC dựa trên sự ganh đua bởi vì một gói được truyền đi có thể có khả năng có nhiều máy thu. Nói chung ở đây có một sự không dung hòa giữa các yêu cầu chặt chẽ về sự duy nhất và kích thước của địa chỉ. Mặt khác, các địa chỉ duy nhất cục bộ (mà đủ cho lớp MAC) yêu cầu một giao thức phân phối địa chỉ. Tuy nhiên, nếu phần lớn các nút cảm biến là không di chuyển, thì việc tiết kiệm năng lượng có được là do (các biểu diễn hiệu quả của) các địa chỉ duy nhất cục bộ đóng góp một cách nhanh chóng. Việc chạy các giao thức ấn định địa chỉ với sự chặt chẽ hơn các yêu cầu duy nhất cục bộ (có thể nói, duy nhất toàn mạng) nhanh chóng trở thành không hiện thực trong các mạng cảm biến không dây bởi vì phải giải quyết một vấn đề nhất trí phân tán, mà chắc chắn là có phần tiếp đầu đáng kể. Thêm vào đó, kích thước biểu diễn địa chỉ của các địa chỉ duy nhất cục bộ chỉ phụ thuộc vào mật độ của mạng chứ không phụ thuộc vào số lượng tuyệt đối của các nút trong mạng đó. Đây không phải là trường hợp đối với các địa chỉ duy nhất toàn mạng.

Các giao thức định tuyến truyền thống cần các địa chỉ duy nhất toàn mạng hoặc toàn cầu để tìm các nút cá nhân. Tuy nhiên trong các mạng cảm biến không dây, việc đánh địa chỉ dựa trên nội dung cung cấp một sự thay thế hấp dẫn. Chìa khóa ở đây là sự tích hợp các địa

chỉ dựa trên nội dung với việc định tuyến và khả năng của chúng để cho phép việc xử lý trong mạng.

CÂU HỎI ÔN TẬP CHƯƠNG 7

1. Cách sử dụng địa chỉ và tên trong mạng cảm biến?
2. Nhiệm vụ của quản lý địa chỉ?
3. Phân phối và ấn định địa chỉ?
4. Quản lý tên và địa chỉ trong WSN?
5. Phương pháp ấn định địa chỉ cho giao thức MAC?
6. Cách thức đánh địa chỉ theo địa nội dung và theo địa lý?

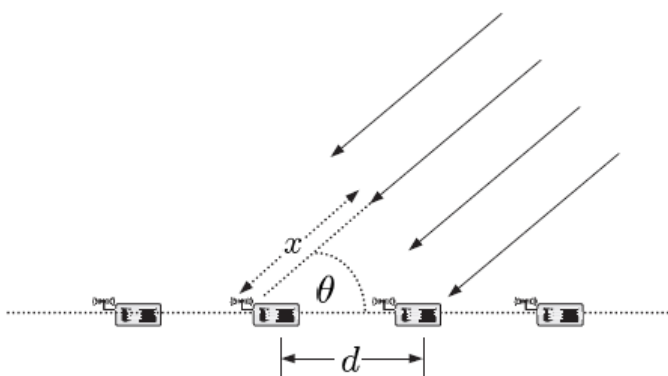
CHƯƠNG 8. ĐỒNG BỘ THỜI GIAN

Mục đích của chương

Thời gian là một khía cạnh quan trọng đối với nhiều ứng dụng và nhiều giao thức được tìm thấy trong các mạng cảm biến không dây. Các nút có thể đo thời gian bằng cách sử dụng các đồng hồ cục bộ, các đồng hồ này được điều khiển bởi các bộ dao động. Do các lệch pha ngẫu nhiên và tốc độ lệch của các bộ tạo dao động, nên việc đọc thời gian cục bộ của nút bắt đầu có sự khác nhau, nghĩa là chúng mất đồng bộ hay không còn chính xác. Vấn đề đồng bộ thời gian là một vấn đề tiêu chuẩn trong các hệ thống phân tán. Trong các mạng cảm biến không dây, những yêu cầu bắt buộc mới phải được quan tâm đến, ví dụ: việc tiêu thụ năng lượng của các thuật toán, số lượng lớn các nút có thể được đồng bộ, và các yêu cầu chính xác khác nhau. Chương này đưa ra một giới thiệu về vấn đề đồng bộ thời gian nói chung và thảo luận các chi tiết riêng của các mạng cảm biến không dây. Tiếp theo đó, một số các giao thức được đề xuất cho các mạng cảm biến sẽ được thảo luận chi tiết hơn.

8.1 Giới thiệu về vấn đề đồng bộ thời gian

Trong chương này, chúng ta giải thích tại sao việc đồng bộ thời gian là cần thiết, tiếp theo là một danh sách các đặc điểm mà các thuật toán đồng bộ thời gian khác nhau có thể có. Chúng ta cũng thỏa thuận các thách thức và các yêu cầu bắt buộc đặc biệt đối với các thuật toán đồng bộ thời gian trong các mạng cảm biến không dây.



Hình 8.1 Xác định góc tới của một sự kiện âm thanh ở xa bằng một mạng các cảm biến âm thanh.

8.1.1 Sự cần thiết của việc đồng bộ thời gian trong các mạng cảm biến không dây

Thời gian đóng một vai trò quan trọng trong hoạt động của các hệ thống phân tán nói chung và trong các mạng cảm biến không dây nói riêng, bởi vì chúng được giả thiết để quan sát và tương tác với các hiện tượng vật lý. Một ví dụ đơn giản sẽ minh họa việc cần thiết của các thông tin định thời chính xác (hình 8.1). Một sóng âm thanh được tạo ra bởi một nguồn âm thanh ở một khoảng cách khá xa tác động vào một mạng các cảm biến âm thanh và góc tới sẽ được ước lượng. Mỗi cảm biến trong các cảm biến biết vị trí riêng của nó một cách chính xác và ghi lại thời gian đến của sự kiện âm thanh. Trong việc thiết lập cụ thể như trình bày ở hình vẽ thì góc θ có thể được xác định khi các chiều dài d và x đã biết, sử dụng mối quan hệ

lượng giác học $x = d \cdot \sin \theta$ và theo đó thì $\theta = \arcsin(x/d)$. (Để cho ví dụ là đơn giản, ta giả thiết rằng các cảm biến ở xa cung cấp các thông tin cần thiết để phân biệt giữa các góc θ và $2\pi - \theta$). Khoảng cách cảm biến d có thể được suy ra từ vị trí đã biết của các cảm biến và khoảng cách x có thể được suy ra từ sự khác biệt thời gian Δt giữa các chỉ thị cảm biến và tốc độ âm thanh đã biết $c \approx 330$ m/s, sử dụng $x = c \cdot \Delta t$. Giả thiết $d = 1$ m và $\Delta t = 0,001$ s thì $\theta \approx 0,336$ (theo radians). Nếu các đồng hồ của các cảm biến chỉ chính xác trong khoảng 500 μ s, sự khác biệt về thời gian thực tế có thể nằm trong phạm vi giữa 500 và 1500 μ s, và do đó các ước lượng cho θ có thể biến đổi giữa $\theta \approx 0,166$ và $\theta \approx 0,518$. Do đó, một lỗi có vẻ nhỏ trong đồng bộ thời gian có thể dẫn đến các ước lượng khác nhau đáng kể. Có ít nhất hai cách để nhận được một sự ước lượng có độ tin cậy cao hơn. Cách thứ nhất (và là cách được tập trung trong chương này) đó là giữ các đồng hồ của các cảm biến càng gần đồng bộ càng tốt, sử dụng **các thuật toán đồng bộ thời gian** chuyên dụng. Cách thứ hai là kết hợp các bộ chỉ thị của nhiều cảm biến và lấy giá trị trung bình của các lỗi ước lượng được. Có nhiều ứng dụng khác yêu cầu đồng bộ thời gian chính xác, ví dụ: ứng dụng tạo chùm sóng (beamforming) Tuy nhiên, không những các ứng dụng WSN mà còn nhiều giao thức mạng được sử dụng trong các mạng cảm biến cần thời gian chính xác. Các ví dụ cơ bản là các giao thức MAC dựa trên TDMA hoặc các giao thức MAC với việc đánh thức phối hợp (coordinated wakeup), giống như giao thức sử dụng trong tiêu chuẩn WPAN IEEE 802.15.4. Các nút cảm biến chạy giao thức TDMA cần phải thỏa thuận về các ranh giới của các khe thời gian; nếu không các truyền dẫn của chúng sẽ bị chồng nhau và xung đột.

Điều quan trọng phải chú ý là thời gian cần thiết trong các mạng cảm biến gắn chặt với **thời gian vật lý**, nghĩa là hai nút cảm biến phải có cùng ý tưởng về khoảng thời gian 1s và thêm vào đó nút cảm biến thứ hai nên tiến càng gần đến 1s của **thời gian thực** (real time) hoặc **thời gian tổng kết hợp** (coordinated universal time (UTC) - Độ dài của một giây được xác định một cách chính xác theo số lượng của các sự chuyển trạng thái giữa hai mức siêu nhỏ của trạng thái nền của Caesium-133 và thông tin này được cung cấp bởi nhiều đồng hồ nguyên tử trải khắp trên toàn thế giới. Các hoạt động của các đồng hồ này cùng với nhau cung cấp thang thời gian UTC quốc tế. Thời gian UTC không sử dụng các đồng hồ nguyên tử một cách trực tiếp nhưng các giây thường được chèn vào hoặc xóa đi để giữ cho thời gian UTC đồng bộ với các thang thời gian thiên văn khi chúng được sử dụng cho các mục đích định vị). Thời gian vật lý phải được phân biệt với khái niệm **thời gian logic**, thời gian logic cho phép xác định thứ tự của các sự kiện trong một hệ thống phân tán nhưng không cần thiết chỉ ra bất cứ sự tương ứng nào với thời gian thực. Nhiều thuật toán đồng bộ thời gian xuất hiện qua thời gian. Chúng ta giới hạn việc thảo luận vào các thuật toán liên quan tới các mạng cảm biến không dây, nghĩa là các thuật toán mà quan tâm đến việc tiêu thụ năng lượng và các thuật toán này có thể chạy trong các mạng kích thước lớn.

8.1.2 Các khóa nút và vấn đề về độ chính xác

Gần như tất cả các thiết bị đồng hồ của các nút cảm biến và các máy tính đều chia sẻ một cấu trúc chung: Nút xử lý một **bộ tạo dao động** (oscillator) của một tần số được chỉ rõ và một **thanh ghi đếm** (counter register), mà được đếm tăng dần trong phần cứng sau một số lượng các xung dao động nào đó. Phần mềm của nút chỉ truy cập vào giá trị của thanh ghi này và thời gian giữa hai lần tăng lên xác định **giải pháp thời gian** (time resolution) có thể nhận

được: các sự kiện xảy ra giữa hai lần tăng lên không thể phân biệt được bằng nhãn thời gian của chúng.

Giá trị của **đồng hồ phần cứng** (hardware clock) của nút i tại thời gian thực t có thể được biểu diễn dưới dạng $H_i(t)$. Nó có thể được hiểu như một khái niệm trừu tượng của thanh ghi đếm cung cấp một giá trị thời gian tăng liên tục. Một phương pháp chung để tính toán một **đồng hồ phần mềm** cục bộ (software clock) $L_i(t)$ từ giá trị này sẽ cung cấp một phép biến đổi có liên quan tới đồng hồ phần cứng:

$$L_i(t) := \theta_i \cdot H_i(t) + \varphi_i.$$

φ_i được gọi là **góc dịch pha** (phase shift) và θ_i được gọi là tốc độ dịch (drift rate). Giả thiết rằng người ta thường hoặc không thể hoặc không muốn có ảnh hưởng đến bộ tạo dao động hoặc thanh ghi đếm, một người có thể thay đổi các tham số θ_i và φ_i để thực hiện **điều chỉnh đồng hồ**. Bây giờ chúng ta có thể xác định khái niệm về **độ chính xác tương đối** (precision) của các đồng hồ trong một mạng, ở đó chúng ta chia thành 2 trường hợp:

Đồng bộ ngoài: các nút $1, 2, \dots, n$ được nói là **chính xác tuyệt đối** tại thời điểm t trong một giới hạn δ nếu $|L_i(t) - t| < \delta$ đúng cho tất cả các nút $i \in \{1, 2, \dots, n\}$.

Đồng bộ trong: các nút $1, 2, \dots, n$ được nói là đồng thuận về thời gian trong một giới hạn δ nếu $|L_i(t) - L_j(t)| < \delta$ đúng cho tất cả các nút $i, j \in \{1, 2, \dots, n\}$.

Để nhận được đồng bộ ngoài, phải có sẵn một nguồn thời gian thực/ thời gian UTC tin cậy, ví dụ: một bộ thu GPS chẳng hạn. Rõ ràng là nếu các nút $1, 2, \dots, n$ được đồng bộ ngoài với giới hạn δ , thì chúng cũng được đồng bộ trong với giới hạn 2δ .

Có 3 vấn đề:

- Các nút được chuyển mạch tại các thời điểm khác nhau và ngẫu nhiên, do đó nếu không có hiệu chỉnh thì các **pha** ban đầu của chúng φ_i cũng là ngẫu nhiên.
- Các bộ tạo dao động thường có trước một độ lệch ngẫu nhiên nhỏ khỏi tần số danh nghĩa của chúng, được gọi là **độ lệch** (drift) và đôi khi gọi là **lệch đồng hồ** (clock skew). Điều này có thể là do thạch anh có tạp chất nhưng các bộ tạo dao động cũng phụ thuộc vào nhiều điều kiện môi trường như áp suất, nhiệt độ, ..., mà các tham số môi trường trong mạng cảm biến được triển khai có thể khác nhiều so với các tham số trong phòng thí nghiệm. Độ lệch của đồng hồ thường được biểu diễn dưới dạng **các phần triệu** (ppm) và cung cấp số lượng của các dao động thêm vào hay mất đi mà một đồng hồ tạo ra trong một lượng thời gian cần thiết cho một triệu dao động ở tốc độ danh định. Nói chung, các bộ tạo dao động rẻ hơn – giống như các bộ tạo dao động được sử dụng trong các thiết kế cho các nút cảm biến rẻ – có các độ lệch lớn hơn với xác suất cao hơn. Trong nhiều tài liệu xuất bản từ lĩnh vực các mạng cảm biến không dây, các độ lệch đồng hồ nằm trong phạm vi giữa 1 và 100 ppm. Trong tài liệu Berkeley, bảng thông số chỉ rõ tốc độ lệch cực đại là 40 ppm. Một độ lệch 1ppm có nghĩa là cứ $\approx 11,6$ ngày thì có 1s lỗi, một độ lệch 100 ppm có nghĩa là cứ $\approx 2,78$ h ngày thì có 1s lỗi.
- Tần số của bộ tạo dao động biến thiên theo thời gian. Có các biến đổi trong giai đoạn ngắn – gây ra bởi các sự biến đổi nhiệt độ, các biến đổi điện áp nguồn điện cung cấp, áp suất không khí, v.v – cũng như các biến đổi trong giai đoạn dài do tuổi thọ của bộ tạo dao động. Thường là an toàn khi giả thiết rằng tần số của bộ tạo dao

động là ổn định chấp nhận được theo thời gian trong phạm vi phút đến vài chục phút. Nói cách khác, điều này cũng ngụ ý rằng các thuật toán đồng bộ thời gian nên tái đồng bộ cứ vài phút một lần để theo dấu các tần số biến đổi.

Người ta ngụ ý rằng thậm chí hai nút có bộ tạo dao động cùng loại và được bắt đầu tại cùng một thời điểm với các đồng hồ logic giống nhau, thì độ lệch $|L_i(t) - L_j(t)|$ có thể trở nên lớn tùy ý khi t tăng lên. Do đó, một giao thức đồng bộ thời gian là cần thiết. Vậy giao thức đồng bộ thời gian như vậy phải chạy thường xuyên như thế nào? Giả thiết rằng một nút chỉ điều chỉnh dịch pha φ_i của nó như là một kết quả của một vòng đồng bộ. Nếu tốc độ lệch của bộ tạo dao động của nút đó là hằng số và được biết bằng x ppm và nếu độ chính xác mong muốn là δ s, thì sau gần $\frac{\delta}{x \cdot 10^{-6}}$ s sự bất buộc chính xác bị vi phạm. Đối với $x = 20$ ppm và một độ chính xác 1 ms, một **sự tái đồng bộ** là cần thiết cứ mỗi 50s. Các phương pháp tiên tiến hơn cố gắng đánh giá và sửa sai không chỉ là độ lệch pha φ_i mà còn độ lệch hiện thời θ_i , hy vọng rằng việc kéo dài các chu kỳ trước khi tái đồng bộ được yêu cầu. Một lần nữa, một sự đồng bộ một lần là không hữu ích, bởi vì tốc độ lệch là biến đổi theo thời gian. Tuy nhiên, người ta thường có thể giới hạn tốc độ lệch cực đại, nghĩa là có một $\rho_i > 0$ sao cho:

$$\frac{1}{1 + \rho_i} \leq \frac{d}{dt} H_i(t) \leq 1 + \rho_i \quad (8.1)$$

và nó được dùng để tìm tần số tái đồng bộ duy trì.

8.1.3 Các đặc điểm và cấu trúc của các thuật toán đồng bộ thời gian

Các thủ tục đồng bộ thời gian có thể được phân loại theo tiêu chí nào đó:

Thời gian vật lý so với thời gian logic: trong các mạng cảm biến không dây, các ứng dụng và các giao thức phần lớn yêu cầu thời gian vật lý.

Đồng bộ trong so với đồng bộ ngoài: các thuật toán có thể hoặc không thể yêu cầu đồng bộ thời gian với các trục thời gian ngoài giống như UTC.

Thuật toán tổng quát hoặc thuật toán cục bộ: một thuật toán tổng quát cố gắng giữ tất cả các nút của một mạng cảm biến (hoặc một phần của mạng) được đồng bộ. Phạm vi của các thuật toán cục bộ thường được giới hạn vào một số vùng lân cận về địa lý của một sự kiện quan tâm (interesting event). Trong các thuật toán tổng quát, các nút do đó được yêu cầu để giữ đồng bộ với không chỉ các nút lân cận một bước nhảy (single-hop neighbors) mà còn với các nút ở khoảng cách xa (nhiều bước nhảy - multihop). Rõ ràng là một thuật toán cung cấp đồng bộ tổng quát cũng cung cấp đồng bộ cục bộ.

Thời gian tuyệt đối so với thời gian tương đối: nhiều ứng dụng giống như ví dụ đơn giản đã được trình bày trong phần 8.1.1 chỉ cần các *sự chênh lệch thời gian* chính xác và nó phải đủ để đánh giá độ lệch thay vì bù pha. Tuy nhiên, sự đồng bộ tuyệt đối là trường hợp tổng quát hơn vì sự đồng bộ tương đối chỉ là một trường hợp đặc biệt của nó.

Thuật toán dựa trên phần cứng so với dựa trên phần mềm: một số thuật toán yêu cầu phần cứng chuyên dụng như các bộ thu GPS hoặc thiết bị truyền thông chuyên dụng trong khi các thuật toán dựa trên phần mềm sử dụng việc chuyển bản tin không mã hóa, sử dụng cùng các kênh như đối với các gói dữ liệu thông thường.

Đồng bộ trước so với một đồng bộ sau (A priori versus a posteriori synchronization): trong một thuật toán đồng bộ trước, giao thức đồng bộ thời gian chạy suốt cả thời gian, thậm chí cả khi không có sự kiện bên ngoài để quan sát. Trong một thuật toán đồng bộ sau (còn gọi là đồng bộ hậu thực tế), quá trình đồng bộ được kích hoạt bởi một sự kiện bên ngoài.

Giới hạn chính xác tiên định (deterministic) so với giới hạn chính xác không tiên định (stochastic): một số thuật toán có thể (dưới các điều kiện nào đó) đảm bảo tuyệt đối các giới hạn trên của lỗi đồng bộ giữa các nút hoặc đối với thời gian mở rộng. Các thuật toán khác chỉ có thể đưa ra các giới hạn không tiên định (stochastic) trong hoàn cảnh lỗi đồng bộ có một xác suất nào đó nhỏ hơn một giới hạn đã được mô tả.

Quy tắc cập nhật đồng hồ cục bộ: Làm cách nào một nút cập nhật các tham số đồng hồ cục bộ φ_i and θ_i của nó? Một yêu cầu thường thấy đó là phải tránh việc nhảy lùi thời gian, nghĩa là đối với $t < t'$ thì sẽ không xảy ra $L_i(t) > L_i(t')$ sau một điều chỉnh. Thêm vào một yêu cầu có thể tránh được các sự nhảy bất thành hình, nghĩa là độ lệch $L_i(t') > L_i(t)$ đối với thời gian t ngay tức thì trước khi và đối với thời gian t' ngay tức thì sau khi tái điều chỉnh phải nhỏ.

Các **tham số biểu diễn** quan trọng nhất của các thuật toán đồng bộ thời gian là như sau:

Độ chính xác tương đối: đối với các thuật toán tiên định thì lỗi đồng bộ cực đại giữa một nút và thời gian thực hoặc giữa hai nút là đáng quan tâm; đối với các thuật toán không tiên định (stochastic) thì lỗi trung bình, độ lệch lỗi là đáng quan tâm.

Giá thành năng lượng: giá năng lượng của một giao thức đồng bộ thời gian phụ thuộc vào nhiều nhân tố: số lượng các gói trao đổi trong một vòng của thuật toán, số lượng các tính toán cần thiết để xử lý các gói, và tần số tái đồng bộ được yêu cầu.

Các yêu cầu bộ nhớ: để đánh giá tốc độ lệch, cần có lịch sử của các gói đồng bộ thời gian trước đó. Nói chung, một lịch sử dài cho phép các đánh giá (ước lượng) chính xác hơn về giá thành của việc tiêu thụ bộ nhớ tăng lên; so sánh với Phần 8.2.2.

Dung sai lỗi: thuật toán có thể đương đầu với các nút lỗi, với lỗi dễ xảy ra (error-prone) và các đường truyền thông biến đổi theo thời gian tốt như thế nào? Nó là hữu ích để phân tích các giao thức đồng bộ thời gian cho các mạng cảm biến không dây thành bốn khối xây dựng trên khái niệm, ba khối đầu tiên của chúng là:

- **Khối phát hiện sự kiện tái đồng bộ:** nhận dạng các điểm theo thời gian mà tại đó việc tái đồng bộ được kích hoạt. Trong phần lớn các thuật toán, việc tái đồng bộ được kích hoạt một cách có chu kỳ với một chu kỳ phụ thuộc vào tốc độ lệch cực đại. Một quá trình tái đồng bộ đơn lẻ được gọi là một **vòng** (round). Nếu các vòng có thể chồng nhau theo thời gian, cần có các số chuỗi (dãy số) để phân biệt chúng và để một nút bỏ đi tất cả các vòng trừ các vòng tái đồng bộ mới nhất.
- **Khối đánh giá đồng hồ từ xa:** yêu cầu các giá trị đồng hồ từ các nút ở xa/ các đồng hồ ở xa. Có hai biến thể chung. Thứ nhất, trong **kỹ thuật truyền thời gian (time transmission technique)**, một nút i gửi đồng hồ cục bộ của nó $L_i(t)$ tại thời điểm t tới một nút lân cận j , nút j nhận đồng hồ cục bộ mà nút i gửi tới vào thời điểm $L_j(t')$ (với $t' > t$). Về cơ bản, nút j giả thiết là $t' \approx t$ và sử dụng $L_i(t)$ như sự đánh giá cho

thời gian $L_i(t')$. Việc đánh giá này có thể thực hiện chính xác hơn bằng cách loại bỏ các nhân tố đã biết khỏi hiệu $t' - t$, ví dụ thời gian mà gói i chiếm kênh và trễ truyền lan. Thứ hai, trong kỹ thuật **đọc đồng hồ từ xa**, một nút j gửi một bản tin yêu cầu tới nút i khác, nút i này trả lời bằng một gói tin trả lời. Nút j đánh giá đồng hồ của i từ thời gian khứ hồi (round-trip time, thời gian đi rồi về) của bản tin và biết các lần truyền dẫn gói tin. Cuối cùng thì nút j có thể thông báo cho nút i kết quả.

- **Khối hiệu chỉnh đồng hồ:** tính sự điều chỉnh của đồng hồ cục bộ dựa trên các kết quả của khối đánh giá đồng hồ từ xa.
- **Khối thiết lập mạng lưới đồng bộ:** xác định nút nào đồng bộ với nhau trong một mạng đa bước nhảy. Trong các mạng kết nối đầy đủ, khối này là ít quan trọng.

8.1.4 Đồng bộ thời gian trong các mạng cảm biến không dây

Trong các mạng cảm biến không dây, có một số thông số mà ảnh hưởng đến các yêu cầu và thiết kế của các thuật toán đồng bộ thời gian:

- Một thuật toán phải tương ứng với các mạng đa bước nhảy lớn, các mạng mà có các nút không tin cậy và có giới hạn năng lượng giới hạn khắt khe. Yêu cầu tỉ lệ tương ứng có liên quan đến số lượng của các nút và mật độ của nút/địa vị của nút trung bình.
- Các yêu cầu chính xác có thể khác nhau, trải dài từ ms đến s.
- Việc sử dụng thêm phần cứng chỉ cho các mục đích đồng bộ thời gian phần lớn được quy định rõ bởi vì giá thành phụ thêm và sự bất lợi về năng lượng xảy ra bởi mạch chuyên dụng.
- Mức độ di động là thấp. Một kết quả quan trọng đó là một nút có thể tiến tới các nút lân cận bất cứ lúc nào, trong khi đó trong các mạng với mức độ di động cao, việc kết nối không liên tục và truyền tin lác đác là chi phối.
- Thường thì không có giới hạn trên cố định đối với trễ chuyển giao gói tin, bởi vì giao thức MAC, các lỗi gói tin, và các sự tái truyền tin.
- Trễ truyền lan giữa các nút lân cận là không đáng kể. Một khoảng cách 30m cần 10^{-7} s với tốc độ ánh sáng $c \approx 300.000.000$ m/s.
- Cấu hình thủ công (Manual configuration) của các nút đơn không phải là một tùy chọn. Một số giao thức yêu cầu điều này, ví dụ: giao thức thời gian mạng (NTP-network time protocol), trong giao thức này mỗi nút phải được cấu hình với một danh sách các máy chủ thời gian.
- Thành ra độ chính xác tuyệt đối của các thuật toán đồng bộ thời gian phụ thuộc trầm trọng vào trễ giữa việc nhận được bit cuối cùng của một gói tin và thời gian khi nó được dán nhãn thời gian. Một cách tối ưu, việc dán nhãn thời gian được thực hiện ở các lớp thấp nhất, ví dụ: ở lớp MAC. Đặc điểm này thực hiện dễ hơn nhiều trong các nút cảm biến với phần mềm và phần sụn (firmware) mở so với nó sử dụng phần cứng thương mại như các card mạng thương mại IEEE 802.11.

Nhiều giao thức đồng bộ thời gian truyền thống cố gắng giữ các nút được đồng bộ tại tất cả các thời điểm, điều này là hợp lý khi không có các yêu cầu bắt buộc về năng lượng và cấu

hình liên kết mạng là đủ ổn định. Do đó, năng lượng buộc phải tiêu thụ liên tục để chạy các giao thức đồng bộ thời gian. Đối với nhiều ứng dụng mạng cảm biến thì điều này là cần thiết, ví dụ: khi nhiệm vụ chính của một mạng là giám sát môi trường cho các sự kiện hiếm khi xảy ra như cháy rừng chẳng hạn. Với **đồng bộ sau** (post-facto synchronization) (hay một posteriori synchronization), thì có thể nhận được một đồng bộ thời gian theo yêu cầu. Ở đây, các nút không được đồng bộ phần lớn thời gian. Khi một sự kiện bên ngoài thú vị được quan sát tại thời điểm t_0 , một nút i lưu nhận thời gian cục bộ $L_i(t_0)$ của nó và kích hoạt giao thức đồng bộ, giao thức ví dụ như cung cấp đồng bộ toàn mạng với thời gian UTC. Sau khi giao thức hoàn thành tại thời điểm t_1 muộn hơn một chút, nút i sẽ biết được sự bù tương đương Δ của nó đối với thời gian UTC, nghĩa là $t_1 = L_i(t_1) + \Delta$. Nút i có thể sử dụng thông tin này để liên hệ đến sự kiện quá khứ tại thời điểm t_0 cũng đối với thời gian UTC. Sau khi nút i chuyển giao thông tin về sự kiện, nó có thể lại đi vào chế độ ngủ, bỏ đồng bộ. Nói ngắn gọn, đồng bộ sau (post-facto synchronization) là đồng bộ theo yêu cầu và cho thời gian ngắn, để thông báo về một sự kiện quan trọng. Trước khi thảo luận một số các đề xuất cho các giao thức đồng bộ phù hợp với các mạng cảm biến, chúng ta hãy thảo luận tóm tắt một số giải pháp “hiển nhiên” và tại sao chúng không phù hợp.

- *Trang bị mỗi nút với một bộ thu GPS*: các bộ thu GPS vẫn có giá là vài đô la, cần một anten rời, cần có năng lượng liên tục để giữ đồng bộ (đạt được đồng bộ ban đầu mất vài phút), và có các hệ số hình dạng (form factors) không phù hợp với ý tưởng của các nút cảm biến rất nhỏ. Hơn nữa, để hữu ích, một bộ thu GPS cần một đường truyền thẳng (LOS-Light Of Sign) tới ít nhất bốn trong số các vệ tinh GPS, điều mà không luôn luôn có thể đạt được trong các địa hình đồi núi, rừng, hoặc các ứng dụng trong nhà. Một ứng dụng của GPS trong một mạng cảm biến là theo dấu động vật hoang dã.
- *Trang bị mỗi nút với bộ thu nào đó cho các tín hiệu UTC*: cùng các sự quan tâm áp dụng như đối với một bộ thu GPS. (Ví dụ: ở Đức một bộ thu DCF77 có thể được sử dụng cho mỗi nút).
- *Cho một số nút ở rìa của mạng cảm biến gửi các tín hiệu định thời mạnh*: giải pháp này có thể sử dụng ở trong nhà và ở địa hình bằng phẳng nhưng yêu cầu một tần số riêng và do đó cần một bộ thu riêng ở mỗi nút để máy chủ thời gian không gây nhiễu đến các truyền dẫn đang diễn ra.

Trong các phần tiếp sau đây, chúng ta trình bày một số đề xuất đối với các giao thức đồng bộ thời gian trong các mạng cảm biến không dây.

8.2 Các giao thức dựa trên đồng bộ máy phát/ máy thu

Trong loại giao thức này, một nút được gọi là máy thu, trao đổi các gói dữ liệu với nút khác, gọi là máy phát, để cho đồng hồ của máy thu đồng bộ với đồng hồ của máy phát. Một trong các giao thức cổ điển của loại này là giao thức thời gian mạng (network time protocol - NTP), giao thức này được sử dụng rộng rãi trong mạng Internet. Nói chung, máy phát/máy thu dựa trên các giao thức yêu cầu các đường kết nối song hướng giữa các nút lân cận.

8.2.1 Giao thức đồng bộ thời gian hạng nhẹ - *Lightweight time synchronization protocol (LTS)*

Giao thức đồng bộ thời gian ít quan trọng (LTS) được van Greunen và Rabaey trình bày [16] cố gắng để đồng bộ các đồng hồ của một mạng cảm biến với các đồng hồ giữ bởi các nút tham chiếu nào đó, ví dụ: các nút có thể có các bộ thu GPS. Giao thức có các nút bấm điều khiển mà cho phép sự không dung hòa giữa phí tổn năng lượng và độ chính xác có thể nhận được, và nó cung cấp các giới hạn chính xác không tiền định dưới các giả thuyết nào đó về hệ điều hành và phần cứng ở dưới. LTS không tạo ra hạn chế nào đối với quy định cập nhật đồng hồ cục bộ và nó không cố gắng đánh giá các tốc độ lệch thực tế.

LTS chia nhỏ thời gian đồng bộ thành hai khối:

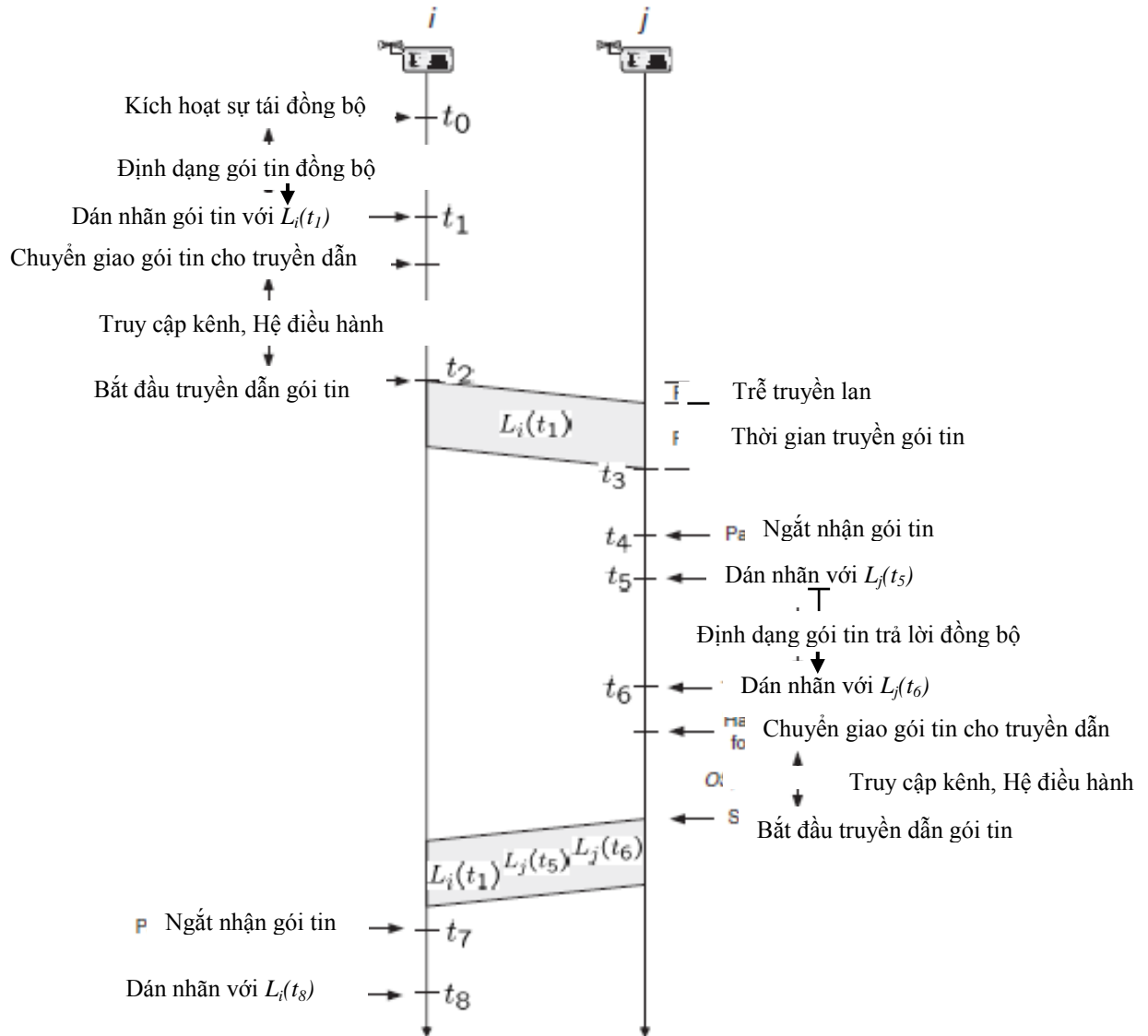
- Một giao thức đồng bộ theo cặp (pair-wise) đồng bộ hai nút lân cận.
- Để giữ tất cả các nút hoặc tập hợp các nút lợi ích đồng bộ với một nút tham chiếu chung, thì một cây bao trùm từ nút tham chiếu tới tất cả các nút được tạo ra. Nếu các lỗi đồng bộ một bước nhảy là độc lập và có phân bố giống nhau và có trung bình bằng 0, các nút lá của cây cũng có một lỗi đồng bộ có thể dự đoán được là bằng 0 nhưng độ lệch là tổng của các độ lệch dọc theo đường dẫn từ nút tham chiếu đến nút lá. Do đó, độ lệch này có thể được giảm nhỏ bằng cách tìm kiếm một cây bao trùm có chiều dài nhỏ nhất.

Đồng bộ theo cặp (Pair-wise):

Đầu tiên chúng ta giải thích giao thức đồng bộ theo cặp (Hình 8.2). Giao thức sử dụng một kỹ thuật chỉ thị đồng hồ từ xa. Giả thiết một nút i muốn đồng bộ đồng hồ của nó với đồng hồ của một nút j .

Sau khi việc tái đồng bộ được kích hoạt tại nút i , một gói tin yêu cầu đồng bộ được định dạng và dán nhãn thời gian tại thời điểm t_1 với thời gian $L_i(t_1)$. Nút i xử lý gói tin qua hệ điều hành và ngăn xếp giao thức, gói tin ở đó trong một khoảng thời gian nào đó. Trễ truy cập đường truyền có thể biến đổi nhiều và bù một phần đáng kể thời gian này. Thường thì, trễ này là một biến ngẫu nhiên. Khi nút i đang gửi bit đầu tiên tại thời điểm t_2 , nút j nhận bit cuối cùng của gói tin tại thời điểm $t_3 = t_2 + \tau + t_p$, trong đó τ là trễ truyền lan và t_p là thời gian truyền dẫn gói tin (chiều dài của gói tin được chia bởi tốc độ bit). Một thời gian sau (trễ ngắt - interrupt latency), tại thời điểm t_4 , gói tin đến được báo hiệu cho hệ điều hành của nút j hoặc ứng dụng thông qua một ngắt và nó được dán nhãn thời gian tại thời điểm t_5 với $L_j(t_5)$. Tại thời điểm t_6 , nút j định dạng gói trả lời của nó, dán nhãn thời gian với $L_j(t_6)$, và chuyển giao gói trả lời này tới hệ điều hành của nó và ngăn xếp của mạng. Gói này cũng bao gồm các nhãn thời gian phía trước $L_j(t_5)$ và $L_i(t_1)$. Nút i nhận ngắt thu gói tin (packet reception interrupt) tại thời điểm t_7 (là t_6 cộng phần tiếp đầu của hệ điều hành/ tiếp đầu của mạng), trễ truy cập đường truyền, trễ truyền lan, thời gian truyền gói tin, và trễ ngắt (interrupt latency) và dán nhãn thời gian cho nó tại thời điểm t_8 với $L_i(t_8)$. Bây giờ chúng ta hãy phân tích làm thế nào mà nút i suy luận ra việc hiệu chỉnh thời gian của nó. Chính xác hơn thì nút i muốn đánh giá $O = \Delta(t_1) := L_i(t_1) - L_j(t_1)$. Để làm điều đó, chúng ta giả thiết rằng không có độ lệch nào giữa các đồng hồ trong khoảng thời gian giữa t_1 và t_8 , nghĩa là $O = \Delta(t^*)$ với tất cả $t^* \in [t_1, t_8]$, và trong thực tế nút i đánh giá O bằng cách đánh giá $\Delta(t_5)$. Từ hình vẽ, nhãn thời gian $L_j(t_5)$,

mà nút i lấy lại, được tạo ra tại thời điểm nào đó không rõ giữa t_1 và t_8 . Tuy nhiên, chúng ta có thể giảm sự không chắc chắn này bằng các quan sát sau đây:



Hình 8.2 Một phần sơ đồ của hoạt động trong đồng bộ máy phát/máy thu

Có một trễ truyền lan τ cộng với thời gian truyền một gói tin t_p giữa t_1 và t_5 .

- Có một thời gian khác $\tau + t_p$ giữa t_5 và t_8 dành cho gói tin phản hồi (trả lời). Đối với các nút dừng, chúng ta có thể giả thiết một cách an toàn rằng các trễ truyền lan là như nhau ở cả hai hướng.
- Thời gian giữa t_5 và t_6 cũng được biết bởi nút i từ hiệu $L_j(t_6) - L_j(t_5)$.

Do đó, sự không chắc chắn về t_5 có thể được giảm xuống tới khoảng $I = [L_i(t_1) + \tau + t_p, L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))]$. Nếu chúng ta giả thiết rằng có các thời gian dành cho hệ điều hành và ngăn xếp mạng, thì các trễ ngắt cũng như trễ truy nhập đường truyền là như nhau ở cả hai hướng, sau đó nút i có thể kết luận rằng j tạo ra nhãn thời gian $L_j(t_5)$ của nó tại thời điểm (từ điểm quan sát i)

$$L_i(t_5) = \frac{L_i(t_1) + \tau + t_p + L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))}{2}$$

Do đó,

$$O = \Delta(t_5) = L_i(t_5) - L_j(t_5) = \frac{L_i(t_8) + L_i(t_1) - L_j(t_6) - L_j(t_5)}{2}$$

Bây giờ nút i có thể điều chỉnh đồng hồ cục bộ của nó bằng cách cộng thêm phần bù O vào đồng hồ của nó. Bằng cách này, nút i đồng bộ với thời gian cục bộ của nút j , với chi phí là hai gói tin. Ngoài ra, nếu mục đích là để nút j biết về O , thì cần thiết phải có một gói tin thứ ba, gửi từ i tới j và bao gồm cả O . Trong trường hợp này, toàn bộ việc đồng bộ cần ba gói tin. Lỗi đồng bộ cực đại của phương pháp này là $|I|/2$ nếu thời gian τ và t_p đã biết với độ chính xác cao. Lỗi đồng bộ thực tế về cơ bản có thể được cho là do các trễ ngắt khác nhau tại i và j , là do các thời gian khác nhau giữa việc lấy một ngắt thu và dán nhãn thời gian cho gói tin, và là do các thời điểm truy cập kênh truyền khác nhau. Các sự không chắc chắn này có thể giảm xuống đáng kể nếu nút yêu cầu có thể dán nhãn gói tin của nó càng muộn càng tốt, tốt nhất là ngay trước khi phát hoặc ngay sau khi nhận được truy cập đường truyền.

Nhiều tác giả bao gồm cả Elson trong [11] giả thuyết để cho máy thu có các gói tin nhận thời gian càng sớm càng tốt, ví dụ: trong một thường trình ngắt gọi là *gói tin đến* (*packet arrival*) từ bộ thu phát. Elson [11] chỉ ra rằng đối với nhiều bộ thu có nhiệm vụ dán nhãn thời gian cùng một gói tin trong các thường trình ngắt của chúng, sự khác nhau từng cặp (pair-wise) trong các thời gian tạo ra nhãn thời gian thực tế được phân phối thông thường với trung bình 0 và một độ lệch tiêu chuẩn $\sigma = 11,1 \mu s$ (so sánh với hình 8.6). Greunen và Rabaey [16] đưa ra giả thiết thêm rằng sự khác biệt trong các thời điểm truy cập kênh truyền tuân theo cùng một sự phân bố. Phụ thuộc vào mức độ của sự kết hợp giữa độ lệch thời gian truy cập và độ lệch nhãn thời gian, sự khác biệt của tổng của các biến này ít nhất gấp bốn lần độ lệch σ_2 của thiết bị này hay thiết bị khác. Do đó, van Greunen and Rabaey [16] phân loại tổng này (nghĩa là lỗi tổng trong đánh giá trên của O) như một biến ngẫu nhiên thông thường với độ lệch $4\sigma_2$ và độ lệch tiêu chuẩn 2σ . Người ta biết rõ đối với phân bố thông thường thì 99% của tất cả các tác động có một độ lệch gần 2,3 lần độ lệch tiêu chuẩn so với giá trị trung bình, và do đó theo các giả thiết này, lỗi cực đại sau khi điều chỉnh đồng hồ của i theo đồng hồ của j với xác suất 99% là nhỏ hơn $2,3 \times 2 \times \sigma \mu s$.

Đồng bộ toàn mạng

Với khả năng thực hiện các đồng bộ theo cặp (pair-wise), LTS kế tiếp giải quyết nhiệm vụ đồng bộ tất cả các nút của một mạng cảm biến (được kết nối) với một nút tham chiếu. Nếu một nút cụ thể i có một khoảng cách là h_i bước nhảy tới nút tham chiếu, và nếu lỗi đồng bộ được phân phối một cách thông thường với các tham số $\mu = 0$ và $\sigma' = 2\sigma$ tại mỗi bước nhảy, và nếu thêm vào đó các bước nhảy là độc lập thì lỗi đồng bộ của i cũng được phân phối một cách thông thường với độ lệch $\sigma_i^2 = 4h_i\sigma^2$. Trên cơ sở của sự quan sát này, LTS nhắm tới xây dựng một **cây bao trùm (spanning tree)** với chiều dài nhỏ nhất và chỉ các cặp nút dọc theo rìa của cây được đồng bộ. Nếu quá trình đồng bộ dọc theo cây bao trùm mất nhiều thời gian, sự trượt giữa các đồng hồ sẽ tạo thêm các lỗi.

Người ta đề xuất hai biến khác nhau, một biến tập trung và một biến phân tán.

LTS đa bước nhảy tập trung

Nút tham chiếu, ví dụ: một nút với một bộ thu GPS hoặc tham chiếu thời gian chất lượng cao khác – xây dựng một cây bao trùm T và bắt đầu đồng bộ: đầu tiên nút tham chiếu đồng bộ với nút con của nó tại T , sau đó nút con này đồng bộ với nút con của nó, và cứ như vậy cho đến hết. Suy ra, mỗi nút phải biết con của nó. Có vài thuật toán đã có sẵn cho việc xây dựng phân tán của một cây bao trùm; đối với LTS thì có hai thuật toán cụ thể được thảo luận trong tài liệu tham khảo [16], và chúng có tên là thuật toán nghiên cứu độ sâu đầu tiên (DDFS - Distributed Depth-First Search) và thuật toán tiếng vọng.

Nút tham chiếu cũng phải quan tâm đến việc tái đồng bộ tần số để bù lại độ lệch. Người ta giả thiết rằng nút tham chiếu biết bốn tham số: chiều cao tối đa h của cây bao trùm, độ lệch cực đại ρ sao cho phương trình 8.1 được thỏa mãn cho tất cả các nút trong mạng, độ lệch tiêu chuẩn cho bước nhảy đơn là 2σ (được thảo luận ở trên), và độ chính xác mong muốn δ . Mục đích là luôn luôn có một lỗi đồng bộ của các nút lá nhỏ hơn δ với xác suất 99%. Ngay lập tức, sau khi tái đồng bộ thì sự chính xác của nút lá nhỏ hơn $h \cdot 2,3 \cdot 2 \cdot \sigma$ và nó được phép phát triển gần tới mức δ . Với tốc độ lệch cực đại ρ , sự phát triển này mất $\delta - 2,3 \cdot h \cdot \sigma \cdot \rho$ thời gian. Sự lựa chọn thực tế của chu kỳ đồng bộ phải nhỏ hơn một chút so với độ lệch xuất hiện trong một sự tái đồng bộ đơn lẻ, có thể làm ảnh hưởng đến độ chính xác ban đầu $2,3 \cdot 2 \cdot h \cdot \sigma$. Một vấn đề then chốt đó là các giá thành của việc truyền thông. Một sự đồng bộ cặp đơn (single pairwise synchronization) có giá tương đương 3 gói tin, và do đó đồng bộ một mạng có n nút mất $3n$ gói tin, không tính đến các lỗi kênh hay các xung đột. Thêm vào đó, cần năng lượng đáng kể để xây dựng cây bao trùm, và người ta giả thiết là lặp lại cấu trúc này trên mỗi vòng đồng bộ để nhận được một dung sai lỗi nào đó. Đối với các lý do về lỗi đồng bộ, việc có nhiều nút tham chiếu cũng có lợi vì: nếu một trong số các nút bị hỏng hoặc nếu mạng trở nên bị chia tách, thì một nút khác sẽ đảm nhiệm. Một giao thức lựa chọn nút lãnh đạo là hữu ích để hỗ trợ các nút tham chiếu động.

LTS đa bước nhảy phân tán

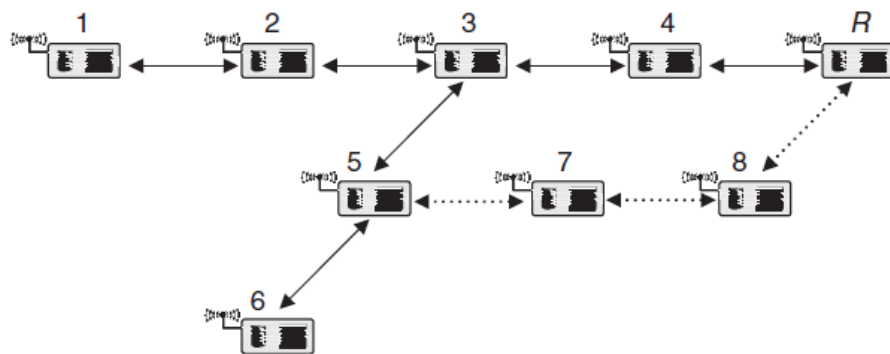
Biến thứ hai là giao thức LTS đa bước nhảy phân tán. Không có cây bao trùm nào được xây dựng, nhưng mỗi nút biết các nhận dạng của một lượng lớn các nút tham chiếu dọc theo các tuyến tới các nút đó. Đó là trách nhiệm của các nút để bắt đầu việc tái đồng bộ theo chu kỳ. Đề cập đến tình huống trình bày ở hình 8.3 và giả thuyết rằng nút 1 muốn đồng bộ với nút tham chiếu R. Nút 1 đưa ra một yêu cầu đồng bộ hướng tới R, mà kết quả là một chuỗi các đồng bộ theo cặp: nút 4 đồng bộ với nút R, nút 3 đồng bộ với nút 4, và cứ tiếp tục như vậy cho đến khi tiến tới nút 1. Có hai cái đáng chú ý:

- Như một sản phẩm phụ, nút 2,3, và 4 cũng nhận được đồng bộ với nút R.
- Cho cùng yêu cầu độ chính xác δ và cùng tốc độ dịch ρ cho tất cả các nút, tần số tái đồng bộ cho một nút i mà nhảy h_i bước khỏi nút tham chiếu được cho bởi $\frac{\delta - 4,2,3 \cdot h_i \cdot \sigma}{\rho}$. Do đó, trong hình vẽ, nút 1 và nút 6 có chu kỳ tái đồng bộ ngắn nhất.

Nếu hai nút này luôn yêu cầu tái đồng bộ với nút R, các nút trung gian 2,3,4, và 5 sẽ không bao giờ tự có yêu cầu tái đồng bộ.

Một nút sẽ chọn nút tham chiếu gần nhất để tối thiểu hóa lỗi đồng bộ của nó. Theo cách này, một cây có trọng số nhỏ nhất không được xây dựng một cách rõ ràng, nhưng nó lãnh

trách nhiệm của giao thức định tuyến để tìm ra các đường tốt nhất. Đối việc thiết lập mạng và các giao thức định tuyến nào đó, việc đưa ra các chu kỳ định tuyến/đồng bộ có thể tăng lên, và điều này cần phải tránh. Từ ví dụ trên đây, nếu một nút có thể lợi dụng việc đồng bộ đang diễn ra thì cũng có lợi. Ví dụ nút 5 trong hình 8.3. Thay vì đồng bộ nút 5 một cách độc lập với nút tham chiếu R (nút mà làm cho nút 3 và 4 xử lý hai yêu cầu đồng bộ một cách đồng thời), nút 5 có thể hỏi tất cả các nút trong khu vực lân cận của nó về các yêu cầu đồng bộ đang diễn ra. Nếu có yêu cầu đồng bộ nào đó thì nút 5 có thể đợi trong một khoảng thời gian nào đó và sau đó cố gắng đồng bộ với bên phát đáp (responder). Một tối ưu hóa khác của LTS cũng được giải thích trong hình 8.3, sử dụng các đường đứt nét. Lại giả thuyết rằng nút 5 muốn đồng bộ. Như giải thích ở trên, một tùy chọn là để nút 5 tham gia vào một yêu cầu đồng bộ đang diễn ra tại nút 3. Mặt khác, việc giữ các nút 7 và 8 cũng được đồng bộ với R có thể là cần thiết. Để có được điều này, nút 5 có thể đưa ra yêu cầu của nó thông qua các nút 7 và 8 tới R và đồng bộ các bước nhảy trung gian như một kết quả phụ. Điều này gọi là **đa dạng hóa đường truyền** (path diversification). Các đặc điểm của các biến LTS được nghiên cứu bằng cách mô phỏng 500 nút phân bố ngẫu nhiên trong phạm vi 120m (120m chữ nhật, mỗi nút có một phạm vi truyền dẫn là 10m). Nút tham chiếu đơn được đặt ở trung tâm. Nó cho thấy rằng LTS đa bước nhảy phân tán có giá thành cao hơn do trao đổi các gói tin khi tất cả các nút của một mạng phải được đồng bộ (giữa 40 và 100% tổng chi phí của thuật toán trung tâm), thậm chí khi các tối ưu hóa như sự đa dạng đường truyền hoặc sự tham gia vào các đồng bộ đang diễn ra được triển khai (giảm giá thành xuống 15 đến 60%). Tuy nhiên, nếu chỉ một phần của các nút phải được đồng bộ, các thuật toán phân tán có thể giới hạn chi phí này theo tập hợp lợi ích và bảo tồn năng lượng của tất cả các nút khác trong khi các thuật toán tập trung luôn bao gồm tất cả các nút và do đó có các giá thành cố định bất chấp việc đồng bộ thời gian hiện tại có được yêu cầu hay không. LTS đa bước nhảy phân tán có ưu điểm là nó có thể là đồng bộ phía sau (post-facto synchronization). Nút i có thể tự quyết định khi nào thì đồng bộ với các nút lân cận gần với nút tham chiếu.



Hình 8.3 LTS đa bước nhảy phân tán

8.2.2 Làm thế nào để tăng độ chính xác và ước lượng độ lệch

Chúng ta sử dụng giao thức đồng bộ theo cặp của LTS để giải thích bằng cách nào lỗi đồng bộ giữa các nút có thể giảm xuống và bằng cách nào độ lệch của đồng hồ của nút x đối với đồng hồ của nút tham chiếu R có thể được ước lượng. Cả hai đều có thể biến thành các vấn đề đánh giá tiêu chuẩn.

Tăng độ chính xác tuyệt đối

Giả thiết rằng độ lệch giữa x và R là không đáng kể đối với một thời điểm nào đó và nút x muốn ước lượng bù pha đối với đồng hồ của nút R . Nút x có thể tăng độ chính xác của ước lượng của nó bằng cách lặp lại việc trao đổi gói tin, và thu được các ước lượng $O(t_0), O(t_1), \dots, O(t_{n-1})$. Cho A là bù pha ban đầu tại thời điểm t_0 và chúng ta giả thiết rằng các lỗi đồng bộ được quan sát tại các thời điểm khác nhau t_0, \dots, t_{n-1} là độc lập. Do đó một người có thể đưa ra mô hình của mỗi $O(t_k)$ như sau:

$$O(t_k) = A + w(t_k)$$

trong đó $w(t_k)$ được lấy mẫu từ một quá trình nhiễu Gaussian trắng với trung bình bằng 0 và độ lệch tiêu chuẩn $\sigma_x = 2\sigma$, nghĩa là tất cả các $w(t_k)$ đều độc lập. Do đó,

$$\hat{A} = \frac{1}{n} \sum_{k=0}^{n-1} O(t_k)$$

là một ước lượng có độ lệch nhỏ nhất không phân cực đối với A , và độ lệch của nó là $\frac{\sigma_x^2}{n}$,

giảm tuyến tính theo n . Do đó, các tỉ lệ độ lệch của ước lượng với n là $O(n^{-1})$.

Nếu nút đáp lại một gói tin yêu cầu gửi câu trả lời của nó nhanh nhất có thể, thì người ta có thể sử dụng một phương pháp khác tạo ra ít giả thiết hơn về các quan sát và nhiễu. Khi xem hình 8.2, chúng ta có thể quan sát thấy như sau: khi nút i làm quan sát thứ k gọi là O_k , nó đo độ lệch thời gian giữa các thời điểm $t_{1,k}$ và $t_{8,k}$, nghĩa là các thời điểm giữa các lần dán nhãn thời gian của gói tin yêu cầu và gói tin trả lời. Rõ ràng là, việc quan sát k với độ lệch nhỏ nhất $t_{8,k} - t_{1,k}$ có sự không chắc chắn nhỏ nhất và do đó là chính xác nhất.

Ước lượng độ lệch

Rõ ràng là người ta có thể ước lượng độ lệch từ chỉ một tái đồng bộ. Do đó, cho $O(t_0), O(t_1), \dots, O(t_{n-1})$ là các bù ước lượng nút x nhận được cho các tái đồng bộ được tiến hành tại các thời điểm t_0, t_1, \dots, t_{n-1} . Chúng ta giả thiết rằng độ lệch ρ_x của nút x là không đổi trong khoảng thời gian $[t_0, t_{n-1}]$. Chúng ta đầu tiên quan tâm đến trường hợp giao thức đồng bộ theo cặp (pair-wise) của phần 8.2.1 chạy một cách lặp lại nhưng nút x không tái điều chỉnh đồng hồ của nó sau khi tiến hành một quan sát. Trong trường hợp này, chúng ta có thể viết $O(t_i)$ như sau: $O(t_i) = A + \rho_x(t_i - t') + w(t_i)$ trong đó A là sự bù chính xác giữa hai đồng hồ tại một thời điểm tham chiếu t' và $w(t_0), w(t_1), \dots, w(t_{n-1})$ là nhiễu Gaussian trắng. Từ thảo luận ở trên, giả thiết nhiễu Gaussian trắng là tương thích với các giả thiết được thực hiện cho LTS. Để đơn giản hóa, chúng ta giả thiết rằng $t' = 0$. Bây giờ có ít nhất hai phương pháp để ước lượng cùng nhau A và ρ_x :

- Chúng ta có thể chọn độ lệch thời gian $t_1 - t_0$ lớn đến nỗi mà độ lớn của $w(t_0)$ và $w(t_1)$ là giống đáng kể so với $\rho_x(t_1 - t_0)$. Suy ra,

$$O(t_1) - O(t_0) = A + \rho_x(t_1 - t') + w(t_1) - A - \rho_x(t_0 - t') - w(t_0) \approx \rho_x(t_1 - t_0)$$

mà cho chúng ta ρ_x từ các quan sát $O(t_0)$ và $O(t_1)$. Trong bước thứ hai, chúng ta có thể rút ra $\rho_x(t_1 - t')$ từ $O(t_1)$ để nhận được một ước lượng của A .

- Nếu có sẵn nhiều ước lượng, chúng ta có khái niệm về vấn đề về sự điều chỉnh đường (line fitting) với dữ liệu bị nhiễu. Hàm mật độ xác suất của việc thực hiện các

quan sát $O(t_0), \dots, O(t_{n-1})$ cho các tham số A và ρ_x giả thiết là đã biết, có thể được viết như sau:

$$\begin{aligned} p(O(t_1), \dots, O(t_{n-1}); A, \rho_x) &= \prod_{i=0}^{n-1} \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{(O(t_i) - A - \rho_x t_i)^2}{2\sigma_x^2}\right) \\ &= (2\pi\sigma_x^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma_x^2} \sum_{i=0}^{n-1} (O(t_i) - A - \rho_x t_i)^2\right) \end{aligned}$$

Kay [414, Chương 3] chỉ ra rằng ước lượng không phân cực độ lệch nhỏ nhất cho loại vấn đề ước lượng này được cho bởi:

$$\begin{aligned} \hat{A} &= \frac{2(2n-1)}{n(n+1)} \sum_{k=0}^{n-1} O(t_k) - \frac{6}{n(n+1)} \sum_{k=0}^{n-1} k \cdot O(t_k) \\ \hat{\rho}_x &= \frac{-6}{n(n+1)} \sum_{k=0}^{n-1} O(t_k) + \frac{12}{n(n^2-1)} \sum_{k=0}^{n-1} k \cdot O(t_k) \end{aligned}$$

(ước lượng bình phương nhỏ nhất là như nhau). Cần nhiều kỹ thuật ước lượng tiên tiến hơn khi $w(t_i)$ được biết là không kết hợp (correlated) hoặc không là hàm Gaussian (non-Gaussian).

Trường hợp quan tâm khác là khi đồng hồ của nút x được tái điều chỉnh theo mỗi lần tái đồng bộ bởi sự bù ước lượng $O(t_k)$. Theo sau đây, chúng ta điều khiển một ước lượng giống nhau tối đa (maximum-likelihood) đơn giản cho tốc độ lệch ρ_x . Theo mô hình dữ liệu của chúng ta từ ở trên, đối với quan sát đầu tiên $O(t_0)$ chúng ta có:

$$O(t_0) = A + \rho_x(t_0 - t') + w(t_0)$$

và ngay sau quan sát này nút x sửa lỗi đồng hồ của nó bằng cách trừ đi $O(t_0)$. Sự bù pha “mới” $A(t_0)$ ngay sau khi sửa lỗi đồng hồ chỉ là $-w(t_0)$. Từ giờ trở đi, chúng ta không phải quan tâm đến sự bù đầu tiên $A = A(t')$ nữa. Đối với $O(t_1)$ ta có:

$$O(t_1) = A(t_0) + \rho_x(t_1 - t_0) + w(t_1) = \rho_x(t_1 - t_0) + w(t_1) - w(t_0)$$

Và sự bù pha mới ngay sau khi sửa lỗi là $A(t_1) = -w(t_1)$. Bằng cách lặp lại đối số này, chúng ta nhận được

$$\begin{aligned} O(t_k) &= A(t_{k-1}) + \rho_x(t_k - t_{k-1}) + w(t_k) = \rho_x(t_k - t_{k-1}) + w(t_k) - w(t_{k-1}) \\ &= \rho_x(t_k - t_{k-1}) + w_*(t_k) \end{aligned}$$

Trong đó $w_*(t_k)$ là các biến ngẫu nhiên Gaussian độc lập với trung bình bằng 0 và độ lệch $\sigma_*^2 = 2\sigma_x^2$. Sự giống nhau tính theo logarit (log-likelihood) của thực hiện các quan sát $O(t_1), \dots, O(t_{n-1})$ cho ρ_x đã cho được cho bởi

$$\begin{aligned} L(\rho_x) &= \log \left[\prod_{k=1}^{n-1} \frac{1}{\sqrt{2\pi\sigma_*^2}} \exp\left(-\frac{1}{2\sigma_*^2} (O(t_k) - \rho_x(t_k - t_{k-1}))^2\right) \right] \\ &= -\frac{n}{2} \log(2\pi\sigma_*^2) + \frac{1}{2\sigma_*^2} \sum_{k=1}^{n-1} (O(t_k) - \rho_x(t_k - t_{k-1}))^2 \end{aligned}$$

Đặt tích phân $\frac{d}{d\rho_x} L(\rho_x)$ bằng 0 và giải cho ρ_x ta có:

$$\rho_x = \frac{\sum_{k=1}^{n-1} O(t_k)(t_k - t_{k-1})}{\sum_{k=1}^{n-1} (t_k - t_{k-1})^2}$$

8.2.3 Giao thức đồng bộ thời gian cho các mạng cảm biến (TPSN)

Giao thức đồng bộ thời gian cho các mạng cảm biến (TPSN) [271, 272] là một giao thức dựa trên máy phát/máy thu thụ vị khác. Một lần nữa, đầu tiên chúng ta giải thích phương pháp đồng bộ theo cặp trước khi chuyển sang trường hợp đa bước nhảy.

Đồng bộ theo cặp (pair-wise)

Giao thức đồng bộ theo cặp (pair-wise) của TPSN có một số điểm chung so với LTS. Nó hoạt động theo cách không đối xứng: nút i đồng bộ với đồng hồ của nút j khác nhưng không ngược lại (Hình 8.4). Hoạt động diễn ra như sau:

- Nút i khởi động việc tái đồng bộ tại thời điểm t_0 . Nó định dạng một **gói tin xung đồng bộ** và chuyển giao nó cho hệ điều hành và ngăn xếp mạng tại thời điểm t_1 .
- Ngăn xếp mạng thực hiện giao thức MAC và xác định một thời gian phù hợp cho việc truyền dẫn gói tin, là thời điểm t_2 . Ngay trước khi truyền dẫn, gói tin được dán nhãn thời gian là $L_i(t_2)$. Bằng cách dán nhãn thời gian gói tin ngay trước khi truyền dẫn và không phải ngay khi gói tin được định dạng trong lớp ứng dụng, hai nguồn của sự không chắc chắn bị loại bỏ, đó là hoạt động của ngăn xếp hệ thống/mạng và trễ truy cập đường truyền. Sự không chắc chắn còn lại là khoảng thời gian nhỏ giữa việc dán nhãn gói tin và sự bắt đầu thực sự việc truyền dẫn nó. Trễ này được tạo ra, ví dụ, là do việc cần thiết tái tính toán tổng kiểm tra (kiểm tra tổng) của gói tin ngay trước khi gửi nó.
- Sau thời gian trễ truyền lan và thời gian truyền dẫn gói tin, bit cuối cùng đến máy thu tại thời điểm t_3 , và đôi khi sau đó ngắt nhận gói tin được kích hoạt, đó là tại thời điểm t_4 . Máy thu dán nhãn thời gian gói tin đã ở trong thường trình ngắt (interrupt routine) với $L_j(t_4)$.
- Nút j định dạng một gói tin xác nhận (**acknowledgement packet**) và chuyển giao nó tại thời điểm t_5 tới hệ điều hành và ngăn xếp mạng. Một lần nữa, ngăn xếp mạng thực thi giao thức MAC và gửi gói tin tại thời điểm t_6 . Ngay trước khi truyền dẫn, gói tin được dán nhãn với $L_j(t_6)$, và gói tin cũng mang các nhãn thời gian khác $L_i(t_2)$ and $L_j(t_4)$.
- Cuối cùng thì nút i dán nhãn thời gian gói tin xác nhận đang đến càng sớm càng tốt với $L_i(t_7)$.

Nếu O là sự bù thực sự của đồng hồ nút i so với đồng hồ j , chúng ta có gói tin xung đồng bộ $L_j(t_4) = L_i(t_2) + O + \tau + t_p + \delta_{s,i} + \delta_{r,j}$, trong đó τ là trễ truyền lan, t_p là thời gian truyền dẫn gói tin (chiều dài gói tin được chia ra bởi tốc độ bit của kênh truyền), $\delta_{s,i}$ là sự không chắc chắn (nhỏ) tại bên phát giữa thời điểm dán nhãn thời gian và thời điểm bắt đầu thực sự của việc truyền dẫn, và $\delta_{r,j}$ là sự không chắc chắn của máy thu tại thời điểm j . Theo hướng khác, đối với gói tin xác nhận, chúng ta có

$$L_i(t_7) = L_j(t_6) - O + \tau + t_p + \delta_{s,j} + \delta_{r,i}$$

(giả thiết rằng gói tin xác nhận và gói tin xung đồng bộ có cùng chiều dài và trễ truyền lan là giống nhau ở cả hai hướng). Bây giờ

$$\begin{aligned} & (L_j(t_4) - L_i(t_2)) - (L_i(t_7) - L_j(t_6)) \\ &= (O + \tau + t_p + \delta_{s,i} + \delta_{r,j}) - (-O + \tau + t_p + \delta_{s,j} + \delta_{r,i}) \\ &= 2 \cdot O + (\delta_{s,i} - \delta_{s,j}) + (\delta_{r,j} - \delta_{r,i}). \end{aligned}$$

Do đó,

$$O = \frac{(L_j(t_4) - L_i(t_2)) - (L_i(t_7) - L_j(t_6))}{2} - \frac{\delta_{s,i} - \delta_{s,j}}{2} - \frac{\delta_{r,j} - \delta_{r,i}}{2}$$

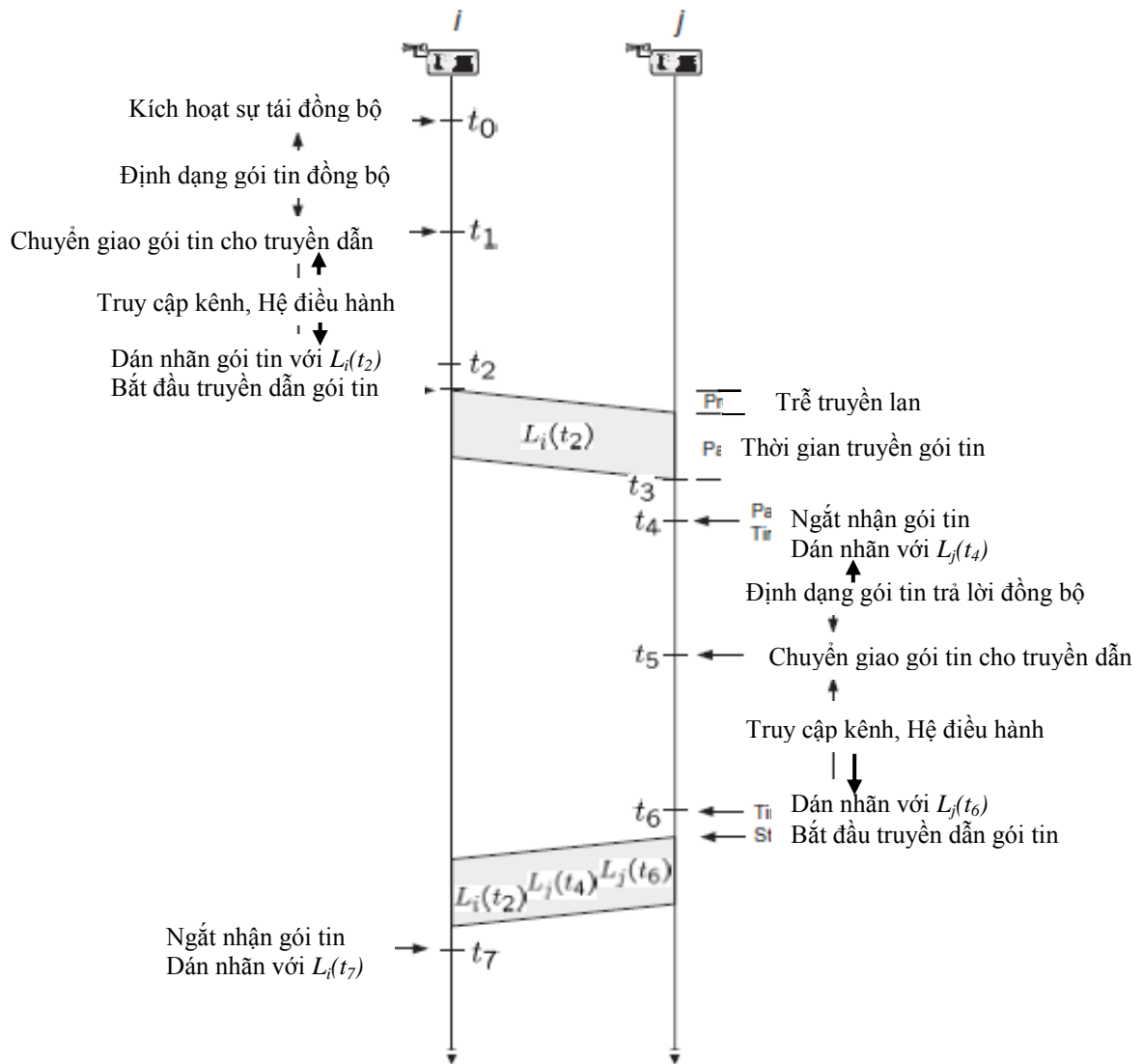
Đặc điểm chủ chốt của phương pháp này đó là nút i dán nhãn thời gian gói tin đi ra càng muộn càng tốt và nút j dán nhãn thời gian gói tin đi vào càng sớm càng tốt. Các yêu cầu này hỗ trợ từ lớp MAC, lớp MAC là dễ dàng có được trong các nút cảm biến hơn là với phần cứng thương mại như các card giao tiếp mạng IEEE 802.11. Hiệu suất tương đối của TPSN so với giao thức RBS được thảo luận trong phần 8.3.1.

Đồng bộ toàn mạng

Thuật toán đồng bộ toàn mạng của TPSN xây dựng chủ yếu một **cây bao trùm** (spanning tree), ở đó mỗi nút biết vị trí của nó trong cây và nhận dạng của nút mẹ của nút đó. Nút gốc (**root node**) được chỉ định ở vị trí 0 và trách nhiệm của nó là kích hoạt cấu trúc của cây. Tất cả các nút có thể tới được trong mạng đồng bộ với nút gốc. Nếu nút gốc truy cập vào một hệ thời gian mở rộng chính xác giống như UTC, thì do đó tất cả các nút đồng bộ với UTC.

Giao thức hoạt động như sau. Để bắt đầu cấu trúc cây, nút gốc gửi một gói tin *phát hiện mức* (vị trí) chứa mức 0 của nó. Tất cả các nút lân cận một bước nhảy của nút gốc tự ấn định cho mình một mức bằng một cộng với mức được chỉ ra trong gói tin phát hiện mức nhận được và chấp nhận nút gốc như mẹ của chúng. Sau đó, các nút mức 1 gửi đi các gói tin *phát hiện mức* mức 1 của bản thân chúng và cứ tiếp tục như vậy. Các nút mức 1 chọn một trễ ngẫu nhiên để tránh các xung đột MAC quá đáng. Mỗi khi một nút nhận được một gói tin *phát hiện mức*, nút khởi tạo ra gói tin được chấp nhận là nút mẹ và tất cả các gói tin tiếp theo sau bị bỏ đi. Sau khi một nút tìm thấy một nút mẹ, nó tái đồng bộ theo chu kỳ theo đồng hồ của nút mẹ. Một nút có thể nhận trượt các gói *phát hiện mức* bởi vì các xung đột MAC hoặc bởi vì nó được triển khai sau cấu trúc cây đầu tiên. Nếu một nút i không nhận được bất cứ gói tin *phát hiện mức* nào trong một khoảng thời gian nào đó, nó hỏi nút lân cận một bước nhảy của nó về một cây đang tồn tại bằng cách đưa ra một gói tin *yêu cầu mức*. Các nút lân cận trả lời bằng cách gửi mức của bản thân chúng. Nút i lựa chọn các câu trả lời từ cửa sổ thời gian nào đó và chọn nút lân cận với mức nhỏ nhất làm mẹ của nó.

Sự duy trì cây được tích hợp với việc tái đồng bộ. Để tính độ lệch (sự trượt), một nút i phải chạy thuật toán theo cặp (pair-wise) với nút mẹ của nó theo chu kỳ. Nếu điều đó không thực hiện được nhiều lần liên tiếp, thì nút i kết luận rằng nút mẹ của nó đã bị dời đi hoặc đã chết. Nếu mức của i là mức 2 hoặc lớn hơn, nó gửi một gói tin *yêu cầu mức*, thu thập các câu trả lời trong một khoảng thời gian nào đó và tự ấn định cho bản thân một mức mới từ gói tin trả lời mức thấp nhất. Nếu i là ở mức một, thì nó kết luận rằng nút gốc đã chết. Có nhiều cách để giải quyết tình trạng này. Một trong số chúng là chạy một giao thức lựa chọn nút lãnh đạo giữa các nút mức 1.



Hình 8.4 Sự đồng bộ máy phát/máy thu trong TPSN

Phương pháp này có các đặc điểm sau:

- Cây bao trùm thu được không nhất thiết là một cây tối thiểu, bởi vì các xung đột MAC và các trễ ngẫu nhiên có thể dẫn đến một tình trạng, ở đó đầu tiên một nút nhận được một *phát hiện mức* từ một nút mức cao hơn. Tuy nhiên, có một sự không dung hòa giữa độ chính xác đồng bộ (các đường dài hơn ngụ ý lỗi trung bình nhiều hơn) và tổng chi phí cho cấu trúc cây. Các thuật toán để tìm các cây bao trùm nhỏ nhất là phức tạp hơn.
- Nếu hai nút i và j là gần nhau về mặt vật lý và nhận cùng mức “*phát hiện mức v* ” trong pha thiết lập cây, cả hai tự ấn định cho mình mức $v + 1$ và cố gắng gửi lại gói tin *phát hiện mức*. Một trong hai nút này sẽ chiến thắng trong cuộc ganh đua. Bởi vì cả hai đều ở gần nhau, các nút lân cận một bước nhảy của chúng là gần giống nhau. Kết quả là tất cả các nút lân cận đến thời điểm đó chưa đồng bộ sẽ chấp nhận nút i như là nút mẹ của chúng và tạo ra tái tái đồng bộ đáng kể cho nút i , trong đó nút j gần như không bị mất năng lượng bởi vì nó không có nút con nào. Để tránh sự

không công bằng, cấu trúc cây phải được lập lại theo chu kỳ, điều này lần lượt tạo ra tải của mạng.

- Độ lớn trung bình của lỗi đồng bộ giữa một nút mức v và nút gốc tăng lên dần dần theo v . Đối với một bước nhảy, lỗi đồng bộ trung bình là $\approx 17 \mu s$ và đối với 5 bước nhảy là $\approx 23 \mu s$.
- Người ta có thể nhận được đồng bộ phía sau (post-facto). Trong trường hợp này, không có cây bao trùm nào được xây dựng. Xem xét một tiêu chí trong đó một nút i_0 muốn thông tin một sự kiện (mà xảy ra tại thời điểm t) tới một nút i_n khác thông qua nhiều bước nhảy trung gian i_1, i_2, \dots, i_{n-1} . Nút i_0 gửi gói tin cùng với nhãn thời gian cục bộ $L_{i_0}(t)$ của nó tới i_1 . Sau đó, nút i_1 đồng bộ đồng hồ của nó với đồng hồ của nút i_0 và chuyển tiếp gói tin đến nút i_2 , và cứ tiếp tục như vậy. Cuối cùng thì tất cả các nút bao gồm nút i_n đồng bộ với nút i_0 và i_n có gói tin với nhãn thời gian $L_{i_0}(t)$ và do đó có thể quyết định về tuổi của sự kiện.

8.3 Các giao thức dựa trên sự đồng bộ máy thu/máy thu

Trong đồng bộ dựa trên máy phát/máy thu, máy thu của một gói tin được dán nhãn thời gian đồng bộ với máy phát của gói tin. Trong các phương pháp đồng bộ đồng bộ máy thu/máy thu, nhiều máy thu của cùng gói tin được dán nhãn thời gian đồng bộ với nhau, nhưng không đồng bộ với máy phát. Chúng ta thảo luận hai giao thức thuộc về loại này.

8.3.1 Sự đồng bộ quảng bá tham khảo (RBS - Reference broadcast synchronization)

Giao thức **đồng bộ quảng bá tham khảo (RBS)** [11] bao gồm hai thành phần. Trong thành phần thứ nhất, một tập hợp các nút trong một miền quảng bá đơn lẻ (nghĩa là một tập hợp các nút mà có thể nghe thấy nhau) ước lượng các đồng hồ của các nút ngang hàng của chúng. Thành phần thứ hai cho phép liên kết các nhãn thời gian giữa các nút ở xa với các miền quảng bá giữa chúng. Chúng ta lần lượt giải thích mỗi thành phần.

8.3.1.1 Sự đồng bộ trong một miền quảng bá

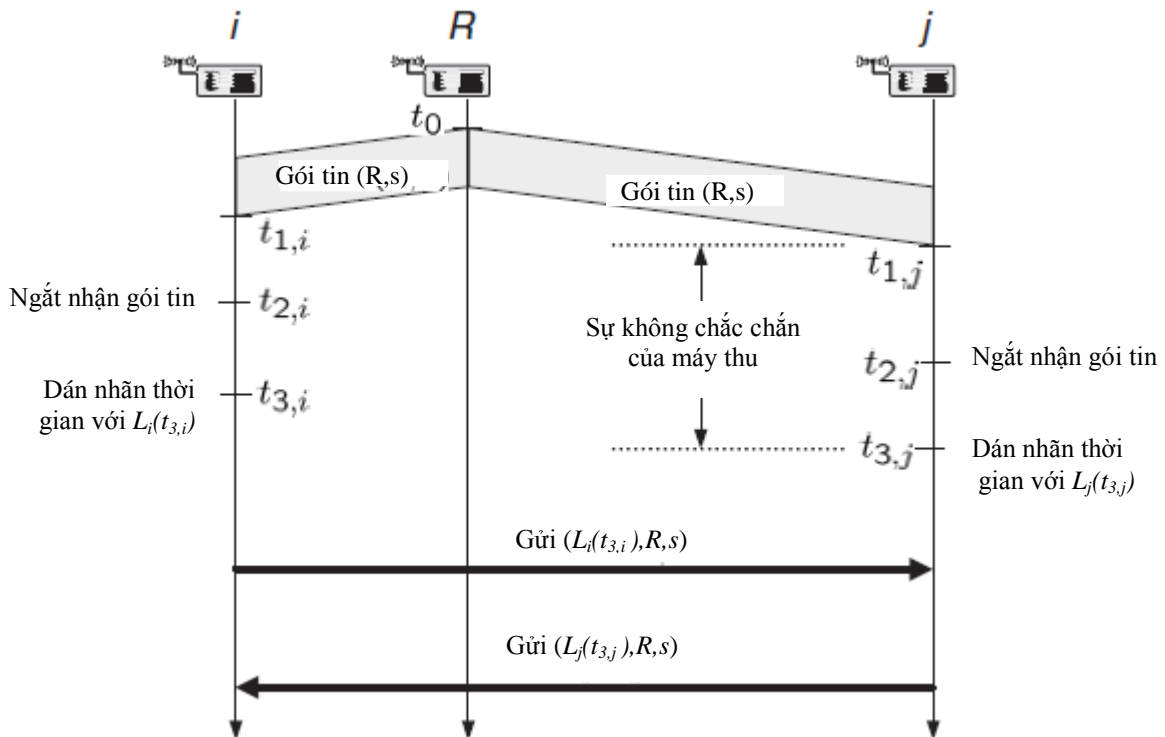
Ý tưởng cơ bản như sau: Một máy phát gửi một gói tin theo chu kỳ tới một kênh quảng bá và tất cả các máy thu dán nhãn thời gian gói tin này. Các máy thu trao đổi các nhãn của chúng và có thể sử dụng dữ liệu này để biết về các đồng hồ của nút lân cận của chúng. Bằng cách lặp lại quá trình này, các nút có thể không chỉ biết về sự bù pha lẫn nhau của chúng mà còn biết về tốc độ lệch của chúng. Các nút không điều chỉnh các đồng hồ cục bộ của chúng nhưng xây dựng một bảng, bảng này lưu các tham số cần thiết để chuyển đổi các giá trị đồng hồ cho mỗi nút lân cận.

Ví dụ:

Một ví dụ được trình bày ở hình 8.5. Hai nút i và j muốn đồng bộ. Tại thời điểm t_0 , nút R khác phát quảng bá một **gói tin xung** (pulse packet), gói tin này bao gồm nhận dạng R của nó và một chuỗi số s . (chú ý: gói tin xung không cần thiết phải là một gói tin đồng bộ thời gian dành riêng, chỉ cần là các gói tin dữ liệu bình thường mang các số chuỗi là đủ). Bởi vì các nút i và j có khoảng cách khác nhau so với nút R , các trễ truyền lan là khác nhau: bit cuối cùng tới nút i tại thời gian thực $t_{1,i}$ và nút j tại thời gian thực $t_{1,j}$, và các trễ truyền lan là τ_i and τ_j tương ứng. Trong hai nút, một ngắt thu gói tin được tạo ra, tại các thời điểm $t_{2,i}$ và $t_{2,j}$ cho các nút i và j . Một thời gian ngắn sau đó, tại thời điểm $t_{3,i}$, nút i dán nhãn thời gian gói tin với

nhãn thời gian cục bộ của nó là $L_i(t_{3,i})$; nút j cũng làm tương tự, và tạo ra một nhãn thời gian $L_j(t_{3,j})$. Sau đó, nút i và j trao đổi các nhãn thời gian của chúng và nhận dạng (địa chỉ máy phát, số chuỗi – sequence number) của gói tin xung tương ứng. Cả hai nút có thể tính toán dễ dàng các dịch pha tương đối của các đồng hồ của chúng bằng cách giả thiết $t_{3,i} = t_{3,j}$. Đặc biệt là nút i lưu giá trị $O(t_{3,i}) = L_i(t_{3,i}) - L_j(t_{3,j})$ như sự bù pha trong một bảng cục bộ mà không phải tái điều chỉnh đồng hồ của nó. Rõ ràng là, phương pháp này có thể có lợi từ việc có ở máy thu các gói tin dán nhãn thời gian đang đến càng nhanh càng tốt trong trường hợp ngắt.

Chấp nhận thuật ngữ được giới thiệu cho TPSN, thời gian giữa việc nhận bit cuối cùng và việc dán nhãn thời gian gói tin được gọi là sự không chắc chắn của máy thu. Nó được biểu thị bởi $\delta_{r,i}$ and $\delta_{r,j}$ cho các nút i và j tương ứng.



Hình 8.5 Ví dụ RBS: hai nút i và j và một máy phát R

Độ chính xác có thể nhận được cho một xung đơn

Cuối cùng thì tính toán tổng quát là chính xác tuyệt đối khi giả thiết $t_{3,i} = t_{3,j}$ là thực sự đúng. Chúng ta hãy phân tích tóm tắt các nguyên nhân của các lỗi đồng bộ có thể.

- *Trễ truyền lan*: trong một mạng cảm biến, miền phát quảng bá là khá nhỏ và trễ truyền lan của gói tin là không đáng kể. Ngoài ra, nó chỉ là *sự khác nhau (difference)* trong trễ truyền lan giữa i và j và sự khác biệt này có xu hướng nhỏ hơn nữa.
- Trễ giữa việc nhận bit cuối và tạo ra ngắt thu gói tin (packet reception interrupt): điều này có thể là do các trễ xử lý phần cứng (như các tính toán tổng kiểm tra - kiểm tra tổng) và cũng chặn các ngắt trong một thời gian ngắn trong trường hợp các phần chủ chốt hoặc các ngắt dịch vụ có độ ưu tiên cao hơn. Một lần nữa, đó là sự khác nhau (*difference*) giữa i và j , sự khác nhau này dùng để tính các lỗi đồng bộ.

- Trễ giữa việc nhận ngắt gói tin và việc dán nhãn thời gian gói tin: nếu nhãn thời gian đã được tạo ra trong thường trình ngắt, trễ này là nhỏ.
- Độ lệch giữa việc dán nhãn thời gian và việc trao đổi các nhãn thời gian quan sát được cũng góp phần vào các lỗi đồng bộ. Càng nhiều thời gian trôi qua, thì lỗi này sẽ là càng lớn.

So sánh với các phương pháp dựa trên máy phát/máy thu, thời gian yêu cầu đối với R để định dạng gói tin của nó, di chuyển nó qua hệ điều hành và phần mềm mạng của nó, cũng như trễ truy cập đường truyền là hoàn toàn không liên quan bởi vì điểm chung tham chiếu cho i và j là thời điểm t_0 mà gói tin xuất hiện trên đường truyền.

So sánh giữa RBS và TPSN

Ganeriwai [13] so sánh RBS và TPSN. Cả hai giao thức dán nhãn thời gian các gói tin vừa nhận được trong ngắt máy thu.

Như được so sánh với các giao thức dựa trên máy phát/máy thu khác, TPSN loại bỏ nhiều sự không chắc chắn tại máy phát (hệ điều hành và ngăn xếp mạng, trễ truy cập đường truyền) bằng cách dán nhãn thời gian một gói tin sắp đi ra ngay trước khi truyền dẫn. Đối với trường hợp TPSN, chúng ta bắt nguồn từ phương trình

$$O = \frac{(L_j(t_4) - L_i(t_2)) - (L_i(t_7) - L_j(t_6))}{2} - \frac{\delta_{s,i} - \delta_{s,j}}{2} - \frac{\delta_{r,j} - \delta_{r,i}}{2}$$

trong đó $L_x(t_y)$ là các nhãn thời gian và $\delta_{s,i}$ and $\delta_{r,i}$ là sự không chắc chắn của máy phát và máy thu tại nút tương ứng. Bây giờ chúng ta xem xét RBS. Chúng ta có hai máy thu i và j , và nút i muốn ước lượng sự bù so với nút j . Hãy để sự bù thực sự là O . Từ hình 8.5, chúng ta có thể viết (với $t_1 = t_{1,i}$, i là thời gian trong đó i dán nhãn thời gian gói tin xung):

$$L_i(t_1) = L_j(t_1) + O - (\delta_{r,i} - \delta_{r,j}) - (\tau_i - \tau_j)$$

Kết quả là:

$$O = L_i(t_1) - L_j(t_1) + (\delta_{r,i} - \delta_{r,j}) + (\tau_i - \tau_j).$$

Giả thuyết rằng (i) sự khác biệt trong trễ truyền lan là không đáng kể, và (ii) sự không chắc chắn của máy phát là nhỏ (Phần 8.2.3), lỗi trong việc ước lượng sự bù chỉ phôi bởi thuật ngữ $\frac{\delta_{r,j} - \delta_{r,i}}{2}$ trong trường hợp của TPSN và bởi $\delta_{r,i} - \delta_{r,j}$ đối với RBS.

Đối với không phải TPSN cũng không phải RBS, độ lệch đồng hồ có thể trong quá trình trao đổi các gói tin đồng bộ được xem xét.

Sử dụng các đa xung

Máy phát có thể gửi các gói tin xung một cách đều đặn, đối với mỗi gói tin xung, các nút i và j trao đổi các quan sát cục bộ của chúng. Sự hồi quy tuyến tính bình phương nhỏ nhất được đề xuất cho RBS gần như tương đương với bộ ước lượng không phân cực khác biệt nhỏ nhất (minimum-variance unbiased estimator) được thảo luận trong phần 8.2.2. Một kỹ thuật loại bỏ bên ngoài (outlier-rejection technique) được sử dụng thêm vào. Bằng cách bắt chấp các quan sát này lâu hơn một ngưỡng nhất định (vài phút trong RBS), các tốc độ lệch biến đổi theo thời gian có thể được xem xét.

Đa nút và giá thành RBS

Nó rất đơn giản khi mở rộng kỹ thuật này tới $m > 2$ nút trong một miền quảng bá: các gói xung bắt được bởi tất cả các nút m và tất cả các nút trao đổi những quan sát của chúng. Bằng cách này, một nút đơn trong số các nút này, chẳng hạn nút i , có thể ước lượng độ lệch và các bù pha cho tất cả các nút ngang hàng của nó trong miền quảng bá. Bằng cách lưu các giá trị này cho mỗi nút ngang hàng trong một bảng, nút i có thể biến đổi các nhãn thời gian cục bộ của nó thành gốc thời gian của bất kỳ nút ngang hàng nào. Như đối với các máy phát xung, có ít nhất các lựa chọn sau đây:

- Trong các mạng nơi mà các nút hạ tầng tĩnh có mặt (ví dụ như IEEE 802.15.4 trong chế độ đèn hiệu - beaconed mode), các nút này có thể gửi các gói tin xung. Chúng có thể làm thậm chí nhiều hơn nữa – chúng cũng có thể lựa chọn các quan sát từ các nút cảm biến, tính toán cho mỗi cặp nút các sự bù và các tốc độ lệch với phương pháp bình phương nhỏ nhất, và gửi trả lại kết quả.
- Mỗi nút hoạt động như một máy thu và một máy phát của các gói tin xung. Theo cách này, cả nhận dạng máy phát và các số chuỗi trong các gói tin đều thực sự là cần thiết.

Sau đó mục tiêu về độ chính xác trong một miền quảng bá là giảm các lỗi pha cực đại giữa tất cả các cặp nút, được gọi là **độ phân tán nhóm** (group dispersion). Nó cho thấy rằng độ phân tán nhóm phụ thuộc vào kích thước nhóm (các nhóm lớn có độ phân tán nhóm lớn hơn) và phụ thuộc vào số lượng các quan sát được sử dụng để ước lượng độ lệch pha và tốc độ lệch (khi tăng số lượng các quan sát sẽ làm giảm độ phân tán nhóm). Đối với hai nút, nó cho thấy rằng việc tăng các quan sát vượt quá 30 không cung cấp thêm lợi ích cho độ chính xác. Giao thức RBS cũng được so sánh với NTP, NTP sử dụng một phương pháp dựa trên máy phát/máy thu. Trong viễn cảnh được nghiên cứu, RBS có độ phân tán nhóm nhỏ hơn nhiều so với độ phân tán nhóm có thể nhận được với NTP. Hơn nữa hiệu suất của RBS cũng gần như không nhạy cảm với tải nền (background load) trong miền quảng bá, trong khi đó độ phân tán nhóm của NTP tăng lên cùng với việc tăng tải. Điều này có thể được giải thích bằng ảnh hưởng của các trễ truy cập đường truyền trong NTP. Trong viễn cảnh đã cho, RBS tốt hơn xấp xỉ 8 lần so với NTP không có tải nền.

Kỹ thuật đồng bộ này giới thiệu tổng chi phí nào đó, mà tổng chi phí này vẫn chưa được mô tả đầy đủ:

- Để trao đổi các quan sát của một nhóm gồm m nút trong một miền quảng bá, cần một số lượng $2m$ gói tin khi các quan sát được chọn lựa và ước lượng tại một nút trung tâm và sau đó gửi trở lại các máy thu. Trong các viễn cảnh không có các nút trung tâm, việc trao đổi các quan sát chiếm $m \cdot (m - 1) = O(m^2)$ gói tin. Trong các mạng cảm biến mật độ cao, một nút có thể là một thành viên trong nhiều miền quảng bá (có thể chồng nhau một phần), và tải làm việc tăng lên một cách phù hợp.
- Tốc độ của các gói tin xung là cần thiết để giữ độ phân tán nhóm nhỏ hơn một mức nào đó phụ thuộc vào kích thước nhóm. Một đặc điểm chính xác hơn của mỗi quan hệ này cần phải được phát triển.
- Sự hồi qui bình phương nhỏ nhất yêu cầu sự tính toán đáng kể.

RBS và đồng bộ phía sau (post-factor)

Khi nhìn tổng chi phí, một người có cảm giác rằng RBS có thể trở nên khá đắt đỏ về mặt số lượng các gói tin trao đổi và số lượng các phép tính. Tuy nhiên, một trong các đặc điểm quan trọng nhất của RBS đó là nó có thể tham gia vào đồng bộ phía sau (**post-facto**). Trong chế độ này, các nút không đồng bộ với nhau trừ phi một sự kiện lợi ích bên ngoài xảy ra. Mỗi nút i dán nhãn thời gian sự kiện này với một nhãn thời gian cục bộ $L_i(t_0)$. Sự kiện này kích thích sự đồng bộ giữa các nút và các nút biết về các lệch pha tương đương và các tốc độ lệch của chúng. Nếu dành được đồng bộ một cách đủ nhanh sau khi sự kiện xuất hiện, nó an toàn khi giả thiết rằng độ lệch tương đối không thay đổi. Do đó, nút j có thể sử dụng ước lượng của nó về độ lệch pha của nút i và tốc độ lệch để ước lượng chính xác $L_j(t_0)$, nghĩa là để làm một *phép ngoại suy lùi*.

8.3.1.2 Đồng bộ mạng qua nhiều bước nhảy

Cái mà chúng ta đã biết đến thời điểm này là đồng bộ cục bộ trong một miền quảng bá. Trong nhiều trường hợp, một miền quảng bá bao phủ chỉ một phần nhỏ của cả mạng cảm biến. Do đó câu hỏi đặt ra là bằng cách nào một nút biết được tại thời điểm nào, được đo với trục thời gian cục bộ của nó, một sự kiện từ xa xuất hiện?

Phương pháp biến đổi nhãn thời gian

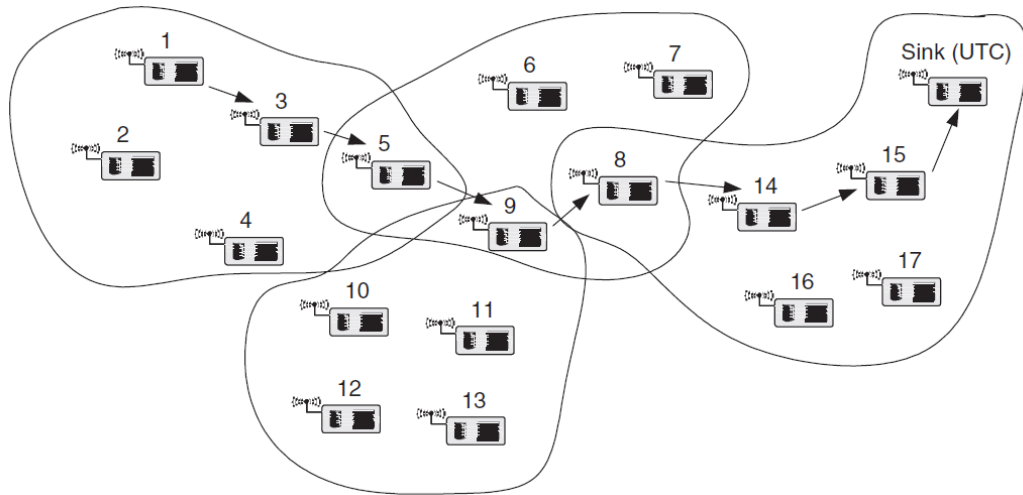
Ý tưởng cơ bản là không xây dựng một trục thời gian cục bộ, nhưng để biến đổi một nhãn thời gian của gói tại mỗi bước nhảy thành trục thời gian của nút của bước nhảy kế tiếp cho đến khi nó tiến đến đích cuối cùng. Hình 8.6 cung cấp một ví dụ minh họa. Chúng ta hãy giả thiết rằng nút 1 quan sát một sự kiện mà nó muốn để thông báo cho nút thu. Người điều hành tại nút thu muốn biết khi nào sự kiện xảy ra và anh ta cần thông tin này trong một trục thời gian được chấp nhận chẳng hạn như UTC. Nút 1 dán nhãn thời gian sự kiện tại thời điểm $L_1(t)$, và gói này phải chuyển qua các nút $1 \rightarrow 3 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 14 \rightarrow 15 \rightarrow$ máy thu. Thêm vào đó, giả thiết rằng các nút tạo ra các miền quảng bá được chỉ ra theo các vòng tròn. Quan sát thấy rằng các nút 5, 8, và 9 là các thành viên của nhiều miền quảng bá. Nút nguồn 1 là ở cùng miền quảng bá như nút 3 kế tiếp nó và do đó nút 1 có thể sử dụng các ước lượng về độ lệch pha của nó và tốc độ lệch của nút 3 để biến đổi nhãn thời gian $L_1(t)$ thành $L_3(t)$. Nút 1 dán nhãn thời gian gói tin sự kiện không phải với nhãn thời gian của bản thân nó mà với $L_3(t)$. Nút 3, dựa trên việc nhận được gói tin này, làm một quyết định định tuyến và suy ra rằng nút 5 là bước nhảy kế tiếp. May là nút 5 ở trong cùng một miền quảng bá như đối với nút 3 và nút 3 do đó có thể biến đổi nhãn thời gian nhận được $L_3(t)$ thành $L_5(t)$. Quá trình này được tiếp tục cho đến khi gói tin tiến tới máy thu. Nút cuối cùng trước máy thu, nút 15, là nằm trong cùng miền quảng bá với máy thu, nơi giữ thời gian UTC. Một lần nữa, nút 15 biết độ lệch pha của nó và tốc độ lệch đối với máy thu và do đó có thể biến đổi nhãn thời gian $L_{15}(t)$ thành thời gian UTC.

Các điểm sau là đáng chú ý:

- Nút thu nhận một nhãn thời gian sự kiện trong trục thời gian UTC.
- Mỗi nút giữ trục thời gian cục bộ của nó và việc biến đổi nhãn thời gian có thể được tích hợp với việc chuyển tiếp gói tin. Thực tế, khi nút i phải quyết định về một sự chuyển tiếp có khả năng xảy ra, chỉ các ứng cử viên j mà có một miền quảng bá giống như với i mới là các ứng cử viên phù hợp. Điều này yêu cầu rằng đối với nút i và một nút chuyển tiếp có khả năng xảy ra j , tồn tại một nút thứ ba mà các gói tin

xung của nó có thể được nghe bởi cả i và j . Điều này có thể không luôn tồn tại trong các mạng cảm biến thưa thớt.

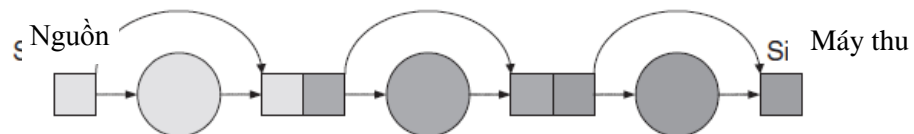
- Không có thông tin liên quan đến đồng bộ thời gian. Đồng bộ thời gian phải có sẵn toàn mạng.
- Nếu lỗi đồng bộ ở một bước nhảy đơn được phân loại bởi một phân bố thông thường với trung bình bằng 0 và độ lệch tiêu chuẩn σ và nếu các lỗi đồng bộ của các bước nhảy khác nhau là độc lập, thì sau n bước nhảy lỗi đồng bộ là hàm Gaussian với trung bình bằng 0 và độ lệch tiêu chuẩn $\sigma\sqrt{n}$. Tốc độ phát triển này cũng được xác định bằng các phép đo.



Hình 8.6 Nhiều miền quảng bá

Bằng cách nào để tạo ra các miền quảng bá

Một trong các điểm mở trong RBS là bằng cách nào các miền quảng bá được tạo thành và bằng cách nào nó có thể đảm bảo rằng một đường biến đổi thời gian tồn tại một cách thực sự giữa hai nút mong muốn, chẳng hạn như giữa một cảm biến và một nút thu.



Hình 8.7 Tích hợp gói tin, chuyển tiếp và biến đổi nhãn thời gian

Chúng ta hãy nhìn vào hai kịch bản khác nhau (Hình 8.7). Trong tiêu chí đầu tiên, số lượng các nút tĩnh hoạt động như các máy phát xung chuyên dụng và cũng như các bộ chuyển tiếp gói tin; chúng được trình bày như là các vòng tròn trong hình vẽ. Các nút cảm biến thông thường (được trình bày như các hình chữ nhật) trong phạm vi của một máy phát xung không cần thiết phải có các nút lân cận trung gian để được đồng bộ. Các nút cảm biến dán nhãn thời gian các gói tin xung, phát các quan sát của nó tới máy phát xung, máy phát xung tính toán các độ lệch pha và các tốc độ lệch tương đương cho mỗi cặp nút cảm biến và phân phối các kết quả trở lại. Do đó, một miền quảng bá được tạo nên bởi phạm vi truyền dẫn của máy phát xung. Một nút cảm biến i sắp sửa chuyển tiếp một gói tin có hai tùy chọn:

- Nó có thể tự xác định nút cảm biến j khác trong cùng một miền quảng bá để chuyển tiếp gói tin tới đó. Nếu cả hai là các nút lân cận một bước nhảy, nút i có thể biến đổi nhãn thời gian và gửi gói tin trực tiếp tới j . Điều này được chỉ ra bởi các vòng cung nét đậm trong hình 8.7. Trong trường hợp khác, nút i có thể hỏi máy phát xung để chuyển tiếp gói tin tới nút j mà không cần thao tác nó. Trong cả hai trường hợp, không cần máy phát xung để giữ một bảng với thông tin biến đổi.
- Nó có thể chuyển tiếp gói tin này tới máy phát xung. Trách nhiệm của máy phát xung là tra bảng một nút kế vị j và để biến đổi nhãn thời gian (được chỉ ra bởi các đường gạch ngang trong hình 8.7). Do đó, cần phải có máy phát xung để giữ một bảng với các tham số biến đổi.

Trong mỗi trường hợp, máy phát xung không lợi dụng đồng hồ cục bộ của nó và do đó không cần thiết phải được đồng bộ với các nút cảm biến của nó. Trong cả hai trường hợp, các miền quảng bá nên được lựa chọn như sau:

- Mỗi nút cảm biến là một thành viên của ít nhất một miền quảng bá.
- Khi một gói tin sắp được chuyển giao từ một nút nguồn tới một nút thu trong một miền quảng bá khác, phải có một dãy các miền quảng bá lân cận trên đường đi giữa nguồn và máy thu và hai miền quảng bá lân cận buộc phải chồng nhau ở ít nhất một nút. Các **nút cổng** này - **gateway nodes** (các nút chữ nhật với hai mức xám khác nhau trong hình 8.7) biết các tham số biến đổi cho tất cả các nút trong cả hai miền quảng bá và do đó có thể biến đổi các nhãn thời gian. Cần phải có đủ sự chồng nhau giữa các miền để đảm bảo sự hiện diện của các nút cổng. Một cách lý tưởng, giữa hai miền có hơn một nút cổng để cho phép một số tải cân bằng và chuyển tiếp.
- Số lượng của các sự biến đổi cần thiết giữa một nút nguồn và một nút thu nên nhỏ để tránh mất chính xác; điều này yêu cầu các miền quảng bá rộng. Nói cách khác, trong các miền rộng lớn, nhiều gói tin hơn phải được trao đổi và một công suất phát lớn phải được sử dụng để đến được tất cả các nút lân cận, điều này làm tiêu hao pin một cách nhanh chóng.

Điều này có thể được xem như một vấn đề phân nhóm (xem chương 10), với mục đích là tìm các **phân nhóm chồng lên nhau (overlapping clusters)**. Có một giao thức phân nhóm dành riêng cho các mục đích của RBS. Giao thức này cho phép điều chỉnh kích thước phân nhóm để tìm ra sự không dung hòa giữa việc tối thiểu hóa số lượng các bước nhảy và việc tối thiểu hóa số lượng các gói tin/công suất phát.

Nếu không có sẵn các nút dành riêng hoặc khi thực tế là tất cả các nút bao gồm cả các máy phát xung cần thiết phải được đồng bộ, các miền quảng bá buộc phải nhỏ hơn và tất cả các thành viên của một miền quảng bá buộc phải là các nút lân cận một bước nhảy để có thể trao đổi các quan sát xung cũng như chuyển tiếp các gói tin dữ liệu. Một miền quảng bá do đó là một **tập hợp** các nút. Theo kịch bản này mỗi nút hoạt động vừa như một máy thu vừa như một máy phát của các gói tin xung, và các quyết định chuyển tiếp là bị giới hạn vào các nút lân cận một bước nhảy, các nút lân cận một bước nhảy này chia sẻ ít nhất một miền quảng bá với nút chuyển tiếp.

8.3.2 Phân tầng tham chiếu đồng bộ thời gian (HRTS - Hierarchy referencing time synchronization)

Giao thức đồng bộ thời gian được trình bày bởi Dai và Han [8] bao gồm kỹ thuật dựa trên một máy thu/máy thu và một máy phát/máy thu. Nó bao gồm hai giao thức con, hai giao thức con này có thể hoạt động độc lập với nhau. Giao thức **Phân tầng tham chiếu đồng bộ thời gian (HRTS)** thảo luận dưới đây là một kỹ thuật máy thu/máy thu đồng bộ tất cả các nút trong một miền quảng bá và thêm vào đó xây dựng một cây đồng bộ để đồng bộ cho toàn mạng. Việc đồng bộ được kích hoạt theo chu kỳ bởi gốc của cây. Giao thức khác, gọi là giao thức ITR (Individual-based Time Request - Yêu cầu thời gian dựa trên cá nhân), tuy nhiên giao thức này không được thảo luận ở đây.

Đồng bộ trong một miền quảng bá

Chúng ta giải thích phương pháp này bằng một ví dụ, trình bày ở hình 8.8. Một nút dành riêng, được gọi là **nút gốc** hoặc **trạm gốc** (nút R trong hình 8.8) kích hoạt đồng bộ thời gian tại thời điểm t_1 (tương ứng với thời gian cục bộ $L_R(t_1)$ bằng cách phát quảng bá một gói tin thông báo **bắt đầu đồng bộ**). Gói tin này bao gồm hai tham số, một giá trị **mức** (được giải thích dưới đây) và nhận dạng của một trong các nút lân cận một bước nhảy của nút R , chẳng hạn như nút i . Nút được lựa chọn i dán nhãn thời gian gói tin nhận được tại thời điểm t_2 với thời gian cục bộ của nó $L_i(t_2)$. Nút khác dán nhãn thời gian cùng gói tin đó tại thời điểm t'_2 với $L_j(t'_2)$. Tương tự với RBS, t_2 và t'_2 có thể khác nhau chút ít bởi vì các sự không chắc chắn ngẫu nhiên của máy thu. Bởi vì nút i tìm thấy nhận dạng của nó trong gói tin, nó định dạng một gói tin trả lời và dán nhãn gói tin trả lời này để truyền dẫn tại thời điểm t_3 với thời gian cục bộ của nó là $L_i(t_3)$. Cả hai nhãn thời gian đều bao gồm $L_i(t_2)$ và $L_i(t_3)$ trong gói tin trả lời. Nút gốc R nhận gói tin và dán nhãn thời gian nó tại thời điểm t_4 với $L_R(t_4)$. Nút gốc R bây giờ có thể ước lượng sự bù $O_{R,i}$ giữa đồng hồ cục bộ của nó và đồng hồ cục bộ của nút i trong một kiểu cách tương tự như giao thức đồng bộ theo cặp (pair-wise) của LTS và TPSN như sau:

$$O_{R,i} = \frac{(L_i(t_2) - L_R(t_1)) - (L_R(t_4) - L_i(t_3))}{2}$$

Trong bước kế tiếp, nút gốc R quảng bá các giá trị O và $L_i(t_2)$ tới tất cả các nút. Bây giờ, chúng ta hãy xem bằng cách nào các nút cá nhân điều chỉnh được các đồng hồ của chúng:

- Nút i đơn giản trừ đi sự bù $O_{R,i}$ khỏi đồng hồ cục bộ của nó.
- Giả thiết rằng nút j khác trong cùng một miền quảng bá có sự bù pha $O_{j,i}$ so với nút i . Dưới giả thiết rằng i và j nhận **bắt đầu đồng bộ** tại cùng thời điểm, chúng ta có $t_2 = t'_2$ và $L_i(t_2) = L_j(t'_2) + O_{j,i}$. Dựa trên việc nhận được giá trị đồng hồ $L_i(t_2)$ từ việc quảng bá cuối cùng của nút gốc, nút j có thể tính toán $O_{j,i}$ trực tiếp như sau $O_{j,i} = L_i(t_2) - L_j(t'_2)$. Mặt khác, bởi vì

$$L_i(t_2) = L_R(t_2) + O_{R,i}$$

$$L_j(t_2) = L_R(t_2) + O_{R,j}$$

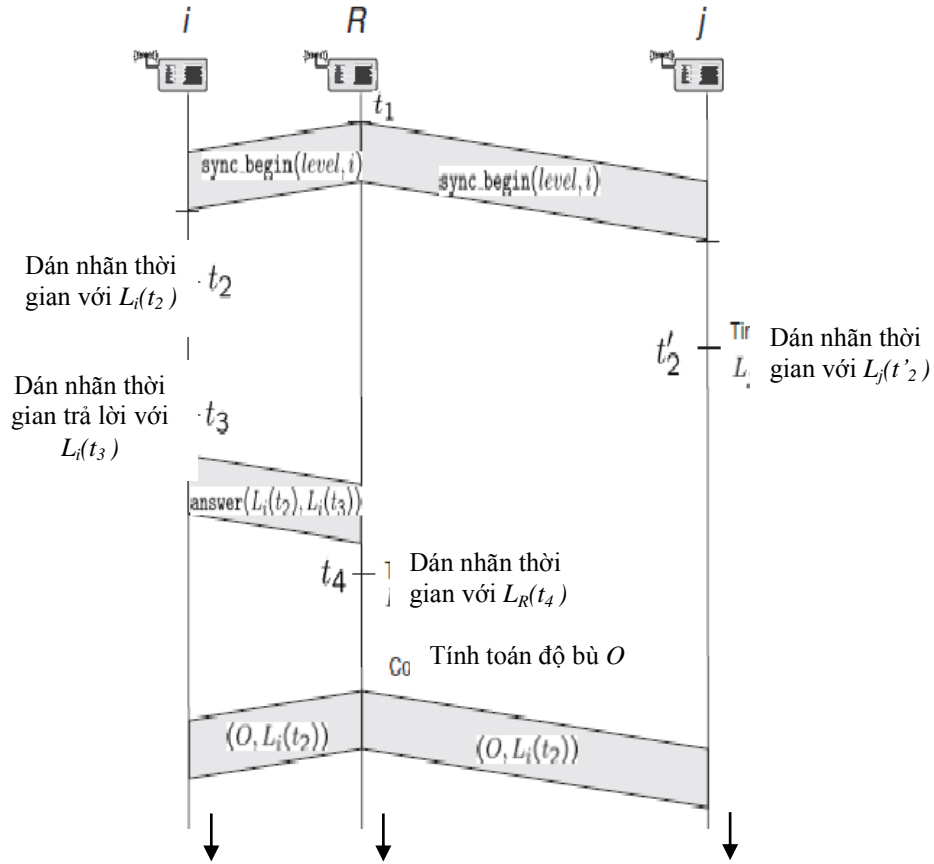
chúng ta nhận được

$$L_R(t_2) + O_{R,i} = L_j(t'_2) + O_{i,j}$$

mà cho

$$\begin{aligned}
O_{R,j} &= L_j(t_2) - L_R(t_2) \\
&\approx L_j(t'_2) - L_R(t_2) \\
&= L_R(t_2) + O_{R,i} - O_{j,i} - L_R(t_2) \\
&= O_{R,i} - O_{j,i} \\
&= O_{R,i} - (L_i(t_2) - L_j(t'_2))
\end{aligned}$$

Do đó, nút j có thể tính toán sự bù pha so với nút R một cách trực tiếp từ việc quan sát $L_j(t'_2)$ của chính nó và các giá trị từ bản tin quảng bá cuối cùng, mà *không phải trao đổi bất cứ gói tin nào với các nút khác*.



Hình 8.8 Đồng bộ thời gian: đồng bộ trong một miền quảng bá đơn lẻ

Đặc điểm quan trọng của phương pháp này đó là ba gói tin đủ để đồng bộ tất cả các nút lân cận của nút R theo đồng hồ của R , bất kể số lượng của chúng là bao nhiêu và liệu chúng có nằm trong phạm vi của nhau hay không. Đối lập với RBS, các máy thu của gói tin **bắt đầu đồng bộ** không phải trao đổi tất cả các quan sát của chúng theo kiểu theo cặp và do đó giao thức này là không nhạy cảm đối với mật độ của mạng.

Dai và Han [8] trình bày một cách thực nghiệm rằng đối với một miền quảng bá đơn lẻ cũng như qua nhiều bước nhảy, giao thức HRTS và RBS có xấp xỉ cùng phân bố các lỗi đồng bộ khi tiến hành một lượng lớn các phép lặp. Tương tự như các giao thức khác, sự không chắc chắn thu và phát có thể giảm đi bằng cách dán nhãn thời gian các gói tin đi ra càng muộn càng tốt và dán nhãn thời gian các gói tin đi vào càng sớm càng tốt.

Mặc dầu không cần thiết đối với việc thực hiện phương pháp này, Dai và Han [8] giả thiết chạy giao thức này trên một kênh MAC tách biệt. Mục đích là để tách biệt thông tin truyền đi liên quan đến đồng bộ với thông tin dữ liệu người sử dụng và do đó làm giảm xác suất của xung đột và các trễ truy cập đường truyền khi nút i được chọn trả lời gói tin **bắt đầu đồng bộ** của nút gốc. Rõ ràng là, nút gốc gửi gói tin **bắt đầu đồng bộ** trên một **kênh điều khiển** và bao gồm các thông số của kênh dành riêng khác, **kênh đồng hồ**. Cả nút gốc và nút được chọn i chuyển sang kênh đồng hồ và i phát gói tin trả lời của nó. Cuối cùng thì nút gốc phát quảng bá gói tin chứa $L_i(t_2)$ và O một lần nữa trên kênh điều khiển. Tất cả các nút khác có thể tiếp tục chuyển tiếp các gói tin dữ liệu như R và i trao đổi các gói tin của chúng.

Đồng bộ mạng qua đa bước nhảy

Chúng ta giả thiết rằng toàn bộ mạng là tĩnh và được nối và có ít nhất một nút tham chiếu có truy cập tới một nguồn thời gian ngoài và do đó tới thời gian UTC; có thể có nhiều nút tham chiếu. Các nút tham chiếu được chỉ định ở mức 0 và trở thành các nút gốc cho các nút lân cận một bước nhảy của chúng. Chúng kích hoạt việc tái đồng bộ theo chu kỳ, theo giao thức được mô tả trong phần trước. Mức này được bao gồm trong gói tin **bắt đầu đồng bộ**. Tất cả các nút xử sự tuân theo một quy luật đơn giản. Chúng duy trì một biến mức cục bộ, được bắt đầu với một giá trị đủ lớn. Nếu một nút thu một gói tin **bắt đầu đồng bộ** với một giá trị mức là thực sự nhỏ hơn so với biến mức của bản thân nó, nút chấp nhận gói tin, đặt mức của bản thân nó bằng giá trị mức nhận được cộng thêm 1, và trở thành một nút gốc của chính nó, bắt đầu đồng bộ với các nút lân cận của nó sau khi việc kích hoạt vòng đồng bộ kết thúc. Trong trường hợp khác, nếu mức thu là lớn hơn hoặc bằng với biến mức của bản thân nó, đơn giản là bỏ đi gói tin **bắt đầu đồng bộ**. Quá trình này tiếp tục theo kiểu đệ quy cho đến khi tiến tới rìa của mạng.

Theo cách này, các nút luôn đồng bộ qua đa bước nhảy với nút tham chiếu gần nhất, và biến mức của chúng chỉ ra khoảng cách nhảy. Nếu một nút vừa được đồng bộ biết muộn hơn về một nút tham chiếu gần hơn, nó ấn định cho bản thân nó một mức mới và bắt đầu việc tái đồng bộ cho các nút con của nó. Tổng chi phí (biểu diễn dưới dạng số lượng các gói tin được trao đổi) của HRTS và RBS được Dai và Han [189] so sánh đối với một kịch bản gồm 200 nút đặt trong một khu vực có diện tích $400 \times 400 \text{ m}^2$ khi biến đổi mức độ nút trung bình. RBS tạo ra tổng chi phí cao hơn nhiều so với HRTS, tuy nhiên, tổng chi phí của HRTS tăng lên khi mật độ nút tăng lên. Một giải thích cho điều này có thể là sự quan sát mà các miền quảng bá của các nút mức cao hơn là nhỏ hơn các miền quảng bá của các nút tham chiếu bởi vì các phạm vi truyền dẫn của các nút con lân cận bị chồng nhau. Theo cách này chúng có khá ít nút con.

Giao thức này có một cơ chế thêm vào để hạn chế chiều sâu của cây đồng bộ có gốc là ở một nút tham chiếu. Thêm vào tham số mức và nhận dạng của nút trả lời, mỗi gói tin **bắt đầu đồng bộ** cũng mang một tham số chiều sâu, tham số này được giảm dần trong từng mức của cây. Việc xây dựng cây dừng lại khi tiến tới chiều sâu bằng 0.

CÂU HỎI ÔN TẬP CHƯƠNG 8

1. Sự cần thiết của vấn đề đồng bộ thời gian trong WSN?
2. Vấn đề về độ chính xác trong đồng bộ thời gian?
3. Đặc điểm và cấu trúc của thuật toán đồng bộ thời gian?
4. Đồng bộ thời gian trong WSN?
5. Giới thiệu khái quát các giao thức dựa trên đồng bộ máy phát/máy thu?
6. Các phương pháp tăng độ chính xác của các giao thức?
7. Giao thức đồng bộ thời gian cho các mạng cảm biến?
8. Giới thiệu khái quát các giao thức dựa trên đồng bộ máy thu/máy thu?
9. Nguyên nhân của các lỗi đồng bộ máy thu/máy thu?
10. So sánh RBS và TPSN?
11. Phương pháp đồng bộ mạng qua nhiều bước nhảy?
12. Đặc điểm của HRTS?

CHƯƠNG 9. ĐỊNH VỊ VÀ XÁC ĐỊNH VỊ TRÍ

Mục đích của chương

Chương này đưa ra một tổng quan về các phương pháp để xác định vị trí tượng trưng – “ví dụ: trong phòng khách” – và vị trí số - “tại các tọa độ (23,54; 11,87)” – của một nút cảm biến không dây. Các đặc điểm của các phương pháp này và các khả năng chính đối với một nút để xác định thông tin về vị trí của nó được thảo luận trong chương này. Các cơ sở toán học cho việc xác định vị trí được giới thiệu và trường hợp xác định vị trí một bước nhảy và nhiều bước nhảy được mô tả bằng cách sử dụng vài hệ thống ví dụ. Ở cuối chương này, người đọc sẽ hiểu được những sự không dung hòa chủ yếu trong thiết kế đối với việc xác định vị trí.

Trong nhiều tình huống, một nút trong một mạng cảm biến không dây nếu biết được vị trí của nó trong thế giới vật lý thì rất là hữu ích hoặc thậm chí là rất cần thiết. Ví dụ, các chức năng theo dấu vết hoặc phát hiện sự kiện sẽ không đặc biệt hữu ích nếu WSN không thể cung cấp bất cứ thông tin nào về việc một sự kiện sẽ xảy ra ở đâu. Thường thì để làm điều đó vị trí của các nút thông báo phải được biết. Việc định dạng thông tin định vị trong mỗi nút trong khi triển khai không phải là một tùy chọn. Tương tự, việc trang bị mỗi nút với một bộ thu Hệ thống Định vị Toàn cầu (GPS) là không thể thực hiện được do hạn chế về giá thành và hạn chế trong việc triển khai (chẳng hạn GPS không hoạt động được trong nhà).

Chương này giới thiệu nhiều kỹ thuật khác nhau về việc bằng cách nào các nút cảm biến có thể biết được vị trí của chúng một cách tự động, hoặc là hoàn toàn tự động bằng cách dựa vào phương tiện của bản thân WSN hoặc bằng cách sử dụng một số hỗ trợ từ hạ tầng cơ sở bên ngoài.

9.1 Các đặc điểm của quy trình định vị và xác định vị trí

Sự hiểu biết bằng trực giác đơn giản về “cung cấp thông tin định vị cho một nút” có một số các khía cạnh mà nên được phân loại để thực hiện các tùy chọn cho một quy trình định vị rõ ràng. Các đặc điểm quan trọng nhất là:

Vị trí vật lý so với vị trí biểu tượng: hệ thống cung cấp dữ liệu về *vị trí vật lý* của một nút (trong hệ thống tọa độ số nào đó) hoặc một nút biết về một *vị trí biểu tượng* – ví dụ, “trong phòng khách”, “văn phòng 123 trong tòa nhà 4”? Thêm vào đó, nó có thể phù hợp với vị trí vật lý với một tên vị trí mang tính biểu tượng?

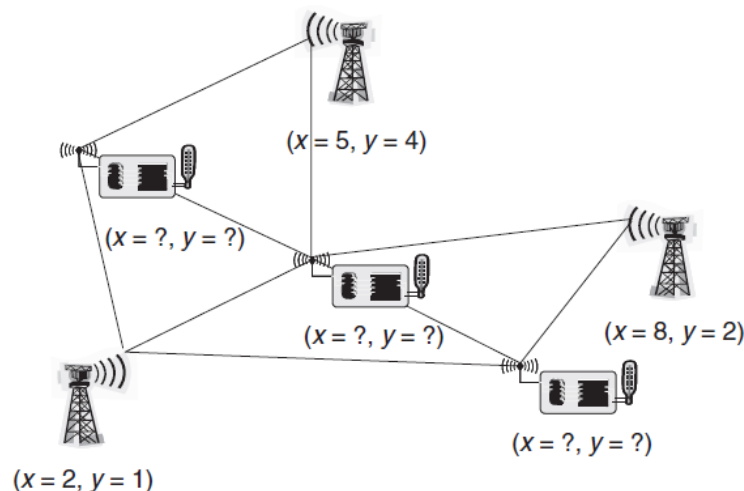
Trong khi hai khái niệm này là khác nhau, thì cũng không có một thuật ngữ nhất quán nào trong tài liệu – việc xác định vị trí và định vị thường được sử dụng hoán đổi cho nhau. Xu hướng là sử dụng “định vị” như một thuật ngữ chung. Chúng ta phải dựa trên tình huống để phân biệt giữa hai thuật ngữ này.

Tọa độ tuyệt đối và tọa độ tương đối: Một hệ thống tọa độ tuyệt đối là hợp lệ cho tất cả các đối tượng và được nhúng trong một khung tham chiếu chung nào đó. Ví dụ: các vị trí trong các tọa độ UTM (Universal Transverse Mercator) tạo ra một hệ thống tọa độ tuyệt đối cho bất cứ vị trí nào trên trái đất. Mặt khác, các tọa độ tương đối, có thể khác đối với bất kỳ đối tượng được định vị nào hoặc tập hợp của các đối tượng – ví dụ: một WSN ở đó các nút có các tọa độ, các tọa độ này là đúng đối với nhau nhưng không có mối quan hệ nào đối với các tọa độ tuyệt đối.

Để cung cấp các tọa độ tuyệt đối, vài **nguồn tin cậy** (nút neo - **anchors**) là cần thiết (ít nhất là ba **nguồn tin cậy** đối với một hệ thống hai chiều). Các **nguồn tin cậy** này là các nút mà biết vị trí của bản thân chúng trong hệ thống tọa độ tuyệt đối. Các **nguồn tin cậy** có thể xoay, tịnh tiến, và có thể tỉ lệ với một hệ thống tọa độ tương đối do đó nó trùng với hệ thống tọa độ tuyệt đối. Các **nút neo** này cũng được gọi chung là “*đèn hiệu - beacons*” hoặc “*mốc giới - landmarks*” ở trong tài liệu.

Tính toán cục bộ so với tính toán tập trung: Có bất cứ tính toán theo yêu cầu nào được thực hiện một cách cục bộ bởi các đối tượng tham gia, trên cơ sở của vài phép đo có sẵn mang tính cục bộ hoặc là các phép đo được thông báo cho một trạm trung tâm, trạm trung tâm này tính toán các vị trí hoặc các định vị và phân phối chúng ngược trở lại cho các đối tượng tham gia hay không? Bên cạnh các mối quan tâm về tỉ lệ và về hiệu suất (cả hai đều có ý nghĩa đối với tổng chi phí cho việc truyền thông và tính toán), ở đây các vấn đề riêng tư là quan trọng khi một đối tượng tham gia có thể không mong muốn để lộ vị trí của nó cho một thực thể trung tâm.

Độ chính xác tuyệt đối và độ chính xác tương đối (Accuracy và Precision): Hai thông số xứng đáng được quan tâm nhất đối với một hệ thống định vị là **Độ chính xác tuyệt đối và độ chính xác tương đối** của kết quả. Xác định vị trí chính xác tuyệt đối là khoảng cách lớn nhất giữa vị trí thực sự và vị trí ước lượng của một thực thể (độ chính xác cao biểu thị một sự sai lệch cực đại nhỏ). Độ chính xác tương đối là một tỉ số, với tỉ số này một độ chính xác tuyệt đối đã cho được đạt tới, được lấy trung bình trên nhiều cố gắng lặp lại để xác định một vị trí. Ví dụ: một hệ thống có thể yêu cầu cung cấp một độ chính xác tuyệt đối là 20-cm với độ chính xác tương đối ít nhất là 95%. Hiển nhiên, các giá trị độ chính xác tuyệt đối và độ chính xác tương đối chỉ có ý nghĩa khi chúng được xem xét cùng nhau, tạo thành đặc điểm độ chính xác tuyệt đối/độ chính xác tương đối của một hệ thống.



Hình 9.1 Xác định vị trí của các nút cảm biến với sự hỗ trợ từ một số các điểm neo; không phải tất cả các nút là cần thiết trong giao tiếp với tất cả các nút neo.

Phạm vi: một hệ thống có thể được dự kiến cho các phạm vi khác nhau, ví dụ - trong việc triển khai trong nhà – kích thước của một phòng hoặc một tòa nhà hoặc - trong việc triển khai bên ngoài nhà – một chỗ đỗ xe hoặc thậm chí một hoạt động trên toàn cầu. Hai vấn đề quan trọng ở đây là khu vực mà hệ thống có thể bao phủ trên một đơn vị cơ sở hạ tầng và số lượng của các đối tượng có thể định vị trên một đơn vị cơ sở hạ tầng trong một khoảng thời gian.

Các hạn chế: đối với một số các kỹ thuật xác định vị trí, có các hạn chế triển khai cố hữu. Ví dụ: GPS không làm việc trong nhà; các hệ thống khác thì chỉ bị giới hạn về các phạm vi, nghĩa là chúng chỉ hoạt động trong các phạm vi này.

Giá thành: Các hệ thống xác định vị trí gây ra chi phí về mặt thời gian (cài đặt hạ tầng, quản lý), về mặt không gian (kích thước thiết bị, không gian cho hạ tầng), năng lượng (trong suốt quá trình hoạt động), và vốn (giá thành của một nút, cài đặt hạ tầng).

Hình 9.1 minh họa vấn đề xác định vị trí. Các hình vẽ trong chương này sử dụng biểu tượng “điểm truy cập” để biểu thị các **nút neo** cho dễ phân biệt. Tuy nhiên, người ta có thể chỉ ra rằng các nút cảm biến thông thường chỉ có thể được sử dụng tốt như các **nút neo**, miễn là chúng biết về vị trí của chúng. Thêm vào đó, một hệ thống xác định vị trí hoặc định vị có thể được sử dụng để cung cấp sự thừa nhận hoặc sự phân loại các đối tượng; đặc điểm này ít quan trọng hơn trong bối cảnh WSN hoặc nếu được sử dụng thì thường không được xem xét như một phần của hệ thống định vị.

9.2 Các phương pháp có thể

Có ba phương pháp chính tồn tại để xác định vị trí của một nút: sử dụng thông tin về một nút lân cận của nút (các phương pháp dựa trên sự gần gũi), khai thác các đặc điểm hình học của một bối cảnh đã cho (phép đặc tam giác và phép đo ba cạnh tam giác), và cố gắng phân tích các đặc điểm tiêu biểu của vị trí của một nút so với các đặc điểm được đo trước đó (phân tích cảnh).

9.2.1 Phương pháp dựa trên sự gần gũi

Kỹ thuật đơn giản nhất là khai thác phạm vi hữu hạn của truyền thông không dây. Nó có thể được sử dụng để xác định liệu một nút mà muốn xác định vị trí của nó hoặc định vị có nằm trong vùng lân cận (trong sự gần gũi) của một **nút neo** hay không. Trong khi điều này chỉ cung cấp thông tin thô, thì các thông tin này có thể là hoàn toàn đủ. Ví dụ: sự hạn chế tự nhiên của việc truyền thông bằng sóng hồng ngoại là các bức tường, các bức tường này có thể được dùng để cung cấp cho một nút các thông tin định vị đơn giản về cái phòng mà nút ở trong đó.

Các hệ thống dựa trên sự gần gũi có thể khá phức tạp và thậm chí chúng có thể được sử dụng để xấp xỉ vị trí khi một nút có thể phân tích thông tin gần gũi của vài **nút neo** chồng nhau. Chúng cũng có thể là tương đối hiệu quả đối với các sự không chắc chắn của kênh không dây – quyết định liệu một nút nằm trong vùng lân cận của nút khác có tương đương với việc quyết định kết nối, các kết nối mà có thể xảy ra trong các phạm vi thời gian tương đối dài, lấy trung bình các sự thay đổi bất thường ngắn hạn.

9.2.2 Phép đặc tam giác (triangulation) và phép đo ba cạnh tam giác (Trilateration)

Cạnh bên đối với sự tạo thành góc

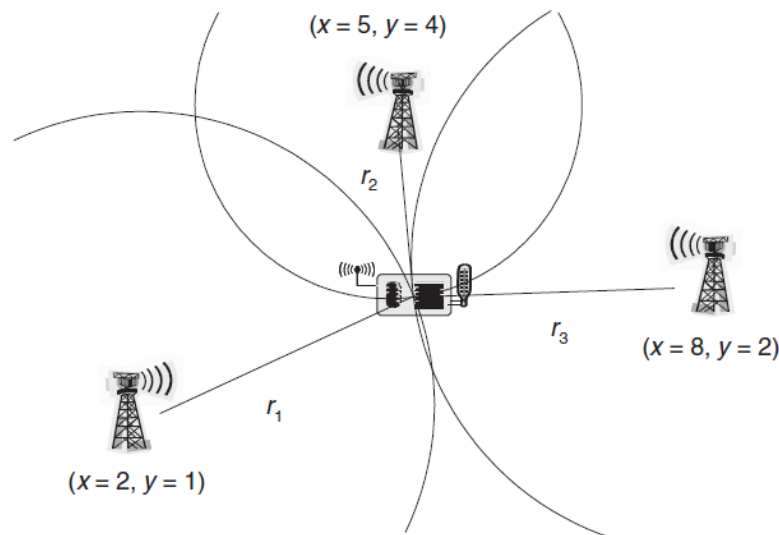
Thêm vào thông tin kết nối/vùng lân cận, việc truyền thông giữa hai nút thường cho phép rút ra thông tin về mối quan hệ hình học của chúng. Ví dụ: khoảng cách giữa hai nút hoặc góc trong một tam giác có thể được ước lượng – bằng cách nào điều này được thảo luận trong hai phần nhỏ sau đây. Sử dụng hình học sơ cấp, thông tin này có thể được sử dụng để thu thập thông tin về các vị trí của nút. Khi các khoảng cách giữa các thực thể được sử dụng, phương pháp này được gọi là **Cạnh bên**; khi các góc giữa các nút được sử dụng, phương pháp

này gọi là **sự tạo thành góc**. Đối với **cạnh bên** trên một mặt phẳng, trường hợp đơn giản nhất là một nút có các phép đo khoảng cách chính xác tới ba **nút neo** không đồng tuyến tính. Việc mở rộng tới một không gian ba chiều là không quan trọng (cần bốn **nút neo**); tất cả thảo luận dưới đây sẽ tập trung vào trường hợp hai chiều cho đơn giản. Sử dụng các khoảng cách và các vị trí **nút neo**, vị trí của nút phải là ở giao điểm của ba vòng tròn xung quanh các **nút neo** (Hình 9.2).

Vấn đề ở đây là, trên thực tế, các phép đo khoảng cách là không bao giờ hoàn hảo và giao điểm của ba vòng tròn này nói chung sẽ không cho kết quả là một điểm đơn lẻ. Để khắc phục sự không hoàn hảo này, các phép đo khoảng cách tạo ra nhiều hơn cái mà ba **nút neo** có thể được sử dụng, gây ra vấn đề **đa cạnh** (multilateration). Sự đa cạnh là giải pháp kỹ thuật lỗi, được sử dụng và tái sử dụng trong nhiều hệ thống cụ thể được mô tả dưới đây. Các chi tiết toán học của nó được đưa ra ở phần 9.3.

Sự tạo thành góc khai thác sự thật rằng trong một tam giác mỗi khi chiều dài của hai cạnh và hai góc là đã biết thì vị trí của điểm thứ ba được biết như là giao điểm của hai cạnh còn lại của tam giác. Vấn đề của các phép đo không chính xác cũng tăng lên ở đây và cũng có thể được giải quyết bằng cách sử dụng nhiều phép đo.

Xác định các khoảng cách



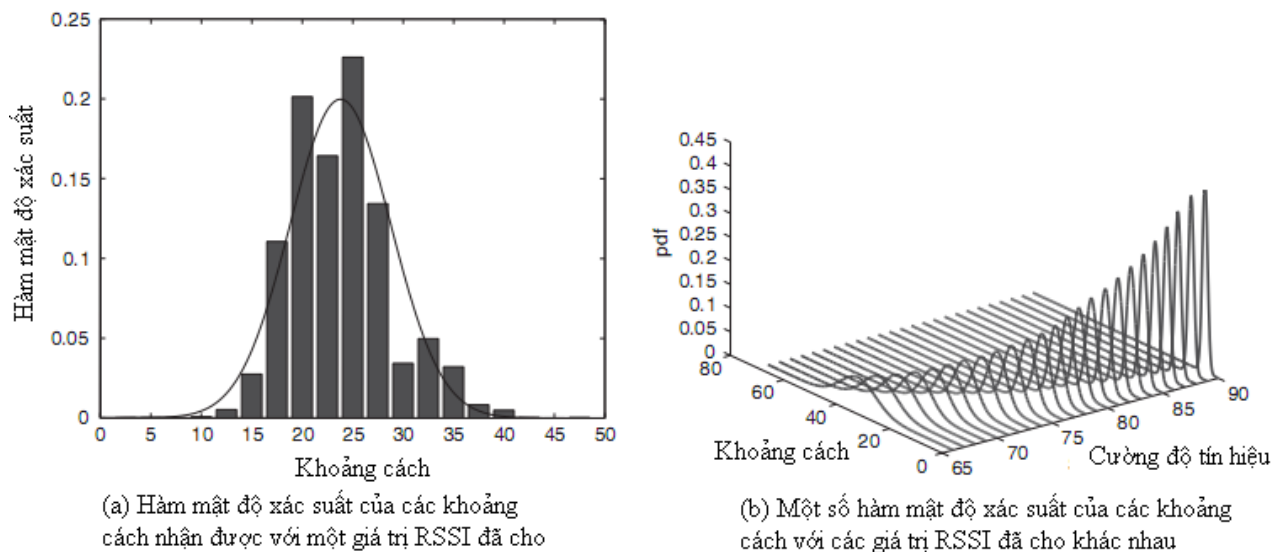
Hình 9.2 Phép đạc tam giác bởi ba vòng tròn giao nhau

Để sử dụng (nhiều) cạnh bên, yêu cầu phải có các ước lượng của các khoảng cách tới **nút neo**. Quá trình lập phạm vi lý tưởng có tác dụng thúc đẩy các điều kiện thuận lợi đã có sẵn trong một nút không dây, đặc biệt là thiết bị truyền thông vô tuyến. Các đặc điểm của truyền thông không dây được xác định một phần bởi khoảng cách giữa máy phát và máy thu, và nếu các đặc điểm này có thể đo được ở phía máy thu, thì chúng có thể được sử dụng như một bộ ước lượng khoảng cách. Các đặc điểm quan trọng nhất là Bộ chỉ thị độ lớn tín hiệu thu (RSSI - Received Signal Strength Indicator), Thời gian đến (ToA - Time of Arrival), và sự khác biệt thời gian đến (TdoA - Time Difference of Arrival).

Nếu bộ chỉ thị độ lớn tín hiệu thu P_{tx} , mô hình suy hao đường truyền, và hệ số suy hao đường truyền α là đã biết, thì máy thu có thể sử dụng độ lớn tín hiệu thu được P_{rcvd} để có lời giải về khoảng cách d trong một phương trình suy hao đường truyền như sau:

$$P_{rcvd} = c \frac{P_{tx}}{d^\alpha} \Leftrightarrow d = \sqrt[\alpha]{c \frac{P_{tx}}{P_{rcvd}}}$$

Điều này là hấp dẫn bởi vì không cần phải thêm phần cứng nào và chúng ta có thể nhận được các ước lượng khoảng cách thậm chí không cần chi phí thêm cho việc truyền thông đang diễn ra. Tuy nhiên, nhược điểm là các giá trị RSSI không phải là hằng số nhưng có thể dao động mạnh, thậm chí khi máy thu và máy phát không di chuyển. Điều này gây ra bởi các tác động giống như fading nhanh và sự biến đổi của môi trường – phạm vi của lỗi có thể là $\pm 50\%$. Đối với mức độ nào đó, tác động này có thể bị mất tác dụng bởi các phép đo lặp lại và lọc đi các giá trị không đúng bằng các kỹ thuật thống kê. Thêm vào đó, các bộ thu phát vô tuyến đơn giản, rẻ tiền thường không được kiểm chuẩn và cùng một độ lớn tín hiệu thực tế có thể tạo ra các giá trị RSSI khác nhau trên các thiết bị khác nhau; tương tự, công suất truyền dẫn thực tế của bộ thu phát này cho thấy những sự trái ngược nhau trong công suất dự kiến. Một vấn đề thứ ba đó là sự có mặt của các vật cản kết hợp với fading đa đường. Ở đây, khi sự suy giảm của tín hiệu dọc theo một đường gián tiếp lớn hơn sự suy giảm của tín hiệu dọc theo một đường trực tiếp có thể dẫn đến việc giả thiết không chính xác một khoảng cách là dài hơn khoảng cách thực tế. Khi nó là một vấn đề cấu trúc, nó không thể bị tác động bởi các phép đo lặp đi lặp lại.



Hình 9.3 Bản chất ngẫu nhiên của việc ánh xạ các giá trị RSSI tới các khoảng cách [25].

Một sự xem xét chi tiết hơn chỉ ra rằng ánh xạ các giá trị RSSI tới các khoảng cách thực tế là một quá trình ngẫu nhiên. Ví dụ: Ramadurai và Sichitiu [25] thu thập, đối với vài khoảng cách, các ví dụ lặp đi lặp lại của các giá trị RSSI trong một thiết lập môi trường mở. Sau đó, họ đếm xem có bao nhiêu lần mỗi khoảng cách đưa ra một giá trị RSSI đã cho và tính toán mật độ của biến ngẫu nhiên này – Hình 9.3(a) trình bày hàm mật độ xác suất đối với một giá trị đơn đã cho của RSSI, Hình 9.3(b) cho nhiều giá trị. Nói riêng, thông tin được cung cấp bởi các giá trị RSSI nhỏ, chỉ thị các khoảng cách dài hơn, là bị hạn chế khi mật độ được trải rộng.

Suy ra, khi sử dụng RSSI như một kỹ thuật lập phạm vi, cần thiết phải chấp nhận và giải quyết các lỗi về phạm vi đáng kể hoặc để xử lý đầu ra của quá trình lập phạm vi như một kết quả không tiên định (stochastic) và để bắt đầu với kết quả đó.

Thời gian đến

Thời gian đến (ToA) (đôi khi được gọi là thời gian bay - “*time of flight*”) khai thác mối quan hệ giữa khoảng cách và thời gian truyền dẫn khi biết tốc độ truyền lan. Giả thiết rằng cả máy phát và máy thu đều biết thời gian này khi một sự truyền dẫn (ví dụ: một xung siêu âm ngắn) bắt đầu, thời gian đến của sự truyền dẫn tới máy thu có thể được sử dụng để tính toán thời gian truyền lan và do đó tính toán khoảng cách. Để làm nhẹ bớt trách nhiệm này của máy thu, máy thu có thể trả lại bất kỳ “xung đo” nhận được nào trong một thời gian tiên định; sau đó máy thu gốc chỉ phải đo thời gian của một vòng với giả thiết các đường là đối xứng.

Phụ thuộc vào môi trường truyền dẫn đang sử dụng, thời gian đến yêu cầu các đồng hồ phải có độ phân giải rất cao để tạo ra các kết quả có độ chính xác có thể chấp nhận được. Đối với các sóng âm, các yêu cầu giải pháp này là bình thường; tuy nhiên chúng là rất khó đối với sự truyền lan sóng vô tuyến.

Một nhược điểm của âm thanh đó là tốc độ truyền lan của nó phụ thuộc vào các nhân tố bên ngoài chẳng hạn như nhiệt độ hoặc độ ẩm – do đó cần thiết phải lấy chuẩn cẩn thận nhưng không bắt buộc.

Sự khác biệt thời gian đến (TDoA)

Để khắc phục sự cần phải có đồng bộ hiện, phương pháp **Sự khác biệt thời gian đến** (TDoA) sử dụng sự đồng bộ ẩn bằng cách cung cấp trực tiếp sự bắt đầu của thông tin truyền dẫn cho máy thu. Điều này có thể được thực hiện nếu sử dụng hai môi trường truyền dẫn có các tốc độ truyền lan rất khác nhau – ví dụ: các sóng vô tuyến truyền lan ở tốc độ ánh sáng và siêu âm, với tốc độ khác nhau khoảng 6 bậc biên độ (ở 21°C: tốc độ của ánh sáng trong chân không là 299.792.458 m/s, tốc độ sóng siêu âm trong không khí vào khoảng 344 m/s). Suy ra, khi một máy phát bắt đầu một truyền dẫn sóng siêu âm và một truyền dẫn sóng vô tuyến một cách đồng thời, máy thu có thể sử dụng sự đến của truyền dẫn sóng vô tuyến để bắt đầu đo thời gian cho đến khi truyền dẫn sóng siêu âm đến, bỏ qua một cách an toàn thời gian truyền lan của sự truyền thông vô tuyến. (Trong thực tế, điều này giống với điều xảy ra khi một người ước lượng khoảng cách tới một cơn bão có sấm bằng cách tính số giây giữa ánh chớp và tiếng sấm).

Nhược điểm hiển nhiên của phương pháp này là ở mỗi nút cần thiết phải có cả hai loại: máy phát và máy thu. Mặt khác, ưu điểm là độ chính xác tốt hơn nhiều so với các phương pháp dựa trên RSSI. Khái niệm này và các biến thể của nó được sử dụng trong nhiều nỗ lực nghiên cứu khác nhau và theo như công bố thì các độ chính xác giảm xuống tới 2 cm.

Thảo luận

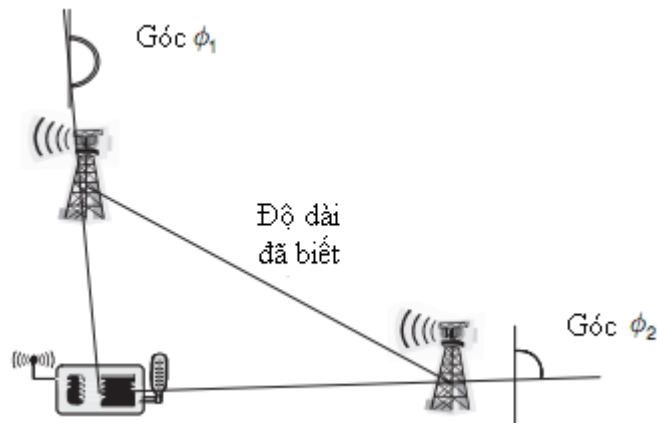
Có một sự không dung hòa rõ ràng giữa lỗi lập phạm vi và giá thành/độ phức tạp. Các phương pháp dựa trên RSSI là đơn giản hơn và làm việc với phần cứng mà nó yêu cầu; Các phương pháp dựa trên TDoA cung cấp các kết quả lập phạm vi cao hơn nhưng cần nhiều phần cứng hơn và phần cứng cũng phức tạp hơn, điều này cũng dẫn đến tiêu thụ năng lượng nhiều hơn. Do đó câu hỏi mở là liệu có thể giải quyết vấn đề xác định vị trí thực tế hoặc định vị dựa

trên các phép đo dễ xảy ra lỗi được các phương pháp dựa trên RSSI cung cấp hoặc liệu tổng chi phí cho TDoA là không thể tránh được.

Cần phải làm thêm nhiều công việc để việc thực hiện các phép đo phạm vi có độ tin cậy cao hơn, ví dụ: bằng cách sử dụng phạm vi âm học tương phản với độ suy giảm tần số vô tuyến hoặc bằng cách xem xét các ảnh hưởng được giới thiệu bởi các giá trị RSSI lượng tử hóa (thực tế thì các hệ thống gần gũi có thể xem như các lượng tử hóa có/không của RSSI).

Xác định các góc

Để thay thế cho việc đo các khoảng cách giữa các nút, người ta đo các góc. Góc này có thể hoặc là góc của một đường kết nối giữa một **nút neo** và một nút không biết vị trí với một hướng tham chiếu đã cho (“0° bắc”). Nó cũng có thể là góc giữa hai đường kết nối này nếu không có hướng tham chiếu nào tới tất cả các nút được biết chung (hình 9.4).



Hình 9.4 Góc tạo bởi hai nút neo

Một phương pháp truyền thống để đo các góc là sử dụng các anten có hướng (các anten mà chỉ gửi tin/ nhận tin theo một hướng đã cho), quay trên trục của chúng, giống như trạm radar hoặc một ngọn hải đăng truyền thống. Điều này làm cho các phép đo góc mang tính khái niệm đơn giản, nhưng các thiết bị này là khá không phù hợp đối với các nút cảm biến; chúng có thể là hữu ích đối với việc hỗ trợ các **nút neo** cơ sở.

Một kỹ thuật khác là khai thác tốc độ truyền lan hữu hạn của tất cả các sóng. Với nhiều anten gắn trên một thiết bị với khoảng cách giữa các anten là đã biết và đo sự khác biệt thời gian giữa sóng đến của tín hiệu tại các anten khác nhau, hướng từ đó một sóng tới thiết bị có thể được tính toán. Khoảng cách giữa các anten càng nhỏ, độ chính xác của sự khác biệt thời gian phải càng cao, điều này gây ra các yêu cầu định thời khó thực hiện khi cho trước kích thước nhỏ như mong muốn của các nút cảm biến.

Tóm lại, việc xác định góc là kỹ thuật ít được thảo luận so với kỹ thuật cạnh.

9.2.3 Phân tích cảnh

Một kỹ thuật khác đó là **phân tích cảnh**. Dạng của nó là phân tích các bức ảnh chụp bởi máy ảnh và cố gắng lấy được vị trí từ bức ảnh này. Điều này yêu cầu nỗ lực tính toán đáng kể và khó có thể phù hợp với các nút cảm biến.

Nhưng ngoại trừ các bức ảnh ảo, đặc điểm có thể đo được khác là “dấu vân tay” của một vị trí đã cho có thể được sử dụng cho việc phân tích cảnh, ví dụ: các mẫu truyền lan sóng

vô tuyến. Một tùy chọn là sử dụng các phép đo độ lớn tín hiệu của (một hoặc nhiều **nút neo**) đang truyền dẫn một độ lớn tín hiệu đã biết và so sánh các giá trị đo được thực tế với các giá trị được lưu trong cơ sở dữ liệu của các giá trị đo được khi mạng không hoạt động (off-line) trước đó của mỗi vị trí. Sử dụng các đặc điểm vật lý khác như đặc tính đa đường cũng có thể biết được.

Trong khi việc phân tích cảnh là đáng quan tâm đối với các hệ thống mà có pha triển khai dành riêng và nơi mà các phép đo khi mạng không hoạt động (off-line) được chấp nhận, điều này không luôn luôn xảy ra đối với các mạng WSN. Suy ra, phương pháp này không phải là tiêu điểm của sự quan tâm ở đây.

9.3 Cơ sở toán học cho các vấn đề cạnh

Do (nhiều) cạnh là một trong những các kỹ thuật phổ biến nhất cho việc xác định vị trí được áp dụng trong các mạng WSN và cung cấp như một kỹ thuật nguyên thủy cho một số các phương pháp được thảo luận phía sau. Chúng ta cần phân tích kỹ hơn các phép toán trong kỹ thuật này.

9.3.1 Giải pháp với ba nút neo và sửa lỗi các giá trị khoảng cách

Giả thiết rằng có ba nút neo với các vị trí đã biết (x, y_i) , $i = 1, \dots, 3$, một nút tại vị trí không biết (x_u, y_u) , và các giá trị khoảng cách hoàn hảo r_i , $i = 1, \dots, 3$. Từ lý thuyết Pytago, dẫn đến một hệ ba phương trình như sau:

$$(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \quad \text{với } i = 1, \dots, 3. \quad (9.1)$$

Để giải hệ các phương trình này, dễ dàng hơn khi viết nó như một hệ các phương trình tuyến tính theo x_u và y_u . Để làm điều này, phải loại bỏ các thuật ngữ bậc hai x_u^2 và y_u^2 . Điều này có thể nhận được bằng cách trừ phương trình thứ ba khỏi hai phương trình đầu, kết quả là còn hai phương trình:

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2 \quad (9.2)$$

$$(x_2 - x_u)^2 - (x_3 - x_u)^2 + (y_2 - y_u)^2 - (y_3 - y_u)^2 = r_2^2 - r_3^2 \quad (9.3)$$

Sắp xếp lại ta có:

$$2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \quad (9.4)$$

$$2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u = (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \quad (9.5)$$

Chúng ta có thể dễ dàng viết lại thành phương trình ma trận tuyến tính:

$$2 \begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix} \quad (9.6)$$

trong đó ma trận bên phía trái và bên phía phải chỉ bao gồm các hằng số đã biết.

Ví dụ 9.1 (Xác định vị trí) sử dụng các vị trí ví dụ của hình 9.2 – $(x_1, y_1) = (2, 1)$, $(x_2, y_2) = (5, 4)$, and $(x_3, y_3) = (8, 2)$ – với các khoảng cách giữa **các nút neo** và nút mà không biết vị trí $r_1 = \sqrt{10}$, $r_2 = 2$, $r_3 = 3$, phương trình 9.6 trở thành

$$2 \begin{bmatrix} 6 & 1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} 64 \\ 22 \end{bmatrix}, \quad (9.7)$$

Cho kết quả $x_u = 5$ and $y_u = 2$ là vị trí của các nút chưa biết.

9.3.2 Giải quyết vấn đề các lỗi khoảng cách

Thách thức thực sự đối với phép đặc tam giác tăng lên khi các phép đo khoảng cách không hoàn hảo nhưng chỉ các ước lượng \tilde{r} với một lỗi chưa biết ε là đã biết. Giải các phương trình trên với $\tilde{r}_i = r_i + \varepsilon_i$ nói chung là sẽ cho các giá trị đúng đối với các vị trí chưa biết (x_u, y_u) .

Giải pháp trực giác cho vấn đề này là sử dụng nhiều hơn ba **nút neo** và các phép đo khoảng cách dư thừa để tính toán lỗi trong mỗi phép đo cá nhân. Về mặt toán học, điều này biến các phương trình trên thành một hệ thống các phương trình xác định quá (overdetermined system of equations), được viết dưới dạng ma trận như sau:

$$2 \begin{bmatrix} x_n - x_1 & y_n - y_1 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (r_{n-1}^2 - r_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix} \quad (9.8)$$

Đối với một hệ thống xác định quá của phương trình tuyến tính như vậy, một giải pháp có thể được tính toán để giảm tối thiểu lỗi bình phương trung bình, nghĩa là giải pháp đó là cặp (x_u, y_u) mà tối thiểu hóa $\|\mathbf{Ax} - \mathbf{b}\|_2$, trong đó $0,5\mathbf{A}$ là ma trận bên tay trái (một ma trận $(n-1) \times 2$), $\mathbf{x} = (x_u, y_u)$ là viết tắt cho vecto mô tả vị trí chưa biết, và \mathbf{b} ở bên tay phải (một vecto hàng $n-1$) từ phương trình (9.8). Bởi vì $\|\cdot\|_2$, dạng chuẩn 2 (2-norm) của một vecto (căn bậc hai của tổng các bình phương của các thành phần vecto), được tối thiểu hóa, điều này phản ánh lời giải cho vị trí mà thỏa mãn tốt nhất, với lỗi trung bình nhỏ nhất, tất cả các sự ràng buộc vị trí từ tất cả n **nút neo**.

Để tìm ra một giải pháp cho vấn đề tối thiểu hóa này, hãy xem biểu diễn cho bình phương của 2-tiêu chuẩn ở trên. Hãy quan sát rằng đối với bất kỳ vecto \mathbf{v} nào, $\|\mathbf{v}\|_2^2 = \mathbf{v}^T \mathbf{v}$. Suy ra,

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 = (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - 2\mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b} \quad (9.9)$$

Tối thiểu hóa biểu diễn này tương đương với việc tối thiểu hóa lỗi bình phương trung bình. Về việc này như một hàm trong \mathbf{x} , gradient của nó phải được đặt bằng 0:

$$2\mathbf{A}^T \mathbf{Ax} - 2\mathbf{A}^T \mathbf{b} = 0 \leftrightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}. \quad (9.10)$$

Phương trình (9.10) được gọi là **phương trình chuẩn tắc** đối với vấn đề bình phương nhỏ nhất tuyến tính. Phương trình này có một lời giải duy nhất dưới các điều kiện nào đó (\mathbf{A} phải có bậc đầy đủ). Có nhiều phương pháp khác nhau để giải phương trình này, ví dụ: khai triển Cholesky.

Ví dụ 9.2 (các vị trí với thông tin không chính xác) minh họa khái niệm này. Hãy xem ví dụ trước đó, giả thiết rằng đối với ba **nút neo** chỉ có sẵn các ước lượng vị trí không chính xác $\tilde{r}_1 = 5$, $\tilde{r}_2 = 1$, $\tilde{r}_3 = 4$. Giải phương trình kết quả tương ứng với phương trình 9.7 cho vị trí không chính xác $(5,2; 4,8)$ với một khoảng cách là $\sqrt{(5,2-5)^2 + (4,2-2)^2} \approx 2,2$ giữa vị trí ước lượng và vị trí chính xác. (Chú ý là không cần giải pháp bình phương trung bình).

Thêm các **nút neo** vào $(x_4, y_4) = (3, 1)$, $(x_5, y_5) = (7, 5)$, $(x_6, y_6) = (2, 8)$, và $(x_7, y_7) = (4, 6)$ với các ước lượng khoảng cách $\tilde{r}_4 = 2$, $\tilde{r}_5 = 3$, $\tilde{r}_6 = 7$, và $\tilde{r}_7 = 4$, sẽ cải thiện sự ước lượng này. Thu được ma trận kết quả **A** và ma trận **b** phía bên tay phải là:

$$\mathbf{A} = \begin{bmatrix} 2 & 5 \\ -1 & 2 \\ -4 & 4 \\ 1 & 5 \\ -3 & 1 \\ 2 & -2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 56 \\ -4 \\ -16 \\ 30 \\ -29 \\ 17 \end{bmatrix} \quad (9.11)$$

Giải $\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$ cho \mathbf{x} thu được kết quả là $\mathbf{x} = (5,5; 2,7)$, với một lỗi khoảng cách bằng $(5,5 - 5)^2 + (2,7 - 2)^2 \approx 0,86$.

Suy ra, hình thức này cho phép tính toán vị trí với lỗi bình phương trung bình nhỏ nhất ngoài $n \geq 3$ **nút neo** với sự có mặt của các lỗi trong các phép đo khoảng cách. Sự khái quát cho ba chiều là hiển nhiên, ví dụ: người ta thảo luận bằng cách nào để đưa sự ước lượng tốc độ truyền lan chưa biết (trong hoàn cảnh của một hệ thống lập phạm vi TDoA) vào trong vấn đề tối ưu hóa này.

9.4 Xác định vị trí trong các môi trường đa bước nhảy

Tất cả các phương pháp và các khái niệm trong việc xác định vị trí trong môi trường một bước nhảy đều giả thuyết rằng một nút đang cố gắng xác định vị trí của nó có thể truyền thông trực tiếp nói chung với vài **nút neo**. Giả thuyết này không luôn luôn đúng trong một mạng cảm biến không dây – không phải tất cả các nút có giao tiếp trực tiếp với ít nhất **ba nút neo**. Các cơ chế là cần thiết, có thể bằng cách này hay cách khác đối mặt với sự hạn chế về mặt địa lý có sẵn của phạm vi (tương đối) chính xác hoặc thông tin vị trí. Các cơ chế này và các phương pháp được mô tả ở đây. Trong dạng này hay dạng khác, các phương pháp đều dựa trên sự thật rằng đối với một biểu đồ kết nối đầy đủ với chiều dài của các cạnh đã biết, người ta có thể tái cấu trúc việc nhúng nó trong mặt phẳng (hoặc trong không gian ba chiều).

9.4.1 Sự kết nối trong một mạng đa bước nhảy

Một chương trình bán xác định công thức khả thi

Phương pháp đầu tiên [10] đối với vấn đề xác định vị trí đa bước nhảy (chủ yếu) dựa trên thông tin kết nối và xem xét việc xác định vị trí như là một vấn đề khả thi. Giả thiết rằng các vị trí của n **nút neo** đều đã được biết và các vị trí của m nút là sắp được xác định, giả thiết rằng kết nối giữa hai nút bất kỳ chỉ có khả năng xảy ra nếu các nút cách nhau tối đa R đơn vị khoảng cách, và rằng đã biết kết nối giữa hai nút bất kỳ. Sự thật là hai nút được nối đưa ra một sự ràng buộc đối với vấn đề khả thi – đối với hai nút được kết nối, người ta không thể chọn các vị trí khi mà đặt các nút xa hơn khoảng cách R . Đối với một nút đơn, nhiều yêu cầu bắt buộc có thể tồn tại đến nỗi mà các yêu cầu này phải được thỏa mãn một cách đồng thời – hơi giống với các vòng tròn chồng nhau như ở trên, các vòng tròn mà giới hạn các vị trí có thể của một nút.

Trên cơ sở của công thức này, Doherty [10] cung cấp một thông tin về vấn đề khả thi như một chương trình bán định nghĩa (một khái quát của các chương trình tuyến tính). Vấn đề này có thể được giải quyết, nhưng chỉ một cách tập trung, yêu cầu tất cả các thông tin kết nối tại một điểm.

Sự quan sát chủ yếu ở đây đó là trong công thức này, sự thật là hai nút không được kết nối không cung cấp bất cứ thông tin thêm nào – người ta không thể viết ra một yêu cầu bắt buộc trong một chương trình bán định nghĩa rằng hai nút cách nhau ít nhất một khoảng cách đã cho; các nút có thể là “được đẩy ra xa”. Như một ví dụ hệ quả, một chuỗi tuyến tính các nút với chỉ một **nút neo** trong chuỗi có thể được phân biệt từ một tình huống mà trong đó tất cả các nút tụ họp lại quanh **nút neo**. Điều này ngụ ý rằng các **nút neo** nên được đặt một cách khéo léo hơn tại các đường biên của mạng, để áp đặt càng nhiều yêu cầu bắt buộc “kéo ra xa” càng tốt. Việc sắp đặt có điều khiển như vậy làm giảm đáng kể lỗi xác định vị trí trung bình so với việc sắp đặt **nút neo** một cách ngẫu nhiên. Doherty [10] cũng thảo luận các biến đổi và các mở rộng của ý tưởng chính này. Nói riêng thì việc sử dụng các ước lượng của khoảng cách thực tế này chỉ thay cho giới hạn trên xuất phát từ kết nối; thông tin góc thay thế cho các phạm vi (lấy cảm hứng từ thông tin quang trực tiếp); và tính toán các giới hạn của lỗi vị trí bằng cách giải quyết nhiều vấn đề khả thi. Tuy nhiên, do đặc điểm tập trung hóa chủ yếu của nó, khái niệm này chỉ được áp dụng có giới hạn cho các WSN.

Chia tỷ lệ đa chiều

Cùng một vấn đề cơ bản của phạm vi tự do, việc định vị dựa trên kết nối được Shang [28] giải quyết sử dụng công thức toán học **Chia tỷ lệ đa chiều (MDS - MultiDimensional Scaling)**. Trên cơ sở của kết nối giữa các nút, một thuật toán đường truyền ngắn nhất cho tất cả các cặp ước lượng xấp xỉ các vị trí của các nút. MDS cải thiện ước lượng ban đầu này, và nếu các nút với thông tin vị trí tuyệt đối là có sẵn, thì các tọa độ kết quả được chuẩn hóa một cách chính xác.

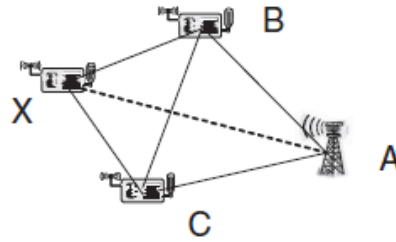
Tham khảo tài liệu [28] để biết thêm chi tiết về kỹ thuật toán học này. Ưu điểm chính của phương pháp này đó là nó khá ổn định đối với việc sắp đặt **nút neo**, nhận được các kết quả tốt thậm chí chỉ có sẵn vài **nút neo** hoặc vài nút neo được đặt, ví dụ như trong mạng này. Thêm vào đó người ta thấy rằng MDS cũng phù hợp với các mạng không đẳng hướng (các mạng mà trong đó khoảng cách giữa các nút lân cận không như nhau trên phần mở rộng của mạng đó).

9.4.2 Việc ước lượng phạm vi đa bước nhảy

Phương pháp đa bên cơ sở (multilateration) yêu cầu một nút phải có các ước lượng phạm vi tới ít nhất ba **nút neo** để cho phép nó ước lượng vị trí của bản thân nó. Niculescu và Nath [22] xem xét vấn đề này khi **các nút neo** không thể cung cấp các ước lượng phạm vi cho tất cả các nút trong mạng, mà chỉ cung cấp cho các nút lân cận trực tiếp của chúng (ví dụ: do các giới hạn về công suất truyền dẫn). Ý tưởng là sử dụng sự ước lượng phạm vi gián tiếp bằng truyền thông đa bước nhảy có khả năng tái sử dụng thuật toán đa bên nổi tiếng.

Để tính toán các ước lượng phạm vi giữa một nút và một **nút neo** ở xa thông qua nhiều bước nhảy gián tiếp, Niculescu và Nath [22] mô tả ba khả năng khác nhau. Tất cả các khả năng này đều dựa trên việc làm tràn ngập mạng với thông tin, bắt đầu một cách độc lập từ mỗi **nút neo**, giống với hoạt động của một giao thức định tuyến vectơ khoảng cách (DV - Distance Vector).

Khả năng đơn giản nhất là phương pháp “nhảy DV”. Ý tưởng là để tính toán số lượng các bước nhảy (dọc theo đường ngắn nhất) giữa hai **nút neo** bất kỳ và sử dụng nó để ước lượng chiều dài trung bình của một bước nhảy đơn bằng cách chia tổng các khoảng cách tới **các nút neo** khác cho tổng của các bước nhảy đếm được. Tất cả **nút neo** tính toán chiều dài bước nhảy ước tính này và truyền nó vào trong mạng. Một nút với vị trí chưa biết sau đó có thể sử dụng chiều dài bước nhảy ước tính này (và số lượng các bước nhảy đã biết đối với các **nút neo** khác) để tính toán một ước lượng phạm vi đa bước nhảy và thực hiện đa bên. Chú ý rằng thực tế đó là một phương pháp phạm vi tự do (range-free) bởi vì không cần thiết phải ước lượng các khoảng cách giữa các nút.



Hình 9.5 Ước lượng khoảng cách Euclidean trong sự vắng mặt của kết nối trực tiếp [22]

Khi các ước lượng phạm vi giữa các nút lân cận là có sẵn, chúng có thể được sử dụng một cách trực tiếp trong cùng một cơ cấu tổ chức, kết quả là phương pháp “Khoảng cách DV”.

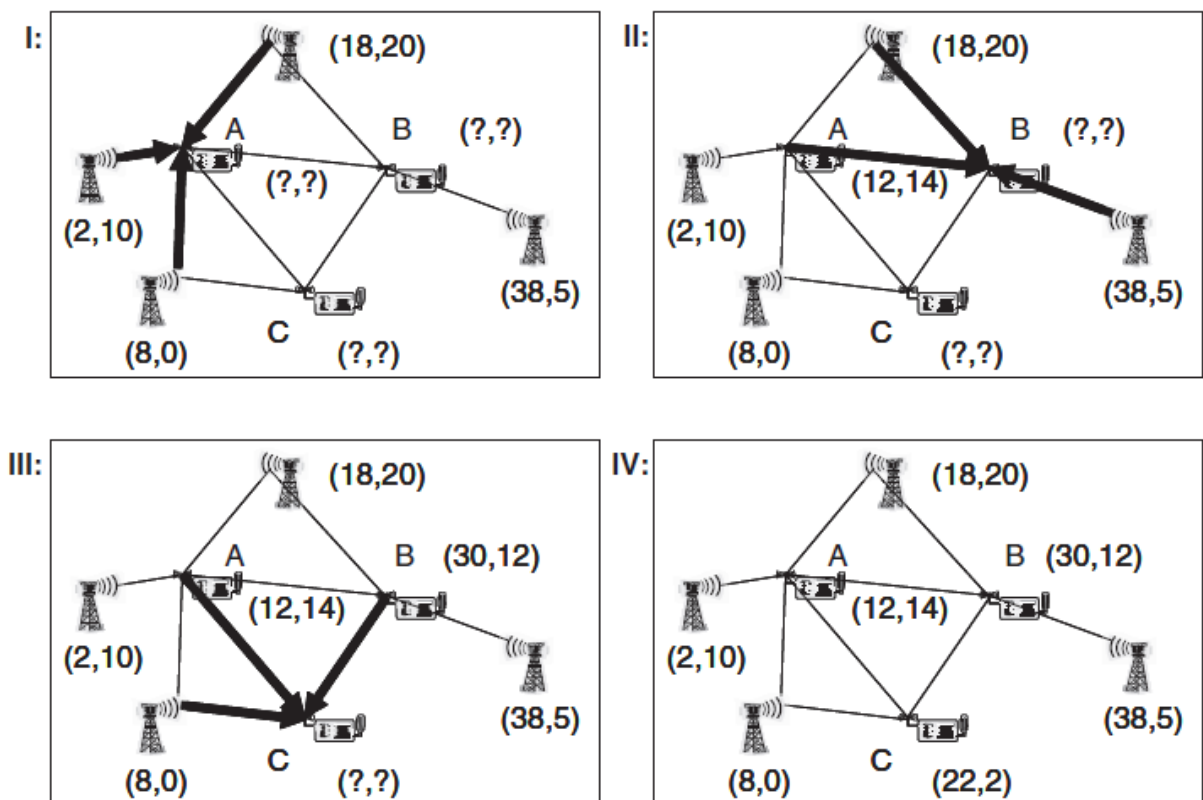
Với sự có mặt của các ước lượng phạm vi và một lượng đủ các nút lân cận, một nút có thể thực sự cố gắng để tính toán khoảng cách Euclidean thực sự của nó tới một **nút neo** ở xa. Hình 9.5 minh họa ý tưởng này: giả thuyết rằng các khoảng cách AB, AC, BC, XB, XC tất cả đều đã biết, người ta có thể tính toán khoảng cách chưa biết XA (thực tế, có hai giải pháp, một là nút X nằm ở phía khác của đường BC – nút X có khả năng phân biệt hai giải pháp này dựa trên thông tin cục bộ). Theo cách này, các vị trí thực sự có thể được chuyển tiếp giữa các nút.

Độ chính xác có thể nhận được ở đây phụ thuộc vào tỷ số của **các nút neo** so với tổng số các nút. Phương pháp “Euclidean” làm tăng độ chính xác khi số lượng của **các nút neo** tăng lên; “vector khoảng cách” – giống như các phương pháp phù hợp tốt hơn với một tỷ số thấp của **các nút neo**. Như dự kiến, các phương pháp vector khoảng cách thực hiện không tốt trong các mạng không đẳng hướng so với trong các mạng phân bố đồng đều; mặt khác, phương pháp Euclidean không thực sự nhạy cảm với tác động này.

9.4.3 Đa bên cộng tác và lặp

Phương pháp trước đây cố gắng ước lượng các khoảng cách giữa các nút có vị trí chưa biết và **các nút neo** để áp dụng sự đa bên với bản thân **các nút neo**. Một phương pháp thay thế đó là sử dụng các nút bình thường, mỗi khi các nút này ước lượng các vị trí của chúng, giống như **các nút neo** trong một thuật toán đa bên. Hình 9.6 chỉ ra một ví dụ: các nút A, B, C là không biết vị trí của chúng. Nút A có thể lập lưới tam giác vị trí của bản thân nó sử dụng ba **nút neo**. Mỗi khi nút A có một ước lượng vị trí, nút B có thể sử dụng nó và hai **nút neo** cho sự ước lượng của bản thân nó, lần lượt cung cấp nút C với thông tin bị mất cho phép đặc tam giác của bản thân nó. Ý tưởng cơ bản cho **đa bên lặp** được đề xuất bởi Savarese [27].

Một sự thực hiện tập trung là ít quan trọng, thường bắt đầu với nút có vị trí chưa được xác định, nút mà có phần lớn các kết nối với **các nút neo/các nút** với các ước lượng vị trí đã được xác định và tính toán lặp nhiều ước lượng vị trí hơn. Trong một sự thực hiện phân tán, các nút có thể tính toán một ước lượng vị trí mỗi khi ít nhất ba nút lân cận có thể cung cấp thông tin vị trí, cho kết quả là một ước lượng khởi đầu về vị trí của nút. Khi có sẵn nhiều thông tin hơn – ví dụ: bởi vì nhiều nút lân cận ước lượng vị trí của bản thân chúng - người ta có thể sử dụng nó để cải thiện ước lượng vị trí và truyền một ước lượng vị trí cập nhật tới một nút lân cận của nút đó. Hy vọng là thuật toán này sẽ hội tụ tới tập hợp các vị trí đúng đối với tất cả các nút. Lỗi vị trí trung bình sau một bộ lọc lặp như vậy phụ thuộc vào độ chính xác của sự ước lượng phạm vi, sự ước lượng vị trí khởi đầu, số lượng trung bình của các nút lân cận, và vào số lượng của các nút neo [27]. Nó cũng không đảm bảo một chút nào rằng thuật toán lọc sẽ hội tụ; có thể có các tình huống trong đó lỗi vị trí làm tăng thời gian chạy của thuật toán. Thực tế là thuật toán lọc không phức tạp sẽ không cho kết quả là một hiệu suất có thể chấp nhận được. Điều này chủ yếu là do sự truyền lan lỗi qua toàn mạng. Để cải thiện, Savarese [27] gợi ý đưa thêm các ảnh hưởng tin cậy vào tất cả các ước lượng vị trí và để giải quyết một vấn đề tối ưu hóa trọng số biến đổi, và cho kết quả là sự hội tụ của gần như tất cả các kịch bản.

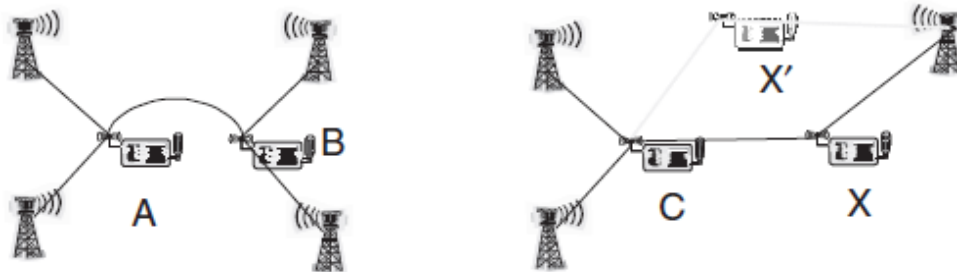


Hình 9.6 Đa bên lặp: các nút A, B, và C có thể xác định các vị trí của nó trong vài phép lặp (các mũi tên đậm chỉ các đường nối mà phạm vi của chúng được sử dụng trong một phép lặp đã cho)

Một thách thức đặc biệt đối với lớp thuật toán này xảy ra khi không phải tất cả các nút trong mạng sẽ có ba nút với các ước lượng vị trí. Điều này là dễ dàng để phát hiện một nút

đơn, nhưng khó thực hiện cho một nhóm các nút. Tuy nhiên, phụ thuộc vào cấu trúc liên kết mạng (tô pô mạng), nó vẫn có thể ước lượng ít nhất vài vị trí - gọi là **đa biên cộng tác**.

Hình 9.7 minh họa hai trường hợp đang tranh luận. Kịch bản phía bên trái vẫn được xác định đầy đủ khi có sẵn thông tin đầy đủ để giải hệ phương trình cho hai nút với vị trí chưa biết. Đối với kịch bản bên phải, có hai giải pháp (X và X') cho vị trí của nút X , mà không thể phân biệt được, nhưng vị trí của nút chưa biết khác vẫn có thể xác định được.



Hình 9.7 Các kịch bản còn phải bàn cho đa bên lặp

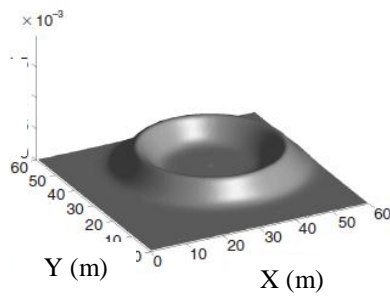
Người ta tiếp cận các trường hợp còn đang tranh luận này bằng cách xác định “các nút tham gia” – các nút mà có ít nhất ba **nút neo** hoặc các nút tham gia khác như các nút lân cận, làm cho nút A và B trong hình 9.7 trở thành các nút tham gia. Đối với các nút tham gia này, việc xác định vị trí có thể được giải quyết.

Savarese [27] dùng một cách tiếp cận hơi khác ở đây. Việc quan sát chủ yếu đó là một nút, để xác định vị trí của nó, cần ít nhất ba tham chiếu *độc lập* đối với các nút neo – các đường tới các nút neo phải là các cạnh tách rời nhau. Các nút như vậy được gọi là *âm thanh* (*sound*). Trong hình 9.7, các nút A, B, C đều là *âm thanh*. Sự đứng đắn có thể được phát hiện trong ước lượng vị trí khởi đầu, ví dụ: bằng cách ghi lại qua nút lân cận nào thì đường ngắn nhất tới một nút neo đã cho mở rộng ra. Nếu ba hoặc nhiều hơn các đường này được phát hiện, nút này tuyên bố âm thanh của bản thân nó và truy cập vào pha lọc. Ví dụ: nút X từ hình 9.7 sẽ không thể tuyên bố âm thanh của bản thân nó. Hậu quả là quy trình “âm thanh” sẽ có thể định vị nhiều nút hơn nút tham gia. Thuật toán trong tài liệu tham khảo [27] cũng phù hợp tốt hơn đối với các tỷ số nút neo thấp và có thể xác định các vị trí của nhiều nút hơn.

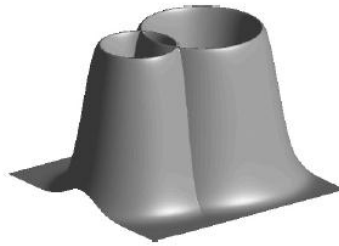
9.4.4 Mô tả và truyền lan việc xác định vị trí theo xác suất

Các phương pháp trước đều mô tả vị trí của một nút mỗi khi nó được xác định, bằng một tập hợp các tọa độ rõ ràng. Việc mô tả “tiền định - deterministic” này làm sụp đổ tính ngẫu nhiên vốn có, ví dụ: gây ra bởi các sự không chắc chắn trong các ước lượng phạm vi. Ramadurai và Sichitiu [25] cố gắng tính đến một cách rõ ràng tính ngẫu nhiên này bằng cách mô tả vị trí của một nút bằng một hàm xác suất của vị trí có thể của nút đó, mô tả lượng thông tin có sẵn cho vị trí của nút này. Ban đầu, một nút có thể ở tất cả các vị trí với xác suất như nhau. Một phép đo khoảng cách cụ thể, ví dụ: một giá trị RSSI, cung cấp sự tăng cho một hàm mật độ xác suất, liên hệ mỗi khoảng cách với một xác suất nào đó và với xác suất này nó tương ứng với giá trị RSSI đó (Hình 9.3). Suy ra, một nút đo một giá trị RSSI nào đó từ một nút neo biết rằng nó có một xác suất cao là nó ở một nơi nào đó trong một vòng tròn xung quanh nút neo (Hình 9.8(a)). Mỗi khi thông tin từ một nút neo thứ hai trở nên sẵn có (Hình

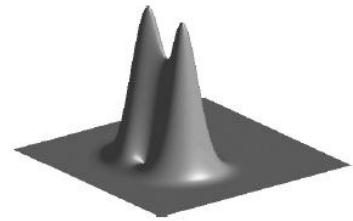
9.8(b)), thì hai hàm mật độ có thể được chập lại và cho kết quả là một mô tả của các xác suất về vị trí của nút được cải thiện (Hình 9.8(c)).



(a) Hàm mật độ xác suất của một nút xác định vị trí sau khi nhận được một ước lượng khoảng cách từ một nút neo



(b) Các hàm mật độ xác suất của hai phép đo khoảng cách từ hai nút neo độc lập



(c) Hàm mật độ xác suất của một nút sau khi cho giao nhau các phép đo khoảng cách của hai nút neo

Hình 9.8 Sự xử lý theo xác suất các vị trí của nút [25]

Ramadurai và Sichitiu [25] mô tả chi tiết theo toán học các hoạt động và công thức cho việc mô tả các hàm xác suất này. Lỗi xác định vị trí thu được biến đổi đáng kể giữa các nút khác nhau, có phạm vi từ cực lớn đến khá lớn.

9.5 Tổng kết

Việc xác định các vị trí – và, một mức độ kém hơn (xác định các định vị) – trong một mạng cảm biến không dây là gánh nặng đáng kể đối với tổng chi phí và nguy cơ của các kết quả không chính xác tuyệt đối và không chính xác tương đối. Một số lượng đáng kể các nút neo là cần thiết cho các hệ thống tọa độ toàn cầu, thời gian và phân tiếp đầu của bản tin là cần thiết để tính toán các vị trí nếu không có sẵn sự truyền thông trực tiếp giữa các nút neo và các nút sẵn có sẽ không được đưa vào để ước lượng. Tuy nhiên, người ta có thể rút ra từ các phép đo có lỗi một mức độ thường xuyên hài lòng của các ước lượng vị trí.

CÂU HỎI ÔN TẬP CHƯƠNG 9

1. Các đặc điểm của quy trình định vị và xác định vị trí?
2. Giới thiệu khái quát các phương pháp định vị?
3. Cơ sở toán học cho các vấn đề cạnh?
4. Sự kết nối trong mạng đa bước nhảy?
5. Cách xác định cạnh?
6. Cách xác định góc?
7. Xác định vị trí theo xác suất?
8. Vấn đề ước lượng phạm vi đa bước nhảy?

CHƯƠNG 10: ĐIỀU KHIỂN CẤU TRÚC LIÊN KẾT MẠNG (TOPOLOGY)

10.1 Mục đích của chương

Trong một mạng không dây được triển khai dày đặc, một nút đơn có nhiều nút lân cận. Việc truyền thông trực tiếp với các nút lân cận này là có thể thực hiện được khi sử dụng công suất truyền dẫn đủ lớn. Tuy nhiên, điều này không thực sự mang lại lợi ích bởi vì: công suất truyền dẫn cao yêu cầu nhiều năng lượng, nhiều nút lân cận là một gánh nặng cho một giao thức MAC, và các giao thức định tuyến phải chịu sự không ổn định trong mạng khi các nút di chuyển xung quanh mạng, thường xuyên tạo nên và tách ra nhiều đường kết nối.

Để khắc phục các vấn đề này, người ta áp dụng *việc điều khiển cấu trúc liên kết mạng*. Ý tưởng là để giới hạn một cách có tính toán tập hợp các nút lân cận của một nút đã cho. Điều này có thể được thực hiện bằng cách điều khiển công suất phát, bằng cách giới thiệu kiến trúc phân tầng trong mạng và báo hiệu vài nút để tiếp quản các nhiệm vụ kết hợp nào đó, hoặc bằng cách đơn giản là tắt đi một số nút trong một thời gian nào đó.

Các ý tưởng cơ bản

Một đặc điểm có lẽ là điển hình của các mạng cảm biến không dây đó là khả năng triển khai nhiều nút trong một khu vực nhỏ, ví dụ: là để đảm bảo việc phủ sóng đủ cho một khu vực hoặc để có sự dư thừa trong mạng nhằm bảo vệ chống lại các lỗi của nút. Trong khi đây là các ưu điểm hiển nhiên của sự triển khai mạng *dày đặc* – ví dụ: mật độ được đo đặc bằng số lượng các nút lân cận trung bình mà một nút đơn có – thì nó cũng có các nhược điểm. Trong một mạng mật độ tương đối cao (Hình 10.1), nhiều vấn đề mạng không dây điển hình trở nên trầm trọng thêm bởi một số lượng lớn các nút lân cận: nhiều nút gây nhiễu cho các nút khác, có nhiều tuyến đường có thể sử dụng được, các nút có thể sử dụng một cách dư thừa công suất truyền dẫn lớn để nói chuyện với các nút ở xa một cách trực tiếp (nó cũng hạn chế việc tái sử dụng băng thông không dây), và các giao thức định tuyến có thể phải tính toán lại các tuyến đường thậm chí chỉ khi có những di chuyển nhỏ của nút xảy ra.

Một số vấn đề trong số các vấn đề trên có thể được giải quyết bằng các kỹ thuật điều khiển cấu trúc liên kết mạng (**topology-control**). Thay vì sử dụng kết nối có thể của một mạng cho sự mở rộng cực đại có thể của nó, người ta thực hiện một lựa chọn thận trọng để hạn chế cấu trúc liên kết của mạng. Cấu trúc liên kết của một mạng được xác định bởi tập con của các nút tích cực và tập hợp của các đường truyền tích cực mà dọc theo các đường truyền này có thể xảy ra sự truyền thông trực tiếp. Nói một cách chính quy, một thuật toán điều khiển cấu trúc liên kết mạng lấy một sơ đồ $G = (V, E)$ để biểu diễn mạng – nơi mà V là tập hợp của tất cả các nút trong mạng và có một cạnh $(v_1, v_2) \in E \subseteq V^2$ nếu và chỉ nếu các nút v_1 và v_2 có thể giao tiếp trực tiếp với nhau – và biến đổi nó thành một sơ đồ $T = (V_T, E_T)$ sao cho $V_T \subseteq V$ and $E_T \subseteq E$.



Hình 10.1 Cấu trúc liên kết mạng trong một mạng cảm biến không dây được triển khai với mật độ cao.

10.1.1 Các tùy chọn cho việc điều khiển cấu trúc liên kết mạng

Để tính toán một sơ đồ T đã được biến đổi từ một sơ đồ G biểu diễn mạng gốc G , một thuật toán điều khiển cấu trúc liên kết mạng có vài tùy chọn:

- Tập hợp các nút tích cực có thể giảm xuống ($V_T \subset V$), ví dụ: bằng cách tắt đi một cách có chu kỳ các nút có dự trữ năng lượng thấp và kích hoạt các nút khác thay thế, lợi dụng việc triển khai dư thừa các nút để làm việc này.
- Tập hợp các nút lân cận/ tập hợp các đường truyền tích cực cho một nút có thể được điều khiển. Thay vì sử dụng tất cả các đường truyền trong mạng, một số đường truyền có thể không cần quan tâm nhiều và sự truyền thông được giới hạn trong các đường tới hạn.

Khi người ta mong muốn có một cấu trúc liên kết mạng *phẳng* (tất cả các nút được xem là ngang nhau), tập hợp của các nút lân cận của một nút có thể được giảm đi bằng cách đơn giản là không truyền thông với vài nút lân cận. Có vài phương pháp có thể để chọn các nút lân cận, nhưng một phương pháp hiển nhiên có triển vọng đối với một WSN là hạn chế *việc tiếp cận* các truyền dẫn của một nút – điển hình là bằng *điều khiển công suất*, ngoài ra còn bằng cách sử dụng các điều chế thích nghi (sử dụng các điều chế nhanh hơn chỉ có thể cho các khoảng cách ngắn hơn) – và sử dụng hiệu suất năng lượng được cải thiện chỉ khi truyền thông với các nút lân cận gần đó.

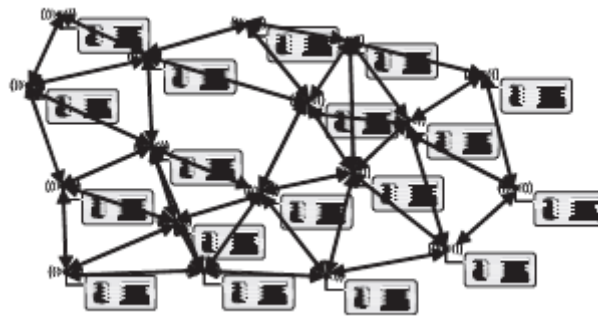
Hình 10.2 minh họa làm cách nào mà cấu trúc liên kết mạng mật độ cao như ở hình 10.1 có thể bị giảm xuống bằng cách áp dụng việc điều khiển công suất. Thực chất, việc điều khiển công suất cố gắng tối ưu hóa sự không dung hòa giữa một mặt là khả năng có thể xảy ra cao của việc tìm kiếm một máy thu (có ích) tại các giá trị công suất cao hơn, và mặt khác là tăng cơ hội xảy ra các xung đột/nhiều/giảm việc tái sử dụng không gian.

- Các đường truyền/các nút lân cận cũng có thể được tái sắp xếp lại trong một cấu trúc liên kết mạng có phân tầng, trong mạng này vài nút giả thiết là có vai trò đặc biệt. Một ví dụ, được minh họa ở hình 10.3, một số nút được lựa chọn như là một “xương sống” cho mạng và chỉ sử dụng các đường kết nối trong xương sống này và các đường kết nối trực tiếp từ các nút khác tới xương sống. Để làm điều này, xương

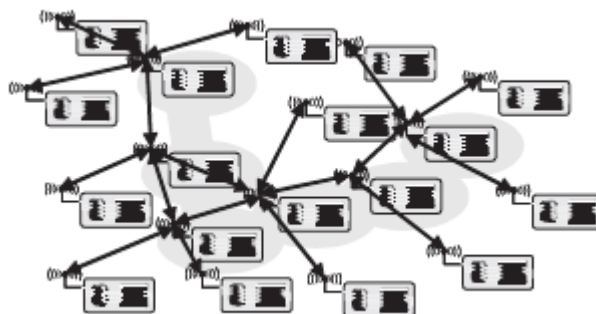
sống phải hình thành một **tập hợp chi phối**: một tập con $D \subset V$ sao cho tất cả các nút trong V đều hoặc trong bản thân D hoặc là các nút lân cận một bước nhảy của nút nào đó $d \in D$ ($\forall v \in V : v \in D \vee \exists d \in D : (v, d) \in E$). Sau đó, chỉ các đường kết nối giữa các nút của tập hợp chi phối hoặc giữa các nút khác và một thành viên của tập hợp tích cực được duy trì. Để một xương sống là có ích thì nó phải được **kết nối**.

Một ý tưởng có liên quan nhưng hơi khác đó là chia mạng này thành các **phân nhóm** (hình 10.4). Các phân nhóm là các tập con của các nút mà bao gồm cùng nhau tất cả các nút của sơ đồ gốc sao cho đối với mỗi phân nhóm, các điều kiện nào đó được giữ lại (các chi tiết thì có thể biến đổi). Vấn đề điển hình nhất của việc tạo thành các phân nhóm là để tìm kiếm các phân nhóm với **các chỉ huy phân nhóm** – một đại diện của một phân nhóm sao cho mỗi nút chỉ cách chỉ huy phân nhóm của nó một bước nhảy. Khi số lượng (trung bình) của các nút trong một phân nhóm được tối thiểu hóa, nó tương đương với việc tìm kiếm một **tập hợp độc lập** (chi phối) cực đại (một tập con $C \subset V$ sao cho $\forall v \in V - C : \exists c \in C : (v, c) \in E$ và không có hai nút nào trong C được kết hợp bởi một cạnh trong $E - \forall c1, c2 \in C : (c1, c2) \notin E$). Trong một mạng chia thành các phân nhóm như vậy, chỉ có các đường kết nối trong một phân nhóm là được duy trì (điển hình là chỉ các đường kết nối mà liên quan đến chỉ huy phân nhóm) như các đường kết nối được lựa chọn giữa các phân nhóm để đảm bảo kết nối cho toàn bộ mạng.

Cả hai vấn đề đều là các xấp xỉ biến đổi và thực sự khó và cần được nghiên cứu dưới dạng đơn giản hơn.

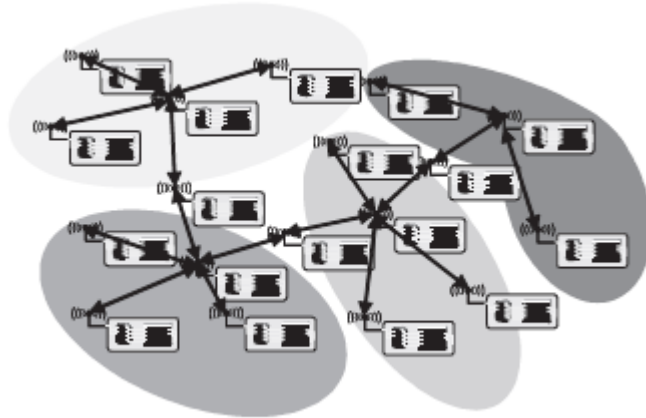


Hình 10.2 Cấu trúc liên kết mạng thưa hơn sau khi giảm công suất phát.



Hình 10.3 Giới hạn cấu trúc liên kết mạng bằng cách sử dụng một xương sống.

Ba tùy chọn chủ yếu này đối với việc điều khiển cấu trúc liên kết mạng – một là các mạng phẳng với sự tập trung đặc biệt vào việc điều khiển công suất, hai là các kiến trúc mạng phân tầng với các xương sống, ba là các phân nhóm – sẽ được nghiên cứu chi tiết hơn tương ứng trong các phần 10.2, 10.3, và 10.4. Đầu tiên, một vài khía cạnh mong muốn của các thuật toán điều khiển cấu trúc liên kết mạng sẽ được thảo luận.



Hình 10.4 Sử dụng các phân nhóm để chia một sơ đồ thành các phần.

10.1.2 Các khía cạnh của các thuật toán điều khiển cấu trúc liên kết mạng

Có vài tham số đo cơ bản để đánh giá hiệu quả và chất lượng của một thuật toán điều khiển cấu trúc liên kết mạng:

Sự kết nối: điều khiển cấu trúc liên kết mạng không được ngắt kết nối của một sơ đồ G đã được kết nối. Nói cách khác, nếu có một đường (đa bước nhảy) trong G giữa hai nút u và v , thì cũng phải có một đường nào đó như vậy trong T (rõ ràng là nó không nhất thiết phải là cùng một đường).

Các hệ số dẫn: việc loại bỏ các đường kết nối khỏi một sơ đồ sẽ có thể tăng chiều dài một đường nối giữa hai nút bất kỳ u và v . Hệ số dẫn bước nhảy được định nghĩa là việc tăng lên tối đa nhất trong chiều dài đường kết nối đối với bất kỳ cặp nút u và v nào giữa sơ đồ gốc G và đường kết nối điều khiển cấu trúc liên kết mạng T . Một cách chính thức,

$$\text{Hệ số dẫn bước nhảy} = \max_{u,v \in V} \frac{|(u,v)_T|}{|(u,v)_G|} \quad (10.1)$$

Trong đó $(u,v)_G$ là đường ngắn nhất trong sơ đồ và $|(u,v)|$ là chiều dài của nó.

Tương tự, **hệ số dẫn năng lượng** có thể được xác định:

$$\text{Hệ số dẫn năng lượng} = \max_{u,v \in V} \frac{E_T(u,v)}{E_G(u,v)} \quad (10.2)$$

Trong đó $E_G(u,v)$ là năng lượng được tiêu thụ dọc theo đường truyền có hiệu suất năng lượng cao nhất trong sơ đồ G . Rõ ràng là người ta mong muốn các thuật toán điều khiển cấu trúc liên kết mạng với các hệ số dẫn nhỏ.

Các tham số của sơ đồ (Graph metrics): các ví dụ trực quan trên chỉ ra tầm quan trọng của một số lượng nhỏ các cạnh trong T và một mức độ cực đại nhỏ (số lượng của các nút lân cận) cho mỗi nút.

Thông lượng: cấu trúc liên kết mạng giảm xuống phải có khả năng duy trì một lưu lượng tương đương với một mạng gốc (điều này có thể là quan trọng thậm chí trong các mạng cảm biến không dây với lưu lượng trung bình thấp, đặc biệt trong trường hợp có các trận mưa sự kiện). Một tham số đo biểu thị khía cạnh này là **tính đua tranh thông lượng** ($\varphi \leq 1$ lớn nhất sao cho, cho một tập hợp các dòng chảy từ một nút s_i tới nút d_i với tốc độ mà có thể định tuyến trong G , tập hợp với các tốc độ φr_i có thể được định tuyến trong T), xem tài liệu tham khảo [24] để biết thêm chi tiết.

Sức mạnh của sự di động: khi các mối quan hệ lân cận thay đổi trong sơ đồ gốc G (ví dụ : bởi vì các nút di chuyển xung quanh hoặc đặc điểm của các kênh vô tuyến thay đổi), một số nút khác có thể phải thay đổi thông tin cấu hình liên kết mạng của chúng (ví dụ: đối với các đường kết nối tái kích hoạt). Rõ ràng là, một cấu trúc liên kết mạng mạnh chỉ nên yêu cầu một lượng nhỏ các sự thích nghi này và tránh các ảnh hưởng tái tổ chức của một dao động do sự di chuyển của nút cục bộ trên toàn bộ mạng.

Tổng chi phí của thuật toán: tổng chi phí bởi bản thân thuật toán là nhỏ (do số lượng nhỏ của các bản tin cần phải thêm vào, tổng chi phí tính toán là nhỏ). Cũng như vậy, việc thực hiện phân tán thực tế là một điều kiện hình sin.

Trong bối cảnh hiện tại của các WSN, các hệ số kết nối và hệ số dẫn có lẽ là các đặc điểm quan trọng nhất của một thuật toán điều khiển cấu trúc liên kết mạng, ngoại trừ bản chất phân tán là không thể thiếu được và tổng chi phí thấp. Tuy nhiên, nhằm để thỏa mãn mục tiêu tối ưu hóa thì cần có báo trước ngắn gọn về việc kết nối.

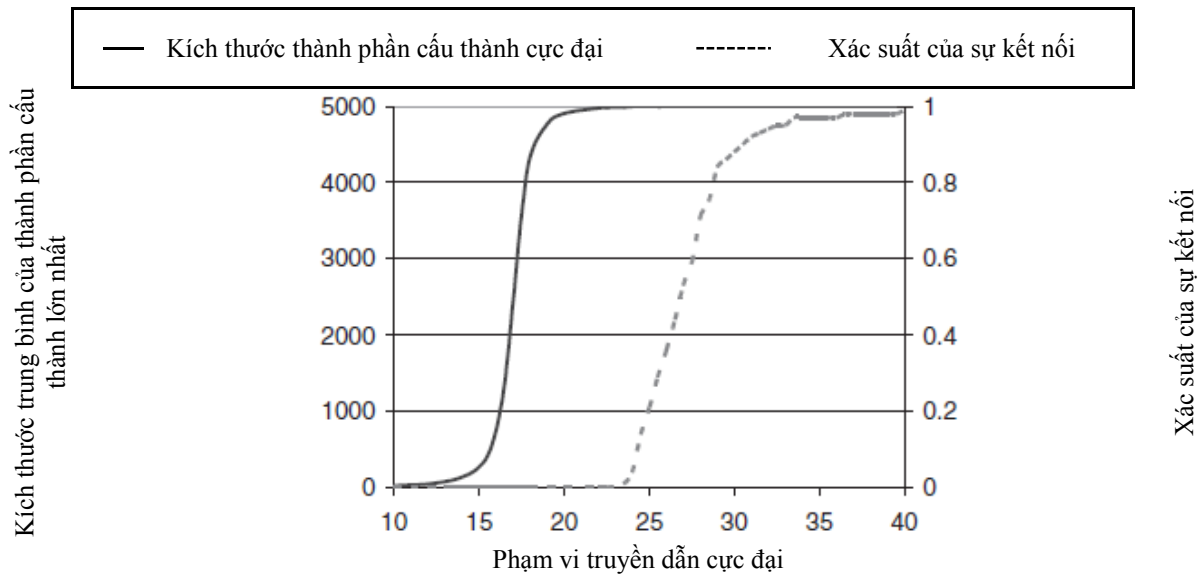
Một sự báo trước cho việc kết nối

Chúng ta xem xét một ví dụ đơn giản về điều khiển cấu trúc liên kết mạng công suất. Năm ngàn nút là giống nhau, được triển khai một cách ngẫu nhiên trên một diện tích 1000×1000 m. Phạm vi truyền dẫn của mỗi nút có thể được đặt bằng một bán kính chính xác là r m (nghĩa là tất cả các nút ở cách khoảng r m có thể truyền thông một cách trực tiếp, và không nút nào khác có thể làm được điều đó). Mô hình này được biết đến như là **mô hình sơ đồ đĩa (disk graph model)**; trường hợp đặc biệt là $r = 1$ được gọi là sơ đồ đĩa đơn vị. Đối với mạng ví dụ này và một phạm vi truyền dẫn đã cho, thì mạng này hoặc được kết nối hoặc không. Việc xác định kết nối cho 100 mạng khác nhau được tạo ra một cách ngẫu nhiên cung cấp một ước lượng xấp xỉ về xác suất của sự kết nối như một hàm của phạm vi truyền dẫn, và được trình bày bởi đường nét đứt ở hình 10.5.

Như mong đợi, xác suất của việc kết nối là 0 cho các phạm vi truyền dẫn nhỏ và tăng lên tương đối nhanh, cho đến khi các mức của nó tiến xa và dần tiến đến xác suất bằng 1 ở phạm vi truyền dẫn khoảng 30m.

Một thí nghiệm tương tự cho phép xem xét thêm một tham số đo. Đối với mỗi lần lặp, kích thước của thành phần được kết nối lớn nhất có thể được tính toán, và kích thước này được lấy trung bình trên 100 phép lặp, như trình bày ở hình 10.5, nó cũng là một hàm của phạm vi truyền dẫn. Rõ ràng là thậm chí đối với các phạm vi truyền dẫn khá nhỏ, gần như tất cả các nút đều được nối vào một thành phần đơn, thậm chí xác suất của việc kết nối thực tế là vẫn bằng 0 bởi vì có ba nút ở vị trí không thích hợp. Ví dụ: đối với phạm vi truyền dẫn là 25m, kích thước trung bình của thành phần lớn nhất là 4997 – nghĩa là, chỉ 3 trong số 5000 nút là được nối – trong đó xác suất của việc kết nối thực tế vẫn là 0. Hiển nhiên, đối với WSN

thì việc kết nối có thể là một tham số đo có liên quan, nhưng một giá trị lớn của kích thước thành phần cực đại sẽ là quan trọng hơn.



Hình 10.5 Mục tiêu tối ưu hóa điều khiển cấu trúc cấu trúc liên kết mạng: xác suất của mạng được kết nối so với kích thước trung bình của thành phần cấu thành lớn nhất.

Phân nổi bật của nghiên cứu là đi sâu vào nghiên cứu các đặc điểm kết nối, tầm quan trọng của kích thước của thành phần cấu thành chỉ được nhận ra một cách từ từ. Một khía cạnh quan trọng khác đó là **vùng phủ** - đảm bảo chắc chắn rằng tất cả các điểm trong mặt phẳng được bao phủ bởi một nút quan sát. Nếu vài nút mất tích đối với sự kết nối thì cũng là quan trọng đối với vùng phủ, và thực sự có thể không tránh được việc phải đầu tư năng lượng cần thiết để kết nối chúng.

Hình 10.5 cũng chỉ ra các ảnh hưởng khác. Kích thước trung bình của thành phần cấu thành lớn nhất và xác suất của việc có một mạng kết nối cũng không tăng lên một cách chậm rãi với phạm vi truyền dẫn cực đại (hoặc với mật độ của mạng). Hơn nữa, cả hai tham số đo đều tăng lên một cách nhanh chóng từ 0 tới các giá trị cực đại của chúng mỗi khi một ngưỡng quyết định nào đó đối với phạm vi truyền dẫn bị vượt quá. Ảnh hưởng này được biết đến đối với một số lượng lớn các khía cạnh của các sơ đồ (ngẫu nhiên) nói chung và được gọi là một **sự quá độ pha**. Việc tồn tại của các ngưỡng này là có thể chứng minh được đối với các sơ đồ ngẫu nhiên thuần túy và là hợp lý đối với các sơ đồ đĩa (đơn vị) khi được sử dụng để mô hình hóa các mạng không dây. Krishnamachari [19] thảo luận nhiều ví dụ khác nhau cho các ngưỡng này và khuyến nghị đặt các tham số hoạt động của một mạng chỉ lớn hơn ngưỡng quyết định một chút để nhận được sự hoạt động mong muốn mà không lãng phí các tài nguyên. Hiện tượng quá độ pha thường được phân loại bằng cách sử dụng các kỹ thuật lý thuyết lọc (percolation theory).

10.2 Điều khiển cấu trúc liên kết mạng trong các mạng phẳng – Điều khiển công suất

Việc điều khiển tập hợp các nút lân cận mà một nút nói chuyện với chúng là phương pháp cơ bản để điều khiển cấu trúc liên kết mạng. Trong phần này, một cấu trúc liên kết mạng phẳng được xem như nơi mà tất cả các nút đang hoạt động và có cùng các nhiệm vụ. Vấn đề

này có mối liên quan mật thiết với việc điều khiển công suất truyền dẫn của các nút và có thể thường được tìm thấy trong các tài liệu dưới tiêu đề này. Nói một cách ngắn gọn, điều khiển công suất ở đây cũng được sử dụng ngắn gọn như một thuật ngữ, mặc dù không chính xác ở một mức độ nào đó.

10.2.1 Một số các kết quả phức tạp

Vấn đề điều khiển công suất có nhiều biến thể khác nhau. Để mô tả một vấn đề điều khiển công suất, một bộ bốn (M, P, O, I) được sử dụng:

- $M \in \{\text{dir}, \text{undir}\}$ mô tả một mô hình đồ thị trực tiếp hoặc gián tiếp (định nghĩa cho ở trên là một đồ thị trực tiếp (V, E) , đồ thị gián tiếp tương ứng là $(V, E \cap E^{-1})$, nghĩa là, các đường truyền không đối xứng bị loại bỏ đi).
- P là đặc tính của đồ thị mà được đảm bảo trong đồ thị có cấu trúc liên kết mạng được điều khiển T .

Ví dụ cho các đặc tính bao gồm trong đó nhưng không bị hạn chế đối với:

- Kết nối mạnh

- Kết nối k nút (k -NC), nghĩa là có k nút trong đồ thị mà việc xóa các nút này làm dừng kết nối sơ đồ gián tiếp; sơ đồ vẫn được kết nối khi xóa bất cứ $k - 1$ nút nào.

- k cạnh được kết nối (k -EC) (tương tự cho các cạnh). Chú ý rằng kết nối k nút ngụ ý là kết nối k cạnh, nhưng không nhất thiết phải ngược lại.

- O là hàm đối tượng mà được tối thiểu hóa. Các ví dụ bao gồm việc tối thiểu hóa công suất cực đại được ấn định cho bất kỳ nút nào ($O = \max P$) hoặc tối thiểu hóa tổng số các giá trị công suất được ấn định cho tất cả các nút ($O = \text{total } P$).
- I là bất cứ thông tin thêm vào nào mà điều khiển cấu trúc liên kết mạng có thể sử dụng. Ví dụ nổi bật nhất là thông tin về các vị trí địa lý của các nút ($I = \text{pos}$).

Lloyd [20] cung cấp tóm tắt vài kết quả phức tạp cơ sở cho điều khiển cấu trúc liên kết mạng:

- $(\text{undir}, 1\text{-NC}, \text{total } P, -)$ là NP-cứng, có một thuật toán xấp xỉ với một đảm bảo hiệu suất bằng 2. (Một thuật toán xấp xỉ có một đảm bảo hiệu suất ρ nếu đối với tất cả các vấn đề ví dụ, giải pháp được xấp xỉ là nằm trong một hệ số ρ của của giải pháp tối ưu). Các phiên bản hình học của vấn đề ở NP-cứng trong 2 hoặc 3 chiều.
- $(\text{undir}, 1\text{-NC}, \max P, -)$ và $(\text{undir}, 2\text{-NC}, \max P, -)$ là có thể giải được trong thời đa thức.
- Bất cứ đặc tính *đơn tính* (*monotone*) P nào có thể kiểm tra được trong thời đa thức trong một đồ thị trực tiếp (gián tiếp), $(\text{undir}, \emptyset, \max P, -)$ and $(\text{dir}, \emptyset, \max P, -)$ có thể được giải trong thời đa thức. Nó không phải là trường hợp cần thiết cho tất cả \emptyset nếu số lượng các nút sử dụng công suất cực đại cũng phải được tối thiểu hóa [20].
- Không *đơn tính* nhưng các đặc tính P có thể kiểm tra một cách hiệu quả (ví dụ: sơ đồ là một cây) sao cho tối thiểu hóa $\max P$ là NP-hoàn thiện [20].
- Lloyd [20] cũng mô tả một phương pháp có đặc điểm chung để thiết kế các thuật toán xấp xỉ cho toàn bộ các vấn đề P , các vấn đề nói chung là NP-cứng.
- Có vài cấu trúc thời đa thức, phân bố cho các đồ thị con T của G với một hệ số dẫn là hằng số - các sơ đồ này được gọi là các sơ đồ bao trùm (**spanner graphs**)

(khoảng cách giữa hai nút bất kỳ trong T là gần như gấp khoảng cách của hai nút trong G một số lần không đổi) [24].

10.2.2 Các giới hạn của các tham số chủ chốt

Cho trước phạm vi của các vấn đề khác nhau, bằng cách nào chúng ta thiết kế các thuật toán cụ thể mà thu được một mục đích tối ưu hóa đã cho bằng cách điều khiển cấu trúc liên kết mạng của sơ đồ? Một ý tưởng đầu tiên là đối với tất cả các nút chỉ để truyền thông với k nút lân cận nhất của nó, trong đó số k (mong muốn) được xác định sao cho nó tối ưu hóa một mục tiêu đã cho, ví dụ: thông lượng hoặc sự kết nối. Tình trạng đơn giản nhất sẽ là nếu nó có thể xác định một giá trị của k mà không phụ thuộc vào sơ đồ thực tế được điều khiển. Một giá trị hằng số k như vậy có thể được gọi là “con số kỳ diệu”. Một số tài liệu đã cố gắng để chứng minh sự tồn tại của con số này, thực sự bắt đầu vào những năm 1970.

Các bài báo nổi tiếng nhất là của Kleinrock và Silvester [17]. Chúng đề cập đến một tập hợp các nút được phân bố đều trên một hình vuông và cố gắng để cực đại hóa tiến trình mong muốn mà một gói tin có thể làm hướng tới đích của nó, giả thiết rằng một giao thức ALOHA MAC kiểu chia khe và vài truyền dẫn gói xảy ra đồng thời. Chúng gợi ý rằng $k = 6$ sẽ cực đại hóa thực tế tiến trình trên mỗi bước nhảy. Kết quả này được chứng minh là đúng với $k \approx 8$, sử dụng cùng một đường lập luận cơ bản.

Trong các kết quả thú vị này, điều quan trọng là người ta không công bố vấn đề kết nối. Hơn nữa, mục đích tối ưu hóa ở đây là thông lượng. Nói cách khác, khi sử dụng “các con số kỳ diệu” này, sơ đồ kết quả không được đảm bảo là sẽ được kết nối. Trong thực tế, trường hợp này không phổ biến.

Khi nhìn vào vấn đề kết nối, có hai tùy chọn để tiếp cận vấn đề này. Một tùy chọn đề cập đến *phạm vi truyền dẫn* của một nút, và tùy chọn kia đề cập đến *số lượng của các nút lân cận*. Với các giả thiết nào đó, hai tùy chọn này là tương đương, nhưng dẫn đến các kiểu chứng minh và các kết quả khác nhau một chút.

Điều khiển phạm vi truyền dẫn

Tùy chọn đầu tiên, điều khiển phạm vi truyền dẫn, là hình học thuần túy và giả thiết một mô hình sơ đồ đĩa đơn vị và một phân bố đồng nhất của các nút trong một khu vực có diện tích A đã cho. Việc giả thiết mô hình này tương ứng với lý thuyết của các sơ đồ hình học ngẫu nhiên, các sơ đồ mà có thể được sử dụng ở đây để có được lợi ích lớn.

Đối với một mạng như vậy, người ta biết rằng xác suất của việc được kết nối của nó tiến

đến 0 nếu $r(V) \leq \sqrt{\frac{(1-\varepsilon)A \log V}{\pi V}}$ đối với bất cứ $\varepsilon > 0$. Nếu, mặt khác,

$r(V) \geq \sqrt{\frac{(1-\varepsilon)A \log V}{\pi V}}$ thì sơ đồ được nối với xác suất che phủ tiến tới 1 nếu và chỉ nếu $\gamma|V|$

$\rightarrow \infty$ như là một hàm của số lượng các nút trong mạng.

Một phương pháp tương tự dựa trên các sơ đồ hình học ngẫu nhiên được sử dụng để xác định một biểu thức cho xác suất của một sơ đồ (mật độ) có k kết nối (làm cho các kết quả này có tính khái quát hơn là kết quả trước đó), phụ thuộc vào phạm vi truyền dẫn r của nút và vào mật độ nút ρ (giả thiết phân bố đồng đều của nút trên một khu vực đã cho). Kết quả khóa

được sử dụng ở đây bởi Penrose [23], ông đã chứng minh rằng một sơ đồ với một số lượng lớn các nút là k được nối ngay khi công suất truyền dẫn trở nên đủ lớn để đảm bảo rằng địa vị nhỏ nhất trong sơ đồ là k ít nhất (với xác suất cao). Kết quả này được sử dụng để phát triển một công thức cho xác suất của địa vị nút cực tiểu trong một sơ đồ, dẫn đến kết quả chính như sau:

$$P(G \text{ là } k \text{ được kết nối}) \approx \left(1 - \sum_{l=0}^{k-1} \frac{(\rho \pi r^2)^l}{l!} e^{-\rho \pi r^2} \right) \quad (10.3)$$

Công thức này là đúng cho các vùng vô hạn nhưng không đúng cho các vùng hữu hạn như được sử dụng trong các mô phỏng và giải thích cách tính **ảnh hưởng cạnh (rìa)** này. Việc xử lý các ảnh hưởng cạnh và các công thức xấp xỉ thực tế làm cho nó ý nghĩa thực tiễn và có giá trị sử dụng trong các nghiên cứu dựa trên việc mô phỏng. Tuy nhiên, nó không rõ ràng là bằng cách nào các kết quả này phổ biến cho việc triển khai nút không đồng dạng và, nói riêng, cho các mạng thưa thớt khi lý thuyết của các sơ đồ hình học ngẫu nhiên không được áp dụng ở đây.

Một nghiên cứu về vấn đề của kết nối k cũng được công bố, trong đó chỉ ra rằng xác suất mà một mạng gồm n nút mà ít nhất $(k+1)$ nút được kết nối là ít nhất $e^{-\alpha}$ khi bán kính truyền dẫn r thỏa mãn $n\pi r^2 \geq \ln n + (2k-1) \ln(\ln n) - 2 \ln k + 2\alpha$ đối với $k > 0$ và n đủ lớn.

Cũng để đánh giá chỉ các mạng thưa thớt, Santi và Blough [26] phát biểu vấn đề một cách có hệ thống với chút ít khác biệt. Họ xem xét việc triển khai đồng dạng và ngẫu nhiên của một số lượng cố định (có thể là di động) của các nút n trong một vùng triển khai d -chiều $R = [0, l]^d$, cho đến nay việc không quan tâm đến trường hợp $d = 3$ cũng được bao gồm ở đây. Không có sự giới hạn nào lên mật độ nút $\rho = n/l^d$. Mục đích là để ấn định một phạm vi tối thiểu cho tất cả các nút sao cho sơ đồ kết quả được kết nối.

Đối với trường hợp một chiều $d = 1$, người ta chỉ ra rằng sơ đồ được nối với xác suất cao nếu $rn \geq 2l \ln l$ và rằng nó được kết nối với xác suất cao nếu $rn < l \ln l$. Đối với trường hợp $d = 2$ và $d = 3$, các kết quả sau được cho:

- Giả thiết rằng $r^d l = kl^d \ln l$ đối với hằng số $k > 0$, $r = r(l) \ll l$ and $n = n(l) \gg 1$ nào đó. Nếu $k > d \cdot 2^d d^{d/2}$ (và thêm vào đó $r = r(l) \gg 1$), thì sơ đồ truyền thông được nối với xác suất cao nhất.
- Giả thiết rằng $r = r(l) \ll l$ and $n = n(l) \gg 1$. Nếu $f^d \in O(l^d)$, thì sơ đồ truyền thông được nối với xác suất cao nhất.

Điều quan trọng là chỉ ra rằng các kết quả này, mà được lập thành công thức như là các ràng buộc trong phạm vi truyền dẫn, có thể chỉ được nhìn thấy là tốt như một ràng buộc trên số lượng các nút nhỏ nhất mà cần thiết để che phủ một khu vực đã cho khi có một phạm vi truyền dẫn cực đại mở rộng (ví dụ như phần cứng).

Santi và Blough [26] cũng làm cho phân tích của họ có hiệu lực bằng cách mô phỏng và cung cấp một số ví dụ dựa trên sự phụ thuộc của kích thước của phần tử cấu thành lớn nhất trong phạm vi truyền dẫn. Các kết quả này là phù hợp với sự kiện ở phía trước trong phần 10.1.2. Tuy nhiên, họ cũng xem xét phiên bản di động của vấn đề kết nối. Phạm vi truyền dẫn có thể bị giảm cơ bản nếu phải chịu các ngừng kết nối ngắn hạn.

Điều khiển số lượng các nút lân cận

Tùy chọn thứ hai không nhìn vào khu vực mà một phạm vi truyền dẫn của nút phải bao phủ, mà xem xét trực tiếp vào số lượng của các nút. Các kết quả trước đây cũng gợi ý rằng số lượng nút lân cận mong muốn của một nút nên tăng lên một cách logic. Chính xác hơn, chúng ta thấy như sau (giả thiết rằng các đường nối là đối xứng, nghĩa là một đường không đối xứng gốc được bổ sung bởi đường nối bị mất trong hướng ngược lại): (i) đối với các mạng được kết nối, số lượng của các nút lân cận của một nút cần tăng lên giống như $\theta(\log |V|)$, với các hằng số được giới hạn như sau đây. (ii) Đối với các nút lân cận nhỏ hơn $0,074 \log |V|$, mạng này không được kết nối một cách tiệm cận. (iii) Đối với các nút lân cận lớn hơn $5,1774 \log |V|$, mạng này được kết nối một cách tiệm cận. Các kết quả này vẫn tiếp tục được giữ khi một đường nối không đối xứng được giải phóng.

Các kết quả này cho thấy rằng không có các số kỳ diệu nào xác định số lượng của các nút lân cận cần thiết để nhận được một mạng được kết nối. Mức độ lớn (số lượng nút theo logarit) được yêu cầu cho kết nối ở một mức độ nào đó là thực sự bất tiện đối với hoạt động của mạng thực tế.

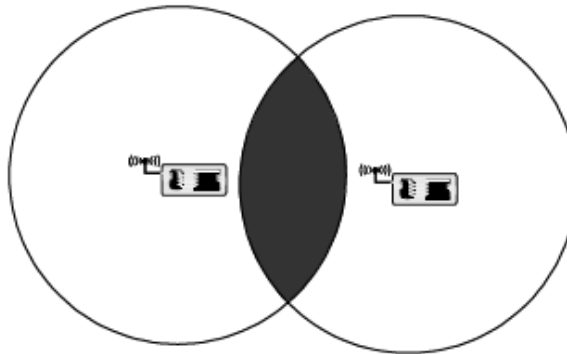
10.2.3 Vài cấu trúc và giao thức ví dụ

Một cấu trúc liên kết mạng có thể được thực hiện một cách có hiệu quả một cách cục bộ nếu sẵn có thông tin về các khoảng cách giữa các nút hoặc các vị trí liên quan của chúng. Vài cấu trúc đối với các sơ đồ gần gũi này tồn tại với các đặc điểm khác nhau – một số được mô tả ở đây. Công việc này dựa nhiều vào các kết quả của hình học sử dụng máy điện toán.

Các kết quả cụ thể đôi khi ở dưới dạng của một cấu trúc hình học trên một sơ đồ đã cho, sơ đồ mà thường là đáp ứng đối với một giao thức tập trung hóa. Đôi khi các giao thức phân tán cũng được cho trước.

Sơ đồ lân cận liên quan

Sơ đồ lân cận liên quan (RNG) T của một sơ đồ $G = (V, E)$ được định nghĩa như sau $T = (V, E')$ trong đó có một cạnh giữa các nút u và v nếu và chỉ nếu không có nút w nào khác (“một nhân chứng”) mà ở gần hơn với hoặc nút u hoặc nút v so với khoảng cách giữa hai nút u và v – một cách chính thức, $u, v \in V : (u, v) \in E'$ nếu $\exists w \in V : \max\{d(u, w), d(v, w)\} < d(u, v)$, trong đó $d(u, v)$ là khoảng cách Euclidean giữa hai nút. Theo một cách khác, cấu trúc RNG loại bỏ cạnh lớn nhất khỏi bất cứ tam giác nào (hình 10.6).



Hình 10.6 Cấu trúc của RNG: vùng tối phải không chứa các nút nào khác cho hai nút được kết nối.

RNG là dễ dàng xác định với một thuật toán cục bộ. Nó cũng cần thiết được kết nối nếu sơ đồ gốc G được kết nối. Tuy nhiên, tỷ số dẫn trong trường hợp tồi tệ nhất là $\Omega(|V|)$ (suy ra, các nút mà chỉ cách nhau vài bước nhảy trong sơ đồ gốc có thể trở thành cách nhau rất xa trong RNG), và năng lượng của nó trái lại là đa thức. Mức độ trung bình của nó là 2,6.

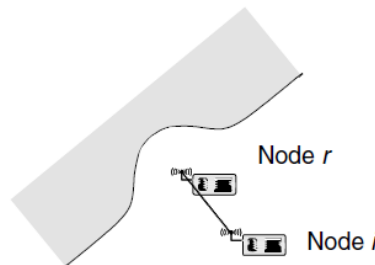
Cấu trúc dựa trên giải thuật cây bao trùm

Một cấu trúc khác là dựa trên các cây bao trùm cực tiểu cục bộ. Ý tưởng là mỗi nút sẽ thu thập thông tin về các nút lân cận của nó (tại công suất truyền dẫn cực đại) và sau đó xây dựng (ví dụ: sử dụng thuật toán của Prim) một cây bao trùm cực tiểu cho các nút này, với giá thành năng lượng được sử dụng như các trọng số của đường nối (các đường nối với cùng giá thành được phân biệt bằng cách thêm vào các nhận dạng nút như một biện pháp quyết định đường nối nào được chọn). Sau đó khóa chỉ dùng để duy trì các cạnh này trong cấu trúc liên kết mạng giảm xuống mà tương ứng với các nút lân cận trực tiếp trong cây bao trùm cực tiểu.

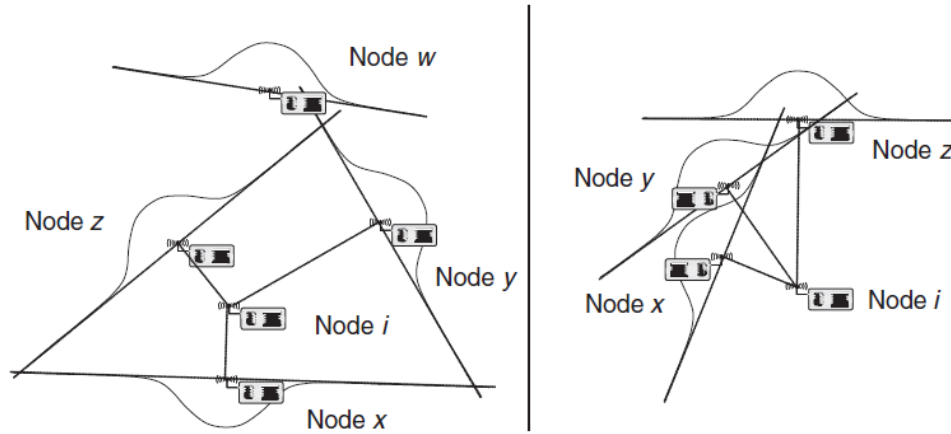
Cấu trúc này tạo ra các đặc tính thú vị. Nó bảo tồn kết nối của sơ đồ gốc, và địa vị cực đại của mỗi nút sẽ là sáu. Nó có thể giới hạn tới các đường nối hai hướng, và việc điều khiển công suất có thể được thêm vào một cách dễ dàng. Thêm vào đó, địa vị nút trung bình là nhỏ (gần với các đường bao lý thuyết). Cấu trúc này là khá triển vọng khi được so sánh với vài cấu trúc khác được thảo luận ở đây.

Các vùng tiếp sóng (Relay) và các phần đi kèm

Một phần thiết yếu của việc xây dựng một cấu trúc liên kết mạng quyết định các nút lân cận nào được sử dụng. Đối với các mạng không dây, việc quyết định này nên dựa trên (ít nhất đối với một mức độ lớn) việc tiêu thụ năng lượng mà là kết quả từ việc sử dụng một nút đã cho như một nút lân cận trực tiếp. Từ quan điểm này khái niệm về **một vùng tiếp sóng** lần đầu tiên được đề cập đến: Cho một nút i và một nút khác r , đối với các điểm nào trong mặt phẳng thì i sử dụng r như một nút tiếp sóng để giảm sự tiêu thụ công suất tổng, so với trường hợp truyền thông trực tiếp? Một cách chính thức, **vùng tiếp sóng** $R_{i \rightarrow r} = \{(x, y) | P_{i \rightarrow (x,y)} > (P_{i \rightarrow r} + P_{r \rightarrow (x,y)})\}$, trong đó $P_{a \rightarrow b}$ là công suất cực tiểu cần thiết để truyền thông một cách trực tiếp từ a tới b (nhận dạng các nút với vị trí của chúng trong kí hiệu). Hình dạng của vùng tiếp sóng là khác nhau, phụ thuộc vào việc mô hình tiêu thụ công suất được sử dụng; chú ý rằng bộ tiếp sóng bản thân nó cũng yêu cầu công suất, vùng tiếp sóng là “được uốn” xung quanh nút tiếp sóng như được vẽ trên hình 10.7 và tiếp cận một cách tiệm cận một đường vuông góc với đường ngắn nhất giữa các nút i và r .



Hình 10.7 Minh họa vùng tiếp sóng của nút i với nút r (vùng tối chỉ thị các điểm nơi mà tiếp sóng có hiệu suất cao hơn thay vì truyền thông trực tiếp)



Hình 10.8 Hai kịch bản làm cách nào xây dựng các nút lân cận cho nút i ngoài các vùng tiếp sóng với các nút khác.

Trên cơ sở của việc định nghĩa công suất đã biết này của tập hợp các điểm mà một nút tiếp sóng được sử dụng cho các điểm này, bước tiếp theo là định nghĩa các nút mà một nút đã cho i nên truyền thông một cách trực tiếp với các nút này. Khu vực nơi mà các nút này được định vị được gọi là **vùng biểu diễn**. Trực giác đầu tiên là phân cắt một cách đơn giản các phần bù của tất cả các vùng tiếp sóng và chỉ xem xét các nút nằm trong sự giao nhau này như là các nút lân cận trực tiếp. Đối với một nút ở bên ngoài sự giao nhau này, có ít nhất một nút khác mà có thể cung cấp một tuyến đường có công suất thấp hơn là truyền thông trực tiếp. Trực giác này đúng cho vài trường hợp, như được thí dụ bởi nửa bên trái của hình 10.8.

Ví dụ nửa bên phải của hình 10.8 làm nổi bật một trường hợp ở đó cả hai nút x và z được duy trì như các nút lân cận, bởi vì nút y , mà sẽ “chi phối” nút z , lần lượt bị chi phối bởi nút x và do đó không được xem xét. Một cách chính thức, đặc điểm này của tập hợp các nút lân cận được biểu diễn dưới dạng một phương trình cố định. Trên đỉnh của sơ đồ được định nghĩa này, một thuật toán định tuyến được sử dụng để tìm kiếm các đường có năng lượng nhỏ nhất trong mạng.

10.3 Các mạng phân tầng – xương sống

10.3.1 Động cơ và định nghĩa

Phần 10.2 phía trước đã đề cập đến việc điều khiển phạm vi truyền dẫn và suy ra *số lượng các nút lân cận* của một nút. Phần này sẽ xem xét các phương pháp để lựa chọn *các nút cụ thể nào* sẽ là các nút lân cận của một nút đã cho, và các nút nào khác (và các đường nối tới các nút này) nên bỏ qua. Trong khi điều này cũng hoàn toàn ảnh hưởng đến số lượng của các nút lân cận tích cực, một số trong số các nút lân cận này có thể ở xa, và một số khác thì ở gần.

Thông thường (tuy nhiên không nhất thiết) thì việc lựa chọn này của các nút lân cận/các đường nối ngụ ý một dạng phân tầng giữa các nút nào đó. Trong phần này, một số nút được lựa chọn từ một “xương sống ảo” hoặc, một cách chính thức, một tập hợp chi phối - một tập hợp các nút $D \subset V$ là một tập hợp chi phối nếu tất cả các nút trong V đều hoặc ở trong bản thân D hoặc là các nút lân cận một bước nhảy của nút $d \in D$ ($\forall v \in V : v \in D \vee \exists d \in D : (v, d) \in E$) nào đó. Hình 10.3 cho chúng ta một hình dung về khái niệm này.

Có một tập chi phối như vậy sẽ đơn giản hóa việc định tuyến, ví dụ: bằng cách cho phép hạn chế giao thức định tuyến thực tế chỉ đối với các nút xương sống; tất cả các nút “chi phối” có thể chuyển tiếp một cách đơn giản một gói tin không cục bộ tới (một trong) các nút xương sống liền kề của chúng, các nút mà sau đó sẽ lãnh trách nhiệm chuyển tiếp gói tin hướng tới đích của nó.

Sẽ là có ích khi một tập hợp chi phối như vậy là “nhỏ” hoặc thậm chí cực tiểu – định nghĩa trên thừa nhận $V = U$ như một giải pháp, giải pháp này không cung cấp bất cứ ưu điểm nào một cách rõ ràng. Một tham số đo điển hình dùng để đo kích thước và/hoặc sự tối thiểu của một tập hợp chi phối U là số lượng các nút của nó. Nó được gọi là vấn đề *Tập hợp chi phối nhỏ nhất* (MDS). Hơn nữa, một cách trực quan một nút xương sống nên được nối bằng cách sử dụng các nút chỉ trong bản thân nó; nó không cần thiết phải nối với nút khác, các nút không xương sống để định tuyến một gói tin từ một vị trí tới vị trí khác.

Suy ra, vấn đề hiện nay là: nghĩ ra một thuật toán phân bố ưu tiên có khả năng xác định một Tập chi phối được nối nhỏ nhất (MCDS). Thêm vào đó, mỗi nút nên biết liệu nó có ở trong tập hợp này hay không và nếu không ở trong tập hợp này thì nút nào trong các nút lân cận của nó là nằm trong tập hợp chi phối.

10.3.2 Một kết quả cứng

Vấn đề MDS là NP- cứng. Nó thậm chí là một vấn đề khó để xấp xỉ nói chung các sơ đồ khi nó thậm chí không thể xấp xỉ trong $c \log |V|$ đối với $c > 0$ nào đó (nghĩa là, không có thuật toán thời gian đa thức nào mà luôn tìm thấy một tập hợp chi phối, tập hợp mà có tối đa $c \log |V|$ nút nhiều hơn so với một tập hợp chi phối nhỏ nhất có thể có). Nó có thể được xấp xỉ trong $1 + \log |V|$. Đối với trường hợp các sơ đồ đĩa đơn vị, người ta có thể tìm thấy một Kiểu xấp xỉ thời gian đa thức (PTAS – Polynomial Time Approximation Scheme).

Khi cũng yêu cầu sự kết nối trong việc giải quyết vấn đề MCDS (mà cũng là NP- cứng), người ta có thể xấp xỉ nó trong $\ln \Delta + 3$, trong đó Δ là mức độ cực đại của sơ đồ gốc. Sự quan sát này khuyến khích việc thiết kế các thuật toán thực tế (và cũng là một sự khích lệ để giải quyết MCDS trên đỉnh của một cấu trúc liên kết mạng có điều khiển công suất, cấu trúc mà hạn chế địa vị của các nút).

Thêm vào đó còn có một kết quả quan trọng đó là bất kỳ thuật toán nào mà tính toán một giải pháp tập hợp chi phối được kết nối quan trọng đều phải gửi ít nhất $\Omega(m \log n)$ bản tin, trong đó kích thước bản tin là theo bậc của số lượng các bit cần thiết để biểu diễn các nhận dạng duy nhất cho tất cả các nút $O(\log |V|)$.

10.3.3 Một số ý tưởng từ các thuật toán tập trung

Trong khi các thuật toán tập trung thường không thể được áp dụng một cách trực tiếp cho các mạng WSN, chúng có thể cung cấp một số ý tưởng về việc bằng cách nào để thiết kế các thuật toán phân tán cho một vấn đề đã cho. Phần này mô tả hai ví dụ tập trung.

Phát triển một cây

Một cách tiếp cận đơn giản

Một ý tưởng của thuật toán đầu tiên là để xây dựng một tập hợp chi phối như một cây bao trùm, thêm các nút và các cạnh một cách lặp đi lặp lại vào cây này cho đến khi tất cả các nút đều được bao trùm. Thuật toán này được lập thành công thức sử dụng “các màu” của các

nút: các nút màu trắng là vẫn chưa được xử lý, các nút màu đen thuộc về tập hợp chi phối này, và các nút màu xám là các nút chi phối.

Thuật toán này được trình bày trong mã giả trong Danh sách 10.1. Nó dùng một sơ đồ $G = (V, E)$ và tạo ra một tập hợp các cạnh T thuộc về tập hợp chi phối hình cây và tạo màu cho các nút. Hình 10.9 minh họa việc chạy thuật toán này.

Thực tế, sơ đồ kết quả cho thấy một cây không bao giờ có một cạnh giữa hai nút màu xám hoặc giữa một nút màu xám và hai nút màu đen; các nút màu đen cũng tạo thành một cây.

Danh sách 10.1: Một thuật toán tập hợp chi phối đơn giản dựa trên việc phát triển một cây

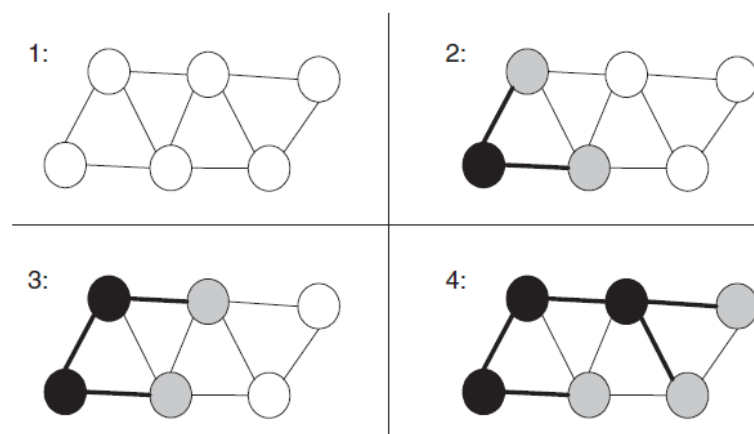
```

initialize all nodes ' color to white
pick an arbitrary node and color it gray

while (there are white nodes) {
    pick a gray node v that has white neighbors
    color the gray node v black
    foreach white neighbor u of v {
        color u gray
        add (v,u) to tree T
    }
}

```

Tập hợp của các nút màu đen (các nút không có lá của cây bao trùm) cũng là một tập hợp chi phối, khi mỗi nút màu xám có một nút màu đen như một nút lân cận và không có các nút màu trắng ở đó. Tuy nhiên, kích thước của tập hợp chi phối này không cần là quá nhỏ hay thậm chí không cần là nhỏ nhất. Trong ví dụ của hình 10.9, hai nút ở phía dưới cùng bên phải cũng tạo thành một tập hợp chi phối (nhỏ nhất). Rõ ràng là, chọn lấy nút màu xám sắp được chuyển thành một nút màu đen là bước chủ chốt trong thuật toán này.



Hình 10.9 Minh họa một thuật toán tập hợp chi phối đơn giản dựa trên việc phát triển một cây (các đường nét đậm chỉ các cạnh trong cây, các nút màu đen tạo thành tập hợp chi phối).

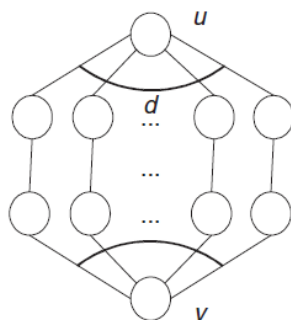
Chọn lựa một cách thận trọng các nút màu xám

Một lựa chọn khám phá hấp dẫn bằng trực giác cho nút màu xám tiếp theo đó là nút mà sẽ biến các nút trắng nhất thành màu xám. Con số này có thể được xem như “lợi ích” của việc chọn một nút màu xám cụ thể.

Thực tế, các nút khám phá tham vọng tìm kiếm giải pháp tối ưu như ví dụ trong hình 10.9. Tuy nhiên, có các sơ đồ đơn giản mà lựa chọn khám phá này là thất bại; hình 10.10 trình bày một sơ đồ như vậy. Ở đó, thuật toán sẽ bắt đầu tại nút u , thêm vào bất cứ một nút nào trong số các nút được nối một cách trực tiếp với u , sau đó phải cắt đứt các mối liên hệ giữa các nút ngang hàng của nút u và lựa chọn một nút được nối tới nút v . Nếu nút u có d nút lân cận, thì trường hợp xấu nhất cho một thuật toán dựa trên lợi ích tham vọng là một kích thước tập con chi phối $O(d)$, trong đó kích thước tối ưu là 4 (u , v , và hai nút từ bất cứ đường thẳng đứng nào – các nút u và v sẽ là đủ nếu không yêu cầu sự kết nối của tập hợp chi phối).

Lý do cho hiệu suất không tối ưu thực tế là thuật toán này không thể phân biệt được giữa việc lựa chọn một nút màu xám từ hàng thứ nhất hay hàng thứ hai – mỗi hoạt động làm giảm số lượng của các nút trắng đi một đơn vị. Hoạt động thiên cận này gây trở ngại cho thuật toán từ việc nhận ra rằng mỗi khi một nút từ hàng thứ hai được lựa chọn, nút bên dưới v là nút màu xám và có thể biến thành nút màu đen, tạo ra tập hợp chi phối được yêu cầu khi nút v có thể che phủ tất cả các nút trắng còn lại trong hàng thứ hai.

Sự thiên cận này có thể được khắc phục bằng cách cho phép thuật toán này nhìn về phía trước một bước. Thêm vào việc xem xét lợi ích của việc biến các nút màu xám riêng lẻ thành các nút màu đen, thuật toán này cũng có thể xem xét các cặp gồm một nút màu xám g và một nút màu trắng w liền kề, nghiên cứu về việc điều gì sẽ xảy ra nếu nút w này cũng bị biến thành nút màu đen trong bước tiếp theo. Đối với mỗi sự kết hợp của các nút g và w , lợi ích của hoạt động này có thể được tính toán. Sau đó phương pháp khám phá sẽ là chọn nút màu xám riêng lẻ hoặc cặp gồm một nút màu xám và một nút màu trắng mà tạo ra lợi ích cao nhất.

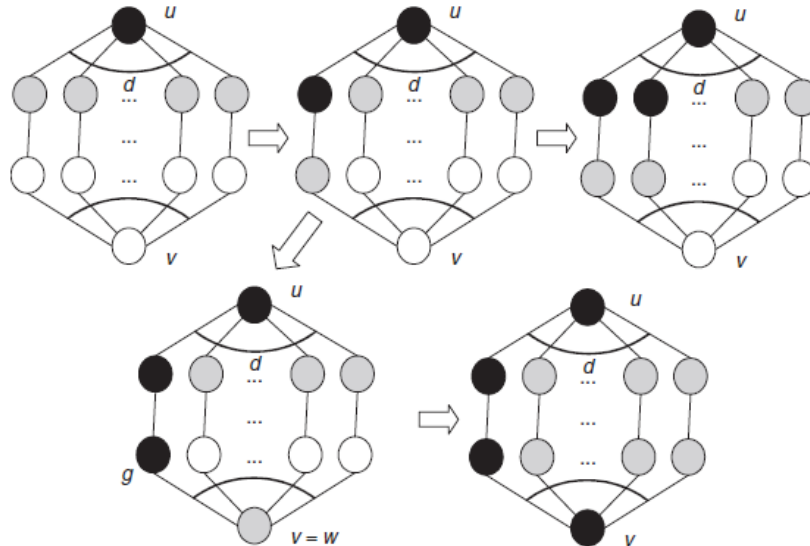


Hình 10.10 Sơ đồ ví dụ ở đó một lựa chọn khám phá tham vọng dựa trên lợi ích một bước bị thất bại

Ý tưởng này được minh họa ở hình 10.11. Phần phía trên chỉ ra rằng việc khám phá dựa trên lợi ích đơn giản, không có khả năng biến các nút màu xám ở hàng phía trên thành các nút màu đen nút này tiếp sau nút khác mà không tạo nên sự tiến triển thiết yếu – chọn lấy bất cứ nút màu xám riêng lẻ nào sẽ đem đến một lợi ích bằng 1 mà không thiên vị một nút trong hàng thứ hai. Ở phần phía dưới, việc nhìn về phía trước của các nút đánh dấu g và w (sau bước đầu tiên của thuật toán này) đem đến một lợi ích bằng $1 + (d - 1)$, nút màu trắng biến thành nút màu xám, rõ ràng là thiên vị hoạt động này thông qua việc lựa chọn các nút màu xám riêng lẻ.

Đây là một ví dụ cụ thể chỉ ra rằng việc hướng dẫn khám phá bằng cách nhìn về phía trước như vậy là có triển vọng. Nó khá là mạnh: nó tạo ra các tập hợp được kết nối chi phối gần như là lớn hơn với một hệ số $2(1 + H(\Delta))$ so với một tập hợp tối ưu ($H(\cdot)$ là hàm điều hòa

$H(k) = \sum_{i=1}^k \frac{1}{i} \leq \ln k + 1$ và Δ là địa vị cực đại của sơ đồ). Hơn nữa, bằng thực nghiệm người ta chứng minh rằng thuật toán này làm việc tốt trong thực tiễn.



Hình 10.11 Cấu trúc tập hợp chi phối tham vọng với và không với việc nhìn về phía trước.

Kết nối các thành phần cấu thành tách biệt

Trong đề cập phía trước, tập hợp các nút mà sẽ rất cuộc tạo nên tập hợp chi phối là luôn được kết nối. Một ý tưởng thay thế đó là: đầu tiên xây dựng một tập hợp chi phối không nhất thiết phải được kết nối và sau đó, trong pha thứ hai, kết nối một cách rõ ràng các nút trong tập hợp này.

Tìm kiếm một tập hợp chi phối không được kết nối

Trong một thuật toán tập trung, việc tìm kiếm một tập hợp chi phối (không được kết nối) nào đó là khá dễ dàng. Sử dụng các mã màu trắng/xám/đen ở trên, chọn lấy một nút màu xám hoặc màu trắng, tô màu cho nó thành màu đen và tô màu tất cả các nút lân cận màu trắng của nó thành màu xám. Thực hiện tiếp tục như vậy cho đến khi không có nút màu trắng nào còn lại. Giống như trong thuật toán phía trước, cần cẩn thận trong việc lựa chọn nút nào để biến thành màu đen để thu được một tập hợp chi phối nhỏ.

Một sự hướng dẫn khám phá đơn giản là lựa chọn nút mà biến phần lớn các nút màu trắng thành màu xám. Cần nhớ rằng việc cần thiết kết nối các nút màu đen vào một tập hợp được nối cũng có ý nghĩa trực giác đối với lựa chọn các nút màu xám, (các nút màu xám này nằm giữa hai nút màu đen) bởi vì có một lựa chọn tốt là các nút màu xám này có thể dù sao chẳng nữa cũng phải được thêm vào tập hợp chi phối. Hình 10.12 minh họa cho một ví dụ trong đó việc đề cập này phá vỡ mối quan hệ giữa hai nút A và B, kết quả là B được thêm vào tập hợp chi phối này.

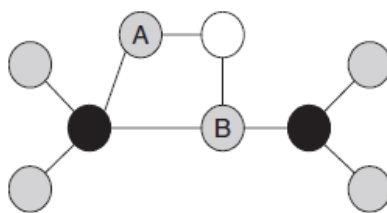
Một cách chính thức, việc hướng dẫn khám phá được Guha và Khuller [14] đề xuất do đó sẽ lựa chọn rằng nút màu trắng hay nút màu xám sẽ như là nút màu đen kế tiếp mà giảm phần lớn số lượng các nút màu trắng và số lượng của các thành phần cấu thành được kết nối của các nút màu đen. Khi không có nút nào còn lại để mà giảm số lượng này, cũng không có nút màu trắng nào còn lại thì pha đầu tiên có thể kết thúc.

Đảm bảo sự kết nối – xây dựng một cây Steiner

Mỗi khi tìm thấy một tập hợp chi phối không được kết nối như vậy, thì nó phải được kết nối, nghĩa là nhiều nút màu xám hơn phải được biến thành màu đen để đảm bảo rằng hai nút màu đen có thể tiến tới lẫn nhau với các nút màu đen khác chỉ như các trung gian. Điều này cũng khá là đơn giản, bởi vì tối đa hai nút màu xám có thể tách hai thành phần cấu thành màu đen “gần kề”. Việc kết nối đệ quy hai thành phần cấu thành màu đen với một hoặc hai nút màu xám ở giữa rốt cuộc sẽ dẫn đến một thành phần cấu thành được kết nối đơn lẻ của các nút màu đen, giải quyết được vấn đề này.

Có một nhận xét bên lề rằng vấn đề này cũng có thể được xem như việc tìm kiếm một **cây Steiner**: tìm kiếm một cây bao trùm nhỏ nhất mà chứa tất cả các nút của một tập hợp các nút được định nghĩa trước, thêm vào các nút khác khi được yêu cầu. Đây là vấn đề NP-hoàn thiện khác, nhưng có một hệ số xấp xỉ không đổi.

Nói chung, các tập hợp chi phối được xác định bởi việc khám phá này lớn hơn các tập hợp có thể tối ưu tối đa là $\ln \Delta + 3$.



Hình 10.12 Tính đến sự cần thiết để kết nối hai nút màu đen phá vỡ mối quan hệ giữa hai nút A và B.

10.3.2 Một số xấp xỉ phân tán

Việc phát triển một cây có tính phân tán

Việc khám phá sự phát triển của một cây [14] có thể được thực hiện một cách dễ hiểu theo kiểu phân tán như được Das và Bharghavan [9] mô tả. Về cơ bản, tất cả các nút màu xám thăm dò các nút lân cận hai bước nhảy, xác định lợi ích lớn nhất mà mỗi nút có thể nhận được. Sau đó, lợi ích lớn nhất được lựa chọn một cách phân tán (trong thực tế, một sự tuyển cử lãnh đạo xảy ra dọc theo xương sống đang có) và nút đen tiếp theo được xác định.

Tập hợp chi phối C được xác định bởi thuật toán này là lớn hơn tối đa $2H(\Delta)$ so với tập hợp tối ưu; nó chiếm $O(|C|(\Delta + |C|))$ thời gian và $O(n|C|)$ bản tin để tính toán.

Kết nối một tập hợp chi phối

Das và Bharghavan [9] cũng đề cập đến bằng cách nào thích nghi thuật toán tập trung khác được mô tả trong phần 10.3.3, xác định một Tập hợp chi phối nhỏ và sau đó kết nối nó ở một bước tách biệt.

Giả thiết địa vị của sơ đồ được giới hạn bởi hằng số c nào đó, $\Delta \leq c$. Hãy để các nút quảng bá địa vị của nó tới tất cả các nút lân cận của nó. Sử dụng thông tin địa vị này từ tất cả các nút, tất cả các nút đánh dấu nút lân cận với địa vị cao nhất như nút chi phối của nó (hoặc bản thân nó). Tập hợp thu được là tập hợp chi phối (hiển nhiên, bởi vì mỗi nút lựa chọn một nút chi phối), nhưng không cần thiết phải được kết nối; một cây Steiner – giống thuật toán kết nối quan tâm đến việc đảm bảo kết nối trong pha thứ hai.

Tập hợp chi phối thu được có một tỷ số hiệu suất $\Delta + 1$ trong thời gian $O(n\Delta)$. Bởi vì $\Delta \leq c$, nó vẫn là một tỷ số hiệu suất hằng số. Người ta nghĩ rằng, việc đảm bảo một địa vị cực đại đối với một triển khai ngẫu nhiên của các nút cảm biến là quan trọng. Nói chung, các thuật toán được Das và Bharghavan [9] trình bày có một hệ số xấp xỉ logarit.

Đánh dấu các nút có các nút lân cận không được nối

Trong khi cả hai thuật toán được thảo luận đến bây giờ cố gắng ngay từ ban đầu để hạn chế kích thước của tập hợp chi phối, ý tưởng khác là đầu tiên tìm kiếm tập hợp chi phối được nối, khá lớn nào đó mà có thể được cấu trúc một cách dễ dàng và sau đó giảm kích thước của nó đi. Nếu một tập hợp được cấu trúc như vậy có một số các đặc điểm thuận lợi khác, thì thậm chí một tập hợp chi phối không phải là nhỏ nhất cũng có thể được chấp nhận.

Một cấu trúc tập hợp chi phối được nối không tối ưu

Một quan sát đơn giản [12] cho phép một cách nhanh chóng, với tổng chi phí bản tin thấp, xây dựng một tập hợp chi phối được kết nối. Một lần nữa giả thiết rằng tất cả các nút ban đầu là không được đánh dấu (có màu trắng). Đánh dấu bất cứ nút nào (làm cho nó trở thành một thành viên của tập hợp chi phối) nếu nó có hai nút lân cận không được kết nối trực tiếp. Quyết định này có thể được thực hiện một cách cục bộ sau khi các nút trao đổi các tập hợp nút lân cận của chúng – đơn giản là bất cứ nút nào cũng phải kiểm tra liệu mỗi trong số các nút lân cận của nó có nằm trong tất cả các danh sách nút lân cận nhận được. Cấu trúc đơn giản này đưa đến một số đặc điểm thú vị:

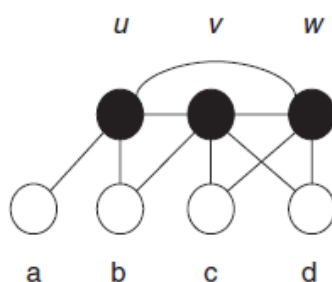
- Nếu sơ đồ gốc được kết nối nhưng không phải toàn bộ sơ đồ được kết nối, tập hợp kết quả của các nút được đánh dấu là một tập hợp chi phối. Điều này có thể được chứng minh bằng việc xem xét vài trường hợp. Trường hợp chủ chốt là một nút không được đánh dấu v với chỉ các nút lân cận không được đánh dấu và ít nhất một nút được kết nối gián tiếp u . Nhận thấy rằng nút đầu tiên trên đường kết nối ngắn nhất giữa v và u sẽ phải được đánh dấu bởi cấu trúc cho thấy sự đối lập.
- Tập hợp kết quả của các nút đánh dấu được kết nối. Chứng minh cũng bằng sự đối lập, sử dụng một đường ngắn nhất giữa hai nút được đánh dấu từ hai thành phần cấu thành được kết nối khác nhau của tập hợp chi phối. Nó cho thấy rằng hoặc là tất cả các nút dọc theo đường này phải được đánh dấu hoặc nó không là đường ngắn nhất, đối lập với giả thiết rằng hai thành phần cấu thành không được kết nối.
- Đường ngắn nhất giữa bất cứ hai nút bất kỳ không bao gồm các nút không được đánh dấu. Chứng minh một lần nữa được thực hiện bởi sự đối lập về đặc điểm của các nút trên một đường ngắn nhất. Đặc điểm này là không được quan tâm bởi các cấu trúc trước đây và làm cho tập hợp chi phối cụ thể này phải chịu trách nhiệm nhiều hơn để sử dụng như một xương sống định tuyến.

- Tập hợp chi phối không là tối thiểu; thực tế, nó có thể không quan trọng (bao quanh tất cả các nút của sơ đồ gốc). Nó được thực hiện tốt trong các sơ đồ thực tế. Hình 10.13 chỉ ra một sơ đồ ví dụ ở đó tập hợp kết quả của các nút được đánh dấu không phải là nhỏ nhất (cả u , v và u , w là đủ).

Hai khám phá lược bớt

Wu và Li [12] đề xuất hai khám phá để giảm kích thước của tập hợp chi phối trong các tình huống như vậy. Khám phá đầu tiên là không đánh dấu bất cứ nút v nào sao cho cả bản thân nút v và nút lân cận của nó đều được bao gồm trong vùng lân cận của nút được đánh dấu u khác nào đó; ngoài ra số nhận dạng duy nhất của v cũng phải nhỏ hơn số nhận dạng duy nhất của u . Khám phá thứ hai không đánh dấu một nút v nếu vùng lân cận của nó nằm trong các vùng lân cận của hai nút lân cận được đánh dấu u và w và v phải có số nhận dạng nhỏ nhất. Hình 10.15 cũng chỉ ra rằng việc phân bố của các số nhận dạng có tác động đáng kể lên các nút mà có thể loại bỏ.

So sánh với các thuật toán của Das và Bharghavan [192], thì thuật toán ở đây chỉ yêu cầu $O(\Delta^2)$ thời gian để trao đổi các tập hợp lân cận và thời gian không đổi cho việc giảm pha. Tuy nhiên, các kết quả mô phỏng chỉ ra rằng tập hợp chi phối (sau các khám phá lược bớt) cũng có xu hướng trở nên nhỏ hơn so với tập hợp được tạo ra bởi các thuật toán trước đó.



Hình 10.13 Sơ đồ ví dụ cho việc đánh dấu các nút với các nút lân cận không được kết nối.

Tổ chức tự phân tầng dựa trên vai trò

Tất cả các phương pháp tập hợp chi phối được thảo luận cho đến giờ hoàn toàn là sơ đồ mang tính lý thuyết; chúng không tính đến các nhiệm vụ và vai trò cụ thể mà các nút trong sơ đồ này phải thực hiện. Kochhal [18] đề xuất một phương pháp xây dựng một tập hợp chi phối mà xem xét một cách rõ ràng các thông tin trên.

Nói riêng thì mức độ của dung sai lỗi trong việc cảm biến một khu vực đã cho, khu vực mà được ngụ ý bởi việc triển khai dự các cảm biến, kết quả là nhiều cảm biến khảo sát một điểm đã cho, được sử dụng để quyết định các vai trò mà một cảm biến đã cho thực hiện. Mức độ của dung sai lỗi được biểu diễn như một thuộc tính của cảm biến đã cho bất kỳ, giá trị Mức độ cảm biến tích lũy - Cumulative Sensing Degree (CSD). Một cách trực giác, một giá trị CSD cao ngụ ý rằng các thay đổi của một sự kiện được phát hiện giữa các cảm biến lân cận của nó là cao.

Đặc điểm này của các giá trị CSD sau đó được sử dụng để hỗ trợ trong việc xây dựng một Tập hợp chi phối được kết nối (CDS). Cấu trúc ban đầu đánh dấu các nút với các nút lân cận không được nối, như được mô tả trong phần trước. Tuy nhiên, các khám phá đơn giản sử

dùng các giá trị CSD để lựa chọn các nút sẽ bị loại bỏ khỏi tập hợp chi phối này (giữa các giá trị khác giống như mức năng lượng hoặc mức độ của nút). Trên cơ sở của tập hợp chi phối được kết nối thu được, cuối cùng thì các điều phối viên cảm biến được lựa chọn.

Giải pháp này là hấp dẫn ở chỗ nó kết hợp một cách tiếp cận lý thuyết của sơ đồ đơn giản với kiến thức về các nhiệm vụ cảm biến thực tế mà các nút riêng lẻ phải thực hiện.

10.4 Mạng phân tầng – phân nhóm (clustering)

10.4.1 Định nghĩa các phân nhóm

Phần 10.3 phía trước được giới thiệu một kiến trúc phân tầng trong một mạng bằng cách thiết kế vài nút khi chúng thuộc vào một xương sống, một tập hợp chi phối. Ý kiến khác đối với một kiến trúc phân tầng sẽ là đánh dấu một cách cục bộ một số nút khi có một vai trò đặc biệt, ví dụ: điều khiển cấu trúc liên kết mạng các nút lân cận. Trong ý kiến này, các nhóm cục bộ hoặc **các phân nhóm** của các nút có thể được hình thành; “các bộ điều khiển” của các nhóm này thường được xem như **các chỉ huy phân nhóm**. Việc phân nhóm này có các ưu điểm mong muốn tương tự với các ưu điểm của một xương sống, nhưng nhấn mạnh thêm vào tùy chọn tải nguyên cục bộ (ví dụ, trong các giao thức MAC), bảo vệ các lớp động cao hơn trong mạng (làm cho các bảng định tuyến ổn định hơn bởi vì tất cả các lưu lượng được định tuyến qua các chỉ huy phân nhóm), và làm cho các giao thức lớp cao hơn có thể thay đổi tỷ lệ linh hoạt hơn (bởi vì kích thước và độ phức tạp của mạng khi nhìn bởi các lớp cao hơn là giảm đi bởi việc phân nhóm). Thêm vào đó, các chỉ huy phân nhóm là các vị trí tự nhiên để tập hợp lại và nén lưu lượng hội tụ từ nhiều cảm biến tới một trạm đơn lẻ.

Một cách chính thức, cho một sơ đồ $G = (V, E)$, việc phân nhóm đơn giản là nhận dạng một tập hợp các tập con của các nút $V_i, i = 1, \dots, n$ sao cho $\cup_{i=1, \dots, n} V_i = V$. Số lượng các câu hỏi về các đặc điểm chi tiết được yêu cầu từ các tập hợp này phân biệt các phương pháp phân nhóm khác nhau:

Có các chỉ huy phân nhóm? Việc chia tách V thành nhiều phân nhóm không ủy thác bất cứ điều gì về cấu trúc bên trong của một phân nhóm; theo quy tắc thì tất cả các nút có thể bằng nhau. Tuy nhiên, điển hình cho mỗi tập hợp V_i có một nút duy nhất c_i , chỉ huy phân nhóm, để biểu diễn tập hợp này và có thể thực hiện các nhiệm vụ khác.

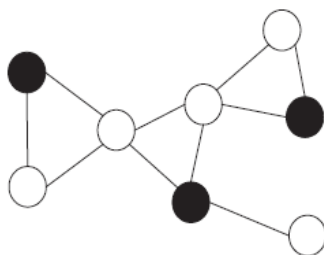
Có thể các chỉ huy phân nhóm là các nút lân cận không? Một lần nữa theo quy tắc thì nó hoàn toàn được chấp nhận đối với hai chỉ huy phân nhóm (của hai phân nhóm khác nhau) là các nút lân cận trực tiếp. Tuy nhiên, người ta thường mong muốn có các chỉ huy phân nhóm được tách biệt. Một cách chính thức, các chỉ huy phân nhóm nên tạo thành một **tập hợp độc lập**: một tập con $C \subset V$ sao cho không có hai nút nào trong C được tham gia bởi một cạnh trong $E - \forall c_1, c_2 \in C : (c_1, c_2) \notin E$.

Việc tìm một tập hợp tùy chọn như thế này là ít quan trọng; trường hợp đáng quan tâm là **các tập hợp độc lập cực đại**, mà chứa càng nhiều nút không vi phạm đến đặc tính độc lập càng tốt, kết quả là càng nhiều phân nhóm xung quanh các chỉ huy phân nhóm càng tốt. Hình 10.14 chỉ ra một sơ đồ ví dụ với một tập hợp độc lập cực đại; các ví dụ khác là có thể (dễ dàng tìm thấy) khi bản thân tập hợp độc lập cực đại, nói chung là không duy nhất.

Một đặc tính quan trọng của một tập hợp độc lập cực đại như vậy, đó là nó cũng là chi phối (chứng minh dễ dàng thông qua sự tương phản). Đặc điểm này điều chỉnh một cách cơ bản tầm quan trọng của vấn đề lập công thức: các tập hợp độc lập cực đại là các phần tự nhiên của mạng và cũng tạo nên một tập con các nút mà có thể điều khiển mạng.

Việc xác định các tập hợp độc lập cực đại như được mong đợi, là NP-hoàn thiện. Nó thừa nhận một PTAS cho các sơ đồ vô hướng và các sơ đồ đĩa đơn vị. Đối với các sơ đồ có đĩa vị bị giới hạn, nó có thể xấp xỉ trong $(\Delta + 3)/5$ đối với Δ nhỏ, và trong $O(\Delta \log \log \Delta / \log \Delta)$ đối với các giá trị lớn.

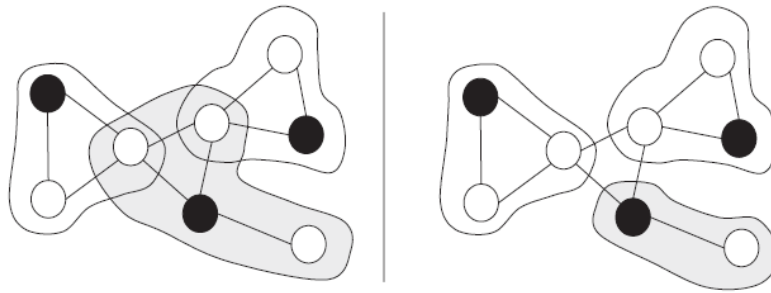
Trong khi sự lập thành công thức tập hợp độc lập cực đại là dễ dàng, nó không cần thiết phản ánh cấu hình mong muốn một cách thực tế của các phân nhóm. Ví dụ, xem xét một sơ đồ $G = (\{v_0, \dots, v_n\}, \{(v_0, v_i) | i = 1, \dots, n\})$ (nghĩa là một nút được nối tới n nút khác mà không được kết nối với nhau). Tập hợp độc lập cực đại cho sơ đồ này là v_1, \dots, v_n , kết quả là n phân nhóm, một phân nhóm có kích thước bằng 2, các phân nhóm khác có kích thước bằng 1. Thực tế hơn, trong nhiều tình huống, sẽ sử dụng nút 0 như chỉ huy của chỉ một phân nhóm đơn. Một trực quan về các mạng như vậy được phản ánh trong phần lớn các khám phá được mô tả sau này mặc dù các mục tiêu tối ưu hóa thực tế thường không được lập công thức đầy đủ. Các mục tiêu như việc trải đều các phân nhóm trên một diện tích đã cho thường được xem là quan trọng.



Hình 10.14 Một sơ đồ ví dụ với một tập hợp độc lập cực đại

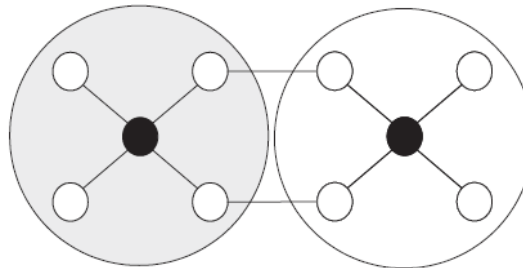
Các phân nhóm có thể chồng nhau không? Khi tạo thành các phân nhóm ngoài tập hợp độc lập cực đại như trình bày trong hình 10.14, câu hỏi đặt ra là phân nhóm nào ấn định các nút không phải chỉ huy phân nhóm, đặc biệt là các nút mà kề cận với hai chỉ huy phân nhóm. Một tùy chọn sẽ ấn định các nút này cho cả hai phân nhóm, kết quả là các phân nhóm chồng nhau. Nếu điều này là không mong muốn, thì cần phải có một quy tắc quyết định nào đó để ấn định một cách rõ ràng các nút là các chỉ huy phân nhóm. Hình 10.15 làm nổi bật các khả năng này.

Làm cách nào các phân nhóm truyền thông? Liệu các phân nhóm có chồng nhau hay không, một nút mà kề cận với hai chỉ huy phân nhóm có thể hỗ trợ một cách tự nhiên trong việc truyền thông giữa hai phân nhóm – nó tạo ra một cổng chính (**gateway**) (các tên khác là cầu (bridge), nút đường bao, hoặc các thuật ngữ tương tự). Ý tưởng là sự truyền thông trong phân nhóm có thể được định tuyến thông qua các chỉ huy phân nhóm, nút mà sau đó sử dụng các cổng chính (**gateway**) cho bất cứ việc truyền thông giữa các phân nhóm nào.



Hình 10.15 Tập hợp độc lập cực đại tạo ra các phân nhóm chồng nhau hoặc không chồng nhau

Tuy nhiên, có thể có các trường hợp, trong đó hai chỉ huy phân nhóm được tách ra bởi hai nút, và không nút đơn nào có thể hoàn thành đầy đủ các trách nhiệm của một cổng chính. Trong tình huống như vậy thì hai nút từ mỗi phân nhóm cùng với nhau có thể hoạt động như một cái gọi là cổng chính phân tán (**distributed gateway**) để cho phép việc truyền thông giữa các phân nhóm. Ý tưởng này được trình bày ở hình 10.16.



Hình 10.16 Hai phân nhóm được nối bởi hai cổng chính phân tán

Các chỉ huy phân nhóm cùng với các cổng chính - gateway (phân tán) một lần nữa hình thành một tập hợp chi phối được kết nối và do đó hình thành một xương sống của toàn bộ mạng. Sự tương đương này cũng có thể hướng theo sự lựa chọn cổng chính, ví dụ như nó có thể không cần thiết để kết nối tất cả các phân nhóm lân cận thông qua các cổng chính (mặc dầu điều này thường được thực hiện bất chấp các cơ hội tối ưu tổng quát) hoặc sự lựa chọn giữa các cổng chính khác nhau có thể được tối ưu hóa bởi các nút ưu tiên để phục vụ như các cổng chính mà có thể kết nối nhiều hơn hai phân nhóm, và trở thành giao điểm của vài phân nhóm. Một lần nữa việc lựa chọn tập hợp tối ưu của các cổng chính để kết nối các chỉ huy phân nhóm đã cho vào một tập hợp được kết nối lại là vấn đề cây Steiner.

Có bao nhiêu các cổng chính tồn tại giữa các phân nhóm? Có thể có vài tùy chọn để kết nối hai chỉ huy phân nhóm thông qua vài cổng chính (phân tán). Phụ thuộc vào mục tiêu tối ưu hóa đối với tập hợp chi phối được kết nối cuối cùng, cần thiết có một mức độ nào đó của sự dư thừa trong việc truyền thông giữa các phân nhóm.

Đường kính cực đại của một phân nhóm là bao nhiêu? Việc biểu diễn của các chỉ huy phân nhóm và mục đích của việc xây dựng một tập hợp độc lập cực đại chỉ rõ một đường kính phân nhóm cực đại của hai nút— mỗi nút trong một phân nhóm là tối đa hai bước nhảy khỏi nút khác bất kỳ. Điều này là không cần thiết cho trường hợp: đôi khi xem xét các phân nhóm

một bước nhảy (mà thường không có các chỉ huy phân nhóm); đôi khi, các phân nhóm đa bước nhảy với các đường kính lớn hơn được sử dụng.

Có một kiến trúc phân tầng của các phân nhóm? Các chỉ huy phân nhóm áp đặt một kiến trúc phân tầng của các nút lên mạng. Thường thì, một kiến trúc phân tầng hai mức như vậy được xem xét là đủ. Tuy nhiên, người ta có thể xem xét các phân nhóm như là các nút trong một sơ đồ mới, cùng với các đường nối giữa các phân nhóm như các cạnh trong sơ đồ này. Đối với sơ đồ này, một lần nữa, việc phân nhóm (hoặc các phương pháp tập hợp chi phối khác) có thể được áp dụng.

10.4.2 Một ý tưởng cơ bản để xây dựng các tập hợp độc lập

Một ý tưởng đơn giản để xây dựng các tập hợp độc lập là khai thác bản chất cục bộ vốn có của sự độc lập – nếu các nút được lựa chọn có thể ngăn trở tất cả các nút lân cận của chúng khỏi bị lựa chọn, thì sinh ra sự độc lập. Do đó, ý tưởng là để tất cả các nút truyền thông với các nút lân cận của nó và để lựa chọn một cách cục bộ các nút tham gia vào tập hợp của các nút độc lập (để cuối cùng trở thành các chỉ huy phân nhóm).

Để làm điều đó, tất cả các nút cần một đặc tính, một đặc tính mà có thể được xác định một cách cục bộ, được trao đổi dễ dàng với tất cả các nút lân cận, và được xếp hạng một cách rõ ràng bởi mỗi nút (các mối liên hệ có thể bị phá vỡ một cách cục bộ). Một ví dụ đơn giản cho một đặc tính như vậy là một số nhận dạng duy nhất của mỗi nút, được sắp xếp ví dụ theo trật tự tăng dần, ở đó các mối liên hệ không thể nào xảy ra. Việc sử dụng các số nhận dạng thực sự là đề xuất đầu tiên cho một thuật toán phân nhóm phân bố.

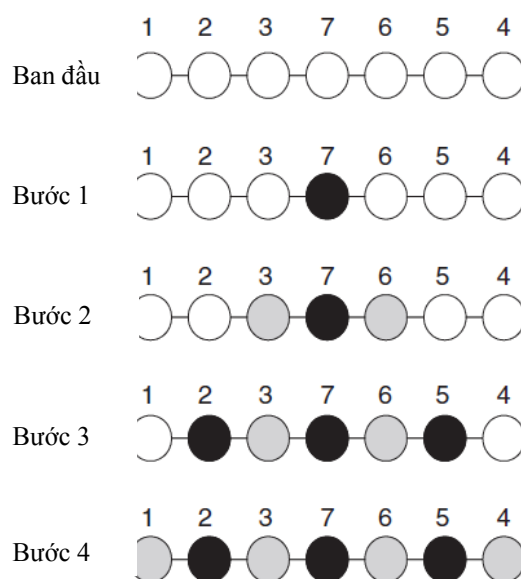
Bất chấp việc lựa chọn chính xác đặc tính này sử dụng cho việc xếp hạng các nút, một thuật toán phân tán cơ bản để tính toán các tập hợp độc lập bắt đầu bằng việc đánh dấu tất cả các nút như là chúng sẵn sàng cho việc trở thành các chỉ huy phân nhóm, nhưng vẫn chưa được quyết định. Trong suốt tiến trình của thuật toán, trạng thái này được chuyển mạch tới “chỉ huy phân nhóm” hoặc “thành viên phân nhóm” (có thể so sánh với các màu trắng, đen, và xám trong phần 10.3). Trong bước đầu tiên, mỗi nút xác định đặc tính xếp hạng cục bộ của nó và trao đổi đặc tính này với tất cả các nút lân cận của nó. Mỗi khi thông tin này là sẵn có, một nút có thể quyết định trở thành một chỉ huy phân nhóm nếu nó có cấp bậc lớn nhất (hoặc nhỏ nhất, phụ thuộc vào định nghĩa) trong tất cả các nút lân cận vẫn chưa được quyết định của nó. Do đó, nút thay đổi trạng thái của nó và thông báo trạng thái mới cho các nút lân cận của nó. Các nút mà biết về một chỉ huy phân nhóm trong khu vực lân cận của chúng sẽ chuyển sang trạng thái thành viên phân nhóm và lần lượt thông báo điều đó cho các nút lân cận của chúng. Chú ý rằng đây là bước thiết yếu: Mỗi khi một nút với một cấp bậc lớn trở thành một thành viên phân nhóm đối với một nút khác nào đó, nó có thể “không chặn” các nút với cấp bậc thấp hơn trong vùng phụ cận của nó để trở thành các chỉ huy phân nhóm trong bản thân chúng. Thuật toán này kết thúc mỗi khi tất cả các nút quyết định để trở thành một chỉ huy phân nhóm hoặc một thành viên phân nhóm.

Thuật toán này được minh họa với một mạng tuyến tính đơn giản như trong hình 10.17. Chú ý, trong bước 1, các nút 2 và 5 không thể trở thành các chỉ huy phân nhóm bởi vì các nút lân cận 3 và 6 của chúng vẫn chưa được quyết định và có khả năng chiếm quyền ưu tiên trên chúng. Mỗi khi các nút 3 và 6 biết rằng nút 7 trở thành một chỉ huy phân nhóm trong vùng phụ cận của chúng, các nút này quyết định trở thành các thành viên phân nhóm và truyền lan

thông tin này tới các nút 2 và 5. Sau đó, các nút này có thể trở thành các chỉ huy phân nhóm trong bước 3.

Thuật toán cơ bản này được xem xét với vài biến đổi nhỏ. Một biến đổi là liệu có thể giữ một cách thực sự các nút phía sau (back nodes) khỏi việc hình thành các phân nhóm miễn là việc quyết định chỉ huy phân nhóm có thể vẫn được xem xét lại, hoặc để cho phép các phân nhóm trung gian được hình thành, các phân nhóm mà sau đó sẽ được tái phân nhóm và các nút có thể tham gia vào chỉ huy phân nhóm khác. Sự biến đổi này có thể đặc biệt hữu ích trong các mạng di động.

Một biến đổi quan trọng khác là bằng cách nào để xếp hạng các nút. Đề xuất đầu tiên là sử dụng các nhận dạng nhỏ nhất (lớn nhất) là (mô tả, ví dụ, “kiến trúc phân nhóm được nối”, với một số dự phòng được thực hiện để trao đổi thông tin kết nối giữa các nút).



Hình 10.17 Thuật toán cơ bản để xác định các tập hợp độc lập, sử dụng các số nhận dạng nút như xếp hạng (các nút màu trắng là chưa được xác định, các nút màu đen là các chỉ huy phân nhóm, các nút màu xám là các thành viên phân nhóm).

10.4.4 Kết nối các phân nhóm

Mỗi khi các chỉ huy phân nhóm được xác định (bằng bất kể thuật toán nào), thường cần thiết phải xác định các cổng chính (có thể phân tán) giữa các phân nhóm. Để cho đơn giản, vấn đề này lại được giảm xuống thành vấn đề cây Steiner.

Nhưng tình huống ở đây đơn giản hơn trong việc thiết lập cây Steiner tổng quát khi vài đặc tính của các chỉ huy phân nhóm đã được biết. Nói riêng, chúng tạo thành một tập hợp độc lập, chi phối trong đó tất cả các nút được tách biệt bởi tối đa ba bước nhảy (trong trường hợp hai thành viên phân nhóm bình thường gặp nhau tại rìa của hai phân nhóm). Một thiết lập như vậy sẽ cho kết quả là một xương sống được kết nối, nếu mỗi chỉ huy phân nhóm kết nối với tất cả các chỉ huy phân nhóm khác mà cách nó tối đa ba bước nhảy. Trong khi đối với một số mạng, điều này có thể có nghĩa là phải cần nhiều kết nối hơn là cần thiết, nhưng cũng có các mạng mà ở đó tất cả các đường kết nối đều cần thiết để đảm bảo việc kết nối.

Cùng với việc xem xét kết nối cơ bản này, các khía cạnh khác như sự cân bằng tải trọng giữa nhiều cổng chính cũng có thể được xem xét.

10.4.5 Quay các chỉ huy phân nhóm

Trở thành một chỉ huy phân nhóm có nghĩa là tiếp quản thêm các nhiệm vụ: tổ chức việc truy cập phương tiện truyền dẫn trong phân nhóm hoặc tham gia vào các quyết định định tuyến. Suy ra, pin của các chỉ huy phân nhóm sẽ có xu hướng bị cạn kiệt nhanh hơn. Thường thì, người ta mong muốn rằng tất cả các nút có dung lượng pin gần bằng nhau tại bất cứ thời điểm nào. Suy ra, trách nhiệm của việc trở thành một chỉ huy phân nhóm nên được chia sẻ giữa các nút. Việc chia sẻ này trong thực tế là một tùy chọn có thể làm được khi thường không chỉ có một giải pháp đơn lẻ đối với một vấn đề tập hợp độc lập cực đại mà còn có nhiều giải pháp tốt như vậy (hoặc gần như vậy).

Để có thể quay các chỉ huy phân nhóm, thuật toán phân nhóm không thể chỉ chạy một lần mà phải được thực hiện lặp đi lặp lại. Việc lặp đi lặp lại như vậy có thể xảy ra theo chu kỳ hoặc có thể được kích hoạt bởi ví dụ: nút di động. Tất nhiên, việc lựa chọn các chu kỳ và các kích hoạt một cách sáng suốt là một vấn đề tối ưu hóa quan trọng, ví dụ phụ thuộc vào tốc độ trung bình của nút, tốc độ tiêu hao của pin...

CÂU HỎI ÔN TẬP CHƯƠNG 10

1. Các tùy chọn cho việc điều khiển cấu trúc liên kết mạng?
2. Các tham số cơ bản để đánh giá hiệu quả và chất lượng của một thuật toán điều khiển cấu trúc liên kết mạng?
3. Trình bày một số kết quả cho việc điều khiển cấu trúc liên kết mạng?
4. Các giới hạn của các tham số chủ chốt?
5. Mục đích và định nghĩa của mạng phân tầng – xương sống?
6. Mục đích và định nghĩa của mạng phân tầng – phân nhóm?

TÀI LIỆU THAM KHẢO

- [1] Holger Karl và Andreas Willig, *Các giao thức và các kiến trúc cho các mạng cảm biến không dây*, NXB Wiley, ISBN: 978-0-470-09510-2, 6/2005.
- [2] Nirupama Bulusu và Sanjay Jha, *Wireless Sensor Networks*, xêri Artech House MEMS, 2005
- [3] Maggie Xiaoyan Cheng và Deying Li (Eds.), *Advances in Wireless Ad Hoc and Sensor Networks*, Springer, 2008
- [4] KS. Phạm Bảo Sơn, *Mạng cảm biến vô tuyến và đánh giá chỉ tiêu của giao thức chọn đường LEACH*, tạp chí Bru chính viễn thông, 11/2006
- [5] Holger Karl, Slide “*Ad hoc and Sensor Networks*”.
- [6] Leonardo Leiria Fernandes, Slide “*MAC Layer Protocols for Sensor Networks*”.
- [7] Satya Sanket Sahoo, Slide “*Sensor Networks*”, Tham khảo: MOBICOM 2002 Tutorial T5 Các mạng cảm biến không dây (CSCI 6760)
- [8] H. Dai và R. Han. Đồng bộ thời gian: Một dịch vụ đồng bộ thời gian hai hướng trọng số nhẹ đối với các mạng cảm biến không dây. *Tạp chí truyền thông và tính toán di động ACM SIGMOBILE*, 8(1): 125–139, 2004.
- [9] B. Das và V. Bharghavan. Định tuyến trong các mạng Ad-Hoc sử dụng các tập hợp chi phối được kết nối nhỏ nhất. Trong *Các kỹ yếu của Hội nghị quốc tế về truyền thông (ICC)*, Montreal, Canada, 6/1997.
- [10] L. Doherty, L. El Ghaoui, và K. S. J. Pister. Ước lượng vị trí lỗi trong các mạng cảm biến không dây. Trong *Các kỹ yếu của IEEE INFOCOM*, trang 1655–1663, Anchorage, AK, 4/2001.
- [11] J. Elson, L. Girod, và D. Estrin. Đồng bộ thời gian mạng tinh – thô sử dụng các quảng bá tham chiếu. Trong *Các kỹ yếu của Hội nghị chuyên đề lần thứ năm về Thiết kế và Thực hiện các hệ điều hành (OSDI 2002)*, Boston, MA, 12/2002.
- [12] J. Wu và H. Li. Về tính toán tập hợp chi phối được kết nối cho việc định tuyến hiệu quả trong các mạng không dây Ad Hoc. Trong *Các kỹ yếu của Hội thảo quốc tế lần thứ 4 về các thuật toán rời rạc và các phương pháp cho truyền thông và tính toán di động*, Boston, MA, 11/8/ 2000.
- [13] S. Ganeriwal, R. Kumar, và M. B. Srivastava. Giao thức đồng bộ định thời cho các mạng cảm biến. Trong *Các kỹ yếu của Hội nghị quốc tế ACM* lần thứ nhất về các hệ thống cảm biến nổi mạng nhúng, trang 138–149, Los Angeles, CA, 11/2003.
- [14] S. Guha và S. Khuller. Các thuật toán xấp xỉ cho Tập hợp chi phối được kết nối *Algorithmica*, 20: 374–387, 1998.
- [15] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, và D. Ganesan. Xây dựng các mạng cảm biến không dây hiệu quả với việc đặt tên mức thấp. Trong *Các kỹ yếu của Hội nghị chuyên đề về các quy tắc của hệ điều hành (SOSP 2001)*, trang 146–159, Lake Louise, Banff, Canada, 10/2001.
- [16] S. M. Kay. *Các nguyên tắc cơ bản của xử lý tín hiệu thống kê: Lý thuyết ước lượng*. Prentice-Hall, Upper Saddle River, NJ, 1993.
- [17] L. Kleinrock và J. Silvester. Bán kính truyền dẫn tối ưu cho các mạng vô tuyến gói hoặc tại sao số sáu là một con số kỳ diệu. Trong *Các kỹ yếu của Hội nghị viễn thông quốc gia*, Birmingham, AL, 12/1978.

- [19] B. Krishnamachari, S. Wicker, R. Bejar, và M. Pearlman. *Các tiến bộ trong lý thuyết thông tin và mã hóa*, chương Các ngưỡng mật độ tới hạn trong Các mạng không dây phân tán. Kluwer.
- [20] E. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, và S. S. Ravi. Các khía cạnh thuật toán của các vấn đề điều khiển cấu trúc liên kết mạng cho các mạng Ad Hoc. Trong *Các kỹ yếu của Hội nghị quốc tế chuyên đề ACM lần thứ ba về Kết nối mạng và tính toán mạng di động Ad Hoc (MobiHoc)*, Lausanne, Switzerland, 2002.
- [21] S. Nesargi và R. Prakash. Hội thảo MANET: Cấu hình của các trạm chủ trong một mạng di động Ad Hoc. Trong *Các kỹ yếu của IEEE INFOCOM 2002*, trang 1587–1596, New York, 6/2002.
- [22] D. Niculescu và B. Nath. Hệ thống định vị Ad Hoc (APS). Trong *Các kỹ yếu của IEEE GlobeCom*, San Antonio, AZ, 11/2001.
- [23] M. D. Penrose. Trên k -Kết nối đối với một sơ đồ ngẫu nhiên hình học. *Các cấu trúc và các thuật ngữ ngẫu nhiên Wiley*, 15(2): 145–164, 1999.
- [24] R. Rajaraman. Cấu trúc điều khiển liên kết mạng và định tuyến trong các mạng Ad Hoc: Một khảo sát. *Tin tức ACM SIGACT*, 33(2): 60–73, 2002.
- [25] V. Ramadurai và M. L. Sichitiu. Sự định vị trong các mạng cảm biến không dây: Một phương pháp tiếp cận thống kê. Trong *Các kỹ yếu của Hội nghị quốc tế 2003 về các mạng không dây (ICWN 2003)*, trang 300–305, Las Vegas, NV, 6/2003.
- [26] P. Santi và D. M. Blough. Phạm vi truyền dẫn tới hạn đối với việc kết nối trong các mạng Ad Hoc không dây thừa thớt. *Kỹ yếu hội nghị IEEE về tính toán di động*, 2: 25–39, 2003.
- [27] C. Savarese, J. Rabay, và K. Langendoen. Các thuật toán xác định vị trí mạnh cho các mạng cảm biến không dây Ad-Hoc phân tán. Trong *Các kỹ yếu của Hội nghị kỹ thuật USENIX thường niên*, Monterey, CA, 2002.
- [28] Y. Shang, W. Ruml, Y. Zhang, và M. Fromherz. Định vị từ kết nối nhỏ. Trong *Các kỹ yếu của Hội nghị chuyên đề quốc tế ACM lần thứ 4 về tính toán và mạng Ad Hoc di động (MobiHoc)*, Annapolis, MD, 2003.