



# 大規模センサネットワークの高効率化および低消費電力化に関する研究

和泉, 慎太郎

---

(Degree)

博士 (工学)

(Date of Degree)

2011-03-25

(Date of Publication)

2011-10-03

(Resource Type)

doctoral thesis

(Report Number)

甲5262

(URL)

<https://hdl.handle.net/20.500.14094/D1005262>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

大規模センサネットワークの高効率化  
および低消費電力化に関する研究

平成 23 年 1 月

神戸大学大学院工学研究科

和泉 慎太郎



# 内容概記

本論文はセンサネットワークの大規模化に関する研究成果をまとめたものである。

近年の情報通信技術（ICT: Information and Communication Technology）の発展により，ユビキタス社会の実現が進んでいる。ユビキタスとは、「いつでも，どこでも，だれでも」がその存在を意識することなく恩恵を受けることのできる環境，技術をあらわす言葉であり，その本来の意味は「遍在」である。あらゆるものに組み込まれたコンピュータがひとの生活を支える「ユビキタスコンピューティング」や，あらゆる場所であらゆるものがネットワークにつながる「ユビキタスネットワーク」は，科学の知見を用いて公共の安全・健康・福祉に有益な技術や環境を実現する「工学」のひとつの理想型と言える。このユビキタス社会の基盤となる技術としてセンサネットワークが注目されている。

センサネットワークとは，センサノード（センシング機能，通信機能，CPUを備えた小型ノード）を対象とするセンシング空間に多数配置し，ネットワークを構築することで高密度な空間センシングをおこなう基盤システムである。これによってあらゆる場所でセンシング情報を取り込み，その情報をどこからでも自由に利用できる環境が得られる。医療，防災，防犯，農林水産業など，さまざまな分野においてセンサネットワークの応用が検討されており，その実現が期待されている。

センサネットワークの実用化における課題は，主にハードウェアリソースを中心とした制約の多さである。従来の伝送速度や通信品質が最優先されるようなネットワーク技術（インターネットなど）とは異なり，消費電力や演算量の削減が大きな目的となる。このため，従来提案されている通信プロトコルやハードウェア構成をそのままセンサネットワークに適用することは困難である。また，センサネットワークの大規模化のためにはネットワーク分散処理への適応，ネットワークトラフィックの削減，センサノードおよびシステム全体の低消費電力化・低コスト化といった大きな課題がある，特に

## ii 内容概記

ノード数が数百個に達するような大規模なセンサネットワークを実現するためには、アプリケーションレベルから通信プロトコル、回路・アーキテクチャまで含めた包括的な研究開発が必要不可欠となる。

本論文は6章で構成されており、第1章は序論である。第2章ではセンサネットワークの特徴を述べ、センサネットワークの大規模化にともなう課題についてまとめる。

第3章では従来の大規模マイクアレイにおける演算量や消費電力の問題を解決する技術として、センサネットワーク技術を用いたネットワーク型マイクアレイシステムを提案する。提案システムでは、小規模なマイクアレイ・センサノード（サブアレイ）を多数配置し、それらの間で有線ネットワークを構築して協調動作をおこなうことで、大規模なマイクアレイ音声処理システムを実現する。また本研究では、ネットワークを介した分散処理に適した音源分離のアルゴリズムを新たに提案した。FPGA ボード SUZAKU を用いて 16ch サブアレイの試作をおこない、3つのサブアレイを用いた実証実験によって提案システムの動作を確認した。

従来のデータ収集のみを目的としたアプリケーションとは異なり、ネットワーク型マイクアレイシステムには「リアルタイム処理」、「センサノード間の時刻同期」、「ネットワーク分散処理」という3つの機能が必要であった。これらの機能は、他のより実用的なアプリケーションでも必須となるものであり、本研究はセンサネットワークの応用分野を拡大するためのテストベッドとしての役割も果たすものである。

第4章ではセンサネットワークにおいて経路構築などに必要不可欠なブロードキャストに着目し、待ち時間制御によるカウンタベース方式の改良アルゴリズム（RAD-Extension と HCA-RAD Extension）を提案する。また、ブロードキャストはネットワーク層だけでなくアプリケーション層でも必要とされるプロトコルである。提案方式では、コストの高いアルゴリズムやハードウェアを必要とせず、ノード密度にあわせて再送信確率を制御できるという従来方式の利点を生かしつつ、さらに再送信ノード数を削減することを目的とした。提案方式では、メッセージ受信ごとに待ち時間を増加させることで、各ノードの面積カバー率を改善する。これによってパケット到着率を維持したまま全体で再送信ノード数を削減できることを述べた。さらに、待ち時間のスロット化によって、経路長も考慮して再送信の優先順位を設定するアルゴリズムを提案した。また、シミュレーションによる比較をおこない、従来方式と比較して提案方式では約10%の再送信ノード数削減効果が得られることを示した。

本研究の提案アルゴリズムは大規模なセンサネットワークのトラフィックを削減することでスケラビリティを高め、また同時にネットワークシステム全体の長寿命化に貢献するものである。

第5章では低消費電力化センサノード LSI の垂直統合設計技術について詳述する。ワイヤレスセンサネットワークを実現するためには、センサノードの消費電力を削減し、システムの可用時間を延ばさなければならない。そのためには、ネットワークの各階層に対して個別に最適化するだけでなく、協調して設計すること(垂直統合設計)が必要不可欠である。しかし、通信プロトコルからセンサノードの回路・アーキテクチャレベルまでの一貫した低消費電力化をおこなうような研究は、ほとんど成されていなかった。

そこで本研究では、ネットワークの階層構造における中間層である MAC (Media Access Control) 層を中核とし、上下方向展開による垂直統合設計をおこなった。まず、センサノード向け低消費電力 MAC プロトコルとして LPL (Low Power Listening) を元にした I-MAC (Isochronous-MAC) を用いることで、アルゴリズム階層で消費電力を削減した。次に、MAC 層の通信処理を専用ハードウェア化した MAC シーケンサと汎用マイクロプロセッサの組み合わせによって低消費電力なセンサノード用マイクロコントローラを実現し、MAC プロトコルと LSI アーキテクチャ間の相互最適化をおこなった。さらに、無線通信回路まで含めて 1-chip システム LSI としてセンサノードを設計し、MAC シーケンサを中核とした電源管理機構によってセンサノード全体の消費電力を削減した。

提案したセンサノードアーキテクチャに基づいてセンサノードの試作し、データ収集時の平均消費電力  $58\mu\text{W}$  を達成した。

最後に第6章で本論文の結論を述べる。

本論文ではセンサネットワークにおいて、ネットワーク分散処理に対応するアプリケーション層、通信トラフィックを削減するネットワーク層、センサノードの消費電力を削減する MAC 層と物理層の研究成果について記述した。これらの提案技術によって大規模センサネットワークが実現され、センサネットワークの普及につながる新たなキラーアプリケーションの創出が期待できる。



# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	センサネットワークの技術背景	1
1.2	研究の目的と本論文の構成	2
<b>第2章</b>	<b>センサネットワークの特徴と大規模化にともなう課題</b>	<b>5</b>
2.1	センサネットワークの特徴	5
2.2	大規模化にともなう課題	6
<b>第3章</b>	<b>センサネットワーク技術を用いたマイクロホンアレイの大規模化</b>	<b>9</b>
3.1	緒言	9
3.2	マイクロホンアレイによる音声インタフェースと課題	9
3.3	ネットワーク型マイクロホンアレイの提案	10
3.4	ネットワーク分散処理のための音源分離アルゴリズム	13
3.4.1	遅延和ビームフォーミングの概要	13
3.4.2	サブアレイ間の遅延和ビームフォーミング	13
3.4.3	データ集約	15
3.5	サブアレイ間の時刻同期	15
3.5.1	時刻ずれによる影響の基礎検討	16
3.5.2	センサネットワークにおける時刻同期の課題	18
3.5.3	既存の時刻同期技術	19
	GPS	19
	長波帯標準電波	19
	Reference Broadcast Synchronization (RBS)	20
	Timing-sync Protocol for Sensor Networks (TPSN)	22

Flooding Time Synchronization Protocol (FTSP) . . . . .	23
3.5.4 提案システムにおける時刻同期の実装 . . . . .	24
3.6 FPGA を用いたプロトタイプ実装と実証実験 . . . . .	25
3.6.1 FPGA 実装 . . . . .	26
3.6.2 性能評価実験 . . . . .	27
3.7 結言 . . . . .	29
<b>第 4 章 待ち時間制御によるブロードキャストアルゴリズムの高効率化</b>	<b>33</b>
4.1 緒言 . . . . .	33
4.2 ブロードキャストアルゴリズムの課題と関連研究 . . . . .	34
4.2.1 経路制御とブロードキャスト . . . . .	34
4.2.2 ブロードキャストの問題点 . . . . .	34
4.2.3 ブロードキャストの改良アルゴリズム . . . . .	36
確率的アルゴリズム . . . . .	36
決定性アルゴリズム . . . . .	38
4.3 カウンタベース方式ブロードキャストアルゴリズム . . . . .	39
4.4 待ち時間制御によるカウンタベース方式の改良 . . . . .	41
4.4.1 基礎検討 . . . . .	42
4.4.2 RAD Extension の提案 . . . . .	44
4.4.3 Hop Count Aware RAD Extension の提案 . . . . .	47
4.5 シミュレーションによる性能評価 . . . . .	49
4.5.1 シミュレーションモデル . . . . .	50
4.5.2 シミュレーション結果と考察 . . . . .	51
4.6 結言 . . . . .	57
<b>第 5 章 低消費電力センサノード VLSI の垂直統合設計</b>	<b>59</b>
5.1 緒言 . . . . .	59
5.2 各プロトコル階層における低消費電力化の課題と関連研究 . . . . .	60
5.2.1 ネットワーク層 . . . . .	60
経路制御 . . . . .	61
データ集約 . . . . .	63

5.2.2	データリンク層 (MAC 層) . . . . .	64
	MAC プロトコルへの要求 . . . . .	65
	MAC プロトコルに関する研究 . . . . .	69
5.2.3	物理層における低消費電力化 . . . . .	70
	物理層への要求 . . . . .	70
	物理層に関する研究 . . . . .	71
5.2.4	各階層ごとの低消費電力化における課題 . . . . .	72
5.3	MAC プロトコルの開発 . . . . .	73
5.3.1	従来手法 (Low Power Listening) の課題 . . . . .	73
5.3.2	Isochronous-MAC プロトコルの概要 . . . . .	74
5.3.3	Isochronous-MAC のプリアンブル長 . . . . .	75
5.3.4	Isochronous-MAC の衝突回避 . . . . .	77
5.3.5	RTS/CTS による Isochronous-MAC の改良 . . . . .	78
5.4	MAC シーケンサの設計 . . . . .	79
5.4.1	通信制御用プロセッサの構成 . . . . .	80
5.4.2	MAC シーケンサの構成 . . . . .	81
5.4.3	ステートマシンの構成 . . . . .	82
5.5	1-chip システム LSI の設計 . . . . .	85
5.5.1	センサノードのアーキテクチャ . . . . .	86
5.5.2	低消費電力化のための工夫 . . . . .	87
5.6	センサノードの VLSI 実装 . . . . .	90
5.7	消費電力の評価 . . . . .	91
5.7.1	各モジュールの消費電力 . . . . .	91
5.7.2	ネットワークシミュレーションによる評価 . . . . .	92
5.8	結言 . . . . .	95
<b>第 6 章 結論</b>		<b>97</b>
<b>謝辞</b>		<b>101</b>
<b>参考文献</b>		<b>103</b>



# 目次

1.1	本論文の構成	3
1.2	プロトコルスタックと各章の対応	3
3.1	ネットワーク型マイクアレイ	11
3.2	サブアレイのワークフロー	12
3.3	ゼロ交差数法	12
3.4	指向性の形成	14
3.5	遅延和ビームフォーミング	14
3.6	ネットワーク分散処理による遅延和ビームフォーミング	15
3.7	データ集約	16
3.8	ノード配置と時刻誤差	16
3.9	時刻ずれ $d$ が 0 の場合にサブアレイ枚数と SNR の関係	17
3.10	時刻ずれ $d$ と SNR の関係	17
3.11	時刻同期の誤差要因	18
3.12	日本における電波時計基地局	20
3.13	Reference Broadcast Synchronization (RBS)	21
3.14	Timing-sync Protocol for Sensor Networks (TPSN)	22
3.15	Flooding time synchronization protocol (FTSP)	23
3.16	周波数ずれの線形補間	25
3.17	サブアレイのプロタイプ実装	25
3.18	サブアレイのブロック図	26
3.19	発話推定および音源定位実験におけるサブアレイ配置と環境の設定	27
3.20	発話推定実験	28
3.21	音源定位実験	28

3.22	FTSP による時刻同期の動作波形 . . . . .	28
3.23	遅延和ビームフォーミング動作実験の環境設定 . . . . .	29
3.24	遅延和ビームフォーミングによって得られた信号の周波数スペクトラム	30
3.25	遅延和ビームフォーミングによって得られた信号の周波数スペクトラ ム (時間軸で見た変化) . . . . .	31
3.26	サブアレイ配置と指向性の関係 . . . . .	31
4.1	ブロードキャストによる経路構築 . . . . .	35
4.2	ブロードキャストストームの例 . . . . .	36
4.3	面積カバー率 (EAC: Expected additional coverage) . . . . .	40
4.4	メッセージの重複受信回数と面積カバー率の関係 . . . . .	40
4.5	カウンタベース方式の動作例 (1 ホップ, $C_{th} = 4$ の場合) . . . . .	42
4.6	カウンタベース方式の動作例 (2 ホップ, $C_{th} = 4$ の場合) . . . . .	43
4.7	RAD Extension の動作例 (1 ホップ, $C_{th} = 4$ の場合) . . . . .	45
4.8	RAD Extension の動作例 (2 ホップ, $C_{th} = 4$ の場合) . . . . .	46
4.9	RAD Extension の動作例 (経路構築, $C_{th} = 3$ の場合) . . . . .	47
4.10	HCA-RAD Extension の動作例 (1 ホップ, $C_{th} = 4$ の場合) . . . . .	49
4.11	HCA-RAD Extension の動作例 (2 ホップ, $C_{th} = 4$ の場合) . . . . .	50
4.12	HCA-RAD Extension の動作例 (経路構築, $C_{th} = 3$ の場合) . . . . .	51
4.13	HCA-RAD Extension による RAD のスロット化: 図 4.9 の例における タイミングチャート (a) と, 図 4.9 の例におけるタイミングチャート (b) .	52
4.14	$C_{th}$ に対するパケット到着率と再送信ノード数の関係 ( $T_{max} = 15s$ ) . . . . .	53
4.15	$C_{th}$ と平均ホップ数の関係 ( $T_{max} = 15s$ ) . . . . .	53
4.16	$T_{max}$ に対するパケット到着率 (a) と再送信ノード数 (b) の関係 ( $C_{th} = 4$ )	54
4.17	$T_{max}$ と遅延の関係 ( $C_{th} = 4$ ) . . . . .	55
4.18	遅延と再送信ノード数の関係 ( $C_{th} = 4$ ) . . . . .	55
4.19	ノード密度に対するパケット到着率 (a) と再送信ノード数 (b) の関係 ( $C_{th} = 4, T_{max} = 15s$ ) . . . . .	56
4.20	データ収集時の平均消費電力 ( $C_{th} = 4, T_{max} = 15s$ ) . . . . .	56

5.1	アドレスセントリックなルーティングとデータセントリックなルーティング . . . . .	61
5.2	Directed Diffusion . . . . .	62
5.3	完全集約と線形集約 . . . . .	64
5.4	MAC プロトコルが保証する 3 種類の通信 . . . . .	65
5.5	間欠動作 . . . . .	67
5.6	通信不可能問題 . . . . .	68
5.7	隠れ端末問題 . . . . .	68
5.8	RTS/CTS 交換 . . . . .	69
5.9	LPL による送受信動作 . . . . .	74
5.10	I-MAC による送受信動作 . . . . .	75
5.11	起動周期と平均消費電力の関係 . . . . .	75
5.12	時刻ずれとプリアンプル長の関係 . . . . .	76
5.13	プリアンプル長と 1 日の同期回数 $C$ の関係 ( $D = 350$ and $100ms$ ). . . . .	77
5.14	I-MAC における衝突回避 . . . . .	78
5.15	RTS/CTS の有無による伝送レートと無線機消費電力の関係 . . . . .	79
5.16	通信制御用プロセッサの構成 . . . . .	80
5.17	I-MAC の通信スキーム . . . . .	81
5.18	I-MAC シーケンサのブロック図 . . . . .	82
5.19	非同期クロック対策モジュールブロック図 . . . . .	82
5.20	非同期クロック対策モジュールタイミング図 . . . . .	83
5.21	I-MAC の受信時状態遷移 . . . . .	83
5.22	I-MAC の送信時状態遷移 . . . . .	84
5.23	1-chip センサノード LSI のブロック図 . . . . .	87
5.24	パワーゲーティングとクロックゲーティング . . . . .	88
5.25	バスキーパー . . . . .	88
5.26	電源管理モジュール . . . . .	89
5.27	提案センサノード LSI のデザインフロー . . . . .	91
5.28	提案センサノード LSI のチップ写真 . . . . .	92
5.29	提案センサノード LSI のパワーバジェット . . . . .	94

xii 図目次

5.30	平均消費電力の比較 . . . . .	95
5.31	TICER による送受信動作 . . . . .	96

# 表目次

3.1	時刻同期技術の比較 . . . . .	24
5.1	センサノードの比較 . . . . .	72
5.2	提案センサノード LSI のチップ諸元 . . . . .	90
5.3	消費電力測定結果 . . . . .	93
5.4	シミュレーション条件 . . . . .	93
5.5	平均動作時間 . . . . .	94



# 第 1 章

## 序論

### 1.1 センサネットワークの技術背景

近年の情報通信技術（ICT: Information and Communication Technology）の発展により、ユビキタス（Ubiquitous）社会の実現が進んでいる。ユビキタスとは、「いつでも、どこでも、だれでも」がその存在を意識することなく恩恵を受けることのできる環境、技術をあらわす言葉であり、その本来の意味は「遍在」である。あらゆるものに組み込まれたコンピュータがひとの生活を支える「ユビキタスコンピューティング」や、あらゆる場所であらゆるものがネットワークにつながる「ユビキタスネットワーク」は、科学の知見を用いて公共の安全・健康・福祉に有益な技術や環境を実現する「工学」のひとつの理想型と言える。このユビキタス社会の基盤となる技術としてセンサネットワークが注目されている。

センサネットワークとは、センサノード（センシング機能、通信機能、CPU を備えた小型ノード）を対象とするセンシング空間に多数配置し、ネットワークを構築することで高密度な空間センシングをおこなう基盤システムである。これによってあらゆる場所でセンシング情報を取り込み、その情報をどこからでも自由に利用できる環境が得られる。医療、防災、防犯、農林水産業など、さまざまな分野においてセンサネットワークの応用が検討されており、その実現が期待されている。

センサネットワークの実用化における課題は、主にハードウェアリソースを中心とした制約の多さである。従来の伝送速度や通信品質が最優先されるようなネットワーク技術（インターネットなど）とは異なり、消費電力や演算量の削減が大きな目的となる。また、センサネットワークと類似した技術としてモバイルアドホックネットワーク（MANET）があげられるが、センサネットワークを構成するセンサノードの処理性能や

## 2 第1章 序論

バッテリー容量は MANET よりもさらに限られている。このため、従来提案されている通信プロトコルやハードウェア構成をそのままセンサネットワークに適用することは困難である。特に、ノード数が数百個に達するような大規模なセンサネットワークを実現するためには、アプリケーションレベルから通信プロトコル、ハードウェアまで含めた包括的な研究開発が必要不可欠となる。

### 1.2 研究の目的と本論文の構成

本研究では、通信プロトコルスタックにおけるアプリケーション層から物理層までの階層間協調設計によるセンサネットワークの大規模化を目的とする。以下に本論文の構成を述べる。また、図 1.1 に本論文の構成をまとめ、図 1.2 に各章で研究対象とするプロトコル階層を示す。

**第1章** センサネットワークに関する技術背景、および本論文の研究内容について概説する。

**第2章** センサネットワークの大規模化にともなう課題と関連研究について記述する。

**第3章** マイクロホンアレイ・センサノードを用いた音声信号ビームフォーミングのネットワーク分散処理技術について記述する。

**第4章** センサネットワークの経路構築や時刻同期に必要なブロードキャストの高効率化技術について記述する。

**第5章** データリンク層を中核とした階層間協調設計によるセンサノード VLSI の低消費電力化技術について記述する。

**第6章** 本研究で得られた研究成果について総括する。

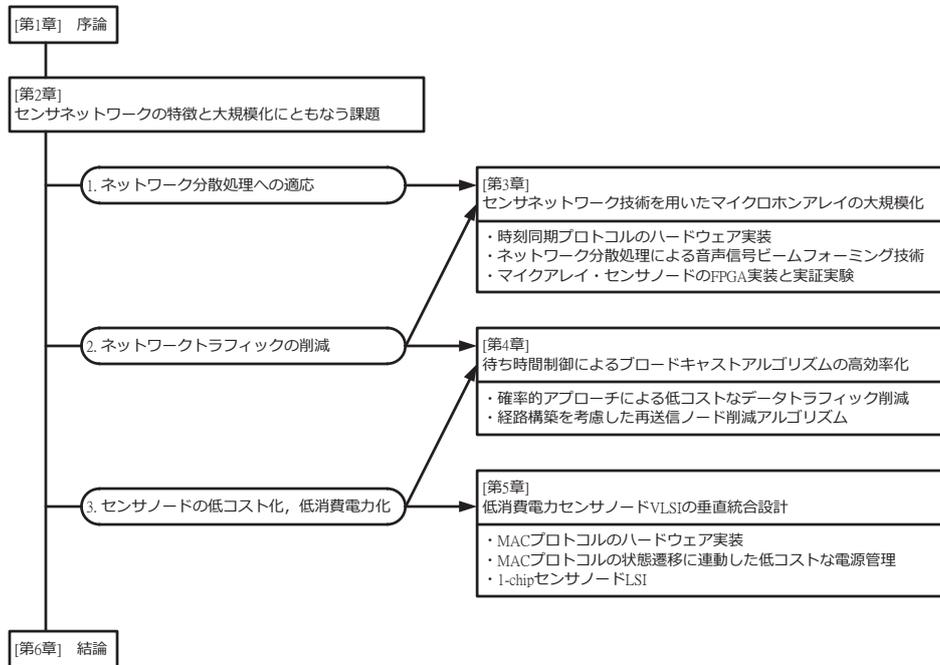


図 1.1 本論文の構成

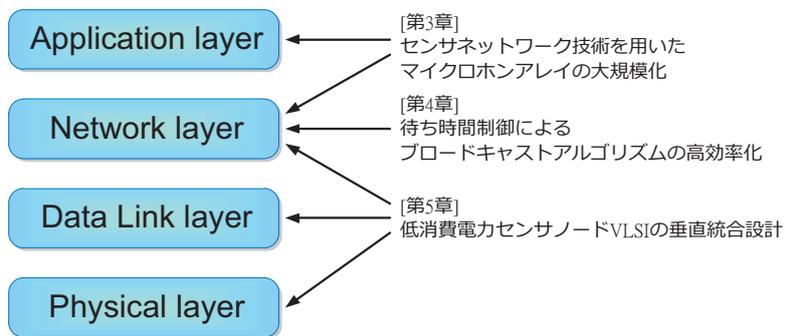


図 1.2 プロトコルスタックと各章の対応



## 第2章

# センサネットワークの特徴と 大規模化にともなう課題

第1章で述べたように，センサネットワークは従来のネットワークシステムとは異なる特徴を持っている．本章ではセンサネットワークの特徴をまとめ，センサネットワークを大規模化するための課題を述べる．

### 2.1 センサネットワークの特徴

センサネットワークにおける最大の特徴にして最大の制約となっているのは，その電源である．バッテリーによって駆動するセンサノードが一般的であり，バッテリーレスを目的とした研究（ワイヤレス給電，エナジーハーベスティングなど）も存在するが，いずれの場合も使用可能な電力は限られている．このため，センシング，ネットワーク処理，データ処理を低消費電力に実行する必要がある．搭載できるハードウェア（通信回路，CPU，データストレージなど）も限られてくる．

また，センサノードの信頼性が低いというのも大きな特徴である．バッテリーの枯渇や故障によってセンサノードが動作不能に陥るような状況は頻繁に発生すると考えられる．さらに，消費電力の制約により通信には単純な方式を採用することが多く，データ誤りや妨害波によるパケットロスが多く発生するという問題もある．このため，あらかじめ冗長経路を持たせるなどシステムレベルでの対策が必要となる．

さらに，センサノードの配置も他のネットワーク技術とは異なった特徴を持つ．上記の信頼性の問題への対策として，あるいは高密度なセンシングへの要求から多数のノードを配置するため，ノードの密度が高くなる傾向がある．また，物理的な配置だけでなく，故障などによってネットワークの論理的な経路変更が頻繁に起こるのも特徴である．

## 2.2 大規模化にともなう課題

1990年代にセンサネットワークの概念が提唱されてから10年以上が経過し、その実現に向けてさまざまな研究がおこなわれてきた。しかし、実際に実用化までたどり着いた例は少なく、特に大規模な（数百ノードに達するような）センサネットワークは未だシミュレーションによる研究段階に留まっているのが現状である。これは、前節で述べたように、センサネットワークは従来のネットワークとは異なる特徴を持ち、それによるさまざまな制約が存在するためである。以下にセンサネットワークを大規模化するための3つの課題について示す。

1. ネットワーク分散処理に適応するアプリケーションの創出
2. スケーラビリティの障害となるネットワークトラフィックの削減
3. センサノードおよびネットワークシステム全体の低消費電力化・低コスト化

1つ目の課題は、アプリケーションのネットワーク分散処理への適応である。最初にセンサネットワークが提案されたとき、そのアプリケーションは索敵・監視のための軍事利用であった（DARPAのSmart Dustなど[1]）。それから現在までにさまざまなアプリケーションが提案されてきたが、いわゆるキラーアプリケーションと呼べるものはいまだに登場していない。アプリケーション分野での研究は多くなされているが、センサノードからのデータ収集のみを目的としたものがほとんどであり、基本的な処理内容はSmart Dustの頃から進化しているとは言えない。一般的な分散コンピューティングと異なり、センサネットワークはネットワーク性能（速度、帯域、信頼性など）が貧弱であり、ノード自体の処理性能もそれほど高くないため、センシングとデータ送受信以外の処理を実装する難しさがアプリケーション開発を滞らせていると考えられる。しかし、より魅力的なアプリケーションを実現するためには、センシングのためのノード間時刻同期やリアルタイム処理、またセンサノード自身による信号処理など、アプリケーションの要求に合わせたデータ転送、データ処理が求められる。そのためには、ネットワーク技術とそれを支える回路技術まで意識した、垂直統合設計によるシステム構築が重要である。

2つ目の課題は、ネットワークトラフィックの削減である。前節で述べたように、センサネットワークは従来のネットワークに比較して配置されるノードの密度が高いとい

う特徴がある。しかし、処理性能に限界のあるセンサノードを高密度に配置するということは、ネットワークの大規模化にともなうトラフィックの増大に対して著しく耐性を欠くことを意味する。無線通信を用いる場合はノードが直接通信できる範囲内に多数のノードが置かれるため、送信データの衝突、送信待ちによる遅延の増大などさまざまな問題を引き起こす。さらに、有線・無線を問わず、ネットワーク規模の増大は送受信データ量の増加に直結する。したがって、センサネットワークの大規模化にはネットワークトラフィックの削減が必要不可欠である。このためには従来の通信速度や信頼性を重視した通信プロトコルでは不十分であり、センサネットワークに適した専用のプロトコルが要求される。また、多くの場合センサノードの演算量とネットワークトラフィックはトレードオフの関係にあり、センサノードの回路・アーキテクチャレベルまで意識したプロトコルの開発が要求される。

3つ目の課題は、センサノードの低消費電力化・低コスト化である。多数のノードを用いる大規模センサネットワークでは、システムのメンテナンスコストが実用化に対する大きな壁となる。センサノードは通常バッテリーで動作すると考えられているが、センサネットワークでは対象となるセンシング空間に多数のセンサノードを分散配置するため、バッテリーの交換が困難である。したがって、メンテナンスコストを削減するためには、センサノードの低消費電力化・長寿命化が不可欠である。しかし、従来の階層構造のプロトコル開発では消費電力の削減に限界があり、通信プロトコルと回路・アーキテクチャの相互最適化が必要となる。また、センサノードのコストを下げ、安価に大量生産可能とすることも重要な課題である。そのためにはセンサノードの部品点数を削減する必要がある。センサノードを SoC (System on Chip) として実装することが非常に有効となる。また、センサノード SoC は低消費電力化にも寄与する。

上記の3つの課題それぞれについては多数の研究がなされているが、ほとんどの研究はプロトコル、回路・アーキテクチャ、アプリケーションの各階層で独立して課題に取り組んだものである。大規模センサネットワーク実現のためには、従来の各階層に分かれた研究ではなく、階層間での協調を意識した包括的な研究をおこなわなければならない。



## 第3章

# センサネットワーク技術を用いた マイクロホンアレイの大規模化

### 3.1 緒言

本研究では、1つのセンサノードに複数のマイクを配置したマイクロホンアレイ・センサノードを用いるユビキタスセンサネットワークシステムを提案する。複数のノードがマルチホップネットワークを構築することによって、多数のマイク入力を用いた音源定位や信号対雑音比の改善など、様々なアプリケーションが実現できる。また、各ノードが分散処理をおこない、送信するデータの集約をおこなうことで、電力消費、データ伝送量、スケーラビリティなどの面においても優れたシステムが実現できる。

### 3.2 マイクロホンアレイによる音声インターフェースと課題

我々の身の回りには数多くの家電製品・電子機器があふれており、ほとんどの機器は手を用いた操作を前提にしたインターフェースを備えている。しかしそのようなインターフェースは、操作の煩わしさや、利用時に手がふさがってしまうという問題を抱えている。このため、ハンズフリーな音声インターフェースへの注目が高まっている。

音声インターフェースでは、信号対雑音比（SNR）の高い音声を集音するため、マイクロフォン（以降マイク）を話者の口に近い位置に設置し、さらに指向性を口元に向けるのが通常である。しかしこのような理想的な状態を実現するためには、ヘッドセットやピンマイクなどを用いる必要があり、インターフェースとしての煩雑さを解決しているとは言い難い。

マイクの存在を意識せずに使用できるユビキタスな音声インターフェース実現のためには、マイクが話者から離れた場所に設置される場合も考慮しなければならない。この

ような条件下では、集音される音の SNR は低くなり、音声の反射や残響の影響が大きくなってしまいます。また、複数の話者が存在するような状況では、それぞれの音声を分離・抽出する必要があります。

これらの問題を解決するアプローチとして、マイクを複数配置したマイクアレイを用いる方法が提案されている。マイクアレイを用いることによって、単一のマイクでは得られない空間的な音の情報を取得でき、音の到来方向の推定や指向性の制御をおこなうことができる。これによって反射や残響の抑制、複数音声の分離・抽出が可能となり、SNR を改善させることができる。

マイクアレイによる音源定位や音源分離の性能は、使用するマイクの数が増えるほど向上する。マイクアレイの大規模化に関する研究では、東京理科大学の 128ch アレイ [2] や電気通信大学の 156ch アレイ [3]、Brown University と Rutgers University の 512ch アレイ [4][5]、Massachusetts Institute of Technology の 1,020ch アレイ [6] などが存在し、音声認識における認識率の向上などが確認されている。しかし、演算量や消費電力、コストの問題から、実用化されているマイクアレイのマイクの本数は数個から十数個に留まっている。実用的な大規模マイクアレイを実現するためには、これらの大規模化に付随する問題を解決する必要がある。

そこで我々は、小規模なマイクアレイ（サブアレイ・センサノード）を多数配置し、サブアレイ間でマルチホップネットワークを構築するネットワーク型マイクアレイシステムを提案する。各サブアレイで分散処理をおこなうことによって、マイクアレイの大規模化に伴う消費電力やコストなどの増加を抑制できる。また、ネットワーク上で音声データの集約をおこなうことで、データ伝送量を削減し、スケーラビリティを確保することができる。

### 3.3 ネットワーク型マイクロホンアレイの提案

図 3.1 に我々の提案するネットワーク型マイクアレイの構成を示す。サブアレイは 16 個のマイクを持ち、有線ネットワークによってメッシュ型に接続される。各サブアレイが 16ch の音声処理をおこない、その結果をネットワーク上で共有することで、システム全体を一つの大規模なマイクアレイとして動作させることができる。

各サブアレイがおこなう処理の流れを図 3.2 に示す。主な処理は以下の 3 つに分けられる。

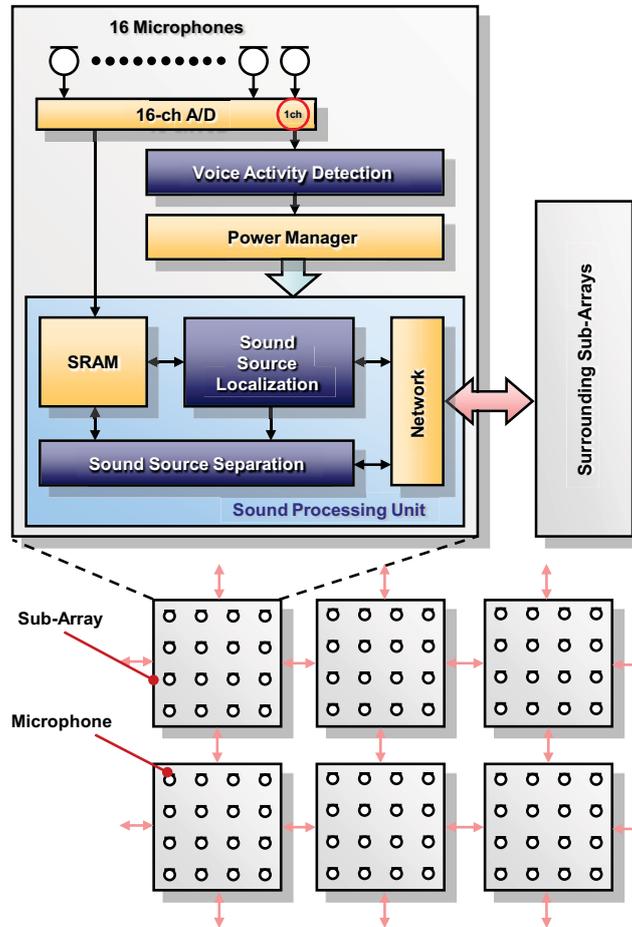


図 3.1 ネットワーク型マイクアレイ

- 1) 発話推定 (Voice Activity Detection: VAD)
- 2) 音源の位置推定 (Sound Source Localization: SSL)
- 3) 遅延和ビームフォーミングによる音源分離 (Sound Source Separation: SSS)

まず、発話を検知するまでの間 (無音区間) は、ある程度距離の離れた幾つかのサブアレイだけが 1ch のマイク入力を用いてゼロ交差数法による発話推定 [7] をおこなう。ゼロ交差数法は、音声信号のゼロ交差数の差異を利用することで、少ない演算量で精度良く発話区間と無音区間を判別できる (図 3.3)。発話を検知したサブアレイは、Wake-up メッセージをネットワーク上にブロードキャストすることで、周辺の発話推定をおこなっていないサブアレイを起動させる。このように、無音区間では使用するノード数とマイク数を最小限に抑え、さらにすべてのノードが音声処理ユニットの電源を遮断することによって、マイクアレイネットワーク全体の待機時消費電力を削減する。

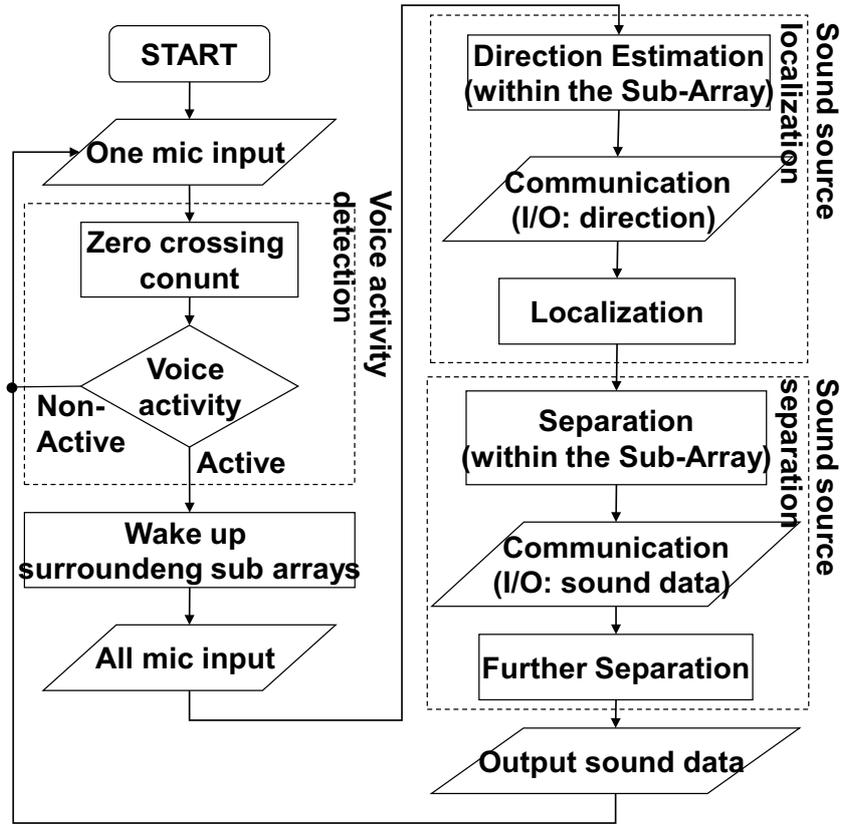


図 3.2 サブアレイのワークフロー

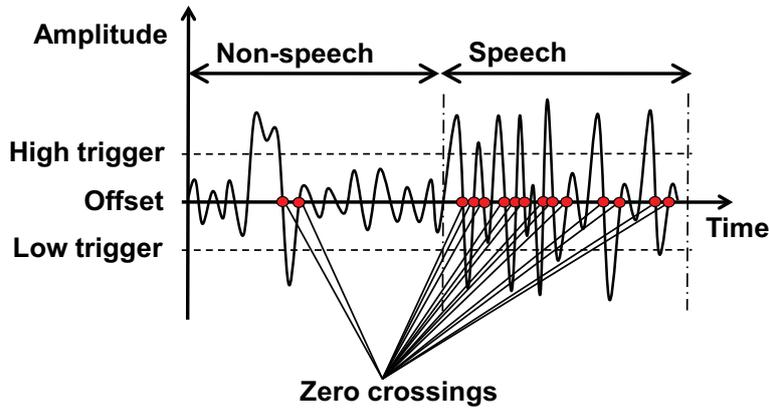


図 3.3 ゼロ交差数法

次に、マイクアレイシステムは音源定位をおこなう。音源定位は、サブアレイによる方向推定と、ベースステーション（サーバに接続されたサブアレイ）による位置推定の2段階に分かれている。各サブアレイは発話検知後、または Wake-up 受信後すぐに複数のマイクを用いて音源方向推定を開始する。方向推定には、少数のマイクでも精度の高

い MUSIC (MUltiple Signal Classification) 法 [8] を用いる。音源方向推定完了後、すべてのサブアレイは推定結果をベースステーションへ向けて送信する。ベースステーションは収集した各サブアレイの方向推定結果を用いて幾何学的に音源位置の推定をおこなう。結果をすべてのサブアレイに向けてブロードキャストする。大規模なマイクアレイに対して MUSIC 法を用いることは演算量の観点から現実的ではないが、提案システムではサブアレイごとの分散処理によって MUSIC 法による音源定位を実現している。

最後に、各サブアレイはベースステーションから受け取った位置推定結果を用いて遅延和ビームフォーミング [7] による音源分離をおこなう。各サブアレイ内で 16ch の音声 を 1ch に集約し、それをベースステーションへ向けて順次送信する。このように段階的に集約することによって、SNR を向上させると同時にネットワーク上のデータ転送量も削減することができる。音源分離については次章で詳しく述べる。

上記の処理によって得られた SNR の高い音声はベースステーションからサーバに転送され、サーバ上で音声認識などのアプリケーションに用いることができる。

### 3.4 ネットワーク分散処理のための音源分離アルゴリズム

提案システムでは音源分離に遅延和ビームフォーミングを用いる。本章では、ネットワークを介してサブアレイ間で音源分離処理をおこなうための方法を述べる。

#### 3.4.1 遅延和ビームフォーミングの概要

遅延和ビームフォーミングでは、図 3.4 に示すように、特定方向の音声に向けて指向性を形成することで雑音を抑圧する。図 3.5 に示すように、各マイクで収録した音声と同位相になるよう遅延を付加し加算することで、指向性の形成を実現する。

#### 3.4.2 サブアレイ間の遅延和ビームフォーミング

提案システムでは、複数のマイクで得た音声データを遅延和ビームフォーミングにより集約し、音源位置で集音するのと同等の音声を再現する。つまり、音声データを音が発せられた時刻で整列させて加算する。本システムではデータの整列をローカル整列 (サブアレイ内処理) とグローバル整列 (ネットワーク処理) の二段階に分けて実行する (図 3.6)。

提案アルゴリズムでは、まず各マイクに関してローカル遅延量を計算する。この遅延

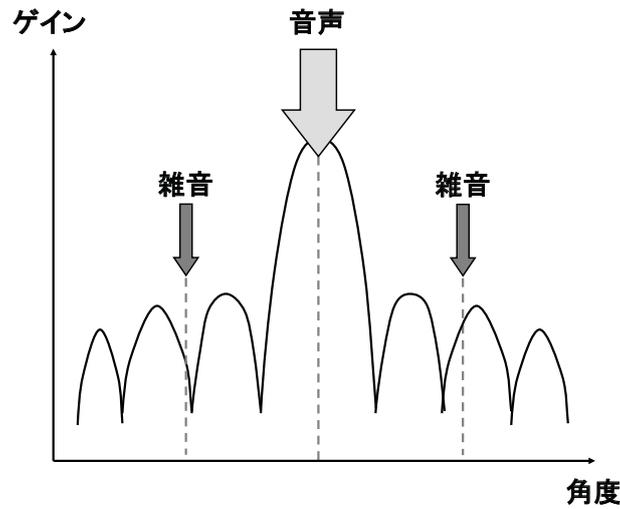


図 3.4 指向性の形成

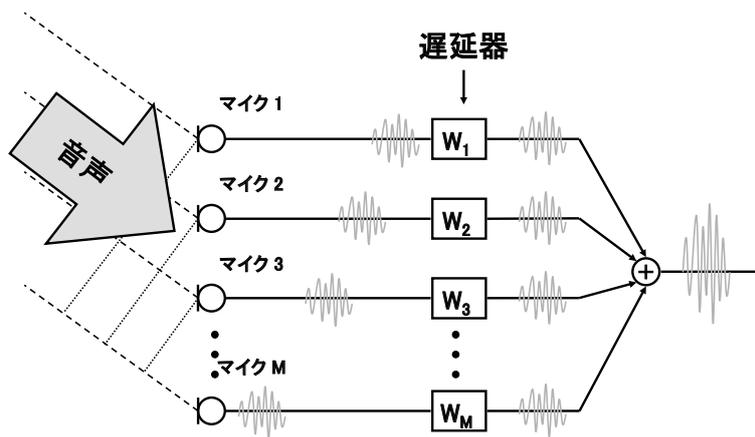


図 3.5 遅延和ビームフォーミング

量は、図 3.6 に示すように、音がマイクに到達するまでの時間と音がサブアレイ原点に到達するまでの時間の差であり、音源およびサブアレイの位置とマイク間の距離から容易に計算できる。計算された値にしたがって音声データを整列、加算することによってサブアレイ原点で集音される音を合成する。

次に、遅延和をおこなった音声データを隣接するサブアレイに送信する。このとき、録音時刻からグローバル遅延量を減算した値をタイムスタンプとしてデータに付加する。グローバル遅延量とは音源から各サブアレイ原点に音が到達するまでの時間であり、音源とサブアレイの座標から計算できる。音声データを受信した側では、受信したタイムスタンプを基に、グローバル整列をおこなって音声データを加算する。これに

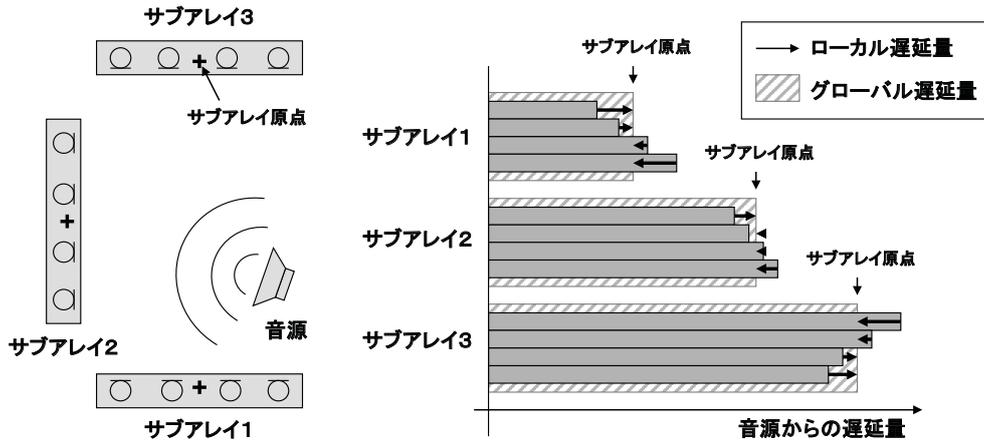


図 3.6 ネットワーク分散処理による遅延和ビームフォーミング

よって音源位置で集音される音を合成することができる。

### 3.4.3 データ集約

提案アルゴリズムでは、各サブアレイは 16ch の音声データから遅延和を計算することによって 1ch の音声データを得る。さらに、他のサブアレイから受信したデータも遅延和を計算することで、すべての子ノードと自分自身の音声データを 1ch に集約してからベースステーション方向へ転送する。

図 3.7 のような直線トポロジのネットワークを考えた場合、データ集約をおこなうことでネットワークに要求される帯域を大きく削減できる（もしサブアレイ間のデータ集約をおこなわなかった場合、Node 8 からの送信には 8 倍の帯域が必要となる）。また、実際にはサブアレイ間のネットワークはツリー型やメッシュ型のトポロジで構築すると考えられるため、データ集約はさらに効果的である。

## 3.5 サブアレイ間の時刻同期

前節で述べたネットワーク分散処理による遅延和計算アルゴリズムは、すべてのサブアレイが同一の時刻を持つことを前提としている。しかし現実には、各サブアレイは異なる水晶発振器を持ち、異なる時刻を有している。このため遅延和ビームフォーミングにおいては、サブアレイ間の時刻同期が非常に重要な要素となる。

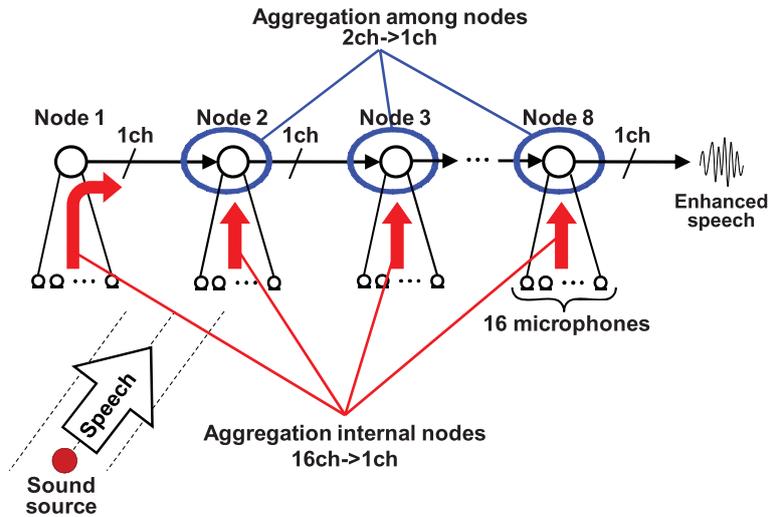


図 3.7 データ集約

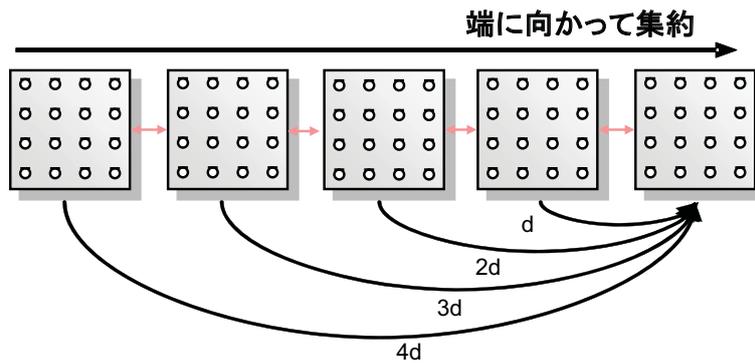


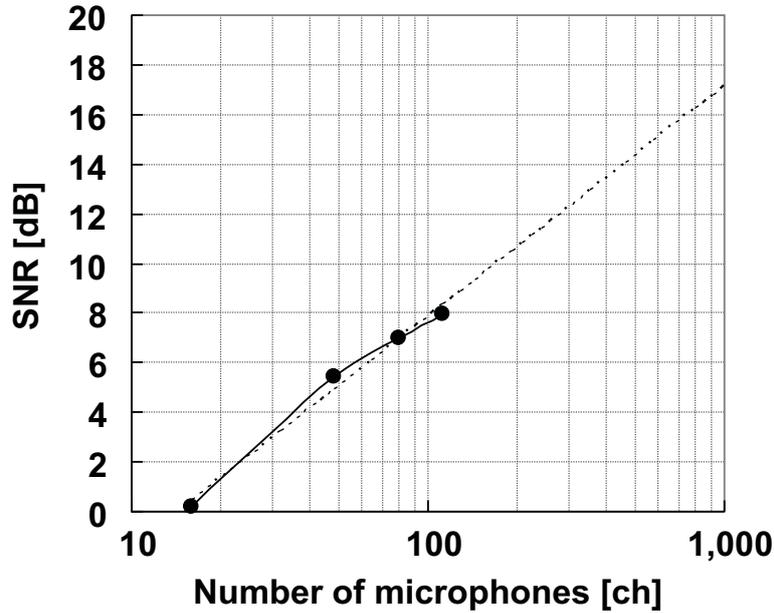
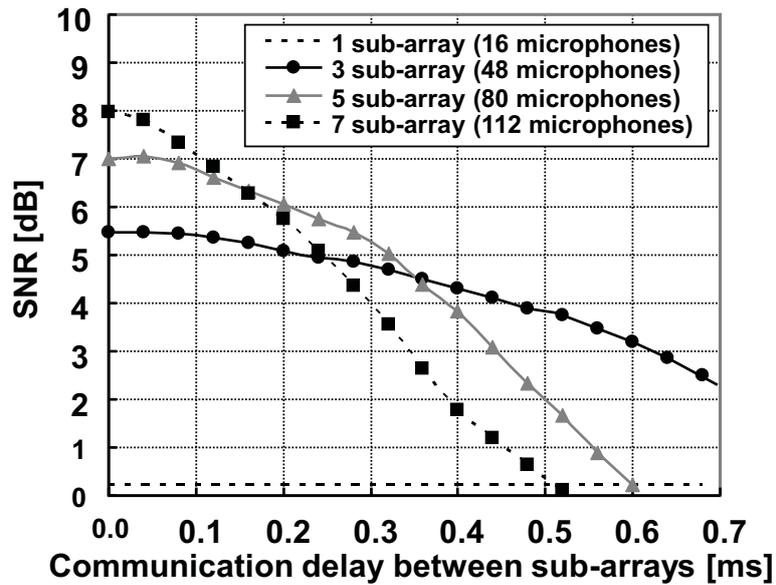
図 3.8 ノード配置と時刻誤差

### 3.5.1 時刻ずれによる影響の基礎検討

時刻ずれが遅延和ビームフォーミングの SNR に与える影響を検討するため、シミュレーションによる評価をおこなった。

シミュレーションでは、16ch サブアレイを 1 次元アレイ状に配置し、中央のサブアレイの正面 1.5m 地点に置いたスピーカーから音声信号を再生する環境を設定した。このとき、残響は 0.1s と仮定した。また、図 3.8 のように音声データは正面から見て右端のノードへ集約するものとし、サブアレイ間の時刻ずれ  $d$  が 1 ホップごとに加算されていく Worst Case を想定した。

まず、サブアレイ間の時刻ずれ  $d$  が 0 となる理想的な条件下で、サブアレイ枚数が増加した場合の SNR の変化を評価した (図 3.9)。次に、時刻ずれ  $d$  を変化させた場合の

図 3.9 時刻ずれ  $d$  が 0 の場合にサブアレイ枚数と SNR の関係図 3.10 時刻ずれ  $d$  と SNR の関係

SNR を図 3.10 に示す。結果から、ノード数が多いほど SNR は改善されるが、時刻ずれによる SNR 劣化が大きくなることがわかる。サブアレイが 7 枚の条件下で SNR の劣化を 1dB 以内に抑えるためには、時刻ずれ  $d$  を  $100\mu\text{s}$  以下にする必要がある。これはワーストケースの結果であるが、設計時の指針となる値であると考え、本研究では時刻同期精度の条件を  $100\mu\text{s}$  に設定する。

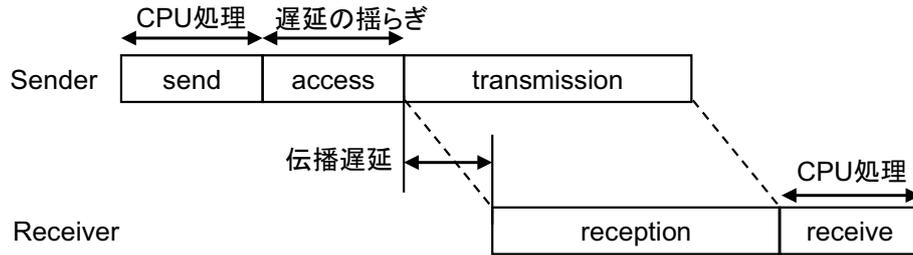


図 3.11 時刻同期の誤差要因

### 3.5.2 センサネットワークにおける時刻同期の課題

センサネットワークで時刻同期をおこなう場合、一般に以下に示す3つの誤差要因が存在する。

1つ目は伝搬遅延である。時刻同期は時刻情報を送信器から送信し、受信器においてその時刻情報を元に時刻合わせをおこなうことで実現される。伝搬遅延は、無線のマルチパスやインターネットのルータにおける待ち時間によって変化する。このため、受信した時刻情報に固定値を増減するだけでは、一般に正しい時刻情報を得ることができない。

2つ目は、CPU処理によって生じる誤差である。これはノードでの処理を低速なCPUで実行している場合に発生する。CPUが数MHz程度の動作周波数の場合、他のタスクの影響によっては、msオーダーで不定の遅延が生じる。

3つ目は、Media Access Control (MAC)層による遅延の揺らぎである。GPSや電波時計といった電波を固定的に割り当てられたインフラと異なり、Carrier Sense Multiple Access (CSMA)型のMACプロトコルを利用している場合、通信開始までの待ち時間は他のノード間の通信トラフィックに依存する。

以上の3要因の影響を図3.11に示す。sendとreceiveはCPU処理であり、アプリケーション層とMAC層の間でデータを渡すときに生じる遅延である。この遅延はオペレーティングシステムやタスクの負荷などに左右される。accessはMAC層で、メディアアクセスするための時間である。計算処理によって生じるCPU処理の誤差やMAC層によって生じる遅延の揺らぎに対して、伝搬遅延による誤差は非常に小さい。センサネットワークで1ホップが最大、数100mと仮定しても、伝搬遅延は高々 $1\mu\text{s}$ 程度である。このような観点から、センサネットワークにおける時刻同期では、CPU処理と

MAC 層における遅延の揺らぎによって生じる誤差をいかにして削減するかが重要な課題となる。

### 3.5.3 既存の時刻同期技術

従来のセンサネットワークでもノード間の時間的整合性が求められており、人工衛星を利用した Global Positioning System (GPS)、長波帯標準電波（電波時計）などの外部信号方式や、センサネットワークのためのパケット交換方式を用いた同期プロトコルとして、Reference Broadcast Synchronization (RBS)[9]、Timing-sync Protocol for Sensor Networks (TPSN)[10]、Flooding Time Synchronization Protocol (FTSP)[11] などが提案されてきた。本節では上記の技術について概説する。

#### GPS

Global Positioning System (GPS) は人工衛星を使った測位技術であり、元々は軍事技術として誕生した。現在では、カーナビや携帯電話でも利用されるようになり、われわれの生活に欠かすことのできない技術となっている。現在、地球の周りを 24~28 個の GPS 衛星が周回しながら、定期的に測位信号を発信している。受信機は、GPS 衛星の発信する測位信号を受け取ることで各衛星の距離を測定し、3 点（実際には 4 点）測位法によって受信機の位置を算出する。センサネットワークにおいて、GPS は測位技術だけではなく、時刻同期にも利用される。

GPS 衛星には原子時計が搭載されて、1ps の精度で同期が取れている。受信機においても 10ns 以下の精度で、正確な時刻を導出することが可能である。ただし、測位と時刻取得のためには最低でも 4 機の GPS 衛星が必要であるため、屋内などの電波状況の悪い環境では使用することができない。また、高性能受信機が必要なことと計算量が大きなことから、0.1W 以上の電力と \$50 程度のコストが必要である。

#### 長波帯標準電波

時計などの自動時刻補正に用いられる長波帯標準電波は、米国、ヨーロッパ、日本で標準時の送信がおこなわれており、簡単な仕組みで時刻同期を実現できる。日本国内では独立行政法人情報通信研究機構が福島県大鷹鳥谷および佐賀県と福岡県の県境の羽金山において送信局を運用している。この電波を受信して時刻合わせをおこなうこと

によって、時刻のずれを自動的に補正することが可能である。変調方式として 1bps の ASK を採用しており、羽金山の送信局においては 60kHz 帯、大鷹鳥谷の送信局においては 40 kHz 帯で全国に放送されている (図 3.12)。また、AM 波が受信できる環境ならば室内であっても利用できるため、GPS による時刻補正に比べて使用環境への制約は小さい。しかし、アンテナの大きさが最低でも腕時計程度となるのが難点である。

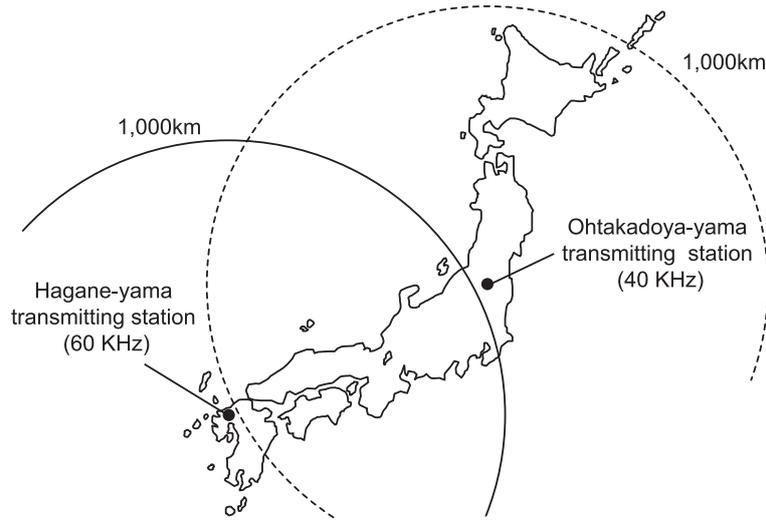


図 3.12 日本における電波時計基地局

タイムコードは西暦から分までの情報が含まれており、AM 変調で 1 秒 1bit (1bps) の速度で送信されている。1つのタイムコードは 1 分かけて送信され、24 時間絶えず送信が続けられている。通常は補正の確度を高めるために 2, 3 個のタイムコードを受信して現在時刻を確定させる。したがって通常 1 回の時刻取得に 2, 3 分を必要とする。

これによって実現される基地局時間に対する相対誤差は  $1.5\mu\text{s}$  程度である。小型化・低消費電力化を狙った、消費電力が  $0.1\text{mW}$  程度の時計用ワンチップ LSI (ML6191-A03; OKI [12]) も開発されており、GPS よりも低電力での時刻同期を実現できる。

また、電波時計と同じ概念を、空間内のローカルな無線環境で (法令が許すならば) 実装することも可能である。同様に、Power Line Communications (PLC) のようにワイヤレス電源に時刻信号を重畳し、ブロードキャストするアイデアも考えられる。

### Reference Broadcast Synchronization (RBS)

Reference Broadcast Synchronization (RBS) [9] は自律的に動作する同期手法であり、GPS や電波時計のように協定世界時 (Coordinated Universal Time: UTC) に時刻を合わ

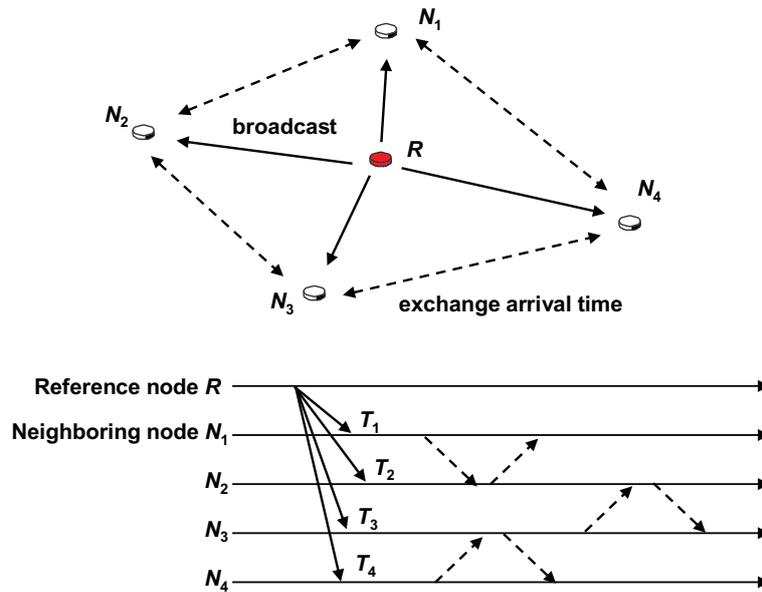


図 3.13 Reference Broadcast Synchronization (RBS)

せるといった用途ではなく、近傍のノードと同期を取る際に有効な手法である。RBS は、ある任意のブロードキャストを受け取った 2 つのノードで、受け取ったローカル時刻を記録し、ブロードキャストを受け取ったノード同士で時刻情報を交換する。図 3.13 に RBS の動作例を示す。

まず、基準ノード  $R$  が同期パケットをブロードキャストする。同期パケットを受け取った各受信ノード内で受信したローカル時刻を記録する。そして各受信ノードは、受信ノード同士で受信した時刻の交換をおこなう。これにより、 $n$  を近傍ノードの数、 $T_0$  を自ノードが同期パケットを受信した時刻、 $T_i$  を近傍ノード  $i$  が同期パケットを受け取った時刻とすると、 $\theta = \frac{1}{n} \sum_{i=1}^n (T_0 - T_i)$  によって時刻の補正值  $\theta$  を求めることができる。 $\theta$  は誤差の平均でもある。このように、受信ノード間で情報を交換することで同期を実現する手法を「receiver-receiver synchronization」と呼ぶ。上式において、ノード間の同期パケットを受け取った時刻のずれがガウシアン分布になると考えられることから、近傍ノード数  $n$  が大きければ大きいほど時刻精度が向上する。しかしながら、受信ノードの数  $n$  が大きくなればなるほど受信ノード間で交換する情報の量が増えるため、消費電力が増すという欠点がある。また、時刻同期はブロードキャストができる範囲に限られ、マルチホップネットワークには不向きである。

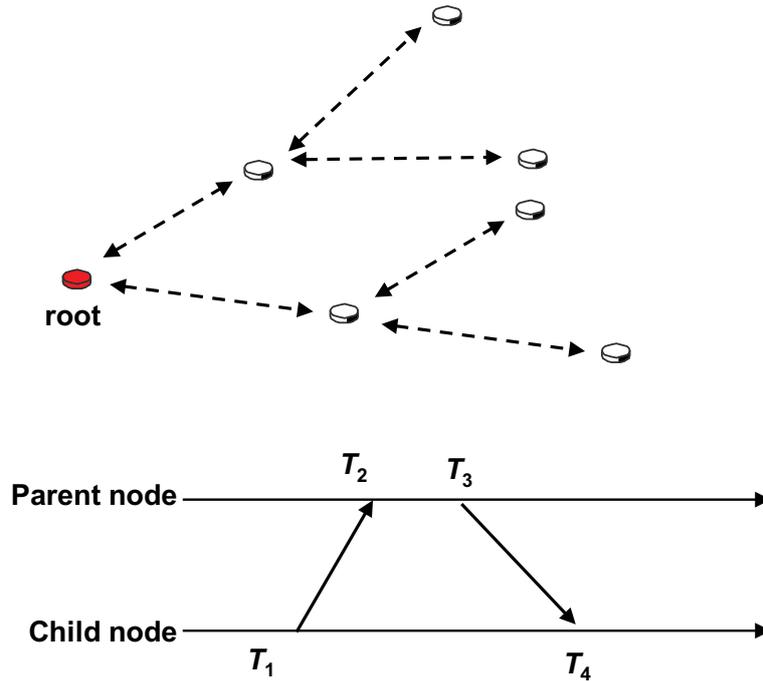


図 3.14 Timing-sync Protocol for Sensor Networks (TPSN)

### Timing-sync Protocol for Sensor Networks (TPSN)

Timing-sync Protocol for Sensor Networks (TPSN) [10] は、インターネットにおいて利用される Network Timing Protocol (NTP) と同様に、2-way handshake によって時刻同期を実現している。RBS が receiver-receiver synchronization だったのに対し、TPSN のような受信ノードが送信ノードに時刻同期する手法を「sender-receiver synchronization」と呼ぶ。TPSN ではまず、同期する際に基準となるノードを root として全域木を構築する。次にその構築した全域木の親ノードと子ノードで 2-way handshake によるタイムスタンプを利用して時刻同期をおこなう。TPSN では子ノードから親ノードに向けて同期を開始する (図 3.14)。2-way handshake をおこなう際、 $T_1 \sim T_4$  の 4 種類のタイムスタンプを利用する。

ここで、 $a = T_2 - T_1$ 、 $b = T_4 - T_3$  とすると、子ノードの親ノードに対する時刻ドリフトは  $\sigma = (a - b)/2$  で求まる。また、子ノードと親ノードの間の遅延は  $\theta = (a + b)/2$  である。

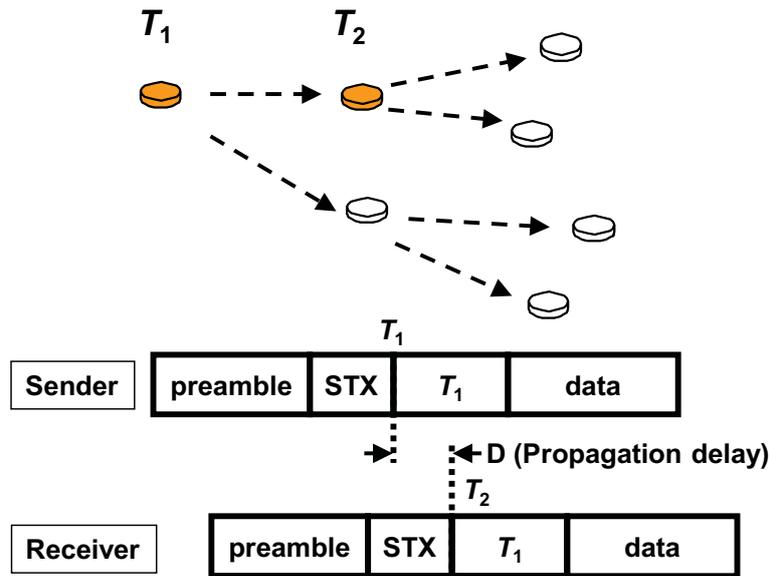


図 3.15 Flooding time synchronization protocol (FTSP)

### Flooding Time Synchronization Protocol (FTSP)

Flooding Time Synchronization Protocol (FTSP) [11] では、一方向の簡略な通信のみによって高精度の同期を実現する。FTSP は送信側と受信側の MAC 層でタイムスタンプを取ることで sender-receiver synchronization を実現する。さらに、FTSP は一方向の送信だけで同期が可能であり、同期パケットのフラッディングによるネットワーク全体の同期も実現できる。図 3.15 に FTSP の動作例を示す。

送信側は、データ送信開始点を示す制御コード (STX) を送信した直後にタイムスタンプ (タイマの値)  $T_1$  を取得し、 $T_1$  をデータとして同期パケットを送信する。受信側では STX を受信した直後にタイムスタンプ  $T_2$  を取得する。これによってタイムスタンプ取得までの遅延を最小化し、同期時の誤差を減らすことができる。送信ノードのタイムスタンプを  $T_1$ 、受信ノードのタイムスタンプを  $T_1$ 、伝播遅延を  $D$  とすると、時刻の補正値は  $\theta = T_1 - T_2 - D$  として求められる。ここで  $D$  は送受信回路の仕様によって決まる固定値である。

同期パケットのデータ部  $T_1$  受信直後に  $\theta$  を用いてタイマの値を更新することで、受信側が送信側の時刻に同期される。また、受信側は続けて同期パケットを作成し、同じ手順で送信する。これがネットワーク全体で繰り返されることで、すべてのノードの時刻が同期される。

### 3.5.4 提案システムにおける時刻同期の実装

前節で述べた各時刻同期技術を表 3.1 にまとめる。提案システムでは、時刻同期精度と低消費電力性能から、FTSP を用いたネットワーク上のパケット交換によってサブアレイ間の時刻同期をおこなう。一定時間ごとにベースステーションから時刻同期パケットをサブアレイにブロードキャストし、各サブアレイは自身のタイマ値を更新していくことで、ネットワーク全体をベースステーションの時刻に同期する。また、FTSP におけるタイムスタンプの付加・取得の操作を MAC 層の処理と合わせて専用ハードウェアで実装する。これにより、リアルタイムクロックで動作するタイマ・カウンタとハードウェアレベルで協調させ、CPU 処理による誤差を回避することができる。

表 3.1 時刻同期技術の比較

		受信機の増加	ノード間通信	時刻精度	低電力性	特徴
外部信号方式	GPS	○	不要	○	×	屋外のみ
	電波時計	○	不要	○	○	アンテナ大
パケット交換方式	RBS	×	双方向	△	×	マルチホップに不向き
	TPSN	×	双方向	○	△	CPU必要
	FTSP	×	単方向	○	○	フラッシングと併用可

FTSP による同期精度は隣接ノード間で 1us 以下であり、音声データを扱ううえで十分な精度が得られる。ただし、各サブアレイはそれぞれが水晶発振器によってクロックを生成しているため、温度変化や製造ばらつきなどによってサブアレイ間で発振周波数のずれが生じる。このため、サブアレイごとにタイマの進む速度が異なるという問題があり、同期完了直後の時刻ずれが 1us であったとしても、1 秒間で数 us から数十 us の誤差を生じてしまう。そこで提案システムでは、FTSP による時刻同期時にサブアレイ間の時刻ずれを記憶し、線形補間によってタイマの進み方を調節する (図 3.16)。1 回目の同期時の受信タイムスタンプを  $T'_1$ 、2 回目の同期時のタイムスタンプを  $T_1$ 、受信側のタイマ値を  $T_2$  とすると、 $\Delta T = T_1 - T'_1$  の期間に  $\theta = T_1 - T_2 - D$  だけタイマの進み方を調整することで、周波数ずれを補間することができる。これにより、同期完了後の時刻ずれを 1 分間で  $10\mu\text{s}$  以内に抑えることができる。

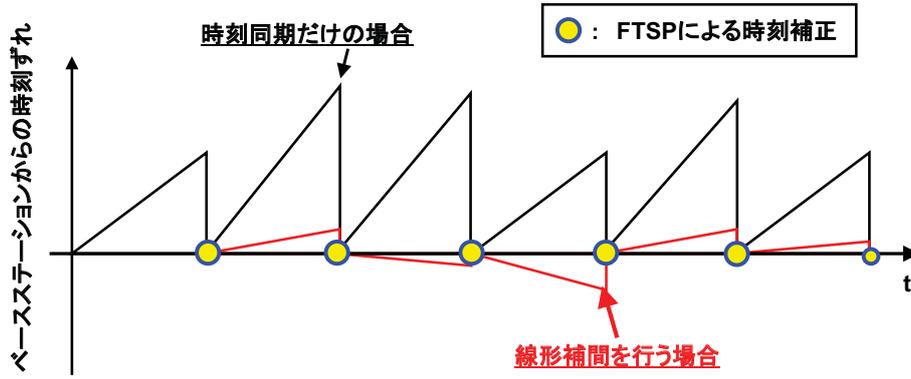


図 3.16 周波数ずれの線形補間

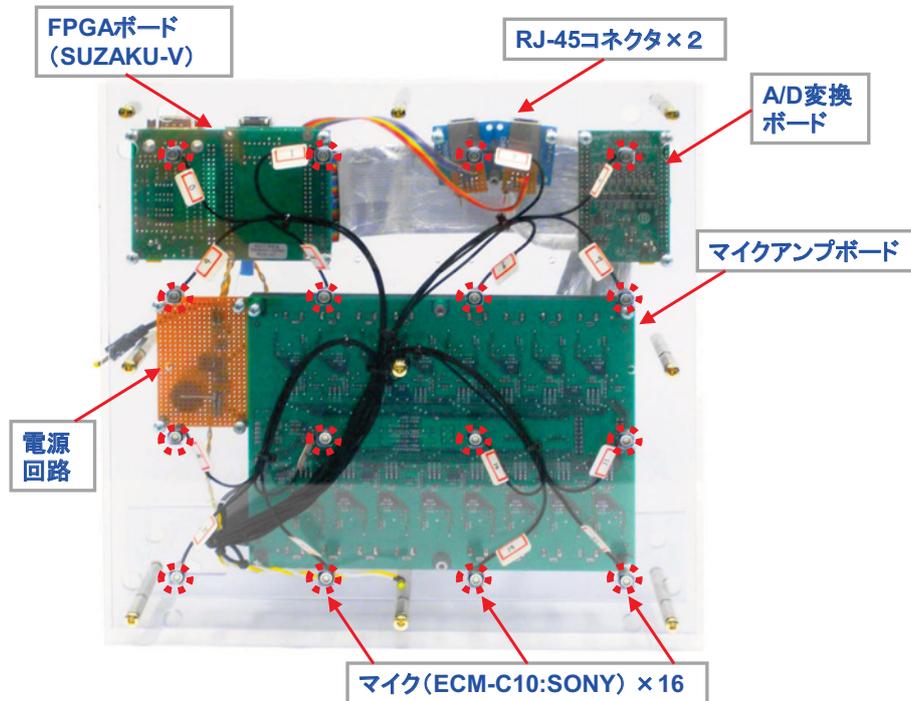


図 3.17 サブアレイのプロトタイプ実装

### 3.6 FPGA を用いたプロトタイプ実装と実証実験

ネットワーク型マイクアレイのプロトタイプとして、発話推定と音源定位、音源分離をおこなうシステムを FPGA に実装した。また、3つのサブアレイを用いて性能評価実験をおこなった。

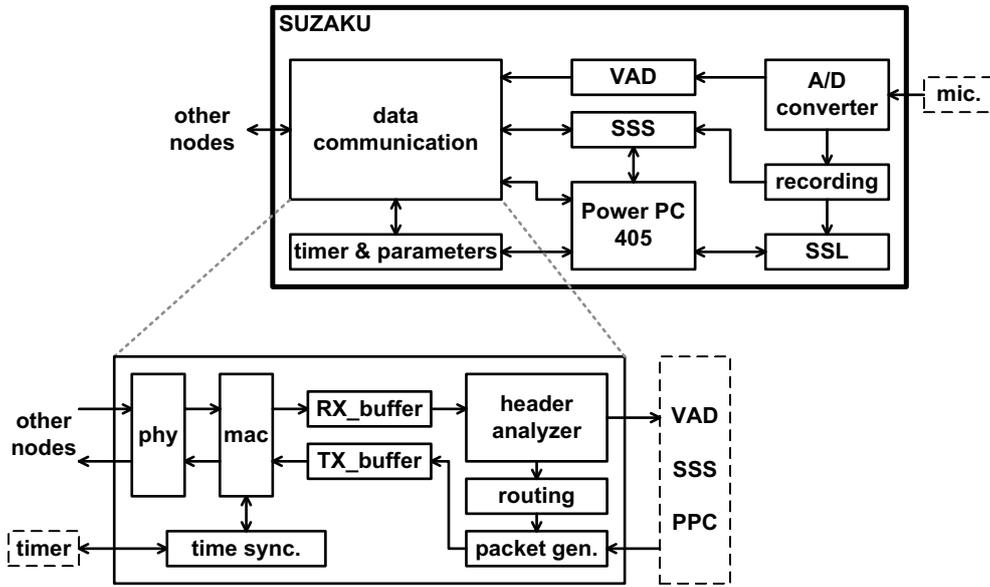


図 3.18 サブアレイのブロック図

### 3.6.1 FPGA 実装

図 3.17 に試作したサブアレイを示す。前面には 16 個のマイクが 7.5 cm 間隔で格子状に配置され、発話検知を表示する LED が設置されている。背面にはマイクアンプと A/D 変換ボード、モジュールを実装する FPGA ボード、通信用 LAN ポートなどが設置されている。サブアレイの寸法は、壁や天井に設置することを考慮し、建材のタイルと同じ 30 cm × 30 cm とした。音声の録音は、人の声 (30~8000Hz) の集音を目的としているため、サンプリング周波数 16kHz でおこなう。モジュールの実装にはアットマークテクノ社製の FPGA ボード SUZAKU[13] を用いる。SUZAKU にはハードプロセッサ (PowerPC : PPC) が搭載されおり、S/W および H/W の両方でモジュールの実装をすることができる。サブアレイ間は UTP ケーブルで接続され、物理層は 10BASE-T のイーサネットを用いる。データリンク層には低消費電力化のために Low Power Listening[14][15] を使用し、ネットワーク層における経路構築には Tiny Diffusion[16] を用いる。

図 3.18 にサブアレイのブロック図を示す。発話推定モジュール (VAD) は専用回路として FPGA 上に実装する。音源定位モジュール (SSL) は FFT の処理を H/W で実装し、他の複雑な行列演算などは PPC 上で実行する。音源分離モジュール (SSS) は遅延

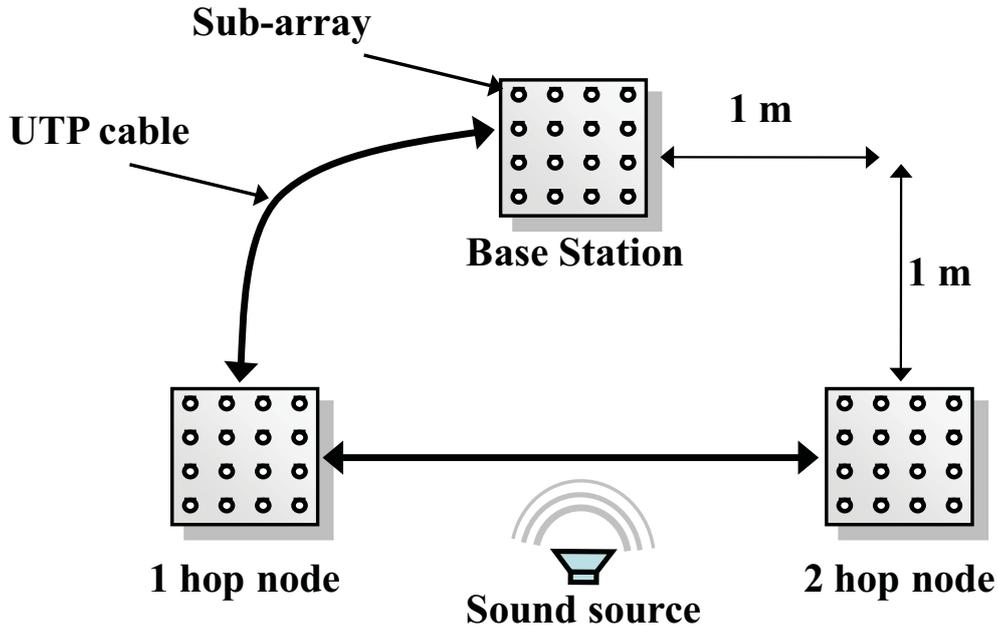


図 3.19 発話推定および音源定位実験におけるサブアレイ配置と環境の設定

計算を PPC でおこない、音声の加算は H/W でおこなう。通信制御モジュールは物理層とデータリンク層のプロトコルを専用回路で実装している。また、時刻同期モジュールも同期精度向上のため専用回路で実装している。

### 3.6.2 性能評価実験

3つのサブアレイを用いて時刻同期、発話推定、音源定位、音源分離の実験をおこなった。

まず、発話推定モジュールと音源定位モジュールの動作検証をおこなった。サブアレイは図 3.19 のように配置し、中央のサブアレイをベースステーションとしてサーバ (PC) に接続した。ただし、マルチホップの性能を評価するため、ネットワークトポロジはベースステーションから 2 ホップの直線トポロジとした。図 3.20 は発話推定の実験結果である。サブアレイの正面 1 m の地点にスピーカを設置したベースステーションの VAD の出力を観測した。すべての発話区間で VAD が反応し、発話から VAD 反応までの遅延は平均 30 ms であった。また、図 3.21 は音源定位の動作例である。音源定位結果の更新に必要な時間は約 0.3 秒であった。

次に、FTSP による時刻同期の実験をおこなった (図 3.22)。同期直後にはすべてのサブアレイが  $1\mu\text{s}$  以下の精度で同期できることを確認した。また、サブアレイ間の周波数

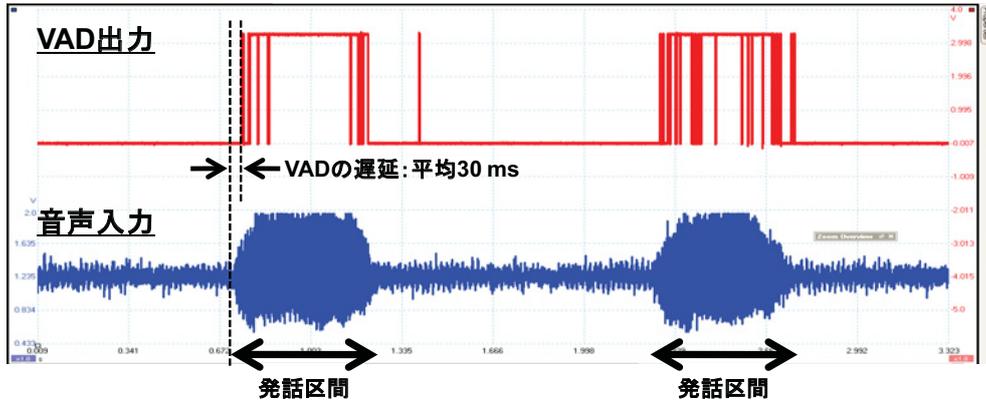


図 3.20 発話推定実験

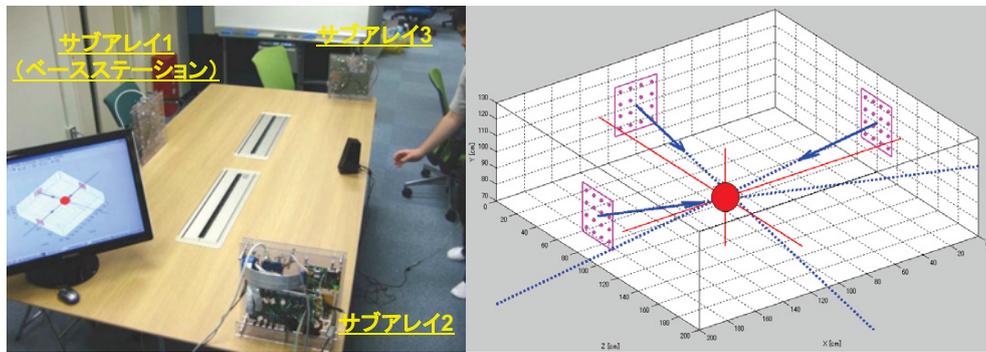


図 3.21 音源定位実験

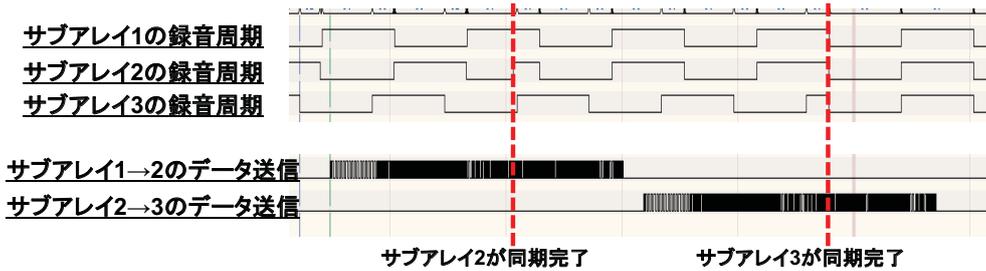


図 3.22 FTSP による時刻同期の動作波形

ずれによる誤差は最大で毎秒  $15\mu\text{s}$  であった。2 回目の同期からは線形補間によって周波数ずれが補正され、同期から 1 分後の最大誤差は  $10\mu\text{s}$  以下に削減された。

最後に、図 3.23 のようにサブアレイを配置し、500 Hz 正弦波の音源と、300 Hz, 700 Hz, 1300 Hz 正弦波の雑音源を用いて遅延和ビームフォーミングによる音源分離実験をおこなった。サブアレイの接続関係は図 3.19 と同じである。

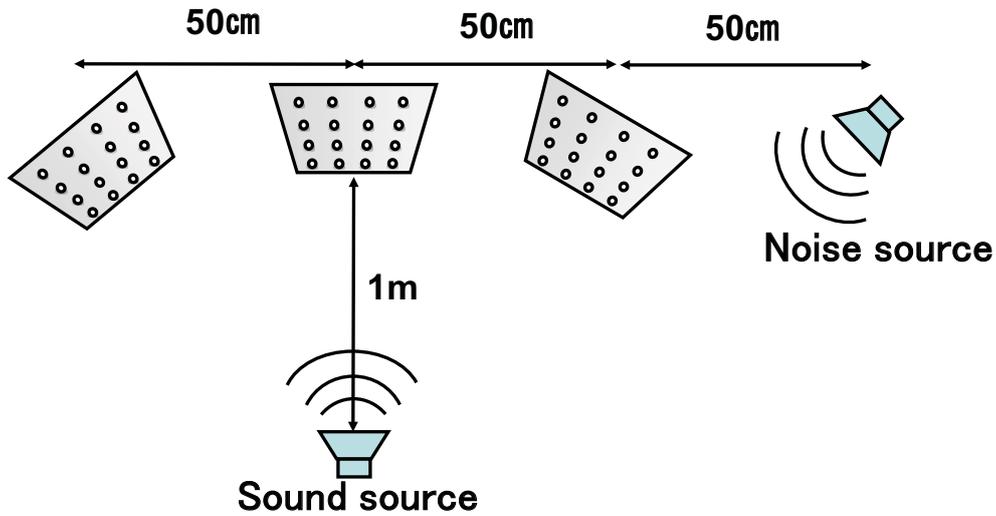


図 3.23 遅延和ビームフォーミング動作実験の環境設定

実験により，サブアレイ 3 つによる音源分離はリアルタイムで動作することを確認した．図 3.24 図 3.25 は 1 ch, 16 ch (サブアレイ 1 つ), 48 ch (サブアレイ 3 つ) の場合の音源分離結果を示している．マイク数の増加に伴って SNR が改善されていることがわかる．48 ch でも 500 Hz の信号に減衰は見られないのに対して，300 Hz と 1300 Hz の信号は 48ch で約 20dB まで低下した．一方，700 Hz の信号は 48 ch でも数 dB しか低下していなかった．シミュレーションによる検証をおこなった結果，これはサブアレイの配置によって特定の周波数（今回の場合 700 Hz）で側面に干渉縞が発生し，指向性が低下することが判明した（図 3.26）．これはサブアレイ数を増やすことで改善できる問題であり，さらなる大規模化が必要であることを示している．

### 3.7 結言

本研究では，従来の大規模マイクアレイにおける演算量や消費電力の問題を解決する技術として，センサネットワーク技術を用いたネットワーク型マイクアレイシステムを提案した．提案システムでは，小規模なマイクアレイを多数配置し，それらの間で有線ネットワークを構築して協調動作をおこなうことで，大規模なマイクアレイ音声処理システムを実現する．また本研究では，ネットワークを介した分散処理に適した音源分離のアルゴリズムを新たに提案した．FPGA ボード SUZAKU を用いて 16ch サブアレイの試作をおこない，3 つのサブアレイを用いた実証実験によって提案システムの動作を確認した．

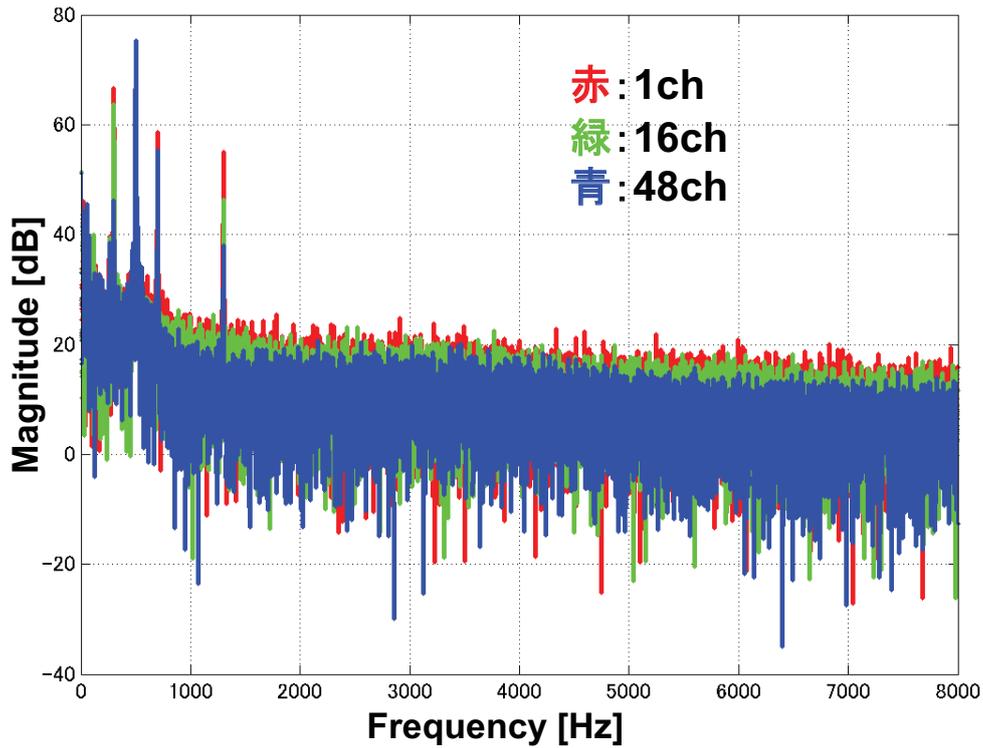


図 3.24 遅延和ビームフォーミングによって得られた信号の周波数スペクトラム

従来のデータ収集のみを目的としたアプリケーションとは異なり、ネットワーク型マイクアレイシステムには「リアルタイム処理」、「センサノード間の時刻同期」、「ネットワーク分散処理」という3つの機能が必要であった。これらの機能は、他のより実用的なアプリケーションでも必須となるものであり、本研究はセンサネットワークの応用分野を拡大するためのテストベッドとしての役割も果たすものである。

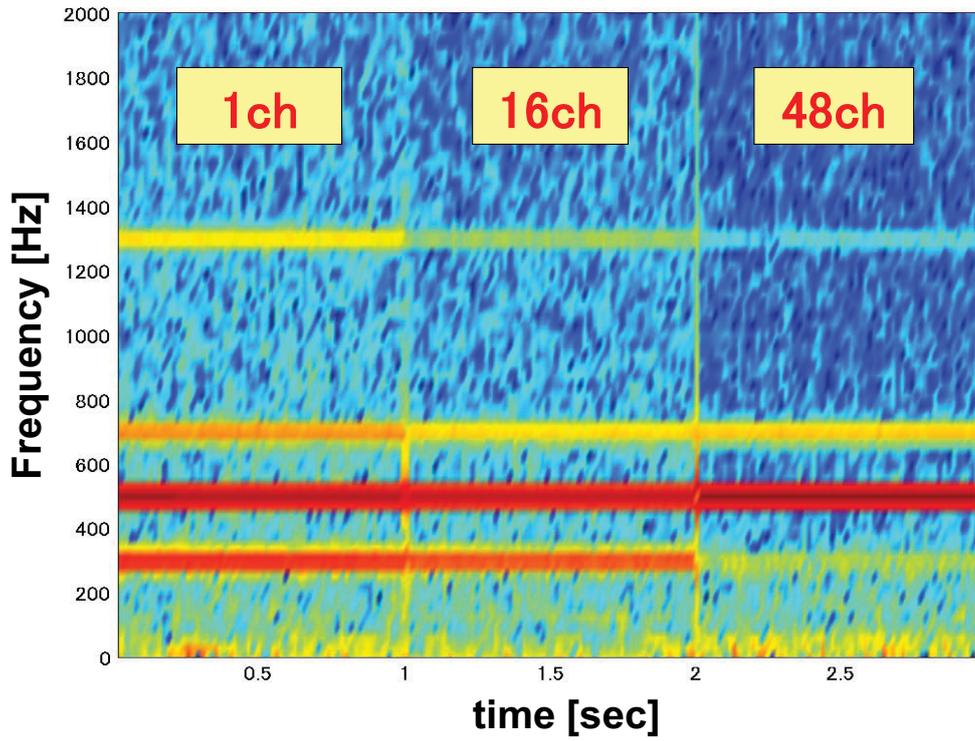


図 3.25 遅延和ビームフォーミングによって得られた信号の周波数スペクトラム（時間軸で見た変化）

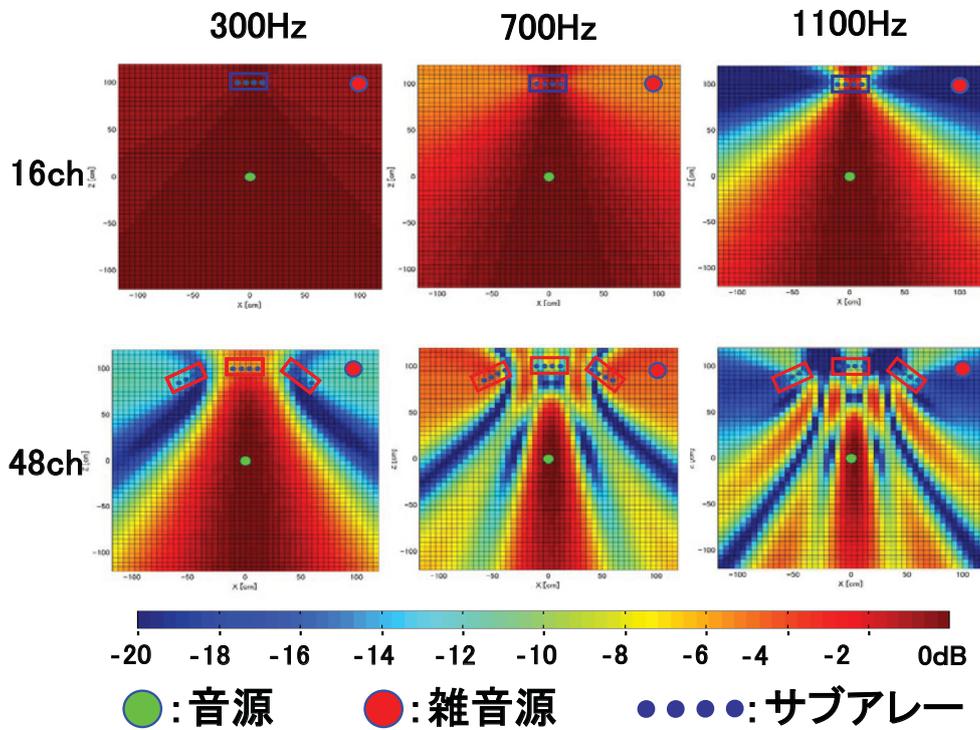


図 3.26 サブアレイ配置と指向性の関係



## 第4章

# 待ち時間制御によるブロードキャストアルゴリズムの高効率化

### 4.1 緒言

本章ではネットワーク層での経路制御において必要不可欠であるブロードキャストの高効率化について述べる。

ワイヤレスセンサネットワークは膨大な数のノードによって構成されるため、従来のネットワークシステムのように固有の ID やアドレスを管理することが困難である。そのためワイヤレスセンサネットワークでは、パケットのペイロード（ヘッダ以外の通信データ）の内容に基づいて経路をホップ・バイ・ホップで決定する、データセントリックな経路制御が考えられている。多くのデータセントリックな経路制御において、経路を構築するためには、情報を入手したいノード（sink）が interest パケットとよばれるメッセージをブロードキャスト（ネットワーク内のすべてのノードに向けて送信）する必要がある。従来は、すべてのノードがメッセージを再送信することでブロードキャストをおこなう方法（フラッディング）が用いられてきた。しかしフラッディングには、冗長な通信が多発するという問題 [17] があり、この問題を改善するためにさまざまな手法が提案されている [18]。

本研究では、その中でもカウンタベース方式 [17] のブロードキャストアルゴリズムに着目し、待ち時間の制御によるカウンタベース方式の改良法を提案した。従来のカウンタベース方式では最初のメッセージ受信時に待ち時間が決定されるが、本方式では待ち時間をメッセージ受信ごとに再設定する。この待ち時間制御によって受信回数の少ないノードが再送信する確率が高くなり、1ノードあたりの面積カバー率が改善されるため、ネットワーク全体の再送信ノード数を削減できる。

## 4.2 ブロードキャストアルゴリズムの課題と関連研究

### 4.2.1 経路制御とブロードキャスト

インターネットなどの従来のネットワークでは、ID やアドレスによってノードを識別し、最終目的ノードの ID によって中間ノードが次ホップを決定する、アドレスセンタリックな経路制御が用いられてきた。しかし、ワイヤレスセンサネットワークは膨大な数のノードによって構成されるため、固有の ID やアドレスを管理することが困難である。また、重要なのはセンシングデータであり、ノードを識別することはあまり重要ではない。そのためワイヤレスセンサネットワークでは、パケットのペイロード（ヘッダ以外の通信データ）の内容に基づいて経路をホップ・バイ・ホップで決定する、データセンタリックな経路制御が考えられている [19][20]。

データセンタリックな経路制御方法として、Directed Diffusion[21] とよばれるプロトコルや、Directed Diffusion を簡易化した Tiny Diffusion[16] などが提案されている。これらのプロトコルでは、経路構築の初期段階で、情報入手したいノード (sink ノード) が interest パケットとよばれるメッセージをブロードキャスト (ネットワーク内のすべてのノードに向けて送信) する必要がある (図 4.1)。また、他にも経路構築にブロードキャストを必要とする経路制御アルゴリズムは数多く存在する。さらに、第3章で述べた時刻同期やアラームの送信、ノード発見など、経路制御以外にも多くの場面でブロードキャストが必要となるため、マルチホップの無線ネットワークを考えるうえでブロードキャストは非常に重要である。

### 4.2.2 ブロードキャストの問題点

従来のネットワークではブロードキャストをおこなうために、フラッディング [22] とよばれる方法が用いられてきた。フラッディングは、

1. 初めてメッセージを受信したノードは、そのメッセージを再送信する。
2. 既に受信したことのあるメッセージは無視する。

という非常にシンプルなアルゴリズムであり、各ノードは過去の受信メッセージを覚えておくだけでよい。そのため実装が容易であり、信頼性が高いという利点もある。しかし、以下のような問題も指摘されている [17]。

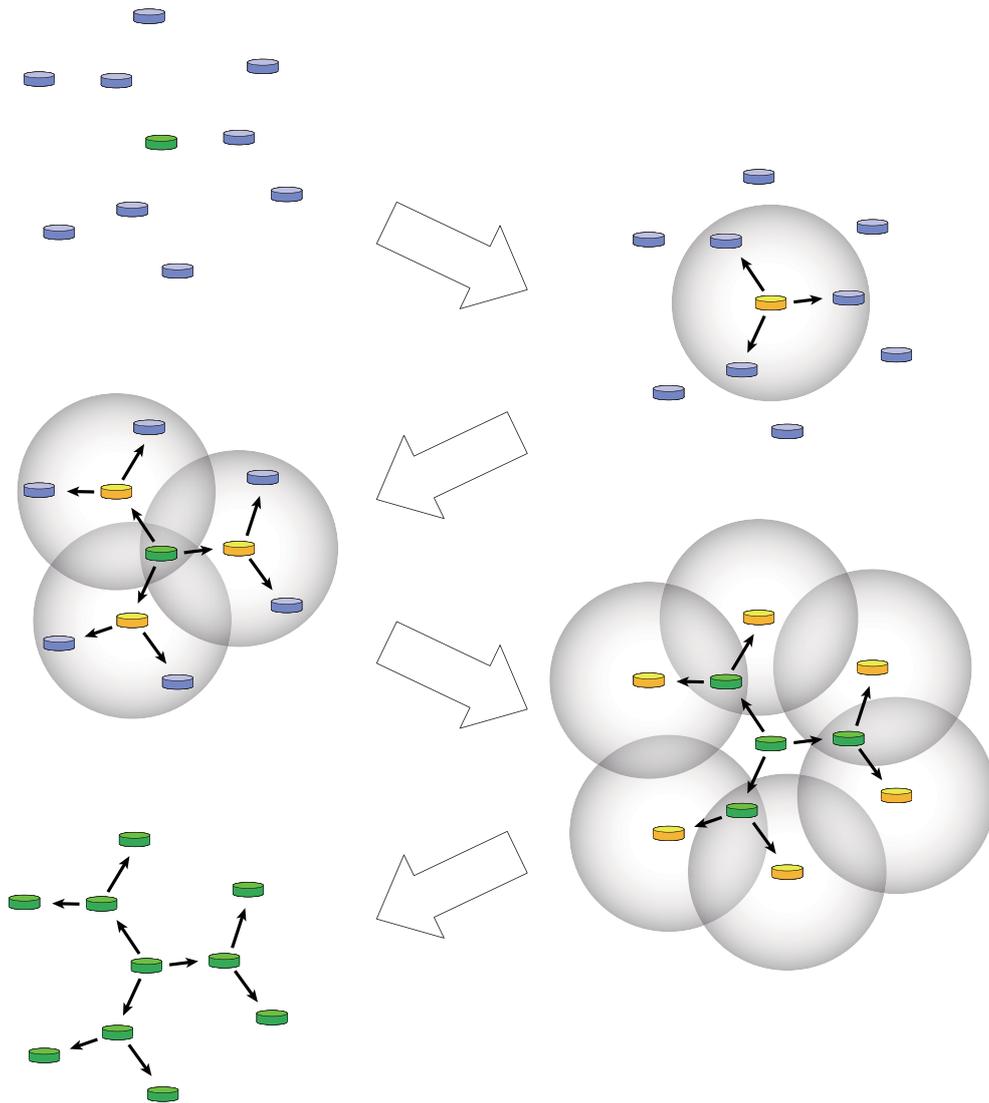


図 4.1 ブロードキャストによる経路構築

**冗長な通信** あるノードが再送信をおこなっても、隣接ノードがすでにそのにメッセージを受信していることが多い。

**競合** すべてのノードが再送信するため、隣接ノード数が多いと帯域の取りあいが起こる。

**衝突** ブロードキャストでは衝突回避や再送信をしないため、衝突が起こりやすく、しかもそれが致命的になる。

これらの問題は sink ノードの近傍で特に顕著であり、ネットワーク全体のエネルギー効率を著しく悪化させる (図 4.2)。特にワイヤレスセンサネットワークでは、センサー

ドのバッテリー容量と処理能力に限られるため、エネルギー効率の悪化によってネットワークシステム可用時間の減少というさらに重大な問題が発生する。

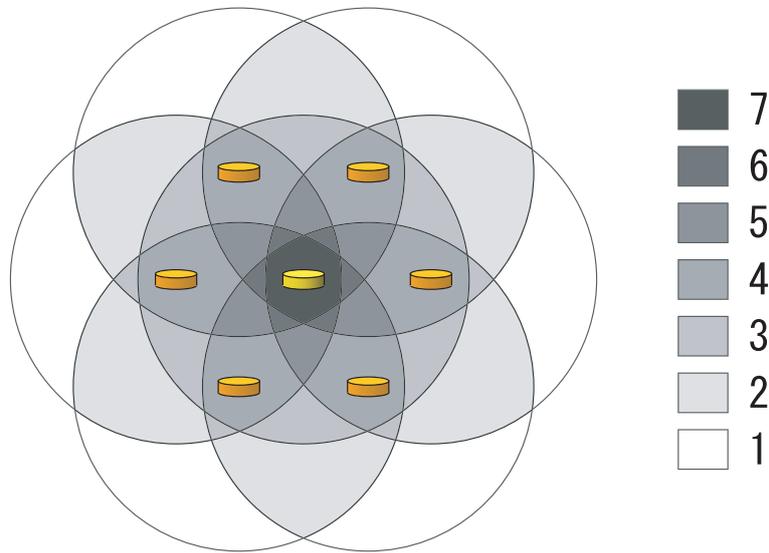


図 4.2 ブロードキャストストームの例

### 4.2.3 ブロードキャストの改良アルゴリズム

フラッディングにおける問題は、すべてのノードが再送信をおこなうことに起因する。この問題を改善するためには、信頼性（高いパケット到着率）を維持しつつ無駄な再送信を減らすことが重要であり、そのためにさまざまなアルゴリズムが提案されている [18]。本節では、これらのブロードキャストアルゴリズムについて述べる。ブロードキャストアルゴリズムは、確率的アルゴリズムと決定性アルゴリズムの2種類に大別できる。

#### 確率的アルゴリズム

ブロードキャストメッセージを再送信するか否かを決定するために乱数を用いるものを確率的アルゴリズムとよぶ。例えば再送信をある確率でもっておこなうものや、メッセージ受信後にランダムな待ち時間を設定し、待ち時間終了後に再送信の判断をおこなうものなどである。確率的アルゴリズムには、ブロードキャストメッセージ以外の通信によるオーバーヘッドが少ないという長所があるが、確率に基づいて再送信するため、すべてのノードに 100% メッセージが到達することを保証できないという短所もある（ただし、限りなく 100% に近づけることは可能である）。

以下、代表的な確率的アルゴリズムについて概説する。

**確率ベース** 初めてメッセージを受信したノードは、ある確率  $p$  でメッセージを再送信する。オリジナルの確率ベースアルゴリズム [17] では、最初に確率  $p$  を決めておき、すべてのノードがその確率で再送信をおこなう。しかし、確率を固定すると、ノードの分布が一様でない場合にノード密度の薄い場所で到着率が悪化するという問題が起こるため、各ノードが周辺のノード密度（隣接ノード数）から個別に確率を決定する改良アルゴリズムも提案されている [23]。

**カウンタベース** 初めてメッセージを受信したノードは、ランダムな待ち時間を設定する。待ち時間終了後、メッセージの受信回数がしきい値（あらかじめ決めておく）以下であればメッセージを再送信する [17]。

カウンタベースの改善アルゴリズムとして、ノードごとに隣接ノード数からしきい値を決めるもの [24] や、距離ベースと組み合わせたもの [25] が提案されている。カウンタベースについては、次節で詳細に述べる。

**距離ベース** 初めてメッセージを受信したノードは、そのメッセージを送信したノードとの距離がしきい値（あらかじめ決めておく）より遠ければ、ランダムな待ち時間を設定する。待ち時間終了後、待ち時間中に受信したメッセージの送信元との距離を調べる。すべての送信元までの距離がしきい値以上であればメッセージを再送信する [17]。

メッセージ送信元との距離情報を用いるため、信号強度測定のためのハードウェア（RSSI: Received Signal Strength Indicator）や、距離推定のためのアルゴリズムが必要となる。

**位置ベース** 初めてメッセージを受信したノードは、ランダムな待ち時間を設定する。待ち時間終了後、受信したメッセージの送信元の位置情報から、自分が再送信した場合に新たにカバーできる面積を計算し、それがしきい値以上であれば再送信する [17]。

ノードの位置情報を用いるため、GPS などのハードウェアや位置推定のためのアルゴリズムが必要であり、ブロードキャスト以外のコストが高い。また、ブロードキャストメッセージに送信元の位置情報を付加する必要がある。

## 決定性アルゴリズム

確率的アルゴリズムに対して、乱数を使わず、例えば隣接ノードの情報やネットワークのトポロジ情報を利用することで、再送信の抑制をおこなうものを決定性アルゴリズムという（フラッディングも決定性アルゴリズムに分類される）。決定性アルゴリズムは100%の到着率を保証できる。さらに、再送信ノード数自体は確率的アルゴリズムより抑えられるものが多い。しかし、各ノードが隣接ノードの情報を保持する必要があるため、ブロードキャストメッセージ以外の通信や、各ノードの情報処理によるオーバーヘッドが多いという問題がある。

以下、代表的な決定性アルゴリズムについて概説する。

**Self Pruning** 各ノードは隣接ノードリストを保持し、メッセージヘッダに自分の隣接ノードリストを付加して送信する。メッセージを受信したノードは、送信元と自分の隣接ノードリストを比較し、自分の隣接ノードにまだメッセージを受信していないノードがあれば再送信する [26].

**Scalable Broadcast Algorithm (SBA)** 各ノードは2ホップ先までの隣接ノード情報を保持する。メッセージを受信したノードは、以下の式によって待ち時間を設定する。

$$T \times \frac{d_{N_{\max}}}{d_{mc}}$$

ここで  $T$ :定数,  $d_{N_{\max}}$ :自分の隣接ノードのうちでもっとも隣接ノードが多いものの隣接ノード数,  $d_{mc}$ :自分の隣接ノード数とする。これは、隣接ノード数が多いノードから再送信をおこなうようにするためである。待ち時間終了後、受信したメッセージの送信元と2ホップ以内の隣接ノードリストを照合し、自分の隣接ノードにまだメッセージを受信していないものがあれば、再送信をおこなう [27].

**Dominant Pruning** 各ノードは2ホップ先までの隣接ノード情報を保持する。メッセージを送信するノードは、隣接ノードリストから再送信ノードを選択し、そのIDリストをメッセージヘッダに付加する。このとき、2ホップ先のノードをすべてカバーできるように再送信ノードを選択しなければならない。メッセージを受信したノードはヘッダを調べ、自分が再送信ノードに指定されていれば、再び自分の隣接ノードリストから再送信ノードを指定し、メッセージを再送信する。これ

を繰り返すことでブロードキャストをおこなう [26].

再送信ノードの選択において、2 ホップ先のノードをすべてカバーできるような最小のノード集合を求めることは NP 完全問題である。そのため Dominant Pruning では、Greedy Set Cover (GSC) とよばれるヒューリスティックなアルゴリズムによって再送信ノードを選択する。GSC ではまず、隣接ノードのうち、そのノードが再送信することによって新しくカバーできるノード数をもっとも多いノードを再送信ノードに選ぶ。残りの隣接ノードに対してもこれを繰り返し、2 ホップ先のノードをすべてカバーできるノード集合を得る。

### 4.3 カウンタベース方式ブロードキャストアルゴリズム

本節では、従来のカウンタベース方式について詳説する。

あるノードがブロードキャストメッセージを複数回受信したあとに再送信した場合、メッセージ受信回数と面積カバー率の関係は図 4.4 のようになる。ここで面積カバー率とは、あるノードが再送信した場合に、送信面積に対して新しくカバーできた面積の割合である。図 4.3 の例では、ノード A, B が送信したあとにノード X が送信した場合、ノード X の送信によって新しくカバーできた面積は  $S$  となる。各ノードの送信面積は  $\pi r^2$  なので、ノード X の面積カバー率は  $S/(\pi r^2)$  となる。図 4.4 から、ノードの受信回数が増えるとカバー率が急激に悪化することがわかる。

カウンタベース方式では図 4.4 の結果を利用し、メッセージ受信回数がしきい値  $C_{th}$  (あらかじめ決めておく) より少ないノードだけがメッセージを再送信することでブロードキャストをおこなう。

以下にカウンタベース方式アルゴリズムの詳細を示す。

1. 初めてブロードキャストメッセージを受信したノードは、カウンタを 1 で初期化し、 $0 \sim T_{max}$  の範囲でランダムな待ち時間 (RAD: random assessment delay) を設定する。
2. RAD の期間は同じブロードキャストメッセージを受信するごとに、カウンタを 1 ずつ増やす。
3. RAD 終了後、カウンタの値  $< C_{th}$  であればブロードキャストメッセージを再送信する。

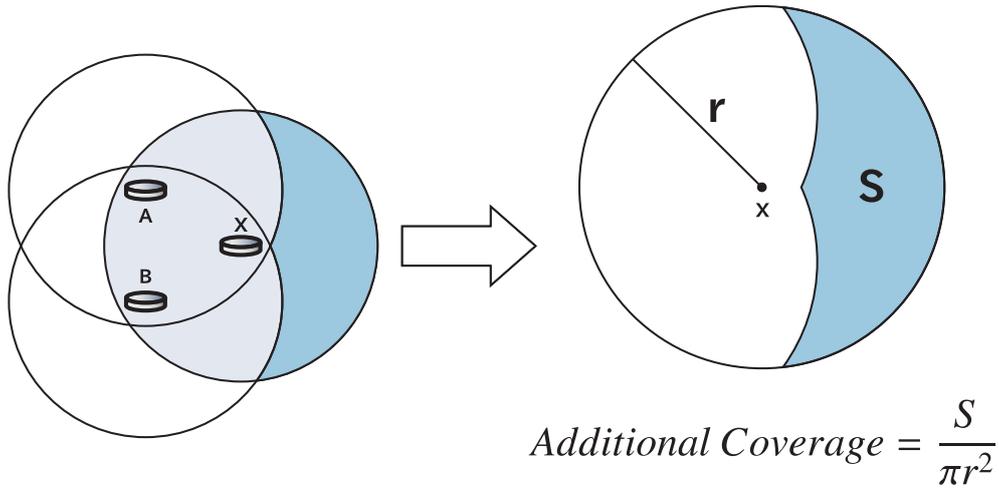


図4.3 面積カバー率 (EAC: Expected additional coverage)

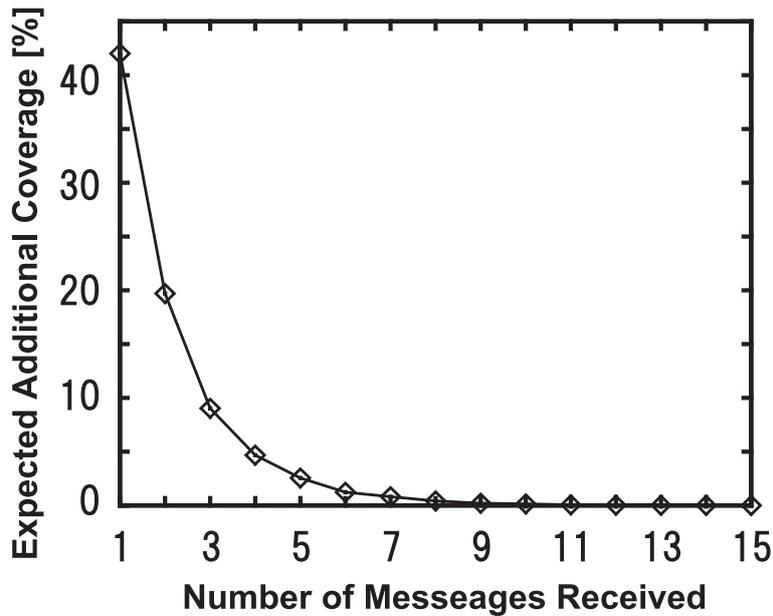


図4.4 メッセージの重複受信回数と面積カバー率の関係

ここでしきい値  $C_{th}$  と  $T_{max}$  はシステム設計段階で決定されるパラメータである。  $C_{th}$  はパケット到着率と再送信ノード数のトレードオフに影響を与える。カウンタベース方式は確率的アルゴリズムに分類され、100%の到着率を保証することはできない。しかし、  $C_{th}$  を4~6程度に設定すれば、少ない再送回数で十分に高い受信率(99.9%以上)が得られる[17][28]。また、後で述べるように、  $T_{max}$  は遅延と再送信ノード数のトレードオフに影響を与えるパラメータであり、要求される遅延の範囲内で可能な限り長く設

定することが望ましい。

カウンタベース方式の最大の利点は、隣接ノードの情報（隣接ノード数など）が無くても、ノード密度に合わせて再送信確率を制御できる点である。ノード数が少ない場所では、各ノードの受信回数がしきい値  $C_{th}$  より少なくなる確率が高くなるので、メッセージを再送信するノードが多くなる。逆に、ノードが多い場所では再送信ノードが少なくなる。したがって、ノード密度の薄い場所でも高い到着率を保ったまま、全体の再送信ノード数を大きく減らすことができる。

また、

- GPS・RSSIなどのノード位置や距離を知るためのハードウェアが不要である。
- 隣接ノード情報の交換など、ブロードキャストメッセージ以外の通信によるオーバーヘッドがない。
- アルゴリズム自体はシンプルなので、実装が容易である。

という利点もあり、ハードウェア・ソフトウェア両面の制約がきびしいセンサネットワークに適したブロードキャストアルゴリズムであるといえる。

カウンタベース方式を改善するためのアルゴリズムとして、[24] や [25], [29] などが提案されている。[24] では各ノードがそれぞれの隣接ノード数に応じて  $C_{th}$  を変化させる方法が提案されている。[25] はブロードキャストメッセージ送信元からの物理的な距離を用いて動的に  $C_{th}$  を変化させるアルゴリズムである。[29] も [25] と同様に距離を用いるアルゴリズムであるが、こちらは  $C_{th}$  ではなく RAD の最大値を最適化する手法である。しかしながら、これらの方法は距離を含む隣接ノードの情報を用いるものであり、ハードウェアや通信オーバーヘッドが少ないというカウンタベース方式の利点が失われている。

そこで本研究では、隣接ノード情報を極力用いずに再送信ノード数を削減することを目的としてカウンタベース方式の改善をおこなった。

#### 4.4 待ち時間制御によるカウンタベース方式の改良

本研究ではカウンタベース方式を改善する、“Random Assessment Delay Extension (RAD Extension)” と “Hop Count Aware RAD Extension (HCA-RAD Extension)” という2つのアルゴリズムを提案する。まず、従来のカウンタベース方式で問題となる典型的

な状況について考察し、次に提案アルゴリズムについて述べる。

#### 4.4.1 基礎検討

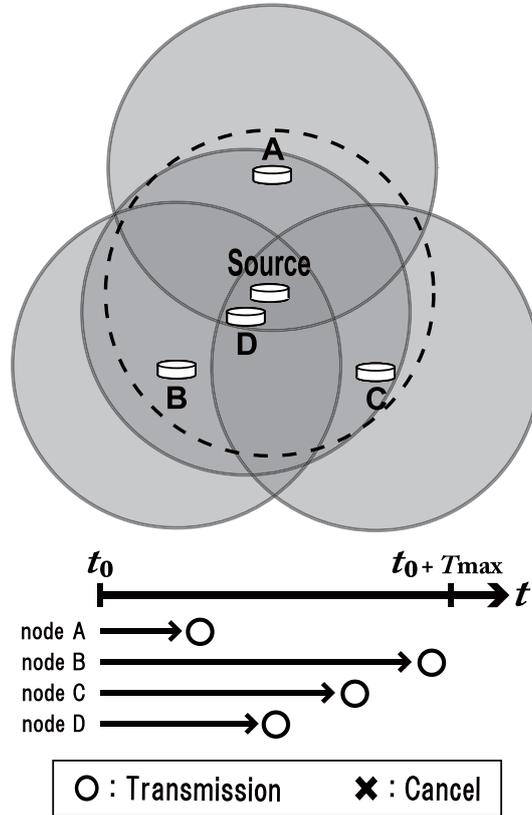


図 4.5 カウンタベース方式の動作例 (1 ホップ,  $C_{th} = 4$  の場合)

本節では、単純なノード配置例 (図 4.5 および図 4.6) を用いて、カウンタベース方式において冗長な再送信が発生する状況を考察する。なお、ここでは  $C_{th} = 4$  と設定する。

まず、図 4.5 のノード配置において Source ノードがブロードキャストをおこなう場合を考える。このときノード D は Source ノードに近い場所に位置しているため、再送信をおこなっても新たにカバーできる領域はほとんど無い。カウンタベース方式を用いて冗長なノード D の再送信を抑制するためには、ノード A, B, C すべてがノード D よりも RAD を小さく設定しなければならない。RAD は一様にランダムであると仮定すると、ノード D の再送信を抑制できる確率は  $1/4$  である。

次に、図 4.6 のノード配置で 2 ホップの例を考察する。この場合ノード A と B は距離が近く、両方のノードが再送信をおこなうのは冗長である。ノード A が先に再送信す

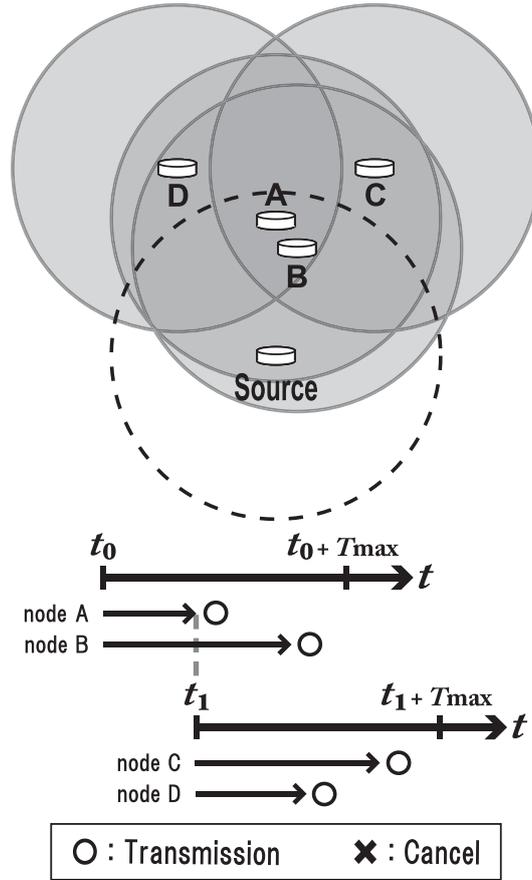


図 4.6 カウンタベース方式の動作例 (2 ホップ,  $C_{th} = 4$  の場合)

る場合, ノード B の再送信を抑制するためには, ノード C と D が先に再送信をおこなわなければならない.

ここで,  $T_{RAD}$  を  $[0, T_{max}]$  の範囲で一様に分布するランダムな RAD の値と定義し, ノード  $i$  の RAD を  $T_{RAD_i}$  と定義する. また,  $T_{RAD}$  の確率密度関数を  $p_{RAD}(t)$ , 確率分布関数を  $P_{RAD}(t)$  とすると, 図 4.6 の状況下でノード A または B の再送信を抑制できる確率は以下のように計算される. まず,  $|T_{RAD_A} - T_{RAD_B}|$  の確率密度関数は

$$p_{|RAD_a - RAD_b|}(t) = 2 \int_0^{T_{max}-t} p_{RAD}(t + \tau) p_{RAD}(\tau) \tau \quad (4.1)$$

となる. 再送信を抑制するための条件は  $\max\{T_{RAD_C}, T_{RAD_D}\} < |T_{RAD_A} - T_{RAD_B}|$  であり, この確率は

$$\int_0^{T_{max}} (P_{RAD}(t))^2 p_{|RAD_a - RAD_b|}(t) dt = \frac{1}{6} \quad (4.2)$$

となる. ノード C と D の RAD はノード A の再送信後に設定されるものであり, ノー

ド B の再送信を抑制できる確率は  $1/6$  しかない。

これらの冗長なノードの再送信を抑制できる確率を高めるには、ブロードキャストメッセージの受信回数が増えるノードほど RAD の期間を長く設定しなければならない。しかしながら、カウンタベース方式ではノードが RAD を設定する時点で受信回数を予測することはできない。

#### 4.4.2 RAD Extension の提案

前節で述べたカウンタベース方式の問題は、RAD を決定する時点ではノードが将来の受信回数を知るための情報（隣接ノードまでの距離、隣接ノード数など）を持っていないことに起因する。この問題を解決するために、ノードが受信回数に応じて動的に RAD を延長する“RAD Extension”アルゴリズムを導入する。提案方式の詳細を以下に示す。

1. 初めてブロードキャストメッセージを受信したノードは、カウンタを 1 で初期化し、 $0 \sim \Delta T$  の範囲でランダムな待ち時間 (RAD) を設定する。ここで  $\Delta T = T_{\max}/(C_{\text{th}} - 1)$  とする。
2. RAD のあいだは同じブロードキャストメッセージを受信するごとに、カウンタを 1 増やし、待ち時間を  $\Delta T$  増やす。
3. RAD 終了後、カウンタの値  $< C_{\text{th}}$  であればブロードキャストメッセージを再送信する。

ノードが再送信をおこなうまでの最大の待ち時間は  $T_{\max} = \Delta T(C_{\text{th}} - 1)$  である。

ここでもう一度図 4.5、図 4.6 の配置例を用いて RAD Extension の動作を示す。図 4.7 は図 4.5 と同じ条件における RAD Extension の動作例である。このときノード D は、ノード A と C からの受信によって RAD を 2 回延長し、ノード B からの受信によってカウンタの値が  $C_{\text{th}} = 4$  に達することで再送信をキャンセルする。再送信の抑制に失敗するのはノード A、B、C すべての RAD がノード D より長い場合だけであり、その確率は  $1/4$  である。したがって、RAD Extension によってノード D が再送信を抑制する確率は  $1/4$  から  $3/4$  へ改善される。

次に、図 4.8 に図 4.6 と同様の動作例を示す。この場合、ノード B はノード A からの受信によって時刻  $t_1$  に RAD を  $\Delta T$  延長する。同時にノード C と D は 0 から  $\Delta T$  の範

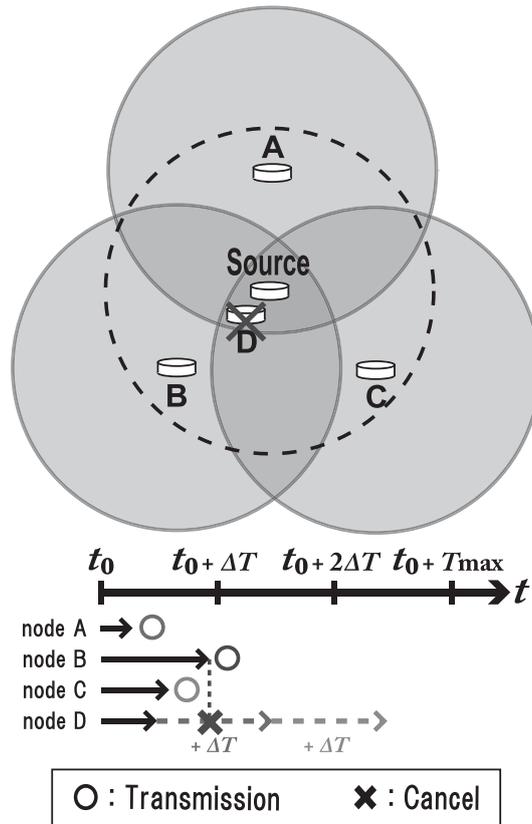


図 4.7 RAD Extension の動作例 (1 ホップ,  $C_{th} = 4$  の場合)

圏で RAD の初期値を設定するため、時刻  $t_1 + \Delta T$  までにノード C または D のどちらかが再送信をおこなう。ノード B の RAD が終了するのは時刻  $t_1 + \Delta T$  よりも後になるので、ノード C と D からの受信によってカウンタの値が  $C_{th} = 4$  に達し、再送信がキャンセルされる。したがって、この配置例の場合、必ずノード A または B の再送信が抑制されることになる。

ネットワーク全体で見た場合、特にノード密度の高い環境では上記の例と同様の状況が多数発生することが予想され、RAD Extension による再送信ノード削減効果が期待できる。しかし、経路構築の観点から見た場合、RAD Extension には経路長が長くなるという問題がある。RAD Extension では受信回数が増えるほど RAD が長くなるが、これはホップ数の大きい（中心から離れた）ノードの再送信確率が高くなることを意味する。よって最短経路を発見できる確率が減り、結果として経路長が増大してしまう。

図 4.9 に最短経路の発見に失敗する場合の例を示す。この場合ノード A と C がノード B より先に再送信をおこなうと、RAD Extension によってノード D が必ずノード B

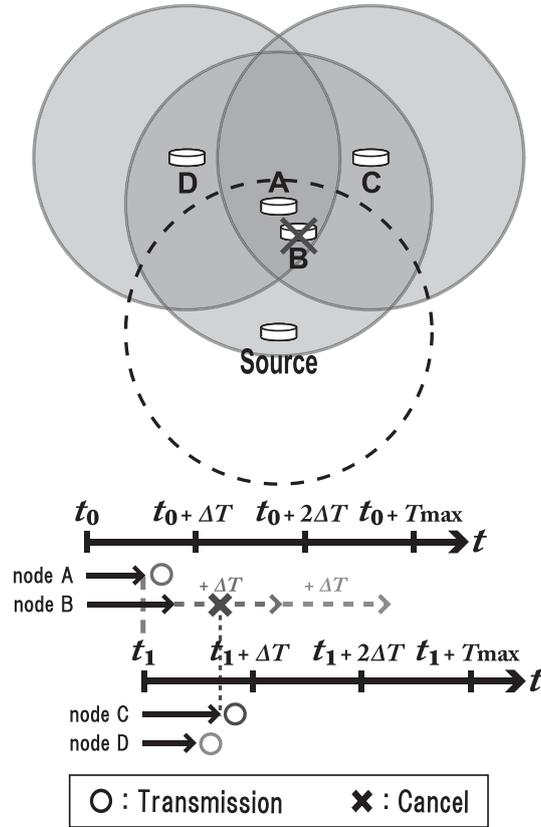


図 4.8 RAD Extension の動作例 (2 ホップ,  $C_{th} = 4$  の場合)

より先に再送信をおこなうため、ノード B は再送信をキャンセルする。Source ノードからノード D への最短経路はノード B を通る 2 ホップの経路であるが、ノード B が再送信をキャンセルするとノード A, C を通る 3 ホップの経路が構築されてしまう。

ここで、式 4.2 の計算と同じ定義を用いて、最短経路構築に失敗する確率を求める。まず、 $T_{nRAD}$  を  $n$  個の独立な  $T_{RAD}$  の和と定義し、その確率密度関数を

$$p_{nRAD}(t) = \int_0^t p_{RAD}(t - \tau) p_{(n-1)RAD}(\tau) \tau \quad (4.3)$$

とする。

カウンタベース方式を用いる場合に最短経路が構築されないのは、ノード A, C, D がすべてノード B よりも先に再送信をおこなう (すなわち  $T_{RAD_B} < T_{RAD_A} + T_{RAD_C} + T_{RAD_D}$  という条件を満たす) 場合のみであり、その確率は

$$\int_0^{T_{max}} P_{3RAD}(t) p_{RAD}(t) dt = \frac{1}{24} \quad (4.4)$$

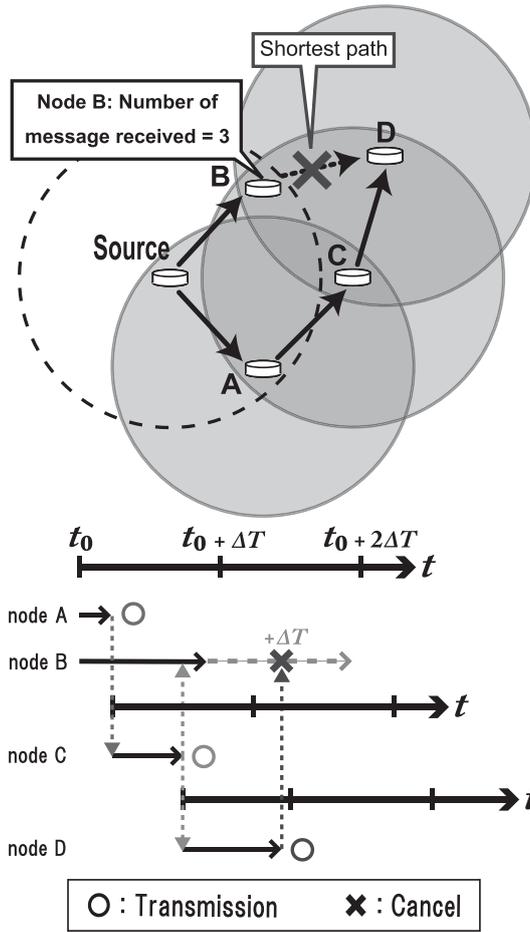


図 4.9 RAD Extension の動作例 (経路構築,  $C_{th} = 3$  の場合)

である。一方, RAD Extension を用いる場合はノード A と C がノード B よりも先に再送信をおこなった時点でノード B の再送信がキャンセルされるため,  $T_{RAD_B} < T_{RAD_A} + T_{RAD_C}$  という条件で最短経路構築に失敗する。この確率は

$$\int_0^{T_{max}} P_{2RAD}(t)p_{RAD}(t)dt = \frac{1}{6} \tag{4.5}$$

であり, カウンタベース方式と比較して 4 倍に増加していることがわかる。

#### 4.4.3 Hop Count Aware RAD Extension の提案

前節で述べた RAD Extension による冗長経路の問題を解決するために, Hop Count Aware RAD Extension (HCA-RAD Extension) を提案する。

RAD Extension は受信回数の多いノードの再送信優先度を下げることによって再送信ノード数を減らすアルゴリズムである。しかし, 冗長経路を防ぐためには, ホップ数の小さい

ノードの再送信優先度を上げなければならない。そこで HCA-RAD Extension では、再送信ノード数を増やさずに経路長を削減するために、受信回数と同じ場合にホップ数の小さいノードを優先できる RAD 調整アルゴリズムを導入する。

HCA-RAD Extension の詳細を以下に示す。

1. 初めてメッセージを受信したノードは、カウンタを 1 で初期化し、 $0 \sim \Delta T_{\text{HCA}}$  の範囲でランダムな待ち時間 (initial RAD) を設定する。このとき  $\Delta T_{\text{HCA}} = \frac{T_{\text{max}}}{2(C_{\text{th}}-1)}$  とする。さらに、initial RAD に  $(\Delta T_{\text{HCA}} - \text{送信元の initial RAD})$  をオフセットとして加算したものを RAD とする。
2. RAD のあいだは同じブロードキャストメッセージを受信するごとに、カウンタを 1 増やし、待ち時間を  $2\Delta T_{\text{HCA}}$  増やす。
3. RAD 終了後、カウンタの値  $< C_{\text{th}}$  であればブロードキャストメッセージを再送信する。

ここで  $\Delta T_{\text{HCA}}$  の値は、RAD Extension における  $\Delta T$  の 2 分の 1 とする。これは図 4.13 に示すように RAD をスロット化し、受信回数は同じだがホップ数の異なるノードに優先順位を付けるためである。図 4.13 から、ホップ数が  $H$  で受信回数が  $N$  のノードは必ず、ホップ数が  $H+1$  で受信回数と同じ  $N$  であるノードよりも先に再送信をおこなうことがわかる。また、このとき受信回数と同じ場合の再送信確率は RAD Extension と等しくなる。したがって、再送信回数を増やすことなく、平均経路長を削減することができる。

すべてのノードの initial RAD は  $\Delta T_{\text{HCA}}$  以下であり、 $(\Delta T_{\text{HCA}} - \text{送信元の initial RAD})$  を加算するので、最初の RAD は必ず  $\Delta T_{\text{HCA}}$  以上  $2\Delta T_{\text{HCA}}$  以下の範囲に設定される。また、RAD は最大で  $(C_{\text{th}} - 2)$  回、 $2\Delta T_{\text{HCA}}$  だけ延長される可能性がある。よって再送信完了までの最大遅延は  $T_{\text{max}} = 2\Delta T_{\text{HCA}} \times (C_{\text{th}} - 1)$  となる。

次に、前節、前々節で示した配置例を用いて HCA-RAD Extension の動作を示す。図 4.10 は図 4.5 および図 4.7 と同様の条件である。また、図 4.11 は図 4.5 および図 4.8 と同様の条件である。どちらの例でも HCA-RAD Extension は RAD Extension と同じ動作をおこない、冗長な再送信を抑制できる確率も同じである。さらに、図 4.12 は図 4.9 と同様の条件であるが、HCA-RAD Extension により必ず最短経路を発見できる。

HCA-RAD Extension を実装する場合に問題となるのは、送信元の initial RAD の値

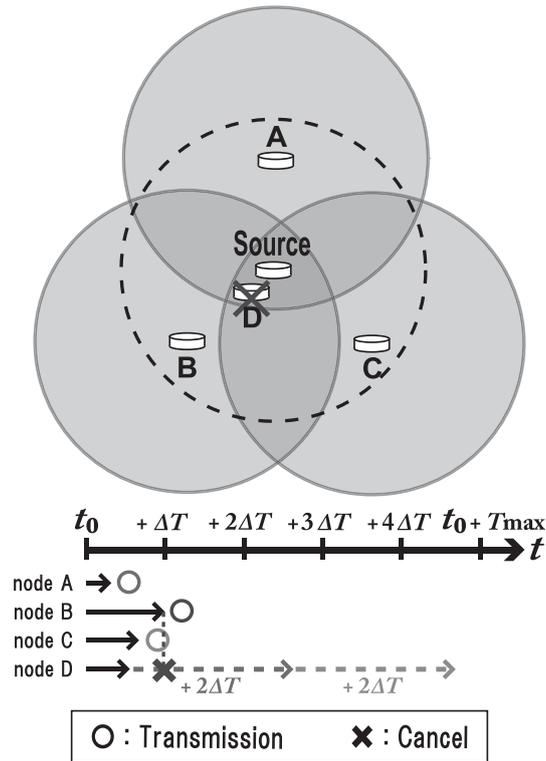


図 4.10 HCA-RAD Extension の動作例 (1 ホップ,  $C_{th} = 4$  の場合)

を受信ノードに伝える方法である。最も単純な方法はブロードキャストメッセージに initial RAD の値を付加することであるが、ブロードキャストメッセージのペイロードサイズが増加するというオーバーヘッドが存在する。(ただし、増加するデータ量は高々数 bit である。) もし、第 2 章で述べた FTSP や、GPS を用いてノード間の時刻が同期されているという前提であれば、単純な計算によって initial RAD を求めることができる。図 4.13 に示したように、すべてのノードの動作は 4.13 でスロット化されるため、受信ノードはメッセージ受信時刻とベースステーションのメッセージ送信開始時刻から送信元ノードの initial RAD を計算できる。

## 4.5 シミュレーションによる性能評価

ネットワークシミュレータ (QualNet[30]) を用いて、従来方式であるフラッディングとカウンタベース方式、提案方式である RAD Extension および HCA-RAD Extension の性能評価をおこなった。本節で示すシミュレーション結果は、すべて乱数の種を変えながら 50 回の試行をおこない、その平均値をとったものである。

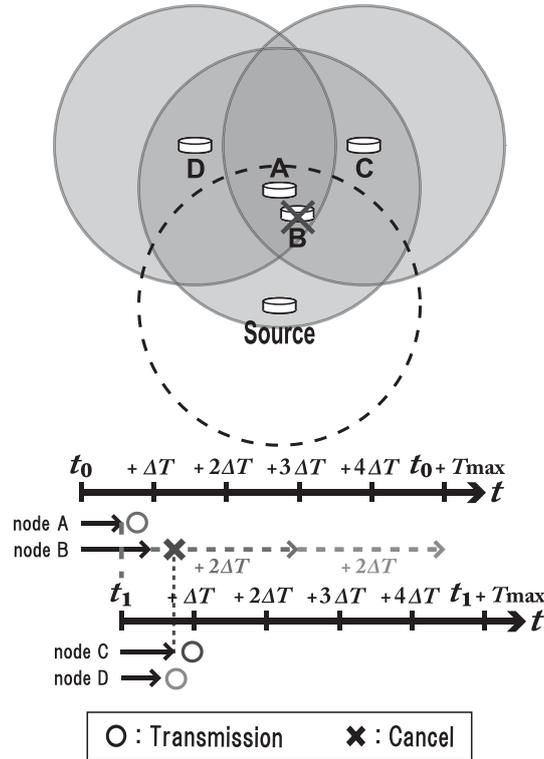


図 4.11 HCA-RAD Extension の動作例 (2 ホップ,  $C_{th} = 4$  の場合)

#### 4.5.1 シミュレーションモデル

まず、シミュレーション時のパラメータ設定について述べる。

フィールドサイズは  $100\text{m} \times 100\text{m}$  または  $300\text{m} \times 300\text{m}$  に設定し、センサノードは一律にランダム配置とする。すべてのシミュレーションでは中央に配置した基地局（ベースステーション）からブロードキャストをおこなう。

すべてのセンサノードの送信距離は  $20\text{m}$ 、変調方式は ASK (On-off keying)、伝送レートは  $10\text{ kbps}$  とする。また、消費電力は送信時  $800\mu\text{W}$ 、受信時  $500\mu\text{W}$ 、待機時  $0.5\mu\text{W}$  とする [31][32]。

MAC 層プロトコルは LPL (Low Power Listening, [14][15]) を用いる。LPL の起動周期は  $50\text{ms}$ [33] に設定する。

ブロードキャストメッセージの packet サイズは  $48$  バイトであり、packet の送信に必要な時間は上記の設定より最低  $38.4\text{ms}$  となる。

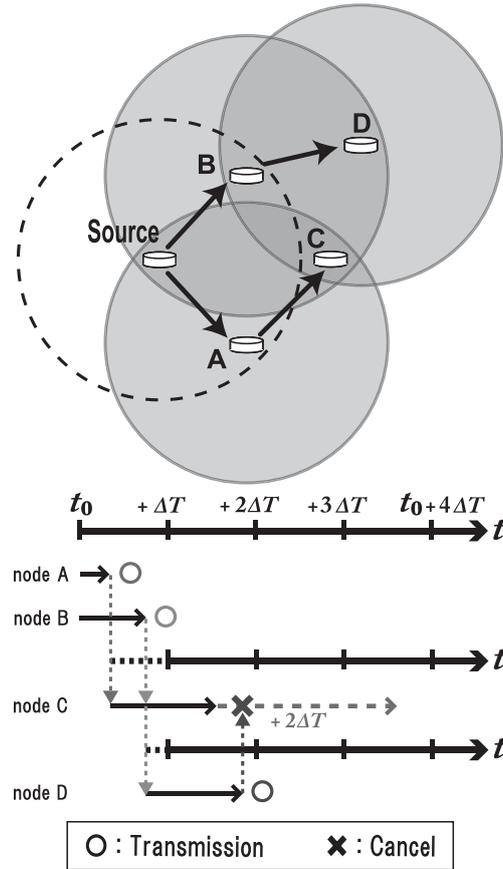


図 4.12 HCA-RAD Extension の動作例 (経路構築,  $C_{th} = 3$  の場合)

#### 4.5.2 シミュレーション結果と考察

ブロードキャストアルゴリズムの性能を評価する指標として、再送信ノード数、パケット到着率、遅延、および構築された経路の平均ホップ数についての比較をおこなう。なお、ここでいう「遅延」とは、ベースステーションがブロードキャストメッセージを送信開始してから、最後の再送信が完了するまでの時間である。また、カウンタベース方式、RAD Extension および HCA-RAD Extension には、RAD の最大値である  $T_{max}$  と再送信しきい値である  $C_{th}$  という 2 つのパラメータがある。 $T_{max}$  を大きく設定すると同時に送信をおこなう確率が減るので再送信ノード数を削減できるが、遅延が増大するというトレードオフがある。また、 $C_{th}$  は再送信ノード数とパケット到着率のトレードオフを決定するパラメータである。

まず、再送信のしきい値  $C_{th}$  に対する再送信ノード数とパケット到着率を図 4.14 に、

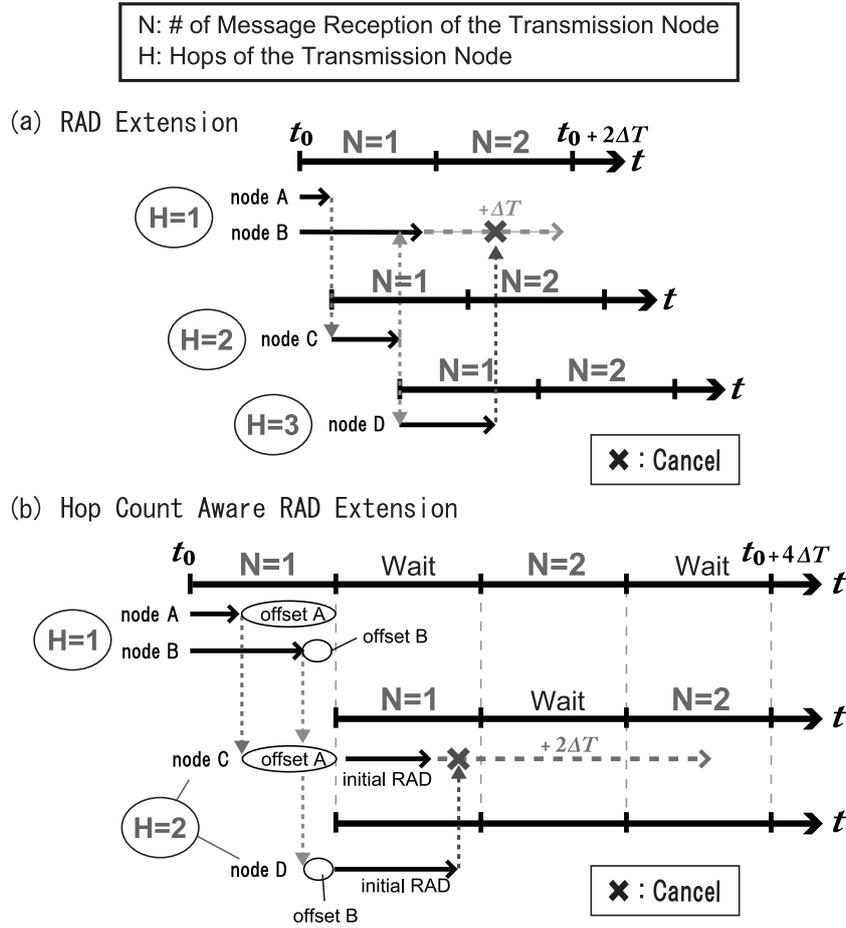


図 4.13. HCA-RAD Extension による RAD のスロット化: 図 4.9 の例におけるタイミングチャート (a) と, 図 4.9 の例におけるタイミングチャート (b)

平均ホップ数を図 4.15 に示す. このとき  $T_{\max}$  は 15s に設定した ( $T_{\max}$  条件については後述する).

図 4.14 (a) および図 4.15 (a) はフィールドサイズ  $100\text{m} \times 100\text{m}$ , 配置ノード数が 100 個の場合のシミュレーション結果である. 各ノードの送信距離は 20m に設定しているため, ノード密度は  $12.56\text{m}^{-2}$  である. 結果から, パケット到着率には有意な差は無く,  $C_{\text{th}}$  が 4 以上であれば 99.5% 以上の到着率となった. 一方,  $C_{\text{th}}$  が 3 以上の場合に RAD Extension と HCA-RAD Extension では再送信ノード数が約 10% 改善した. なお,  $C_{\text{th}}$  が 2 の場合には  $T_{\max} = \Delta T = 2\Delta T_{\text{HCA}}$  となるため, 再送信ノード数には差が無くなる. また, 平均ホップ数については, HCA-RAD Extension は常に RAD Extension より少なく,  $C_{\text{th}}$  が 3 以下であればカウンタベース方式よりもさらに少なくなった.

図 4.14 (b) および図 4.15 (b) は, ノード密度を変えずにネットワークの規模を 9 倍

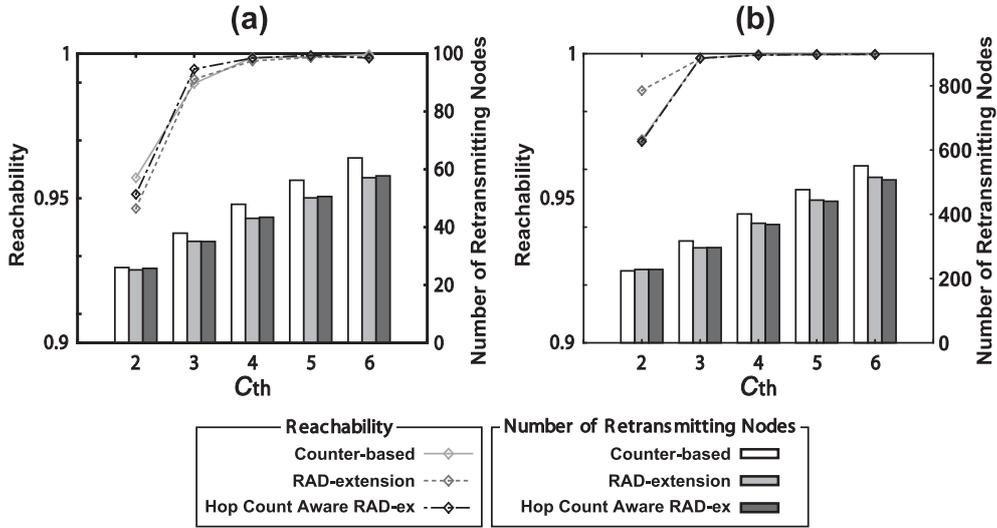


図 4.14  $C_{th}$  に対するパケット到着率と再送信ノード数の関係 ( $T_{max} = 15s$ )

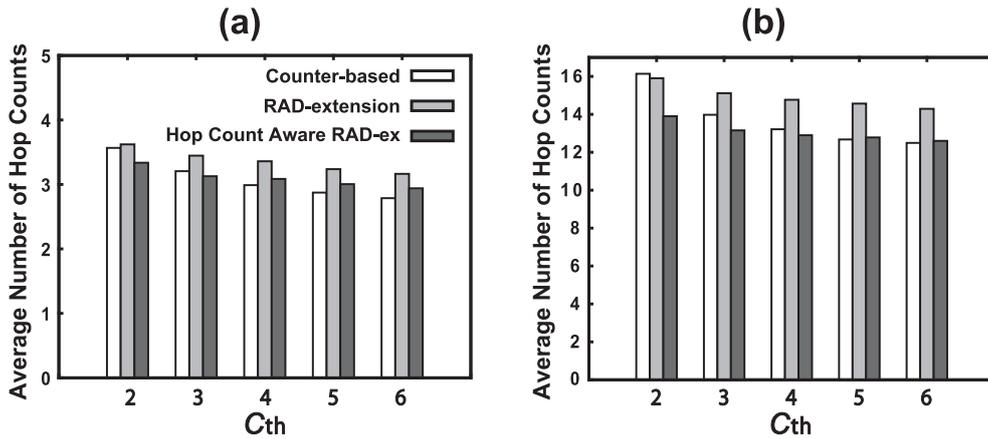


図 4.15  $C_{th}$  と平均ホップ数の関係 ( $T_{max} = 15s$ )

(フィールドサイズ  $300m \times 300m$ , 配置ノード数 900 個) に設定した場合のシミュレーション結果である。再送信ノード数、パケット到着率および平均ホップ数の傾向は変わらないが、 $C_{th}$  が 4 以上でも HCA-RAD Extension の平均ホップ数がカウンタベース方式より少なくなった。したがって、HCA-RAD Extension はネットワークの規模が大きくなればなるほど効果を発揮するアルゴリズムであると考えられる。

次に、 $T_{max}$  を変化させて評価をおこなった。ここではフィールドサイズを  $100m \times 100m$  とし、配置ノード数は 100 個、 $C_{th}$  は 4 に設定した。図 4.16 (a) はパケット到着率と  $T_{max}$  の関係を示しており、 $T_{max}$  はほとんどパケット到着率に影響を及ぼさないことがわかる。 $T_{max}$  が非常に小さいところではパケットのドロップが増加しているが、

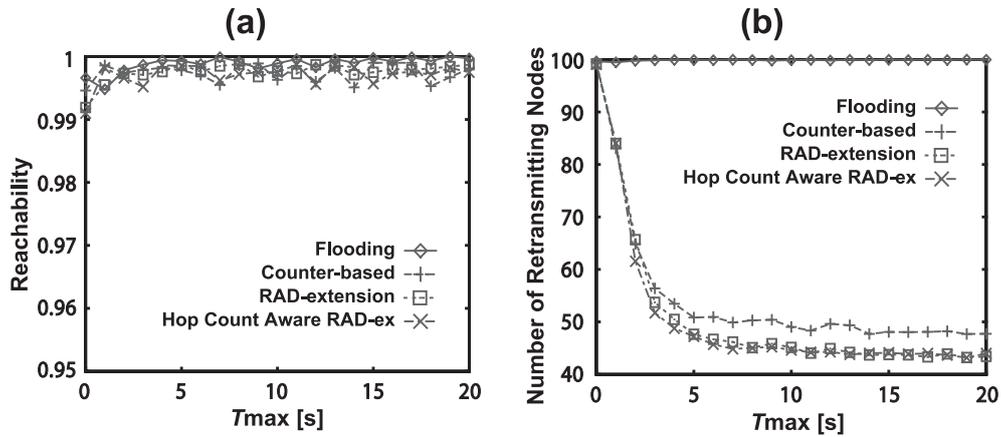
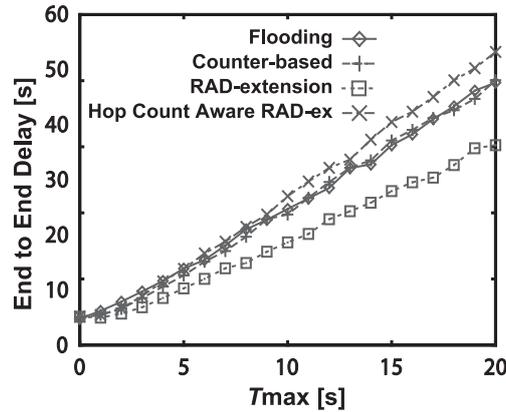
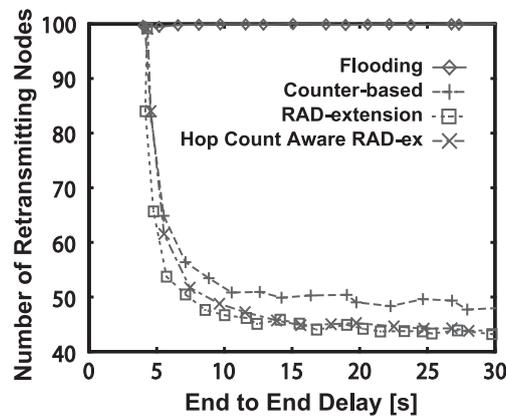


図 4.16  $T_{max}$  に対するパケット到着率 (a) と再送信ノード数 (b) の関係 ( $C_{th} = 4$ )

これは複数のノードが同時に送信をおこなうことで衝突が発生するためである。また、図 4.16 (b) は再送信ノード数と  $T_{max}$  の関係を示している。 $T_{max}$  が 5s より大きければ、RAD Extension と HCA-RAD Extension は常にカウンタベース方式よりも再送信ノード数が少なくなった。 $T_{max}$  が小さい場合に再送信ノード数が削減できないのは、到着率と同様に、同時に送信をおこなうノードが増えることが原因である。同時に複数のノードが再送信をおこなうと、受信メッセージ数がカウントされない状況が発生するため、結果として再送信ノード数が増加する。

図 4.16 (b) において再送信ノード数が十分に少なくなる  $T_{max}$  の条件は 5s 以上という結果が得られたが、これはシミュレーションパラメータから説明できる。まず、シミュレーションで LPL (MAC プロトコル) の起動周期を 50ms に設定しており、ブロードキャストメッセージ送信に必要な時間は伝送レートとメッセージサイズから最低 38.4ms である。したがって、受信メッセージ数が  $C_{th} = 4$  に達するまでには、最低でも 353.6ms の時間が必要となる。また、今回のシミュレーションモデルでは、ネットワーク層から MAC 層へ渡されたメッセージの送信はキャンセルできない仕様となっている。このため、平均隣接ノード数が 12 個を超える環境下では  $T_{max}$  を 5s 以上に設定する必要があったと考えられる。

図 4.17 は遅延と  $T_{max}$  の関係を示しており、End to End の遅延は  $T_{max}$  に対して線形に増加することがわかる。RAD Extension の遅延が他の方式と比べて小さくなっているのは、RAD の初期値が他の方式よりも小さく、ブロードキャストメッセージの拡散が早くなったためである。HCA-RAD Extension も RAD の初期値はカウンタベース方式よ

図 4.17  $T_{\max}$  と遅延の関係 ( $C_{th} = 4$ )図 4.18 遅延と再送信ノード数の関係 ( $C_{th} = 4$ )

り小さくなるが、ホップ数の小さいノードの再送信を待つというアルゴリズムの特性により、遅延は他の方式より大きくなっている。このように、ノードの最大待ち時間  $T_{\max}$  が同じであっても、アルゴリズムによって平均待ち時間は異なっている。そこで図 4.18 では、遅延が同じになる条件下での再送信ノード数を比較した。結果から、遅延時間が同じ場合でも再送信ノード削減率の傾向は変わらないことがわかる。

次に、ノード密度の変化に対する性能評価をおこなった。 $T_{\max}$  は 15s とし、 $C_{th}$  は 4、フィールドサイズは  $100\text{m} \times 100\text{m}$  に設定した。図 4.19 は配置ノード数（すなわちノード密度）に対するパケット到着率 (a) と再送信ノード数 (b) の変化を示している。ノード密度が等しければ、アルゴリズム間でパケット到着率に差は見られなかった。また、フラッディングはすべてのノードが再送信をおこなうのに対して、それ以外のアルゴリズムではノード密度が増加しても再送信ノード数はほとんど増加しなかった。さらに、

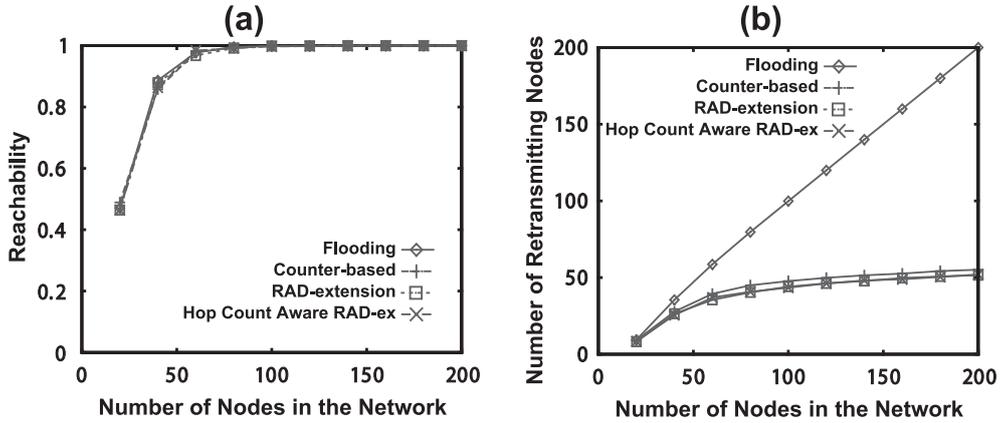


図 4.19 ノード密度に対するパケット到着率 (a) と再送信ノード数 (b) の関係 ( $C_{th} = 4, T_{max} = 15s$ )

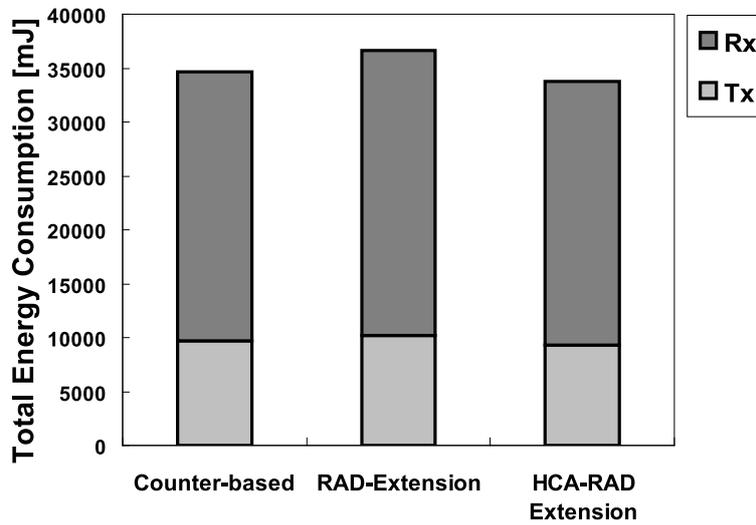


図 4.20 データ収集時の平均消費電力 ( $C_{th} = 4, T_{max} = 15s$ )

RAD Extension と HCA-RAD Extension の再送信ノード数は、常にカウンタベース方式よりも約 10% 少なくなった。

最後に、平均ホップ数の削減による効果を示すために、データ収集時におけるネットワーク全体の消費電力を評価した。シミュレーションでは、各ブロードキャストアルゴリズムによってベースステーションから経路を構築し、その経路に沿ってすべてのノードからベースステーションへデータの収集をおこなった。ルーティングプロトコルには Tiny Diffusion[16] を用いた。図 4.20 は、すべてのノードのデータを 1 回ベースステーションへ収集するために消費したエネルギーを示している。フィールドサイズは  $300m \times 300m$  であり、配置ノード数は 900 個である。また、 $C_{th}$  は 4、 $T_{max}$  は 15s、パ

ケット長は 48 バイトに設定した。なお、ここでは経路長の削減による効果のみを評価するため、ブロードキャストによる消費電力は含んでいない。

結果から、HCA-RAD Extension の消費電力は、カウンタベース方式より 2.6%、RAD Extension より 7.9% 少なかった。これは図 4.15 に示した平均ホップ数の削減効果によって平均経路長が短くなり、結果としてデータ送信回数とそれにもなうオーバーヒアリングが削減されたためである。

## 4.6 結言

本章では、センサネットワークにおいて経路構築等に必要不可欠なブロードキャストに着目し、待ち時間制御によるカウンタベース方式の改良アルゴリズム (RAD-Extension と HCA-RAD Extension) を提案した。提案方式では、コストの高いアルゴリズムやハードウェアを必要とせず、ノード密度にあわせて再送信確率を制御できるという従来方式の利点を生かしつつ、さらに再送信ノード数を削減することを目的とした。提案方式では、メッセージ受信ごとに待ち時間を増加させることで、各ノードの面積カバー率を改善する。これによってパケット到着率を維持したままで全体の再送信ノード数を削減できることを述べた。さらに、待ち時間のスロット化によって、経路長も考慮して再送信の優先順位を設定するアルゴリズムを提案した。また、シミュレーションによる比較をおこない、従来方式と比較して提案方式では約 10% の再送信ノード数削減効果が得られることを示した。



## 第5章

# 低消費電力センサノード VLSI の 垂直統合設計

### 5.1 緒言

本章では、プロトコル階層からアーキテクチャ階層までの垂直統合設計による低消費電力センサノード VLSI の設計研究について述べる。

センサネットワークの実用化において、センサノードの低消費電力化によるシステム可用時間の長期化は大きな課題である。センサノードは通常バッテリーで動作すると考えられているが、大規模なセンサネットワークでは多数のセンサノードが分散配置されるため、バッテリー交換の頻度を最小化しなければならない。したがって、センサネットワークシステムの低消費電力化が重要となる。また、ノードを安価に大量生産可能とすることも重要な課題である。そのためにはセンサノードの部品点数を削減する必要があり、センサノードを SoC (System on Chip) として実装することが非常に有効となる。また、センサノード SoC は低消費電力化にも寄与する。

センサノード SoC を検討するにあたっては、垂直統合設計をおこなうことが重要である。従来のネットワークシステムとは異なり、消費電力が重要な性能指標となるセンサノードの設計においては、ハードウェアの回路構成(下位層)が、データリンク層、ネットワーク層(上位層)の設計に大きな影響を及ぼす。また、上位層のプロトコル設計により、下位層に相当する回路設計が影響を受ける。よって、システム全体の低消費電力化には、データリンク層以上の通信プロトコルと物理層に相当するシステム回路の双方にわたる垂直統合設計が必要である。しかし従来の研究は、通信プロトコル、SoC のどちらか一方を主体とする研究がほとんどであった。

本研究の目的は、垂直統合設計による超低消費電力なセンサノードの実現である。本

研究では、まず MAC プロトコルの低消費電力化をおこない、MAC 層からの上下方向展開によるアルゴリズム階層と回路アーキテクチャ階層との相互最適化をおこなう。

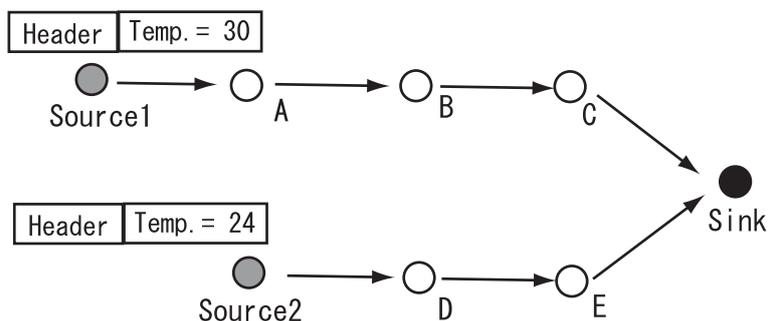
## 5.2 各プロトコル階層における低消費電力化の課題と関連研究

本節ではセンサネットワークの各階層における低消費電力化のためのアプローチについて概説し、従来の研究における課題について述べる。センサネットワークでは、主にネットワーク層、データリンク (MAC) 層、物理層がセンサノードの消費電力に大きな影響を与えるため、この3つの階層における研究が多く進められている。ネットワーク層では、主に経路制御とデータ集約が低消費電力化につながっている。また、MAC 層ではアイドルリスニングの削減、パケットの衝突の回避、オーバーヒアリングの回避が低消費電力化のアプローチとしてあげられる。さらに、物理層では MAC 層と連携した通信回路の間欠動作、各モジュールの電源管理がアプローチとしてあげられる。しかし、これらの階層を越えて、通信プロトコルからセンサノードの回路・アーキテクチャレベルまでの一貫した低消費電力化をおこなう研究は、ほとんど成されていないのが現状である。

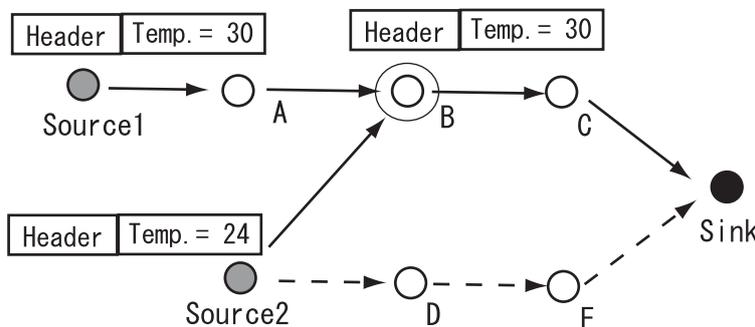
### 5.2.1 ネットワーク層

本節ではネットワーク層における低消費電力化のためのアプローチについて述べる。従来のインターネットなどにおけるネットワーク層は、基本的にアプリケーションのデータには依存しない、アドレスセントリックな考え方であった。しかしセンサネットワークにおいては、ユーザにとって重要なのはデータそのものであり、ノードの識別は重要ではない。そこでセンサネットワークでは、従来のアドレスセントリックではなくデータセントリックな経路制御が考えられている [19] (図 5.1)。

データセントリックには2つの利点がある。まず1つ目の利点は、ネットワーク全体のノードの ID を管理する必要がなく、隣接ノードが識別できれば十分とであるという点である。これはノード ID の管理にかかわるオーバーヘッドの削減につながる。もう一つの利点は、複数のセンシングデータから必要な情報のみを抽出したり、あるいは複数のセンシングデータを結合して一つのパケットを格納したりすることで、データ転送量を削減できる点である。この手法はデータ集約とよばれる。データ集約によるデータ転送量の減少は、ネットワーク全体の消費電力削減につながる。以下に経路制御および



(a) Address-centric Routing



(b) Data-centric Routing

図 5.1 アドレスセントリックなルーティングとデータセントリックなルーティング

データ集約の概要を述べる。

### 経路制御

センサネットワークでは、パケットのペイロード（通信するデータのうちヘッダを除いた部分）の内容に基づいて経路がホップ・バイ・ホップで決定される。センシングデータは温度や湿度といった“属性”とそれぞれの属性に対する“値”として表現される（例えば、〈属性=温度，値=30度〉のように表現される）。このようなセンシングデータがネットワーク内を転送される際、中継ノードは、パケットのペイロードに格納されたセンシングデータの属性と値のペアを調べ、この情報を転送すべき隣接ノードを決定する。すなわち、中継ノードは最終目的ノードのノード ID に基づいて次ホップを決定するのではなく、パケットペイロード内に格納されたセンシングデータの内容に従って次ホップを決定する。このデータセントリックの代表的な経路制御としては Directed Diffusion, Sensor Protocols for Information via Negotiation (SPIN), Low-Energy Adaptive

Clustering Hierarchy (LEACH), Power-Efficient Gathering in Sensor Information Systems (PEGASIS) があげられる。

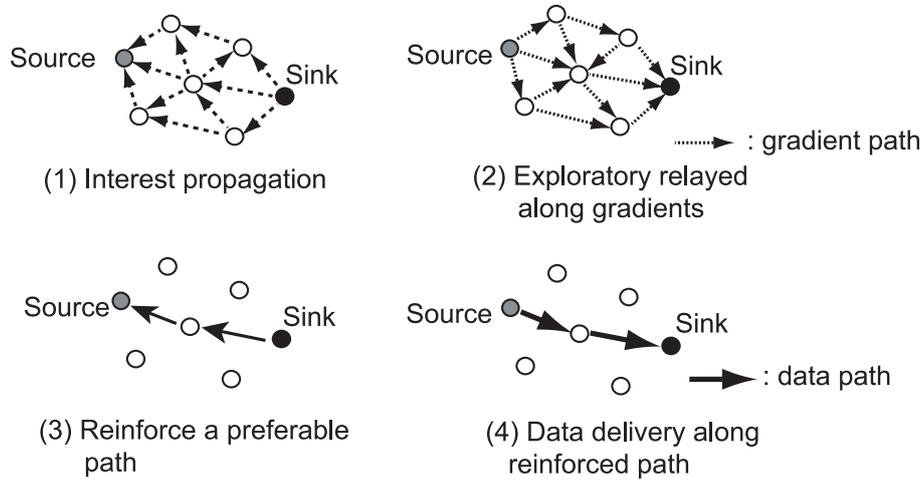


図 5.2 Directed Diffusion

図 5.2 に示した Directed Diffusion[21] は属性ベースの名前付けを利用し、データセンタトリックを意識している。Directed Diffusion は情報を必要としている Sink ノードと、Sink ノードが必要な情報を持つ Source ノードとの経路をつくるための経路制御プロトコルである。Directed Diffusion の経路設定の概略としては、まず Sink ノードが interest とよばれるパケットをネットワーク全体に送信する。Interest パケットを受信したセンサノードのうち、欲しい情報を持ったセンサノードが Source ノードとなり、interest が送られてきた方向すべてに対して exploratory data を送る。Exploratory data を受信した Sink ノードはいくつかの経路の中から一番早く送られてきた経路に対してさらに reinforce パケットを送信する。Sink ノードはこの経路を遅延の小さい確実な経路として判断し、以後この経路を用いて通信をおこなうように決定する。Source ノードは reinforce パケットの戻ってきた経路に対してデータを送ることになる。また Directed Diffusion は four-way handshake プロトコルであるが、exploratory data と reinforce を簡易化した two-way handshake の Tiny Diffusion とよばれるプロトコルも提案されている [16].

SPIN [34] は Directed Diffusion の経路構築手順を逆におこなうプロトコルである。まずデータを保持しているノードは、周辺のノードに対して自分がどのような情報を持っているかを広告する (ADV)。周辺のノードは、その情報が必要であれば、ADV を

送信したノードに対して要求メッセージ (REQ) を送信する。そして ADV を送信したノードは、REQ を返してきたノードに対してデータを送信する。Directed Diffusion が four-way handshake だったのに対して、SPIN は Three-way handshake で通信することになる。

LEACH[35] および PEGASIS[36] は Directed Diffusion のようなツリー型とは異なるクラスタ型の経路制御方法である。クラスタ型経路制御では、ネットワークをいくつかのクラスタに分割し、その中でクラスタヘッドとよばれるノードを決める。クラスタヘッドはクラスタ内のノードからデータを集め、基地局に送信する。クラスタ内で送信を TDMA でスケジューリングして通信し、クラスタヘッドから基地局へは CDMA で通信をおこなう。

### データ集約

データセントリックなネットワークであるセンサネットワークにおいて、データ転送量は消費電力に大きな影響をおよぼす。そこでデータ転送量を削減するために、データ集約とよばれる手法が用いられる。データ集約とは複数のデータを圧縮したり、データの冗長な部分を削除することによって、ネットワーク全体のデータ転送量を削減する手法である。データ集約は、完全集約（非可逆性の集約）と線形集約（可逆性の集約）という 2 種類の方法に大別できる [37]。図 5.3 は各集約方法の概念図である。

**完全集約** 完全集約とは、データパケットを中継するノードが、データそのものを圧縮・加工する方法である。例えばセンシングしたデータの最大値、最小値あるいは平均値を必要とするようなアプリケーションでは、中継ノードはすべてのデータを次ホップに送信する必要はない。最大値のみを必要とする場合は、中継ノードで複数のデータから最大値を計算し、その演算結果のみを次ホップのノードへ転送すればよい。

**線形集約** 線形集約とは、データパケットを中継するノードが、転送先が同一のデータを一つのパケットに格納して転送する方式である。個別のパケットに格納してデータを転送する場合はそれぞれのパケットにヘッダが必要であるが、線形集約によってデータを一つのパケットにまとめることでヘッダを削減できる。センサネットワークでは、ヘッダはペイロードに比べて無視できないサイズになると予

想されていたため、線形集約によるデータ転送量の削減効果が大きいと考えられる。

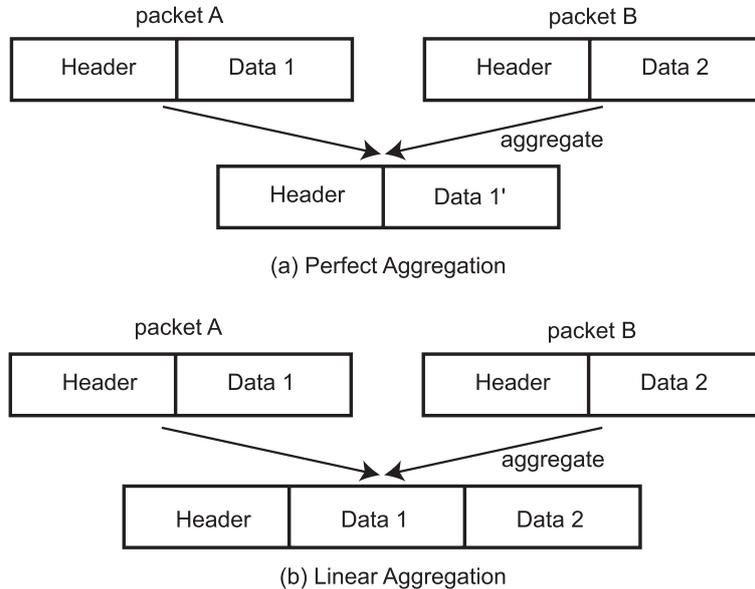


図 5.3 完全集約と線形集約

このほかにもデータの相関を利用した beam forming[38] や、データの並び順によってデータを圧縮する data funneling[20] などもある。しかしほとんどのデータ集約方法はアプリケーションに依存する。特に完全集約を用いることができるアプリケーションは限定される。これに対し、線形集約はどのようなアプリケーションにも応用が可能である。

### 5.2.2 データリンク層 (MAC 層)

センサネットワークの MAC プロトコルを設計するにあたって、最も重要な特性は消費エネルギーである。現在研究されているセンサネットワークの MAC プロトコルは、様々な方法で無駄な消費エネルギーの要因となるオーバーヘッドを削減している。本節では、まずセンサネットワークの MAC プロトコル設計において要求される特性を述べる。その中で最も重要な特性は消費電力であり、アイドルリスニングによる消費エネルギーを削減することが必要であることを述べる。次にセンサネットワークの MAC プロトコルに関する関連研究について概説する。

### MAC プロトコルへの要求

MAC プロトコルはユニキャスト、マルチキャスト、ブロードキャストという3種類の通信方式を保証しなければならない。ユニキャストとは、送信すべきデータを持った「Sender」が、1つの「Receiver」に向けてデータを送信することである。1対1の通信方式であり、最もよく用いられる。マルチキャストとは Sender が任意の複数 Receiver にデータを送信することである。センサネットワークではあまり用いられない。ブロードキャストとは Sender が送信できる Receiver すべてにデータを送信することである。ブロードキャストされたデータはアドレスに関係なく受信される。マルチキャストよりよく用いられる (図 5.4)。

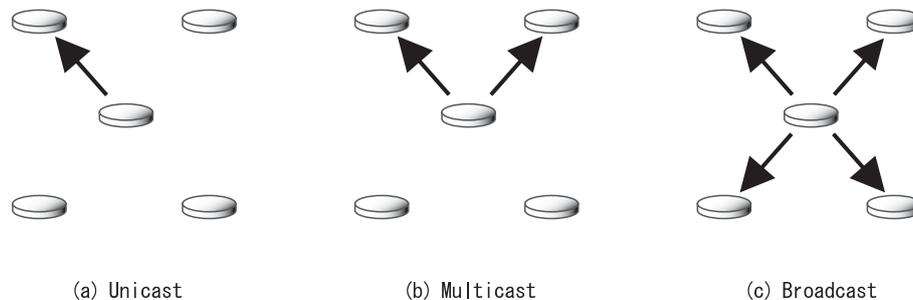


図 5.4 MAC プロトコルが保証する3種類の通信

さらにセンサネットワークの MAC プロトコル設計においては、通信保証と共に以下の7つの特性を意識する必要がある。

- 消費エネルギー
- 衝突回避
- 遅延
- 適応性
- チャンネルの利用効率
- スループット
- 公平さ

7つの特性の中で最も着目すべき特性は消費エネルギーである。なぜなら大量に散らばったセンサノードのバッテリー交換の手間を省くために、限りあるエネルギーを有効

に利用しノードの寿命を伸ばす必要があるからである。センサノードの通信モジュールによる消費電力は、一般的に他のモジュールと比べて大きく、全体の消費電力に対して支配的であるため、通信モジュールを制御する MAC 層は非常に重要である。したがって、消費電力オーバーヘッドを抑えることに重点を置いた MAC プロトコル設計が必要である。一方、その他の特性の重要度はアプリケーションに依存する。例えば防犯のようリアルタイムなセンシング情報を収集するアプリケーションであれば遅延、スループットを意識する必要がある。また、センシングする対象が移動するような場合には、隣接ノードの頻繁な変更に対する適応性が重要になる。

MAC プロトコルにおける消費エネルギーのオーバーヘッドには、以下のような要因がある。

- データの衝突による再送
- オーバヒアリング
- アイドルリスニング
- プロトコルの制御パケット

1つ目のデータの衝突による再送は MAC プロトコルの特性にも係わる重要な要因である。衝突とは、Receiver がデータを受信している際に、Receiver 以外の隣接ノードが同時に通信をおこなうことによって、正常に通信できなくなる現象である。センサネットワークでは通信媒体として電波を使用するため、隣接する複数のノードが同時に通信をおこなうと電波干渉が起き、データを正しく受信できない。Sender は送信している間、Receiver がデータを正しく受信しているかを確認することができない。したがって、Receiver で衝突が発生している場合でも、Sender は途中で送信を中断することができず、すべてのデータを送信する。このため、衝突が発生すると1回のデータ送受信のエネルギーを無駄に消費することになる。また、衝突はデータの遅延増大の要因にもなる。

2つ目のオーバヒアリングとは指定した Receiver 以外のノードがデータを受信することである。無線通信においてデータは電波伝搬により同心円状に広がるため、周囲の隣接ノードすべてにデータが届くことになる。ノードはデータ受信によりエネルギーを消費するため、Receiver 以外のノードの受信はエネルギーの浪費となる。特にノードの密度が高い場合は、Receiver 以外のノードのオーバヒアリングを回避することが必要に

なる。

3つ目のアイドルリスニングとはデータの送受信がない時(アイドル時)に受信機が動作している状態を意味する。受信機が動作している場合、データ非受信時でも受信時とほぼ同量のエネルギーを消費する。したがってアイドルリスニングによって大きく電力を消費してしまう。データ受信時以外は受信機を非動作状態にすることが理想である。このため、アイドルリスニングを減らしつつもデータの送受信が確実におこなえるような MAC プロトコルを設計する必要がある。

4つ目のプロトコルの制御パケットについて述べる。スケジュール管理、同期、衝突回避などのために、ノードはデータパケットとは別の制御パケットを送受信する場合がある。制御パケットを用いることで様々な機能をプロトコルに実装することが可能である。例えば IEEE802.11[39] に実装されている RTS(Request To Send) と CTS (Clear To Send) のような、衝突を回避するための制御パケットなどがある。

これら4つの中で特に意識すべきオーバーヘッドの要因はアイドルリスニングである。センサネットワークでは稼働時間の約9割が待機状態となるため、アイドルリスニングを削減することによって消費電力を効果的に削減できる。アイドルリスニングを削減する方法として、間欠動作が用いられる。受信機を一定の割合でスリープさせることにより、アイドルリスニングの割合を相対的に削減する(図5.5)。

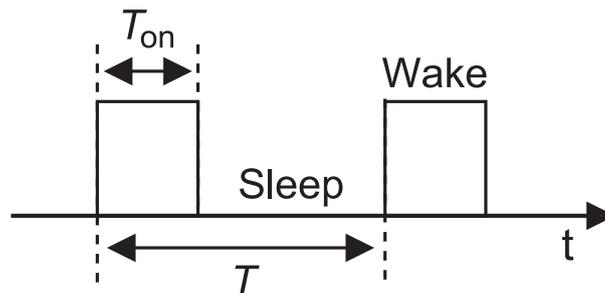


図5.5 間欠動作

しかし、間欠動作はアイドルリスニングが抑えられる一方で、通信不可能問題が生じてしまう。図5.6は3つのノードがそれぞれ独立に間欠動作をした例である。ノードAとBは動作タイミングが近いために通信可能である。一方ノードBとCが通信可能となる時間がほとんど無く、ノードAとCでは相互通信が不可能である。このため間欠起動型 MAC プロトコルでは、同期・非同期・準同期のいずれかの方法により間欠動作

における相互通信を確立する必要がある。また衝突の回避にはキャリアセンシングが有効である。キャリアセンシングとは、送信前に受信状態となることで媒体が使われているかを確認する方式である。ただし、隠れ端末問題とよばれるデータ衝突は、キャリアセンシングだけでは回避することができない。

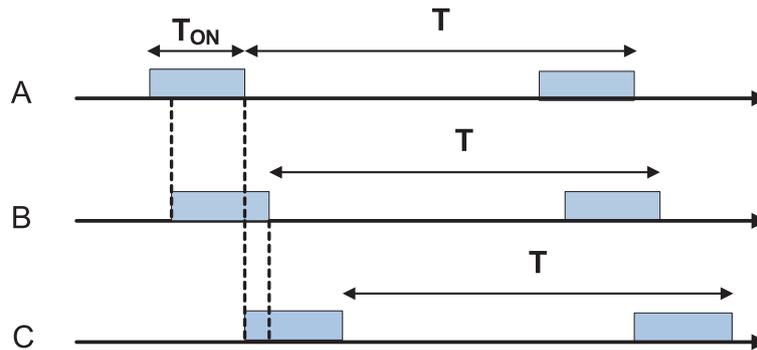


図 5.6 通信不可能問題

隠れ端末問題とは二台の端末がお互いに電波を送出していることを認識できずに通信を始め、衝突が発生するという問題である。図 5.7 を用いて隠れ端末問題について説明する。まずノード B がノード C にデータを送信していたとする。ノード A はノード B の通信を検知できるので、もしノード A が送信したいデータを保持していたとしてもノード B の送信が終了するまでデータ送信を控える。しかしノード D はノード A の通信を検知することができない。そのためノード B の送信中にノード D がノード E に対して送信を開始すると、ノード C でデータの衝突が発生する。これが隠れ端末問題によるデータの衝突である。

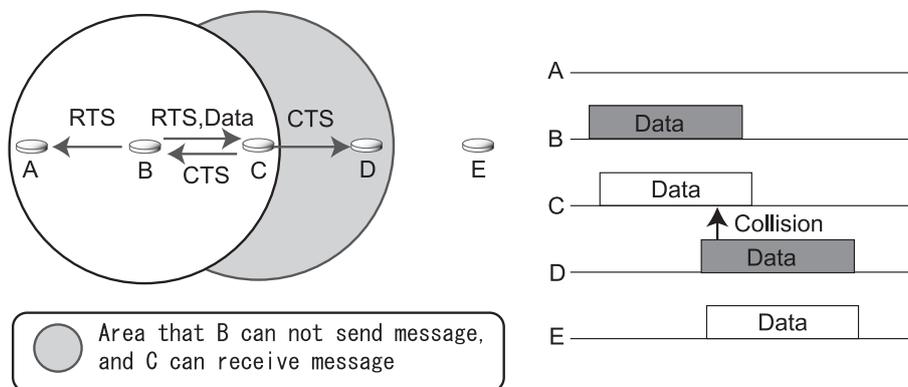


図 5.7 隠れ端末問題

隠れ端末問題を回避するために、IEEE 802.11 では Request To Send(RTS) と Clear To Send(CTS) という制御パケットを用いてデータを送受信する。図 5.8 を用いて RTS/CTS 交換について説明する。まずノード B はデータ送信前に RTS メッセージを送信する。RTS メッセージには送信ノード ID と受信ノード ID が記述されている。RTS メッセージをノード C が受信し、データ受信可能状態であれば CTS メッセージを送信する。CTS メッセージにも RTS メッセージと同様に送信ノード ID と受信ノード ID が記述されている。このとき RTS/CTS メッセージを周囲のノードも受信する。よって RTS/CTS を用いない場合と異なり、ノード D もノード B とノード C 間の通信を検知できるため、ノード E に対するデータ送信を控え、衝突を防ぐことができる。

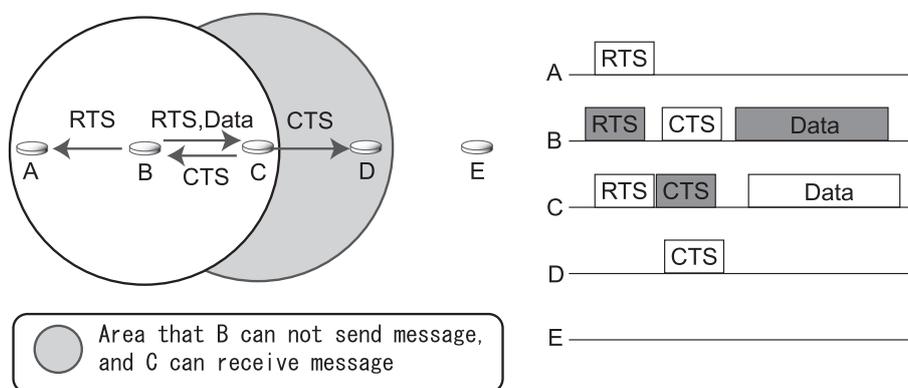


図 5.8 RTS/CTS 交換

以上のように RTS/CTS を用いることによって隠れ端末問題を回避することができるが、制御パケットを多用すると制御パケットによる送受信の消費エネルギーが増大することになる。したがってセンサネットワークの MAC プロトコル設計においては、制御パケットを用いることで得られる機能と消費エネルギーのトレードオフをアプリケーションに応じて考慮する必要がある。

### MAC プロトコルに関する研究

センサネットワーク用の MAC プロトコルとしてこれまで多くのプロトコルが提案されている。MAC プロトコルを大別するとコンテンション型、スケジューリング型、リソース分割型などに分類できる。

**コンテンション型** 通信資源をチャネルのランダムアクセスにより共有する方式。

Carrier Sense Multiple Access (CSMA), ALOHA をベースに, センサネットワークにおいては, CSMA/CA[40], MACA[41], SMAC[31][32], TICER/RICER[42], LPL[14][15], WiseMAC[43], Wake up beacon[44] などが提案されている.

**スケジューリング型** 通信資源をチャネルのスケジューリングにより共有する方式. 時間軸において分割するため Time Division Multiple Access (TDMA) とよばれる. センサネットワークにおいては, PACT[45], BMA[46], EMACs[47], LMAC[48], TRAMA[49], SSTDMA[50], Arisha[51] などが提案されている.

**リソース分割型** 周波数において通信資源を分割する Frequency Division Multiple Access(FDMA) 方式や拡散符号を用いる Code Division Multiple Access(CDMA) 方式など.

TDMA 方式は資源の割り当てに限界があることに加え, 高い精度の同期が前提となるために同期誤差(クロックドリフト)による通信障害時の通信保障が困難である. 膨大な数のノードが通信資源を共有する必要があるセンサネットワークにおいては, スケーラビリティに優れるコンテンション型が主流となっている.

### 5.2.3 物理層における低消費電力化

センサノードにおける物理層は, センサ機能, 通信機能, CPU, Real Time Clock (RTC) を有するセンサノードから成り立っている. 物理層における課題は, センサノードの消費電力を減らしシステムの可用時間を延ばすことである. そのためにセンサノードをシステム LSI 化する必要がある.

本節ではまず物理層に対する要求を明確化し, 次に物理層における他研究の成果について概説する.

#### 物理層への要求

センサネットワークの実用化における最大の課題は, システムの可用時間を最大限に延長することである. センサノードは通常バッテリーで動作すると考えられるが, 多数のセンサノードが分散配置されるため, バッテリーの交換頻度を最小化する必要がある. 限られた電源容量で長時間動作を可能とするためには, 物理層の低消費電力化が必須となる.

例えば MAC 層では, センサノードを間欠動作させることで, ネットワーク全体とし

ての低消費電力化を図る。各ノードは必要なときのみ通信をおこない、それ以外のほとんどの時間は通信回路の電源を落とす。MAC 層の役割はこのセンサノードの状態を制御し、ノード間の通信を補償することである。このとき、センサノードを機能ブロックごとに電源管理し、必要なときに必要な機能だけを動作させるために、電源管理機構が必要となる。

また MAC 層は通信終了時において、休止状態に入ってから復帰するまでの時間を正確に把握している必要がある。この時間カウント用のクロック発生回路 (RTC) は、ノードの休止状態でも常に動作している必要があるため、超低消費電力でなければならない。RTC の提供には通常、低周波数な水晶発振回路が用いられる。これは水晶発振回路が非常に高い Q 値を持ち、数 ppm オーダーの安定な発振が可能であるためであり、低周波数なものほど発振に必要な電力を小さくできる。

さらに、同期通信をおこなう場合には多数のノードが同時に通信をおこなおうとすることから、衝突を回避するためにキャリアセンス機能を持つ必要がある。

ノードを安価に大量生産可能とすることも重要な課題である。そのためには、センサノードの部品点数を削減する必要があり、センサノードの機能をシステム LSI (Large-Scale Integration) 上に実装することが有効となる。センサノードの LSI 化は、低消費電力化にも寄与する。

以下に物理層に対する要求をまとめる。

- 電源管理機構の実装
- 間欠動作のための RTC の低消費電力化、および高精度な同期機構
- キャリアセンス機能の実装
- FDMA 方式に対応した通信回路
- 外付け部品の削減による低コスト化

これらの実装により、ネットワークシステム全体としての低消費電力化・低コスト化を実現する。

### 物理層に関する研究

センサネットワーク用のセンサノードとして、これまで多くの小型モジュールが開発されている。センサネットワークの発想はもともとアメリカの軍事運用が根底にあつ

たことから、センサノードの開発も主にアメリカ UCB による功績が大きい。表 5.1 に UCB/Crossbow が開発した Mica シリーズ [15]、および Pico Radio[52]、日本の Ni3[53] のスペックの比較を示す。Mica シリーズ、Ni3 に関してはすでに試験的運用もされている。

表 5.1 センサノードの比較

Project Team		UCB/Crossbow				UCB	Waseda
Project Name		Mica(MOTE)	Mica2Dot	Mica2	Telos-B	PICO Radio	Ni3
RF	Modulation Type	ASK	FSK	FSK	O-QPSK	OOK/FSK	ASK/OOK
	Frequency[MHz]	433/915	315/433/915	315/433/915	2400~2483.5	1900	303.825
	TX Power[mW]	36	42	42	38	8.5	36
	RX Power[mW]	12	29	29	35	3.6	13.5
	Bitrate[kbps]	40	38.4	38.4	250	40	19.2
MCU	Data RAM[kbyte]	512	512	512	1024	64	8
	Program RAM[kbyte]	128	128	128	48	64	14.336
	RAM(cash)[kbyte]	4	4	4	10	4	0.368
	Active[mW]	8	8	33	3	0.527	1.8
	Sleep[uW]	75	75	75	15	10.5	2.7
	Wakeup Time[us]	180	180	180	6	10	-
RTC	Frequency[kHz]	32.768	32.768	32.768	-	-	8000
note		DATA RAM is a Nonvolatile	w/ MCU(i8051) w/ Power management	DATA RAM is an EEPROM			

通信に用いる周波数は、Mica では免許不要の国際標準である ISO18000 系、Ni3 では微弱無線規格となっており、データレートは数 10kbps 程度となっている。ほとんどのセンサノードは電源が OFF になってもデータが消えないように不揮発性メモリを搭載している。Pico Radio においては不揮発メモリを使用せず、細かな電源管理機構を実装することで消費電力を削減している。

#### 5.2.4 各階層ごとの低消費電力化における課題

従来のネットワークシステムとは異なり、消費電力が重要な性能指標となるセンサノードの設計においては、ハードウェアの回路構成(下位層)は、MAC 層およびネットワーク層(上位層)の設計に大きな影響を及ぼす。また、上位層のプロトコル設計により、下位層に相当する回路設計が影響を受ける。その結果、回路レベル、あるいはセンサノード単体で見れば非常に小さな消費電力を達成できたとしても、ネットワークレベルで見れば結果として消費電力が悪化するということが容易に起こりえる。したがって、システム全体の低消費電力化をおこなうためには、MAC 層以上の通信プロトコルと物理層に相当するシステム回路の双方にわたる垂直統合設計が必要である。

近年のセンサネットワークに関する国内外の研究発表を見ると、プロトコル開発に関する研究においても、実際にセンサノードを用いた実証実験をおこなう研究が主流にな

りつつある [54][55]. しかしこれらの研究でも, センサノードの回路・アーキテクチャレベルまでの検討はおこなわれていない. また, センサノードの設計研究も多く発表されているが, 回路レベル, あるいはノード単体での評価しかおこなわれていない研究がほとんどである. したがってネットワークを構築して評価をおこなった場合, 最新のセンサノード [56][57] でも平均消費電力は数百  $\mu$ W に達する. センサを外付けすることを考えると, この消費電力ではまだ十分に低いとは言えない. 例えば, 市販のボタン電池で数年間動作させることを想定した場合,  $10\mu$ W 程度の消費電力達成が必要である.

### 5.3 MAC プロトコルの開発

本節では, 本研究でセンサノードに実装した MAC プロトコルについて述べる. 本プロジェクトではワイヤレスセンサネットワークのための MAC プロトコルとして間欠起動型 MAC である LPL (Low Power Listening) に着目し, 時刻同期を用いて LPL を改良した I-MAC (Isochronous-MAC) を提案している [33]. アイドル時に通信モジュールの起動/停止を周期的におこなう間欠起動型 MAC の中でも, LPL はアイドル時の消費電力削減効果に優れている. しかし, LPL にはネットワークシステム全体の最適なプリアンブル長の設定が困難であるという問題がある. そこで I-MAC では LPL を基に, 第3章で述べた時刻同期技術を利用し, センサノード間の同期をおこなう. I-MAC は受信ノードの起動時刻が予想できるため, LPL に比べてプリアンブル長を短くでき, 消費電力を削減できる. また本研究では, I-MAC への RTS/CTS 機能追加を検討し, データ収集型アプリケーションにおいて RTS が受信電力削減に有効であることを示す.

#### 5.3.1 従来手法 (Low Power Listening) の課題

LPL(Low Power Listening)[14][15] では各ノードは受信機の起動, 停止を一定の周期で繰り返し, 起動している間にデータを送受信する (図 5.9). LPL におけるアイドル時の動作の特徴は各ノードが定期的にキャリアセンスをおこなう点である. キャリアセンスのために通信モジュールが起動している期間は空間の電波の有無を確認するのに必要な時間であり, 非常に短い. Sender は Receiver と通信するためにプリアンブルとよばれる信号をデータ送信前に送信する. プリアンブルはアドレスなどのデータを持っておらず, 隣接ノードをキャリアセンスによって起動させるために用いられる. そのため, キャリアセンスの間隔時間以上のプリアンブルを送信し続ける必要がある. キャリアセ

ンスによってプリアンプルを検知したノードはデータ受信に備えて受信機を起動させる。Receiver がデータを正しく受信した場合、Sender に ACK を返信した後アイドル状態に遷移する。

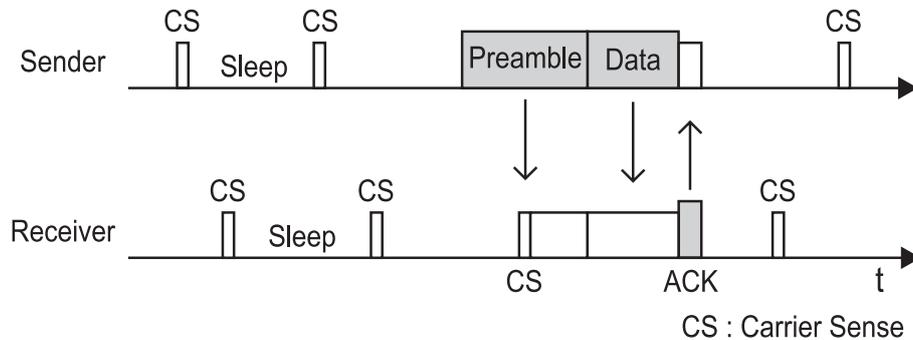


図 5.9 LPL による送受信動作

LPL は定期的なキャリアセンスを特徴としているため、アイドルリスニングによる消費エネルギーを削減することができる。起動する期間が短いため消費エネルギー削減効果がより高い。

LPL の欠点としてオーバヒアリング、プリアンプルによる消費エネルギーオーバーヘッドが挙げられる。プリアンプルによって隣接ノードをすべて起動させた後にデータを続けて送信するため、Receiver 以外のノードもデータを受信してしまい、オーバヒアリングによるエネルギー浪費が発生する。また、LPL ではノード間で起動するタイミングが一致していないため、プリアンプルの長さを起動周期（キャリアセンス間隔時間）以上にする必要がある。したがって、起動周期を長くするほどプリアンプルも長くなり、通信時の電力消費が増大する。

これらの問題を改善するために、我々は Isochronous-MAC(以下、I-MAC) を提案した [33].

### 5.3.2 Isochronous-MAC プロトコルの概要

図 5.10 は I-MAC におけるパケットの送受信の様子を示している。

I-MAC では長波帯標準電波（電波時計）や時刻同期プロトコルなどを用いて全センサノードの時刻を同期させ、各ノードの起動タイミングを一致させる。したがって、送信ノードは受信ノードの起動タイミングを高精度に予測することが可能となる。送信ノードは受信ノードが起動するタイミングに合わせてプリアンプルを送信することによ

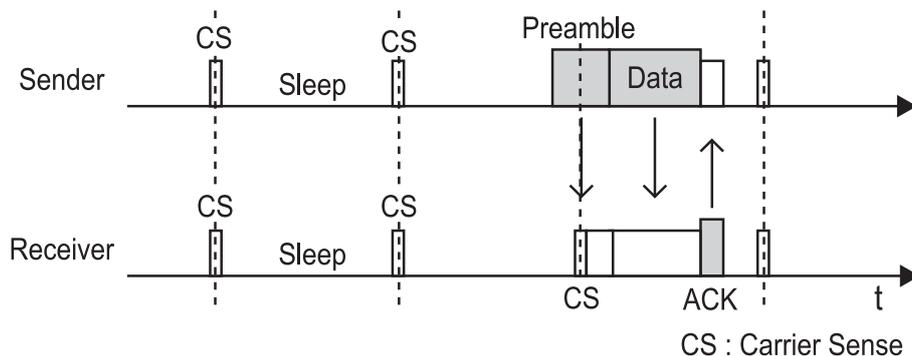


図 5.10 I-MAC による送受信動作

て、短いプリアンプルで通信することができる。このため、I-MAC では LPL よりも通信時の電力消費を削減することが可能である。また、I-MAC ではプリアンプルの長さは時刻同期の精度に依存し、起動周期に依存しないため、起動周期を長くしても電力消費は増大しない (図 5.11,  $D$  は水晶発振器の日差)。

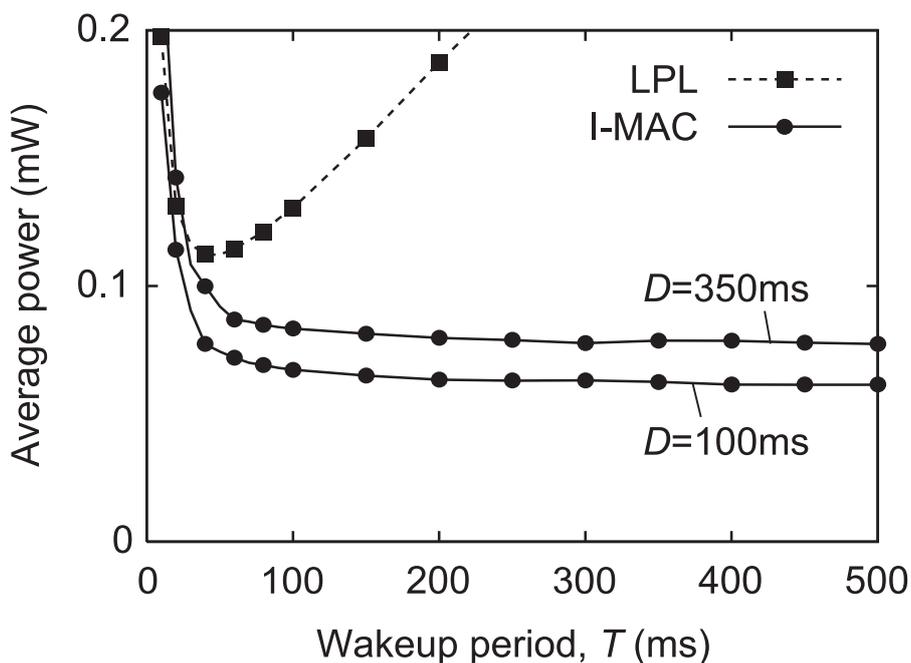


図 5.11 起動周期と平均消費電力の関係

### 5.3.3 Isochronous-MAC のプリアンプル長

I-MAC のプリアンプル長は各ノードの時刻のずれを考慮して決定する必要がある。時刻のずれは水晶発振器の性能に依存し、高性能な水晶発振器を用いることでプリアン

ブル長を短くすることが可能である。また1日の時刻補正の回数は50回程度で十分にプリアンブル長を短くすることが可能である。

キャリアセンスは各ノードが保持している時刻を基準におこなう。時刻精度は各ノードの水晶発振器の精度に依存し、水晶発振器の発振周波数はノード間で僅かに異なる。また時間の経過と共に発振周波数が温度や電圧の影響をうけて変化する。この現象はドリフトとよばれる。時刻補正直後における各ノードの時刻のずれは時刻補正時の誤差に依存する。一方、時刻補正直前では各ノードのドリフトにより時刻のずれが発生していることが考えられる。したがって通信の信頼性を保証するため、時刻補正直前の時刻のずれを考慮に入れる必要がある。

水晶発振器の精度を表すパラメータとして日差がある。日差は1日あたりの時刻のずれである。また1ヶ月あたりの時刻のずれを表す月差というパラメータも存在する。本論文では日差からプリアンブル長を決定することを試みる。ここで日差を $\pm D$ とし、時間の経過とともに線形に時刻がずれると仮定する。また1日あたりの時刻補正の回数を $C$ 、補正時の誤差を $F$ とする。ノード間の相対的なキャリアセンス時間のずれは、最大で $2D = C + 2F$ である。したがって Sender は自身のキャリアセンス時間を中心に $4D = C + 4F + T_{on}$ の長さのプリアンブルを送信する必要がある(図5.12)。水晶発振器の日差は平均的に $\pm 1s$ 以下が多い。

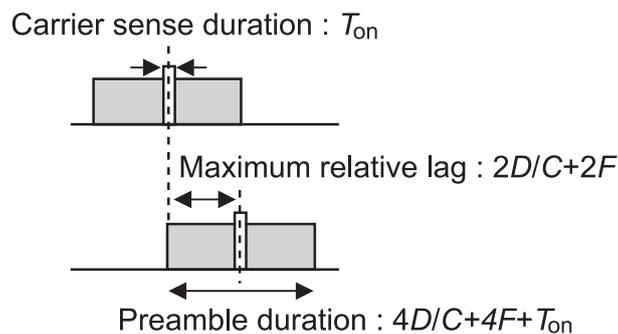


図 5.12 時刻ずれとプリアンブル長の関係

また温度補償水晶発振器 (TG-3530SA; EPSON TOYOCOM[58]) が開発されており、月差 $\pm 10s$  (日差 $\pm 350ms$ ) 程度の性能を達成している。さらに、第3章で述べたドリフトを予測して補正するアルゴリズムを利用することで、時刻のずれをより小さく抑えることが可能である。

図 5.13 は 1 日の時刻補正回数  $C$  とプリアンブル長の関係を示している。時刻補正時の誤差  $F$  は  $1.5\mu\text{s}$ 、 $T_{on}$  は  $50\mu\text{s}$  である。プリアンブル長は時刻補正回数と反比例の関係にあり、1 日に 50 回程の時刻補正で十分プリアンブル長を短くできることがわかる。また時刻のずれを小さくすればするほどプリアンブル長を短くできる。

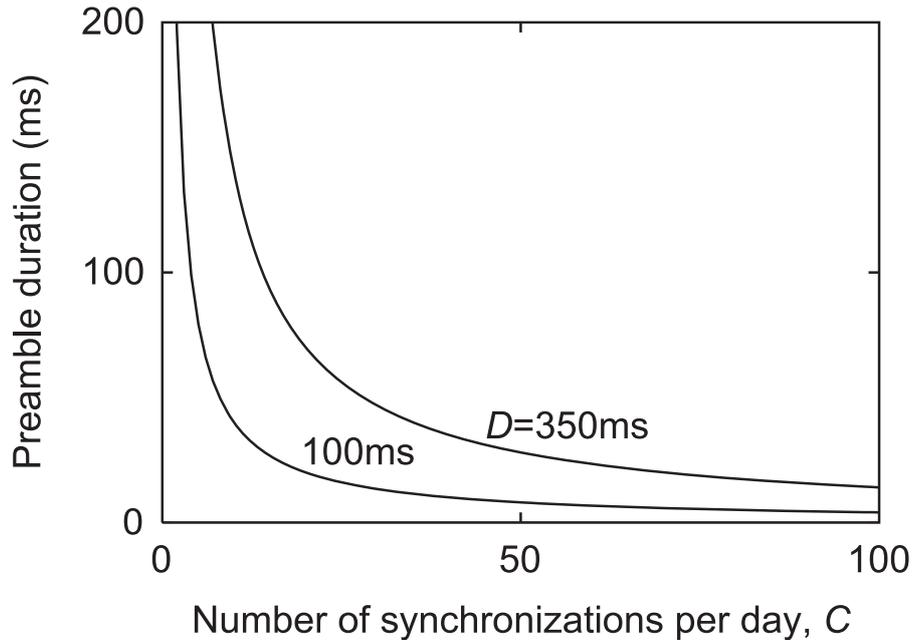


図 5.13 プリアンブル長と 1 日の同期回数  $C$  の関係 ( $D = 350$  and  $100\text{ms}$ ).

### 5.3.4 Isochronous-MAC の衝突回避

I-MAC は各ノードの送信タイミングが同じであるため、衝突回避機能を実装する必要がある。衝突を回避するためにプリアンブル送信前にキャリアセンスをおこなう。また、プリアンブルの送信開始タイミングをランダムにする (図 5.14)。送信前にキャリアセンスをおこなう期間はアイドル時のキャリアセンス期間と等しい。まず、Sender は  $[0, CW_{max}]$  からランダムに整数値  $CW$  を選択する。 $CW$  は Contention Window サイズ、 $CW_{max}$  は Contention Window サイズの最大値である。Sender は  $CW \times t_{slot}$  時間だけ早くプリアンブルを送信することにより、各 Sender のプリアンブル送信開始タイミングをずらす。 $t_{slot}$  はスロット時間であり、ここでは  $T_{on}$  とする。複数 Sender のデータ送信タイミングが重なった場合でも、最初に送信を開始した他の Sender のプリアンブルをキャリアセンスで検知することにより、プリアンブルの送信を延期して衝突を回避する

ことが可能である。

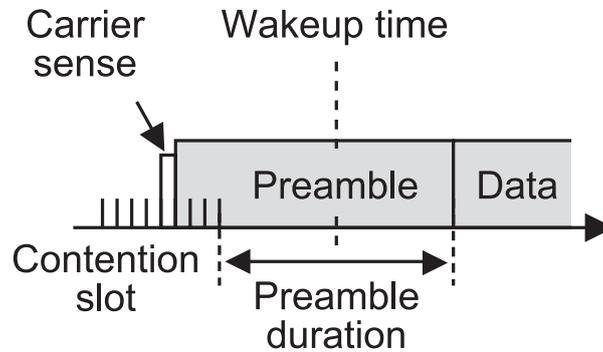


図 5.14 I-MAC における衝突回避

### 5.3.5 RTS/CTS による Isochronous-MAC の改良

本節では、I-MAC への RTS/CTS 機能追加について検討する。第 2 章で述べたとおり、RTS/CTS を用いることによって隠れ端末問題を回避することができるが、制御パケットによる送受信の消費エネルギーが増大することになる。しかし、RTS/CTS には隠れ端末問題の回避以外にも、オーバーヒアリングによる消費電力オーバーヘッドを抑制できるという利点がある。

例えば Sender と Receiver の周辺に多数の隣接ノードが存在するような状況を仮定する。このとき通常の I-MAC では Receiver 以外の隣接ノードもすべての通信パケットを受信してしまう。これに対し、RTS/CTS を用いた場合、RTS を受信し終わった段階で隣接ノードは無線通信回路を OFF にできるため、すべての通信パケットを受信することはない。通常 RTS はデータパケットに比べて十分小さいため、オーバーヒアリングによる消費電力が削減できる。本研究では平均隣接ノード数が 10 個程度の環境でのデータ収集型のアプリケーションを想定しており、制御パケットによる Sender と Receiver の送受信電力オーバーヘッドよりも、オーバーヒアリング削減による受信電力削減量が大きいと考えられる。特に、送受信データ量の最も多い、ベースステーション周辺ノードの受信電力削減が期待できる。

図 5.15 は RTS/CTS の ON/OFF による無線機の平均消費電力の変化を、データ転送レートを変えて評価したシミュレーション結果である。100m×100m の領域に送信半径 20m のノード 100 個をランダム配置し、全ノードから中央のベースステーションまで

データを収集するシミュレーションをおこなった。データ転送レートに関係なく、RTSによるオーバーヒアリングの削減によって消費電力が60%程度削減されている。

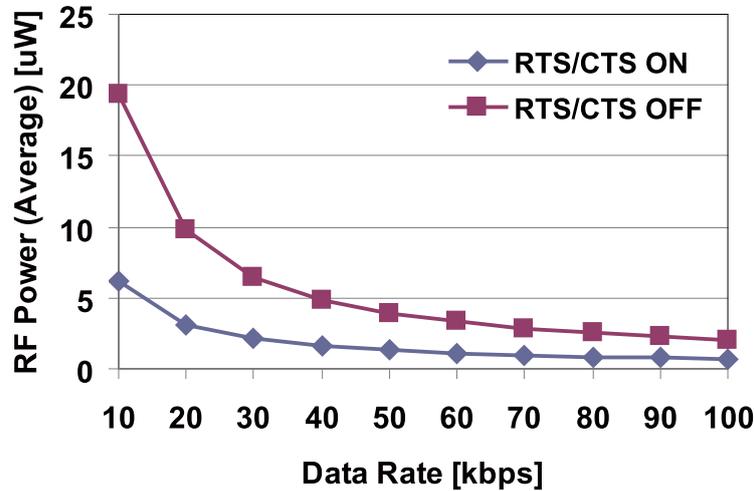


図 5.15 RTS/CTS の有無による伝送レートと無線機消費電力の関係

## 5.4 MAC シーケンサの設計

本節では前節で提案した間欠動作メディアアクセス制御の実装アプローチと、提案センサノードのプロセッサ構成について述べる。

通常のセンサノードでは、データリンク層以上のプロトコルはノードに搭載した汎用マイクロプロセッサによって実現される。しかし、ワイヤレスセンサネットワークは低消費電力化のために通信速度を遅く（～数百 kHz）設計することが多いのに対し、マイクロプロセッサはネットワーク層の処理をおこなうために数 MHz で動作する。したがって、マイクロプロセッサによって MAC プロトコルを実現した場合、データ送受信中の消費電力が非常に大きくなってしまふ。これに対して本研究では、MAC プロトコルが通信発生ごとに必ず決まった手順を実行することに着目し、ステートマシンによって MAC 層の通信処理を専用ハードウェア化した MAC シーケンサの開発によって消費電力の削減をおこなう。汎用マイクロプロセッサの機能をネットワーク層以上に制限することにより、その待機電力を大幅に削減できる。

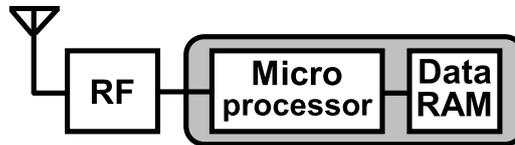
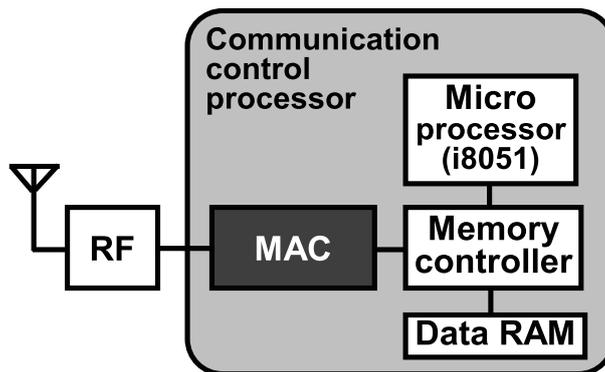
**(a) Conventional****(b) Proposal**

図 5.16 通信制御用プロセッサの構成

**5.4.1 通信制御用プロセッサの構成**

図 5.16 (a) に従来のセンサノードにおけるマイクロコントローラ周辺のブロック図を示す。無線機の通信速度が 10~100kbps であるのに対し、動作周波数が数 MHz の汎用マイクロプロセッサが送受信データをすべて処理するため、非常に無駄が多い。汎用マイクロプロセッサの消費電力は小さな物でも数 mW であり、これが通信時に常に稼働していることになる。

これに対し提案センサノード (図 5.16 (b)) では、MAC 層の通信処理を制御するモジュール (MAC シーケンサ) をハードワイアードによって実現し、汎用マイクロプロセッサとのマルチコア構成とした。MAC プロトコルの動作を状態遷移に見立て、ステートマシンとして MAC シーケンサを実現する。プリアンブル長および起動周期を入力することでこれらの値を変更することができるよう、柔軟性も持たせる。さらに、時刻同期機能を外部タイマに持たせることにより、同期式間欠起動型 MAC である I-MAC として動作する。また、通信時に汎用マイクロプロセッサの電源を切ることで消費電力を削減する。

### 5.4.2 MAC シーケンサの構成

本研究で定義した I-MAC の通信スキームを図 5.17 に示す。ここで、Preamble は連続した”1”のデータ、STX (開始フラグ) と ETX (開始フラグ) は”01111110”の 8bit、HEADER は送受信ノード ID (各 16bit) と制御フラグ (8bit) とデータ長 (8bit) と CRC (16bit)、ACK は CRC (16bit) である。データ長は最小 64bit, 最大 1KByte である。

図 5.18 は図 5.17 を実現する MAC シーケンサのブロック図である。MAC シーケンサは 4 ビットの状態マシン、デコーダ、送受信バッファ、CRC ユニット、24 ビットカウンタ、初期化ブロックで構成される。初期化ブロックは電源復帰時の状態初期化に用いる。

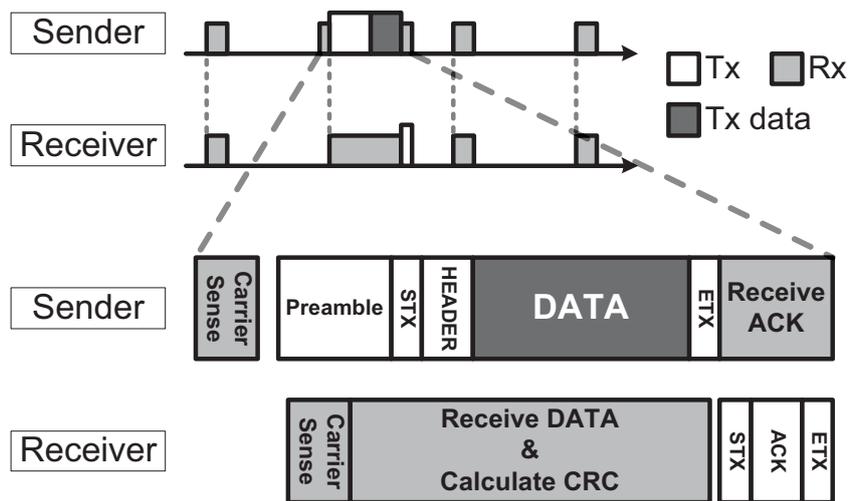


図 5.17 I-MAC の通信スキーム

また、MAC シーケンサへの供給クロックは 1MHz であるが、送受信バッファと CRC ユニットはクロックゲーティングにより伝送レートに合わせたクロックで動作する。これは、通信系のクロック (10~100kHz) が MAC シーケンサの動作クロック (1MHz) のわずか数 % であり、無視できない電力オーバーヘッドが存在するためである。

そこで、無線機からの受信データを処理する Baseband 回路と MAC シーケンサの間に、伝送レートに合わせてクロックを生成する回路を設けた。図 5.19 は通信系と MAC とのクロック差を吸収するモジュールの回路構成である。CLK Digital (1MHz) を基準に CLK RF (10~100kHz) を同期する。CLK RF の立ち下がりから CLK Digital にお

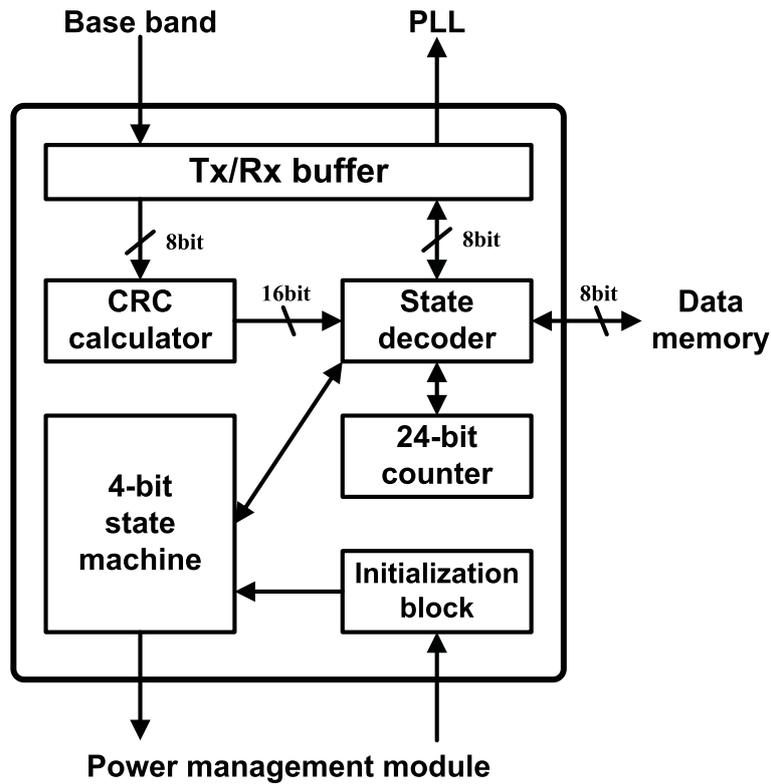


図 5.18 I-MAC シーケンサのブロック図

いて2クロック後、1クロックの間 RF Enable 信号が立ちあがる (図 5.20). レジスタを3段通すことで、この回路は2つのクロックが同時に変化したときもメタステーブルが発生しないように構成している. この RF Enable を MAC シーケンサ内部の送受信バッファと CRC ユニットへのクロックとして供給する.

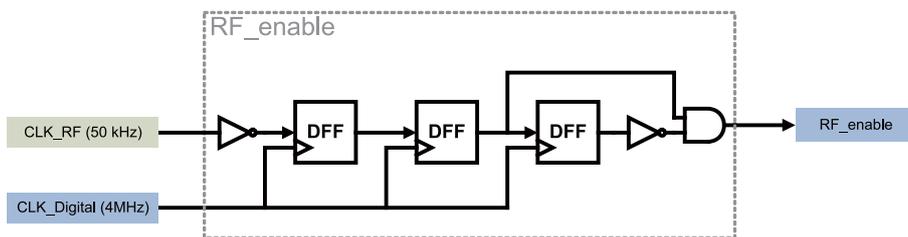


図 5.19 非同期クロック対策モジュールブロック図

### 5.4.3 ステートマシンの構成

MAC シーケンサのステートマシンの構成について述べる.

図 5.21 は受信時の状態遷移図, 図 5.22 は送信時の状態遷移図を示している. ここ

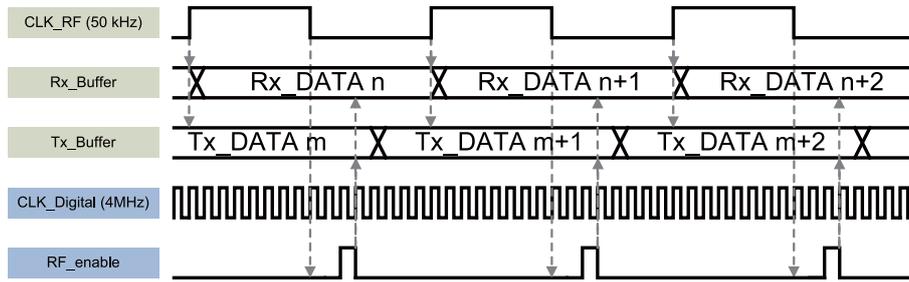


図 5.20 非同期クロック対策モジュールタイミング図

で注意すべきは、受信時に送信先ノード ID を受信した時点でスリープの判定をして状態遷移している点である。前節で述べた RTS/CTS によるオーバーヒアリング削減効果を考慮し、受信と同時にヘッダの解析をおこなうよう MAC シーケンサを構成した。RTS/CTS そのものを実装したわけではないが、単純にオーバーヒアリング削減による受信電力の削減だけを考えて場合、この実装で十分である。

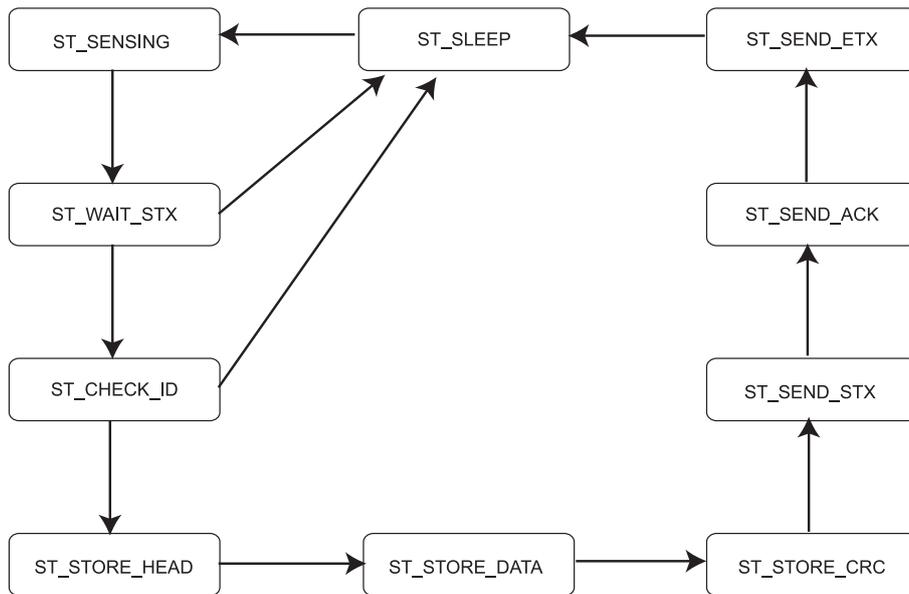


図 5.21 I-MAC の受信時状態遷移

次に、各状態における動作の詳細を述べる。

**ST SLEEP (STATE=0000)** スリープ状態。

**ST SENSING (STATE=0001)** センシング状態。RSSI が 1 のときは受信状態となり ST WAIT STX へ遷移する。RSSI が 0 かつ TX FLAG が 1 のときは送信状態とな

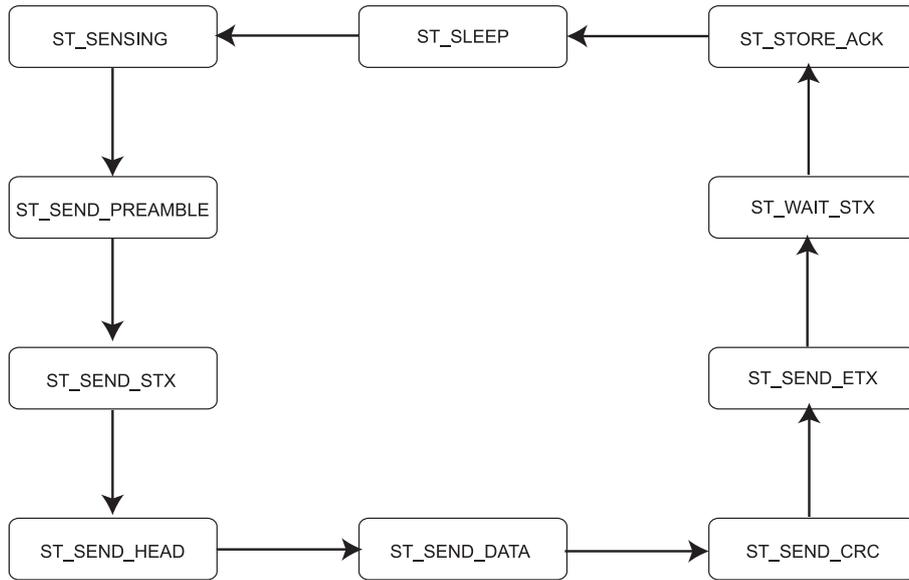


図 5.22 I-MAC の送信時状態遷移

り ST SEND PREAMBLE へ遷移する。RSSI が 0 かつ TX FLAG が 0 のときはスリープ状態となり ST SLEEP へ遷移する。

**ST STORE DATA (STATE=0010)** データ受信状態。ヘッダのサイズ情報の bit 分受信する。同時に CRC を計算する。受信完了後 ST STORE CRC へ遷移する。

**ST STORE CRC (STATE=0011)** CRC 受信状態。16bit 分受信した後 ST STORE DATA で計算した CRC と比較する。CRC 比較後 ST SEND STX へ遷移する。

**ST STORE ACK (STATE=0100)** ACK 受信状態。16bit 分受信した後 ST SEND DATA で計算した CRC と比較する。比較結果が一致した場合は、メモリの送信キューからパケットを削除する。また、比較結果が不一致の場合は再送回数を更新する。CRC 比較後 ST SLEEP へ遷移する。

**ST WAIT STX (STATE=0101)** STX 待ち状態。“01111110”のパターンを受信するまでカウンタをアップ。“01111110”のパターンを受信した後、ST CHECK ID へ遷移する。カウンタが閾値を超えた場合は ST SLEEP へ遷移する。

**ST STORE HEAD (STATE=0110)** ヘッダ受信状態。32bit 分受信する。先頭 16bit を送信元 ID としてメモリに格納。さらに後半 8bit をパケットサイズ情報としてメモリに格納する。受信後 ST STORE DATA へ遷移する。

**ST CHECK ID (STATE=0111)** ID チェック状態。16bit 分受信する。MY ID と比

較し一致した場合は ST STORE HEAD へ遷移する。また、不一致の場合は ST SLEEP へ遷移する。

**ST SEND ACK (STATE=1000)** ACK 送信状態。ST SEND DATA で計算した CRC16bit 分を送信する。送信完了後 ST SEND ETX へ遷移する。

**ST SEND PREAMBLE (STATE=1001)** プリアンブル送信状態。PREAMBLE 分だけ”1”を送信する。送信終了後 ST SEND STX へ遷移する。

**ST SEND STX (STATE=1010)** STX 送信状態。”01111110”の 8bit を送信する。データ送信状態では同時にヘッダを作成する。送信終了後 ST SEND HEAD へ遷移する。

**ST SEND CRC (STATE=1011)** CRC 送信状態。ST SEND DATA で計算した CRC16bit を送信する。送信終了後 ST SEND ETX へ遷移する。

**ST SEND DATA (STATE=1100)** データ送信状態。メモリに格納されているデータサイズ分だけデータを送信する。データは 8bit 送信ごとに 8bit ずつメモリから読み出しする。送信終了後 ST SEND CRC へ遷移する。

**ST SEND ETX (STATE=1101)** ETX 送信状態。”01111110”の 8bit を送信する。送信終了後 ST SLEEP へ遷移する。

**ST SEND HEAD (STATE=1110)** ヘッダ送信状態。48bit 分送信する。データ送信のための最初の 8bit をメモリから読み出ししておく。送信終了後 ST SEND DATA へ遷移する。

**ST INIT (STATE=1111)** 初期化状態。8051・PLL の初期化をおこなう。

## 5.5 1-chip システム LSI の設計

本節では、前節で提案した MAC シーケンサを中核とした、1-chip システム LSI としてのセンサノード設計について述べる。プロセッサ、メモリ、無線通信といった回路をすべて 1 つの LSI に集積することにより、モジュール間のデータ転送などの電力オーバーヘッドを削減できる。また、部品点数も抑えられるため、製造コストの削減にもつながる。

### 5.5.1 センサノードのアーキテクチャ

提案するセンサノードは，以下のモジュールによって構成されている。

- LNA (Low Noise Amplifier)
- PA (Power Amplifier)
- PLL (Phase-locked loop)
- ADC (Analog to Digital Converter)
- Decimation Filter
- BB (Baseband Processor)
- MAC Processor
- Memory Controller
- Data RAM
- i8051：汎用マイクロプロセッサ
- Program ROM：i8051 用プログラム ROM
- Power Management：電源管理レジスタ
- Data Register：電源遮断時のデータ待避用レジスタ
- Timer
- Clock Generator：32MHz のクロックを分周（デジタル回路用）
- Crystal Oscillator：水晶発振器（32MHz と 32.768kHz の 2 系統）

図 5.23 にセンサノード全体の詳細なブロック図を示す。

データの受信時には，まずアンテナから受信された信号が LNA を通った後 ADC でデジタルデータに変換される。データは Decimation Filter を通して Baseband Processor で復調され，MAC シーケンサが MAC 層の処理をおこない，Memory コントローラから Data RAM に保存される。

データ送信時には Memory コントローラを通して MAC シーケンサが Data RAM に保存されている送信データを読み出し，データに合わせて PLL を動作させ，PA を通して外付けアンテナから信号を送信する。

また，スケジューリングやルーティングなどの処理は汎用マイクロプロセッサがおこなう。Memory コントローラを通して汎用マイクロプロセッサ i8051 がデータを読み出

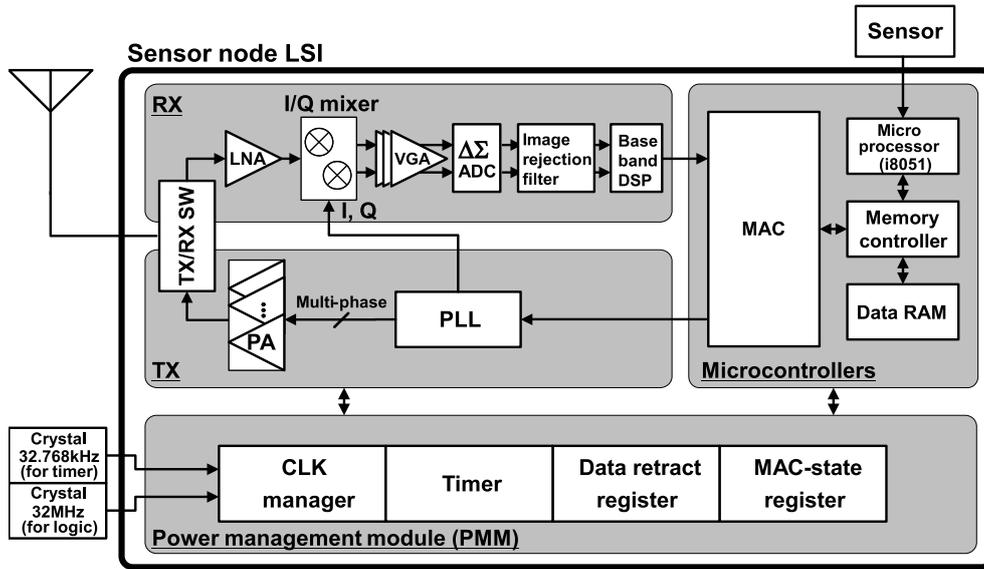


図 5.23 1-chip センサノード LSI のブロック図

し、ネットワーク層以上の処理をおこなう。

### 5.5.2 低消費電力化のための工夫

本研究では、センサノード全体の消費電力を削減するために、センサノードの各モジュールごとのパワーゲーティングとクロックゲーティングによる徹底した電源管理を導入した。センサノードは常にすべてのモジュールを動作させている必要はなく、動作状態に合わせて不必要なモジュールの電源管理をおこなうことで消費電力を大きく削減できる。

パワーゲーティングとは、論理回路の電源側（または接地側）に電源スイッチを挿入し、不要な回路ブロックのスイッチをオフにしてリーク電流を遮断する技術である（図 5.24 (a)）。電源スイッチのサイズが大きくなりやすい、順序回路はデータが保持できない、遮断される回路からの出力がフローティングになるのでバスキーパ（図 5.25）などによる保護が必要、といった課題がある。

クロックゲーティングとは順序回路へのクロック供給を遮断し、スイッチング電力を削減する技術である。クロックラインと Enable 信号の AND をとり、それを対象となる回路へクロックとして供給する（図 5.24 (b)）。このときメタステーブル（フリップフロップのクロック立ち上がりとデータ変化が重なり、発振状態になること）のようなタイミング上の問題を回避するために、Enable 信号をもとのクロック信号でラッチする必

要がある。クロックゲーティングではリーク電流を減らすことはできないが、スイッチング電力が支配的な回路には効果的である。また、データの保持が必要な順序回路に対しても適用可能である。

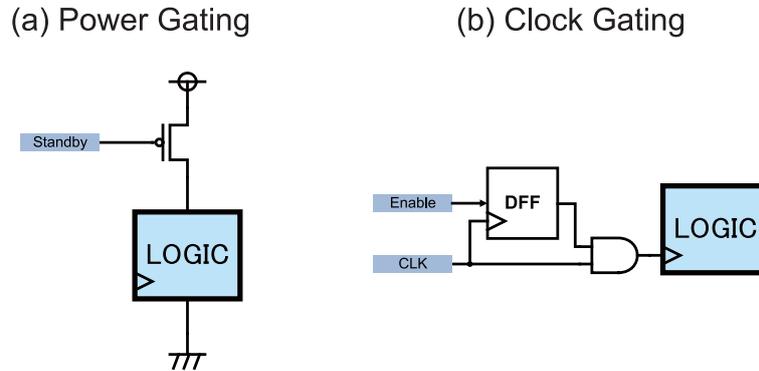


図 5.24 パワーゲーティングとクロックゲーティング

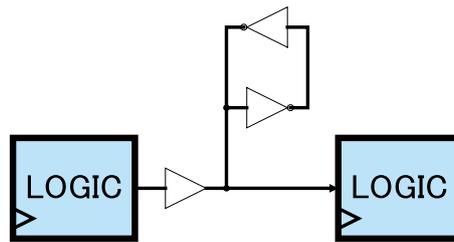


図 5.25 バスキーパー

これらの技術を組み合わせ、センサノードの動作モードに合わせてモジュールごとの電源管理をおこなう。ここで、センサノードの動作モードを以下のように定義する。

**Sleep** データ用メモリの中にデータが無く、周期的にキャリアセンスしているだけの状態。

**Standby** データ用メモリの中にデータがあり、送信待ちの状態。データ用メモリは電源を切ることができない。

**Tx** 無線機を使ってデータを送信している状態。

**Rx** 無線機を使ってデータを受信中状態。

**CPU** マイクロコントローラを使ってデータ処理やスケジューリング、同期処理をおこなっている状態。

この中で最も重要なのは Sleep モードである。ワイヤレスセンサネットワークでは、

ノードは稼働時間のうち 9 割以上がアイドル状態であり、Sleep モードの消費電力は可能な限り小さく抑える必要がある。本研究では、Sleep モードにおいてほとんどのモジュールの電源またはクロックを遮断し、32.768kHz のクロック系統と Timer だけが動作するように設計をおこなった。各動作モードにおけるモジュールごとの状態と消費電力は次節で述べる。

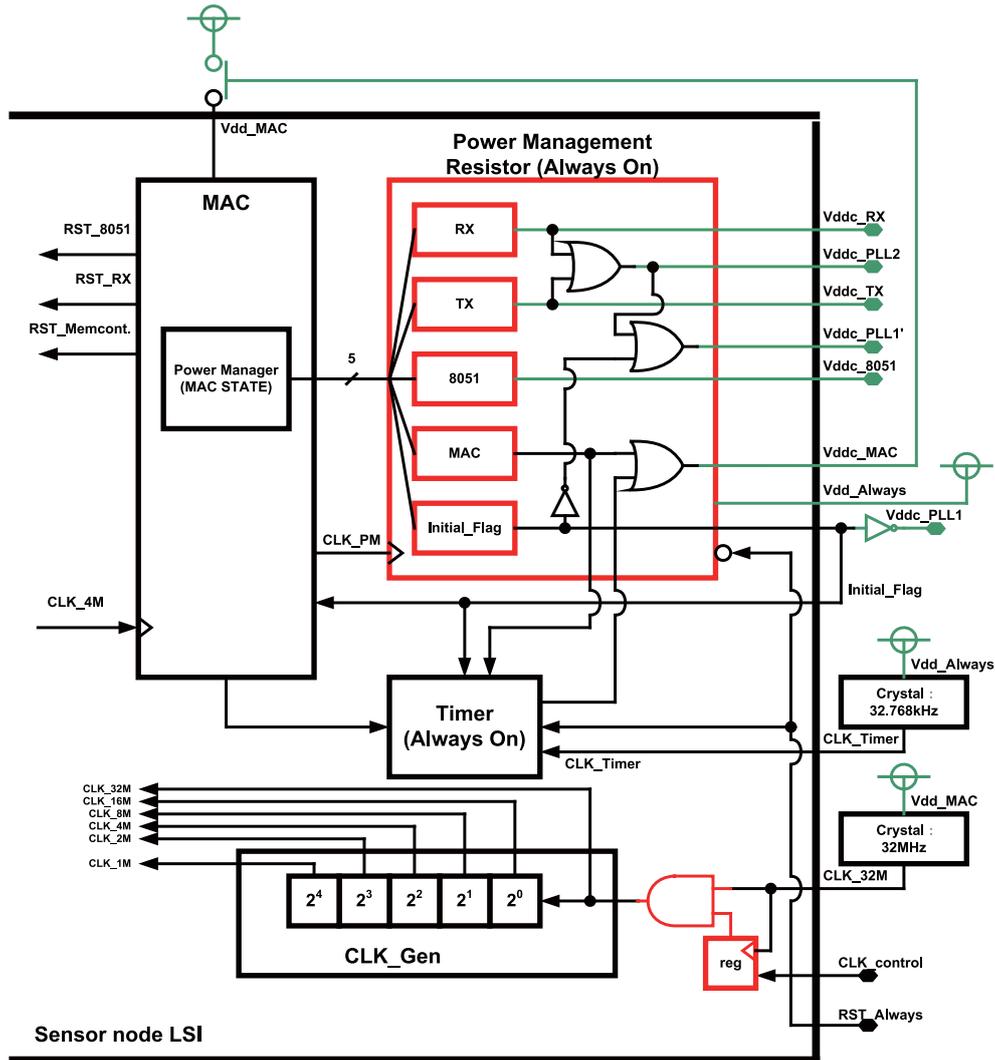


図 5.26 電源管理モジュール

最後に、電源管理システムの設計について述べる。上記の動作モードが切り替わるタイミングは、図 5.21、図 5.22 に示した MAC シーケンサの状態遷移のタイミングと一致している。そこで本研究では、MAC シーケンサへの電源管理システム統合を提案する。MAC シーケンサが状態遷移するごとに各モジュールへの電源管理信号を保持する

レジスタ (Power Management レジスタ) を更新する。Power Management レジスタからの出力は論理を挟んで各モジュールの電源スイッチ, またはクロックイネーブルに接続される (図 5.26)。Sleep モードでは MAC シーケンサの電源も遮断されるが, Power Management レジスタはクロックゲーティングをおこなって状態を保持しており, Timer からの復帰信号によってノードは Rx モード (MAC シーケンサのキャリアセンス状態) に移行できる。

## 5.6 センサノードの VLSI 実装

提案したセンサノードアーキテクチャを基にアナログ・無線回路設計と Verilog RTL によるデジタル論理回路設計をおこない, アナログ・デジタル混載の 1-chip センサノード LSI を実現した (図 5.27)。

表 5.2 に設計したセンサノード LSI のチップ緒言を示す。また, 図 5.28 にレイアウト図を示す。

表 5.2 提案センサノード LSI のチップ諸元

<b>System</b>	
<b>Process</b>	<b>CMOS 180nm</b>
<b>Supply voltage</b>	<b>1.8V</b>
<b>Communication range</b>	<b>0.1-20m</b>
<b>Carrier frequency</b>	<b>433.67 - 434.17MHz</b>
<b>Bitrate</b>	<b>10 - 60kbps</b>
<b>Channel bandwidth</b>	<b>150kHz</b>
<b>Clock</b>	<b>32.768kHz, 32MHz</b>
<b>Frequency tolerance</b>	<b>+/- 20ppm</b>
<b>Modulation</b>	<b>FSK</b>
<b>Transceiver</b>	
<b>Transceiver</b>	<b>Low-IF</b>
<b>Transmitter power</b>	<b>9.00mW</b>
<b>Receiver power</b>	<b>12.05mW</b>
<b>Output power</b>	<b>-4dBm</b>
<b>RX sensibility</b>	<b>-72.8dBm (For 1E-5 BER)</b>

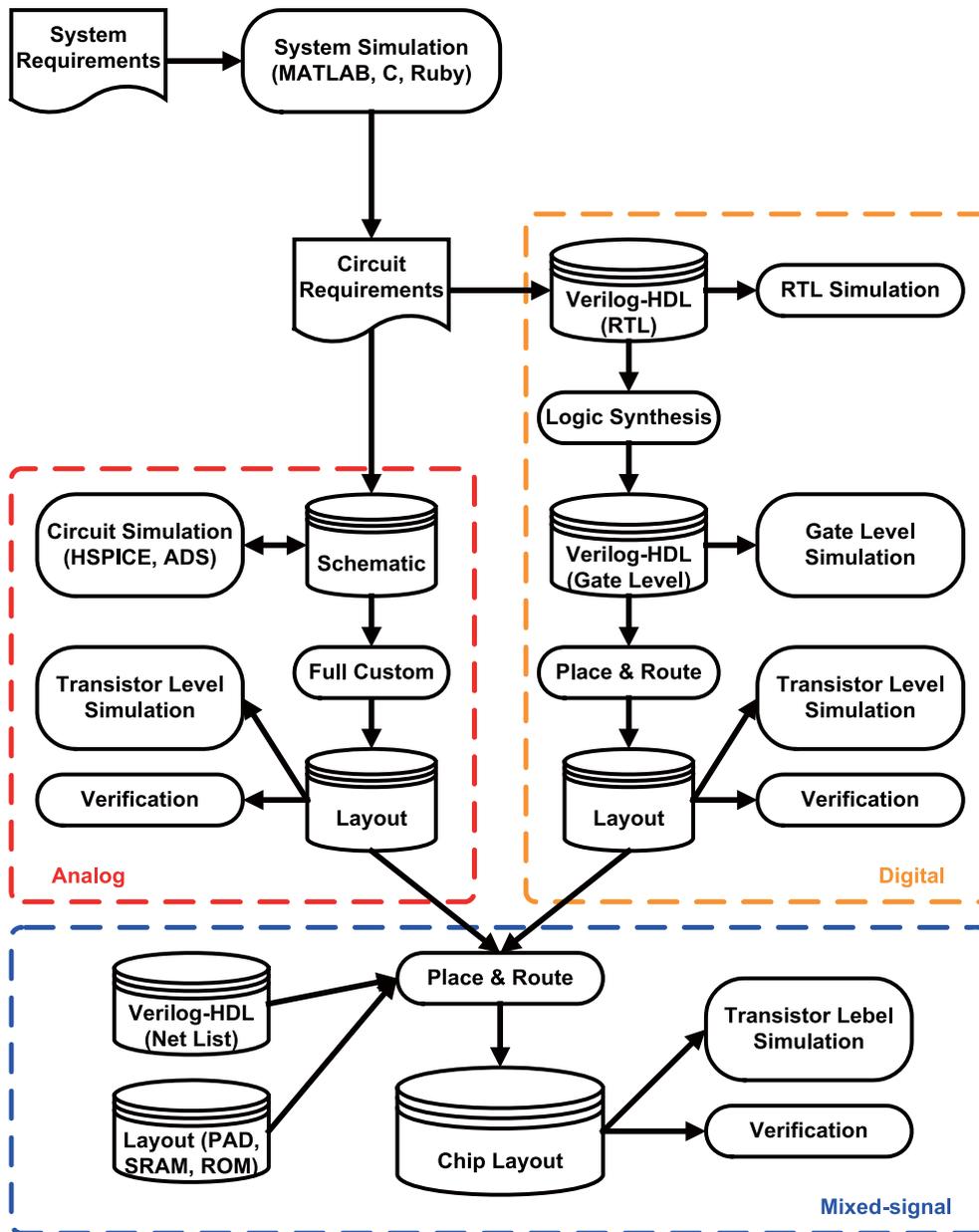


図 5.27 提案センサノード LSI のデザインフロー

## 5.7 消費電力の評価

本節では、設計したセンサノード LSI の性能評価をおこなう。

### 5.7.1 各モジュールの消費電力

表 5.3 に各モジュールごとの消費電力、および動作モードごとの消費電力を示す。

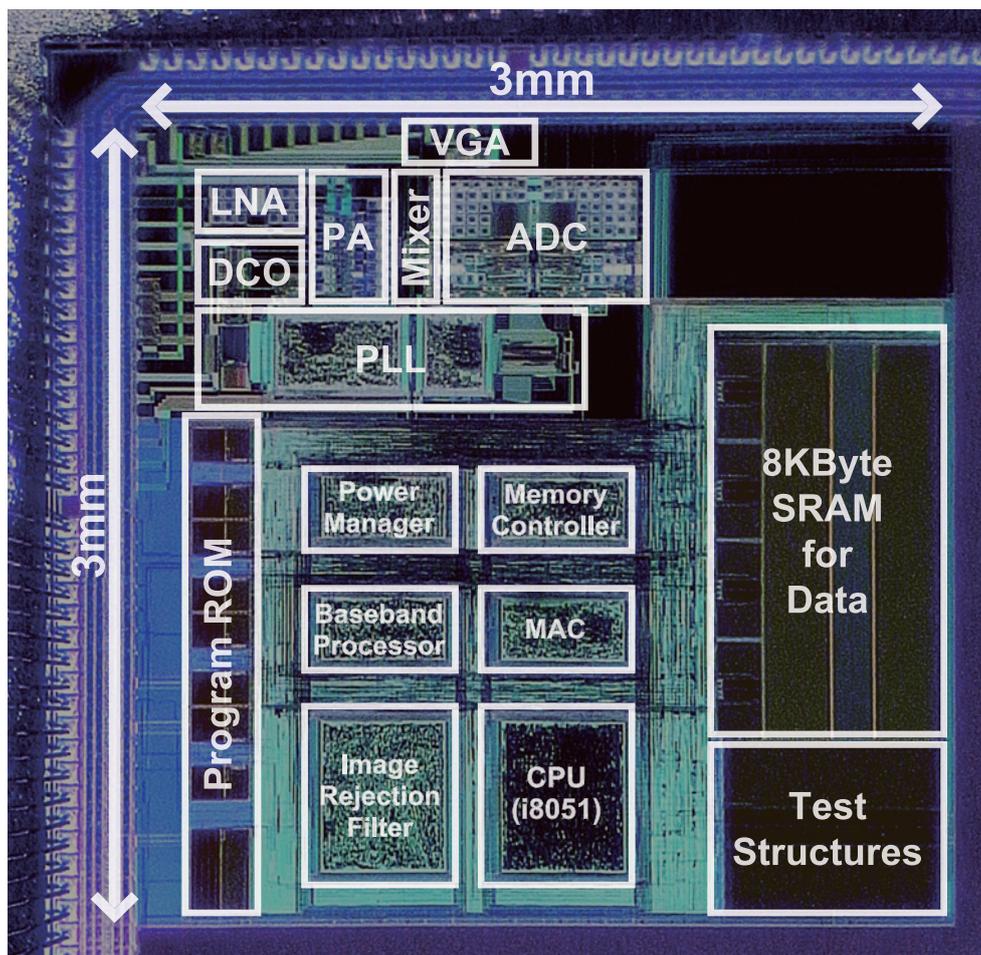


図 5.28 提案センサノード LSI のチップ写真

### 5.7.2 ネットワークシミュレーションによる評価

本節では表 5.3 の結果を用いて、ネットワークシミュレータ QualNet[30] によってデータ収集時の平均消費電力を評価する。

シミュレーション条件を表 5.4 に示す。100m × 100m の領域にデータ転送レート 20kbps, 送信半径 20m のノード 100 個をランダム配置し、全ノードから中央のベースステーションまでマルチホップでデータを集めるシミュレーションをおこなった。I-MAC の起動周期は 100ms, プリアンブル長は 50us である。

まず、シミュレーションによって得られた、センサノードの各状態における平均動作時間を表 5.5 に示す。ここで”Total time”は、ネットワーク内の全ノードからベースステーションへデータを集め終わるまでに掛かった時間である。

表 5.3 消費電力測定結果

	Block power		State of a sensor node			
	Dynamic/Leak		Sleep	TX	RX	Network layer
LNA	1.17mW/-		OFF	OFF	ON	OFF
PA	4.26mW/-		OFF	ON	OFF	OFF
PLL	3.28mW/-		OFF	ON	ON	OFF
VGA	955 $\mu$ W/-		OFF	OFF	ON	OFF
ADC	999 $\mu$ W/-		OFF	OFF	ON	OFF
Image rejection filter	2.75mW/1.02 $\mu$ W		OFF	OFF	ON	OFF
Baseband	1.34mW/1.4 $\mu$ W		OFF	OFF	ON	OFF
MAC	11.7 $\mu$ W/135nW		OFF	ON	ON	ON
Memory controller	14.4 $\mu$ W/88.3nW		clk gating	ON	ON	ON
Data RAM	710 $\mu$ W/11 $\mu$ W		clk gating	read	write	read/write
i8051	787 $\mu$ W/8.2 $\mu$ W		OFF	OFF	OFF	ON
PMM	3.97 $\mu$ W/156nW		ON	ON	ON	ON
Crystal oscillator	32.768kHz	3.6 $\mu$ W/-	ON	ON	ON	ON
	32MHz	2.88mW/-	OFF	ON	ON	ON
Total power			18.8 $\mu$ W	11.16mW	14.12mW	4.41mW

表 5.4 シミュレーション条件

<b>Simulator</b>	<b>QualNet 3.8</b>
<b>Number of simulation</b>	<b>50</b>
<b>Application layer</b>	<b>Data gathering</b>
<b>Network layer</b>	<b>Tiny Diffusion [13]</b>
<b>Location of the base station</b>	<b>Center of the network area</b>
<b>Number of nodes</b>	<b>100</b>
<b>Node deployment</b>	<b>Random</b>
<b>Network size</b>	<b>100 m x 100 m</b>
<b>Transmission range</b>	<b>20 m</b>
<b>Data rate</b>	<b>20 kbps</b>
<b>Data length</b>	<b>6 bytes</b>
<b>Average hop counts</b>	<b>3.07</b>
<b>Wake-up period</b>	<b>100 ms</b>

次に、表 5.5 の結果と表 5.3 に示したセンサノードの状態ごとの消費電力から、センサノードの平均消費電力を計算した。図 5.29 に状態ごと、およびブロックごとのパワー

表 5.5 平均動作時間

<b>Total time [ms]</b>	<b>84767.8</b>
<b>TX active time [ms]</b>	<b>22.5</b>
<b>RX active time [ms]</b>	<b>213.5</b>
<b>Network time [ms]</b>	<b>4.5</b>
<b>Sleep time [ms]</b>	<b>84527.0</b>
<b>Wake-up ratio [%]</b>	<b>0.3</b>

バジェットを示す。提案センサノード LSI の平均消費電力は  $58\mu\text{W}$  であった。

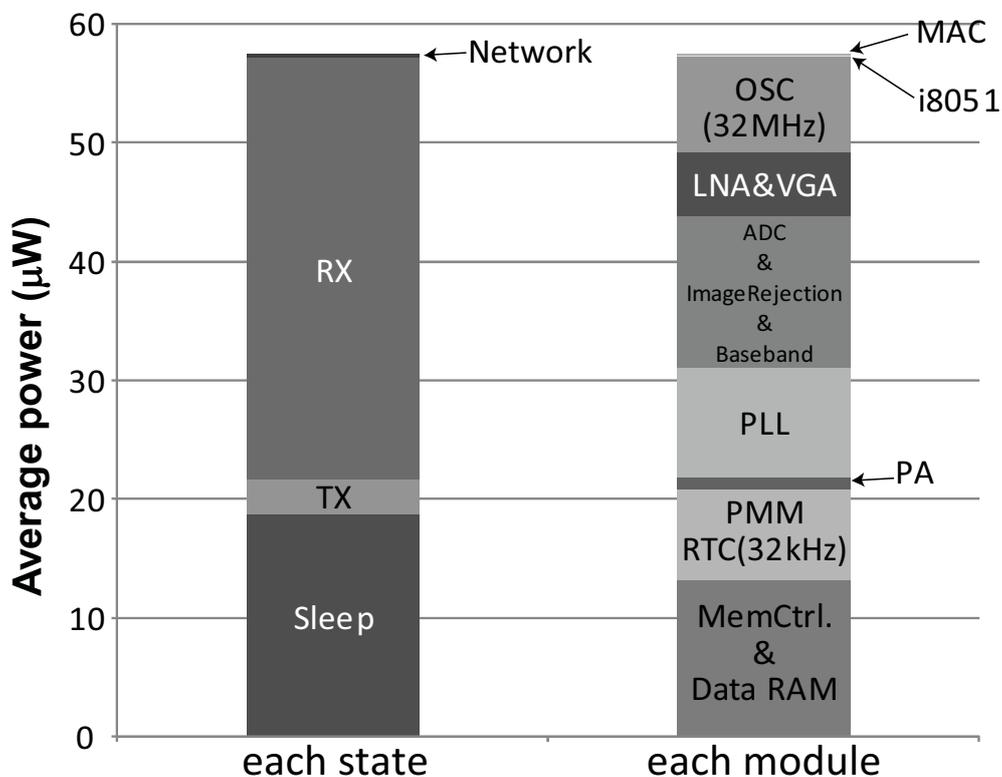


図 5.29 提案センサノード LSI のパワーバジェット

最後に、従来研究である UC Berkley の PicoRadio[56] の消費電力パラメータと MAC プロトコル (TICER) を基に同様のシミュレーションをおこない、本研究で設計したセンサノードとの比較をおこなった。図 5.30 に提案センサノードと UCB の従来センサノードのシミュレーション結果を示す。提案センサノードの平均消費電力が  $58\mu\text{W}$  であったのに対し、UCB の平均消費電力は  $442\mu\text{W}$  であった。UCB が MAC プロトコル

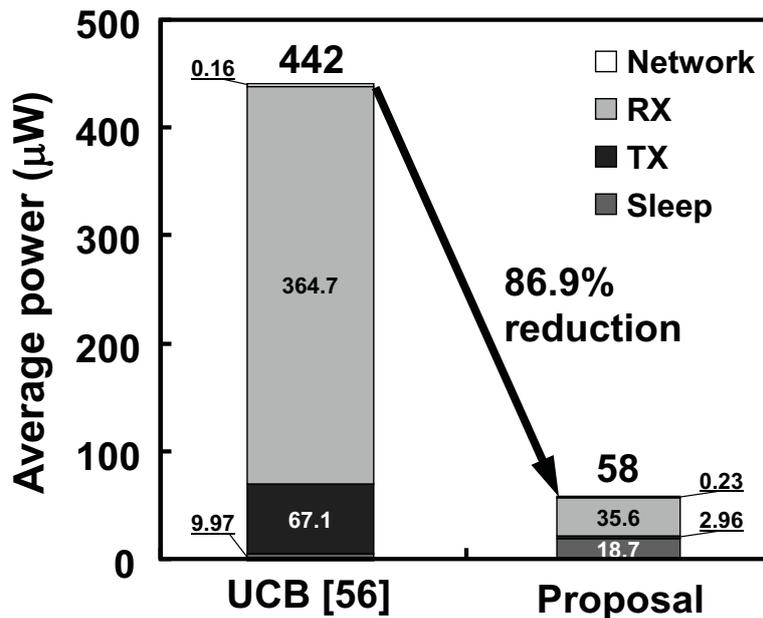


図 5.30 平均消費電力の比較

に TICER を用いているのに対して，提案センサノードでは I-MAC を用いている点がこの消費電力削減に最も大きく寄与していると考えられる。

TICER も I-MAC と同様，間欠動作型の MAC プロトコルである（図 5.31）。I-MAC との違いは，ノード間の同期がおこなわれていない点と，Sender が Receiver が起動するまで一定の間隔で RTS を送信し続ける点である。Receiver が起動時に RTS を受信し，Sender に CTS を返すことでデータの送受信を開始する。このため，TICER は通信を確立させるために RTS を多数送信しなければならない，制御パッケージによる消費エネルギーが大きくなる。また，TICER には制御パッケージによる衝突が発生しやすくなるという欠点が存在する。これに対して I-MAC はノード間の同期によって通信確立までのオーバーヘッドが小さくなるため，提案センサノードは電源管理機構との組み合わせによるアイドルリスニングおよびオーバーヒアリングの削減効果が高い。

## 5.8 結言

ワイヤレスセンサネットワークを実現するためには，センサノードの消費電力を削減し，システムの可用時間を延ばさなければならない。そのためには，ネットワークの各階層に対して個別に最適化するだけでなく，協調して設計すること（垂直統合設計）が必要不可欠である。しかし，通信プロトコルからセンサノードの回路・アーキテクチャ

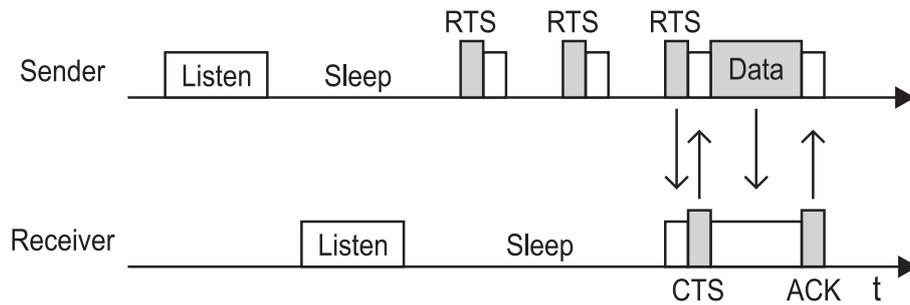


図 5.31 TICER による送受信動作

レベルまでの一貫した低消費電力化をおこなうような研究は、ほとんど成されていなかった。

そこで本研究では、ネットワークの階層構造における中間層である MAC 層を中核とし、上下方向展開による垂直統合設計をおこなった。まず、センサノード向け低消費電力 MAC プロトコルとして LPL を元にした I-MAC を用いることで、アルゴリズム階層で消費電力を削減した。次に、MAC 層の通信処理を専用ハードウェア化した MAC シーケンサと汎用マイクロプロセッサの組み合わせによって低消費電力なセンサノード用マイクロコントローラを実現し、MAC プロトコルと LSI アーキテクチャ間の相互最適化をおこなった。さらに、無線通信回路まで含めて 1-chip システム LSI としてセンサノードを設計し、MAC シーケンサを中核とした電源管理機構によってセンサノード全体の消費電力を削減した。

提案したセンサノードアーキテクチャに基づいてセンサノードの試作し、データ収集時の平均消費電力  $58\mu\text{W}$  を達成した。

## 第6章

# 結論

本論文では大規模センサネットワークにおけるアプリケーション実装技術，ネットワーク技術，回路技術について詳述した。

### 第3章 センサネットワーク技術を用いたマイクロホンアレイの大規模化

従来の大規模マイクアレイにおける演算量や消費電力の問題を解決する技術として，センサネットワーク技術を用いたネットワーク型マイクアレイシステムを提案した。提案システムでは，小規模なマイクアレイを多数配置し，それらの間で有線ネットワークを構築して協調動作をおこなうことで，大規模なマイクアレイ音声処理システムを実現する。また本研究では，ネットワークを介した分散処理に適した音源分離のアルゴリズムを新たに提案した。FPGA ボード SUZAKU を用いて 16ch サブアレイの試作をおこない，3つのサブアレイを用いた実証実験によって提案システムの動作を確認した。

従来のデータ収集のみを目的としたアプリケーションとは異なり，ネットワーク型マイクアレイシステムには「リアルタイム処理」，「センサノード間の時刻同期」，「ネットワーク分散処理」という3つの機能が必要であった。これらの機能は，他のより実用的なアプリケーションでも必須となるものであり，本研究はセンサネットワークの応用分野を拡大するためのテストベッドとしての役割も果たすものである。

### 第4章 待ち時間制御によるブロードキャストアルゴリズムの高効率化

センサネットワークにおいて経路構築などに必要不可欠なブロードキャストに着目し，待ち時間制御によるカウンタベース方式の改良アルゴリズム（RAD-Extension と

HCA-RAD Extension) を提案する。また、ブロードキャストはネットワーク層だけでなくアプリケーション層でも必要とされるプロトコルである。提案方式では、コストの高いアルゴリズムやハードウェアを必要とせず、ノード密度にあわせて再送信確率を制御できるという従来方式の利点を生かしつつ、さらに再送信ノード数を削減することを目的とした。提案方式では、メッセージ受信ごとに待ち時間を増加させることで、各ノードの面積カバー率を改善する。これによってパケット到着率を維持したままで全体の再送信ノード数を削減できることを述べた。さらに、待ち時間のスロット化によって、経路長も考慮して再送信の優先順位を設定するアルゴリズムを提案した。また、シミュレーションによる比較をおこない、従来方式と比較して提案方式では約 10% の再送信ノード数削減効果が得られることを示した。

本研究の提案アルゴリズムは大規模なセンサネットワークのトラフィックを削減することでスケーラビリティを高め、また同時にネットワークシステム全体の長寿命化に貢献するものである。

## 第5章 低消費電力センサノード VLSI の垂直統合設計

ワイヤレスセンサネットワークを実現するためには、センサノードの消費電力を削減し、システムの可用時間を延ばさなければならない。そのためには、ネットワークの各階層に対して個別に最適化するだけでなく、協調して設計すること(垂直統合設計)が必要不可欠である。しかし、通信プロトコルからセンサノードの回路・アーキテクチャレベルまでの一貫した低消費電力化をおこなうような研究は、ほとんど成されていなかった。

そこで本研究では、ネットワークの階層構造における中間層である MAC (Media Access Control) 層を中核とし、上下方向展開による垂直統合設計をおこなった。まず、センサノード向け低消費電力 MAC プロトコルとして LPL (Low Power Listening) を元にした I-MAC (Isochronous-MAC) を用いることで、アルゴリズム階層で消費電力を削減した。次に、MAC 層の通信処理を専用ハードウェア化した MAC シーケンサと汎用マイクロプロセッサの組み合わせによって低消費電力なセンサノード用マイクロコントローラを実現し、MAC プロトコルと LSI アーキテクチャ間の相互最適化をおこなった。さらに、無線通信回路まで含めて 1-chip システム LSI としてセンサノードを設計し、MAC シーケンサを中核とした電源管理機構によってセンサノード全体の消費電力

を削減した。

提案したセンサノードアーキテクチャに基づいてセンサノードの試作し，データ収集時の平均消費電力  $58\mu\text{W}$  を達成した。

以上，本論文ではセンサネットワークにおいて，ネットワーク分散処理に対応するアプリケーション層，通信トラフィックを削減するネットワーク層，センサノードの消費電力を削減する MAC 層と物理層の研究成果について記述した。これらの提案技術によって大規模センサネットワークが実現され，センサネットワークの普及につながる新たなキラーアプリケーションの創出が期待できる。



## 謝辞

本研究の機会を賜り，熱心に御指導頂きました神戸大学大学院システム情報学研究科教授 吉本雅彦先生に深く感謝するとともに厚く御礼申し上げます。

本研究を進めるにあたり，直接の御指導と多くの貴重な御助言を頂きました神戸大学大学院システム情報学研究科准教授 川口博先生，神戸大学大学院システム情報学研究科准教授 太田能先生に心から深謝致します。

本研究において，数々の有益な御助言を賜りました神戸大学大学院システム情報学研究科教授 永田真先生，神戸大学大学院工学研究科教授 沼昌宏先生に深く感謝致します。

本研究において，様々な御助言と適切な御指導を頂きました神戸大学大学院システム情報学研究科教授 玉置久先生，神戸大学大学院システム情報学研究科助手 高木由美先生に心から感謝致します。

同じプロジェクトに属し，本研究に際して有益な議論をおこない，数々の貴重な御意見を頂いた三上真司博士，藤原英弘博士，松田隆志博士，竹内隆博士，青西孝文氏，一圓真澄氏，祇園昭宏氏，野口紘希氏，李赫鍾氏，大竹優氏，小西恵大氏，酒井康晴氏，鶴田嵩氏，黒田光彦氏，藤永剛史氏，嶋井優介氏，高木智也氏，岡頭久氏，何光霽氏，菅原隆伸氏，久賀田耕史氏，寺地陽祐氏，西野允雅氏，奥野圭祐氏，祖田心平氏，宮本優貴氏に深く感謝致します。

本研究に対して有益な御助言，御指導を賜りました川上健太郎博士，宮越純一博士，新居浩二博士，村地勇一郎博士，森田泰弘博士，鈴木利一博士，鈴木弘明博士，松野哲郎博士，松本卓也博士，Augusto Foronda 博士に厚く感謝の意を表します。

本研究に際し，適切なる御指導，御意見を頂いた長井健一氏，濱本真生氏，福山祐貴氏，山本亮氏，飯沼隆弘氏，石原朋和氏，印芳氏，樋口雄飛氏，吉野圭一氏，杉川智氏，伊藤義人氏，大原誠氏，桂章仁氏，小林裕史氏，小山陽平氏，清水則史氏，三木正樹氏，

渡部祐也氏に感謝致します。

研究生生活を共にし、多くのご助言とご協力を頂いた井口友輔氏、上田哲也氏、坂田義典氏、高橋巧至氏、堀上真司氏、村井俊哉氏、村田亘氏、矢倉健一郎氏、李将充氏、奥村俊介氏、上農哲也氏、谷純一氏、中田洋平氏、三浦和夫氏、水野孝祐氏、岡田和也氏、加藤祐介氏、竹川久美子氏、徳永悠氏、安西達哉氏、山林潤氏、竹内幸大氏、山口幸介氏、吉本秀輔氏、寺田正治氏、天下卓郎氏、鍵山祐輝氏、Pham Nguyen Tu Nhy 氏、高原耕平氏、田邊雄也氏、田村一起氏、戸谷洋輔氏、中田健太郎氏、松元隆志氏、渡邊俊介氏、竹内勇介氏、鄭晋旭氏、柳田晃司氏に感謝致します。

研究生生活を温かくサポートしてくださいました研究室秘書の呉恵美氏、坪井彩氏、松岡啓子氏、泉由里絵氏に感謝致します。

本研究の一部は総務省戦略的情報通信研究開発推進制度 (SCOPE)、及び文部科学省科学研究費補助金 (基盤研究 (C) 課題番号 18500052, 若手研究 (B) 課題番号 16760271, 特別研究員奨励費 課題番号 21000333) のもとに実施したものであり、ここに謝意を表します。

本研究の一部は株式会社半導体理工学研究センター (STARC) の協力により行われたものであり、ここに謝意を表します。

本研究の一部は東京大学大規模集積システム設計教育研究センターを通し、シノプシス株式会社、日本ケイデンス株式会社、メンター株式会社、アジレント・テクノロジー株式会社の協力で行われたものであり、ここに謝意を表します。

最後に、私の研究生生活をささえてくれた家族に心から感謝致します。

## 参考文献

- [1] J. M. Kahn, R. Katz, and K. Pister, “Next century challenges: mobile networking for “Smart Dust”,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 271–278 (1999).
- [2] Y. Tamai, S. Kagami, H. Mizoguchi, Y. Amemiya, K. Nagashima, and T. Takano, “Real-time 2 dimensional sound source localization by 128-channel huge microphone array,” in *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pp. 65–70 (2004).
- [3] T. Wakabayashi, K. Takahashi, and H. Iwakura, “Independent component analysis using large microphone array,” *Technical Report of IEICE EA*, Vol. 102, No. 322, pp. 29–34 (2002).
- [4] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan, “The huge microphone array,” *IEEE Concurrency*, Vol. 6, No. 4, pp. 36–46 (1998).
- [5] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan, “The huge microphone array, part 2,” *IEEE Concurrency*, Vol. 7, No. 1, pp. 32–47 (1999).
- [6] E. Weinstein, K. Steele, A. Agarwal, and J. Glass, “Loud: A 1020-node modular microphone array and beamformer for intelligent computing spaces,” *MIT/LCS Technical Memo MIT-LCS-TM-642* (2004).
- [7] J. Benesty, M. M. Sondhi, and Y. Huang, “Springer Handbook of Speech Processing,” Springer (2007).
- [8] R.O. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, Vol. 34, No. 3, pp. 276–280 (1986).
- [9] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proceedings of the 5th symposium on Operating System De-*

- sign and Implementation (OSDI)*, pp. 147–163 (2002).
- [10] S. Ganeriwal, R. Kumar, and M.B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings of the 1st ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 138–149 (2003).
- [11] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 39–49 (2004).
- [12] “<http://www.okisemi.com>,” Oki Electric Industry Company, Limited, (accessed 2010-12-10).
- [13] “<http://suzaku.atmark-techno.com/>,” Atmark Techno, Inc., (accessed 2010-12-10).
- [14] A. El-Hoiydi, “Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Vol. 5, pp. 3418–3423 (2002).
- [15] J. Hill and D. Culler, “Mica: a wireless platform for deeply embedded networks,” *IEEE Micro*, Vol. 22, No. 6, pp. 12–24 (2002).
- [16] J. Heidemann, F. Silva, and D. Estrin, “Matching data dissemination algorithm to application requirements,” in *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SENSYS)*, pp. 218–229 (2003).
- [17] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 151–162 (1999).
- [18] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 194–205 (2002).
- [19] B. Krishnamachari, D. Estrin, and S. Wicker, “Modelling data-centric routing in wireless sensor networks,” in *Proceedings of the 21st IEEE Conference on Computer Communications (INFOCOM)*, Vol. 2, No. 4, pp. 1–11 (2002).
- [20] D. Petrovic, R.C. Shah, K. Ramchandran, and J. Rabaey, “Data funneling: routing with aggregation and compression for wireless sensor networks,” in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*

- (*SNPA*), pp. 156–162 (2003).
- [21] C. Intanagonwiwat, R. Godinvan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 2–16 (2003).
- [22] H. Lim and C. Kim, “Flooding in wireless ad hoc networks,” *Journal of Computer Communications*, Vol. 24, No. 3–4, pp. 353–363 (2001).
- [23] J. Cartigny and D. Simplot, “Border node retransmission based probabilistic broadcast protocols in ad-hoc networks,” *Journal of Telecommunication Systems*, Vol. 22, No. 1–4, pp. 189–204 (2003).
- [24] Yu-Chee Tseng, Sze-Yao Ni, and En-Yu Shih, “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network,” in *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS)*, pp. 481–488 (2001).
- [25] Chien Chen, Chin-Kai Hsu, and Hsien-Kang Wang, “DISCOUNT: A hybrid probability-based broadcast scheme for wireless ad hoc networks,” in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pp. 2706–2710 (2005).
- [26] H. Lim and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks,” in *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)* (2000).
- [27] W. Peng and X. Lu, “On the reduction of broadcast redundancy in mobile ad hoc networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)* (2000).
- [28] J. Arango, A. Efrat, S. Ramasubramanian, M. Krunz, and S. Pink, “Retransmission and back-off strategies for broadcasting in multi-hop wireless networks,” in *Proceedings of the 3rd IEEE International Conference on Broadband Communication, Networks and Systems (BROADNET)* (2006).
- [29] Chun-Chuan Yang and Chao-Yu Chen, “Reduction of the broadcast redundancy by location awareness in mobile ad hoc networks,” *Journal of Computer Communications*, Vol. 26, No. 18, pp. 2082–2089 (2003).
- [30] “<http://www.scalable-networks.com/>,” Scalable Network Tecnology, (accessed 2010-

- 12-10).
- [31] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st IEEE Conference on Computer Communications (INFOCOM)*, Vol. 3, pp. 1567–1576 (2002).
- [32] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on networking*, Vol. 12, No. 3, pp. 493–506 (2004).
- [33] M. Ichien, T. Takeuchi, H. Kawaguchi, C. Ohta, and M. Yoshimoto, "Isochronous MAC using long-wave standard time code for wireless sensor networks," in *Proceedings of the International Conference on Communications and Electronics (ICCE)*, pp. 172–177 (2006).
- [34] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom)*, pp. 174–185 (1999).
- [35] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Vol. 2, p. 10 (2000).
- [36] S. Lindsey, C. Raghavendra, and Krishna M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 9, pp. 924–935 (2002).
- [37] C. Intanagonwiwat, D. Estrin, R. Godinvan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," *Technical Report of University of Southern California 01-750* (2001).
- [38] J.C. Chen, K. Yao, and R.E. Hudson, "Source localization and beamforming," *IEEE Signal Processing Magazine*, Vol. 19, No. 2, pp. 30–39 (2002).
- [39] LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," *IEEE Std 802.11-1997 edition* (1997).
- [40] J. Jubin and J. Tornow, "The DARPA packet radio network protocols," in *Proceedings*

- of the IEEE*, Vol. 75, No. 1, pp. 21–32 (1987).
- [41] P. Karn, “MACA - a new channel access method for packet radio,” in *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, Vol. 1, pp. 134–140 (1990).
- [42] E. Lin, J.M. Rabaey, and A. Wolisz, “Power-efficient rendez-vous schemes for dense wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Vol. 7, pp. 3769–3776 (2004).
- [43] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks,” in *Proceedings of the 9th International Symposium on Computers and Communications (ISCC)*, Vol. 1, pp. 244–251 (2004).
- [44] C. Guo, L.C. Zhong, and J.M. Rabaey, “Low power distributed MAC for ad hoc sensor radio networks,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Vol. 5, pp. 2944–2948 (2001).
- [45] G. Pei and C. Chien, “Low power TDMA in large wireless sensor networks,” in *Proceedings of the Military Communications Conference (MILCOM)*, Vol. 1, pp. 347–351 (2001).
- [46] J. Li and G. Lazarou, “A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 55–60 (2004).
- [47] L. van Hoesel, T. Nieberg, H. Kip, and P. Havinga, “Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs,” in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pp. 1598–1602 (2004).
- [48] L. van Hoesel and P. Havinga, “A lightweight medium access protocol (LMAC) for wireless sensor networks,” in *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS)*, pp. 205–208 (2004).
- [49] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, “Energyefficient, collision-free medium access control for wireless sensor networks,” in *Proceedings of the 1st ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 181–192 (2003).

- [50] S. Kulkarni and M. Arumugam, "TDMA service for sensor networks," in *Proceedings of the IEEE 24th International Conference on Distributed Computing Systems (ICDCS)*, pp. 604–609 (2004).
- [51] K. Arisha, M. Youssef, and M. Younis, "Energy-aware TDMA-based MAC for sensor networks," in *Proceedings of the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and NeTworking (IMPACCT)* (2002).
- [52] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design," in *Proceedings of the International Solid-State Circuits Conference (ISSCC)*, pp. 3–7 (2002).
- [53] N. Yamauchi, I. Urushibara, A. Aizawa, H. Sato, H. Hosaka, K. Sasaki, and K. Itao, "Nature interfacier version 3 (Ni3): A wearable wireless sensor module with flexible protocol configurability for ubiquitous sensor networks," in *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS)*, p. 20 (2004).
- [54] Ye Wen, Wei Zhang, and Rich Wolski, "Simulation-based augmented reality for sensor network development," in *Proceedings of the 5th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 275–288 (2007).
- [55] Adrian Perrig, Mark Luk, and Cynthia Kuo, "Message-in-a-bottle:user-friendly and secure key deployment for sensor nodes," in *Proceedings of the 5th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 233–246 (2007).
- [56] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y. H. Chee, and J. Rabaey, "A power-managed protocol processor for wireless sensor networks," *Digest of Technical Papers 2006 Symposium on VLSI Circuits*, pp. 212–213 (2006).
- [57] Alan C-W. Wong, Declan McDonagh, Ganesh Kathiresan, Okundu C. Omeni, Omar El-Jamaly, Thomas C-K. Chan, Paul Paddan, and Alison J. Burdett, "A 1V, micropower system-on-chip for vital-sign monitoring in wireless body sensor networks," in *Proceedings of the International Solid-State Circuits Conference (ISSCC)*, pp. 138–139 (2008).
- [58] "<http://www.epsontoyocom.co.jp>," Epson Toyocom Corporation, (accessed 2010-12-10).

## 著者の研究業績目録

### A. 学会誌論文（筆頭論文，査読有）

1. S. Izumi, T. Takeuchi, T. Matsuda, H. Kawaguchi, C. Ohta, and M. Yoshimoto, “Counter-Based Broadcasting with Hop Count Aware Random Assessment Delay Extension for Wireless Sensor Networks,” *IEICE Transactions on Communications*, vol. E91-B, no. 11, pp. 3489–3498, Nov. 2008.
2. S. Izumi, T. Takeuchi, T. Matsuda, H. Lee, T. Konishi, K. Tsuruda, Y. Sakai, H. Kawaguchi, C. Ohta, and M. Yoshimoto, “A 58-uW Single-Chip Sensor Node Processor with Communication Centric Design,” *IEICE Transactions on Electronics*, vol. E93-C, no. 3, pp.261–269, Mar. 2010.

### B. 学会誌論文（共著論文，査読有）

1. H. Noguchi, T. Takagi, K. Kugata, S. Izumi, M. Yoshimoto, and H. Kawaguchi, “Data-Intensive Sound Acquisition System with Large-Scale Microphone Array,” *IPSJ Journal of Information Processing*, 2011. (in press).

### C. 国際会議（筆頭論文，査読有）

1. S. Izumi, T. Matsuda, H. Kawaguchi, C. Ohta, and M. Yoshimoto, “Improvement of Counter-based Broadcasting by Random Assessment Delay Extension for Wireless Sensor Networks,” *Proceedings of IARIA International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 76–81, Oct. 2007.
2. S. Izumi, T. Matsuda, H. Kawaguchi, C. Ohta, and M. Yoshimoto, “Hop Count Aware Broadcast Algorithm with Random Assessment Delay Extension for Wireless Sensor

- Networks,” Proceedings of IEEE/IEICE Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT), pp. 30–35, Apr. 2008.
3. S. Izumi, K. Tsuruda, T. Takeuchi, H. Lee, H. Kawaguchi, and M. Yoshimoto, “A Low-Power Multi Resolution Spectrum Sensing (MRSS) Architecture for a Wireless Sensor Network with Cognitive Radio,” Proceedings of Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 39–44, Jul. 2010.
  4. S. Izumi, T. Takeuchi, T. Matsuda, H. Lee, T. Konishi, K. Tsuruda, Y. Sakai, C. Ohta, H. Kawaguchi, and M. Yoshimoto, “A 58-uW Sensor Node LSI with Synchronous MAC Protocol,” Proceedings of Asia-Pacific Radio Science Conference (AP-RASC), D1-5, Sep. 2010.

#### D. 国際会議（共著論文，査読有）

1. T. Matsuda, S. Izumi, T. Takeuchi, H. Fujiwara, H. Kawaguchi, C. Ohta, and M. Yoshimoto, “Impact of Random Access Memory aware Data Aggregation for Wireless Sensor Network,” Proceedings of IEEE/IEICE Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT), pp. 130–134, Apr. 2008.
2. K. Tsuruda, S. Izumi, H. Lee, T. Takeuchi, H. Kawaguchi, and M. Yoshimoto, “A Flexible Baseband Processor with Multi-Resolution Spectrum-Sensing Functionality,” Proceedings of IEEE/IEICE International Symposium on Information Theory and its Applications (ISITA), pp. 1423–1428, Dec. 2008.
3. T. Takeuchi, S. Izumi, T. Matsuda, H. Lee, Y. Otake, T. Konishi, K. Tsuruda, Y. Sakai, H. Fujiwara, C. Ohta, H. Kawaguchi, and M. Yoshimoto, “A 58-uW Single-Chip Sensor Node Processor Using Synchronous MAC Protocol,” IEEE/JSAP Symposium on VLSI Circuits Digest of Technical Papers, pp. 290–291, June 2009.
4. T. Konishi, K. Tsuruda, S. Izumi, H. Lee, H. Fujiwara, T. Takeuchi, H. Kawaguchi, and M. Yoshimoto, “A 60-dB Image Rejection Filter Using  $\Delta$ - $\Sigma$  Modulation and Frequency Shifting,” Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp. 565–570, Aug. 2009.

5. T. Takeuchi, S. Izumi, T. Matsuda, H. Lee, T. Konishi, K. Tsuruda, Y. Sakai, H. Kawaguchi, C. Ohta, and M. Yoshimoto, "A Single-Chip Sensor Node LSI with Synchronous MAC Protocol and Divided Data-Buffer SRAM," Proceedings of IEEE International SoC Design Conference (ISOCC), pp. 202–207, Nov. 2009.
6. H. Noguchi, J. Tani, Y. Shimai, M. Nishino, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 34.7-mW Quad-Core MIQP Solver Processor for Robot Control," Proceedings of IEEE Custom Integrated Circuits Conference (CICC), Sep. 2010.

## E. 本研究に関するその他の発表（査読無）

1. 和泉 慎太郎、松田 隆志、三上 真司、川口 博、太田 能、吉本 雅彦、「ワイヤレスセンサネットワークのためのタイマ制御によるカウンターベースブロードキャスト方式の改良」、電子情報通信学会総合大会、A-21-26、417 ページ、名城大学、2007 年 3 月。
2. 酒井 康晴、松田 隆志、和泉 慎太郎、竹内 隆、藤原 英弘、川口 博、太田 能、吉本 雅彦、「ワイヤレスセンサネットワークのためのデータ集約を考慮した部分起動メモリの電力削減効果に関する研究」、電子情報通信学会ソサイエティ大会、B-20-10、346 ページ、明治大学、2008 年 9 月。
3. 鶴田 嵩、和泉 慎太郎、李 赫鍾、竹内 隆、川口 博、吉本 雅彦、「コグニティブ無線向け可変帯域デジタルバンドパスフィルタの設計」、電子情報通信学会技術研究報告、ICD2008-82、137–142 ページ、2008 年 10 月。
4. 和泉 慎太郎、松田 隆志、竹内 隆、川口 博、太田 能、吉本 雅彦、「低消費電力センサノード VLSI のための時刻同期型 MAC プロトコルの研究」、電子情報通信学会技術研究報告、NS2008-174、177–182 ページ、2009 年 3 月。
5. 和泉 慎太郎、松田 隆志、竹内 隆、川口 博、太田 能、吉本 雅彦、「時刻同期型 MAC プロトコルの垂直統合設計によるセンサノード VLSI の低消費電力化」、LSI とシステムのワークショップ 2009 ポスタセッション、238–240 ページ、北九州市、2009 年 5 月。
6. 和泉 慎太郎、竹内 隆、松田 隆志、李 赫鍾、小西 恵大、鶴田 嵩、酒井 康晴、川口 博、太田 能、吉本 雅彦、「時刻同期型 MAC プロトコルを用い

る 58-uW ワンチップセンサノードプロセッサ」、電子情報通信学会技術研究報告、ICD2009-58、141–145 ページ、2009 年 10 月。

7. 和泉 慎太郎、李 赫鍾、小西 恵大、岡 顕久、松田 隆志、竹内 隆、太田 能、川口 博、吉本 雅彦、「時刻同期型 MAC プロトコルを用いる 6.4  $\mu$ W シングルチップセンサノード LSI」、LSI とシステムのワークショップ 2010 ポスターセッション、249–251 ページ、北九州市、2010 年 5 月。
8. 小西 恵大、李 赫鍾、和泉 慎太郎、竹内 隆、川口 博、吉本 雅彦、「ワイヤレスセンサネットワークのための  $\Delta$ - $\Sigma$  変調とデジタルアシストを用いたイメージ信号除去に関する研究」、LSI とシステムのワークショップ 2010 ポスターセッション、252–254 ページ、北九州市、2010 年 5 月。
9. 祖田 心平、久賀田 耕史、高木 智也、和泉 慎太郎、野口 紘希、吉本 雅彦、川口 博、「分散処理を用いた超低消費電力 ネットワーク型マイクロホンアレーの研究」、日本音響学会 2010 年秋季研究発表会、585–588 ページ、関西大学、2010 年 9 月。
10. 和泉 慎太郎、野口 紘希、高木 智也、久賀田 耕史、祖田 心平、吉本 雅彦、川口 博、「ネットワーク型マイクロホンアレー間のデータ集約による音声信号ビームフォーミング」、電子情報通信学会 CEATEC JAPAN 2010 連携企画研究報告 (Digital Harmony を支えるプロセッサと DSP、画像処理の最先端)、95–100 ページ、2010 年 10 月。