

博士学位論文

センサネットワークにおける  
高効率分散型制御に関する研究



指導教員 瀬崎 薫 助教授

東京大学大学院情報理工学系研究科  
電子情報学専攻

関根 理敏

# 目次

第1章	序論	1
1.1	研究の背景と目的	2
1.2	本論文の構成と要旨	3
第2章	実空間情報の収集と活用に向けたセンサネットワークの展開	7
2.1	本章の概要	8
2.2	実空間情報を利用したコンテキストウェアサービスの展開	8
2.3	無線センサネットワーク	11
2.3.1	現状と展開	11
2.3.2	要求事項と技術的課題	16
2.3.3	送信距離と消費電力	17
2.4	グローバルなセンサネットワーク	19
2.4.1	現状と展開	19
2.4.2	要求事項と技術的課題	22
2.5	本章のまとめ	23
第3章	適応的スロット予約による低消費電力メディアアクセス制御	24
3.1	本章の概要	25
3.2	関連研究	26
3.2.1	CSMA方式	26
3.2.2	TDMA方式	26
3.3	提案手法	27
3.3.1	概要	27
3.3.2	送信要求の手順	30
3.3.3	スロット割り当ての手順	31
3.3.4	データパケットの送受信の手順	33
3.3.5	スロット予約期間の適応的制御	35
3.3.6	制御パケットにおける情報集約の効果	36
3.4	性能評価	37
3.4.1	概要	37
3.4.2	スロット予約期間における平均アクティブ期間特性	38
3.4.3	制御パケット送信数に関する特性	39
3.4.4	データ伝送量に関する特性	41
3.4.5	平均消費電力特性	43
3.4.6	消費電力効率特性	44
3.4.7	エンドツーエンド伝送における遅延特性	45

3.4.8	フレーム長の差異の影響	46
3.4.9	部分的なノードの送信の影響	48
3.4.10	干渉距離の影響	51
3.4.11	実装による動作確認	54
3.5	本章のまとめ	61
<b>第4章</b>	<b>間欠通信適応型同期制御</b>	<b>63</b>
4.1	本章の概要	64
4.2	関連研究	64
4.2.1	完全同期方式	64
4.2.2	部分的同期方式	65
4.2.3	非同期方式	66
4.2.4	S-MAC	67
4.2.5	スケジュール同期のアルゴリズム	68
4.2.6	従来手法の問題点	69
4.3	提案手法	70
4.3.1	概要	70
4.3.2	非同期状態からの同期	70
4.3.3	スケジュールの整合化及び一時的な同期の維持	72
4.3.4	ノードの追加と離脱の検出	76
4.4	性能評価	77
4.4.1	概要	77
4.4.2	同期パケットの総送信数	77
4.4.3	総消費電力	79
4.5	本章のまとめ	80
<b>第5章</b>	<b>P2P ネットワークにおける適応的センサデータ配置法</b>	<b>81</b>
5.1	本章の概要	82
5.2	関連研究	82
5.2.1	概要	82
5.2.2	分散ハッシュテーブル	85
5.2.3	ユビキタスネットワーク環境におけるリソース発見機構	88
5.2.4	PHT (Prefix Hash Tree) を利用したセンサデータの収集と管理	94
5.3	提案手法	98
5.3.1	概要	98
5.3.2	データ更新における負荷分散	98
5.3.3	データ更新におけるオーバーヘッドの削減	98
5.3.4	データ検索における負荷分散	99

5.3.5	検索におけるオーバヘッドの削減 . . . . .	101
5.3.6	クエリ配信とデータ登録のオーバヘッドを反映した適応的なデータ・ メタデータの配置 . . . . .	102
5.4	性能評価 . . . . .	104
5.4.1	データ更新における負荷分散の効果 . . . . .	105
5.4.2	データ更新における履歴の利用の効果 . . . . .	105
5.4.3	データ検索における負荷分散の効果 . . . . .	106
5.4.4	データ検索におけるポインタ配置の効果 . . . . .	106
5.4.5	データ検索における履歴の利用の効果 . . . . .	107
5.4.6	適応的データ・メタデータ配置の効果 . . . . .	108
5.5	本章のまとめ . . . . .	111
<b>第6章</b>	<b>結論</b> . . . . .	<b>113</b>
6.1	本論文のまとめ . . . . .	114
6.2	今後の課題と展望 . . . . .	114

## 目 次

1.1	Smart Dust の Mote(左) と研究開発用の MICA Mote(右)	2
2.1	実空間情報の取り扱いにおける主な構成要素	9
2.2	リアルタイム実空間情報検索のアーキテクチャ	9
2.3	実空間情報検索におけるユーザインタフェースの例	10
2.4	UCLA CENS における環境モニタリングの実験装置 (左) と林野における配置 (右)	12
2.5	センサネットワークのプロトコルスタック	12
2.6	TCP/IP スタック (左) とセンサネットワーク型スタック (右) の通信形態の相違	13
2.7	他ノードを利用した推定距離を用いた位置の推定	14
2.8	Greedy Forwarding	15
2.9	Art Gallery 問題	16
2.10	消費電力の内訳の例	17
2.11	無線通信モデル	18
2.12	エンドノードまでの距離と総消費電力の関係	19
2.13	ネットワークリソースの時間的・空間的概念	20
2.14	ネットワークリソースへのアクセスの手順	21
3.1	1 フレームにおけるタイムスケジュール	28
3.2	送信要求 (TR) とスロット割り当て (SA) の各パケットのフォーマット	29
3.3	ユニキャスト・ブロードキャストにおける動作例	30
3.4	動作例におけるネットワークトポロジー	31
3.5	スロット予約における不整合の発生の例	32
3.6	制御パケット損失によるデータパケット衝突の発生の例	34
3.7	スロット予約期間の適応化制御	35
3.8	制御パケットにおける情報集約	36
3.9	スロット予約期間における平均アクティブ期間特性	39
3.10	データパケット送信あたりの平均制御パケット送信数特性	40
3.11	平均総制御パケット送信数特性	41
3.12	MAC 層における平均スループット特性	42
3.13	配送率特性	43
3.14	平均消費電力特性	44
3.15	消費電力効率特性	45
3.16	エンドツーエンド伝送における遅延特性	46
3.17	フレーム長とエンドツーエンド伝送における遅延特性	47
3.18	フレーム長と配送率	47
3.19	フレーム長と平均消費電力特性	48

3.20	送信エリアの割合とスループット特性	49
3.21	送信エリアの割合と配送率特性	49
3.22	送信エリアの割合と平均消費電力特性	50
3.23	送信エリアの割合と消費電力効率特性	50
3.24	干渉距離とパケットロス率特性	51
3.25	干渉距離と配送率特性	52
3.26	干渉距離と平均消費電力特性	52
3.27	干渉距離と消費電力効率特性	53
3.28	スロット予約のフローチャート	55
3.29	スロット予約期間の適応的制御のフローチャート	56
3.30	実装における通信スタック	57
3.31	提案手法のプログラム関数	58
3.32	提案手法の Mote を用いた動作確認	59
3.33	制御パケット送受信数と無線インタフェースの状態の観測	60
4.1	完全同期型の動作例	65
4.2	動作例におけるネットワークポロジ	65
4.3	部分的同期型の動作例	66
4.4	動作例におけるネットワークポロジとアクティブ・スリープのスケジュール	66
4.5	非同期型の動作例	67
4.6	非同期型の動作例におけるネットワークポロジとスケジュール	67
4.7	通信時における送受信ノード間の同期の動作例	70
4.8	通信時における送受信ノード間の同期のネットワークポロジ	71
4.9	スケジュールの整合化の動作例	72
4.10	スケジュールの整合化の動作例におけるネットワークポロジ	73
4.11	FTSP における送信ノードによる同期パケット送信の動作及び履歴情報の例	74
4.12	タイマ修正前の時刻のずれ	75
4.13	タイマ修正後の時刻のずれ	75
4.14	同期パケット送信回数	78
4.15	総消費電力	78
5.1	ハイブリッド型リソース発見方式 (例: Napstar)	83
5.2	ピュア P2P 型リソース発見方式 (例: Gnutella)	83
5.3	構造型リソース発見方式 (例: DHT)	84
5.4	2次元空間における CAN の動作	86
5.5	キーとノード ID のマッピングと $O(N)$ の参照	87
5.6	ショートカッターティングと $O(\log(N))$ の参照	87
5.7	JXTA の仮想ネットワーク	89
5.8	ファイアウォールや NAT 環境での P2P 通信	90
5.9	INS におけるリソース更新	91

5.10	INS における記述の分割	91
5.11	利用可能な駐車場発見のクエリの記述例	93
5.12	位置情報の階層化構造	93
5.13	PHT のツリー構造	95
5.14	leaf node の発見とクエリの配信 (パターン 1)	96
5.15	leaf node の発見とクエリの配信 (パターン 2)	97
5.16	データ更新の負荷分散の動作	99
5.17	データ検索の負荷分散の動作	100
5.18	ポインタの配置の動作	101
5.19	データ・メタデータの配置	103
5.20	ノード順位とデータ更新数の関係	105
5.21	データ発生率とデータ登録における制御メッセージ送信数の関係	106
5.22	ノード順位と leaf node におけるクエリの受信数の関係	107
5.23	データ発生率と検索における制御メッセージ送信数の関係 (検索範囲の広 さ=10×10)	108
5.24	クエリ発生率と検索における制御メッセージ送信数の関係 (検索範囲の広 さ=0.01×0.01)	109
5.25	データ発生率と総制御メッセージ送信数の関係	110
5.26	クエリ発生率と総制御メッセージ送信数の関係	110
5.27	データ発生率とクエリ発生率の分布に偏りがある場合のクエリ発生率と総 制御メッセージ送信数の関係	111

## 表 目 次

1	メディアアクセス制御の性能評価におけるシミュレーション諸元	37
2	同期制御の評価におけるシミュレーション緒元	79

## 第1章 序論

---



## 1.1 研究の背景と目的

近年の電子デバイスの微細化技術 (MEMS: Micro-Electro-Mechanical Systems) や無線通信技術の発展に伴い、温度、光、音などのセンシング機能を持った小型デバイスが、無線リンクで自律分散的にネットワークを構成し、実空間の環境情報を取得することで、さまざまなサービスを実現するセンサネットワークがユビキタスコンピューティング環境において注目されている。生活空間のいたるところにコンピュータが遍在し、ユーザがいつでもどこでも柔軟に利用できる環境において、実空間の環境情報をコンピュータの仮想空間上に取り込むインタフェースとなるセンサネットワークは、今後、重要な役割を担うと考えられる。

1999年に Kris Pister らによって立ち上げられた米国の UC Berkeley を中心とした研究開発プロジェクトである Smart Dust[1] が発端となり、近年盛んにセンサネットワークに関する研究開発が行われている。図 1.1 左のように、Smart Dust では大きさが数ミリ程度の無線センサで、太陽電池もしくは厚いフィルム状のバッテリーの電源供給装置、センサ、通信回路で構成され、化学センサとして環境情報を収集し、音波、振動、磁気センサとしても使用できる。また、図 1.1 右は UC Berkeley を中心にして開発が進められている MICA Mote と呼ばれるセンサノードである。基本部分がわずかに数百バイト程度で、メモリが少ないセンサにも搭載できる OS で起動し、自律分散的にネットワークを構成する。センサネットワークから得られる情報をさらに解析、加工してユーザにサービスとして提供することで生活環境がより充実していくと考えられる。

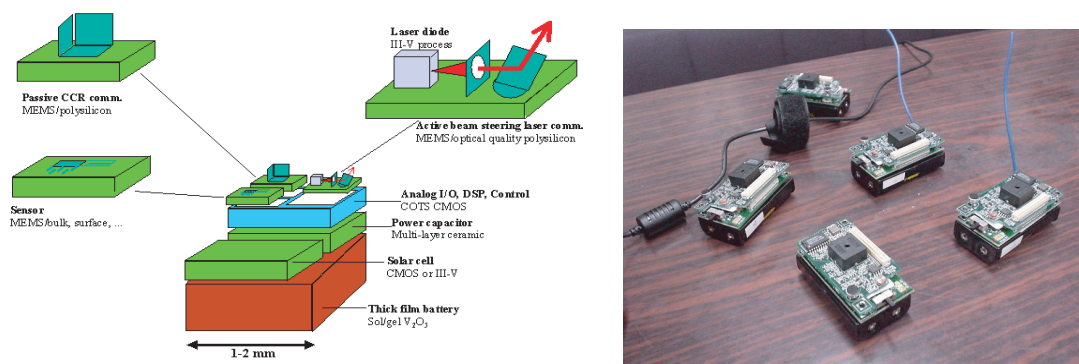


図 1.1 Smart Dust の Mote(左) と研究開発用の MICA Mote(右)

センサはこれまでも電化製品・建築環境設備など、さまざまな所で利用されている。しかしながら、それらは主に、限定的な情報を得て特定の目的を達成するための装置として、スタンドアロンのように利用されてきた。近年、無線技術やネットワーク技術を利用して

技術により小型センサのネットワークを行うことが可能となり、微細なスケールで広範囲にわたる面的なデータを取得することが可能となってきた。例えば建築環境や農作物が生育するビニルハウスの温度、湿度などの環境管理システムにネットワーク化されたセンサが利用されている [2]。また、地震を検知してガス管を制御したり、老人ホームでの入居者の生活を管理したりすることなどに利用されている。また、センサは、一般家庭だけでなく、都市のインフラや森林、草原などの自然環境を含めた広範囲に配置されることが考えられる。そのような環境を想定すると、位置や時間的な要素との関連も深い時空間データが遍在し、どこに何があるか、あるいはどんな状態か、という環境情報の収集と活用が重要味を帯びてくると考えられる。今後はセンサがネットワークに接続されることにより、新たなサービスが創出されることが期待される。

WWW(World Wide Web)上の汎用的なコンテンツと比較して、実空間情報は個人依存性が高く、ある人にとってあるときだけ有効な一過性のものとなることもある。ユーザのデータの変動する環境情報を、ユーザやアプリケーションの要求に応じて取得することが必要になる。実空間から得られる情報は多種多様で、その量も多いため、情報が流通するネットワークの形態が無線ネットワーク、有線ネットワーク問わず、数百、数千、あるいは数万にのぼる端末からなるセンサネットワークの扱うデータは膨大である。センサデータが流通するネットワークの特徴を考慮したネットワーク制御が必要となる。アプリケーションの要求に応じて、多数のノードが自律的に情報の収集・管理を行うため、効率的にデータを取得するためにはスケーラブルな自律分散的アルゴリズムが必要となる。また有線ネットワークにおいても各端末のリソースの能力、アプリケーションやユーザからの要求を考慮して、柔軟でスケーラブルなデータの収集・管理機構を構築する必要がある。

## 1.2 本論文の構成と要旨

本論文では、多種多様な実空間情報を、センサネットワークを介して収集し、それを活用する環境において、ネットワークを適応的かつ自律分散的に制御することで、ネットワークリソースを有効に活用し、効率的に実空間情報を流通させ、さまざまなサービスに活用させるということを目的とする。その技術的課題の解決について、具体的には無線センサネットワークを想定し、長期間のネットワーク稼働や運用コスト削減に必要な低消費電力化に向けたメディアアクセス制御 (MAC: Medium Access Control) や、ノード間のアクティブ・スリープのスケジュールにおける同期制御について論じている。またインターネット上でのセンサデータを収集や管理において、特に高頻度なデータの更新を考慮して、適応的にコンテンツを配置する手法に論じている。

第2章では、実空間情報を収集し、活用するためのセンサネットワークの概要を述べるとともにセンサネットワーク特有の技術的課題や要求事項を明らかにしている。無線センサネットワークでは、技術的課題として、低消費電力化が挙げられる。無線センサネットワークではバッテリーやメモリの容量、CPU(Central Processing Unit)の能力などに大きな制約があり、無線リンクで接続されたノードから構成されるセンサネットワークでは、電力の消費を極力抑え、ネットワークをなるべく長期間稼働させることが必要である。特に無線通信における電力消費が大部分を占めるため、メディアアクセス制御によって、通信を行わない間はスリープ状態となる間欠通信を行うことが有効である。また間欠通信におけるオーバーヘッドが小さい同期制御も重要となる。さらにインターネット上で多種多様で大量のセンサデータをスケラブルに収集、管理、また配信を扱うメカニズムも必要となる。特にセンサデータは一般のWWW上のコンテンツと比較して、更新頻度が高く、その更新頻度が地理的・時間的によって異なるため、特定のノードに負荷が集中してボトルネックが発生する可能性がある。そのために適応的かつスケラブルにコンテンツを配置することが重要となる。

第3章では、トラフィックの高低が時間的・空間的に偏りがあるセンサネットワークのアプリケーションを想定し、トラフィックの発生に応じて自律分散的かつ動的にスロットを割り当てることで、スループットの低下を抑制しつつ、高い消費電力効率を実現する無線メディアアクセス制御について論じている。無線通信では実際にパケットの送受信を行わなくとも、無線インタフェースがアクティブ状態であれば電力を消費するため、一般に通信を行う期間が比較的短いセンサネットワークにおいては、送受信を行わない期間は、極力無線インタフェースはスリープ状態になることが望ましい。センサネットワークでは、通常はトラフィックがほとんどなく、イベント検知、クエリの配信などの際に、一時的にノードがデータの送受信を行う場合がある。また、異なる位置に配置されたノードによってもトラフィックが異なる場合もある。そのため、MACプロトコルの設計の上で、時間的・空間的なトラフィックの相違に対して、柔軟に対応できることが重要であると考えられる。センサネットワークにおける従来のCSMA(Carrier Sense Multiple Access)方式では、キャリアセンスやバックオフが必要となるので、自分に無関係なパケットの送受信を行う場合があり、消費電力を浪費する可能性がある。一方、TDMA(Time Division Multiple Access)方式では、データパケットの送受信は予め割り当てられたスロットに行われるため、CSMA方式と比較して、データパケットの送受信に伴う冗長なアクティブ期間が少なく、消費電力は抑制される。厳密なスロット割り当てによるオーバーヘッドの増加やノードによるトラフィックの違いに応じた割り当てが困難という課題がある。提案手

法では、データパケットの送受信を行うノードのみがスロット予約を行うことで、制御パケットのオーバーヘッドが削減される。さらに各ノードが、スロット予約期間においてアクティブ状態になる期間を、トラフィックの変動に応じて動的に制御する。提案手法をセンサ実機で動作確認を行った上、計算機シミュレーションによる評価の結果、提案手法では、従来手法と比較して、消費電力を抑制しつつ、トラフィックが増加しても、スループットを維持することができることを示している。

第4章では、間欠通信に適したアクティブ・スリープのスケジュール同期制御について論じている。間欠通信を行うためには送受信ノードがアクティブ状態でなければならないため、間欠通信を行う際にはこのアクティブ・スリープ状態の同期制御が必要となる。この間欠通信における同期制御において、ネットワーク全体で1つの同期を維持する手法は同期制御のオーバーヘッドが大きいという問題がある。通常時は同期制御を行わず、通信を行う場合には、送信側が1周期の間受信ノードを起こすための信号を送信し続けることで周囲のノードをアクティブ状態にし、通信を行う方式もある。この方式ではアクティブ・スリープ状態のスケジュールリング同期が不要あるが、通信に無関係なノードもアクティブ状態になり、通信を行うたびに同期制御が必要となるため、消費電力を浪費する可能性もある。そこで本研究では、間欠通信における同期制御のオーバーヘッドの削減による省電力化を目的とし、送信要求の発生に応じて同期制御を行い、さらにクロックスキューを考慮に入れることで、長期間、同期を維持し、同期制御のオーバーヘッドを抑制したオンデマンド型同期制御手法の提案を行う。提案手法では、同期パケットを受信したノードは、その後送信ノードが次にアクティブとなる時間までスリープ状態となることで、無駄な同期パケットを受信することなく消費電力を削減できる。提案手法では、データパケット送信要求が発生した場合のみ同期を取り、その後は定期的な同期パケットの送信は行わない。同期の維持は従来手法の高精度同期方式を基本に行っている。送信側の時刻と受信側の時刻の差を履歴情報として、送信側のクロックスキューを算出することで、送信側と受信側の時刻を同期させる時刻同期プロトコルである。まず、同期の基準となるノードが同期パケットを送信する。このときの同期パケットには基準ノードのパケットの送信時間が記述される。この同期パケットを受信したノードはパケットの受信時間を記憶する。そして送信ノードとの時刻のずれを記憶する。この動作を繰り返すことにより、送信ノードとのタイマのずれの履歴を取得することができる。この履歴情報を直線近似し、その直線の傾きであるクロックスキューを算出することで送信ノードとのタイマのずれを補正することが可能となる。計算機シミュレーション評価により、提案手法では、制御パケットの削減と消費電力の削減できることを示している。

第5章では、リアルタイムに変動するセンサデータのインターネット上の流通を想定し、構造型ピアツーピア(P2P)ネットワークにおいてデータ登録やクエリ配信における履歴情報の利用やポイントの配置、また適応的なコンテンツ配置により、データの更新や検索における負荷分散を行う手法について論じている。P2Pネットワークでは、効率化、ロバスト性の向上、また負荷分散などの観点から、適切にコンテンツを配置する必要がある。センサデータをP2Pネットワーク上でグローバルに共有する場合、センサデータはリアルタイムに変動するため、データ更新におけるオーバーヘッドが大きくなることが想定される。しかしながら、従来手法では、主に検索における負荷分散や、オーバーヘッドの削減に着目し、データの更新における負荷分散や、オーバーヘッドの削減に十分対応しているとはいえない。そこで本章では、連続的な検索範囲を含むレンジクエリに対応した構造型P2Pネットワーク上で、高頻度に更新されるデータを適応的に配置する手法について提案を行っている。提案手法では、ノードが自分の保持するデータを分割して複数のノードに保持させることで、データの更新における負荷分散を行っている。また、データの複製を配置することにより、データ検索における負荷分散を行っている。さらに、データの登録や検索における履歴情報を利用したり、データを提供するノードが、適応的にデータやメタデータの登録や更新を行ったりすることで、データの更新および検索におけるオーバーヘッドを削減している。計算機シミュレーションによる評価により、提案手法に有効性を示している。

## 第2章 実空間情報の収集と活用に向けたセンサネットワークの展開

---

## 2.1 本章の概要

本章では多種多様な実空間情報の収集・提供において主要な役割を果たすセンサネットワークについて、まず展開されるサービスについて述べる。さらに、サービスを実現するための要求事項と、具体的な技術課題について、無線および有線ネットワーク環境のそれぞれについて述べる。

## 2.2 実空間情報を利用したコンテキストウェアサービスの展開

インターネットは原型である ARPANET での軍事的・学術的な利用を経て、1993 年に最初の WWW ブラウザである MOSAIC がリリースされたのを契機に、この 10 余年間で飛躍的に発展・普及してきている。ウェブ上には数十億以上のページが存在するといわれており、現在も爆発的な勢いで増加している。インターネットという広域ネットワークの発展により、ユーザが時間や距離といった物理的制約の束縛から開放され、瞬時にさまざまな情報を取得することが可能となった。また端末の小型化、端末やサービスの低廉化、またネットワーク技術の発展などによって、携帯電話も爆発的に普及し、多くの一般の人々にとって、ネットワークへのアクセスが身近のものとなった。そのような環境において、ユーザのネットワークサービスの高度化への関心も高まってきている。

ユビキタス環境では、パーソナライズ化が進み、個々のユーザの要求に適った木目細やかなレベルでそれぞれの環境や利用するアプリケーション特化したシステムの構築が進むと考えられる。現在の普及しているネットワークサービスでは多くのユーザに有益な汎用的なサービスを提供しているが、ユーザやユーザを取り巻く環境のコンテキストを十分考慮してサービスに還元しているとはいえない。例えば全く同じ検索キーワードで検索したユーザが複数いても、それぞれに求められる”最適な検索結果”はユーザによって異なる。また、例えば「半径 300 メートル以内の空いている駐車場は?」「最寄の本屋は?」といったクエリはユーザの位置情報などのコンテキストを加味しなければならない。コンテキストはユーザや時と場合によって千差万別であり、さまざまな複雑性、集約性、時間依存性を含んださまざまなクエリに対処できる機構が必要となる。身の周りの温度や、位置情報といった情報を収集することでより状況に適したサービスが可能となる。また、ユーザに直接作用するサービスだけでなく、環境情報をコンピュータに取り込むことによって環境制御、防災、また防犯といった環境やモノに作用することでユーザが恩恵を受けられる間接的なサービスも可能となる。それらのコンテキストを収集するために、ネットワーク化された多種多様なセンサを利用することがきわめて有効である。

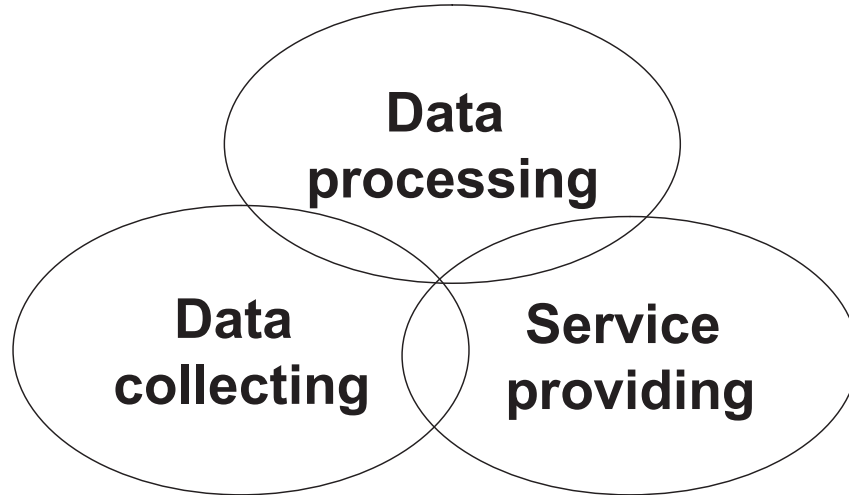


図 2.1 実空間情報の取り扱いにおける主な構成要素

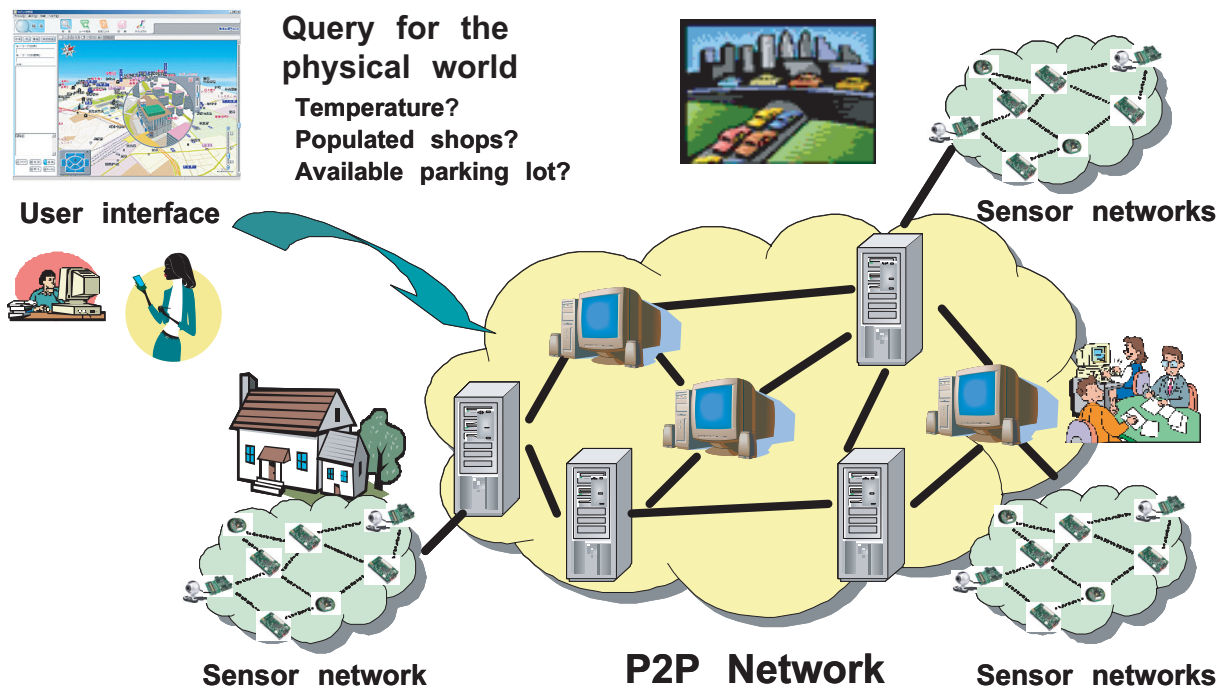


図 2.2 リアルタイム実空間情報検索のアーキテクチャ



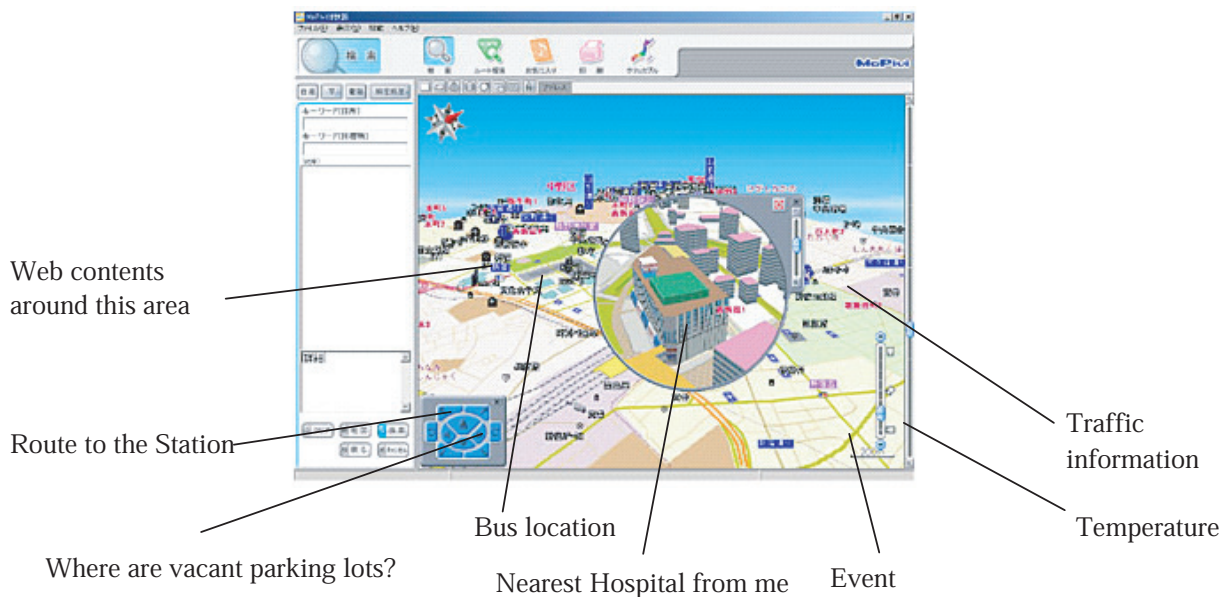


図 2.3 実空間情報検索におけるユーザインタフェースの例

図 2.1 に実空間情報の取り扱いにおける主な構成要素を示す。センサデバイスやネットワークなどを利用した実空間情報の取得，その情報をユーザやアプリケーションにとって有意義なものにする処理，また処理されたデータをサービスとして提供することに分類される。これらの 3 要素を基盤としてさまざまなサービスが実現される想定される。例としてリアルタイム実空間情報の検索が挙げられる。実空間で起こっていることをコンピュータの仮想空間に取り込み，ユーザやアプリケーションが必要に応じてその情報を取得するサービスが今後重要となってくると考えられる。また，図 2.2 にそのシステムのアーキテクチャの例を示す。ローカルには無線センサネットワークを構築し，各センサネットワークがゲートウェイを介してインターネットに接続される。実空間情報を取得するセンサ群が無線でネットワーク化され，それがインターネットに接続され，広域なセンシングシステムを構成する。このようなプラットフォームを利用してさらにセンサネットワークを利用して動的に変動するさまざまなコンテキストを抽出できれば図 2.3 のようにリアルタイムに変動する実空間情報を検索できるようなユーザインタフェースの提供も可能となるだろう。そのためにはリソースそのものを量的・質的に充実させていくとともに，リソース発見技術を向上させていく必要があると考えられる。

## 2.3 無線センサネットワーク

### 2.3.1 現状と展開

近年，無線通信技術及び半導体技術の発展に伴い，無線通信機能を持つ情報通信端末の小型化・低コスト化が進んでいる．同様に，実空間から情報を収集するデバイスとしてセンサデバイスの小型化が進み，センサデバイス自体に無線機能を備えたものをノードとし，それらを無線ネットワーク化する無線センサネットワークが注目され，様々な研究が盛んに行われている [3][4][5]．このような小型の情報通信端末がセンシング機能を保持することは，実空間とコンピュータネットワークの融合であるユビキタスコンピューティングの実現に向けて重要な役割を担うと考えられる．センサネットワークでは，通信ノードは同時にセンサとなり，自身が検出した情報を発信するだけでなく他のノードが発した情報を中継することにより，多数の通信ノードが面状に広がってセンシングを行うことができる．さらに技術の進歩により，無線センサノードを1つのLSI(Large Scale Integration)とすることまでも可能となってきた [6]．このような無線センサノードを用いれば，これまでの計測システムとは比較にならないくらい大規模かつ高解像度の分散計測システムを，莫大な数の安価な無線センサノードによって構築することが可能となり，それによって超高層ビルや大型橋梁などの建造物のモニタリング，道路や鉄道などの交通網の監視システム，都市環境制御などの実現が期待できる．UCLA (University of California, Los Angeles) CENS (Center for Embedded Networked Sensing)[7][8]ではセンサネットワークを用いて地球物理学，土壌・水質汚染，生態システムの観点から研究を行っている．図2.4のように風力や温度などさまざまな測定デバイスを搭載した機器を自然環境に配置することで環境情報を取得している．

しかし，その一方で，無線センサネットワークの実現には様々な技術課題がある．具体的には，センサネットワークにおいて，制約のある資源(電力資源，計算資源，通信資源)をどのように効率的に利用するかということが第一の課題となる．その中でも，電力が低下すると，センサノード自体が機能できなくなるため，特に電力資源において効率の良い利用が要求される．センサネットワークはインターネットのTCP/IPプロトコルスタックと比較していくつかの相違点がある．図2.5にセンサネットワークの各層と管理機構を示したプロトコルスタックのモデル例を示す [9]．この例で，センサネットワークのプロトコルスタックにおいてもTCP/IPプロトコルスタックと同様に，物理層，データリンク層，ネットワーク層，トランスポート層，アプリケーション層の各層で表現される．また，低消費電力な電力管理機構，ノードの移動に対するモビリティサポート機構，センシングやデータ送信のタイミングを制御するタスク管理機構など，複数の層にまたが



図 2.4 UCLA CENS における環境モニタリングの実験装置 (左) と林野における配置 (右)

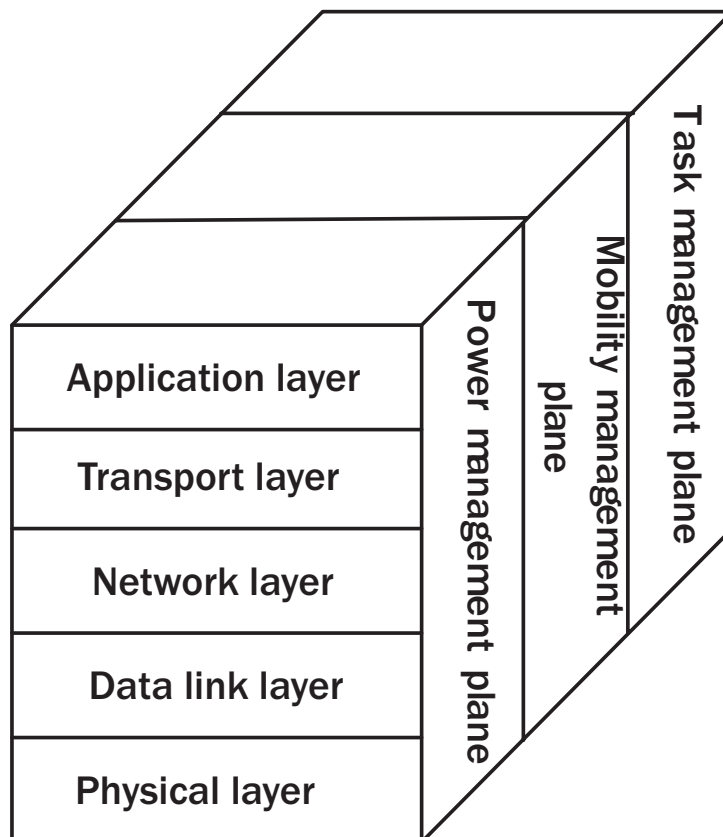


図 2.5 センサネットワークのプロトコルスタック

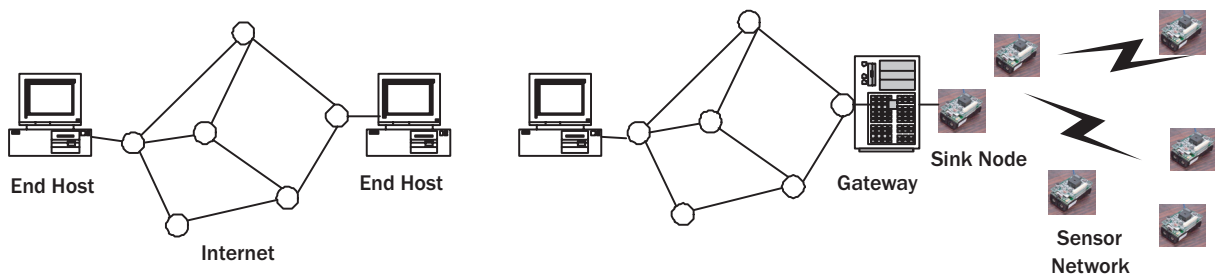


図 2.6 TCP/IP スタック (左) とセンサネットワーク型スタック (右) の通信形態の相違

る管理機構が存在する．以下に各層ごとにその特徴を述べる．

物理層では周波数や変調方式の設定，それに関わるハードウェア制御などを規定する．データリンク層ではリンク間通信の誤り制御やメディアアクセス制御 (MAC) を規定する．誤り制御としては Forward Error Correction (FEC) や Automatic Repeat Request (ARQ) などが用いられ，メディアアクセス制御では CSMA や TDMA などの制御が用いられる．ネットワーク層では経路制御を行う．低消費電力志向のルーティングアルゴリズムやトポロジの変化に強いアルゴリズムが考案されている．またノードの持つ属性によって経路制御するなど，センサネットワーク独特のアルゴリズムが存在する．トランスポート層ではエンドツーエンドの通信において，信頼性の高いデータ通信を行うための制御をする．しかしながら，センサネットワークでは，一般にトランスポート層によるネットワーク制御が想定されていない．一般的にセンサネットワークでは，リソースの制約の大きい多数のセンサから構成されるため，図 2.6 のように IP を持つホストとの一対一のエンドツーエンド通信は行われず，ゲートウェイなどを介して間接的に通信する．また，End-to-End の再送はオーバーヘッドとなるため，情報伝送の信頼性に関する制御はリンク層に委ねられるのが一般的である．アプリケーション層ではデータ処理を行う．特にネットワーク内でのデータ集約における情報処理はネットワーク層のルーティングとも関係し，センサネットワークの大きな特徴の一つとなっている．

各層のプロトコルに要求される機能や性能は一般にアプリケーションに依存する．プロトコル設計が特定のアプリケーションを想定し，それに依存する場合，そのアプリケーションに対して性能が良いが期待されるという長所があるが，他のアプリケーションに対して性能が劣る場合があるという短所がある．一方，アプリケーションに依存せず，一般的な利用状況を想定してプロトコル設計が行なわれた場合，さまざまなアプリケーションに適用可能という長所があるが，特定のアプリケーションに対してアプリケーション依存型の場合より性能が劣る場合があるという短所がある．現段階では，一つのネットワーク

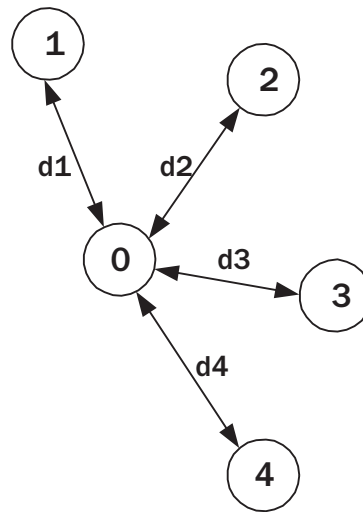


図 2.7 他ノードを利用した推定距離を用いた位置の推定

で複数の目的のために複数のアプリケーションを動作させることはまだ一般的でないため、インターネットのようにプロトコルのレイヤリングを明確化する必然性は大きくない。しかし、今後さまざまなアプリケーションが利用されることが想定すると、より汎用的なプロトコル設計が重要になってくると考えられる。

センサネットワーク上の各ノードは周辺の情報を地理的に分散されたセンサによって取得する。そのため、例えば、ある地域の平均気温を調べたり、ある物体やイベントをトラッキングしたりするというアプリケーションもあるため、その情報にはどこで取得したかという位置情報がきわめて重要な情報となる。また、低消費電力、低遅延をターゲットとしたルーティングなどのネットワーク制御においても位置情報の取得と利用は重要である。センサネットワークなどで利用される小型デバイスでは、サイズや消費電力などの観点からすべてのノードがGPSが利用できるとは限らない場合がある。そこでインフラに設置されたデバイスを利用せず、図2.7のように、近くのノードを利用して自分の位置を推定するシステムを利用する[10]。3点以上の周辺のノードが基準となる位置情報を獲得していれば、それを利用して他のノードも自分の位置を知ることができる。ノード1, 2, 3, 4はそれぞれ、自分の位置情報をノード0に送る。基準となる位置情報はさまざまな位置取得デバイスや位置推定システムによって得られた情報である。ノード0は他の各ノードからの距離を測定し、自分の位置を推定する。また、部分的なネットワークの障害や消費電力を抑えるために適応的に精度を変えることが重要となっている。

位置情報を用いた Greedy Forwarding[11] は自分の隣接ノードの位置情報を取得し、宛先ノードの位置を推定した後、隣接ノードのうち、最も宛先ノードに近いノードにパケッ

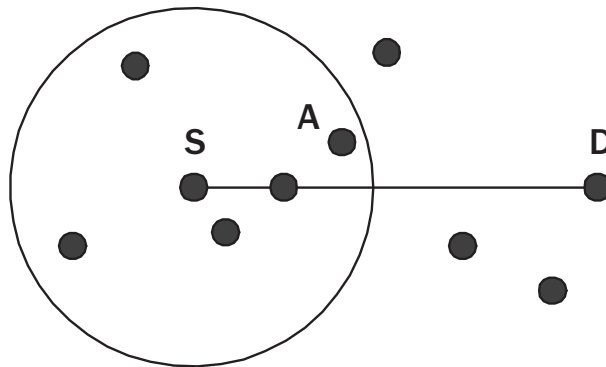
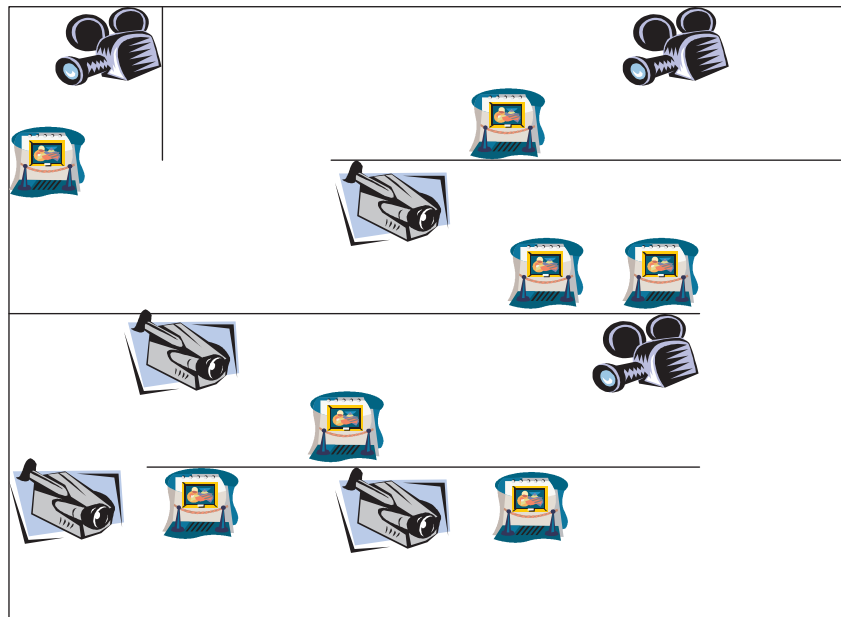


図 2.8 Greedy Forwarding

トを配送することを繰り返し、宛先ノードにパケットを到達させる方法である。図 2.8 では宛先ノードに最も近い位置にあるノード A が選択される。

センサネットワークでは、通信の接続性を保ちながらカバーするエリアを最大化するなどのノードの位置を最適に配置するというアルゴリズムも必用となってくる。空間的な配置問題に関しては、図 2.9 で示すように、Art Gallery 問題を準じた問題が存在する。Art Gallery 問題とは、美術館で三次元空間の映像をすべてカバーするカメラの個数を最小化する問題である [12]。可能性のある場合の数が膨大となるため、ヒューリスティックな方法でさまざまなアルゴリズムが考案されている。センサネットワークでは実用的なセンサ配置アルゴリズムの研究が行われている。自己配置アルゴリズム [13] のような、元のセンサの集合が、接続性を保ったままノードのカバーエリアが最大となるように、自律分散的に最適な配置を取る。厳密な最適配置を決定することは難しい。静的なノードの検出エリアを最大化するとともに、フリースペースにノードを向かわせるなど障害によって検出できないエリアをなるべくなくすること、動的な障害に対応させることが必用となる。さらに、位置配置が不安定にならないようにすること、スケーラビリティの確保が課題となる。

センサネットワークではエンドツーエンドで個々のノードの ID を指定してデータ通信を行うというよりは、観測されたデータの属性や値といった、ノードの持つ属性をベースとして、その属性を持つ不特定のノードと通信を行うのが一般的である [14][15]。このように、センサネットワークは一般に ID を基調としたネーミングよりも位置や観測された内容といったノードの持つ属性を基調としたネーミングが行われる。この特徴を利用して、例えば Directed Diffusion [16] では、ユーザが必要とする情報の属性に応じてルーティングが行われる。このような特徴からアドホックネットワークのようにネットワーク内で一対一の通信を行うよりも、ある特定の属性を持ったノードのグループとの一対多通信、あるいは多対一通信が行われることが多い。



How many cameras do we need to guard a given gallery?

図 2.9 Art Gallery 問題

### 2.3.2 要求事項と技術的課題

センサネットワークにおいてはインターネットや携帯電話などのセルラーネットワーク、ラップトップパソコンやPDAで構成されるようなアドホックネットワークと一般的に以下の点で性質が異なる。以下では、その相違とセンサネットワーク特有の課題を述べる。観測所で観測されたデータを中央局に送信して集約し、データ解析を迅速かつ統合的に行うテレメータシステムや、工場の生産現場などにおいてFA(Factory Automation)という形で、複数のセンサの協調動作により複雑な工程を制御するシステムなど、センサにより情報収集を行うシステムは古くから利用されてきた。

センサネットワークを構成するノードは一般的に小型のデバイスであり、利用できるCPU、メモリ、通信帯域、電力に大きな制約があり、これらの制約事項を考慮する必要がある。また、一般に各々のノードの機能の信頼性が低く、故障も起こりやすい。環境条件によっては通信できなくなるときもある。そのため、ノードのエラーに対してネットワーク全体で強い耐性を持つことも必要となる。また、数百から数千という、多数のノードからネットワークが構成され、扱う総データ量も多いため、スケーラブルなネットワーク制御が必要となる。一般に、コストや利便性などの観点から、センサノードはできる限り小

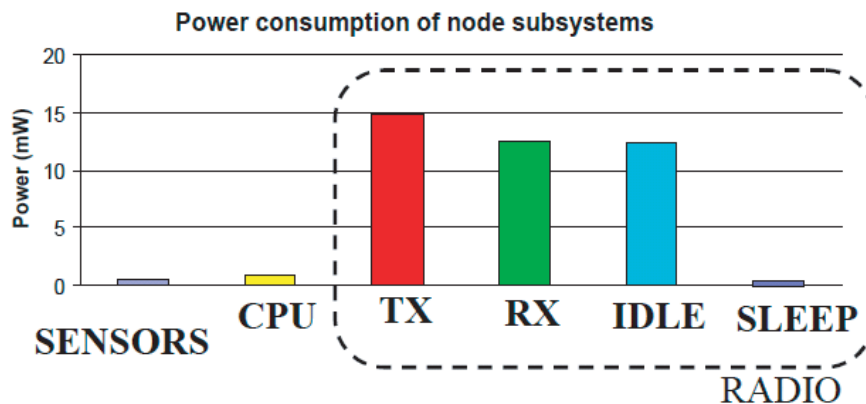


図 2.10 消費電力の内訳の例

型・軽量であることがのぞましい．多数のセンサに対してどんな場所においても電力を有線で供給することは難しく，電池などのバッテリーで駆動することが必須となる．しかしながら，自律分散的にネットワークを構築し，他のノードを中継ノードとして通信を行うセンサネットワークにおいて，バッテリーで駆動した場合，ネットワークの寿命がセンサノードの持つバッテリーに依存することとなる．また，バッテリーの低下したノードが生じた場合においても，一般に数百から数千個のノードで構成されるセンサネットワークでは，バッテリーを交換することは現実的ではない．そのため，センサネットワークを開発する上でセンサノードにおける消費電力の削減が最大の課題となる．

また，一般にセンサネットワークでは多数のノードが高密度に配置される．センサネットワークではネットワークのノード数，通信半径，通信状況といった環境条件や，何をどのように観測するのか，というセンシングタスク，取得するデータの粒度，また遅延，消費電力といった性能などの，エンドユーザやアプリケーションの要求に応じて，適応的かつ自律分散的にネットワークを制御する必要がある．次の2.3.3において無線通信における送信距離と消費電力の関係について検討を行う．

### 2.3.3 送信距離と消費電力

図 2.10 に各状態で消費される電力の内訳を示す [17]．無線センサネットワークにおける消費電力の内訳として，データ通信によるもの，CPU で消費されるもの，センシングにおいて消費されるものなどが列挙されるが，主な要因はデータ通信によるものである．

センシングや CPU で消費される電力に比べて，送信状態・受信状態・アイドル状態といった無線通信で消費される電力が圧倒的に大きいことがわかる．例として 1bit の情報



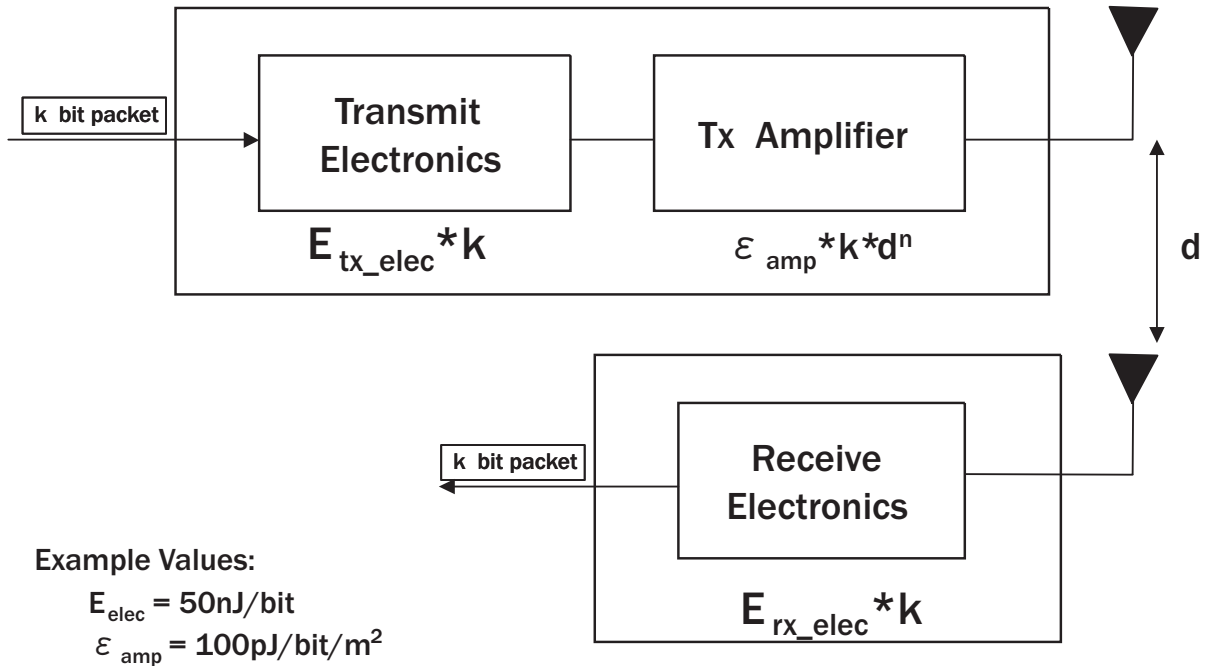


図 2.11 無線通信モデル

を 100m 先に伝送するのにかかる電力コストが，CPU で 3000 の命令を実行するのに匹敵するという報告 [18] もあり，送信・受信にかかる消費電力は演算にかかる電力よりもはるかに大きいことが分かる．このため，各ノード内でできるだけ通信のオーバーヘッドが小さくなるような制御が必要となる．

無線通信のモデルはさまざまなものがあるが，その一例として，LEACH[19] で述べられているモデルについて述べる．図 2.11 の無線通信モデルで示すように， $k$  ビットの情報を距離  $d$  にある受信器に送信するのにかかる電力，受信するのにかかる電力をそれぞれ  $E_{Tx}(k, d)$ ， $E_{Rx}(k)$  とすると，

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^\alpha \quad (2.1)$$

$$E_{Rx}(k) = E_{elec} * k \quad (2.2)$$

となる． $E_{elec}$ ， $\epsilon_{amp}$  は無線回路に特有な定数である．また， $\alpha$  は無線の電波伝搬環境に依存する．これらの関係式より，エンドツーエンドの送信ノードから宛先ノードまでの距離が同じであっても，パケット転送におけるホップ数の相違によって全体で消費される電力も変わってくる．

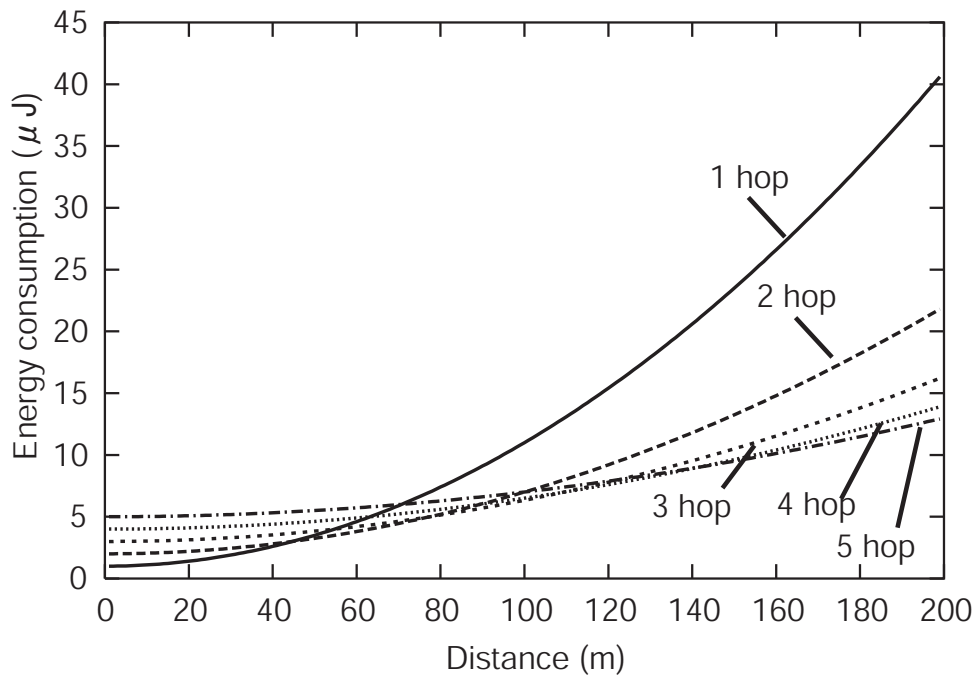


図 2.12 エンドノードまでの距離と総消費電力の関係

式 2.1 および式 2.2 を用いて図 2.12 にエンドノードまでの距離と消費電力の関係を示す。なお、 $E_{elec} = 50[nJ/bit]$ 、 $\epsilon_{amp} = 100[pJ/bit/m^2]$ 、また  $\alpha = 2$  とした。図 2.12 より、エンドノードまでの距離が比較的短いときには少ないホップ数で、また距離が長いときには多いホップ数で転送した方が、総消費電力が小さくなることが分かる。そのため、エンドノードまでの距離に応じて、適応的に通信半径やルーティングなどを制御することが必要と考えられる。

## 2.4 グローバルなセンサネットワーク

### 2.4.1 現状と展開

ローカルに存在する無線センサネットワークがゲートウェイを介すなどしてインターネットに接続されることが想定される。近年、そのような環境を意識した研究開発も行われるようになってきている。Intel の IrisNet[20] では、位置情報とセンサデータのデータベースを構築し、空き駐車場の発見などのアプリケーションを想定している。UCLA の Urbun Sensing[21] や MIT の SENSEable City Laboratory [22] が行っているプロジェクトなどの、携帯電話や PDA を用いて都市規模でのセンシングを行う研究が行われている。また Meteor プロジェクト [23] では、オーバレイネットワークを形成してセンサやアクチエー

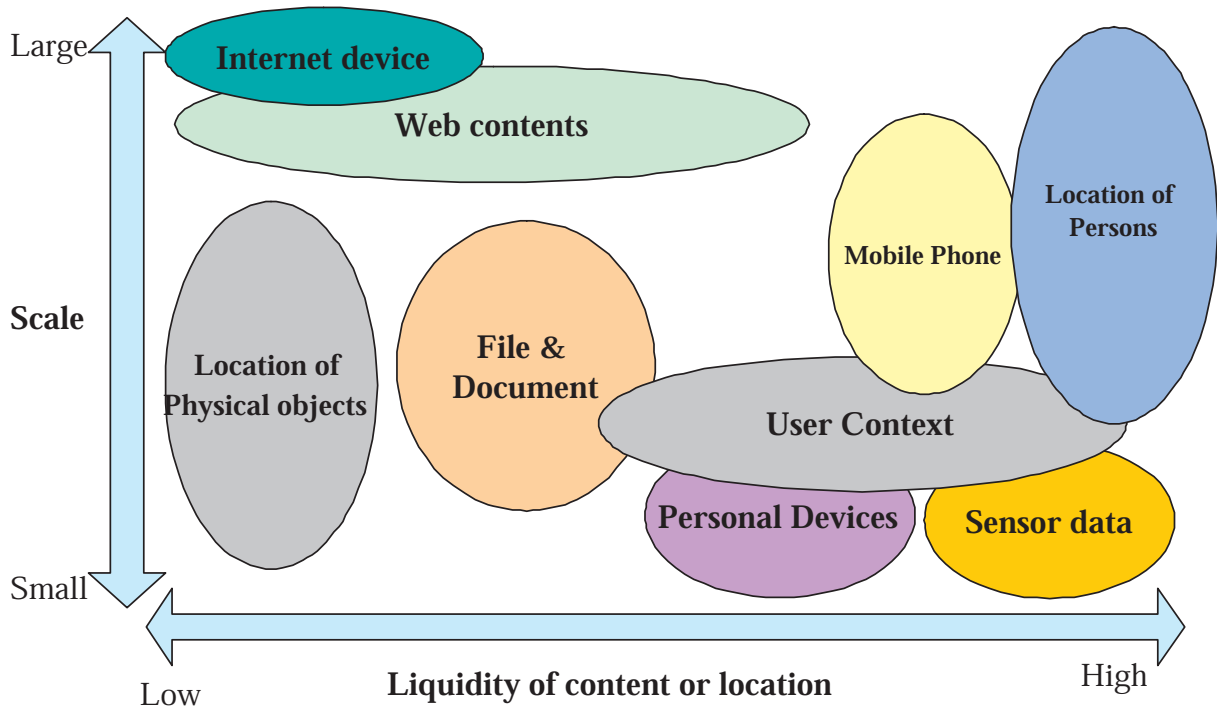


図 2.13 ネットワークリソースの時間的・空間的概念

タを接続するプラットフォームの開発を行っている。さらに，Live E! [24] ではインターネットを利用した気象データの取得などの実験が行われている，今後はインターネットや携帯電話網のような広域ネットワークを介してセンサデータが流通することが想定される。現在 Google Maps[25] のような実空間の地物情報を検索するシステムが存在するが，センサデータの取り込みにより，より動的な実空間情報の取得が可能となる。

このようにセンサデータの膨大な情報からユーザが自分にとって必要な情報を抽出するためには何らかの検索システムがなくてはならないものとなっている。ネットワーク上のハードウェア，ソフトウェアといったリソースやサービスを識別してネットワーク上の位置を特定し，名前解決を行う機構が必要となる。また，今後のユビキタス環境を想定すると，さまざまなオブジェクトがネットワークに接続することが想定され，ユーザにとっていかに自分にとって必要なリソースを発見し，利用していくかはより重要な課題になると考えられる。図 2.13 にネットワーク上のリソースを時間軸・空間軸の観点から捕らえたおおよその概念図を示す。時間軸はリソースの位置や内容の更新頻度（更新頻度が高いか低い），空間軸はリソースがネットワーク上で分散している範囲（存在範囲が広い狭いか）を表している。時間的観点からは，ウェブコンテンツといった比較的更新頻度の低いリソースからセンサデータという時々刻々とその情報が更新されるもの，また，空間的観

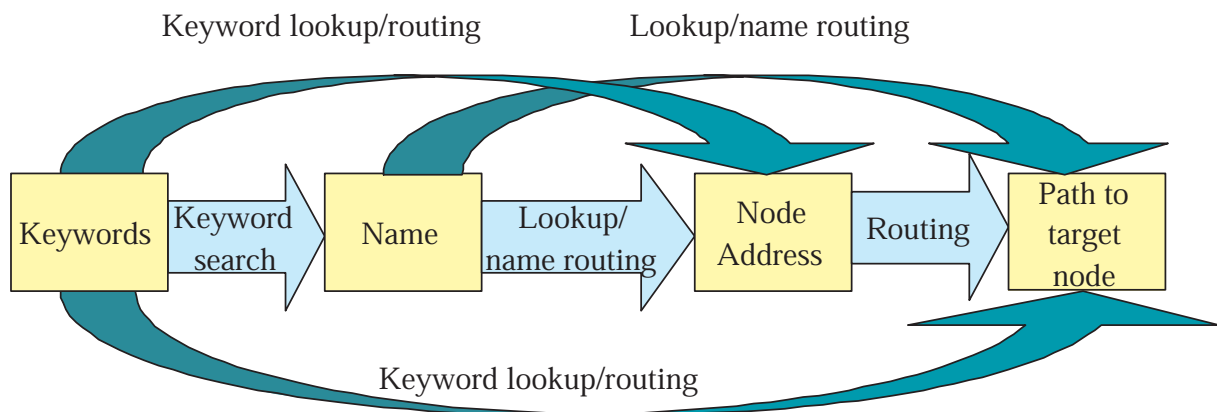


図 2.14 ネットワークリソースへのアクセスの手順

点からはインターネット上の大規模に分散するリソースから，ホームネットワーク，アドホックネットワーク，センサネットワーク上のリソースといった比較的小規模まで多種多様である．

図 2.14 上にネットワークリソースへのアクセスの手順を示す [26]．例として，ウェブ検索エンジンでキーワード検索を行って，キーワードに関連するコンテンツを取得する場合を考える．まず，ユーザは必要とするリソースをコンテンツに含まれる言葉やメタ情報に基づいたキーワードによって表現する．キーワード検索によって検索エンジンはリソースのメタ情報を複数のリソース名 (URL) などにマッピングする．リソース名は DNS による名前解決によりネットワーク上のアドレスに変換される．さらにルーティングによってエンドノードへ接続し，コンテンツを得る．これらの DNS やウェブの検索エンジンはあるサーバを儲け，リソースの位置に関する情報を集中管理している．このような集中制御を行うサーバ型システムの欠点としてサーバの障害によるネットワークの耐障害性が弱いということ，ネットワークに参加するノード数の増加に対してスケラビリティが十分でないなどの点が挙げられる．これらの問題点を考慮し，近年では中央サーバを必ずしも必要としないピアツーピア (以下 P2P) 型リソース発見機構についてさまざまな研究が行われている．このような分散環境では一元的なサーバ型リソース管理機構を持たず，各ノードがリゾルバとして機能し，図 2.14 のようにキーワードから直接リソース自体やそれに関連する情報を保持するノードを求めたり，キーワードやリソース名を用いたクエリルーティングを行うことによってリソース自体を保持するノードやリソースの場所を示すポイント情報を保持するノードを発見する．

### 2.4.2 要求事項と技術的課題

実空間から得られる環境情報，災害情報，また交通情報など多種多様なセンサデータは，データの容量，更新頻度もさまざまである．都市部などの人間活動が活発なところでは得られるセンサのデータ量や更新頻度も高くなる可能性がある．また，災害発生時には一時的にデータの更新頻度や検索数が高くなる可能性もある．このように空間的，時間的に変動するトラフィックに適応的に対処できる必要がある．

データを集中管理する方式ではアクセスやデータ処理の負荷が集中するノードの故障に対するロバスト性が脆弱であること，ストレージ容量が不足する可能性があること，またデータ登録のオーバーヘッドが大きくなることなどの欠点がある．一方，データをすべてローカルのノードに保持する方式では検索のオーバーヘッドが大きくなるため，ネットワーク上のすべてのノードを探索してユーザやアプリケーションが必要とするデータを発見するのは現実的でない．データの集約による冗長なデータ配信の削減や，データ処理が行えないこと，などの欠点がある．両者の欠点を補うべく，アプリケーションの要求に応じて要求に応じて適応的に分散管理することが課題となる．分散管理の利点としては以下の点が挙げられる．

1. 耐障害性集中的にリソースを管理するノードが不要なため，一部のノードの機能停止によってシステム全体が大きな影響を受けにくい．
2. 規模拡張性ノード数の増加に対して管理する情報や発見のためのホップ数の増加率を抑制できる．
3. 柔軟性目的に応じて比較的簡単にオンデマンドでネットワークを構築できる．

一方，欠点としては，一般に検索のためのオーバーヘッドが一元管理と比較して増加するなどが挙げられる．

センサネットワークではある特定の端末をユーザやアプリケーションが明示的に指定するというよりはむしろある特定のコンテンツを指定して，そのコンテンツを保持する不特定多数の端末と通信することが一般的であると考えられる．P2P ネットワークではアプリケーション層の要求に応じて柔軟に論理的なネットワークを構成することができる．そのため，適していると考えられる．本論文ではデータの地理的位置を限定して検索を行うことを想定する．一般に基本的なDHTアルゴリズムでは，一様なキーワード空間に対してほぼ一様な分布で出現するハッシュ値を利用してデータの管理を行うが，キーワード空間に偏りがあると，データの管理する量に偏りが生じる可能性がある．センサデータを

扱う場合，発生するデータの地理的位置の分布，更新頻度，データに対するアクセス頻度には一般には偏りが生じる場合がある．P2P ネットワークは処理能力の異なる端末から構成される可能性もあり，処理能力の低い端末に負荷が集中してしまう可能性もある．アクセス頻度の高いコンテンツには負荷分散を行うことで効率の良いデータの検索や管理を行う．時々刻々と変化する災害状況から最適な経路を提供災害地や人気のあるエリアへのアクセスが集中する可能性もある．そのため，センサデータの管理，発見，また収集をスケーラブルに行うメカニズムが必要となる．

### 2.5 本章のまとめ

本章では実空間情報の収集に向けた無線環境およびインターネット環境のセンサネットワークの展開について述べるとともに，要求事項と技術的課題を明らかにした．以降の章では，その技術的課題を掘り下げて，具体的な解決策について議論を行う．

### 第3章 適応的スロット予約による低消費電力メディアアクセス制御

---

### 3.1 本章の概要

実空間の環境情報を収集する上で、無線センサネットワークは配置が簡易なことやコストが低いことなどから極めて有効な技術である [27]。一般に多数のノードから構成されるため、個々のノードのバッテリーを交換することはコストが大きく、ネットワークを長期間稼働させるために、消費電力を抑制することが重要な課題となる [18]。特に、実際にパケットの送受信を行わなくとも、無線インターフェースがアクティブ状態であれば電力を消費するため、一般に通信を行う期間が比較的短いセンサネットワークにおいては、送受信を行わない期間は、極力無線インターフェースはスリープ状態になることが望ましい。そのため、チャンネルがビジー状態かアイドル状態を判断し、チャンネルアクセスを制御する MAC プロトコルで無線インターフェースの制御を行うことが有効である。

センサネットワークでは、通常はトラフィックがほとんどなく、イベント検知、クエリの配信などの際に、一時的にノードがデータの送受信を行う場合がある。また、異なる位置に配置されたノードによってもトラフィックが異なる場合もある。そのため、メディアアクセス制御のプロトコル設計の上で、時間的、空間的なトラフィックの相違に対して、柔軟に対応できることが重要であると考えられる。一般に TDMA 方式では、CSMA 方式と比較して、ノードのランダムアクセスによる冗長なアクティブ期間が少なく、通信時の消費電力が抑制される。従来の TDMA 方式の MAC プロトコルでは、予め割り当てられたスロットに送受信ノードがデータパケットの送受信を行うために、CSMA 方式と比較して冗長なアクティブ期間が削減され、消費電力を抑制することができる。しかしながら、従来の TDMA 方式では、ノードがトラフィックの高低に依存せずスロット予約の制御を行うため、トラフィックが低い場合は、スロット予約に関するオーバーヘッドが相対的に増大する。さらに、スロット予約に関する制御パケットを交換する期間が固定長であるために、消費電力を浪費するという課題がある。

そこで本章では、新たにセンサネットワークにおける TDMA 方式の MAC プロトコルの提案を行う。提案手法では、データパケットの発生時に送受信ノードのみがスロット予約を行うことで、スロット予約に関する制御パケットのオーバーヘッドを削減する。また、各ノードがスロット予約期間におけるアクティブ期間を動的に制御することによって、トラフィックが低いときに省電力化し、トラフィックが高いときにスループットを維持する。

計算機シミュレーションによる評価を行った結果、提案手法では、従来手法と比較して、スループットが確保されつつ消費電力が削減されるために、消費電力の効率が向上することが分かった。

以降では、まず 5.2 で、センサネットワークを対象にした MAC プロトコルの関連研究



の概要と問題点について述べる。次に5.3で、提案手法の説明を行い、続く5.4で、提案手法の性能評価を行う。最後に5.5で、本章のまとめと今後の課題について述べる。

## 3.2 関連研究

近年、センサネットワークを対象とした、さまざまなMACプロトコルが考案されている。以下では、省電力化についての関連研究をCSMA方式[28]~[32]とTDMA方式[19]~[35]に分類して述べ、TDMA方式については、スロット割り当ての手法も紹介する。

### 3.2.1 CSMA方式

CSMA方式として、周期的にアクティブ期間とスリープ期間を遷移させることにより、省電力化を図るS-MAC[28]、また、S-MACの拡張的な機能として、データ送受信終了後に再び一定期間アクティブ状態になることでパケット送受信の機会を増やすadaptive listen[29]も提案されている。S-MACでは、1周期の長さに対して、最低限アクティブ状態になる期間の長さの割合であるduty cycleが低く設定されることで、消費電力が抑制される。また、1周期におけるアクティブ期間の長さをトラフィックによって動的に制御するT-MAC[30]やduty cycleを動的に制御するDSMAC[31]、定期的に送信ノードのプリアンブルを検出して送受信を行うB-MAC[32]などが提案されている。CSMA方式の問題点として以下の点が挙げられる。まず、冗長なアクティブ期間が増加しやすいことである。CSMA方式ではキャリアセンスやバックオフが必要となるので、自分に無関係なパケットの送受信を行う場合があり、消費電力を浪費する可能性がある。一般に、消費電力やスループットの性能は、duty cycleに依存する。duty cycleが高い場合、送受信の機会が増加し、スループットは高くなるが、消費電力が増大する可能性がある。一方、duty cycleが低い場合、消費電力は抑制されるが、トラフィックが増加すると、限られたアクティブ期間にアクセスが集中し、スループットが低下する可能性がある。さらに、RTS/CTSのハンドシェイクを行う場合、データパケット送信ごとにRTS/CTSの送信が必要となるので、比較的データパケットの大きさが小さいセンサネットワークでは、制御パケット送受信の相対的なオーバーヘッドが増加する可能性がある。

### 3.2.2 TDMA方式

TDMA方式として、LEACH[19]、TRAMA[33]、TDMA service [34]、またLMAC[35]などが提案されている。一般にTDMA方式では、データパケットの送受信は予め割り当

てられたスロットに行われるため、CSMA方式と比較して、データパケットの送受信に伴う冗長なアクティブ期間が少なく、消費電力は抑制される。LEACHでは、ネットワーク内でクラスタリングが行われ、各クラスタの代表となるノードが、クラスタ内のノードにスロットを割り当てる。このようなクラスタリングを行う方式では、ノードの故障によるトポロジの変化に対するロバスト性の確保が容易ではないという課題がある。TRAMAでは、各ノードが自分から2ホップの範囲内に存在する近隣ノードのトポロジ情報を把握し、各スロットにおける近隣ノードの優先度を計算する。また、定期的なトラフィック情報の交換によって、動的にスロットを割り当てる。課題として、データパケットの送受信がほとんどないときは、スロット割り当ての制御のオーバーヘッドが増大することが挙げられる。具体的には、近隣ノードのトポロジ情報を交換するシグナリングパケットや、自分の送信するスロットと他のノードに開放するスロットに関する情報を記述したスケジュールパケットを、定期的に隣接ノードにブロードキャストする。そのため、データパケットのトラフィックが低いときは、計算コストや制御パケットの送受信に伴う消費電力が相対的に大きくなる可能性がある。さらに、電波干渉の問題が存在する。受信ノードが保持する近隣トポロジ情報に含まれないノードであっても、受信ノードに対して、電波干渉を及ぼすことがある。その結果、2ホップ以内の近隣ノードのトポロジ情報を利用してスロットを割り当てる手法は、不十分であることが指摘されている [36]。特にスケジュールアクセス期間のトラフィックが増加すると、干渉によるパケットの損失が増加する可能性がある。また TDMA service では、電波干渉を考慮してスロット割り当て制御が行われている。しかしながら、実際のセンサネットワークが利用される環境では、ノードが固定的に配置される場合にも、ノードの追加や離脱、電波伝搬の不規則性 [37] などが存在するため、一般に干渉を及ぼすノードを含めた近隣ノードのトポロジ情報を常時正確に把握し、スロット割り当ての整合性を保つことは困難であると考えられる。また LMAC では、あるノードに1周期に割り当てられるスロット数が固定的であるため、トラフィックの変動に対する動的なスロット割り当てについて考慮されていない。

### 3.3 提案手法

#### 3.3.1 概要

提案手法では、送信するデータパケットが発生したときのみ、送受信ノード間でスロット予約が行われる。このスロット予約によって、定期的に制御パケットを交換してスロット予約を行う場合より、特に低トラフィック時におけるスロット予約の計算や制御パケッ

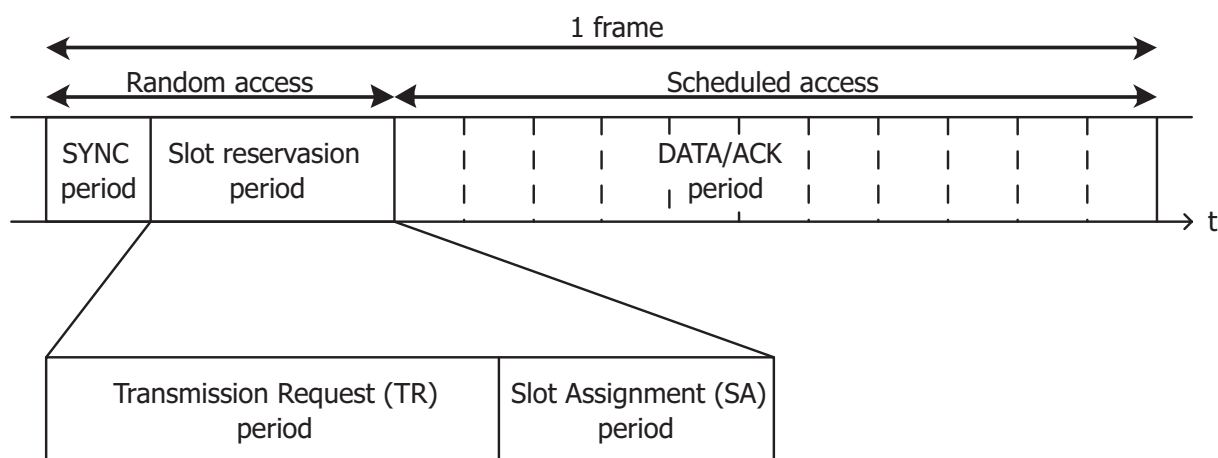


図 3.1 1 フレームにおけるタイムスケジュール

トのオーバーヘッドを削減する．さらにスロット予約期間におけるアクティブ期間を，トラフィックによって動的に制御する．この適応的制御により，トラフィックが低いときは，アクティブ期間を短縮することで省電力化する．一方，トラフィックが高いときは，アクティブ期間を延長してスロット予約に必要な制御パケットの送受信を行い，スループットの低下を抑制する．

前提として，利用する無線周波数は単一であるとする．各ノードはネットワークに参加するとき，隣接ノードとお互い同一のタイムスケジュールで同期をとる．従来の TDMA 方式で利用されている同期手法の利用が可能である．本章は Global Schedule Algorithm (GSA) [38] を利用する．GSA では，理論的にネットワーク全体で同じタイムスケジュールの同期を取ることが可能である．

提案手法では，隣接ノードに対してユニキャストとブロードキャストを行うことが可能である．図 3.1 に提案手法の 1 フレームにおけるタイムスケジュールを示す．1 フレームは，CSMA/CA (Collision Avoidance) で送受信が行われるスロット予約期間と，タイムスロットで送受信が行われるスケジュール期間から構成される．CSMA/CA の最大バックオフ時間は固定とし，その最大バックオフ時間の中で一様分布の乱数を発生させてバックオフ時間を決定する．スロット予約期間は，送信要求 (Transmission Request, 以下 TR) 期間と，スロット割り当て (Slot Assignment, 以下 SA) 期間から構成される．TR 期間は，送信ノードが受信ノードに対して送信要求を行う期間である．また，SA 期間は，受信ノードが送信ノードに対してスロットを割り当ててる期間である．SA 期間，TR 期間の最大長は，それぞれの期間にノードが十分に TR パケット，SA パケットの送受信を行うことが十分可能な長さに設定される．1 フレームの長さによって，スロット予約を行う間

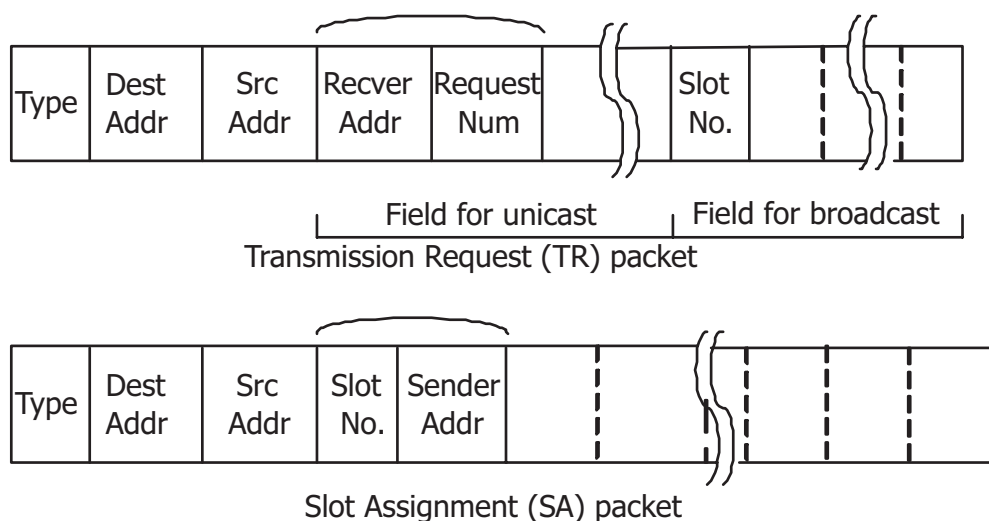


図 3.2 送信要求 (TR) とスロット割り当て (SA) の各パケットのフォーマット

隔が異なってくる．一般にスロット予約を行う間隔が短ければ，データパケットが発生してから送信するまでの遅延は小さくなるが，アクティブ期間が増加するため，消費電力は大きくなる．一方，スロット予約を行う間隔が長ければ，遅延は大きくなるが，消費電力は小さくなる．このように1フレームの長さによる消費電力と遅延は一般にトレードオフの関係にある．データパケットは発生した順にキューに格納される．データパケットが発生した直後のフレームで，以下に述べる送信要求，スロット割り当て，またデータパケットの送受信のハンドシェイクが行われる．そのフレームでデータパケットの送受信に失敗した場合は，次のフレームでハンドシェイクが再試行される．TR 期間に送信される TR パケットと，SA 期間に送信される SA パケットの主なフォーマットを図 3.2 に示す．TR パケットには受信ノードのアドレスとそのノードに対するユニキャストデータパケット送信要求数の組，またブロードキャストの場合には自分が送信するスロット番号などが記述される．一方，SA パケットには，ユニキャストの際割り当てるスロット番号と送信ノードのアドレスの組などが記述される．

提案手法では，どの隣接ノード間でもスロット予約を行うことが可能である．また，近隣ノードのトポロジー情報ではなく，送信ノードの送信要求に基づいてその都度，スロット予約が行われる．そのため，送受信を行わないノードの故障がしても，スロット予約に影響を及ぼさない．

以下では，スロット予約の制御に関して 3.3.2 項で送信要求の手順について，3.3.3 項でスロット割り当ての手順について，述べる．また，3.3.4 項で実際のデータパケットの送受信の手順について述べる．さらに，3.3.5 項でスロット予約期間におけるアクティブ期

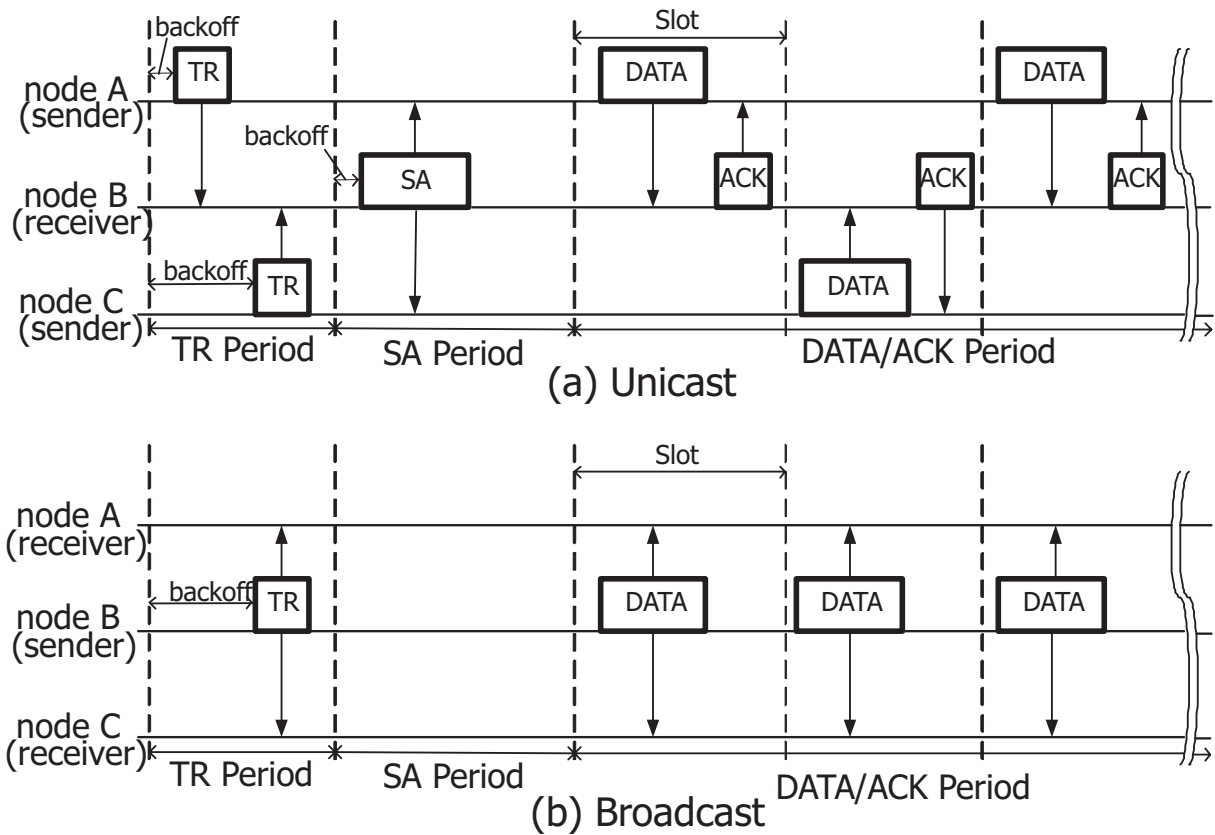


図 3.3 ユニキャスト・ブロードキャストにおける動作例

間の適応的制御について述べる．最後に，3.3.6 項で制御パケットの集約の効果について述べる．

### 3.3.2 送信要求の手順

送信ノードは TR パケットの送信を TR 期間に試行する．CSMA/CA によりバックオフ制御を開始し，バックオフタイムが 0 になり送信権を得ると，以下の処理を行う．まず，ユニキャストを行う場合，バッファに格納されているユニキャストデータパケットを検査して，宛先ノードのアドレスと，そのノードに対して送信したいユニキャストデータパケットの個数を TR パケットに記述する．ブロードキャストを行う場合，送信ノードが現時点で空いていると判断したスロットの中からランダムにスロットを各ブロードキャストデータパケットに割り当て，それぞれのスロット番号を TR パケットに記述する．そして TR パケットを送信する．隣接ノードから自分宛ての TR パケットを受信した場合は，TR パケットに記述された送信したいユニキャストデータパケットの個数や，ブロードキャスト

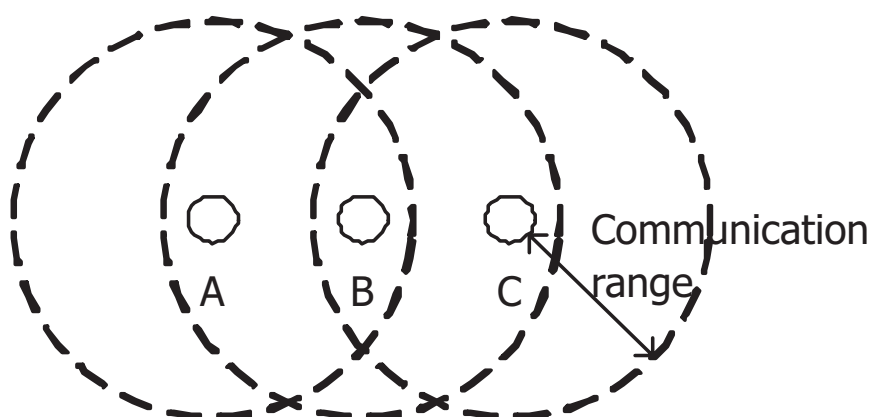


図 3.4 動作例におけるネットワークポロジ

トが行われるスロット番号を把握する。

ユニキャストとブロードキャストの動作例を図 3.3 に示す。また、その動作例におけるネットワークポロジを図 3.4 に示す。図 3.3(a) のユニキャストの例では、送信ノードであるノード A とノード C が、受信ノードであるノード B に対して TR パケットを送信する。また、図 3.3(b) のブロードキャストの例では、送信ノードであるノード B が受信ノードであるノード A とノード C に対して TR パケットを送信する。

### 3.3.3 スロット割り当ての手順

ユニキャストを行うための TR パケットを送信または受信したノードは、SA 期間に SA パケットを送受信することによってスロット割り当てを行う。それ以外のノードは、SA 期間においてすべてスリープ状態となる。受信ノードは 3.3.2 項の送信要求の手順で述べたのと同様に、CSMA/CA のバックオフ制御を開始し、バックオフタイマが 0 になると以下の処理を行う。受信ノードは各送信ノードに対して、要求されたスロットの割り当て数に応じて、空きスロットの中からランダムにスロットを選択して割り当てる。受信ノードは送信ノードのアドレスとスロット番号を対にして記述し、SA パケットを送信する。このように、受信ノードが送信ノードにスロットを割り当てることで、隠れ端末の関係にあるノード同士が同じスロットに送信することを防止する。SA パケットを受信したノードは、ユニキャストのデータパケットの送受信を行うノードのアドレスとスロット番号を把握する。

図 3.3 の例では (a) のユニキャストの場合、ノード B がノード A とノード C に対して、SA パケットを送信する。

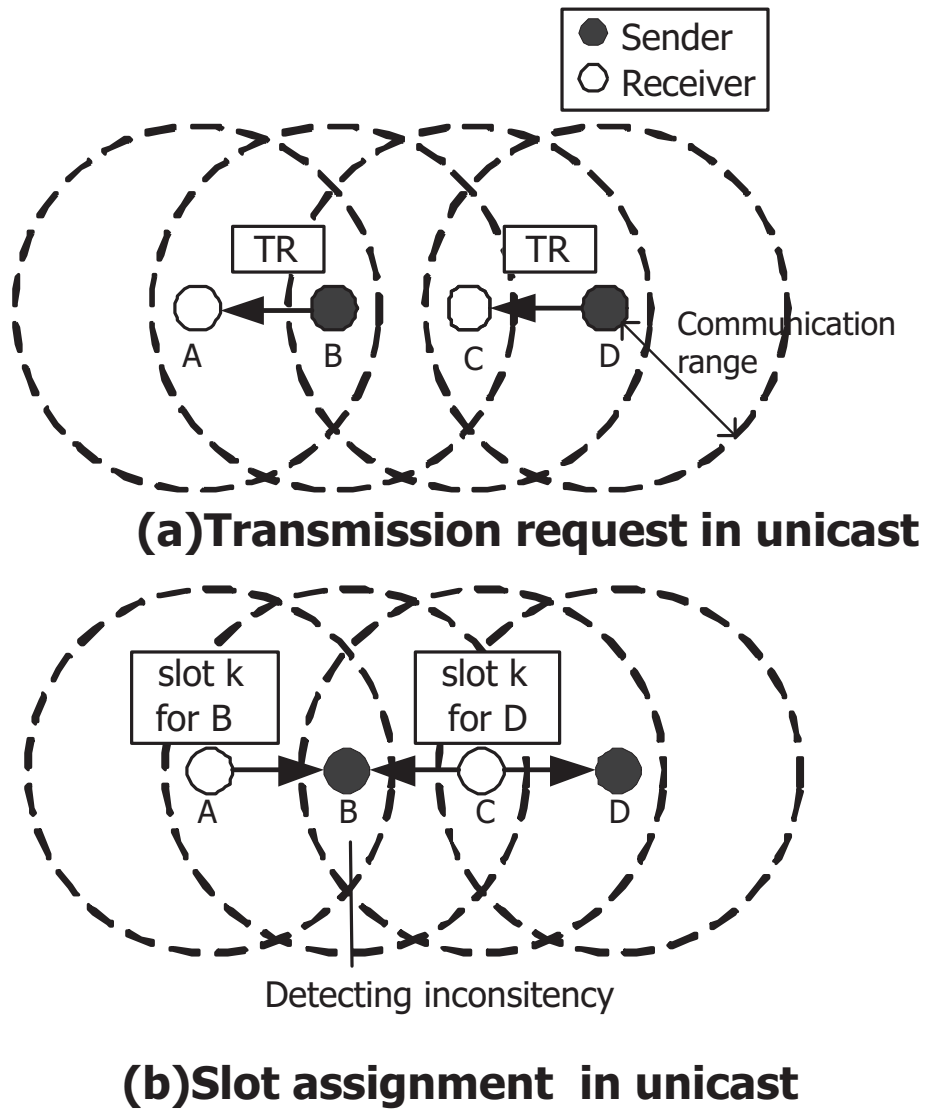


図 3.5 スロット予約における不整合の発生の例

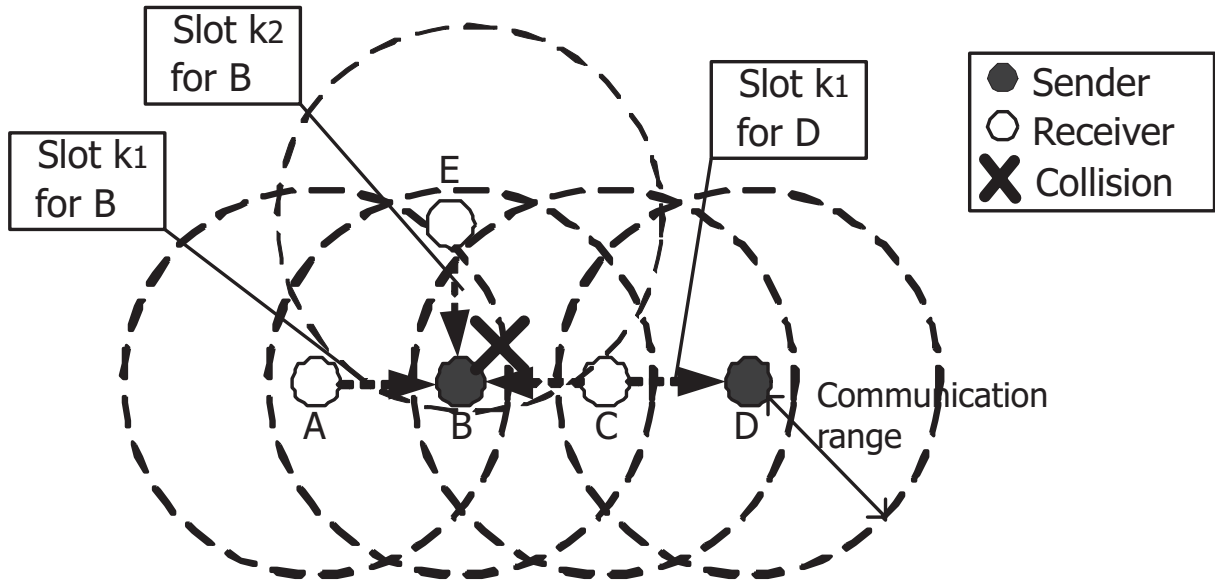
### 3.3.4 データパケットの送受信の手順

3.3.2項と3.3.3項で述べた送信要求とスロット割り当ての情報に基づいて、スケジュールアクセス期間に送受信ノードはデータパケットの送受信を行う。送信ノードは、自分が獲得した各スロットの受信ノードに対して、早くキューに格納されたデータパケットから順に送信する。ユニキャストの場合は、データパケットの受信を確認できた受信ノードはACKパケットで送信ノードに対して応答する。

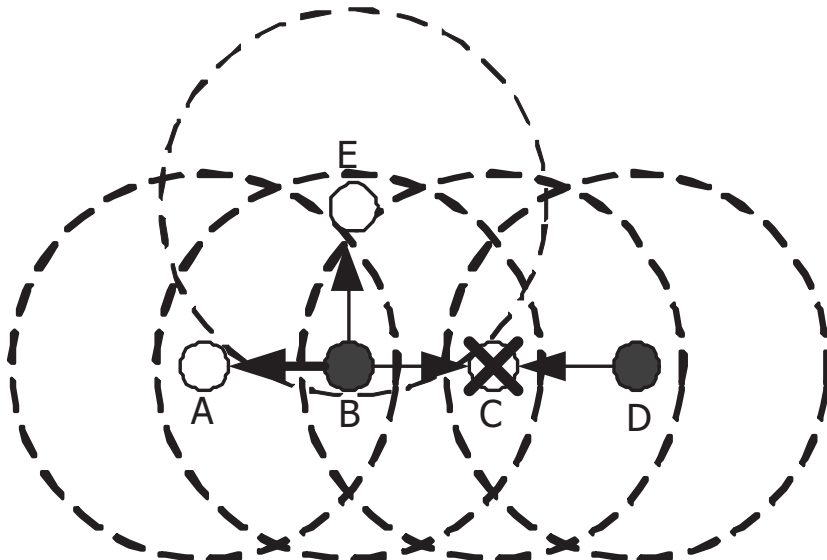
図3.3の例では(a)のユニキャストの場合、SAパケットに割り当てられたスロットにおいて、ノードAとノードCはノードBに対してデータパケットを送信し、ノードBはデータパケットの受信に応じてACKパケットを送信する。一方、図3.3(b)のようにブロードキャストの場合は、送信ノードであるノードBがTRパケットにより広告したスロットにおいて、データパケットの送信のみ行われる。なお、自分が送受信を行わないスロットはすべてスリープ状態となる。

提案手法では、同じスロットにおいて、ある受信ノードに影響する複数の送信ノードが送信権を得るスロット割り当ての不整合が生じる可能性がある。図3.5に、スロット割り当てにおける不整合が発生する場合の具体例を示す。ユニキャストの場合、図3.5(a)のTR期間において、ノードBはノードAに対して、またノードDはノードCに対してそれぞれ送信要求を行う。図3.5(b)のSA期間においてノードAがノードBに対してスロット $k$ を、ノードCがノードDに対してスロット $k$ を割り当てた場合、スロット割り当ての不整合が生じ、もしノードBとノードDがスロット $k$ で送信すると、ノードCにおいて両者のデータパケットが衝突してしまう。このため、SAパケットによって不整合を検知したノードBは、スロット $k$ にパケットを送信を行わずスリープ状態となることで、データパケットの衝突を防止する。ブロードキャストの場合、TRパケットの受信によってスロット割り当ての不整合を検知したノードは、不整合が生じるスロットではスリープ状態となる。もし、送受信ノードが、隣接ノードからのTRパケットやSAパケットを損失なく受信できれば、スロット割り当ての不整合に起因するパケットの衝突は生じない。しかしながら、制御パケットの損失により、スロット割り当てに関する情報の損失が起こった場合は、上記で述べた不整合を送受信ノードが検知できずに、データパケットの衝突が生じる可能性がある。その例を図3.6に示す。図3.6(a)のSAパケット送信の際、まずノードBはノードAからスロット $k_1$ に送信権を与えられる。ノードCがノードDに対して、ノードEがノードBに対してそれぞれスロット $k_1, k_2$ を割り当てるためのSAパケットを送信する。その際、送信が同時に起こると、ノード2で両者のSAパケットが衝突してしまう。これにより、図3.6(b)のDATA/ACKパケット送受信の際、ノードBはスロット





**(a) Slot assignment in unicast**



**(b) Data packet collision (in slot k1)**

図 3.6 制御パケット損失によるデータパケット衝突の発生例

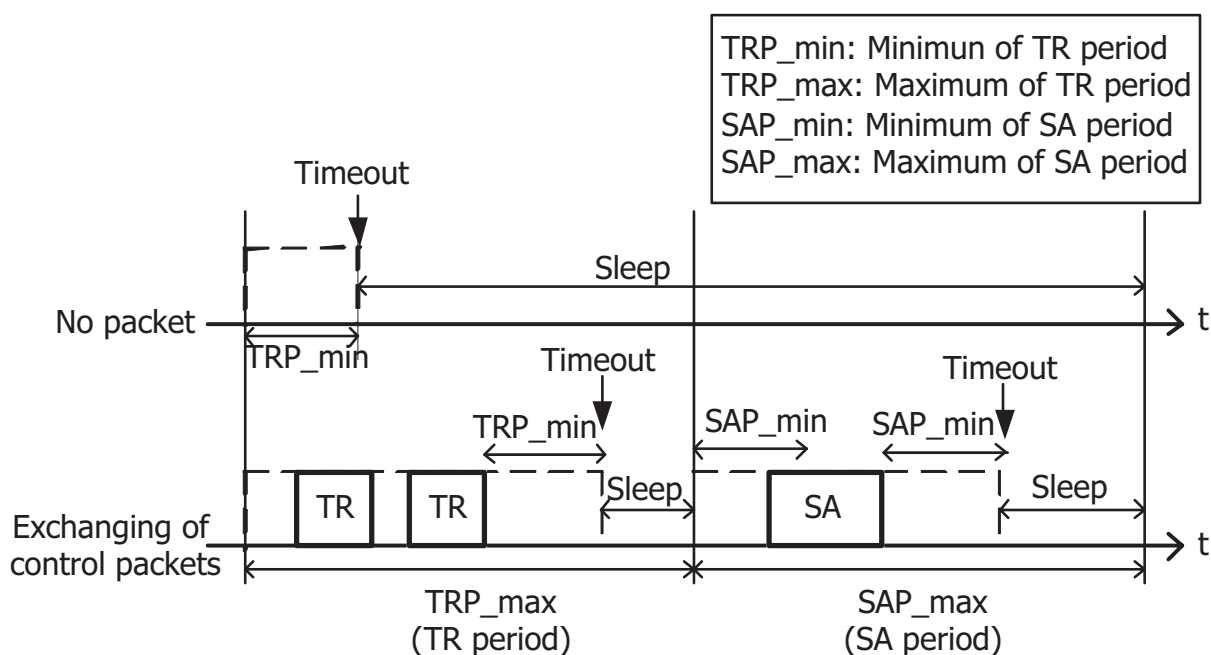


図 3.7 スロット予約期間の適応化制御

$k_1$  でノード D が送信するという不整合の発生の情報を得られないため、ノード B ではスロット  $k_1$  でデータを送信してしまい、ノード C でノード B とノード 4 が送信した DATA パケットの衝突が起きる。

また、同じスロットにおいて、ある受信ノードに対して送信ノード以外に、干渉を及ぼすノードが送信すれば、データパケットは損失する。損失したデータパケットは次に割り当てられたスロットで再送される。

### 3.3.5 スロット予約期間の適応的制御

図 3.7 のように、各ノードはまず、共通かつ固定長の最大 TR 期間と最大 SA 期間を設定する。また、それぞれの期間の長さを  $TRP_{max}$ 、 $SAP_{max}$  とする。これにより、各ノードで TR 期間と SA 期間の各開始時と各終了時は共通となる。TR 期間の開始から、少なくとも 1 個の TR パケットの送受信の検出が可能となるまでの期間を最小 TR 期間とし、その長さを  $TRP_{min}$  とする。 $TRP_{min}$  は MAC 層での CSMA/CA の最大バックオフ時間と、パケットの送信開始から受信検出までの遅延時間の和となる。各ノードはまず、最小 TR 期間の間、アクティブ状態となる。もし最小 TR 期間に TR パケットの送受信を検出しなければ、最小 TR 期間の終了時のタイムアウトによって、残りの TR 期間ではスリープ状態となる。一方、もし最小 TR 期間に TR パケットの送受信を検出すると、そのパケット

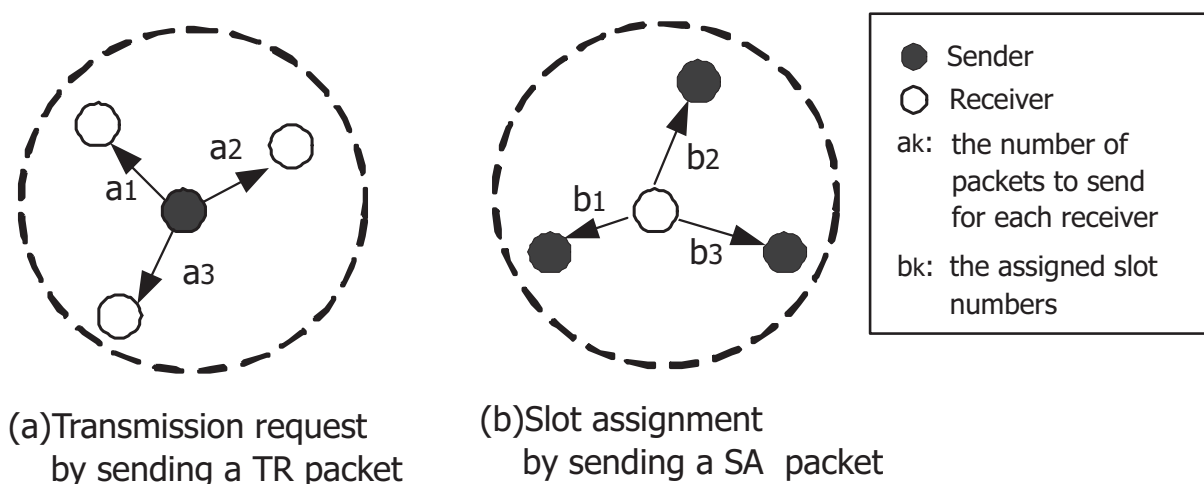


図 3.8 制御パケットにおける情報集約

の送受信後，さらに  $TRP_{min}$  の長さ分，アクティブ期間を延長する．延長された期間においても同様に，TR パケットの送受信の有無によってアクティブ期間の長さを変更する．TR 期間におけるアクティブ期間は，最長  $TRP_{max}$  となるまで延長される．一方，SA 期間では，SA パケットを交換しないノードは，全 SA 期間の間，スリープ状態となる．TR 期間と同様に，SA 期間においても最小 SA 期間を設定し，その長さを  $SAP_{min}$  とする．そして，SA パケットの交換を行うノードは，SA パケットの送受信の有無によって，SA 期間におけるアクティブ期間の長さを変更する．SA 期間におけるアクティブ期間は，最長  $SAP_{max}$  となるまで延長される．

### 3.3.6 制御パケットにおける情報集約の効果

センサネットワークでは，2. で述べたように，送信するデータパケットの大きさが比較的小さいため，個々のデータパケット送受信ごとに行う制御パケットの相対的なオーバーヘッドが大きくなるという課題がある．センサネットワークでは，そのデータ伝送の特徴により，ネットワーク内のあるノードがある任意のノードと通信する一対一通信の形態よりも，あるノードが複数のノードと通信する一対多通信や，複数のノードがある特定のノードと通信する多対一通信の形態が比較的多いと考えられる．図 3.8(a) に示すように，提案手法では，送信ノードが 1 回の TR パケットを送信することによって，複数のスロットを要求することが可能である．また，図 3.8 (b) に示すように，受信ノードが 1 回の SA パケットを送信することによって，複数のスロットを割り当てることが可能である．その結果，1 回の TR パケットや SA パケットの交換によって複数のデータパケットの送受信

が可能となる．このように，提案手法では RTS/CTS のハンドシェイクのように，個々のデータパケット送受信ごとに制御パケットの交換を行う必要がない．そのため，1 フレームの間に複数のデータパケットを送受信する場合は，RTS/CTS のハンドシェイクより，制御パケットのオーバーヘッドは抑制される．

### 3.4 性能評価

#### 3.4.1 概要

提案手法の MAC プロトコルの性能評価を計算機シミュレータ ns-2[39] を用いて行った．70m 四方の空間に通信距離 20m 干渉距離 40m のノードを 50 個配置し (1 個のシンクノードを含む)，各ノードが一定時間 (3000 秒) の間，ランダムにデータを発生し，空間領域の中央に存在するシンクノードまでユニキャストする場合において，2 章で述べた CSMA 方式の S-MAC と TDMA 方式の TRAMA と性能を比較した．主なパラメータの値を表 1 に示す．送信時，受信時およびアイドル時，またスリープ時における単位時間あたりの

表 1 メディアアクセス制御の性能評価におけるシミュレーション諸元

Parameter	Value
Communication range	20 m
Interference range	40 m
Tx power	24.75 mW
Rx/Idle power	13.5 mW
Sleep power	15 $\mu$ W
Buffer size	30 packets
Max retry limit	5
Payload length	36 byte
Bandwidth	40 kbps
Duty cycle (#1)	10%,2%
Slots per frame (#2,#3)	100 slots
Slot length (#2,#3)	52 ms
$TRP_{min}$ (#3)	42 ms
$TRP_{max}$ (#3)	500 ms
$SAP_{min}$ (#3)	42 ms
$SAP_{max}$ (#3)	500 ms

#1: S-MAC #2: TRAMA #3: Proposed method

各消費電力を，RF Monolithics 社製の無線機 TR1000 [40] の仕様に基づいて，それぞれ 24.75mW，13.5mW，また 15  $\mu$ W とした．なお，平均隣接ノード数は 9.6，各ノードからシンクノードまでの平均ホップ数は 2.0 であり，ネットワーク内には隠れ端末の関係にあるノードが存在した．前提として，各 MAC プロトコルはネットワークで同一のスケジュールで同期をとっているものとした．また，ルーティングは，隣接ノードのうちシン

クノードに最も物理的に近い位置にあるノードにパケットを転送していく方式を用いた。S-MAC の場合，adaptive listen[29] を行うものとした。また，各プロトコルにおいて，送信ノードは受信ノードの ACK パケットによりデータパケット受信の成否を確認し，ACK パケットを受信しなかった場合は再送を行うものとした。データパケットのペイロード長を 36 バイトとした。TRAMA と提案手法において，1 フレーム内のスロット数は 100 とした。各ノードでのデータパケットの発生間隔は指数分布に従うものとした。トラフィックの変動に対する各 MAC プロトコルの性能を評価するために，平均発生間隔を 0.001 パケット毎秒から 0.10 パケット毎秒まで変動させた。スロット予約期間における平均アクティブ期間特性，制御パケット送信数に関する特性，データ伝送量に関する特性，平均消費電力特性，消費電力効率特性，またエンドツーエンドの伝送遅延特性をそれぞれ測定した。

次に，フレーム長を変化させたときの特性をそれぞれ測定した。

また，一部のノードのみが送信を行うときの影響を調べるため，エリア内に存在するセンサの一部のノードのみが送信を行うとし，エリア内で送信を行うノード数を変化させたときの各特性をそれぞれ測定した。送信を行うノードの平均送信間隔は一定 (5.0 秒) とし，送信を行うノードエリア全体に対して，送信を行うエリアの割合を変化させた。送信を行うエリア範囲にノードが存在する場合は送信を行い，それ以外の範囲に存在するノードは送信を行わないものとする。

さらに干渉距離の違いによる特性をそれぞれ測定した。評価項目の概要とその結果について，以下に述べる。

#### 3.4.2 スロット予約期間における平均アクティブ期間特性

提案手法における TR 期間と SA 期間の 1 ノード 1 フレームあたりのアクティブ期間の平均長をそれぞれ測定した。図 3.9 にスロット予約期間における平均アクティブ期間特性を示す。

図 3.9 より，データ発生率が低いときには TR パケットや SA パケットの送受信数も少なくなるため，3.3.5 項で述べた適応的制御によって，各期間におけるアクティブ期間は抑制されていることが分かる。一方，データ発生率の増加とともに，TR パケットや SA パケットの送受信数も増加するため，TR 期間，SA 期間のそれぞれのアクティブ期間は増加していることも分かる。どのノードも最小 TR 期間の間，必ずアクティブ状態となるのに対し，SA 期間では，データパケットの送受信に無関係なノードは全 SA 期間の間，スリープ状態となる。さらに，データパケットの送信ノードの数より受信ノードの方

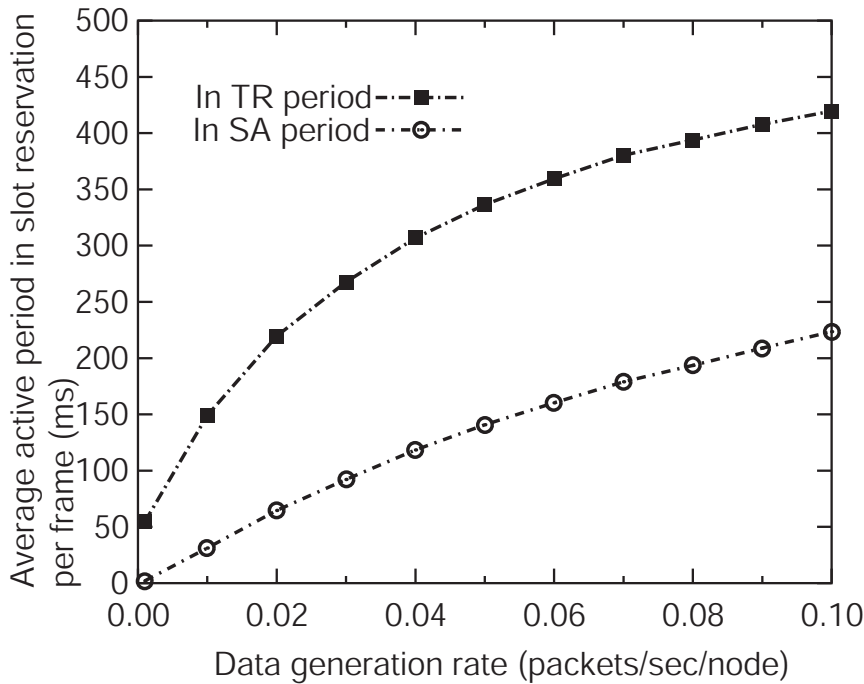


図 3.9 スロット予約期間における平均アクティブ期間特性

が少なく、また TR パケットの送信数より、SA パケットの送信数の方が少ない。その結果、SA 期間におけるアクティブ期間は TR 期間におけるアクティブ期間より短いことが分かる。

### 3.4.3 制御パケット送信数に関する特性

図 3.10 に MAC 層における 1 データパケット送信あたりの各制御パケットの平均送信数特性を示す。なお、S-MAC の duty cycle が 2 % の場合は、後述する 3.4.4 のデータ伝送量に関する特性で述べるように、データパケット送受信の性能が十分でないため、比較の対象外とした。

データパケット発生率が増加すると、隠れ端末や電波干渉の影響などにより、RTS パケットが損失しやすくなる。そのため、RTS パケットの再送数が増加し、図 3.10 より、1 データ送信あたりの RTS パケット送信数が 1 より大きくなることが分かる。提案手法の場合も、TR パケットの損失による再送によって、データ発生率が 0.01 パケット毎秒までは、1 データパケット送信あたりの TR パケット送信数が増加していることが分かる。しかしながら、データパケット発生率が増加すると、一度の TR パケットの送信によって複数のデータパケットを送信できるという、制御パケットの集約の効果が大きくなる。この

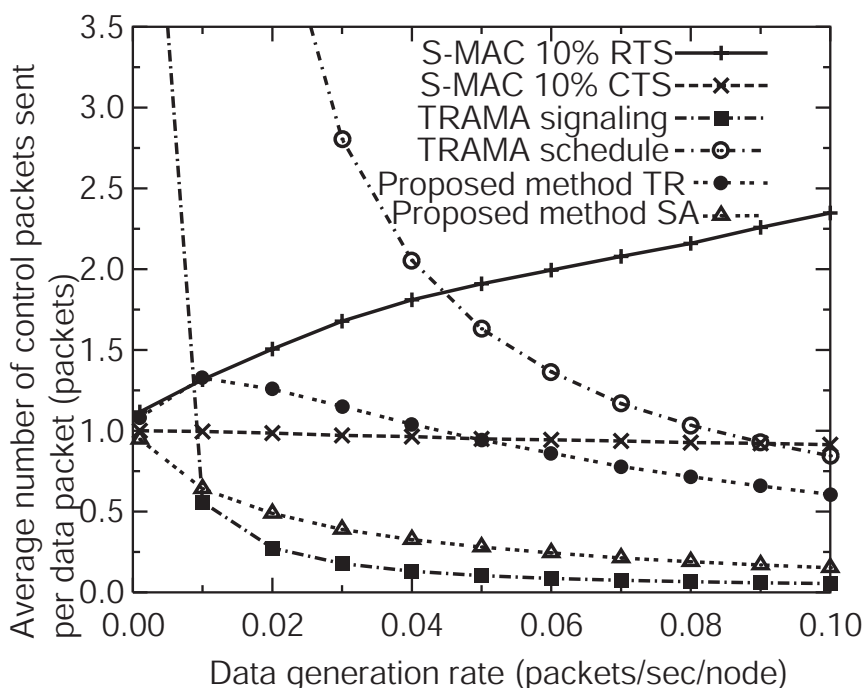


図 3.10 データパケット送信あたりの平均制御パケット送信数特性

効果が、TR パケットの損失による再送による影響を上回るため、1 データパケット送信あたりの TR パケットの送信数が低下していることが分かる。また、SA パケットもデータパケット発生率が増加すると、一度の SA パケットの送信によって複数のデータパケットを送信できるため、1 データパケット送信あたりの SA パケットの送信数も低下していることが分かる。

また、同期に関する制御パケットを除くすべての制御パケットの送信数を総制御パケット送信数と定義する。図 3.11 に単位時間 1 ノードあたりの平均総制御パケット送信数特性を示す。

TRAMA の場合、3.2.2 項で述べたように、データパケット発生率に関わらず、シグナリングパケットやスケジュールパケットが交換される。そのため、図 3.10 と図 3.11 より、データ発生率が低いときでも、1 データパケット送信あたりの制御パケット送信数や総制御パケット送信数が比較的高いことが分かる。一方、提案手法ではデータパケット発生に応じて、制御パケットが交換されるため、図 3.11 より、総制御パケット送信数が抑制されることが分かる。さらにデータパケット発生率が増加しても、3.3.6 項で述べた制御パケットの集約の効果により、総制御パケット送信数の増加率は S-MAC の duty cycle が 10 % の場合よりも抑制されることが分かる。ただし、ネットワークの環境や制御パラメータの設

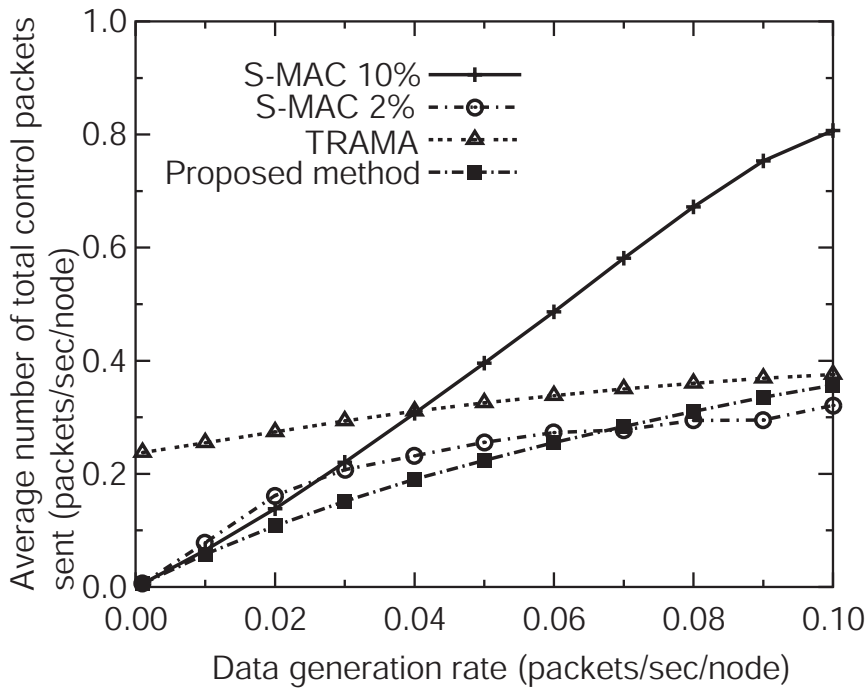


図 3.11 平均総制御パケット送信数特性

定によっては、データパケット送受信数が増加すると、提案手法の場合の方が TRAMA の場合より、総制御パケット数が大きくなる可能性がある。また、S-MAC の duty cycle が 2% の場合、データパケット発生率が 0.07 パケット毎秒以上の領域で、提案手法より総制御パケットの送信数は少なくなっていることが分かる。これはデータパケット送信数が提案手法より低く、それに伴って総制御パケットの送信数が少なくなっているためである。

#### 3.4.4 データ伝送量に関する特性

単位時間および MAC 層における 1 組の送受信ノード間あたりのデータの伝送量を平均スループットとする。なお、データの伝送量は、自分が発生した分と他のノードで発生し、中継する分の両者を含む。図 3.12 に MAC 層における平均スループット特性を示す。また、シミュレーションの時間内において、総データパケット発生数に対するシンクノードにおけるデータパケットの受信数の割合を配送率と定義する。図 3.13 に配送率特性を示す。

S-MAC の duty cycle が 2% の場合、パケットを待ち受けるアクティブ期間が減少するため、限られた期間にアクセスが集中することによりパケット衝突が増加し、パケット衝突を防止するために受信ノードがスリープ状態となる。そのため、送信ノードからの



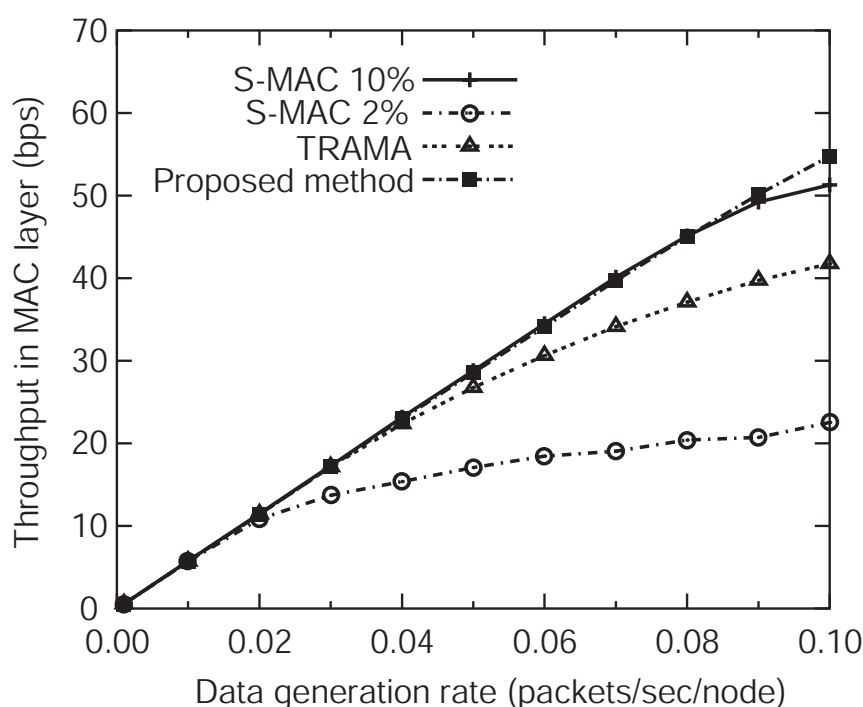


図 3.12 MAC 層における平均スループット特性

RTS パケットの受信に失敗することがある。その結果、データパケット発生率が増加すると、図 3.12 で示すように、MAC 層において十分なスループットが得られない。また図 3.13 より、配送率がデータパケット発生率の上昇とともに減少することが分かる。また、TRAMA の場合は、自分が優先権を持つ各スロットにおける各ノードの優先度はハッシュ関数によって算出されるため、各ノードに割り当てられるスロット数はほぼ一様となる。送信可能なスロットは、自分が初めから優先権を持つスロットと、隣接ノードから開放され、さらに自分が優先権を持つスロットに限られる。そのため、ノード密度が高く、転送するデータパケット数が比較的多いシンクノード付近のノードは、データパケットを送信するために必要なスロットを十分に確保できないことがある。さらに干渉によるパケットの損失による再送も増加する。そのため、データパケット発生率が増加すると、MAC 層における平均スループットの増加率が低下し、配送率も低下することが分かる。それに対して、提案手法では、表 1 のパラメータ設定において、duty cycle は 2.2 % であり比較的低い。しかしながら、アクティブ期間は制御パケットの送受信に応じて延長される。さらに、空いていると判断したスロットをどのノードも自由に利用できるため、ノード間のデータパケットの送信量の差があっても、それに応じてスロットを獲得する。そのため、提案手法はデータパケット発生率が測定領域において S-MAC の duty cycle が 10 % の場

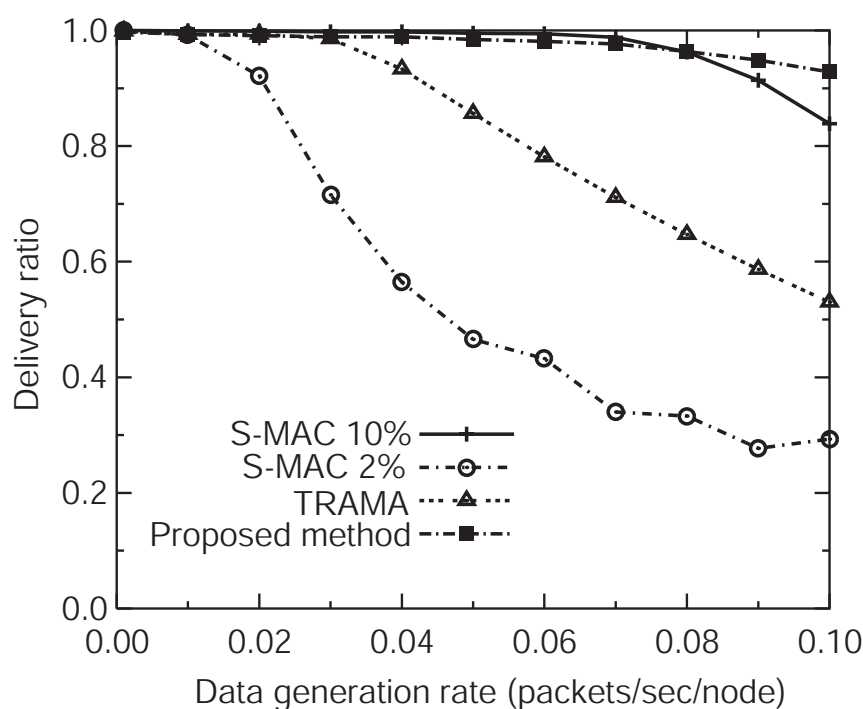


図 3.13 配送率特性

合と同程度の MAC 層におけるスループットや，配送率を保っていることが分かる．

### 3.4.5 平均消費電力特性

単位時間当たりの，送信時消費電力，受信時消費電力，アイドル時消費電力，またスリープ時消費電力をそれぞれ  $e_{tx}$ ， $e_{rx}$ ， $e_{idle}$ ， $e_{sleep}$  とする．また，全ノードの総送信時間，総受信時間，総アイドル時間，総スリープ時間をそれぞれ， $T_{tx}$ ， $T_{rx}$ ， $T_{idle}$ ， $T_{sleep}$  とする．全ノードの総消費電力を  $E_{total}$  とすると，

$$E_{total} = e_{tx}T_{tx} + e_{rx}T_{rx} + e_{idle}T_{idle} + e_{sleep}T_{sleep} \quad (3.1)$$

となる．総消費電力から単位時間 1 ノードあたりの平均消費電力を求めた．図 3.14 に平均消費電力特性を示す．

TRAMA の場合，干渉によるスケジュールパケットの損失が生じると，ノードがデータパケット送受信に関する情報を正確に把握できず，隣接ノードが送信しないスロットにおいてもアクティブ状態となる．さらに，データパケットの送受信の有無に関わらずス

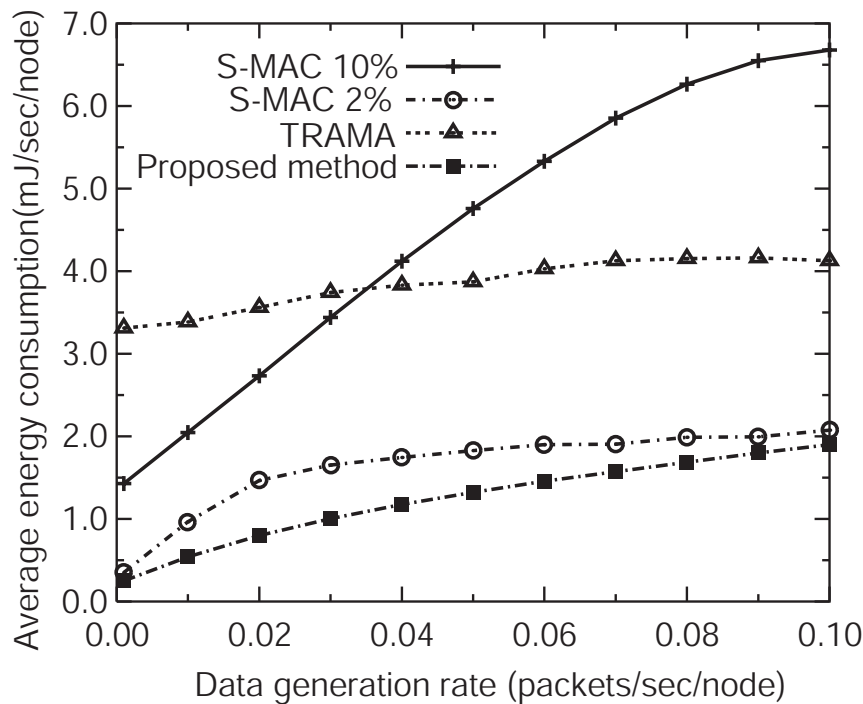


図 3.14 平均消費電力特性

ロット予約に関する制御パケットを定期的を送受信する．そのため，図 3.14 よりデータパケット発生率が低いときにおいても消費電力が比較的大きいことが分かる．一方，提案手法では送信するデータパケットが発生した場合のみ行うため，3.3.6 項の制御パケットにおける情報集約の効果で説明したように，スロット予約に関する制御パケットの送受信のオーバーヘッドが削減される．さらに適応的制御により，スロット予約期間におけるアクティブ期間が短縮される．そのため，提案手法の場合，データパケット発生率が比較的低いときにおいても，S-MAC や TRAMA と比較して消費電力は抑制されることが分かる．また，データパケット発生率が上昇しても，予め割り当てられたタイムスロットに送受信を行うため，キャリアセンスやバックオフ制御によって冗長なアイドル期間が比較的多い S-MAC の場合と比較して，消費電力の増加が抑制されていることも分かる．

### 3.4.6 消費電力効率特性

消費電力効率を以下のように定義する．

消費電力効率

$$= \frac{\text{シンクノードにおけるデータパケット受信数}}{\text{全ノードの総消費電力}}$$

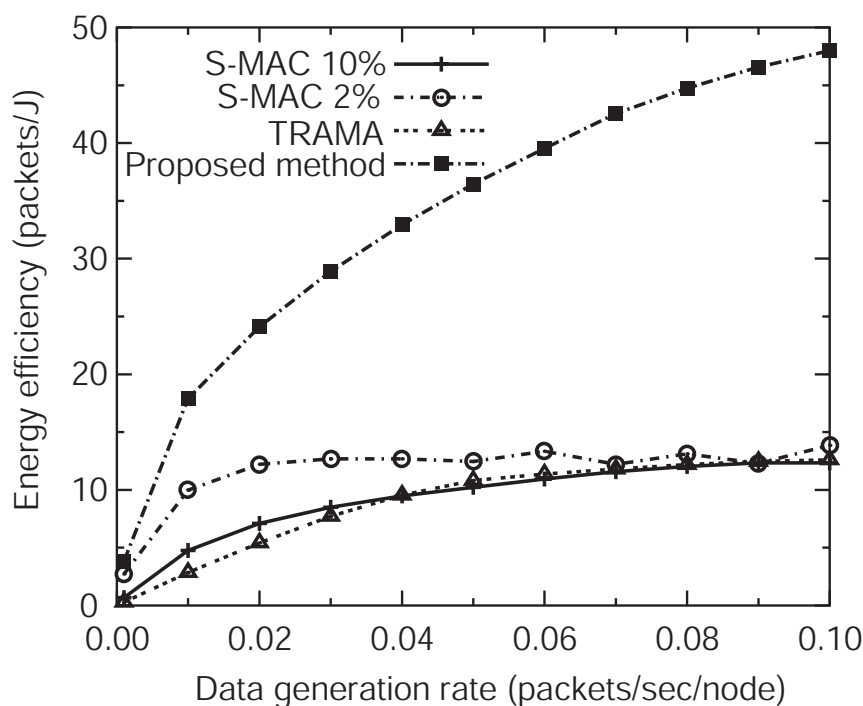


図 3.15 消費電力効率特性

(3.2)

図 3.15 に消費電力効率特性を示す。

図 3.15 より、提案手法では S-MAC や TRAMA と比較して、消費電力効率が高くなる事が分かる。これは、3.4.4 で述べたようにシンクノードにおけるデータパケット受信数を維持しつつ、3.4.5 で述べたように消費電力が抑制されるためである。また、提案手法ではデータパケット発生率の上昇に伴う消費電力効率の増加率が、S-MAC や TRAMA の場合より高くなっていることが分かる。これはシンクノードにおけるデータパケット受信数の増加に対して、消費電力の増加を抑制しているためである。

### 3.4.7 エンドツーエンド伝送における遅延特性

各ノードにおいて、データが発生した時点からあて先ノードであるシンクノードに到達する時点の間隔をエンドツーエンド伝送における遅延とし、その特性を測定した。図 3.16 にエンドツーエンド伝送における遅延特性を示す。

図 3.16 より S-MAC の duty cycle が 2 % の場合はトラフィックが増加するとスループットが十分に確保できず、遅延が大きくなる事が分かる。TRAMA や提案手法の場合は 1

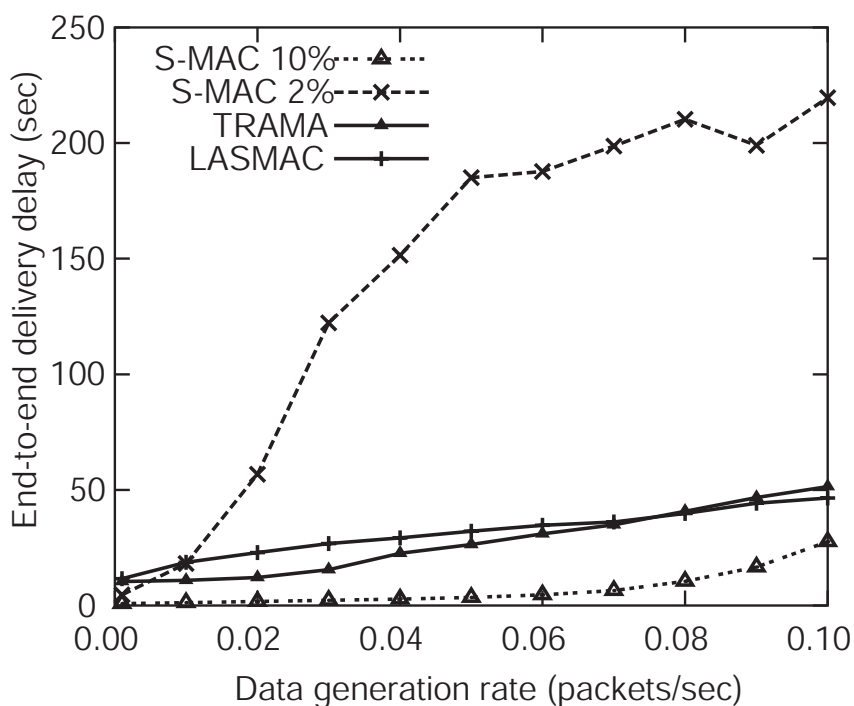


図 3.16 エンドツーエンド伝送における遅延特性

フレーム中に転送できるホップ数は1と限られているのに対し，S-MACのduty cycleが10%の場合はAdaptive Listenにより，1周期に転送できるホップ数は複数可能であるため，S-MACの方が遅延が小さいことが分かる．

### 3.4.8 フレーム長の差異の影響

図 3.17 にフレーム長を 25 スロット (2.352 秒)，50 スロット (3.652 秒)，100 スロット (6.252 秒)，150 スロット (8.852 秒)，200 スロット (11.452 秒) としたときのエンドツーエンド伝送における遅延特性を示す．また，図 3.18 にフレーム長を変化させたときの配送率特性を示す．一般にフレーム長が短いほど，スロット予約を行う頻度が増えるため，エンドツーエンド伝送における遅延が小さくなることが分かる．しかしながら，フレーム長が短いほど全期間に対して，スロットでデータパケットを送信する期間が相対的に短くなり，スロット予約期間が相対的に長くなるため，図 3.18 のフレーム長が 25 スロットのときのように，配送率が低下し，伝送遅延も大きくなることが分かる．

図 3.19 にフレーム長を変化させたときの，平均消費電力特性を示す．一般にフレーム長が短いほどスロット予約のアクティブ期間の頻度が高くなるため，平均消費電力が増加することが分かる．

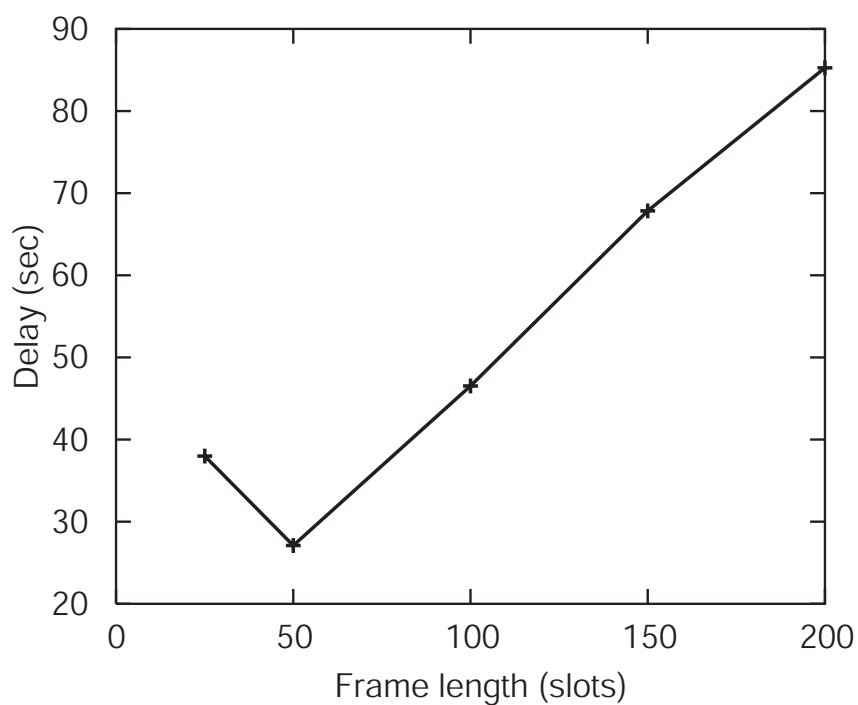


図 3.17 フレーム長とエンドツーエンド伝送における遅延特性

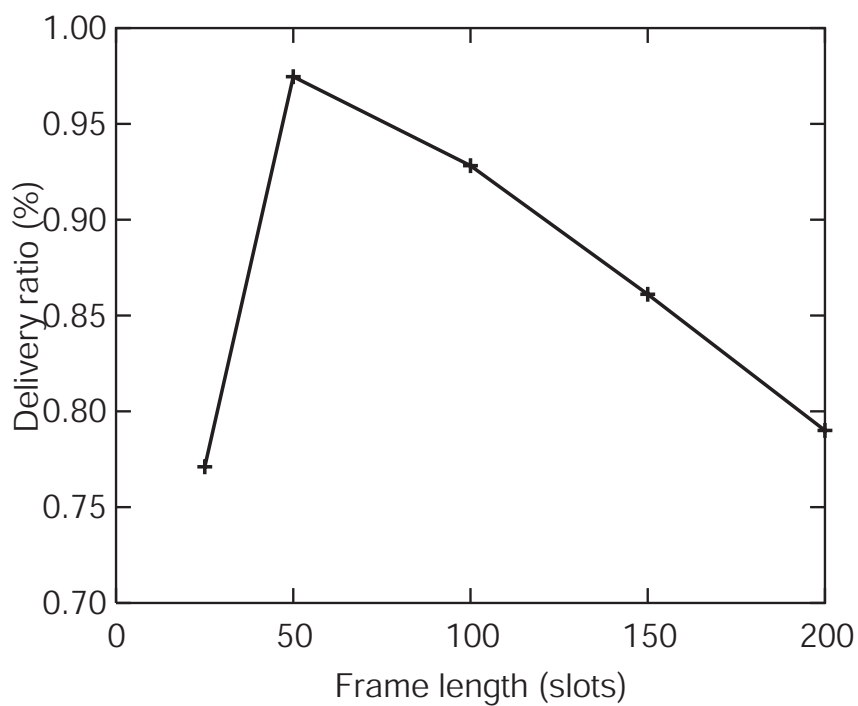


図 3.18 フレーム長と配送率

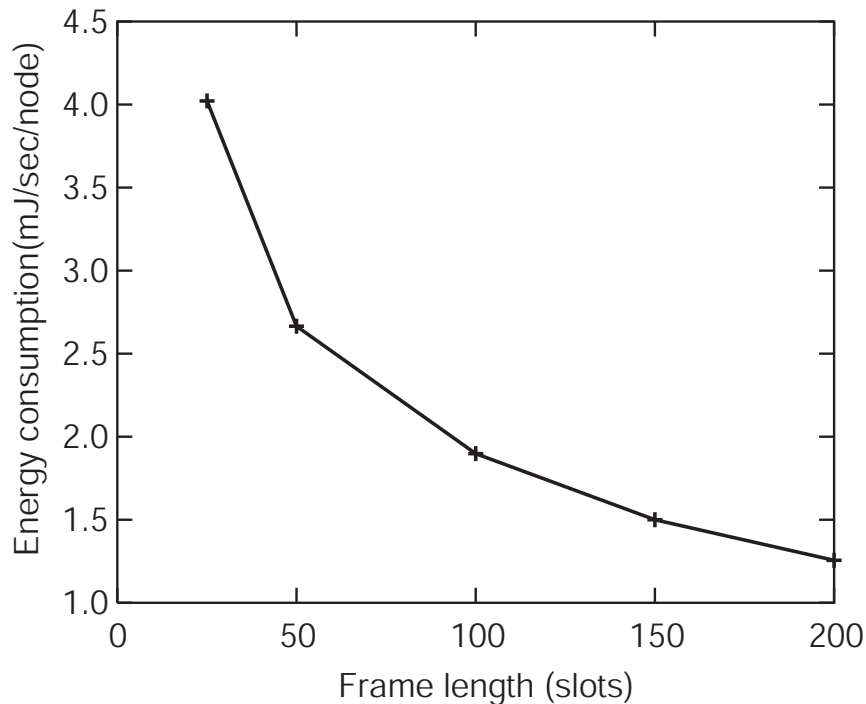


図 3.19 フレーム長と平均消費電力特性

### 3.4.9 部分的なノードの送信の影響

図 3.20 に送信エリアの割合とスループット特性，図 3.21 に送信エリアの割合と配送率特性を示す．図 3.20 や図 3.21 より，S-MAC の 10 % の場合のスループットや配送率が高くなっていることが分かる．これは S-MAC の 10 % の場合他の場合と比較して duty cycle が高く，通常時のアクティブ期間が長い上にランダムアクセスによりどのノードも自由にチャンネルにアクセスできる機会が多いためである．一方，TRAMA の場合はスロット予約が基本的にハッシュ関数を利用した優先権の計算に基づいているため，ノード間でトラフィックの偏りがある場合はチャンネルを有効に利用することが困難である．提案手法の場合は TDMA 方式でありながらランダムアクセスによってタイムスロットを獲得できるため，TRAMA よりはスループットが高いことが分かる．

図 3.22 に送信エリアの割合と平均消費電力特性を示す．3.4.5 項で述べたように，提案手法は予め割り当てられたタイムスロットでデータを送受信し，ランダムアクセス期間のスロット予約期間におけるアクティブ期間も適応的制御により抑制されるため，消費電力を抑制することができることがわかる．

消費電力効率をデータパケット受信成功数 ÷ 総消費電力とし，図 3.23 に送信エリアの割合と消費電力効率特性を示す．提案手法はスループットの低下を抑制しつつ消費電力を

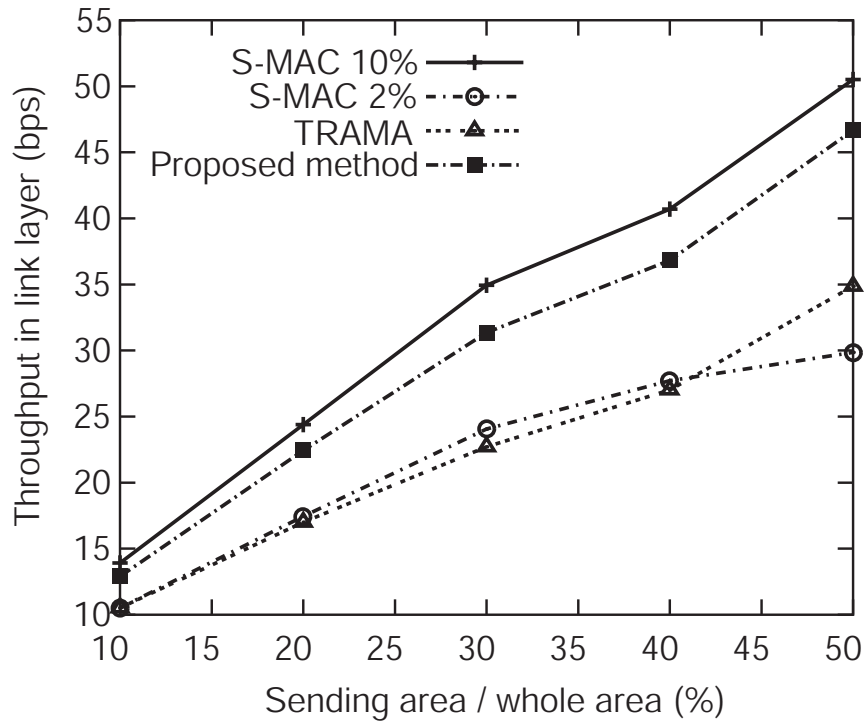


図 3.20 送信エリアの割合とスループット特性

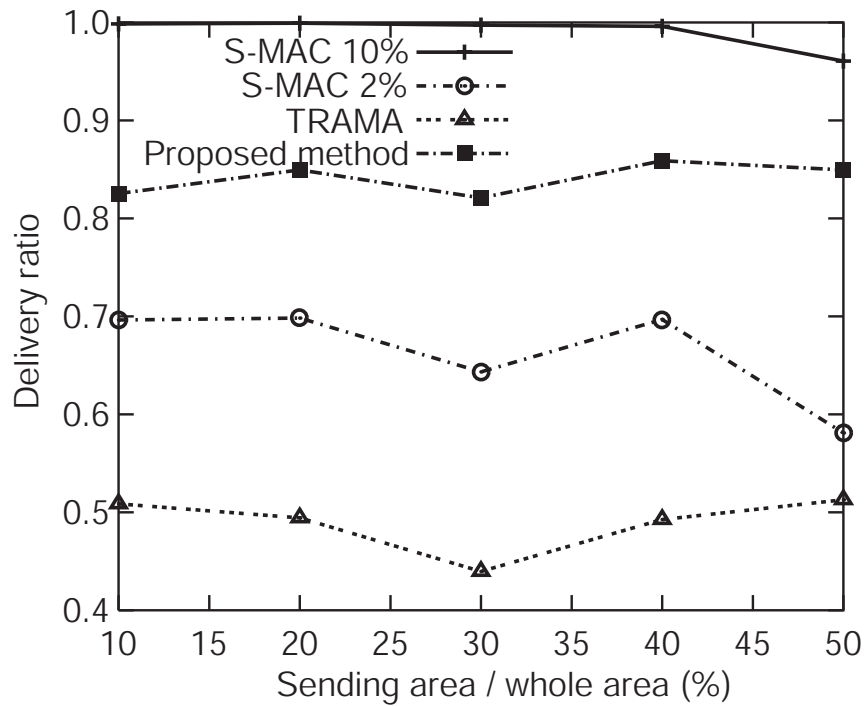


図 3.21 送信エリアの割合と配送率特性



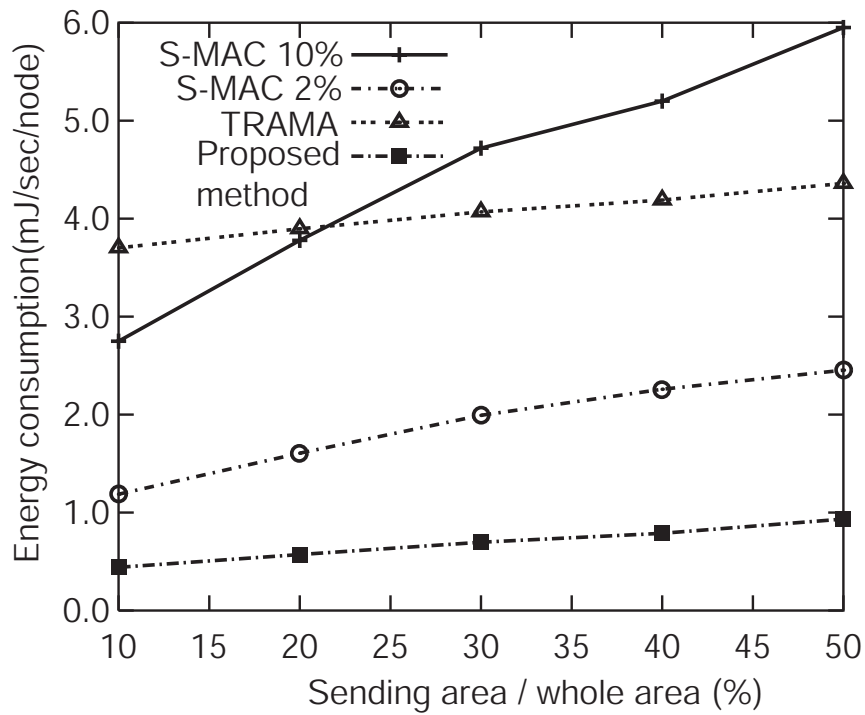


図 3.22 送信エリアの割合と平均消費電力特性

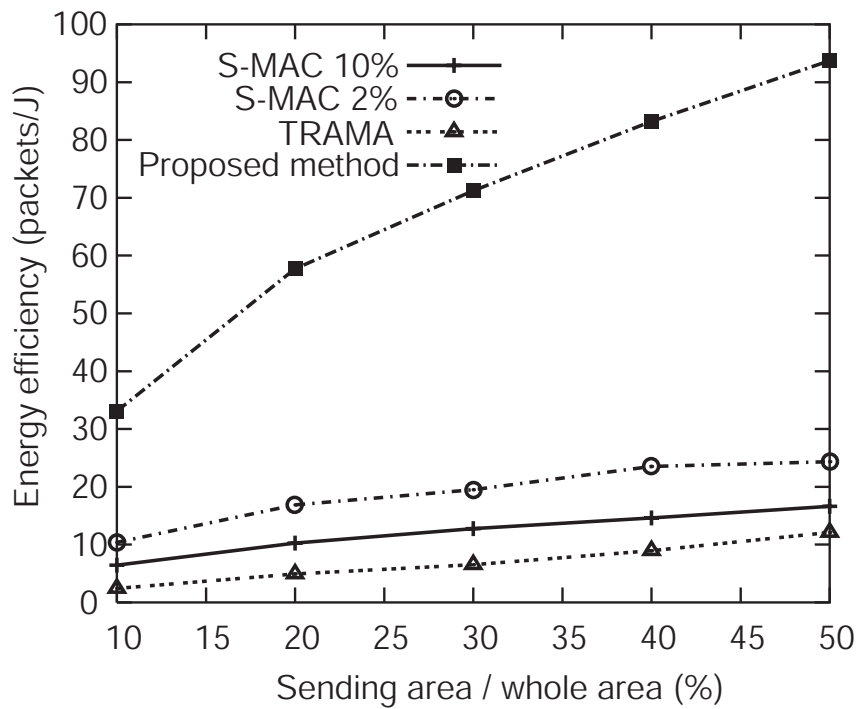


図 3.23 送信エリアの割合と消費電力効率特性

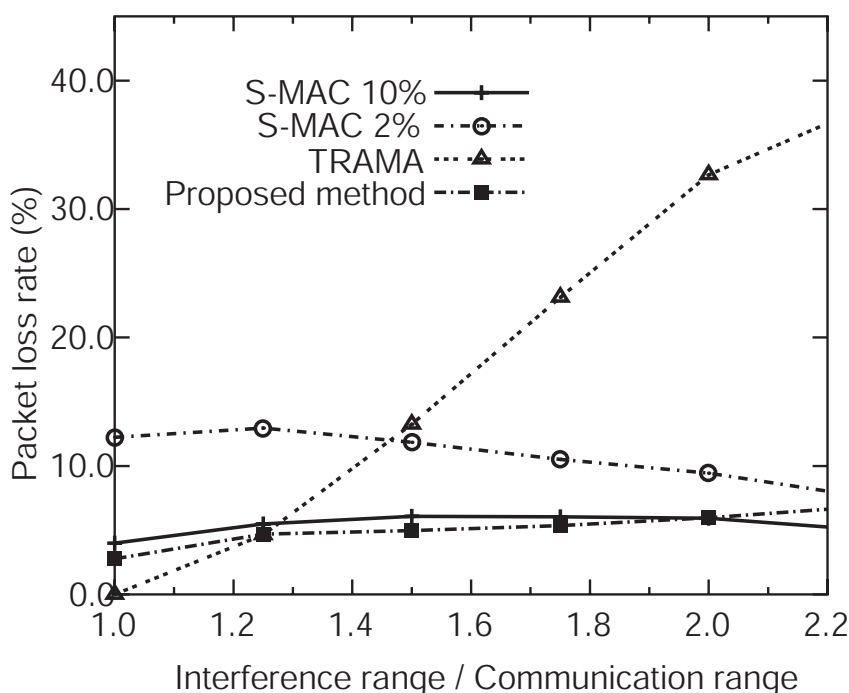


図 3.24 干渉距離とパケットロス率特性

抑制するため、消費電力効率が比較的高いことが分かる。

### 3.4.10 干渉距離の影響

干渉距離を変化させたときのパケットロス率特性、配送率特性、平均消費電力特性、また消費電力特性をそれぞれ図 3.24, 図 3.25, 図 3.27, また図 3.27 に示す。なお、パケットロス率は  $1 - (\text{MAC 層でのデータパケット総受信数}) \div (\text{MAC 層でのデータパケット総送信数})$  として求めた。図 3.24 より、TDMA 型の TRAMA では通信距離に対する干渉距離が 1.0 のときは干渉の影響がなく、2 ホップ内でのスロット割り当ての整合性が取れるため、全くパケットロスが生じないことが分かる。しかしながら、通信距離に対する干渉距離が増加すると、干渉の影響が大きくなり、制御パケットの送信量が多い TRAMA の場合はデータパケットの送信と干渉するため、パケットロス率の増加率が高いことが分かる。また S-MAC の duty cycle が 2 % の場合は干渉距離の増加するとパケットロスによって送信を抑制し、スリープ状態になる機会が増加するため、パケットロスはやや低下していることが分かる。提案手法の場合は競合によってスロットを獲得するため、TRAMA の場合と異なり、通信距離に対する干渉距離が 1.0 のときでもパケットロスは生じる。しかしながら TRAMA と比較して制御パケットの送信が少なく、空きスロットを確率的に選択す

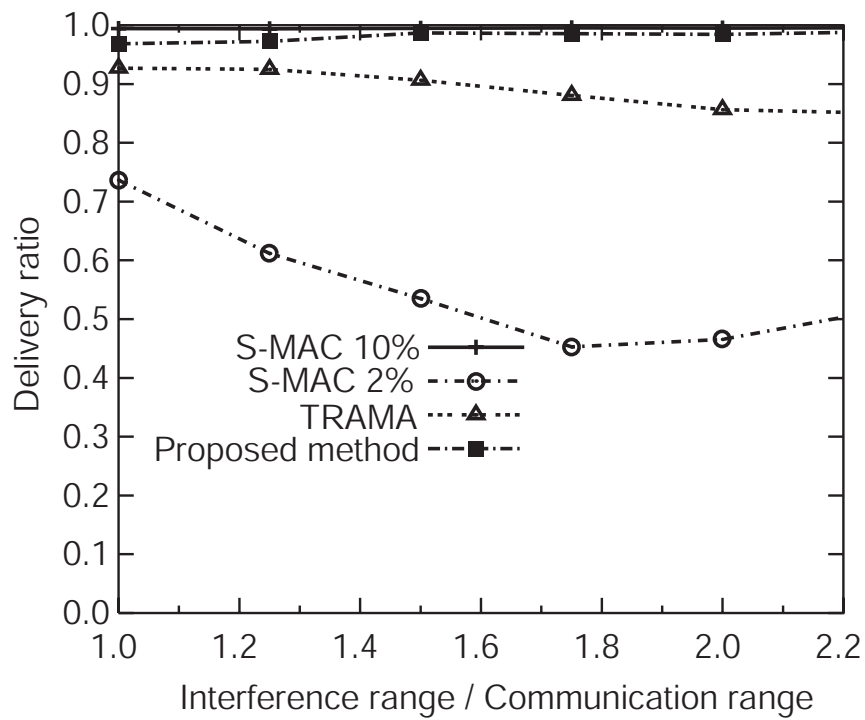


図 3.25 干渉距離と配送率特性

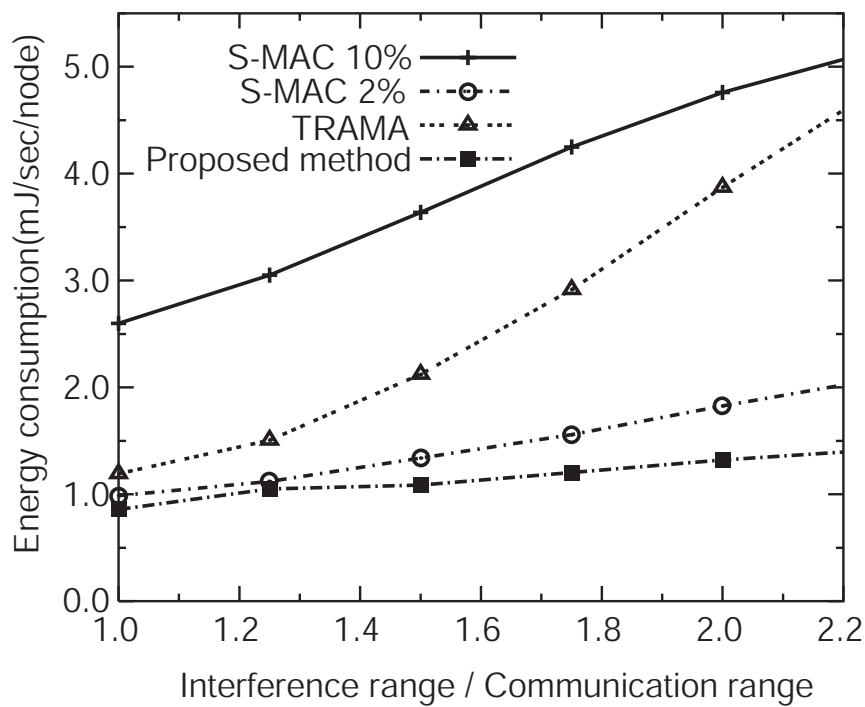


図 3.26 干渉距離と平均消費電力特性

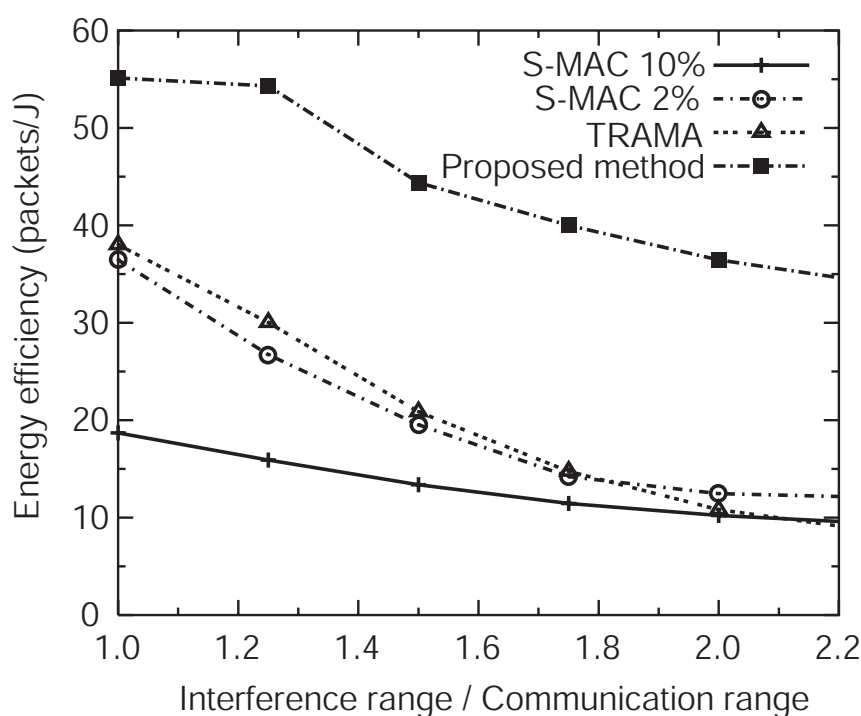


図 3.27 干渉距離と消費電力効率特性

ること干渉の影響を抑制するためパケットロスの増加を抑制することができる。図 3.25 より、S-MAC の duty cycle が 2 % の場合は干渉距離に関わらず十分なデータパケット送受信が得られないために、比較的配送率が低いことが分かる。また TRAMA の場合は干渉距離に関わらず、ネットワーク内でトラフィックが部分的に集中すると、送信量が多いノードがスロット割り当てにおいて十分なスロットを得られず、配送率が S-MAC の duty cycle が 10 % の場合や、提案手法の場合よりも配送率が低いことが分かる。

図 3.26 より、一般に、干渉距離が増加すると、パケットロスによる増加するため、消費電力が増大することが分かる。TRAMA ではパケットロス率の増加によってデータパケットの再送も増加するため、消費電力の増加率が比較的高いことが分かる。また、S-MAC の duty cycle が 10 % の場合はランダムアクセスによる受信ノードの数も増加するため、消費電力が増加しやすいことが分かる。一方、提案手法の場合は干渉距離が増加してもパケットロス率の増加を抑制でき、スロット予約によって余計なパケットを受信するノードの数を抑制できるため、消費電力の増加を抑制していることが分かる。

図 3.26 より、一般に、干渉距離が増加すると、消費電力が増大するため、消費電力効率が低下することが分かる。

## 3.4.11 実装による動作確認

本提案手法による MAC プロトコルを TinyOS ver.1.0[42] 上で実装し、動作を確認した。プログラムの基本構造はタイマを設定し、タイマで設定された値をデクリメントしていった、タイムアウトになったらイベント実行するというものである。図 3.28 に実装におけるスロット予約のフローチャートを示す。また、図 3.29 に実装におけるスロット予約期間の適応的制御のフローチャートを示す。TR 期間の初めにおいてチャンネルを観測し、アクティブ期間の長さを制御する channel timer と最大の TR 期間の長さを制御する period timer を設定する。 $TRP_{min}$  の間チャンネルがフリーで channel timer がタイムアウトとなるとスリープ状態となる。一方、 $TRP_{min}$  の間、自分や他ノードが送受信を行ってチャンネルがビジーとなればその送受信が終わってから再び channel timer を設定する。 $TRP_{min}$  の間チャンネルがフリーであるまで再帰的にこの動作が行われる。一方、period timer は  $TRP_{max}$  だけ時間が経過すると SA 期間にノードの状態を以降させる。SA 期間についても同様に、SA 期間の開始時に channel timer と period timer を設定し、 $SAP_{min}$ 、 $SAP_{max}$  の値を用いてアクティブ期間の長さや、最大の SA 期間の長さを制御する。

図 3.30 に実装における通信スタックを示す。実装は TinyOS ver. 1.0 の S-MAC の通信スタック (ISI stack)[43] をベースに行った。ISI stack はデフォルトの Berkeley stack と比較して、MAC 層と物理層におけるプログラム関数のレイヤリングがより明確に行われているため、MAC 層のプログラムの変更が比較的容易であることが主なメリットである。

図 3.31 に実装におけるプログラム関数の構成を示す。各関数を大別すると、初期化関数、隣接ノード管理関数、タイマ制御関数、状態制御関数、アクティブ・スリープスケジュール同期制御関数、送信・受信関数、またスロット予約期間における適応的制御関数となる。

動作確認は図 3.32 で示すように、1 ホップでお互い通信できる範囲内にある 3 個のノードを設置し、1 個の受信ノードに対して 2 個の送信ノードがある一定間隔でデータパケットをユニキャストする動作、また、隣接ノードに対してブロードキャストする動作を行った。各ノードが、制御・データの各パケットの送受信数をカウントし、その情報を PC に収集することにより、図 3.33 で示すように、各状態の総時間に対する送受信状態、アイドル状態、スリープ状態の各割合を観測した。一般に送受信の割合は小さく、アイドル状態やスリープ状態が大部分を占めることが分かる。

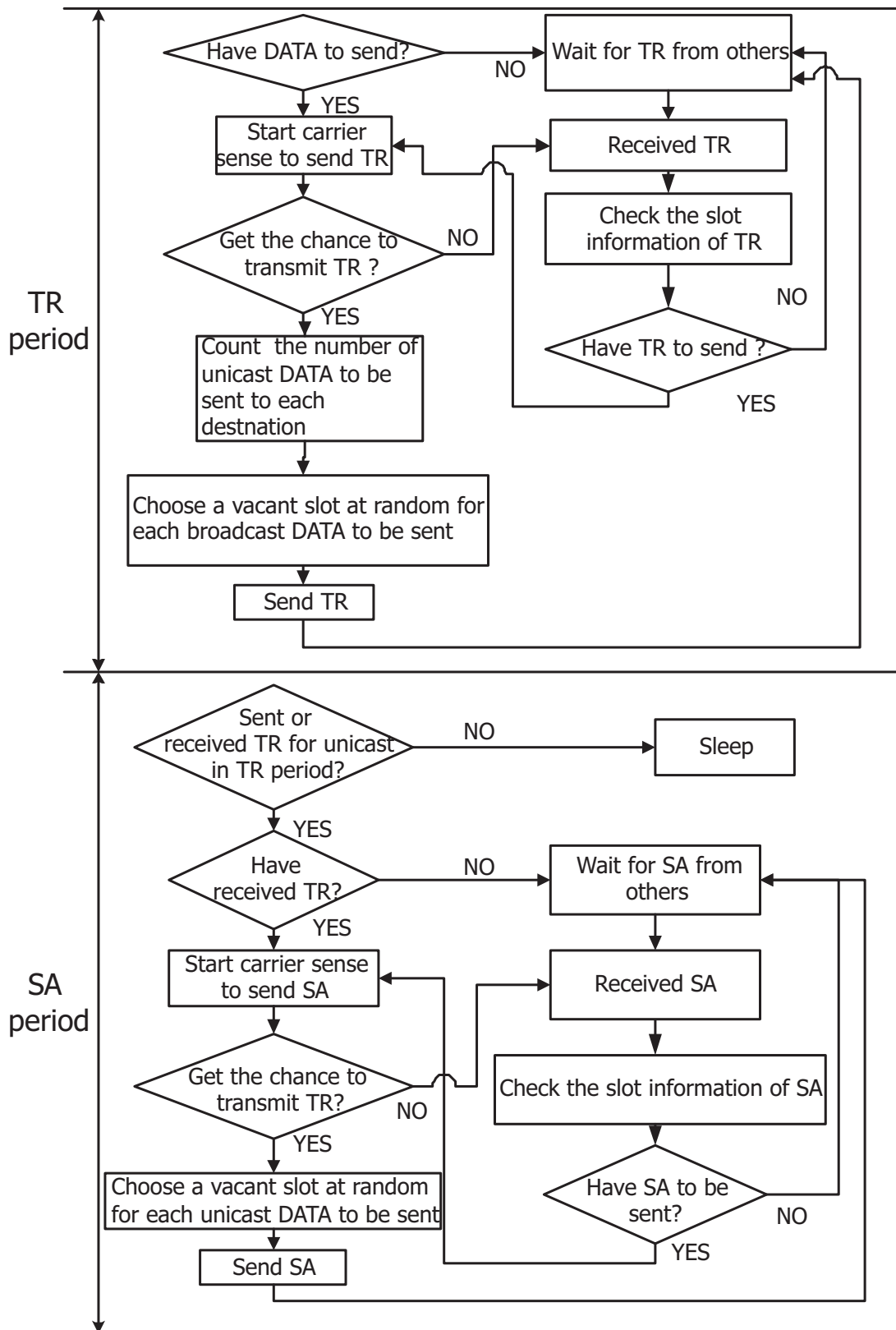


図 3.28 スロット予約のフローチャート

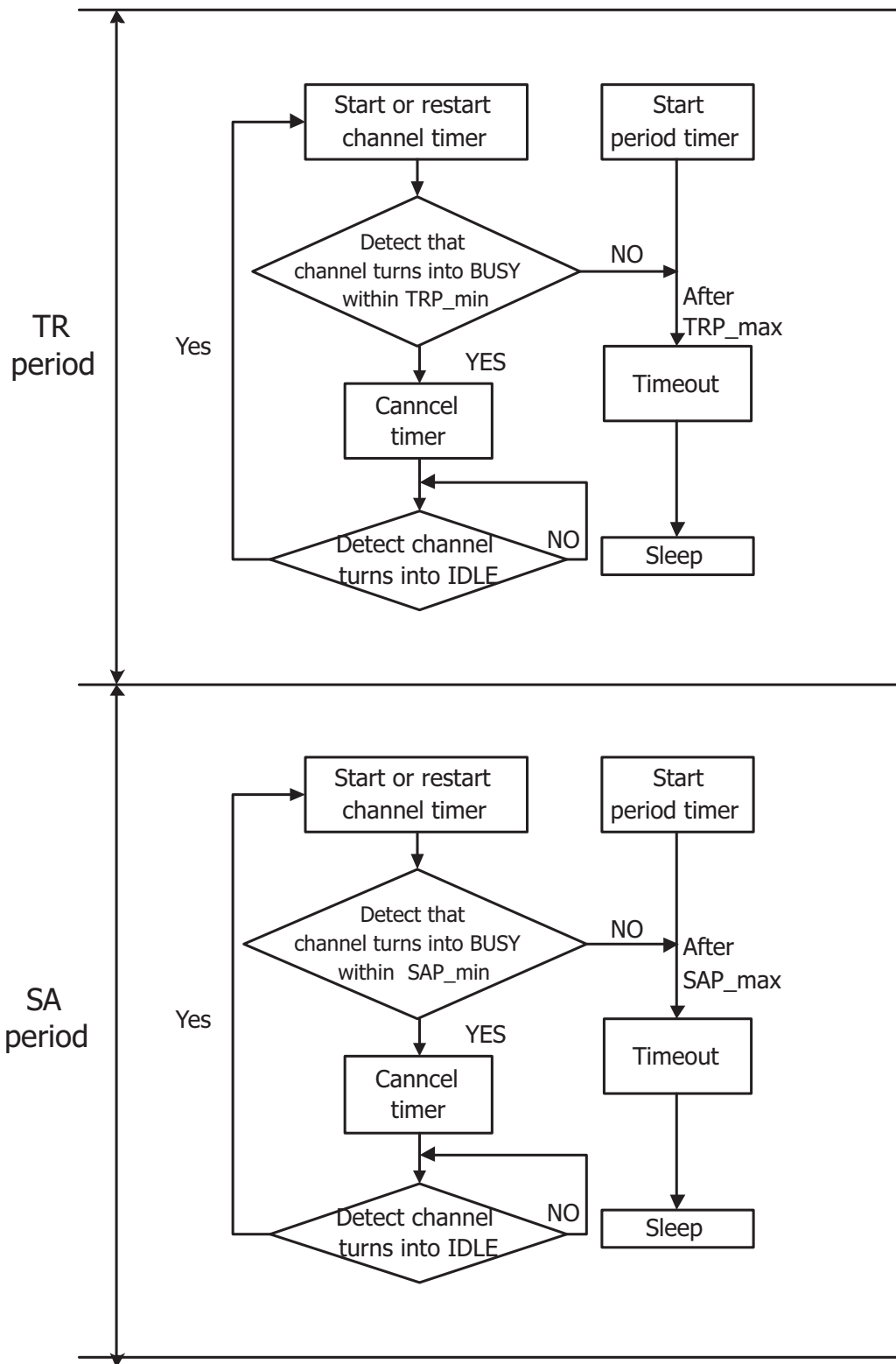


図 3.29 スロット予約期間の適応的制御のフローチャート

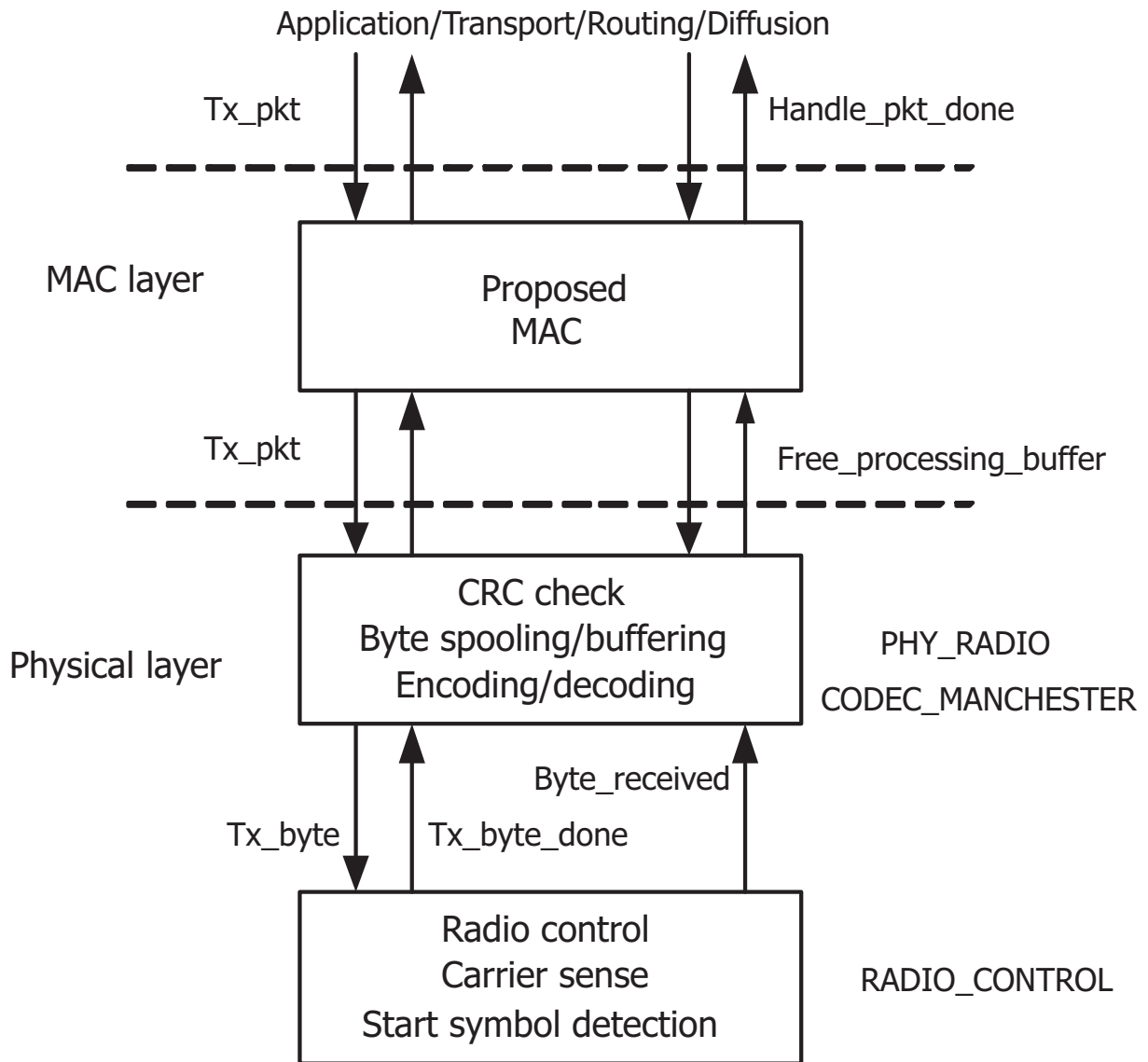


図 3.30 実装における通信スタック



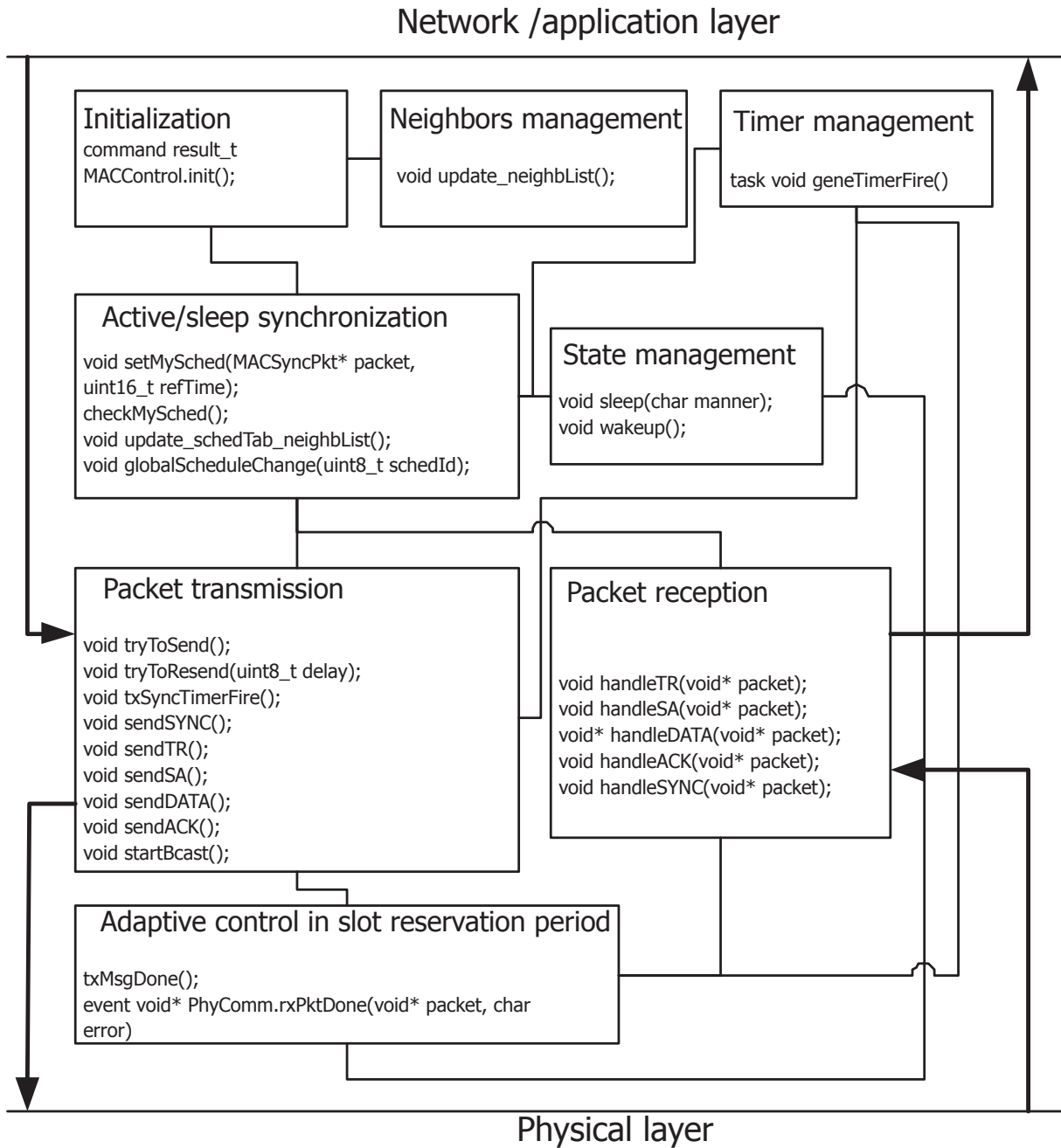


図 3.31 提案手法のプログラム関数

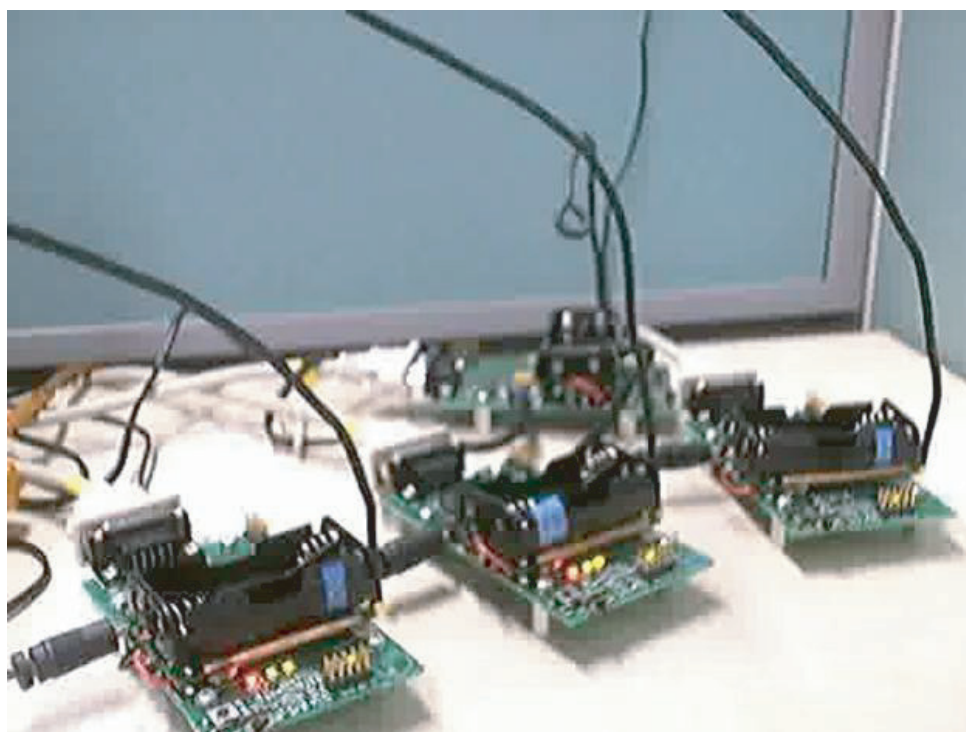


図 3.32 提案手法の Mote を用いた動作確認

```
At 134:01:56.92
 52 51 34 00 33 00 D2 09 16 28 00 00 D2 03 00 00
 00 00 00 00 B8 1C 00 00 72 F2 6B 01 0C 0B 00 00
 3D 3C 0E 00 01 00 2D 00 0E 00 0B 00 7F 6B EE 19
 1B 13 6C 01 CC 56 0C 00 B1 97 52 00 1B 6D B9 1B
 C9 C3 54 5F 00 00 00 00 01 00 00 00 90 AD F7 35
 F6 D0
pktLen= 82
TRpacket from 51 to 52 134:01:56.92
51 pkttype= 81
51 slottime= 2514
51 numsendTR= 10262
51 numsendSA= 978
51 numrecvSA= 7352
51 numrecvACK= 62066
51 numrecvDATA= 363
51 sendslot[0]= 14
51 sendslot[1]= 1
51 sendslot[2]= 45
51 sendslot[3]= 14
51 sendslot[4]= 11
51 txTime= 5412785 1.163697 134:01:56.92

51 rxTime= 808652 0.173852 134:01:56.92
51 idleTime= 23859995 5.129671 134:01:56.92
51 sleepTime= 435055487 93.532778 134:01:56.92
51 clockTime= 465136923 100.000000 134:01:56.92
51 numcycle= 50121
```

図 3.33 制御パケット送受信数と無線インタフェースの状態の観測

### 3.5 本章のまとめ

本章では、センサネットワークにおいて、データ通信要求発生時にのみ隣接ノードの制御パケットの交換によってスロット予約を行い、スロット予約期間におけるアクティブ期間を動的に制御する低消費電力MACプロトコルを提案した。計算機シミュレーションによる評価を行い、従来方式と比較して、duty cycleが低いことによって生じるスループットの低下を抑制でき、消費電力をより削減することができるため、消費電力の効率を改善できることを示した。

提案手法の利点としては以下の点が挙げられる。

1. 低消費電力であること提案手法ではトラフィックが発生していないときはスロット予約期間のアクティブ期間を適応的制御により抑制できるため、低いduty cycleで稼働できる。一方、トラフィックが高くなっても割り当てられたタイムスロットでデータパケットの送受信を行えるため冗長なアクティブ期間を抑制できる。
2. 消費電力のわりにスループットが高いこと提案手法ではスロット予約期間のアクティブ期間をトラフィックの増加に応じて適応的に延長するために、各ノードが十分な制御パケットを交換しやすい。またタイムスロットの獲得は各ノードのトラフィックに基づいて行われるため、特にノード間でトラフィックの差異がある場合に従来のTDMA方式と比較してチャンネルを有効に利用することができる。

一方、欠点としては以下の点が挙げられる。

1. 消費電力と遅延のトレードオフが存在すること提案手法では1フレームに転送できるのは1ホップのみである。そのため、Adaptive Listenにより1サイクルの間に複数のホップ転送できるS-MACと比較してホップバイホップの転送遅延が大きくなる。またフレーム長を短くして遅延を低下させると消費電力が増加し、フレーム長を長くして消費電力を低下させると遅延が増加するという消費電力と遅延のトレードオフの問題がある。
2. パケットロスを完全に抑制できていないこと提案手法ではスロットの割り当てが1ホップ間のノードの情報交換のみによって行われるため、制御パケットの損失によるデータパケットの衝突や、干渉の影響によりパケットロスが生じる。

提案手法は、従来手法と比較して低消費電力であり、トラフィックが増加してもスループットを維持する。そのため、トラフィックが時間や場所によって異なるアプリケーション

ンでの利用に適していると考えられる。例えばある不規則に発生するイベントを観測するとき、大部分の期間は通信を行わないため、低い duty cycle で稼働できる提案手法が有効である。またイベントを検知した一部のノードが送受信を行う場合、そのトラフィックに応じてスロット予約期間のアクティブ期間を延長することで、スループットが向上させることができる。

一方、提案手法はホップバイホップの転送遅延が大きいいため、一般に低遅延性が要求される緊急時の通報システムには不向きであると考えられる。

以下に今後の課題について述べる。提案手法は、各ノードがお互い同一のタイムスケジュールで同期して動作することを仮定している。提案手法では少なくともリンク層の送受信間ノードで同期が取れていれば送受信は可能となる。また、送受信ノードのそれぞれのノードの隣接ノードの範囲で同期が取れていれば、送受信ノードの制御パケットを受信することが可能となる。しかしながら、同期制御において、ノード間で場合によっては常時同期をとらない場合や、同期の誤差によりタイムスケジュールがずれることも予想される。そのため今後は、提案手法における同期制御におけるロバスト性の向上について検討していく必要があると考えられる。また、トラフィックに応じた1フレーム長の動的な変更や、スロット割り当ての優先制御についても検討していく必要があると考えられる。実際の利用環境を想定して、提案手法と、アプリケーションレイヤやネットワークレイヤにおける制御との連携も必要となると考えられる。

## 第4章 間欠通信適応型同期制御

---

## 4.1 本章の概要

センサネットワークにおける省電力化の非常に有効なアプローチとして間欠通信を行う手法がある。間欠通信とは、通信を行わない間はスリープ状態となることで消費電力を削減する手法である。しかし、通信を行うには送受信ノードがアクティブ状態であればならないため、間欠通信を行う際にはこのアクティブ・スリープ状態の同期制御が必要となる。この間欠通信における同期制御において、従来の手法では通信要求の有無に関わらず同期制御を行い、ネットワーク全体で同期の維持を行うため、同期制御のオーバーヘッドが大きいという課題がある。そこで本章では、間欠通信における同期制御のオーバーヘッドの削減による省電力化を目的とし、送信要求の発生に応じて同期制御を行うことで、一部のノードが一時的に同期制御を行うオンデマンド型同期制御手法の提案を行う。提案手法ではさらにノード間のクロックスキューを考慮してタイミングを同期させることで、高精度な同期を長期間維持し、同期制御のオーバーヘッドを抑制する。本章では、まず間欠通信の概要及び関連研究である S-MAC の同期制御手法について述べる。続いて、提案するオンデマンド型同期制御手法とシミュレーション結果及び考察を述べる。最後に今後の課題を述べる。

## 4.2 関連研究

センサネットワークにおいて、通信を行わないアイドル状態を抑制し、消費電力を削減する手法に間欠通信がある。間欠通信では、センサノードは通信を行わない間、通信機の電源を切ることでスリープ状態となり消費電力の削減を行うが、スリープ状態では通信を行うことができないため、通信を行う際には、他のノードとアクティブ状態となるタイミングを同期させる必要がある。間欠通信におけるアクティブ・スリープ状態の同期制御手法は、以下に示すように、大きく分けて完全同期型、部分的同期型、非同期型の3種類に分類される。

### 4.2.1 完全同期方式

完全同期方式はネットワークの全てのノードが同一の基準により同期する手法である。図 4.1, 図 4.2 にそれぞれ、完全同期方式の動作例と動作例におけるネットワークトポロジーを示す。完全同期方式の代表的なプロトコルに TRAMA[33] や Z-MAC[44] がある。これらのプロトコルでは、基本的にアクティブ期間の通信には細かく区切られたタイムスロットでデータの送受信を行う TDMA を利用するため、非常に高い同期精度を維持する

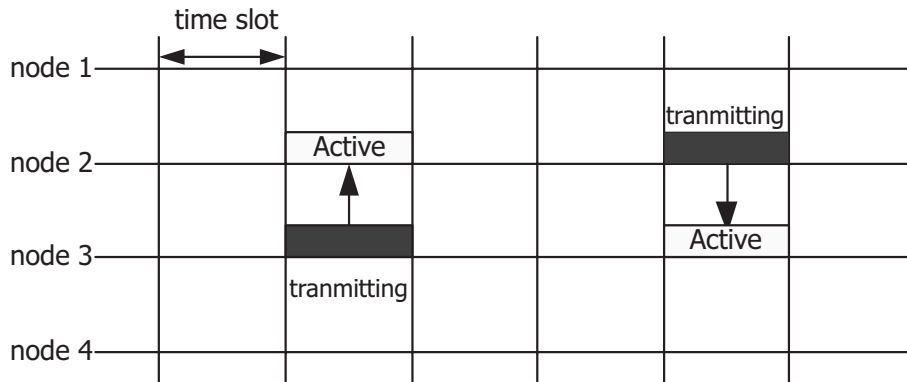


図 4.1 完全同期型の動作例

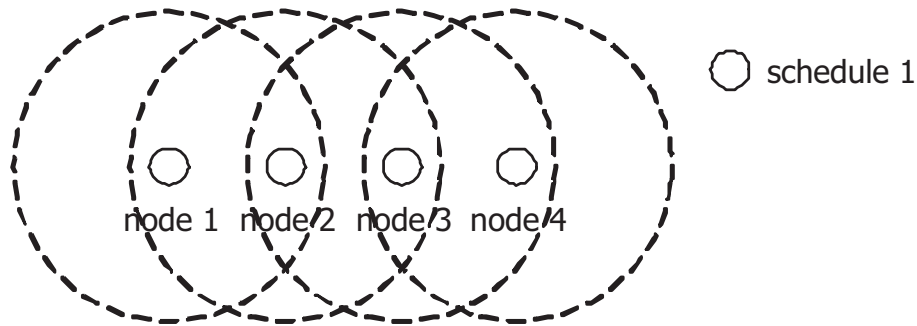


図 4.2 動作例におけるネットワークトポロジー

必要がある。

完全同期型では、冗長なアイドル期間を削減できるため消費電力の効率が良いというメリットがあるが、ネットワーク全体で1つの同期を維持する必要があるため同期制御のオーバーヘッドが大きいという問題もある。

#### 4.2.2 部分的同期方式

部分的同期方式は周期的に変移するアクティブ・スリープ期間を同期させることで通信を可能にする。図 4.3, 図 4.4 に部分的同期型の動作例と動作例におけるネットワークトポロジーとアクティブ・スリープのスケジュールを示す。部分的同期型ではネットワーク内に複数のスケジュールが混在するため、図 4.3 のノード 3 のように複数のスケジュールの境界に存在するノードは複数のスケジュールで動作する。代表的なプロトコルに S-MAC[28] がある。

部分的同期方式では、完全同期型に比べ同期制御にかかるオーバーヘッドを削減できると



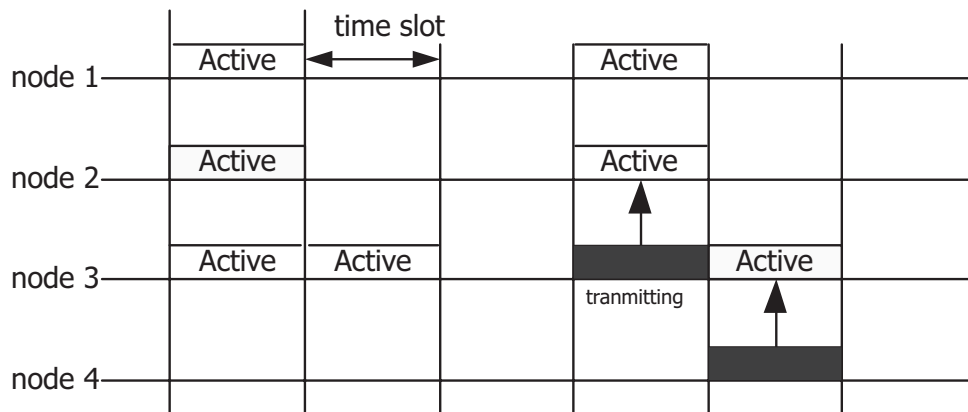


図 4.3 部分的同期型の動作例

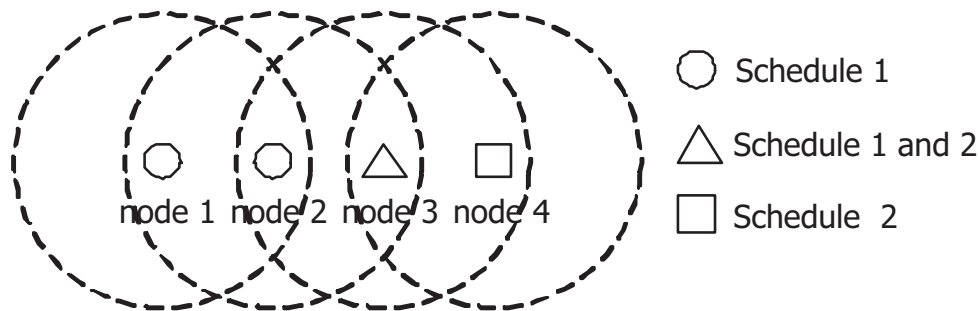


図 4.4 動作例におけるネットワークポロジとアクティブ・スリープのスケジュール

という利点があるが、複数のスケジュールの境界に存在するノードで電力を消費しやすいという欠点もある。

#### 4.2.3 非同期方式

非同期方式は通常時は同期制御を行わず、全てのノードがばらばらなスケジュールでアクティブ・スリープの状態を変移させる。図 4.5, 図 4.6 にそれぞれ、非同期方式の動作例と動作例におけるネットワークポロジを示す。非同期方式では基本的にこのアクティブ・スリープ状態の同期制御は行わず、通信を行う場合には、送信側が 1 周期の間受信ノードを起こすための信号を送信し続けることで周囲のノードをアクティブ状態にし、通信を行う。代表的なプロトコルに B-MAC[32] や WiseMAC[45] がある。

非同期型ではアクティブ・スリープ状態のスケジュール同期が不要であり、同期制御におけるオーバーヘッドがゼロであるというメリットがあるが、通信に無関係なノードもアクティブ状態となるため消費電力を浪費する可能性もある。

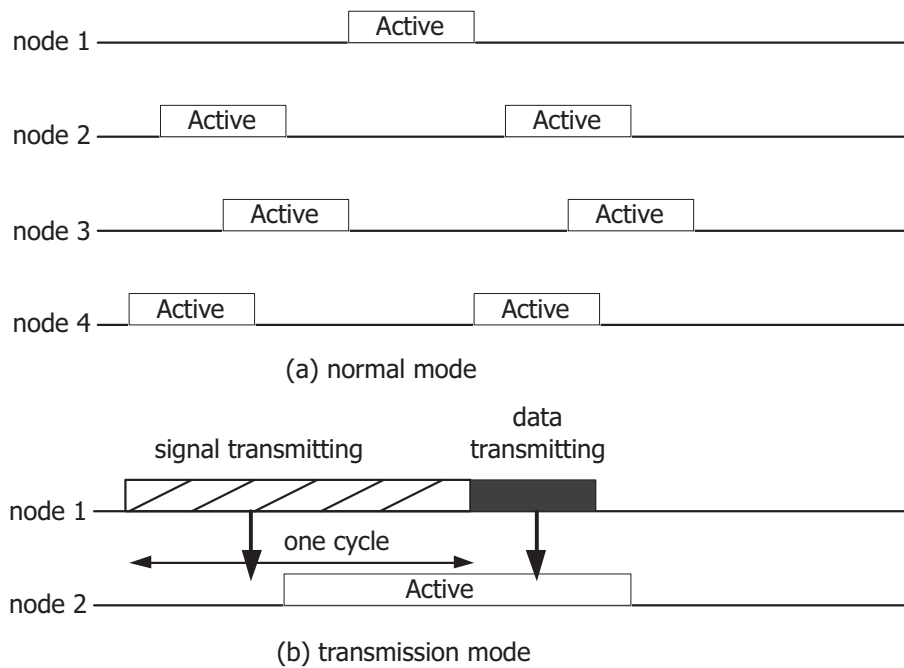


図 4.5 非同期型の動作例

#### 4.2.4 S-MAC

本章では、間欠通信の中でも CSMA における同期制御のオーバーヘッド削減を目的としている。そのため、ここでは CSMA での通信を行う部分的同期型で代表的な S-MAC の同期制御手法を述べる。

間欠型 MAC プロトコルの例として S-MAC がある。このプロトコルでは、ネットワー

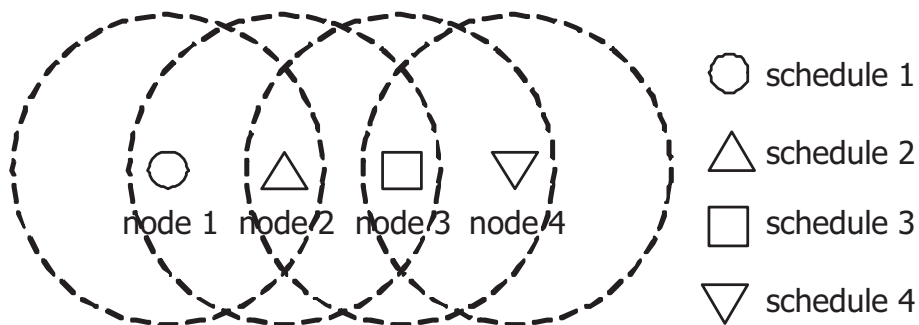


図 4.6 非同期型の動作例におけるネットワークトポロジーとスケジュール

クの各ノードは周期的にアクティブ状態とスリープ状態を変移させることにより、消費電力を削減する。S-MACではまず、1回のアクティブ期間の長さがバックオフ時間やパケット長などから決定される。次に全期間に対するアクティブ期間の割合を *duty cycle* とし、1周期の長さが式 4.1 より求められる。

$$cycle\_time[sec] = \frac{ListenTime}{duty\_cycle} \quad (4.1)$$

S-MACにおいて、アクティブ状態では、通信路が使用中かどうかを判断するため、搬送波の検出をおこなう。スリープ状態では電波の送受信を停止し、次のアクティブ状態までのタイマを設定する。ノードはアクティブ・スリープのスケジュールを決めるため、まず隣接ノードのスケジュールの受信を試み、受信できればそれに合わせることで自ノードのスケジュールを決定する。受信できなければ自身でスケジュールを独自に作成する。その後、自ノードのスケジュールを同期パケットにより隣接ノードへ定期的にブロードキャストする。独自のスケジュールで動作しているノードは短い間隔で1周期の間アクティブ状態となることで、他のスケジュールを検出する。既に他のスケジュールを発見しているノードも長い間隔で1周期の間アクティブ状態となることで、新しいスケジュールの検出を行う。これにより、隣接ノード同士でアクティブ・スリープスケジュールを同期することが可能となる。

#### 4.2.5 スケジュール同期のアルゴリズム

S-MACでのアクティブ時の動作は同期制御期間と送信要求期間に分けられている。同期制御期間は、隣接ノードからの同期パケットにより隣接ノードとの同期のずれを計算し、修正するための期間である。送信要求期間は送信要求が発生したノードが実際にデータを送信するための期間である。S-MACでの同期制御はこれらのうち同期制御期間で行われる。ただしネットワークの初期化の段階では、隣接ノードを発見するために数周期に1度、1周期の間アクティブ状態となり、他のノードからの同期パケットを発見する動作を行う。この場合はアクティブな期間の全てが同期制御期間となる。

隣接ノードとの同期が完了すると、同期パケットはアクティブ期間中の同期制御期間でのみ受信することとなる。しかし、クロックスキューにより各ノードでのタイマの進む速さが異なっているため、同期パケットをやり取りする際にずれが生じる。そのため、このずれを修正する必要がある。式 4.2、式 4.3 に同期制御期間に同期パケットを受信した場合のずれの計算方法を示す。ClockTime は受信ノードの現在の時間、Pkt.sendTime はパケット内に記述された送信時間、PrePktBytes はパケット受信時のプリアンブル、TxTransitionTime

は送信状態に遷移する時間をそれぞれ表している．式 4.2 では，同期パケット送信及び受信にかかる時間の計算方法を示している．また，式 4.3 は同期パケット送信ノードの次にスリープとなるまでの時間と自身の次のスリープになるまでの時間の差を求めている．これにより，同期パケット送信ノードと自身とのタイマのずれを修正することが可能となる．

$$rxDelay[ms] = (ClockTime - Pkt.sendTime) + \frac{PrePktBytes \times 8}{BANDWIDTH} + TxTransitionTime \quad (4.2)$$

$$refTime = Pkt.sleepTime - rxDelay \quad (4.3)$$

S-MAC ではネットワークの初期化の段階でネットワーク内の全てのノードで同期制御を行う．これにより，ネットワーク内の全てのノードが他のノードと通信可能な状態となる．しかし，初期のスケジュールはばらばらであり，ネットワーク内に複数のスケジュールが混在する状態となる．この状態が持続されると，図 4.4 のノード 3 のように異なるスケジュールの境界に位置するノードが多く，多くの時間アクティブ状態となるため，他のノードよりも多くの電力を消費してしまう．これでは，ネットワークのライフタイムが短くなってしまうため，S-MAC では複数のスケジュールを整合化しネットワーク全体で 1 つのスケジュールとなるように，スケジュールの整合化を行う．

#### 4.2.6 従来手法の問題点

従来の間欠通信における同期制御では，基本的にネットワークの初期化の段階で全てのノードで同期制御を行う．また，スケジュールの整合化によりネットワーク全体で基準となるノードを選出し，そのノードのスケジュールに他のノード全てが同期する．さらに定期的に同期パケットを送信することによりネットワーク全体での同期の維持を行っている．しかし，センサネットワークでは，多くの場合環境情報の長期的なモニタリングや局所的に発生するイベント駆動型のアプリケーションなどが利用されるため，通常時はほとんど通信を行わない場合が多い．そのため，従来同期制御では定期的な同期パケットの送信や通信を行わないノードでの同期の維持などによる同期制御のオーバーヘッドが問題となる．また，ネットワーク全体でのスケジュールリングの基準となるノードが死滅してしまった場合，再度ネットワーク全体で基準となるノードの選出を行い全てのノードにおいて新しいスケジュールを取りなおす必要があるためロバスト性にも問題があるといえる．そのため，従来手法において同期制御におけるオーバーヘッドの削減が課題となる．

### 4.3 提案手法

#### 4.3.1 概要

本章では提案するオンデマンド型同期制御手法の説明を行う。提案手法は基本的に、イベント駆動型アプリケーションを想定しており、イベントを検知したノードはその情報をシンクノードに知らせる。その際、ネットワーク全体ではなく部分的に同期制御を行うことで同期制御にかかるオーバーヘッドを抑制することを目的としている。また、上位プロトコルであるルーティングプロトコルで利用される隣接ノードの情報は、ネットワークの初期化の段階で既に取り得できていると仮定する。

#### 4.3.2 非同期状態からの同期

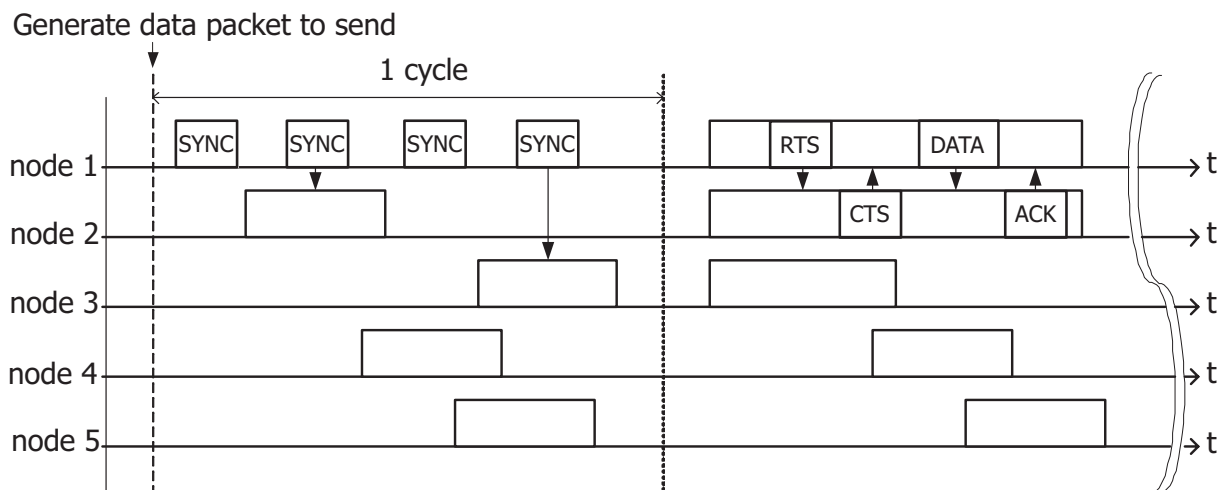


図 4.7 通信時における送受信ノード間の同期の動作例

ネットワーク内の全てのノードは、データ通信のない通常時は定期的な同期パケットの送信を行わないため、各ノードが独自のスケジュールでアクティブ状態とスリープ状態を周期的に変移させる。これにより、通常時の同期に関するオーバーヘッドを削減することができる。しかし、データパケット送信要求が発生した場合、周囲のノードを同期させる必要がある。図 4.7 に提案手法における同期が取れていない状態から周囲のノードを同期させる動作例を示す。また、図 4.8 にその動作例のトポロジを示す。図 4.7 のノード 1 のようにデータパケット送信要求が発生したノードは、まず 1 周期の間同期パケットをアクティブな期間より短い間隔で送信する。これにより隣接ノードはデータ通信要求が生じ

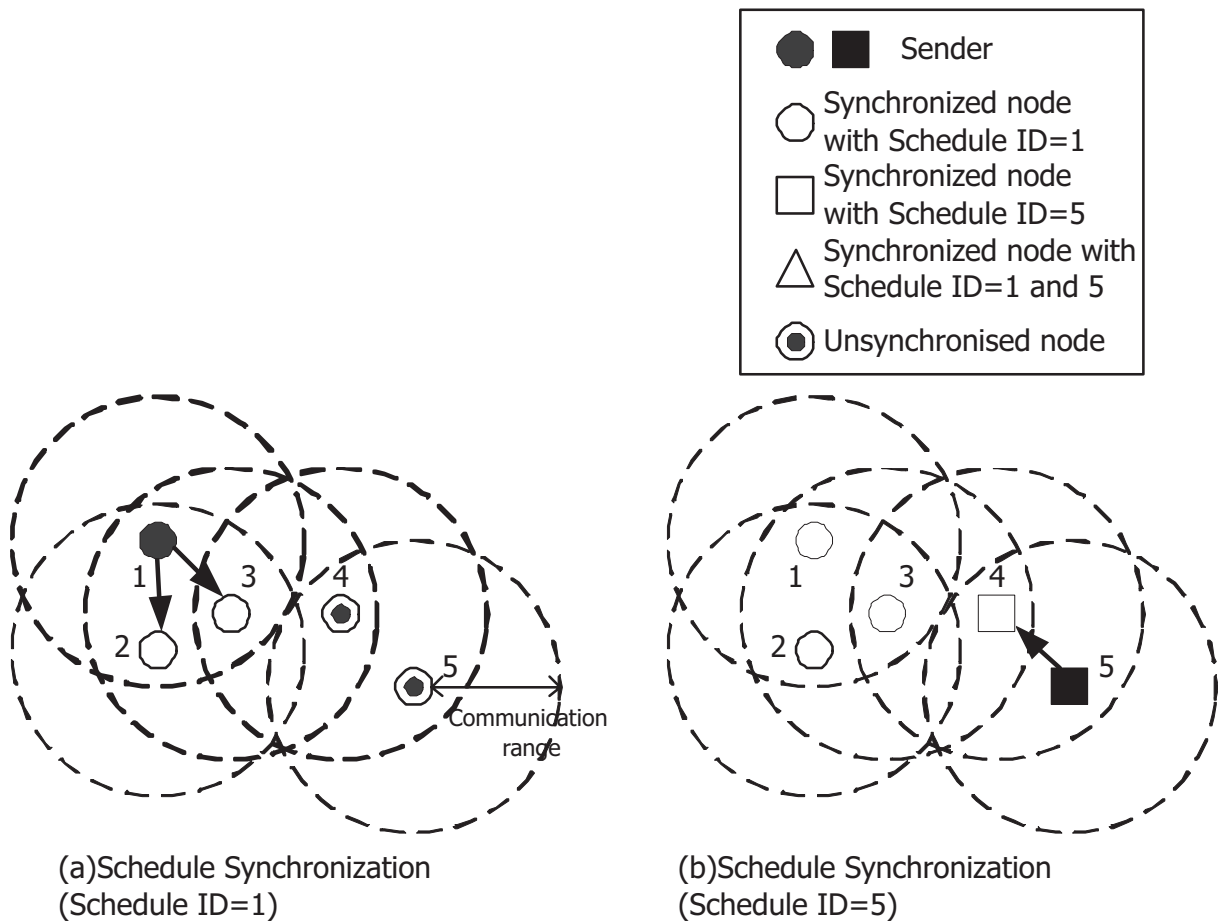


図 4.8 通信時における送受信ノード間の同期のネットワークポロジ

たことを知り、そのノードのスケジュールに同期する。この時、隣接ノードで同様のデータパケット送信要求を持つノードが存在する可能性があるが、その場合も一番早く同期メッセージを送信したノードのスケジュールに同期する。同期が完了すれば通信可能状態となるので、送信要求が生じたノードはデータパケットを送信する。データパケットを受取ったノードも同様の動作を行うことで、最終的にシンクノードへデータパケットを届けることができる。

提案手法における同期が取れていない状態からの同期制御手法は従来手法の B-MAC や WiseMAC などの手法と動作は類似している。しかし、B-MAC や WiseMAC では、1 周期の間 WakeUp パケットを送信し続けることで周囲のノードをアクティブ状態にするが、WakeUp パケットを受信したノードはデータ送信が終了するまでアクティブ状態を維持するため無駄な制御パケットを受信する可能性が高く、多くの電力を消費する可能性がある。

る．一方提案手法では，同期パケットを受信したノードは，その後送信ノードが次にアクティブとなる時間までスリープ状態となることで無駄な同期パケットを受信することなく消費電力を削減できるため，効率よく同期制御を行うことができる．

#### 4.3.3 スケジュールの整合化及び一時的な同期の維持

イベントが起こると前述のような動作によって同期制御を行い，シンクノードへのデータ送信が可能となるが，この手法だけでは同時に地理的に離れた場所でイベントが発生した場合，複数のスケジュールで動作しなければならないノードが発生するため，これらのノードの消費電力が大きくなる．そのため，複数のスケジュールが発生した場合，S-MAC同様それらを1つのスケジュールに収束させる必要がある．

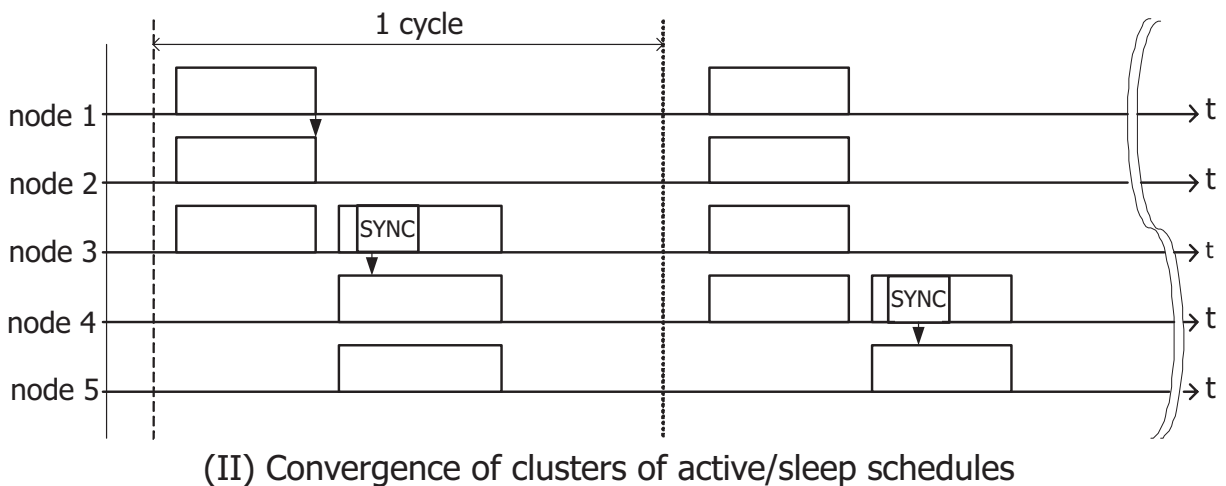


図 4.9 スケジュールの整合化の動作例

図 4.9 に提案手法におけるスケジュールの整合化の動作例を示す．また図 4.10 にその動作例のトポロジーを示す．提案手法でのスケジュールの整合化は S-MAC の手法とほぼ同じ動作を行う．しかし，S-MAC の手法と大きく異なる点は優先なスケジュールを判断するメトリックである．S-MAC では単純に基準となるノードの ID で判断していたが，提案手法ではデータパケット送信要求が発生した時刻が重要なメトリックとなる．図 4.9 の例ではノード 1 がスケジュール ID1 の基準ノードであり，というスケジュール ID はデータパケット送信要求が発生した時刻である．また，ノード 5 はスケジュール ID5 の基準ノードとなっている．ノード 3 は両者のスケジュールの境界となっているノードであり，このノードが優先なスケジュールを判断し，スケジュールの整合化を行う．提案手法でデータ

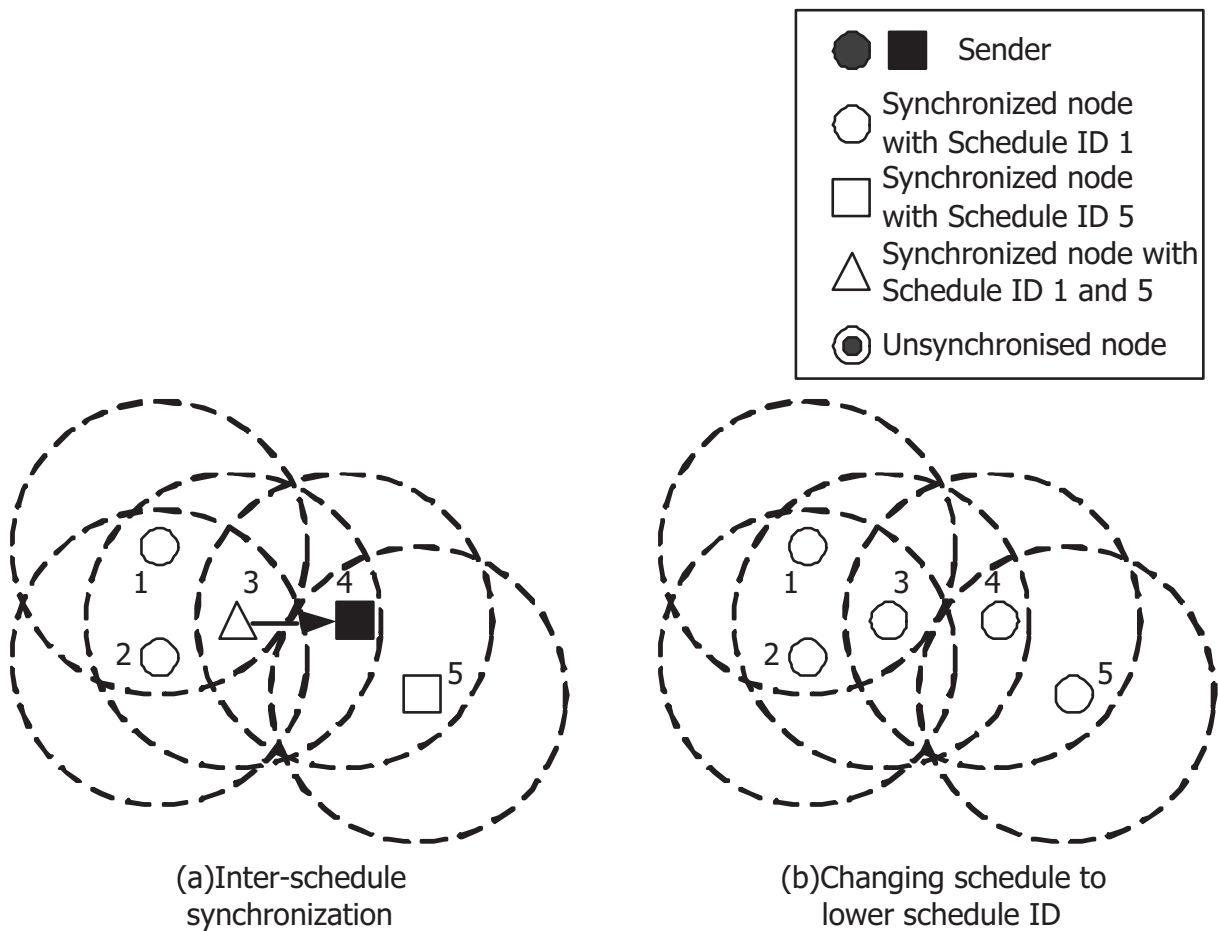


図 4.10 スケジュールの整合化の動作例におけるネットワークポロジ

パケット送信要求が発生した時刻が早い方がより優先されるスケジュールとしている。利点として、S-MACのようにノードIDだけで判断した場合、提案手法では部分的なノード情報しか取得できないため、データパケット送信要求が生じる度にスケジュールの整合化を広い範囲で行わなければならない可能性がある。そこで、データパケット送信要求が最初に発生したスケジュールを優先にすることで、以前に作られた経路のスケジュールを生かすことができるため、スケジュールの整合化を行う範囲を小さくすることが可能となる。

提案手法において、データパケット送信要求が発生し、1周期の間同期パケットを送信した後、スケジュールの整合化などを行ったとしても、従来のプロトコルと異なり定期的な同期パケットの送信を行わないため、時間が経過するとスケジューリングの同期が崩れてしまう。しかし、一度同期したスケジュールを維持できていると、スケジューリングの



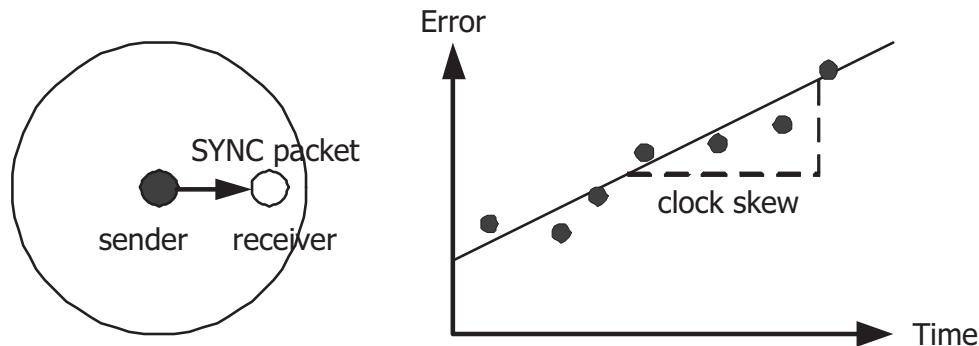


図 4.11 FTSP における送信ノードによる同期パケット送信の動作及び履歴情報の例

同期が取れているノード内でデータパケット送信要求が発生した場合、すぐにデータパケットの送信を行うことができる。また、それ以外のノードでデータパケット送信要求が発生した場合でも、同期が取れているノードまでデータパケットを送信すれば、その後は同期パケットを送信することなくデータパケットをシンクノードまで届けることができる。そのため、提案手法ではデータパケット送信要求が発生した場合に、一時的な同期の維持を行う。

提案手法では、データパケット送信要求が発生した場合のみ同期を取り、その後は定期的な同期パケットの送信は行わない。そのため、同期が取れている短い時間で同期の維持に必要な処理を行う必要がある。このような環境で適応できると考えられる同期の維持を行う手法に Flooding Time Synchronization Protocol(FTSP)[46] がある。FTSP は送信側の時刻と受信側の時刻の差を履歴情報として送信側のクロックスキューを算出することで、送信側と受信側の時刻を同期させる時刻同期プロトコルである。FTSP の簡単な動作を図 4.11 に示す。FTSP では、まず同期の基準となるノードが同期パケットを送信する。このときの同期パケットには基準ノードのパケットの送信時間が記述される。この同期パケットを受信したノードはパケットの受信時間を記憶する。そして式 4.4 にて送信ノードとの時刻のずれを記憶する。この動作を繰り返すことにより、グラフのような送信ノードとのタイムのずれの履歴を取得することができる。この履歴情報を直線近似し、その直線の傾きであるクロックスキューを算出することで送信ノードとのタイムのずれを修正することが可能となる。

$$error[\text{microsecond}] = ReceiveTime - SendTime \quad (4.4)$$

前述した FTSP の手法を利用すれば同期の一時的な維持を行うことが可能となるが、実

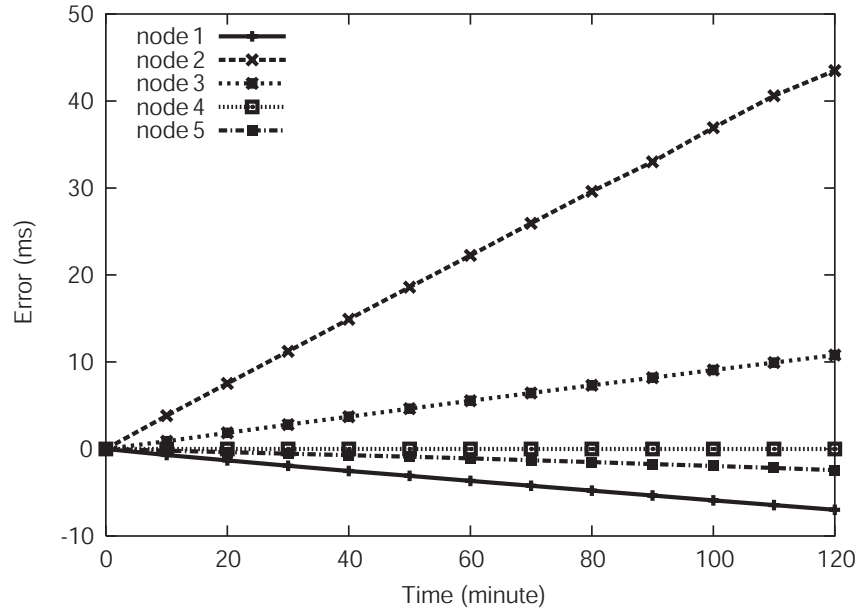


図 4.12 タイマ修正前の時刻のずれ

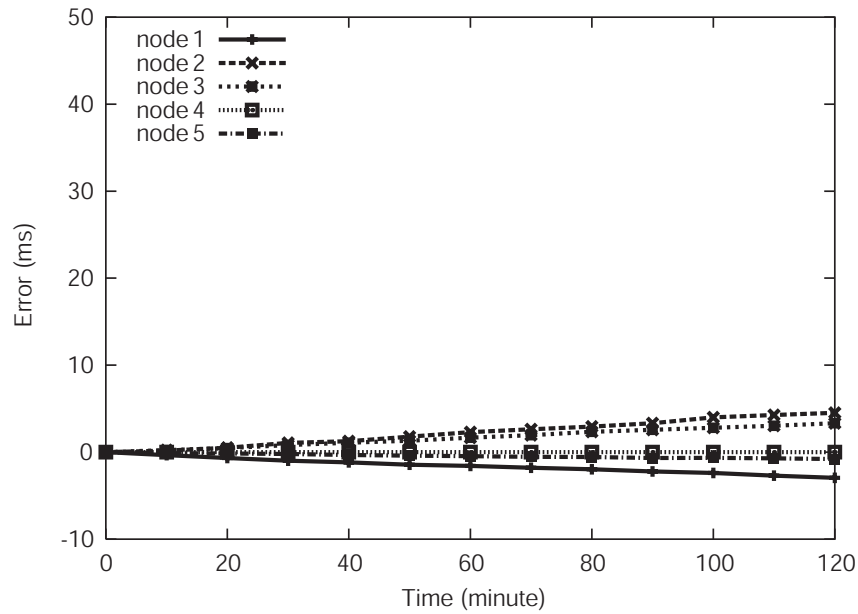


図 4.13 タイマ修正後の時刻のずれ

際にどの程度の時間、同期を維持することが可能なのかわからないため、簡単な実験を行った。FTSP を実装した Crossbow 社製の Mica2[41] を 5 個利用し、履歴数の同期パケットを送信した後、同期パケットの送信を停止した場合の各ノードの時刻のずれを取得した。ここでの履歴数は FTSP のデフォルトの設定である 10 とし、同期パケット送信間隔は 1[s] とした。結果を図 4.13 に示す。実験の結果、1[ms] の同期を約 30 分維持することが可能であることが分かった。ここでの 1[ms] という基準は S-MAC での同期維持の処理で 1[ms] 以上のずれが発生した場合にスケジュールの修正を行うためである。

以上の実験結果から、提案手法においてスケジュールの整合化後に FTSP の手法により一時的な同期の維持を行うことにより、約 30 分の間同期の維持が可能であることが分かった。

#### 4.3.4 ノードの追加と離脱の検出

従来の間欠通信において、ノードの追加や離脱が生じた場合には、全てのノードで同期が取れているためすぐに検出できる。しかし、提案手法では通常状態で同期制御を行わないため、ノードの追加や離脱は検出できない。そのため、新しい検出方法を検討する必要がある。

提案手法において、ノードの追加や離脱を検出するには唯一同期制御を行うデータパケット送信要求が発生した段階で行うことが考えられる。新しくノードが追加された場合は、同期制御が行われた場合でも、他のノードのルーティングテーブルに自身の情報が登録されていないため、データパケットの経路となることはない。そのためデータパケットの通信が完了した後、周囲のノードとは同期が取れているため自身の情報を隣接ノードへ送信することで、周囲のノードに自身がネットワークに追加されたことを知らせることができる。ノードの離脱の場合は離脱するノードが他のノードに知らせることができないため、隣接ノードが検出する必要がある。そのため、本章ではデータ通信時の RTS/CTS を利用したノードの離脱の検出を検討した。データ送信ノードはデータ受信ノードからの CTS によりデータ送信を開始する。しかし、データ受信ノードが離脱している場合は CTS を受信することができないため、データ送信を行うことができない。そこで提案手法ではこの CTS を受信できるかどうかで受信ノードの離脱を判断し、受信ノードが離脱したと判断した場合は、その情報をルーティングプロトコルへ知らせることで、他のルートでのデータ送信を行う。その後、同期の取れた隣接ノードへノードの離脱の情報を送信する。これにより、提案手法においてもノードの離脱を検出することが可能となる。

## 4.4 性能評価

### 4.4.1 概要

提案手法の性能を評価するためにシミュレーションを行った。シミュレーションにはネットワークシミュレータ ns-2[39] を利用した。シミュレーション環境は以下のように設定した。本章ではパケットタイプにユニキャストを利用した。提案手法ではブロードキャストパケットではノードの追加や離脱が検出できないため、基本的にデータ通信にはユニキャストを利用するためである。また、各ノードでのデータパケット発生間隔は指数分布に従うものとする。1 ノードあたりのデータパケット発生の平均間隔は 6400 秒から 100 秒とした。ルーティングプロトコルには Greedy Perimeter Stateless Routing(GPSR)[11] の Greedy Forwarding を利用する。Greedy Forwarding は位置情報を利用し、隣接ノード内で宛先に最も近いノードへデータパケットを送信することでルーティングを可能にしている。本章ではデータパケット送信要求が発生するとデータパケットをシンクノードへ送信する。そのため、宛先ノードは常にシンクノードとなる。

ノード密度はフィールドの大きさとノードの通信半径に依存する。フィールドの幅はマルチホップ環境となるようノード単体の通信可能範囲の直径 40m よりも大きくなるように設定した。これにより全ノードにおける、各ノードの隣接ノード数(通信範囲内に存在するノード数)の割合を減らすことができる。シミュレーションではノード数は GPSR が最も機能する平均隣接ノード数 10 に近くなるように 50 とした。これにより、ルーティングプロトコルでの宛先ノードへ転送することができないデッドエンド状態は起こらないと仮定することができる。1 ノードあたりのデータ発生頻度を 6400 パケット毎秒から 100 パケット毎秒とした。比較対象は S-MAC, 提案手法でスケジュールの整合化と同期の維持を行わなかった場合, また提案手法でスケジュールの整合化と同期の維持を行なった場合とした。シミュレーションにおけるパラメータを表 2 に示す。

### 4.4.2 同期パケットの総送信数

図 4.14 に同期パケットの総送信数を示す。S-MAC ではデータパケットの発生確率が変化した場合でも一定量の同期パケットを送信していることが分かる。これは、データパケットの送信要求の有無に関わらず同期を維持しているため、定期的に同期パケットを送信しているからである。一方、提案手法においてスケジュールの整合化を行わない場合、データパケット発生確率が低い状態ではほとんど同期パケットを送信していない。これは提案手法ではオンデマンドな同期制御を行うため、データパケット発生確率が低くなると

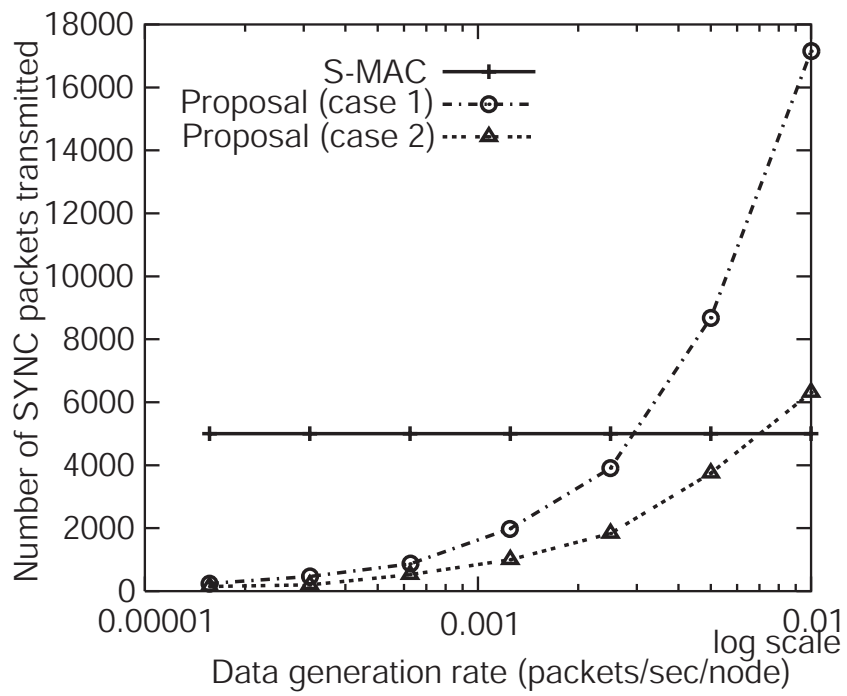


図 4.14 同期パケット送信回数

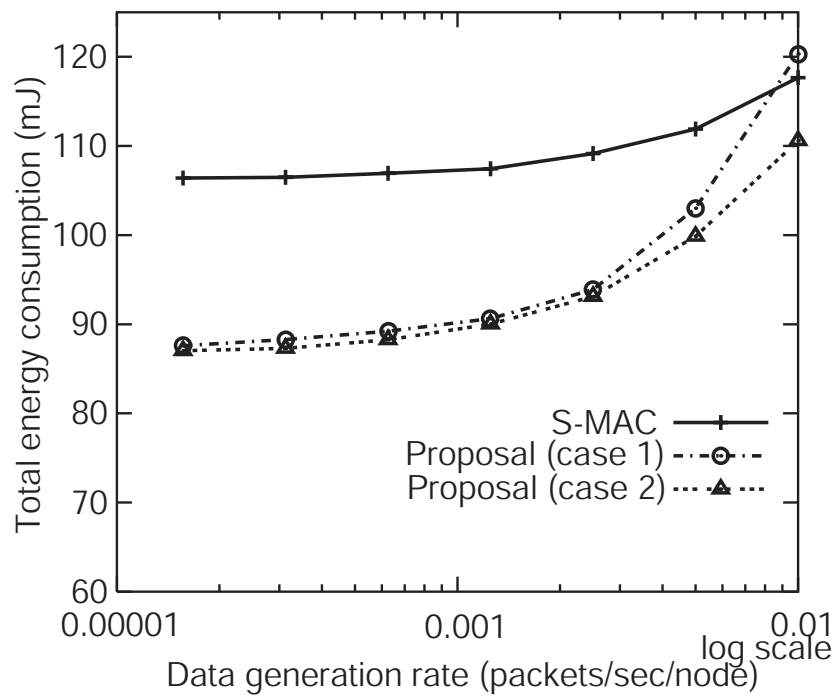


図 4.15 総消費電力

表2 同期制御の評価におけるシミュレーション緒元

Parameter	Value
Rate	20 kbps
Communication range	20 m
Tx power	24.75 mW
Rx/Idle power	13.5 mW
Cycle	1000ms
<i>SYNC_PERIOD</i>	10 sec
<i>SRCH_CYCLES_SHORT</i>	3
<i>SRCH_CYCLES_LONG</i>	30
SYNC period	55ms
Listen period	73ms
Duration of data packet	43ms
Max retry limit	5

同期制御を行う回数及び期間が短くなるためである。しかし、データパケット発生確率が比較的高くなった場合、S-MAC に比べ非常に多くの同期パケットを送信していることが分かる。これは、提案手法で同期の維持を行わない場合、データパケットが発生する度に1周期の間、同期パケットを複数個送信するためデータパケットの発生及び転送の頻度が高くなると最低でもその回数の周期分、同期パケットを送信しなければならないためである。しかし提案手法でスケジュールの整合化及び同期の維持を行う場合、データパケット発生確率が高くなった場合でも同期パケットの送信回数が低く抑えられていることが分かる。これは、スケジュールの整合化及び維持を行う場合、データパケット送信確率が高くなると頻りにスケジュールの整合化が発生し、最終的にS-MAC同様にネットワーク全体で同期を行うこととなるため、同期が取れている状態でデータパケットが発生した場合に同期パケットを送信することなくデータパケットの送信を行うことが可能だからだと考えられる。

#### 4.4.3 総消費電力

図4.15 に総消費電力を示す。S-MAC ではデータパケット発生確率が低い場合と高い場合で大きな変化はなく緩やかに変化しているのに対し、提案手法ではデータパケット発生確率が低くなればなるほど、多くの消費電力が削減できていることが分かる。しかし、スケジュールの整合化及び同期の維持を行わない場合は、データパケット発生確率が高くなればなるほど消費電力が急激に高くなっている。これは、同期パケットの送信数が非常に高くなっているためであると考えられる。一方、スケジュールの整合化及び同期の維持を行う場合は、データパケット発生確率が高くなった場合でも、S-MAC同様の増加率に抑えられ、また約10%の消費電力を削減できていることが分かる。これは、スケジュー

ルの整合化及び維持により最終的にネットワーク全体での同期を行うことができるため、S-MAC と同様の動作となるためであると考えられる。

#### 4.5 本章のまとめ

本報告書では、センサネットワークにおける間欠通信を考慮した同期制御手法の検討を行い、オンデマンド型同期制御手法を提案した。また、シミュレーションにて S-MAC の同期手法との比較を行い、消費電力が削減されることを示した。今後は Mica2 などへの実装及び様々な環境での動作を検証する必要があると考えられる。その際にシミュレーション同様多面的な分析が必要となるため、消費電力などのデータの取得方法などを検討する必要がある。また、今回は CSMA ベースの間欠型 MAC プロトコルにおいてオンデマンド型同期制御手法を提案したが、今後は TDMA 方式の間欠型 MAC プロトコルへの適用についても検討を行っていく必要があると考えられる。

## 第5章 P2P ネットワークにおける適応的センサデータ配置 法

---



## 5.1 本章の概要

近年，実空間情報を収集するセンサネットワークが注目されている [27]．ローカルに配置された無線センサネットワークはゲートウェイを介すなどして，インターネットのような広域ネットワークに接続されることが想定され，多種多様なセンサデータをインターネット上で共有することが想定される．そのため，センサデータの管理，発見，また収集をスケーラブルに行うメカニズムが必要となる．このような大量のコンテンツを扱う環境において，集中管理方式では，大量のデータの管理やロバスト性の確保が困難であるなどの理由から，本章では，ネットワーク内のノードが，センサデータを分散的に管理する状況を想定する．従来研究では検索における負荷分散や，オーバヘッドの削減に着目し，データの更新における負荷分散や，オーバヘッドの削減に十分対応しているとはいえない．

そこで本章では，P2P ネットワーク上でリアルタイムに変動するデータを適応的に配置する手法について提案を行う．提案手法では，連続的な範囲を含むクエリを扱う構造型 P2P ネットワークにおいて，データの更新やデータの検索における負荷分散を行う．また，データの登録や検索の履歴を利用したり，データ検索において，検索パス上にポイントを配置してクエリ転送のショートカットを行ったりすることで，データの更新および検索のオーバヘッドを削減する．さらに，データ更新やクエリ配信のオーバヘッドに応じて，データを提供するノードがデータの代わりにメタデータを登録することでオーバヘッドを抑制する．

以下では，データの登録や検索における負荷分散とオーバヘッドの削減の方法を示す．さらに4章で提案手法の性能評価を行う．

## 5.2 関連研究

### 5.2.1 概要

近年，インターネットの帯域幅の増加やエンド端末の高性能化を背景として，ネットワーク上のノードが対等に通信しあう P2P 型ネットワークが一定の合理性を持つようになっている．またこの数年，ファイル検索の分野でもさまざまな分散検索技術が考案されている．このような分散型リソース発見機構は集中的な制御や管理を行わないため，システムの動作のコストが低い，耐障害性が強い，スケーラビリティが高いなどの利点がある．階層構造と異なり，各ノードは通常，システム内の検索パスのほんの一部にしかかわらない． 図 5.1 にハイブリッド P2P(Napstar) 型リソース発見法を示す．まず，ネットワーク上にインデックスを集中管理する中央サーバを設け，クライアントピアは中央

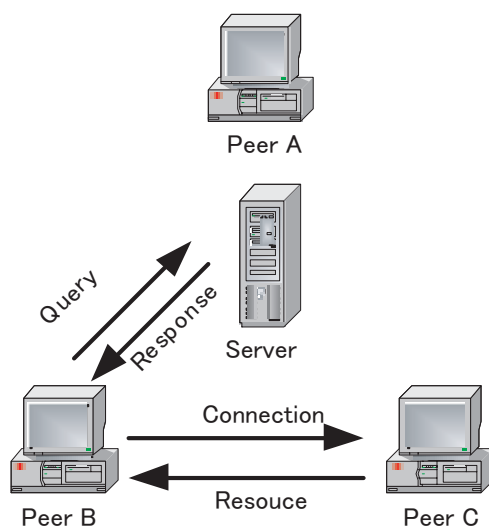


図 5.1 ハイブリッド型リソース発見方式 (例 : Napstar)

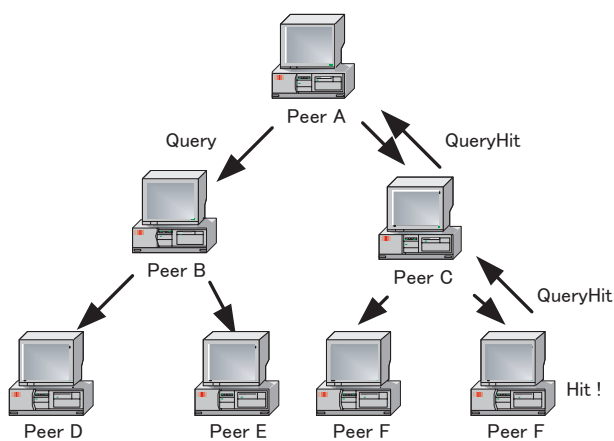


図 5.2 ピュア P2P 型リソース発見方式 (例 : Gnutella)

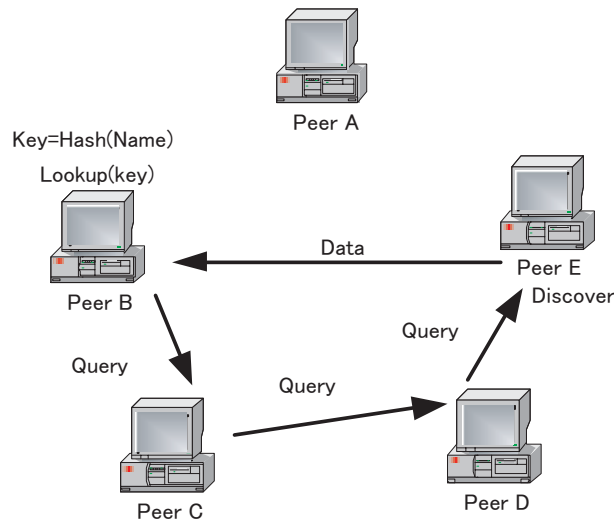


図 5.3 構造型リソース発見方式（例：DHT）

サーバに対して自分が必要とするリソースのインデックス情報を問い合わせる．中央サーバで一元管理するため，管理機構が単純で，必要な情報の検索や発見がしやすいが中央サーバが故障するとサービスが機能しないなどのロバスト性に欠けるという問題がある．また，負荷が特定のノードに集中するなど，スケーラビリティが乏しい[47]．

一方，図 5.2 のピュア P2P(Gnutella) 型リソース発見法では，隣接ノードに自分が必要とするリソース名を記述したクエリをブロードキャストしていくことによりそれを保持するノードを発見する．該当するリソースを保持するノードはクエリヒットにより応答する．クエリの結果はクエリが転送されてきたルートを逆にたどったり，それぞれのピアから要求元のピアに直接返される．各ノードがそれぞれ 10 のノードにクエリを転送すると，ねずみ算式にクエリを受信するノードが増加し，1 分も満たないうちに 100 万ものノードがそのクエリをリアルタイムに処理することになる．各ピアはクエリの結果をローカルに格納しておくことで，次回以降，より賢い検索が可能となる．Gnutella ではこの方法とリクエストのループを避ける方法を併用している．一元管理するピアがいないため，ハイブリッド P2P 型システムよりもロバスト性は高いが，ブロードキャストする方法は検索可能なノード数の増加に伴って線形にパケットの増加が起きる．帯域や，多数のノードの計算処理のリソースを浪費するといった効率が劣る問題が生じる．

### 5.2.2 分散ハッシュテーブル

2.1 節で説明した2つの手法の欠点を補うべく、図 5.3 のようにコンテンツやインデックス情報を分散して保持するアプローチがある。中央サーバが存在しない環境において、どのノードがどのインデックス情報を保持するかという問題が生じる。これを解決するため、どのノードもリソースの名前解決を行うリゾルバとなりうる対称的検索アルゴリズム (symmetric lookup algorithm) である分散ハッシュテーブル (Distributed Hash Table) (以下 DHT) が考案されている。DHT はキーと値を提供し、仮想的なネットワーク上で、データとデータを保持しているノードのマッピングを行う。DHT はキーと値のペアを必ずしもネットワーク上の近い位置に格納しない。DHT は任意の文字列を一意的な値にマッピングする。ネットワークの上の位置に依存しない名前付けの汎用インタフェースを提供することにより、ネットワークから変換された値に対応するデータの取得を行うことができる。フレーズ検索のようなあいまい検索や、前方一致検索といったことはできず、値に対応するデータが一意的に決定される。

インデックステーブルを各ピアが分散管理し、そのインデックス情報を各ピア間で協調動作させることにより、ピア間のトラフィックの削減を図っている。スケラブルで耐障害性の高い検索、発見を実現しようとしている。ハッシュ法は任意の文字列をハッシュキーに変換し、データベースに格納する。共通のハッシュキーを使うことにより、ピアの名前、ファイル名、コンテンツ情報などをハッシュ空間上に統一的にマッピングすることができる。DHT は小さな遅延でノードの参加、脱退を扱い、個々のノードの経路表の維持を簡単にし、参加しているノード間で鍵が公平に分散するように調整する。データが数値鍵で一意的に特定できること、ノードがお互いに鍵を保持しあうことが必要である。DHT は与えられたキーに対してその情報を保持するネットワーク上のノードの位置を返す。負荷を分散させた鍵とノードの対応付け一般に、鍵とノードはすべて  $m$  ビットの数または識別子を用いて特定される。個々の鍵は、ID 空間においてその鍵と近い ID をもつ一つまたは複数のノードに保持される。キーに対応するデータを探すクエリをルーティングすることで、P2P ネットワーク内でデータを探す。

DHT の具体例としてさまざまなシステムが考案されており、それぞれハッシュ空間の次元数、目的ノードまでの平均ホップ数、隣接ノード数が異なるなどの特徴がある。以下では例として CAN[48]、Chord[49] について説明する。

CAN では  $d$  次元のデカルト座標を用いて DHT を実装している。図 5.4(a) のように、座標空間はゾーンと呼ばれる超四角形に分類され、個々のノードはある一つのゾーンを担当する。CAN のネットワークに参加するとき、ノードはある点  $P$  をランダムに選択し、

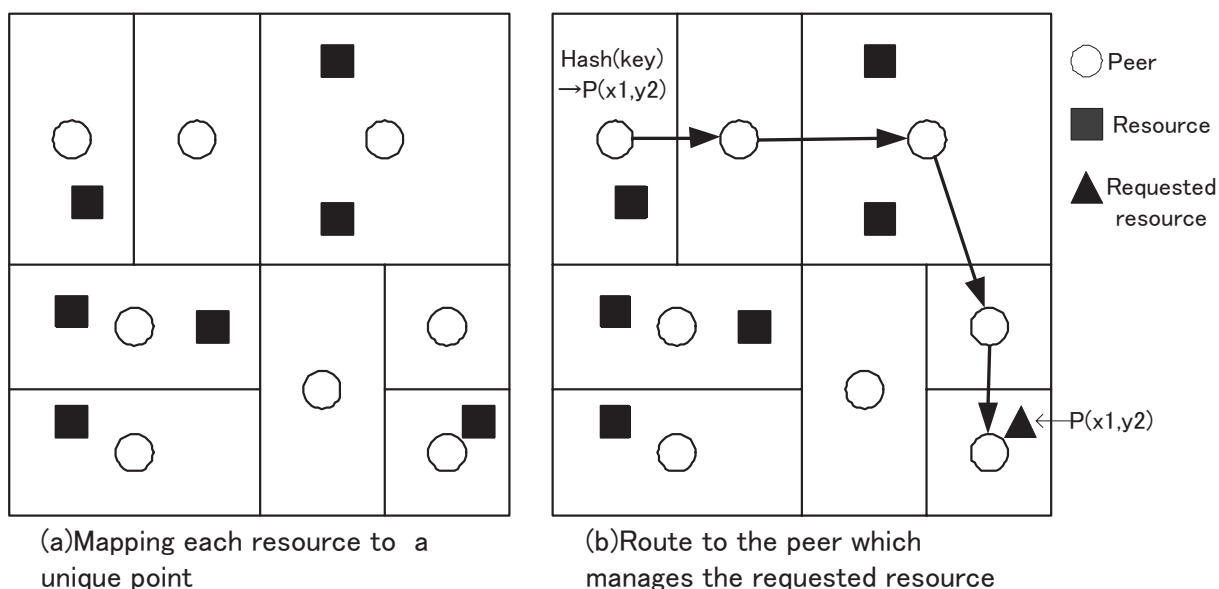


図 5.4 2次元空間におけるCANの動作

JOIN 要求を送信する．すでにCANに参加しているノードはその点Pを担当しているノードにそのメッセージを配信し，点Pを担当しているノードは新規に参入したノードのために自分のエリアを半分に分割し，分割されたエリアを新規に参入したノードに割り与える．

ノードがあるゾーンキーは座標空間上の点にマッピングされ，その座標点を含むゾーンを担当するノードに保存される．ハッシュテーブルはキーを値にマッピングする．図5.4(b)のように，あるキーに対するリクエストはそのキーを持つピアにルーティングされる．キーを保持するノードが隣接ノード内で存在しなければ，隣接ノードのうち，最もキーを持つノードに近いノードに転送される．平均ホップ数は  $(d/4)(n^{1/d})$  となる．例えばノード数が10000の時，次元数が2, 4の時，平均ホップ数はそれぞれ50, 10となる．一般に次元数が挙げればあるノードにおける隣接ノード数も増加する．

ChordはCANと同様にリソースをハッシュキーに変換するところは同じであるが，ハッシュ空間は一次元空間を想定し，ノードに付けられた名前もハッシュキーに変換するところがCANと異なる．CANでは参加ピアがどの空間をカバーするかはピアの任意の意思で決められるが，Chordではハッシュ関数により自動的に位置決めされる．あるピアはハッシュ空間上で1つ前のピアの位置から自分の位置までのキーを守備範囲として管理する．ノードが隣接する前後のリゾルバをそれぞれ，predecessor, successorと呼ばれ，各リゾルバが担当するハッシュ空間の領域は以下のように決められる．

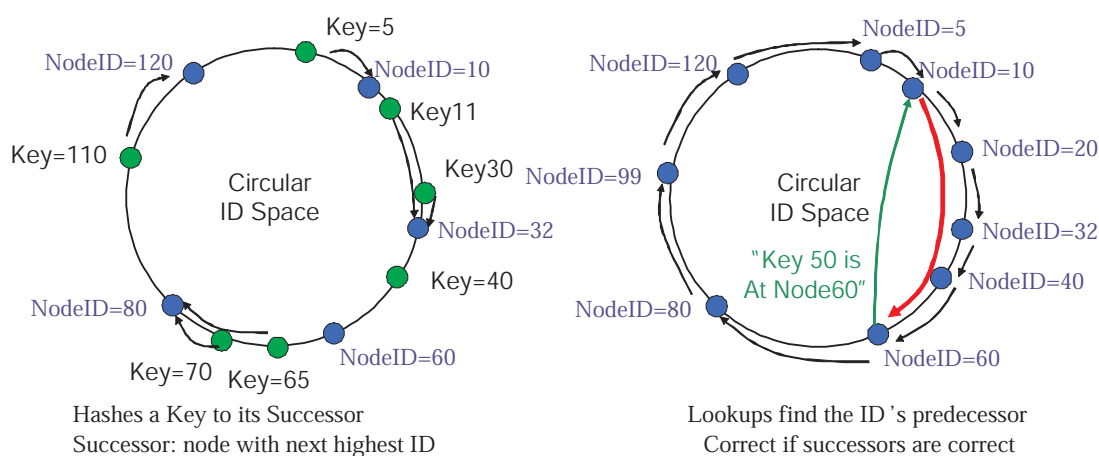


図 5.5 キーとノード ID のマッピングと  $O(N)$  の参照

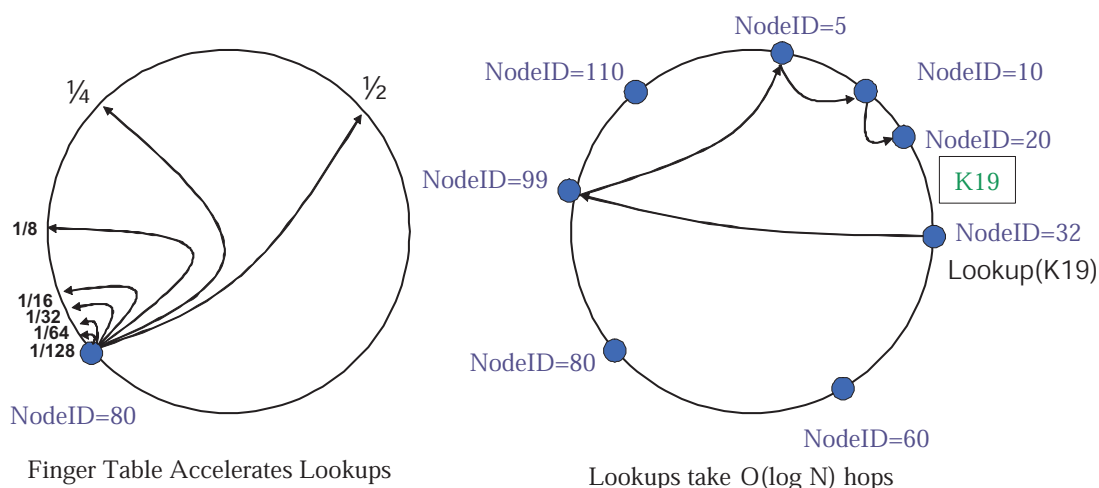


図 5.6 ショートカットルーティングと  $O(\log(N))$  の参照

predecessor のハッシュキー < 担当するハッシュ空間の領域  $\leq$  自ノードのハッシュキー受け持ちの守備範囲に入らないキーのリクエストに対しては successor へ転送する。

図 5.5 左の例に示すように各ノードは仮想的なリングを形成し、ノード 10 ノード 32 ノード 60 ノード 80 ノード 110 ノード 120 のように接続されている。各キーは時計回りに最も近い意ノードに格納される。基本的な参照方法を図 5.5 右に示す。検索要求ノードであるノード 10 がキーの値が 50 に対応するデータを取得したい場合、ノード 10 ノード 20 ノード 32 ノード 40 ノード 60 のようにクエリが転送される。ノード 60 はキーの値が 50 に対応するデータを保持しているため、そのデータを検索要求ノードであるノード 10 に応答する。この方法ではネットワークを構成するノード数を  $N$  とすると

$O(N)$  のホップ数がかかる．Chord ではより性能を向上させるため，より高速に参照できる方法を定義している．図 5.6 右に示すように，7 ビットのハッシュ値により 128 の ID 空間が存在する場合，自ノードの ID より， $2^k (0 \leq k < 7)$  先の IP アドレスの IP アドレスを常に保持することにより，クエリの転送をショートカットすることができる．ノード 80 はノード 81, 82, 84, 88, 96, 112, 17 の IP アドレスを保持する．これにより  $\log(N)$  のルーティングテーブルを保持することになるが，最大  $\log(N)$  のホップ数でキーの値に対応するデータを保持することができる．

Chord の特徴として，各ピアがアドレス空間で自分の直後に位置しているノードのアドレスを把握しておくことによって，指標中の多くのピアが故障していたり，存在しなかったりする場合でもユーザのクエリがアドレス空間で徐々に目標のピアに近づいていくことができる．このため，ノードの同時の参加や退出，あるいは故障に対するロバスト性が高い．

### 5.2.3 ユビキタスネットワーク環境におけるリソース発見機構

コンピュータが日常生活に浸透してユーザに対して支援を行うユビキタス環境ではさまざまなネットワークが共存することが想定される．このような環境ではネットワークシステムの異種性をユーザに意識させずにネットワーク透過なリソース発見機構が必要となると考えられる．

また，インターネット上のリソースのみならず，ユーザが自分の周辺に存在するリソースを発見し利用することも想定され，無線ネットワーク環境では，よりリソースの位置がより動的に変化することが想定され，そのような環境に柔軟に対処することが必要となる．

現在ではインターネットのリソース発見機構は主に比較的リソースの位置や内容の更新頻度が低い静的なコンテンツを対象としている．インターネットを介して実空間情報を収集するサービスを想定すれば，よりスケールが大きい動的な広域分散型リソースを取り扱うことになる．以下では，このような異種ネットワーク間の接続，ユーザが周辺リソースを利用する環境，グローバルなスケールでの実空間に依存したリソースを取得する環境のそれぞれの環境を考慮したリソース発見機構について述べる．

多数のリソースが異種ネットワーク上に分散する環境において，オーバーレイネットワークを実際にどのように構築するかが課題となる．JXTA Project [50] は Sun Microsystems がピアツーピアのアプリケーションの共通基盤を確立することを目標とするプロジェクトであり，ピア間通信のためのさまざまなサービスを提供する相互運用性 (Interoperability)，

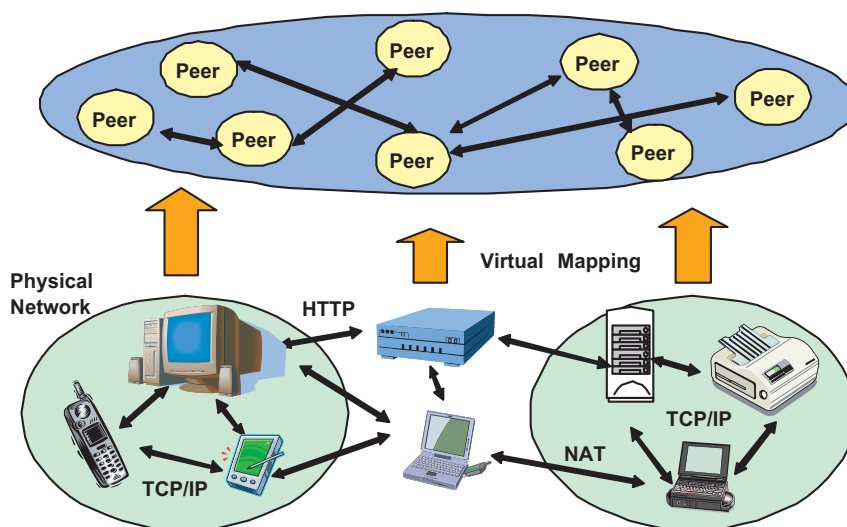


図 5.7 JXTA の仮想ネットワーク

プログラム言語，OS，通信プロトコル，ハードウェアに依存しないプラットフォーム非依存性 (Platform independence)，さまざまなデバイスに実装可能な遍在性 (Ubiquity) の達成を主な目標としている。

ピアの相互発見，ピアグループの構成，リソースの広告と発見，ピア間の通信，ピアの監視などの基本的なプロトコルを提供している。Jini は Java 環境を前提とし，ローカルエリアネットワーク上のリソース発見手段となることを想定しているのに対して，JXTA は Java に非依存で，主に異種ネットワーク間の連携も含めたグローバルネットワーク環境での利用を想定している。図 5.7 のように通信のトランスポートの仕様に依存しない仮想的なオーバーレイネットワークを構築する。IP ネットワークを必須として折らず，アプリケーションレベルのアドレスである 64 バイトの UUID(Universally Unique Identifier) によってピアを識別する。

LAN 上のピアはブロードキャストを用いて相互に発見する。それ以外はグループ内のピアを管理するランデブーピアを用いた間接的な発見を行う。ピアグループを構成し，ランデブーピアを設けることでピアグループの内部に広告されたリソースは相互に発見可能となる。NAT，ファイアウォールなどによりピア同士が直接通信できない環境においてもリレーピアと呼ばれるゲートウェイの役割を果たすノードが中継することで双方向のメッセージ転送が可能となる。リレーピアはグローバルな IP アドレスを持ち，HTTP サーバを立てて，NAT やファイアウォール内にあるピアからの接続を待つ。図 5.8 にお互い NAT やファイアウォール内にあるピア同士がメッセージを交換する場合の動作を示す。ピア 1 がピア 2 にメッセージを転送する場合，まずピア 1 のリレーピアであるリレー



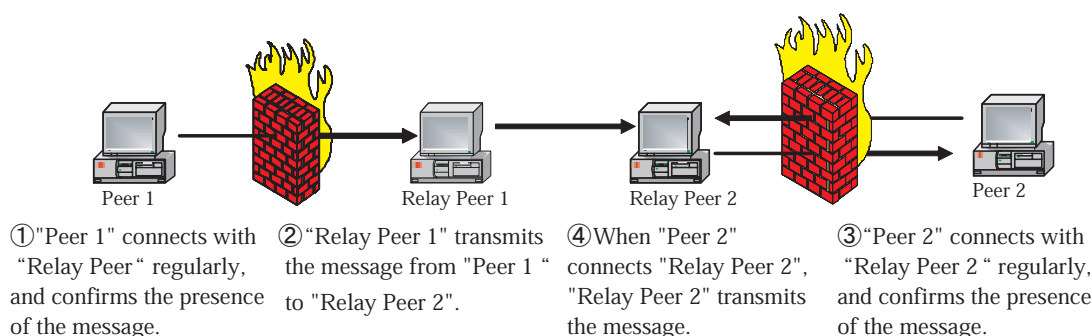


図 5.8 ファイアウォールや NAT 環境での P2P 通信

ピア 1 にメッセージの転送要求を行う (①) . アプリケーションレベルのルーティング情報により, メッセージはリレーピア 2 に転送される (②) . ピア 2 は定期的に関リレーピア 2 に接続し, メッセージの有無を一定間隔のポーリングにより確認している (③) , リレーピア 2 がメッセージを受信すると, ピア 2 の接続があり次第, メッセージをピア 2 に転送する .

インターネット上のリソースのみならず, パーソナルエリアネットワーク上のリソースも発見できれば, 例えば複数のユーザが存在する環境において, 予め携帯電話や PDA などのモバイル端末上に存在するユーザのプロファイル情報を検索し, そのユーザに特化したサービスの提供が可能になると考えられる .

インターネット上に広域に分散したリソースのみならず, コンピュータが日常生活に浸透する環境において, ユーザが身近に存在するリソースを利用してサービスを受けることが想定される . モバイル環境ではネットワークのトポロジーが頻繁に変化し, リソース発見においては効率だけではなく信頼性を確保することも必要である . ノードの移動に加えてコンテンツやサービスの更新頻度が高ければ必要とされる検索情報は完全に個々のノードに保持されなければならない . このような環境ではネットワーク上のすべてのノードにリソース要求をフラッディングする場合もある .

INS(Intentional Naming System)[51] ではネットワーク内のすべてのリソースを網羅する名前とルーティングアドレスのマッチングを管理する . サービスを利用するクライアントは INS のリゾルバである INR(Intentional Name Resolver) の場所を一元管理するブートストラップサーバに接続する .

あるノードが INS のネットワークに接続する際は, ブートストラップサーバに参加要求を通知し, INS のリストを得て, その中からそのノードにとって名前解決の処理が最小となる 1 台の INR を選択する . リソースを指定する名前は name-specifier と呼ばれ, 属性と

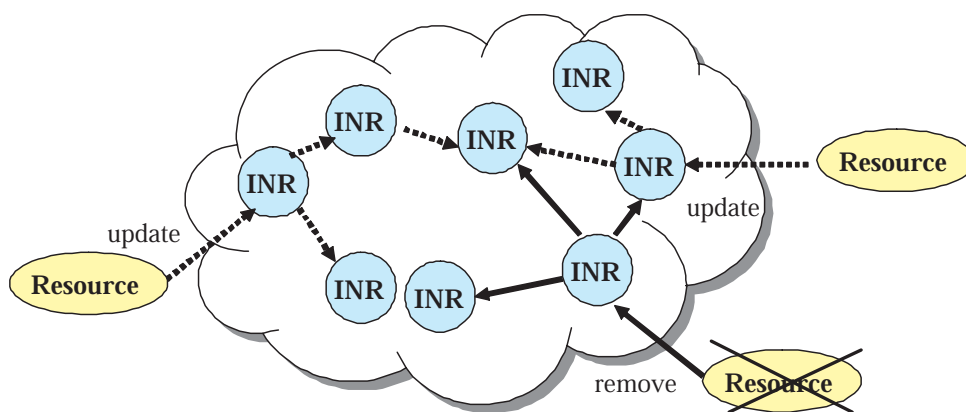


図 5.9 INS におけるリソース更新

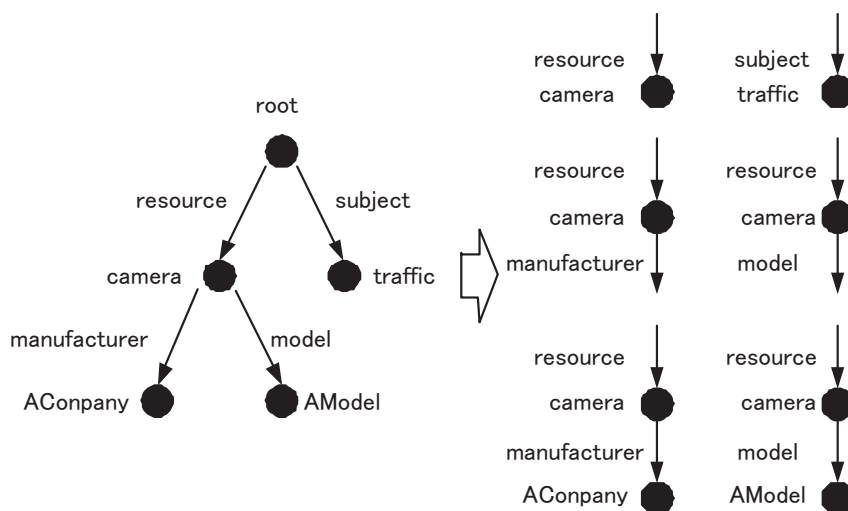


図 5.10 INS における記述の分割

値で表現された AVTree によってクライアントの目的や意思 (intent) を表現する。各レゾルバはあるサービスに対してサービスの接続先の IP アドレス情報を含んだ name-record との対応付けを行い、そのデータ構造は図 5.10 のようにツリー構造の形態をとっている。独立なキーにマッピングされると階層的な表記構造が失われ、意味的なクエリの表現ができない。そのため、図 5.10 のように、ある記述に対してそれに続く属性とその値も抽出する。クライアントからの name-specifier を受信したリゾルバは name-specifier の内容により root から葉へ向かって順番に検索していき、該当する name-record を発見する。クライアントに name-record を回答するだけでなく、クエリをサービスに転送する仕組みもある。サービスをリゾルバが自動的に選択する late-binding によって名前解決後に name-specifier と name-record の対応付けが変化した場合でも動的なネットワーク環境への対応ができる。

複雑なリソースやクエリの記述に対処すること、リソースの状態やネットワークの接続ポイントなどの動的な環境の変化に対応すること、大規模なシステムに対応することが課題となる。

Twine ではピアレゾルバ間で、Chord を用いてハッシュ関数に基づいたリソースの分類を行うことでスケーラビリティを向上させている。任意の属性のセットを用いるコンテンツや位置の制約を受けることなく、属性を階層化することによってクエリに対応する。指定する属性をもったリソースを発見する部分的クエリ各々のリソース表記を数値キーに変換する。クエリルーティングは属性と値のマッチングに基づく。スケーラビリティを確保するために、一部のノードに関するルーティングテーブル情報を保持する。 $N$  をオーバーレイネットワーク上のリゾルバとすると、一般に  $O(\log(N))$  のホップ数が必要となる。図 5.9 のようにリソースがネットワークに参加したり移動したりする際、その更新情報が適切なリゾルバに通知される。リソースが予告なく脱退したり故障した場合においてもその情報はすべてのリゾルバに通知される。自分が選択した INR への更新頻度は高く、それ以外の INR への更新頻度は低い。

ネットワークが動的に変化する場合、リソースの管理のオーバーヘッドが増加すると考えられる。アドホックネットワークにおいて宛先ノードを発見するルート構築やセンサネットワークにおいてある属性を持ったデータを発見する際はフラッディングが用いられることがある。このような場合はリソースを明示的に管理する機構がなくてもリソースを発見することができ、想定環境に応じた機構が必要となると考えられる。

IrisNet(Internet-scale Resource-Intensive Sensor Network Service)[20] ではのシステムの統合化を行い、比較的能力の高いウェブカメラやセンサを利用してインターネット規模

```

<USRegion id="NE">
  <state id="PA">
    <city id="Pittsburgh">
      <neighborhood id="Oakland">
        <block id="block3">
          <parkingSpace>
            <available>yes</available>
          </parkingSpace>
        </block>
      </neighborhood>
      <neighborhood id="Shadyside">
        <block id="block1">
          <parkingSpace>
            <available>no</available>
          </parkingSpace>
        <block id="block2">

```

図 5.11 利用可能な駐車場発見のクエリの記述例

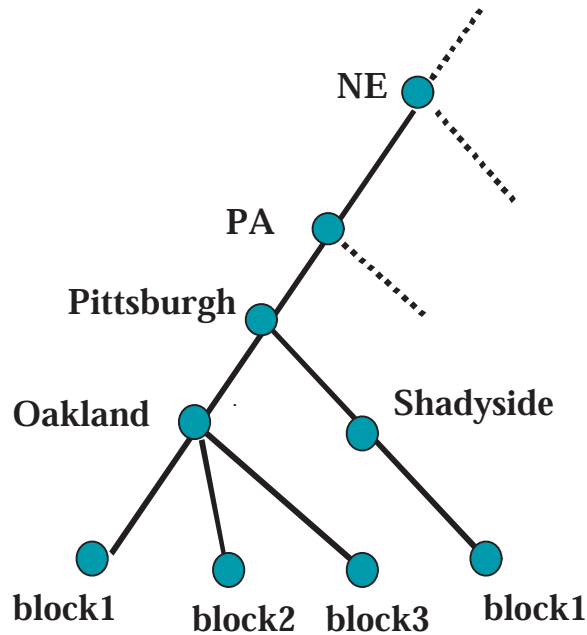


図 5.12 位置情報の階層化構造

で実空間情報を収集・提供するサービスの実現を目標としている．IrisNet ではセンサの種類に依存せず，データ収集の基本的な機能を備えた Sensing Agent(SA) と，センシングサービスごとにデータを格納するデータベースである Organizing Agent(OA) から構成される．OA がどのセンサのデータを保持するかということは動的に決定される．適応的なデータ配置によりクエリに対する応答時間の短縮やトラフィックの削減が見込める．SA のデータはローカルに集約されることにより，大幅にネットワークの帯域の利用を抑えることができる．OA はインターネットの DNS に自分の名前を登録する．例として図 5.2.3 に駐車場発見サービスの XML で記述されたスキーマを示す．データの中には場所のような静的なものや使用中か否かといった利用情報のように動的に変更されるものがある．

リソースを発見するためのクエリルーティングには XPATH が用いられる．図 5.12 のようにセンサのデータを保持する OA は階層化構造をとる．クエリはまず，要求するデータを保持していると考えられる一番下位となるノードへルーティングされる．そのノードの名前は DNS により IP アドレスに変更される．階層構造の頂点であるルートとなるノードに最初にルーティングしないことでルートノードがボトルネックになることを避けている．クエリを受信した OA はクエリに記述されたデータを持っていなければ部分的に他の OA に問い合わせる．このようにクエリの応答を階層化構造をとる複数の OA に分担させたり，OA が応答をキャッシングすることでクエリ処理の負荷分散を行っている．またデータの複製を複数の OA が保持することで耐障害性を高めている．

センサのデータは物理的な位置と密接な関連性があり，リソースの位置は一般に固定的であるためこのような位置に依存した名前付けとリソース管理が適していると考えられる．一方で例えば家庭やオフィスといった比較的規模の小さい空間や複雑な環境における階層的なサービスの記述が可能かどうか，それぞれの SA や OA の名前付けをスケーラブルに行えるかどうかについては，検討の余地が考えられる．

#### 5.2.4 PHT (Prefix Hash Tree) を利用したセンサデータの収集と管理

一般に，センサデータはデータを取得した時刻，センサの種別など，さまざまな属性情報を持つことが考えられる．その中でも地理的位置情報は実空間に依存したセンサデータの属性情報としてきわめて重要であると考えられる．そのため，以下では，ユーザが地理的範囲を指定してデータを検索することを想定する．

DHT を利用してある値の範囲内にあるコンテンツを検索する PHT (Prefix Hash Tree)[52] が提案されており，OpenDHT[54] のプラットフォーム上で実装も行われている [53] ．

PHT の特徴として以下の特徴がある．まず，DHT の参照の機能を利用して，ある連続

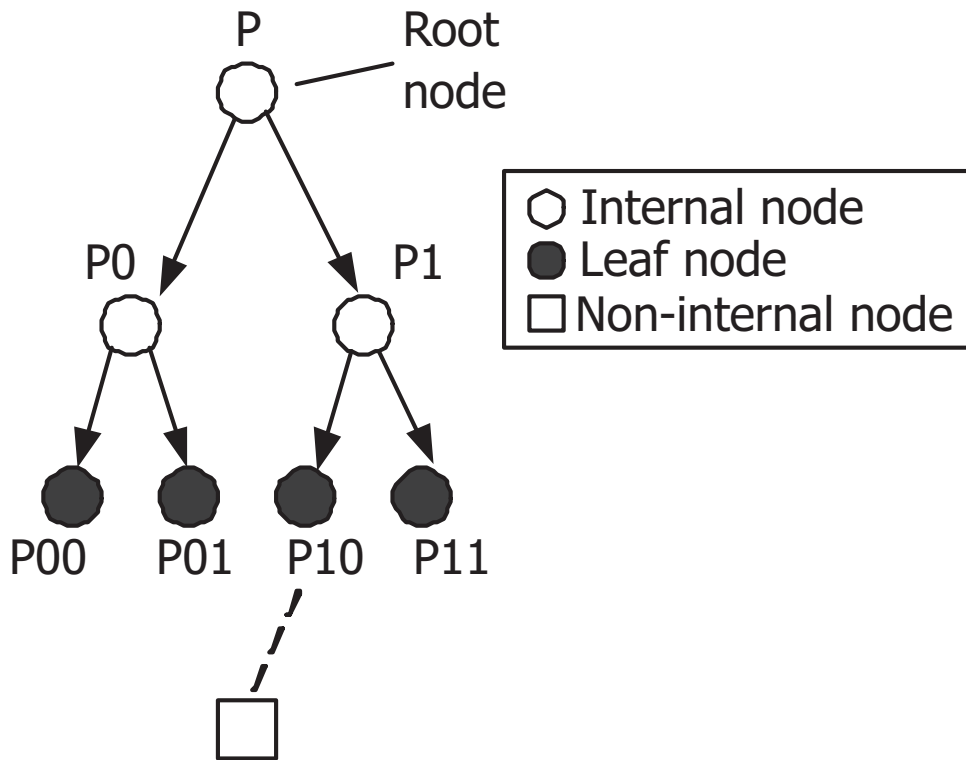


図 5.13 PHT のツリー構造

的な値にあるコンテンツを検索するレンジクエリをサポートすることである。次に、自律分散的に保持するコンテンツの容量の上限を決定できることである。コンテンツの容量の上限はノードの能力から生じるものや、ユーザやアプリケーションの要求から生じるものがある。

以下では、PHT を利用したセンサデータの登録や検索の基本動作を示す。

・データの登録

まず、センサデータが地理的位置情報に基づいて、対応するノードに登録される。2次元の地理的位置情報は、まず1次元の2進数に変換されて扱われる。例えば(2,4)という絶対座標は、2進数表記で(010,100)となる。次に二つの2進数のビット列を上位ビットから交互に並べた011000が1次元の2進数となる。基本的に、ある地理的位置情報  $K$  を保持するセンサデータは、PHT 上で  $K$  と共通のプレフィックスのラベルに対応するノードに格納される。あるラベルに対応するノードは、DHT を利用して求めることができる。

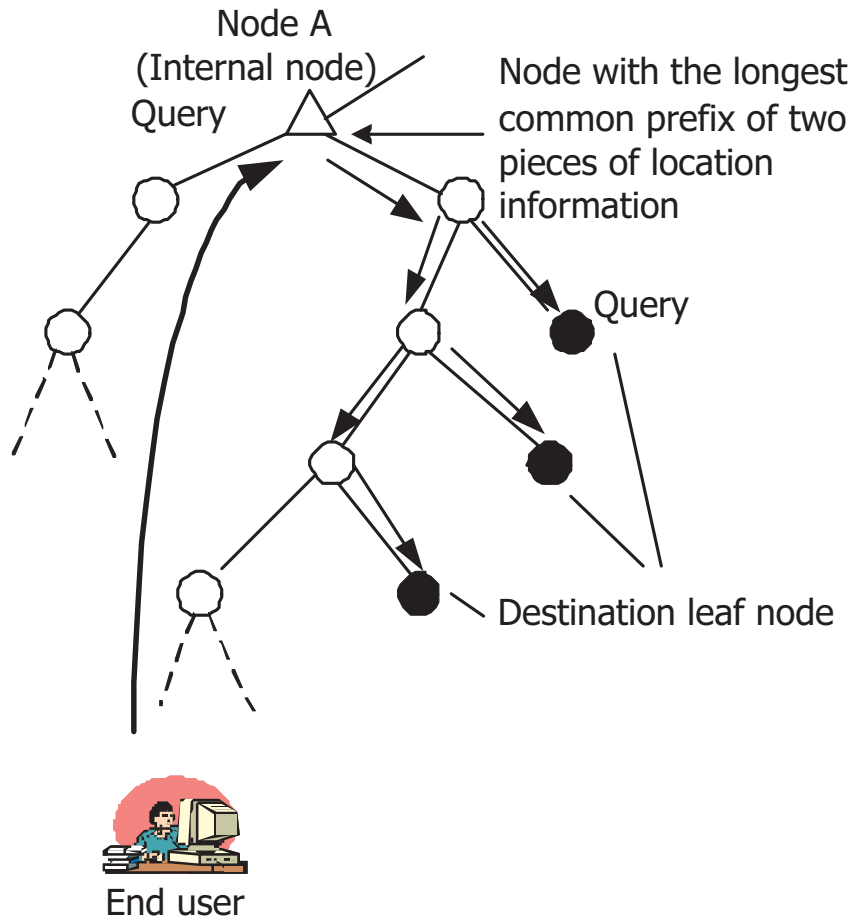


図 5.14 leaf node の発見とクエリの配信 (パターン 1)

例えばラベル P110 対応するノードは，DHT において，キーワード “P110” に対応付けされるノードとなる．

図 5.13 に示すように，PHT のツリーは，ルートノードのラベルが P，その子ノードが P0 と P1，P0 と P1 の子ノードがそれぞれ，P00 および P01，また P10 および P11 というように，再帰的に構成される．親ノードは 2 つの子ノードへのポインタを保持する．データはルートノードから順に格納され，ルートノードのコンテンツの保持量が上限に達すると，子ノードにデータを分割して格納する．新しくデータを保持する子ノードも同様に，自分が保持するコンテンツ保持量が上限に達すると，新たな子ノードにデータを分割して格納する．ツリー上でデータが保持されるノードを leaf node，また，leaf node よりもツリーの上位（ルートノードに近い方）にあるノードを internal node，さらに leaf node よりもツリーの下位にあるノードを非 internal node と呼ぶものとする．

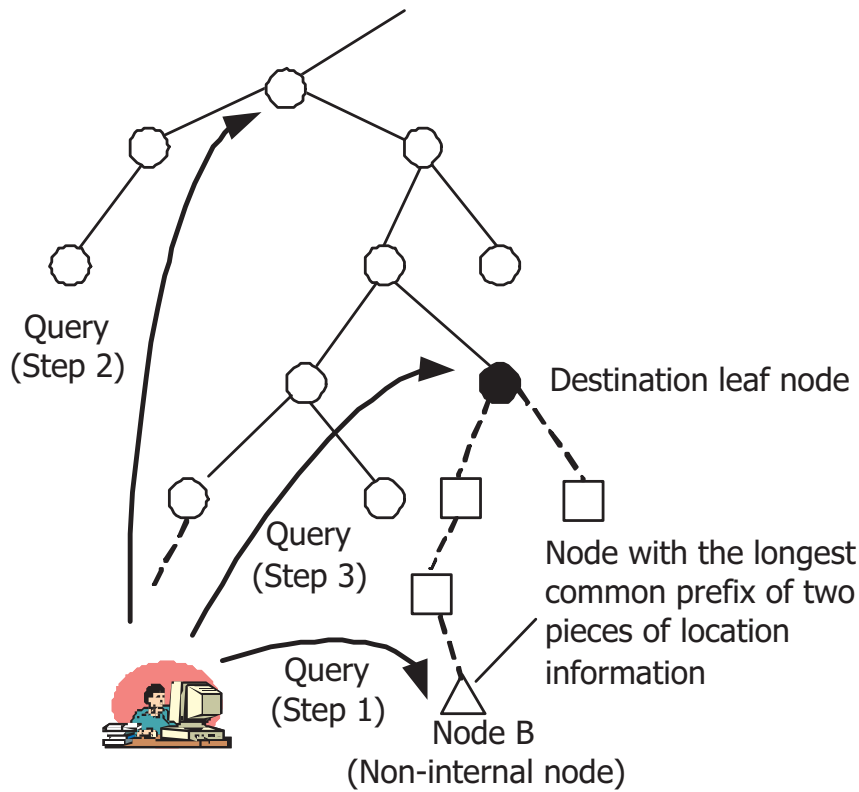


図 5.15 leaf node の発見とクエリの配信 (パターン 2)

・データの検索

クエリに指定される範囲は矩形の左上の点と右下の点の2点で表現される．例えば(2,1)-(3,2) ([001001,001110]) となる．クエリは指定された範囲内に存在する地理的位置情報を含むデータを保持する leaf node に配信される．エンドユーザは、ある地理的位置に対応するデータはその地理的位置のプレフィックスに対応するノードが保持することは分かっているが、どのレベルのプレフィックスかは、検索によって発見する必要がある．クエリに指定される地理的範囲と leaf node の存在位置によってその検索のパターンが異なる．まず、範囲を示す2つの地理的位置情報の共通かつ最長となるプレフィックスのラベルに対応するノードが DHT によって求められる．クエリに指定される範囲が [001001,001110] の場合、P001 のラベルを持つノードが該当するノードとなる．そのノードが図 5.14 のパターン 1 のノード A のように、internal node である場合は、クエリが、要求される地理的範囲に該当するノードにマルチキャスト転送される．一方、図 5.15 のパターン 2 のノード



Bのように、そのノードが leaf node よりも下方にある非 internal node である場合は、2分木探索法で非 internal node よりも上位に存在するノードに順次問い合わせて、leaf node を発見する。いずれのパターンにおいても、クエリの地理的範囲内にあり、クエリの条件を満たすデータを保持する leaf node が応答することによって、エンドユーザが必要とするセンサデータが収集される。

### 5.3 提案手法

#### 5.3.1 概要

データの登録や検索の基本動作は 5.2.4 項で述べた PHT の動作に従うものとする。以下では、まず、データ更新とクエリ配信における負荷分散とオーバーヘッドの削減についてそれぞれ述べた後、適応的なデータ・メタデータの配置によって、データ更新とクエリ配信の総オーバーヘッドを削減する方法について述べる。

#### 5.3.2 データ更新における負荷分散

以下に示すように、leaf node は自分に対する単位時間当たりのデータの更新数が高くなると、自分のデータを分割して、他のノードに保持させることで、更新の集中を回避する。

まず、データを保持する各 leaf node は、データ更新における負荷分散制御を開始するための、単位時間当たりのデータの更新回数の閾値を設定する。データが更新される leaf node において、データの更新があるごとに単位時間当たりのデータの更新回数をカウントする。図 5.16(a) のように、その閾値を超えた場合は、図 5.16(b) のように、子ノードとなる新たな leaf node を作成する。そして管理するデータを、そのデータが持つプレフィックスに応じて分割して、新たな leaf node に格納する。新たな leaf node においても同様に、データの更新頻度の閾値を超えた場合は再帰的に子ノードを作成し、管理するデータを分割して、さらに新たな leaf node に格納する。データの分割は leaf node におけるデータの更新頻度が閾値以下になるまで行われる。

#### 5.3.3 データ更新におけるオーバーヘッドの削減

以下に示すように、データ提供ノードがデータ登録の履歴を利用することで、データ更新における制御メッセージの送信数を削減する。

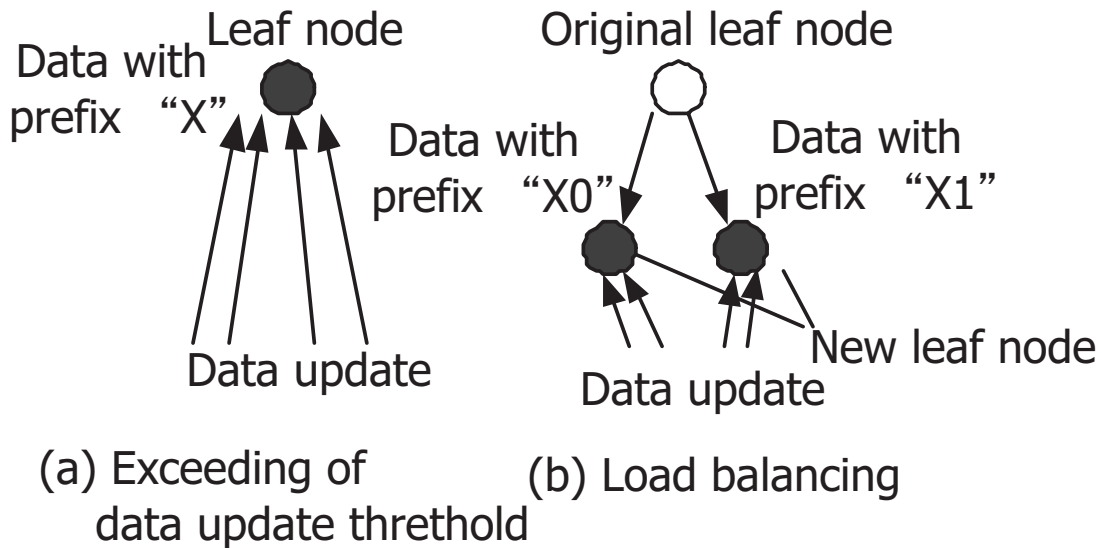


図 5.16 データ更新の負荷分散の動作

まず，初回のデータ登録は PHT のアルゴリズムを用いて自分のデータを登録する leaf node を発見し，その leaf node の IP アドレスを履歴として残す．2 回目以降は，その IP アドレスを参照して，そのデータを更新するノードに対して，そのデータを更新する leaf node であるかを問い合わせる．問い合わせたノードが該当する leaf node であればデータを更新する．一方，ツリーの構造が変化して該当する leaf node でなければ，leaf node を再発見する．

#### 5.3.4 データ検索における負荷分散

leaf node は単位時間当たりのクエリの受信数が大きくなると，自分のデータの複製を生成し，他のノードに保持させることで，クエリのアクセスの集中を回避する．

各 leaf node は，データ検索における負荷分散制御を開始するための，単位時間当たりに受信するクエリの数の閾値を設定する．クエリの受信があるごとに，単位時間当たりのクエリの受信数をカウントする．図 5.17(a) のように，もしそれが閾値を超えていれば，図 5.17(b) のように，複製先のノードに自分のデータの複製を配置する．複製先のノードは，PHT のツリー上にあるラベル以外で，自分のラベルに文字を付け足したラベルに対応するノードとなる．その複製先のノードに問い合わせ，もし，leaf node のデータを配置できる容量があれば，そのノードに複製を配置する．配置できる容量がなければ，自分のラベルに別の文字別の文字を付け足したラベルに該当するノードに問い合わせて複製

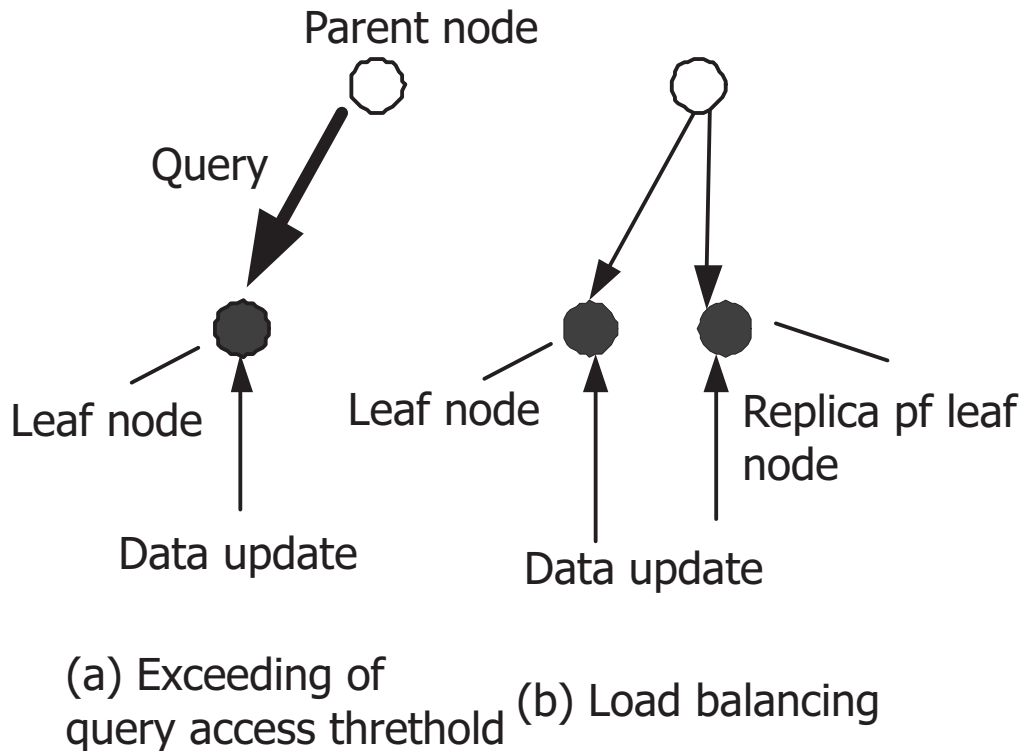


図 5.17 データ検索の負荷分散の動作

先のノードを発見する．複製を配置した leaf node は親ノードに複製先の IP アドレスを通知する．データ提供ノードは，leaf node にデータを登録・更新する度に，leaf node から複製先ノードの IP アドレスのリストを取得し，データをその複製先ノードにも登録・更新する．leaf node が  $N$  個の複製を配置したとすると， $(N+1)$  個の子ノードができる．

クエリが図 5.14 のパターン 1 のように配信される際，親ノードは子ノードにクエリを転送するとき， $(N+1)$  個の子ノードの中から一様な確率でノードを 1 つ選択し，クエリを転送する．クエリのその leaf node へのアクセス数は  $(N+1)$  分の 1 に軽減される．

一方，クエリが図 5.15 のパターン 2 のように配信される場合では，前述したように，leaf node を検索したエンドユーザが，leaf node から複製先ノードの IP アドレスのリストと，leaf node が保持しているデータの地理的範囲を取得する．2 回目以降，クエリが leaf node が保持しているデータの地理的範囲内であれば，leaf node および複製先ノードの中から一様な確率でノードを 1 つ選択し，そのノードがクエリの地理的範囲内のデータを保持しているノードであるかを問い合わせる．もしクエリの地理的範囲内のデータを保持して

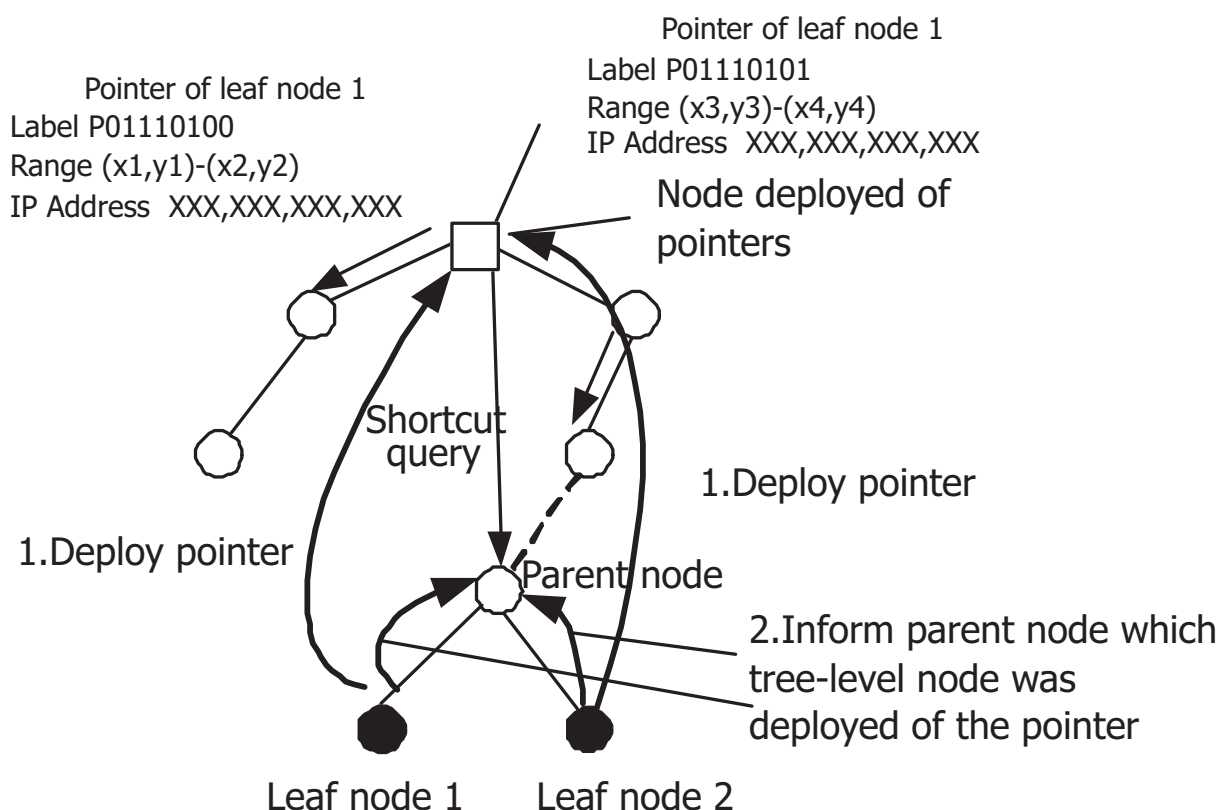


図 5.18 ポインタの配置の動作

いるノードであれば，エンドユーザはそのノードからクエリにヒットしたデータを取得する．一方，そうでなければ，データを再検索する．パターン1の場合の説明で述べたのと同様に，エンドユーザが  $(N+1)$  個のノードの中から一様な確率でノードを一つ選択するため，クエリのその leaf node へのアクセス数は  $(N+1)$  分の1に軽減される．

### 5.3.5 検索におけるオーバーヘッドの削減

図 5.14 のパターン 1 のようにクエリが配信される場合は，検索パスに leaf node へのポインタを配置して，クエリ転送のショートカットを行うことが考えられる．検索オーバーヘッドの削減はデータを保持するノードへのポインタを検索パス上のノードに配置することで行う．ポインタにはデータを保持するノードの IP アドレス，ラベル情報，保持するデータの地理的範囲などの情報が含まれる．図 5.18 にポインタの配置とクエリの転送の一連の動作を示す．各 leaf node はクエリがツリーの親ノードから転送されたときに，クエリが転送された上のノードの IP アドレスを履歴として残す．あるクエリが親ノード

から届いたとき，あるノードの検索パスの順位を  $i$  番目とすると  $1/i$  がそのノードのポイントとして履歴に追加される．検索パスの順位が早ければ早いほど大きなポイントが追加されることになる．これは同じ回数検索パスに現れても，パスの順位が早いほど大きなポイントが追加されることで，ポイントが配置されるのが，よりツリー上で上位になることを狙ったためである．ポイントが配置されるツリー上で上位にあれば，ポイントにクエリが該当したとき，ショートカットによって，より多くの転送回数の削減を行えると考えられる．

単位時間当たりのアクセス数が閾値を超えた leaf node は，検索パスの履歴を参照し，最もポイントが高いノードにポイントに登録する．また leaf node は親ノードにポイントをどのレベルに置いたかを通知する．親ノードは左子ノードと右子ノードがおいたポイントのレベル情報を保持する．ポイントが置かれたノードはクエリを受信したとき，ポイント情報を参照し，もし，クエリに該当する leaf node が存在すれば，leaf node の親ノードにクエリを転送することでクエリ転送のショートカットを行う．leaf node の親ノードは複数の子ノードが存在する場合はその一つを選択する．そしてクエリを子ノードに転送する．なお，2つの子ノードのポイントがなるべくツリーの上位のノードへ配置されれば効果がある．片方のノードだけのポイントが配置されたり，上位のノードにポイントが配置されない場合はクエリの制御メッセージのオーバーヘッドが増加する場合がある．

一方，図 5.15 のように，パターン 2 のように配信される場合では，クエリの送信元ノードであるエンドユーザが一度アクセスした leaf node の IP アドレスと，その leaf node が保持するデータの地理的範囲などの履歴を残しておく．次回以降，クエリに該当する地理的範囲のデータを持つ leaf node の履歴があれば，直接アクセスする．該当するエリアのデータを保持していれば，データを取得する．該当する leaf node でなければ，leaf node の検索をやり直す．

### 5.3.6 クエリ配信とデータ登録のオーバーヘッドを反映した適応的なデータ・メタデータの配置

データ提供ノードでのデータの発生度に leaf node のデータを更新する必要がない場合もある．その場合，データが頻繁に更新される環境において，データ自身の代わりに，データが得られる地理的位置情報，時間，センサの種別といったデータの属性を示すメタデータを登録することが考えられる．メタデータにクエリがヒットした場合に，データを提供するノードにクエリを転送する．これによって，データ自身を高頻度に更新する必要がなくなるため，データの更新のオーバーヘッドは抑制される．しかしながら，データの

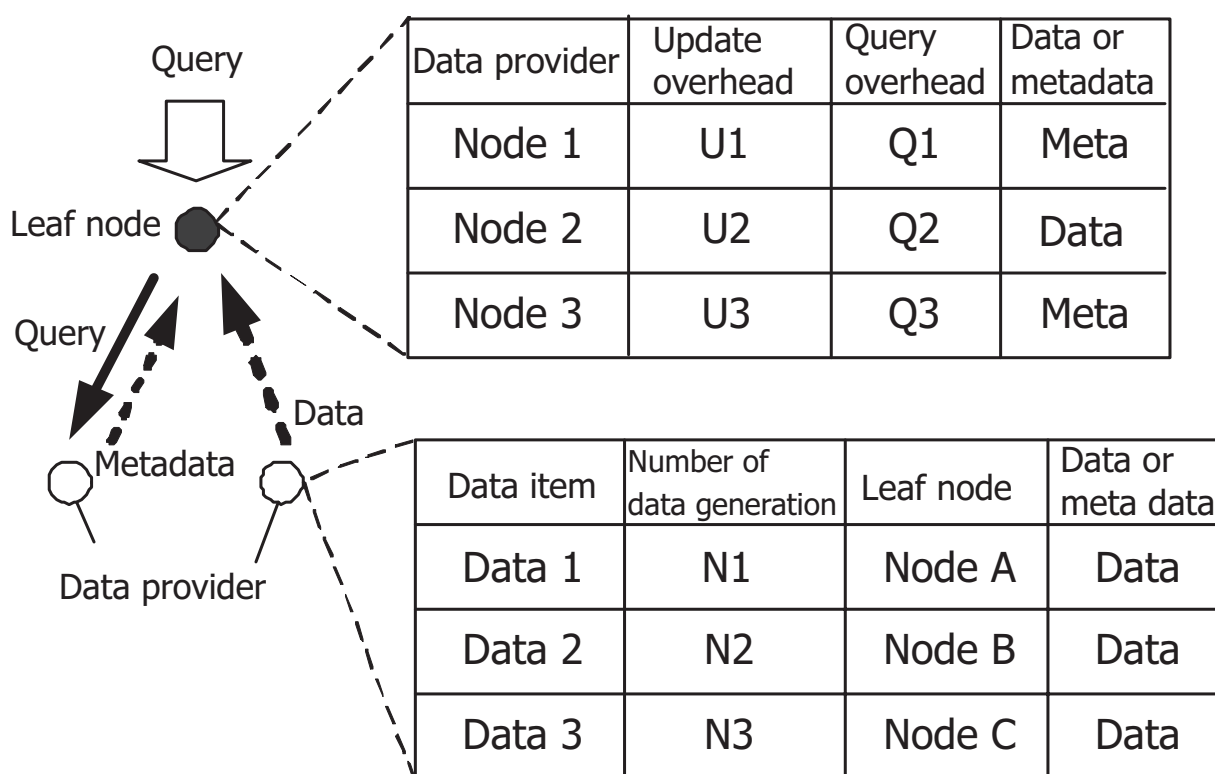


図 5.19 データ・メタデータの配置

更新のオーバーヘッドと比較してクエリの転送のオーバーヘッドが高い場合、メタデータのみを登録した場合は、データのみを更新した場合より、クエリ転送のオーバーヘッドが増加するため、データ更新のオーバーヘッドとクエリ転送のオーバーヘッドの総和が増加する可能性がある。そのため、データの更新頻度やクエリの配信オーバーヘッドを反映させて、メタデータやデータを適応的に配置する必要があると考えられる。以下にそのアルゴリズムを述べる。

データ提供ノードはまず、図 5.19 に示すように、そのデータのメタデータを leaf node に登録する。メタデータには、データ登録ノードが提供するデータの単位時間当たりの更新回数と、メタデータを 1 回登録したときの制御メッセージの送信回数が含まれる。leaf node は、両者の積から単位時間当たりの更新オーバーヘッドを求め、これを記録する。これによって、leaf node は、実際にデータ提供ノードからのデータの送信がなくとも、データ提供ノードでどのくらいの頻度でデータが発生し、仮にデータ自体を登録したときに、どのくらいの頻度でデータの登録があるかを知ることができる。

leaf node は、自分が保持しているメタデータにクエリがヒットすると、データ提供ノード

ドごとに、クエリを転送する。複数のメタデータにクエリがヒットしても、そのメタデータを提供するノードが同一であれば、クエリの転送回数は1となる。クエリはデータ提供ノードに直接送信されるため、1回クエリを転送したときのクエリの制御メッセージの送信回数は1となり、クエリの制御メッセージ送信数がクエリのオーバーヘッドとなる。ノードごとに、単位時間あたりのデータ更新のオーバーヘッドとクエリのオーバーヘッドを比較し、もし、クエリのオーバーヘッドの方が大きければデータ自体を更新させる。データ自体を登録した場合、データを更新したときの単位時間あたりの制御メッセージの送信回数の総和がデータの更新のオーバーヘッドとなり、ノードごとのクエリのヒット回数がクエリのオーバーヘッドとなる。一定期間おきにノードごとにデータ更新のオーバーヘッドとクエリのオーバーヘッドを比較し、データ登録のオーバーヘッドの方が大きければデータ提供ノードにメタデータを登録させる。反対にクエリのオーバーヘッドの方が大きければデータ自体を登録させる。

#### 5.4 性能評価

提案手法の有効性を検証するために、ネットワークシミュレータ p2psim[55] を用いて性能評価を行った。DHT のアルゴリズムは Chord[49] を利用した。100×100 のエリアにノードを配置し、各ノードはそれぞれ 10 種類の位置情報を持つデータを、発生間隔が指数分布に従うランダムな間隔で発生するものとした。データ更新およびデータ検索の負荷分散の評価ではノード数は 5000、それ以外の評価では 1000 とした。検索範囲の広さを、検索履歴利用の評価では 0.01×0.01、それ以外の評価では 10×10 とした。各ノードのデータ保持数の上限を 50 とした。

データ更新の頻度に偏りがある環境では、エリア内に 10 箇所存在するホットスポットエリアの半径はデータ更新における負荷分散の評価、適応的データ・メタデータ配置の評価においてそれぞれ 10、5 とした。ホットスポット内のデータの平均発生間隔は、データ更新における負荷分散の評価、適応的データ・メタデータ配置の評価において、それぞれ 1 データあたり 50 秒、10 秒として、ホットスポット以外のエリアの平均発生間隔は 1 データあたり 3000 秒とした。ホットスポットエリアのデータ更新およびデータ検索における負荷分散制御を行うデータ更新数やクエリの受信数の閾値は 0.5 回毎秒とした。

検索されるデータの地理的範囲に偏りがある環境では、検索範囲の中心点の発生頻度分布は、エリアの中心点が平均点となる 2 次元のガウス分布 (標準偏差  $\sigma=10.0$ ) に従うものとした。

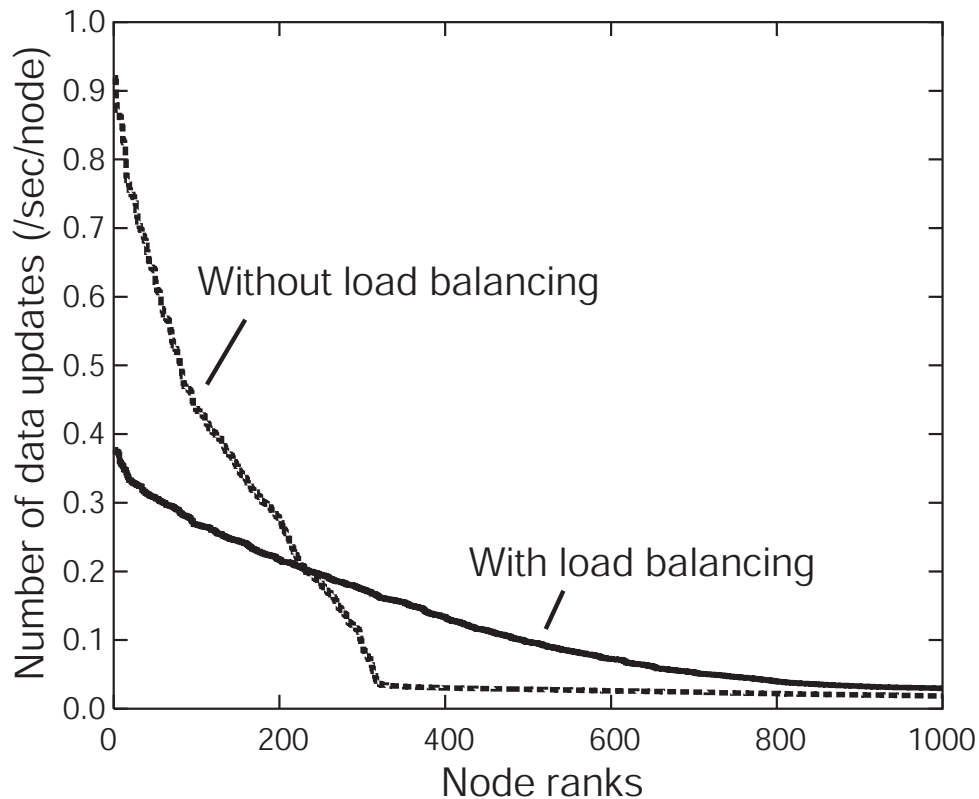


図 5.20 ノード順位とデータ更新数の関係

#### 5.4.1 データ更新における負荷分散の効果

図 5.20 にデータ更新の頻度に偏りがある環境における，データ更新数の上位 1000 位まで順位と，そのノードにおけるデータの単位時間当たりの更新回数との関係を示す．なお，データを提供するノードでデータが更新されると，そのデータの登録先のデータも更新されるものとする．図 5.20 より，負荷分散制御を行った場合は，データの更新回数の上限が抑制されて，更新回数のノード間による差異が小さくなっているが分かる．これはデータの更新頻度がある一定値を越えた場合は，leaf node が自分のデータを分割して複数のノードで管理させるため，1 つの leaf node あたりのデータ保持数が減少し，それに伴ってデータの更新頻度も減少するためである．

#### 5.4.2 データ更新における履歴の利用の効果

図 5.21 にデータ発生率と，データ更新における制御メッセージ送信数の関係を示す．図 5.21 より，データ登録の履歴を用いることで，データ発生率が増加しても，制御メッセー



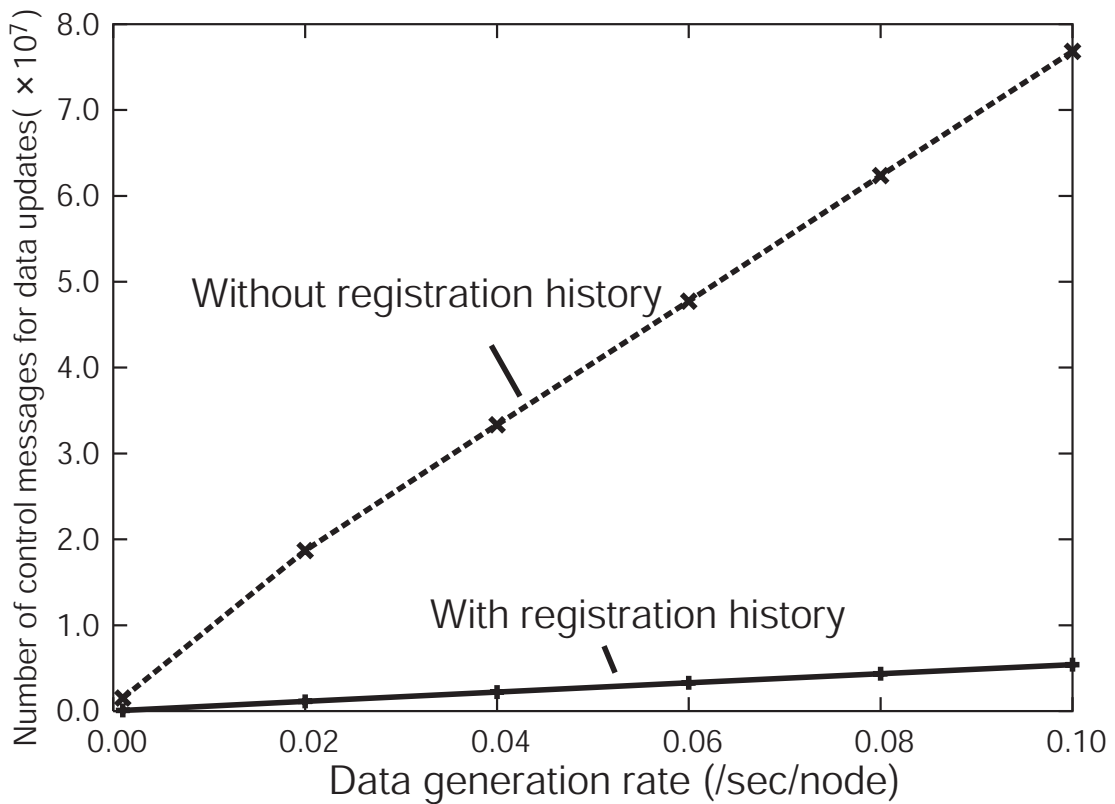


図 5.21 データ発生率とデータ登録における制御メッセージ送信数の関係

送信数を抑制できることが分かる。

#### 5.4.3 データ検索における負荷分散の効果

図 5.22 に、検索されるデータの地理的範囲に偏りがある環境における、leaf node におけるクエリの受信数の上位 1000 位まで順位と、そのノードにおけるクエリの単位時間当たりの受信回数との関係を示す。図 5.22 より、データの複製を配置することにより、クエリのアクセスが複数のノードへ分散されるため、クエリの受信数の上限が抑制されていることが分かる。

#### 5.4.4 データ検索におけるポインタ配置の効果

図 5.23 に、検索範囲の広さを  $10 \times 10$  としたときの、データ発生率と検索における制御メッセージ送信数の関係を示す。検索エリアが  $100 \times 100$  では、図 5.14 のパターン 1 の場合のように、クエリがツリー上でマルチキャストされることが多い。図 5.23 より、ポイ

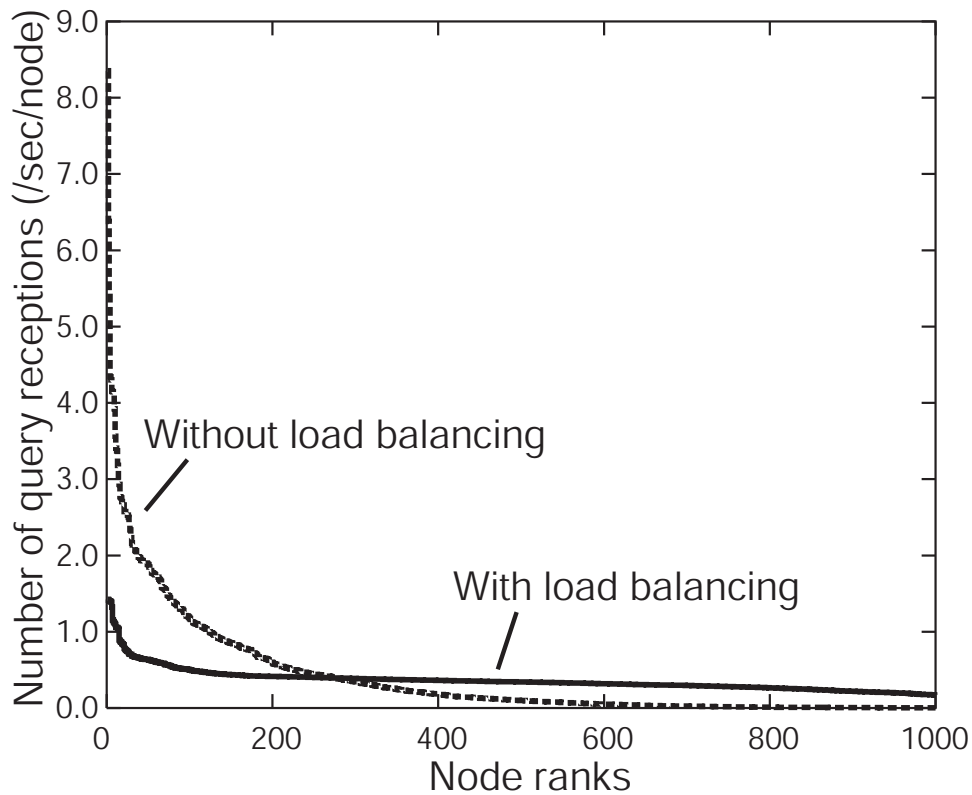


図 5.22 ノード順位と leaf node におけるクエリの受信数の関係

インタを配置する方式はトラフィックが低いときにはポインタの配置数が少なく、片方の子ノードだけのポインタが配置されることが多いため、冗長な経路で制御メッセージが送信される可能性があり、ポインタを配置しない場合よりも制御メッセージ送信数が多くなる場合もある。一方、トラフィックが高くなると、多くのポインタが配置されるため、クエリの制御メッセージのショートカットの効果が大きくなり、ポインタを配置しない場合よりも制御メッセージ送信数が少なくなる。図 5.23 より、本評価ではポインタの配置によるクエリの送信数の削減の効果は弱く、今後より効果的なポインタ配置については検討の余地があると考えられる。

#### 5.4.5 データ検索における履歴の利用の効果

図 5.24 に、検索されるデータの地理的範囲に偏りがある環境における、検索範囲の広さを  $0.01 \times 0.01$  としたときの、クエリ発生率と検索における制御メッセージ送信数の関係を示す。検索エリアが  $100 \times 100$  では、図 5.15 のパターン 2 の場合のように、1 回の検索で

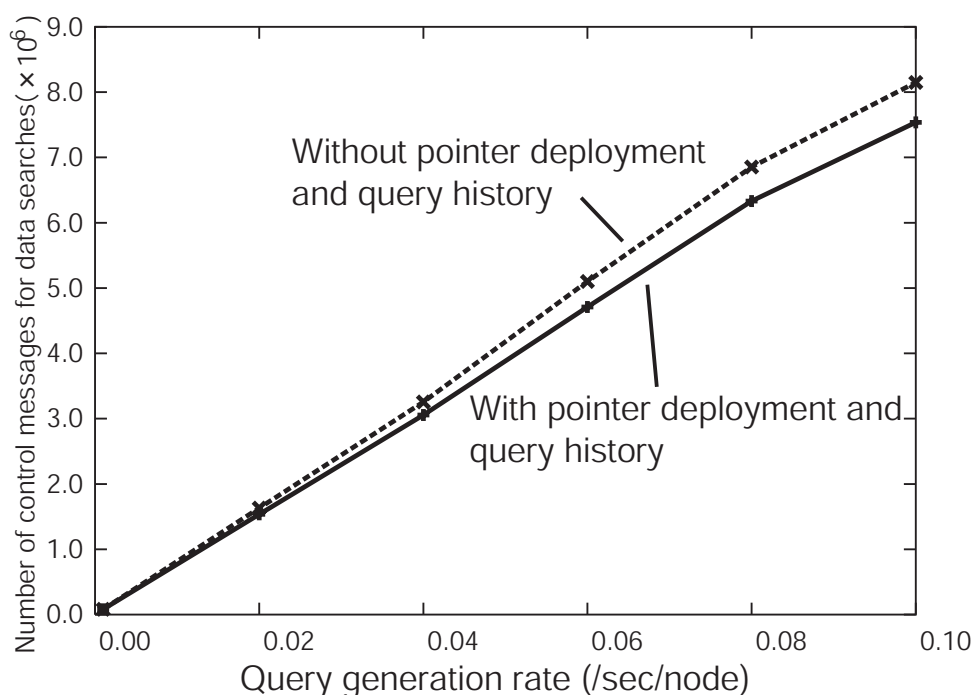


図 5.23 データ発生率と検索における制御メッセージ送信数の関係 (検索範囲の広さ=10×10)

クエリが一つ leaf node にのされることが多い。図 5.15 のパターン 2 の場合において、履歴を利用することで、leaf node の検索のオーバーヘッドが抑制されるため、制御メッセージ送信数を削減できることが分かる。

#### 5.4.6 適応的データ・メタデータ配置の効果

図 5.25 にクエリの発生率を平均 0.001 回毎秒とし、データの発生率を変動させたときの総制御メッセージ送信数を示す。データのみを登録する場合は、データが発生するたびにデータを登録するため、データの発生率の増加に応じて総制御メッセージ送信数が増加していることが分かる。一方、メタデータを用いる場合や提案手法の場合は、頻繁に更新されるデータではなく、メタデータを登録すればよい。そのためデータ発生率が上昇したときでも、データのみを用いる場合よりも、総制御メッセージ送信数が抑制されていることが分かる。

図 5.26 にデータの発生率を平均 0.001 回毎秒とし、クエリの発生率を変動させたときの総制御メッセージ送信数を示す。メタデータのみを用いる場合は、leaf node がメタデータにヒットしたすべてのクエリをデータ提供ノードに転送する。一方、データのみを登録

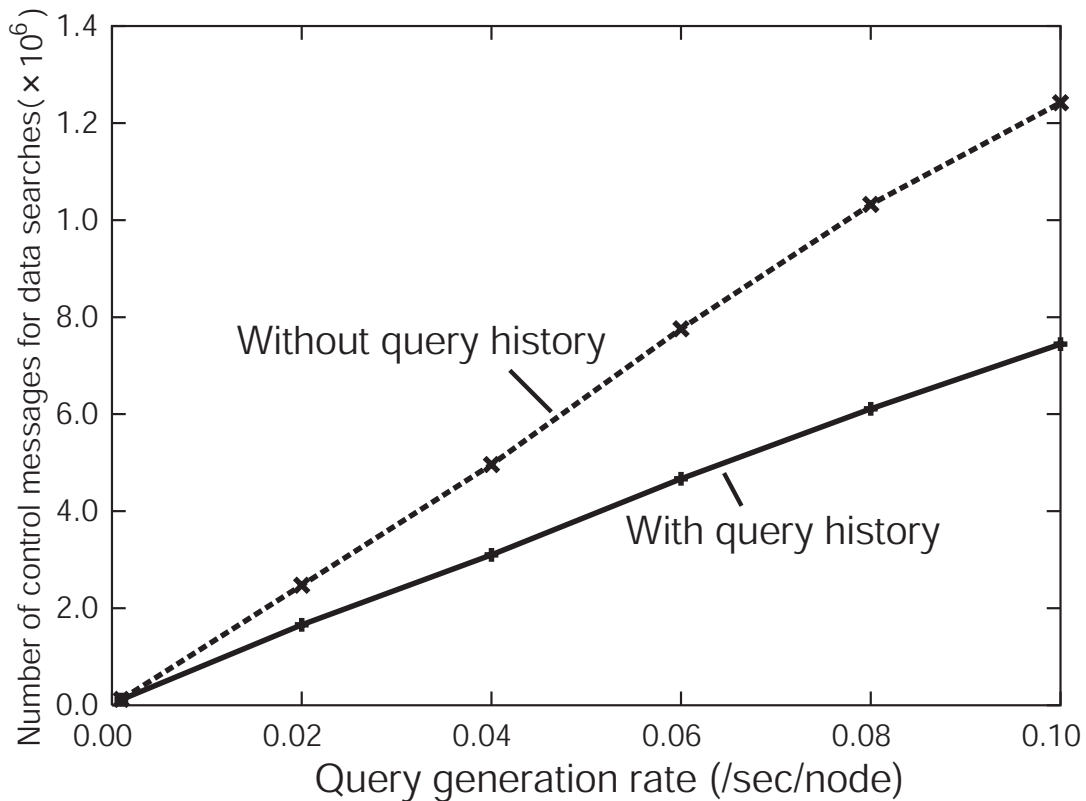


図 5.24 クエリ発生率と検索における制御メッセージ送信数の関係 (検索範囲の広さ =  $0.01 \times 0.01$ )

する場合や提案手法の場合は，leaf node が，クエリにヒットしたデータを直接返答するため，クエリの転送のオーバーヘッドが抑制される．そのため，クエリの発生率が上昇したときでも，メタデータのみを用いる場合よりも，総制御メッセージ送信数が抑制されていることが分かる．

図 5.27 にデータ更新の頻度と検索されるデータの地理的範囲に偏りがあるときの，クエリ発生率と総制御メッセージ送信数を示す．データの発生率が高いときは，メタデータを用いる．これによって，データ更新における制御メッセージ数がデータのみを更新する場合と比較して，データ更新における制御メッセージ送信数を抑制できる．さらに，クエリの発生率が高いときは，データを登録・更新することによって，メタデータのみを登録する場合と比較して，クエリの配信における制御メッセージ送信数を抑制できる．提案手法では，ノードがデータ登録・更新のオーバーヘッドとクエリ配信のオーバーヘッドに応じて，データ・メタデータの登録・更新を適応的に行うため，どちらか一方のみの登録・更新を行う場合よりも，総制御メッセージ送信数を抑制できることが分かる．

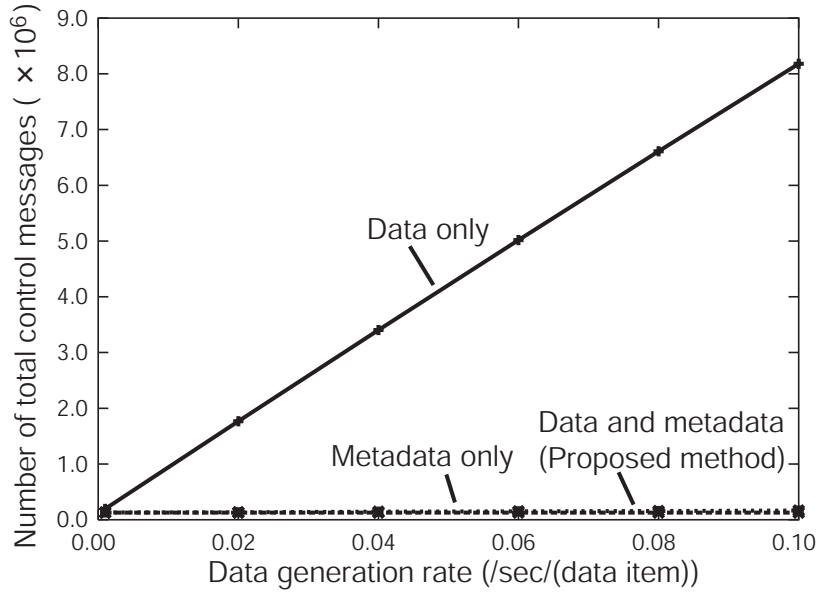


図 5.25 データ発生率と総制御メッセージ送信数の関係

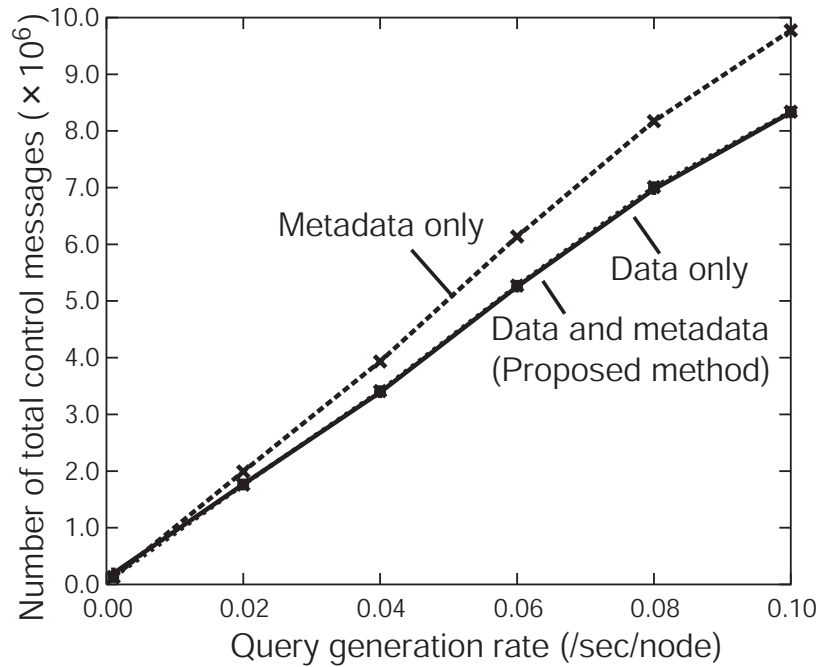


図 5.26 クエリ発生率と総制御メッセージ送信数の関係

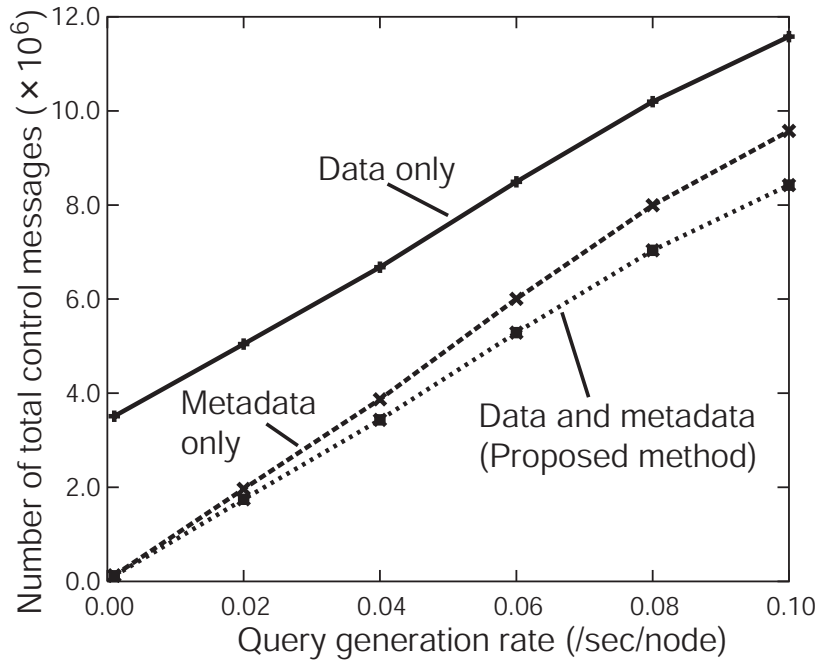


図 5.27 データ発生率とクエリ発生率の分布に偏りがある場合のクエリ発生率と総制御メッセージ送信数の関係

## 5.5 本章のまとめ

本章では、高頻度なデータ更新がある P2P ネットワークにおいて、適応的にコンテンツ配置することで、データ更新やデータ検索の負荷分散やオーバーヘッド削減を行う手法の提案を行った。今後はデータの更新頻度やクエリの検索範囲の空間的な相違だけでなく、時間的な相違の影響についても検討を行う必要があると考えられる。また、地図を用いたユーザインタフェースを伴った実装による動作確認や、センサデータと格納されるノードの両者の地理的位置の関連付けについても、検討の余地があると考えられる。

センサデータと格納されるノードの両者の地理的位置の関連付けに関しては、以下のよう考えられる。通常の DHT ではノードやコンテンツはハッシュ関数により、DHT 空間においてマッピングされるため、そのノードの地理的位置と格納されるデータの地理的位置の相関性はない。ユーザやアプリケーションが自分のいる地理的位置に近い地理的位置を持つデータを格納したいという要求がある場合、ノードの地理的位置と格納されるデータの地理的位置に相関性を持たせるためのアルゴリズムが必要となる。PHT の基本動作では、P01 や P101 などといったラベルは、DHT を利用してノードの地理的位置と無

関係に付与される。しかしながら、ラベルはそれが格納されるデータの地理的位置と関連がある。このラベルを、ノードの地理的位置と関連付けすることによって、ノードの地理的位置と格納されるデータの地理的位置に相関性を持たせることができると考えられる。一例として、あるラベルに対応するノードを求めるときに、ラベル自体をハッシュ関数にかけずに、まず、ラベルのビットをそのまま残し、地理的位置情報のビット列の長さと同じになるように、 $\text{hash}("0" \times (\text{ラベルの最大のビット数} - \text{ラベルのビット数}))$  の値を何ビットか付け足したビット列を求める。このビット列に最も近い地理的位置情報を持つノードがそのラベルに対応するノードとする。そうすると、ラベルとノードの地理的位置が関連付けられ、データはその地理的位置と、ある程度相関のあるラベルに対応するノードに格納されるため、結果的に、ノードの地理的位置と格納されるデータの地理的位置に相関性を持たせることができると考えられる。

## 第6章 結論

---



## 6.1 本論文のまとめ

本論文では、センサデータをネットワークを介して収集、配信する機構において、高効率な自律型制御により、ネットワークリソースを有効に活用する手法を提案し、その有効性をシミュレーションや実装による評価により、有効性を示した。本論文における提案方式により、膨大な実空間情報の高効率な取得が可能となり、今後それを利用したさまざまなサービスが展開されていくことが見込まれる。

3章ではネットワークのトラフィックに応じてタイムスロットを自律分散的に割り当てる低消費電力型メディアアクセス制御についての研究について述べた。提案方式により、消費電力効率が高く、トラフィックが増加してもスループットを維持して通信ができるため、バッテリー交換のコストを抑制することができ、リソースの制約が大きい無線センサネットワークに適用していくことが期待される。

4章では間欠通信に適したアクティブ・スリープ制御の研究について述べた。アプリケーションの通信要求に応じてオンデマンドに同期をとるため、不規則に発生するイベントを検知したり、一部のノードだけが通信を行ったりするようなアプリケーションにおいて効率的に同期をとって通信を行うことができる。

5章ではP2Pワークにおける適応的センサデータの配置法についての研究を行った。データの更新による制御コスト削減や負荷分散が可能となる。そのため、センサデータのような高頻度に更新が行われるデータをP2Pネットワークでスケーラブルに流通させることが可能となる。

## 6.2 今後の課題と展望

実空間の情報を利用したサービスは人間や人間を取り巻く環境に大きな影響を与えると考えられる。そのため、サービスの提供を行う技術について、さらに幅広く高い見地から取り組む必要があると考えられる。今後も、端末の小型化・高性能化・低廉化の傾向は続き、ネットワークに接続される端末や接続可能なネットワークも多種多様になると考えられる。センサネットワークにおいても現在のユーザの住居、オフィスを中心とした実空間情報を利用したサービスだけでなく、屋外の都市や自然環境で受けられるサービス、地球上のどこにいてもグローバルに享受できるサービスという具合に、ユビキタスサービスが空間的な拡がり呈していくと考えられる。それに伴い一般ユーザのネットワークサービスに対する要求も高度化していくと考えられる。高度な情報技術を持たない一般の人でも快適にその恩恵を受けられるようにすることが重要であると考えられる。そのためには、

例えば快適で使いやすいユーザインタフェースの提供などの効果的にネットワークサービスの提供する手段の充実が必要であると考えられる。

また、センサから得られるデータの信頼性や流通におけるセキュリティを確保していくことも重要であると考えられる。特にコンピュータの仮想空間のみならず、物理的な世界にも作用するサービスではセンサデータの信頼性が保障されなければ社会に重大な影響を与えかねず、いかに確保するかも課題となると考えられる。またデータにどのユーザがアクセス可能か可能でないかは、ユーザによっても時と場合によっても千差万別であり、さまざまな状況における個人レベルのセキュリティに対する要求をどのように満たすかも課題となると考えられる。

## 参考文献

- [1] SMART DUST, <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>
- [2] 特集 ”センサがネットにつながれば” , 日経エレクトロニクス 2002-7-15 no.826 , 日経 BP 社 .
- [3] Estrin, D., Govindan, R., Heidemann, J.S., and Kumar, S.: “Next century challenges: Scalable coordination in sensor networks ”, Proc. of ACM MOBICOM’99, pp.263–270, Aug.1999.
- [4] Hill, J. and Culler, D. “A wireless embedded sensor architecture for systemlevel. optimization”, UC Berkeley Technical Report, 2002.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniarm, and E. Cayirci, “A survey on sensor networks ”, IEEE Communications Magazine, vol. 40, no. 4, pp. 102–104, 2002.
- [6] Pister, K. S. J.: “Smart Dust - Hardware limits to wireless sensor networks, ” Proc. of IEEE Distributed Computing Systems, Keynote Address, May 2003.
- [7] UCLA (University of California, Los Angeles) CENS (Center for Embedded Networked Sensing), <http://www.cens.ucla.edu/>
- [8] Gregory T. Huang, ”Casting the Wireless Sensor Net”, MIT Technology Review, July/Aug. 2003.
- [9] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, ”A Survey on sensor Networks”, IEEE Communications Magazine, pp.102-114, Aug 2002.
- [10] S. Meguerdichian, S. Slijepcevic, V. Karayan, M. Potkonjak, ”Localized Algorithms In Wireless Ad-Hoc Networks: Location Discovery And Sensor Exposure ”, Proc of ACM MobiHoc, pp.106-116, Oct. 2001.
- [11] B.Karp and H.T Kung, ”GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, Proc of ACM/IEEE MOBICOM’00, pp.243-254, Aug. 2000.
- [12] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava, ”Coverage Problems in Wireless Ad-hoc Sensor Networks”, Proc of INFOCOM’01, pp.1380-1387, April 2001.

- 
- [13] A. Howard, M.J. Mataric and G.S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks", *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, pp.113-126, Sept. 2002.
- [14] Sylvia Ratnasamy, Brad Karp, Scott Shenker, Deborah Estrin, Ramesh Govindan, Li Yin, Fang Yu, "Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table", *Proc of ACM Mobile Networks and Applications (MONET)*, pp.427-442 Aug. 2003.
- [15] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, Scott Shenke, "GHT: A Geographic Hash Table for Data-Centric Storage", In *First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA) 2002*.
- [16] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Proc of MobiCOM'00*, pp.56-67, Aug. 2000.
- [17] Deborah Estrin, Akbar Sayeed, Mani Srivastava, "Wireless Sensor Networks", *Mobicom 2002 Tutorial*.
- [18] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," *Mobile Computing and Communications Review*, vol.6, no.2, pp.28-36, April 2002.
- [19] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. the Hawaii Conference on System Sciences (HICSS)*, pp.3005-3014, Jan. 2000.
- [20] Phillip B. Gibbons, Brad Karp, Yan Ke, Suman Nath, Srinivasan Seshan , "Iris-Net:An Architecture for a Worldwide Sensor Web", *IEEE Pervasive Computing*, pp.22-33, Oct.-Dec. 2003.
- [21] CENS Research Project, Urban Sensing, [http://research.cens.ucla.edu/projects/2006/Systems/Urban\\_Sensing/](http://research.cens.ucla.edu/projects/2006/Systems/Urban_Sensing/)
- [22] MIT SENSEable City Laboratory, <http://senseable.mit.edu/>

- 
- [23] Meteor project, <http://www.caip.rutgers.edu/TASSL/Projects/Meteor/>
- [24] Live E!, <http://www.live-e.org/>
- [25] Google Maps, <http://maps.google.com/>
- [26] J. Mischke, B. Stiller, "A Methodology for the Design of Distributed Search in P2P Middleware", IEEE Network, Vol.18, pp.30-37, Jan./Feb. 2004.
- [27] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the Physical World with Pervasive Networks," IEEE Pervasive Computing, vol.1, no.1, pp.59-69, Jan.-March 2002.
- [28] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proc. IEEE INFOCOM, vol.21, no.1, pp.1567-1576, June 2002.
- [29] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," ACM/IEEE Transactions on Networking, 12 (3 ), pp.493-506, June 2004.
- [30] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proc. ACM SenSys, pp.171-180, Nov. 2003.
- [31] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," Proc. IEEE WCNC, vol.3, pp.1534-1539, March 2004.
- [32] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," Proc. ACM SenSys, pp.95-107, Nov. 2004.
- [33] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," Proc. ACM SenSys, pp.181-192, Nov. 2003.
- [34] S.S.Kulkarni and M.Arumugan, "TDMA service for sensor networks," Proc. IEEE ICDCSW, pp.604-609, March 2004.
- [35] L. van Hoesel and P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," Proc. INSS, pp.205-208, June 2004.

- 
- [36] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," Proc. IEEE INFOCOM, pp.891-901, vol.2, March 2005.
- [37] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," Proc. ACM MobiSys, pp.125-138, June 2004.
- [38] Y. Li, W. Ye and J. Heidemann, "Energy and Latency Control in Low Duty Cycle MAC Protocols," Proc. IEEE WCNC, pp.676-682, vol.2 March 2005.
- [39] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>
- [40] RF Monolithics Inc., <http://www.rfm.com/>, ASH Transceiver TR1000 Data Sheet.
- [41] MICA2 Mote Datasheet, [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/6020-0042-04\\_A\\_MICA2.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-04_A_MICA2.pdf)
- [42] TinyOS Community Forum, <http://www.tinyos.net/>
- [43] Wei Ye, John Heidemann and Deborah Estrin, "A Flexible and Reliable Radio Communication Stack on Motes," Technical Report ISI-TR-565, USC Information Sciences Institute, Sept. 2002.
- [44] Ajit Warrier, Jeongki Min and Injong Rhee, "Z-MAC: a Hybrid MAC for Wireless Sensor Networks", Proc of ACM SenSys'05, pp.90-101, Nov 2005.
- [45] A. El-Hoiydi and J. -D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks", IEEE Symposium on Computers and Communication (ISCC), pp.244-251, June 2004.
- [46] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The Flooding Time Synchronization Protocol", Proc of ACM SenSys, pp.39-49, Nov. 2004.
- [47] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, I. Stoica, "Looking Up Data in P2P Systems", Communications of the ACM, Vol. 46, No. 2, pp.43-48, Feb 2003,
- .

- 
- [48] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker, "A Scalable Content-Addressable Network", Proc. of ACM SIGCOMM'01 , pp.161-172 , Aug. 2001.
- [49] I.Stoica, R.Morris, D.Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," Proc. of SIGCOMM'01, pp.149-160, Aug. 2001.
- [50] Project JXTA, <http://www.jxta.org/>
- [51] Malena Balazinska, Hari Balakrishnan, David Karger, "INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery", Proc of International Conference on Pervasive Computing (Pervasive) 2002, pp.195-210, Aug. 2002.
- [52] S.Ramabhadran, J.M.Hellerstein, S.Ratnasamy, and S.Shenker, "Prefix Hash Tree: An Indexing Data Structure over Distributed Hash Tables," IRB Technical Report, 2004.
- [53] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, J. Hellerstein, and S. Shenker, "Case Study in Building Layered DHT Applications," Proc. of ACM SIGCOMM'05, pp.97-108, Aug. 2005.
- [54] OpenDHT, <http://opendht.org/>
- [55] p2psim, a simulator for peer-to-peer protocol, <http://pdos.csail.mit.edu/p2psim/>

## 発表文献

### 論文誌

[1] 関根理敏, 竹内彰次郎, 瀬崎薫, “センサネットワークにおける適応的スロット予約による低消費電力メディアアクセス制御プロトコル,” 信学論 B, Vol.J90-B, No.1, pp.25-36.

[2] Masatoshi Sekine, Shojiro Takeuchi, and Kaoru Sezaki “An Energy-efficient Medium Access Control Protocol with Adaptive Slot Reservation for Sensor Networks,” , Electronics & Communications in Japan: Part 1 (to appear) ([1] の英訳論文).

[3] 関根理敏, 瀬崎薫, “P2P ネットワークにおけるセンサデータの適応的収集・管理法 (仮題),” 信学論 B (投稿準備中).

### 国際会議

[4] Masatoshi Sekine, Kaoru Sezaki, “Hierarchical Aggregation of Distributed Data for Sensor Networks,” Proc. of IEEE TENCON 2004 (Chiang Mai, Thailand), Vol.2, pp.545-548, Nov. 2004

[5] Masatoshi Sekine, Shinji Nakamura, and Kaoru Sezaki, “An Energy-Efficient Protocol for Active/Sleep Schedule Synchronization in Wireless Sensor Networks,” Proc. of Asia-Pacific Conference on Communications (APCC) 2006 (Busan, Korea), Aug. 2006.

[6] Masatoshi Sekine, Shojiro Takeuchi, and Kaoru Sezaki, “An Energy-Efficient MAC Protocol with Lightweight and Adaptive Scheduling for Wireless Sensor Networks,” Proc. of IEEE Radio and Wireless Symposium (RWS) 2007 (Long Beach, CA, USA), pp.161-164, Jan. 2007.



---

## 研究会

[7] 関根 理敏, 瀬崎 薫, “センサネットワークにおける属性データの階層的集約法,” 信学技報, NS2003-111, IN2003-77, pp.27-32, 東北大, Sept. 2003.

[8] 関根理敏, 竹内彰次郎, 瀬崎薫, “センサネットワークにおけるパケット衝突抑制型 MAC プロトコル,” 信学技報, NS2004-65, RCS2004-101, pp.73-76, 広国大, July 2004.

[9] 関根理敏, 竹内彰次郎, 瀬崎薫, “低消費電力 MAC プロトコルにおけるスロット予約期間適応的制御,” 信学技報, NS2005-2, pp.7-10, 機械振興会館, May 2005.

[10] Xinfu Wei, Masatoshi Sekine, and Kaoru Sezaki, “A Peer-to-Peer System for Internet-Scale Location-Based Service,” 信学技報, IN2005-30, pp.7-12, 北大, July 2005.

[11] 中村新治, 関根理敏, 瀬崎薫, “センサネットワークにおける間欠通信を考慮した同期制御に関する一検討,” 信学技報, NS2005-211, pp.213-216, リザンシーパークホテル 谷茶ベイ (沖縄), March 2006.

[12] 関根理敏, 瀬崎薫, “P2P ネットワークにおける高頻度なデータ更新を考慮した適応的コンテンツ配置法,” 信学技法, 沖縄コンベンションセンター, March 2007 (発表予定).

## 全国大会

[13] 関根理敏, 瀬崎薫, “センサネットワークにおける階層的集約法,” 信学ソ大 B-7-71, 新潟大, Sept. 2003.

[14] 関根理敏, 竹内彰次郎, 瀬崎薫, “センサネットワークにおけるパケット衝突抑制アクセス制御手法,” 信学総大, B-7-49, 東工大, March 2004.

[15] 関根理敏, 竹内彰次郎, 瀬崎薫, “MAC 層におけるスロット割り当て制御期間の適応的制御に関する一検討,” 信学総大, A-21-18, 阪大, March 2005.

[16] 関根理敏, 竹内彰次郎, 瀬崎薫, “低消費電力スケジューリングベース MAC プロトコルの実装,” 信学ソ大, A-21-6, 北大, Sept. 2005 .

[17] 関根理敏, 瀬崎薫, “P2P ネットワークにおけるセンサデータの動的複製配置手法の検討,” 信学ソ大, B-7-38, 金大, Sept. 2006 .

[18] 関根理敏, 瀬崎薫, “P2P ネットワークにおける高頻度なデータ更新を考慮した負荷分散法,” 信学総大, B-7-47, 名城大, March 2007 (発表予定) .

## 謝辞

博士論文の研究を進めていく過程で、指導教員である瀬崎薫助教授には研究における有益な知識、研究に対する心構えなどをご教授くださり、また、充実した研究環境を与えていただきました。感謝いたします。

博士論文の研究内容について、主に電子情報学博士特別研究、博士予備審査、また博士本審査において、青山友紀教授、浅野正一郎教授、浅見徹教授、江崎浩教授、森川博之教授、また松浦幹太助教授には有益なご助言をいただきました。感謝いたします。

早稲田大学大学院の博士課程に在籍された竹内彰次郎さんには、特に第3章の適応的スロット予約による低消費電力メディアアクセス制御の研究において、多くの有益な助言と励ましの言葉をいただきました。また、日立電子サービスの研究生であった中村新治さんには、特に第4章の間欠通信適応型同期制御の研究において、実装や計算機シミュレーションによる性能評価において協力していただきました。さらに、研究生のマーティンス・マルセロさんには英語論文の作成において、多くの有益な助言をいただきました。感謝いたします。

瀬崎研究室において、快適な研究環境の整備にご尽力くださった助手の小松邦紀さん、秘書の松本夏穂さん、小野裕子さんに感謝いたします。また、研究室ではさまざまな活動を共にして、充実した生活を送ることができました。木實新一さん、クレイセル・ベルネルさん、引地謙治さん、黄楽平さん、山崎浩輔さん、テープウィロー・ジャナポン・ニワットさん、任明、魏新法さん、寺田真介さん、角田忠信さん、岡野諭さん、田中隆浩さん、セーンラッタナシャイクン・オラナットさん、一松隆平さん、また島田健太さんに感謝いたします。

最後に、自分をさまざまな面において支えてくれた家族、親戚、また友人に感謝します。