

Title	近距離無線通信規格ZigBeeにおけるノードグループ ング機構に関する研究
Author(s)	磯貝, 彰則
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3573
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

近距離無線通信規格 ZigBee における
ノードグルーピング機構に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

磯貝 彰則

2007 年 3 月

修 士 論 文

近距離無線通信規格 ZigBee における
ノードグループ핑機構に関する研究

指導教官 丹康雄 助教授

審査委員主査 丹康雄 助教授
審査委員 篠田陽一 教授
審査委員 日比野靖 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

510009 磯貝 彰則

提出年月: 2007 年 2 月

概要

本稿では、センサ・ネットワークと呼ばれる分野で活躍が期待されている近距離無線通信規格 ZigBee [1] において、任意のノードを論理的にまとめるノードグルーピングを実現し、センサ・ネットワーク運用上の課題となっているデータ転送や消費電力といった要点の効率化を行う方式を提案する。提案システムにより、家庭内や工場、またはビルディングなどで ZigBee の利用を想定した際、各ノードの長期にわたる安定した動作と共に、グルーピングによる複数の特定ノードへの操作・制御が可能となる。本研究では、施設制御や環境モニタリングを行ういくつかの性質の異なるアプリケーションシステムを考案した後、それぞれのアプリケーション特性を考慮したグルーピングを適用し、データ転送の効率化や省電力化などの観点で提案方式の検証、評価を行う。

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	1
1.3	本論文の構成	2
第2章	ZigBee	3
2.1	ZigBee の概要	3
2.2	ZigBee の通信技術	3
2.2.1	ZigBee のデバイス・タイプ	3
2.2.2	ZigBee プロトコルスタック	5
2.2.3	物理層	6
2.2.4	MAC 層	7
2.2.5	ネットワーク層	9
2.2.6	アプリケーション・サポート・サブ層	14
第3章	グルーピング実現レイヤの検討	15
3.1	MAC 層	15
3.2	ネットワーク層	15
3.3	アプリケーション・サポート・サブ層	16
3.4	検討	16
第4章	グルーピング方式の提案	17
4.1	グループ ID の定義	17
4.2	各トポロジ毎のグルーピング方式	20
4.2.1	スター型トポロジ	20
4.2.2	クラスタ・ツリー型トポロジ	22
4.2.3	メッシュ型トポロジ	29
4.3	拡張方式の問題点	41
4.4	提案方式のまとめ	47
第5章	提案システムの実装	50
5.1	実装の検討	50

5.2	性能の評価	52
第 6 章	提案システムの評価	55
6.1	各トポロジにおけるアプリケーションの考案・評価	57
6.1.1	スター型トポロジ	57
6.1.2	クラスタ・ツリー型トポロジ	61
6.1.3	メッシュ型トポロジ	66
6.2	アプリケーションからトポロジ・グルーピングの検討・考案	74
6.2.1	FA アプリケーションの概要	74
6.2.2	FA の構成・トポロジ・グルーピング考案	75
6.2.3	BA アプリケーションの概要	79
6.2.4	BA の構成・トポロジ・グルーピング考案	80
6.3	メッシュ型トポロジの経路探索待ち時間の検討	84
6.4	評価のまとめ	87
第 7 章	まとめと今後の課題	88
第 8 章	謝辞	89
付 録 A	MSSTATE_LRWPAN	92
A.1	情報の定義	92
A.2	GroupID 情報追加	93
A.3	送受信時の GroupID 情報付与・解析	93
A.4	有限状態機械	95
付 録 B	ZigBee Document 053474r13	96
B.1	概要	96
B.2	ネットワーク層ヘッダ構成の変更	96
B.3	ソース・ルーティング方式の実現	98
B.4	マルチキャストの実現	98
B.5	比較	101
B.6	考察	102

目次

2.1	ZigBee プロトコルスタック	5
2.2	Beacon フレームの構造	7
2.3	子から親へのデータ転送モデル	8
2.4	親から子へのデータ転送モデル	8
2.5	子から親へのデータ転送モデル	8
2.6	親から子へのデータ転送モデル	8
2.7	ZigBee におけるネットワーク・トポロジの種類	9
2.8	ZigBee におけるネットワーク・アドレッシング	11
4.1	ネットワーク層フレーム・フォーマット	18
4.2	Frame Control フィールドの詳細	18
4.3	スター型トポロジ	20
4.4	クラスタ・ツリー型トポロジ	22
4.5	コマンドフレーム・フォーマット	23
4.6	グループ ID 追加コマンド・フレームフォーマット	24
4.7	グループ ID 削除コマンド・フレームフォーマット	24
4.8	クラスタ・ツリー・ネットワーク例	27
4.9	クラスタ・ツリーにおけるグルーピング・テーブル完成図	28
4.10	メッシュ型トポロジ	29
4.11	Route Request ペイロード	32
4.12	Route Reply ペイロード	32
4.13	Route Error ペイロード	32
4.14	メッシュ型ネットワーク例	34
4.15	通常 AODV のルーティング・テーブル完成図	35
4.16	Multi Route Request ペイロード	37
4.17	Multi Route Reply ペイロード	37
4.18	Multi Route Reply の中継	38
4.19	送信元ノードと宛先グループ・ノードの ID の問題	40
4.20	メッシュ型ネットワーク構成例	45
4.21	メッシュ型ネットワークのグルーピング・テーブル完成図	46
5.1	実装した MSSTATE LRWPAN システム	54

6.1	室内のスター型アプリケーションのイメージ図	57
6.2	スター型トポロジの評価用グループ	58
6.3	スター型トポロジにおける各ノード必要通信回数	59
6.4	スター型トポロジにおける各ノードデータ送信量	59
6.5	屋内のクラスタ・ツリー型アプリケーションのイメージ図	61
6.6	クラスタ・ツリー型トポロジの評価用グループ	62
6.7	クラスタ・ツリー型トポロジにおける各ノード必要通信回数	63
6.8	クラスタ・ツリー型トポロジにおける各ノードデータ送信量	63
6.9	クラスタ・ツリー型トポロジにおけるノードのテーブル・サイズ	65
6.10	ビルにおけるメッシュ型アプリケーションのイメージ図	66
6.11	メッシュ型トポロジの評価用グループ	67
6.12	経路探索が完了した評価用メッシュ型トポロジ	68
6.13	メッシュ型トポロジ (パターン A) の各ノード必要通信回数	69
6.14	メッシュ型トポロジ (パターン B) の各ノード必要通信回数	69
6.15	メッシュ型トポロジ (ユニキャスト) の各ノード必要通信回数	70
6.16	メッシュ型トポロジ (パターン A) の各ノードデータ送信量	71
6.17	メッシュ型トポロジ (パターン B) の各ノードデータ送信量	71
6.18	メッシュ型トポロジ (ユニキャスト) の各ノードデータ送信量	72
6.19	メッシュ型トポロジにおけるノードのテーブル・サイズ	73
6.20	FA を想定する工場の構造	74
6.21	スター型トポロジを想定した FA アプリケーション	76
6.22	クラスタ・ツリー型トポロジを想定した FA アプリケーション	77
6.23	メッシュ型トポロジを想定した FA アプリケーション	78
6.24	BA を想定するビルの構造	79
6.25	スター型トポロジを想定した BA アプリケーション	81
6.26	クラスタ・ツリー型トポロジを想定した BA アプリケーション	82
6.27	メッシュ型トポロジを想定した BA アプリケーション	83
6.28	拡張経路探索時のノード処理時間と送信待ち時間の関係	85
6.29	拡張経路探索時のリトライと送信待ち時間の関係	86
B.1	新しいネットワーク層フレーム・フォーマット	97
B.2	新しい Frame Control フィールドの詳細	97
B.3	ソース・ルーティング・リストの詳細	98
B.4	MultiCast Control フィールドの詳細	98
B.5	MultiCast mode フィールドの値	99
B.6	グループ・テーブル・エントリの構成	100

表 目 次

2.1	論理デバイス・タイプの比較表	4
2.2	物理デバイス・タイプの比較表	4
2.3	物理層の周波数帯とパラメータ一覧	6
2.4	隣接テーブル・フォーマット	12
4.1	隣接テーブルに追加するフィールド	21
4.2	グルーピング・テーブルのフォーマット	23
4.3	ネットワーク層コマンド ID の種類	23
4.4	追加コマンド ID	25
4.5	ZigBee ルータ (0x0002) 保持の隣接テーブル・エントリ	28
4.6	ルーティング・テーブル・フォーマット	33
4.7	ルート・ステータス	33
4.8	ルート発見テーブル・フォーマット	33
4.9	グループ拡張ルーティング・テーブル	37
4.10	ZigBee ルータ (0x0013) 保持の隣接テーブルの例	46
5.1	MSSTATE LRWPAN 動作プラットフォーム	51
5.2	MSSTATE LRWPAN 実装環境	51
5.3	ノート型パソコン実験環境	52
5.4	各 PC におけるレイテンシの比較	53
B.1	提案方式と新規格の比較表	101

第1章 はじめに

1.1 研究の背景

近年，通信技術や半導体技術の目覚ましい発達により，センサ・ネットワークが非常に注目されている．その中でも特に，無線で通信を行うワイヤレス・センサ・ネットワークは，家庭内の照明や空調といった家電機器の制御を始め，機器の監視やセンシングを容易に実現できるとして，ビルディングや工場のオートメーション化を実現するものとしても期待されている．しかし，ワイヤレス・センサ・ネットワークの現状は幾多の課題を抱えている．その一つの例が消費電力の問題である．センサ・ノードは，人の手が入りにくい場所に設置されることが多いため，電源の確保や電池の交換が難しいといわれる．そこで近年の研究では，省電力化を実現するために，アンテナの指向性やノード間の同期の方法を始め，様々な視点から研究が行われている．

1.2 研究の目的

本研究では，ワイヤレス・センサ・ネットワークとして非常に注目されている規格である ZigBee において，先に挙げた消費電力のみならず，膨大なノード数が存在する時のデータ転送の効率化やデータ処理の即時性という部分に着目し，任意のノード群を論理的にまとめる事が可能となるノードグルーピング方式を提案する．提案するノードグルーピング方式を利用して ZigBee ネットワークに存在するノードを論理的にグループ化することにより，グルーピングを行った特定のノード群に対し，一括でのデータ配信を実現できる．これにより，データの送信元ノードや中継ノードは送信回数の削減が実現でき，消費電力やデータ転送の効率化が達成できると考えられる．またアプリケーション構成の観点では，複数の任意ノードに対し柔軟にグループを割り当てることで，アプリケーション動作の幅を広げることが可能となる．グループ割り当てでは，機器の種類といった固定的なパターンのみでなく，一定の範囲内に存在する機器や協調動作を行う各機器を同一のグループとして扱うといった事柄にも利用でき，状況や用途に応じてグループを指定することで従来よりも柔軟な操作性を提供できる．本研究ではグルーピング方式の考案に際し，各要点を改善するためにどのレイヤで実現すべきかを検討し，ZigBee におけるグルーピング方式を提案する．

1.3 本論文の構成

本論文は以下の構成になっている。

- 第1章
研究の背景と目的，本論文を通しての全体の流れの説明を行う。
- 第2章
本研究の研究対象である ZigBee 規格に関して，概要や通信技術についてまとめる。
- 第3章
提案グルーピング方式を実現するレイヤの検討を行う。
各レイヤを比較・検討し，最適なレイヤを選択する。
- 第4章
前章で行った検討を元に，グルーピング方式の提案を行う。
ZigBee の各トポロジにおいてデータを配信する仕組みや処理の流れを明確にし，説明を行う。
- 第5章
提案システムの実装について，説明を行う。
- 第6章
提案システムの有効性に関する評価を行う。
- 第7章
提案システムについて考察を行い，今後の展望や課題について述べる。
- 第8章
本論文の総括を行う。

第2章 ZigBee

本章では、ZigBee の概要と通信技術についてまとめる。特に本稿で扱うグルーピング方式の検討・提案に大きく関連する部分を重点的に取り扱う。

2.1 ZigBee の概要

ZigBee は、無線通信を利用して短距離のデータ通信を実現する技術である。ZigBee は現在、ZigBee Alliance [1] と呼ばれる団体によって、標準化が進められている。ZigBee の特徴として、低位レイヤに IEEE802.15.4 [2] を採用したこと、低消費電力で長期間運用が可能なこと、低コストで製造が可能であること、クラスタ・ツリーやメッシュ型トポロジではマルチホップに渡る通信が可能であることなどが挙げられる。

2.2 ZigBee の通信技術

2.2.1 ZigBee のデバイス・タイプ

ZigBee ノードのデバイス・タイプには、大きく分けて論理デバイス・タイプと物理デバイス・タイプが存在する。この2つのデバイス・タイプには、更に細かい分類が存在する。その詳細¹について以下に述べる。

- 論理デバイス・タイプ (3 種類)
 - ZigBee コーディネータ
 - ZigBee ルータ
 - ZigBee エンド・デバイス
- 物理デバイス・タイプ (2 種類)
 - Full Function Device (FFD)
 - Reduced Function Device (RFD)

¹論理デバイス・タイプには ZigBee で利用される名称を記述した。

論理デバイス・タイプは、ZigBee ネットワーク内に存在するノードの役割を示したものであり、物理デバイス・タイプは、各ノードがどの程度の機能を持っているかを分類するものである。ZigBee コーディネータは論理デバイス・タイプとして存在し、PAN 内にたった一つしか存在してはならない。通常は PC 等と接続され、データの送信元あるいは収集を行う役割を果たすことが多い。ZigBee ルータは、文字通りルーティング機能を持ったノードであり、マルチホップの通信の際にルーティングを行うことができる。ZigBee エンド・デバイスは、ZigBee コーディネータと ZigBee ルータ以外のノードであり、通常はネットワークの末端に位置する。つづいて物理デバイス・タイプの説明を行う。物理デバイス・タイプに存在する FFD と RFD の一番の違いは、ルーティング能力を持つかどうかである。FFD がルーティング能力を持つのにに対し、RFD はルーティング能力を持たない。これは、各ノードにルーティングテーブルを保持するだけの RAM が搭載されているかどうかに関わり、製造コストにも大きな関係が現れる。以上の事柄を以下の表 2.1, 2.2 にまとめる。

名称	ZigBee コーディネータ	ZigBee ルータ	ZigBee エンド・デバイス
ルータ機能	有り	有り	無し
PAN 内の存在数	1	複数	複数
消費電力	大	大	小
処理量	大	中	小

表 2.1: 論理デバイス・タイプの比較表

名称	FFD	RFD
ルータ機能	有り	無し
ZigBee コーディネータ	可能	不可能
ZigBee ルータ	可能	不可能
ZigBee エンド・デバイス	可能	可能
製造コスト	高い	低い

表 2.2: 物理デバイス・タイプの比較表

2.2.2 ZigBee プロトコルスタック

ZigBee のプロトコルスタックは以下の図 2.1 のようになる。ZigBee は物理層と MAC 層に IEEE802.15.4 を採用し、ネットワーク層とアプリケーションのためのフレームワークを提供するものである。ここでは、プロトコルスタックの構成と各レイヤ毎の仕組みについて、下位のレイヤから順に説明を行う。

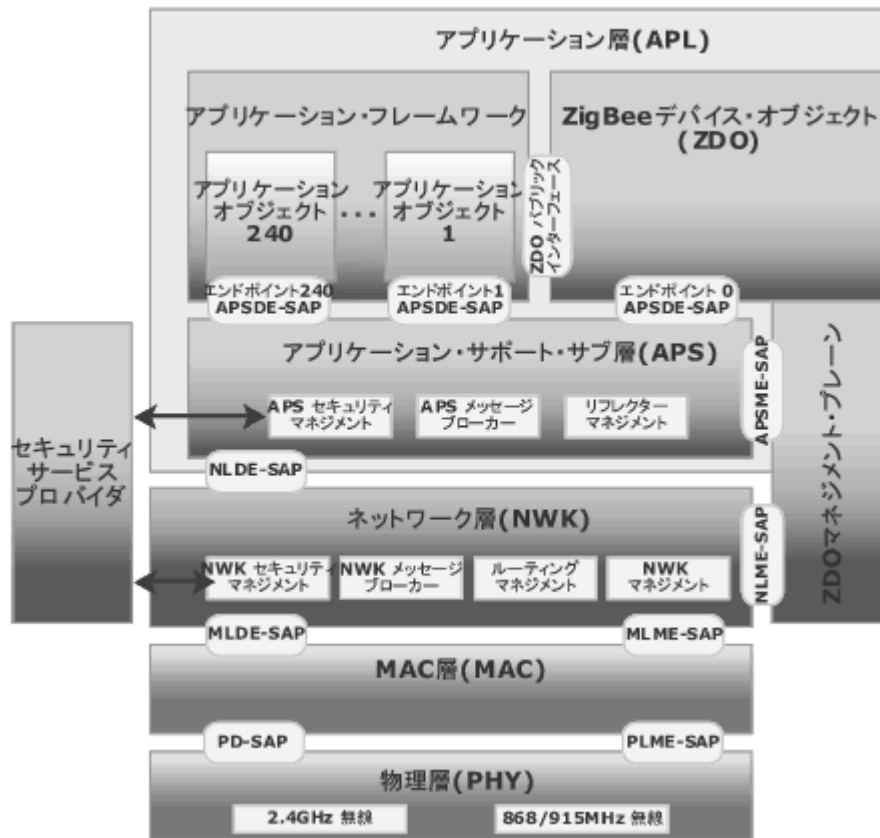


図 2.1: ZigBee プロトコルスタック

2.2.3 物理層

ZigBee の物理層の周波数帯として、868MHz 帯、915MHz 帯、2.4GHz 帯があり、通信方式として DSSS² が採用されている。日本国内では現在、電波法の規制により 2.4GHz 周波数帯においてのみ通信が許可されている。通信に利用される周波数帯、チップ・レート、変調方式、ビット・レート、シンボル等について、以下の表 2.3 にまとめる。

PHY (MHz)	周波数帯 (MHz)	拡散パラメータ		データパラメータ		
		chip rate	変調方式	bit rate	symbol rate	symbols
868/915	868-868.6	300 kchip/s	BPSK	20 kbps	20 ksymbol/s	2 進
	902-928	600 kchip/s	BPSK	40 kbps	40 ksymbol/s	2 進
2450	2400-2483.5	2000 kchip/s	O-QPSK	250 kbps	62.5 ksymbol/s	16 進

表 2.3: 物理層の周波数帯とパラメータ一覧

日本国内で使用が可能な 2.4GHz 帯には 16 のチャンネルが存在する。その 16 のチャンネル (11 ~ 26) の中心周波数 $F(n)$ は以下の式 2.1 から算出することが出来る。

$$F(n) = 2350 + 5n \quad (n = 11 \sim 26) \quad (2.1)$$

これらの周波数帯において、ZigBee は干渉の少ないチャンネルを選択し通信を行うことができる。実際に利用する ZigBee の通信においては、国内で ISM 帯を利用する通信との干渉を考慮する必要がある。その際のリンクの通信品質を決定する値として、LQI³ を利用する。LQI の詳細な計算方法は規定されていないが、LQI を利用することで上位層での最適なチャンネル選択や、経路選択などのサービスを利用することができる。

²直接スペクトラム拡散方式

³Link Quality Indication

2.2.4 MAC 層

MAC 層では、リンク間の通信方法について説明を行う。IEEE802.15.4 の特徴として Beacon フレームの利用が挙げられる。Beacon フレームは、定期的に特定のノードから送信され、受信したノードがそのタイミングに合わせて動作を行うというものである。Beacon フレームは送信のタイミングを計るための同期を取るという目的の他に、IEEE802.15.4 の大きな特徴である間欠動作を可能にすることができる。Beacon フレーム間隔は 2 つの期間に分けられ、データの送信などを行うアクティブ期間と、行わないインアクティブ期間に分けられる。アクティブ期間のフレームをスーパー・フレームと呼び、時分割されたスロットにおいて CSMA/CA を利用してデータの送受信を行う。スーパー・フレーム内にはさらに 2 つの期間があり、CAP⁴ と CFP⁵ に分けられる。CAP が上で述べたような CSMA/CA で送信するのに対し、CFP は予め割り当てられたノードが競合無しにデータを送信する事が出来る。以上の事柄を以下の図 2.2 にまとめて示す。

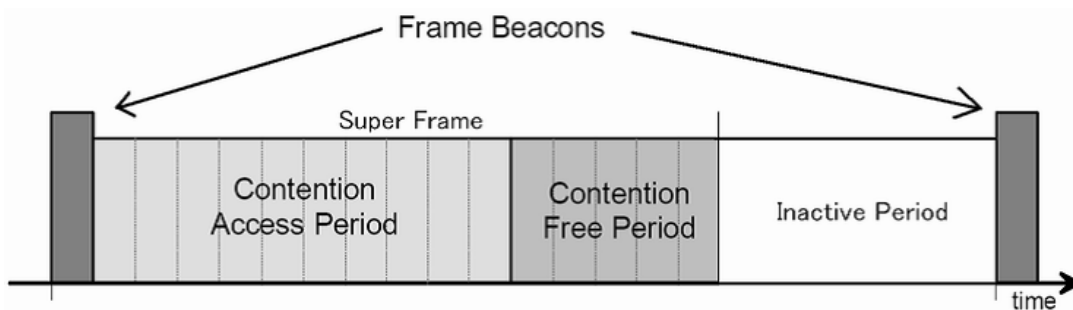


図 2.2: Beacon フレームの構造

⁴Contention Access Period

⁵Contention Free Period

続いて、データの転送モデルについて説明を行う。データの転送モデルとして、Beacon フレームを利用する Beacon モードと Beacon を利用しない非 Beacon モードについて取り扱う。また、各ノードについて Beacon 発行ノード (親ノード) と Beacon 受信ノード (子ノード) の双方向の転送モデルについても取り扱う。

子から親へのプッシュ型データ転送において、非 Beacon モードの場合はいつ子から送信が行われるか予測不能なため、親は間欠動作を行う事ができない。

反対に親から子に対しデータ転送を行う場合は、常に子側から Data Request を発行しデータの要求を行う。Beacon モードの際は、子宛のデータがあるという事を Beacon 情報に含ませ通信を行う。そのため、非 Beacon モードの場合は、定期的の子から親へと自分宛のデータが存在しないかどうかを尋ねる必要がある。これらをまとめたものを以下の図 2.3 から 2.6 に示す。

- Beacon モード

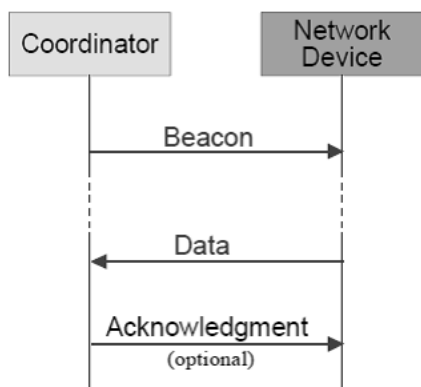


図 2.3: 子から親へのデータ転送モデル

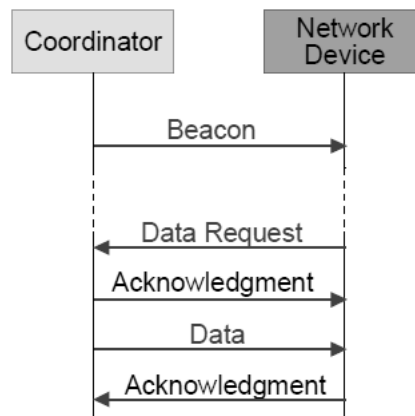


図 2.4: 親から子へのデータ転送モデル

- 非 Beacon モード

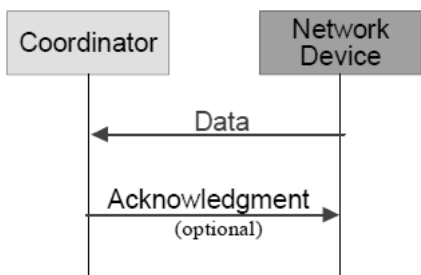


図 2.5: 子から親へのデータ転送モデル

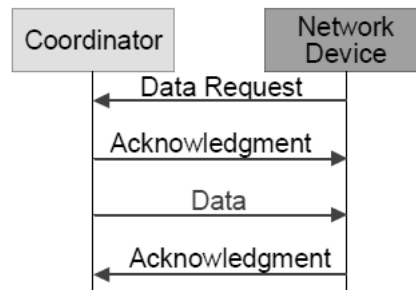


図 2.6: 親から子へのデータ転送モデル

2.2.5 ネットワーク層

ZigBee のネットワーク層では，図 2.7 の 3 種のトポロジをサポートしている．

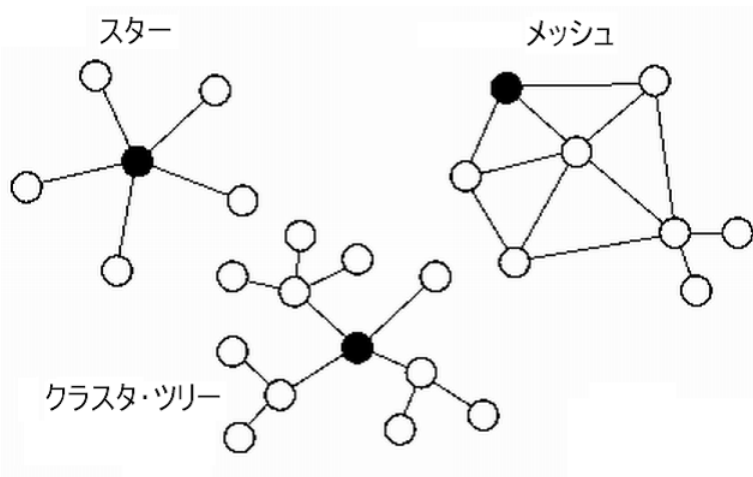


図 2.7: ZigBee におけるネットワーク・トポロジの種類

以下に，各トポロジの特徴について説明する．

- スター型トポロジ
一番シンプルなネットワークトポロジであり，中心のノードが ZigBee コーディネータである．規格上最大 65535 ノードまで接続が可能で，通信はすべて ZigBee コーディネータを経由して行われる．最もシンプルな構成ゆえに，間欠動作を行い省電力化を最も効率よく実現できるが，通信距離は長くはない．
- クラスタツリー型トポロジ
後述の ZigBee 特有のアドレス付けにより，ルーティング・テーブルが必要なくともマルチホップでの通信が行えるトポロジである．マルチホップの通信ゆえに比較的大規模なネットワークを構成しやすい．しかし，ツリーという構成上，ツリーに沿った最適な経路ではない経路でルーティングが行われる可能性がある．また，マルチホップという特性上，中継ノードは間欠動作を行い難いという特徴もある．
- メッシュ型トポロジ
メッシュ型トポロジでは，一つの宛先に対し冗長な経路を確保する事が出来るため，通信の信頼性を向上する事が出来る．しかし，中継を行う各ノードがルーティング機能を持たなければならず，実装のコストが他の 2 つのトポロジに比べて高い．またクラスタツリー同様マルチホップのための間欠動作を行い難く，ルーティング・テーブル保持のためのリソースの要求や平均的に消費電力が高いという特徴がある．

ZigBee のネットワーク層における，ネットワーク・アドレス付けの方法を説明する．ZigBee では，新たに PAN に参加してきたノードに対し，アドレスを割り振る．その際に，ネットワーク層の Information Base のパラメータである `nwkUseTreeAddrAlloc` が TRUE に設定されている場合，PAN 内のツリーの位置に対応したアドレス付けが行われる．以下にその手順を示す．

- ネットワークを構成するのパラメータを設定
ネットワーク・アドレス付けに際し，関連するパラメータは以下の 3 つであり，このパラメータを元にアドレスを算出する．
 1. 最大子ノード数 C_m
 2. 最大ルータ子ノード数 R_m
 3. 最大階層数 L_m
- 与えられたパラメータより計算
計算式は，以下の式 2.2 の様になる．この式において， C_m ， R_m ， L_m は上記のパラメータ， d は算出する階層の深さを示す． $C_{skip}(d)$ は階層 $d+1$ における ZigBee ルータ・ノードに割り当てるアドレスをスキップする間隔となる．

$$C_{skip}(d) = \begin{cases} 1 + C_m(L_m - d - 1), & \text{if } R_m = 1 \\ \frac{1 + C_m - R_m - C_m R_m^{L_m - d - 1}}{1 - R_m}, & \text{otherwise} \end{cases} \quad (2.2)$$

具体例として、 $(C_m=4, R_m=4, L_m=3)$ の例を考えてみる。
 式 2.2 により、 $C_{\text{skip}}(0)=21, C_{\text{skip}}(1)=5, C_{\text{skip}}(2)=1, C_{\text{skip}}(3)=0$ となる。これを図で示すと、下図 2.8 のようになる。

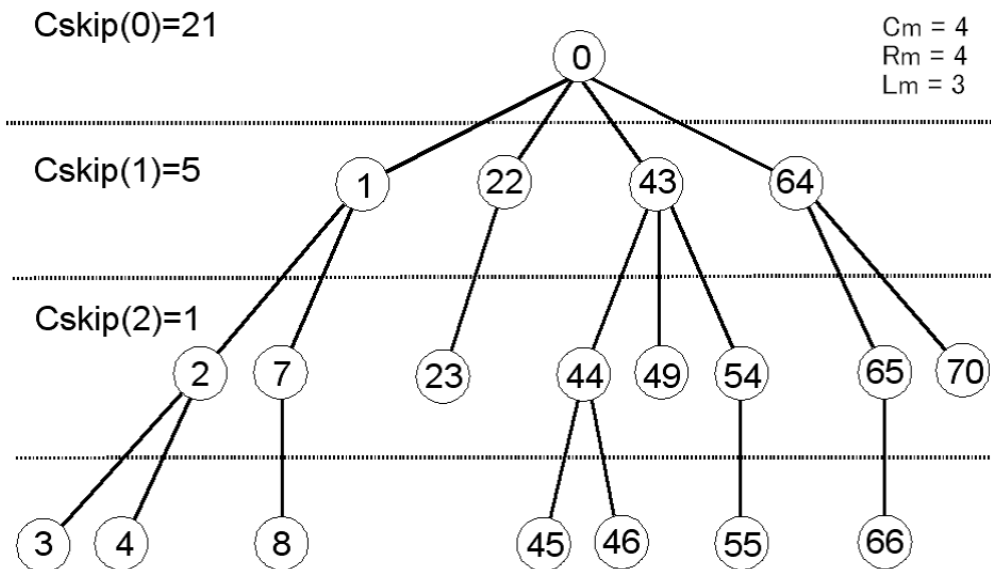


図 2.8: ZigBee におけるネットワーク・アドレッシング

ZigBee では、図 2.8 にのようにしてアドレスと位置を関連付け、ルーティングすべき方向を割り出す。その際に各ノードは、自分と接続関係を持つノード等の情報を持つ必要がある。それを行うのが、ZigBee の全てのノードが保持する必要がある隣接テーブル (Neighbor Table) の存在である。Neighbor Table は、隣接ノードのネットワーク・アドレス、IEEE 拡張アドレス (MAC アドレス) 等の情報を保持しており、データ転送や PAN 構成サービスの実行等に利用される。以下の表 2.4 に隣接テーブルの構成を示す。

フィールド	値	説明
PAN ID	0x0000-0x3FFF	隣接ノードの PAN ID
Extended Address	64 bit	隣接 IEEE 拡張アドレス
Network Address	0x0000-0xFFFF	隣接ネットワークアドレス
Device Type	0x00-0x02	論理デバイス・タイプ
RxOnWhenIdle	TRUE/FALSE	待機状態における受信フラグ
Relationship	0x00-0x03	接続ノードとの関係
Depth	0x00-最大階層数	隣接ノードの階層数
Beacon order	0x00-0x03	隣接ノードのビーコンオーダ
Permit joining	TRUE/FALSE	隣接ノードへの参加の許可
Transmit Failure	0x00-0xFF	隣接ノードへの送信失敗回数
Potential parent	0x00-0x01	隣接ノードを親にする可能性
LQI	0x00-0xFF	リンク通信品質
Logical channel	11 ~ 26	利用するチャネル
Incoming beacon timestamp	4 byte	ビーコン受信タイムスタンプ
Beacon transmission time offset	4 byte	隣接/親ビーコンの時間差

表 2.4: 隣接テーブル・フォーマット

表 2.4 で示したフィールドは、全て実装要求があるわけではなく一部はオプションとして実装される。また ZigBee Alliance が定める ZigBee Specification ver1.0[3] では、想定するアプリケーション毎に必要な隣接テーブルのエントリ数が定められている。詳細な値として、ホーム・コントロールには 29 レコード以上、ビルディング・オートメーションは 30 レコード以上、プラント制御には 87 レコード以上と、利用する用途により大きく実装要求が異なる。

前述のアドレッシング 2.8 を利用したルーティングの他に，メッシュ型ネットワークでは，状況に応じて動的に経路を変更するルーティング・アルゴリズムが用いられる．ルーティング・アルゴリズムとして，AODV⁶方式が利用される．AODV はリアクティブ型の方式であり，RFC3561[6] で定義されている．ZigBee では，センサノードで処理を行うため，従来の AODV 方式をより簡略化して採用している．簡略化を行った AODV の方式の詳細については，提案方式の説明を行う第 4 章で行う．

⁶Ad hoc On-Demand Distance Vector

2.2.6 アプリケーション・サポート・サブ層

アプリケーション・サポート・サブ層は、ZigBee ネットワークに存在するデバイスの接続等を行う際に、互いのノードの情報を伝えるための枠組みや、ZigBee Alliance や各ベンダーが作成したプロファイル (Profile) を利用してアプリケーションのフレームワークを提供する。ここではアプリケーション・サポート・サブ層での大きな特徴であるプロファイル、エンド・ポイント、バインディングについて説明を行う。

- プロファイル

プロファイルとは、同じアプリケーション・オブジェクトに採用される全てのクラスタの集合である。クラスタとは、同じ方向へと通信するパラメータの集合であり、入力クラスタと出力クラスタが存在する。プロファイルは、上述の通り、ZigBee Alliance により標準化されたもの、各ベンダーが作成し公開を行うもの、そしてメーカー固有の非公開のプロファイルに大別できる。各々のプロファイルはプロファイルIDを持ち、IDを識別する事で該当オブジェクト同士が通信を行う事を可能としている。現在、ZigBee Alliance により公開されているものだけでも Dimmer Remote Control、Dimming Load Controller、Light Sensor Monochromatic、Occupancy Sensor、Switching Load Controller などが存在する。

- エンド・ポイント

ZigBee センサノードにおいて、一つのノードに対し一つのアプリケーション・オブジェクトしか存在できないのでは効率が悪い。そこで、エンド・ポイントと呼ばれる TCP/IP のポートに該当する番号が存在する。最大 240 個のアプリケーション用のエンド・ポイントが利用できる。一つのエンドポイントに対し、一つのアプリケーションオブジェクトしか存在することはできないが、複数のエンド・ポイントが存在するために、一つのノード上で同じプロファイルのアプリケーション・オブジェクトは複数動作させる事が出来る。

- バインディング

バインディングは、操作するものと操作されるものの関係に対応付ける。具体的には、とあるセンサノードの照明スイッチボタンを押した際に、どの照明が点灯するかという関連付けの操作を行う事ができる。そのためには、プロファイル、エンドポイント等を利用する事により実現する事が可能となるが、操作するものと操作されるものの関連情報を保持する必要があるため、バインディング・テーブルと呼ばれる送信元アドレス、エンドポイント、クラスタIDと送信先アドレス、エンド・ポイントを結びつけたものを保存する必要がある。

アプリケーション・サポート・サブ層では、これらの機能を利用しアプリケーションのフレームワークを提供する。

第3章 グループリング実現レイヤの検討

本章では、グループリング方式を実現するレイヤの検討を行う。グループリングを実現する可能性があるレイヤとして、MAC 層、ネットワーク層、アプリケーション・サポート・サブ層が存在する。以下では、各層で実現する際の比較を行った。

3.1 MAC 層

まず、MAC 層でのグループリング方式実現について考える。MAC 層で実現する場合、センサノード間でのデータの送受信のポイントとなるのは Beacon フレームである。Beacon モードでの親から子へデータ転送を行う場合、Beacon フレーム内の Pending フィールドを見て自分宛のデータが存在するかどうかを把握する。そして自分宛のデータがあれば、子ノードが Data Request を親に向かって送信する事で、親から子へとデータが送信される。複数のノードをグループ化して扱い、まとめてデータの転送をする場合を考えると、Beacon フレームの Pending フィールドの部分に変更を加え特定のグループ宛のデータ転送を実現するという方法が一つ考えられる。しかし MAC 層では、リンク間の通信を行うためにスター型のようなトポロジでは有効であるが、ツリー型やメッシュ型のようなマルチホップの際には複雑な処理が必要となる。また、有線における特定ポート方向にデータの送信を行う VLAN と違い、無線においての電波到達範囲は曖昧なものであるため、どのノードが宛先までの中継を行うかの判断が難しいと考えられる。これらにより、グループリングを実現することは困難であると考えられる。

3.2 ネットワーク層

続いて、ネットワーク層でのグループリング方式について考える。ネットワーク層で、特定のグループに対してデータの配信を行うには、特定グループに存在するノード全てに対する経路情報をデータ中継ノードが知る必要が在る。一般的にこれは、ルーティング・アルゴリズムに依存する。ZigBee の場合、前述の図 2.8 や AODV 等を採用しているため、マルチホップ通信を行う際のルーティング経路は確保できる。そのアルゴリズムを拡張し、特定のグループ宛の到達性を確保出来れば、グループに対してのデータ配信が可能になると考えられる。マルチホップ時のデータ中継ノードは、適当な方向へのみルーティン

グを行い，かつ複数の宛先がある際は受信データの複製を行うことによりグループへの配信が行えグルーピングを実現することが出来ると考えられる．

3.3 アプリケーション・サポート・サブ層

アプリケーション・サポート・サブ層においては，既に提供されているバインディング機能を利用することでグルーピング方式は実現できると考えられる．ただしバインディング機能の特徴上，以下の性質を保有する．通信において，バインディング・テーブルを実装したノードを経由して通信が行わなければならない，またバインディング・テーブルを保持したノードから該当宛先ノードに向かって，片方向の通信しか実現することが出来ないため，逆方向の通信を行う際は逆方向のバインディング・テーブルを作成する必要がある．また，ネットワーク層のように中継時にデータを複製する事はせず，全てバインディング・テーブルが保持されているノードからユニキャストでデータの送信が行われる．データ転送を中継する複数のノードでバインディング・テーブル実装を行うことで，転送の効率化を達成できるものとする．これにより，グルーピングの実現は可能であると考えられる．

3.4 検討

上記の事柄より，ネットワーク層かアプリケーション・サポート・サブ層が妥当だと考えた．この2つの層についてさらに比較すると，ネットワーク層がデータ中継時に複製し適当な方向へのみ中継をおこなう事から，アプリケーション・サポート・サブ層で行うユニキャストより低位レイヤでの処理が行え，センサ・ノードにおける処理時間の低減，そして送信回数の削減による省電力化やトータルした送信量の削減からネットワークの輻輳の問題，そして送信時の効率化を達成できると考えられる．特に，クラスタ・ツリー型のようなツリー型である場合，ZigBee コーディネータ等のルート側のノードであるほど送信回数の増加やリンク間におけるネットワークの輻輳が発生しやすいと考えられる．その点において適宜データの複製を行う本提案方式は効率化を図ることができる．また，アプリケーション・サポート・サブ層のプロファイル，エンドポイントといった機能を利用し，ネットワーク層でグルーピングを実現すれば，より柔軟な操作や制御が可能になると考えた．そのような仮定のもと，ネットワーク層でのグルーピング方式を実現することが出来れば，各要点が改善できると考えられる．

これらの条件より，本研究ではネットワーク層にグルーピング方式の機能を追加することとした．

第4章 グルーピング方式の提案

本章では、提案するグルーピング方式についての説明をする。まず、グルーピング方式に先立ちグループ形成に必要となるグループIDの定義を行う。次いで、ZigBee ネットワークを構成する各トポロジ毎のグルーピング方式の実現方式について説明を行う。

4.1 グループIDの定義

まずグループIDの定義を行う。グループIDはZigBee ネットワークに存在するPAN 内で一意に識別できる値を採用する。また、グループIDは各ノードが保持するものとし、センサ・ノードに書き込む際に指定するものとする。センサ・ノードに書き込みを行う際、以下の2種類の方法を利用してノードにグループIDを保持させる。

- 各センサ・ノードにIDを指定
書き込みを行うノードにグループIDを指定する。グループIDを割り当てられたノードは、配置やリンク品質等から動的にネットワークに参加し通信を行う。
例：ノードAにグループID(GID1)を割り当てる
- 参加要求受け付けノードに情報を登録
新規ネットワーク参加のノードがPAN参加要求を送信し、要求を受け付ける親ノードに予め参加要求を行うノードのIEEE拡張アドレスを割り当てておく、これによりユーザが意図したネットワーク構成を静的に構成する事が可能となる。そのため、参加の要求・応答を行うノード間には実際に通信できる距離に配置する必要がある。

これらの二種類の方法を組み合わせる利点として、各センサ・ノードにグループIDを指定する場合は、各センサ・ボードの種類や保持する機能などをまとめることにより、各ノード操作時に開発者の意思を反映させることが出来る。参加要求受け付けノードにIEEE拡張アドレスの登録を行う場合は、ネットワーク構成に開発者の意思を反映させることができ、構成を静的に決定できるという点から、通る経路や配置など厳密な仕様でアプリケーションを動作させたい時などに有効活用できると考えられる。

続いて、グループ ID 数に関する検討を行う。ルーティングを行う方法として、ネットワーク層ヘッダにグループ ID に関連する情報を持たせるという方法を採用した。まず、ネットワーク層フレーム・フォーマットとその詳細を図 4.1, 4.2 に示す。利用フィールドの考慮の際、新たにフィールドを追加しネットワーク層ヘッダのサイズが大きく変わる事は望ましくないと考えたため、図 4.2 より、グループ ID を格納するフィールドとして 6bit の未使用のフィールド (10-15 bit) を採用した。これにより、グループ ID の総数は 6bit 分 (=64 個) となる。またグループ ID 値 0 は従来の通信方式の際に利用し、グループ ID として利用しないこととする。このフィールドを調べる事で従来の通信方式か提案方式かを見分ける事が可能となる。

Octets: 2	2	2	1	1	Variable
Frame Control	Destination Address	Source Address	Radius ^a	Sequence Number ^b	Frame Payload
	Routing Fields				
NWK Header					NWK Payload

図 4.1: ネットワーク層フレーム・フォーマット

Bits: 0-1	2-5	6-7 ^a	8	9	10-15
Frame type	Protocol version	Discover route	Reserved	Security	Reserved

↓
Group ID

図 4.2: Frame Control フィールドの詳細

柔軟なグループの構成を行うためには、各ノードが1つのグループにしか所属しないのでは、特定の限定的なグループしか形成できない。ネットワーク層で実装を行い、上位層であるアプリケーション・サポート・サブ層の機能を十分に利用するためには、1つのノードが複数のグループに所属できる必要があると本研究では考慮した。そこで ZigBee ノードは複数のグループに所属できるようにする。これにより、上位層であるアプリケーション・サポート・サブ層のバインディング・テーブルやエンド・ポイントの概念と組み合わせるネットワーク層でグルーピング方式を利用する事により、さらに柔軟性と効率を向上させられると考えた。具体例として、アプリケーション・サポート・サブ層で利用するプロファイルで定められた照明機能のうち、特定位置付近のグループ化を行った照明のみスイッチを ON/OFF することが可能にであり、かつあるグループ化された照明はある時刻になると自動的に暗さを感知し、照明の電源を操作するなどの操作性向上を実現することができる。

4.2 各トポロジ毎のグルーピング方式

各トポロジにおいて提案するグルーピング方式の実現の詳細について説明を行う。

4.2.1 スター型トポロジ

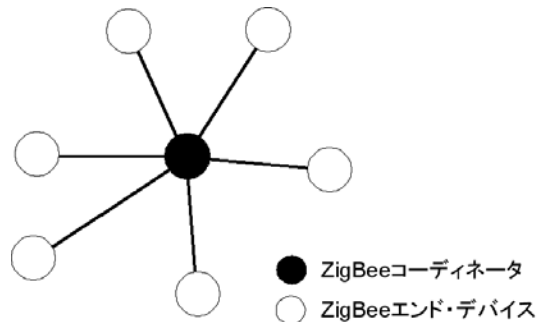


図 4.3: スター型トポロジ

スター型トポロジでは，中心に存在するのが ZigBee コーディネータであり，末端に存在するのが全てエンド・デバイスであると仮定する．ZigBee コーディネータから末端ノードである ZigBee エンド・デバイスまでは 1 ホップ通信範囲にある必要がある．そこでスター型の場合は，ZigBee コーディネータからブロードキャストを利用する事によりグループに対して配信を行う．コーディネータからのグループ配信の処理は以下の様になる．

- ZigBee コーディネータからグループへ送信
 1. 送信宛先ネットワーク層アドレスを 0xFFFF(ブロードキャスト)に設定
 2. ネットワーク層ヘッダの Frame Control フィールドの宛先グループ ID を設定
 3. ZigBee コーディネータがデータ送信

上記の流れを利用してデータ送信を行うが，データ受信確認の ACK¹が大きな問題となる．スター型トポロジにおいては，理論上は最大 65535 ノードまで接続が行える．そのため，ブロードキャストを利用して膨大なノード数が接続されているネットワークにおいてデータ送信を行った時には膨大な数の MAC 層 ACK が返ってくる ACK Implosion 問題が発生する．そのため，ZigBee では，ブロードキャストに対しては明示的に ACK を返さずにブロードキャストが中継されることをもって正式にデータが転送できた事を確認する Passive ACK 方式が採用されている．しかし，このスター型トポロジの場合，末端ノードが ZigBee エンド・デバイスであるためにデータの中継を行うことが不可能である．そのため，スター型トポロジにおいてはネットワーク層より上のアプリケーション層のレベルで明示的に ACK を返す必要がある．

¹Acknowledgement

引き続き，スター型トポロジの末端ノードからのグループに対する配信を説明する．スター型という性質上，先ほどの親から子への通信だけでなく，子から親を経由して再び子へと通信を行う場合も想定しなければならない．ZigBee エンド・デバイスである末端ノードは，子から子へ直接送信する事は基本的に不可能なため，親となる ZigBee コーディネータへ一旦データを預けるという方針をとらなければならない．受信した親は先に述べた処理の手順を行うことで通信を実現する．その処理の流れを以下に示す．

- ZigBee エンド・デバイスからグループへ送信

1. ZigBee エンド・デバイスは宛先ネットワーク層アドレスを自分の親 (ZigBee コーディネータ) 宛に設定
2. ネットワーク層ヘッダの Frame Control フィールドの宛先グループ ID を設定
3. ZigBee エンド・デバイスが ZigBee コーディネータに対し送信
4. 受信した ZigBee コーディネータが宛先アドレスを 0xFFFF に置換
5. ZigBee コーディネータがデータを送信

このような処理の流れを経て，末端ノードからグループへの送信が行われる．また該当グループへの配信を請け負う ZigBee コーディネータは，自分と接続関係を持っている各ノードがそれぞれどのグループに所属しているかを管理するために，隣接テーブル(表 2.4)に新たにフィールドを追加する．追加するフィールドは以下の表 4.1 になる．このフィールドを追加する事で，ZigBee コーディネータは自分が通信できる範囲にどのようなグループ ID に所属しているノードがいるかを管理する事が可能となる．つまり，ZigBee コーディネータはグループへのデータ送信後，ZigBee エンド・デバイスから返ってくる ACK を元にどのノードへのデータ到達が確認できたかを判断する事が可能となる．この様にスター型トポロジにおいては，ZigBee エンド・デバイスが自分の周りに存在する ZigBee エンド・デバイスのグループ情報を管理する事により，グループへの配信方法を実現する．ZigBee コーディネータが隣接テーブルに ZigBee エンド・デバイスの情報を取得し，グループ情報を登録する手順については次説のクラスタ・ツリー型トポロジにおける提案で述べる．

隣接テーブル追加フィールド	値	説明
Group ID	0x00-0x3F	隣接ノードのグループ ID

表 4.1: 隣接テーブルに追加するフィールド

4.2.2 クラスタ・ツリー型トポロジ

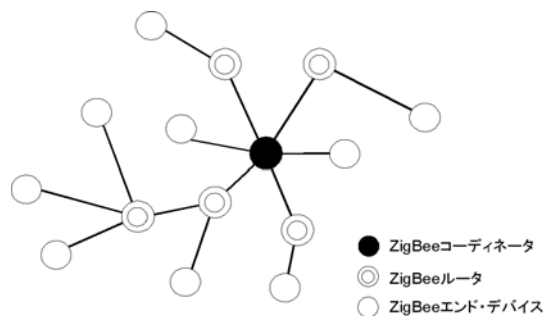


図 4.4: クラスタ・ツリー型トポロジ

クラスタ・ツリー型のトポロジでは、ZigBee コーディネータは必ずネットワーク・アドレス 0x0000 を割り当てルートを務めなければならない。また、ツリーのリーフ側に存在する末端ノードまでの中継ノードとして ZigBee ルータが必要となる。スター型との最も大きな違いは、ZigBee コーディネータ以外のノードも状況に応じて適宜ルーティングをしなければならないという点である。クラスタ・ツリーにおいては、ネットワーク・アドレッシング (2.8 節) の原理を利用し、ルーティングを行うことが可能である。しかしそれは、アドレス情報を元にしたものであり、グルーピング方式においてグループ ID からルーティングすべき方向を割り出す手段が必要となる。そこで、本研究では従来ツリーのルーティングを実現するための隣接テーブルのみならず、グループ宛に対してもルーティングを行うことができるグルーピング・テーブル (Grouping Table) を定義・作成する。グルーピング・テーブルはルーティング機能を持つ ZigBee コーディネータ、ZigBee ルータが保持し、グループ宛のデータを受信した際にルーティングすべき方向を割り出す。図 4.2 がグルーピング・テーブルのフォーマットである。このテーブルを構成する要素として、Next Hop アドレス、Group ID が存在する。テーブルの読み方として、クラスタ・ツリー型において Next Hop アドレスより下流に存在するグループ ID を示している。このグルーピング・テーブルを利用する事で、各中継ノードは下流方向へのルーティングを実現することが可能となる。またスター型と同様、各ノードが保持する隣接テーブルに追加フィールドとして Group ID を追加する。

ここで注意する点としては、グルーピング・テーブルに追加する Next hop アドレスとして、ルーティング機能を持つノード・アドレスのみ登録するという点である。ルーティング機能を持たない ZigBee エンド・デバイスは、隣接テーブルの追加フィールドである Group ID の欄に登録を行うため、ZigBee ルータは、グループ宛のデータをグルーピング・テーブルを参照・送信したのち、隣接テーブルに登録されている ZigBee エンド・デバイスにユニキャストで送信を行う。

Next Hop アドレス	Group ID	説明
0x0000-0xFFFF	0x00-0x3F	Next hop よりリーフ側に存在するグループ ID

表 4.2: グルーピング・テーブルのフォーマット

ZigBee では安定した有線の通信とは違い、無線状態の問題などから既にネットワークに参加しているノードの離脱やネットワーク・トポロジが変化する事が頻繁にあると考えられる。その際にグルーピング・テーブル等の情報を更新(追加・削除)する手段が必要となる。本提案方式では、ZigBee のネットワーク層の管理機能であるコマンドに新しくグループ更新の方式の定義を行い、利用する。ネットワーク層コマンド・フレームのフォーマットとしては以下の図 4.5 の構成となっている。また、実際に利用されるネットワーク層のコマンドとは、ルーティングに関連する機能をまとめたものであり、以下の表 4.3 の種類がある。これらのコマンドは一般的に、メッシュ型ネットワークで AODV 方式を利用する際等に用いられる。

Octets: 2	Octets: 6	1	Variable
Frame control	Routing fields	NWK command identifier	NWK command payload
NWK header		NWK payload	

図 4.5: コマンドフレーム・フォーマット

コマンド ID	コマンド名	説明
0x01	Route Request	経路探索要求
0x02	Route Reply	経路探索応答
0x03	Route Error	経路探索エラー
0x04	Leave	ネットワーク離脱
0x00, 0x05-0xFF	Reserved	予約済み

表 4.3: ネットワーク層コマンド ID の種類

ネットワーク層コマンドとして追加するフォーマットの説明を行う．追加実装するコマンド・フレームとして，グループ ID の追加・削除の 2 種類が挙げられる．各ノードはグループ ID を複数持つ事が出来るため，追加実装を行うコマンド・フレームは複数のグループ ID を同時に扱えるのが良い．

そこで本研究では，追加実装するコマンド・フレームを以下の図 4.6，図 4.7，表 4.4 のように定義した．図 4.6 のグループ ID 追加コマンドでは，3 つのグループを追加する例，図 4.7 では 2 つのグループが削除される例が示されている．Command ID フィールドには表 4.4 のコマンド値を入れ，Group ID Num で追加するグループ ID 数，その後実際グループ ID 値という順に続く．図 4.7 も同様に，コマンド ID の指定を行った後，削除するグループ ID 数，値の順に並ぶ．

グループ追加コマンドが送信されるタイミングとして，新規参加ノードが参加直後に親へ自分のグループ ID を通知する．これにより，参加を許可した親ノードは参加した子ノードがどのようなグループに所属するかを知ることができる．また，子ノードがルータとして動作をする際には，親ノードは隣接テーブルのみならずグループ ping ・テーブルにもその情報を登録・管理する．

グループ削除コマンドについては，ノードの離脱要求があった時，あるいは子ノードとのリンクが切断されていると判断した際にグループ ID 削除コマンドを親に対して通知する．削除コマンドは該当グループ情報が全て抹消された場合に限り，親ノードへと順次中継・広告していく．追加・削除コマンドは最終的にネットワーク・アドレスが 0x0000 である ZigBee コーディネータに到達したら，それ以上は中継せずそこで終了する．

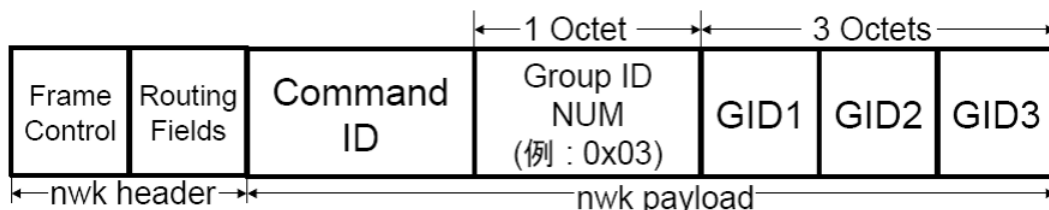


図 4.6: グループ ID 追加コマンド・フレームフォーマット

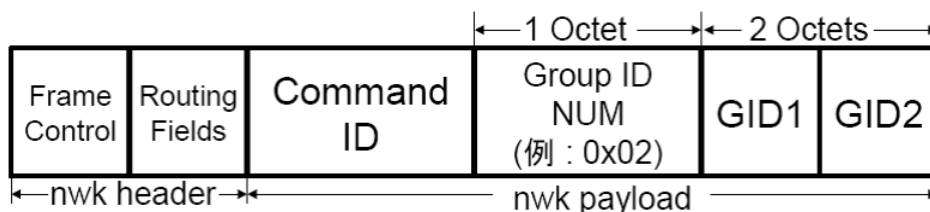


図 4.7: グループ ID 削除コマンド・フレームフォーマット

追加コマンドID	コマンド名	説明
0x07	Add Group ID	グループID 追加通知
0x08	Delete Group ID	グループID 削除通知

表 4.4: 追加コマンド ID

ここでは、クラスタ・ツリー型トポロジにおけるグループへのデータの配信処理の流れについて示す。ZigBee コーディネータから特定グループに所属するノードに対して、データを配信する処理は以下の様になる。

- ZigBee コーディネータからグループへ送信
 1. ZigBee コーディネータは該当グループID と該当する次ホップをグルーピング・テーブルから検索
 2. 検索後、該当ノードに対してユニキャストでグループID を設定して送信
 3. 隣接テーブルに登録されている該当グループの ZigBee エンド・デバイスを検索
 4. 検索後、該当ノードに対してユニキャストでグループID を設定して送信
 5. グループ宛データを受信したルータは、同様にグルーピング・テーブルと隣接テーブルを検索後、リーフ側へと中継
 6. グループに所属する全てのノードに行き渡り次第終了

以上の処理を行う事により、ツリー上の全てのノードにデータが行き渡る事が可能となる。これは、ルータ機能を持つ中継ノードが自分より下流にどのグループID を保持したノードが存在するかをグルーピング・テーブルによって保持する事により、適切な方向へルーティングを可能にしている。また、自分の子ノードとして存在している ZigBee エンド・デバイスに対しては隣接テーブルに追加したフィールドである Group ID を検索することにより、送信を行っている。ツリーに従いルートからリーフ側へ適切な方向へデータを配信する事で、PAN 内全体にデータを行き渡らせる事ができる仕組みである。続いて、ツリーの末端に存在する ZigBee エンド・デバイスから特定グループに対して、データを配信する処理は以下の様になる。

- 末端 ZigBee エンド・デバイスからグループへ送信

1. ZigBee エンド・デバイスは該当グループ ID で自分の親に向かいユニキャストで送信
2. 受信した ZigBee ルータは、さらに自分の親へとユニキャストで中継
3. ZigBee ルータはグルーピング・テーブルと隣接テーブルの検索を行い、データを受信した方向以外の該当すべき方向へとユニキャストで送信
4. 上記の 2, 3 の処理を ZigBee コーディネータに到達するまで繰り返し行う
5. ルートである ZigBee コーディネータが受信後、グルーピング・テーブルと隣接テーブルを検索し、データを受信した方向以外の下流に向かってデータを送信
6. ZigBee ルータはグルーピング・テーブルと隣接テーブルを検索し、繰り返しリーフ側へと中継
7. グループに所属する全てのノードに行き渡り次第終了

ツリーのルートでも末端でもない ZigBee ルータから始まるグループへの送信は、グルーピング・テーブルを参照して自分の下流方向へ送信すると共に上流へ中継を行うことで、PAN 内にデータが行き渡るように実現する。

- クラスタ・ツリー型トポロジにおけるグルーピング方式の動作例

クラスタ・ツリー型トポロジにおけるグルーピング・テーブル構成の仕組みをより明確にするため、以下に例を示す。ここで図4.8のような $C_m=3$, $R_m=3$, $L_m=3$ のクラスタ・ツリーを想定する。与えられた3つの構成パラメータより、 $C_{skip}(0)=13$, $C_{skip}(1)=4$, $C_{skip}(2)=1$ となる。これを元にアドレスを割り振ったものが図4.8に記載されている。

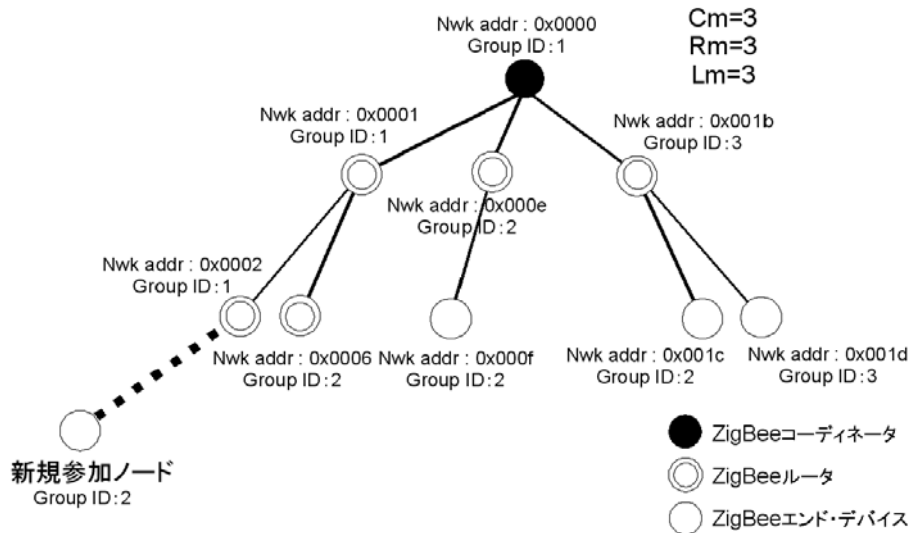


図 4.8: クラスタ・ツリー・ネットワーク例

ここでは、グループ ID 値が 2 の新規参加ノードがネットワーク・アドレス 0x0002 のノードの子になる場合を想定する。また、既に他のノードに関する各テーブルは作成されているものとする。

まず参加ノードは、参加と同時にネットワーク・アドレス 0x0003 を割り当てられる。そしてその際に、ネットワーク・アドレス 0x0002 のノードに対し、グループ ID 追加コマンド・フレームを利用して自分のグループが 2 であることを通知する。ネットワーク・アドレス 0x0002 のノードは隣接テーブルに新規参加ノードのグループ情報を登録し、さらに自分の親 0x0001 に向かいグループ ID 値 2 のノードが参加したことを同様に通知する。この時ネットワーク・アドレス 0x0001 のノードは、親の ZigBee コーディネータに対してグループ ID 追加コマンドは送信しない。何故なら、既にネットワーク・アドレス 0x0001 の子ノードとして 0x0006 (グループ ID 値 2) ノードが存在していることを、ZigBee コーディネータは知っているからである。このように既に同じグループ ID が存在していた際には、自分より親にグループ ID 追加コマンドを送信する事はしない。同様にノードが離脱を行う場合には、ノードの離脱により自分よりリーフ側に該当グループ ID が一つも存在しなくなった時のみ親にグループ ID 削除コマンドを送信するようにする。

最終的に ZigBee コーディネータ、ネットワーク・アドレス 0x0001, 0x0002 の ZigBee ルータが構成するテーブルを図 4.9, 表 4.5 に示す。

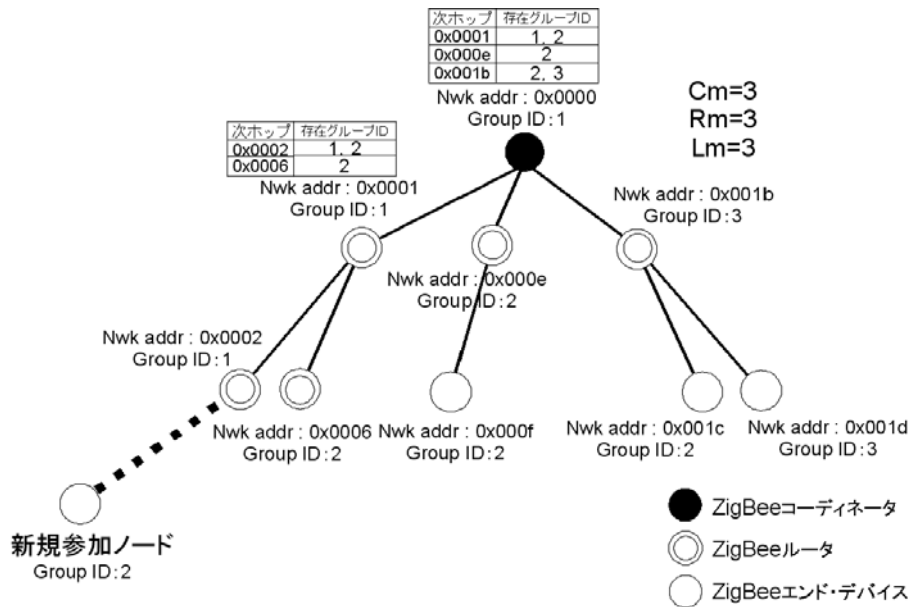


図 4.9: クラスタ・ツリーにおけるグルーピング・テーブル完成図

フィールド	値
PAN ID	省略
Extended Address	省略
Network Address	0x0003
Device Type	0x02(ZigBee エンド・デバイス)
Group ID	0x02

表 4.5: ZigBee ルータ (0x0002) 保持の隣接テーブル・エントリ

4.2.3 メッシュ型トポロジ

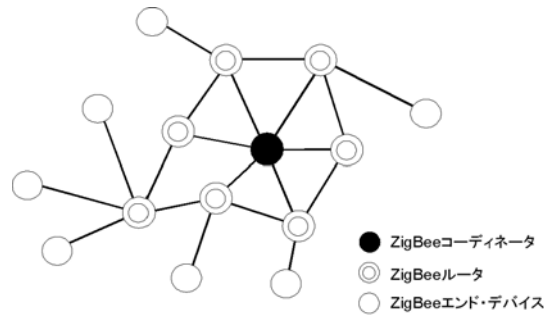


図 4.10: メッシュ型トポロジ

メッシュ型トポロジの特徴は、クラスタ・ツリー型トポロジと同様マルチホップの通信を行えるという点である。さらにクラスタ・ツリー型のトポロジと比較を行うと、ツリーに沿った固定的なルーティングのみならず、電波の状態やそれに伴うリンクの品質 (LQI) などの情報を利用する事で、ルーティング機能を拡張し経路に冗長性を持たせる事が出来るという点が大きな特徴として挙げられる。メッシュ型トポロジにも、クラスタ・ツリーなどで利用されるネットワーク・アドレッシング (2.8 節) の原理を適用する事により、ツリー・ルーティングを利用する事が出来る。

しかし最適な経路を利用してルーティングを行うためには、固定的なツリー・ルーティングではなく、常に最適な経路選択の方式をとらなければならない。そこで本研究は、メッシュ型ネットワークにおいて、全ての ZigBee ルータはアドレスのみで適切な方向を算出できるツリー・ルーティングを利用せず、AODV[6] を利用して作成されるルーティング・テーブルを用いてルーティングを行うものとする。メッシュ型でのグルーピング方式の説明に先立ち、まずメッシュ型ルーティングの AODV の簡単な動作概要について説明を行い、その後にグルーピング方式に対応した AODV についての説明を行う。

- ZigBee における AODV の概要

ZigBee において，メッシュ型に利用されるルーティングとして AODV を簡略化したものが採用されている．AODV に関する ZigBee ネットワーク層コマンド・フレームとして定められているのは，前述の表 4.3 より次の 3 種類である．本節では，Route Request を送信するノードを送信元ノード，Route Reply を返すノードを宛先ノードとして区別する．

1. Route Request
2. Route Reply
3. Route Error

Route Request

Route Request は，これから送信を行う送信元ノードが宛先までの経路の探索を行うために発行するコマンドである．Route Request のネットワーク・ペイロードは図 4.11 のようになる．

AODV はリアクティブ型²のプロトコルであるために，送信要求発生時にフラッディングを用いて PAN 内のノード全体にどの送信ノードがどの宛先までの経路を要求しているかを知らせる．その際，図 4.11 内の Route Request identifier を利用する事により，一度中継を行ったノードが再び同じ Route Request コマンドを中継するというブロードキャストの巡回を防ぐ．また，Path Cost フィールドではリンクの品質を加算して中継を行うことで，宛先ノードは自分までの経路中から最適なパスを選択する事が可能となる．

中継ノードは，シーケンスナンバー値，どの方向から Route Request が来たか，送信元ノードから自分までのコストを一時的に作成されるルート発見テーブル (表 4.8) に保存する．これにより，自身のノードから送信元ノードへ送信するためにはどちらに送信すべきか判断できる．何故なら，Route Request は各ノードを中継されて到達しているため，Route Request を受け取った方向へとルーティングすることにより，送信元までの経路は確保できる．

Route Reply

続いて Route Reply の説明を行う．Route Reply は，経路探索を行う送信元ノードが発行した Route Request に対し，宛先ノードがユニキャストで送信元へ返信を行うことにより経路を確立するコマンドである．Route Reply のネットワーク・ペイロードは図 4.12 のようになる．コマンド・フレームには，Route Request に対応する ID，送信元のアドレス，Route Reply を返信するノード・アドレス，そしてコストなどが含まれている．Route Reply を送信元まで中継する際に，中継ノードと送信元ノードはルート発見テーブルを完

²予めルーティング・テーブルを作成・管理しておくプロアクティブ型と要求が発生してから作成するリアクティブ型に分けられる

成させ、ルーティング・テーブル(表 4.6)に、送信元と宛先の2つのノードに対するエントリ (Next hop) を作成する事が出来る。つまり、送信元が Route Reply コマンドを受信し、出来上がったルーティング・テーブルを利用する事でデータを適切な方向へルーティングすることが可能となる。

Route error

最後に Route Error の説明を行う。Route Error のネットワーク・ペイロードは図 4.13 のようになる。Route Error は、何らかの要因によりルーティングが行われなない際にエラーを通知するコマンドである。ZigBee では、ルーティングが行われなない原因としてバッテリー電圧の低下を始め、ノードのルーティング容量の問題など様々な値が定められているが、ここでは省略をする。

Octets: 1	1	1	2	1
Command frame identifier (0x01)	Command options	Route request identifier	Destination address	Path cost
NWK payload				

図 4.11: Route Request ペイロード

Octets: 1	1	1	2	2	1
Command frame identifier (0x02)	Command options	Route request identifier	Originator address	Responder address	Path cost
NWK payload					

図 4.12: Route Reply ペイロード

Octets: 1	1	2
Command frame identifier (0x03)	Error code	Destination address

図 4.13: Route Error ペイロード

フィールド	サイズ	説明
Destination address	2 bytes	宛先アドレス
Status	3 bits	表 4.7 の値
Next hop address	2 bytes	次に送るべき方向

表 4.6: ルーティング・テーブル・フォーマット

値	状態
0x0	ACTIVE
0x1	DISCOVERY_UNDERWAY
0x2	DISCOVERY_FAILED
0x3	INACTIVE
0x4-0x7	Reserved

表 4.7: ルート・ステータス

フィールド	サイズ	説明
Route Request ID	1 byte	Route Request で利用されるシーケンス番号値
Source address	2 bytes	要求送信元アドレス
Sender address	2 bytes	要求宛先アドレス
Forward cost	1 byte	送信元から自分までのコスト
Residual cost	1 byte	自分から宛先までのコスト
Expiration time	2 bytes	ルート発見の失敗までの時間

表 4.8: ルート発見テーブル・フォーマット

- メッシュ型トポロジにおける通常のルーティング動作例

ここで、今まで示してきた AODV の処理の流れについてルーティング処理の動作例を見ていく。ここでは、メッシュ型ネットワークとして以下の図 4.14 の構成をとるものとする。送信元ノード S が宛先ノード D までの経路探索を行うシナリオを想定する。他のノードはルータ機能を持つノードであり、実線で結ばれたノード間のみで通信可能であると想定する。通信の状態から最適な経路を選択するために、実線付近の LQI 値をコストとして利用する。本節において、表現 (A, B, C) は通信の際にデータは A から B を経由して C に到達したことを示す。

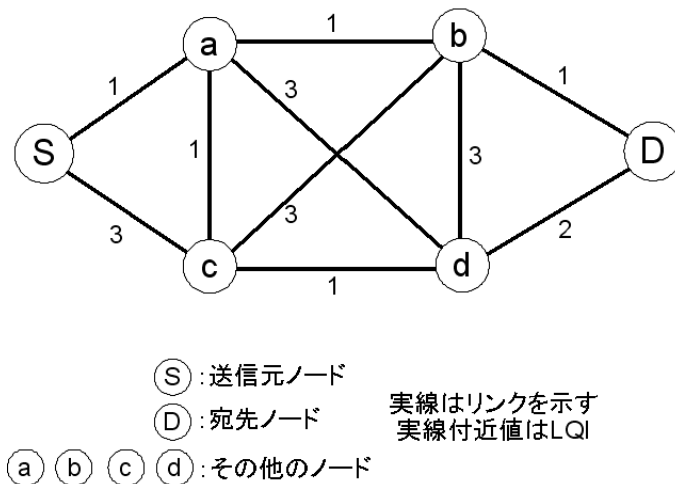


図 4.14: メッシュ型ネットワーク例

送信元ノード S は、宛先ノード D に対する送信要求が発生した際、ルーティング・テーブルに該当情報が無ければ経路探索を行う。まず送信元ノードである S は、Route Request(以下 RREQ) をブロードキャストする。ノード a とノード c は、ノード S が RREQ を送信したと判断し、シーケンス番号、ルート発見テーブルに送信元ノードの情報、送信元からのコストの情報を登録する。

次いで、a と c はランダム待ち時間後ブロードキャストの中継を行う。この時、送信元ノード S は自分が送信したパケットを受信する事により、自らが送信したパケットが正確に隣接ノードに中継されたことを確認する事が出来る。これは Passive ACK 方式と呼ばれ、明示的に ACK を受け取らずともデータの到達を確認する事が可能となる。

a, c が RREQ 中継後、c は RREQ について (S, a, c) の順で到着したものをより良いと判断する。なぜなら (S, c) と直接やってきた RREQ より (S, a, c) とやって来たパケットのほうがトータルのコスト値が低いからである。これにより、c はルート発見テーブルを最適な値で更新・再度中継する。この処理を繰り返し行うことで、最終的に RREQ が宛先ノード D に到着する。S から D への最適な経路は複数存在するが、コストの計算から (S, a, b, D) と決定される。

続いて、宛先ノードである D は、Route Reply(以下 RREP) を送信元ノード S へユニキャストで送信する。その際に、最適パス上に存在する中継ノード a と b は、ルート発見テーブルに登録した情報より、送信元ノードの方向へと RREP を中継する。RREP を受信・中継したノードは、送信元ノード S と宛先ノード D への双方向の経路を知る事ができ、それぞれに対してルーティング・エントリを作成³する事が出来る。

中継ノード c, d は、RREQ を中継したが RREP の受信処理を行うことはない。そのため、一定時間経過後に自動的に該当するルート発見テーブルが削除される。最終的に、ルーティング・テーブルの状態が ACTIVE となり利用できる状態を以下の図 4.15 に示す。

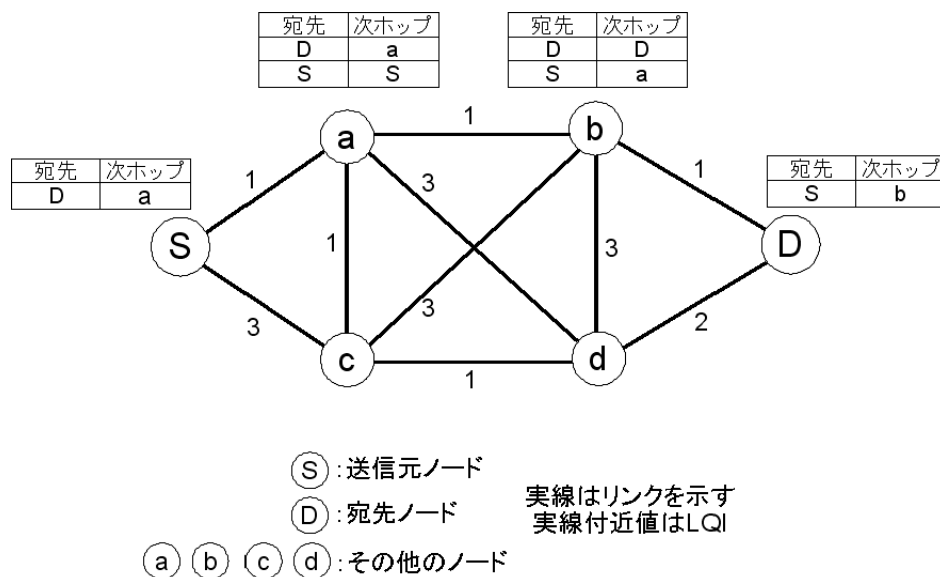


図 4.15: 通常 AODV のルーティング・テーブル完成図

³Network Information Base の属性値 nwkSymLink が TRUE の時、双方向の経路を登録する事が可能

- AODV の拡張

前節までに示した AODV について、特定のグループに対してデータを送信可能な方式を提案する必要がある。そこで、従来の AODV 方式に機能やコマンドを追加する事で従来の方式との共通性や類似性を持たせたまま、改良を最小限に抑える方式を考案する。以降、AODV を機能拡張、メッシュ型においてグループへのデータ転送を実現するための方式について説明を行う。提案方式においては、以下のコマンド・フレームを追加する。

1. Multi Route Request

2. Multi Route Reply

これら 2 つのコマンド・フレームは、通常の AODV 時の Route Request と Route Reply を多数のノード (グループ) に対応させたものであり、これを利用する事で、グループに対する経路探索を実行する事が出来る。本節では、Multi Route Request を送信するノードを送信元ノード、Multi Route Reply を返す複数のノードを宛先グループ・ノードとして区別する。以下に各コマンド・フレームの詳細を示す。

Multi Route Request

Multi Route Request は、送信元ノードが宛先グループ・ノードまでの経路探索を行うために発行するコマンドである。Multi Route Request のコマンド・ペイロードは図 4.16 のようになる。Route Request とのフォーマットの違いは、宛先アドレスが入っていた部分が宛先グループに変更された点である。Route Request からの大きな変化は、Multi Route Request では宛先グループ・ノードに Multi Route Request が到着し、Multi Route Reply を返した後にも、再びフラッディングを利用して Multi Route Request の中継を続けるという点である。これは、従来の Route Request と違い宛先の対象となるグループのノードが複数存在しているため、最適なパスの選択を行うために同じグループに属すノードを経由して送信元ノードまで応答を返す可能性があるからである。また、通常時の AODV では Route Request 受信時に、ルート発見テーブルに情報を登録し、Route Reply 受信時にルーティング・エントリが有効になっていたが、拡張版においては少し問題が発生する。この問題については後節の 4.3 で取り上げる。最適パスの選択に利用するコストについては従来方式と同様であり、中継ノードごとに値が加算されるという方式が採られる。

Multi Route Request

続いて、Multi Route Reply の説明を行う。Multi Route Reply のコマンド・ペイロードは図 4.17 のようになる。Route Reply からのフォーマットの変更は無いが、処理や動作が異なるために別のコマンド ID として定義を行った。Route Reply からの大きな変化は、Multi Route Reply を返すノードは一つではないという点である。そこで提案方式の Multi Route Reply ではある特徴を持つ。それを次に示す。

Octets: 1	1	1	1	1
Command frame identifier (0x05)	Command options	Route request identifier	Destination Group ID	Path cost
NWK payload				

図 4.16: Multi Route Request ペイロード

Octets: 1	1	1	2	2	1
Command frame identifier (0x06)	Command options	Route request identifier	Originator address	Responder address	Path cost
NWK payload					

図 4.17: Multi Route Reply ペイロード

フィールド	サイズ	説明
Destination Group	1 byte	宛先グループ
Status	3 bits	表 4.7 の値
Next hop address	2 bytes	次に送るべき方向

表 4.9: グループ拡張ルーティング・テーブル

複数回の Multi Route Reply の排除

特徴の一つとして、Multi Route Reply を既に返信したノードは、同じ Multi Route Request に対して Multi Route Reply の中継を行わないという事がある。これは二つの意味を持つ。一つ目は自身が Multi Route Reply を返信したという事より、送信元から自分までの経路の最適性は確定している。そのため他ノードから自分に対し Multi Route Reply が帰ってきたとしても、その到着に大きな時間差がない限り再び同じ経路を通過して送信元ノードまで到達するはずである。Route Reply は必ず Route Request を受信・中継したノードにしか送信されないため、他のノードから Multi Route Reply が帰ってくることは、自分からそのノードまでの経路の最適性も確保されていることを示している。二つ目は無駄なパケットの削減である。同じ要求に対して同じ意味を持つ応答を複数回返すことは、電力消費やネットワークの輻輳の観点からあまり得策でないと考えたために、この複数 Multi Route Reply を排除する方式を採用した。

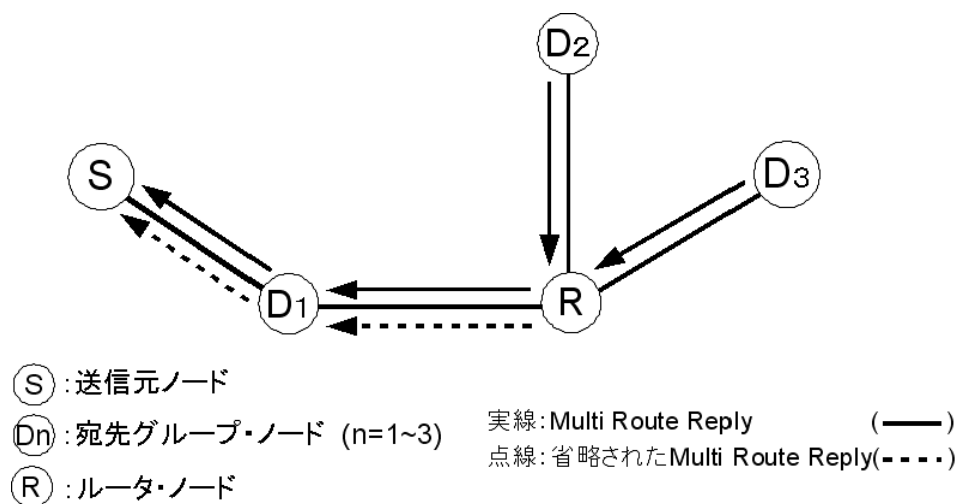


図 4.18: Multi Route Reply の中継

グループ ID の違いによるテーブルの変化

もう一つの大きな特徴として、送信元ノードの所属グループの違いにおけるルーティング・テーブル変化が挙げられる。作成するグループ拡張ルーティング・テーブルでは、送信元ノードの所属グループの違いにより異なる処理が発生する。具体例として次の2つのシナリオを想定する。

シナリオ1 送信元ノードが宛先グループ・ノードと同じグループ ID である場合

送信元ノードが宛先グループ・ノードと同じグループ ID に所属している場合、従来の経路探索方式と似た処理を行うだけで良い。具体的には、送信元ノードがブロードキャストを利用して Multi Route Request を発行し、周りのノードが中継する際にルート発見テーブルに送信されてきた来た方向の情報を登録し、続いて Multi Route Request を中継する。この時、受信したノードが送信要求を行っている宛先グループ・ノードであれば Multi Route Reply も返信する。そして、Multi Route Reply を返した中継ノードが、再び他ノードから同じ要求に対する Multi Route Reply を受信した際には、該当グループのルーティング・エントリに受信したノードのアドレスを登録する事によって、該当グループへの次ホップを確保する事が可能となる。これにより送信元ノードと該当グループ ID を持った宛先グループ・ノードの双方向のリンクを持つ事が出来る。

シナリオ2 送信元ノードが宛先グループ・ノードと違うグループ ID である場合

この場合は、シナリオ1の時とは異なる処理が必要となる。必要な処理を順に見ていく。送信元ノードが、ブロードキャストを利用して Multi Route Request を送信するところまでは同様だが、それを受信・中継するノードは送信元ノードがそもそもどのグループに所属しているかは、追加したコマンドフレームでは判別する事が出来ない。何故なら、情報として送信元の情報は Multi Route Request コマンドには含めないからである。また Multi Route Reply も、ユニキャストで送信されるため、送信元側の状況を知る事は出来ない。このサンプルとして、図4.19を示す。図4.19では、ノード a, D2, D3 は送信元方向を登録すべきか判断できないことを各ノードが保持するのルーティング・テーブル上に表現した。Multi Route Request に送信元や中継するノードのグループ情報を含めなかった理由として、一つのノードが柔軟なグループを形成するため複数グループ ID を持つため、ルート探索時にパケットサイズが大きいデータがネットワーク中にフラッディングされるのは望ましくないと考えたためである。

そのため、各中継ノードは該当グループ ID のルーティング・エントリに送信元ノードの方向の情報(具体的に言うと前ホップ)を登録すべきか、そうでないかは判断することができない。つまりこのシナリオの場合、送信元ノードの情報は登録せず自分より下流の情報のみを登録するという処理をとる必要がある。

解決策

これらの2つのシナリオの処理の違いを吸収するために本提案では、Multi Route Request を中継したノードは必ず Multi Route Request がやってきた方向をグループのルーティング・テーブルのエントリに追加する処理を行う。つまり、送信元ノードのグループIDが送信宛グループIDと同じかどうかに関わらず、送信元方向の情報は該当グループ・エントリに登録する。このルーティング・テーブルを有効にするタイミングとして、Multi Route Reply を中継する時が最適である。また、この時ルート発見テーブルはすぐ削除せず、タイムアウトまで保持し続け、他の Multi Route Reply を待つ。これにより、送信元ノードがどのグループIDに所属するかを問わずに確実にデータを届ける事は可能となる。しかしグループIDが異なる場合は、誤ったルーティング・テーブルが完成してしまっている恐れがある。そこで、実際にグループのルーティング・テーブルでデータ配信が行われ、データを受信したノードが該当グループIDでなく、かつ自分のルーティング・テーブルを検索して該当グループへのルーティング・エントリが存在しなければ、クラスタ・ツリー型トポロジで利用した図4.7のグループ削除コマンドを利用し、末端から上流に向かい削除通知を中継していく事で適正なルーティング・テーブルを完成させる。このときの処理は、クラスタ・ツリー型トポロジにおけるルーティングを参考にして欲しい。またこの問題に対して、Multi Route Request コマンド・ペイロードに宛先グループ情報と経由するノードのグループ情報を示す flag フィールドを用意することで対処することもできると考えられるが、それは今後の課題である。

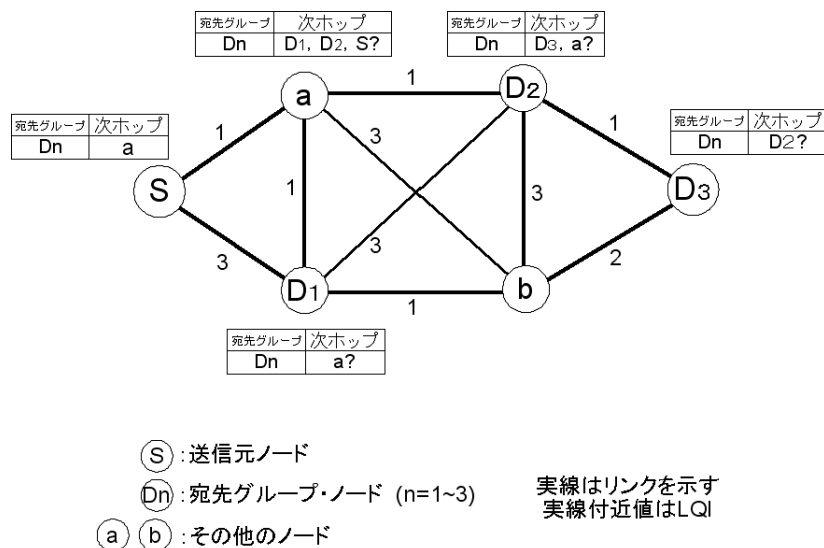


図 4.19: 送信元ノードと宛先グループ・ノードの ID の問題

4.3 拡張方式の問題点

前節まで AODV を拡張し、グループ宛にデータを配信する事が可能になった。本節では、メッシュ型トポロジにける AODV 拡張方式において経路確立からデータ送信を行うタイミングについての検討を行う。従来の AODV 方式と拡張 AODV 方式での一番の異なる点は、従来は Route Request を発信した送信元ノードに Route Reply が一つしか返って来なかったのに対し、拡張では Multi Route Reply が文字通り複数返ってくることである。これは以下の点を考慮する必要がある。

従来との相違点

1. 送信元ノードは、Multi Route Reply の数だけ送信を行う必要がある。
2. 完全なルーティング・テーブルが確立できるまでの時間にばらつきが発生する

この2つの問題は大きく関連性がある。例えば、送信元とリンクが確立されているノード(ノード A)と多数のホップを経由しないと通信できないノード(ノード B)が同じグループに所属し、送信元ノードがこのグループにデータ送信を行うとする。また、経路の状態としてノード B から送信される Multi Route Reply はノード A を中継し、ノード A は既に自分が Multi Route Reply を送信し、再び中継しない状況を想定する。

この時、ノード A からの Multi Route Reply はすぐ返ってくるが、ノード B は多数の中継を要するため Multi Route Reply が送信元ノードに到達するまでに非常に時間が掛かると考えられる。その時、送信元ノードがノード A からの Multi Route Reply が返ってきた時点ですぐに送信を行ってしまうと、ノード A 受信後、同一データをノード B 方向へルーティングすることが出来ない可能性が大きい。何故なら、ノード B からの Multi Route Reply がノード A に到達するまでに非常に時間が掛かるため、ノード A はそちら方向へルーティングすべきだということを知る事が出来ない。これは言い換えると、中継ノードがグループへの完全なルーティング・テーブルが作成される前にデータを受信してしまうと、完全な到達性が失われるということである。つまり送信元ノードは Multi Route Reply 受信後、データ送信を開始する前に一定時間待つ必要がある。これに対し、本研究では以下の解決策を定めた。これらの解決策をとる事により、問題点は克服する事が出来る。次にこの2つの解決策について詳細に説明を行う。

解決策

1. 経路確立までに十分な時間待ってから送信元は複数方向へデータを一括送信する
2. 待ち時間はネットワーク構成パラメータ値を基準に算出する

提案する解決策では，十分な時間待ってから該当グループIDが存在する方向へのみユニキャストを利用して送信を行う．ただし，前述の問題を解決するためには，待ち時間の算出方式を明確にする必要がある．そこで拡張 AODV 方式においてルート探索時に発生する各時間について以下に示す．

時間パラメータ

1. P：プロトコル・スタックにおける処理時間
2. T：通信時間
3. J：Multi Route Request 中継ランダム遅延

ネットワーク構成パラメータ

1. D：PAN の最大の階層数
2. H：通信に必要なホップ数
3. N：ノード数

以上のパラメータを利用する．次に各パラメータ値について説明する．

- P：プロトコル・スタックにおける処理時間
プロトコル・スタックにおける処理時間は，ノードがデータを受信・送信に必要な処理時間とする．この処理時間には，処理するプロセッサの能力やセンサ・ノードにおける割り込み遅延やキューイング遅延などの OS 実装に依存する部分を総合して，ここでは P と表す．P はデータ受信後から送信までの時間を 1 単位として考える．P はプロセッサや，OS の実装により値が変動するためここでは P という記号で表す．
- T：通信時間
データが無線として伝播される際の遅延を示す時間である．ZigBee が日本国内で利用できる周波数帯は 2.4GHz 帯であり，理論上は最大 250kbps で通信が可能とされている．ZigBee 規格で定義されたネットワーク層コマンド・フレームの Route Request，Route Reply はヘッダ情報を含め，約 30Byte である．つまり，Route Request，Route Reply を送信するのに掛かる時間はおよそ

$$T = \frac{30 \times 8}{250,000} = 0.96[ms] \quad (4.1)$$

およそ 0.96[ms] となる．

- J : Multi Route Request 中継ランダム遅延

J は Multi Route Request 中継ランダム遅延時間である。Multi Route Request 中継ランダム遅延とは、通常 AODV 等で Route Request コマンドを中継する際、何も考慮せずに受信後すぐに中継を行うと同じ Request に対して他のノードと中継が重なったり、Passive ACK 方式のためのノードの準備のための確認時間が十分でない可能性がある。そこで ZigBee 規格 [3] では、Route Request を中継する際に以下のパラメータに従ってランダム遅延を発生させるように規定している。

ランダム遅延時間 J

$$J = 2 \times R[nwkcMinRREQJitter, nwkcMaxRREQJitter] \quad (4.2)$$

($R[A, B]$: A から B までの範囲のランダム値)

但し、ZigBee 規格上の固定値として

$$nwkcMinRREQJitter = 2, nwkcMaxRREQJitter = 64$$

が採用される。結果としてランダム遅延 J の値の範囲は以下の様になる。

$$4 \leq J \leq 128 \quad [ms] \quad (4.3)$$

- D : PAN の最大の階層数

ZigBee において PAN を構成している階層数が分かれば、最も長い距離における送信待ち時間の推測が可能となる。パラメータ D は PAN を管理・構成する ZigBee コーディネータの持つ $nwkcMaxDepth$ 値が採用される。しかし、 $nwkcMaxDepth$ 値は規格上の最大階層 (15) を規定するものであり、実際に PAN 構成時にはパラメータ $nwkMaxDepth$ が利用される。一般的に $nwkMaxDepth$ は 5 と設定され、本提案においてはその値を採用する。ネットワークを構成・通信を行うために少なくとも階層が必要であるため、最小値として 1 を採用する。

- H : 通信に必要なホップ数
通信に必要なホップ数はHで表される。最長の場合を想定すると、通信に必要なホップ数Hは次のようになる。

$$H = 2 \times D \quad (4.4)$$

- N : ノード数

実際に通信される際に経由するノード数(含:送信元ノード,宛先ノード)は,

$$\begin{aligned} N &= 2D + 1 \\ &= H + 1 \end{aligned} \quad (4.5)$$

以上のパラメータより、待ち時間のトータル値を計算する推定する事が出来る。待ち時間として、Multi Route Request が行き渡るのに必要な時間と Multi Route Reply が返ってくる時間について、それぞれ最悪のケースを想定しその和を求めればよい。計算式を式 4.6 に示す。送信元ノードはこの待ち時間だけ待った後にデータを送信する事で、拡張 AODV 方式の問題点で指摘した問題に対処することが出来ると考えられる。なお、各スタックの必要処理時間や送信失敗におけるリトライ回数などについて考慮したものは後章の 6 で評価を行い、各パラメータを変更した上で必要な待ち時間についての検討・考察を行う。

最悪なケースの送信待ち時間計算式

$$\begin{aligned} \text{待ち時間} &= (\text{MultiRouteRequest 時間}) + (\text{MultiRouteReply 時間}) \\ &= \{J \times (2D - 1) + H \times T + N \times P\} + (H \times T + N \times P) \\ &= J(2D - 1) + 2HT + 2NP \quad [ms] \end{aligned} \quad (4.6)$$

例として、 $P = 100[ms]$ 、 $T=1[ms]$ 、 $J = 64[ms]$ 、 $D = 3$ とすると

$$\begin{aligned} \text{待ち時間} &= 64 \times (6 - 1) + (4 \times 3 \times 1) + (2 \times 7 \times 100) \\ \text{最悪なケースの待ち時間は} & \quad 1732 \quad [ms] \quad \text{となる。} \end{aligned}$$

- メッシュ型トポロジグループ핑擴張方式のルーティング動作例

メッシュ型トポロジにおけるグループに対して配信するための経路探索の仕組みをより明確にするため、以下に例を示す。ここで図 4.20 のようなメッシュ型のネットワークを想定する。各ノードのアドレスは、各ノードを区別するため $C_m=5$, $R_m=4$, $L_m=2$ のツリー・アドレッシングを利用した。しかし、本提案では完全メッシュを想定しており、ツリー・ルーティングは利用せず、拡張 AODV 方式を用いて経路探索を行う様子を示す。

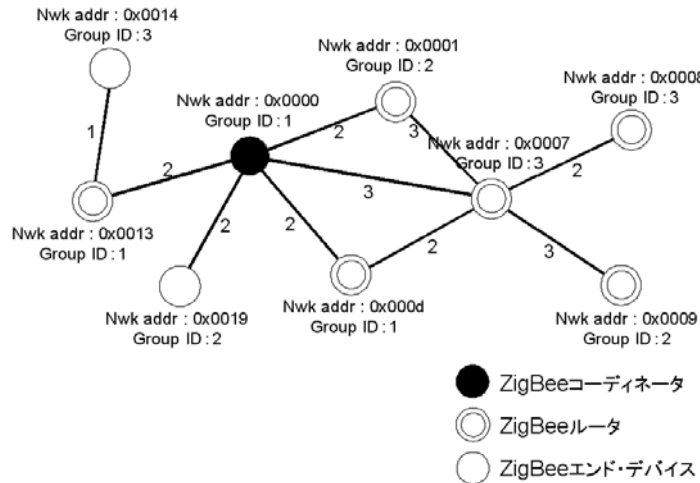


図 4.20: メッシュ型ネットワーク構成例

想定するシナリオとして、グループ ID 値 1 に所属する ZigBee コーディネータがグループ ID 値 3 のノード群に対し、一括に送信を行うための経路探索を行う様子を想定する。まず ZigBee コーディネータ (0x0000) は、グループ ID 値が 3 のノードが PAN 内のどこに存在するかを知る事が出来ないため、Multi Route Request をフラッディングを利用して送信する。アドレス 0x0001, 0x0007, 0x000d, 0x0013 などのノードはこの要求を受信し、中継を続けると共に、該当グループに所属する 0x000d のみ Multi Route Reply を ZigBee コーディネータに向かって送信する。続いて、該当グループに所属する 0x0008 や 0x0014 といったノードが Multi Route Request を受信し、送信元の ZigBee コーディネータに対し、Multi Route Reply を送信する。この時 ZigBee ルータである 0x0007 は、既に自分が一度 Multi Route Reply を返信しているため、0x0008 から送られてきた Multi Route Reply は送信元へと中継を行わない。しかし応答の受信と同時に、0x0007 のグループ ID 値 3 のルーティング・エントリに 0x0008 のアドレスを次ホップとして登録する事で、データ受信時にそちらの方向にルーティングする事が可能となる。まとめると、このネットワーク構成例の場合 ZigBee コーディネータが受信した Multi Route Reply の数は 2 つ (0x0007, 0x0013 方向より) となる。また、送信元の ZigBee コーディネータが所属するグループ ID 値が 1 であるのに対し、宛先グループ ID 値が 3 であるということから、前述のグループ ID によるテーブルの変化を考慮しておく必要がある。

具体的には、この場合アドレス 0x0007 の ZigBee ルータは、ZigBee コーディネータがグループ ID 値 1 に所属していること、ZigBee コーディネータの反対側にグループ ID 値 3 のノードが存在している事など判断する術が無い。そこで、不確定ではあるが Multi Route Request を受信した方向 (前ホップ) を該当グループ ID のルーティング・エントリに登録する。登録後、Multi Route Reply が返ってきたタイミングでこのエントリを有効にする。これにより、最適な経路として選ばれたノードだけが、該当グループへの経路を保持する事になる。また本例の場合、ノードアドレス 0x0014 が存在するために 0x0007 のノードが登録した情報に誤りは存在しない。もしノードアドレス 0x0014 やグループ ID 値 3 のノードが送信元ノード側に存在しなければ、正確な動作はするが余分で unnecessary 情報を持ったルーティング・テーブルになってしまう。データを配信され自分が該当グループ ID でなく、かつ自分より下流に該当グループ ID が存在しなかった場合に限り、上流方向へとグループ削除コマンドを送信する事で、各ノードの拡張ルーティング・テーブルの修正を行う。また各ノードのグループ情報を隣接テーブルへ登録する際の手順は、他のトポロジと同様参加直後に実行する。

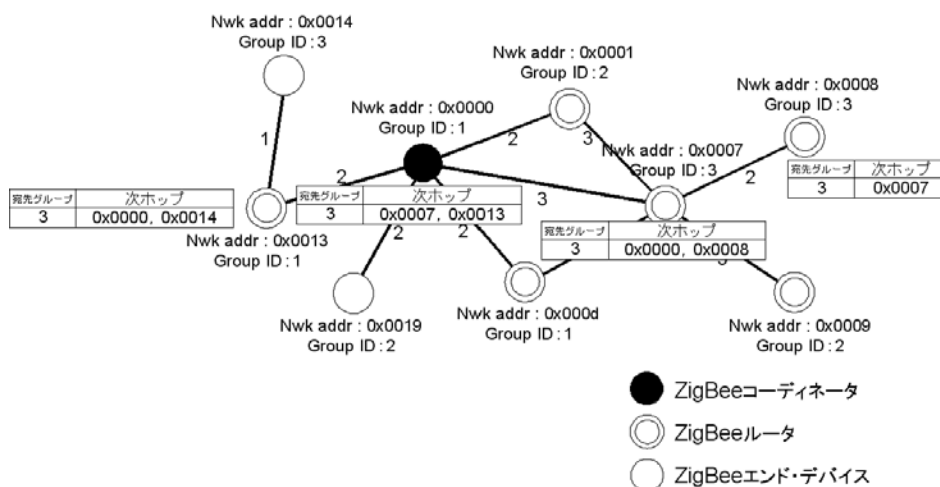


図 4.21: メッシュ型ネットワークのグルーピング・テーブル完成図

フィールド	値
PAN ID	省略
Extended Address	省略
Network Address	0x0014
Device Type	0x02(ZigBee エンド・デバイス)
Group ID	0x03

表 4.10: ZigBee ルータ (0x0013) 保持の隣接テーブルの例

4.4 提案方式のまとめ

これまでに、ZigBee ネットワークにおけるスター型、クラスタ・ツリー型、メッシュ型のグルーピング方式について提案してきた。本節では各提案方式のまとめを行う。

スター型トポロジ

- 必要となるテーブル
 - グループ ID フィールドを含んだ隣接テーブル
- 必要なフレーム・タイプ
 - グループ ID フィールドを含んだネットワーク層ユニキャスト・データ・フレーム (ZigBee エンド・デバイスが利用)
 - グループ ID フィールドを含んだネットワーク層ブロードキャスト・データ・フレーム (ZigBee コーディネータが利用)
- 問題点
 1. 中央に存在する ZigBee コーディネータは、接続ノードがどのグループに所属しているか管理する必要がある。
 2. ZigBee コーディネータのブロードキャストに対し、MAC 層での ACK を利用すると該当グループ・ノードが大量に存在すると ACK の集中が発生。
- 解決策
 1. ZigBee コーディネータは、隣接テーブルに新規のグループ ID フィールドを定義する事により管理を行う。
 2. ブロードキャストに対しては MAC 層では ACK を返さずアプリケーションレベルでの ACK を利用する。

クラスタ・ツリー型トポロジ

- 必要となるテーブル
 - グループ ID フィールドを含んだ隣接テーブル
 - グルーピング・テーブル
- 必要なフレーム・タイプ
 - グループ ID フィールドを含んだネットワーク層ユニキャスト・データ・フレーム
 - グループ ID 追加コマンド・フレーム (Add Group ID)
 - グループ ID 削除コマンド・フレーム (Delete Group ID)

- 問題点
 1. 同じグループ ID を持つノードに対してデータを配信する手段が必要 .
 2. 大きなツリーを構成するため , 中継機能を持つ ZigBee コーディネータや ZigBee ルータは効率良くグループ情報を管理する必要がある .
- 解決策
 1. 全ての中継ノードはグルーピング・テーブルを作成し , 新規ノードのネットワーク参加 , 離脱にともないテーブルを更新し管理する . また , スター型と同様にグループ ID を追加した隣接テーブルも保持する . これにより RFD , FFD を区別して管理を行うことができる .
 2. ツリーというトポロジの構造上 , リーフ側に存在するグループを管理するだけで全てのノードへデータを行き渡らせる事が可能となる .

メッシュ型トポロジ

- 必要となるテーブル
 - グループ ID フィールドを含んだ隣接テーブル
 - グループ拡張ルーティング・テーブル
- 必要なフレーム・タイプ
 - グループ ID フィールドを含んだネットワーク層ユニキャスト・データ・フレーム
 - 該当グループへの経路探索要求するコマンド・フレーム (Multi Route Request)
 - 該当グループへの経路探索に回答するコマンド・フレーム (Multi Route Reply)
 - グループ ID 追加コマンド・フレーム (Add Group ID)
 - グループ ID 削除コマンド・フレーム (Delete Group ID)
- 問題点
 1. 同じグループ ID を持つノードに対して経路探索を行う方法が必要
 2. 送信元と宛先ノードのグループの違いにおいてテーブル構成が変化
 3. 中継ノードが保持するグループ拡張ルーティング・テーブルの完成までの時間
- 解決策
 1. 従来の AODV 方式を拡張することで , 該当グループまでの経路探索を行う . 従来の AODV 方式と区別するために , 新たなコマンド・フレーム (Multi

Route Request/Reply) を利用し, 経路情報は, 各中継ノードがグループ拡張ルーティング・テーブルとグループIDを含んだ隣接テーブルに保持する.

2. 送信元ノードと宛先ノードが違うグループに所属すると Multi Route Reply を返すノードは, 送信元方向をグループ拡張テーブルに登録すべきかを判断できない. そこで送信元と宛先グループの違いによらず双方向へのテーブル情報を保持し, 不必要であればグループID 削除コマンド・フレームを利用して経路情報の削除を行う.
3. 従来の AODV とは違い経路探索時に応答が複数返ってくるため, 送信元ノードからデータ配信時には中継ノードが完全なグループ拡張テーブルを構成するための待ち時間を設ける.

第5章 提案システムの実装

5.1 実装の検討

これまでに提案を行ってきた方式の実装について検討を行う。提案方式を実装する際、

1. 実機のセンサ・ボードを利用して実装
2. PC等の環境において仮想センサ・ボードを作成して実装

という方法が考えられる。まず、実機での実装を検討する。実機で実装するには、各トポロジを実現できるだけのいくつかのセンサ・ボードとその ZigBee 規格に相当するソースコードを手に入れる必要がある。2006年現在、ソースコード付きでセンサ・ボードを販売しているところは稀であり、実際に手を加えられるのはアプリケーションに関する部分のみというセンサ・ボードが大多数であるため、こちらでの実装は困難な点が多いと判断した。続いて仮想センサ・ボードでの検討を行う。仮想センサ・ボードの作成に当たり、MSSTATE_LRWPAN[7]と呼ばれるスタックを検討する。これは IEEE802.15.4 とその上位であるネットワーク層以上を簡略化した ZigBee 仕様で実装したスタックであり、Web上で教育用・研究用などを対象に非商用目的で利用する事が出来る。このスタックにおいてノード種類、機能としてサポートするのは、

ノードの種類

- Coordinator(ZigBee コーディネータ)
- Router(ZigBee ルータ)
- RFD(ZigBee エンド・デバイス)

サポート・トポロジ

- スター型トポロジ
- クラスタ・ツリー型トポロジ

機能

- ツリー・ルーティング
- スタティック・バインディング

となっている。機能としてツリー・ルーティングが実装されているが、2006年の年末現在、ZigBee版 AODV やコマンド・フレーム機能が未実装であるため、スター型トポロジとシンプルなクラスタ・ツリー型トポロジしかサポートされていない。しかし、MSSTATE_LRWPAN スタックは既に販売されている次のプラットフォーム(表 5.1)でも動作し、動作の信頼性が高いために、実機にとっても近い動作が実現できると想定している。そこで本研究では、MSSTATE_LRWPAN スタックに提案グルーピング方式の実装を行った。

プラットフォーム名	製造元	コンパイラ
PICDEM Z	Microchip Inc.	MCC18 又は HI-TECH
CC2430 Evaluation board	Texas Instruments, Inc.	IAR51, HI-TECH C51
WIN32 Virtual Board	Robert B. Reese	MS Visual Studio .NET 2005

表 5.1: MSSTATE LRWPAN 動作プラットフォーム

項目	名前・値
OS	Windows XP Professional(5.1, ビルド 2600)
プロセッサ	Intel Xeon CPU 3.80GHz (2CPUs)
メモリ	2048MB RAM
開発環境	Microsoft Visual Studio 2005 Version 8.0.50727.42 Microsoft .NET Framework Version 2.0.50727

表 5.2: MSSTATE LRWPAN 実装環境

5.2 性能の評価

本節では、PC に実装した仮想センサ・ボードの処理性能について調査する。調査項目として、PC 上の仮想センサ・ノードがデータの送信を行うまでのレイテンシを取り上げた。本研究の提案方式では、メッシュ型における経路探索時に送信待ち時間が発生する。その際に、各センサ・ノードのスタックの処理時間などは待ち時間と大きく関連するため、本節において仮想センサ・ボードの送信のレイテンシを参考までに計測した。レイテンシの調査に当たり、比較を行うためにデスクトップ型パソコンでの評価とノート型パソコンにおいて複数台の仮想センサ・ボードを起動させ、スター型トポロジを形成して送信時のレイテンシについて調査を行った。デスクトップの環境の詳細については表 5.1 に記してあるとおりである。またデスクトップ型パソコンとノート型パソコンの同じ開発動作環境とするために Visual Studio や .NET Framework について全く同じバージョンをインストールし評価を行った。以下にノート型パソコンのスペックと実験の構成の詳細について説明を行う。

項目	名前・値
OS	Windows XP Home Edition(5.1, ビルド 2600)
プロセッサ	Intel Pentium4-M CPU 2.00GHz
メモリ	768MB RAM

表 5.3: ノート型パソコン実験環境

実験の構成の詳細

レイテンシを測定する実験の構成として、シンプルなスター型トポロジを想定する。PANを構成するノード数として、必須な中心の ZigBee コーディネータを 1 つ、そして ZigBee エンド・デバイスの個数として 1, 5, 9 という 3 つのパターンを用意した。つまり、一つの PC 上において最大 10 台の仮想センサ・ボードをエミュレートしていることになる。想定するアプリケーションの処理の流れとして、ZigBee エンド・デバイス側から ZigBee コーディネータに対し Ping Request に相当するデータを送信し、ZigBee コーディネータは、受信後 Ping Reply を送信元ノードへと返す処理を繰り返し行う。この時に、各機器における ZigBee コーディネータ・ZigBee エンド・デバイスの送信レイテンシを調べる。その処理の結果が以下の様になった。

		ZigBee エンド・デバイス個数		
		1 ノード	5 ノード	9 ノード
Desktop PC	Coord	約 15(μ s)	約 15(μ s)	約 14 ~ 300(μ s)
	RFD	約 15(μ s)	約 15(μ s)	約 15 ~ 35(μ s)
Note PC	Coord	約 50(μ s)	約 80 ~ 320(μ s)	約 170 ~ 890(μ s)
	RFD	約 50(μ s)	約 100 ~ 500(μ s)	約 210 ~ 300(μ s)

表 5.4: 各 PC におけるレイテンシの比較

各 PC において ZigBee エンド・デバイスの数を変更しながら比較した結果、1 台の PC 上に多くの仮想センサ・ノードを作成するほどレイテンシ値が大きくなったことが表 5.4 から見て取れる。特にデスクトップ型パソコンの方は、ノート型と比較してプロセッサやメモリ等の点で非常に高性能であるため 10 程度のノード数ではレイテンシにそこまで大きな変化は見られなかった。またノート型 PC においては、ZigBee エンド・デバイスを 1 個だけエミュレートした際にも、およそデスクトップの 3 倍程のレイテンシが発生していることが判明した。両機器とも稀に大きなレイテンシ値を取ることがあるが、殆どの時間において最低値あたりの付近で落ち着いていた。

結果としては、PC という高性能環境で実験を行ったため、本実験の場合は両機器ともレイテンシ値が大きくても 1[ms] 程度に収まることが分かった。

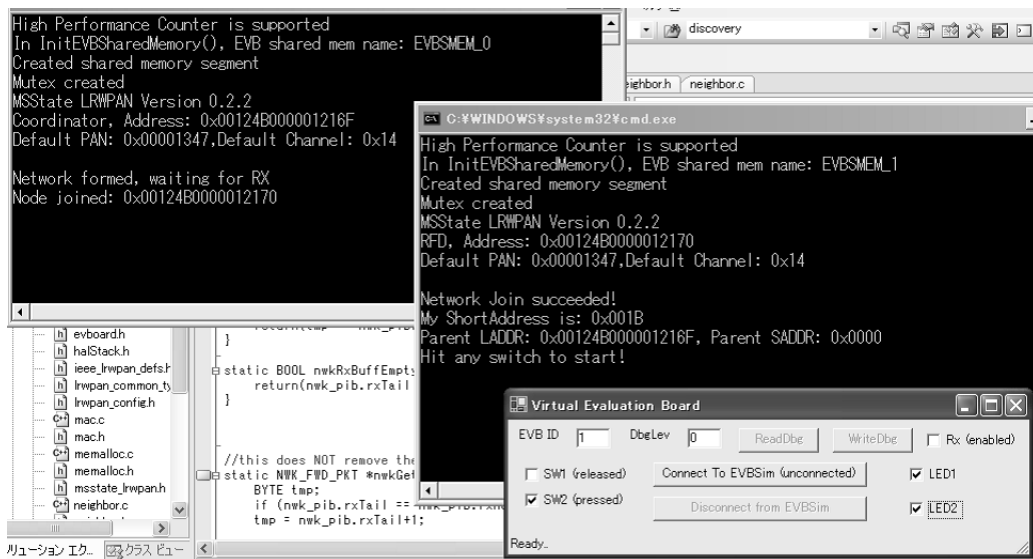


図 5.1: 実装した MSSTATE LRWPAN システム

第6章 提案システムの評価

本章では、実装した提案システムを様々な点から比較を行い、各要点が従来の方式からの程度変化したかを評価する。評価の種類として大きく分けて三種類存在する。

1. 各トポロジ毎にアプリケーションを考案し、従来方式と提案方式において各評価項目の比較を行う。
2. ZigBee に求められているアプリケーション例に対し、トポロジ・グルーピングを検討し、それぞれ考察を行う。
3. メッシュ型トポロジにおいてグループに対する経路探索後の送信待ち時間における最適性について検討を行う。

まず、一つ目の各トポロジにおけるアプリケーションの構築について具体的な評価の流れを説明する。最初に、各トポロジを形成するために必要な情報(ノード数、最大階層数、最大ルータ子ノード数、最大子ノード数等)を決定する。続いて、従来の方式と提案方式において異なるグルーピングを用いて、評価を行う。評価パラメータとして、以下を評価する。これらの評価項目は、各トポロジ毎について取り上げる。グループの分割方法では複数のパターンを検討し、グループの分け方の違いにおける変化を評価する。評価項目として通信回数や送信量の他に、グループに対する特別なテーブルを作成した場合に、それらのサイズがグループの構成とどのような関係になるかを取り扱う。

- グループの分割方法
- 各ノードの必要な通信回数
- 各ノードのデータ送信量
- 各ノードのテーブル・サイズ

二つ目の ZigBee に求められているアプリケーション例を元にネットワークを構築する考察については、求められているアプリケーションの実態を調査し、それを構築する際にどのようなトポロジや提案方式のグルーピング手法を利用すると効率化を図ることができるかについて比較を行う。具体的な評価の事柄として、以下の点に着目しそれぞれについて比較・考察を行う。この評価項目では、各トポロジ毎の評価で得られた情報を元にアプリケーションに対して最適な構成を追求する。

- 最適なトポロジの種類
- 最適なグループの割り当て

評価の種類の一つ目として、メッシュ型トポロジの経路探索時の待ち時間が挙げられる。先の4.3節で述べた通り、経路探索時に完全なルートが構築されるまでに、送信元ノードはいくらかの待ち時間を要する。そこで以下の項目について注目し、最適な待ち時間について検討・考察を行う。

- ネットワーク構成の違い・スタック処理時間による待ち時間
- リトライ発生回数の違いによる待ち時間

6.1 各トポロジにおけるアプリケーションの考案・評価

6.1.1 スター型トポロジ

トポロジ構成情報の決定

ノード数として、スター型の中心である ZigBee コーディネータを除いて 28 のノードが存在すると仮定する。これらの 28 個の ZigBee エンド・デバイスはある程度のグループ単位に分割し、特定グループ ID に所属するノードへ ZigBee コーディネータからデータを送信するシナリオを想定する。

想定するスター型アプリケーション

スター型を用いて構築されるアプリケーションとして、ある家のリビングに存在する家電の機器の電源 ON/OFF 操作を行うシステムを想定する。従来の ZigBee の機能で実現可能な単体の機器の電源の ON/OFF 操作に加え、特定の場所付近にある照明の操作、あるいは特定の家電に関連した機器のみの電源操作など、提案方式を用いることでより柔軟に操作する事が出来る。イメージ図 6.1 のようなスター型トポロジをイメージしてもらうと良い。このような構成で ZigBee コーディネータ (1 個) と ZigBee エンド・デバイス (28 個) が同じ室内に存在していると考える。

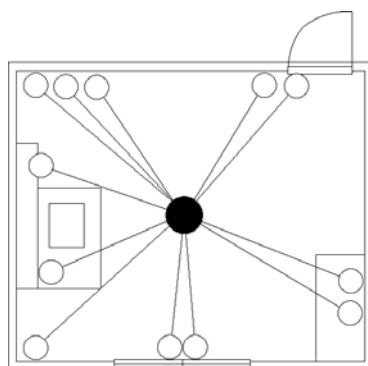


図 6.1: 室内のスター型アプリケーションのイメージ図

- グループの分割方法

提案方式のスター型トポロジの場合、ネットワーク層ヘッダにグループ ID 情報を含みブロードキャストを利用して送信するため、ユニキャストと比較した際、どれ程の数のノードが同じグループ ID に所属するかによって効率化の度合いが変化する。また全てのノードが同じグループに所属するということは全てのノードに対し一括でデータ転送を行えるという点で、従来方式のブロードキャストとは相違ないが、データを破棄するノードが存在した際に低位レイヤで破棄することが可能になるというメリットが挙げられる。パターンとして以下の 3 種類を想定する。

1. 特定グループの8ノード(パターンA:8ノード×1グループ)に送信
2. 特定グループの8ノード(パターンB:4ノード×2グループ)に送信
3. 8ノード全てにユニキャスト送信

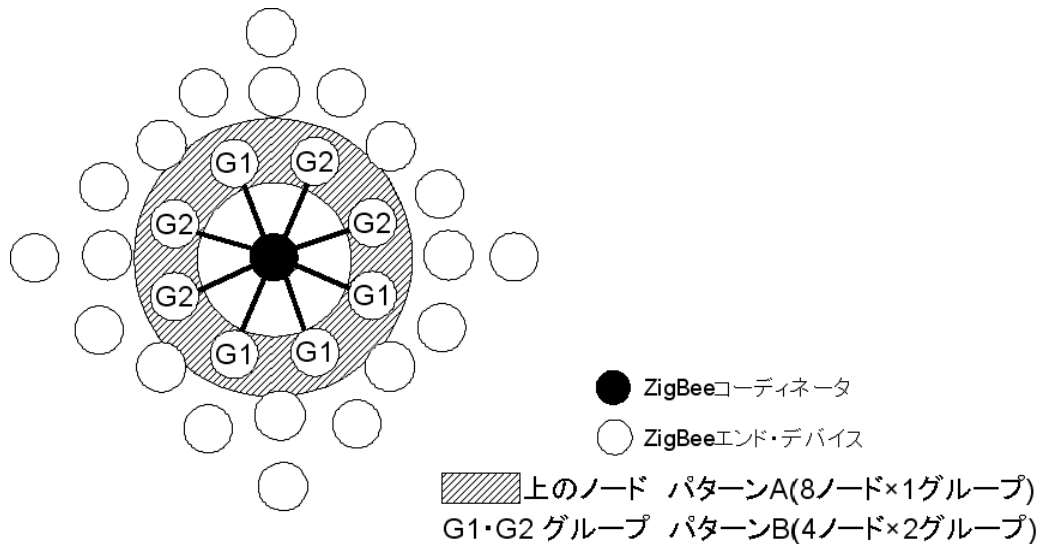


図 6.2: スター型トポロジの評価用グループ

続いて、評価における基本的な項目やサイズの検討を行う。

ZigBee は一般に音声や画像の転送とは違い、家電などの制御を行うシンプルなデータの送受信を行うように設計されている。そのため、物理層・MAC 層を構成する IEEE802.15.4 では一度には物理層の最大ペイロード長の 127 バイトまでしか送信する事が出来ない。一般的に、センサ・ネットワークにおいて通信に利用されるデータ・サイズは 30 バイト前後であると仮定されているため、ここでは物理層の情報として付与されるプリアンブルなどを含め 36 バイトであると想定する。

また、MAC 層レベルの ACK は 11 バイト、プロファイルなどのアプリケーション情報が規定されているアプリケーション・サポート・サブ層の ACK として 25 バイトという固定値を採用する。また通信の形態として、一般にスター型では Beacon モードが利用される事が想定されている。これらの通信の形態は 2.2.4 節で示したであるが、本提案方式ではスター型ではブロードキャストを利用するため 2.2.4 節で示したような形態ではなく、CSMA/CA を利用して全ノードが起動状態で送信可能であると判断した時点でブロードキャストを行う。そこでスター型の評価においては実際にデータ部分を転送した量のみ焦点を当て評価を行う。

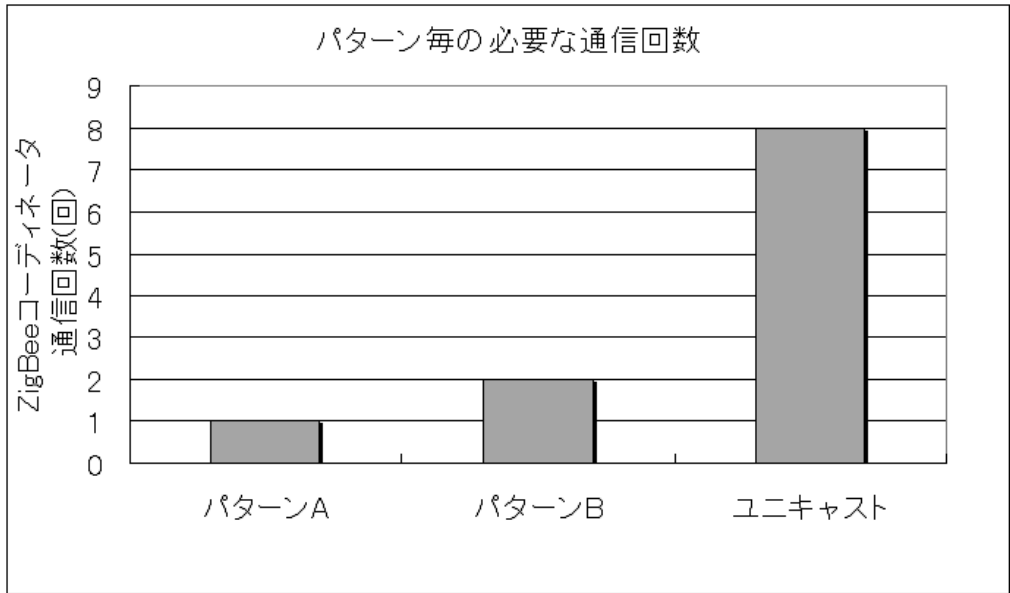


図 6.3: スター型トポロジにおける各ノード必要通信回数

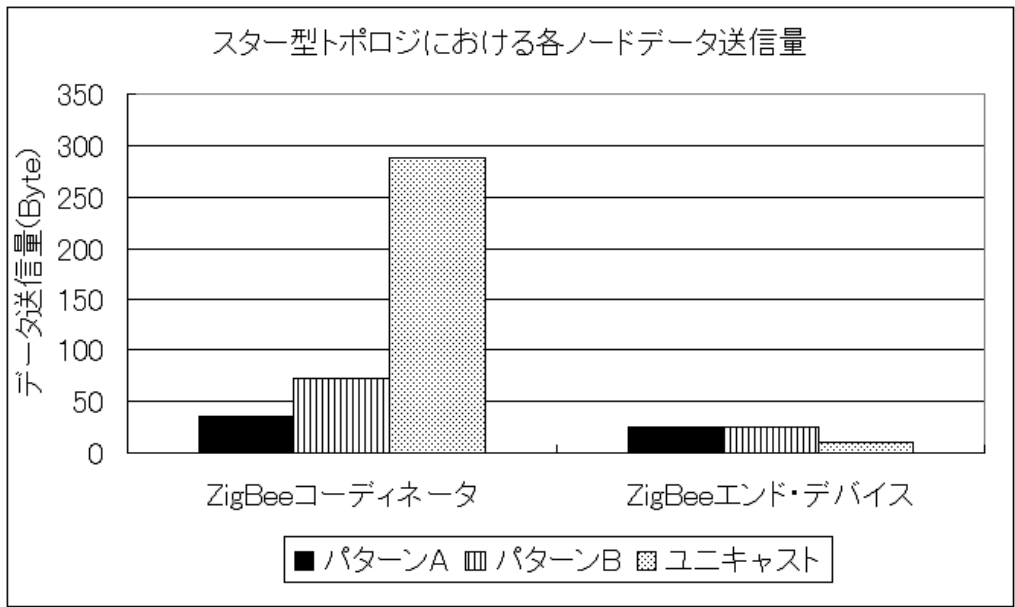


図 6.4: スター型トポロジにおける各ノードデータ送信量

必要な通信回数の考察

図 6.3 より必要な通信回数についてスター型の効率化は、ZigBee エンド・デバイスがどれだけ同じグループに所属しているかによって程度が異なるかが分かる。宛先の該当グループに多数のノードが存在する場合、純粋にそれだけユニキャストより送信回数の面では有効であり、アプリケーションレベルでの実装されているグルーピングより本提案方式は低位レイヤでパケットを破棄できるため、ノードにおけるスタックの処理時間などは削減できると考えられる。また消費電力の面では予め特定の用途に使われるノードなどをグルーピングしておき、グループ宛にデータ送信することでユニキャストより効率化が望める。つまり開発者の意図やアプリケーション仕様に応じて通常のブロードキャストや本提案方式のグループへの送信、そしてユニキャストを併用、あるいは使い分けを行う事により送信回数の大幅な削減が実現でき、消費電力を低減させ長期間に渡りネットワークを活動させる事が出来ると考えられる。

データ送信量の考察

スター型のデータ送信量について、ZigBee コーディネータはブロードキャストを行うことで、パターン A、B においてはユニキャストを行うことで送信量を削減する事が可能となる。しかし、ZigBee においてブロードキャストに対し明示的な MAC 層 ACK は返信しないため、プロファイル等が規定されているアプリケーション・サポート・サブ層においてアプリケーションレベルとして状態を管理し ACK を返す必要がある。そのため、通信を実現させるためという観点では ZigBee エンド・デバイスは上位層で通信状態を管理する枠組みが必要となる。従来のユニキャスト時に発生する MAC 層 ACK よりも多量のデータとしての ACK を送信する必要があることが見て取れる。

テーブル・サイズの考察

スター型トポロジにおいては隣接テーブルのみでグルーピング方式を実現しており、本評価ではテーブル・サイズを表示していない。何故なら本評価のパターン等のグループの規模によらず、ノード毎に隣接テーブル・エントリは作成される。そのためスター型において ZigBee コーディネータが作成する隣接テーブルは PAN 内の ZigBee エンド・デバイス分だけエントリが作成される。つまりサイズ的には隣接テーブル・エントリ毎に少なくともグループ ID 情報の 1 バイトが ZigBee エンド・デバイスのノード数だけ増加するという事となる。

6.1.2 クラスタ・ツリー型トポロジ

トポロジ構成情報の決定

ネットワーク・トポロジ構成するを決定するパラメータとして、最大階層数 L_m 、最大ルータ子ノード数 R_m 、最大子ノード数 C_m を設定する必要がある。ここでは、 $L_m=3$ 、 $R_m=2$ 、 $C_m=4$ として設定を行う。ZigBee コーディネータを除くノード数は最大 28 個まで接続が可能という事になる。シナリオとして、ZigBee コーディネータからツリー上の各所に存在する該当グループ・ノードまでの送信を想定する。

想定するクラスタ・ツリー型アプリケーション

クラスタ・ツリー型トポロジのアプリケーションでは、各部屋単位はスター型トポロジ構成し、中心となる管理場所において ZigBee コーディネータがルート・ノードとして存在する一軒の家にセンサを張り巡らしたクラスタ・ツリーを形成するアプリケーションを考案する。グループとして、各部屋毎の機器、特定の種類の機器(照明機器、電子錠等)を柔軟にグルーピングすることで、家中のどこからでも対象機器をまとめて操作する事が可能となる。例として、ある一部屋から色々な機器の電源 ON の状態で外出した際に、各機器を遠隔操作でまとめて電源 OFF にすることが可能となる。しかも操作する場所は固定でなく家中の何処からでも操作可能である。アプリケーションイメージとしては、図 6.5 のようなクラスタ・ツリー型トポロジを想定している。

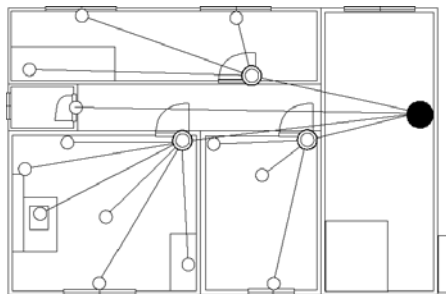


図 6.5: 屋内のクラスタ・ツリー型アプリケーションのイメージ図

- グループの分割方法

提案方式のクラスタ・ツリー型トポロジの場合，グルーピング・テーブルには自分の下流に存在するグループを管理する事でグルーピングを実現している．そのため，評価において効率化を達成するためには，自ノードの下流にどれ程の数の同じグループIDを持つノードが存在するかに大きく依存する．グループ分けのパターンとして以下の4種類を想定する．

1. 末端の特定グループ 16 ノード (パターン A : 16 ノード × 1 グループ) に送信
2. 末端の特定グループ 16 ノード (パターン B : 8 ノード × 2 グループ) に送信
3. 末端の特定グループ 16 ノード (パターン C : 4 ノード × 4 グループ) に送信
4. 末端 16 個のノード全てにユニキャスト送信

通信の形態として，非 Beacon モードにおけるデータ転送を想定する．IEEE802.15.4 では，スター型以外では事実上 Beacon モードは利用できないとされており，親から子へは 2.2.4 節の非 Beacon モードのプル型のデータ転送を利用する．今回提案するグルーピング・テーブルが必要になるのは ZigBee コーディネータ，第 1 階層 ZigBee ルータのみである．何故なら第 2 階層は末端の ZigBee エンド・デバイスのグループ情報を隣接テーブルに登録・管理し，データの中継を行うからである．通信回数の評価に当たり，実際に電波が送信されるデータ，Data-Request，ACK をそれぞれ 1 回とカウントし通信回数を算出した．グルーピングテーブル・サイズについては，エン트리単位 (Next hop と GroupIDs フィールドで構成) 当たり 3 バイトと計算し，該当エントリに GroupID が増加する度に GroupID を構成する 1 バイトずつ増加すると仮定し，純粋に値のみが格納されるサイズについて評価を行った．

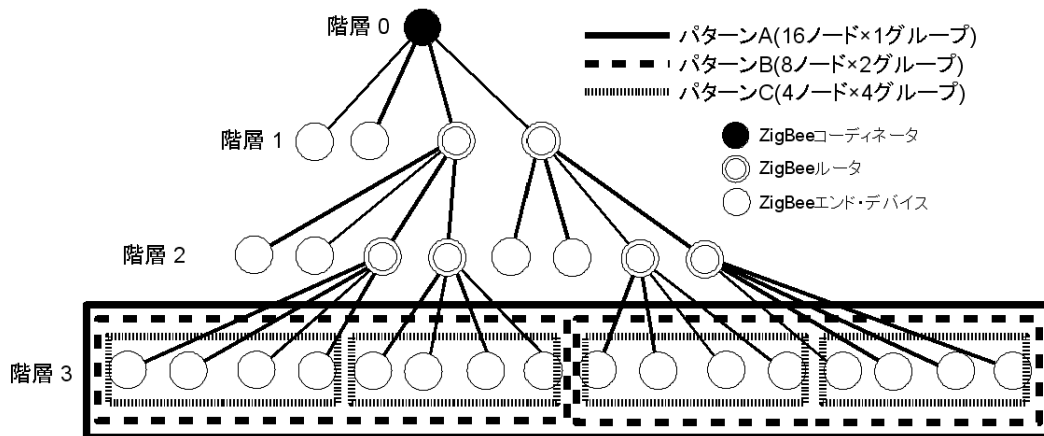


図 6.6: クラスタ・ツリー型トポロジの評価用グループ

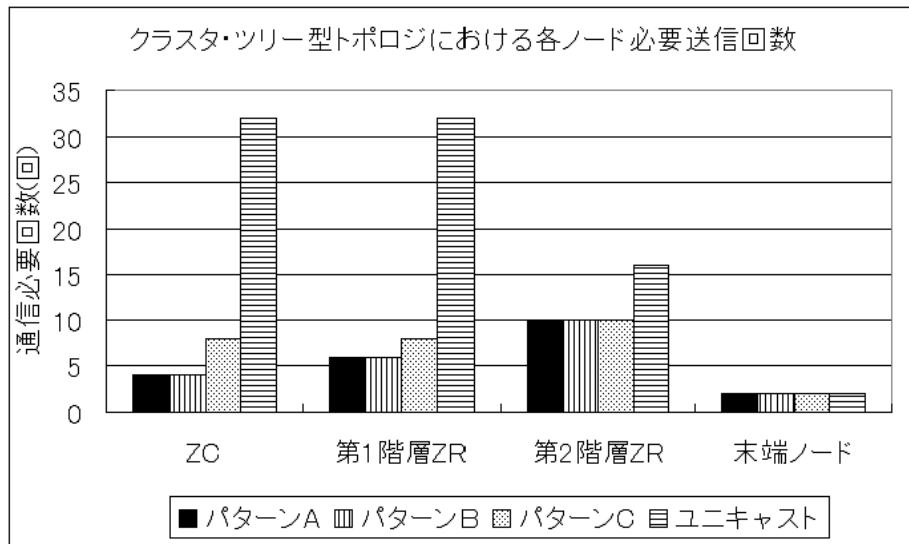


図 6.7: クラスタ・ツリー型トポロジにおける各ノード必要通信回数

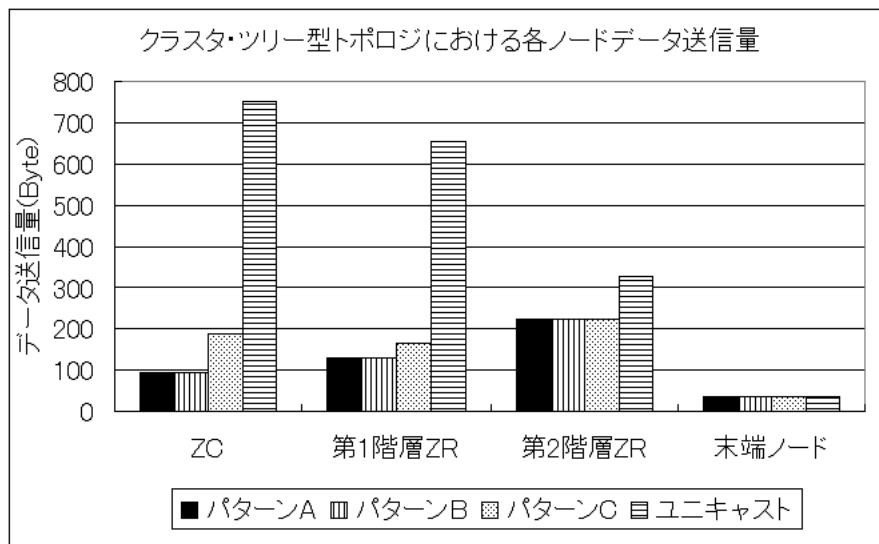


図 6.8: クラスタ・ツリー型トポロジにおける各ノードデータ送信量

必要な通信回数の考察

クラスタ・ツリー型においてはトポロジの構造上、自ノードの下流にどれほど同じグループのノードが存在するかで効率化の度合いが大きく変わってくる。この図6.7において注目すべき点の一つはパターンAとパターンBの際のノードの送信回数は全く等しいという事である。これは今回のグループの分けとグルーピング・テーブルの特徴に注目すれば容易に理由が想像出来る。グルーピング・テーブルは自分と接続されている下流ルータ・ノードよりリーフ側にいかなるグループが存在しているかを示すものである。そしてそれらはデータをルータに転送し、そこで複製・中継を行うことによって効率化を図るものである。つまり今回のパターンA・Bにおいては、ZigBee コーディネータは第1階層の各ルータに必ず1回ずつデータを送信しなければならないためグループの分け方は違いはあれども送信する回数は全く等しいという結果が発生した。つまり、差別化を図るためには一つのノードが同時に複数のグループに所属し、さらに該当グループ・ノードが物理的な配置に拠らないという条件を満たせば違いが発生する。

提案方式のもう一つ注目すべき点としては、階層上位のルートに近いノードほど通信の負荷が軽減されるという点である。こちらもツリーという構造上、ルート付近のノードほどデータの集約が発生し構造上の利点を生かした結果となっている。

データ送信量の考察

データ送信量の評価結果として、グラフの形状は送信回数の評価と非常に似た形になっている。特にユニキャスト時の第1階層ルータは、送信回数についてはZigBee コーディネータと等しいが、実際にデータを送信する回数よりも接続関係を持つノードとの通信を成立させるためのData-Request や確認応答のACK という処理が多いために、ZigBee コーディネータより少ないデータ転送量となっている。

また、提案方式を利用してパターンA～Cの様にシンプルにグループを形成するだけで、各階層のノードにおけるデータ転送量は大幅に削減されるのが見て取れる。これは実際のデータの転送量削減と共に、その通信に必要なData-Request やACKといった処理手順においても転送量が削減したことを示している。

本評価ではシンプルに末端ノードのみにグルーピングを適用したが、実際のクラスタ・ツリー型トポロジにおいては、各階層に存在するZigBee エンド・デバイスにもグループを割り当てることにより、ユニキャストと比較してさらにより良い評価結果を達成する事が出来ると考えられる。

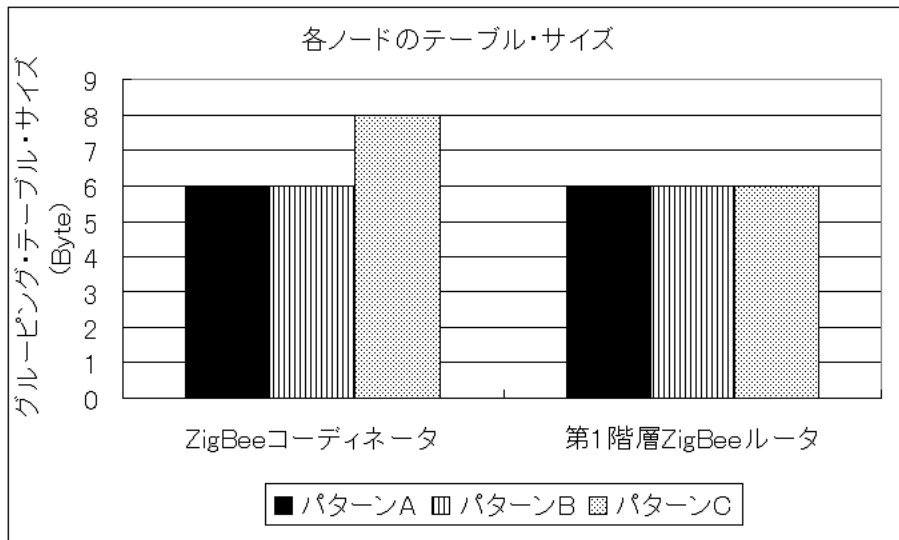


図 6.9: クラスタ・ツリー型トポロジにおけるノードのテーブル・サイズ

テーブル・サイズの考察

クラスタ・ツリー型トポロジで構成されるグルーピング・テーブルのサイズ評価を行った。クラスタ・ツリー型トポロジでグルーピングを実現するのに必要なテーブルとして、グルーピング・テーブルとグループ情報を追加した隣接テーブルが必要となる。しかしスター型で述べた様に、隣接テーブルでは全てのパターンにおいて差が発生しないため、隣接テーブルについてはグラフ化を行わなかった。隣接テーブルでグループ情報を管理するための値としては、従来と比較し、グループ情報を格納する1バイト×存在するエントリ数分増加することとなる。

クラスタ・ツリー型におけるグルーピング・テーブルの特徴として、下流で管理しているグループ情報は必ずその上流に存在するノードが管理する必要がある。そのため、上流に存在するノードが下流に存在するノードのテーブル・サイズより小さくなる事はない。今回の図 6.9 では、一つのグループに関して評価を行ったため差異は見られなかったが、末端ノードが多数のグループに所属する場合を想定すると、そのノードの上流に存在するルート側のノードはその分多量のテーブル・サイズを確保する必要がある。そのため特に多大なRAMが必要であると考えられるルートの ZigBee コーディネータについては、接続された管理PCなどに管理情報を保存しておくという手段を取ることで、容量の制限から開放されると考えられる。

6.1.3 メッシュ型トポロジ

トポロジ構成情報の決定

ネットワーク・トポロジ構成するを決定するパラメータとして、最大階層数 L_m 、最大ルータ子ノード数 R_m 、最大子ノード数 C_m を設定する必要がある。ここでは、 $L_m=3$ 、 $R_m=2$ 、 $C_m=4$ として設定を行う。ZigBee コーディネータを除くノード数は最大 28 個まで接続が可能という事になる。シナリオとして、ZigBee コーディネータからメッシュ型トポロジの各所に存在する該当グループ・ノードまでの送信を想定する。

想定するメッシュ型アプリケーション

メッシュ型アプリケーションとして、図 6.10 の様なビルディング・オートメーションを想定した。ビルディング・オートメーションの特徴として、非常に長い距離の通信が必要ということが挙げられる。そこで長距離での通信を実現するためにマルチホップでの通信は欠かせないものとなる。しかしクラスタ・ツリー型トポロジの様に各ノードへデータを中継する際の経路が一つしか存在しない場合、電波の干渉や中継機器のトラブルによって、宛先までの到達性が失われる。そのような事態を想定し、メッシュ型の特徴である経路に冗長性を持たせることにより到達性を確保する。ビルディングでのグループとして、特定のフロアの照明や特定区画の空調の他に特定方角の遮光機器をグルーピングすることで、人が居なくなったフロアの照明操作、季節・時間帯毎に変わる陽射しの遮光などに柔軟に対応することが可能となる。

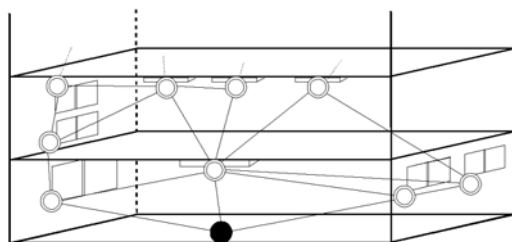


図 6.10: ビルにおけるメッシュ型アプリケーションのイメージ図

- グループの分割方法

提案方式の場合，メッシュ型ネットワークにおいてデータ転送などの処理を効率化達成できるかどうかは，特定グループに所属するノードの配置や経路のリンクの品質の状態に大きく左右される．そこでグループ分けのパターンとして以下のパターンを想定する．本評価においては送信元ノードと宛先グループ・ノードは同一である(4.2.3節参照)とし，完全なグループ拡張ルーティングを形成できるものとする．またリンクの品質は固定値を採用し，ホップ数が少ない経路ほど最適であると想定する．

1. 特定グループの 10 ノード (パターン A : 10 ノード × 1 グループ) に送信
2. 特定グループの 10 ノード (パターン B : 5 ノード × 2 グループ) に送信
3. 特定グループの 10 ノード全てにユニキャスト送信

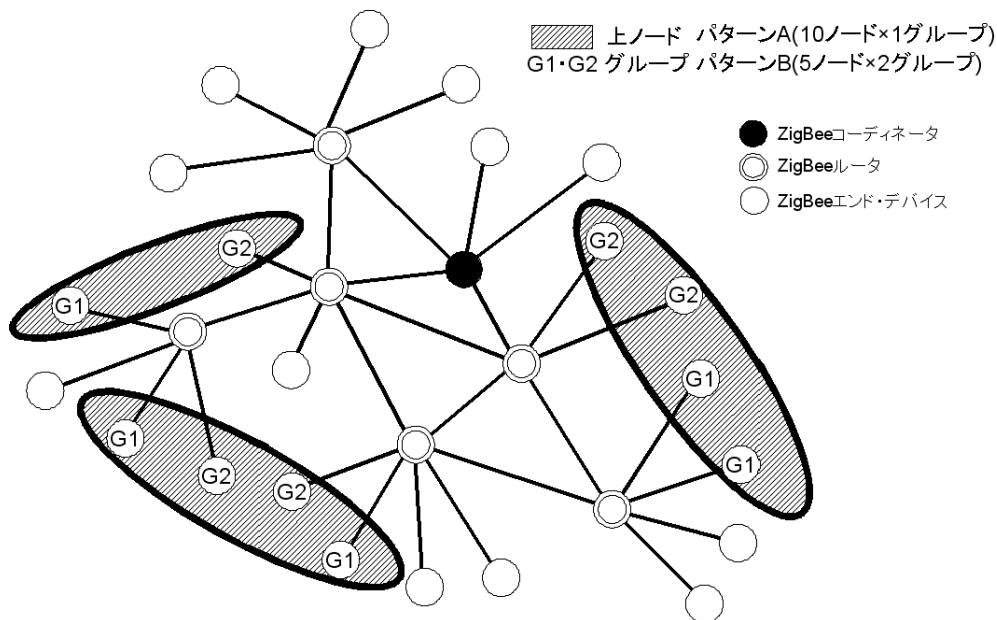


図 6.11: メッシュ型トポロジの評価用グループ

メッシュ型の評価では、PAN 内に存在するどのノードに対象を絞るかを決定する必要があるため、経路探索が完了した時点での最適なルートを以下の図 6.12 に示す。経路選択の方法として前述の通り、全てのリンク品質は固定値を採用し、宛先ノードへ同ホップで到達可能な場合、ネットワーク参加時に親・子関係を構築したノード方向からのルートを最適なものと判断した。図 6.12 はこれらの条件から、メッシュ型トポロジの評価用ネットワークにおいて最適な経路を算出し、今回のグルーピング方式の評価に関連するノードを ZigBee コーディネータ (以下 ZC)、ZigBee ルータ (以下 ZR1~5)、ZigBee エンド・デバイス (以下末端ノード) と名付けることにより区別した。

またメッシュ型ネットワークにおいて評価を行う際の通信方式決定をする。まず通信方式としては非 Beacon モードを想定し、2.2.4 の様に親から子へのデータ転送ではプル型のデータ転送を行い、子から親へはプッシュ型のデータ転送で通信するものとする。そして、ルート探索時のフラッディングについて、全てのルータ機能保持ノードは 1 度の Route Request の中継で全て最適な情報を得る事が出来るものと仮定してある。データサイズとしては、スターやクラスタ・ツリー型と同様にデータ・サイズは 36 バイト、MAC 層レベルの ACK は 11 バイト、Data-Request は 24 バイトとして計算する。そして、今回の提案方式で実現した Multi Route Request/Multi Route Reply は 28/31 バイト (Route Request/Route Reply は 29/31 バイト) として評価を行った。

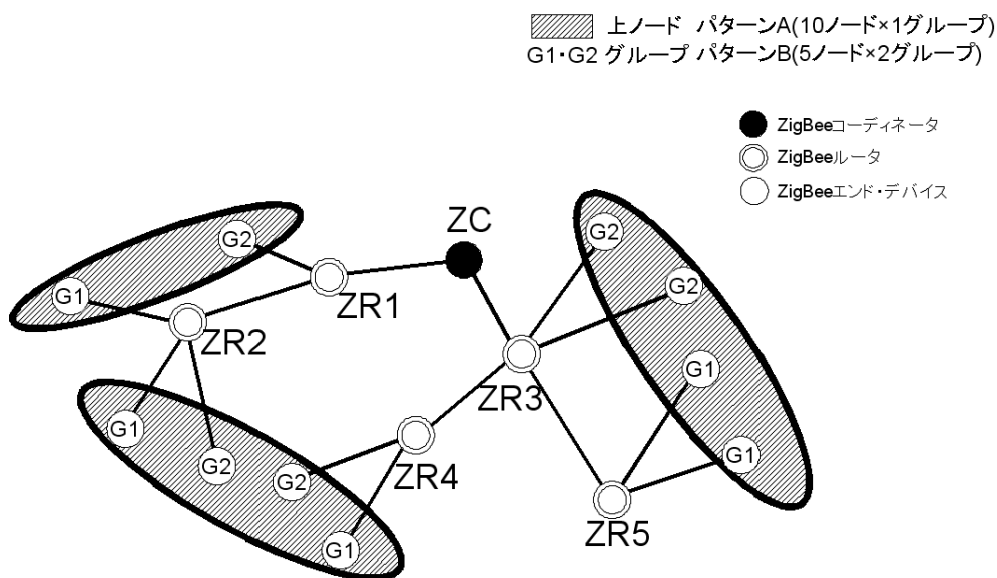


図 6.12: 経路探索が完了した評価用メッシュ型トポロジ

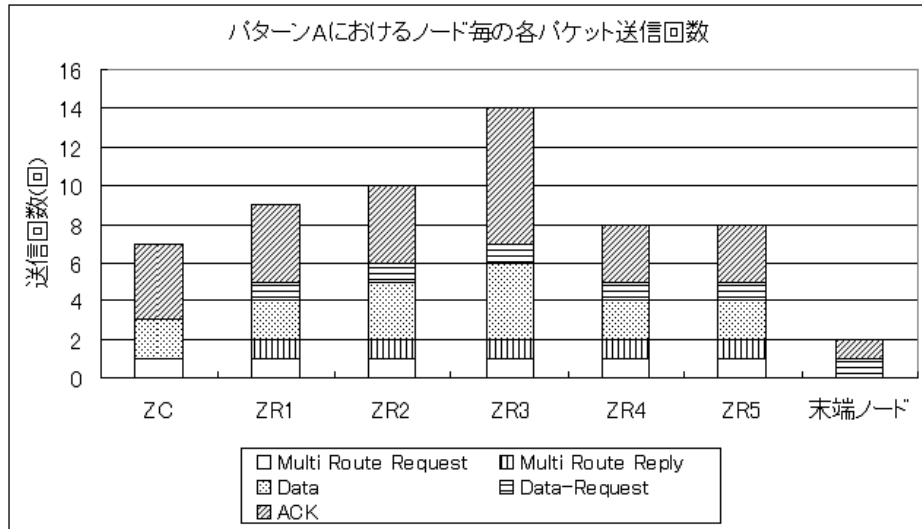


図 6.13: メッシュ型トポロジ (パターン A) の各ノード必要通信回数

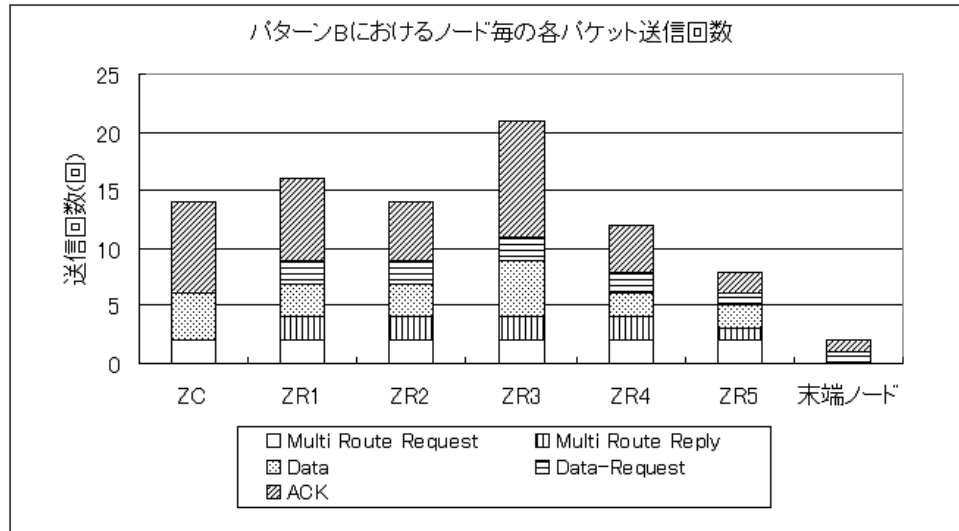


図 6.14: メッシュ型トポロジ (パターン B) の各ノード必要通信回数

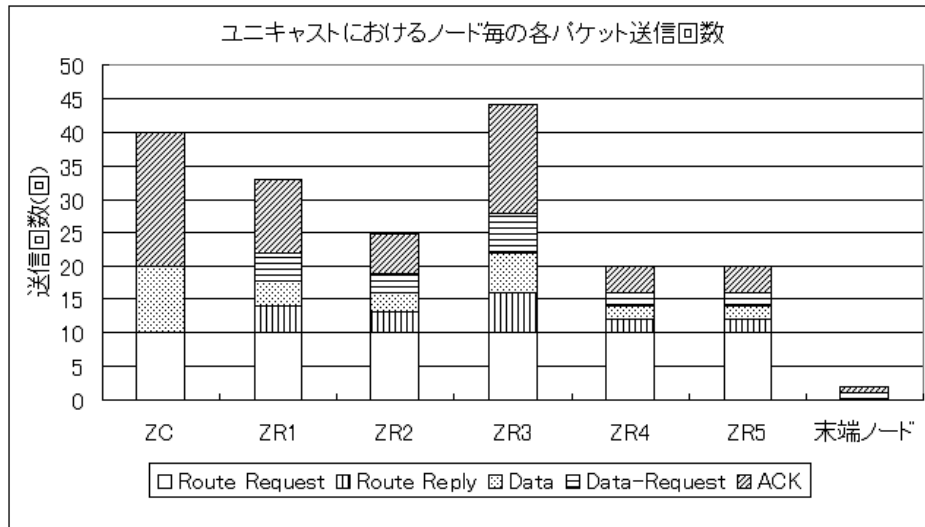


図 6.15: メッシュ型トポロジ (ユニキャスト) の各ノード必要通信回数

必要な通信回数の考察

メッシュ型はクラスタ・ツリー型と違い、同じ階層でもグループの構成法によりパケット毎の通信回数が大幅に変わるため、関連するノード ZC, ZR1~5, 末端ノードについてパケットの種類についても評価対象とした。各パターンの通信回数の図から読み取れる事として、ユニキャスト時はクラスタ・ツリー型トポロジとは違い、必ずしもルートに近いノードが通信回数が多いとは一概に言う事は出来ない。提案方式の場合、経路探索の結果、自分とリンク構築しているルータ・ノードがどれだけ存在しているかとそのノードの下流に存在するグループ ID の組み合わせに通信回数は大きく依存している。これにより、多くのグループ情報を管理するルータ・ノードは多数の Multi Route Reply を受信して経路を確保する必要があり、消費電力の面でも大きな負荷が掛かると考えられる。本評価の ZR3 がそれに該当し、多くの中継ノードと多種のグループを管理する結果、送信元ノードの ZC よりも通信回数が必要とされている。

また本評価においては、全ての ZigBee ルータは一度の Route Request で最適な経路を把握するという仮定であるが、実際にはフラッディングによるブロードキャストフレームが PAN 内に多く発生することとなり、ネットワーク全体が輻輳状態を引き起こすことが考えられ、結果として CSMA/CA における待ち時間の増大やリトライ等の再送信が発生しやすい。つまり、ユニキャスト時では短い周期で複数回の経路探索を行うのは現実的ではないが、提案方式を採用する事で複数の経路を同時に探索できるため効率化を図る事が出来ると考えられる。

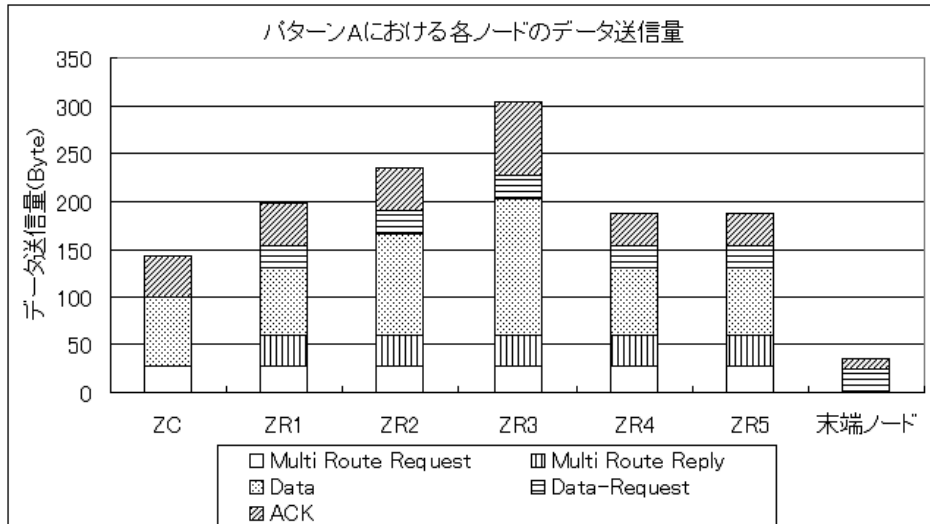


図 6.16: メッシュ型トポロジ (パターン A) の各ノードデータ送信量

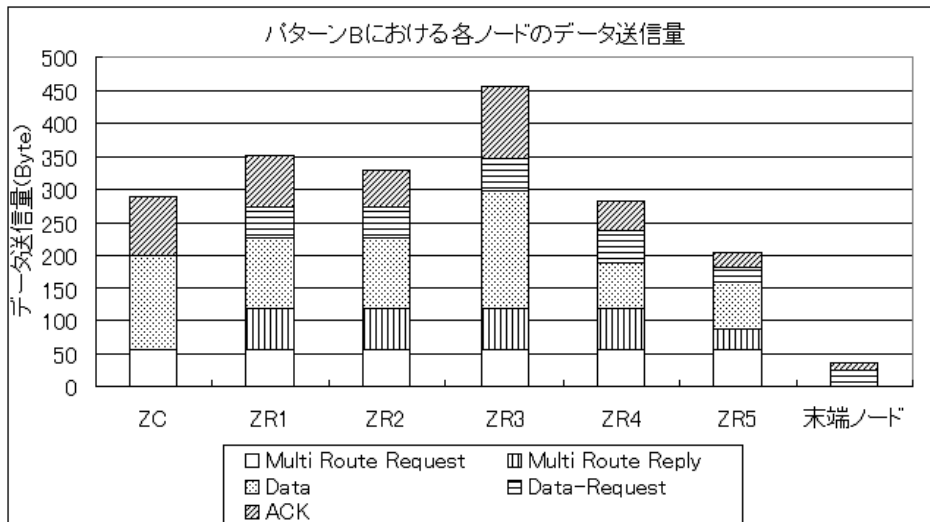


図 6.17: メッシュ型トポロジ (パターン B) の各ノードデータ送信量

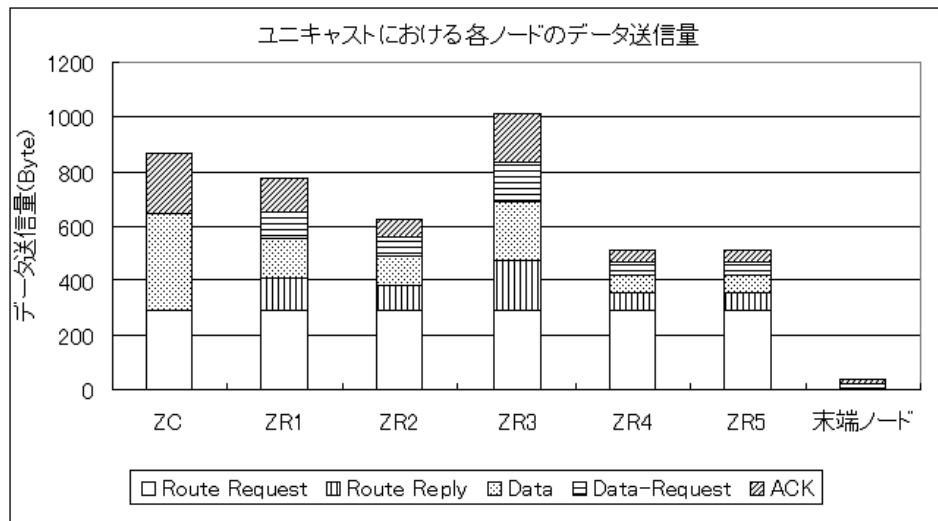


図 6.18: メッシュ型トポロジ (ユニキャスト) の各ノードデータ送信量

データ送信量の考察

メッシュ型におけるパターン A や B の提案方式では、全体送信量に占めるデータの量が大きい。これは本提案方式を適用し適切案ルータが適切な方向へとデータの複製・中継を行ったのが理由だと考えられる。これにより、通信処理に必要な送信量などが削減されたと考えられる。それに対してユニキャストでは、経路探索時の Route Request などが大きい割合を占めている。特にメッシュ型通信回数の評価考察で述べたとおり、フラッディングを用いた経路探索を幾度と無く繰り返す必要があるため輻輳が発生しやすく、実環境では再送信のを行う必要性が高い。

メッシュ型においてグループに対してデータ送信を行うもう一つの方法として、特定グループのノードに対してデータ配信を行う時は、全てブロードキャストを利用するという方法が考えられる。しかしこれは、FA¹等の頻繁に通信が行われるアプリケーション仕様では、フラッディングを理由とした輻輳や消費電力などの問題で致命的であり、このような場合には提案方式を用いて該当ノードをグルーピングし、一度経路探索を行い経路情報を確立した後、ユニキャストを用いて定期的に通信を行い、テーブル情報を維持し続けるという方法が得策であると考えられる。逆に言えば、HA²における家電の制御等グループ対象の通信が極稀にしか行われないようなメッシュ型トポロジにおいては、経路探索のための手間や、経路情報維持のための RAM などの資源の点で非効率だといえるかもしれない。

¹Factory Automation

²Home Automation

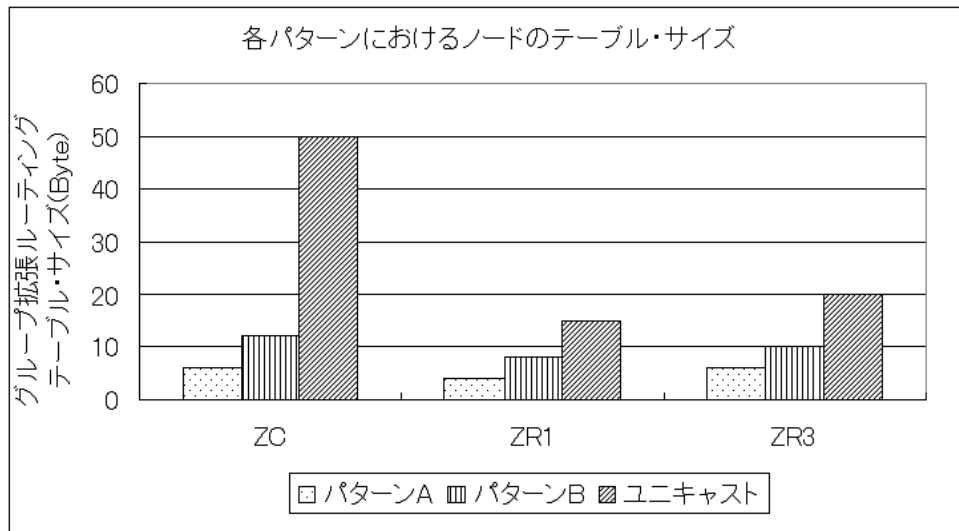


図 6.19: メッシュ型トポロジにおけるノードのテーブル・サイズ

テーブル・サイズの考察

メッシュ型トポロジにおけるグループ拡張ルーティング・テーブル・サイズについて各パターンについて評価を行った。図 6.19 では、評価用のトポロジにおいてグループ拡張ルーティング・テーブルが構築されたノードについて列挙してある。ZR2, 4, 5 においては、隣接テーブルにのみ末端ノードのグループ ID を登録するため、本評価においては取り扱わない。

まず図 6.19 において最も注目すべき点は、提案方式であるパターン A、パターン B とユニキャスト時における ZC のテーブル・サイズの違いである。ユニキャストでは、ルーティング・テーブル情報として一つの宛先に対し一つのエントリを作成しなければならない。それに対し、提案方式では宛先グループに対し送信すべき方向を幾つも管理することができるため、効率良くセンサ・ボードの RAM を利用することが可能となる。

メッシュ型のトポロジにおいては、ノードの配置やグルーピングの仕方によって、効率化の度合いは大きく変化すると考えられる。経路探索後に送信元をルートとしたツリーが形成されるため、ノードの配置やグループへの所属による経路の集約が、通信回数やデータ送信量と同様に各ノードが保持するテーブル・サイズに大きな影響を与える。

配置やグループの点で効率化が図れなかった際にも、提案方式を用いてグループに所属するノードへの経路探索を同時に行えることは、輻輳問題や処理時間の大幅な改善に繋がると考えられる。

6.2 アプリケーションからトポロジ・グルーピングの検討・考案

6.2.1 FA アプリケーションの概要

今回の評価で利用するアプリケーション例としてFA³を想定する。FAとは、工場内のライン製造の自動化や24時間体制の監視にセンサ・ネットワークを利用し、人間の手間を削減し金銭や労力等のコストの効率化を図ることを目標としている。一般的にライン作業で製造等を行っている工場においては、時期によって製造する部品が異なる場合がある。そのため、製造する部品によりライン構成が異なる場合が存在する。そこで、本提案方式のグルーピング技術を利用する事により、同じ種類の作業を行う製造機等をグルーピングすることにより工場内の管理性が向上することが考えられる。本評価では、工場の構造や構成によりどの形態のトポロジが最適であり、どのようにグルーピングを適用していく事が効率化が図れるという事を念頭に検討・評価を行う。本評価に利用する工場の構造は図6.20の通りであると想定する。

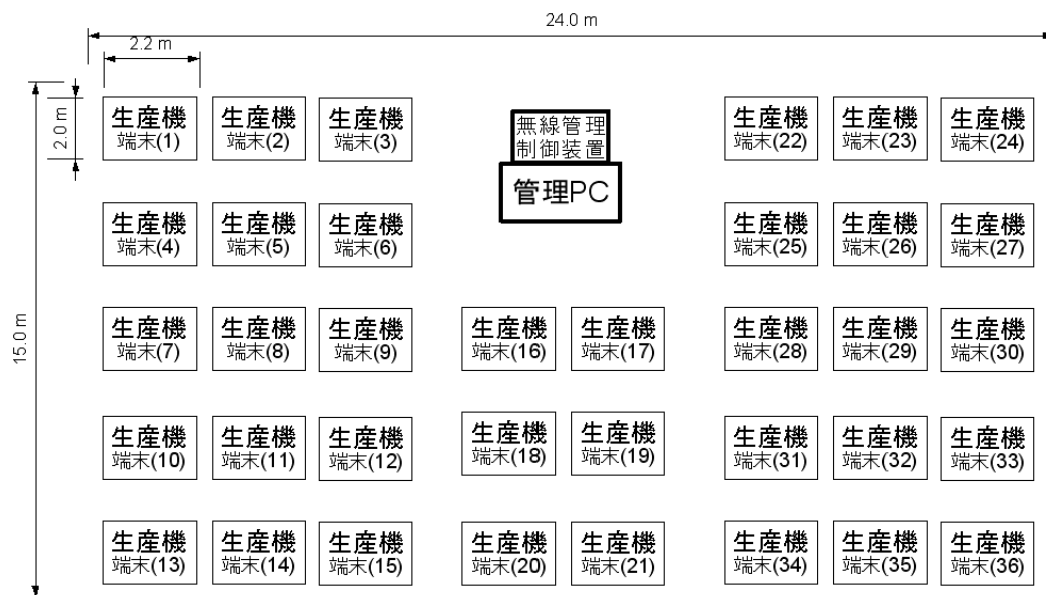


図 6.20: FA を想定する工場の構造

³Factory Automation

6.2.2 FA の構成・トポロジ・グルーピング考案

このシステムの構成として、中央にある無線管理制御装置に接続された PC によって固有の識別子を割り当てられた製造機の部品の製造状況 (製造ミス発生を検知) を把握しているものと想定する。製造ミスの発生を徹底的に減少させるには、ライン上にてミスが発生した際に関連するラインを一時停止させる事が必要であると考えられる。そこで、エラーが発生した製造機からの製造ミスを検知した場合、そのラインに関連した該当グループに所属する製造機に対し、停止命令を送信する。これにより、関連した製造機のみ動作を止めることが可能となる。

次いでアプリケーションの構成として、中央にある無線管理制御装置から各ノードに状況を尋ねるクエリを送信し、それを受信した各ノードから応答が返ってくるというシステムを想定している。これは本提案方式での送信元から該当グループに対し一斉に送信を行えるという利点を生かし、管理 PC から一斉に送信するシナリオを想定した。

トポロジはスター型とクラスタ・ツリー型、そしてメッシュ型トポロジにおいて考案する。工場の範囲として ZigBee を利用すれば管理 PC から 1 ホップもしくは 2 ホップ程度で全ての製造機への到達性を網羅する事が可能であると考え、各トポロジにおいてネットワークの設計を行った。スター型においては、論理デバイスとして ZigBee コーディネータと ZigBee エンド・デバイスを、クラスタ・ツリー型トポロジとメッシュ型トポロジにおいては ZigBee コーディネータ、ZigBee ルータ、ZigBee エンド・デバイスによって構成されるものとする。以下では、上記の 3 トポロジについてグルーピングの差による利点・欠点の考察を行う。

スター型トポロジ

スター型トポロジにおいては、全て無線管理制御装置から1ホップで通信可能であり、かつ提案方式でネットワーク層アドレスのブロードキャストを活用するということから、利点としては物理的な配置に全く依存しない形態のアプリケーションが作成できる。これは工場内の製造機の配置が規則正しくない場合に非常に有効である。クラスタ・ツリー型トポロジでは前節の各トポロジの評価結果より、特定中継ノードの子ノードが同一グループへ所属すると効率化を達成できるという点から、物理的にある程度近隣のノードでクラスタを形成していかなければならない。そのため、スター型としてはグループ形成の自由度という点は大きな利点である。逆に欠点として、工場などの電波状況が悪いと想定される場所においては、ある程度距離が離れたノードとの通信の到達性が難しいという点があげられる。また工場内では電波の反射や干渉によりこれらの要因は発生しやすいと考えられる。また、末端ノードの数が增大するほど送信の衝突回避のための待ち時間などが増大するものと考えられる。図6.21において、スター型の例としてグルーピングを行った例を示した。こちらの図ではグルーピングを行ってある様子を示すため特定の近距離のノードをまとめてグルーピングを行ったが、前述の通り位置的な要因に依存しないため、各グループは物理的な位置のどこに存在しても問題は無い。このことから、図では Group A ~ Group C にグループを分割してあるが、グループ数が増大することによるデメリットはほとんど無いため状況に応じた柔軟なグルーピングが容易に実現可能である。

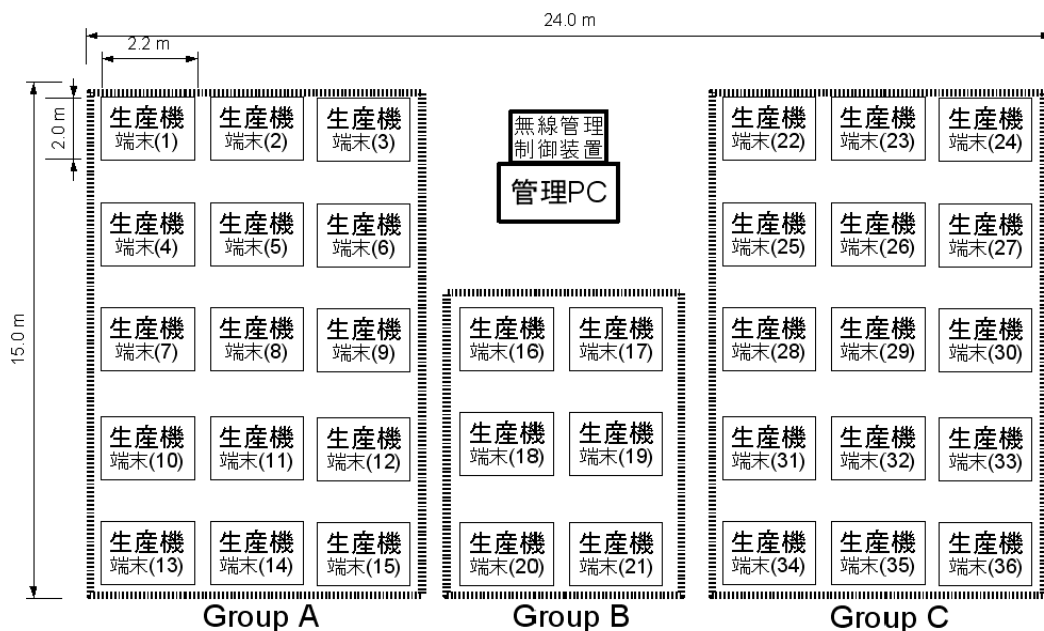


図 6.21: スター型トポロジを想定した FA アプリケーション

クラスタ・ツリー型トポロジ

クラスタ・ツリー型トポロジにおいては、無線管理制御装置は1ホップ通信だけでなくマルチホップ通信を実現できる。提案方式においては、クラスタ・ツリー型トポロジに対して自分の下流にどのようなグループIDが存在するかをグルーピング・テーブルとして保持し、宛先グループ宛の packets に関してユニキャストを用いて適宜複製・中継を行いつつルーティングを実現するというものであった。本評価では、工場内の特定の領域毎に並列に同じ製造機が存在するようなパターンを想定したグルーピングを図6.22に示した。図6.22では、6台の製造機で一つのラインとして形成しており、それぞれのラインで同一(又は異なる)部品を製造していると想定する。各ラインのいずれかにおいて異常を検知した場合、スター型と同様に管理PCから制御停止信号を送信する事で該当ラインを停止させることが可能となる。一般的にある程度近い距離のノードでスターを形成し、クラスタ・ツリーを形成することから管理PCに接続された無線管理制御装置が指向性アンテナを保持していれば、該当クラスタ方向を主目的とした通信が可能となる。また、通信状況の悪い工場内で長距離の通信を行うことは現実的でない想定すると、近距離間で複数回通信して宛先に近づけた方が信頼性は高いと考えられる。しかし、中継ノードにトラブルが発生した際の到達性の問題やノード一つ当たりの処理が増加するため、管理PCや中継役を果たす製造機において、電力消費や送受信の転送量が純粋なスター型と比べ増大すると考えられる。そして、各ノードに複数のグループIDを割り当てることは、上流のノードにおいて多くのグループ情報を管理する必要があるため、RAM容量は圧迫される。

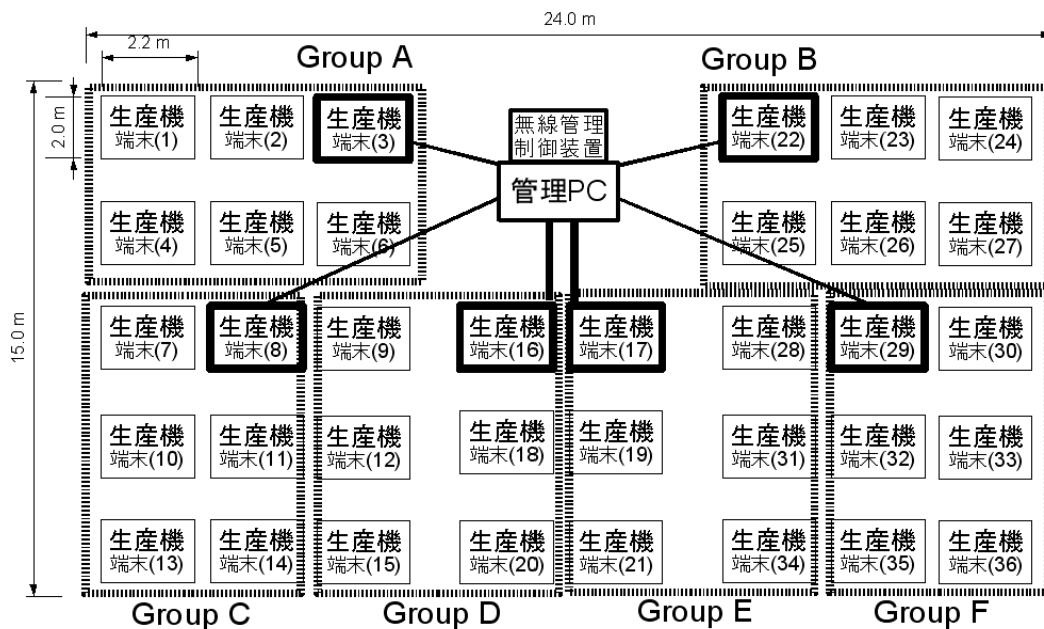


図 6.22: クラスタ・ツリー型トポロジを想定した FA アプリケーション

メッシュ型トポロジ

メッシュ型トポロジにおいて最も優れている点は、宛先までの経路を複数持つことができるという点である。今回のアプリケーションで想定した工場のような環境では、電波の品質低下による通信不能状態や何らかの原因によって無線機が動作しなくなるといった事態を想定する必要がある。そのような場合、メッシュ型トポロジにおいては動作しなくなったノードの代替となるノードの存在により、センサ・ネットワークシステム自体の動作を停止させずに修理・確認を行えるというメリットがある。具体例として示したのが図 6.23 である。この図では各グループ毎に中継機能を保持したノードが 2 台ずつ存在している。1 台中継ノードは末端のスター型を構成し、もう 1 台のノードは通常動作を行いつつ、バックアップ用として動作を続ける。スターを構成しているノードが不慮のトラブルにより活動できなくなった際に、バックアップ用のノードがスターを代替管理できる。この概念はクラスタ・ツリー型トポロジにも適用できるが、末端ノードが ZigBee エンド・デバイスで同じネットワーク構成パラメータだと想定すると改良を加えなければ実装は不可能である。つまり利点としては障害への対応しやすいトポロジだと考えられる。グルーピングについて考慮すると、効率化を図るためにはノードはクラスタ・ツリー型と同様、配置に依存する。そのため、ある程度まとまったラインをグループ化することで柔軟な制御は行えると考えられる。しかし、メッシュ型のデータ送信量の評価からも分かるように柔軟に経路を選択・維持する分、データ送信量等は増加するため各ノードにおける負荷要因は他の 2 つと比べて最も大きいと考えられる。

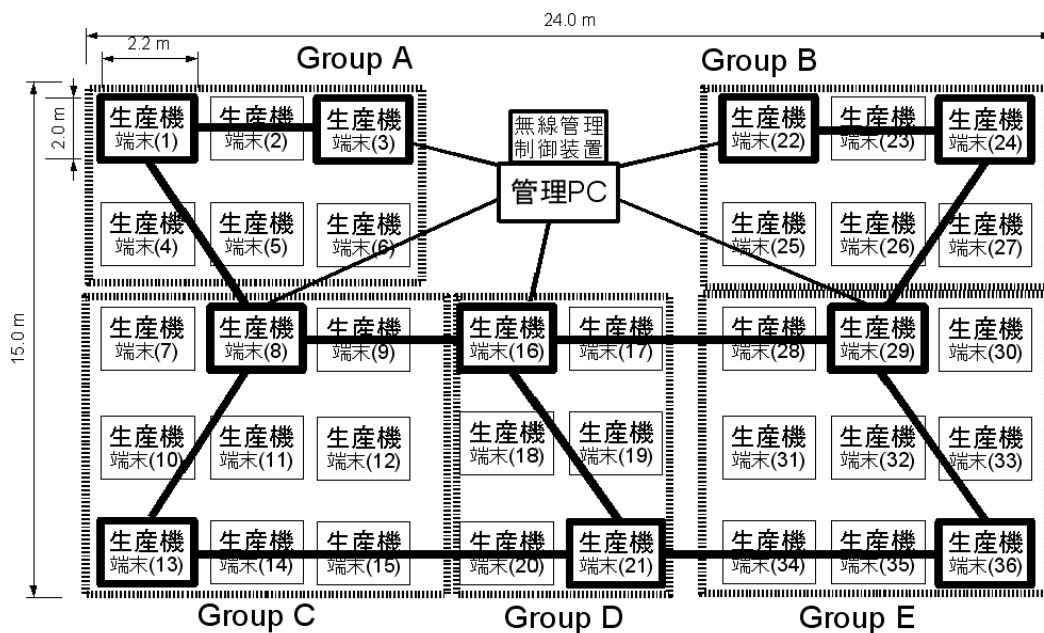


図 6.23: メッシュ型トポロジを想定した FA アプリケーション

6.2.3 BA アプリケーションの概要

続いて先程とは違うアプリケーションを想定する．図6.24はBA⁴を実現するためのアプリケーションである．BAとは，ビルディング内の空調，照明，遮光用であるブラインドなどの操作を自動化することにより，ビルディング内の環境を快適に保つことが出来るという点だけでなく，各機器を効率よく操作することで省エネによる電気代の削減や手間を省くことによる人件費の削減といった目的を達成することが出来る．ビルの様な巨大な建築物内においては，各機器を操作する場所が限定されていると人間がその場所まで移動した後，機器を制御する必要がある．機器操作・制御に適したプロファイルを持ち，場所に拠らない無線通信を用いて機器操作を実現できる ZigBee は，BA という分野において非常に期待されている．本評価では，前のFAアプリケーションの時と同様にビルディングの構造に対してどのような形態のトポロジが最適であり，どのようにグルーピング方式を利用して効率化を図ることが出来るかという事を念頭に検討・評価を行う．

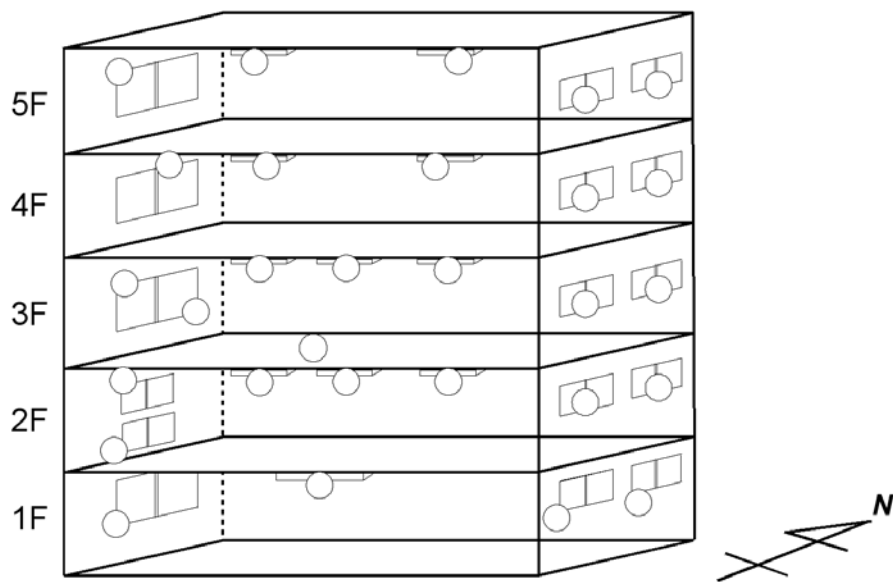


図 6.24: BA を想定するビルの構造

⁴Building Automation

6.2.4 BA の構成・トポロジ・グルーピング考案

アプリケーションを構成するビルの情報について説明を行う。図 6.24 では、対象建築物は 5 階建ての建物であり、東西に窓が存在すると想定している。窓には遮光用のブラインドが取り付けられていて、センサから常に照度の情報を取得しているものとする。そしてブラインドに対して操作命令を送信することで、ブラインドによる遮光具合を制御することが出来る。また、各階のフロアの天井にはエアコンが設置されており、センサによる温度や湿度の監視と共にこちらも ZigBee を用いて遠隔で制御を行える。今回の BA アプリケーションの制御対象機器はこの 2 種類であり、管理制御装置においてセンシングの値が常時観測可能で同一 PAN 内に存在するノードであれば、どこからでも対象機器を制御できるものとする。処理の流れとして、管理制御装置からノードあるいはグループ宛に操作命令を発信し、機器が管理制御装置に対し応答を返すという処理を想定している。

配置に関する特徴として、FA 時の管理装置と違い管理制御装置の位置は特に指定されていない。つまり今回の BA アプリケーションの構成に際し、トポロジの形態やグループの割り当て方だけでなく、管理制御装置の位置情報も考慮する必要がある。

ノード数にはついては、数百程度存在していると想定し、トポロジはスター型、クラスタ・ツリー型、メッシュ型を利用する。構成した各トポロジにグルーピングを適用し、BA において提案方式がどのような利点あるいは欠点をもたらすか考察を行う。

スター型トポロジ

スター型トポロジにおいては、図 6.25 の様なイメージとなる。管理制御装置と各機器の間の通信は 1 ホップで行わなければならないが、ビルのような巨大な建物を想定した場合、全て ZigBee コーディネータからの 1 ホップで通信を行うのは近距離無線通信規格の ZigBee においては必ずと言って良いほど不可能である。仮に全てのノードの中央にあたる位置に配置したとしても、通信距離が短い ZigBee においては不可能だと考えられる。

通信範囲の問題以外にも、ZigBee コーディネータは自分と接続関係を持つ全てのノードの隣接テーブルを保持しなければならない。本アプリケーションの仮定においては、数百単位のノードの情報を一元管理することは ZigBee コーディネータに非常に負荷が掛かる事であると容易に判断することが出来る。これらの問題より、スター型トポロジを利用して BA アプリケーションを構築するのは難しいと判断した。

グループの割り当てについては、エアコン、ブラインドといった特定の種類の機器でのグルーピング、特定のフロアに存在する機器をグルーピング、特定の方角の機器をグルーピングするといった種類が考えられるが上述の通り、通信を実現することが難しいと考えられるため、グループに関する考察はクラスタ・ツリー型やメッシュ型トポロジの項目において示す。

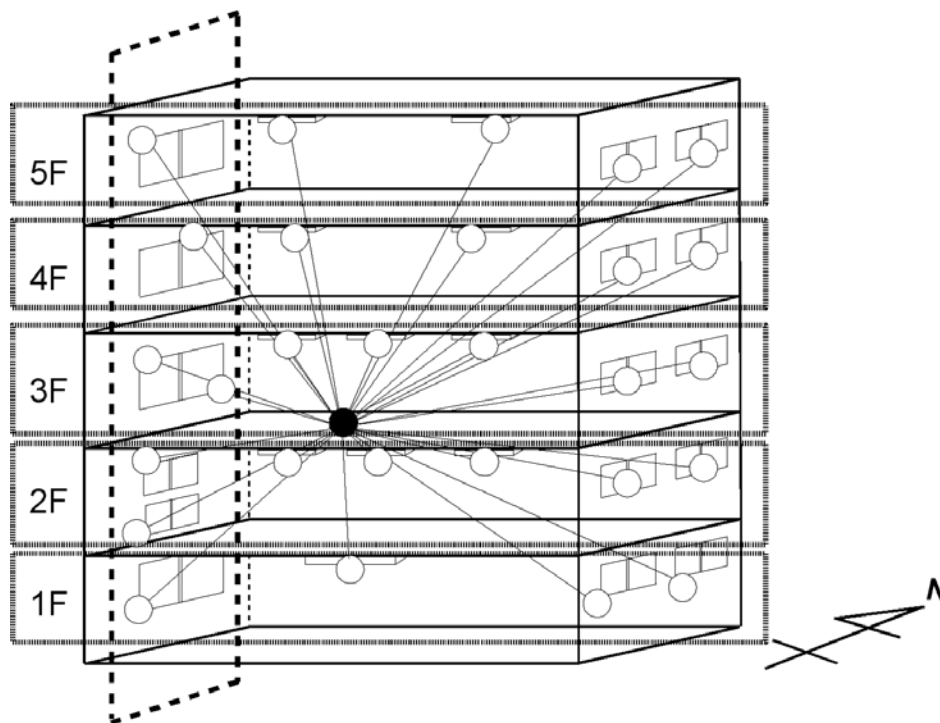


図 6.25: スター型トポロジを想定した BA アプリケーション

クラスタ・ツリー型トポロジ

クラスタ・ツリー型トポロジにおいては、スター型では実現できないマルチホップ通信を利用することによって本アプリケーションを実現することが出来る。アプリケーションの構成イメージは図 6.26 の様になる。クラスタ・ツリー型におけるグルーピングとして、東西の両方角に存在するブラインドについてグルーピングを行った。これらの二つの方角に存在するブラインドは全く同じ機器であり、かつ全く同じ機能を持っているが、別のグループに分けて管理を行う理由として、方角の違いにおける活用方法の差を考慮した点が挙げられる。高層のビルでは、夕方などの特定の時間帯に太陽の西日が差し込むといった事態が頻繁に発生する。そのような場合、ブラインドうち特定方角グループのブラインドを一括操作することで、同じブラインドの中でも様々な分類を表現することが可能となる。ノードの配置はツリーにおいて効率化を達成する大きな要因であることから、同じ方角に存在するノードをグルーピングする事はツリーにおいてある程度の効率化を実現できると推測できる。問題点として、クラスタ・ツリー型トポロジの性質上、ルーティングの経路が固定であるため、あるルータの故障や電波干渉などが発生すると、宛先ノードまでの到達性が失われるという点が挙げられる。

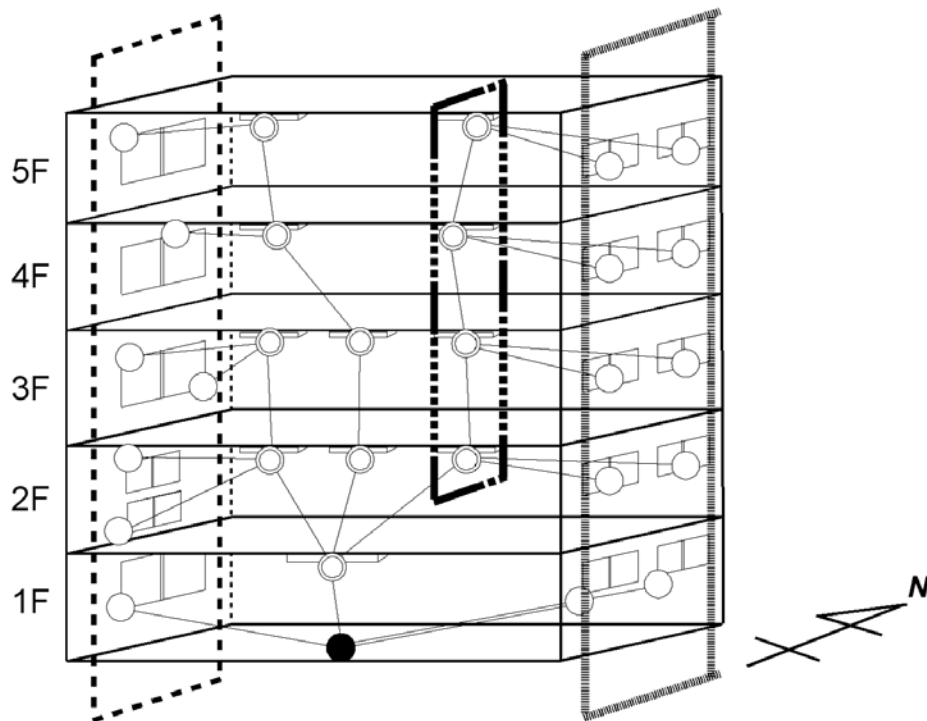


図 6.26: クラスタ・ツリー型トポロジを想定した BA アプリケーション

メッシュ型トポロジ

ビルディングの様な大規模なネットワークにおいては通信障害の要因が多数存在すると考えられるため、クラスタ・ツリーの様にルーティング経路が固定なトポロジでは信頼性に欠ける。そこでメッシュ型トポロジを用いることで経路に冗長性を持たせることが出来る。グループの構成としては、各フロアに存在する機器でグルーピング、特定方角に存在する機器でグルーピング、機器の種類毎にグルーピングなど、様々なパターンを考えることが出来る。特にフロアや方角といったグルーピングにおいては、位置情報に関連するグループ分けであるため、データの送信時にある程度似た経路を経由する可能性が高く、送信回数やデータ送信、必要テーブル・サイズといった点で効率化を図ることが出来ると思われる。しかしノード数が数百単位であるということとを考慮すると、経路探索時のフラグディングは決して効率の良いものとは言えない。また、クラスタ・ツリー型のグルーピング・テーブルは値を常時保持することが出来るために適宜利用可能であるが、メッシュ型の経路探索で作成されたグループ拡張ルーティング・テーブルは、一定時間利用されないと無効になるという特徴があった。今回のBAを実現するアプリケーションは、定期的に通信を行うFAアプリケーションと比べれば、ZigBee コーディネータから発する操作命令の通信頻度は少ない。そのため、要求が発生する度に経路探索を行わなければならないという欠点があるだろう。

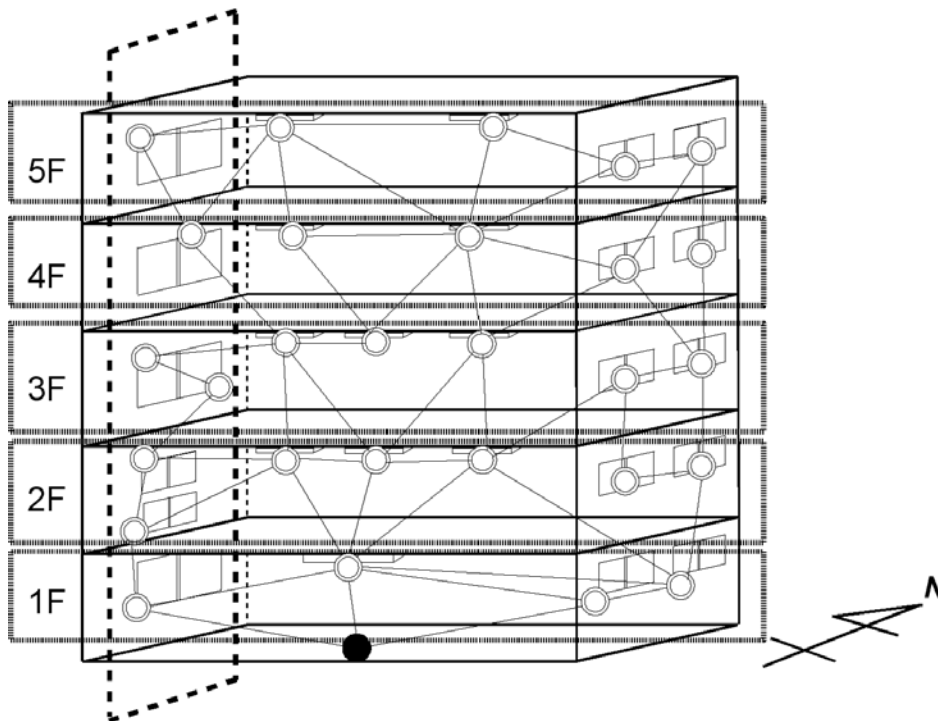


図 6.27: メッシュ型トポロジを想定した BA アプリケーション

6.3 メッシュ型トポロジの経路探索待ち時間の検討

三つ目に、メッシュ型トポロジにおいて経路探索を行う際の送信待ち時間について検討する。送信待ち時間の各パラメータについては 4.3 を参考にしてほしい。ここでは以下の項目に焦点を当て、評価を行う。

- ネットワーク構成の違い・スタック処理時間による待ち時間
- リトライ発生回数の違いによる待ち時間

まず、ネットワーク構成について検討を行う。ZigBee Specification[3] によると、PAN を構成するパラメータの一つである最大階層数を規定する `nwkcMaxDepth` は通常値として `0x05` を採用している。これは最大で 5 階層までのネットワークを構築する事を示している。また、待ち時間の評価のもう一つの大きな要因である J: Multi Route Request 中継ランダム遅延は 4 ~ 128[ms] までの値をとる。そこで以下の 4 種類を用意し、各パラメータを元に構成されるネットワークにおいてスタック処理時間を含めた送信待ち時間に関する評価を行った。

1. J: MRREQ 中継ランダム遅延 64[ms] , D: ネットワーク階層数 3
2. J: MRREQ 中継ランダム遅延 128[ms] , D: ネットワーク階層数 3
3. J: MRREQ 中継ランダム遅延 64[ms] , D: ネットワーク階層数 5
4. J: MRREQ 中継ランダム遅延 128[ms] , D: ネットワーク階層数 5

評価に当たり経路探索について最大でどれだけの時間待つことができるかを調査した結果 ZigBee Specification[3] において経路発見のタイムアウトは 10 秒⁵と設定されている事が判明した。つまりデフォルトで最大 10 秒まではルート発見要求が有効であり、その間は送信元は待つことが可能という事を示している。また、Multi Route Request の送信失敗時のリトライに関しては、Route Request と同様に 254[ms]⁶として評価を行った。次に各項目の結果について示す。

⁵`nwkcRouteDiscoveryTime=0x2710[ms]`

⁶`nwkcRREQRetryInterval=0xfe[ms]`

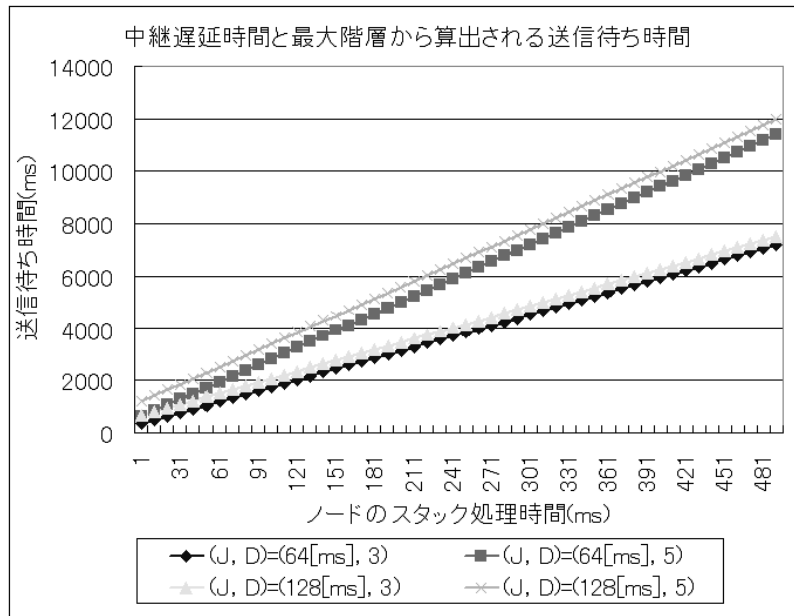


図 6.28: 拡張経路探索時のノード処理時間と送信待ち時間の関係

ネットワーク構成とスタック処理時間からの考察

図 6.28 の評価は、横軸に各ノードのプロトコルスタック処理時間を取り、縦軸に 4.3 の計算式の値が算出されている。まず、ルート発見タイムアウトを考慮した際に待ち時間の合計は 10 秒を超えてはならないという前提条件が存在した。そこで 4 パターンの構成を比較すると、 $(J, D)=(64[\text{ms}], 5)$ の場合と $(J, D)=(128[\text{ms}], 5)$ の場合は各ノードのプロトコル・スタックが $400[\text{ms}]$ 程度まで許容されるのに対し、通信ホップ数の少ない $(J, D)=(64[\text{ms}], 3)$ の場合と $(J, D)=(128[\text{ms}], 3)$ の場合は $600[\text{ms}]$ 超程度まで許容される事が見て取れる。これはホップ数が送信待ち時間を決定する大きな要因だと捉えることが出来る。この場合、ホップ数に関連して中継するノード数が増え、結果としてノードでの処理時間が大きな要因を占めていると考えられる。特にセンサ・ネットワークのような処理能力が低いチップ等では、これらの問題はより顕著に現れると考えられる。

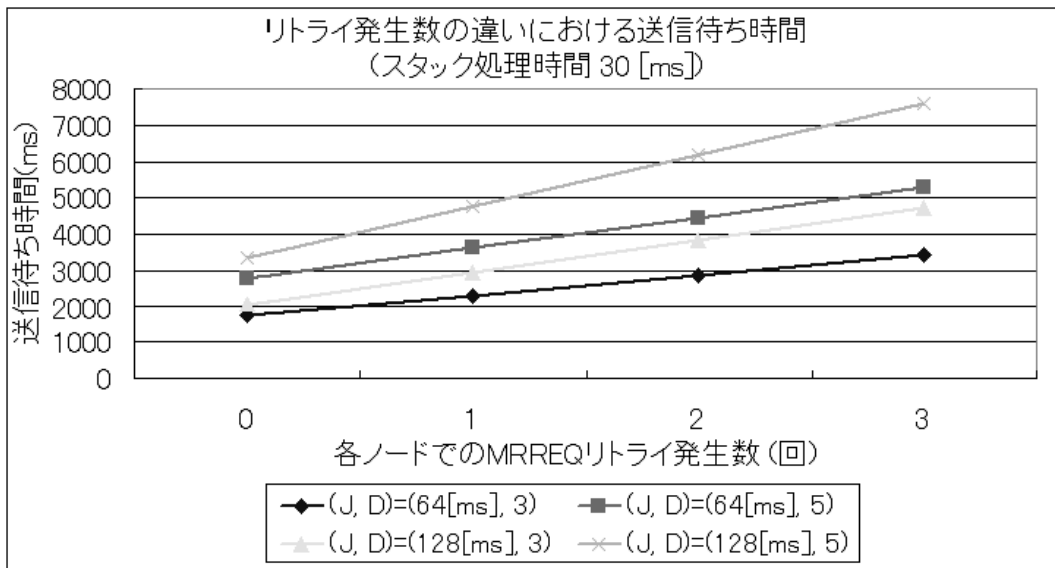


図 6.29: 拡張経路探索時のリトライと送信待ち時間の関係

リトライ回数と送信待ち時間からの考察

図 6.29 はリトライと送信待ち時間の関係を示したものである。各ノードにて発生したリトライ回数が横軸に、送信待ち時間について縦軸に示してある。ノードの割り込みやキューイング遅延などをまとめてプロトコル・スタック処理時間として定義しており、図 6.28 の結果を参考にしたうえで今回はプロトコル・スタック処理時間として 100[ms] という値を採用した。各ノードにおいて同じ数だけリトライが発生するという仮定であるため、多くのホップ数を經由するノードほど、Multi Route Request 送信リトライの際に故意に発生させる遅延が送信待ち時間と大きな関係を持っていると想像できる。そして、経路探索タイムアウトの 10 秒という制限から考えると、この 4 パターンにおいてはどれも許容範囲内に収まっていることが図から見て取れる。しかし、 $(J, D)=(64[ms], 3)$ や $(J, D)=(128[ms], 5)$ のような多くのホップを經由するネットワークを想定した場合、階層の深さ故 PAN 内にノードが非常に多く存在するため、CSMA/CA における衝突を避けるための待ち時間や、輻輳問題といった多くの点を考慮する必要があると考える。

6.4 評価のまとめ

本章では、トポロジ構成・アプリケーション構成の観点、経路探索の待ち時間という項目について評価を行った。トポロジ構成の観点について、スター型で構成する場合は純粹にどれ程のノードが同じグループに所属するかというパラメータが重要であり、アプリケーションの仕様や構成によって、ユニキャスト・ブロードキャスト・提案方式を使い分けると効率化が達成できると考えられる。

クラスタ・ツリーで構成する場合は、効率化という点に注目すれば、自分の子ルータ・ノードより下流に存在するグループ情報が上流から受信したデータの中継送信回数や送信量に大きく反映する事が判明した。従来であればツリーという構造上、上流に存在するノード程送信の際の負荷は増大したが、提案方式を採用することにより上流ノードへの負荷を低減させる事が出来た。

メッシュ型トポロジにおいては、評価項目の効率化を達成する事が出来た。特に経路状態や下流に存在するグループ情報次第で効率化は大きくとなるということが判明した。また経路探索時には非常に時間が掛かることも判明し、送信待ち時間について別評価として検討を行った。

アプリケーション構成の観点について、求められているアプリケーションを明確にし、提案方式を適用した各トポロジにおいてグルーピングの検討を行い、各トポロジで運用することの利点・欠点を明確にした。特にスター型においては、物理的な場所に依存せずシンプルな構成で送信回数や送信量等の点において効率化を図れることが判明した。

最後に経路探索時の待ち時間について評価と検討を行った。評価結果として各ノードの処理時間、リトライ回数、ネットワーク構成のパラメータを設定し、各パラメータの違いにおける送信待ち時間について検討し、ZigBeeにおいて経路探索後実際にどの程度待てば良いか値を算出した。

第7章 まとめと今後の課題

本論文では、近距離無線通信規格 ZigBee においてグルーピング方式を実現する手段について研究を行った。

グルーピング方式実現の際に、ZigBee におけるどのレイヤで実現すべき点を検討した。実現するレイヤの決定後、ZigBee ネットワークを構成する各トポロジについてグルーピング方式を実現する手段を提案した。グループ ID の定義では、各ノードがグループ ID を保持する方法について検討を行い、定義を行った。スター型トポロジにおいては、隣接テーブルに GroupID 情報を管理するフィールドを定義し、ブロードキャスト・フレームに定義したグループ ID 情報を付加することにより、グループに対し一斉にデータの送信を行う。クラスタ・ツリー型トポロジでは、グルーピング・テーブルと呼ばれるグループを管理するためのテーブルを定義しユニキャストを用いて適宜該当すべき方向へとルーティングを行う。またメッシュ型トポロジでは、グループへのルーティングを実現するために、従来のルーティング・テーブルを拡張し、グループ宛のデータを送信できるように変更を加えた。

実装について、PC において仮想評価ボードを利用し従来のネットワーク層の機能に本提案方式を追加を行った。

評価については、トポロジ構成の観点、アプリケーション構成の観点、経路探索時の待ち時間という3つの項目に対して評価とその考察を行った。トポロジ構成の観点では、各トポロジ毎に送信回数やデータ送信量、テーブル・サイズについて比較・検討し、アプリケーション構成の観点では、実際に求められているアプリケーションを説明した後、提案方式をどのように適用するのが最も効率が良いかについて各トポロジ毎に利点・欠点を明確にした。送信待ち時間では各ノードの処理時間や送信リトライ回数から送信待ち時間を算出・検討を行った。

本研究で提案する ZigBee におけるグルーピング機構を利用する事により、PAN を構成時にノードの送信回数の削減における送信電力・データ送信量の削減を実現することが可能となる。また家電やビルディング、工場といった場所での利用を想定した際に、各機器に対し従来より柔軟な操作性を提供することが可能となる。

今後の課題として、グループ ID 情報の動的割り当てやグループ宛に行った要求に対してデータを集約収集を行う方式が挙げられる。これらを達成することでさらにグルーピング機能が効率的に利用できると考えられる。

第8章 謝辞

本研究を進めるに当たって、研究の方向性について指針を与えて下さり、また熱心なご指導ご助言を頂きました丹康雄助教授に深く感謝致します。

また本研究に関して、非常に多くの助言・ご意見を頂きました博士課程の中田潤也氏、牧野義樹氏に心から感謝致します。

そして、励まし合いながら研究生生活・学生生活を過ごしてきた丹研究室の皆様に深く感謝致します。最後に私を支えてくれた多くの友人、そして学ぶ機会を与えてくれた家族に感謝し、ここに謝辞と致します。

参考文献

- [1] ZigBee Alliance ,
On-line available at <http://www.zigbee.org/>
- [2] IEEE802.15.4 WPAN-LR Task Group ,
On-line available at <http://www.ieee802.org/15/pub/TG4.html>
- [3] ZigBee Document 053474r06, Version 1.0 , Dec 2004
- [4] ZigBee Document 053474r13 , Dec 2006
- [5] IEEE Std 802.15.4-2003 , May 2003
- [6] Ad hoc On-Demand Distance Vector Routing ,
On-line available at <http://www.ietf.org/rfc/rfc3561.txt>
- [7] A ZigBee/IEEE 802.15.4 Multi-platform Protocol Stack ,
On-line available at <http://www.ece.msstate.edu/~reese/msstatePAN/>
- [8] 磯貝 彰則, 牧野 義樹, 中田 潤也, 丹 康雄, ”近距離無線通信規格 ZigBee における
ノードグルーピング方式”, 平成 18 年度電子情報通信学会ソサイエティ大会, 2006.9
- [9] 中井 彦一郎, 出内 将夫, 榊原 寛, 船木 康平, 徳田 英幸, ”サービス支援型センサ
ネットワークグルーピング機構”, 情報処理学会システムソフトウェアとオペレーティ
ングシステム研究会, 2005.5
- [10] 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”, 電
子情報通信学会, 2003.9
- [11] 三村 和, 飛岡 良明, 森川 博之, 青山 友紀, ”サービス指向グルーピング機構を用い
たユーザ主導ネットワークの構築”, 第 13 回マルチメディア通信と分散処理 (DPS)
ワークショップ, pp. 290-294, 2005.11
- [12] M. Karir , D. Dharmaraju , J.S.Baras , S. Das, ”An Implementation Study of Multi-
cast Extensions of AODV” Proceedings of SPECTS 2003 Conference,-CD, Montreal,
Canada, July 20-24, 2003.

- [13] Kenji Leibnitz , Naoki Wakamiya , Masayuki Murata , "Impact of Clock Skewness on Synchronized Sensor Clusters Operating with IEEE 802.15.4" , Ubiquitous Computing & Networking Systems , 2005
- [14] A. Koubaa , M. Alves , E. Tovar , "A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks" , WFCS 2006 , 2006
- [15] 鄭立 , "実践入門ネットワーク Zigbee 開発ハンドブック" , 株式会社リックテレコム , 2006
- [16] 阪田 史郎 , 田中 成興 , 西室 洋介 , 川崎 光博 , 福井 潔 , ユビキタスネットワークングフォーラム センサネットワーク部会 , "ZigBee センサーネットワーク 通信基盤とアプリケーション" , 株式会社秀和システム , 2005

付録A MSSTATE_LRWPAN

A.1 情報の定義

仮想センサ・ノードの持つ機能やセンサ・ボードとしての情報を定義するために、プリプロセッサの定義において各ノードのコンパイル時に値を定義する必要がある。特に仮想センサ・ボード機器自体が持つ固有のアドレス情報やどの論理デバイスの役割を果たすか等の情報について予め定めておく必要があるため、以下のような定義を行うことで実現する。

PAN 内における役割

LRWPAN_COORDINATOR (LRWPAN_FFD)

LRWPAN_ROUTER (LRWPAN_FFD)

LRWPAN_RFD

IEEE 拡張アドレス情報の例 (64bit)

aExtendedAddress_B7=0x00

aExtendedAddress_B6=0x12

aExtendedAddress_B5=0x4B

aExtendedAddress_B4=0x00

aExtendedAddress_B3=0x00

aExtendedAddress_B2=0x01

aExtendedAddress_B1=0x21

aExtendedAddress_B0=0x71

トポロジ情報

TOPOLOGY STAR

TOPOLOGY TREE

(TOPOLOGY MESH)

評価ボード ID (ボードの操作時に利用)

EVB_ID=1

A.2 GroupID 情報追加

仮想センサ・ノードにグループ情報を保持させる方法として、こちらもプリプロセッサの定義を利用して GroupID に関する宣言を行う。

GroupID 情報

```
Group_ID=0x3F
```

A.3 送受信時の GroupID 情報付与・解析

送信時の GroupID 情報の指定

送信時に宛先の Group 情報を指定する方法として、アプリケーションで送信宛先アドレス等を指定する際に同時に宛先グループを指定する。指定されたグループはネットワーク層においてヘッダに情報を付与され送信される。その際の送信関数と情報を付与する変数について以下に説明を行う。LADDR_UNION については、ZigBee で利用される LongAddr と ShortAddr の組み合わせた共用体であり BYTE は unsigned char 型で宣言してあるものとする。フレーム・コントロール・フィールドについては、図 4.2 を参照してほしい。

送信を行う関数

```
void aplSendMSG (LADDR_UNION dstMode,    /* 宛先アドレスモード */
                BYTE dstEP,             /* 宛先エンド・ポイント */
                BYTE cluster,           /* クラスタ */
                BYTE srcEP,             /* 送信元エンド・ポイント */
                BYTE pload,             /* ペイロード */
                BYTE plen,             /* ペイロード長 */
                BYTE tsn,               /* トランザクション番号 */
                BYTE reqack,            /* ACK 要求の可否 */
                BYTE GroupID)           /* グループ ID 情報を指定 */
```

送信時に指定したの GroupID 情報の追加

```
BYTE fcfmsb    /* フレーム・コントロール・フィールドの最上位バイト */
    fmfmsb 内に送信時に指定した GroupID 変数に含まれるグループ情報を付加
```

受信時の GroupID 情報の解析

パケットを受信した際にどのグループ宛のデータ化を判別する必要がある．そこで A.4 の NWK_RXSTATE_START 内においてヘッダ解析を行う際に，従来のアドレス情報に加え，グループ情報の解析も行うように変更を行った．グループ情報を解析する際に，ヘッダの位置をポインタとして渡し，解析を行う．

```
/* ヘッダのポインタからアドレス・グループ情報を解析 */  
void nwkParseHdr(BYTE *ptr)  
    引数 *ptr: ヘッダ情報の開始位置を指すポインタ
```

A.4 有限状態機械

MSSTATE_LWWPAN において、NWK 層での受信用の有限状態機械 `nwkRxFSM` は以下の状態から構成される。本研究における提案方式を実装するため、NWK_RXSTATE に `DO_MYGROUP_ROUTE` と `DO_OTHERGROUP_ROUTE` という新しい状態に追加した。`DO_MYGROUP_ROUTE` は自分が所属するグループ宛のデータを受信した際の受信・ルーティング処理を行い、`DO_OTHERGROUP_ROUTE` は自分が所属しないグループ宛のデータを受信した際のルーティング処理を行う。
構成するトポロジ情報や論理デバイスの種類により行う処理を変更する。

`nwkRxFSM` : 受信時の有限状態機械

```
typedef enum _NWK_RXSTATE_ENUM {  
    NWK_RXSTATE_IDLE,  
    NWK_RXSTATE_START,  
    NWK_RXSTATE_APS_HANDOFF,  
    NWK_RXSTATE_DOROUTE,  
    NWK_RXSTATE_DO_MYGROUP_ROUTE,  
    NWK_RXSTATE_DO_OTHERGROUP_ROUTE  
} NWK_RXSTATE_ENUM;
```


付録B ZigBee Document 053474r13

B.1 概要

2006年12月末, ZigBee Alliance より新たなる ZigBee Specification[4] が公開された。公開された Specification では, 以前公開されていた Specification への機能の追加や大幅な仕様の変更が行われている。付録Bでは, 追加された変更点のうち本研究と深く関係のある事柄を取り扱い, それぞれの変更点について簡潔に説明を行う。ネットワーク層が保有する機能のうち, 本研究に大きく関わる変更点として以下が挙げられる。

- ネットワーク層ヘッダ構成の変更
- ソース・ルーティング方式の実現
- マルチキャストの実現

B.2 ネットワーク層ヘッダ構成の変更

新しく公開された Specification において, 大きな変更点の一つとしてネットワーク層ヘッダ構成情報が変更されたという点が挙げられる。新しく公開されたバージョンにおいては, Reserved として予約済みフィールドであったフィールドが新規フィールドとして利用されただけでなく, 今まで存在しなかった新規フィールドの定義や可変長フィールドの利用におけるネットワーク・ヘッダ・サイズの増大を見て取れる。以下の図 B.1 が新しいネットワーク層フレーム・フォーマットであり, FrameControl の詳細を B.2 に示す。

Octets: 2	2	2	1	1	0/8	0/8	0/1	Variable	Variable
Frame control	Destination address	Source address	Radius	Sequence number	Destination IEEE Address	Source IEEE Address	Multicast control	Source route subframe	Frame payload
NWK Header									Payload

図 B.1: 新しいネットワーク層フレーム・フォーマット

Bits: 0-1	2-5	6-7	8	9	10	11	12	13-15
Frame type	Protocol version	Discover route	Multicast flag	Security	Source Route	Destination IEEE Address	Source IEEE Address	Reserved

図 B.2: 新しい Frame Control フィールドの詳細

図 B.1, B.2 を見て判断できることとして, ネットワーク層フレーム・フォーマットにおいて MultiCast control や Source route subframe といったフィールドが新規追加されていることである. また FrameControl フィールド内において, MultiCast flag, Source Route, IEEE 拡張アドレス情報などが新たに定義されている点が以前の Specification との違いと取ることができる. 従来仕様では, ネットワーク層ヘッダに含むルーティング情報はネットワーク・アドレスのみであった. そこで各センサ・ノード固有の IEEE 拡張アドレスを保持させることにより, 通信の際の宛先指定の幅を広げることができると考えられる. 本研究の提案方式では, 該当フィールドの一部をグループ ID 情報格納フィールドとして利用していたため, Group が存在することを示す flag とネットワークヘッダの可変長フィールドの後に専用のフィールドを追加するなどの工夫を行うことで解決できると考えられる. マルチキャストとソース・ルーティングに関連するフィールドの詳細については以下で説明を行う.

B.3 ソース・ルーティング方式の実現

ソース・ルーティングを利用することで特定のノードを経由させた通信を行うことが可能となる。新しく公開された仕様においては、図 B.1, B.2 の Source Route subframe, Source Route フィールドを利用することにより機能を実現している。FrameControl フィールド内の Source Route(1bit) を 1 に設定すると、Source Route subframe において可変長サイズのアドレス・リスト(図 B.3) を保持するようになる。各フィールドは Relay count から順に、指定したリレー総数、何番目の情報まで利用したか、各アドレス情報の順に格納されている。

Octets: 1	1	Variable
Relay count	Relay index	Relay list

図 B.3: ソース・ルーティング・リストの詳細

B.4 マルチキャストの実現

マルチキャスト機能を利用することにより、宛先グループに所属する複数のノード同時にデータを送信することが可能となる。マルチキャスト利用時には、宛先アドレスとしてブロードキャスト・アドレスが用いられ、送信元アドレス情報とシーケンスナンバーから既に該当マルチキャストフレームを受信したかどうかを判別する。具体的な利用方法として前述の FrameControl フィールドに存在する MultiCast flag を 1 とすることで、MultiCast Control フィールド内の値が有効となる。MultiCast Control フィールドの詳細は以下の図 B.4 のようになる。以下では、各フィールドの詳細について説明を行う。

Bits: 0-1	2-4	5-7
Multicast mode	NonmemberRadius	MaxNonmemberRadius

図 B.4: MultiCast Control フィールドの詳細

- Multicast Mode

2bit で構成される Multicast Mode フィールドは、名前の通りマルチキャストのモードを指定する。マルチキャストのモードとして、メンバーモード、非メンバーモードの2種類が存在する。メンバーモードは既にマルチキャストグループに所属しているメンバーからグループへのデータの送信を実現し、非メンバーモードはマルチキャストグループに所属していないメンバーからグループへのデータ送信を実現する。図B.5は、マルチキャストを送信する際のパラメータを規定している。

Multicast Mode Field Value: b ⁰ b ¹	Field Meaning
00	Non-member mode
01	Member mode
10	Reserved
11	Reserved

図 B.5: MultiCast mode フィールドの値

- NonmemberRadius

3bit で構成される NonmemberRadius は、メンバーモード時のマルチキャストにおいて、メンバーでないノードの中継回数を示している。具体的には、受信したノードがメンバーである場合は MaxNonmemberRadius の値で更新され、メンバーでないノードが中継を行う毎に値を1ずつ引き、0になった時点で破棄する。有効な値の範囲として 0x01-0x06 を取り、0x07 は到達範囲無限として中継されても減算を行われない。

- MaxNonmemberRadius

3bit で構成される値であり、メンバーでないノードにおいて最大中継回数を示す。利用の詳細については NonmemberRadius において説明を行った。

グループ・テーブル・エントリ

各ノードは自身がどのグループに所属するかを把握するためにグループ・テーブルを保持する。グループ・テーブル・エントリの構成は以下の図 B.6 のようになる。各ノードはグループ宛のデータを受信した際、テーブルと自身のグループと照らし合わせることで各処理を実現することが可能となる。

Field Name	Field Type	Valid Range
GroupID	Integer	0x0000 – 0xffff

図 B.6: グループ・テーブル・エントリの構成

マルチキャストの活用

新しく公開された Specification において、上記のマルチキャストを様々な方法に利用している。利用時にブロードキャスト・アドレスが用いられることから、例としてメッシュ型で主に利用される経路探索が挙げられる。経路探索時の Route Request コマンドはネットワーク層のデータ・フレームとして送信されるため、ネットワーク層ペイロードに含まれるオプション・フィールドにマルチキャストを指定することにより、宛先グループへの経路探索や、そのグループ宛の情報を確保するためのルーティング・テーブルを構築することが出来る。このルーティング・テーブルは、非メンバーモードの際に該当グループの経路情報を知ることができ、メンバーモードにおいては、前述した Multicast control フィールドが利用される。これらの処理の詳細については、参考文献 [4] を参考にしてほしい。

B.5 比較

本節において、提案方式と新しく公開された ZigBee Specification において、宛先となるグループへのデータ転送を実現する際、必要な情報の項目の比較を行う。

比較項目	提案方式	新しい ZigBee Specification
グループ数	64 個 (6bit)	65536 個 (16bit)
所属グループ	複数可	複数可
必須テーブル (ルーティング関連)	グループ拡張隣接テーブル グループピング・テーブル グループ拡張ルーティング・テーブル	隣接テーブル グループ・テーブル ルーティング・テーブル ブロードキャスト・トランザクション・テーブル
利用アドレス	ユニキャスト ブロードキャスト	ブロードキャスト (新定義のブロードキャスト)
利用方法	グループ ID フィールド指定	宛先+該当フィールド指定
グループ間の通信	可	可
到達範囲	該当グループ・ノード全て	該当グループ・ノード全て 限定ホップ内のグループノード

表 B.1: 提案方式と新規格の比較表

B.6 考察

本節では、本論文において提案した ZigBee におけるノード・グルーピング機構と新しく公開されたバージョンの Specification における方式の違いや類似に関する考察を行う。

まず、本提案方式においてネットワーク層でグルーピングを実現した一つの理由として、上位層のアプリケーション・サポート・サブ層においてプロファイルやエンドポイント等の機能と組み合わせることにより、ZigBee センサ・ノードに対してより柔軟な操作・制御手段を提供することが出来るという点があった。公開された Specification[4] では、この概念と同様に上位層であるアプリケーション・サポート・サブ層において GroupID を APS ヘッダに採用・活用することにより、データ配信を実現している。この点において、提案方式と Specification の目的は酷似していると考えられる。

また提案方式がユニキャスト、ブロードキャストにおいてグループ ID を指定して配信を実現するのに対し、Specification においてはブロードキャストを利用するため、頻繁なデータ通信を行う際にはネットワークの輻輳や各ノードが保持するブロードキャスト・トランザクション・テーブル容量を考慮しなければならない。特に多数のノードで構成されるメッシュ型トポロジにおいては、これは特に顕著な問題点と捉えることができ、提案方式のメッシュ型の様に適宜ユニキャストを利用すれば頻繁に各所からグループに対して通信が行われるネットワークにおいては特に有効であると考えられる。

本考察のまとめとして、以前の ZigBee Specification[3] において必要と考えられた結果、新しく追加された機能の一つがこのマルチキャストである。本提案方式で示した ZigBee ネットワークにおいて複数のグループを構成し、一括したデータ配信方法を実現、効率化を図るという事柄は、従来のこの規格において必要な機能であり、本研究の方向性や目的は正しいものであったと考えられる。