

平成18年度
研究開発成果報告書

ZigBee を利用したユビキタスネットワーク
技術の研究開発

沖電気工業(株)

平成19年4月

情報通信研究機構

平成18年度 研究開発成果報告書 (一般型)

「ZigBee を利用したユビキタスネットワーク技術の研究開発」

目 次

1	研究開発課題の背景	2
2	研究開発の全体計画	
2-1	研究開発課題の概要	3
2-2	研究開発目標	7
2-2-1	最終目標	7
2-2-2	中間目標	8
2-3	研究開発の年度別計画	10
3	研究開発体制	11
3-1	研究開発実施体制	11
4	研究開発実施状況	
4-1	無線ノードの開発	12
4-1-1	18年度の開発目標	12
4-1-2	実施状況	12
4-1-3	達成状況	16
4-2	技術方式の研究開発	18
4-2-1	18年度の開発目標	18
4-2-2	実施状況	19
4-2-3	達成状況	27
4-3	実証実験	28
4-3-1	18年度の開発目標	28
4-3-2	実施状況	28
4-3-3	達成状況	44
4-4	総括	44
5	参考資料・参考文献	
5-1	研究発表・講演等一覧	

1 研究開発課題の背景

人やモノを始めとして、様々な対象・場所に小型の無線ノードを遍在させ、それらに搭載したセンサが集める情報を交換し、無線ノードを通じてあらゆる機器を制御することで、新しい機能・サービスを提供するという、ユビキタスセンサネットワークの実現が望まれている。総務省「ユビキタスセンサネットワーク技術に関する技術調査会」の最終報告でまとめられているように、ユビキタスセンサネットワークは、近距離低レート低消費電力低コストの無線ノードが無数に設置され、自律的にアドホックネットワークを形成するものである。「安全・安心」「快適・ゆとり・娯楽」「最適・効率」をキーワードとして、多くの分野での応用アプリケーションが期待されている。

ユビキタスセンサネットワークの研究は、欧米で軍事向けとして始まったものが多いが、日本でも民生向けの研究開発が取り組まれている。大学を中心にさまざまな無線ノードが開発され、それを使用したプロトタイプの実験が行われている。無線の方式やネットワークキングの方式も千差万別であり、その上に実装されるセキュリティ機能やアプリケーションなども多岐にわたっている。

これらの無線ノードを各企業が独自に開発しては、実用化に向けた課題を解決するために多くの時間と工数を要してしまう。また、将来インフラとして普及させていくためには、相互接続性の確保も必要となる。そこで、当然ながら標準化という方向性が重要となる。

ユビキタスセンサネットワークに対する技術的要求はさまざまであるが、多数のノードをいろいろな場所にばら撒いて設置するということを実現するためには、低コスト、低消費電力という課題は必須となる。この要求項目を重視して取り組まれた標準のひとつが ZigBee である。ZigBee はユビキタスセンサネットワークを実現する無線ネットワーク方式として、有力な方式といえる。

ZigBee は、ZigBee Alliance という世界各国の企業の連合からなる団体で規定された方式であり、デファクトスタンダードを目指している。無線の物理(PHY)レイヤと MAC レイヤは国際標準である IEEE802.15.4 を使用しており、ネットワークレイヤでのルーティング方式やセキュリティ方式などを規定している。また、相互接続性を高く保証するために、アプリケーションごとにプロファイルを定めて、そのインタフェースや利用する技術を規定している。

ZigBee は、2004 年 12 月に初版(ZigBee 1.0)が規定され、無線部の LSI やモジュールがメンバ各社で開発されている。沖電気もそれらの開発に取り組んでおり、それらを組み合わせたアプリケーションシステムの試作にも取り組んでいる。

ただし ZigBee 1.0 は、基本セットを優先して規格化を進めてきたため、実用化に向けて必要な技術方式のいくつかが不足している。そこで、不足した技術を標準外の周辺技術として補いながら、コア部分への修正は、次世代の ZigBee へ提案して、標準化を推進していくことが重要である。

また、ZigBee は無線ノードの技術は規定されているが、それらを利用したバックボーン側のシステムの規定も十分ではない。ユビキタスセンサネットワークを利用してシステムを構築しようとする企業は、必ずしも無線の技術を有しているわけではなく、既存のシステムとセンサネットワークをどのように接続してよいかわからない。そこで、普及のためには、共通プラットフォームとなるシステムの開発も必要であり、標準化を進めていくことが重要となる。

2 研究開発の全体計画

2-1 研究開発課題の概要

ZigBee の普及を促進するために、以下の開発を行う。

- ① ZigBee 規定に準拠した無線ノードの開発(ハードウェア、ファームウェア)
- ② ZigBee 規定に不足する技術を補うための技術方式開発
- ③ 開発成果を評価するための実証実験

以下にそれぞれ詳細な開発内容を記載する。

ア 無線ノード開発

ZigBee 1.0 やそれ以降のバージョンに準拠した無線ノードを開発する。無線ノードは建物などに設置する固定ノードと、人やモノにつける移動ノードでは、要求項目が異なるため、それぞれ適切なノードを数種類開発することになる。また、その中でも、ルーティング機能やネットワーク管理機能を持つノード(ZigBee ルーターや ZigBee コーディネータと呼ばれる)など、持つべきネットワーク機能も多様である。

また、アプリケーションごとに要求機能やコスト(価格)も異なるため、本来は、アプリケーションごとの最適設計を行うのが理想であるが、開発効率やコストパフォーマンスも考慮し、共通化を行う。さまざまな応用分野に適応可能とするために、共通項目を抽出し、汎用的なモジュールで構成した部分と、アプリケーションごとの要求に依存する個別モジュールで構成する部分を、整理して設計する必要がある。ただし、汎用性を持たせるために、搭載機能が冗長になり、高コスト、高消費電力になってしまうと、本来の目的から外れる。実用面と開発効率面を考慮したノードの開発を行う。

無線ノードのラインナップがある程度揃ってくれば、アプリごとに必要な機能を持つ無線ノードを組み合わせることでネットワークを構築することができ、システム設計が容易になる。本提案では、そのような汎用な基本無線ノードを数タイプ開発することを目標とする。

開発する代表的なタイプは、大きく分類して以下の 3 とおり。

①他の機器に挿入する携帯ノード(タイプ A)

PDA や携帯電話、小型のノート PC など、既存の携帯機器と一緒に動作する無線ノードを開発する。具体的には、USB や SD カード、携帯電話インタフェースなど、代表的なインタフェースを持つ無線ノードをそれぞれ開発する。電源は、携帯機器から供給を受けることを基本とし、アプリケーションソフトウェアも携帯機器側の CPU で動作させることを考える。そのため、無線ノードに搭載するマイコンは、ネットワーク機能までを動作させればよいため、低コスト化が実現できる。

携帯ノードであるため、基本的にはルーティング機能を持たない ZigBee エンドデバイスタイプで構成する予定であるが、必要に応じて、ルーティング機能を持つ ZigBee ルータータイプで構成することもある。例えば、携帯電話を中心に、音楽再生機器やゲーム機、各種生体センサ搭載ノードなどが、ツリー構造でぶら下がることも考えられる。

最終的には、携帯機器内に組み込むモジュールにすることが目標であり、ZigBee 一体型の携帯機器を開発・事業化することを目指す(これは、本提案の目標外とし、終了後に取り組む)。

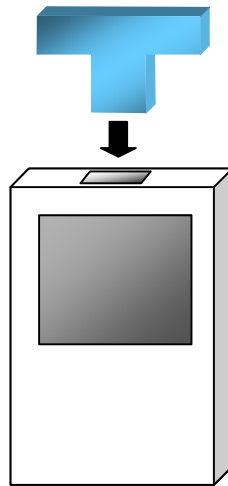


図1 タイプ A の概念図の例

②ルーティング機能を持つ固定ノード(タイプ C)

建物などに設置するタイプであり、電源が取れていることを基本とする。ただし、街中の街灯など、AC 電源を確保するのが困難な場所への設置も考えられるため、電源は太陽電池等の電源を利用する構成も検討する。ルーティング機能を持ち、設置した複数のノード間でマルチホップの通信を行う。マイコンは高機能なものを搭載し、メモリ容量も大きくする。設置場所に制限を与えないためには、できるだけ小型であることも重要となる。

③ゲートウェイ機能を持つ固定ノード(タイプ D)

タイプ C に、IP や他の専用プロトコル間を変換する GW 機能を追加して持つ。その他の機能はタイプ C と同等と考える。他のプロトコルとして、IP に変換するゲートウェイの試作は必須目標と考える。

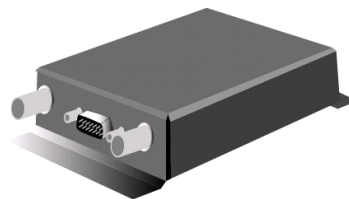


図2 タイプ C,D の概念図の例

イ 技術方式開発

ZigBee 1.0 では、まず固定的に設置した無線ノードを対象として規定されているなど、実用に向けてはいくつかの技術的な不足がある。そこで、以下のような技術を開発する。

- ① 移動ノードに対応する機能
- ② 無線ノードの故障時に、ルートを再設定する技術
- ③ スリープ制御と省電力化の向上
- ④ 無線 LAN との連携

⑤ 実用化に向けた、新しいプロファイルの規定

これらの技術は、従来でも研究されてきている方式でもあるが、ZigBee 1.0 を準拠させながら動作することが必須となるため、従来手法をそのまま利用することができず、新規の開発となる。

以下に、それぞれの課題の概要を説明する。

① 移動ノード対応機能

ZigBee 1.0 では、クラスタツリールーティング方式を規定し、マルチホップルーティングでの省電力と省メモリを実現している。

IETF などでは標準化が進められている従来のマルチホップルーティング方式では、大きく 2 種類の手法に分けることができる。1 つは、通信する際に受信ノードがどこにいるかを探索して、マルチホップルートを設定する「オンデマンド型ルーティング」であり、もう 1 つは、予め周囲のノードの場所を記憶したリストを用意しておき、それにしたがってルートを判断する「プロアクティブ型ルーティング」がある。オンデマンド型はリストを記憶しておく必要がないため、メモリ容量は少なくすることができるが、通信のたびにルート探索を行うため、通信のオーバーヘッドが大きい。一方、プロアクティブ型は、ルートは既にわかっているためオーバーヘッドは少ないが、周辺のノード数が多くなるにつれ、リストのメモリ容量が大きくなる。

これらの従来手法に対して、ZigBee 1.0 では、通信オーバーヘッドが少なく、メモリ容量もほとんど必要としない「クラスタツリールーティング方式」を規定している。各ノードはツリー状のトポロジーで接続され、どのノードの下にぶら下がっているかによって、各ノードが一意に決まるルールで ID(ネットワークアドレス)を付与される。また、他のノードの ID と自分の ID をある式に入力して比較することで、それが自分より上にいるのか下にいるのか、判断できるようになっている。

ただし、ツリー状に接続されて、ID を割り当てておくことが必要であるため、移動ノードへの対応が柔軟にできないという欠点がある。ノードの追加・削除の仕組みは規定されているため、ツリーの枝を渡り歩くような移動ノードに対して、ノードの追加・削除を繰り返すことで、接続を確保することは可能であるが、枝に接続するたびに異なる ID が割り当てられるため、同じノードであると認識することができない。

そこで、これらの課題を解決するような以下のような機能を、本提案で開発する。

- ・ 移動ノードに動的に同じ ID を付与する機能、または異なる ID を付与する場合にそれを同定する機能
- ・ ルーター間を移動(ハンドオーバー)できる機能

② 故障対応機能

上述のように、ZigBee 1.0 では、ツリー形状でトポロジーが固定されて各ノードが結びついているため、どこかの幹のノードが故障した場合、そこから下のノードへの通信ができなくなる。これも、ノードの追加・削除機能を繰り返して、接続することは可能であるが、元の ID が維持できないため、ネットワーク管理側で混乱が生じる。また、どの枝につながるおせばよいかを判断する仕組みは、ZigBee 1.0 では規定していない。

そこで、本提案では、故障時の以下のような機能を開発する。

- ・ 故障したノード以下のツリーを、最適な場所に再接続する機能
- ・ 元の ID と新しい ID を同定する機能

③ スリープ制御

ZigBee では、省電力を考慮した工夫が多く取り入れられているが、その多くは、ツリーの葉となる「エンドノード」に対して規定されている。例えば、通信制御の仕組みとして、帯域保証を行う GTS (Guaranteed Time Slot) 方式が規定されており、エンドノードは割り当てられたスロットで通信を行い、それ以外のスロットでは、機能を停止した「スリープ状態」になることで省電力を向上させる。ただし、ツリーの幹に当たる「ルーターノード」はエンドノードとの通信の間はずっと起動しておく必要があり、省電力効果は少ない。ただし、GTS で割り当てるタイムスロットを隣のルーターノードとぶつからないようにずらす必要があり、隣のルーターノードのスロットの間にスリープすることは可能である。これを実現するためには、隣のルーターとスロット割り当てのネゴシエーションを行う必要があるが、ZigBee 1.0 ではその仕組みは規定範囲外となっている。

そこで、本提案でスロット割り当ての仕組みとルーターのスリープ制御の機能を実現する。

④ 無線 LAN との連携

ZigBee は、センシングした短いデータを集めたり、制御ノードへ簡単なコマンドを送ったりすることを目的とした低レベルの無線ネットワーク方式であり、通信レートも 250kbps と非常に低い。このようなネットワークを例えばビル全体に敷設する場合、数千から数万のノードで構成する必要があるが、マルチホップ数が増えるにつれ、中継すべきデータが増え、上位ノードの負荷は重くなる。したがって、あまり多くのノードからなるネットワークをすべて ZigBee で実現することは困難である。そこで、少数のノードからなる PAN (Personal Area Network) を、GW (Gateway) を介してバックボーンネットワークにぶら下がる構成をとるのが普通である。

ただし、このような構成にした場合、隣の PAN との連携をバックボーンネットワーク経由で行う必要があり、ノード間の連携を柔軟に実現することが難しくなる。

また、バックボーンネットワークとしては、普及が加速的に向上している無線 LAN が候補となるが、無線 LAN のメッシュ網化(アドホックネットワーク化)も開発が進められているため、その網との連携も必要になる。

さらには、無線 LAN と ZigBee は、日本では同じ 2.4GHz 帯を使用するため、効率的な通信制御も必要となる。

そこで、本提案では、無線 LAN と ZigBee を融合させ、最適なネットワークを開発する。

⑤ 新しいプロファイル

ZigBee では、相互接続性を高めるため、応用用途ごとに利用する機能やパラメータの範囲を詳細に定めるプロファイルを規定している。例えば、家庭内の照明用のプロファイル(Home Control Lighting Profile)などが規定されている。今後、多くの応用分野でプロファイルの検討を行うことになっているが、すべての分野で規定することは困難であり、また、企業ごと、国ごとの条件や規制に対応したものにすることは難しい。そこで、ZigBee では、「プライベートプロファイル」として、企業ごとに独自のプロファイルを規定しても良いルールを設けている。

ZigBee では、日本企業の活動がそれ程アクティブではないため、日本の規制や条件を考慮したプロファイル検討が遅れている。本提案では、日本で利用する際に必要な条件を考慮した日本向けプロファイルを検討し、普及するように、他の企業も巻き込んだ

実証実験を行う計画である。必要に応じて、検討したプロファイルは日本提案として ZigBee への提案活動も実施していきたい。

ウ 実証実験

上述の課題ア、イで開発した無線ノードと技術方式の有効性を確認するために、京都市において実フィールドで実証実験を行う。

2-2 研究開発目標

2-2-1 最終目標（平成20年3月末）

ア 無線ノード開発

① タイプ A

PDA に挿入できる SD カードインタフェース、携帯電話に挿入できるインタフェース、ノート PC などに挿入できる USB インタフェースを持つノードをそれぞれ開発する。特に、小型化や低消費電力化、品質/剛性向上を目指すため、それぞれのタイプを数回、設計/製造を繰り返し、成熟度を高める。

アンテナ部を装置の外側に出す必要もあり、本体装置からはみ出る部分が存在するが、そのサイズは 5cc 以下となることを目指す。

電池寿命は、本体装置の消費電力に大きく依存するが、ZigBee を動作させることで、本体の動作寿命が 80%以下にならないことを目標とする。

② タイプ C

AC 電源で動作することを基本とするが、小型の太陽電池で動作するタイプも開発する。

サイズは、50cc 以下を目指す。

③ タイプ D

IP プロトコルとの変換 GW を持つタイプを開発する。IP 以外のプロトコルへの変換も必要により開発する。

イ 技術方式開発

① 移動ノード対応

移動ノードを持って、人が歩くスピードで固定ノード間(10m 間隔)を通り過ぎる場合、遅延なく ID が切り替わり、データ通信が途切れないことも目標とする。

② 故障対応

ツリーの幹のノードが故障した場合、1 分以内にトポロジーを再構築し、ネットワークが安定動作することを目標とする。

③ スリープ制御

ルーターノードの消費電力が、方式を適用しない場合と比較して 10%以下になることを目標とする。

④ 無線 LAN との連携

メッシュルーティングが動作する無線 LAN 網と、GW やブリッジで ZigBee 網が接続し、1000 ノードで安定したネットワークを開発する。

⑤ プロファイル

3 つのアプリケーションに適したプロファイルをそれぞれ検討し、ZigBee Alliance に対して、プライベートプロファイルとして認証を得る。

ウ 実証実験

京都市において屋外無線 LAN と ZigBee を連携させた実証実験を実施する。実証実験は「まちの防犯」アプリケーションへの適用を想定し、携帯ノード保持者があらかじめ設定された危険区域に近づいたり、特定区域から出入りした際に、携帯ノード保持者に警告メッセージを送出するとともに、監視者に通知メッセージを送出する機能を実現する。

2-2-2 中間目標（平成19年1月末）

ア 無線ノード開発

① タイプ A

SD カードタイプ、携帯電話タイプ、USB タイプを1回ずつ開発し、品質向上を目指すための課題を整理する。

サイズは 20cc 以下を目指す。

電池寿命は、ZigBee を動作させることで、本体の動作寿命が 80% 以下にならないことを目標とする。

② タイプ C

AC 電源で動作するタイプを開発する。

サイズは、100cc 以下を目指す。

③ タイプ D

IP プロトコルとの変換 GW を持つタイプを開発する。

イ 技術方式開発

① 移動ノード対応

移動ノードを持って、人が歩くスピードで固定ノード間(30m 間隔)を通り過ぎる場合、遅延なく ID が切り替わり、データ通信が途切れないことも目標とする。

② 故障対応

ツリーの幹のノードが故障した場合、5 分以内にトポロジーを再構築し、ネットワークが安定動作することを目標とする。

③ スリープ制御

ルーターノードの消費電力が、方式を適用しない場合と比較して 50% 以下にな

ることを目標とする。

④ 無線 LAN との連携

メッシュルーティングが動作する無線 LAN 網と、GW やブリッジで ZigBee 網が接続し、100 ノードで安定したネットワークを開発する。

⑤ プロファイル

1つのアプリケーションに適したプロファイルの検討に着手する。

ウ 実証実験

実証実験場所を特定するとともに、実証実験システムの設計を完了し、無線ノードへのアプリケーションを実装する具体的作業が開始可能な状態にする。

2-3 研究開発の年度別計画

金額は非公表

研究開発項目	17年度	18年度	19年度	年度	年度	計	備考
ア 無線ノード開発							
① タイプA	SD版開発	USB版、携帯版開発	システム連携・改良				
② タイプC		開発・評価					
③ タイプD	Ether版、シリアル版開発						
イ 方式開発							
① 移動ノード対応	基本アイデア検討	方式検討・評価	方式改良・システム上での評価				
② 故障対応							
③ スリープ制御							
④ 無線LAN連携							
⑤ プロファイル				プロファイル基本要件整理	提案活動	新プロファイル標準化	
ウ 実証実験	実験内容基本検討	内容精緻化、予備実験システム開発	予備実験実施、本格実験システム開発・実施				
間接経費							
合計							

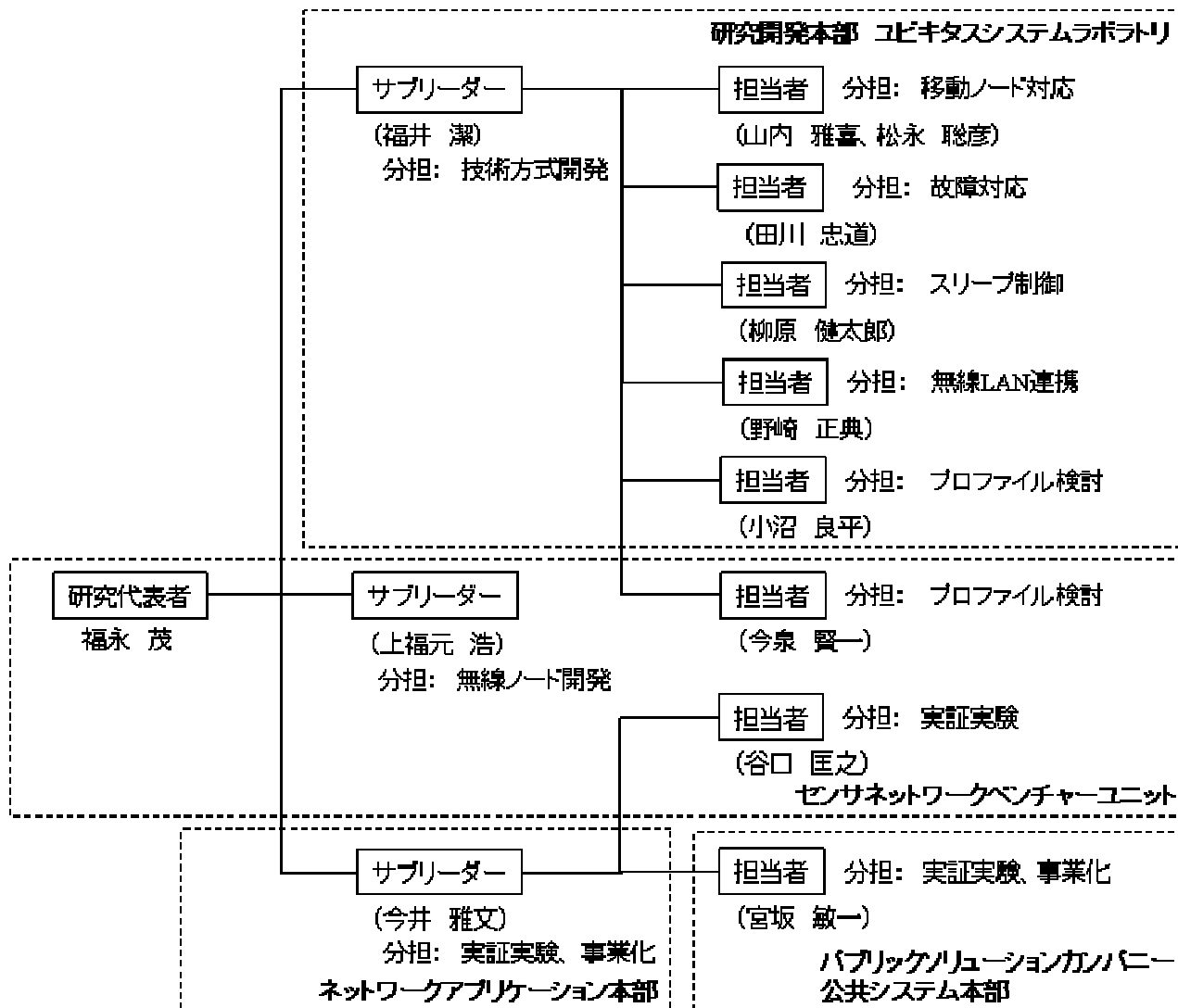
注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上(消費税を含む)。

2 備考欄に再委託先機関名を記載

3 年度の欄は研究開発期間の当初年度から記載。

3 研究開発体制

3-1 研究開発実施体制



4 研究開発実施状況

4-1 無線ノードの開発

4-1-1 18年度の開発目標

ZigBee 1.0 のバージョンに準拠した無線ノードを開発する。18年度では、提案した 4 タイプのうち、タイプ A と C の 2 タイプを中心に開発する。

① タイプ A

SD タイプのユーティリティ AP の開発および USB 接続タイプの試作開発および携帯接続タイプの基本設計を行う。

(目標値)

- タイプ D に接続するサーバにアクセスして情報を取得する、および通知する
- ZigBee ネットワークに参加・離脱しながら、移動する端末の情報を取得する

② タイプ C

ZigBee ネットワークでルーティング機能を有するタイプを開発する。

当該タイプは上位にタイプ C または D、下位にタイプ A、C または D が接続され、ルーティング・プロトコルにしたがって、ZigBee パケットを転送する機能をもつ。ネットワークに収容するノード数を増やすために、大容量のメモリを実装することでルーティングテーブルを大容量化する。

受信するパケット毎の RSSI (電波受信強度) を検出する機能をもつ。当該機能は受信強度による測位などを実現するのに必要な機能である。

省電力化するための、電力制御とルーティング・プロトコルを連携させる機能を実装可能とする。当該機能は、太陽電池等の外部給電機構を小型化し、設置コスト・維持コストを低減化するのに必要な機能である。

開発のパフォーマンスは、ルーティング時間、省電力化とする。

なお、屋外に設置することを想定した収容ボックスも併せて開発する。

(目標値)

- 電池駆動と AC 電源を選択して動作するタイプを開発する。
- サイズは 100cc 以下とする。
- エンドデバイス (タイプ B) の機能として電池を内蔵して省電力回路設計と検証、ルータ (タイプ C) の機能として、技術方式研究の成果を実装するに十分な性能をもたせる。内蔵センサと外部センサの切り替え機能をもたせる。
- サンプリング機能をもち、センシングデータの一時的な蓄積、加工ができる

4-1-2 実施状況

① タイプ A

①-A タイプ A (USB 接続タイプ)

以下の仕様の USB 接続タイプの無線ノードを開発した。

本ノードは PC 等と USB インタフェースを介して、PC 上のアプリケーションから ZigBee 通信を行うものである。USB カードの構成図を示す：

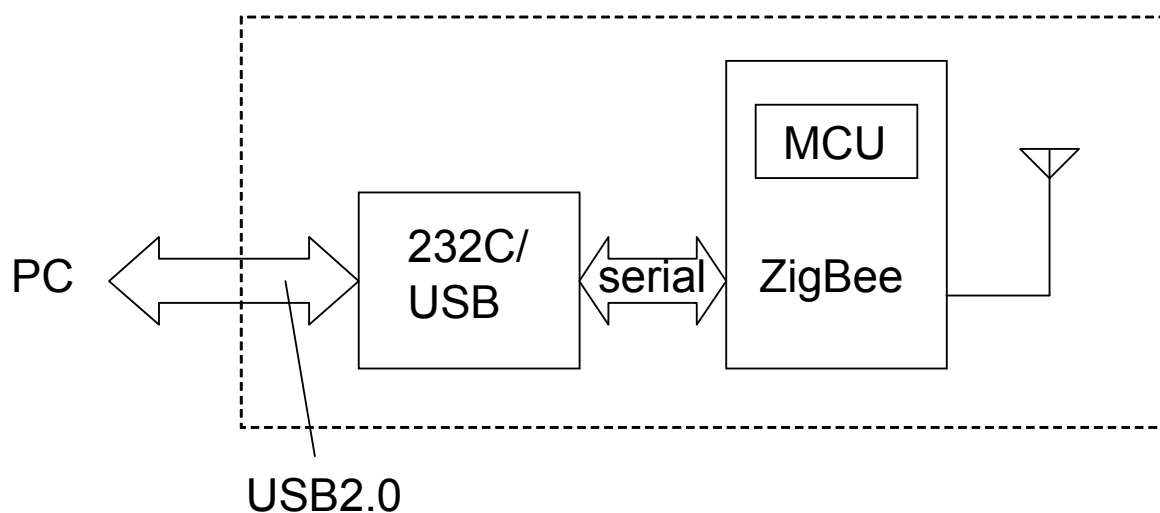


図 3 USB カードの構成図

(実装する機能)

- ① 給電動作
PC より USB インタフェースを介して、電力をとりだし、カード内の LSI 等に適した電圧で安定した電力を供給できる。
- ② 232C/USB 変換
USB カードは USB インタフェースのスレーブ機能を有する。PC が HOST となり、PC との間で送受したデータを一旦バッファし、232C インタフェースに変換して、ZigBee LSI との間で転送する機能を有する。
通信速度については、設定により可変とする。
- ③ ZigBee
ZigBee LSI は IEEE802.15.4 を実装し、上位プロトコル (ZigBee) スタックについても不揮発性メモリ等よりアップロード可能であり、一体化した ZigBee の機能を実装することができる。通信アプリケーションは ZigBee スタックの上位に実装されるが、AP 機能は PC 上で実現するものである。

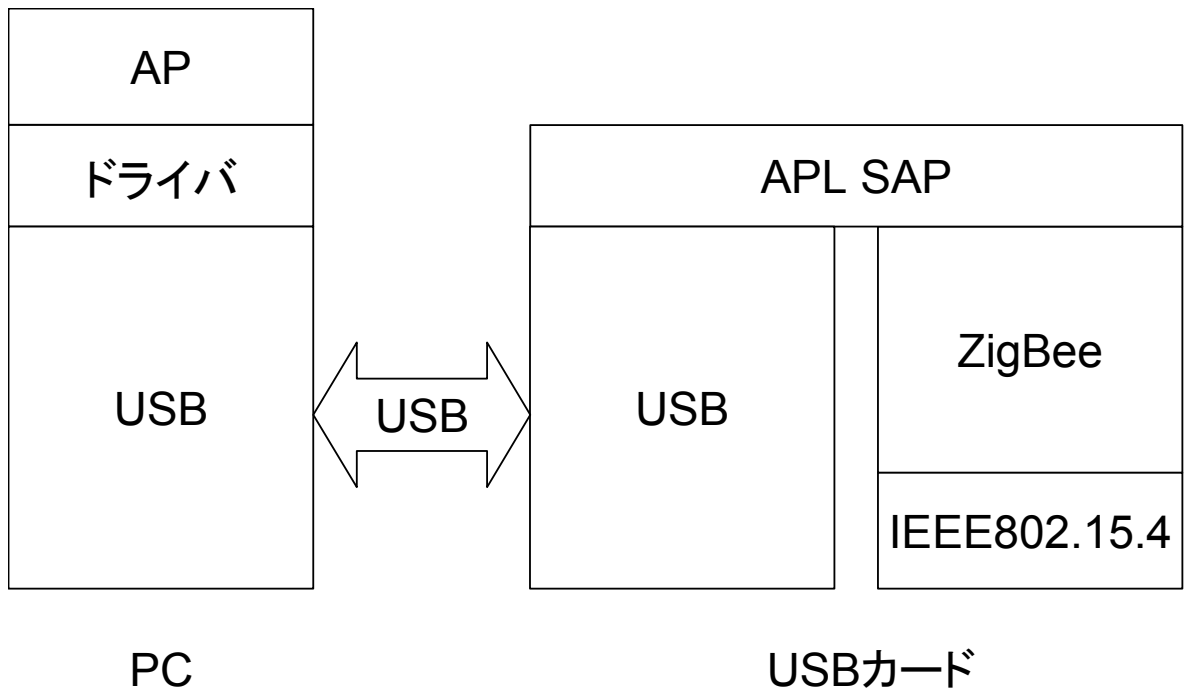


図4 USBカードのプロトコルスタック

APからの送信データ、送信コマンドをドライバに渡すと、PC上のドライバでUSBインタフェースを介した転送を行い APL SAP にデータが渡される。ZigBee スタックで、コマンドを解釈し、通信プロトコルに従って送信が行われる。受信の場合も同様のルートを経由する。

①-B タイプA (携帯電話機接続タイプ)

以下の仕様の、携帯電話接続タイプの無線ノードの開発に向けて、基本設計を行った。

本ノードは携帯電話とSDインタフェースを介して、携帯電話機上のアプリケーションから ZigBee 通信を行うものである。

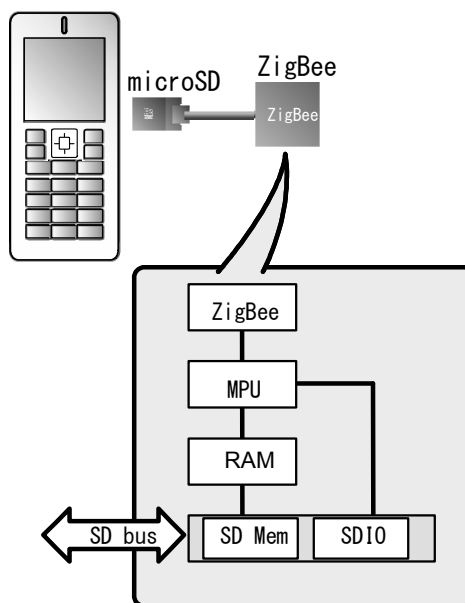


図5 携帯電話タイプのイメージ

携帯電話インタフェースの変更時期にあり、動向をみながら、基本設計までを行った。

②タイプ C

以下の仕様の無線ノードを開発した。

(基本機能)

本品は IEEE802.15.4 5 章の FFD として動作可能とする。

スター、またはピア・ツー・ピアのいずれのトポロジでも動作する。ピア・ツー・ピアで動作する場合、テンポラリに付加されるアドレスによるクラスタツリー、および、ユニークなアドレスによってメッシュが構成でき、PAN の参加順序によらずに、ad hoc に PAN に参加可能とする。構成された PAN において、ZR として動作する場合、隣接する ZC、ZR、または ZED との間でフレームの転送をするホッピング動作をする。隣接する ZR、ZED の個数およびホッピングの段数は、PAN を構築するにあたって、事前に決定される。

データ転送は、コーディネータとの通信（スター形）、ピア・デバイスとの通信（ピア・ツー・ピア）を可能とし、ビーコンモード／非ビーコンモードのいずれでも動作する。ビーコンモードで動作する場合、コーディネータからの最短経路の段数 N をもつものとする、段数 N+1 のピア・デバイスに対して、ビーコンを発出することを可能とする。

本品はルータとして動作するが、あて先アドレスが当該品を指定する場合、および特定アドレス（例：オール F）の場合は終端し、上位層との SAP を介して上位層に転送する。上位層から発出する場合も SAP を介する。AP 層では、下位層からの情報、または他の物理インタフェースからの情報をもとに、所定の処理を行う。

物理インタフェースの情報として、デジタルの入力 3、アナログ入力 1（10 ビット以上の A/D 変換を内蔵）、デジタル出力 4 をもつ。

(拡張機能)

省電力ルーティング機能を実現する。内蔵タイマをもち、トポロジ上位のピア・デバイスからのビーコンに同期して起動する。内蔵タイマの精度は 100ppm 内とする。自動修復のプロトコルを有する。自動ネットワーク構成機能を有する。

4-1-3 達成状況

① タイプA

USBタイプは、図6のような試作品を開発した。



図6 試作したUSBアダプタ外観

実施内容に記載した項目①～③に関する達成状況は

	項目	達成状況	
①	給電動作	問題なし	○
②	232C/USB 変換	問題なし	○
③	ZigBee 動作	19年度評価予定	×

目標値として掲げた容積、消費電力については

	目標値	達成値	
①	20cc 以下	30cc	△
②	本体への影響20%以下	常時動作で90mW 30WHのPCで1%以下	○

容積については、回路技術および実装技術ではクリアしているが、試作経費節約目的で市販ケースを使っているために、オーバーしている。ZigBee動作確認に関しては、19年度に実施する予定である。

携帯電話タイプについては、携帯電話のインタフェースからminiSDが無くなり、microSDへ全面的に変更になるという情報により、インタフェースを当初の予定から変更することにした。基本設計は終わっているが、microSDの物理的条件が厳しいことから、開発期間が延び、詳細設計以降は19年度に実施予定。

② タイプC

タイプBの一部検証も行う新タイプCとして見直した。また、想定顧客等のヒアリングにより、ルーティングとセンシングの両方をもとめたいという意見が多かったため、センシングデータのサンプリングと一時的蓄積を行う機能を加えることにした。

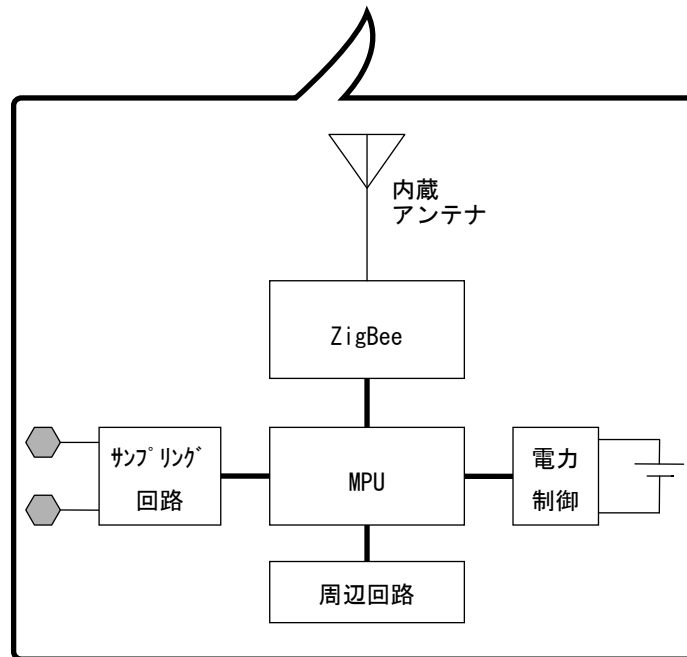


図7 新タイプCの構成

寸法の超過については、機能追加によるものであるが、軽微なものであるために特に対応は行わず、今後の改良(本委託業務外)にあたって小型化を図る予定である。

4-2 技術方式の研究開発

4-2-1 18年度の開発目標

ZigBee 1.0 に対して、以下のような実用化技術を開発する。

- ① 移動ノードに対応する機能
- ② 無線ノードの故障時に、ルートを再設定する技術
- ③ スリープ制御と省電力化の向上
- ④ 無線 LAN との連携
- ⑤ 実用化に向けた、新しいプロファイルの規定

① 移動ノード対応

移動先で接続するルーターノード選択精度、ルーターノードの接続切替遅延を改善する方式を開発する。

(目標値)

移動ノードを持って、人が歩くスピードで固定ノード間(30m 間隔)を通り過ぎる場合、遅延なく ID が切り替わり、データ通信が途切れないことも目標とする。

② 故障対応

17 年度に検討した方式を定量評価し、ルーターノードが故障してもネットワークを安定動作させることができることを確認する。さらに、ルーターノード故障時のネットワーク回復遅延時間を短縮するための方式改良を検討する。

(目標値)

ツリーの幹のノードが故障した場合、5 分以内にトポロジーを再構築し、ネットワークが安定動作することを目標とする。

③ スリープ制御

17 年度に検討した方式を評価し、スリープ制御が安定して動作することを確認する。17 年度に検討した方式は、安定動作を優先して検討したため、スリープしないルーターの配置を最適化することはあまり考えられていない。本年度は、スリープしないルーターの配置の冗長度が出来るだけ小さくなるように効率的なルーターの選択が出来るように方式改良を検討する。

(目標値)

ルーターノードの消費電力が、方式を適用しない場合と比較して 50%以下になることを目標とする。

④ 無線 LAN との連携

無線 LAN を介して構築した ZigBee ネットワークのトポロジー構成を最適化する方法を開発する。ブリッジ間の効率の良いルーティング方式を開発する。

(目標値)

メッシュルーティングが動作する無線 LAN 網と、GW やブリッジで ZigBee 網が接続し、100 ノードで安定したネットワークを開発する。

⑤ プロファイル

アプリケーションに必要な機能を標準に追加するために、プロファイル提案活動をすすめる。

(目標値)

ZigBee Alliance において、プロファイル提案し、ネットワークやセキュリティなどの技術要求が整理されている。

4-2-2 実施状況

① 移動ノード対応

昨年度開発した方式には、移動ノード数に比例してルーターノード切り替えトラフィックが増加するため、収容可能な移動ノード数に対するスケーラビリティが低いという課題があることが分かった。切り替えトラフィックが多くなる原因としては、以下が考えられる。

(ア) 各移動ノードが定期的に行う近隣ルーターノードの探索トラフィックが非常に多い

(イ) ノードの移動距離にかかわらず、定期的にルーターノード切り替えを実施しているため、不必要なルーターノード切り替えが行われている。

この課題を解決するために、昨年度開発方式のシステム構成とルーターノード切り替え手順を再検討し、新たな方式を開発した。

図 8 に開発した方式のシステム構成を示す。

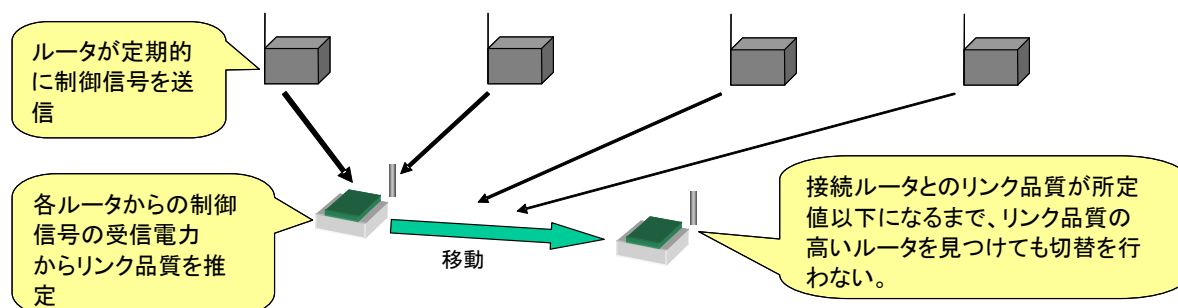


図 8 移動ノード対応方式のシステム構成

開発方式の動作概要は以下の通りである。

- ルーターノードが定期的に制御信号をブロードキャストする。
- 移動ノードは、ルーターノードからの制御信号により近隣ルーターノードおよびそのルーターノードとのリンクの品質を認識する。
- 接続しているルーターノードとのリンク品質が低下した場合のみ接続先を切り替える。
- 接続先の切り替え時にルーターノード探索を行わず、ルーターノードからの制御信号により認識している近隣ルーターノードから適切なルーターノードを選択して切り替えを実行する。

開発方式で、リンク品質がどの程度の値になったときにルーターノード切り替えを行うべきかを検討するために、ルーターノードと移動ノードの距離に対する受信電力値を測定した。測定場所は、ビル内の廊下で、幅 2.2 メートル天井高 2.6 メートル、長さ 52.5 メートル、ルーターノードを天井に設置し、移動ノードは、廊下の中央、高さ 20 センチメートルに設置した。図 9 に測定結果を示す。この結果から、受信電力値にはフェージングによる変動があることが分かる。また、距離減衰の特性を調べたところ、距離の -2.45 乗に比例して減衰することが分かった。

実験で用いたノードで採用している 2.4GHz 帯の ZigBee では、受信感度は -85dBm 以下と規定されている。フェージングによる変動を許容するために、ルーターノード切り替えの許容受信電力値を受信感度より 10dB 大きい値とすると、ル

ーターノードと移動ノード間の距離は 10 メートル程度となることが分かった。この結果より、ルーターノードの間隔が 10 メートルより十分短い場合は、最も近いルーターノードに切り替えなくても、接続しているルーターノードとの距離が 10 メートル程度になるまでは、品質の良い通信が可能であると考えることができる。このように動作させることにより、ルーターノードの切り替え処理に必要なオーバーヘッドを削減することができる。

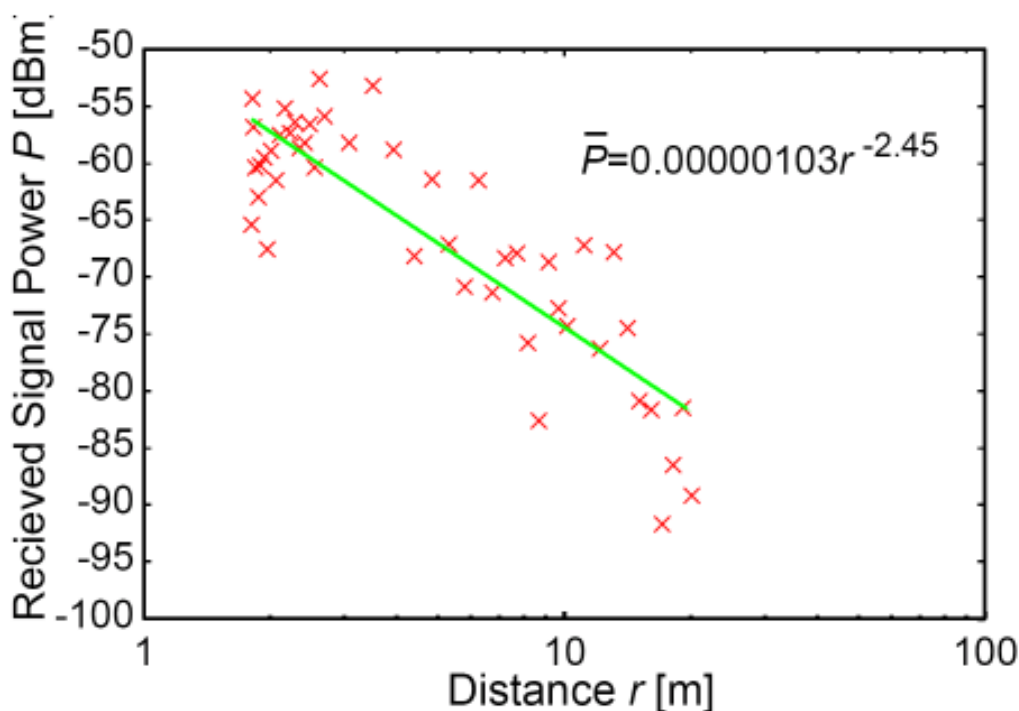


図9 ルーターノードと移動ノード間の距離とリンク品質の関係

さらに、効率よくルーターノードの切り替えを行う方法として、移動ノードの動きを考慮して切り替え先のルータを選択する方法を検討した。ルーターノードの切り替え時に、移動ノードから最も近いルーターノードを切り替え先とするのではなく、移動方向にあるルーターノードでリンク品質が許容値以上のルーターノードを切り替え先ルーターノードとして選択する方法である。この方式により、ルーターノードの切り替え処理を行う回数をさらに削減できると考えている。

また、開発方式を実証実験で使用するために開発したノードに実装した。来年度の実証実験で開発方式を評価する予定である。

② 故障対応

スリープ制御では、ルーターノードを冗長に配置し、交代でスリープすることでルーターノードの省電力化を図るアプローチを取っている。このアプローチでは、今まで動作していたルーターノードが故障により動作を停止する現象と、動作していたルーターノードがスリープ状態に移行する現象を同じ現象としてとらえることが出来るため、故障対応を個別に検討することはせず、スリープ制御と合わせて検討を進めた。スリープ制御で検討している方式を採用することにより、故障が発生した場合でも、スリープ制御が動作しネットワークを再構築可能である。また、中間目標もこの方式で十分達成可能であることを確認した。

しかし、スリープ制御による省電力化と故障対応にかかる遅延時間は、トレードオフの関係にあるため、現在のアプローチには限界があることが分かった。そこで、

スリープ制御とは別に故障対応する方式を検討した。

故障対応方式として、ZigBee 1.0 の機能を使ってできる方式を含め、5 つの方式を検討した。

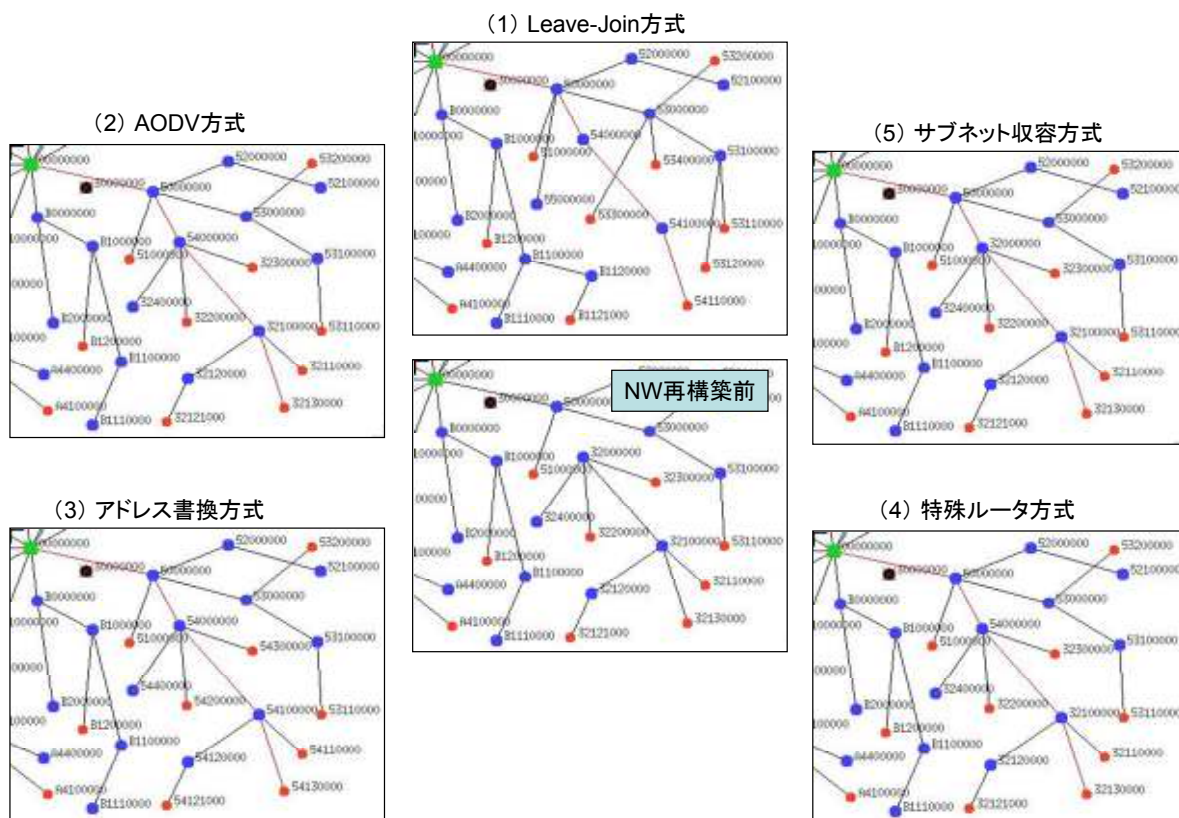


図 10 故障対応方式

図 10 は、あるルータが故障により停止したときに、それぞれの方式で再構築し得られるネットワークを示している。真ん中の図が再構築前のネットワークで、黒丸のルーターノードが故障ノードである。それぞれの方式の動作概要を以下に示す。

1. Leave-Join 方式

故障を検出したノードは、配下のノードを一旦 **Leave** させた後、健全ノードへの **Join** を行う。一方、配下ノードは **Leave** の後、通常の初期シーケンスにより所定ノードへの **Join** を行い、健全状態へと移行する。

2. AODV 方式

故障を検出したノードは、送り先ノードへのルート要求ブロードキャストパケットと、それに応答した送り先ノードのルート応答パケットによりルート探索を行う。その結果ルート上ノードにはルーティング情報が保持され、ルーティングが行われることにより、送り先ノードへの通信が成立する。

3. アドレス書換方式

故障を検出したノードは、配下のノードの接続を保持したまま、健全ノードへの **Join** を行う。ここで付与された新ショートアドレスから書換情報を抽出し配下ノードへユニキャストもしくはブロードキャストにより通知する。配下ノードはその情報を元に、旧ツリーポロジが保持されるようにショートア

ドレスの書換えを行い、健全状態へと移行する。

4. 特殊ルータ方式

故障を検出したノードは、配下のノードの接続を保持したまま、健全ノードへのJoinを行う。上位との通信はここ付与された新ショートアドレスを用い、下位との通信は旧ショートアドレスを用い、その仲立ちを行う特殊ルータを機能させる。さらに、旧ショートアドレス配下ノードへのルーティングを行うため、上位ノードへのルーティング情報通知パケット送出を行い、健全状態へと移行する。

5. サブネット収容方式

故障を検出したノードは、配下のノードの接続を保持したまま、サブネット収容機能を持った健全ノードへのショートアドレス指定Joinを行い、旧ショートアドレスおよび旧ツリートポロジを保ったまま網が構成される。サブネットを収容したノードは、当該サブネット配下のノードとは指定されたショートアドレスを用いて通信を行う。さらに、当該サブネット配下ノードへのルーティングを行うため、上位ノードへのルーティング情報通知パケットの送出を行い、健全状態へと移行する。

表 1 故障対応方式の比較結果

	Leave-Join 方式	AODV 方 式	アドレス 書換方式	特殊ルータ 方式	サブネット 収容方式
標準親和性	◎	◎	△	×	×
故障修復時間	×	△	○	◎	◎
ノードリソース	◎	△	◎	△	△
修復品質	アドレス変化	×	◎	×	○
	トポロジ変化	×	×	○	○
	方式変化	◎	○	◎	△

これらの方式の特性を比較するために、シミュレータを作成した。これまでに、シンクノード付近のルータが故障した場合の動作を解析し、上記5つの方式を比較した。その結果、それぞれの方式は表に示す傾向を持つことが分かった。

今後は、本年度作成したシミュレータを使って、より詳細な評価を行う予定である。

③ スリープ制御

ルーターノードを冗長に配置し、交代でスリープすることによりルーターノードを省電力化するアプローチで検討を進めている。前年度はネットワークの接続性に着目したスリープ制御方式の検討を行った。本年度は、前年度に検討した方式の評価を行う予定であったが、前年度開発した方式は、ZigBee ネットワーク上で動作させるためには、ZigBee プロトコルを大きく改良する必要があることが明らかになった。中間目標レベルだと、基本的な方針は変更せずに、ZigBee ネットワークに容易に適用できる方式を開発できると考え、本年度は前年度の検討を踏まえた上で、ネットワークの接続性の確保及びZigBee ネットワークへの適応を考慮した方式の検討を進めることとした。

ZigBee ネットワークにおいてはクラスタツリー方式によるネットワークが実現されている。これはネットワークをツリー状に形成し、ツリー構造に依存したアド

レスを用いる。構成する全てのノードが木構造を示すパラメータを共有することで、ルーティングテーブルを用いることなく、任意の端末への転送を可能とするものである。以下、木構造の末端（葉）に位置するノードをエンドノード、他のノードをルーターノードと記載する。

本検討はエンドノードのみがスリープ制御を行うことができる ZigBee ネットワークの特徴を利用し、クラスタツリーの特徴を残しつつ、ルーターノードのスリープ制御を可能とするものである。

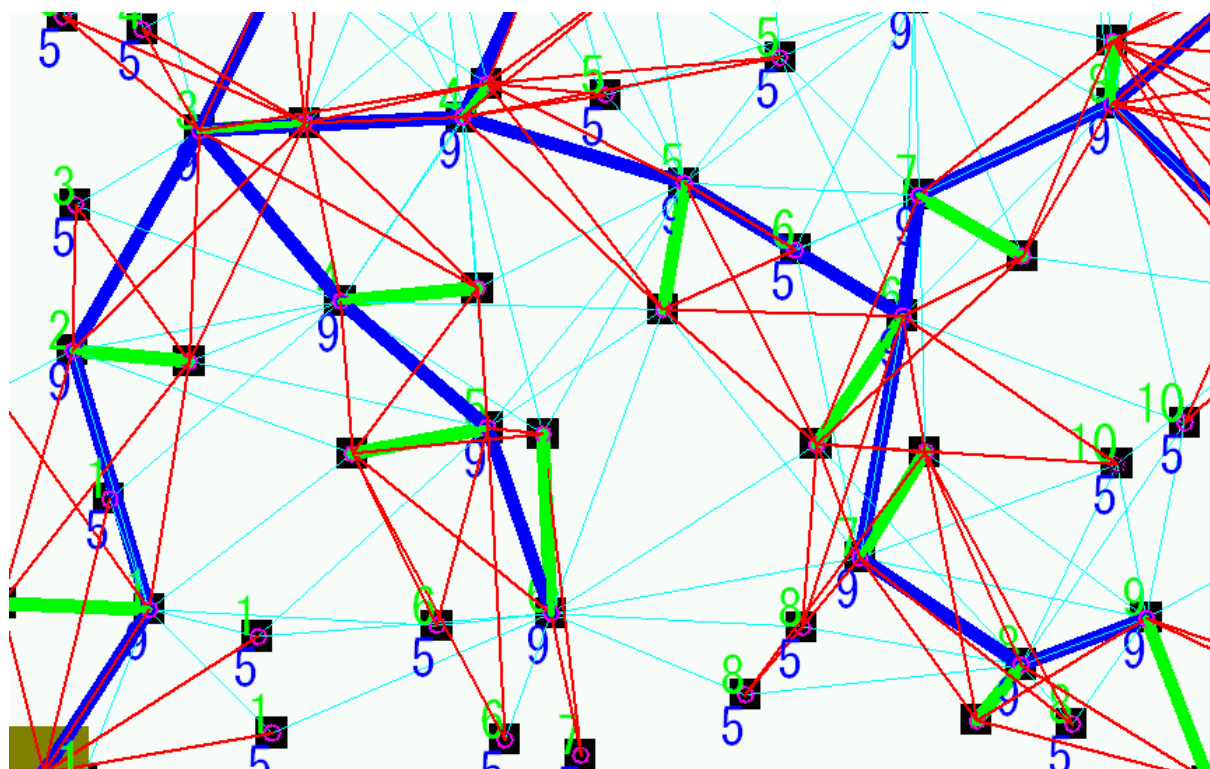


図 11 ルータのスリープ制御を実現するクラスタツリー

今回の検討では、以下のようにしてネットワークの構成とスリープ制御を実現する。

まず、通常の ZigBee の方式に従ってネットワークを構成する。その後、各ノードはブロードキャストの信号を送受信し、直接通信が可能な周辺ノードのリストを作成する。次にルーターノードは周辺のエンドノードと情報を交換し、自身がツリー上で接続している、すなわちパケットの転送を行う必要のあるノード全てと接続可能なエンドノードを選出し、ルーターノードとエンドノードとの間でペアを構成する。ペアとなるエンドノード（以下代替ノード）はルーターノードがスリープ中にルータ機能を代替する。周辺のルーターノードがすでにペアを構成している場合には、代替ノードはルーターノードだけではなく、当該ルーターノードの代替ノードとも直接通信可能である必要がある。

しかしながら、ZigBee の通常のツリー作成方法では上述の条件を満たすエンドノードの選出は困難である。そこで、ツリーを作成する前にノード間で必要な情報を交換し、全てのルーターノードについて、ペアの構成を可能とするツリー形成方式を開発した。結果、ノードの設置密度が一定以上あれば、全てのルーターノードにペアを構成できることを確認した。さらに本方式を用いることによって、ネットワーク中のルーターノード数を削減することが可能となり、ネットワーク全体の消

費電力の削減が実現される。

代替ノードがルーターノード機能を代行する場合には、ルーターノードと代替ノードが通信を行い、2ノードが通信に用いるアドレスを入れ替える。アドレスはツリー上の相対的な位置関係を示しているために、アドレスの入れ替えによって、エンドノードとルーターノードの役割が入れ替わる。ルータ機能を代行されたルータはエンドノードとして動作することが可能となるため、スリープ制御を行う。

図 11 がルーターノードのスリープ制御を実現するクラスタツリーの例である。四角で示されているのがノードで、青線がルーターノード間の接続、緑線がルーターノードと代替ノードを接続している。赤線は代替中のルーターノードが通信に利用する経路を示している。水色線は結ばれたノードが直接通信可能であることを示している。右下端はコーディネータ、青線・緑線が到達していないノードはエンドノードである。

エンドノードは ZigBee で規定されている **indirect transmission** を用いることで大幅な省電力化が可能である。一方、ルーターノードは他の端末から送られるパケットを中継する必要があり、受信待機状態を継続するためにスリープ制御を行っていない。従ってルーターノードの消費電力はエンドノードと比較すると極めて大きい。

ルーターノードは代替ノードがルータ機能を実行している間は消費電力を大幅に抑えることが可能である。図に示した例では全てのルーターノードに代替ノードが存在しているため、ルータ機能を実行する時間と実行しない時間を同じになるように制御した場合には、全てのルーターノードの消費電力を約 50%にすることが可能である。1つのルーターノードに複数の代替ノードを作成することでルータの消費電力をさらに削減することも可能である。

さらに、定期的にネットワークを再構成し、ルーターノード・代替ノードのいずれにもなっていないエンドノードをルーターノードに、ルーターノード・代替ノードをエンドノードとすることで、ルーターノードの消費電力を削減することもできる。

本方式はスリープ制御による省電力化以外に、ルーターノードの故障時における代行ノードの確保、障害物等によるルーターノード間の転送経路の切断への対応が可能となり、ロバストなネットワークの構築にも資するものと考えられる。

④ 無線 LAN との連携

平成 17 年度には、ZigBee と無線 LAN の連携方式として、無線 LAN ノードに ZigBee プロトコルを透過的に中継する機能を備えることで、異なる場所に設置された ZigBee ノード同士をシームレスに接続する方式の検討を行った。

本方式の特長として、

- 1) ZigBee の電波到達距離の増幅機能 (リピータ機能)
- 2) PAN 内のシームレスな接続機能 (ブリッジ機能)
- 3) 無線 LAN のメッシュ化による ZigBee トラフィックの効率化

などが挙げられる。

ZigBee 1.0 で規定されているクラスタツリールーティングでは、パケットは ZR(ZigBee Router)を頂点としたツリー構造に沿ってルーティングされるため、異なる親 ZR に帰属する ZED(ZigBee End Device)間の通信は、ツリー上の共通の親 ZR を経由する必要がある。そのため ZED 間の通信が増加すると、親 ZR での中継負荷が増大し、輻輳が発生する恐れがある。そのため中間目標としている 100 ノードで安定したネットワークを実現するためには、親 ZR へのトラフィック集中を

回避する手段が必要となる。そこで本年度は特長 3 の無線 LAN のメッシュ化を行うことによる ZigBee トラフィックの効率化の詳細検討を行った。図7 は ZigBee ノードとメッシュ機能拡張された無線 LAN ノードとの接続構成例を示しており、ZigBee ノード A~L がそれぞれ 4 箇所設置され、それらのゲエリアを接続する無線 LAN ノード M~P がそれぞれメッシュ状に接続されている様子を示している。ここでは、無線 LAN ノードには ZigBee 通信機能を備えており、ZigBee ネットワークに参加可能であるものとする。

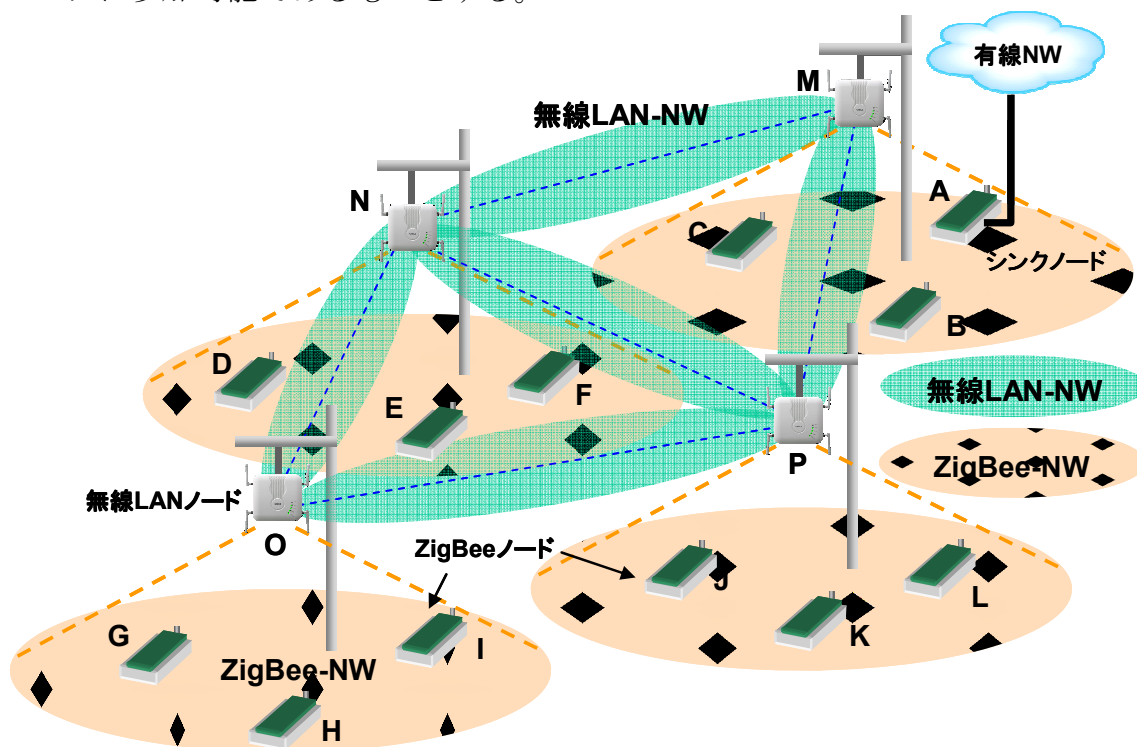


図 12 ZigBee ノードと無線 LAN ノードの接続構成例

例えば ZigBee ノード G が ZigBee ノード L に対してパケットを送信する場合、従来のクラスタツリールーティングでは、パケットは $G \Rightarrow O \Rightarrow N \Rightarrow M \Rightarrow P \Rightarrow L$ の順に転送される。この様子を図 13 の a) に示す。本年度検討した方式では、ZigBee フレームを無線 LAN フレーム内にカプセル化する構成を用いており、これにより、ZigBee フレームを透過的に他の ZigBee-NW に中継することが可能となり、図 13 の b) に示すような論理的なスター型のトポロジーを構成することが可能となる。

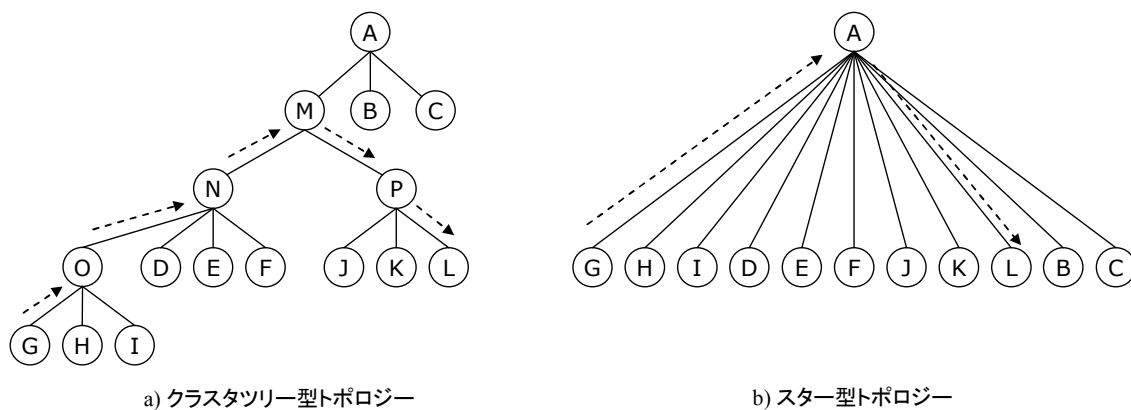


図 13 ZigBee ネットワークトポロジー

図 14 は本方式の無線 LAN ノードの構成とフレーム転送フローを示しており、本中継方式によりクラスタツリーの経路をバイパスするようなルートを形成することができ、経路上に介在する親ノードの中継負荷を低減することが可能となる。

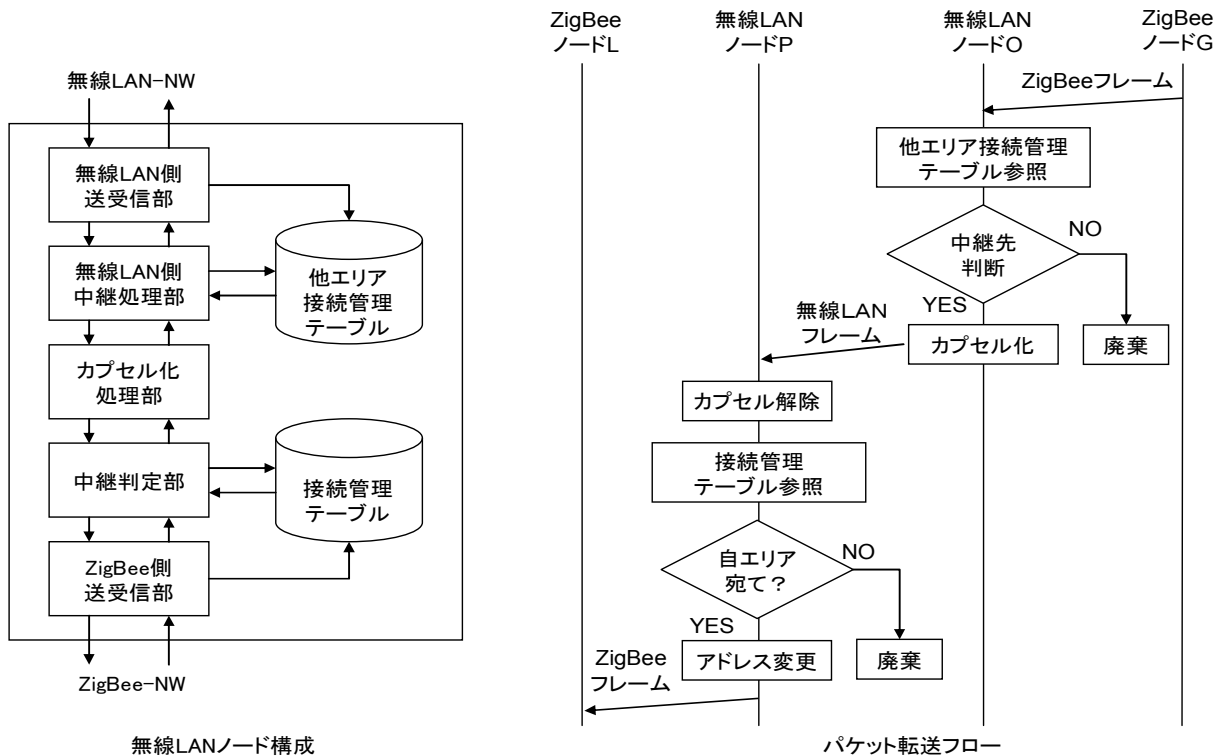


図 14 無線 LAN ノードの構成例とフレーム転送フロー

本方式では ZigBee ノード同士の接続は、一つの親ノードを頂点としたスター型トポロジーを形成することができるため、ZigBee ノード数の増加に対するスケーラビリティを実現できる。例えば無線 LAN のメッシュ化の標準化組織である IEEE802.11s では、メッシュ機能拡張された無線 LAN 装置の数は少なくとも 32 台程度を想定しているため、本方式にこのメッシュ基地局を適用した場合、100 ノード規模の ZigBee ネットワークを構成するには、一つのメッシュ基地局あたり 3～4 台程度で十分に実現可能である。これにより本方式では 100 ノード程度の ZigBee ノード数に対しても安定的なネットワークを構成することが可能であると言える。

⑤ プロファイル

ZigBee Alliance では、アプリケーションプロファイルは Application Framework Work Group (AFG) の下部組織である Profile Task Group (PTG) で策定されている。

本業務委託で対象としている応用例に対して、ZigBee Alliance 内でも標準のプロファイルとして審議が進められている。

現在どのような PTG が存在し、どんな計画で標準化が進められているかは、現時点で ZigBee Alliance 外に公表されていないため、ここでは詳しくは報告できないが、ある PTG において、以下のような作業を行った。

- PTG 立ち上げにあたり、方向性の議論

- 市場要求をまとめた MRD (Marketing Requirements Document)の作成協力
- 当 PTG では、エンドデバイスの移動性を高める必要があるため、その実現方式概要を提示しながら、技術要求 TRD (Technical Requirements Document)の作成協力

今後、作成した TRD にもとづき必要な技術方式の審議も行うことになっており、本業務委託で検討した内容も提案していく予定である。

4-2-3 達成状況

① 移動ノード対応

未達成事項なし。

今後、実証実験で方式を評価する予定である。

② 故障対応

未達成事項なし。

今後、各方式の詳細評価を実施する予定である。

③ スリープ制御

未達成事項はなし。

今後、ルータを 2 重化する方式の ZigBee プロトコル上での動作検証および SPAN ベースの省電力方式を検討する予定である。

④ 無線 LAN との連携

未達成事項はなし。

今後、開発方式のシミュレーションによる評価およびシステム設計時の課題を検討する予定である。

⑤ プロファイル

未達成事項はなし。

今後、方式提案も実施していく予定である。

4-3 実証実験

4-3-1 18年度の開発目標

(1) 目標

実証実験場所を特定するとともに、実証実験システムのシステム設計を完了し、無線ノードへのアプリケーションを実装する具体的作業が19年度に開始可能な状態にする。

(2) 実証実験システム概要

実証実験システムは、研究開発成果を効果的に評価でき、かつ今後の事業展開に有益となる次の4機能を保有するシステムである。

◇ 実証実験システム機能

- ・ 移動 ZigBee 移動ノード特定エリア通過相互検知機能移動
- ・ 移動 ZigBee ノード情報収集機能
- ・ ZigBee ネットワーク監視機能

◇ 性能評価シュミレーションツール

4-3-2 実施状況

(1) 実証実験シナリオ設計

消防活動を想定した実証実験シナリオをまとめる。

- ① 消防隊員が所持する ZED はデバイス ID を格納し、事前にデバイス ID と隊員 ID の関連をホスト PC に登録しておく。
- ② 活動を開始する段階で、ホスト PC、ZC を侵入基点に設置する。
- ③ 消防隊員は多数の ZR を所持し、基点より順次建物内に侵入する進路に沿って、ZR を地面又は壁に設置し、電源を ON する。
- ④ ZR が無線ネットワークに Join すると、発生音と共にランプが点滅する。
- ⑤ 進入基点のホスト PC には、進入隊員 ID、進入時刻、接続 ZR ノード番号が表示される。
- ⑥ 隊員の進入後、ホスト PC 画面に経過時間が表示される。
- ⑦ 消防隊員が所持する ZED の温度センサ及び加速度センサの値が指定範囲を超えたとき、ホスト PC に Alert 通知する。
- ⑧ 消防隊員が所持する ZED の温度センサ及び加速度センサの値が指定範囲を超えたとき、ホスト PC に Alert 通知する。

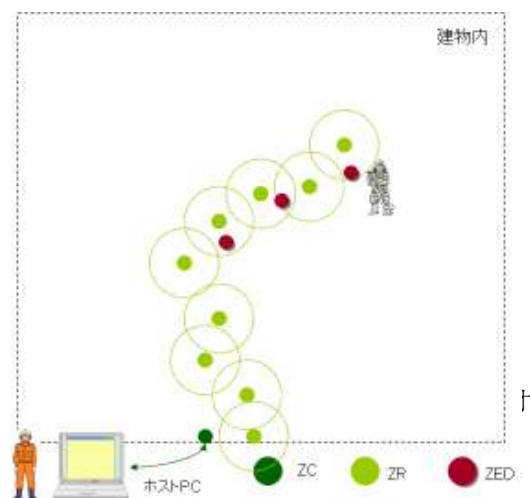


図 15 実証実験シナリオイメージ図

(2) 目的サービスパターンとの対応

上記の実証実験シナリオによって、所期の目的であった3つのサービスパターンの実証を同時に行う。

① 移動 ZigBee ノード近接相互検知サービスパターン

実証実験シナリオ NO⑤の「進入基点のホスト PC には、進入隊員 ID、進入時刻、接続 ZR ノード番号が表示される」によって近接相互検知サービスパターンを実現する。

② センサ ZigBee 固定ノード緊急事態検知サービスパターン

実証実験シナリオ NO⑦の「設置した ZR の温度センサ及び加速度センサの値が指定範囲を超えたとき、ホスト PC に Alert 通知する」によってセンサ ZigBee 固定ノード緊急事態検知サービスパターンを実現する。

③ 移動 ZigBee ノード緊急事態検知サービスパターン

実証実験シナリオ NO⑧の「消防隊員のウェアラブルな装置として ZED にディスプレイを接続し、進入情報、退路情報、行方不明の隊員情報を表示する」によって移動 ZigBee ノード緊急事態検知サービスパターンを実現する。

(2) 構成設計

ZigBee スタックアーキテクチャの表現に準じ、各ノードの機能構成を下図に示す。

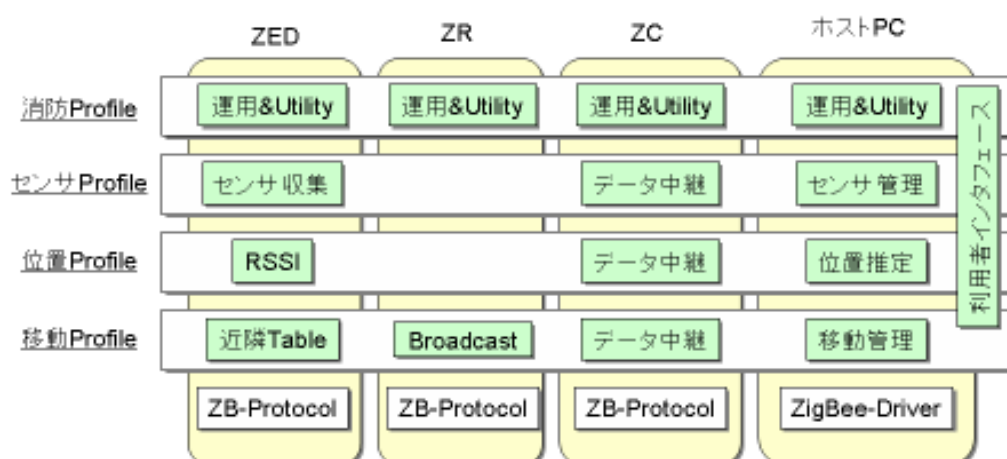


図 16 実証実験システム機能構成

いくつかのプロファイルからアプリケーションは構成される。上図のプロファイル分割は論理的なもので、実装を制約するものではない。移動プロファイルと位置プロファイルは1つのプロファイルになるかもしれないが、位置プロファイルの位置計算をZEDで処理する形態をオプションと考え、論理的には分けておく方がよい。

消防プロファイルは消防アプリケーションに関する運用をつかさどる。その他のプロファイルは、ネットワーク正常系の基本機能である。消防プロファイルからこれら3つの基本機能プロファイルを起動(Call)する。起動のタイミングは消防プロファイルでシステム性能を考慮して作成する。消防プロファイルは、これ以外に、ヘルスチェック、故障復帰、システムログ、時刻同期のユーティリティ機能をもつ。ノード障害や性能劣化要因を調べるときに必要なプロファイルである。各プロファイル間の関係を下図に示す。プロファイル・インターフェースは、実証実験の構築段階で、担当者間で定められるものである。

ホスト PC の処理内容は ZC 処理の延長と考えることができる。もし ZC にユーザイ

インタフェースを記述する機能があれば、本消防アプリケーションはスタンドアロンなので、ホスト PC は不要になる。本実証実験システムにおいて外部ネットワークとの接続はない。

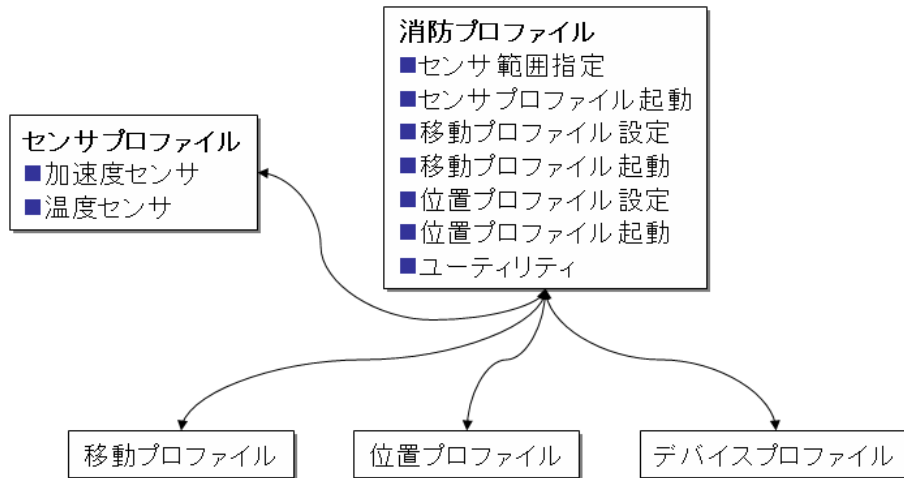


図 17 消防プロファイルと他のプロファイルとの関連

(3) 機能設計

実証実験システムのアプリケーションは、消防プロファイル、センサプロファイル、位置プロファイル、移動プロファイルから構成される。ここで消防プロファイルはユーティリティを含む。プロファイルのソフトウェアは、ホスト PC と ZC、ZR、ZED の各ノードに配置される。以降、ホスト PC のアプリケーション機能、および消防プロファイルのユーティリティ機能を説明する。消防プロファイルの運用機能は、運用シナリオの更なる詳細化により明らかにする。

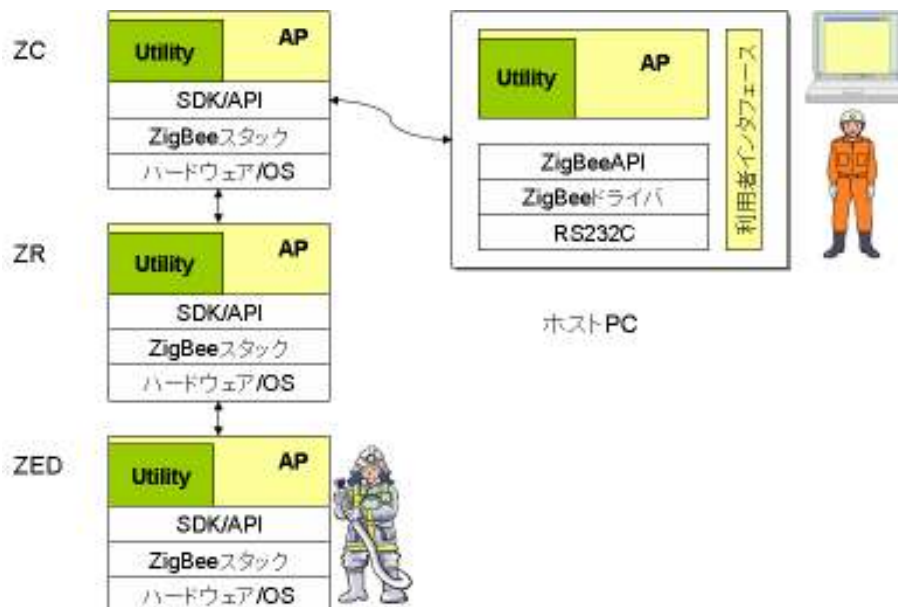


図 18 アプリケーション機能実装イメージ

(あ) ホスト PC のアプリケーション

消防アプリケーションは、ZigBee ノードを事前に固定設置して使用する形態ではなく、その都度アドホックに ZigBee ノードを配置して一時的に使用する。また、現場に到着したら PC の電源を ON にし、本アプリケーションを起動する。そのため 24 時

間常時運用し続けるサーバタイプのソフトウェアよりも、使用する時に起動して作業が完了したら終了するスタンドアロンタイプの実装が適している。ホストアプリケーションやユーティリティは、ZigBee ドライバ上に構築する。ドライバ機能要件は次の通り。

- マルチスレッド対応
- 同期処理
- フラグメンテーション対応
- 紐付け
- タイムアウト
- 同期 I/O or 非同期 I/O
- java API 対応
- RS 対応

また、ホストアプリケーションのユーザインタフェースには次の項目がある。

- 消防隊員が所持する移動ノード ZED の管理
- 入退出管理
- ネットワークの状態

(い) ヘルスチェック

ヘルスチェックは、ZigBee ノードの故障検出を行う。ここでの故障検出はネットワークにアクセスできなくなったノード検出のことを意味する。消防サービスへの利用では、日ごろの定期点検や消防活動を実施している途中でヘルスチェックを行うなどの利用がある。3つの処理方式（ホスト PC からの Ping 方式、ノード自律方式、相乗り方式）を検討した。実証実験システムに対しては、①ネットワーク負荷軽減の必要性、②実証実験結果評価が行いやすいという観点から、ZigBee ノード自律方式を採用することとした。以下にノード自律方式について説明する。

(a) ZigBee ノード自律方式

「ホスト PC からヘルスチェックを行う方式」は、ZC から各ノードに ping を送信するので、ZC から遠くなるほどマルチホップのためのネットワーク負荷が生じることになる。その負荷を改善した「ZigBee ネットワーク内で自律的にヘルスチェックする方式」について記述する。

「ホスト PC 側でヘルスチェックを行う方式」との違いは、親ノードが子ノードに対して定期的にヘルスチェックメッセージを送信する点である。故障であると判断された場合は故障イベントをホスト PC へ送信する。ヘルスチェックプログラムはホスト PC 以外に、ZC, ZR, ZED などのノード上に実装する。

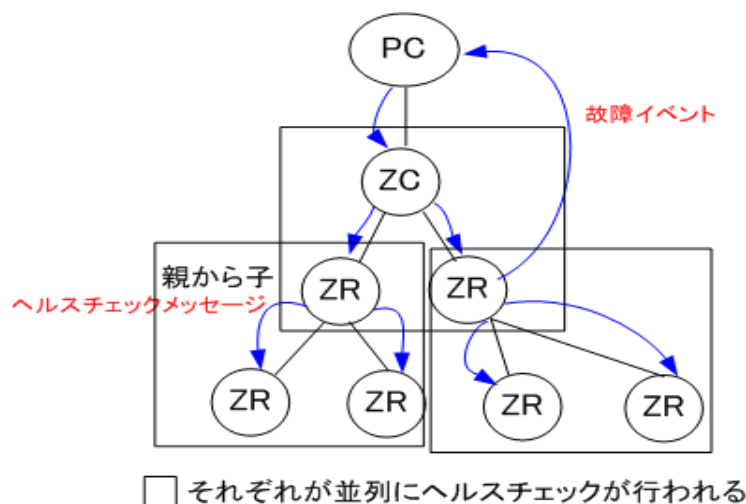


図 19 ZigBee ノード自律方式

実装条件として、自身に接続している子ノードに関する情報を、アプリケーションで参照できる必要がある(Network Information Base 中の Neighbor table)。

(う) 故障復帰

ノードが故障した場合自動的に故障検出をし、他のノードへ自動的に接続を変更してネットワーク動作を継続する。この機能は ZC,ZR,ZED 上に実装する。

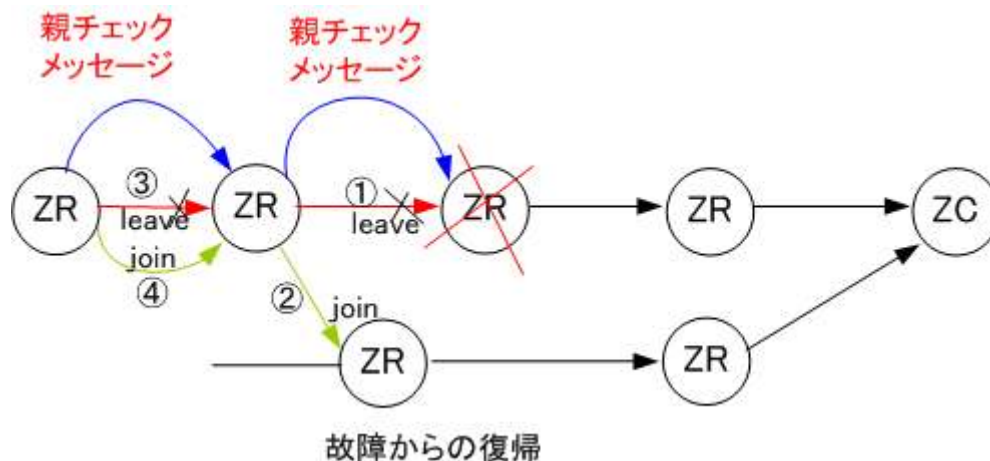


図 20 故障からの復帰機能

子ノードは親ノードに対して親チェックメッセージを定期的を送信する。応答がなければ、親ノードが故障したと判断し、leave を行う①。次に join を試みる②。①②の leave/join が行われると、その子ノードの親チェックメッセージ送信がエラーとなるので、同様に③④の leave/join が行われ、故障復帰が行われることになる。

消防への応用を考えると、人の安全に関わることであるので故障からの自動復帰は重要な機能である。そのためには、ノードを冗長的に配置して自動復帰できるようにすることが重要である。

(え) ログ

ZigBee ノードの動きを調べる方法を提供するために、ZigBee ノード内にアプリケーションのログを記録する。ログを記録するライブラリを作成し、SDK 呼び出しの結果やアプリケーションの状態を記録する。ログは ZigBee ノード上の RAM エリアに保存し、必要があった時、PC(ホスト)へ転送して PC 上で内容を確認する。またこのログには、時刻同期されたタイムスタンプを添付する。

(お) 時刻同期

各 ZigBee ノードの時刻同期方式として、3つの処理方式(クライアントサーバ同期方式、RBS 同期方式、RBS 改良同期方式)を検討した。実証実験システムに対しては、①ネットワーク負荷軽減の必要性、②厳密な時刻同期は求めないという観点から、クライアントサーバ同期方式を採用することとした。以下にクライアントサーバ同期方式について説明する。

(a) ZigBee ノードのクライアントサーバ同期方式

SNTP の様なクライアントサーバ方式を使用して ZigBee ノードの時刻同期を行う。基本的な方式は、ノードの電源がオンしてから最初の join 後に親ノードから時刻を取得して自信のローカルクロックに設定する。その際、親ノードは時刻同期サーバとし

て働き、自身はクライアントとして働く。時刻設定プログラムをノード上に実装する。

動作は次の通り。最初に ZC の電源オン後、ホスト PC の時刻を ZC に設定する①。次に ZR が ZC に join し、ローカルクロックの時刻設定が行われていなければ、ZC に対して時刻を要求して、時刻設定処理を行う②。ZED が join した時も同様の処理が行われ、時刻設定を行う。親ノードの時刻設定が行われていない場合に、子ノードが時刻要求に来る場合があるので、その場合は子ノードの時刻設定は行わないようする。このような制御によって、PC の時刻が ZC に設定され、最初の join によって子ノードに時刻設定が順に行われる。



図 21 ZigBee ノードのクライアントサーバ時刻同期方式

電源オン後の時間が長く続くと各ノードのローカルクロックの誤差が蓄積される。そのため長時間運用する場合、定期的にホスト PC から ZC に対して時刻の再設定命令を発行するようにする。時刻の再設定も ZC から子ノードに向かって伝播し、全体の設定が行われるようにする。

本方式は、比較的制御が容易で CPU やネットワーク負荷への影響が少ないという利点がある。アドホックなネットワーク環境においても柔軟に働く。欠点としては、ZC からのホップ数が増えるほどに設定値の誤差が増加する。データ送信を頻繁に行っているシステムの環境下で使用すると、時刻設定に要する通信は 1 ホップですが、1 ホップ分のバックオフ等の待ち時間が場合によっては時刻設定の誤差に含まれることになる。

(4) 性能設計

(あ) シミュレーションの目的

(a) ZigBee ネットワークアドレスの設定

ZigBee 仕様に基づいてアドホックネットワークを構築する際には、IEEE アドレス (MAC アドレス) やデバイス ID とは別に ZigBee ネットワークアドレスと呼ばれるアドレスが各ノードに割り振られる。この ZigBee ネットワークアドレスはパラメータとして与える 3 つの値、すなわち最大子ノード数 (maxChildren) , 最大ルータノード数 (maxRouters) , 最大階層数 (maxDepth) の値によりできる最大のツリー構造の各リーフに一意的にアドレスを割り振っている。各ノードの ZigBee ネットワークアドレスはノードがネットワークに join する際に親ノードのアドレスと階層数、親ノードにすでにぶら下がっている子ノードの数、それと先に挙げた 3 パラメータから算出される。実際には常に最大の tree 構造を構成するわけではないので、必要なノード数を超えるアドレスは間引く形となる。

本シミュレーションではまず ZC ノードを、次に ZR ノードを順次 join させることにより、あらかじめ与えたパラメータの値にしたがって各ノードに ZigBee ネットワークアドレスが割り振られる様子を仮想的に再現することを目的としている。

シミュレーションの結果からはネットワーク形状の基本的な情報として、ノードが join できたかどうか、ノードの親子関係や深さ、子ノードの数などが予測できる。また、ノード数やネットワークを張る対象のエリアの広さや形状に応じたパラメータの

最適値を策定する手がかりにもなる。

(b) パケット送信による輻輳

ネットワークが張られた後、各ノードからパケットが一斉に送信される。ここでは先の ZigBee ネットワークアドレスの設定シミュレーションによって ZigBee ネットワークアドレスを与えられた各ノードから一定時間間隔で複数回パケットを送信した際のパケットの輻輳状況を仮想的に再現している。

今期のシミュレーションではまず、パラメータとして設定できるパケット送信間隔や回数、データサイズ、ノード内部でのパケット転送に要する時間などを変更して試行することにより、パラメータ値の変更がどの程度結果に影響するのかを予測するのが主な目的になる。したがって ACK の戻りや各ノードの送信バッファの容量などを考慮しない条件でのシミュレーションを行い、データ損失率の楽観的予測値を算出する。

出力データとしては、各ノードからのパケット送信状況(パケット送信時刻、バックオフの回数とバックオフがかかった時刻、親ノードへの到着時刻、ZC への到着時刻、送信の成功・失敗)、パケット送信状況の時系列データ、パケット損失率(ノード別、深さ別)がある。

(い) シミュレーションモデル

(a) パラメータ

(ア) 領域とノードに関するパラメータ

シミュレーション領域はメッシュで覆い、このメッシュの格子点上にノードを置いていく。設置するノードの位置は格子点番号(左上から行番号・列番号の2つの整数の組)で表す。各 ZR ノードは join する順に従って親ノードが決定し ZigBee アドレスが付与されるため、ZR ノードが join する順序もパラメータとして設定する必要がある。したがって、ノードを置く格子点の番号を join する順番にパラメータ `node` に設定することでノードの位置と join 順序を定める。

メッシュの間隔は一律であるものとしてパラメータ `mlength` に設定する。更に全ノードは同一の電波強度を持つものと仮定し、電波の届く範囲をノードを中心とした円とみなしその半径をパラメータ `pradius` に設定する。

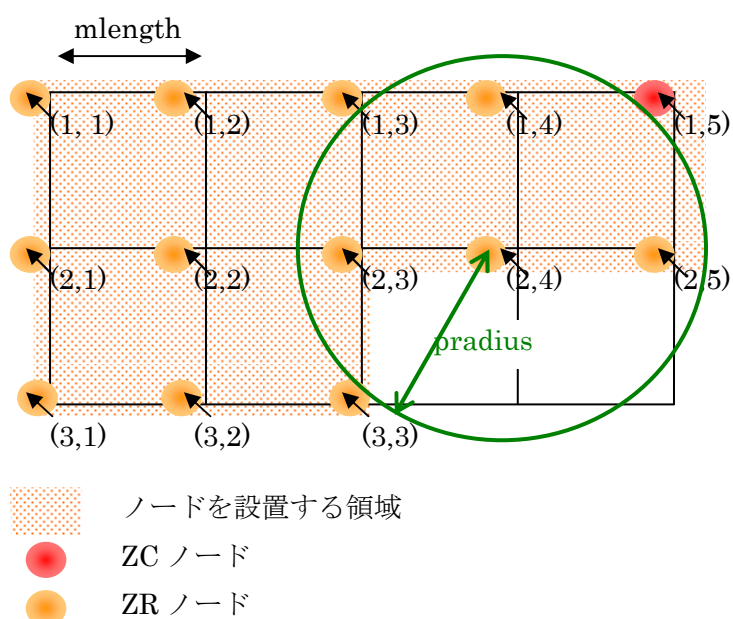


図 22 シミュレーション領域のノード構成

表 2 ZigBee ノードに関するパラメータ

パラメータ	値(データ型)	説明
latticeRow	1 以上の整数(int)	領域を覆うメッシュの行数
latticeClm	1 以上の整数(int)	領域を覆うメッシュの列数
node	$(x_1, y_1) (x_2, y_2) (x_3, y_3) \dots$ $(x_i, y_i) \dots$ $x_i=1, \dots, \text{latticeClm}$ $y_i=1, \dots, \text{latticeRow}$	ノードの格子点番号 (左上から行番号・列番号の 2 つの整数の組) を join する順に並べた列。 (x_1, y_1) は ZC。
mlength	0 より大きい有理数(double)	メッシュの間隔 単位はメートル
pradius	0 より大きい有理数(double)	ノードから電波の届く範囲の最大値 単位はメートル

(イ)NW 層に関するパラメータ

nwkMaxChildren, nwkMaxRouters, nwkMaxDepth の各値を同名のパラメータとして設定する。これらの値は ZigBee Alliance による ZigBee Specification で示すとおり 1 以上の整数値として設定することができるが、値が極端に小さい場合には ZigBee ネットワークアドレスが振られないノードがでてくる。本シミュレーションでも同様な動きとなる。

表 3 NW 層に関するパラメータ

パラメータ	値(データ型)	説明
nwkMaxChildren	1 以上の整数(int)	1 ノードに対する ZED タイプの子ノード数の最大値
nwkMaxRouters	1 以上の整数(int)	1 ノードに対する ZR タイプの子ノード数の最大値
nwkMaxDepth	1 以上の整数(int)	ネットワークの深さの最大値 ZC は深さ 0 とする。

(ウ)MAC 層に関するパラメータ

macMaxCSMAbackoff, macMinBE, macMaxBE の各値を同名のパラメータとして設定する。これらの値は CSMA-CA シーケンスに対応する。

表 4 MAC 層に関するパラメータ

パラメータ	値(データ型)	説明
macMaxCSMAbackoff	1 以上の整数(int)	CSMA-CA パラメータ
macMinBE	1 以上の整数(int)	CSMA-CA パラメータ
macMaxBE	1 以上の整数(int)	CSMA-CA パラメータ

(エ)データサイズと転送時間に関するパラメータ

各ノードから送信するデータのサイズは同一であるものとする。またデータサイズはデータ送信所要時間を算出する目的でのみ考慮するパラメータであるため、データ

サイズをパラメータとして与えるかわりにデータ転送所要時間としてパラメータに設定する。

ノード内での転送時間については、NW層-APS層-AP層の間の転送時間は考慮していない。したがってAP層から発生したデータの内部転送所要時間はNW層-MAC層を通過するのに要する時間としている。またデータ転送(他ノードから受信したデータの送信)の際の内部転送所要時間はMAC層-NW層-MAC層を通過するのに要する時間とし、それぞれパラメータ

`internalProcessingTimeForAP`

`internalProcessingTimeForTransfer`

として設定する。

さらに、送信ノードのMAC層から受信ノードのMAC層にたどり着くまでの所要時間は`transferTime`として設定する。

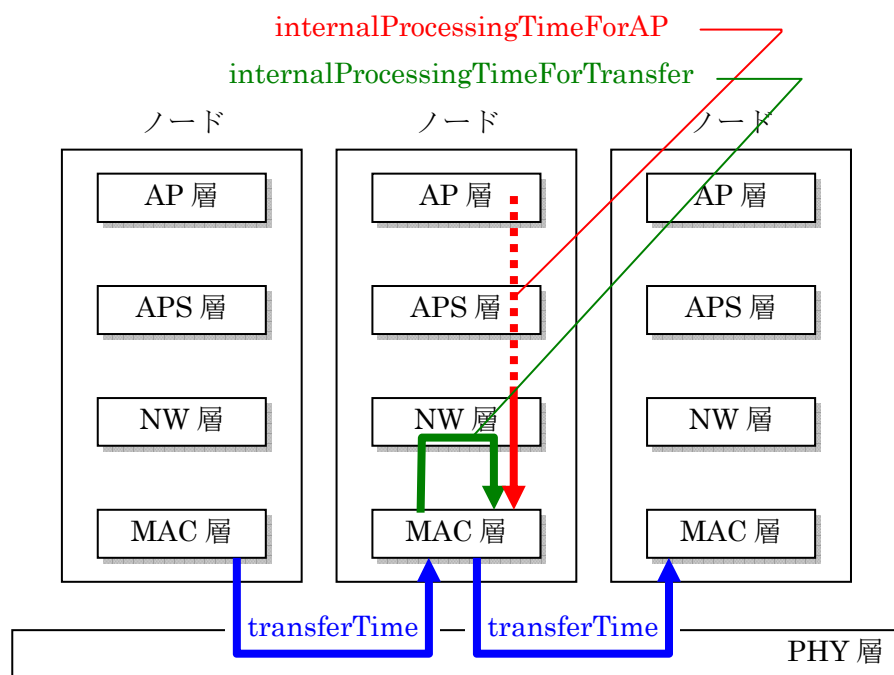


図 23 データ転送所要時間に対する考え方

表 5 データ転送時間に関するパラメータ

パラメータ	値(データ型)	説明
<code>transferTime</code>	0 以上の double 値	図 5.2.1 参照 単位はミリ秒
<code>internalProcessingTimeForAP</code>	0 以上の double 値	図 5.2.1 参照 単位はミリ秒
<code>internalProcessingTimeForTransfer</code>	0 以上の double 値	図 5.2.1 参照 単位はミリ秒

(オ)コスト計算に関するパラメータ

ノードがネットワークに `join` して親ノードを選択する際にはコスト計算を行う。コストは親候補ノードとの距離と、親ノードの ZC からの `hop` 数の各値にパラメータをかけた値の和として算出する。

コスト = 親候補ノードとの距離 $\times \alpha$
 + 親候補ノードの ZC からの hop 数 $\times \beta$
 このときの α と β をそれぞれパラメータ `coeffLQI`, `coeffHOPS` として設定する。

表 6 コスト計算に関するパラメータ

パラメータ	値(データ型)	説明
<code>coeffLQI</code>	0 以上の double 値	コスト計算式の距離の係数
<code>coeffHOPS</code>	0 以上の double 値	コスト計算式の HOP 数の係数

(カ) パケット送信間隔と回数に関するパラメータ

各ノードの AP 層からは一定間隔でパケットが送信される。間隔はパラメータ `PacketInterval` で設定する。各ノードの AP 層からパケットを送信する回数も全ノード一律として、パラメータ `NumberOfSending` で設定する。

また、各ノードの AP 層から 1 回目のパケットを送信するタイミングにばらつきを与える場合には、パラメータ `timelag` を設定します。各ノードの AP 層から送信される 1 度目のパケットは 0 から `timelag` の間でランダムに 1 度送信され、以降の AP 層からのパケットは `PacketInterval` で指定した間隔で送信される。特に、`timelag` を 0 とすると各ノードの AP 層からパケットが一斉に送信される。

表 7 パケット送信間隔と回数に関するパラメータ

パラメータ	値(データ型)	説明
<code>NuberOfSending</code>	1 以上の整数(int)	ZR ノード AP 層からのパケット送信回数
<code>PacketInterval</code>	double	パケット送信間隔 単位はミリ秒
<code>timelag</code>	int	パケットの初期ばらつき 単位はミリ秒

(b) モデル

(ア) ZigBee ネットワークアドレスの設定シミュレーションのモデル

ノードはシミュレーションパラメータ `node` で定めた順序で `join` を開始し、順次 ZigBee アドレスを取得する。ただし、最初に `join` するノードは ZC で、ZC の ZigBee アドレスは ZigBee Specification に則って 0 となる。

他の ZR ノードは以下の図に示す手順で親ノードを決定する。親ノードが決まると、親ノードの ZigBee アドレスと階層数、パラメータ `nwkMaxChildren`、`nwkMaxRouters`、`nwkMaxDepth`、更に親ノードにすでにぶら下がっている子ノードの数から自身の ZigBee アドレスを決定する。親候補がない場合には接続できるノードがないと判断し `join` 失敗となる。

親ノードの N 番目の子ノードである ZR の ZigBee アドレスは $A(N)$ は親ノードのアドレス A_{parent} から以下の計算式で得られる。

表 8 子ノード ZR の ZigBee アドレス取得計算式

$$A(N) = \begin{cases} A_{parent} + \frac{1 + C_m \cdot R_m - C_m \cdot R_m^{L_m - d + 1}}{1 - R_m} (N - 1) & R_m \neq 1 \\ \{1 + C_m \cdot (L_m - 1)\} \cdot (N - 1) & R_m = 1 \end{cases}$$

A_{parent} : 親ノードのアドレス
 C_m : nwkMaxChildren(最大子ノード数)
 R_m : nwkMaxRouters(最大子ルータ数)
 L_m : nwkMaxDepth(最大階層数)
 d : 親ノードの深さ

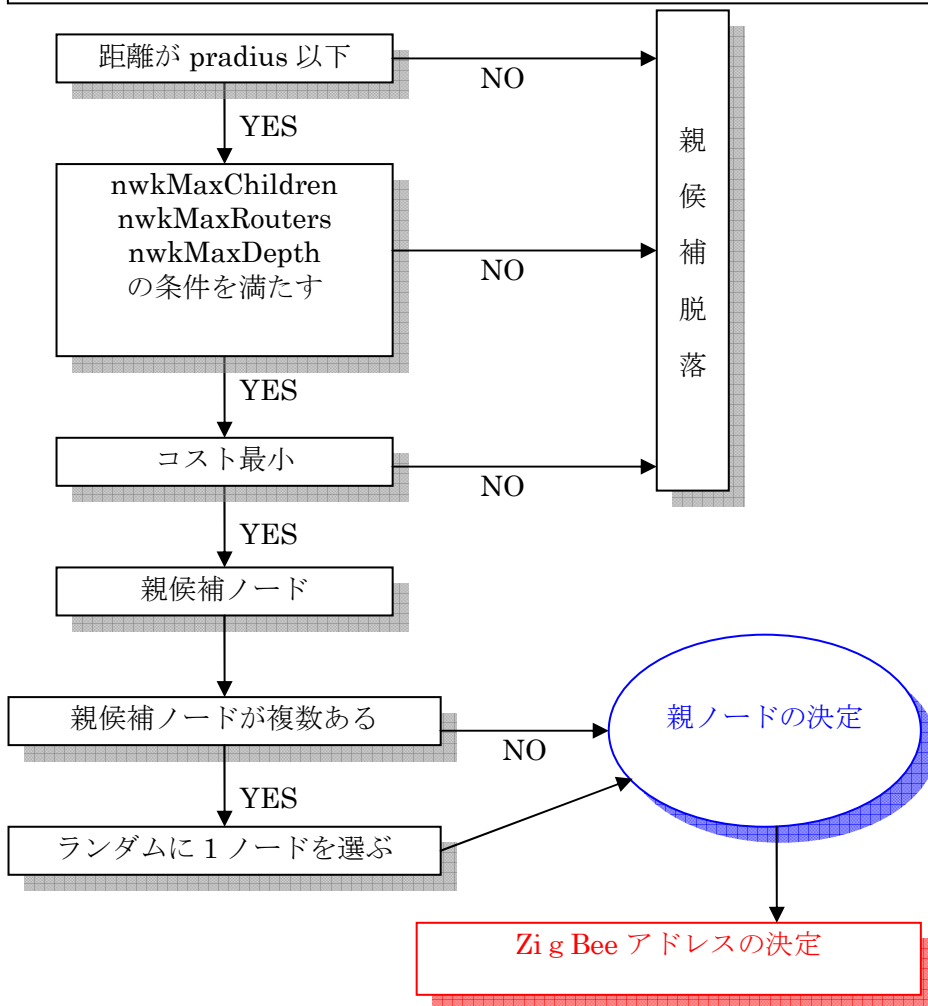


図 24 ZigBee アドレス決定処理プロセス

(イ) パケット送信による輻輳シミュレーションのモデル

パケットはデータ発生元の ZR ノードから親ノードを次々にたどって ZC へ送信される。AP 層から発生したパケットを親ノードへ送信する場合についても、他ノードから受け取ったパケットを親ノードへ送信する転送処理の場合についても、データはまず MAC 層にある送信バッファに保存される。送信バッファには MAC 層に到着し

た時刻に従ってパケットが保存される。仮にバッファ先頭のパケットに **backoff** がなかったとしてもバッファに保存されたパケットの順序が変わることはない。

以下の図ではパケットが送信元ノードの **MAC** 層に到着してから送信先ノードの **MAC** 層に送信されるまでのフローを示す。

- ※1 本シミュレーションでは各ノードの送信バッファの容量は無限大とする。
- ※2 パケットの輻輳を検出する処理では、**ACK** を確認しない代わりに隠れ端末などによる現実には検出できない輻輳も **backOff** の対象とする。
- ※3 **backOff** 回数が 0 回の場合には無条件に 1 度 **backoff** を行う。
バックオフのロジックは下の図のとおり。
- ※4 1 **backoff** 単位時間は $20\text{Symbols} = 20 \times 16 \mu\text{s}$
($\text{SymbolRate} = 62.5\text{k symbols/s}$ より $1\text{symbol} = 1/62.5\text{K} = 16 \mu\text{s}$)
- ※5 $\text{CCA} = 12\text{symbols} = 12 \times 16 \mu\text{s} = 192 \mu\text{s}$

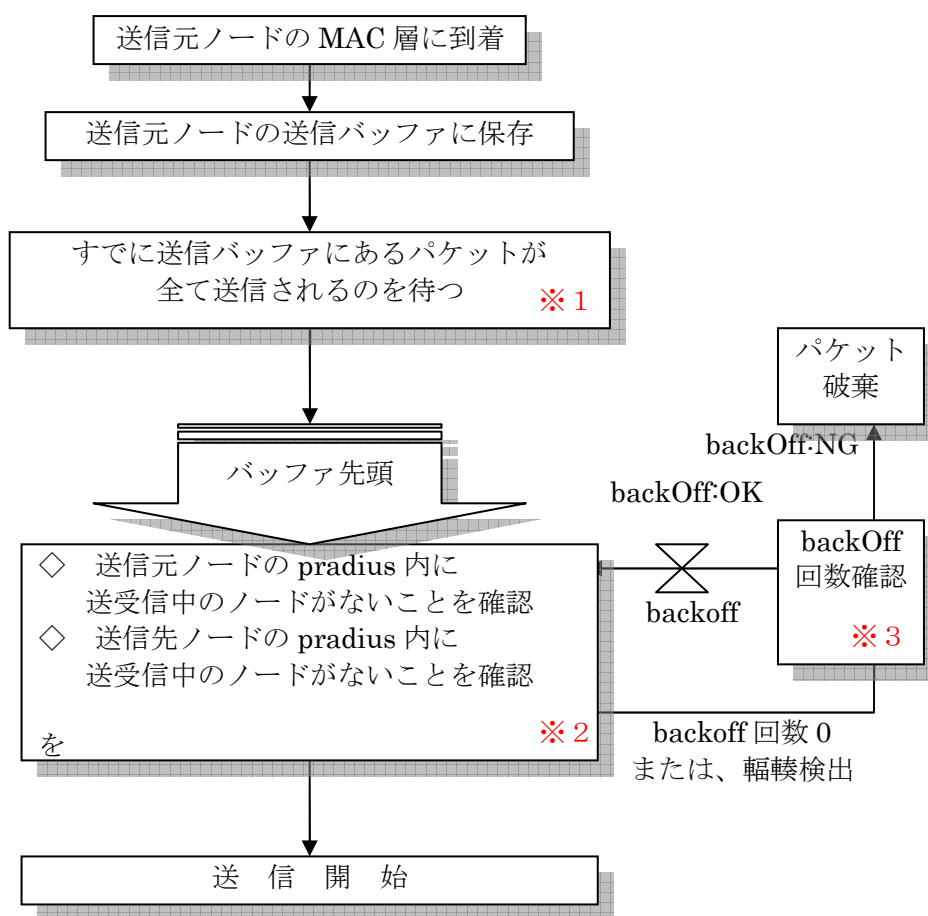


図 25 MAC 層間の送信処理フロー

(5) シミュレーション実行結果と評価

上述のシミュレーションツールを利用して、机上で **ZigBee** ノードの動作シミュレーションを行った結果とその評価を以下に示す。

(あ) ZB アドレス設定の実行結果

シミュレーション領域を広げたエリアで **ZB** アドレス設定シミュレーション結果を次にしめす。ここでの主な設定値は以下のとおり。

表9 シミュレーション設定値

nwkMaxChildren	5
nwkMaxRouters	5
nwkMaxDepth	5
Mlength	5
latticeRow	4
latticeClm	4
Pradius	10
coeffLQI	1
coeffHOPS	0

(a) 単純な長方形のモデルにおけるシミュレーション結果

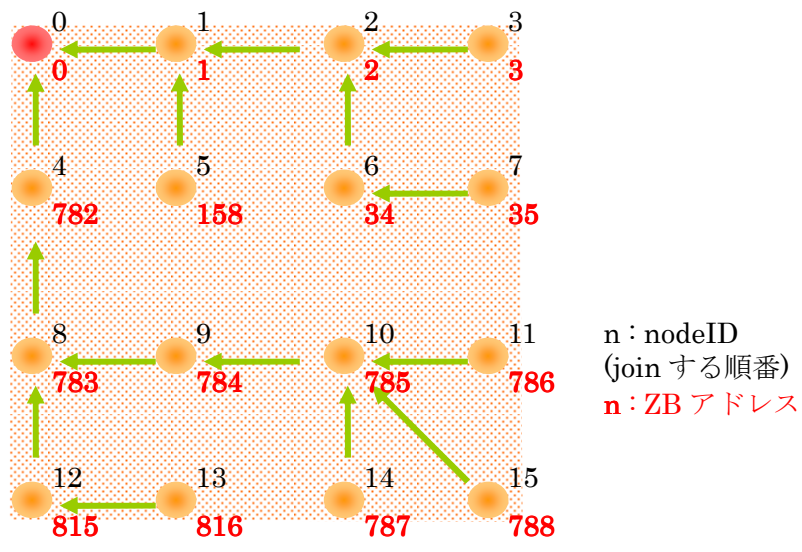


図 26 単純な長方形のモデルにおける ZB アドレス設定シミュレーション結果

(b) L字型のモデルにおけるシミュレーション結果

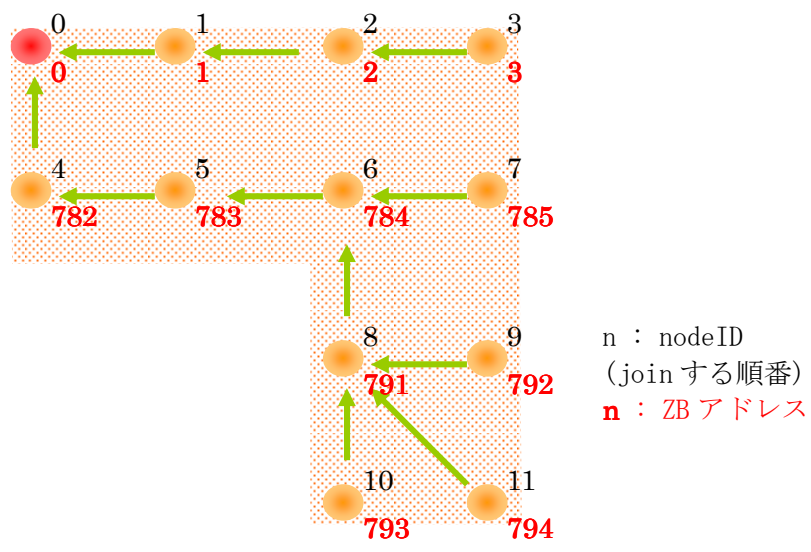


図 27 L字型のモデルにおける ZB アドレス設定シミュレーション結果

(c) 深さが深くなった場合におけるシミュレーション結果

表 17 のように値を設定し、深さが深くなってもしかるべきトポロジーでネットワークが構成されることを実証した。

表 10 シミュレーション設定値

nwkMaxChildren	5
nwkMaxRouters	5
nwkMaxDepth	8
mlength	5
latticeRow	3
latticeClm	8
pradius	6
coeffLQI	1
coeffHOPS	0

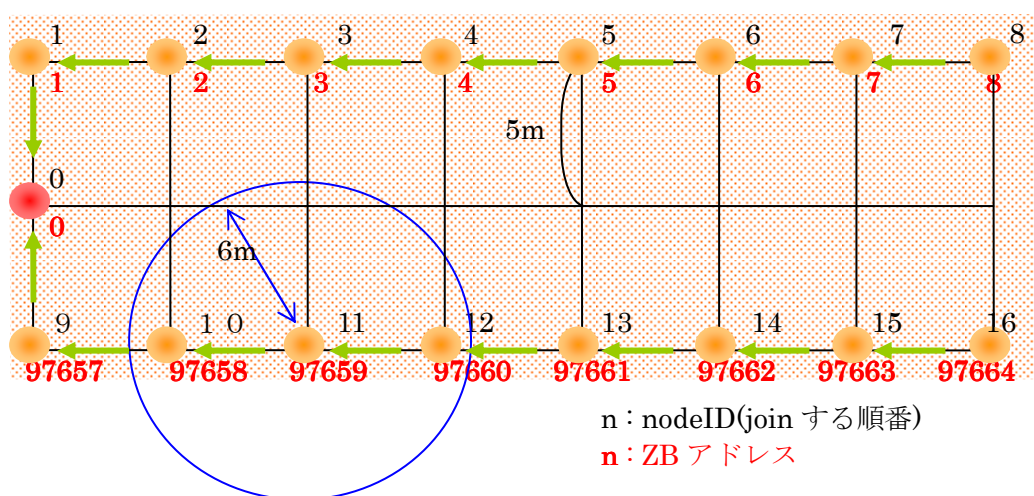


図 28 深さが深くなった場合の ZB アドレス設定シミュレーション結果

(い) 輻輳シミュレーションに関する考察

4×4 のメッシュ上の全格子点にノードを置いたモデルで、全ノードからパケットを一斉に送信したシミュレーションの結果について考察する。共通の設定値は次のとおり。

表 11 輻輳シミュレーションのための共通パラメータ設定値

nwkMaxChildren	5
nwkMaxRouters	5
nwkMaxDepth	5
mlength	5
latticeRow	4
latticeClm	4
pradius	10
coeffLQI	1
coeffHOPS	0
timelag	0
CCATime	0.192
macMaxCSMAbackoff	4
macMaxBE	5
macMinBE	3

NumberOfSending	100
-----------------	-----

ZigBee ネットワークによる一般的なデータ送信のヘッダは 20~30byte なので、ヘッダサイズを仮に 20byte とするとデータサイズは最大 108byte となる。そこでヘッダのサイズを 20byte とし、データサイズ最大の場合(108+ヘッダ 20byte)、半分の場合(54+ヘッダ 20byte)、3分の1の場合(36+ヘッダ 20byte) のそれぞれについてシミュレーションを行った。データサイズに関する設定値は以下の通りとなる。

表 12 データサイズに関する設定値

	最大の場合 (108+20byte)	半分の場合 (54+20byte)	3分の1の場合 (36+20byte)
TransferTime	20	1.15325	0.875
InnerProcessingTimeforAP	10	5.78125	4.075
InnerProcessingTimeforTransfer	20	11.5625	8.15

(ア)送信間隔が 500msec の場合

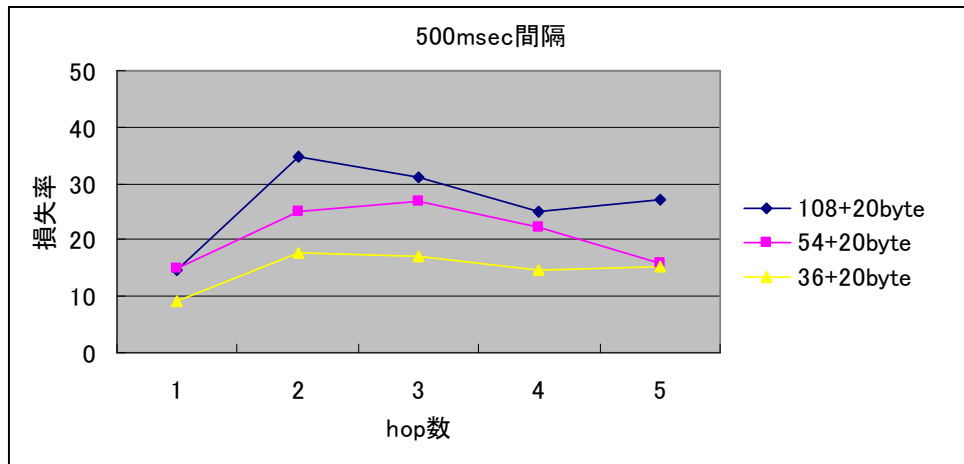


図 29 送信間隔が 500msec の場合の輻輳シミュレーション結果

(イ)送信間隔が 1000msec の場合

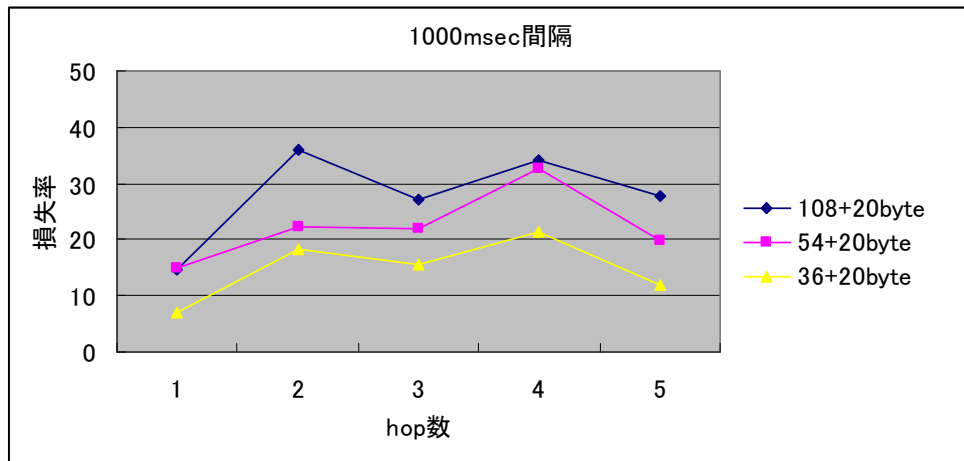


図 30 送信間隔が 1000msec の場合の輻輳シミュレーション結果

(ウ)送信間隔が 1500msec の場合

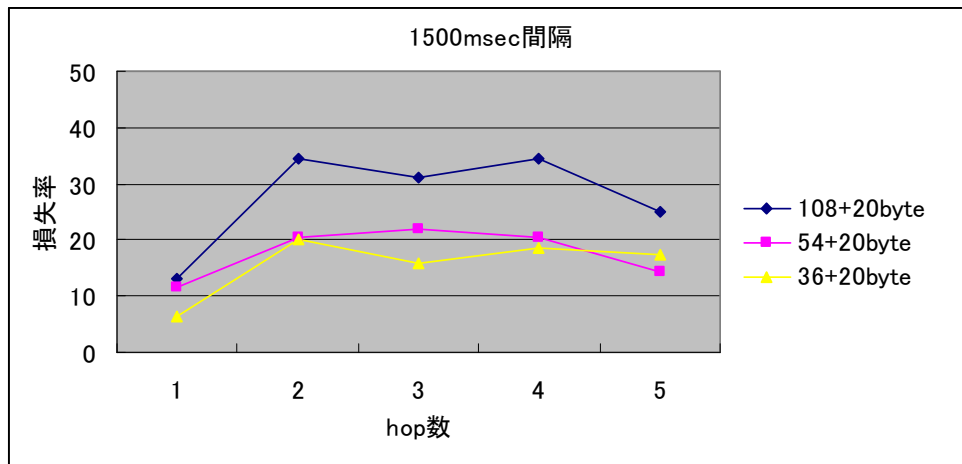
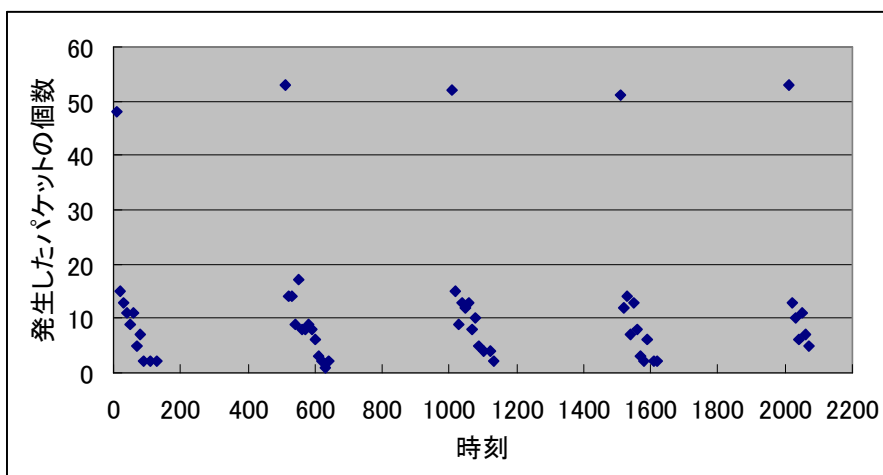


図 31 送信間隔が 1500msec の場合の輻輳シミュレーション結果

(エ)上のグラフから損失率はデータサイズに応じて低くなるものの、送信間隔にはさほど影響がないことがわかる。実際 CCA 送信のログ(TimeLine.csv)ファイルによると、1 回目の全ノードからのパケット送信はおよそ 100~140msec で終了し、その後 2 回目のパケット送信時刻の 500、1000、1500msec まで通信は行われていないことから、これは妥当な結果と言える。

	A	B	C	D	E	F	G	H	I	J
1 時刻	時刻	パケットID	送信	受信	送信	送信完了	リトライ	次回送信時刻		
122	93.04	14		6	2	○	95.04			
123	115.04	14		2	1	×		0	115.232	
124	115.232	14		2	1	○	117.232			
125	137.232	14		1	0	×		0	138.064	
126	138.064	14		1	0	○	140.064			
127	510	17		1	0	×		0	511.472	
128	510	18		2	1	×		0	511.152	
129	510	19		3	2	×		0	511.472	
130	510	20		4	0	×		0	512.112	

TimeLine.csv



時刻ごとのパケット送信回数(送信間隔 500msec)

図 32 輻輳シミュレーション結果の評価

4-3-3 達成状況

計画とおり、目標を達成した

- ①実証実験システム設計第1版作成完了
- ②実証実験場所の概定→ 京都市消防局訓練場
- ③性能シミュレーション評価実施

性能シミュレーション評価結果は、07年度に計画している ZigBee を実際に実装して行う実証実験における ZigBee ネットワーク構築に資する情報として、また実証実験結果評価のために比較論理値として活用していく予定である。

4-4 総括

課題アの無線ノードの開発においては、市場の状況変化により、当初想定したものと異なる要求やコストが発生してきており、一部、計画変更をお願いしたが、それ以外は、ほとんどすべての課題において、目標どおりの成果を達成できている。

なお、無線ノード開発では、開発に必要な費用が多く、本委託業務の予算だけでは不足している。今年度は、OKI の予算を追加して開発を進めている部分もある。19年度においては、開発項目を見直し、予算内での業務遂行ができるように調整していきたい。

5 参考資料・参考文献

5-1 研究発表・講演等一覧

課題イ ③スリープ制御 において、以下の発表を行った。

1. 柳原健太郎、福井潔、野崎正典、福永茂、“マルチホップセンサネットワークにおける消費電力均等化方式”、電子情報通信学会、IN研究会、(2007年3月)。

また、課題イ ⑤プロファイル において、ZigBee Alliance に対して、2件の提案と2件の共同執筆を行った。

1. Shigeru Fukunaga, “064642r00ZB_TA_PTG-Basic_Idea_for_Easy_Association”, (Dec. 2006).
2. Shigeru Fukunaga and Kentaro Yanagihara, “074752r00ZB_TA_PTG-Experimental_Result_of_Fast_Associationon_TA”, (Feb. 2007).
3. “064199r13ZB_TA_PTG-Telecom-Applications-Market-Requirement-Documents”, (Sep. 2006).
4. “064715r01ZB_TA_PTG-Technical_Requirements_Document”, (Feb. 2007).