

# 大規模無線センサネットワーク向け センシングシステムアーキテクチャの提案

鈴木貴久<sup>†1</sup> 栗原康志<sup>†1</sup>

ワイヤレスセンサネットワークにより環境モニタリングのような広域監視システムを実現しようとする無線センサノードを広範囲に大量に設置して大規模無線センサネットワークを構築する必要がある。本稿ではこのような大規模無線センサネットワークによる環境モニタリングで重要となる要素としてメンテナンス性に着目しこれを考慮したセンシングシステム全体のアーキテクチャについて検討し提案を行った。

## A Sensing System Architecture for Large-Scale Wireless Sensor Networks

TAKAHISA SUZUKI<sup>†1</sup> KOJI KURIHARA<sup>†1</sup>

To construct environment monitoring systems by using wireless sensor network technologies, it is required to deploy huge numbers of sensor nodes to wide areas and construct large-scale wireless sensor network systems. High maintainability becomes an important element for these systems to manage numerous sensor nodes remotely. This paper made a proposal of sensing system architecture which can achieve high maintainability for environment monitoring systems.

### 1. はじめに

圃場監視や水質監視、自然災害監視に代表される環境モニタリングの分野では広範囲を長期間にわたって監視することが求められる。たとえば、地滑りなどの土砂災害の監視では地滑りの恐れのある斜面全域にわたって地中水分量の変化や斜面の変位などを長期間監視し続ける必要がある。

従来は、光ファイバー式の歪み計[1]やマイクロ波[2]、カメラなど大型で高価なセンサを利用して広範囲を監視していたが、近年では単体のセンサで広範囲を測定するのでは無く、小型で安価なセンサノードを広範囲に多数設置し、センサノード間をマルチホップの無線網で接続すること広範囲の監視を実現するワイヤレスセンサネットワーク (WSN) 技術[3]が注目されている。

### 2. 関連研究

WSN 技術は環境モニタリングの分野以外にも、構造物監視や設備監視など様々な分野での利用も研究されている。

構造物監視の分野では地震動や風荷重などの外乱に対する構造物の応答を計測するセンサを有線から無線にすることで要求される性能を維持しつつ設置コストを下げる技術が研究されている[4]。

また、WSN を利用した環境モニタリングの分野では慶應義塾大学の Airy Notes プロジェクトと Mebius Sensor プロジェクトがある[5]。Airy Notes プロジェクトでは 150 個のバッテリー動作のセンサノードにより広範囲の測定を行っているが、ここで用いたセンサノードはマルチホップ通信の機

能が無く、別途ネットワークと電源に接続されたシンクノードを大量に設置する必要があった。一方で、Mebius Sensor プロジェクトではマルチホップの無線センサノードを利用しているが、多くて 16 ノード程度を想定していた。

また、WSN を用いて遠く離れた島の生態系調査を 9 ヶ月にわたって行った研究[6]や、ワイン畑にセンサノードを設置して圃場のモニタリングを行う研究[7]もあるが、これらも利用したセンサノード数は 10 ノードや 32 ノード程度である。

WSN を用いた環境モニタリングシステムを実現する場合、多い場合は数百に及ぶセンサノードで長期間にわたって広範囲を監視することが想定される。WSN で提唱されているマルチホップ方式の通信規格では、規格上は最大で数千～数万のノードに対応したものもあるが、環境モニタリングの分野を前提として実際に大量のセンサノードをマルチホップ方式で接続することを想定したワイヤレスセンサシステムのアーキテクチャについて研究しているものは少ない。

### 3. 分析

#### 3.1 環境モニタリングの分析

環境モニタリングを行うためには、温度・湿度・地中水分量といった環境情報をセンサで測定し、これを記録・分析し、必要な形式に変換して表示する必要がある。当然ながら、環境情報は斜面など監視対象となる場所にセンサを設置して取得する必要がある。一方で、センサから取得・分析した情報は例えば監視センターなどの施設で表示するといったことが考えられるが、安全性や利便性の観点から一般的に監視センターは実際に監視対象となる場所とは離

<sup>†1</sup> 株式会社富士通研究所  
Fujitsu Laboratories Ltd.

れた場所に設置することが求められる。

このため、環境モニタリングシステムではセンシングと表示は物理的に離れたところに設置され、その間をネットワークで接続するという構成になる。測定情報の記録と分析に関しては、センシング側で行うか、表示側で行うか、もしくはその中間で行う構成を取ることが考えられる。

### 3.2 WSN の分析

WSN では、センサノードが測定したデータは近傍のセンサノードに無線で送られ、受け取ったセンサノードはこれをまた近傍の別のセンサノードに無線で送るというマルチホップ通信を繰り返して最終的には終端となるノード（ゲートウェイ）に全てのデータが集約される。

ゲートウェイにデータを集約するためにはノード間でのマルチホップ通信におけるルート決定が非常に重要となるが、多数のセンサノードがある場合、ノードの故障や通信障害などにより一時的にもしくは長期的に通信不能なノードが出てくる可能性がある。このようなノードが出てきた場合は通信不能なノードを避けるようにルートを再構築する必要がある。

このことから、WSN システムでは、センサノードにおけるセンシングと無線通信に加えて、通信不能なセンサノードを検出してルートの再構築を行うネットワーク管理の機能が必要となる。ネットワーク管理機能については、それぞれのセンサノードで周辺ノードと通信しあってネットワークを管理する分散方式と、センサノードはセンシングと無線通信のみを行いネットワーク管理はゲートウェイなどの別ノードで行う集中方式が考えられる。

### 3.3 環境モニタリングへの適用

環境モニタリングでは、場合によっては数キロメートル四方に及ぶ広範囲のセンシングが求められる。このため、WSN 技術を環境モニタリングに適用すると、最大で数百にも及ぶ大量のセンサノードを設置して大規模な無線ネットワークを構築することを想定する必要がある。

ここで重要になるのがメンテナンス性である。環境モニタリングではセンサノードが設置される場所と監視を行う場所は離れた場所となると想定されるうえに、特に災害監視などでは人が立ち入ることが危険な場所にセンサノードを設置することもあるため、電池交換といった定期的なセンサノードの保守作業の頻度は極力少なくすることが望ましい。

また、定期的なメンテナンスに加えて不定期的な故障を修理するためのメンテナンスも考慮する必要がある。センサノード単体の故障率を  $\lambda$ 、設置するノード数を  $N$  とすると、センシングシステムの稼働率  $A$  は式(1)(2)より求められる[8]。

$$MTBF = \frac{1}{N\lambda} \quad (1)$$

$$A = \frac{MTBF}{MTBF + MTTR} \quad (2)$$

ここで、MTBF は平均故障間隔(Mean Time Between Failures)を示し、MTTR は平均修理時間(Mean Time To Repair)を示す。環境モニタリングでは、前述のようにセンサノードを設置する場所と監視を行う場所が離れているため、故障が発生した場合はセンサノードが設置された場所まで移動し、故障したセンサノードを探し出して修理もしくは交換するという作業が必要になるため、MTTR には単にセンサノードの修理・交換の時間だけでなく移動時間等も考慮に入れる必要がある。

例えば、故障確率  $\lambda$  を年に 1 回、MTTR を 1 日としてノード数  $N$  を変更したときの稼働率を算出すると図 1 のようになる。

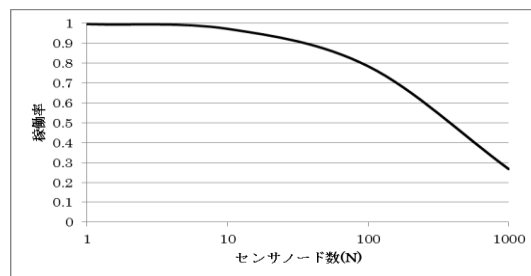


図 1 センサノード数による稼働率の変化

このように、ノード数が少ないときは稼働率が十分高くても、ノード数が増えると稼働率は大きく低下する。

システムの稼働率を向上させる方法としては、一つはセンサノードの品質を向上させて、単体での故障率を低下させる方法がある。もう一つの方法として、このように多数のノードからなるシステムでは、必要ノード数よりも多くのノードを設置しておき、一部のノードが故障しても冗長に設置したノードが故障したノードの機能を補完することでシステムの動作を維持させる冗長化の手法がある。

冗長に設置するノードの割合（冗長度）を  $\alpha$  とすると、ノードが故障する間隔は設置するノードが増えた分短くなるが、今回は  $N \times \alpha$  個のノードが故障するまでシステムは稼働を続けることが出来るため、冗長化を行った場合の MTBF は単純計算で式(3)のようになる。

$$MTBF = \frac{1}{(1+\alpha) \times N\lambda} \times N\alpha = \frac{\alpha}{(1+\alpha) \times \lambda} \quad (3)$$

冗長化を行わずにセンサノードの故障率を 10 分の 1 にした場合と、故障率を 1 のままで冗長度を 0.1(10%)としたときのノード数による稼働率の変化を比較すると図 2 のようになる。

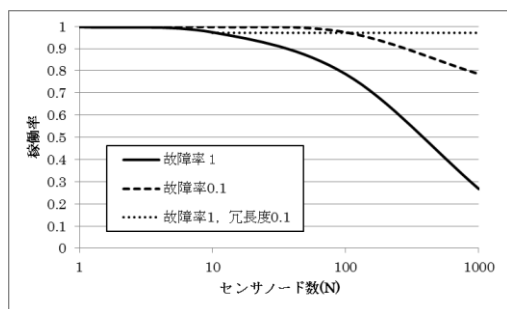


図 2 故障率低下と冗長化の比較

グラフを見ると、ノード数が少ない場合は故障率を下げた方が稼働率は高くなるが、ノード数が増えると稼働率は逆転し冗長化した方が稼働率は高くなる。このように、センサノード数が多い場合は故障率を低下させるよりも冗長化させた方が有効になる場合があると考えられる。

上記の計算では、通常の冗長化に対する考え方とは異なり、システムが停止する、すなわち故障ノード数が冗長に設置したノード数を上回るまで何もしないことを前提としているが、当然ながら稼働センサノード数が必要数を下回る前に故障したセンサノードを修理・交換することで理論上は無停止で長期間稼働し続けることも可能となる。

この場合、センサノードが故障するたびに修理を行うと修理の頻度が非常に高くなるため、通常は故障したセンサノード数が冗長に設置したセンサノード数に対して一定割合に達したら修理をするようにしたほうが合理的だと考えられる。この割合を  $\beta$  とすると、故障センサノード数の冗長に設置したノード数に対する割合が  $\beta$  を超えてからシステム停止までの猶予時間  $\tau$  が MTTR を下回ると修理中にシステムが停止する恐れがあるため、 $\beta$  は式(4)のように MTTR を下回らないように設定する必要がある。

$$\tau = (1 - \beta) \times MTBF > MTTR$$

$$\beta < \frac{MTBF - MTTR}{MTBF} \quad (4)$$

$\lambda$  が 1 で  $\alpha$  が 0.1 のときの MTBF はおよそ 33 日であり MTTR の 1 日より十分大きいので、この場合は余裕を持って  $\beta$  を設定出来る。また、MTBF が MTTR を下回る場合は冗長化させてもシステムの停止を避けることは出来ないと言える。

前述のように MTTR は監視を行う場所からセンサノード設置場所への移動時間といった非技術的な要素も大きく影響するため、MTBF をいかに大きくできるかが不規則的な故障によるメンテナンスの頻度や稼働率の観点で重要であると言える。冗長化を行った場合、MTBF は式(3)のようにセンサノードの故障率  $\lambda$  と冗長度  $\alpha$  に影響される。

## 4. 提案するアーキテクチャ

### 4.1 全体アーキテクチャ

前節での分析内容を踏まえて、本論文では図 3 に示すように「アプリケーション」「サーバ」「ゲートウェイ」「センサノード群」の 4 階層からなるアーキテクチャを提案する。

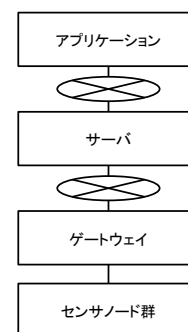


図 3 全体アーキテクチャ

各階層の役割は以下のように規定する。

- アプリケーション  
環境モニタリングにおける、センサから取得・分析したデータの表示を行う
- サーバ  
環境モニタリングにおける、センサから取得したデータの蓄積と分析を行う
- ゲートウェイ  
WSN におけるセンサノードからのデータの集約と、センサノード群の集中制御を行う
- センサノード群  
多数のセンサノードからなり、WSN におけるセンシングとマルチホップ通信を行う

また、階層間の接続は以下の接続を想定する。

- アプリケーション/サーバ間  
インターネット網のような遅延は大きくても長距離かつ大容量の通信が可能なネットワークで接続
- サーバ/ゲートウェイ間  
インターネット網のような遅延は大きくても長距離かつ大容量の通信が可能なネットワークで接続
- ゲートウェイ/センサ群間  
センサノード間通信と同じ低速の近距離無線通信方式で接続

### 4.2 センサノードの構成

環境モニタリングシステムではセンサノードを監視対象となる場所へ設置するため、センサノードの機能と構成が 3.3 で述べたメンテナンス性を向上させるために重要になる。

メンテナンス性を高めるためには、センサノードは極力単純な機能と構成であることが望ましい。システムの故障

率はこれを構成する要素の故障率の総和になるため、ソフト・ハード含めた構成要素数を少なくすることでセンサノードの故障率 $\lambda$ を下げる事が出来る。また、単純な構成にすることで低価格化や消費電力の削減も可能になるため、設置ノード数を増やして冗長化率 $\alpha$ を向上させることや、電池寿命が向上することで定期的なメンテナンスの頻度を減らすことも期待出来る。

そこで、本アーキテクチャではセンサノードは基本的にゲートウェイからの指示を受けてデータのリレーとセンシングのみを行う受動型の動作を行う。データのリレーに関しても、ルーティングはゲートウェイが行いセンサノードはそれに従って近傍ノードとデータの受け渡しをする。

このようにセンサノードの機能を単純化することで、センサノードのソフトウェアやそれを動かすMCU等を単純化出来る。また、動作を受動的にすることでセンサノードが管理する状態を減らすことが出来、これによりソフトウェアの故障に起因するリセットからの復帰や後述する低消費電力化のためのスリープやスリープからの復帰を容易にすることも可能となる。

#### 4.3 低消費電力制御

センサノードの動作を受動的にすることでゲートウェイでは短い期間に集中してデータ収集を行い、残りの期間は一切通信を行わないということを繰り返すいわゆる間欠動作を行うことで、通信を行わない期間はセンサノードをスリープ状態にすることが出来、センサノードの消費電力を削減することが可能となる。

間欠動作の間隔は対象となる環境モニタリングの内容により異なるが、間欠動作を行う場合センサノード間の時刻同期が問題になる。そこで、本アーキテクチャではセンサノードが受動的に動作することを利用してノード間で時刻同期を取らずに間欠動作とセンサノードのスリープを実現する手法をとる。

これを実現するために、ゲートウェイでは周期動作の期間をデータ収集などの通信を行う期間と通信を行わない期間に分けて管理し、通信を行う期間が終了すると、センサノードに対して遠いノードから順番にスリープの指示を送る。スリープの指示にはゲートウェイから見た現在時刻と次に通信を行う期間が開始する時刻との差分からマルチホップ通信による遅延やその間の時刻のズレやスリープ・起床にかかる時間などを考慮したマージンを引いた値をスリープ時間として指定する。センサノードではスリープ指示を受け取ると指定された時間スリープしてから起床して通信待機状態に戻る。

このように制御を行うことで、センサノードは本来通信を行う期間よりも停止指示待ちやマージンなどで長い時間稼働することになるが、その代わりに時刻同期を行わずとも非常に単純な仕組みで周期的な動作が可能となる。

ここで、通信を行う時間は少なくとも必要数センシング

データを収集する時間確保する必要がある。センシングデータに加えてルーティングのために周辺ノードの情報なども収集する場合は、そのための時間も考慮して通信を行う時間を決定する。 $i$  番目のセンサノードへ通信するためのホップ数を  $h_i$ 、平均のホップ数を  $h_{Ave}$ 、センサノード間で単位時間あたりに通信可能な回数を  $c$ 、収集するノード数を  $N$  とすると、通信を行う時間  $t_{Active}$  は、連続的に要求を行わずにセンシングデータが届くのを待ってから次の要求を行うような制御を行うと仮定すると、式(5)で表現出来る。

$$t_{Active} > \sum_{i=1}^N 2 \frac{h_i}{c} = 2 \frac{h_{Ave}}{c} \times N \quad (5)$$

また、間欠動作の間隔を  $t_{Interval}$  とすると、通信を行わない期間  $t_{Sleep}$  は式(6)で表現出来る。

$$t_{Sleep} = t_{Interval} - t_{Active} \quad (6)$$

スリープ指示は原理的にはセンサノードからの応答を待つ必要は無いため、冗長ノード含んだ全てのノード  $N_{Max}$  に送信するための時間  $t_{Ind}$  は式(7)で計算出来る。

$$t_{Ind} = \frac{N_{Max}}{c} \quad (7)$$

当然ながら、 $t_{Ind}$  は  $t_{Sleep}$  よりも小さくなる必要がある。

$j$  番目にスリープ指示を受け取ったノードがスリープする時間  $\Delta_j$  は式(8)で算出される。

$$\Delta_j = t_{Sleep} - \frac{j-1+h_j}{c} - M \quad (8)$$

ここで、 $M$  は時間のズレやスリープ・起床にかかる時間を考慮したマージンで、センサノードの設計情報などに基いてあらかじめ決められた値となる。また、 $\Delta_j$  がマイナスとなる場合は、そのノードはスリープ出来ないことになる。

前述のように、スリープ指示は遠いノードから順番に出していくが、通常は最大ホップ数  $h_{Max} < N_{Max}$  のため、ゲートウェイに最も近い場所にある最後 ( $N_{Max}$  番目) にスリープ指示を受け取るノードのスリープ時間が最も短くなる。このノードのホップ数は1のため、センサノードの最小スリープ時間  $\Delta_{Min}$  は式(9)のようになり、センサノードの起床中の平均消費電力を  $W_{Active}$ 、スリープ中の平均消費電力を  $W_{Sleep}$  とすると、低消費電力制御による電力削減の最小効率  $E_{Min}$  は式(10)で算出することが出来る。

$$\Delta_{Min} = t_{Sleep} - \frac{N_{Max}}{c} - M \quad (9)$$

$$E_{Min} = \frac{(W_{Active} - W_{Sleep}) \times \Delta_{Min}}{W_{Active} \times t_{Interval}} \quad (10)$$

ここで、間欠動作の間隔  $t_{Interval}$  や必要センサノード数  $N$  は

環境モニタリングの要件により固定的に決定されるが、一般的に低電力制御の効率に影響する各状態での消費電力やスリープ時間、起床時間等のセンサノードの設計に関わるパラメータに加えて、通信速度 ( $c$  に影響) や冗長度 ( $N_{Max}$  に影響) も低電力制御の効率に関係することがわかる。

特に冗長度については図 4 に示したように故障によるメンテナンスの観点からは冗長性を増やすことで MTBF や稼働率を上げた方が良いが、定期的なメンテナンスの観点からは冗長度を増やしすぎると低電力制御による消費電力の削減率が下がりバッテリーの持ちが悪化する。このため、コストの他にこれらも考慮してバランスの良い冗長度を決定する必要がある。

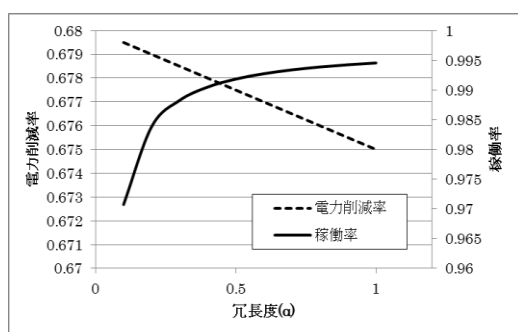


図 4 冗長度の影響

( $N=1000, \lambda=1$  年/回, MTTR=1 日,  
 $W_{Active}=100mW, W_{Sleep}=10mW, t_{Interval}=30$  分,  
 $t_{Sleep}=1400$  秒,  $c=100$  回/秒,  $M=30$  秒)

## 5. 実装例

### 5.1 全体構成

センサネットワーク技術による環境モニタリングシステムの実装例を示す。ここでは具体的なモニタリング対象は限定せずに、汎用的に環境モニタリングに適用可能なアーキテクチャを想定して実装を行った。

アプリケーションは一般的な PC やスマートフォン等で利用出来るように Web ブラウザ上で動作する Web アプリケーションとして作成しており、サーバとは一般的に Web アプリケーションで利用される TCP/IP ベースのプロトコルで接続される。

サーバとゲートウェイは Linux のソフトとして作成しており、これらも TCP/IP で接続されている。

センサノードは入手性から市販の無線モジュールを利用しており、センサノード間およびセンサノードとゲートウェイ間は無線モジュールが対応している ZigBee プロトコル[9]で通信を行う。

### 5.2 アプリケーション

アプリケーションではサーバを介して任意のゲートウェイが管理するセンサノード群のネットワーク構造とセン

サが取得したデータを図 5 のように GUI 上に表示する。

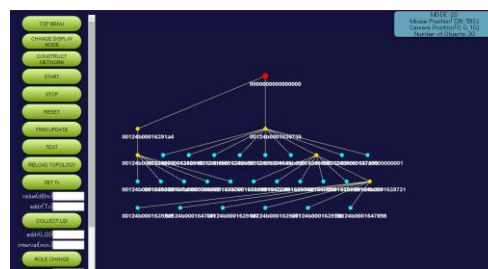


図 5 アプリケーション実装例

図中、頂点に位置するノードがゲートウェイ、複数の線が接続されている点がルータ、一つの線のみが接続されている点がエンドデバイス、それぞれの線がネットワークの接続関係 (親子関係) を示す。

また、画面左側のボタンを操作することで、サーバを経由してゲートウェイに対してセンシングの開始や停止を GUI から指示出来る。

### 5.3 サーバ

サーバではゲートウェイから送られてきたネットワーク情報と各センサノードの測定情報をデータベースに保存し、アプリケーションに表示される情報の更新を行う。

また、センシング開始などのアプリケーションからのコマンドをゲートウェイに通知する。

解析については環境モニタリングの対象により異なるためここでは実装を行っていないが、センシングデータの解析を行う場合はこの階層で実装する。

### 5.4 ゲートウェイ

今回の実装ではセンサノード間の通信は ZigBee プロトコルを用いるため、ゲートウェイに ZigBee におけるコーディネータの機能を持たせ、基本的なルーティングについては ZigBee が提供する機能を利用する。そのため、ゲートウェイの主な役割としては以下の 2 つとなる。

- 周期動作に基づく低電力制御
- ZigBee で対応出来ないケースでのルート変更

ここで後者について詳しく説明すると、ZigBee のセンサノードは通信の中継機能を持つルータと中継機能を持たないエンドデバイスの 2 種類がある。このうち、中継機能を持つルータノードが故障した場合は、これに接続されていたノードは近傍の他のルータに接続してネットワークを維持することが出来る。

しかしながら、近傍に図 6 (左) のように接続可能なルータが存在しない場合はそのノードはネットワークに接続出来ない。そこで、ゲートウェイではルータが故障したときに、図 6 (右) のようにその近傍にあるエンドデバイスをルータへ変更するよう指示を出してルータに変更することで ZigBee の機能のみでは接続出来なかったセンサノードを接続可能にするよう制御を行う。

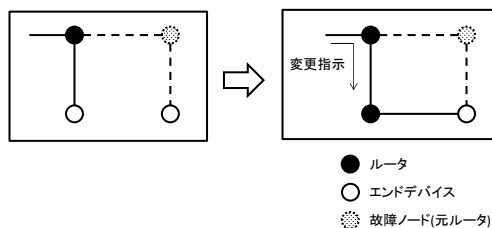


図 6 ゲートウェイ制御によるルート変更

## 5.5 センサノード

センサノードに搭載する無線モジュールのスペックを表 1 に、ソフトウェアの構成を図 7 に示す。

表 1 無線モジュールスペック

|     |                             |
|-----|-----------------------------|
| MCU | 8bit マイコン                   |
| RAM | 8KB                         |
| ROM | 256KB                       |
| 無線  | 2.4GHz 帯<br>IEEE802.15.4 準拠 |

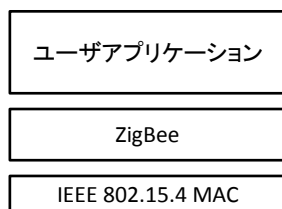


図 7 センサノードソフトウェア構成

センサノード間のマルチホップ通信などは ZigBee 以下のレイヤーで行われるため、ユーザーアプリケーションのレイヤーでは以下の機能のみを提供することになる。

- センシングを行いゲートウェイに測定データを送る
- 指定された時間スリープする
- エンドデバイスをルータに変更する

4.2 節で述べた通り、これらの機能はゲートウェイからの指示に基づいて受動的に動作する。このようにセンサノード上のアプリケーションを単純にすることで、表 1 に示した非常にシンプルな MCU と小容量のメモリでも動作させることが可能となる。

## 6. 評価

ここでは、シミュレータを用いた多センサノードでのシステム動作と、実際のセンサノードを用いたシステム動作と、ゲートウェイで低電力制御を行ったときのセンサノードにおける電力削減の効果の点で評価を行った。

### (1) シミュレータを用いた多ノード動作

まずは大量のセンサノードがあるときにセンシングシステムが想定通りに動作するかを、図 8 のように実ノード

に代わって別途開発したマルチエージェント型のシミュレータ[10]をゲートウェイに接続して評価を行った。

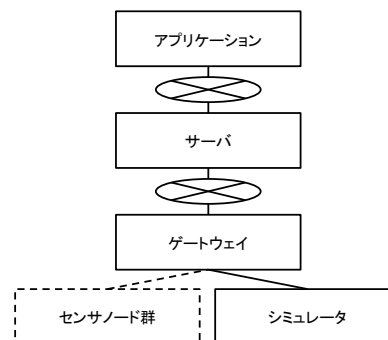


図 8 多ノード評価環境

シミュレータでは、図 9 に示すように 250 個のノードを平面上のフィールドにランダムに配置し、ノードには ZigBee の動作モデルを実装して多センサノードの動作シミュレーションを行った。

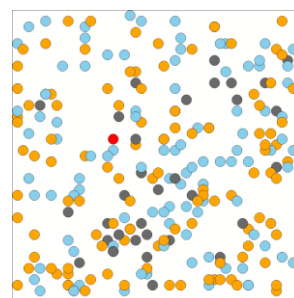


図 9 シミュレータ上のノード配置

シミュレータ上ではコーディネータ、ルータ、エンドデバイスといったノードの種類や状態が色で表現されている。

このシミュレータをゲートウェイと接続してアプリケーションでネットワーク構造を表示したものが図 10 になる。

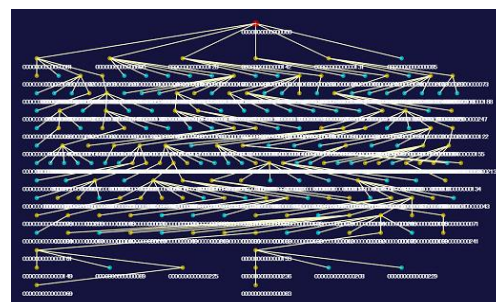


図 10 アプリケーションでの表示結果

シミュレータ上のノードが接続されている様子が GUI 上に表示され、これらのセンサノードに対してアプリケーションからデータ収集などの操作を行うことが可能である。



このように、数百規模のセンサノードがあっても提案したセンシングシステムにより管理出来ることが確認出来た。

## (2) 実機を用いた動作評価

次に、実際に複数の実センサノードを設置してシステムの動作確認を行った。ここでは、図 11 に示すように屋外に 13 台のセンサノードを設置したところ、図 12 のように全てのノードがネットワークに接続され、実機においてもセンシングが可能なが確認出来た。



図 11 屋外での測定

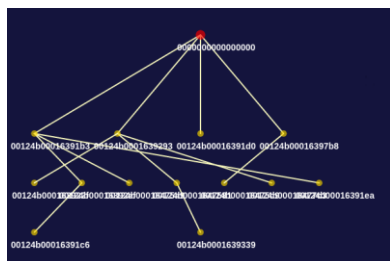


図 12 接続結果

## (3) 低電力制御

ZigBee のルータでは動作中は常に無線の受信回路がオンになっているため、ゲートウェイからの指示を受信待機しているだけでも受信回路により多くの電力を消費している。ノードをスリープさせることで受信回路を停止し、消費電力量を削減することが出来る。

実際にルータノードをスリープさせたときの電力波形を図 13 に示す。

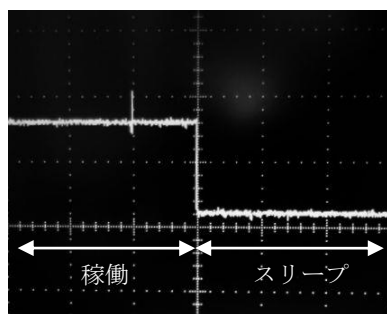


図 13 電力波形

稼働時の消費電力が 105mW に対して、スリープさせることで消費電力を 13mW にまで落ちることが実際に確認出来た。

## 7. まとめ

WSN 技術を利用した環境モニタリングシステムでは遠隔地に大量のセンサノードを設置するため、センサノードのメンテナンス性が重要になる。ここでは、センサノードを非常にシンプルにし、ルーティングや低電力制御などの複雑な処理はゲートウェイで行わせることでセンサノードの価格や消費電力を抑えながらも稼働率を向上させるアーキテクチャを提案し、実際に実装を行った。これをシミュレータでの大量ノード環境および実機での少数ノード環境と接続し、実際にネットワークを構築してアプリケーションから遠隔で制御出来ることが確認出来た。理論上は大量の実ノードでも制御することが可能だと言える。また、ノードの低消費電力制御もゲートウェイから遠隔で行えることが確認出来た。今後は今回の結果を実際のノード制御に反映させていくことが必要となる。

## 参考文献

- 1) 小島謙一, 伊藤裕昌, 村田修, 秋山洋, 山浦剛俊, 棚村史郎: 土構造物の計測管理に用いる光ファイバーの基本特性, 土木学会論文集, No.711/III-68, pp.187-197 (2004)
- 2) 三宅克行, 田中正弘, 玉木茂, 市川正二郎: マイクロ波を利用した岩盤斜面危険予測システムの検証, 第 30 回岩盤力学に関するシンポジウム講演論文集, pp.193-197 (2000).
- 3) C.Y. Chong and SP Kumar: Sensor Networks: Evolution, Opportunities, and Challenges, Proceedings of the IEEE, Vol. 91, pp. 1247-1256 (2003)
- 4) 長井望, 三田彰, 矢向高弘, 佐藤忠信: 構造ヘルスモニタリング用ワイヤレス検査に関する研究, 日本地震工学会論文集 第 3 巻, 第 4 号 (2003)
- 5) 伊藤昌毅: センサネットワーク技術を活用した環境モニタリングの実現, 電気学会 次世代産業システム研究会, IIS-10-070, pp. 23-28 (2010)
- 6) Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J.: Wireless sensor networks for habitat monitoring. In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 88-97 (2002)
- 7) Liscano, R., Jacoub, J. K., Dersingh, A., Zheng, J., Helmer, M., Elliott, C., & Najafizadeh, A.: Network performance of a wireless sensor network for temperature monitoring in vineyards. In Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 125-130 (2011)
- 8) 中原啓一, 加藤榮護: システムエンジニアリングハンドブック, IV 編, オーム社 (1991)
- 9) ZigBee SIG ジャパン  
<http://www.zbsig.org>
- 10) 栗原康志, 鈴木貴久, 山下浩一郎: 大規模センサネットワークのアーキテクチャ探索環境の実装と評価, 第 204 回 計算機アーキテクチャ研究発表会 (2014)