

# 情報ネットワーク管理運用要求に基づいた WSN の状態可視化手法の提案

浦田佑貴<sup>†1</sup> 吉廣卓哉<sup>†2</sup> 川橋裕<sup>†3</sup>

**概要：**無線センサネットワーク（Wireless Sensor Networks; WSN）は IoT（Internet of Things）を実現する技術の一つとして注目されている。WSN の管理運用においては、少ないオーバーヘッドで必要な情報を効率的に収集し、ネットワークの状態を少ない労力で把握できる可視化が必要である。近年の WSN 運用支援に関する研究には、機械学習や確率モデル等を用いて異常検知する研究が多く見られる。しかし、これらは予め想定できる定型的な異常の検知のみが可能であり、運用管理において頻繁に発生すると考えられる例外的な異常は発見できない。また、異常検知は可能であるが、問題解決のために詳細を効率的に調べる方法はサポートされない。本研究では、従来の情報ネットワーク管理運用の実績から導き出された管理運用における要求項目に基づいた WSN の状態可視化システムを提案する。提案システムにおいては、WSN の管理に必要なと考えられる構成管理、障害管理、性能管理の必要事項を適切に監視し、必要に応じて詳細な現象を調査できる WSN の状態の可視化を実現する。提案システムにより、WSN の管理運用における要求事項を満たすことが可能になり、効率的な WSN 監視が可能になる。

**キーワード：**無線センサネットワーク，ネットワーク管理

## 1. はじめに

無線センサネットワーク（Wireless Sensor Networks; WSN）は IoT（Internet of Things）を実現する技術として注目されており、その特徴からネットワーク接続された各種センサを農業・医療・環境モニタリング等に活用することが期待されている。実用的な WSN を実現するには、省電力で信頼性の高い MAC プロトコルの開発だけでなく、ネットワーク上で発生する問題を的確かつ迅速に検知・対処できる管理手段が必要である。

WSN は、その利用用途から長期的な利用が望まれており、そのための技術開発や研究が進められている。例えば、各センサ端末が可能な限りスリープ状態を維持することで消費電力を抑えるデューティサイクルに基づいた MAC プロトコルが開発されている。しかし、このような WSN を管理する場合には、SNMP のような能動的な問い合わせは、マルチホップ通信である WSN では応答までの時間が非常に長くなり、実用的ではない。そのため、管理に必要な情報を常にシンクノードに蓄積しておき、その情報を用いてセンサの状態を監視する方法が検討されている。

また、WSN には他にも運用管理を行う上で障害となる特性がある。WSN では様々な環境に適応するため無線通信が用いられるが、一般に通信帯域が限られる。このため、管理のためのオーバーヘッドを最小にする努力が必要となる。つまり、WSN の管理運用においては、できる限りオーバーヘッドを抑えた上で管理運用に必要な情報を効率的に収集し、これらを用いて、ネットワークの状態を少ない労力で把握できる可視化ツールが必要であるここで、オーバーヘ

ッドを抑えるためにパケットに付与する情報量を小さくすると、故障や通信障害等の際に把握できる内容が限られるため、原因の特定や対策が困難になり、管理業務においては不十分である。WSN の管理・運用において必要十分な情報量を把握し、できる限り通信オーバーヘッドを抑えることが求められる。

先述の通り、シンクに情報を常時集約し、それらの情報を用いて WSN の状態を把握する研究開発が行われている。Octopus[1]や SRNET[2]等が知られており、Parbat らのサーベイ論文にも多数のシステムが紹介されている[3]。しかし、これらの研究開発においては管理に必要なデータの議論は存在せず、WSN のデータ管理において必要十分なデータ量は不明確である。

本研究では、従来の情報ネットワーク管理運用の実績から導き出された管理運用における要求項目に基づいた WSN の管理運用を検討し、必要十分なデータ量に基づいた WSN の可視化システムを提案する。提案システムにおいては、WSN の管理に必要なと考えられる構成管理、障害管理、性能管理の必要事項を適切に監視し、必要に応じてネットワークの不具合の原因を特定するための WSN の状態可視化を実現する。なお、提案システムでは、WSN が取得したデータの精度や正当性の監視は対象外とし、あくまでもノードやネットワークの状態の把握を対象とする。

提案システムの有用性を示すために、シミュレーションデータを用いて提案システムの利用評価を行なった。シミュレーション中で WSN の不具合を発生させ、被験者が提案システムを用いることで不具合の原因の特定を試みる。これにより、本システムが上記 WSN 管理における 3 つの要

<sup>†1</sup> 和歌山大学大学院システム工学研究科  
930 Sakaedani, Wakayama, 640-8510, Japan  
<sup>†2</sup> 和歌山大学システム工学部  
930 Sakaedani, Wakayama, 640-8510, Japan

<sup>†3</sup> 和歌山大学システム情報学センター  
930 Sakaedani, Wakayama, 640-8510, Japan

求を満たすことを確認する。提案システムにより、必要最低限のオーバーヘッドで WSN の管理運用における要求事項を満たすことが可能になり、効率的な WSN 監視が可能になる。

本論文の構成は以下の通りである。第2章では、WSN の管理を支援する関連研究について説明する。第3章では、WSN の運用管理に必要なと考えられる管理要求について議論した上で、提案システムの設計方針を説明する。第4章では、WSN を管理するための提案システムについて詳細を述べ、第5章では、運用管理の要求を考慮した上で提案システムを評価する。最後に6章をまとめとする。

## 2. 関連研究

WSN の管理を支援する可視化システムや異常検知手法は複数提案されている。本章では、これらの関連研究について述べる。

WSN では、各ノードがウェイクとスリープを繰り返すため、ネットワーク内の同期通信ができないため、異常の根本原因やネットワーク内での各端末の振る舞いをその場で把握することが困難である。このため、ネットワークの状況を各ノードが常時観測し、その情報をシンクに集約し、その情報から WSN の挙動や状態を把握する試みがなされている。

Ramanathan らは、シンクノードに受動的に収集されたデータから診断ダイアグラムを構築し、問題の根本的な原因を推定する Sympathy[4]と呼ばれる診断システムを提案した。Liu らは、シンクノードの情報から確率的な因果ダイアグラムを作成し根本原因を推定する手法 PAD[5]を提案した。Liu らはまた、複数のノードが互いに協調することで問題の根本原因を推定する診断システム TinyD2[6]を提案した。Khan らは、バグや問題の特定に関するイベントの繋がりを見つけるために、WSN のイベントログを利用するデバッグシステム DustMiner[7]を提案した。しかしながら、紹介したこれらの手法はあらかじめ定義された原因や問題を診断するため、新たに生じた現象や定義されていない問題を扱うことができない。実際のインターネット管理では、管理者は頻繁に新しい問題に遭遇するため、柔軟な対応を可能にする情報提示が必要である。

一方、WSNs の通常時の状態や挙動を学習しておくことで、通常時と現在の間の「距離」に基づいて、異常時を判別する手法も提案されている。Li らが提案した VN2[8]では、WSN で取得できる 43 項目のメトリックを使用することで、イベント履歴ログから通常時のネットワーク状態を学習する。その後、NMF モデル (Non-negative Matrix Factorization model) を適用することで WSNs の例外的な状態を検出する。Miao らは、ログから 22 項目のメトリック間の相関関係を計算することで異常を検知する Agnostic Diagnosis (AD)[9]を提案した。これらのシステムは、管理者が異常を

検知するための労力と時間を軽減するのに有用である。

しかし、WSN の運用管理においては、単に異常検知ができれば良いのではなく、管理者は検知された異常の真偽や詳細の状況を確認したうえで対策を行う必要がある。このためには、たとえ異常検知手法を用いたとしても、WSN の状況を詳細に調べ、異常の原因を特定するための効率的な可視化ツールは不可欠である。

WSN の可視化ツールとして、Octopus[1]や SRNET[2]などが提案されており、サーベイ論文にも多数の可視化ツールが紹介されている[3]。しかし、これらはネットワーク管理の観点に基づいて設計されておらず、ネットワーク管理における有用性の検討が不十分である。また、ネットワーク管理の観点からの評価もなされていない。本論文は、インターネット管理の原理に基づいて設計された初めての可視化ツールであり、既存研究に対する優位性がある。

## 3. 要求定義と設計

### 3.1 ネットワーク管理に基づいた管理要求

提案システムでは、任意の通信プロトコルが動作する WSN の監視を目的とする。一般にセンサノードはウェイクとスリープを繰り返しながら、全てのノードが一定時間毎にセンシングデータを取得し、シンクに送信すると考えられる。このようなネットワークでは SNMP のような、ノードにリアルタイムに状態を問い合わせるプロトコルは利用できないため、各ノードが生成する各パケットに管理情報を付与し、これを集約して可視化することで、ネットワークの状態や挙動を把握することになる。

WSN の管理においては、ネットワークが常に正常に動作しているかを監視する必要がある。このためには、発生した問題を迅速に検知できるだけでなく、通信性能の低下や経路の変更等、その兆候を捉えることが重要である。シンクに集められたデータを可視化することで、これらを迅速かつ的確に把握できるようにすることが、WSN の管理において重要となる。

本研究では、このような WSN の管理を行うために、一般的なインターネットにおいて適用されている管理要求を適用する。インターネットにおいてはネットワークの管理・運用は成熟しており、管理要求に関する共通認識が存在する。一般的には、ネットワークの管理要求として、構成管理、セキュリティ管理、資源管理、性能管理、障害管理の 5 つの管理項目が重要とされる[10]。これらの管理項目の具体的な内容を以下に示す。本研究では、WSN において特に構成管理、性能管理、障害管理の 3 つが必要十分であると考え、これらを実現する WSN 管理システムを設計する。

**構成管理:** 構成管理とは、WSN を構成する物理的要素と論理的要素を個々に管理し、加えてノード同士の繋がりであるリンクの状態を管理することである。

このとき管理される情報がすべての管理項目の基礎となる

ため、常に最新の状態に保つことが重要である。WSN を構成する物理的要素とは、センサノードやシンクノードであり、論理的要素とは、それらの物理的要素が持っているノード ID や通信速度、データ生成間隔やスリープ時間といった設定情報である。

**障害管理：** 障害管理とは、WSN で起きるどのような事象を障害と見なすのかを定義し、定義された障害ごとに検出や対策、または予防の方法を検討して実施することである。加えて、各センサノードでのログ情報を管理することも障害管理の 1 つである。障害検知の方法としては各機器に状態の問い合わせをおこなう能動型と、各機器からの報告を待つ受動型が存在する。

**性能管理：** 性能管理とは、ネットワークの性能を一定のレベルに維持することを目的とした管理である。ネットワークの性能としては、パケットロス率、トラフィック量、輻輳回数、CPU やバッファの使用率などがある。これらの値に閾値を設けて監視をおこない、閾値を上回った際の対処についても決めておく。加えて、これらの値を定期的に測定し、結果を蓄積していくことでネットワークの動向を把握することも重要である。

**資源管理：** 資源管理とは、ネットワーク機器とは別に、ネットワークケーブルの配線や停電時の電源管理、冷却のための空調などの設備に関する管理をおこなう。

**セキュリティ管理：** セキュリティ管理とは、ウィルスや不正アクセスからネットワーク内の資産を保護し、それぞれの脅威に対する対策を実施することである。

### 3.2 WSN 管理を考慮した管理システムの設計

WSN においては、資源管理とセキュリティ管理は除き、構成管理、障害管理、性能管理の 3 点が重要になる。資源管理は、WSN の資源がセンサ端末であり管理すべき資源が存在しないことから不要と判断した。セキュリティ管理は WSN においても重要ではあるが、情報漏えい等の被害は暗号化等による対策が本質的であり、本研究の目的に照らすと本質的でないことから除外した。なお、端末への不正侵入・操作等の被害はノードの異常動作として観測されるため、性能管理等の項目で検出されるべきである。WSN の管理における、3 つの管理要求は次のように記述できる。

**構成管理：** ノードの状態や経路が頻繁に変化する WSN では直感的に素早くネットワーク構成を把握することが重要である。そのため、センサ端末の配置や通信経路に加えて、電力状況や内部ステータスなどの各端末の状態を、時間を遡って確認できることが重要である。

**障害管理：** 遅延が生じる WSN において発生直後に障害を発見することは困難であるが、シンクに集積されたデータからいかに迅速に障害を管理者に通知するかが重要である。想定する WSN では各ノードが定期的にデータをシンクに送ることを想定しているため、基本的には各ノードからのパケット送信記録により障害が把握できる。

**性能管理：** WSN では通信遅延やパケット損失率等により性能が測定されると考えられる。これらの値を観測しておき、異状が発生した場合にはできるだけ早期に管理者が発見できることが重要である。また、問題が発生したばあいには、容易に過去に遡ってこれらの性能を確認することも性能管理のうえでは重要な作業になる。

提案システムでは、管理者はまず警告表を見てネットワークの異常をチェックする。3 つの管理項目に基づいたネットワークの変化は常に警告表に上がるようにする。そのうえで、警告表に上がった事象を詳細に調べ上げるための画面を用意して、その画面で調査を行えるようにする。直感的に操作できるインタフェースを整えることで、容易に調査を行い、ネットワーク内の状態や挙動を把握できる。

## 4. 提案システムの仕様

### 4.1 システム構成

本システムは、シンクノードに集約された管理用データを入力として解析することで、ネットワークの状況を可視化する機能を持つ。本システムの構成を図 1 に示す。本システムはサーバ上で動作し、シンクノードから定期的にデータを受け取る。なお、1 つの WSN に対してシンクノードが複数存在することが考えられるが、その場合には、サーバはすべてのシンクから管理データを受け取り、可視化を行う。サーバは Web サーバ機能を持ち、WSN の状況を可視化した Web ページを表示する。管理者は Web ブラウザを利用してサーバにアクセスすることで本システムを利用する。

### 4.2 生成パケットに付加する管理用データ

本提案では、WSN の状況を把握するために、図 1 の下部に示すように、センサノードが定期的に送信する生成パケットに管理に必要なデータ項目を付与する。これらの管理項目は、パケット生成ノードの状態を表している。以下に、管理に必要なデータ項目と、そのために実際にパケットに付与する必要のあるデータ項目について説明する。

#### (1) 管理に必要な情報

WSN の状況を把握するために解析する必要のあるデータ項目として以下の 9 項目を挙げる。

- (a) パケットを生成した時刻
- (b) シンクに到達した時刻
- (c) パケット生成ノード ID
- (d) パケットの次ホップ ID
- (e) シンクまでの距離 (ホップ数)
- (f) 電力残量
- (g) パケット到達までの遅延時間
- (h) パケットロス率
- (i) 各センサの位置

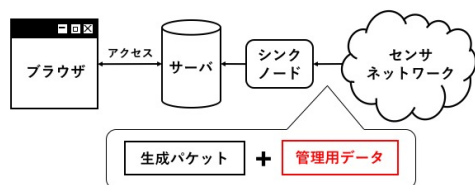


図 1 システム構成

(e)は、あるノードからシンクノードまでの経路上に存在するリンク数である。(g)はパケット生成からシンクノード到達までにかかった時間であり、(b) - (a)で計算される。(h)は、生成データがシンクへの到達に失敗した確率である。生成データの生成間隔時間を $\alpha$ 、確率を計算する期間を $\beta$ 、その期間内のシンクへのパケット到達回数を $\gamma$ とすると、

(h)は $\left\{\gamma \div \left(\frac{\beta}{\alpha}\right)\right\} \times 100$ で計算される。生成間隔時間は、WSN

実装時のパラメータを取得しておくか、不可能な場合にはパケットの受信履歴から推定することが可能である。(i)は、省電力化のためにノードがGPSを搭載していない場合が多い。その場合にはノードの設置時などに位置情報を保存しておくことで対応できる。

## (2) 生成パケットに付与するデータ項目

(a)から(i)のデータ項目において、(g)、(h)は上記の通り算出できる。また、WSNはできる限り省電力化とオーバーヘッドの削減が求められるため、(i)に関しても別途テキストファイルで保存しておくものとする。したがって、実際に生成パケットに付与する必要があるデータ項目は以下の6項目となる。

- (a)パケットを生成した時刻
- (b)シンクに到達した時刻
- (c)パケット生成ノードID
- (d)パケットの次ホップID
- (e)シンクまでの距離(ホップ数)
- (f)電力残量

また、これらを付与するにあたり危惧されるオーバーヘッドのサイズについて述べる。想定として、(a)、(b)は8byte、(f)は4byte、それ以外は2byteのデータサイズであるため、合計で26byteとなる。これは、省電力に特化した通信規格であるZigbeeの2.4Ghz帯(転送速度: 250kbps)においても非常に小さな値であることから、オーバーヘッドによる影響はほとんどないと考える。なお、このデータ項目は各センサノードがパケットを生成するタイミングでのみ付与されるため、ノードを経由するたびにデータ量が増加することはない。

## 4.3 画面遷移

提案システムの画面遷移図を図2に示す。図2の左側に示している通り、トップ画面には、警告表と未到達警告表の2つの表を表示する。警告表には、図2の右側に示す4つのインターフェースに関わる警告が表示される。表示されている警告をクリックすることで、4つのインターフェース

ースのうち該当するものに切り替わる。それぞれのインターフェースは、可視化する対象が、経路、電力、遅延、パケットロス率の4つに分かれており、それぞれの特徴をつかみやすいよう、可視化する対象毎に適切なグラフや表を組み合わせる。つまり、管理者は通常時にはトップ画面に表示された警告表を確認しておき、何らかの異変を確認した際には、各種4つのインターフェースを用いて詳細に調査するという使い方を想定している。

## 4.4 警告表

管理者がネットワーク状況の変化や異常にいち早く気付くために、注意すべき状況を自動判別して警告表に表示する。シンクノードに蓄積された、もしくは今現在受信しているパケット全てに対して解析を行い、特定のデータ項目の閾値を超えた場合にはその都度データベースに格納する。ある時刻での警告表の例を図3に示す。図3の(a)では、時刻を指定することによって、本来その時刻において表示されていた警告内容を表示することができ、過去の調査をスムーズに行える。標準では、現在時刻を指定したものとして現状の把握に用いる。警告表には、検出した警告の種類、その警告を検出した根拠となるパケットの到達時刻、該当するノードID、そして警告の内容を表示する。ここで、警告の対象とするデータ項目とそれにより把握できる事象を次に示す。

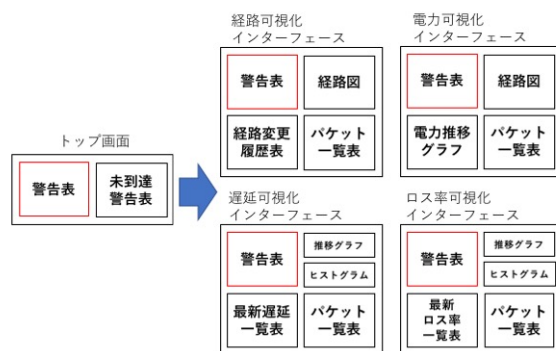


図 2 画面遷移

| 【警告表】 2022-01-05 12:00:00 を指定                                    |                     |     |                     |  |  |
|------------------------------------------------------------------|---------------------|-----|---------------------|--|--|
| 【値】 *経路変更:2回以上 *電力残量:320mAh(12.8%) *遅延:300秒 *未到達判定:24時間 *ロス率:20% |                     |     |                     |  |  |
| Search [a]                                                       |                     |     |                     |  |  |
| 指定した時点での警告を表示 2022-01-05 12:00:00 日付指定                           |                     |     |                     |  |  |
| 警告の種類                                                            | 警告時刻                | 回数  | 内容                  |  |  |
| 経路変更あり                                                           | 2021-03-28 23:59:59 | 340 | 経路を2回変更しました。        |  |  |
| 経路変更あり                                                           | 2021-03-28 23:59:59 | 294 | 経路を2回変更しました。        |  |  |
| 経路変更あり                                                           | 2021-03-28 23:59:59 | 217 | 経路を2回変更しました。        |  |  |
| 経路変更あり                                                           | 2021-03-25 23:59:59 | 149 | 経路を2回変更しました。        |  |  |
| 遅延                                                               | 2021-03-15 13:14:00 | 34  | 到達までに825.45秒かかりました。 |  |  |
| 経路変更あり                                                           | 2021-03-06 23:59:59 | 450 | 経路を2回変更しました。        |  |  |
| 遅延                                                               | 2021-02-24 23:02:00 | 88  | 到達までに659.37秒かかりました。 |  |  |
| 遅延                                                               | 2021-02-24 23:02:00 | 141 | 到達までに682.73秒かかりました。 |  |  |
| 遅延                                                               | 2021-02-24 23:02:00 | 191 | 到達までに719.37秒かかりました。 |  |  |
| 遅延                                                               | 2021-02-24 23:02:00 | 31  | 到達までに869.37秒かかりました。 |  |  |

図 3 警告表

## (1) 経路変更回数

WSNでは、状況に応じてトポロジが変更されることは頻繁

に起こりうるが、特定のノードや、ある付近に集中しているノードの次ホップの変更が多い場合等は異常と疑うことができる。そこで、次ホップが変更された回数に閾値を設けることで、普段の次ホップとは異なるイレギュラーな事象にも対応する。

## (2) 電力残量

WSN におけるノードは基本的にはバッテリー駆動が前提であり、定期的にバッテリー交換の必要がある。そのため、電力残量に閾値を設け、下回った場合には交換すべきノードとして通知を行う。

## (3) 電力消費

電力を消費しているということは何らかの動作を行なっている証拠となる。そのため、明らかに異なる電力の消耗が見られる場合に異常を疑うことが可能になる。周囲のネットワーク状況の変化による異常だけでなく、内部的なバグや、パケットには現れない無駄な動作による電力の消耗等を検出できる可能性がある。

## (4) 長期間のパケット未到達

生成パケットの送信間隔に比べて異常に長い間パケットが受信されない場合に、提案システムではエッジやノードの故障を疑う。この異常は、迅速に対応するべき事象である。警告表には表示するが、多数ある警告の中から探す手間や、途中で普及したりすると整理できなくなる可能性があるため、未到達一覧表に別途表示する。

## (5) 遅延時間

トポロジの変更途中や、ノードの応答がない場合等、迂回経路が発生する場合、普段よりも遅延時間は長くなる可能性がある。この事象を捉えることで、ネットワーク内の構成を把握する助けとなる。

## (6) パケットロス率

生成パケットをシンクまで送信する際に、通信が不安定なエリアや障害物等によりパケット送信がうまくいかない状況がある。例えば、数回に一度だけパケット送信に成功する場合では、上記(4)の長期間のパケット未到達では検出されないため発見に遅れてしまう。このような場合においてパケットロス率を警告対象とすることは有効な対策となる。

# 4.5 4つのインターフェース

図2に示したように、全てのインターフェースには警告表が含まれているが、詳細に調査したい場合は4つのデータ項目ごとに用意したインターフェースを使い分ける。次の項からそれぞれのインターフェースについて説明する。

## 4.5.1 経路インターフェース

経路インターフェースでは、経路変更を詳細に確認することができる。警告表だけでは一定期間内で変更した回数しか把握できないため、該当ノードだけでなく周囲のノードの経路状況や、経路変更していた頻度などを把握する際に用いる。次に、経路インターフェースに含まれる図表について説明する。

## (1) 経路図

経路変更が起きている時刻を指定した経路図の例を図4に示す。経路インターフェースにおける経路図では、指定した時刻から過去1日以内に経路変更があった場合に、図4のように該当ノードの色を赤く表示するとともに、経路の変更が描画される。赤い矢印は現在の次ホップを、灰色の矢印は直前の次ホップを示しており、経路がどのように変更されたかが一目でわかる。また、各ノードの中央に表示されている数値は、指定した時刻時点において各ノードが過去に経路変更した累計であり、どの付近で経路変更が頻繁にされているか、またはその逆も把握できる。なお、各ノードの中央の数値は、ノードIDと経路変更回数を切り替えることができる。

## (2) 経路変更履歴表

経路図では過去1日以内の経路を表示するが、過去の経路変更を詳細に見たい場合に経路変更履歴表を使用する。経路図でノードをクリックするか、ノードIDを指定することで、そのノードが過去に変更した経路を全て表示することができ、ネットワーク構成を把握する手助けとなる。

## (3) パケット一覧表

パケット一覧表では、指定期間内にシンクノードで受信した各ノードからのパケットを、全て確認することができる。例外的な問題が発生した場合や、各インターフェースで確認しきれなかった情報を把握する場合に、詳細な調査を行うことができる。

## 4.5.2 電力インターフェース

電力インターフェースでは、各ノードの電力情報を詳細に把握することができる。電力が残り少なくなり交換が必要であるノードの位置関係や、電力残量の推移を把握する際に用いる。次に、電力インターフェースに含まれる図表について説明する。なお、パケット一覧表は4.5.1で説明しているため省略する。

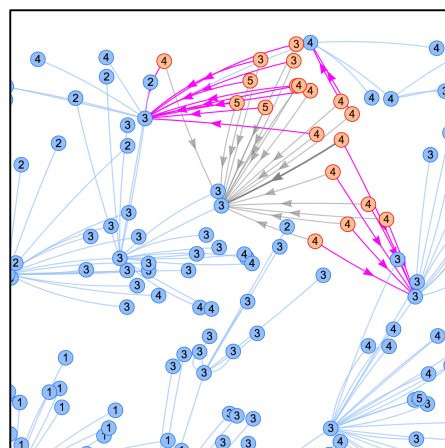


図4 経路図



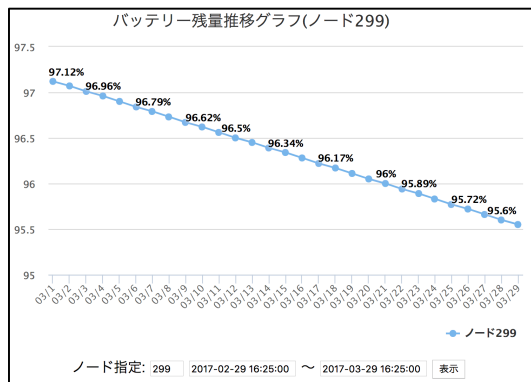


図 5 電力推移グラフ

### (1) 経路図

電力インターフェースにおいても、基本的な機能は経路インターフェースと同様であるが、指定時刻時点での電力残量をトポロジに反映し、閾値を下回る場合には該当ノードの色を赤く表示する。ノード交換に必要な位置情報を知ることができ、どの辺りで電力が消費しやすいのか推測することも可能である。また、各ノードの中央にその時刻での電力残量(%)を表示することができ、経路図上でノードの配置による偏りを把握できる。

### (2) 電力推移グラフ

例としてノード 299 の電力推移グラフを図 5 に示す。1 日に一度、その日の電力残量をデータベースに格納している。図 5 の下部にある入力フォームで指定した期間の電力残量を取得し、電力推移としてグラフ表示する。この推移を確認することで、急激な変化や内部状態の変更などを捉えることができる。

### 4.5.3 遅延インターフェース

遅延インターフェースでは、過去の遅延推移グラフやヒストグラムを確認することで、遅延時間を把握することができる。警告表によって遅延が大きいと検出された場合に、通常時の遅延時間を確認することで、どの程度の遅延であるかを推測できる。次に、遅延インターフェースに含まれる図表について説明する。

### (1) 遅延推移グラフ

ノード 4 のある時刻から 1 ヶ月間の遅延推移グラフを図 6 に示す。横軸は時刻、縦軸は遅延時間 (秒) を表している。このグラフからは、遅延時間が 0 と 30 を繰り返すような通信を行っていたが、ある時刻からは遅延時間が一定の値になったことがわかる。このように、過去の通信はどのような遅延であったかが一目でわかる。

### (2) ヒストグラム

図 6 と同じノード、期間における遅延ヒストグラムを図 7 に示す。このグラフには遅延ヒストグラムとホップ数のチャートグラフの 2 つのグラフを同時に描画している。はじめにヒストグラムでは、横軸を遅延時間、縦軸を該当遅延パケットの総数としている。これにより、遅延時間の分布を把握できる。次に、ホップ数のチャートグラフでは、横

軸は同じであるが縦軸がホップ数となっており、遅延時間の増加がホップ数によるものかを判断できる。このグラフでは図 7 の (a), (b), (c) に示すように、3 種類全てのホップ数が 2 となっており、今回の遅延時間のばらつきはホップ数の変化は関係していないことがわかる。

### (3) 最新遅延一覧表

この表では、指定時刻における最新パケット送信時の遅延時間を全ノード分表示する。つまり、この表を確認することで、指定した時刻においてネットワーク内でどの程度の遅延が発生していたかを把握できる。また、表のソート機能を用いて遅延時間を降順で並べ換えることで、最も遅延時間が長いノードを特定できる。

### 4.5.4 ロス率インターフェース

ロス率インターフェースでは、遅延インターフェースと同じく、ネットワーク内全体や該当ノードのロス率がどのように変化したかを把握できる。図表の構成は遅延インターフェースと同じであるため各図表の説明は省略する。

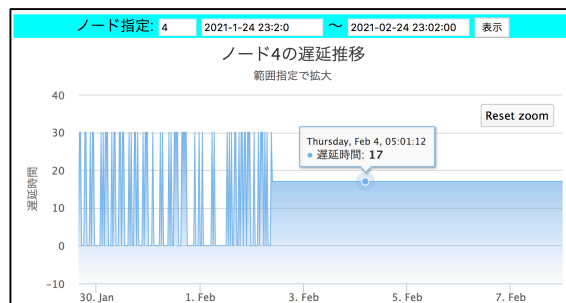


図 6 遅延推移グラフ

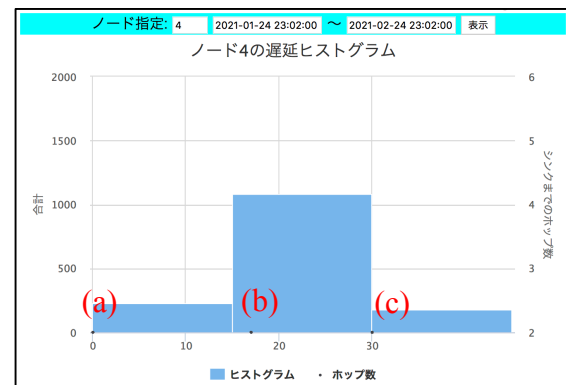


図 7 遅延ヒストグラム

## 5. 評価

### 5.1 実装環境

#### (1) 実装したサーバ環境

提案システムは、サーバ上で動作する Web アプリケーションとして実装した。Web サーバとして Apache ver.2.4.6[11] を用い、アプリケーション内では Javascript ライブラリである highcharts[12] と vis.js[13] をそれぞれ用いて折れ線グラフとネットワークを表示した。

#### (2) シミュレーションに実装した WSN の仕様

評価を行うにあたり、提案システムに必要な管理データ項目を収集する機能を持った WSN をシミュレーションにより実装した。WSN の MAC 及び経路制御プロトコルとしては、受信ノード主導型 MAC プロトコル RI-MAC を拡張し、消費電力をより低減するために我々の研究グループで提案しているプロトコル[14]を用いた。この手法を自作のシミュレータでシミュレーションを行った。2017 年 1 月 1 日から 8 年間における WSN の動作をシミュレートし、そのデータを用いた。また、500 個のノードを設置し、各ノードは 30 分周期でパケットを生成する。その他のシミュレーションパラメータは表 1 に示す。

### (3) 警告の判定に用いる閾値

評価実験を行うにあたり、4.4 で述べた警告表に表示する各データ項目の閾値を以下のように設定した。

- (a) 経路変更回数：1 回以上
- (b) 電力残量：320mAh 以下
- (c) 電力消費：5mAh 以上
- (d) 遅延時間：300 秒以上
- (e) 未到達判定時間：24 時間以上

なお、今回はシミュレータに実装しており、パケット送信は必ず成功するものとなっているため、パケットロス率については評価しないものとする。これらの閾値は、シミュレーションの結果から妥当な値を決定したものである。

## 5.2 評価実験シナリオ

評価のため、シミュレーション期間内にいくつか異常を発生させた。発生させた異常を以下に挙げる。

1. ノード 496 からノード 500 までの 5 つのノードを、スリープ状態にならないよう設定した。
2. 開始 100 日目から 100 日おきにランダムに 1 つノードを故障させた。

表 1 シミュレーションパラメータ

|           |                  |
|-----------|------------------|
| 実験開始時刻    | 2017年1月1日 00時00分 |
| 実験期間      | 8年               |
| 配置領域      | 500m四方の正方形       |
| ノード数      | 500個             |
| 通信可能範囲    | 半径50mの円内         |
| データ生成間隔   | 30分              |
| ビーコン送信間隔  | 30秒              |
| ノードの総電力容量 | 2500mAh          |

## 5.3 評価方法

評価実験は、以下の方法で実施した。まず、上記方法に従ってシミュレーションデータを提案システムに入力する。その後、本システムを用いて WSN で起きた事象を確認する。その際に、シナリオによって引き起こされた変化を捉えられるかを実験し、捉えた変化が WSN の管理に必要なと考える 3 つの管理要求を満たしているかを評価する。

## 5.4 評価結果

### 5.4.1 構成管理要求の評価

実装した MAC プロトコルでは、できるだけ葉ノードの

数が多くなるような次ホップを選択して経路が変更される。したがって、5.2 で述べたシナリオによるノード故障や電力不足により、このような経路変化を確認できれば構成管理における要求を満たしたものと評価する。

警告表を確認すると、警告として多数の経路変更があったことを確認できた。この時の経路図を図 8 に示す。この図から、直前までノード 436 を次ホップとしていた全てのノードは、周囲の中継ノードへと次ホップを切り替えていることが一目でわかる。これは、MAC プロトコルの仕様とも一致する。他の経路変更も同様に確認できた。また、故障したノードが葉ノードの場合には、経路の変更がされておらずトポロジの変化が見られないことを把握できた。

上記から、シナリオによって引き起こした経路の変化を捉えることができ、提案システムを用いることで、構成管理の要求を満たすことができる。

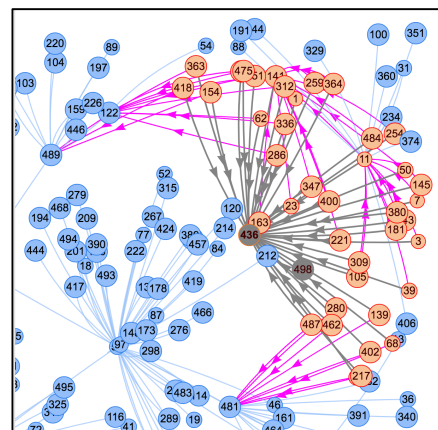


図 8 経路図 構成管理評価

### 5.4.2 障害管理要求の評価

障害管理における管理要求とは、ノードの故障または不具合を見つけ出し、その原因を特定することである。そのため、5.2 のシナリオによる障害を特定し、その原因を推測できたかを評価とする。

トップ画面に表示される未到達警告表には、計 51 個のノードからのパケット送信が途絶えていることがわかった。調査の結果、これらのノードは 3 種類に分類できた。1 つ目は、何らかの不具合によって起きた激しい電力消費によるバッテリー切れである。これは、電力推移グラフの傾きが、ある時刻を境に大きくなったことから判断できた。パケットロスは起きない仕様であり、経路の変更もなかったため、内部的な異常だと判断できる。2 つ目は、ノードの急な故障である。これは、電力残量も十分にあり、かつ経路変化や遅延の増大等も見られないままパケットが途切れたために判断できた。3 つ目は、電力を消耗しやすい中継ノードの単純なバッテリー切れであり、6 年を過ぎたあたりから徐々にバッテリー切れを起こしていた。

上記から、シナリオによって発生した障害が起きたノードの特定と、その原因を推定することができ、提案システムを用いることで障害管理の要求を満たすことができる。

### 5.4.3 性能管理要求の評価

性能管理における管理要求とは、ネットワーク性能を表すデータ項目が正常値とは異なる値を示した際に、いち早く気付くことである。その次にその原因を特定できることが望ましい。

評価対象として遅延時間に着目することで、ネットワーク性能をいち早く捉えることができると考える。ある時刻において、複数のノードから一斉に遅延時間の警告が検出された。そのうち1つのノードの遅延推移グラフを図9に示す。この図から、ある時刻の遅延時間が通常時よりも大幅に増加していることがわかる。この時点では他に異常は見当たらなかったが、この後約20分後には、数ホップ先の経路で大規模な経路変更が確認できた。その経路図を図10に示す。

上記より、ネットワーク性能を表すデータ項目に着目することで、事前に異常や変化を捉えることが可能であり、性能管理における要求を満たすことができる。

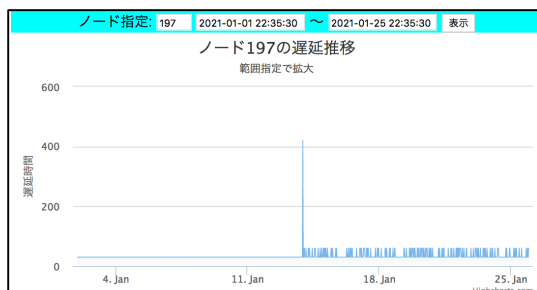


図9 遅延推移グラフ 性能管理評価

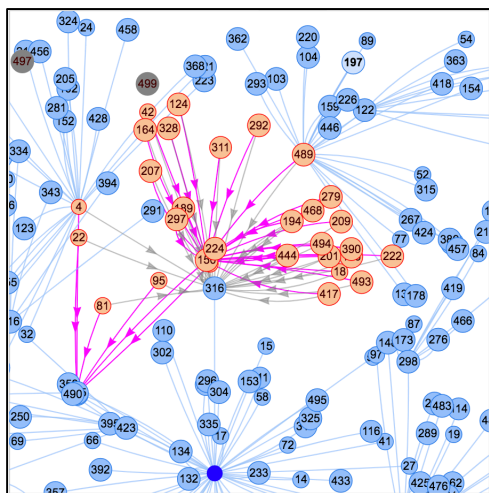


図10 経路図 性能管理評価

## 6. まとめ

本研究では、ネットワーク管理の観点から、WSNを適切に管理するための管理要求に基づいた可視化システムを提案した。管理要求は、障害管理、構成管理、性能管理の3つの管理要素を考慮して設計された。また、シミュレーション結果を入力として提案システムを実装した。意図的に設けたシナリオから、これらの3要求の全てを適切に満た

すことを確認した。これにより、提案システムの有用性が示された。

**謝辞** 本研究はJSPS 科研費 15H02691 の助成を受けたものである。ここに示して謝意を表す。

### 参考文献

- [1] R.Jurdak, A.G. Ruzzelli, A. Barbirato, and S. Boivineau, "Octopus: Monitoring, Visualization and Control of Sensor Networks" Wireless Communications and Mobile Computing, Vol. 11, Issue 8, pp. 1073–1091, (2011).
- [2] E. Karapistoli, P. Sarigiannidis, and A.A. Economides, "SRNET: A Real-time Cross-based Anomaly Detection and Visualization System for Wireless Sensor Networks," In Proc VisSec'13, (2013).
- [3] B.Parbat, A.K.Dwivedi, and O.P.Vyas, "Data Visualization Tools for WSNs: A Glimpse," International Journal of Computer Applications, Vol.2, No.1, pp.14–20, (2010).
- [4] N.Ramanathan, K.Chang, R.Kapur, L.Girod, E.Kohler, and D.Estrin "Sympathy for the Sensor Network Debugger," In Proc. Sensys'05, (2005).
- [5] K. Liu, M. Li, and Y. Liu, "Passive Diagnosis for Wireless Sensor Networks," IEEE/ACM Transactions on Networking, Vol.18, Issue 4, pp.1132–1144, (2010).
- [6] K. Liu, Q. Ma, Xibin Zhao, and Yunhao Liu, "Self-Diagnosis for Large Scale Wireless Sensor Networks," In Proc. INFOCOM'11, (2011).
- [7] M. Khan, H. Le, H. Ahmadi, T. Abdelzaher, and J. Han, "Dustminer: Troubleshooting Interactive Complexity Bugs in Sensor Networks," In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, pp.99–112, (2008).
- [8] X.Li, Q.Ma, Z.Zhicao, K.Liu, and Y.Liu, "Enhancing Visibility of Network Performance in Large-scale Sensor Networks," In Proc. of 34th International Conference on Distributed Computing Systems (ICDCS'14), (2014).
- [9] X. Miao, K. Liu, Y. He, D. Papadias, Q. Ma, and Y. Liu, "Agnostic Diagnosis: Discovering Silent Failures in Wireless Sensor Networks," IEEE Transactions on Wireless Communications, (2013).
- [10] 日本オープンソースソフトウェア (OSS) 推進フォーラム "ネットワーク運用管理に求められる作業" <http://ossforum.jp/node/594>
- [11] The Apache <http://www.apache.org/Software> (referred Foundation, December 10, 2015).
- [12] Highcharts, <http://www.highcharts.com/> (referred January 15, 2016).
- [13] vis.js, <http://visjs.org/> (referred January 15, 2016).
- [14] 横谷晟人, 吉廣卓哉, 受信ノード主導型 MAC プロトコルのビーコン削減に基づいた長寿命センサネットワーク, 情報処理学会論文誌, 58(2), 2018 年 2 月.