

THE IEICE TRANSACTIONS ON COMMUNICATIONS (JAPANESE EDITION)

# **IEICE** 電子情報通信学会 **B** 論文誌

通 信

VOL. J103-B NO. 11

NOVEMBER 2020

本PDFの扱いは、電子情報通信学会著作権規定に従うこと。

なお、本PDFは研究教育目的（非営利）に限り、著者が第三者に直接配布することができる。著者以外からの配布は禁じられている。

## 通信ソサイエティ

一般社団法人 **電子情報通信学会**

THE COMMUNICATIONS SOCIETY

THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS

# 無線センサネットワークの同期及び非同期を組み合わせたデューティサイクルプロトコル

川端 俊禄<sup>†</sup>      松崎 頼人<sup>††</sup>      徳永 潤平<sup>†</sup>      榎原 博之<sup>†</sup>

## Synchronous and Asynchronous Combination Duty-Cycling Protocol in Wireless Sensor Network

Shunroku KAWABATA<sup>†</sup>, Raito MATSUZAKI<sup>††</sup>, Junpei TOKUNAGA<sup>†</sup>,  
and Hiroyuki EBARA<sup>†</sup>

あらまし 近年、ワイヤレスセンサネットワークは IoT (Internet of Things) の中核を成す技術として注目される。しかしながら、各センサノードはバッテリーの電力に依存しているため、センサノードの寿命が限られている。そのため、ワイヤレスセンサネットワークでは省電力化のための研究が盛んに行われており、電力消費を削減する方法の一つとしてデューティサイクルが知られている。ネットワークの寿命を延ばすためには、デューティサイクルスケジュールを適切に決定し、効率的なデューティサイクルを実行する必要がある。そこで本研究では、同期型プロトコルと非同期型プロトコルを組み合わせて使用することにより、スケーラビリティを確保しながら効率的なデューティサイクルを実行するハイブリッドプロトコルを提案する。

キーワード 無線センサネットワーク, デューティサイクル, 時刻同期, IoT

### 1. ま え が き

近年、身近のありとあらゆるモノをインターネットに接続する IoT (Internet of Things) が注目されている。IoT では家電製品や医療機器など、パソコンやサーバといったコンピュータ以外の多種多様なモノがインターネットに接続し相互に情報をやりとりすることで、遠隔から認知、計測、制御を行い様々なサービスを提供する。IoT の中核を成す技術として無線センサネットワーク (WSN: Wireless Sensor Network) がある。WSN は、温度などのセンシングデータを取得し、取得したデータをパケットリレー方式で伝送するマルチホップアドホックネットワークを構成する。ノードを配置するだけで自律的なネットワークを構成できるため、防犯や防災、医療、気象観測、農業などの様々な分野での利用が期待されている。

環境モニタリングなど屋外設置の WSN では、設置位置の問題からノードに電源を確保することが困難な場合があるため、各センサノードはバッテリー駆動していることが多い。更に、利用されるノードの数が多くバッテリーの交換が困難である。センサノードのバッテリーが切れると、そのセンサノードとは通信ができなくなる。このようなセンサノードが多くなるとデータの収集に遅れが発生し、最悪の場合シンクノードまでの経路が確保できずデータの収集が行えなくなることがある。センサネットワークを利用するシステムにおいてデータの欠落は大きな問題となるため、省電力化によりネットワーク寿命を伸ばすことは重要な課題である。このような背景から、センサネットワークの研究では省電力化に関する研究が多く行われている。

省電力化の技術の一つとしてデューティサイクルが注目されている。デューティサイクルでは周期的にアクティブ状態とスリープ状態を繰り返すことで省電力化を行っており、1 周期のうちアクティブ状態である時間が占める割合をデューティ比という。このデューティ比が小さいほど消費電力は小さくなるが、アクティブ状態のときのみ情報を受信できるという制約が

<sup>†</sup> 関西大学, 吹田市

Kansai University, 3-3-25 Yamate-cho, Suita-shi, 564-8680 Japan

<sup>††</sup> 東京工業高等専門学校, 八王子市

National Institute of Technology, Tokyo College, 1220-2 Shibata-cho, Hachioji-shi, 193-0942 Japan

DOI:10.14923/transcomj.2020JBP3005

あるためスループットは低下してしまう。このように省電力化と通信性能はトレードオフの関係にある。また、隣接するセンサノードと時刻同期されていない場合、送信先のアクティブ状態に合わせてプリアンプを送ることができないため、デューティ比を小さくするとそのノードに対して送信を行うノードの再送回数が増加し消費電力が大きくなる。

本研究では、デューティサイクルを用いたセンサネットワークにおいて、非同期プロトコルと同期プロトコルを組み合わせたハイブリッドプロトコルを提案する。

提案手法はセンサノードの隣接関係や親子関係を掌握するためにネットワークを階層化する。同期プロトコルと非同期プロトコルの切り替えはこの階層値を利用する。同期プロトコルと非同期プロトコルの境界はノード配置後の初期設定や予備実験により決定し、ネットワークの形状に沿った最適な階層値を設定する。提案手法を評価するため、比較手法のADCCと併せてネットワークシミュレータNS-3[1]を用いた性能評価を行う。

本論文は、以下の章で構成されている。2.では、本研究の背景技術としてデューティサイクルや時刻同期手法について述べる。3.では、関連研究について述べる。4.では、提案手法について述べる。5.では、提案した手法についてネットワークシミュレータを用いた性能評価を行う。最後に6.では、本論文のまとめについて述べる。

## 2. 背景技術

### 2.1 デューティサイクル

デューティサイクルとは、あるシステムを周期的にアクティブ状態とスリープ状態を切り替える動作またはその技術のことであり、センサネットワークの消費電力を削減する技術の一つとして注目されている。センサネットワークにおいて電力を浪費する要因の一つとして、パケットを送受信していないアイドル期間に受信可能な状態を継続してしまうアイドルリスニング問題がある。デューティサイクルは、このアイドル期間のセンサノードをスリープ状態にすることで省電力化を図る。図1に横軸を時間とした状態変化のグラフを示す。縦軸の0はスリープ状態、1はアクティブ状態を表している。

ここで、 $T$ は1周期の長さ、 $\tau$ はアクティブ状態の時間を表し、 $\tau < T$ とする。1周期のうちでアクティブ

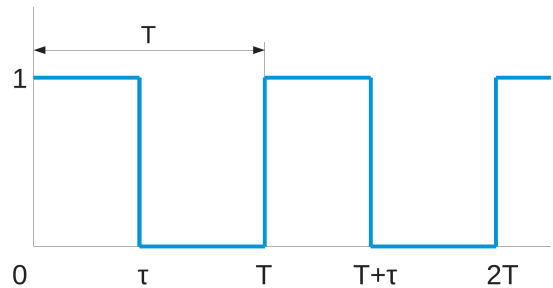


図1 デューティサイクルの状態変化  
Fig. 1 Duty cycling.

状態の時間が占める割合をデューティ比といい、 $d$ を用いて以下の式で定義される。

$$d = \frac{\tau}{T} \quad (1)$$

デューティサイクルではこのデューティ比が小さくなればなるほど、アクティブ状態の時間が短くなるため消費電力が少なくなる。しかし、スリープ状態ではデータの受信ができない制約があるため、データを受信できる時間が減りスループットが低下してしまう。このように、消費電力とスループットはトレードオフの関係にあり、通信性能と省電力化を両立するためには適切なデューティ比に設定する必要がある。

### 2.2 時刻同期

各センサノードはクロック周期のずれなどにより、内部時刻が実際の時刻からずれることが多々ある。そのため、同期プロトコルを使用することで、センサノードのクロックを実際の時刻に合わせる時刻同期を行う。

センサネットワークでは従来のネットワークと違い、電源の制約や配置されるノードの密度が高くなる傾向があるため、従来の通信速度や信頼性を重視した同期プロトコルでは消費電力やトラヒックの増大などの問題により最適な手段とはいえない。そのため、MAC層以上のプロトコルをアプリケーションに特化して開発できることや伝搬遅延が小さいといった無線センサネットワークの特徴を活用することで、単純な仕組みで高精度な時刻同期を実現するRBS (Reference Broadcast Synchronization)[2]、TPSN (Timing-sync Protocol for Sensor Networks)[3]、FTSP (Flooding Time Synchronization Protocol)[4]など、無線センサネットワークのための同期プロトコルが提案されている。

RBS は、ある任意のブロードキャストパケットを受け取った二つのノードで受け取った時刻を記録し、ブロードキャストパケットを受け取ったノード同士で時刻情報を交換することで、送信時のパケット組み立て時間や MAC 層への送信要求にかかる時間などの不確定要素を無視している。このとき、RBS はブロードキャストパケットを受け取った異なるノードの時刻のずれが正規分布に従うという性質を利用して、受信器間での時刻同期を実現する。

TPSN は、同期する際に基準となるノードをルートとして全域木を構築する。その構築した全域木の親ノードと子ノードで 2 ウェイ・ハンドシェイクによるタイムスタンプを利用して同期を行う。このとき、MAC 層でタイムスタンプを取ることで、時刻同期の不確定要素を削減している。

FTSP では誤差の要因を RBS や TPSN と比較してより詳細に分析することによって  $1\mu s$  程度の同期精度を実現している。FTSP は送信側と受信側の MAC 層でタイムスタンプを取ることで一方向の同期パケットの送信だけで精密な同期を実現できる。更に、一方向で同期可能という性質を利用し、同期パケットのフラグディングによるネットワーク全体の同期も実現できる。送信ノードのタイムスタンプを  $T1$ 、受信ノードのタイムスタンプを  $T2$ 、無線の通信速度を  $speed$  とすると、時刻の補正值  $\theta$  は式 (2) によって求められる。式 (2) に従って  $\theta$  を複数回取得し、その値を平滑化することで時刻同期の揺らぎを削減している。

$$\theta = T2 - T1 - \frac{1}{speed} \quad (2)$$

本研究では同期プロトコルとして FTSP を使用している。

### 2.3 コンバージキャスト

WSN の各ノードは演算能力をもたない場合が多く、センサで計測・認識したデータを演算能力をもつシンクノードへ集約することが多い。そのため図 2 に示されるように、ネットワークのトラヒックはシンクノードへの単方向通信になる。このような、シンクノードへの通信のことをコンバージキャストと呼ぶ。

コンバージキャストはシンクへデータを集約する通信であるため、シンク周辺では通信が集中しトラヒックが増加する特徴がある。この特徴からシンク周辺のノードは消費電力が増加し、遠方のノードに比べて早く動作を停止してしう。コンバージキャストにおいて動作を停止したノードが増加すると、データの収集に

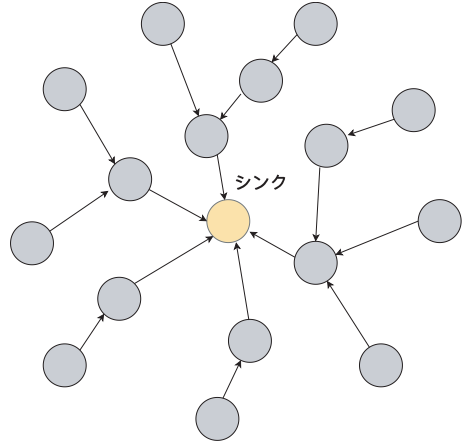


図 2 コンバージキャスト  
Fig. 2 Convergecast.

遅れが発生し、最悪の場合シンクノードまでの経路が確保できずデータの収集が行えなくなることがある。そのため、シンク周辺のノードの消費電力を減少させ稼働時間を延長することはネットワーク寿命を延ばす上で重要である。

### 3. 関連研究

WSN の寿命を伸ばす研究として、ハイブリッド型のルーチングを利用するもの [5] がある。この手法では、シンクから遠距離にあるノードはクラスタリング型の通信を行ってセンシングデータをシンク周辺に伝送し、シンク周辺のノードは非クラスタリング型の通信によってそのデータをシンクへ転送する。

クラスタリング型のプロトコルでは、特定の範囲内にあるノードがグループとなりクラスタを構成、グループ内の一つのノードがクラスタヘッド (以下 CH) として選択される。クラスタ内のノードはセンシングデータを CH に転送し、CH は送られてきたデータを集約してデータ量を削減した後、CH 間のマルチホップ通信によってシンクまで転送する。非クラスタリング型のプロトコルでは、全ノードがシンクに向けたマルチホップ通信によってデータを転送する。この手法ではこれらのプロトコルを組み合わせ、シンクから遠距離のノードはクラスタリングを利用して効率的にシンク周辺までデータを転送し、シンク周辺ではマルチホップ型の通信を行うことで、再クラスタリングによるオーバーヘッドをなくし各ノードに与える負荷の偏りを低減している。

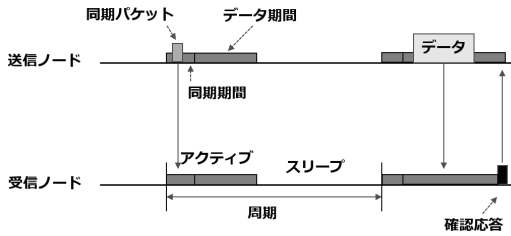


図3 S-MACの動作  
Fig.3 Operation of S-MAC.

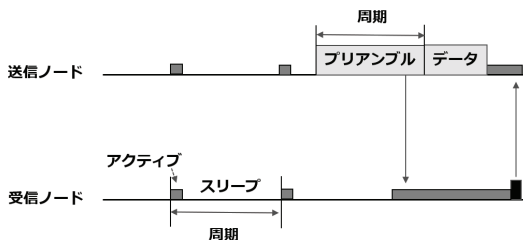


図4 B-MACの動作  
Fig.4 Operation of B-MAC.

また、センサネットワークの省電力化技術としてさまざまなMAC制御方式が提案されている[6]。代表的なMAC制御方式として、同期型にはS-MAC[7]、非同期型にはB-MAC[8]があげられる。図3、図4にS-MACとB-MACの動作を示す。MAC制御方式では通信を行っていないノードをスリープ状態にすることで消費電力を抑えている。

S-MACは無線回路を駆動している時間(アクティブ期間)と無線回路をスリープさせている時間(スリープ期間)を周期的に繰り返す動作を基本としている。アクティブ期間は更に同期期間とデータ期間に分かれており、同期期間において同期パケットをブロードキャストすることで、近傍ノードに自身のスリープスケジュールを通知し、それを受け取ったノードは自身のアクティブ期間を同期パケットの送信元に合わせることで、送受信ノード間でアクティブ期間を同期する。このようにS-MACでは、ノードのアクティブ期間を同期させることで、スリープによってパケットの受信に失敗することをなくしている。データパケットの送信はアクティブ期間中のデータ期間で行い、期間内ではCSMA/CAによるコンテンション回避を行い送信する。この方式では、ノード間で同期を行うため大規模なネットワークの同期に課題がある。

B-MACでは、センサノードは一定の動作周期ごとに数ミリ秒間無線デバイスをアクティブ状態にし、そ

れ以外ではスリープ状態をとる動作を基本としている。アクティブ時にはキャリアセンスを行い、チャンネルが使用中かを判定する。チャンネルが使用中であれば、アクティブ状態のままパケットの受信を待ち、チャンネルが使用されていない場合は再びスリープ状態に戻る。パケットの受信を待つ場合は、パケットを正しく受信するか、タイムアウトするまで待った後にスリープ状態に戻る。これに対して送信側ノードは動作周期以上の長さのプリアンブルをつけてパケットを送信する。受信ノードは動作周期ごとに必ずチャンネルをチェックしているため、送信元がプリアンブル送信をしている間にパケット受信待機状態に移行することができ、プリアンブルの後のデータを受信することが可能になる。しかし、送信時に必ずプリアンブルを送信するコストが発生する問題がある。

Zhuangbin Chen氏らの研究では、パケット転送フローで消費させる消費電力のフィードバック情報に基づいて、デューティ比を動的に変化させるAdaptation Duty Cycle Control (ADCC)[9]を提案している。

ADCCでは、シンクに近い領域のノードの消費電力が高く、シンクから遠い領域のノードの消費電力が低いために、大量の残留電力があることに注目している。この手法では各ノードはパケット転送時に自身の電力残量をヘッダへ付与することで、最も消費電力が多いノードを特定している。この情報をシンクノードから各ノードへフィードバックし、最も消費電力が大きいノードのデューティ比を減少させ消費電力を少なくしている。また、それ以外のノードではデューティ比を増加させ残余電力を活用することで伝送遅延を減らしている。

#### 4. 提案手法

センサネットワークでは、各センサの情報が取得できなくなると十分な機能を果さないため、各センサノードの寿命をできるだけ長くすることが重要である。デューティサイクルを利用したセンサネットワークでは、デューティ比の値はセンサノードの寿命と通信性能に大きな影響を及ぼす。ノード間で同期を取り通信時間のスケジューリングを作成すると、デューティ比を小さくしてもパケットの再送が増加することを防止しでき省電力なデューティサイクルを行える。そこで本研究では、デューティサイクルを考慮したセンサネットワークとして、非同期型プロトコルと同期型プロトコルを併用したハイブリッドプロトコルを提

案する。

#### 4.1 ネットワークモデル

前提としてノードが移動せず固定されたセンサネットワークを仮定する。デューティサイクルのスケジュールには、単純にアクティブとスリープを繰り返す単一アクティブモデルを採用する。ネットワークに存在するノードにはシンクノード、中継ノード、ソースノードの3種類があり、それぞれの動作について以下に示す。

- シンクノード

シンクノードは基本的にデータの送信を行わず、データの受信のみを行うノードである。シンクノードはサーバなどを想定しており、安定した電源供給がありデューティサイクルを考慮せず常にデータを受信できるものとする。

- 中継ノード

中継ノードは他のノードから受信したデータパケットを中継するノードである。シンクノードまたはソースノード以外は全て中継ノードに分類される。

- ソースノード

ソースノードはシンクノードを宛先としたデータパケットの生成を行うノードである。ソースノードは中継ノードの機能も兼ねており、他のノードから受信したデータパケットの中継処理も行う。

#### 4.2 ネットワークの階層化

提案手法では最初に各センサノードの隣接関係や親ノード候補を知るためにネットワークの階層化を行う。これらの処理はネットワークが展開された直後のみ行われる。

この手法では、シンクノードからホップ数を付加した制御パケットをブロードキャストすることで各ノードの階層を決定する。以下 Step1~4 で階層決定方法を示す。

##### Step 1

シンクノードは自身のホップ数に1を加算したものをデータとしてもつ制御パケットを生成し、周囲のノードへ送信する。このとき、シンクのホップ数は0とする。

##### Step 2

制御パケットを受信したノードは、制御パケットに含まれるホップ数を自身の階層値として保持する。既に階層値を保持していた場合、制御パケットのホップ数が現在の階層値よりも小さければこれを新しい階層値として保持し、ホップ数を1加算して周囲のノード

へ制御パケットを送信する。ホップ数が現在の階層値以上であれば、階層値を更新せず制御パケットを破棄する。このとき、制御パケットの送信元とそのノードの階層値（ホップ数-1）を隣接ノードとして保持しておく。

##### Step 3

Step 2 の処理を制御パケットが発生しなくなるまで繰り返す行い、全てのノードの階層値を決定する。

##### Step 4

各ノードは隣接ノードの中から自身の階層値より1だけ階層値の小さいノードを親ノード候補として保持し、自身と同じ階層値のノードを同階層ノードとして保持する。この親ノード候補及び同階層ノードがシンクへのパケット伝送時に次の送信先候補となる。

ここで、階層値はシンクノードまでの最短ホップ数を表すことになるため、シンクノードまでの距離と言い換えることができる。

#### 4.3 経路決定

提案手法においてシンクノードまでのルーチングは確率的かつ動的に行われる。各ノードはパケット送信時に親ノード候補の中から確率的に次の送信先となる親ノードを決定する。あるノードが親ノード候補の集合  $N_p$  の中からノード  $n_i$  ( $n_i \in N_p$ ) を親ノードとして選択する確率  $P(n_i)$  は式 (3) に従って計算される。

$$P(n_i) = \frac{d'(n_i)^2}{\sum_{n_j \in N_p} d'(n_j)^2} \quad (3)$$

ここで、 $d'(n_i)$  はノード  $n_i$  のデューティ比を表している。ただし、ノード  $n_i$  がソースノードのときは  $d'(n_i)$  を後述する  $d_{min}$  として計算する。ソースノードでは他のノードのデータパケット中継以外に自身のノード送信を行うため、選択確率を低くすることで負荷が集中することを抑えている。また、 $d'(n_i)$  の値が大きい方がより選択確率が高くなるように  $d'(n_i)^2$  として計算を行っている。

選択確率  $P(n_i)$  はデューティ比  $d'(n_i)$  に比例し、デューティ比と同様にバッテリー残量の減少とともに小さくなる。この関係性より、他のノードより多くデータ転送を行いバッテリーを消費したノードでは選択確率が低下し、相対的に他のノードが選ばれやすくなるため、負荷分散される。

ノードの配置によっては、よく利用される中継ノードが時間の経過とともに酷使され続け、結果として

バッテリー切れで利用できなくなる場合がある。シンクノードへの経路が失われてしまうとデータの収集ができなくなるため、親ノード候補  $N_p$  が空のときは隣接ノードを候補対象に追加する。この場合はシンクへのホップ数が増え遅延が増加するが、データの欠落を防ぐことができる。

#### 4.4 提案プロトコル

センサネットワークのデューティサイクルプロトコルには、大きく分けて非同期型と同期型に分けることができる。以下に非同期型プロトコルと同期型プロトコルの特徴を示す。

##### • 非同期型プロトコル

非同期型プロトコルでは、ノード間で同期を行わない状態で通信を行う。このため、デューティサイクルを利用したセンサネットワークでは、通信タイミングを決定するための制御パケットを交換する必要があり、余分に電力を消費してしまう問題がある。しかし、同期が不要であるためスケラビリティの点で同期型プロトコルより優れている。

##### • 同期型プロトコル

同期型プロトコルでは、ノード間で時刻同期を行い通信スケジュールの共有を行う。これにより、通信を停止させておく期間を理想的に決定することができる。そのため、通信タイミングを決定するための制御パケットを交換する必要がなく、電力消費を抑えることができる。しかし、同期を取ることに伴うオーバーヘッドやネットワーク全体で高い精度の時刻同期が必要など、スケラビリティの問題がある。

センサネットワークでは、従来のネットワークと違いシンクに近いノードと遠いノードで違った特徴がある。また、ノード間での時刻同期においても広範囲での同期維持が困難であることなど、スケラビリティの問題から異なった特徴がでてくる。以下にシンクに近いノードと遠いノードの特徴を示す。

##### • シンクに近いノード

センサネットワークでは、全ノードがシンクに向けたマルチホップ通信によりデータを転送する。そのため、シンクに近づくにつれて転送負荷が高くなり、消費電力が大きくなる。時刻同期においては、シンクからの距離が短く少ないホップ数でデータを転送できるので、同期誤差の影響が小さく同期がとりやすい。

##### • シンクから遠いノード

シンクから遠いノードでは、シンク周辺のノードとは異なりシンクから遠くなるにつれて、遠方ノードから

のデータを中継する頻度が少なくなる。そのため、通信回数が少なくシンク付近に比べ消費電力が低くなる。また、シンクからホップ数が増えるにつれて、同期誤差の影響が大きくなり同期がとりづらくなる。このため、ネットワークの規模が大きくなるほど同期誤差の影響が大きくなる。

シンクに近いノードではオーバーヘッドが大きくなるが同期を取ることは容易である。そこで提案手法は、同期型プロトコルを適用することで効率の良いデューティサイクルを行い消費電力を減少させる。一方シンクから遠距離のノードでは、同期型プロトコルを適用すると同期誤差などによるスケラビリティの問題が生じるため、非同期型プロトコルを適用する。このように提案プロトコルでは、スケラビリティを確保しながらシンク周辺のノードの消費電力を減らすことで、ネットワーク全体の消費電力のバランスをとりネットワークの長寿命化を図る。

##### 4.4.1 非同期型プロトコル

非同期型プロトコルでは、デューティ比は動的に決定される。ネットワーク階層化の完了後、各ノードのデューティ比  $d$  は、式 (4) によって動的に決定される。

$$d = \begin{cases} d_{min} & \left( \frac{E_r}{E\sqrt{L}} \leq d_{min} \right) \\ \frac{E_r}{E\sqrt{L}} & \left( d_{min} < \frac{E_r}{E\sqrt{L}} < d_{max} \right) \\ d_{max} & \left( d_{max} \leq \frac{E_r}{E\sqrt{L}} \right) \end{cases} \quad (4)$$

このとき、 $d_{min}$  はデューティ比の最小値、 $d_{max}$  はデューティ比の最大値を表すパラメータであり、 $0 < d_{min} \leq d_{max} \leq 1$  である。また、 $E$  はバッテリー容量、 $E_r$  はバッテリー残量、 $L$  はそのノードの階層を表している。各ノードの寿命延長を図るために、バッテリー残量が少ないほどデューティ比を小さくし、消費電力を抑えるようにしている。また、シンクノードから離れるほどトラフィック量が少なくなる傾向があるため、階層値が大きくなるほどデューティ比が小さくなるように設定している。ネットワークのエリアを円とし、円の中心にシンクノードがあるとき、第1階層の面積と第  $n$  階層の面積は  $n^2$  に比例して増加する。エリア内にノードが一様に分布している場合、各階層までに含まれるノード数も  $n^2$  に比例して増加する。このことから、ノードのもつ子孫ノードの総数は  $\sqrt{n}$  に反比例して減少していくため、階層の逆数の平方根をとっている。

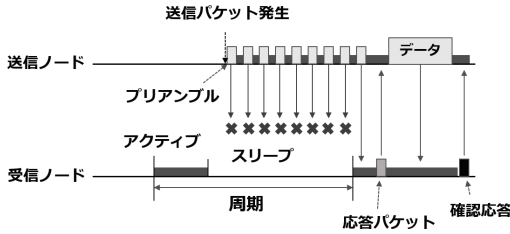


図5 非同期型プロトコルの動作  
Fig.5 Operation of asynchronous protocol.

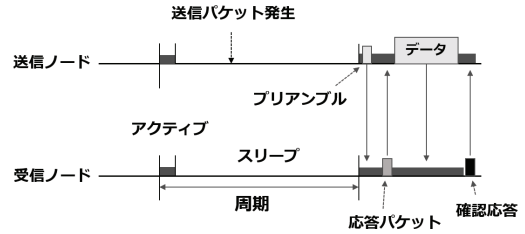


図6 同期型プロトコルの動作  
Fig.6 Operation of synchronous protocol.

デューティ比の計算にはバッテリー残量を用いており、バッテリー残量は時間の経過で変化するため、定期的な式(4)を用いて再計算を行う。

非同期型プロトコルの動作を図5に示す。データバケットが発生すると、送信ノードはプリアンブルにより受信ノードがアクティブ状態であることを確認する。受信ノードがアクティブ状態の場合、データバケットを送信する。そうでない場合は、受信ノードがアクティブ状態になるまでプリアンブルを送信し続ける。受信ノードは、データバケットを受信した後にACKを返すことにより、送信ノードに受信完了を通知する。ACKを受信した送信ノードは、通常のデューティサイクルに戻る。これを繰り返すことでパケットが伝送される。

#### 4.4.2 同期型プロトコル

同期型プロトコルでは、各ノードは時刻同期を行うことにより、同期中にデューティサイクルスケジュールを共有する。同期を行わない場合、ノードは隣接するセンサノードのデューティサイクルスケジュールを知らないため、送信先のアクティブ状態に合わせてプリアンブルを送信できない。そのため、デューティ比を小さくしすぎるとそのノードに対して送信を行うパケットの再送回数が増加し消費電力が大きくなるという問題点がある。しかし、同期型プロトコルではデューティサイクルスケジュールを共有しているため、送信先のアクティブ状態に合わせてプリアンブルを送ることができ、デューティ比の値を小さくしてもパケットの再送が増加しない。そのため、同期型プロトコルでは、ネットワーク通信開始時から各ノードのデューティ比を小さい値に固定することで効率のよいデューティサイクルを行う。

同期型プロトコルの動作を図6に示す。デューティサイクルスケジュールの同期により、送信ノードは隣接ノードがアクティブ状態になる時間を確認する。宛

先ノードがアクティブ状態の場合、データの送信が開始される。アクティブ状態でない場合、送信ノードは、宛先ノードがアクティブになるまでデータを保持し、その後データを送信する。各ノードはこれを繰り返すことでデータをシンクへ伝送する。

## 5. 性能評価

提案手法の有効性を示すためシミュレーション実験を行う。シミュレーションにはNS-3[10]を使用し、提案手法とADCC[9]の比較を行う。今回シンク周辺の時刻同期には、一方向で同期可能という性質からFTSP[4]を使用した。シミュレーション環境は以下を想定する。

ADCCの経路はダイクストラ法を使用した最短経路を採用する。

### 5.1 シミュレーションシナリオ

シミュレーションエリアは半径を200mの円と200m×800mの長方形を考え、その範囲内に一様にノードが配置されている。また、シンクノードをシミュレーションエリアの中央に配置した。

シミュレーションのパラメータを表1に示す。実環境に即したシミュレーションを行うため、各消費電力のパラメータは無線端末のMPR400CB[11]を参考に設定している。また、バッテリー容量は単3または単4の乾電池1本分を想定している。

デューティ比の設定について、小さいほど消費電力が抑えられるものの、小さすぎると通信の機会が極端に奪われ多量の再送が発生し、逆にネットワーク寿命が短くなってしまふ。非同期型のプロトコルをシミュレーションにより評価した辰巳の実験[12]で最良の結果となった $d_{min} = 0.1, d_{max} = 1.0$ を採用し以降のシミュレーションパラメータとして使用する。

パケット生成間隔は辰巳らの実験[13]を参考に60秒を基準として、誤差としてランダムな増減を加えた



表1 シミュレーションのパラメータ  
Table 1 Simulation parameters.

項目	パラメータ値
シンクノードの位置	エリアの中央
総ノード数	400
ソースノード数	20-300
パケット生成間隔	60sec
データパケットサイズ	256Byte
同期パケットサイズ	10Byte
同期間隔	60sec
通信距離	50m
通信速度	2Mbps
デューティサイクルの周期 $T$	1sec
デューティ比の最小値 $d_{min}$	0.1
デューティ比の最大値 $d_{max}$	1
デューティ比更新周期	1sec
バッテリー容量	10800J
1bit 送受信時の消費電力 $E_{elec}$	15nJ/bit
送信時の増幅電力 $\varepsilon_{amp}$	250nJ/bit/m <sup>2</sup>
アイドル時の消費電力 $E_{idle}$	30mJ/s
スリープ時の消費電力 $E_{steep}$	3μJ/s

ものを使用する。また同期間隔はパケット生成間隔に合わせた 60 秒に設定する。

本シミュレーションでは、各手法のネットワーク寿命と伝送遅延を測定するため、ノードの配置をランダムに変化させたシミュレーションを行った。得られた結果はブレが少なかったため 5 回のシミュレーションで十分だと判断し、その平均をとった。シンクヘデータを集約する際、いずれかのソースノードからシンクノードまでの経路が存在しなくなったときにネットワークの寿命が切れたものとし、ネットワークの寿命が切れるまでの時間を測定する。

### 5.2 電力モデル

無線通信による電力の消費モデルは、文献 [14] と同じものを用いる。  $k[\text{bit}]$  のパケットを  $x[\text{m}]$  離れたノードへ送信するときの消費電力  $E_{Tx}$  は式 (5)、  $k[\text{bit}]$  のパケットを受信するときの消費電力  $E_{Rx}$  は式 (6) でそれぞれ表される。

$$E_{Tx} = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot x^2 \tag{5}$$

$$E_{Rx} = E_{elec} \cdot k \tag{6}$$

ここで、  $E_{elec}$  は 1bit のパケットを送受信する際に消費する電力、  $\varepsilon_{amp}$  は送信時にかかる増幅電力を表している。

### 5.3 シミュレーション結果

まず、提案手法の同期型プロトコルを適用する範囲を決定するため、同期範囲を変化させたシミュレーションを行った。図 7 に円形のシミュレーションエリアの場合を、図 8 に長方形の場合のシミュレーション

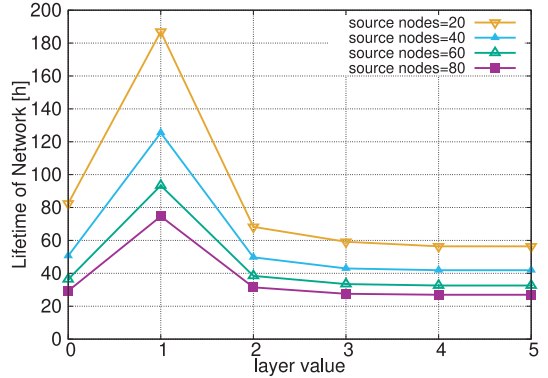


図7 同期範囲に対するネットワーク寿命 (円形)  
Fig. 7 Network lifetimes for layer values (circle).

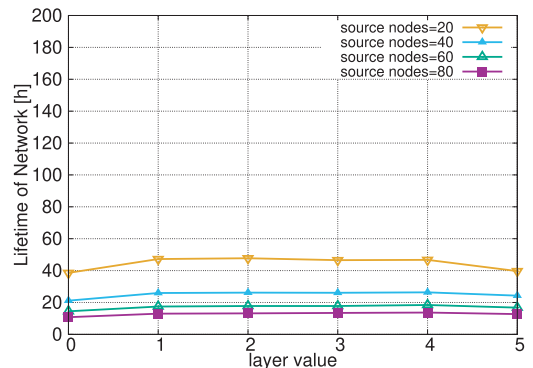


図8 同期範囲に対するネットワーク寿命 (長方形)  
Fig. 8 Network lifetimes for layer values (rectangle).

結果を示す。予備実験はシミュレーションを 50 回実施し、その平均をとっている。横軸が同期を行っている範囲、縦軸がネットワーク寿命となっている。円形のエリア設定では、全てのソースノード数において同期範囲を 1 階層までのノードにしたときに最もネットワーク寿命が長くなる結果が得られた。これは、同期範囲が広がることで同期を取るための制御パケットの通信が増えることにより、消費電力が増加したためだと考えられる。またエリア設定が長方形の場合も、円形と同じ 1 階層を含む幾つかの同期範囲でネットワーク寿命が長くなる結果が得られた。この結果から、以降のシミュレーションでは同期範囲をシンクから 1 階層以内のノードとした。

提案手法では、最適なパフォーマンスを発揮するために適切な階層値を設定する必要がある。実際の運用ではさまざまなノード配置が予想されるため、運用前のノード初期化時に最適な階層を決定するフェーズを

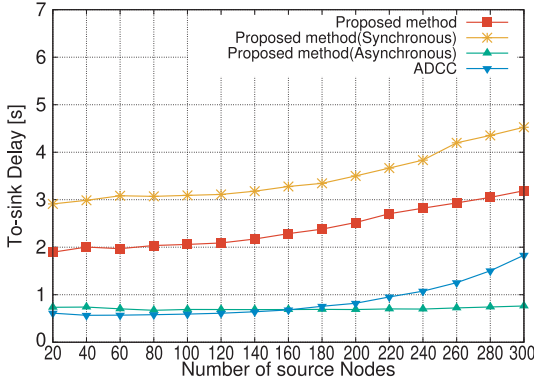


図9 円形エリアの伝送遅延  
Fig. 9 Transmission delay in circular area.

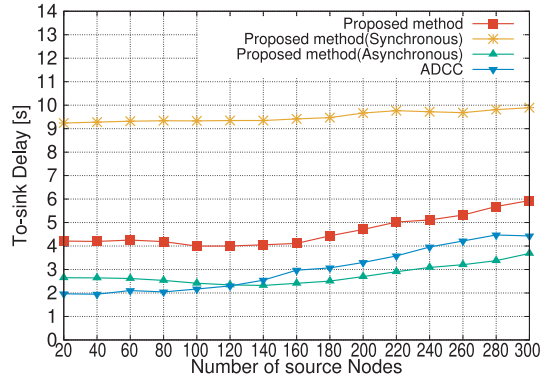


図10 長方形エリアの伝送遅延  
Fig. 10 Transmission delay in rectangular area.

必要とする。

次に、比較手法、提案手法、ネットワーク全体に同期型プロトコル、非同期型プロトコルのみを適用した提案手法の四つを比較する。図9、図10に伝送遅延のグラフを示す。このとき、図9は円形エリアであり、図10は長方形エリアである。

伝送遅延はソースノードからのパケット送信時刻とシンクへの到達時刻の差で求める。グラフではネットワーク内で発生した全ての通信について、平均を取ったものを採用している。

提案手法と同期型プロトコルのみを適用したものは他の手法より伝送遅延が大きくなった。これは、同期を行っていることが原因だと思われる。同期を行っているノードでは、デューティ比の値を動的に決定するのではなくネットワーク開始時から小さい値に固定している。これにより、データを受けとってから送信するまでの待機時間が増加し、伝送遅延が大きくなったと考えられる。一方、提案手法は同期型プロトコルのみを適用したものより伝送遅延が小さくなっている。このことから、同期・非同期を組み合わせることで伝送遅延を改善できていることが分かる。

図11、図12にネットワーク寿命のグラフを示す。このとき、図11は円形エリアであり、図12は長方形エリアである。全ての場合において、提案手法は他の手法よりネットワーク寿命が長くなっている。提案手法では、シンク周辺のノードのみで同期を取ることで、デューティサイクルによるパケットロスを防止しパケットの再送を減少させ、同期によるオーバーヘッドを減少させた。これによって、最もバッテリー残量が減少しやすいシンク周辺のノードの寿命を延長で

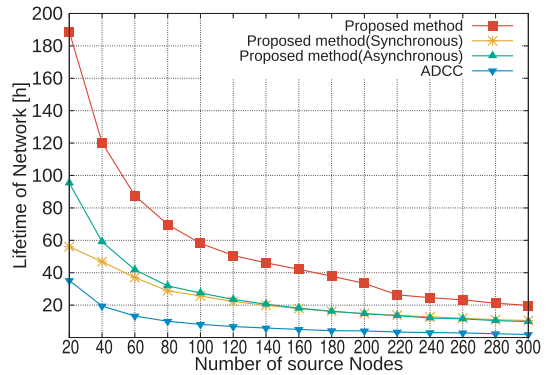


図11 円形エリアのネットワーク寿命  
Fig. 11 Network lifetimes in circular area.

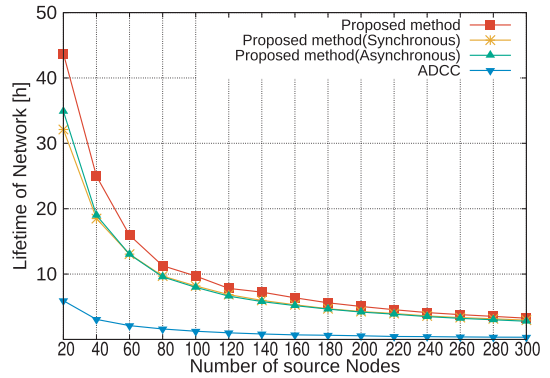


図12 長方形エリアのネットワーク寿命  
Fig. 12 Network lifetimes in rectangular area.

き、ネットワーク全体の寿命を伸ばすことができた。さいごに、図13にシミュレーションエリア形状ごとのネットワーク寿命の変化を提案手法、ADCCについて示す。また図14に同シナリオでの手法ごとの伝送

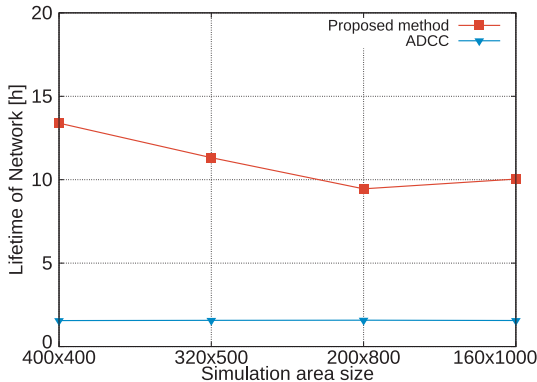


図 13 エリアの形状ごとのネットワーク寿命  
Fig. 13 Network lifetimes for each area shape.

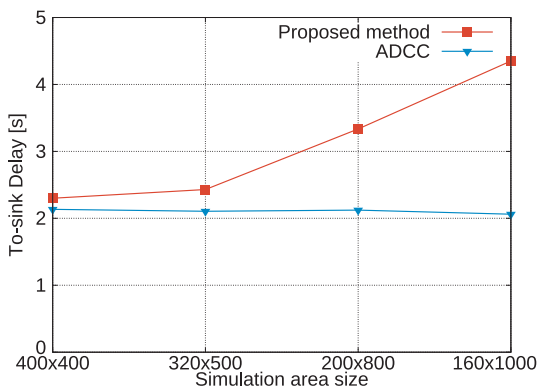


図 14 エリアの形状ごとの伝送遅延  
Fig. 14 Transmission delay for each area shape.

遅延について示す。エリア設定は正方形の  $400 \times 400$  から  $320 \times 500$ ,  $200 \times 800$ , 細長い  $160 \times 1000$  までの 4 種類、同じ面積の設定をした。ソースノード数は全て 80 ノードに設定した。ADCC についてはエリア形状の影響を受けにくく、細長いシナリオでも寿命がほとんど変化しなかった。いずれの場合も提案手法が長いネットワーク寿命を実現しているが、提案手法は正方形に近い形状になるほどネットワーク寿命を伸ばす傾向にある。また伝送遅延は提案手法がいずれの場合も多く、特にエリアが細長い形状になるにつれて大きくなる傾向が見られる。これはエリアが細長い形状になるとホップ数が増加し、更に横幅が狭くなったためシンクノードへ至る親ノード候補の選択肢が少なくなり、必ずしも最短経路ではないノードを経由した転送が増加したためと考えられる。

## 6. む す び

近年、センサネットワーク技術は様々な場面での活躍が期待され、注目されている。しかし、センサネットワークにおいて各センサノードは電力をバッテリーに依存していることが多く、センサノードの寿命を制限している。

本論文では、ノードが移動せず固定されたセンサネットワークにおいて、シンクに近いノードでは同期型プロトコルを適用することで効率の良いデューティサイクルを行い消費電力を減少させ、一方、シンクから遠距離のノードでは非同期型プロトコルを適用することでスケラビリティを大きくするハイブリッドプロトコルを提案した。更に、提案手法の有効性を示すため NS-3 を用いてネットワークシミュレーションを行った。結果より提案手法は従来手法に比べて遅延は大きくなるもののネットワーク寿命を伸ばすことができた。

本研究の一部は、JSPS 科研費 18K11484 と、JSPS 科研費 17K01309、関西大学大学院理工学研究科高度化推進研究費、関西大学先端科学技術推進機構「緊急救命避難支援のための情報通信技術に関する研究開発」研究グループの助成を受けている。

## 文 献

- [1] nsnam, ns-3 network simulator, <https://www.nsnam.org/> (accessed 2020-4-30).
- [2] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol.36, no.SI, pp.147–163, 2002.
- [3] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-sync protocol for sensor networks," *Proc. 1st International Conference on Embedded Networked Sensor Systems*, ACM, pp.138–149, 2003.
- [4] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," *Proc. 2nd International Conference on Embedded Networked Sensor Systems*, ACM, pp.39–49, 2004.
- [5] 木下和彦, 戸出英樹, 村上孝三他, "ノード性能の不均質な無線センサネットワークのためのハイブリッド型ルーチング方式," *信学論 (B)*, vol.96, no.2, pp.102–113, Feb. 2013.
- [6] R. Yadav, S. Varma, N. Malaviya, et al., "A survey of mac protocols for wireless sensor networks," *UbiCC journal*, vol.4, no.3, pp.827–833, 2009.
- [7] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw. (ToN)*, vol.12, no.3, pp.493–506, 2004.
- [8] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," *Proc. 2nd International Conference on Embedded Networked Sensor Systems*, ACM, pp.95–107, 2004.
- [9] Z. Chen, A. Liu, Z. Li, Y.-j. Choi, and J. Li, "Distributed duty cycle

control for delay improvement in wireless sensor networks,” Peer-to-Peer Networking and Applications, vol.10, no.3, pp.559–578, 2017.

- [10] T.R. Henderson, M. Lacage, G.F. Riley, C. Dowell, and J. Kopena, “Network simulations with the ns-3 simulator,” SIGCOMM Demonstration, vol.14, no.14, p.527, 2008.
- [11] I. Crossbow Technology, “Mica2 wireless measurement system,” [http://www.cmt-gmbh.de/Produkte/WirelessSensorNetworks/Datenblaetter/MICA2\\_Datasheet.pdf](http://www.cmt-gmbh.de/Produkte/WirelessSensorNetworks/Datenblaetter/MICA2_Datasheet.pdf) (accessed 2020-05-03).
- [12] 辰己真言, “センサネットワークにおける負荷分散を考慮したコンバージェンキャストに関する研究,” 関西大学理工学研究科修士論文, pp.7–33, 2015.
- [13] 辰己真言, 榎原博之, 松崎頼人, 菅沼初希, “センサネットワークにおける負荷分散を考慮したコンバージェンキャストの動的経路構築,” 研究報告数理モデル化と問題解決 (MPS), vol.2015, no.11, pp.1–6, 2015.
- [14] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” Proc. 33rd Annual Hawaii International Conference on System Sciences, IEEE, p.10, 2000.

(2020年1月28日受付, 5月18日再受付,  
7月15日早期公開)



榎原 博之 (正員)

1982 大阪大学工・通信卒. 1987 同大学大学院博士(通信)課程了. 同年大阪大学工学部助手. 1994 関西大学工学部専任講師となり現在, 教授. 組合せ最適化問題, 計算幾何学, 並列アルゴリズムなどの研究に従事. 工学博士. 情報処理学会, IEEE,

ACM 各会員.



川端 俊禄

2018 関西大学システム理工学部電気電子情報工学科卒. 同年関西大学大学院理工学研究科システム理工学専攻電気電子情報工学分野に入学. 無線センサネットワークの研究に従事.



松崎 頼人 (正員)

2011 関西大学システム理工学部電気電子情報工学科卒. 2016 同大学大学院博士課程総合理工学専攻了. 同年首都大学東京特任助教. 現在, 東京工業高等専門学校助教. 無線センサネットワーク, 無線アドホックネットワークの研究に従事. 情報処理学会,

IEEE 各会員.



徳永 潤平

2017 関西大学システム理工学部電気電子情報工学科卒. 2019 同大学大学院理工学研究科博士課程前期課程システム理工学専攻了. 同年より同大学大学院理工学研究科博士課程後期課程総合理工学総合理工学専攻在籍. 遅延耐性ネットワークの研究に従事.

情報処理学会学生会員.