

ESP8266EX を用いた無線センサ網構築のための低消費電力な通信プロトコル設計と基礎評価

A Low-power Communication Protocol and its Basic Evaluation for Wireless Sensor Networks with ESP8266EX Microchip

田代 志保† 吉廣 卓哉‡
Shiho Tashiro Takuya Yoshihiro

1 はじめに

近年、電源のない場所に無線通信機能をもつセンサを設置してネットワークを構築し、センサの観測値を無線マルチホップ通信により収集する無線センサネットワーク(WSN)の研究が実用化しつつある[1]。低消費電力化が進んだことにより、電源電力のない山間部や農地等の広い領域に設置することが可能であることから、環境や防災といった広い分野での応用が期待されている。また、WSN 向けの通信規格として IEEE802.15.4[2] が標準化され、製品化も進んでいる。

しかし、センサ端末を用いた商用サービスは、期待するよりも普及していない。この原因は複数考えられる。まず、センサ端末自体が高価であり、享受するサービスの価値に対して価格が高い点が挙げられる。センサ端末が量産されることによる価格の低下が期待されるが、現時点ではそのような目処は立っていない。次に、消費電力が意外に大きい点が挙げられる。商用サービスにするのであれば、例えば単純な環境モニタリングでは、単 3 乾電池 2 本で数年程度は動作することを期待したいが、現状の IEEE802.15.4 で無線マルチホップ通信を行うと、せいぜい数ヶ月といったところである。また、センサ端末の管理・運用上の問題として、動作確認や設定変更のために、物理的にセンサ端末を回収・接続をする必要がある点も、長期運用における課題である。

このような状況から、本研究では、現在安価に入手可能なマイクロコントローラである ESP8266EX を用いて、できるだけ現実的な無線マルチホップ通信を行うセンサ端末を試作する。ESP8266EX は Wi-Fi による通信機能を搭載している。Wi-Fi を用いたマルチホップ通信を実現するためには、いくつかの検討課題が考えられる。WSN として利用する場合には、消費電力を抑えるため、スリープなどを定期的に行い、かつ効率よくデータ転送を行う必要がある。また、安価なハードウェアでは時刻カウントに関する精度が低く、スリープを行う時間や時刻同期に誤差が発生するため、時刻同期誤差に対してロバストでなければならない。本稿では、これらの要件を満たすために、ESP8266EX で利用可能な Wi-Fi Direct を用い、時間分割したスロットにより、ノードが動的に親機と子機を切り替え、ノード間の時刻同期にずれがあってもロバストに動作する通信プロトコルを設計する。ESP8266EX は安価であるが、Wi-Fi を用いるため、消費電力が大きい問題がある。

これは、提案する通信プロトコルによって消費電力の低減を行った上、ある程度の大きさの太陽光パネル等を併用することで、電力収支をプラスにすることを旨とする。また、ESP8266EX は Wi-Fi 経由でログインやプログラムの書き換えが可能であり、これらの機能によって、将来的にセンサ端末の管理に係る負担の軽減が期待できる。

提案プロトコルでは、まず全ノードがデータ収集を行うシンクノードまでの距離(ホップ数)を把握する。次に、省電力化のためにセンシング間隔を 1 周期とし、その 1 周期を時間分割したタイムスロットを導入する。各ノードにホップ数によって動作するスロットを割り当てることにより、末端ノードからデータ送信を開始させることが可能となる。さらに、各ノードはシンクノードまでの距離が自身よりも小さいノードへデータ送信を行うことにより、シンクノードへデータを到達させる。この提案プロトコルを ESP8266EX に実装し、基礎評価を行ったので、ここに報告する。

本論文は以下のように構成される。第 2 章では実験で用いる ESP8266EX や Wi-Fi Direct などの基礎となる技術の説明する。第 3 章では安価に使用可能な ESP8266EX を用いた無線センサネットワークを構築するためのプロトコル設計を述べる。第 4 章では提案したプロトコルを用いて実機による評価実験を行った結果とその考察を述べる。最後に第 5 章で本論文をまとめる。

2 基礎となる技術

2.1 ESP8266EX

Espressif Systems 社によって開発、製造されている TCP/IP スタックとプログラム可能なマイクロコントローラを備えた低価格な Wi-Fi チップである[5]。対応している通信規格は IEEE 802.11 b/g/n であり、日本国内の技適マークを取得しているため、日本国内でも使用が可能である。2019 年 7 月時点で 1 つあたり 500 円と安価に入手可能であり、普及した Wi-Fi チップの 1 つである。また、Wi-Fi 通信を用いてのログインやプログラムの書き換えの機能がある。このチップを用いて無線センサネットワークの構築を行った際には、無線ログインやプログラムの書き換えの機能によってネットワークの管理が容易になることが考えられる。本研究では ESP8266EX を用いて実機実験を行う。

2.2 Wi-Fi Direct

Wi-Fi Direct とはアクセスポイントを用いずに機器が相互に接続するための技術である[4]。通信規格には Wi-Fi と同じく IEEE 802.11 a/b/g/n/ac が用いられる。通常の Wi-Fi では機器間の通信にはアクセスポイントを介する必要があるが、Wi-Fi Direct を用いると、端末が仮想的な親機である SoftAP (ソフトウェアアクセスポイント) となること

† 和歌山大学大学院システム工学研究科, Graduate School of Systems Engineering, Wakayama University

‡ 和歌山大学システム工学部, Faculty of Systems Engineering, Wakayama University

ができるため、アクセスポイントなしで通信を行うことができる。そのため、ルータや Wi-Fi ベースステーションなどのアクセスポイントとして用いるハードウェアがなくとも他の Wi-Fi 機器からの接続を受け付けることができる。

3 ESP8266EX を用いた無線センサ網のプロトコル設計

3.1 想定する無線センサネットワーク

本研究では、無線通信機能とセンシング機能を持った多数のセンサノードとセンシングデータを蓄積する役割を持ったシンクノード 1 つで構成されているネットワークを想定する。各センサノードは一定時間ごとに搭載したセンサから値を読み取ってセンシングデータを生成し、マルチホップ通信を用いてデータをシンクノードへ送信する。各センサノードはバッテリー駆動を前提としており、センサは安価なハードウェアである ESP8266EX を用いる。また、バッテリー駆動であることから、低消費電力で動作し、長期的な運用が可能であることが求められる。そのため、ある程度の大きさの太陽光パネル等を併用することにより、電力収支がプラスになることを目指す。

3.2 提案プロトコルの概要

本研究では、ESP8266EX で利用可能な Wi-Fi Direct を用いてセンサネットワークを構築するために省電力化した通信プロトコルを設計する。

提案手法では、各ノードは省電力化のために、センシング間隔を 1 周期とし、時間分割したスロットによって動作を遷移させる。スロットには、データを送信する送信スロットとデータを受信する受信スロットの 2 種類があり、それら以外のスロットでは省電力なスリープ状態となる。シンクノードから一番遠い末端ノードから送信スロットを開始することで、1 周期でセンシングデータの収集を可能にする。そのため、スロットの割り当てには各ノードのシンクノードまでの距離情報を用いる。ここで距離はシンクノードからのホップ数によって測るものとする。

各ノードはシンクノードまでの距離を把握するため、AP モードとなり、ビーコンフレームに含まれる SSID に距離情報を格納する。まず、自身の距離が確定するまではウェイク状態となる。Wi-Fi AP のスキャンを行うことスキャンングといい、このスキャンングを行う。隣接ノードから距離情報を受け取れば、その値に 1 を加えた値を自身の距離とする。自身の距離が確定すれば、AP モードとなり、SSID に距離情報を含めて広告を行う。以上の動作を末端ノードまで行うことにより距離情報の把握を行う。

各ノードは送信スロットのときに次ホップへデータ送信を行う。次ホップの選択はデータ送信毎に以下のように行われる。各ノードは隣接ノードの中から自身の距離より小さいノードを次ホップの候補とし、候補が複数あった場合には、その中からランダムに選択されたノードを次ホップのノードとする。

3.3 距離情報の伝搬

本論文で提案するプロトコルでは、まず、すべてのノードがシンクノードまでの距離（ホップ数）を把握する。初期状態では各センサノードはシンクノードまでの距離情報は持っておらず、シンクノードのみが距離情報を 0 として持っている。各ノードはまず、スキャンングを行い、次ホップとなるノードを探す。初めにシンクノードの隣接ノードが距離を 1 とし、距離のブロードキャストを行う。その隣接ノードがこれを検出し、自身の距離を 2 として距離のブ

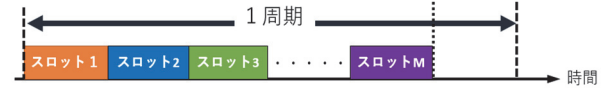


図1 スロットと1周期の関係

ロードキャストを行う。同様の動作を全ノードが逐次的に行うことにより、全ノードが距離の把握を行う。

各ノードの動作を詳しく述べる。ノード n_s はシンクノードまでの距離を把握するためにまず AP のスキャンングを行い、そのノード n_s と同じネットワークに属する隣接ノード集合 N_r を把握する。次ホップとなり得るノードが見つかる。そのノードをすぐに次ホップとするのではなく、一定期間はスキャンングを続ける。その後、次のように動作を行う。ノード n_s のシンクノードまでの距離 $d(n_s)$ と、ノード集合 N_r の中で距離が一番小さいノード n_r の距離 $d(n_r)$ を比較する。距離 $d(n_s)$ がまだ設定されていないか、 $d(n_s) > d(n_r) + 1$ であれば、 $d(n_s) = d(n_r) + 1$ とする。距離 $d(n_s)$ が確定すれば、シンクノードへ距離が決定したことを伝える距離決定データの送信を行い、その後、一定期間、距離のブロードキャストを行う。ブロードキャストは Wi-Fi Direct の AP モードを用いる。AP モード中は通常のインフラストラクチャモードの AP と同様に一定間隔でビーコンフレームが送信されており、ビーコンフレームに含まれる SSID に距離情報を含めることにより、距離情報のブロードキャストを行う。

すべてのノードがこの手順を繰り返すことにより、各ノードがシンクノードまでの距離を把握することができる。なお、ノードの故障等で距離の再設定を行う場合があるが、これについては 3.6 章で詳しく述べる。

SSID にはネットワークを判別するネットワーク ID、各ノードの距離、ノード ID から成り立っている。ノード ID はすべてのノードに対する識別を可能とするために、ネットワーク内で一意になるよう管理者が設定を行う。

3.4 時刻同期とスロット分割

全ノードは時刻同期を行い、時間分割した一定期間（タイムスロット）ごとに動作を遷移させる。タイムスロットの導入により、動作を行わないスリープ期間を長くすることができ、低消費電力な動作を実現する。また、各ノードが正しい時刻で動作を遷移させるために時刻同期を行う。

時刻同期は、ノード n_s の距離決定時に初めて行われる。ノード n_s は次ホップとしたノード n_r にデータを送信し、時刻情報をノード n_r から受け取る。その後、ノード n_s は自身の時刻を受信した時刻に設定する。これにより時刻同期を行う。すべてのノードがこの手順を行うことにより末端ノードまで時刻同期を行うことが可能となる。しかし、各ノードはスリープ等を繰り返すことにより、ノード間の時刻に誤差が生じる。そのため、データの送信を行うたびに時刻同期を行う。仮にノード n_s の時刻とノード n_r の時刻に誤差が生じたとしても、データを送信する毎に時刻が再同期される。

各ノードは距離（ホップ数）によってスロットの設定を行う。スロットにはデータの送受信が可能なウェイクスロットとスリープ状態となるスリープスロットのどちらかが設定される。周期とスロットの関係を図 1 に示す。最大スロット数を M 、スロット長を t_{slot} 、1 周期を R とすると $R > M \times t_{slot}$ となり、左辺は右辺に対して十分に大きいものとする。また、最大スロット数、スロット長、周期はそれぞれ時刻情報とともに次ホップから送信されるため、現在時刻から現在のスロットを計算することができる。スロット

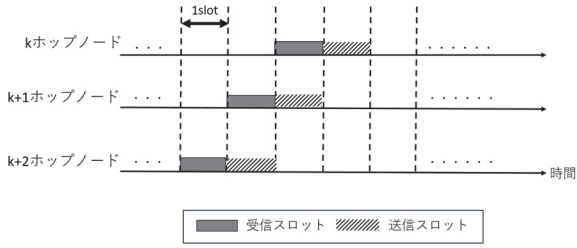


図2 k ホップノードから k+2 ホップノードの
スロット割り当て

割り当てと各スロットにおける詳しい動作については次章で述べる。

3.5 各ノードのスロット割り当て

本章では、スロットの割り当て方法と各スロットの動作を説明する。各ノードのウェイクスロットは受信スロットか送信スロットのどちらかであり、それ以外のスロットではスリープとしデータ通信は行わない。受信スロットではデータ受信を行い、送信スロットではデータ送信を行う。また、割り当てるスロットはノードの距離（ホップ数）から求められ、ある時刻のときに前後のノードでデータの送受信が可能となるようにスロットの割り当てを行う。全ノードは受信スロットののち、送信スロットへ移行する。例えば、 $k+2$ ホップのノードから k ホップのノードまでデータ送信を行う場合を述べる。 $k+2$ ホップのノードが送信スロットのとき、 $k+1$ ホップのノードは受信スロットであり、このとき、 $k+2$ ホップのノードから $k+1$ ホップのノードへデータ送信が行われる。また、 $k+1$ ホップのノードが送信スロットのとき、 k ホップのノードは受信スロットであり、このとき、 $k+1$ ホップのノードから k ホップのノードへデータ送信が行われる。図2は k ホップのノードから $k+2$ ホップのノードのスロット割り当てを表している。また、末端ノードからデータ送信を開始することにより、全ノードからシンクノードまでデータ送信を可能にする。

各ノード n の受信スロットと送信スロットの割り当てはノード n の距離情報 $d(n)$ と最大スロット数 M から、受信スロット $T_r = M - d(n)$ 、送信スロット $T_s = M - d(n) + 1$ として求める。スロットの割り当てにより、1周期のなかで末端ノードからデータ送信が開始される。また1スロットごとにデータはよりシンクノードに近いノードに送られていき、シンクノードまで各ノードのセンシングデータが届く。

各ノードは受信スロットでは、スロット開始からスロット終了までAP（アクセスポイント）として動作する。APである間はビーコンの送信が行われているため、距離情報を含むSSIDのブロードキャストが行われる。これにより、データ送信を行いたいノードはデータ受信が可能なノードを見つけることができる。またこの間データ受信が可能であり、ノードからデータを受け取れば時刻情報の送信を行う。

各ノードは送信スロットでは、スロット開始時刻からランダムな時間が経過してからデータ送信を行う。同ホップのノードは送信スロットが同じであり、隣接ノードである可能性が高い。そのため、スロット開始と同時にデータ送信を行うと、高周波で電波干渉が発生する。これを回避するためにランダムな時間、スリープとなってからデータ送信を行う。このランダムなスリープ時間 t_w は1スロットの長さ t_{slot} から、 $t_w < t_{slot}/2$ として求められる。これは、同

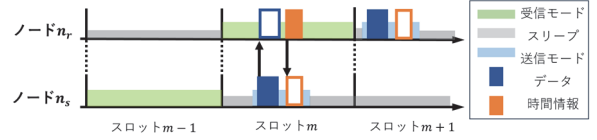


図3 k ノード n_s とノード n_r の動作

ホップのノード集合 N_e のランダムなスリープ時間 t_w が $t_w > t_{slot}/2$ として設定された場合、集合 N_e のノードはスロット終了までにデータ送信ができずに、受信ノードがスリープへ移行してしまう可能性があるからである。データ送信を行ったのち、受信ノードから時刻情報を受け取れば、次の受信スロットまでをスリープ時間と設定し、スリープへ移行する。

これら一連の流れを各ノードが周期的に行うことにより、各ノードからシンクノードまでデータの送信とシンクノードからの全ノードへの時刻同期が可能となる。

図3の例を用いて受信スロットと送信スロットでの詳しい動作を説明する。図3はスロット m のとき、ノード n_s からノード n_r へデータ送信を行う場合の動作である。ノード n_s はスロット m のとき、スロット長の半分以上のランダムな時間スリープしてから、ノード n_r へデータ送信を開始する。この間、ノード n_r は受信スロットであるため、ノード n_s からデータを受信すれば時刻情報をノード n_s へ送信する。ノード n_s は時刻情報を受信したならば、自身の時刻を再設定したのち、スリープとなる。

3.6 次ホップの決定

本章では各ノードの次ホップの決定方法を述べる。初期状態では、隣接ノードの中から最小ホップ数のノード1つを次ホップノードとして選択する。その後、各ノードは受信スロットのとき、自身のホップ数よりも小さいホップ数のノードを次ホップに選択する。これにより、データはシンクノードへ近づく。また次ホップに選択可能な隣接ノードの中で最小の距離のノードが複数存在した場合は、その中からランダムに1つのノードを次ホップノードとして、ノードの偏りを防ぐ。なお、送信スロット期間中に、自身のホップ数より小さいホップ数が見つからなければ、隣接ノードのなかで最小ホップ数を広告しているノードを次ホップと選択し、距離の再設定を行う。

初期状態では、ノード n_s の距離 $d(n_s)$ は未設定である。このとき、次ホップに選択可能な隣接ノード集合 N_r からシンクノードまでの距離（ホップ数）が最小のノードを次ホップの候補とする。この選択可能な隣接ノード集合 N_r はAPモードとなりSSIDを広告しているノードの集合であり、隣接ノード集合 N と等しい。次ホップの候補が複数あった場合、ランダムに選択することにより、データ受信を行うノードの偏りを防ぐ。

送信スロットのとき、 $d(n_s)$ は確定済みである。このとき、最大1スロット間スキューニングを行う。もし、受信ノード集合 N_r から距離 $d(n_s)$ より小さい距離のノードが見つければスキューニングを終える。この受信ノード集合 N_r は隣接ノード集合 N の部分集合であり、集合 N_r の要素はスロット毎に変わる。ノードが見つければ、その中から距離が最小のノードを次ホップの候補とする。次ホップの候補が複数見つかった場合、ランダムに選択することにより、データ受信を行うノードの偏りを防ぐ。選択したノードの距離 $d(n_s)$ により以下のどちらかの動作を行う。

- $d(n_r) = d(n_s) - 1$ の場合、受信ノード n_s にデータ送信

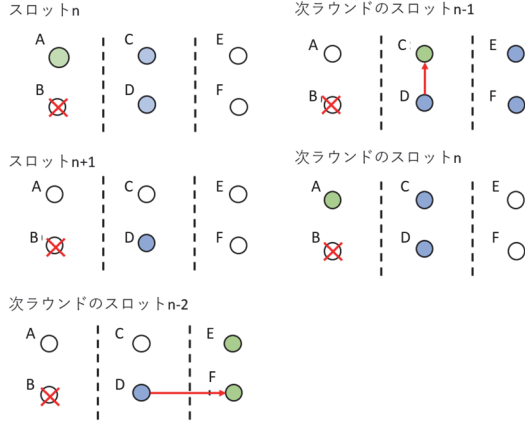


図4 距離再設定時の動作例

- $d(n_r) < d(n_s) - 1$ の場合、受信ノード n_s にデータ送信したのち、 $d(n_s) = d(n_r) + 1$ として距離を再設定し、ウェイクとなるスロット割り当てを変更

送信スロットのとき 1 スロット間のスキューニングを行っても、受信モードとなっているノード集合 N_r から $d(n_r) < d(n_s)$ となるノード n_r が見つからなかった場合について説明する．ここで次ホップが見つからない原因として、次ホップとしていたノードに故障や時刻に大きな誤差が発生した場合などが考えられる．集合 N_r の中から最小ホップ数を広告しているノードを次ホップの候補とする．次ホップの候補が複数あった場合、ランダムに選択することにより、データ受信を行うノードの偏りを防ぐ．選択したノードを n_m とすると、仮設定の距離を $d'(n_s) < d(n_m) - 1$ として設定を行う．距離の仮設定後は、2 スロット間はスキューニングを行い、距離 $d(n_m)$ よりも小さいホップ数のノードを探す．ここで 2 スロットの間スキューニングを行うのは、ノード n_s の隣接ノードすべてが受信モードとなる時間を網羅するためである．この 2 スロット中に、距離 $d(n_m)$ よりも小さいホップ数のノードが見つければ、再度距離を設定し、見つからなければ、 $d(n_s) = d'(n_s)$ として距離の設定を終える．

図4に動作例を示す．図4の例はノードDの次ホップであるノードBが何らかの原因で故障した場合の動作である．ノードDはスロットnのときにデータ送信を行うために次ホップを探す、隣接ノードで受信モードとなっているノードを見つけることができない．その後、受信ノードが見つかるのは次周期のスロットn-2のときである．このとき、ノードFを次ホップとして選択し、ノードDの仮設定の距離 $d'(D)$ を $d'(D) = d(F) + 1 = d(D) + 2$ と設定する．スロットn+1のとき、 $d'(D) > d(C) + 1$ となるノードCが受信モードとなるため、ノードDはノードCを次ホップとして選択する．ノードDの距離 $d'(D)$ は更新され、 $d'(D) = d(C) + 1 = d(D) + 1$ となる．スロットnのとき、受信モードとなっている隣接ノードは見つからない．以上で隣接ノードすべてが受信モードとなる時間を網羅することができる．よってノードDの距離を新たに $d'(D) = d(C) + 1$ として設定し、各スロットの割り当てを行う．

4 評価

4.1 評価方針

提案手法では、安価なマイクロコントローラである ESP8266EX を用いたセンサ端末を想定し、Wi-Fi Direct を用いた実用的な無線センサネットワークを実現する通信

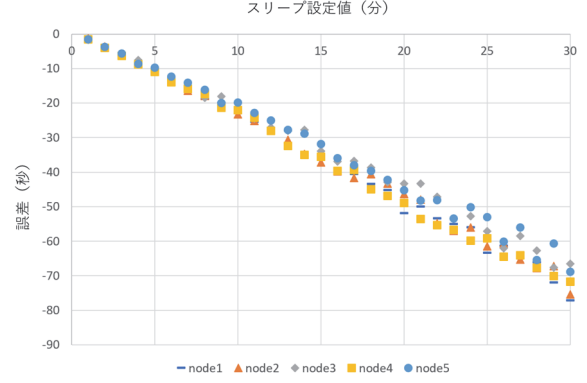


図5 スリープ設定値と誤差

プロトコルの設計を意図している．本章では、提案プロトコルを実機実装し、10 ノードを用いた簡単なシナリオを用いて基礎的な実機評価を行う．特に、提案手法においては、送信スロットと受信スロットを設定し、各ノードが適切な時間にウェイクとスリープを繰り返す周期的な動作を行うことが基本である．一方で、安価なマイクロコントローラでは水晶振動子の精度が低い等の理由で、スロット切り替えのタイミング同期に誤差が生じる可能性が考えられる．本評価では、そのようなタイミング誤差に対する提案プロトコルの頑強性を評価し、一定のタイミング誤差の下でセンシングデータの集約が可能であることを検証する．

4.2 タイミング誤差の測定

本研究で用いる ESP8266EX では水晶振動子の精度が低い等の理由で、各ノード間の時刻同期に誤差が発生する可能性が考えられる．特にスリープ動作における誤差は大きく、以下に複数の ESP8266EX に対してスリープ動作における時刻の誤差について調査した実験の結果を示す．

実験では、5 つの個体に対してスリープ設定値と実際にスリープを行った時間の調査を行った．スリープ設定値とはスリープ動作を行う前に、スリープを行う時間を設定する値のことである．スリープ設定値を 1 分から 1 分間隔で値を増やし、最大 30 分まで調査を行った．実験の結果を図5に示す．図5はスリープ設定値を横軸にとり、スリープ設定値と実際にスリープをおこなった時間との差分である誤差の値を縦軸にとった散布図である．誤差がマイナスの値である場合、スリープ設定値よりも実際にスリープを行った時間が短く、想定よりも早く起動したことを意味する．結果から ESP8266EX では、スリープ設定値が大きくなるほど、スリープ設定値よりも早く起動する傾向が観測された．どの個体であっても、誤差の大きさはスリープ設定値とある程度の相関があることが確認できる．また、スリープ設定値が 30 分である場合には、概ね 10 秒程度の範囲内で起動時間が分布することがわかる．それぞれの個体についてスリープ設定値に対する誤差の近似曲線を求めたところ、概ね約 4% の値をとったことから、評価実験ではスリープ設定値に対してスリープ動作を行う前に 4% の補正をいれることによって誤差を小さくすることとする．

4.3 評価シナリオ

提案手法を組み込んだ ESP8266EX をセンサノードとして室内に設置し、動作させた．また先述したとおり、スリープ動作を行う前にはスリープ設定値に対して 4% の補正を加えた．本実験では各ノードが次ホップに選択するノードは固定して行う．本来であれば次ホップの選択は自律的に

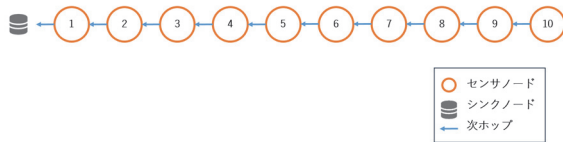


図 6 実験時のネットワークトポロジ

表 1 実験パラメータ

種別	意味	値
ノード配置	ノード間の距離	1.3[m]
	ノードの地上高	0.8[m]
パラメータ	スロット長	30[秒]
	1 周期	6[分]
	実験時間	17[周期]

表 2 ログに出力される項目

状態	ログ項目	値の種別
共通項目	スリープ設定値	数値
初期状態のみ	Wi-Fi スキャン結果 (検出 AP リスト)	文字列
	送信先 SSID	文字列
	ノード時刻	数値
	受信したノード時刻	数値
	ホップ数	数値
受信スロット	広告 SSID	文字列
	送信元ノード ID	数値
	スロット番号	数値
送信スロット	スキャン結果	文字列
	送信先 SSID	文字列
	ノード時刻	数値
	受信したノード時刻	数値
	スロット番号	数値

行うが、本実験では基礎的な動作実験を目的としているため、常に同じノードを次ホップとして選択するように設定した。つまり、各ノードは互いに近距離に設置されているが、ネットワークのトポロジは直線となる。図 6 にそのトポロジを示す。図 6 で示されるセンサノード内の数字は各センサノードの ID である。実験パラメータを表 1 に示す。表 1 における実験時間の 17 周期は、1 周期が 6 分間であることから 102 分間となる。

実験環境は室内であり、通常の Wi-Fi の AP 等が設置されていることから、電波干渉は頻繁に発生することが予想される。ESP8266EX 内に動作ログを保存する。この動作ログとして保存されるデータの一覧を表 2 に示す。ノードの状態ごとに表 2 のログ項目で示されている情報が ESP8266EX に保存される。ここで状態とは、距離が未設定時の初期状態、データ受信を行う受信スロットのとき、データ送信を行う送信スロットのとき、のいずれかである。

受信スロットと送信スロットの終了時に、その後にスリープを行う時間の設定値を保存する。スキャンを行う初期状態と送信スロットでは、スキャンを 2 秒間行った結果である隣接のノード SSID、選択した次ホップの SSID、受信したノード時刻とそのときの自身のノード時刻を保存し、送信スロットでは、自身の SSID と受信したデータの送信元ノード ID を保存する。

10 個あるノードの中で末端となるノードから起動し、その後、シンクノードの動作を開始させた。またシンクノードへ送信させるデータはダミーデータとし、データサイズは 2 バイトとした。

以上の条件のもと、センサノードを動作させ、ネットワ

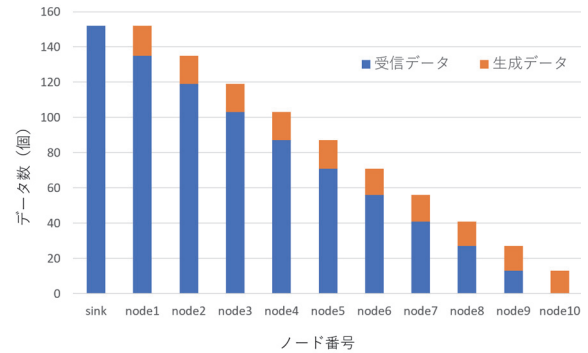


図 7 各ノードのデータ受信数と生成数

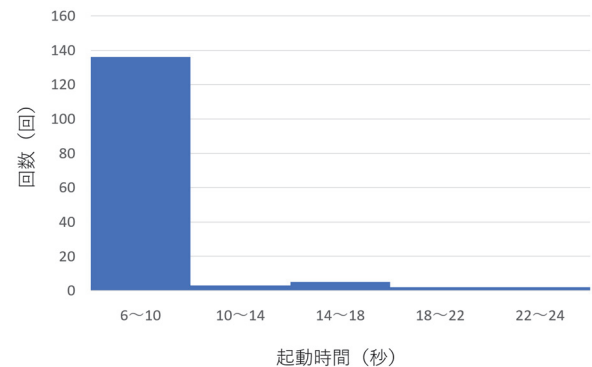


図 8 全ノードの送信スロット時の起動時間

表 3 各データがシンクノードへ到達するまでの時間

	距離決定データ	センシングデータ
1 周期で到達したデータ数	7	148
2 周期で到達したデータ数	3	0
合計	10	148

ークの構築を試みる。

4.4 結果

図 7 に、実験の結果得られた各ノードの受信データ数と生成データ数を示す。実験では、データは損失することなくシンクノードまで到達していたことが確認できた。

次に、各ノードがデータを生成してから、シンクノードに到達するまでの時間を表 3 に示す。距離決定データはセンシングデータではないため、データを生成したその周期内でシンクノードまで到達していない場合でも、問題にはならない。センシングデータ全 148 個のうち、すべてにおいてデータが生成されたその周期内にシンクノードへデータが到達している。このことから、すべてのデータは本研究ではスロットにより動作遷移を行うことで末端ノードからシンクノードまでデータを収集するため、データを生成したその周期内でシンクノードへデータが到達していればよいと考えられる。

最後に、ノードの送信スロット時の起動している時間を示す。図 8 は全周期、全ノードの送信スロット時の起動時間を 4 秒間隔でヒストグラムにあらわしたものである。この結果から、送信スロットでは起動してからデータ送信を完了しスリープするまでに、概ね 6 秒から 10 秒程度かかっ

ていることが確認できる。また、それ以上時間を要しているノードがあるが、これらについて調べた結果、親ノードとなるノードを見つけてから、コネクションを確立するまでに時間を要した場合と親ノードとの時刻同期において大きなずれが発生していた場合が確認された。実験では複数の AP が乱立しているような環境であったことから、電波干渉などにより時間を要したと考えられる。18 秒以上を要した場合もあるが、もしスロット内に送信完了できない場合にはデータ到達が 1 周期遅れるため、用途に応じた最適なスロット長の検討が必要になることを示唆している。本実験では、すべてのセンシングデータにおいて、生成されてから、1 周期以内にシンクノードへ到達していたことと送信スロット時の起動時間は最大 24 秒であったことから、すべてのセンシングデータは送信スロット以内に送信できていたことが確認できる。

以上の結果から、データ通信は問題なく行えており、ある程度のタイミング誤差の下でも、送信スロット時の起動時間が概ね 10 秒程度に抑えられることが確認できた。

4.5 考察

ESP8266EX が AP として動作する場合には、最大 4 つのクライアントと同時に Wi-Fi 接続が可能であり、同時接続数に制約がある。実験結果から、クライアントからのデータ送信に約 10 秒が必要であるため、想定するクライアント数に応じてスロット長を検討する必要がある。一樣配置の場合には、クライアントが AP をランダム選択することから、平均的な接続クライアント数は大きくならず、ある程度の偏りに耐えられるだけのスロット長を設定すれば良い。一方で、タイミング同期誤差の影響もスロット長の設定に影響する。今回の実験ではスロット長を 30 秒とし、前半 15 秒の間でクライアントがランダムに起動するように設定したが、この場合にはタイミング同期誤差が 15 秒を超えると、スロット終了後にクライアントが起動することになり、動作周期内でのデータ送信に失敗する結果となる。本実験ではスリープ動作に伴う起動時刻の誤差に対して、スリープ設定値と誤差の値にはある程度の相関があることを利用し、スリープ設定値に補正を加えた。しかし、補正を入れたとしても個体差などから値のばらつきが発生するため、スリープ時間の設定値は値のばらつきがスロット長の半分より小さい値で設定する必要があると考えられる。1 周期、及び 1 スロットの長さは、消費電力に大きな影響を及ぼす。本研究では、小型太陽光パネル等の設置により、エネルギー収支をプラスにすることを意図するが、どの程度の給電を想定するかにより、許容できる周期が変わる。今後は給電性能を考慮した周期の設定について検討を進める必要がある。なお、このバランスを超える周期を設定したいのであれば、RTC(Real-time Clock) モジュールを接続する必要があるが、センサ端末のコストが少し上がる。一般的な RTC モジュールを接続した場合には、時刻の誤差は 1 ヶ月あたり約 1 分程度になる。

5 おわりに

本研究では、ESP8266EX を用いた無線センサ網構築のための低消費電力な通信プロトコルを提案した。提案プロトコルでは、ノードの動作をタイムスロットにより遷移させることで省電力化し、また自身の距離に基づいてスロットの割り当てを行うことにより、センサネットワークの構築を実現するものである。また実験では、提案手法を ESP8266EX に組み込み、基礎評価を行った。

実験の結果、データ到達率は 100 % であり、送信スロ

ットでは起動してから親ノードを見つけ、データ送信を行い、スリープするまでの時間が概ね 10 秒に抑えられることが確認された。1 周期におけるノードの起動時間は受信スロット時のスロット長の時間と送信スロット時の 10 秒であり、1 周期の時間が長くスロット長が短い場合、消費電力は抑えられると考えられる。また時刻同期に誤差があったとしても補正を加えることにより、ロバストに動作することが確認された。

今後の課題として、現在の起動時間や通信にかかる時間等から、バッテリー消費量を計算し、どの程度の長期的運用が可能なのかを検討する必要がある。また、将来的には実際にバッテリーを用いて計測することを目標としているが、この場合、バッテリーの残量電力が枯渇するまで長期に使用し、結果を得る必要がある。また、本稿ではネットワークポロジは直線とし親ノードを固定して実験を行ったが、ランダムにノードを配置し、自律的に親ノードを選択する場合においても実験を行う必要があるため、今後の課題とする。

参考文献

- [1] 阪田史郎, 戸辺義人, 南正輝, 猿渡俊介, 鈴木誠, 石原進, 若宮直紀, 鈴木敬, 西山裕之, 福永茂, 河野隆二, “3 章 センサネットワーク”, 電子情報通信学会「知識ベース」4 群-5 編 (2010).
- [2] IEEE Computer Society, “IEEE Std 802.15.4-2003, IEEE Standard for Information technology-Telecommunications and information exchange between systems- Local and metropolitan area networks Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer,” (2003).
- [3] Wi-Fi Alliance, “Wi-Fi Direct,” 入手先< <https://www.wi-fi.org/ja>>(参照 2019-07-01)
- [4] Espressif Systems, “ESP8266EX Overview,” 入手先< <https://espressif.com/en/products/hardware/esp8266ex/overview>> (参照 2019-07-01)