

# 電池残量にもとづく無線中継機能のON/OFFによるデータ受信期間の延長

杉本 一彦<sup>1</sup> 串田 高幸<sup>1</sup>

**概要：**水田の水質監視での IoT デバイスを利用する需要が増加している。水田がある屋外では電池を用いて IoT デバイスを動作させる。送信されるデータは IoT デバイスを中継しサーバに届く。しかし中継機能によって常にデータが転送されるため消費電力が削減できず、データの受信期間が減少する。データ受信時間の減少を解決するため中継機能を電池残量と接続先の状況に基づいて ON/OFF を行うことでより長い期間データを受信する手法を提案する。末端デバイスが接続可能な中継デバイス群をグループとする。どの中継デバイスの中継機能を起動するかは重みと電池残量によって決められる。中継デバイスへの重み付けは中継デバイスが属しているグループ数となる。重みが最大と最小である中継機能を起動させる。各グループにて最低 1 台以上の中継デバイスの中継機能が起動するようにする。評価では ESP32 を 9 台用い、中継デバイスが 5 台、末端デバイスが 4 台となるように設置し実験を行った。電池容量が 1,000[mAh] ありとしてサーバにて末端デバイスのセンサーデータを受信した期間を比較した。評価結果ではセンサーデータの最大受信期間が既存手法では 8.0 時間、提案手法が 9.7 時間となり 21%伸びた。

## 1. はじめに

### 背景

水田の水質監視における IoT を活用する需要は増加している [1]。IoT は近年急速に成長している [2]。水田における IoT の活用の背景には日本の農業従事者の高齢化と労働者不足が要因の 1 つとして挙げられる [3]。この背景による課題を解決するために IoT を利用したスマート農業が用いられており、人的資源や高い費用対効果を得ることができる [4]。屋外において電力やネットワークインフラを整備することなく IoT デバイスを動作させる必要がある。IoT デバイスは一次電池や二次電池を用いることで電力システムからの電力供給を受けることなく動作することが可能である。また付近にインターネットに接続するための Wi-Fi アクセスポイントがない場合、既に接続されている他の IoT デバイスを介してセンサーデータを送信する手法であるメッシュネットワークも用いられている [5]。

### 課題

電池で動作する IoT デバイスにてメッシュネットワークを構築する際、センサーデータが中継機能によって常に転送されるため、消費電力が削減できていないことが課題で

ある。図 1 にて課題の例を示す。

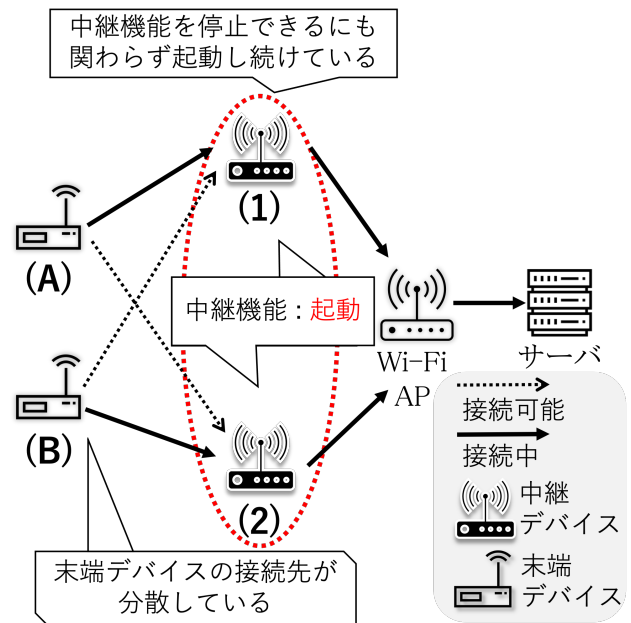


図 1 中継機能を停止できるにもかかわらず起動している際の課題

図 1 では中継デバイスは 2 台とも中継機能を起動している。末端デバイス (A) は中継デバイス (1) へ接続しており、末端デバイス (B) は中継デバイス (2) へ接続している。それぞれの末端デバイスは接続している中継デバイスを介してサーバへデータを送信する。なお Wi-Fi AP は Wi-Fi のアクセスポイントである。また図 1 ではそれぞれの末端

<sup>1</sup> 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻  
〒192-0982 東京都八王子市片倉町 1404-1

デバイスは全ての中継デバイスへ接続できる状態となっている。

図1における課題は片側の中継デバイスの中継機能を停止できるのにも関わらず、起動し続けていることである。末端デバイス(A)と(B)はどちらの中継デバイスにも接続することが可能である。またどちらか一方の中継デバイスのみにも接続することも可能である。中継デバイスの中継機能を起動している状態は消費電力が停止時より高い状態となる。従って中継機能が起動している中継デバイスの台数が増えることで消費電力が増加し、中継デバイスの動作時間が短くなることでサーバのセンサーデータ受信期間も減少する。

## 2. 関連研究

Jiang らは Wi-Fi とサブギガ帯無線通信を組み合わせた低電力な IoT メッシュネットワークを提案している [6]。パケットの衝突を避けるために多くの IoT デバイスへデータを転送することにより IoT デバイスのスリープ時間を削減する可能性がある。また電力の大部分を消費するデータの転送における省電力手法が取られていない。

Nair らはメッシュネットワークにおけるスター型とメッシュ型を組み合わせる手法を提案している [7]。データ収集を一部の IoT デバイスが行うことで全体としての平均消費電力を削減している。この手法は他のピコネットへデータを転送する必要があるためにデータの中継機能を起動したままとしている。これはセンサーデータ受信期間を減少する要因である。

Sahitya らは農業用のワイヤレスセンサーネットワークを提案している [8]。経路の生成には電力消費を削減するために電波強度を計測している。しかし電池残量によって使用可能な電力が異なるため電池残量を考慮した上で経路を決定する必要がある。

## 3. 提案手法

提案手法は中継機能を電池残量と末端デバイスの接続状況に応じて中継デバイスの中継機能の ON/OFF を行う。これにより末端デバイスからサーバへデータが送信される期間の延長を行う手法を提案する。提案手法における前提条件は IoT デバイスは既に設置されているものとし、追加、撤去、移動を行わないものとする。末端デバイスは2台以上の中継デバイスと接続されるものとする。またネットワークにおける最大のホップ数は2とする。

### 基礎実験

本節では課題で先述した中継機能の起動により消費電力が高くなることを確認するために基礎実験を実施した結果を示す。また通信時における消費電力や、転送データ量に

おける消費電力の比較を行う。

基礎実験で使用した IoT デバイスは ESP32 である。消費電力の測定は 0.5 秒に 1 回とし、測定したセンサーは INA219 である。

図2は ESP32 の5つの処理状態における消費電力を計測した結果を示している。

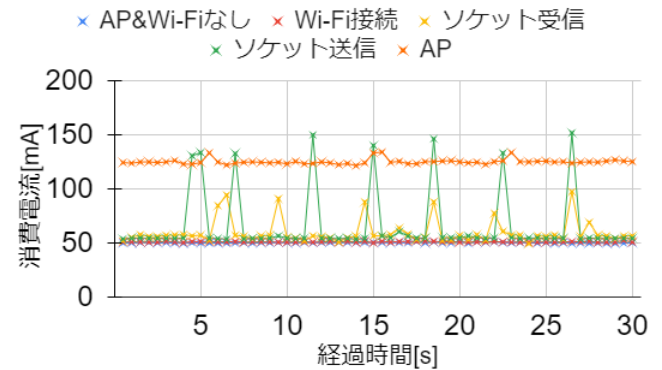


図2 各計測手法における消費電力の比較

図2にて示している5つの処理状態は中継機能を停止し、かつ Wi-Fi への接続をしない (AP&Wi-Fi なし) 場合、中継機能を起動している (AP) 場合、Wi-Fi へ接続しデータの送受信は行っていない場合 (Wi-Fi 接続)、ソケット通信にてデータを4秒ごとにデータの送受信を行なっている場合 (ソケット受信/送信) である。送受信におけるデータは ESP32 の固有の ID である 11[Byte] と温度データの 7[Byte] を足し合わせた合計 18[Byte] である。Wi-Fi 接続時の消費電力は平常時と大きな差はない。中継機能を起動している際の消費電力は平常時の約 1.8 倍ほど高い。ソケット通信における送受信時における消費電力は受信時が送信時の約 1.3 倍ほどになる。図2ではソケット通信によるデータ送受信のタイミングが正確な4秒間隔となっていない。これは実験で仕様した ESP32 のスリープ時間の精度が低いことが原因であると考えられる。

図3は図2にて計測した消費電力の平均値をグラフとして示したものである。

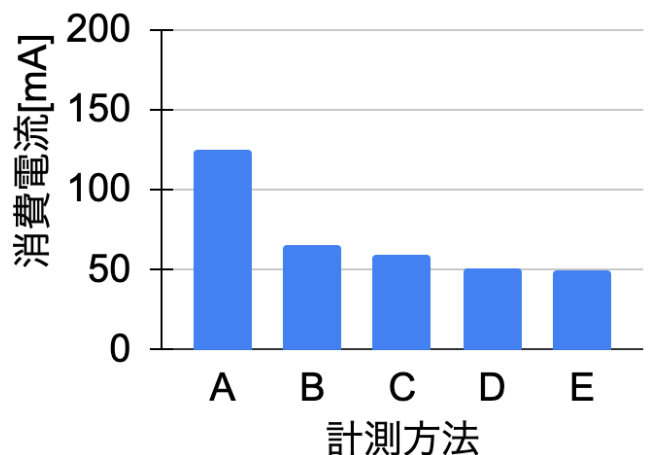


図3 基礎実験における消費電力の平均値の比較

図3における横軸はESP32の動作状況、縦軸は消費電流である。消費電流の平均は高い順に A. 中継機能起動時が125.2[mA], B. ソケット送信が66.0[mA], C. ソケット受信が59.8[mA], D. Wi-Fi 接続が51.1[mA], E. 中継機能停止&Wi-Fi 接続無が50.2[mA]である。

次に転送するデータ量を5段階に分けて消費電力の増加率を測定した。1つの転送データは合計18文字の文字列データである。データはESP32の固有のIDである11[Byte]と温度データの7[Byte]を足し合わせた合計18[Byte]である。基礎実験では転送データ量を0, 18, 36, 54, 72, 90[Byte]と変化させてそれぞれ消費電流を測定し、その増加率を示した。また転送データ量はペイロードのデータ量を指している。結果を図4に示す。

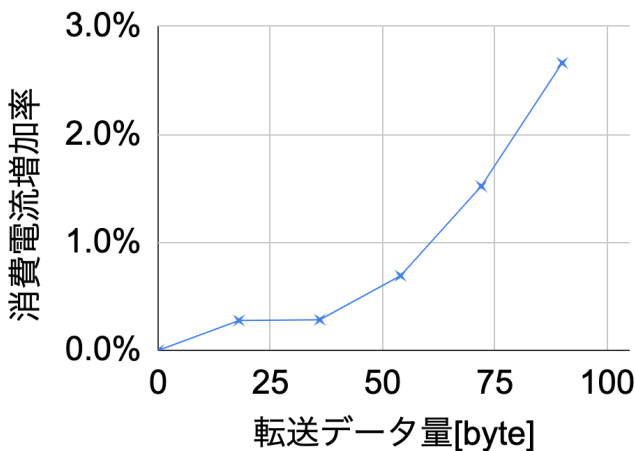


図4 転送データ量に対する消費電流の増加率

図4におけるそれぞれの転送データ量の消費電流の増加率は転送データ量が18[Byte]の場合は0.27%, 36[Byte]の場合は0.28%, 54[Byte]の場合は0.69%, 72[Byte]の場合は1.52%, 90[Byte]の場合は2.66%である。この結果より転送データ量が増加するにつれて消費電流は増加する傾向がある。しかし転送データ量が0[Byte]と90[Byte]の消費電流を比較した場合、2.66%である。基礎実験の結果より転送データ量が増加したとしても消費電流の増加率が約3%以内と低いため動作時間に大きな影響を及ぼすことにはならない。転送データ量の増加と比較し、中継機能を起動することで停止している場合と比べ消費電流が1.7倍から1.9倍ほど高くなる。このことから転送データ量を増加させたとしても中継機能を停止させることでより消費電力を削減することが可能である。

### 提案手法

中継デバイスの中継機能を起動と停止を動的に行うことでセンサーデータがサーバへ送信される期間を延長する手法を提案する。図5へ本提案手法の概要を示す。

図5では前提条件と合致しているIoTデバイスによるメッシュネットワークの一例を示している。末

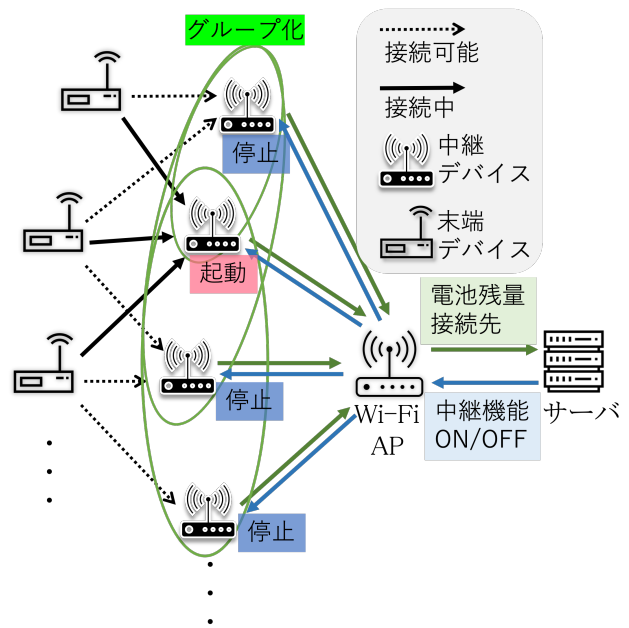


図5 提案のイメージ図

端デバイスはセンサーデータをサーバへ送信することを目的としている。しかし末端デバイスはWi-Fi APへ直接接続することができない。したがって末端デバイスが送信するセンサーデータは中継デバイスを経由してサーバへ送信される。

中継デバイスの中継機能の起動と停止はサーバが中継デバイスへ指示をすることで実行される。また中継デバイスは電池残量と、どの末端デバイスが接続可能であるかの接続先情報をサーバへ送信する。これらのデータはサーバが中継デバイスの中継機能の起動と停止を定める際に必要となる。従って中継デバイスはこれらのデータを予め定められたタイミングでサーバへと送信する。なお末端デバイスのセンサーデータの転送とはまた別の処理となる。本提案におけるフローを図6に示す。

図6で示すフローチャートでは合計10個の処理と判断がある。提案手法は主に3つの目的によって構成されており、それを下記へ示す。

- 1 グループ化と重み付け
- 2 中継機能を起動する中継デバイスの決定
- 3 中継デバイスが停止した場合の中継機能を起動するデバイスの再決定

### 1 グループ化と重み付け

グループ化と重み付けは中継機能を起動する中継デバイスを決定するために必要な処理となる。図6では処理①と①に当たる。

中継デバイスのグループ化は末端デバイスからセンサーデータを確実にサーバへ送るために必要な処理である。グループにおいて1台以上の中継デバイスの中継機能が起動することにより末端デバイスがセンサーデータを送信できない状況を回避する。グループ化は中継デバイス

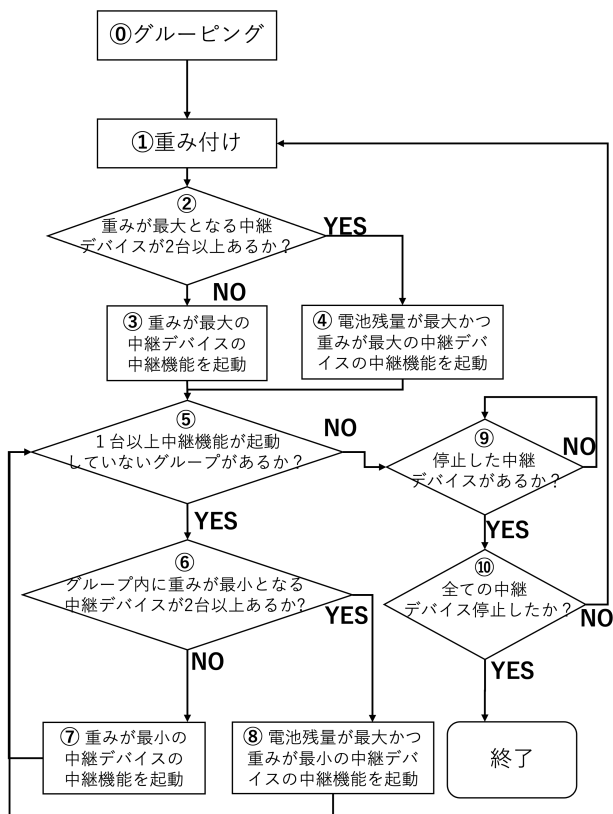


図 6 中継機能の起動と停止を決定するアルゴリズムのフローチャートに対して行われ、グループの付与はサーバにて処理される。グルーピングは末端デバイスの接続先の中継デバイスを1つのグループとする。図7ではグルーピングの一例を示している。

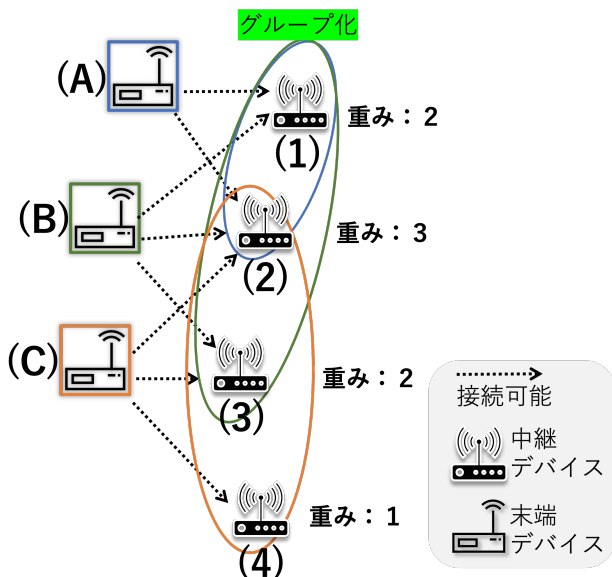


図 7 中継デバイスのグルーピングと重み付け

図7における末端デバイス(A)は中継デバイス(1)と(2)に接続可能であるため青色の枠線で示されたグループが生成される。末端デバイス(B)は中継デバイス(1),(2),(3)に接続可能であるため緑色の枠線で示されたグループが生成される。末端デバイス(C)は中継デバイス(2)と(3),(4)

に接続可能であるため橙色の枠線で示されたグループが生成される。生成されたグループは処理が全て終了するまで変更されることはない。

中継デバイスのグルーピングが完了すると次にそれぞれのの中継デバイスへ重み付けを行う。中継デバイスの重み付けでは中継デバイスがいくつのグループに属しているかを重みとしている。重みは全てのの中継デバイスにそれぞれ与えられる。末端デバイスは必ず2台以上の中継デバイスと接続される。従って付与される重み  $R_w$  は  $1 \leq R_w \leq E_n$  となる。 $E_n$  は末端デバイスの台数である。重み付けされた数値は中継機能の起動と停止処理の際に必要なとなる。

図7ではそれぞれの中継デバイスに重みを与えている一例を示している。中継デバイス(1)は末端デバイス(A)と(B)によって生成されたグループに属しているため重みは2となる。中継デバイス(2)は末端デバイス(A),(B),(C)によって生成されたグループに属しているため重みは3となる。中継デバイス(3)は末端デバイス(B)と(C)によって生成されたグループに属しているため重みは2となる。中継デバイス(4)は末端デバイス(C)によって生成されたグループに属しているため重みは1となる。

## 2 中継機能を起動する中継デバイスの決定

重み付けの後、中継機能を起動する中継デバイスの決定を行う。中継機能を起動する中継デバイスを決定には末端デバイスからのセンサーデータを確実に中継させることを最優先とする。このため生成したグループを利用する。グループは末端デバイスの接続可能な中継デバイスによって構成されているため、グループの中で最低1台の中継デバイスの中継機能が起動すれば末端デバイスからセンサーデータの送信が可能となる。従って決定には最終的に全てのグループで最低1台以上の中継デバイスの中継機能を起動するように設定する。この処理を行なっているのが図6における判断②から処理⑧までとなる。

中継機能を起動する中継デバイスの決定手法は重みと電池残量によって求められる。まず初めに中継デバイス全体の重みと電池残量を参照し決定を行う。これは図6における判断②から処理④に当たる。中継機能を起動する中継デバイスの台数は可能な限り最小となるように設定する。従って複数台の中継デバイスの中継機能を起動させる必要はない。この処理では重みが最大である中継デバイスの中継機能を起動するデバイスを1台とするために、重みが最大の中継デバイスが2台以上あるか判別している。重みが最大の中継デバイスが1台のみであれば、その中継デバイスの中継機能を起動する。重みが最大の中継デバイスが複数台あれば電池残量を比較しより多い中継デバイスの中継機能を起動する。電池残量が多い中継デバイスを選択する理由はより多い電池残量を持つ中継デバイスの中継機能を起動することでより長い時間、その中継デバイスを動作さ

せることができることである。

中継デバイス全体の重みと電池残量から1台のみ中継機能を起動する中継デバイスを決定した後、全てのグループにおいて中継機能が起動しているかの確認を行う処理を行う。これは図6における判断⑤となる。仮にグループ内に中継機能を起動している中継デバイスがない場合、末端デバイスからセンサーデータを送信することができなくなってしまうためである。

グループの中に中継機能を起動した中継デバイスが1台も存在しなかった場合は、そのグループの中において中継機能を起動する中継デバイスを決定する。ここではグループに属している中継デバイスの重みと電池残量をもとに決定する。この決定の際にはグループ内で最も重みが小さいものが優先される。図8にてグループ内で重みが最小となる中継デバイスの中継機能を起動する利点を説明する。

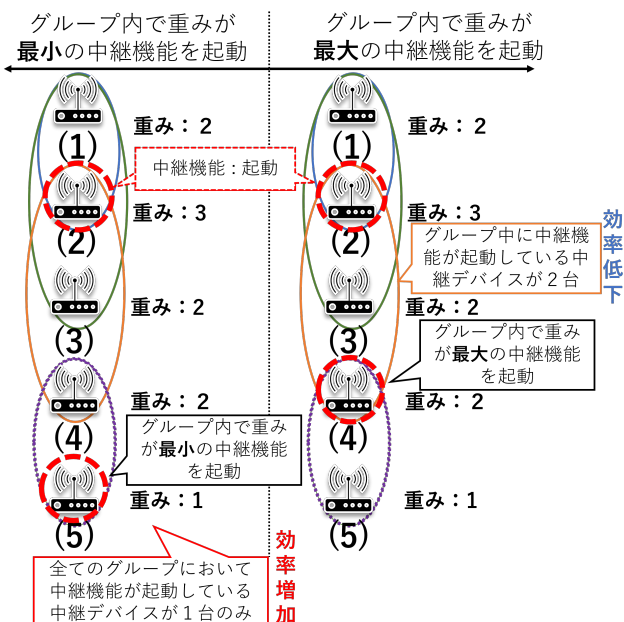


図8 グループ内で重みが最小となる中継デバイスの中継機能を起動する利点

中継デバイス(2)は処理③にて中継機能を起動している状態となっている。図8は構成2つを左側と右側に分けて示している。左側の構成は1台も中継機能が起動していない中継デバイスがあるグループにおいて重みが最小となる中継デバイス(5)の中継機能を起動した場合を示している。右側の構成は1台も中継機能が起動していない中継デバイスがあるグループにおいて重みが最大となる中継デバイス(4)の中継機能を起動した場合を示している。

右側の構成において中継デバイス(4)の中継機能を起動した場合、既にグループ内に中継機能が起動している中継デバイスがある。グループ内に複数の中継機能が起動している中継デバイスがある場合、中継機能を停止できるはずであるにも関わらず中継機能を起動していることになり電力を削減できない。

左側の構成において中継デバイス(5)の中継機能を起動した場合は全てのグループにおいて中継機能が起動している中継デバイスが1台のみとなる。全ての中継機能が起動している中継デバイスへ末端デバイスのセンサーデータが送信され、効率よく中継デバイスの中継機能を起動することができる。従って消費電力を削減することができる。

### 3 中継デバイスが停止した場合の中継機能を起動するデバイスの再決定

1台以上の中継機能が起動していないグループがないと判定された場合にこの処理へ進む。ここでは処理①から⑧にて中継機能を起動する中継デバイスを決定したのち、いずれかの中継デバイスが停止することを判定するための処理を行う。中継デバイスが停止することなく動作しているうちは次の処理へ進むことはない。しかしいずれかの中継デバイスが停止した場合、処理は判断⑩へと進む。

全ての中継デバイスが停止しているかどうかの判定を行う。中継機能の起動と停止を行う際には消費電力が増加する。そのため中継機能を起動した中継デバイスが電池切れで停止した段階で再度どの中継デバイスの中継機能を起動するか決める処理を行うこととする。従ってより少ない切り替え回数でセンサーデータの送信期間をより伸ばすことが可能となる。

いずれかの中継デバイスが動作している場合は、動作している残りの中継デバイスの中で再度処理①へ進み、動作している中継デバイスの中から中継機能を起動する中継デバイスを決定する。全ての中継デバイスが停止していると判定された場合、アルゴリズムは終了する。

### ユースケース・シナリオ

本節では前節で先述した提案を用いたユースケースシナリオについて説明する。ユースケースシナリオでは水田におけるIoTデバイスを用いた水質監視システムの構築を想定する。水質監視は水量の測位と水温の計測を行う。本提案では電池にて動作するIoTデバイスの利用を想定している。そのため屋外において電力系統からの電源供給がなされず、インターネットへ接続するためのWi-Fiアクセスポイント(AP)を設置する、ネットワークインフラの構築が出来ない状況を想定する。本ユースケースシナリオの構成図を図9へ示す。

水質監視は中継デバイスと末端デバイスにて行われる。中継デバイスは直接APへと接続できる。末端デバイスはAPへ直接接続することができないため、中継デバイスを介してセンサーデータをサーバへ送信している。サーバへ送信されたセンサーデータはWebを介してユーザが閲覧可能な状態となる。

本ユースケースにおいて、提案手法を用いる事でユーザである水田管理者はより容易に水田の水質管理を行うこと

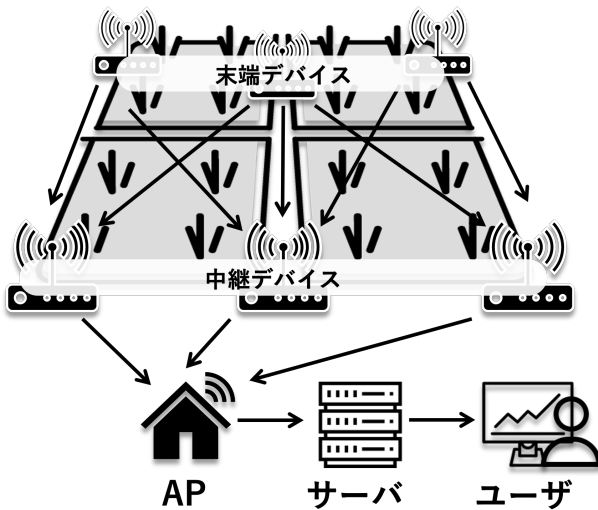


図 9 水田の水質監視におけるユースケース

ができる。水位管理は気温や湿度、時期によって変化をさせる稲作を行う上で重要な作業である。水位の確認を行うために、水田管理者（農家）は 30 分から 1 時間間隔でその水田まで向かい直接視認するしかない。しかしこの作業はこまめに自宅と水田を往復する必要がある、肉体的にも時間的にも負担となる。この課題を解決するための従来の手法では IoT デバイスに電池を搭載し、メッシュネットワークにてデータを送信できたとしても IoT デバイスの動作時間が短くなり、そのメンテナンスに大きな能力を取られてしまう。既存、提案手法ともに IoT デバイスを電力供給インフラやネットワークインフラをコストをかけて整える必要なく水田の水質監視を行うことが可能となるが、提案手法では IoT デバイスの動作時間を既存手法と比較し延長するため電池交換のメンテナンス回数を減らすことが可能である。

#### 4. 実装と実験方法

本章では提案を実装した環境や各種設定やソフトウェア構成図について述べる。

##### 実装

本節では提案の実装手法について述べる。実装はサーバと IoT デバイスである ESP32 を用いる。またサーバと ESP32 はネットワークにて接続されている。図 10 ではサーバと ESP32 の接続における実装構成図を示している。また中継デバイスと末端デバイスは ESP32 を用いる。

図 10 における ESP32 は中間デバイスと末端デバイスともに充電可能な二次電池に接続されており、電力供給を受けて動作している。また温度測定のための温度センサーである BMP280 を搭載している。

図 10 における ESP32 のデータ送受信は ESP32 からサーバの通信に必要な処理を行っている。また ESP32 とサーバ

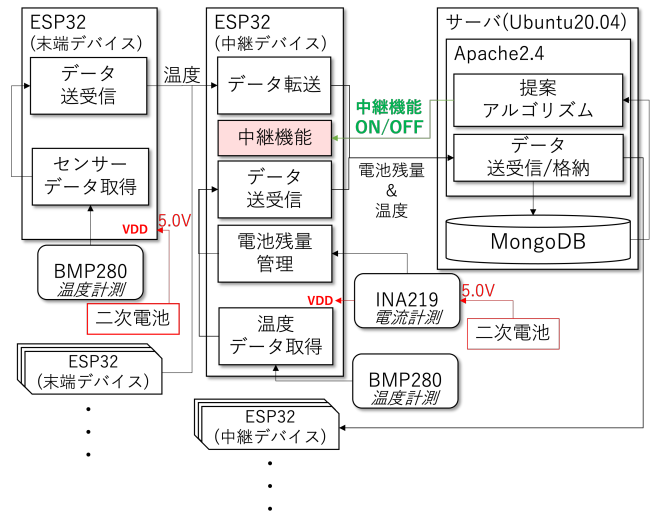


図 10 ESP32(中間デバイスと末端デバイス)とサーバの接続における実装構成図

との通信ではソケット通信を用いる。また実装では ESP32 に電池を取り付けず、実際の電池残量の計測を実施しない。そのため電池残量を予め定め、INA219 にて計測した消費電流を電池残量から減算していき、電池残量の推移を求める。この処理を行うのが電池残量管理である。またデータ取得は温度センサーである BMP280 と電流計センサーである INA219 からそれぞれデータを取得するための処理を行う。

また図 10 におけるサーバは Web サーバとして Apache2.4 がインストールされている。またデータベースには MongoDB を用いる。データ送受信/格納は ESP32 から送信されたデータをデータベースへ格納する。もしくは電池残量を比較し ESP32 へ中継機能を起動または停止要求データを送信する処理を行う。電池残量比較は ESP32 から送信される電池残量のデータの比較を行う。

##### 実験環境

表 1 は実装におけるハードウェアとソフトウェアを示している。

IoT デバイス	ESP32
ファームウェア	MicroPython1.17
電力計センサー	INA219
温度センサー	BMP280
サーバ	Ubuntu20.04
Web サーバ	Apache2.4
データベース	MongoDB 4.2.11

表 1 実装におけるハードウェアとソフトウェアの詳細

図 11 に実験構成図を示す。実験では ESP32 を 9 台とする。中継デバイスの名称を ESP32(1) (6) とし、末端デバイスの名称を ESP32(A) (D) とする。末端デバイスには温度センサーである BMP280 が取り付けられており、10 分間隔で中継デバイスと Wi-Fi AP を介してサーバへ送信される。

実験における中継デバイスのグルーピングはサーバ側に

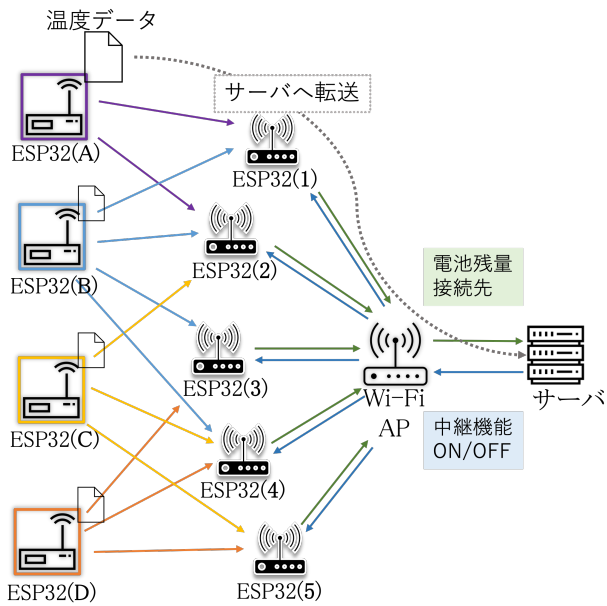


図 11 実験構成図

て疑似的に設定をする。また電池残量の計測には実際に電池残量を搭載するのではなく、ESP32 側で予め消費電流を測定し、その測定値を電池残量から減算していくものとする。電池容量は 1,000[mAh] である。

## 5. 評価と分析

評価では既存手法と提案手法を比較し消費電流とセンサーデータの受信期間を計測した。既存手法として全ての中継デバイスの電池残量になるべく均等となるように中継機能の起動と停止を行う。これを実現するため 30 分毎に中継機能を起動もしくは停止する中継デバイスを決定する。

図 12 と図 13 を用いて説明する。なお図 12 と図 13 における ESP32(1)～(5) と ESP32(A)～(D) は図 11 の ESP32 と同義である。

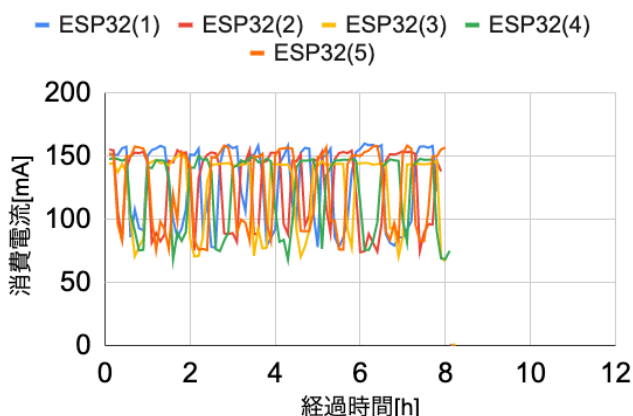


図 12 既存手法における各中継デバイスの消費電流の比較

図 12 では全ての中継デバイスの消費電流が高い頻度で増減を繰り返しており、平均の消費電流は 120[mA] から 150[mA] の間で増減をしている。これは電池残量を均等に保つために中継機能の起動と停止を高い頻度で繰り返し実行しているからである。

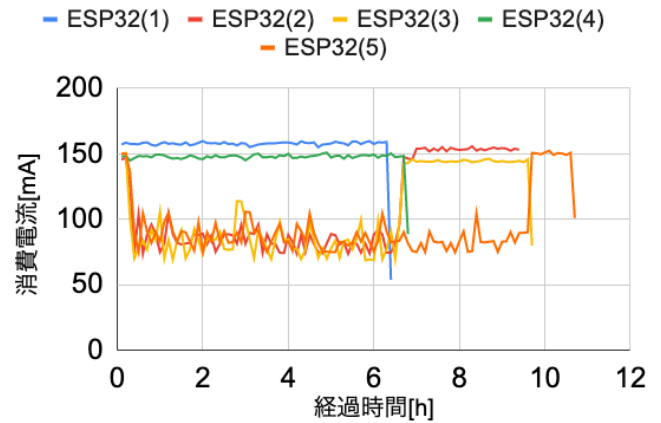


図 13 提案手法における各中継デバイスの消費電流の比較

図 13 では消費電流が 150[mA] を超える中継デバイスと、100[mA] を下回る中継デバイスで分かれている。平均の消費電流は 0 時間目から 6 時間目まで 110[mA] から 130[mA] の間で増減しており、9.5 時間目までは約 130[mA] となり、10 時間目あたりでは約 150[mA] となっている。

このように既存手法と提案手法で消費電流に差があり、動作時間が異なっている。より高い頻度で中継機能の起動と停止を繰り返した場合、切り替えの際にそうでない場合より高い消費電力が必要となる。そのため既存手法ではその消費電力が高くなることで動作時間が提案手法と比較し短くなる。

図 14 へ既存手法と提案手法における中継デバイスの動作時間を比較した結果を示す。図 15 へ既存手法と提案手法における末端デバイスのセンサーデータをサーバが受信した期間を比較した結果を示す。

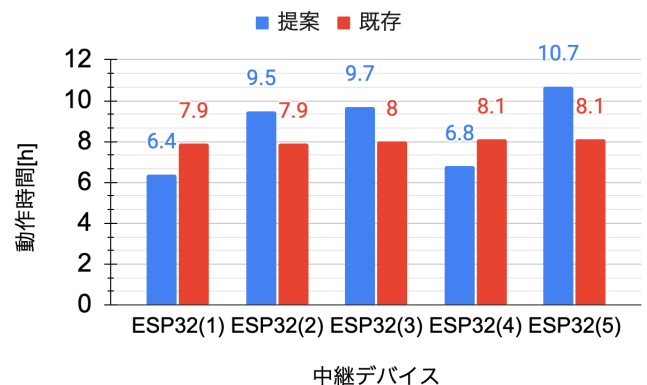


図 14 既存手法と提案手法における中継デバイスの動作時間の比較

図 14 では中継デバイスの動作時間が既存手法が最小で 7.9 時間、最高で 8.1 時間であるのに対して提案手法では最小で 6.4 時間、最高で 10.7 時間となっている。それぞれの中継デバイスの動作時間に大きなばらつきは既存手法では動作時間の最大と最小の差が 0.2 時間であるのに対して、提案手法では動作時間の最大と最小の差が 4.3 時間となっている。

全ての末端デバイスのセンサーデータが受信された時間は図 15 から、既存手法では最大で 8.0 時間となり、提案手

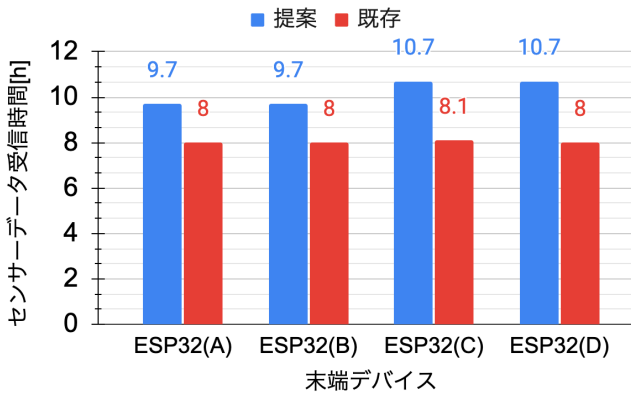


図 15 既存手法と提案手法における末端デバイスのセンサーデータをサーバが受信した期間の比較  
法では最大で 9.7 時間となった。また最低でも 1 台の末端デバイスのセンサーデータが受信された時間は既存手法では最大で 8.1 時間となり、提案手法では最大で 10.7 時間となった。このことから全ての末端デバイスのセンサーデータが受信された時間は既存手法に比べ提案手法が約 21% 延長された。また最後の 1 台の末端デバイスからセンサーデータが受信された時間は既存手法に比べ提案手法が約 32% 延長された。

図 15 から提案手法は既存手法に比べ、全ての末端デバイスのセンサーデータが受信した時間を約 21% 延長したことに加え、最後の 1 台の末端デバイスが停止するまでの時間も約 32% 延長することができた。

## 6. 議論

評価実験ではセンサーデータの送受信時以外に ESP32 を通常通りに動作させている。しかしセンサーデータの送受信時以外において計算処理や通信処理は実施されない。結果として処理を行っていないのにも関わらず消費電力が高い状態が続いている。より消費電力を下げ動作時間を伸ばすためにセンサーデータの送受信時以外はスリープモードに切り替える。

前提条件としてホップ数を最大で 2 ホップとしているため、2 ホップを超えたより長い距離の通信を行うことができない。3 ホップ以上となる場合、現状の提案手法では中継機能を起動する中継デバイスを決定することができない。これはグルーピングを末端デバイスの接続可能な中継デバイス群としているためである。3 ホップ以上では中継デバイスから中継デバイスへセンサーデータが転送されることとなるため、生成するグループを中継デバイスが接続可能な中継デバイス群と追加し中継機能を起動する中継デバイスの台数を削減する。また中継デバイスの重み付けを中継デバイスがグループに属している数ではなく、その中継デバイスの中継機能を起動することでセンサーデータを中継することができるデバイスの台数とする。従ってより少ない台数でより多くのセンサーデータを転送することが

可能となる。

## 7. おわりに

電池で動作する IoT デバイスにてメッシュネットワークを構築する際に消費電力を削減する手法を提案した。課題ではセンサーデータの中継時に、停止可能な中継機能を起動していることとした。これを中継デバイスの中継機能を電池残量と末端デバイスの接続状況に応じて動的な ON/OFF による省電力化を提案した。評価結果ではセンサーデータの受信期間が既存手法では 8.0 時間、提案手法が 9.7 時間となり最大で 21% 伸びた。

謝辞 本研究は、JSPS 科研費 JP20K11776 の助成を受けたものです。

## 参考文献

- [1] Sethy, P. K., Behera, S. K., Kannan, N., Narayanan, S. and Pandey, C.: Smart paddy field monitoring system using deep learning and IoT, *Concurrent Engineering*, Vol. 29, No. 1, pp. 16–24 (online), DOI: 10.1177/1063293X21988944 (2021).
- [2] Bellavista, P., Cardone, G., Corradi, A. and Foschini, L.: Convergence of MANET and WSN in IoT Urban Scenarios, *IEEE Sensors Journal*, Vol. 13, No. 10, pp. 3558–3567 (2013).
- [3] Veloo, K., Kojima, H., Takata, S., Nakamura, M. and Nakajo, H.: Interactive Cultivation System for the Future IoT-Based Agriculture, *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pp. 298–304 (online), DOI: 10.1109/CANDARW.2019.00059 (2019).
- [4] Dagar, R., Som, S. and Khatri, S. K.: Smart Farming – IoT in Agriculture, *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1052–1056 (online), DOI: 10.1109/ICIRCA.2018.8597264 (2018).
- [5] Anurag, D., Roy, S. and Bandyopadhyay, S.: Agrosense: Precision agriculture using sensor-based wireless mesh networks, *2008 First ITU-T Kaleidoscope Academic Conference - Innovations in NGN: Future Network and Services*, pp. 383–388 (online), DOI: 10.1109/K-INGN.2008.4542291 (2008).
- [6] Jiang, X., Zhang, H., Barsallo Yi, E. A., Raghunathan, N., Mousoulis, C., Chaterji, S., Peroulis, D., Shakouri, A. and Bagchi, S.: Hybrid Low-Power Wide-Area Mesh Network for IoT Applications, *IEEE Internet of Things Journal*, Vol. 8, No. 2, pp. 901–915 (2021).
- [7] Nair, K., Kulkarni, J., Warde, M., Dave, Z., Rawalgaonkar, V., Gore, G. and Joshi, J.: Optimizing power consumption in iot based wireless sensor networks using Bluetooth Low Energy, *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 589–593 (2015).
- [8] Sahitya, G., Balaji, N. and Naidu, C.: Wireless sensor network for smart agriculture, *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 488–493 (online), DOI: 10.1109/ICATCCCT.2016.7912049 (2016).