

マルチホップネットワークにおけるリンクごとの送受信の同期によるIoTデバイスの省電力化

河竹 純一¹ 串田 高幸¹

概要：マルチホップネットワークではIoTデバイスによってサーバまでのホップ数が異なる。課題はホップ数の違いにより中継するIoTデバイスがネットワーク機能を停止できないことから消費電力を削減できないことである。提案はセンサデータを送信するタイミングをリンクごとにサーバで算出するスケジューリング手法である。IoTデバイスのサーバまでのホップ数、プログラム実行時間、通信時間の3つの値を使い、1ホップあたりにかかる時間から送信タイミングをサーバで算出する。送信タイミングはプログラム実行時間と通信時間を末端のIoTデバイスからのホップ数の分、足し合わせた時間とする。提案方式により送受信のタイミングを一致させてネットワーク機能の停止が可能な時間を増やす。実験として、ネットワーク機能を通信終了時に随時停止して5秒間のDeep Sleepを入れる場合と、Deep Sleepを入れない場合のESP32の消費電力を1分間計測した。実験の結果、前者の消費電力は2.65[mWh]であり、後者の消費電力は3.62[mWh]であった。ネットワーク機能を停止することによって消費電力を約27%削減できることがわかった。

1. はじめに

背景

IoTデバイスがセンサから取得したデータ（以下、センサデータ）を無線通信を介してサーバに送信するときの手法の一つにマルチホップネットワークがある。マルチホップネットワークはネットワークゲートウェイに対してセンサデータを直接送信することができないときに複数のIoTデバイスを経由して送信する手法である。マルチホップネットワークを用いたユースケースのひとつにスマートパーキングアプリケーション（以下、スマートパーキング）がある [1]。スマートパーキングでは、駐車場における車両の入出庫を距離センサを用いて検出する。それらのセンサデータをサーバに集約し、管理することによってユーザが空き状況の確認や駐車予約が可能になる。スマートパーキングにおいてマルチホップネットワークはセンサデータをサーバに集約するために使用される。

マルチホップネットワークを使用したスマートパーキングにおいて、IoTデバイスは2つの役割が存在する。1つ目の役割は距離センサのデータを取得してサーバに向けて送信する役割である。2つ目の役割は近傍のIoTデバイスから送信されたセンサデータを中継する役割である。距離

センサのデータを取得し、近傍のセンサデータを中継する場合は1つ目の役割と2つ目の役割の両方を持つ。例えば全てのIoTデバイスに距離センサが取り付けられている場合、末端のIoTデバイスが1つ目の役割を持ち、それ以外のIoTデバイスは両方の役割を持つ。両方の役割を持つ場合、近傍のIoTデバイスのセンサデータに距離センサから取得したセンサデータを結合して送信する。

スマートパーキングにおいて屋外の駐車場に設置されるIoTデバイスはバッテリーで駆動している [2-4]。バッテリーで得られる電力には限りがあるため、IoTデバイスの消費電力を削減する研究がなされている [5]。消費電力を削減することで得られるメリットは長時間稼働することやバッテリーの交換頻度が削減されることである [6, 7]。IoTデバイスが最も電力を消費するのはネットワーク機能を起動するときであり、起動しない場合に比べて約2.5倍の電力を消費する。したがってネットワーク機能を起動する時間を短縮することで消費電力の削減につながる。

課題

課題はサーバまでのホップ数がIoTデバイスごとに異なることにより中継するIoTデバイスがネットワーク機能を停止できないことである。マルチホップネットワークではIoTデバイスからネットワークゲートウェイまでの物理的な距離に応じてサーバまでのホップ数が異なる。そのた

¹ 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

め周期的にセンサデータを送信するスマートパーキングにおいてセンサデータを取得した直後にサーバに送信する場合、中継する IoT デバイスがセンサデータを受信するタイミングが IoT デバイスごとに異なる。受信するタイミングが異なる場合、中継する IoT デバイスがネットワーク機能を停止することができないため、消費電力を削減することができない。

各章の概要

第 2 章 関連研究ではマルチホップネットワークの省電力化に関する研究について説明する。第 3 章 提案では課題に対する提案手法とそのユースケースについて説明する。第 4 章 実装では開発したソフトウェアの構成について述べる。第 5 章 実験では提案手法を評価する実験の方法と結果の分析方法について述べる。第 6 章 議論では提案手法について議論すべき内容について説明する。第 7 章 おわりにでは本稿の課題、提案、評価について簡潔に述べる。

2. 関連研究

マルチホップネットワークにおけるデータ集約スキームである QADA が提案されている [8]。この研究は、データパケットを一部のノードに集約することによって省電力化やトラフィック負荷を軽減している。ただし、IoT デバイスにおける送受信の同期に関しては言及されていない。

BLE (Bluetooth Low Energy) を使用したネットワークとしてスター型ネットワークとメッシュ型ネットワークを複合したハイブリッド型のマルチホップネットワークが提案されている [9]。この研究では従来の Zigbee を使用したマルチホップネットワークよりも大幅に電力を削減している。一方で Bluetooth を使用する場合には定期的なアドバタイズ及びスキャンの処理が必要なため、通信回数が増加するという課題がある。

省電力なマルチホップネットワークを実現するための MAC プロトコルである WiseMAC が提案されている。WiseMAC では送受信の同期を取っており、Zigbee を使用したマルチホップネットワークと比較して消費電力を 85% 削減している。ただし、この研究ではダウンリンクの通信のみに焦点が当てられている。

無線センサネットワークのアルゴリズムにおいて一般的に知られている LEACH を改善した E-LEACH が提案されている。E-LEACH はセンサノードのバッテリー残量を参照することでネットワーク負荷のバランスを取っており、クラスタの規模に応じてクラスタヘッダの役割を担う時間を動的に変更している。ただし、この研究ではシミュレーションによる実験のみに留まっており、ユースケースに即した環境での電力測定は行われていない。

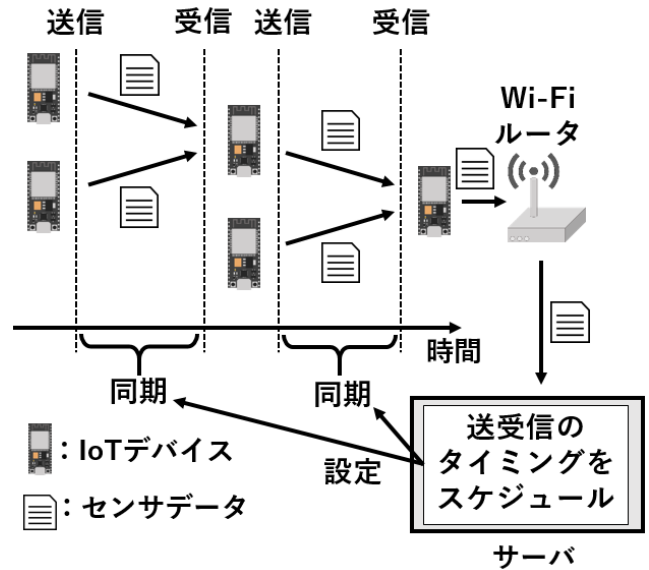


図 1 提案の概要図

3. 提案

提案の目的は中継する IoT デバイスのネットワーク機能の起動時間を短縮し、中継する IoT デバイスの消費電力を削減することである。提案方式ではネットワーク機能の起動時間を減らすためにマルチホップネットワークのリンクごとに送信と受信のタイミングを設定し、送受信の同期を図る。リンクごとに送信と受信のタイミングを設定するにあたって、ネットワークの経路に変更があると設定したタイミングで同期することができなくなる。そこで提案方式の前提条件として以下の 3 つを設定する。

- 設置する IoT デバイスは位置が固定されており、追加、撤去がされない。
- 構築するマルチホップネットワークにおいて IoT デバイスが孤立していない。
- ネットワークゲートウェイは一つとする。

提案方式

図 1 に提案の概要図を示す。図 1 は IoT デバイスがマルチホップネットワークを介してセンサデータをサーバに送信する状況を表している。IoT デバイスはセンサデータを取得しサーバに向けて送信する役割と近傍の IoT デバイスから送信されたセンサデータを中継する役割を持つ。提案方式は IoT デバイスがセンサデータを送信するタイミングをサーバで算出する。サーバで算出したタイミングでセンサデータの送信と受信の時刻を同期させることによって、IoT デバイスのネットワーク機能の起動時間を削減する。

提案手法の手順とその目的について詳しく述べる。提案手法の手順は以下の 5 つである。 t_s 、 t_c はそれぞれプログラム実行時間と通信処理時間を表す。

- (1) 経路の確立 (IoT デバイスがどの IoT デバイスを中継してサーバにセンサデータを送信するかを決める.)
- (2) t_s , t_c の取得 (センサデータの送信の前段階として内部処理時間と通信処理時間を計測し, サーバに送信する.)
- (3) 送信タイミングの決定 (IoT デバイスの送信タイミングをサーバで算出する. 送信タイミングはプログラム実行時間と通信時間を末端からのホップ数の分, 足し合わせた時間とする.)
- (4) 送信タイミングの取得 (サーバで算出した送信タイミングを IoT デバイスが問い合わせる.)
- (5) ネットワーク機能を起動する時間の設定 (送信タイミングと時刻の誤差をもとに, IoT デバイスでネットワーク機能を起動する時間を設定する.)

(1) 経路の確立の目的は, センサデータを送信する際の中継先を IoT デバイスが把握することである. まず, IoT デバイスが通信可能範囲内の他の IoT デバイスに対して経路制御パケットを送信する. 経路制御パケットはセンサデータを取得した IoT デバイスの MAC アドレス, 中継した IoT デバイスの MAC アドレス, およびホップ数を表すホップカウントを含んでいる. サーバが経路制御パケットを受信した際には, 最も小さいホップカウントの値をその IoT デバイスのホップ数として登録し, 中継した IoT デバイスの MAC アドレスから経路表を作成して IoT デバイスに応答として返す.

(2) t_s , t_c の取得の目的は, サーバが送信タイミングを算出するために t_s と t_c を把握することである. t_s と t_c はそれぞれ 1000 回計測した後, 最頻値を算出する. 計測が終了した後, 各 IoT デバイスが t_s と t_c の最頻値をサーバに送信する.

(3) 送信タイミングの決定の目的は, サーバが送信タイミングを算出し, IoT デバイスから参照できるようにすることである. 送信タイミングは, IoT デバイスが最後にセンサデータを取得してからの時間とする.

(4) 送信タイミングの取得の目的は, 各 IoT デバイスがサーバで算出した送信タイミングに基づいてセンサデータを送信することである. IoT デバイスは送信タイミングをサーバに問い合わせ取得する. 取得した送信タイミングに基づいて, 各 IoT デバイスは最後にセンサデータを取得した時点からタイマーを起動し, 送信タイミングの時間が経過したときにセンサデータを送信する.

(5) ネットワーク機能を起動する時間の設定の目的は, 隣接する IoT デバイスがセンサデータの送信よりも前にネットワーク機能を起動することである. 中継する IoT デバイスは隣接する IoT デバイスの送信タイミングを参照し, 送信される時刻よりも前にネットワーク機能を起動する必要がある. そのため, ネットワーク機能を起動する時刻は送信タイミングよりも「センシング間隔 $\times 3\sigma$ 」だけ早い時刻

アルゴリズム 1 送信タイミングの算出アルゴリズム

Input: N : ホップ数, N_{max} : ネットワークの最大ホップ数, t_s : プログラム実行時間, t_c : 通信時間, $interval$: センシング間隔

Output: $send_time$: 送信タイミング

```
1: function CALC_SEND_TIME( $N, N_{max}, t_s, t_c, interval$ )
2:   Initialize:
3:      $send\_time \leftarrow interval$ 
4:   while  $N_{max} - N \geq 0$  do
5:      $send\_time \leftarrow send\_time + t_s$ 
6:     if  $N_{max} - N \neq 0$  then
7:        $send\_time \leftarrow send\_time + t_c$ 
8:     end if
9:      $N \leftarrow N + 1$ 
10:  end while
11:  return  $send\_time$ 
12: end function
```

と設定する. なお, 3σ は IoT デバイスの時刻の誤差の標準偏差を 3 倍した数値であり, 例えば ESP32 の場合には 10 分間で約 30ppm である [10]. 標準偏差を 3 倍する理由は, 99%の時刻の誤差が平均値- 3σ ~ 平均値+ 3σ の範囲に存在しているためである. また, この数値よりも大きい値にしたとしても使用する電力に対してカバーする誤差範囲の効率性が低下する. センサデータの受信後はネットワーク機能を停止する.

次に提案手法の送信タイミングの算出アルゴリズムをアルゴリズム 1 に示す. アルゴリズムの入力値は, IoT デバイスのサーバまでのホップ数 (N), ネットワークの最大ホップ数 (N_{max}), プログラム実行時間 (t_s), 通信時間 (t_c), センシング間隔 ($interval$) とする. 出力として送信タイミング ($send_time$) を返す.

アルゴリズムの流れを説明する. 変数 $send_time$ を $interval$ で初期化する. ループを開始し, $N_{max} - N \geq 0$ の条件を満たす間, 以下の処理を繰り返す. 条件式の $N_{max} - N$ は末端の IoT デバイスからのホップ数を表す. $send_time$ に t_s を加える. これは中継時のプログラムの実行にかかる時間を考慮している. もし $N_{max} - N \neq 0$ であるなら, $send_time$ に t_c を追加する. これは通信時間を考慮している. 最後に N をインクリメントする. ループが終了したら算出された $send_time$ を返す.

ユースケース・シナリオ

本稿のユースケースはスマートパーキングである. 図 2 にユースケース図を示す. スマートパーキングは駐車場において距離センサを用いて車両の入出庫を検出し, IoT デバイスがセンサデータをサーバに向けて送信する [11-14]. サーバに届いたセンサデータとその送信元から駐車場の空き状況を把握し, ユーザが空き状況を確認したり駐車予約をすることができる.

本稿のユースケースシナリオが適用できるアプリケー

ションとして、株式会社 NTT ドコモが開発した Smart Parking Peasy がある*1。Smart Parking Peasy は、車両の出入庫を検知するスマートパーキングセンサと、センサデータを管理サーバに転送するゲートウェイで構成される。スマートパーキングセンサは内蔵のバッテリーによって動作し、ゲートウェイは AC 電源に接続されている。また、スマートパーキングセンサは定期的にセンサデータを取得する。提案手法を導入することでスマートパーキングのユースケースにおいて消費電力の削減が実現できる。

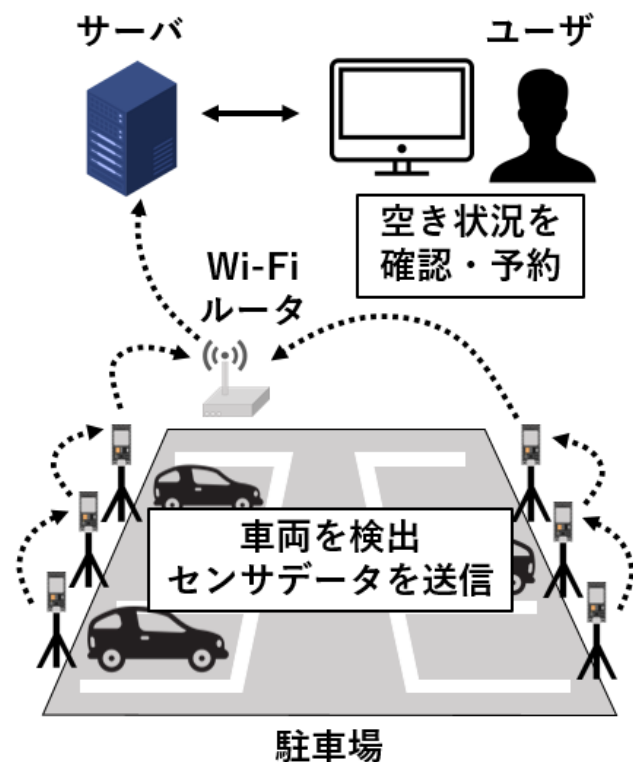


図 2 ユースケース図

4. 実装

実装には ESP32 を IoT デバイスのハードウェアとして採用し、MicroPython を使用してソフトウェアを実行する。サーバ側は Python で API を実装する。ESP32 同士の通信プロトコルとしては ESP-NOW を採用し、サーバとの通信には HTTP を用いる。ソフトウェア構成図を図 3 に示す。IoT デバイスには初期設定プログラム、中継プログラムの 2 つがあり、サーバ側には経路算出プログラムと送信タイミング算出プログラムの 2 つがある。各プログラムの詳細について以下で説明する。

初期設定プログラム

初期設定プログラムの役割は、IoT デバイスの t_s と t_c を測定してサーバに送信し、またサーバから経路表を取得する。

*1 NTT DOCOMO, INC., <https://smartparking.peasy.jp/>
(閲覧日: 2022/12/21)

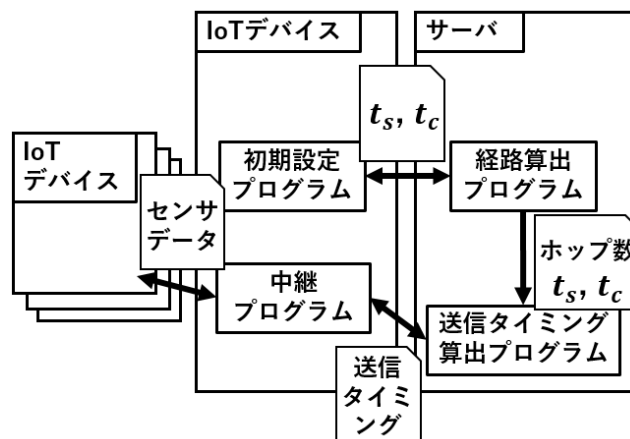


図 3 ソフトウェアの概要図

ることである。IoT デバイスは初めに経路制御パケットを送信し、サーバから経路表を取得する。その後、取得した経路表に基づいて t_s と t_c を測定し、サーバに送信する。

経路算出プログラム

経路算出プログラムは、IoT デバイスから送信された経路制御パケットのペイロードを解析し、センサデータを取得した IoT デバイスの MAC アドレスと中継した IoT デバイスの MAC アドレスのリストから、センサデータを送信するための経路を算出する。経路算出プログラムは REST API を介して、HTTP の GET リクエストと POST リクエストを受け付ける。

中継プログラム

中継プログラムは他の IoT デバイスから送信されたセンサデータを中継してサーバに送信するためのプログラムである。サーバで算出された送信タイミングに基づいて、センサデータの中継を制御する。中継プログラムは初期設定プログラムから送信される経路制御パケットと t_s , t_c の中継も行う。

送信タイミング算出プログラム

送信タイミング算出プログラムは、提案手法に基づいて IoT デバイスの送信タイミングを算出するプログラムである。算出した送信タイミングは API を介して IoT デバイスが取得できるようにする。

5. 実験

実験環境

実験として、ネットワーク機能を通信終了時に随時停止して 5 秒間の Deep Sleep を入れる場合と、Deep Sleep を入れない場合の ESP32 の消費電力を計測する。実験環境の概要図を図 4 に示す。東京工科大学のクラウド・分散システム研究室で IoT デバイスとして 2 台の ESP32 (IoT デバイス A と IoT デバイス B) を ESP-NOW で無線接続した。送信するセンサデータはタイムスタンプ、送信元の IoT デバイス名、車両を検出するための距離センサの距離

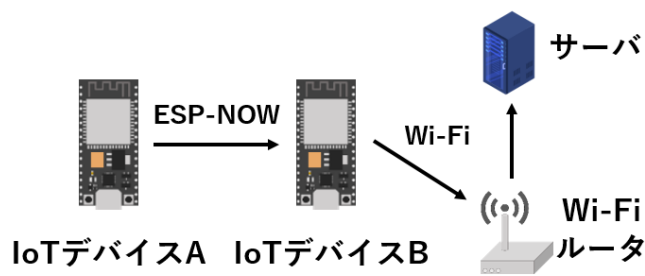


図 4 実験環境の概要図

データに見立てた 32bit の浮動小数点とした。センサデータ 1 個あたりのデータ量は 16bytes である。

実験方法としては、IoT デバイス A が IoT デバイス B と Wi-Fi ルーターを中継して 10 秒に一回の頻度でサーバにセンサデータを 1 個送信し、これを 1 分間繰り返す。このときセンサデータの中継する IoT デバイスにおいて電流値を計測する。電流値は ESP32 の電圧 (3.3[V]) を乗算し、積算電力として表す。

実験結果と分析

実験結果のグラフを図 5 に示す。ネットワーク機能を随時停止して 5 秒間の Deep Sleep を入れる場合を「Deep Sleep あり」、Deep Sleep を入れない場合を「Deep Sleep なし」として経過時間に対する積算電力 (mWh) を表している。Deep Sleep ありのグラフに揺らぎがあるのは、Deep Sleep と起動及び通信を 10 秒ごとに繰り返しているためである。実験結果から、Deep Sleep ありの 1 分間の積算電力は 2.65[mWh] であるのに対し、Deep Sleep なしの 1 分間の積算電力は 3.62[mWh] である。この結果から、1 台の IoT デバイスを経由する場合にネットワーク機能を停止す

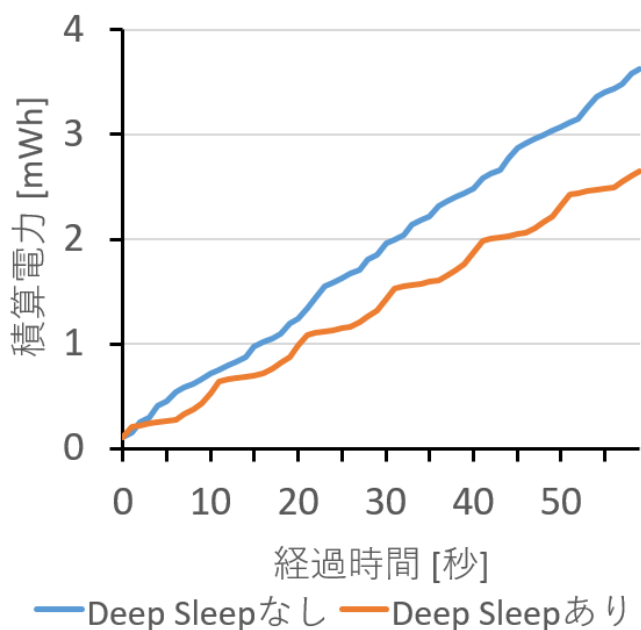


図 5 実験結果

ることによって約 27%削減できることがわかった。

6. 議論

提案手法では、送受信のタイミングを初期設定で決めたのちにそれを固定値として使用していた。これはスマートパーキングのユースケースからネットワーク環境が変動することがないと想定したためである。スマートパーキングでは IoT デバイスは固定されているが、駐車場において人や車の出入りがあると電波が遮られることがあるため、最初に決めた送受信のタイミングが常に使用できるとは限らない。したがって、送受信のタイミングを再度算出する必要がある。再度算出する方法としては、センサデータの送信に一定回数失敗したときにサーバに対してタイミングの更新のリクエストを送る方法がある。

7. おわりに

課題はホップ数の違いにより中継する IoT デバイスがネットワーク機能を停止できないことから消費電力を削減できないことである。課題に対して、センサデータを送信するタイミングをリンクごとにサーバで算出するスケジューリング手法を提案した。実験として、ネットワーク機能を通信終了時に随時停止して 5 秒間 Deep Sleep を入れる場合と、Deep Sleep を入れない場合の ESP32 の消費電力を計測した。実験の結果から、1 台の IoT デバイスを経由する場合にネットワーク機能を停止することによって消費電力を約 27%削減できることがわかった。

参考文献

- [1] Chinrungrueng, J., Sunantachaikul, U. and Triamlumlerd, S.: Smart Parking: An Application of Optical Wireless Sensor Network, *2007 International Symposium on Applications and the Internet Workshops*, pp. 66–66 (online), DOI: 10.1109/SAINT-W.2007.98 (2007).
- [2] Radunovic, B. and Le Boudec, J.-Y.: Rate performance objectives of multihop wireless networks, *IEEE transactions on Mobile computing*, Vol. 3, No. 4, pp. 334–349 (2004).
- [3] Cai, Z. and Chen, Q.: Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks, *IEEE Transactions on Wireless Communications*, Vol. 20, No. 3, pp. 1770–1784 (2020).
- [4] Iqbal, S., Qureshi, K. N., Kanwal, N. and Jeon, G.: Collaborative energy efficient zone-based routing protocol for multihop Internet of Things, *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 2, p. e3885 (online), DOI: <https://doi.org/10.1002/ett.3885> (2022).
- [5] Fourati, L., El-Kaffel, S., Mnaouer, A. B. and Touati, F.: Study of Nature Inspired Power-aware Wake-Up Scheduling Mechanisms in WSN, *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 2154–2159 (online), DOI: 10.1109/IWCMC48107.2020.9148433 (2020).
- [6] Jayakumar, H., Lee, K., Lee, W. S., Raha, A., Kim, Y. and Raghunathan, V.: Powering the internet of things,

- Proceedings of the 2014 international symposium on Low power electronics and design*, pp. 375–380 (2014).
- [7] Iannacci, J.: Microsystem based Energy Harvesting (EH-MEMS): Powering pervasivity of the Internet of Things (IoT)—A review with focus on mechanical vibrations, *Journal of King Saud University-Science*, Vol. 31, No. 1, pp. 66–74 (2019).
- [8] Rahman, H., Ahmed, N. and Hussain, M. I.: A hybrid data aggregation scheme for provisioning Quality of Service (QoS) in Internet of Things (IoT), *2016 Cloudification of the Internet of Things (CIoT)*, pp. 1–5 (online), DOI: 10.1109/CIOT.2016.7872917 (2016).
- [9] 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 (online), DOI: 10.1109/ICGCIoT.2015.7380533 (2015).
- [10] 河竹純一, 杉本一彦, 串田高幸: ”無線 LAN のマルチホップネットワークにおける送受信のタイミングの一致による IoT デバイスの省電力化”, *クラウド・分散システム研究室*, Vol. CDSL-TR-109 (2022).
- [11] Benson, J. P., O’Donovan, T., O’Sullivan, P., Roedig, U., Sreenan, C., Barton, J., Murphy, A. and O’Flynn, B.: Car-park management using wireless sensor networks, *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, IEEE, pp. 588–595 (2006).
- [12] Urdiain, L. O., Romero, C. P., Doggen, J., Dams, T. and Van Houtven, P.: Wireless sensor network protocol for smart parking application experimental study on the arduino platform, *2nd International Conference on Ambient Computing, Applications, Services and Technologies*, Citeseer (2012).
- [13] Lin, T. S., Rivano, H. and Le Mouël, F.: Performance comparison of contention-and schedule-based mac protocols in urban parking sensor networks, *Proceedings of the 2014 ACM international workshop on wireless and mobile technologies for smart cities*, pp. 39–48 (2014).
- [14] Wenzhi, C. and Bai, L.: A smart roadside parking navigation system based on sensor networks for ITS, *2006 IET International Conference on Wireless, Mobile and Multimedia Networks*, IET, pp. 1–4 (2006).