

# イベント駆動型プロトコルによる IoTデバイスの効率的制御手法

杉崎 勇介<sup>1,2</sup> 中里 仁<sup>1</sup> 塚田 学<sup>1</sup>

**概要:** 昨今, IoT は急速に進化し, エッジデバイスをネットワークに接続して遠隔操作/遠隔監視やデバイス間通信を実現する手段として注目されている. これに加え, エッジデバイスに AI を搭載するエッジ AI 技術が導入され, 取得したデータをデバイス上でリアルタイムに解析することが可能になった. これらの技術は社会の様々な分野での応用が進んでおり, データ収集・処理の効率を向上させるなど日常生活においてもその影響は拡大している. 一方で, これらの汎用デバイスが直面している主要な課題として, バッテリー寿命が短く環境によっては継続的な電源供給が困難であること, ネットワーク帯域幅・処理能力・ストレージなどのリソースが限られていることが挙げられる. 本研究では, これらの課題を解決するためのイベント駆動型プロトコルを提案する. 本プロトコルの核となる主要な技術として, (1) バッテリー駆動を想定した間欠的なデバイス制御・状態管理 (2) エッジ処理・クラウド処理を適切に使い分ける最適化 (3) 限られた通信帯域を有効活用する転送制御がある. 本稿では, 提案プロトコルの概要やコンセプト, および設計内容について述べる. また, 提案手法の概念実証として, 室内の人数検知を行うシナリオにおいてイベント駆動型プロトコルを適用した IoT デバイスの運用を行った. システム稼働時のデバイスの消費電力や処理時間を計測し, その結果を比較した. これらの実験結果に基づいて提案手法の有効性やユースケースについて議論し, 課題と今後の方針を整理した.

## Efficient Control of IoT Devices with Event-Driven Protocols

YUSUKE SUGIZAKI<sup>1,2</sup> JIN NAKAZATO<sup>1</sup> MANABU TSUKADA<sup>1</sup>

**キーワード:** IoT, エッジコンピューティング, クラウドコンピューティング, リソース管理, デジタルツイン

### 1. はじめに

日本における未来の社会のビジョンとして「Society5.0」が提唱されている [1]. これは, 人工知能 (AI), ロボティクス, ビッグデータ, モノのインターネット (IoT) などの先進的な技術を活用して経済発展と社会課題の解決を両立させることにより豊かな社会を目指す概念である. 特にその中で社会の情報化は重要な要素であり, 近年では現実空間の情報をサイバー空間上で再現構築してシミュレーション・分析を行う「デジタルツイン」や, サイバー空間上で様々なコミュニケーションや経済活動を行う「メタバース」などが注目されている [2], [3].

社会の情報化において, 現実空間のデータを取得してクラウドやメインサーバ等に送信するプロセスが不可欠である. この役割を担い, 現実空間とサイバー空間のインタフェースとなる IoT デバイスの利活用が拡大している [4]. デバイスが直接インターネットに接続されることにより, 遠隔からのデバイス操作や状態監視, あるいはデバイス間での通信が可能となった. また, IoT デバイスに直接 AI を搭載し, 取得したデータをデバイス上でリアルタイムな学習・解析が可能であるエッジ AI も登場した. これらの技術の活用の幅はさらに広がっており, 工業・農業・医療など様々な分野での応用が期待されている [5].

一方で, 現実空間のデータ取得における IoT デバイスの運用には以下のような制限があり, デジタルツインなどの更なる普及・活用に向けてこれらの課題の解決が求められる.

1. IoT デバイスのバッテリー寿命は短く, 環境によって

<sup>1</sup> 東京大学大学院情報理工学系研究科創造情報学専攻  
IST-CI, The University of Tokyo

<sup>2</sup> 責任著者, y-sugizaki6@g.ecc.u-tokyo.ac.jp

は継続的な電源供給が困難である [6], [7].

2. ネットワーク帯域幅・処理能力・ストレージなどが限られている [8].

本研究では、様々な制約のある IoT デバイスを効率的に運用するためのイベント駆動型プロトコルを提案する。本プロトコルでは、バッテリー駆動を想定した間欠的なデバイス制御・状態管理、エッジ処理・クラウド処理を適切に使い分ける最適化、限られた通信帯域を有効活用する転送制御を行う。本プロトコルによってデータを取得する必要があるイベントの発生に基づいた IoT デバイスの制御を行い、電力消費やデバイスでの処理や通信などの最適化を図る。IoT システムの概観とその中における提案手法の位置付けを図 1 に示す。図より、実際の現実空間のデバイスやネットワークなどの状況を観測し、それに基づいてデバイスの状態管理や通信の制御を行うことで、IoT デバイスの運用を効率化することを目指す。

2 章では、IoT 技術に関連する研究について紹介する。3 章では本研究にて提案するイベント駆動型プロトコルの内容を説明し、4 章では実際のユースケースを想定して行った概念実証 (PoC) としての実験について述べる。5 章では提案手法に関する議論や今後の展望を整理し、最後に 6 章にて本論文をまとめる。

## 2. 関連研究

### 2.1 IoT デバイスの効率的運用に関連する研究

リソースに制約のある IoT デバイスの効率的に運用することには需要があり、IoT デバイスのエネルギー効率に関する研究はこれまでも行われてきた [9]。その代表的なものとして、通信プロトコルの特性を活用して IoT デバイスの省電力を実現するアプローチがある。例えば、低消費電力で広範囲通信が可能な LoRaWAN ネットワークにおいて、可変近傍探索や最小コストスパニングツリーなどのアルゴリズムを用いてネットワークの構成を最適化する手法が提案されている [10]。この最適化により、IoT デバイスのエネルギー効率と通信範囲が向上することが確認された。

現在、広く普及している Wi-Fi は、その高いデータ転送速度や信頼性、IP の互換性や既存のインフラを利用できることなどの様々な観点から IoT デバイスのデータ通信においても推奨されてきた。しかし、Wi-Fi の使用はバッテリーの消費が大きく、特に IoT デバイスでの Wi-Fi の使用においては課題であった [6]。Wi-Fi 接続のデバイスのエネルギー消費を削減するための基本的な考え方は、トランシーバを長くスリープ状態で保つことで Wi-Fi のリスニングに費やす時間を短縮することであるが、実際のデータトラフィックは到達の予測が不可能であるため度々長い通信遅延が生じ、IoT システムの要件を満たせないことがあった [11]。このジレンマの解決策として ZigBee ベースの低消費電力無線インターフェースを利用し、データ通信が必要なタイミン

グに合わせて Wi-Fi インタフェースの起動を行う手法が開発された [12]。なお、ZigBee を用いた手法ではバックグラウンドで帯域幅を消費してしまうことからデバイスの多い環境やデータ通信が頻繁に行われる場面では通信速度の低下が生じていたが、文献 [7] で提案された帯域幅を考慮したマルチインタフェーススケジューリング方式では、各デバイスを二つの Wi-Fi 伝送モードを動的に切り替えながら使用することによって通信速度の低下を抑制した。

### 2.2 先進的技術における IoT の位置付け

本節では近年研究が盛んに行われている AI やデジタルツインなどの先進的技術における IoT の位置付けについて整理する。IoT の普及と合わせて AI 技術のニーズも様々なシナリオで高まっているが、この二つの分野を融合し、IoT デバイスに直接 AI を搭載したエッジ AI が注目されている [13]。従来のシステムでは AI を搭載しているのはクラウドやメインサーバであり、IoT デバイスで取得したデータをクラウドなどに送信し、クラウド上で学習や推論を行うのが一般的であった。しかし、多数の IoT デバイスがクラウドにデータを送信する場合、大量のトラフィックが発生して遅延が生じることや、データのプライバシーに関する懸念などが課題として存在する [14]。また、システムや取得するデータの種類の多様化に伴って IoT デバイスの多様化が進んでいるが、クラウドと接続できるデバイスはその一部に限られているといった問題もある。これらの課題の解決策として期待されているのがエッジ AI である。エッジ AI の特長として、外部との通信を必要としないことから遅延やプライバシーの問題が生じないこと、導入する環境や場面に合わせた設定や構築をすることができる拡張性があることが挙げられる [15]。

IoT デバイスと AI は親和性が高く、この二つを組み合わせたシステムが様々な分野で活用されることが期待されている。例えば文献 [16] では、スマートシティにおける AI と IoT の融合について述べられており、IoT によって大量に収集されたデータを AI が解析することによって様々なシステムを最適化することが可能になり、スマートシティの実現に貢献するとされている。また、文献 [8] では、リソースに制約のある IoT デバイスを用いた分散機械学習についてまとめられている。

IoT はデジタルツインにおいても重要な要素の一つという位置付けにある。デジタルツインとは現実空間を模したモデルをサイバー空間上に構築し、現実空間のデータと統合してシミュレーションや解析を行う技術であり、AI・IoT・ビッグデータ・クラウドコンピューティングなどの技術の発展とともに、近年急速に普及・拡大してきた [17]。デジタルツインでは、特にリアルタイムでのデータ解析に焦点が当てられるが、その有効な手段として IoT が注目されている [18]。

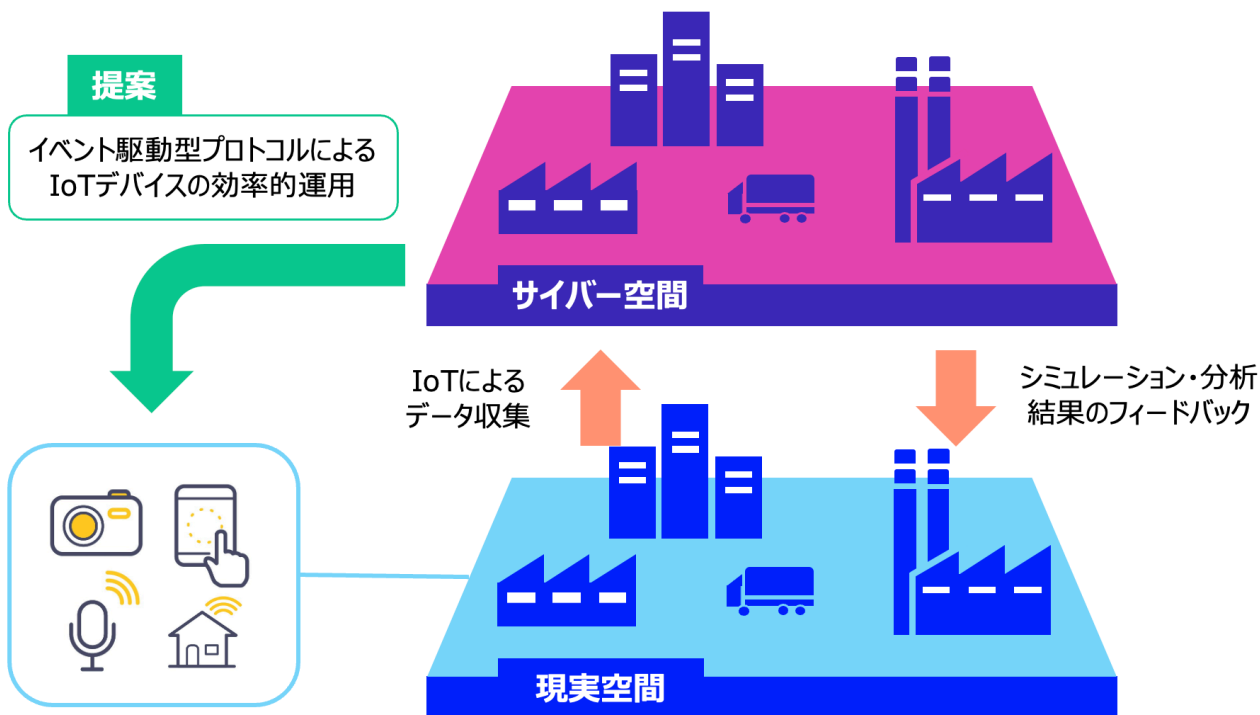


図 1: IoT システムの概観と提案手法の位置付け

### 3. 提案内容

本章では、IoT デバイスを効率的制御を目的とした提案手法の処理やその意図について述べる。なお、ここでいう IoT デバイスとは主に Raspberry Pi や iPhone など安価・汎用的で広範囲のユーザに受け入れられているデバイスを指し、これらのデバイスが持つリソース制約下でいかに効率的にデータ収集や分析を行うかに焦点を当てている。また、具体的な技術的課題として、次の三つを設定している。

1. バッテリー駆動を想定した間欠的なデバイス制御・状態管理
2. エッジ処理とクラウド処理を適切に使い分けるデータ分析処理の最適化
3. 限られた通信帯域を有効活用する転送制御

提案内容の全体像を図 2 に示す。上記の三つの課題について、電力・通信・遅延のトレードオフを考慮しつつ、最適な制御を行う。

以降の各節で課題とその解決のための提案手法を詳述する。

#### 3.1 バッテリー駆動を想定した間欠的なデバイス制御・状態管理

バッテリー駆動の IoT デバイスでは、バッテリーの寿命が短く、継続的な電源供給が困難な屋外や高温といった厳しい環境下での運用ではバッテリーの課題が顕著に現れる。

具体的なシナリオとして、スタジアムでのスポーツイベントや音楽ライブ・フェスイベントなどの大規模な野外活動がある。このような場面において、間欠的なデバイス制御・状態管理によって IoT デバイスのエネルギー消費を減らし、バッテリー寿命を延長する運用を目指す。

従来の多くの IoT システムでは、データ取得を一定間隔で行う定期実行方式が用いられてきたが、この方法では状態変化が少ない場合に不必要なデータ取得とその処理における電力消費を引き起こしていた。そこで本研究では、イベント駆動型プロトコルを導入し、必要に応じてのみデータ取得や処理を行うことでデバイスの消費電力を削減する方法を提案する。

イベント駆動型プロトコルの利用により IoT デバイスは重要なイベントが発生した際にのみ活動し、それ以外では待機状態を維持する。具体的にはスポーツイベントやライブコンサートのような環境でのデータ取得を例として、音量の急激な増加や光の変化などの特定の環境イベントに反応してデータ取得を行うシナリオを考えている。さらに、IoT デバイスのバッテリー消費の要因の一つである Wi-Fi 接続の管理についても改善を行う。デバイスがデータ通信を必要としないときは Wi-Fi をオフにし、データ通信が必要なおきのみ Wi-Fi を再度有効化する制御を行う。

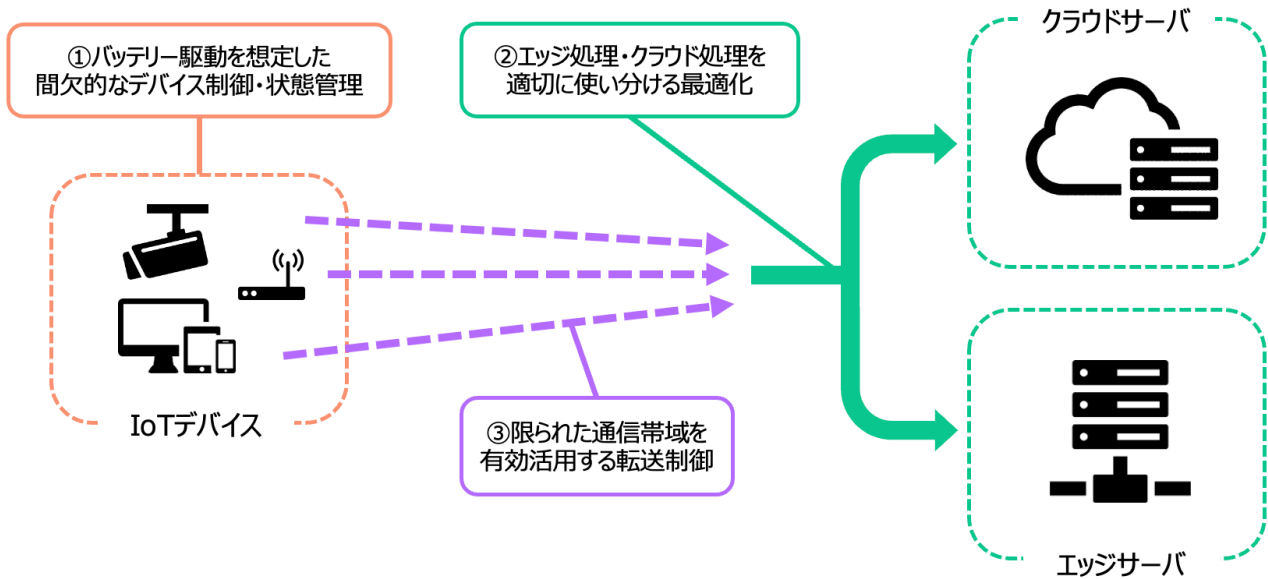


図 2: イベント駆動型プロトコルによる制御

### 3.2 エッジ処理とクラウド処理を適切に使い分けるデータ分析処理の最適化

データ処理技術は、主にエッジ処理とクラウド処理の二つのアプローチに大別される。エッジ処理では毎回のデータ送信が不要であるため、応答時間を短縮しリアルタイム処理が可能となるほか、ネットワーク帯域の使用を減らすという利点がある。その他にも、データ漏えいのリスクを低減することによるプライバシー・セキュリティの向上も実現する。しかし、エッジデバイスには処理能力やストレージなどのリソースが限られているという制約があり、管理・保守が複雑化しやすいというデメリットも持っている。また、バッテリー駆動のデバイスではエネルギー消費の増大も課題の一つである。一方で、クラウド処理は高い計算能力を有しているため大規模データや複雑なアルゴリズムの処理が可能であり、またスケーラビリティと管理の容易さも利点である。ただしクラウド処理ではエッジとの通信における遅延が発生しやすく、帯域幅の制約が問題となることもある。またインターネット接続の依存度が高いほか、データ漏えいなどのプライバシー・セキュリティに関するリスクも含んでいる。こうしたエッジ処理とクラウド処理の特性を踏まえ、IoT デバイスで取得したデータに対して最適な処理を選択し、データ分析の効率を最大化することを目指す。

### 3.3 限られた通信帯域を有効活用する転送制御

IoT デバイスでは多様な種類のデータを扱うが、データはその種類ごとにサイズが異なり、例えばテキストメッセージは小さく、画像や音声、動画データになるとサイズが大幅に大きくなる。IoT デバイスでは、その特性上、限られた通

信帯域でのデータ転送が求められるため、それぞれのデータの性質や転送に必要な通信帯域・速度の違いに対応する必要がある。また、通信はIoT デバイスが接続されるネットワーク環境に強く依存する。

そこで、システムで扱うデータの種類とネットワークの状況を考慮し、通信の遅延を最小限に抑えつつ、システム全体にとって最適なデータ転送を実現することを目指す。本プロトコルでは、データの種類やネットワークに基づいた動的な転送制御を行い、遅延や帯域幅を最適に調整する。例えばテキストメッセージのような小さいデータは即座に送信し、大容量の動画データはネットワークの状況に応じて圧縮や分割を行い、負荷を軽減する。これにより、限られた通信帯域内でIoT デバイスから効率的なデータ収集・分析を可能にする。

## 4. 実験

本章では、本研究で提案するイベント駆動型プロトコルの PoC として行った実験について述べる。

### 4.1 実験設定

#### 4.1.1 シナリオ

本稿では、IoT のユースケースの一つである室内の人数検知を行うシナリオを想定した実験について説明する。カメラで撮影した室内の様子を人数検知モデルに入力し、その結果を管理サーバへと送信するシステムを構築した。システムのアーキテクチャを図 3 に示す。本システムでは、(1) カメラ撮影による室内の画像の取得、(2) 画像の人数検知モデルへの入力、(3) 管理サーバでの出力結果の反映をこの順

番で行う。

#### 4.1.2 実験環境

図 3 に示すシステムについて、東京大学工学部 2 号館 10 階 102A1 室 (図 4) において実際に稼働し、検証を行った。使用した IoT デバイスは Raspberry Pi Model 3B であり、その仕様は表 1 の通りである。

項目	情報
CPU	クアッドコア ARM Cortex-A53, 1.2GHz
メモリ	1GB LPDDR2 RAM
ストレージ	microSD 16GB
OS	Debian GNU/Linux 11 (bullseye)

表 1: Raspberry Pi の仕様

Raspberry Pi では USB ポートを Web カメラと接続して室内の撮影を行うほか、部屋のドアに取り付けたセンサスイッチと GPIO ピンを接続し、ドアの開閉の監視も行う。実験機器の接続の様子を図 5 に示す。人数検知モデルについて、学習済みの YOLOv8 [19] を使用する。管理サーバは IoT デバイスとは異なるネットワークに属する PC (M1 MacBook Pro, 14 インチ, 2021) 上でホストしている。Raspberry Pi は Wi-Fi ネットワークに無線で接続している。

#### 4.1.3 実験内容

図 3 に示すシステムを 30 分間稼働させ、IoT デバイスの電力消費を監視する。また、データ取得・人数検知処理・データ通信を含むシステム全体の処理に要する時間を計測する。このとき、IoT デバイスのデータ (画像) 取得について、次の二つの手法でそれぞれ行う。

手法 1. 1 分毎に室内の様子を撮影する (定期実行)。

手法 2. ドアの開閉を検知した 15 秒後に室内の様子を撮影する (イベント駆動)。

手法 1 では従来の IoT システムを想定した定期実行方式を採用している。これに対して、手法 2 ではドアの開閉というイベントに基づいて IoT デバイスによるデータ取得を行う方式となっている。室内の人数の変化が生じる場合にはドアの開閉が伴うことから、このイベントに基づきデータ取得を行うことで効率化を図っている。

また、クラウド処理・エッジ処理の比較のため、上記の二

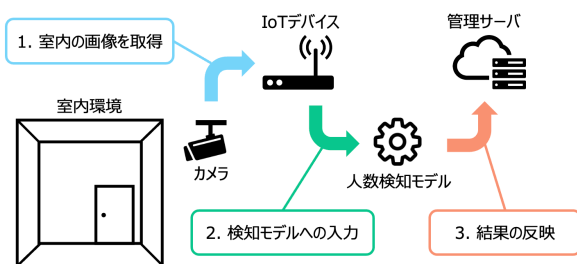


図 3: 人数検知システムのアーキテクチャ



図 4: 東京大学工学部 2 号館 10 階 102A1 室

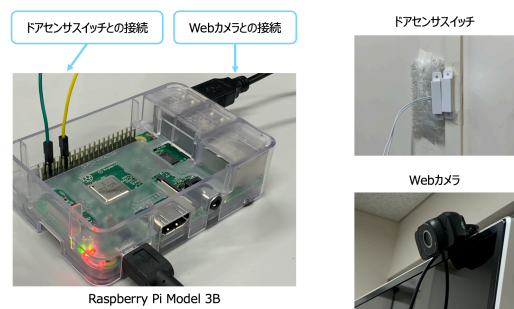


図 5: 実験に使用した機器

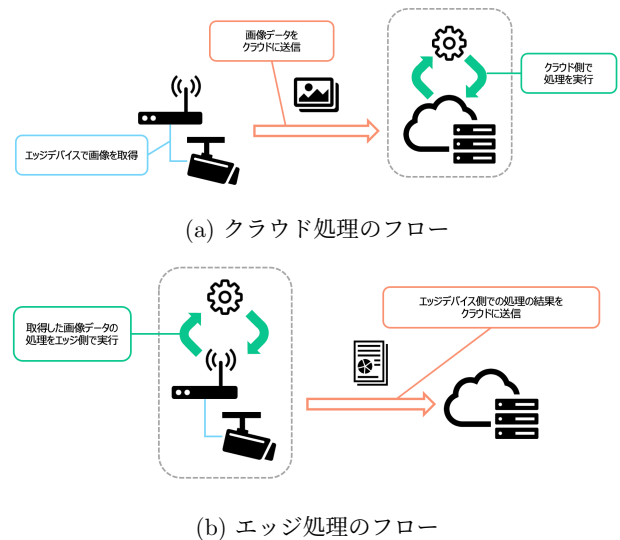
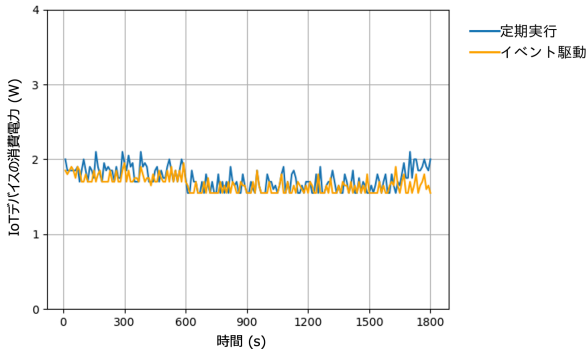
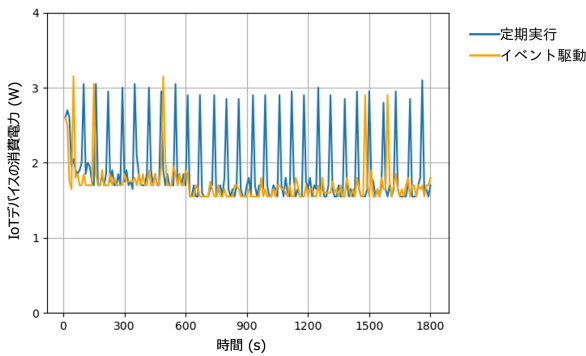


図 6: クラウド処理とエッジ処理のフローの比較

種類の手法のそれぞれにおいてクラウド処理・エッジ処理の二通りのシステムで実験を行った。クラウド処理とエッジ処理のフローを図 6 に示す。図 6a に示すクラウド処理では、IoT デバイスは取得した画像を直接管理サーバに送信し、YOLOv8 による人数検知の処理をクラウド上で行う。一方、図 6b に示すエッジ処理では、画像取得から人数検知までの処理を IoT デバイス上で行い、その出力結果をテキ



(a) クラウド処理システム



(b) エッジ処理システム

図 7: クラウド処理とエッジ処理の消費電力の比較

ストメッセージで管理サーバに送信する。なお、IoT デバイス・管理サーバ間の通信は全て HTTP で行われる。

## 4.2 結果

### 4.2.1 IoT デバイスの電力消費

4.1.3 項の実験について、システムを稼働した 30 分間における IoT デバイスの消費電力を測定した。消費電力の測定結果を図 7 に示す。図 7a は図 6a に示すクラウド処理を行う場合の測定結果、図 7b は図 6b に示すエッジ処理を行う場合の測定結果である。クラウド処理を行う場合では、IoT デバイスは画像を取得して管理サーバに送信する役割を担う。定期実行方式では毎分、イベント駆動方式ではドア開閉検知後に画像の取得及び送信を行うが、図 7a より両手法間の消費電力の差異は見られなかった。一方でエッジ処理を行う場合では、画像取得から人数検知まで IoT デバイス上で行う。図 7b よりエッジ処理が行われるタイミングにおいて消費電力の増加が確認でき、定期実行方式の方がこのスパイクの発生頻度が多く、消費電力も大きいことがわかる。また、イベント駆動方式では 5 回の消費電力の増加が確認できるが、このスパイクの発生時刻はドアの開閉が行われた直後と対応している。なお、本実験では 30 分間にドア開閉が行われる回数はいずれの場合でも 5 回程度であった。

### 4.2.2 クラウド処理・エッジ処理に要する時間の比較

クラウド処理・エッジ処理の両システムにおいて、IoT デ

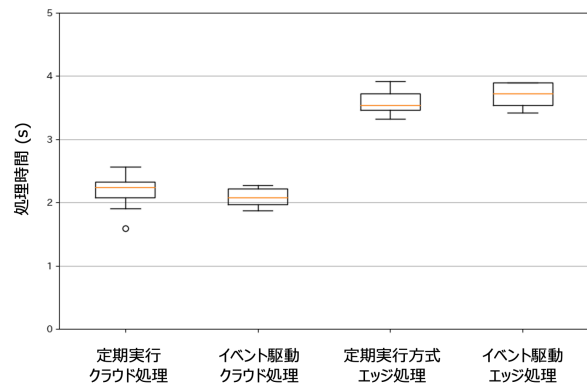


図 8: IoT デバイスの処理時間

種類	平均処理時間 (s)
定期実行 (クラウド処理)	2.206
イベント駆動 (クラウド処理)	2.086
定期実行 (エッジ処理)	3.585
イベント駆動 (エッジ処理)	3.641
管理サーバでの画像受信	2.998
管理サーバでの人数検知	0.191
管理サーバでのテキスト受信	0.464

表 2: 各種処理時間の平均

バイスでの処理にかかる時間を計測した結果を図 8 に示す。ここで、クラウド処理システムでは IoT デバイスでの画像取得処理の開始から画像データの送信完了まで、エッジ処理システムでは IoT デバイスでの画像取得処理の開始から人数検知結果の送信完了までの時間を計測した。なお、図 8 ではひげの長さの上限・下限を四分位範囲の 1.5 倍としており、実際に計測したデータのうちこの範囲にあてまらないものを外れ値としている。また、各方式の平均処理時間、およびクラウド処理システムにおける管理サーバでの画像受信・人数検知処理・テキスト受信に要した時間の平均を表 2 にまとめる。ここで、管理サーバでの画像受信とは IoT デバイスが画像データを送信してから管理サーバで受信するまで、画像サーバでの人数検知とは画像を受信してから YOLOv8 による人数検知処理が終了するまで、管理サーバでのテキスト受信とは IoT デバイスが人数検知結果をテキストメッセージで送信してから管理サーバで受信するまでの時間である。図 8 より、IoT デバイスにおける処理に要する時間はエッジ処理の方が長くなっているが、表 2 より管理サーバでのデータ受信や人数検知処理が終了するまでの時間を考慮すると、システム全体としてはクラウド処理の方が時間を要する結果となった。加えて、定期実行方式とイベント駆動方式を比較した場合、今回のシステムにおける処理時間については差異はなかった。

## 5. 議論

本章では、実験結果をもと提案手法の有効性や応用可能性について議論・考察を行い、課題や今後の方向性につい

て整理する。

## 5.1 実験結果の考察

### 5.1.1 IoT デバイスの消費電力

クラウド処理・エッジ処理の人数検知システムによる実験をもとに、定期実行方式とイベント駆動方式におけるIoTの消費電力について考察する。まずクラウド処理システムについて、IoTデバイスでは画像を取得し、画像データをサーバに送る処理を行う。定期実行方式では上記の処理を毎分行うのに対してイベント駆動方式ではドアの開閉を検知した後に行うため、その実行回数は定期実行方式の方が多くなるが、図7aより消費電力について両者の間に大きな差異は見られなかった。それぞれのグラフ単体に着目しても大きな変化が確認できないことから、クラウド処理システムにおけるIoTデバイスの画像取得・送信処理による消費電力の増加はないということが考えられる。一方でエッジ処理システムでは、特徴的な消費電力の増加が確認できる。エッジ処理システムにおけるIoTデバイスの処理について、取得した画像からYOLOv8による人数検知を行い、その結果をサーバに送信する。図7bの定期実行方式のグラフに注目すると約60秒間隔でスパイクが確認できることから、この消費電力の増加はYOLOv8による人数検知処理に伴って発生していることが考えられる。また、イベント駆動方式のグラフにおけるスパイクの発生時刻もドアの開閉に伴って発生する人数検知処理と対応している。このように、IoTデバイス上で消費電力の増加が発生する処理を行うシステムにおいて、その必要があるイベントの発生頻度が少ない場合では、イベント駆動方式によるデバイス制御によって消費電力を減らすときに有効であると考えられる。

### 5.1.2 処理時間の比較

実験に使用した人数検知システムにおいて、定期実行方式・イベント駆動方式やクラウド処理・エッジ処理について比較し、議論を行う。IoTデバイスでの処理時間について、図8に示すようにエッジ処理の方がクラウド処理より時間が長くなる結果となった。表2のクラウド処理・エッジ処理の平均処理時間の差に注目すると、エッジ処理の方がクラウド処理よりも1.5秒程度長いことがわかる。この差はエッジ上で実行される人数検知処理によるものだと考えられる。また、表2より管理サーバでの画像受信および人数検知処理には約3.2秒かかっているため、全体の処理時間としてはクラウド処理システムの方が長くなると考えられる。管理サーバにおける人数検知処理に要する時間は0.19秒とIoTデバイス上での処理よりも速いことが考えられるが、IoTデバイスから管理サーバに送られるデータについて、クラウド処理システムでは画像、エッジ処理システムではテキストであり、通信時間の差が2.5秒程度存在した。ただし、データの通信にかかる時間については、使用するネットワークの状況にも依存すると予想される。

## 5.2 今後の方向性

### 5.2.1 通信の制御

今回の実験においては検証できなかった通信の制御について、実装方針や課題などを整理する。IoTデバイスと管理サーバのデータ通信について、ネットワークに状況によって通信速度や通信に要する時間は変化することが考えられ、例えばネットワークが混雑している場合では通信に大幅な遅延が生じる可能性がある。またクラウド処理・エッジ処理のトレードオフについて、クラウド処理では強力な計算資源を活用した大規模なデータ解析が可能であり、エッジ処理ではクラウドとの通信回数を最小限に抑えることで通信帯域の有効活用ができるなど、両者の特性はそれぞれ異なる。そこで、ネットワークやIoTデバイスの状態を監視し、クラウド処理・エッジ処理にかかる時間やデータ通信にかかる時間、消費電力などを考慮して通信の制御を行う手法の開発・実装を行うことを考えている。この手法ではネットワークの状況に応じてクラウド処理とエッジ処理を動的に切り替え、また通信時のデータ圧縮や分割・データ送信のタイミングの制御を行う。これにより、デバイスの稼働状況やネットワークの混雑状況に応じたシステムの運用が可能になることが期待される。

### 5.2.2 ユースケースの拡充とシステム基盤の開発

本稿で提案したイベント駆動型プロトコルの適用範囲を拡大するため、様々なユースケースを想定したIoTシステムを構築し、実証実験を実施する計画である。これにより、イベント駆動型プロトコルの有効性を多様なシナリオで検証し、汎用性や適応性を強化できると期待される。本研究では現場のIoTシステムへの導入を目指しており、そのための足がかりとしてユースケースの拡充が求められる。各ユースケースに対応するプロトコルの実装を通じて、実用に適したシステム基盤の開発を目指す。

本プロトコルの実装に関しては、システム全体の中核技術として機能するプロトコルの軸となる要素を開発し、最終的にはIoTデバイスのプロトコルへの参入・離脱、データの取得・処理・送信の一元管理が可能なアプリケーションを構築する。これは、多様なユースケースに対応する包括的なシステム基盤の実現に向けた重要なステップとなる。

開発にあたっては、Raspberry PiやiPhoneなどの汎用的なデバイスを使用し、これらのリソース制約に対応したシステム基盤を設計することで、現場での実用性や応用性を高める。リソースが限られた環境であっても効率的に動作するイベント駆動型プロトコルは、多様な分野でのIoTシステムの実装と運用において、重要な基盤技術となるだろう。

## 6. まとめ

本稿では、IoTデバイスの効率的な制御を目的としたイベント駆動型プロトコルを提案した。さらに、実環境において人数検知システムを構築し、提案手法の概念実証(PoC)と

して実験を行った。実験では、一定時間間隔で動作する定期実行方式とドアの開閉に基づいて動作するイベント駆動方式のそれぞれにおいて、IoT デバイスの消費電力や処理時間を測定・比較した。実験結果からイベント発生頻度が低い環境における提案手法の有効性を確認したとともに、関連するユースケースや課題、今後の方針について議論を進めた。

今後は提案手法のさらなる改良と実装を進め、その他の複数のシナリオやユースケースにおいて本プロトコルが有用であるか評価を行う。また、実際の現場での IoT システム導入を見据え、デバイスのプロトコルへの参入・離脱、データの取得・処理・送信の一元管理が可能なシステム基盤の開発をに取り組む。これにより、IoT デバイスを利用しやすい環境を整備し、様々な産業や日常生活への応用を目指す。最終的には、イベント駆動型プロトコルを中心とした効率的な IoT システムの運用が可能な環境を構築することで、IoT デバイスが多様な分野での発展を促進し、より幅広い応用分野に貢献することを期待する。

## 謝辞

本研究の一部は、JST, CREST, JPMJCR22M4 および科研費 (22H03574) の助成を受けたものである。ここに記して感謝の意を表す。

## 参考文献

- [1] Masaharu AKATSU. Synthesis of knowledge for realizing society 5.0. *Oukan (Journal of Transdisciplinary Federation of Science and Technology)*, Vol. 14, No. 1, pp. 3–6, 2020.
- [2] Mohsen Attaran and Bilge Gokhan Celik. Digital twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*, Vol. 6, p. 100165, 2023.
- [3] Hang Wang, Huansheng Ning, Yujia Lin, Wenxi Wang, Sahraoui Dhelim, Fadi Farha, Jianguo Ding, and Mahmoud Daneshmand. A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges. *IEEE Internet of Things Journal*, Vol. 10, No. 16, pp. 14671–14688, 2023.
- [4] Kwok Tai Chui, Brij B. Gupta, Jiaqi Liu, Varsha Arya, Nadia Nedjah, Ammar Almomani, and Priyanka Chaurasia. A survey of internet of things and cyber-physical systems: Standards, algorithms, applications, security, challenges, and future directions. *Information*, No. 7, 2023.
- [5] S. Narasimha Swamy and Solomon Raju Kota. An empirical study on system level aspects of internet of things (iot). *IEEE Access*, Vol. 8, pp. 188082–188134, 2020.
- [6] Partemie-Marian Mutescu, Adrian Ioan Petrariu, and Alexandru Lavric. Wireless communications for iot: Energy efficiency survey. In *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 1–4, 2021.
- [7] Hao Chen, Hua Qin, Weimin Chen, Ni Li, Tao Wang, Jianxin He, Gelan Yang, and Yang Peng. Bms: Bandwidth-aware multi-interface scheduling for energy-efficient and delay-constrained gateway-to-device communications in iot. *Computer Networks*, Vol. 225, p. 109645, 2023.
- [8] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. A survey on federated learning for resource-constrained iot devices. *IEEE Internet of Things Journal*, Vol. 9, No. 1, pp. 1–24, 2022.
- [9] Guneet Kaur Walia, Mohit Kumar, and Sukhpal Singh Gill. Ai-empowered fog/edge resource management for iot applications: A comprehensive review, research challenges, and future perspectives. *IEEE Communications Surveys Tutorials*, Vol. 26, No. 1, pp. 619–669, 2024.
- [10] Matheus Araujo Gava, Helder Roberto Oliveira Rocha, Menno Jan Faber, Marcelo Eduardo Vieira Segatto, Heinrich Wörtche, and Jair Adriano Lima Silva. Optimizing resources and increasing the coverage of internet-of-things (iot) networks: An approach based on lorawan. *Sensors*, Vol. 23, No. 3, 2023.
- [11] Ke Wang. Migration strategy of cloud collaborative computing for delay-sensitive industrial iot applications in the context of intelligent manufacturing. *Computer Communications*, Vol. 150, pp. 413–420, 2020.
- [12] Hua Qin, Buwen Cao, Jianxin He, Xiang Xiao, Weihong Chen, and Yang Peng. Cross-interface scheduling toward energy-efficient device-to-gateway communications in iot. *IEEE Internet of Things Journal*, Vol. 7, No. 3, pp. 2247–2262, 2020.
- [13] Pethuru Raj, J. Akilandeswari, and M. Marimuthu. Chapter five - the edge ai paradigm: Technologies, platforms and use cases. In Pethuru Raj, Kavita Saini, and Chellammal Surianarayanan, editors, *Edge/Fog Computing Paradigm: The Concept Platforms and Applications*, Vol. 127 of *Advances in Computers*, pp. 139–182. Elsevier, 2022.
- [14] Hessel Stefan and Rebmann Andreas. Regulation of internet-of-things cybersecurity in europe and germany as exemplified by devices for children. *International Cybersecurity Law Review*, Vol. 1, , 2020.
- [15] Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, Vol. 20, No. 9, 2020.
- [16] Md Eshrat E. Alahi, Arsanchai Sukkuea, Fahmida Wazed Tina, Anindya Nag, Wattanapong Kurdthongmee, Korakot Suwannarat, and Subhas Chandra Mukhopadhyay. Integration of iot-enabled technologies and artificial intelligence (ai) for smart city scenario: Recent advancements and future trends. *Sensors*, Vol. 23, No. 11, 2023.
- [17] Maulshree Singh, Evert Fuenmayor, Eoin P. Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. Digital twin: Origin to future. *Applied System Innovation*, Vol. 4, No. 2, 2021.
- [18] Hansong Xu, Jun Wu, Qianqian Pan, Xinpeng Guan, and Mohsen Guizani. A survey on digital twin for industrial internet of things: Applications, technologies and tools. *IEEE Communications Surveys Tutorials*, Vol. 25, No. 4, pp. 2569–2598, 2023.
- [19] A. Chaurasia G. Jocher and J. Qiu. Yolo by ultralytics, 2023. Available online at: <https://github.com/ultralytics/ultralytics>, last accessed on March 21, 2024.