動的に変化するIPアドレスからIoTデバイスを識別する方法

河竹 純一[1,a)]　串田 高幸[1]

概要：IoT デバイスからサーバに送られるセンサデータを識別する場合，通常 IP アドレスを用いて識別を行う．しかし，DHCP サーバによって動的に変更される IP アドレスでは，必ずしも一意な識別ができるとは限らない．また，IP アドレスを使用せずに固有の識別子を使用する場合，ある 1 つのデバイスからセンサデータを送る度に同一の識別子を共に送ることになる．この課題を解決するため，固有の識別子をサーバから IoT デバイスに付与し，IP アドレスと付与した識別子を使用した IoT デバイスの識別方法を提案する．また，IoT デバイスからサーバにセンサデータを送信してから応答までの時間を計測し，固有の識別子をセンサデータと共に送る手法と比較を行った．その結果，提案手法によってセンサデータの送信から応答までの時間を平均して 6.2%削減することができた．

1. はじめに

背景

Internet of Things（IoT）に接続されるデバイス（IoT デバイス）の数は世界規模で年々増加しており，今後 10 年以内に 500 億台を超えると予測されている [1]．2020 年度の総務省の情報通信白書によると，2015 年時点で 165.6 億台，2020 年時点で 280.4 億台であることから，増加傾向が続いており今後も増加すると考えられている [2].

これらの IoT デバイスはセンサーから集約されたデータの可視化及び分析のために一台一台を管理する必要があり，IoT デバイスの数が増えるほど管理が困難になる [3]. IoT デバイスを管理するということは，個々の IoT デバイスの登録・編成・モニタリングを行うということであり，すなわち各 IoT デバイスを一意に識別していることが前提となる [4]．IoT デバイスを識別するにあたっては，ローカル環境で無線 LAN を用いる場合，IP アドレスや MAC アドレスの識別番号が用いられる．　IP アドレスは，通常 DHCP サーバによって動的に割り振られる．DHCP サーバから割り当てられる IP アドレスにはリース期間が設定されており，IoT デバイスはこの期間 IP アドレスを利用できる [5]．IoT デバイスがネットワークに接続されている場合には，リース期間を超える前にデバイスが DHCP サーバに DHCP リクエストを送ることによってリース期間が延長される．したがって，デバイスがリース期間を超えてスリープ及び電源をオフにしていた場合や，ネットワークの電波が及ばない場所に持ち出した場合，再度ネットワークに接続したときに IP アドレスが割り当てられる．

MAC アドレスは，すべてのネットワークデバイスにつけられる識別番号であり，ベンダによって一意に割り振られる．ただし，2014 年に RFC7136 で指摘されているように，近年製造業者が複数のデバイスに対して同じ MAC アドレスを割り当てているという報告が頻発している [6]．加えて IoT デバイスの数が年々増加していることや，ネットワークの知識を持っているエンドユーザが IP アドレスや MAC アドレスの識別番号を変えることが可能となっていることから，必ずしも一意に識別することができない．

課題

上記のことから，IoT デバイスにおいて重複のない識別を行う場合，ハードウェアやネットワークに依存しない識別子を用いる必要がある．この場合識別子が通信パケットのヘッダから読み取れないため，パケットのペイロードとして識別子を含めることになる．よって一部が同じデータを繰り返し送り続けることになることから，オーバーヘッドとして通信量や通信及び処理にかかる時間が識別子とするタグの bit 数の分だけ増加することになる．例えば，ある IoT デバイスでセンサから取得した温度データをサーバに送信するとき，このデータを 4Byte と仮定する．そして識別子として 128bit すなわち 16Byte の UUID を用いるとする．このとき，送信するデータ量は 4Byte+16Byte によ

[1] 東京工科大学コンピュータサイエンス学部
〒 192–0982 東京都八王子市片倉町 1404-1
[a)] C0118083

り 20Byte となり，このうちの 20%が意味のあるデータとなる．したがって，送信データの 80%がオーバーヘッドとなる．

上記の温度データを，UUID と共に JSON 形式で HTTP リクエストとしてサーバに送信するものとする．このとき送信データの総量を D_A，通信プロトコル（IP，TCP，HTTP）の各階層のヘッダのデータ量を D_H，辞書型データの key のデータ量を D_{KEY}，value のデータ量を D_{VALUE}，識別子（UUID）のデータ量を D_{ID}，各種記号（括弧，コロン，カンマ）のデータ量を D_S とすると，D_A はこの総和になることから式 1 で表される．

$$D_A = D_H + D_{KEY} + D_{VALUE} + D_{ID} + D_S \quad (1)$$

また，このデータを N 回送信すると，その総データ量 D_A^N は式 2 で表される．なお，i 回目（$i \in \mathbb{N} \mid 1 \leq i \leq N$）の温度データを D_{VALUE}^i とする．

$$\begin{aligned} D_A^N &= \sum_{i=1}^{N}(D_H + D_{KEY} + D_{VALUE}^i + D_{ID} + D_S) \\ &= N(D_H + D_{KEY} + D_{ID} + D_S) + \sum_{i=1}^{N} D_{VALUE}^i \end{aligned} \quad (2)$$

したがって，D_A^N の送信時間 T_{TX} は式 3 で表される．ただし，ネットワーク速度を V_{NET} とする．

$$T_{TX} = \frac{D_A^N}{V_{NET}} \quad (3)$$

ここで，式 2 より T_{TX} は式 4 で表されることから，識別子を送信する時間は $\frac{ND_{ID}}{V_{NET}}$ となる．

$$\begin{aligned} T_{TX} = &\frac{ND_H}{V_{NET}} + \frac{ND_{KEY}}{V_{NET}} + \frac{ND_{ID}}{V_{NET}} + \frac{ND_S}{V_{NET}} \\ &+ \frac{1}{V_{NET}} \sum_{i=1}^{N} D_{VALUE}^i \end{aligned} \quad (4)$$

ゆえに，T_{TX} に対して $\frac{ND_{ID}}{V_{NET}}$ がオーバーヘッドとしてかかる時間となり，その割合を R_O とすると R_O は式 5 で表される．

$$R_O = \frac{\frac{ND_{ID}}{V_{NET}}}{T_{TX}} = \frac{ND_{ID}}{V_{NET} \cdot T_{TX}} \quad (5)$$

各章の概要

第 2 章 関連研究では，上述した課題に関連する研究とその比較をする．第 3 章 提案では，課題を解決する本研究の提案方法とその方式について詳しく述べる．第 4 章 実装と実験環境では，提案内容の実験についてハードウェアとソフトウェアの構成や実験環境について述べる．第 5 章 評価と分析では，実験の結果を分析して提案内容の評価及び検証をする．第 6 章 議論では，本研究の提案，実験，評価について議論すべき内容について述べる．最後に，第 7 章 おわりにで本稿のまとめと貢献した内容，今後の展望について述べる．

2. 関連研究

この章では，本研究に関連した研究を挙げたうえで本稿の位置づけを述べる．

最初に，IP アドレスから識別する方法を挙げる．IP アドレスはインターネットプロトコル（IP）においてパケットを送受信するためにつけられる識別番号であり，手動で固定しない限りは DHCP サーバによって自動的に設定される [7][8]．自動化するメリットとして，新しいデバイスをネットワークに追加する際に重複が起こらない点や多くの無線機器で標準搭載となっているため IP アドレスの知識がなくても誰でも利用できる点がある．デメリットとしては，ユーザの意図にそぐわず IP アドレスが変わってしまう点である．

次に，MAC アドレスから識別する方法を挙げる．MAC アドレスはすべてのネットワークデバイスにつけられる識別番号であり，ベンダによって一意に割り振られる．したがって理論上ネットワーク上のどのデバイスにおいても MAC アドレスによって一意に識別することができるということになる．しかし，MAC アドレスは変更することも可能である点やプライバシー保護の観点から Wi-Fi ネットワークが変化するたびに MAC アドレスを変えているデバイスも存在することから，厳密には MAC アドレスのみでの識別は必ずしも一意であるとは限らない [9]．

最後に，機械学習によって識別する方法を挙げる．Meidan らは，ネットワークトラフィック分析に基づく IoT デバイスの識別のために機械学習によるアプローチを行った [10]．この研究では，9 つの異なる IoT デバイスと PC，スマートフォンからネットワークトラフィックデータを収集してラベリングし，教師あり学習によって多段階の分類を行った．その結果，モデルの分類精度は 99.281%以上となった．また，Meidan らは別の研究で，セキュリティーの観点から不正な IoT デバイスを識別する際に機械学習によるアプローチを行った [11]．この研究では，信頼できる IoT デバイスのリスト”ホワイトリスト”の IoT デバイスかどうかをネットワークトラフィックデータから得られた特徴からラベリングし，17 の異なるデバイスを 99.49% 以上の精度で識別することができた．Meidan らが行った研究によって，ネットワークトラフィックデータの特徴から機械学習を行うことで異種センサデバイスの識別を高い精度で行うことが可能であることが明らかとなった．しかし，これらの研究は異種センサデバイスの場合に限定しており，同じ構成のデバイスから送信される類似データの識別は行っていない．

3. 提案

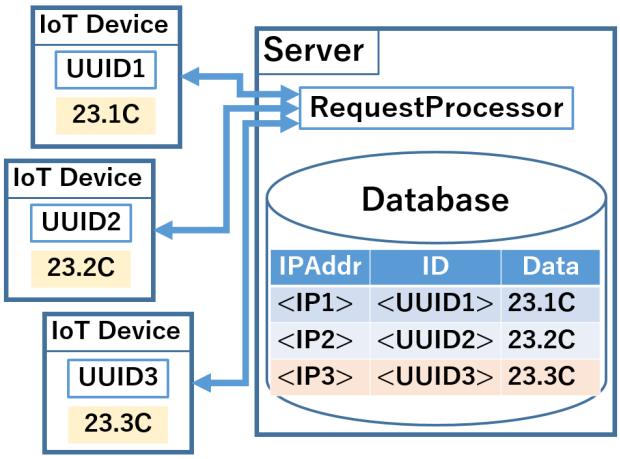

図 1 提案内容のコンセプト図

提案内容のコンセプトを図 1 に示す．図 1 の IoT Device は，温度センサから温度データを取得しており，Server と無線 LAN を経由して通信することができるデバイスとする．Server は，IoT Device から温度データを HTTP リクエスト処理オブジェクト RequestProcessor で受け取り，Database に IP アドレス（IPAddr）及び識別子（ID）と共に格納する．ここで，IoT Device の IP アドレスが Server の Database に格納されていないことを**未知**，格納されていることを**既知**とする．提案は，未知の IoT デバイスに対してサーバから UUID を付与することによって，DHCP サーバが IP アドレスを変更してもデバイスを識別することができるというものである．以下にその手法の詳細を述べる．また，手法全体のフローチャートを図 2 に示す．

まず，Server と初めて通信する IoT Device に ID を付与する方法を説明する．IoT Device は温度センサーから温度データをセンシング（Sensing temperture data）し，そのデータを HTTP リクエストとして送信する（Sending sensor data）．その際，IoT Device で IP アドレスが変わったかどうかを判別する．なお，温度データを送る度にそのときの IP アドレスを IoT Device で保持し，これを更新する．それによって直前に温度データを送った際の IP アドレスが把握できることから，IP アドレスが変わったかどうかを判別できる．判別した結果は，IP アドレスが変わっていない場合を「0」，IP アドレスが変わった場合を「1」としてこれを flag とする．

Server は IoT Device から送信された sensor data を受け取り（Receiving sensor data），Database のレコードを参照して（Referring to the data）IP アドレスが未知か既知かを判別する．Server は，未知の IoT Device から HTTP リクエストがあった場合，その応答（response）として未知であること（”Unknown”）を返す．IoT Device は Server から付与される ID を持っていないため，Server に ID を要求する（Requesting ID）．ID を要求された Server が UUID バージョン 4 を生成し（Generating ID），それを ID として IoT Device に返す．これによって IoT Device に ID が付与される（Assigning ID）．IoT Device は付与された ID を受け取り（Receiving ID），ID をテキストデータとして保存する（Saving ID as a text）．Server は ID を付与したのち，温度データは IP アドレス及び ID と共に Server の Database に格納する（Storing data）．ID が付与された IoT Device は，IP アドレスが変わらない限り ID は送信せず，温度データを HTTP リクエストとして Server に送る．その場合，Server は応答として既知であること（Known）を返すため，その後温度データは Database に格納される（Storing data）．

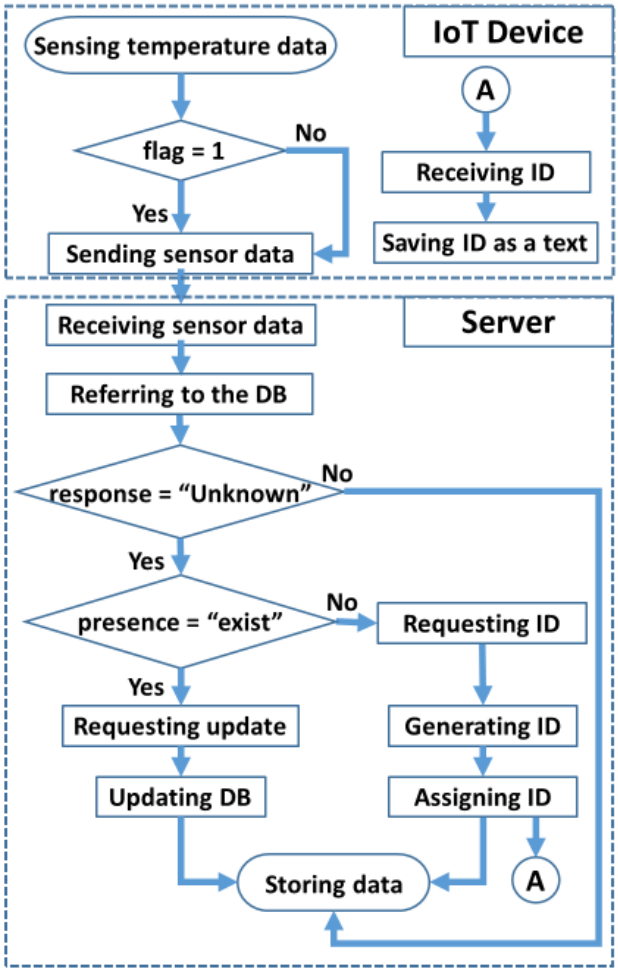

図 2 提案内容のフローチャート図

次に，IoT Device の IP アドレスが変わったときに識別する方法を説明する．IoT Device の IP アドレスが変わるとき，未知の IP アドレスに変わる場合と既知の IP アドレスに変わる場合の 2 つのケースがある．前者の場合，IoT Device が HTTP リクエストを送った場合に Server は応答として”Unknown”を返す．ただし，IoT Device にはすでに Server から付与された ID が存在するため，それを判別

したうえで Database の IP アドレスと ID の結びつきを更新する要求（Requesting update）を Server に対して送る．これを受けた Server は Database を更新し，温度データを IP アドレス及び ID と共に Database に格納する．

最後に，IoT Device の IP アドレスが既知の IP アドレスに変わった場合に識別する方法を説明する．まず Server に温度データを送る前に IoT Device で IP アドレスが変わったかどうかを判別する．Server は，flag の値を参照することによって IP アドレスが変わったことを判別することができるため，HTTP リクエストの応答として未知であること（”Unknown”）を返す．以降の手順は IoT Device の IP アドレスが未知の IP アドレスに変わったときの識別方法と同様になる．

以上の機能を実装することにより，IP アドレスによって識別することができるようになり，ID をセンサデータと共に送る必要がなくなる．したがって，送信するデータ量は削減する．

4. 実装と実験環境

4.1 実装

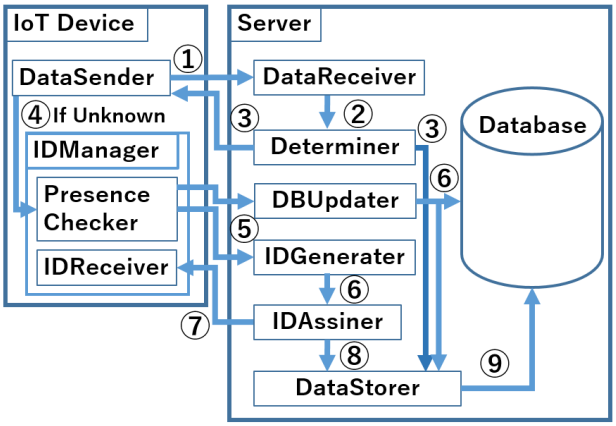

図 3 ソフトウェア構成図

図 3 は作成したソフトウェアの構成を示している．ソフトウェアの挙動を以下に記す．

① IoT Device の DataSender モジュールは，センサから読み取った温度データを Server の DataReceiver モジュールに HTTP リクエストとして送信する．その際，IoT Device の IP アドレスが変わったかどうかを判別する．

② 受け取った温度データを Determiner モジュールに送信元 IP アドレスと共に送る．

③ 受け取った IP アドレスと Database のレコードをもとに未知の IP アドレスと既知の IP アドレスを判別する．既知の場合には IP アドレスが一致するレコードから ID を参照し，DataStorer モジュールに IP アドレスと ID と温度データを渡す．また，HTTP リクエストの応答として DataSender に”Known”を返す．未知の場合には，DataStorer モジュールに IP アドレスと温度データを渡す．HTTP リクエストの応答として DataSender に”UnKnown”を返す．

④ ③の応答が”Unknown”の場合，ID Manager モジュールの PresenceChecker プログラムを起動し，Server から付与された ID が存在するかを確認する．

⑤ IoT Device に ID が存在しない場合には，IDGenerater に ID の生成と付与を要求する．ID が存在する場合には，DBUpdater に Database のレコードの IP アドレスと ID の結びつきを更新することを要求し，IoT Device が保持している ID を送信する．

⑥ IDGenerater では新規の UUID（バージョン 4）を生成して IDAssigner に渡す．DBUpdater では，IP アドレスと IoT Device から送られた ID をもとに Database のレコードを更新する．その後 DataStorer に IP アドレスと ID を渡す．

⑦ IDAssigner モジュールでは，IDGenerater で生成された UUID を ID として IoT Device に付与する．

⑧ IDAssigner モジュールが DataStorer モジュールに IP アドレスと ID を渡す．

⑨ DataStorer は Database に IP アドレスと ID と温度データを格納する．

4.2 実験環境

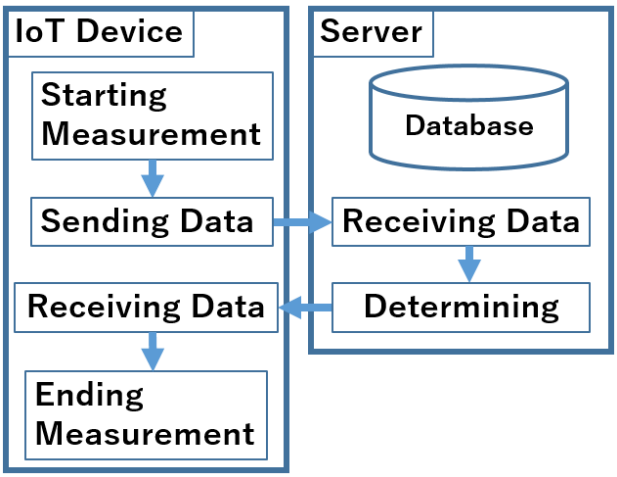

図 4 実験環境の構成図

この節では，実験を行った環境のハードウェアとソフトウェアの構成について説明する．まず，ハードウェアの構成としては，IoT デバイスとして Wi-Fi を内蔵しているマイクロコントローラである Espressif Systems 社の ESP-WROOM-32（以下 ESP32）を用いており，センシングデータを取得するセンサーモジュールとして温度・湿度・気圧が測定できる Bosch Sensortec 社の BME280 を用

いている．したがって，BME280 から温度データを取得し，ESP32 から取得したデータをサーバに送るという流れになっている．なお，Server の OS として Linux ディストリビューションの一つである Ubuntu 18.04 LTS を使用している．また，Database としてドキュメント指向データベースである MongoDB 4.0 を使用している．

次に，実験で使用したソフトウェアの構成を説明する．図 4 に実験環境の構成図を示す．評価実験として，IoT Device が温度データを送信する（Sending Data）前に時間の計測を開始する（Starting Measurement）．その後 Server が温度データを受け取り（Receiving Data），IP アドレスの未知と既知を判別（Determining）した結果を応答として返す．IoT Device が Server からの応答を受け取った（Receiving Data）時点で時間の計測を終了する（Ending Measurement）．また，温度データを常に ID と共に送る手法をタグ付け手法と定義したうえで比較を行い評価する．

5. 評価と分析

評価は，以下のようにして行う．なお，比較対象との差異が誤差の範囲に収まる可能性を考慮して，送信回数を 1 回から 100 回まで 1 ずつ増やしていき，それぞれにおいて掛かった時間を計測する．

(1) 本研究の手法を用いて IP アドレスによる識別を行う．

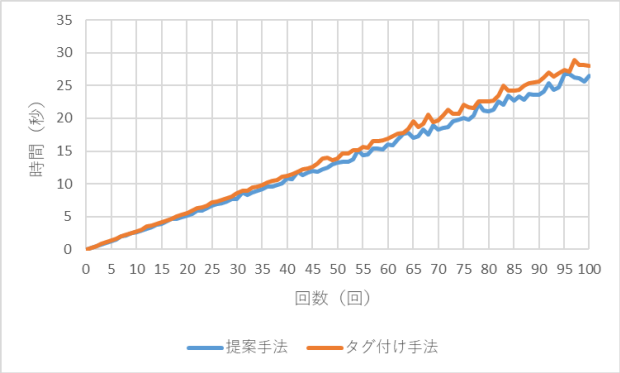

図 5 提案手法とタグ付け手法における送信回数ごとのセンサデータ送信時の通信時間

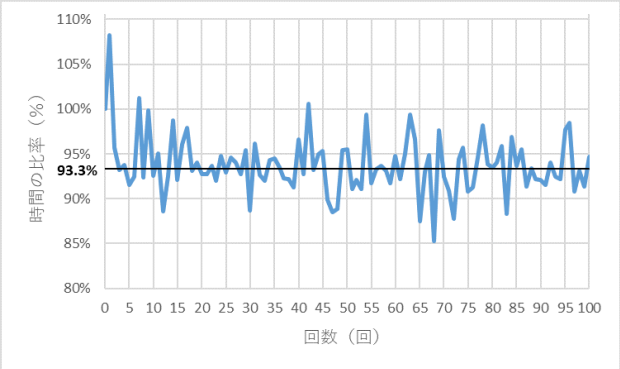

図 6 タグ付け手法に対する提案手法の送信回数ごとのセンサデータ送信時の通信時間の比率

(2) IoT デバイスで，センサデータの送信から返答までの時間を計測する．

(3) (1) の実験で用いた識別子をセンサデータにカンマ区切りでタグ付けして JSON 形式でサーバに送る．

(4) (2) と同様に、(3) の手法（タグ付け手法）によるセンサデータが送信してから返答するまでの時間を計測する．

計測結果を図 5 に示す．図 5 は提案手法とタグ付け手法における送信回数ごとのセンサデータ送信時間を 10 回計測し，その中央値を算出してプロットしたものである．また，図 6 は図 5 におけるタグ付け手法の時間を 100%としたときの提案手法の時間の比率である．図 5 のグラフでは，一回ごとにセンサデータを送信してから返答するまでの時間を計測しているため，回数と時間が線形の関係になっている．また，提案手法にかかる時間がタグ付け手法にかかる時間よりわずかに下回っていることがわかる．図 6 のグラフの結果から，提案手法の時間の比率はおよそ 85% から 108% までの振れ幅があり，中央値を計算した結果，93.3% であることがわかった．また，これを平均して，提案手法がタグ付け手法に比べて 6.2%時間を削減できることがわかった．

6. 議論

本研究では，動的な IP アドレスによる識別の課題である DHCP サーバによる IP アドレスの変更，及び識別子をデータにタグ付けする方法の課題である通信にかかる時間のオーバーヘッドの上昇の課題を解決する手法を提案した．しかし，IP アドレスによる識別を行うことによるデメリットも同時に存在する．

例えば，異なるローカル環境のサーバのセンサデータを一括管理する場合，ある環境の DHCP サーバから付与された IP アドレスが別の環境の DHCP サーバから付与された IP アドレスと競合するという課題がある．したがってこのようなケースでは，本提案で採用した UUID を用いてセンサデータを識別することが可能となる．

7. おわりに

最後に，本研究のまとめを述べる．まず，本研究では主に二つの課題を扱っている．一つ目に，IP アドレスや MAC アドレスといったハードウェアに依存している識別子を用いた際に，変更することが可能であることから一意に識別することが出来ないという課題．二つ目に，送信するデータに識別子をタグ付けする際に，通信時間のオーバーヘッドが識別子の bit 数の分だけ上昇するという課題がある．これらの課題を解決するための手法として，未知の IoT デバイスに対してサーバから ID を付与することによって，DHCP サーバが IP アドレスを変更してもデバイスを識別することが可能になっている．したがって，本研究の手法

を用いることによって通信時間のオーバーヘッドの上昇を抑えたIoTデバイスの識別が可能であるため，それらのデバイスやセンサデータの一括管理に貢献している．

参考文献

[1] Jayakumar, H., Raha, A., Kim, Y., Sutar, S., Lee, W. S. and Raghunathan, V.: Energy-efficient system design for IoT devices, *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 298–301 (online), DOI: 10.1109/ASP-DAC.2016.7428027 (2016).

[2] Ministry of Internal Affairs, M. and Communications: White Paper on Information and Communications in Japan (2020).

[3] Kortesniemi, Y., Lagutin, D., Elo, T. and Fotiou, N.: Improving the privacy of iot with decentralised identifiers (dids), *Journal of Computer Networks and Communications*, Vol. 2019 (2019).

[4] Baranwal, T., Nitika and Pateriya, P. K.: Development of IoT based smart security and monitoring devices for agriculture, *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pp. 597–602 (online), DOI: 10.1109/CONFLUENCE.2016.7508189 (2016).

[5] R.Drmos: RFC2131 - Dynamic Host Configuration Protocol, (online), available from ⟨`https://tools.ietf.org/html/rfc2131`⟩ (1997).

[6] B.Carpenter, S.: RFC7136 - Significance of IPv6 Interface Identifiers, (online), available from ⟨`https://datatracker.ietf.org/doc/html/rfc7136`⟩ (2014).

[7] Mohsin, M. and Prakash, R.: IP address assignment in a mobile ad hoc network, *MILCOM 2002. Proceedings*, Vol. 2, pp. 856–861 vol.2 (online), DOI: 10.1109/MILCOM.2002.1179586 (2002).

[8] Ford, P. S., Bahl, P., Khaki, J. M. J., Burns, G. and Beeson, F. J.: Method and computer program product for automatically generating an internet protocol (IP) address (2000).

[9] Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E. C. and Brown, D.: A study of MAC address randomization in mobile devices and when it fails, *Proceedings on Privacy Enhancing Technologies*, Vol. 2017, No. 4, pp. 365–383 (2017).

[10] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O. and Elovici, Y.: ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis, *Proceedings of the Symposium on Applied Computing*, SAC '17, New York, NY, USA, Association for Computing Machinery, p. 506–509 (online), DOI: 10.1145/3019612.3019878 (2017).

[11] Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D. and Elovici, Y.: Detection of Unauthorized IoT Devices Using Machine Learning Techniques (2017).

