

クライアントの共通したパケットロスごとのマルチキャストグループを使用した重複パケットの抑制による消費電力の削減

越後谷 滯¹ 大沢 恭平² 串田 高幸¹

概要: IoT 機器のファームウェアアップデートはゼロデイ攻撃よりも早くアップデートを行わなければならない。多数の IoT 機器のファームウェアアップデートには時間がかかる。そこでマルチキャストを使用することで一斉に更新することができ、更新にかかる時間が短くなる。マルチキャストで送信を行うとパケットロスすることがあり、再送信をしなければならない。課題は、重複パケットを受信した時にそのパケットを不要と判断し、破棄することが余分な処理になることである。本稿では、重複パケットの削減のため Default Multicast Group (DMG) と Packet loss Multicast Group (PMG) を提案する。DMG はマルチキャストでパケットを送信し、更にマルチキャストグループを切り替えるために使用する。PMG はパケットロスしたクライアントに共通しているシーケンス番号のパケットをマルチキャストで再送信するのに使用する。提案の方法で重複パケットを処理しないようにする。この結果、重複パケットが削減でき、消費電力が削減される。実験には ESP32 のファームウェアに見立てた 750KB のファイルを使用した。また ESP32 は 4 台を使用する。評価は 1 つのマルチキャストグループを使用した時と提案方式を使用した時の伝送時間と消費電力を比較した。結果として、提案方法を使用した時の伝送時間は 1 つのマルチキャストグループを使用した時と比べて平均で約 6 秒減少した。提案方法を使用した時の消費電力は 1 つのマルチキャストグループを使用した時と比べて平均で約 10.4% 減少した。

1. はじめに

背景

Internet of Things (IoT) とは、デバイス、車両、建物、ソフトウェア、センサーを例とする物理オブジェクトにネットワーク接続機能が組み込まれたものである [1]。IoT 機器はセキュリティの観点からファームウェアアップデートが必要になる [2]。ファームウェアとは、ハードウェアと通信する一種のソフトウェアである [3]。更に、ゼロデイ攻撃よりも早くアップデートを完了させなければならない [4]。複数の IoT 機器へのアップデートには時間がかかるため、マルチキャストを使用する [5]。マルチキャストはクライアントが所属している IP マルチキャストグループアドレスに対して送信する [6]。そのため、同時にパケットを送信することができる。マルチキャストを使用することで、多数のクライアントへの同時アップデートが可能になる [7]。マルチキャストの実装には User Datagram Protocol (UDP) が使用される [8]。UDP とは相互接続さ

れたコンピュータネットワークの環境で、パケット交換コンピュータ通信のデータグラムモードを利用できるようにするためのプロトコルである [9]。UDP は、コネクションレスプロトコルである。そのため、TCP よりも早くデータを送信することが出来る [10]。しかし、UDP はパケットロスする時がある [11]。パケットロスは、人や壁を例とする障害物があると発生する [12–14]。そのため、パケットロスを許容しないファームウェアは、パケットを再送信する必要がある [15]。パケットを識別するためにシーケンス番号がある。シーケンス番号があることにより、パケットロスを検知することが出来る [16]。しかし、UDP にはシーケンス番号が実装されていない [17]。

ネットワーク接続を行うため、サーバとクライアント間のデータの送受信はソケット通信を介して行う [18]。ソケット通信とは特定のポートを介してサーバとクライアントを接続し、リアルタイムの双方向通信を行う手法である [19]。

IoT 機器はスマートホームに使われている [20]。スマートホームとは、Information and Communication Technology (ICT) を使用して家電製品の制御が出来る。例えば窓の開閉、照明の点滅を自動化することである。スマートホームの中には IP カメラがある [21]。IP カメラとはイン

¹ 東京工科大学コンピュータサイエンス学部
クラウド・分散システム研究室
〒192-0982 東京都八王子市片倉町 1404-1

² 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

ターネットに接続し、ネットワーク経由で遠隔から映像を見ることが出来るカメラである [22]. 更に IP カメラはリアルタイムに情報を取得している [23].

IoT 機器には ESP32 がある. ESP32 とは、低コスト、低消費電力であり、Wireless Fidelity (Wi-Fi) および Bluetooth 機能を備えているマイクロコントローラである [24]. Wi-Fi は IoT 機器がインターネット接続を確立できるようにする無線通信技術である [25]. また、Wi-Fi はアクセスポイントから最大 30m の通信を可能とする [26].

課題

マルチキャストで再送信を行うと既に受信した IoT 機器にまで再送信を行う. この結果、既に受信した IoT 機器はパケットが重複する. この重複したパケットを重複パケットと定義する. 課題は、重複パケットを受信した時にそのパケットを不要と判断し、破棄することが余分な処理になることである. また、重複パケットの受信は、消費電力の増加を引き起こす [27]. 図 1 は重複パケットによる余分な処理が発生した時である. クライアント A, クライアント B, クライアント C, クライアント D は同じマルチキャストグループに所属している. 例えば、パケットロスをしたクライアントが A のみだったとする. サーバはクライアント A に送信パケットを再送パケットとして、再送信しなければならない. しかし、サーバはマルチキャストグループに再送信するため、既にパケットを受信したクライアント B, クライアント C, クライアント D にまで再送パケットを送信する. その結果、クライアント B, クライアント C, クライアント D で再送パケットが重複パケットとなる. この重複パケットは冗長であるため破棄する.

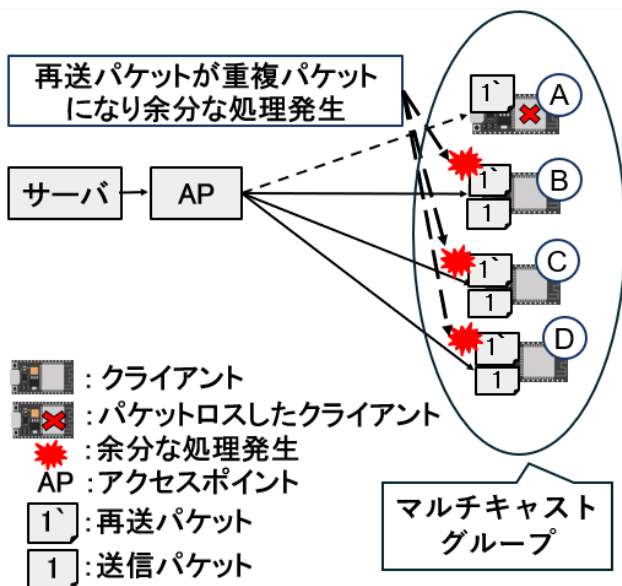


図 1 重複パケットによる余分な処理が発生した時

各章の概要

2 章では、本稿の関連研究について記述する. 3 章では、本稿の提案方式について記述する. 4 章では、提案方式の実装内容について記述する. 5 章では、1 つのマルチキャストグループと提案方法を使用した評価実験をし、その評価を記述する. 6 章では、提案方式についての議論を記述する. 7 章では、本稿のまとめを行う.

2. 関連研究

エンドツーエンドのレイテンシを低く抑えながらデータ配信率を向上させるマルチキャストデータ回復方式である PRM(Probabilistic Resilient Multicast) を研究している論文がある [28]. これは、オーバーレイノードの障害によってデータ配信パスが分割された場合に、高速なデータ回復を提供するために、ランダム化転送を行う. ランダム転送では、各オーバーレイノードは、小さな確率で、ランダムに選択されたオーバーレイエッジに沿って、いくつかの追加送信を送信する. この追加送信が重複パケットとなる. この解決策として確率を下げることで解決する. しかし、本稿の重複パケットは再送信によるものであるため、本稿の課題とは異なる.

Mobile Ad hoc Network (MANET) 環境でマルチキャストパケットを効率的に伝播するための Simplified Multicast Forwarding (SMF) がある. ハッシュ関数を使用しパケット識別子をパケットに割り当てる. そして、ゲートウェイ内で同一のハッシュ値をもったパケットを検出し、破棄する. これにより重複パケットが削減できる [29]. しかし、本研究ではハッシュ値を用いて破棄している. そのため、異なるパケットに同一のハッシュ値が割り振られる時、重複パケットと誤検知を行う.

大規模な信頼性の高いマルチキャストのための新しい損失回復方式である Active Reliable Multicast (ARM) を提案している研究がある. これは、ルータを使用し、多数の受信者からの NACK の数と修復トラフィックの両方を削減している. これにより、ルータは複数の受信者からの重複した NACK を抑制し、NACK のインプロージョンの問題を制御している [30]. しかし、NACK の削減をしているが、これは重複パケットの削減ではない.

3. 提案

提案方式

本稿では、再送信によって発生する重複パケットを削減し、消費電力を削減することが目的である. 再送するパケットを 1 つのマルチキャストアドレスに対して送信すると、再度 IoT 機器がパケットを受信し、処理する. したがって不要なパケットを処理しない仕組みが必要である. そこで、Default Multicast Group(DMG) と Packet loss Multicast Group(PMG) を提案する. DMG とは通常のパケットの

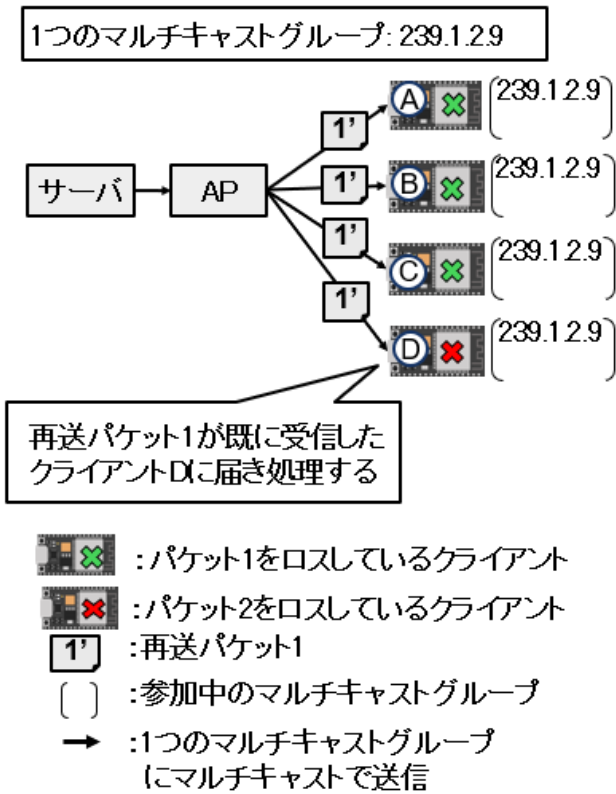


図 2 1つのマルチキャストグループを使用した場合

受信とマルチキャストグループの切り替えの通達するグループである。PMG とはクライアントに共通してロスしたシーケンス番号の packets を受信するグループである。

次に、1つのマルチキャストグループを使用した場合と DMG と PMG を使用した場合の比較を説明をする。図2は1つのマルチキャストグループを使用した時である。クライアント A, クライアント B, クライアント C, クライアント D は DMG に既に所属している。そして、クライアント A, クライアント B, クライアント C はパケット 1 のみをロスし、クライアント D はパケット 2 のみをロスしている状況である。パケット 1 を再送パケット 1 とした場合、1つのマルチキャストグループ宛てに再送パケットを送信する。しかし、クライアント D も再送パケット 1 を受信し処理する。

図3は DMG と PMG を使用した時である。クライアント A, クライアント B, クライアント C, クライアント D は DMG に既に所属している。そして、クライアント A, クライアント B, クライアント C はパケット 1 のみをロスし、クライアント D はパケット 2 のみをロスしている状況である。パケット 1 を再送パケット 1 とした場合、PMG 宛てに再送パケット 1 を送信する。その時、クライアント D は PMG に参加していないため、再送パケット 1 を処理しない。このようにして重複パケットを削減する。PMG の参加方法は後述する。

クライアントは自身以外のクライアントの packet loss

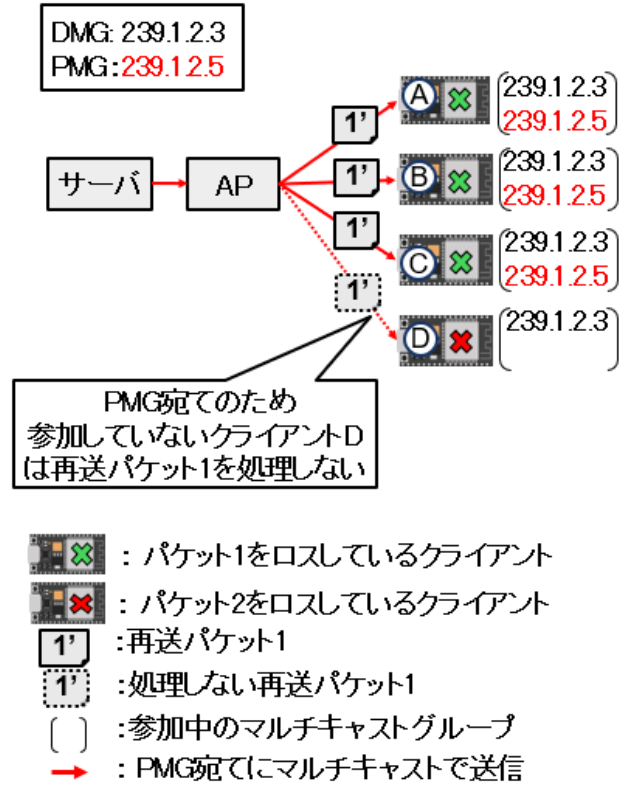


図 3 DMG と PMG を使用した場合

の状況が分からない。したがって、自身が packet loss したシーケンス番号は他のクライアントと共通しているのかが分からないため、PMG に参加する必要があるかが分からない。この解決策として、クライアントはサーバに逐一 packet loss の状況を報告し、サーバはその報告をもとにクライアントが PMG に参加する必要があるかを促さなければならない。図4は PMG への参加方法である。クライアントは packet loss があれば Packet Loss list (PL リスト) にロスしたシーケンス番号と自身の IP アドレスを格納し、サーバにユニキャストで送信する。その後、サーバはクライアントから収集した全ての PL リストを比較し、共通しているシーケンス番号があれば、それらを Common Packet list (CP リスト) に格納し、サーバはクライアントに DMG 宛てにマルチキャストで送信する。クライアントは自身の PL リストと CP リストを比較し、1つ以上一致しているシーケンス番号があれば PMG に参加する。その後、サーバは PMG 宛てに CP リストの packet をマルチキャストで再送信する。

アルゴリズムの概要

送信フェーズ、再送信フェーズ、共通あり再送信フェーズ、共通なし再送信フェーズの順に実行される。送信フェーズでは、サーバはクライアントに送信するデータを packet に分割し、DMG 宛てにマルチキャストで送信する。この時、packet loss があればクライアントからサーバにユニキャストで PL リストを送信する。全ての packet を送

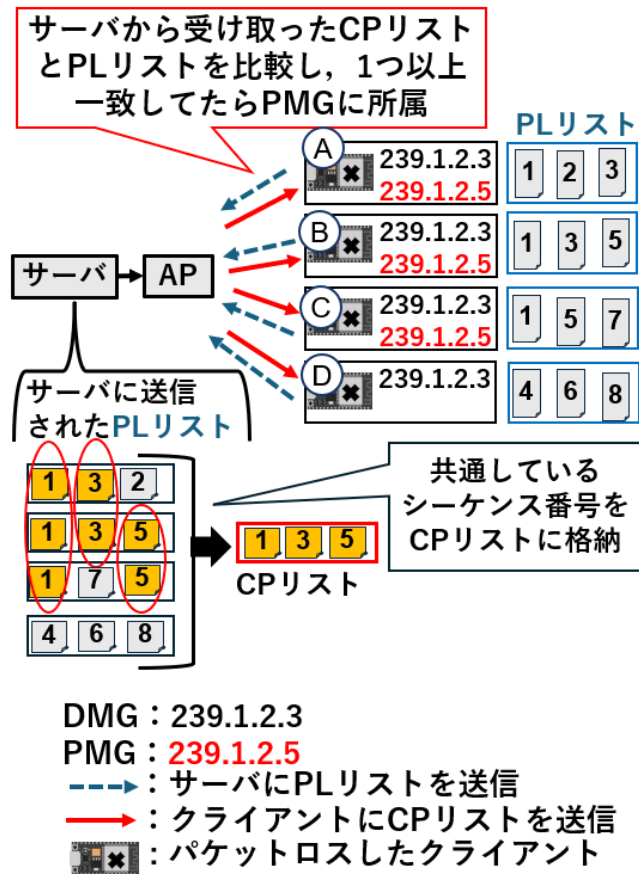


図 4 PMG への参加方法

信後、サーバはクライアントから送信された PL リストがあるならば再送信フェーズに移行し、無ければ実行終了する。再送信フェーズでは、サーバはクライアントから送信された全ての PL リストを比較し、共通しているシーケンス番号があれば共通あり再送信フェーズに移行し、無ければ共通なし再送信フェーズに移行する。共通あり再送信フェーズ移行時、サーバはクライアントに DMG 宛てに CP リストを送信する。クライアントは自身の PL リストと CP リストを比較し、1つ以上共通したシーケンス番号があれば、共通あり再送信フェーズに移行し、無ければ待機をする。共通あり再送信フェーズでは、サーバはクライアントに CP リストの packets を PMG 宛てにマルチキャストで再送信する。共通なし再送信フェーズでは、サーバはクライアントに CP リスト以外の PL リストのシーケンス番号の packets をユニキャストで再送信する。その後、クライアントから再度 PL リストを受け取った場合、再送信フェーズに戻る。

ユースケース・シナリオ

本稿のユースケースとして、ホテルでの複数の IP カメラの運用を想定している [31]。IP カメラはインターネット経由で画像や動画を送信する機能を備えたカメラである [32,33]。この際、セキュリティの脆弱性の改善のため

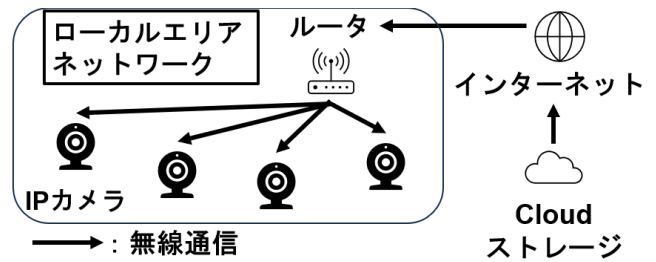


図 5 ユースケースの図

ファームウェアアップデートが必要になる [34]。

図 5 はユースケースの図である。IP カメラはルータを介し、インターネットと繋がり Cloud ストレージの監視情報をやり取りしている [35]。今回はこのトポロジーを想定し、ファームウェアアップデートを行う。このユースケースに本提案を使用することで、マルチキャストによるアップデートの時間が短縮、重複パケットの削減による消費電力の抑制が出来る。

4. 実装

実装では、提案したアルゴリズムの詳細を説明する。送信フェーズ、再送信フェーズ、共通あり再送信フェーズ、共通なし再送信フェーズの順で説明する。またクライアントは MicroPython を使用し、サーバは Python を使用して実装を行う。

4.1 送信フェーズ

アルゴリズム 1 は送信フェーズについてである。事前にクライアントとサーバは DMG に参加する。サーバから

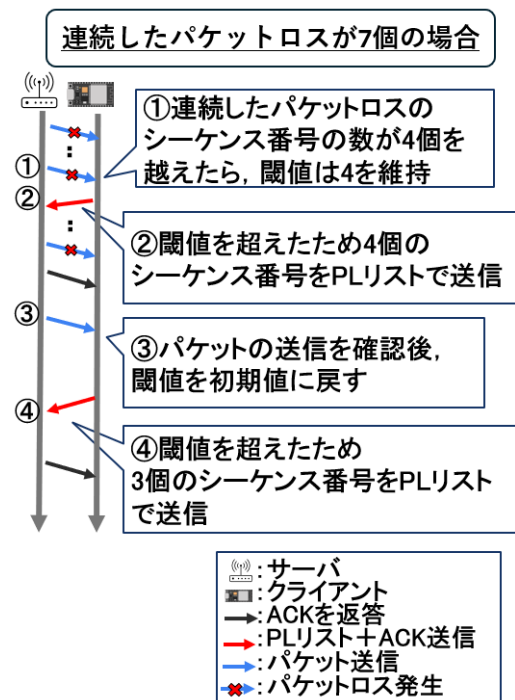


図 6 PL リストについて

アルゴリズム 1 送信フェーズについて

Require: サーバとクライアントは DMG に参加済

```
1: function SERVER
2:   FILE ← send750KB.txt           ▷ 送信するファイル
3:   chunks ← FILE を 1kB に分割 ▷ 1kB に分割したパケット
   を格納するリスト
4:   for all chunk ← chunks do
5:     chunk ← シーケンス番号を付与
6:     chunk を DMG 宛てにマルチキャストで送信
7:   end for
8: end function
9: function CLIENT(n)               ▷ n はクライアントの数
10:  receive_dict                    ▷ 受信したパケットを辞書型で保管
11:  PL_list                          ▷ PL リスト
12:  while サーバが 750 個のパケットを送信し終えるまで do
13:    receive_dict ← 受信したパケット
14:    A ← receive_dict[-2]          ▷ 前回受信したシーケンス番号
15:    B ← receive_dict[-1]          ▷ 今回受信したシーケンス番号
16:    if A + 1 ≠ B then             ▷ もしパケットロスがあれば
17:      PL_list ← パケットロスしたシーケンス番号
18:      N ← len(PL_list)
19:      if N > 閾値 then           ▷ PL リスト送信時の閾値
20:        PL_list を送信
21:      end if
22:    end if
23:  end while
24: end function
```

DMG 宛てにマルチキャストでパケットを送信する。送信するパケットは「シーケンス番号+データ」である。クライアントはパケットを受け取った時、シーケンス番号を確認する。確認時、前回で受け取ったシーケンス番号と連続でない場合、その欠番した全てのシーケンス番号をパケットロスと判定する。このパケットロスしたシーケンス番号

アルゴリズム 2 再送信フェーズについて

```
1: function SERVER
2:   for クライアントの PL リストを全て比較 do
3:     if 共通したシーケンス番号があれば then
4:       CP リストを DMG 宛てにマルチキャストで送信
5:     else
6:       ユニキャストで再送信するクライアントに ACK を
   送信
7:     end if
8:   end for
9: end function
10: function CLIENT(n)              ▷ n はクライアントの数
11:  if CP リストを受信したら then
12:    if CP リスト in PL リスト then
13:      PMG に参加する
14:      共通あり再送信フェーズに移行
15:    else
16:      ACK を受信するまで待機
17:    end if
18:  end if
19:  if ACK を受信したら then
20:    共通なし再送信フェーズに移行
21:  end if
22: end function
```

アルゴリズム 3 共通あり再送信フェーズについて

```
1: function SERVER
2:   for losses_packet ← CP リスト do
3:     losses_packet を PMG 宛てにマルチキャストで再送信
4:   end for
5: end function
6: function CLIENT(n)              ▷ n はクライアントの数
7:  receive_list                    ▷ 受信したパケットを辞書型で保管
8:  while PL リスト in CP リスト do
9:    receive_list ← 受信したパケット
10:   if パケットロスがあるかを確認 then
   ▷ もしパケットロスがあれば
11:     PL_list ← パケットロスしたシーケンス番号
12:     PL_list を送信
13:   end if
14: end while
15: end function
```

を PL リストに格納する。その後、格納されたシーケンス番号の個数が閾値を越えたらサーバに送信する。PL リストは最大で 4 個のシーケンス番号をサーバに送信出来る。閾値は以下で説明する。

4.1.1 PL リスト

PL リストとはクライアントのパケットロスしたシーケンス番号を格納するものである。PL リスト送信時、共通なし再送信フェーズでユニキャストを使用した再送信を行うため、クライアント自身の IP アドレスを付与する。全てのパケットを受け取るまで、クライアントは PL リストを保持し続ける。

次に閾値について説明する。閾値の初期値は 2、最大値を 4 とする [36]。閾値は連続したパケットロスのシーケンス番号の数が 2 個の場合は 2 のままにし、3 個の場合は 3 に増やし、4 個の場合は 4 に増やす。5 個以上の場合は、閾値を 4 のままにする。次に、パケットロスのシーケンス番

アルゴリズム 4 共通なし再送信フェーズ

```
1: function SERVER
2:   while x > 0 do                 ▷ x は PL リストの数
3:     x のクライアントにユニキャストで再送信
4:   end while
5:   if 再度受け取った PL リストがあるかを確認 then
   ▷ もし PL リストがあれば
6:     再送信フェーズに移行
7:   end if
8: end function
9: function CLIENT(n)              ▷ n はクライアントの数
10:  receive_list                    ▷ 受信したパケットを辞書型で保管
11:  while 750 個のパケットを全て受け取るまで do
12:    receive_list ← 受信したパケット
13:    if パケットロスがあるかを確認 then
   ▷ もしパケットロスがあれば
14:      PL_list ← パケットロスしたシーケンス番号
15:      PL_list を送信
16:    end if
17:  end while
18: end function
```

号の数が1個の場合を説明する。1個のペケットロスしたシーケンス番号をPLリストに保持し、次のペケットロスを待つ。次のペケットロスが発生した際、PLリストに保持した1個とそのペケットロスを包括し、閾値を決定する。サーバが全てのペケットを送信し終え、PLリストに保持した数が1個の場合、タイムアウト時間を設け、タイムアウト後にPLリストを送信する。また正常にペケットの受信が確認出来たら閾値を初期値にする。

図6はPLリストの送信タイミングの例についてである。連続したペケットロスのシーケンス番号の数が7個の状態である。連続したペケットロスのシーケンス番号が4個を越えた際、4個のシーケンス番号を格納したPLリストをサーバに送信する。その後、ペケットの送信が確認が出来たら閾値が初期値に戻り、閾値を超えるため残りの3個のシーケンス番号をPLリストでサーバに送信する。

4.2 再送信フェーズ

アルゴリズム2は再送信フェーズについてである。サーバはクライアントから受け取った全てのPLリストのシーケンス番号を比較する。各PLリストのシーケンス番号の比較で、共通しているシーケンス番号があれば共通あり再送フェーズに移行し、共通しているシーケンス番号が無ければ共通なし再送信フェーズに移行する。サーバは共通しているシーケンス番号があれば、CPリストに格納し、DMG宛てにマルチキャストでクライアントに送信する。クライアントは受け取ったCPリストとPLリストを比較し、一致しているシーケンス番号があればクライアントは共通あり再送信フェーズに移行する。サーバからACKを受け取ったら、共通なし再送信フェーズに移行する。

4.3 共通あり再送信フェーズ

アルゴリズム3は共通あり再送信フェーズについてである。サーバはCPリストにあるシーケンス番号のペケットをPMG宛てにマルチキャストで再送信する。再送信するペケットを送信中にロスした場合、クライアントはそのロスしたシーケンス番号をPLリストに再登録し、サーバに送信する。

4.4 共通なし再送フェーズ

アルゴリズム4は共通なし再送信フェーズについてである。サーバはCPリスト以外のPLリストのペケットを各クライアントのIPアドレス宛てにユニキャストで再送信する。クライアントはこの再送信でロスしたシーケンス番号をPLリストに再登録し、サーバに送信する。サーバはクライアントにユニキャストで再送信後、クライアントから再度受け取ったPLリストを確認し、再送信フェーズに移行する。この操作を全てのクライアントがPLリストが無くなるまで行う。

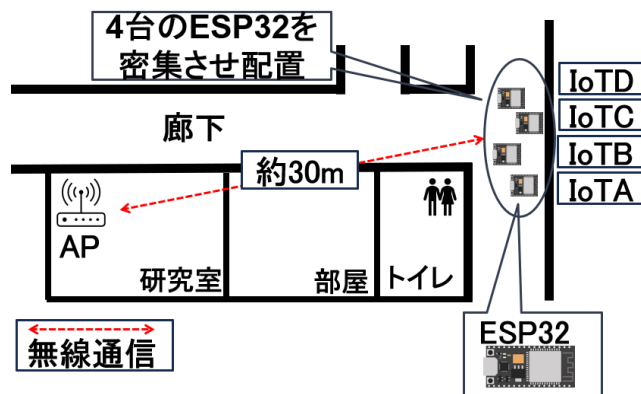


図7 ESP32の配置図

5. 評価実験

評価方法

評価方法は、1つのマルチキャストグループを使用した時と提案方式を使用した時の伝送時間と消費電力を比較する。伝送時間はクライアントがペケットを受信し始めてから全てのペケットを受信し終えるまでの時間とする。消費電力は伝送時間と同様の時間を計測し、10msごとに値を取得する [37].

実験環境

実験で使用するクライアントはEspressif Systems社のIoT機器であるESP32を使用した。更に、ESP32は図7の配置の方法で行った。サーバはBroadcom社のVMware製品であるESXiを使用した。またESP32は4台使用した [38]. サーバから送信するファイルはESP32のファームウェアに見立てた750KBのファイルとする [39]. 実験方法は、ESP32のファームウェアに見立てた750KBのファイルをクライアント4台に向けて送信した。この時、送信する方法は1つのマルチキャストグループを使用した時と提案方式を使用した時の2種類を行った。この操作を各15回行った [40]. 送信するファイルのペケットサイズはMaximum Transmission Unit(MTU)をもとにペケットサイズの上限は1500バイトとする [41]. また、MTUの上限の限界のペケットサイズの送信はスループットの低下やペケットロスに繋がる [42]. したがってペケットサイズには余裕を持たせるべきである。そのためペケットは1KBとした。

実験結果と分析

図8は1つのマルチキャストグループを使用した時の伝送時間と消費電力である。1つのマルチキャストグループを使用した時の伝送時間の平均値はIoT Aが約102秒、IoTBが約102秒、IoT Cが約106秒、IoT Dが約112秒であった。1つのマルチキャストグループを使用した時の消

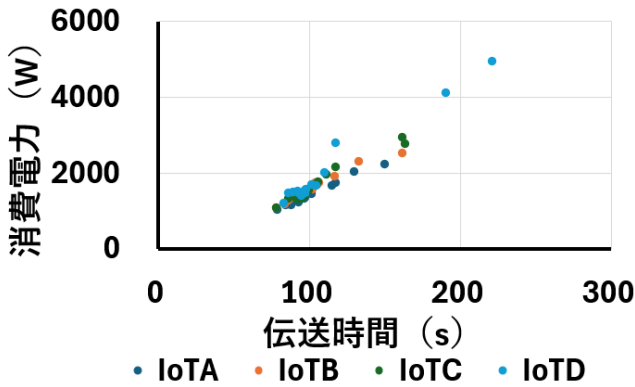


図 8 1つのマルチキャストグループを使用した時の伝送時間と消費電力

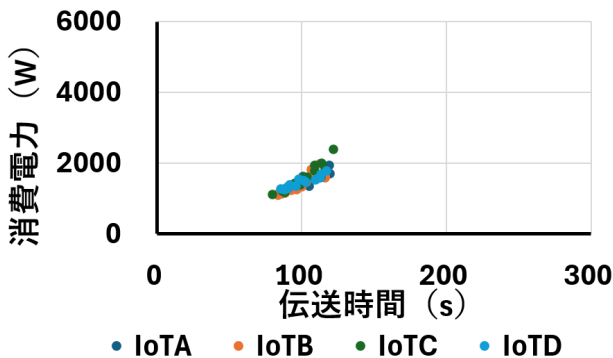


図 9 提案方式を使用した時の伝送時間と消費電力

消費電力の平均値は IoTA が約 1436 W, IoTB が約 1577W, IoTC が約 1658W, IoTD が約 2014W であった。

図 9 は提案方式を使用した時の伝送時間と消費電力である。提案方式を使用した時の伝送時間の平均値は IoTA が約 101 秒, IoTB が約 99 秒, IoTC が約 101 秒, IoTD が約 100 秒であった。提案方式を使用した時の消費電力の平均値は IoTA が約 1469W, IoTB が約 1409W, IoTC が約 1568W, IoTD が約 1454W であった。

これらの結果より提案方式を使用した時の伝送時間は 1 つのマルチキャストグループを使用した時と比べて IoTA が約 1 秒減少, IoTB が約 4 秒減少, IoTC が約 5 秒減少, IoTD が約 12 秒減少し, 平均で約 6 秒減少した。したがって, 全てのクライアントで伝送時間が減少した。提案方法を使用した時の消費電力は 1 つのマルチキャストグループを使用した時と比べて IoTA が約 2.3%増加, IoTB が約 10.7%減少, IoTC が約 5.4%減少, IoTD が約 27.8%減少し, 平均で約 10.4%減少した。したがって, クライアント A 以外のクライアントで消費電力が減少した。

IoTA の消費電力が増加した原因は, 共通無し再送信フェーズにある。IoTA 以外の再送信中, IoTA は他のクライアントのユニキャストのため, 待機をする。この待機時間に余分な消費電力が発生したと考える。

6. 議論

本稿ではクライアントから PL リストを収集するための時間やマルチキャストグループの切り替えのための時間を設定していなかった。このことによって, サーバが全てのクライアントからの PL リストの受信を待機する時間を増やすのか, あるいは再送信を優先するのか分からない。この対策として Retransmission Timeout (RTO) をもとにタイムアウトを設定することによってこの問題を解決出来る [43]。

PL リストは UDP で再送信されているため, 人や壁を例とする障害物があるとロスする時がある。この結果, 永久的に再送要求が続き, クライアントはいつ接続が切れたのか分からない。したがって, 再送回数を設定する必要がある。再送信の上限回数を 4 回とする [44]。この上限回数を上回り, クライアントからサーバに PL リストが届かない時, クライアントは接続に問題があるとし, 再送信を中断する。

評価実験では IoTA のみが消費電力が増加した。原因として, ユニキャストで再送信中, 自身は他のクライアントのユニキャストのため, 待機をする。この待機中に余分な消費電力が発生した。この対策として, クライアントは他のクライアントのユニキャスト中に, スリープを行うことでこの問題を解決出来る [45]。

本稿の提案である PMG はクライアントが CP リストを受け取って参加する。しかし, CP リストはマルチキャストで送信しているため, 人や壁を例とする障害物でロスする時がある。その結果, クライアントは PMG に参加出来ない。この対策として, 送信フェーズ中に収集した PL リストを確認し, PMG に参加するクライアントがいた場合, 参加を促す通知を DMG 宛てに送信する。更に, ACK を用いて PMG に参加をしているかを確認し, 参加が出来なければ CP リストを 2 秒おきに, 再送信する [46]。

7. おわりに

課題はマルチキャストで再送信を行うと正常に受け取った機器にまで再送処理を行い, 重複パケットが発生することである。提案では, IoT 機器が再度受信したパケットを処理しないために, DMG と PMG を設定した。評価実験はクライアントとして ESP32 を 4 台, サーバとして ESXi の仮想環境を使用し, 750KB のファイルを送信した。評価方法は 1 つのマルチキャストグループを使用した時と提案方式を使用した時の伝送時間と消費電力である。結果として, 提案方式を使用した時の伝送時間は 1 つのマルチキャストグループを使用した時と比べて平均で約 6 秒減少した。提案方式を使用した時の消費電力は 1 つのマルチキャストグループを使用した時と比べて平均で約 10.4%減少した。

参考文献

- [1] Gokhale, P., Bhat, O. and Bhat, S.: Introduction to IOT, *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 5, No. 1, pp. 41–44 (2018).
- [2] Zandberg, K., Schleiser, K., Acosta, F., Tschofenig, H. and Baccelli, E.: Secure firmware updates for constrained iot devices using open standards: A reality check, *IEEE access*, Vol. 7, pp. 71907–71920 (2019).
- [3] Hemram, S., Kathrine, G. J. W., Palmer, G. M. and Edwards, S. V.: Firmware vulnerability detection in embedded systems and internet of things, *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, IEEE, pp. 1161–1167 (2022).
- [4] El Jaouhari, S. and Bouvet, E.: Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions, *Internet of Things*, Vol. 18, p. 100508 (2022).
- [5] El Heni, N. and Lagrange, X.: Multicast vs multiple unicast scheduling in high-speed cellular networks, *VTC Spring 2008-IEEE Vehicular Technology Conference*, IEEE, pp. 2456–2460 (2008).
- [6] Li, X., Paul, S., Pancha, P. and Ammar, M.: Layered video multicast with retransmission (LVMR): Evaluation of error recovery schemes, *Proceedings of 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97)*, IEEE, pp. 161–172 (1997).
- [7] Li, J., Zhou, Y. and Chen, H.: Age of information for multicast transmission with fixed and random deadlines in IoT systems, *IEEE Internet of Things Journal*, Vol. 7, No. 9, pp. 8178–8191 (2020).
- [8] Bae, S. H., Lee, S.-J. and Gerla, M.: Multicast protocol implementation and validation in an ad hoc network testbed, *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)*, Vol. 10, IEEE, pp. 3196–3200 (2001).
- [9] Postel, J.: Rfc0768: User datagram protocol (1980).
- [10] Malhotra, A., Sharma, V., Gandhi, P. and Purohit, N.: UDP based chat application, *2010 2nd International Conference on Computer Engineering and Technology*, Vol. 6, IEEE, pp. V6–374 (2010).
- [11] Kumar, S. and Rai, S.: Survey on transport layer protocols: TCP & UDP, *International Journal of Computer Applications*, Vol. 46, No. 7, pp. 20–25 (2012).
- [12] Lee, T.-H., Xie, X.-S. and Chang, L.-H.: RSSI-based IPv6 routing metrics for RPL in low-power and lossy networks, *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, pp. 1714–1719 (2014).
- [13] Sasiwat, Y., Jindapetch, N., Buranapanichkit, D. and Booranawong, A.: An experimental study of human movement effects on RSSI levels in an indoor wireless network, *2019 12th Biomedical Engineering International Conference (BMEiCON)*, IEEE, pp. 1–5 (2019).
- [14] Dong, Q. and Dargie, W.: Evaluation of the reliability of RSSI for indoor localization, *2012 International Conference on Wireless Communications in Underground and Confined Areas*, IEEE, pp. 1–6 (2012).
- [15] Bauwens, J., Ruckebusch, P., Giannoulis, S., Moerman, I. and De Poorter, E.: Over-the-air software updates in the internet of things: An overview of key principles, *IEEE Communications Magazine*, Vol. 58, No. 2, pp. 35–41 (2020).
- [16] Bohacek, S., Hespanha, J. P., Lee, J., Lim, C. and Obraczka, K.: A new TCP for persistent packet reordering, *IEEE/ACM Transactions on networking*, Vol. 14, No. 2, pp. 369–382 (2006).
- [17] Games, E. and Surós, R.: An upper bound model for TCP and UDP throughput in IPv4 and IPv6, *Journal of Network and Computer Applications*, Vol. 31, No. 4, pp. 585–602 (2008).
- [18] Radhika, K., Raksha, B., Sujatha, B., Pruthviraj, U. and Gangadharan, K.: IoT based joystick controlled pi-bot using socket communication, *2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, IEEE, pp. 121–125 (2018).
- [19] An, J. H., Jung, K. H., Kim, S. Y., Mun, J. S. and Han, M. G.: Intelligent real-time control system through socket communication using deep learning-based de-hazing and object detection in an embedded board environment, *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, pp. 1494–1497 (2021).
- [20] Malche, T. and Maheshwary, P.: Internet of Things (IoT) for building smart home system, *2017 International conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, IEEE, pp. 65–70 (2017).
- [21] Abdallah, R., Xu, L. and Shi, W.: Lessons and experiences of a DIY smart home, *Proceedings of the Workshop on Smart Internet of Things*, pp. 1–6 (2017).
- [22] Tamiya, K., Bokhari, A. H., Ezawa, Y., Nakayama, S., Tie, Y., Tanabe, R., Fujita, A., Yoshioka, K. and Matsumoto, T.: Dangers of IP Camera-An Observational Study on Peeping, *Journal of Information Processing*, Vol. 28, pp. 502–510 (2020).
- [23] Farley, A., Ham, H. et al.: Real time IP camera parking occupancy detection using deep learning, *Procedia Computer Science*, Vol. 179, pp. 606–614 (2021).
- [24] Maier, A., Sharp, A. and Vagapov, Y.: Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things, *2017 Internet Technologies and Applications (ITA)*, IEEE, pp. 143–148 (2017).
- [25] Wang, F.: Neuro-WiFi: A novel neuronal connection underlies the potential interventional target, *Science Insights*, Vol. 43, No. 6, pp. 1179–1196 (2023).
- [26] Yoo, S., Shin, Y., Kim, S. and Choi, S.: Toward realistic WiFi simulation with smartphone “Physics”, *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, IEEE, pp. 1–6 (2014).
- [27] Gharat, C. and Krishnan, S.: Effects of duplicate packet transmission in timer based co-ordination opportunistic routing Scheme, *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, pp. 401–405 (2019).
- [28] Banerjee, S., Lee, S., Bhattacharjee, B. and Srinivasan, A.: Resilient multicast using overlays, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 31, No. 1, pp. 102–113 (2003).
- [29] Landmark, L., Lacharite, Y. and Lamont, L.: Multicast forwarding using multiple gateways and hash for duplicate packet detection in a tactical MANET, *MILCOM 2007-IEEE Military Communications Conference*, IEEE, pp. 1–7 (2007).
- [30] Lehman, L.-W. H., Garland, S. J. and Tennenhouse, D. L.: Active reliable multicast, *Proceedings. IEEE INFOCOM'98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the*

- IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No. 98, Vol. 2, IEEE, pp. 581–589 (1998).*
- [31] Roy, V., Alam, M. I. and Alam, M. S.: Various use of cloud computing with the help of IP camera based on line, *International Journal of Computer Applications*, Vol. 157, No. 2, pp. 4–8 (2017).
- [32] Hintermaier, W. and Steinbach, E.: A system architecture for IP-camera based driver assistance applications, *2010 IEEE Intelligent Vehicles Symposium*, IEEE, pp. 540–547 (2010).
- [33] Yang, M.-J., Tham, J. Y., Wu, D. and Goh, K. H.: Cost effective IP camera for video surveillance, *2009 4th IEEE conference on industrial electronics and applications*, IEEE, pp. 2432–2435 (2009).
- [34] Puesche, A., Bothe, D., Niemeyer, M., Sachweh, S., Pohlmann, N. and Kunold, I.: Concept of Smart Building Cyber-physical Systems Including Tamper Resistant Endpoints, *2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, IEEE, pp. 000127–000132 (2018).
- [35] Manske, A.: Conducting a vulnerability assessment of an IP camera (2019).
- [36] De Oliveira, R. and Braun, T.: A dynamic adaptive acknowledgment strategy for TCP over multihop wireless networks, *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 3, IEEE, pp. 1863–1874 (2005).
- [37] Rice, A. and Hay, S.: Decomposing power measurements for mobile devices, *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, pp. 70–78 (2010).
- [38] Wedashwara, W., Wijayanto, H., Jatmika, A. H. and Arimbawa, I. W. A.: Smart EV Navigation and Data Collection System for Tree Based Data Modeling Using IoT, *First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)(MIMSE-IC-2022)*, Atlantis Press, pp. 130–141 (2022).
- [39] Börsig, M., Nitzsche, S., Eisele, M., Gröll, R., Becker, J. and Baumgart, I.: Fuzzing framework for ESP32 microcontrollers, *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, pp. 1–6 (2020).
- [40] Hindle, A.: Green mining: A methodology of relating software change to power consumption, *2012 9th IEEE working conference on mining software repositories (MSR)*, IEEE, pp. 78–87 (2012).
- [41] Behnam, M., Marau, R. and Pedreiras, P.: Analysis and optimization of the MTU in real-time communications over Switched Ethernet, *ETFA2011*, IEEE, pp. 1–7 (2011).
- [42] Islam, N., Bawn, C. C., Hasan, J., Swapna, A. I. and Rahman, M. S.: Quality of service analysis of Ethernet network based on packet size, *Journal of Computer and communications*, Vol. 4, No. 4, pp. 63–72 (2016).
- [43] Kesselman, A. and Mansour, Y.: Optimizing TCP retransmission timeout, *Networking-ICN 2005: 4th International Conference on Networking, Reunion Island, France, April 17-21, 2005, Proceedings, Part II 4*, Springer, pp. 133–140 (2005).
- [44] Balandina, E., Koucheryavy, Y. and Gurtov, A.: Computing the retransmission timeout in CoAP, *Conference on Internet of Things and Smart Spaces*, Springer, pp. 352–362 (2013).
- [45] Mezghani, O. and Mezghani, M.: An energy-efficient sleep scheduling protocol for large-scale cluster-based Mobile Wireless Sensor Networks, *2022 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, pp. 512–517 (2022).
- [46] Gyamfi, K. S., Brusey, J., Gaura, E. and Wilkins, R.: Heartbeat design for energy-aware IoT: Are your sensors alive?, *Expert Systems with Applications*, Vol. 128, pp. 124–139 (2019).