「マルチメディア，分散，協調とモバイル
(DICOMO2019) シンポジウム」 令和元年7月

許容待ち時間の短いパケットが多いネットワークにおけるパケット集約手法

古田 貴也[1] 高橋 竜一[2] 深澤 良彰[1]

概要：センサが様々なデータを送信するスマートホームをはじめとする IoT 環境では，小サイズのセンサデータがリアルタイムで大量にやりとりされる．リアルタイム性が重要なサービスでは，データの提供までに許容できる待ち時間が重要であり，この時間はそのデータを持つパケットの許容待ち時間と呼ばれる．本稿では IoT アプリケーションのように小サイズで許容待ち時間の短いパケットが通信の大部分を占めるような環境において，パケットを複数のバッファに割り振ることにより，効率的なデータ集約が可能となる手法を提案する．

A Packet Aggregation Scheme for Communication Networks Consisted of Mainly Low-Latency Required Packets

TAKAYA FURUTA[1] RYUICHI TAKAHASHI[2] YOSHIAKI FUKAZAWA[1]

1. はじめに

昨今のインターネットの発展とともに，インターネットに繋がる「モノ」の急激な増加が見込まれる．総務省が発行している平成 30 年版情報通信白書においても IoT デバイスの急速な普及が取り上げられている．その調査によると世界の IoT デバイス数は 2017 年には 274.9 億台であるが 2020 年には 403 億台にまで増加すると推測されている [1]．特に今後増加すると予想されている自動車産業，医療分野，そしてスマートシティやスマート工場など，どの分野においても，大量のセンサが用いられている．

一つ一つのセンサが生成するデータは，気温を表した数値やものが触れたことを示す信号などであるが，全体的に小さいサイズのものが多い．さらにそのような情報を分析してリアルタイムで結果を反映していくことが求められる．つまり IoT 環境では，パケットサイズが小さく，素早いデータのやりとりが求められる．

やりとりされるデータはパケット形式で送受信がおこなわれる．**図 1** にパケットの例として TCP のパケット構成

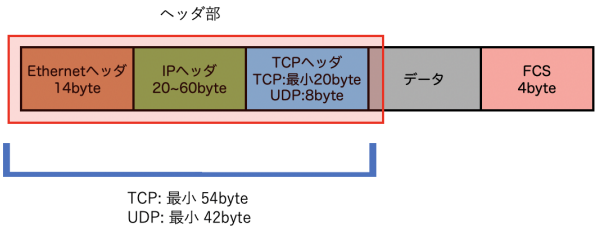

図 1 TCP パケット内部構成

図を示す [2]．図 1 に示したように，UDP ではヘッダ部分が最低でも 42byte，TCP では最低でも 54byte もある．数バイトであることが多いセンサからのデータのようにデータサイズが小さいときはパケット全体に対してヘッダ部の占める割合が大きいことがわかる．

このため，送信するデータの量を減らすためにパケットヘッダが消費するデータ量を小さくする方法が考えられる．その一つの方法としてパケットをそのまま送信するのではなく，集約をしてから送信しようという研究が数多くされている．一方でパケットの集約はデータの送受信に遅延をもたらすというデメリットも存在する．

多くのセンサアプリケーションのようにリアルタイム性が重視される場合には，データの提供までに許容できる待ち時間が重要である．本論文では，パケットの許容待ち時

1 早稲田大学
Waseda University
2 茨城大学
Ibaraki University

間の短いパケットが多いネットワークにおける効率のよいパケット集約手法を提案する．

2. 関連研究

本章ではパケット集約に関連する研究を紹介する．

2.1 パケット集約時のトレードオフに着目した研究

Deng らはパケットサイズおよびパケットの到着時間の様々な性質に適応し，遅延を最小限に抑えて集約を行うことのできる調整可能な選択ウィンドウを利用した集約アルゴリズムを提案した [3]．彼らはパケット集約における集約効率と集約によって生じる遅延のトレードオフに着目しており，そのトレードオフを最適化できる手法となっていた．

Yasuda らはコンテンツ指向型ネットワーク (CCN) におけるパケット集約手法を提案した [4]．集約される確率と集約にかかる時間の間の関係に重点をおいた研究で，集約をしながら全体的な遅延の短縮を実現していた．

2.2 パケットのヘッダのデータ量を削減する集約手法

Wang らはパケット集約のメリットの一つであるヘッダの圧縮に焦点をあてた研究を行なっている [5]．彼らは自動車ネットワークアプリケーションをターゲットとしてパケットのヘッダを圧縮するアルゴリズムを提案した．このアルゴリズムは通信における End-to-End の遅延とパケットの到着レートの観点で既存のパケット伝送手法に対して優れている．

Tetsuya Yokotani らは IoT 環境を想定し，センサから生成されるデータサイズがとても小さいことに着目し，複数のパケットを一つに集約することでヘッダ部分のデータの無駄遣いを抑えるような手法を二つ提案した [6]．彼らは，センサからの信号を受信した IoT デバイスでパケットが生成され，それらがデータ処理プラットホームに転送されるようなネットワーク構成を想定した．センサからの信号を受信するたびに転送するのではなく，何個かの信号を受信したら一つのパケットとしてまとめて送信を行う．二つ目の提案は，センサからのデータがあまり変化しないことを前提に，センサの値が変化したタイミングで集約したデータを送信する方法である．ただし，パケットロスがあった場合にデータの変更が反映されなくなってしまうため，パケットの送受信を保証する仕組みをあわせて提案している．

Nomura らはマルチレート伝送と VoIP パケットの伝送によって生じる伝送効率の低下を防げる高効率なパケット集約手法を提案した [7]．VoIP パケットのように長い遅延が許されないものを対象に高い QoS の実現を目指した．

2.3 ルータでのパケット集約手法に関する研究

VoIP アプリケーションをはじめとするアプリケーションの普及により，小さいサイズのパケットがネットワークに大量に放出される環境を問題視し，ルータでパケットを集約することで，送信されるパケットの個数を少なくする方法も研究されている．Tounsi らはこの問題設定のもと，同じ目的地へと送信されるパケットをネットワークの入り口のルータで集約して送信することを提案した [8]．彼らは到着したパケットを単純な FIFO で送信する方法と，提案した集約ありの FIFO 方式を比較し，集約によって通信資源の有効活用ができることを示した．Sawabe らはこの手法をさらに改善し，集約を複数のキューを利用して行う方法を提案した [9]．しかし，短い遅延時間を要求するパケットが全体の大部分を占めるような偏ったトラフィックについての議論はされていない．

3. IoT 環境における問題点

Anan Sawabe らはパケットを集約して送信することでパケット数を減らす一方でサービスの QoS を保証できるような集約手法を提案した [9]．彼らはルータが一秒間に送信することのできるパケット数の限界を示す packet/sec に着目した．パケット数が極端に多く，一つ一つのパケットのサイズが小さいような場合，本来そのルータが出せるはずの bit/sec を大きく下回るスループットしか出せない．Sawabe らはこの事実に基づき送信パケット数を減らすことでスループットの向上を目指した．

パケットの集約を行うと送信までの待ち時間が発生する．しかしアプリケーションごとにデータ処理に要求される遅延時間は異なるため，集約にかけることが許される時間はパケットによって異なる．その集約にかけられる時間を考えるにあたり，彼らは，通信全体を通して遅延が生じる場所は共通していることを利用した．片方向通信において，パケットが通信のどの過程で遅延を生じているか，またそれにかかっている時間の種類を**図 2** に示した．図 2 に示したように，一つ目の遅延は転送に必要なパケットの情報を識別する時間 X である．二つ目は集約にかけられる時間 W である．三つ目はネットワークで生じる遅延時間 D である．アプリケーションとして待つことのできる時間 Q がわかっているとすると，集約にかけられる最大時間 W_{max} は以下の式 1 で得られる．

$$W_{max} = Q - (X + D) \tag{1}$$

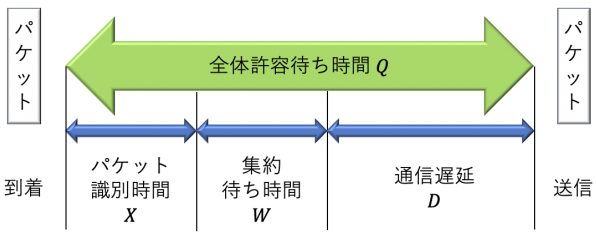

図 2 片方向通信遅延時間
Fig. 2 One Way Communication Delay

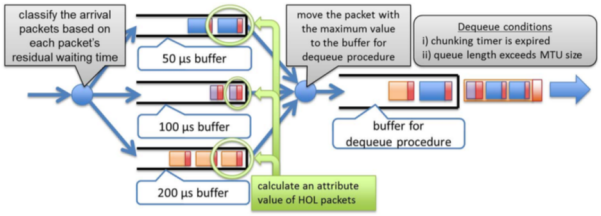

図 3 Sawabe らの手法 (出典：A. Sawabe et al.[9])
Fig. 3 Approach Proposed by Sawabe et al.[9]

彼らは式 1 に基づいてルータでの集約手法を提案した．その仕組みを**図 3** に示す．パケットにはそれぞれ集約のために待つことのできる最大時間 W_{max} が存在する．その時間は最初のパケットの識別を行なう過程でわかる．まず，ルータに到着したパケットは W_{max} に応じて三つのバッファに割り振られる．次に，三つのバッファの先頭のパケットの残りの待てる時間を比較し，最も短い待ち時間のパケットを集約用のバッファに移動させる．最後に，集約用バッファでは，集約したパケットのサイズが送信できる最大のサイズ（この提案では 9000byte としている）に達するかパケットが待てなくなるまで時間がたったら集約を終えて送信する．

IoT 環境では短い許容待ち時間のパケットが大量に流れる偏ったトラフィックの議論が必要であるが，この研究では十分に検討されていない．

[9] ではパケットの種類を三種類に設定し，それぞれの種類のパケットは一つのパケットサイズと許容待ち時間を定めて研究を行なった．この値に基づいてパケットのバッファへの割り振り方が決定されたが，その時間の設定方法の最適値はトラフィックごとに異なると考えられる．Sawabe らが行なったように，バッファへのパケットの割り振り方をアプリケーションの種類で分ける方法では，トラフィックの内訳が均等なときには十分よい結果が出ていた．ただし種類の偏ったパケットが到来するときのバッファのパケットの割り振る時間設定については触れていないので本研究ではその点について議論していく．

4. 本研究の特徴と詳細

本章では本研究の概要と提案の詳細を述べる．

4.1 概要

単一のバッファで集約をおこなうとき，後から到着したパケットが一定時間を経過しても集約されず送信に至らないために遅延を超過してしまう可能性が高い．これに対して，これまでの研究では，IoT 環境のようにリアルタイム性が求められるパケットが大量に到着するような状況において課題がある．この問題点を**図 4** に示した．図 4 に示したように，アプリケーションごとに時間を設定していてはそのアプリケーションの待ち時間のパケットが大部分を

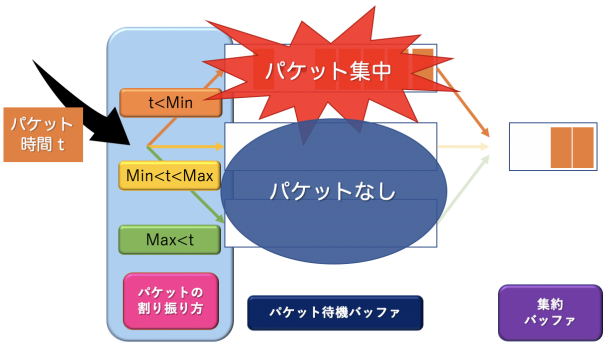

図 4 許容待ち時間の短いパケットが多いトラフィック
Fig. 4 Traffic Consisted of Mainly Low-Latency Required Packets

占めるトラフィックのもとでは複数のバッファを用いていても，実際には，一つのバッファしか働いていないことになる．単一のバッファのみを使用した集約の場合，パケットの待ち行列が長くなったとき，集約を待っているパケットの許容待ち時間を超過してしまう可能性が高くなってしまうという欠点がある．この欠点を改善するために複数のバッファを利用しているにも関わらず，偏ったトラフィック環境下で一つしか稼働していないのではこの手法の目的が達成されない．そこで，本研究では許容待ち時間の近いパケットが大量に到着するような環境でも，複数のバッファを有効に利用して特長を活かせるようにパケットの割り振り方を定める方法を提案する．

本研究では，小サイズで許容待ち時間の短いパケットが通信の大部分を占めるような環境においても効率的なデータ集約を実現できるように，バッファへのパケットの割り振り方を決定する時間設定を求める手法を提案する．

4.2 方法の詳細

本研究で着目したいパケットの割り振り方を**図 5** に示した．

Sawabe らの研究と同様に，許容待ち時間とサイズが定められたパケットが到着したら，その待ち時間に基づいて三つの待機バッファに割り振る．短い許容待ち時間のパケットが入る待機バッファから順に最小バッファ，中間バッ

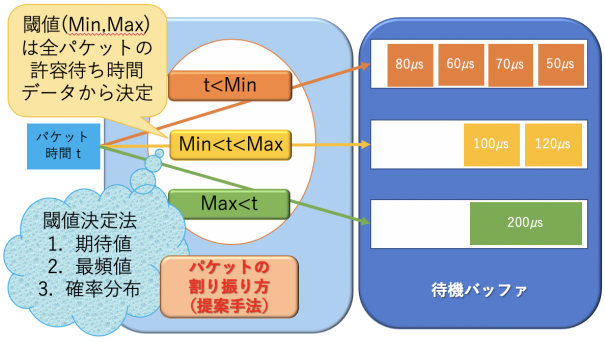

図 5 提案手法
Fig. 5 Proposal

表 1 パケットの分類
Table 1 Classification of Packets

アプリケーションタイプ	パケットサイズ	許容待ち時間
センサ	小	短
リアルタイム	中	中
ファイル転送	大	長

ファ，最大バッファと呼ぶこととする．それぞれのバッファで待機している先頭のパケットの内最小の待ち時間のパケットを集約用バッファに移動させる．パケットを待機バッファから集約用バッファに移動させるたびに，集約にかけられる時間は送信待ちのパケットの中で最も早く時間切れになってしまうものと同じに更新される．集約用バッファでは 9000 バイトに達するかそれ以上待てなくなったらパケットの集約を終えて送信する．9000 バイトは一つのパケットとして送信可能な最大サイズである．

本研究では，従来の研究とは異なり，パケットの待ち時間が固定値からある程度の幅を持った値に変えることにより用意するバッファの時間の設定方法の調整が重要である．本提案では図 5 に示したように，パケットがどのバッファに入るかを決定する下側閾値 Min と上側閾値 Max の二つの時間を設定することでパケットの許容待ち時間に応じて三つのバッファに分配する．最小バッファには許容待ち時間が下側閾値 Min よりも短いパケットが，中間バッファのバッファには許容待ち時間が下側閾値 Min よりも長いが上側閾値 Max よりも短いパケットが，最大バッファには許容待ち時間が上側閾値 Max よりも長いパケットが割り振られる．到着するパケットの待ち時間とその割合によって最適な下側閾値 Min と上側閾値 Max の値が異なると予想される．

IoT 環境のようにパケットサイズが小さく許容待ち時間の短いパケットの多い状況を考えやすくするため，利用されているアプリケーションによってパケットを大きく三種類に分類し，**表 1** に示した．

以下，ルータに到着するパケットの許容待ち時間は何らかの方法でわかると仮定する．また，どの許容待ち時間のパケットがどの程度割合で到着するかもわかっているとする．

4.3 下側閾値の決定手法

下側閾値 Min の決定方法として期待値を利用した方法と最頻値を利用した方法の二つを提案する．

4.3.1 期待値を利用した下側閾値の決定手法

データアグリゲータに到着するパケットの許容待ち時間の期待値をもとに下側閾値 Min の最適値を求めようとする手法である．

期待値は実際に到着するパケットの割合とそれぞれの待ち時間を考慮して計算を行う．許容待ち時間幅の最小値と最大値をそれぞれ，センサタイプは $sensor_{min}$ と $sensor_{max}$，リアルタイムタイプは $real_{min}$ と $real_{max}$，ファイル転送タイプは $file_{min}$ と $file_{max}$ とおく．パケットの各タイプの許容待ち時間の平均値は一様分布に従うとすると，すべてのパケットの許容待ち時間を用いて計算するかわりにその時間幅の平均値を用いて計算が可能である．

$$sensor_{avg} = \frac{sensor_{min} + sensor_{max}}{2} \quad (2)$$

$$real_{avg} = \frac{real_{min} + real_{max}}{2} \quad (3)$$

$$file_{avg} = \frac{file_{min} + file_{max}}{2} \quad (4)$$

これらの値を用いて期待値を計算する．パケットの種類別の割合をそれぞれ Γ_{sen}，Γ_{rea}，Γ_{fil} で表すと，以下の式で求められる．

$$Time_{exp} = \Gamma_{sen} \times sen_{avg} + \Gamma_{rea} \times rea_{avg} + \Gamma_{fil} \times fil_{avg} \quad (5)$$

4.3.2 最頻値を利用した下側閾値の決定手法

ルータに到着するパケットの許容待ち時間の最頻値をもとに下側閾値 Min の最適値を求めようとする手法である．

まず，最もよく到着する許容待ち時間の階級を判定する．本研究ではその階級をパケットの種類でわけるとする．最頻値はその階級値にあたる．パケットの許容待ち時間は設定された時間幅の中から一様分布で値が決定されるとき，その時間幅の最小値 $time_{min}$ と最大値 $Time_{max}$ を用いて以下の式で求められる．

$$Time_{mode} = \frac{time_{min} + time_{max}}{2} \quad (6)$$

4.4 上側閾値の決定手法

上述の二つの手法では $Time_{exp}$ や $Time_{mode}$ をバッファの下側閾値 Min として利用する．このとき，パケットは許容待ち時間で二つのグループに分けられることになる．ただしパケットタイプに偏りがあるとき，一つのバッファに割り振られるパケットの許容待ち時間の範囲が大きくなる．例えばセンサアプリケーションタイプパケットが多い環境で期待値が 80 μ 秒だった場合，片方のバッファに 50 μ 秒から 80 μ 秒のパケットが入る一方，もう一方のバッファには 80 μ 秒から 200 μ 秒のパケットが割り振られ，一つ目では 30 μ 秒の範囲なのに対して二つ目では 120 μ 秒の範囲になる．範囲が広くなるとパケットの許容待ち時間の超過が起こりやすくなるという欠点がある．これを解消するように上側閾値 Max を定める．

4.4.1 待機バッファの設定時間幅を重視した上側閾値の決定手法

到着頻度の高いバッファの待ち時間幅を重視し，最小バッファと中間バッファに設定する許容待ち時間の幅を同じにする手法である．到着するパケットの許容待ち時間が一様分布によって定まっている場合，この方法によって集

約前に待つバッファに入るパケットの数が均等になることが期待できる．したがって集約まで待たされるバッファに入るパケットの数を均等にすることでパケットの待ち時間を短くできると予想した．

最小バッファに入るパケットの許容待ち時間の最小値を $Time_{Min}$，最大値を $Time_{Max}$，求めたい上側閾値 Max は以下の式で求められる．

$$Max = Time_{Max} + (Time_{Max} - Time_{Min}) \quad (7)$$

$$= 2Time_{Max} - Time_{Min} \quad (8)$$

具体例をあげる．最小バッファに入るパケットが 50 μ 秒から 75 μ 秒だとすると，中間バッファには同じ 25 μ 秒の幅でパケットが割り振られるように 100 μ 秒を設定する．このとき，最小バッファには 50 μ 秒から 75 μ 秒の，中間バッファには 75 μ 秒から 100 μ 秒の，最大バッファには 100 μ 秒以上の許容待ち時間のパケットが割り振られるようになる．

4.4.2 到着パケットの許容待ち時間を重視した上側閾値の決定手法

一つのバッファに入るパケットの待ち時間の差を重視した時間設定方法である．一つのバッファに入るパケットの待ち時間の差を考慮する時間設定方法パケットを事前に割り振るときに，パケットの待ち時間にあまりに差がある場合，パケットの待ち時間が超過してしまう可能性が高い．特に短い許容待ち時間が定められているパケットの方が長い待ち時間を定められたものよりも影響が大きいと考えた．したがって待ち時間の差を考慮することによって一つのバッファで待つパケットの許容待ち時間がある程度近いパケットが同じバッファで待つことができるようになる．

この値を決めるときには最小バッファに入るパケットの量を決めている下側閾値 Min は定まっているので，到着するパケットの許容待ち時間の最大時間を $Time_{max}$ とすると，上側閾値 Max は以下のようにして求められる．

$$Max = \frac{Min + Time_{max}}{2} \quad (9)$$

具体例をあげる．最小バッファに入るパケットの許容待ち時間が 50 μ 秒から 75 μ 秒だとする．到着するパケットの許容待ち時間の最大値が 200 μ 秒だとわかっていたら，中間バッファと最大バッファに入るのは 75 μ 秒から 200 μ 秒の許容待ち時間のパケットなので式 9 より上側閾値 Max は 137.5 μ 秒となる．

4.5 確率分布を利用して下側閾値と上側閾値を決定する手法

本項では到着パケットの許容待ち時間の確率分布を利用して下側閾値 Min と上側閾値 Max の値を決定する手法を提案する．本手法では，三つのバッファに割り振られるパケットの個数が均等になるようにすることで，特定のバッファにパケットが集中してしまうことによって許容待ち時間を超過してしまう可能性を下げることを目的としている．

パケットの許容待ち時間を確率変数とした累積分布関数を考える．パケットの個数が三等分される時間を短い方から順に下側閾値 Min と上側閾値 Max として設定する．確率変数（許容待ち時間）X の累積分布関数を $F_X(x)$ とし，その X を求める逆関数を F^{-1} とすると，以下の式 10 と式 11 で下側閾値 Min と上側閾値 Max が求められる．

$$Min = F^{-1}(\frac{packetnum}{3}) \quad (10)$$

$$Max = F^{-1}(\frac{2 \times packetnum}{3}) \quad (11)$$

5. 評価

5.1 シミュレーション条件

本研究では IoT を意識したトラフィックを考えるため，小さいサイズで許容待ち時間の短いパケットが多いと仮定した．

パケットのルータへの到着は平均到着率 λ=1.1 のポアソン分布に従うとしてシミュレーションを行なった．表 1 に示したパケットの種類別のサイズと待ち時間の設定値は **表 2** に示す通りである．それぞれのパケットは許容待ち時間に幅を持たせてあるが，どの待ち時間のパケットが生成されるかは一様分布に従う．

以上の条件のもと，パケットを 5,000,000 個生成し，パケットの割り振りを行う．この作業を 1000 回行い，その平均値を結果として利用した．

5.2 評価項目

提案手法を四つの項目をもとに評価する．

パケット集約の利点のひとつにネットワーク全体のパケットの数を減らせることがあげられるのでパケット数を調べる．また，ヘッダの影響はパケットのサイズによって変化するので，集約したことによってどの程度のサイズになったかも調べる．IoT では大量のデータを処理し，それをもとに新しい情報を知ることができるがそのためにはリアルタイム性も重要な場面がある．したがってどの程度待たされるのかが利用者にとっては重要な指標である．ましてや遅延要求を守ることのできないシステムは利用不可能である．したがってどの程度遅延を超過してしまうか調べる必要がある．

表 2 パケットのシミュレーションパラメータ
Table 2 Simulation Settings of Packets

アプリケーション	データサイズ (byte)	許容待ち時間 (μ秒)	生起割合
センサ	40	50-100	0.88
リアルタイム	200	100-150	0.01
ファイル転送	1500	150-200	0.01

以下，提案手法の評価のための指標について詳しく説明する．

送信パケット数　送信パケット数は集約をした後送信したパケット数を数えたものである．送信パケット数が少ないほど集約効率が高いことを意味する．送信パケット数がなるべく少なくなるような時間設定ほどよい割り振り方といえる．

パケットサイズ　パケットサイズは集約して送信したパケットが平均的にどのサイズになったかを表したものである．パケットの集約は，送信可能な最大サイズに達するかパケットの残り許容待ち時間が 0 に達した時に終わるので，このパケットサイズが大きいほど効率的に集約できたことを示す．

待ち時間超過回数　待ち時間超過回数は，到着したパケットが集約して送信されるまでに設定されている許容待ち時間を超えてしまった回数である．この回数が多いと良いサービスとはいえない．

各時間設定の細かい違いを分析するために，バッファ別待ち時間超過回数の計測も行った．バッファ別待ち時間超過数はどの時間幅のパケットが入るバッファで許容待ち時間を超えてしまったかを表す数値である．

平均待ち時間　平均待ち時間は集約バッファにおいて，送信されるまでにどの程度待たされたかを表す値である．この値が短いほどパケットが待たされる時間が少なく，遅延が少ないといえる．

5.3　シミュレーション結果

本節では，提案した時間設定方法を第 5.2 節の項目で評価した結果を示す．また，結果の理由を調べるため，設定する時間を期待値や最頻値以外の値に変化させたときの挙動の結果も示す．

調べた時間設定を**表 3** に示した．提案手法との比較のため，工夫を行なっていない下側閾値 $Min = 100$，上側閾値 $Max = 150$ の設定と，期待値手法の式 5 を用いて得た下側閾値 $Min = 69$ に対して式 8 と式 9 を用いて得た二種類の上側閾値 $Max = 88$ と $Max = 134.5$，最頻値手法の式 6 を用いて求めた下側閾値 $Min = 75$ に対してそれぞれ式 8 と式 9 を用いて得た二種類の上側閾値 $Max = 100$ と $Max = 137.5$ の 5 種類である．

表 3　下側閾値 Min と上側閾値 Max の設定
Table 3　Settings of Min and Max

手法	Min(μ秒)	Max(μ秒)
工夫なし	100	150
期待値バッファ重視	69	88
期待値パケット重視	69	134.5
最頻値バッファ重視	75	100
最頻値パケット重視	75	137.5
確率分布	67	84

5.3.1　提案手法を利用した時間設定の評価

バッファの時間設定を変更すると送信パケット数，パケットサイズ，待ち時間超過回数，平均待ち時間がどのように変化するか調べた結果を**表 4** に示す．

どのバッファで待ち時間を超過したのかについて詳しく調べた結果を**表 5** に示す．

表 4 から，期待値バッファ重視手法と確率分布の結果と残りの四つの設定はそれぞれ似た傾向を示していることがわかる．特に顕著な差として，期待値バッファ重視手法と確率分布の設定では工夫なしの場合も含めた他の設定と比べて遅延超過合計数が多くなっていることがあげられる．特に最大バッファ遅延超過数が極端に多いが，最小バッファと中間バッファでの遅延超過数は他の手法よりも良い結果を示した．

残りの提案手法を利用した場合，利用しなかった場合と比べて最小バッファで発生する遅延超過回数が大幅に減少している．具体的には，工夫を行なっていない 100/150 の設定と比較して遅延超過回数が期待値パケット重視手法を利用した 69/134.5 の設定のときは約 13.4%減少，最頻値バッファ重視手法を利用した 75/100 の設定のときは約 11.0%減少，最頻手法 2 を利用した 75/137.5 の設定では約 19.9%も減少した．

提案手法を利用すると全体的に送信パケット数が減った．期待値パケット重視手法では約 5.2%の減少を，最頻値バッファ重視手法では約 6.5%の減少，最頻値パケット重視手法では約 7.3%の減少を示した．

パケットサイズについては，提案手法を利用することで利用しない場合に比べて大きくなった．具体的には工夫なしの場合と比べて期待値パケット重視手法では約 5.5%増加，最頻値バッファ重視手法では約 6.8%増加，最頻値パケット重視手法では約 7.8%増加した．

平均待ち時間は提案手法を利用することで長くなった．工夫をしていない 100/150 のときの値と比べて，期待値パケット重視手法では約 5.5%増加，最頻値バッファ重視手法では約 7.0%増加，最頻値パケット重視手法では約 7.9%増加した．

次に結果の傾向の似ている手法を比較するため，69/88 と 67/84 を除いた時間設定について表 5 の結果を**図 6** に示した．

図 6 からわかるように，提案手法を用いると最小バッファで生じる遅延の超過回数が著しく減少している．また，期待値を利用した 69/134.5 の時間設定と最頻値を利用した 75/137.5 の時間設定の結果はどちらも中間バッファでの遅延超過回数が最大であった．最頻値を利用してバッファの最大の値を 100 μ 秒に設定した 75/100 の時間設定では全てのバッファでおおよそ同じ回数遅延が超過していた．

表 4 時間設定の影響
Table 4 Effects of Time Settings

手法	時間設定 Min/Max	送信パケット数	パケットサイズ (byte)	遅延超過 合計数	平均待ち時間 (μ秒)
工夫なし	100/150	45691.054	6274.141	9863.133	21.625
期待値バッファ重視	69/88	106921.922	2657.844	70722.551	9.1280
期待値パケット重視	69/134.5	43328.396	6616.955	8538.014	22.810
最頻値バッファ重視	75/100	42700.881	6700.824	8775.379	23.144
最頻値パケット重視	75/137.5	42373.749	6766.238	7896.689	23.327
確率分布	67/84	94260.485	3020.571	58446.383	10.380

表 5 時間設定に対するバッファ種類別待ち時間超過回数
Table 5 Number of Times Unacceptable Delay Occurred

手法	時間設定 Min/Max	遅延超過 合計数	最小バッファ 遅延超過数	中間バッファ 遅延超過数	最大バッファ 遅延超過数
工夫なし	100/150	9863.133	9824.67	19.256	19.207
期待値バッファ重視	69/88	70722.551	617.083	617.243	69488.225
期待値パケット重視	69/134.5	8538.014	1041.003	7348.645	148.366
最頻値バッファ重視	75/100	8775.379	2676.387	2680.329	3418.663
最頻値パケット重視	75/137.5	7896.689	2197.887	5575.2	123.602
確率分布	67/84	58446.383	326.384	328.609	57791.39

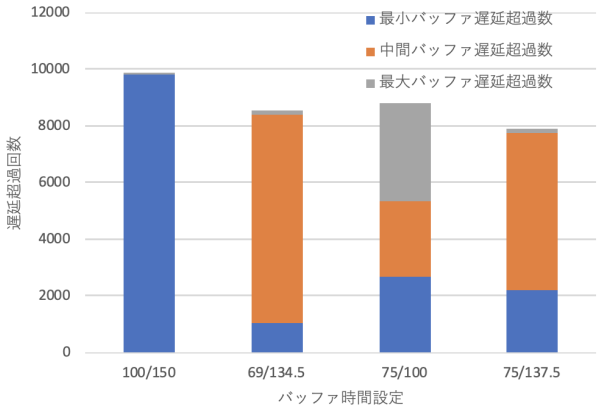

図 6 時間設定ごとの遅延超過を生じるバッファ比較
Fig. 6 Comparison of Number of Times Unacceptable Delay Occurred

表 6 その他の時間設定
Table 6 Other Time Settings

時間設定 Min/Max	送信 パケット数	パケットサイズ (byte)	待ち時間 超過回数	平均待ち時間 (μ秒)
100/125	45908.339	6244.181	10075.388	21.521
69/100	43497.536	6579.412	9144.679	22.719
75/125	40516.875	7076.937976	6297.636	24.404
75/150	45550.725	6293.297	10600.600	21.689

5.3.2 様々な時間設定における結果

表 4 で示した本提案で実験した値の設定方法のさらなる分析をするために様々な時間設定のもと実験を行った．その結果を**表 6** に示す．

6. 考察

本節では第 5.3 節の結果に基づいて考察を行う．

6.1 期待値バッファ重視手法と確率分布を利用した時の特徴

まず第 5.3.1 項の結果において遅延超過回数が特に多かった期待値バッファ重視手法と確率分布手法の時間設定について考察をおこなう．

遅延超過回数が増えた最大の原因は，最大バッファに到着するパケットの種類とその割合が偏っていることによると考えられる．69/88 の設定を例に考えると，全三種類のパケット，すなわち 88 μ秒から 200 μ秒までの 112 μ秒もの時間幅のパケットが到着している．さらに，88 μ秒から 100 μ秒のパケットの生起確率が 0.88 であるのに対して残りの 100 μ秒以上の許容待ち時間を設定してある二種類のパケットの生起確率がそれぞれ 0.01 であることが大きく影響していると思われる．このような設定では，待機バッファの先頭に許容時間の長いパケット到着している場合や連続している場合，そのバッファにはパケットが到着することがあっても集約バッファに送られるまでに時間が発生する．許容待ち時間が短いパケットが到着するとこの待ち時間の間に遅延超過してしまう．

例えば，200 μ秒待てるパケットが待機バッファの先頭にある状態で 88 μ秒の許容待ち時間のパケットが到着すると，先頭のパケットが 88 μ秒たってもまだ集約バッファに移動しない場合，遅延の超過が発生してしまう．短い時間しか待てないパケットが多数生起する状況においてはこのような遅延超過が多数発生してしまう．

したがって，バッファに設定したパケットの許容待ち時間の最小値と最大値の差が大きすぎたことと，その時間幅の中に大量に発生している短い許容待ち時間のパケットを

含んでしまったのが遅延超過回数の面での問題点であったと考えられる．

6.2 バッファの遅延超過回数について

表 5 で示した結果の中で最も着目すべき点は期待値パケット重視手法，最頻値バッファ重視手法，最頻値パケット重視手法を利用した時間設定では工夫をしなかった場合と比べて遅延の超過回数が減少した点である．IoT のサービスの利用者にとって，必要なデータ処理が時間内に行われることは最重要である．したがって遅延が超過してしまう可能性を下げることができた点で本手法は評価できる．

次に，どのバッファで遅延が超過したかを比較した図 6 をもとに考察をおこなう．提案手法を利用した場合，利用しなかった場合と比べて最小バッファで発生する遅延超過回数が大幅に減少している．最も到着頻度の高い遅延許容時間をもつパケットをただ一つのバッファに入れるのではなく，複数のバッファに分散して待機させるような設定にすることの効果があったといえる．その反面，中間バッファと最大バッファの遅延超過回数が増加しており，トレードオフがある．ただし，中間バッファと最大バッファの時間設定次第で全体の数を減らすことが可能である．

6.3 バッファの最大の値の決定法の影響

本項ではバッファのパケット割り振り時間に対して様々な変化をさせた時にどのような結果をもたらすかを考察する．特に，図 5 の上側閾値 Max がパケット集約にどのような影響を与えているかの結果と考察を述べる．下側閾値 Min はシミュレーションで利用した期待値，最頻値，確率分布の手法で求めた値とする．

6.3.1 パケットの発生頻度と時間設定方法の関係

表 4 と図 6 には時間設定のうち一つの値だけ異なるようなデータの組み合わせがあげられている．データの比較を行うため**表 7** にまとめた結果を示す．期待値，最頻値を利用したときの最大値の設定時間を変化させた結果の比較をあらわしている．

まず，100/150 の時間設定と 100/125 の時間設定を比較することで，到着頻度が低いが時間幅の広いパケットの分割方法を変えた時の影響を調べた．

本研究のシミュレーションの設定ではリアルタイムアプリケーションパケットの許容待ち時間が 100 μ秒から 150 μ秒の間の値を，ファイル転送アプリケーションパケットの許容待ち時間が 150 μ秒から 200 μ秒の間の値をとる．また，それぞれのパケットの割合が 0.01 ずつなので極めて発生割合が小さいパケットを異なるバッファで待機させる意味の有無がわかる．表 7 より，送信パケット数は約 0.48%，パケットサイズは約 0.48%，待ち時間超過回数は約 2.2%の差，平均待ち時間は約 0.48%の差しか見られなかった．このことから，どの評価項目についても二つの差はほとんど見られない．したがってパケットの時間幅が大きくなるような時間設定だとしてもその到着頻度が低い時は送信パケット数，パケットサイズ，待ち時間超過回数，平均待ち時間全てほとんど影響はないと考えられる．

6.3.2 期待値と最頻値の利用に対してバッファの最大の値の与える影響

期待値手法を用いて求めた下側閾値 Min の値 69 μ秒と，最頻値手法を用いて求めた下側閾値 Min の値 75 μ秒を利用したときのそれぞれに対して上側閾値 Max を，式 8 や式 9 とパケットの待ち時間の閾値である 100 μ秒と 150 μ秒の 4 種類に変えたときの各評価値の様子を以下に示した．期待値手法と最頻値手法のそれぞれについて，送信パケット数の変化は**図 7** と**図 8** に，平均送信パケットサイズの変化は**図 9** と**図 10** に，平均待ち時間の変化を**図 11** と**図 12** に示した．

図 7 と図 8 からわかるように，上側閾値 Max を小さい値から大きい値へと増加させていく過程で，送信パケット数はあるところまで減少をし，ある点から増加している．また，期待値を利用した場合と最頻値を利用した場合とを比較するとどの最大値設定の場合も最頻値を下側閾値 Min として利用した場合の方が送信パケット数が抑えられてお

表 7 時間設定の詳細条件による比較
Table 7 Comparison Among Detailed Time Settings

時間設定	送信パケット数	パケットサイズ (byte)	待ち時間超過回数	平均待ち時間 (μ秒)
100/150	45691.054	6274.141	9863.133	21.625
100/125	45908.339	6244.181	10075.388	21.521
69/88	106921.922	2657.844	70722.551	9.1280
69/100	43497.536	6579.412	9144.679	22.719
69/134.5	43328.396	6616.955	8538.014	22.810
69/150	46544.787	6158.087	11513.772	21.222
75/100	42700.881	6700.824	8775.379	23.144
75/125	40516.875	7076.938	6297.636	24.404
75/137.5	42373.749	6766.238	7896.689	23.327
75/150	45550.725	6293.297	10600.600	21.689

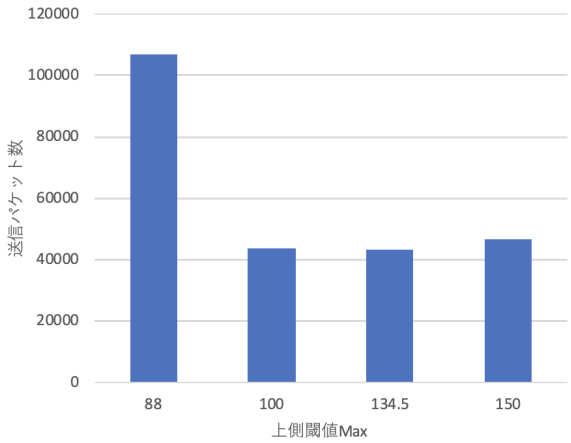

図 7 期待値手法における最大時間設定と送信パケット数 (Min=69 μ秒)
Fig. 7 Time "Max" and Number of Sent Packets (Min=69 μ sec)

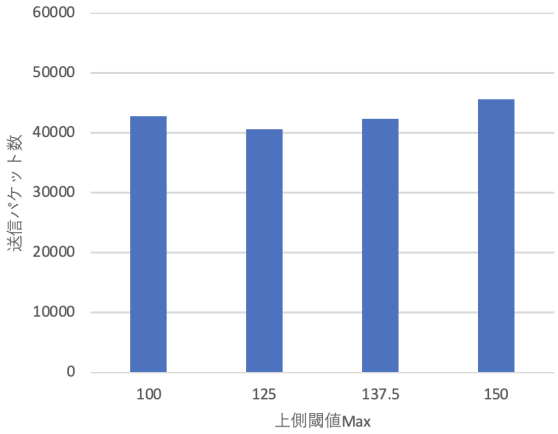

図 8 最頻値手法における最大時間設定と送信パケット数 (Min=75 μ秒)

Fig. 8 Time "Max" and Number of Sent Packets (Min=75 μ sec)

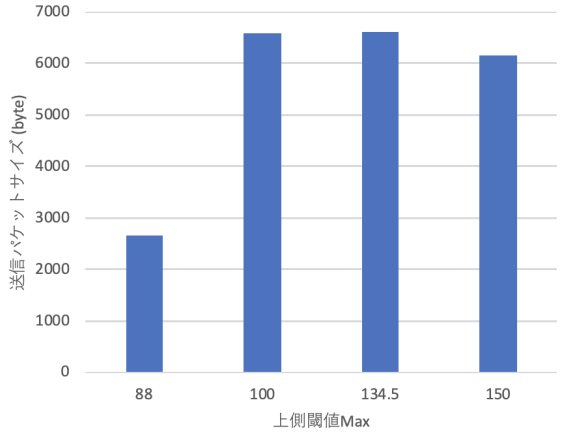

図 9 期待値手法における最大時間設定と送信パケットサイズ (Min=69 μ秒)

Fig. 9 Time "Max" and Average Size of Sent Packets (Min=69 μ sec)

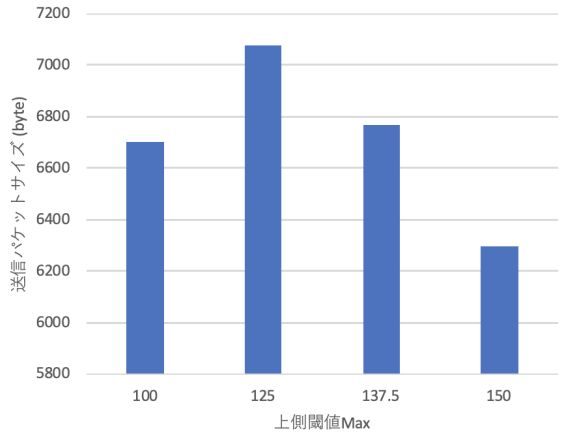

図 10 最頻値手法における最大時間設定と送信パケットサイズ (Min=75 μ秒)

Fig. 10 Time "Max" and Average Size of Sent Packets (Min=75 μ秒)

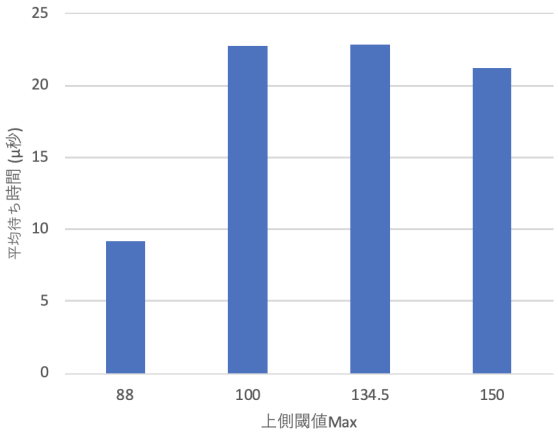

図 11 期待値手法における最大時間設定と平均待ち時間 (Min=69 μ秒)

Fig. 11 Time "Max" and Average Waiting Time (Min=69 μ sec)

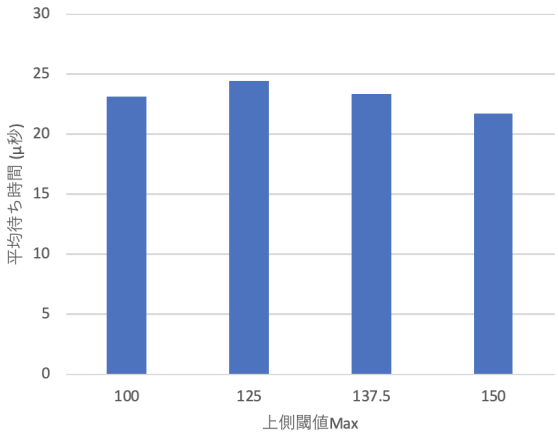

図 12 最頻値手法における最大時間設定と平均待ち時間 (Min=75 μ秒)

Fig. 12 Time "Max" and Average Waiting Time (Min=75 μ秒)

り，集約効率がより高いことがわかる．

図 9 と図 10 からわかるように，パケットの平均送信サイズは，上側閾値 Max を小さい値から大きい値に変えていく過程である値まではサイズも増加するが，一定値以上になると減少に転じている．

図 11 と図 12 からわかるように，パケットの平均待ち時間は上側閾値 Max が小さい時は平均待ち時間は短く，上側閾値 Max の増加とともに増加していくが，ある点を超えると減少していっている．

これらのことから，まず，送信パケット数，平均送信パケットサイズ，平均待ち時間が互いに関係していることがわかる．上側閾値とパケットサイズの関係を示した図 9 と図 10，待ち時間との関係を示した図 11 と図 12 を比較すると同じ変化の仕方をしていることからこの平均送信パケットサイズと平均待ち時間は比例の関係にあると考えられる．この理由として，集約にかける時間が増えれば増える

ほどサイズが増加しているからだと思われる．一方で，送信パケット数とこの二つは反比例の関係にある．これは，式 12 で示したように，到着しているデータの総量が変わらないことが理由として挙げられる．

$$DataSize = PacketNum \times PacketSize \quad (12)$$

バッファの上側閾値 Max を変化させることで，送信パケット数，平均送信サイズ，平均待ち時間を変化させることができることがわかった．特に，中間バッファと最大バッファのパケット割り振り時間を同じ時間幅に設定する式 9 を使用した方法のとき，期待値を利用した場合も最頻値を利用した場合もともに送信パケット数と平均送信サイズの観点でよい数値を示した．一方で，この手法を用いると平均待ち時間が伸びてしまうという欠点もあった．

6.3.3 提案手法におけるトレードオフ

表 4 と表 6 のデータをもとに作成した送信パケット数とパケットサイズの関係のグラフを**図 13** に，送信パケット数と平均待ち時間の関係のグラフを**図 14** に示した．

図 13 は時間設定を変更していったときにパケットの送信数が少ない時は平均的なサイズが大きく，逆に多い時は小さくなっていることを示している．

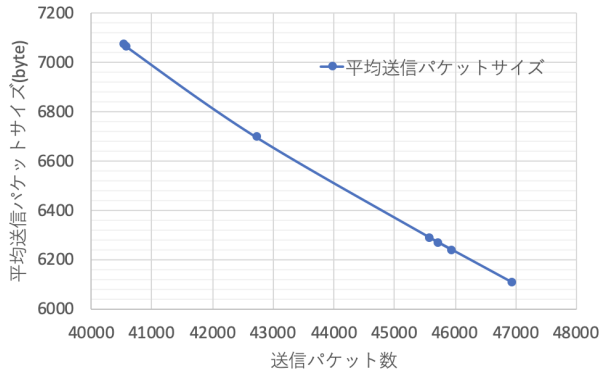

図 13 送信パケット数ーパケットサイズ

Fig. 13 Correlation between Number of Sent Packets and Packet Size

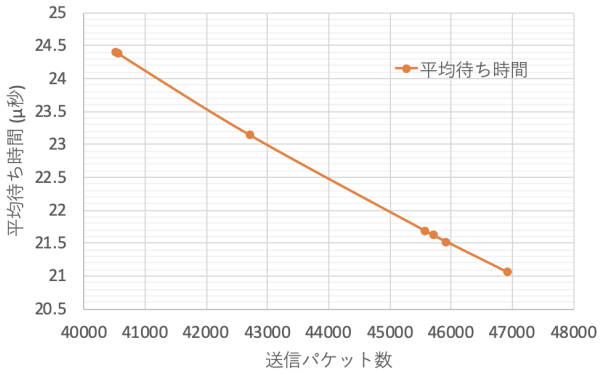

図 14 送信パケット数ーパケット平均待ち時間

Fig. 14 Correlation between Number of Sent Packets and Average Waiting Time

また，図 14 は様々なバッファの時間設定に対してパケットの送信数と集約にかけられている時間がトレードオフの関係にあることを示している．まとめるとこの手法ではパケットの送信数を抑える，すなわち集約の効率をあげると集約にかける時間が増加するが一つ一つのパケットのサイズを大きくすることができることがわかる．

トレードオフの関係は期待値を利用しても最頻値を利用しても生じた．本シミュレーション結果より最頻値を利用したときのほうが評価値を改善できたといえる．しかしトラフィックを占めるパケットタイプの割合によっては最頻値よりも期待値を利用した方が良い結果になる可能性もある．その点についてはさらなる調査が必要である．

7. おわりに

本稿では，既存のパケット集約手法を利用した際，到着するパケットの種類が偏ったトラフィックのもとで発生する問題点を指摘し，その改善方法を提案した．改善方法として，パケットの許容待ち時間に応じて適切にバッファへのパケットの割り振り時間を設定できるようにした．

提案手法を利用しない場合と比較し，集約の効率を向上させ，IoT アプリケーションの利用者にとって非常に重要な要素である許容待ち時間を超過してしまう確率をさげることができた．なかでも最頻値を利用した手法が IoT 環境ではもっとも有効な方法であった．一方でパケットの平均的な待ち時間が増加してしまうという課題も残った．

本研究では期待値と最頻値と確率分布を用いたパケットの割り振り方を提案し，工夫をおこなわないときよりも良い結果となることを示した．しかし見つけた値が最適値とは限らず，その最適値を探す方法として提案した確率分布を用いた手法は最善策ではなかった．その理由を調べるために様々な時間設定に変更をして調査をおこない，パケットの許容待ち時間と，それぞれの到着割合が大きな影響を与えることがわかった．今後の課題として，提案した三つの手法のうち，この二つのパラメータを元に時間の設定を決めやすい確率分布の手法をベースに手法の発展をさせていきたい．

参考文献

[1] 総務省. 平成 30 年版 情報通信白書. http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd111200.html. 2019/1/20 閲覧.

[2] 竹下隆史, 松山公保, 荒井透, and 苅田幸雄. **マスタリング** *TCP/IP* **入門編 第５版**, chapter 2. 株式会社オーム社, 2017.

[3] Jianhua Deng and Mark Davis. An adaptive packet aggregation algorithm for wireless networks. In *Wireless Communications & Signal Processing (WCSP), 2013 International Conference on*, pages 1–6. IEEE, 2013.

[4] Yuichi Yasuda, Ryo Nakamura, and Hiroyuki Ohsaki. A probabilistic interest packet aggregation for content-centric networking. In *2018 IEEE 42nd Annual Com-*

puter Software and Applications Conference (COMPSAC), pages 783–788. IEEE, 2018.

[5] Tsan-Pin Wang and Yu-Chun Chen. Adaptive packet aggregation for header compression in vehicular wireless networks. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, pages 935–939. IEEE, 2011.

[6] Tetsuya Yokotani, Akihiro Shimuzu, Yuya Sasaki, and Hiroaki Mukai. Proposals for packet processing and performance evaluation of iot devices. In *Electronics, Communications and Computers (JAC-ECC), 2017 Japan-Africa Conference on*, pages 5–8. IEEE, 2017.

[7] Yoshihide Nomura, Kazuo Mori, Katsuhiro Naito, and Hideo Kobayashi. High efficient packet aggregation scheme for multi-rate and voip packet transmissions in next generation mu-mimo wlans. In *Advanced Technologies for Communications (ATC), 2014 International Conference on*, pages 517–521. IEEE, 2014.

[8] Hajer Tounsi, Laurent Toutain, and Farouk Kamoun. Small packets aggregation in an ip domain. In *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, pages 708–713. IEEE, 2001.

[9] Anan Sawabe, Kazuya Tsukamoto, and Yuji Oie. Qos-aware packet chunking schemes for m2m cloud services. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 166–173. IEEE, 2014.