DEIM Forum 2017 H2-2

多数台 Android 端末が接続する無線 LAN 環境における距離を考慮した通信制御手法によるバッテリ消費の低減

小柳　文乃† 　山口　実靖†† 　小口　正人†

† お茶の水女子大学　〒 112-8610 東京都文京区大塚 2-1-1
†† 工学院大学　〒 163-8677 新宿区西新宿 1-24-2
E-mail: †{ayano,oguchi}@ogl.is.ocha.ac.jp, ††sane@cc.kogakuin.ac.jp

あらまし　近年スマートフォン端末は爆発的に普及しており，我々の生活に無くてはならないものとなりつつある．そのためスマートフォンのバッテリの持ち時間は非常に重要な課題である．本研究では，スマートフォン，特に Android 端末において，複数台の端末が同時に 1 つのアクセスポイントに無線接続し通信する劣悪な環境における性能を考慮し，アクセスポイント周りの通信の混雑具合や，距離による通信性能について調査する．そして，状況に応じた通信方法の提案を行うことで通信制御を行わない際より大幅なバッテリ消費の削減に成功した．

キーワード　スマートフォン，Android，省電力，無線 LAN

Ayano KOYANAGI†, Saneyasu YAMAGUCHI††, and Masato OGUCHI†

† Ochanomizu University
2-1-1 Otsuka, Bunkyou-ku, Tokyo, 112-8610 JAPAN
†† Kogakuin University
1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo,163-8677, Japan
E-mail: †{ayano,oguchi}@ogl.is.ocha.ac.jp, ††sane@cc.kogakuin.ac.jp

1. はじめに

近年スマートフォン端末は爆発的に普及しており，我々の生活に無くてはならないものとなりつつある．そのため，スマートフォンのバッテリの持ち時間の改善は非常に重大な課題となっている．本研究では，代表的なスマートフォンの一つである Android 端末における通信機能に着目し，そのバッテリ消費について考察する．

具体的には，複数台の Android 端末が同時に 1 台のアクセスポイント (AP) に無線 LAN 接続通信する際の性能を考慮し，AP 周りの混雑や，距離による通信制御について調査する．そして，状況に応じた通信制御手法の提案とバッテリ性能の考察を行う．

2. 研究背景

2.1 Android OS

本研究では, 近年大変注目されている Android 端末を用いる．Android 端末とは, Google 社を中心に開発されている Android OS を搭載したスマートフォンであり, オープンソースで提供されているためキャリア間の制約がなく, 様々なデバイスに自由に応用することができる．そのような背景から, スマートフォン OS の中 でのシェア率も年々上昇しており，今後ますます需要が増加すると考えられている．(2016 年第 3 四半期では世界で 86.8 %のシェア) [1].

2.2 スマートフォンのバッテリ消費

スマートフォンのバッテリ消費の原因は主に「ディスプレイ」「通信機能」「便利機能」の三点があげられる．

ディスプレイの明るさや画面の回転などは，ユーザの好みで自由にカスタマイズするものであり，バッテリの持ちのためにスマートフォンが使いにくくなってしまうことは好ましくない．しかし，二つ目の「通信機能」については，効率の悪い通信をユーザ個人で改善することは難しく，また，効率の悪い通信によるバッテリ消費を減らすことはユーザのためになる．そこで本研究では通信，特に多数台の端末が同時に通信する劣悪な環境でのバッテリ性能に着目する．

2.3 無線 LAN 通信

本研究では通信方法として，研究室にアクセスポイント (AP) を立てその AP に無線 LAN 接続通信させた．近年公共無線 LAN の普及が急がれている．日本では 3G や LTE の全国普及が進んでいることもあり，無料 Wi-Fi の利用機会が少ないものと考えられている．そのため他国と比較すると普及が遅れており，

外国人観光客への「おもてなし」という観点から，現在政府も Wi-Fi 環境の整備に力を入れている．以上の背景から，今後ますます無線 LAN 環境の整備は進んでいき，私たちの暮らしの中での Wi-Fi による通信の機会が増加していくと考えられる [2].

2.4 関 連 研 究

本節にて，スマートフォン端末の省電力化に関する関連研究を挙げる．

[3] はユーザがアプリケーションを実際に使用している時に情報を収集し，その情報から消費電力を見積もることで，省電力効果シュミレーションを行った. [4] は CPU, 液晶などの端末リソース毎の消費電流量や， Web ブラウザで 様々な操作をした際の消費電流量を実測し，Android OS における操作とバッテリ消費の関係を調査した. [3] [4] は実際のユーザの操作を想定し調査をすることで成果を得られたが，アプリケーションは新しいものが次々とリリースされ流行は変化し続けており，またスマートフォンの使い方は老若男女，住む土地や職業などで全く異なると考えられる．そこで本研究では，スマートフォンユーザ誰しもが行う通信について調査をする．

[5] は Wi-Fi 接続時における，スマートフォンの消費電力とスループットのトレードオフの関係について述べている．大量のパケットが MAC 層で損失した際，端末は消費電力が急増することでその損失を検知できる．しかし，損失が多くなればなるほどその分消費電力は多くなり，またこのような時に最大スループットが得られても，送信に失敗しているので無駄なバッテリを消費していると言える．このような無駄を防ぐ為，Roy Friedman らは最大スループットを保ったまま， TCP フローを停止するような受信バッファサイズを設定するための方法を提案し，実際に機能することを実証した．[6] しかしこの設定は，各デバイスの特定のハードウェアとネットワーク内の負荷に依存し，したがって自己適応するための動的メカニズムが必要となることが課題である．

[6] はスマートフォンのハードウェア・サブシステムの全ての電力使用量を考慮したシステムを実装し調査した．Wi-Fi の実験において，消費電力は送受信したパケット量に依存することを示したが，Wi-Fi の AP との接続を求めるための消費電力は考慮されていない．

以上より，本研究では Wi-Fi 接続時に通信性能を保ったまま，AP との接続も踏まえた上でバッテリ省電力を目指す．

3. 実験環境と評価方法

本章では実験環境における基本性能測定について説明する．

3.1 実 験 環 境

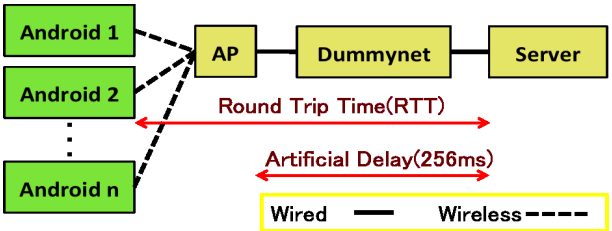

図 1　実験トポロジ

実験は図 1 で示す様にサーバ，ダミーネット，AP は有線で繋ぎ，Android 端末を無線 LAN で AP に接続させる．バッテリ残量，通信量を記録する事ができるように，カーネルにコードを加える改変を行った Android 端末を用い，表 1 で示した環境で実験を行った．スリープ状態になることを防ぐため，端末がスリープすることを防ぐ自作のアプリケーションを起動させ，その他の操作は一切行わない．iperf を用いて Android 端末を AP に 30 分間接続し通信させる．

表 1　実 験 環 境

Android	Model number	nexus S
	Firmware version	4.1.1
	Baseband version	I9023XXKD1
	Kernel version	3.0.31-ai
	Build number	JRO03L
Server	OS	Ubuntu 14.04(64bit)/Linux3.0.1
	CPU	Intel(R)Core 2Quad CPU Q8400
	Main Memory	7.8GiB
AP	Model	MZK-MF300N(Planet)
	Communication system	IEEE 802.11g

3.2 評 価 方 法

評価式は式 1 で表す．本研究では，バッテリ消費の大小やデータ送信量の大小をただ単純に比較するのではなく，少ないバッテリ消費でいかにデータ送信量を大きくできるかを評価する．この評価方法によって，例えばデータ送信量が同じ場合でもバッテリ消費が小さい方が評価は高くなる．

$$\text{バッテリ消費 1\%あたりの送信量} = \frac{\text{30 分間のデータ送信量 } (MBytes)}{\text{30 分間のバッテリ消費量 } (\%)} \quad (1)$$

4. 基本性能測定

本章では実験環境における基本性能測定について説明する．

4.1 アクセスポイントに接続する台数変化

図 2 は，AP に接続する端末台数がバッテリ消費と送信量に与える影響を示す．なお，AP と Android 端末の距離は 0.3m とする．

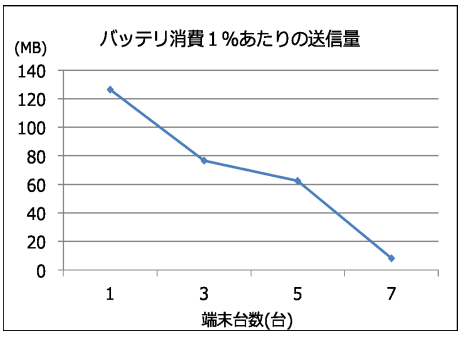

図 2　接続する端末台数が与える影響

図より，端末台数が増加するとバッテリ消費 1 %あたりの送信量は減少することがわかる．

4.2 アクセスポイントと端末の距離変化

図 3 は，AP と Android 端末の距離がバッテリ消費と送信量に与える影響を示す．

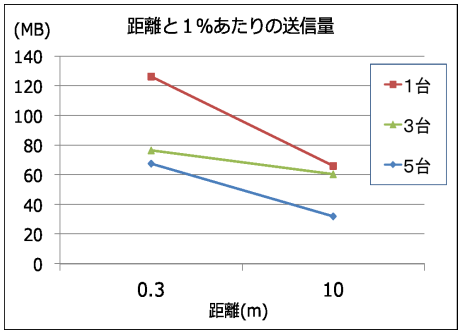

図 3　AP と端末の距離が与える影響

図より，端末台数に関わらず，AP と端末の距離が遠くなるとバッテリ消費 1 %あたりの送信量は減少することがわかる．

5. 先 行 研 究

5.1 カーネルモニタリングツール

カーネル内部の処理は，バックグラウンドで進められているため，通常ユーザ空間からその 処理を監視することはできない．近年，汎用 PC 向け Linux ディストリビューションにおいては， TCP Probe が利用できる場合もあるが，Android においてはサポートされていない．そこで本研究では，カーネル内部の通信処理を解析するために，カーネルモニタを利用する．カーネルモニタとは，Linux システムのカーネル内部の処理を解析する汎用 PC 向けシステムツールである．既存研究 [7] は，既にカーネルモニタの Android への組込みに成功しており，同様の方法で導入を行った端末を本研究においても利用する．図 4 にその概要を示す．カーネルモニタは，通信時におけるカーネル内部のパラメータ値の変化を記録することが可能である．カーネル内部の TCP ソースにモニタ関数を挿入し再コンパイルすることで，輻輳ウィンドウ値やソケットバッファの キュー長，各種エラーイベントの発生タイミングなどの TCP パラメータを見ることができる．このツールの解析パラメータに RTT を加えることで，本システムにおいてもカーネル内の RTT 値をリアルタイムに解析することができる．

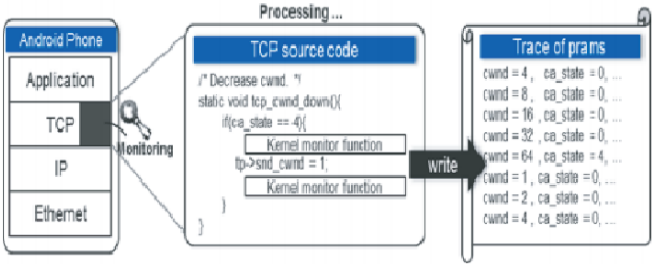

図 4　カーネルモニタの概要

5.2 通信制御ミドルウェア

先行研究で開発された輻輳制御ミドルウェア (MW) [8] は，同一 AP に接続した端末間でお互いの接続状況を把握し，その接続台数によって混み具合を予測し，輻輳ウィンドウ値の上限と下限を設定する．この通信制御手法により可用帯域を公平に分け合うことで，全体の通信速度と公平性の向上に成功している [9]．

5.3 通信制御ミドルウェアの導入

前節で説明した通信制御 MW は，バッテリ性能の向上にも効果が見られるのか調査した．図 5 は AP と端末の距離が非常に近い場合 (0.3m) の，バッテリ消費 1%あたりのデータ送信量を表す．図より，通信制御手法は Android 端末の台数が増加した場合も効果があるといえる．

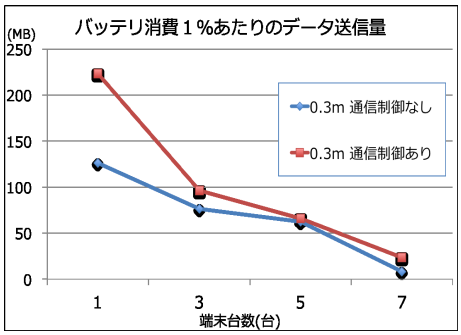

図 5　距離と通信成功率

同様に，AP と端末の距離を遠くして (10m) 測定した結果を図 6 に示す．図より，距離が離れると通信制御をしていない方がバッテリ消費の効率は良くなっていることが確認できた．

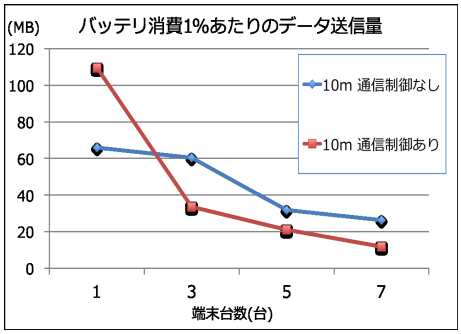

図 6　距離と通信成功率

この結果より，AP と端末の距離が近い場合でのみ先行研究の通信制御手法は有効であり，距離に応じた制御が必要であるといえる．

5.4 通信成功率

図 4 は，AP と Android 端末の距離が，通信の成功率にどのような影響を与えるのかを示す．なお，端末台数は 5 台とする．

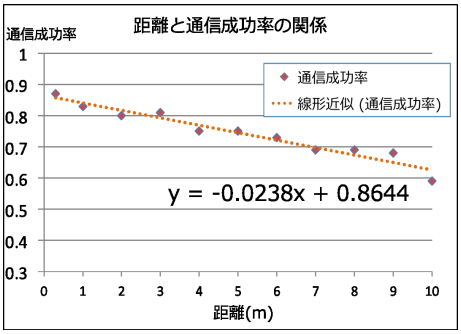

図 7　距離と通信成功率

図より，距離と通信成功率は比例の関係であるといえ，式 2 で表すことができる．

$$通信成功率 = -0.0238 \times 距離 + 0.8644 \qquad (2)$$

以上の基本性能測定の結果より，AP に同時接続する端末台数が多いほど，また，AP と端末の距離が離れるほど，バッテリ消費 1%あたりのデータ送信量は悪化することがいえる．また，AP と端末の距離は，エラーのない通信の成功率と比例関係であるといえる．

6. 提 案 手 法

6.1 提案手法の概要

前章より，先行研究で開発された輻輳制御 MW を本研究に導入したところ，AP と通信端末の距離を考慮しなくては，かえってバッテリ消費と送信量の比を悪化させてしまう場合があることが確認された．そこで本研究では，距離に応じた柔軟な制御手法を提案する．なお，端末台数は 5 台とする．

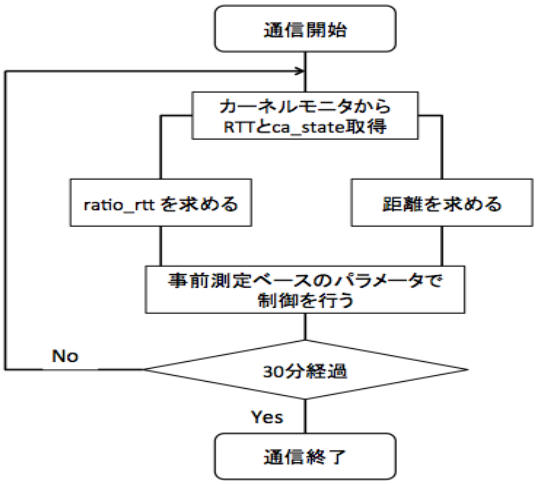

図 8　フローチャート

本提案手法の大まかな流れは図 8 で表す．先行研究と同様に，カーネルモニタで通信中 RTT と CA_STATE を取得し，その値から本研究では距離に応じた適切な制御をすることで，バッテリ消費とデータ送信量の比の向上を目指す．ratio_rtt と距離を求めるステップは，以下で説明する．まず，ratio_rtt の求め方について図 9 で説明する．

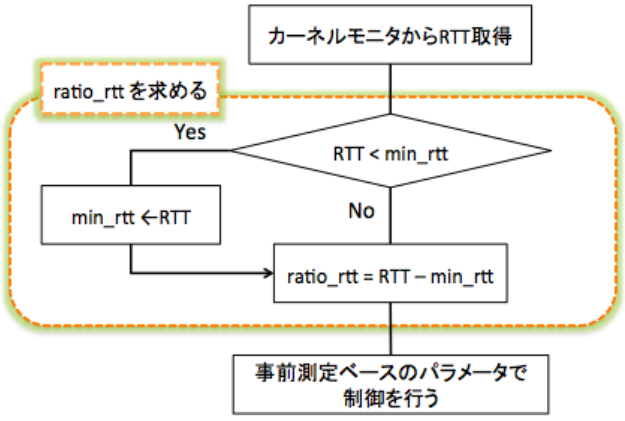

図 9　ratio_rtt の求め方

ratio_rtt とは，RTT の増減の比率を表している．通信中 RTT とその最小値 (min_rtt) を常に取得する．min_rtt は通信中で最も小さい RTT を上書きしていくことで値を更新する．取得した値をもとに，式 3 を用いて増減の比率 (ratio_rtt) を求める．

$$ratio\_rtt = RTT - min\_rtt \qquad (3)$$

この ratio_rtt の値でトラフィックが混雑しているかを判断し，輻輳ウィンドウ値を補正するフェーズに切り替わる．

次に，距離の求め方について図 10 で説明する．

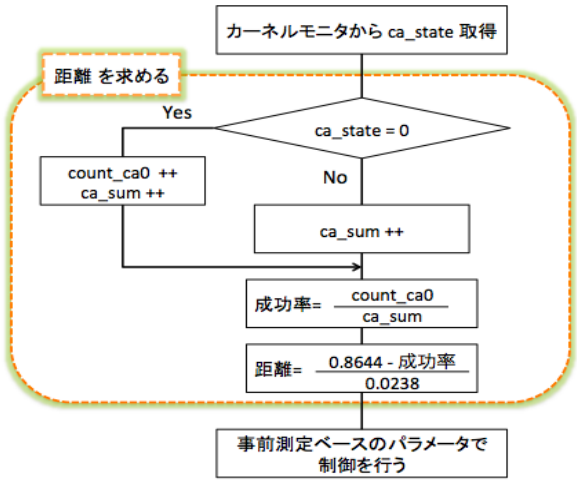

図 10　距離の求め方

この図で表している count_ca0 は CA_STATE が 0 の総計，ca_sum は CA_STATE0〜4 の総計を指す．また，図示していないが count_ca0 と ca_cum は初期値を 0 と設定する．

カーネルモニタから CA_STATE を取得し，0 か否かで場合分けする．0 の場合は count_ca0 と ca_sum の両方を＋ 1 し，0 でない場合は ca_sum のみ＋ 1 とカウントする．そして count_ca0 と ca_sum から通信成功率を求める．通信成功率が求まれば，式

2 より，通信端末が AP からどれだけ離れているのか距離が算出できる．

以上の方法により算出した ratio_rtt の値と距離を元に，本提案手法では距離に応じた制御を行う．

AP と端末の距離が 0.3m と 10m の時，輻輳の最大値と最小値を表 2 のように設定した際の評価が最も高かった．この事前測定ベースのパラメータの結果を元に，3m，5m，7m のパラメータの max，min を式 4, 5 のように設定した．3m，5m，7m のパラメータの max，min は表 3 で示す．

$$Xm\_max = \frac{10m\_max - 0.3m\_max}{10 - 0.3} \times (X - 0.3) + 0.3m\_max \quad (4)$$

$$Xm\_min = \frac{10m\_min - 0.3m\_min}{10 - 0.3} \times (X - 0.3) + 0.3m\_min \quad (5)$$

表 2　基準とするパラメータの値

	0.3m		10m	
	max	min	max	min
0 <= ratio_rtt < 51	330	330	330	300
51 <= ratio_rtt < 106	330	300	300	200
106 <= ratio_rtt < 161	330	200	250	150
161 <=ratio_rtt < 331	330	200	250	100
331 <= ratio_rtt	330	100	200	50

表 3　それぞれの距離における最適なパラメータの値

	3m		5m		7m	
	max	min	max	min	max	min
0 <= ratio_rtt < 51	330.0	321.6	330.0	315.5	330.0	309.3
51 <= ratio_rtt < 106	321.6	272.2	315.5	251.5	309.3	230.9
106 <= ratio_rtt < 161	307.7	222.2	291.2	201.5	274.7	180.9
161 <= ratio_rtt < 331	307.7	172.2	291.2	151.5	274.7	130.9
331 <= ratio_rtt	293.8	86.1	267.0	75.8	240.2	65.5

先行研究と同様に，通信中 RTT とその最小値 (min_rtt) を常に取得する．min_rtt は通信中で最も小さい RTT を上書きしていくことで値を更新する．取得した値をもとに，式 3 を用いて増減の比率 (ratio_rtt) を求める．

$$ratio\_rtt = RTT - min\_rtt \quad (6)$$

本研究では，この事前測定ベースのパラメータ値を用いて，算出した距離が 0〜2m の場合 0.3m の，2〜4m の場合 3m の，4〜6m の場合 5m の，6〜8m の場合 7m の，8m〜の場合 10m の表 2，3 で示したパラメータ値を設定をする．

この提案手法により，AP と通信端末の距離を人手に設定せずに自動で MW が距離を推測し，その距離に最適なパラメータを設定し通信することで，バッテリ消費あたりのデータ送信量の向上を目指す．

6.2　性 能 評 価

図 11 は通信制御手法オフの場合と，それぞれの距離における最適パラメータを設定した場合と，距離を自動算出し制御する MW を用いた提案手法の測定結果である．図より，提案手法は距離に関係なく通信制御を行わない場合より通信効率を良くすることができた．

しかし，最適パラメータを設定した結果より性能が悪化していることから，距離を算出する過程で誤差が生じていると考えられる．

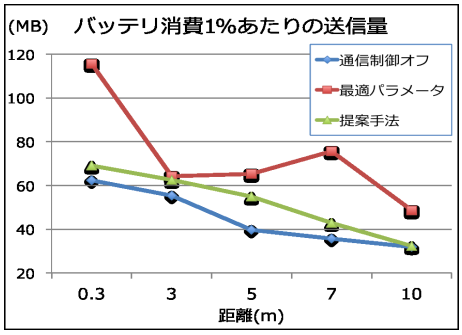

図 11　提案手法の実験結果

6.3　距離の算出結果

第 6.2 章より，本提案手法は距離の自動判別を行うことで，距離が変化する毎に人手に設定をせずとも，柔軟に対応可能な通信制御を実現できることが，通信制御オフの状態より性能向上していることから確認できた．しかし，手動で各距離における事前測定ベースのパラメータを設定した場合よりも，性能が悪化してしまうことも確認できる．これは，距離の判定の誤差によるものだと考えられる．図 12 は提案手法を各距離で測定した際の，MW により算出された距離を示す．

結果より，実際の距離と算出された距離にかなりの誤差があることがわかる．このことから，適切な制御が行われておらず，その分事前測定ベースのパラメータを設定した場合に性能が負けてしまったといえる．また距離の算出方法は式 2 で表しているように，通信成功率が悪くなればなるほど長距離の通信に設定されてしまうことから，距離が急上昇している通信開始から 20 分間はエラーの多発する，非常に効率の悪い通信が行われているといえる．しかし，時間が経つにつれて算出された距離は本来の正しい値に収束していくことが確認できる．よって，距離の算出方法自体は有効であるが，実験時間が 30 分と短く，最初の 20 分は通信がなかなか安定しないことが事前測定ベースのパラメータに大きく負けてしまう原因だと考えられる．以上のことより，測定時間を長くすることで提案手法の有効性が期待できる．

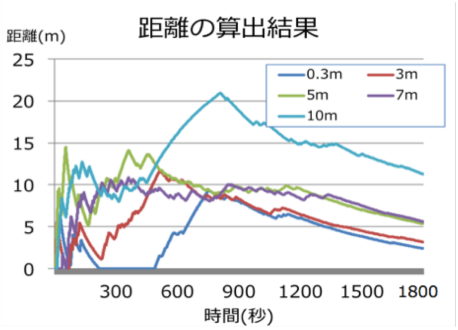

図 12　時間経過における提案 MW の距離の算出結果

6.4　測定時間を変更した場合の評価

第 6.2 章より，提案手法はデフォルトの通信制御オフの状態よりも，バッテリ消費 1%あたりのデータ送信量が大きくなることから有効であると確認できた．しかし，各距離の最適なパラ

メータを設定する事前測定ベースのパラメータ手法に近付かない．これは提案手法にとって測定時間が短いため，本来の性能を十分に出せていないことが考えられる．そこで，実験時間を 30 分から 1 時間に変更し，0〜1800 秒，600〜2400 秒，1200〜3000 秒，1800〜3600 秒と，30 分区切りで性能評価を行う．30 分で区切り評価を行うのは，これまでの測定時間は全て 30 分であり，評価をそろえるためである．

6.4.1 実験概要

第 3 章で説明した実験環境のもと，測定時間を 60 分と変更し測定を行った．また，時間変化とともに通信の状態が変わることが確認できたため，その時間にあった最適な制御を行うため，パラメータの設定を 10 分毎に変更した．具体的には，第 6.1 章で用いたパラメータは，30 分間測定した結果最も良い結果を示した MW のパラメータを用いたが，より時間経過に合ったパラメータにするために，AP と端末の距離が 0.3m と 10m で 60 分間測定をし，10 分毎の結果を算出し，その結果が最も良い結果のパラメータを使うことにした．

パラメータを決定するために用いた MW の設定値は以下の通りである．

表 4　基準とするパラメータの値

	MW A		MW B		MW C		MW D	
	max	min	max	min	max	min	max	min
0 <= ratio_rtt < 51	330	330	330	300	330	330	250	200
51 <= ratio_rtt < 106	330	300	300	200	330	150	250	150
106 <= ratio_rtt < 161	330	200	250	150	330	100	250	100
161 <=ratio_rtt < 331	330	200	250	100	330	50	250	50
331 <= ratio_rtt	330	100	200	50	330	10	250	10

実験結果は図 13，14 で示す．

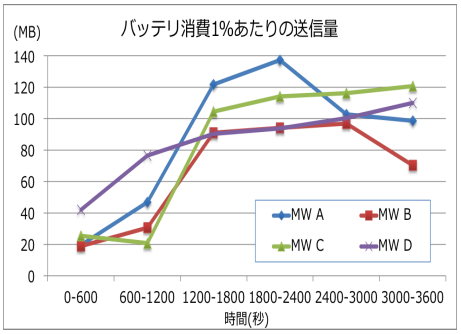

図 13　MW の性能測定（0.3m）

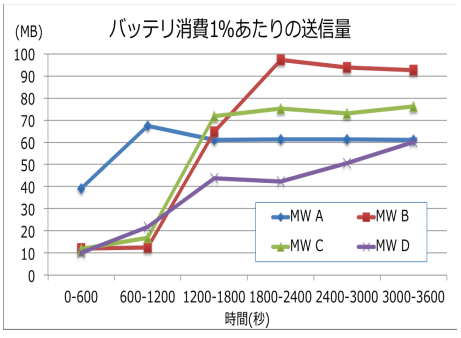

図 14　MW の性能測定（10m）

以上の結果より，時間毎に設定するパラメータの値は表 5 のようにした．また，それぞれの時間における 3m，5m，7m のパラメータの値は式 4，5 より算出した．図 15 に大まかな流れを示す．測定時間によりパラメータの値が変わるように設定されている．それぞれの時間の設定したパラメータの値は表 6，7，8，9 で示す．

表 5　時間毎に設定したパラメータの値

時間	0.3m	10m
0〜1200 秒	MW D	MW A
1200〜1800 秒	MW A	MW C
1800〜2400 秒	MW A	MW B
2400〜3600 秒	MW C	MW B

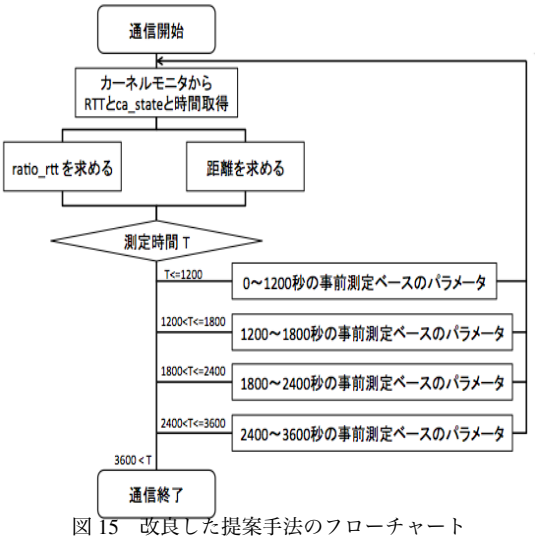

図 15　改良した提案手法のフローチャート

表 6　事前測定から設定した 0〜1200 秒のパラメータの値

	0.3m(D)		3m		5m		7m		10m(A)	
	max	min	max	min	max	min	max	min	max	min
0 <= ratio_rtt < 51	250	200	272.3	236.2	288.8	262.9	305.6	289.8	330	330
51 <= ratio_rtt < 106	250	150	272.3	191.8	288.8	222.7	305.6	253.6	330	300
106 <= ratio_rtt < 161	250	100	272.3	141.8	288.8	172.7	305.6	203.6	330	200
161 <=ratio_rtt < 331	250	50	272.3	91.8	288.8	122.7	305.6	153.6	330	200
331 <= ratio_rtt	250	10	272.3	35.1	288.8	53.6	305.6	72.2	330	100

表 7　事前測定から設定した 1200〜1800 秒のパラメータの値

	0.3m(A)		3m		5m		7m		10m(C)	
	max	min	max	min	max	min	max	min	max	min
0 <= ratio_rtt < 51	330	300	330	330	330	330	330	330	330	330
51 <= ratio_rtt < 106	300	200	330	258.3	330	227.3	330	196.3	330	150
106 <= ratio_rtt < 161	250	150	330	208.3	330	177.3	330	146.4	330	100
161 <=ratio_rtt < 331	250	200	330	158.3	330	127.3	330	96.4	330	50
331 <= ratio_rtt	330	100	200	74.9	330	56.4	330	37.8	330	10

表 8　事前測定から設定した 1800〜2400 秒のパラメータの値

	0.3m(A)		3m		5m		7m		10m(B)	
	max	min	max	min	max	min	max	min	max	min
0 <= ratio_rtt < 51	330	330	330	321.7	330	315.5	330	309.3	330	300
51 <= ratio_rtt < 106	330	300	321.7	272.2	315.5	251.6	309.3	230.9	300	200
106 <= ratio_rtt < 161	330	200	307.7	222.2	291.2	201.6	274.7	180.9	250	150
161 <=ratio_rtt < 331	330	150	307.7	172.2	291.2	151.6	274.7	130.9	250	100
331 <= ratio_rtt	330	100	293.8	86.1	267.1	75.8	240.2	65.5	200	50

表 9　事前測定から設定した 2400～3600 秒のパラメータの値

	0.3m(C)		3m		5m		7m		10m(B)	
	max	min	max	min	max	min	max	min	max	min
0 <= ratio_rtt < 51	330	330	330	321.7	330	315.5	330	309.3	330	300
51 <= ratio_rtt < 106	330	150	321.7	163.9	315.5	174.2	309.3	184.5	300	200
106 <= ratio_rtt < 161	330	100	307.7	113.9	291.2	124.2	274.7	132.5	250	150
161 <=ratio_rtt < 331	330	50	307.7	63.9	291.2	74.2	274.7	84.5	250	100
331 <= ratio_rtt	330	10	293.8	21.1	267.0	29.4	240.2	37.6	200	50

6.4.2　実験結果と考察

図 16 は通信制御オフのデフォルト時の測定結果，図 17 は各距離における事前測定ベースのパラメータを設定した場合の測定結果，図 18 は提案手法を導入した場合の測定結果を示す．それぞれ 30 分ごとの結果を表している．図より，デフォルト時も，事前測定ベースのパラメータを設定した場合も，提案手法も，時間が経つにつれて通信効率が大幅に向上することが確認できた．また，いずれの場合も 1200～3000 秒，1800～3600 秒の結果が高く保たれていることから，通信が開始してから 20 分程度経過するとその環境，条件での性能が確認できると考えられる．

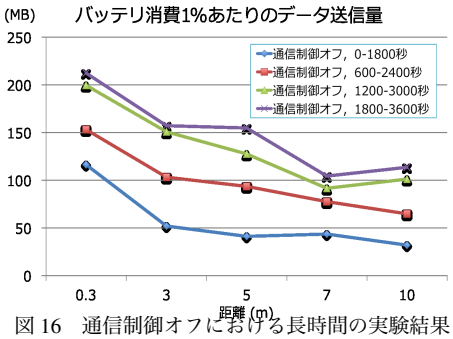

図 16　通信制御オフにおける長時間の実験結果

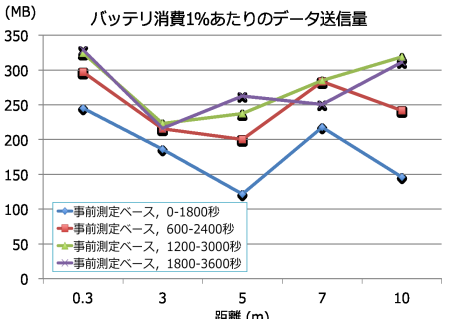

図 17　事前測定ベースのパラメータ手法における長時間の実験結果

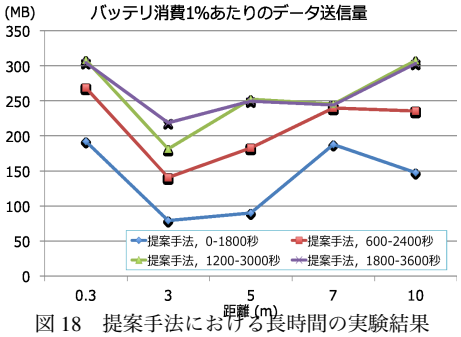

図 18　提案手法における長時間の実験結果

次に，時間ごとに比較をする．図 19 は 0～1800 秒，図 20 は 600～2400 秒，図 21 は 1200～3000 秒，図 22 は 1800～3600 秒の結果である．0～1800 秒の 30 分間の結果を見ると，提案手法の性能は，通信制御オフの手法には各距離で勝っているものの事前測定ベース手法には近付いていない．しかし，第 6.1 章で考察したように，始めの 30 分は事前測定ベース手法と提案手法では差が開いているが，実験が開始されてから時間が経つにつれて事前測定ベース手法と提案手法では差がほとんどなくなり，実験開始から 30～60 分の 30 分間の結果を確認すると，いずれの距離でもほとんど性能は等しくなっていることがわかる．この結果より，第 6.2 章で提案手法の有効性はあまり確認できなかったのは測定時間が短く，通信が開始してからの最初の 30 分のみだと距離の算出が安定せず，性能が悪くなってしまったと考えられる．

よって本提案手法は，測定時間が極端に短すぎなければ十分に有効であるといえる．

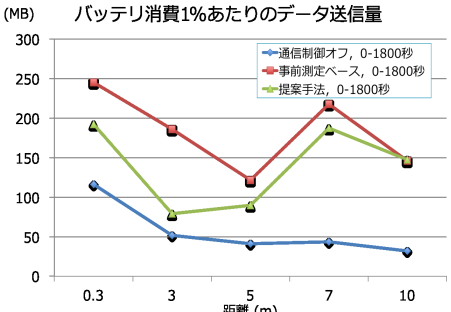

図 19　0～1800 秒の各手法の比較

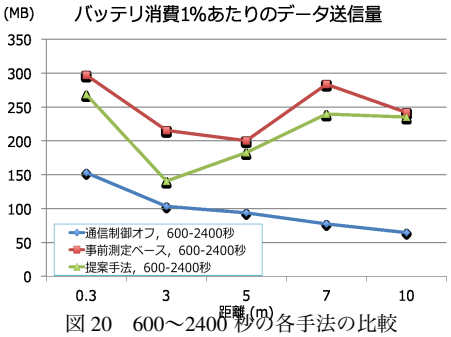

図 20　600～2400 秒の各手法の比較

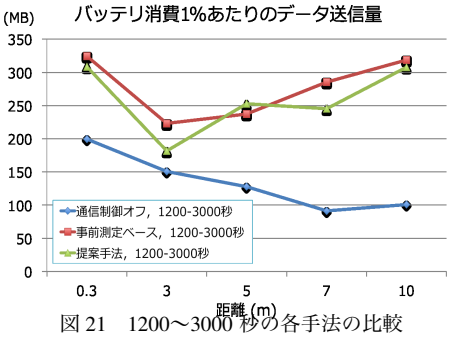

図 21　1200～3000 秒の各手法の比較

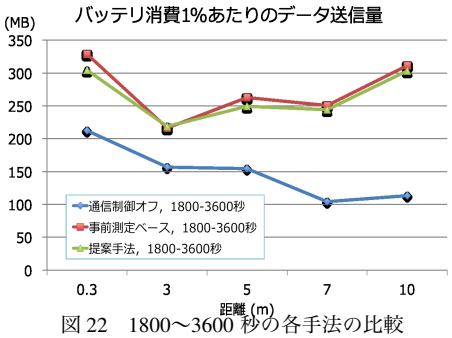

図 22　1800〜3600 秒の各手法の比較

6.5　同時接続する端末の距離が異なる場合の評価

6.5.1　実験結果と考察

これまでの測定では，AP に同時に接続する端末の距離は全て同じであった．しかし，現実世界では AP までの距離はユーザによって異なり，皆が同じ距離で接続することは考えにくい．そこで本章では，AP までの距離を 0.3m，3m，5m，7m，10m と全て異なるものとし，提案ミドルウェアで距離の判定ができるのか，また通信効率にどのような影響を及ぼすのかについて調査を行った．

図 23 は端末 5 台の平均のバッテリ消費 1%あたりのデータ送信量を，各時間ごと集計した結果である．図より，0〜1800 秒の性能があまり期待できない実験開始直後から，提案手法は良い性能を示している．また，通信が安定してきても常に高い値を取り続けていることから，非常に有効であることが確認できる．

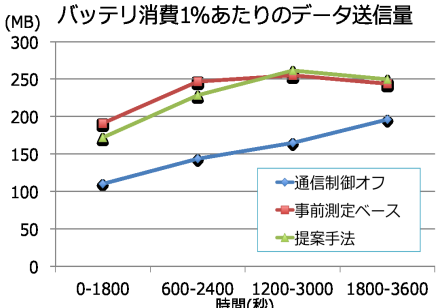

図 23　AP に同時接続する端末の距離が異なる通信時の実験結果

図 24 は提案ミドルウェアで測定した際の，MW が算出した距離の結果を示している．この図から，提案手法では各距離できちんと距離の算出ができており，そのため良い性能を示したといえる．通信端末の距離が異なる場合でも，提案 MW は正しく機能することが確認できた．

以上のことより，事前測定ベースは時間，環境，台数などそれら全ての条件を考慮して最適なパラメータを設定するのは不可能であり，無限通りあるパターンの中でのあくまで測定したデータの最適なものを選んでいるに過ぎない．しかしながら有限回の測定を元に設定した事前測定ベースのパラメータを提案手法で設定したことで，想定しきれない未知の条件下での測定でもある程度の性能を出すことが確認できた．よって本提案手法は，まだ設定するパラメータの値によって性能は向上する可能性もあるものの，十分に有効であるといえる．

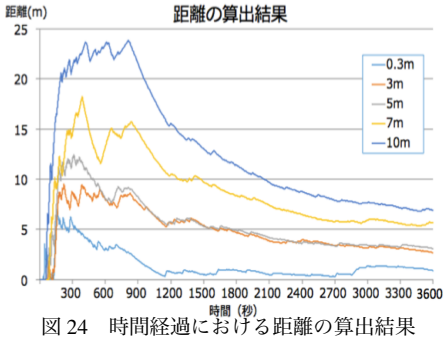

図 24　時間経過における距離の算出結果

7.　まとめと今後の課題

本研究では, スマートフォンのバッテリ消費の原因として，通信による消費に着目をした．AP に接続されている端末台数が多いほど，また AP と端末の距離が遠くなるほど通信効率は下がる．先行研究で提案された通信制御手法を用いると，台数が多い場合の通信効率を上がることができるが，距離が遠い場合はうまく働いていない．そこで距離に応じた柔軟な制御手法を提案した．提案手法により，バッテリ消費あたりのデータ送信量の向上に成功した．また，同時に AP に接続する端末の距離が異なる場合も，それぞれが距離を正しく算出し，バッテリ消費の削減に大きな効果を得られることが確認できた．このことから，有限回の測定から設定したパラメータであっても，想定しきれない未知の条件下でもある程度の性能を示すことがいえる．よって本提案手法は十分に有効であることが証明できた．

今後は距離測定の誤差を小さくするために，距離を算出するための手法を改善し，より性能のよい制御手法を用いて更なるバッテリ性能の向上を目指す．

文　　献

[1] IDC :http://www.idc.com/promo/smartphone-market-share/os;jsessionid=84526CAAEE8B1379B4224F60E822DEDF

[2] 総務省:http://www.soumu.go.jp/main_content/000322502.pdf

[3] 古庄 裕貴，久住 憲嗣，神山 剛，稲村 浩，中西 恒夫，福田 晃, : Android アプリケーションの運用時消費電力分析, 電子情報通信学会, IEICE Technical Repor ， SS2012 － 58, 2013 年 1 月

[4] 井原 卓也，田坂 和之，大岸 智彦，小花 貞夫 : スマートフォンの Firefox OS と Android の 消費電流量に関する一考察, 情報処理学会第 76 回全国大会, 4W-3 900-908

[5] Murmuria, R., Medsger, Jeffrey, Stavrou, A., Voas, ”Mobile Application and Device Power Usage Measurements,” Software Security and Reliability (SERE), 2012 IEEE Sixth International Conference, pp. 147-156

[6] Roy Friedman, Alex Kogan, ” On Power and Throughput Tradeoffs of WiFi and Bluetooth in Smartphones,” INFOCOM, 2011 Proceedings IEEE, pp. 900-908

[7] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi : ” Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals,” Proceedings of The Tenth International Conference on Networks (ICN), pp. 297-302. 2011.

[8] Hiromi Hirai, Saneyasu Yamaguchi, and Masato Oguchi: ” A Proposal on Cooperative Transmission Control Middleware on a Smartphone in a WLAN Environment, ” In Proc. the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob2013), pp.710-717, October 2013.

[9] 早川愛， 山口実靖， 小口正人:無線 LAN-AP における TCP ACK パケット蓄積回避のための協調的輻輳制御手法の提案と実装, DEIM2015, C2-2,2015 年 3 月.