

無線リンクの高負荷状態における アプリケーションレベル通信遅延低減方式の評価実験

西川 由明^{1,a)} 大芝 崇^{1,b)} 金友 大^{1,c)} 中島 一彰^{1,d)}

概要：本稿では、アプリケーション間のデータ通信の遅延時間を 100ms 以下に低減可能なアプリケーションレベル通信遅延低減方式の評価実験について報告する。1 つの無線チャネルを複数トラフィックで共有する無線 LAN はホットスポットやテザリングなどで利用される機会が増えている。無線 LAN では、大きなクロストラフィックが発生するとコリジョンが増えて通信品質の劣化を招き、データ通信の遅延時間が 300ms を超えることが問題となっている。そこで、遅延時間を低減可能なアプリケーションレベル通信遅延低減方式を開発して、関東地方にある複数の拠点を商用固定網で相互に接続し、無線 LAN 接続の実機を用いた環境で評価実験を行った。評価実験では、送信するデータ数を削減する方式と TCP の高速再送を早期に誘発する方式、ACK を即座に返信させる方式の 3 つのアプリケーションレベル通信遅延低減方式の遅延の改善効果を測定した。評価実験の結果、各方式を単独で用いると遅延時間の 95% 確率値をそれぞれ 83%、73%、65% 低減でき、ITU の勧告で要求される 100ms 以下を達成した。単独で要求を達成した 2 つの方式を組み合わせると遅延時間の 95% 確率値を最大で 84% 低減できた。

1. はじめに

スマートフォンやタブレットなどのモバイル端末の普及に伴い、無線 LAN を利用する機会が増えている。最近では、家庭やオフィス、ホテルなど様々な場所で無線 LAN によるインターネット接続が提供されている。スマートフォンやタブレットのような LAN ケーブル接続端子を持たないモバイル端末が普及してきており、主に PC を接続していた頃と比べて無線 LAN の利用機会が増えている。さらに、スマートフォンの無線 LAN テザリングやモバイルルータの無線 LAN を利用して外出中に、モバイル PC やタブレット、ゲーム機などでインターネットに接続することも一般的になった。スマートフォンの無線 LAN テザリングとは、スマートフォンを無線 LAN アクセスポイントとして利用することである。スマートフォンのインターネット側の通信に 3G や LTE などの移動無線通信を用い、ローカル側に無線 LAN を用いる。モバイルルータとは WAN 側の通信に移動無線通信を用いるルータであり、ローカル側

に無線 LAN アクセスポイントの機能を持っている。これらを利用することで、個人が好きなときに無線 LAN を構築し、その人が持つモバイル端末を用いてインターネットに接続することができる。

無線 LAN やスマートフォンの普及により、通信品質が劣化する問題が顕在化してきている。個人が好きなときに無線 LAN を構築するようになると、それぞれ独自に管理されている多数の無線 LAN が集合する環境が形成される。そのような環境では、同じチャネルを共有する他の無線 LAN からの影響で通信品質が劣化するチャネル競合問題と異なるチャネルを利用しているがチャネルの帯域幅に重なりのある無線 LAN からの影響で通信品質が劣化するチャネル干渉問題が発生する。さらに、スマートフォンの普及により、3G や LTE などの移動無線通信からより広帯域な無線 LAN へのオフロードが進んでいる。無線 LAN では従来からのトラフィックと増加したオフロードのトラフィックの影響で通信品質が劣化するクロストラフィック増加問題が発生する。無線 LAN の通信品質が劣化すると、頻繁に TCP 通信を行うリアルタイムアプリケーションの遅延時間が大きく増加してしまう。

本稿では、通信品質が劣化し遅延が増加した状態におけるアプリケーションレベル通信遅延低減方式の評価実験について報告する。小サイズで高頻度に発生するデータを TCP で通信するリアルタイムアプリケーションにおいて、

¹ NEC クラウドシステム研究所
NEC, Cloud System Research Laboratories
1753 Shimonumabe Nakahara-ku Kawasaki,
Kanagawa 211-8666, Japan

a) y-nishikawa@ah.jp.nec.com

b) oshiba@cp.jp.nec.com

c) d-kanetomo@ce.jp.nec.com

d) nakajima@ah.jp.nec.com

送信するデータ数を削減する方式と TCP の高速再送を早期に誘発する方式、ACK を即座に返信させる方式の 3 つのアプリケーションレベル通信遅延低減方式を動作させ遅延の改善効果を測定した。評価実験は、関東地方にある複数の拠点を商用固定網で相互に接続し、無線 LAN 接続の実機を用いた環境で行った。評価実験の結果、無線 LAN 区間のクロストラヒックとしてファイル転送を行った場合に、各方式を単独で用いると遅延時間の 95% 確率値をそれぞれ 83%、73%、65% 低減でき、送信するデータ数を削減する方式と TCP の高速再送を早期に誘発する方式でリアルタイムアプリケーションの遅延時間として求められる 100ms 以下 [10] を達成した。単独で目標を達成した 2 つの方式を組み合わせると遅延時間の 95% 確率値を最大で 84% 低減できた。

2. 関連研究

この章では本稿と関連する研究を紹介し、違いを述べる。無線 LAN の競合問題や干渉問題に対処する研究がなされている。[3] では、他のアクセスポイントが使うチャネルをスキャンし自動的に利用するチャネルを選択する方法が示されている。[2] では、チャネル干渉を考慮したアクセスポイントの最適な配置とチャネルの選択方法が提案されている。[4] では、無線チャネルの干渉と競合を考慮したチャネル割当方法が提案されている。本稿では、通信を行うエンド-エンドのリアルタイムアプリケーションでの通信制御で遅延低減を行うので、これらの研究とは異なる。[5] では、複数のコネクションを用いたリアルタイム伝送方式が提案されている。[6] では、NIST Net を用いた実験を行い、クロストラヒックがある環境でデータ分割制御と複数コネクションを用いた送信制御を行うと遅延時間を低減できることが示されている。本稿では、無線リンクが高負荷となりリアルタイムアプリケーションでの通信遅延が増大する問題にたいして、送信するデータ数を削減する方式と TCP の高速再送を早期に誘発する方式、ACK を即座に返信させる方式の 3 つのアプリケーションレベルで制御する方式を実網で評価する。

3. TCP 通信を行うリアルタイムアプリケーションの課題

本章では、TCP 通信を行うリアルタイムアプリケーションにおける通信遅延低減の重要性和通信遅延増大の原因について説明する。TCP 通信を行うリアルタイムアプリケーションとは、リモートデスクトップアプリケーションのようなものである。リモートデスクトップアプリケーションでは、マウスや画面操作などのユーザの操作情報を遠隔にある端末へ周期的に送信し、操作の結果である画像のような情報を手元にある端末で受信する。このとき操作情報を送信してから操作結果を受信するまでの遅延時間がアプリ

ケーションの使い勝手に大きな影響を与える。操作結果を早く受信するためには、ユーザの操作情報の通信遅延であるエンド-エンド遅延を低減することが重要になる。ITU の勧告 [10] では、リアルタイムアプリケーションに求められるエンド-エンド遅延時間を 100ms 以下としている。以降では、エンド-エンド遅延を単に遅延と呼ぶ。

リアルタイムアプリケーションで TCP 通信を用いる場合、遅延増大には 3 つの原因があると考えている。第 1 の原因は、TCP や NW でのキューイングである。TCP 通信を行う場合は、TCP のウィンドウサイズによる制限と NW 中の中継装置の能力による制限で可用帯域が決定され、その可用帯域を越える頻度でデータ送信を行うとデータがキューイングされ遅延が累積的に増加してしまう。例えば、TCP の送信ウィンドウサイズの制限を越えて送信しようとした場合には、TCP の送信キューに一時保管され、ACK 受信により送信可能な状態になるまで遅延が増加してしまう。また、転送能力を超える頻度でパケットが NW 上の中継装置に到着した場合には、中継装置内でキューイングされ転送処理を待つ遅延が増加してしまう。第 2 の原因は、ロス発生時の順序制御である。ロスしたセグメントより後で送信され、ロスしたセグメントの再送完了より前に受信側に届いたセグメントは、ロスしたセグメントの再送が完了されるまで受信側のバッファで待機させられる。リアルタイムアプリケーションは、低頻度でデータ送信を行うアプリケーションよりも待機させられる時間が長く、待機させられるセグメント数も多くなるので、順序制御による影響を強く受ける。第 3 の原因は、遅延 ACK 制御である。遅延 ACK とは、複数のセグメントを受信するかセグメント受信後一定時間経つかのどちらかを契機に ACK を返送する制御である。データ送信が高頻度であれば、遅延 ACK により ACK 返送が待機させられている間に、送信セグメント数がウィンドウ制御の制限に達してしまい遅延が増加してしまう。

4. 提案方式

本章では、アプリケーションレベル通信遅延低減方式について説明する。各方式は、前章で述べた 3 つの原因に対応している。

4.1 方式 1 : Reduction- P 方式

アプリケーションでは、送信するデータ数を減らし前章で述べた第 1 の原因である TCP や NW でのキューイングによる遅延を低減することができる。この方式を Reduction- P 方式と呼ぶ。 P とはデータ削減率である。図 1 に、可用帯域以上にデータを送信したときに遅延が累積的に貯まる場合とアプリケーションレベルで Reduction- P 方式を用いデータの送信を可用帯域以下に抑え遅延が貯まらない場合のシーケンスを示す。リアルタイムアプリケーション

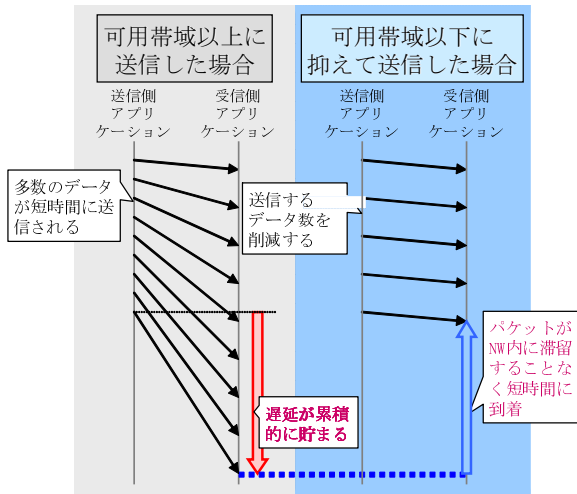


図 1 可用帯域以上にデータを送信したときに遅延が累積的に貯まる場合（左側）とアプリケーションレベルで Reduction-P 方式を用いデータの送信を可用帯域以下に抑え遅延が貯まらない場合（右側）

が行う通信では高頻度でパケットが送信されるので消費帯域が可用帯域を超過してしまうことがある。Reduction-P では、 $P\%$ のデータ送信を控えることで、消費帯域を可用帯域以下に抑える。例えば Reduction-66 では 3 回に 2 回データ送信を行わないことで、データ送信間隔を均等に延長し、パケットの送信頻度を低減する。

4.2 方式 2: Trigger 方式

アプリケーションでは、TCP の高速再送を誘発するための数個のデータを送信し、前章で述べた第 2 の原因であるロス発生時の順序制御による遅延を低減することができる。この方式を Trigger 方式と呼ぶ。図 2 に TCP の高速再送のシーケンスとアプリケーションレベルで Trigger 方式を用いて高速再送を早期に誘発させた場合のシーケンスを示す。トリガー方式では、まず受信側アプリケーションがデータの未到着を検出する。今回は周期的にデータを送信するアプリケーションを選んだため、前回データが到着してから時間をかけて未到着を検出する。データの未到着を検出すると送信側へ未到着を報告する。未到着報告を受信した送信側は、高速再送を誘発するために複数個の誘発用データを送信する。誘発用データが受信側に到着したとき、誘発用データより先に送信されたデータが未到着である場合には重複 ACK が自動的に返信される。一定数の重複 ACK を受信した送信側では、高速再送アルゴリズムにより再送が自動的に実施される。再送されるために必要な重複 ACK の数は OS に設定されており、通常は 2 個か 3 個である。本稿では 3 個の誘発データを送信し、3 個の重複 ACK を送信させる。

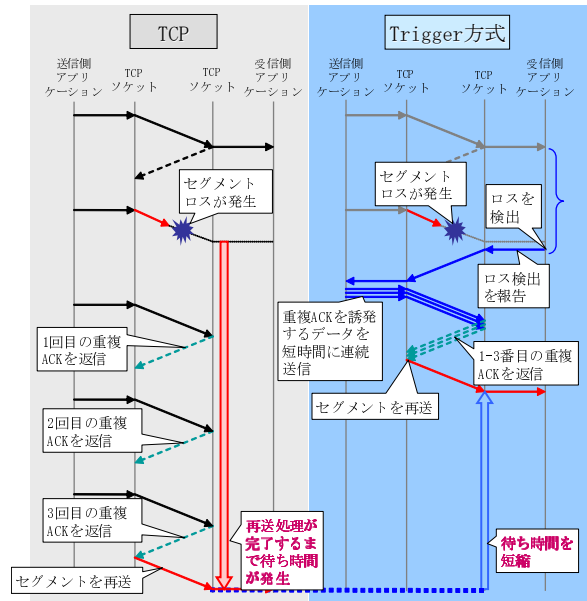


図 2 TCP の高速再送のシーケンス（左側）とアプリケーションレベルで Trigger 方式を用いて高速再送を早期に誘発させた場合のシーケンス（右側）

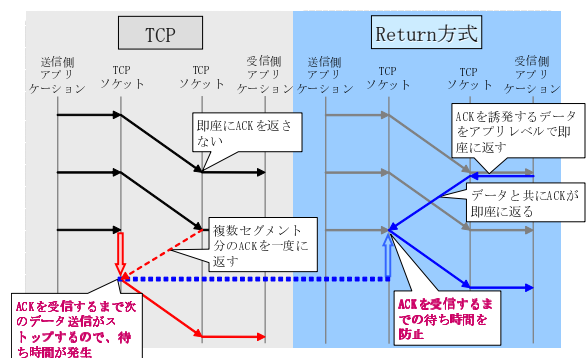


図 3 遅延 ACK 制御のシーケンス（左側）と遅延 ACK 制御が有効である場合にアプリケーションレベルで Return 方式を用いて即座に ACK を返信させた場合のシーケンス（右側）

4.3 方式 3: Return 方式

アプリケーションでは、データ受信後に即座にデータを返信し前章で述べた第 3 の原因である遅延 ACK 制御による遅延を低減することができる。この方式を Return 方式と呼ぶ。図 3 に、遅延 ACK 制御のシーケンスと遅延 ACK 制御が有効である場合にアプリケーションレベルで Return 方式を用いて即座に ACK を返信させた場合のシーケンスを示す。Return 方式では、受信側アプリケーションはデータを受信すると即座に ACK 送信用データを返信する。TCP には、ACK セグメントのオーバーヘッドを低減するためにデータセグメントに ACK を載せる制御を行う。即座に返信した ACK 送信用データのセグメントに TCP が ACK を載せるので、ACK が即座に返信される。

5. 実験環境

我々は前章で述べた方式の評価実験を行った。本章で

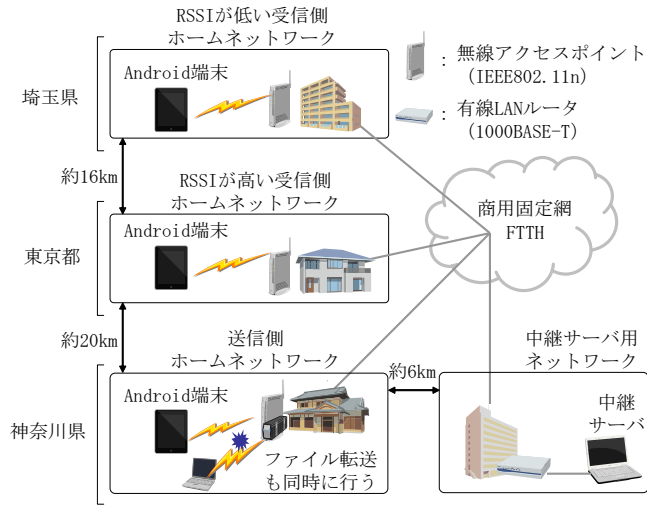


図 4 実験ネットワーク図

表 1 端末とサーバの仕様

	Android 端末	中継サーバ
製品名	SGH-N023[8]	VY14AC-W[9]
OS	Android OS 2.3	Windows XP SP3
CPU	S5PC110 1GHz	core2duo 1.4GHz
RAM	0.51 GB	2.96 GB
NIC	無線 (IEEE802.11n)	有線 (1000BASE-T)

はその実験の環境と設定について説明する。図 5 に実験ネットワークの図を示す。ネットワークは互いに数 km 以上離れた位置にある 3 つのホームネットワークと 1 つの中継サーバのネットワークからなる。4 つのネットワークは神奈川県と東京都、埼玉県にあり、商用の固定網を介して FTTH で結ばれている。神奈川県のホームネットワークでは送信側アプリケーションが無線接続された Android 端末上で動作する。東京都と埼玉県のホームネットワークでは受信側アプリケーションが無線接続された Android 端末上で動作する。無線 LAN 接続にはすべて IEEE802.11n 方式を用いている。東京都と埼玉県のホームネットワークでは RSSI が異なり、東京都では -40dbm であり埼玉県では -58dbm である。中継サーバのネットワークでは、中継サーバが有線接続されている。送信側アプリケーションが中継サーバに操作データを送信し、中継サーバが受信側アプリケーションのどちらかに操作データを転送する。前章で述べた方式は、送信側アプリケーションと中継サーバ間、中継サーバと受信側アプリケーション間で別々に制御される。無線リンクに負荷をかけるために、神奈川県のホームネットワークでは Android 端末と同じアクセスポイントに同じチャネルで WindowsPC を接続し、アクセスポイントに接続されている NAS から WindowsPC にファイル転送を行った。Android 端末や WindowsPC はそれぞれ表 1 に記載の機種を用いた。無線アクセスポイントと有線のルータにはすべて AtermWR8750N を用いた。

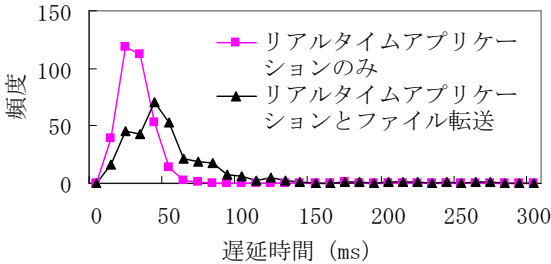


図 5 リアルタイムアプリケーションのみが通信を行った場合とファイル転送を同時に行い無線 LAN に負荷をかけた場合のリアルタイムアプリケーションの遅延時間の頻度分布

AtermWR8750N の実効帯域は、無線で 184Mbps、有線で 840Mbps である [7]。

実験では、操作データとして Android 端末のタッチパネル上での指のスライド情報を用いた。操作データは自動的に 30ms 間隔で 60 個連続して発生する。操作データのサイズは、1 つ 50byte である。Nagle のアルゴリズム [1] を無効にしており、アプリケーションは操作データが発生すると即座に TCP に送信する。60 個のデータをまとめてストリームと呼び、本稿ではストリームの遅延時間の 95% 確率値をリアルタイムアプリケーションに要求される遅延時間である 100ms 以下 [10] にすることを目標とする。遅延時間を評価する場合、典型的には平均や中央値、最頻値を用いる。しかし、リアルタイムアプリケーションでは一部の長い遅延が使い勝手に影響するので適切ではないと考える。本稿では、よく使われている 95% 確率値を用いる。

予備実験として、無線リンクに負荷をかけるために導入するファイル転送によって無線 LAN の通信品質が劣化することを確認する実験を行った。図 5 に、ファイル転送がなくリアルタイムアプリケーションのみが通信を行った場合とファイル転送中にリアルタイムアプリケーションが通信を行った場合でそれぞれ 300 ストリーム計測したときの遅延時間の頻度分布を示す。横軸が遅延時間であり、縦軸が頻度である。ファイル転送がある場合の方が遅延時間が長くなっていることがわかる。ファイル転送がない場合は、最大遅延時間は 162ms で 95% 確率値は 42ms であった。ファイル転送がある場合は、図示していないが最大遅延時間は 2451ms で 95% 確率値は 354ms であった。ファイル転送が存在することで 95% 確率値が 7 倍程度増加しており、ファイル転送により無線 LAN の通信品質が劣化することを確認できた。

本稿の実験では、前章で述べた方式を次のように設定する。Reduction- P 方式では、送信データ数を $1/2$ または $1/3$ にするために、 P を 50 または 66 に設定する。Trigger 方式では、受信側がデータ未到着を検出するための時間を、Reduction- P 方式の設定で変化するデータ送信間隔に 20ms 加えた値に設定する。つまり、Reduction- P を用い

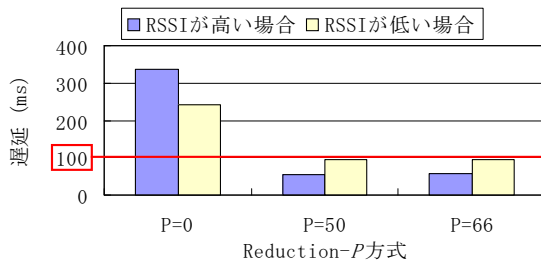


図 6 RSSI が高い場合と低い場合それぞれについて、アプリケーションレベル制御を行わなかった場合と Reduction-P 方式を用いてデータを削減した場合の遅延時間の 95% 確率値

ない場合は 50ms、 P が 50 の場合は 80ms、 P が 66 の場合は 110ms である。

6. 実験結果

本章では、TCP 通信を行うときにアプリケーションレベル制御として各提案方式を単独で用いた場合の結果を示し、その後各提案方式を複数個組み合わせて用いた場合の結果を示す。実験は 4800 ストリーム行い、各 95% 確率値は 300 個の値から算出した。本章で参照する図では RSSI が高い場合と低い場合のそれぞれの結果を同じ図に示している。

6.1 提案方式を単独で用いた場合の結果

図 6 に、アプリケーションレベル制御を行わず発生したデータをすべて送信した場合と Reduction-P 方式を用いてデータを削減して送信した場合の遅延時間の 95% 確率値を示す。横軸はデータ削減率 P であり、縦軸は遅延時間の 95% 確率値である。 P が 0 のグラフが発生したデータをすべて送信した場合のグラフである。RSSI が高い場合は、発生したデータをすべて送信した場合は 336ms、Reduction-50 で 56ms、Reduction-66 で 58ms となった。RSSI が低い場合は、発生したデータをすべて送信した場合は 242ms、Reduction-50 で 95ms、Reduction-66 で 98ms となった。送信するデータ数を 50% 削減することで、遅延時間を最大で 83% 低減できている。50% 削減した場合と 66% 削減した場合では遅延時間の低減効果に大きな差はなかった。

図 7 に、アプリケーションレベル制御を行わなかった場合と Trigger 方式を用いた場合の遅延時間の 95% 確率値を示す。横軸はアプリケーションレベル制御を行わなかった場合と Trigger 方式を用いた場合であり、縦軸は遅延時間の 95% 確率値である。RSSI が高い場合は、Trigger 方式を用いると遅延時間は 96ms となり、73% 低減できた。RSSI が低い場合は、Trigger 方式を用いると遅延時間は 126ms となり、48% 低減できた。

図 8 に、アプリケーションレベル制御を行わなかった場合と Return 方式を用いた場合の遅延時間の 95% 確率値を示す。

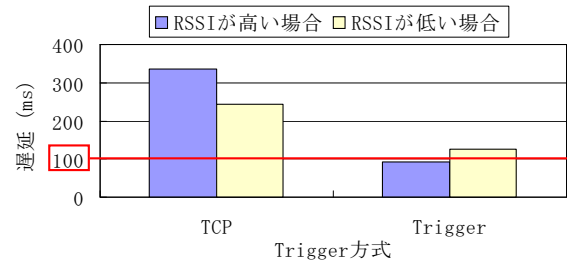


図 7 RSSI が高い場合と低い場合それぞれについて、アプリケーションレベル制御を行わなかった場合と Trigger 方式を用いた場合の遅延時間の 95% 確率値

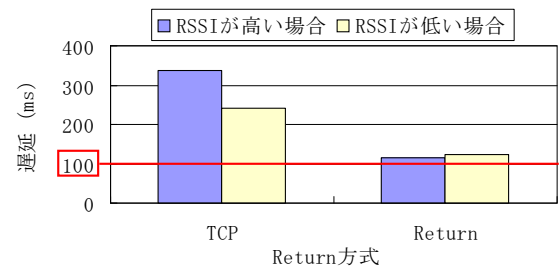


図 8 RSSI が高い場合と低い場合それぞれについて、アプリケーションレベル制御を行わなかった場合と Trigger 方式を用いた場合の遅延時間の 95% 確率値

示す。横軸はアプリケーションレベル制御を行わなかった場合と Trigger 方式を用いた場合であり、縦軸は遅延時間の 95% 確率値である。RSSI が高い場合は、Return 方式を用いると遅延時間は 116ms となり、65% 低減できた。RSSI が低い場合は、Return 方式を用いると遅延時間は 124ms となり、49% 低減できた。

6.2 提案方式を組み合わせ用いた場合の結果

図 9 に、Reduction-50 のみ用いた場合と Reduction-50 と Trigger 方式もしくは Return 方式を組み合わせ用いた場合の遅延時間の 95% 確率値を示す。横軸は Reduction-50 のみ用いるか Reduction-50 と Trigger 方式もしくは Return 方式を組み合わせ用いるかであり、縦軸は遅延時間の 95% 確率値である。RSSI が高い場合は、Trigger 方式と組み合わせ用いると遅延時間は 55ms となり、Return 方式と組み合わせ用いると 60ms になり、一般的な TCP 通信を行う場合と比較して最大で 84% 低減できた。RSSI が低い場合は、Trigger 方式と組み合わせ用いると遅延時間は 101ms となり、Return 方式と組み合わせ用いると 94ms になり、一般的な TCP 通信を行う場合と比較して最大で 61% 低減できた。RSSI が高い場合には Reduction-50 と Trigger 方式を、RSSI が低い場合には Reduction-50 と Return 方式を、それぞれ組み合わせると Reduction-50 を単独で用いる場合よりも遅延低減効果があった。ただし、異なる組合せを用いた場合の逆効果の方が大きい。

図 10 に、Trigger 方式のみ用いた場合と Return 方式

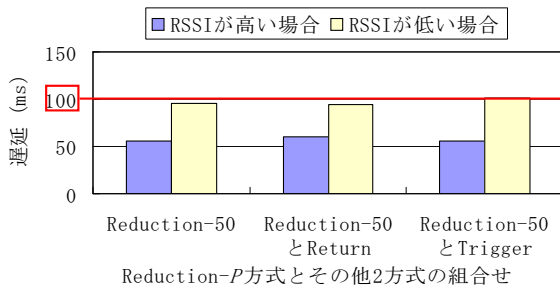


図 9 RSSIが高い場合と低い場合それぞれについて、Reduction-50を単独で用いた場合とReduction-50とReturn方式を組み合わせ用いた場合、Reduction-50とTrigger方式を組み合わせ用いた場合の遅延時間の95%確率値

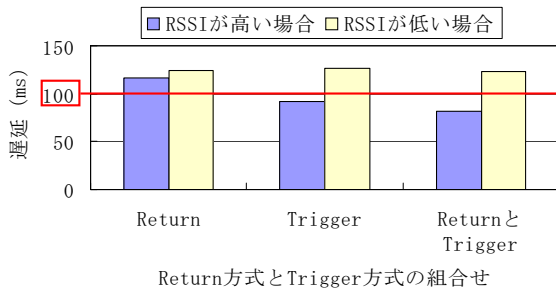


図 10 RSSIが高い場合と低い場合それぞれについて、Return方式を単独で用いた場合とTrigger方式を単独で用いた場合、Return方式とTrigger方式を組み合わせ用いた場合の遅延時間の95%確率値

のみを用いた場合、Trigger方式とReturn方式を組み合わせ用いた場合の遅延時間の95%確率値を示す。横軸はTrigger方式のみ用いるかReturn方式のみを用いるかTrigger方式とReturn方式を組み合わせ用いるかであり、縦軸は遅延時間の95%確率値である。RSSIが高い場合は、Trigger方式とReturn方式を組み合わせ用いると遅延時間は81msとなり、一般的なTCP通信を行う場合と比較して最大で76%低減できた。RSSIが低い場合は、Trigger方式とReturn方式を組み合わせ用いると遅延時間は123msとなり、一般的なTCP通信を行う場合と比較して最大で49%低減できた。RSSIが高い場合には、2つの方式を組み合わせると相乗効果を得られたが、RSSIが低い場合にはほとんど相乗効果を得られなかった。

Trigger方式とReturn方式は誘発用データとACK送信用データを送信するため、オーバーヘッドが存在する。Trigger方式において全データに対して誘発用データを返信し、Return方式において全データに対してACK送信用データを返信した場合、消費する帯域は最大になり、2つの方式を用いない場合と比較して3倍程度に増加する。しかし、消費帯域の絶対値は100Kbpsを超えず、無線LANの実効帯域である180Mbps[7]と比較すると十分に小さい。もし仮にこの消費帯域の増加が問題になる場合は、reduction-Pを用いると消費帯域を低減することができる。

7. おわりに

提案するアプリケーションレベル通信遅延低減方式を用いると、ファイル転送により無線リンクに負荷がかかった状態において、遅延時間の95%確率値を最大で84%低減でき、目標としていた100ms以下を達成する55msにできることを実験結果から確認した。送信データ数を削減する方式はRSSIの高低によらず単独では最も効果的であった。TCPの高速再送を早期に誘発させる方式はRSSIが高い場合にはACKを即座に返信させる方式よりも効果的であったが、RSSIが低い場合は、逆にACKを即座に返信させる方式の方が効果的であった。RSSIが高い場合には遅延時間が安定しているがRSSIが低い場合には遅延時間の変動が激しくなるので、データ未到着の検出に固定値のタイマーを用いている現状の方式では遅延時間の変動が激しい環境では誤検出や検出遅れが発生しやすい。これが原因となり余計な通信や高速再送誘発の遅れが発生し、効果が薄まったのだと考えられる。ネットワークの通信遅延を正確に推定し、推定値に基づいてデータ未到着を検出するとより効果が得られると考えている。

参考文献

- [1] John Nagle : Congestion Control in IP/TCP Internetworks, RFC 896, January 1984.
- [2] Youngseok Lee, Kyoungae Kim and Yanghee Choi : Optimization of AP Placement and Channel Assignment in Wireless LANs, Proc. Local Computer Networks 2002(LCN 2002), pp.6-8(2002).
- [3] Jihoon Choi, Kyubum Lee, Sae Rom Lee and Jay (Jong-tae) Ihm : Channel Selection for IEEE 802.11 Based Wireless LANs Using 2.4GHz Band, ELEX, Vol.8, No.16, pp.1275-1280(2012).
- [4] 熊谷 菜津美, 磯村 美友, 村瀬 勉, 小口 正人 : 無線LANアクセスポイントのチャネル内競合とチャネル間干渉を同時に考慮したチャネル割当手法, 電子情報通信学会技術研究報告, vol.112, no.288, CQ2012-68, pp.59-64 (2012).
- [5] 馬場 昌之, 西川 博文, 加藤 嘉明 : 複数TCPコネクションを利用したリアルタイム伝送方法, 電子情報通信学会技術研究報告. IE, 画像工学 vol.107, No.380, pp.71-76 (2007).
- [6] Eli Brosh, Salman Abdul Baset, Dan Rubenstein and Henning Schulzrinne : The Delay-Friendliness of TCP, In SIGMETRICS, June, 2008.
- [7] NEC Corporation : Product Information of AtermWR8750N, NEC (オンライン), 入手先 (<http://121ware.com/product/atermstation/product/warpstar/wr8750n-hp/>) (参照 2013-03-18).
- [8] SAMSUNG : Product Information of GALAXY Tab SC-01C, SAMSUNG (オンライン), 入手先 (<http://www.samsung.com/jp/support/model/SGH-N023CWND CM>) (参照 2013-03-18).
- [9] NEC Corporation : Product Information of VersaPro, NEC (オンライン), 入手先 (<http://121ware.com/e-manual/m/nx/vp/base/vy14acw.html>) (参照 2013-03-18).
- [10] ITU-T Recommendation Y.1541 : Network performance objectives for IP-based services. 2011.