

RSSIを利用した複数経路利用時のパケット分配手法

坪井 祐樹^{†1} 相田 仁^{†1}

無線通信技術の発展により、複数の無線インタフェースをサポートした端末が普及してきている。しかし、一般的に複数のインタフェースを同時に利用し、通信を行ってはいない。そこで、本稿ではそれらの端末において同時に無線インタフェースを利用し、効率的な通信を行うためのパケット分配手法について提案する。具体的には、各無線インタフェースから RSSI を取得し、現在の通信路の状態を推定することで、パケットロスの起こりにくいパケットの分配を行う。

RSSI based Packet Distribution Scheme for a Multi-path communication protocol

YUKI TSUBOI^{†1} and HITOSHI AIDA^{†1}

Recently, mobile hosts having several wireless interfaces are becoming popular because of the advancement of wireless communication technologies. However, generally, most of them use only one of interfaces. In this paper, we propose a packet distribution for a Multi-path communication protocol to communicate effectively. Specifically, we employ RSSI to distribute the packets over the multiple paths while reducing loss rate.

1. はじめに

近年、無線通信技術の発展により、インターネットがいつでもどこでも利用可能となっている。そして、無線環境における広帯域なアプリケーションやリアルタイム通信を必要とするアプリケーションへの需要も高まってきており、様々な無線通信方式が利用可能となっ

ている。代表的なものには、IEEE802.11 の無線 LAN 規格や、メトロポリタン・エリア・ネットワークの実現に向けた技術として IEEE802.16 などがある。しかし、未だ無線を用いた通信ではフェージング等の影響で通信が不安定となることにより、広帯域なアプリケーションやリアルタイム通信には向いていない。それらの無線技術の進歩や無線通信可能な環境の整備に伴い、複数の無線インタフェースを搭載した端末が普及してきている。複数の無線インタフェースを持つ端末では、それらを同時に利用した通信を行うことが可能であるが、一般的には利用していない。モバイル端末によるインターネットへのアクセスの需要が半分以上を占め、広帯域を利用するアプリケーション（ゲーム、動画 etc）の需要も高まってきている中、片方のインタフェースのみの通信は、資源の有効活用ができていないと考えられる。また、トラフィックを複数の経路に分散することで、耐障害性をあげることもでき、輻輳の発生防止につながると考えられる。

本稿では、無線環境での複数経路を用いた通信において、RSSI を用いたパケット分配手法を提案する。RSSI より通信路の状態を推定することで、無線区間が不安定な状態によるパケットロスを未然に防ぐことを目的とする。

本稿は以下のような構成になっている。第 2 節では複数経路を用いた通信に関する技術について概観する。第 3 節では、本稿で採用した複数経路同時利用のためのプロトコルについて簡単に説明を行う。その後、RSSI を用いた提案手法について述べ、実環境での予備実験やシミュレーションを用いた結果を載せる。第 5 節では本稿のまとめを述べる。

2. 関連研究

複数経路を用いた通信を行う研究には、オペレーティングシステム内にあるトランスポート層において実装されるものや、アプリケーション層において複数のソケットを用いて実装される物がある。他にも、トランスポート層だけでなく下位層と協調することで、効率的な複数経路利用通信を行う研究などがある。その中でも、トランスポート層の TCP を改良して実現されているものとして、pTCP¹⁾, mTCP²⁾, cTCP³⁾, Multi-path TCP⁴⁾ などがある。また、TCP ではなく SCTP (Stream Control Transmission Protocol)⁵⁾ を改良したものとして、CMT⁶⁾ などがある。他には SCTP を用いてアプリケーション層での実装を行っている ARMS⁷⁾ などがある。複数経路を用いた通信を行う際に、輻輳制御とパケット分配をどのように行うのか、という点に着目して以上の中からいくつかを詳しく述べる。

^{†1} 東京大学
Tokyo University

2.1 pTCP

pTCP(parallel TCP) はトランスポートプロトコルの TCP を改良することで複数経路の同時利用を可能とした研究である。上位層としての SM(Striped connection Manager) が二つの端末を結ぶ TCP コネクションをまとめる役割を担っている。SM はアプリケーションから pTCP を用いて接続の要求があると、接続可能なインターフェースの数だけ TCP-v を作成する。この TCP-v は通常の TCP を少し変更したものである。通信する際の輻輳制御はパスが各々行なっていて、データの送信量の制御は SM が行っている。また、コネクションが現在使用可能かどうかについても SM が管理を行っており、使用できないコネクションについては SM が close 命令をだし、使用可能な新しいアドレスを発見した際には open 命令をだして、新たに TCP-v を作成する。データはポリシーベースでそれぞれのインターフェースに分配が行われる。

2.2 Multipath TCP

MTCP(Multipath TCP) は pTCP と同じくトランスポート層を改良して複数経路の同時利用を実現している。この研究では、輻輳制御は下位層にあたる一つ一つの経路を担当する層が行っている。そして、複数経路利用を原因としたパケットの順序逆転による無用な再送を防ぐために、パケットヘッダーでサブフローごとのシーケンス番号を付加していることが特徴として挙げられる。サブフローごとに順序が正しければ、通常付加されるシーケンス番号を用いて再構成され、アプリケーションにデータが渡される。この研究は上位層のアプリケーションや下位層と連携することなく通信するもので、特定の回線やアプリケーションに依存せず複数経路を利用することができる。そして、特にアプリケーションや有線か無線かなどの取り決めを行っておらず、汎用的なプロトコルを目指している。

以上より、複数経路を実現するには実際に通信を行う層とその上に管理する層をおき、輻輳制御は下位層で各々行うことが、標準的なアプローチであることがわかる。本稿でも、同じ二層構造を取るプロトコル上での提案手法の有効性について検討していく。

3. 複数経路プロトコル

無線環境における複数経路の同時利用を実現するプロトコルの構造について述べる。アプリケーションへの依存をなくすため、トランスポート層で実装を行う。トランスポート層での重要な機能としては、

- 輻輳制御
- パケットの分配制御

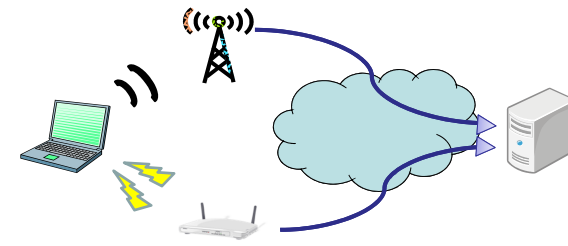


図 1 シナリオ
Fig.1 Scenario

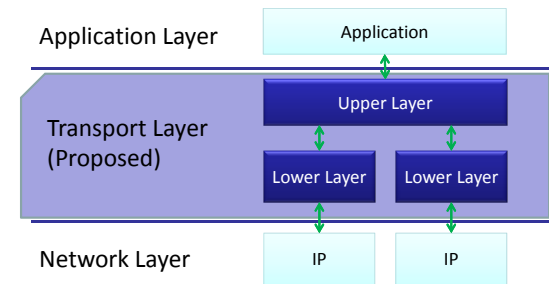


図 2 レイヤー構成
Fig.2 The architecture of Layer

があり、これらの機能について、本研究室で考案された R-M/TCP⁸⁾ をベースに検討を行う。また、提案手法として MAC 層から得られる RSSI を元に経路選択を行う機能を述べる。以下の図 1 のような環境を想定する。

3.1 概観

提案するレイヤー構成は以下の図 2 のようになる。

図のようにトランスポート層を上位副層と下位副層に分割して、複数経路での通信を実現する。下位副層では、一つ一つの経路に対する輻輳制御を行う。また、上位副層では下位副層での通信から得られる情報を元にフロー制御とパケット分配制御を行う。

3.2 輻輳制御

下位層において実現される輻輳制御について説明する。輻輳制御とは、ネットワークが混雑した状況になることを防ぐための機構である。

TCP の輻輳制御は、有線ネットワークで使用されるという前提をもとに設計されているため、パケット損は輻輳によってのみ起こると考えている。そのため、輻輳以外の無線区間によるパケット損などを特定できず、 unnecessaryな再送制御を行ってしまう問題点が指摘されている。そこで本稿では、通常の TCP に用いられている輻輳制御に比べ、パケットエラーなどによる不要な再送を行わないように提案された手法をとる。それは、RTT や到着頻度などからネットワークの状況を推定し、輻輳制御に利用するものである。

3.2.1 帯域の推定

帯域の推定は、輻輳制御を行うためにネットワークの状況を推定する機能である。RTT や ACK の到着頻度などの、エンドツーエンドで得られる情報を基に利用可能な帯域を見積もる。このような帯域の大きさを見積もる研究には、大きく二つの方法がある。前回の RTT の値とその間に受信した ACK のデータ量を用いる方法⁹⁾¹⁰⁾ と N 個前から現在までに届いたデータ量を ACK から取得し、N 個前に受信した ACK から現在のまでの時刻の時間を用いる方法¹¹⁾ である。本稿では短期間で見積もり帯域を更新する手法として後者を採用した。計算方法は帯域の推定値を bw とし、N 個前に ACK されたデータ量を $D(ACK_N)$ 、受信した時刻を T_N とすると、以下のように得られる。

$$bw = \frac{D(ACK_N) - D(ACK_i)}{T_N - T_i}$$

各パラメータは上の図 3 の部分を用いている。

3.2.2 輻輳検知時のウィンドウ制御

帯域の推定によって得られた値をもとに、輻輳検知時に行う処理について述べる。cwnd は輻輳ウィンドウと呼ばれ、送信者が管理するウィンドウサイズである。また、ssthresh は輻輳を検知した際に輻輳ウィンドウを変化させる値で、スロースタートの閾値である。本稿では、従来の TCP に用いられている高速リカバリアルゴリズムを採用した。詳しい挙動は以下になる。

(1) 2 つ目の重複 ACK

輻輳が起きる可能性があるかと判断し、パケット転送の停止を行い、cwnd を更新しない。

(2) 3 つ目の重複 ACK

損失したと判断されるパケットを再転送する。その際 cwnd と ssthresh を以下のように更新する。

$$cwnd = bandwidth \times Min_RoundTripTime$$

$$ssthresh = bandwidth \times Min_RoundTripTime$$

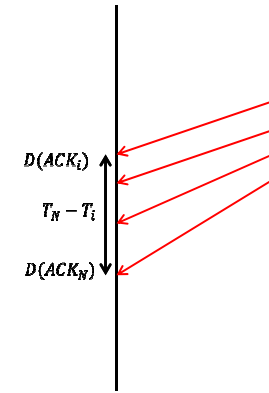


図 3 帯域の推定
Fig. 3 Bandwidth Estimation

bandwidth はパケットから見積もった値で、Min_RoundTripTime は RoundTripTime の最小値である。これらの数値の見積り方法については前節で述べた。

このようにして cwnd を素早くネットワークの状態に追従させるようにしている。これにより、無線区間でのパケットロスが原因で重複 ACK が届く場合も、帯域推定より輻輳が起きたわけではないと判断できる。輻輳ウィンドウを下げることにはなるが、従来の有線に向けた輻輳制御を用いる時に比べて、快適な通信ができると考えられる。上の図 4 のように推移するよう制御できることを目指す。

3.3 パケットの分配制御

上位副層で実装されるパケットの分配制御について述べる。パケットの分配制御とは、複数経路を用いた際のパケット分配とパケットをどのような順序で送信するか判断する機能である。これらの機能は、経路ごとにネットワークの品質が異なる環境において重要となる。異なるネットワーク品質の経路を用いて均等に分配した場合、以下の図 5 のように高速な経路から (1,3,5,7...) はデータが届くが、低速な経路から (2,4,6,8...) のデータが届かずボトルネックとなる。これにより、複数経路を利用しているにも関わらず、単一経路よりも通信効率を悪くしてしまう可能性もある。

図 5 のように、送信側ではパケットロスが発生していないにもかかわらず、受信側から届く重複 ACK により再送を行ってしまう、という問題点がある。本稿ではこれに対し、各経路で通信にかかる遅延時間を見積もり、分配を行うことで解決を試みる。加えて、無線環境では有

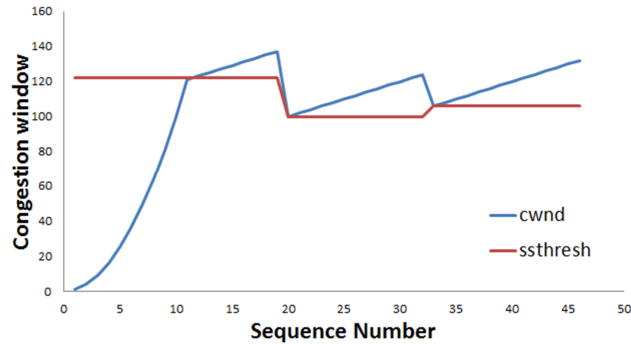


図 4 輻輳制御
Fig.4 Congestion Control

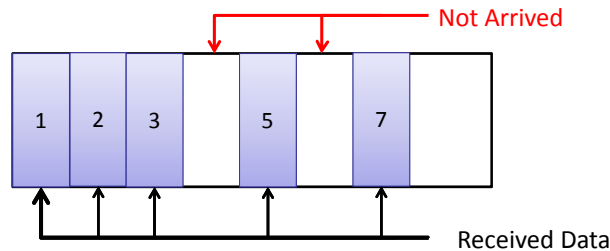


図 5 受信者のバッファ
Fig.5 Receiver's buffer

線環境にはないリンクエラーが起こる。AP から距離がある場合や遮蔽物により電波が届きにくい状況では、パケットロスが発生しやすく通信路の状態が悪い。このような無線区間特有のリンクエラーが起こりやすい状態を RSSI から判断し、パケットロスを防ぐ手法を提案する。その提案手法を述べる前に、実環境での RSSI とパケット損失の相関について述べる。

3.3.1 実験環境

パケットの損失率と RSSI との相関をみるために上の表 1 ような環境で、実験を行った。実験は屋内で行い、概要は図 6 に示す。アクセスポイントからの距離を変えて、数回データを採取した。一回のデータ取得には、ラップトップ端末で定期的に RSSI を記録し、アクセスポイントに対して ping を送り続けることによって行った。

表 1 実験環境

Table 1 Experiment environment

アクセスポイント	メーカー	Apple
	製品名	AirMac Express
	無線規格	802.11g

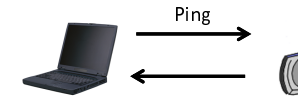


図 6 実験概要
Fig.6 Experimental overview

3.3.2 評価

RSSI とパケット到達率との関係を表したものは図 7 のようになる。実験から得られた値と共に、理論値を以下のように計算し、比較として載せた。

信号電力と雑音電力の比より、ビットエラーレート BER は以下のように求めることができる。

$$BER = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{SNR}{2}}$$

今回実験で用いた ping パケットのフレーム長は、実データ (56 バイト)+ICMP ヘッダ (8 バイト)+IP ヘッダ (20 バイト) であるため 84 バイトとなる。それにより、パケット到達率 P_a は以下のように求められる。

$$P_a = (1 - BER)^{(84 \times 8)}$$

これより RSSI とパケット到達率との関係があることがわかる。しかし、結果からわかるように理論値と実環境での数値には大きな差があるため、実際にパケット分配を行う際には、RSSI からパケット到達率を見積もるのではなく、パケット損失が起こるといったシグナルとして用いるのが適しているのではないかとと思われる。

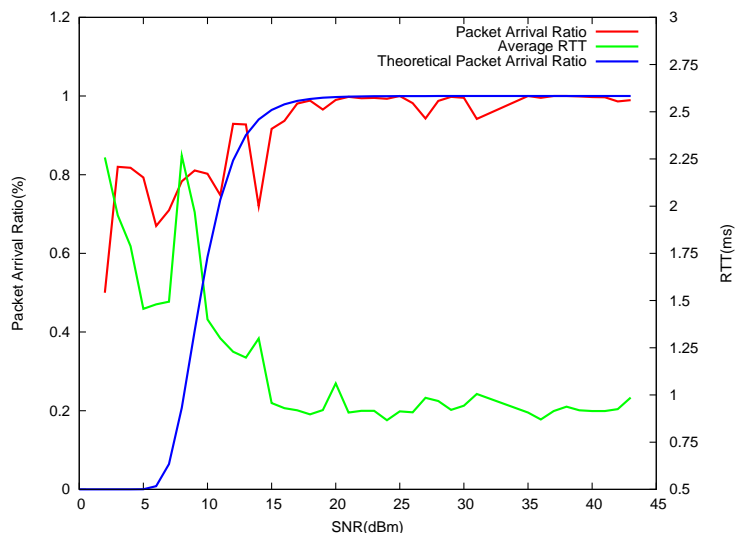


図 7 RSSI とパケット到達率
Fig. 7 RSSI vs Packet Arrival Ratio

以上を踏まえ、本パケットの分配制御において、問題となるのは以下であると考えられる。

- パケット損失を予測するシグナル
- 正常時のパケット分配方法
- パケット損失が起こると判断したときの分配方法

実際に RSSI をシグナルとして利用していく上で、シグナルを発するアルゴリズムを提案し、有効性を確かめなければならない。RSSI の値は、ベンダーによって基準が異なるため経験的に判断する必要が出てくる。そのため、機械学習やモデル化、クラスタリングなどの実測値からの推定をしなければならない。RSSI をシグナルとする手法については、今後の課題とし、パケット到達率の値によって以下に説明するパケットの分配方法が有効であるか確かめる。

3.3.3 RSSI が正常値の場合

RSSI から通信路は安定な状態と判断し、正常に通信可能と判断した場合のパケット分配手法について述べる。

各経路の片道にかかる時間を見積もり、受信側に順序通り届くよう分配するを行う。ACK が

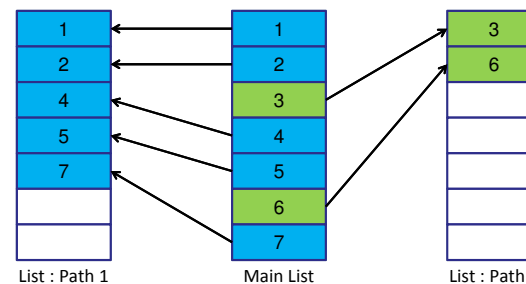


図 8 パケットの分配リスト
Fig. 8 Distribute Packet

届くたびに経路ごとにかかる時間 $OWTT$ を算出する。この $OWTT$ と、同じ経路から到着するパケットの到着間隔 $interval$ を用いて、到着時刻を見積もる。

$$OWTT + interval \times num$$

num は各経路において何個目のパケットなのかを表しており、この式が受信側に届く時間の見積りとなる。1 パケットごとにこの計算を毎回するのではなく、輻輳が検知されない間はネットワーク状況に急激な変化がないと捉え、パケットの並び替えを行わない。よって、輻輳検知時のみパケットを並び替える。並びかえの実装は、上位層においてパケットのリストを保持し各経路ごとにもリストを保持することで行う。分配のタイミングにおいて上位層のリストにいったん全て格納し、その後見積もった値から分配を行う。イメージは図 8 のようである。

3.3.4 RSSI からパケット損失が予想される場合

RSSI とは、無線通信機器が受信する信号の強度で、この値が下がることは端末が圏外になることや通信状況の悪化が予想される。本稿では、その RSSI の低下から通信経路の悪化を予想し、事前にパケットが損失することを防ぐ手法を以下に述べる。

輻輳が検知されていない経路を優先的に利用する 他方の経路の RSSI が正常であれば、パケット損失が起こると思われる経路を利用せず通信を行うことで、再送の発生を抑制できると思われる。ただし、両経路ともパケット損失が予想される場合には有効でないと思われる。

複数経路に同じパケットを送る 両方向にパケットを流すことで冗長化を行い、パケットを相手側に届ける方法である。今回対象とするアプリケーションは、リアルタイム性を重視したものであるため、適していると考えられる。また、両経路ともパケット損失

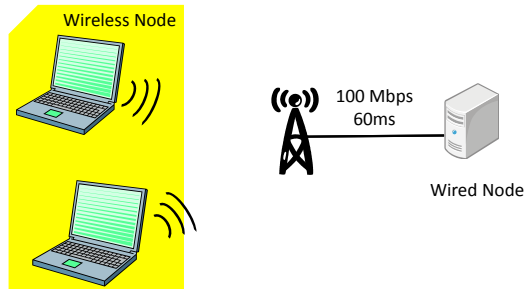


図 9 シミュレーションのシナリオ
Fig.9 Scenario of Simulation

が予想される場合においても、悪くない手法だと思われる。

3.3.5 シミュレーションの構築

先に述べた 2 つの手法について、ns2¹²⁾ を用いて比較、検討を行う。シミュレーションは、RSSI から通信状況が正常かどうか判断できるという前提で行う。無線通信区間内でのパケット損失は、ギルバートモデル¹³⁾ を用いてモデル化をを行った。シミュレータに標準で搭載されているエラーモデルでは、パケットを送信するごとに状態変化を行うよう実装されているが、実環境ではパケットを送信していない間にも通信路の状態が変化しうる。そのため、本稿ではある一定時間ごとに通信路の状態が変化するように実装を行った。

ギルバートモデルでは通信チャンネルが Good か Bad のどちらかの状態をとるため、Good を RSSI が正常な場合とし、Bad を RSSI からパケット損失が予想される場合とする。Bad な状態におけるパケット損失率を変化させシミュレーションを行った。シミュレーション環境としては以下の図 9 のように行った。また、シミュレーションに用いた各パラメータは以下の表 2 のようになる。

3.3.6 シミュレーションの結果と考察

シミュレーションによる結果を載せる。

図 10 より、同じ時間内で Normal(通常通り通信路の状態似関係なく送る手法),single(片方の通信路の状態が悪い場合に正常な通信路で送信する手法),multi(片方の通信路の状態が悪い場合に両経路で送信する手法) の 3 手法を比べると通信路の状態を把握した上で、避けることによりより多くのデータを送ることが可能となることがわかる。single と multi の手法には、ACK されたシーケンス番号に、大きな差は見られなかった。図 11 より、Normal 手法

表 2 シミュレーションに用いたパラメータ

Table 2 Simulation Parameter

無線規格	802.11g
Slot Time	9 μ s
SIFS	16 μ s
DIFS	34 μ s
CWmin	15
CWmax	1023

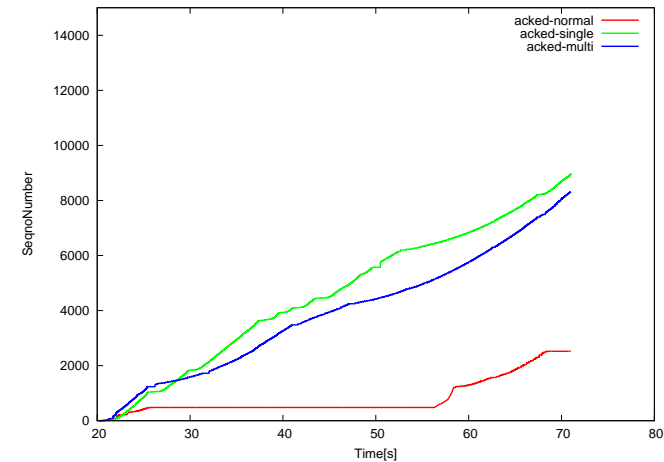


図 10 ACK されたパケットのシーケンス番号 N

Fig.10 Sequence Number of Aced Packet

では通信路が悪化した際には、cwnd の大きな変化が見られないことから幾つかのパケットが返ってくることによって、低速ながらも通信を行なっていることがわかる。本結果で手法 single が手法 multi に比べ、良い結果を残しているが性能の優劣につながる原因については、現在精査中である。

4. ま と め

RSSI を利用した無線環境における複数経路でのパケット分配を手法について検討を行った。まず、予備実験により RSSI とパケット到達率には関係性があることを示した。その上で、RSSI をパケット損失の予兆として使える可能性があると判断した。また、RSSI から通

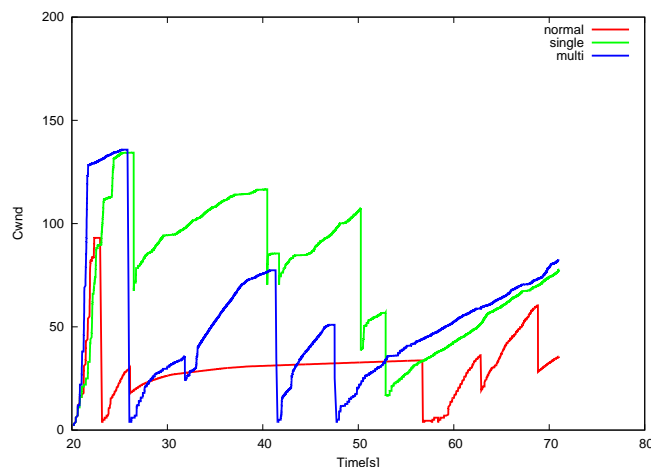


図 11 輻輳ウィンドウサイズ
Fig. 11 Congestion window

信状態が正常かどうか判断できるという前提のもとで、パケットの分配制御についてシミュレーションを行った。シミュレーション結果からは、複数経路が利用可能な場合においては、通信路の状態を把握し適切な分配を行うことで、通常の分配に比べ同時間内に多くのデータを送信することが可能であることがわかった。

しかし、本稿でのシミュレーションは対象な通信路におけるシミュレーションを行なっている。今後非対称な通信路での有効性を確認すると共に、RSSI をシグナルとして用いるためのアルゴリズムの提案や他にも下位層から得られる情報 (MAC 層での再送回数など) を元に同様なシグナルを出すことができないか考えていきたい。

参 考 文 献

- 1) Hsieh, H.-Y., Kim, K.-H. and Sivakumar, R.: An end-to-end approach for transparent mobility across heterogeneous wireless networks, *Mob. Netw. Appl.*, Vol.9, pp.363–378 (online), DOI:<http://dx.doi.org/10.1145/1012215.1012227> (2004).
- 2) Zhang, M., Lai, J., Krishnamurthy, A., Peterson, L. and Wang, R.: A transport layer approach for improving end-to-end performance and robustness using redundant paths, *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '04, Berkeley, CA, USA, USENIX Association, pp.8–8 (online),

- available from (<http://dl.acm.org/citation.cfm?id=1247415.1247423>) (2004).
- 3) Dong, Y., Pissinou, N. and Wang, J.: Concurrency Handling in TCP, *Proceedings of the Fifth Annual Conference on Communication Networks and Services Research*, Washington, DC, USA, IEEE Computer Society, pp.255–262 (online), DOI:10.1109/CNSR.2007.23 (2007).
- 4) Han, H., Shakkottai, S., Hollot, C. V., Srikant, R. and Towsley, D.: Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet, *Networking, IEEE/ACM Transactions on*, Vol.14, No.6, pp.1260–1271 (online), DOI:10.1109/TNET.2006.886738 (2006).
- 5) Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and Paxson, V.: Stream Control Transmission Protocol, *RFC 2960* (2000).
- 6) Iyengar, J.R., Amer, P.D. and Stewart, R.: Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths, *IEEE/ACM Trans. Netw.*, Vol.14, pp.951–964 (online), DOI:<http://dx.doi.org/10.1109/TNET.2006.882843> (2006).
- 7) 野澤高弘, 本多倫夫, 榊原 寛, 中澤 仁, 徳田英幸: ARMS: Application-level Concurrent Multipath Utilization on Reliable Communication, *Internet Conference 2008 論文集*, pp.108–113 (2008).
- 8) Rojviboonchai, K., Osuga, T. and Aida, H.: A Study on Rate-Based Multi-Path Transmission Control Protocol (R-M/TCP) Using Packet Scheduling Algorithm, *IEICE - Trans. Inf. Syst.*, Vol. E89-D, pp. 124–131 (online), DOI:10.1093/ietisy/e89-d.1.124 (2006).
- 9) Xu, W., Qureshi, A. and Sarkies, K.: Novel TCP congestion control scheme and its performance evaluation, *Communications, IEE Proceedings-*, Vol.149, No.4, pp. 217 – 222 (online), DOI:10.1049/ip-com:20020508 (2002).
- 10) Mascolo, S. and Grieco, L.: Additive increase early adaptive decrease mechanism for TCP congestion control, *Telecommunications, 2003. ICT 2003. 10th International Conference on*, Vol. 1, pp. 818 – 825 vol.1 (online), DOI:10.1109/ICTEL.2003.1191514 (2003).
- 11) 榎本 正, 渥美幸雄: ACK 到着間隔を使用した帯域推定方式の改良と評価, *電子情報通信学会技術研究報告*. CS, 通信方式, Vol.98, No.302, pp.43–48 (オンライン), 入手先(<http://ci.nii.ac.jp/naid/110003282740/>) (1998-09-25).
- 12) McCanne, S. and Floyd, S.: The Network Simulator NS-2, <http://www.isi.edu/nsnam/ns/>.
- 13) Gilbert, E.N.: Capacity of a burst-noise channel, *Bell System Technical Journal*, Vol.39, pp.1253–1265 (1960).