

## 無線環境における複数経路通信プロトコルの検討

坪井 祐樹<sup>†1</sup> 相田 仁<sup>†1</sup>

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きた。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いるが、有線に比べ十分な帯域を確保できていないのが現状である。そこで、本稿では、近年普及し始めている複数の無線規格をサポートしたモバイル端末において、同時にインターフェースを利用可能な通信プロトコルを提案する。まず、これまでに提案されてきた通信プロトコルの技術を概観し、それを元に無線環境でリアルタイム通信に適したプロトコルの概要を説明する。その後、プロトコルの機能として本稿で提案している RSSI による通信制御方式の有効性を示すための予備実験について述べる。その後、実験結果を踏まえた上で今後の課題を示す。

### Study of a Multi-path communication protocol in wireless environment

YUKI TSUBOI<sup>†1</sup> and HITOSHI AIDA<sup>†1</sup>

In recent times, we are able to access the Internet anywhere and anytime because of the advancement of wireless communication technologies. In tandem with these developments, demand for applications on mobile hosts which require wireless broadband capabilities or real-time communication are on the rise. However, at present, the transmission speed of a wired link is significantly lower than that of a wireless link. This issue necessitates the development of a transport protocol that can support the concurrent utilization of both wired and wireless channels in a multipath scheme, increasing both network bandwidth and stability. Several protocols that allow for multipath communication have been proposed, but most of them are based solely on wired networks. In this paper, we propose a transport protocol which integrates both wireless and wired transmissions in a multipath setting. Experiments have been conducted to evaluate the performance of the proposed protocol on the grounds of congestion control, bandwidth estimation, and packet distribution. Through the experiments, we analyze the effectiveness of RSSI in improving congestion control.

### 1. はじめに

無線通信技術の発展により、無線を用いてネットワークにアクセスするユーザーが増えてきている。その無線での通信手段には様々なものがあり、WiFi のような無線 LAN(Local Area Network) や WiMAX のような無線 MAN(Metro Area Network), HSDPA(HighSpeed Packet Access) などの無線 WAN(Wide Area Network) が提供されている。そして、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、複数のインターフェースを持ったノートパソコンやモバイル端末が登場してきている。複数のインターフェースを持った端末は、同時にそれらのインターフェースが接続可能な環境にあれば、複数の通信経路を利用可能である。しかし、一般的にはどちらかのインターフェースのみを利用し、通信を行っている。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、片方のインターフェースのみの通信は、資源の有効活用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、輻輳の発生防止につながる。本稿では、複数経路を耐障害性のためだけでなく、通常の通信においても利用することで、広帯域かつリアルタイム性を要求されるアプリケーションを利用可能にできると考えた。そして、複数経路通信可能な無線環境に適したトランスポートプロトコルの提案を行う。

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を概観する。第 3 節では、目的である複数経路同時利用のためのプロトコルについて説明し、どのように改善するか述べる。また、RSSI を用いた提案手法のための予備実験を行い、結果を踏まえた上でプロトコルの提案を述べる。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

<sup>†1</sup> 東京大学  
Tokyo University

実現されているものとして、pTCP<sup>1)</sup>, mTCP<sup>2)</sup>, cTCP<sup>3)</sup>, Multi-path TCP<sup>4)</sup> などがある。また、TCP ではなく SCTP(Stream Control Transmission Protocol)<sup>5)</sup> を改良したものとして、CMT<sup>6)</sup> や BA-SCTP<sup>7)</sup> 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 ARMS<sup>8)</sup> 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。

## 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。このようにして動的なアドレスの追加にも対応しており、並列転送を可能としている。

## 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が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。

## 2.3 ARMS

ARMS ではマルチホーム通信が可能なトランスポートプロトコルである SCTP を用いて、アプリケーション層での複数経路の同時利用を実現している。また、SCTP を選ぶ理由として、動的なアドレスの変化にも対応により、移動無線端末に適しているからである。この研究では SCTP を用いて、複数のパスに転送する比率を決めるアプリケーションを上位層に

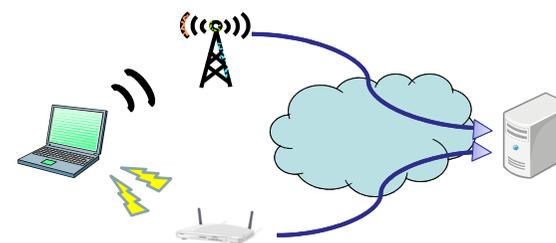


図1 シナリオ  
Fig.1 Scenario

実装している。この転送比率は、帯域幅と RTT(Round Trip Time) を考慮して決められている。実装結果から同じ性質のパスを複数用いた場合は理想的な結果を得ることができ、異なる帯域幅を用いた際には約 1 割のスループットの向上がみられている。

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。また、具体的なアプリケーションを想定して複数経路の通信を行おうとしている物は少ない。そこで、本稿では、無線環境においてリアルタイム性を必要とするアプリケーションに向けたプロトコルを目指す。次節からはそのプロトコルについて詳しく述べる。

## 3. 複数経路プロトコルの検討

本節では、無線環境において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。実装候補としての層はいくつかあるが、その中でもアプリケーションやハードウェアへの依存を少なくするため、トランスポート層を中心とした実装をめざす。提案するトランスポート層の機能として特に重要となるのは以上の 2 つである。

- 輻輳制御
- パケットの分配制御

これらの機能について、本研究室で考案された R-M/TCP<sup>9)</sup> 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。

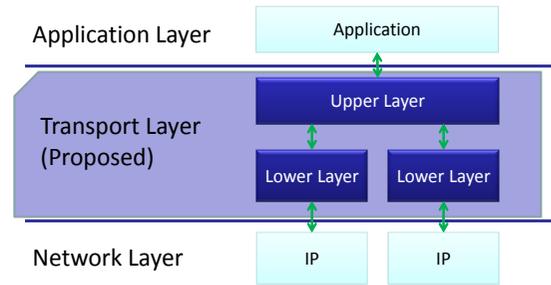


図 2 レイヤー構成  
Fig. 2 The architecture of Layer

### 3.1 概 観

提案するレイヤ構成は以下の図 2 のようになる。

図のようにトランスポート層を上位層と下位層に分割して、複数経路での通信を実現している。実際に通信を行うのは下位層になり、輻輳制御はここで行う。上位層においては、下位層からデータを収集することによりフロー制御とパケットの分配制御を行う。このようにトランスポート層において実現することによってアプリケーション層からは一つの接続として見えるので、従来の TCP と同様に扱うことが可能となる。

### 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、ネットワークが複雑化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、不必要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

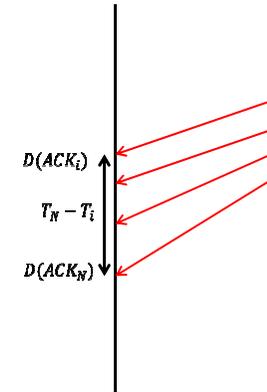


図 3 帯域の推定  
Fig. 3 Bandwidth Estimation

値とその間に受信した ACK のデータ量を用いる方法<sup>10)11)</sup> 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<sup>12)</sup> 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。詳しい挙動は以下ようになる。

- (1) 2 目目の重複 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 目目の重複 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

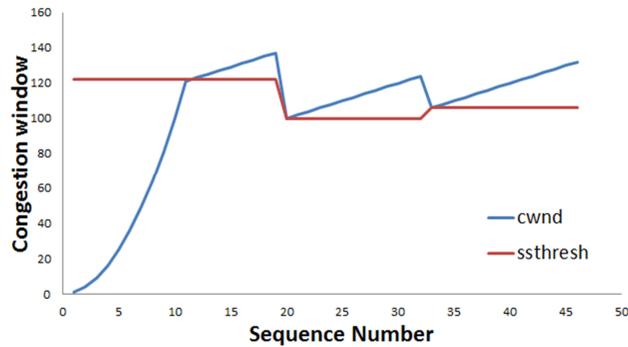


図 4 輻輳制御  
Fig.4 Congestion Control

に更新する.

$$cwnd = bandwidth \times Min\_RoundTripTime$$

$$ssthresh = bandwidth \times Min\_RoundTripTime$$

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目指す.

### 3.3 パケットの分配制御

上位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, アプリケーション全体の効率を下げってしまう.

このような状態になると, 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. これに対して, 経

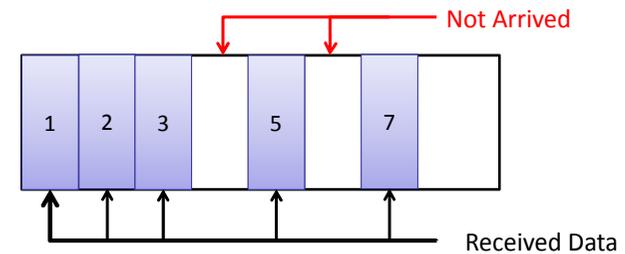


図 5 受信者のバッファ  
Fig.5 Receiver's buffer

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

路ごとのシーケンス番号を別で定義しパケットヘッダーに付加することによって, 不要な再送を起こさないという対処方法もあるが, 本稿ではリアルタイム性を重視するために上位層でどちらの経路を利用した方がリアルタイム性を維持できるのか判断して送信することにする.

また, 提案手法として RSSI を用いた通信制御について述べる. RSSI からパケットの損失が発生することを予想する手法である. 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.

#### 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. 図を見るとおり, RSSI が約-73dBm を下回るとパケット到達率が低くなるのがわかる. 加えて, パケットの送受信にかかった時間である RTT も安定しない結果が得られた.

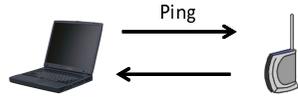


図 6 実験概要  
Fig. 6 Experimental overview

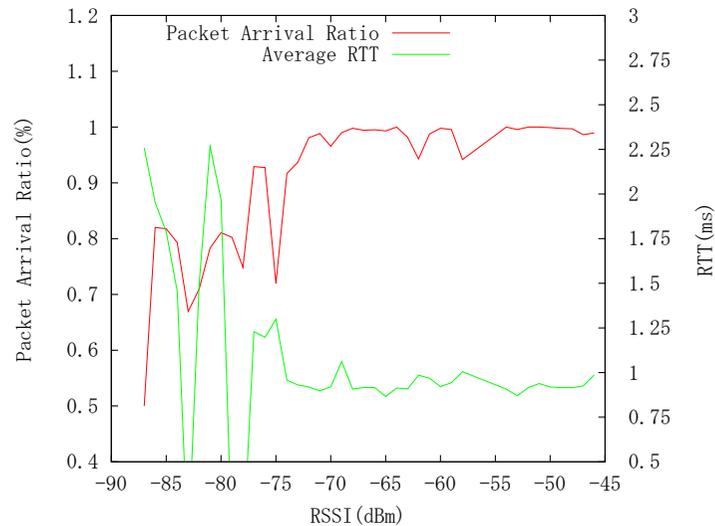


図 7 RSSI とパケット到達率  
Fig. 7 RSSI vs Packet Arrival Ratio

パケット損失があまり起きない状態と起きる状態をわける閾値 (今回は約-73dBm) を正しく設定もしくは見つけることができれば, RSSI による分配の制御が可能であると考えられる。

よって, 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。

- パケット損失が起ころうと判断する RSSI の閾値
- 正常時のパケット分配方法
- パケット損失が起これると判断したときの分配方法

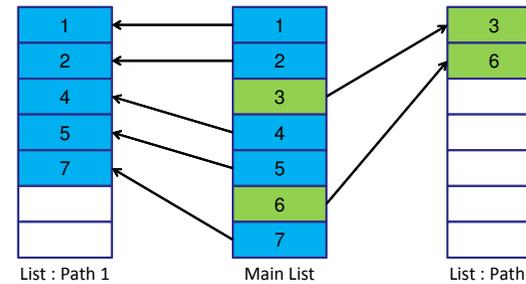


図 8 パケットの分配リスト  
Fig. 8 Distribute Packet

今回の実験での判断基準は結果を可視化することによって, 約-73dBm であると判断した。しかし, 実際にはこの閾値を設定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。本稿では, この問題については今後の課題とする。

2 つ目と 3 つ目の問題点についての提案手法を以下に述べる。

### 3.3.3 RSSI が正常値の場合

RSSI に変動がなく, 正常な状態で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アルゴリズムを提案する。ACK が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される。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としているため、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。

1 経路で全く同じパケットを送る パケット損失が予想される場合、同経路で全く同じパケットを送ることで、冗長化する手法。複数経路している際に、片方の経路で通信可能であれば不適切な手法だと考えられる。それは、複数経路を利用できるメリットを有効活用していない点と、同じ経路内のパケット量を増やすことで、輻輳を起こす原因ともなることから、不適切だと判断した。ただし、両経路ともパケット損失が予想される場合に、優位性が示せる可能性があると考えている。

以上の 3 つの手法について、ns2<sup>13)</sup> 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。無線通信区内でのパケット損失は、ギルバートモデル<sup>14)</sup> を用いてモデル化する。ギルバートモデルでは通信チャネルが 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。シミュレーション結果については、発表当日に紹介する。

## 4. ま と め

無線環境における複数経路を利用可能なプロトコルを検討した。中でも提案手法となる RSSI によるパケット損失のロールフォワード機能を実現するために、予備実験を行った。実験結果からは、適切な閾値を設定することが出来れば、パケットロストの予想をすることが可能であるとわかった。また、RSSI から通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。

今後は、RSSI からネットワークの状況が正常かどうか判断するアルゴリズムを検討すると共に、既存手法のままである輻輳制御について改善を検討する予定である。

## 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) Argyriou, A. and Madisetti, V.: Bandwidth aggregation with SCTP, *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, Vol.7, pp.3716–3721 vol.7 (online), DOI:10.1109/GLOCOM.2003.1258927 (2003).
- 8) 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).
- 9) 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).

- 10) 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).
- 11) 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).
- 12) 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).
- 13) McCanne, S. and Floyd, S.: The Network Simulator NS-2, <http://www.isi.edu/nsnam/ns/>.
- 14) Gilbert, E.N.: Capacity of a burst-noise channel, *Bell System Technical Journal*, Vol.39, pp.1253–1265 (1960).