

## 帯域集約のためのバッファ状態に応じたパス選択法の提案

小口 潔<sup>†</sup> 田中 大吾<sup>†</sup> 重野 寛<sup>‡</sup>

慶應義塾大学大学院理工学研究科<sup>†</sup> 慶應義塾大学理工学部<sup>‡</sup>

現在、無線アクセス技術の開発と普及が急速に進展しており、複数のネットワークに接続できるマルチホーミング環境が登場してきている。このマルチホーミング環境の利用方法の1つとして帯域集約が挙げられる。しかし、帯域集約を実現するのにあたり、パスごとに帯域幅、遅延時間、ジッタなどが異なることによりトランスポート層での性能が劣化し、帯域を十分利用できないという問題が発生する。本稿ではこの問題点に対し、バッファ状態に応じたパス選択法を提案する。提案手法では、送信バッファ中のパケットをチェックすることにより、各パスの状態を推測し、より送信能力のあるパスに送信を担当させることで帯域を有効利用する。また、シミュレーションによる評価により、提案手法がより送信能力のあるパスにデータを割り当てることができ、順序違いの影響を緩和できたことが示された。

### Path Selection using Buffer Conditions for Bandwidth Aggregation

Kiyoshi OGUCHI<sup>†</sup> Daigo TANAKA<sup>†</sup> Hiroshi SHIGENO<sup>‡</sup>

Faculty of Science and Technology, Keio University<sup>†</sup>

Graduate School of Science and Technology, Keio University<sup>‡</sup>

Recently, the introduction of new protocols such as IEEE 802.11a/b/g, 802.16a/e, Bluetooth and zigbee are making possible the deployment of multi-homing environment. One of the usage of multi-homing is bandwidth aggregation that makes transfer rate high and serves high quality service. However, the performance is degraded because of the deference of bandwidth, delay and jitter of the aggregated paths. In this paper, we propose the path selection using buffer conditions that selects the path having more transmission capacity. Using simulations, we evaluate proposal technique and it shows the better performance.

#### 1 はじめに

現在、無線アクセス技術の開発と普及が急速に進展しており、無線 LAN(WLAN) では IEEE 802.11a/b/g、無線 PAN(WPAN) では Bluetooth や ZigBee、無線 MAN(WMAN) では IEEE 802.16 が登場してきている。この無線アクセス技術の進展により、端末が複数の通信インターフェースを装備し、複数のネットワークに接続することが可能となってきている。このような環境を一般にマルチホーミング (multi-homing) 環境と呼ぶ。現時点では、端末が複数のインターフェースの中から1つだけ選択して利用する方法が一般的である。しかし、今後は複数のインターフェースを同時に使用したり、積極的に相互補完して使用したりすることが想定される。こ

のことによって、異なる無線アクセスネットワーク間のシームレスハンドオーバーや、複数パスの切り替えによる耐障害性の向上、複数パスを同時使用することによる帯域集約などの利用法が考えられる。しかし、帯域集約を実現するのにあたり、パスごとに帯域幅、遅延時間、ジッタ、パケットロス率などが異なることにより、パケットの到着順序が乱れ、性能が大きく劣化してしまうという問題点がある。本稿では、マルチホーミング環境での帯域集約実現において、TCP コネクションの送信バッファ中のパケット量においてパス状態を推定し、より送信能力を有するパスに送信を担当させて帯域の有効利用を行う、バッファ状態に応じたパス選択法を提案する。また、シミュレーションによる評価を行い、提案手法がより有効に帯域を利用できることを確認する。

## 2 帯域集約

図1に帯域集約の概念を示す。

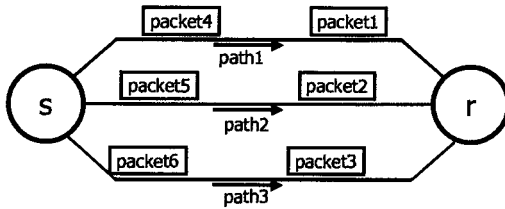


図1: 帯域集約の概念

データ送信時に送受信ホスト間に通信経路が複数存在するとき、送信パケットを各経路に振り分けることで帯域資源を有効に活用することを帯域集約と呼ぶ。また、個々の通信経路をパス (path) と呼ぶ。図1では、送信ホスト (s) と受信ホスト (r) 間に存在する3つのパスに対し、各パスにパケットを振り分け、帯域集約を行っている。

帯域集約では、複数のパスを束ねることで転送速度の飛躍的な向上が期待できる。また、集約の方法を戦略的に行うことで、ハンドオフ時に移動前のリンクから移動後のリンクへ帯域を移動させたり、帯域の安定しないパスを別のパスで補完して通信を安定化させたりすることができる。このように帯域集約技術は、マルチホーミング環境の利点であるシームレスハンドオーバーや通信の信頼性向上などと密接に関係する重要な技術である。

しかし、帯域集約を実現するのにあたり、パスごとに帯域幅、遅延時間、ジッタ、パケットロス率などが異なることにより、パケットの到着順序が乱れ、性能が大きく劣化してしまう。図2にパスの性能差によるパケットの到着順序の乱れの例を示す。この

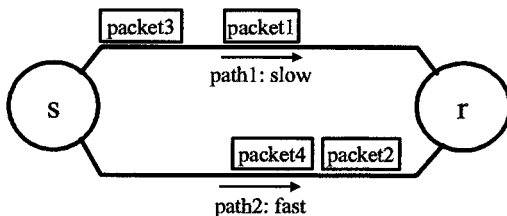


図2: パケットの到着順序の乱れ

例では、パス1の遅延時間がパス2の遅延時間に対して大きい場合、すなわち各パス間に性能差がある

場合を想定している。このとき、パケット1はパケット2よりも早く送出されたにもかかわらず、パケット2よりも到着が遅れ、到着順序が乱れてしまっている。パケットの到着順序が乱れることで、受信ホストはパケット2を受信しているにもかかわらずパケット1の受信を待たなければならず、帯域を十分利用することができない。パケットロス率や帯域幅が異なる場合にもこのパケットの順序違いが発生し、パス間での性能差が大きいほど順序違いが発生しやすくなる。

帯域集約を実現するのにあたり、パケットの順序違いによる性能低下の問題をどう解決するかが大きな課題となっている。

## 3 関連研究

複数の通信パスを利用した伝送とそれを利用した帯域集約は、リンク層、ネットワーク層、トランスポート層、アプリケーション層の各層で実施できる。

特にデータリンク層における帯域集約は古くから研究されており、ISDNやATM等でも実現されている。有線ネットワークではマルチリンク PPP、イーサネットにおけるスイッチ間、スイッチ・サーバ間の接続に用いられる IEEE 802.3ad などの実用例がある。また、無線リンクの集約についても様々な研究例が存在する。

ネットワーク層では、QoS ルーティング、マルチパスルーティングなどの、コンテキストでパケットを複数の経路を利用して伝送することが試みられている。しかし、ネットワーク全体におけるトラフィックの制御を対象としており、1つのコネクションのデータを複数のパスに振り分けた場合についての検討例は多くない。移動環境を対象として、トンネリング (IP-in-IP 化) を利用したパケットの複数パスでの配信方式や Mobile IP SHAKE などがある。

トランスポート層における研究では、TCP をベースとして、複数のコネクションを利用した帯域集約の研究が行われている。TCP は最も広く利用されているトランスポート層プロトコルであり、インターネットの全トラフィックの約 70% が TCP によって伝送されていると言われている。TCP は帯域集約に限らず、多くのトランスポート層プロトコルの出発点となっている。

トランスポート層でマルチホーミング環境を意識したプロトコルとして SCTP [1, 2] や Migrate-Permitted TCP オプションがあるが、これらは帯域集約を目指したものではない。

複数のコネクションを利用した帯域集約を実現するプロトコルとして、pTCP[3]やmTCPなどの研究例がある。TCPは比較的安定したネットワークではおおむね上手く動作するが、パケットの到着順序が乱れると大幅な性能劣化が発生する。これらの研究例では、このような問題に対応することで帯域集約を実現している。

アプリケーション層においても複数のTCPコネクションをまとめることで帯域集約を実現する技術が提案されている。しかし、アプリケーション層での集約では、集約されるパスにスループットの低いパスが含まれると、期待される集約効果を得ることはできない。例えば、ある一定の情報をダウンロードする場合を想定すると、ダウンロードにかかる時間は、複数のTCPコネクションを集約することで低減できる。しかし、全体の転送時間は、最もスループットの低いコネクションにおけるダウンロード時間に支配される。

## 4 提案

マルチホーミング環境において帯域集約を実現する際、パスごとのパラメータの違いによるパケットの順序違いの到着が大きな問題となる。本章では、パス状態に応じてデータを割り当てることで、パケットの順序違いの到着の影響を少なくし、帯域利用効率を上げるアルゴリズムを提案する。

### 4.1 順序違いの到着の影響

本提案では、各パスの帯域幅や遅延時間などのパス状態を考慮し、各パスへのデータ割り当て順序、データ割当量を調整することで、パケットの順序違いの到着自体を減らすのではなく、その影響を少なくしようと考えた。図2を図3のようにすることが目標である。

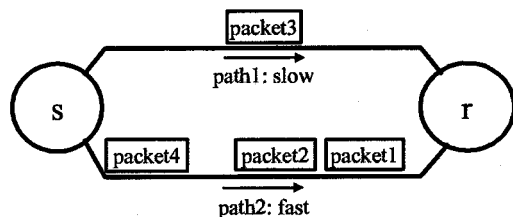


図 3: 順序違いの到着の影響の軽減

### 4.2 送信バッファとパス状態

パスがアイドルな時とビジューな時のパスと送信バッファの状態のイメージ図を図4に示す。

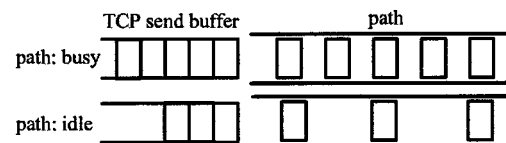


図 4: 送信バッファとパス状態のイメージ図

図4の上のように、帯域を最大限利用しておらず、パスにパケットが詰まっていない状態であるときは、それに伴ってTCP送信バッファ中のパケット量も少なくなっていると考えられる。また逆に、パケットが詰まっている状態のときは、TCP送信バッファからパスへパケットが送られづらく、TCP送信バッファ中のパケット量も多い状態であると考えられる。

また、送信バッファ中のパケット量とTCP輻輳ウィンドウの関係を考えてみる。送信バッファ中のパケット量が大きい場合、TCPコネクションは以下の状態のどちらかであると考えられる。

- パケットを送信し、ACKを待っている状態(輻輳ウィンドウ大)
- パケットロスが発生し、パケットが送れない状態(輻輳ウィンドウ小)

送信バッファ量が小さい場合、以下の状態のどちらかだと考えられる。

- コネクション確立直後の状態(輻輳ウィンドウ小)
- ACKパケットが返ってきて送信バッファからパケットを消去した状態(輻輳ウィンドウ大)

この4つの状態を考えたとき、まず、パケットロスが発生してパケットが送れない状態であるときは、送信能力がないと判断できる。また、ACKが返ってきて送信バッファからパケットを消去した状態であるときは、送信能力があると判断できる。そして、コネクション確立直後であるときは、使用可能な帯域を推定するためにスロースタートを行う段階であるので、送信能力があると判断できる。

ここで問題となるのが、パケットを送信し、ACKを待っている状態のときであり、この状態のときは送信能力が残っているかどうか判断できない。この

状態から ACK が無事到着すると、送信バッファからパケットを消去でき、送信能力がある状態になるが、パケットがロスしてしまっていた場合、送信能力がない状態となる。よって、この ACK 待ちの状態であるときには、ACK またはパケットロスを待つべきだと考えられる。

以上から、送信バッファ中のパケット量が大きいときには送信能力がない(もしくは判断できない)と考えることができ、パケット量が小さいときには送信能力があると考えられることができる。

### 4.3 バッファ状態に応じたパス選択

本提案方式では、TCP コネクションへのデータ割り当て時に TCP 送信バッファをチェックし、パケット量が最も少ないパスを最も送信能力のあるパスと判断し、データを割り当てる。このパス選択はコネクションにデータが割り当てられる度に行われ、その時点で最も送信能力のあるパスにデータが割り当てられることとなる。

### 4.4 提案方式の動作

提案方式の詳細を図 5 に示す。

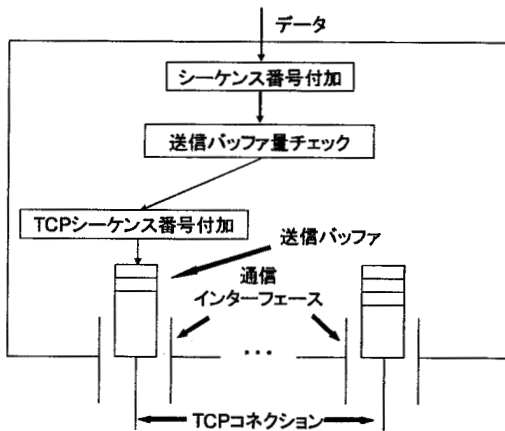


図 5: 提案方式モデル

提案方式では、複数の有線・無線パスを想定し、それぞれのパスに 1 本ずつ TCP コネクションを確立させ、コネクション単位でフロー制御・輻輳制御を行う。コネクションはそれぞれのフロー制御・輻輳制御に関する情報を管理し、提案方式はその情報を操作することはない。TCP コネクションのオープンにはアプリケーションからオープン要求が来た際に行い、使用可能な通信インターフェースを全て割り

出してそれぞれスリーウェイハンドシェイクを行う。その際コネクションが確立できなかったパスはその後使用されず、またコネクションの再確立も行われず、確立されたコネクションは提案方式で管理される。

以下に動作例を示す。

**コネクション確立** 自ノードの使用可能な通信インターフェースを全て割り出し、サーバ側では Passive Open 処理を実行して TCP コネクションは LISTEN 状態に、クライアント側では Active Open 処理を実行して SYN\_SENT 状態になる。その後は全てのコネクションでスリーウェイハンドシェイクを実行し、全てのコネクションが確立されるまで待機する。

**送信処理** アプリケーションから送信すべきデータを受け取ると、まずデータパケットにシーケンス番号を付加する。そして全てのアクティブな TCP コネクションの送信バッファをチェックする。そして送信バッファ中のパケット量を比較し、パケット量が一番少ないコネクションにデータを割り当てる。データを割り当てられたコネクションは TCP シーケンス番号を付加し、輻輳制御に従って送信する。

**受信処理** TCP コネクションはデータを受信すると、まずコネクション内で順序制御を行う。その後全体での順序制御を行い、アプリケーションにデータを渡す。

**コネクション切断** アプリケーションからクローズ要求を受け取ると、全てのアクティブな TCP コネクションを解放する。

## 5 評価

### 5.1 シミュレーション環境

ネットワークシミュレータ QualNet[4] に、提案方式を含む帯域集約環境を構築した。本シミュレーションでは、データ分配をラウンドロビンで行う方式と提案方式の比較を行う。図 6 にシミュレーションモデルを示す。提案方式をトランスポート層に構築し、また提案方式用のバルク転送アプリケーションをアプリケーション層に構築した。

シミュレーショントポロジを図 7 に示す。このモデルでは、2 つのノード (sender, receiver) それぞれが 2 つの通信インターフェースを持ち、2 本のパスで接続されている環境を想定した。この 2 本のパス

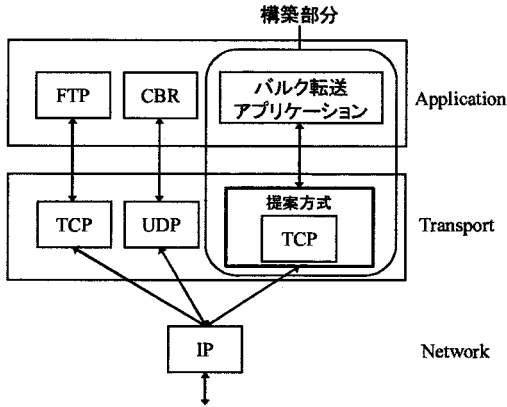


図 6: シミュレーションモデル

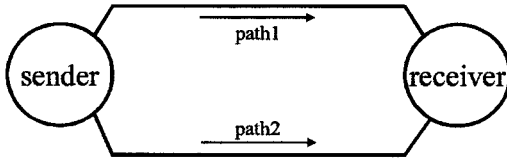


図 7: シミュレーショントポロジ

に対し、path1 と path2 にそれぞれ 1 本ずつ TCP コネクションを張り、通信を行う。

全シミュレーションで共通の条件・パラメータを表 1 に示す。

シミュレータ	Qualnet 3.9.0
シミュレーション時間	100 sec
有線帯域	10 Mbps
有線伝播遅延	1 msec
送信・受信バッファサイズ	16384 byte

また、本シミュレーションでの参考値として用いる、有線、IEEE 802.11a/b 単独での 100 秒間の平均スループットを測定した。それぞれのスループットを表 2 に示す。

本シミュレーションでは以下の 2 項目について評価を行う。

#### 帯域幅の変化に対するスループットの推移 :

path1 に IEEE 802.11b, path2 に有線リンクを用い、30 秒から 60 秒まで 500Kbps の UDP フ

表 2: 伝送方式ごとの平均スループット

伝送方式	スループット [Kbps]
有線リンク	8824.002
IEEE 802.11a	3231.674
IEEE 802.11b	941.207

ローを流し、帯域幅が変化したときのラウンドロビン方式と提案方式のスループットの比較を行う。

#### 帯域幅の変化に対する平均スループット :

path1 に IEEE802.11a または b, path2 に有線リンクを用い、path1 に UDP フローを流し、UDP フローのデータ転送速度を変えて帯域幅が変化したときのラウンドロビン方式と提案方式の平均スループットの比較を行う。

## 5.2 帯域幅の変化に対するスループットの推移

図 8 に 30 秒から 60 秒まで UDP フローを流し、帯域幅が変化したときのスループットの推移を示す。

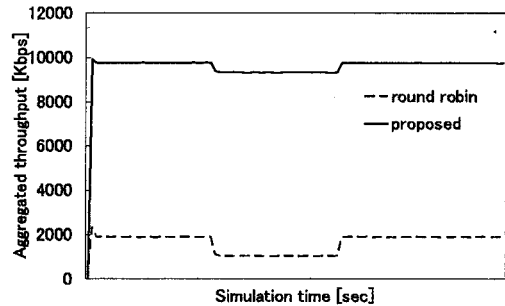


図 8: 帯域幅の変化に対するスループットの推移

図 8 を見ると、30 秒から 60 秒で path1 の使用可能な帯域が 500Kbps 減ったのに対し、ラウンドロビン方式ではスループットの同期が起ってしまい、全体としては 1000Kbps 程度スループットが落ちているのに対し、提案方式では 500Kbps 減程度で抑えている。これは、帯域幅の変化に対し、送信バッファ中のパケット量を観測することでデータ分配パスを選択でき、より送信能力のあるパスにデータを割り当てられたためだと考えられる。

また表2と見比べても、提案方式は10Mbps弱のスループットを得られていることから、それぞれのパスのスループットの合計程度のスループットが得られ、帯域集約機能を十分に持っていることがわかる。

### 5.3 帯域幅の変化に対する平均スループット

図9にシミュレーション結果を示す。ここではpath1にIEEE 802.11aを用い、path1のUDPフローのスループットを、0Kbps, 100Kbps, 200Kbps, 400Kbps, 500Kbps, 800Kbps, 1000Kbps, 2000Kbps, 4000Kbpsと変化させてシミュレーションを行った。

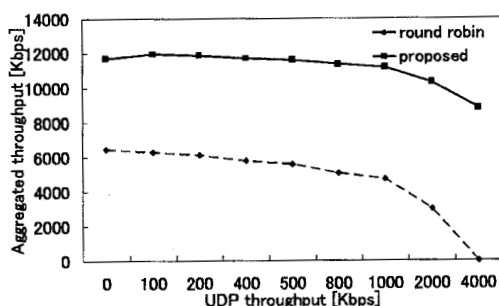


図9: 11aと有線リンクの帯域幅の変化に対する平均スループット

図9を見ると、ラウンドロビン方式では使用可能な帯域の減少量の二倍の速度でスループットが減少していきってしまうのに対し、提案方式では帯域の減少量のみ影響に抑えていることがわかる。これにより、パスの状態をバッファ中のパケット量によって把握し、適切にパスを選択することでスループットを維持できていることが確認できた。

また、図10にpath1にIEEE 802.11bを用いたシミュレーション結果を示す。path1のUDPフローのスループットを、0Kbps, 100Kbps, 200Kbps, 400Kbps, 500Kbps, 800Kbps, 1000Kbpsと変化させてシミュレーションを行った。

図10を見ると、図9と同様に、提案方式ではスループットが維持できていることが確認できる。

## 6 おわりに

マルチホーミング環境を利用して帯域集約を実現するのにあたり、パスごとに帯域幅、遅延時間、ジッタなどが異なることによりトランスポート層の性能

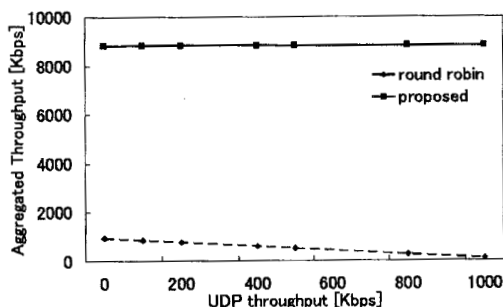


図10: 11bと有線リンクの帯域幅の変化に対する平均スループット

が劣化し、帯域を十分利用できないという問題が発生する。この問題点に対し、本稿ではバッファ状態に応じたパス選択法を提案した。

提案手法では、データ送信時の送信バッファ量をチェックし、送信バッファ中のパケット量の一番少ないパスを送信能力の高いパスと判断し、データを分配する。この手法により、パスの帯域の有効利用が可能となる。

コンピュータシミュレーションを用いて評価を行った結果、提案手法は帯域を十分使用することができ、パスごとの単独のスループットの合計に相当するスループットを示すことができた。また、通信環境の変化に対し、バッファ状態からパス状態を推定でき、より送信能力の高いパスへデータを割り当てることが確認できた。以上より、提案手法は帯域集約での問題点を克服するのに有効な手段であるといえる。

## 参考文献

- [1] W. Guo S. Cheng H. Huang J. Shi, Y. Jin and D. Zhang. Performance evaluation of sctp as a transport layer solution for wireless multi-access networks. pp. 453-458, 2004.
- [2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. Stream Control Transmission Protocol. RFC 2960, , Oct. 2000.
- [3] H.-Y. Hsieh and R. Sivakumar. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. *ACM/IEEE MOBICOM'02*, Sep.23-28,2002.
- [4] Qualnet, qualnet user manual. URL: <http://www.scalable-networks.com>.