ソケット層における帯域統合機構

榊原 寛[†] 守分 滋[‡] 斉藤 匡人[‡] 徳田 英幸[†,‡]

[†] 慶應義塾大学 環境情報学部, [‡] 慶應義塾大学大学院 政策・メディア研究科
{skk, duh, masato, hxt}@ht.sfc.keio.ac.jp

近年通信技術が急激に発達している．それに伴い計算機に複数の通信デバイスが接続される環境が増えてきた．現在これら通信デバイスは一度に一つしか利用されていないことが多い．そこで，本稿では，計算機に接続された通信デバイスを有効に利用するため，それぞれの通信デバイスを同時に利用し帯域を向上させる，ソケット層における帯域統合機構 SBAM (Socket-level Bandwidth Aggregation Mechanism) を提案する．SBAM をオペレーティングシステムにおけるソケット層で実現することにより，トランスポート層などのネットワークスタックやアプリケーション層において実現する際に生じる欠点を回避し，既存ソフトウェアの変更を必要とせず，効率的に帯域統合を実現できる．

Socket-level Bandwidth Aggregation Mechanism

Hiroshi Sakakibara[†] Shigeru Moriwake[‡] Masato Saito[‡] Hideyuki Tokuda[†,‡]

[†]Faculty of Environmental Information, Keio University
[‡]Graduate School of Media and Governance, Keio University

Due to the recent explosive growth of network technologies, the number of network devices that are connected to computers have increased. However, these network devices are only used one at a time. This paper describes Socket-level Bandwidth Aggregation Mechanism (SBAM), which aggregates the bandwidth of all the devices connected to the computer. Since SBAM is implemented in socket layer of the operating system, it can avoid the drawbacks caused by the implementation in network stack or application layer and achieves bandwidth aggregation efficiently without changes to existing software.

1 はじめに

近年，通信技術の発達にともない，さまざまな規格が制定されている．例えば，802.11b [7]，802.11a [6]，Bluetooth [1]，Ultra WideBand (UWB) [13] などが挙げられる．新たな通信規格が制定されるに従い，一つの計算機に複数のデバイスが接続している状況が増えている．しかし，それら複数の通信デバイスを有効に使っていないことが多い．例えば無線 LAN を利用している間は，PHS によるデータ通信サービスは利用しない．つまり，複数の通信デバイスを搭載している計算機があったとしても，片方を利用している間にもう片方のデバイスは利用しないことが多く，資源が有効に活用されていない．

本稿では，これら計算機に接続された通信資源を有効に利用するため，それぞれの通信デバイスを同時に利用し帯域を向上させる，ソケット層における帯域統合機構 SBAM (Socket-level Bandwidth Aggregation Mechanism) を提案する．

本稿の構成として，まず 2 節で SBAM の設計について述べる．そして，第 3 節で本稿で行なったプロトタイプ実装について述べ，第 4 節で評価を行なう．第 5 節で関連研究を紹介し，最後に第 6 節でまとめと今後の課題を述べる．

2 設計

本節ではまず本研究が想定する環境について説明し，そこで発生する問題を述べる．次に問題点を踏まえた上で設計方針を示し，SBAM のシステム構成とその要素機能について述べる．

2.1 想定環境

本小節では，本研究がターゲットとする環境について説明する．本研究では図 1 に表されるように，異なる無線ネットワークが同一地域に存在し，そのネットワーク内をユーザが移動している状態を想定する．ユーザは無線ネットワークインタフェース (以下 N/I) が複数接続されているラップトップ PC や PDA などの計算機を利用しているとする．図 1 では Bluetooth と 802.11b が接続されている．このような状態において，特にユーザ b のように，複数の無線 N/I を利用できる状態を本研究では扱うこととする．

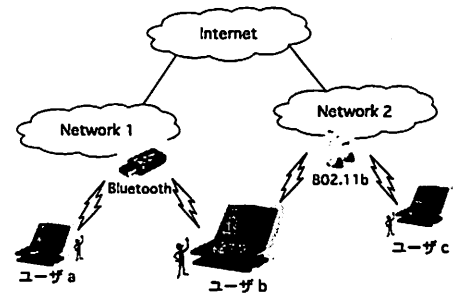

図 1: 想定環境

次小節以降，このような環境から生じる問題点を挙げ，設計の方針について述べる．

2.2 問題点

本小節では，帯域統合機構を実現する際に生じる問題点を 3 つ述べる．

1. 様々な無線デバイスの利用を想定していることから，各 N/I から送信先ホストまでのリンクの状態が異なるため，単純なラウンドロビンスケジューリングではデータ送信を行なえない．例えば，帯域が異なるリンクに対してラウンドロビンスケジューリングでデータ送信を行なった場合，速いリンクは遅いリンクのデータ送信が完了するまで，次の送信データを割り当てられないので，2 つのリンクの帯域を効率的に利用できない．
2. 無線デバイスの特性に合わせた利用要求に応えられなければ実用的なシステムにならない．例えば，データ転送量に応じて課金される PHS データ通信サービスと無線 LAN が接続されている計算機を持ったユーザが，無線 LAN の利用できる環境に移動した場合，帯域を向上指せるために 2 つの N/I を同時に利用するよりも，課金されずに通信を行

なうために無線 LAN のみを利用したいという要求に応える必要がある.

3. データが複数の経路を通って通信先ホストに届くので，データの整合性を保つのが難しくなる．送信元ホストにおいて，各リンクに対しデータを順番に送信しても，途中経路の状態によって送信先ホストではその順番通りにデータが到着するとは限らない.

2.3 設計方針

帯域統合機構を実現するネットワークスタックとしては，データリンク層，ネットワーク層，トランスポート層，アプリケーション層などが考えられるが，本研究では OS の機能である ソケット層における実現手法を採った．以下，各ネットワークスタックにおける利点・欠点を挙げ，ソケット層における実現手法の優位性を示す.

ネットワーク層/データリンク層

帯域統合機構を実現するには，エンドノード間のネットワークの状態を把握し，その情報に応じて各 N/I からデータ送信を行なう必要がある．Hop-by-Hop で処理されるプロトコルではエンドノード間のネットワーク状態を取得できないので，ネットワーク層やデータリンク層における実現は現実的ではない．但し，帯域や遅延などのリンク状態が，各 N/I において全く同一ならば，単純なラウンドロビンスケジューリングを用いることで帯域統合を実現でき，複雑な処理が必要ないという利点が存在する.

トランスポート層

新たなトランスポートプロトコルを実装した場合，3 つの欠点があげられる．1) そのプロトコルを利用するために，既存アプリケーションを書換える必要がある．2) 相手ホストでそのプロトコルを利用できない場合，通信は不可能である．この場合，アプリケーションにおいて動的に利用プロトコルを切替える必要がある．3) TCP を改良する場合，TCP の保持している帯域・遅延情報を利用し，効率的に帯域統合機構を実現できるという利点が存在するが，TCP はインターネット上で広く利用されており，全てを新たなトランスポートプロトコルに置き換えるのは現実的ではない．しかし置換を行なわなければ通信できないホストが発生してしまう.

アプリケーション層

Hung-Yung Hsieh ら [5] によると，アプリケーション層において帯域統合機構を実現した場合，N/I の数に対するスケーラビリティが低いという結果が報告されている．また，既存のプログラムで帯域統合を実現する場合，そのプログラムに帯域統合機構を実装する必要がある.

ソケット層は OS の機能でありながら，ネットワークスタックではないので，ネットワークスタックにおける既存のプログラムの変更という欠点を回避しつつ，アプリケーション層における欠点も回避できる．以下にソケット層における利点を挙げる.

- 既存ソフトウェアの変更が不要
- End-to-End でネットワークの状態の把握が可能
- N/I の増加に対して，パフォーマンスが落ちない
- 既存のトランスポート層を利用するので，新たなトランスポートプロトコルが開発されても対応可能

また，通信開始時に帯域統合機構は，通信先ホストにおいて同機構の有無を調べるので，ネットワーク内に徐々に設置できる.

以上の理由により，SBAM では ソケット層における実現手法を採った.

2.4 構成要素

本小節では前小節の問題点と設計の方針を踏まえ，SBAM のシステム構成について述べる．システム全体の動作概要について述べた後，各機能の説明を行なう.

2.4.1 動作概要

各機能がどのように連係し動作するかについて述べる．図 2 にシステム構成図を示す.

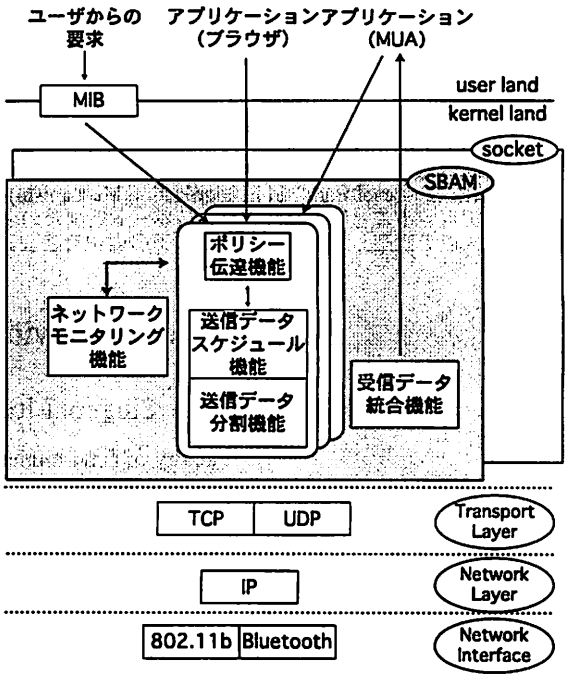

図 2: システム構成図

データ送信時にはまず，ポリシー伝達機能が MIB [2] (Management Information Base) から情報を読みとり，その値を送信データスケジュール機能に渡す．次に，ネットワークモニタリング機能が相手先ホストまでのネットワーク状態の計測を開始し，その値を送信データスケジュール機能に渡す．計測は定期的に行なわれ，値は適宜送信データスケジュール機能に渡される．送信データスケジュール機能では，利用可能 N/I を調査し，ポリシ伝達機能とネットワークモニタリング機能から得た情報をもとに，各 N/I に対しどのようにデータを配分するか決定し，その情報を送信データ分割機能に渡す．送信データ分割機能では，渡された値をもとに送信データの配分を決定し，SBAM ヘッダを付加した後，下位トランスポートレイヤにデータを渡し，データの送信を行なう.

受信時には，SBAM ヘッダを取り除き，パケット順の並べ替えを行なった後，データを統合しアプリケーションにデータを渡す.

2.4.2 ポリシー伝達機能

本機能は，MIB を通してカーネル空間とユーザ空間の間でユーザポリシのやりとりを行う．ユーザポリシとは，ユーザの各 N/I に対する利用要求である．2.2 節の 2 つ目の問題点で挙げた例のような場合に必要となる機能である.

2.4.3 送信データスケジュール機能

送信データスケジュール機能には，送信データ再スケジュール部と各 N/I の状態に応じたデータ送信部がある．本節ではそれぞれについて述べる.

送信データ再スケジュール部

何らかの理由で N/I が利用不可能になった場合，他の利用可能な N/I に送信データの再スケジュールを行う．また，利用可能な N/I の把握も行なう．例えば，ユーザが N/I をとりはずしたり，電波が届かなくなり通信を行なえなくなった場合，送信データの再スケジュールを行なう．再スケジュールを行う際，図 3 のように，キュー内のデータを並べ替える．図 3 は N/I A と B

が存在し，それぞれがキュー内にデータを保持していることを示している．このような状況で N/I B が利用不可能になったとする．この場合，N/I B のキュー内の送信データは図のように，N/I A のキュー内で並べ変えられ送信される．このことにより，送信先ホストにおけるデータの並べ替え作業の軽減を行う．

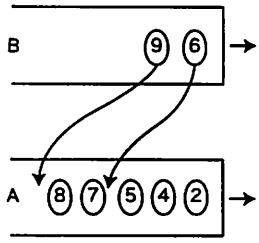

図 3: キュー内のパケット例

各 N/I の状態に応じたデータ送信部

後述のネットワーク状態モニタリング機能より，各リンクの帯域・遅延情報を受けとり，MTU (Maximum Transfer Unit) を考慮しつつデータ配分量を決定する．まず，通信開始時には，各リンク間の帯域遅延積差をなくすため，帯域遅延積が最小のリンク以外に対し，式 (1) 中の P_n で表されるパケット数を送信する．d_n は遅延，b_n は帯域，m_n は MTU をそれぞれ表す．また，α_n は，ポリシー伝達機能によって指定された各 N/I の重みを表す．

$$P_n = \frac{\alpha_n b_n d_n - min(\alpha_1 b_1 d_1 ... \alpha_n b_n dn)}{m_n} \quad (0 \leq \alpha \leq 1) \quad (1)$$

この操作により，各リンク間の帯域遅延積の量が一定になる．次に，各リンクの帯域に比例したデータ量を送信する．データ量に比例したパケット数を N/I に対して割り当てるため，パケット数は式 (2) 中の P_n で表されるパケット数を送信する．

$$P_n = \alpha_n \frac{b_n \cdot lcm(m_1 \cdots m_n)}{m_n \cdot gcd(b_1 \cdots b_n)} \quad (2)$$

例えば，帯域が 2Mbps，MTU が 500 バイト，遅延が 100ms のリンク A と，帯域が 1Mbps，MTU が 1000 バイト，遅延が 50ms のリンク B が存在したとする．この場合，通信開始時はリンク A に対して，式 (1) より，$(2000000(bps) \cdot 0.1(sec) - 1000000(bps) \cdot 0.05)/(500 \cdot 8) = 37.5 \fallingdotseq 38$ パケット送信する．次に，各リンクに対する送信パケット数は，式 (2) より，リンク A に対して $(2000000 \cdot 1000)/(500 \cdot 1000000) = 4$ パケット，リンク B に対して同様に，1 パケットとなる．このことにより，それぞれのリンクのパイプを埋め，効率的に各リンクを利用できる．

2.4.4 ネットワーク状態モニタリング機能

本機能は前項で述べた送信データ分割機能と送信データスケジュール機能のために送信先ホストまでのネットワークの状態をモニターする．具体的には，通信相手までの遅延，帯域，パケットロス率を取得する．遅延データは，式 (3) で表される TCP で利用されている遅延の平滑化手法を利用し保持する．$SRTT$ は平滑化された Round Trip Time (RTT) を表し，RTT は計測された RTT を表す．α は平滑化係数であり，TCP における推奨値は 0.9 となっている．

$$SRTT = \alpha SRTT + (1 - \alpha) \times RTT \quad (0 \leq \alpha \leq 1) \quad (3)$$

帯域情報の取得には，packet pair 方式 [9] を利用する．この方式ではまず，2 つの同じ大きさのパケットを同時に送信先ホストへ送信する．そしてこの 2 つのパケットの ack の到着時間の差より，送信先ホストまでの帯域を測定する．

$$\text{帯域} = \frac{\text{パケットサイズ}}{ack \text{ の到着時間の差}} \quad (4)$$

表 1: 実装環境

マシン	ThinkPad X30, ThinkPad T30
OS	FreeBSD 5.1-Release
NIC	BUFFALO WLI-PCM-L11, Intersil Prism2.5

また，本機能は帯域統合機構が動作するホスト内において，一つだけ動作する機能である．送信データスケジュール機能はアプリケーション毎に動作する必要があるが，本機能はホスト内において統一的にネットワーク状態をモニターし，効率的にその情報を他の機能に渡す．

2.4.5 送信データ分割機能

送信データを分割し，適切なポリシーに従ってトランスポート層に渡すための機能である．適切なポリシーとは，後述のネットワーク状態モニタリング機能から得られる遅延情報やパケットロス率や帯域情報，そして，ポリシー伝達機能から得られるユーザポリシーなどから決定する．また，受信データ統合機能においてデータの再構築を行なうための SBAM ヘッダの追加も行なう．

2.4.6 受信データ統合機能

本機能は，分割されて送られてきたデータを統合する機能である．後述の SBAM ヘッダを読みとり，パケットの並べ換えを行ない，受信データをアプリケーションに渡す．

パケットの再構築をどのように行なうかは，下位レイヤーで利用されているトランスポートプロトコルによって 2 つに分けられる．TCP や STP [12] のように信頼性が求められるプロトコルが利用されている場合は，ヘッダ情報を利用して正確にデータを再構築する．これに対し UDP のような信頼性は必要とされていないが高速なネットワーク転送と少ない遅延が求められるプロトコルが利用されている場合は，パケットの並べ換えよりもアプリケーションに対してデータを早く渡すことを優先する．

3 プロトタイプ実装

本節では SBAM のプロトタイプ実装について述べる．今回の実装では，送信データ分割機能と受信データ統合機能を実装した．また，利用するトランスポートプロトコルは UDP とした．

3.1 実装環境

本研究では，表 1 に示すような実装環境で実装を行なった．また，ネットワーク層のプロトコルとして IPv4 を利用しているので，送信先ホストが N/I の所属しているどちらのネットワークにも属していない場合，送信データ分割機能においてデフォルトゲートウェイを変更している．

3.2 パケットフォーマット

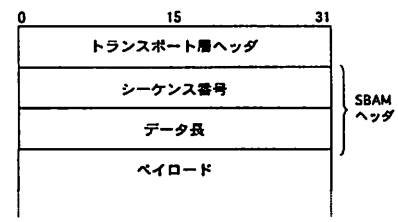

図 4: パケットフォーマット

SBAM では，図 4 ようなヘッダをパケットに付与する．シーケンス番号は，ソケット層でデータの順序を入れ換えるために存在する．データ長は，そのパケットが SBAM 対応のパケット

なのかを確認するためと，その場合，ペイロード部分がどれくらいの長さなのかを把握するために存在する．

3.3 送受信の流れ

本小節では，送信と受信の時の処理の流れを示す．図 5 に，送受信時にカーネル内でどのような関数が呼び出されるかを示す．

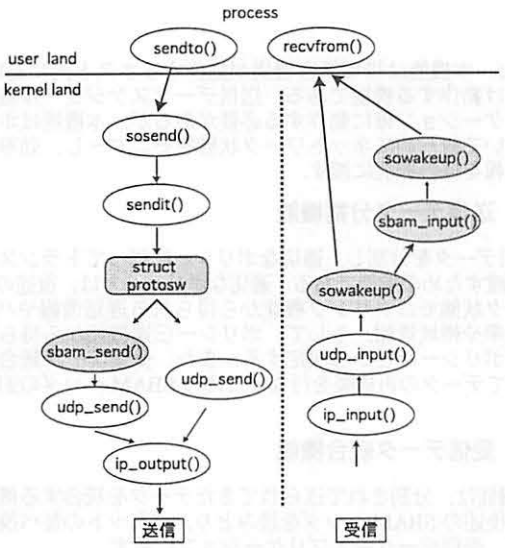

図 5: 送受信における関数の呼び出し

送信時

通常のデータ送信の場合，FreeBSD の実装では，sosend 関数内で利用するトランスポートプロトコルに応じた関数を protosw inet 構造体を利用し呼び出す．sosend は下位トランスポートプロトコルが必要とする形に従い，send システムコールや write システムコールの情報を渡す関数である．下位トランスポートプロトコルは TCP/IP プロトコルスタックの場合，protosw inet 構造体に関数ポインタの形で登録されている．SBAM では UDP プロトコルスタックを呼び出す前に，sbam_send という SBAM の処理を行なう関数を呼びだし，udp_send 関数を呼んでいる．sbam_send は 2.4 節で述べた送信データ分割機能と送信データスケジュール機能にあたる．

受信時

通常，受信したデータは，ip_input，udp_input 関数で処理されソケットバッファにためられる．そして sowakeup 関数により，プロセスはそのデータを読み出す．SBAM では，sowakeup 関数内より input 関数を呼び出し，SBAM ヘッダの処理を行なう．input 関数が 2.4 節で述べた受信データ統合機能にあたる．

4 評価

SBAM のプロトタイプ実装を，図 6 に示すネットワーク環境で評価した．実際のネットワーク構成は，図 6 の Network 1 と Network 2 を ルータに接続し，そこから Network 3 へ接続している．無線デバイスは 802.11b を 2 枚利用し，それぞれ 2 チャネルと 11 チャネルの帯域を利用した．また，今回はネットワーク状態モニタリング機能を実装していないので，コントロールパケットはネットワーク上に送信していない．トランスポートプロトコルは，UDP を利用した．

今回は，SBAM と評価用アプリケーション実装におけるスループットと，スケジューリング機能の有無によるスループット，そして受信ホストにおける到着パケットシーケンスを評価項目とした．

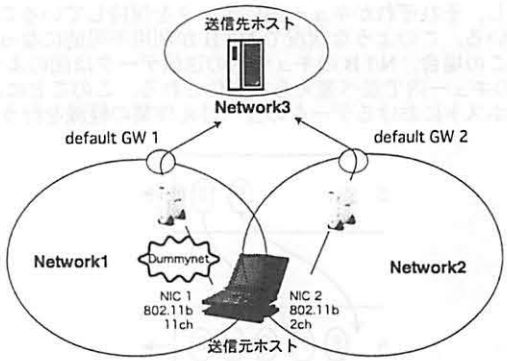

図 6: 実験ネットワークの構成

4.1 スループットの比較

SBAM と評価用アプリケーション

アプリケーション層における評価用実装 (APP1) と SBAM を用いてスループットの評価・比較を行なった．APP1 は 1 パケット毎に Network 1 と Network 2 に対し交互にデータを送信した．また比較のため，UDP データを送信するだけのアプリケーション (APP2) を用いて，N/I を 11Mbps モードで動作させた場合と 5.5Mbps モードで動作させた場合のスループットの計測も行なった．スループットは 5.5MB のメディアデータを 50 回送信し，その計測結果の平均をとった．結果を表 2 に示す．

表 2: SBAM と評価用アプリケーションのスループットの比較

N/I	実装	スループット
11Mbps + 11Mbps	SBAM	15.0 Mbps
11Mbps + 11Mbps	APP1	15.4 Mbps
11Mbps	APP2	9.2 Mbps
5.5Mbps	APP2	6.1 Mbps

11Mbps の N/I を 1 枚利用した場合と比べ，11Mbps の N/I を 2 枚利用した APP1 も SBAM も約 1.6 倍程度スループットが向上していることが分かる．APP1 と SBAM を比較すると，SBAM のスループットは APP1 に対して 97.4% となっていることが分かる．SBAM では APP1 と比べて，利用可能インタフェースや現在のネットワークトポロジのチェック，ヘッダの付与を行なっているので，APP1 よりも処理が多くなっている．このオーバヘッドの原因を調べるために，N/I を 2 枚利用した場合の sbam_send 関数と，N/I が 1 枚の場合の sbam_send 関数，そして，rtalloc 関数の実行時間を計測した．N/I が 1 枚の場合，sbam_send 関数はリンクアップしている N/I の数を調べるために線形探索を 1 度行ない，udp_send 関数を呼び出す．rtalloc 関数とは，ルーティングテーブルを検索するための関数であり，sbam_send 関数の中で 2 回呼び出されている．計測は 1400 バイトのパケットを 50 個送信し，1 パケット毎に sbam_send 関数と rtalloc 関数の実行時間を計測した．その平均結果を表 3 に示す．N/I が 2 枚の場合，sbam_send 関数は N/I が 1 枚の場合と比べ 47.75 倍のオーバヘッドが存在することが分かる．また，N/I が 2 枚の場合のオーバヘッドの内，30.2% が rtalloc 関数のオーバヘッドとなっている．よって，SBAM 機構に特化した rtalloc 関数の代替を考案することで，スループット改善の余地があると考えられる．

実装内容について比較すると，APP2 では raw socket の利用，指定されたデフォルトゲートウェイへの変更，UDP データグラムの送信をアプリケーションにおいて全て実装しなければならず，C 言語で実装したプログラムの行数は 666 行に達した．しかし SBAM では既存のソケットプログラミングによる UDP 送

表 3: 各関数の実行時間

N/I 2 枚時の sbam_send()	0.15μs
N/I 1 枚時の sbam_send()	1.06μs
rtalloc()	7.02μs

表 4: スケジューリングの有無によるスループットの比較

N/I	実装	スループット
11Mbps + 5.5Mbps	APP1	9.6 Mbps
11Mbps + 5.5Mbps	APP3	11.7 Mbps

信手法で同等の機能を実現でき，プログラムの行数は 206 行であった．このことから，アプリケーション層における実装と比べて同等の機能を実現するのが容易であることが分かる．

スケジューリング機能の有無

スケジューリング機能を持たない APP1 と，式 (1), (2) に基づいたスケジューリング機能を実装した評価用アプリケーション (APP3) を用いてスループットの比較を行なった．APP3 に必要な帯域と遅延情報はあらかじめ計測した値を利用し，MTU は 1500 バイトで統一した．また，N/I の重み α も 1 で統一した．結果を表 4 に示す．

各リンクの帯域が異なる場合，スケジューリング機能のない APP1 は，11Mbps と 5.5Mbps の N/I のスループットの和と比べて，62.7% のスループットとなっている．これに対し，帯域に比例したデータ送信を行なう APP3 では，同様に 76.5% のスループットとなっている．このことから，リンクの帯域に応じたデータ送信を行なうことで，各リンクの帯域を効率的に利用できることが分かった．

4.2 到着パケットシーケンス

ネットワーク環境の変化による到着パケットシーケンス番号のずれについての実験を行なった．表 5 に今回行なった実験環境を示す．NIC1, NIC2 はそれぞれ図 6 に対応している．

遅延は Dummynet [11] を Network 1 のリンク上に設置し，発生させた．各実験とも，5.5MB のメディアデータを各 N/I に対しラウンドロビンで送信した．UDP パケットには，送信元ホスト上で，それぞれの NIC において通し番号となるような 4 バイトのシーケンス番号を付与した．パケット到着時間を送信先ホスト上において，Pentium Counter を利用して計測した．それぞれの環境における実験結果を図 7 ～ 9 に示す．各図の縦軸は，送信元ホストで付与されたパケットシーケンス番号を表し，横軸は送信開始からの経過時間を示す．

図 7 より，2 枚の N/I に大きな帯域差がなく，遅延もないネットワークにおいて，データ送信後 1 秒以降，パケットシーケンス番号のずれが目立ちはじめていることが分かる．データ送信開始後 1 秒まではパケットシーケンス番号のずれはほとんど見られない．データ送信開始後 1.7 秒の時点で，100 パケット程度のずれが見られるようになる．しかし，遅延が 50ms ある環境で

表 5: 実験環境

	NIC1	NIC2
環境 1	11Mbps, 遅延 0ms.	11Mbps, 遅延 0ms
環境 2	11Mbps, 遅延 50ms	11Mbps, 遅延 50ms
環境 3	11Mbps, 遅延 0ms	5.5Mbps, 遅延 0ms

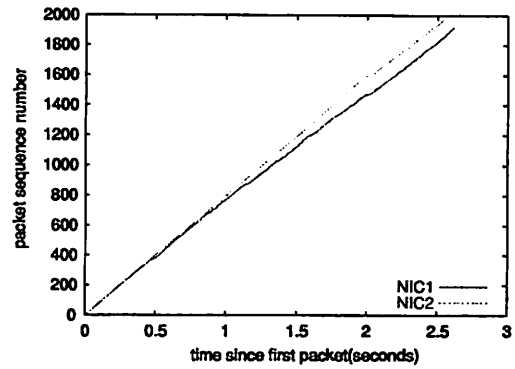

図 7: NIC1:11Mbps, 遅延 0ms. NIC2:11Mbps, 遅延 0ms

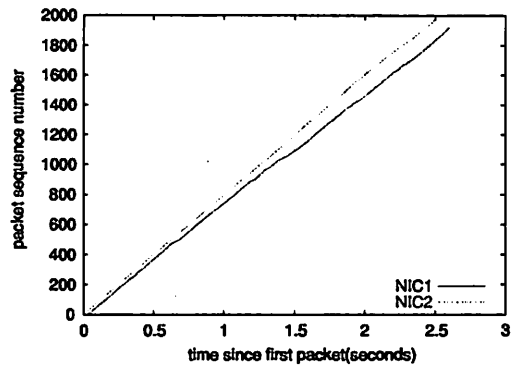

図 8: NIC1:11Mbps, 遅延 50ms. NIC2:11Mbps, 遅延 0ms

は，送信当初よりパケットシーケンス番号のずれが見られ，データ送信開始後 1.5 秒には 100 パケット程度のずれが生じていることが図 8 より分かる．また，NIC 1 の帯域を 11Mbps，NIC 2 の帯域を 5.5Mbps とした実験では，データ送信開始後 0.5 秒後には，100 パケット程度のずれが生じていることが図 9 より分かる．

これらの実験より，送信元/送信先ホスト間の帯域と遅延がパケットの到着順に影響を与えることが判明した．受信した通りのパケット順のままデータをアプリケーションに渡した場合，データを全く再構築できないという問題が発生する．例えば映像ストリーミングを行なっている場合，パケットロスが起きたら，そのパケットを含むフレームを破棄するだけで良いが，結果のように到着パケットシーケンスがずれた場合，フレームを全く構成できない．本稿の結果より，今回実装されていない送

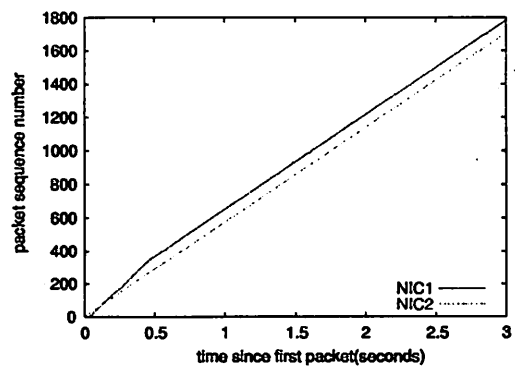

図 9: NIC1:11Mbps, 遅延 0ms. NIC2:5.5Mbps, 遅延 0ms

信データスケジュール機能では，送信元/送信先ホスト間の帯域・遅延を考慮したデータ送信スケジュールを行なわなければならないことが分かる．ソケット層においてパケット順を入れ換える受信データ統合機能のみでは，TCP のような遅延の値を利用するトランスポートプロトコルの性能に悪影響を与えてしまうことが想定される．また，帯域と遅延情報を取得するためのネットワークモニタリング機能の実装も必要であることが分かる．

5 関連研究

本節では，ネットワークスタックを改良した帯域統合機構の関連研究を挙げる．我々が調査した限りでは，アプリケーション層における帯域統合機構として，PSocket [3] や XFTP [10] など，高速ネットワーク上において，TCP のスループットを向上させるためのものは存在するが，SBAM のようにマルチホーム環境を想定した機構は存在しない．

5.1 トランスポート層

本稿で提案する SBAM と同様に，帯域統合を実現する機構として，Hung-Yung Hsieh らの PTCP [4] と RCP/R2CP [5] が挙げられる．TCP はデータ送信側で輻輳制御や再送制御などを行っているが，RCP では，それらの機能をデータ受信側で実現し，無線メディアに適応した信頼性のあるトランスポートプロトコルを提案している．また，マルチホーム環境の場合には，R2CP というモジュールが N/I 毎に RCP を用意し，効率的な帯域統合機構を実現している．

R2CP はトランスポート層で実現されているため実現が容易という利点があるが，ネットワークの自由度が低くなる欠点がある．例えば，RCP は TCP のような信頼性のあるトランスポートプロトコルであるが，ユーザが UDP のような信頼性のないプロトコルを利用したい場合，新たに RCP に相当するトランスポート層を用意しなければならない．また，トランスポート層において送信側と受信側が対応しなければならないので，通信が全く行えない可能性がある．SBAM では，ソケット層で実現することにより，トランスポート層に対して透過的な機構となっている．SBAM 機構を利用するかどうかについては，通信相手が対応しているかによって，選択可能になっているので，プロトコルが違うために通信が行なえないという問題は起きない．

5.2 リンク層

IEEE 802.3ad [8] において，リンク層における帯域統合機構が制定されている．これは，ホスト上に同じ N/I を複数装着し，スイッチとの間の帯域を向上させるための仕組みである．この方法は OS やアプリケーションなどに全く影響を与えることなく実現できる点において優れているが，利用できるネットワークは 1 つに限られている．つまり，そのネットワークのアップリンク以上の帯域は出ないことになる．また，802.3ad は有線のための規格なので，無線リンクには対応していない．

SBAM では，異なるネットワークに接続することを想定しているので，一つのネットワークのアップリンクに縛られることはない．また，N/I として無線を想定しているので移動可能である．

6 まとめ

本論文では ソケット層における帯域統合機構である SBAM を提案し，プロトタイプ実装を用いて評価を行った．SBAM はソケット層において，送信データスケジュール機能，ポリシー伝達機能，送信データ分割機能，ネットワーク状態モニタリング機能，受信データ統合機能を持ち，複数 N/I の帯域を利用することができる．本論文では SBAM の設計・実装を述べ，実際に SBAM を用いて複数 N/I を利用しデータ送信を行なった．

評価実験では SBAM を用い，802.11b の N/I を 2 枚利用し，UDP のスループットをアプリケーション実装と比較した．また，送信先ホストにおける到着パケットシーケンスについて評価を行なった．

ソケット層で実装されている SBAM とアプリケーション層での実装では，約 2.6% のスループットの差があることが分かった．しかし，ソケット層で実現しているので同等の機能を持つプログラムを容易に実装できることも分かった．到着パケットシーケンスに関して，11Mbps と 5.5Mbps のリンクよりデータを分割して送信した場合，送信開始後 1.5s の時点で 8% のずれが生じることと，相手先ホストまでの各 N/I 間の遅延の差が 50ms の場合，送信開始後 1.5s の時点で約 8% ずれることを観測した．

今後の課題として，以下が挙げられる．

- 各機能の実装
  未実装である，ネットワーク状態モニタリング機能，送信データスケジュール機能，ポリシー伝達機能を実装する必要がある．
- アドレス解決
  今回のプロトタイプ実装ではアドレスについての考慮をしていないが，送信先ホストにおいて受信データが同一アドレスから到着しているように見える機構の導入を行なう．
- 他のトランスポートプロトコルでの評価
  今回のプロトタイプ実装ではネットワーク状態の取得，アドレス解決機構が導入されていないので，UDP を用いて実験を行なった．しかし，TCP のような信頼性のあるプロトコルでの評価を行ない，再送機構，輻輳回避機構と SBAM がどのような相関を示すかを評価する必要がある．
- 3 枚以上の N/I を利用した評価
  SBAM の N/I の枚数に対するスケーラビリティを評価し，アプリケーション層に対する優位性を示す必要がある．

Acknowledgement

この研究は総務省「ユビキタスネットワーク制御・管理技術の研究開発 (ubila プロジェクト)」の一部として行なわれました．

参考文献

[1] Bluetooth. http://www.bluetooth.com/.

[2] K. McCloghrie etc. Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991.

[3] S. Bailey H. Sivakumar and R. Grossman. Psockets: The case for application-level network striping for data intensive applications using high speed wide area networks. In *IEEE Supercomputing (SC)*, November 2000.

[4] Hung-Yun Hsieh and Raghupathy Sivakumar. A Transport Layer Approach for Achiving Aggregate Bandwidths on Multi-homed Mobile Hosts. In *Proceedings of ACM MOBICOM'02*, 2002.

[5] Yujie Zhu Hung-Yun Hsieh, Kyu-Han Kim and Raghupathy Sivakumar. A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces. In *Proceedings of ACM MOBICOM'03*, 2003.

[6] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz band*, February 2000.

[7] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher speed Physical Layer (PHY) extension in the 2.4 Ghz band*, February 2000.

[8] IEEE 802.3ad Standard (IEEE Computer Society LAN MAN Standards Committee). *Aggregation of Multiple Link Segments*, 2000.

[9] Kevin Lai and Mary Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, March 2001.

[10] H. Kruse M. Allman and S. Ostermann. An application-level solution to tcp's satellite inefficiencies. In *Workshop on Satellite-Based Information Services(WOSBIS)*, November 1996.

[11] Luigi Rizzo. Dummynet: a simple approach to the evaluation of network protocols. In *ACM Computer Communication Review*, 1997.

[12] R. H. Hatz T. R. Henderson. *Transport Protocols for Internet-Compatible Satellite Networks*, Vol. 72 of *IEEE Journal*. Feb 1999.

[13] Ultra Wideband (UWB). http://www.uwb.org/.