モーバイルコンピューティング　9－2
（1999．5．28）

最低帯域保証型サービスのための無線スケジューラの実装

鈴木啓介† 吉村健†
森川博之† 青山友紀†

近年、無線ネットワークにおいてもQoSの保証が必要となりつつある。しかしながら無線ネットワークは通信品質が低くかつ経路が常に変化する。そこで無線環境に適したサービスとして最低帯域保証型サービスを示す。最低帯域保証型サービスでは必要最低限の帯域のみを保証するため、混雑時においても最低限の通信品質を保証し帯域に余裕があれば残余帯域を活用してより高品位な通信を提供できる。本稿では、最低帯域保証型サービス実現に必要なスケジューラのスケジューリング手法を検討し、LinuxとWaveLANからなるシステムへの実装を試みる。

Implementation of Wireless Scheduler for Minimum Bandwidth Guaranteed Service

KEISUKE SUZUKI,† TAKESHI YOSHIMURA,† HIROYUKI MORIKAWA† and TOMONORI AOYAMA†

Wireless networks are expected to "guarantee" quality of service. Wireless links, however, have poor quality and should support host mobility. We introduce the concept of a minimum bandwidth guaranteed service as a mobile service. Our service model guarantees only minimum bandwidth with which minimum tolerable quality can be obtained, and shares the leftover bandwidth fairly among all flows to enhance their quality of service. In this paper, we present a packet scheduling method to provide the service and implement it over Linux and WaveLAN.

1. はじめに

昨今のインターネットの爆発的普及、ネットワークの高速化、マルチメディア関連技術の進歩により、リアルタイム性を必要としかつ大容量なマルチメディア通信への要求が高まりつつある。通信のサービス品質（Quality of Service: QoS）を保証するため、IETF（Internet Engineering Task Force）においてguaranteed service[1]やcontrolled-load service[2]、さらにはdifferentiated service[3),4)]といった種々のサービスが提案されている。

ネットワークのラストワンホップとなる無線LANにおいても種々のサービスクラスをサポートする必要がある。しかし、無線ネットワークには狭い帯域幅、バーストエラー、ハンドオフによる通信経路変化などの特性がありQoSを維持するのが難しい。そのため前述のような有線ネットワークにおけるサービスをそのまま無線ネットワークに適用するのは困難である。

このような視点から、無線ネットワークの性質を考慮しつつ種々のサービスクラスを提供できるサービスである最低帯域保証型サービスの研究を進めている[5]。最低帯域保証型サービスは、アプリケーションが要求する全帯域を保証するのではなく、アプリケーションが動作するのに最低限必要な帯域のみを保証する。そして、残余帯域を全フローで共有することにより帯域保証と同時に無線資源の利用効率向上を実現する。

最低帯域保証型サービスでは最低限必要な帯域のみを保証するため、広範囲な伝送速度を許容する適応型アプリケーションが適している[6]。最低帯域保証型サービスに適応型アプリケーションを組合わせることにより、帯域に余裕がなくても最低限のQoSを保証し、残余帯域があればより高品位な通信を提供できる。

本稿では、最低帯域保証型サービスを実現する最低帯域保証型スケジューラのスケジューリング手法を示し、その実装法を検討する。そして、Linux搭載PCおよびWaveLANからなる無線LANシステムへの実装を試みる。さらに実装システムに対する評価実験を

† 東京大学工学系研究科
School of Engineering, University of Tokyo

通じて最低帯域保証型サービスが有効であるかどうかを検証する。

2. 無線スケジューリング手法

さまざまなサービスクラスをサポートするにはスループットや最大遅延、通信の継続性に関するの保証が必要である。

有線ネットワークにおいてはエラーがほとんど無くまた帯域幅にも余裕があるため、最大遅延を完全に保証する Guaranteed Service などによってこれらの条件を満たすことができる。しかし、無線ネットワークには狭い帯域幅、バーストエラー、ハンドオフによる通信経路変化などの特性があり QoS を維持するのが難しい。仮に最大遅延を完全に保証しようと試みると、通信状況が最悪になる場合を考慮し必要以上の帯域を割り当てなければならない（図1左）。この帯域の浪費により、帯域幅の狭い無線環境においては他のフローの呼を受け付けることができず呼損率増加を招く。また、最大遅延の完全遅延を要求するフローはハンドオフの際の移動先セルにおいて帯域確保失敗による強制切断率が高くなり通信の継続性が失われてしまう。このような観点から、無線ネットワークの特性を考慮したサービスとして最低保証帯域保証型サービスについて検討を進めている。

2.1 最低帯域保証型サービス

無線ネットワークにおいては QoS を完全に保証することは困難であるとともに、QoS を完全に保証しようと試みることで呼損率や強制切断率が増加してしまう。したがって、無線ネットワークにおいては QoS を緩く保証するサービスが適当であると考える。アプリケーションの要求する全帯域を保証するのではなく、最低限必要とする帯域のみを保証し残余帯域をすべてのフローで共有するものである（図1右）。最低限の帯域のみを保証することにより、混雑したネットワークにおいても各フローの最低帯域は保証されるとともに、帯域に余裕のあるネットワークでは各フローにより多くの帯域が割当てられ高品質の通信が可能となる。また、新たに発呼したフローやハンドオフにより移動してきたフローに対しては、最低保証帯域を確保できる場合に呼を受け付ける。したがって、要求全帯域が確保されないと呼を受け付けない場合に比べて呼損率やハンドオフ時の強制切断率が軽減され通信の継続性が確保される。

最低帯域保証型サービスでは更に次の2つの事項を規定している。第一点はチャネル状態に応じたスケジューリング手法である。これは、バーストエラーに陥っているフローに対しては無線資源を割り当てないことでチャネル利用効率をさらに向上させるものである。第二点は基地局によるチャネルアクセス制御である。基地局が適切なスケジューリングを行っていても、アップリンクによって基地局の送信が妨げられて最低保証帯域が満たせない場合が考えられる。アップリンクに対してもチャネルアクセス制御を行うことで確実に最低保証帯域を満たすことができる。以上の事項は IEEE802.11 上での実装方法について検討を行っている。

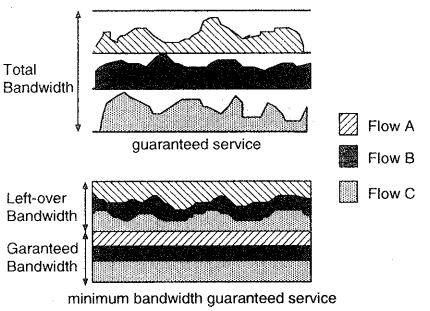

図1 Guaranteed service と最低帯域保証型サービス
Fig. 1 Guaranteed service and Minimum bandwidth guaranteed service

適応型アプリケーション

無線ネットワークでは帯域幅が狭いことに加えバーストエラー等により使用可能な帯域幅が時間的空間的に変化するので、広範囲な伝送速度を許容する適応型アプリケーションが適している。非適応型アプリケーションでは要求帯域 b 未満では全く動作できないのに対し（図2 a）、適応型アプリケーションは最低帯域幅 b_{min} 以上であれば品質を変化させながら動作することができ、全要求帯域幅 b_{max} 以上の帯域幅がある場合、最高品質で動作できる（図2 b, c, d）。すなわち、最低帯域保証型サービスで b_{min} を保証すればアプリケーションは動作し続けることができ、通信の継続性が確保される。さらに最低帯域保証型サービスによって残余帯域が割当てられればより高品質で動作すること可能である。適応度によって4種類の適応型アプリケーションが考えられる。（図2 a, b, c, d）

以上から、最低帯域保証型サービスと適応型アプリケーションを組み合わせることによって、チャネルの有効利用やハンドオフ時の強制切断率の低減を維持しながら最低品質のアプリケーション動作保証、残余帯域を利用したより高品質のアプリケーション動作を実現できる。

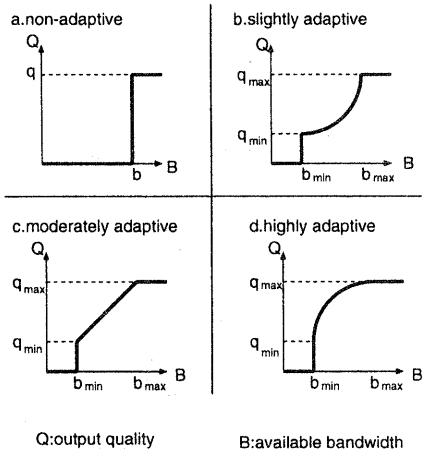

図2　適応型アプリケーション
Fig. 2　Adaptive Application

3. 最低帯域保証型スケジューラ

前節で述べた最低帯域保証型サービスを実現するためには、無線資源を各フローに割り当てる最低帯域保証型スケジューラが必要である。このスケジューラに備わっていなければならない機能は、最低帯域保証と残余帯域の共有である。それぞれ guaranteed scheduler と sharing scheduler がその役割を果たす。

最低帯域保証型スケジューラのスケジューリング手法の概略を図 3に示す。まず、guaranteed scheduler が最低保証帯域が満たされているかどうかチェックする。もし最低保証帯域の満たされていないフローを発見したら、そのフローのパケットを送信する。すべてのフローの最低保証帯域が満たされている場合は、sharing scheduler に制御が移る。sharing scheduler は残余帯域を平等に分配するようフローを選択してパケット送信を行う。

このように、guaranteed scheduler を先に動作させることにより最低帯域を保証し、guaranteed scheduler によるスケジューリングが行われなかったときのみ sharing scheduler が残余帯域のスケジューリングを行う。

guaranteed scheduler

guaranteed scheduler はパケット転送毎に現在時刻を表すように更新されるパラメータ *current_time* を保持する。また、各フロー毎に *start_time* というパラメータを有している。$start\_time_i$ はフロー i が最低保証レート $b_{min\ i}$ を確保するために次のパケットを送信すべき時刻を表す。したがって、$start\_time_i$ が現

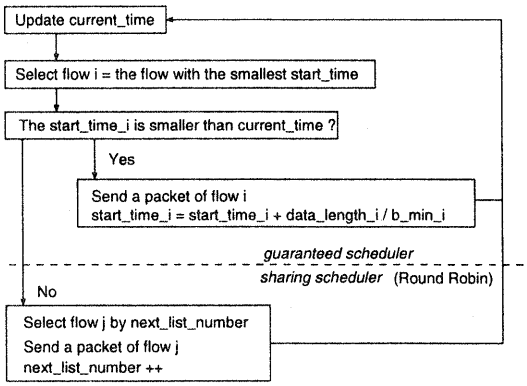

図3　スケジューリング手法
Fig. 3　Scheduling method

在時刻 *current_time* より小さいとフロー i は最低帯域幅を満たしていない状態であり、逆に $start\_time_i$ が *current_time* より大きいとフロー i は最低保証帯域を満たしていることになる。

guaranteed scheduler は、最初に *start_time* が一番小さいフロー i を検索する。選択されたフロー i は最低保証帯域が満たされていない可能性が最も高いフローである。次にその $start\_time_i$ と *current_time* を比較する。

$start\_time_i$ の方が大きい場合　フロー i は最低保証帯域を満たしている。すなわち、全フローが最低保証帯域を満たしているので、sharing scheduler に制御が移行する。

current_time の方が大きい場合　フロー i は最低保証帯域を満たしていない。したがってフロー i のパケットを送信する。そして、$start\_time_i$ を

$$start\_time_i \leftarrow start\_time_i + data\_length_i / b_{min,i}$$

により更新する。$data\_length_i$ は送信したパケットのデータ長、$b_{min,i}$ はフロー i の最低保証帯域である。

このように、全フローの *start_time* が *current_time* より大きくなるまで guaranteed scheduler のみがスケジューリングを行なうことによって最低帯域保証を実現する。

sharing scheduler

sharing scheduler は guaranteed scheduler によるスケジューリングが行われなかった場合、すなわち全フローが最低保証帯域を満たしている場合にスケジューリングを行う。本実装では全フローが残余帯域を等しく共有するように、Round Robin scheduler を

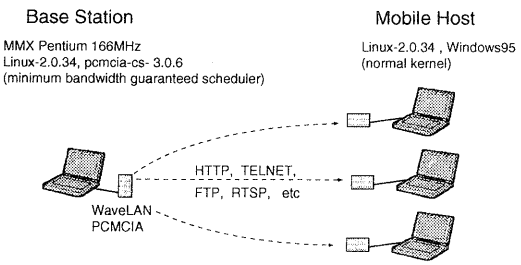

図4　実装環境
Fig. 4　Implementation platform

用いる。

スケジューラは *next_list_number* というパラメータを保持する。これは Round Robin において次に送信されるべきフロー番号である。スケジューラはこの値にしたがって送信フローを選択し、そして次のフローを送信するため *next_list_number* を 1 増やす（*next_list_number* がフロー番号の最大値になると *next_list_number* を 0 に初期化する）。

以上のように guaranteed scheduler と sharing scheduler の独立な 2 つのスケジューラから構成される最低帯域保証型スケジューラによって最低帯域保証型サービスを提供する。

4. 無線 LAN システムへの最低帯域保証型スケジューラの実装

4.1 実装システム

実装システムの構成を図 4に示す。スケジューラを実装する基地局用 PC(MMX Pentium 166MHz) に Linux2.0.34[7] (Slackware 3.4 適用後カーネルのみ 2.0.34 へアップデート) をインストールした。Linux を使用するのは、ソースコードが公開されており OS のネットワーク関連コード [8] やデバイスドライバ [9] の改造が可能なためである [10]。また、無線 LAN は WaveLAN/PCMCIA（チャネルレート 2Mbps）を使用する。WaveLAN は現在広く普及している無線 LAN であり、Linux 用のドライバ (pcmcia-cs-3.0.6[11]) が配布されている。モバイルホスト用 PC には各々評価実験に応じて Linux と Windows95 の両方を使用した。モバイルホストに対してはスケジューラの実装を行なわない。

4.2 アルゴリズム

実装にあたっては第 2 節で述べたスケジューリング手法を Linux のネットワークコードに適合させなければならない。本実装で実際に使用したアルゴリズムを図 5に示す。主な改良点は以下の通りである。

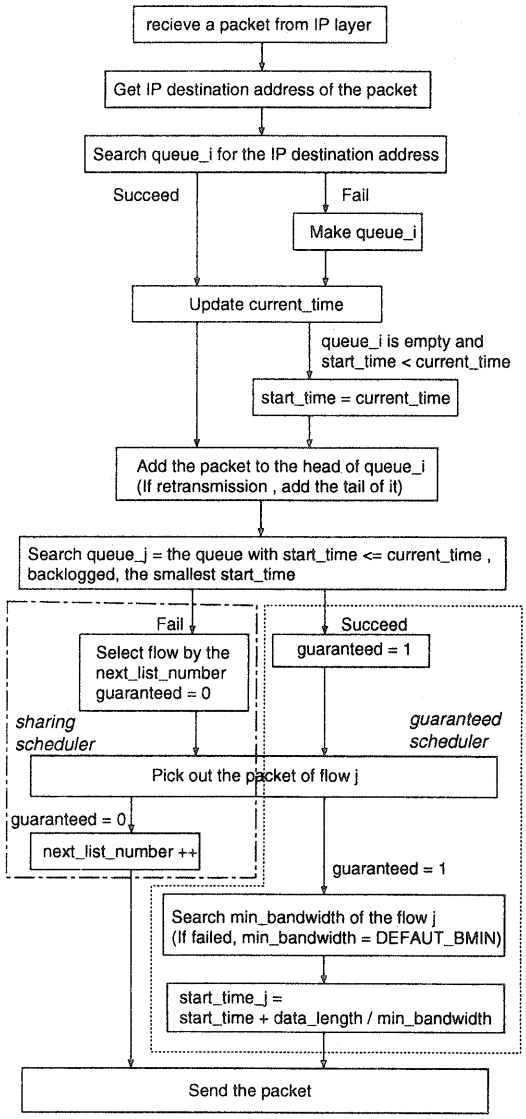

図5　アルゴリズム
Fig. 5　Scheduling algorithm

- フローの識別は IP アドレスによって行う。
- 1 つのフローにつき、前節で述べたキューを 1 つ使用する（IP アドレスとキューが 1 対 1 に対応する）。
- パケット到着時に IP アドレスによってそのパケットが入るべきキューを探索する。見つからない場合はリストを新規生成する。
- 誤って空のパケットを送信することのないよう、スケジューリング時にキューが空でないか確認する。

- コードの変更が極力少なくすむようキューへの付加、キューからの取りだし、パケットの送信はguaranteed scheduler と sharing scheduler で同一のコードを使用する。そのため、*guaranteed* というフラグを利用して guaranteed scheduler と sharing scheduler を区別する。
- 空のキューにパケットが入りかつ $start\_time < current\_time$ である場合、$start\_time = current\_time$ と $start\_time$ を初期化する。
- 最低保証帯域を設定していないフローに関しても同様に guaranteed scheduler によるスケジューリングが行われる。この場合最低保証帯域幅は DEFAULT_BMIN という既定値を用いる。この DEFAULT_BMIN の値を非常に小さくすることにより無保証フローとなる。

このアルゴリズムに作成可能キュー数に対するオーバーフロー処理を加えてプログラミングを行なった。プログラム作成後、カーネルを再コンパイルすることにより最低帯域保証型スケジューラの組み込まれたカーネルを作成した。実装において手を加えたのはカーネルのみで、他のプログラム等は一切変更していない。このカーネルで Linux を起動することにより常に最低帯域保証型スケジューラが動作する。

5. 評価実験

前章で実装した最低帯域保証型スケジューラの動作を検証するため、2 種類の評価実験を試みた。

5.1 擬似ハンドオフ

ハンドオフや新規フロー発呼時には各フローに割り当てられる帯域が変化する。このような場合においても最低保証帯域が満たされていることを確認するため擬似ハンドオフ実験を行った。

スケジューラを搭載した基地局からモバイルホスト 3 台 (Linux) へ UDP データグラム (1024byte) を送信する。送受信には Linux 上でクライアント / サーバ型 UDP 送受信プログラムを用いた。このプログラムでは送信側に常に送信待ちデータグラムが存在し (バックログ状態)、一つのデータグラムの送信が完了するとすぐに次のデータグラムの送信動作が行われる。また、受信側ではパケットを受信するとデータ長および受信時刻を記録する。

擬似的にハンドオフを実現させるため、各 UDP 送信フローを実験途中に生成消滅させた。各フローの存在時間および最低保証帯域幅を表 1に示す。モバイルホスト側では受信時刻と累計受信データ長を記録し、スループットの時間変化を算出した。

flow	alive time (sec)	minimum guaranteed bandwidth (kbps)
flow1	0 ~ 30	600
flow2	10 ~ 40	400
flow3	20 ~ 50	0

表 1　各フローのパラメータ
Table 1　Parameter of each flow

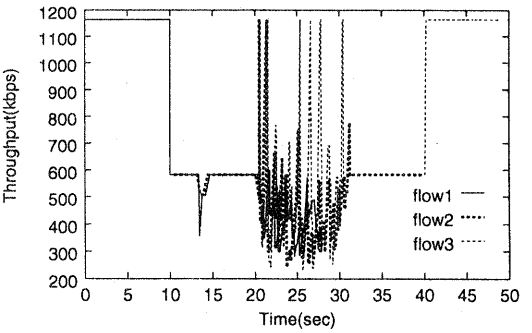

図 6　スループット (スケジューラ無)
Fig. 6　Throughput (without scheduler)

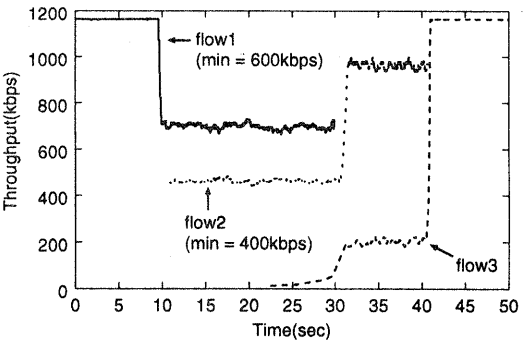

図 7　スループット (スケジューラ有)
Fig. 7　Throughput (with scheduler)

スケジューラ無 (標準のカーネル:FIFO) の場合の結果を図 6に、スケジューラ有の場合の結果を図 7 に示す。

スケジューラ無の場合は、フローが生成すると各フロー間で帯域幅が等分配されていることがわかる。すなわち、flow1,2 が共存している場合 (10 ~ 20sec) 及び flow2,3 が共存している場合 (30 ~ 40sec) は 600kbps 弱になっている。flow1,2,3 が共存している場合 (20 ~ 30sec) は期間平均のスループットは 400kbps 弱に落ち着いている。しかし、グラフでは瞬間的にスループットが大きく変動している。この瞬間的なスループット変動に関しては次節で詳しく検証する。

スケジューラ有の場合は、新たなフローが生起して

flow	alive time (sec)	minimum guaranteed bandwidth (kbps)
flow1	0 ~ 30	300
flow2	10 ~ 40	300
flow3	20 ~ 50	300

表2　各フローのパラメータ
Table 2　Parameter of each flow

も各フローの最低保証帯域が満たされていることがわかる。すなわちflow1は常に最低保証帯域600kbps以上であり、flow2は常に最低保証帯域400kbps以上である。

以上のことから、最低帯域保証型スケジューラによってハンドオフ時や呼受付時においても最低帯域が保証されることを示した。

瞬間的なスループット変動

図6で見られる瞬間的なスループット変動（20 ~ 30sec）について更に検証する。スループット変動のみに着目して表2の条件で再度擬似ハンドオフ実験を行った。

表2の条件ではすべてのフローに対して同じ最低保証帯域を設定した。したがって、スケジューラの有無に関わらず帯域幅は各フローで等分割されるはずである。

スループットを算出する際には

$$throughput = \Delta data\_length / \Delta time$$

という計算式を用いる。スループット変動を比較するためにそれぞれ $\Delta time = 100msec$, $200msec$、さらにスケジューラ無の場合は $\Delta time = 800msec$ としてスループットを算出した。$\Delta time$ が大きいほどより平均的なスループットを示している。そしてフローが1本、2本、3本それぞれ存在している場合の瞬間的なスループット変動がわかるよう、flow3（約20 ~ 50sec）のみに注目してスループット変化を算出した。スケジューラ無の場合の結果を図9に、スケジューラ有の場合の結果を図9に示す。

スケジューラ無の場合、flow1, 2, 3が共存する期間（20 ~ 30sec）においてスループットの値が非常に変動していることがわかる。そして、$\Delta time = 100msec$ から $\Delta time = 200msec$, $800msec$ となるにつれて、スループットの変動が小さくなり平均スループットの400kbps弱に近付いていることがわかる。したがって、スループットは瞬間的に不規則に変動しているがより長時間の視点で見ると安定しているといえる。

スケジューラ有の場合、flow1, 2, 3が共存する期間（20 ~ 30sec）においてもスループットの値は400kbps弱で安定している。そして $\Delta time = 200msec$ の方がスループットの値がより安定するのはスケジューラ無の場合と同じである。

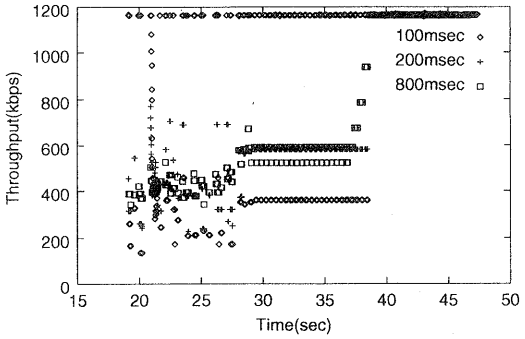

図8　flow3のスループット（スケジューラ無）
Fig. 8　Throughput of flow3 (without scheduler)

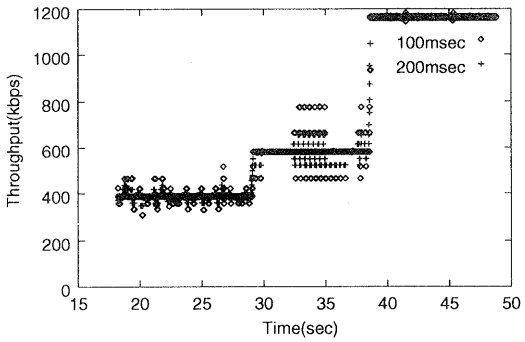

図9　flow3のスループット（スケジューラ有）
Fig. 9　Throughput of flow3 (with scheduler)

このスループットの瞬間的変動については次の原因が考えられる。複数のフローが存在する場合にはネットワークに対するアクセスが同時並行的に起こる。その場合、TCP層やIP層では各フローのパケットが混合された状態でFCFS (First Come First Service) で処理される。このとき、特定のフローのパケットが片寄って存在するとスループットの瞬間的な変動が起こると考えられる。しかし、最低帯域保証型スケジューラが存在する場合、スケジューラ内においてパケットが各フロー毎独立に管理されるのでスループットの変動が抑制される。

また、スケジューラの有無に関わらずflow2, 3が共存する期間（30 ~ 40sec）において周期的なスループットの変動がみられるのは、受信パケットのデータ量が一定で、かつパケットの送出にかかる時間がほぼ一定なためスループットを算出する際にどうしても離散的になるからであり、瞬間的なスループット変動とは根本的に異なる。

以上のことから、スケジューラが無い場合は平均的にはスループットが安定するもののフロー毎にパケットを管理できないのでスループットが瞬間的に不規則変動するといえる。これに対して、最低帯域保証型スケジューラの guaranteed scheduler および Round Robin scheduler によって各フロー毎にパケットを管理することによってスループットの瞬間的変動を抑制することが可能であるといえる。

このスループットの瞬間的な変動を抑制することは、遅延保証を必要とするアプリケーションにおいて効果があると考えられる。詳しくは次節にて述べる。

5.2 ストリーミング配信

最低帯域保証型スケジューラによる最低帯域保証サービスが実際のアプリケーションに対して有効であるか検証するため、ストリーミング配信の実験を行った。

実験ではスケジューラを実装した基地局、ネットワークに負荷をかけるために用いるモバイルホスト1台（Linux）、そしてストリーミングのクライアント1台（Windows 95）の3台を使用した。ストリーミング配信ソフトウエアは

- サーバー Real Server Basic G2 (for Linux)
- エンコーダー Real Producer G2 (for Win)
- クライアント Real Player G2 (for Win)

を使用した。トランスポートプロトコルは UDP を使用し、ビデオエンコードの際の Target Audience は Corporate LAN とした。また、ストリーミング制御プロトコルとして RTSP[12] を使用した。配信したストリーミングビデオは 30 秒間でファイルサイズは 1.8MB である。ビデオ再生時には使用帯域幅が 150kbps である。

実験方法は以下の通りである。前節の擬似ハンドオフ実験で使用した UDP 送信プログラムによって基地局からモバイルホストへ UDP データグラムを送信しておく。そして、クライアントが基地局の Real Server に対し配信要求を出すことによりストリーミング配信が行われる。本実験では最低帯域保証型スケジューラの有無、最低保証帯域幅の大小によりパケットロス率がどう変化するのか調べた。パケットロス率は Real Player G2 に表示されるデータを使用した。試行 10 回の平均結果を表 3に示す。

ネットワーク無負荷時が最もパケットロス率が少ないものの、最低帯域保証型スケジューラで最低帯域保証することによりパケットロス率が大幅に低減されていることがわかる。スケジューラ無の場合（パケットロス率 28%）では画面の大部分が崩れてしまうが（図 10上）、スケジューラで使用帯域幅 150kbps を保証した場合（パケットロス率 4.5%）では画面のごく一部が崩れる程度で視聴に支障がない（図 10下）。

UDP load	minimum guaranteed bandwidth (kbps)	average of packet loss ratio (%)
無	0 (FIFO)	0.3
有	0 (FIFO)	28.0
有	0	7.4
有	50	6.8
有	100	5.6
有	150	4.5
有	300	1.1

表3 ストリーミングにおけるパケットロス
Table 3 Packet loss ratio on the streaming video

本実験ではフローが2つしかなく帯域にも余裕があるので、スケジューラの有無に関わらストリーミング帯域 150kbps が十分確保され、パケットロス率は変わらないのではないかと考えられる。しかしながら、最低帯域保証型スケジューラが無い場合、ストリーミング配信によるフローが生起すると瞬間的なスループットが大きく変動する。遅延保証を必要とするストリーミング配信ではこの変動によってパケットが大量に落ちてしまう。一方、最低帯域保証型スケジューラが有る場合はストリーミング配信によるフローが生起しても各フローを独立に管理しているためスループットが安定しパケットがあまり落ちない。ゆえに、パケットロス率の低減の主要因はフロー別の管理であるので、スケジューラの有無によるパケットロス率低減効果は最低保証帯域幅の増加によるパケットロス率低減効果より顕著である。

以上のことから、最低帯域保証型スケジューラによって提供される最低帯域保証型サービスは混雑時においても確実に最低帯域を保証し、さらにフローが複数存在する場合のスループットの瞬間的変動を抑えることがわかる。したがって、スループットや遅延等の QoS 保証が必要なアプリケーションには最低帯域保証型サービスが効果的であるといえる。

6. おわりに

本稿では、無線環境で QoS を緩く保証する最低帯域保証型サービスのための最低帯域保証型無線スケジューラの実装方法について検討し、Linux と WaveLAN からなる無線 LAN システムへの実装を行なった。そして、UDP フローによる擬似ハンドオフ実験においてハンドオフや呼受付時に最低帯域が保証されることを通じ、実装したスケジューラが正しく動作することを確認した。さらに、スケジューラが瞬間的な

図10　ストリーミング画像　上：スケジューラ無　下：スケジューラ有
Fig. 10　A Image on the streaming video (upper:without acheduler lower: with scheduler)

スループットの変動を抑えることも示した。最後にマルチメディアアプリケーションでの評価により最低帯域保証型サービスはマルチメディア通信に有効であることを示した。

今後の課題としては、

- チャネル状態に応じたスケジューリング
- ダウンリンク及びアップリンク双方の基地局によるチャネルアクセス制御

が挙げられる。チャネル状態に応じたスケジューリングを導入すると、バーストエラーに陥っているフローに対してチャネルを割り当てないことにより帯域浪費を減らすことができる。また、現時点での最低帯域保証では基地局のチャネルアクセスがアップリンクから妨害されることにより最低帯域を保証できない危険性がある。最低保証帯域を確実に保証するためにはアップリンクに対するチャネルアクセス制御が必要である。

現在、以上についてIEEE802.11上での実装方法について研究を進めている。今後、IEEE802.11 MAC層のパラメータを制御可能なデバイスを用いてこれらの実装および評価を試みたい。

参考文献

1) S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. *IETF RFC 2212*, 1997.
2) J.Wroclawski. Specification of the Controlled-Load Network Element service. *IETF RFC 2211*, 1998.
3) Y. Bernet et al. A Framework for Differentiated Services. *Internet draft <draft-ietf-diffserv-framework-00.txt>*,, 1998.
4) K. Nichols and Steven Blake. Differentiated Services Operational Model and Definitions. *Internet draft <draft-nichols-dsopdef-00.txt>*, 1998.
5) T.Yoshimura, M.R.Jeong, H.Morikawa, and T.Aoyama. Wireless Packet Scheduling for Adaptive Service over IEEE 802.11. In *Proc. of Wireless Personal Multimedia Communications'98*, 1998.
6) Kam Lee. Adaptive Network Support for Mobile Multimedia. In *Proc. of MobiCom'95*, 1998.
7) Linux 2.0.34. ftp://ftp.kernel.org, 1998.
8) M.Besk, H.Bohme, M.Dziadzka, U.Kunitz, R.Magnus, and D.Verworner. *A Linux Kernal Internals, second edition.* Addison Wesley, 1997.
9) Alessandro Rubini. *Linux Device Driver.* O'Reilly & Associates,Inc., 1998.
10) Kenjiro Cho. A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers. In *Proc. of .USENIX*, 1998.
11) Linux pcmcia-cs-3.0.6. ftp://csb.stanford.edu/pub/pcmcia, 1998.
12) H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). *IETF RFC 2326*, 1998.