

モバイルストリーミング QoS サーバにおけるファイル切り替え方式

萩野 浩明[†] 尾上 裕子[†] 安木 成比古[†] 渥美幸雄[†]
駒木 寛隆^{††} 村尾高秋^{††} 串田高幸^{††} 山内長承[‡]

[†]株式会社 NTT ドコモ マルチメディア研究所
〒239-8536 神奈川県横須賀市光の丘 3-5

^{††}日本アイ・ビー・エム株式会社 東京基礎研究所
〒242-8502 神奈川県大和市下鶴間 1623-14

[‡]東邦大学理学部情報科学科
〒274-8510 千葉県船橋市三山 2-2-1

E-mail: [†] {hagino, yuko, yasuki, atsumi}@mml.yrp.nttdocomo.co.jp,
^{††} {HKOMAKI, MURAO}@jp.ibm.com, KUSHIDA@trl.ibm.co.jp
[‡] yamanouc@HyperResearch.com

あらまし 本研究では、これまでに実装しているモバイルマルチメディア QoS 制御システムにおけるサーバアルゴリズムとして、ファイル切替方式を提案、実装する。ファイル切替方式では、サーバは異なるビットレートでエンコードされたコンテンツファイルを保持しており、それらのファイルを切り替えることで送出量を制御する。また、モバイルマルチメディア QoS 制御システムでこれまでに用いていたオンザフライトランスコーディング方式との定量比較を行い、それぞれの特徴を明らかにした上で、より優れた制御方式について考察を行う。
キーワード モバイル、マルチメディア、QoS、IMT-2000

File Change Method for Mobile Streaming QoS Server

Hiroaki HAGINO[†], Yuko ONOE[†], Narihiko YASUKI[†], Yukio ATSUMI[†],
Hirotaka KOMAKI^{††}, Takaaki MURAO^{††}, Takayuki KUSHIDA^{††}
and Nagatsugu YAMANOUCHI[‡]

[†]Multimedia Laboratories, NTT DoCoMo
3-5, Hikarinooka, Yokosuka, Kanagawa, 239-8536, Japan
^{††}IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato, Kanagawa, 242-8502, Japan

[‡]Dept. Information Science, Toho University
2-2-1, miyama, funabashi, chiba, 274-8510, Japan
E-mail: [†] {hagino, yuko, yasuki, atsumi}@mml.yrp.nttdocomo.co.jp,
^{††} {HKOMAKI, MURAO}@jp.ibm.com, kushida@trl.ibm.co.jp
[‡] yamanouc@HyperResearch.com

Abstract In this paper, we propose the file changing method as a new control algorithm for the mobile streaming QoS system which we have already implemented. In the method, a server have several multimedia files with several bit rates, and it changes files to control QoS. We compare the file changing method with another method used in our system by experiments. Moreover, we consider the two methods and a new better method for our system.

Key words Mobile, Multimedia, QoS, IMT-2000

1. はじめに

近年、無線通信技術の発展に伴い、無線通信路の高速化が進んでいる。特に、IMT-2000 では当初 384Kbps, 将来的には 2Mbps での通信が可能である [3]。このように、無線通信基盤が整備されることによって、ユーザは、これまで広帯域の有線通信環境を前提としていたサービスを無線通信環境で利用することが可能となる [1]。

筆者らの研究グループでは、そのようなサービスの一つとして、マルチメディアストリーミングサービスを対象とし、実現のための問題抽出およびその解決のための方式提案と実装を目的に研究を進めている。ここで、この研究の主目的は、有線に比べ通信状態の変動が著しく頻繁で大きい無線通信環境において、クライアントがサーバから送信されるマルチメディアストリーミングデータを受信しながら、出来る限り途切れることなく再生できるような QoS 制御を End-to-End の通信において実現することである。また、無線通信路の断線等によりストリーミング再生がどうしても途切れてしまう場合の解決策の実現も重要な課題である。

これまでに筆者らの研究グループでは、これらの主目的の実現に必要な QoS 制御のためのフレームワーク、サーバにおける QoS 制御アルゴリズム、断線時の制御のためのセッション制御プロトコルについて研究を進めてきた。文献 [12] では、無線通信環境でのマルチメディアコンテンツ配信のためのバッファ管理について議論している。文献 [6][13] では、QoS 制御を行うためのプロトコル階層構造を定義し、各層内/間でのパラメータ処理および交換の整理、モデル化を行っている。また、文献 [7] では、サーバにおける QoS 制御アルゴリズムとして、ある再生レートでエンコーディングされたマルチメディアコンテンツをサーバがクライアントへ送信する際に、ダイナミックに画質の調整等の送出量制御を行うアルゴリズムを提案し、文献 [11] においてその分析を行っている。一方、文献 [8] では、クライアントがマルチメディアストリーミングサービスを受けている途中で断線してしまった際に、その状態を一時的に保存しておき、通信路復帰後にコンテンツを続きから再生できるための、セッション管理プロトコルの拡張を行っている。

本報告では、サーバにおける QoS 制御アルゴリズムとしてこれまでに提案している方式に対して、も

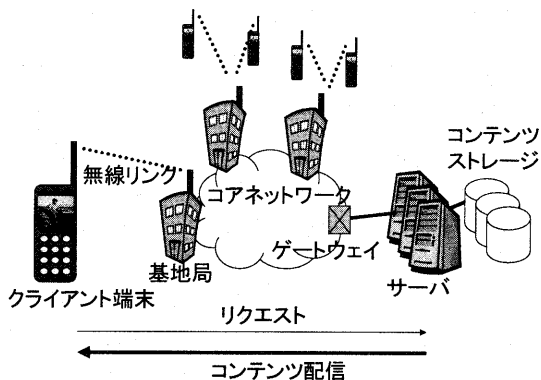


図1: 想定する環境

う一つの新しい QoS 制御アルゴリズムを提案し、比較を行う。本報告で提案する QoS 制御アルゴリズムは、実装上の問題点を解決するための様々なアプローチを追加したものである。ファイル切り替え方式とは、サーバが複数の再生レートでエンコーディングされた複数のマルチメディアコンテンツを管理しておき、帯域幅の変化に伴って、クライアントに送信するコンテンツのファイル自体を切り替える方式である。また、ファイル切替方式とこれまでのモバイルマルチメディア QoS 制御システムで用いている方式とを定量比較し、その特徴を明らかにする。

以下、2 章ではこれまでに実装しているモバイルマルチメディア QoS 制御システムについて説明する。特に、QoS 制御アルゴリズムとして用いている、オンザフライトランスコーディング方式 [7] について詳説する。次に 3 章では今回提案するファイル切替方式について説明する。4 章でオンザフライトランスコーディング方式とファイル切替方式との定量比較を行い、それぞれの特徴を明らかにした上で、より優れた制御アルゴリズムについて考察する。最後に 5 章で本報告のまとめを行う。

2. モバイルマルチメディア QoS 制御システムについて

本章では、マルチメディアストリーミングサービスを無線通信環境において実現するためのいくつかの課題について述べ、それらの課題を解決するために、筆者らの研究グループがこれまでに提案している、モバイルマルチメディア QoS 制御システムの概要について説明する。

2.1 無線通信環境でのマルチメディアストリーミングサービスの問題点

一般に無線通信環境は、遮蔽物等の存在によるパケット損失や、通信可能範囲が存在することなど、有線通信環境と異なる特性をもつことが多い。また、各無線アクセスメディアは有線と異なる制御を用いている。例えば IMT-2000 では、無線通信区間におけるパケット再送とそれに伴うバッファの存在、および異なる帯域幅をもつチャネルの切り替えなどがそれに当たる。この中で、筆者らの研究グループが目的としている、クライアントにおける途切れのないコンテンツの再生を実現するために問題となるのは、無線区間でのパケット再送とチャネルの切り替えである。クライアントにおいてコンテンツを連続再生するためには、少なくとも再生と同じスピードで、クライアントがコンテンツを受信する必要がある。つまり、高遅延はクライアントにおけるコンテンツ連続再生のための直接の要因とならないため、再送バッファの存在による遅延はここでは大きな問題にならない。一方、無線区間でのパケット再送は、遅延のばらつきを誘発するため、クライアントがあるフレームを再生しようとしたときに、そのフレームを構成するためのパケットがクライアントに到達しておらず再生できない場合がある。同様に、チャネル切り替えによって、帯域幅が離散的に大きく変動するため、それによって遅延が大きく変化し、クライアントでの再生時刻にパケットが間に合わない場合が考えられる。

また、無線通信環境では、ユーザが意図しない無線通信路の断線が頻繁に発生する。有線通信環境を想定した従来のマルチメディアストリーミングサービスでは、このような場合、セッションがタイムアウトしてしまう。この現象は有線通信環境においても輻輳などによって起こり得るのだが、無線通信環境ではより頻繁に発生することが予想されるため、解決すべき重要な課題の一つである。

2.2 モバイルマルチメディア QoS 制御システム

筆者らの研究グループでは、2.1節で述べた無線通信環境特有の問題点を解決するためのモバイルマルチメディア QoS 制御システムの提案、実装を行っている。このシステムでは、基本的に End-to-End での制御を前提としており、サーバはクライアントから得られるフィードバック情報のみに基づいて送量制御を行う。クライアントがサーバへ返すフィー

ドバック情報には、次の二つの情報が含まれる。

- クライアントの受信ビットレート
- パケットロス数

サーバではクライアントの受信ビットレートに合わせて送出ビットレートを調整する。また、受信ビットレートが下がっていない場合でも、パケットがロスしているときは、なんらかの障害が発生しているものと判断し、送出ビットレートに小さな値を設定する。

また、クライアントがサーバからコンテンツを受信、再生している途中で断線してしまった場合は、サーバ、クライアントそれぞれがセッションを一時停止し、状態を保持しておく。そして再びクライアントが通信可能状態に復帰すると、サーバは保存した情報に基づいてコンテンツの送信を再開し、クライアントは同様にコンテンツの受信および再生を再開する。

モバイルマルチメディア QoS 制御システムでは、UDP/IP 上で RTP[9]を用いて、Mpeg4 Simple Profile でエンコードされたマルチメディアコンテンツを配信する[5]。また、サーバ、コンテンツ間での情報交換には、RTP の制御プロトコルである RTCP の SR メッセージ、RR メッセージおよび APP メッセージを拡張したものを用いる。さらに、断線・復帰時のセッション管理には RTSP[10]と SDP[2]を拡張したものを用いている。

2.3 オンザフラインランスコーディング方式の概要

本節では、筆者らの研究グループがこれまでに提案、実装しているモバイルマルチメディア QoS 制御システムにおける、サーバの QoS 制御アルゴリズムである、オンザフラインランスコーディング方式について説明する。このシステムのサーバは、2.2節で述べたように、クライアントからのフィードバック情報から得られる、クライアントの受信ビットレートに基づいて、QoS 制御を行う。以下にアルゴリズムを示す。

1. サーバは、クライアントからフィードバック情報を受け取ると、その中からクライアントの受信ビットレートを取り出す。
2. 受信ビットレートが送信ビットレートより高い場合、受信ビットレートより高めの値にコンテンツビットレートを設定し、それと同じビットレートで送出する。それに対して、受信ビットレートが送信ビットレートより低い

場合、受信ビットレートより低めの値にコンテンツビットレートを設定し、それと同じビットレートで送出する。

このアルゴリズムでは、受信ビットレートと同じ値にコンテンツビットレートを設定するのではなく、受信ビットレートより高め、あるいは低めの値に設定している。これによって、クライアントの受信ビットレートが変化していない場合でも、サーバの送信ビットレートはクライアントの受信ビットレートの値付近を上下してしまい、収束しない可能性がある。このような制御を行う理由を、次に示す。

- 実効帯域の上昇は減少と比べると検出が困難である。例えば、192Kbpsの実効帯域に対して192Kbpsでサーバがコンテンツを送出している場合、実効帯域が384Kbpsに上昇しても、サーバが192Kbpsで送出している限り、クライアントは192Kbps付近のビットレートでコンテンツを受信するため、サーバはクライアントの受信ビットレートから実効帯域が上昇したことを検出できない。そのため、送信ビットレートより受信ビットレートの方が高い場合は受信ビットレートより高めの値にコンテンツビットレートを設定し、実効帯域が上昇しているかどうかを調べている。
- 実行帯域に対して、サーバが高い値のコンテンツビットレートを設定してしまうと、それによってネットワーク中に滞留するパケットが発生する。これが何度も繰り返されると、中継局においてバッファ溢れが発生する場合がある。そのため、コンテンツビットレートより受信ビットレートが低い場合は、受信ビットレートより小さめのコンテンツレートを設定することで、ネットワーク中のパケットの滞留を解消する。

3. ファイル切り替え方式

本章では、もう一つの送出量制御方式である、ファイル切替方式について説明する。送出量制御方式はモバイルマルチメディア QoS 制御システムの他の部分と独立であり、ファイル切替方式を採用しているサーバをオンザフライトランスコーディング方式を採用するサーバと入れ替えることが可能である。

3.1 制御手段について

ファイル切替方式では、オンザフライトランスコーディング方式と同様に、クライアントからのフィードバック情報により得られる、クライアントの受信レートおよびパケット数に関する情報に基づいて送出量を制御する。また、これも同様に、送出量の制御は、画質およびフレーム数の調整によって行う。

ファイル切替方式では、サーバは異なるビットレートでエンコードされた、複数のコンテンツファイルを保持している。クライアントからのフィードバック情報に基づいて、ストリーミング中にクライアントへ配信するファイルを動的に切り替えることで、画質の調整を行う。また、フレーム数の調整はオンザフライトランスコーディング方式と同様に、ストリーミング中の P ピクチャを削減することで実現する。なお、今回実装したシステムでは、制御の簡化のために、ファイルの種類とフレーム数の組み合わせをモードと呼び、クライアントからのフィードバック情報に基づいてモードを切り替えることで本方式を実現している。以下では、ビットレートの高いモードに切り替えることを、モードを上げる、ビットレートの低いモードに切り替えることを、モードを下げる、と呼ぶ。

3.2 コンテンツビットレートと送出ビットレートの独立化

制御の困難さや物理記憶領域の限界から、ファイル切替方式においてサーバが保持するファイル数は制限される。そのため、コンテンツビットレートを変化させるとき、オンザフライトランスコーディング方式と比較すると、変化の粒度が大きくなってしまふ。これは、コンテンツビットレートを下げるときは問題ではないが、コンテンツビットレートを上げるときに大きな問題となる。なぜなら、もし実効帯域が大きくなっていないときに、サーバが誤ってコンテンツビットレートを上げてしまうと、それによってネットワークに大きな負荷がかかり、クライアントでのコンテンツ再生が破綻してしまう可能性があるからである。この問題を解決するために、ファイル切替方式では、コンテンツビットレートと送出ビットレートを独立に制御できるようにしている。ここで、送出ビットレートとは、単位時間当たりの送出量である。例えば、192Kbpsのビットレートのコンテンツを210Kbpsで送出することができる。

ファイル切替方式では、これを用いて、トライアルモードと呼ぶモードを用意している。このモード

```

IF trial_mode
  IF packet_loss
    trial.stop
  ELSE IF receive_rate <= send_rate * rate_threshold_1
    trial.stop
  ELSE
    mode.up
ELSE
  IF receive_rate < send_rate || receive_rate_av <= send_rate_av * rate_threshold_2
    IF rtt > rtt_threshold
      mode.down
    ELSE IF receive_rate <= send_rate * rate_threshold_3
      mode.down
  ELSE IF packet_loss
    mode.down
  ELSE IF receive_rate >= send_rate
    trial.start

```

図2：制御アルゴリズム

は、サーバがモードを上げる前に、本当に実効帯域が大きくなっているかを調べるためのもので、サーバはコンテンツビットレートよりも大きめの値に送出ビットレートを設定し、その結果得られるフィードバック情報に基づいてモードの切り替えを行う。これにより、コンテンツビットレートを上げる際の変化の粒度の問題をある程度解決できる。

3.3 制御アルゴリズムについて

本説では、モードの切り替えを用いた、実際の制御アルゴリズムについて説明する。アルゴリズムの流れを図2に示す。以下では、図2の流れに沿って、アルゴリズムを説明する。

1. サーバは、クライアントからフィードバック情報を受け取ると、まずはじめに自身がトライアルモードで動作しているかどうかをチェックする。トライアルモードで動作しており、なおかつフィードバック情報からパケットロスの発生を検出した場合あるいは受信ビットレートが送出ビットレートの一定倍よりも小さい場合はトライアルが失敗したと判断し、トライアルモードを中止する。それ以外の場合はトライアルが成功したものと判断し、モードを上げる。
2. トライアルモードで動作しておらず、受信ビットレートが送出ビットレートより小さい、もしくは過去数秒間の受信ビットレート平均が送出ビットレート平均の一定倍より小さい場合は、以下の制御を行う。まずRTTを調べ、RTTが一定値より小さい場合はモードを下げる。RTTが一定値より大きい場合でも、受信

ビットレートが送出ビットレートより大幅に小さい場合もモードを下げる。

3. トライアルモードで動作しておらず、なおかつ受信ビットレートが良好な状態であっても、パケットロスが発生するとモードを下げる。
4. それ以外の場合で受信ビットレートが送出ビットレートよりも大きい場合はトライアルモードを開始する。

4. QoS 制御方式の比較

本章では、ファイル切替方式およびオンザフライトランスコーディング方式を定量的に比較し、それぞれの特徴について考察する。測定は、それぞれの方式を実装したサーバを用意し、W-CDMA エミュレータ[4]をサーバ、クライアント間に接続して行った。

以下の実験では、オンザフライトランスコーディング方式はGOP構成がIPPPPで、360Kbpsでエンコードした映像ストリームと8Kbpsの音声ストリームをコンテンツとして用いた。一方、ファイル切り替え方式では、GOP構成がIPPPPで、それぞれ384Kbps、256Kbps、192Kbpsでエンコードされた3個のファイルによって映像ストリームを構成した。なお、ファイル切り替え方式で用いた音声ストリームはオンザフライトランスコーディング方式と同様に8Kbpsのものである。

4.1 大きな帯域変化に対する耐性について

まず、帯域が384Kbpsから64Kbpsに変化した場合のサーバの耐性を比較した。IMT-2000シミュレータでは、通信中に割り当てられたチャネル速度を擬似的に変更できる。その機能を用いて、通信中に

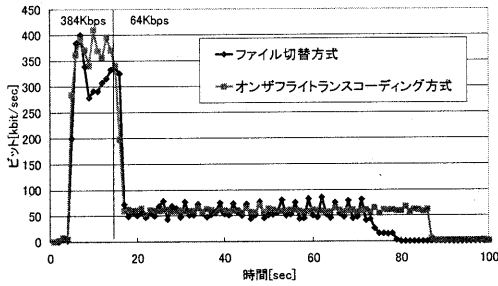


図 3：送出力の変化(384Kbps→64Kbps)

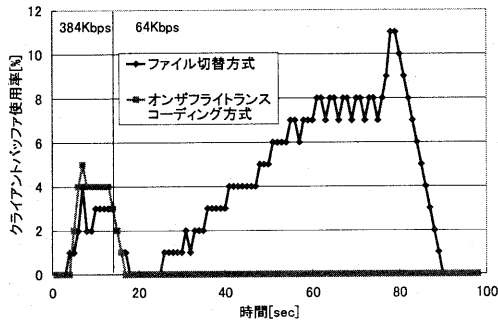


図 4：クライアントバッファ使用率
(384Kbps→64Kbps)

384Kbps のチャンネルから 64Kbps の帯域に変更した。その結果得られたサーバの送出レートの変化を図 3 に示す。図中の横軸は経過時間、縦軸はサーバの送出レートを表している。グラフ中に縦の線で区切られた領域は、左が 384Kbps のチャンネルが割り当てられている時間、右が 64Kbps のチャンネルが割り当てられている時間を表している。このグラフから、どちらのサーバにおいても、ほとんどかわらない実効帯域への追従性が認められる。次に、図 4 にこの実験時のクライアントバッファの変化を示す。横軸は時間を縦軸はバッファ使用率を表す。縦の線で区切られた二つの領域は図 3 と同様の意味である。なお、今回実装したクライアントでは再生時刻に間に合っていないパケットはバッファリングせず破棄する。このグラフから、ファイル切替方式では一時バッファが枯渇するものの、すぐに回復しているのに対して、オンザフラインスコーディング方式では一度バッファが枯渇してしまうと、二度と復帰していないことがわかる。つまり、オンザフラインスコーディング方式では、クライアントがコンテンツを再生できない状態になってしまっている。この結果から、オンザフラインスコーディング方式では、DCT 係数の削減によって得られる送出力制御幅

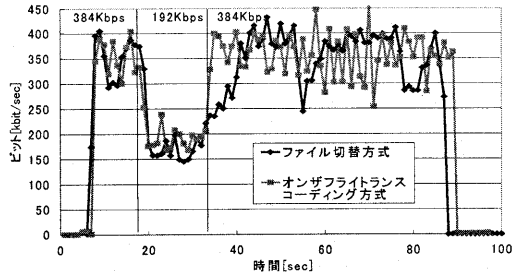


図 5：送出力の変化

(384Kbps→192Kbps→384Kbps)

に限界があるため、64Kbps の実効帯域に対してやや大きめの値で送出してしまっていると考えられる。それぞれのサーバが送出量を下げ終わってから、コンテンツの送出が終了するまでの間の平均送信ビットレートを計算すると、ファイル切替方式では 53.5Kbps であるのに対して、オンザフラインスコーディング方式では 59.0Kbps であった。通信時にはこれにヘッダが付加されるため、64Kbps の帯域を超えてしまったものと考えられる。ただし、現在の実装では P フレームのみを削減の対象としているため、今後 I フレームも削減することで 64Kbps の実効帯域でも対応できる可能性がある。

4.2 送出力増加の即時性

次に、実効帯域が大きくなったときに送出量を上げる処理の実効帯域に対する追従性を比較する。この実験では、実効帯域を 384Kbps から 192Kbps に変化させ、さらに 384Kbps に戻す。その結果として、図 5 にサーバの送出レートの変化を示す。縦軸は時間、横軸は送出ビットレートを表す。また、縦の線で区切られた三つの領域は、左から実効帯域が 384Kbps、192Kbps、384Kbps である時間を表す。このグラフから、帯域が 384Kbps に復帰したときに、それに追従して送出レートが回復するのに必要な時間が、ファイル切替方式よりオンザフラインスコーディング方式の方が約 10 秒程度早いことがわかる。これは、ファイル切替方式では、モードを上げることによるデメリットが大きいため、トライアルモードの時間を設けて、そのときに一定の条件を満たさなければ送出レートが復帰しないのに対して、オンザフラインスコーディング方式では帯域の変化に即座に対応できるようになっているからであると考えられる。

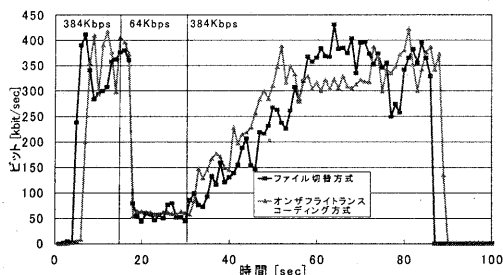


図6：送出量の変化
(384Kbps→64Kbps→384Kbps)

なお、384Kbps から 64Kbps、そして 384Kbps へと実効帯域を変化させた実験の結果を図 6 に示す。この実験では追従性にほとんど違いがない。これは、4.1 節の議論から、64Kbps に実効帯域が変化した時点でオンザフライトランスコーディング方式のサーバが動作対象外の状態になってしまっているため、それが復帰にも影響しているものと考えられる。しかし、このような場合を除けば、オンザフライトランスコーディング方式がファイル切り替え方式と比較して帯域変動に対する追従性が高いことは定性的な観点からも明らかである。

4.3 考察

以上の結果を考察すると、次のようになる。

- ファイル切替方式は大きな帯域変化に対する追従性に優れるが、送出量制御の粒度が大きいため、特に送出レートを上げる際の早さに問題がある。
- オンザフライトランスコーディング方式は、実効帯域に対する送出量の追従性に優れるが、大きな帯域変化には対応できない場合がある。今後フレームの削減方法を拡張することでより大きな帯域変化に対応できるようになると予想されるが、今後最大 11Mbps もの広帯域が利用可能な無線 LAN などでの実験を行った場合、すべての帯域に対応するのはやはり困難である。

このことから、大きく帯域が変化するときにはファイル切替で対応し、小さい帯域変化の場合はオンザフライのトランスコーディングで対応する、二つの方式を組み合わせた方式が有効であると考えられる。筆者らの研究グループでは、現在このような制御が可能なサーバの制御アルゴリズムを実装中である。

5. まとめ

本報告では、これまでに実装したモバイルマルチメディア QoS 制御システムに追加可能なサーバの送出量制御方式として、ファイル切替方式を提案した。さらに元来このシステムで用いていた方式とファイル切り替え方式を定量的に比較し、それぞれの特徴を明らかにした上で、この二つを組み合わせた方式が有効であるとの結論を導いた。今後は二方式を組み合わせたサーバの QoS 制御方式の検討と実装、FOMA 等の実環境での評価実験を進める。

文 献

- [1] D. Chalmers and M. Sloman, "A Survey of Quality of Service in Mobile Computing Environments," IEEE Communications Surveys, Second Quarter 1999, pp. 2-10, 1999.
- [2] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC2327, IETF, Apr. 1998.
- [3] H. Holma and A. Toskala, "WCDMA for UMTS," John Wiley and Sons, LTD., 2000.
- [4] 稲村浩, 石川太郎, 高橋修, "W-CDMA 環境での TCP トラフィック特性評価," 情報処理学会 MBL 研究会研究報告, sep. 2001.
- [5] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. matsui, H. kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams," RFC3016, IETF, Nov. 2001.
- [6] 串田高幸, 富田アルベルト, 黒川雅人, 山内長承, 尾上裕子, 安木成比古, 渥美幸雄, 高橋修, "モバイルマルチメディア QoS の構成方法," マルチメディア, 分散, 協調とモバイル (DICOMO 2001) シンポジウム論文集, pp. 723-728, June. 2001.
- [7] 村尾高秋, 谷口雅昭, 串田高幸, 萩野浩明, 尾上裕子, 高橋修, "ワイヤレス区間を想定したビデオストリーミングシステム," 情報処理学会マルチメディア通信と分散処理(DPS)研究会研究報告, Sep. 2001.
- [8] 尾上裕子, 安木成比古, 渥美幸雄, 成乃箴, 村尾高秋, 串田高幸, 山内長承, "マルチメディアセッション制御プロトコルにおけるモビリティ拡張機能について," 情報処理学会モバイルコンピューティングとワイヤレス通信 (MBL)研究会研究報告 18-34, pp.253-259, Sep. 2001.
- [9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Streaming Protocol," RFC1998, IETF, Jan. 1996.
- [10] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol," RFC2326, IETF, Apr. 1998.
- [11] 谷口雅昭, 村尾高秋, 串田高幸, 萩野浩明, 安木成比古, 稲村浩, 山内長承, "無線ネットワーク環境における帯域幅の予測とマルチメディア・コンテンツの送出量制御," マルチメディア通信と分散処理(DPS)研究会研究報告, pp.65-70, Nov. 2001.
- [12] 山内長承, 串田高幸, 富田アルベルト, 尾上裕子, 渥美幸雄, "再送をとまなう無線区間を持つインターネットでの動画転送時のバッファ制御," 情報処理学会マルチメディア通信と分散処理(DPS)研究会研究報告 103-7, pp.37-42, Feb. 2001.
- [13] 安木成比古, 渥美幸雄, 高橋修, 尾上裕子, 黒川雅人, 串田高幸, 富田アルベルト, 山内長承, "モバイルストリーミングのための QoS 制御フレームワーク," マルチメディア, 分散, 協調とモバイル (DICOMO 2001) シンポジウム論文集, pp.717-722, June. 2001.