

## 複数端末を用いた同期分散型マルチメディアデータ伝送のための一方式

齊藤義仰† 小西洋祐† 坂根裕†† 石原進††† 水野忠則††

† 静岡大学大学院情報学研究科

†† 静岡大学情報学部

††† 静岡大学工学部

あらまし 筆者らはこれまでに、PHSなどの低速な通信機能しか持たない携帯端末ユーザであっても、ネットワークを介したスポーツ中継などのライブ放送が快適に鑑賞できるように、高速かつ効率的なマルチメディアデータ伝送方式 Multitrack を提案している。Multitrack は、筆者らが提案している通信回線共有方式 SHAKE に基づき、複数端末を一時的に集めてクラスタネットワークを構成し、各端末の持つ無線リンクを並列に束ね、通信データを共有することで高速なマルチメディアデータ伝送を実現する。さらに、クラスタ内の複数端末上で受信したマルチメディアデータを共有することで、通信にかかわる全ての端末に通信速度向上の利益を与える。しかしながら、帯域幅や遅延が異なる複数のリンクをマルチメディアデータ伝送に用いた場合、適切なパケット分配が行われないとパケットの到着順序の逆転が頻繁に起こり、データ再生が不安定になるという問題がある。本稿では、すでに提案しているパケット分配時にデータの到着時刻を予測することでパケットを順序通りに到着させるパケット分配方式について、シミュレーションに基づく基礎評価を述べる。

キーワード マルチメディアデータ伝送, 複数経路の共有, パケット分配方式, Multitrack, SHAKE

## A Method of Synchronous Distributed Multimedia Data Transmission using Multiple Links

YOSHIA SAITO†, YOUSUKE KONISHI†, YUTAKA SAKANE††, SUSUMU ISHIHARA†††, and

TADANORI MIZUNO††

† Graduate School of Information, Shizuoka University

†† Faculty of Information, Shizuoka University

††† Faculty of Engineering, Shizuoka University

**Abstract** Authors have proposed a system, Multitrack, realizes high speed multimedia data transmission, even on low speed links. The Multitrack is based on SHAKE technology which authors have proposed. In the Multitrack system, two or more mobile terminals are connected temporarily, and form a cluster network. Streaming multimedia data is transmitted via multiple links which mobile terminals in a cluster network have. The multimedia data can be received by all the mobile hosts in the cluster. we proposed traffic distribution methods to solve problems on transmitting multimedia data with multiple links that have different properties. In this paper, we analyzed the proposed method with simulations.

**Key words** Multimedia Data Transmission, Sharing Multiple Links, Traffic Distribution, Multitrack, SHAKE

### 1. はじめに

近年、マルチメディアコンテンツはデスクトップ PC に限らず、ノート PC や PDA、携帯電話などの携帯端末でも利用できる環境が整いつつある。IMT2000 のサービスである NTT DoCoMo の FOMA [2] では、上り最大 64kbps・下り最大 384kbps の高速パケット通信が可能であり、携帯電話上で映像や音楽が

楽しめるようになった。

しかしながら、IMT2000 サービスの通信料金は高く、サービス提供地域は限定されている。このため、マルチメディアコンテンツを利用できるのは少数の限られたユーザだけであり、現在広く普及している PHS などの低速な通信機能しか持たない携帯端末ユーザは、映画などのコンテンツを鑑賞できない。

筆者らは文献 [1] で、マルチメディアデータのビットレー

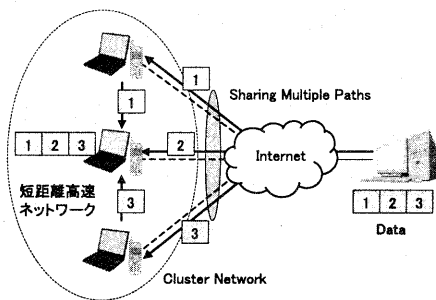


図1 SHAKE

トに対し受信端末の帯域幅が狭い場合でも、安定したストリーミング再生を実現し、さらに複数端末上で同時に鑑賞可能にする高速かつ効率的なマルチメディアデータ伝送方式 Multitrack(MULTImedia data TRAnsmisssion for Cluster networkK)を提案した。Multitrackは筆者らが提案した通信回線共有方式 SHAKE [3] の概念に基づき、利用できる複数の無線リンクを一時的に並列に束ねることで、高品質なマルチメディア伝送を実現するものである。

しかしながら、複数のリンクを並列に用いたデータ伝送では、帯域幅や遅延といった各リンクの品質の違いにより、パケットの到着順序が安定しない。これは、即時性が要求されるストリーミングでは大きな問題となる。文献[1]では、各リンクの品質からパケットが到着するまでの時間を予測し、最も早く到着するリンクに対しパケットを分配することでパケットの到着順序を安定させるパケット分配方式を提案している。本稿では、提案手法の有効性をシミュレーションにより検証した。

以下本稿では、2章で低速な携帯端末でも高速な通信を実現する SHAKE の概念について述べる。3章では SHAKE を応用したマルチメディア伝送方式 Multitrack について述べ、パケットを分配する上での問題と、いくつかのパケット分配方式について述べる。4章では3章で提案したパケット分配方式を評価するシミュレーションモデルについて述べる。5章でシミュレーションの結果と考察について述べ、6章にまとめを示す。

## 2. SHAKE

本稿で扱う Multitrack は、SHAKE の概念を応用したシステムである。SHAKE(SHARing multiple paths procedure for cluster networkK Enviroment)では、複数の携帯端末を一時的に集め、無線 LAN や bluetooth といった短距離高速ネットワークで各々を相互に接続しクラスタネットワークを構築する。このクラスタネットワーク内の各端末がもつ外部通信路を論理的に束ね、複数の端末で外部通信路を共有する。共有された外部通信路を用いて図1のようにデータを分配し伝送することで、低帯域幅の端末でも高速な通信を実現できる。

現在、筆者らのグループでは、IP 層でのアプローチとアプリケーションレベルでのアプローチで SHAKE の研究を進めている。IP 層でのアプローチとして、IP SHAKE [4] を提案している。IP SHAKE では、ネットワーク上の固定の位置、あるいは

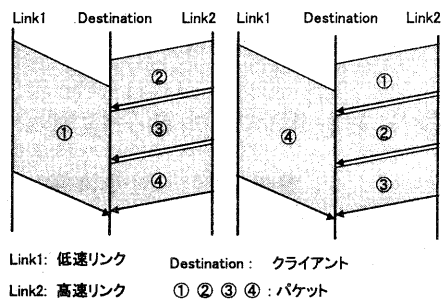


図2 パケットの到着順序

任意の位置に IP 層でデータを分配するホストを配置する。クラスタネットワーク内の端末がデータの送受信を行う場合は、そのホストを経由させることにより、各端末がもつ共有された複数の外部通信路にデータを分配し、高速な通信を実現する。アプリケーションレベルでのアプローチとしては、HTTP に SHAKE の概念を適応した Web SHAKE [5] が提案されている。

IP SHAKE や Web SHAKE の仕組みを利用すれば、個々の端末の持つ通信速度が遅くても、それらを複数台同時に利用することで、大容量のデータ伝送が可能となる。しかしながら、大容量のマルチメディアデータ伝送を行う場合、複数リンクの同時使用による通信速度向上の恩恵を得られるのは、クラスタ内の一部の端末のみである。たとえば、映画などを鑑賞する場合、受信者となった一人だけが映画を見ることができ、リンクを貸与している他の端末のユーザは映画が見られない。リンクをクラスタ内のほかの端末に提供しトラフィックの中継を行う端末も、受信したマルチメディアコンテンツを鑑賞できる仕組みがあれば、クラスタ内の端末全てが恩恵を得ることができる。

## 3. Multitrack

### 3.1 概要

Multitrack では、SHAKE と同様に複数の携帯端末を一時的に集め、無線 LAN や bluetooth といった短距離高速ネットワークで各端末を相互に接続し、クラスタネットワークを構築する。クラスタネットワーク内の各端末は、それぞれが持つ外部リンクを複数同時に使用し、ストリーミングサーバからマルチメディアデータを分割受信する。分割受信したデータをクラスタネットワーク内で相互に共有することで、クラスタ内の全ての端末で高品質なストリーミング再生を実現する。

### 3.2 パケット分配

Multitrack を実現する上で、パケット分配アルゴリズムは全体の性能に大きな影響を及ぼす。なぜなら、リンク間に帯域幅の差がある場合、帯域幅に合わせたパケット分配を行わないと帯域の狭いリンクの影響を受けて、全体のスループットが低下してしまうからである。また、リンク間に一定以上の遅延差があった場合、パケットの到着順序の逆転が頻発し、再生時に必要なデータが到着せず、ストリーミングデータ再生に悪影響を与える。図2の左図で示すように、連続したパケット1, 2, 3, 4があった場合、帯域幅と遅延の違いにより、2, 3, 4, 1 という順

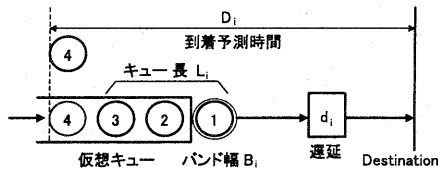


図3 仮想送信キュー

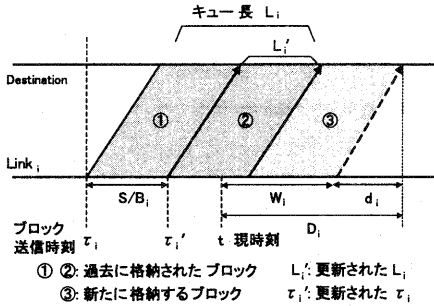


図4 パケット到着時間予測

番でパケットが到着すると、1が到着するまでデータの再生を開始できない。パケットを分配するホストは、クライアントに適切な順序でパケットを到着させるため、図2の右図のような遅延と帯域幅を考慮した適切なパケット分配を行う必要がある。

本稿では、シンプルで実現が容易であるという点から、パケット分配方式として次の5つの方式を考えた。

方式1: 各リンクの帯域幅の比を重みとしてランダムに分配する。

方式2: 各リンクの帯域幅の比を重みとして順番に分配する。

方式3: クライアントからのフィードバックで、パケット到着までの遅延時間を得、その逆比を重みとしてランダムに分配する。

方式4: 送信キューの待ち時間が最も短いリンクに分配する。

方式5: 送信キューの待ち時間とクライアントからフィードバックされた遅延時間の和を求め、パケットが最も早く到着すると予測されるリンクに分配する。これは筆者らが文献[1]で提案した方式である。

方式4, 5における送信キューの待ち時間は、仮想的な送信キューを想定した以下の式で予測する。図3に示すように、リンク番号:  $i$ 、パケットサイズ:  $S$ 、帯域幅:  $B_i$ 、キュー長:  $L_i$ 、現在送信中のブロックが送信された時刻:  $\tau_i$ 、現在時刻:  $t$  とすると、送信キューの待ち時間:  $W_i$  は図4より、以下の式で求められる。

$$W_i = \frac{S}{B} (L_i + 1) + \tau_i - t$$

$W_i$  にクライアントからフィードバックされた遅延時間:  $d_i$  を加えることで、方式5のパケット到着までの予測時間:  $D_i$  が求められる。

$$D_i = W_i + d_i$$

方式5では、 $D_i$  が最小となるリンク  $i$  を選び、パケットを分

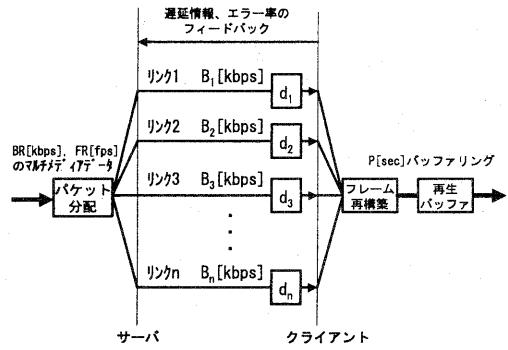


図5 シミュレーションモデル

配することで順序通りにパケットを到着させることを目指している。

#### 4. シミュレーションモデル

前章で提案した5つのパケット分配方式を評価するため、シミュレーションを行い評価した。シミュレーションモデルを図5に示す。本シミュレーションではサーバから  $b$ [kbps]、 $f$ [fps] のマルチメディアデータを、 $i$ 本のリンク ( $i = 1, 2, \dots, n$ ) を同時に用いてクライアントへ配信する。各リンクの最大帯域幅:  $B_i$  は既知であるとし、配信開始時点の遅延:  $d_i$  も計測されているものとする。プロトコルはUDPを想定し、パケットサイズは  $S$ [byte] とした。

最初にサーバでは、 $1/f$  毎に発生するフレームをパケットに格納し、 $S$ [byte] に達した時点で各リンクに分配し送信する。各パケットにはパケットロス検出のためにシーケンス番号を与え、遅延を計測するためパケットにタイムスタンプを付加する。

各リンクの品質は、バックグラウンドトラフィック、遅延、パケットロス率により制御した。バックグラウンドトラフィックはストリーミングデータ以外にリンクに流れているトラフィックである。バックグラウンドトラフィックを変化させることで、リンクの利用可能な帯域幅を変化させることができる。遅延は文献[6]のモデルを用いた。片側の固定遅延時間:  $DC_i$ [msec]、遅延ゆらぎ:  $DV_i$ [msec] とし、平均1秒の指数分布に従う時間間隔で、平均  $DV_i$ [msec] の指数分布に従う遅延時間を発生させ遅延ゆらぎとした。

クライアントにパケットが到着すると、分割されたフレームの再構築を行い、再生バッファに格納する。また、パケットに付加されたタイムスタンプから遅延時間を算出し、一定間隔毎にサーバにフィードバックする。しかし、この方法ではリンクにパケットが分配されない場合、遅延情報を得ることができない。そこで、一定時間(本シミュレーションでは3[sec])リンクにパケットが分配されない場合は、そのリンクにタイムスタンプを付加したパケットを送信し、遅延情報を更新する。さらに、シーケンス番号からパケットロスを検出し、エラー率を遅延情報と共にフィードバックする。

最後に、クライアントは  $P$ [sec] のバッファリングを行った後、

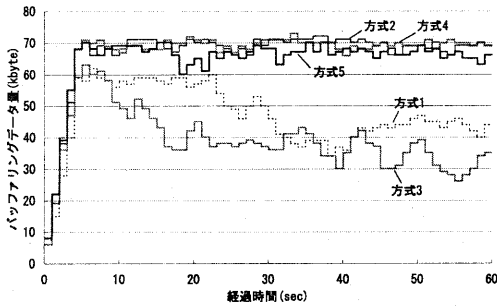


図6 対称なリンク使用時の性能：バッファリングデータ量

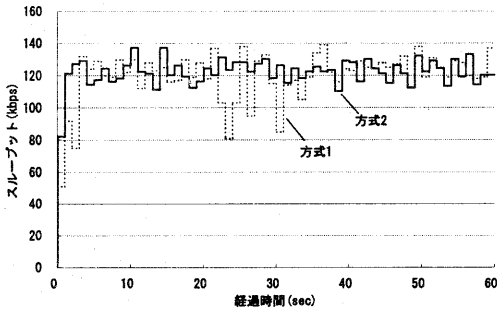


図7 対称なリンク時の性能：スループット

再生バッファから  $1/f$  毎に1フレームずつ順番に取り出す。再生バッファに取り出すはずのフレームがない場合は、フレームドロップしたものとみなすが、再送処理は行わない。

なお、本シミュレーションでは、パケット分配やフレームの再構築処理にかかる時間は考慮していない。

## 5. シミュレーション結果と考察

ここでは、リンクの品質やリンク数を変化させた場合における、方式1~5の性能を評価した。評価は各方式の再生バッファに格納されているデータ量や、フレームドロップ数を比較することで行った。

本シミュレーションでは、リンク品質の初期値をバックグラウンドトラフィック：0[kbps]、固定遅延  $DC$ ：300[msec]、遅延ゆらぎ  $DV$ ：100[msec]、パケットロス率：1[%]とし、上りリンクと下りリンクは対称とした。また、パケットサイズは1500[byte]とし、遅延とエラー率のフィードバック間隔は3[sec]とした。

### 5.1 対称なリンクを用いた場合

64kbpsのリンクを2本使用し、128kbps(15fps)の動画相当データを送信をした。各方式のバッファリングデータ量の変移を図6に示す。

シミュレーションの結果、方式2, 4, 5のバッファリングデータ量はほぼ安定しているのに対し、方式1, 3は不安定であった。これは、方式1, 3は比率を重みとしたランダムな分配を行うため、片側のリンクに偏ってしまうことがあり、2本のリンクを十分に活用できないからだと考えられる。安定したバッ

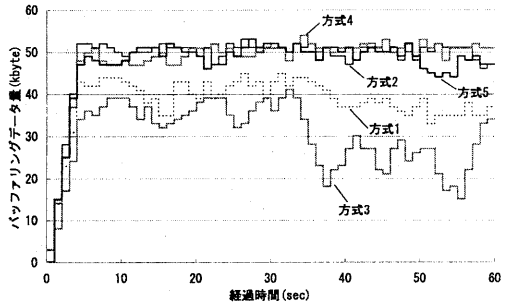


図8 非対称なリンク使用時の性能：バッファリングデータ量

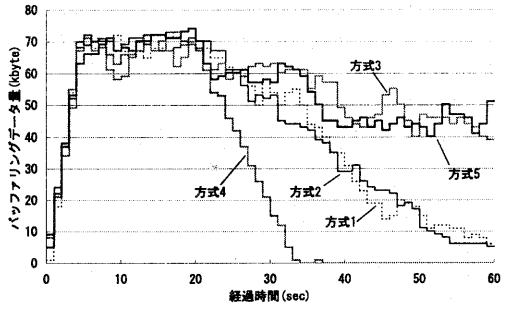


図9 バックグラウンドトラフィック変化時の性能：バッファリングデータ量

ファ量が得られた方式5と、不安定だった方式1のシステム全体のスループットを図7に示す。図7より、方式5は安定して128kbpsのスループットを保っているが、方式1はランダムな分配のため128kbpsのスループットを下回っていることが確認できる。バッファリング時間が長ければフレームドロップを起こさないが、バッファリング時間が短くなるにつれコマ落ちが発生してしまうものと考えられる。

### 5.2 非対称なリンクを用いた場合

リンク1(64kbps)とリンク2(32kbps)を同時に用いて、96kbps(15fps)の動画相当データを送信した。各方式のバッファリングデータ量の変移を図8に示す。

シミュレーションの結果、非対称なリンクを用いた場合も、対称なリンクを用いた場合とほぼ等しい結果となった。このことから、対称なリンクであっても非対称なリンクであっても性能に変化がないことが確認された。

### 5.3 バックグラウンドトラフィック変化が与える影響

リンク1(128kbps)とリンク2(64kbps)を同時に用いて、128kbps(15fps)の動画相当データを送信した。20秒経過後に64kbpsのバックグラウンドトラフィックをリンク1に与え、各方式の性能を評価した。バッファリングデータ量の変移を図9に示す。

シミュレーションの結果、図9からわかるように、20秒までは帯域に余裕があることから、どの方式も安定したバッファリングを実現している。20秒経過後からは方式3, 5は多少のバッファリングデータ量の減少が見られるが、その後安定して

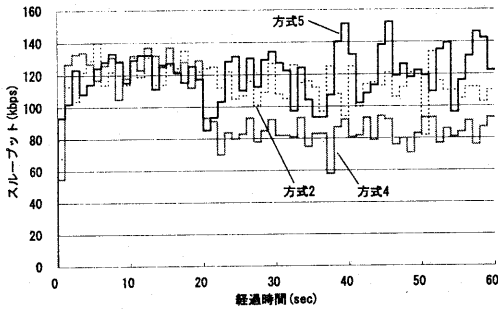


図 10 バックグラウンドトラフィック変化時の性能：スループット

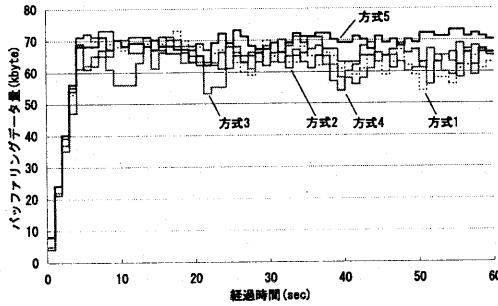


図 11 遅延変化が与える影響

いる。一方、方式 1, 2, 4 はバッファリングデータ量が減少し、後半のフレームはほとんどドロップされていた。これは、方式 3, 5 はクライアントから遅延のフィードバックがあるため、リンク品質が変化した場合にパケット分配率を変更できるのに対し、方式 1, 2, 4 はフィードバックがないため、リンク品質が変化した場合にパケット分配率を変更できないからである。安定したバッファリングを行うことができた方式 5 と、徐々にバッファリングデータ量が減少していった方式 2、急激にバッファが空になった方式 4 のスループットを図 10 に示す。図 10 より 20 秒経過後も方式 5 は平均して 128kbps のスループットを保っているのに対し、方式 2 は 110kbps 程度のスループットしか得られず、方式 4 に至ってはリンク 2 をまったく用いていないためスループットが著しく減少していることが確認できる。

このことから、利用可能な帯域幅に変化がある場合は方式 3, 5 が優れており、クライアントからのフィードバックが重要であることが確認できた。

#### 5.4 遅延変化が与える影響

リンク 1(128kbps) とリンク 2(64kbps) を同時に用いて、128kbps(15fps) の動画相当データを送信した。20 秒経過後に 1000msec の固定遅延をリンク 2 に与え、遅延変化あった場合の性能を評価した。バッファリングデータ量の変移を図 11 に示す。

シミュレーションの結果、方式 5 が他の方式よりも僅かながら優れていた。これは、方式 5 がリンク 2 をほとんど利用しないため、遅延変化の影響を受けなかったと考えられる。しかしながら、各方式の差はわずかであり、遅延変化における顕著な

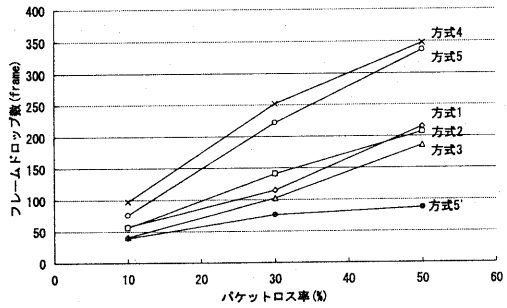


図 12 パケットロスに対するフレームドロップ数

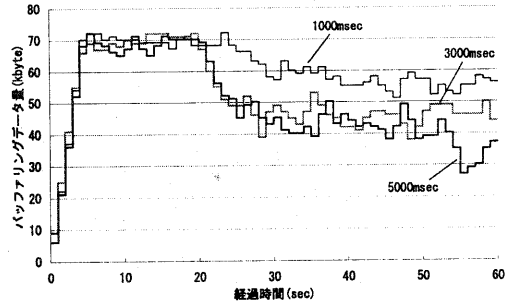


図 13 フィードバック間隔が与える影響

性能差は見られなかった。

#### 5.5 エラー率が与える影響

リンク 1(128kbps) とリンク 2, 3(64kbps) を同時に用いて、128kbps(15fps) の動画相当データを送信した。20 秒経過後にリンク 1 のパケットロス率を 10%, 30%, 50% と変化させ、パケットロス率が増加した場合の性能を評価した。ドロップしたフレーム数を図 12 に示す。

シミュレーションの結果、パケットロス率が高くなるにつれ方式 4, 5 のフレームドロップ数が高くなった。この原因は、パケットロス率が高くても、方式 4, 5 は帯域幅が広いリンク 1 にほとんどのパケットを分配してしまうからである。そこで、あるリンクに一定以上のエラー率(本シミュレーションでは 5%) が検出されると、そのリンクにパケットを分配する場合、他のリンクにパケットの複製を分配するように方式 5 を改良した。これを方式 5' とする。

方式 5' を用いた場合、図 12 からわかるように、フレームドロップを大幅に改善できた。この手法を用いることで、方式 5 が他の方式よりもフレームドロップが増加する問題を解決でき、かつ、他の方式よりもフレームドロップを減少させることが可能であった。

#### 5.6 フィードバック間隔が与える影響

リンク 1(128kbps) とリンク 2(64kbps) を同時に用いて、128kbps(15fps) の動画相当データを送信した。20 秒経過後に 64kbps のバックグラウンドトラフィックをリンク 1 に与え、フィードバック間隔を 1 秒、3 秒、5 秒と変化させた場合の方式 5 の性能を評価した。バッファリングデータ量の変移を図 13

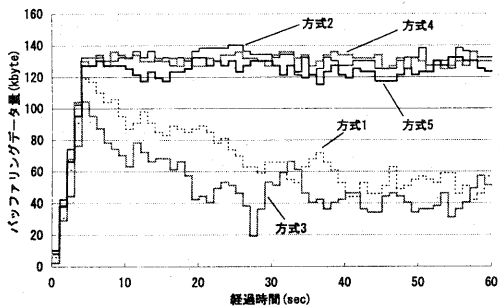


図 14 多数リンク使用の影響

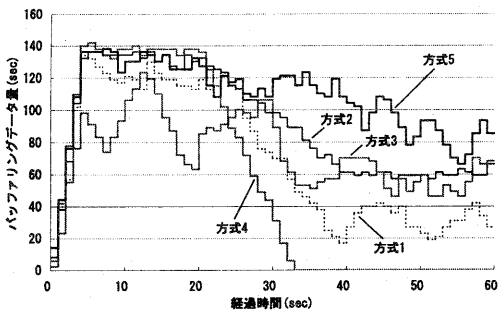


図 15 多数リンク使用 + バックグラウンドトラフィック変化

に示す。

シミュレーションの結果、フィードバック間隔が短いほどリンク品質が変化した場合のバッファリングデータ量減少を抑えることができた。これは、サーバがリンク品質の変化をより早く感知でき、パケット分配比率を変化させることができるためである。同様に、フィードバック情報を利用する方式 3 も、図 13 とほぼ等しい結果となった。したがって、方式 3, 5 を用いる場合は、可能な限りフィードバック間隔を短くすることで、性能を向上できることが確認された。

### 5.7 多数リンク使用の影響

まず、リンク 1, 2, 3(64kbps) とリンク 4, 5(32kbps) の計 5 本のリンクを同時に用いて、256kbps の動画相当データを送信した。バッファリングデータ量の変移を図 14 に示す。

シミュレーションの結果、各方式ともリンク数が増加した場合でも、リンク数が 2 本の場合と同じ特徴を示した。ただし、方式 1, 3 と他の方式とのバッファリング量の差が広がった。これは、リンク数が増加するにつれ、方式 1, 3 は遅延および帯域幅の比率を重みとしたランダムなパケット分配であるため、安定した伝送が困難になったためだと考えられる。

次に、リンク 1(128kbps) とリンク 2, 3(64kbps)、リンク 4, 5(32kbps) を同時に利用して 256kbps の動画相当データを送信した。20 秒経過後に 64kbps のバックグラウンドトラフィックをリンク 1 に与え、リンク数が多く、さらにリンク品質が変化する場合の各方式の性能を評価した。バッファリングデータ量の変移を図 15 に示す。

シミュレーションの結果、リンク数 2 本の図 9 と比較すると、

方式 3 と方式 5 の差が開いたことが確認された。これは方式 3 が前述したように、ランダムなパケット分配による結果と考えられる。一方、リンク数 2 本の場合には不安定なバッファリングであった方式 1 が安定しているように見える。しかし、後半のフレームがほとんどドロップしており、安定した伝送は実現できていなかった。

よって、5 方式の中で方式 5 が最も安定した伝送と、バッファリングデータ量を実現可能であることが確認された。ただし、高速なリンクのエラー率が高くなった場合にフレームドロップが増加するという問題があるため、5.5 節で述べたようなエラー対策手法が必要となる。本稿で述べたエラー対策手法では、複製の送信を開始するエラー率の閾値を適切に設定する必要がある。なぜなら、閾値が低すぎると、冗長なパケットを他のリンクに送信するため、帯域を占有し性能が悪化するが、閾値が高すぎると、フレームドロップ数が増加するからである。適切な閾値の求め方と、他のエラー対策手法の考案が今後の課題である。

## 6. まとめ

本稿では、携帯端末を用いた高速かつ効率的なマルチメディアデータ伝送方式 Multitrack において、安定したマルチメディアデータ伝送を実現するための 5 つのパケット分配方式について、シミュレーションによる評価を行った。その結果、筆者らが文献 [1] で提案した方式が、最も安定したマルチメディアデータ伝送を実現できることが確認できた。提案方式では高速なリンクのエラー率が高くなった場合、フレームドロップ数が増加するという問題があったが、他のリンクに複製データを同時に送信することでフレームドロップを抑えることが可能であった。

今後は、エラー対策手法をさらに追求し、より安定した伝送を目指す。その後、アプリケーション層で Multitrack システムを実装する予定である。

## 文 献

- [1] 齊藤義仰, 小西洋祐, 石原進, 水野忠則: 複数端末を用いた同期分散型マルチメディアデータ伝送の提案, 放送コンピューティング研究グループ 第 1 回研究会 研究報告, pp66-73 (2002).
- [2] FOMA. <http://foma.nttdocomo.co.jp/>
- [3] Mineno, H., Ishihara, S., Ohta, K., Aono, M., Ideguchi, T. and Mizuno, T.: Multiple paths protocol for cluster type network, INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, pp.391-403 (1999).
- [4] 石原進, 飯田峰彦, 橋本豊大, 富強, 水野忠則: IP トンネリングを用いたモバイルマルチパス環境の実現, YRP 移動体通信産学官交流シンポジウム 2000, pp.43-44 (2000).
- [5] Konishi, Y., Ishihara, S., and Mizuno, T.: A fast WWW access method for multiple wireless links shared by mobile computers, In proc. of The 16th International Conference on Information Networking, vol.3, pp.9C-2.1-9C-2.12 (2002).
- [6] 林考典, 山崎真一郎, 森田直人, 相田仁, 武市正人, 土居範久: インターネットを用いた複数経路データ伝送方式の性能評価, 電子情報通信学会論文誌 (B), Vol.J84-B, No.3, pp.523-533 (2001).