

## FECとバッファリングを利用した複数経路リアルタイムストリーミング通信の評価

鈴木 宏 治<sup>†</sup> 峰 野 博 史<sup>††</sup> 田 中 希 世 子<sup>‡</sup>  
鈴木 偉 元<sup>‡</sup> 石 川 憲 洋<sup>‡</sup> 水 野 忠 則<sup>††</sup>

モバイル環境でPAN(Personal Area Network)を形成し、PAN内の端末の通信リソースを共有して通信(PAN2PAN通信)することで、P2Pで通信するより高速・高信頼の通信が可能になると考える。ただし、リアルタイムにマルチメディアデータを扱うPAN2PAN通信では、トラフィックを複数経路に適切に分配し組み立てる仕組みが必要となる。我々は、FEC(Forward Error Correction)とバッファリングによって効率よくPAN2PAN通信を実現する通信制御ミドルウェアを開発している。本稿では、複数経路通信環境においてFEC(Forward Error Correction)とバッファリングを利用することでリアルタイムマルチメディア通信で課題となるパケット損失や遅延・ジッタの影響を軽減できる方式を検討し評価した。

### Evaluation of FEC and Buffering for Multiple Paths Real-time Streaming

KOJI SUZUKI,<sup>†</sup> HIROSHI MINENO,<sup>††</sup> KIYOKO TANAKA,<sup>‡</sup> HIDEHARU SUZUKI,<sup>‡</sup>  
ISHIKAWA NORIHIRO<sup>‡</sup> and TADANORI MIZUNO<sup>††</sup>

We believe that the communication using whole links to the Internet within a PAN (Personal Area Network) provides more reliable and higher bit rate than the peer-to-peer communication in mobile computing environment. We call this method PAN2PAN communication. Since there are some paths to the Internet within a PAN in the PAN2PAN communication, traffic distribution mechanism is one of the key issues to use a real-time streaming application stably. In this paper, we describe a middleware to provide the ability of stable PAN2PAN communication using FEC (Forward Error Correction) and buffering techniques. The effect of FEC and buffering for multiple paths real-time streaming is evaluated through actual environment.

#### 1. はじめに

近年、様々なネットワークアクセス技術の出現、携帯端末の小型化・高性能化により一台の携帯端末が複数のネットワークインターフェース(N/I)を持つことが一般的になった。現在は有線、無線通信用N/Iを一つずつ保持するのが一般的だが、将来的には無線通信用のN/Iは短距離高速無線と長距離中速無線用のN/Iを持つようになる。このような環境では、短距離無線により近隣の端末間で直接通信可能なPAN(Personal Area Network)を形成しつつ、長距離無線によりインターネット上の資源を利用できる。このようにPANを形成し、PAN内の端末の通信リソースを共有しながら通信(PAN2PAN通信)すると一台の端末同士がP2Pで通信するより高速・高信頼の通信が可能になると考える。

我々は、最今注目を集めているリアルタイムストリーミングアプリケーションを移動中や外出先で快適に利用可能にするための通信制御ミドルウェアを開発している。本通信制御ミドルウェアは協調通信に同意した端末同士

でPANを形成し、PANとPANが通信を行う際にPAN内の端末の通信リソースを共有することで高速・高信頼通信を実現する。しかし、モバイル環境における複数経路リアルタイムストリーミングでは、パケット損失、各経路の遅延差・ジッタの影響で映像の乱れや音声の途切れが発生する。本通信制御ミドルウェアはFECとバッファリングを利用することで損失したパケットを復元し、さらに各経路の遅延差・ジッタの影響で遅れたパケットも早期復元することで余分な遅延を削減する。

#### 2. リアルタイムストリーミング通信

##### 2.1 リアルタイムストリーミング通信の課題

近年、光ファイバやDSL技術の進歩から広帯域ネットワークが普及し、NetMeetingやMSN Messengerといったリアルタイムストリーミングアプリケーションを利用してリアルタイムに遠隔で会議、ゼミ、会話などを行える環境が整ってきた。しかし無線環境下では、有線に比べて帯域が狭い、パケット損失が発生しやすい、リンクが切断しやすい、ジッタが大きい、ネットワーク状態が変化しやすいといった課題がある。特に同じ無線基地局へアクセスが集中、乗り物で移動中、無線電波が弱い環境などでは映像の乱れや音の途切れが発生しやすく、快適なリアルタイムストリーミング通信を行うのは難しい。このような課題を解決する方式として、一台の端末が

<sup>†</sup> 静岡大学大学院情報学研究科

Graduate School of Information, Shizuoka University

<sup>††</sup> 静岡大学情報学部

Faculty of Information, Shizuoka University

<sup>‡</sup> 株式会社 NTT ドコモ ネットワークマネジメント開発部

Network Management Development Department, NTT DoCoMo, Inc

複数の N/I を持っている、もしくは他の通信機器が持つ N/I を一時的に借りられるといった環境を利用し、複数経路通信することで信頼性や通信速度の向上を図る研究が進められている<sup>1)</sup>。以下に概要を示す。

## 2.2 関連研究

PDF システム<sup>2)</sup> や文献<sup>3)</sup> は FEC と Path Diversity を利用することでネットワークで発生するパケット損失を削減する方式を採用している。FEC(Forward Error Correction) はパケット損失が発生したときに復元するための冗長パケットをデータパケットとともに送信する。受信側でデータパケットの損失を検出した場合、冗長パケットからデータパケットを復元する。Path Diversity はホスト間に複数の経路があるとき、それぞれの経路にパケットを分配、または同一パケットを送信することで信頼性を高める。PDF システムは FEC を利用するとともに、traceroute により取得した経路情報から共有するリンクの少ない経路にパケットを分配することでパケット損失を大幅に削減できることをシミュレーションにより実験・評価している。PDF システムは traceroute からの情報を主に利用しており、各経路の帯域や遅延を考慮していないため各経路で遅延差や帯域差が大きい場合に正常に動作しないと考える。文献<sup>3)</sup> では送信者と受信者の間で複数経路がある場合、システムがオペレータに各経路のネットワーク状態を逐次報告する。オペレータはその情報から FEC 冗長符号化、負荷分散、分配配送のどれか一つを選択することでパケット損失を大幅に削減できることをネットワークエミュレータを利用して実験・評価している。このシステムはオペレータがネットワークの状態を見てどのように通信を制御するか決めるため、ネットワーク状態が頻繁に変化する環境ではオペレータが正しい通信制御を判断できないと考える。これらのシステムは有線環境や無線 LAN などの安定した無線環境下では有効であるが、実効スループットや遅延が変動しやすい、リンクが切断しやすいといったネットワーク状態の不安定なモバイル環境での利用は難しいと考える。

モバイル環境で高品質・高信頼のマルチメディア通信を行う手法として、文献<sup>4)</sup> や Multitrack<sup>5)</sup> がある。文献<sup>4)</sup> では送信者は 802.11b 無線 LAN の複数の AP からマルチメディアデータを送信し、受信側がアドホックモードを利用して複数の AP からマルチメディアデータを受信する。ビデオの圧縮方式はパケットロス耐性に強く、低遅延伝送の可能な H.264/MPEG-4 AVC を利用し、一定のビットレート、パケットサイズで圧縮している。圧縮したビデオを複数の AP を利用してストリーミングすることで一つの AP を利用してストリーミングするよりパケット損失、PSNR、遅延を改善できることを実環境で実験・評価している。無線 LAN は高速だが利用範囲が限られてしまうため、本研究では利用できる範囲が広い FOMA や PHS などの長距離無線を利用してインターネットに接続することを考える。Multitrack は同じマルチメディアコン

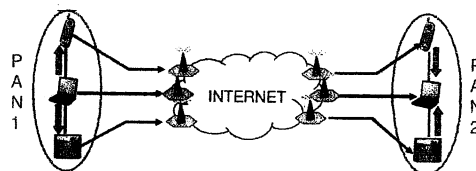


図 1 PAN2PAN 通信

テンツを視聴したい人によって PAN を形成する。そして PAN 内の端末がマルチメディアデータを分担して受信し、各端末が受信した部分データを PAN 内で交換し相互補充する。このように通信することで PAN 内の全ての人が一台で受信するよりも高品質のマルチメディアデータを視聴できる。Multitrack はインターネット上のサーバからモバイル端末への片方向通信を想定しており、モバイル端末から映像を配信することや双方向でビデオチャットすることは考慮していない。無線環境下では、一般的に帯域は下りより上りの方が狭いため、モバイル端末からデータを送信するときに帯域を増やすことも必要となる。また、双方向通信では遅延の制約があり、バッファリングは順序入れ替わりを保証可能な最低限の時間にしなければならない。Multitrack はリアルタイムビデオ伝送アプリケーションの EMON システム<sup>6)</sup> に実装し評価を行っている。ある特定のアプリケーションに特化したシステムは、そのアプリケーションの使用しているプロトコルや伝送方法の制約を受ける。また、Multitrack はそれぞれの経路間で遅延差のある場合でもネットワークの状態に応じたパケット分配を行う方式を提案しているが、パケット損失やジッタを考慮していない、パケットサイズは全て同一サイズであるといった課題がある

## 3. PAN2PAN 通信制御ミドルウェア

### 3.1 概要

本稿では、移動中や外出先で TV 電話で会議をする、スポーツ・イベントや災害現場から映像配信をするといったモバイル環境でリアルタイムに高品質・高信頼の映像や音声のやり取りを行うための通信制御ミドルウェアを提案する。本通信制御ミドルウェアは無線 LAN や Bluetooth などの短距離高速無線により協調通信に同意した端末と PAN を形成し、FOMA や PHS などの長距離中速無線を共有することで高速・高信頼のマルチメディア通信を実現する。一般的に PAN とは、数十 cm から十数 m といった非常に狭い範囲内に存在する端末やデバイス同士を相互に接続したネットワークのことをいう。本システムが想定している PAN も近隣の端末やデバイスによって形成されるが、特にあるサービスを効率的に提供したり利用したりするために協調通信に同意した端末群と一時的に形成するネットワークのことを指す。図 1 に A から X にデータを送る際の PAN2PAN 通信の例を示す。本通信制御ミドルウェアの機能によって A は近隣の B、C とい

0			7			15			31		
V	X	F	reserve	payload type	sequence number						
timestamp											
code symbol size			number of data			SN base					

図 2 拡張ヘッダ

た通信リソースを借りられる端末と提携を結び **PAN1** を形成する。同様に **X** も周囲の **Y**, **Z** と **PAN2** を形成する。**A** と端末 **X** は通信する際に、**PAN** 内の端末が持つインターネット経路も利用することで **PAN2PAN** 通信を実現する。このとき **A** のようにデータを送信する端末を送信端末、**X** のようにデータを受信する端末を受信端末、**B**, **C**, **Y**, **Z** のようにデータを中継する端末を中継端末とそれぞれ定義する。

以上のように本通信制御ミドルウェアは **PAN2PAN** 通信を実現する。さらにモバイル環境で映像や音の品質を落とす原因となるパケット損失、輻輳、遅延・ジッタを保証するための機能を持つ。モバイル環境では、ネットワーク状態が動的に変化するため通信制御ミドルウェアは各経路の状態を定期的に監視し、経路情報から予測されるネットワーク状態に応じて通信制御を行う。

### 3.2 実装レイヤ

リアルタイムストリーミングアプリケーションは様々なタイプがあり、現状ユーザは用途によって使用するアプリケーションを使い分けている。そのため提案する通信制御ミドルウェアを様々なアプリケーションで利用できるようにするため IP 層とトランスポート層の間でパケットをインターセプトする方法で実装する。リアルタイムストリーミング通信には、H.323, SIP, RTSP など様々な通信プロトコルがあるが、IP 層とトランスポート層の間でインターセプトすることにより通信プロトコルに関係なく **PAN2PAN** 通信が可能となる。通信制御ミドルウェアは iptables と Divert Socket<sup>7)</sup> を用いてパケットをインターセプトし、IP 層とトランスポート層の間で制御を加えることで IP 層とトランスポート層の間での処理を実現する。

本システムでは IP ヘッダの後ろに新たに通信制御に必要な情報を含む拡張ヘッダを追加し、通信制御ミドルウェアは拡張ヘッダに含まれる情報を参照して通信制御する。拡張ヘッダのヘッダフォーマットを図 2 に示す。拡張ヘッダは *V*, *X*, *F*, *payload type*, *sequence number*, *timestamp*, *code symbol size*, *number of data*, *SN base* で構成される。*V* は通信制御ミドルウェアのバージョンを表す。*X* は拡張ビットを表し、通信制御ミドルウェアを拡張するためのビットである。*F* は FEC ビットであり、FEC を利用していることを表す。*payload type* はペイロードの情報が何かを示し、パケットが映像、音声、制御パケットかを判断するために使われる。*sequence number* はパケットの順番を表し、順序制御に使われる。*timestamp* は送信側のパケット送出時刻を表す。*code symbol size* は Reed-Solomon 符号のコードシンボルサイズを表し、*number of data* は

route ID	sequence number
throughput	packet loss rate
timestamp	

図 3 測定用パケット

Reed-Solomon 符号のデータシンボル数を表す。*SN base* は Reed-Solomon 符号の 1 ブロック内の最小のシーケンス番号を格納する。コードシンボルサイズ、データシンボル数、SN ベースは後述する Reed-Solomon FEC に使用される。

### 3.3 PAN 形成・PAN 情報の交換

通信制御ミドルウェアは **PAN2PAN** 通信を行う際に自動的に **PAN** を形成する。**PAN** は送信端末と受信端末が形成、管理する。この端末を **PAN Leader(PL)** と定義し、また **PL** に通信リソースを貸す端末を **PAN Member(PM)** と定義する。図 1 では、送信端末 **A**, 受信端末 **X** が **PL**, 中継端末 **B**, **C**, **Y**, **Z** が **PM** となる。**A** は、端末 ID, 上り帯域, 下り帯域, 短距離無線リンクの IP アドレス, 長距離無線リンクの IP アドレスを含む自分の端末情報から端末情報登録リクエストを作成し、**B**, **C** にブロードキャストする。**A** からブロードキャストされた端末情報登録リクエストを受信した **B**, **C** は自分の端末情報から端末情報登録レスポンスを作成し、**A** に返信する。**B**, **C** から返信された端末情報登録レスポンスを受信した **A** は **B**, **C** の端末情報を **PAN** 情報リストに格納する。**A** はこのように **B**, **C** と **PAN1** を形成する。**PAN** 情報リストは **PAN** 内の端末の端末情報を保持するリストである。この端末情報のやり取りは定期的に行われ、送信端末の **PAN** 情報リストは随時更新される。同様に **X** は **Y**, **Z** と **PAN2** を形成する。

**A** は **PAN1** を形成すると、通信相手 **X** と **PAN** 情報の交換を行う。**A** は、**PAN ID**, **PAN** 内の端末のそれぞれの端末情報を含む **PAN1** の **PAN** 情報から **PAN** 情報登録リクエストを作成し、**X** に送信する。**X** は **A** から送信された **PAN** 情報登録リクエストから **PAN1** の **PAN** 情報を **PAN** 情報リストに登録し、**PAN2** の **PAN** 情報から **PAN** 情報登録レスポンスを作成し、**A** に返信する。**X** から返信された **PAN** 情報登録レスポンスを受信した **A** は **PAN2** の **PAN** 情報を **PAN** 情報リストに登録する。このように **A** は **PAN** 情報の交換を行うことで **PAN2** の情報を取得することができる。この **PAN** 情報の交換は定期的に行われ、受信端末の **PAN** 情報リストは随時更新される。

### 3.4 経路情報測定

通信制御ミドルウェアは、モバイル環境における実効スループット、遅延、パケット損失の動的な変化に対応するため、各経路の状態を定期的に監視する。送信端末は 1 秒毎に測定用パケットを各経路に送信することで各経路の受信スループット、パケット損失、遅延を測定する。測定用のパケットフォーマットを図 3 に示す。測定用のパケットは *route ID*, *sequence number*, *throughput*, *packet loss rate*, *timestamp* フィールドにより構成される。

経路	中継する端末	パケット
1	A→X	X
2	A→B→X	B X
3	A→C→X	C X
4	A→Y→X	Y X
5	A→Z→X	Z X
6	A→B→Y→X	B Y X
7	A→B→Z→X	B Z X
8	A→C→Y→X	C Y X
9	A→C→Z→X	C Z X

図4 各経路へ分配されるパケットの例

route ID は経路の識別子を格納する。sequence number はパケットの順番を格納する。timestamp はパケットの送信時刻を格納する。throughput と packet loss rate は 0 を格納する。送信端末は、このように作成した測定用パケットを 1 秒毎に受信端末へ送信する。受信端末は 10 秒毎にパケット損失率と受信スループットを測定している。受信スループット throughput は 10 秒間の受信データ量から算出する。recvsize は 10 秒間に受信したデータ量である (式 (1))。

$$\text{throughput} = \frac{\text{recvsize}}{10} \quad (1)$$

次にパケット損失率 packetlossrate は 10 秒間に受信したパケット数と損失したパケット数から算出する。packetnum は 10 秒間に受信したパケット数、packetloss は 10 秒間に損失したパケット数である。(式 (2))。

$$\text{packetlossrate} = \frac{\text{packetloss}}{\text{packetnum} + \text{packetloss}} \quad (2)$$

受信端末は測定用パケットを受信すると、測定した受信スループットを throughput に、パケット損失率を packet loss rate に格納する。他の情報には手を加えず、送信端末へ返信する。送信端末は受信端末から返信された測定用パケットの受信時刻と測定用パケットの timestamp に格納されている送信時刻との差を求め RTT を取得する。また、送信端末は受信端末から返信された測定用パケットの bandwidth と packetlossrate から受信端末で測定された受信スループットとパケット損失率を取得する。送信端末は、測定した受信スループット、パケット損失、各経路の遅延を後述するパケット分配、FEC、バッファリングの処理に利用することで動的なネットワークの変化に応じた通信制御を行う。

### 3.5 パケット分配

通信制御ミドルウェアはパケット分配・中継をすることで複数経路通信を実現し、帯域集約する。図1のAとXがPAN2PAN通信をする場合を考える。AはXにパケットを送信するとき、図4に示す計9本の経路を使用できる。AはIPカプセルングすることで各経路にパケットを分配する。図4に各経路のカプセルングによりAで作成されるパケットの例を示す。Aが送信したパケットはB、

C、Y、Zがデカプセルングを行うことで中継されXに送信される。例えばA→C→Z→Xの経路では、X宛のパケットをZ宛のIPヘッダでカプセルングし、さらにそのパケットをC宛のIPヘッダでカプセルングする。このパケットはIPヘッダがC宛であるためCに送信され、Cがデカプセルングする。Cでデカプセルングされたパケットは宛先がZであるためZに送信され、Zがデカプセルングする。Zでデカプセルングされたパケットは宛先がXであるためXに送信される。このようにパケットをA→C→Z→Xの経路で配送する。

### 3.6 バッファリング・順序制御

一般的にリアルタイムストリーミング通信では、RTP(Real Time Protocol)によってアプリケーション層でバッファリング・順序制御が行われるため、バッファリングの長さはアプリケーションによって異なる。PAN2PAN通信では、複数経路通信時に順序入れ替わりが頻繁に発生するためアプリケーションによっては正しく順序制御を行えない場合が発生する。通信制御ミドルウェアは受信したパケットをそのままアプリケーションに渡すのではなく、ある程度バッファリングし正しく順序を並び替えた後にアプリケーションへ渡す。

### 3.7 FEC

モバイル環境ではパケット損失が発生しやすいため、リアルタイムストリーミング通信の際に映像の乱れや音の途切れが発生しやすい。また、リアルタイムストリーミングアプリケーションはパケット損失を保証していないことが多い。そのため通信制御ミドルウェアはFECを利用することで損失したパケットを復元する。FECはパケット損失が発生したときに復元するための冗長パケットをデータパケットとともに送信し、受信側でデータパケットの損失を検出した場合冗長パケットからデータパケットを復元する。さらに各経路間の遅延差やジッタの影響でバッファリング時間内に到着しないパケットも復元することで余分な遅延を減らすことができる。

## 4. プロトタイプ実装

PAN形成・PAN情報交換、経路情報測定、パケット分配、FEC、バッファリングの機能を持つプロトタイプを実装した。パケット分配は各経路の上りと下り帯域、遅延を考慮した方式を実装した<sup>9)</sup>。バッファリングとFECはパケット損失、各経路の遅延差の遅延差・ジッタの影響を軽減する機能を実装した。以下に今回実装したバッファリングとFECの実装の詳細を示す。

### 4.1 バッファリングの実装

PAN2PAN通信では、複数経路通信時に順序入れ替わりが頻繁に発生するため、受信したパケットをすぐにアプリケーションに渡すのではなく、ある程度バッファリングし正しく順序を並び替えた後にアプリケーションへ渡す。図5にバッファリングの動作を示す。通信制御ミドルウェアは受信処理で入力バッファと出力バッファを

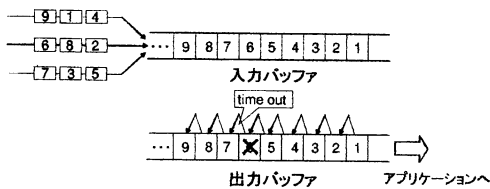


図5 バッファリングの動作

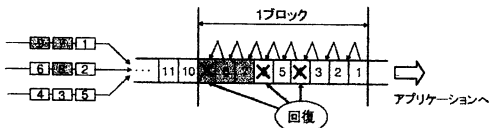


図6 FECの動作

持つ。入力バッファと出力バッファは配列構造になっており、配列番号がパケットのシーケンス番号に対応する。受信端末は送信端末から各経路に分配されたパケットを受信すると、拡張ヘッダに格納されたシーケンス番号に対応して入力バッファにパケットの情報を格納する。入力バッファにパケットの情報を格納すると、この情報から拡張ヘッダを取り除き、アプリケーションに渡すパケット情報にして出力バッファに格納する。出力バッファのパケット情報はシーケンス番号の順番にアプリケーションに渡すことで正しい順序に並び替えることができる。通信制御ミドルウェアはアプリケーションにパケットを渡すとシーケンス番号をインクリメントし、次のパケットの処理に移る。しかしパケット損失や各経路の遅延差・ジッタの影響でシーケンス番号のパケットが来ない、または大きく遅れることがある。そのため通信制御ミドルウェアはタイムアウト時間を設定し、その時間内にパケットが到着しないときはシーケンス番号をインクリメントし、次のパケットの処理に移る。双方向通信で違和感なく会話をするため多くても遅延を400ミリ秒以下にする必要がある<sup>9)</sup>ため、今回のプロトタイプではタイムアウト時間は400ミリ秒とした。

#### 4.2 FECの実装

FECの使い方は損失したパケットを復元することが一般的だが、通信制御ミドルウェアはFECを損失パケットの復元だけでなく、複数経路通信時に発生する各経路の遅延差・ジッタの影響を軽減するためにも利用する。FECの冗長符号は実ネットワークでパケット損失に対する効果が実証されているReed-Solomon符号を用いる<sup>10)</sup>。Reed-Solomon符号は、ブロック符号の1つであり、送信すべき情報を連続した複数のビット単位(シンボル)で分割し、符号化、復号化を行う誤り訂正符号である。1ブロックは $N$ 個のシンボルによって構成され、そのうち $K$ 個がデータシンボルであり、 $N - K$ 個が冗長シンボルである場合、 $(N, K)$ の冗長度と呼ぶ。図6にFECの動作を示す。通信制御ミドルウェアはバッファリングの処理で出力バッファのアプリケーションに渡すべきパケットが未到着で、

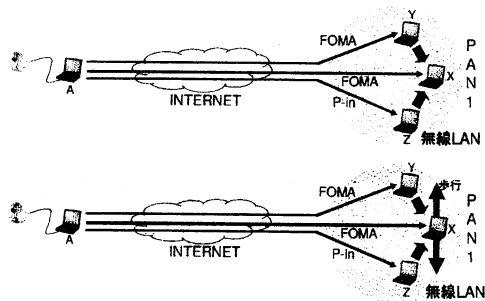


図7 実験環境

FEC処理に必要なパケット $N$ 個のうち $N - K$ 個到着していたらタイムアウト時間を待たずにFEC処理を行いパケットを復元する。このようにFEC処理に必要な数のパケットを受信したらタイムアウト時間を待たずにパケットを復元することで、損失したパケットの復元だけでなく、各経路の遅延差やジッタの影響によって生じる遅延を削減しアプリケーションに渡すことができる。

## 5. 評価

### 5.1 実験環境

本通信制御ミドルウェアのバッファリングとFECの機能を評価するため、FOMA、P-in、無線LANを使用して実験を行った。図7に実験環境を示す。Aはストレートケーブルでインターネットにつながれている。一方X、YはFOMAのデータ通信、ZはP-inでインターネットに接続する。FOMAのデータ通信のスループットは約50kbpsで、RTT(Round Trip Time)は500~600msecであった。P-inのスループットは約40kbpsで、RTTは200~300msecであった。AにUSBカメラを接続し、Linux用のTV会議システムであるGnomeMeetingでXに映像を流す。AはPANを形成していないが、通信制御ミドルウェアを起動することでPANを一台で形成しているように仮想し、この間の通信はPAN2PAN通信とみなす。X、Y、Zは無線LANのアドホックモードを利用してPAN1を形成し、XはAからマルチメディアデータをPAN内の他端末と分担して受信する。この環境でPAN内の端末が静止しているときと一台の端末が移動しているときのバッファリングとFECの効果を評価した。

### 5.2 実験1-バッファリングとFECの効果:静止状態

本実験では、PAN内の端末が静止した状態でFECとバッファリングを利用したときパケット損失の復元と遅延の削減の効果を評価した。FECの冗長度は(6,4)、(8,6)、(9,6)、(12,9)で実験した。(6,4)、(9,6)の冗長度ではトラフィック増加率は約1.5倍で、(8,6)、(12,9)の冗長度ではトラフィック増加率は約1.3倍である。この環境でPAN2PAN通信でXからAへ最大ビットレートが100kbpsの映像を5000パケット流した。パケットはラウンドロビンでそれぞれの端末に分配した。この実験でパケット損失数、FEC

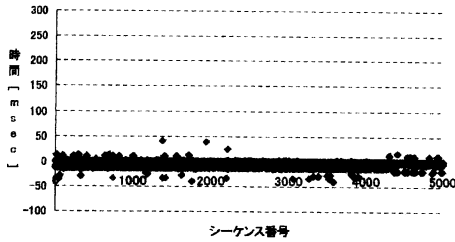


図 8 冗長度 (6.4) における遅延削減時間：移動なし

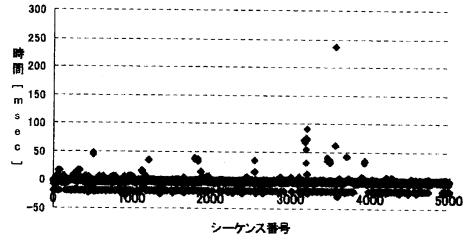


図 10 冗長度 (9.6) における遅延削減時間：移動なし

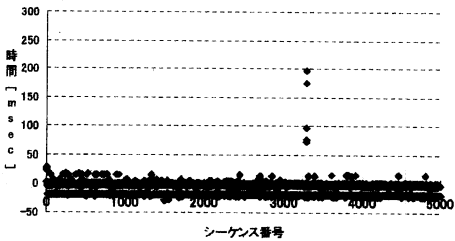


図 9 冗長度 (8.6) における遅延削減時間：移動なし

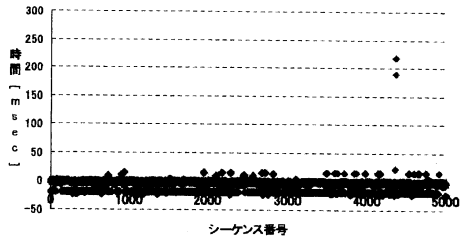


図 11 冗長度 (12.9) における遅延削減時間：移動なし

表 1 パケット損失に対する FEC の効果：移動なし

	(6,4)	(8,6)	(9,6)	(12,9)
パケット損失数 [個]	0	2	1	0
パケット復元数 [個]	0	1	1	0
パケット復元率 [%]	*	50	100	*

によるパケット復元数、パケット復元率を比較した。また、各冗長度における FEC により削減できた遅延時間を比較した。遅延削減時間は FEC なしの場合のアプリケーションへの到着時刻と FEC ありの場合のアプリケーションへの到着時刻の差で求めた。

表 1 に各冗長度におけるパケット損失数、パケット復元数、パケット復元率を示す。また各冗長度における FEC により削減できた遅延時間を図 8、図 9、図 10、図 11 にそれぞれ示し、表 2 に 100msec 毎の削減できた遅延の回数を示す。表 1 に示すように PAN1 の端末が静止した状態ではパケット損失はほとんど起こらず、発生しても 5000 パケットのうち 1、2 個程度であった。(9,6) の冗長度のとき 2 個のパケットが損失したが、1 個のパケットしか復元できなかった。図 8、図 9、図 10、図 11、表 2 に示すように FEC により大きく遅延を削減できたパケットはほとんどなく、100msec 以上削減できたパケットは 5000 パケットのうち 1、2 個程度であった。静止状態では FEC を利用する程通信品質が劣悪でなく、パケット損失復元と遅延削減に対する FEC の効果があまりなかったと考える。バッファリングで順序を並び替えるだけで各経路の遅延差・ジッタの影響を吸収できると考える。

### 5.3 実験 2-バッファリングと FEC の効果：移動状態

本実験では、PAN 内の端末の一台が移動した状態で FEC とバッファリングを利用したときパケット損失の復

表 2 各冗長度における遅延：移動なし

	(6,4)	(8,6)	(9,6)	(12,9)
300-200	0	0	1	1
200-100	0	2	0	1
100-0	517	304	285	417

元と遅延の削減の効果を評価した。X は無線 LAN の通信可能範囲で歩くスピード移動させた。FEC の冗長度は実験 1 と同様に (6,4)、(8,6)、(9,6)、(12,9) で実験した。この環境で PAN2PAN 通信で X から A へ最大ビットレートが 100kbps の映像を 5000 パケット流した。パケットはラウンドロビンでそれぞれの端末に分配した。この実験でパケット損失数、FEC によるパケット復元数、パケット復元率を比較した。また、各冗長度における FEC により削減できた遅延時間を比較した。

表 5 に各冗長度におけるパケット損失数、パケット復元数、パケット復元率を示す。また各冗長度における FEC により削減できた遅延時間を図 12、図 13、図 14、図 15 に示し、表 4 に 100msec 毎の削減できた遅延の回数を示す。表 5 に示すように静止状態とは違って X が移動した状態ではパケット損失が発生し、それぞれの冗長度でパケット損失を復元できた。表 5 に示すように各冗長度では、トラフィック増加率が 1.5 倍である (6,4) と (9,6) の冗長度を比べると、(6,4) の冗長度のときの方がパケットを復元できた。またトラフィック増加率が 1.3 倍である (8,6) と (12,9) の冗長度を比べると、(8,6) の冗長度のときの方がパケットを復元できた。(6,4)、(8,6)、(9,6)、(12,9) の冗長度ではデータシンボルの数はそれぞれ 4、6、6、9 である。データシンボル数が少ない方がパケット損失率を効果的に復元できた。図 12、図 13、図 14、図 15、表

表 3 パケット損失に対する FEC の効果：移動あり

	(6,4)	(8,6)	(9,6)	(12,9)
パケット損失数	75	113	74	56
パケット復元数	52	59	33	23
パケット復元率	69.3	52.2	44.6	41.1

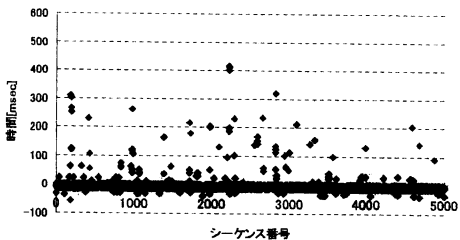


図 12 冗長度 (6.4) における遅延削減時間：移動あり

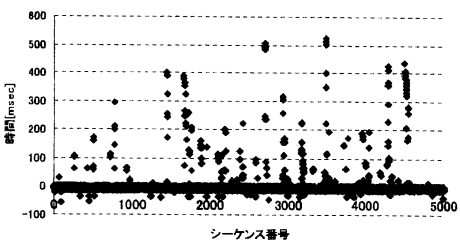


図 13 冗長度 (8.6) における遅延削減時間：移動あり

表 4 各冗長度における遅延：移動あり

	(6,4)	(8,6)	(9,6)	(12,9)
600-500	0	5	0	2
500-400	3	13	1	0
400-300	3	27	5	2
300-200	12	30	20	16
200-100	28	56	17	5
100-0	566	693	607	695

4 に示すようにどの冗長度でも遅延を大きく減らすことができた。(6,4), (8,6), (9,6), (12,9) の冗長度を比べると (8,6) の冗長度のときに遅延を大きく減らすことができた。これは表 5 に示すように (8,6) の冗長度のときに行った実験の時に通信品質が悪かったが、バッファリングと FEC の処理が効果的に作用し、品質が大幅に改善できたためと考える。その他の冗長度を比べると効果に大きく差はないが、データシンボル数が少ない方が効果的に遅延を減らすことができた。X が移動した状態では、PAN 内の端末が静止している状態より通信品質が悪化しパケット損失、遅延、ジッタの影響が大きくなった。しかし、バッファリングと FEC を利用することでパケット損失、遅延ともに削減することができた。

#### 5.4 考 察

実験から FEC によるパケット損失復元と遅延削減する手法は PAN 内の端末が静止している環境ではあまり効果的がなく、PAN 内の端末が移動する環境では効果的であった。FEC を利用するとトラフィックを冗長パケ

表 5 パケット損失に対する FEC の効果：移動あり

	(6,4)	(8,6)	(9,6)	(12,9)
パケット損失数	75	113	74	56
パケット復元数	52	59	33	23
パケット復元率	69.3	52.2	44.6	41.1

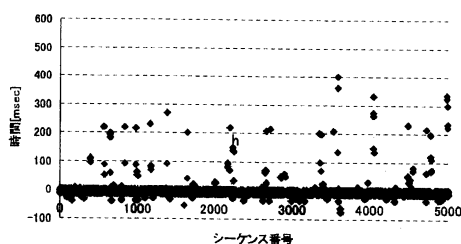


図 14 冗長度 (9.6) における遅延削減時間：移動あり

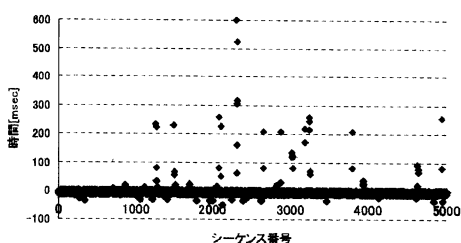


図 15 冗長度 (12.9) における遅延削減時間：移動あり

トの分だけ増加させてしまう。静止している状態で通信品質が良い環境ではパケット分配による帯域集約とバッファリングによる順序制御を行うだけで、FEC は移動などにより通信品質が悪くなったときのみ利用することが望ましいと考える。各冗長度でパケット損失を復元できたが、復元率は 70~40% で復元できないパケットもある程度あった。また各冗長度を比べると、FEC のデータシンボルの数が少ない方がパケット損失復元と遅延削減において効果的であった。表 6 に復元できなかったパケットの FEC ブロック内の位置を示す。復元できないパケットは FEC ブロックの始めに集中しており、後ろに行くほど復元できないパケット数は減っている。今回映像の伝送に使用した GnomeMeeting では、h.261 で映像を圧縮し、送信する。100kbps の映像を伝送したとき、パケットの送信間隔は平均 100 ミリ秒であった。N をデータシンボル数、K をがデータシンボル数とすると Reed Solomon FEC は  $N - K$  個のパケットが到着するまで、パケットを復元することができない。そのため Reed Solomon FEC のブロックの始めの位置のパケットが損失すると、他のブロック内のパケットが到着するまでの時間がタイムアウト時間である 400 ミリ秒を超えてしまい、パケット損失とみなしてシーケンス番号を更新してしまうと考える。データシンボル数が小さければ、到着するまで待つパケット数が少ないためパケット復元率が高かったと考える。N

表 6 復元できなかったパケットの FEC ブロック内の位置

	(6,4)	(8,6)	(9,6)	(12,9)
1 番目	23	18	13	7
2 番目	0	14	1	0
3 番目	0	10	20	8
4 番目	0	5	5	8
5 番目	*	1	2	0
6 番目	*	2	0	5
7 番目	*	*	*	4
8 番目	*	*	*	0
9 番目	*	*	*	0

が 6 である (8,6), (9,6) の冗長度を考えると、例えば 1 番目のパケットが損失し、パケットの送信間隔が 100 ミリ秒であった場合、7 番目の冗長パケットが到着するまで 600 ミリ秒待たなければいけなくタイムアウトを起こしてしまう。今回の環境では、タイムアウト時間が 400 ミリ秒の環境において (6,4) の冗長度が一番パケット損失を復元できた。パケット復元率を上げるためにはタイムアウト時間を増やす、データシンボル数といった方法が考えられる。タイムアウト時間を増やせば遅延が大きくなってしまい、データシンボル数を減らすと FEC の効率が落ちてしまうため状況に応じて設定すべきである。適切なタイムアウト時間でデータシンボル数はパケット送信間隔に依存し、他の映像圧縮方式や音声伝送などで送信間隔がもっと短い場合では、タイムアウト時間を短くする、データシンボル数を増やすということをしてパケット損失を復元できると考える。

## 6. まとめと今後の課題

PAN2PAN リアルタイムストリーミングにおける通信制御機構を検討し、そのプロトタイプを実装し実験によりバッファリングと FEC の効果を評価した。その結果、静止状態では FEC は効果がほとんどなく、バッファリングだけで各経路の遅延差・ジッタの影響を吸収できると考える。移動状態ではバッファリングとともに FEC を利用することで各経路の遅延差・ジッタの影響を吸収できると考える。また FEC の冗長度はブロック数が少ない方が FEC 処理が短い周期で実行されるため、早くパケットが復元されると考える。

今後、帯域が非対称性である環境で効果的に帯域を節約しつつ、バッファリングと FEC によりパケット損失と遅延の削減できる手法を検討する。また、今回の実験は PAN 内の 1 台の端末歩くスピードで移動する環境で実験を行ったが、今後は車や電車などで PAN が高速で移動する環境でも実験、評価を行う。

## 参 考 文 献

- 1) John G. Apostolopoulos and Mitchell D. Trott, Streaming Media Systems Group, HP Labs: "Path Diversity for Enhanced Media Streaming", IEEE Communications Magazine, pp. 80-87, (August 2004).
- 2) T. Nguyen and A. Zakhor, "Path Diversity with Forward Error Correction(PDF) System for Packet Switched Networks", INFOCOM 2003, pp.663-672.
- 3) 川西 智也, 中村 素典, 岡部 寿男, 美濃 導彦, "FEC

と Path-Diversity を利用した回復可能なストリーミング", 2004-DPS-116, pp.79-84.

- 4) A. Miu, J. Apostolopoulos, W. Tan, and M. Trott, "Low-latency wireless video over 802.11 networks using path diversity", ICME 2003, pp.441-444.
- 5) Y. Saito, S. Ishihara, T. Mizuno and T. Watanabe, "Evaluation of Traffic Dispersion Methods for Synchronous Distributed Multimedia Data Transmission on Multiple Links", AINA 2003, pp.644-649.
- 6) 笠松健一, 藤川賢治, 岡部寿男, 古村隆明, "EMON システム: コマンドパイプラインによるマルチメディアストリーム処理", 情処大全 2002, pp.485-486.
- 7) Divert Socket for Linux.  
<http://sourceforge.net/projects/ipdivert>.
- 8) 鈴木宏治, 峰野博史, 田中希世子, 鈴木偉元, 石川憲洋, 水野忠則, "PAN2PAN リアルタイムストリーミング通信における制御機構の開発", 2005-DPS-122, pp.265-270.
- 9) ITU-T Rec. G.114 "One-way transmission time", May 2002.
- 10) 近堂徹, 大塚玉記, 西村浩二, 相原玲二, "MPEG over IP 伝送システム mpeg2ts の開発と性能評価", DI-COMO 2002, pp.157-160.