

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

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^{††}

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、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験により評価した。

A development of communication control method for PAN2PAN realtime streaming

KOJI SUZUKI,[†] HIROSHI MINENO,^{††} KIYOKO TANAKA,[‡] HIDEHARU SUZUKI,[‡]
ISHIKAWA NORIHIRO[‡] and TADANORI MIZUNO^{††}

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 technics. The effect of PAN2PAN communication is evaluated through prototype testbed.

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。さらに複数経路通信で重要となる順序制御やパケット損失回復といった機能を提供することでリアルタイムストリーミングアプリケーションに対応する。

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。

このような問題を解決する方式として、一台の端末が複数の N/I を持っている、もしくは他の端末が持つ 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。以下に概要を示す。

2.2 関連研究

PDF システム¹⁾や文献²⁾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)

[†] 静岡大学大学院情報学研究所

Graduate School of Information, Shizuoka University

^{††} 静岡大学情報学部

Faculty of Information, Shizuoka University

[‡] 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

はパケット損失が発生したときに回復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。受信側でパケット損失があった場合、冗長パケットからデータパケットを回復する。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²⁾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経路ごと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、帯域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が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、Multitrack³⁾がある。Multitrack は同じマルチメディアコン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 内で交換し相互補完する。このように通信することで PAN 内の全ての人が一台中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⁴⁾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帯域差や遅延差が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

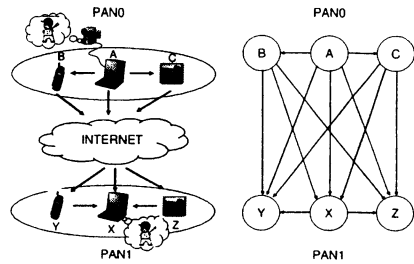


図1 PAN2PAN 通信

トサイズは全て同一サイズであるといった問題がある。

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によって無線 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受け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 に端末 A と端末 X が通信する際の 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端末 A は近隣の端末 B、端末 C といった通信リソースを借りられる端末と提携を結び PAN0 を形成する。同様に端末 X も周囲の端末 Y、端末 Z と PAN1 を形成する。端末 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。端末 A は端末 X にパケットを送信するとき、図 1 右に示す計 9 本の経路を使用できる。実際の通信で利用する経路は本通信制御ミドルウェアによって制御される。

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、帯域不足、遅延・ジッタ、リンク切断を保証するための機能を述べる。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。

パケット損失への対応

パケット損失が発生しやすい環境では、FEC を利用することで遅延を小さくしたままパケット損失を減らす。FEC

は冗長パケットにより帯域が増加するため帯域にある程度余裕がありパケット損失がランダムに発生する場合に利用すると効果大きい、また、パケット損失が FEC だけで回復できないほど大きい場合は同一パケットをコピーして複数の経路へ送信する。

帯域不足への対応

現在のリアルタイムストリーミングアプリケーションは輻輳制御を行わないものがほとんどで、通信経路の帯域が不足している場合でもデータを送信し続けてしまう。そのため送信経路で輻輳が発生し、パケット損失や遅延の増大によって映像や音声の品質が劣化する。自分自身の経路の帯域だけでデータを送信できない場合、PAN 内の端末が持つ複数経路にパケットを分配してトラフィックを分散することで輻輳を回避する。

遅延・ジッタへの対応

リアルタイムストリーミング通信では遅延の制約があり、遅延が大きい通信では会話がスムーズにできない。各経路で帯域に余裕がある場合、最も遅延の小さい経路を使用して通信することで低遅延の通信が実現できる。また、帯域に余裕がある場合 FEC を利用することでジッタや各経路の遅延差が原因でバッファリング時間内に届かなかったパケットを復元でき、低遅延の通信が可能となる。

リンク切断への対応

リンクが切断されやすい環境では、同一パケットをコピーして複数の経路へ送信する。ある経路が切断されても他の経路でデータを送信できる可能性が高まるため持続性を高めることができる。

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 Socket⁵⁾ 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。

3.3 提供する機能

3.3.1 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 は実際に通信を行う端末が形成、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。PL は自分の端末情報を含む端末情報登録リ

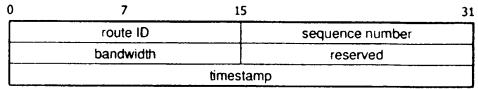


図 2 測定用パケット

クエストを周囲の端末にブロードキャストする。PL 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 PM は、PL の通信に協調する場合自分の端末情報から端末情報登録レスポンスを作成し、PL に返信する。PM から返信された端末情報登録レスポンスを受信した PL は PM の端末情報を PAN 情報リストに格納する。PAN 情報リストは PAN 内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 3 秒毎に実行され、PL の保持する自 PAN 情報リストは 3 秒周期で更新される。

通信制御ミドルウェアは PAN が形成されると、通信相手の PL と PAN 情報の交換を行う。自分が形成している PAN の PL を PL1、通信相手が形成している PAN の PL を PL2 とする。PL1 は PAN 情報を含む PAN 情報登録リクエストを PL2 に送信する。PL2 は自身で PAN を形成している場合 PAN 情報登録レスポンスを返信する。PL2 から返信された PAN 情報登録レスポンスを受信した PL1 は PL2 の PAN 情報を PAN 情報リストに登録する。このように PAN 情報の交換が行うことにより通信相手の PAN の情報を取得することできる。この PAN 情報の交換は 5 秒毎に実行され、PL1 の保持する通信相手 PAN 情報リストは 5 秒周期で更新される。

3.3.2 経路情報測定

モバイル環境では、帯域や遅延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する必要がある。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帯域、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 2 に示す。測定用のパケットは route ID、sequence number、bandwidth、timestamp フィールドにより構成される。route ID は経路の識別子を表す。sequence number はパケットの順番を表す。timestamp はパケットの送信時刻を表す。bandwidth は送信端末では 0 となる。このように作成した測定用パケット 2 つを連続して送信し、受信側で 2 つのパケットの到着時刻の差からパケットペア方式⁶⁾を用いて利用可能な帯域を算出する(式(1))。

$$bandwidth = \frac{packetsize}{T_2 - T_1} \quad (1)$$

受信側では、測定用パケットの bandwidth に算出した利用可能な帯域を格納し、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、測定用パケットの sequence number からパケット損失率を求める。通信制御ミドルウェアは、測定した帯域、パケット損失、遅延を利用することで効率良く通信制御を行う。

経路	パケット
A→X	[X]
A→B→X	[B X]
A→C→X	[C X]
A→Y→X	[Y X]
A→Z→X	[Z X]
A→B→Y→X	[B Y X]
A→B→Z→X	[B Z X]
A→C→Y→X	[C Y X]
A→C→Z→X	[C Z X]

図3 各経路のパケットフォーマット

3.3.3 複数経路通信

通信制御ミドルウェアはパケット分配・中継することで複数経路通信を実現し、複数経路にパケットを分配することで帯域を論理的に増やす、同一のパケットを送信して信頼性を高めるといったことができる。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1の右側に示す計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できる。図3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。

3.4 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。バッファリング時間は各経路間の遅延差やジッタを吸収するため、各経路で一番大きい遅延に100ミリ秒を足した時間とする。ただし双方向通信では、違和感なく会話をするため最低でも遅延を400ミリ秒以下にする必要がある⁷⁾。そのためバッファリング時間は最大で400ミリ秒とし、遅延の大きい経路は使用しない。

3.4.1 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を利用してパケット損失を回復させる。FECの冗長符号

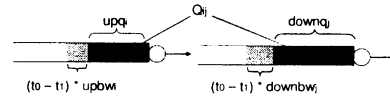


図4 仮想キュー

は実ネットワークで効果を実証されている Reed-Solomon 符号を用いる⁸⁾。また、ジッタの影響で遅延が大きくなったパケットがあつた場合でも、FECを利用してパケットを復元し余分な遅延を減らすことができる。

4. プロトタイプ開発

4.1 パケット分配方式

PAN形成・PAN情報交換、経路情報測定、複数経路通信、バッファリング・順序制御の機能を持つプロトタイプを実装した。複数経路通信機能の実装では、各経路の帯域を効率よく利用する以下の3つのパケット分配方式を実装した。一般的に長距離無線は上りと下りの帯域が非対称であるため、各経路の上りと下り帯域を考慮したパケット分配方式となっている。

方式1

送信端末はまず自分が形成しているPAN内の端末の各上り帯域の比率で自PAN内のどの端末にパケットを中継させるか決定する。次に通信相手が形成しているPAN内の端末の各下り帯域の比率で通信相手のPAN内のどの端末にパケットを中継させるか決定する。このようにどの端末に中継させるかを決定し、その経路にパケットを送信することで各経路の上りと下りの比率でパケット分配を行うことができる。

方式2

送信端末は各経路ごとに図4に示すような仮想キューを持つとする。仮想キューは送信側のPAN内の端末が持つ上りキューと受信側のPAN内の端末が持つ下りキューの組み合わせで表す。 Q_{ij} を仮想キューに貯まっているデータ量、 upq_i と $downq_j$ をそれぞれ端末*i*の上りキューと端末*j*の下りキューに貯まっているデータ量とする。 Q_{ij} は upq_i と $downq_j$ を足した量となり、送信端末は Q_{ij} が最小の経路にデータを送信する(式(2))。

$$Q_{ij} = upq_i + downq_j \quad (2)$$

upq_i , $downq_j$ はパケット送信前と送信後に随時更新する。 $upbw_i$, $downbw_j$ を端末*i*の上りの帯域と端末*j*の下り帯域、 t_0 を次のパケットを送信する時刻、 t_1 を1つ前のパケットを送信した時刻すると、 $(t_0 - t_1) * upbw_i$ と $(t_0 - t_1) * downbw_j$ は1つ前のパケットを送信した後から次のパケットを送信するまでの仮想キュー内のデータ減少量となる。送信パケットの送信先経路を決定する際、 upq_i と $downq_j$ からそれぞれ $(t_0 - t_1) * upbw_i$, $(t_0 - t_1) * downbw_j$ を引き、最小 Q_{ij} となる経路へパケットを送信する。パケット送信後は、パケットを送信した経路の上りキューと下りキューへ送信したパケットのサイズを足し、 Q_{ij} を更新する。

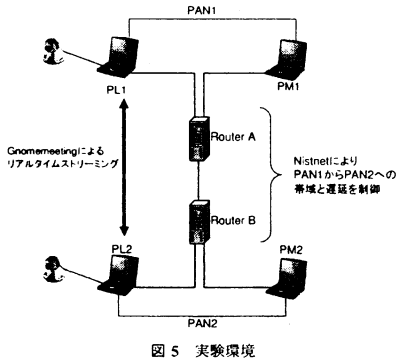


図5 実験環境

方式3

方式3は方式2で提案した仮想キュー内に貯まっているデータを送信し終える時間と経路の伝送遅延 $delay_{ij}$ を用いて、送信するパケットが受信端末に到着する時刻を予測し、到着時刻が最小の経路にパケットを分配する方式である。 $delay_{ij}$ は測定用パケットで測定したRTTである。各経路のパケット到着予測時刻を D_{ij} とする。上りキューと下りキューに貯まっているデータを送信し終えるまでの時間は、それぞれ $upq_i/upbw_i$, $downq_j/downbw_j$ で求まる。 D_{ij} は $upq_i/upbw_i$, $downq_j/downbw_j$, $delay_{ij}$ を足した時間となり、送信端末は D_{ij} が最小となる経路へパケットを送信する(式(3))。

$$D_{ij} = upq_i/upbw_i + downq_j/downbw_j + delay_{ij} \quad (3)$$

upq_i , $downq_j$ は方式2と同様にパケット送信前後で更新され、 D_{ij} へ反映される。

4.2 予備実験

4.2.1 実験環境

本ミドルウェアのプロトタイプと提案したパケット分配方式の評価を行うため、ネットワークエミュレータ NIST Net を利用して実験環境に近い仮想的なネットワークを構築して実験を行った。図5に実験環境を示す。通信制御ミドルウェアはPL1とPL2で送受信モードとして動作させ、PM1とPM2で中継モードとして動作させる。プロトタイプでは通信制御ミドルウェアは送受信モードと中継モードで区別して動作させているが、今後の実装では通信制御ミドルウェアが通信時に自動的にモードを判断して動作できるようにする。PL1とPM1はクロスケーブルで直接つながれており、PAN1を形成する。同様にPL2とPM2はPAN2を形成する。PLとPMは実際の環境では、短距離高速無線で通信されるが、この間の通信は帯域が十分にあり遅延も十分に小さいと考え有線で模擬している。一方、PAN1からPAN2への通信は長距離中速無線での通信を想定し、ルータ1とルータ2でNIST Netにより長距離無線環境に近い帯域と遅延で設定する。ルータ1のNIST NetでPL1とPM1の上り帯域と遅延、ルータ2のNIST Netで下り帯域で設定する。今回実装したプロトタイプはパケット損失を保証する機能をまだ実装していないため、ランダムに発生するパケット損失

表1 経路表

経路	経路
経路1	PL1 → PL2
経路2	PL1 → PM2 → PL2
経路3	PL1 → PM1 → PL
経路4	PL1 → PM1 → PM2 → PL2

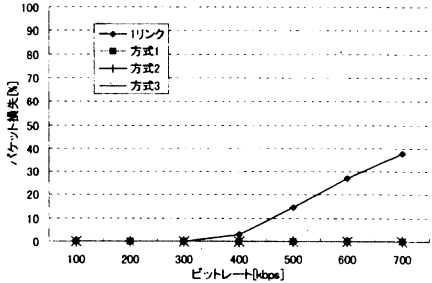


図6 実験1における各通信のパケット損失率

表2 実験1における各経路の帯域と遅延の設定

	経路1	経路2	経路3	経路4
上り帯域	384kbps	384kbps	384kbps	384kbps
下り帯域	384kbps	384kbps	384kbps	384kbps
遅延	200msec	200msec	200msec	200msec

は0%とした。また、ルータ1とルータ2でパケットをキューイングする長さは50個とした。PL1とPL2へはUSBカメラを接続し、Linux用のTV会議システムであるgnomemeetingでリアルタイムストリーミング通信を行う。実際の環境では帯域と遅延は動的に変化するが、今回の予備実験は通信中に帯域や遅延を変化させずに行った。またプロトタイプでは、測定用パケットによる帯域測定の精度が低いため、各パケット分配方式で使用する上りと下りの帯域は静的な値を予め設定しておくこととした。なお表1にPL1とPL2の間でできる4つの経路1~4の詳細を示す。ここで経路1と経路2の上りリンク、経路3と経路4の上りリンクは共有される。また、経路1と経路3の下りリンク、経路2と経路4の下りリンクも共有される。

4.2.2 実験1

表2に実験1で設定した各経路と帯域を示す。実験1では各パケット分配方式の基本的な特性を評価するため、全ての経路の上りと下りの帯域が対称であり各経路間で遅延差のない環境を想定する。現在NTTドコモが提供するFOMA F2402では上り下りとも384kbpsであるため全ての端末の帯域を上り下りとも384kbpsとなるように設定した。また遅延は、FOMAで実際にRTTを測定したところ、約400msecであったため各経路の片道遅延を200msecとした。この環境で最大ビットレートが100kbps~700kbpsであるストリーミングを100kbpsづつ変化させて流した。それぞれのストリーミングにおいて、PL1とPL2が直接通信する場合とパケット分配方式1~3を用いたPAN2PAN通信を行う場合のパケット損失率を比較した。実験結果を図6に示す。PL1とPL2が直接通信する場合、400kbps以上のストリーミングを流したと

	経路 1	経路 2	経路 3	経路 4
上り帯域	128kbps	128kbps	384kbps	384kbps
下り帯域	384kbps	128kbps	384kbps	128kbps
遅延	150msec	150msec	200msec	200msec

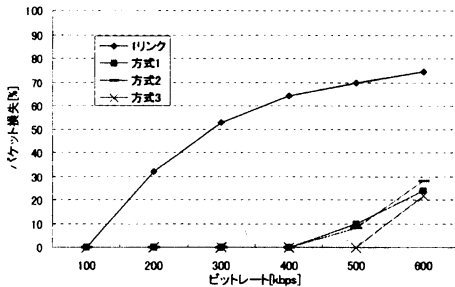


図7 実験2における各通信のケット損失率

ころからケット損失率が増加している。これは PL1 と PL2 の上りと下りの帯域がともに 384kbps であったため 400kbps 以上のストリーミングでは輻輳が発生し、ケットが損失したからであると考えられる。一方、方式1~3を用いた PAN2PAN 通信は 700kbps のストリーミングを流してもケット損失を起こさず通信できた。本実験環境の場合、PAN2PAN 通信で全ての経路を有効に利用できれば、上りと下りの帯域はともに 768kbps まで論理的に増やすことができる。帯域にある程度の余裕があり各経路の上りと下りの帯域が対称であり各経路間で遅延差がない場合、方式1~3を用いた PAN2PAN 通信においてケット損失率に差がなく効率的にケット分配ができる。

4.2.3 実験 2

表3に実験2で各経路に設定した帯域と遅延を示す。実験2では、各経路の上りと下りの帯域が非対称であり各経路間で遅延差のある環境を想定し、より現実的な環境で各ケット分配方式に差が生じるかを評価した。PL1の上り帯域とPM2の下り帯域が128kbpsで、遅延が150msecであるとする。この環境で最大ビットレートが100kbps~600kbpsであるストリーミングを100kbpsづつ変化させて流した。実験1と同様にそれぞれのストリーミングにおいて、PL1とPL2が直接通信する場合と方式1~3を用いたPAN2PAN通信を行う場合のケット損失率を比較した。実験結果を図7に示す。PL1とPL2が直接通信する場合、200kbps以上のストリーミングを流したところからケット損失率が増加している。これはPM2の下り帯域は384kbpsだがPL1からの上り帯域が128kbpsであるため上りリンクがボトルネックとなり、200kbps以上のストリーミングでは輻輳が発生し、ケットが損失したからであると考えられる。一方、方式1~3を用いたPAN2PAN通信を比較すると、方式1と方式2は500kbpsのストリーミングを流したところからケットが損失率が増加しているが、方式3では500kbpsのところまでケット損失が発生せず600kbpsのところまでケット損失が増加している。本実験環境の場合、PAN2PAN通信ですべての経路を

有効に利用できると、上りと下りの帯域は512kbpsまで論理的に増やすことができる。500kbpsのストリーミングではこの論理帯域にもほとんど余裕がなく、ケット分配がかなり効率良く行われないと輻輳が発生すると思われる。方式1~方式3でケット損失が発生しているが、方式1と方式2では帯域を最大限に利用できなかったため輻輳が発生してケット損失が発生したと考えられ、方式3では帯域を最大限に利用できたため輻輳を回避できケット損失が発生させず通信できたといえる。600kbpsのストリーミングでは、上述の論理帯域を上回っているため、どの方式も輻輳が発生し、ケット損失が発生したと考える。

以上の結果より、各経路の上りと下りの帯域が非対称であり各経路間で遅延差がある環境では、方式3を用いたPAN2PAN通信において効率的にケット分配ができたといえる。実際に、方式3は各経路の上りと下りの帯域だけでなく動的に測定した遅延も考慮してケット分配しているため、安定したストリーミング映像を確認できた。

5. 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評価した。その結果、ケットの到着時刻を予測してケット分配を行った場合、最もケット損失の少ない安定したストリーミング通信を実現できることを確認した。

今後、ケット損失への対応として未実装であるFEC機能を加えるとともに、ネットワークの状態に応じて動的に制御方法を変える方式についても検討し、実際のモバイル環境で評価を行う。

参考文献

- 1) T. Nguyen and A. Zakhor, "Path Diversity with Forward Error Correction(PDF) System for Packet Switched Networks", INFOCOM 2003, pp.663-672.
- 2) 川西 智也, 中村 泰典, 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.
- 3) 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.
- 4) 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", 情処全大 2002, pp.485-486.
- 5) Divert Socket for Linux.
<http://sourceforge.net/projects/ipdivert>.
- 6) Kevin Lai and Mary Baker, "Nettimer:A tool for measuring bottleneck link bandwidth", USITS 2001, pp.123-134.
- 7) ITU-T Rec. G.114 "One-way transmission time", May 2002.
- 8) 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", DICOMO 2002, pp.157-160.