

アドホックなネットワークコミュニティのための
複視点ライブ映像共有システム
Multi-view Live Sharing and Recording System
for Ad Hoc Network Community

廣野 大地[†] 高井 昌彰[‡]
Daichi Hirono Yoshiaki Takai

1. はじめに

本論文ではアドホックなネットワークコミュニティにおいて個人が複視点映像を自由に撮影・編集できるよう支援するライブ映像共有システムを提案する。

近年、個人が映像を扱う機会は急速に増加している。以前はビデオ撮影のために専用のカメラを持ち出さなければならなかったが、現在は各個人が持ち歩いているスマートフォンや携帯ゲーム機などに搭載されているカメラを用いて、事前準備なしに撮影を行うことができる。また Youtube などの動画投稿サイトの普及により、撮影した映像を公開することも容易になった。

しかし個人により撮影・公開される映像は 1 つのカメラで撮影された単視点映像であるため、テレビなどで放映されている複視点映像と比べて表現の自由度が低い。複視点映像は複数のカメラで撮影された映像を編集作業によってまとめあげたもので、アップとロングの映像を切り替えたり、複数の視点から撮影された映像を切り替えたり、別の対象を撮影した映像へ瞬時に移ったりといった表現が可能である。複視点映像を撮影するためにはカメラと撮影者を必要な分用意し、撮影後に編集を行わなければならない、個人には手間が大きい。

我々は個人がビデオ撮影に使用しているスマートフォンなどのデバイスが無線ネットワーク機能も有していることに着目し、各々のカメラ映像を共有することで、個人の複視点撮影をサポートするシステムを開発した。本システムのユーザーは、ライブ会場など多数のユーザーが集まる場所でアドホックなコミュニティを形成し、自分のカメラ映像と他人のカメラ映像を自由に切り替えながら映像を撮影することができる。事前に複数台のカメラを用意したり、撮影後に映像の編集を行ったりする必要はない。

1.1 本論文の構成

本論文は以下の構成をとる。

第 1 節では提案システムの概要について説明する。

第 2 節では先行研究および類似システムについて述べる。本システムのように映像を複数のノード間でやり取りするシステムは多数存在している。それらのシステムの代表例を紹介し、本システムとの違いを示す。

第 3 節では本システムの構成を述べる。本システムは Wi-Fi ネットワーク、各ユーザーが使用するノード端末およびその上で動作する専用ソフトウェアから構成される。

第 4 節では映像伝送について述べる。本システムではプ

レビュー用と録画用の 2 種類の映像に適するように伝送を行っている。レビュー映像は各ノードが現在撮影している映像を他ノードに示すもので、ユーザーはこの映像を参考に撮影に用いるカメラを選択する。そのためレビュー映像の伝送にはリアルタイム性が要求される。一方、録画用の映像は実際に記録される映像であり、データの損失なく伝送することが求められる。本システムは 2 種類の映像伝送に対する要件を満たすために、レビュー映像の伝送にはリアルタイム性の高い UDP と RTP/RTCP を用い、録画時には失われた情報をパケットの再送により修正するという手段をとる。また、録画時にはネットワーク上の遅延を補正し、切り替える映像間での同期を実現する。

第 5 節では映像伝送以外のシステムの補助的な機能について述べる。各ノードを操作しているユーザーのコミュニティにおいて、各映像のカメラワークはそのカメラのユーザーにゆだねられている。しかし時には他のユーザーに「このカメラワークはよい」「もっと対象に寄ってもらいたい」など簡単なメッセージを伝達することで協調的な撮影が形成されやすい。そのため本システムはマークによる簡単なコミュニケーションをサポートしている。

第 6 節ではシステムの動作確認について説明する。上記で述べたシステムの動作を確認するために、本システムを Linux 上のアプリケーションとして実装し、USB カメラを取り付けたノート PC 上で動作確認を行う。

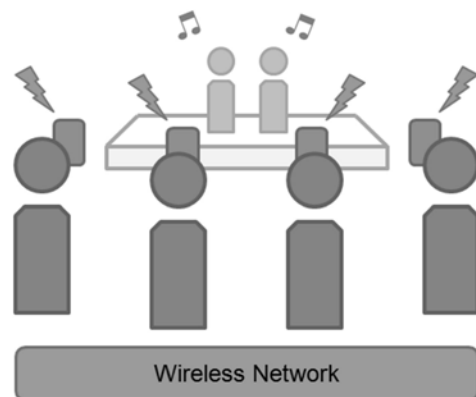


図 1 システムの使用イメージ

2. 既存システムとの比較

本システムのようにネットワーク上で映像のやり取りを行うシステムは数多く存在している。本節ではそれらのシステムを紹介し、本システムとの違いを 3 つの点から説明する。

第 1 の観点はそのシステムが映像を伝送する際、遅延を許容できるか否かである。本システム上の各端末は、自身のカメラ映像を常にネットワークに対して配信している。

[†] 東京大学大学院情報理工学系研究科, Graduate School of Information Science and Technology, The University of Tokyo

[‡] 北海道大学情報基盤センター, Information Initiative Center, Hokkaido University

この映像はプレビュー映像として他の端末の撮影画面に表示され、ユーザーのカメラ選択の指針となるので、できるだけ低レイテンシーで伝送されることが求められる。一方録画処理はソフトウェア内部で行われるため、録画用映像の伝送には遅延があっても問題ない。

第2の観点はそのシステムが映像を伝送する際に映像の損失を許すか否かである。本システムではプレビュー映像の伝送については多少の映像の損失は許容できるが、録画映像の伝送時には映像の損失がないことが望ましい。

第3の観点は複数のサブネットをまたぐ伝送が必要であるか否かである。ほとんどの映像伝送システムは遠隔地間で利用されるため、複数のサブネットをまたぐ伝送を行う必要があるが、本システムはあるひとつの場で映像を共有するものであるためその場のネットワークの外へ伝送を行う必要がない。

2.1 動画共有サイト

Youtube に代表される動画配信サイトは、あらかじめユーザーからアップロードされたビデオ映像を一旦システム内に保存し、他のユーザーの再生要求に対応して映像データを伝送する、巨大な映像のデータベースである。システムにはブラウザを通じてアクセスする。ユーザーはアップロードする映像を一旦ファイルの形で用意し、それをシステムに送信する。システムはそれを適切なフォーマットに変換して保存し、他のユーザーからの指示に応じて配信する。映像の伝送には HTTP と TCP が用いられ、データの損失はない。リアルタイム性が要求されないため遅延には寛容である。

2.2 動画ストリーミングサイト

UStream.tv に代表される動画ストリーミングサイトはブラウザからアクセスする点を含めて動画共有サイトとよく似ている。しかし、動画共有サイトが保存済みの映像を配信するのに対して、動画ストリーミングサイトは送信ユーザーが現在カメラで撮影中の映像を、そのまま視聴者であるユーザーへ伝送する。伝送はリアルタイムで行われるが、ユーザー間で双方向に行われるわけではないため、実際にはある程度の遅延は許容される。データの損失はないことが多い。

2.3 テレビ会議システム

Skype や Polycom に代表されるテレビ会議システムは、遠隔地間で映像を多対多かつ双方向にやりとりするシステムである。システムは専用のソフトウェアを用いて利用する。ユーザー間のコミュニケーションが第一目的であるため、多少のデータの損失は許容しても、できるだけ低レイテンシーで映像をやり取りすることが求められる。

2.4 まとめ

以上3つのシステムの特徴をまとめたのが表1である。この表からわかるように、動画共有サイトと動画ストリーミングサイトが行う映像の伝送は、損失がない代わりに遅延が大きく、テレビ会議システムで行われている伝送は、遅延が大きい代わりに多少のデータの損失がある。また、上記のシステムはすべてサブネットをまたいだ伝送を行っている。

一方本システムは、プレビュー映像では遅延の少ない伝送を、録画映像ではデータの損失がない伝送を行う必要があるため、各映像に適した伝送を行う工夫を行っている。複数のサブネットをまたいだ伝送を行う必要がないため、マルチキャストを利用して効率的に一对多の伝送を行うことができる。

| | 損失がない | 遅延が少ない | サブネットをまたぐ |
|--------------|-------|--------|-----------|
| 動画共有サイト | ○ | × | ○ |
| 動画ストリーミングサイト | ○ | × | ○ |
| テレビ会議システム | × | ○ | ○ |
| 本システム | ○ | ○ | × |

表1 各システムの特徴

3. システムの構成

本システムは Wi-Fi ネットワークとそのネットワークに参加しているノード端末およびその端末上で動作する専用ソフトウェアから構成されている。ノード端末はカメラと無線 LAN アダプタを持っており、撮影した映像を、無線ネットワークを通じて他ノードへ配信する。

実際のシステムの動作はノード端末上で動作するアプリケーションによって実現される。図2はアプリケーションの撮影画面である。アプリケーションは初期状態では、中央部に自身のカメラ映像を、下部に同じネットワークの別端末のカメラ映像を表示する。ユーザーがマウスクリックで別端末の映像を選択すると、その映像が中央部の映像と入れ替わり、録画される内容も別カメラの映像になる。切り替えは撮影中に行うこともできる。最終的に録画される映像は図3のように自分のカメラ映像と他人のカメラ映像が交互に現れる複視点映像となる。



図2 撮影画面



図3 録画される映像のイメージ

アプリケーション内部の構成は図4に示すとおりである。カメラモジュールで撮影・エンコードされた映像は送信モ

ジュールによって RTP ネットワークへマルチキャストされる。送られた映像は別ノードのプレビューモジュールと録画モジュールが受信し、ウィンドウへの表示やファイルへの記録を行う。録画モジュールは、受信したデータに欠落があった場合再送要求をデータの送信者に対して送信する。再送要求は送信者端末の再送モジュールが受け取り、あらかじめ保存しておいた自身のカメラ映像データから対応するデータを選択して再送する。

以上のモジュールとは独立に、同期モジュールとコミュニケーションモジュールが動作している。各端末の同期モジュールはお互いにパケットをやり取りしてネットワークの遅延を計測し、そこで得た情報を録画モジュールに報告する。ネットワークの遅延情報を得た録画モジュールは各端末のカメラ映像を切り替える際に適切な遅延を与え、映像間の時間差が小さくなるように補正を行う。

コミュニケーションモジュールはユーザーの操作に応じてデータをやり取りし、ユーザー間でマークを使ったコミュニケーションを可能にする。

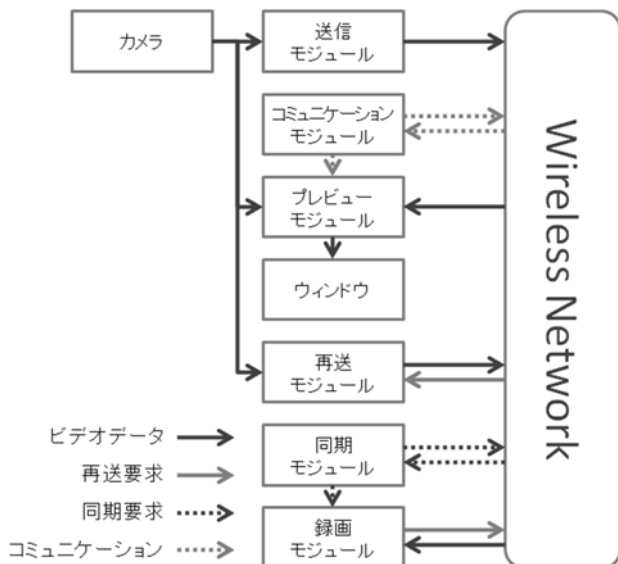


図4 ソフトウェアの構成

4. 映像の伝送

4.1 プロトコル

本システムでは UDP と RTP/RTCP[3]を用いて映像の送信を行っている。ネットワーク上の各端末は、映像切り替えの参考になるように、プレビュー用の映像を常に配信している。この映像はユーザー操作の参考になるため、できるだけ少ない遅延で伝送できることが望ましい。また、複数の端末に向けてまったく同じデータを送るため、一回の送信で複数の端末にデータを送信できると効率が良い。

UDP は同じトランスポート層のプロトコルである TCP と比較して、データの欠損を許容する、即時性に優れる、マルチキャスト配信[1]に対応するなどの特徴がある。TCP は高信頼性の通信を実現するために、受信データをバッファリングし欠落パケットがあった場合それを再送によって埋めてからアプリケーションにデータを渡す。そのため欠落パケットがあった場合、パケットの再送が行われるまでアプリケーションはデータを受け取ることができない。

一方 UDP はパケットの欠落を許容し受信したデータをそのままアプリケーションに渡す。また UDP はマルチキャストと呼ばれる送信方法に対応し、一回の送信で複数の相手にデータを送ることができる。

RTP/RTCP はメディアのリアルタイム伝送に使用するプロトコルで、UDP 通信によって欠損したデータを補って再び映像データを復元する機能を持つ。RTP は実際のメディアデータを伝送し、RTCP は RTP 伝送の制御を行う。

RTP/RTCP によって送られたパケットには SSRC と呼ばれる番号が振られる。システムは SSRC を参照することで、パケットの送信元を区別することができる。

4.2 プレビュー映像の伝送

映像伝送に使用するプロトコルに UDP、RTP/RTCP を選択したことで、プレビュー映像は低レイテンシーで伝送することができる。カメラモジュールが撮影した映像は、送信モジュールがエンコード・RTP パケット化し、マルチキャストによってネットワークに配信される。マルチキャストはネットワーク上のマルチキャストグループに対して行われ、同グループに属するすべての端末が受信できる。これにより各端末は 1 回の送信ですべての端末に映像データを配信することができる。

4.3 録画映像の伝送

システムはプレビュー用に受信した映像データを録画にも使用する。ただし、パケットの再送による欠落データの補償と、複数映像間の同期を行い、録画される映像の品質を高める努力をしている。

4.3.1 再送

プレビュー映像のマルチキャスト配信によって各ノードはネットワーク上のすべてのノードのカメラデータを受信することができる。ユーザーから録画の指示があった場合、このデータをファイルに保存していくが、UDP 通信によって欠損したデータは再送により補償する。

録画モジュールはリングバッファと書き込み用・読み込み用の 2 つのカーソルを持っている。リングバッファは多数のセクションに分かれており、各セクションには RTP パケットを 1 つ保存することができる。また各カーソルはリングバッファ上のいずれかのセクションを指し示し、書き込みカーソルは常に読み込みカーソルに先行する。

録画処理が始まると、録画モジュールは記録対象となったカメラ映像のパケットを一旦リングバッファに保存する。図 5 はリングバッファを直線に表した様子を示している。パケットが受信されると、書き込みカーソルは自身が指しているセクションに受信パケットを保存して次のセクションへ進む。RTP パケットに欠落が生じた場合、書き込みカーソルは欠落部分のセクションを空にして次へ進む。(図 5、2 段目) 読み込みカーソルはセクションからパケットを読み出し、そのセクションを初期化し、次へ進む動作を繰り返しながらデータを読み出していく。欠落パケットによって生まれた空セクションに到達した場合、そこで読み込みをいったん止める。再送モジュールは定期的に書き込みカーソルと読み込みカーソルの間の領域を検査し、空セクションがあった場合は RTP パケットの再送要求を行う。

(図 5、3 段目) 再送要求は RTCP のアプリケーション定義パケットを用いて送信される。要求に応じて送信者ノ-

ドから再送されたデータは、録画モジュールにより受信されリングバッファの対応する空セクションに書き込まれる。空セクションが埋まると読み込みカーソルは読み込み処理を再開し、書き込みカーソルに追いつくか次の空セクションに到達するまで読み込み処理を続ける。

パケットの再送は、送信者ノードの再送モジュールが行う。再送モジュールは自身の過去のカメラ映像を一定時間分保存しており、再送要求に応じて欠落パケットを再送する。

もしネットワーク上の理由により、パケットが著しく欠落し書き込みカーソルと読み込みカーソルが一定以上に離れた場合、再送モジュールはパケットの再送をあきらめて読み込みカーソルを強制的に進める。この場合、欠落したデータは補償されず、録画される映像は不完全なものとなる。

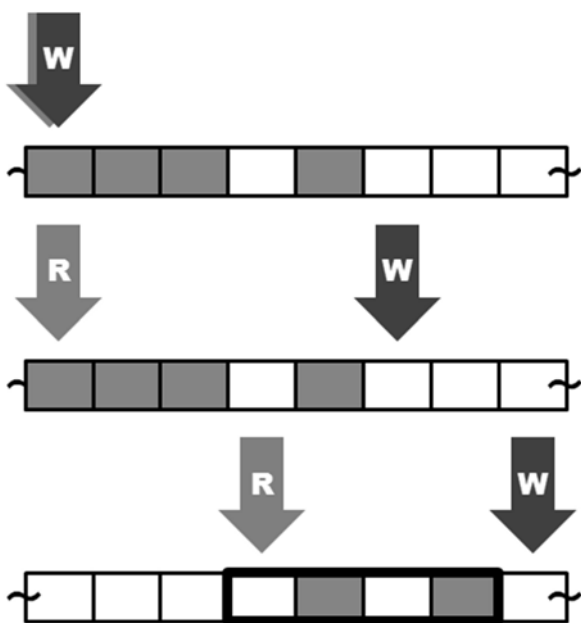


図5 パケットの再送

4.3.2 同期

本システムは録画される映像に対し、映像間の同期を行っている。本システムは自分のカメラの映像や他人のカメラ映像など、複数の映像を切り替えて録画している。自分のカメラ映像はまったく遅延を受けないが、ネットワークを通じて送られてきた映像はそれぞれ異なる時間遅延しているかもしれない。現在録画されている映像よりも先行している映像に切り替えを行うと、切り替えた瞬間に映像内の時間が先へ飛んでしまう。逆に現在録画されている映像よりも遅れている映像へ切り替えを行うと、同じタイミングのシーンが2重に録画されることになる。

そこでシステムは、各映像がネットワーク上で受けた遅延を計測し、同期したうえで映像を録画する。遅延の計測は自ノードから他の各ノードに対してパケットを送信し、そのパケットが送り返されるまでの時間を測ることで行う。各ノードに対する遅延を導いたあと、システムは最も遅延が大きいノードに合わせるように他の映像を遅らせながら録画を行う。また録画に関するユーザーの操作（録画の開始や映像の切り替えなど）が適用されるタイミングも同様に遅らせられる。すべての映像が同じ量の遅延を受けてい

るため、映像の切り替えがどのように行われても、時間的なずれは発生しない。

システムは録画映像に対してのみ同期を行い、プレビュー映像に対しては同期を行わない。同期を行うとすべての映像は最も遅延が大きいノードに合わせる形で遅らされてしまい、低レイテンシー性が保てないからである。

5. 補助的な機能

本システムにはユーザーの複視点映像録画をサポートする目的で2つの補助的な機能が実装されている。1つ目はコミュニケーション機能で、ユーザー間でマークを用いた簡単なコミュニケーションを行えるようにするものである。2つ目は切り替え効果で映像を切り替える際に映像にエフェクトを追加する。

5.1 コミュニケーション機能

本システムを利用してライブ映像を共有し合っているユーザーは必ずしも知り合い同士とは限らない。システムは同じネットワークにいるノードを自動的にマルチキャストグループに参加させるため、その場にたまたま居合わせた人が映像の交換を行うこともある。見知らぬ人同士でも協調的に撮影を行えるように、システムはユーザーがマークによる簡単な意思疎通を行う機能を提供する。この機能を用いると、ユーザーは任意のユーザーに指定したマークを送ることができる。受信ユーザーの画面には送られたマークが一定時間オーバーレイ表示される。

マークの送受信はコミュニケーションモジュールが、RTCPパケットを用いて行う。ユーザーが送信したいマークと送信先を一覧から選択すると、コミュニケーションモジュールは対象のノードにRTCPパケットを送信する。RTCPパケットにはマーク番号が含まれており、これを受信したノードは対応するマークを画面に表示する。

5.2 切り替え効果

テレビ等でみられる複視点映像は、映像の切り替え時にエフェクトを適用していることがある。本システムも「通常切り替え」と「クロスフェード」の2種類の切り替え効果を用意している。ユーザーは事前に使用する効果を選択することで、切り替え時に効果を適用できる。



図6 クロスフェードによる画像の切り替え

6. 動作確認

前述したシステムの機能を確認するために、動作実験を行った。実験に使用するノード端末としてノートPC(表2)にUSBカメラをつけた物を使用し、そのPC上で専用のLinuxアプリケーションを動作させた。アプリケーションはC++で作成し、以下のライブラリを用いた。

- GStreamer
- GTK
- Boost

動作確認はローカルテストと Wi-Fi テストの 2 種類を行った。ローカルテストでは単一のノート PC 上に仮想的なノードデバイスのインスタンスを立ち上げ、不安定なネットワークをシミュレートしてシステムの動作を確認した。Wi-Fi テストでは実際の Wi-Fi ネットワーク上でシステムを動作させた。

| | |
|-----------|--|
| Machine | Dell Inspiron Mini 10 |
| Processor | Intel Atom Processor N450 (1.66GHz, 512KB L2 cache) |
| OS | Ubuntu 10.04 |
| Memory | 1GB DDR2-SDRAM |
| Wireless | Dell Wireless 1397 Integrated Wireless LAN Half-Mini Card (802.11b/g) |

表 2 動作確認で使用した PC

6.1 ローカルテスト

初めに単体のノート PC 上でローカルテストを行った。ローカルテストではひとつの PC 上に仮想的に複数のノードを配置し通信を行わせる。ノード同士はローカルループバックを用いて通信を行うため、パケットには損失がなく遅延もほとんどない。

ローカルループバック上で Wi-Fi ネットワークの様子を再現するためにミラーノードを使用した。ミラーノードは受信した RTP/RTCP パケットの SSRC を自分の物に書き換え、残りのデータはコピーして送信しなおす。データを再送信する際に指定の比率でパケットを破棄したり、パケットに対して遅延を加えたりすることができるため、任意のネットワークの状態 (パケットの損失率・遅延) でのシステムの振る舞いを調べることができる。ミラーノードは映像データを運ぶ RTP ノードだけではなく、再送要求や同期に使用している RTCP パケットにも同様のミラーリング・パケットの破棄・遅延の追加を行う。

ここではノードインスタンスとミラーノードを 1 台ずつ起動して 30 秒間の伝送を行った。通常のノードが送った内容は、ミラーノードで指定の損失率に従ってパケットが破棄され、さらに遅延が加えられたあとノードに戻される。そのため、ノードからみた場合、ミラーノードから送られてきたデータは不安定なネットワークを通したように見える。

はじめに損失パケットの再送機能を確認するために、ミラーノードに 0.05 から 0.25 までのロス率を与え、プレビューモジュールと録画モジュールでのパケットの損失率を計測した。録画モジュールでは再送によって損失パケットを補った後の値を調べるため、損失率がプレビューモジュールよりも低くなるのが期待される。

次に映像間の同期機能を確認するためにミラーノードに 100ms から 600ms の遅延を与え、ミラーリングされた映像と基準映像との時間差を調べた。基準映像は自身のカメラによって撮られた映像で、遅延がかかっている。一方ミラーリングされた映像は、送信した自身のカメラ映像をミラーノードを経由して受信したもので、指定の遅延がかかっている。プレビュー時には映像間の同期は行わないため、ミラーリングされた映像と基準映像には時間差が存在する。録画時には同期モジュールがミラーリングされた映像にかかっている遅延を調べ、同じだけの遅延を基準映像に与え

るために、両映像間の時間差はプレビュー時よりも小さくなるのが期待される。2 つの映像にどれだけの時間差があるかを調べるために、ノードは受信されたパケットのシークエンス番号を調べている。基準映像のパケットが受信された時に、そのパケットのシークエンス番号を記録し、同じシークエンス番号のパケットがミラーリング映像のパケットとして受信されるまでの時間を調べる。

ローカルテストの結果は以下ようになった。

図 7 はミラーノードに指定するロス率を 0.05 から 0.25 まで変化させた場合の、プレビューモジュールと録画モジュールでのロス率の変化である。specified で示されている値がミラーノードに指定されたロス率で、preview で示された値がプレビューモジュールで実際に計測されたパケットのロス率である。プレビューモジュールではパケットの再送を行わないため、ほぼ指定されたロス率と同じだけのロス率が出力されている。一方 recording で示されている録画モジュールでのパケットのロス率は、すべての場合で 0 である。これはパケットの再送によってすべてのパケットが完全に補償されたことを示す。

図 8 はネットワークの遅延を 100ms から 600ms へ変化させた時の各モジュールでの基準映像とミラーリング映像との時間差の変化である。パケットのロス率と同様に、プレビューモジュールではミラーノードに指定したのとほぼ同じだけの時間差が検出されているが、録画モジュールではこの時間差が補正されて両映像が同期されているのがわかる。

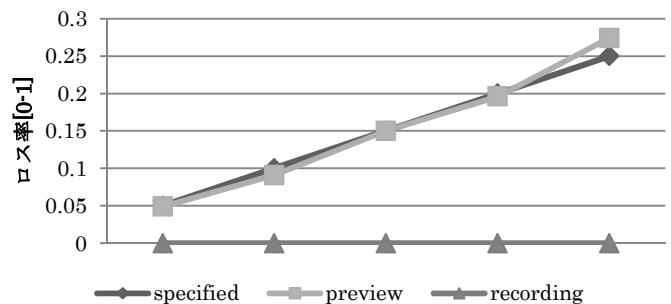


図 7 ローカルテストでのロス率

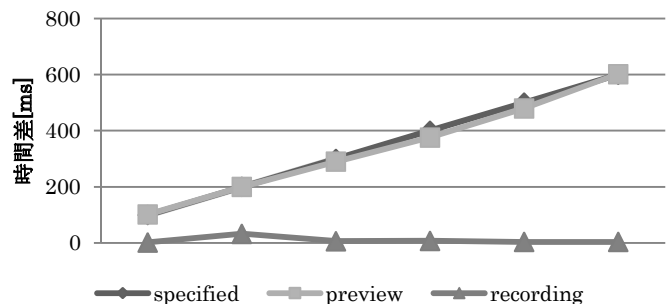


図 8 ローカルテストでの映像間の時間差

6.2 Wi-Fi テスト

Wi-Fi テストでは 2 台のノート PC を使い、実際の Wi-Fi ネットワーク上でシステムを実行した。2 つの PC は送信用と計測用で、送信用の PC では 1~4 台のノードインス

タンスを起動してそれぞれ映像の送信を行い、ネットワークに負荷をかけた。計測用の PC では、送信側から送られてきた映像を受信して、そのロス率を調べた。伝送は各ノードインスタンスの台数で3回、各30秒間行った。

Wi-Fi テストの結果は図9のようになった。横軸は送信側 PC で動作させたノードインスタンスの台数と受信モジュールの種類を、縦軸は3回の実験それぞれでのロス率を色分けして示している。たとえば preview1 の棒グラフは送信ノード数が1の時プレビューモジュールで検出されたロス率を表している。3回の伝送のうち1回はロス率が0であったため、棒グラフには2つ分の値しか表示されていない。

結果をみるとノード数にかかわらず、プレビューモジュールよりも録画モジュールのほうが、ロス率が低くなっていることが分かる。またノード数が4に達すると、録画モジュールでのロス率が大きく増加している。これはノード数が増えることで無線ネットワークの信頼性が低下したことと、それに伴い再送要求や応答のパケットの数が増えたことでさらにネットワークの帯域が圧迫されたことが原因であると考えられる。

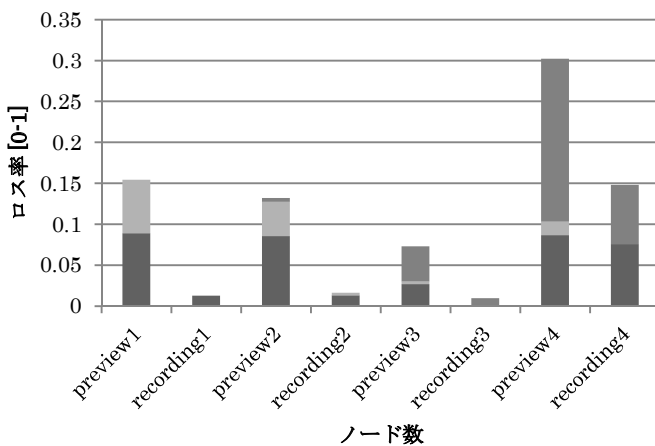


図9 Wi-Fi テストのロス率

7. まとめ

本論文では Wi-Fi ネットワークによってライブ映像を互いに共有する、複視点映像撮影支援システムを提案し、実装したシステムを用いて動作実験を行った。本システムを用いることで、ライブ会場など多数のユーザーが集まる場所でアドホックなコミュニティを形成し、自分のカメラ映像と他人のカメラ映像を自由に切り替えながら映像を撮影することができる。本システムはユーザーが事前準備や事後編集なしに複視点映像を撮影することが可能である。

各端末のカメラ映像は UDP マルチキャストと RTP/RTCP を用いて他の端末に低レイテンシーで送信される。また、配信された映像が実際に録画される場合は、システムがロスパケットの再送や映像間の同期などを行い、より映像の品質を高めようとする。

パケット再送機能や映像同期機能によって実際に映像に改善がみられるか確認するために、ローカルテストと Wi-Fi テストの2つの動作確認を行った。パケット再送機能や映像の同期機能が正しく働いていることが確認できたが、

同時に送信ノード数が増えることで再送に限界が表れることも分かった。

本研究の課題として以下の点があげられる。

- ネットワークの負荷に応じて配信ノード数を制限する機能が必要である。Wi-Fi テストの結果から配信ノード数が増加するとパケットのロス率が増加することが分かっている。
- 映像のビットレートや再送要求を出す頻度などのパラメーターは、ネットワークの状態に応じて調節されるべきである。
- 録画対象のノードが突然ネットワークから抜けた場合に、次に切り替える映像をあらかじめ指定できるようにしてあるとよい。

参考文献

- [1] B. Quinn and K. Almeroth. IP Multicast Applications: Challenges and Solutions. RFC 3170 (Informational), September 2001.
- [2] D.L. Mills. Network Time Protocol (NTP). RFC 958, September 1985. Obsoleted by RFCs 1059, 1119, 1305.
- [3] H. Schulzrinne, S. Casher, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761, 6051.
- [4] 前田 朋孝, 小塚 真啓, 岡部 寿男, “PR-SCTP を用いた高信頼性ストリーミング伝送”, 情報処理学会研究報告, IOT, 2009(21), 43-47, 2009-02-26
- [5] 増井 信彦, 下倉 健一郎, “映像を共有するコミュニティシステムの構築と検証”, 電子情報通信学会技術研究報告, HIP, 104(746), 19-24, 2005-03-17