

マルチメディアストリーミングにおける 再生時刻を考慮したパケット再送制御

萩野 浩明^{†1} 宮崎 雄一朗^{†1} 尾上 裕子^{†1}
 渥美 幸雄^{†2} 谷口 雅昭^{†3}
 駒木 寛隆^{†3} 山内 長承^{†4}

本論文では、マルチメディアストリーミングにおける再生時刻を考慮したパケット再送制御方式を提案する。マルチメディアストリーミングではクライアントはパケットを受信しながらコンテンツの再生を行う。そのため、コンテンツの再生が開始されるとそれぞれのパケットについて再生すべき時刻が決定し、それまでにパケットがクライアントに到着しなければそのパケットはクライアントアプリケーションによって破棄される。提案方式では、サーバが RTT とクライアントバッファの時間的蓄積量を比較し、損失したパケットを再送した際に再生時刻までにクライアントに到着するかどうかを予測する。そして再生時刻に間に合うと予測されるパケットのみが再送される。これによって、パケットが損失した際の無駄なパケット再送を回避することが可能となる。さらに本論文では提案方式の評価実験を行い、単純な NACK ベースの再送制御との比較を行う。この実験結果から、提案方式が再送されるパケット数を大きく減少させるにもかかわらず、再送が成功するパケット数がほとんど低下しないことを示す。

Playtime-oriented Retransmission Control Method for Multimedia Streaming

HIROAKI HAGINO,^{†1} YUICHIRO MIYAZAKI,^{†1} YUKO ONOE,^{†1}
 YUKIO ATSUMI,^{†2} MASAOKI TANIGUCHI,^{†3} HIROTAKE KOMAKI^{†3}
 and NAGATSUGU YAMANOUCHI^{†4}

In this paper, we propose playtime-oriented retransmission control method for multimedia streaming. In multimedia streaming, playtime of each packet is decided when a client starts playing contents. If a packet does not arrive at a client until its playtime, the packet is discarded by the client application. In our method, a client sends a server NACK packet when it detects packet losses. NACK packet includes not only identifiers of lost packets but also playtime of a top packet of client buffer to a server. The server subtracts playtime of the top packet of client buffer from that of lost packet, and compares result of the calculation to RTT between the server and the client. If former is larger, the server retransmits the lost packets to the client. If not, the server retransmits no packets. Therefore, the number of unnecessary retransmission of packets is decreased. Moreover, we estimate our method and compare it to the simple retransmission method using NACK packets. From the results, we show that Our method hardly decreases the number of retransmitted packets which achieve a client until their playtime while our method drastically decreases the number of retransmitted packets.

1. はじめに

近年、有線、無線を問わずネットワークの広帯域化が進んでいる^{1),2)}。これにともない、より高品質のサービスに対する要求が高まっている。特に、これまでのネットワークではネットワーク上での交換が困難であった、動画などのマルチメディアコンテンツに対する期待が大きい。動画配信サービスは、ダウンロード型サービスとストリーミング型サービスとに大きく

†1 株式会社 NTT ドコモネットワーク研究所
Network Labs, NTT DoCoMo

†2 専修大学経営学部
Sensyu University

†3 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research, Tokyo Research Laboratory

†4 東邦大学理学部情報科学科
Department Information Science, Toho University

分類できる。ダウンロード型サービスは、動画ファイルをユーザがサーバからダウンロードして、ローカルディスクに保存し、保存したファイルを動画再生ソフトで再生する。このサービスでは、配信されるファイルがたとえば動画コンテンツであるか、テキストファイルであるか、といったことは制御にほとんど影響を与えない。従来、狭帯域ネットワークで交換されていたファイルのサイズが大きくなったものが、広帯域ネットワークにおいてダウンロード型サービスでやりとりされるコンテンツであると考えられることができる。ネットワークの広帯域化によって、サイズの大きいファイルの交換が現実的なものとなったが、これは TCP などの従来技術によって十分実現可能である。

一方、ストリーミング型サービスは、ユーザはクライアント端末からサーバにアクセスし、サーバから動画コンテンツや音楽/音声コンテンツの配信を受けながら、同時に再生を行うものである。ローカルディスクに保存する必要がないため、不正コピーが行われる可能性が低く、著作権のあるコンテンツ配信に有効である。また、ライブ放送など、リアルタイム性が必要とされるコンテンツの配信手段として重要であると考えられる。

ここで、ネットワークにおけるパケットの損失は、マルチメディアストリーミングのサービス品質を大きく劣化させる要因の1つである。特に、Mpeg フォーマットでエンコーディングされたコンテンツなどのように、フレーム間の差分情報を用いているコンテンツの場合、1つのパケット損失によって、さらに多くのフレームが乱れてしまうことがある。しかし、多くのマルチメディアストリーミングサービスでは、リアルタイム再生を実現するためにコンテンツ配信に RTP³⁾/UDP を用いているため、パケットの損失をリカバリすることはできない。この問題を解決するために、IETF では、RTP を用いてパケットを再送するためのプロトコル⁴⁾が提案されている。現在、このインターネットドラフトは expired されているが、同様の内容のものが別のドラフトとして提案されている^{5),6)}。しかしこれらの差分は主にパケットフォーマットの軽微な変更であり、方法論や有効性の議論にはほとんど影響がない。そこで以下の議論では文献 4) を IETF 提案として扱い、新しいフォーマットへの対応は今後の課題とする。文献 4) で提案されているプロトコルでは、クライアントは RTP の制御プロトコルである RTCP を用いてサーバに再送要求メッセージを送信し、サーバは再送要求メッセージの中で指定されているパケットのうち、重要度の高いものを再送する。しかし、ストリーミン

グ型サービスでは損失したパケットを再送しても、そのパケットが構成するフレームの再生時刻までにクライアントに到着しなければそのパケットは無駄になってしまううえに、再送によってサーバがクライアントに送信する総トラフィックが増加し、メインのストリームとして送信しているパケットの到着時刻までが遅くなってしまう可能性もある。以上から、マルチメディアストリーミングでは、再送によるさらなる通信状況の悪化を避けるために、再送するパケットを厳選し、本当に必要なパケットのみを再送しなければならない。

そこで本論文では、クライアントでのコンテンツ再生状況を考慮して、再送するパケットが再生時刻に間に合うかどうかを計算し、間に合うと考えられるパケットのみを再送する、リアルタイム再送方式を提案する。この方式によって、無駄な再送を極力抑えたうえでのサービス品質改善が可能になり、マルチメディアストリーミングでの再送制御が現実的なものとなる。さらに、IETF で提案されている RTP 再送プロトコルを拡張してプロトコルの設計を行い、ストリーミングシステム上に実装して評価実験を行う。実験結果から、再送時に再生時刻を考慮した再生判定を行うことで、再送成功パケット数をほとんど変えることなく、再送パケット数を大幅に減らせることを示す。

以下では、まず 2 章においてリアルタイム再送方式について説明し、3 章では IETF 提案を拡張したプロトコルの設計について述べる。4 章では、プロトコルの実装と検証結果について述べる。最後に、5 章で本論文のまとめを行う。

2. リアルタイム再送方式

1 章で述べたように、マルチメディアストリーミングでは、損失したパケットを再送しても、そのパケットが構成するフレームの再生時刻までにクライアントに到達しなければそのパケットは無駄になってしまう。さらに、マルチメディアストリーミングでは、サーバが送出するビットレートよりも実際に利用可能な帯域幅が小さいとき、ネットワークに滞留するパケットはサーバが想定する時刻よりも遅れてクライアントに到着することになる。これが累積されると、いずれ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がクライアントバッファ内に存在しない、という状況が起こりうる。このようなことを防ぐために、マルチメディアストリーミングシステムには、なんらかの QoS 制御を行うことでネットワーク帯域の変化にともなってネットワークを通過するデータのビットレートを調整しているものもある^{7),8)}。しかし、

QoS 制御によってメインのストリームにおけるビットレートが適正に保たれていても、そこへさらに再送パケットが加わることで帯域を圧迫し、その結果上記のような状況に陥る可能性がある。本章では、この問題を解決するために筆者らが提案する、リアルタイム再送方式について説明する。

2.1 再送制御の流れ

マルチメディアストリーミングでは、サーバおよびクライアントにおける処理遅延が大きいと、コンテンツのリアルタイム性が損なわれる可能性がある。リアルタイム再送方式では、サーバおよびクライアントの処理遅延を小さくするために、TCP などで用いられる ACK に基づいた再送ではなく、NACK に基づいて再送を行う。クライアントは、サーバからマルチメディアストリーミングを受信しているときにパケットの損失を検出すると、再送要求メッセージを作成しサーバへ送信する。再送要求メッセージには、損失したと考えられるパケットを示す識別子と、クライアントバッファの先頭にあるパケットが構成するフレームの再生時刻が含まれる。なお、各ストリーミングパケットが構成するフレームの再生時刻は、それぞれのパケットのヘッダに記載されているものとする。

サーバは、クライアントにストリーミングパケットを送信した後、そのパケットを再送バッファと呼ぶバッファにバッファリングしておく。サーバはクライアントから再送要求メッセージを受け取ると、その中で指定されているパケットが再送バッファ内に存在するかどうかを調べる。存在していない場合は、パケットの再送は行わない。存在していた場合、まず IETF 提案の RTP 再送と同じように、パケットの重要度を調べる。たとえば、Mpeg フォーマットコンテンツでは、単体でフレームを表示できる I ピクチャを構成するパケットは重要度が高く、差分情報で構成される P ピクチャを構成するパケットは重要度が低いと考えることができる。再送要求メッセージによって指定されているパケットの重要度が低い場合、そのパケットは再送されない。重要度が高いパケットであった場合、このパケットを再送したときに再生時刻までにクライアントに到着可能かを計算で見積もる。その結果、間に合うものと判断されたパケットのみを再送する。

クライアントでは、再送パケットを受け取ると、そのパケットの再生時刻を調べ、再生時刻よりも先に到着している場合のみ、パケットの順序が正しくなるように、クライアントバッファの中へ挿入する。再生時刻より遅く到着したパケットは、バッファリングせずに廃棄する。

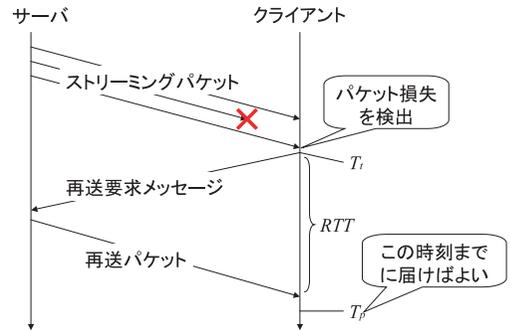


図1 パケット再送のシーケンス

Fig.1 Sequence of packet retransmission.

2.2 リアルタイム性を考慮した再送判定

本節では、2.1 節で述べた再送制御において、再送したパケットが再生時刻までにクライアントに到着するか判定する方法について述べる。このときの計算には、以下に示す変数を用いる。

- サーバ-クライアント間の RTT
サーバはクライアントとの間で交換する制御パケットを用いて、RTT を定期的に測定しているものとする。
- 再送要求メッセージ作成時にクライアントバッファの先頭にあったパケットの再生時刻
2.1 節で述べたように、この情報は再送要求メッセージに含まれている。
- 再送対象であるパケットが構成するフレームの再生時刻

この情報はストリーミングパケットのヘッダに含まれる。

これらの変数が次の条件を満たすとき、サーバはそのパケットを再送する。

$$T_p > T_t + RTT + \alpha$$

ここで、 T_p は再生対象であるパケットが構成するフレームの再生時刻を、 T_t は再生要求メッセージ作成時にクライアントバッファの先頭にあったパケットの再生時刻を、RTT はサーバ-クライアント間の RTT を示す。また、 α はサーバ、クライアントにおける処理遅延や誤差を取り扱うための変数である。 α にはたとえば、サーバおよびクライアントのバッファにおける処理遅延やネットワーク伝送による遅延などが含まれる。これらの値は状況によって変化するため、システム構築の際に試行を重ねて決定されるべきである。図 1 にパケット再送のシーケンスを示す。図 1 にもあるように、上記の条件式は再送要求メッセージが作成された時刻を起点として、再送パケットがクライアントに到着すると考えられる時刻が到着期限時刻より

も早いかどうかを調べるものである。この条件を満たすパケットのみを再送することで、再送しても無駄になってしまうパケットの再送を防ぎ、メインストリーム以外のトラヒックの増加を最低限に抑えることができる。

3. リアルタイム再送制御プロトコルの設計

筆者らは2章で述べたマルチメディアストリーミングのための再送制御を、IETFで提案されているRTP再送プロトコルを拡張することで実現する。本章では、そのプロトコルの設計について述べる。

3.1 ペイロードパケット

IETFで提案されているRTP再送プロトコル⁴⁾では、従来のRTPペイロードパケットのヘッダに、パケットの重要度を示すPRIフィールド、以前の重要度の高いパケットからのシーケンス番号の差を示すDSNフィールド、以前の重要度の高いパケットのタイムスタンプとそのパケットのタイムスタンプの差を示すDTフィールドが追加されている。リアルタイム再送制御方式においてもパケットの重要度を考慮しているため、PRIフィールドの存在は有効であるのだが、その他の追加フィールドの必要性が低いと考えられるため、ペイロードパケットヘッダには従来のRTPペイロードパケットヘッダを用いる。

3.2 再送パケット

IETFで提案されているRTP再送プロトコルでは、再送されるパケットをメインストリームのパケットと区別するために、再送のためのヘッダを改めて定義し、再送対象となるRTPパケットを再送パケットヘッダでカプセル化する方法がとられている。カプセル化を行うことで、再送されるパケットをメインのストリームに挿入し、一連のシーケンス番号を割り振ることができる。こうすることで、クライアントは再送されたパケットの損失も検出することができる。

しかし、筆者らはこの再送のための特殊なヘッダを用いず、TCPなどと同じように、メインストリームにおいてクライアントへ送信したパケットをそのまま再送する方法を採用した。この方法では再送されたパケットの損失は検出できない。しかし、リアルタイム性の高いストリーミング通信の特性を考慮すると、2度目の再送を行ってもそのパケットが再生時刻までに到着する可能性が低いと考えられることから、サーバおよびクライアントにおける処理遅延を小さくするためにこのような設計を行った。

3.3 再送要求パケット

RTCPのRR(Receiver Report)では、ロスパケッ

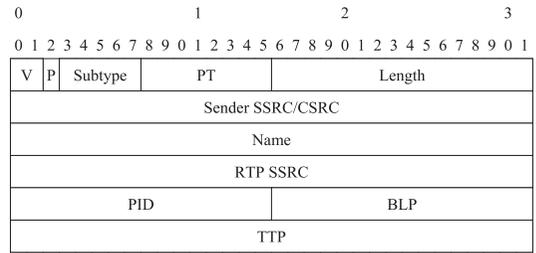


図2 再送要求パケットフォーマット
Fig. 2 Format of retransmission request packet.

ト数を記述するフィールドは定義されているものの、損失したパケットを同定する情報は記述できない。RTPでの再送を実現するためにIETFで提案されているRTP再送プロトコルでは、RTCPのAPPメッセージを用いて再送要求パケットを実現している。APPパケットはアプリケーションが自由に利用可能なパケットで、基本となるパケットヘッダのみが定義されており、それに続く部分は自由に定義可能である。本研究では、文献4)で提案されているRTP再送プロトコルにおける再送要求パケットのフォーマットに基づいて再送要求パケットを設計する。このパケットフォーマットを図2に示す。なお、RTP SSRCフィールド以下がRFC1889で定義されたAPPパケットフォーマットに追加するフィールドである。図中の各フィールドについて以下で説明する。

- Vフィールド：RTPバージョン。ここでは、V=2
- Pフィールド：パディングビット
- Subtypeフィールド：サブタイプ。Subtype=1(再送要求)
- PTフィールド：RTCPパケットタイプ。PT = APP = 204
- Lengthフィールド：パケット長
- Sender SSRC/CSRCフィールド：パケット送信元のSSRCもしくはCSRC
- Nameフィールド：APPパケット名。Name="RETX"。
- RTP SSRCフィールド：RTPパケットのSSRC。文献4)における再送要求パケットに新たに追加するフィールド。セッションレベルでの宛先同定を行うために追加している。
- PIDフィールド：損失したパケットのシーケンス番号
- BLPフィールド：PIDで指定されたパケットの後続16パケットの損失を示すためのフラグとして用いる。最上位ビットがPID+16，最下位ビットがPID+1のRTPパケットに対応し、パケットが損失した場

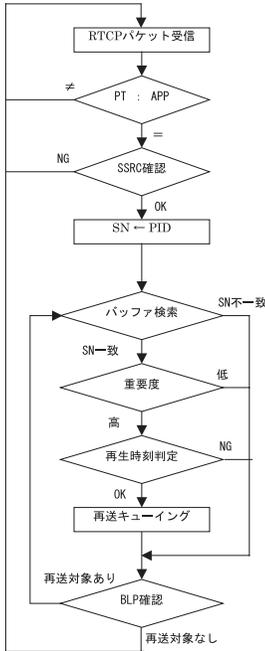


図3 再送処理の流れ

Fig. 3 Flow chart of retransmission control.

合, そのパケットに対応するビットに 1 をセットする.

TTP フィールド: クライアントバッファの先頭パケットの RTP タイムスタンプ. 文献 4) における再送要求パケットに新たに追加するフィールド.

クライアントにおいて, パケットの損失は到着するパケットのシーケンス番号の不連続として検出される. クライアントはパケットの損失を検出すると, すぐに再送要求パケットを作成する. このとき, 連続して損失したパケットが複数であった場合に BLP フィールドを用いる.

3.4 サーバの動作

再送要求パケット受信時のサーバの動作フローを図 3 に示す. サーバでは再送要求パケットの PID フィールドおよび BLP フィールドで指定されるすべてのパケットに対して, 重要度と再送時刻の判定を行い, 2 つの条件を満たすパケットのみを再送バッファにキューイングする.

4. リアルタイム再送制御方式の実装と評価

本論文で, 提案するリアルタイム再送制御方式をストリーミングシステム上に実装する. ストリーミングシステムには, 筆者らが研究を進めているモバイルマルチメディア QoS 制御システム^{9)~11)}を用いる.

4.1 モバイルマルチメディア QoS 制御システムへの実装

筆者らが研究を進めるモバイルマルチメディア QoS 制御システムでは, サーバとクライアントは 1 対 1 で通信を行う. クライアントはサーバの提供する HTML ページにアクセスし, ページ上で視聴したいコンテンツを選択する. サーバは RTSP¹²⁾を用いてコンテンツ配信のためのセッション制御情報をクライアントと交換し, その後 RTP を用いてコンテンツ配信を開始する. クライアントはサーバからコンテンツを受信しながら, 定期的に RTCP の RR を用いて, 受信した最大のシーケンス番号とロスパケット数をサーバへ通知する. サーバではこれらの情報とは別に, RTCP の SR (Sender Report) と RR を用いて RTT を測定している. サーバは以上の情報に基づき, 送出量をネットワーク帯域幅の変動に応じて変化させる^{10),11)}. ただし本論文で行う実験では, 再送による効果を明らかにするために送出量制御を行わない. サーバがクライアントに配信する映像は Mpeg-4 ファイルを, 音声/音楽は G.723 ファイル, MP3 ファイルおよび AAC ファイルを利用できる. なお, サーバ, クライアントともに Windows2000 をプラットフォームとしている.

リアルタイム再送方式はこのシステム上に, Visual C++6.0 を用いて実装した.

4.2 実験結果

提案方式の有効性を検証するために, 4.1 節で述べたシステムを用いて評価実験を行う. 以下では, 実験の条件と結果, および結果に対する考察を述べる.

4.2.1 実験の条件

筆者らが行った実験では, ネットワークの遅延やロス率を自由に設定するためにネットワークエミュレータ NE3000¹³⁾を用いる. 再送制御を実装したモバイルマルチメディア QoS 制御システムのサーバとクライアントの間にネットワークエミュレータを設置し, 平均遅延を 50 ~ 250 ms, 平均パケットロス率を 1 ~ 10% まで変化させる. 実験時にはモバイルマルチメディア QoS 制御システムのレート制御機能を OFF とした. サーバがストリーミング配信するコンテンツは 90 秒の長さで映像が 400 Kbps の Mpeg4 フォーマット, 音声は 8 Kbps の G.723 フォーマットのものを用いた. 映像は 15 fps, GOP が IPPPP から成る. この結果, 映像, 音声を合わせてコンテンツ開始から終了までに 7,202 個のパケットがサーバからクライアントへ送信される. クライアントはコンテンツ再生の安定化のために RTP パケットを受信してから 2 秒間バッファリングした後に再生を行う. 以上のような条件のもとで,

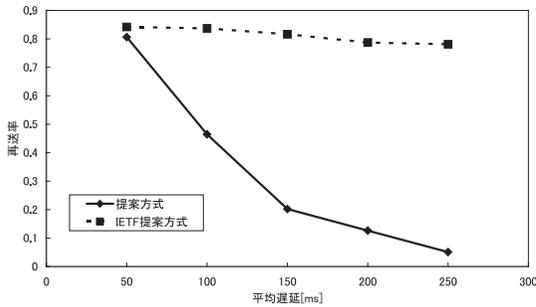


図4 パケットロス率1%における再送率

Fig. 4 Retransmission rate on 1% packet loss rate.

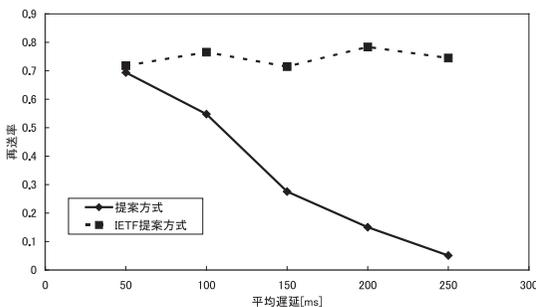


図5 パケットロス率5%における再送率

Fig. 5 Retransmission rate on 5% packet loss rate.

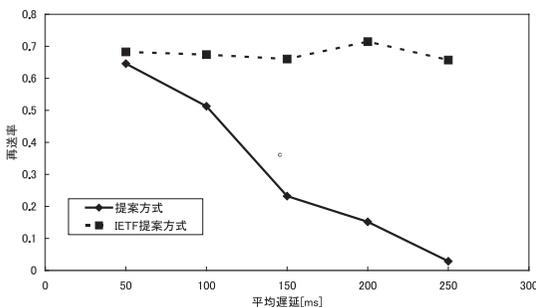


図6 パケットロス率10%における再送率

Fig. 6 Retransmission rate on 10% packet loss rate.

パケットの重要度判定のみを行う IETF 提案の方式と筆者らの提案するリアルタイム再送制御方式とを比較する。なお、リアルタイム再送制御方式で用いる計算式の α の値は 0 に設定する。

4.2.2 実験結果

[再送回数について]

提案方式によって再送されるパケット数が減少する数を調べるために、IETF 提案と筆者らの提案方式を比較する。その結果を図 4, 図 5, 図 6 に示す。これらの図はあるパケットロス率のときにネットワークの平均遅延を 50 ms から 200 ms まで変化させたときの、再送されたパケット数とロスパケット数との比(以下、再送率)をとったものである。横軸にネットワークの平

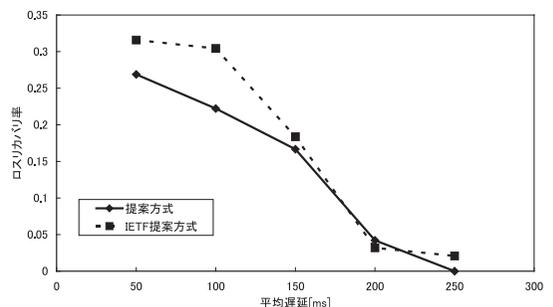


図7 パケットロス率1%におけるロスリカバリ率

Fig. 7 Recovery rate of lost packets on 1% packet loss rate.

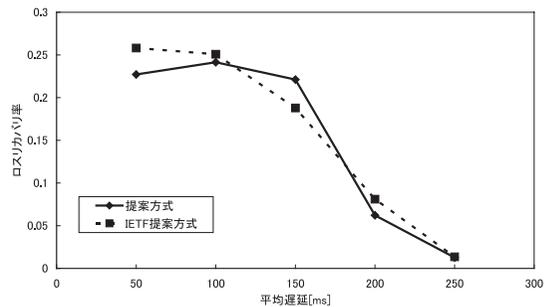


図8 パケットロス率5%におけるロスリカバリ率

Fig. 8 Recovery rate of lost packets on 5% packet loss rate.

均遅延、縦軸に再送率をとったものである。それぞれ、図 4 はパケットロス率が 1% のとき、図 5 はパケットロス率が 5% のとき、図 6 はパケットロス率が 10% のときのグラフである。これらのグラフから、IETF 提案ではパケットの重要度のみを調べて再送を行うため、遅延が大きくなって再送率がほとんど変化しない一方、提案方式ではネットワークの平均遅延が大きくなると再送されたパケットが再生時刻までに到着する可能性が小さくなるため、平均遅延が大きくなるとパケット再送率が小さくなるのが分かる。ネットワークの平均遅延が小さいときは IETF 提案と筆者らの提案方式とはほとんど同じ再送率であるが、特にネットワークの平均遅延が大きい場合に筆者らの提案方式は再送率を大きく低減できる。

[ロスリカバリ率について]

パケットの再送がどれだけ有効に機能しているかを調べるために、再送の結果再生時刻までに到着したパケット数(以下、再送成功数)とロスパケット数の比をロスリカバリ率とし、その値について IETF 提案と筆者らの提案する方式を比較した。その結果を図 7, 図 8, 図 9 に示す。それぞれのグラフでは、横軸はネットワークの平均遅延、縦軸はロスリカバリ率を示してい

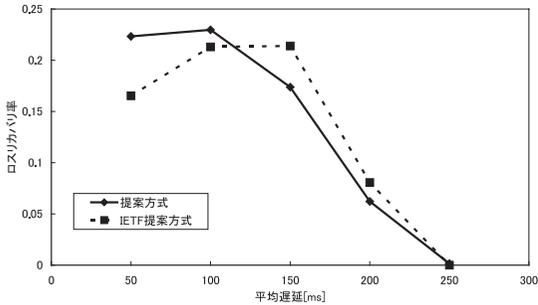


図9 パケットロス率 10%におけるロスリカバリ率

Fig. 9 Recovery rate of lost packets on 10% packet loss rate.

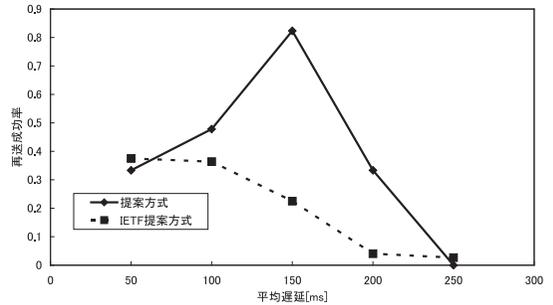


図10 パケットロス率 1%における再送成功率

Fig. 10 Successful retransmission rate on 1% packet loss rate.

る．これらの3つのグラフは順に、パケットロス率が1%、5%、10%のときの実験結果である．IETF提案では、サーバはパケットの重要度判定のみを行ってパケットを再送する．図7～9のグラフからIETF提案ではネットワークの平均遅延が大きくなるとロスリカバリ率が小さくなること分かる．図4～6の結果より、IETF提案ではネットワークの平均遅延が大きくなっても再送率がほとんど変化しないことが明らかになっている．これらのことから、IETF提案ではネットワークの平均遅延が大きくなることによって、パケットの無駄な再送回数が多くなるといえる．一方、筆者らの提案方式では、IETF提案と同様にネットワーク遅延が大きくなるに従ってロスリカバリ率が小さくなるものの、あらゆるパケットロス率の場合においてIETF提案とほとんど同じロスリカバリ率が得られることが分かる．以上より、筆者らの提案するリアルタイム再送制御方式では、ネットワークの遅延時間が長いとき、つまり再送されたパケットが再生時刻までにクライアントに到着する可能性の低いときに、再送機能の性能をほとんど劣化させることなく、再送されるパケット数を大きく低減できることが分かる．

[再送成功率について]

再送されたパケットの有効性を検証するために再生時刻までにクライアントに到着した再送パケット数とサーバが再送した全パケット数の比(以下、再送成功率)についてIETF提案とリアルタイム再送制御方式を比較した．その結果を図10、図11、図12に示す．これらの図も順にパケットロス率が1%、5%、10%の場合を示す．横軸にはネットワークの平均遅延、縦軸には再送成功率をとる．これらのグラフから、まずIETF提案ではネットワークの遅延時間が大きくなるに従って再送成功率が劣化することが分かる．これは図4～6においてIETF提案では再送率がネットワークの遅延にかかわらずほぼ一定であること、および図

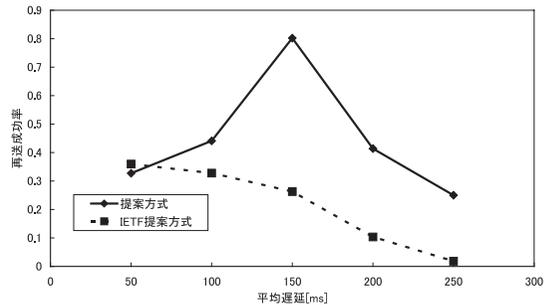


図11 パケットロス率 5%における再送成功率

Fig. 11 Successful retransmission rate on 5% packet loss rate.

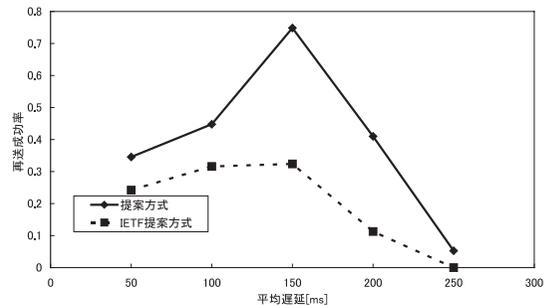


図12 パケットロス率 10%における再送成功率

Fig. 12 Successful retransmission rate on 10% packet loss rate.

7～9においてIETF提案ではネットワーク遅延が大きくなるに従ってロスリカバリ率が小さくなることから自明である．

平均遅延が50msの場合、図10、11ではリアルタイム再送制御方式とIETF提案はほとんど同じ値をとっている．これは、2方式で再送されるパケット数がほとんど同じであるためであるが、遅延が小さいうえにロスパケット数もそこまで大きくないことも要因として考えられる．ちなみにリアルタイム再送制御方式がIETF提案よりも悪い再送成功率を示している

が、これは再生時刻までに間に合わないと判断されたにもかかわらず、実際は間に合ったパケットが存在したためであると考えられる。一方、図 12 では平均遅延が同じ 50 ms の場合でも提案方式の方が明らかに良い結果を示している。これは IETF 提案ではパケットロス率が高いためにパケットの再送が大量に行われることでネットワーク遅延が大きくなり、その結果再生までに間に合わないパケットが多く発生したものと考えられる。リアルタイム再送制御方式では図 6 でも分かるように、わずかだが再送率が低くなっている。リアルタイム再送制御方式は RTT を用いて再送の判定を行うため遅延の変化に敏感であり、このような障害が起こることを回避できているため、図 12 のような結果が得られているものと考えられる。

平均遅延が 250 ms の場合、図 10 では IETF 提案とリアルタイム再送制御方式の再送成功率がほとんど同じ値になっている。これはそもそもパケットロス率が 1% であるためにロスパケット数が少なく、そのうえ遅延が大きいために再生に間に合うパケットがほとんど存在しないためである。なお、IETF 提案の方がわずかに優れた結果を示しているのは、平均遅延が 50 ms の場合と同様の理由である。図 11 と 12 では程度の大小に差はあるものの、明らかにリアルタイム再送制御方式の方が優れている。再送しても再生に間に合わないパケットが多いことは図 10 の場合と同様だが、ロスパケット数が多いため IETF 提案の再送数が増加するのに対して、リアルタイム再送制御方式での再送数があまり変化しないためであるものと考えられる。なお図 11 よりも 12 の方がリアルタイム再送制御方式の有効性が低くなっている。これはパケットロス率が高くなることでパケットロスが連続して発生する可能性が高まり、パケットロスの検出が遅れてしまうためである。検出が遅れると再送されたパケットが再生時刻に間に合う可能性が全体的に低下してしまう。つまり再送成功率自体があまりに少ないために平均遅延 250 ms におけるリアルタイム再送制御方式の再送成功率が図 11 よりも図 12 の場合の方が低くなるものと考えられる。

また、リアルタイム再送制御方式はネットワーク遅延がある値のとき（今回の実験では 150 ms）、再送成功率が最大となることが分かる。この理由を説明するために図 13 を示す。このグラフはパケットロス率が 5% のときにリアルタイム再送制御方式を用いた場合、遅延の変化にともなうパケット再送数と再送成功率の変化を示したものである。このグラフの横軸はネットワークの平均遅延、縦軸はパケット数を表して

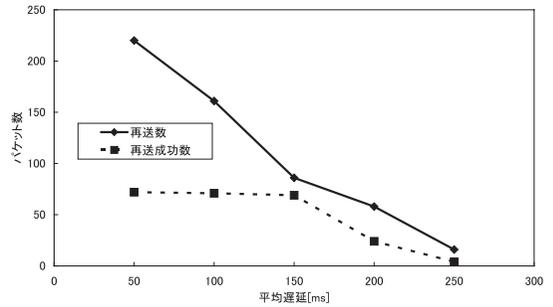


図 13 パケットロス率 5% における再送数と再送成功率

Fig. 13 The number of retransmission packets and successful retransmission packets on 5% packet loss rate.

いる。このグラフから、平均遅延が 150 ms 以下では再送成功率がほとんど変化していないことが分かる。これは平均遅延が 150 ms の場合、損失したパケットのほとんどの再生が成功するためであるものと考えられる。一方、平均遅延が 150 ms を超えると再送成功率は減少を始める。これらのことから、次のように考えられる。

- 150 ms 以下のときは再送成功率に対して再送数が多いために良い再送成功率が得られない。
- 150 ms 以上では再送成功率が少なくなるために良い再送成功率が得られない。

これらの理由によりリアルタイム再送制御方式の平均遅延が 150 ms のときに再送成功率が最高になるものと考えられる。なお、最高の再送成功率が得られる遅延の値はクライアントのバッファ量によって変化する。たとえば今回の実験ではクライアントがコンテンツの再生を開始する際のバッファリング時間を 2 秒と設定したが、この値が長くなれば再送成功率が最高となるネットワークの遅延時間も変化する。これは逆に、ネットワークの遅延時間が分かれば、リアルタイム再生制御方式を有効に活用できるバッファリング時間を決定できることを示している。またこのグラフから、遅延の値が小さい場合はリアルタイム再送制御方式は多数の再生に間に合わないパケットを再送してしまっていることが分かる。これはネットワーク遅延が小さいためにサーバ、クライアントにおける処理遅延が無視できなくなっているためであると考えられる。つまり本来なら α に適正値を代入すべきところを 0 に設定しているために提案方式が正常に動作していないものと考えられる。実際に運営する場合は、テストによって α の適正値を発見する必要がある。

[再送数と再送成功率の比較について]

最後に、ネットワークの平均遅延を 200 ms に固定

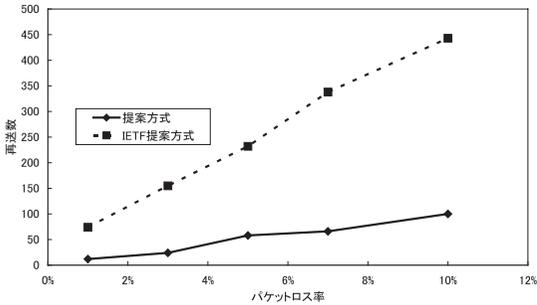


図 14 平均遅延 200 ms における再送数

Fig. 14 The number of retransmission packets on 200 ms latency.

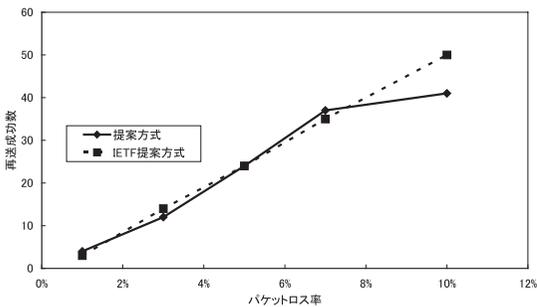


図 15 平均遅延 200 ms における再送成功数

Fig. 15 The number of successful retransmission packets on 200 ms latency.

し、パケットロス率の変化にともなう再送数と再送成功数の変化を IETF 提案とリアルタイム再送制御方式とで比較する。実験結果を図 14、図 15 に示す。図 14 は再送数の比較、図 15 は再送成功数の比較である。横軸はパケットロス率、縦軸はそれぞれ再送数と再送成功数を示している。この 2 つのグラフから、リアルタイム再送制御方式はパケットロス率が変化してもほとんど再送数が変化していないにもかかわらず、再送されるパケット数を大きく減少できることが分かる。特にパケットロス率が高い場合、約 9 倍程度まで再送数を低減できる。これによってリアルタイム再送制御方式はストリーミング配信での再送によって発生するトラフィックを大幅に抑え、再送によってメインのストリーミングの通信性能が低下することを防ぐことができる。ただし、図 15 を見ても分かるようにパケットロス率が高い場合には、判定の誤りにより再送されるべきパケットが再送されない場合がある。先に述べたように、このような判定誤りについては α の値の設定によって対処可能であるものと考えるが、その決定方法は今後の重要な課題の 1 つである。

5. まとめ

本論文では、マルチメディアストリーミングのためのリアルタイム再送方式を設計し、筆者らが研究をすすめるモバイルマルチメディア QoS 制御システム上に実装した。さらに、筆者らの構築した実験環境において検証実験を行い、提案方式が従来方式と同等の再送性能を満たしたうえで再送パケット数を大きく低減できることを示した。今後は、遅延が小さいネットワークにおける再送成功率の向上について検討をすすめる予定である。

参考文献

- 1) 澤田 寛, 有馬秀平: IMT-2000 ネットワークアーキテクチャ, 電子情報通信学会誌, Vol.82, No.2, pp.145-152 (1999).
- 2) 高木雅裕, 加藤紀康, 渋谷尚久, 森谷 修, 坂本岳文: ハイブリッド MMAC 実験システムにおけるネットワーク機能, 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 (June 2001).
- 3) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Streaming Protocol, RFC1889, IETF (Jan. 1996).
- 4) Miyazaki, A., Fukushima, H., Hata, K., Wiebke, T., Hakenberg, R., Burmeister, C., Takatori, N., Okumura, S. and Ohno, T.: RTP Payload Formats to Enable Multiple Selective Retransmissions, draft-ietf-avt-rtp-selret-03.txt, IETF (May 2002).
- 5) Ott, J., Wenger, S., Sato, N., Burmeister, C. and Rey, J.: Extended RTP Profile for RTCP-based Feedback (RTP/AVPF), draft-ietf-avt-rtp-feedback-07.txt, IETF (June 2003).
- 6) Rey, J., Leon, D., Miyazaki, A., Varsa, V. and Hakenberg, R.: RTP Retransmission Payload Format, draft-ietf-avt-rtp-retransmission-09.txt, IETF (Aug. 2003).
- 7) Aurrecochea, C. Campbell, A.T. and Hauw, L.: A survey of QoS architectures, *Multimedia Systems*, Vol.6, pp.138-151 (1998).
- 8) Chalmers, D. and Sloman, M.: A Survey of Quality of Service in Mobile Computing Environments, *IEEE Communications Surveys*, Second Quarter 1999, pp.2-10 (1999).
- 9) 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).

- 10) 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 (Sept. 2001).
- 11) 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: 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, 電子情報通信学会技術研究報告 SST2001-141, pp.91-98 (Mar. 2002).
- 12) Schulzrinne, H., Rao, A. and Lanphier, R.: Real Time Streaming Protocol, RFC2326, IETF (April 1998).
- 13) NTT エレクトロニクス ネットワークエミュレータ, <http://www.nel.co.jp/product/emulator/>
(平成 15 年 4 月 16 日受付)
(平成 15 年 9 月 5 日採録)



萩野 浩明 (正会員)

平成 13 年大阪大学大学院工学研究科博士後期課程修了。同年 (株) NTT ドコモ入社, 現在に至る。主にモバイル環境下での通信プロトコル, マルチメディアストリーミング, およびユビキタスコンピューティングの研究に従事。工学博士。



宮崎雄一朗

平成 14 年中央大学大学院理工学部情報工学修士課程修了。同年 (株) NTT ドコモ入社, 現在に至る。主に, モバイルマルチメディア通信プロトコルの研究に従事。



尾上 裕子 (正会員)

平成 3 年慶應義塾大学大学院理工学研究科数理科学科修士課程修了。同年 NTT 入社。現在 (株) NTT ドコモネットワーク研究所主任研究員。ユビキタスネットワーク技術の研究に従事。工学博士。電子情報通信学会会員。



渥美 幸雄 (正会員)

昭和 50 年慶應義塾大学工学部電気工学科卒業。昭和 52 年同大学大学院修士課程修了。同年電電公社 (現, NTT) 横須賀電気通信研究所入社。主に通信プロトコル, 通信制御ソフトウェアの研究開発に従事。平成 6 年より (株) 超高速ネットワーク・コンピュータ技術研究所。平成 11 年より (株) NTT ドコモマルチメディア研究所。平成 14 年より専修大経営学部教授。博士 (情報工学)。電子情報通信学会会員。



谷口 雅昭 (正会員)

平成 7 年 4 月, 日本アイ・ピー・エム (株) 冬季用基礎研究所入社, モバイル・マルチメディアシステムに関する研究・開発に従事。



駒木 寛隆

東京大学大学院工学系研究科電子情報工学専攻修士。日本アイ・ピー・エム (株) 入社後, 同社東京基礎研究所にてモバイル・マルチメディアシステムに関する研究・開発に従事。現在は WEB アプリケーションのプログラミングモデルに関する研究・開発活動を行っている。



山内 長承 (正会員)

昭和 50 年東京大学工学部電子工学科卒業。昭和 58 年同大学大学院情報工学専門課程中退。昭和 53 年~59 年スタンフォード大学大学院在学。昭和 59 年~平成 12 年日本アイ・ピー・エム (株) 勤務。平成 14 年より東邦大学理学部情報科学科教授。主として, OS, 並列プログラムの検証, 計算機ネットワークの応用の研究開発に従事。工学博士。ACM, IEEE, 日本ソフトウェア科学会各会員。