

## QoS保証されたインターネット放送のための前方誤り訂正

古村 隆明

藤川 賢治

池田 克夫

京都大学大学院情報学研究所

QoS保証されたインターネット上の経路で前方誤り訂正を利用してデータ誤り率を下げる方法について検討した。アプリケーションが要求するデータ誤り率に合わせて誤り訂正符号の強度を変化させて、目的の誤り率を達成する方法を提案する。また、Comfortable Serviceとして提案されているキューイング方式を用いて伝送品質を向上するためにルータでのパケットロス率を下げる方法と、誤り訂正符号でデータの誤り率を下げる方法とのネットワーク帯域の利用効率について比較する。

### Forward Error Correction for QoS-Reserved Internet Broadcasting

KOMURA TAKAAKI

FUJIKAWA KENJI

IKEDA KATSUO

Graduate School of Informatics, Kyoto University

We examined a method to lower error rate of data with Forward Error Correcting(FEC) on QoS-reserved Internet. We propose a method which can change a power of FEC in accordance with error rate requested by application programs. We also compared a method to lower packet loss rate by routers with a method by FEC in order to improve transmission quality in terms of efficiency of network bandwidth, when the queuing method named "Comfortable Service" is used.

#### 1. はじめに

音声・動画データなどをマルチキャストで送信し、受信ホストでは受信と同時に再生を行う環境で、パケットロスなどの原因によるデータ誤りに対処するために、前方誤り訂正 (Forward Error Correcting; FEC) を利用することが有効である。

本稿では、マルチキャストを行うネットワーク上の経路は QoS 保証によって帯域、遅延、パケットロス率が保証されているものとする。パケットロス率を 0 に近づけることはできるが、0 にすることは不可能である。また、ある程度パケットロスを許すことで、QoS を保証するルータの設計が容易になるという研究結果もある。

2 章で、本稿で想定しているインターネット上での音声・動画配信の環境について述べ、3 章で既存の研究の問題点を指摘し、4 章で QoS 保証された経路での利用を前提とした FEC 符号を利用した送受信ライブラリを提案する。5 章では、

Comfortable Service(CS)<sup>2)</sup>として提案されているキューイング方式を用いる場合に、ルータでパケットロス率を下げる方法と、FECによりデータ誤りを訂正する方法で、どちらがネットワークの利用効率を高めることができるかを比較する。6 章で、実装したライブラリについて紹介し、実測した性能について論じる。

#### 2. インターネット放送

インターネット放送とは、IP マルチキャストを利用したインターネット上での音声・動画などのデータ送信のことを指す。

データに誤りがあると音声や画像に乱れが発生するが、この乱れが利用者の許容範囲内であれば必ずしも誤ったデータを訂正する必要はない。

受信者の許容範囲を越えた乱れが発生する場合はデータ誤りを訂正することが望ましい。しかし、データの受信後ただちに再生を行なうアプリケーションでは、データの再送によって誤りを訂正し

ようとするとデータの到着が再生に間に合わない可能性があるため、FECを利用してデータを訂正する。

ここで、本稿ではデータ誤りはパケットロス<sup>\*</sup>によってのみ発生するものとし、受信ホストが受信したパケット内にはデータ誤りは無いものとする。

### 3. 既存の研究

#### 3.1 FEC専用のフローを用いる方法

RTP(Real-Time Transport Protocol) パケット<sup>4)</sup>に対してFEC符号を付加する場合のパケットフォーマットがRFC 2733<sup>3)</sup>で提案されている。この方式では、通常のRTPパケットからFEC用のRTPパケットを生成して、それぞれを別のフローで送信する。FEC用のパケットを別フローで扱うため、FECに対応していないアプリケーションでも正常に動作することが期待できる。

#### 3.2 IPパケットのフラグメント処理を利用する方法

IPv6のパケットフラグメント処理<sup>1)</sup>を拡張して、フラグメントと同時にFEC用のパケットも送信する方法<sup>5)</sup>が提案されている。この手法を用いると、送信アプリケーションはFEC用のパケットやMTUサイズを気にすることなく一つの大きなサイズのデータを送信すれば良い。送信しようとするデータがMTUサイズに収まり切らない場合、データはフラグメント処理で複数のIPパケットに分割される。提案手法では、フラグメント処理と同時にFEC用のパケットを付加する処理が追加されている。

#### 3.3 既存研究の問題点

3.1の方法ではFEC符号にパリティを用いている。より強力な誤り訂正能力を持つReed Solomon符号を利用することができない。

また、3.1、3.2の両方法とも、

- FECを付加した後のデータ量がどの程度になるか分からない
- 最終的な誤り率をどの程度にまで抑えられるかが分からない

という問題がある。

本研究ではデータを流す経路の帯域をあらかじめ予約する方法を考えているので、ネットワークに流れるデータ量がFECを利用することによっていくら増加するかをアプリケーションがあらかじめ知っていなければならない。

また、既存研究では、誤り率を小さくすることはできるが、最終的な誤り率をいくらにしたいのかは考えずにFECを利用している。本来は、利用者やアプリケーションが要求する誤り率を満たせるようにFECを利用することが重要なはずである。

本稿で提案する手法でこれらの問題を解決する。

### 4. QoS保証経路での前方誤り訂正

#### 4.1 Reed Solomon符号による前方誤り訂正

FEC符号として、Reed Solomon(RS)符号を採用する。RS符号は、符号語長を $n$ 、情報ビット長を $k$ とすると、ビット誤り位置が既知の場合には $n-k$ ビットの誤りを訂正する能力を持つ。

本研究では、データ誤りはパケットロスによってのみ発生するものと仮定しているため、図1のように元データの格納されているパケット(これをメディアパケットと呼ぶ)の同じ位置に並ぶデータから誤り訂正符号を算出してFEC用のパケット(これをFECパケットと呼ぶ)の同じ位置に格納する。

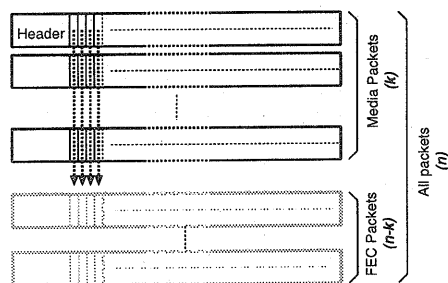


図1 RS符号によるFECパケットの付加

$k$ 個のメディアパケットに対し、 $n-k$ 個のFECパケットを付加し、全体で $n$ 個のパケットを送信する。この場合、送信された $n$ 個のパケットのうち、任意の $n-k$ 個以下のパケットを受信できなくとも、誤り訂正によって正しいデータを得ることができる。

<sup>\*</sup>パケットロスは、ルータのキューが溢れた場合や、伝送路でのエラーによって発生する。伝送路でエラーが発生したパケットはルータが破棄する。

#### 4.2 データ誤り率

マルチキャストを行う経路は、QoS 保証によって帯域、遅延、パケットロス率が既知であるとする。

インターネット放送では、電話音質程度の低品質な音声放送から、映画のような高品質な動画放送まで、様々な応用が考えられる。また、それぞれの応用ごとに、要求されるデータの誤り率も変化すると考えられる。そこで、目的のデータ誤り率に応じて、FEC 符号の強度を変化させて対処する。

3章で指摘したように、アプリケーションプログラムにとって必要なのは、単に誤り率を下げるのではなく、誤り率を一定値以下にすることであると考えられる。本稿での提案は、アプリケーションプログラムが必要としているデータ誤り率に合わせて、誤り訂正符号の強度を設定することに特徴がある。

以下では、目標とする誤り率を得るために、どの程度の強度の RS 符号を使えば良いかを決定する方法を述べる。

提案方式の誤り訂正符号では、 $n$  個のパケットのうち  $n-k$  個より多くのパケットロスが発生した場合に、正しいデータを得ることができなくなる。この、データを訂正できない確率を  $E$  とすると、

$$E = \sum_{i=n-k+1}^n n C_i e^i (1-e)^{n-i} \quad (1)$$

と表わされる。ここで、 $e$  は送受信ホスト間でのパケットロス率を表わす。

ここで、 $e$  と  $k$  は既知の値であるから、 $E$  は  $n$  の値によって決定される。何パターンかの  $n$  について  $E$  を計算すれば、目的の誤り率を達成できる  $n$  を求めることができる。この  $n$  で RS 符号の強度が決まる。

また、 $n$  から実際に送信するデータ量も決まるため、QoS 保証の際にいくら帯域を予約すれば良いかを、アプリケーションプログラムが知ることもできる。

#### 5. ネットワーク帯域利用効率の比較

Comfortable Service(CS)では、ルータが転送可能な全帯域の一部を QoS 保証のために利用す

る。ここで、全帯域に占める QoS 保証用の帯域の割合を  $b$  とする。CS では、ルータのキュー長と  $b$  に依存して、ルータでのパケットロス率が決まることが知られている。 $b$  を小さくすることでパケットロス率を下げるができるが、QoS 保証できる帯域が小さくなる。

また、FEC でデータ誤り率を下げる場合も、メディアパケット以外に FEC パケットを送信するため、利用可能な帯域に占める実データの割合が下がることになる。

本章では  $b$  を小さくしてパケットロス率を下げる方法と、FEC を用いてパケットロス率を下げる方法の二つについて、ネットワークの利用効率を比較する。

図 2 の二つのグラフはどちらも、ルータのキュー長が 25 の場合について、 $b$  が 0.5, 0.6, 0.7, 0.8, 0.9 の五つの場合についてのパケットロス率を表わしている。

グラフは、縦軸が全帯域に占める実データの転送に利用できる帯域の割合を表し、横軸が 100 段のルータを通過したときのパケットロス率 (右側ほどロス率が小さい) を表している。

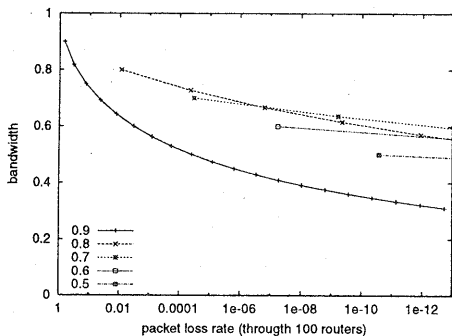
各折線はそれぞれ異なった  $b$  についての、帯域利用率とパケットロス率の推移を表わしている。折線の左端の点は FEC を用いなかった場合の値で、右側に延びる部分は FEC を適用して誤り率を下げた場合の値になっている。

本稿で提案している FEC 符号では、より多くのパケットに対して FEC を利用するほど帯域の利用効率上がる。図 2(a) は FEC を適用するパケット数が 10 個の場合、図 2(b) は 100 個の場合についての結果である。

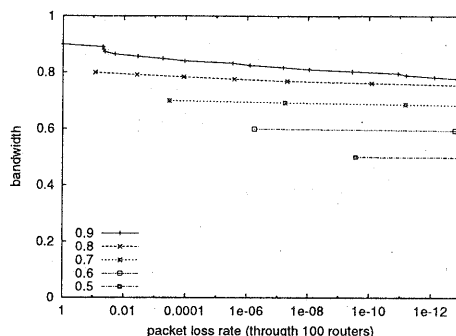
実時間通信では、低画質な動画であれば一度に FEC を付加するパケットは 10 個程度、高品質な動画であれば 100 個以上に適用することが考えられる。

図 2(b) から、パケット数が多い場合は FEC 付加による帯域利用率の低下はほとんどないまま誤り率を小さく抑えることが可能であることが分かる。

図 2(a) からは、パケット数が少ない場合は FEC による帯域利用率の低下も幾分見られる。しかし例えば、誤り率を  $10^{-10}$  程度に抑えるためには、FEC を利用しない場合には帯域利用率を 0.5 程



(a) パケット数 10



(b) パケット数 100

図 2 キュー長 25 のときの帯域利用率とデータ誤り率

度にも下げる必要があるが、 $b$  を 0.7 ~ 0.8 程度に設定したうえで FEC を併用すれば帯域利用率を 0.6 程度に高めることが可能であることが分かる。

これらの結果から、CS 対応のルータだけでパケットロス率を下げる方法よりも、FEC を併用してデータ誤り率を下げる方法の方が、ネットワークの利用効率を高めることができることが示された。

## 6. 実 装

RS 符号を用いた FEC 符号を付加するライブラリを実装した。

送信アプリケーションプログラムからは、{一度に処理するパケット数、ネットワークのデータ誤り率、達成したい誤り率} の三つのパラメータを渡すと、式 (1) から「達成したい誤り率」を満たせる  $n$  を求めて送信側のライブラリを初期化し、アプリケーションに  $n$  を返す。この  $n$  は、アプリケーションが帯域予約を行う場合に必要となる。

ライブラリの初期化が終われば、アプリケーションは FEC ライブラリに実データを渡すだけで、自動的に FEC パケットが付加されてネットワークに送信される。

受信ホストでは、FEC 符号の強度などを表わすパラメータを受信側ライブラリに与えると、受信バッファ等が初期化され受信を開始する。

アプリケーションがライブラリに対して読み込みを要求すると、バッファ内からデータが取り出

される。この時、パケットロスがあれば誤り訂正が自動的に行われる。

ライブラリで FEC 符号の付加と、誤り訂正にかかる時間を計測した。\*

1280byte のパケット 10 個に対して FEC パケットを付加して、 $10^{-3}$  であるパケットロス率を  $10^{-9}$  まで下げる場合で測定を行なった。この場合、10 個のメディアパケットに対し 3 個の FEC パケットが付加される。このデータ量は、画像サイズが  $320 \times 240$  程度の圧縮画像を想定して決定した。

このときの FEC パケットを生成するための演算時間は 0.005 秒以下となった。誤り訂正のための演算にかかる時間は、パケットロスが 3 個のとき 0.14 秒以下、2 個のとき 0.11 秒以下、1 個のとき 0.09 秒以下となった。

このような画像データ (メディアパケット 10 個 + FEC パケット 3 個) を、秒間 30 フレーム分送受信する環境\*\*を考えると、誤り訂正に時間がかかるため、パケットロスが頻発すると秒間 30 フレームを表示するのは難しくなる。しかし、ネットワークでのパケットロス率は  $10^{-3}$  を想定しているため、実際に誤り訂正を行うのは 2.5 秒

\* 計測には PentiumIII 600MHz の機材を使用した。

\*\* このような環境でパケットロス率が  $10^{-3}$  で前方誤り訂正を行わない場合は約 2.5 秒に一度の確率で画面全体の 1/10 のデータが正しく復号できなくなる。誤り率を  $10^{-9}$  まで下げた場合は、数十日に一度しかエラーが発生しなくなる。

以上に一回になる。理論上は、この実験を行なった環境でも、毎秒 30 フレーム分のデータの誤り訂正を行うことが可能である。

## 7. おわりに

本稿では、QoS 保証された経路での利用を前提とした、前方誤り訂正について考察と提案を行なった。

アプリケーションプログラムが必要とする誤り率を指定して、誤り訂正符号の強度を自動的に設定するライブラリを提案し実装した。誤り訂正に符号よって増加するデータ量も分かるため、帯域を予約する場合にも利用可能である。

また、ルータでのパケットロス率を下げる方法と、誤り訂正でデータ誤り率を下げる方法について、ネットワークの利用効率を比較した。誤り訂正符号を用いることで、わずかにデータ量を増加させることで誤り率を非常に小さくすることが可能であり、多くの場合でルータで誤り率を下げるよりもネットワークの利用効率を高められることが示された。

実装を行なったライブラリについて紹介し、処理時間を測定した。実験に用いたデータならばリアルタイムに動画を再生することが可能である。データ量が増加すると、実験に用いた機器での処理では間に合わなくなるが、計算機の処理能力の向上や、演算ルーチンの最適化などで対処できると考えられる。

## 参 考 文 献

- 1) Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification [RFC2460] (1998).
  - 2) Kenji, F., Yoshito, F., Katsuo, I., Norio, M. and Masataka, O.: Comfortable Service: A New Type of Integrated Services Based on Policed Priority Queuing, 情報処理学会研究報告書, Vol. 2000-DPS-98, pp. 25-30 (2000).
  - 3) Rosenberg, J. and Schulzrinne, H.: An RTP Payload Format for Generic Forward Error Correction [RFC 2733] (1999).
  - 4) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications [RFC1889] (1996).
- 5) 米山清二郎, 垣内正年, 砂原秀樹: Reliable Multicast - Using FEC on IP version 6. IC'99 発表資料, December 1999.