

# FEC を用いた MPEG2 over IP システムの開発と評価

大塚玉記<sup>†</sup> 西村浩二<sup>‡</sup> 相原玲二<sup>‡</sup> 前田香織<sup>††</sup>

<sup>†</sup> 広島大学大学院工学研究科

<sup>‡</sup> 広島大学情報メディア教育研究センター

<sup>††</sup> 広島市立大学情報処理センター

近年の高速通信網の普及により動画伝送の本格的利用が期待されている。インターネット上では今後、より高品質の動画伝送が増加し、ネットワーク上のトラフィックのほとんどは動画になるとも言われている。一方、マルチメディアデータ転送では輻輳等によるパケット損失や遅延対策を考慮する必要があるが、一般的にこれらを回避することは不可能である。そこで本稿では我々の開発した IPv6 マルチキャストに対応した MPEG2 伝送システムにインターネット上のデータ転送で生じるパケット損失を回復する技術、前方誤り訂正 (FEC: Forward Error Correction) 機能の実装を行なった。実測評価を行なうとともにパケット損失回復の有効性と実用に耐えるものであることを示す。

## Development and Evaluation of an MPEG2 over IP Transfer System using FEC

Tamaki OTSUKA<sup>†</sup>, Kouji NISHIMURA<sup>‡</sup>,  
Reiji AIBARA<sup>‡</sup> and Kaori MAEDA<sup>††</sup>

<sup>†</sup> Graduate School of Engineering, Hiroshima University

<sup>‡</sup> Information Media Center, Hiroshima University

<sup>††</sup> Information Processing Center, Hiroshima City University

In recent years, motion picture increases the Internet traffic because of the wide spread deployment of high-speed applications over the Internet. It is expected that most of the Internet traffic will encompass motion picture in the near future. On the other hand, the problems of packet loss and delay that occur due to collision have much negative effects on the quality of pictures. Even though it is generally difficult to avoid these problems, we need to consider how to minimize the effects of such problems. To address these problems, we design and implement Forward Error Correction mechanism, and integrate it to our previous developed MPEG2 over IP Transfer System. In this paper, we present the evaluation results of the performance of our system.

### 1 はじめに

近年インターネット上を流れる情報には TV 会議や映像、音声を含むマルチメディア情報がその多くを占めるようになってきた。リアルタイムにマルチメディアデータを再生、転送するアプリケーションもさまざまであるが、これまでは通信帯域の制限により低品質の動画が主に利用されていた。しかし最近では通信帯域も拡大し今後高速通信網の普及、整備が進むとともによりいっそう高品質動画の伝送の期待が高まってくると考えられる。そのような中で画像圧縮方式である MPEG2 はデジタル放送にも広く採用されるなど標準として用いられる傾向にもある。さらに高品質、高圧縮率という特徴がある。高品質を提供する画像圧縮方式として MPEG2 のほか DV 方式もあるがフレーム間圧縮を行

なう MPEG2 方式と行なわない DV 方式では圧縮率の点で大きな違いがある。また通常テレビ品質 (SDTV) で比較するとフレーム圧縮を行なう MPEG2 方式の方が 0.1~0.2 秒程度より大きな圧縮遅延が発生するが画像自体の品質にそれほど大きな差は見られない。我々は帯域圧縮率の高い MPEG2 を用いて IP ネットワーク伝送を行なうシステムを構築してきたが MPEG2 方式はもともと低品質回線での使用は考えられてなく、フレーム間圧縮の高さ故にパケット損失に対して脆いという側面もある。そこで FEC (Forward Error Correction: 前方誤り訂正) を導入することによりこの点の改善を図った。本稿では実測評価とともにパケット損失回復の有効性、実用に耐えるものであることを示す。

## 2 インターネットにおける MPEG2 伝送

MPEG2 はフレーム間圧縮などにより高圧縮率を実現しているためわずかなパケット損失でも画像のノイズ等の影響が顕著に現れる [1] ため、広域ネットワークでの使用を考えるとパケット損失対策が必要不可欠である。ネットワーク伝送での誤り訂正は大別すると誤りの検出を自動的に送信側に伝えて再送要求を出す自動再送要求 (Automatic Repeat Request: ARQ) 方式と情報ビットに冗長ビットを付加して伝送し、伝送の途中で発生したビット誤りを冗長ビットを利用して受信側でビット訂正する前方誤り訂正 (Forward Error Correction: FEC) 方式とに分けられる。特に TV 会議、遠隔広義などに代表されるようなリアルタイム性の要求される通信においては誤り制御方式として FEC が一般に用いられる。誤りを検出、訂正する能力は FEC でも使用する符号化方式によって違いがある。

### 2.1 パリティ符号方式による FEC の実現

パリティを用いる FEC のための RTP (Real-Time transport Protocol) フォーマットが RFC2733 [2] で標準化されている。RFC2733 では RTP ヘッダの特定フィールドについても冗長化され、可変長データにも対応している。また、データパケットと冗長パケットは別ストリームとして伝送され、後方互換性を提供する。

### 2.2 Reed-Solomon 符号方式によるアプローチ

我々の実装には Reed-Solomon 符号方式を用いたアプローチをとった。Reed-Solomon 符号は誤り訂正符号の一つであり、連続した複数ビット単位 (シンボル) で誤りの検出、訂正をすることが可能である。一般にビット誤りが集中して発生するバースト誤りに強いといわれており、主に無線通信や DSL モデム、デジタル TV などに広く用いられている。Reed-Solomon 符号方式を認識しないノードではデータ部分をエンコードしないため、デコード処理の必要がなく、冗長部分のみを取り除くことで元のデータを得られる。一方、処理過程に複雑な演算を多用するために処理速度の低下を引き起こす。

## 3 MPEG2 over IP 伝送システム

我々の構築してきた MPEG2 over IP 伝送システム [3] は MPEG2 トランスポートストリーム (以下 MPEG2-TS) 形式の動画データデータを IP ネットワークを通して伝送するシステムであり、IPv6 (IP version 6) や IP マルチキャストにも対応している。これまでは IPsec (IP Security) を併用した場合の通常のテレビ品質 (SDTV) 画像や高品質テレビ (HDTV) 画像の伝送などの実験を行ってきた [4]。今回 FEC を導入することでパケット損失によって引き起こされる影響を軽減した。

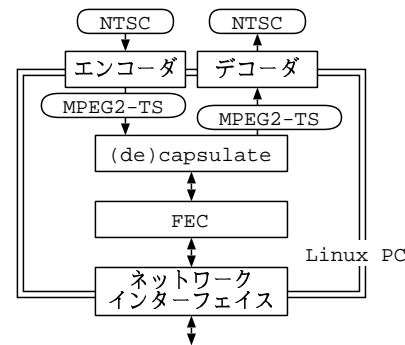


図 1: 伝送システムの処理。

本システムの処理概要を図 1 に示す。本システムでは送信と受信は一台の Linux マシンで動作可能である。エンコーダ、デコーダにはカメラやモニタが NTSC 接続される。エンコーダとデコーダにはそれぞれ Kfir MPEG2 encoder [5]、CineView [6] <sup>1</sup> を使用し <sup>2</sup>、MPEG2 Transport Stream (以下、MPEG2-TS) を対象としている。以下送信側、受信側の動作を示す。送信側ではまず NTSC 信号をエンコーダにより MPEG2-TS へ変換する。次に TS パケット (MPEG2-TS を伝送する際の固定長最小基本構成単位) 偶数個をカプセル化して固定長データを作る。FEC において固定長データから冗長パケットを生成し、順に送信キューに格納する。最後に FEC、RTP、UDP、IP の順に各ヘッダを付加して IP パケット (図 5 パケットフォーマット参照) を構成した後、ネットワークインターフェイスから IP ネットワークを介して受信ノードに転送する。受信処理では送信処理の逆を辿り、受信したパケットから FEC により可能な限り損失パケットの復元と誤り訂正を行ない、MPEG2-TS を再構成する。この際に FEC 復元不可能である場合は受

<sup>1</sup>CineView は製造中止となったため現在 Optibase VideoPlex XPress [7] への対応を予定。

<sup>2</sup>沖電気工業 (株) の YS3200 をエンコーダとして用いた場合、FreeBSD-4.x での動作も可能。

信したデータパケットのみを出力する。

### 3.1 FEC 処理の実装

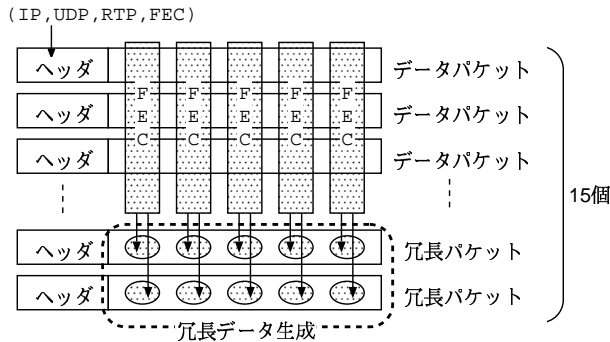


図 2: FEC エンコード処理.

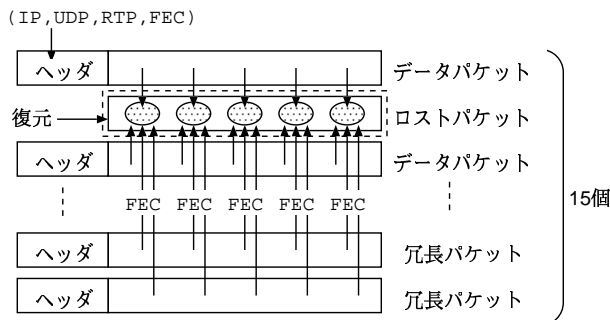


図 3: FEC デコード処理.

FECのエンコード/デコード処理には2.2で示したように誤り訂正符号として回復効率がよいとされるリードソロモン符号を用いた。FECエンコード処理を図2、同じくデコード処理を図3に示す。FECエンコード処理あるいはデコード処理では複数パケットデータを1つのユニットとし、パケットデータを順に並べた垂直方向のFECブロックに対してFECエンコーディング、デコーディングを行なう。ユニット内でのパケット順序がFECシーケンスによって示される(CS:Code Symbol size, DS:Data Symbol number, FECシーケンスは図4で示されるFECヘッダ中の各フィールドに対応)。CSはビット単位のシンボルの長さ(一連のFEC処理ユニットに含まれるパケットの数は $2^{CS} - 1$ 個)で、DSはデータシンボルの長さ(データパケットの個数)である。リードソロモン符号は複数個の連続したビット単位(シンボル)の誤りを訂正することが可能で高度な訂正能力を持つ反面、処理に複雑な演算を多用するため処理速度がネックになりやすい。そこで処理速度、

処理遅延、処理の簡単化を考慮して1シンボル4ビット( $CS = 4$ )とした。そのためパケット15個( $2^{CS} - 1$ )を1つのユニット(そのうちDS個が冗長パケット)として処理している。図2で示されるようにFECによって生成されるのはデータ部分のみであり、ヘッダ(IP, UDP, RTP, FEC)は冗長化しない。

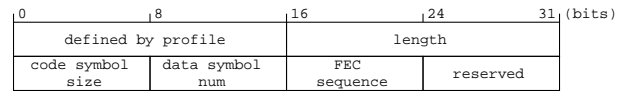


図 4: FEC ヘッダフォーマット.

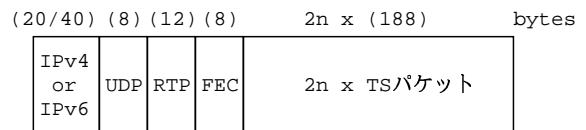


図 5: パケットフォーマット.

本システムでのパケットフォーマットを図5に示す。IP version 6, IPマルチキャストにも対応しており、UDPとRTPを用いる。RTPはリアルタイム性の要求されるマルチメディアデータの伝送には一般的に用いられる。FECヘッダはRTPの拡張ヘッダとして現在実装している。RTP拡張が理解出来ないノードではこのFECヘッダは無視されるため、FECヘッダに対応していない受信ノードにおいても悪影響を及ぼさない。ペイロードにはMPEG2-TSパケットを偶数個カプセル化する。図4に示されるFECヘッダフォーマットのdefined by profile, lengthフィールドはRTP拡張で定義された固定フィールドであり、defined by profileには0を使用している。CS, DS, FECシーケンスフィールドを追加してFECヘッダとした。

## 4 FECの性能測定

本システムのFECパケット損失回復性能について測定する。FEC回復における誤りは雑音などによるビット誤りとシンボル自体の損失誤りの2つに分けられる。それぞれの誤り訂正能力は決まっており、パケット損失の復元(損失誤り回復)の限界は付加する冗長パケットの数に依存し、ビット誤り回復は損失誤り回復の $\frac{1}{2}$ である。ここではパケット損失率10%以内のネットワークを対象とし、損失誤りの回復性能に注目しFEC前後のパケット損失率を測定する。また、実験による測定結

果とともにパケット損失率の理論値も計算し比較する。

#### 4.1 測定方法

表 1, 図 6 に測定に使用した機器とその構成をそれぞれ示す。本システムを用い送信 PC からパケットロス生成 PC 経由で受信 PC へ伝送する。その際, LOSS 生成 PC では 0, 2, 4, 6, 8, 10% の確率でパケット損失をランダムに発生させた。パケットの回復率はユニット内の全パケット数 (N) に対するデータパケット数 (K) で決まる。この測定では N は 15 固定とし, K を 10, 11, 12, 13 と変化させ, それぞれの場合について FEC デコード後のパケット損失率を測定した。以下, ユニット内のデータパケット数の組合せを (N, K) と示す。

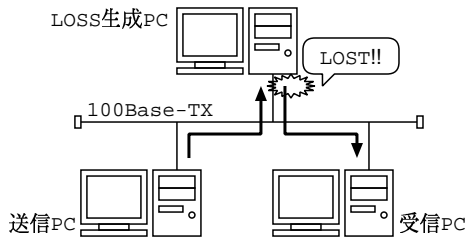


図 6: 測定に使用した機器の構成。

表 1: 測定に使用した機器

PC	CPU	memory	OS
送信	P3-800MHz	256MB	FreeBSD 4.2R
LOSS 生成	P3-450MHz	128MB	FreeBSD 4.2R
受信	P3-866MHz	128MB	Vine Linux 2.1

#### 4.2 理論値の算出

FEC 復元後のパケット損失率の理論値は次のように計算できる。データパケット数の組合せを (N, K) とし, 各パケットの損失は独立であり個々のパケットの損失率を  $p$  とする。FEC デコード後の損失パケット数の期待値を求め, K で除算すると FEC デコード後のパケット損失率が得られる。

ここでデータパケット K 個の中から FEC デコード後に  $i$  個損失しているという事象を  $X_i$  とすると,  $X_i$  が起こる確率  $P[X_i]$  は次のように表すことができる。

$i \neq 0$  の場合

$$P[X_i] = {}_K C_i p^i (1-p)^{K-i} \sum_{j=t}^{N-K} {}_{N-K} C_j p^j (1-p)^{N-K-j}$$

$$t = \begin{cases} 0 & \text{if } N-K < i \leq K \\ N-K-i+1 & \text{if } 0 \leq i \leq N-K \end{cases}$$

$i = 0$  の場合は全ての損失パケットが復元できる場合である。したがって損失パケット数が許容範囲内でないといけないので

$$P[X_0] = \sum_{s=0}^{N-K} {}_N C_s p^s (1-p)^{N-s} \quad (1)$$

損失パケット数の期待値は  $E[X] = \sum_{i=0}^K i P[X_i]$  であるのでパケット損失率  $P[X]$  は次のようになる。

$$P[X] = \frac{E[X]}{K} \quad (2)$$

$$= \frac{1}{K} \sum_{i=0}^K i {}_K C_i p^i (1-p)^{K-i}$$

$$\sum_{j=t}^{N-K} {}_{N-K} C_j p^j (1-p)^{N-K-j} \quad (3)$$

$$t = \begin{cases} 0 & \text{if } N-K < i \leq K \\ N-K-i+1 & \text{if } 0 \leq i \leq N-K \end{cases}$$

#### 4.3 測定結果

パケット損失率の測定結果とともに式 (3) から算出したそれぞれの場合における理論値を図 7 に示す。横軸は FEC 復元前のパケット損失率, 縦軸は FEC 復元後のパケット損失率である。

本システムにおける理論値と実験結果 (図 7) を比較すると理論値と実験値の間に多少の違いが見られる。一般に今回の実験環境を含むインターネット環境においてはバースト的なパケット損失が発生するが理論値の導出過程においてはパケット損失がそれぞれのパケットで独立であると仮定している。そのためこの違いが影響したものと考えられる。

#### 4.4 パリティ使用時のパケット損失率

パリティ方式を用いた場合によるパケット復元後のパケット損失率を計算し, 本システムで用いた Reed-Solomon 符号方式と比較することでその有用性を示す。

データパケット数を  $n$  個, その中からパリティチェック後に  $i$  個損失しているという事象を  $X_i$ , 各パケットのパケット損失率を  $p$  とし, パケットの損失確率  $P_i$  を計算する。まずパリティチェック後のパケット損失が 0 個となる確率は  $(n+1)$  個全て届く場合とデータ  $n$  個が届き, パリティパケットが損失する場合があるので

$$P[X_0] = (1-p)^{n+1} + {}_{n+1} C_1 p (1-p)^n$$

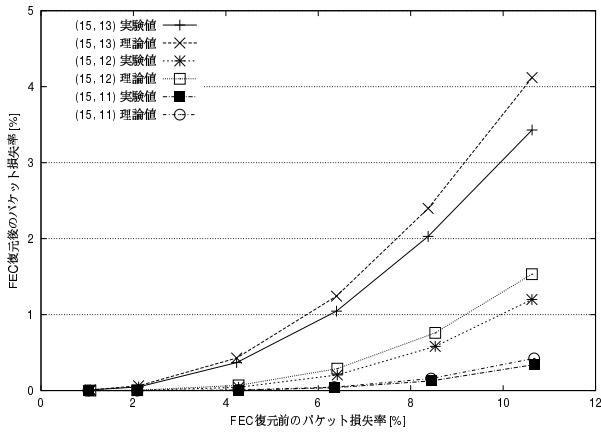


図 7: 本システムにおける FEC 性能測定.

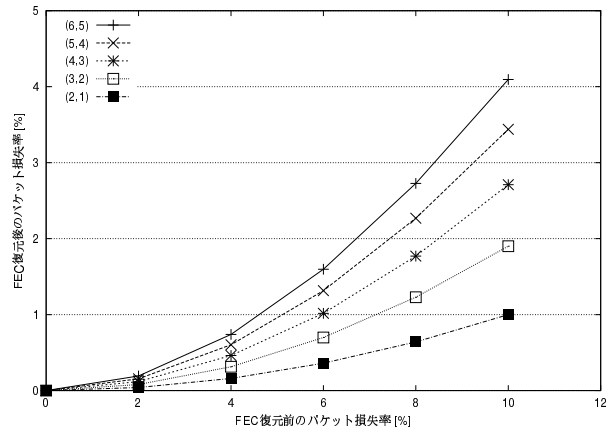


図 8: パリティを用いた場合の FEC 性能測定.

1 個の場合はデータ  $n$  個中 1 個損失かつパリティパケットも損失する場合であるので

$$P[X_1] = {}_n C_1 p(1-p)^{n-1} \times p$$

$i (> 2)$  個の場合はデータパケット  $n$  個から  $i$  個だけ落ちる場合であるので

$$P[X_i] = {}_n C_i p^i (1-p)^{n-i}$$

従ってパリティチェック後の損失パケット数の期待値  $E[X]$  は  $E[X] = \sum_{i=0}^n iP[X_i]$  となるのでパリティチェック後のパケット損失率は次のようになる.

$$\begin{aligned} \frac{E[X]}{n} &= \frac{1}{n} \sum_{i=0}^n iP[X_i] \\ &= {}_n C_1 p^2 (1-p)^{n-1} + \sum_{i=2}^n i {}_n C_i p^i (1-p)^{n-i} \end{aligned} \quad (4)$$

パケット損失率を 0 から 10% まで 2% ずつ変化させた場合において式 (4) から算出したパリティチェック後のパケット損失率の変化を図 8 に示す.

表 2: 帯域増加量の比較.

Reed-Solomon		パリティ	
冗長度	帯域増加量 [倍]	冗長度	帯域増加量 [倍]
(15,13)	1.15	(6,5)	1.2
(15,12)	1.25	(5,4)	1.25
(15,11)	1.36	(4,3)	1.33
		(3,2)	1.5
		(2,1)	2.0

パリティによるパケット損失回復性能 (図 8) と本システムによるもの (図 7) とを比較すると, Reed-Solomon

符号方式を用いた本システムの方が付加する冗長パケットの個数を増やすにしたがって大きくパケット損失率が改善されていることが分かる. 特に (15, 13) から (15, 12) において顕著に現れている. また帯域増加量 (表 2) の点からも 2 倍の帯域を必要とするパリティ方式 (2, 1) の場合と 1.25 倍の帯域となる Reed-Solomon 方式 (15, 12) がほぼ等しい回復性能を示していることが分かる. パリティの場合の (N, K) も同様に (全パケット数, データパケット数) を示す.

## 5 皆既日食ライブ中継実験

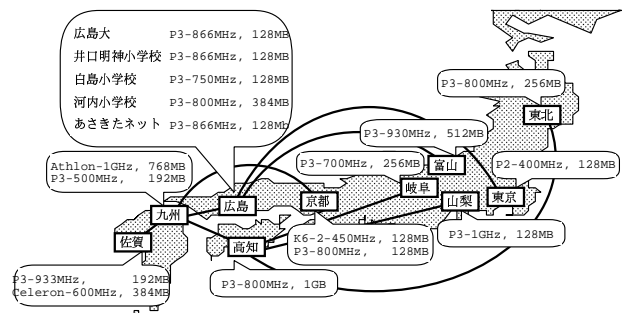


図 9: 日食中継実験の時の各拠点と接続.

我々は本システムを用いて皆既日食ライブ中継や遠隔合唱などを実施している. ここでは皆既日食ライブ中継 [8] について述べる. MAMEdeGansu プロジェクト [9] に参加する広島市内の小学校等 4 拠点と研究開発用ギガビットネットワーク (JGN)[10] 上の 9 拠点を接続する IP マルチキャスト網 (図 9) を構築し, ライブ!

エクリプス実行委員会 [11] の実施する皆既日食ライブ中継を FEC を用いた MPEG2 over IPv4 マルチキャストで配送した。中継に使用したパケットの組合せは (15, 13)(データパケット 13 個に対して冗長パケット 2 個) である。その結果ほとんどの拠点において受信時のパケット損失率約 0.4% からほぼ 0% まで FEC によって復元出来るという測定結果 (表 3) が得られた。FEC 復元後の理論値は 4.2 節の式 (3) から算出した値である。この結果も FEC 復元後のパケット損失率と理論値を比較した際バースト的パケット損失が起こっていると考えられるが復元前のパケット損失率自体が低いいためほぼ一致しているものとみなせる。受信拠点東京は FEC による復元を行わないオプションを付加していたため FEC 後のパケット損失率は空欄としている。日食中継時の佐賀における FEC 復元前のパケット損失率の変動を図 10 に示す。図 10 は各 10 秒間の平均パケット損失率をプロットしたものである。他の拠点においてもほぼ同じような傾向であった。FEC 復元後は表 3 のようにパケット損失率ほぼ 0% まで回復した。

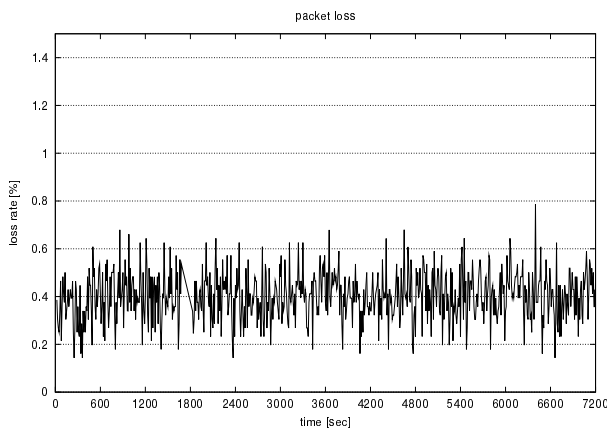


図 10: 佐賀におけるパケット損失率。

## 6 おわりに

本稿では MPEG2 動画像伝送システムに Reed-Solomon 符号方式による FEC の実装を行なった。さらにパケット損失に対する性能評価ではその回復能力の高さと有効性を示した。また皆既日食ライブ中継では広域ネットワークでの使用におけるデータも得られ、実際のネットワークでも有効に働くことを確認した。現在は RTP の拡張として実装し、固定長データであるトランスポートストリームを対象としているが、可変長である

表 3: パケット損失率

受信拠点	パケット損失率 [%]		
	FEC 復元前	FEC 復元後	理論値
広島大	0.00062	0.00025	/ 0.00000
井口明神小学校	0.40449	0.00069	/ 0.00050
白島小学校	0.40526	0.00015	/ 0.00050
河内小学校	0.40033	0.00025	/ 0.00049
あさきたネット	0.40034	0.00025	/ 0.00049
富山	0.40792	0.00021	/ 0.00051
東京	0.20653	—	/ —
九州	0.40115	0.00026	/ 0.00049
	0.40518	0.00225	/ 0.00050
佐賀	0.40243	0.00134	/ 0.00049
	0.40225	0.00133	/ 0.00049
京都	0.41219	0.00555	/ 0.00053
	0.40786	0.00446	/ 0.00051
高知	0.40255	0.00193	/ 0.00049
岐阜	0.40094	0.00052	/ 0.00049
東北	0.37727	0.00000	/ 0.00041
山梨	0.40312	0.00226	/ 0.00050

プログラムストリームなどへの対応のために RFC2733 の拡張として実装すること、バースト的パケット損失に対する側面の弱さの軽減のためのパケットの送信順序を変えるなども今後検討される。

## 参考文献

- [1] 西村浩二, 松本勝之, 相原玲二: “MPLS ネットワーク上での格差サービスの実装と評価”, 情報処理学会論文誌, vol.42, No.2, pp.213–221, Feb.2001.
- [2] J. Rosenberg, H. Schulzrinne: An RTP Payload Format for Generic Forward Error Correction, RFC 2733, December 1999.
- [3] <http://net.ipc.hiroshima-u.ac.jp/mpeg2ts/>
- [4] Kouji NISHIMURA, Reiji AIBARA, Tamaki OHTSUKA, Tohru KONDO: An MPEG2 Transfer System using IPv6 Multicast and IPsec, Proceedings of the 7th Asia-Pacific Conference on Communication (APCC2001), pp.578–581.
- [5] <http://www.linuxtv.org/>
- [6] <http://www.vela.com/>
- [7] <http://www.optibase.com/>
- [8] 相原玲二, 大塚玉記, 近堂徹, 西村浩二, 前田香織: “MPEG2 over IPv6 システムの開発と皆既日食中継実験”, 電子情報通信学会インターネットアーキテクチャ研究会, 2001 年 11 月 (予定).
- [9] <http://www.csi.ad.jp/activity/MAMEdeGansu/>
- [10] <http://www.jgn.tao.go.jp/>
- [11] <http://www.live-eclipse.org/>