

マルチキャストの特徴を利用した非インライン型FEC装置の検討

佐々木 力[†] 田上 敦士[†] 長谷川輝之[†] 阿野 茂浩[†]

[†] (株) KDDI 研究所 〒356-8502 ふじみ野市大原 2-1-15

E-mail: [†]ch-sasaki@kddilabs.jp

あらまし IP マルチキャストを用いたコンテンツ配信の高信頼化手法として、FEC (Forward Error Correction) が広く利用されている。通常、FEC の符号化処理は配信元で一括して行われるが、高品質な受信者に対する必要以上の冗長配信が問題となる。これに対し、低信頼区間に別途 FEC 装置を導入し、配信元での冗長化率を抑制するアプローチが考えられる。本稿では、マルチキャストの経路制御機能に着目し、区間両端のルータへ付加的に接続した非インライン型 FEC 装置を用いて対象マルチキャストのみを高信頼化する方式を提案する。併せて、試作装置を用いた評価結果を述べる。

A Study on Non-inline Implementation of Add-on FEC System considering Multicast Forwarding Mechanism

Chikara SASAKI[†], Atsushi TAGAMI[†], Teruyuki HASEGAWA[†], and Shigehiro ANO[†]

[†] KDDI R&D Laboratories Inc., 2-1-15 Ohara, Fujimino-shi, 356-8502 Japan

E-mail: [†]ch-sasaki@kddilabs.jp

Abstract FEC (Forward Error Correction) is widely used in content delivery service based on IP multicast to provide reliable data transmission. Generally FEC encoding is applied at a multicast source using a common redundancy rate. However, unnecessary transmission of redundancy data will occur from the view point of receivers realizing high transmission quality. It could be a solution that add-on FEC system is selectively introduced to both edges of network segments with low transmission quality. In this paper, we study on non-inline implementation of FEC system which is set aside the edge routers, and pulling only target multicasts to improve reliability.

1. ま え が き

近年、IPTV (Internet Protocol TeleVision) に代表されるように、大容量の映像・音声コンテンツを IP ネットワーク上でストリーミング配信するサービスが増加しつつある。このようなサービスに対しては、多数の受信者に同時配信可能な IP マルチキャスト技術が有効である。一方、高信頼なマルチキャスト配信を実現するためには、パケットロスに対する誤り回復技術が必要となる。誤り回復技術としては再送 (ARQ: Automatic Repeat reQuest) [1], [2] や配信データの冗長化 (FEC: Forward Error Correction) [3], [4] などが挙げられるが、多くの受

信者が想定されるマルチキャスト配信では、サーバでの応答爆発等の問題から、ARQ よりも FEC を利用するのが一般的である。

通常、FEC の符号化処理は配信元で行い、全受信者で同じ冗長化率を用いる。このため、パケットロスが最も頻繁に発生する低品質な受信者に合わせた冗長化が必要となる。これは、高品質な通信環境の受信者にも、余分なデータが送信される事を意味する。ユビキタス環境や FMC (Fixed Mobile Convergence) 環境のように、様々なコスト・品質の回線が混在する場合、各回線における帯域の有効活用は重要である。また、ネットワークの再設計・設備更改や障害等により特定区間の通信品質が変

化することもある。このような環境では、FEC の冗長化率を特定区間のみ簡易に変更できる技術が有効となる。

特定区間の高信頼化手法としては、FEC 装置を区間両端の回線の上にインライン設置する方法が考えられる。この場合、FEC 装置は対象マルチキャストの高信頼化と同時に、それ以外のパケットの中継転送も行わなければならないため、広帯域な回線では転送ボトルネックとなる可能性がある。また、FEC 装置に障害が発生すると、その区間を通る全てのトラフィックに影響が及ぶことも問題となる。

これに対し我々は、高信頼化対象区間の両端のルータに付加的に設置可能な非インライン型の FEC 装置を検討する。本 FEC 装置のエンコーダとデコーダは、マルチキャスト転送の仕組みを応用し、対象マルチキャストだけを引き込み FEC 処理を付加する。他の通信は本装置を経由しないため、装置設置による影響を受けない。本稿では、具体的な非インライン型 FEC 装置として、動作原理の異なる 2 種類の方式を提案し比較検討する。

本稿の構成は以下の通りである。2. では既存技術として FEC の概要とインライン型 FEC 装置について説明する。3. で非インライン型の FEC 装置として、対象マルチキャストの転送はそのままに冗長データのみを追加伝送する冗長データ追加型と、対象マルチキャスト転送のみインライン型と同様に FEC 装置を通過させるリダイレクト型を提案する。4. において試作機による性能評価を行い 2 つの方式を比較する。5. で結果を考察し 6. でまとめを行う。

2. 既存技術

本節では、パケットレベル FEC の概要、ならびに、インライン型の FEC 装置について説明する。

2.1 FEC の概要

マルチキャスト通信では、トランスポート層のプロトコルとして UDP (User Datagram Protocol) を用い、FEC を組み合わせて通信を保護する。一般に、IP ネットワーク上の通信エラーはパケットロスのみとなるため、パケット単位で FEC 処理を行う。

図 1 に FEC 処理の流れを示す。符号化と復号化は固定長のデータブロック単位で行うものとする。エンコーダ (送信者) では、データを固定長のデータブロックに分割し、データブロックをさらに情報シンボルに分割する。次に、同一データブロック内の情報シンボル同士を演算し、その結果を符号化シンボルとする。このとき、生成する符号化シンボル数 n を情報シンボル数 k よりも多くすることで冗長化が行われる。これら同一のデータブロックから生成された符号化シンボルを結合したものを

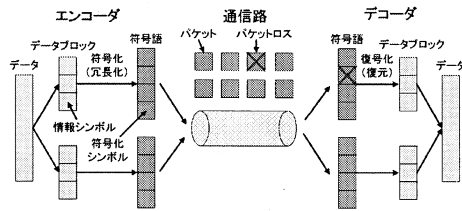


図 1 FEC 処理の流れ

Fig. 1 FEC processing flow.

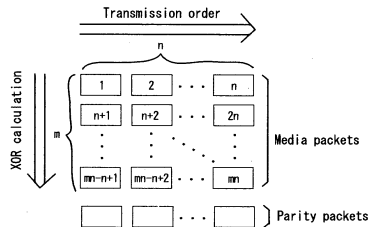


図 2 $(mn + n, mn)$ パリティ符号

Fig. 2 $(mn + n, mn)$ parity codes.

符号語と呼ぶ。本稿では、情報シンボルと符号化シンボルの単位はパケットとする。通信経路上でいくつかのパケットロスが発生したとしても、デコーダ (受信者) は冗長化した程度のロスであれば、データブロックを復元することができる。以上のような冗長化率 n/k の符号を (n, k) 符号と呼ぶ。

次に、単純パリティ符号 $(mn + n, mn)$ の例を図 2 に示す [5]。エンコーダは、データブロックをそのまま mn 個のメディアパケットに分けて送信し、さらに、冗長性を持たせるために n 個のパリティパケットを送信する。パリティパケットのペイロード部は、 $m \times n$ 行列を用いた符号化処理により生成され、 m 個のメディアパケットのペイロード部を行列の列方向に XOR 演算したものに、FEC ヘッダや RTP (Real-time Transport Protocol) ヘッダ [6] を付加したものである。これらのヘッダには FEC パラメータやシーケンス番号等が含まれる。図 2 の例では、メディアパケットの送信はデータ入力後すぐに行うことができる一方、パリティパケット送信はこれに必要な m 個のメディアパケット送信後となる。デコーダは、同列内の 1 メディアパケットのロスについては、受信した $m - 1$ 個のメディアパケットと対応するパリティパケットの XOR 演算により復元することができる。また、同行列内の n 個のバーストロスと復元することができる。

2.2 インライン型 FEC 装置

全受信者に対して同一の冗長化率を用いる一般的なマルチキャスト配信形態では、品質が最も劣る受信者に合

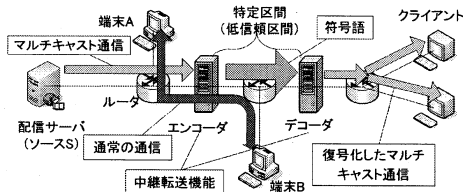


図3 インライン型 FEC 装置
Fig. 3 Inline type FEC system.

合わせた冗長化率での運用が必要となる。即ち、保守作業等で一時的に発生した低信頼区間の品質を確保する場合や、宅内無線 LAN 等の送信者とは異なる管理下にある低信頼区間に対応する場合は、全体の冗長化率を変更しなければならない。しかしながら、送受信装置の FEC 冗長化率変更が困難、あるいは、冗長化率拡大に伴い一部回線で帯域が不足するといったシナリオも考えられる。この場合、特定の低信頼区間（以下単に特定区間と呼ぶ）のみを高信頼化する外付け FEC 装置を適用する方法が考えられる。

具体的な外付け FEC 装置としてはインライン型 [7] が提案されている。これをマルチキャストに適用することも可能性である。図 3 にインライン型の概要を示す。エンコーダは、通過するパケットの中から対象とするマルチキャストパケットを抽出し、符号化処理を行ってデコーダ側へマルチキャスト転送する。デコーダは、同様に通過するパケットの中から符号化処理されたパケットの抽出やロスの検出等を行った上で、復号化処理を行う。本形態では、対象とするマルチキャストパケットを符号化・復号化しつつ、それ以外のパケット（図 3：端末 AB 間の通信）を中継転送する必要がある。そのため、特に広帯域回線においては、FEC 処理対象となるマルチキャストの増加による中継転送スループットの低下や、その逆の現象が発生する可能性がある。また、FEC 装置に障害が発生すると、対象マルチキャストのみならず、装置を経由する全通信に影響が及ぶこととなる。このように、インライン型 FEC 装置では、スループットや可用性の観点から性能ボトルネックとなり得る。

3. 提案手法

3.1 非インライン型

本節では、FEC 装置がボトルネックとならないように、対象マルチキャスト以外の中継転送を殆ど必要としない非インライン型 FEC 装置を検討する。具体的には、図 4 に示すように、FEC 装置を対象区間の両端のルータに追加接続し、対象マルチキャストやその冗長データのみをエンコーダとデコーダ経由で転送させる一方、端末 AB

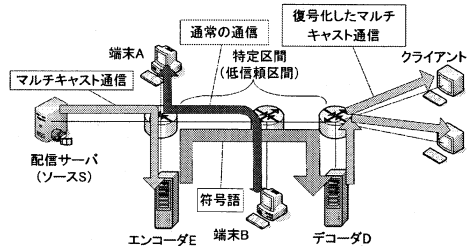


図4 非インライン型 FEC 装置
Fig. 4 Non-inline type FEC system.

間の通信は従来の経路を保持するように設計する。

非インライン型の形態として、以下 2 つを検討する。

- 対象マルチキャストをエンコーダに引き込む一方、そのまま特定区間に流す。エンコーダでは冗長データを追加生成しデコーダへ転送する。
 - 対象マルチキャストをエンコーダに全て引き込み、特定区間にはそのまま流さない。エンコーダは冗長データを含む符号語全体をデコーダへ転送する。
- 前者を冗長データ追加型と呼び、後者をリダイレクト型と呼ぶ。以下、3.2 で冗長データ追加型を説明し、3.3 でリダイレクト型を説明する。

方式検討にあたり、経路制御・グループ管理プロトコルとして以下を想定した。

- OSPF (Open Shortest Path First) [8]
- PIM-SM (Protocol Independent Multicast-Sparse Mode) [9]
- IGMP (Internet Group Management Protocol) [10]

以下、マルチキャストのソースを S 、エンコーダを E 、デコーダを D とし、 (S, G) でソース S 、マルチキャストアドレス G のマルチキャストを表す^(注1)。PIM-SM では本来、マルチキャストがソース S を根とする SPT (Shortest Path Tree) 以外にも、RP (Rendezvous Point) を根とする共有木 (RPT) 経由で転送されることがある。ただし、本検討では簡単のため、

- RPT から SPT への切替が即時に行われ主な配信は SPT 経由で行われること、
 - RPT と SPT は経路が共通であること、
- を前提に、SPT $((S, G)join)$ に対する高信頼化手順を述べることとする。

3.2 冗長データ追加型

冗長データ追加型では、高信頼化対象マルチキャスト G を対象区間にそのまま転送しつつ、追加の冗長データをエンコーダからデコーダへ送信する。エンコーダから

(注1) $(*, G)$ は任意のソースからの送信されるマルチキャストアドレス G のマルチキャストを表す。

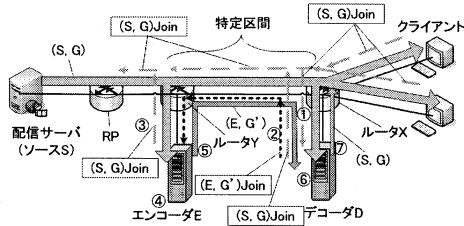


図5 冗長データ追加型 FEC 装置
Fig. 5 Parity injection type FEC system.

デコーダへの転送は、マルチキャスト G' (別マルチキャストアドレス) を用いる。図5に冗長データ追加型の概要を示す。具体的手順は以下の通りである。

冗長データ追加型の手順:

- (1) デコーダは高信頼化対象マルチキャスト (S, G) について、ルータ X からルータ Y へ向かう (S, G) join を監視する。
- (2) 当該 join を検出すると、デコーダ受信用に (S, G) join と (E, G') join を行う。
- (3) エンコーダは (E, G') join を検出すると (S, G) join する。
- (4) 受信した (S, G) に符号化処理を行う。
- (5) 生成した冗長データは、別マルチキャスト (E, G') を用いて送信する。
- (6) デコーダは受信した (S, G) のシーケンス番号からロスを検出すると、 (S, G) と (E, G') からロスパケットを復元する。
- (7) デコーダは復元パケット (S, G) をルータ X における S 宛ての経路に対応したインタフェースへと送信する。

□

ルータでは、ループ防止のために、RPF (Reverse Path Forwarding) チェック [11] が動作しており、ソース S (もしくは RP) に向かうユニキャスト経路に対応するインタフェースのみマルチキャスト (S, G) の受信が許可される。そこで、手順 (7) においてデコーダが送信した復元パケットが破棄されないように、ルータ X では S 向きの上流ルータ Y とデコーダを同一 LAN セグメントに収容する。本収容方法により、手順 (1) でルータ Y へ送信された (S, G) join も、デコーダで受信することができる。また、高信頼化対象はマルチキャストであるため、手順 (3) では、エンコーダが (S, G) join するだけで、当該マルチキャストパケットを受信することができる。

なお、冗長データ追加型で利用できる誤り訂正符号は、単純パリティ符号や LDPC [12] のように、符号語自身にデータブロックが含まれる符号に限定される。エンコー

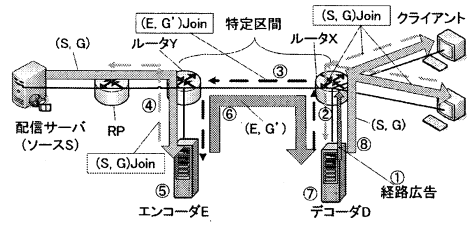


図6 リダイレクト型 FEC 装置
Fig. 6 Redirection type FEC system.

ダへの入力と同一のデータはルータ Y によってそのまま対象区間へ転送されるので、エンコーダは生成データの中から入力と同一内容のデータを除いた冗長分のデータのみ転送する。エンコーダからデコーダへの転送には別マルチキャスト (E, G') を用いる。そのため、1台のエンコーダに対して、下流に多数のデコーダを設置することが可能であり、またデコーダ以外には転送されないようにできる。

デコーダは、受信した (S, G) の中でロスパケットのみ復元し転送すれば良いが、ロス検出や復号処理等により、復元したパケットの到達に遅延が発生する。即ち、クライアント側では到達パケットに順序逆転が発生する可能性があり、アプリケーション層で到達逆転を補償する必要がある。なお、デコーダからの復元パケットは、ルータ Y にも転送される。多数のロスが発生すると復元パケット数も増大するため、ルータ X でのフィルタリングが有効となる場合もある。

3.3 リダイレクト型

3.3.1 動作説明

冗長データ追加型では、復元パケットが遅れて到達し、また使用できる誤り訂正符号が限られる。そのため、本節では、マルチキャストの経路制御を利用して対象マルチキャストの転送経路を変更させることで、マルチキャスト全体をインライン型のようにエンコーダとデコーダ経由で高信頼化可能とするリダイレクト型の FEC 装置を検討する。図6にリダイレクト型の概要を示す。リダイレクト型では、ソース S からのマルチキャスト (S, G) をエンコーダに引き込み、エンコーダとデコーダ間は符号語をマルチキャスト (E, G') に変換してリダイレクトし、デコーダからの復号データをマルチキャスト (S, G) に戻してクライアントへ転送する。具体的手順は以下の通りである。

リダイレクト型の手順:

- (1) デコーダは接続するルータ X に適切なコストに基づく経路情報を広告し、ルータ X の保持するソース S への経路をデコーダに向ける。

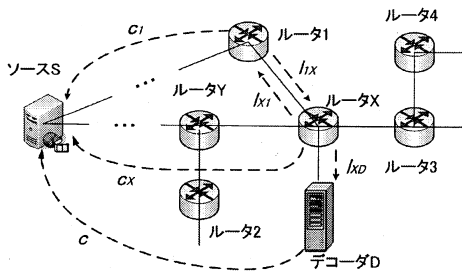


図7 OSPFのコスト設定例

Fig.7 Example of OSPF cost setting.

- (2) ルータ X はクライアントからの *join* を受信後, (S, G) を配信するために, ルーティングテーブルに従って, デコーダへ (S, G) *join* する.
- (3) ルータ X から (S, G) *join* を受信したデコーダは, (E, G') に *join* する.
- (4) エンコーダは (E, G') *join* を検出すると (S, G) *join* する.
- (5) エンコーダは受信した (S, G) に符号化処理を行う.
- (6) 生成した符号語は, 別マルチキャスト (E, G') を用いて送信する.
- (7) デコーダは受信した (E, G') を復号する.
- (8) デコーダは復号した全パケットを (S, G) に変換してルータ X に送信する.

□

手順 (1) において, ルータ X の保持する S 宛での経路情報を変更させることによって, RPF チェックを回避し, デコーダから送信されたマルチキャストを (S, G) としてクライアントに転送することができる. ただし, ルータ X でソース S 宛での経路をデコーダ方向に変更しているため, ルータ X を経由して S に至るユニキャスト通信もデコーダに転送される. そのため, デコーダとエンコーダ間にユニキャストの IP in IP トンネル [13] を設定し, S への到達性を確保する.

3.3.2 広告するコスト

以下, 図 7 を用いて, デコーダが広告するコスト C について検討する. コスト C は以下の 2 つの条件を満足する値に設定する.

条件 1: ルータ X の保持する S 宛での経路をデコーダに向けさせる.

条件 2: 広告したコストによって, X 以外のルータの経路にできるだけ影響を与えない.

ルータ X に隣接するルータ集合を Z_1 ($Z_1 = \{\text{ルータ 1, ルータ 3, ルータ Y}\}$), Z_1 の中でソース S

までの経路にルータ X を含むルータの集合を Z_2 ($Z_2 = \{\text{ルータ 3}\}$) とする. ルータ i から S に至る経路のコストを C_i , ルータ i からルータ j へのリンクコストを l_{ij} , ルータ X からデコーダ D へのリンクコストを l_{XD} とする. なお, これらのルータは OSPF の同一エリア内に属しているものとし, C_i や l_{ij} は LSA (Link State Advertisement) を収集し計算することとする. 条件 1 に関しては, ルータ X で保持しているソース S 宛での経路コスト C_X よりもデコーダからの広告に基づく経路コストの方が小さくなればよいため,

$$C + l_{XD} < C_X \quad (1)$$

を満足すればよい. ただし, $C > 0$ である必要があるため, 式 (1) より本方式は,

$$C_X > l_{XD}$$

を満足している環境でのみ有効である.

次に, 条件 2 を検討する. 式 (1) を満足するようにコスト C を小さく設定したとしても, Z_2 に含まれるルータのネクストホップはルータ X のままである. したがって, $Z_1 - Z_2$ に対してのみ, 経路変更が起こらないようにする. 条件 2 に対しては,

$$\begin{aligned} C + l_{XD} + l_{iX} > C_i \quad \text{for all } i \in Z_1 - Z_2 \\ \Leftrightarrow C > \max_{i \in Z_1 - Z_2} (C_i - l_{iX}) - l_{XD} \quad (2) \end{aligned}$$

を満足すればよい. ただし, $\max_{i \in A} f(i)$ は空間 A における $f(i)$ の最大値を表す.

式 (1), 式 (2) より, コスト C は,

$$\max_{i \in Z_1 - Z_2} (C_i - l_{iX}) - l_{XD} < C < C_X - l_{XD} \quad (3)$$

の範囲内で設定する.

なお, ルータにおいて転送パケットがルーティングテーブル上の複数のプレフィックスに一致する場合, 各プレフィックスのコストの大小にかかわらず, 最長一致 [11] するものが選ばれる. そのため, デコーダから広告するソース S に関する経路情報のプレフィックス長は, エリア内で交換されている当該経路情報と同一にする必要がある. デコーダからの広告プレフィックスが長い場合は, 同一エリア内での S 宛での通信が全てデコーダ経路となり, また短い場合はより長いプレフィックスを持つ従来の経路情報が有効となり, デコーダへのリダイレクトが実現できないことに注意する.

4. 性能評価

4.1 評価環境

本節では試作した FEC 装置の評価を行う. 評価環境

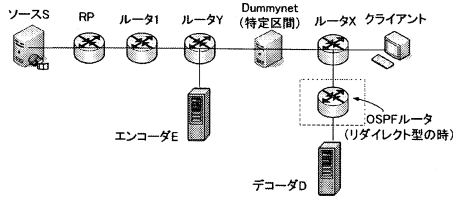


図 8 評価環境

Fig. 8 Network configuration for evaluation.

表 1 装置諸元

Table 1 Hardware specification.

Multicast source	IXIA
RP	Catalyst3560
Other routers	Catalyst3750
Encoder	Linux CPU 2.0GHz メモリ 1GB
Decoder	Linux CPU 2.0GHz メモリ 1GB
dummynet	FreeBSD CPU 533MHz メモリ 760MB

を図 8 に、装置諸元を表 1 に示す。具体的には、ソース S にトラフィックジェネレータ IXIA [14] を用いて、マルチキャスト (S, G) を 10Mbps で送信する。パケット長は 1400Byte とする。誤り訂正符号は $(mn + n, mn)$ パリティ符号を使用する。リダイレクト型のデコーダではバッファリング機能を実装しており、 c 個のデータブロック (mn 個のパケットから構成される符号化行列) の範囲内であればパケットの順序逆転を復元することができる。以下の評価では $c = 1$ に固定する。特定区間でのパケットロスは確率 p のランダムロスを想定し、dummynet [15] によって模擬する。なお、定常状態を評価するために、エンコーダとデコーダは常時マルチキャストの受信と符号化、復号化を行っているものとする。

リダイレクト型では、OSPF を広告するルータ (OSPF ルータ) をデコーダとルータ X の間に設置する。通常、OSPF のリンクコストは基準値を 100Mbps として以下により計算される。

$$\text{リンクコスト} = \text{基準値} / \text{インターフェイスの帯域幅}$$

使用したルータの各インターフェイスの帯域は基準値以上の 1Gbps になっているため、リンクコストは 1 となる。そこで、基準値を 100Mbps から 10Gbps に変更する。これにより、ソース S の所属するネットワークまでの経路コストとしてルータ X とルータ Y はそれぞれ 40 と 30 を保持する。よって OSPF ルータでは、 S のネットワーク宛てのネクストホップをデコーダ D に設定すると共に、式 (3) を満たすように、ルータ X に対して当該経路をコスト 25 で広告することとした。

表 2 誤り訂正能力

Table 2 Error correction capability.

	$m = 5, n = 100$	$m = 10, n = 100$
$p = 1\%$	5.1×10^{-4}	9.9×10^{-4}
$p = 5\%$	1.1×10^{-2}	2.0×10^{-2}
$p = 10\%$	4.2×10^{-2}	6.5×10^{-2}

4.2 評価結果

本節では、誤り訂正能力と遅延について評価結果を述べる。

4.2.1 誤り訂正能力

本評価試験では、ソース S からのパケット転送に対して dummynet によりパケットロスを発生させ、クライアントでパケットロス率 (復号失敗率) を測定する。誤り訂正能力は用いる符号によって決まるため、冗長データ追加型のみ評価する。表 2 にパラメータと得られた復号失敗率を示す。なお、復号処理遅延によってパケットの到達順序が逆転したとしても、ロスパケットが正しく復元できれば復号成功と見做す。FEC 装置の導入により、パケットロス率 1% が 0.05~0.1% に誤り修正されていることが分かる。

4.2.2 遅延

遅延の評価試験では、図 8 においてクライアントにソース S と同一筐体の IXIA を使用し、送信時刻と受信時刻の差から最大遅延と平均遅延を測定した。本評価においては、dummynet の処理等で転送遅延が発生するため、FEC 装置を設置しない場合の遅延も併せて測定し、FEC 装置設置による遅延増加量を評価する。パケットロス率 $p = 1\%, 5\%$ に対して、10 万パケット送信時の結果を表 3, 4 に示す。表 3, 表 4 より、リダイレクト型は冗長データ追加型に比べて最大遅延、平均遅延ともに大きくなる事が分かる。なお、FEC 装置を配置しない場合は、ロス率 p によらず最大遅延約 25ms、平均遅延 1ms 未満であった。

5. 考察

5.1 誤り訂正能力

パリティ符号の復号失敗率を検討する。 m 個のメディアパケットの中で k 個のロスが発生する確率 $g(m, k)$ は ${}^m C_k \cdot p^k (1-p)^{m-k}$ である。そのため、その平均ロス個数 $E_l(m)$ は $\sum_{k=1}^m k \cdot g(m, k) = mp$ となり、1 パケット当たりに換算すると、 $E_l(m)/m = p$ となる。一方、パリティ符号使用時は、デコーダにおいてパリティパケットを受信できれば (確率 $1-p$ で発生)、同列内の m 個のメディアパケットのうち 1 個のロスは復元可能である。したがって、 $k = 1$ 個のロスが発生しても復号後のロス個数は 0 となり、復号後の平均個数 $E'(m|reception)$ は

表 3 ロス率 $p = 1\%$ 時の遅延増加量 (ms)

Table 3 Delay increment at loss ratio $p = 1\%$ (ms).

	冗長データ追加型		リダイレクト型	
	最大遅延	平均遅延	最大遅延	平均遅延
$m = 5, n = 100$	424	2	655	292
$m = 5, n = 1000$	4456	20	7011	6134
$m = 10, n = 100$	983	4	1242	1325
$m = 10, n = 1000$	10056	44	13174	12114

表 4 ロス率 $p = 5\%$ 時の遅延増加量 (ms)

Table 4 Delay increment at loss ratio $p = 5\%$ (ms).

	冗長データ追加型		リダイレクト型	
	最大遅延	平均遅延	最大遅延	平均遅延
$m = 5, n = 100$	431	8	685	575
$m = 5, n = 1000$	4462	79	6960	6792
$m = 10, n = 100$	984	15	1278	1222
$m = 10, n = 1000$	10068	153	13420	12796

$\sum_{k=2}^m k \cdot g(m, k)$, 復号失敗率は $E'(m|reception)/m$ となる。一方, パリティパケットをロスした場合 (確率 p で発生), 復号後のロス個数 $E''(m|loss)$ は $E_l(m)$ と同じになる。以上より, パリティ符号の復号失敗率は,

(復号失敗率)

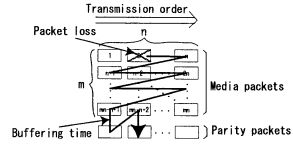
$$\begin{aligned}
 &= (1-p) \cdot (\text{パリティパケット受信時の復号失敗率}) \\
 &\quad + p \cdot (\text{パリティパケットロス時の復号失敗率}) \\
 &= (1-p)E'(m|reception)/m + pE''(m|loss)/m \\
 &= (1-p)\{E_l(m) - 1 \cdot g(m, 1)\}/m + pE_l(m)/m \\
 &= E_l(m)/m - (1-p) \cdot g(m, 1)/m \\
 &= p\{1 - (1-p)^m\} \tag{4}
 \end{aligned}$$

となる。式 (4) に各パラメータを代入することで表 2 と近い値を得ることができる。

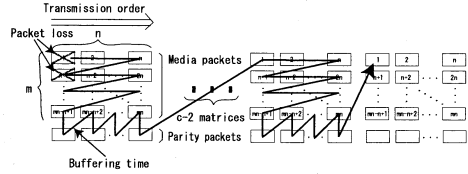
また, 表 2 より, m が小さな値ほど (冗長化率が大きいほど) 優れた訂正能力を得ることを確認した。なお, 本評価では特定区間において順序逆転がないランダムロスのみを前提としたため, n, c によって誤り訂正能力は変わらない。

5.2 遅延

FEC 装置による遅延の大部分は, デコーダにおけるバッファリング時間である。これは, ロスパケットを復元するために, デコーダがロスパケットと同列のメディアパケットとパリティパケットを受信しなければならないことに起因する。冗長データ追加型において, この受信待ち時間が最大となるのは, 符号化の $m \times n$ 行列の 1 行目のメディアパケットがロスした場合であり, その値はおおよそ $(m-1)n$ 個のメディアパケットの送信時間と



(a) Parity injection type.



(b) Redirection type.

図 9 デコーダの最大バッファリング時間

Fig. 9 Maximum buffering time in decoder.

なる^(注2) (図 9(a)). メディアパケットの送信速度が t (パケット/秒) であれば, デコーダでのバッファリングによる最大遅延は約 $(m-1)n/t$ 秒である。例えば, $t = 893$ (10Mbps 時), $m = 5, n = 100$ とすると, 最大遅延は約 448ms となる。なお, 表 3, 4 は 10 万パケット送信時の最大値であるため, 大きな p ほどデコーダにおけるバッファリング時間が $(m-1)n/t$ に近い値となる確率が高くなる。

一方, リダイレクト型では, $m \times n$ 行列の 1 行 1 列に対応するメディアパケットがロスし復元できない場合に最大遅延が発生する。この場合, デコーダでは, $c = 1$ 個のデータブロックの範囲内であれば到達順序を保証しなければならないため, $c \cdot mn$ 個のメディアパケット (および $c \cdot n$ 個のパリティパケット) の受信を待たなければならない (図 9(b))。したがって, デコーダでのバッファリングによる最大遅延は約 cmn/t 秒である。例えば, $t = 893$ (10Mbps 時), $m = 5, n = 100, c = 1$ とすると, 最大遅延は約 560ms となる。

冗長データ追加型では, 復元パケットの遅延量が到達順序逆転の大きさに対応する。クライアントの実装によっては遅れて到着した復元パケットが破棄される可能性もある。ただし, FEC 装置の導入に際しては, クライアント・ソース間で別の FEC 適用が想定されており, そのバッファリング処理を通じてある程度の順序逆転は対応可能である。一方, リダイレクト型では転送遅延は付加されるものの, 到着順序は保証されることが特長である。前者の場合は, クライアントで対応可能な順序逆転の大きさ, 後者についてはサーバ・クライアント間で許容されるエンド・エンドでの遅延量を考慮して, メディアパケッ

(注2): パリティパケットは生成後即座に送信されるものとする。

トの送信速度 t に応じた FEC のパラメータ (m, n, c) を設定する必要がある。

5.3 障害時の動作

冗長データ追加型では対象マルチキャストがそのまま特定区間に流れる。そのため、エンコーダもしくはデコーダに障害が発生した場合も、冗長データの転送が停止するのみであり、対象マルチキャスト自身の配信は継続する。一方、リダイレクト型では、障害時に対象マルチキャストの転送経路をエンコーダ・デコーダを経由しない通常の経路に戻す必要がある。具体的には、デコーダからソース S 宛ての経路が消失したことを OSPF で広告するか、デコーダ・ルータ間の OSPF 隣接を解消する等の処理が要求される。デコーダ障害時は前述のいずれかの方法が適用できるが、エンコーダ障害時にはデコーダがこれを検出する必要がある。検出方法としては、エンコーダから死活監視のための試験パケットをマルチキャストアドレス G' を用いてデコーダへ定期的に送信する方法等が考えられる。可用性の観点からは、装置障害時に OSPF の経路更新に伴う配信断が発生し得るリダイレクト型よりも、配信が継続する冗長データ追加型が優れていると言える。

6. まとめ

本稿では、特定区間におけるマルチキャスト通信の品質向上を可能とする非インライン型の FEC 装置を提案した。本 FEC 装置では、マルチキャストの複製転送機能や経路制御機能を応用し、対象となるマルチキャストのみを特定区間の両端のルータに付加接続したエンコーダ、デコーダ経由で転送させ高信頼化する。これにより、一部の低信頼区間を高信頼化しエンド・エンドでの FEC 冗長化率を抑制できるため、様々なコスト・品質の回線が混在する環境に対して有効である。また、対象マルチキャスト以外の通信は、原則としてエンコーダとデコーダを経由しないため、可用性・転送性能の面で FEC 装置がボトルネックにならない。

試作機を用いて、2 種類の提案方式の動作確認と性能評価を行い、理論値に近い誤り訂正能力を確認した。また、使用する符号化行列のサイズに応じて、冗長データ追加型では復元パケットの到達順序逆転が大きくなり、他方、リダイレクト型ではパケット到達順序を保証するためのバッファリング機能によって遅延が大きくなった。クライアントで大きな到達順序逆転に対応する必要がないなどの理由により、リダイレクト型 FEC 装置の方が適用が容易であると言える。

謝辞 日頃御指導頂く KDDI 研究所秋葉所長に感謝する。

文 献

- [1] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing," IEEE/ACM Trans. Netw., vol.5, no.6, pp.784-803, Dec. 1997.
- [2] J. Nonnenmacher, E. Biersach, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," Proc. of ACM SIGCOMM'97, pp.289-299, Cannes, France, Sept. 1997.
- [3] "Forward error correction (FEC) building block," IETF RFC3542, Dec. 2002.
- [4] "The use of forward error correction in reliable multicast," IETF RFC3543, Dec. 2002.
- [5] "An RTP Payload Format for Generic Forward Error Correction," IETF RFC2733, Dec. 1999.
- [6] "RTP: A Transport Protocol for Real-Time Applications," IETF RFC1889, Jan. 1996.
- [7] 宮坂, 吉村, 鎌田, 照日, 植松, "無中断輻輳回避制御方式の有効性について," 信学技報 vol.1. 104 no. 433, NS2004-141, pp. 31-34, Nov. 2004.
- [8] "OSPF Version 2," IETF RFC2328, April 1998.
- [9] "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification," IETF RFC2362, June 1998.
- [10] "Internet Group Management Protocol, Version 2," IETF RFC2236, Nov. 1997.
- [11] "Catalyst 3560 スイッチ ソフトウェア コンフィギュレーションガイド Cisco IOS Release 12.2(25)SEE(OL-8553-01-J)," cisco systems, 2006.
- [12] R. G. Gallager, "Low-density parity check codes," IRE Trans. Theory, IT-8, pp.21-28, 1962.
- [13] "IP in IP Tunneling," IETF RFC1853, Oct. 1995.
- [14] IXIA, <http://www.ixiacom.com>.
- [15] L.Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," ACM Computer Communication Review., vol.27, No.1, pp.31-41, Jan.1997.