IP レベルの前方誤り制御プロトコル (IP-FEC) の実装と評価

IP Level Error Control Protocols (IP-FEC)

金井 久美子、松澤 茂雄、角田 啓治、江崎 浩
Kumiko KANAI, Shigeo MATSUZAWA, Keiji TSUNODA, Hiroshi ESAKI
(株) 東芝 研究開発センター
R&D Center, TOSHIBA Corp.

概要

本論文は IP レベルの前方誤り訂正 (FEC: Forward Error Correction) 制御を実行するためのプロトコル (IP-FEC) の実装と評価を行った。IP-FEC では、ネットワーク内での IP パケットの紛失・誤りの復元を、従来の IP パケットの再送による方法ではなく、前方誤り訂正技術を用いて実現する。IP-FEC では、複数のユーザーパケットの他に冗長パケットを転送し、冗長パケットによりネットワーク内で紛失・誤ったユーザーパケットの復元を行う。

本論分では、IP-FEC プロトコルの実装と、実ネットワーク環境での性能評価を行った。BSD/OS 上でのソフトウェア実装でも、2Mbps ~ 5Mbps のエンドエンドでのスループットを実現可能であることを確認することができた。さらに、広域の実ネットワーク上での性能評価を行い、IP-FEC がかなり有効な誤り制御方法であることを確認することができた。特に、IP パケット廃棄のパタンがバルス的な場合には、非常に高い確率で、廃棄パケットの復元を行うことができることを確認することができた。

1 はじめに

TCP/IP 技術を適用したインターネットやイントラネットは、トラヒック量や接続ノード数の増大が、地理的に拡大しながら、急激に進行している。すなわち、インターネットにおけるエンドエンドでの通信は、これまでのキャンパス内に閉じていた LAN 環境から、既に地球規模での WAN 環境でのデータ通信を必要としている。さらに、新しいアプリケーションの登場やインターネットの地理的な拡大は、これまでのベストエフォートサービスだけでは、ユーザおよびアプリケーションが必要とする通信品質を満足することができなくなってきている。具体的には、エンドエンドでの通信品質 (QoS:Quality of Service、CoS:Class of Service) の提供や、M-Bone に代表されるマルチキャストサービスの提供が必要となってきている。これらの新しいサービスを提供するは、新しい技術をインターネットに導入する必要がある。通信品質の重要なパラメータとしては、エンドエンドでのパケット転送遅延および帯域幅があげられる。さらに、近年、ポスト M-Bone サービスとして、誤りのない大規模マルチキャストサービスの必要性が認識されるようになってきた。IETF の RM(Reliable Multicast) WG では、エンドエンドに誤りのない大規模マルチキャストデータ転送サービスの提供を実現するアーキテクチャおよび制御方式の検討を進めている。その検討の中で、IP パケットレベルでの前方誤り訂正制御方式 [8][9] が有効な制御方式の 1 つとして提案されている。

コンピュータ間でのデータ通信においては、データの廃棄や紛失は基本的には許容されず、エンドエンドに誤りのないデータ転送を提供する必要がある。エンドエンドに誤りのないパケット転送を実現するための方法としては、TCP での再送制御のように、紛失・誤りパケットの送信元ノードからの再送によるパケットの再生が通常の方法であった。再送による誤り訂正制御は、リアルタイム性やエンドエンドでの遅延時間の上限を必ずしも要求しないこれまでのアプリケーション、あるいは、端末間の距離が短い場合には問題は発生しなかった。しかし、ネットワークの物理的に広がりが巨大化し、さらに、ネットワークに転送されるパケットがよりバースト的で高速になり、エンドエンドでのパケット転送遅延の絶対値および遅延揺らぎが大きくなってきている。したがって、エンドエンドにある遅延時間の上限を持つようなアプリケーションでは、従来の再送方式によるデータ再生制御では不十分であり、前方誤り訂正制御が有効であるといわれている [2] [3] [5] [6] [8]。さらに、マルチキャストサービスにおいては、マルチキャストツリーで観測されるパケットの廃棄・エラー確率は、ほぼ受信ノードの数に比例して大きくなってしまう。したがって、誤りのない大規模なマルチキャストサービスを実現するためのパケット生成制御として、従来の再送による方法を適用すると、再送パケットのオーバーヘッドが非常に大きくなってしまうことがよく知れれている [12] 。このような、誤りのない大規模マルチキャストサービスにおいて、前方誤り制御技術が、有効な制御技術になるという提案・評価も行われている。

前方誤り訂正 (FEC: Forwared Error Correction) 制御においては、ユーザデータに対して予め誤り訂正コード (訂正制御を行うための冗長データ) を付加して送信する。転送中に誤り・データ廃棄が生じた場合には、受信側ノードが訂正コードを用いて自律的に訂正する [2][3][8]。したがって、(1) 再送によるネットワークトラヒックの増加を防止することができ、かつ (2) 訂正すべきパケットの再送に必要な遅延時間が不要となる。

前方誤り訂正 (FEC) 制御の技術課題としては、(a) FEC の実ネットワーク環境での有効性と、(b) 十分な処理速度の実現可能性 の 2 つがあげられる。

1. FEC の有効性
   前方誤り訂正能力以上のパケットの廃棄や誤りが発生した場合には、前方誤り訂正制御によるパケットの再生を行うことができない。本論では、実トラヒック上での実験評価を行ない、FEC の有効性の評価を行った。
2. 処理速度
   前方誤り訂正のアルゴリズム自体は演算処理がかなり多くハードウェア実装されることが一般的である。本論文ではソフトウェア実装により前方誤り制御を実現し、スループットの評価を行った。

本論文で提案・評価している IP-FEC は IP レベルの前方誤り訂正制御である。データリンク技術に依存しない IP パケットレベルであるために、任意のデータリンクを経由したエンドエンドパケットフローに対して、前方誤り訂正制御を適用することができる。さらに、提案方式は、IP トンネリングに似たパケット転送手法を適用しているので、エンドエンドではなく、任意のノード (ルータおよびホスト) 間での前方誤り制御が適用可能となっている。

以下が、本論分の構成である。2 章では提案する IP-FEC プロトコルの仕様について述べ、3 章ではその実装の詳細に関して、プロトコルスタック、モジュール構成等を述べている。4 章でプロトタイプシステムを用いた IP-FEC の性能評価を行った。

2 IP-FECプロトコル

2.1 IP-FEC の動作概要

図 1 に、IP-FEC の動作概要およびプロトコルスタックを示した。

送信端末は実ユーザデータ (図中の data パケット) の他に誤り訂正用データ (図中の code パケット) を送信する。セグメント化された実ユーザデータや誤り訂正データは IP データグラムにカプセル化されているので途中のルータでは一般の IP データグラムと同じように IP ヘッダ情報のみを用いてフォワーディングされる。以下の節で送受信ノードでの符号化および復号化の手順を述べる。

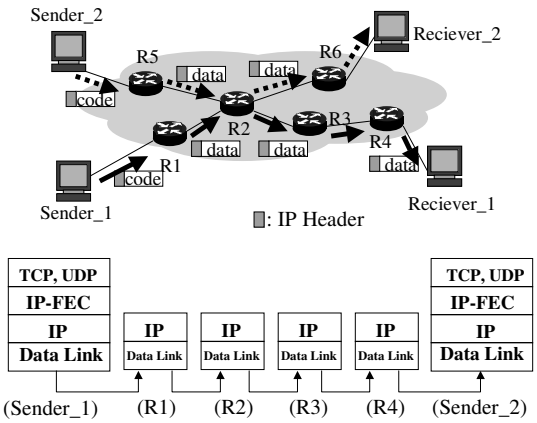

図 1: IP-FEC の使用形態

2.2 送信ノードの処理手順

図 2 に送信ノードの処理手順を示した。IP パケット (i.e., Original IP Packet) をネットワークレイヤ (IP) エンティティーから IP-FEC モジュールエンティティーが受け取る。IP-FEC エンティティーは、誤り訂正符号の生成、FEC ヘッダの付与し、インターリーブマトリックスを生成する。インターリーブマトリックスのデータは、IP パケット化 (IP-FEC パケットと呼ぶ) され、データリンクレイヤエンティティーに渡される。

1. Original IP packet を interleave matrix に書き込み、誤り訂正符号である Reed-Solomon code を interleave matrix 内の code part へ生成する。
2. Original IP packet に付与されている IP ヘッダ (IP-hdr) も含めて interleave matrix の縦方向に書き込む。
3. data part の最終列にパディング (PAD) を行う。
4. 行毎に誤り訂正符号化の計算をし、code part の同じ行に書き込む。
   (注)interleave matrix の収量は可変であるが最大値が決まっている。縦のサイズ × 横のサイズ = 一定値で、ただし、横のサイズは最大で、255−code part 列数までである。Reed-Solomon 符号は、1 バイトを 1 シンボルとする誤り訂正符号である。

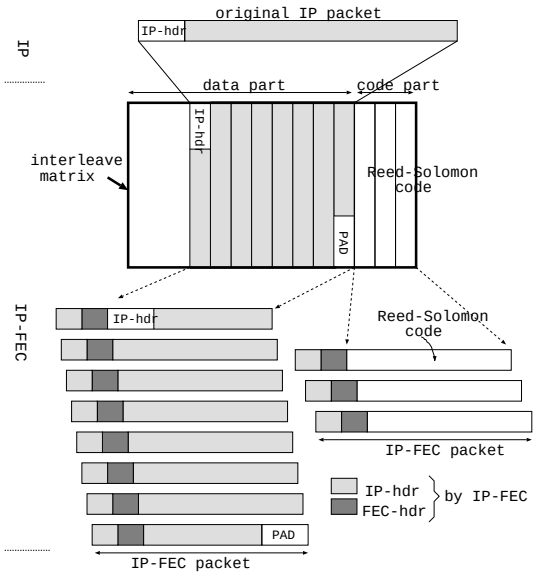

図 2: 送信ノードでの処理手順

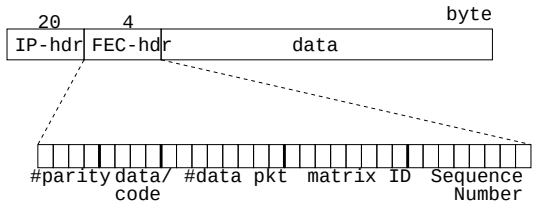

図 3: FEC-hrd の構成

5. data part の各列、code part の各列に対して、FEC ヘッダ (FEC-hdr) を付与する。FEC ヘッダは 4 バイトからなり、以下のフィールドを持つ (図 3 参照)。(括弧内はビット数)
   - SN (8): マトリクス (interleave matrix) 毎に初期化され付与されるシーケンスナンバである。同一のマトリクスに同じ SN を持ったパケットは存在しない。
   - #parity (4): マトリクス 1 行に対して何シンボルの訂正シンボルを付与しているかを示す。よって、各マトリクス毎に付与する訂正シンボル数が異なっていることも可能である。
   - data/code (4): ヘッダが付与されているペイロードがデータを含む (0001) のか、それとも訂正コードを含む (0010) のかを示す。
   - #data pkt (8): マトリクスが含んでいたデータ列数を示す。
   - Frame ID (8): マトリクス単位に送信側で順次インクリメントしていく ID である。(IP レイヤはコネクションレス通信なのでパケットの到着順序が入れ替わることがある)
6. IP カプセル化後の IP ヘッダを付与する。基本的には上位レイヤから渡された時に付与されていた original の IP パケットのヘッダのコピーである。変更点は、プロトコルフィールドを FEC に設定、パケットのトータル長、チェックサムである。
7. 出来上がった IP-FEC パケットをデータリンクレイヤに順次渡す。

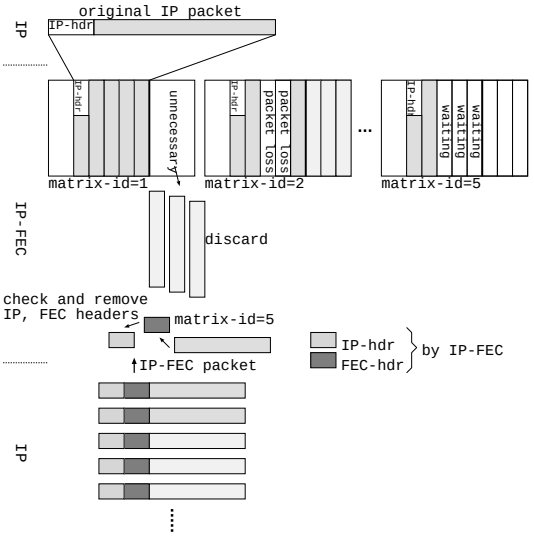

図 4: 受信ノードでの処理手順

2.3 受信ノードの処理手順

図 4 に受信ノードでの処理手順を示した。data part、code part の各々から成る IP-FEC パケットは FEC ヘッダと IP ヘッダが付与されており、一度 IP レイヤがパケットを受け取り、これを IP-FEC へスイッチングする。IP-FEC では、1 マトリクス分の IP-FEC パケットを処理し original IP パケットを組み立て、再度 IP レイヤに渡し、さらに上位のプロトコルへスイッチングしてもらう。このように受信ノード側では IP 処理を 2 度実行する構成となる。

1. データリンクレイヤからネットワークレイヤに渡された IP-FEC パケットは、一般の IP パケットと同じようにスイッチングされ、ヘッダのプロトコルフィールドに FEC と明記されているので IP-FEC へスイッチングされる。
2. IP-FEC では、カプセル化の IP ヘッダを取り除き、FEC ヘッダをチェックする。matrix_id と seq_nm からペイ

ロードの部分を適切な matrix の適切な列に書き込む。IP プロトコルの転送では正しい到着順序が保証されていないため、図 3 で示されているように複数のフレームを同時に保持できることが必要となる。

3. id=1 の matrix のように data part が全て揃えば original IP packet は組み立てが完了するので、code part の IP-FEC パケットである冗長符号は不必要なので廃棄する。
4. id=2 の matrix の場合は code part が到着しても data part が到着していないのでパケットロスした可能性がある。そこで誤り (消失) 訂正を行う機会を待つ。
5. パケットロスなしで original IP packet が出来上がった、もしくは、パケットロスはあったが訂正し original IP packet が再生できたら、再度、IP レイヤへ渡す。
6. パケットロスが多すぎて訂正能力を越えてしまっている場合は、訂正をあきらめフレームごと廃棄する。

2.4 IP-FEC 終端ルータ

IP-FEC はルータ上でも動作させることが可能である。基本的に IP-FEC はエンド-エンドホスト間で適用する誤り訂正方式であるが、パケットの中継装置であるルータが、IP-FEC 機能を終端することで、必ずしもすべてのホストが IP-FEC を実装する必要がなくなる。さらに、ネットワーク内の特定の区間の通信品質が十分でない場合に、ルータにおいて IP-FEC 機能の実現を行なうことにより、等価的にその区間の通信品質を向上さたのと同じ効果を実現させることが可能となる。

IP-FEC の符号化を行うノード (ルータあるいはホスト) は、IP-FEC の復号化を行うルータで正しく IP-FEC の復号化プロセスが実行されるように、転送される IP パケットの 宛先アドレスを FEC 化される IP パケットの宛先ではなく、IP-FEC の復号プロセスを実行するルータの IP アドレスにする必要がある。これは、トンネリング (i.e., IP-in-IP) と同等の作業である。

case (a) では、送受信双方のホストが IP-FEC を実装している通常のケースである。case (b) では、双方のホストのアクセスルータが IP-FEC を実行しており、IP パケットのフォワーディングの際に IP-FEC の符号化 / 復号化を行う。case (c)(d) では、ホストの一方が IP-FEC を実装しているケースである。

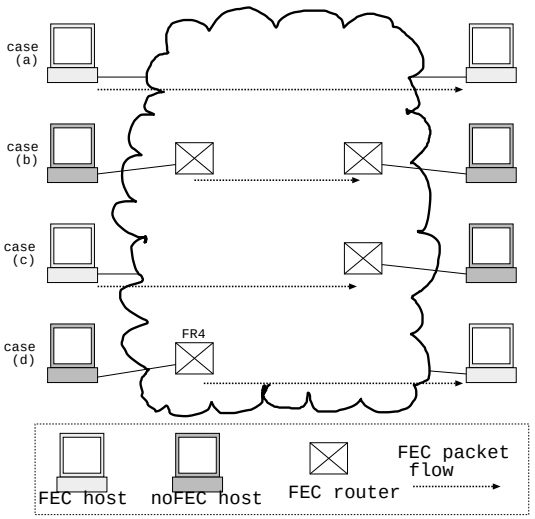

図 4: IP-FEC 終端ルータの利用形態

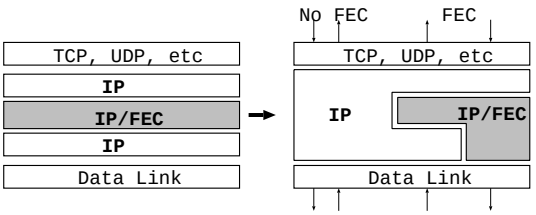

図 5: IP-FEC のプロトコルスタックと実装的レイヤ構造

3 IP-FEC プロトタイプ実装

3.1 IP-FEC 実装の概要

IP-FEC では、データ (Original IP データグラム) を複数の IP パケットに分割しそれらを用いて誤り訂正パケットを生成する。図 5 に、送受信ノードのプロトコル構造を示した。上位のトランスポートプロトコルおよび下位のデータリンクレイヤプロトコルに非依存な実装となっており、任意のアプリケーションおよび任意のデータリンクに対応することができる構造になっている。

3.2 IP-FEC プロトタイプ実装環境

IP-FEC を、BSD/OS Ver.3.0 のカーネル内に IP と隣接して実装した。誤り訂正符号は RS 符号 ($G(x) = x^8 + x^4 + x^3 + x^2 + 1$) で、パケット廃棄のみを訂正する。本論文での実装においては、ビット誤りは訂正しない実装としている (ビット誤りの訂正への対応は比較的容易である)。実装ハードウェアは、Pentium 166MHz の CPU を持つ、IBM 互換のパソコン (東芝製 PV3000 5166) である。メインメモリは、48MB(= 16MB+ 増設メモリ 32MB) である。

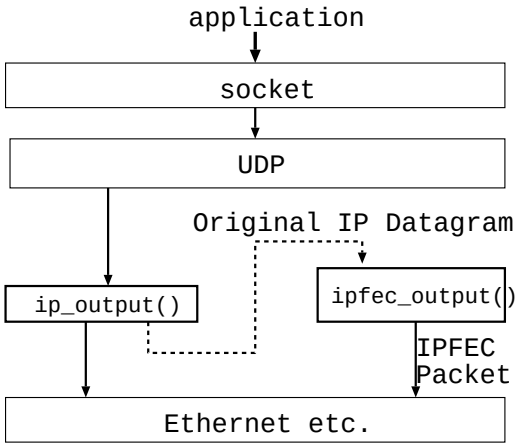

図 6: 送信側処理モジュール構成

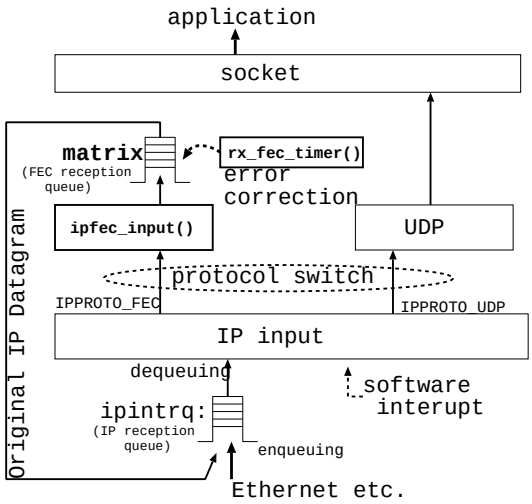

図 7: 受信側処理モジュール構成

3.3 送信側処理プロセス: ipfec_output.c

図 6 に送信ホストのモジュール構成を示した。ipfec_output() は ip_output() からコールされ、完成した IP データグラムが渡される (この IP データグラムをオリジナル IP データグラムと呼ぶ)。受け取ったオリジナル IP データグラムを固定長にセグメント化し、FEC ヘッダおよび IP ヘッダを付加する。IP ヘッダのプロトコルフィールドには FEC のプロトコル番号 (200) を書き込む。オリジナル IP データグラム (ヘッダも含む) 全体に対して誤り訂正符号を生成し、同様に FEC ヘッダ、IP ヘッダを付与する。こうして訂正符号を含む IP データグラムが生成されるが、その数は本実装では 1 ~ 4 となっている。

3.4 受信側処理プロセス: ipfec_input.c

図 7 に受信側ホストのモジュール構成を示した。プロトコルフィールドに FEC と記してある IP データグラムは、IP のプロトコルスイッチで ipfec_input() へ渡される。ipfec_input() 内に持つマトリクスにパケットをためていき、データを含むパケットが全て到着したらオリジナル IP データグラムをリアセンブルする。それ以降に到着した訂正コードを含むパケットは捨てられる。

また、rx_fec_timer() がポーリングしており、タイムアウトの時刻になっても全てのデータパケットが到着していなければ、誤り訂正を行ないオリジナル IP データグラムを再生する。また、タイムアウト時間に達する前にデータグラムの再生が可能になった場合に、rx_fec_timer() は再生を開始する機能も持つ。ただし、パケットロスが多すぎて訂正不可能な場合には諦める。(到着した全パケット数 $dlen + clen$ がデータのパケット数 $dmlen$ 以上であれば、訂正可能な範囲である。)

リアセンブルし正しく再生されたオリジナル IP データグラムは、再度、ipintrq に渡され、正規の IP input の処理を施される。

3.5 IP-FEC コマンド仕様

IP-FEC には指定すべきいくつかのパラメータが存在し、アプリケーションやネットワークの特性に応じて変更することができるように実装している。例えば、誤り訂正の強さ (= 重さ) やセグメントした場合のパケット長などである。

コマンドは ./fecset -[option] (引数) で、オプションは以下の通りになっている。符号化を行なう送信ホスト (ルータ) で使うコマンドと、復号化を行なう受信ホスト (ルータ) で使うコマンドとに分けられる。

送信ホスト (ルータ) のコマンド

-t のトリガは、6 つの引数並びが必要で、これはオリジナル IP パケットのヘッダと比較するものである。引数は、src-address, dst-addres, src-port, dst-port, ip-len, ip-proto。削除も同様である。また、疑似アドレスとは FEC ルータとしてパケットをフォワードする場合に用いる。-p のトリガは、4 つの引数並び (A,B,C,D) が必要で、src-address=A かつ dst-addres=B ならば、src-address=C, dst-addres=D とカプセル化してフォワードするものである。削除もこれも同様である。

- -c FEC パケット数の設定 (1 ~ 4)

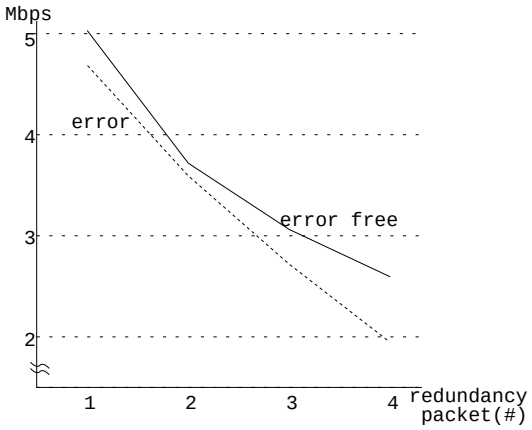

図 8: IP-FEC のエンドエンドスループット

- -l FEC パケット長の設定 (128 ~ 1024)
- -t トリガの追加
- -r トリガの削除
- -g トリガの全削除 (引数なし)
- -p 疑似アドレスの追加
- -s 疑似アドレスの削除
- -d 疑似アドレスの全削除 (引数なし)

受信ホスト (ルータ) のコマンド

受信側ではセグメント化したパケットをストックしていって、到着が遅れているのか、廃棄なのかをタイムアウトで判断する。エンドエンドが時間のかかる場合は相応に大きな値を設定する必要がある。

- -o マトリクス毎のタイムアウト時間の設定

4 IP-FEC の性能評価

4.1 スループットと基本性能評価

TCP/IP を用いて約 8kB のフレーム転送を行い通信速度を測定した。誤りのない環境で 2.6 ~ 5.0Mbps、誤り (パケット廃棄) のある環境で 1.9 ~ 4.7Mbps のスループットが得られた。(図 8 参照)

誤りのない場合、すなわち、パケットロスを検出せずに data part の IP-FEC パケットを全て受信できた場合には、復号処理を行わない。いくつかの IP-FEC パケットをリアセンブルし、original IP packet に戻すだけであるので、処理速度は送信時の符号化の速度に依存することになる。誤りのある場合には、誤りの検出されたマトリクス単位に復号化の処理を行わなければならない。誤りのある場合には、従順側での復号化処理が律速プロセスとなっており、エンドエンドスループットは、受信側ホストでの復号化の処理速度に依存することになる。

データ転送品質に関する主観的な評価としては、音声通信や動画像転送などのリアルタイムメディアのデータ転送アプリケーションに、IP-FEC 制御を適用し、その有効性を確認することができた。

- 音声転送での評価 vat(visual audio transport protocol) を用いて評価を行った。パケット廃棄がランダムに 5%発生する環境をエミュレートし、評価を行った。IP-FEC なしではパケット廃棄によって生じる音声の途切れが目立ち聞き取りにくいが、IP-FEC ありでは音声の途切れはおおむね回復し、人間の耳では廃棄のない状態と区別がつかないほど改善された。

- 動画転送での評価 H.263 を用いて、音声転送での評価と同じようにパケット廃棄がランダムに 5%発生する環境をエミュレートし評価を行った。H.263 では、パケット廃棄が生じると画面にはブロック誤りとして表示され、しばらく連続すると画面が固まってしまう。一方、IP-FEC を適用した転送ではパケット廃棄がない場合と同程度の画質で受信可能なことを確認することができた。

4.2 実ネットワークでの性能評価

WIDE 研究会では毎年 2 回の合宿形式の研究会が開催されている。この合宿研究会で構築された合宿ネットワークを利用し、リアルトラヒックを用いた IP-FEC プロトコルの実験・評価を行った。合宿参加者 250 名程度が合宿ネットワークを常時使用しており、各参加者がネットワークを用いた様々な活動を行なっている。

4.2.1 合宿ネットワークのトポロジー構成

図 10 に IP-FEC 評価実験に関するの合宿ネットワークの構成を示した。実験では、東芝→ WIDE ネット→合宿ネット という経路で、UDP パケットを送信し、受信状態を観測した。合宿ネットから外部へのゲートウェイでは ALTQ (Alternate queueing support for FreeBSD-2.2.[1,2,5,6]R) によるパケット転送の優先制御が行なわれている。

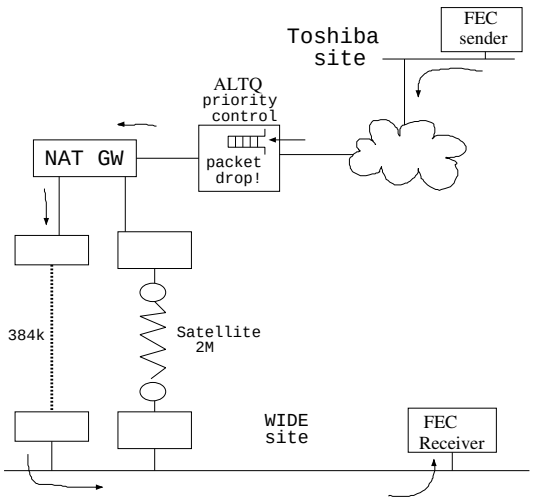

図 10: IP-FEC 評価実験関連のネットワーク構成

4.2.2 評価方法

東芝サイトの IP-FEC ホストから UDP パケットを一定の速度で送信し、合宿ネット内に設置した IP-FEC ホストで受信状態を観測する。東芝サイトと WIDE ネット間のリンクは、128Kbps である。

送信ホストから 32Kbps, または 16Kbps の送信速度で UDP パケットを送信する。 IP-FEC では、1500 バイトのパケットを 3 つに分割し、訂正用の冗長パケットを数パケット (1,2,3,4) 生成し、連続して送信する。パケットの送信間隔はおおよそ 3msec ごとであるが、マトリクスの境界では 8msec かかっていた。NAT ルータの付近には ALTQ というパケットの優先制御を行なうルータが設置されており、基本的には、IP-FEC のパケットは最低の優先度で扱うようにパラメータ設定を行った。すなわち、ALTQ ルータ内のパケットキューが増加した場合には、IP-FEC を適用しているパケット流のパケットが先に廃棄される。なお、外部から合宿ネットに入って来るトラヒックは衛星を経由することになっているのだが、輻輳して自然にパケットが落ちる環境を作るために、地上線の 384k を利用した。(図 10 参照)

4.2.3 評価結果

IP-FEC パケットが最低の転送優先度になっている環境では、より優先度の高いコネクションが存在している間、長時間連続的に IP-FEC のほとんど全てのパケットが廃棄されてしまうことになり、FEC を適用している意味がなくなってしまうことが分かった。一方、優先制御が行なわれていない環境 (全てのパケットが優劣なく同等に扱われる) では、IP-FEC が多くの条件下で良好に動作することを確認することができた。またバースト廃棄の前後では、約 50%程度の UDP パケットを訂正により復元させていることがを確認することができた。

以下に、評価実験における典型的な 3 つの環境での評価結果を示した。なお、誤り訂正用の冗長パケットは特に明示のない限り 3 ないしは 4 パケット分生成し送信している。

- パルス的な廃棄 (廃棄の継続時間が短い) が数回生じたケース。

	エラーなし	訂正 OK	訂正 NG	成功率
1	81	5	0	100
2	1081	24	4	85

- バースト的な廃棄が生じたケース。

	エラーなし	訂正 OK	訂正 NG	成功率
3	494	40	92	30
4	107	14	19	42
5	309	100	100	50
6	453	20	4	83

- 優先制御なし、3 時間分のトータル。16k で送信、冗長パケットは 2 個、タイムアウトは 3 秒に設定。

	エラーなし	訂正 OK	訂正 NG	成功率
7	13878	815	16	98

合宿ネットワークのトラヒックは予想ほどは混雑せず、定常的には廃棄が全く起きなかった。ただし、何らかの TCP コネクションがはられた時には大量データの移動が起こり、IP-FEC パケットは数秒から十数秒のバースト廃棄を経験した。また、http 等で生じると思われるパルス的なトラヒックによって、IP-FEC のパケットがパルス的に廃棄される場合が観測された。ALTQ で優先制御を行なっている場合、FEC パケットは最低の優先度に設定されているので、ほとんど廃棄されてしまうのである。

通信量が帯域以上になるような輻輳が継続する場合、UDP パケットが数十個まるまる落ちてしまうが、これらの誤りに関しては FEC による訂正は全く不可能である。よって、バースト廃棄が継続して起こっている状況で、IP-FEC が何らかの誤り訂正を行なえるのはバーストの前後の廃棄の部分であり、訂正を試みた場合の訂正成功率は概ね 50%である。小さなパルス的な廃棄 (= FEC の訂正能力内に収まる) の場合は訂正 NG がなく、完全に訂正できることも確認できた。

4.2.4 評価結果に対する考察

今回の評価実験は ALTQ の制御ポリシーとパケットの廃棄パターンの特長から 3 つの状況に分類できる。

1. ALTQ 優先制御ありでのパルス的なパケット廃棄
   訂正確率からも分かるように、パルス的な廃棄には十分対応できることを確認することができた。パケット廃棄のパルスの大きさが IP-FEC の誤り訂正能力内ならば、もちろん完全に再生を行なうことができる。なお、ここで数値として挙げていないが、まれに起こる小さなパルス的な廃棄には 100%回復できている。
2. ALTQ 優先制御ありでのバースト的なパケット廃棄
   優先制御ありでバースト的な廃棄の場合には、訂正成功率はかなり低い。これは、優先制御のために IP-FEC のパケットが大量に廃棄されており、訂正能力を越えたパケット廃棄が発生しているものと考えられる。なお、数値として表示していないが、バースト廃棄では UDP パケットが数十個連続して廃棄されてしまっており純粋な IP-FEC のみでは全く対応することができない。
3. ALTQ 優先制御なし
   優先制御なしで、IP-FEC のパケットも他のパケットと同様に優劣なく扱われているので、廃棄量や廃棄間隔は IP-FEC のレベルから見ると比較的バースト性は強くない。訂正成功率も 100%ではないがかなり高く、良好な状態である。廃棄の間隔が疎、よりランダムになるので IP-FEC のアルゴリズムには適する運用条件である。

今回の性能評価実験の結果から、IP-FEC はランダムなパルス的な廃棄には強いということは確認することができたが、バースト的な廃棄への対応には問題があることが明らかになった。したがって、バースト的なパケットは行きに対応するためには、他の制御方法との組合せが必要であると考えられる。バースト的なパケット廃棄に対応するための方法としては、以下の 2 つのアプローチが有効ではないかと考えられる。

1. アクティブキュー管理制御 [12] との共存ネットワークが輻輳状況に陥らないように、輻輳が発生する前にパケットを強制的に廃棄させるなど、アクティブキュー管理制御が議論されている。例えば、RED(Random Early Ditection)[10] は、積極的に輻輳が発生する前にパケットを選択的に廃棄させることで、輻輳に陥らないようにホストから転送されるパケットの量を制御する方式である。アクティブキュー管理制御を適用することで、パケット廃棄パターンはよりランダム化けし、パケット廃棄の継続時間が短くなることが期待される。したがって、アクティブキュー管理制御との共存により、IP-FEC が良好に動作する環境が提供される可能性があると考えられる。
2. レート制御の適用
   TCP におけるフロー制御に、レート制御を導入しようという動きがある [13]。ウィンドー幅の大きさに応じて、ホストから転送するパケットの転送レート (= パケットの送信間隔) を変化させる制御である。パケットの転送間隔が保たれるので、結果的に、パケットの廃棄パターンをよりランダム化させる可能性がある。すなわち、レート制御の導入により、パケットの廃棄パターンがよりランダム化されることで、IP-FEC が良好に動作する環境が提供される可能性があると考えられる。

IP-FEC では、100 統合が必要である。 ARQ との統合を行う場合、IP-FEC の訂正能力をネットワークの状態に対応させて動的に適応させることができれば、効率的な冗長パケットの転送を実現させることが可能となる。すなわち、ネットワークが混雑していない状況では、少ない冗長パケットの転送で十分であるが、ネットワークが混雑するようになると、冗長パケットの量を増加させ、結果的に、ユーザデータに対するオーバーヘッド (再送パケット + 冗長パケット) を極小化させることができると考えられる。ECN(Explict Congestion Notofication) 制御 [14] は、陽に、エンドホストに対して、ネットワークの輻輳を通知する制御方法であり、ECN 制御の情報を用いて、冗長パケットの量を制御するアプローチが考えらる。

5 まとめ

IP パケットの廃棄を復元するための前方誤り訂正プロトコル (IP-FEC) の方式提案、プロトタイプ実装、さらに実ネットワークを用いた性能評価を行った。2 ~ 5Mbps 程度の実効スループットを実現することができ、動画像通信を含む多くのアプリケーションで利用可能、かつ有効であることを確認することができた。 さらに、広域の実ネットワーク上での性能評価を行い、IP-FEC がかなり有効な誤り制御方法であることを確認することができた。特に、IP パケット廃棄のパタンがパルス的な場合には、非常に高い確率で、

廃棄パケットの復元を行うことができることを確認することができた。今後は、他の制御方法との統合や協調により、現在の IP-FEC の性能改善を行っていく予定である。

参考文献

[1] 江崎 浩、金井 久美子、角田 啓治、“誤りのない大規模マルチキャストアーキテクチャと前方誤り訂正技術の応用”、信学技法 SSE97-130.

[2] A. J. McAuley, ” Reliable Broadband Communication Using a Burst Erasure Correcting Code”, ACM, SIGCOMM90, pp.297-306, 1990.

[3] B. Ernst, ” Performance Evaluation of Forward Error Correction in an ATM Environment”, IEEE JSAC, May 1993.

[4] H. Esaki, Y. Tsuda, K. Kanai, “Evaluation of High Speed Multimedia Communication Architecture in ATM Networks”, IEICE Trans. Communications, Vol.E77-B, No.11, November 1994.

[5] J.C. Lin and S. Paul, “RMTP: Realiable Multicast Transport Protocol”, INFOCOMM’96, pp.1414-1424, San Francisco, CA, March 1996.

[6] J. Nonnenmacher, E. Biersack and D. Towsley, “Parity-Based Loss Recovery for Reliable Multicast Transmission”, SIGCOMM’97 April 1997.

[7] K. Kanai, R. Grueter, K. Tsunoda, T. Saito and H. Esaki, “Forward Error Correction Control on AAL5 : FEC–SSCS”, IEEE ICC’96 pp.384-391.

[8] L. Rizzo, “Effective erasure codes for reliable computer communication protocols”, Computer Communication Review, April 1997.

[9] L. Rizzo and L. Vicisano, “A Reliable Multicast data Distribution Protocol based on software FEC techniques”, IEEE workshop on th HPCS97, June 1997.

[10] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE ACM Trans. on Networking, August 1993.

[11] S. Floyd, V. Jacobson, C. Liu, S. McCanne, L, and Zhang, “A Reliable Multicast Framework for Lightweigt Sessions and Application Level Framing”, IEEE ACM Trans. on Networking, 1996.

[12] B.Braden, D.Clark, J.Crowcroft, et.al., “Recommendations on Queue Management and Congestion Avoidance in the Internet”, IETF RFC2309, April 1998.

[13] Craig Partridge, ”ACK Spacing for High Delay-Bandwidth Paths with Insufficient Buffering”, draft-partridge-e2e-ackspacing-00.txt, July 1997

[14] S.Flyod, ”TCP and Explicit Congestion Notification”, ACM Computer Communication Review, V. 24 N. 5, October 1994, p. 10-23.