社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＤＳＭ－33　(15)
2004／5／14

IPストリーム伝送のための誤り訂正機能をもつアプリケーションゲートウェイの開発

岸田　崇志†　鈴木　宏治‡1　河野　英太郎†　前田　香織†

†広島市立大学大学院情報科学研究科，‡広島市立大学情報科学部

〒731-3194　広島市安佐南区大塚東 3-4-1

E-mail:　{takashi, suzuki}@v6.ipc.hiroshima-cu.ac.jp,　{kouno, kaori}@ipc.hiroshima-cu.ac.jp

あらまし　広帯域ネットワークの普及が進み，TV 会議などに利用できるストリーム伝送アプリケーションも数多く存在する．しかし，IP ネットワーク上でのストリーム伝送は UDP や RTP を用いて通信を行うため，パケット損失やジッタによる影響を大きく受ける場合が多く，品質の確保が難しい．本研究では，既存のストリーム伝送アプリケーションとともに使用でき，Reed-Solomon FEC による誤り訂正機構を持つアプリケーションゲートウェイを開発し，評価を行った．また，様々なネットワーク構成や性能をもつ環境でストリーム伝送が高品質かつ実用的に行えるためのゲートウェイの機能の検討も行う．

キーワード　ストリーム伝送，FEC，アプリケーションゲートウェイ，インターネット

Development of an Application Gateway with Forward Error Correction for IP stream

Takashi KISHIDA†　Koji SUZUKI‡2　Eitaro KOHNO†　and　Kaori MAEDA†

†Graduate School of Information Sciences, Hiroshima City University,

‡Information Sciences, Hiroshima City University

3-4-1 Ozuka-Higashi, Asaminami-ku, Hiroshima, 731-3194,　Japan
E-mail:　{takashi, suzuki}@v6.ipc.hiroshima-cu.ac.jp,　{kouno,kaori}@ipc.hiroshima-cu.ac.jp

Abstract　Streaming data use UDP and RTP in many cases on IP networks.　So, it is difficult to assure quality since they are often affected by packet losses, jitter and delays.　In our study, we developed an application gateway using Forward Error Correction mechanism with Reed-Solomon code to recover packet losses and evaluated it.　Some advantages of this gateway are that it can be used with the existing stream transmission tools and improves quality of moving pictures.　Also, we consider necessary features of this gateway to transmit stream data practically and high quality in various network configurations and performances.

Keyword　Stream transmission,　FEC,　Application gateway,　the Internet

1. はじめに

近年，広帯域ネットワークの普及とともに TV 会議システムが広く利用されている．Netmeeting[1]，Polycom ViewStation[2]，vic/rat[3]などや映像の圧縮方式に Mpeg2 や Mpeg4 を用いた高品質な TV 会議システムの製品も増えてきている．しかし，インターネットはベストエフォート型のネットワークであるため遅延，パケット損失，ジッタなどが生じてしまう．特に，TV 会議システムでは IP 上で UDP や RTP でデータを送受するものが多く，パケット損失やジッタによる影響が大きくなり，その影響で映像や音声の品質を落とすことになる．そこで，エンドホストのアプリケーションに誤り訂正機能を実装し，パケット損失を回復する研究[4][5]が進んでいる．しかし，エンドホストで誤り訂

[1] 平成 16 年 4 月より，静岡大学大学院情報学研究科
[2] Graduate School of Information, Shizuoka University

正を行うことは以下のような問題点が挙げられる．

- エンドホストのアプリケーションを変更することは容易ではない
- 特定のアプリケーションに限定される
- エンドホストに余分な負荷をかける

エンドホスト以外に誤り訂正機能を実装する研究[6][7]も進んでいるが，これらの研究も特定のアプリケーションでの利用を考えており，利用範囲は限定されてしまう．

そこで，本研究では，既存のアプリケーションで利用できるパケット回復用のアプリケーションゲートウェイを開発し，評価を行う．このようなシステム構成とすることでパケット損失が生じる信頼性の低いネットワークなど特定の区間に設置して信頼性を確保するといった利用方法も可能になる．

本稿では，2 章でシステムの特徴やシステムの概要について述べる．さらに 3 章で開発したシステムに関しての評価を行い，4 章で IP ストリーム伝送システムを使用する際に検討すべき事項に関しての考察を行い，5 章では今後の課題とまとめを述べる．

2. 誤り訂正機能をもつアプリケーションゲートウェイ

2.1. システムの特徴

本システムの特徴を以下に挙げる．

1) ゲートウェイでの誤り訂正

本システムはゲートウェイに誤り訂正機能を実装しているので，特にエンドホストに手を加えることなく様々な TV 会議システムを利用できる．

2) 可変長ペイロードの通信に対応

アプリケーションによってはペイロードが固定長でなく，可変長のものもある．本システムは，損失したパケットのペイロードサイズも回復できるようにすることで可変長パケットの通信にも対応している．

3) 双方向通信に対応

リアルタイムに複数の人が会議や会話をするアプリケーションを想定しているため双方向通信に対応している．

4) バースト損失に強い

インターネットで起きるパケット損失はバースト的なものが多いため，パケット損失への対応としてバースト損失に強い Reed-Solomon 符号を用いた FEC(Forward Error Correction) を採用した．Reed-Solomon 符号は，ブロック符号の 1 つであり，送信すべき情報を連続した複数のビット単位で(シンボル)で分割し，符号化，復号化を行う誤り訂正符号である．1 ブロックは N 個のシンボルによって構成され，そのうち K 個がデータシンボルであり，$N-K$ 個が冗長シンボルである場合，(N, K)の冗長度と呼ぶ．

2.2. システム構成と動作概要

図 1 にシステム構成とパケットの流れを示す．開発したゲートウェイ(図 1 では Gateway 1 と Gateway 2 に相当)はインターネットなどのパケット損失が起こり得る信頼性の低いネットワーク(Lossy network と称する)の出入り口に設置される．ホストとゲートウェイは LAN など比較的信頼性のあるネットワークを想定しており，この間ではパケット損失がほとんど起きないことを想定している．Lossy network でパケット損失が生じた際はゲートウェイの誤り訂正機能によって，エンドホスト間(図 1 では Host A と Host B に相当)の通信品質を向上できる．

Host A から Host B への通信動作概要を以下に示す．

①Host A はアプリケーションの出力データを Gateway 1 に送信する．

②Gateway1 は Host A から受信したメディアパケットにヘッダを付けて拡張パケットを生成する．また，FEC エンコードにより，冗長パケットを生成し拡張パケットとともに送信する．

③Gateway 2 は Gateway 1 から受信した拡張パケットと冗長パケットから FEC デコードにより，損失したパケットを回復させ，拡張パケットからヘッダを取り除きメディアパケットに復元し，Host B に送信する．

④Host B は Gateway 2 からメディアパケットを受信する．

Host B から Host A への通信も同様に並行して行われる．

現在，本システムは RTP/RTCP を対象に実装を行っている．このとき，メディアパケット(RTP)に併設する制御用パケット(RTCP)はゲートウェイでメディアパケットと同様にヘッダを付加して拡張パケットとして生成されるが，メディアパケットに比べ送信頻度が少ないため冗長パケットを生成しない仕様となっている．

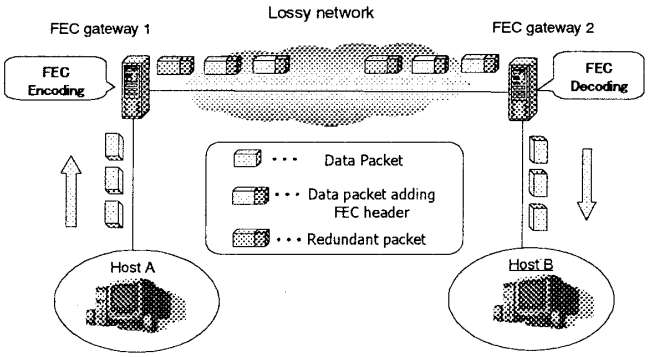

図 1. システム構成とパケットの流れ

2.3. パケットフォーマット

(a) ヘッダ構成

ゲートウェイ間での通信に用いられるヘッダの構成を図2に示す．ここでは，ストリームデータの転送プロトコルとしてRTP，ストリームデータの制御プロトコルとしてRTCPを想定している．ゲートウェイ間を通信するパケットにはゲートウェイヘッダが付けられる．ゲートウェイヘッダは基本ヘッダと拡張ヘッダにより構成される．基本ヘッダは8byteでゲートウェイを中継する全てのパケットに付けられる．拡張ヘッダはゲートウェイに付加機能を加える時にだけ付けられる．本研究では，信頼性を高めるためにReed-Solomon FECを付加機能として実装した．そのため，拡張ヘッダとしてFECヘッダを定義した．また，ゲートウェイで基本ヘッダが付加されたものを拡張パケットと呼ぶ．ゲートウェイでは受信したRTP/RTCPパケットにゲートウェイヘッダを付加して拡張パケットとする．本システムでは，RTP/RTCPを想定して実装を行っているが，RTP/RTCP以外のプロトコルにも対応できる．図2の例のようにRTPパケットの場合は，FECにより信頼性を高めるため，基本ヘッダとともに拡張ヘッダのFECヘッダ(12byte)を付けることで拡張パケットとなる．一方，RTCPパケットは基本ヘッダのみを付けることで拡張パケットとなる．

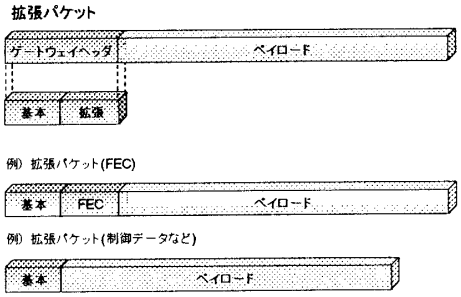

図2．拡張パケット

(b) ヘッダフォーマット

ヘッダフォーマットを図3に示す．図3の点線より上部が基本ヘッダで，下部が拡張ヘッダ(FECヘッダ)である．表1は図3の基本ヘッダ部分の各フィールドの説明であり，表2は図3中の拡張ヘッダの説明である．現在，拡張ヘッダはFECヘッダのみだが，今後，様々な付加機能を加えていく予定である．

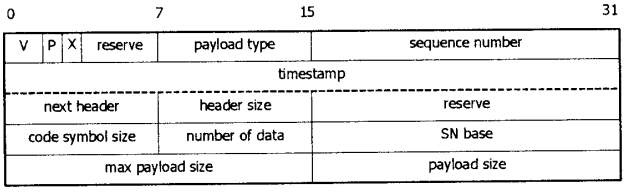

図3．ヘッダフォーマット

表1．基本ヘッダ

フィールド名	Size(bit)	説明
バージョン(V)	2	バージョン（現在は1）
パディング(P)	1	パディング
拡張ビット(X)	1	拡張ヘッダがある場合は1．なければ0．
予約(reserve)	4	予約フィールド
ペイロードタイプ (payload type)	8	メディアパケット(FECヘッダ)の場合は33．冗長パケット(FECヘッダ)の場合は44．
シーケンス番号 (sequence number)	16	パケットのシーケンス番号を格納する．
タイムスタンプ (time stamp)	32	エンドホストからのパケット送出間隔を格納する．

表2．拡張ヘッダ

フィールド名	Size(bit)	説明
後続ヘッダ (next header)	8	ゲートウェイ拡張ヘッダの後にさらに拡張ヘッダが続く場合に使用する．後続ヘッダがない場合は99とする．(デフォルトでは99)
拡張ヘッダ長 (header size)	8	拡張ヘッダのサイズをバイト単位で表す．
予約(reserve)	16	予約
コードシンボル長 (code symbol size)	8	Reed-Solomon符号のコードシンボル長を格納する．
データシンボル数 (number of data)	8	Reed-Solomon符号のデータシンボル数を格納する．
SNベース (SN base)	16	Reed-Solomon符号の1ブロック内の最小シーケンス番号を格納する．
最大ペイロード長 (max payload size)	16	エンドホストのアプリケーションが利用する最大のペイロード長を格納する．
ペイロード長 (payload size)	16	パケットのペイロード長を格納する．

2.4. 動作確認をおこなったアプリケーション

現在，本システムではpayloadが1500byte以下，及び使用ポートが2ポート以内，トランスポートプロトコルにUDPを使用しているアプリケーションに対応している．現在のところ以下のアプリケーションと併用して，パケット損失回復の動作を確認している．

- VIC/RAT
- VideoLAN[9]
- DVTS[10]
- Robst[11]

上記の内，VIC/RATやVideoLANのような秒間500パケット程度を送出するアプリケーションであればCeleron300MHzでも動作を確認している．今後，これらのアプリケーションを用い実ネットワークでの評価を行うことを検討している．

3. 評価

3.1. 評価項目

本システムの評価としてReed-Solomon FEC処理によるパケット回復能力の測定とオーバヘッドの測定を行った．パケット回復能力の測定は実測値と理論値での比較を行い．オーバヘッドの測定としてFEC処理の処理時間の測定と帯域増加率の測定を行った．

3.2. 実験構成

実験は図 4 で示した構成で行った．表 3 に使用した機器の仕様を示す．実験では，Host A と Host B で VLC(VideoLanClient)[9]を起動し，Host A から Gateway 1 に向けてパケットを送信し，Gateway 1 でエンコード処理を行いパケット損失生成用 PC[8]に送信する．パケット損失生成用 PC ではパケット損失を発生させ Gateway 2 へ送信し，Gateway 2 でデコード処理を行い Host B に送信する．VLC では，圧縮方式を mjpeg，帯域を 3 Mbps，解像度を 640×480 に設定した．この設定で RTP パケットを 10 万パケット送信した．

表 3．実験環境

	HostA/B	Gateway1	Gateway2	パケット損失生成用 PC
OS	Windows XP	Fedora core 1	Redhat linux 8.0	VineLinux 2.6
CPU	PentiumM 1.7GHz	Pentium4 3.2GHz	Pentium4 3.2GHz	PentiumIII 1GHz
Memory	512Mbyte	1024Mbyte	1024Mbyte	512Mbyte

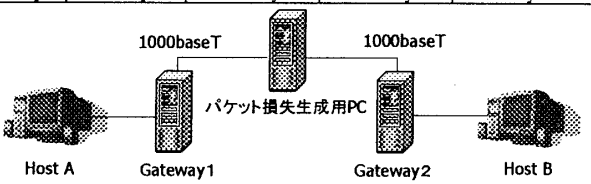

図 4．測定環境

3.3. パケット回復能力

測定環境(図 4)中のパケット損失生成用 PC では，2,4,6,8,10%の確率でパケット損失を発生させた．N は 15 固定とし，K を 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，算出した理論値(算出方法は[8]の通り)と比較した．図 5 に示すようにそれぞれの冗長度において FEC復元前のパケット損失率とFEC復元後のパケット損失率の理論値と実測値はほぼ等しくなり文献[4][5]と同様に本システムのパケット回復プロセスが正しく動作していることが確認できた．

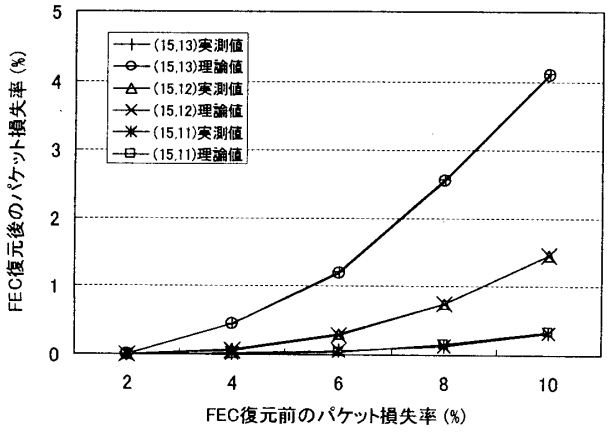

図 5．実測値と理論値の比較

3.4. システムのオーバヘッド

本システムではパケットのゲートウェイ中継時に，拡張ヘッダや冗長パケット分の帯域増加と遅延が生じる．本節では，本システム使用時の帯域増加率と FEC 処理による遅延時間を測定した．

3.4.1. 帯域増加率

ゲートウェイ間の通信は，冗長パケットのサイズ(最大ペイロードサイズ)と拡張パケットのヘッダのサイズ分帯域が増加する．帯域の増加率 B は(1)式により算出した．mps が最大ペイロードサイズ，ext は拡張パケットのサイズ(20byte)，(n, k)は冗長度で k は 1 ブロック内のメディアパケット数，x は受信したパケットのサイズである．最大ペイロードサイズは 1 ブロック内の最大ペイロードサイズで，このサイズで Reed-Solomon エンコードの演算処理を行う．受信側で各パラメータを測定し，(1)式に適用することで帯域増加率を求めた．測定は，冗長度が(15,13),(15,12),(15,11)の場合について行った．測定結果を表 4 に示す．測定結果は測定したパケット数 10 万個分の平均値である．

今回の測定では，圧縮方式が mjpeg を使用し，ペイロードは固定長で 1328byte である．これより(1)式の x 及び mps は 1328 であり，B と実測値は同じになる．しかし，ペイロードが可変長の場合(1)式で求めた B よりも値は大きくなる．これは冗長パケットが FEC エンコードを行う際に最大のペイロードサイズに合わせてしまうためである．

$$B = \frac{(mps + ext) \times (n-k) + \sum_{i=1}^{k} (x_i + ext)}{\sum_{i=1}^{k} x_i} \quad (1)$$

表 4．帯域増加率

冗長度	帯域増加率
(15,13)	1.171
(15,12)	1.269
(15,11)	1.384

3.4.2. 処理遅延

FEC処理を行うことにより処理遅延が増加してしまう．そこで処理遅延についての評価を行った．パケットの流れを図 6 に示す．図 6 より，エンコードプロセスで遅延を大きくする要因として影響を及ぼす処理は，冗長パケットを生成するときに行われる FEC エンコード処理(図中の delay1)である．また，デコードプロセスで遅延を大きくする要因として影響を及ぼす処理は，パケットを回復するときに行われるバッファリング処理(図中の delay3)と FEC デコード処理(図中の delay2)である．パケット損失生成用 PC ではパケット損失を 0%にして測定を行っている．処理遅延の測定は該当の箇所を gettimeofday()関数を用いて測定を行

った.測定は冗長度が(15,13),(15,12),(15,11)の場合について行った.

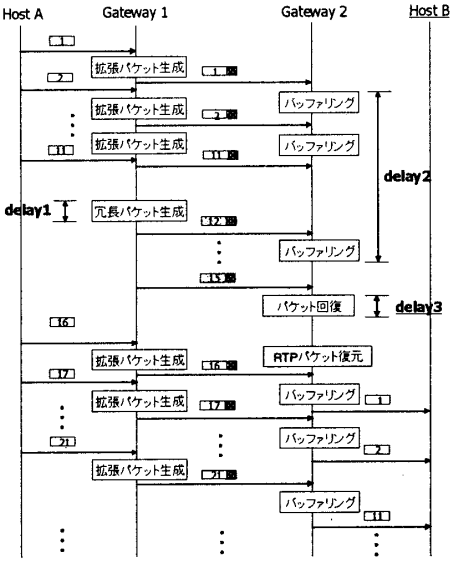

図 6. パケットの流れ

(a) エンコードプロセスにおける処理遅延

FECエンコードによる処理遅延の測定は，FECエンコードの処理の時間(delay1)を測定した．また，冗長パケットを生成する時間をプロセス全体の遅延として測定した．図 7 に測定結果を示す．x 軸は冗長度で，y 軸は遅延時間である．

FEC処理を行うにあたって,1 ブロック中に 1 度 FEC エンコード処理を行う．図 7 より，FECエンコードの占める時間の割合が多いことが分かる．また，FECのエンコード遅延時間は冗長度の増加とともに増加することが分かる．

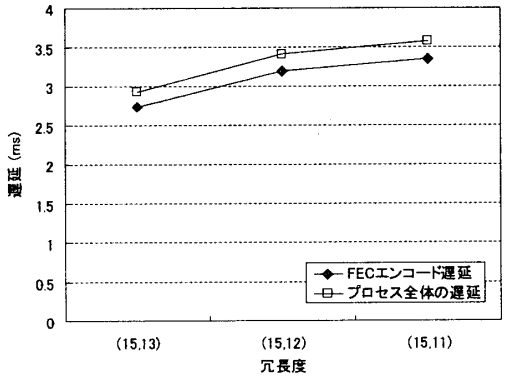

図 7. エンコードプロセスで生じる遅延

(b) デコードプロセスにおける処理遅延

FEC デコードによる遅延は FEC デコード処理(delay3)の時間を測定した．また，バッファリングによる遅延はFECデコード処理に必要なN個分(N=15)のパケット数をバッファする時間(delay 2)を示す.1 ブロックの処理の時間をプロセス全体の遅延として測定した．

測定結果を図 8 に示す．x 軸は冗長度で，y 軸は遅延時間を示している．図 8 に示す通り，FECのデコード遅延は冗長度とともに増加した．また，バッファリングによる遅延時間は冗長度の増加とともに減少した．

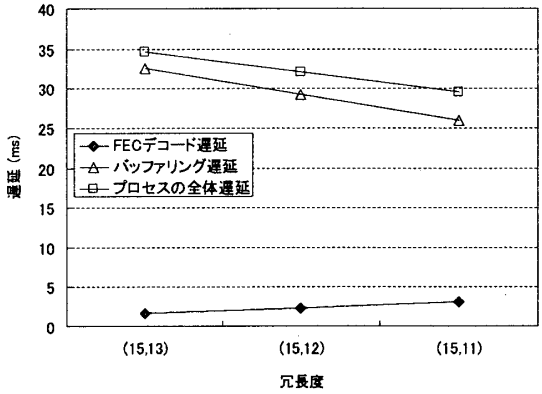

図 8. デコードプロセスで生じる遅延

(c) パケット損失による遅延増加量

FECエンコード処理やバッファリング処理はパケット損失により遅延が増加することはないが，FECデコード処理はパケット損失を起こすと計算量が多くなるため，遅延が増加すると考えられる．パケット損失発生用 PC で 2,4,6,8,10%のパケット損失を起こし，それぞれのパケット損失のときの FEC デコード処理の遅延をデコードプロセスの遅延を測定した方法と同じ方法で測定した．図 9 に測定結果を示す．x 軸はパケット損失率で，y 軸は遅延時間である．これより，冗長度とパケット損失の増加により Reed-Solomon 演算の計算量が増えたため，遅延の増加量に変化がでていることがわかる．

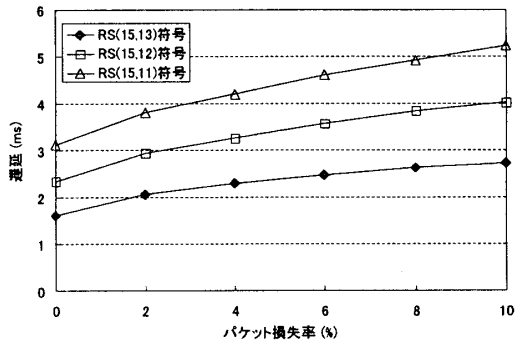

図 9. パケット損失で生じる遅延

3.4.3. 考察

まず，帯域増加率について考察する．高い冗長性をもたせると帯域増加率は大きくなる．また，エンドアプリケーションが広帯域を利用すればするほど，帯域使用率も増加するので，ネットワークに適する冗長性の持たせ方に関して検討を行うことも重要である．

次に，遅延について考察を行う．FECエンコードとFECデコードで生じる遅延は，冗長度の増加とともに増加するが,バッファリングで生じる遅延は減少する．バッファリングによる遅延はエンドホストのアプリケーションのパケット送出間隔に依存し，おおよそ，[送出間隔]×[1ブロック内のデータパケット数(K)]となる．

mjpeg(実験では VLC で送信)の送出間隔は約 2.7ms なので図 8 よりバッファリングによる遅延は約(2.7×K) msec となっている．圧縮方式を H.261 や H.263 とした場合，mjpeg より送出間隔が大きくなるため遅延は増える．また，帯域を 3 Mbps 以下にすると，パケットの送出間隔が大きくなるため遅延は増える．このように遅延はエンドホストのアプリケーションによっても変化する．

図 8 や図 9 より，パケット損失が多い場合や FEC の冗長度を大きくすると，遅延が増加する傾向にあることから，多くのパケットを送信するエンドアプリケーションでは，本システムで FEC 処理可能なパケット数に限界が生じる．例えば，図 9 よりパケット損失 10% で冗長度が(15,11)の場合，1 ブロック(15 パケット)に必要な処理遅延は約 5.2ms なので，FEC デコード可能なパケット数は 1 秒間約 2863 パケットである．今回の実験では，ペイロードが 1328byte なので，約 29.7Mbps の帯域が使用できることとなる．ただし，実際にはバッファリングやその他の処理時間も加算されるため，利用可能な帯域はこの値よりも小さくなる．

4. IP ストリーム伝送を行う上での検討事項

本章では，様々なネットワーク構成や性能をもつ環境でストリーム伝送が高品質かつ実用的に行えるためのゲートウェイの機能の検討を行う．

4.1. ネットワークの性能

ネットワークによっては，パケット損失，ジッタ，遅延などのネットワークの性能がストリーム伝送に適していない場合がある．本稿で提案するアプリケーションゲートウェイは，このうち，パケット損失に関して，FEC によるパケット回復機能を実装した．

アプリケーションによってネットワークのジッタや遅延の影響を大きく受ける．ジッタが大きなネットワークを通過したパケットはエンドシステムのアプリケーションのジッタバッファなどによっても吸収されるが，ジッタが大きすぎるとそのパケットがエンドシステムで廃棄されてしまう．こうした問題を軽減するため，本システムの追加機能としてジッタによる影響を少なくする機構も検討している．

4.2. ネットワークの構成

ストリーム伝送を行う際，ネットワークの運用システムを変えることなくストリーム伝送を行うことも重要となる．ネットワークによっては外部ネットワークと通信を行う際にポートに制限がかけられている場合もある．こうしたネットワーク構成でストリーム伝送を行いたい場合，ファイアーウォールの設定をその利用だけで変更することはシステムの運用面からも容易なことではない．そこで，本提案システムのようなゲートウェイを配置し，ストリームアプリケーションで利用している特定のポートを変換して通過させる機能も必要と考えている．

その他，多地点間で通信を行いたい場合，マルチキャスト利用要望があるが，ネットワークの構成によってはマルチキャスト通信も行えない環境も多い．本システムではマルチキャストトンネル機能も実装中である．これにより，本システム間は FEC エンコードを行ったパケットをユニキャストで複数地点に送信するポイントマルチポイント通信も可能である．さらに，FEC デコードを行ったのち LAN 内のエンドホストへマルチキャスト若しくはポイントマルチポイントでの配信も可能となるような機能も備えている．

5. 終わりに

本稿では，ストリーム伝送アプリケーションの機能を補填するアプリケーションゲートウェイのフレームワークの一つとしてパケット損失回復機能を持つゲートウェイの開発と評価について述べた．現在の実装では RTP/RTCP を想定した実装で，ストリーム伝送に使われる RTSP や H.323 などに対応していないが，それらのプロトコルへの対応や複数ポートの冗長化や，TCP への対応なども行いたい．また，映像伝送の品質向上を目指し，FEC だけでなく遅延やジッタの制御などを行う機能の検討も行っていきたい．

謝辞

本研究の一部は広島市立大学特定研究費(平成 15 年度 3207)の支援を受けて実施されている．ここに記して感謝の意を示す．

文　献

[1] “Netmeeting,”http://www.microsoft.com/

[2] “PolycomViewstation,”http://www.polycom.com/

[3] “VIC,RAT,”http://www-mice.cs.ucl.ac.uk/multimedia/software/.

[4] 岸田崇志，前田香織，河野英太郎，“多様なコラボレーションを実現する音声伝送システム，” 情報処理学会論文誌,Vol.45,No.2, pp.517-525 (2004).

[5] 近堂徹，西村浩二，相原玲二，前田香織，大塚玉記，“高品質動画像伝送における FEC の性能評価，” 情報処理学会論文誌,Vol.45, No.1, pp.84-92, (2004).

[6] 新井雅之，黒須一司，福本聡，岩崎一彦，“畳込み符号とエックスキャストを用いた高信頼化 TV 会議システムの実装，” 電子情報通信学会研究報告，IN-2002-241, pp.49-54 (2002).

[7] 加島伸悟，後藤幸功，荒木啓二郎，“Reed-Solomon 符号を用いた通信のインターネット音声放送への応用と評価，” DICOMO 2002 シンポジウム論文集，pp.503-506, (2002).

[8] 大塚玉記，西村浩二，相原玲二，前田香織，“FEC を用いた MPEG2 over IP システムの開発と評価，” 情報処理学会研究報告，2001-DSM-24-8，pp.43-48(2001).

[9] “VideoLAN，” http://videolan.org/.

[10] “DVTS，” http://dvts.jp/.

[11] “Robst，” http://net.ipc.hiroshima-u.ac.jp/mpeg2ts/.