

エンコーダ、デコーダ制御によるインターネット経由の映像音声 ストリームの高信頼、低遅延伝送制御の実現

村本 衛一[†] 米田 孝弘[†] 小西 一暢[†] 蓑田 佑紀[†] 知念 賢一[‡]

[†] パナソニック株式会社 本社R & D部門 〒571-8501 大阪府門真市大字門真 1006

[‡] 北陸先端科学技術大学院大学 情報科学研究科 〒923-1292 石川県能美市旭台 1-1

E-mail: [†] {muramoto.eiichi, yoneda.takahiro, konishi.kazunobu, minoda.yuki}@jp.panasonic.com,

[‡] k-chinen@jaist.ac.jp

あらまし インターネット経由で映像音声のストリームを用いたインタラクティブな会話を実現するには、変動する利用可能な帯域や伝送遅延に適応し、送信端末で冗長符合の量やエンコーダを制御することで符号量や調整することでパケットの流量制御し、パケット損失を防ぐ必要がある。また、発生するパケット損失を FEC, ARQ 等により回復させる必要がある。インターネットではトラフィック競合により、伝送遅延時間やパケットの損失度合いが変わるので、状況がよいときは低遅延伝送を実現しつつ、状況が悪化した場合に、ARQ による回復機会を確保するには、デコーダの再生時刻を制御する必要がある。これらの要素技術の研究はこれまで行われてきているが、エンコーダ、デコーダ制御を含めた映像音声ストリームの高信頼、低遅延伝送システム的设计実装の報告はない。我々の実装と他社機の実装をインターネットでの利用を模倣した環境で主観評価を実施し、他社機が MOS (Mean Opinion Score) 値 2 以下となる条件で我々の実装が MOS 値 3 を満たすことを確認した。

キーワード 適応流量制御, 適応 FEC, 適応遅延制御, トラフィックエミュレーション

Reliable and Low Delay Transport via the Internet Using Encoder and Decoder Control for Audio Video Streaming

Eiichi MURAMOTO[†] Takahiro YONEDA[†] Kazunobu KONISHI[†] Yuki MINODA[†]

and Ken-ichi CHINEN[‡]

[†] R&D, Panasonic Corporation 1006 Kadoma, Kadoma City, Osaka 571-8501, Japan

[‡] School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan

E-mail: [†] {muramoto.eiichi, yoneda.takahiro, konishi.kazunobu, minoda.yuki}@jp.panasonic.com, [‡] k-chinen@jaist.ac.jp

Abstract For the real-time audio and video streaming over the Internet, the adaptive flow control at the sender controlling encoding rate according to the available bandwidth is necessary to prevent the packet loss. Additionally, the packet loss experienced in the Internet should be recovered using ARQ, FEC. The adaptive play-out time control at the decoding receiver is necessary to achieve low delay delivery and to keep the opportunity for ARQ to recover lost packets because transmission delay and loss rate vary dynamically on the network. There are plenty of prior works for each individual technology but with our best knowledge there is no design and implementation academic literature on reliable and low delay transport including encoder and decoder control. We have evaluate our implementation in the test-bed where the realistic internet traffic has been emulated and confirm that in the subjective evaluation our implementation marks MOS (Mean Opinion Score) 3 where the competitor implementation can mark only 2.

Keyword Adaptive flow control, Adaptive FEC, Adaptive play-out time control, Traffic emulation

1. はじめに

インターネットのようなベストエフォートネットワークは、回線交換網や専用線で構築された網と比較して構築維持コストが低く抑えられるので、広く用い

られている。

ベストエフォートネットワークの利用者や利用頻度が増えると利用者一つの通信に利用可能な帯域が減少し、同じデータの伝送に多くの時間を要する。しかし、映像音声ストリームを遠隔間での会話に用いる場

合、利用可能な帯域が減少した場合でも、インタラクティブな会話を実現するため、低遅延で映像音声を通信相手に伝送する必要がある。また、伝送中にパケット損失が起こると音声途切れたり映像がみだれたりするで利用可能な帯域が変化した場合、できるだけ高精度に追従し、伝送中にパケット損失を発生させないことが望ましい。すなわち、ネットワークで利用可能な帯域に追従してエンコーダの符号量を制御し、ネットワークでパケット落ちを発生させない流量でパケットを送信する必要がある。また、伝送中に発生したパケット損失は、冗長符号化や再送機構で回復してからデコーダに挿入することで音声や映像の途切れや乱れを防止する必要がある(図1)。

ネットワークで発生する伝送遅延時間の変化を吸収したり、再送によりパケット損失を回復したりするには、受信端末で再生時刻を延ばす必要がある。再生時刻を延ばすと受信端末で映像音声再生されるまでの遅延時間が遅れるのでインタラクティブな会話には望ましくない。このため、ネットワークでのパケット伝送状況に合わせて一定の制約の範囲でデコーダの再生時刻を制御する必要がある。

ネットワークの利用可能な帯域を推定したり、パケットが損失した場合に冗長符号化や再送によって回復させたりする要素技術は古くから研究されている。しかし、エンコーダやデコーダの性能面での制約、実装する機器の制約を考慮しながら適応的に送信レートや再生時刻を制御することで高信頼、低遅延な映像音声のストリーム伝送を実現する実装の報告は著者の知る限りない。また、単純なランダムパケットロス環境ではなくインターネット上でのトラフィック競合を模倣した環境で評価した報告は見当たらない。

本稿では、これらの制約条件を満たしながら、利用可能な帯域、パケット損失の発生頻度、伝送遅延時間の変動するインターネットを経由して、高信頼、低遅延な映像音声のストリーム伝送を実現する方式の設計と実装およびインターネットのトラフィック競合を模倣した環境での評価について報告する。2章で、インターネット経由で映像音声ストリームを伝送するための要求事項と既存の要素技術について述べる。3章で映像音声ストリーム伝送制御(以後、AVQoS伝送制御と呼ぶ)の設計要件と制約事項について述べる。4章でAVQoS伝送制御の設計と実装について述べる。5章でインターネットでの利用を想定した評価について記述し、6章でまとめる。

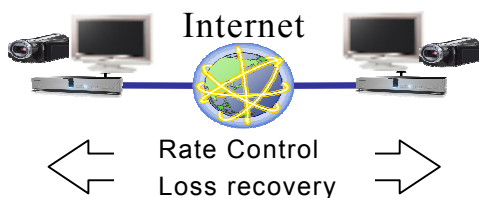


図1：インターネット経由の映像音声ストリーム伝送

2. インターネット経由の映像音声ストリーム伝送への要求と設計方針

AVQoS 伝送制御への要求と関連する要素技術、AVQoS 伝送制御の設計方針を説明する。

2.1. 変動する利用可能帯域への適応

インターネットでは利用者のトラフィックの増減によって、利用可能な帯域が変化する[1]。

AVQoS 伝送制御は、利用可能な帯域に合わせて送信するストリームの流量を変更する必要がある。

利用可能な帯域を推定する技術として、インターネット上の特定の経路のパケット損失の発生度合いと往復時間(RTT: Round Trip Time)から利用可能な帯域を等式で算出するTFRC(TCP Friendly Rate Control)[2]が知られている。また、TFRCを用いた端末間のプロトコルとしてDCCP(Datagram Congestion Control Protocol) Congestion Control ID=3[3]が定義されている。

DCCPでは、端末スタック内で推定した利用可能な帯域をエンコーダ等のアプリケーション層へ通知する仕組みが定義されていない。また、競合トラフィックによるパケット損失が発生する状況を前提としており、競合トラフィックがない回線でパケット損失を発生させずに帯域を推定するには適さない。

遅延の変動を検知することで、利用可能な帯域に高精度に適応しようとするTCPの拡張Vegasが知られている[4]。TCPはスライディングウィンドウ形式の流量制御であり、高頻度に送信量を変更するので低遅延のストリーミングに適さない。また、TCP Renoとの公平性に問題があることが指摘されている[5]。これらのことから、我々はTFRCをベースに遅延変動によりネットワークの空き帯域を推定する方式を設計する。

2.2. 低遅延伝送

インターネットを経由してテレビ会議のようなインタラクティブな会話を実現するために、可能な限り低遅延で伝送を行なう必要がある。

時間制約付きの信頼性伝送を実現するプロトコルとして、SCTP(Stream Control Transmission Protocol)が定義されている[6,15]。SCTPは、スライディングウィンドウ形式の流量制御である。また、SCTPでは、送信端末が時間制約を指定する。我々は、再送用の遅延時間を必要に応じて受信端末が動的に変更する時間制約付き信頼性通信を実現する方針とした。

2.3. 損失したパケットの回復、補完

インターネット上で映像の乱れや音声の途切れを発生させず伝送するには、発生するパケット損失を可能な限り回復させる必要がある。パケット損失を回復する技術としてFEC(Forward Error Correction)やARQ(Automatic Repeat request)がある[7]。

FECを利用する場合、利用可能な帯域から冗長符号を付与する帯域を減算したものをエンコーダが利用可能な帯域に割り当てる必要がある。一方ARQは、再送パケット分のみ帯域を消費するので、帯域の利用効率がよい。しかし、再送パケットが到着するまでデコーダでストリームを再生できないので、利用者がインタラクティブな会話を行なった場合に知覚するEndtoEnd遅延時間が大きくなるという問題がある。

我々は、パケット損失率に基づきFECの冗長符号の量を変更する適応FEC、および一定のEndtoEnd遅延時間以下となる範囲で、デコーダでの再生時刻を遅らせることでARQによる再送機会を確保する適応再生時刻制御を適用している。

一方、発生したパケット損失を補完するエラーコンシールメント技術として、正常に受信できたもののみを継ぎ合わせる方式、無音やホワイトノイズを挿入したり話速を制御したりすることで知覚しにくくする方式、波形やデコーダ中の情報を用いて合成する方式等が知られている[8]。我々は、音声伝送については、十分な冗長符号を用いて回復させることを基本として、さらに再送を併用した。また、音声の補完としては、単純であるため高速に処理ができる無音挿入方式を採用した。映像伝送については、エンコーダへの割り当て帯域を多くするため、必要以上の冗長符号の付与は行なわない方針とし、デコーダでの最終的な損失を検知した場合、デコーダでロスなく受信した映像を元に表示する簡素な方式のみとし、かわりに、即座にエンコードに参照依存関係の無いエンコードを促すフィードバックを返送する方法(リフレッシュ)を採用した。

3. AVQoS 伝送制御の設計要件・制約事項

本章では、AVQoS 伝送制御の設計に対する制約事項を示す。

3.1. パケットロスに依存しない帯域推定

競合する TCP と公平な帯域を算出する TFRC は、パケットロスが発生することを前提としている。TFRC のみで利用可能な帯域を推定すると回線帯域の上限を超えた帯域を推定し、オーバシュートする（回線帯域の上限を送信レートが上回る）ことがある。しかし、インターネットに接続する回線の帯域がボトルネックとなる場合、このボトルネック帯域以内で流量を制御したほうが自ストリームのオーバシュートによる不要なパケットロスの発生を防ぐことができる。

このような要求を満たすにはパケットロスに依存しない帯域推定を導入する必要がある。

3.2. 流量制御と ARQ, FEC との関係・設計要件

● ARQ 利用の条件となる適応流量制御

ARQ を用いた場合、輻輳発生時に再送パケットにより輻輳状況を悪化させる。このため、利用可能な帯域の変動を即座に検知し、特に輻輳時に即座に送信端末から送信するパケットの流量を抑止できることが ARQ を適用する条件となる。

● 適応 FEC と同期した流量制御

ロス状況が悪化した場合に FEC の符号量を増やすと冗長符号の分だけ送信帯域が増え、これが新たなロスを発生させる原因になるので、FEC の符号量制御とエンコーダの符号量制御は同期して行う必要がある。

3.3. FEC, ARQ の帯域利用効率、遅延時間

● 帯域の利用効率

パケット損失回復機構としては、FEC より ARQ の方が帯域の利用効率がよい。これは ARQ が発生したロス分のみ帯域を消費するためである。一方、FEC では、ロスの有無に関わらず冗長符号を送付するため帯域の利用効率が悪くなる。

● 遅延時間

ARQ を使うためには、最低 1 RTT (Round Trip Time) 以上再生時刻を遅らせる必要がある。インターネット

上で輻輳が発生すると、輻輳リンクの手前のルータ上のキューでのパケットの滞在時間が増えるため、結果として RTT も増える。ARQ によりパケット損失を回復させるには、輻輳状況の RTT の増加を考慮して再生時刻を遅らせておく必要がある。一方、FEC では送信端末と受信端末との間の RTT に関わらず FEC ブロックの大きさと伝送レートで受信端末での復元遅延時間が決定されるという特徴がある。

3.4. 流量制御とエンコーダ制約

高精度な適応流量制御を実現するには、インターネットで利用可能な帯域を高精度に推定し、推定した値に忠実にデータ量を制御する必要がある。

しかし、エンコーダでは、画質を保護するため、エンコードレートを変更する頻度と変更量が制限される場合がある。我々が用いた H.264 のエンコーダでは、160ms 毎に±25%の以内のみエンコードレートを変更との規定がある。また、動きが少ない画像では、エンコーダは目標ビットレート以下の符号量のデータしか出力しない。

3.5. 遅延制御とデコーダ制約

遅延制御では、以下の2つの機能を実現する必要がある。

● 伝送遅延時間の揺れの吸収

インターネット上で輻輳が発生した場合、伝送遅延時間が伸びる。これは、受信端末では、伝送遅延時間の揺れ(ジッタ)として観測される。ジッタが発生するとデコーダへのデータ供給が遅れ、データの枯渇が発生するため、データの再生時間に間に合わず、音声の途切れや映像の飛びが発生する。そのため、ジッタを観測し、その分再生時刻を遅らせる必要がある。

● 再送機会の確保

ARQ を効かせるためには、パケット損失の発生が増加すると RTT 分、再生時刻を遅らせ再送機会を増やす必要がある。また、パケット損失の発生が減少すれば、再生時刻を早くすることで低遅延伝送を実現すべきである。

再送機会の確保、ジッタ吸収のため、受信端末では、デコーダの再生時刻を適応的に変更する。再生時刻を急速に変更すると、音声では、音飛びが発生したり、音途切れが発生したりする。映像では、映像飛びが発生したり、映像の一時停止が発生したりする。

インタラクティブな会議用途では、音飛びや音途切れが発生すると利用者が気になるので、デコーダでの再生速度(話速)制御することで、音飛びや音途切れを感じさせない速度で緩やかに再生時刻を変更することが望ましい。音の高低の変化が気にならない範囲で、話速制御による遅延制御を行うには、1秒あたり 1ms~15ms の範囲で再生時刻を変更する必要があることを我々の主観実験で確認した。具体的には、音楽では 2ms 変化/秒以内、会話では、8ms 変化/秒以内の変化であれば、被験者 3 名は変化が気にならなかった。遅延制御は、できる限りこの範囲で制御することが望ましい。

3.6. 実装対象機器の制約

組み込み機器の CPU は、同じ時期に一般に利用可能なパーソナルコンピュータの CPU と比較して非力である。我々が実装した機器の CPU は、AM34 という名称で呼ばれる 350Mhz×2 のデュアルコアである。この上にエンコーダ、デコーダ、HDMI 装置等のデバイスドライバの処理を抽象化し統合的に制御しやすくするため OS を搭載する必要がある。AM34 は、暗号演算や複雑な冗長符号用の演算を行なわせるには非力であった。

3.7. ソフトウェアシェイパの必要性と制約

利用可能な帯域をオーバーしてパケットを送信するとパケット落ちを発生させてしまう。推定した帯域に基づいてエンコーダの符号量を制御するだけでは、エンコーダによるデータ出力タイミングでバースト的にパケット送信が行われ、パケット損失が発生する。パケット落ちを発生させずに送信するには、送信端上でパケットの送信間隔を調整する必要がある。例えば、I ピクチャなどエンコーダから符号量が多いピクチャが出力された場合、平常と比べて 10 倍以上の符号量が発生する (図 2)。

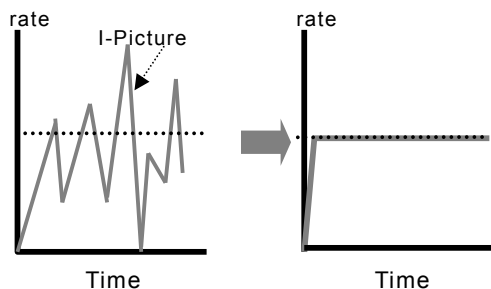


図 2 : シェイパの必要性

1000 バイト長のパケットを 8Mbps で送付するためには、1ms 毎にパケットを送付できることが望ましい。ソフトウェアシェイパで実現するためには、OS タイマーの動作頻度 (HZ) は、1000、即ち 1ms 毎のタイマー駆動が必要となる。しかし、上記の CPU で HZ=1000 で動作させるとその駆動だけで 10%程度 CPU を消費してしまうため、できるだけタイマーの動作頻度 (HZ) を落とした状態で運用する必要があった。

4. AVQoS 伝送制御の設計と実装

本章で、AVQoS 伝送制御の設計と実装について述べる。

4.1. TFRC と遅延変動を用いた帯域推定方式

3.1 節で述べた要求を満たすため、AVQoS 伝送制御では RTT の変動、パケット損失の有無、および TFRC の推定を組み合わせた帯域推定方式を採用した。

RTT の変動は、過去の RTT 計測値を異なる 3 つの集計区間で移動平均を取り、これらの 3 つの平均値の変動から RTT が増加傾向であるのか、減少傾向であるのか判定する。

増加傾向で有る場合、輻輳と判断し、送信レートを下げ、逆に増加傾向で無い場合、送信レートを上げるという動作を基本としたレート制御をビデオの報告パケットの到着毎に送信端末で行う (表 1)。ビデオの報

告パケットは 30ms 毎に受信端末から送信端末されている。

このとき、送信レートを増加・減少させるときの目標値を式(1)で示す TFRC 推定値で決定する。これにより、極力 TCP との公平性を保ちつつ、回線帯域の上限が設定された環境でのオーバシュートによるパケット損失を防ぐ。

表 1 : パケット損失・RTT の変動傾向による帯域推定

パケット損失	RTT の変動	帯域推定方針
なし	増加傾向	レート減少
なし	非増加傾向	レート増加
あり	増加傾向	レート減少
あり	非増加傾向	レート維持/減少

TFRC 推定値算出式

$$X_{cal} = \frac{8s}{R(\sqrt{2p/3+t_RTO} \times \sqrt{3p/8 \times p \times (1+32p^2)})} \dots(1)$$

s はパケットサイズ [バイト], R は、RTT [秒], p は損失イベント率, t_RTO は、再送タイムアウトで 4R であり、Xcal が推定レートとなる。

4.2. 帯域推定と同期した適応 FEC

3.2,3.3 節で述べた通り、FEC の冗長度の変更は、エンコーダへのレート変更と同期して実施する必要がある。推定レートから、音声伝送に必要な帯域と、映像パケットの FEC に必要な帯域および後述するオーバーヘッドを減算した帯域を目標ビットレートとして映像エンコーダに通知する [8]。また、このとき 3.4 節で述べた制限の範囲で目標ビットレートを滑らかに変更する。

4.3. ダミーパケット送信処理

エンコーダは、単純な映像の場合目標のビットレートより低い符号量しか発生しないことがある。ネットワークで TCP フローと競合状態のときに適切なタイミングでパケットを送出していないと競合フローに機会を譲り、結果として自フローの平均レートが上らなくなるため、帯域推定方式で算出した結果よりエンコーダが発生する符号量が少ない場合、ダミーパケットを自動送信する機構を導入した。

また、利用可能な帯域を超えて帯域を推定している場合、ダミーパケットにより確実に RTT が増加するため、4.1 節で述べた帯域推定が設計通りに動作するという効果が得られる。

4.4. FEC, ARQ ハイブリッドによるロス回復機構

パケット損失率に応じて、FEC の冗長符号の量を {0%, 25%, 50%, 100%, 150%, 200%} で変更する適応 FEC を導入した。往復時間が短く ARQ が有効に働く場合、FEC の冗長符号度は半分すなわち {12.5%, 25%, 50%, 75%, 100%} の範囲で冗長符号の量を変更する。急激な冗長符号の量の変更は、エンコーダへの符号量の急激な変更につながるため、パケット損失率は、過去 16 世代のパケット損失間隔の逆数の履歴に対して指数加重移動平均を取ったものを採用した。

FEC ブロックの大きさは、低遅延伝送のため、1 ピ

クチャの大きさに判定する。

インターネットの輻輳度合いが低い場合、パケット損失率がほぼゼロの状態が続くことがある。このときエンコーダに多くの目標ビットレートを与えるため、FECの冗長符合の量をゼロにして運用する。この運用を実現するため、FECの回復よりARQを先に動作させるTYPE2のハイブリッド方式を採用した。

このハイブリッド方式により、FECで回復困難なパーストロス（パケットロス）はARQで回復を試み、再送パケットの損失分をFECで回復させる。

4.5. 再送機会確保のための遅延制御

2.2,2.3,3.3節で説明した通り、ARQでの少ない帯域消費量でロス回復できるが、遅延時間を多く必要とする。我々は自由会話において往復遅延が与える会話への影響（主観値）を測定した。その結果、カメラやディスプレイの表示遅延を含むEndtoEnd遅延時間が往復1.2秒を超えると悪化することから、片方向600ms以内で遅延時間制御を実施する（図3）。

なお、再送用に確保する遅延時間は状況に合わせて適応的に制御するため、損失の少ないネットワークでは低遅延になるように制御を行う。

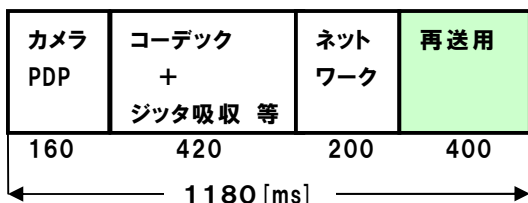


図3：往復1.2秒以内での再送機会確保

4.6. 遅延制御時の音切れを防止機構

3.5節で説明した通り、急速な遅延変更は、音切れに繋がる。このため、再送機会確保、ジッタ吸収用バッファ確保のための遅延延長および短縮は、1秒間に8msの範囲で変更することで、話速制御による音切れ防止を働かせながら変更する。

一方、ネットワーク上での経路変更に伴う遅延延長（再生時刻延長）は、パケットの到着タイミングが急速に変化するため、8msの範囲を超えて直ちに適応させる。

4.7. 実装対象機器の制約と対策

3.6章で述べた通り、我々が実装した機器のCPUは、暗号演算や複雑な冗長符合用の演算を行なわせるには非力であった。このため、SRTPのペイロード部分の暗号化には、AM34とは別のAM32と呼ばれるサブCPUを用いることにした。CPU間やエンコーダ・デコーダとの間のデータ伝送はDMAを利用した（図4）。

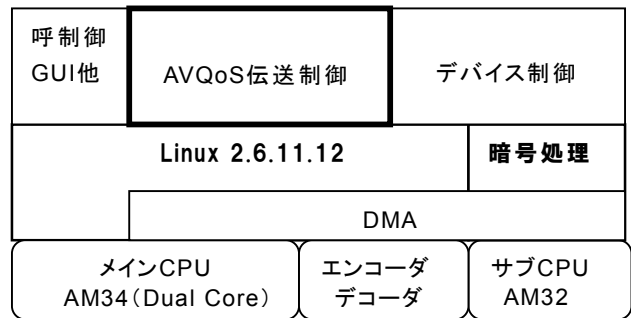


図4：サブCPUを用いた処理負荷分散

また、FECの冗長符合演算は、AM34で行なう必要があったため、冗長符合生成方式としては、演算量が少ないビット演算XORを用いた冗長符合化を用いた。

5. インターネット模倣環境を用いたAVQoS伝送制御の性能評価

本章ではインターネットのトラフィック競合を模したエミュレーション環境でのAV-QoS伝送制御の性能評価について述べる。インターネットエミュレーションとはインターネット上で発生する現象を実験室で模倣する装置であり、実インターネット網を使うよりも再現性や網羅性の観点で効率的な評価が可能となる。

本稿では市販の簡易装置(NXS7000F)とStarBED[12]を使った評価を実施した。

5.1. 簡易環境での評価

帯域制限やロスや遅延等のネットワーク条件を設定できるソフトウェア実装として、dumynet[9], netem[10], NISTnet[11]が知られている。2つ以上のNICを持った1台のPCにこれらのいずれかの実装を導入し、その両端に送信端末、受信端末を配置する試験環境は簡易に構築できる（図5）。また、市販装置を用いても同様のネットワーク条件を設定できる。



図5：主観評価の構成（簡易環境）

この簡易環境を用いて、我々の実装と競合他社の実装を評価した。

5.1.1. 評価条件、評価指標

表2にネットワーク条件を示す。

表2：簡易評価のネットワーク条件

条件	内容
帯域制限	なし
パケット損失方式	ランダムロス
パケット損失率	3%(双方向に設定)
パーストロス数	3(双方向に設定)
片方向伝送遅延	50ms(双方向に設定)
装置名（市販装置）	富士通九州ネットワークテクノロジー社製 NXS7000F

パケット損失率（5秒間統計値）は、我々が複数のフィールド試験（東京－大阪、大阪－上海、大阪－北米、北陸先端大－NSPIX－NICT北陸リサーチセンタ）において経験している値（0%～10%程度）から比較的通信経路上が混雑していると考えられる値として3%を選択した。パーストロス数も同様にフィールド試験で経験している値（0～7）の中間的な値を選択した。

評価指標は、受信端末で再生される映像、音声のMOS(Mean Opinion Score)値とした。表3に評価条件を示す。

表3：簡易評価の評価指標

条件	内容
入力映像	会議映像（約1分30秒）
映像符合化方式	H.264 HD（1280x720）
入力音声	会議を想定した自由会話
MOS 値の評価基準	5：映像と音声の品質劣化に気づかない 4：品質劣化に気付くが気にならない 3：品質劣化が気になるが、通信として十分使える 2：通信として使いにくい 1：通信として使えない
評価者	3名（MOS 値は0.5単位で評価）

5.1.2. 評価結果

表4に評価結果を示す。

表4：簡易評価の結果

	映像 MOS 値	音声 MOS 値
我々の実装	4	4.5
他社機 A	1.5	4.5
他社機 B	3	4.5

図6,7,8は、それぞれ我々の実装、他社機A、他社機Bのスクリーンショットの例である。我々の実装では観測されない画像の乱れが他社機Bでは観測されることがわかる。なお、他社機Aでは、乱れた画像は再生されないが、映像が停止する場面が多発したため、MOS 値が低くなっていると考えられる。これに対し我々の実装では、パケット損失により、目標ビットレートを低下させ、さらにFECの冗長符合の度合いを高めるため、エンコーダに設定する目標ビットレートが低下し、画質が低下するが、ARQ,FECによりほぼ回復できている状態が続いた。



図6：我々の実装の画面例（ロス率3%）

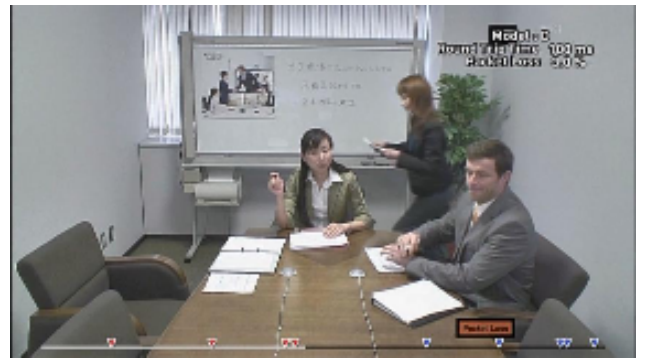


図7：他社機Aの画面例（ロス率3%）

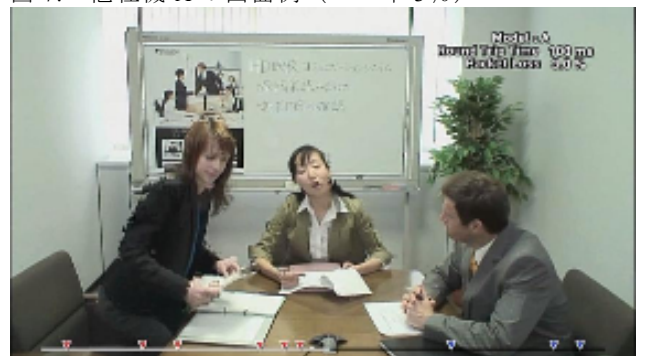


図8：他社機Bの画面例（ロス率3%）

5.2. インターネットエミュレーション環境での評価

インターネット上でのトラフィック競合状態での性能を評価するため、StarBED[12]上にアクセス網でのトラフィック競合状態を模倣した環境を構築した。トポロジは図9で示すダンベルトポロジーを採用した。トラフィック競合時のパーストロス発生頻度は、長時間TCP（例：FTP）と短時間TCP（例：HTTP）の混在度合いによって異なることが知られている[13]。そこで、実際にFTPとHTTPを用いて長短TCPが混在する競合トラフィックを発生させた。このような環境で我々と他社の実装を評価した。我々の実装ではタイマ一粒度{1ms,4ms}を変更した二つの版を用いた。

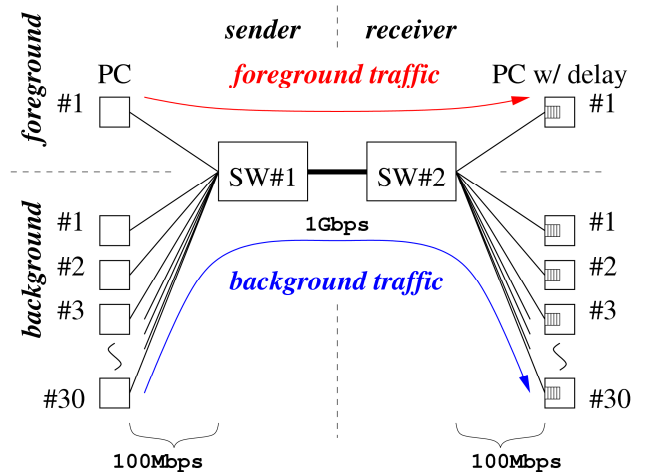


図9：StarBED上に構築したダンベルトポロジー

5.2.1. 評価条件、評価指標

トラフィック規模決定方法：

小規模な地域のアクセス網にて輻輳が発生したと想定して、実験のパラメータ設定を行った。最繁忙時発呼率が60%とし、5万加入者の最大同時接続数の上限は30,000同時接続となる。最繁忙時トラヒックの1/10を再現することを目標とし、TCP競合の本数を3,000本の同時接続数とした。

HTTPのトラフィック構成方法：

短時間接続トラフィックであるHTTPは、現実性を考慮するとファイルサイズ分布を考慮する必要がある。今回、このHTTPファイルサイズ分布には文献[14]より対数正規分布を使用し、平均、分散値は $\mu = 9.357$, $\sigma = 1.318$ を用いた。

FTPのトラフィック構成方法：

長時間接続トラフィックを生成するため十分に大きなファイル(1GB)をFTPで繰り返しダウンロードするTCP接続を用いた。

5.2.2. 評価条件

現実に発生する様々なロス環境での比較実験のため、表5に示す評価指標で表6にする評価条件で我々の実装と他社機A2(表7)の比較試験を行なった。

表5：小規模アクセス網実験の評価指標

条件	内容
入力映像	実験室をカメラで撮影した映像
映像符合化方式	H.264 HD (1280x720)
入力音声	なし(評価対象としない)
MOS値の評価基準	5：映像の品質劣化に気づかない 4：品質劣化に気付くが気にならない 3：品質劣化が気になるが、通信として十分使える 2：通信として使いにくい 1：通信として使えない
評価者	3名

表6：小規模アクセス網実験の評価領域

評価パラメータ	評価レンジ
ボトルネックリンクの帯域	1Gbps
長短TCP混在率	HTTPとFTPの混在率 = 0, 0.25, 0.5, 0.75, 1 (ex. 混在率0のとき, HTTP 0%, FTP 100%)
総TCP本数	120本, 480本, 1920本, 3000本
RTT	10ms(東京-大阪の理論値)

表7：比較した機種との機能比較

	我々の実装	他社機A2
レート制御	TFRCベース	TFRCベース
ロス回復機能	FEC, ARQ	FEC, ARQ
ロス復帰機能	リフレッシュ	リフレッシュ

5.2.3. 評価結果

1920本, 3000本競合時は、我々の実装、他社機A2とも映像MOS値=1となり比較にならなかった。また、120本競合時は、我々の実装、他社機A2とも映像MOS値=4~5となり、大差は認められなかった。480本競合時の映像MOS値を表8に示す。

表8：TCP480本競合時の映像MOS値

	映像MOS値
我々の実装(1msのタイム粒度)	3
我々の実装(4msのタイム粒度)	3
他社機A2	1.5

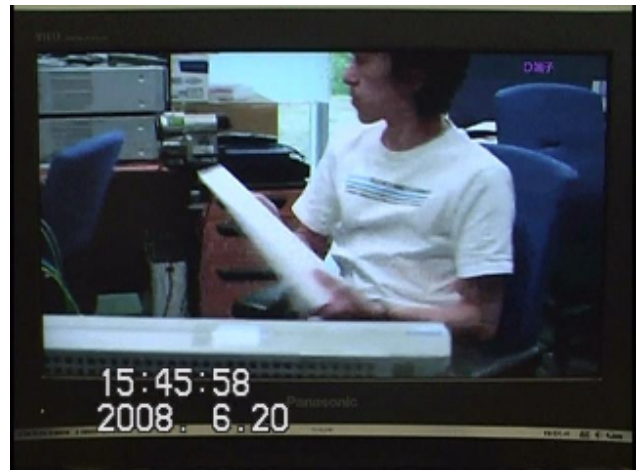


図10：我々の実装(1msのタイム粒度)の画面例



図11：我々の実装(4msのタイム粒度)の画面例



図12：他社機A2の画面例

我々の実装 1ms,4ms では差が認められなかった。

他社機 A2 は、絵が止まる時間が多いため低い値となっている。

6. まとめ

インターネット経由でインタラクティブな映像音声のストリームを用いた会話を実現するには、変動する利用可能な帯域や伝送遅延に適応しながら、エンコーダ、デコーダを高精度に制御する必要がある。我々は、1) TFRC と伝送遅延時間の変動により利用可能な帯域を推定しエンコーダを制御する機能、2) エンコーダが生成する符号量の偏りをダミーパケットにより補正しながら、利用可能な帯域の利用機会損失を防ぐ機能、3) インターネット上で発生するパケットロスを FEC と ARQ のハイブリッド方式で回復させる機能、4) パケット損失率に対応した再送機会の確保および伝送遅延の揺れの吸収のため、デコーダでの再生時刻を変更する機能を設計実装した。また、本実装と現存する他社機の実装の総合性能をインターネットでの利用を模倣した環境で評価した。この結果、我々の実装が MOS 値 3 を維持できる条件で他社機は MOS 値 2 以下であることを確認した。

これまで要素技術は過去に研究されてきたものであるが、エンコーダ、デコーダの制御を含めたシステム全体の実装を、インターネットを模倣した環境で繰り返し評価、調整した報告は無い。

今後は、フィールド実験で得られた遅延やパケット損失の発生度合いの変動を模倣できる環境を構築し、無線を含めた様々なネットワークの挙動に適応する流量制御、再生時刻制御を設計・実装・評価していく予定である。

謝辞

北陸先端科学技術大学院大学 篠田陽一教授には、インターネット模倣環境含め検証手法について幅広く指導いただいた。また、NICT 北陸リサーチセンタおよび篠田研究室各位には、StarBED での検証実験および NXISP 折り返しフィールド実験に協力をいただいた。水野治展氏はじめパナソニック株式会社 ビジュアルコム開発室のメンバには、フィールド実験の結果に反映した検証環境の実現等の方向性の提示および実施、エンコーダ、デコーダとの密連携機能の設計実装調整、パナソニックアドバンステクノロジー株式会社の関連メンバには、本方式の詳細設計実装に協力をいただいた。パナソニックシステムソリューションズ株式会社の関係各位には本実装の商品 (KX-VC500) への搭載にあたり品質見極め等に多大な協力をいただいた。各位に感謝申し上げたい。

文 献

- [1] V.Paxson, "End-to-end Internet packet dynamics", ACM SIGCOMM Computer Communication Review, Volume 27, Issue 4, October 1997
- [2] S. Floyd, et al., "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC5348, September 2008.
- [3] S. Floyd, et al., "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC4342, March 2006.
- [4] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In Proceedings of the SIGCOMM '94 Symposium (Aug. 1994) pages 24-35.
- [5] 倉田謙二, 長谷川剛, 村田正幸, "TCP Reno と TCP Vegas の混在環境における公平性の評価", 『電子情報通信学会技術報告』, Vol.99, No.428 pp.67-72, 1999.
- [6] R. Stewart, et al., "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC3758, May 2004
- [7] E. Berlekamp, et al., "The Application of Error Control to Communications", IEEE Communlcatlonr Magazine, Aprll 1987-Vol. 25, No. 4
- [8] 伊藤 智祥, 山口 孝雄, 吉羽 治峰, 大利 直行, 中村 学, 濱田 準一, メディア伝送方法及びその送信装置, 特開 2004-72720
- [9] Luigi Rizzo, Dummynet: a simple approach to the evaluation of network protocols, ACM SIGCOMM Computer Communication Review Volume 27, Issue 1, PP.31-41 JAN 1997
- [10] Net:Netem, <http://www.linuxfoundation.org/en/Net:Netem>
- [11] Mark Carson, Darrin Santay, NIST Net — A Linux-based Network Emulation Tool, ACM SIGCOMM Computer Communications Review Volume 33, Number 3 PP.111-126 July 2003
- [12] Toshiyuki Miyachi, Ken-ichi Chinen and Yoichi Shinoda, StarBED and SpringOS large-scale general purpose network testbed and supporting software, International Conference on Performance Evaluation Methodologies and Tools (Valuetools) 2006, ACM Press, ISBN 1-59593-504-5, Pisa, Italy, Oct.2006
- [13] Y. Joo, et al., "TCP/IP Traffic Dynamics and Network Performance: A Lesson in Workload Modeling, Flow Control, and Trace-driven Simulations.", CCR, April 2001
- [14] Paul Barford and Mark Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In ACM SIGMETRICS 98, June 1998.
- [15] R. Stewart, "Stream Control Transmission Protocol (SCTP)", RFC4960, September 2007