マルチメディア通信と分散処理 71−19
（１９９５．７．１３）

汎用ネットワークとオペレーティングシステムによる動画転送の限界

山口 信明[1] 杉浦 一徳[1] 徳田 英幸[1,2]

[1]慶應義塾大学 政策・メディア研究科
[2]慶應義塾大学環境情報学部

概要

現在主流を占める汎用ネットワークとオペレーティングシステムを利用した実時間による動画と音声の転送は限界がある．しかしその全てを改良し，まったく新しい環境を構築するのは困難である．現状のシステムを活かした動画転送の評価を行うことで，ボトルネックを見つけだし，そこを改良し動画転送の可能性を検証する．本論文では end-to-end のデータ転送を最大のボトルネックと判断し，転送前にデータを圧縮する手法を提案し，比較を行なう．

Limitations of Transfering Digital Video Files via Generic Networks and Operating Systems

Noburaki Yamaguchi[1] Kazunori Sugiura[2] Hideyuki Tokuda[3]

[1]noburin@sfc.wide.ad.jp, Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252 Japan,
[2]uhyo@sfc.wide.ad.jp, Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252 Japan,
[3]hxt@sfc.wide.ad.jp, Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252 Japan,

Abstract.

There is a limitation of transfering real-time digital video and audio data using generic network and operating systems mainly used today. It is somehow, very difficult to construct a totally new system environment from scratch. We evaluate the performance of video data transfer using generic networks and operating systems and address the performance bottleneck of the system. In this paper, we identify the end-to-end data transfer as the performance bottleneck of the system. Then, we propose digital vido data transfering schemes with data compression and evalute these schemes.

1 はじめに

近年，複数のコンピュータを接続したコンピュータネットワークの普及により，双方向の情報交換が多くの人々の間で可能となった．だが，現状ではネットワーク上に流れている情報の多くは文字情報や静止画像が主流を占め，音声，動画といった情報はあまり流れていない．

リアルタイムで映像と音声を転送するNv[6]やvat[7]のようなアプリケーションも存在するが，テレビ放送に匹敵する画像を手軽に視聴することはできない．

汎用のワークステーションを使用し，ネットワークを通じて動画と音声を転送するには以下のような問題点がある．

1. ネットワークの容量
2. 端末の処理速度
3. プロセスのスケジューリング管理
4. ディスクの速度

これらの4つ問題点を解決するには使用しているオペレーティングシステムそのもの，もしくは問題となる一部分を再設計する必要がある[1]．本論文では，現状のオペレーティングシステムを改良せずに，どの程度まで映像の転送が保障できるのかその限界点を検証する．

汎用の環境での映像転送を行い，データを採集し，現状のシステムでの最大のボトルネックを調査する．オペレーティングシステムを改良することなくアプリケーション側で問題点となる部分の改良を行い，現状のシステムでの映像転送の可能性を検証する．

本論文では始めに既存の環境において動画・音声などの転送を行う場合の問題点を述べ，本研究でのモデルと実験環境について述べる．これらの実験環境を基にした評価と考察を行い，汎用ネットワークとオペレーティングシステムによる動画転送の限界についてまとめる．

2 既存環境

本章では既存環境で映像などの実時間性を持つデータを転送し，再生する際の問題点について述べる．

2.1 ネットワークの容量

ワークステーション上で，テレビと同等な画質を保ちながら動画を表示する場合，必要となるデータ転送量は現行のネットワーク回線を利用する限り，その許容を越えている．

仮に現行のテレビの1フレームを解像度横320ピクセル，縦240ピクセル，1ピクセルにつき24bit(3byte:16777216色数)でキャプチャし，その画像を毎秒30フレームでネットワークを経由し表示すると仮定すると，そのデータ量は約53Mbpsとなる．($320\times240\times3\times30fps = 6,912,000Byte/sec. = 6,750KByte/sec. \approx 52.73Mbps$)

一般的なネットワークで使用されている情報転送手段であるイーサネットの転送容量は10Mbpsであり，これだけの大容量のデータを汎用のネットワーク上でそのまま実時間で転送することは不可能である．

2.2 端末の処理速度

汎用に使われている端末には，端末によって処理能力や搭載されているメモリの量などのマシン資源にかなりの差がある．プロセッサの処理能力が不足していると，前節にかかれたネットワークの問題を解消できたとしても，マシン資源の不足でそのデータを表示する事が追い付かない場合，転送されたデータを効率的に処理しているとはいえない．

しかし能力の高い端末を標準にし，低い端末を見捨てるわけにはいかない．もし能力が高い端末を標準にしてシステムを構築してしまうと，そのシステムは能力が低い端末から見て特化したシステムになってしまう．能力が低い端末でもその端末の能力の範囲内で最善をつくし，対応を可能にする必要性がある．すなわち，システム内を一定のQOS(Quality of Serice)レベルに設定して，データ転送を行うのではなくネットワークや各端末の能力見合ったQOSレベルに適応させて処理する必要がある[3][5]

2.3 プロセスのスケジューリング管理

BSD等のUNIXはマルチタスク・タイムシェアリングシステムとして設計された．タイムシェアリングシステムでは映像の再生，音声の転送といった連続メディアに対する定められた時間どおりに動くことを要求されるタスクも，実時間で動作することを要求されないタスクもRound Robinによって公平に実行される．そのためにタスクが数多くある場合には単位時間あたりの映像の再生を行なう割合が

低下し，実時間での作動に悪影響が出る．またメモリの限界を越えるとシステムによっては仮想記憶領域に対してスワップが発生し，実行処理能力が著しく低下する．

仮にプライオリティをあげたとしても，実時間処理を行うプロセスの数が多くなれば処理の限界を越え意味が無くなる．また，プロセスの数が少なくプロセス単体では十分な処理速度を持っていたとしても，そのプロセスが他のプロセスと交信を行う場合，例えばファイルをアクセスした際に，ファイルマネージャのキューにプライオリティの低いプロセスが詰まっていた場合にはそこで，プライオリティインバージョン [4] が起き待たされることになる．

2.4 ハードディスクの速度

UNIX のようなオペレーティングシステムではメモリを効率良く確保するためにスワッピング，ページングが行われ，スワップには二次記憶としてハードディスクが用いられる．この二次記憶に使用されているハードディスクは一次記憶よりもはるかにデータの読み書きの速度が遅く，かつ不定であり，二次記憶を使用することはオーバーヘッドとなる．

スワップを避けるため動画を再生する実時間のプロセスを実メモリに固定した場合でも，他のプロセスが仮想領域にいるならば，それらのためのディスク操作をするプロセスが高いプライオリティで動き始め動画再生のプロセスが実時間で動作できなくなる [9]．

2.5 問題点のまとめ

これらの問題が実時間での動画再生を困難にしている．実時間で動作するコンピュータシステムは，ソフトウェアとは独立に発展し，制御機器として進歩をしている．ワークステーションのようなソフトウェアによって進歩してきたシステムに，それとは別に進歩してきた制御機器を融合させることは困難である．完全な実時間の保障をせず，妥協できる程度の平均速度と速度の分散により動画の転送・再生を行なうべきであろう．

3 実験モデル

3.1 実験システム構成

実験システムではキャンパスネットワーク内での運用を前提にしている為，ネットワークの最小構成として図1に示すような FDDI と Ethernet を利用したネットワーク構成とした．

開発・実験はクライアント用に SUN SPARC station 2 を，サーバ用に SPARC station 20 を用い，ネットワークは，10Mbps のイーサネットで各ワークステーションを接続し，100Mbps の FDDI を使用してそれらのバックボーン部分を構築した．バックボーンとイーサの接続には，proteon p4200 と Cisco 4000 を使用した．

また実験に使用した動画データは QuickTime(Raw format) 形式で，160x120 ピクセル，8bit カラーである [8]．

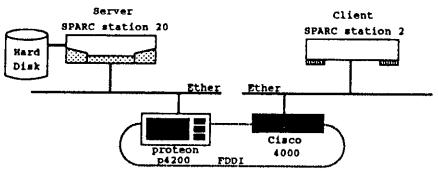

図 1: 実験環境

3.2 実験内容

まず図1の環境を用いて動画の転送・再生を行った．これを標準方式として，次にサーバに対してデータ転送時にデータを圧縮する機能を取り付け，無圧縮の標準方式サーバとの比較を行った．圧縮方式は以下に示す2種類を開発・実装した．

- 可逆圧縮方式

  データを転送時に lzw 法 [2] によって圧縮し，転送するように改良したものである．圧縮後のデータサイズは一定ではないため，クライアントにデータのサイズを通知する必要がある．

- 非可逆圧縮方式

  転送時に表自画像のピクセル数を縦横とも半分にし，データサイズを$\frac{1}{4}$にする．画質は著しく低下するが，圧縮後のデータ量が一定のため，データサイズをクライアントに毎回通知する必要が無い．

これらの実験は，いずれも図1の環境で 10,000 フレーム分の動画の転送を行い，ディスクからのデータの読み込み，データ転送，画面表示の各行程の所用時間を計測した．データの転送速度の計測では画面表示過程を省略したシステムを製作し，データの転送のみを行ない計測した．

4 評価

この章では前章で行なった実験の結果を分析し，ボトルネックの判定と，データの圧縮による効果の評価を行なう．

4.1 標準方式

図 1の環境で，10,000 フレーム分の動画の転送を行いその際のコストを計測した．“ディスクからのデータ読み込み”，“end-to-end でのデータ転送”，“それ以外のサーバに対するタスク”，“データの画面表示”，“それ以外のクライアントタスク” の 5 つの行程に分け，各行程のコストを計測し，グラフを作成した．グラフを図 2に示す．サーバからクライアントへのデータ転送には UDP プロトコルを使用した．

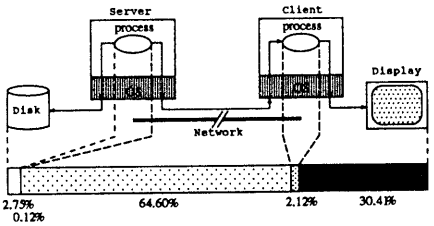

図 2: 各行程の割合

この測定結果によると，“end-to-end のデータ転送” と “XPutImage による画像表示” 部分に多くの処理時間を要し，結果としてボトルネックとなっているのが分かる．この部分の改良を行う事で，より効率良く動画転送の性能をあげる事ができる．本研究では，“end-to-end のデータ転送” の時間の短縮に焦点を絞り，転送時にデータを圧縮することで転送にかかる時間を短縮する実験を行った．

4.2 可逆圧縮方式

図 3は標準方式と可逆圧縮方式の各行程の所用時間を比較したグラフである．可逆圧縮方式は標準方式に比べると転送にかかる時間は減少したが，圧縮と解凍に処理を要するため，性能は標準方式とほぼ同じである．可逆圧縮のため空間的解像度は基本的に標準方式と同一である．ただし，転送時にデータの欠損が起きた場合は正常に解凍されない．

4.3 非可逆圧縮方式

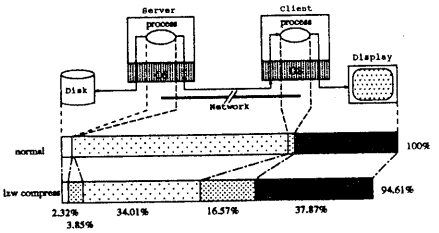

図 3: 可逆圧縮時の割合

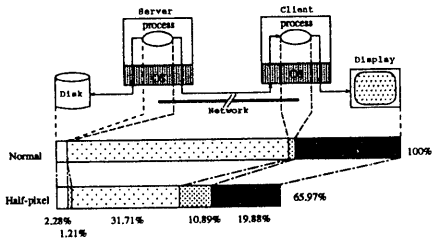

図 4: 非可逆圧縮時の割合

図 4は標準方式と非可逆圧縮方式の各行程の所用時間を比較したグラフである．非可逆圧縮方式は標準方式と比べ転送にかかる時間が半分程になったため，標準方式の 66%程の時間で全行程を行う事ができる．そのため fps を標準方式よりも高くできる．ただし前述したように空間的解像度は落ちている．

4.4 各行程ごとの比較

この節では，標準方式，可逆圧縮方式，非可逆圧縮方式の 3 方式を，ディスクからのデータ読み込みにかかる時間，データの転送にかかる時間，ディスプレイへの表示にかかる時間，フレームごとの fps の 4 項目で比較を行う．グラフは上から標準方式，非可逆圧縮方式，圧縮方式の順である．

方式	標準	非可逆圧縮	可逆圧縮
時間 (msec)	1.86	1.72	1.57

表 1: ディスクからの読み込みにかかる時間の平均

図 5は各方式のディスクからのデータ読み込みにかかる時間のグラフである．どの方式でも同じように読み込んでいるためにほとんど変化はみられない．表 1にその平均時間を示す．

図 6は各方式の 1 フレーム分のデータ転送に要する時間のグラフである．転送時間が短い順に，非可逆圧縮，圧縮，標準であり，そのまま転送されるデータサイズに比例する．表 2にその平均時間を

示す．

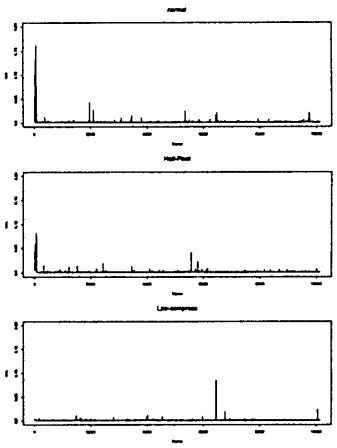

図 5: ディスクからのデータ読み込みにかかる時間

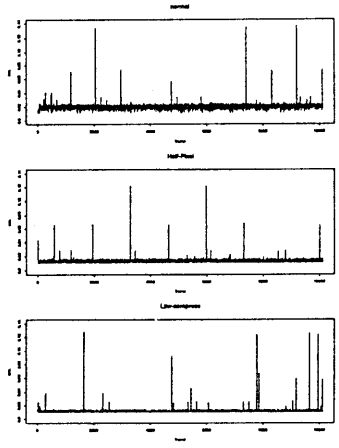

図 7: ディスプレイへの表示にかかる時間

方式	標準	非可逆圧縮	可逆圧縮
時間 (msec)	20.61	13.49	11.22

表 3: ディスプレイへの表示にかかる時間の平均

図 7は各方式の 1 フレーム分のデータを XPutImage を使って表示するのに消費する時間のグラフである．どの方式も同じように表示しているが，標準方式は他の二つよりもわずかに余計に時間を消費している．表 3 にその平均時間を示す．

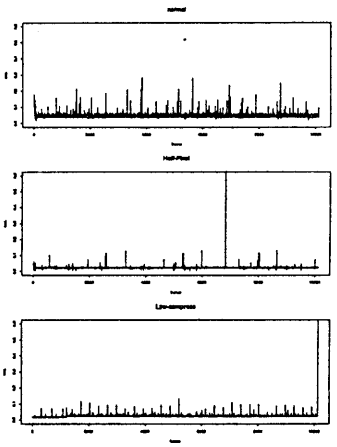

図 6: データの転送にかかる時間

方式	標準	非可逆圧縮	可逆圧縮
時間 (msec)	43.77	21.48	23.05

表 2: データ転送にかかる時間の平均

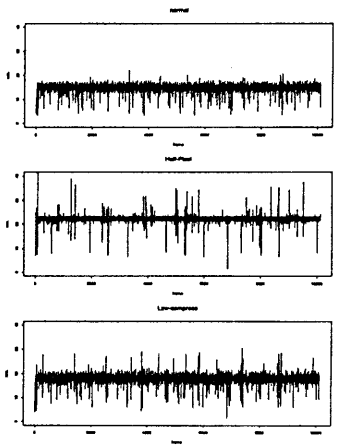

図 8: 各方式の fps

方式	標準	非可逆圧縮	可逆圧縮
平均 (msec)	15.04	22.52	18.01
分散	2.26	2.62	2.92

表 4: 各方式の fps の平均と分散

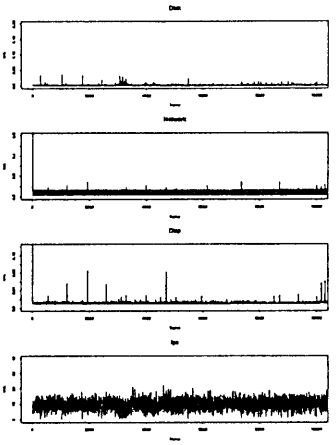

図 9: 3 つのクライアントを接続した場合

図 8は各方式で動画を転送した際のフレームごとの fps の変化をグラフ化したものである．表 4にその平均時間と分散を示す．平均時間で見れば圧縮を行なう方式の方が優れているが，より分散している．

このほか，非可逆圧縮方式を用いて 1 つのサーバに対し，3 つのクライアントを接続させる実験も行なった．その際のグラフを図 9に示す．上からディスクの読み込み時間，データの転送にかかる時間，ディスプレイへの表示にかかる時間，fps である．平均では 10fps 程度で再生を行なうことができた．

5 考察

可逆圧縮方式はデータの転送にかかる時間そのものは短いものの，データの圧縮・解凍に処理を要するため結果として性能は標準方式と変わらない．特にクライアント側のマシンの負荷が非常に高くなっている．より高速で負荷の低い圧縮法で実験しなおす必要がある．

非可逆圧縮方式は，データの圧縮，解凍に消費する時間が短く，またデータ量もより縮小できるために，データ圧縮によってデータの転送時間を短縮できた．クライアント側の負担も可逆圧縮を行った場合程には高くならない．画質の低下が妥協できる程度ならば，十分有益な方式である．どちらの方式でも fps の平均は約 20fps であるが，グラフにはかなりのばらつきが現れている．このばらつきを軽減させることが今後の課題となるだろう．

6 まとめ

平均速度で言えば非可逆圧縮方式では 22fps 程度の再生が可能となる．しかし可逆圧縮方式では圧縮・解凍に大きな負荷がかかり，端末の性能によっては期待した効果は得られない．また非可逆圧縮では比較的負荷は軽いものの解像度は低下してしまう．使用する環境や利用者の要望に合わせ，自動で選択を行なう機構を現在研究中である．

謝辞

常に相談に応じてくれた共同研究者の方たちに感謝する．また，さまざまな議論・意見を提供して下さった慶應義塾大学徳田・村井研究室の諸氏と WIDE プロジェクトの研究者の方々に感謝する．

参考文献

[1] 山口 信明: “非特化ネットワークとオペレーティングシステムによる動画転送の限界”, 徳田・村井研究会卒業論文, 1994 年度.

[2] Nelson Mark: “LZW Data Compression”, Dr. Dobb's Journal, Vol.14, No.10, Oct. 1989, pp 29-37.

[3] H. Tokuda, Y. Tobe, S. T.-C. Chou, and J. M. F. Moura, “Continuous Media Communication with Dynamic QOS Control Using ARTS with an FDDI Network”, Proc. of ACM SIGCOMM'92 Symp., Aug. 1992.

[4] L. Sha, R. Rajkumar, and J. Lehoczky, “Priority Inheritance Protocols: An Approach to Real-Time Synchronization”, IEEE Trans. on Computers, Vol. 39, No. 9, Sept. 1990.

[5] Y. Onoe, K. Fujii and H. Tokuda: “QOS-based Multicast Communication”, Proc. of The 2nd International Workshop of Advanced Teleservices and Hig h-Speed Communication Architectures, 1994.

[6] Nv: NetVideo source code, Ron Frederick, 1994.

[7] vat: Van Jacobson, Steve McCanne, 1994.

[8] 多田征司, “実時間オペレーティングシステムにおける連続メディアファイルサーバーの実験”, マルチメディア通信と分散処理学会, May 1994.

[9] 柴田巧一, 浅井光男, 佐藤未来子, 井川勝, Yunnghee Kim, 滝安美弘 “ビデオオンデマンドサーバの基本モデルの開発”, オーディオビジュアル複合情報処理, Oct. 1994.