

第 XXVI 部

超広帯域ネットワークにおける キラーアプリケーション

第 26 部

超広帯域ネットワークにおけるキラーアプリケーション

第 1 章 10G WG の概要

10G WG は超広帯域ネットワークにおけるキラーアプリケーションの構築を目指し、2002 年 12 月に発足した。本章ではまず超広帯域ネットワークにおけるキラーアプリケーションの方向性を解説し、本年度の活動について述べる。

1.1 超広帯域ネットワーク環境の有効利用

IEEE802.3ae[123]、OC-192、OC-768 といった超広帯域ネットワークの登場に伴い、その有効利用方法が求められている。

10 Gbps、40 Gbps というネットワークにおいて 2 つの観点からキラーアプリケーションのアプローチを行う。1 つは帯域を潤沢に使い、スケーラビリティの高いサービスを低コストでユーザに提供するタイプのものである。VoD (Video on Demand) などが例として挙げられる。もう 1 つはネットワークを潤沢に消費しない代わりに高品質なサービスを提供するタイプのものである。低遅延、低ジッタな映像配信システムなどが好例である。

これらのアプリケーションを実現するにあたり、ネットワークのパフォーマンスを活かしきるために PC の設定、ネットワークの構築方法などが重要となる。特に超広帯域ネットワークでは転送機器のパフォーマンスがネットワークのパフォーマンスを下回ることが多い。具体的にはハードディスク、メモリ、バスの選定、OS の設定などが挙げられる。限られた資源の中で最大のパフォーマンスを発揮する必要がある。本 WG ではキラーアプリケーションの構築に加え、このような最適な広帯域ネットワーク環境の模索を行う。

1.2 本年度の活動

本 WG では 2003 年 WIDE 春合宿において、超広帯域ネットワークを題材とした特設セッションを行っ

た。特設セッションの詳細を第 2 章にて述べる。特設セッションにおいて、超広帯域ネットワークにおけるアプリケーション開発に対する方向性を導きだした。

今年度は、最適な広帯域ネットワーク環境の模索をテーマに 2 つの実験を行った。1 つは有用なサービスによって 1 Gbps の帯域を消費する、というものである。有用なトラフィックによる 1 Gbps 帯域消費テストを第 3 章で述べる。もう 1 つは FreeBSD のカーネルオプションの変更により、ネットワークパフォーマンスの変化の測定を行う、というものである。カーネルオプションの変更によるネットワークパフォーマンステストを第 4 章で述べる。

キラーアプリケーションの開発としては、4 つのアプリケーションについて行っている。スケーラビリティの高いサービスの観点より 3 つのアプリケーションに関する報告を行う。まずはじめに DV による遠隔 TV 視聴システムを第 5 章にて述べる。次に分散環境におけるオーケストラ演奏機構を第 6 章にて述べる。3 つめに分散環境における効率的な高品質映像制御に関する研究を第 7 章にて述べる。

高品質なサービス提供の取り組みとして、適応型映像・音声配信機構の設計と構築を第 8 章にて述べる。

第 2 章 特設セッション

本 WG では、2003 年 WIDE 春合宿において超広帯域ネットワークを題材に特設セッションの運営を行った。特設セッションは以下の 2 部によって構成された。

- 超広帯域ネットワークの現状と技術
10 Gbps 通信対応のインフラストラクチャに関する開発状況を各ベンダの方に speaker として話して頂き、議論
- 超広帯域ネットワークとキラーアプリケーション
超広帯域ネットワークにおけるアプリケーションのあり方、有意義なアプリケーション開発の方向性に関する議論

2.1 超広帯域ネットワークの現状と技術

本セッションでは 10 Gbps ネットワークを中心に技術動向、現状での運用上の問題点、技術課題などが話し合われた。

本セッションでの議論より、WIDE として振動や配線に弱い 10 Gbps ネットワークの扱い方マニュアルの必要性が導きだされた。

2.2 超広帯域ネットワークとキラーアプリケーション

本セッションでは超広帯域ネットワークの有効利用、普及を進めるためのキラーアプリケーションに付いて議論された。

ここで議論されたアプリケーション、方向性の一例として以下の 4 つのものが挙げられた。

- n*ストリーム
複数の映像・音声配信
- PC のコンポーネントをネットワーク接続
iSCSI のようなもの。キーボード、マウスといったものからビデオカード、ディスプレイのようなものまで全てをネットワーク接続する
- 投機的な通信
ユーザが次に要求すると思われるコンテンツを先に送信し、エンドノードで選択するスタイル
- ストレージとしてのネットワーク

本 WG では今年度の活動として、n*ストリーム形式によるアプリケーション制作として遠隔 TV 操作、視聴システムに取り組んだ。DV による遠隔 TV 視聴システムに関する進捗状況は第 5 章にて述べる。

第 3 章 有用なトラフィックによる 1 Gbps 消費テスト

10 Gbps という超広帯域ネットワークに対する事前の取り組みとして、1 Gbps のネットワークを有用なトラフィックで消費するテストを行った。

次節では本実験で用いた DVBS (Digital Video Broadcast System) について述べる。ネットワークベンチマークソフトウェアなどではなく、映像コンテンツの転送を行う。これを有用なトラフィックと定義し、実験を行った。

3.1 DVBS

DVBS[349] は、リアルタイム転送を行う DVTS (Digital Video Transport System) [220] に対し、HDD (Hard Disc Drive) より DV データを読み込み、マルチキャスト、もしくはユニキャストで転送を行うアプリケーションである。

デモは 500 MB の DV ファイルを用い、複数のユニキャスト転送を行うことで行った。

3.2 実験内容

実験トポロジを図 3.1 に示す。

実験には 5 台の PC と 3 台のスイッチを用い、計測は MRTG を用いて行った。それぞれの PC は DV ストリームの送受信の両方を担当した。

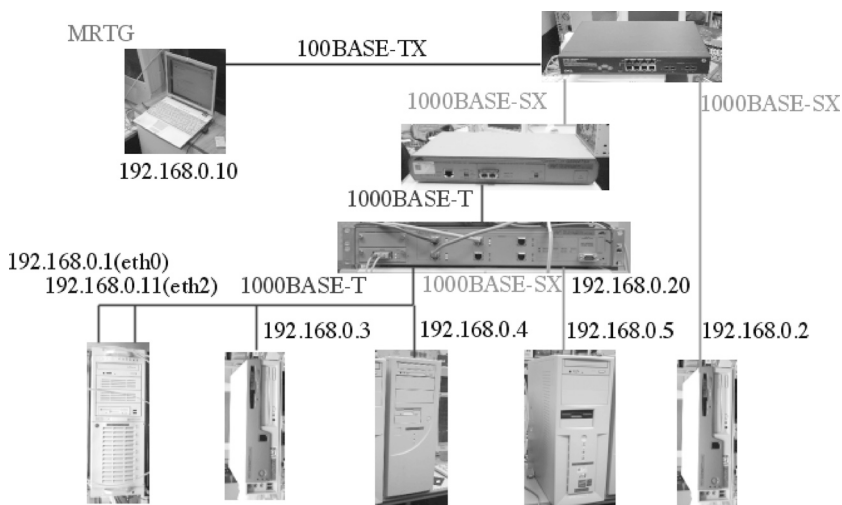


図 3.1. 1 Gbps 消費テスト：実験トポロジ

PCのスペックはPentium III 1.24 GHz から Pentium4 3.0 GHz までのマシンを用い、HDD は全て ATA-100 のものを使用した。OS は Linux (Kernel2.4.20) である。

DV ストリームは1ストリーム辺り 32 Mbps を消費する。本実験では1台に付き約8ストリームの送信を行った。

3.3 実験結果

1280 Mbps のトラフィックを記録することができた。中心となるスイッチでは1ポートあたり 250-650 Mbps のトラフィックの確認ができた。ただし途中で zombie プロセスが出現するなど、安定性に欠ける面があり、DVBS の改善が将来的に必要なである。

3.4 まとめ

本実験の課題を以下に示す。

- Kernel チューニングを施しての実験
ネットワーク周りのパラメータの変更等によるパフォーマンスの改善
 - HDD に SCSI や RAID を用いての実験
- 本 WG では実験を通して得た課題より、本実験

では採用しなかった FreeBSD に対する Kernel パラメータのチューニングを行い、評価を行った。次章では FreeBSD Kernel パラメータの変更によるスループット計測の実験について述べる。

第4章 FreeBSD Kernel パラメータの変更によるスループット計測

前章で述べた 1 Gbps 帯域消費テストでの課題を受け、FreeBSD における Kernel パラメータの変更によるスループット計測実験を行った。

4.1 実験内容

スループットの計測には NetPerf を用いた。NetPerf は5回ずつ実行し、平均値の取得を行った。ジャンボパケットと MTU1500 を想定したパケットサイズで実験を行った。

変更の対象としたパラメータは以下の2つである。それぞれのパラメータは `sysctl` にて変更を行った。

- `kern.ipc.maxsockbuf`
ソケットのバッファサイズを変更するパラメータ

NetPerf PC



- CPU: Xeron 3.06GHz
- メモリ: DIMM 2GB
- NIC: 1000BASE-T(intel)
■ PCI-Xにて接続
- FreeBSD 4.9PreRelease

スイッチ



1000BASE-T

NetServ PC



1000BASE-T

- CPU: PIII 1.24GHz
- メモリ: PC-133 512MB
- NIC: 1000BASE-T(nge)
- FreeBSD 4.9PreRelease

図 4.1. Kernel パラメータの変更によるスループット計測：実験トポロジ

表 4.1. kern.ipc.maxsockbuf の変更 (単位 Mbps)

| 値 | TCP | TCP MTU1500 | UDP | UDP MTU1500 |
|--------|--------|-------------|--------|-------------|
| 262144 | 693.84 | 545.48 | 911.14 | 642.32 |
| 393216 | 700.8 | 546.97 | 911.14 | 646.69 |
| 524288 | 696.94 | 556.71 | 911.12 | 649.89 |
| 655360 | 693 | 546.74 | 911.16 | 644.138 |
| 786432 | 691.84 | 547.2 | 911.18 | 645.27 |

タ。初期値は 262144。NetLib でエラーが発生するため、今回の環境では上限は 786432 であった。

• net.inet.tcp.sendspace

TCP バッファサイズの標準値。初期値は 32768。NetLib でエラーが発生するため、今回の環境では上限は 233016 であった。

4.2 実験結果

kern.ipc.maxsockbuf の値を変更、それぞれのスループットを計測した結果を表 4.1 に示す。今回の実験環境では 524288 が最も適当な結果となった。これは TCP の最大ウィンドウサイズである 65535 よ

り $65535 * 2^3 = 524288$ に関係する結果となった。また、MTU が 1500 のときに最も効果が大いという結果になった。

kern.ipc.maxsockbuf の値をデフォルト値に設定し、net.inet.tcp.sendspace の値を変更、それぞれのスループットを計測した結果を表 4.2 に示す。同様に kern.ipc.maxsockbuf の値を実験の結果最もパフォーマンスの高かった 524288 に設定した場合の結果を表 4.3 に示す。

今回の実験では kern.ipc.maxsockbuf の値に関係なく、36864 以上が適当という結果が得られた。

4.3 今後の課題

今後の課題は以下の 2 つである。

- net.inet.tcp.recvspace の設定による評価
今回は送出する TCP バッファサイズのみ注目して実験を行った。同様に受信する TCP バッファサイズの設定を行い、実験を行う必要がある。
- NetServ を行うマシンをサーバスペックのものに変更しての評価
今回の実験では NetPerf の受信ソフトウェアである、NetServ の実行環境として一般的な PC を用いた。サーバスペックのものを用いることにより、値の向上が予想される。

また、今回得られた結果を元に本 WG では超広帯域ネットワークにおける次世代 TCP の開発にむけて活動を行う。

表 4.2. net.inet.tcp.sendspace の変更 (1)
(単位 : Mbps)

| 値 | TCP | TCP MTU1500 |
|--------|---------|-------------|
| 32768 | 693.29 | 548.27 |
| 33792 | 723.95 | 547.8 |
| 34816 | 727.388 | 548.66 |
| 35840 | 732.26 | 548.05 |
| 36864 | 755.352 | 547.45 |
| 49152 | 754.12 | 548.1 |
| 65536 | 752.08 | 548.88 |
| 98304 | 750.84 | 546.84 |
| 196608 | 753.22 | 546.52 |
| 233016 | 763.71 | 542.69 |

表 4.3. net.inet.tcp.sendspace の変更 (2)
(単位 : Mbps)

| 値 | TCP | TCP MTU1500 |
|--------|---------|-------------|
| 32768 | 698.46 | 543.50 |
| 33792 | 713.35 | 544.43 |
| 34816 | 739.92 | 543.8 |
| 35840 | 742.76 | 543.19 |
| 36864 | 755.63 | 544.47 |
| 37888 | 755.234 | 544.14 |
| 65536 | 751.13 | 545.27 |
| 233016 | 757.414 | 541.17 |
| 466033 | 751.48 | 531.88 |

第 5 章 Establishment of Controlling IEEE1394 devices over the network

This chapter proposes the design and implementation of IEEE1394 device control over network. We have extended the wired connection between IEEE1394 devices and computer into

IP network, formerly connected with IEEE1394 interface. Implemented system could use all of the features which IEEE1394 device carries, via IP network.

Using implementation on IEEE1394 device control over IP network linked with DVTS which transports DV Data through IP network, we can construct radio-wave broadcast transport system through IP network. Using implemented radio-wave broadcast transport system, we can watch and store radio-wave broadcast over IP network.

From this research we achieved extension of the connection between IEEE1394 device and computers to control IEEE1394 devices. By constructing a radio-wave broadcast transport system, we have experimentally proved the availability of our implementation on IEEE1394 device control over IP network.

5.1 Introduction

Consumer AV appliances are being transformed from analog to digital technology. Digitalization of AV appliances made simpler interconnectivity between computers. IEEE1394[2, 116, 118] is a common interface connecting AV appliances and computers or other digital AV appliances. DV camcorders are the mainstream of AV appliances using IEEE1394 interface.

There are three reasons why IEEE1394 adopted in these kinds of AV appliances. 1) Fast transfer speed is indispensable for audio and video data. 2) Small connector is convenient to connect small devices. 3) Supports hot-swap connection. When new IEEE1394 device is connected or removed, IEEE1394 bus automatically reconfigures the new node.

There are two functionalities on IEEE1394. 1) Controlling the device, and 2) transportation of the media.

Capability of IEEE1394 to control its connected device is due to the fact that IEEE1394 interface is not designed to connect only with computers. IEEE1394 are also designed to connect with consumer appliances. There are a few software imple-

mentations to control IEEE1394 devices.

There are two sample applications transporting IEEE1394 features through the network, 1) DVTS[220, 221] (Digital Video Transport System), and 2) Video Transport System designed for MPEG2TS[202] (MPEG2 Transport Stream). By using these applications, construction of Internet broadcast system using IEEE1394 connected device and computer with Internet Protocol (IP) can be established in simple manners. These implementations are already in practical use for video conferencing and high quality streaming.

Comparing the feature of data transmission in implementation of IEEE1394 interface, there are only few softwares implementing the control mechanism of IEEE1394 connected device. However, there is no implementation to control devices over IP network. In some operating systems, Application Programming Interface (API) for controlling IEEE1394 device connected with IEEE1394 do exists.

There is specification for IEEE1394 device control for several types of devices. For example, VCR, TUNER and TAPE RECORDER. In spite of the existence of specification, there are many devices without these implementations.

In this paper, we define “Radio-wave Broadcast” for television broadcast which uses radiowave for transmission and “Internet Broadcast” for broadcast system which uses IP network for transmission.

5.2 Controlling the devices connected

locally

Some IEEE1394 devices require vendor distributed applications for controlling the device. This paper implements controlling of these devices connected via IEEE1394. In this research, we designed and implemented IEEE1394 device control over IP network. IEEE1394 has platform capability to control devices locally. IEEE1394 has a limitation in its physical interface for the wire length of the controlling media. IEEE1394 cable connection has a restriction in length of

4.5 meters.

There is no implementation to extend controllability of the device from local cables to over IP network. For sample implementation of IEEE1394 device control over IP network, we designed and implemented “Radiowave Broadcast Transport System” linking with DVTS.

5.3 IEEE1394 Device drivers

Some Operating System (OS) supports API for IEEE1394 device control (Table 5.1).

Table 5.1. IEEE1394 API for several OS

| OS | API |
|-----------|-----------------------|
| FreeBSD | Device file |
| Linux | Device file & Library |
| WindowsXP | Driver & Library |

OHCI (Open Host Controller Interface) link chips are de-facto standard for IEEE1394. OHCI chip cannot be placed in promiscuous mode. Therefore, monitoring the traffic through the interface is difficult. Before OHCI link layer appears, Texas Instruments PCI Lynx link chip has been used. This link chip could be placed in promiscuous mode.

5.4 Evaluation

We compared Windows application comes with DV converter for local feature with our implementation for remote feature. Comparison of the local features and remote features are shown in Table 5.2.

Feature of locally connected with IEEE1394 cable has been satisfied in remotely connected with IP network. All of the feature IEEE1394 device carries could be used from remote with no limitation in length.

Table 5.2. Comparison of local and remote features

| | Local | Remote |
|----------------------|-----------|-----------|
| Channel selection | Supported | Supported |
| Audio selection | Supported | Supported |
| Frequency adjustment | Supported | Supported |

5.5 Conclusion

We have constructed practical sample implementation for IEEE1394 device control over IP network. Our method to control IEEE1394 device over IP network has been confirmed. Our implementation of IEEE1394 device control over the IP network can extend the connection between IEEE1394 device and computers on the IP network.

Using our method, both IEEE1394 devices following the AV/C specification and IEEE1394 devices which doesn't follow the AV/C specification can be controlled over IP network. Functions IEEE1394 device carries can be used from over IP network and could work on more advanced application.

Using this implementation with existing audio and video transport system, we have designed and implemented audio and video transport system using the network. Using this system, client can watch the radiowave broadcast outside the coverage and store the radiowave broadcast to D-VHS tape and into DV files.

In future work, we will assort each type of device to construct framework for IEEE1394 device control over IP network.

第 6 章 分散環境におけるフィードバックを用いたオーケストラ演奏機構の構築

本章では超広帯域ネットワークにおけるアプリケーションの開発として、ネットワークを介した分散環境における音楽共同制作モデルを提案し、モデルに基づいた設計と実装を行った。本機構を用いることにより、指揮者映像、他の演奏者の音声を参照しながら演奏を行うことが可能となった。

6.1 目的：共同制作環境の構築

ネットワークの広帯域化に伴い、データサイズの大きな品質の高い映像・音声データを、リアルタイムに双方向、複数転送することが容易となった。こ

W I D E P R O J E C T 2 0 0 3 a n n u a l r e p o r t

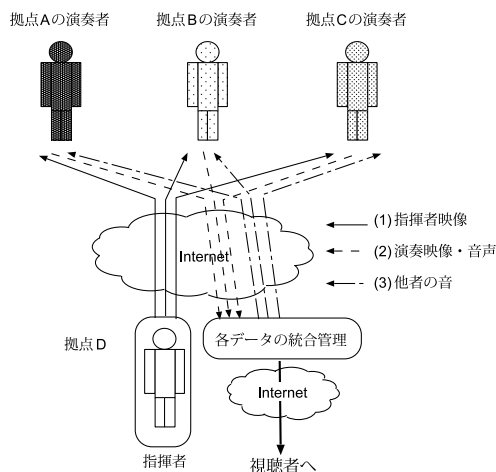


図 6.1. 分散環境における音楽共同制作モデル

れにより、ネットワーク上でのマルチメディアコンテンツ共同制作活動が活発化している。

本研究の目的は、広帯域化・低遅延化が進む IP ネットワーク環境を利用したマルチメディアコンテンツのリアルタイム共同制作環境を実現することである。本章ではリアルタイム共同制作環境の実現に向け、以下の3つに着目し、分散環境におけるリアルタイムメディア共同制作を実現する。

1. 共同制作者間の連携

他者の音声をモニタリングすることによる分散環境における共同制作の効率化

2. 共同制作者間の時間軸共有

動作の開始、間隔を制作者間で共有

3. 拠点毎に異なる遅延時間の管理

ネットワーク伝送遅延、収録環境・転送機器の差異による遅延時間の取得

リアルタイムメディア共同制作のうち、上記の3点の要求が厳しいものの1つにオーケストラのライブコンサート、練習がある。本研究ではリアルタイムメディア共同制作の一つとしてオーケストラを対象に設計、実装を行う。

6.2 提案：分散環境における音楽共同制作

前節で述べた3つの要件を元に、本章で想定する分散環境における音楽共同制作環境のモデルを図6.1に示す。中心を指揮者のいる拠点をDとする。まず指揮者の映像を各拠点の演奏者に送信する。タイミングを取得するための指揮者映像をタイミングデータと呼ぶ。送信された映像を元に各演奏者は演奏を行い、拠点Dへと演奏データを送信する。拠点D

は各拠点から受信した演奏データを各拠点に送信する。これにより、各拠点の演奏者は他の演奏者の音を聴取することができる。他の演奏者の音声データをフィードバックデータと呼ぶ。

これらのデータは全て安定したデータ転送がなされる必要がある。特に映像・音声データの送信が不安定な場合、音声途切れるなどの影響があるため、合奏全体の品質に影響する。

途中経路に遅延発生装置を使用し、ネットワークを介した異なる拠点の2人の演奏者に音のみの到達性を確保し、合奏を行ったレポート [356] によると、許容遅延時間範囲は50ms-80msとされている。この許容値より、異なる拠点のクライアントを $P_{(1)} \dots P_{(2)}$ とした場合、それぞれのクライアントに対し、タイミングデータ出力時間からフィードバック受信時間の差分 $Pd_1 \dots Pd_n$ が存在する。これより、演奏可能許容遅延時間は

$$\text{Max}(\overline{Pd_{(1)-(n)}}) = \text{Delay} < 80 \text{ ms}$$

によって求められる。

6.3 実装：オーケストラ演奏機構の構築

IEEE1394 から Isochronous 転送を行い、安定した映像・音声データの入出力を行う。これらの映像・音声データをネットワークを介して転送を行うために DVTS、特に DVTS for MacOSX をベースとした実装を行った。

本研究では以下の4つのモジュールを作成した。

- タイミングデータ送受信部
指揮者映像などを送受信することにより、演奏のタイミングの共有を行う。
- フィードバックデータ送受信部
他の演奏者の音声をマスタからクライアントに対し送信することで音声データからの和音、ダイナミックスの確認を行う。
- クライアントデータベース
クライアント毎に異なる IP アドレス、音声データ送信用ポート番号といった状態の他、映像・音声データ用バッファなどを登録する。
- 遅延管理部
ネットワーク伝送遅延時間、クライアント内部処理時間の取得を行う。

6.4 評価：本機構の実現した機能

測定環境を図 6.2 に示す。クライアントはマスタから受信したタイミングデータ、フィードバックデータを IEEE1394 機器に出力する。そのため、クライアント側にはそれぞれ 2 台のメディアコンバータを IEEE1394 ハブを介して接続する。また、映像・音声データの入力を IEEE1394 機器 (DV カメラ) から受ける。

6.4.1 データ送信安定性の評価

クライアントから送信される映像・音声データ、マスタから送信されるタイミングデータ、フィードバックデータの 3 つに対し、パケット送信の安定性と粒度の測定を行う。

クライアントがタイミングデータ、フィードバックデータ受信を同時に平行して行った際の映像・音声データ送出量を計測した。計測は tcpdump を用い、該当するポート番号に対し 10 分間測定した。

クライアントにおける映像・音声データ送信量の結果を図 6.3 に示す。クライアントがタイミングデータ、フィードバックデータの受信を平行して行いながら一定の映像・音声データを送出していることが分かる。

マスタにおけるタイミングデータ (映像・音声) の測定結果を図 6.4 に示す。タイミングデータの送信は 1 つのスレッドで全てのクライアントに対して行われる。そのため若干のばらつきが発生するが、ほぼ一定の映像・音声データを送出していることが分かる。

フィードバックデータ (音声のみ) 送信量の測定結果を図 6.5 に示す。マスタがタイミングデータ送信、映像・音声データ受信を平行して行いながら一定のフィードバックデータを送信していることが分かる。

6.4.2 演奏可能許容時間と処理速度

演奏開始から他の演奏者の音声を聴取するまでの時間が、80 ms 以内に行われる必要がある。本項では演奏の開始の合図となるタイミングデータ受信時から、映像・音声データの送信、フィードバックの受信までの一連の処理に必要な時間を計測した。80 ms から計測された値の差分を取ることで、ネットワーク伝送時間として扱える時間が導きだされるため、本研究の有効範囲が求められる。

演奏者が演奏を開始し、他の演奏者の演奏を聴くまでの時間を拡張 RTP フォーマットの一部を変更することにより計測を行った。計測は計算能力の高

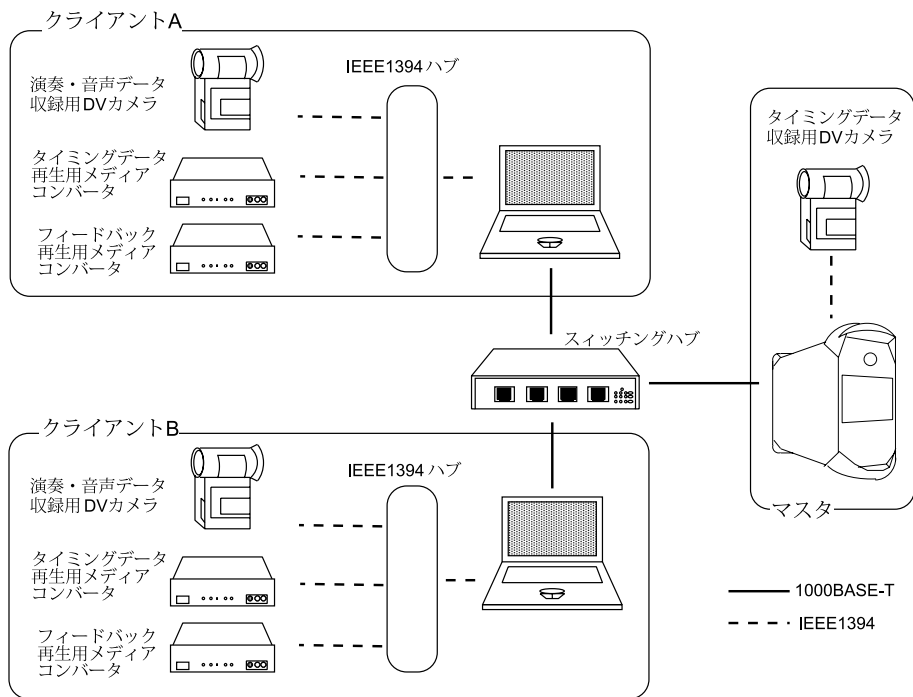


図 6.2. 測定時の環境

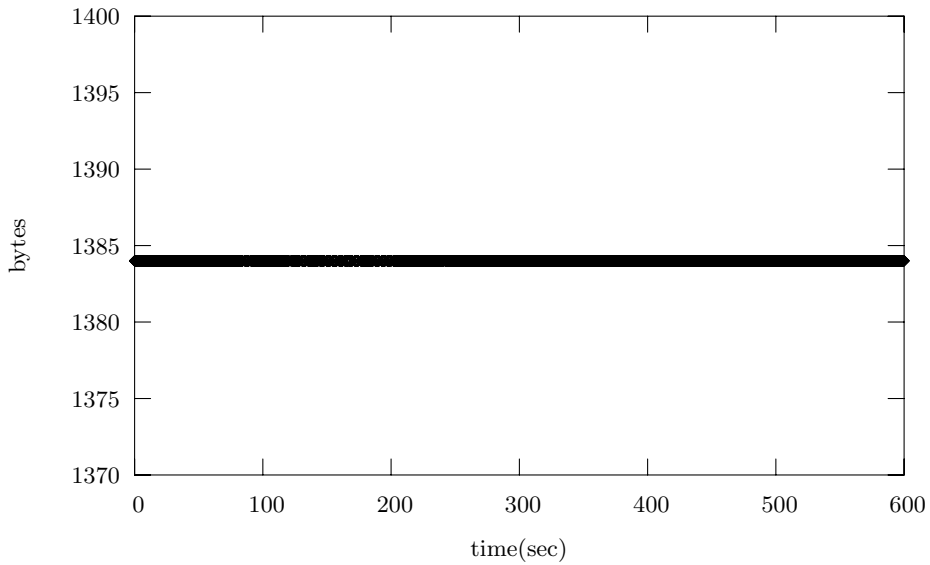


図 6.3. クライアントにおける映像・音声データ送信量

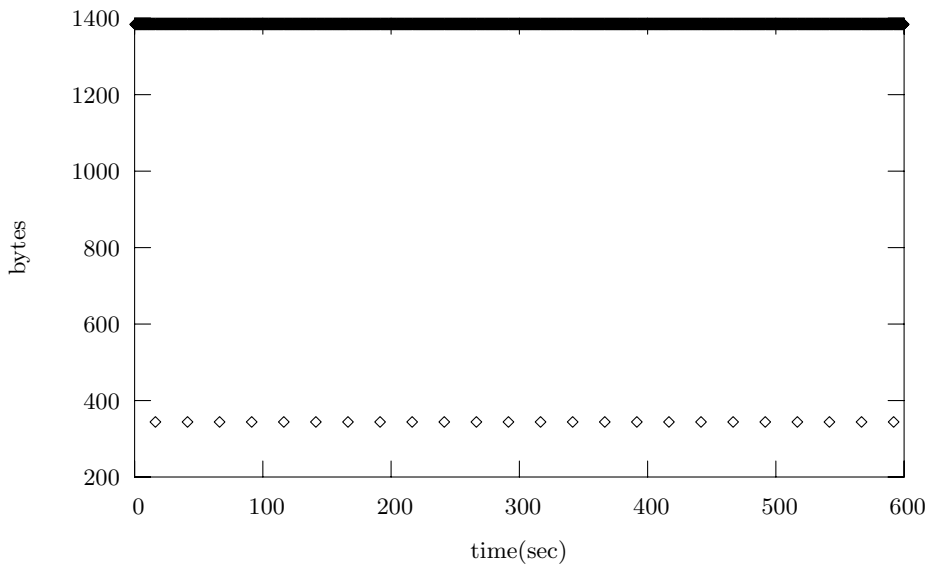


図 6.4. マスタ：タイミングデータ送信量

クライアント A において、10 分間行った。

演奏開始をタイミングデータ受信時の時間と仮定した計測を行う。マスタがタイミングデータを 1 から t まで送信する際にパケット (t) 内の拡張 RTP フォーマット内にマスタ側 CPU 時間 $M_{(t)}$ を挿入する。この際の単位は 10 msec である。クライアントがタイミングデータ t 受信時に、取得されるクライアント側 CPU 時間を $C_{(t)}$ とする。受信したクライアントは、マスタ側 CPU 時間 $M_{(t)}$ を映像・音声データ内の拡張 RTP フォーマットに複写する。映像・音声データを受信したマスタは、映像・音声データ内

のマスタ側 CPU 時間 $M_{(t)}$ をフィードバックデータ内 RTP フォーマットに複写、クライアントに転送する。クライアントが拡張 RTP ヘッダ内に $M_{(t)}$ を含むフィードバックデータ受信時の CPU 時間を $F_{(t)}$ とする。この際、タイミングデータ受信からフィードバック受信するまでの時間 $D_{(t)}$ は

$$D_{(t)} = F_{(t)} - C_{(t)}$$

によって求められる。

以上の測定の結果を図 6.6 に示す。数 ms の遅延時間で処理を行う場合が多いが、ばらつきが目立つ。

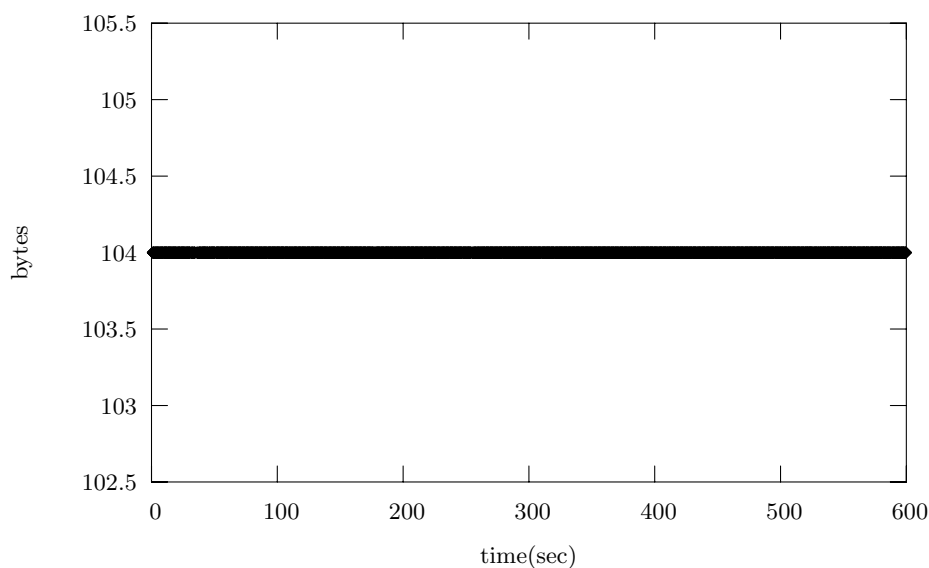


図 6.5. マスタ：フィードバックデータ送信量

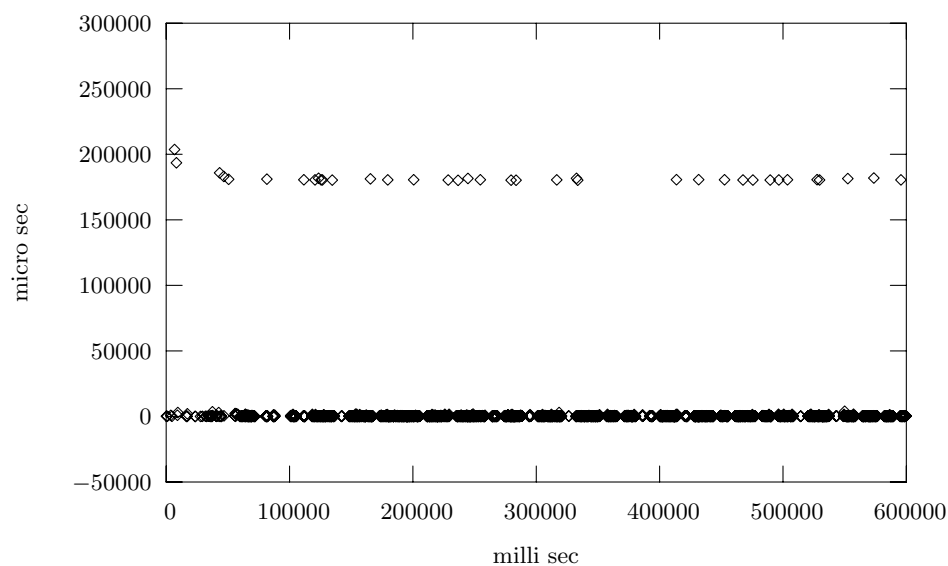


図 6.6. 演奏可能許容時間と処理速度

平均所要時間は 37 ms であるため、実験環境におけるネットワーク伝送遅延を 0 と仮定した場合、ネットワーク伝送遅延が往復で 43 ms 以内の拠点であれば演奏可能であると言える。

所要時間のばらつきは本実装の母体となっている DVTS for MacOSX に起因するところが多い。DVTS for MacOSX の受信機構のパフォーマンスの改善を行うことにより、本研究の有効範囲の拡大が予想される。

6.5 まとめ・今後

本研究ではフィードバックを用いたオーケストラ演奏機構の構築を行った。本機構を用いることにより、指揮者映像、他の演奏者の音声を参照しながら演奏を行うことができる。演奏の基準となる映像データの転送、他の演奏者の音声のフィードバックを行うことにより、映像データによる演奏タイミングの共有と、音声データによる和音・ダイナミクス確認の両立ができる。

今後はパフォーマンスの改善、Viewer の作成など

を行う。また、実際のオーケストラにおいて実験を行う。

第7章 分散協調環境における効率の良い高品質映像制御に関する研究

本章では超広帯域ネットワークにおけるアプリケーションとして、SD (Standard Definition) DV を用いた True VoD システムの設計・実装・評価を行い、既存の VoD システムと比較して、時間粒度の細かい高品質映像の再生制御を可能にした。本システムでは、実時間処理を用いることでサーバ上の計算機資源を効率的に利用し、SDDV を用いた大容量映像データ転送、および映像制御を許可した場合のストリーミングセッション増加に対応した。本研究によって、有限な計算機資源上でより計算機資源の利用効率のよい、高品質映像フォーマットを用いた True VoD システムが実現できる。

7.1 背景・目的

計算機・ディスプレイなどの高性能化によって、高品質なデジタル映像を用いたアプリケーションが発達した。また、データリンクが広帯域化し、高品質映像の転送アプリケーションが登場した。これらの環境、および技術の発達から VoD やノンリニア編集への要求が高まっている。

本研究の目的は、DV 等の高品質映像フォーマットを用いて、映像の再生制御が可能な True VoD システムの構築である。また、VoD サーバ上の計算機資源を効率よく活用することで、高品質映像フォーマットを用いる際のオーバーヘッドを考慮し、多数のクライアントに乱れることのない映像を配信することを目標とする。

7.2 True VoD 実現へのアプローチ

映像の再生制御は、早送りや巻き戻しを含め、映像のフレームを制御することで実現することができる。本研究ではフレームの制御に対して、フレーム内圧縮フォーマットに着目した。DV フォーマットのタイムコードを用いることで完全なフレームの特定と制御ができる。また、再生制御を可能にすることによって発生するクライアント数の増加、SDDV

を用いることによって発生する大容量データ転送に対して、VoD サーバの OS にリアルタイム OS を用いて、サーバ上の計算機資源の有効活用することで解決を試みた。

7.3 設計・実装：True DV VoD システム

本研究では、前節で述べたアプローチを基に以下の4点を設計要件とした。

- クライアント・サーバ間で再生制御要求の送受信
- 任意映像フレームの特定・読み込み
- 安定した DV データ入出力
- 情報家電への移植性の考慮

ART-Linux (Kernel 2.4.20 base) + gcc2.95.3 の環境でこれらの要件を実装した。図 7.1 に実装概要図を示す。映像の再生制御要求が無い場合は、DV フレームを読み込み、読み込んだフレームのタイムコードを抽出・格納した上でクライアントへ送信する。その後 DV フォーマットにおいて次のフレーム転送に必要な時間まで休眠する。

再生制御要求があった場合は、格納されているタイムコードと要求されるタイムコードを比較し、次の DV データ読み込みに対して、読み込む DV フレームを通達する。また、再生制御要求が DV 再生速度に関係する場合は、フレーム間引き、もしくは読み込みサイクルの周期時間を変更することで対応する。

また、VoD クライアントには DVTS のクライアントソフトウェア (xdvshow, Windows 版 dvrecv) を使い、その拡張としてタイムコード表示機能を付加した。

7.4 評価

7.4.1 定性評価：再生制御

本研究では、DV フォーマットのタイムコードを用いて、すべての映像フレームの特定を可能にした。また、実時間処理を用いて、映像フレームの転送タイミングの制御を可能にした。これによって、あらゆる VCR 制御が可能になった。

7.4.2 定性評価：情報家電への移植性

実装環境に ART-Linux を用いることによって、既存の汎用 API、ライブラリが使用可能で、情報家電分野で実績の高い Linux を Kernel に持つため、移植性が高い。

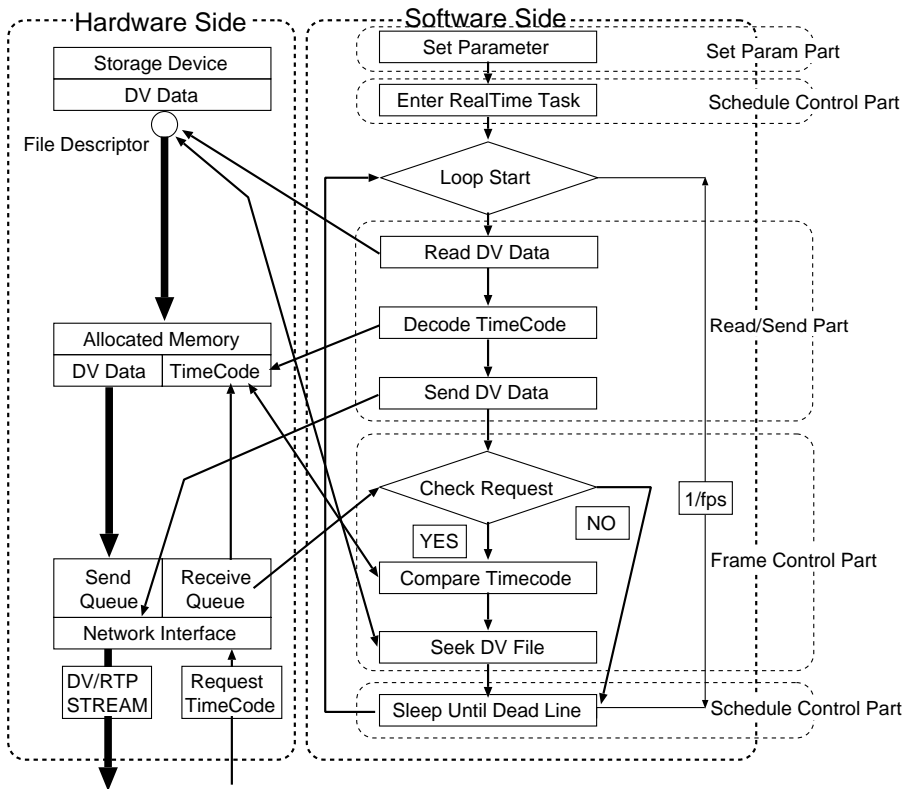


図 7.1. 実装概要図

7.4.3 定量評価：データ入出力の安定化

実時間処理と通常処理の差を映像データ読み込み、および送信において比較を行った。

優先度最高リアルタイムタスクにおける、DV の読み込みを行った際の計測結果を図 7.2 に示す。複数の DV データ読み込みタスクが存在した場合でも、互いのタスクが協調して、データの読み込みに必要な時間の揺らぎが小さく、安定したデータ入力を実現していることが読み取れる。

非リアルタイムタスクにおける、DV の読み込みを行った際の計測結果を図 7.3 に示す。複数の非リアルタイムタスクを立ち上げた場合、個々のタスクはタイムシェアリングシステムで管理されるため、頻りにタスクスイッチングを繰り返し、データ読み込みに必要な時間がリアルタイムタスクと比較して長時間で、大きく揺らいでいることが読み取れる。

メモリ上にマップされた DV フォーマット NTSC 方式の映像 1 フレームを、Network Interface から送出するまでに要する時間の揺らぎを計測した結果から、通常の Linux Kernel を用いた場合と、ART-Linux のリアルタイムタスクを用いた場合の差異を示す。

計測に用いたタスクは以下の通りである。

1. 優先度最高のリアルタイムタスク + 優先度最高のリアルタイムタスク

図 7.4 中 prio max * 2

2. 非リアルタイムタスク + 非リアルタイムタスク

図 7.4 中 non real time * 2

図 7.4 は、上記の項目から得たデータをグラフに示している。

第 1 項は、リアルタイムタスクの利用によって、DV データ送信中にネットワークインターフェースの資源を最大限に活用していることがわかる。これは、送信処理の開始から終了までに他のタスクからの影響を最小限に押さえ、優先度最高のタスクが最小限のタスクスイッチングで送信処理を終えていることと、タスクスイッチングの回数が毎回一定であることを意味している。そのため、読み込んだ DV データの送信処理終了までの時間の揺らぎはほとんどない。

第 2 項は、ネットワークインターフェースの送信キューに対して、読み込んだ DV データを書き込んでいる最中にタスクスイッチングが発生した場合、データ送信処理終了までに長時間要していることがわかる。逆に、他のタスクからの割り込みによって

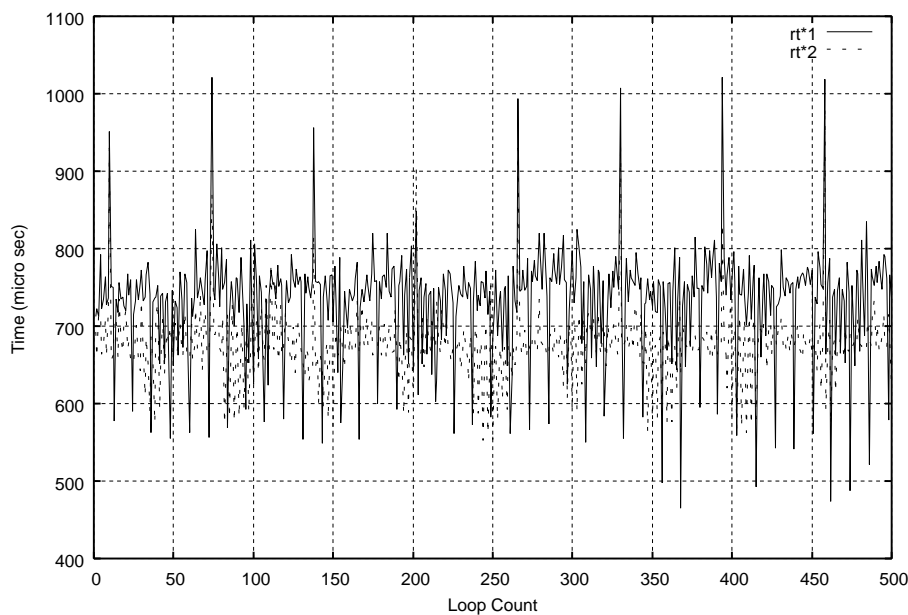


図 7.2. read に要する時間#1

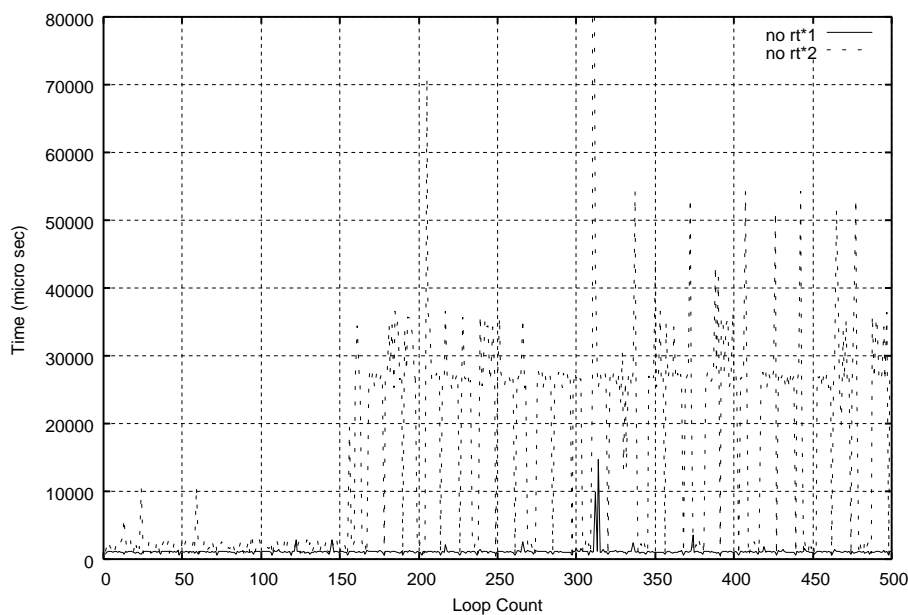


図 7.3. read に要する時間#2

タスクスイッチングが行われない場合は、第1項と同様に短時間で送信処理が終了していることがわかる。そのため、読み込んだDVデータの送信処理終了までの時間が大きく揺らいている。

7.5 まとめ・今後

映像再生制御に適したフレーム内圧縮フォーマットに着目し、SDDVを用いたTrue VoDシステムを構築した。また、転送する映像にSDDVを選択す

るために発生する大容量データ転送、映像再生制御を許可するために発生するストリーミングセッションの増加に対して、実時間処理を用いたサーバ上の効率の良い計算機資源活用を実現し、本研究のアプローチの有用性を示した。

今後は、RAID0・5やSCSI・SerialATAデバイスが持つCommand Queuingなどの評価を行い、最適なハードウェア構成などを模索する。その上で、本研究の成果を情報家電や既存の高品質映像転送技術

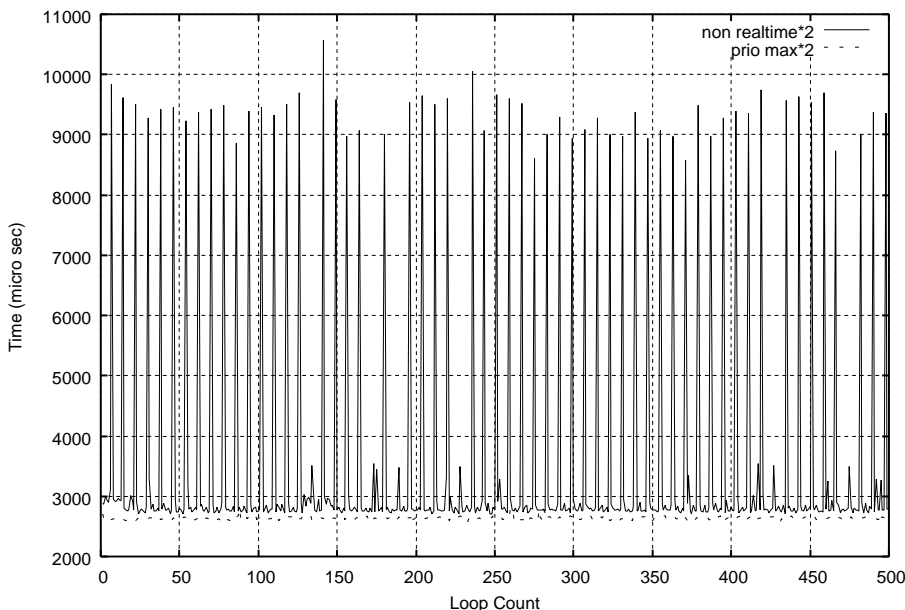


図 7.4. sendto に要する時間

ヘフィードバックしていく。また、クライアント環境の整備や、RTSP などと連携したインターオペラビリティの確保などを旨とする。

第 8 章 伝送特性に応じた適応型映像・音声配信機構の設計と構築

本章では超広帯域ネットワークにおける高品質なサービス提供として、ネットワーク状態の変化がエンドノード間で取得可能な伝送特性として現れることに着目し、エンドノード間で取得可能な到達パケットに対するジッタの揺らぎを主指標に用いた、新しい映像・音声配信のための輻輳制御手法を提案する。本手法は、送受信ノード間で必要な情報を交換し伝送特性の計測を行うため、中間ルータなどの特別な機器を必要としない End-To-End モデルでの輻輳制御が可能となる。本研究の提案する手法の有用性を実証するため、既存の映像・音声配信システムとして DVTS を採用し、本手法を適用するための設計、実装を行い、設計に基づき実装した DFCS (Dynamic Framerate Control) の評価を行った。評価結果より、伝送路状態の検知及び状態に適応したレート制御が行えることが確認された。

8.1 背景：通信ネットワークの広帯域化

データリンクの急速な広帯域化により、インターネットを介した比較的高品質な映像や音声の配信が可能となった。映像・音声配信の一般化により、企業レベルだけでなく、個人レベルでのストリーミングなども広く行われるようになった。

映像・音声配信の多くは、多少のデータ損失よりもリアルタイム性が重視されるため、再送制御や輻輳制御を考慮しない UDP (User Datagram Protocol) を用いる。インターネットは伝送路品質が常に変化するため、UDP を用いる映像・音声配信では、トランスポート層ではなく、アプリケーション層で輻輳制御が行う必要がある。

8.2 本研究の概要

本研究は、UDP を用いた映像・音声配信システムを対象とし、エンドノード間で取得可能な伝送特性としてパケット破棄率や到達時間の揺らぎ (ジッタ) を用いたネットワーク状態予測手法と状態適応型映像・音声配信機構を提案する。ジッタやパケット破棄率は、受信品質に大きく関係するため、これらの値に着目した輻輳制御手法は、End-To-End モデルを前提としたインターネットを介した映像・音声配信システムに適している。

本手法は、アプリケーション層にて、送受信ノ

ド間で時間情報を交換し伝送特性の計測を行い、その値に応じたフレームレート制御を行う。本研究では、本手法の有用性を実証するために、DVTS に対し本手法を適用し、ネットワーク状態に適応した輻輳制御を行う機構を設計、実装、評価する。

本手法を用いることで、中継ルータなどの特別な機器の準備なしにネットワーク状態にあわせた品質の決定、その品質で映像・音声を配信することが可能となる。

8.3 伝送特性による協調的輻輳制御手法

ジッタの揺らぎの多くは、計算機やネットワーク状態が高負荷になることで発生する。図 8.1 に DV ストリーム送信中に NetPerf による瞬間的なパーストトラフィックが発生した際の送信ノードでのジッタとパケットロスの変動を、図 8.2 に DV ストリーム送信中に別の DV ストリームが送信された場合の送信ノードでのジッタとパケットロスの変動をそれぞれ示す。

図 8.1 から、パーストトラフィックの発生とともにジッタは大きく増加し、パーストトラフィックの収束とともにジッタは大きく減少していることが確認できる。ネットワークの状態に応じてジッタは、正負に変動する。

図 8.2 から、パーストトラフィック発生から一定時間が経過すると、ジッタは小さな変化へ移行し、トラフィック収束まで一定以上の増減を繰り返すことが確認できる。ジッタは、輻輳状態直前やパーストトラフィックの収束直後など急激なネットワーク変化時に大きく変動するが、輻輳状態の継続やネットワーク状態の緩やかな変化時は緩やかな変動を続ける。

そこで、ジッタの値だけでなく、パケットロスの値を同時に補助的な指標とすることで、ネットワーク状態が緩やかな変化をしている場合でもネットワーク状態を把握できる。

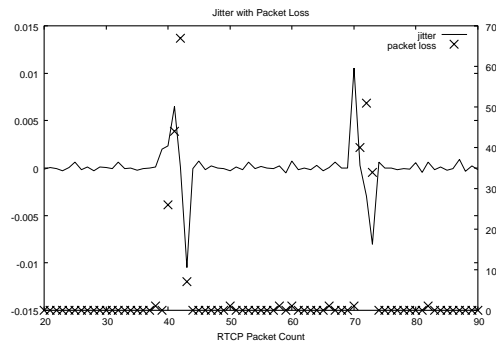


図 8.1. Rapid Burst Traffic (DV with NetPerf)

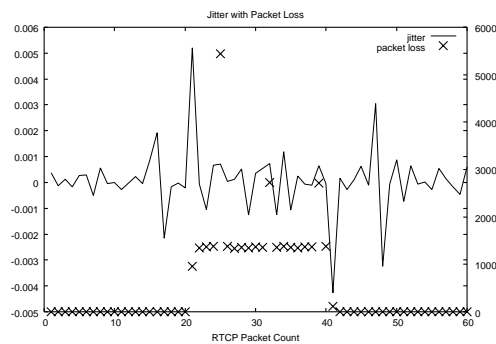


図 8.2. Continual Burst Traffic (DV with DV)

8.4 設計

図 8.3 に想定するシステム概要図を示す。

以下の 5 つの機能を本機構の満たすべき機能とする。

- 1 対 1 の通信ノード間で協調したジッタ及びパケットロス値の計測
- マイクロ秒精度でのジッタ計測
- 通信ノード間で通信状態情報の交換
- ジッタ及びパケットロスに基づく協調的輻輳制御
- 輻輳制御手法にのっとった DVTS のフレームレート制御

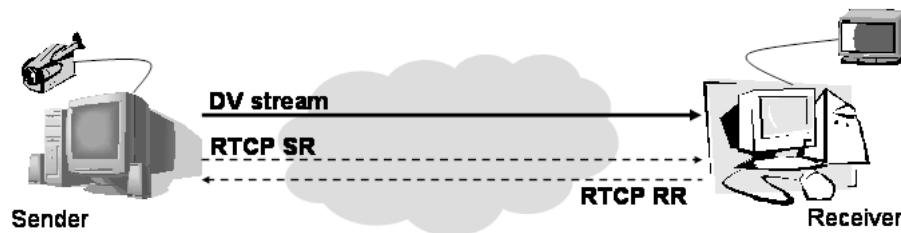


図 8.3. 想定するシステム概要

本機構は、送信部および受信部により構成され、それぞれ計算機および DV 機器を利用する。送受信ノード間での情報の交換を行うために、RTCP(Real-time Transport Control Protocol)を用いる。本機構は以下のモジュールで構成される。

- 送信者情報交換モジュール：
RTCP SR(Sender Report)を用いた送信者情報の交換
- 受信者情報交換モジュール：
RTCP RR(Receiver Report)を用いた受信者情報の交換
- 計測モジュール：
ジッタやフレームレートの値を計測
- フレームレート制御モジュール：
受信者情報に基づく映像レート決定とセット

動作概要

1. パケットの送信時間およびシーケンス番号を取得し、RTCP SR の必要なフィールドへセットする
2. RTCP SR パケットを送信する
3. RTCP SR パケットから送信者情報を取得する
4. パケットの受信時間及びシーケンス番号を取得する
5. ジッタ及びパケットロスの値を計測し、RTCP RR の必要なフィールドへセットする
6. RTCP RR パケットを送信する
7. ジッタ及びパケットロスの値を取得する
8. 新たなフレームレートの値を決定し、送信品質としてセットする

処理に必要な時間精度

ジッタ値の計測処理において時間情報を利用する。この時間情報の精度は、ネットワーク状態把握の精度に影響を与える。DV/RTP パケット 1 パケットが通過するのに必要な時間(理論値)は、112 マイクロ秒であるため、ネットワーク状態把握には、少なくともマイクロ秒単位での計測が必要である。

フレームレート制御アルゴリズム

送信レート制御は、送信中の映像フレーム間引き率を動的に変化させることで行う。フレーム間引き率の変更アルゴリズムを図 8.4 に示す。フレーム間引き率は、以下に挙げる 4 つの処理により変更され

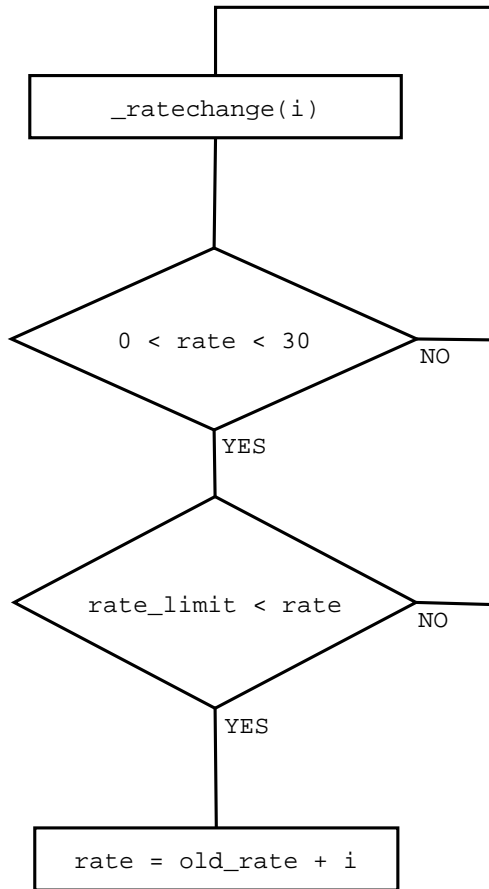


図 8.4. フレームレート制御処理

る。フレーム間引き率の値は、 $\frac{1}{2}$ であれば 2 というように、値が大きくなればなるほど間引き率が増加し、実際に必要な帯域幅は減少する。動的に変化する値の幅は、1 から 30 とする。また、不必要なフレームレートの変動を抑制するため、変更限度値を設定し、間引き率変更の際に動的に値を変更する。

- ジッタによる間引き率の加算処理
- ジッタによる間引き率の減算処理
- パケットロスによる間引き率の加算処理
- ジッタ及びパケットロスによる間引き率の減算処理

8.5 実装

実装は、FreeBSD-4.7R (IEEE1394 Patched) 上で行い、コンパイラは gcc 2.95.4 を用いた。また、ベースとなる DVTS は、dvts-1.0a を使い、必要な機能を拡張する形で実装を行った。

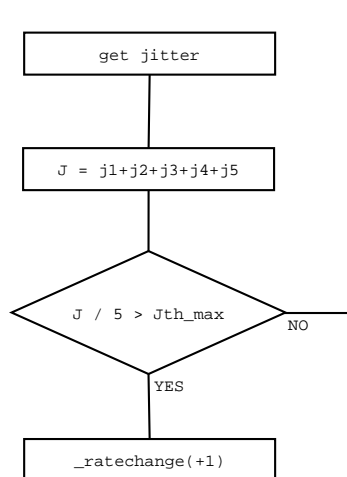


図 8.5. Rate-up by jitter

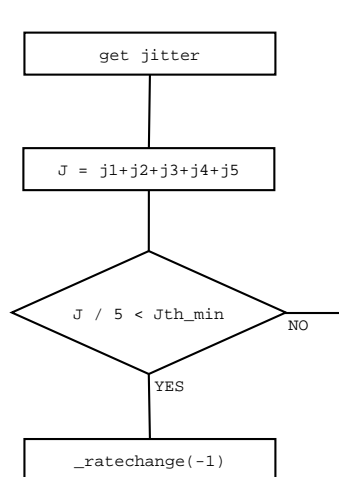


図 8.6. Rate-down by Jitter

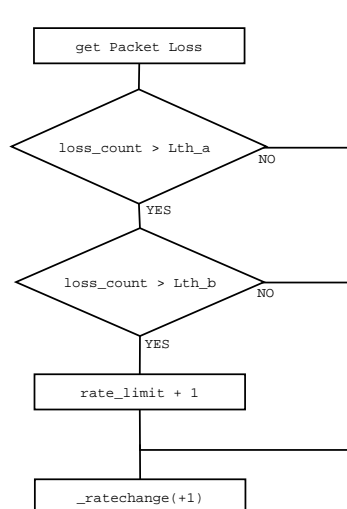


図 8.7. Rate-up by Packetloss

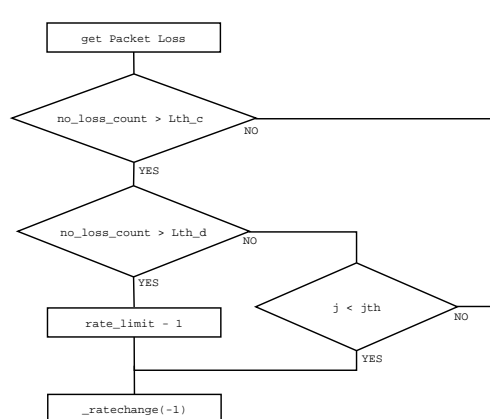


図 8.8. Rate-down by Jitter/Loss

8.6 評価

複数種類の機器での動作

本機構の動作を確認した機器の組み合わせを表 8.1 に示す。

フレームレートコントロールの実現

DFCS を搭載した DVTS の挙動を確認し、フレームレートコントロールが実現できたかどうか評価す

表 8.1. 動作を確認した機器の組み合わせ

| 送信側 | | 受信側 | |
|-----|----------|-----|----------|
| 種類 | 機種 | 種類 | 機種 |
| CAM | MV-EX21 | M/C | ADVC-100 |
| M/C | DVMC-DA1 | M/C | DVMC-DA2 |

る。本評価は、以下に挙げる実験を行い、フレーム間引き率の変動やネットワークにかかるトラフィック量の変化を確認することで行う。

- 帯域幅の不足
- 遅延時間の変化
- DV ストリーム 2 本での協調動作
- DV ストリームと TCP ストリームの協調動作

8.7 まとめ・今後

本研究では、ジッタやパケットロスといったエンドノード間で取得可能な伝送特性のみを用いた映像・音声配信のための新しい輻輳制御手法を提案し、その実証例として DVTS に対して本手法を適用した。

本研究により、中継ルータなどの特別な機器を必要とせず、ネットワークの状態に適応させた映像・

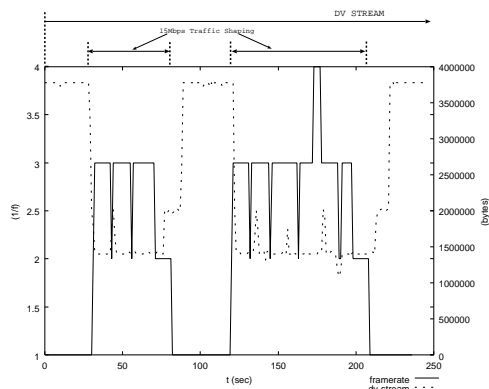


図 8.9. Result of Eval.1
(Decreasing Bandwidth)

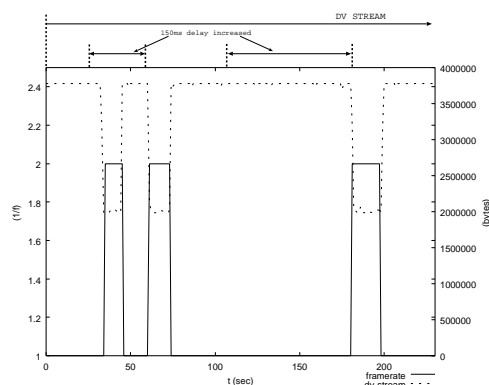


図 8.10. Result of Eval.2
(Increasing DelayTime)

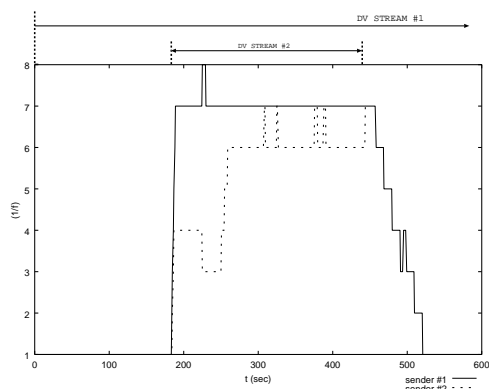


図 8.11. Result of Eval.3
(DV with DV:Framerate)

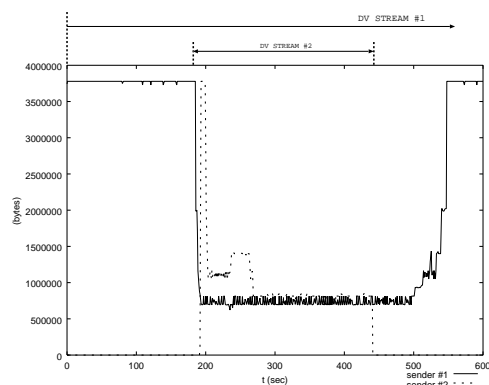


図 8.12. Result of Eval.3
(DV with DV:Traffic)

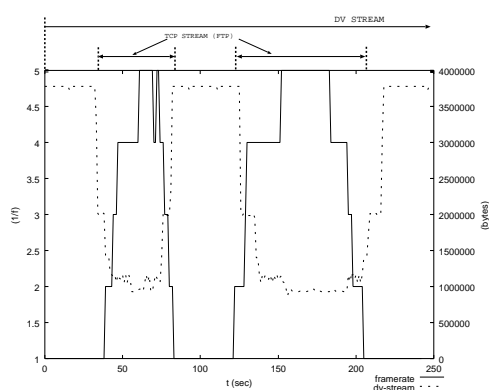


図 8.13. Result of Eval.4
(DV with TCP:Framerate)

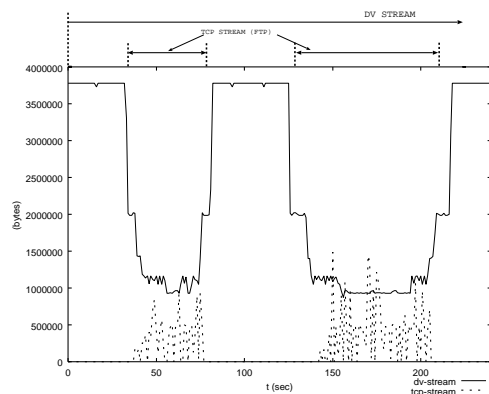


図 8.14. Result of Eval.4
(DV with TCP:Traffic)

音声配信が可能となった。この手法を応用することにより、DVTS 以外の映像・音声配信システムに対してもネットワーク状態に適応した配信を行う機構を実現できる。

今後は、フレームレート制御の精度向上や安定化

のためのアルゴリズム改良、フレーム間引き率以外のレート制御パラメータへの対応、EmbeddedDVTS への組み込みと拡張、DVTS へのマージなどを行う必要がある。