

第 XXVII 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27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

伝送最適化モデルを提案し、開発を行っている。

第1章 10G ワーキンググループの概要

10G ワーキンググループ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。本章ではまず本年度の方向性を解説し、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[125]、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。

本年度は主に遅延をキーワードに二つの観点からアプローチを行った。一つは帯域を潤沢に使い、低遅延、低ジッタな高品質サービスの提供である。もう一つは遅延時間が発生する環境の上でのコンテンツ共同制作環境の構築である。

1.2 前年度の活動

本ワーキンググループでは2003年度の活動として主に以下の6つの項目に対して取組みを行った。

超広帯域ネットワークの有効利用方法として1) 帯域を大量消費することによる有効利用、2) 大量消費はしないが品質の高い通信を保証、という2点が考えられる。前年度は主に以下の活動を行った。

- 計算機資源に応じた映像配信機構の研究
帯域を大量消費することによる有効利用を狙ったシステムとして開発に取り組んでいる。計算機資源に応じたストリームを受信ノードに転送することにより、効率よく多くの受信ノードが情報取得できることを目指した機構である。
- パケットフロー特性に着目した映像音声配信モデルの研究
品質の高い通信を保証するためのシステムとして開発に取り組んでいる。ネットワーク伝送遅延の揺らぎは長時間発生するものではなく、一時的なものが殆どであることに着目し、パケット

1.3 本年度の活動

前年度の活動を踏まえ、今年度は以下の活動を行った。

- JGNII シンポジウム 2005
慶応大学湘南藤沢キャンパス・大阪間、および University of Washington・大阪間において非圧縮 HDTV による高精細映像伝送の実証実験を行った。
- インターネットメトロノームを用いた遠隔協調演奏環境の実現
インターネットを用いた複数地点間での映画、音楽の共同制作を行うには、伝送メディア長、途中経路などによるデータ到達の差異を考慮する必要がある。ネットワーク遅延時間を演者が知覚し、リズム管理という形で共同制作可能な環境の構築を行っている。本機構を用い、愛・地球博のクロージングフォーラムにおいてジャズセッション実験を行った。

第2章 JGNII シンポジウム 2005

2.1 概要

JGNII シンポジウム 2005 では、村井純氏、Larry Smarr 氏による講演を遠隔から行った。その際に非圧縮 HDTV を用いたストリーミングアプリケーションを用いて、遠隔講演を支援した。村井純氏の講演は慶應義塾大学湘南藤沢キャンパスより行われ、Larry Smarr 氏の講演は University of Washington より行われた。プログラム概要は以下の通りである。

- 開催期間：1月17日(月)～19日(水)
- 場所：大阪国際会議場
- 村井純氏の講演：「JGNII：グローバル R&D への貢献と責任」
- Larry Smarr 氏の講演：「Using OptIPuter Innovations to Enable LambdaGrid

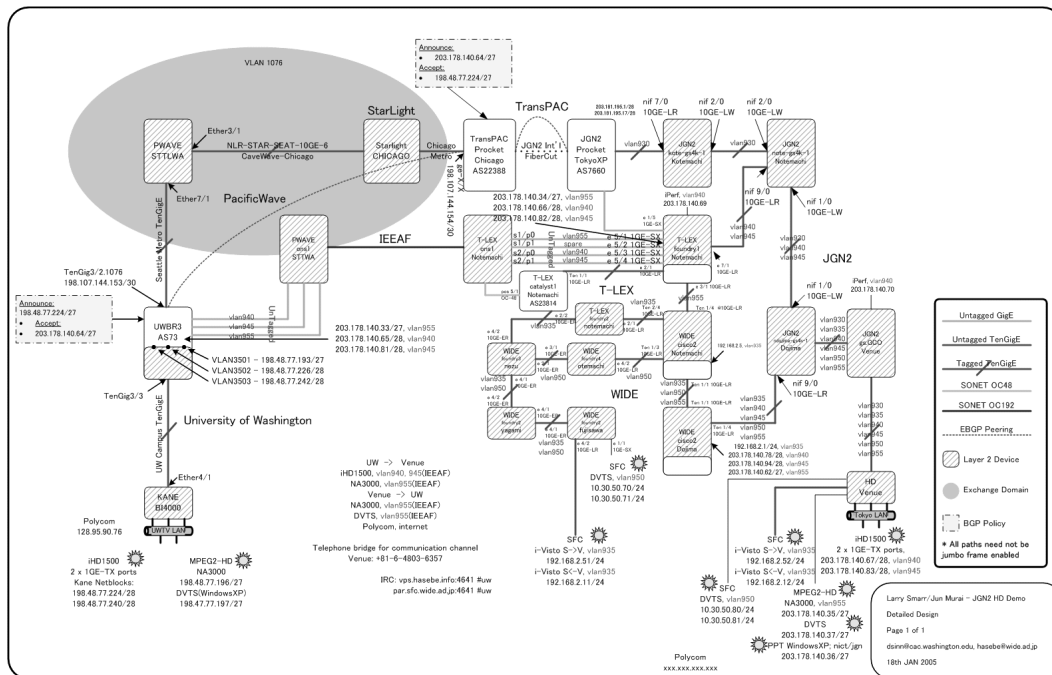


図 2.1. トポロジ図

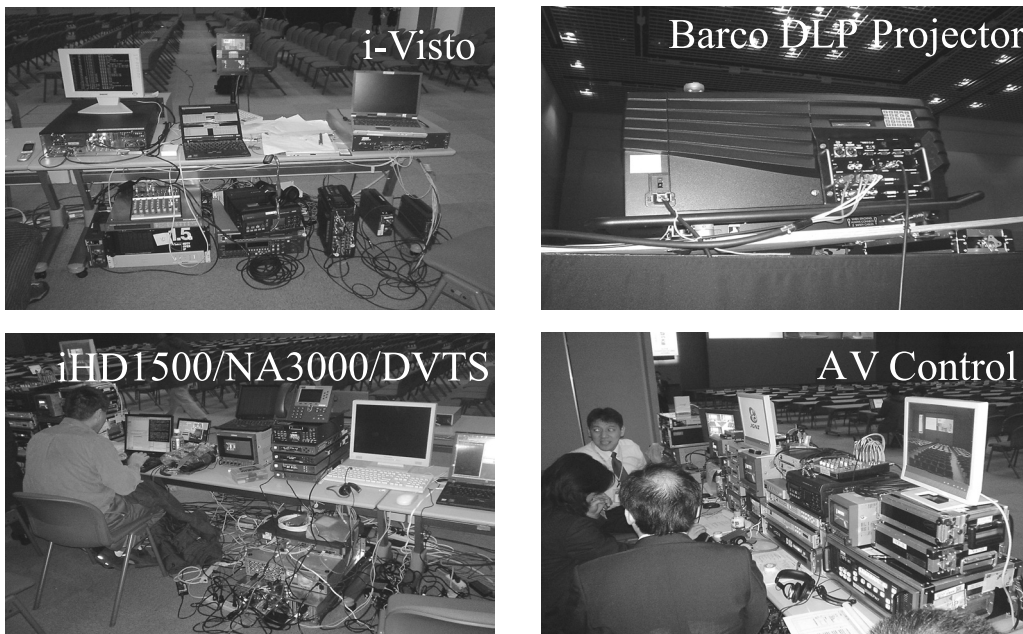


図 2.2. 大阪会場の様子

Applications」
 図 2.1 にトポロジ図を示す。図に示された AS PATH を経由し、慶応大学湘南藤沢キャンパス・大阪間および、University of Washington・大阪間の通信において、非圧縮 HDTV 双方向の通信を実現した。

また、図 2.2・2.3 に大阪・UW における機材を示

す。大阪-SFC 間では NTT 研究所の i-Visto[334] を利用し、大阪-UW 間では iHD1500 を利用した。各アプリケーション共に安定したストリーミングを供給し、非圧縮 HDTV による高精細かつ低遅延な映像伝送を実証した。

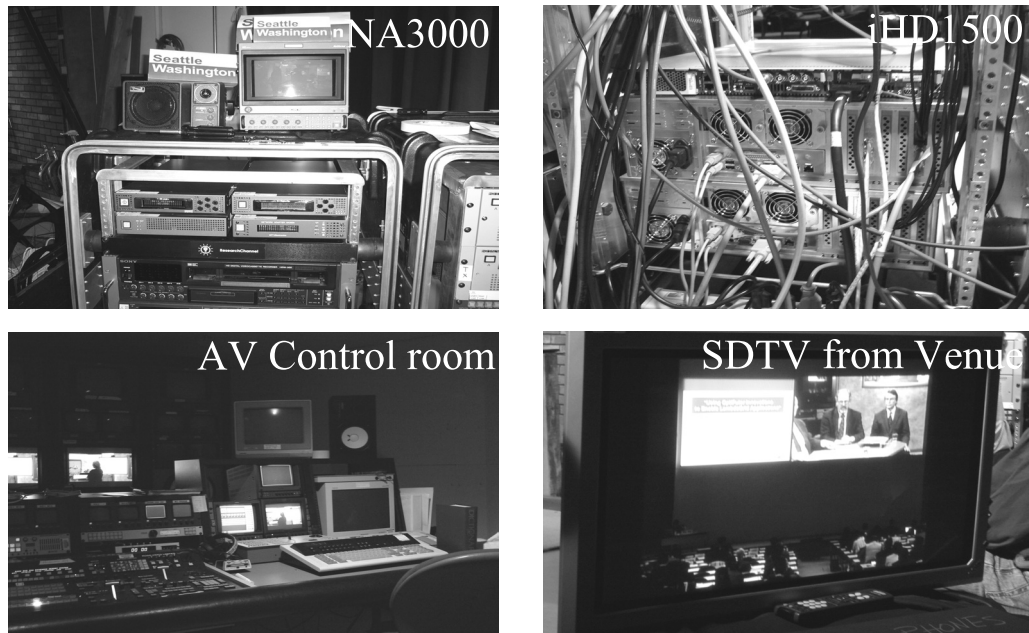


図 2.3. UW 会場の様子

第3章 インターネット・メトロノームを用いた遠隔協調演奏環境の実現

インターネットを用いた複数地点間での映画、音楽の共同制作をリアルタイムに行うには、伝送メディア長、途中経路などによる、データ到達時間の差異を考慮しなければならない。

本研究では、異なる2地点間における遠隔協調演奏を実現する上で問題となる、リズム管理手法に着目した。2地点間のネットワーク遅延を演奏者が知覚するためのアプリケーションとして、「インターネット・メトロノーム」の設計・実装を行った。本実装をもとに、「愛・地球博」のクロージングフォーラムにおいて、日蘭間でのジャズセッションによる実証実験を行った。

3.1 はじめに

インターネットを用いた映画、音楽の遠隔地点間における共同制作に関する関心が高まっている。その中で、音楽の共同演奏は、今まで同じ地点にいなければ行うことは出来なかった。

「愛・地球博 [315]」のクロージングフォーラムにおいて、日本・愛知の万博会場とオランダのアムステルダムをインターネット回線で接続し、ジャズの

セッションを行った。

3.1.1 問題点

インターネットを用いた遠隔地点間での音楽協調演奏をリアルタイムで行う上で、インターネットによる伝送遅延は無視することができない。近年のインターネット技術の発展により、光の早さに近い速度で情報の転送を行うことが可能になった。しかし、光の早さを超えて情報の転送を行うことは現状では不可能である。たとえば、光の早さを $300,000 \text{ km/sec}$ とし、地球の外周を $40,000 \text{ km}$ とした場合、ある地点から地球の反対側まで光が到達するのにかかる時間を求めると、式 (4) のようになる。

$$20,000/300,000 = 1/15 \text{ sec} \approx 67 \text{ ms} \quad (4)$$

すなわち、地球の反対側との通信における伝送遅延時間は、 67 ms より短くすることができない。 67 ms という時間は、人間同士のコミュニケーションにおいてはそう大きな遅延とならない。しかし、音楽の協調演奏においては、とても大きな遅延として演奏に影響が生じてしまう。

他に、遅延が発生する要因として、映像と音声のエンコード、デコードにかかる処理時間が挙げられる。この問題に対しては、今回の実験では、映像・音声の転送に非圧縮 HD 転送を用いることによって極小に抑えることとした。

3.1.2 目的

このように、伝送遅延を回避できない状況において、音楽の協調演奏を行うためには、互いに音楽を演奏するための拍子を共有する必要がある。システムイメージ図を図 3.1 に示す。実時間が異なっても、音楽を演奏するための時間軸を共有することによって、一つの音楽を違う地点から、伝送にかかる遅延時間をこえて演奏することができる。

本研究では、伝送遅延にあわせて時間軸を共有することのできるような機構を設計・実装し、この機構を用いることによって、離れた 2 地点間において、インターネットを用いて音楽を協調演奏することのできる環境を実現する。

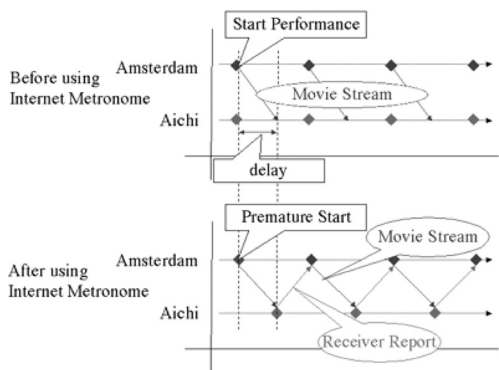


図 3.1. システムイメージ図

3.2 設計

本研究では、3.1.2 項で述べた目的を達成するためのアプリケーションとして、インターネット・メトロノームを実装する。その上で、本節では、3.1.1 項で述べた問題点を解消するための機構の設計について述べる。

3.2.1 遅延時間計測

インターネット上で発生する伝送遅延時間を計測するために、ICMP のような挙動をとるパケットを送信する。片側からパケットの送出を開始し、パケットを受信した後、パケットを相手側に送信する。前回パケットを送出してからパケットを受け取るまでにかかった時間の半分の時間が、片道の伝送遅延時間であると推測される。この際、送信するパケットは、他のトラフィックに影響を与えないように、出来る限り小さいサイズのものとする。

3.2.2 ジッタと遅延時間の変動を考慮した遅延時間算出

遅延時間は常にほぼ一定であるが、ネットワークの状況など様々な要因によって細かなジッタが発生する。ジッタは細かければ映像転送・再生には影響がないが、音楽を演奏するための拍子を不安定なものとしてしまう。

拍子を刻む上で、このジッタが影響を及ぼさないように、伝送遅延時間の平均を求めようとした。これを、本機構では伝送遅延時間として採用している。伝送遅延時間の平均は式 (5) で求めている。(図 3.2)

$$\text{average (ms)} = (\text{current_delay_time} + \text{previous_average})/2 \quad (5)$$

しかし、これだけでは突発的にジッタが大きくなった際や、ネットワークの遅延が大きく変動した際に、正確に遅延にあわせて拍子を刻むことができなくなってしまふ。そこで、平均から $m\%$ 外れた遅延時間を計測した場合には、平均遅延時間にその値を採用せず、 n 回外れた値を計測した際に、その値を新しい平均遅延時間として採用する。(図 3.3)

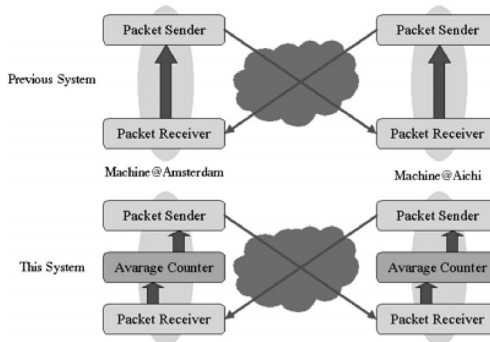


図 3.2. 平均遅延時間の算出

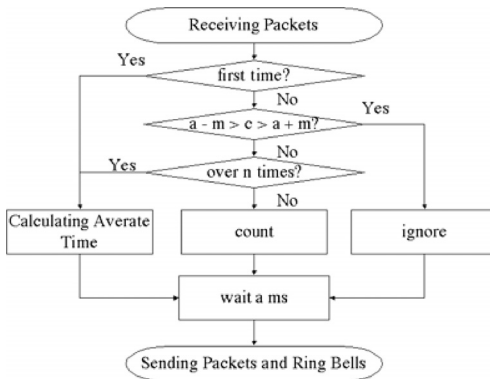


図 3.3. 平均遅延時間の算出方法

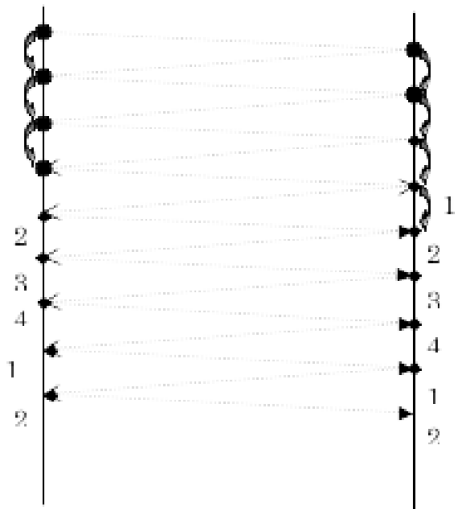


図 3.4. 本機構におけるパケットの挙動および拍子の再生

本機構では、最初にパケットの受信を 5 回行い、その平均遅延時間を算出した後、拍子音の再生・表示を行う。それまでの間は、パケットの送信は、相手側からのパケットを受信した直後に行う。また、拍子音の再生・表示の開始後は、パケットの送信は、拍子音の再生・表示と同時に進行。これは、拍子音の再生・表示の間隔に揺らぎが発生しないようにするためである。(図 3.4)

3.2.3 拍子音の再生・表示

伝送遅延時間をもとにした拍子を演奏者に伝える方法として、短い音の再生とアイコンの点滅を行う。

本機構では、拍子音の再生とアイコンの点滅は、パケットを相手側に送信してから、算出された平均遅延時間経過後に、音の再生とアイコンの点滅を行う。拍子音の再生とアイコンの点滅は、本機構の他の処理とは非同期に行われる。

3.3 実装

本節では、本機構の実装の詳細について述べる。

本機構の実装環境を表 3.1 に示す。本機構は、ダイアログベースの Windows アプリケーションとして実装した。(図 3.5)

表 3.1. 実装環境

OS	Microsoft WindowsXP
Language	Visual C++
Developing environment	Visual Studio .NET

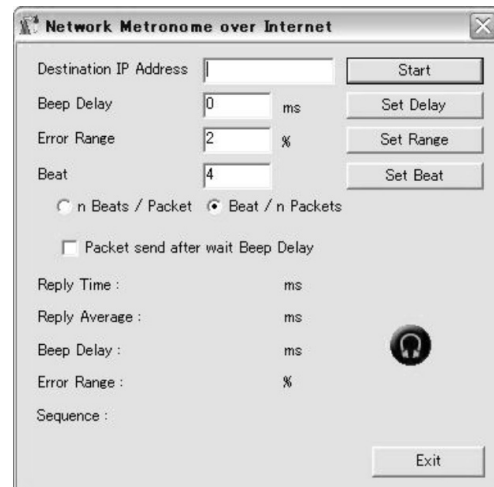


図 3.5. インターネット・メトロノーム

3.3.1 CmetroDlg class

この class では、ダイアログの初期化を行う。また、メトロノームの開始処理、および終了処理を行う。

開始処理の際に、入力されたパラメータの取得、サーバ・クライアントの判定、UDP ソケットの作成を行う。この際、WSAAsyncSelect() 関数を用いて、ソケットからの read、write のシグナルを Window Message として受け取れるようにする。これにより、ソケットからの read を非同期に行うことが出来る。クライアントであった場合、宛先 IP アドレスにパケットを送信する。

サーバ・クライアントの判定は以下のように行う。

- サーバ

Destination IP Address フィールドが空白。

- クライアント

Destination IP Address フィールドに IP アドレスが入力されている。

終了処理の際には、UDP ソケットの close を行う。

今回、Window Message は WMLREAD、WMLPING、WMLPONG、WM_TIMER の 4 つを使う。WMLPING、WMLPONG は、自分で定義したメッセージである。

3.3.2 WM_READ

ソケットに相手側からのパケットが届くと呼ばれる。

この Window Message を受け取ると、ソケットから受け取ったパケットを read する。同時に、遅延時間の計測、平均遅延時間の計算を行う。最初の 5 回は、ここから WMLPONG Message を自 Window に送信する。

3.3.3 WM_PING

自ホストがクライアントであった場合、最初のパケットを送信するときに呼ばれる。

ここでは、拍子音の再生と、アイコンの点滅を行う。ここでの拍子音のみ、他とは違う音声ファイルが再生される。

3.3.4 WM_PONG

5 回目にパケットを受け取るまで、WM_READ から呼ばれる。

ここでは、相手側へのパケットの送信、拍子音の再生、アイコンの点滅、タイマの設定、ログファイルへの出力を行う。

タイマは、SetTimer() 関数を用いる。タイマには、平均遅延時間を設定する。ログファイルへは、以下のようなフォーマットで出力する。

- Sequence, 年月日時分秒, Reply Time, Reply Average, Beep Delay, Error Range, Beat, Beat Setting

3.3.5 WM_TIMER

SetTimer() 関数によって指定された間隔ごとに、呼び出される。

ここでは、相手側へのパケットの送信、拍子音の再生、アイコンの点滅、タイマの設定、ログファイルへの出力を行う。

タイマは、この Window Message が呼び出されるたびに終了し、現時点での平均遅延時間を用いて、再設定する。

3.4 実証実験

愛知万博会場（愛知県名古屋市）とオランダアムステルダム SARA の双方に演奏者が存在する状態で遠隔ジャズセッションを行った。本イベントのイメージ図を図 3.6 に示す。演奏映像は NTT 研究所が開発した非圧縮 HDTV 動画転送システムである i-Visto[334] を用いて行われた。

このジャズセッションは日欧間の通信となるため、その伝送遅延時間は 120–130 ms 程度となる。そのため 10 GbE 対応の dummy net[227] 環境を構築し、数 ms から数 1000 ms までの遅延時間をさらに加え、演奏者間の遅延時間制御を行った。これにより、伝送遅延と dummy net による遅延と、音楽における小節などの区切りを同期させた。これに加え、遅延

Global view of This event

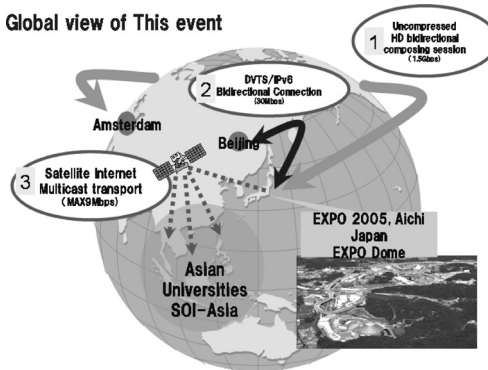


図 3.6. 「愛・地球博」イベントの概要図



図 3.7. イベント会場の様子

時間を基本としたインターネットメトロノームを用い、一拍や一小節などの音楽で用いられるタイミングを制御することにより、遠隔ジャズセッションを実現した。

3.5 今後の課題

現状で考えられる、本実装の問題点を以下に挙げる。

- インターフェイスの改善
現状でのインターフェイスでは、設定が直感的に分かりにくい。また、音楽的な表現を用いていないので、利用者にわかりづらい。
- 設定の自動化
現状では、平均遅延時間算出のためのしきい値の設定など、実験を重ね、最適値をあらかじめ設定しておく必要がある。また、ネットワーク以外の機器において発生する、エンコードおよびデコードにおける遅延を自動的に測定し、遅延時間に追加して拍子を刻む必要がある。
- 音楽表現への変換の自動化
現状では、遅延時間の取得のみの機能しかないので、音楽のテンポと拍子にあった遅延時間を生

成できるように、dummy net と協調して、遅延の表現を出来るようにする必要がある。

