RMCP：遠隔音楽制御用プロトコルを中心とした音楽情報処理

後藤真孝[†,☆] 根山亮[†] 村岡洋一[†]

本論文では，シンボル化された音楽情報をネットワークを介して共有するための通信プロトコルRMCPについて述べる．本研究は，音楽情報処理システムを分散実装したり，ネットワークを利用したアプリケーションを実現する際に有効な，音楽情報処理のためのネットワークプロトコルを設計することを目的とする．そのような目的では，音楽情報の効率の良い共有が望ましいが，従来の音楽情報用の関連プロトコルの多くは1対1通信を基本としたコネクション型であり，複数プロセス間での効率的な情報共有は十分考慮されていなかった．RMCPはコネクションレス型であり，全通信をブロードキャストで行うため，各プロセスへ個別に送信するオーバヘッドがなく情報共有の効率が良い．さらにRMCPは，リアルタイム音楽情報処理のために，タイムスタンプを用いた時間管理の機能を提供し，遠隔地間の合奏のために，信頼性を確保しながら遠隔地間で双方向にパケットを中継する機能も提供している．本論文では，これらの機能を活用することで実現できる，遅延を考慮した新たな形態の遠隔地間の合奏も提案する．RMCPはすでに様々な音楽情報処理システムを実現するために運用されてきた．その経験から，RMCPの通信遅延時間が十分小さいことが確認されただけでなく，RMCPを用いることで必要な機能が再利用できて実装が容易になり，拡張性が高くなることも確認された．

RMCP: Musical Information Processing Based on Remote Music Control Protocol

Masataka Goto,[†,☆] Ryo Neyama[†] and Yoichi Muraoka[†]

This paper describes a communication protocol, called RMCP (Remote Music Control Protocol), which is designed for sharing symbolic musical information through computer networks. The purpose of this research is to design a network protocol which is suitable for musical information processing and facilitates distributed implementation of music-related applications. Although efficient musical information sharing is desirable for such a purpose, most previous music-related protocols were connection-oriented and did not emphasize efficient information sharing among multiple processes. Since the RMCP is a connection-less protocol, it supports broadcast-based efficient information sharing without the overhead of multiple transmission. It also supports time scheduling using time stamps for real-time musical information processing and reliable bidirectional packet relay for remote sessions. This paper also proposes an innovative remote session over the Internet that has a long delay. RMCP has been utilized in various applications and we found that the communication delay of RMCP was enough small and RMCP facilitated system implementation and expansion because of good reusability.

1. はじめに

MIDI (Musical Instrument Digital Interface)[1]情報などのシンボル化された音楽情報を通信するためのネットワークプロトコルは，音楽情報処理システムを分散実装するためだけでなく，ネットワークを活用した新たな音楽アプリケーションを実現するためにも重要になってきている．たとえば，LAN (Local Area Network) やWAN (Wide Area Network) などの計算機ネットワーク上でMIDI情報を通信することで，ネットワーク経由のライブMIDI中継や遠隔地間の合奏，ネットワーク上の多様な計算機によって支援（演奏に関連した情報のコンピュータグラフィックスによる視覚化など）された合奏，人間とネットワーク上に分散した複数の計算機間の即興演奏といった様々なアプリケーションが可能になる．さらに，音楽アプリケーションを分散実装することは，負荷分散以外にも，異なる計算機に接続されている多様な機器を同時に活用できるといった利点を持つ．これらのアプリケーションのためには，ネットワークを経由した効率の良い音

† 早稲田大学理工学部
School of Science and Engineering, Waseda University
☆ 現在，電子技術総合研究所
Presently with Electrotechnical Laboratory

楽情報の共有が重要となる．

従来の MIDI に基づいたネットワークプロトコル[2)~6)]の多くはコネクション型（connection-oriented）であり，複数の分散したプロセス間で情報共有する際の遅延時間を小さくおさえることは，十分考慮されていなかった．また，MIDI だけを用いた場合，同じ情報を共有するために MIDI 機器間で全対全通信を実現するのは効率が悪かった．さらに，MIDI は通信のバンド幅が低く制限されており，局所的な通信のために設計されていた．MIDI に基づかない演奏情報の通信プロトコル[7)]も提案されているが，その多くは特殊な機器の使用を前提としていた．

そこで本研究では，計算機ネットワークを介してシンボル化された音楽情報を共有するために，音楽用通信プロトコル RMCP（Remote Music Control Protocol）を設計した．RMCP は，Ethernet[8)]などの計算機ネットワークと MIDI を融合した分散協調システムにおける，コネクションレス型（connection-less）の通信プロトコルである．RMCP では，再送（各プロセスへの転送）のオーバヘッドがないブロードキャストを利用して通信するため，複数プロセス間での情報共有の効率が良い．さらに，ネットワークを活用した音楽情報処理システムを実現する際に望まれる機能として，タイムスタンプを用いた時間管理や，遠隔地間の信頼性を確保した双方向パケット中継の機能を提供している．

以下，2 章において RMCP がサーバ・クライアント・モデルに基づいていることを述べ，サーバ・クライアントの設計指針と基本的な機能を紹介する．そして，3 章において RMCP の具体的な実装を説明し，時間管理の実現方法と信頼性を確保した遅延付きパケット中継の方法を提案する．RMCP プログラミングライブラリは C 言語上と Java 言語上に実装され，すでに様々な OS 上で RMCP が運用されてきた．そこで 4 章では具体的な運用例として，ネットワークセッション，遠隔地間の合奏，仮想ジャズセッションシステムなどの RMCP を用いた様々なアプリケーションを紹介する．最後に 5 章でまとめを述べる．

2. RMCP の概要

RMCP では，様々な音楽情報は RMCP パケットという単位で伝送され，分散した計算機上の複数のプロセスによって共有される．たとえば，MIDI note on/off などの各 MIDI メッセージは，1 つの RMCP パケットの中に包まれた後に，LAN を経由して他のプロセスへと送信される．

RMCP はマルチサーバ・マルチクライアント・モデルに基づいている．複数のサーバが様々なクライアントからの要求を受信して処理するモデルである．そのため，RMCP に関するすべての計算機プロセスは，**RMCP クライアント**と **RMCP サーバ**に分類される．

各 RMCP クライアントは，MIDI 機器（MIDI IN）から得た MIDI メッセージや，ユーザ（演奏者など）のインタラクション内容に対応するメッセージなどを含む RMCP パケットを生成する．そして，これをすべての RMCP サーバへ向けてブロードキャストする．このブロードキャストはコネクションレス型の片方向の送信であるため，サーバからクライアントへの返答（受取通知パケット）はない．

一方，各 RMCP サーバは，ブロードキャストされたすべてのパケットを受信し，様々な方法で利用する．たとえば，MIDI 機器（MIDI OUT）を制御して音を鳴らしたり，音楽情報を視覚化したり，音楽に反応するコンピュータグラフィックスを生成したりする．このように，クライアントから一度ブロードキャストされたパケットの内容は，すべてのサーバが共有して同時に活用できるため，複数回送信するオーバヘッドがなく，効率が良い．さらに，通信量を増やすことなく，演奏の視覚化などの多様なサーバを追加することもできる．

2.1 設計指針

RMCP サーバ・クライアントを設計する際の指針として，RMCP では，実現したい機能を，再利用が可能なようにできるだけ小さなプロセスに分けて実装する．この設計指針により，以下の 3 つのメリットが得られる．

- 各サーバ・クライアントはそれぞれに特化した処理に専念すればよいため，実装が容易になる．
- サーバを単に追加するだけで新たな機能を実現できるので，拡張性が高くなる．
- 効果的に負荷分散できるように，各サーバ・クライアントを異なる計算機上に割り当てることが容易になる．

2.2 基本的なサーバ・クライアント

2.1 節の指針に基づいて設計された，基本的な RMCP サーバと RMCP クライアントの一覧を**表 1** に示し，これらの運用例を**図 1** に示す．図中には各サーバ・クライアントを 1 つずつしか示していないが，実際にはそれぞれが複数あってもよい．

ユーザは，MIDI 楽器と RMCP MIDI Receiver か，計算機のキーボード・マウスと RMCP MIDI Station

表1 基本的なRMCPサーバとRMCPクライアント
Table 1 The basic RMCP servers and RMCP clients.

RMCPサーバ	機能
RMCP Sound Server	受信パケット中のMIDIメッセージをMIDI機器へ送信（主に聴覚化）
RMCP Display Server	受信パケット中のMIDIメッセージを画面上の鍵盤の色を変えて視覚化
RMCP Animation Server	受信パケットを用いて音楽に反応するコンピュータグラフィックスを生成
RMCP Packet Recorder	全受信パケットを受信時刻とともに「RMCPパケット記録ファイル」へ記録

RMCPクライアント	機能
RMCP MIDI Receiver	MIDI楽器を用いて演奏（MIDIメッセージをMIDI機器から受信）
RMCP MIDI Station	MIDI楽器の代わりに計算機のキーボードとマウスを用いて演奏
RMCP SMF Player	標準MIDIファイル（SMF）を再生
RMCP Packet Player	「RMCPパケット記録ファイル」を再生

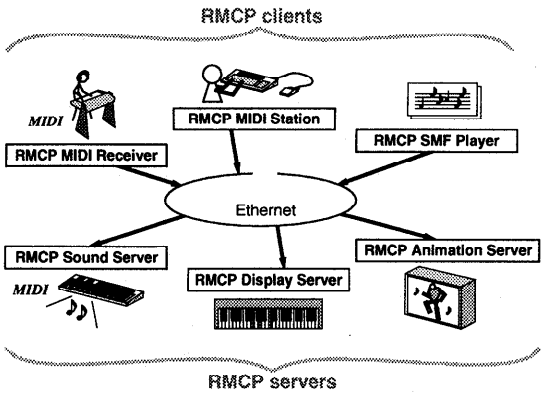


図1 RMCPサーバとRMCPクライアントの運用例
Fig. 1 An example of using RMCP servers and RMCP clients.

を用いて演奏する．演奏手段としてMIDI楽器を用いる場合には，MIDI楽器から送られてくるMIDIメッセージをRMCP MIDI Receiverが受信し，これをRMCPパケットとしてブロードキャストする．一方，演奏手段として計算機のキーボードとマウスを用いる場合には，RMCP MIDI StationがMIDIメッセージを生成し，RMCPパケットとしてブロードキャストする．この場合，ユーザは各キーが鍵盤に割り当てられたキーボードを押すか，画面上の鍵盤をマウスでクリックして演奏する．また，RMCP SMF Playerにより，標準MIDIファイル（SMF: Standard MIDI File）を再生することもできる．

こうしてブロードキャストされたパケットがRMCP Sound Serverに受信されると，パケットに含まれるMIDIメッセージがMIDI楽器へ送られて，実際の演奏音として出力される．これらのパケットはRMCP Display Serverにも受信され，画面上の鍵盤の色を変えることで視覚化される．さらにRMCP Animation Serverにも同時に受信され，その内容に反応して動作する仮想のダンサーや演奏者などのリアルタイムコンピュータグラフィックスが生成される．

RMCP Packet RecorderとRMCP Packet Playerは，ネットワーク上のRMCPパケットの状況を記録・再現するために用いる．RMCP Packet Recorderは全RMCPパケットを，その受信時刻とともに「RMCPパケット記録ファイル」へ記録する．一方，RMCP Packet Playerは，そのファイル中のパケットを，受信時刻どおりの適切なタイミングでブロードキャストする．これらは，RMCPに基づいた音楽情報処理システムにおいて，ユーザの演奏やインタラクションの状況，システムの出力などを保存・再現したいときに便利である．

3. RMCPの実装

通信プロトコルRMCPを設計し，それに基づいてRMCPプログラミングライブラリと様々なサーバ・クライアントを実装した．RMCPは，下位レイヤーとしてUDP/IPを用いたコネクションレス型のプロトコルである．Ethernet LANなどの十分高速で信頼性の高いネットワークで運用されることを前提として，RMCPレイヤーでは基本的にブロードキャストで通信し，その信頼性は確保しない．これは，情報共有する際の遅延時間を小さくおさえるためである．ただし3.3節で述べるように，InternetのようなWANで運用する際には，そもそも遅延時間が大きいので，信頼性を確保した通信を提供する．

3.1 RMCPパケット

RMCPレイヤーのパケットをRMCPパケットと呼び，これはRMCPクライアントからRMCPサーバへ片方向に送信（ブロードキャスト）される．RMCPパケットはヘッダー（RMCP Header）とそれに続くメッセージ本体（RMCP Message Body）からなる（図2）．ヘッダーには，メッセージの種類（type of message），サーバ番号，クライアント識別子（ユーザ識別子），タイムスタンプ情報，中継禁止フラグ（flag），

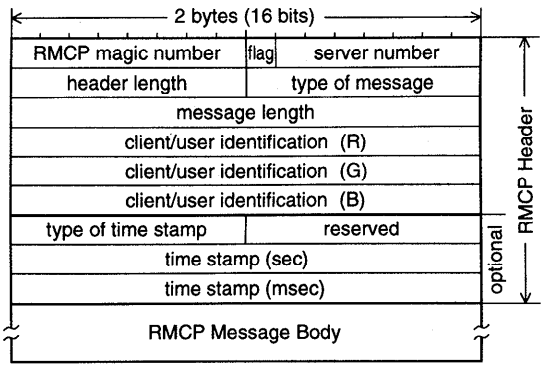


図2 RMCPパケットの構造
Fig. 2 The RMCP packet format.

ヘッダー長，メッセージ長などが含まれている．

主要なメッセージの種類には，以下のものがある．

- MIDI情報
  MIDIメッセージを伝送するために用いる．メッセージ本体がそのままMIDIメッセージ（複数のMIDIメッセージでもよい）となっている．
- ビート情報
  テンポの同期のためにビートの時刻などを伝送する際に用いる．
- コード情報
  コード名やボイシングなどを伝送するために用いる．曲の調の指定も伝送することができる．
- アニメーション情報
  コンピュータグラフィックスを生成・制御するために用いる．
- ジェスチャー情報
  ジェスチャー認識の結果などを伝送するために用いる．

ほかにもサーバを終了させるメッセージや，3.2節で述べる時刻同期のためのメッセージ，メッセージ本体が規定されていない拡張用メッセージなどがある．

サーバ番号はサーバを区別するための7ビットの識別子である．RMCPサーバは，起動時に自分のサーバ番号を設定でき，同じサーバ番号を持つものどうしでグループを作ることができる．RMCPクライアントは，ブロードキャストするときにどのサーバ番号（グループ）に対する要求かを指定する．このとき，全サーバという指定もできる．複数の音楽アプリケーションが同一ネットワーク上でRMCPを利用するときに，それぞれがサーバ番号を変えて運用すれば，お互いに独立に実行できる．

クライアント識別子は，RGBそれぞれ2バイトずつの計6バイトのカラーコードで表現される．特にRMCP Display ServerがMIDI情報を画面表示する際に，このカラーコードを用いて鍵盤の色を変えることで，その演奏がどのクライアントによるものかをユーザが判断できる．また，これはユーザ識別子として用いることも可能であり，各ユーザが自分固有のカラーコードを定義すれば，複数のユーザが合奏しているときなどにユーザ間の区別ができる．

タイムスタンプ情報については3.2節において，中継禁止フラグについては3.3節において説明する．ヘッダー長，メッセージ長は，それぞれの長さをバイト単位で示したものである．

3.2 時間管理

RMCPサーバがRMCPクライアントの指定する時刻にパケットを処理できるように，RMCPではタイムスタンプ☆を用いた時間管理機能を提供する．RMCPパケットには，それが処理されるべき時刻を示すタイムスタンプを付与されたパケットと，タイムスタンプなしパケットの2種類がある．これによりRMCPサーバは，タイムスタンプの時刻前に到着した前者のパケットを，その時刻が来るまでバッファリングして時刻どおりに処理することができ，それらの時間順も保証できる．タイムスタンプの時刻後に到着したパケットやタイムスタンプなしパケットは，基本的にサーバに到着した直後にその場で処理される．

たとえば，RMCP SMF PlayerがSMFを再生する際には，SMF中のMIDIメッセージをタイムスタンプ付きパケットとして，数秒前にブロードキャストしておく．これにより，ネットワークの遅延時間が揺らいでも，RMCP Sound Serverによる演奏音は影響を受けずに適切なタイミングとなる．また，事前に演奏情報が分かることで，RMCP Animation Serverが表示する仮想の演奏者が，「予備動作を行ってから実際の発音タイミングで楽器を弾く」といった動作をすることも可能になる．

3.2.1 時刻同期サーバ

タイムスタンプによる時間管理は，複数の計算機の間で時刻が同期していることを前提としている．同じタイムスタンプのパケットは，異なる計算機上のRMCPサーバによって，同一時刻に処理される必要があるからである．

そこで，同期を必要とする各計算機上に1つのRMCP Time Synchronization Server（時刻同期サーバ）を導入する．時刻同期サーバによって，あたかも異なる計算機の内部時計が同期しているかのように，

☆ 現在の実装では1 ms単位の絶対時刻表現のみに対応している．

RMCP サーバが受信パケットのタイムスタンプを扱うことができる☆．この時刻同期に関する処理はすべてプログラミングライブラリ内に隠蔽され，RMCP の上位レイヤーからはすでに同期した時刻上のタイムスタンプしか見えないため，RMCP サーバの開発者は時刻同期を意識する必要がない．

各時刻同期サーバは，自分の計算機の内部時計と他のそれぞれの計算機の内部時計との時刻差の表（オフセットテーブル）を作成する．そして，このオフセットテーブルに送信時刻（現在の内部時計の時刻）のタイムスタンプを付けて，定期的にブロードキャストする．ただし，RMCP パケットのトラフィックが多いときはブロードキャストしない．つまり極力アプリケーションによってネットワークが利用されていないときに時刻差を測定する．これは，アプリケーション動作中に余計なトラフィックが発生するのを防ぐだけでなく，より適切な時刻差を得ることが期待できるという効果もある．

一方，時刻同期サーバが他の計算機上の時刻同期サーバからこのテーブルを受信すると，自分が所持するテーブル中の関連する時刻差の項目を更新する．具体的には，自分の内部時計で計測したテーブルの受信時刻と，そのテーブルのタイムスタンプに書かれた送信時刻との差を計算し，テーブル送信者（具体的にはその IP アドレス）に対応する項目を更新する．ただし，LAN の遅延時間は十分小さく無視できると仮定している．

各 RMCP サーバは，同じ計算機上の時刻同期サーバのみからオフセットテーブルを受信する．そして，タイムスタンプ付きのパケットを受信するたびに，その送信者の時刻差の項目をテーブル中から見つけ，その時刻差を加えてタイムスタンプを補正する．こうすることで，実際には内部時計がずれているにもかかわらず，見かけ上はすべてが同期しているかのようにタイムスタンプを扱うことができる．また，RMCP クライアントの側は時刻差について知る必要がないため，実装が簡潔になる利点も持つ．

3.2.2 時刻同期サーバの運用例

時刻同期サーバの運用例を図 3 に示す．ある時点における 3 台の計算機 Host A，B，C の内部時計が，右下に示された 1000，1050，980（時間単位は省略する）であり，時刻同期サーバによって図中に示したオフセットテーブルが作成されたとする．Host A の RMCP クライアントが，時刻 1200 のタイムスタンプを付けたパケットをブロードキャストすると，各計算機上の RMCP サーバは，オフセットテーブルから Host A の IP アドレスの項目を探し，タイムスタンプにその時刻差を加えて補正する．その後，Host B がこの時刻から 100 後の時刻 1350 のタイムスタンプを付けたパケットをブロードキャストすると，同様に他の計算機上で適切に補正される．

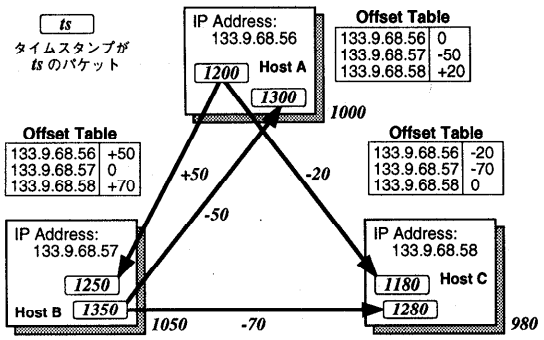


図 3 オフセットテーブルによるタイムスタンプの補正例
Fig. 3 An example of adjusting time stamps using offset tables.

実際に Host A の時刻が 1200 になったときには，すべての計算機の時刻が 1 つ目のパケットのタイムスタンプの時刻と等しくなり，同時に処理される．それから 100 後にも同様に 2 つ目が同時に処理される．このように，内部時計があたかも同期しているかのように，各 RMCP サーバはタイムスタンプを処理できる．

3.3 信頼性を確保したパケット中継

RMCP を Internet などの WAN 上で運用するために，WAN で結ばれた 2 つの LAN 間で双方向にパケットを中継する（共有する）RMCP Gateway を導入する．RMCP Gateway は，下位レイヤーとして TCP/IP を用いたコネクション型の通信を行い，信頼性を確保してパケットを中継する．

遠隔地にある 2 つの LAN を結ぶときに，まず双方で RMCP Gateway を実行してコネクションを確立する☆☆（図 4）．各 RMCP Gateway は，自分の LAN 上にブロードキャストされた RMCP パケットを，他方の RMCP Gateway へ送信する．逆に，他方から中継されてきたパケットを受信すると，自分の LAN 上にブロードキャストする．ただし，一度中継したパ

☆ 時刻同期サーバを用いる代わりに，NTP (Network Time Protocol)[9] などの他の方法によって，事前に計算機間で内部時刻を合わせておいてもよい．ただし，一般に内部時計の補正は管理者特権が必要である．一方，我々が提案する手法は内部時計を直接補正するわけではないので，管理者特権は必要ない．

☆☆ 複数の LAN 間で RMCP パケットを共有する場合，各 LAN 間で RMCP Gateway によるコネクションを確立すればよい．

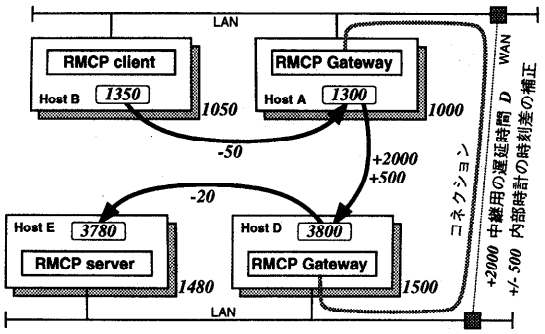


図4 RMCP Gateway による遅延付きパケット中継
Fig. 4 Packet relay with a delay using RMCP Gateways.

ケットは再び中継されることがないように，ヘッダー中の中継禁止フラグをセットしておく．これにより中継の無限ループを回避する．

RMCP サーバ・クライアントは，RMCP Gateway によって結ばれた 2 つの LAN が，あたかも同一のネットワークであるかのように通信できる．遠隔地間用に新たに開発し直す必要はない．ただし，ユーザは WAN 経由の通信の遅延時間が，LAN 上の通信よりも長いことを考慮する必要がある．

遠隔地間で通信する際には，ネットワークの遅延時間を回避することは不可能である．光速でさえ地球を半周するのに約 66 ms もかかってしまう．したがって，RMCP Gateway による中継では，遅延時間を小さくするよりも，変動（ジッタ）が非常に小さい一定の遅延時間を提供する方が重要だと我々は考える．そこで，ネットワークの遅延時間よりも十分に大きい遅延時間 D を中継用に設定する．受信側では，中継されてきたパケットをバッファリングし，D の分だけ正確に遅らせてブロードキャストすることで，ネットワークに起因する変動の影響を受けないようにする．

RMCP Gateway は，LAN から受信したパケットを中継する際に，パケット中のタイムスタンプとは別に，その受信時刻もタイムスタンプとして付加する．他方の RMCP Gateway は，両者のタイムスタンプに D を加えたうえに，両端の計算機の内部時計の時刻差も補正する（図 4）．そして，それを送信すべき時刻が来るまでバッファリングして，時刻どおりにブロードキャストする．適切な D を設定し，内部時計の時刻差を補正するために，RMCP Gateway はコネクション確立直後に，ネットワークの遅延時間と内部時計の時刻差をパケットを往復させて計測する．計測方法は NTP[9] の手法に基づいている．

以上の遅延付き双方向パケット中継により，Internet を介したライブ MIDI 中継だけでなく，4.3 節で述べる新たな形態の遠隔地間の合奏 RemoteGIG も可能になる．

3.4 実験結果

RMCP プログラミングライブラリを C 言語上と Java 言語上☆にそれぞれ実装し，IRIX-5.3, IRIX-6.2, IRIX-6.3, IRIX-6.4, Solaris-2.5, SunOS-4.1.3, HP-UX, Linux-2.0, Windows-95, Windows-NT などの様々な OS 上で RMCP を運用した☆☆．現在の実装では，IRIX OS 上の RMCP サーバは 2 ms，他の OS 上の RMCP サーバは 10 ms の分解能でタイムスタンプを処理する．

RMCP は実際に様々な分散音楽情報処理システムを実現するために運用されてきた．その経験から，RMCP を用いることで必要な機能が再利用できて実装が容易になり，拡張性も高くなることを確認した．特に，リアルタイムシステムやインタラクティブシステムの実現に効果的であった．また，一度 RMCP のライブラリを用いて MIDI 関連のソースコードを書けば，実装時に MIDI 関連 API の OS 依存の違いに悩まされることがなくなって移植性が高くなり，しかもネットワーク透過なプログラムになるという点も，高く評価された．

LAN 上の RMCP パケットの遅延時間の測定結果を示して考察する．異なる計算機上にあるクライアントからサーバへ RMCP パケットが到達するまでの遅延時間を，MIDI note on メッセージを 1 万回伝送して測定した．実験には C 言語上の RMCP ライブラリを用い，10 Mbps Ethernet 上の 2 台の SGI Indigo2 (IRIX-5.3) 間で測定した．ただし，測定時の LAN と計算機の負荷は小さかった．その結果，遅延時間は平均値 0.30 ms，標準偏差 0.06 ms，最小値 0.28 ms，最大値 1.24 ms であった．これから，RMCP は速度的に MIDI (31.25 Kbits/sec) に遜色なく情報伝送できることが分かった☆☆☆．また，パケットロスや到着順の入れ換えなどの問題も発生しなかった．

さらに，RMCP Gateway を用いた遠隔地間のパケット中継も実施した．4.3 節で詳細は述べるが，東京（日

☆ Java アプレット上で安全性を確保しながら RMCP を利用する方法も提案されている[10),11)]．

☆☆ OS によっては，複数のプロセスが同一 UDP ポートからパケットを受信できないため，パケット分配用プロセスを用いる必要がある．

☆☆☆ MIDI で 1 音分の演奏情報 (3 bytes) を伝送するのに 0.96 ms かかり，MIDI 楽器内部では 10 ms 以上も遅延する場合があることを考慮すると，RMCP の遅延時間は十分小さいといえる．

本）とニューヨーク（北米）間☆でInternet経由のパケット中継を達成でき，遠隔地間の合奏（RemoteGIG）が実際に行えることを確認した．

4. アプリケーション

1992年以来，RMCPに基づいた様々なアプリケーションが実現されてきた．以下，これらの中から主要なものを紹介する．なお，表1以外のRMCPサーバ・クライアントは，基本的にアプリケーション固有のものである．

4.1 ネットワークセッション

RMCPは設計当初，複数の演奏者がLANを介して合奏するために用いられた[12),13)]．各演奏者は，単に他の演奏者の音を聞けるだけでなく，より協調して演奏しやすいように，視覚化された他の演奏者の演奏状態も見ることができる．

本アプリケーションでは，RMCP MIDI ReceiverあるいはRMCP MIDI Station，RMCP Sound Server，RMCP Display Serverを利用する（図5）．RMCP SMF Playerによる伴奏に合わせて合奏してもよい．また，合奏を録音・再生するためにRMCP Packet RecorderとRMCP Packet Playerを用いることもできる．

4.2 Improvision

Improvisionとは，鍵盤演奏ができない人でも計算機のマウスをクリック・ドラッグするだけで即興演奏できる，音楽演奏用インタフェースである[13)]．ただし，無調性でリズムのない音楽を対象としている．基本的にマウスの方向で音高が，マウスの移動速度で音量が決まる．さらに画面上を流れる色も演奏に応じて変化し，マウスの方向で色が，マウスの移動速度で明るさが変わるなど，聴覚と視覚をともに用いて他の演奏者と合奏ができる．

各演奏者は，ネットワークセッションにおけるRMCP MIDI StationのかわりにRMCP Improvisionを用いて合奏でき，RMCP Display ServerのかわりにRMCP Improvision Display Serverを用いて，他の演奏者の演奏状態を見ることができる．

4.3 遅延を考慮した遠隔地間の合奏：RemoteGIG

Internetのような遅延の大きいWANを使ってリアルタイムに合奏するのは困難なため，従来は演奏を片方向へ送信するライブMIDI中継が主流であった．遠隔地間の合奏に関しては，WAN（ISDN）を用いた実施例[4),14)]も報告されているが，3.3節で述べたように遠隔地間では遅延時間の影響は不可避である．そこで本研究では，従来の1カ所で合奏をするモデルを遠隔地間に持ち込むのではなく，遅延を前提とした遠隔地間の合奏用のモデルを提案する．

RemoteGIGとは，Internetなどで不可避な遅延を積極的に利用した，新たな形態の遠隔地間の合奏である．RemoteGIGでは，同一のコード進行（12小節のブルース進行など）の繰返しを，テンポ一定で演奏することを前提とする．遠隔地にいる2人の演奏者が合奏する様子を図6に示す．演奏者はお互いの演奏を，コード進行のちょうど1周期分の時間だけ遅れて聞き合うことで合奏する．コード進行は繰り返すため，演奏が1周期分遅れれば再び同じコードとなり調和することが期待できる．このとき，演奏者は1カ所で合奏

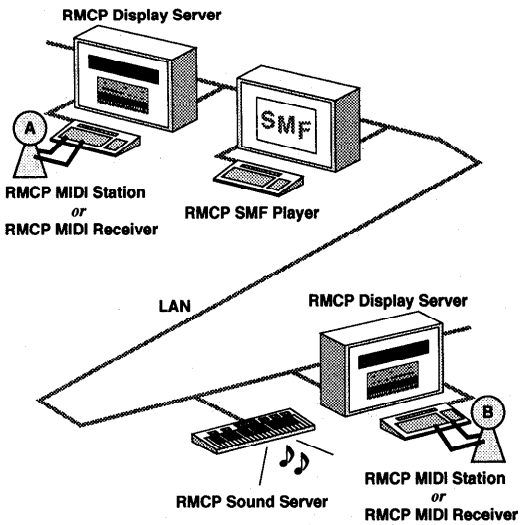


図5 ネットワークセッション
Fig. 5 Networked session.

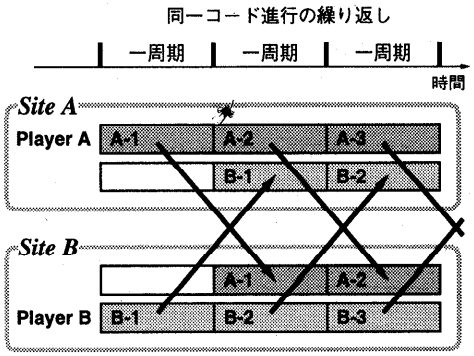


図6 RemoteGIG
Fig. 6 RemoteGIG.

☆ NTT ICC（InterCommunication Center）とコロンビア大学間，早稲田大学とコロンビア大学間などで実施した．

するときと同様なインタラクションを試みるのではなく，新たなインタラクションを探求するとよい．なお，この合奏のモデルは無調性音楽などにも適用できる．

RemoteGIG では，4.1 節の RMCP サーバ・クライアントに加え，1 組の RMCP Gateway を必要とする．その際，3.3 節の D を 1 周期分の時間の倍数に設定する．特に，一定のテンポを維持するために，RMCP SMF Player で伴奏用のドラムスの演奏を流すとよい．

RemoteGIG の有効性を実証するために，日米インターカレッジ・コンピュータ音楽フェスティバルの一環として，日本時間 1997 年 12 月 14 日午前 11 時より，コンサート形式での公開実験を実施した．東京の NTT ICC とニューヨークのコロンビア大学との間で，専用回線ではなく通常の Internet を経由して実験を行った☆．それぞれのサイトでは SGI O2 (IRIX-6.3) を 1 台占有使用し，RMCP Gateway, RMCP Sound Server, RMCP MIDI Receiver などを動作させた．RMCP Gateway による中継用遅延時間 D は，両サイト間のネットワークの遅延時間（約 0.25 秒）に比べて十分大きい値である 30.72 秒に設定した．そして，コード進行の 1 周期が 30.72 秒となるテンポで，伴奏用のベースとドラムスが含まれた SMF ファイルを再生しながら即興演奏を行った．その結果，知覚できるテンポの揺らぎやパケットロスは発生せず，実際に遠隔地間で RemoteGIG が行えることを確認した．

4.4 仮想ジャズセッションシステム：VirJa Session

VirJa Session (Virtual Jazz Session System) とは，人間の演奏者と計算機演奏者が，お互いの演奏を聞き合いながら対等な立場で即興演奏するだけでなく，お互いの姿やジェスチャーも見ながらインタラクションできるセッションシステムである[15)~17)]．現実のセッションにより近い全演奏者間のマルチモーダルインタラクションを実現したことで，従来の音だけのセッションシステムに比べ，より臨場感のあるセッションが達成された．現在の実装では，ジャズのピアノトリオを対象とし，人間がピアニスト，計算機がベーシストとドラマーを担当する（図 7）．

このシステムは，RMCP MIDI Receiver, RMCP Sound Server のほかに，カメラ画像からジェスチャーを認識する RMCP Camera Analyzer, 各計算機演奏者の演奏を担当する RMCP Player Server, その姿を 3 次元 CG（コンピュータグラフィックス）で表示

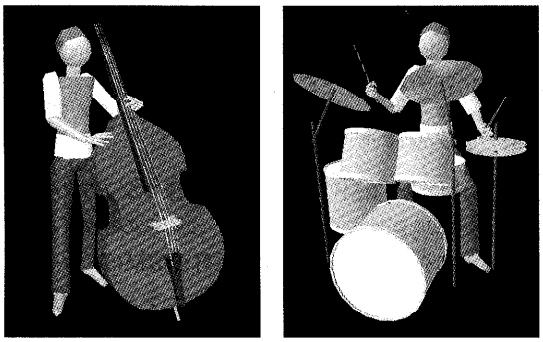


図 7 VirJa Session
Fig. 7 VirJa Session.

する RMCP Animation Server, テンポを管理する RMCP Beat Provider で構成された．このように分散実装したことで，開発の分担がしやすく実装が容易になっただけでなく，拡張性も高くなった．

4.5 仮想のダンサー “Cindy”

RMCP を応用して，音楽に踊らされる CG ダンサーによるインタラクティブパフォーマンスシステムが実現された[18)]．このシステムでは，2 人の演奏者が自分たちの即興演奏する音楽によって，仮想のダンサー “Cindy”☆☆の踊りをリアルタイムに振付けできる（図 8）．2 人は通常のジャムセッションのように即興演奏するだけでなく，異なる役割で振付けを分担しながら，協調してダンサーを踊らせようとする．つまり，音楽的にも映像的にも協調して演奏しようとする．こうして聴覚と視覚によるマルチモーダルインタラクションを達成することで，CG ダンサーは単に音楽に付随したものでなく，2 人の演奏者にとっての新たな表現手段となる．

このシステムは，RMCP MIDI Receiver, RMCP Sound Server のほかに，Cindy を表示する RMCP Animation Server, 演奏を踊りに反映させるために分析する RMCP Music Analyzer で構成された．

4.6 そ の 他

CG 上の仮想犬と，音楽も含めた 3 種類のインタラクションができるシステム[23)]を実現する際にも，RMCP が用いられた．また，仮想空間中の仮想キャラクタの動きを複数の仮想カメラで同時に撮影するように CG 画像が生成できる，分散 CG アニメーションシステム VirStA System (Virtual Stage and Actors System)[24)~26)]の実装においても，RMCP は有効であった．これらのアプリケーションにおいては，RMCP が

☆ NTT ICC では Rick Bassett が演奏し，コロンビア大学では Brad Garton と Perry Cook が演奏した．

☆☆ ここで紹介したものとは別の応用例になるが，Cindy は受動的に踊らされるだけでなく，ビートトラッキングシステム[19)~22)]により音楽に合わせて踊ることもできる．

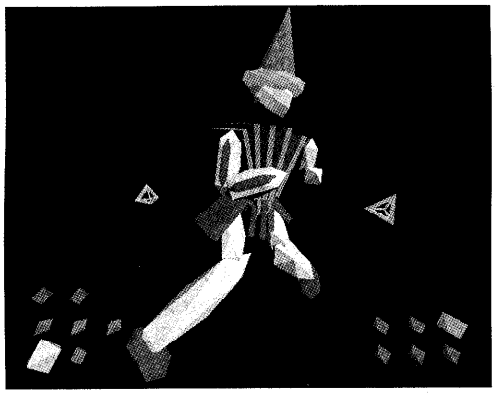


図8 仮想のダンサー “Cindy”
Fig. 8 A virtual dancer “Cindy”.

音楽情報だけでなくアニメーション情報やジェスチャー情報も扱えることが効果的であった．

さらに，ジャズらしいコードにリハーモナイズするシステム「ハービー君」[27]や，ネットワーク経由で相互作用するアルゴリズム作曲システム[28]の実装にもRMCPが用いられた．

5. おわりに

本論文では，複数の分散したプロセス間で，MIDIのようなシンボル化された音楽情報を共有するためのネットワークプロトコルRMCPについて述べた．RMCPでは，タイムスタンプを用いた時間管理とそのための時刻同期の機能により，音楽やCGに関連した情報をリアルタイムに扱うことを可能にした．また，運用するネットワークの性質・信頼性に応じて下位レイヤーを選択して通信することで，分散音楽情報処理に適した効率が良く実用的な情報共有を実現した．具体的には，LAN内の通信にはUDP/IPを用いてブロードキャストによるオーバヘッドの小さい情報共有を実現する一方，Internet経由などの遠隔地間の通信にはTCP/IPを用いることで，信頼性を確保した情報共有を実現した．RMCPはすでに複数のOS上に実装され，様々な目的のために運用されてきた．そして，実用的な応用例の開発を通じて，その有効性も確認されている．

RMCPソフトウェアパッケージは，すでにWWWページ上で公開され（WWWサーチエンジンなどで，“RMCP” と “Masataka Goto” をともに含むページを検索するとアクセスできる），様々な研究機関で使われ始めている．今後は，より多くのユーザにとって使いやすくなるようにRMCPサーバ・クライアントを拡充しながら，さらなる応用例を実現していく予定である．

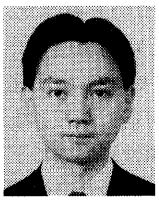
謝辞 RMCPの研究を始めるきっかけを与えていただいたSALA（Science Art Laboratory）に感謝する．特に研究の初期の段階において多大なご協力・ご指導をいただいた，元SALAメンバの橋本裕司氏，千葉滋氏，宮川晋氏に感謝する．また，本研究に対し有益なご助言・ご協力をいただいた菊地淑晃氏，日高伊佐夫氏，阿部哲也氏，松本英明氏，黒田洋介氏，鷲坂光一氏，平田圭二氏，長嶋洋一氏に感謝する．最後に，RemoteGIGの実験にご協力いただいたRick Bassett氏，Brad Garton氏，Perry Cook氏，莱孝之氏，松田周氏に感謝する．

参考文献

1) MIDI規格協議会規格検討委員会：MIDI 1.0規格（Document Ver.4.1日本語版），MIDI規格協議会 (1989).
2) Aoyagi, T. and Hirata, K.: Music Server System – Distributed Music System on Local Area Network, *Journal of Information Processing*, Vol.15, No.1, pp.1–9 (1992).
3) 森 光彦：音楽情報処理システムの構築とネットワーク技術の活用，日本音響学会音楽音響研究会，MA92-3, Vol.11, No.1, pp.21–26 (1992).
4) Nielsen, O.: MIDI and Audio via ISDN, *Proc. 1994 International Computer Music Conference*, pp.451–454 (1994).
5) Fober, D.: Real-time Midi data flow on Ethernet and the software architecture of MidiShare, *Proc. 1994 International Computer Music Conference*, pp.447–450 (1994).
6) Fober, D., Letz, S. and Orlarey, Y.: Recent Developments of MidiShare, *Proc. 1996 International Computer Music Conference*, pp.40–42 (1996).
7) McMillen, K., Simon, D. and Wright, M.: A Summary of the ZIPI Network, *Computer Music Journal*, Vol.18, No.4, pp.74–80 (1994).
8) Metcalfe, R.M. and Boggs, D.R.: Ethernet: Distributed Packet Switching for Local Computer Networks, *Comm. ACM*, Vol.19, No.7, pp.395–404 (1976).
9) Mills, D.L.: Network Time Protocol (Version 3) Specification, Implementation and Analysis, RFC-1305 (1992).
10) 根山 亮，後藤真孝，村岡洋一：WWW上で公開可能な分散音楽情報処理システム—ダウンロードしたJavaアプレットのための安全なローカル通信方法，日本ソフトウェア科学会第14回大会，B4-1 (1997).
11) 根山 亮，後藤真孝，村岡洋一：WWW上の異なる計算機環境で動作する音楽セッションシステ

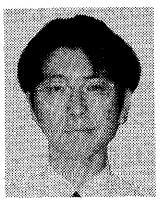
ム—VirJa Session on WWW へ向けて，第54回情報処理学会全国大会論文集，7J-05 (1997).
12) 後藤真孝：MIDI 制御のための分散協調システム，jus 設立10周年記念 UNIX 国際シンポジウム論文集，pp.161–171 (1992).
13) 後藤真孝，橋本裕司：MIDI 制御のための分散協調システム—遠隔地間の合奏を目指して，情報処理学会研究報告，音楽情報科学 93-MUS-4-1, Vol.93, No.109, pp.1–8 (1993).
14) 高山 明，千葉雅之，三木恵造，首藤一彦：遠隔同時演奏システム，日本音響学会音楽音響研究会，MA89-10, Vol.8, No.3, pp.23–28 (1989).
15) 後藤真孝，日高伊佐夫，松本英明，黒田洋介，村岡洋一：すべてのプレーヤーが対等なジャズセッションシステム I. システムの全体構想と分散環境での実装，情報処理学会研究報告，音楽情報科学 96-MUS-14-4, Vol.96, No.19, pp.21–28 (1996).
16) 日高伊佐夫，後藤真孝，村岡洋一：すべてのプレーヤーが対等なジャズセッションシステム II. ベーシストとドラマーの実現，情報処理学会研究報告，音楽情報科学 96-MUS-14-5, Vol.96, No.19, pp.29–36 (1996).
17) Goto, M., Hidaka, I., Matsumoto, H., Kuroda, Y. and Muraoka, Y.: A Jazz Session System for Interplay among All Players – VirJa Session, *Proc. 1996 International Computer Music Conference*, pp.346–349 (1996).
18) 後藤真孝，村岡洋一：音楽に踊らされる CG ダンサーによるインタラクティブパフォーマンス，コンピュータソフトウェア，Vol.14, No.3, pp.20–29 (1997).
19) Goto, M. and Muraoka, Y.: A Beat Tracking System for Acoustic Signals of Music, *Proc. 2nd ACM International Conference on Multimedia*, pp.365–372 (1994).
20) 後藤真孝，村岡洋一：ビートトラッキングシステムの並列計算機への実装—AP1000 によるリアルタイム音楽情報処理，情報処理学会論文誌，Vol.37, No.7, pp.1460–1468 (1996).
21) 後藤真孝，村岡洋一：音響信号を対象としたリアルタイムビートトラッキングシステム—コード変化検出による打楽器音を含まない音楽への対応，電子情報通信学会論文誌（D-II），Vol.J81-D-II, No.2, pp.227–237 (1998).
22) Goto, M. and Muraoka, Y.: An Audio-based Real-time Beat Tracking System and Its Applications, *Proc. 1998 International Computer Music Conference*, pp.17–20 (1998).
23) 阿部哲也，後藤真孝，黒田洋介，村岡洋一：インタラクティブドッグ～仮想犬との3種類のインタラクション，インタラクティブシステムとソフトウェア III，pp.19–28, 近代科学社 (1995).
24) 後藤真孝，阿部哲也，松本英明，村岡洋一：VirStA System：仮想ステージと仮想アクターによる分散 CG アニメーションシステム I. システムの全体構想，第53回情報処理学会全国大会論文集，1P-05 (1996).
25) 阿部哲也，後藤真孝，松本英明，村岡洋一：VirStA System：仮想ステージと仮想アクターによる分散 CG アニメーションシステム II. 分散環境でのリアルタイム実装，第53回情報処理学会全国大会論文集，1P-06 (1996).
26) 松本英明，後藤真孝，阿部哲也，村岡洋一：VirStA System：仮想ステージと仮想アクターによる分散 CG アニメーションシステム III. ジャズセッションプレーヤーの実現，第53回情報処理学会全国大会論文集，1P-07 (1996).
27) 後藤真孝，平田圭二：ハービー君：演繹オブジェクト指向に基づいてジャズらしいコードにリハーモナイズするシステム，情報処理学会研究報告，音楽情報科学 96-MUS-16-6, Vol.96, No.75, pp.33–38 (1996).
28) 長嶋洋一，中村文隆，後藤真孝，片寄晴弘，井口征士：ネットワーク上で相互作用するアルゴリズム作曲系を用いた音楽教育システム，第54回情報処理学会全国大会論文集，7J-04 (1997).

(平成10年 3 月26日受付)
(平成10年11月 9 日採録)



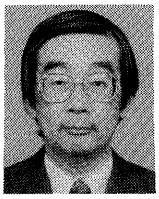
後藤 真孝（正会員）

1993年早稲田大学理工学部電子通信学科卒業．1998年同大大学院博士後期課程修了．同年，電子技術総合研究所に入所し，現在に至る．博士（工学）．音楽情報処理，マルチモーダルインタラクション等に興味を持つ．1992年 jus 設立10周年記念 UNIX 国際シンポジウム論文賞受賞．1993年 NICOGRAPH '93 CG 教育シンポジウム最優秀賞受賞．1997年情報処理学会山下記念研究賞受賞．電子情報通信学会，日本音楽知覚認知学会，ICMA 各会員．



根山 亮（学生会員）

1997年早稲田大学理工学部情報学科卒業．現在，同大大学院修士課程在学中．音楽情報処理，分散並列コンピューティング等に興味を持つ．日本ソフトウェア科学会会員．



村岡 洋一（正会員）

1965年早稲田大学理工学部電気通信学科卒業．1971年イリノイ大学電子計算機学科博士課程修了．Ph.D.この間，Illiac-IVプロジェクトで並列処理ソフトウェアの研究に従事．同学科助手ののち，日本電信電話公社（現NTT）電気通信研究所に入所．1985年より早稲田大学理工学部教授．現在同大学メディアネットワークセンター所長．並列処理，マンマシンインタフェース等に興味を持つ．「コンピュータアーキテクチャ」(近代科学社)等著書多数．