

## 不特定多数による遅延を考慮した遠隔セッションシステム

後藤 真孝

電子技術総合研究所  
goto@et1.go.jp

根山 亮

早稲田大学 理工学部  
ryo@muraoka.info.waseda.ac.jp

あらし 本稿では、Internetのような広域ネットワークを経由して、不特定多数のユーザが遠隔地間でセッションできるシステム Open RemoteGIG を提案する。従来、占有使用回線を介した MIDI 中継は実現されていたが、遅延の大きい Internetなどを介した双方向の MIDI 中継によって、調性とリズムのある音楽を合奏することはできなかった。我々の Open RemoteGIG は遅延を活用した新たな合奏モデルに基づいており、複数のユーザが互いの演奏を聞き合いながらセッションすることを可能にする。さらに、演奏相手の発見環境や打ち合せ用のチャット機能も提供する。MIDI 中継は音楽用通信プロトコル RMCP に基づいて実装され、実際に遠隔地間でセッションをできることが確認された。

## An Open-to-the-Public Remote Session System Overcoming Network Latency

Masataka Goto

Electrotechnical Laboratory  
1-1-4 Umezono, Tsukuba, Ibaraki 305-8568 Japan

Ryo Neyama

School of Sci. and Eng., Waseda Univ.  
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555 Japan

**Abstract** This paper proposes a remote session system, called *Open RemoteGIG*, which enables many users to join a jam session via the Internet. Although MIDI transmission via an exclusive network has been achieved, it was considered impossible to achieve a remote session with tonality and rhythm via a long-latency network like the Internet. Our Open RemoteGIG, which is based on an innovative session model overcoming the long latency, enables multiple users to jam together while listening to others' performances. It also supports a user-finding and group-chatting environment. The MIDI transmission was implemented by using a music network protocol called *RMCP* and we confirmed that the proposed remote session was achieved.

### 1 はじめに

遠隔地間でジャムセッションや合奏をおこなうことができれば、物理的に異なる場所にいる相手と、一カ所に集まらなくても音楽的なインタラクションを楽しむことができる。さらに、実世界では困難な、意識のない複数の演奏者と国境を越えたセッションも可能となる。このような遠隔地間の人間同志のコミュニケーションは、Internetのような広域ネットワークやネットワークコンピューティングの普及に伴い、専用回線や専用装置を用いずに実現できる環境が整いつつある。本研究は、このような遠隔セッションを可能にするシステムを実現することで、新たな形態の遠隔地間の合奏や、音楽における新たなインタラクションを探求することを目的とする。

遠隔セッションを実現するには双方向の演奏情報の中継が不可欠となるが、不特定多数が利用するInternetのようなネットワークを介した遠隔セッションは、従来実現されていなかった。衛星中継等を利用し

て演奏中の互いの音と映像を直接伝送する試みは以前からあるが、基本的に遅延時間が長く、伝送量が多い上に情報の再利用・加工がしにくいという欠点があった。そのため、MIDI信号のような電子楽器の制御情報を、ネットワークを通じて中継する試みがおこなわれてきた。しかし、双方向のMIDI中継をISDN等の占有使用できる回線を使って実施した事例<sup>1),2)</sup>では、伝送が短距離で遅延時間が小さいことを前提としており、長距離になると同時性が失われて実用上問題があることが指摘されていた<sup>1)</sup>。Internetのような遅延時間の長いネットワークでも同様の問題があり、実用的には企業等によって片方向のMIDI中継しか試みられていなかった。このように、調性とリズムのある音楽を遠隔地間でセッションすることは、遅延時間の長い状況下では困難と考えられており、Internetを経由した遠隔地間の合奏(双方向のMIDI中継)は実現されていなかった。

このような遠隔セッションを可能にするために、我々は文献3),4)において、RemoteGIGを提案した。

RemoteGIG は、Internet 等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セッションである。RemoteGIG では、同一のコード進行 (12 小節のブルース進行など) の繰り返しを、テンポ一定で演奏することを前提とし、遠隔地にいる二人の演奏者が即興でネットワーク経由のセッションをおこなう。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インタラクションをおこなう。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。

本稿では、この RemoteGIG を不特定多数のユーザが演奏できるように拡張した遠隔セッションシステム **Open RemoteGIG** を提案する。Open RemoteGIG では、遠隔地にいる全ユーザの演奏情報が、Internet 等を経由したスター状のネットワーク接続によって、すべて中央に集められる。そして、自分以外の演奏情報が、コード進行の 1 周期の整数倍の時間だけ遅延して送り返されてくることで、全ユーザが互いの演奏を聞き合いながらセッションできる。Open RemoteGIG ではさらに、演奏相手を発見するための環境や、演奏内容の打ち合せ等を可能にするチャット機能も提供する。なお、ユーザは演奏をおこなわずに、他人の演奏を観客として鑑賞してもよい。これを活用すれば、Internet 上での Open RemoteGIG の公開ライブパフォーマンスも実施できる。

以下、2 において通信プロトコル RMCP を用いた RemoteGIG の実現方法を述べ、東京とニューヨーク間で Internet を介して実施した実験コンサートの結果を報告する。次に、3 において本研究が提案する Open RemoteGIG の機能を紹介し、システムの構成と具体的な実装方法を説明する。本実装では、演奏情報の中継機能は C 言語で実装されているが、ユーザは WWW ブラウザを用いて Open RemoteGIG の WWW ページにアクセスすることで、遠隔セッションに参加することができる。最後に 4 でまとめと今後の研究の方向性を述べる。

## 2 遅延を考慮した遠隔セッション: RemoteGIG

RemoteGIG は、遠隔地間で遅延が大きい状況を考慮した、新たな演奏のモデルに基づくセッションである。演奏者が一箇所に集まってセッションをおこなうときの伝統的なモデルは、他者の演奏音を小さい遅延時間で聞くことができ、それに反応して演奏する自分の演奏音も小さい遅延時間で相手に届くことを前提としている。しかし、光速でさえ地球を半周す

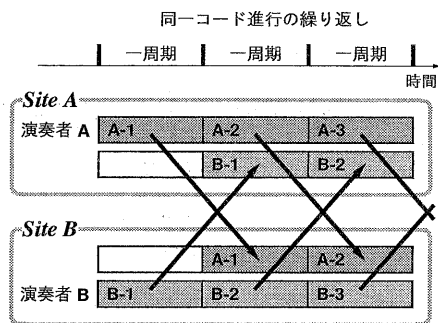


図 1: RemoteGIG のモデル

るのに約 66 ms もかかってしまうことからわかるように、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。したがって、遠隔地間のセッションのためには、伝統的なモデルに代わる演奏のモデルが必要となる。

遠隔地にいる二人の演奏者がセッションする際の RemoteGIG のモデルを、図 1 に示す。本モデルは、同一のコード進行の繰り返しを、その 1 周期分の時間 ( $D$ ) を一定にして演奏することを前提とする。そして、演奏者はお互いの演奏を、ちょうど周期  $D$  分の時間だけ遅れて聞き合い、それに反応して即興演奏する。例えば、図 1 において、演奏者 A の A-1 の区間の演奏音は、演奏者 B のものでは B-2 の区間で鳴る。コード進行は繰り返すため、A-1 のある演奏音が鳴っているときのコードは、その音が伝送されて B-2 で鳴るときコードと一致して調和する。このように、RemoteGIG のモデルは遅延がコード進行 1 周期分の時間よりも短ければよく、遠隔地間のセッションに適している。

### 2.1 実現方法

RemoteGIG を実現するには、大別して、音響信号を直接中継する方法と、MIDI 信号のようなシンボリック化した演奏情報を中継する方法の二種類が考えられる。音響信号の中継は、使用楽器の制約がない反面、MIDI 信号に比べて非常に伝送量が多く、受信側で演奏情報を再利用・加工しにくいという欠点がある。そこで本研究では、MIDI 信号を双方向に中継することで RemoteGIG を実現する。

一定の遅延時間で双方向の MIDI 中継をおこなうには、ネットワーク経由で MIDI 信号を伝送し、適切な時刻となるまでバッファリングして再生する必要がある。本研究では、このために、我々が設計した音楽用通信プロトコル RMCP (Remote Music Control Protocol)<sup>3)~7)</sup> を用いた。RMCP では、演奏情報は RMCP パケットという単位で伝送され、MIDI note

on/off などの各 MIDI メッセージは、一つの RMCP パケットの中に含まれて送信される。RMCP は、

- タイムスタンプを用いたパケットの時間管理
- 信頼性を確保した遠隔地間の双方向パケット中継

の二つの機能を提供できることから、今回の目的に適している。さらに RMCP は、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使い分けることで、分散音楽情報処理に適した効率が高く実用的な情報共有を達成している。具体的には、Ethernet などの LAN (Local Area Network) 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ーヘッドの小さい情報共有を実現する一方、Internet などの WAN (Wide Area Network) 経由の通信には TCP/IP を用いることで、信頼性を確保した情報共有を可能にしている。

## 2.2 RMCP による RemoteGIG の実装

RMCP によって RemoteGIG を実現する際の、プロセス構成の例を図 2 に示す。RemoteGIG は、MIDI メッセージを MIDI 機器から受信するための RMCP MIDI Receiver、計算機のキーボードとマウスを用いて演奏するための RMCP MIDI Station、受信パケット中の MIDI メッセージを MIDI 機器へ送信するための RMCP Sound Server、WAN で結ばれた二つの LAN 間で双方向に RMCP パケットを中継するための RMCP Gateway、標準 MIDI ファイル (SMF: Standard MIDI File) を再生するための RMCP SMF Player によって実現される。

演奏者が MIDI 楽器等を弾いて生成した MIDI メッセージが、どのように伝送されて RemoteGIG が実施されるかを、LAN と WAN の場合を分けて説明する。

### 1) LAN 上での MIDI メッセージの伝送

演奏者は、RMCP MIDI Receiver か RMCP MIDI Station を用いて演奏する。前者は、MIDI 楽器から受信した MIDI メッセージを RMCP パケットとしてブロードキャストする。後者は、鍵盤に

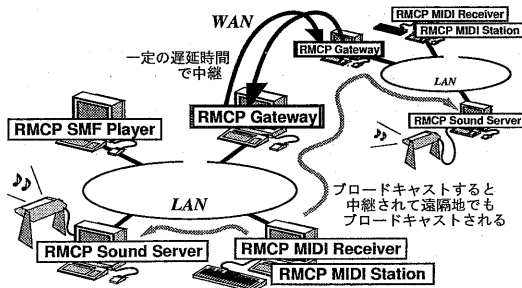


図 2: RMCP による遠隔地間のパケット中継

マッピングされたキーボードの操作や画面上の鍵盤のマウスクリックに応じて 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。そして、ネットワーク上のパケットを RMCP Sound Server が受信すると、パケットに含まれる MIDI メッセージを MIDI 楽器へ送り、演奏音として出力する<sup>★1</sup>。

### 2) WAN を経由した MIDI メッセージの伝送

図 2 のように、双方向のパケット中継をする WAN の両端で RMCP Gateway を実行し、TCP/IP のコネクションを確立する<sup>★2</sup>。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、他方からのパケットを受信すると、遅延時間が一定となるようにタイムスタンプ (パケットが処理されるべき時刻を表す) を書き換えて、自分の LAN 上にブロードキャストする<sup>★3</sup>。これにより、一方の演奏者が RMCP MIDI Receiver 等を用いて演奏した MIDI メッセージは、RMCP Gateway によって中継されて他方の LAN の RMCP Sound Server にも届き、演奏音として出力される。

以上のように、自分の演奏音が、自分と相手のもとにある MIDI 楽器によって出力されることで、遅延付きの遠隔地間のセッションがおこなえる。なお、コード進行の周期を遅延時間と同一に保つためには、コードとリズムがわかるような伴奏 (ベースとドラムス等) を SMF に記録しておき、RMCP SMF Player を用いていずれかの LAN 上で再生するとよい。この伴奏も RMCP Gateway によって中継され、両者で共有される。

## 2.3 実験結果

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 (第 23 回音楽情報科学研究会)<sup>(8),(9)</sup> の一環として、日本時間 1997 年 12 月 14 日 (日) 午前 11 時 (ニューヨークでは 13 日 (土) 午後 9 時) より、コンサート形式での公開実験 “Internet RemoteGIG” を実施した<sup>★4</sup>。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

★1 詳しい記述は文献 3), 4) に譲るが、これらのパケットを、RMCP Display Server が同時に受信して視覚化したり、RMCP Packet Recorder が受信して記録したりできる。これは、RMCP がブロードキャストによって LAN 上で通信する利点の一つである。

★2 RMCP Gateway の実装方法は、文献 3) に述べられている。

★3 中継の無限ループを回避するために、パケットヘッダー中の中継禁止フラグをセットしてからブロードキャストする。

★4 調性とリズムのある音楽を Internet を経由して遠隔地間でセッションしたコンサートとしては、我々の知る限り世界で初めての試みであった。

て RemoteGIG をおこなった。日本側の会場となった NTT ICC では、Rick Bassett が MIDI キーボードで演奏した。一方、USA 側の会場となったコロンビア大学では、2 曲演奏した中で、1 曲目を Brad Garton が MIDI キーボードで演奏し、2 曲目を Perry Cook が自作の MIDI コントローラーを用いて演奏した。

それぞれのサイトでは SGI O2 (IRIX-6.3) を一台占有使用し、RMCP Gateway、RMCP Sound Server、RMCP MIDI Receiver を動作させた。そして、同一の MIDI 音源 (Roland SC-55mkII) により演奏音を出力した。また、会場にいる聴衆が、日米のどちらの演奏者が弾いた音かを容易に確認できるように、画面上の鍵盤の色を変えて MIDI 情報を視覚化する RMCP Display Server を動作させた。さらに、ビデオカメラで撮影した動画を Internet を経由して中継し、演奏会場の風景をお互いに見ることを可能にした。

RMCP Gateway による中継用遅延時間は、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、これがコード進行 1 周期となるようなテンポで、伴奏用のベースとドラムス等が含まれた SMF を再生した。具体的には、1 曲目 (後藤 真孝 作曲) は 4/4 拍子 12 小節のブルース進行の曲であったため、テンポ 93.75 M.M. (4 分音符 = 640 ms) で演奏した。2 曲目 (Rick Bassett 作曲) は 4/4 拍子 16 小節のコード進行を持っていたため、テンポ 125 M.M. (4 分音符 = 480 ms) で演奏した。

実験の結果、知覚できるテンポの揺らぎやパケットロスが発生せず、実際に遠隔地間で RemoteGIG がおこなえることを確認した。ただし RemoteGIG とは直接関係ないが、1 曲目と 2 曲目の間で、画像中継用プログラムの置かれたファイルサーバがクラッシュしてしまい、動作させることができないというトラブルがあった。そのため、予定していた Perry Cook の自作 MIDI コントローラーの画像付き紹介は、断念せざるをえなかった。RMCP 関連のプログラムはすべて別のハードディスクにあり、RemoteGIG の演奏に関しては全く問題なくおこなうことができた。

## 2.4 考察

RemoteGIG の実験を通じて得られた考察を以下にまとめる。

### □ 演奏の形態

RemoteGIG では、演奏開始時、ソロ演奏時、演奏終了時において、一箇所に集まっておこなうセッションとは異なる演奏形態が求められた。その一例を図 3 に示す。

まず演奏開始時には、B-1 の演奏が演奏者 A に届



図 3: RemoteGIG の演奏形態の一例

くまでに 2 コーラス (コード進行 2 周期) かかってしまう。その A-1 と A-2 の期間、演奏者 A は一人で演奏をする必要があった。

次にソロ演奏に関しては、通常は 1 コーラスごとに交代してソロを弾くことがよくおこなわれている。しかし、例えば演奏者 A が A-1 と A-2 のソロ演奏の後に A-3 で演奏者 B のバックギングに移るとすると、本来は演奏者 B は B-1 でソロを弾いていなければならないが、演奏者 B のものでは A-1 のソロと B-1 のソロが重なってしまう問題が生じる。そこで今回は、2 小節交代あるいは 4 小節交代で、ソロとバックギングを繰り返し演奏する形式を考案した。例えば、16 小節のコード進行を 4 小節ずつに分割して、「A のソロを 4 小節 → B のソロを 4 小節 → A のソロを 4 小節 → B のソロを 4 小節」という形式で演奏した。さらに、演奏の進行に伴い、交代する小節数を短くしていくような形式も効果的であった。

演奏終了時には、伴奏の終了と共に演奏を終えるための工夫が必要であった。そこで、図 3 の S-6 と S-8 の伴奏の位置にフィルインなどの合図を入れておき、演奏者 B は S-6 を聞くと演奏を終了し、演奏者 A は S-8 を聞くと演奏を終了することとした。

以上を踏まえた RemoteGIG の拡張案としては、SMF を事前に入れておいて、両者の演奏開始時刻を合わせる機能を実装することが挙げられる。これにより、図 3 の演奏者 B の演奏開始を 1 コーラス早めることができる (演奏者 A のものでは A-2 と B-1 が重なる)。そして、両者が奇数番目のコーラスにソロを演奏し、偶数番目のコーラスにバックギングを演奏することで、1 コーラスごとに交代する演奏形式も可能となる。また、RMCP Gateway に MIDI 中継を途中で中止する機能を実装すれば、MIDI 中継を S-8 の直後に止めることができ、両者が伴奏の最後まで演奏できるようになる。

### □ 音楽以外のコミュニケーション手段の必要性

演奏前後の準備等でお互いの状況を的確に伝達するためには、演奏音以外のコミュニケーション手段も不可欠であった。そこで今回は、UNIX の talk コマンド

によりテキストベースでリアルタイムチャットをおこなった。このような手段の確保は、今後 RemoteGIG を実施する際にも必ず考慮すべき点の一つである。

#### □ ネットワークトラフィックの問題

コンサート中は問題が起きなかったが、現在の Internet の混雑の状況では、時間帯によって、演奏中の遅延時間がコード進行 1 周期よりも突然長くなってしまうことがある。コンサート前に早稲田大学とコロンビア大学間でリハーサルを繰り返した際には、日本時間の夕方などに、ほとんど TCP/IP パケットが通らない状況が発生したことがあった。

### 3 不特定多数のユーザが参加できる遠隔セッション: Open RemoteGIG

Open RemoteGIG は、多数の人が自由に RemoteGIG に参加することを可能にする遠隔セッションシステムである。これは、基本的には 2 で述べた RemoteGIG に基づいているが、RemoteGIG をそのまま不特定多数のユーザに対して運用しようとすると、以下のような問題が生じてしまう。

1. 演奏者を増やすためには、すべての演奏者の LAN 間で、RMCP Gateway による全対全結合のコネクションを確立しなければならなかった。
2. 遅延時間等はコネクションを確立した後に変更できなかったため、コネクションを張ったままで様々な曲を演奏するには、コード進行の周期を合わせなければならないという制約があった。
3. 事前に演奏相手を定め、両者の LAN 間でコネクションを確立しておく必要があった。そのため、面識のない相手と演奏することはできなかった。
4. 伴奏に用いる SMF の情報は、事前に内容が確定しているにも関わらず、演奏中に伝送していた。これは汎用性を持たせるためには有効な設計であったが、Open RemoteGIG のように多数の演奏者が参加する場合には、伝送容量を圧迫してしまう点が問題となる。
5. 2.4 でも指摘したように、音楽以外のコミュニケーション手段は提供していなかった。しかも、演奏者が 3 人以上の場合は、多対多のコミュニケーション手段が必要となる。

そこで本研究では、これらの問題を以下のように解決することで、Open RemoteGIG を実現する。

#### 1. スター状のネットワーク接続

Open RemoteGIG では、演奏中継用のセンターを用意する。そして、各ユーザは自分の計算機

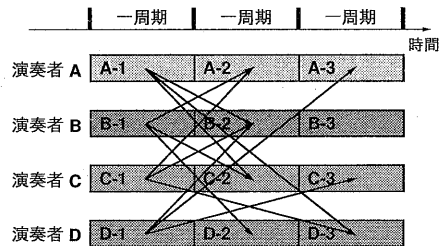


図 4: Open RemoteGIG のモデル

から、Internet 等の WAN 経由でセンターに接続(コネクションを確立)するだけで、他の複数のユーザの演奏音を同時に聞きながら即興演奏ができる。四人の演奏者がセッションする際の Open RemoteGIG のモデルを図 4 に示す。このようなシステムでは時間軸方向の制御が重要であるが、Open RemoteGIG では全ユーザ間で共通の仮想時刻(例えばセンターの時刻)を設定し、仮想時刻上で図 4 のような演奏がおこなえるようにする。

2. 中継用遅延時間を動的に変更できる演奏中継 RemoteGIG と同様に同一のコード進行を繰り返し演奏することを前提とし、コード進行 1 周期の整数倍の遅延時間で演奏情報(MIDI メッセージ)を中継する。その際、コネクション確立後の任意の時点で遅延時間を変更できるようにすることで、様々なコード進行の異なる周期に対応可能とする。また中継用遅延時間は、ネットワークの遅延時間よりも十分長くなければならないという前提を考慮しながら、コード進行の何周期分とすべきかを決定する。
3. 演奏相手の発見環境  
センターに接続している他のユーザを見つけ、一緒に演奏を開始できる環境を用意することで、面識のない相手との遠隔セッションを可能にする。ただし、各ユーザの演奏が正しく再生されることを保証し、それぞれが異なる音色を指定できるようにするために、ユーザに対して MIDI チャネルを排他的に割り当てるよう管理する(現在の実装では、1~9 の MIDI チャネルを割り当てている)。そのため、同時に演奏できる人数は割り当て MIDI チャネル数に制限される。
4. SMF の事前伝送  
コード進行 1 周期分が記録された SMF を演奏前にダウンロードしておき、各ユーザのもとで繰り返しループさせて再生する。その際、ユーザ同士の即興演奏が時間軸上で調和するために、SMF の再生開始時刻が全ユーザのもとで同一の仮想時刻となるように調整する。なお、SMF が使用

する MIDI チャンネルは、上記の割り当て用 MIDI チャンネル以外を使用しなければならない(現在の実装では、伴奏が 10~16 の MIDI チャンネルを使用する)。

5. テキストベースのリアルタイムグループチャット演奏音以外のコミュニケーション手段として、任意のユーザ間の多対多のチャットを実現し、演奏内容の打ち合せ(例えば、ソロを何小節ごとに交代するか)等を可能にする。

本研究ではさらに、Open RemoteGIG の機能として以下の二つを提案する。

- 仮想演奏スタジオの切り替え(入退室)機能  
複数の仮想演奏スタジオ(GIG チャンネルと呼ぶ)を用意し、異なるユーザグループが独立に RemoteGIG セッションをおこなうことを可能にする。GIG チャンネルごとに異なる伴奏の SMF を用意しておけば、ユーザは GIG チャンネルを切り替えることで、好みの伴奏に合わせて即興演奏ができる<sup>★5</sup>(現在の実装では、異なる伴奏を持つ 3 つの GIG チャンネルを試験的に用意している)。前述の MIDI チャンネルの割り当てでは、この GIG チャンネルごとに独立におこなう。
- 観客の立場での演奏の鑑賞機能  
演奏に参加せずに、他のユーザの演奏音だけ聞く機能(鑑賞モード)を提供する。これにより、即興演奏のできないユーザでも観客の立場で Open RemoteGIG を利用できる。あるいは、すべての MIDI チャンネルが割り当て済みのときに、MIDI チャンネルが空くまでは観客として待機してもよい。この機能を活用すれば、大勢の観客に対し、Open RemoteGIG の公開ライブパフォーマンスを Internet 上で実施できる。

### 3.1 システム構成

Open RemoteGIG のシステムは、演奏中継用のセンターである RemoteGIG Center と、各ユーザがセンターに接続するための RemoteGIG Transceiver で構成される。システムの構成図を図 5 に示す。Open RemoteGIG の機能は、演奏中継機能と GUI・コミュニケーション機能に二分でき、前者は RemoteGIG と同様に RMCP を用いて実現し、後者は WWW ブラウザ上で動作するように Java Applet として実現する。実際の MIDI 楽器との入出力には、2.2 で述べたように RMCP Sound Server, RMCP MIDI Receiver 等を用いる。

★5 GIG チャンネル 1 はフュージョン風、GIG チャンネル 2 はジャズ風、GIG チャンネル 3 はロック風、といったように、様々なジャンルの伴奏をセンター側で用意しておく効果的である。

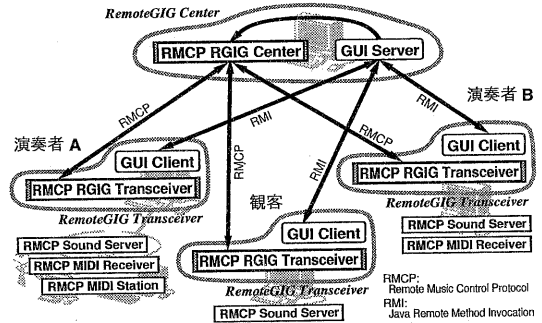


図 5: Open RemoteGIG のシステム構成

#### 3.1.1 RemoteGIG Transceiver

RemoteGIG Transceiver は、ユーザの演奏情報をネットワークを経由して送受信するための RMCP RGIG Transceiver と、ユーザからの指示を伝達したり他のユーザの属性を表示する GUI Client から成る。ただし、図 5 に示したように、両者は直接通信せず、常に RemoteGIG Center を経由する。

##### □ RMCP RGIG Transceiver

演奏中継に関しては基本的に RMCP Gateway と同等の機能を持ち、後述する RMCP RGIG Center と TCP/IP のコネクションを確立して RMCP パケットを送受信する<sup>★6</sup>。その際、RMCP RGIG Center へ送信する前に、パケットのタイムスタンプを実時刻(ローカル時刻)から仮想時刻に変換する<sup>★7</sup>。また、RMCP RGIG Center から受信したパケットのタイムスタンプを仮想時刻から実時刻に変換し、中継用遅延時間を加えた後に LAN 上にブロードキャストする。

さらに、RMCP RGIG Center から受信する以下のような指示に応じた処理をおこなう。

「伴奏用 SMF のダウンロード」 RMCP RGIG Center に接続したときに、全 GIG チャンネル分の SMF を予め受信する。同時に、それぞれのテンポとコード進行 1 周期の長さ、開始仮想時刻も受け取る。

「GIG チャンネルの指定」 これを受信すると伴奏用 SMF を切り替え、そのコード進行の周期とネットワークの遅延時間に基づいて、新たな中継用の遅延時間を決定する。

「SMF の演奏開始/演奏停止」 演奏開始を受信すると、開始仮想時刻とコード進行の周期から次の伴奏開始実時刻を計算し、演奏停止を受信するまで無限に繰り返しながら SMF を再生する。

★6 異なる RMCP サーバ番号やソケットポート番号を用いれば、同一 LAN 上で複数の RMCP RGIG Transceiver を独立に運用できる。

★7 これは、文献 3) の RMCP Gateway による内部時計の時刻差の補正と同等の処理である。

「鑑賞モード／演奏モード (割り当て MIDI チャンネルの指定付き) の変更」 これを受信すると、鑑賞モードのときは RMCP RGIG Center への MIDI メッセージの送信を停止する。演奏モードのときは、指定された MIDI チャンネルの MIDI メッセージのみを送信する。

「MIDI のオールノートオフ／プログラムチェンジの送信命令」 これを受信すると、当該 MIDI メッセージをブロードキャストする。

## □ GUI Client

ユーザ用の GUI は、演奏管理ウィンドウ (図 6) とコミュニケーションウィンドウ (図 7) で構成される。ユーザは、GUI Client を起動して GUI Server に接続した後に、コミュニケーションウィンドウで演奏相手を見つけたり、希望の GIG チャンネルに切り替えたりする。その後、演奏管理ウィンドウで演奏状態の変更を指示する。

コミュニケーションウィンドウ (図 7) の左下で、ユーザは名前を入力して login する。すると、右側の “All Users” のサブウィンドウに、現在接続している全ユーザの一覧が属性と共に表示される。ここでは演奏中であるかどうかを問わず、任意のユーザ (複数も可) を指定してチャットができる。グループチャットを簡

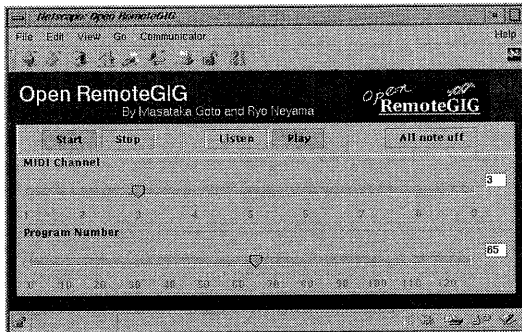


図 6: GUI Client の演奏管理ウィンドウ

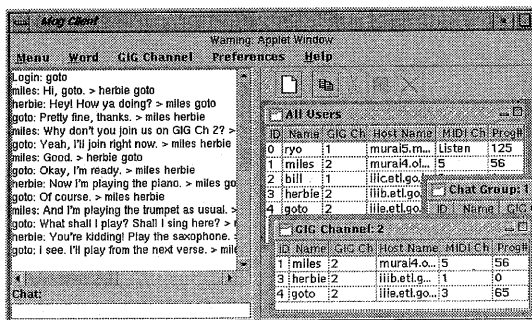


図 7: GUI Client のコミュニケーションウィンドウ

便におこなうために、新たなサブウィンドウ (“Chat Group: 番号”) を作成し、選択したユーザを登録しておく機能もある。チャット相手を切り替えるには、このサブウィンドウをクリックするだけでよい。そして、“GIG Channel” のメニューから希望の GIG チャンネルの番号を選ぶと、“GIG Channel: 番号” のサブウィンドウが現れ、その GIG Channel で演奏／鑑賞しているユーザの一覧が表示される。その後、演奏管理ウィンドウの操作に移れるようになる。

演奏管理ウィンドウ (図 6) では、SMF の演奏開始 (“Start”)・停止 (“Stop”) と、鑑賞モード (“Listen”) と演奏モード (“Play”) の切り替えをボタンで指示する。演奏モードに切り替える前に、割り当て希望 MIDI チャンネルをスライダで指定する。どの MIDI チャンネルが空いているかは、“GIG Channel: 番号” のサブウィンドウを見るとわかる。他にも、MIDI プログラム番号の変更 (音色切り替え) や MIDI オールノートオフの送信を指示できる。なお、全ユーザのものでプログラム番号と音色の対応の互換性を保つために、MIDI 音源として GM (General MIDI) 音源を用いることを前提とする。

## 3.1.2 RemoteGIG Center

RemoteGIG Center は、演奏情報を全ユーザのもとへ中継する RMCP RGIG Center と、全ユーザの属性を管理・中継する GUI Server から成る。ユーザの属性の変更や RMCP RGIG Transceiver への動作指示は、図 5 のように GUI Server から RMCP RGIG Center へ一方向に通知される。ユーザの属性には、ユーザ ID、GIG チャンネル、鑑賞モード／演奏モード、MIDI チャンネル、MIDI プログラム番号があり、動作指示には、演奏開始／停止命令、MIDI オールノートオフ送信命令がある。

## □ RMCP RGIG Center

全 RMCP RGIG Transceiver からの接続を受け付け、GIG チャンネルごとに分けて管理する。RMCP RGIG Transceiver の接続時に伴奏用 SMF を送信した後は、受信したユーザの演奏 (RMCP パケット) を、それ以外の全ユーザへ送信する処理をおこなう。また、GUI Server からユーザ属性の変更や動作指示を通知されると、対応する処理をしたり、当該 RMCP RGIG Transceiver へ送信したりする。MIDI プログラム番号の変更に関しては、その GIG チャンネルの全 RMCP RGIG Transceiver へ送信する。

## □ GUI Server

全 GUI Client のユーザの属性管理と変更の通知、チャットの中継をおこなう。GUI Client の操作による属性変更や動作指示は GUI Server へ集められ、RMCP

RGIG Centerへ通知される。同時に、画面表示を更新するために全 GUI Clientにも通知される。RMCP RGIG TransceiverがRMCP RGIG Centerに接続されていることを確認する処理(IPアドレスに基づいて同定する)や、MIDIチャンネルの割り当ての排他処理もおこなう。

### 3.2 実装

RMCP RGIG CenterとRMCP RGIG TransceiverはC言語で実装し、両者間の通信にはRemoteGIGと同様にTCP/IP上のRMCPを用いた。RMCP RGIG TransceiverはUDP/IP上のRMCPにより、RMCP Sound Server, RMCP MIDI Receiver等と通信する。

一方、GUI ServerとGUI ClientはJava言語(JDK/JRE 1.1.5, 1.1.6)で実装した。GUI ClientはJava Appletとして実装され、WWWブラウザ(Netscape Communicator 4.05, HotJava Browser 1.1.4等)やappletviewer上で動作する。HTTP(Hyper Text Transfer Protocol)によってダウンロードしたGUI Clientは、RMI(Java Remote Method Invocation)を利用してGUI Serverとの通信をおこなう。GUIの実装には、JFC(Java Foundation Classes) 1.1を利用した。また、GUI ServerからRMCP RGIG Centerへのユーザ属性変更・動作指示の通知には、JNI(Java Native Interface)を用いた。

### 3.3 実験結果

Open RemoteGIGによって実際に遠隔地間でセッションが実施できることを確認するために、電子技術総合研究所(電総研)と早稲田大学(早大)間でInternetを経由した遠隔セッションを試みた。Open RemoteGIGを運用するOSには、IRIX 6.2, 6.3, 6.4 およびSolaris 2.5を用いた。RemoteGIG Centerを電総研あるいは早大に設置し、電総研と早大において複数のRemoteGIG Transceiverを動作させて実験した。伴奏用SMFには、2.3の実験の際のSMF等を使用し、一周期の時間は30秒前後でSMFごとに異なる値を設定した。

実験の結果、遠隔地間で全ユーザの演奏が適切に中継され、チャット等のGUIに関連した機能も設計通り動作することを確認した。特に、GUIやチャット等を利用して、他のユーザの状況を把握しながら遠隔セッションをできる点が有効であった。

## 4 おわりに

本稿では、1997年12月のRemoteGIGの公開実験の結果と考察を述べ、それを踏まえた上で、面識の

ない多数の人がInternet経由でRemoteGIGの遠隔セッションに参加できるシステムOpen RemoteGIGを提案した。従来の音楽では、演奏者は全員ほぼ同じ演奏を聞くものという暗黙の前提があったが、本研究での音楽では、全員が同じ演奏を聞くことがなく相手の反応もすぐにはわからないという状況下での、従来なかった音楽的なインタラクションの方法論が求められる。RemoteGIG及びOpen RemoteGIGは、将来普及するであろう遠隔セッションの一つの新しい方向性を示すものであると言える。

我々は、Open RemoteGIGを実際に多くの人に使うてもらえるよう、WWW上で公開運用を開始する予定である。今後は、遠隔作曲、遠隔編曲、遠隔演奏レッスン等の、RMCPによる双方向MIDI中継の他の可能性も探求していきたい。

### 謝辞

村岡 洋一 教授には、本研究を様々な角度から御支援頂いた。また菊地 淑晃 氏からは、本研究に対し有益な御示唆を頂いた。1997年12月のRemoteGIGの公開実験においては、Rick Bassett 氏、Brad Garton 氏、Perry Cook 氏に演奏者として協力して頂き、兼 孝之 氏にはコンサートの機会を与えて頂いた。また、松田 周 氏を始めとする国立音楽大学の方々にはNTT ICCでの準備に協力して頂き、興相 正克 氏には動画像の伝送で協力して頂いた。本研究に協力して頂いたすべての方々に、心から感謝の意を表する。

### 参考文献

- [1] 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).
- [2] Nielsen, O.: MIDI and Audio via ISDN, *Proceedings of the 1994 ICMC*, pp. 451-454 (1994).
- [3] 後藤真孝, 根山亮, 菊地淑晃, 村岡洋一: RMCP: Remote Media Control Protocol — 時間管理機能の拡張と遅延を考慮した遠隔地間の合奏 —, *情処研報 音楽情報科学 97-MUS-21-3*, Vol. 97, No. 67, pp. 13-20 (1997).
- [4] Goto, M., Neyama, R. and Muraoka, Y.: RMCP: Remote Music Control Protocol — Design and Applications —, *Proceedings of the 1997 ICMC*, pp. 446-449 (1997).
- [5] 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).
- [6] 後藤真孝: 分散協調インタラクティブシステム — 音とCGによる遠隔地間のインタラクション —, *NICOGRAPH'93 CG教育シンポジウム プロシーディングス*, pp. 44-49 (1993).
- [7] 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 —, *情処研報 音楽情報科学 93-MUS-4-1*, Vol. 93, No. 109, pp. 1-8 (1993).
- [8] Rai, T. and Bassett, R.: Program book of U.S.A./Japan InterCollege Computer Music Festival: Computer Music Concerts and Lectures (1998).
- [9] Hashimoto, S., Moon, B., DuBois, R. L. and Castle, H.: Reginal news, *ARRAY: Communications of the ICMA*, Vol. 18, No. 1, pp. 2-4 (1998).