疑似ライブ音楽演奏システムの改良

2007MI134 光枝靖章　　2007MI268 横川敬弘
指導教員　後藤 邦夫

1　はじめに

現在リアルタイムで音声や映像の配信が可能である．日々インターネットを用いた音楽によるコミュニケーションや情報供給の需要は増え，今後も普及すると考えられる．しかし複数の相手とリアルタイムで演奏をやりとりし，手軽に配信できるシステムは普及していない．伝送路による遅延の解消が困難さが原因である．

そこで離れた場所にいても実用的な音楽の演奏視聴をリアルタイムに近い状態で共有できるシステムがあると良いと考え，複数人による疑似ライブ演奏システム [1] [2] を提案する．本研究は 2007 年度卒業論文として橋口，川井が考案したオーバレイマルチキャストを用いて複数人に同時に演奏データを送信，受信させるシステム [2] に改良を加える．

システム構築・考案は共同で実施し，主に光枝は演奏者のルーティングをするツールとチャット機能，横川は既存システムの修正と GUI 部を担当した．

2　既存システムの概要

既存システムは C++ と Qt3 で実装された．GUI 部と録音システム部の 2 つに分かれる．GUI 部と録音システム部は TCP 通信方式で接続される．演奏データはリアルタイムで重ね録りされ，最後に MIXER 役の人間がミキシングをする．ミキシング後のデータがライブ演奏データとして配信される．配信システムは実装されていない．

3　既存システムの変更点

この節では既存システムからの変更点を述べる．

3.1　既存システムの改良点

既存システムを現環境にて実行した際に不具合が生じた．Qt のバージョンが古くコンパイルが出来なかったので Qt3 から Qt4 に移植した．また複数ターミナルを開く手順を 1 つのターミナルから操作できるよう改良した．

3.2　演奏データについて

演奏者間の通信方式は UDP を用いる．また一般家庭で想定される使用可能なバンド幅の数値を数 M〜数十 Mbps と改める．最大使用トラック数は既存システムの 14 トラックから 16 トラックに変更するため，ステレオ録音 (2 トラック使用) で最大 8 人までがセッションに参加可能となる．

独自ヘッダ情報を次に示す．演奏データには 2 種類あり演奏データが入ったもの，楽器名が入った演奏者の情報を含めたものがある．図 1 にそれぞれの構造体を示す．

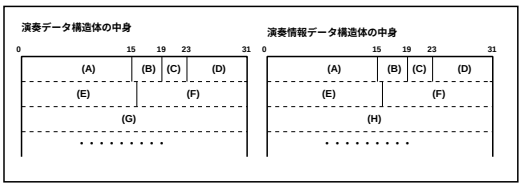

図 1　データ構造体

- (A) type：パケットの情報を識別する．0 の時演奏データ，1 の時演奏情報データ，2 の時配信データとなる．
- (B) version：プロトコルのバージョン．通信する際に必要な交信方法を示す．本システムのデータはビッグエンディアンである．
- (C) channnels：演奏データが格納されている使用チャンネル数を示す．
- (D) seq：パケットの番号を表す．パケットのロスや入れ換えに利用される．
- (E) junc_flag：分岐のしるしを表す．データが分岐後のものであるのかの判別，分岐後のデータの合流時に利用される．
- (F) s_track：分岐後の使用トラックを表す．分岐後のデータの時使用したトラック数が加算される．
- (G) sample：演奏データ．サイズは 1 トラックあたり 100byte から 88byte に変更する．
- (H) part：演奏している楽器名．各トラックが各楽器を演奏しているかを示す．1 トラックあたり 20byte とする．

演奏データ構造体は演奏した音楽のデータが入る．演奏情報データ構造体は演奏しているパートの情報が入り，8 秒ごとに送信される．既存システムのデータ構造のヘッダ定義は 32bit 境界になるように再定義した．

4　提案するシステム

通信時の演奏データの再生及び書き込みのズレを自動補正する機能，演奏ルーティングをより簡単に操作する演奏者ルーティングシステム，演奏者同士でコミュニケーションをとるチャット機能を追加する．

5　演奏データの遅延処理

リアルタイムでの演奏録音をする際，データの書き込みのずれは致命的である．人の耳で聞いて許される演奏のずれは 20 ミリ秒であり [3]，20 ミリ秒以内に納める必

要がある．既存システムではキューを用意し，書き込み遅延を解消した．ただしキューにためるパケット数を自分で測定・計算し，入力する必要があった．

そこで本システムでは GUI アプリケーションに MicTest ボタンを追加する．クリックすることで出力と書き込みのずれから書き込み遅延を測定し，計測値からキューにためるパケット数を算出し書き込み遅延の処理をする．データ構造体の変更に伴い，1 パケットのあたりのデータ量は 4.5 ミリ秒だったが 4.0 ミリ秒となる．1 パケット (4.0 ミリ秒) 単位でのずれの修正が可能である．

6 演奏者ルーティングシステム

演奏者ルーティングシステムとは演奏者がどの演奏者に演奏データを送るかを決めるシステムである．

本システムは次のように構成されている．

1. XML データ構造
2. XML ファイル生成機能
3. データベース登録機能
4. XML ファイル結合機能
5. XML ファイル取得機能

それぞれの機能については次の節以降で解説する．

次の図 2 を使って演奏者ルーティングシステムとは何かを具体的に説明する．

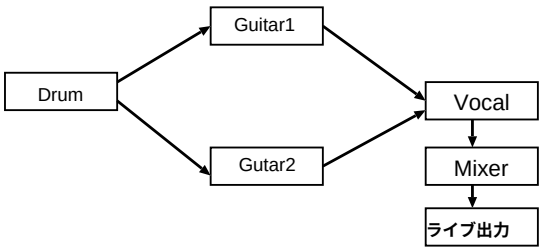

図 2　演奏形態の例

Drum を担当している演奏者が Guitar1，Guitar2 に演奏データを送信する場合を考える (図 2)．従来のシステムではデータを送信する相手の IP アドレスと port 番号をあらかじめ調べ，GUI アプリケーションの入力欄に入力する必要があった．本システムでは各演奏担当者が自分の情報を入力し，XML ファイルとして生成する．生成ファイルは web ページからサーバのデータベースに登録する．Drum 演奏者が GUI アプリケーションの GET ボタンをクリックすると入力欄に Guitar1, Guitar2 の情報が自動的に入力される．なお XML ファイルを生成，結合する機能は Qt4 ライブラリの XML モジュールを利用した．

6.1 XML ファイルデータ構造

XML を使いデータを次の図 3 ように定義した．バンド全体で 1 つのツリー構造となっている．各パート同士は兄弟のノードとなっている．

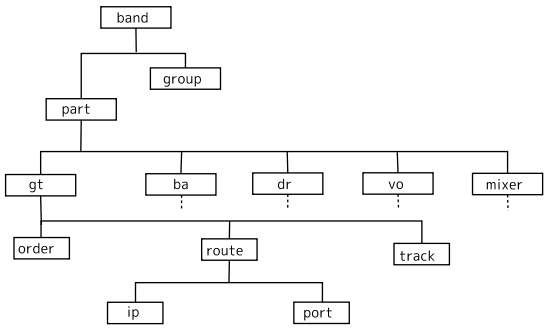

図 3　データ構造

1. group: バンド名を表す
2. part：パート名を表す
3. order: 演奏データを送信する順番を表す
4. route: ip タグ，port タグを持つ
5. ip：IP アドレスを表す
6. port：port 番号を表す (60000 で固定)
7. track: 仕様トラック数を表す

なお他のパートも gt タグと同じ子ノードを持っている．

6.2 XML ファイル生成機能

演奏者の情報を XML で定義したタグにしたがって表現されたファイルを生成するアプリケーションである．各演奏者は GUI アプリケーションを起動し必要な情報を入力し，SAVE ボタンをクリックすると演奏者の情報が書かれた XML ファイルが生成される．

IP アドレスは自動で取得され，port 番号は 60000 で固定なので入力しない．

6.3 データベース登録機能

次の図 4 は XML ファイルをデータベースに登録する流れを簡単に示したものである．

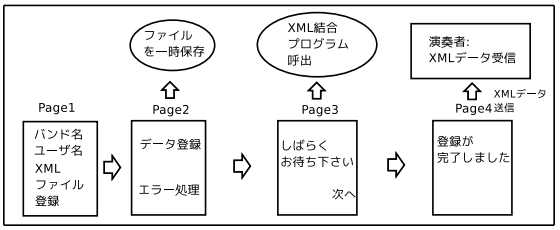

図 4　web ページ

データベース登録機能は postgresql と php を使い実装した．

- ページ 1: バンド名，ユーザ名と XML ファイルを登録するページ．
- ページ 2: データベースに登録するページ．

- ページ 3: 各演奏者の XML ファイルを結合するプログラムの呼出.
- ページ 4: 結合したファイルを演奏者に送信.

演奏者はページ 1 に web ブラウザでアクセスしユーザ名，バンド名，生成した XML ファイルを入力する．upload ボタンをクリックするとページ 2 に遷移し XML ファイルがデータベースに登録される．ページ 2 の Next ボタンをクリックしページ 3 に遷移する．この時演奏者からは見えないが他の各パートの XML ファイルを結合するプログラムを呼び出している．この機能については次の 6.4 節で説明する．チャットで連絡をとり全員の XML ファイル登録が完了後ページ 4 に遷移し，バンド全体の情報が記載された XML ファイルをサーバから受信する．送信については 6.5 節で述べる．

なお XML ファイル登録については bytea 型のテーブルを作成しバイナリデータとして登録している．bytea 型はデータを処理する場合大量のメモリを消費するが本システムの XML ファイル程度なら問題ないと判断した．XML ファイルを結合するプログラムの呼出しに関しては PHP の proc_open 関数を用いた．この関数を用いることによってターミナルとのやりとりを想定しているアプリケーションにも引数を渡すことが可能となった．

6.4 XML ファイル結合機能

各演奏者の情報を表現した XML ファイルを結合しバンド全体の情報を 1 本のツリーとして結合する機能である．このプログラムはサーバ側にあり，PHP から呼び出す．

XML ファイル結合機能は Qt4 の XML モジュールを用いて実装した．Qt4 は本来 GUI アプリケーションを実装するライブラリだが，サーバで動作させるので GUI 部分は実装していない．

最初に Drum が図 4 のページ 4 までアクセスする．この時に XML を結合するプログラムを PHP の proc_open 関数を使って呼び出す．プログラムに必要な引数はパイプして渡す．一番最初にアクセスしてきた場合バンド情報を保持した XML ファイルがないので，新規に作成する．次にベースがアクセスした場合，同様に結合するプログラムを呼出す．この時バンド情報を保持したファイルが存在するので，map コンテナに情報を一時保存し，ベースの情報と map コンテナの情報を合わせて保存する．次にアクセスしてきた演奏者も同様に処理する．結合が完了したファイルはデータベースの別のテーブルに登録される．

6.5 XML ファイル取得機能

結合された XML ファイルのデータをサーバから取得する機能である．6.3 節の図 4 で page4 に演奏者がアクセスした時に，サーバは TCP 通信で演奏者にデータを送信する．演奏者はアプリケーションを起動した段階で TCP の通信を待っており，サーバからデータを受信すると終了する．

演奏者が GUI の GET ボタンをクリックすると演奏者の order を引数にして次の順番の演奏者の情報を取得する．例えば先の例で挙げた Drum が Guitar1, と Guitar2 に演奏データを送信する場合，演奏順番は Drum が 1，Guitar1 と Guitar2 が 2 になる．そして順番が 2 である Guitar1，Guitar2 の情報が GUI アプリケーションの入力欄に自動で取得される．

7 チャット機能

既存システムでは演奏者がチャット機能を用いてコミュニケーションをとることを前提としていた．しかし既存システムでは実装されていないので，チャット機能を PHP スクリプトを使って実装した．チャット機能を用いて各演奏者の Part やルーティング，演奏タイミングを決める．

8 ユーザビリティ評価

本節では実際に作成した Qt アプリケーションの評価方法・結果について述べる．

8.1 評価方法

先行研究と本研究で作成した Qt アプリケーションのそれぞれの評価をアンケート用紙に記入してもらう．旧システムと新システムの評価点 (1 から 5) の平均から相対評価を算出する．評価は本研究室及び他研究室の卒論生，計 17 名に協力してもらった．

8.2 ユーザビリティ評価結果

次の表 1 に評価結果を示す．評価結果から，評価した

表 1　ユーザビリティ評価結果

評価項目	旧研究の評価点	本研究の評価点
操作性	2.65	4.06
操作手順	2.53	4.06
利便性	2.71	4.59
総評	2.65	4.29
合計	10.53	17

すべての点において改善を図ることができた．相対的な全体の評価は約 1.6 倍まで上昇した．

9 録音実験

先行研究と同じように録音が出来るか録音実験を行う．

9.1 実験方法

先行研究 [1] と同じ構成とする．ノート PC4 台，デスクトップ PC1 台の計 5 台を使用して実験をした．演奏に使用したパートは Drum, Bass, Guitar, Mixer の 4 つである．デスクトップ PC は実ネットワークの模倣に遅延・ロスを起こすネットワークエミュレータ GINE3[4] として動作し，ノート PC はエミュレータを経由して通信する．

次の図 5 のように NIC を 4 つ搭載しているデスク

トップ PC を中央に配置し，それぞれの NIC に PC を接続する．演奏の順序は (1)Drum，(2)Bass と Guitar，(3)Mixer である．

HostA がエミュレータを通して HostC と HostD へ演奏データを送信する．HostC と HostD では HostA の演奏を聞きながら自分の演奏を加え，HostB に演奏データを送信する．HostB では HostC と HostD の演奏を聞きながら自分の演奏を加える．

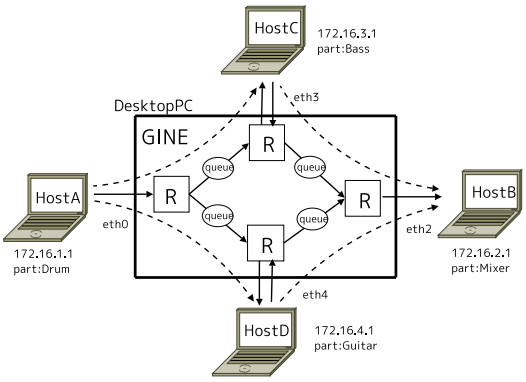

図 5　実験構成

9.2　検証方法

評価ポイントは次のとおりとする．

1. 演奏データの複数同時送信ができる．
2. 演奏データの合流ができる．
3. Mictest ボタンで書き込み速度の調整ができる．
4. パケットロスの対処．
5. 遅延の対処
6. 演奏情報パケットの送信と受信，保存ができる．
7. 演奏者の情報を Web サーバに登録できる．
8. 登録情報を元に演奏者のルーティングができる．
9. チャット機能が使用できる．
10. 以上を含め，全ての操作を Qt アプリケーションにて操作できる．

9.3　評価結果

チャット機能にて打ち合わせた後，演奏者の情報の登録とルーティングが Qt アプリケーションで操作できた．図の構成で遅延やロスの無い状態で実験をしたところ，各ホストの録音データが HostB で聞けたことから，分岐・合成は出来ていたと判定した．また，Mictest ボタンを使用して音声のずれを調整できた．演奏情報の表示を GUI にて確認し，演奏情報パケットの送受信と保存ができた．1，2，3，6，7，8，9 は成功した．

次に同じ実験構成でロス率を変化させ，評価した．結果は次の表 2 のとおりである．

上記より，ロス率が 5% 以内ならば問題なくシステムは実行できる．しかし，それを越えると雑音が入り，高音質な作品の製作は困難と思われる．

次に同じ実験構成で遅延を 10msec から 500msec まで変化させた．結果は片道のみの遅延，両方の遅延いずれの場合もデータの受信・合成ができた．ただし，先行研究と同じく 100msec 以上の遅延があると HostA から HostB へデータが受信・合成されるまでの時間が非常に長くなった．

表 2　評価結果

ロス率	リズム	音質
1%∼5%	乱れない	ほぼ普通に聞ける
6%∼10%	やや乱れる	多少雑音が入る
11%∼15%	乱れる	雑音がする
15%∼20%	乱れる	悪い

10　おわりに

本研究ではオーバレイマルチキャストを用いて複数人に同時に演奏データを送信受信する機能を新たな GUI 追加により簡単な手順にて出来るよう改良した．

このシステムがより使い易くなったことにより，同じ趣味を持つ人数の限られたコミュニティの中で，一つの音楽ツールとして利用できると考える．

本研究では同研究室の牧・長江の卒業研究，P2P ストリーミング放送システムの改良 [5] と内容が重複するので，リスナー機能は実装していない．P2P ストリーミング放送システムの改良のシステムを組み合わせる事で，より完成度の高いシステムとなる．

今後の展望としてはリスナーの実装，スマートフォンなど携帯端末への組み込みが考えられる．実現すればさらに便利なアプリケーションとなるだろう．

参考文献

[1] 橋口雅一, 川井一希: オーバーレマルチキャストを利用した疑似ライブ録音システムの試作, 卒業論文, 南山大学数理情報学部 情報通信学科 (2008).

[2] 近藤美希, 中野好絵: 音楽の遠隔疑似ライブシステムの試作, 卒業論文, 南山大学数理情報学部 情報通信学科 (2007).

[3] U.Peter SVENSSON, Asbjorn SAEBO: A low-latency full-duplex audio over ip streamer, graduation thesis, Norwegian Univercity of Science and Technology (2006).

[4] Y. Sugiyama and K. Goto: Design and implementation of a network emulator using virtual network stack. In *Proc. of the Seventh International Symposium on Operations Research and Its Applications (ISORA2008), Lecture Notes in Operations Research, Vol.8*, pp. pp.351–358 (2008).

[5] 牧祐希, 長江翔: P2P ストリーミング放送システムの改良, 卒業論文, 南山大学数理情報学部 情報通信学科 (2011).