推薦論文

IP Telephonyにおける クライアント依存性を排除した多者間通話サービス

大島浩太[†] 安藤公彦[†]
但馬康宏[††] 寺田松昭[††]

IP Telephony の応用サービスとして多者間通話システムへの関心が高まっている．本研究の目的は，クライアントに特殊な環境を必要とせず，かつ 1 対 1 通話に近い音声 QoS で多者間コミュニケーションを可能とする方式の開発にある．クライアントに特殊な環境を必要としないようにするため，サーバでマルチセッションを処理する方式を採用し，音声ストリームのミキシングに焦点を当てて，サーバの内部処理方式を開発した．提案方式は次の特徴を持つ．(1) 揺らぎをともなう複数の RTP ストリームの時刻同期手法，(2) キュー数の最少化，(3) 波をサンプルごとに重ね合わせるのではなく各サンプルの持つ音声レベルを比較することによって高速な音声ミキシングを行う．提案手法を実装した多者間通話システムのプロトタイプを開発し，性能を評価した．その結果，(1) 音声ミキシング処理を従来方式に比べて約 5 倍高速化できることを実測により確認した．(2) 総合的な遅延時間として 190 ms 以内を達成し，高い音声品質を確保できる見通しを得た．

A Multiparty Call Service in IP Telephony Environment

Kohta Ohshima,[†] Kimihiko Ando,[†] Yasuhiro Tajima[††] and Matsuaki Terada[††]

The purpose of this study is developing a method which does not need special environment for a client and enables multi party call with the voice QoS near 1 to 1 telephone call. A proposed method has the following two features. (1)In order to make it not need special environment for a client, perform a multi-session controlling and mix two or more voice streams by the server. (2)In order to improve voice QoS, perform the time synchronization of two or more streams with jitter, minimize the number of cue and high-speed voice mixing by comparison of a voice level.We developed a prototype of multiparty call system based on proposed method and evaluated performance.As a result, (1)We confirmed a time of voice mixing was accelerable about 5 times compared with the conventional method. (2)We attained 190 ms as total delay and it was proved that high voice QoS was realizable.

1. はじめに

大容量回線の低価格化や，通信業界の規制緩和等により，安価に電話を行うことが可能な IP Telephony サービスの利用者は増加傾向にある．従来の電話は回線交換網を利用しているため，1 対 1 通話が基本の音声通信であり，大手通信会社の提供するサービスであった．

IP Telephony は，IP を利用したパケット交換網で電話を行うサービスである．したがって，電話サービスにパケット交換網の特性を活かすことが可能である．音声データは Web 等のデータと同列に扱われ，また特定のコネクションに回線を占有されない．このことは，従来の電話では実現が困難であった応用サービスの提供を容易にする可能性を有している．特に共有回線であるインターネットを利用する IP Telephony では，個人が応用サービスを提供することも可能である．ところが，IP Telephony ではパケット交換網の利用

† 東京農工大学大学院工学教育部
Graduate School of Technology, Tokyo University of Agriculture and Technology
†† 東京農工大学大学院共生科学技術研究部
Institute of Simbiotic Science and Technology, Tokyo University of Agriculture and Technology

本論文の内容は 2003 年 6 月のマルチメディア，分散，協調とモバイル（DICOMO2003）シンポジウムにて報告され，DICOMO2003 プログラム委員会委員長により情報処理学会論文誌への掲載が推薦された論文である．

による問題も生じる．パケット交換網はデータの配送に特化した網であり，到着時間を考慮した設計になっていない．そのため，データの伝送経路のトラヒック状況により遅延が生じ，特に到着時間に一貫性がない「揺らぎ」が生じる．揺らぎは，リアルタイム性が要求される電話には問題であり，揺らぎを吸収する制御は重要である．また，送話者から受話者へデータが到着するまでに，送話者側では音声サンプリング，エンコード，パケット化といった処理遅延が，伝送経路では伝播距離に応じた伝播遅延やルータにおけるキューイング遅延等が，そして受話者側ではパケットの処理，揺らぎの吸収，受信データのチェック，デコードといった処理時間が必要となる．高い音声 QoS（Quality of Service）を実現するには遅延時間を短くする必要があり[1),2)]，特に ITU-T がリアルタイム双方向通信アプリケーションとして満足できる値として勧告している 150 ms 以内にすることが望ましい[3)]．音質への要求が大きいため，現状では 1 対 1 通話のみ行える IP Telephony サービスがほとんどであり，パケット交換網の特性を活かしたサービスは行われていない．

本論文では，パケット交換網の特性を利用した応用サービスとして，多人数が同時にコミュニケーションを行う多者間通話に関して研究を行う．

多者間通話に類似したシステムである従来の電子会議システムには，参加可能人数が 3〜5 人という制限のある PBX の付加サービス，一般に MCU（Multipoint Conference Unit）と呼ばれている高価な専用機器を必要とするシステム，コンピュータベースでシグナリングに SIP（Session Initiation Protocol）[4)] を用いているもののクライアント環境に専用の会議機能を要求する SipConf [5)] 等が存在する．すべての共通点として特殊な環境が必要になるという点をあげることができ，また会議を前提としたサービスであるため，通常の 1 対 1 通話に比べて遅延時間に対する制約が緩い．

本研究の目的は，クライアントに特殊な環境を必要とせず，かつ 1 対 1 通話に近い音声 QoS で多者間のコミュニケーションを可能とする方式を開発することである．クライアントに特殊な環境を必要としないようにするため，サーバでマルチセッションを処理する方式を採用し，音声ストリームのミキシングに焦点を当てて，内部処理方式の開発を行う．音声 QoS を向上するために，揺らぎをともなう複数ストリームの時刻同期，キュー構造の最適化，高速な音声ミキシング手法の開発を行う．

以下，2 章では多者間通話の定義を述べ，遅延時間の原因となる課題を明らかにし，3 章では 2 章における課題の解決方式を示す．4 章では 3 章の方式を用いて開発したプロトタイプシステムに関して述べ，5 章では実験的評価により提案方式の有効性を実証する．最後に 6 章でまとめを述べる．

2. 多者間通話

2.1 定　　義

本論文では「多者間通話」を，“3 人以上が同時にコミュニケーション可能で，参加者全員が同時発声可能である” 状態として定義している．同時発声者数に制限がある場合は討論等のコミュニケーション形態には適しているが，雑談等の気楽なコミュニケーションには適していない．

2.2 設 計 方 針

IP Telephony の応用サービスとしての位置付けである多者間通話サービスでは，クライアント環境に依存しないサービスであることが要求される．これは，IP Telephony がすでにサービスを開始しているという現状と，ハードウェアベースの専用機や VoIP ゲートウェイ等で従来の電話機を使用している環境では，機能拡張が困難であることに起因している．多者間通話用の専用環境に依存するサービスでは，特定コミュニティ向けのサービスとなる．

特定のクライアント環境に依存しないサービスを構築するためには，クライアントの最低限備えている機能でサービスを受けることができるよう設計することが求められる．IP Telephony クライアントの最低限備えている機能として，次の 2 つの機能がある．

- 1 対 1 の呼を確立可能なシグナリングプロトコルのサポート
- 音声の全 2 重通信機能

つまり 1 対 1 で電話を行うことのできる最低限の機能である．したがって，これらの機能のみを使用することで多者間通話サービスを提供することにより，クライアント依存性を排除することが可能となる．

2.3 サービス提供形態

多者間通話は，サービスを提供するノードで性格が異なる．基本的なサービス提供形態として次の 2 種類（**図 1**）が考えられる．

(1)　クライアントがサービス提供
(2)　サーバがサービス提供

(1) の場合，サービスをクライアントが提供するため，クライアントには「マルチセッションが確立可能なシグナリング」,「複数ストリームの同時処理」といった，サービス提供に必要な機能を，基本的に参加者

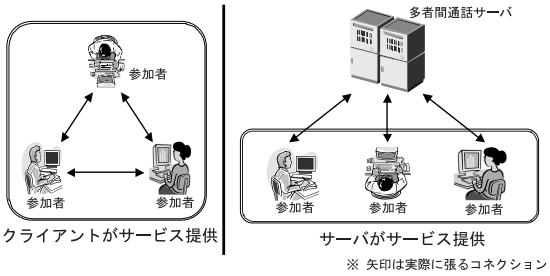

図 1 サービス提供形態
Fig. 1 Service image by teleconferencing server.

全員がサポートしている必要がある．特にマルチセッションを確立するシグナリングに関しては，既存のシグナリングプロトコルの拡張やシグナリング以外の手段で参加者の情報をやりとり[6),7)]する必要がある．そのため，環境に依存したサービスになる．対して (2) の場合，シグナリングの方法の工夫と，やりとりされる音声ストリームをサーバでミキシングすることで，クライアント環境に依存しないサービスにすることが可能である．本論文では，環境に依存しないという設計方針から (2) の形態でサービス提供を行うことにした．(2) の形態も (1) と同様，多者間通話サービスの提供にはマルチセッションを確立する必要がある．そこで，同時に通話を行うグループに対してユニークな 1 つの ID をサーバに割り当て，参加者はその ID に対して電話をかける．サーバでは，同一 ID に電話をかけてきたクライアントを多者間通話の 1 つのグループと見なし，セッション管理を行う．セッション確立後にやりとりされる複数の音声ストリームはミキシング処理をすることで 1 つのストリームに圧縮して送信する．この処理により，クライアントは全 2 重通信機能（上りと下りで 1 ストリームずつ）をサポートしていればよい．実際に確立しているセッションは，クライアントとサーバでの 1 対 1 セッションであるが，サーバでセッション管理やミキシング処理を行うことにより，見かけ上マルチセッションを確立している．また，サーバに複数の ID を持たせることで，1 つのサーバで複数の多者間通話のグループを扱うことが可能である．

2.4 課　　題

多者間通話をサーバで提供し，シグナリングの工夫と音声ストリームのミキシングにより，クライアント依存性は低減した．しかし，サービスをサーバで提供することにより処理内容が増えるため，遅延が増加する．電話にとって遅延の増加は無視できない問題であり，遅延の低減は重要な課題である．遅延時間に関する ITU-T 勧告 G.114 において，0〜150 ms 間では多くのユーザが許容可能，150〜400 ms 間では管理側が考慮していれば許容範囲であり，400 ms 以上は一般的に許容不可と勧告しており，150 ms 以内に抑えることが望ましい．

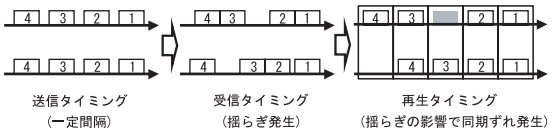

図 2 揺らぎによるストリームの同期ずれ
Fig. 2 A synchronous gap of the streams by jitter.

多者間通話サーバでは，次の流れで処理を行う．

(1) 受信したパケットの整合性チェック
(2) 揺らぎ吸収バッファへのデータ格納
(3) 複数ストリームの時刻同期処理（ミキシング可能なデータの判別）
(4) データのミキシング
(5) ミキシングデータの送出

主に (2)(3)(4) での処理遅延が大きい．特に，(2) の処理では，各々のストリームには独立した揺らぎが生じているため，最初にすべてのストリームの揺らぎを吸収可能な時間を待機しないと，(3) の処理で意図する結果を得ることが難しくなる[8)]．そうなると，揺らぎ吸収バッファにある程度多くのパケットを貯めておくことで問題は解決すると思われるが，遅延の増加により音声 QoS の低下を引き起こす．

RTP パケットはクライアントの利用している音声 Codec によりサイズ，録音時間は異なる．音声 Codec に G.711 を利用するクライアントでは，固定長 160 バイト（20 ms 録音）のデータが 20 ms 周期で送信される．したがって，許容可能な 150 ms という遅延時間は，RTP パケット 1 つの録音時間から計算するとパケット 7.5 個分という少ない数であるので，揺らぎの影響を緩和するのに十分な数をバッファに貯めることは難しい．さらに，揺らぎは送信タイミングと受信タイミングをずらすため，同期ずれ（**図 2**）が生じ，クライアントでの再生音声が間延びや途切れを起こす可能性もある．

また，(2)(3) の処理では，パケットをバッファに貯める処理や，バッファから取り出す処理を行うため，キューイング遅延が生じる．キューイング遅延は処理内容にもよるが，キューの数に応じて大きくなるため，キューの数を可能な限り減らす構造にすることが望ましい．

特に (3) の処理は課題が多い．時刻同期処理は複数のクライアントから同時刻に送出された音声データどうしを時刻同期済みデータとすることが望ましい．し

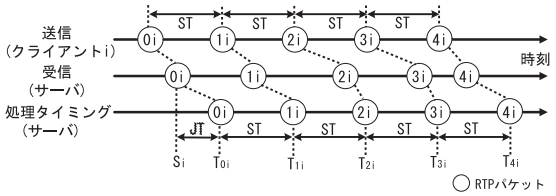

図 3 RTP ストリームの再生タイミング
Fig. 3 Play timing of a RTP streams.

かし，揺らぎの影響で音声データの到着時刻は不規則であり，また，クライアントどうしの時計をミリ秒レベルで同期することが不可能なことから，実時間に準拠した処理は難しい．さらに，ストリーム伝送に広く用いられている RTP（Real-Time Transport Protocol）[9] には帯域節約を目的とする「無音制御」がある．無音制御とは，一定時間音声入力がない場合に，再生タイミングを制御するタイムスタンプのみ増加させ，実際に RTP パケットを送出しない制御である．無音制御が生じた場合，受信側では RTP パケットが到着しない理由が単なる遅延なのか無音制御なのか判断できず，時刻同期処理には問題となる．本論文では，広く利用されているという背景から，ストリーム伝送プロトコルに RTP を採用している．したがって，無音制御による問題を解決する必要がある．

(4) のミキシング処理では計算時間がかなり必要になる．本論文で使用する，公衆電話網と同等の品質である音声 Codec: G.711（サンプリング周波数：8,000 Hz，符号化ビット数：8 ビット）の場合，2 つのストリームをミキシングするためには 8,000 回/秒の処理が必要になり，参加者数に応じてさらに大きくなる．最大で 8000*(“参加者数”−1)*“参加者数” 回/秒の処理を必要とする．

3. 実 現 方 法

3.1 ストリームの時刻同期

共通する時間軸として多者間通話サーバの内部時計を用い，無音制御による問題の解決のため RTP のタイムスタンプ情報を利用することで，複数ストリームの時刻同期処理を実現した．

ミリ秒レベルの時刻同期処理に利用可能で，かつ各参加者に共通の時間情報は “サーバのローカルな時刻” のみである．そこで，各ストリームの RTP パケットについて，「揺らぎ吸収バッファ内の RTP パケットが処理されるべき時刻であるか否かの判別」をすることで時刻同期処理を行う．

図 3 にクライアント “i” からの RTP パケット送信タイミング，サーバでの受信タイミングと処理すべきタイミングの関係を示す．図 3 は処理タイミング T_{0i}〜T_{4i} を示している．初期値 T_{0i} は，揺らぎの影響をなくすため，サーバで最初に受信した時刻 S_i に揺らぎ吸収バッファサイズ JT を加算した時刻とする．以降の T_{1i}〜T_{4i} は，音声 Codec に G.711 を用いているため，RTP パケットは固定周期でクライアントから送信されていることから，周期 ST になる．

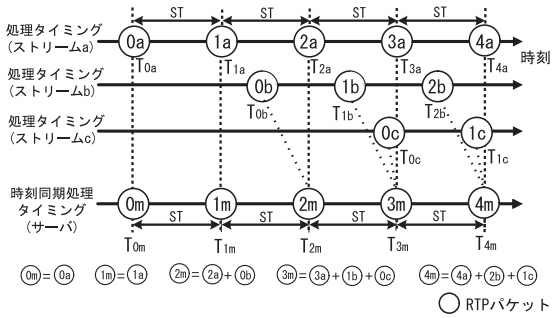

図 4 複数ストリームの時刻同期
Fig. 4 A time synchronizing of RTP streams.

図 4 に，3 つのストリーム a，b，c の処理タイミングと時刻同期タイミングの関係を示す．図 4 は，時刻同期処理タイミング T_{0m}〜T_{4m} を示している．初期値 T_{0m} は，最初に開始されたストリーム a における処理タイミングの初期値 T_{0a} とする．以降の T_{1m}〜T_{4m} は周期 ST になる．ある RTP パケットが時刻同期処理をされるべきかどうかは，ストリーム “i” における “j” 番目の RTP パケットの処理時刻 T_{ji} が，n 番目の時刻同期処理タイミング T_{nm} より前であるかどうかを判別することで行う．これは，RTP パケットは固定周期で送出されているものの，各ストリームの開始時刻は異なり，また他のストリームの処理タイミングと開始時刻は同じになる保証がないことに起因する．T_{ji} は，$T_{ji} = S_i + JT + j * ST$ で導出する．以上から，RTP パケットが処理されるべき時刻である場合，次の不等式が成り立つ．

$$T_{ji} \leq T_{nm} \tag{1}$$

j 番目の RTP パケットについて n 番目の処理タイミングにおいて不等式 (1) が成り立つ場合，j 番目の RTP パケットは同期済みと判別し，成り立たない場合は，まだ処理されるべきでない RTP パケットとして，n+1 番目の時刻同期処理タイミングを待つ．図 4 では，処理タイミング T_{0m}，T_{1m} では開始しているストリームが a のみであるので，同期済み RTP パケット $0m$，$1m$ はそれぞれ $0a$，$1a$ となる．T_{2m} では，ストリーム b が開始しているので，$2m$ として $2a$+$0b$ が同期済み RTP パケットとなる．同様に，T_{3m}，T_{4m} では，

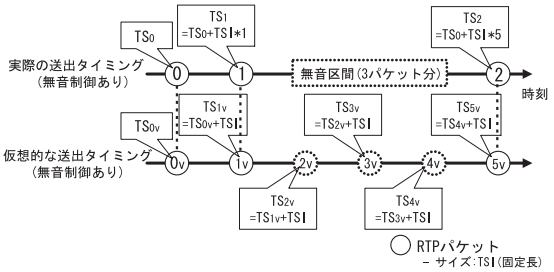

図 5 RTP タイムスタンプの推移
Fig. 5 The value of RTP timestamp with silent control.

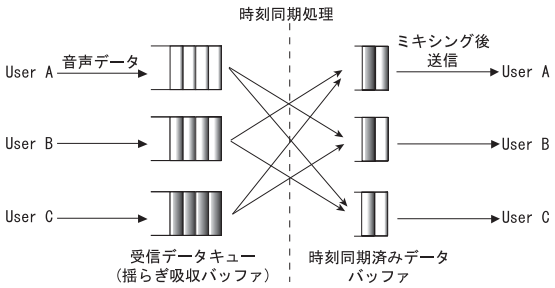

図 6 キュー構造
Fig. 6 Queue structure.

ストリーム c が開始しているので，$3m$ は $3a+1b+0c$，$4m$ は $4a+2b+1c$ となる．

無音制御による処理タイミング計算への影響を**図 5** を用いて説明する．図 5 は，無音制御が生じている場合の実際の送出タイミングと，同タイミングで無音制御時に RTP パケットを送信していると仮定した場合のタイムスタンプ推移を示している．RTP タイムスタンプは送出したパケットサイズの累計として表現されている．実際の送出タイミングは，無音制御が発生している場合のタイムスタンプの推移である．まず 0 番目のパケットのタイムスタンプを TS_0 とする．次に 1 番目のパケットには無音制御が発生していないため，タイムスタンプ TS_1 は増加量 TSI を加算した値 $TS_0+TSI*1$ となる．次に 2 番目のパケットは 3 パケット分の無音区間の後であるため，タイムスタンプ TS_2 は $TS_0+TSI*5$ となる．仮想的な送出タイミングは，実際の送出タイミングの無音区間に RTP パケットが送出されていると仮定した場合のタイムスタンプ推移である．実際の送出タイミングと区別するため，送出順には v を付けている．まず，0v 番目のタイムスタンプ TS_{0v} を TS_0 とする．次に 1v 番目のパケットは $TS_{0v}+TSI$ となる．そして実際の送出タイミングでは無音区間である 2v，3v，4v 番目の RTP パケットのタイムスタンプはそれぞれ TSI ずつ増加し，5v 番目の RTP パケットのタイムスタンプ TS_{5v} は $TS_{4v}+TSI$ となり，TS_2 と等しくなる．

ここで，図 3 から，ストリーム i における j 番目の RTP パケットの処理タイミング T_{ji} は，初期値を T_{0i} とし，周期 ST で増加するため，$T_{ji}=T_{0i}+j*ST$ と表現できる．しかし，無音制御をともなう実際の送出タイミングで送出された RTP パケット数を j として T_{ji} を計算することはできない．無音制御の影響で，j 番目のパケットは途切れなく周期 ST で送出されている保証がなく，T_{ji} は意図する値より小さな数値になる可能性がある．そこで，タイムスタンプから途切れなく周期 ST で RTP パケットが送出された場合の仮想的なパケット数 k を求めることでこの問題を解決する．

問題の原因は「周期的に送出された状態で何番目のパケットか分からない」ことである．したがって，「無音が生じなかった場合に送出されたパケット数」が分かれば問題は解決する．仮想的な送出タイミングが示すように，無音区間でも RTP タイムスタンプは一定増加していることを利用すると，k は j 番目の RTP タイムスタンプ TS_{ji}，初期値 TS_{0i}，増加量 TSI から，$k=(TS_{ji}-TS_{0i})/TSI$ となる．k から T_{ji} は式のように表現できる．

$$T_{ji}=T_{j0}+\{(TS_{ji}-TS_{0i})/TSI\}*ST \quad (2)$$

式 (2) を用いて処理タイミングを計算することで，無音制御による問題が解決する．

提案手法は RTP パケットの処理されるべき時刻の計算に，各ストリームが各々保持している情報のみを用いる．そのため，他のストリームの RTP パケット受信状況に依存することなく処理することが可能である．したがって，複数ストリームの同期処理を簡単にすることができ，多者間通話サービスのような動的に参加者が変動するサービスにおけるストリーム同期手法として有効である．

3.2 キュー構造

多者間通話サーバで必ず必要となるキューに，クライアントからの RTP パケットを格納するための受信データキュー（揺らぎ吸収バッファ）がある．これは，参加者数と同じ数だけ確保する．サーバでは，受信，ストリームの時刻同期，ミキシング，送出という流れになるため，ストリームの時刻同期済み RTP パケットでキューを作成すると制御が容易になる．しかし，キューの数が増えるため，キューイング遅延が増加する．

そこで，本論文では**図 6** で示すキュー構造を採用した．このキュー構造では，まず受信した RTP パケットを受信データキューに格納し，揺らぎを吸収する．

次に，任意のタイミングで行う時刻同期処理により，現在時刻において処理されるべき RTP パケットをそれぞれ受信データキューから直接取得する．このとき，参加者自身が送信した RTP パケットは同期処理の対象としない．最後に同期済み RTP パケットはミキシング処理を行った後送出される．このキュー構造では，受信データキューのみになり，キューイング遅延の発生を最低限に抑えることが可能である．

3.3 音声ミキシング

従来手法である波形の重ね合わせは 2.4 節で記述したように処理回数がかなり多く，処理遅延の原因になる．そこで本論文では，処理回数を低減可能な新たなミキシング手法を考案した．考案手法は，次の 4 つの特徴を利用している．

(1) 聴覚における不完全な波の補完機能
(2) 聴覚における合成波の分解機能
(3) 小さな音は大きな音でかき消される
(4) 標本化されたデジタル音声は，隣り合うデータとの離散値差が極小である[10)]

(1)(2)(3) の特徴を利用することにより，デジタル音声の数値差を比較し，比較結果に応じてデータを組み替えることでミキシングが可能である（**図 7**）．データの組替え処理は，まずミキシング対象波形#1, #2 についてサンプルを取り出し，横軸からの絶対値を計算する．次に導出した絶対値を比較し，大きな方を「強い値」と見なす．最後に，強い値になったサンプルをミキシング結果波形のサンプルとして適用する．サンプルの処理が終わると 2 番目のサンプルについて同様の処理を行い，この操作をサンプルすべてについて行う．

この手法により，重ね合わせ処理が比較処理に置き換わる．また，(4) の特徴を利用することにより，任意のサンプルどうしの比較結果を次のサンプルへ高確率で適用することが可能である．4 つ先のサンプルまで適用した場合の計算量は，従来手法の 25% になる．これはかなり粗い制御のようだが，8 kHz サンプリングにおける 1 つのサンプルあたりの録音時間は $1.25 * 10^{-4}$ 秒であるのでミクロな制御になる．従来の波形重ね合わせによるミキシングによる波形と，考案手法による波形では，視覚で違いを識別することが困難であった．聴覚的な面でも，音声と音楽を重ね合わせた場合，音声と音楽がともに容易に聞き取れる程度の品質は確保できている．また，3 つ以上のデータをミキシングする場合には比較処理を同時に行えばよいため，1 度の計算でミキシングを行える．したがって本手法は，2 つ以上の音声をミキシングする多者間通話サービスに適した手法でもある．

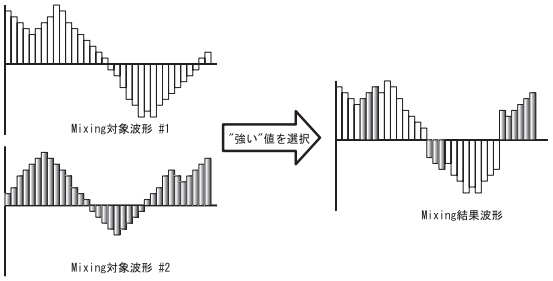

図 7 データの比較・組換えによるミキシング
Fig. 7 Mixing by comparison and recombination of data.

従来手法によるミキシングは波の重ね合わせを前提としており，「2 つ以上の音波が同時に媒質中の 1 点に来たとき，その点における変動はそれぞれの音波による変動をベクトル的に加え合わせたものである」という音波の重ね合わせの原理により実現している[11)]．したがって，従来方式によるミキシングは可逆変換であり，ミキシングされた音声から特定の音声を減算することが可能である．

これに対し提案手法は音声レベルの大きさの比較により実現しているため，従来手法よりミキシングに必要な計算量が少ない．しかし，単純な比較処理のみでは従来手法と同様，1 秒間に音声のサンプリング周波数と等しい回数の処理を行う必要がある．そのため，連続する音声サンプル間の差が小さいことを利用し，比較処理結果を次のいくつかのサンプルの比較結果に適用することで計算量の低減を行っている．また，付加価値として，従来手法ではミキシング結果にノイズも含まれる．一方，提案手法では “ノイズ=小さな音” と見なすことができ，音声とノイズの重なっている部分ではノイズを除去する効果がある．しかし，提案手法は音声レベルに応じてデータを組み替えるだけのため，不可逆変換になる．つまり複数の音声をミキシングした音波から特定の音波を減算により消去することができない．そこで，不要な音声はミキシング対象から外す必要がある．

4. プロトタイプの概要

前述の方式の有効性を実証するため，方式を実装した多者間通話システムのプロトタイプを開発した．

本システムは，シグナリング処理スレッド，ストリーム受信スレッド，ストリーム処理・送信スレッド，の 3 種類の機能で構成している．

シグナリングスレッドは，VoIP の代表的なプロトコルである SIP を利用している．クライアントから

表 1 評価システムの仕様
Table 1 Specification of an evaluation system.

項目	仕様
OS	Windows2000 Professional SP2
CPU	Intel PentiumIII 1 GHz
Memory	512 MB
LAN	10Base-T
Protocol	シグナリング：UDP/IP，音声：UDP/IP

接続要求があった場合，セッションを確立するために必要な情報を交換し，ストリーム受信スレッドとストリーム処理・送信スレッドを 1 つずつ起動する．その後，希望する多者間通話のグループに対するセッションを確立する．

ストリーム受信スレッドは，クライアントから送信された RTP パケットの整合性をチェックし，受信データキューに格納する．

ストリーム処理・送信スレッドは，1 つの RTP パケットにおける録音時間と同じ周期で受信データキューを走査し，処理すべきデータが存在すればミキシング処理を行う．このスレッドでは，送出するデータに適切な RTP ヘッダを付与することも行う．

5. 評　　価

本章では，作成したプロトタイプに関して，スケーラビリティを測る指針として CPU 負荷，音声 QoS を測るための指針として MOS 値および総合遅延時間に関して評価を行った．評価用クライアントとして Microsoft の RTC API で作成したクライアントソフトウェアを用い，Artiza VoIP Analyzer（株式会社アルチザネットワークス）を用いて MOS 値を測定することにより音質を測定した．評価に用いた実験機器を**表 1** に示す．

5.1 CPU 負荷

図 8 は，参加人数に応じた CPU 負荷率の平均値の推移を示したグラフである．横軸は参加人数を表し，縦軸は CPU 使用率を表している．

評価は，クライアントを 1 台ずつ，計 8 台サーバに接続し，つねに音声パケットをサーバに送信している状態で計測を行った．これは無音制御を考慮せず，つねに有音の状態を想定しているため，つねにミキシングを行った際の CPU 負荷率である．結果から，参加者数が 2 から 4 人の場合は緩やかな線形的上昇であるが，5 人以上になると上昇率が増えている．これは，参加者の増加にともないアクティブなスレッド数が増加し，スレッドのスケジューリングおよび排他制御等により CPU 資源を消費しているものと推測する．

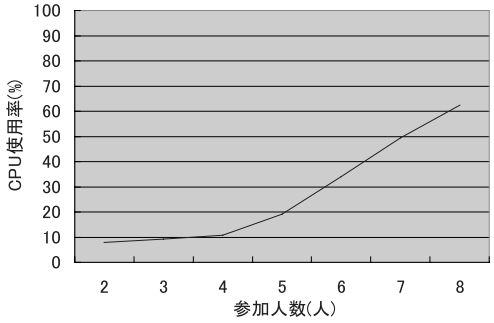

図 8 CPU 使用率の平均
Fig. 8 Load average of CPU.

表 2 ミキシング時間の比較
Table 2 Comparison of the mixing time.

従来手法	考案手法
1,468.7(ms)	301.5(ms)

また，今回の試作サーバを用いて安定したサービスが可能な人数は，負荷が 60〜70%台である 8 人程度である．

5.2 ミキシング速度

本論文で提案した，サンプルの音声レベル比較によるミキシング手法と，波の重ね合わせによる従来のミキシング手法[11] に関して計算時間の比較を行う．

比較は次の手順で行った．考案方式は 1 サンプルの比較結果を 3 つ先のサンプルまで適用している．

- 2 つの 160 Byte のデータに対し 100 万回処理を行った際の所要時間を計測．
- 上記を 20 回行い，平均値を評価値とする．

結果は**表 2** のようになり，考案手法は従来手法の 20.5%の時間でミキシングが可能であることを示している．

5.3 音　　質

音質評価は回線品質を評価する R 値と，音質の主観評価である MOS 値（特に日本人に特化した MOSj 値）を用いた．

評価は，3 台のクライアントがサーバに接続し，会話を 1 分行った結果の平均値をとることで行った．音声 Codec は G.711 を利用している．その結果，**表 3**，**表 4** で示す結果が得られた．

結果から，R 値はアナログ電話に近い回線品質で，MOS 値は “普通（多少の努力が必要）” であり，MOSj 値は “良い（通常の会話には特に支障がないレベル）” であった．

5.4 総合遅延時間

今回使用した評価環境では，開発サーバ以外で遅延

表 3 音質評価結果
Table 3 Evaluation result of voice quality.

項目	概要	結果
Network R Factor	Codec の種類，Network によるパケットロス，揺らぎ等から算出した R 値（評価基準[13]−80 以上：アナログ回線並，70 以上：携帯電話並）	81.2
User R Factor	Codec の種類，パケットロス，遅延，バースト等から算出した R 値	76.6

表 4 音質評価結果
Table 4 Evaluation result of voice quality.

項目	概要	結果
Conversational Quality MOS	会話 MOS 値（評価基準[14]−5：非常に良い，4：良い，3：普通，2：悪い，1：非常に悪い）	3.14
Conversational Quality MOSj	Conversational Quality MOS 値から換算した，日本人に特化した MOS 値[12]	3.6

時間を正確に測定することができない．そこで，サーバ以外での遅延時間を推測することで総合的な遅延時間を算出することにした．サーバ以外で必要となる処理時間を**表 5** に示す．実験は研究室内の LAN 環境で行っているため，Round Trip Time が小さい．そこで，キャンパスネットワークを用いて遅延時間の計測を行った．ホップ数 6 の対象に対して ping ツールで Round Trip Time を 200 回計測し，それらの平均値を求めた．経路上の機器は，FireWall や L2 スイッチである．結果，平均 Round Trip Time は 5.8 ms であった．また試行の 97%は 3 ms 以内，残り 3%は 10 ms 以上（特に大きなものでは 75 ms）という計測結果であった．

次に，サーバで計測した処理時間を**表 6** に示す．計測は 3 台のクライアントをサーバに接続し，3 分間データを送信しつづけた状態を実測した．これは，帯域が上り：192 kbps, 下り：192 kbps であり，RTP の無音制御を考慮せずつねにミキシングを行う状態である．その他の要因で 50 ms かかっていることが実測された．これは，スレッドのスケジューリング，OS のタイマの分解能，ネットワークインタフェースにおける負荷によるものと考えられる．総合遅延時間は表 5，表 6 の合計値であり，185 ms であった．表 5 の Round Trip Time にキャンパスネットワークで計測した 5.8 ms を適用した場合でも合計値は 190 ms 以内であり，高い QoS を実現できていると言える．

表 5 サーバ以外による遅延時間の推測値
Table 5 The guess value of delay time except a server.

要因	時間 (ms)
Round Trip Time	0.35
音声サンプリング	22
揺らぎ吸収	60
合計	82.35

表 6 サーバにおける遅延時間
Table 6 Delay time in a server.

要因	時間 (ms)
揺らぎ吸収	40
時刻同期	5
音声ミキシング	8
その他	50
合計	103

6. ま と め

本研究では，多者間通話サービスをサーバで提供する方式において，クライアントに特殊な環境を必要とせず，かつ 1 対 1 通話に近い音声 QoS で多者間のコミュニケーションを可能とする方式を開発した．開発方式は，高い音声 QoS を実現するための，(1) 揺らぎをともなう複数の RTP ストリームの時刻同期手法，(2) キュー数の最少化，(3) 波をサンプルごとに重ね合わせるのではなく各サンプルの持つ音声レベルを比較することによって高速な音声ミキシングを行うことを特徴とする．提案方式に基づいて多者間通話システムのプロトタイプを開発し，音質を評価した．その結果，MOSj 値 3.6, 総合的な遅延時間として 190 ms 以内を達成し，ユーザに違和感のない会話を可能にした．

今後は，さらに揺らぎの大きいインターネットで利用する際のストリーム同期手法の研究，遅延時間のさらなる低減とスケーラビリティの向上を目指した研究を行っていく．

参 考 文 献

1) Melvin, H. and Murphy, L.: Time Synchronization for VoIP Quality of Service, *IEEE INTERNET COMPUTING*, MAY/JUNE, pp.57–63 (2002).
2) 星 徹，谷川桂子，松井 進，石見直子，寺田松昭：LAN 環境における負荷適応制御を用いた低遅延リアルタイム音声通信システム，情報処理学会論文誌，Vol.40, No.7, pp.3063–3073 (1999).
3) ITU-T Recommendation G.114: One Way Transmission Time (1996).

4) Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, RFC3261 (2002).
5) Singh, K., Nair, G. and Schulzrinne, H.: Centralized Conferencing using SIP, *IP Telephony Workshop* (2001).
6) Miladinovic, I. and Stadler, J.: SIP Extension for Multiparty Conferencing, *IETF Internet Draft* (2002).
7) Miladinovic, I. and Stadler, J.: Multiparty Conference Signaling Using the Session Initiation Protocol (SIP), International Network Conference (2002).
8) 清末悌之，湯田佳文：IP ネットワーク上のリアルタイム音声ミキシングに対してバッファサイズが与える影響に関する一考察，情報処理学会論文誌，Vol.41, No.10, pp.2742–2751 (2000).
9) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, RFC3550 (2003).
10) Kientzle, T.: *A Programmer's GUIDE TO SOUND*, Addison Wesley, p.464 (1997).
11) 小橋　豊：音と音波，p.226, 裳華房 (1969).
12) TTC 標準 JJ-201.01：IP 電話の通話品質評価法，社団法人情報通信技術委員会 (2003).
13) ITU-T Recommendation G.107: The E-Model, a computational model for use in transmission planning (2003).
14) ITU-T Recommendation P.800: Methods for subjective determination of transmission quality (1996).
15) Baldi, M., Risso, F. and di Torino, P.: Efficiency of Packet Voice with Deterministic Delay, *IEEE Communications Magazine*, pp.170–177 (2002).
16) Bessler, S., Nisanyan, A.V., Peterbauer, K., Pailer, R. and Stadler, J.: A Service Platform for Internet-Telecom Services using SIP, *Publication to conference Smartnet* (2000).
17) Camarillo, G.: *SIP Demystified*, p.320, McGraw Hill (2001).
18) Davidson, J. and Peters, J.（著），風工舎，シスコシステムズ（訳）：VoIP 基本ガイド，p.399, ソフトバンクパブリッシング (2001).

(平成 15 年 12 月 4 日受付)
(平成 16 年 9 月 3 日採録)

推　薦　文

本論文は，IP パケット交換網上で多者間音声通話を可能にする新規な IP Telephony サービスについての提案である．IP Telephony サービスは昨今その需要が急速に高まっており，この技術を遠隔会議用途に適用できるようにする本提案の価値は高い．また，音声合成時の CPU 負荷を抑制する独自の音声合成方式を提案し，実際にシステムを実装したうえで，音質・遅延時間・CPU 負荷等の各種性能を定量的に評価しており信頼性が高い．テーマ的に重要で有用性が高く，かつ，信頼性も高いことから技術論文として評価できる．（DICOMO2003 プログラム委員会委員長　高橋　修）

大島　浩太（学生会員）

昭和 53 年生．平成 15 年東京農工大学大学院工学研究科電子情報工学専攻博士前期課程修了．同年東京農工大学大学院工学研究科電子情報工学専攻博士後期課程進学．現在同大学院工学教育部電子情報工学専攻博士後期課程在学中．コンピュータ・ネットワークの研究に従事．

安藤　公彦

昭和 54 年生．平成 15 年東京農工大学大学院工学研究科電子情報工学専攻博士前期課程修了．同年東京農工大学大学院工学研究科電子情報工学専攻博士後期課程進学．現在同大学院工学教育部電子情報工学専攻博士後期課程在学中．コンピュータ・ネットワークの研究に従事．

但馬　康宏（正会員）

1994 年電気通信大学電気通信学部電子情報学科卒業．1996 年電気通信大学大学院電気通信学研究科博士前期課程修了．同年石川島播磨重工業（株）入社．2001 年電気通信大学大学院電気通信学研究科博士後期課程修了．同年東京農工大学工学部情報コミュニケーション工学科助手．2004 年東京農工大学大学院共生科学技術研究部システム情報科学部門助手．現在に至る．博士（工学）．ネットワークにおける知的処理の研究，計算論的学習理論の研究に従事．EATCS，電子情報通信学会，人工知能学会各会員．

寺田 松昭（正会員）

1970 年岡山大学工学部電気工学科卒業．同年（株）日立製作所入社．同社システム開発研究所において，制御用分散処理システム，LAN，プロトコル高速処理，VoIP，次世代インターネットの研究に従事．工学博士．著書『制御用計算機におけるリアルタイム技術』（共著，コロナ社），『デジタルサービス革命』（共著，日刊工業新聞社）．1999 年 4 月より東京農工大学工学部情報コミュニケーション工学科教授．IEEE，ACM，電子情報通信学会各会員．