

LANとISDN回線間の音声通信

篠田 晃 新井克也 石井一彦 桑名栄二*

*NTTソフトウェア研究所
NTT

最近ではコンピューターネットワーク上での音声／映像通信も可能となってきた。一方、音声通信として主流である電話は電話網という環境に閉じて使用されている。今回、我々はISDN回線を使用して、公衆回線上の電話通信とコンピューターネットワーク上の音声通信の統合環境を開発した。本論文では、ISDN回線とコンピューターとのデータインターフェース方法、このインターフェースを使った実験報告さらに応用例について述べる。

1. はじめに

最近ではコンピューターネットワーク上で音声／映像通信も可能となり、インターネット上でもmboneを使ってスペースシャトルの打ち上げの模様やETFのミーティング、さらにはローリングストーンズのコンサートが送られている。実際にはQOS制御等の研究課題はあるものの実用となりつつある。

音声通信としては古くから電話がある。最近では高性能な通信サービスを提供するために、電話機やPBXなどの通信機器コンピューターでコントロールする方式やインターフェースなどが提案されている[1][2]。しかし、これらは、電話に対するコントロールとしてコンピューターを使用するということであり、通信を行うのは電話同士ということになっている。また、オフィスではSISやOAのためコンピューターが設置されているが、音声通信は電話を使用するという状況である。つまり、コンピューターと電話という使い分けがされているのである。しかし、前述のようにコンピューターネットワーク上での音声通信は実用に近いレベルまででき、音声と比較的に簡単にデジタル処理できるようになってきている今日では、電話との通信も不可能ではないはずである。

そこで、今回我々は、ISDN回線が音声をデジタルデータとして転送していることに着目して、コンピューター上の音声通信と公衆回線

上の電話による音声通信間のシームレス環境を構築した。具体的にはISDN回線上のデジタル化した音声信号をコンピューターへ取り込むインターフェース（ハードおよびソフト）を開発し利用することによりシステムを構築した。

過去においては、BellcoreのStephen UhlerがPhoneStation [3]を作成し実験している。これは電話回線に対してアダプターボックスをリアルインターフェースや外部音声入出力を介して接続し、PhoneScriptというシェルを利用することにより電話としてのコントロール／音声入出力を行っていた。これはコンピューターの電話化を行ったことになる。これに対し、我々はコンピューター環境と電話環境のシームレス化を目標とした。そのため、コンピューターに電話機能を付加することから、PBXのような機能を所有するゲートウェイを実現すること、さらに電子メールと音声通信のゲートウェイを実現すること等の広い応用範囲をカバーするため、UNIX上でのインターフェースとして実現し、C言語ライブラリーとコンピューターへの内臓カードとして実現した。

本論文では、ISDN回線とコンピューターとのデータインターフェース方法、このインターフェースの可能性を確かめるために行った実験の報告さらに応用例について述べる。

A seamless audio communication system between LAN and telephones

Akira Shinoda, Katsuya Arai, Kazuhiko Ishii, Eiji Kuwana *

* NTT Software Laboratories, NTT

2. ISDN回線上の音声データ

ISDN回線は制御チャンネルであるDチャンネルとデータチャンネルであるBチャンネルより構成され、BRIの場合、2B+Dというチャンネル構成である。Bチャンネルには、それぞれサブアドレスが付与され独立に使用することが可能である。Bチャンネル上で音声を送信する場合は、音声信号をA/D変換しデジタル信号として伝送する。

A/D変換時の仕様は以下の通りである。

符号化方式	μ -Law
サンプリング周波数	8000Hz
精度ビット数	8bit
チャンネル数	1

符号化方式は日本、アメリカ等では μ -Law方式を使用し、ヨーロッパ等ではA-Law方式を使用している。

ISDN回線はシリアルラインであるため、データはシリアルデータとして転送される。

3. ISDN回線インターフェース

2章で述べたようにISDN回線の音声データはデジタル信号である。そのためインターフェースを仲介しコンピューターと入出力することが可能となる。ただし、ISDN回線上のデータはシリアルデータであるため、コンピューター上で処理を行うためにパラレルデータに変換することが必要になる。

3. 1 ハードウェア

ハードウェア構成の概要を図1に示す。機能としては、ISDN回線制御、ISDN回線レベルコンバーター、シリアル/パラレルデータ変換、によりISDN回線とコンピューターのデータ入出力を行う。

入出力にはハードウェアバッファを介している。方式はFIFOである。バッファによりデレイが発生するため、バッファサイズは重要なチューニングファクターとなる。

3. 2 ソフトウェア

ソフトウェアは、デバイスドライバー、API、入出力ドライバーの3つよりなる。

・デバイスドライバー

インターフェースをドライブするためのものでデーモンとして起動する。

・API

インターフェースに対するAPIであり、インターフェースのオープン/クローズ、チャンネルのオープン/クローズ、発信/着信のようなISDN回線の制御を行う。できるだけ電話の動作シーケンスを直接実行する基本的な機能単位とし、C言語ライブラリーとして実現した。

・入出力ドライバー

ISDN回線からの音声データの入出力インターフェースである。UNIXのデバイスファイルとして実現した。そのため、UNIXシステムコールのread/writeを使用することができる。ブロック/ノンブロックの両モードにて使用可能である。

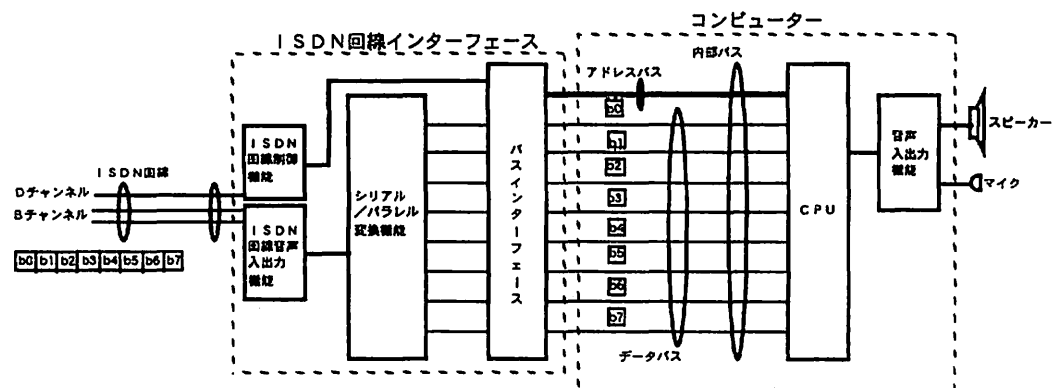


図1 ハードウェア構成の概要

4. ワークステーションへの 電話機能の付加

3章で述べたハードウェアとAPIを使用することによりコンピュータで電話機能を実現することができる。

APIは基本機能単位にC言語関数となっているため、必要に応じた電話の制御シーケンスを組み立てることが可能となっている。ここでは一般的な電話を想定する。

(1) 電話機能のインプリメント

電話の制御は、人間による電話の操作シーケンスと回線状態の2つの状態を組み合わせた状態遷移によって行う。そこで、まず操作シーケンスが必要になる。これは一般的に図2のようなシーケンスになる。

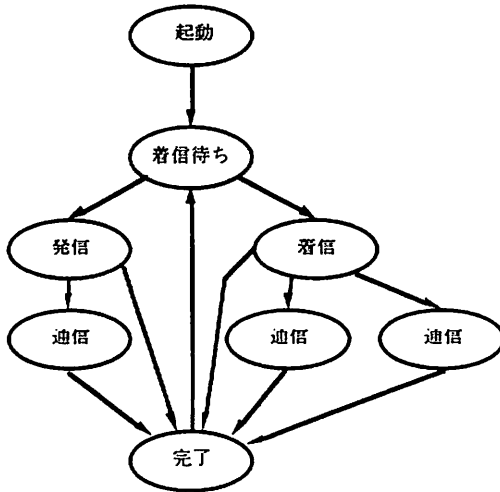


図2 電話の動作シーケンス

次に、回線状態が必要である。回線状態はX.31[4]に従った状態を回線状態取得関数を使用して取得する。

音声データの入出力は、ISDN回線インターフェース側ではUNIXのデバイスファイルとしているため、UNIXシステムコールのread/writeを使うことより入出力を行う。例えば、Sun Sparc StationのスピーカーにB1チャンネルの音を出力するには図3のようになる。ISDNのデバイスからの出力はサンプリング周波数に同期して行われるので、ブロッキングモードで使用するにより、プログラム上での同期が不用となる。

また、コンピュータ側での音声のデコードをISDN回線での仕様と同じにすることができるなら、図3に示すようなread/write動作のみで済むため、デコード時におけるオーバーヘッドを少なくすることができる。

```
isdn_fd = open (/dev/audio_b1, 0_RDWR);
audio_fd = open (/dev/aidio, 0_RDWR);

n=read (isdn_fd, buff, 1024);
write (audio_fd, buff, n);
```

図3 音声データの受け渡し例

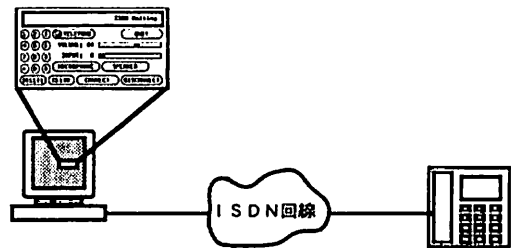


図4 ワークステーションと
ISDN電話の通信

(2) バッファサイズと音声の連続性 および音声遅延について

音声を再生する場合、デコーダーでISDN回線と同じ仕様でデコードできるなら、ISDN回線インターフェースからデコーダーへデータをバケツリレーのように転送すればよい。これを図5に示し、例として述べる。

・音声の連続性

Bチャンネルのスピードは64 kbpsであるので、8 kByte/sでデータを取り込めれば音切れがなく再生ができる。

図5において、ハードウェアであるISDN回線インターフェースとデコーダーは入力データを逐次処理するため連続性が保たれる。

データを転送するソフトウェア部分ではバッファを介してread/writeの繰返し処理を行う。この繰返し処理において8 kByte/sのデータ転送を保持しなければ、音声の連続性を保つことができない。ここでの時間は仲介しているバッファサイズに依ることになる。

バッファサイズds (Byte) に対してのデコードに必要な時間tdは次式であり、

$$td = \frac{ds}{8000} \text{ (sec)}$$

バッファサイズとデコードに必要な時間の関係は図6のようになる。

これより、例えば512Byteのバッファを使用するなら、64ms以下にループの処理時間を設計しなければならないことがわかる。

・音声遅延

図5においてバッファリングは①～③の3回行う。②はサンプリング時間内での処理となり、③は前処理からの出力を逐次的に受けるため遅延は問題ない。実際に遅延に関わるのは、ISDN回線インターフェース内のバッファリングであり、この大きさによって遅延時間が決まる。遅延時間は前出の式で求めることができる。そのため、実用上問題のない遅延時間に収めるためバッファサイズを調整する必要がある。

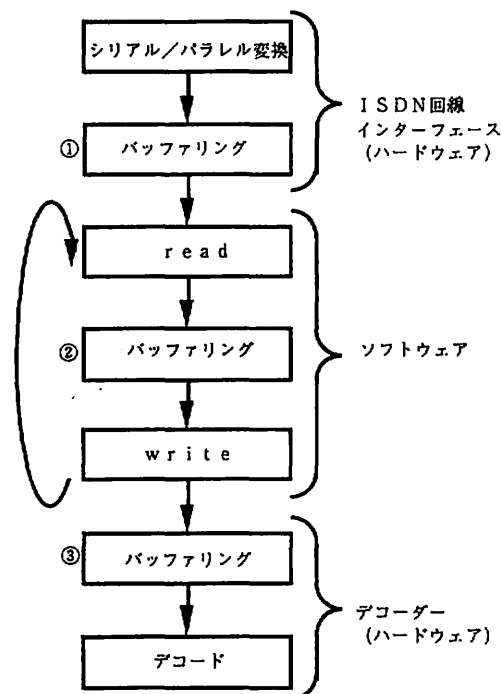


図5 ISDN回線からの音声データのデコードシーケンス

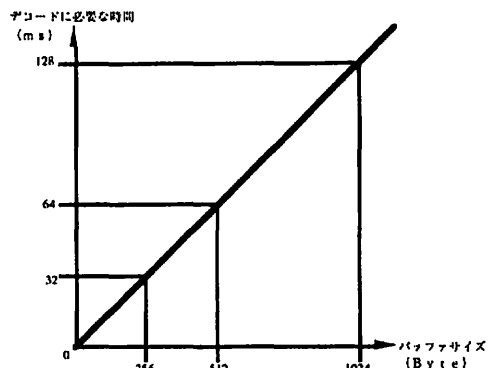


図6 バッファサイズとデコードに必要な時間

5. LANとISDN回線間の音声通信

コンピュータネットワーク上の音声通信ツールとしてはvat[5]、nevo[6]等があり、我々も自作の実験ツール[7]を使って実験を進めている。

4章で述べたツールの出力をワークステーションのオーディオデバイスの代わりに、ネットワークインターフェースに行うことにより、ISDN回線とコンピュータネットワークとのシームレスな音声通信環境を構築することが可能となる。

我々は実験としてLAN上のワークステーション（Sun SPARCstation10）とISDN回線上の電話機と通信を行った。図7に実験構成を示す。

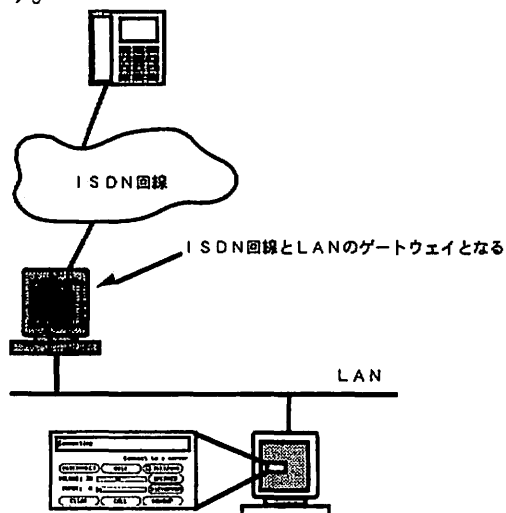


図7 実験構成

ISDN回線とLANの間にはゲートウェイとなるワークステーションを設置し、ゲートウェイプログラムをデーモンとして起動し、これを介して通信を行うようにした。ゲートウェイ／ワークステーション間の接続はTCPにて行い、クライアント／サーバー構成とした。

ワークステーションでのデコードはISDN回線での仕様に合わせゲートウェイではデータの転送のみとし、処理の簡素化を図った。

LANはFDDIを使用した。LAN上のトラフィックは常に15%程度かかっているが、音声伝送によるトラフィック64 kbpsにとっては影響もなく、また影響を受けることもなく、LANネックとはならない条件であった。

・品質

音声データはISDN回線上の電話からクライアントのワークステーションまでデジタルで伝送されるため、音自体の劣化はない。やはり、4章で示したように、音の連続性と遅延になる。連続性はプログラム構造を適性することにより対処できるので、遅延が問題となる。

ネットワークを介するため図5の処理でwriteでオーディオデバイスへの書き込みがソケットへの書き込みとなり、それ以降からクライアントでのソケットからの読み込みまでの処理が追加となる(図8)。

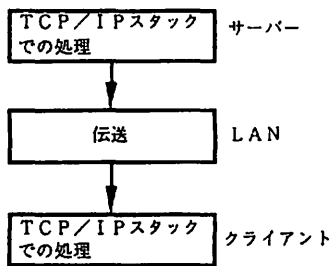


図8 図5に対する追加処理

つまり、ネットワークを介する分の遅延が生じる。

6. 応用例

3章で述べたようにAPIは電話に必要な基本機能単位にC言語関数としたため、様々なアプリケーションに対し自由に動作シーケンスを組み立てることが可能となった。さらに、今回のISDN回線インターフェースではDTMFデコーダーを装備した。DTMF信号はAPIから取得できるため、従来の電話のように電話のみのコントロールだけでなく、コンピュータ自体のコントロールも可能となる。これにより、対向の電話機からコンピュータネットワークに対するさまざまなコントロールが可能となった。

以下応用例について述べる。

(1) ISDN回線とLANとのゲートウェイ

5章での実験構成を拡張したものである。

LANに接続したワークステーション間で通信を行い、また、それぞれのワークステーションからゲートウェイマシンを介してISDN回線上の電話機との通信を可能とする(図9)。

機能的には現在市販されているPBXの機能を包含することができる。そのため、オフィス等ではLANと電話のケーブル配線をそれぞれ別に行っているものを(図10)LANケーブルの敷設のみで同等の実現できる。

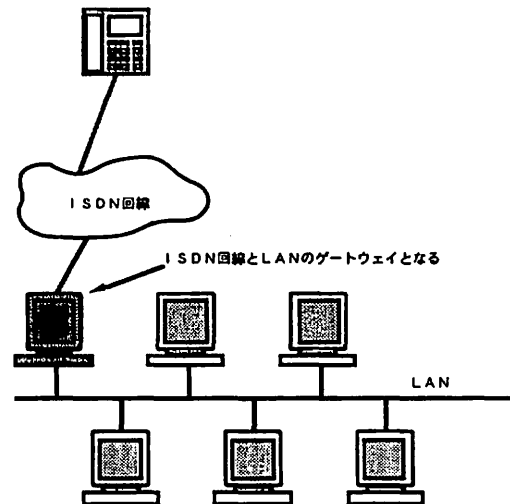


図9 ISDN回線とLANのゲートウェイの拡張

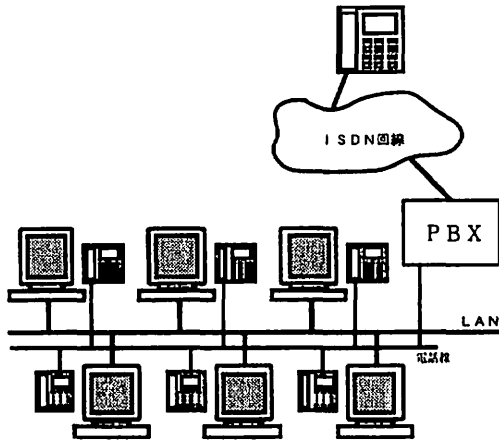


図10 オフィス内の配線例

- (2) 留守番電話
- (3) 伝言電話

いずれも音声蓄積を行う応用例である。

留守番電話、伝言電話のいずれについても通信対向からの音声を蓄積し、電話上でDTMFコントロールを使用して再生する。留守番電話ではワークステーション上のオーディオツールで再生することも可能である。

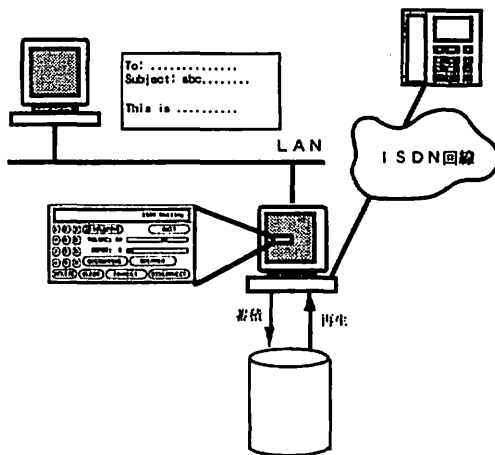


図11 音声蓄積アプリケーションの例

(4) MIMEメールゲートウェイ

前述と同様に音声蓄積タイプの応用例であるが、電子メールと電話の組合せである。非リアルタイム通信である電子メールとリアルタイム通信である電話を仲介するため、時間的な緩衝を行うことが可能となる。

動作としては、MIMEメールで送られてくる音声ファイルをゲートウェイとなるマシンに蓄積し、電話上で再生を行う。また、電話からの音声をファイルとして蓄積しMIMEメールとして送信する。電話からのコントロールはDTMFを使用して行う。

(5) ネットワーク管理

ネットワークトラブル発生時にNOCからページャーを鳴らす、さらにはメッセージを送信するという方法[8]がある。ここではページャーに代えて携帯電話を利用する方法である。

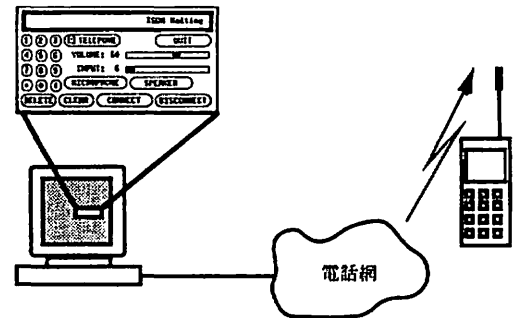


図12 NOCと携帯電話

音声メッセージは、あらかじめファイルとして蓄積しておき、トラブル発生時には携帯電話への自動発信を行う。携帯電話からはDTMFを使用してコントロールする。

7. おわりに

本報告では、コンピュータネットワークと電話のシームレス環境を構築すべく I S D N 回線インターフェースとその実験報告、応用例について述べた。

インターフェースの実現はデータのシリアル／パラレル変換のみという簡素な方法で行った。また、A P I は電話動作に必要な機能を基本機能単位で実現したためアプリケーションの可能性を広げることができ、さらに L A N と接続することにより、従来の電話が L A N 上のアプリケーションと融合することができ、目標とするシームレス環境実現の見通しを得た。

従来からコンピュータネットワークと L A N は分離された環境であり、本報告での方法により音声通信が可能となった。このことだけでなく、他の観点から見ると次のような意味もあることがわかる。

C S C W の観点から見ると電話という同期通信の世界と電子メールという非同期通信の世界をつなぐことができる。

マルチメディアの観点から見ると電話による音声通信とコンピュータによるデータ通信を融合させることができる。

音声品質はエンコーダー／デコーダーの性能によるところが大きいが、システムとしての品質は遅延である。遅延については今後、測定を行い遅延要素を解決する予定である。

また、6章で示したようなアプリケーションへの適応性も検証していく予定である。

【参考文献】

- [1] Windows(TM) Telephony API Preliminary, Intel/Microsoft, May 3, 1993.
- [2] Windows(TM) Telephony SPI Preliminary, Intel/Microsoft, May 3, 1993.
- [3] Stephen A. Uhler, "PhoneStation, Moving the Telephone onto the Virtual Desktop", USENIX Winter conference proceedings, January, 1993, pp131-140..
- [4] 財団法人日本 I T U 協会 "CCITT BLUE BOOK,."
- [5] Van Jacobson, Steven McCanne, "vat," Lawrence Berkeley Laboratory, University of California, Berkeley,, February 1992.
- [6] Henning Schulzrinne, "Voice Communication Across the Internet: A Network Voice Terminal," University of Massachusetts, July 1992.
- [7] 篠田、石井、桑名, "クロスプラットフォームコンピュータネットワーク上での音声通信の実現", マルチメディア通信と分散処理研究会, 1993年7月.
- [8] WIDEプロジェクト, "WIDE/PhoneShell", WIDE プロジェクト 1992年度研究報告書, 第16部, 1993年7月.

謝辞

本検討にあたり、NITTソフトウェア研究所 中村主幹研究員および関係諸氏から貴重なコメント及びご指導を頂いた。記して感謝する。