

ISDNを用いたネットワークアーキテクチャに関する実験

稗田薫	加藤朗	村井純
上智大学	慶応大学	慶応大学
電気電子工学科	環境情報学部	環境情報学部

技術の進歩により広域エリアネットワークで利用可能な通信媒体は高速で信頼性のあるものとなってきた。しかし、我々が利用しているIPネットワークでは、交換網の利用経験が少ないために交換網の特徴を生かして利用するには至っていない。本研究では、ISDNの回線交換網を有効に利用できるIPネットワークを構築するために必要な技術の開発を行なった。

An experimental network architecture with ISDN

Kaoru Hieda	Akira Kato	Jun Murai
Sophia Univ., Faculty of Science and Technology	Keio Univ., Faculty of Environmental Information	Keio Univ., Faculty of Environmental Information
7-1 Kioicho, Chiyodaku, Tokyo 102, Japan 5-322 Endo, Fujisawa, Kanagawa 252, Japan		

By the development of communication medium, especially switching circuits, data communication in a wide area network became more reliable and fast. Though we don't have enough experience and technic to support switching circuits such an ISDN yet. Therefore we tried to study the technic to use ISDN over IP network effectively.

1 序論

技術の発展により広域ネットワークでも信頼性があり高速なデータ通信が可能となった。また、色々な交換網が利用できるようになり、必要な時に必要な相手と通信することで効率の良いネットワークを構築するための基盤が整ってきた。しかし、現状では、通信媒体が開発されたのみで、ネットワークの利用が改善されたわけではなく、解決しなければならない問題が多く存在する。オペレーティングシステムでの交換網の扱い方、ネットワークでの交換網の扱い方などを検討し必要な技術を確認して、はじめて、これらの網を有効的に利用することができる。

本研究では、ISDN の回線交換網を利用できるようにオペレーティングシステム (BSD UNIX) に変更を加え、回線交換網を用いたネットワークをモデルに分類し、各々のモデルを実現するための問題点や解決方法について述べる。

2 目的

ISDN に代表されるデジタル公衆網の発達により、従来のアナログ公衆網に比べて高速な通信が可能となった。現在我々が使用している専用線などと同じ帯域幅を持つ回線を必要の際に必要な相手と接続することが出来るために、広域ネットワークでの通信を経済性やスループットの面でより効率的に行なえる基盤が整った。しかし、そのためには実際に公衆網を利用したネットワークを構築するための技術を確認しなければならない。

従来、公衆回線で利用していたアナログ回線はデータ通信を行なうためには帯域幅が狭く、信頼性が低いためデータ通信媒体としては非常に限られた範囲でのみしか利用できなかった。公衆網がアナログからデジタルに変わることにより、広範囲での利用が可能になってきている。しかし、交換網を効果的に利用するためには、Ethernet や専用線と異なる特質を持つ交換網をオペレーティングシステムで取り扱うための技術を確認する必要がある。また、既存のネットワークアーキテクチャの中で、公衆網が持っている特徴を生かした利用形態を考える必要がある。

本研究では、公衆網の特徴を生かした広域ネットワークを構築するために、実現すべきネットワークの形態および、その過程で必要となる技術について述べる。

3 モデル

目的とするネットワークの形態が異なる場合、ネットワークを構築する際に必要となる技術は本来異なり、それぞれの技術を開発する必要がある。しかし、ここでは、交換網を用いたネットワークの形態を分類しモデル化するこ

とで、実際に全ての形態で技術を開発することなく、交換網を利用するための技術の開発が行なう。

ネットワークを構築する際には色々な要求が発生し、これらの要求を何らかの制限により優先度を付け、優先度の高い要求を満たすようにネットワークを構築する。従って、実際に構築されたネットワークの形態が異なり、そのネットワークで必要とされる技術も異なってくる。モデルを交換網を用いてネットワークを構築する目的、つまり、ネットワークを構築する際の要求と制限によって分類して考えることで、異なる技術を必要とする交換網を用いたネットワークの分類が行なえる。

モデルの分類を交換網を利用する目的で分けるためには、どのような要求や制限があるのかを考える必要がある。要求は出来るだけ満たされる方が望ましいが、制限は必要な時のみ満たせばよい。交換網を利用したネットワークを構築する際の制限には、経済的問題、セキュリティ、ネットワークの管理、現在のネットワーク技術などがある。ここでネットワークの管理、つまり、ネットワーク全体を維持していく上での制限は、自主的に守らなければならない。よって、ネットワーク全体への影響を考えて分類する。まず、ネットワークに対して全く影響を与えない利用方法を行なわなければならないのは、個人的に交換網を利用する時である。個別のサービスによりネットワーク全体に影響を与える、つまり設定の変更等を行なう必要があってはならない。逆に、システムレベルで提供するサービスに交換網を利用する時はネットワークに対して影響を与える必要がある。

そこで、交換網を利用するネットワークを

- 個別サービスのために交換網を利用するネットワーク
- システムが提供するサービスのために交換網を利用するネットワーク

の二つに分類し、各々のネットワークを構築する際に必要となる技術について次に述べる。

3.1 個別サービスのための交換網の利用

個人的に交換網を利用する場合には、ネットワーク全体に与える影響が出来るだけ少ない形態を実現する必要がある。個人的に交換網を用いることによってネットワークに与える影響とは、例えば、回線の接続や切断によってネットワークの系に変化が生じ、その変化を伝達することである。具体的には、経路情報の伝達がこれに当たる。個人が交換網を利用するために生じる接続や切断の度にネットワークでの経路が変わり、経路情報がネットワーク全体

に伝搬した場合、ネットワーク運営上問題となる。従ってこのような利用の場合には、ネットワークに影響を与えない方法で実現する必要がある。

3.2 システムによるサービス提供のための交換網の利用

システムレベルでサービスを提供する際でも、出来るだけネットワークへ与える影響を減らすことが望ましい。しかし、ネットワークに影響を与えることによってサービスを提供する場合もある。例えば、既に接続されている専用線などに障害が生じ交換網を代用として利用する場合や既に接続されている専用線などと併用して負荷分散を行なう場合などがこれにあたる。すなわちネットワークの系の変化を必ずネットワーク全体に伝達しなければならない時は、既に何らかの接続が存在した場合である。逆に回線交換網のみを利用する場合を考えると、前述のような個人的な利用方法を除いた場合は、純粋に交換網の特徴である手軽に不特定多数と通信が出来る特徴を生かした利用をする際である。実際にシステムレベルでサービスを提供する際には、システム間で効率良く情報を交換するために利用する。この様な場合経路を静的に設定しておくことで、ネットワークへの影響を減らすことができるが、経路を静的に設定することで不特定と通信することはなく限られた通信相手のみと通信を行なうことになる。

従って、ここでは、

- 交換網のみを利用する場合
- 専用線などと併用して利用する場合

に細分化して考える。

4 実験

4.1 ドライバの作成

この節では、実際に ISDN の回線交換網を BSD UNIX [5,6,7,8] を用いて Internet で利用するために必要なドライバの構築について述べる。

4.1.1 ハードウェア環境

使用する回線は、NTT が提供している 64Kbps の INS64 を用いた。また、X.21 プロトコルによって回線制御が可能なターミナルアダプタを網とホスト間で用いた。ホストには、Sun OS を用いた Sun ワークステーションと UniB OS を用いた Luna ワークステーションを使用した。(Table 1)。

4.1.2 階層化インターフェイス

最近のインターフェイス、特に公衆網用のインターフェイスは、OSI の参照モデルに基づき階層的なプロトコル構造を用いている。また、各層で利用するプロトコルは ISO や CCITT

などで規定されたプロトコルを用いることが多い。プロトコル毎にモジュール化して作成することにより、作成したモジュールを他の網でも利用できる。従って、役割毎にモジュールに分割して作成し、階層的に重ね合わせて利用することで、効率的にネットワークインターフェイスを扱うことが出来る。

3.2節で述べたようにシステムレベルで交換網を利用するためには、交換網をシステムの一つの資源として扱うことが出来なければならない。ネットワークインターフェイスをシステムの資源として扱うためには、システムが制御できる必要がある。従って、交換網の接続、切断等を行なうモジュールを上位層に位置させ、回線制御をシステムが行なえるようにする必要があるのである。

BSD UNIX では複数のプロトコル体系を提供しているが、X.21 や SDLC は上位層を多重化する機能を持たないために、受信したデータがどのプロトコル体系のデータであるか判断できない。また、回線の速度などホスト間でのネゴシエーションを行なったりするものがあるため、point-to-point プロトコルが必要となる。

X.21 プロトコルで使用するデータリンクプロトコルは Bi-sync であるが、Bi-sync プロトコルはキャラクタ単位でインタラプトを生じるためにホストにかける負荷が大きい。負荷がかかることによりデータを紛失し、最転送を行ない負荷を増大させ悪循環を引き起こす可能性がある。X.21 プロトコルは回線制御手順であり、データ転送の際の規定ではない。つまり、データ転送にはパケット単位でインタラプトを発生させる負荷の軽いプロトコルを用いる必要がある。また、X.21 プロトコルを解釈するモジュールは接続状態か非接続の状態かで下位層を入れ換える機能を持つ必要がある。

このような要求を満たすために今回の実験では、INS64 の回線交換網を利用するために Table2 のようなモジュールを階層的に重ね合わせて利用した。

最下層の if_zss,if_zsy はハードウェア依存性が高いため今回は SUN の Sunlink/INR のパッケージのものを利用した。また、MCP ボードを使用した場合は、Sunlink/MCP パッケージのものを利用した。その他のモジュール if_duc,if_xsp,if_xcs を作成した。

4.1.3 自動回線制御

我々が実験に使用した Internet 環境では、ネットワークプロトコルに IP [2]、トランスポートプロトコルに UDP [3] や TCP [4] を用いている。ネットワークプロトコルである IP はデータグラムによる転送を行っており、トランスポート層でも UDP はデータグラム転送を行なう。アプリケーションで UDP をトランス

計算機	Sun Sparc Station I	Sun3/60	Sun3/160	Luna SX9100DT
拡張ボード	-	-	Sun MCP	GPX
変換器	RS232C-RS499	RS232C-RS499	-	RS232C-RS499
ターミナルアダプタ	NH-5101-1	Aterm 112	ISPTC	Aterm 112
回線	INS 64	INS 64	INS 64	INS 64
チャンネル	1 B channel	1 B channel	2 B channels	1 B channel
速度	64K bps	64K bps	64K bps	64K bps

MCP: Multiprotocol Communications Processor 変換器: Black Box 社製 NH-5101-1: 日立製
Aterm 112: NEC 社製 ISPTC: 富士通製

Table 1: 利用したハードウェア環境

アプリケーション	
socket	
TCP	UDP
IP	
if_duc : ダイヤルアップインターフェイスの操作	
if_xsp : Xerox Synchronous Point To Point プロトコルの解釈	
if_xcs : X.21 プロトコルの解釈	
if_zss,if_zsy : 同期モードでの通信を行なうドライバ	

Table 2: INS64 の回線交換網を利用するためのモジュール階層

ポートプロトコルとして使用した場合や交換網を経由する場合など、全てコネクションレスである。しかし、回線交換網などは、回線の接続や切断を行なうコネクション型データ転送を仮定している。従って、コネクション型のデータ転送を仮定するネットワークインターフェイスでコネクションレスのデータ転送を行なうための技術を確認しなければならぬ。

ここで必要となる技術は自動回線制御である。発信着信や切断などを自動的に行なうことで、上位層に対して従来の専用線等と同じように見せかけることで、コネクション型インターフェイスであることを意識せずに利用することが出来る。しかし、複雑な自動接続や切断の機構を用いることで、モデルへの適応力を失う可能性がある。従って、最も簡単な自動発信着信や自動切断の機構を組み込む必要がある。

切断接続を自動的に行なう際に問題となる点が回線の接続切断のタイミングと接続相手の認識である。最も簡単な方法は、最初にネットワークインターフェイスに出力されたパケットを解析し、パケットの宛先に自動発信を行なう。この方法を用いてシステムが接続相手やタイミングを制御するためには、経路情報を制御し、パケットの出力インターフェイスを選択させることで実現できる。従って、ネットワークインターフェイスが最初のパケットで自動的

に接続する設計になっていても、必要に応じて接続相手やタイミングを操作することが可能である。

自動切断に関しては、回線の利用が終了した時点で切断することが最も望ましい。しかし、実際に回線の利用が終了したことを知る方法はないため、利用されていないと予想し切断する必要がある。最も簡単な予想方法は、一定間隔でパケットの入出力を数え、増加がなければ利用されていないと判断して切断する方法である。この方法を用いる際に問題となるのが、上位プロトコルには一定時間アイドルな状態が続くとパケットを送信し、その応答を受けることで相手の状態を確かめるものがあることである。この様なプロトコルを利用すると回線が自動的に切断されない。今回の実験で用いるプロトコルで、この様なパケットを送出するプロトコルは、Xerox synchronous プロトコルと TCP のオプション機能の keep alive を設定したアプリケーションと経路制御情報を転送する routed がある。まず、Xerox synchronous プロトコルは、回線を制御するモジュールの下層に位置するため問題ない。次に経路情報を転送する routed に関しては、回線交換網上の経路に関しては静的に設定し、その他の経路情報は動的に設定することが出来る gated を用いてこの問題を解決した。最後に、Keep alive を用い

たアプリケーションに関しては対処方法がないため、パケットの送出間隔より長く設定する必要がある。

今回の実験では、最初にネットワークインターフェイスに出力されたパケットの宛先に対して自動的に発信するようなドライバを作成した。また、切断に関しては、ある一定時間間隔でパケットの入出力の総和を数えゼロになった時点で切断を実行するようにした。

自動回線制御を行なう際のもう一つの問題がアドレス変換である。上位プロトコルから利用している回線が交換網であることを意識させないためには、上位のアドレスから網でのアドレスに自動的に変換する必要がある。

アドレスの変換方法には、テーブルを用いて静的にアドレスを変換を行なう方法がある。テーブルによる変換方法は高速に変換が行なえるために、接続作業も高速に行なえる反面、事前に設定してある相手としか回線の設定が出来ない。又、動的に問い合わせを行ないアドレスを変換する方法を用いると、ネットワーク内に登録されている全ての計算機と接続が出来る反面、問い合わせに時間がかかり接続作業がスムーズに行なわれない。そこで、今回の実験では頻繁に使用するデータのみを事前にオペレーティングシステム内のテーブルに登録し、もし、テーブルを用いて変換できない場合には、問い合わせを行なう方法を用いて実現した。

4.2 個別サービスのために交換網を利用したネットワークの構築

個人的に回線を利用する場合に重要なことは、3.1節で述べたように回線の接続切断の際に出来るだけ他に影響を与えないことである。また、利用者の立場から考えると経済性やスループットなどの効率に比べ、回線を利用できる状態の方が重要である。

ネットワークが動的に接続切断された場合、接続や切断の度に経路情報が伝送されない限り接続先の計算機以外への通信は不可能である。このような個人的な利用により接続の度に経路情報が伝搬することはネットワーク管理上望ましくない。また、静的に経路情報を設定することも不可能である。なぜなら、経路上の計算機が必ずしも利用者の要求を受け入れる補償がないからである。個人的な利用目的のために経路情報を動的・静的に設定することは事実上不可能であれば、個人的な利用目的では利用者の権限がおよぶ範囲内でしか利用できないことになる。

経路情報の伝搬の問題が起こる理由は、経路情報を新たに加えることによるものである。新たに経路情報を加えず、既に伝搬している(設定されている)経路情報を利用する方法を考え

てみる。例えば、接続先のサブネットの1つ取得し、そのサブネット上のアドレスを自システムに付ける。次にサブネットに関する経路情報を接続先で設定する。この方法を用いると接続先の組織内のみ経路情報を伝搬させることで利用が可能となる。しかし、個人のために割り振ることの出来るサブネットの数には限界があるため、接続相手毎に動的に割り振る必要がある。この際には経路情報や接続相手とアドレスの動的マッピングなど動的変更による問題が生じる。

二つ目の方法として、接続先の計算機が存在するネットワーク上に位置するように見せかける方法がある。この方法の実現は、Ethernetのようにハードウェアアドレスと上位プロトコルアドレスとの変換を行なっているネットワークインターフェイスでのみ可能である。接続相手のネットワークアドレスと同じネットワークアドレスを持つアドレスを事前に取得しておく。Ethernetの場合、ハードウェアアドレスと上位プロトコルアドレスの変換をARP[1]で行なっているため、ARPを用いて接続相手の上位アドレスと接続先のハードウェアアドレスをネットワークに送信する。接続先へのパケットは全て接続相手に送られ、接続相手ではそのパケットの宛先をみて回線交換網のネットワークインターフェイスに出力することで通信が可能となる。同時に接続が可能な接続相手の数は、ネットワーク上で利用されていないアドレスの数となる。また、経路情報は一切変更する必要がない。この方法を用いることでネットワーク上の計算機の設定を変更することなく、個人的な利用が可能となる。

今回は事前にアドレスを割り当て、proxyarpを用いて接続先のネットワーク上にあるように見せかけて、自宅などの接続を行ない実験した。

4.3 回線交換網のみを利用したネットワークの構築

システムレベルでサービスを提供するために、交換回線網のみを利用する場合は、システム間での情報交換があげられる。実際には、ニュースシステムの記事や電子メールを組織に配送する場合などが考えられる。配送する情報が生じた際に回線を接続して配送すると、経済的にもスループット的にも効率が非常に悪い。ある程度配送する情報を蓄積し、まとめて配送することにより効率を良くすることができる。しかし、必ずある程度蓄積するまで配送しなければ情報の伝達遅延が大きくなる可能性がある。伝達遅延をある程度に抑えて、効率良く情報を伝達させるためには、情報を蓄積し定期的に配送する機構が必要となる。これは従来から行なっていたUUCPプロトコルなどを用いて実現できる。

このようなシステム間の情報転送に加えて、個

人的な利用も考えておく必要がある。個人的利用を考慮すると UUCP プロトコルの様な単純なプロトコルではなく、多重化やエラーチェックなどを行なうプロトコルの上の豊富なアプリケーションを利用できる方が望ましい。従って、TCP/IP プロトコルを用い、情報伝達には UUCP-T プロトコルを用いる方法が有効である。

今回は、UUCP-T プロトコルを用いて情報の転送を行ない、組織間や自宅などと接続して実験を行なった。また、回線の利用の制限に関してはパケットの転送機能を停止させることで、経由する計算機を利用できるもののみに限った。

4.4 既存の回線と併用して交換網を利用したネットワークの構築

既存の回線の補助として利用する方法は、既存の回線が何らかの理由により使用不能になった場合に回線交換網を利用する方法と、トラフィックが増えた際に既存の回線と回線交換網の両方を同時に利用し負荷分散を計る方法がある。既存の回線と交換網を用いて負荷を分散させる場合、既存の回線の状態から利用率を推測して負荷を分散させる必要がある。負荷分散を実現するための技術を用いて、既存の回線のバックアップとしての利用は実現できるが、ここでは、両者を個別に検討する。

既存の回線のバックアップとして利用する場合

既存の回線のバックアップとしての利用を考えると、経済性やスループットの効率より、データ転送が可能であることが重要となる。しかし、転送されたデータを一度紛失した場合、紛失したデータをネットワーク内で作り直すことは不可能である。従って、上位のトランスポート層でデータの補償を行なっていると仮定し、データが紛失した場合再転送することを前提にする。回線の補償を行なうためには、まず既存の回線状態を常時把握しなければならない。回線状態を把握するために最も簡単な方法は両端の計算機で回線状態をモニタする方法である。しかし、両端の計算機で異常が起きた際には、回線状態を把握することが不可能となる。よって両端より外側の計算機で回線状態をモニタする必要がある。応答のあるパケットを回線や両端の計算機を通る経路で送信し、応答を受信することで回線や両端の計算機の状態を把握する方法が最も一般的である。この方法を用いた場合に問題になることが、回線や両端の計算機が本当に異常なのか、トラフィックが多いためにデータの紛失、遅延が起きているのかの判断である。外部から回線の状態を適格に判断をすることは非常に難しい。また、前述のように上位プロトコルでも同様の方法を用いて回線の状態をモニタしている。従って、上位プロ

トコルが回線異常と判断する間隔より短い間隔で回線の状態をモニタし、回線異常と判断し回線を接続し交換網を利用するようにする設定しなければならない。加えて、この間隔を決定する際には、経路情報が伝達するのに必要な時間を含めて計算しなければならない。

バックアップ回線が利用可能となった後に、既存の回線の状態を調べる時は、IP プロトコルのオプション機能であるソースルーティングを用いて確認のパケットを送信する必要がある。なぜなら、パケットの経路を指定しなければ、新しく設定された経路情報に基づいてバックアップ回線を通り相手側に到達し応答が返ってくる。よって経路を指定してパケットを送信する必要がある。

今回の実験では、Figure1の構成で、専用線の状態を検知する役目を回線交換網を持つ計算機上で行なった。

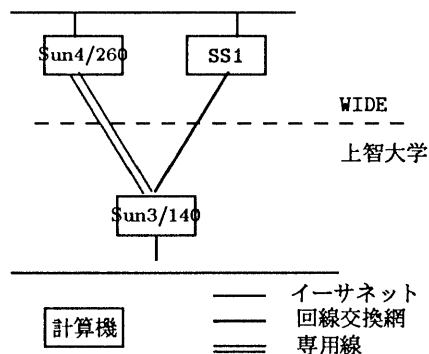


Figure 1: 上智大学と WIDE との接続図

既存の回線と併用し負荷分散を計る場合

回線交換網を用いて動的に負荷の分散を計る場合、ネットワークのトポロジーやある瞬間のネットワークの利用状況を瞬間に把握する必要がある。

ここで問題となるのが TCP プロトコルが用いているコンジェッションコントロールである。コンジェッションコントロールとは、経由する回線を占有することなくデータ転送を行なうために、応答時間からデータ転送の間隔を制御する機構である。特に、経由する回線の中に帯域幅の狭い回線がある場合は非常に有効な機能である。しかし、コンジェッションコントロールが行なわれていないと、実際に必要とされている帯域幅の推定が非常に難しくなる。また、動的に回線を増やし帯域幅を広げても、コンジェッションコントロールが帯域幅の変化を検知し、推定し直し、回線を有効に利用でき

るまでには時間がかかる。加えて、応答による回線状況の判断は非常に難しいために、TCPのコンジェッションコントロールは、応答から徐々に転送の間隔を変える方法を用いているために帯域幅の変化を有効に利用されるまでに時間がかかる。TCPのコンジェッションコントロールを利用しない場合を考えると、経由する回線に帯域幅の狭い回線があり、回線交換網などを利用した負荷分散が可能でない場合、データの転送により回線が飽和状態に達する。加えて、送信したデータが紛失したものと考え、最転送を行なった場合などは最悪の状態になる。この問題を解決するためには、ネットワークの構成を把握し、両端でコンジェッションコントロールを行なうかどうかを決定し転送する必要がある。現在の技術でネットワークのトポロジーを自動的に調べることは非常に難しい[9]。また、バランスの良い負荷分散を行なうためには、現在の回線の状態を動的に把握する必要がある。例えば、RTT(Round Trip Time)などを利用して回線の利用状況を判断する方法がある。しかし回線の利用状況とRTTの関係を明確でないために必ずしも現在の利用状況を把握できるとは限らない。このようなネットワークの状態を把握する研究は現在いろいろ行なわれている[10]。従って、今回の実験では、この様なネットワークのトポロジーや利用状況を事前に把握した上で、負荷分散を行なう実験ネットワークを構築して、その有効性についてのみ評価を行なう。

今回の実験では上智大学とWIDEを接続するネットワーク上で、ニュースの記事を転送する際のみ回線交換網を利用し、その他のデータは既存の専用線を利用するように負荷を分散させて実験を行なった。

5 実験結果と評価

作成したドライバを用いてINS64を1チャンネル使用した場合で、回線接続にかかる時間が800msから4秒位で、RTTが約30ms、ftpによるファイル転送が約6.7KB/sである。64Kbpsの専用回線を利用する場合と比較すると最初に接続にかかる時間以外は全く同じ結果が得られた。回線接続にかかる時間は、実際に電話回線が接続されるのに必要な時間が3秒以内で、相手の上位アドレスから網でのアドレスに変化し、接続後に相手とのネゴシエーションを行なうのに必要な時間が1秒以内である。

個人的な利用目的のためにproxyarpを用いた実験ネットワークでは、接続後必要な作業が全くないために、上記のドライバで得られた結果がそのまま得られる。接続相手の網内での位置により、接続が完了するまでの時間が異なる。実験の結果では4秒以内に接続する。最初のパケットのみ影響を受けるが接続後は通常の

専用線と全く同じように扱える。

交換網のみを利用して構築したネットワークの実験では、転送すべき情報を蓄積し定期的に接続することで経済効率をあげることが出来た。また、接続回数を増やすことで情報の伝達遅延を減少させることが出来た。しかし、接続回数を増やすことにより不必要な接続が増え効率が悪くなる。従って、転送する情報の量や発生する時間などを考慮して接続時間を設定する必要があった。

既存の回線のバックアップとして回線交換網を利用した場合は、既存の回線が切断された時は自動的にバックアップを行なった。実験で行なったように既存の回線とバックアップとして利用する回線を持つ計算機が隣接している場合では、経路情報を伝達する遅延が少ないために回線のバックアップが行なえた。しかし、既存の回線とバックアップ回線を持つ計算機がネットワークを介して接続されている場合などでは、経路情報が必要な計算機に伝わるまでは、送信したデータが計算機間でパケットがループしたり、上位プロトコルが一定時間内に応答がないために回線の異常と判断し、強制的にアプリケーションを終了させたりすることが起こる。加えて、バックアップ回線を持つ計算機と既存の回線を持つ計算機とを接続しているネットワークに障害が起きた場合、バックアップ回線を接続して、バックアップ回線を持つ計算機側に位置する計算機はバックアップを利用してデータを送信するが、逆側からのデータは全て既存の回線を持つ計算機側に送られるために一方向のみの通信になる。従って、バックアップをより有効的に利用する場合には、現在のネットワークの状況を把握し、経路の変更をすみやかに伝達する必要がある。

既存の回線と回線交換網を併用して負荷分散を計る場合について考えてみる。実験を行なった上智大学とWIDEインターネットバックボーンとの接続では継続的に大容量のデータ転送を行なうアプリケーションであるニュースの記事転送のみを回線交換網でおこなうことで、既存の専用回線の帯域幅を減少させることが出来た。上智大学では、瞬間的に生じる大容量のデータ転送と帯域幅を必要としないが常時接続されていることを重視しているような専用線の利用形態を取っている。大容量のデータ転送を回線交換で行なうことで常時接続に利用する専用線の帯域幅を大幅に減少させることが可能となり経済的な効率をあげることができる。ネットワークの利用状況を把握することが可能な場合では、簡単な負荷分散の方法で効果をあげることが出来る。しかし、実際に動的に負荷分散を行なう際には、ネットワークの状態を瞬間で把握し、経路を制御する必要がある。

今回の実験では、三種類のモデルに分割して交換網の利用を考えた。このモデルによって実際に現在我々が交換網を利用しているネットワークは、全て表すことが出来た。しかし、現在の技術がおよばないために、専用線などと交換網を併用する場合はより細分化する必要がある。モデルの正当性を示すためにも、ネットワークの状態の把握、経路情報の高速伝搬などの技術を確立する必要がある。

6 結論

ネットワークインターフェイスをプロトコルや役目毎にモジュール化し階層化することで、新しいネットワークインターフェイスを簡単に利用できる。また、ネットワークでの回線交換の利用目的が異なるために、回線の制御方法を変更したり、新たなプロトコルを追加も簡単に実現できる。自動接続や切断を行なうことで、コネクション型の通信を仮定している回線交換網などを用いてデータグラム通信が可能となり、Internet 上で利用していた既存の豊富なアプリケーションに一切変更を加えることなく利用できた。今後、パケット交換網などにもこの考え方を導入して行く必要がある。

ネットワークに影響を与えず利用する方法では、現在の技術を用いて交換網を有効に利用できた。しかし、逆に交換網の接続によりネットワークの状態に変更を与える目的で交換網を利用する際には、経路情報の伝達遅延や整合性などが問題となり、限られた範囲でしか交換網を有効に利用することができない。このような問題を解決するためには、分散処理環境下で経路情報などの情報を確実にかつ高速に伝搬させる方法が必要となる。また、ネットワークでの負荷分散を自動的に行なうためには、動的にネットワークの状態を把握する必要がある。このような問題はネットワークの動的な変化により生じるものである。従って、今後、交換網をより有効に利用するためにも、ネットワークの状態が動的に変化することによって生じる問題を研究する必要がある。

References

- [1] David C. Plumer, "An Ethernet Address Resolution Protocol," *DoD Protocol Handbook*, PFC 826, Nov. 1982.
- [2] Jonathan B. Postel, "Internet Protocol," *DoD Protocol Handbook*, PFC 791, Sep. 1981.
- [3] Jonathan B. Postel, "User Datagram Protocol," *DoD Protocol Handbook*, PFC 768, Aug. 1980.

- [4] Jonathan B. Postel, "Transmission Control Protocol," *DoD Protocol Handbook*, PFC 793, Sep. 1981.
- [5] Samuel J. Leffler, Marshall Kirk McKusick and Michael J. Karels, *The Design and Implementation of the 4.3BSD UNIX Operating System*, Addison-Wesley Publishing Company, Inc. 1989.
- [6] Samuel J. Leffer, William N. Joy, and Robert S. Fabry, "4.2 BSD Networking Implementation Notes," *Unix Programmer's Manual*, CSRG University of California at Berkely, Jul. 1983.
- [7] Samuel J. Leffer, William N. Joy, and Robert S. Fabry, "4.2 BSD Interprocess Communication Primer," *Unix Programmer's Manual*, CSRG University of California at Berkely, Jul. 1983.
- [8] "Interprocess Communication Primer," Sun Microsystems, Inc., 1986.
- [9] "大規模広域分散環境におけるネットワーク管理モデル", 上水流由香, 佐藤智満, 中村修, 砂原秀樹, マルチメディア通信と分散処理研究会, 1991.
- [10] "大規模広域ネットワークにおけるRTTに関する考察", 中村修, 北島剛, 村井純, マルチメディア通信と分散処理研究会, 1991.

謝辞 議論をして頂いた WIDE プロジェクトメンバに感謝します。