第 4 部

ISDN

第 1 章

はじめに

1.1 ネットワークの広域化と安定性

ローカルエリアネットワーク (LAN) の技術的発展と普及に伴い、計算機ネットワークは、LAN 間接続により急速に広域化が進んできた。その結果、地理的に分散した計算機やファイルなどの資源を手近な資源として扱えるようになり、ネットワークを利用するユーザにとって、資源が分散しているという意識は次第に薄れてきている。つまり、いつでも、どこでも、使用したい資源を利用できる環境になってきているということである。

しかし、それらの資源を結びつけているネットワークに突然障害が起これば、“資源が分散していた” という事実を思い知らされるとともに、ネットワーク障害によって受ける影響の大きさに驚くことになる。ネットワーク障害が頻繁に起こり、その度に資源の利用を中断されるのであれば、ネットワークを利用して、資源を扱うことに対する信頼性は失われ、ネットワークで接続されている意味はなくなってしまう。ネットワークで接続する以上はユーザに見放されない程度の信頼性を提供する必要がある。そのためには、ネットワークの障害に対処できる安定したネットワークを構築しなければならない。

ネットワークの障害といっても、LAN 間を接続するのに使われる計算機（ゲートウェイ）の障害や、ネットワークを提供する物理接続の障害などさまざまな要因がありこれらを防ぐことには物理的な限界がある。また、ネットワークに障害が発生していない場合でも、ネットワークが混雑し通信の遅延が生じることがある。ネットワークは、それを通じて流れるデータの量（トラフィック）により通信速度や安定性に対する影響を受けやすいという性質がある。しかしながら、ネットワークの混雑によって、通常の通信状態と比較して品質（通信速度、安定性等）が著しく低下する場合には、ネットワークの信頼性を失墜させる要因となり得る。従って、ネットワークの混雑についても対応していかなければならない。

同一 LAN 間を接続するには、まず経路を一つ確保することが必要であ

るが、それだけではその経路に障害が起きた時にはその LAN 間は接続不可能となってしまうので安定したネットワークを提供しているとはいい難い。したがって、安定したネットワークを構築するためには、同一 LAN 間を接続する経路を複数にすることが考えられる。

ところで、ネットワークを運用するのに費用がかかるのは事実であるので、安定したネットワークを構築するために同一 LAN 間を接続する経路を常時複数、つまり冗長構成とすることは経済面を考慮すると実現が困難である。そこで、ネットワーク障害が発生した場合にのみそれを補完するような代替経路を提供することが考えられている。この方法をとればネットワークの障害にも対応でき、通信コストを押えることも可能である。

このような、障害が発生して必要が生じた時のみ提供される代替経路は間欠リンクと呼ばれる。また、ネットワークに混雑が生じた時も、混雑の緩和のために一時的に間欠リンクを利用する方法も考えられる。さらに、発展的応用としてネットワークを利用したいという要求が生じた時にのみ間欠リンクを用いて接続する方法も必要になってくる。

間欠リンクの実現のためには、接続したいと思った時にすぐ計算機ネットワークとして利用でき、また、上で述べた経済面を考慮すると、利用しない時には課金が安価で済むような物理媒体を選択しなければならない。

計算機ネットワークが発展するにつれ、ネットワークを提供する物理媒体も多種多様になってきているが、このうちの一つに日本では 1988 年 4 月からサービスの開始された ISDN(Integrated Services Digital Network) がある。サービスが開始されて間もないということもあり、ISDN をどのように計算機ネットワークの中に取り込んでいくかについての議論が盛んである。ISDN は従来の電話網に匹敵する大規模な網を提供し、その網に加入したもの同士で、通信したい時に容易に通信することが可能であるのでこれは間欠リンクに適していると思われる。従って、本論文では間欠リンクを提供する物理媒体として ISDN に焦点を当てて考えていく。

1.2 本論文の目的

ISDN を特に広域な計算機ネットワークに取り入れるためには、他の物理媒体によるネットワークとの整合をとることが必要である。多くの異種の物理ネットワークを相互接続する技術は internet と呼ばれ、TCP/IP[1],XNS[2](XeroxNetwork System) 他多数のプロトコルが存在し、その中でも TCP/IP は比較的広く利用されているので本論文では TCP/IP に絞って考えてみる方針である。

[1] 大学、政府機関、軍関係の繋がった DARPA インターネットで用いられているプロトコル群。以後、インターネットは DARPA インターネットと略す。

[2] Xerox Corporation によって開発されたネットワーク間プロトコル群

具体的には、インターネットにおける広域分散環境構築プロジェクトの一つである **WIDE** (Widely Integrated Distributed Environment) において、ISDN を用いた IP 接続の方法を考えることにより、ISDN による間欠リンクの実現を試みる。本論文では間欠リンクの実現に当たって、従来のインターネットの機構を混乱させないということを基本方針とする。また、実現の初期の段階では、接続するということに重点をおく。その後、間欠リンクにとって重要な接続・切断のタイミングについて実験を重ねて効率のよい通信を提供できるような方法を練っていく所存である。さらに、間欠リンクそのものの発展的利用法についても考えていく。

1.3　本論文の構成

本論文では、まず第 2 章で ISDN という物理媒体の特徴について述べ、第 3 章で間欠リンクの必要性・ISDN を間欠リンクとして利用することの意義について考える。そして、第 4 章では、ISDN をインターネットにおける間欠リンクとして利用する上で必要な IP 接続の具体的設計として、IP アドレスの割当てや接続・切断のタイミング・ソフトウェア構成などについて議論する。第 5 章では ISDN を用いた IP 接続の部分的実装として手動の接続・切断の制御について述べる。第 6 章では ISDN を用いた場合の転送速度等について評価・考察を行ない、改良すべき点などについて考える。最後に、第 7 章で今後の課題として ISDN を用いた IP 接続の間欠リンクとしての応用的側面について論じる。

第 2 章

ISDN(Integrated Services Digital Network) とは

計算機ネットワークを提供する多種多様の物理媒体の中で、ISDN とはどのような特徴を持った物理媒体なのかについて述べる。[95]

2.1 ISDN 出現の背景

昨近の通信技術の発展には目覚しいものがあり、それに伴って社会における通信のあり方や利用目的も多様化しつつある。このような多様な情報通信の要求に対応可能な世界共通の技術標準を作ろうと、国際的な共同研究や技術開発が進められた。

ISDN は、国連の専門機関である ITU（国連電気通信連合）の中の CCITT（国際電信電話諮問委員会）を中心に進められてきた国際標準規格に基づいて構築されるネットワークである。ISDN はその名が示す通り、ディジタル網であり、電話サービス、データサービスを始めとする多種多様の通信サービスを単一の網で総合的に提供することが可能である。ISDN は日本では 1988 年 4 月からサービスが開始され、現在まだ ISDN 利用者が少ないことから従来の電話網に付加する形となっており、長い年月をかけて将来は従来の電話網がすべて ISDN に移行するとされている。

2.2 ISDN の提供する通信サービスの特徴

2.2.1 通信能力

ISDN のディジタルソケットを通じて提供される通信能力は基本的には $2B + D$ となっている。つまり、64kbps の通信用チャネル（B チャネル）2 本と 16kbps の制御用チャネル（D チャネル）から成っている。物理的な線は 2 対で時分割多重化することにより 3 本のチャネル $2B + D$ を実現

している。通信を確立するために D チャネルを利用し、確立した後は B チャネル 1 個を用いて通信を行うというチャネルの明確な使い分けが特徴である。

従来の電話網の通信能力は 9.6kbps 程度以下であることと比較してみると ISDN の 1 チャネル当たりの通信能力 64kbps というのは非常に優れていると言える。また、従来の電話網はアナログ処理であるので電話サービスのみの単一サービスに限定されていたが、ISDN はディジタル処理であるので、電話サービスだけでなく、ディジタル処理の通信機器（ファクシミリ、計算機）を用いた通信サービスも同時に提供できるようになった。$2B + D$ というチャネル構造のため、ISDN 用の加入者線を 1 本引くことにより基本的には通信機器を同時に 2 個接続して利用することが可能である。従って、1 個は電話に、もう 1 個は計算機に同時に接続することや、2 個とも計算機に接続することなど、利用者が選択することができるようになっている。

2.2.2　情報転送モードと課金

ISDN の提供する情報転送モードには次のようなものがある。

- 専用線モード
- 回線交換モード
- パケット交換モード

専用線モードは回線交換において、固定した相手を指定するように設定するので、回線交換と同様である。

回線交換モードは、一度「呼」が確立すると切断するまで継続的に通信チャネルが確保されるもので、B チャネルを使用する。課金は通信時間と通信距離によって決められる。これは、従来の電話網とほぼ同じといえる。

パケット交換モードは、情報を適当な大きさに区切りひとまとめにしてヘッダを付加し、散発的に送受信するもので、D チャネルを使用する。課金はパケット数によって決まる。しかし、パケット交換モードは現在ではまだ提供されていない。

従って、本論文では ISDN の回線交換モードを利用することを中心に議論を進めていく。

2.2.3　アドレス

ISDN を用いて通信を行う際のアドレスには ISDN アドレスを使う。ISDN アドレスは ISDN 番号と ISDN サブアドレスからなる。ISDN 番号

は、従来の電話番号の延長であるが、最大で 15 桁までとなっている（従来の電話番号は最大 12 桁。）ISDN の場合、チャネルの特徴により、1 本の加入者線に対して複数の通信機器（最大 8 個の端末）を接続できるので個々の機器を指定することができるようにサブアドレスがつけられ、最大 40 桁までの ISDN サブアドレスとなっている。
具体例を Figure2.1 に示す。

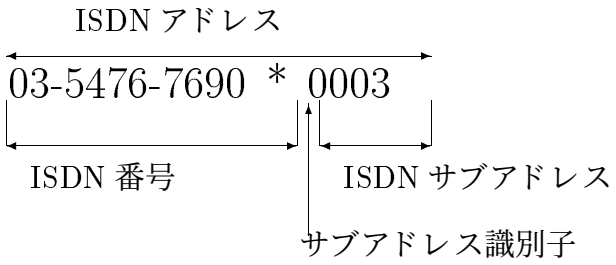


図 2.1: ISDN アドレスの例

2.3 ISDN の将来的展望

ISDN の基本は $2B+D$ であるが、$23B+D$、$30B+D$ も提供され始めており、将来的にはさらに広帯域な通信（384kbps の H_0 など）が提供される見通しである。ISDN による高速な通信が提供されると、これまでの通信では困難であった画像転送も容易になる。ISDN は通信技術の進歩をもたらすものとして期待されている。計算機ネットワークに ISDN を組み込むことは、通信技術の進歩による恩恵を被ることにより、計算機ネットワークの新しい側面をも切り出すことになる。そのためには、まず、計算機ネットワークに ISDN を組み込むことから始める必要がある。具体的には、ISDN をインターネットに接続する方法を考え、ISDN を利用した間欠リンクの実現を目指す。

第 3 章

間欠リンクと ISDN

3.1 WIDE の現状と間欠リンクの必要性

日本のインターネット構築プロジェクトである **WIDE** ではここ数年で急速にネットワークが広域化し大規模となってきた。現在、Figure3.1 に示されるような広域分散環境を実現している。

WIDE への参加組織は年々増加し、海外へもネットワークが広がり始めている。ローカルネットワーク（例えばキャンパス）間の接続に用いられている物理媒体としては 64kbps のディジタル専用線、また、X.25[1]などがあり、ローカルネットワーク内（例えばキャンパス内）は、主にイーサネット[2]で接続されている。

東大–東工大間は、専用線と X.25 を共に接続するという冗長構成になっており 、専用線を用いた通信に障害が起きた時には X.25 を用いることにより、ネットワークの分断を防ぐことが可能である。また、東大、慶応大、ハワイ大では輪状にネットワークが構成されているので、2 大学を直接接続している専用線が利用できない時には迂回経路をとることが可能である。しかし、その他ではローカルネットワーク間には 1 つの経路が存在するだけなので、ゲートウェイの故障や、その経路の物理接続の障害により、ネットワークが分断される結果となる。

広域ネットワークに限ったことではないが、目的のホストに到達するためには、それまでに通過するゲートウェイとネットワークが全て正常に機能していなければならない。広域ネットワークになると、目的のホストに到達するまでに通過するゲートウェイ数は多数となるので、ゲートウェイの故障やネットワークの障害により通信が不能となる可能性が高くなる。また、広域ネットワークでは、ゲートウェイやネットワークのハードウェア・ソフトウェアの障害による影響が広範囲に渡るので、広域ネットワークとしての機能を完全に克服するまでに時間がかかりやすいことが挙げら

[1] トランスポートレベルのネットワークサービスの CCITT の標準プロトコル。
[2] LAN の技術の一つ。イーサネット (Ethernet) は Xerox 社の登録商標です。

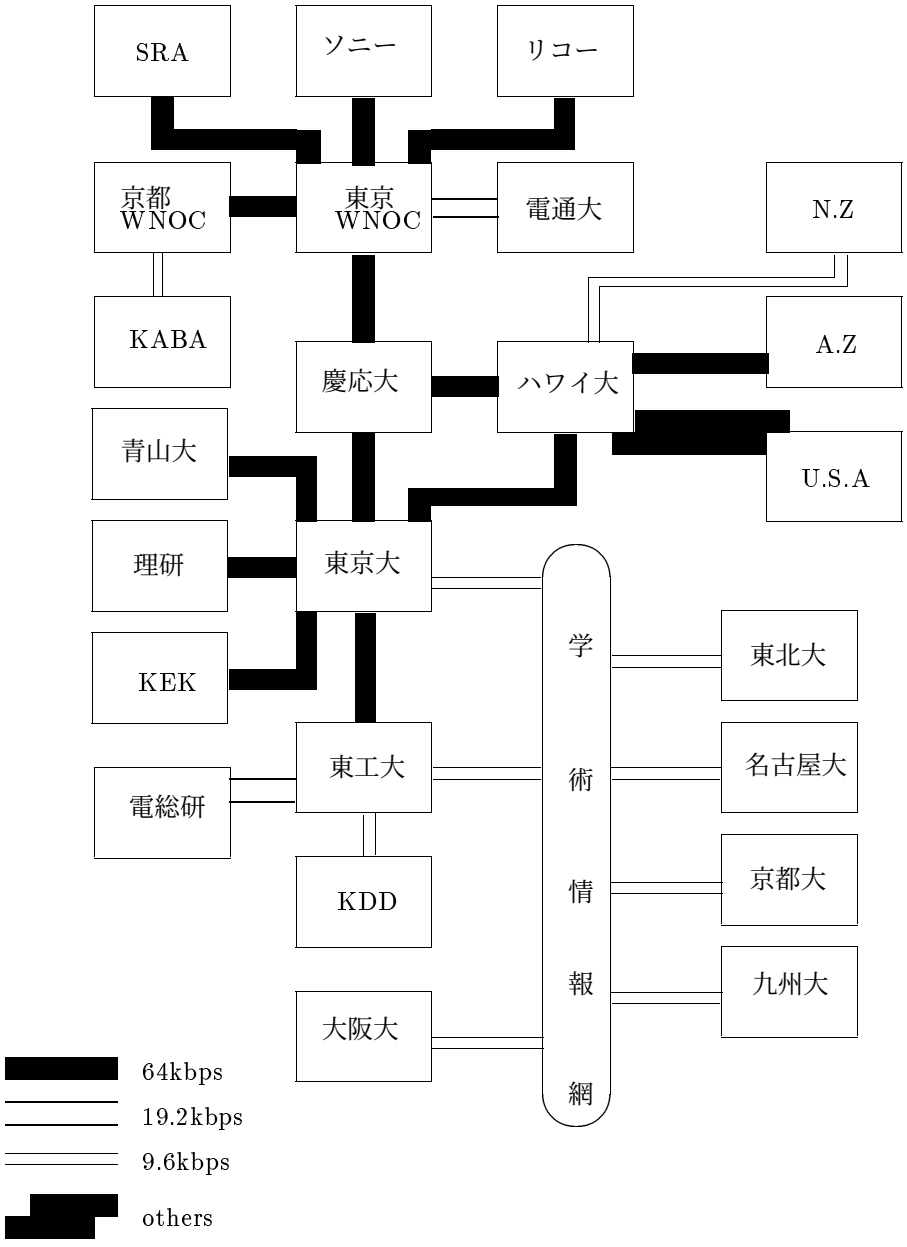


図 3.1: **WIDE** における広域分散環境の現状

れる。従って、広域ネットワークでは、各種の通信の障害に対する迅速な対処が望まれる。つまり、障害が起こっても広域ネットワークとして機能することが可能となるような安定性が必要である。 ネットワークの安定の条件をまとめると、次のような点が挙げられる。

1. 同一ローカルネットワーク間を接続するリンクが常に 1 つは確保できること。

2. ハードウェア・ソフトウェアの各種の障害から迅速に回復できること。

WIDE では今のところ、ネットワークの広域化が急速に進んでいるが、リンクを複数確保することには手が回らない状態である。このままの状態ではネットワークが安定であるとは言い難い。少なくとも、ネットワーク障害時には代替リンクを確保することが必要となる。また、ネットワークのトラフィックはユーザがネットワークをどのように利用するかに依存し定常的ではない。従って、通常 9.6kbps で滞りなく通信が済むようなリンクであっても、ネットワークが混雑し、著しい通信の遅延が生じた場合には、一時的にリンクを補充することも必要である。しかし、代替リンクとしてのリンクであっても、補充のためのリンクであっても、一時的な使用のために常に専用線等の常設リンクを確保しておくのはそのリンクが使用していない間も課金されるのでネットワークの運用費が非常に高価となる。

そこで、必要な時のみリンクを起動することのできる間欠リンクというものが重要になってくる。間欠リンクは使用しない時には課金されないか、非常に安価であることが常設リンクと明確に異なるべき点である。従って、間欠リンクを効率的に使うことによって専用線などの常設リンクよりも安価な通信が提供可能であるといえる。

間欠リンクが発展すれば、専用線を張るほど頻繁に通信が行なわれないような組織間の通信に間欠リンクを利用することも可能となる。この場合では、専用線等の常設リンクと間欠リンクのどちらを選択するかは、リンクの能力に差がなければ、どちらがより安価であるかによると思われる。間欠リンクにより効率的で安価な通信を提供することが可能となれば、**WIDE** の広域化に伴う安定性の追求と発展に貢献するはずである。

ところで、間欠リンクは必要な時のみ起動されるものだが “必要な時 ” をどのように判断するのが適当かを明確にしなければならない。つまり、通常のリンクに異常が発生したことや補充が必要なことを誰が判断するのかということである。

まず考えられるのが『ネットワークを利用しているユーザ』である。リンクの異常発生により最終的に影響を受けるのはユーザだからである。しかし、全てのユーザがネットワークについて診断する能力を持っていると

は限らない。従って、ユーザが影響を受けてから間欠リンクを起動させるのは確実な方法とは言い難い。

従って、ネットワーク機能の内部で通常リンクの異常を検知し間欠リンクを起動させる方が、ユーザはネットワークについて深い知識を持たなくても、通常通り通信を行うことができるので、間欠リンクとしての威力を十分に発揮できる。また、間欠リンクが必要でなくったことの判断も同様に、ネットワーク機能の内部で判断をするのが妥当である。つまり、ネットワークの制御機構の内部で、間欠リンクの起動・切断の機能を組み込むことが必要になる。

3.2 間欠リンクの現状と問題点

間欠リンクは、必要な時にいつでもリンクを起動することができなければならず、しかも使用しない時には通信費用がかからないあるいは十分に安価であることが条件である。すると、間欠リンクとして適した物理媒体は限定される。

現在間欠リンクとして用いられているものには電話網を IP 接続として利用する dialup IP[35][?] がある。しかし、dialup IP は LAN 間を接続するというよりは自宅の計算機と自分の所属する組織間を接続するために用いられることが多い。従来の電話網を利用するため、呼を確立するまでの時間が長く、TCP のタイムアウト[3]に間に合わないことがある。それから、網の能力の限度から、高速なモデムを用いても通信速度は 9.6 kbps 程度以下となっている。この通信速度では、通信量の多いリンクの代替リンクとして十分でない場合が多い。

広域ネットワークでは一般に、代替リンクが必要なリンクというのはそれが利用不可能となった時の影響が大きい。そのような通常リンクを代替する一時的なリンクも、通常リンクと同等の通信能力を提供できることは望ましいことである。専用線が 64kbps であることを考慮すると、専用線と同等の通信能力を提供する間欠リンクは従来の dialup IP では実現できない。
そこで、従来の電話網を拡張したものといえる ISDN を用いて間欠リンクを実現する方法を考えてみる。

[3] Transmission Control Protocol. トランスポート層で信頼性のあるデータ配送を行なう。その際タイムアウトまでに配送ができない時は再送を行なう。

3.3　ISDN を用いた間欠リンクの利点・特徴

それでは、ISDN を間欠リンクとして用いた場合どのような利点があるかについて述べる。
まず、ISDN の B チャネルが提供する 64kbps という通信速度は、現在 LAN 間の接続によく用いられている専用線が提供する通信速度と同じであるので、通信量の多いリンクの代替リンクとしての能力は従来の dialup IP と比較して 6~7 倍になる。

これまでは一時的であれば質が良いとは言えないリンクであっても我慢してきた感があるが、ISDN を用いれば、利用者にとって代替リンクということが意識されずに通常と同じように通信を行なうことが可能となる。それから、ISDN の回線交換を利用する際は、実際に接続しなければ課金されないので、間欠リンクに求められる条件を満たしているといえる。

但し、ISDN を用いた間欠リンクを代替リンクとして使うだけでなく、LAN 間を常設リンクを用いずに間欠的に接続するために利用することになると、経路制御の方法が問題となってくる。従来の電話回線を用いた dialup IP では、マシンの起動時に間欠リンクを接続して利用する（つまり接続してから使う）という方法が一般的であったが、これからは、規模の大きい LAN 間の接続に用いられるので、マシンの起動と間欠リンクの接続は非同時に行なわれる（つまり、ゲートウェイは既に存在しているがリンクが存在していないことがある）ことになる。これまでの経路制御では既に存在するリンクに対して経路情報から最適経路を選択して行なってきたが、間欠リンクは通常は接続せずに必要な時だけ実際に接続するので、“接続する可能性のあるリンク ”を経路制御でどのように扱うかを工夫する必要がある。

将来的に ISDN の交換サービスはパケット交換と回線交換を共に提供するので、これらを使い分けることが可能である。具体的には、リンク自身の容量を動的に調整することに応用することなどが挙げられる。パケット交換と回線交換の使い分けは ISDN は間欠リンクの発展に大いに貢献することが期待される。

WIDE ではインターネットの標準プロトコルである TCP/IP プロトコルを扱っている。TCP/IP プロトコルは、ネットワーク機能の中枢であるデータの配送の方法や経路の制御方法に関わるプロトコルである。インターネットへ ISDN を組み込むために、ISDN を経路制御に関わる IP に接続することを考える。

第 4 章

ISDN を用いた IP 接続の設計

計算機ネットワークを構成するソフトウェアは階層構造になっているが、その中で TCP/IP プロトコル群における IP モジュールはどのような役割を果たしているかを述べる。

4.1 インターネットにおける TCP/IP プロトコル群の階層構造

インターネットにおける概念的レイヤリングは、アプリケーション層・トランスポート層・インターネット層・ネットワークインターフェース層の 4 つからなり、その下にハードウェアがある。そして、レイヤ間でオブジェクトを交換し、データを配送する。それぞれのレイヤの役割について簡単に説明する。

アプリケーション層
: ユーザはインターネットにアクセスするためにこの層のプログラムを起動する。プログラムで扱うデータは、転送のためにストリームとして 1 つ下のトランスポート層へ渡す。逆に、他のホストから転送されたストリームがトランスポート層から渡される。

トランスポート層
: この層ではあるアプリケーションプログラムと他のプログラムとの通信を提供する。つまり、あるホストのあるポートと他のホストのあるポート間の通信を提供する。アプリケーション層から来たデータのストリームをパケットに分割し、パケットの流れを調整して信頼性のあるトランスポート (TCP を用いる) を提供することができる。信頼性のあるトランスポートとは通信するデータが正しく配送されたかどうかを確認するもので、一定時間経過しても配送の確認が得られなければデータを再送する。また、信頼性のないトランスポート (UDP—User Datagram Protocol—を用いる) も提供できる。信頼性のないという

ことは一度データを転送した後は再送しないということである。
複数のアプリケーションプログラムからのデータを受け入れるため、どのアプリケーションプログラムかを判断することができる。下のインターネット層から来たパケットを指定されたプロトコル (TCP,UDP) で処理し、アプリケーション層へ渡す。

インターネット層

この層では、ホストとホストの通信を取り扱う。目的のホストに到達するための経路制御を行なう層である。トランスポート層から来たパケットも、ネットワークインターフェース層から来た IP データグラムも、次の経路を決定してから次の層へ渡される。経路制御の際に判明する転送のエラーに関する ICMP(Internet Control Message Protocol) メッセージも必要に応じて取り扱う。ホストが所有する IP アドレスは、そのホストが所有するネットワークインターフェースに対してつけられている。従って、複数のネットワークインターフェースを持っているホストには複数の IP アドレスがつけられる。IP アドレスを見ればホストや、ホストが属している LAN のネットワークを識別することができる。また、IP データグラムの IP ヘッダから、デスティネーションホストでトランスポート層のどのモジュールへ渡すかがわかる。

ネットワークインターフェース層

物理層に依存する層で、インターネット層から来た IP データグラムは各物理媒体で定められた物理フレームに入れて次に渡すべき同一物理ネットワーク上のホストへ送られる。次に渡すべきホスト（ゲートウェイ）はインターネット層で決められている。データリンク層とも呼ばれる。

4.2　UNIX における IP モジュールの機能

WIDE では UNIX[1] システム上でインターネットを構築しているので、UNIX システム上の IP モジュールについて説明する。

4.2.1　IP input routine

IP input routine は、ネットワークインターフェース層から渡された IP データグラムの処理を行うルーチンである。ネットワークインタフェース層から渡された各 IP データグラムに対して、まず、自分のホストへの

[1] UNIX は AT&T の登録商標。

データグラムかどうかを、IP ヘッダのデスティネーションアドレスと、自分のホストの所有するインターフェースのアドレスを比較して調べる。自分宛ならば、着信データグラムとして上位モジュール（トランスポート層の TCP,UDP 等）に渡す。

自分宛でなければ、データグラムをデスティネーションホストに到達させるために必要な経路を定める。経路決定のためには、デスティネーションホストやネットワークに基づいた経路情報が記されたルーティングテーブルを参照し、デスティネーションホストへ到達するために次にどのゲートウェイにデータグラムを渡せばよいか、さらにどのインターフェースへデータグラムを渡すのかという情報を得る。その情報とともに、IP output routine へ IP データグラムを渡す。

4.2.2 IP output routine

IP output routine は、上のモジュールである TCP や UDP から来たパケットを処理したり、IP input routine から来たパケットを処理するルーチンである。パケットに IP ヘッダを付加し、経路を決める必要がある時はルーティングテーブルを参照して次の経路を決定してから、インターフェースに IP データグラムを渡す。IP input routine から渡されたパケットで、既に次の経路が決まっている場合は、経路が使用可能か、インターフェースが起動状態かを確認してインターフェースに IP データグラムを渡す。

4.2.3 ルーティングテーブルの参照

ルーティングテーブルは経路の決定のために参照される。ルーティング情報は相手ホストに到達するために何個のゲートウェイを経由するかというホップ数等を、RIP[2]等のプロトコルに従って定期的に交換されている。これらの情報から様々なデスティネーションホスト、あるいはデスティネーションネットワークに到達するための最短経路が得られる。そして、この最短経路を利用するために必要な情報からなるものがルーティングテーブルである。ルーティングテーブルのエントリ数が多数の場合、ルーティングテーブルのエントリを上から順に参照するのは効率が悪いので、実際にはハッシュ関数を用いて効率化を図っている。ハッシュ関数から得られるエントリは 2 つある。1 つはデスティネーションのホストアドレスに基づいたエントリで、もう 1 つはデスティネーションのネットワークアドレスに基づいたエントリである。2 つのエントリにつきホストアドレスに基づいて得たエントリを優先して参照する。具体的な参照方法は次の通りである。

[2]Routing Information Protocol.Berkley 4.3 BSD UNIX システムでルーティング情報を交換するためのプロトコル。

(1) デスティネーションについての情報を見て、ハッシュ関数により、ルーティングテーブルのどこを参照するかの参照番号を求める。このとき、デスティネーションアドレスに基づいた参照番号、デスティネーションのネットワークアドレスに基づいた参照番号を求めておく。

(2) まず、デスティネーションアドレスに基づいた方について、ルーティングテーブルを見て次に投げるインタフェースが起動状態かどうか調べる。ルーティングテーブルにあるデスティネーションと、目指すデスティネーションが同じかどうか、調べる。

(3) (2) で見つからなければ、今度は、デスティネーションのネットワークアドレスに基づいた方で、インタフェースが起動状態か、ルーティングテーブルのデスティネーションと目指すデスティネーションの、所属する domain や network が一致するかどうかを調べる。

(4) (3) でも見つからなければ、wildcard gateway を探す。(これは、デフォルトルート[3])

4.2.4　パケットのフラグメント化とリアセンブル

インターフェースには、一度に送出可能なデータサイズの最大長 (MTU—Maximum Transfer Unit—) が定められており、IP からインターフェースに送出するデータグラムの長さがこれを越える時には MTU に合わせてパケットを分割しなければならない。これにより分割されたパケットはフラグメントと呼ばれる。複数のゲートウェイを通過して配送されるパケットは、
(現在のパケット長) > (次に渡されるインターフェースの MTU)
が成立する時のみフラグメント化を行なう。

デスティネーションホストに到達したパケットは、フラグメントを集め 1 つのパケットに再構成される。フラグメントは、決められた時間内にすべて集められなければ捨てられる。このように、IP モジュールの機能は IP アドレスを基盤として成り立っているということがわかる。すると、ISDN に接続されたホストには、どのように IP アドレスを割り当てればよいかということが問題になってくる。

[3] ルーティングテーブルから経路が得られない時に用いる省略時の経路。ルーティングテーブルを小さくするために、デスティネーションアドレスを 0.0.0.0 としている。

4.3 ISDN 網の捉え方

IP アドレスはネットワーク id(identifier) と、ホスト id からなっている。一般に IP アドレスのネットワーク id は、LAN あるいは LAN の集合に対し、その規模に応じて 1 つ割り当てられている。そして、その LAN あるいは LAN の集合に属する各ホストを識別するためにホストアドレスが割り当てられる。このとき、 同一ネットワークアドレスを持つホスト同士は同一ネットワーク内に属すると考えられ、そのネットワーク内のホストに到達するための入口がインターフェースであり、インターフェースに対して物理的な通信ポートが 1 つ対応している。

ISDN 網は非常に大規模な網であるが、ISDN 網をどのように捉えるかは IP アドレスのネットワークアドレスをどのように割り当てるかということに帰着し、経路制御に重大な影響を与える。ISDN 網に対しては次のような捉え方が考えられる。それぞれについて特徴・問題点を述べる。

1. Point-to-Point ネットワークとして捉える。

2. マルチアクセスネットワークとして捉える。

4.3.1 Point-to-Point ネットワークとして捉えた場合

Point-to-Point とは、2 つの物理的ポート（以下、ポートと略す）間の通信に着目したもので、1 つのポートと通信する可能性のあるポートとの間に 1 つずつネットワークアドレスが割り当てられる。計算機に接続される ISDN 網のポート数を N とし静的にアドレスを割り当てると、$N(N-1)$ 個のネットワークアドレスが必要になる。**WIDE** において ISDN 網を用いて計算機の接続を行うゲートウェイは将来多数になると考えられる。$O(N^2)$ でネットワークアドレスを割り当てていくと N が 100 程度でもネットワークアドレスが足りなくなることが予想される。ネットワークアドレスを節約するために動的にネットワークアドレスを割り当てる 方法もあるが割り当て方が複雑になる。

4.3.2 マルチアクセスネットワークとして捉えた場合

マルチアクセスとは、同時に多数のノードと通信することができるということである。非常に多数のノードと通信できる例としてはイーサネットがある。また、X.25 は物理的なポートは 1 つであるが論理的チャネルは例えば 128 というように多数の設定が可能であるので、マルチアクセスネットワークとして扱われることが多い [94][111]。

ISDN は同一の加入者線（つまり同一の ISDN 番号）に基本的にチャネル数（つまり物理的ポート）が 2 であり、同一ホストに複数のポートが接続される可能性がある。同一ホストに複数のポートが接続される場合には、同時に、所有するポート数と同数のノードと通信を行なうことが できる。つまり、ISDN をマルチアクセスネットワークとして捉えることができる。X.25 と比較すると、加入者線 1 本に対するチャネル数は少ないので加入者線 1 本ではマルチアクセス性は弱いが、 同一ホストが加入者線を複数本接続する場合や $23B + D$ のチャネル構造の提供を考慮すると、将来的には ISDN のマルチアクセスとしての性質が強くなってくることが予想される。マルチアクセスネットワークでは、IP のネットワークアドレスはある広域ネットワークに属する ISDN ゲートウェイ全体で 1 つを割り当てればよい。ネットワークアドレスの数や、ISDN の将来の発展形態を考慮し、ISDN 網をマルチアクセスネットワークとして捉えることにする。

4.4　IP アドレス割当ての単位

ISDN 網には 1 つのネットワークアドレスを割り当てることにしたが、次に IP アドレスを具体的に何に対して割り当てるかが問題になる。先も述べたように通常は IP アドレスのホストアドレスは、ホストの所属するネットワークのインタフェースに対して割り当てられるが、この時あるネットワークに所属するホストについて、 （ホスト: 物理的ポート：IP アドレス）は、(1:1:1）が成り立っている。ISDN 網の場合は、同一ホスト（ゲートウェイ）に複数の物理的ポートが与えられる可能性があるので、IP アドレス割り当ての単位としていくつかが考えられる。

- ISDN アドレス（ポート）に対して割り当てる
- ISDN 番号に対して割り当てる
- ゲートウェイに対して割り当てる
- 等価なゲートウェイグループに対して割り当てる

ここではそれぞれについて特徴を考える。
ゲートウェイが所有するポートが n であるとして議論を進める。

4.4.1　ISDN アドレスに対して割り当てる場合

この場合、ISDN 網に関しては（ゲートウェイ:物理的ポート:IP アドレス）は （1:n:n） となる。

同一の相手ゲートウェイに対し、複数のポートを用いて複数のリンクを張った場合 (Figure4.1[4](b)) には、1 つ 1 つのパケットをそのうちのどのリンクを用いて流すかという負荷分散をしなければならないが、IP アドレスがポート毎に割り当てられているので、経路制御アルゴリズムの中に負荷分散の機能を含める必要がある。現在は最適経路を 1 つだけ採用しそれを用いてパケットを送るという経路制御アルゴリズムであるから、負荷分散の機能を含めると経路制御が複雑になり、混乱することが予想される。

4.4.2 ISDN 番号に対して割り当てる場合

（ゲートウェイ:IP アドレス:物理的ポート）は、(1:m:n$\cdots$m $\leq$ n）となる。ゲートウェイの所有する物理的ポートがどの ISDN 番号に属するものかによって IP アドレスの数は異なる。同一の ISDN 番号に属しているポートが異なったゲートウェイに所有されている場合 (Figure4.2 (b)) は、異なるゲートウェイが同一の IP アドレスを割り当てられることになるので、希望のゲートウェイに接続できない可能性がある。これは経路制御を混乱させる。従って、これは適当な方法ではない。

4.4.3 ゲートウェイに対して割り当てる場合

（ゲートウェイ:IP アドレス:物理的ポート）は、(1:1:n) となる。

つまり、単一ゲートウェイに収容されている ISDN のインターフェースの集合全体に 1 つの IP アドレスを割り当てることになる。

この場合は、IP や経路制御のレベルでポートの割当てをする必要がなく、ISDN へのインタフェースが選択されてからポートの割当てをすればよい。その際にはポートを割り当てるコントローラが必要である。

同一の相手ゲートウェイに複数のリンクを張った場合のリンク間の負荷分散については、ISDN ドライバによって実現可能である。ゲートウェイに対して IP アドレスを割り当てる最大の利点は経路制御アルゴリズムの変更が不要なことである。

4.4.4 等価なゲートウェイグループに対して割り当てる場合

目的ノードに到達するためにゲートウェイが複数存在し, どれに接続しても目的を達することができる場合、その複数のゲートウェイ (Figure4.4 の G_1、G_2、G_3) を等価なゲートウェイグループと呼ぶことにする。

[4]ISDN 網と計算機はハードウェアで直接接続するのではなく、Terminal Adaptor(T.A.) を介して接続される。

この場合、グループ内の 1 つのゲートウェイについて、(ゲートウェイ：IP アドレス:物理的ポート）は、(1:1:n）となる。 但し、この場合の IP アドレスは同一グループ内のゲートウェイが共有するものである点が、ゲートウェイに対して IP アドレスを割り当てる場合と異なる点である。
従って、経路制御では ISDN へのインタフェースを選択した後、グループ内のどのゲートウェイを用いるかを決定することになる。これは、どの物理的ポートを用いるかを決定すれば、そのポートを所有するゲートウェイも自動的に決定される。結局ポートを割り当てればよいので、ゲートウェイに対して IP アドレスを割り当てる場合と同様である。

グループに対して IP アドレスを割り当てる利点はグループ内のゲートウェイのダウンに対し、グループ内の他のゲートウェイがカバーすることが可能であるので、経路の安定性が高まるということである。

問題点としては、グループ内にネットワークの障害が発生した場合は、グループ内のゲートウェイが同一の IP アドレスを持つために、IP アドレスを用いたグループ内の通信が不能になることが挙げられる。
インターネットは、異なる種類の物理媒体を用いて接続がなされている場合でも、共通のプロトコルで通信が行なえることを目指すものであるから、その共通のプロトコルの変更は避けるべきである。つまり、ISDN を用いた通信に IP アドレスを割り当てるには、それにより、経路制御アルゴリズム全体に変更を加えずに済む方法をとるのが望ましい。

従って、上の 4 つの方法のうち、経路制御アルゴリズムに最も影響を与えずに済む方法として、ゲートウェイに対して IP アドレスを割り当てることにする。ISDN ゲートウェイへの IP アドレスの割り当てについて Figure4.5に示す。

4.5　呼の自動制御方法

第 3 章で述べたように、ISDN を用いた間欠リンクでは、発呼や呼の切断の必要性をネットワーク制御機構の内部で検知し、発呼・呼の切断を行なうことが効率的な運用に大きな影響を与える。そこで、具体的にいつ接続し、いつ切断すればよいのかを明確にすることから始める。4.1 で述べたように、インターネットにおける TCP/IP プロトコル群には概念的なレイヤリングがあり、各レイヤによって、パケットから得ることのできる情報には制限がある。各層でどの程度の情報を得られるかを述べ、発呼や切断のタイミングについて考える。

アプリケーション層

個々のデータやプログラムの内容を知ることができる。

トランスポート層
どのホストのどのポート間の通信かがわかると共に、ポート番号からどのアプリケーションのものかがわかる。

インターネット層
どのホスト間の通信かが IP アドレスによりわかる。また、上のトランスポート層でどのようなプロトコル（TCP、UDP 等）を用いているかがわかる。

ネットワークインターフェース層
自分と次にパケットを渡すゲートウェイの IP アドレスがわかるので、経路の一区間, つまり同一物理ネットワーク内の通信について知ることができる。

間欠リンクを主に主要経路をカバーするリンクとして用いる場合には主要経路の障害や回復をソフトウェアが検知して発呼・切断をする方法が一般的である。また、間欠リンクの応用的な利用のために、インターフェースにパケットが到着した時発呼し、ある程度パケットが到着しなければ切断するという方法が考えられる。この場合はパケットといってもどのようなパケットなのかを区別して判断する必要が生じるかも知れない。そのためには、インターフェースにどのようなパケットが流れてくるのかを調べることが必要である。

4.5.1 パケットのフローパターンについての推察

etherfind[5]の ようなパケットの観察機能を用いて、あるイーサネットのインターフェースに実際に入ってくる物理フレームのうち IP データグラムを TCP パケットと UDP パケットに分けて処理を行ったグラフを FigureA.1、FigureA.2に示す。グラフ上の線は 1 秒間に入ってきたデータグラムのバイト数の合計である。

総合的に見て断続的に流れてくるパケットと、連続的に流れてくるパケットの 2 種類があることがわかる。断続的に流れてくるパケットの代表例としては経路情報等のネットワークを管理するためのパケットがあるが、これは接続されていれば定期的に流れてくるものである。従って断続的なパケットは定期的であることも多い。

グラフに示されたパケットフローの特徴としては、次のような点が挙げられる。

[5] NIT(Network Interface Tap) というデバイスを用いて特にイーサネットのインターフェースに入ってくる物理フレームを監視するためのコマンド。

- 定期的に流れてくるパケットはデータ量が固定的であることが多い。
- トランスポート層で信頼性を保証しない UDP を用いたパケットでは定期的に流れるパケットが多く、データ量は、流れる時は 1 秒間に約 200 〜 2000byte 程度である。
- トランスポート層で信頼性を保証する TCP を用いたパケットでは、定期的に流れるパケットはデータ量が 1 秒間に約 100byte 程度であり、一時的に流れるパケットに比べて著しく少ない。アプリケーションプログラムを利用している間に流れるものと推察される。

連続的に流れるパケットではリモートホストの資源を利用したアプリケーション、つまり、ユーザが実行するアプリケーションとほぼ一致しているといえる。これは、インターフェースから出ていく物理フレームについても同様であると考えられる。

断続的かつ定期的に流れるパケットについては、例えば経路情報は 30 秒程度の短い間隔で流れてくるので、“切断するタイミングを判断する対象となるパケット ”として含めると、切断の必要性を検知することは実質的に不可能である。
一方、断続的に流れるパケットの間隔（例えば 30 秒）を基準としてその間隔内にパケットが到着しなければ呼を切断すると決めてしまうと頻繁に切断され過ぎて、アプリケーションプログラムの実行は不可能となる。

また、発呼に関して、パケットが来たら発呼するという方法をとる場合でも断続的に流れるパケットを考慮に入れると、切断しても間もなく接続されてしまうことになるので間欠リンクとしての意味をなさなくなってしまう。従って、パケットの出入りを監視して発呼や切断の判断をする場合には断続的に流れるパケットは無視した方がよさそうである。
これらを考慮すると、発呼や切断についてのタイミングの設計は次のようになる。

4.5.2　発呼

発呼のタイミングは ISDN を用いるリンクの他に通常別のリンクが存在する場合（連結な場合）と存在しない場合（非連結な場合）とで間欠リンクの使用目的が異なるので、それぞれの場合について考える。

連結な場合

ISDN を用いた間欠リンクを使用するのは、通常使用されているリンクに障害が起きたときや、混雑が生じたときである。

まず考えられる方法は経路情報に頼るやり方である。つまり、間欠リンクについては、通常使用されているリンクの経路情報よりもホップ数[6]の多い経路情報を流しておく。通常リンクに障害が起こり、相手ネットワークへ到達不可能になった時は、間欠リンクの経路情報が取り入れられる。間欠リンクのゲートウェイは通常リンクの障害を知らせる経路情報を取り込むと発呼する。

しかし、この方法の難点としては障害の起きたことを知らせる経路情報は伝搬が遅いので障害が発生してから間欠リンクが利用されるまでの遅延が大きくなることが挙げられる。従って遅延が小さくなる方法を検討する必要がある。

遅延を減らす方法として通常使用されているリンクの状態を監視し、障害や、混雑などの異常を検知して経路情報が広まるまでに発呼することが考えられる。ところで、リンクの状態の監視は次の 2 通りの方法が考えられる。

1. ISDN ゲートウェイが通常リンクを監視する
2. 通常リンクのゲートウェイ（これと ISDN ゲートウェイが一致する場合もあり得るが）がリンクの障害・混雑を検知して ISDN ゲートウェイに知らせることによって監視を行なう。

しかし、ゲートウェイのダウンによりネットワークが機能しないということはしばしば発生することである。2 の場合、ゲートウェイがダウンするときに、確実に ISDN ゲートウェイに通知することが可能であるとは限らないので、適した方法とは言えない。

従って、ISDN ゲートウェイが通常リンクを監視する 1 の方法を採用する。

広域的な LAN 間接続は、専用線などのように各 LAN のゲートウェイの間に物理的な接続をすることが一般的であるので、通常リンクの両端にゲートウェイが存在している。すると、具体的なリンクの監視方法としては、ISDN ゲートウェイが、通常リンクによって接続される相手のネットワーク側のゲートウェイ（つまり、他のネットワーク側のホスト）に通常リンクを通って到達できるかを定期的に調べるという方法が考えられる。そして、到達できない状態を、ネットワーク障害が発生したと解釈して発呼すればよい。(Figure4.6参照)

これには 1 回の調査にかかる時間の基準時間を設け、その時間内に応答が返ってこないならば到達できない状態と判断するのがよいと思われる。さらに、通常リンクの標準転送時間を設け、この時間内に応答が返ってこ

[6] 目的ホストに到達するまでに通過するゲートウェイ数

ないならば、リンクが混雑していると判断し補充リンクとして発呼をすることも考えられる。しかし、混雑に対する補充リンクとして用いるには、等価経路に対する負荷分散の機能が必要とされ、それに対する経路制御アルゴリズムの対応が求められるので、現在では実現が難しいと言える。

非連結な場合

間欠リンクを使用するのは、主にユーザが別のネットワークパーティション[7]上の資源を利用する必要があるときである。従って、ユーザが自分以外のネットワークの資源を利用するようなアプリケーションプログラムを実行するときに発呼すればよい。
具体的には、ユーザが流すデータのうち、デスティネーションホストが自分のネットワークパーティション以外のホストであるときに、発呼をするということである。自分のネットワークパーティションでないということを判断するには IP アドレスが必要となる。

さらに、4.5.1 で述べたように、ネットワークを利用する必要のある IP データグラムの中で、トランスポート層の UDP を用いるパケットについては断続的・定期的に流れる場合が多いので発呼のタイミングを測るパケットの中に含めないことにする。TCP を用いるパケットの中にも定期的に流れるパケットがあるが、これはアプリケーションプログラムを実行している時に流れるパケットであるので、アプリケーションプログラムを実行していない時は流れないと予想される。従ってアプリケーションを始めようとする最初の TCP のパケットを検知すればよい。

このように UDP と TCP のパケットを区別するには、IP の上のモジュールのプロトコルタイプを見て判断することが考えられる。パケットに対しこれらの情報を得ることができるレイヤは 4.5 の最初で述べたようにインターネット層である。

そこでまず、インターネット層から発呼することが考えられる。この場合ユーザが流すデータがインターネット層へ来たときに、IP アドレスと、IP の上のモジュールのプロトコルタイプを検査する機能が必要となる。つまり、IP アドレスから間欠リンクが必要なことを検知し、しかも IP の上位モジュールとして TCP を使用するような IP データグラムであるならば発呼をするという方法である。

しかし、これは、経路制御部分で、TCP を用いているかどうかを検査する新しい機能を付加することになる。これは、ISDN ゲートウェイにしか必要がない機能であるので、間欠リンクを利用しないパケットの経路制

[7] 複数の LAN、あるいはローカルネットワークの集合。論理的なネットワークとして捉える場合に使われる。例えば 1 つのキャンパスネットワークなど。

御には影響を与えないようにすることが必要である。

ところで、間欠リンクを利用するパケットであっても、間欠リンクが接続されていれば他のパケットと同様の経路制御で扱うことができるので問題は発呼するまでである。従って、まだ接続されていない間欠リンクについては、接続される可能性のあることを情報としてとっておく必要がある。つまり、経路制御において、まだ接続されていない間欠リンクが存在することがわかった時に接続を試すことになる。

別の方法としては、先に述べたようにインターフェースにパケットが到着してからそのパケットが TCP を用いているかどうかを判断して発呼するという方法が考えられる。このときは、IP アドレスが自分のネットワーク以外のものかを判定するのを経路制御の機能に任せ、インターフェースにおいてパケットが TCP を用いているかどうかだけを判定するものである。インターフェースが受け取るのは、IP データグラムであるから、IP ヘッダから TCP を用いる通信かどうかを判断することが可能である。

インターフェースにパケットが到達するためには、実際に接続される以前から経路情報を流しておく必要がある。経路情報を流すだけではなく、インターフェースが起動状態であることもアナウンスしなければならない。つまり、従来通りの経路情報の機能を利用して、一種の正しくない経路情報を流すことになる。この方法では、経路情報の機能自体を変える必要がない点が利点といえる。但し、接続されれば通常の経路情報を流すので、パケットをインターフェースに集めるための経路情報と区別がつくような工夫をしないと、経路情報の扱いに混乱が生じる可能性がある。
区別をつけるためには例えば、接続された後でホップ数を変えることなどが考えられる。しかし、ルーティングの管理をする際に、ルーティングテーブルからはホップ数がわからないので外見上偽の情報かどうかの区別は難しい。

このように考えていくと、非連結な場合の間欠リンクの自動発呼の実現のためには、経路制御の機能や経路情報の意味に多少の変更は認めざるを得ないことがわかる。それならば、最小限の変更で済む方を選択するのが妥当である。そのためには、これらの方法について更に深く考察を進める必要がある。

4.5.3 切断

切断のタイミングについても、発呼の場合と同様に連結な場合と非連結な場合に分けて考える。

連結な場合

ネットワークの障害が回復、あるいは、混雑状態でなくなったならば間欠リンクは必要でなくなる。切断の場合は発呼の時ほど一刻を争う必要はない。また、現在の経路よりも優れた経路（ホップ数が少ないなど）についての情報は早く伝搬されるという性質があるので、ルーティングテーブルの参照により通常リンクの回復を検知して切断するだけでも切断のタイミングとして十分である。

ところで、発呼の際に通常リンクを監視する方法を取り入れるならば、監視は常に続けることになるので、切断のタイミングの検知もこれを利用することが考えられる。 通常リンクを監視する際注意すべき点は、間欠リンクが接続されている時には間欠リンクを通らないようにして監視しなければならないということである。従って ネットワークの障害からの回復は、通常リンクにより接続されている相手のネットワークへ通常リンクを通って、 到達できることにより検出される。

また、通常リンクの混雑状態からの回復は、標準時間内に到達できるようになることを検知すればよい。この場合は、いつ送ったパケットに対する応答かがわかるようにすることも考慮しなければならない。

非連結な場合

間欠リンクが必要でなくなるのは、主にユーザが相手のネットワーク上の資源を利用し終えた時である。

ユーザが相手のネットワークを必要としていないことを判断する条件としてまず考えられることは、ユーザがアプリケーションプログラムの実行を終了した時である。

相手のネットワークを利用するアプリケーションプログラムには、telnet[8]・rlogin[9] などがある。これらは、一度相手のネットワークへアクセスすると、ユーザがそのアプリケーションプログラムの実行を終えない限り (telnet や、rlogin を終えない限り)、ユーザの利用状況に関係なく、目的のホストとのコネクションを維持している場合が多い。そのようなアプリケーションプログラムでは、実行を始めてから終るまでの時間は、実際にユーザがそのアプリケーションプログラムを利用して作業を行った時間とは差が大きいこともある。従って、“アプリケーションプログラムの実行が終了した時” 以前にユーザが相手のネットワークを既に必要としていない場合もある。つまり、アプリケーションプログラムの実行の終了は、効率的な通信

[8] TELNET プロトコルを用いて他のホストと通信を行うためのユーザインターフェース
[9] 遠隔ログイン。ローカルホスト上の端末をリモートホストへ接続する。

を追求する間欠リンクの切断のタイミングとしては、不十分であると考えられる。

ISDN の回線交換を用いるこの間欠リンクでは、効率的に利用すれば常設リンクよりも安価となる時には間欠リンクを常設リンクの代わりに利用することも目指しているので、当然効率的な通信を追求するべきである。

理想的には、アプリケーションプログラムを用いた実際の作業時間の特性を導き、間欠リンクの切断のタイミングの決定材料としたいところである。しかし、アプリケーションプログラムの実行は、各ユーザが各々の利用目的に合わせて自由に行うもので一般的な作業時間の予測を行うことは非常に難しい。つまり、ユーザが間欠リンクを必要としなくなったという状況は、ユーザ自身にも不明確である場合が多いので、一般的なネットワーク機構のレベルから検知することは難しい。

ところで、ユーザがネットワークを利用するアプリケーションプログラムを実行している時には、インターフェースを通じてデータが流れる。しかも、ユーザがアプリケーションプログラムの実行中に流すデータは通常は一時的であることが多い。4.5.1 で述べたように、一時的に流れるデータは TCP パケットを用いることが多いという傾向がある。この傾向に着目し、インターフェースに出入りするパケットを監視し、TCP パケットの出入りが一定時間なければユーザが相手のネットワークを利用する必要がなくなったと判断する方法が考えられる。

この方法で考えなければならない問題点としては次のことが挙げられる。

1. 定期的に流れる TCP パケットと一時的に流れる TCP パケットをどのように区別するか。
2. パケットの出入りのない一定時間をどのように決定するか。

1 に関しては、定期的に流れる TCP パケットでは、パケットの長さがが短いことが 4.5.1 で述べたパケットフローの傾向として言える。従って、単位時間内に出入りする TCP パケットのパケットの長さの合計が一定量以上であるならばユーザが実際に作業をしていると判断することが考えられる。アプリケーションプログラムによって定期的に流れるパケットの長さの違いや、実行されるアプリケーションプログラムの数が関係してくるので、今度は単位時間や一定量をどのように決定するかが問題となる。それにはパケットフローのさらなる考察により実験を重ねる必要がある。

2 に関しては、一定時間の決定が効率の良い間欠リンクを実現できるか否かに重大な影響を及ぼすと考えられる。一定時間を決定して切断することによって生じる問題点は、ユーザがまだそのアプリケーションプログラムを用いて作業をする必要があるにも関わらず（つまり、相手のネット

ワークの利用を必要としている）切断されてしまうことである。つまり、ユーザの要求通りに間欠リンクを運用することができない場合があるということである。ユーザの要求が満たされない場合が頻繁になると、必要な時に接続するという間欠リンクの根本に矛盾が生じる。

非効率な通信は避けたいが、可能なかぎりユーザの要求通りに間欠リンクを実現できるよう工夫するべきである。そのためには、一定時間を固定するのではなく、動的に対応できるようにすることが必要である。一定時間を決定するための要素の例としては、次のような要素を挙げることができる。

1* アプリケーション毎の実行傾向

2* 課金の性質

1* は、例えば telnet であれば、実行直後○分程度はユーザが作業を行うことが多い、ユーザが△分程度作業を行わないときは telnet を終了する場合が多い、などの傾向である。このような時には、○分間はインターフェースに出入りするパケットを考慮せずに接続しておき、その後インターフェースに出入りするパケットを△分間を一定時間の基準とすると、一定時間が始めから固定されている時よりユーザの要求に合う可能性が高い。

一方、2* では、例えば 3 分で 10 円課金されるという場合は 3 分という課金時間と呼の接続時間のバランスを考える必要があるかも知れないということである。(Figure4.7参照）

地理的に離れた LAN 間の接続では特に、課金の性質を考慮に入れることは、効率の良い通信を提供するために重要なことである。

これらの要素を考慮するならば、発呼の際にアプリケーション側から、課金やアプリケーションの実行の特性の考慮を呼の制御部分に要求し、呼の制御部分が切断のタイミングを動的に決定することが 1 つの方法として考えられる。その場合には、呼の制御部分は、通信距離と課金の関係や、アプリケーション別の平均所要時間などを情報として持っていることが必要となる。しかしながら、一般的にどの程度パケットが到着しなければ間欠リンクが必要ないと判断してよいかという目安は必要になると思われる。一定時間を決定するのには多くの実験を重ね、見積もることが必要である。

4.5.4　着呼に対する処理

ISDN 網は近い将来規模が拡大し、公衆網として計算機以外の通信にも多く利用され、また ISDN 網に接続される計算機の数も増加の一途をたどることが予想される。そのため、通信をすることを予期しない計算機からアクセスされる可能性が高く、インターネットが悪用される危険性も高い。

現在インターネットへの不法なアクセスを防止するために、様々な認証の機能が考えられ実装されているが、ISDN を用いる上でもこのような機能が必要である。ISDN には、着呼に際し、発呼してきた相手側の ISDN アドレスの通知機能がある。これを利用して通信相手の ISDN アドレスをあらかじめ登録しておくことにより着呼は登録した相手のみを許可することが可能である。従って、通信相手を限定することによって下位レイヤで認証を行う。

4.6 非連結な場合の経路制御の最適化の問題

インターネットにおいて、ISDN を用いて通信を行うゲートウェイ（ISDN ゲートウェイ）が多数になってくると、複数の相手ゲートウェイが存在する可能性が生じる。ISDN による通信を用いなければ、他のネットワークと通信することができないような非連結な場合は、適切なゲートウェイを選択できるような経路情報を流さなければならない。(Figure4.8)

また、全体のネットワークが複数のパーティションからなり、しかも 1 つのパーティション内に複数の ISDN ゲートウェイが存在する場合 (Figure 4.9) には、最適なゲートウェイを選択することができるようにしなければならない。この場合には、呼が確立した時の経路情報を保存しておき、パーティション内部の経路制御や相手ゲートウェイの選択に役立てることが必要である。

4.7 ソフトウェアの構成

4.6 までに述べたように、ISDN 網に接続されるゲートウェイの IP アドレスは ISDN 網に与えられた 1 つのネットワークアドレスと、ISDN ゲートウェイの識別のためのホストアドレスからなる。そして、連結な場合でも、非連結な場合でもソフトウェアが発呼あるいは切断の必要性を検知し、発呼・切断を行なう。発呼・切断を行う方法はどちらの場合でも共通である。発呼・切断の動作を実際に行なうのはデバイスドライバであるが、これに発呼・切断を命令するモジュールを ISDN ゲートウェイ毎に 1 つ持ち、発呼や切断の必要性を検知したら、このモジュールに発呼や切断の命令を要求することにする。ISDN ゲートウェイが複数のポートを所有する場合には使用可能なポートへの発呼や切断の命令をしなければならないので、発呼や切断の命令を行なうモジュールはポートの管理も行う必要がある。また、デバイスドライバはポート毎に 1 つ存在するが、同一相手に複数のリンクを張った時はリンクの負荷分散をする必要があるので、負荷分散を

行なう機能を持つデバイスドライバも必要である。以上から、ISDN を用いた IP 接続のためのソフトウェア構成は Figure4.10のようになる。

各モジュールについて説明する。

4.7.1 isdn_controller

controller は次のような役割を担う。

ポートの割り当てと管理

controller は、そのゲートウェイの所有するポートをすべて管理する。そのため、controller は、ポートの状態表を管理し、これを参照して未使用のポートを割り当てる。

通信可能相手の確認

通信可能相手とは通信を許可する相手である。
通信可能相手は予めファイルに登録しておいて、これをもとに通信可能か、あるいは通信を許可してよいかを判断する。

通信可能相手の IP アドレスから ISDN アドレスへの変換

通信可能相手登録のファイルに書かれた IP アドレスと ISDN アドレスの対応から、IP アドレスを ISDN アドレスに変換する。

デバイスドライバへの発呼命令

ISDN アドレスを特定して発呼を要求する。
デバイスドライバとの通信では、/dev/isdn$N$ といった特殊ファイルを用いて行なうことにする。

デバイスドライバからの着呼許可要求に対する指示

デバイスドライバからの着呼許可要求に対して、通信可能相手からの着呼のみを許可する。isdn_controller は、複数のデバイスドライバからの着呼許可要求を処理しなければならないので、どのデバイスドライバからの着呼許可要求であるかを特殊ファイル、/dev/isdn$N$ の N から判断できるようにする。デバイスドライバとの通信に通常用いられる ioctl[10]を用いないのは、各ドライバからの着呼許可要求を受け付けられるようにするためである。

4.7.2 if_isdn_ll

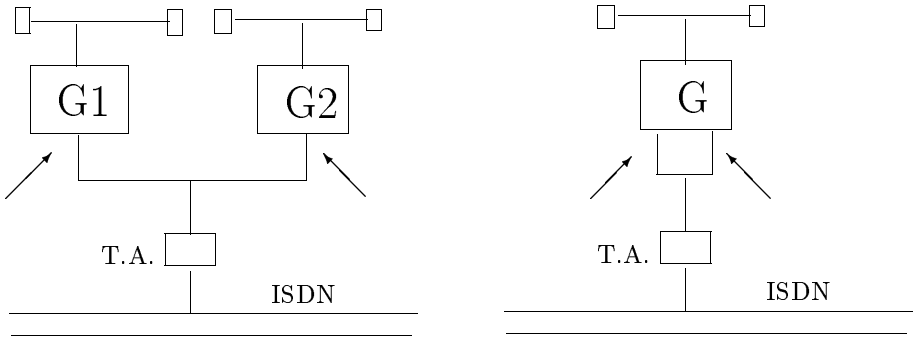
ISDN の物理的ポートに対応するモジュールでデバイス独立である。デバイスに依存する部分は zs_isdn などの ISDN 用ドライバにより実現する。

[10] UNIX のシステムコール。

if_isdn_ll は、isdn_controller の命令・指示により X.21[9] 手順に基づいた発呼・着呼の処理を行う。呼が確立すると、メッセージの入出力を if_isdn との間で行う。

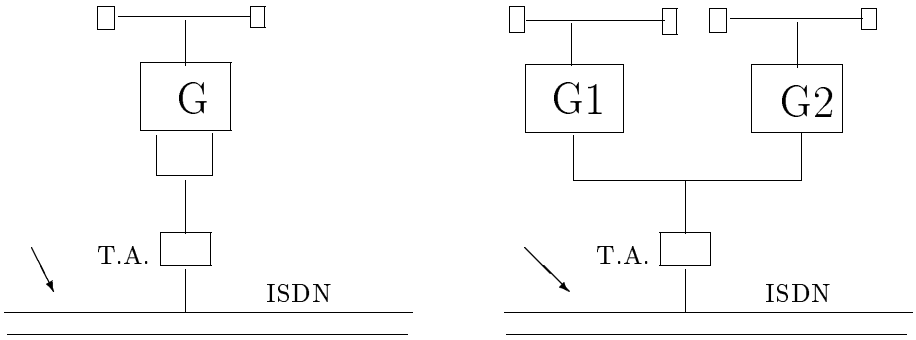
4.7.3 if_isdn

if_isdn_llが各物理的ポートに対応するモジュールであるのに対し、if_isdn は複数の物理的ポートを総括するモジュールである。主な機能は、呼が確立した後、IP モジュールから与えられたパケットを適切な if_isdn_ll に渡すことである。このモジュールは、将来的には同一相手に複数のリンクを張った際の各 if_isdn_ll への負荷分散の機能も受け持つ。また、現在は ISDN 網をマルチアクセスネットワークとして捉えているが、point-to-point ネットワークについてのプロトコル (PPP[53]) が整備されつつあるので、point-to-point ネットワークとしても扱えるように対応するのもこのモジュールの機能の 1 つになると思われる。



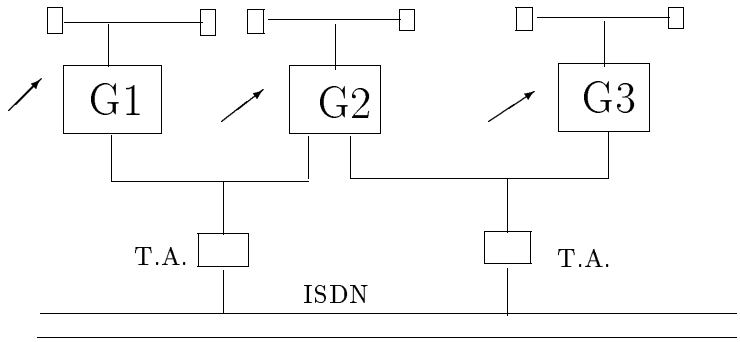
(a) ゲートウェイ:IP アドレス=1:1 の場合　(b) ゲートウェイ：IP アドレス=1:n の場合

図 4.1: IP アドレスの割当て方その 1



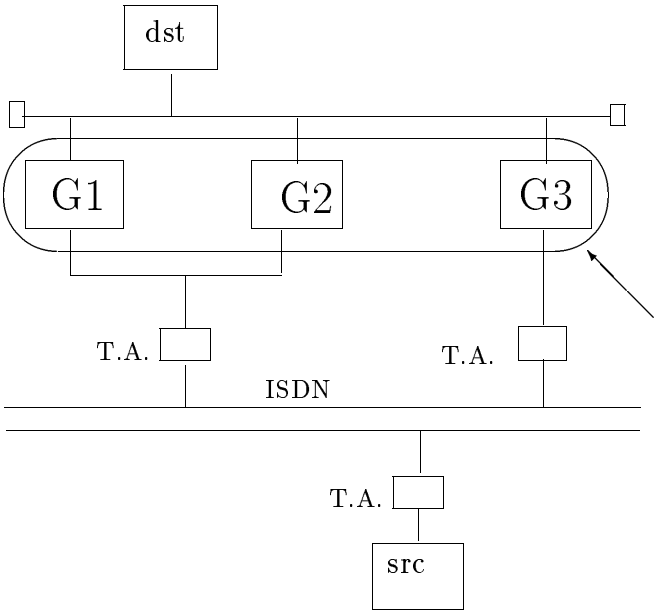
(a) ゲートウェイ:IP アドレス=1:1 の場合　(b) ゲートウェイ:IP アドレス=2:1 の場合

図 4.2: IP アドレスの割当て方その 2



ゲートウェイ:IP アドレス=1:1

図 4.3: IP アドレスの割り当て方その 3



ゲートウェイ:IP アドレス=1:1

図 4.4: IP アドレスの割り当て方その 4

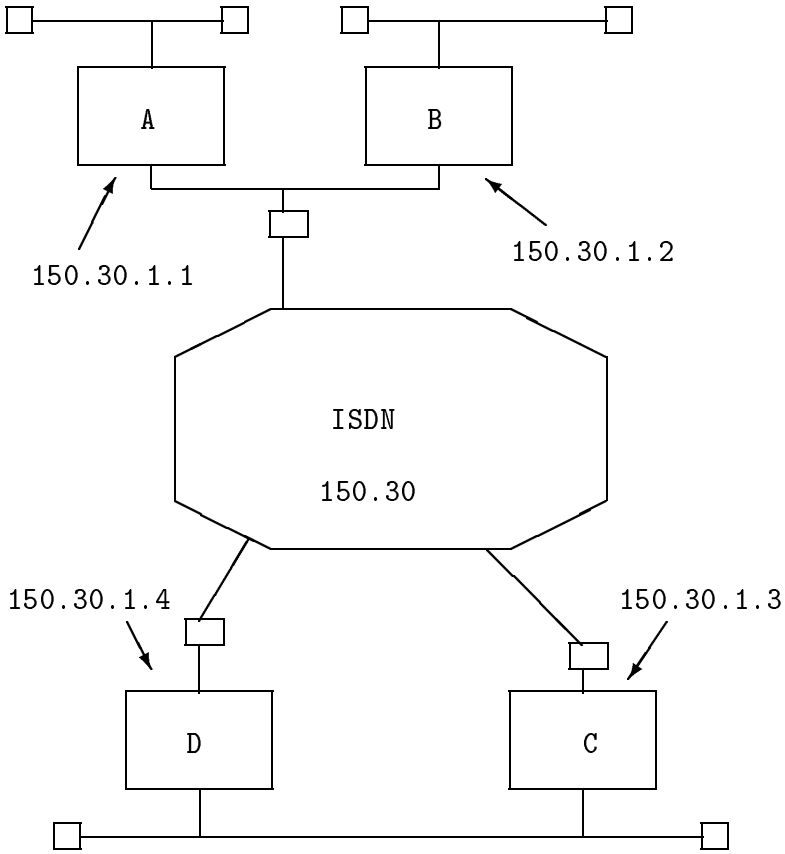
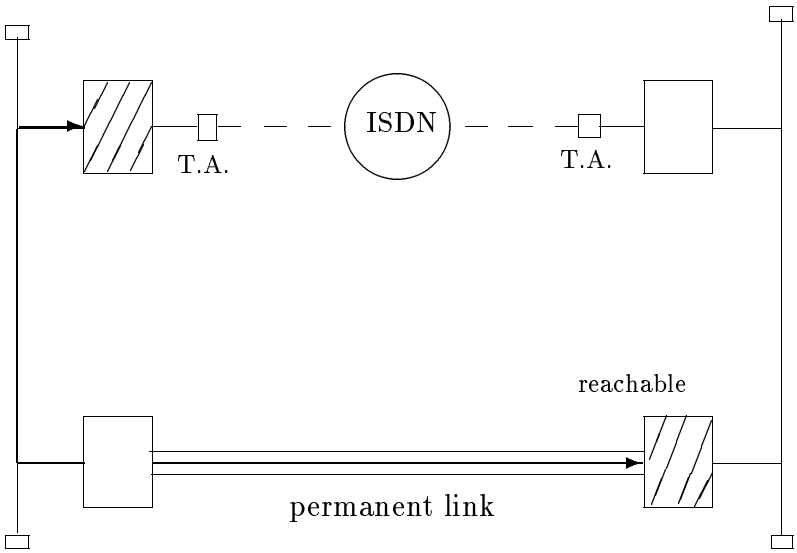
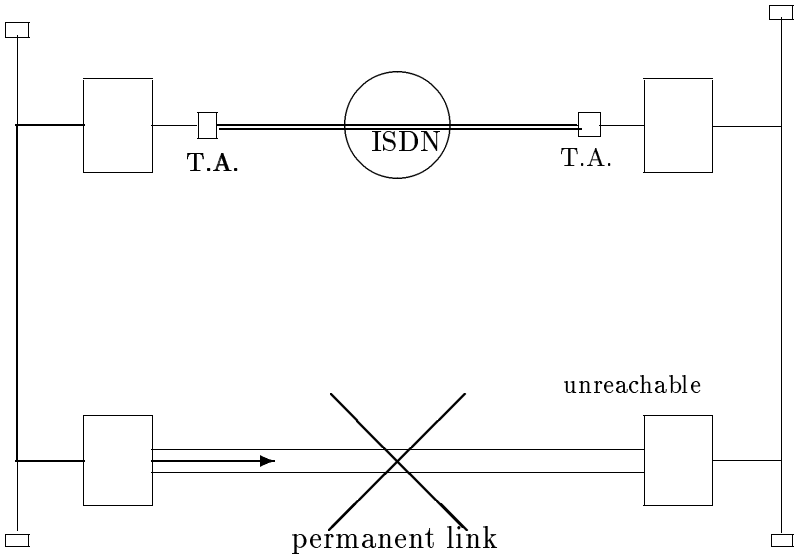


図 4.5: ISDN 網と ISDN ゲートウェイ



(a) 相手ネットワークへ到達可能な場合



(b) 相手ネットワークへ到達不可能な場合

図 4.6: 連結な場合の通常リンクと間欠リンクの切替

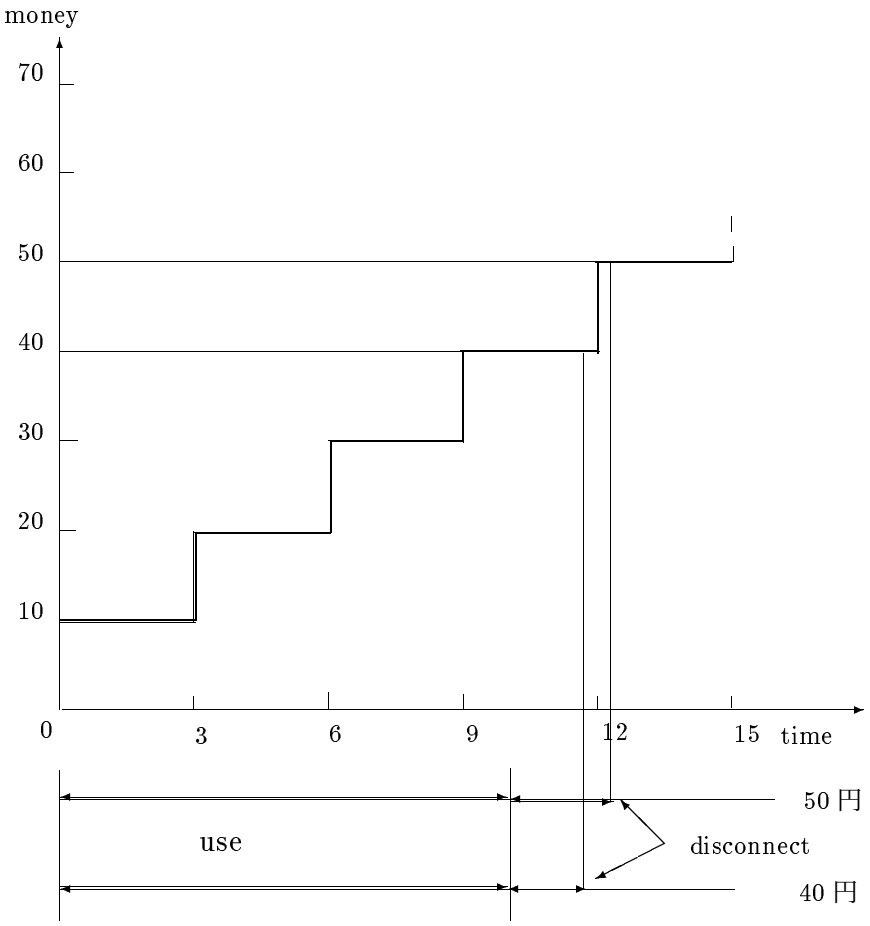
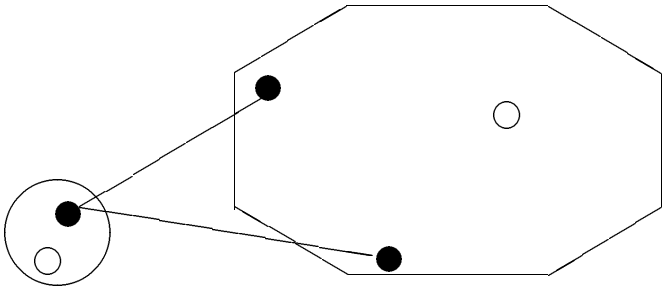
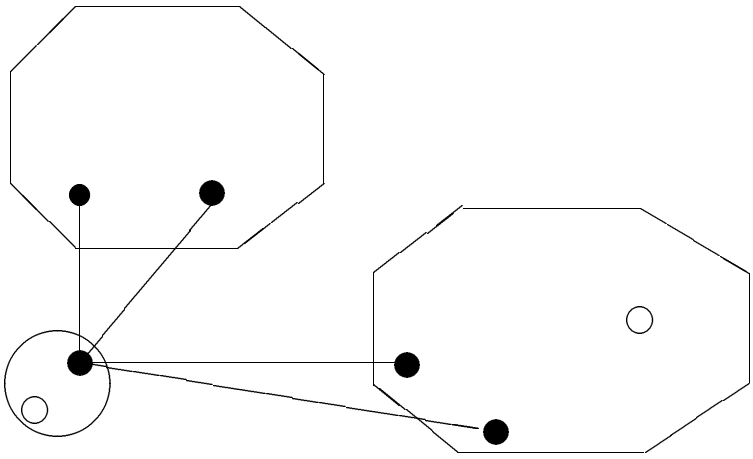


図 4.7: 通信時間と課金の関係の例



(a) パーティション内に複数のゲートウェイが存在する場合

図 4.8: 非連結な場合のネットワークパーティションの形状その 1



(b) 複数のパーティション内に複数のゲートウェイが存在する場合

図 4.9: 非連結な場合のネットワークパーティションの形状その 2

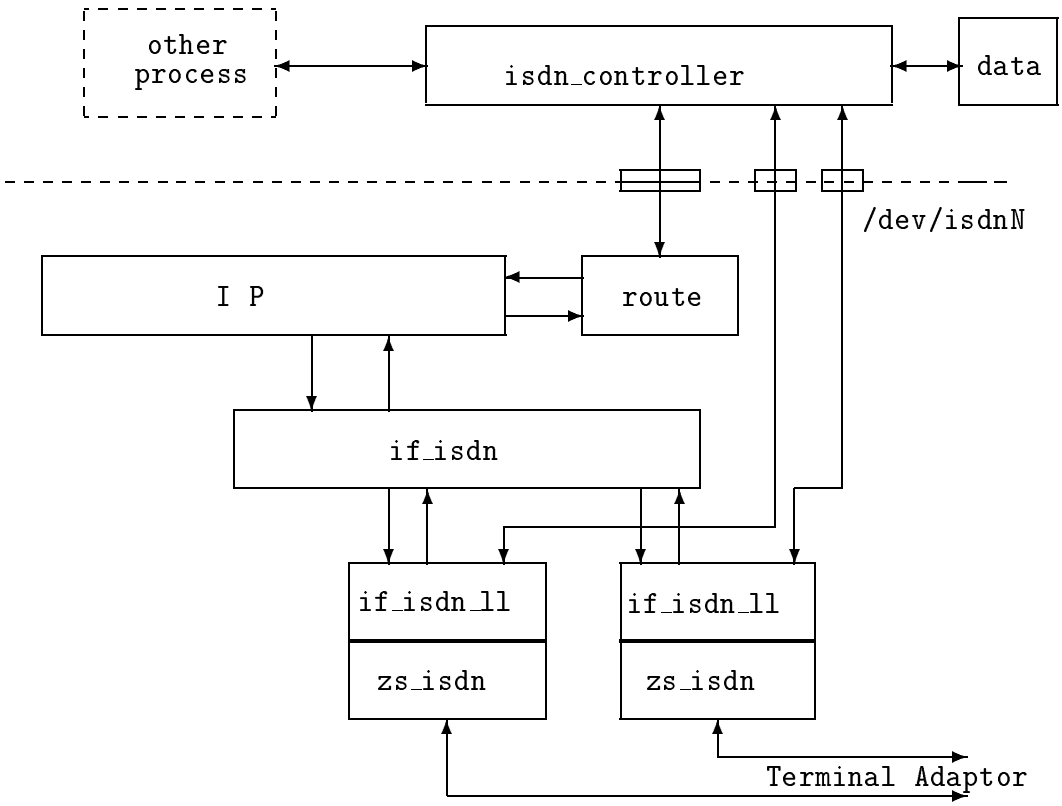


図 4.10: ISDN を用いた IP 接続のためのソフトウェア構成

第 5 章

呼の制御部分 (isdn_controller) の実装

第 4 章で述べた設計をもとに、呼の制御部分である isdn_controller の実装を行った。

5.1 実装環境

第 4 章で ISDN を用いた間欠リンクの他にリンクの存在する連結な場合と、存在しない非連結な場合とに分けて、呼の制御に関する議論を進めてきたが、非連結な場合で実験を行った。つまり、慶応大の三田キャンパス内の LAN と、東京大との接続を試行した。ISDN ゲートウェイは一対で、各マシンの収容する物理的ポートも 1 個という最も単純な環境である。

実装の際に用いる資源は次の通りである。今回の実装では、ISDN ゲートウェイが 1 対であるので、第 4 章で述べたように ISDN ネットワークとしての IP アドレスを割り当てるのではなく、point-to-point としてアドレスを割り当てることにした。また、デバイスドライバは、第 4 章で述べた構成のものとは異なるものを用いて、デバイスドライバと isdn_controller の間に device emulator を介して通信を行なった。

ゲートウェイマシン···SUN4/110 SUNOS4.0 と SUN3/260 SUNOS4.0.3
ISDN 用 Terminal Adaptor···NEC Aterm100
terminal adaptor との接続に使用されるインターフェース···RS-232-C

5.2 発呼及び切断の具体的実現方法

第 4 章で、ISDN を用いた間欠リンクの発呼や切断はソフトウェアが必要性を検知することにより実現すると述べたが、効率的な通信を提供するための発呼や切断のタイミングを決めるためには実験を繰り返すことが必要である。発呼に関しては、まずユーザが発呼要求をしてからパケットを流す方法から始める。切断については、次の 3 つのモードが考えられる。

- 手動切断モード
  ユーザが切断したい時に切断要求を出す

- タイマー切断モード
  接続時間を決めておき、時間が経過したら切断する。
  ユーザが接続要求を出す時に接続時間を宣言しておく。

- 自動切断モード
  インターフェースに一定時間 TCP のパケットが来なかったら切断する。
  ユーザが接続要求を出す時に自動切断にすることを宣言する。

このうち、今回の実装ではユーザが必要でない時に切断を要求するような ”手動 ”の方法をとる。段階的には、その後自動切断モードを実装し、切断のタイミングについて実験を重ねる方針である。
手動といっても次の方法が考えられる。

1: スーパーユーザなどの特定のユーザが発呼や切断についての責任を持ち、一般のユーザは呼が確立されている間通常のネットワークを利用するのと同じように利用する。

2: 間欠リンクを利用するユーザが各自で発呼や切断についての責任を持って利用する。

1:の場合は、特定のユーザが発呼や切断を行うので、isdn_controller は、特定の人からの要求のみを処理すればよいが、まだ他に間欠リンクを利用したいユーザがいるのに特定のユーザの判断によって切断されてしまい、思うように利用することができないという欠点がある。

2:の場合では、間欠リンクを利用するユーザがそれぞれ isdn_controller へ要求をすることが可能なので、ユーザは必要なだけ間欠リンクを使用することができる。この場合 isdn_controller は複数のユーザの要求を受け付けることになる。ただし、この場合は、一人でも切断要求するのを忘れると、リンクが切断されないということになるので通信の効率が悪くなるという欠点がある。

今回の実装環境のように非連結な場合、また、連結な場合では、発呼・切断のタイミングは異なるが、どちらの場合でも同一の isdn_controller を用いて発呼・切断ができるようにするには、isdn_controller は、複数の要求を受け付けることができるようにした方がよい。また、今回は isdn_controller の実装の初期段階であり、今後は実験を重ね、発呼や切断を自動化し通信の効率を改良していくことを考慮し、2:の方式を採用する。

すると、isdn_controller は複数のユーザからの要求を受けて、if_isdn_ll への必要最小限度の要求としてまとめる必要がある。なぜなら、1 つでも要求があれば呼は確立しておかなければならないし、要求が来た時に既に呼が確立されていたならば if_isdn_ll へ発呼を要求する必要はないからである。つまり、isdn_controller がユーザからの複数の要求を if_isdn_ll のレベルへの要求に対応させる。

切断については、一人でも ISDN を利用しているユーザがいれば切断することはできないので、切断要求数と発呼要求数が等しくなるまで if_isdn_ll には切断命令を出さない。切断についても、複数の要求を if_isdn_ll へ要求として対応させる必要がある。ユーザの要求と物理レベルの対応について Figure5.1に示す。

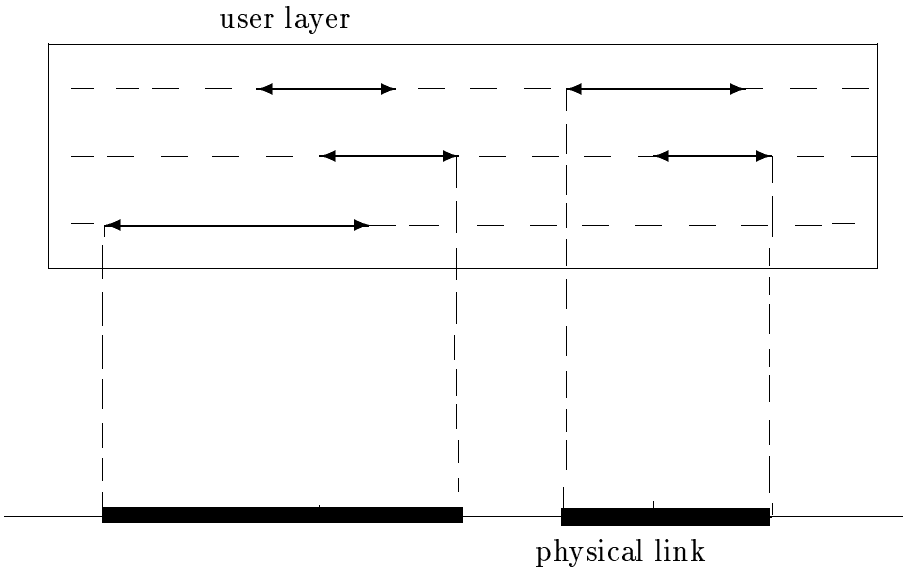


図 5.1: ユーザレベルと物理レベルの要求対応

isdn_controller に要求を出すのはユーザであるので、アプリケーション層のみから要求を出すように思われるが、呼の自動制御の際にはインターフェース層から要求を受け付けることもある。isdn_controller の受け付ける要求を、if_isdn_ll への要求として対応させる時は、isdn_controller に対する要求がどのレイヤから来るものかにはよらないので、対応付けは一段階でよい。

5.3　isdn_controller の詳細

5.3.1　isdn_controller に対するアクセス

isdn_controller には、Figure5.2に示すように、他のプロセスと、デバイスドライバの両方からアクセスを受ける。

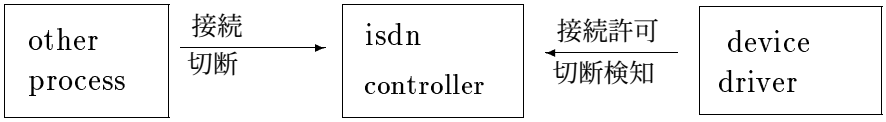


図 5.2: isdn_controller の受けるアクセスの種類

今回の実装ではデバイスドライバが 1 個の場合であるが、複数個の場合には isdn_controller は個々のデバイスドライバからの情報を待たなければならない。但し、個数はそのゲートウェイが持つ ISDN の物理的ポートの数であるので、固定的である。他のプロセスからは、どの程度のアクセスを受けるのかは動的であるので、要求を受け付ける通信口は 1 個とする。isdn_controller は、要求を受け付けた後処理を行なうので、複数の要求に対処するために isdn_controller としてのプロセスを複数にした。つまり、isdn_controller には親プロセスと子プロセスがあり、それぞれ次に示すような処理を行なう。

親プロセス

他のプロセスからの要求の受付
デバイスドライバからの情報の受付
通信相手の確認
ポートの割当て
所有するポートの接続状態に関する情報の管理
子プロセスへの要求受渡し

子プロセスの数は最大で所有する物理的ポートの数になる。親プロセスがあいているポートを割り当てると子プロセスを作り、その後の処理を子プロセスが行なう。子プロセスは呼を切断すると消滅する。

子プロセス

ISDN アドレスへの変換

デバイスドライバへの発呼命令
ポートの接続状態変更を親プロセスへ報告
ユーザレベルと物理レベルのマッピング
デバイスドライバへの切断命令

発呼・切断の際の親プロセスと子プロセスの関係を Figure5.3、Figure5.4 に示す。Figure5.3では、client が発呼を要求すると (2)、controller が子プロセス (child) を生じ、子プロセスが driver1 に発呼を命令する (3)。その後 driver1 への client(client to 1) が来ると子プロセスは要求を受け取り、ユーザ数を増やすだけで driver1 には発呼命令を出さない (4)。

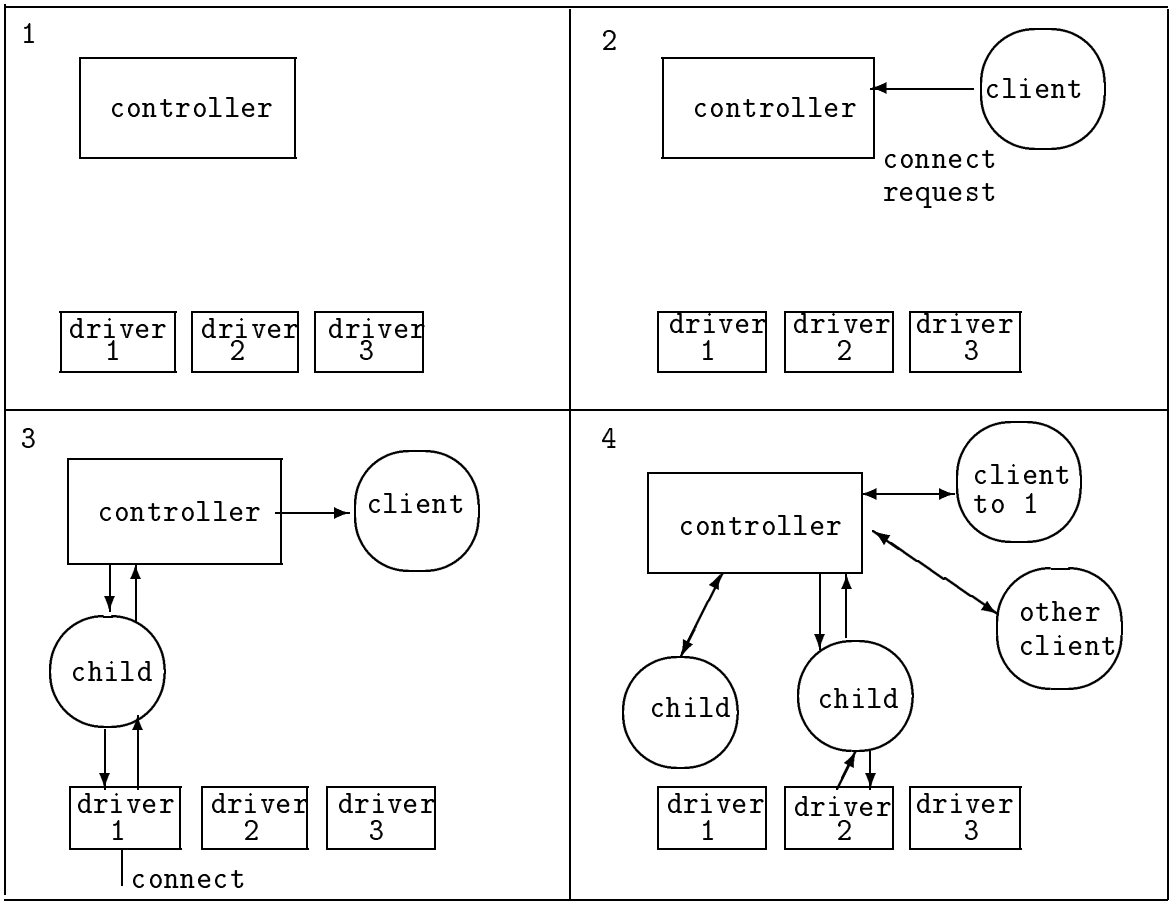


図 5.3: 発呼要求処理の際のプロセス間の関係

Figure5.4では、client が切断を要求すると (2)、driver1 を管理する子プロセスはユーザ数を減らし、ユーザ数が 0 になったならば driver1 に切断命令を出し (3)、親プロセスに切断を報告した後子プロセスが消滅する (4)。

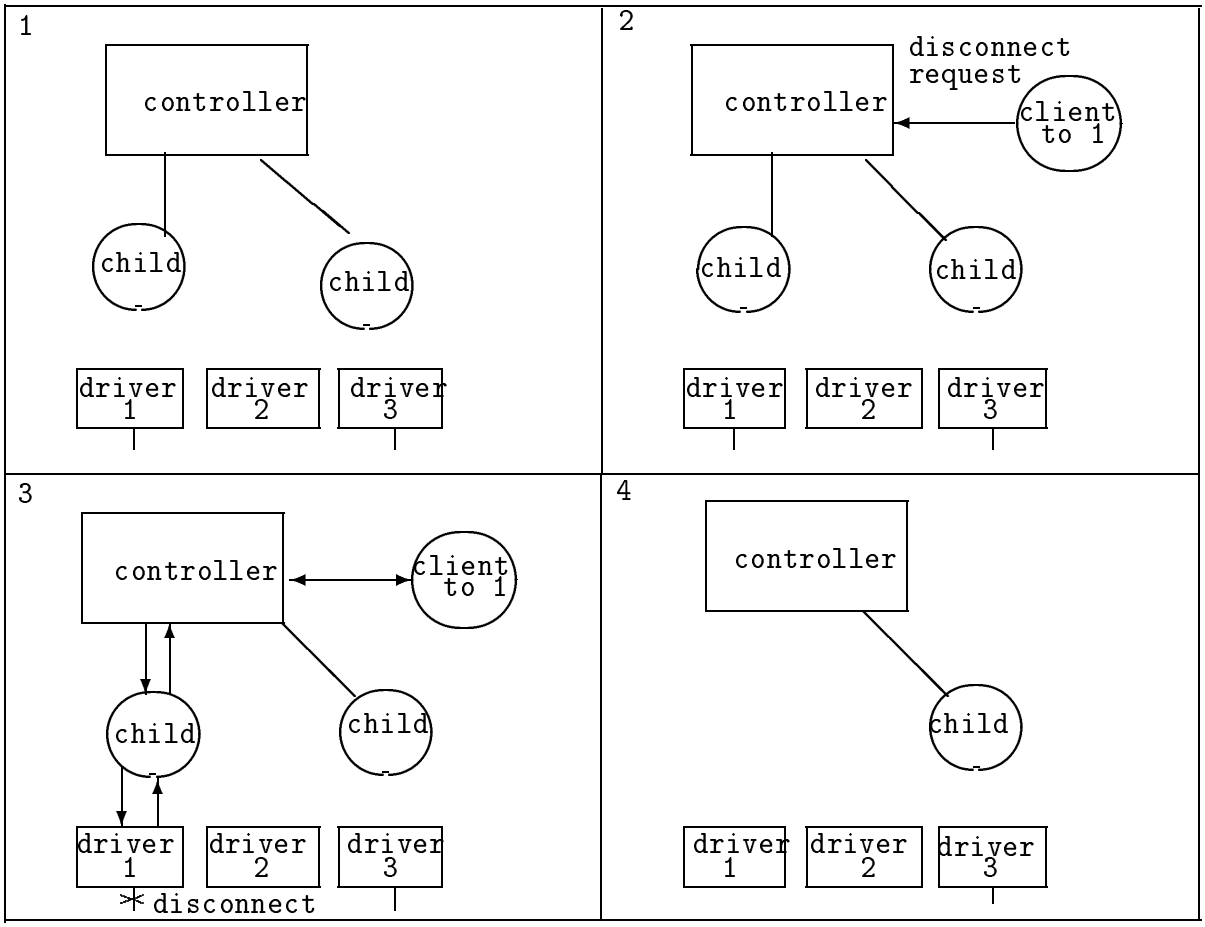


図 5.4: 切断要求処理の際のプロセス間の関係

5.3.2　発呼側と着呼側の処理の流れ

発呼側の isdn_controller と、着呼側の isdn_controller の発呼・切断に対する処理の流れをまとめると、Table5.1、Table5.2のようになる。

発呼側	着呼側
client 側による発呼要求 ↓ 発呼すべき相手の確認 ↓ 未接続の確認とポートの割当て ↓ ⟶ISDN アドレスへの変換 ↓ ↑ driver への発呼命令 ↘ ↓ ⟵ （通話中の場合） ▷ すべて通話中なら client に報告	↘ driver からの着呼許可要求 ↓ 着呼許可の判断 ↓ driver へ指示 ↙ ↘ 拒否 許可 ↙ ↓
client に報告 ⟵	↓
	↙
ポートの情報更新 ⟵ client に報告	⟵ ↓ ポートの情報更新

表 5.1: 接続の際の流れ

切断側	被切断側
client 側による切断要求	
↓	
切断すべき相手との接続確認	
↓	
driver への切断命令	
↓	
driver による切断の確認 ⟶	相手の切断の検出
↓	↓
client に報告	ポートの情報更新
↓	
ポートの情報更新	

表 5.2: 切断の際の流れ

5.3.3 isdn_controller と、それに対する client 間のインターフェース

client はそのプログラムの中で isdn_controller に対してアクセスをするが、この時の通信では、発呼要求に対する結果を受け取るまでに何度かのメッセージの交換があるので、アプリケーションレベルでの通信には TCP のソケットを用いている。

メッセージの交換は、client,isdn_controller が共に理解できるプロトコルを決めそれに従っている。しかし、1 つ 1 つの client のプログラムの中でこのプロトコルに従い、ソケットを作って isdn_controller と通信を行なうのは繁雑である。そのため、client が isdn_controller へ容易にアクセスできるようにするためにインターフェースとしての関数を用意することにした。この関数を用いると、client のプログラムが容易に書ける。

今回の実装における client は、手動切断モードにより発呼や切断の要求をするためのものである。この関数を用いて ISDN による通信を利用した様々なアプリケーションプログラムを作成することも可能である。

5.3.4 isdn_controller と、driver(if_isdn_ll) 間のプロトコル

isdn_controller から driver へ発呼や切断の命令を出したり、driver から isdn_controller へ着呼や相手側の切断を報告したりするのに、両者の間でプロトコルを定めて通信を行なう。driver はメッセージをビット列とし

て扱うので、複雑なプロトコルを使わずに、例えば 1 ビット目は『接続命令ならば 1』というように数字と意味を対応させるという扱いやすいプロトコルにした。isdn_controller と driver 間で行なわれる通信のメッセージのフォーマットを Table5.3に示す。今回の実装においては、isdn_controller と device emulator でこのようなメッセージを交換している。

メッセージの長さ
メッセージのタイプ （接続命令、切断命令など）
通信すべき相手の IP アドレス
通信すべき相手の ISDN アドレス

表 5.3: controller と driver 間通信のメッセージフォーマット

5.3.5 isdn_controller の所有するデータベース

通信相手の確認や着呼許可の判断のために、isdn_controller は、通信可能相手の登録表を持っている。これは、Figure5.5に示すような形式で書かれている。

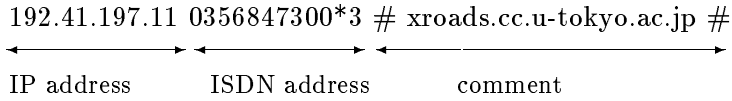


図 5.5: 通信可能相手登録表の一例

発呼側では、同一相手に対し複数の ISDN アドレスが対応する場合は ISDN アドレスの書かれている順番に成功するまで発呼命令を出す。

5.3.6 ポート管理に関する情報の要素

isdn_controller は自分のゲートウェイが所有するポートの管理を行なわなければならない。Table5.4にポート管理に必要な情報の構造を示す。

isdn_controller は、ポートが使用中か使用中でないかを見て、あいているポートを割り当てることや、使用中のポートがどの相手と通信を行なっているかを調べて、driver に無駄にアクセスするのを防止している。

識別番号
通信相手のインターネットアドレス
通信相手の ISDN アドレス
接続状態
デバイスドライバへのポインタ
通信開始時間
入出パケットの統計情報
ポートを管理する子プロセスへのポインタ
ポートを管理するためのスケジュール表へのポインタ （タイマー切断モード・自動切断モード用）

表 5.4: controller の持つポート管理表の項目

5.4　発呼側と着呼側の isdn_controller 間の通信

isdn_controller は、自分のところに来た要求の処理を行うが、各ユーザは、接続状態に関わらず接続要求を出すという方法であるので、着呼側の isdn_controller にも着呼側のユーザからの要求が来る。手動切断モードでは一人でも接続要求を出していれば切断しないことにしているので、発呼側の要求だけでなく着呼側の要求も満たさなければ十分とはいえない。従って発呼側の isdn_controller が発呼側に来た要求を処理するだけでは不十分であり、発呼側と着呼側の isdn_controller 間で通信を行い両側の要求を処理をしなければならない。

具体的に考えるとその際の通信方法としては 2 通りの方法が挙げられる。

1: 着呼側の isdn_controller は、自分のところに来た要求の処理を行わずに、その要求を発呼側の isdn_controller に渡す。従って、切断命令を出すのは常に発呼側の isdn_controller である。

2: 着呼側の isdn_controller も、自分のところに来た要求の処理を行い、どちらかの側が切断命令を出す。一方の isdn_controller が要求処理を終了したときは、他方の側の isdn_controller に要求処理の終了を報告し、他方の isdn_controller の要求処理が終了している時に切断命令を出す。

1 の場合の利点は、要求の処理を片方で行うので、切断命令を出す時には着呼側の状況を調べなくて済むことである。着呼側は発呼要求が来ても処理を行わないので子プロセスを生じない。

また、2 の場合には、発呼側と着呼側で行う通信を簡単な数字（例えば『要求処理終了は 1』）を用いて行うことができる。ただし、切断命令を出せる状態になっていても、すぐに発呼要求が来ることがあるので、要求処理の状態の変化を迅速に isdn_controller へ知らせる必要があると思われる。この場合は着呼側も要求処理を行うので、子プロセスを生じる。

第 6 章

評価・考察

6.1 パフォーマンスについての評価

エンド–エンド (アプリケーションレベル間) でデータの送受信にかかる時間を測定するプログラムを作り RTT(Round Trip Time) を測定した。UDP ソケット・TCP ソケットを用いた場合についてそれぞれデータ量を変えて測定を行った。比較のために、東京大と東工大、東京大と慶応大の間の 64kbps の専用線についても同様の測定を行った。グラフを FigureB.1、FigureB.2 に示す。

まず、UDP ソケットを用いた場合 (FigureB.1) では、ほぼ同じ RTT が示されており、ISDN が専用線に匹敵する通信を提供することができるということがいえる。

また、TCP ソケットを用いた場合 (FigureB.2) では、ISDN を用いた場合の方が転送速度が若干早いことが示されている。しかし、専用線と ISDN では、現在は ISDN を用いる場合でトラフィックが著しく少ないので専用線でトラフィック量が近い状態を作り出し再度 RTT を測定した (FigureB.3 参照)。専用線では 0.02 秒から、0.1 秒毎にパケットが流れていると観察されたので ISDN の方でも 0.02 秒、0.03 秒、0.04 秒、0.1 秒という間隔でパケットを流しておいて測定した。この結果、0.02 秒間隔でパケットが流れている時は RTT が他の場合の 2 倍となり、転送に支障があるがそれ以外は専用線と同様の混雑状況でも専用線と同程度の RTT であることがわかった。

また、エンド–エンドでなく IP のレイヤ間での転送速度を IP のレイヤの icmp を用いる echo サーバを用いて測定した。その結果を FigureB.4 に示す。これより、IP のレイヤでも専用線との差はないと見ることができる。つまり、ISDN が IP に接続されれば 64kbps の転送速度は十分生かされているといえる。

6.2 IP レベルでの通信開始時間の改善について

物理的な接続にかかる時間は数秒程度であるが、IP レベルで通信可能となるまでに約 30 秒かかる。これは、経路情報の伝搬が遅いことが原因であると思われる。今回の実装では、ISDN の他にネットワーク間にリンクが存在しない場合であるので、間欠リンクが接続されない時には経路情報が流れない。接続された時に初めて経路情報を流し始めると経路情報が伝搬するのに時間がかかる。

IP 接続までに約 30 秒かかるとすれば、パケットが届いてから発呼をする自動発呼での影響が大きいといえる。自動発呼の実現のためには IP レベルで通信可能となるまでの時間を短縮することは重要である。通常、経路情報は 30 秒毎に出されることを考えると、物理的な接続をしたときのタイミングによって次の経路情報が出されるまでの時間が異なるので、IP レベルでの通信可能までの時間が左右されるといえる。従って、物理的接続が確立すると同時に経路情報を流すということが考えられる。

最終的には、IP レベルで通信可能になるまでの時間を考慮してタイムアウトを決め、自動発呼までに来るパケットをまとめてとっておく場所が必要であると思われる。

6.3 ISDN を用いたリンクのパケットのフローパターンについて

第 4 章で行なったパケットのフローパターンの考察は常設リンクの観察によっていたが、ISDN を用いたリンクについて同様の観察を行なった。現段階では実験中ということもあり ISDN を用いたリンクのトラフィック量は少ない。従って、常設リンクの場合よりも単純なパケットフローが得られ、特徴が捉えやすくなっている。

グラフを FigureA.3、FigureA.4に示す。UDP のパケットは断続的・定期的ではあるが、TCP を用いた断続的・定期的なパケットに比べ、パケットの流れる時間間隔が短いことが読みとれる。このグラフでは TCP を用いた断続的・定期的なパケットは約 110byte であることもわかる。この 110byte のパケットは、rlogin という１つのアプリケーションプログラムの実行中に定期的に流れている。他のアプリケーションプログラムについても、定期的なパケットが流れるのか、あるいは流れないのかを知るために、ネットワークを利用する個々のアプリケーションプログラムに焦点を当ててその特性を捉えることが重要である。さらに、特に自動切断のタイミングを決める時に必要な、アプリケーションプログラムの実行傾向を捉えるため

にもこのような考察が必要である。

6.4 PPP(point-to-point protocol) への対応への考察

4.3 では、ISDN 網をマルチアクセスネットワークとして捉えると述べてきたが、この時 point-to-point ネットワークとして捉えなかったのは多くのネットワークアドレスを必要とするため、ネットワークアドレス数が足りなくなる恐れがあるからということであった。

point-to-point ネットワークは相手が固定的であるため、相手との通信に用いられるプロトコルは多種多様で一貫性があるとは言えなかった。最近、point-to-point protocol(PPP) により、point-to-point ネットワークで用いるプロトコルが整備されつつある。この中には、point-to-point ネットワークのアドレスを相手側と交渉することにより動的に割り当てるというプロトコルもある。これらを用いるとネットワークアドレス数の節約が可能となり、ISDN 網を point-to-point ネットワークとして取扱うことも可能である。但し、その場合には ISDN 網へのインターフェースを通信相手数分用意するのか、所有するポート数分用意するのかといった点も考えていく必要がある。point-to-point として捉えた場合でもどのポートを使用するかを動的にするならばポートを割り当てる controller は必要であると思われる。

6.5 ISDN 普及に伴う通信相手増加への対策

通信相手が増加するとポートの使用率は高まり、ポートの確保が問題となる。通信相手の登録はそれぞれのゲートウェイのポート数を考慮して登録する必要があるかも知れない。例えばポート数が 1 であるのに通信可能相手数を 100 とすることには無理がある。逆に通信可能相手を多数とするならば、それに見合うポート数を用意することが肝心である。通信可能相手の決定には、ネットワークの運営上の立場やネットワークの利用目的等により、各ネットワークの管理者同志の合意が必要となる。管理者はポート数をある程度考慮して通信可能相手を決定することも必要である。

通信相手増加に対する技術的な対応としては、ポートが全て使用中の時点でさらに新しい接続要求が来た場合には、使用中のポートの中で次に挙げるような要素のどれかに該当するポートを 1 つ選択して、そのポートの使用を終了し、新しい要求に対応することも考えられる。

- 他のポートと比較してトラフィック量が少ないポート。

- 接続時間が一定時間を越えているポート。

各ポートの使用状況の差が小さければ、新しい要求には対応しない方がよい場合もあると思われる。

第 7 章

今後の課題

7.1 ネットワークマネージメント機構の利用

ネットワークの障害を検知するのに、独自の機能を作ることも可能であるが最近ネットワークマネージメント機構が構築されており、様々なネットワーク監視機能が存在する。このようなネットワークマネージメント機構を利用してネットワークの障害を検知し、代替リンクを起動させることは有効である。間欠リンクをインターネットに統合していくという目的のためにもインターネットの他の機能を利用することは意義のあることである。

従って、連結な場合、通常リンクの障害や混雑を検知してから発呼する際にはネットワークマネージメント機構を利用して簡単なアプリケーションプログラムとして実現したい。

7.2 ポートの優先順位を考慮した controller の拡張

通常リンクのカバーとして間欠リンクを用いるときには必要な時はいつでもポートが確保されていたほうがよい。従って、通常リンクのカバーのために間欠リンクを用いる要求が来たら、ポートが全て使用中であっても優先的にその要求を通すようにするなどの工夫が必要となることがある。

また、複数のポートを収容する場合は、通常リンクのカバーを優先に用いるポートと、そうでないポートを分けて使うなどポートに役割を付ける工夫が必要である。ポートの管理は isdn_controller が行うので、ポートの優先順位付け・役割付けにより isdn_controller の機能が拡張される。しかし、ポートの優先順位付け・役割付けの具体的方法は各 ISDN ゲートウェイによって異なるので、ポートの役割付けを自由に設定できるような方法を提供する必要がある。

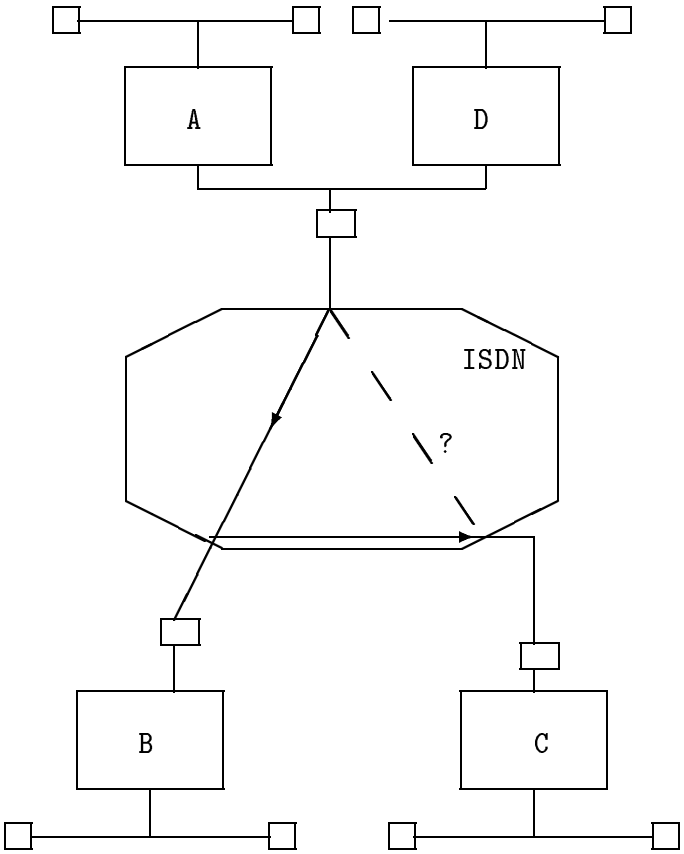


図 7.1: multihop-link の例

7.3 multihop-link の実現に関する問題点

ゲートウェイ A とゲートウェイ C を ISDN を用いて接続する時、ゲートウェイ A とゲートウェイ B、ゲートウェイ B とゲートウェイ C がそれぞれ ISDN によって接続されているならば、新たにゲートウェイ A とゲートウェイ C を接続しなくてもゲートウェイ A とゲートウェイ C の間にゲートウェイ B を間接ゲートウェイとして通信を行うことが考えられる。

このような multihop-link をサポートするためには、通信可能相手に関する接続状況を把握することが重要である。接続状況は動的であるので、経路情報のように、接続状況についての情報を定期的に流して管理することが必要になると予想される。応用的には、ゲートウェイ A がゲートウェイ C と接続しようとする時、ゲートウェイ B とゲートウェイ C が接続され

ており、ゲートウェイ C の物理的ポートが全て使われている時には、ゲートウェイ A はゲートウェイ B に接続することによって、ゲートウェイ C と通信を行なうことも考えられる。

multihop-link を扱う際の問題点は、認証である。ゲートウェイ A がゲートウェイ B と、ゲートウェイ B がゲートウェイ C と接続されている場合、ゲートウェイ A とゲートウェイ C が接続可能相手でなければ、ゲートウェイ A とゲートウェイ C の通信は不可能となるようにしなければならない。そのためには、ISDN ゲートウェイは ISDN を通じてきたパケットの経路を監視し, 自分の通信可能相手以外の ISDN ゲートウェイを経由していればそのパケットを通さないようにする必要があると思われる。

また、multihop-link を積極的に応用して故意に ISDN ゲートウェイを複数介して目的のネットワークへ到達することも考えられる。これは、目的のネットワークの ISDN ゲートウェイが他のネットワークと通信中である場合やその ISDN ゲートウェイが故障している場合などで有効であると思われる。最適経路でなくても接続可能となることを優先したいことがあるからである。このようなことを考えると間欠リンクを必要とする場合の経路制御では、isdn_controller が所有する通信可能相手の表から得られる到達可能なネットワークも情報として取り入れる必要があると考えられる。

7.4　回線交換とパケット交換の使い分け

ISDN のパケット交換サービスが提供されれば通常はパケット交換を利用し、通信量が多くなったら回線交換を利用するというリンクの使い分けが可能となる。パケット交換を利用することにより非連結な場合でも経路情報を流すことができるので、回線交換に切替えるのに必要な時間は、パケット交換を利用しない時よりも改善されると予想される。回線交換とパケット交換の使い分けでは通信量がどの程度になったら回線交換に切替えるかを決定し、常にリンクの通信量を監視する機能が必要である。これは、自動切断の方法に新しい機能を加えれば実現されると思われる。パケット交換サービスの提供が待たれるところである。

付録 A

パケットフローの例

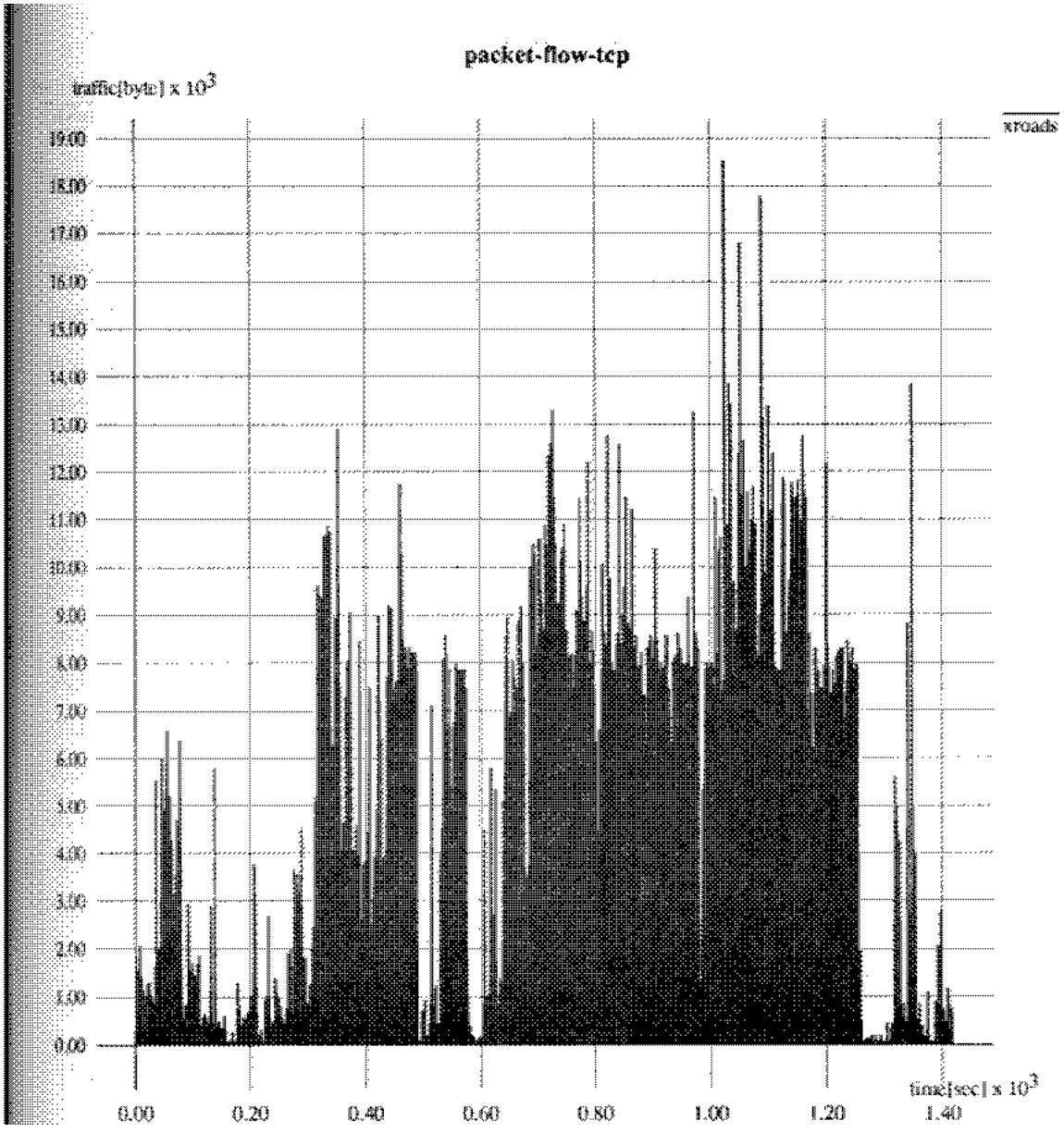


図 A.1: TCP に着目したパケットフローの例（イーサネット）

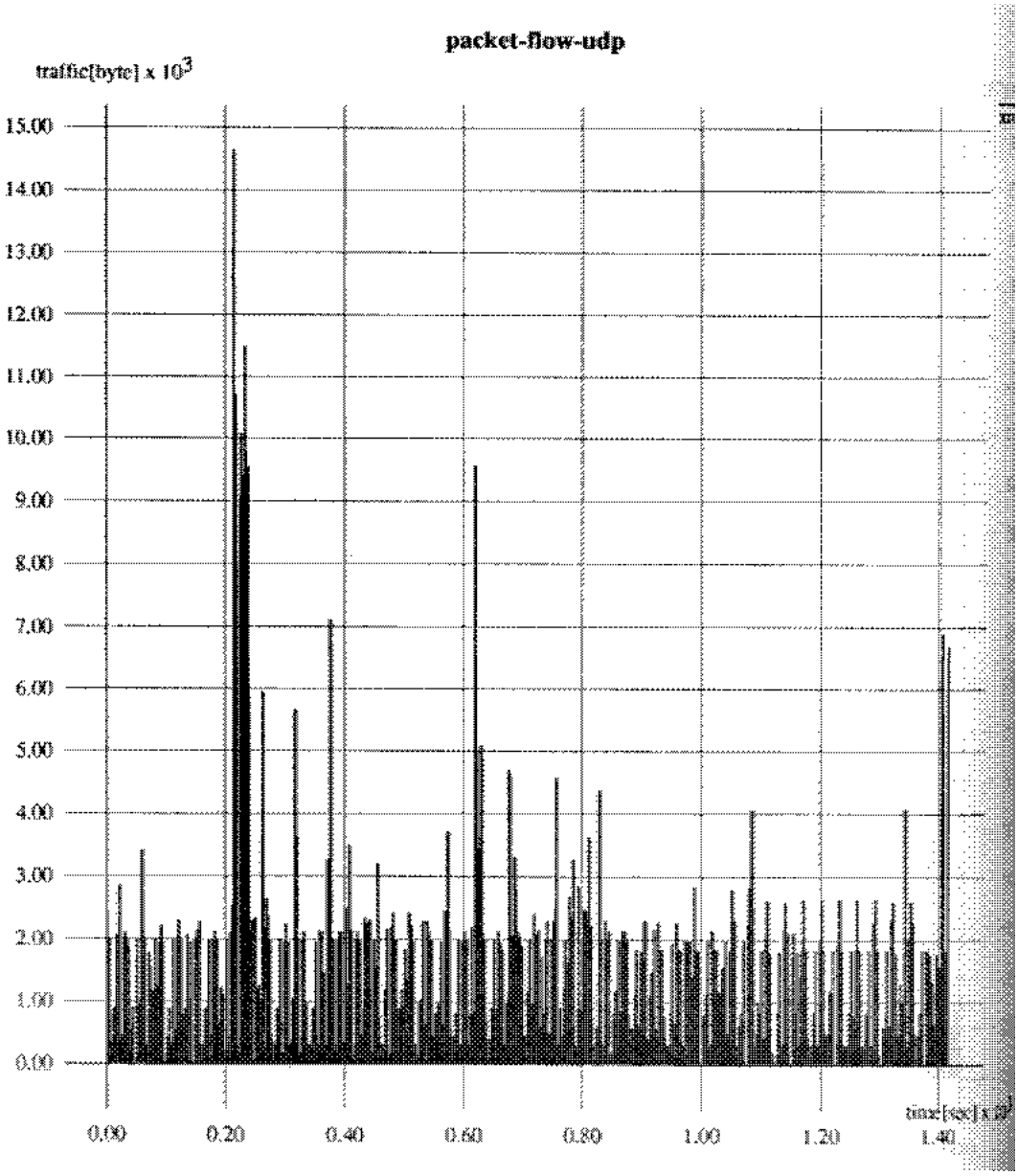


図 A.2: UDP に着目したパケットフローの例（イーサネット）

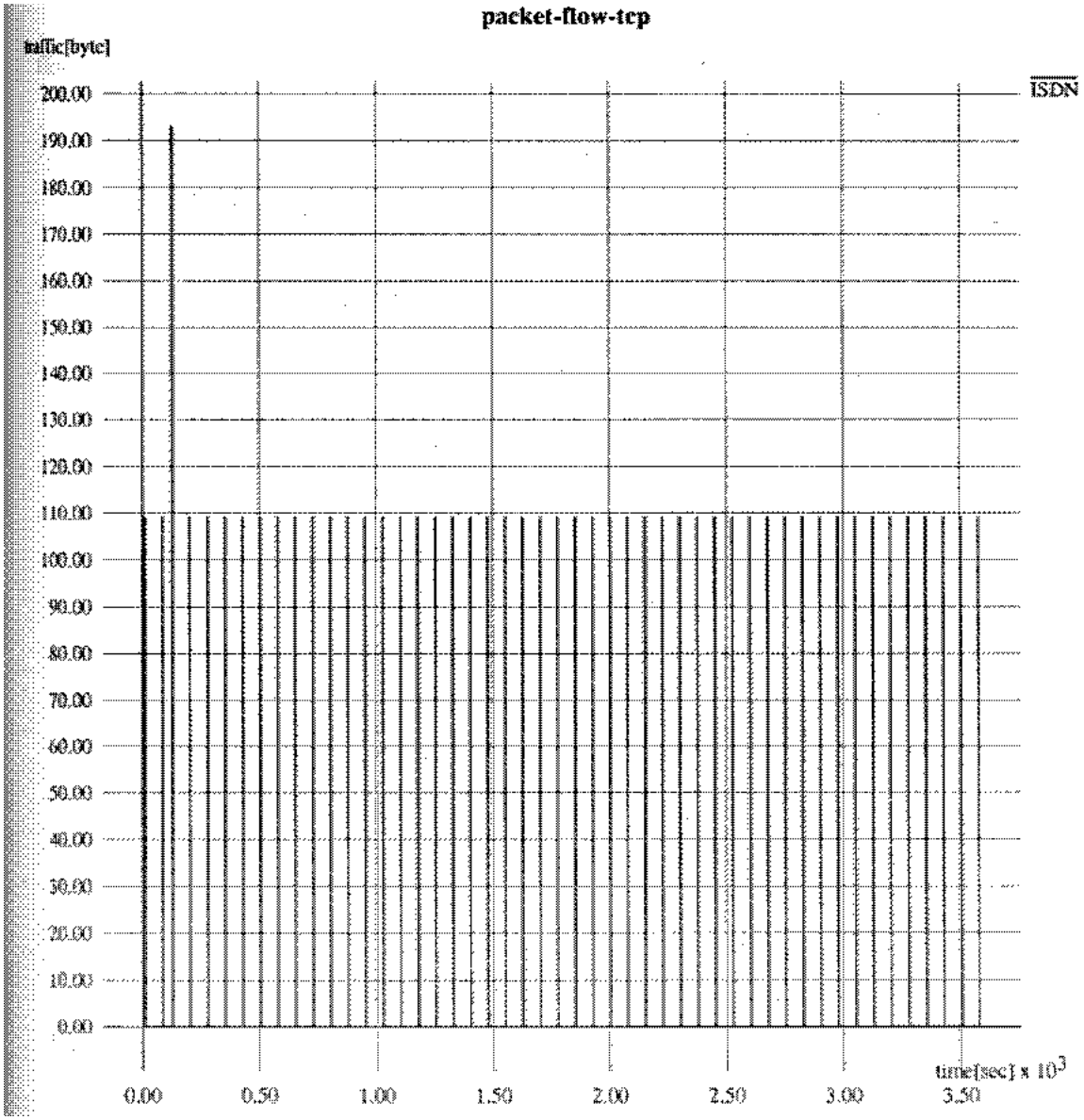


図 A.3: TCP に着目したパケットフローの例（ISDN）

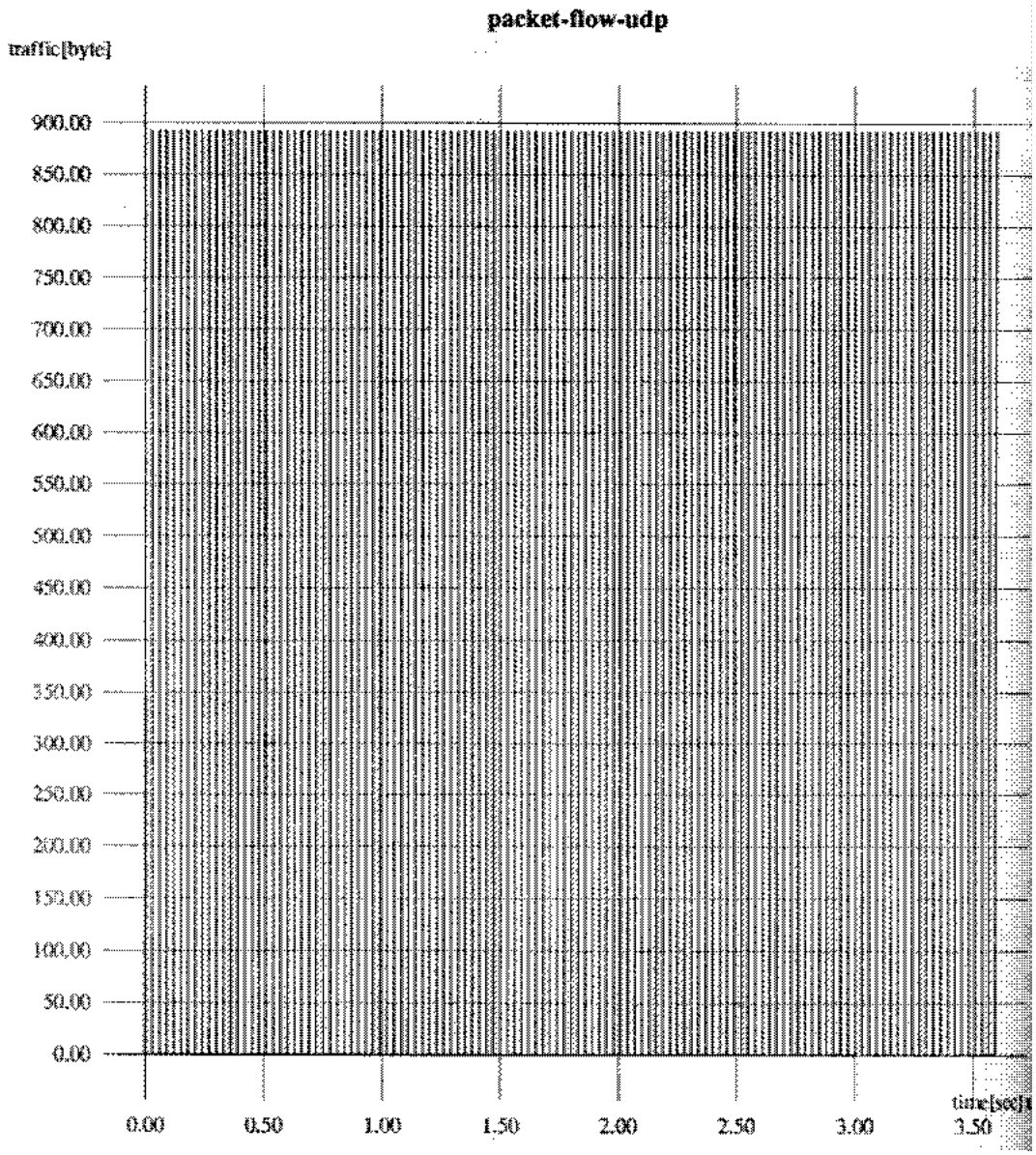


図 A.4: UDP に着目したパケットフローの例（ISDN）

付録 B

ISDN を用いた IP 接続のパフォーマンス

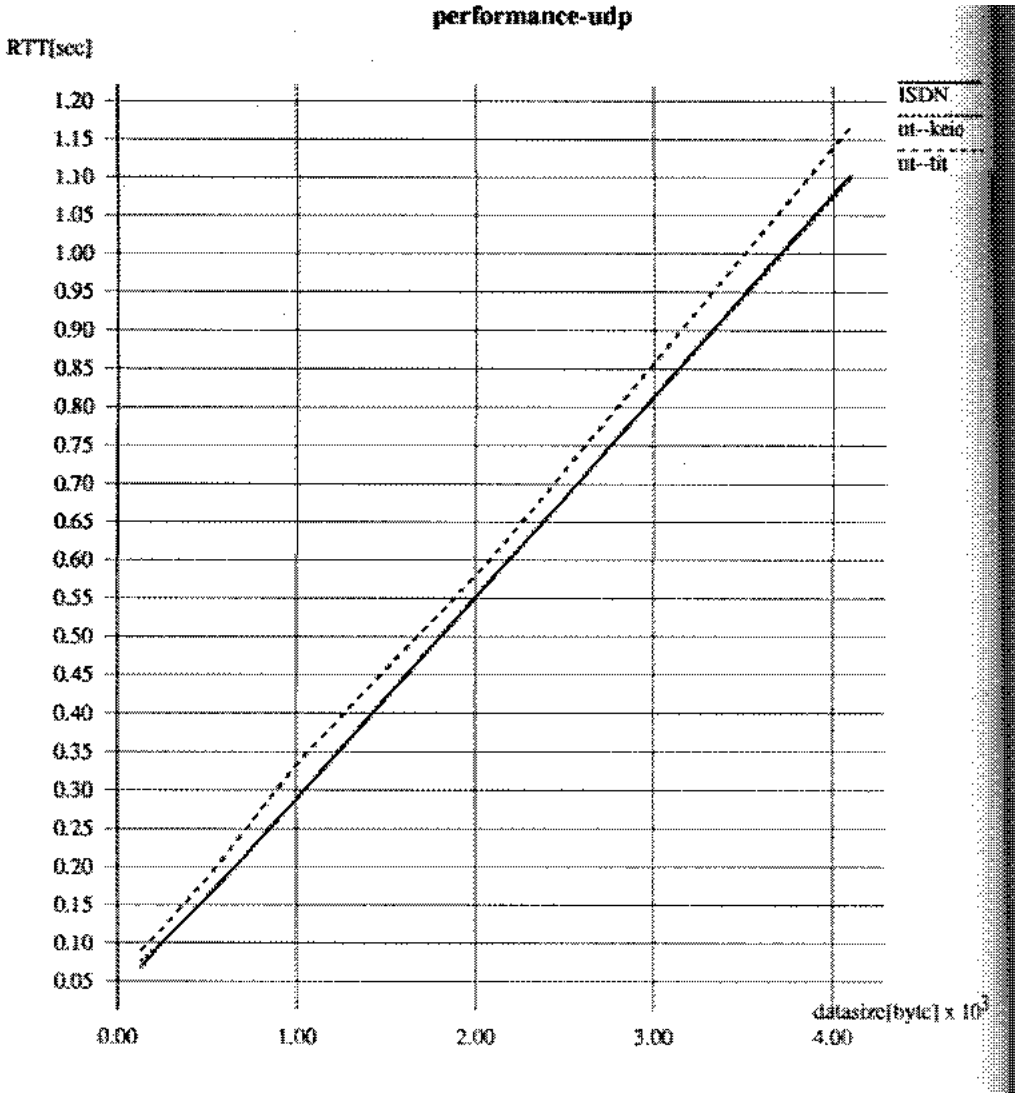


図 B.1: UDP ソケットを用いたエンド–エンドのパフォーマンス

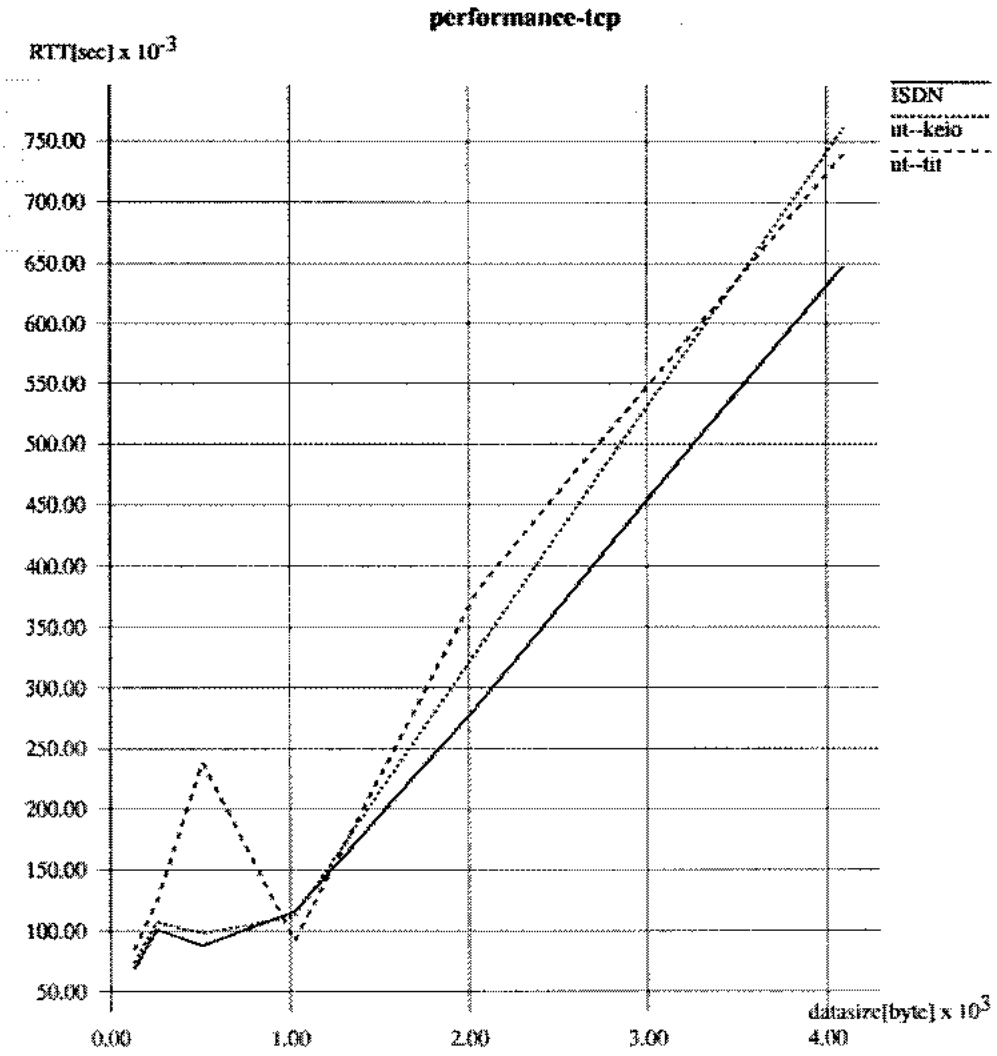


図 B.2: TCP ソケットを用いたエンド–エンドのパフォーマンス

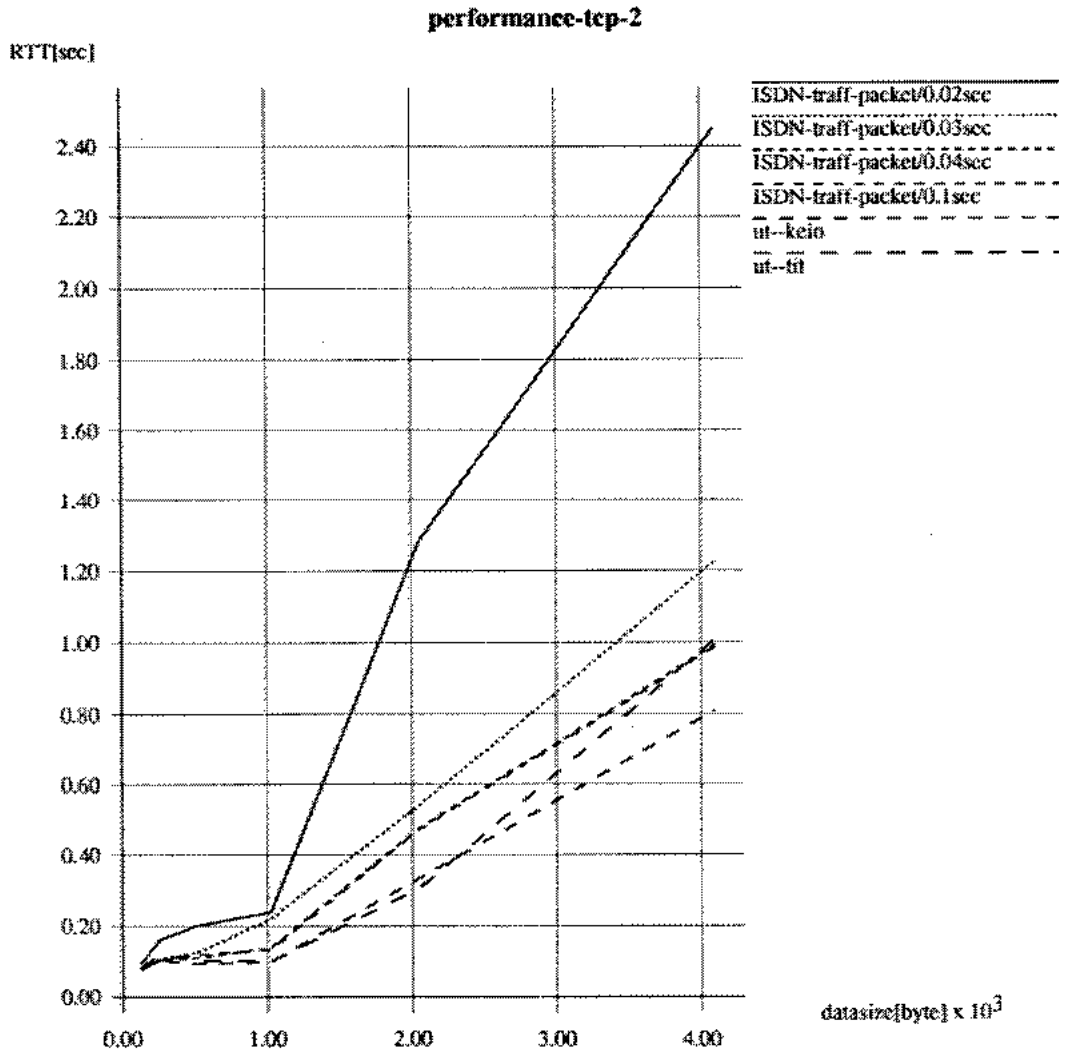


図 B.3: トラフィックを考慮し、TCP を用いたエンド–エンドのパフォーマンス

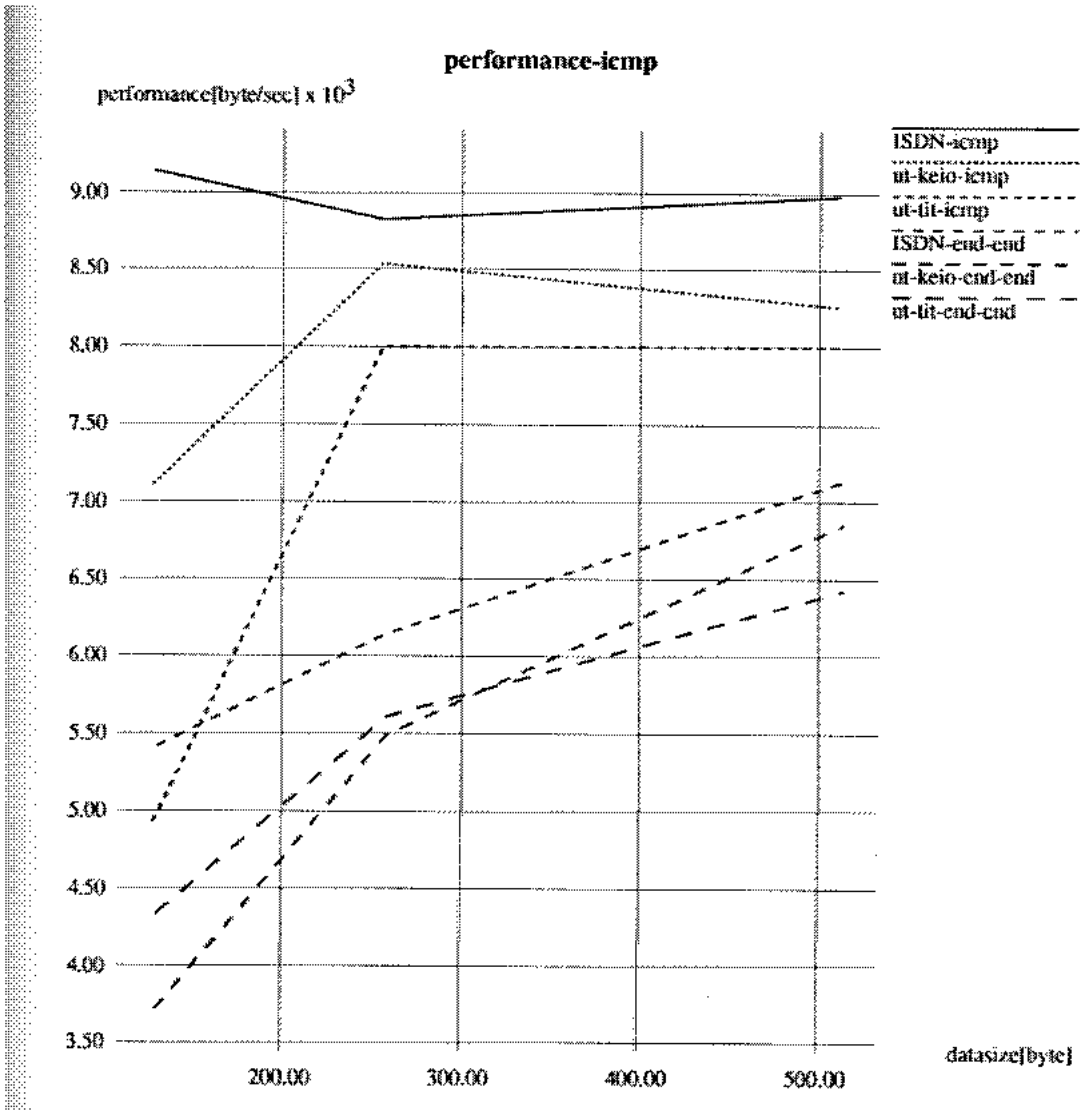


図 B.4: ICMP を用いた IP レベルのパフォーマンス