

AppleTalk と TCP/IP の相互接続の試み

田井村明博*・鶴 正人**・木村 広***

1993年10月29日受理

Internetworking between AppleTalk and TCP/IP

Akihiro TAIMURA*, Masato TSURU**
and Hiroshi KIMURA***

1 はじめに

ネットワークのないコンピュータはさびしい。コンピュータの利点は、大量のデータを迅速かつ正確に、収集、加工、伝達できることである。どんなにコンピュータの処理能力が高くても、ネットワークにつながっていなければ、有用なデータを迅速に収集あるいは伝達することができない。逆に、コンピュータの能力が相対的に低くても、ネットワークにつながっていれば、しかるべきコンピュータ、あるいは遠隔地の人的資源と通信しながら作業を効率的に進めることができる。ネットワークにつながることによってコンピュータの使い勝手、そしてわれわれの生産性は格段に向上するといつてよい。

長崎大学教養部で稼働中の AppleTalk ネットワークには、11台のマッキントッシュ（以下マックと略）、3台のレーザプリンタがつながっており、ファイルやプリンタの共有、オンラインメッセージの交換などが可能となっている。

しかし、その教養部 AppleTalk ネットワークは、教養部外部とのコネクションがなく、いわば、孤立したネットワークであった。外部のネットワークと交信するためには、シリアル回線を使って長崎大学総合情報処理センターのマシンにログインするしかなかった。

そのシリアル回線の通信スピードは最高9.6Kbps であり、AppleTalk の230Kbps と比べて20分の1以下である。また、シリアル通信を実現するためには、モデムに相当するデジタルフォン（約10万円）を別途購入しなければならず、通信速度に対する割高感はぬぐえない。さらに、シリアル通信では、通常、同時にはただひとつのプロセスしか通信できないから、最近はやりのマルチウィンドウ環境を生かしたコンピュータネットワークの利用は望めない。

現在の利用形態を維持しつつ、さらに効率的なコンピュータおよびネットワークの利用を可能とするために、われわれは教養部 AppleTalk ネットワークを TCP/IP ネットワークに接続する

*長崎大学教養部

**長崎大学総合情報処理センター

***九州工業大学

ことを計画した。AppleTalk は、本来、マックどうし、あるいは Apple 社製のレーザプリンタをマックのネットワークに接続するためのプロトコルであった。これに対し、ネットワークプロトコルの defacto スタンダードは TCP/IP である。この TCP/IP によって世界中の主要なコンピュータセンターがネットワークされており、長崎大学も、学内 LAN プロジェクトの一環で、TCP/IP ネットワークの環境が整いつつある。

その TCP/IP ネットワークとの接続が実現すれば、孤立していた教養部 AppleTalk ネットワークは世界中のネットワークと交信できるようになり、教育・研究活動にとってさらに重要な役割を果たすことが十分に期待できる。

今回、われわれは旧式となった古い機材を再利用し、コスト的にも安く、AppleTalk と TCP/IP ネットワークの接続を実現した。本稿ではこの接続の方法について説明するとともに、新たに TCP/IP の世界とコネクションを持った教養部 AppleTalk ネットワークの使用感について報告する。

2 接続方法

マックを TCP/IP ネットワークにつなぐには、まず、イーサネットカード（約5万円）をネットワークするそれぞれのマックにインストールして、マックが直接 TCP/IP を理解できるようにする方法がある。この方法では、通信速度の向上は見込めるが、マックの台数分のイーサネットカードとトランシーバおよびトランシーバケーブル、それにイーサネットケーブルを引き回す経費が必要になってしまう。¹⁾

それとは別に、AppleTalk プロトコルと TCP/IP プロトコルの仲介をする機材（これをゲートウェイという）を一台設けて、AppleTalk ネットワークと TCP/IP ネットワークとを接続する方法もある。今回われわれはこちらの方法を採った。

このゲートウェイのはたらきについて簡単に説明する。

ゲートウェイは2つのポートを持ち、一方は AppleTalk に、もう一方は TCP/IP ネットワークにつながっている。

標準的なマックは TCP/IP プロトコルを理解できないが、MacTCP というソフトウェアをインストールすることによって、マックの TCP/IP オリエンテッドなアプリケーション（たとえば NCSA-telnet）は TCP/IP のパケットを埋め込んだ AppleTalk パケットを作り、そのパケットを LocalTalk ケーブル上に送り出すことが可能になる。ゲートウェイは受け取ったパケットを分解して TCP/IP パケットを取り出し、Ethernet ケーブル上に転送する。Ethernet につながった UNIX マシン²⁾ はその TCP/IP パケットを受け取ることができる。

逆に、UNIX 側からマックへ向けて送信された TCP/IP パケットは、Ethernet ケーブルを伝わってゲートウェイに達すると、ゲートウェイがそのパケットを AppleTalk で包んで、LocalTalk ケーブルに転送する。その AppleTalk パケットを MacTCP がつかまえて、AppleTalk の包みはずして TCP/IP パケットを取り出し、しかるべきアプリケーションに渡す。こうして通信が成

¹⁾ 現在ではネットワークの重要性がより強く認識されつつあり、イーサネットカード程度の支出はコンピュータがネットワークにつながることで十分にペイできると考えるユーザが増えてきている。

²⁾ UNIX マシンに限らず、TCP/IP を理解できるマシンなら何でもよい。

表1. 教養部内から TCP/IP ネットワークをアクセスするための方法と費用

方法	外部との接続	スピード	ネットワーク化のコスト	ソフトの対応
AppleTalk Only*	×	230kbps	—	—
無手順シリアル	△	9.6kbps	△	△
イーサネットカード	○	10,000kbps	△	○
ゲートウェイ専用機	○	230kbps 以下	○	○
Mach ^{Ten} でゲートウェイ	○	230kbps 以下	◎	○

*AppleTalkだけではTCP/IPネットワークをアクセスできないが比較のためにあげた。

立する。 3)

さて、長崎大学教養部内には AppleTalk のケーブルがすでに敷設されているし、学内 LAN の Ethernet ケーブルも届いているから、適切な場所にゲートウェイを設置さえすれば、AppleTalk ネットワークは TCP/IP ネットワークと通信できるようになる。

市販のゲートウェイ専用機としては FastPath や GatorBox などがよく知られている。それらの価格は安くなったとはいえ、1993年10月現在、40万円を越えている。

われわれが計画したのは、型遅れになった旧式のマック（マック SE, CPU は MC 68000, クロック 7 MHz, 4 MB の RAM, 20MB の SCSI ハードディスクを内蔵）に Mach^{Ten} というアプリケーション（Tenon Instruments 社製品、最新のバージョン2.1の価格は \$ 495）をインストールし、AppleTalk と TCP/IP とのゲートウェイとして復活させることであった。この方法であれば、新たに必要となる機材は、

1. マック SE にインストールすべきソフトウェア, Mach^{Ten}
2. マック SE にインストールすべきイーサネットカード
3. 学内 LAN にアタッチするイーサネットトランシーバ
4. イーサネットトランシーバからマック SE までのトランシーバケーブル
5. マック SE を AppleTalk ネットワークにつなぐためのコネクタおよびケーブル

であり、マックが直接 TCP/IP を理解するようにする方法やゲートウェイ専用機を購入する方法に比べて、格段に経済的である。

この方法により旧式のマシンをゲートウェイに仕立てた場合、満足な通信速度、信頼性が確保できるかどうかの問題であった。

3 Mach^{Ten}

Mach^{Ten} はマックで動作する Mach カーネルである。カーネルには UNIX4.3BSD が載っている。

最近、386BSD など、パーソナルコンピュータで動作する UNIX が登場しているが、その中で Mach^{Ten} の存在を際立たせているのが、Mach^{Ten} はマック OS 上のアプリケーションの一つとして実行されるという点であろう。

DOS マシンとしての386機上で動作していたアプリケーションは、386BSD のもとではまった

3) ゲートウェイのより詳しいはたらきに関しては、宮地 [4] などを参照されたい。

く動かないか、動かすのは難しい。386BSD を導入することは、それまで DOS 上に蓄えた資産を捨てて、まったく新しい環境へ移行することであると言えよう。

それに対して、*MachTen* は旧来のマックのアプリケーションと共存とも言うべき関わり方をする。*MachTen* の立ち上げは、マックの通常のアプリケーションを起動するのと同様、アイコン上のダブルクリックである。*MachTen* が立ち上がった後でも、Excel や Mathematica 等のマックのアプリケーションを *MachTen* から (つまりシェルのプロンプトからコマンド名をタイピングすることによって)、あるいはマック OS から (つまり、ファインダ上でアイコンをダブルクリックすることによって) 立ち上げることができる。

バッチ処理に向く仕事は従来マックではしにくいと言われていたが、そのような作業は UNIX のシェルスクリプトで実行し、その結果をマックのアプリケーションである Excel で受け取り、きれいにグラフ化する、というようなことを、*MachTen* の動く一台のマック上で実行することが可能である。

パソコンから UNIX に乗り換えようというユーザの懸念のひとつは、パソコンアプリケーションに比した UNIX アプリケーションの少なさであろう。しかし、*MachTen* の場合はそれまで使っていたマックのアプリケーションをそのまま使うことができる。もちろん、*MachTen* をインストールしたといっても、CPU が速くなるわけではないのでアプリケーションの処理速度が上がるわけではない。しかし、百数 MIPS もの処理能力を持つ他の UNIX マシンでサーバを動かす、クライアントを *MachTen* で実行させることができる。コンピュータネットワーク上では良質のソフトウェアがフリーで流通しているので、プログラムさえ手に入れることができれば、上記のことは一般にたやすい。

もちろん、マックのアプリケーションとして UNIX のアプリケーションと通信するソフトウェアをプログラムすれば同じことができるが、すでに UNIX 上で動作して実績のあるプログラムを *MachTen* 移植する方が楽である。*MachTen* の言語処理系は gcc および g++ である。⁴⁾

さらに、マックには使いこなされたウィンドウシステムがある。*MachTen* では新たにウィンドウシステムをインストールする必要がない。たとえば、*MachTen* では UNIX のシェルはマックのウィンドウ上で起動される。そのウィンドウの隣では Excel のウィンドウがオープンされたりする。図1は、UNIX の標準エディタである vi とマックの代表的なグラフ作成ソフトである Cricket Graph が同時にオープンされているマック SE の画面例を示したものである。

これに対して、386BSD などのパソコン UNIX でウィンドウ環境がほしい場合、ユーザは X 11 などのウィンドウシステムを追加インストールしなければならない。追加インストールにはハードディスクスペースが数十メガバイトも必要となってしまう。

以上、述べたように、安価である点、旧来のアプリケーションと共存できる点、使いこなされたウィンドウシステムが存在する点、などの理由から、*MachTen* はマックユーザになじむ、そして、インストールしたその日から使える UNIX である、と言っていいかもかもしれない。

⁴⁾ ここで述べている *MachTen* は最新版のバージョン 2.1 である。筆者たちが使っている *MachTen* のバージョンは 1.01 と古く、gcc 等、開発環境は付属していなかった。さらにつけ加えるならば、バージョン 1.01 では仮想記憶もインプリメントされていない。仮想記憶のない、実メモリ 4 MB のマシンで UNIX が動くとは驚きである。もちろんマルチタスクである。しかも、マック OS も同時に動いているのである...

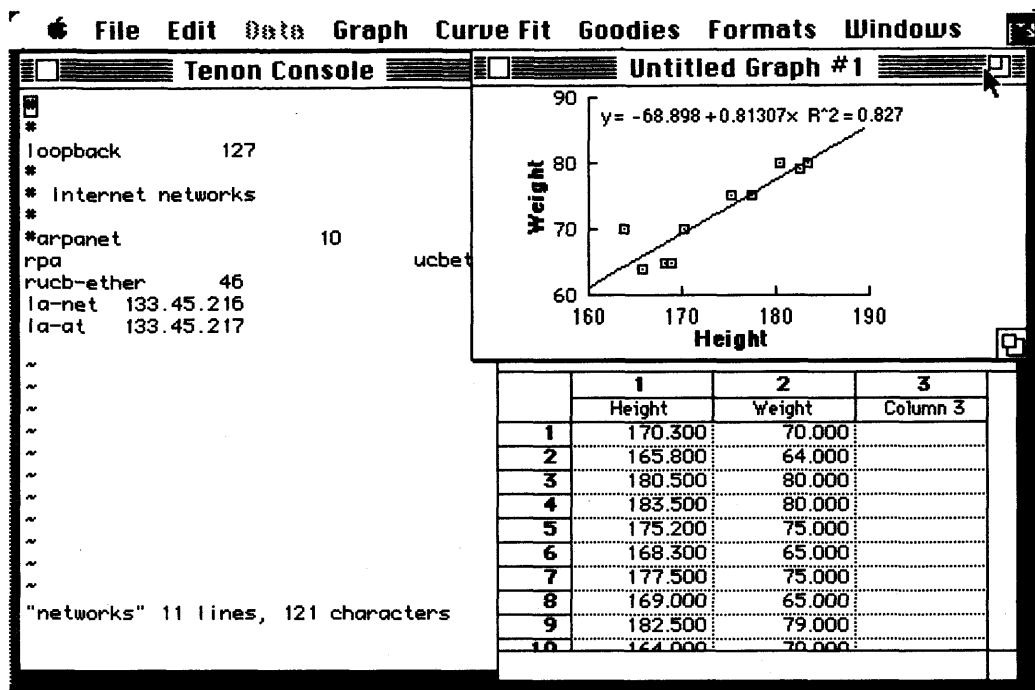


図1. MachTen の画面

4 学内 LAN

長崎大学の学内 LAN においては、B クラスの IP ネットワークアドレス 133.45 を 255.255.255.0 でマスクして（部局、建物等の単位で）サブネット化している。そして、サブネットアドレスの割り当てや、それらを相互に結ぶ“基幹 LAN”の運用は総合情報処理センターが行なっている。

今回の *MachTen* + マック SE による AppleTalk ネットワークの IP 接続は、IP 的に言えば通常のサブネットが1つ増えることにすぎず、各マックは、そのサブネット内のアドレスを持つ“IP ホスト”として見える。

AppleTalk サブネットが直接つながる教養部幹線 LAN は、10Mbps の通信速度を持つ 10 BASE5 イーサネットであり、教養部幹線がぶらさがる基幹 LAN は、100Mbps の通信速度を持つ FDDI である。図4参照。

5 経路制御

5.1 通常の動的経路制御

サブネット化したネットワークでは経路制御が必要になるが、長崎大学の学内 LAN では、基幹 LAN では原則的に RIP (Routing Information Protocol) を用いた動的制御を行なっており、部局内部のサブネット化にも動的制御を推奨している。

そこで今回もまず動的制御を行なうために、*MachTen* + マック SE 上で経路制御デーモンを動かそうとした。

この方法は、以下の仕組みで、AppleTalk サブネット上のマック (133.45.217.??) から/へ

の経路を制御する。なお、A,B,C 等の記号は、
図4参照。

- (1) A が B, C に対して、学内や学外への default 経路を RIP によって知らせる。よって、B や C は、(自分が直接知らない) すべての宛先の IP データは、A に投げつけることになる。
- (2) C が A, B に対して、133.45.217への経路を RIP によって知らせる。よって、A や B は、133.45.217.??宛の IP データは、C に投げつけることになる。
- (3) さらにAは、(2) のRIPで知った133.45.217の存在を学内の他のサブネットに対して(再び RIP によって)知らせる。よって、学内の他のサブネットからも、最終的には A 及び C を経由して、133.45.217.?? へ IP データを送ることができる。

ところが、

- a) まず、最も簡単な方法として、C 上で UNIX 標準の経路制御デーモンである routed を動かしてみた。(具体的には、*MachTen*:/etc/rc でコメントアウトされている routed の起動の行を生かすだけである)

ところが、C の routed が送出する RIP パケットが不正なフォーマットをしていて、AやBで正しく処理できなかった。

- b) そこで、routed より高機能な無償ソフトウェアである、gated を C 上で動かすことを考えた。しかし、*MachTen* のバージョンが1.01と古く、gated をメークすべき C コンパイラが付属していない(バージョン2.0以降は gcc が使えるのだが)。
- c) また、マック SE のメモリやパワーを考えると、動的制御はやめようということになった。

5.2 静的経路制御

次に考えたのが、AとC上にそれぞれ静的に(相手への)経路を設定する方法である。

この方法は、以下の仕組みで、133.45.217.??から/への経路を制御する。

- (1) C は、学内/学外への default 経路(中継先=A)を静的に知っている。よって、C は、(自分が直接知らない) すべての宛先の IP データは、A に投げつけることになる。
- (2) A は、133.45.217への経路(中継先=C)を静的に知っている。A は、133.45.217.??宛の IP データは、C に投げつけることになる。
- (3) さらに A は、(2)で静的に知っている133.45.217の存在を学内の他のサブネットに対して(今度は RIP によって)知らせる。よって、学内の他のサブネットからも、最終的にはA及びCを経由して、133.45.217.??へIPデータを送ることができる。
- (4) 例えば B のようなサブネット133.45.216内のホストから133.45.217.??へ IP データを送

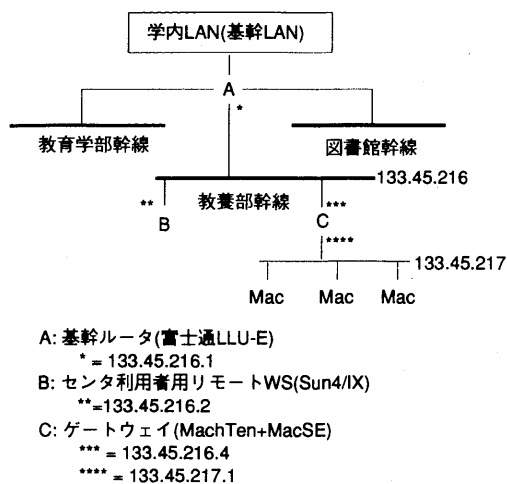


図2. 学内 LAN との接続関係

```
% traceroute 133.45.217.2
traceroute to 133.45.217.2 (133.45.217.2), 30 hops max, 40 byte packets
 1 lady (133.45.8.15)  2 ms  4 ms  4 ms
 2 localhost (127.0.0.1)  6 ms  5 ms  5 ms
 3 localhost (127.0.0.1) 12 ms  7 ms  8 ms
 4 c-la-01 (133.45.216.2) 10 ms  9 ms  9 ms
 5 133.45.216.4 (133.45.216.4) 158 ms  51 ms  51 ms
 6 * * *
```

図 3. traceroute

```
traceoptions internal external route icmp ;
tracefile "/var/adm/gated.log" replace;
interface all passive ;          # don't time out my interfaces
rip supplier {
    trustedgateways 133.45.216.1 ; # llu-e
};
# need some static routes
static {
    133.45.217.0 gateway 133.45.216.4 ;
};
propagate proto rip interface
    133.45.216.2
{
    proto direct {
        noannounce all ;
    };
    proto static {
        announce 133.45.217.0 metric 3 ; # path to 217
    };
    proto rip {
        noannounce all ;
    };
};
```

図 4. gated.conf

る場合は、最初は、default 経路に従って、A に送る。しかし、A は、C に送ればよいことを知っているの、そのデータを C へ転送すると共に、B に対しては、ICMP Redirect 通知を送る。これにより、次回からは、B は C へ直接送るようになる。

ところが、また問題が発覚した。富士通社製のルータ A が、(3)の機能を待たず、静的に設定した経路情報は、RIP で他に伝えることができなかった（ちなみに、gated はこの機能を持っている）。

そこで、根本的解決ではないが、どこまで使えるかの興味もあり、次に示すように、B を偽のルータに仕立てた方法（仮に、proxy ルータと呼ぶ）を試してみた。

5.3 proxy ルータを用いた経路制御

ここで行なったのは以下のような方法である。

- (1) C は、学内/学外への default 経路（中継先=A）を静的に知っている。よって、C は、（自分が直接知らない）すべての宛先の IP データは、A に投げつけことになる。

具体的には、C の /etc/rc の中で、

```
route add default 133.45.216.1 2
```

とやればよい。

- (2) B(Sun4/IX) 上で gated を動かし、133.45.217への経路（中継先=C）を静的に設定する。gated.conf の設定は、図4参照。これにより、B は、133.45.217.??宛の IP データは、C に投げつけることになる。
- (3) ここで、B は、(2) で静的に知っている133.45.217の存在を A に対して（今度は RIP によって）知らせ、A はそれを学内へ知らせる。
- (4) A は、133.45.217.??宛の IP データは、B に投げつけることになる。よって、学内の他のサブネットからも、最終的には A, B 及び C を経由して、133.45.217.??へ IP データを送ることができる。

図3は、133.45.8.2のマシンから133.45.217.2宛のデータがどこを経由しているかを調べた様子である。3がA,4がB,5がCにあたる。

なお、B, C での（IP ルータとして）の IP データ転送（forwarding）に関して、

B は、ネットワークインタフェースを1つしか持たないので、（通常の UNIX カーネルの設定では）IP データの forwarding をしない。カーネルの再コンフィグ、または adb によるパッチも可能であるが、今回は、ダミーの slattach(SLIP 接続用デーモン)を、ttya で上げて、ネットワークインタフェースを2つにし、IP データの forwarding 機能が有効になるようにした。

C は、ネットワークインタフェースを2つ持つわけだが、/etc/rc の中で、ipforwarding on というコマンドを発行して、明示的に forwarding 機能を有効にする必要があった。

5.4 AppleTalk 上での IP ホストアドレス

AppleTalk 上の各マックには、133.45.217.??という IP ホストアドレスが振られるが、マックの TCP/IP ドライバである MacTCP の機能を用いると、それをダイナミックにおこなうことができる。マックはブート時に自分のつながっているネットワークのアドレスと、そのネット

ワークにつながっていて、かつ、自分よりも先にブートしたマシンのアドレスをチェックし、その時点で使用されていないアドレスを探して、それを自分のアドレスとすることができる。

これを利用すれば、マシンのアドレスを管理する必要がなく、ネットワーク管理者の負担は減る。しかし、TCP/IP 側からはアドレスによってマシンを特定できなくなるため、Eudora(後述)や、一部の anonymous ftp が利用できなくなるなどの弊害を避けられない。そのため、今回はダイナミックにアドレスを割り振ることをやめ、マシンにスタティックなアドレスをマニュアルで割り振ることにした。

6 ネットワークアプリケーション

Mach^{Ten} をゲートウェイとして用いることで LocalTalk 接続されているマック上でも MacTCP のアプリケーションソフトが利用できる。

それらのソフト(クライアント)から利用する TCP/IP の世界の代表的なネットワーク機能(サービス)として、telnet, ftp, 電子メール, 電子ニュース, DNS ネームサーバ等がある。

一方、*Mach^{Ten}* 自体は UNIX4.3BSD コンパチであり、電子メール, 電子ニュース, DNS ネームサーバのサーバを行なうことも原理的には可能である。しかし、マシンパワー的にも管理体制的にも無理があるので、この3つに関しては、センターのサーバをアクセスするという方法を取る。

1. 電子メール

電子メールは、センターに利用者登録し、センターのメールサーバを、Eudoraでアクセスして利用する。Eudora は POP(Post Office Protocol) 手順を使ってリモートのメールボックスからメールを読みだし、SMTP(Simple Mail Transfer Protocol) 手順を使ってリモートのメールサーバに送信すべきメールを渡すような、フロントエンドである。

なお、利用者の電子メールアドレスは、以下のようになる。

```
xxxx@cc.nagasaki-u.ac.jp
```

2. 電子ニュース

電子ニュースは、センターのニュースサーバを、NewsAgent 等でアクセスして利用する。NewsAgent は NNTP(Network News Transfer Protocol)手順を使ってリモートのニュースサーバを利用するフロントエンドである。

この時、なお、ニュースを投稿する時、自分の電子メールアドレスが、上で述べたものになるように、設定に注意する。

3. DNS ネームサーバの参照

マック上のソフトから DNS を参照する場合、センターのネームサーバをアクセスする。

4. DNS ネームサーバへの登録

マックを、正式な IP ホストとして DNS に登録するためには、センター(のネームサーバ)に対して登録申請を行なう必要がある。センタのネームサーバに登録されると、マックのアドレスは全世界に公開されることになる。

現在のセンターの方針として、ローカルにメールシステムを運用できない(パソコン等の)ホストは、申請があれば逆引き DNS には登録するが、正引き DNS には登録しないことにな

っている。

この方針を適用して、教養部 AppleTalk ネットワークにつながったマックは、現在、正引き DNS には登録されていない。

7 ゲートウェイとしての *Mach^{Ten}* の評価

7.1 システムの安定性

Mach^{Ten} をインストールしたマック SE を AppleTalk と TCP/IP とのゲートウェイとして利用を開始してすでに5カ月が経過した。現在（1993年10月）のところ AppleTalk ゾーンに接続されたマックの内4台のマックから、telnet, Eudora, Fetch などのアプリケーション利用がなされているが、サービス開始初期のころ各種パラメータ設定ミスによって接続が思うように出来なかった例はあったもののこれまでに何の障害も発生していない。

また、ゲートウェイとして利用しているマック SE は最初設定したままで何の変更もなく正常に動作している。これまでに要した保守作業といえば工事による予告停電に備えて事前にシャットダウン、工事終了後に再起動を2回ほど行ったのみであり、本システムの安定性に関してはまったく問題ないと言えよう。

7.2 アプリケーションの利用

1. 電子メール・電子ニュース

デジタルフォン経由の場合、センターのワークステーションにログインした後、mail コマンド、gn コマンドあるいは emacs の RMAIL, GNUS などによって電子メールや電子ニュースを読んだり、送ったり、投稿したりするのにそれぞれのコマンドについてある程度の知識が必要であった（ワークステーションはヘルプ機能やオンラインマニュアルが充実しているが、快適に利用するにはいくつかのコマンドの利用法を覚える必要がある）。

一方、先に紹介した Eudora, NewsAgent などはマックのアプリケーションであることから特にその操作性に関してはワープロソフトなど他のマック用のアプリケーションの利用経験があればなんの問題もなく使える。この点がデータの転送速度の比較と同じくシリアル回線接続と比較してもっとも大きな違いであると言えよう。それぞれのソフトの各種パラメータ設定も各マシンのアドレス、個人のデータ（センターの登録番号、パスワードなど）が異なるだけなので初心者でも比較的容易に行える。

2. Fetch

Fetch は、センターのホストマシンや他のマシンとのファイルの送受信を行うソフトである。シリアル回線接続では xmodem や kermit などのファイル転送プロトコルによる細かいパラメータやファイル名の設定を必要としたファイル転送が、Fetch を使うことでその操作の全てがマウスでのボタン操作だけで可能である。さらに転送速度も比較にならないほど速い（今回のファイル転送速度の比較を行う際にもこのアプリケーションを利用した）。さらに、シリアル回線接続とのもっとも大きな違いはセンター外のホストからファイルを転送するのに、直接自分のマシンにダウンロード出来ることである。

7.3 ファイル転送速度

デジタルフォン経由のシリアル接続とゲートウェイ利用（図5参照）とのファイル転送速度の比較を行った。シリアル接続ではターミナルソフトとして `rz`, `xymodem` のファイル転送機能を追加した `NinjaTerm0.962PB`, 回線速度は9600bps, センターホスト側のファイル転送手順は `zmodem` 手順を利用した。ゲートウェイ接続では、先に紹介した `Fetch` を用いてファイ

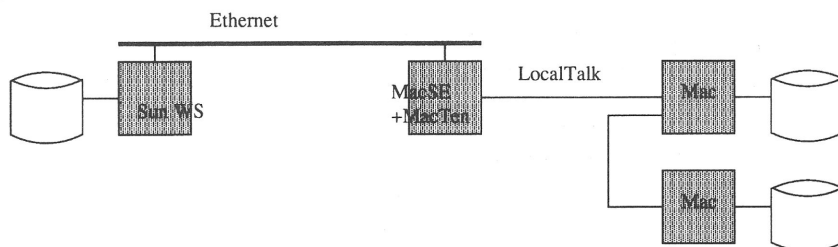


図5. ゲートウェイ利用のファイル転送経路

ル転送を行った。転送時間の測定は、ソフトによって転送時間表示の有無があるのでストップウォッチによって測定した。その場合、最終的な転送のためのパラメータ設定やソフトからの入力促進に対する最終入力と同時に計測を開始して、転送終了の警告音あるいはコマンド入力可能な状態になった時点を終了とした。計測は1秒以下は切捨てた。また、ゲートウェイ接続では他の3台のマシンをシャットダウンした状態（ゲートウェイマシンとテストに使ったマック SE/30のみが AppleTalk ゾーンで起動されている状態：Get, Put）と他の一台で同じく `Fetch` を利用してファイル転送を行っている状態（Loaded-get:ダウンロードのみ）の二通りの条件で測定した。さらに AppleTalk ゾーンに接続された2台のマック間（Mac-Mac）のファイル転送時間も参考のため測定した。測定に用いたサンプルデータはホストマシンにおいて `mkfile` コマンドで100kb, 200kb, 500kb, 1mb の4つのファイルを作成し、ホストからダウンロードしたファイルを再度アップロードする方法を取った。

測定の結果は表2に示すとおりである。

まず、全体的にデジタルフォン経由のシリアル接続（S-get, S-put）ではその転送時間がどの大きさのファイルにおいてもほぼ同じで安定しているが、ゲートウェイ接続ではかなり変動した。特にゲートウェイ接続でのダウンロード（Get）時のみテストした1mbのファイル転送では5回の測定のうち最も遅い場合（116秒）は速い場合（74秒）の約1.5倍もの時間を要した。デジタルフォン経由のシリアル接続では、電話回線を専有しているので比較的安定した転送速度が得られるが、ネットワーク利用では同じ回線上をいくつもの異なるデータが流れることになるのでその状況（トラフィック量）によって転送速度が左右されることになる（電車はほぼ時刻表通りに運行されるが、高速バスなどの道路を利用する交通機関はその運行時間が混雑の程度に左右されることと同じである。）

ここでゲートウェイ接続でのダウンロード（Get）に要する時間、即ちホストマシンとマックとのファイル転送時間が、2台のマック間（Mac-Mac）の転送時間（この場合は、AppleTalk上に2台のマックの IP アドレスが流れているのでゲートウェイマシンを経由しないでデー

表2. ファイル転送時間(秒)の比較

	Get	S-get	Loaded-get	Mac ← Mac	Put	S-put
100kb	11	113	20	9	33	160
	11	112	19	9	48	162
	8	112	18	9	38	165
	9		13	9	56	
	8		12	9	41	
Mean	9.4	112.3	16.4	9	43.2	162.3
200kb	15	222	39	15	40	324
	21	222	35	23	70	328
	16	223	30	22	46	325
	19		32	23	40	
	17		49	21	64	
Mean	17.6	222.3	37.0	20.8	52.0	325.7
500kb	40	554	82	45	225	816
	40	554	116	52	90	820
	39	554	89	57	180	812
	41		83	43	180	
	38		100	53	250	
Mean	39.6	554.0	94.0	50.0	185.0	816.0
1mb	77					
	113					
	116					
	115					
	74					
Mean	99.0					

タの転送ができる) とほぼ同等, あるいは短かった. このことからゲートウェイとして利用した *Mach^{Ten}* が十分機能していると言えよう.

図6は各転送方法毎に100kbのファイル転送時間を基準(1に設定)としたときの200kb, 500

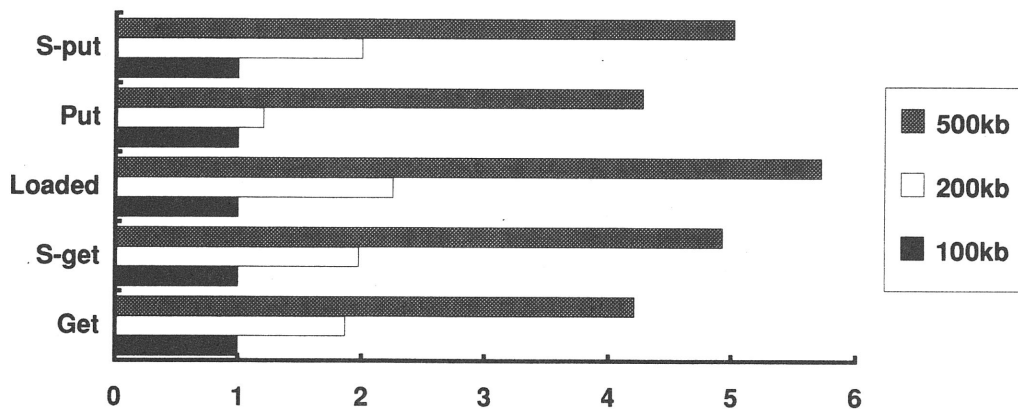


図6. ファイル転送方法別の転速度比較

kb のファイル転送に要した値を示したものである。デジタルフォン経由のシリアル接続ではファイルのサイズにほぼ比例して転送時間が増加している。一方、ゲートウェイ接続の場合はファイルサイズが大きくなるとその転送効率がやや向上したが、ネットワークに負荷がかかった状態 (Loaded-get) ではシリアルの場合よりもその値が大きくなった。

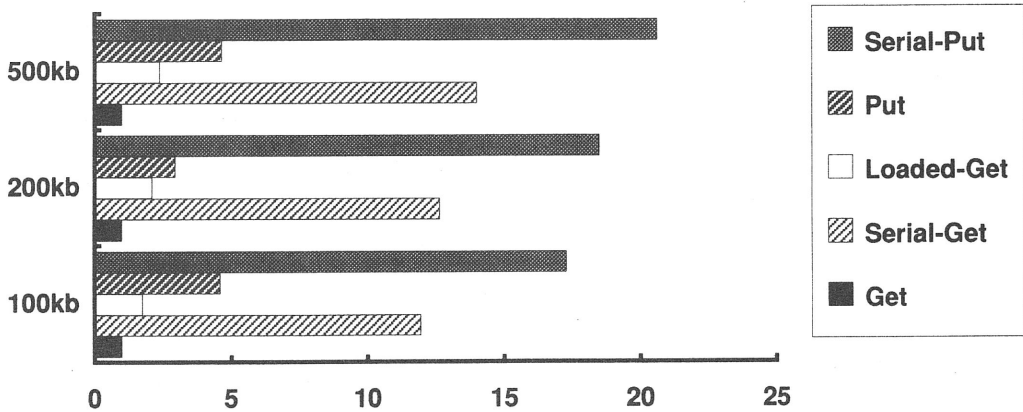


図7. ファイルサイズ別の転送速度比較

図7は、ファイルサイズ毎にゲートウェイ接続におけるファイル転送時間（ダウンロード：Get）を基準（1に設定）としたときの各転送方法によるファイル転送に要した値を示したものである。ダウンロードの場合、シリアル接続では12-14倍、負荷状態では1.7-2.3倍程度の時間を要した。アップロードではダウンロードと比較して3-4.6倍程度、シリアル接続では17-20倍程度の時間を要した。アップロードがダウンロードと比較して3-4.6倍も時間を要したのは *MachTen* や *MacTCP* と関係していると考えられる。

接続形態によるハード上の差異が存在するので直接的な比較にはならないが、以上の測定結果よりゲートウェイ接続でのファイル転送はデジタルフォン経由のシリアル接続と比較してダウンロード時で10倍強、アップロード時で3.7-6.3倍程度の転送速度が得られ、ネットワーク上に負荷がかかるとその転送速度は約半分程度になるもの実際の利用においては、比較にならないほど十分に速い転送速度と考えられる。*AppleTalk* は230kbps でデジタルフォンを利用した場合のシリアル接続では最高9.6kbps であるので単純計算では約24倍になるはずであるが、先に述べたようにネットワーク上のトラフィックの状態、あるいはゲートウェイによるプロトコル変換に要する時間などが存在するので実際には妥当な速度であると考えられる。

8 まとめ

TCP/IP ネットワークに *AppleTalk* ネットワークを接続することを計画した。マック SE に *MachTen* というアプリケーションをインストールし、マック SE を *AppleTalk* と TCP/IP とのゲートウェイとして利用した接続を試みた。その結果、ゲートウェイの安定性、電子メール、電子ニュース、ファイル転送などデジタルフォン経由のシリアル接続と比較してその操作性、性能において格段に優れその有用性を確認できた。さらに、この試みを実行するにあたっては、筆者の一人が教養部に設置されていた端末のある部屋にマック SE を持ち込み、端末から他の2

人の指示を talk や mail によって受けながら作業を進め、筆者 3 人が一度も直接介することなく行われた。本論をまとめるにあたっては電子メール、ファイル転送をフルに活用したことはない。このことは、最初にも述べたように遠隔地の人的資源と通信ながら作業を効率的に進めことのできるコンピュータネットワークの威力を証明する一例であるとも考えられる。

謝 辞

本試行を行うに当り、長崎大学工学部電気情報工学科情報システム学、中村千秋先生には Ethernet への接続をはじめ多大なるご援助、ご指導をいただいた。深く、感謝の意を表します。

参考文献

- [1] Comer, D. (邦訳:村井純, 楠本博之): Internetworking With TCP/IP Vol.I (TCP/IP によるネットワーク構築), Prentice Hall (共立出版), 1991.
- [2] Fedor M. and J. C. Honig: *gated.man/gated-config.man*, 1991.
- [3] 花田英輔, 鶴正人, 野崎剛一: 長崎大学全学 LAN の構築と運用, 長崎大学総合情報処理センター, センターレポート, 12, 33-38, 1993.
- [4] 宮地力: *Macintosh* インターネット入門. アジソン・ウェスレイ, 東京, 1993.
- [5] 大喜雅文: 長崎大学歯学部附属病院の LAN について, 長崎大学総合情報処理センター, センターレポート, 12, 51-61, 1993.
- [6] 鶴正人: TCP/IP ネットワーク長崎大学総合情報処理センター, センターレポート, 10, 73-115, 1991.
- [7] Tenon Intersystems : *Mach^{Ten}User's, Guide*, 1991.