

## 第 5 部

### パソコン通信との相互接続実験



# 第 1 章

## パソコン通信との相互接続実験

### 1.1 はじめに

発達したインターネット環境においての重要な課題のひとつに、グローバルな接続性をあげることができる。コンピュータとコミュニケーションによる環境の典型的な事例であるインターネットは、研究・教育組織や関連企業などのネットワークシステムが相互に接続され、全体としてグローバルなコミュニケーション環境を形成している。

インターネット環境の接続性の広がり第一段階は電子メールによる相互接続であり、インターネットで利用されている RFC822 形式 [9] を中心に、MHS 形式などの他の一般的な形式との変換機能を開始して、120カ国以上の接続性を実現している。このうち約 50カ国では IP によるより高度な相互接続をおこなっていて、ここでは資源や知識の共有を実現し、共同作業を可能にする巨大な分散システムを形成している。

電子メールの接続性の拡大を考えた場合に、一般に「パソコン通信」と呼ばれている集中型センター形式の電子掲示板、情報提供サービスを考える必要がある。パソコン通信は、1987年ごろから商業サービスが開始され、パーソナルコンピュータ、ワードプロセッサを端末として、モデム接続でアクセスするという形態で、インターネットとは別のコミュニティを形成している。パケット交換網を用いたアクセスポイントを用意し、地域格差をなくし、従来型のオンラインデータベースを統合し、国内に数十万人の利用者がいると言われている。

商用電子メールとインターネットの接続事例は米国の MCI mail などの接続を行った [10] や、CompuServe の例がある。しかし、応用技術、接続技術や利用形態、社会背景などわが国の独自性があり、この分野の実験と検討は急務である。

このような状況を背景として、WIDE インターネット と商業パソコン通信サービスとの実験接続を実現し、拡張するネットワーク相互接続の検討を行なった。実験は、接続技術、応用機能変換技術、管理運用ポリシーの整合、社会科学の各側面からの評価を目的として行なった。

インターネットとパソコン通信の接続の意味としては、双方のユーザ間での情報交換、双方のサービスの交換の 2つを考えることができる。

ここで、WIDE Project と各パソコン通信を接続する実験の第一段階に、上記の 2つにそれぞれ対応するものとして、以下の実現を検討した：

- インターネット とパソコン通信間でのメール交換
- インターネット からのパソコン通信の利用

実験は検討と評価を行ないながら段階を分けた。各段階では実験対象ネットワークの範囲を拡大している。

第一段階の実験は、技術的検討と、WIDE Project, 慶應義塾大学湘南藤沢キャンパスにおける試験運用を行い、第二段階で WIDE インターネット における技術的、運用的検討をし、最終段階で、その他のネットワークへの拡大の検討を行うこととして実験を進めた。

実験は、NIFTY-Serve, PC-VAN, ASCII-net の 3 つのサービスとの間での相互接続を計画したが、本報告ではまず、前半で現在までに行なった、NIFTY-Serve, PC-VAN の 2 者について、本実験で明らかになった問題点の報告と、利用状況および、利用者アンケートに関する結果について述べ、後半でゲートウェイの実装に関しての詳細を述べる。

## 1.2 実験の経過

日付	実験範囲	参加ドメイン
1992 年 9 月 1 日	sfc.keio.ac.jp, wide.ad.jp, nic.ad.jp, junet.ad.jp	4
1992 年 10 月 12 日	WIDE Internet	80
1992 年 12 月 21 日	TISN, HEPNET-J	117
1993 年 2 月 4 日	TRAIN, KARRN, JAIN の一部	160
1993 年 5 月 (予定)	jp 全域	832

表 1.1: 相互接続実験の経過

実験は対象範囲を順次広げるという方針で、その都度利用統計、システム負荷などを監視しながら、問題点を確認しながら行った。表 1.1 にスケジュールを示した。

## 1.3 相互接続の設計

具体的な電子メールの相互接続は次の各項目の方式を定め実現した。

- アドレスの変換
- 配送方式の確立
- 運用・管理方針の整合

### 1.3.1 アドレスの変換

パソコン通信の利用者 ID は、ひとつのアドレス空間で一意的なアドレスを提供している。一方、インターネットでは階層的なアドレス空間によって定義される [11]。わが国では、第一レベルに国識別子、第二レベルに組織の種別、第三レベルに組織名を割り当てている [12]。

そこで、本実験では各パソコン通信に対してそれぞれ一つの第三レベルドメインを割り当て、既存の利用者 ID と組み合わせて整合性をはかった。

### 1.3.2 配送方式の確立

全く異質のシステム間でメール交換を行う場合、配送メカニズムの変換、メッセージ形式の変換を行う必要が生じる。この実現のための機構として、インターネットにおけるメール配送システムを受け持ち、パソコン通信側との間で配送を行うゲートウェイと、パソコン通信側とのメッセージ交換、配送メカニズムを実現するゲートウェイを結びこれを行った。

### 1.3.3 運用・管理方針の整合

インターネットと商用サービスとして提供されているパソコン通信の間にはポリシー、運用方針に相違がある。一方、集中システムと、インターネットではメール配送システムの違いによる信頼性に対する考え方、認証システム、セキュリティに関しても明らかな差異が存在している。これを互いに尊重し、かつ信頼性ある通信を実現する必要がある。

一般にインターネットのメールシステムでは受信側で、発信者のリストによってスクリーニングする機能はないが、今回はこれを組み込むことだ。双方のポリシーが尊重されるように互いの側のゲートウェイを運用している。パソコン通信がインターネットに直接接続するのではなく、一カ所のインターネット側のゲートウェイを必ず通過するようになっている。

## 1.4 相互接続の実現

### 1.4.1 アクセス制御

ゲートウェイである `pcgw.wide.ad.jp` では、`sendmail` を改造して、アクセス制限を行なった。`sendmail` の改造は、WIDE 版 `sendmail 5.67+1.6W`, `sendmail.cf-2.8b` を基に行なった。`sendmail` に今回加えた改造は、From による制限の導入と IP アドレスによる経路制御機構である。これに、`sendmail.cf` での制御を合わせて実現している。

この詳細に関しては、本章後半で述べる。

## 1.4.2 配送方式

接続は、WNOC 藤沢の SparcStation 1+ (pcgw.wide.ad.jp) から、NIFTY-Serve, PC-VAN 側のゲートウェイまで、9.6Kbps での接続を行なっている (図 1.1)。NIFTY-Serve は、SunLink INR により、NIFTY-Serve センターの SparcStation 2 と接続、PC-VAN 側は SLIP で PC-VAN 側の EWS4800 と接続している。両方とも pointpoint 接続で routing はインターネット側にアナウンスしない。

パソコン通信側の独自のメールシステムと、インターネットアドレス、RFC822 に沿ったのメッセージ形式への変換は各パソコン通信サービスセンター側ゲートウェイで行なわれることとした。

図 1.1 に示したように、1 台のゲートウェイとなるマシン (pcgw.wide.ad.jp で NIFTY-Serve, PC-VAN のゲートウェイを行う。DNS 的には、niftyserve.or.jp, pcvan.or.jp に対する MX が、pcgw.wide.ad.jp に設定されている。また、NIFTY-Serve, PC-VAN からインターネットに送られるメールは一旦全部 pcgw.wide.ad.jp に送られ、ここからインターネットに対して配送される。

これにより、pcgw.wide.ad.jp の sendmail でのアクセス制御を行い、同時にメールの利用の統計情報の取得を行った。アクセス制御は送信者、受信者のアドレスによる制限と、PC-VAN と NIFTY-Serve の間を中継することを禁止する From/To の組み合わせの特定の場合の制限である。

## 1.5 相互接続の評価

### 1.5.1 アクセス制御とメールのルーティングの問題点

sendmail には From/To の両方の組み合わせによるのチェック機構はない。これを一般的に実装することは、今回行わなかった。PC-VAN から NIFTY-Serve へ (またはその逆) を禁止することは、checkcompat()(conf.c) を用いて行ったが、これはソースに直接アドレスが記述してあり、一般性はない。このチェックによって制限では、メーリングリストの配送に問題が生じることがある。すなわち、NIFTY-Serve のユーザがインターネット上のメーリングリストに発信した時、そのリストに PC-VAN のユーザが含まれるとき、その PC-VAN ユーザだけに送られないでエラーになるということが起こる。これを回避する手段はあるが、これを中継するゲートウェイ行うことは望ましくないと考えられる。

また、現在の DNS によるメールの配送メカニズムでは、メールだけを通常の IP の routing と異なる経路を指定することは、個別に sendmail.cf で明示的ゲートウェイ指定をする以外の手段はない。電子メールの受信者側で発信者ごとに、到達経路を個別に指示することはできない。途中で特定の相手との通信を拒む経路が存在する場合にはこれを発信者ごとに迂回することが困難であり、現状ではやむなく通信を断念するしかない。この問題は、いわゆるポリシーベースルーティングの問題の一種である。

### 1.5.2 メールヘッダと日本語の問題

パソコン通信側ゲートウェイでは、メッセージ形式の変換、インターネット側で使われる文字コード [13] との変換が行われている。当初、メールボディにパソコン通信側の情報が含まれていたが (図 1.2)、これは自動応答をするメールサーバでエラーになるなどの問題があったので (たとえば mail-server@nic.ad.jp など)、修正した。

現在の商用サービスとして提供されているパソコン通信では、ユーザの情報として、漢字以外の情報を持っていないことが多い。またメールの題名に漢字を使うのが一般的である。

これに対し、インターネットメッセージの標準形式である RFC822 では管理情報はヘッダ部としてまとめ、この部分は ASCII Text とされている。一部のユーザメールプログラムでは、ヘッダ部に日本語が含まれた場合、ESC (0x1B) を取るなどで扱えないことが起こる。

今回、実験的に図 1.3 に示すように、パソコン通信側メールシステムの情報をヘッダとして取り込んだ。Subject に関しては、利用者に使わないように指導することで解消できるが、ユーザの情報を何らかの形で伝搬することも望まれる。何らかの対処をして、ヘッダに日本語を取り込めるようにすることを検討する必要がある。これに関して、RFC1342[14] の採用など、今後検討していく予定である。

### 1.5.3 パソコン通信でのインターネットメールの実際

#### PC-VAN からインターネットへ

1. 電子メールのメニューから、宛先コードに MHSX400 を指定する。
2. 文書名 (RFC 822 の Subject) を一行入力する
3. TO:INET#shin(a)wide.ad.jp という形式でアドレスを入力する
4. 一行空ける
5. 本文を入力する

PC-VAN からは、インターネットへのメールの仲介として、X.400 システムを使用している。インターネットのアドレスは、@ に代えて、(a) とする必要がある。

インターネットから受信したメールは、PC-VAN 内でのメールと同等になるが、MHSX400 という ID からきたものとして見える。

#### NIFTY-Serve からインターネットへ

1. GO WPNC または GO INTERNET
2. サービスに入る (GO WPNCSEND でここまでスキップできる)

### 3. 本文、宛先 (インターネットアドレスをそのまま) を入力する

インターネットから受信したメールは NIFTY-Serve 内のメールと同等に扱うことができるようになるが、そのメールの発信者は ID WID00100 という利用者から発信されたように見える。インターネットから受信したメールに返信する場合には、アドレスを指定する必要はないように工夫されているが、RFC822 のヘッダのうち Reply-To: の取り扱いなどができないなど、一部不都合がある。

一通のメールが 300 行まででこれを越える場合、複数のメールに分割される。しかし、一般の場合メールボックスが 25 通まで (50 通に拡張可能) であり、これがあふれて受け取ることができないとエラーになる。メールの行数の問題は、1993 年 4 月 1 日より、NIFTY-Serve のメールボックスは 1 通 3000 行に拡大されたため、ほとんどなくなったと思われる。

#### 1.5.4 今後の課題

パソコン通信が、他システムとの接続を考慮して設計されていないのでやむを得ないことは多いが、ユーザにとっては、通常のメールインターフェースから送信受信でき、アドレスの指定が簡単であるほうが望ましい。RFC822 のヘッダの機能のサポートはインターネットユーザとの整合性の点で急務であろう。また、SMTP の拡張として提案されている RFC1427[15] などはシステムの制限によるトラブルを未然に防ぐ意義で検討が必要である。

## 1.6 利用状況

利用状況は、9 月 1 日から、10 月 11 日までの間は、sfc.keio.ac.jp, wide.ad.jp, nic.ad.jp, junet.ad.jp の 4 ドメインのみであり、利用はわずかであったが、10 月 12 日以降 WIDE Project 参加組織全体に広げた後は非常に多くの利用がされている (図 1.5, 1.6)。

12 月 21 日以降、TISN, HEPNET-J も参加している。図 1.7, 1.8 はそれぞれインターネット側からパソコン通信へ送信された数、パソコン通信側からインターネット側へ送信された数を 1 週間ごとにグラフにしたものである。これは pcgw.wide.ad.jp の syslog の解析によって得た数値であるが、メーリングリストなどの、同報メールは受信アドレスごとに数えてある。利用は、土日には激減しており、平日には 3000 ないし 4000 の利用者がいる。インターネット側からのメールは次第に明らかにメーリングリストと思われるメールが増加している。アンケートには現れなかったが、インターネット側から発信されるメールが、パソコン通信側から発信されるメールの 2 倍程度あることからメーリングリストの利用がうかがわれる。

メールのサイズについての統計は、やはりインターネット側から、パソコン通信側へのメールの方が大きい傾向にある。ユーザインターフェースの影響があると思われるが、ftpmail などの極端に大きなメールの利用は目だっていないようである。



	第一回		第二回	
	パソコン通信	WIDE Internet	パソコン通信	Internet
回答総数	83	112	114	134
1. 研究連絡	22(27%)	24 (21%)	34 (30 %)	33 (25 %)
2. 研究用データの送付	5 (6%)	5 (4%)	7 (6 %)	14 (10 %)
3. 私信	55 (66%)	71 (63%)	100 (88 %)	100 (75 %)
4. テスト	51 (61%)	93 (83%)	58 (51 %)	65 (49 %)
5. メイリングリスト	19 (23%)	21 (19%)	15 (13 %)	34 (25 %)
6. その他	5 (6%)	4 (4%)	10 (9 %)	10 (7 %)

表 1.2: アンケートの質問 1 の回答。カッコ内は回答総数に対する割合

## 1.7 利用実態調査

9月1日の接続開始直後、東大での調査で、パソコン通信を利用して、以下の学会のアクティビティが行なわれていることがあることが分かった。日本機械学会、日本分子生物学会、森林利用研究会（平成5年に学会発足予定）、精密工学会、日本発達心理学会、日本化学会。この他に、科研費重点領域研究（戦後日本形成の研究）の研究グループがNIFTY-ServeにCUG（Closed Users Group）を持ち、研究活動を行なっている。

利用実態を把握するために、インターネット、NIFTY-Serve、PC-VANのユーザ各200人（第二回は各300人）に対して、10月26日に第一回目のアンケートを、1月16日に第二回目のアンケートを行った。内容は第一回目は、利用目的（1. 研究連絡 2. 研究用データの送付 3. 私信 4. テスト 5. メーリングリスト 6. その他）と、実験に対する意見を、第二回目はこれに加えて、文字コードに関するトラブルはなかったかということと、課金に関する意見を聞いた（図 1.9, 1.10）。

第一回では、範囲を拡大することを望むという意見が多かった。大学で電子メールを使っていたのが、卒業後使えなくなったという人にとって、今回の実験によって電子メールでのコミュニケーションが再開できるようになったケースは目立ったものである。研究連絡の手段として、かなり多く使われており（20～30%）確かに、研究コミュニティの拡大に貢献しているということがわかる。

第二回のアンケートでは、やはりインターネット側のユーザから、Subjectに日本語が使われている時に、取り扱いできないというものと、パソコン通信側からはIDだけなので、誰から来たか分からないという意見が数件ずつあった。やはり、この点が今後の課題であると思われる。

## 1.8 おわりに

今回パソコン通信との電子メール交換を WIDE インターネット を中心に日本のインターネットの参加組織のうち 20 % 弱の組織で行ったが、一週間あたり 15000 通を越える利用がされており、この実験によりコミュニティが非常に大きく広がったという実感が得られた。今後さらに日本全域、世界との電子メールを交換できるような環境が提供されたときには、膨大な量の電子メールが取り交わされるであろう。パソコン通信形式の商用通信サービスに利用者にもインターネットとのコミュニケーションの必要性が大きいことがこの実験によって示された。

今回対象とした、パソコン通信サービスでは、メールボックスは集中的に管理されているので、これに他システムとの交換の機能を加えるのは意外な困難があった。また、日本語だけでサービスが構築されているため、メールヘッダなどで思わぬ問題点が発生した。

しかし、電子メールは電話、FAX と同様の通信手段として普及しつつあることが、この実験の中で感じられた。電子メールがアプリケーションとして、今後より一般的なものになるため、その状況で最も使いやすい環境を考えていくべきであろう。

利用調査にも現れているように、このような商用サービスを研究基盤として利用している研究者は多く、ネットワーク提供者の形態とその利用は必ずしも同一であるとみなすわけにはいかない。今後は、このようなネットワークの広がりを、より確実なものにしていく基盤作りが望まれると思われる。

## 1.9 電子メールの配送制御の背景

ここからは、今回のパソコン通信との相互接続において実装を行ったゲートウェイの技術の詳細について述べる。

電子メールシステムはインターネット上のユーザ間における最も基本的、かつ一般的なメッセージ交換の手段の 1 つとして非常に重要な地位を占めており、インターネットの黎明期より広く利用されてきた。また、近年、アカデミックネットワークと商用ネットワークといった、互いの運用ポリシーの異なるネットワーク間を相互接続し、その間で電子メールを用いたメッセージ交換を行おうとする試みが行なわれようとしており、また一部実際に行なわれている。この場合、互いの運用ポリシーに基づいた経路制御を電子メールシステムにおいて導入する必要が生じてくる。

従来の JUNET、あるいは現在の WIDE Internet 等の国内広域ネットワークにおいてこのような経路制御が行なわれた、あるいは行なわれている例としては、KDD 研究所における InetClub の利用に関する制御、東京大学における CSnet(現 CREN) の利用に関する制御がある。これらはメールの差出人および受取人をチェックし、不正な利用を制限するために行われていたものである。すなわちこれらの場合、どの組織からの、あるいはどの組織へのメールであるか、あるいはどのユーザのメールであるかのみが着眼点であり、例えばどのネットワークに所属している組織からのメールであるか、といったことは明示的にチェックされていなかった。この理由としては、これらのアクセス制限が行な

われはじめた当時、国内ネットワークにおいては運用ポリシーに基づいた相互接続という概念が希薄であったということが挙げられるであろう。またネットワーク参加組織数も現在と比較してかなり少数であり、制御も現在と比較してかなり容易であったであろうと考えられる。

しかしながら現在、国内 IP ネットワークにおいても、WIDE、TISN、JAIN、JOIN、SINET 等、運用ポリシーの異なった多数のネットワークが存在しており、またそれぞれが様々な形態で相互接続を行なっている。あるいは KARRN、TOHOKU-INET 等、地域ネットワーク形成の動向も各地域において行なわれており、実際これらの地域ネットワークもさまざまな相互接続点を通じてバックボーンネットワークに接続されてきている。このように複雑な構成を持つ現在のインターネット環境においては、従来行われてきた単なる start-point/end-point における経路制御にとどまらず、経路ネットワーク等も考慮した、より高度な経路制御を行う必要があると考えられる。

広域ネットワークにおける電子メール配送システムとして現在までに sendmail をはじめ、X.400/MHS、CC/Mail、あるいは RSCS 等、システムの種類、あるいはその規模によってさまざまなものが用いられてきた。これらのうち従来から現在に至るまでわれわれが構築、運用してきた TCP/IP、あるいは UUCP プロトコルを用いた広域ネットワークにおいては、UNIX システムにおける sendmail プログラムが広く用いられてきた。

この sendmail プログラムは、このような複雑な経路制御を行なおうとした場合、そのままでは後述するようなさまざまな問題点が存在する。しかしながら sendmail は従来よりわれわれの UNIX 環境で広く用いられてきており、またその動作もかなり安定している。またコンフィグレーションファイル (sendmail.cf) を書き換えるよって、ユーザレベルにおいてかなり柔軟に設定を行なうことも可能となっている。またそのソースファイルはバークレー版 UNIX の一部として無償で公開されており、これに若干の修正を加えることによってかなりのレベルの制御を行なうことも可能である。

将来的にはさらに進化した電子メール配送プログラムを開発、作成することが必要であろうが、現在の環境における short time solution として従来より広く用いられてきた sendmail プログラムを用いることはかなり有用であろうと考えられる。

## 1.10 経路制御における sendmail の問題点

前章でも述べたように、sendmail プログラムを用いて複雑なポリシーコントロールに基づいた制御を行うことはかなり困難である。その理由としては、

- From/To のペアを使用した制御の機能がない  
sendmail では From/To について全く個々に解釈し、それらを関連付けて解釈することを行っていない。そのため、ある特定のドメイン (ユーザ) からある特定のドメイン (ユーザ) へのメールについて選択的に制御するといったことはそのままでは不可能である。

- From による制御の機能がない

sendmail では、From アドレスによるアクセスコントロールを行なう機能がない。そのためどのような From アドレスを持ったメールであっても、その受信を防止する手段がない。またよく言われている ruleset 1 あるいは ruleset 2 による記述では From: フィールド、あるいは To: フィールドを単純に書き換えるだけであり、このようなアクセスコントロールを行なうことは不可能である。

- IP アドレスにおける経路制御の機能がない

sendmail は、IP アドレスに関連した経路制御を行なう機能を持っていない。そのため例えばどの IP アドレスから配送されてきたメールであるかということを利用した制御を行うことができない。

などが挙げられる。また sendmail の設定をユーザ側で行なう場合、sendmail.cf ファイルを修正することによって行なうが、このファイルの記述は一般的なものではなく、その修正にはかなりの経験が必要になるということも設定を困難にしていることのひとつとして挙げられるであろう。このことは電子メール配送プログラムとして sendmail を用いる限り不可避であるが、上記の 3 点については調査の結果、sendmail のソースプログラムの若干の修正、あるいは sendmail プログラムの修正と sendmail.cf ファイルにおける記述とを組み合わせるによって、かなりのレベルまで達成できることが判明した。

## 1.11 経路制御を行なう場合に考察すべき要素

具体的な事例について説明する前に、われわれが普段利用している広域ネットワークにおける電子メールに関する経路制御を行なう場合に、考慮すべき点について以下順を追って記述する。

### 1.11.1 どのネットワークから配送されたメールか

運用ポリシーの異なったネットワークが相互に接続される場合、どのネットワークから配送されてきたメールであるかを考慮する必要がある。なぜなら運用ポリシーの決定は通常各バックボーンネットワークの所有者によって決定され、またそのネットワークがどのバックボーンネットワークに所属するかはそのネットワークが所有する IP アドレスによって決定されるためである。勿論このためには IP レベルにおいて然るべき経路制御が行なわれている必要がある。

### 1.11.2 どのネットワークに配送されるべきメールか

同様に、送信されようとしているメールがどのネットワークをどのように経由しようとしているのかを考慮する必要がある。またこの場合、電子メール等のアプリケーション

ンレベルにおける経路の判断は、通常時の経路によって行われるべきであると考えられる。即ち異常時(緊急時)における経路に関しては、各ネットワーク間で別個に合意を定めるべきであるということである。これは 3.1 で述べたことに関しても同様である。

### 1.11.3 不正な経路指定を用いていないか

電子メールシステム、特に RFC822 に基づいたシステムにおいてはユーザレベルで容易に%等を用いた明示的ゲートウェイ指定をすることが可能となっている。このため直接 IP パケットのやりとりができないドメイン間においても、電子メールのやりとりができてしまう場合がある。この点についても考慮する必要がある。

### 1.11.4 正当なユーザから発信されたメールであるか

電子メールシステム、特に sendmail/RFC822 に基づいたシステムにおいては、ユーザレベルである程度メッセージヘッダの合成が可能である。そのため、これについても考慮する必要がある。

またこれらは個々に考慮するのではなく、それぞれを立体的に組み合わせて考慮する必要があることは言うまでもない。

## 1.12 sendmail における From/To について

sendmail を電子メール配送プログラムとして用いているシステムにおいては、From/To に関して特に注意すべき点がいくつか存在する。以下それについて記述する。

### 1.12.1 From:フィールドと UNIX From

sendmail では、一般的に From:フィールドは(スーパーユーザではない)一般ユーザにおいて容易に変更が可能であるため、From:フィールドを経路制御の際の判別材料として用いることは好ましくない。またメーリングリストにおいては From:フィールドにおけるユーザ名と発信ドメインが一致していない(すなわち From:フィールドにおけるポリシーと発信ドメインにおけるポリシーとが一致していない)場合がある。そのため sendmail においては From:フィールドではなく UNIX From(Sender) を経路制御の要素として用いるべきである。

### 1.12.2 To:フィールドと Mailer の引数

同様のことが To:フィールドにおいても適応される。即ち真の送り先として Mailer に与えられる引数によって経路制御を行なうべきである。

### 1.12.3 直接送信できない相手に対してどのようにエラーメールを伝えるか

この問題は非常に重要である。経路制御を行なった場合、即ち自らのメールプログラムが「送れない」よう設定している宛先に対してどのようにエラーメールを送るかということを検討する必要がある。このことに対する解決策としては、例えばエラーメールを送信する sendmail を別の sendmail.cf ファイルで動作させる、sendmail 以外のプログラムをエラーメールの配送に用いる、SMTP のセッション中に相手にエラーを伝える、等の方法が考えられる。ちなみに後述する実装例においては三番目の方法を用いている。

## 1.13 sendmail における実装例 - pcgw における実装 -

WIDE Project WPNC-WG では、WIDE Internet および NIFTY-Serve(以下単に NIFTY とする)、PC-VAN(以上 2 つについては運用中)、ASCII-NET(現在準備中)の各商用ネットワークとの間で相互接続を行ない、WIDE Internet をはじめとするアカデミックネットワーク(現在のところ WIDE Internet 全域)とこれらの商用ネットワークとの間で電子メールによるメッセージ交換を行なう実験を 92 年 10 月より開始した。

この実験では、電子メールにおける経路制御を WIDE Internet 側のゲートウェイマシンである pcgw.wide.ad.jp<sup>1</sup>(以下単に pcgw とする)上で動作している sendmail プログラムを用いて行なっている。以下に pcgw において行なっている経路制御の実装について記述する。

現在 pcgw は NIFTY、PC-VAN に対してそれぞれ point-to-point のシリアル回線を所有しており、その間は TCP/IP プロトコルによって結合されている。pcgw では gated プログラムを用いることにより NIFTY、あるいは PC-VAN 側のゲートウェイには pcgw 以外のマシンからは直接到達することができないように設定されている。同様に NIFTY、PC-VAN 側のゲートウェイから pcgw 以外の WIDE Internet 側のマシンにも到達できないよう設定されている。NIFTY、PC-VAN 共に相手側の受け口は UNIX ワークステーションであり、pcgw と SMTP プロトコルを用いて電子メールの交換を行なう。

即ちインターネット側から発信されたメールはいったん pcgw がすべて受信し、それぞれの商用ネットワークのゲートウェイに渡される。商用ネットワーク側に渡されたメールは、相手側のゲートウェイにおいて RFC822 形式からそれぞれの商用ネットワーク側のフォーマットに変換され、相手側のシステムに渡される。また同様に各商用ネットワーク側からのメールは、相手側のゲートウェイにおいて RFC822 形式に変換され、SMTP プロトコルで static に pcgw に渡され、pcgw においてその宛先に応じて配送される。すなわち全てのメールは必ずいったん pcgw を通過するため、pcgw の sendmail において経路制御を行なうことによって、すべてのメールの経路制御を行なうことが可能となる。

pcgw の sendmail において行なった手法について以下に説明する。なお pcgw における

<sup>1</sup>pcgw.wide.ad.jp [133.4.11.20]、pcgw は WNOC-SFC に設置されている。

sendmail プログラムは WIDE 版 sendmail の 5.67+1.6W 版を使用し、sendmail.cf ファイルは WIDE 版 sendmail.cf の 2.8Wb 版に基づいたものを使用した。

### 1.13.1 To による制限

To による制限は比較的容易である。sendmail では To 宛先 (RCPT To: で与えられる宛先) に対してどのメーラを呼び出すかを sendmail.cf の ruleset 0 で決定しており、これを用いて経路制御を行なうことが可能である。

pcgw において ruleset 0 で行った設定の概要を以下に示す。

- “.UUCP” 形式のメール  
UUCP 形式のメールは扱わないよう変更した。即ちや.UUCP のようなアドレスを持つメールは pcgw においてはエラーとなる。
- “.BITNET” 形式のメール  
現在のところ BITNET(JP) との間で相互接続の合意が得られていないため、.BITNET で終わる宛先に関してはエラーとなるよう設定した。
- “.JUNET” 形式のメール  
無条件にエラーとした。
- NIFTY/PC-VAN あてのメール  
static にそれぞれのゲートウェイに転送するよう設定した。この記述をするにあたっては、WIDE 版 sendmail において拡張された機能 (MX を参照しない配送においても IP アドレスでなくホスト名を直接記述可能) を用いることにより、よりエレガントな記述を行なうことができた。
- その他の宛先をもつメール  
特定のドメインあての以外のメールはすべてエラーとするように設定した。具体的には、あるクラスに許可ドメイン名を列挙しておき、それに合致したメールだけを配送するように設定した。またこの設定を行なう場合、クラスにドメイン名を使用可能であると sendmail.cf ファイルにおける設定が非常に容易になるが、WIDE 版 sendmail ではオリジナルの Berkeley 5.67 版に対してこの機能が拡張されているため、容易に設定を行うことができた。
- user%dom.xx.jp@wide-domain.xx.jp という形式のメール  
pcgw を通過できないように対策した。具体的には “@” の左側のトークンが 1 つ、あるいはピリオドのみで区切られている場合にのみ tcp メーラを呼び出すように記述した。これは主として商用ネットワーク側からのアタックに対する対策である。
- user%niftyserve.or.jp@pcgw.wide.ad.jp という形式のメール  
通常であると ruleset 0 で Local domain part として @ から右が無条件に剥がされて

しまうため、上記の制限を利用することができない。そのため“@”の右側を剥がす際に左側を構成するト - クンが1つだけであるかどうかをチェックすることとした。これによりエラーとすることが可能となった。

### 1.13.2 From による制限

From アドレスによる制限を導入した。チェックには ENVELOPE 構造体の `e.from` を用いているため、ヘッダの `From:` フィールドではなく `MAIL From:` によって与えられる `From`(UNIX `From/Sender`) になる。これにより前述した問題を避けることができる。チェック方法としては `sendmail.cf` の `ruleset 0` における記述および `sendmail` のソースプログラムの変更の双方を用いることとした。具体的には、`ruleset 0` が `sendmail` プログラムにおいて `From` の `parse` にも用いられていることを利用し、`From` が `ruleset 0` によって `parse` できなかった場合にエラーとするよう、`sendmail` の該当部分のソース (`envelope.c`) を書き換えた。

### 1.13.3 IP アドレスによる制限

あらかじめ登録されたホスト群以外から受信されたメール以外はすべてエラーとして発信人に返送するように設定した。このチェックには NNTP-1.5.11 が `nntp_access` ファイルによってアクセス制限を行なっているルーチンを参考にした。なお NNTP-1.5.11 はフリーソフトウェアとして公開されている。実際のチェックは SMTP セッションを実際に行っているルーチン (`svrsmtp.c`) から呼び出された部分で行っている。

### 1.13.4 NIFTY↔PC-VAN 間のメール

以上で通常のインターネットと商用ネットワーク間のメールに関する経路制御を行なうことが可能となるが、NIFTY↔PC-VAN 間において `pcgw` を経由して直接やりとりされるメールに関しては制限を行なうことができない。そのためこのメールに関しては、`sendmail` のソースに用意されているフックルーチンである `checkcompat()(conf.c)` を用いて行なうこととした。このルーチンはメールの配送の際に1回毎に必ず呼び出され、通常は常に `TRUE` を返すよう設定されている。そこでここに `From/To` をペアでチェックするルーチンを書き加え、NIFTY↔PC-VAN 間のメールの交換を制限することとした。具体的には `From/To` の“@”の右側を `exact` にチェックし、NIFTY↔PC-VAN 間を通過するメールであった場合には `FALSE`(エラー) とするように変更を加えた。

### 1.13.5 エラーメールを返送する機構

前述したように、電子メールにおいてこのような機構を導入する場合、直接返送出来ない相手に対して、どのようにエラーメールを返すか、ということが大きな問題となる。`pcgw` ではエラー判定を SMTP のセッション中で相手システムからの `RCPT To:` に対して



否定応答 (554) を行なうことで自分が直接送信することのできない相手に対してもエラーを通知することが可能なよう配慮している。またこのように RCPT To: においてエラーとすれば、相手のシステムからみた場合、誤ったアドレスを宛先に指定した場合 (User unknown) と同様のエラー処理が行なわれることになるため、通常のエラーメールと同様に扱われることとなる。これは様々な点で都合がよい。この設定は `svrsmtp.c` (ユーザ側の `sendmail` からみた場合 `usersmtp.c`) において行なわれている。

なお、実際の変更に関しては、本論文の付録を参照されたい。

## 1.14 まとめ、及び今後の展望

今後の展望として、`checkcompat()` を用いて From/To の比較をさらにエレガントに行なう (例えば DBM 等を用いる、ファイルに From と To のペアを記述する)、あるいは経路制御に関する記述を `sendmail.cf` 以外のファイルにまとめてより平易に記述可能にする、といったことが挙げられる。また今回の `pcgw` における試みはあくまで short time solution であり、今後のことを長期的に考えた場合、`sendmail` の alternative を開発することまで眼中において考慮する必要があるように思われる。

また現在 `pcgw` で行われているアクセスコントロールが必ずしも完璧なものであるとは限らない。そのためこのアクセスコントロールは管理者による定期的なログファイルのチェックと併用する必要があると考えられる。ちなみに筆者は `sendmail/SMTP` を用いる限り完璧なアクセスコントロールは事実上不可能であると考えている。

このあたりは、例えば SMTP プロトコルを拡張する、あるいは IP プロトコルにさらに高度な authentication を導入するといったアプローチが必要になってくるであろう。

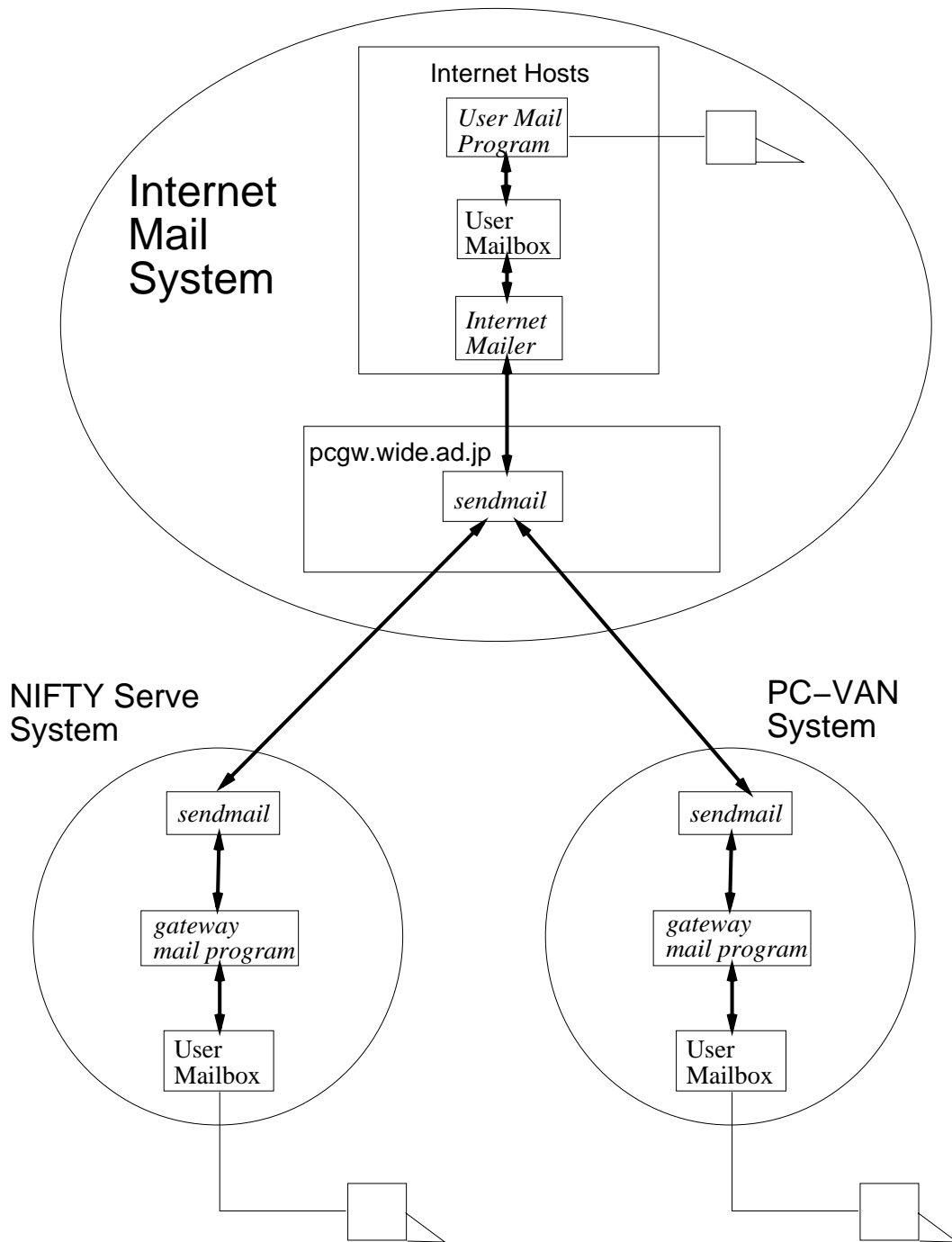


図 1.1: 接続形態 (電子メール) の概念図

From: GBE00070@niftyserve.or.jp  
Return-Path: <GBE00070@niftyserve.or.jp>  
Message-Id: <9209080655.AA14255@inetnif.niftyserve.or.jp>  
Date: Tue, 8 Sep 1992 15:54:00 JST  
Subject: Re:WIDE-NIFTY Interconnection  
To: shin@wide.ad.jp

赤木宏行                    GBE00070 09/08 15:54  
題名: Re:WIDE-NIFTY Interconnection

発信者:

From: CHEMOSL@pcvan.or.jp  
Return-Path: </C=JP/ADMD=CANDCVAN/PRMD=PCVAN/DD.ID=CHEMOSL/@pcvan.or.jp>  
Received: by pcvan.or.jp; Mon, 14 Sep 92 15:13:42 +0900  
Original-Encoded-Information-Types: IA5-Text  
Priority: normal  
Message-Id: <920914151335.230@pcvan.or.jp>  
Subject: Please reply to me.  
Date: Mon, 14 Sep 92 15:13:39 +0900  
To: shin@wide.ad.jp

PCVAN-Message-Id: #04411087  
PCVAN-Date: 92/09/13 12:57:22

初めまして。電子メールの相互接続実験はたいへん有意義なも

図 1.2: 実験開始直後のメール (NIFTY Serve:上、PC-VAN:下)

```
From: PIA00202@niftyserve.or.jp
Return-Path: <PIA00202@niftyserve.or.jp>
Message-Id: <9209160250.AA21226@inetnif.niftyserve.or.jp>
Date: Wed, 16 Sep 1992 11:47:00 JST
Subject: ミーティング
X-Original-From: 中居 良則          PIA00202 09/16 11:47
To: shin@wide.ad.jp, PIA00202@niftyserve.or.jp
```

吉村殿

```
From: SUF68034@pcvan.or.jp
Return-Path: <SUF68034@pcvan.or.jp>
Received: by pcvan.or.jp; Mon, 26 Oct 92 10:53:58 +0900
Original-Encoded-Information-Types: IA5-Text
Priority: normal
Message-Id: <921026105352.15604@pcvan.or.jp>
Subject: Re: short questionnaire from WIDE project
Date: Mon, 26 Oct 92 10:53:55 +0900
To: wpnc-answer@wide.ad.jp
X-Pcvan-Message-Id: #04557953
X-Pcvan-Date: 92/10/26 10:47:42
```

1) (1 から 6 までの数字、6 の場合、具体的に)

図 1.3: 現在の NIFTY-Serve:上、PC-VAN:下 からのメール

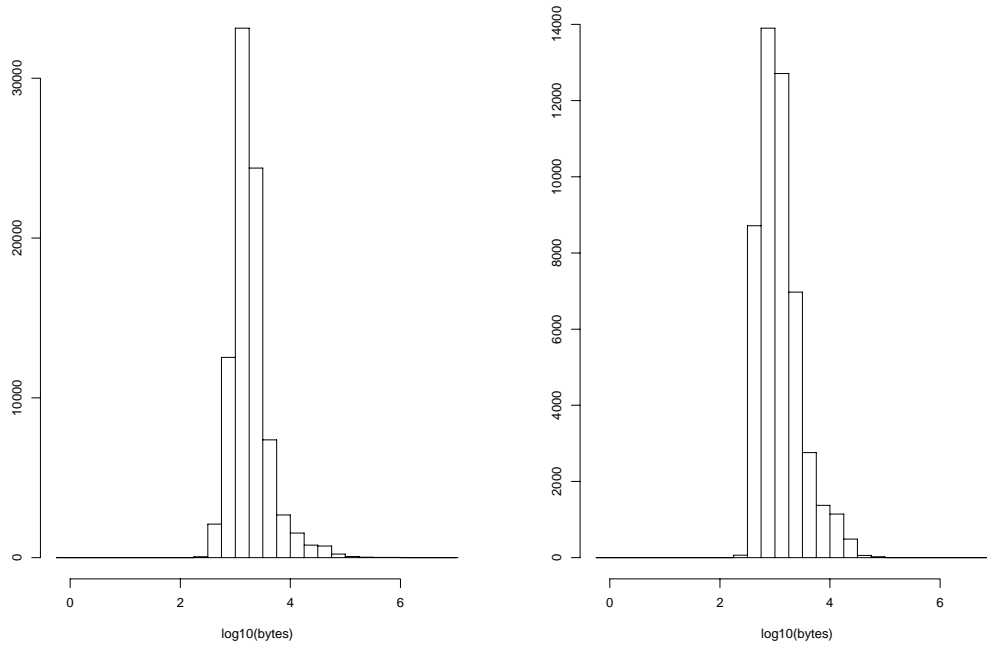


図 1.4: メールのサイズ: インターネットからパソコン通信へ (左)、パソコン通信からインターネットへ (右)

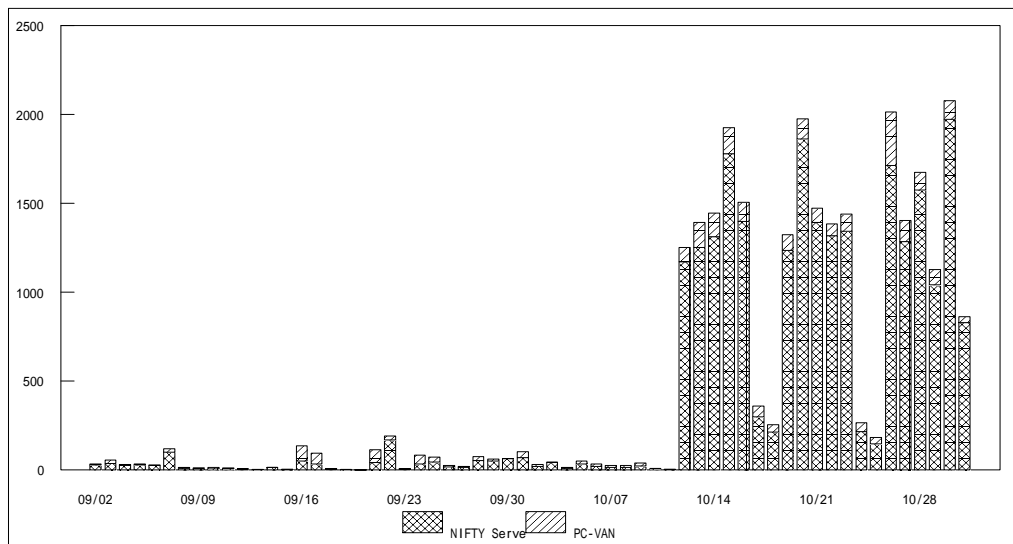


図 1.5: WIDE Internet から、パソコン通信 へのメール (日計)

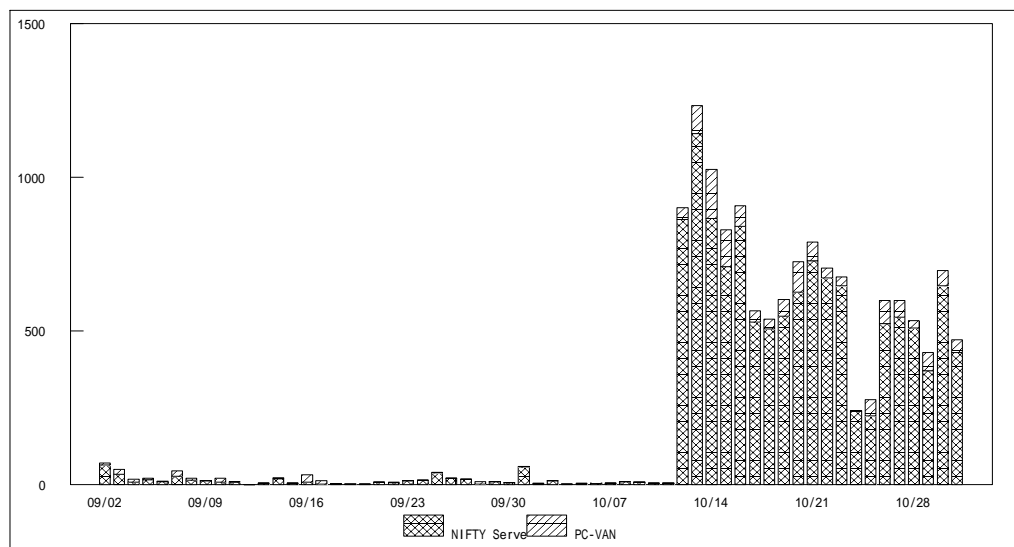


図 1.6: パソコン通信から WIDE Internet へのメール (日計)

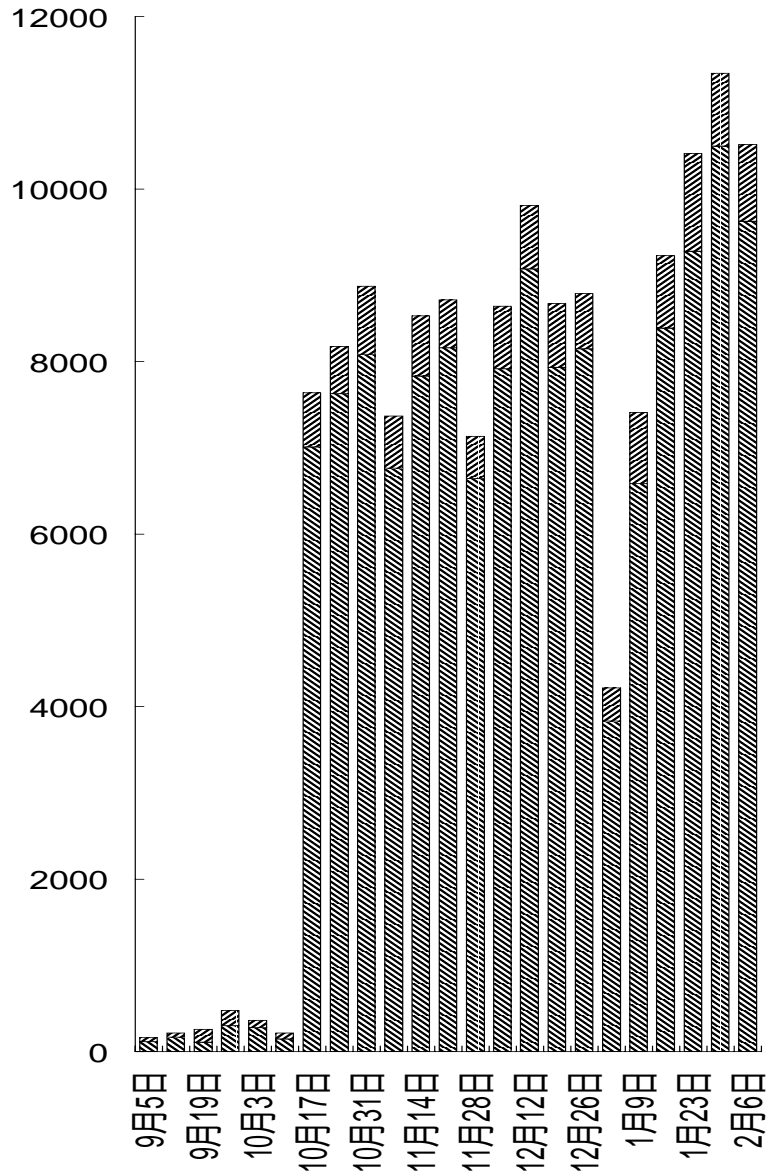


図 1.7: 利用者数: インターネットからパソコン通信へのメール (週計)、上: PC-VAN、下: NIFTY-Serve

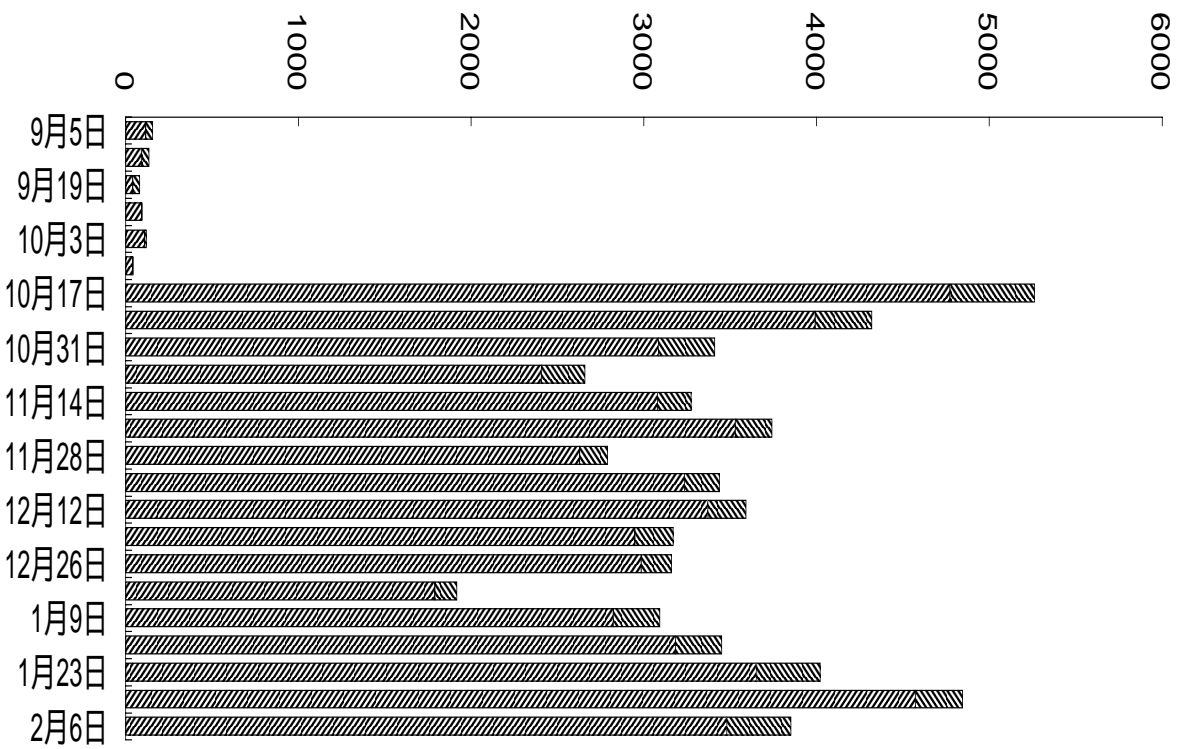


図 1.8: 利用者数: パソコン通信からインターネットへのメール (週計)、上: PC-VAN、  
下: NIFTY-Serve



メール相互接続利用者各位

1992 年 10 月 26 日

利用状況調査のお願い

WIDE プロジェクト吉村 伸

WIDE Project の吉村です。今回の WIDE Internet とパソコン通信の相互接続に関して、利用者の方に簡単な調査を行なわせていただきたいと思います。なお、調査の回答に関しては個別には公開いたしませんので、率直に書いてください。

回答は、wpnc-answer@wide.ad.jp 宛に電子メールでお送りください。調査に関してのご質問は、wpnc-wg@wide.ad.jp 宛にお願いします。

- 1) どのような目的で利用しましたか。(複数回答可)
  1. 研究連絡 2. 研究用データの送付 3. 私信 4. テスト
  5. メイリングリスト 6. その他(利用目的を具体的に書いてください)
- 2) 今回の実験に関する感想、問題点など感じたことがあれば簡単にお書きください。

回答欄

- ----- cut here ----- cut here -----

1) (1 から 6 までの数字、6 の場合、具体的に)

2)

-----

図 1.9: 第一回利用者アンケート本文

メール相互接続利用者各位

1993 年 1 月 16 日

利用状況調査のお願い

WIDE プロジェクト吉村 伸

WIDE Project の吉村です。今回の WIDE Internet とパソコン通信の相互接続に関して、利用者の方に簡単な調査を行なわせていただきたいと思います。なお、調査の回答に関しては個別には公開いたしませんので、率直に書いてください。なお、メーリングリストでの利用では、意図的にこのゲートウェイを利用したつもりがなくて、このメールを受け取ることもあるかと思えます。

回答は、wpnc-answer@wide.ad.jp 宛に折り返し電子メールでお送りください。整理の都合上、かならずこのメールを受け取ったところから返信してください。調査に関してのご質問は、wpnc-wg@wide.ad.jp 宛にお願いします。

1) どのような目的で利用しましたか。(複数回答可)

1. 研究連絡
2. 研究用データの送付
3. 私信
4. テスト
5. メーリングリスト
6. その他 (利用目的を具体的に書いてください)

2) 利用している際に、文字コードに関して何か困ったことや問題点に出会ったことがありますか? あればその内容について簡単にお書き下さい。

3) 将来、運用ベースに移行する場合には、CompuServe と Internet の間で行われているのと同じように、パソコン通信のユーザには、メール一通あたり、メールの大きさに応じてなどの方式で課金が行われることが考えられますが、その場合にも継続して使いますか? また料金に関して意見があればお書きください。

4) 今回の実験に関する感想、問題点など感じたことがあれば簡単にお書きください。

回答欄

- - - - - cut here - - - - - cut here - - - - -

1)

2)

3)

4)

## 第 2 章

### 付録

#### 2.1 ruleset 0

S0

```
# first make canonical
R$*<$*>$*          $1$2$3          defocus
R$+                $:$>3$1        make canonical
R$*<@$=w>$*        $:$1<@$w>$3      get into u@$w form
R$*<@$=w.$m>$*     $:$1<@$w>$3

# handle special cases
R$*<@[+]>$*        $:$1<@$ [[2]]$>$3      numeric internet addr
R$*<@[+]>$*        $#tcp$@[2] $:$1@[2]$3  numeric internet spec
R$+                $:$>6$1
R$-<@$w>           $#local$:$1
R@                 $#error$:Invalid address  handle <> form

# canonicalize using the nameserver if not internal domain
R$*<@$*$~W>$*      $:$1<@$ [[2$3]]>$4
#R$*<@$->$*        $:$1<@$ [[2]]>$3      not work in NFS4.0
#R$*<@$->$*        $:$1<@$2.$m>$3      if nameserver fails

# now delete the local info
#R<@$w>:$*         $$>0$1          @here:... -> ...
#R$*<@$w>         $$>0$1          ...@here -> ...
# we accept exact match ONLY
R<@$w>:$-         $$>0$1          @here:... -> ...
R$-<@$w>         $$>0$1          ...@here -> ...

# PC-VAN SPECIAL
#R<@pcvan.or.jp>:$-  $$>0<@pcvan.or.jp>:$P$1/
#R$-<@pcvan.or.jp>  $$>0$P$1/<@pcvan.or.jp>
```

```
#####
# End of ruleset zero preamble #
#####

#####
### Machine dependent part of Rule Zero ###
#####

# alias resolution XXXXXXXX
R$*<@JUNET>$*      $:$1<@JUNET.AD.JP>$2
R$*<@JUNET>$*      $#error$:Obsolete address: JUNET @JUNET
R$*<@JUICE>$*      $:$1<@JUICE.AD.JP>$2
R$*<@$*.JUICE>$*   $:$1<@$2.JUICE.OR.JP>$3
#
R$*<@$*ALIAS_NAME>$* $:$1<@$2REAL_NAME>$3

# domain that connects this host by uucp to gateway mapping XXXXXXXX
R$*<@$*DOMAIN_NAME>$* $:<@GATEWAY.UUCP>:$1<@$2DOMAIN_NAME>$3
R$*<@$*=$A>$*        $:<@$A.UUCP>:$1<@$2$3>$4
R$*<@$*=$B>$*        $:<@$B.UUCP>:$1<@$2$3>$4
R$*<@$*=$C>$*        $:<@$C.UUCP>:$1<@$2$3>$4
R$*<@$*=$D>$*        $:<@$D.UUCP>:$1<@$2$3>$4

#####
# Dispatch to the mailer #
#####

# resolve UUCP connections
R<@+$+.UUCP>:$+      $#uucp@$1$:$2 @host.UUCP:...
R$+<@+$+.UUCP>      $#uucp@$2$:$1 user@host.UUCP

# resolve fake top level domains by forwarding to other hosts
R$*<@$*JUNET>$*     $#tcp@$X$:$1<@$2JUNET>$3 user@dom.JUNET
R$*<@$*JUNET>$*     $#error$:Obsolete address: JUNET user@dom.JUNET
# .BITNET is not allowed, yet.
R$*<@+$+.BITNET>$*  $#tcp@$2.$Y$:$1<@$2.BITNET>$3 user@host.BITNET

# resolve SMTP traffic
# (1.1) I cast all $*niftyserve.or.jp mail via inetnif.niftyserve.or.jp
R$*<@$*niftyserve.or.jp>$* $#tcp@[inetnif.niftyserve.or.jp]$: $1<@$2nifty
serve.or.jp>$3
# (1.2) I cast all $*pcvan.or.jp mail via inet_gate.pcvan.or.jp
R$*<@$*pcvan.or.jp>$*  $#tcp@[inet_gate.pcvan.or.jp]$: $1<@$2pcvan.or.jp>$3
# (2) Access control now available
```

```

R$-<@*$=$=0>$*          $#tcp$@$2$3$:$1<@$2$3>$4          user@host.domain
R$-.$-<@*$=$=0>$*        $#tcp$@$3$4$:$1.$2<@$3$4>$5        user.name@host.domain
R$-.$-.$-<@*$=$=0>$*     $#tcp$@$4$5$:$1.$2.$3<@$4$5>$6     user.middle.name@host.
domain
# other mails, error.
R$*<@*$>$*              $#error$:You are not allowed to use this address

# remaining names must be local
R$+                      $#local$:$1                      everything else

```

## 2.2 patch(抜粋)

```

*** conf.c.ORG Wed Mar 25 18:43:35 1992
--- conf.c Tue Oct 13 14:45:54 1992
*****
*** 387,392 ****
--- 390,445 ----
                return (FALSE);
        }
        # endif EXAMPLE_CODE
+ # ifdef RISTRICT
+     register STAB *s;
+     register char *p, *q;
+     register bool nifty = FALSE, pcvan = FALSE;
+
+     /* niftyserve <=> pcvan mail check routine, it's very dirty */
+     s = stab("tcp", ST_MAILER, ST_FIND);
+     /* First: checking this mail is binded into tcp mailer */
+     if (s != NULL && to->q_mailer == s->s_mailer)
+     {
+         /* Second: address check */
+         p = CurEnv->e_from.q_user; /* from username */
+         if ((q = index(p, '@')) != NULL)
+         {
+             if (strncmp(++q, "niftyserve", 10) == 0)
+                 nifty = TRUE;
+             else if (strncmp(q, "pcvan", 5) == 0)
+                 pcvan = TRUE;
+             else /* OK */
+                 goto skip;
+             p = to->q_user; /* to username */
+             if ((q = index(p, '@')) != NULL)
+             {
+                 if (strncmp(++q, "niftyserve", 10) == 0)
+                     nifty = TRUE;

```

```

+         else if (strncmp(q, "pcvan", 5) == 0)
+             pcvan = TRUE;
+         else /* OK */
+             goto skip;
+         if ((nifty == TRUE) && (pcvan == TRUE))
+             MarkFail = TRUE; /* mark failure flag */
+     }
+ }
+ }
+ if (MarkFail == TRUE)
+ {
+     MarkFail = FALSE; /* reset flag */
+     usrrerr("You are not allowed to use this address");
+     syslog(LOG_INFO,
+         "%s: CHECK: Illegal request from %s (%s), wants to %s",
+         CurEnv->e_id,
+         RealHostName,
+         inet_ntoa(RealHostAddr.sin_addr),
+         CurEnv->e_to);
+     /* NoReturn = TRUE; to supress return copy */
+     return (FALSE);
+ }
+ skip:
+ # endif RISTRIC
+
+     return (TRUE);
+ }
+ ^L/*
*** envelope.c.ORG      Sun Apr 21 07:55:05 1991
--- envelope.c Thu Sep  3 00:36:06 1992
*****
*** 484,489 ****
--- 484,492 ----
    }

    SuprErrs = TRUE;
+ # ifdef RISTRIC
+     MarkFail = FALSE; /* initial value (OK) */
+ # endif RISTRIC
+     if (from == NULL || parseaddr(from, &CurEnv->e_from, 1, '\0') == NULL)
+     {
+         /* log garbage addresses for traceback */
+         *****
+         *** 492,500 ****
+         --- 495,512 ----
+         # ifdef LOG

```



```
+         message("554",
+                 "You can not send mail to this domain");
+         syslog(LOG_INFO,
+                 "%s: CHECK: Illegal request from %s (%s),
wants to %s",
+                 CurEnv->e_id,
+                 RealHostName,
+                 inet_ntoa(RealHostAddr.sin_addr),
+                 CurEnv->e_to);
+         break;
+     }
+ #endif    RISTRICT
           if (!bitset(QBADADDR, a->q_flags))
               message("250", "Recipient ok");
           else
```