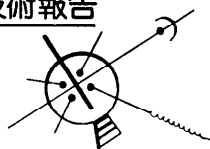


技術報告



シリアルラインによるコンピュータ間通信†

浅 見 徹** 徳 田 雄 洋***
徳 田 英 幸**** 村 井 純****

1. は じ め に

郵政省のパソコン推奨方式²⁶⁾の登場により、個人がデータ通信を行うための安価な手段が提供できるようになった。この結果、ユーザ相互間で自由に通信を行う気運が盛り上がり、我が国もようやく電子コミュニティの時代を迎えつつある。一方、国際的には米国を中心に、種々の通信手段を用いて、共通の興味・利害関係を持ったユーザ・グループが種々の電子コミュニティを形成するに至っている。研究開発者の形成する電子コミュニティの場合も、代表的な ARPA インタネット¹⁴⁾、CSNET⁷⁾、USENET (USErs' NETwork)¹²⁾のコミュニティを中心に、それぞれのコンピュータネットワークおよび研究開発活動の発展とともに世界各地でますます盛んになってきている。我が国においても、1984年に USENET への参加地点が出現するに至り、本格的な研究開発用電子コミュニティの時代が始まろうとしている^{2), 3), 19), 21)~23)}。

これらの研究開発用電子コミュニティを支えるネットワークは種々の通信媒体から構成されている。高度なサービスを提供する米国国防総省の ARPA インタネットは、56 Kbit/s 専用線とローカルエリアネットワークを、北米の大学の計算機科学科を中心に結ぶ CSNET は、ARPA インタネット、公衆パケット通信網および公衆電話網を、そして UNIX の利用者たちを結んで電子ニューズレターや電子メールなどのサービスを提供している USENET や

UUCPNET は、主として公衆電話網を媒体にして構成されている。

これらのネットワークに見られる種々の通信技法においてもっとも安価に異機種ホスト間での通信を実現しているのが、シリアルラインを利用した通信である。EIA RS-232C¹¹⁾に基づくコネクタおよび信号規格で、ASCII 文字を調歩同期方式で送受するこの通信方式は、多くのコンピュータ・システムで端末・ホスト間通信に利用されているもっとも普遍的な通信手段である。ここでは、このような通信方式を総称してシリアルライン通信と呼ぶことにする。CSNET や UUCPNET におけるモデムを介した通信、パーソナルコンピュータ間やパーソナルコンピュータとホスト間でのファイル転送、およびプログラムのダウンロードなどもシリアルライン通信を使って実現されている。

これらのコンピュータ通信プロトコルは、日本の研究者が国際的な研究開発用電子コミュニティに参加していく際に欠くことのできない技術であるにもかかわらず、その実態について従来ほとんど解説されなかった。本稿では、日本の研究者が当面理解する必要がある UUCP プロトコル²⁵⁾を軸に、これらシリアルラインを介したコンピュータ通信技法について解説する。

2. 通信網とのインタフェース

シリアルラインによる通信プロトコルの代表的なものには、UNIX 間通信に使われている UUCP 用の G-プロトコル²⁹⁾や F-プロトコル²⁰⁾、Ward Christensen が CP/M 用に開発した XMODEM (MODEM 7)¹⁷⁾、および米国コロンビア大学が無料で配布し、IBM 370 シリーズのようなメインフレームから APPLE II などのマイクロコンピュータに至るコンピュータ上で実現されている KERMIT^{9), 10)}があり、欧米で広く使用されている。通信媒体は、スルモデムケーブル、専用

† Computer Communications through Serial Lines by Tohru ASAMI (KDD Research & Development Labs.), Takehiro TOKUDA (Dept. of Computer Science, Yamanashi University), Hideyuki TOKUDA (Dept. of Computer Science, Carnegie-Mellon University) and Jun MURAI (Tokyo Institute of Technology, Computer Center).

** KDD 研究所

*** 山梨大学計算機科学科

**** カーネギーメロン大学計算機科学科

***** 東京工業大学総合情報処理センター

線、公衆電話網、公衆パケット交換網など多岐にわたっている。専用線や公衆電話網の場合はモデムを、公衆パケット交換網の場合は PAD を用いて、網と計算機のインタフェースをとっている。

これらのプロトコルでは、送信データはすべて ASCII 文字 (7 ビット + 1 パリティビット) に変換して送られる。パリティビットの機能上の意味は通信網やオペレーティングシステムに依存するため、8 ビット中の何ビットを用いてデータを送信すべきかをあらかじめ決めておく必要があり、これをデータパス長という。データパス長は、公衆電話網やスルモデムケーブルを対象にした XMODEM では 8 ビット固定であるのに対し、KERMIT では電話網か公衆パケット交換網かに応じて 8 ビットまたは 7 ビット長をユーザが選択できる。また、UUCP は複数のプロトコルの存在を許しているため、相手システムとのハンドシェイク時に公衆電話網の場合 8 ビット (UUCP G-プロトコル) を、公衆パケット交換網の場合 7 ビット (フロー制御機能を持つ UUCP F-プロトコル) を自動選択する。

モデムに関しては、ベル 212 A (1200 bps) または CCITT V. 22 (1200 bps)⁵⁾ 相当の全二重モデムを用いるのが一般的である。これは、既存の通信プロトコルとコンピュータ・端末間インタフェースの両者に全二重のものと半二重のものがあるため、種々のコンピュータ間で相互通信するには全二重モデムの方が汎用性が高いためである。また、HDLC のような全二重プロトコルをソフトウェアで実現する場合にも、全二重モデムの方がソフトウェアのオーバーヘッドが少ない利点がある。特に米国においては、1980 年前後に、LSI 化により小型、低価格 (約 300 米国ドル) 化を図ったモデムが数多く市販され、1200 bps 全二重モデムの使用傾向を助長した。これらは、RS-232 C ポート 1 本で自動発信・受信が可能であり、マイクロプロセッサ内蔵でプログラム機能があるため、回線や利用形態に応じて通信速度などのパラメータを容易に

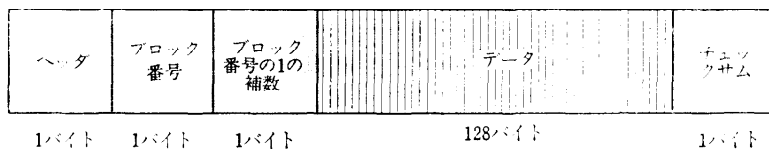
変更でき、それ以前のモデムと比較して大幅に操作性が向上している。このような歴史的背景からベル 212 A 型のモデムが米国で普及しているが、このベル 212 A と V. 22 モデムは相互通信可能であるため、これらのモデムにより事実上全世界の多数の計算機間での通信が可能になっている。

3. パケット構成と誤り制御

本章では、パケット構成と誤り制御方式について解説する。XMODEM は図-1 に示すパケット構成²⁴⁾を持ち、パケット長は固定で 132 バイトである。この方式の伝送手順では、1 パケットごとに ACK、および NACK 文字を用い、送信側で ACK (受信通知) を受信すると次のパケットを送信し、NACK (再送要求) を受信したときには前のパケットを再送する。簡易ではあるが伝送効率はあまり高くない。これに対して KERMIT (パケット長のデフォルト値は 96 バイトで、ユーザが選択可能) や UUCP ではさらに複雑な伝送手順を採っている。

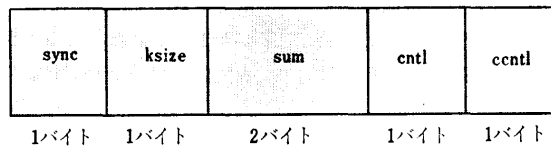
たとえば UUCP の G-プロトコルではパケットをデータ転送用パケットと制御用パケットに分け、パケット制御テーブルの初期設定、データ転送用パケットの受信通知、再送要求などに用いられる制御用パケットにも誤りチェックができるようにしている。図-2 に制御用パケットおよびデータ転送用パケットの構成を示す。パケット長は、相手受信バッファ長に合わせて $(6 + 2^N)$ バイト ($N=0$ または $5 \leq N \leq 12$) の中から自動選択される。また、伝送制御手順にはスライディング・ウィンドウ・プロトコル²⁵⁾を採用しているため、伝送効率は XMODEM に比して格段に向上している。

UUCP G-プロトコルのスライディング・ウィンドウ・プロトコルでは、送信側は、受信側からの確認のない送出パケットのシーケンス番号の最小値 S_1 、最後の送出パケットのシーケンス番号 S_2 、次に実際に送出するパケットのシーケンス番号 S_3 によってパケ

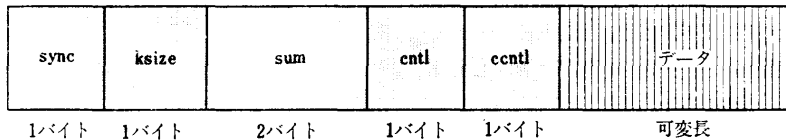


ここで、ヘッダは SOH (01_h)、ブロック番号は 00_h~FF_h、チェックサムは 128 個のデータの和 (キャリは無視) である。

図-1 XMODEM のデータパケットの構成²⁴⁾



(a) 制御用パケットの構成



(b) データ転送用パケットの構成

sync: パケットの先頭をしめすヘッダで, 020_(a).

ksize: 受信バッファ長を示すバッファ長テーブルの配列番号. 制御用パケットのとき 011_(a), データ転送用パケットのとき 000_(a)~010_(a) で 1, 2¹, ..., 2¹² バイトを表す.

ccntl: チェックサム. $ksize \oplus \text{sum 上位 8 ビット} \oplus \text{sum 下位 8 ビット} \oplus \text{cntl}$. ただし, $\oplus$ は排他的論理和.

sum: 制御用パケットのときはパケット制御情報部のチェックサムで, 上位 8 ビットと下位 8 ビットに (0125252_(a)-cntl) のそれぞれ下位 8 ビットと上位 8 ビットを持つ. データ転送用パケットのとき, 上位 8 ビットと下位 8 ビットに (0125252_(a)-(データ部に対する関数 chksum の値) $\oplus$ cntl) のそれぞれ下位 8 ビットと上位 8 ビットを持つ.

cntl: パケット制御情報. 制御用パケットのとき 2~4 ビットに処理指定, 5~8 ビットにパケット制御テーブルに設定する値を入れる. データ転送用パケットのとき 2~4 ビットにパケットシーケンス番号を, 5~8 ビットに相手ホストの受信したパケットシーケンス番号を, 1 ビット目に 1 を, 0 ビット目にはデータ部のデータが受信バッファ長より小さいとき 1 を入れる.

図-2 UUCP G-プロトコルのパケット構成

ット転送の管理を行い, S_1 - S_2 がウィンドウ長 WS 以下であれば, 送信側は受信側からの送信許可を得ずに次のパケットを送信し, S_2 をインクリメントする. 受信側からの S_1 パケット受信確認通知を得ると S_1 をインクリメントする. 受信側からのパケット再送要求があった場合は, そのパケットを送信して S_1 , S_2 とする. また, 受信側からの応答待ちでタイムアウトの場合は, S_1 からパケットを順に再送する.

受信側は受信ウィンドウ長 WR 分のパケットをいったん収容し, 受信パケットの順序をチェックする. したがって, 受信確認をしていないシーケンス番号の最小値を R_1 とすると, シーケンス番号 N が $R_1 \leq N \leq R_1 + WR$ のパケットの受信を許可する. ここで, パケットのシーケンス番号は 2 の累乗 WS を法とし, WR は WS の 1/2 以下とする. 実際には, $WS=8$ $WR=1$ であり, 図-3 に送信側から 0~3 のシーケンス番号のパケットを送り, 正しく受信できたときの手順を示す.

4. ホスト間接続方法

これらシリアルライン通信は, ISO の OSI モデル

に従ったプロトコルではない. 特にホスト・端末間通信を拡張するという方向で設計された経緯から, ネットワークとトランスポート層に該当する概念が希薄である. これらの接続は, 人間が端末エミュレータ・プログラムを使って設定しなければならない. したがって, これらのシリアルライン通信プロトコルは端末エミュレータ・プログラムの下位プロトコルとして実装されているのが普通である. KERMIT プロトコルの端末エミュレータ・プログラムは KERMIT 自体である. XMODEM プロトコルの場合, CP/M 上では MODEM や XMODEM, UNIX 上では TIP¹⁸⁾, Macintosh 上では MacTerminal¹⁹⁾を端末エミュレータ・プログラムとして利用している. UUCP では uucico がこれに該当する.

リンク設定や切断に必要なモデムあるいは PAD の制御, パケットの各種パラメータ設定 (データパス長, 伝送速度など), 相手コンピュータへのログイン制御も端末エミュレータ・プログラムに入ってから実行する. KERMIT では, パケット長, 誤り制御方法, パケットヘッダなど各種のパケットパラメータもまたユーザが選択指定できる.

信することを意味する。また、expect 1-send 1-expect 2-send 2-expect 3...では、expect 1 が受信できなかったときは send 1 を送信して expect 2 を待ち、expect 2 も受信できないときは send 2 を送信して expect 3 を待つ...といったシーケンス制御ができる。

ここで使用できる文字セットには ASCII 文字のほかに、" "(expect の場合、無条件を、send の場合はリターンを意味する)、\d (send に使われ、1秒のポーズ)、\r (リターン)、\n (改行)、\b (ブレイク)、\数字 (8進数字で示された ASCII 文字) が含まれている。send で使用される文字列の最後にはリターンが送信される。ただし、文字列が \c で終わっているときはリターンの送信は行わない。図-7 に、CDS 224, Hayes Smartmodem 1200, FACOM 1931 HA, Racal-Vadic VI 3400 J をモデムに使用して公衆電話網に接続した場合と専用線を使用した場合の L.sys ファイルの例を示した。ここで、相手ホスト名 sitel, ログイン名 usitel, パスワード sitelpassw, 電話番号 0-123-1234 (例では内線発信) とした。expect-send 制御は単にモデムやログイン制御だけでなく、たとえば公衆パケット交換網に接続する際に X.28 プロトコル⁶⁾の PAD パラメータ制御用コマンド信号をエミュレートしたりすることなどにも応用できる。KERMIT ではこの expect-send 制御を採用したもののほかに INPUT/OUTPUT 制御を採用したものが存在するが基本機能は同じである。

L-devices ファイルは、uucico と各通信機器 (モデムなど) とのインタフェースを記述する。そのフォー

- (1) CDS 224 または CDS V. 22 bis
`site1 Any DIR 1200 tty14 "" "\r\r\d AUTODIAL \r\d CDS DPW0W
1231234\r NUMBER \d\d\d\d DIALING \d\d\d\d TONE \d\d\d\d
DIALING \d\d\d\d DIALING \d\d\d\d INITIATING \r\d ogin:-\b\
r-ogin: usitel ssword: sitelpassw`
- (2) Hayes Smartmodem 1200
`site1 Any DIR 1200 tty14 "" "" "" ATS7=80\r OK ATS8=8\r OK
ATDP0,123-1234\r CONNECT "" ogin:-\b\r-ogin:-\b\r-ogin: usitel
ssword: sitelpassw`
- (3) FACOM 1931 HA
`site1 Any DIR 1200 tty14 "" CRN0>>1231234\015\012 ogin:-\b\
ogin:-\b\r-ogin: usitel ssword: sitelpassw`
- (4) Racal-Vadic VI3400J
`site1 Any DIR tty14 "" "\002\d0\d=\d1\d2\d3\d1\d2\d3\d4\d\074\d\017\
d\003\d ogin:-\b\r-ogin:-\b\r-ogin: usitel ssword: sitelpassw`
- (5) Hard-wired
`site1 Any DIR 1200 tty14 "" "" ogin:-\b\r-ogin:-\b\r-ogin: usitel
ssword: sitelpassw`

図-7 UNIX 4.2 BSD UUCP の L. sys による制御例

Type Line Calldev Class Brand

図-8 L-devices ファイルのフォーマット

- (1) dnll のとき
`ACU cu10 cua0 1200 dnll`
- (2) ventel のとき
`ACU tty14 unused 1200 ventel`
- (3) vadic のとき
`ACU tty14 unused 1200 vadic`
- (4) hayes のとき
`ACU tty14 unused 1200 hayes`
- (5) 専用線のとき
`DIR tty14 unused 1200 direct`

図-9 モデムを使用する場合の L-devices 例

マットを図-8 に示す。ここで、Type は各種通信機器への接続方法を記述する。指定できる組み合わせは、ACU (自動発信ユニット付き)、DIR (専用線などの直接接続の場合)、PNET (Purdue ネットワーク)、UNET (3Com UNET)、MICOM (Micom Mux port)、DK(AT&T の datakit)、Slave (Slave モードのみで使用する場合) に限る。

通信機器と接続して使用するデバイス名 (tty ドライバ名や cul ドライバ名) などは line で指定する。自動発信ユニット用に別の制御ラインを使用しなければならない場合には、そのために必要なデバイス名を calldev に記す。自動発信機能付きのモデムの場合または、使用しない場合は unused とする。回線速度は Class で指定する。

使用する通信機器のブランド名は brand で示す。登録されているブランド名には、dnll, ventel, vadic, hayes, micom などがある。使用しない場合は、direct と指定する。

UUCP システム中に、デバイス・ドライバが書き込まれている dnll, ventel, vadic, hayes などのモデムを使用する場合は、図-9 のように指定すればよいが、一般にこれらのモデムは日本では使用できないので、国産のモデムを使用する場合には、各ユーザがそのモデムに合ったデバイス・ドライバを書くか、L. sys の expect-send 機能を利用する。図-7 に CDS 224, Hayes Smartmodem 1200, FACOM 1931 HA, Racal-Vadic VI3400 J をモデムとした場合の expect-send による制御法を示したが、これに対応する L-devices ファイルの内容は、専用線の場合と同形式でよい。L.cmds

```
PATH=/bin:/usr/bin:/usr/ucb
rmail
rnews
ruusend
mail
uusend
unbatchnews
```

図-10 USENET 環境下での L.cmds の例

ファイルは相手ホストが uucico を使って実行できるコマンドのリストであり、そのためのディレクトリ・パスなどの情報を含む。機密保持のため、コマンド数、ディレクトリはなるべく少ないことが望まれる。USENET 環境下で必要最小限の L.cmds ファイルを図-10 に例示する。

5. データ変換

シリアルライン通信では、ASCII 伝送を基本にしているため、EBCDIC データを転送する場合にはいったん ASCII 文字に変換してから、パケットを作成しなければならない。また、テキスト・ファイルの伝送の場合、UNIX 以外のオペレーティング・システムで最初に開発された XMODEM や KERMIT では行区切りが CR、LF であるため、UNIX システムは、送信側のとき LF を CR と LF に、受信側のとき CR と LF を LF へ変換する前処理を行う必要がある。一方、UNIX 間通信を前提にして開発された UUCP にはこの種のデータ変換機能はない。

ファイル転送に関しては、UUCP や KERMIT はデータパス長が8ビットでも7ビットでもバイナリ・ファイルの伝送が可能であるが、XMODEM のデータパス長は8ビットであるため、シリアルラインが7ビットのパスしか許さない場合にはバイナリ・ファイルの伝送はできない欠点がある。

データ圧縮および暗号化に関しては、各システムでサポートされている既存プログラムを用いる。たとえば、UNIX では、データ圧縮に関して Adaptive Huffman 符号¹³⁾による compact や Lempel-Ziv 符号³⁰⁾による compress が、暗号化に関して Enigma による crypt が標準コマンドとして用意されている。USENET では大量の電子ニュースレターを送受する際、回線使用コスト削減のため compact や compress を活用している。

6. プログラム駆動法

XMODEM や一部の KERMIT では、相手ホストからファイル転送するには、ユーザが端末エミュ

レータ・プログラムを使って相手ホストへログインし、ファイル送信コマンドを打ち、ローカル・ホストへエスケープしてファイル受信コマンドを実行する。

これに対して UUCP の uucico は、遠隔実行コマンド uux やファイル転送コマンド uucp によって駆動したり、システム・デモンによって、決められた時刻に駆動したりできるように設計されているため、対話型のプログラムではない。ユーザが uucp や uux を実行すると、ローカル・ホスト上に、ファイル転送指示ファイルとデータファイルが作成される。ファイル転送指示ファイルは、ホストが相手ホストとの間で必要なファイルを送受するための作業指示メッセージを収容したファイル、データファイルは、uucp のときには相手ホストへ送信するファイルのコピー、uux のときにはコマンドファイルになる。ここで、コマンドファイルは、uux によって指定したコマンドを相手ホスト上で実行するための環境(コマンド、引き数、コマンド実行に必要なファイル名、標準入出力の指定、コマンド実行を要求したユーザ名など)を収容している。

Uucico プログラムは、相手ホストへ1ユーザとしてログインする。ただし、相手ホスト上には uucico がログインした場合のコマンドインタプリタを uucico とするようセットしてあるため、両ホストの uucico 間での相互通信が自動的に開始される。その後、uucico はファイル転送指示ファイルの内容に従ってファイル転送などの通信を実行する。USERFILE は、その際に uucico プログラムによって使用できるファイルのアクセス権を示し、図-11 のフォーマットを持つ。ここで、sitename、loginname および fileprefix (/usr/spool/uucppublic と通常指定する)は、それぞれ、各相手ホスト名、相手ホストの uucico プログラムに対して割り当てた UUCP ログイン名、および fileprefix ディレクトリ以下をアクセスできることを示す。相手ホスト名 sitel、ログイン名 usitel に対応する USERFILE を図-12 に例示する。ここで、第2行は、ローカル・ホスト上の全ユーザに対してすべてのアクセス権を与えることを示している。図-13 は uux によって相手ホスト上でコマンドを実行することを指示したときの処理の流れである。

```
loginname, sitename fileprefix1, fileprefix2,...
```

図-11 USERFILE のフォーマット

```
usitel, sitel /usr/spool/uucppublic
, /
```

図-12 USERFILE の設定例

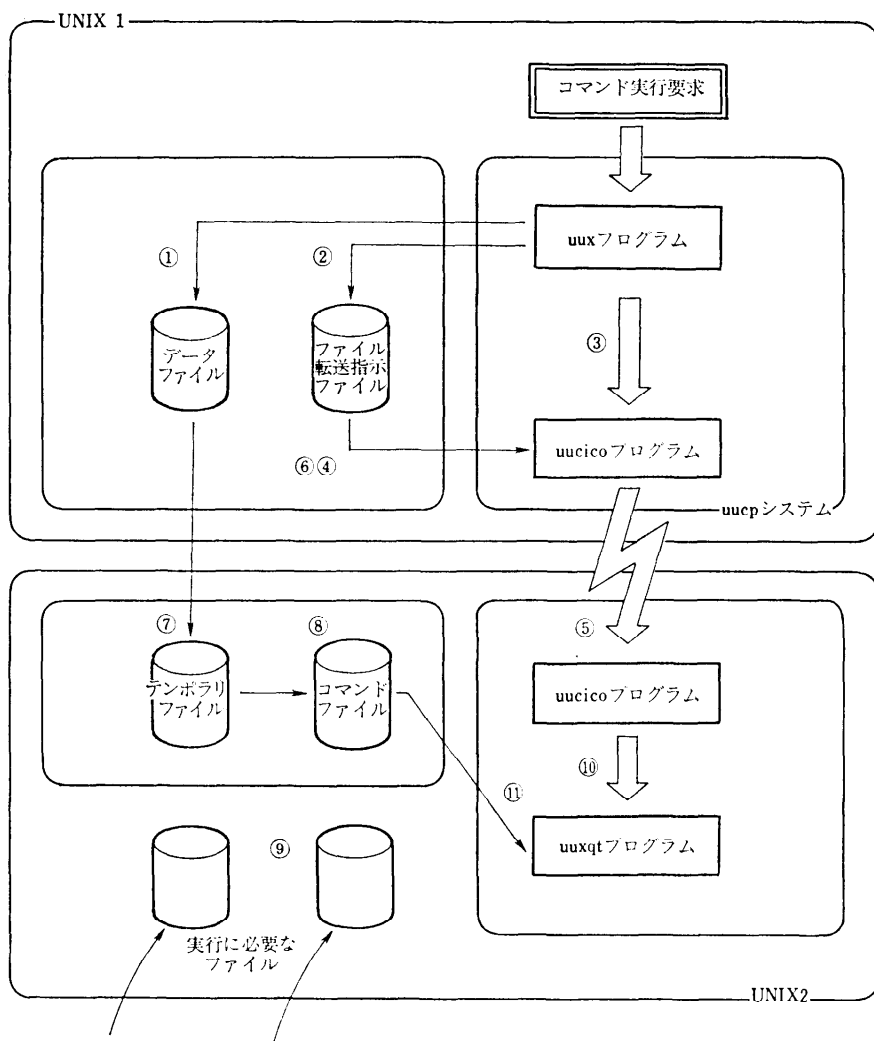


図-13 uux によって相手ホスト上でコマンド実行するときの処理の流れ (各段階の解説は付録Aに示す)

7. 経路指定方法

KERMIT や XMODEM と異なり、UUCP の uucico はシステムデモンによっても実行できるため、相手ホスト上でも uux を駆動することによって、直接接続していないホストとの通信も可能である。たとえば、図-14 のホスト A のユーザ a がホスト E のユーザ e に電子メールを送信するには、接続対情報 (A が B と C に、B が A、C と D に、D が B と E に接続している) を基にして、接続経路 (B!D!E!) を算出した

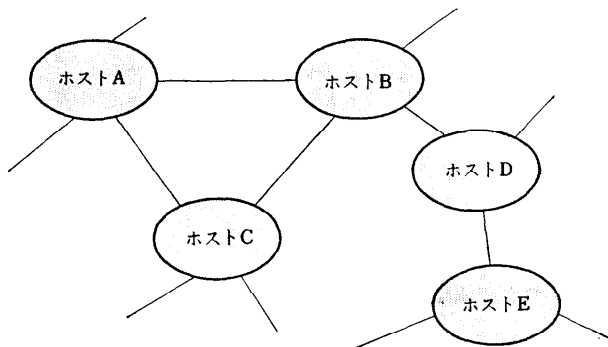


図-14 UUCP による通信網の接続構成例

後、電子メール・プログラムに指示する。プログラムは電子メールの内容をホスト B にコピーし、ホスト B 上で電子メール・プログラムを接続経路 D!E!e で遠隔実行する。ホスト B, D 上でこのプロセスを繰り返すことによって、電子メールがホスト E まで中継される。

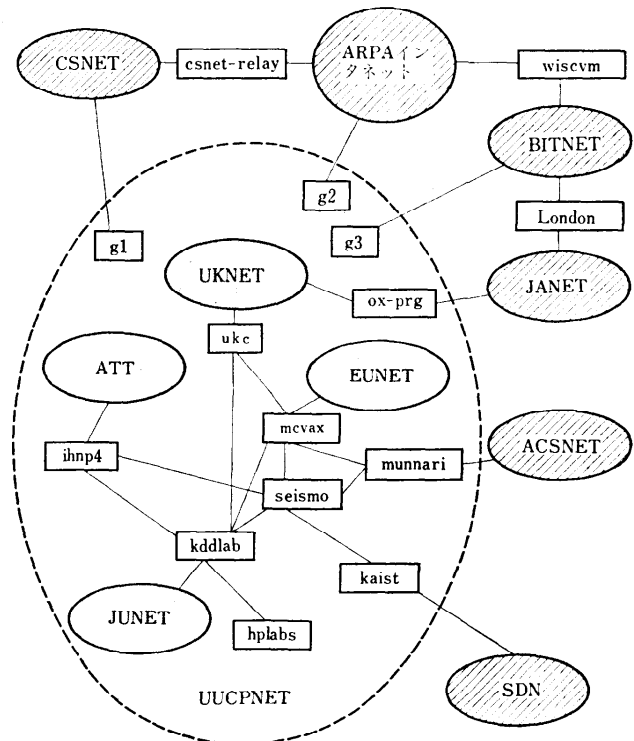
中継処理は、電子掲示板や電子ニュースレターシステムではさらに単純でよい。たとえば USENET のニュースレター・システム B news では、各ニュースレターは対象分野(たとえば人工知能)を示すニュース・グループをヘッダに含んで転送される。各ホストのニュースレターのサーバは、隣接ホストの要求するニュース・グループのリストを持っていて、ローカル・ホストのユーザの投函したニュースレターや遠隔ホストから中継されてきたニュースレターをローカル・ホスト上にコピーし、さらに、隣接ホストがそのニュースレターの属すニュース・グループの受信を希望している場合には、隣接ホストへ転送する。ニュースレターの重複は、投函時のホストで作成したメッセージ ID によって判別している。したがって、メッセージのルーティングは自動的になされ、ユーザが指定する必要はない。

8. USENET の発展

このような技術的背景から UUCP-NET や USENET が、研究開発用ネットワークとして発展してきた。UUCP-NET は UUCP プロトコルを用いて電子メールの送受を行うネットワークである。一方 USENET は、電子ニュースレター・システム netnews (現在のバージョンは B) を用いて不特定多数への通信である電子ニュースレターを送受できるネットワークである。USENET は特定の通信媒体用の通信網ではなく、公衆電話網、公衆データ網、各種 LAN、既存電子メール網を通信媒体にしている。電子メール網を利用する場合は、いくつかのニュースレターをまとめたものに電子メールのヘッダを付けて相手ホストへ送り、相手ホストでこの電子メールからニュースレターを分離、再構成する手順を用いている。図-15 に 1985 年 7 月現在

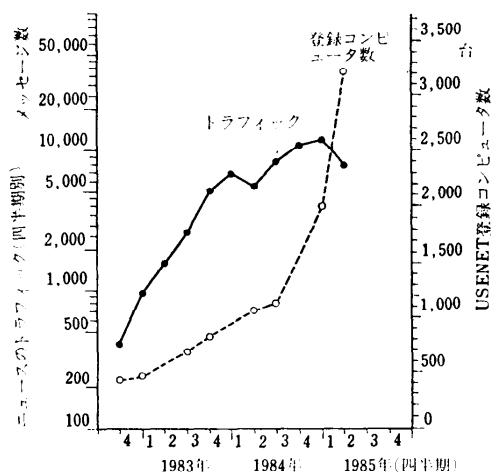
の USENET を構成する各種ネットワークの相互接続関係を示す。これらの電子メール網上で電子ニュースレターを送受できるホスト群を USENET と定義している。USENET の加入コンピュータやオペレーティング・システムには、UNIX 系のものが多いが、ほかのコンピュータ・システムも数多く加入している。(たとえば CP/M を用いて個人参加する場合もある。)

USENET は、当初北米を中心に発展し、その後ヨーロッパの研究諸機関が構築した EUNET、英国の研究諸機関が構築した UKNET や JANET、オーストラリアの研究諸機関が構築した ACSNET、韓国の SDN、および日本の JUNET^{21), 31), 19), 21)-23)}などのネットワークを相互接続することによってより大規模なネットワークとして統合された。その規模は現在も急速に拡



長方形はゲートウェイ、楕円は各ネットワークを表す。破線は UUCPNET (UUCP プロトコルによる電子メール網)であり、UKNET (United Kingdom Network), EUNET (European Unix Network), ATT (AT & T の UNIX 網), JUNET (Japan Unix Network) をサブネットワークに持つ。英国の JANET (Joint Academic Network), ACSNET (Australian Computer Science Network), 韓国の SDN (System Development Network), BITNET (IBM の OS VM/SP のネットワーク), CSNET, ARPA インタネットおよび UUCPNET 上で電子ニュースレター netnews を送受できるホスト群を USENET と定義する。

図-15 USENET の網接続構成



四半期を第一(1~3月),第二(4~6月),第三(7~9月),第四(10~12月)とし,トラフィックの単位をメッセージ数/3ヵ月としている。データは1982年11月~1985年6月までにオランダのCWI (mcvax),米国カリフォルニア大学サンディエゴ校(sdchema),韓国KAI ST (kaist),およびKDD研究所(kddlab)で受信したニュースを基に,各サイトで受信するニュースグループの種類の違いを考慮して推定した。また,登録コンピュータ数は,それぞれ3,6,9,12月時の台数である。

図-16 USENETの登録コンピュータ数とニュースレターのトラフィック(各段階の解説は付録Aに示す)

大しつづあり,1985年7月25日現在の正式加入コンピュータ数は,3196台である。図-16に加入者数と交換されている電子ニュースレターのトラフィックの経過を示す。

図-15の各電子メール網間でのメールの中継処理に関しては,ユーザによって投函されたり,隣接ホストから中継されてきた電子メールは,UNIX 4.2 BSDでは,異種電子メール網間ゲートウェイ・プログラム sendmail¹⁾によってアドレスを分析された後,適当な電子メール網に渡されるので,ARPA インタネットなどの電子メール網に属した利用者とUUCPNETの利用者との間での通信も原理的に可能である。sendmailは電子メールのヘッダの変換規則とそのインタプリタからなる一種のプロダクション・システムである。

UUCPNETでは,ユーザが電子メールの送信をするには,ホスト間の接続状況を把握し,通信相手までのすべての中継ホストを指定するというソース・ルーティングを行わなければならない。このようなルーティングを各ユーザが行うのは,煩雑であるため,ヨーロッパでは特定のホスト(バックボーン・ノード)を

表-1 インタネット・アドレスとソースルーティング

宛先	インタネット・アドレス	ソース・ルーティングの場合のアドレス
一般のUUCPNETユーザ	user@site.UUCP	ihnp4!site!site2!...!site!user
UUNETのユーザ	user@site.UK	ukc!user@site
ACSNETのユーザ	user@site.AU	mcvax!munari!user:site

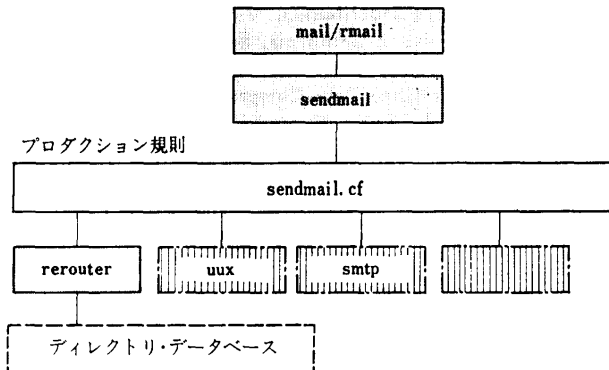
決め,そこにルーティングを任せざることを早くから行ってきた。図-17にヨーロッパのバックボーン・ノードで電子メールのルーティングに利用されているrerouter⁴⁾の構造とアルゴリズムを示す。rerouterは,接続経路が不明のとき sendmail のアドレス分析規則によって呼ばれ,あらかじめ電子ニュースレターなどから得た接続情報を基に作成しておいたディレクトリ・データベースを用いて最適な接続経路を生成する。rerouterでは,ARPA インタネットの電子メール規格 RFC822⁵⁾とUUCPなどの既存のネットワークが採用しているアドレスとの共存が図られていて,表-1のいずれのアドレスをも使用することができる。RFC 822形式のインタネット・アドレスの場合には,それから最適経路を生成してくれる。

接続情報の交換は,UUCPNET 内部だけでなく,一般に複数のネットワーク間における通信時には不可避の問題になる。

9. ま と め

シリアルラインによるコンピュータ通信を行うための技術的手法を実際のプログラムを例にして説明し,それらの技術を基礎にアプリケーション・プログラムとして電子メールや電子ニュースレター・システムを構成する手法を解説した。パソコン推奨通信方式の規格が定まり,標準化が進行中であえて単純なシリアルラインを用いたプロトコルを取り上げたのは,このように単純なプロトコル上でも,使い易次第で,十分実用に供することのできるネットワークを構成できることを示すためであり,その成功した利用事例としてUUCPNETとUSENETを挙げた。

公衆電話網を対象にした郵政省のパソコン推奨通信方式は,モデムとしてCCITTのV.27terモデム(4800/2400 bps 半二重モデム)を採用した,ISOのOSIモデルに基づくプロトコルである。このような標準通信方式は将来多くのユーザと通信できるようになる利点がある反面,専用ハードウェアを必要とするた



uux, smtp は各電子メール網へのインタフェース。また rerouter のアルゴリズムは次の通りである。ここで DOMAIN は各電子メール網のドメイン名である。

```

if 受信アドレス=host1!host2!...!hostn!user
then
  if host1 が隣接ホスト
  then call uux
  else call rerouter (hostn までの経路をディレクトリ・データベースから生成する)
else
  if 受信アドレス=hostn!user または user@hostn.DOMAIN
  then call rerouter
  else
    if 受信アドレス=user
    then call local mailer
    else fail
  
```

図-17 rerouter のアルゴリズムとプログラム構造

め、全世界的規模での普及にはかなりの時間を要する。そこで早急に通信を希望する利用者を中心に簡易型通信システムへの需要も多い。特に、ARPA インタネットのように、この種の標準通信方式に全く興味を持たない電子コミュニティに対して日本の情報処理技術者が情報交換を希望する場合には、標準通信方式のためのソフトウェアおよびハードウェア上の投資を彼らがすることは望めず、シリアルラインを用いた簡易型通信システムに頼らざるをえない。

当初小規模システムとして設計した UUCPNET もユーザの増加に伴って、ルーティング方法に深刻な問題が生じ、各種のインテリジェント・ルータが各ホストで開発されていることを rerouter を例に説明した。現在 UUCPNET を、ARPA インタネットのドメイン・アドレッシング²⁷⁾に基づく階層構造のネットワークに移行させようというプロジェクトが進行している¹⁵⁾のは、このような技術的背景による。

一方、我が国は、この種の電子コミュニティの形成に関して遅れたため、過去の遺産によらず自由にネットワーク設計できる立場にあるといえる。パソコン推

奨通信方式も、アダプタとのインタフェースを考えると、ここでいうシリアルラインを用いた通信方式の一種であり、その意味で従来方式との融合、あるいは従来方式を発展的に解消できる技術として大いに期待できる。電子ニュースレターのような放送型メッセージを効率良く伝送するための通信方式（通信媒体、プロトコル）や、より高度なサービス（たとえば、ARPA インタネットの FTP のようなファイルの共有）を実現するための有効な通信方式に関しても、このような観点から検討を加える必要があろう。

参 考 文 献

- 1) Allman, E.: Sendmail—An Interconnecting Mail Router, Version 4.2, UNIX PROGRAMMAR'S MANUAL, 4.2, Berkeley Software Distribution, Vol. 2c, Virtual VAX-11 Version, Univ. of California, Berkeley (Aug. 1983).
- 2) 浅見他: USENET への国際リンク, 情報処理学会第31回全国大会, 9Q-4 (1985).
- 3) 浅見他: 電子メール網におけるアドレス情報管理システム, 昭和60年度電子通信学会総会全国大会, 1704 (1985).
- 4) Beertema, P.: Router, EUUG Distribution 3, Release 3b, EUUG (1984).
- 5) CCITT: 1200 Bits Per Second Duplex Modem Standardized for Use on the General Switched Telephone Network and on Leased Circuits, Vol. III, Fascicle III. 1, Rec. V. 22 (1981).
- 6) CCITT: DTE/DCE Interface for a Start-Stop Mode Data Terminal Equipment Accessing the Packet Assembly/Disassembly Facility (PAD) in a Public Data Network Situated in the Same Country, Vol. III, Fascicle III. 2, Rec. X. 28 (1981).
- 7) Comer, D.: The Computer Science Network, CSNET, Comm. ACM, Vol. 26, No. 10, pp. 747-753 (1983).
- 8) Crocker, D.: Standard for the Format of ARPA Internet Text Messages, RFC 822, Dept. of Electrical Engineering, Univ. of Delaware (1982).
- 9) Da Cruz, F. et al.: Kermit: A Filetransfer Protocol for Universities, Part 1, Byte, Vol. 9, No. 6, pp. 255-278 (1984).
- 10) Da Cruz, F. et al.: Kermit: A Filetransfer Protocol for Universities, Part 2, Byte, Vol.

- 9, No. 7, pp. 143-403 (1984).
- 11) EIA : Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, EIA RS-232-C, EIA (1969).
- 12) Emerson, S.L. : USENET, A Bulletin Board for Unix Users, Byte, Vol. 8, No. 10, pp. 219-236 (1983).
- 13) Gallager, R.G. : Variations on a Theme of Huffman, IEEE Trans. on Information Theory Vol. IT-24, No. 6, pp. 668-674 (Nov. 1978).
- 14) Hinden, R., Haverly, J. and Sheltzer, A. : The DARPA Internet : Interconnecting Heterogeneous Computer Networks with Gateways, IEEE Computer, Vol. 16, No. 9, pp. 39-48 (1983).
- 15) Horton, M. : UUCP Mail Transmission Format Standard, Winter '85 USENIX Conference (1985).
- 16) Johnson, L. : MacTerminal, Apple Computer, Inc. (1984).
- 17) Jordan, L. : The Mac Connects, MacWorld, July/August, pp. 64-69 (1984).
- 18) Leffler, S.J., Joy, W.N. and Mckusick, M. K. : TIP, UNIX PROGRAMMAR'S MANUAL, 4.2 Berkeley Software Distribution, Virtual VAX-11 Version, Univ. of California, Berkeley (Aug. 1983).
- 19) 松方他 : JUNET のソフトウェア, 情報処理学会第 31 回全国大会, 9 Q-2 (1985).
- 20) Mckie, J. and Beertema, P. : UUCP Release 3, EUUG Distribution 3, Release 3b, EUUG (1984).
- 21) Murai, J. and Asami, T. : A Network for Research and Development Communities in Japan -JUNET-, Proceedings of PCCS, pp. 579-588 (Oct. 1985).
- 22) 村井他 : 計算機研究者用 ネットワーク JUNET, 情報処理学会第 31 回全国大会, 9 Q-1(1985).
- 23) 村井他 : JUNET の名前管理, 情報処理学会第 31 回全国大会, 9 Q-3 (1985).
- 24) パーソナル・データ通信用プロトコル, 日経バイト, August, pp. 16-17 (Aug. 1984).
- 25) Nowitz, D.A. : Uucp Implementation Description, UNIX PROGRAMMER'S MANUAL, Seventh Edition, Vol. 2 B (Oct. 1978).
- 26) パーソナル・コンピュータ通信に関する調査研究会 : パソコン・ネットワーク規格書, アスキー (1984).
- 27) Postel, J. and Reynolds, J. : Domain Requirements, RFC 920, USC/Information Science Institute (Oct. 1984).
- 28) Tanenbaum, A.S. : Computer Networks, Prentice-Hall (1981).
- 29) UUCP Source Program, 4.2 Berkeley Soft-

- ware Distribution, Virtual VAX-11 Version, Univ. of California, Berkeley (Aug. 1983).
- 30) Welch, T.A. : A Technique for High Performance Data Compression, IEEE Computer, Vol. 17, No. 6, pp. 8-19 (June 1984).

(昭和 60 年 9 月 6 日受付)

付録 A UUX による遠隔コマンド実行

(図-13) の解説

(1) UNIX 上の uux プログラムは, コマンド実行環境およびコマンドストリングを収容するコマンドファイルを, データファイルとして作成する。

(2) uux プログラムは, 実行システムである相手ホスト UNIX 2 へファイルを転送するために, ファイル転送指示ファイルを作成し, ファイル転送メッセージを書き込む。

(3) uux プログラムは uucico プログラムを起動する。

(4) ファイル転送指示ファイルを得る。

(5) UNIX 1 上の uucico プログラムは UNIX 2 上の uucico プログラムを起動する。

(6) UNIX 1 上の uucico プログラムは, ファイル転送指示ファイルからファイル転送のための作業指示メッセージを得, UNIX 2 上の uucico プログラムに送る。

(7) UNIX 1 上の uucico プログラムはデータファイル (コマンドファイル) を UNIX 2 上のテンポラリファイルに転送する。

(8) UNIX 2 上の uucico プログラムは, テンポラリファイルをコマンドファイルにコピーする。

(9) コマンドの実行に必要なファイルをすべて準備する。(uux プログラムでファイル転送指示ファイルを作成し, uucico プログラムでファイルを集める。)

(10) UNIX 2 上の uucico プログラムは, uuxqt プログラムを起動する。

(11) uuxqt プログラムは, コマンドファイルの指示に従ってコマンドを実行する。

付録 B UUCP 接続実験の手順

最小構成の設備で, UUCP 接続実験を行う手順を示す。以下では, 情報処理学会のホスト計算機 ipsja へ, 山田太郎氏のパーソナル計算機 txypc から UUCP 接続するものとして説明する。

(1) 必要設備

音響カプラ (300 bps), 直通電話 AT&T 社の

SYSTEM III UNIX システム, パーソナル計算機を用意する。

(2) 接続許可

ホスト計算機 ipsja の管理者にログイン名 uutxypc, パスワード weneedyou で, パーソナル計算機 txypc, から UUCP 接続実験を行う許可をもらう。

(3) inittab の設定

/etc/inittab.cu の内容を /etc/inittab へコピーする。ログアウト後, コマンド init 2 を実行し, マルチユーザモードとする。

(4) 記述ファイルの設定

/usr/lib/uucp の次のファイルの内容を, それぞれ以下のように設定し, 所有者を uucp に戻しておく。

i) SYSTEMNAME

txypc

ii) L-devices

DIR cul0 0 300

iii) L. SYS への 1 行追加

ipsja Any cul0 300 cul0 ogin:--ogin:
uutxypc sword:--sword:weneedyou

iv) USERFILE への 1 行追加

uutxypc, ipsja /usr/yamada

(5) 手動ログインテスト

直通電話と音響カプラを用い, ホスト計算機 ipsja を呼び出す。

コマンド cu -l /dev/cul0 dir により, ホスト計算機と遠隔接続する。

手操作で, ログイン名とパスワードを試し, ログイ

ンできるかテストする。Shere と出たら成功となる。遠隔接続を終了し, 直通電話を切る。

(6) コマンドテスト

コマンド

uucp myfile ipsja\! /usr/spool/uucppublic

コマンド

mail ipsja\! sakamoto

あるいはコマンド

mail ipsja\! txypc\! yamada

をシェルから実行する。

(7) 一時停止

/usr/spool/uucp に移動し, (6) のコマンドの実行に対応したファイル群が作られていることを確認し, ファイル STST.* や LCK.* を除去する。

(8) 手動ダイヤル

直通電話と音響カプラを用い, ホスト計算機 ipsja を呼び出す。

(9) 実行再開

コマンド /usr/lib/uucico -rl -sipsja -x4 を実行する。ファイル /usr/spool/uucp/LOGFILE の末尾の記述を見て, 無事転送が行われたかどうかを確認する。

以上の手順が完全に (9) まで終了すれば, 接続実験は終了である。以後の運営においては, 手順 (6) ~ (9) を行えばよい。手順 (6) で自分あての電子メールを送っておけば, しばらく後の手順 (9) の実行の際に, 自分あての電子メールが ipsja から txypc へ届くことになる。