

解説

— コンピュータ・ネットワーク その 2 —

HOST-HOST プロトコルおよびハイレベル・プロトコルの動向*

伊藤 哲史**

1. はじめに

Host-Host プロトコルおよびハイレベル・プロトコルの重要性が認識され、実際に規約として作り上げられたのは、1968年にアメリカ国防省のARPAがスポンサになって開発されたARPAネットワークにおいてであった。

このコンピュータ・ネットワークが予想を上回る規模にまで発展し、その成功が伝えられた頃から、世界各地において独自のコンピュータ・ネットワークが開発されてきた。これらの独立して開発されたネットワークはARPAとALOHAあるいはCycladesとNPLのように相互に結合される方向へと発展していった。

このように独立に開発されてきたコンピュータ・ネットワークは必然的に異なるプロトコルを持っていた。このために相互の結合においては、ハードウェア結合というよりはむしろプロトコルの変換というソフトウェア的な結合に多くの労力を必要とした。

コンピュータ・ネットワークの有効性は、その規模に比例することが初期の研究において述べられているが、実用レベルのネットワークは近い将来において相互結合をして規模が大きくなってゆくであろうことが明らかになってきている。この傾向は、それぞれのレベルのプロトコルを標準化するという原動力となっている。

コンピュータ・ネットワークは、データ交換網とHostコンピュータより構成されている。それぞれの系は課せられた役割を果たすために種々のプロトコルを持っている。

Hostコンピュータにおけるプロトコルは、図-1に示すように階層構造を持っており、次の3つのレベル

からなっている。

- (1) ネットワーク・プロトコル
- (2) Host-Host プロトコル
- (3) ハイレベル・プロトコル

ネットワーク・プロトコルおよびデータ交換網のソフトウェア/ハードウェア・インタフェースに関しては、CCITTの勧告案であるX.25等がこれにあたるため、今後はこの案がこのプロトコルの主流となっていくであろう。

Host-Hostプロトコル、ハイレベル・プロトコルに関しては、ネットワーク・プロトコルにおけるX.25のような役割を果たすものはいまだに作成されていない。本稿では、Host-Hostプロトコル、ハイレベル・プロトコルに関して、標準化あるいは共通化にあたっての論点になるとと思われる項目について次の順序で述べる。

2.においては、ISOなどの標準化機関でこれらの問題がどのように取り扱われているかを述べる。

3.においては、Host-Hostプロトコルの標準化への

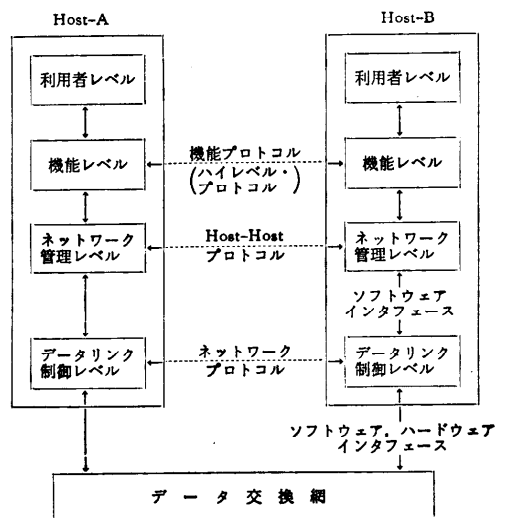


図-1 プロトコルとインタフェース

* The trend of Host-Host Protocol and High-Level Protocols by Tetsufumi ITO (Japan Information Processing Development Center)

** (財)日本情報処理開発協会

過程の成果として国際情報処理学会 (IFIP) のワーキング・グループがまとめた案を紹介する。

4. においては、会話形端末アクセスプロトコル (IT-P) およびファイル転送プロトコル (FTP) などのハイレベル・プロトコルに関し、標準化を考える場合に問題点となると思われる個所について述べる。

2. 標準化への動向

2.1 I S O

Host-Host プロトコルおよびハイレベル・プロトコルに関する動きは、N 1167¹⁾、N 1173²⁾、N 1249³⁾ の3つの文書によって推測できる。

N 1167 は英国の提案であって概要は次のとおりである。

データ交換網の開発にともなってネットワークに結合された種々の端末間の End-to-End で働くハイレベル・プロトコルの標準形の必要性がでてきた。ハイレベル・プロトコルはシステムの利用者が使用できるアプリケーションにもとづく機能をサービスするものである。これに該当するものとしては、ファイル転送、会話形端末利用、リモート・ジョブ・エントリなどのプロトコルがあるが、グラフィックあるいはデータベースに関するものも考えられている。

これらのプロトコルは通信サブシステムの機能を利用する方向で定義されるべきことを述べ、結論として次のようなことを要望している。

- (1) 共通のハイレベル・プロトコルに向って作業を始めるという利用者の要求は十分に理解できる。
- (2) この作業はいくつかのネットワーク関係者の協同作業としてなされるのが最上である。この方法はネットワークに独立なものを生むという利点を持っている。
- (3) この作業計画についての長期的なかかわりを満たすために、標準に対する討論や開発は ISO のような機構の下におこなわれるべきである。

N 1173 は ECMA の提案であって概要は次のとおりである。

コンピュータ・ネットワークにおいては、基本通信プロトコル、ネットワーク・コールの確立プロトコル、機能レベルのプロトコルなどが必要である。ISO に関する限り、すべての作業はデータリンク制御プロトコルや関連する通信インタフェースの規定に集中してきた。ISO が次のレベル、特にパケットあるいは回線変

換の両者の公衆データ網に必要とされるフロー制御および呼制御に関するものの研究を開始しなければならないことが明らかになった。このことから、ISO はネットワーク・アーキテクチャおよびその構成、階層、プロトコルとインタフェースの規定に対して責任を持つのがよいと思われる。

N 1249 は 1975 年 10 月にワシントンで開催された第 13 回 ISO/TC 97/SC 6 の会議のレポートの一部であって End-to-End プロトコルに関して次の決議がなされたことを述べている。

N 1167, 1173 の文書は交換網によって結合された DTE (Data Terminal Equipment) 間での End-to-End プロトコルがないために、データ交換のアプリケーション、特に公衆データ網の利用が妨げられていることを述べている。この新しい分野の研究は初期的な段階にあることが確認された。それ故に、SC 6 に新しいプロジェクトが作られること、将来の研究計画の作成や標準作成のためにとられるべき行動について、よりよい定義ができるように委員を任命することが同意された。

ワシントン会議以降、各国から End-to-End あるいはハイレベル・プロトコルに関する文書が寄せられている。これらの意見にもとづいて標準化のための作業を進めて行くのが今後の作業となるであろう。

2.2 I N W G

INWG は、国際情報処理学会のワーキング・グループ 6.1: Packet Switched Network Interworking として、いままでコンピュータ・ネットワークに関してパケット交換網からハイレベル・プロトコルにいたるまでの広い範囲にわたって意見の交換、標準化案などの作成をおこなっているグループである²²⁾。

INWG はパケット・フォーマット、Host-Host プロトコルの標準化案を作り上げてきた。これらの案は CCITT, ISO に提案されている。この中で注目する必要があるのは Host-Host プロトコルの標準化案⁴⁾である。この案は Cerf (SRI), McKenzie (BBN), Scantlebury (NPL), Zimmermann (IRIA) の 4 人によってまとめられており、それぞれ ARPA, NPL, Cyclades の関係者であることから現状における実際の技術が生かされている。この案の賛否の投票が INWG のメンバーによっておこなわれたが、その結果は次のとおりである。なお、投票の結果が整数でないのは、同一機関において賛否両側の意見があったときは、賛成 0.5、反対 0.5 のような投票が許されているためである。

賛成	25.8 票
反対	7.5 票
棄権	8.7 票

このように全体のコンセンサスはとれていないにしても、現状においては最も進んだ内容を持った Host-Host プロトコルである。

3. INWG の End-to-End プロトコル

3.1 概要

このプロトコルは、データ交換網からみたときに一つの実体として認識できるリソースの集合を仮想ホスト (Virtual Host) としてとらえ、仮想ホスト中のプロセス間でのデータ交換を実現する機能を持っている。仮想ホストは Transport Station (TS) を持っており、TS がこのプロトコルの処理を担当する。

プロセス間通信のためにはプロセスを識別するための名前空間が必要となる。これは図-2 に示すように TS 名、加入者名、番号の 6 オクテットで構成されるポートが使用される。

プロセス間通信は、発信ポートから受信ポートに向けてレターを送ることによっておこなわれる。ポートというのは電話でいえば電話番号にあたるもので、電話番号が大代表を持っているのと同じように、1つのポートで同時に複数個のプロセスと交信することもできるように設定されている。

プロセス間通信は、2つのモードで動作することができる。1つはレター・グラム・モードであり、もう1つは結合 (liaison) モードである。

レター・グラム・モードは、郵便システムの手紙の配達にたとえることができるもので、データ交換網がデータ・グラムのサービスをおこなうことを想定し、これをプロセス間通信のレベルでサービスしようとするものである。

結合モードは、バーチャル・コールに相当するものであって、レターの伝送に先だって初期設定コマンドが交換される。初期設定では次のことが確認され、同意が得られれば双方向のレターの伝送が可能となる。

- (1) 結合の両端のプロセスがアクティブであることの確認。
- (2) オペレーションにはいるサービス種類の同意。
- (3) 付随したパラメータの初期化。

結合モードにおいては、エラー制御、フロー制御などの付加サービスを受けることもできる。

0	1	2	3	4	5	オクテット
発信地あるいは目的地 TS 名				加入者名	番号	

図-2 ポート・アドレス

レターの最大長は 27648 オクテットであって、最後のものを除いて 216 オクテットの固定長のフラグメントと呼ばれるものに分割されて送られる。このためにレターからフラグメントへの分割 (フラグメンテーション) とフラグメントを編集してレターに組み立てる操作 (リアセンブル) は TS の役割となる。

このプロトコルは、図-3(次頁参照) で示される 10 種類のコマンドによって TS 間の会話がおこなわれ、上記の機能が達成される。TS におけるそれぞれの機能関係は図-4 のとうりである。

3.2 問題点

INWG 案は完成してしまったものではなく、将来のトピックスをドキュメント中で述べてある。これは次の点である。

- (1) LI-PURG と、それ以前に送信されプロセスにおいて未処理のレターの関係。
- (2) LI-NACK の利用方法。
- (3) Ack をフラグメントに関して送るか、レターに関して送るかという問題。
- (4) ソフトウェアで計算が簡単なチェックサムの方式について。
- (5) レター・グラムとそれに関するコマンドの利用について。

このような問題点はあるが技術的にはよくまとまっ

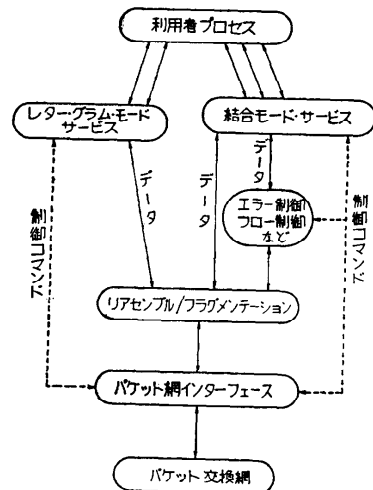


図 4 INWG-End-to-End プロトコルの機能関係

ビット ニモーニック	0	1	2	3	4	5	6	7	意味
LG-LT	R	0	0	1	0	0	0	0	レター・グラム・モードにおいて、レターのフラグメントを送るために使用される。
LG-ACK	0	0	0	1	0	0	0	1	レター・グラム・モードにおいて、レターの Ack を送るために使用される。
LG-NACK	0	0	0	1	0	0	1	0	レター・グラム・モードにおいて、否定の Ack を送るために使用される。
LI-LT	R	0	0	0	0	0	0	0	結合モードにおいて、レターのフラグメント及び逆方向の Ack を送るために使用される。
LI-ACK	R	0	0	0	0	0	0	1	結合モードにおいて、Ack を送るために使用される。
LI-INIT	0	0	0	0	0	0	1	1	結合モードの初期設定に使用される。
LI-TERM	0	0	0	0	0	1	0	0	結合モードを終了するために使用される。
LI-PURG	R	0	0	0	0	1	0	1	結合モードにおいて割り込みを送り、残っているレターの消去のために使用される。
LI-ERR	0	0	0	0	0	1	1	0	結合モードにおいて、エラー信号を送るために使用される。
LI-NACK	R	0	0	0	0	0	1	0	結合モードにおいて、否定の Ack を送るために使用される。

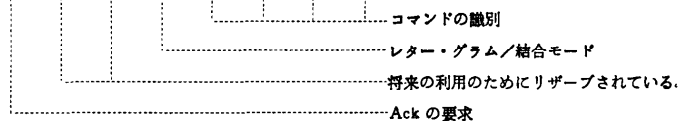


図-3 Transport コマンドの Op-code

たプロトコルであるということができる。

しかしながらこのプロトコルは、パケット網がデータ・グラム・サービスをおこなうことを前提として作られている。X.25 の機能をみるかぎり、現在のところこの前提は公衆パケット交換網に関してはくずれたということができる。このために図-4 におけるパケット網インタフェースの部分を更新したプロトコルとして独立させる方向も考えなければならない。

4. ハイレベル・プロトコルの構造

ハイレベル・プロトコルの重要性に関しては認識されているが、議論の対象となるような共通の原案はまだ提案されていない。このために、ARPA-net⁹⁾⁻⁸⁾, Cyclades⁹⁾⁻¹¹⁾, NPL¹⁶⁾, JIPNET^{17), 18)}, EPSS Liaison Group^{12), 13)}, N 1²¹⁾, EIN^{14), 15)} のようなコンピュータ・ネットワークのハイレベル・プロトコルにおいてサポートされている機能を調べ、持つべき機能および形態、問題となる機能について述べることにする。

ハイレベル・プロトコルの規定において最も重要なことは、そのプロトコルが単に遠隔地にある2つのプロセス間でのトランスペアレントなデータの授受をサービスするものであるか、ネットワークにおける標準化というものをどこまでとり込んだものであるかとい

う点である。利用者にとっては後者の方が望ましいことは明らかである。データ・ベースの共有をめざすようなハイレベル・プロトコルの中でも非常にハイレベルなものは、ネットワークにおける標準化の追求なくしては達成できないであろうし、他のプロトコルもこの方向に向かって進んでゆくものと考えられる。

しかしながら現状においては、ホストの管理、サービス形態がバラエティに豊んでいるために、標準化はきわめてむづかしい状態になっている。

このために、現在のハイレベル・プロトコルでは利用度が高く、規定が比較的簡単な範囲に限って決められ、プログラムが作成されているとよい。

ここでは会話形端末プロトコル、ファイル転送プロトコルの機能について概説する。

4.1 会話形端末アクセスプロトコル

このプロトコルは、遠隔端末からタイムシェアリングシステム、オンラインシステムなどをアクセスするためのものであり、ARPA-net の実績、Euro-net の利用者予測を見れば明らかなように、遠隔端末からの資源の共有がコンピュータ・ネットワークにおける主たる手段である。この意味において ITP は最も重要なプロトコルであり、各コンピュータ・ネットワークにおいて最も整った形をしている。

ITP は2つの基本機能より構成される。

まず第1は、初期会話および終了会話手順にあたるもので、サービスを要求する側(ユーザ)とサービスをする側(サーバ)との間でのサービス開始の確認をおこなう手順、終了をする手順を決めたものである。

第2番目は、仮想端末 (Network Virtual Terminal: NVT)といわれるものでネットワークにおける仮想的な標準の端末機能を定義したものである。

ITPはこの2つの機能を利用することによって、仮想端末から特定のサービスを受ける、あるいは仮想端末間での通信を可能にしている。このプロトコルは端末レベルの標準化をおこなっており、TSSなどの利用に当たってのコマンド体系の標準化をめざしたのではない。このために、利用者はサービスを受けたいすべてのホストの機能についての知識が必要である。

4.1.1 初期会話, 終了会話手順

これは Initial Connection Protocol (ICP) とも呼ばれているものである。

一般的に言って終了会話手順は、使用したデータリンクの切断順序がサーバとユーザの間で異なっていると、デッドロックが発生する可能性があるので切断順序を決めるだけで十分であろう。

初期会話手順では、リンクの設定順序ばかりでなく、どのようにしてユーザの要求するサービスにつなげるかという要素を持っているために、大きくわけて次の2通りの方法のいずれかがとられている。

第1番目は、それぞれのサービスに対して特別なポートを割り当てておき、ユーザからサーバのポートに対してコネクションの確立要求をおこなう。サーバはこれによって相手側の要求を知ってコネクション確立の応答をし、サービスを開始する。これは JIPNET で利用しているものであり、ARPA-net のそれも手順はかなり複雑になっているが基本的には同様の方式である。

第2番目は、ユーザ側からのサービス要求を受け付けるポートを1個だけあるいはサービス・グループごとに1個ずつ用意しておく。ユーザはこのポートに対してコネクションを確立するなどして、サービスの要求をおこない、その後、受けたいサービスの内容についての指示をおこなう。サーバ側は、相手の要求するサービス内容を知った後に、そのサービスが可能なプロセスとのコネクションを確立させるものである。

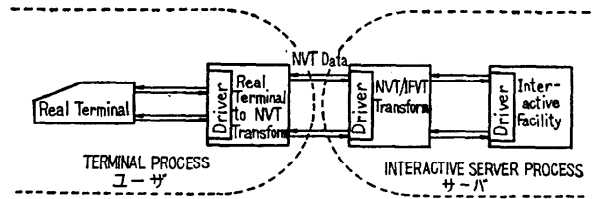


図-5 サーバとユーザ ITP の関係

これは Cyclades, NPL で利用されている。

この2つの方法を検討する上で考慮しなければならないのは、加入者が使用できるポートの数、プロトコルの種類、1つのプロトコルでサービスする種類(たとえば ITP で TSS, リモート・バッチの2つのサービス)、INWG の End-to-End プロトコルにおけるようなレター・グラム・サービスの有無などである。

第1番目の方法は非常に単純であるのでプロトコルの種類が少ないときには有利である。しかしながらプロトコルの種類が多い場合あるいは1つのプロトコルで多くの種類をサービスする場合などになると第2番目の方法が有利になってくる。

4.1.2 仮想端末

オンラインあるいは TSS において、各ホスト・コンピュータでは手順の異なったクラスの端末について、それぞれの手順ごとにターミナル制御プログラムを用意している。ローカルな TSS においてはこの手順の種類は数種であるが、コンピュータ・ネットワーク全体で考えると非常に多くの種類の端末になるので、すべての端末の制御をホストでおこなうのは不可能になってくる。このような背景で、サーバとユーザの間でネットワーク標準端末という既念が導入された。これが仮想端末であり、図-5 のようにサーバはユーザを仮想端末とみなして制御し、実際の端末への対応づけはユーザにおいておこなわれる。

仮想端末は、入力用としてキーボード、出力用としてプリンタ機能を持つタイプライタ的な装置を想定している。これを設定するとき考慮するのは次の点である。

a. 基本機能と拡張機能

仮想端末はタイプライタの如きものが想定されているが、実際の端末は、タイプライタ、キャラクタ・ディスプレイ、グラフィック・ディスプレイなど非常に広い範囲にわたっており一意的に機能設定するのが困難である。このために、最小機能としてすべてのホストにおいてサービスすることを原則とする基本機能

と、より良いサービスをするために、ユーザとサーバ間での合意ができれば一部の機能を追加、変更できるオプションより構成される。

オプションの選択に関しては、サーバとユーザ間での会話が必要になってくる。この会話方式としては、初期設定手順が終了した段階でのみおこなうものと、任意の時点で必要な会話をおこなうものがある。後者の場合には、非同期的におこなわれるので、要求と拒否、承諾のコマンドの機能設定に対する考慮、同じコマンドを2度以上だしてはいけないなどの制限をつけないとデッドロック、要求のループにはいる可能性がある。

b. コード系

仮想端末としては、国際的な標準コードに準拠するコードを使用することによって標準化している。これは IA5, USASCII, JIS などのコードを利用することを意味している。

このような標準コードがあるとはいっても、次のような点についての問題点が残されている。

まず第1番目は、CR (復帰), LF (改行), NL (復帰改行) の問題である。すなわち NL 機能しか持たない端末に対して、CR, LF を独立した機能としてサービスするように要求するかどうかということである。

一般的な利用方法の場合には、CR, LF は全く独立に動作させることはあまりないが、パスワードを黒くぬりつぶしたり、FORTRAN の 1H+ の機能をサービスするとき利用される。

この問題に関して、ARPA-net においては、NL 動作をおこなわせる場合には CR, LF の順で送り、それぞれを独立に使用するときには、CR の後に NULL コードをいれることを要求している。しかしながら、これを本気でしようとするサーバ側に非常に大きな負担がかかることがあるために、あまり良い解決方法とはいえない。

JIPNET のように、NL 端末のときには、原則として LF を NL とみなすようにして、それ以上に関してはプロトコルでは決めていない例もある。

c. 割り込み信号と同期の問題

無限ループに入ってしまったプログラムの強制終了、大量出力を始めたプログラムの終了などのために、端末装置から使用されるのが割り込み信号である。

割り込み信号は、緊急性を要するものであるから、入力されているがプロセスに渡されていないデータを追い越すことがある。ほとんどの場合にプロセスは

割り込み信号の受信によって入力命令を出すので、このようなときには、割り込み信号を入力する前に送られたデータは捨てられないと、処理結果がおかしくなってしまう。

このような同期の問題に対しては、次のような対策がとられる。

第1番目は、Host-Host プロトコルで、サービスされる割り込み信号の送信において、先行するデータを捨てるものである。

第2番目は割り込み信号を送ると同時に、データが流れるリンクに対して同期をとるための文字を挿入する方法である。

第3番目は、同期の問題は端末に利用者がいるとすれば、多くの場合に対応した処置がとれるので、利用者の端末操作手順で解決してもらう方法である。

同期をとる問題に関しては、第2番目の方法が最良であると考えられる。

4.2 ファイル転送プロトコル

ファイル転送プロトコル (FTP) は、大容量データを効率よく正確に送受信することを目的としたものであり、動作モデルは基本的には図-6 の形態となる。

FTP の基本的な機能としては次のものがあげられる。

- (1) 初期会話、終了会話手順
- (2) ネットワーク仮想ファイル (NVF)
- (3) データ転送方式
- (4) FTP のコマンド

FTP と ITP が非常に異なる点は、ITP が通信路の両端におけるコード系、制御手順だけを共通化したのに対して、FTP はユーザとサーバ間でのコマンドの共通化をはかっていることである。コマンドの共通化は、必然的に両ホストにおけるファイル管理などの違いを利用者に意識させないようにすることになる。

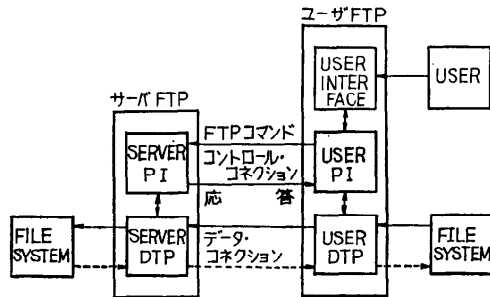


図-6 FTP の動作モデル

上記のうち(1)に関しては ITP においてすでに述べたので(2)~(4)についてのみ述べる。

4.2.1 ネットワーク仮想ファイル

NVF は、FTP において取り扱うファイルを ITP における NVT と同じように標準化しようとするものである。

いままでのところでは、NVF の設定には成功していないといつてよい。この理由としては、ファイルの管理、アクセスがそれぞれのホストで異なっていることがあげられる。ホストにおいては、順編成、直接編成などのファイルのアクセスをサービスしているが、FTP としては比較的標準化しやすい順編成ファイルだけを扱っている。

転送されるファイルの内容としては、プログラム、データ、ジョブ出力結果などがあり、ホストごとにそれぞれの表現が異なっている。FTP においては、基本的にはファイルの内容をそのまま他のホストにコピーすることを前提としている。もちろん、EBCDIC と IA5 間のコード変換などはサービスしている例も多いが、固定小数点および浮動小数点数の表現形式を相手に知らせて変換をするというサービスまでをオプションとして含むのは EIN の FTP だけである。

4.2.2 データ転送方式

ファイルの送信側と受信側の間にあって、実際のデータがどのような形で転送されるかを規定するものである。データの転送に関しては原則としてトランスペアレンシーを保証しながら効率よく転送することである。データ転送方式において利用されているのは次の方式である。

- (1) ストリーム転送
- (2) レコード化転送
- (3) コンプレス転送

ストリーム転送は、ファイルの最初から最後までをデータストリームとして扱って転送するものであり、送られた側でのファイルの記憶形態は、そのホストにおける標準的なファイルのそれと一致することになる。

レコード化転送は、一定のレコード長あるいはブロック長を持ったファイルを送るときに使用されるもので、この方式で送られた場合には、新たに作られたファイルは、もとのファイルと論理的には同じ構造となる。

コンプレス転送は、プリンタ出力、カード入力などのようにブランクなどの特別なコードが多いデータを

圧縮して効率よく転送できるようにするものである。

その他の機能として、受信側のファイルがラインプリンタなどの場合には、紙が切れるなどの理由によって再送を要求しなければならないことがある。この機能をサービスするために、リスタート・マーカーを転送データに挿入することがおこなわれる。

4.2.3 FTP のコマンド

FTP のコマンドは、ファイルの指定、転送方式、ファイルの送受信のいずれであるかなどを指示する機能を持たせている。

ファイル転送以外にも、ある利用者のファイル名のリスト、ファイルのダンプ、ファイル名の変更、ファイルの消去などのサービスをするコマンドも持つ。

FTP コマンドは、ファイル転送においてネットワーク標準という形で統一されているが、ファイル名やファイルの存在場所を示すようなコマンドにおいては、パラメータとしてローカル・ホスト特有の表現を許さなければならない場合がでてくる。このようなローカル・ホスト特有な表現を標準化するのが今後の大きな課題であり、この結果が FTP の有効性を決めるカギとなる。

5. おわりに

プロトコルの標準化というものは、コンピュータ・ネットワークにおいて避けて通ることができないものである。Host-Host およびハイレベル・プロトコルがどのような形でまとまるかという見通しは、現在のところではつかない。しかしながら標準化への要請というものが大きいので徐々にまとまった形になってゆくとと思われる。

参 考 文 献

- 1) ISO/TC 97/SC 6 N 1167: Agenda item 24-Future program of work, United Kingdom contribution on high level protocols, p. 5 (1975)
- 2) ISO/TC 97/SC6 N 1173: Working Document from ECMA TC 9 to ISO/TC 97/SC 6, p. 5 (1975)
- 3) ISO/TC 97/SC 6 N 1249: Minutes of meeting of ad hoc WG 1, p. 22 (1975)
- 4) Cerf, McKenzie, Scantlebury, Zimmermann: A proposal for an Internetwork End-to-End Protocol, INWG General Note # 96, p. 29 (1975)
- 5) Telnet Protocol Specification, NIC # 15371~15373, 15389~15393, p. 42 (1972)

- 6) Official Initial Connection Protocol, NIC # 7101, 7155, 7103, p. 7 (1971)
- 7) File Transfer Protocols, NIC # 17759, p. 50 (1973)
- 8) Remorte Job Entry Protocols, NIC # 12112, p. 24 (1974)
- 9) H. Zimmermann: Terminal Access in Cyclades Computer Network, Reseau Cyclades Ter # 510, p. 8 (1974)
- 10) H. Zimmermann: Virtual Terminal Protocol-Proposed Specifications, Reseau Cyclades Ter # 503.1, p. 14 (1974)
- 11) H. Zimmermann: Protocole Client-Serveur de Terminaux, Reseau Cyclades Ter # 502.2, p. 8 (1974)
- 12) EPSS Liaison Group: An Interactive Terminal Protocol, INWG General Note # 94, p. 61 (1975)
- 13) ESPP Liaison Group: A Basic File Transfer Protocol, INWG General Note # 93, p. 46 (1975)
- 14) P. Schicker: Virtual Terminal Protocol, INWG Protocol Note # 32, p. 25 (1976)
- 15) P. Schicker, et al.: Bulk Transfer Function, INWG Protocol Note # 31, p. 24 (1976)
- 16) R. Scantlebury, P. Wilkinson: The National Physical Laboratory Data Communication Network, Proc. of 2nd ICCS, Stockholm, pp. 223~228 (1974)
- 17) 日本情報処理開発協会: コンピュータ・ネットワーク JIPNET の研究開発, 50-S 003, p. 361 (1976)
- 18) 山本, 伊藤, 小川, 鍛冶他: 情報処理学会第16回大会論文集, 論文番号 # 108~# 114, pp. 215~228 (1975)
- 19) 電子計算機利用に関する技術研術会: リソースシェアリングシステム, p. 197 (1976)
- 20) H. Zimmenmann: High Level Protocol Standardization: Technical and Political Issues, Proc. of 3rd ICCS, pp. 373~376 (1976)
- 21) 北川敏男, 島内武彦編: 広域大量情報の高次処理, 東京大学出版会, p. 1300 (1976)
- 22) 吉田, 松下: パケット交換網の現状と国際標準化について, 情報処理, Vol. 16, No. 3, pp.220~229 (Mar. 1975)

(昭和51年11月30日受付)

(昭和52年2月10日再受付)