コンピュータ・ネットワーク 19－4
（1979. 2. 22）

ユーザレベルプロトコル間の構造に関する一考察

松下　温（沖電気工業）　今村幸雄（東京芝浦電気）
勅使河原　可海（日本電気）　山本欣子（(財)日本情報処理開発協会）

1. まえがき

近年，コンピュータシステムの発展と社会への浸透には著しいものがある。こうした情報化社会では，一方ではデータベースを中心とした情報データ資源の共用および交換を，又他方ではコンピューティングおよび通信設備資源の共用等を，より広い地域でかつより多くの情報処理システム間で実現することが要求される。ARPANET[1] の研究に始まり，現在，一層さかんに研究されているコンピュータネットワークあるいは情報ネットワークは，まさに，この要求の実現を目指すものと言える。しかしながら，この分野の進展は急速であり，その中心となる研究対象は，研究初期と現在とで大きく異なる。研究初期では，いわゆる，パケット交換やＨＤＬＣといった回線を介したデータ転送および交換，すなわち，コンピュータシステム同士の低レベルの結合を，その主な研究課題としていた。これに比べ，現在では，その対象はよりアプリケーションに近く，アプリケーションの特性を考慮した有機的且つ高位レベルの結合に移って来ている。低レベルの結合を実現するプロトコルについては，種々のコンピュータネットワークにおける精力的な研究[2]により問題点が明らかになり，ＣＣＩＴＴやＩＳＯといった国際標準化機構による標準化活動も進められて，その具体的な姿や方向はめどがついたと言える。一方，高位レベルの結合を実現するプロトコルについては，ネットワークに対するユーザニーズも固っておらず，現状ではネットワーク毎に異なった体系が採られ，標準化には至っていない。しかしながら，ネットワークの最終的な目的が利用者に種々の情報処理サービスを提供することにあることを考えれば，低レベルの結合は貧弱かつ消極的な資源共用しか実現できない。アプリケーションに近い高位レベルの結合，すなわち，高位レベルプロトコルを早急に定め，積極的な資源共用を実現することこそが重要であり望まれていることである。

この様な状況から本論文では高位プロトコルであるユーザレベルプロトコルに視点を置き，その構造化とコネクションの設計思想，仮想端末とユーザレベルプロトコルとの関係を論じる。また，代表的なユーザレベルプロトコルおよびその適用例の概要を示す。

2. ユーザレベルプロトコルの設計思想

2.1　ユーザレベルプロトコルの位置と問題点

先ず最初に，本論文の対象とするユーザレベルプロトコルのプロトコル階層上の位置を明確にする。次にユーザレベルプロトコルがそれより下位のプロトコルと異なる構造上の特徴を述べ，それにより生ずる設計上の問題点を明らかにする。問題点に対する考察は 2.2 節以降で行なう。

今，我々の関心は１章でも述べたようにアプリケーションに依存したレベルである。いいかえれば，転送する情報の内容の意味および属性が意識される通信のレベルである。このことは，これよりも下位でトランスポートレベルとかエンド　ツー　エンドと呼ばれるレベルが情報の透過性を保障し，単なるデータの転送を行なうだけであることに比べ，このレベルが情報処理に強く関与するということで特徴づけられる。

次にこのレベルをもう少し詳しく見てみると，このレベルを２つに分けることができる。一つは，アプリケーションに依存しないか，あるいは多くのアプリケーションに共通した情報処理および通信を行なうレベルであり，具体的にはファイル転送とかリモートジョブエントリといったものである。もう一つは，アプリケーション毎に異なった業務処理を行なうレベルであり，前者のユーザとなるアプリケーションレベルである。

ユーザレベルプロトコルの標準化によるシステム構築やメインテナンスの容易化を考えれば，前者の共通的かつ標準的なレベルがより重要であり，本論文の対象でもある。このレベルでは次の節でも示すように幾つかのプロトコル，例えばメッセージ転送プロトコル，ファイル転送プロトコル，リモートジョブエントリプロトコル等が存在する。しかし，このユーザレベルはそれより下位のプロトコルにくらべ，構造的に際立って異なる。すなわち，ユーザレベルを構成する各プロトコルは，アプリケーションレベルに対して本質的に目的の異った機能を対等な位置で提供する。にもかかわらず，その機能の実現に当っては，例えばファイル転送の中のメッセージ転送機能とか，リモードジョブエントリの中のファイル転送機能といった様に同一レベルに属する機能間に共通性および重復性の関係がある。このことは，下位レベルが単一目的でありかつ同一レベル内の共通あるいは重復関係がないということにくらべ極めて特徴的である。この特徴のために，個々のプロトコル詳細をどう定めるかという問題を論ずる前に，プロトコル間の関係構造に関して次の

3つを論ずることが必要となる。

(1) プロトコル間の階層関係……プロトコルの共有化や機能の効率的実行を実現するためにはプロトコル間にどの様な（階層）構造関係を持たせるべきか。

(2) コネクション……プロトコル間に階層関係がある場合，コネクションはどのように設定されるのか。

(3) 仮想端末との関係……仮想端末との通信制御はユーザレベルの構造とどの様に関わっているのか。

2.2. 階層構造化

ユーザレベルプロトコルを眺めてみると，2.1節でも述べた様にあるプロトコルの機能が別のプロトコルの機能よりも基本的であるという関係を見つけることができる。すなわち，ある基本的なプロトコルの機能が別のプロトコル機能のビルディングブロックとなり得る関係である。この時，その基本的なプロトコルが別の複数のプロトコルに対してもビルディングブロックとなるならば，自然な発想として，プロトコルを階層化して，より基本的なプロトコルを共用しようという要求が起こる。しかし，階層化は一方ではオーバヘッドを増加させ効率悪化の原因となる。このことは，ARPA[3]ネットワークでもすでに示されており，JIPNET[4]のように意識的に階層化を避けているネットワークもある。これらのことを考えると，プロトコルの階層化は共有による利点とオーバヘッド増加の欠点等を十分検討したうえで決定しなければならない。ここでは表1に示される基本的で重要な4つのプロトコル，メッセージ転送プロトコル(MTP)，ファイル転送プロトコル（FTP），リモートジョブエントリプロトコル（REJP），TSSプロトコル（TSSP）について，MTPの構造上の位置付けを中心に考察する。

MTPの位置付けとしては，MTPを他プロトコルの下位に共通プロトコルとして置くか，またMTPの機能をアプリケーションレベルへ直接提供するかの観点から，図1に示される次の3方式が有力である。

（方式1）— RJEP，TSSP，FTPの共通下位プロトコルとしてMTPを位置付けるとともにそのMTP機能を直接アプリケーションレベルにも提供するものである。この方式において，MTPとFTPやTSSPやRJEPとのインタフェイスはMTPとアプリケーションレベルのそれとは一般に異なる。すなわち，後者の方がより高レベルなインタフェイスとなるのが普通である。

（方式2）— 各プロトコルを対等な位置に置き，RJEPやTSSPやFTPにおけるメッセージ転送機能についてはそれぞれ別個なプロトコルを定める方式である。

（方式3）— MTPを独立したプロトコルとして置かずRJEPやTSSPやFTPの中で別個に定める方式である。この方式ではMTP機能がアプリケーションレベルに直接提供されることはない。

これらの方式の評価については以下の様に考えられる。

先ず情報処理ネットワークのアプリケーションを見ると，必ずしも全てのアプリケーションがFTP，TSSP，RJEP等の機能を使うとは限らない。むしろアプリケーションの多様性から，これらの機能を使わない場合も多いと考えられる。従って，MTP機能をアプリケーションレベルが直接使える方式1又は方式2が良い。

次に，方式2はTSSPやRJEPやFTPにおいてそれぞれ最適なメッセージ転送機能を設計できる。反面別個のプロトコルを設計することになり，共通化の利点は失なわれる。しかし，ファイル転送やRJEやTSSのメッセージ転送機能については共通部分が非常に多く，共通化のメリットは非常に大きいと考えられる。

以上の考察から結果として方式1が最も優れていると考えられる。

但し，ここで注意しておきたいのは，上記の階層化の議論がプロトコルについてのものであり，プロトコル制御プログラムについてのものではないということである。つまりプロトコル階層に対応してプログラムモジュール構造を設計するかどうかはインプリメンテーションの問題であり各プロダクト設計者の自由である。

各プロトコル間の階層関係を明確にしておくためにプロトコル制御ヘッダ間の関係を図2に示す。

表１　基本的ユーザレベルプロトコルの機能概要

		RJEP	FTP	TSSP
特別機能		○ユーザ識別子，パスワード ○入力ジョブファイルやジョブ結果ファイルの入出力要求および入出力停止 ○実行中ジョブや実行待ちジョブ等の打切り ○ジョブ処理状態の問合わせ ○システム／利用者メッセージの通知 ○出力の戻し／スキップ ○RJEサービス終結 ○その他	○送／受信ファイルのイニシエート ○チェックポイント ○転送中止 ○レコードのリトリーブ，ストア，スキップ ○領域確保 ○ファイル名の変更 ○ファイルの削除 ○ファイル情報のリスティング ○ファイルのセーブ（パーマネントファイル化） ○その他	○キャラクタ形式コマンドによる会話 ○相手確認 ○ステータス確認 ○送信スキップ ○プロセス割込み ○エコーモード指定 ○その他
共通機能	MTP	○コネクション設定／解放 ○メッセージ転送開始／終了の同期 ○フロー制御，応答制御，順序制御 ○送信権制御 ○オプションネゴシェーション ○エラー制御 ○割込み制御 ○データ圧縮，コード変換，データ形式変換 ○バイナリ／キャラクタ転送指定 ○その他		

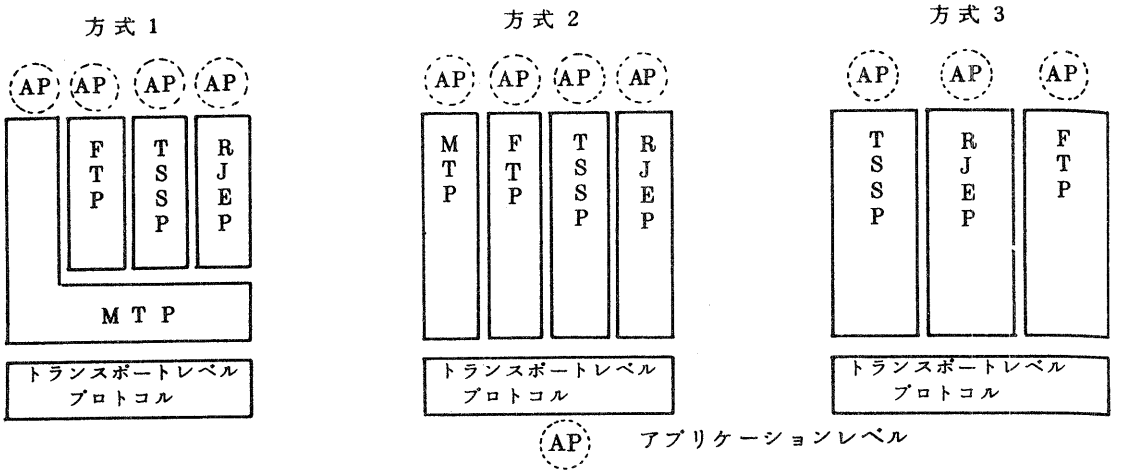

図１　ユーザレベルプロトコルの階層化方式

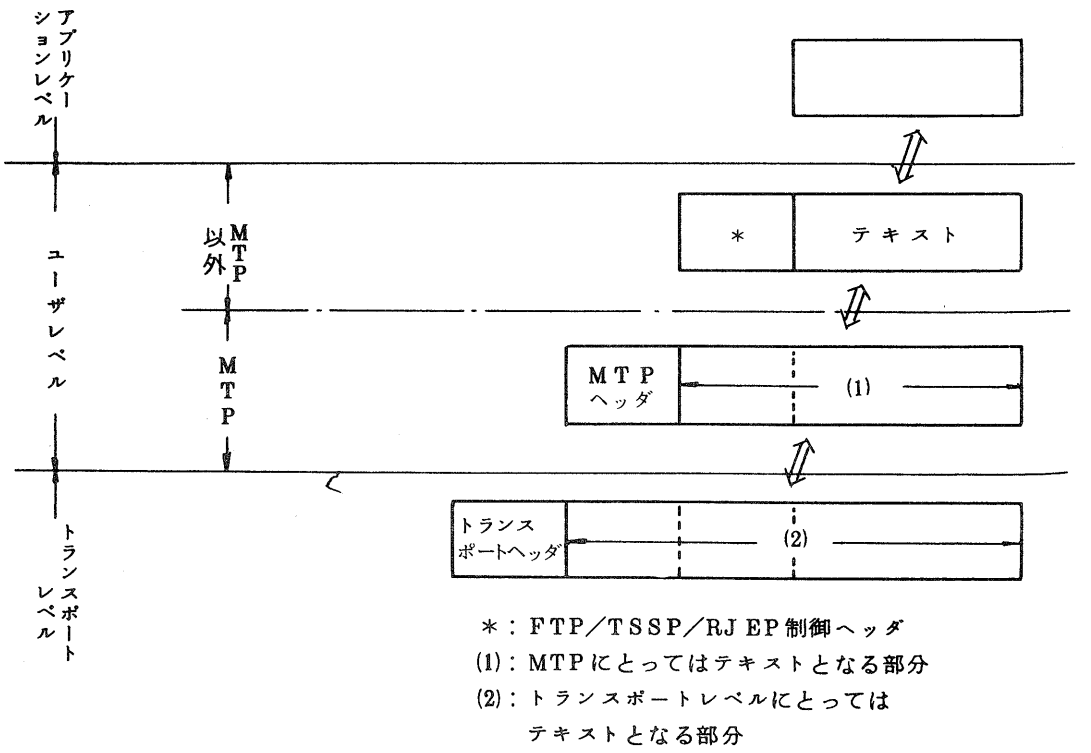

図2　プロトコル階層の制御ヘッダ形式による表現

2.3　コネクション

2.3.1　コネクションの数と設定および解放機能の位置

2.2節の議論によってＭＴＰと他プロトコルが階層化されることが明らかになったが，それに伴ないコネクションに関して問題が生ずる。すなわち，ＭＴＰを下位プロトコルとして使う場合

(1)　ＭＴＰのコネクションと他プロトコルのそれが共通なのか別ものなのか。

(2)　コネクションが共通であれば，その設定および解放をどのプロトコルに位置付けるか。

という問題を解決する必要がある。

先ず(1)の問題については次の様に考えられる。ＭＴＰをプロトコルとして独立させない場合は元々コネクションは１本ですむ。すなわち，コネクションの多重化を考えなければ，ＭＴＰと他プロトコルは１本の共通コネクションで制御が可能である。一方，コネクションを別々に設けると，資源としてのコネクション数が増えるだけでなく，通信オーバヘッドの増加や障害処理およびその回復が複雑化する。こうしたことから，ＭＴＰと他プロトコルは，多重化の必要性がなければ共通にするのが良いと考えられる。本来，多重化はコネクションの共有による有効利用を目指すものである。しかしながら，それは上位プロトコルの適用サービスの性質が似ている場合に効果があるのであって，例えばＦＴＰとＴＳＳＰといった通信特性の大きく異なったものの多重化を行うと，ＴＳＳＰの会話がＦＴＰの大量データ転送に影響されてスムーズに行えなくなるといったような多重化の欠点，すなわち，相互干渉が顕著になり望ましくない。その結果上位プロトコルが明確になればなる程，多重化は避けられるようになると考えられる。

次に(2)の問題であるが，ＭＴＰは共通機能であるが故に下位に位置づけられる。また，コネクションの設定解放機能は全プロトコルに共通する。従って，コネクションは，ＭＴＰが設定および解放し，他プロトコルがＭＴＰにその契機を指示するとするのが良いと考えられる。

2.3.2　コネクションの種類

ユーザレベルプロトコルが階層構造化されていることとの関係は無いが，次のコネクションに関する問題はプロトコルの構造を特徴づける重要な問題である。問題とは，コネクションにコントロール用とデータ転送用の２種類を設けるか

どうかである。2種類のコネクションを設ける利点は，EINのBTF[5]によれば次のように主張される。すなわち，「図3(a)においてAがBにあるファイルをCに移したい場合，コントロールとデータコネクションの区別がなければ，図3(b)のように，B→A→Cというように Aを経由して転送しなければならない。一方，コントロールとデータコネクションを区別すれが図3(c)のようにそれぞれのコネクションを設定することにより，Aを経由せずにBからCへ直接ファイルを転送でき効率が良い。」と主張される。しかし，BTFではコントロールに関してプロトコルに主従関係を前提としており，図3ではAがマスタでBがスレーブであり，BがCを制御できないために上記の利点が出て来ている。従って，プロトコルに主従関係を前提としなければ，図3(d)のようにAがBにファイル転送を依頼し，BがCとの間のコネクションを設定すれば，BからCへの直接のファイル転送が可能となる。すなわち，コネクションをコントロールとデータで分ける必要性は特にないと考えられる。

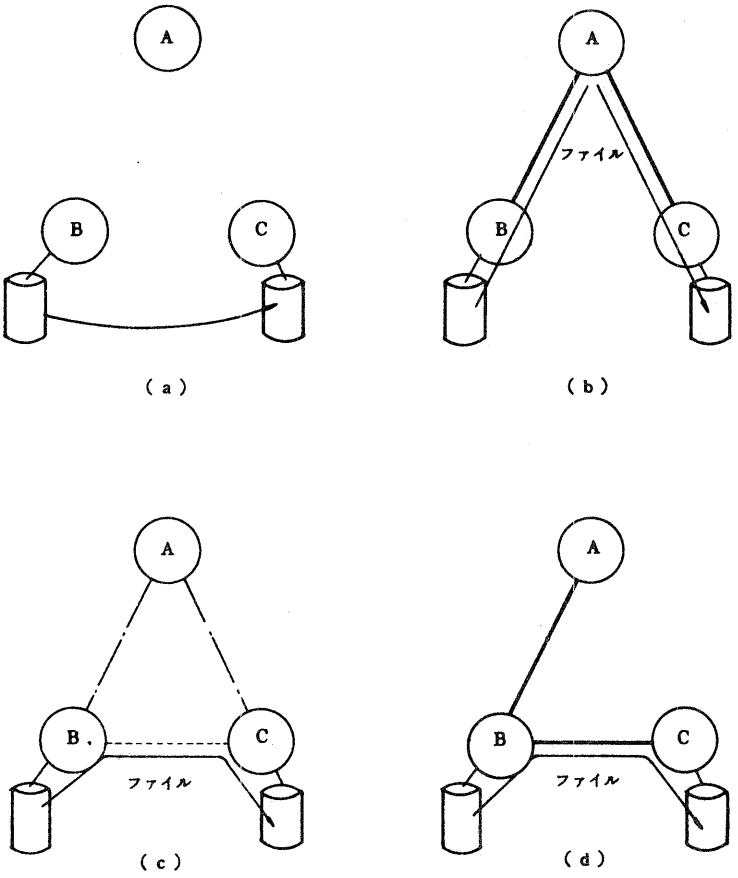

図3　コントロール／データコネクション

2.4　仮想端末とユーザレベルの構造との関係[10][11]

ネットワークに仮想端末の考え方が導入されると，ネットワーク内でリモートにある端末は，ネットワーク仮想端末（ＮＶＴ）仕様に従って制御されることになる。このとき，当然のことながら通信が生ずるため何らかのプロトコルが必要となる。このプロトコルが独立したＮＶＴプロトコルとして存在するのかあるいはユーザレベルプロトコルの内に吸収されてしまうのかを明確にすることが重要である。これはＮＶＴとユーザレベルプロトコルとの関係を明確化することだけにとどまらず，ＮＶＴのネットワークへの体系的かつ効果的なインプリメントを実現するうえでも重要な問題である。

対端末のユーザレベルプロトコルおよび通信機能を考えると，その通信の流れや手順，および個々の機能はそれぞれその処理対象となる端末を想定することにより設計される。すなわち，ＲＪＥ通信はＲＪＥ端末を，ＴＳＳ通信は，ＴＳＳ端末を，ファイル転送はファイル付端末を，またメッセージ転送は汎用キャラクタ端末を想定して設計される。この想定する端末は個々の実端末ではなく，標準化かつ仮想化された端末である。すなわち，ＮＶＴの仕様に加えて各プロトコルに特有で端末やホストで差を生じない標準化された通信機能を持つ端末である。例えば，ファイル転送は対端末ということで特有に生ずる端末構成関連のＮＶＴ仕様と，端末やホストで差を生じないリトリーブ，ストア等ＮＶＴ仕様とは関係のない，標準のファイル転送コマンドの両方を含む。

この様に，ユーザレベルプロトコルはそれぞれＮＶＴとの通信規約を含んでおり，その通信規約を決定するために必要なＮＶＴ仕様がユーザレベルの背後に横たわっているものとしてとらえることができる。但し，各プロトコルにどの様なＮＶＴ仕様を取込むかの基準は共通性により決定されるものと考えられる。表２に以上の設計思想に基づく各プロトコルへのＮＶＴ仕様の取込み例を示す。

表２　各プロトコルで取込むＮＶＴ仕様の例

プロトコル	取込むＮＶＴ仕様
ＭＴＰ	・書式編集機能（入出デバイスの行幅，行数，ＴＡＢ設定，改行，改頁など） ・端末構成機能（入出力デバイス，入出力アドレスの指定，エラー検出およびエラー回復の通知等） ・基本機能（コード，機能キャラクタ等） ・オプションネゴシェーション ・そ　の　他
ＦＴＰ	ＦＴＰとしては特になし，ファイル転送としてＭＴＰを介して，次のものを取込む。 ・オプションネゴシエーション ・基本機能 ・端末構成機能
ＴＳＳＰ	・相手確認 ・ステータス確認 ・プロセス割込み ・そ　の　他 ｝などＮＶＴ仕様の中の会話機能
ＲＪＥＰ	・ＲＪＥＰ独自としてはなし，リモートジョブ転送としてＭＴＰを介してＮＶＴ仕様を取込む

3.　ユーザレベルプロトコル概要

本論文では，現存の種々のネットワークにおけるアプリケーションに対して，広範囲に使用されている機能に着目し，ユーザレベルの標準プロトコルの対象とする。次の４種のプロトコルである。

(a)　メッセージ転送プロトコル（ＭＴＰ）
(b)　ファイル転送プロトコル（ＦＴＰ）
(c)　リモートジョブエントリプロトコル（ＲＪＥＰ）
(d)　会話型通信プロトコル（ＴＳＳＰ）

もちろん，これらの他にも，ユーザレベルプロトコルとして，標準的なものを確立する必要がある機能も数多くある。しかしながら，現段階では，コンピュータ機種の違いによるオペレーティングシステムの相違等の理由により，標準プロトコルとすることは現実的でないと考えられる。今後の課題であろう。

前述の4種の各ユーザレベルプロトコルは，共通の基本概念に基づいて考えることができる。すなわち，各プロトコルで実現するサービスのサーバプロセスと，ユーザプロセスとが存在し，この両者間のデータ転送に必要な規約が当該サービス用のユーザレベルプロトコルであるとする考え方である。これを図4に示す。

図4に示した様に，ユーザプロセスおよびサーバプロセスそれぞれは，アプリケーションとは，ローカルインタフェースを持ち，互いに相手のアプリケーション（上位レベル）とのインタフェースについては知る必要がない。

また，2章で述べた様に，MTPはFTP，TSSP，RJEPを実現する際に，ホスト間でメッセージ転送に関する共通機能を抽出したものであり，これらのユーザレベルプロトコルの下位に位置付けられ，トランスペアレントな転送機能を果すトランスポートレベルの上位に存在する。よって，図4は図5の様に書き直すことができる。

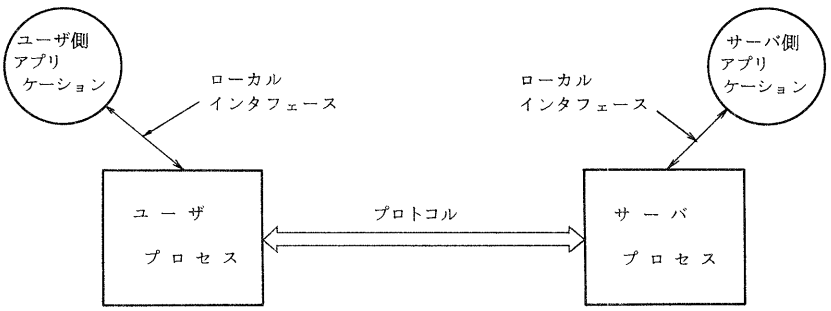

図4　ユーザレベルプロトコルの基本概念

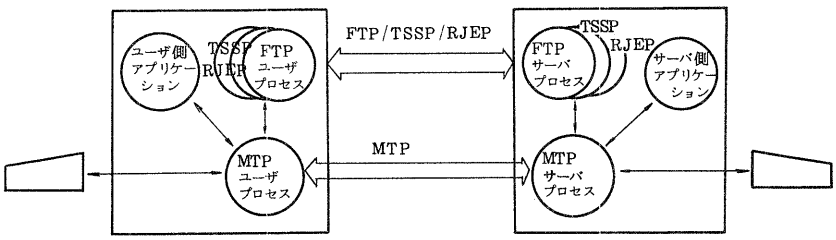

図5　MTP，FTP，TSSP，RJEPの基本概念

3.1　各ユーザレベルプロトコルの機能と特徴

前述の4種のユーザレベルプロトコルの機能項目は表1に示した。ここでは，各プロトコルでの特徴的な事項について示す。

(1)　ＭＴＰ

ＭＴＰは，図1に示したように他のユーザレベルプロトコルの共通機能として，上位のユーザレベルプロトコルに使われる場合と，アプリケーションから直接使われる場合とがあり，それぞれインタフェースが異なる。つまり，アプリケーションとのインタフェースが仮想端末制御用のインタフェースを含むこと，および他のインタフェースに比べ，アプリケーションが使い易い様に高度化されている点が異なる。

(2)　ＦＴＰ

図6に示す様に，ファイル転送を要求した場所と，ファイルを持った実際のサービス場所とが離れた位置にある場合に対する処置も考慮されている。つまり，コントローラはＭＴＰを使用してサービス点(A)に対しＦＴＰサーバ

の起動をかけ，実際のファイル転送は，サービス点(A)とサービス点(B)との間で行ない，ファイル転送終了後，サービス点(A)はコントローラに対して，ＭＴＰを使用して，その旨通知する。

(3) ＴＳＳＰ

ＴＳＳＰは，ＮＶＴの考え方に基づいている。つまり，ネットワーク全体にわたって適用される標準端末の存在を仮定し，この標準端末を用いて会話型サービスを実現する場合に必要となるプロトコルを定めている。従って，ＴＳＳＰでは個々の端末の相違は意識する必要がなく，ＴＳＳユーザプロセス間でやりとりされるコマンド群だけを規定している。なお，ＮＶＴ機能と１対１に対応するコマンドについては，ＮＶＴ機能そのものを使用している。また，会話型サービスでは不要なＮＶＴ機能（例えば，行挿入の様な複雑な編集制御や全二重通信制御等）は，ＴＳＳＰでは使用しない。

以上の様に，ＮＶＴをＴＳＳＰの背後に横たわる標準端末と考えることにより，簡明なＴＳＳＰを設計することが可能となる。

(4) ＲＪＥＰ

通常のＲＪＥサービスが，遠隔地の端末からセンタの計算機を利用するのに対し，ここでは，ネットワークに接続された複数ホストのＲＪＥサービスを利用できるようにするものとして考えた。また，ＲＪＥサービスでは，ジョブを入力するデバイスと，ジョブ結果を出力するデバイスとは一般には異なる。したがって，入力ジョブや出力データの入出力を正しく制御するためには，現在アクセスしているデバイスを正しく把握していなくてはならない。つまり，各デバイス毎にＮＶＴを規定し，ＲＪＥ端末としては複数台の仮想端末を扱うことが必要となる。この場合，ＲＪＥＰサーバおよびユーザプロセス間のコネクションを１本だけとし，オーバヘッドを少なくできる様考慮している。

図７にＲＪＥＰのコマンドおよびレスポンスの具体例を示す。

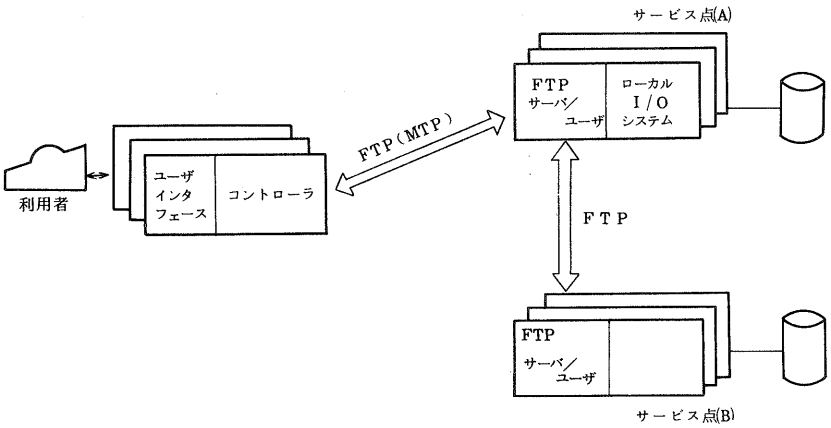

図６　要求場所とサービス点とが離れている場合のＦＴＰ

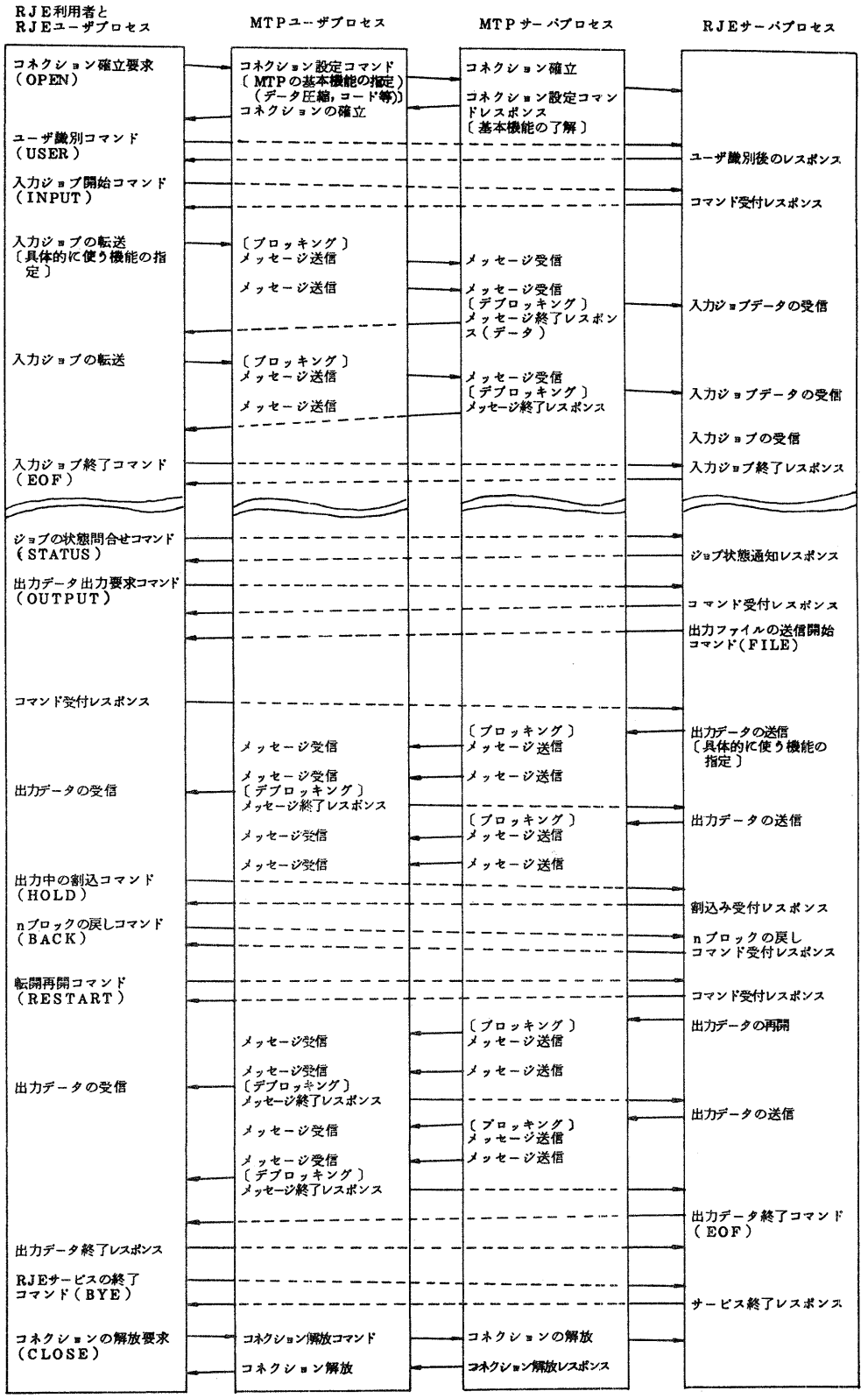

図7　ＲＪＥＰコマンド／レスポンスの具体例

3.2　プロトコルの応用例

前節まで述べた各ユーザレベルプロトコルの応用例として，ＦＴＰとＲＪＥＰを使用する３者通信について，プロトコルシーケンスを示す。この例は，図８に示すように，ＲＪＥサービスを受けるノードＡ，ＲＪＥサービスを提供するノードＢ，およびジョブの実行中に使用されるデータファイル(F)の存在するノードＣの３者間で通信が行われる場合である。

プロトコルシーケンスを図９に示す。

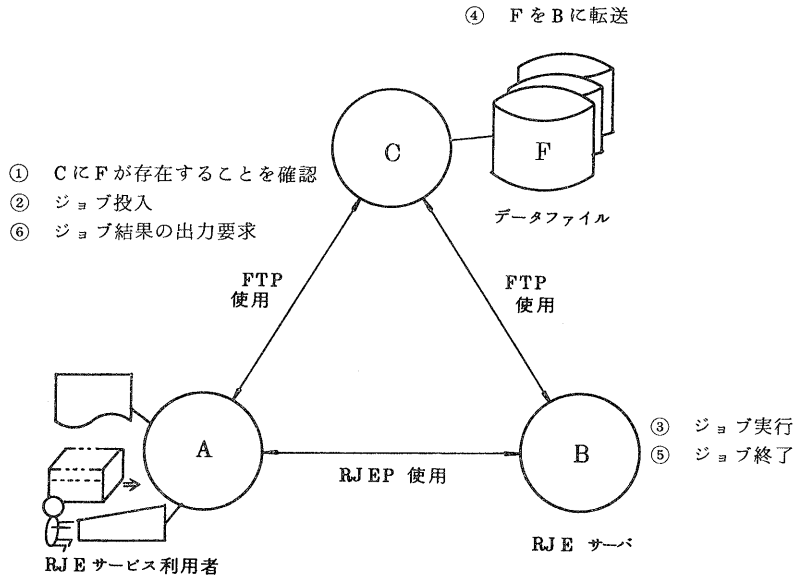

図８　ＲＪＥＰとＦＴＰを使った３者通信の例

4. ま と め

本論文では，4つの非常に基本的なユーザレベルプロトコルについて，その階層構造関係を中心にコネクションや仮想端末との関係を議論した。また，各プロトコルの概要と使用例を示した。本論文の目的はユーザレベルの設計思想にある。その意味では個々のユーザレベルプロトコルを相互に無関係に設計する場合には本議論はあまり役に立たないかもしれない。しかし，情報ネットワークの発展を考える場合はネットワークのプロトコル構造を体系的に構築して行く必要があり，その一端として本議論は非常に重要であると考えられる。

この論文でとりあげた問題の他にもユーザレベルとして今後早急に解決して行ねばならない問題点は幾つかある。それらはネットワークの運用を制御するネットワーク管理や，機密保護，ＮＶＴ仕様の完全性等である。特にネットワーク管理については，実際のネットワークシステムを安全にまた効率良く動かしていくために是非とも議論される必要がある。

謝　辞

本テーマの検討に当り，多大なる御協力をいただいた㈶日本情報処理開発協会，国際情報ネットワーク技術調査委員会の大島裕（日本電信電話公社），大野圭三（国際電信電話㈱），伊藤哲史（㈶日本情報処理開発協会）平子叔男（㈱日立製作所），中村利武（富士通㈱），松永宏（三菱電気㈱）の各委員および事務局各位に深く感謝申し上げます。

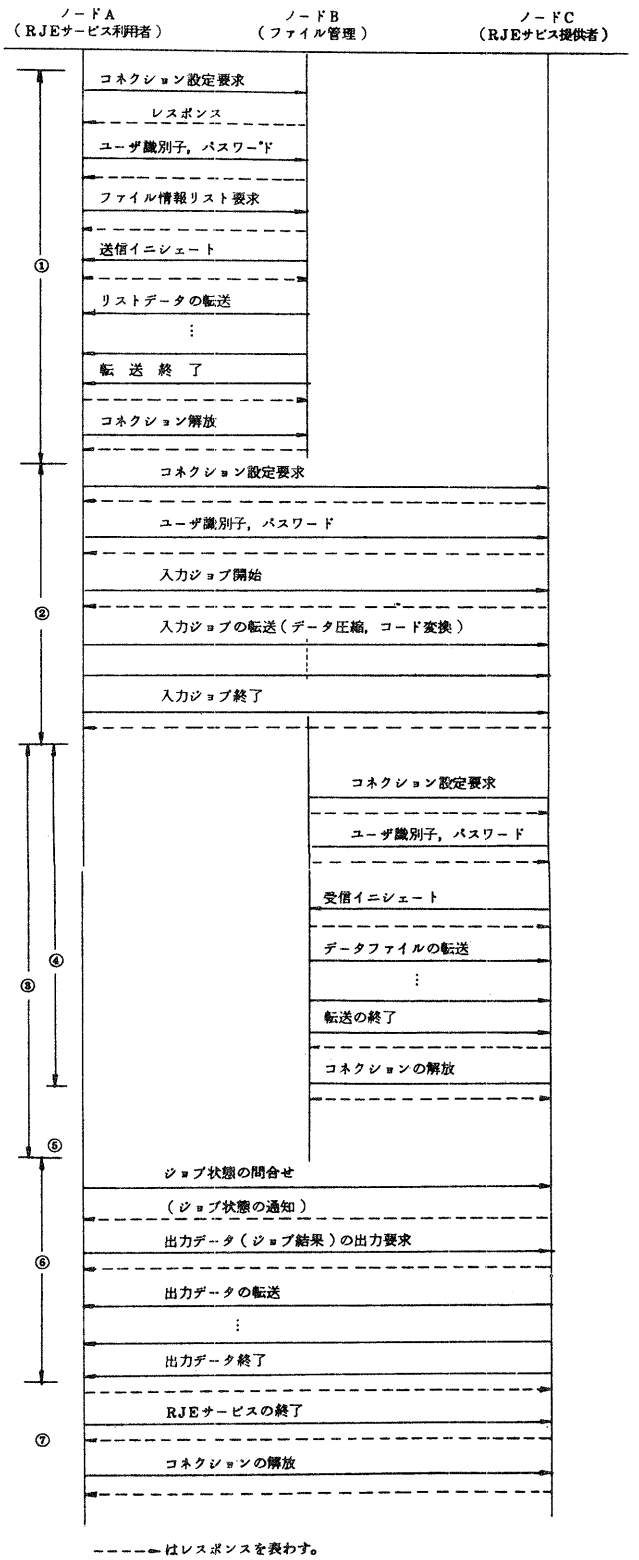

図９　３者通信例におけるプロトコルシーケンス例

参照文献

(1) L. G. Roberts, B. D. Wessler : Computer Network Development to Achieve Resource Sharing, AFIPS Conf. Proc, Vol. 36, pp 543-549 (May 1970)

(2) 国際情報ネットワークに関する調査研究報告書　51-R-008　日本情報処理開発協会

(3) S. D. Crocker et al.: Function-Oriented Protocols for ARPA Computer Network, AFIPS SJCC, (1972)

(4) コンピュータネットワークJIPNETの研究開発　50-S-003　日本情報処理開発協会

(5) P. Schicker, A. Duenki and W. Baechi: Bulk Transfer Function, Proposed Specification, INWG Protocol #31 Sep. 1975

(コンピュータネットワーク)

(6) C.S.Corr, et al.: Host-Host Communication Protocol in the ARPA Network, AFIPS Conf. Proc, Vol. 36, pp589-597 (May 1970)

(7) L. Pouzin : Presentation and Major Design Aspect of the CYCRADES Computer Network, 3rd Data Communication Symposium, IEEE (Nov. 1973)

(ユーザレベルプロトコル)

(8) The High Level Protocol Group : A Network Independent File Transfer Protocol, INWG Protocol Note 86, (Dec. 1977)

(9) Remote Job Entry Protocols, Standard Research Institute, NIC # 12112, (Dec. 1974)

(仮想端末)

(10) H. Zimmermann, et al. : Preliminary draft version of the IFIP WG 6.1 proposal for a VIRTUAL TERMINAL PROTOCOL, July 1977 INWG Protocol 78

(11) J. Davidson, W. Hathaway, et al.: The ARPANET TELNET PROTOCOL : Its purpose, principles, implementation, and impact on host operating system design