

## T-NETの再結合プロトコルと仮想網

川合英俊(電総研), 海老原義彦(筑波大学術情報処理センタ)  
八重樫純樹(東北大応情研), 高橋薫, 野口正一(東北大電通石研)

はじめに

T-NETは公衆電話網を通信手段にもつ計算機網(1)で、電話の取次ぎにあたる再結合プロトコル(2)によれば、使用者が遠隔ホスト同士の間にはファイルの転送を行なわせ結果を自分の所にとりよせることが出来る。つまり網全体をひとまとまりの資源と仮想する途がひらける。

しかし、一本の電話線でいくつもの再結合を演ずると誤りの回復手順等に無理を全するるので、一個だけに制限を設ける手段を紹介し、あわせてより進んだ仮想網に発展させようとするときの問題点を探ってみる。

### 1 T-NETの環境と再結合プロトコルのねらい

計算機網とは複数のホストの間に論理的通信経路(リンク)が網状に存在するものをいうが、T-NETはホストと回線の間にミニコンをFEP(フロントエンドプロセッサ)として介在させており、次のいくつかの特徴のうち特に3巻目を主眼にしている。現バージョンのFEPは回線を一本(かまたは、一人のユーザにしかサービスしていない。

- (1) 自分のホストだけでなく遠隔ホストをも、同一の端末から用いる。
- (2) 複数のユーザが網内の資源を共用する。
- (3) 複数のホストに分散する資源をユーザが総合的に利用する。
- (4) 広域に分散して発生するデータを階層的に処理する。

これらは互いに独立でない上、すべて

に(1)や(4)は実用化されている。

(3)を重視したときT-NETは次の要素からなると考えるのが適当である。

- a. 網使用者: FEPに1人ずついて開発をかねている。遠隔のユーザはホストから自分のユーザにみえる。
- b. 網資源: FEPには、テストファイルとコンソールユーザインタフェースからなるミニホストを置いて、ファイル転送の実験に用いた。仮想端末やデータベースなどほかのプロトコル実験には、下図におけるホストの資源を用いた。仮想端末プロトコル(VTPあるいはTELNETともいうことが多い)によってTS6やACOSを用いることができ、データベースプロトコル(DBP)によって文献情報(ERIC, CISEなど)を用いることができる。

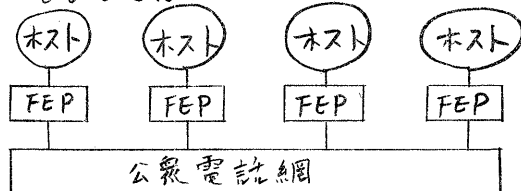


図1 FEPによる網構成

- c. プロトコル処理系: 階層構造をもつてFEPにおかれ、回線の制御、パケットの交換、プロセス間リンクの管理、資源やユーザの制御の各階層(1), (3)からなる。

一番上の資源制御層のプロトコルを処理するプロセスは対になっていて、次回に示すようにこの層のプロトコルはこれら一対のプロセス間のインタフェースとして記述される。特定の遠隔ホスト固有

の知識はサーバプロセスに独占されており、網では規定されない。この層のプロトコル族はそのうちのVTPとFTP(ファイル転送プロトコル)とをベースにしており、2.はその再結合の場合の拡張にあつてゐる。

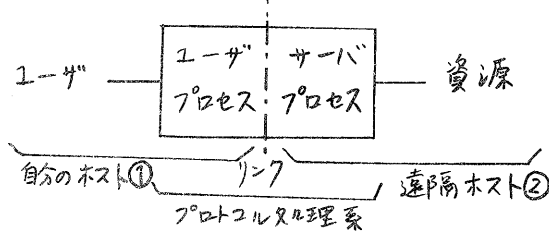


図2 プロトコル処理系の位置と構造

再結合プロトコルは、①のユーザーが②とある遠隔ホスト③との間にファイル転送を行わしめることを基本的なぬらいとしている。その実現のためには②のサーバプロセスが③に電話を取次ぐことができればよい。こうすれば①の資源が小さくても①のユーザーは多数ホストの環境に適応することができる。

**2** 再結合によるファイル転送プロトコルの拡張

再結合プロトコル(RCP)はFTPに3個のサブコマンドを追加して拡張したものである。

a. FTPのサブコマンド

- 1). Define : 転送モード(バイナリか文字列か)を打ち合わせる。
- 2). Catalog : 遠隔ホスト内のファイル名一覧表をとりよせる。
- 3). Store : ファイルを①から②に送る。②で必要なら作成される。
- 4). Retrieve : ファイルを②から①にとりよせる。複写が行なわれる。
- 5). Delete : ②のファイルを削除する。本体は②の固有ファイルとなる。

ユーザープロセスはこれらのサブコマンドの実行にあつて、その返をサーバプ

ロセスに届け、実行結果の応答を受けとる。特に3).と4).では新しいリンクを設定する(1).③)。コマンド実行時にユーザーは'割り込み'を起すことができ、このときサブコマンドの実行が中断されシステムはサブコマンド入力待ち状態に戻る(1)。

b. 追加したサブコマンド

- 1). SUB mit : ②が解釈実行すべきコマンド列を送る。
- 2). SWitCh : ②にコマンド列の実行開始を命じ、①は回線を切る。
- 3). STAtus : ①が②からコマンド列の実行結果をとりよせる。

シーケンスに制約があり例外は最初のSTAはユーザープロセスによって禁じられる。応答の例として③から②へファイルをとりよせるコマンド列を①のユーザーが網へ発せする場合を紹介する。大文字はタイプイン、小文字が網からユーザーに表示される応答でいずれもリンク上では終了符号が末尾に付加される。@は網コマンド、?はサブコマンド、\*はコマンド列の投入をユーザーに促している。

@ LOG ON ②

logged on

@ RE CONNECTION (注:RCP)

Input RCP command

? @ SUB

\* @ LOG ON ③

\* @ FTP

\* @ DEF A, R

\* @ RET F2, F3

\* @ BYE

\* @ LOG OFF

Submitted n

Input RCP command

? @ SWT

Switching

Successful reconnection

Input RCP command

? @ STA N

-----

Status file n Date Time

① LOGON ③ logged on

② FTP Input FTP command

? ④ DEF Input FTP command

? ⑤ RET F2, F3

File transfer completed

Input FTP command

? ⑥ BYE Byed

⑦ LOGOFF logged off

-----

End of status

Input RCP command

? ⑧ BYE

Byed

⑨ LOGOFF

logged off

この例で、\*の6行がコマンド列で、②のサーバプロセスはこれを番号mのコマンド表として保存する。Switchingのあとの空白の行は、①のユーザプロセスが電話を切ったあと再び②から電話がかかってきたことを表わしていて、ここでメッセージは終了してない。この再び電話をかけることが再結合と呼ばれる。このときタイミングによっては4.にのべるような問題が起る。

桌線ではさされたステータス表は、②のサーバプロセスが③と通信した結果を表わしていて、コマンド列を実行したとき②で作成される。誤りを発見したときこのサーバプロセスだけはデバッグに備えて停止するなびという事はあつてはならない、誤り応答をステータス表に書き込んでユーザに返すため再結合しなければならない。①や③は誤り発見時に停止しても網全体のサービスを損ねることにならない。

**3** 再結合プロトコルによるファイルの網内直並列転送

コマンドのシーケンスを工夫すると、

ファイルを多数のホストにばらまいたり、分散したファイルを特定のホストに集中したりすることができる。

a. ファイルの並列転送

下図のように②のファイルを③, ④, ---の数多くのホストにばらまくときには数多くのコマンド列をSUB mitすればよい。

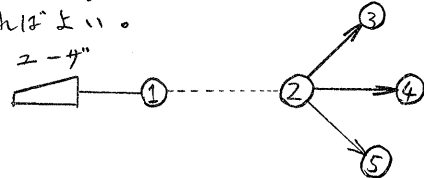


図3 ファイルの並列転送

②のファイルF2を(m)のFmに複製するには次のようにコマンド列を直列に並べる。応答を省略すると、

① LOGON ②

④ RCP

? ⑤ SUB

\* ⑥ LOGON (m)

\* ⑦ FTP

\* ⑧ DEF A, R

\* ⑨ STR Fm, F2

\* ⑩ BYE

\* ⑪ LOGOFF

? ⑫ SUB

\* コマンド列のmにm+1を入れたもの

⋮

? ⑬ SUB

\* コマンド列のmにm+2を入れたもの

⋮

? ⑭ SW T

⋮

②が①に再結合したときユーザがどのステータス表を見るかは、④SUBの応答にサーバプロセスが挿入した番号nで指定する。TELNETの各FTPはそれぞれ2~8枚分のステータス表を保持できる。

②でコマンド表の内容を解釈実行するとき、FTPのユーザプロセスが起動される。したがって、全てのコマンド表の実行が終わったとき、RCPのサーバプロセスに制御を戻して②SWTの後処理をしなければならぬ。つまり②は再結合に続いて①のユーザプロセスを再起動し、それに応答メッセージの後半の文章を届けるのである。再結合の発呼時に①が「話し中」だったときの処置については4.でふれる。

この例でSTRをRETとすればファイルのかき集めを行なうことができる。

### 4. ファイルの直列転送

下図のようにホストからホストへ順にファイルをわたしたり、ホスト同士の向かい合ったファイル交換するには、コマンド列の中にさらにRCPを用いることができる。

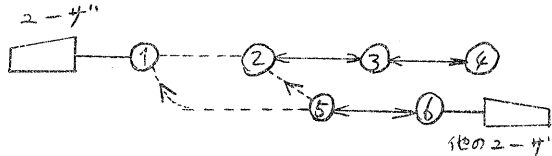


図4 直列転送と競合

①のファイルF1を順に④のF4に届けるとき、①のユーザがシステムに与えるコマンドシーケンスは次の通りである。

```

① LOGON ②
① RCP
?@ DEF A, R
?@ STR F2, F1
?@ SUB
*@ LOGON ③
*@ RCP
*@ DEF A, R
*@ STR F3, F2
*@ SUB
*@ LOGON ④
*@ FTP
*@ DEF A, R
*@ STR F4, F3
*@ BYE
  
```

②が実行する。

③が実行する

```

*@ LOG OFF
*@ SWT
*@ STA [ ]
*@ BYE
*@ LOG OFF
?@ SWT
?@ STA
?@ BYE
@ LOG OFF
  
```

②が作るステータス表は③が作ったもののうしろに付け加えたものである。[ ]印は①のユーザがデフォルト指定することを表わしていて、そこには②のユーザプロセスが③のサーバプロセスから割りつけたもらった表番号を挿入する。α.の並列転送のコマンド列と組み合わせることもでき、そんなときには沢山のSTAを並べておく。

コマンドシーケンスの検査はユーザプロセスとサーバプロセスの両方で行われ、前者はユーザに訂正を促かし、後者は②のFTPにマビンプレメントせられていないサブコマンドを断絶するのに用いられる。したがって、全てのコマンド処理ルーチンをもっていない未熟のFTPも網のサービスに参加できる。ユーザがその場で学習する。

以上の2例から、③が実行するコマンド列のようなものをマクロ化することが示唆される。これは5.でべる網の仮想化につながる。次にのべる障害はFTPに設けられている回線数(現在は1)の期に網を同時に使用するユーザ数が多いときに起る。

### 4. 多重サービスにおける障害

多重サービスとはFTPが沢山のトランザクションを処理するという意味でなく、それぞれ限られた範囲の資源を用いる複数のユーザに網全体がサービスする意味である。左図でホストが

6台あるものとする。

- 1). ⑥が⑤と通信するだけのとき①のユーザの占有資源との間に競合はない。
- 2). ⑥が④と通信しようとする時、それぞれ③が④と通信している間だけ④は'話し中'となる。T-NETでは20秒おきにする度まで発呼を試みる。
- 3). ⑤が④を用いるとき'話し中'の前後でF4の内容は異なる。内容の保証問題は5.で考えるが、CATで調べたファイルが使うときにはもう他人によってDEしされていることすらありうる。
- 4). ③と④が通信している間、①や②の電話は'話し中'でないのに①のユーザは②を使用中である。このとき着呼を許さなければ再結合を受入れることができなくなる。T-NETでは強制していないが、実効的システムロックを避けるため他のユーザのサービス要求は受入れた方がよい。
- 5). ③と④が通信中に⑤が②を使おうとすると、②のサーバプロセスの中止再開始を制御しなければならぬが、時間切れ制御(1),(4)との重なりも起るのでT-NETのようなミニコンFEPでは困難である。特に②がRCP実行中に重ねて⑤からのRCPを受けたときの誤り回復手順は複雑すぎる。

そこで次の妥協策を構じた。すなわち、RCP両プロセスはSWT実行中に電話が空いたとき、着呼は受けるが重ねてRCPのサービスは始めないものとするのである。このサービスを限定すると次のようになる。

- a. 再結合を受入れるときのプロトコル番号を新しいものRCP2とする。
- b. SWT実行中に重ねてRCP以外のサービスをしたときには、それが終わったとき再結合の催促のためユーザプロセスが発呼してRCPのサーバを起動する。
- c. (再結合しようとしたとき)相手が'話し中'だったら20秒おきにする回まで発呼を試みる。

これを図示すると下図のようになり、具体的にはRCPプロセスの中で擬似的にLOGONコマンドを発行することによってなされる。またRCPの重複を避けるには、着呼処理中にプロセスイニシエータ(PI)(1)がプロトコル番号を解釈したときPI対向リンクを切ることによって'断る'意を知らせる。応答を伴った方が親切かも知れないが、T-NETではこの場面での例外事象を嫌ったのである。 時間

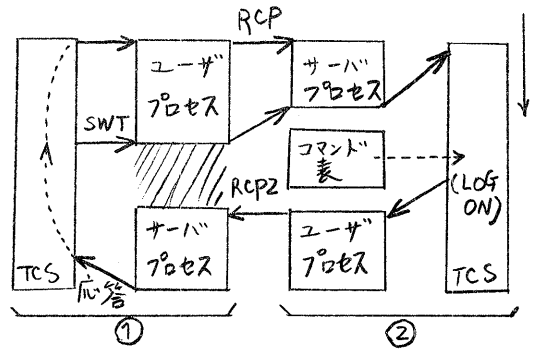


図5 再結合RCP2のタイミング

上図のRCP2に伴うプロセス対は、実はほとんど抜けがらでもよい。なぜなら①は2.の Successful reconnection という応答の内容は解釈しないし、RCP2で'話し中'だったときすでにこの応答メッセージは②の出力バッファに入ってしまっていて、制御はRCPサーバプロセスに戻り、長い時間切れ(現在40分程度)待ちになっているからである。上図のハツ4の部分が限定サービス実行中であることを表わしている。またTCSはユーザ端末を制御したり、コマンド表から読んだコマンドをも解釈実行したりすることができる端末制御システムである。

### 5 仮想網における問題

仮想網には2段階考えられる。回線を仮想して計算機網をホストの集合とみるものと、リンクを仮想して計算機

網を(ユーザを含む)資源またはそれを制御するプロトコル処理プロセスの集合とみるものである。ここではプログラムファイルについては前者、データファイルについては後者をとる立場に立つ。言い換えればファイルの転送は所属するホストを意識しないで行ない、プログラムを走らせるときに限って所属するホストを意識するのである。

まず、同一の資源を複数のユーザが同時に使用する問題を論外におく。同時になくても共用することには意義がある。1の(2)はせしめ資源の作製または保守者とユーザが互いに遠隔地にいるとき(異なったホストに所属しているとき)にあてはまる。T-NETのようなデータベースを重視する実験はこの状況に合っている。また、同一の資源を奪い合っていないのに通信経路を共用する(電話を取次ぐ)ため他のユーザに障害を与えることがあり以下の検討を要する。

a. 網の資源にはプログラムとデータとの2種があり、プログラムは特定のホストと結びつくまでデータとみなす。データは網に所属して、特定のホストに論理的に所属しない。

ユーザはファイルが物理的に所属するホストをあらかじめ指定しなくても目的のファイル名を指定できる。資源の所在に関する知識は網側でもつのである。そのためには向い合わせセンタ<sup>(15)</sup>が必要。データの転送はFTPで、プログラムの実行はVTTPで行なう。つまりRUNコマンドはFTPに設けないのである。

b. 網がサポートする料能は、VTTPとFTPを基本とする。

プログラムを実行するにはVTTPを用いるから、このとき遠隔ホストについての知識はユーザが持っているのがあって網は提供しない。VTTPのサーバプロセスは遠隔ホストの端末に関する知識はもつていても、プログラムという網の資源に関する知識はもち合わせていない。

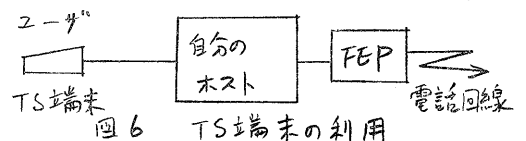
T-NETではプログラム資源の種類別に資源制御層のプロトコルを設けている。例えばデータベース用のDBPはデータベース使用法に関するガイダンスをユーザに提供している。この種の知識を理解することは、サーバプロセスにとっては困難であっても、計算機技術に関する基本的な知識をもつ人間にとっては容易である。例えば、データベースの場合には、データの論理的な構造、探索要求の形式、サブセット間の論理演算形式などを自然言語でガイドすればよいのである。

c. 網コマンド言語をマクロ化すると自分のホストの処理系にプロトコル処理系を見かけ上埋めこんでしまうことができる。しかし、双方の処理系はいずれも固定しないから現実的でない。ゆるい結合が望まれる。

T-NETではプロトコル処理系の起動と制御の発端を端末コマンドにしている。コマンドの形式はファイル等の所在に関する知識をユーザに期待しており、サーバプロセスがせえ当の遠隔ホストの資源に関するとしが知らない。

この網コマンド系をホストの処理系に埋めるには、@で始まるアセンブラマクロを設けておいて、これを四つのTCSの入力とみてユーザプロセスが扱えばよい。応答の解読能力に限界があったり、異常応答の処置にコマンドシーケンスを模倣的に発令したりするほかホストの処理系を変更する必要もあるので、T-NETでは試みていない。

擬似コマンドの発令は4のb, c, 以外にも下図のようにユーザがTS端末を用いるとき必要になる。このとき、



網コマンドはTS端末から投入できるようになるが、コマンドシーケンスの途中でユーザが「居なくな」ったり④LOGOFF以外の手段で使用中止を表明したりすることがある。FEPは新しいユーザへのサービスに備えてLOGOFFまでのシーケンスを擬似的に発生して初期状態に戻らなければならない。そうしないとサーバプロセスから見るとこのホストの応答が遅いのか異常なのか判別できないからである。

d. ユーザは資源の所在に関する知識を網から得るものとする。

ホストの数が多くなるとユーザが目的のファイルのありかを憶えていることは困難である。まして、1の(2)でいう資源の供給が日常行われている網では資源の所在に関する知識は刻々と更新される。文書の郵送だけでは完全を期しがたい。

資源の所在に関する知識を網がユーザに提供するひとつの方法が2.a.2)である。一度おれかがCATしたあとは、その人がBYEするまでDELETEしないとする。遂にファイルの内容を変更できる機会がないという場面も起るので、逆に内容は保証しないのがよい。ユーザ別にCATの履歴を保有するのは複雑だからである。サブコマンド実行中だけ保証することにして、内容の変更は網サービスの外側で行なうとする。網内では登録・削除と複写だけが行われる。VTPにおけるホスト固有コマンドの転送は網のサービスであるが、遠隔ホストにおけるこのコマンドの実行は網が保証しない。これと同じように考えるのである。

このようにして、仮想網においては資源の所在を網が知らせるいわば資源プロトコル(Resource Protocol: RSP)を設けておくこととなる。次のような性格をもつ。

1) ファイルの複写(転送)と登録・削除を区別し、後者のときだけ網全体の管理表を修正する。削除のとき

管理表にフラグをおいてから、充分長時間あとで本体を削除する。CATの応答はファイルの存在を保証しない。

2) CATのパラメータには目的とするファイルのキーワードも用いることができる。こうして、ユーザはホストをあらかじめ指定したり、ファイル名を直接指定したり、所在をあらかじめ言わないでキーワードで候補のファイル名一覧表をとりよせたり、いくつかの方法を使いわけると。

サーバプロセスは管理センタに再結合して一覧表を作成し、CATに応答する。

3) 管理表には各ファイルの使われ方も記しておき、多くのホストから参照されるファイルは複数のホストに重複して登録するときの判断資料を提供する。また各ファイルの本籍ホスト名もあって、プログラムファイルはそこで実行することも奨める。互換性の大きいプログラムには沢山のホスト名を登録されていることだろう。キメ細かい問合せもできるとよい。

4) ユーザは網の使用を終了しようとする前に、自分の発生した資源を整理するための情報を網から得ておくことも重要である。課金の対象になるからである。

課金的方式とも関係するが、数多くのホストに中間結果をばらまいたままにしておくと、支払能力と比較して網側で自動的にゴミ処理をすることも必要だろう。一般にユーザに自分の持物の管理能力を期待してはならないことはよくある例が示す通りである。

仮想化の進んだ網では、このシーケンスは次のようになる。つまり、

④ CAT F

⑤ TRAnSfer F1→F2→F3→F4

ユーザはファイル名をCATで知り、→記号で複写を要求している。これで④で実行可能なプログラムを⑤に複写し

さえすればFmも処理することができることになった。ユーザプロセスは複数の→記号を分解してもよい。この④CATにみられるようにユーザとユーザプロセスは網とインタラクションしながら資源に関する知識を得、その範囲内で資源を使いこなすのである。このことから、ホストの処理系にプロトコル処理系を埋めこんでも定常業務向きになって、網の成長には適応しにくいことがわかる。また管理センタはひとつのデータベースマンであることも容易にわかることである。

### まとめ

電話による計算機網に、電話を取次ぐいわゆる再結合プロトコルを設けると、多くの遠隔ホストに資源が分散している場面にも適応できる。しかし、このとき少数のユーザが網内の多くの資源を占有してしまい、他のユーザからみると網がロックしたようにみえかねない。この障害を除く途として、重複して電話を取次がないようサービスに制約をおく方法を紹介した。

さらに進んで、網を仮想化しようとするとき、資源の種類や網のサービス性能を制約せざるを得なくなる。資源の所在を問わない合わせる資源プロトコルを設けることを検討して、このプロトコルも他のプロトコルと同じように仮想端末プロトコルとファイル転送プロトコル等の基本的なものを拡張して作成できる見通しがあるとのべた。仮想網では全網管理表の制御方式が決まっていればよいのである。

各ホストは全網管理表にまず、遠隔ユーザ用マニュアルを収容しているファイル名を登録すべきだろう。

### 参考文献

1. 川合厚か：公衆電話網による計算機網の研究——高水準プロトコルとその実装——，電総研内部レポート，1978年9月
2. 野口ほか：再結合による分散データファイルの共有化方式について，通信学会，EC76-62，1976
3. H. Kawai, et al : Front End Processing of User Protocol with Public Telephone Network, ICC-78, 3322, 1978
4. 八重樫ほか：再結合による分散情報資源の共有化実験ネットワーク，東北大応用研年報4巻1号，1978年3月
5. たとえば浜中ほか：JIPNETにおけるリソース統合システム，情報学会大会，2E-1，1978