網コマンドのフロントエンド処理

川合英俊（電総研），海老原義彦（筑波大），八重樫純樹，高橋薫，野口正一（東北大）

はじめに　計算機網のそもそもの動機は資源の共用にあるが，異種機の接続にもプログラムやデータの互換性にも問題が多い。しかし他ホストに分散するデータベースをインタラクティブに共用する形は有用性に富むので，この場面に適合するよう次のように問題をしぼって，ミニコンによるフロントエンド処理技術で解決を図ってみた。　1)公衆電話網を簡単に用いる。　2)遠隔データベースを対話型で用いるのに割込みを可能にする。　3)多ホストに分散しているデータベースを各ホスト固有の方法を越えた方法で使用する。　4)ホストのＯＳにあまり影響を与えないで異種網を張る。

電話網に接するフロントエンドプロセッサ（ＦＥＰ）は単なるミニコンで，その固有端末インタフェースを介して接続する。開発段階では使用者端末と共用資源からなるミニホストをＦＥＰに内ぞうしたミニＦＥＰ網を実現した。端末使用者は固有のホストのコマンド言語のほかに，ＦＥＰを操作する網コマンドを使えることになる。その処理もＦＥＰ内で行われる。

1. ＦＥＰ網の構成　図１の通りで，特徴は　1)公衆電話回線を使っていること，2)網制御機能はユーザプロトコルの処理を含めて全てミニコンにつめこんだこと，3)多重リンク制御によってプロトコルを単純化したこと，4)複合コマンドによって多ホスト環境にも適応したことである。設けつつあるユーザプロトコルと，そのサービスコマンドおよび管轄する網資源との関係を表に示した。表の各欄に対応する網コマンドは網使用者にＦＥＰ網が提供する端末言語で，網資源を使用するのに用いる。網コマンドは手順に制約がある。一例を図２に示した。

プロトコルは四層であり，アルゴリズムを附録１，２に示した。そこでの［・・・・・］はパケットの種別名称でその説明は附録3, 4, 5にある。プログラムモジュールはプロトコルに対応したもののほかに，ミニＦＥＰ網のとき網使用者インタフェースまたは網資源インタフェースが，ＦＥＰ網のときこの２つのインタフェースをもつホストとのインタフェースが活性化される。つまり通信中の各ＦＥＰには５個のモジュールが走る[5]。

網コマンドはＦＥＰ対に活性化された（プロトコル）プロセスがユーザプロトコルに沿って実行する。実行に伴う通信は下位プロトコル実行モジュールが下請けするが，論理的には図３のような多重リンクを使い分ける。すなわちＦＥＰ網は次のアルゴリズムを実行する機械である。

Ａｌｇｏｒｉｔｈｍ　ＦＥＰＮＥＴ　／*　網コマンドを解釈実行するためプロセスが活性化され，必要なら，相手ＦＥＰの遠隔プロセスとリンクを通して通信する。*／

Ｍ１.　（初期化）　ｎを１にする。

Ｍ２.　（回線をつなぐ）　手続きCONNECT（附録２）を行う。先天的リンクＬ０ができ，プロセスが活性化される。

Ｍ３.　（プロトコルを実施）　網使用者の選んだユーザプロトコルを判別して，アルゴリズムPROC（ｎ）を行う。PROCはリンク対Ｌｎを作り網コマンドを解釈実行する。

Ｍ４.　（続けるか？）　網使用者が続いてプロトコルを選んだらＭ３.に戻る。

Ｍ５.　（回線を切る）　手続きＤＩＳＣＯＮＮＥＣＴ（附録２）を行い，このアルゴリズムを終わる。

ここでPROC（ｎ）は附録１のPROCESSと類似のもので一般にはリカーシブである。実は新しいプロトコルを選んだときＬ１を使ってＰＩ（protocol initiator）という一対が［protocol segment］を交換し，PROCESSを起動する。リンク対はＬ３まであり，図３で下から順に開設し上から順に解除する。Ｌ３が起きたときＬ２は眠る。

2. プロトコル構造　ユーザプロトコルは網資源の種類ごとに存在し，再結合プロトコルは回線をスイッチするコマンドをもち[1]，複合コマンドを処理する[6]。仮想端末プロトコルは端末の定義コマンドをもつがＦＥＰＮＥＴでは有用でなく用いていない。また附録１のＦ６の上に［ end−of−file ］を書いて戻れという行が必要であり，［ end mark ］はＤＣ１でなければならない。下位に以下の３つの層がある[5]。

1) ホストプロトコル　制御メッセージに関する規約で，ＦＥＰに常駐している網制御プログラム（ＮＣＰ）が実施する。ＮＣＰはホストのＯＳのうち通信に関する部分を請負っているもので，網に関する知識を集約してもっている。Ａｒｐａｎｅｔのいうホストプロトコル[2]と同等で，ＯＳ間の通信規則ともいえ，機能はリンクの開設・解除，メッセージの送出・受入れ，遠隔プロセスへの割込み，割込みの受付，回線の接続・切断である[3]。

2) Ｉｍｐプロトコル　ＦＥＰ間で転送するパケット形式（図４）に関するもので，ヘッダ部の解釈・作成，［ＩＭＰ ｃｏｍｍａｎｄ］の発生・応答，伝送誤りの検出・回復をＩＭＰプログラムが行なう。否定的な受取確認（ＮＡＫ）を４回続けて発したり受けたり，タイムアウトの再送を４回続けたりしたＦＥＰは通信をやめて回線を切ることになっている。回線誤りは最悪で10^{-5}ビットであった。［ here ］を催促しないとか［ term ］の受取確認（ＡＣＫ）に固執しないという変則がある[4]。

3) 回線プロトコル　ＣＣＩＴＴ−Ｖ.２４と互換性がある。どのＦＥＰにもモデムＤ２１２（非同期半２重）とＡＡ型のＮＣＵがある。ＮＣＵからはスイッチのたびにノイズが出るので，最短パケットの半分以下の入力は無視した。

おわりに　ＦＥＰＮＥＴの実現に７名で一年かかったが，設計が長かったので，この仕様でＦＥＰを作るには半人年の規模と思われる。電話はつながるのに１０秒ほどかかるが伝送遅れは知覚できない。割込み処理は少しこみ入っているが，使用上の自然さにはかけがえがない。多重化してはなく，実験規模は小さいが，多ホスト環境で充分使いものになることがわかった。この程度の網制御機能はアセンブラにも高級プログラム言語にもとり入れられることが好ましい。

謝辞　この研究の出発にあたって，九州大北川名誉教授，通信大大泉教授，東北大城戸教授，電総研中山企画室長らの大変な努力があった。研究の途中には筑波大中山教授，電総研黒川，西野両部長の絶ゆまぬ激励があった。電総研の弓場氏は設計およびこの報告に多くの助言をし，ＲＣＣの萩島，三上両氏およびＪＣＳの市野氏は電総研のＦＥＰを製作した。これらの諸氏に深謝して止まない。

参考文献

［１］　海老原：ＦＴＰ仕様書，内部報告，１９７７
［２］　ARPA　PROTOCOL Hand book, Stanford Research Institute, １９７６
［３］　八重樫：ホストプロトコル仕様書，内部報告，１９７７
［４］　高　橋：Ｉｍｐプロトコル仕様，内部報告，１９７６
［５］　八重樫ほか：分散形ファイル共有化のための実験ネットワークシステム，信学全大，Ｓ３−１１，昭５２−８
［６］　海老原：再結合プロトコルについて，信学全大，Ｓ３−１２，昭５２−８

表　網コマンドレパートリ

Network user commands				Network resource managed by the User-protocol
Network access command	User-protocol selection commands	Service commands: Resource control commands	Service commands: Transfer commands	
° log on to a host ° test reply and exit from the protocol ° interrupt the command ° log off	° VTP (Virtual Terminal Protocol)	° define a NVT (network virtual terminal) ° through the NVT connect the remote host ° bye	° terminal commands language of the remote host	° the remote host
	° FTP (File Transfer Protocol)	° list catalogue ° define a file ° register a file ° release a file	° store the file ° retrieve the file	° network user files ° catalogue file
	° RCP (Reconnection Protocol)	° update the command file ° switch the line	° submit the command file ° sense the status file	° command file for a third host ° status file
	° DBP (Data Base Protocol)	° select a data base	° access language of the data base	° the data base ° the data base management system
	° NMP (Network Measurement Protocol)	° define a measurement file	° retrieve a contiguous group of data in the measurement file	° measurement file ° user profile of command sequence

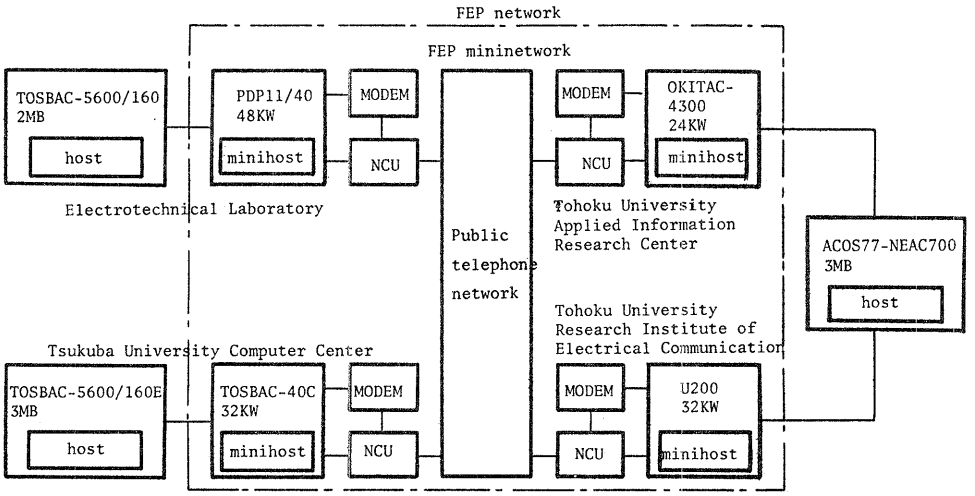
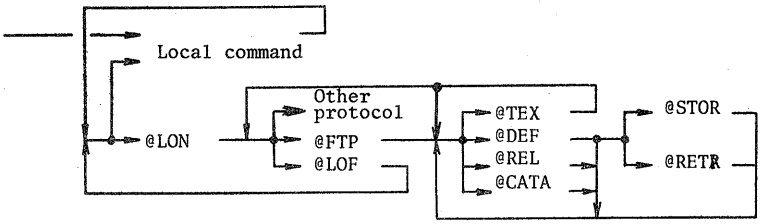


図１　３ホストＦＥＰ網と４ミニホストＦＥＰ網



Note - @LON : log on
@LOF : log off
@FTP : file transfer protocol
@TEX : test a reply and exit from the protocol
@DEF : define a file
@STOR: store the file
@RETR: retrieve the file
@CATA: list the catalogue file
@REL : release the file

図２　ファイル転送プロトコルの網コマンド許容順序

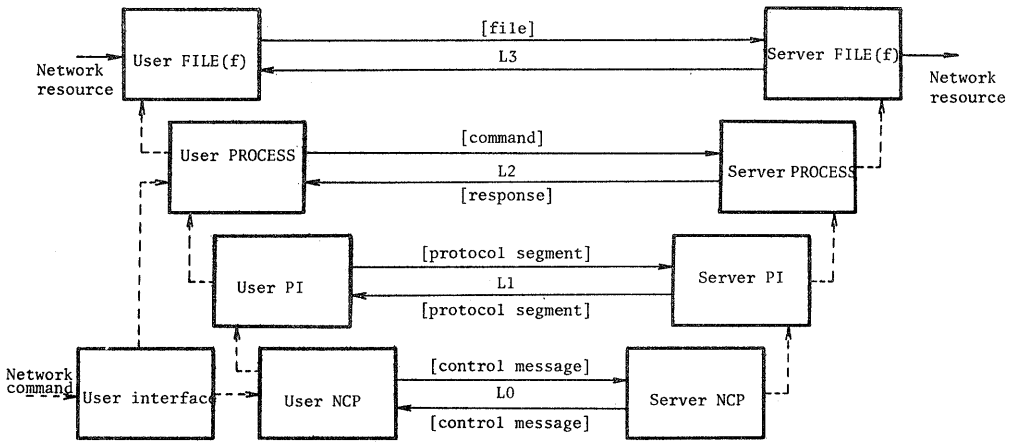


図３　ユーザプロトコルに則って網コマンドを実行するのに用いる４対のリンク

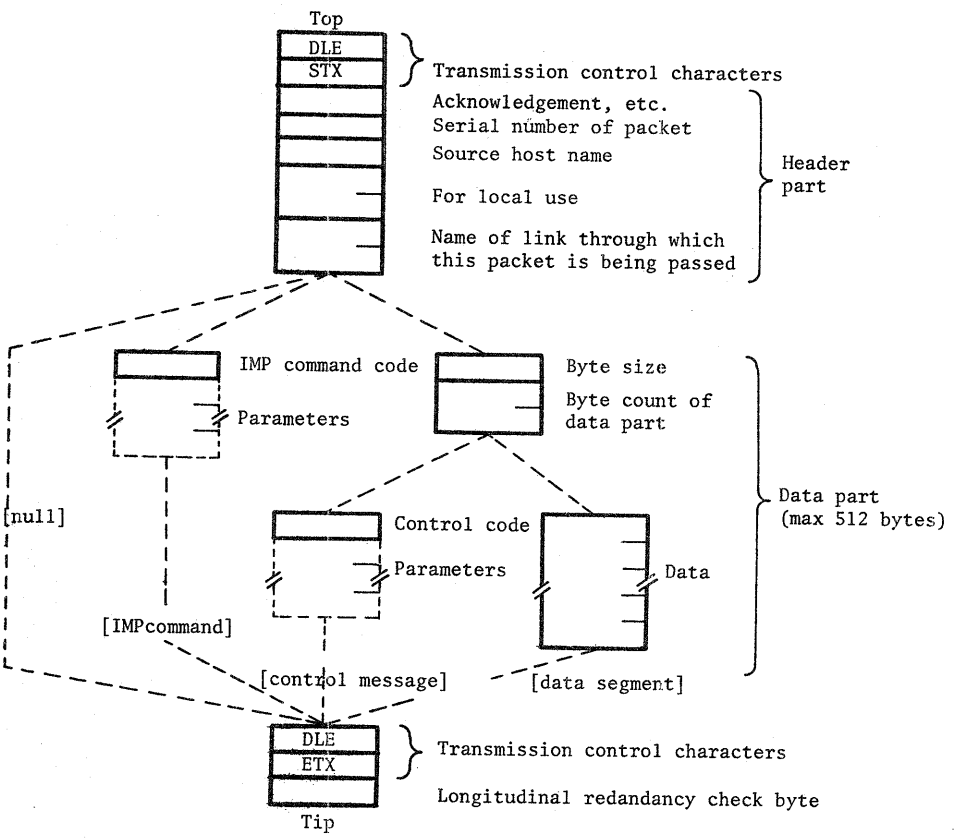


図4　パケット形式

附録 D　プロトコルインタープリタ（PI）のアルゴリズム

Algorithm PI /*　回線がつながったとき活性化されたユーザPIが，ユーザプロトコル選択コマンドを解読，通知，相互確認したのち，後刻網使用者が投入する網コマンドの解釈実行に備えてプロセスを活性化する。サーバPIは，I2のない，かつWriteとReadを交換したアルゴリズムを実行する。*/

I1. (Open)　Call the Procedure OPEN(I1).

I2. (Get command)　Accept new terminal command.

I3. (Write the protocol)　Write the [protocol segment].

I4. (Read reply)　Read a [protocol segment].

I5. (Activate process)　Activate process.

I6. (Close)　Call the Procedure Close(L1) and the algorithm terminate.

付録１　プロセスのアルゴリズム

Algorithm PROCESS /*　ユーザプロセスがユーザプロトコルに従って網コマンドを解釈する。サーバプロセスも似たアルゴリズムを実行するが，それは次の諸点で異なり双方同時 "read" によるシステムロックを避けている。　1)　P 4 と P 5 はない。　2) readとwrite, user と server, interrupt と interruptedを互いに交換する。*/

P1. (Open)　Call the procedure OPEN(L2) and reset the flag INTRP.

P2. (Closed?)　If a link of the link pair L2 is closed, go to P12.

P3. (Advance)　Read an [end-of-response] to call user's attention to typing in an additional network command.

P4. (Get command)　Accept new terminal command.

P5. (Is the command 'exit'?)　If the command is 'exit from the User-protocol', go to P12.

P6. (Write the command)　Write the [command] to the server process.

P7. (Interruption detected?)　If interrupted, set the flag INTRP and go to P11.

P8. (Need to interrupt?)　If the flag INTRP is reset and an interruption is needed, for instance the user has requested it, generate an interruption. If the flag INTRP is set, reset it.

P9. (Read reply)　Read a [data segment].

P10. (Is another link needed?)　If the [data segment] is[prompter], set f depending on the command gotten in step P4 and call the procedure FILE(f). Otherwise returen back to P2.

P11. (Write reply)　If the flag INTRP is set, write a [reply segment] and go back to P2. Otherwise, go back to P8.

P12. (Close)　Call the procedure CLOSE(L2) and the algorithm terminate.

Procedure FILE(f) /*　f が "read" の意味なら，このアルゴリズムでプロセスは転送コマンドを解釈している遠隔プロセスからファイルを受取り，遠隔プロセスはreadと writeを交換したアルゴリズムを実行する。f が "write" の意味なら，以上のコメントで，「遠隔プロセス」と「プロセス」を交換し，　"read"　を "write"　に変えればよい。*/

F1. (Open)　Call the procedure OPEN(L3).

F2. (Interruption detected?)　If interrupted, set a flag INTRP and go to F6.

F3. (Need to interrupt?)　If an interruption is needed, generate an interruption and go to F6.

F4. (Read file)　Read a [data segment].

F5. (End of file?)　If the [data segment] is not an [end-of-file], go back to F2.

F6. (Close)　Call the procedure CLOSE (L3) and return.

附録 2　網制御プログラムのアルゴリズム

Algorithm NCP /* NCP はホストのOSの中で通信に関する基本的な機能を分担するもので，リンク（論理的通話路）を開設・解除したり，[data segment] を送出・受入したり，遠隔プロセスに割込んだり割込まれたりしたり，電話回線をつないだり切ったりする。FEP ネットワークではミニコン FEP に実装された。*/

N1. (Initialize) Get into the initial state by logging onto the time sharing system of own hosts.

N2. (Connect a line) Call the procedure CONNECT and wait until the telephone line is connected and an appropriate process is activated.

N3. (Get new request) Get a new request from the process.

N4. (Is the request disconnection?) If the request is the disconnection of the telephone line, call the procedure DISCONNECT and then the algorithm terminate.

N5. (Do algorithm HOST) To take an action according to the Host-protocol, call the procedure HOST (request, LN) and then go back to N3.

Algorithm HOST (request, Ln) /* NCP は，ホストプロトコルに従って遠隔 NCP と通信するために，このアルゴリズムを実行する。このアルゴリズムはプロセスと遠隔プロセスの同期化も行なっている。 */

H1. (Is the request to open?) If the request is to open a link pair Ln, call the procedure OPEN(Ln) and return.

H2. (Is the request to close?) If the request is to close the link pair Ln, call the procedure CLOSE(Ln) and return

H3. (Does the link pair exist?) If the link pair Ln does not exist, return unsuccessfully.

H4. (Is the request interruption?) If the request is to generate an interruption to the remote process, send [ins] and return.

H5. (Try to write) If the request is write operation, wait for receiving a [segment].

H6. (Try to read) If the request is read operation, send [num] and then receive a [segment].

H7. (Error detected?) If the received [segment] is [err], stop the system.
If any error is detected in the [segment], send [err] and stop the system.

H8. (Interruption detected?) If the received [segment] is [ins], notify it to the process and return.

H9. (Advance to read/write) If the received [segment] by the step H5 is [num], sed a [data segment] of the process and return. Otherwise, as the received [segment] is a [data segment], hand over it to the process and return.

Procedure OPEN (Ln) /* ユーザ NCP がリンク対 Lnを作る。サーバ NCP に対してはこのアルゴリズムの'send'と'receive'を交換すればよい。 */

01. (Request) Send [rts] having n through the link pair L0 being set a priori between a pair of NCPs, and receive a [control segment].

02. (Not accepted?) If the received [control segment] is [cls], return unsuccessfully.

03. (Made) If the [control segment] is [str], a link to receive is made. Otherwise, send [err] and stop.

04. (Make pair) Receive [rts] having n and send [str], then a link to send is also made, and return.

Procedure CLOSE (Ln) /* ユーザ NCP がリンク対 Lnを解除する。サーバ NCP は'send'と'receive'を交換したアルゴリズムを実行すればよい。*/

C1. (Request) Send [cls] having n and receive a [data segment], to close a link to receive.

C2. (Error?) If the received [data segment] is not [cls], send [err] and stop.

C3. (Accepted) Receive [cls] having n, then send [cls], to close a link to send and return.

Procedure CONNECT /* ハードウェアを扱っているプログラムであるドライバが公衆電話回線をつなぎ，通信プロトコルを処理する IMP は遠隔ホストの名称を検査する。後刻与えられる網コマンドを解釈するためのプロセスが(P I により間接的に)活性化される。*/

C1. (Initialize) Get into the initial state and wait for a hardware trap which is ringing of a telephone line or making call by the user.

C2. (Ring detected?) If the telephone line is ringing, hook off the telephone. Otherwise go to C4.

C3. (Exchange names) Receive [here], then send [+here], activate a server process and return.

C4. (Make a call) Dial out and wait a trap of hooking-off by a remote FEP.

C5. (Exchange names) Send [here], then receive [+here], activate a user process and return.

Procedure DISCONNECT /* IMP が通信を終えるのに IMP コマンドを遠隔 IMP との間に交換し，ドライバは公衆電話回線を切る。 */

D1. (Termination detected?) If [term] is received, send [+term], deactivate process and go to D3.

D2. (Exchange term) Send [term], receive [+term] and deactivate a process.

D3. (Hanging up) Hand up the telephone line and return.

附録 3 ユーザプロトコルのデータメッセージ

注 附録 3 ～ 5 では次の記号を用いる。

記号	意味
:=	等しい。
[...]	パケットの内容の種別名称。
!	または。
/* ... */	説明。
$	連結。
[...]$ ... $[...]	くり返しまたは何もない。

[data message] := [command]![response]![file]

:= /* 多重パケットでもよい。[data segment] から合成される。[end mark] をもつ。*/

[data segment] := [command]![prompter]]![reply segment]![end-of-response]![file segment]![protocol segment]![end-of-file]

:= /* 必らずデータ部を伴った単一パケットである。プロセスの read 又はwrite操作の要求一個に対応する。 */

[command] := /* 網使用者が発生したサービスコマンドで，自分の側のユーザプロセスから遠隔サーバプロセスに送られる。末尾は伝送制御文字 DC1で終わっていて，続く[data segment] の送信権を放棄したことを意味する。 */

[response] := [prompter]$[reply segment]$[end-of-response]
:= /* [command]に対する応答で主としてサーバが発する。2ないし3パケットよりなる。*/

[prompter] := /* リンク対の開設を要する[command]に対する肯定的確認応答で，この1対のリンクがいらないときには必要ない。*/

[reply segment] := [reply code]$[text]

[reply code] := /* コマンドを実行した結果を報告する2バイトコードで，実行の首尾を表わしている。割込みを受けたことを知らせることもある。*/

[text] := /* [reply code]に関する詳しい説明で，自然言語による。ユーザプロセスから発せられるときは割込受け応答にほかならず DC1 で終端している。*/

[end-of-response] := /* サーバプロセスの発する網使用者への知らせで，次のコマンドを受け入れる用意ができたことを表わし，DC1 で終端している。*/

[file] := [file segment] $.....$ [file segment] $ [end-of-file]
:= /* レコードかまたは非レコード型のバイト列であり，[end mark]を末尾にもっている。ホスト自身の固有のファイルへ変換可能である。*/

[file segment] := [records] $.....$ [record]![non-record]

[record] :=/* 80字以下の長さのバイト（8ビットコード）列に末尾符号としてレコード区切りコードEOR（オクタルで301）をつけたもの。これからなるファイルをレコード型という。*/

[non-record] :=/* 末尾に必ずしもEORをもたない。1パケット（512バイト）以下の長さのバイト列。*/

[end-of-file :=[file segment]$[end mark]

[end mark] :=/* バイト列の終了符号で伝送制御文字DC1かあるいは次の4文字である。EOFコード（オクタルで300），2バイト長の全バイト数，EOF。*/

[protocol segment]:=/* 網使用者が選択したユーザプロトコルの名称（1バイト，固定）とそれを処理するプロセスのソケット番号（4バイト，一意的）とからなる。PI(protocol initiator)が用い，[protocol segment]に対する応答にも使う。PI自身のソケット番号は固定である。*/

附録4　ホストプロトコルの制御メッセージ

```
[segment]          :=[data segment]![control segment]

[control segment]  :=[rts]![str]![cls]![ins]![num]![err]
                   :=/*  ＮＣＰがリンク対やプロセス対を制御するのに用いる網制御メッ
                         セージでパラメータをもつことはあるが，データ　はもたない。
                         先天的リンクＬ０を通る。*/
[rts]              :=/* [data segment]を送信するリンクの開設を要求する。パラメータはリ
                         ンク名と既知の両端ソケット番号である。*/
[str]              :=/*  受信した [rts] に対する肯定的応答。パラメータは同じ。*/
[cls]              :=/*  リンクの解除を要求するか [cls] に応ずるか [rts] に否定的応答
                         をするかする。パラメータは両端のソケット番号。*/
[ins]              :=/*  パラメータで指定した名称のリンクにつながる遠隔プロセスに割
                         込みを発する。*/
[num]              :=/*  続いて [data segment] を受信できることを遠隔 NCP に通知するもの
                         で，リンク両端のプロセス対が同期をとるのに使われる。パラメ
                         ータはこの [data segment] が通過するリンク名である。*/
[err]              :=/*  受信した [segment] に形式的（順序と各フィールド）誤りがあった
                         ことを通知し，デバグのためにシステムを止める。*/
```

附録5　ＩｍｐプロトコルのＩＭＰコマンド

```
[packet            :=[segment]![IMPcommand ]![null]
[null]             :=/*  データ部をもたない受取り確認（ＡＣＫ，ＮＡＫ）で，タイムア
                         ウト検査によるトラップを避けるのに使われる。*/

[IMPcommand]       :=[here]![term]![echo]![IMPreply]
[here]             :=/*  互いにＦＥＰ内にあるＩＭＰ対間の確認で，パラメータは双方の
                         ホスト名称である。*/
[term]             :=/*  公衆回線を切りたいという意志表示で，パラメータはない。*/
[echo]             :=/*  パラメータ部のテスト用データをそのまま元のＩＭＰに送り返し
                         てほしいという要求を表わす。*/

[IMPreply]         :=[+here]![+term]![+echo]![error]
[+here]![+term]![+echo]:=/*それぞれ[here]![term]![echo]に対する肯定的応答である。*/

[error]            :=/*  受信した[IMPcommand]に対する否定的応答で，形式的誤りを検出し
                         たことを表わす。[error]を送受信すると回線は切断され，デバグ
                         のためシステムは止まる。パラメータは誤りの発見個所や判断理
                         由を示している。*/
```