

科内ネットワークシステム NITNET の開発と応用

丹羽敏行 井口竜治* 藤社 彰*

情報工学科

(1979年 9月 12日受理)

Implementation of the Inhouse Computer Network System NITNET

Toshiyuki NIWA, Tatsuji IGUCHI and Akira TŌSHA

Department of Information Engineering

(Received September 12, 1979)

This paper describes a new computer network system. With multi-terminal structure it extends RACS system, which serves mainly for RJE. NITNET has four Terminal Hosts and Central Host, and all their resources can be shared together.

In NITNET, we can directly edit the program in the source library file of Central Host by using On-line Editor which serves under the Network Control Program. Therefore, the turn-around time is greatly reduced. We can also easily transfer the file data, in which spaces are efficiently compressed, and then the transmission time can be saved at 20~40% as compared with RACS system.

1. まえがき

本学情報工学科では、教育用中型機を始めとして、研究用として多数のミニコンやマイコン・システムが、各研究室で独立に運用されてきた。それぞれのシステムには特色あるハードウェアやソフトウェアが用意されていて、以前よりそれらシステム・リソースの共有化が強く望まれていた。

これまでに、我々は RJE を中心としたリポート・バッチ・システム RACS¹⁾ を開発し運用してきたが、当初より端末局の増設が課題であった。

そこで今回、多端末化を契機に、端末局と中央局間は勿論のこと、任意の端末局間でのリソースの共有をも可能とするよう、新たに、ネットワーク・システム NITNET を開発することになった。

ネットワーク化に当っては、KUIPNET²⁾ や N-1³⁾ などの他システム⁴⁾⁵⁾ を参考にし、RACS での成果を加味して、本情報工学科固有の研究に対する操作性を最も重視している。

本システムの特徴を挙げると、次の5点にまとめることができる。

(1) オンライン・エディタ

RACS では、デバッグの度に全ソース・プログラムを入力し直すことが必要で、伝送速度の低さとあいまってターン・アラウンド・タイムが増大し、プログラム開発の大きな支障となっていた。そこで、NITNET では NCP (Network Control Program) の直接管理下で動作する独自のオンライン・エディタをサポートした。

このエディタは、端末局から会話形式で中央局のユーザ・ファイルを直接参照して、行及び文字単位の編集を行うと共に、中央局へバッチ処理を依頼することができる。

(2) プロトコルの改良とデータの圧縮

同期のためのメッセージを少くすると共に、メッセージ中のヘッダ部を短縮したり、エラーメッセージのコード化や独自の方法による入出力テキスト中のスペースの圧縮を行って、可能な限り伝送効率の向上を目指した。

(3) ファイル伝送

任意の端末局間でのファイル・データの伝送を、

* 名古屋工業大学院工学研究科

簡単な操作で可能とし、RACS の場合のようなユーザ・プログラムの存在を不必要とした。またその際、ユーザの指定によるコード変換も可能にしている。

(4) 簡略化ジョブコン

RACS の場合と同様に、簡略化ジョブコンの使用によって、入力 RJE ジョブ・デッキの構成が簡単にな

るばかりか、簡略化ジョブコンそのものがオンライン・エディタのデータとしても扱ひ得るので便利である。

(5) オペレータ間通信

各端末局のオペレータ間でのメッセージ交換を容易にしている。

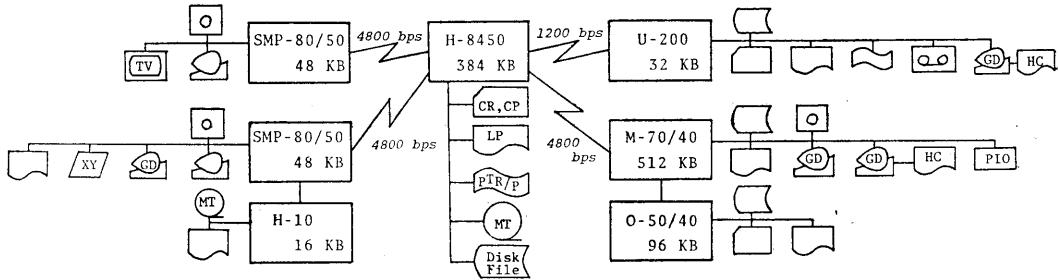


Fig. 1 Hardware configuration of NITNET

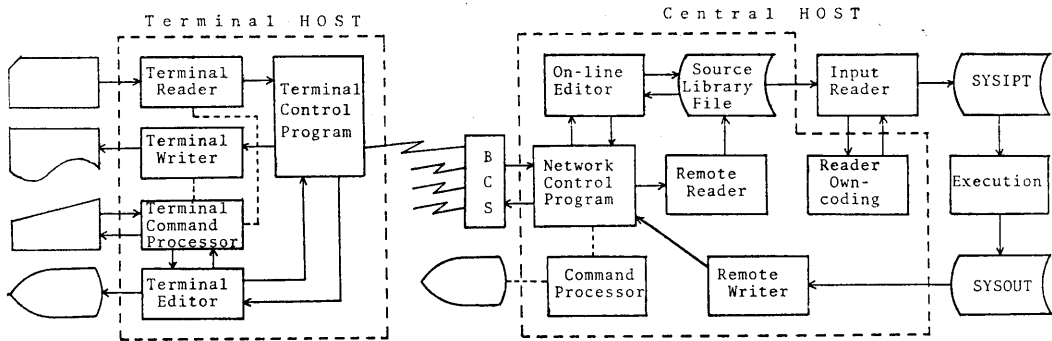


Fig. 2 Software structure and the data flows of NITNET

2. NITNET の構成

2.1 ハードウェア構成

今年度中に増設予定のミニコン・システムを含めて、NITNET のハードウェア構成を Fig. 1 に示す。

各端末局と中央局 H-8450 は 4 線式の専用回線で接続されており、U-200 局は半 2 重 1200bps の調歩同期式を、他の 3 局はいずれも半 2 重 4800bps の SYN 同期式を採用している。

2.2 ソフトウェア構成

NITNET のソフトウェア構成を Fig. 2 に示す。

図中の破線で囲った部分が、今回開発したプログラムであり、ほとんどが RACS で得られたデータを基に、新たに設計された。なを、中央局側の BCS とは、メーカ提供の通信管理プログラムであり、コントロール・プログラムに組み込まれ、ユーザ・プログラムからのマクロ・コールによって特権モードで動作する。

2.3 処理形態

NITNET でサポートされるジョブ処理の形態は次の 3 つに大別される。

(1) RJE の場合は、任意の端末局よりジョブを入力して、リスト出力とパンチ出力をそれぞれ任意の局へ出力することが可能である。なを、現時点では、RJE ジョブの処理は中央局でしか実行できない。

(2) オンライン・エディタを使用の場合は、自端末局より編集データを入力して、自端末局への出力のみが許される。

(3) ファイル伝送を指定する場合は、現時点では、中央局を除いて端末局間のみで可能である。

以上のように、RACS に比べて、可能な処理形態はより多様となり、さらに操作性も向上している。

2.4 処理手順

RJE ジョブの処理は RACS の場合とほぼ同様である

ので、ここではオンライン・エディタによる処理手順の概要を Fig. 2 を用いて説明する。

(1) '/RJEDIT' コマンドによって、端末局及び中央局のエディタが起動され、初期化が行われる。

(2) ソース、ライブラリの作成の場合は、端末局エディタは '//PARAM' コマンドによって指定されたファイルからデータを読み込み、ブロッキングして、端末制御プログラムを通じて回線へ送出する。中央局エディタは NCP を通じてデータを受け取り、デブロッキングして、ソース・ライブラリ・ファイルに書き込む。

```
//JOB
//RJIN
!JOB J599ABCDEF,01,G
!DTF OUT=WR,3
!FORTRN
:
!RUN
:
!END
!JOB
:
!END
//PARAM OUT=DD00/DFILE/A,ENQ=YES
//RJOUT
//END
```

(a) RJE

```
//JOB
//PARAM INPUT=CT00/CFILE
//PARAM WF=FD00/FFILE,DST=2
//WRITE
//END
```

(b) File transmission

```
//JOB
//RJEDIT
/LOAD TESTPROG
!JOB J599ABCDEF,02,A
!FORTRN
:
!END
/FEND
/EDIT TESTPROG
/MOVE 2500
/CHAR F6,IABC,D4
/DELETE 3200,3600
/START TESTPROG
/END
//END
```

(c) Editing

(3) プログラム及び実行データの修正の場合は、まず '/EDIT' サブコマンドにより、修正対象となるプログラムを指定した後、修正内容に従ったサブコマンドを発行して、端末局及び中央局のエディタと会話しながら編集作業を進める。

(4) 編集の終わったプログラム及び実行データは、端末局からの '/START' サブコマンドにより、中央局の非常駐入力リーダーが起動され、入力スタック・エリア SYSIPT に登録されてバッチ処理の対象となる。

3. 端末局側から見た NITNET

3.1 入力ジョブの構成

端末局より入力して、リスト・データを中央局に出力し、パンチ・データを端末局に出力するRJEのカード・デックの例を Fig. 3(a) に示す。

ここで、入力ジョブは一まとまり毎に '/RJIN' ジョブコンの後に入力し、次のジョブコンで入力終了となる。したがって、中央局本来のジョブコンは端末局のものと区別するため '!...' の形の簡略化ジョブコンを使用すると便利である。

ファイル伝送の場合は、Fig. 3(b)で示されるように、直接コマンドのキーインによっても簡単に行うことができる。

3.2 ユーザ・コマンド及びジョブコン

端末局で用意されているユーザコマンド及びジョブコンの一覧を Table 1 に示す。

ここで、'/RJIN' のみは機能上 SYSIN 装置より入力しなければならないが、他は全てコンソールより直接キーインすることによって起動することができる。

3.3 簡略化ジョブコン

入力された RJE ジョブ中の簡略化ジョブコンは、入力リーダー・オウン・コーディング・プログラムにより正規のジョブコン群に展開される。その際、利用者コードその他のチェック、プライオリティと要求メモリ容量の設定、ジョブ名の附与、デバッグ用パラメータの設定、コンソールへのコメント表示なども行われる。

簡略化ジョブコンとその展開形を Table 2 に示す。

4. 中央局の NCP

4.1 NCP の構成

NITNET では、端末局での制御プログラムを簡単にするため、NCP を中央局側で持ちネットワーク全体を管理している。したがって、中央局は RJE 局と IMP (Interface Message Processor) の合体したものとみることが出来る。

Fig. 3 The example card deck for NITNET from the Terminal HOST

Table 1. NITNET user commands for the Terminal Host

	Operation	Operand	Action
//	NET	[parameter]	Open the Terminal Host
//	PARAM	[DST=...] [, IN=...] [, OUT=...] [, ENQ=...] [, LIST=...] [, RF=...] [, WF=...]	Identify the destination Host Identify the input file of source Host Identify the output file of source Host Request for the instant output Request for the listing of input/output data Identify the input file of destination Host Identify the output file of destination Host
//	RJIN		Send the input job
//	RJOUT	[job-name[, job-number]]	Receive the output job
//	RJCAN		Cancel the operation
//	RJCNT		Continue the operation
//	RJEDIT		Start the editor
//	READ	{host-code, devicenumber, file- name, code}	Send the file data
//	WRITE	{host-code, device-number, file- name, code, mode}	Receive the file data
//	D	{JOB ...} {MEM...} {NET ...}	Display the current job status Display the current memory allocation Display the current network status
//	E	{SRT... STP... HTJ... DLT job-name[, job-number] RLS job-name[, job-number]}	Start the job stage Stop the job stage Halt and delete the job stage Delete the output job in SYSOUT file Release the output job from the hold status
//	RJCOM	comment	Send the comment
//	RJWAIT		Wait the request for some Host
//	BYE		Close the Terminal Host

Table 2 Simplified Job Control cards and their original formats

Simplified Job Control card for JYOHO H-8450 SYSTEM		Original formats under EDOS-MSO	Notes
!JOB aabbccccc, nn, 1[, t]		//aacc#nn JOB JBN=... //SYSUT1 DTF... //SYSUT2 DTF... //SYSUT3 DTF... //SYSOML DTF... //SYSLIB DTF...	Priority is variable according to the jobnumber and the jobclass 1
!RUN	!LINK	// {LNKEDT} {LNKEX} PROG aaccnn INCLUDE SYSUT1	Values of TIME and LST are subst- ituted corresponding to jobclass 1
	!EXEC	// EXEC aaccnn, UT2 TIME=... LST=...	
!END		// END	
!#xxxxxxx		// CATLG #xxxxxxx	
!yyyyyy		// yyyyyy	

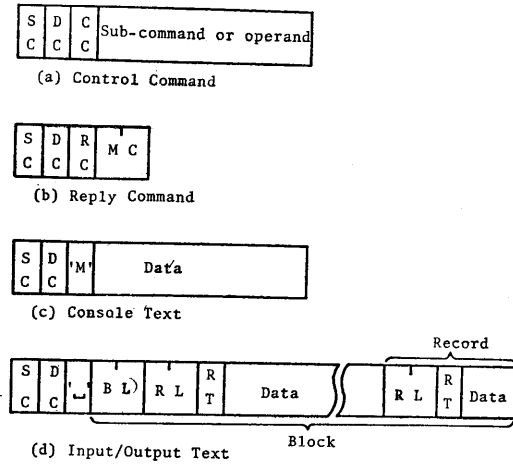


Fig. 4 Data formats

SC: Source HOST Code DC: Destination HOST Code
 CC: Command Code RC: Reply Code
 MC: Message Code BL: Block Length
 RL: Record Length RT: Record Type

NCP の機能は次のように分類することができる。

- (1) 全端末局及び中央局と回線の状態管理
- (2) 伝送メッセージのチェックとフロー制御
- (3) オンラインエディタの起動
- (4) RJE ジョブのバッチ処理依頼及び処理結果の管理

4.2 伝送メッセージ

回線を通じて伝送されるデータ (メッセージ) には、制御コマンド、応答コマンド、コンソール・テキスト、入出力テキストの4種があり、そのフォーマットを Fig. 4 に示す。

応答コマンドには0次肯定、1次肯定と否定の3種があり、データ伝送における同期をとっている。0次肯定は3バイト長、1次肯定と否定の場合はメッセージ番号が含まれて5バイト長になっている。

コンソール・テキストは、各局の状態表示データあるいはオペレータ間の通信データであり、受信した局のコンソールに出力される。メッセージ長は53バイト以下である。

Table 3. Control commands between the Terminal Host and the IMP (Central Host)

Command	Operand	Action
ABEND		Abandon the connection
STATUS	$\begin{Bmatrix} D \text{ JOB} \dots \\ \vdots \\ E \text{ SRT} \dots \\ \vdots \end{Bmatrix}$	Start the Key In/Out simulation
INPUT OUTPUT READ WRITE BEDIT	[job-name [, job-number]] device. code. file-name device. code. mode. file-name	Start the remote reader Start the remote writer Start the terminal reader for the file data Start the terminal writer for the file data Start the editor
FILE EOF CANCEL CONTINUE		Send the input job or the file data End of the input job or the file data Cancel the command Continue the command

Table 4. Circuit connection

Item	H-8787-15=FACOM U-200	H-8788-55=SORD SMP-80/50
Communication system	Half-duplex premises line	Half-duplex premises line
Transmission code	JIS 7 bit code	JIS 8 bit code
Transmission rate	1,200 bps	4,800 bps
Synchronization	Asynchronous with the start-stop bit	Synchronous with the SYN character
Link access	Contention	Contention
Transmission mode	Normal mode	(1) Normal mode (2) Transparent mode ●
Error control	(1) Horizontal parity check BCC (Block Check Character) (2) Vertical parity check	CRC (Cyclic Redundancy Check) $X^{16}+X^{12}+X^5+1$
Line control program	BLCOM	BLCCS
Transmission message format	Single block only STX...ETX	Single block only (1) STX...ETX (2) DLE. STX...DLE. ETX
Transmission control character	STX, ETX, EOT, ENQ, ACK, NAK	STX, ETX, EOT, ENQ, ACKO, ACKI, NAK, DLE. STX, DLE. ETX, DLE. ENQ, SYN

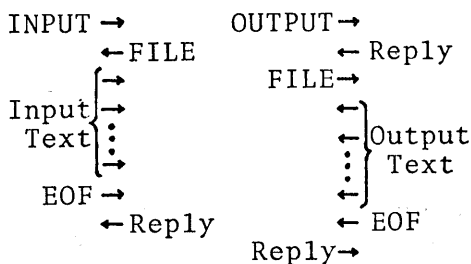


Fig. 5 Control command sequence for RJE

入出力テキスト中のデータは、スペースの圧縮が行われた後ブロッキングされる。メッセージ長は1000バイト以下である。

4.3 制御コマンド

制御コマンドは応答コマンドと共に用いられ、コンソール・テキスト及び入出力テキストの伝送を制御する。制御コマンドの一覧を Table 3 に示すと共に、RJE ジョブの入出力時におけるコマンド・シーケンスを Fig. 5 に示す。

4.4 伝送制御

端末局と中央の接続方式は Table 4 に示すように2種

あるので、それぞれ異なった伝送制御手順を採用している。中央局 H-8450 でサポートされる BLCOM 及び BLCCS といった手順⁶⁾を変更することは不可能なので、端末局側はそれぞれ BLCOM あるいは BLCCS 手順に準じたものを用意しなければならなかった。

5. オンライン・エディタ

5.1 概要

オンライン・エディタは次の6つの機能を有している。

- (1) 端末局より入力されたソース・プログラムを中央局のソース・ライブラリ・ファイルに登録する。
- (2) ソース・ライブラリ・ファイル上のプログラムを、端末局よりのコマンドにより修正を行う。
- (3) 登録されているプログラムを端末局に退避する。
- (4) 登録されているプログラムのリストを端末局に出力する。
- (5) 登録されているプログラムのシーケンシャル・ナンバをリナンバする。
- (6) 登録されているプログラムのバッチ処理をNCPに依頼する。

簡略化ジョブコン付きのプログラムを登録し、修正を行った後実行させる場合のカードデッキの例を示す。

Table 5. Online-editor commands for the Terminal Host

	Operation	Operand	Action
//	RJEDIT		Start the editor
/	LOAD	program-name[,D]	Write the program to the source-file
/	LIST	[program-name]	Display the program list
/	RENUMBER	program-name	Renumber the sequential-number
/	SAVE	program-name	Save the program
/	START	program-name	Execute the program
/	OMIT	program-name[,D]	Delete the program
/	EDIT	program-name	Edit the program
/	FEND		End of the file
/	END		Halt the editor
/	DELETE	seq[,seq']	Delete the statements
/	INSERT	[seq]	Insert the statements
/	MOVE	seq	Advance the pointer
/	DISPLAY	{[seq]}	Display 16 statements
/	RENAME	seq, seq', S1, S2	Change S1 for S2 between seq and seq'
/	CHAR	Fn, Dn, Bn,...etc	Correct the characters
/	SEARCH	string	Search the string
/	BACK	seq	Retreat the pointer

5.2 ファイル形式

ソース・ライブラリ・ファイルの構成は、入力リーダーを用いて読み込みができるように、メーカー規定のフォーマットに従っていて、80バイト/レコード、4レコード/ブロックで、ブロックの先頭に8バイトのポインタ部があり328バイト/ブロックとなっている。各々のブロックはポインタによりリスト形式になっているが、本エディタでは '/BACK' コマンドの実現のため、予備エリアを使用してポインタを増やし、双方向性リスト形式としている。'BACK' コマンドは、プログラムの一部をそっくり移動させたり、交換する場合に便利である。

実行データに対しては、シーケンシャル・ナンバを付けないから、ファイル登録時に、プログラム部とデータ部とに分けられる。

5.3 エディタ用コマンド

エディタ用コマンドの一覧を Table 5 に示す。このうち '/DELETE' から '/BACK' までは '/EDIT' コマンドに従属するものである。各コマンドは端末局のコンソールより入力することによって、会話形式で編集作業を行うことができる。

6. エラー処理

データ伝送中に生じたエラーのうち、回復可能なエラーについてはコマンドの保留を行い、他回線の処理に移

る。保留されたコマンドは CONTINUE コマンドによって再開することができる。回復不能なエラーについては、中央局コンソールにエラーメッセージを表示し、システムを停止させる。

回復可能なエラーメッセージは中央局発行が40種（端末局へ伝送されるのは19種）あり、端末局発行のものが37種用意されている。

7. 考 察

7.1 NITNET の評価

RACS から NITNET の運用に切り換えたばかりで、制御プログラムを開発中の端末局もあり、新システムでの応用はまだ少ない。

しかし、比較のため RACS での応用プログラムを NITNET に適用して次のような結果を得ている。

(1) 通常のコmpایل、リンクのリスト出力で約20%の時間が短縮できた。また、行番号つきプログラムのアセンブリ・リストのように、スペースの圧縮効果が大きく表われる場合は約40%の時間が短縮されている。

(2) オンライン・エディタの使用の場合、U-200 端末局 (1200bps) では、削除や置換に約4秒/100ステートメント、挿入で約40秒/100ステートメントの応答時間が得られている。ミニコンのカード・リーダーから入力の場合は100~200秒/100ステートメントであるから、エディタの使用効果は大きい。

(3) 現時点でのシステムの大きさは、中央局でエデ

ィタを含めて約 31KB, 端末局で約 25KB と, コンパクトに構成されている。しかし, 中央局側では BCS やリモートリーダーのためのメモリが必要なので, RJE ジョブに対するユーザ領域はバッチのみのシステムに比べて約 80KB の減少になっている。これは, RACS の場合の 60KB の減少に比べると, かなり良い値と思われる。

7.2 システムの拡張

現時点での NITNET では, メモリ所要量を抑えるため, サポートできなかった機能が2つある。

すなわち, SYN 同期方式での透過伝送を採用すれば, 画像データや医用時系列データの伝送が約 1/2 の時間で可能となる。また, ファイル伝送を中央局でもサポートすれば, より便利で効率も良くなる。

対策としては, NCP の分散化が考えられるが, 最終的にはメモリの増設が望ましい。

なを, M-70 端末局での RJE のサポートも今後の課題である。

8. あとがき

情報工学科での研究・教育が今後ますます多様になるに従い, 計算機間の結合も多様化し不可欠のものとなる

時, NITNET 開発での有形無形の成果が活用できるものと思われる。その意味において, NITNET は不満足な現状に対する一解決策であると同時に, 最新のハードウェアとソフトウェア・システムへのリプレイスの必要性を強調するものでもあると考えられる。

最後に, 日頃熱心に御討論いただく名古屋大学工学部情報工学科の福村教授と吉田助教授, ならびにファコムハイタック (株) の楠葉氏と伊藤氏に感謝致します。

文 献

- 1) 丹羽敏行: リモート・バッチ・システム RACS の開発, 名古屋工業大学学报, 30, p. 333 (1978)
- 2) 坂井, 田畑, 大西, 北沢: インハウス・コンピュータ・ネットワークと HOST コンピュータ, 情報処理, 15 (12), p. 948 (1974)
- 3) 文部省科学研究費特定研究「広域大量情報の高次処理」総括班報告, 1976-3
- 4) 分散処理, 情報処理, 20 (4), (1979)
- 5) コンピュータ・ネットワーク, 情報処理, 16 (7), (1975)
- 6) EDOS-MSO BCS/MCS プログラム・マニュアル, 日立製作所