

KUIPNETの総合報告

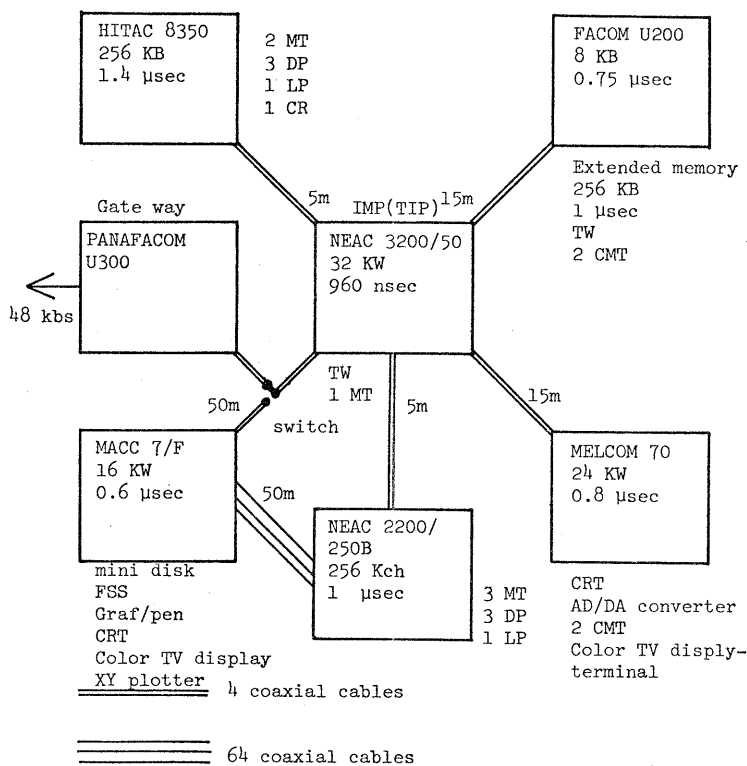
坂井 利之, 北澤 茂良, 林 恒俊, 田畑 孝一, 金出 武雄.
(京都大学工学部)

1. まえがき

計算機ネットワークとは、いくつかの独立した計算機システムが対等の立場にたつて、高速の通信回線によって結合されたシステムとして定義される。KUIPNETもこの方針に従って開発されたインハウス向きの計算機ネットワークであり、画像処理、シーン・アナリシス、音声理解システムあるいは人工知能などのより高度な情報処理の能力を強化することを意図している。このような処理においては、各 Host 計算機間のファイル、周辺装置などの資源の共有が不可欠である。この共有を可能にするためにはデジタル音声信号の実時間転送、画像データの高速転送などが必要である。

KUIPNET のサブネットでは、IMP は異なる Host 内の二つのプロセス間に連続実効転送速度 200 kbps 以上のデータ転送能力を持ち、これらの資源の共有も可能にしている。このように文字メッセージ・データのみでなく生データの処理が可能である点が KUIPNET のもっとも重要な特徴の一つである。

この KUIPNET に参加している Host 計算機の構成及びその資源の概要について説明するとともに、いくつかの IMP の利用度のパラメータとサブネットのトポロジックとについておこなった計測結果を報告する。又、中心となる Host 計算機 NEAC 2200/250B の OS : NOS 会話型処理システムの概要を説明する。



2. KUIPNETの構成及び各Host計算機の概要

KUIPNETは第1図に示されているように、メッセージ交換用計算機IMPを中心にして星型に結合されたHost計算機から構成されてい

第1図、 KUIPNETの構成。

る。IMP と Host 計算機は4本の同軸ケーブルによって (FACOM U-200 を除いて) 半二重方式で結合されている。この結合の通信速度は最大 1.6 Mbps である。

2.1 メッセージ交換用計算機 IMP⁽³⁾

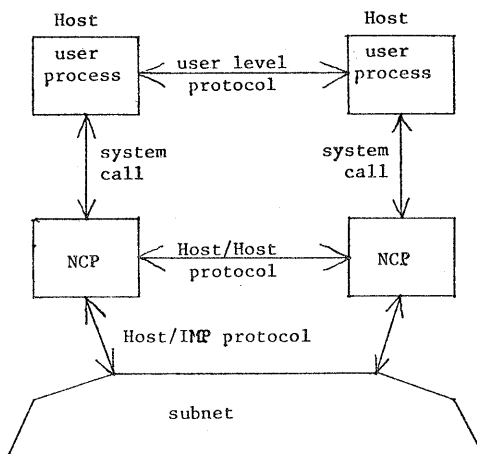
NEAC 3200/50 を KUIPNET の IMP として採用した。この計算機は DDP-516 と同じであり、高速度の処理が可能である。

メッセージは IMP と Host 計算機の間をビット直列で転送される。転送の単位となるメッセージは最大 8096 ビットの長さである。メッセージの高速転送を可能にするため、IMP のデータ・チャネルの優先順位はチャネルに接続された Host 計算機対の要求に従って動的に可変にする必要がある。NEAC 3200 のチャネルの優先順位はチャネルの番号に対応して固定されているため、可変優先順位指定回路をチャネルと通信インタフェースの間に設置し、この回路によって優先順位を制御している。この回路は、ある通信路の優先権をその通信路のリンクの確立から解放まで保持するが、その期間内でも優先度の低い通信路のデータ転送を禁止することはない。その動作は通常のマスク回路によるものとは全く異なっている。

2.2 プロセス間通信とプロトコル⁽²⁾

プロセス間における通信の手順は、ネットワーク内全体を通して統一されている。この手順はプロトコルと呼ばれ、第2図に示すように KUIPNET の場合には ARPANET とほぼ同一の階層構造を持つものを採用している。各 Host 計算機にはこのプロトコルを実現するための網制御プログラム (NCP) が作成され、Host 計算機の規模に合わせて使用されている。

2.3 各 Host 計算機の概要



第2図、プロトコルの階層構造。

NEAC 3200/50: メッセージ交換用計算機で、KUIPNET の通信網の中心に位置すると同時に、その端末タイプライタから TIP として KUIPNET 中の TSS にアクセス可能になっている。また、9トラック磁気テープ装置が1台接続されている、主にネットワークのトラフィックの測定結果の記録に用いられている。サイクルタイム 960 msec の主記憶は 16 ビット/語で 32 K 語 (フル実装) を持ち、これらのメッセージ交換用プログラム及びメッセージ・バッファ以外に、端末制御、ユーザ TELNET、トラフィック計測、磁気テープ制御などのプログラムを格納している。現在、文字ディスプレイ装置を端末として追加設置予定である。

NEAC 2200/250B: 中型 Host 計算機で、本格的な網向き OS すなわち NOS を持ち、ネットワークの計算・情報処理能力の中心となっている。主記憶はサイクルタイム $1 \mu\text{sec}$ 、6ビット/字で 256k 字実装されている。ラインプリンタ、紙テープ・リーダー、パンチ、ワトラック磁気テープ装置3台、ディスク・バック3台を備えている。また、中央処理装置には記憶保護、アドレス・リロケーションなどの多重プログラミング機能と共に浮動小数点演算機能も備えられている。NOS の下で動作する言語処理システムとして、マクロ・アセンブラ、フォートラン、コボル、LISP 及び汎用クロス・アセンブラがあり、これらのプログラムを NOS の TSS の下で利用することも可能である。

MACC 7/F: 主に画像処理研究用に使用されている計算機で画像関係の周辺装置を豊富に備えている。白黒/カラー・フライング・スポット・スキャナ、グラフ・ペン座標入力装置、蓄積型 CRT ディスプレイ、カラー TV ディスプレイ、X・Y プロットなどがある。その他にキャラクタ・ディスプレイ、ミニ・ディスクなどが備えられている。これらの装置は会話型画像処理システムによる制御されている。サイクルタイム $0.6 \mu\text{sec}$ 、16ビット/語、16k 語の主記憶が実装されている。さらに、NEAC 2200/250B と MACC 7/F との間がチャネル対チャネル・アダプタによって相互結合されていて、この2つの計算機によって計算機複合体を構成している。この計算機複合体も NOS によってサポートされている。

MELCOM 70: 主に音声研究用に使用されている計算機で、音声信号変換関係の周辺装置を豊富に備えている。蓄積型ディスプレイ、AD/DA 変換器、カラー TV ディスプレイ端末機、カートリッジ・ディスク装置などがある。さらにシリアル・プリンタ、カセット磁気テープ装置などが用意されている。サイクルタイム $0.8 \mu\text{sec}$ の主記憶装置 16ビット/語、24k 語が実装されている。この Host と NEAC 2200/250B の間でサブネットを通じて音声データの実時間転送が可能である。

FACOM U-200: バルク記憶を目的とした計算機で、サイクルタイム $0.75 \mu\text{sec}$ 、8ビット/バイト、8k バイトの主記憶装置の他に拡張 IC 記憶装置・サイクルタイム $1 \mu\text{sec}$ 、256k バイトが装備されている。この拡張 IC 記憶装置は中央処理装置の共通バスに接続されている。

PANAFACOM U-300: 異計算機網間のゲートウェイ計算機として使用されている。KUIPNET と京大・東大・大型計算機センタ・ネットワークの間のプロトコルの相異を吸収し、プロセス・プロセス間の通信を可能にしている。⁽¹⁾

3. ネットワークの計測システム⁽⁴⁾

計測を行う場合にネットワークに特別に負荷を与える場合と、現実のユーザの利用形態のままその利用傾向を含めて行う場合とがある。本システムは KUIPNET のメッセージ交換用計算機 (IMP) に於いてネットワークの日常の使用状態でのパラメータについて計測することを目的としている。ここでは IMP の資源の利用状態、ネットワーク内のトラフィック、プロセス間の接続状態に関する各種パラメータについて計測した。

3.1 計測システムの扱うパラメータ

IMP 内で計測を行うため、IMP 内で観測可能なパラメータとして第1表に示すものを選んだ。IMP 内の資源利用状態として蓄積交換バッファ、Host-IMP 通信

路及びプロセッサについてみた。蓄積交換用バッファの利用回数を計数した。特殊な状態として、IMPの蓄積交換用バッファが全部使用中で空きバッファが無い状態、さらに空きバッファがなく Host からの転送要求があった場合 (この場合 Host は IMP で空きバッファができるまで持たされる) について、これらの事象の生じた回数を計数した。これは IMP の蓄積交換バッファ容量が十分であるか、バッファがどの程度の頻度でオーバーフローするかをみる。Host-IMP 間は半二重であるか全二重に比べてどの程度不利であるかを知る一つの尺度として、半二重路上の衝突の (すなわち半二重路の両端から同時に転送要求を出したとき) 回数とこのために増加した転送再試行の回数を計数した。IMP の負荷は IMP の利用率から知ることが出来る。ここでは IMP プログラムのバックグラウンド・ルーチンで、ルーチンの実行回数を計数し、CPU のアイドル時間の割合から IMP 利用率を求めた。サンプリング間隔は 25ms とした。計測のために IMP に与える負荷を小さく保つためにこの程度とした。瞬時負荷は 25ms で平均化されるが、ちなみに 1 メッセージの処理時間は約 1ms である。IMP 利用率のヒストグラムを 1 分間について作成し、記録した。トラヒック・マトリクスを記録するために 1 分毎に交互に使用される 2 つのトラヒック・テーブルが備えられている。1 つのテーブルの大きさは 256 語で 1 分間通過メッセージを累算した後、バックグラウンド・タスクで磁気テープに書き込まれる。プロセス間接続状態はネットワークがどのようなジョブに使用されているかを知るために重要な情報を与えてくれる。ここでは、IMP で通過する Host-Host プロトコル・レベルのコントロール・コマンド (リンク #0) のメッセージをモニタした。測定の間隔及び記録するデータ量は、測定対象の時間的分解能と記録装置 (磁気テープ) の容量とデータ転送速度との兼ね合いで設定した。

3.2 計測データの記録

全この計測データは第 1 表のレコードの形式で IMP に取付けられた磁気テープ装置で記録される (1/2 インチ, 800 bpi, 1200 フィート, 9トラック, 30字/秒) 毎分記録されるデータは第 1 表のレコードタイプ 2 (80 バイト) とタイプ 3 (512 バイト) の計 592 バイトで、テープの容量は約 5M バイトとして、約 140 時間記録できる。トラヒック無しときは記録されないのが実際にはもっと長く記録できる。IMP の起動から停止までの間に記録されるデータ (レコードの集合) が一つのファイルを構成する。日々に起動する毎に生成されるファイルの集合が本計測システムの扱うデータである。起動時に最初に書き込まれるレコードは

第 1 表 計測システムの扱うパラメータ。

レコード No	項目	レコード長	パラメータ	事象	記録間隔
1	ファイルヘッダレコード	80 バイト	日付、時刻	起動時にキキレコード	IMP 起動時
2	IMP 内の資源利用状態	80 バイト	フリーリストから取られたバッファ数	受信終了時	毎分
			フリーリストから空リソースにバッファを割付けられた回数	受信終了時	
			フリーリストに返されたバッファの数	送信終了時	
			バッファの無い状態に達した Host からの転送要求の数	受信起動時	
			Host-IMP 間を通過する路上の衝突回数	受信起動時	
			衝突の再実行回数	送信起動時	
3	トラヒックマトリクス	512 バイト	逐格一宛先 Host 毎に分類した通過メッセージ数	メッセージ毎	毎分
4	プロセス間接続状態	1024 バイト	プロセス間接続のためのコントロールコマンド	コマンド毎	記録用バッファがいっぱい時
			CPU のアイドル時間	25ms 毎	

日付と時刻を示す(レコードタイプ1(第1表中))。1つのファイル内の全てのレコードはそれが記録された順にリンクを付加される。(このリンクは、フィードバック付シフトレジスタの演算をプログラムで模擬して生成する16ビットのコードである。)しかし異なるファイルのレコード間ではリンクしない。IMP起動時に磁気テープファイルの初期設定手順が各レコードに付随しているリンクが正しいことを確認しながら以前に書かれたファイルのレコードを順に読み込み、最後に書かれたレコードの次の位置にテープのヘッドを設定することができる。従ってIMP停止時には計測データのファイルに并して何の操作も必要でない。これはIMPが異常状態で停止した場合にも計測データが保存されることを保障し、オペレータの介入を不要にしてシステムの信頼性を高めるためである。

3.3 結果と考察

(1) バッファの利用状態: 使用バッファ数はトラヒック・マトリクス中の通過メッセージの総和と等しい。フリーリストから取り出されたバッファ数とフリーリストに返されたバッファ数の差は処理中か待ち行列上にあるメッセージの数になるが、計測結果では1以上の場合はなかった。これはトラヒックが軽かったためであろう。同様に、フリーリストにバッファが無くなったり、またバッファの無いときにさらにHostからの転送要求が来ることもなかった。またHost-IMP間の通信路上の衝突も非常にまれ(全計測期間を通じて数回程度)

であった。これは独立なジョブが多量にネットワークを使用することが少なかったことと、トラヒック密度が小さかったこととによるとみられる。

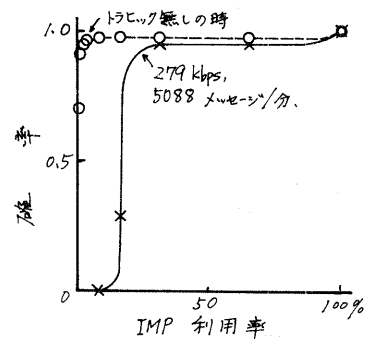
(2) トラヒックの無いときのCPUの利用率: ネットワーク内にトラヒックの無い時、IMPプログラムはタイマによる起動されるタイム・タスクとバックグラウンドのループだけを実行する。このときCPUの利用率は約20.8%でIMPはこれを次の機能に分割して使用する。(a) タイマ, 4.6%, (b) 端末装置の状態センサ 16.2%。(a)はメッセージ交換タスクより高い優先順位で実行されるIMP固有のオーバーヘッドになる。(b)は最も低い優先順位のバックグラウンドのループ内で実行されるのでトラヒック密度が増大すれば0%まで減少することがある。

(3) IMP利用率とトラヒック: 第3図の結果はIMPを通過する279kbpsのトラヒックがあったときに25ms間隔で測定したIMPの利用率の分布を示している(図中の実線)。比較のためトラヒック無しの時のIMP利用率を点線で図中に示した。前者は約20%の利用率を示しているが、これはチャンネルの利用率

$$\frac{279 \text{ kbps}}{2080 \text{ kbps}} \times 100\% = 13.4\%$$

とメッセージ交換プログラムのメッセージ当りの処理に対応するCPUの利用率 $\frac{5088 \text{ メッセージ/分} \times 1 \text{ ms/メッセージ}}{60 \text{ 秒/分}} \times 100\% = 8.4\%$ の合計21.4%に対応するものとみられる。

(4) トラヒック・マトリクス: 第2表はIMPを通過した全てのメッセージの集計である。各欄について上段は総ビット数、下段はメッセージ数である。表から見られるように最も活発にネットワークを使用したHostはNEAC 2200と



第3図. IMP利用率の分布

MELCOM 70 である。これらの Host 間では実時間データのデジタル音声波形の転送を行っている。自 Host から自 Host への転送すなわち自 Host 内のプロセス間での転送も可能で、NEAC 2200 及び TIP で行われた。

(5) プロセス間の接続状態: プロセス間の接続状態は Host-Host プロトコル・レベルのコントロール・コマンドから知ることが出来る。磁気テープにコピーしたコントロール・コマンドを含む IMP の蓄積交換バッファから次のような情報を得ることができた。

(a) NCP 間の通信を記録した磁気テープの内容を解析して、接続確立のためのソケットの不整合などを検出して、ユーザ・プログラムのデバッグに役立った。

(b) 蓄積交換バッファにはメッセージ発生時刻に関する情報が追加されており、これから接続確立に要する時間、接続継続時間等を知ることができた。第3表は接続確立に要する時間を各 Host の NCP について平均値で示したものである。

(c) 接続に関するプロセス名を識別してネットワークを使用している業務を識別できた。

第2表、トラヒックの総和。

{ IMP を通過したビット数
IMP を通過したメッセージ数 }

from APR. 19, 1976 through DEC. 8, 1976

Destination Source	NEAC2200	PF-U300	MELCOM70	FACOMU300	TIP	7050A5600
NEAC2200	104110448 99712	14902246 8328	33953504 179410	192 6	312672 2876	8327168 5978
PF-U300	14629216 48080	0 0	315296 232	0 0	0 0	0 0
MELCOM70	589357492 512847	11808 77	0 0	19688 181	0 0	0 0
FACOMU300	48592 23	0 0	10880 23	0 0	0 0	0 0
TIP	292276 2204	0 0	0 0	0 0	53232 536	320 2
7050A5600	949232 5781	0 0	0 0	0 0	32 1	0 0

第3表、接続確立に要する時間。

from \ to	NEAC2200	MELCOM70	TIP
NEAC2200	16.25ms	4.25ms	14.40ms
MELCOM70	15.50ms	-	-
TIP	14.40ms	-	4.16ms

- はデータ無し。

4. Host 計算機のネットワーク向き OS: NOS の会話型処理⁽⁵⁾

ネットワーク中の情報処理に対する様々な要求に応じるため、中型 Host 計算機 NEAC 2200/250B 上に、ネットワーク向き OS: NOS が開発された。

NOS は (1) 多重プログラミング、(2) リアルタイムでデータを処理するプロセスに対する高速スケジューリング、(3) ネットワーク側からのジョブの投入 (TSS 処理)、(4) 計算機複合体に対するサポート、(5) 多種のユーティリティ及び言語処理プログラムなどの特徴を持っている。

これらの特徴を実現するため、NOS の設計方針に中核・拡張方式を採用した。これは政策部と機構部を分離したうえで、効率を向上させるため機構部にジョブ・スケジュールに必要な (場合によっては政策部に含まれる) 機能を持たせるようにした方式である。

この機構部は NOS モータと呼ばれて、NOS オペレーティング・システムを中心に、この NOS モータの上にそれを拡張して、様々なジョブ・ス

スケジュールの方式として、NOS オペレーティング・システムは作成されている。これらのスケジュール方式の中に TSS システムとしてサーバ TELNET があり、主にネットワークからジョブを投入するのに用いられる。

4.1 サーバ TELNET システム⁽⁶⁾

NOS の TELNET システムは ARPANET における TELNET プロトコルとほぼ同様のプロトコルに基づいて設計されている。主として NEAC 2200 以外の計算機から NOS を会話型式で使用するため、サーバ TELNET が用いられる。

NOS のサーバ TELNET は TELNET 日か・プロセスと TELNET 転送プログラムから構成される。TELNET 日か・プロセスはユーザ TELNET からのログインを受け、ユーザ TELNET にメッセージ転送のためのコネクション・ソケット番子を送信すると同時にサーバ・プロセスを生成する。サーバ・プロセスはメッセージ転送のためのコネクションを確立し、TELNET 転送プログラムをユーザ・プログラム領域の最後に格納する。この TELNET 転送プログラムはユー

- | | | |
|------|--|---------------------------------|
| (1) | @ L O | (1) TELNET に対してログ・イン |
| (2) | TR OPEN | (2) TIP からの応答 |
| (3) | NOS TELNET V-02, L-00 RDY. | (3) TELNET からの応答 |
| (4) | A DISK, NO, LOGU=(04,0),
A TGO, N1,
X, SAKAII01, DNO=1, | (4) NOS のジョブ制御コマンドの入力 |
| (5) | BGN STEP | (5) 会話型バッチ処理プログラムの開始 |
| (6) | ? EDIT/EXEC/SYSOUT/SAVE/END | (6) 会話型バッチ処理プログラムからトップレベル |
| (7) | EDIT
? LANGUAGE
EASYCODER
? NEW/OLD/REEDIT
NEW
? AUTO/MANUAL
AUTO | コマンドを要求
(7) トップ・レベル編集コマンドの入力 |
| (8) | ?00010 SYSOUT ASGN NO,
?00020 DISK ASGN N1, LOGU=(04,0),
?00030 ASM EX *PROGDEV.
?00040 E FUNCT EASYCODER,
?00050 PROG DUMMY
?00060 END
?00070 ENDPDS
?00080 ENDJOB
?00090 END | (8) 原プログラムの入力 |
| (9) | END EDIT | (9) 編集の終了 |
| (10) | ? EDIT/EXEC/SYSOUT/SAVE/END
EXEC GO,5 | (10) バッチ・ジョブの開始コマンド |
| (11) | END EXEC
? EDIT/EXEC/SYSOUT/SAVE/END
END | (11) 会話処理の終了コマンド |
| (12) | END STEP, CC = 00.
ENDJOB
END TELNET.
TR CLOSED | (12) TELNET に対するセッション終了コマンド |

第4図. 会話型バッチ処理の実例.

が、プログラム中の操作卓に対する入出力命令によってひきおこされるトランプの入口であり、このトランプした命令のデータを TELNET プロトコルに従ってコード変換などを行い、TELNET コネクションを通して転送する。これらのプログラムによって、NOS の TSS 形式による会話型処理が実現されている。

4.2 会話型バッチ処理

TELNET システムの下で動作するプログラムとして、現在会話型バッチ処理及び会話型デバッグ・システムが作成されている。

会話型バッチ処理は、プログラムのテキスト編集プログラムを中心として、プログラムのセーブ、再編集、ジョブの実行、ジョブの出力などから構成されている。ユーザはプログラムを会話形式で入力、編集したものを保存したり、実行したり、あるいは実行した結果を端末へ出力したりすることが可能となっている。第4図に会話型バッチ処理の実例をあげて説明する。下線の部分がシステムからの出力を示している。

4.3 会話型デバッグ・システム

会話型デバッグ・システムは、トレーサ・プログラムを中心としてトレース条件、トレース終了条件、トランプ条件、レジスタや主記憶の内容の読み出し、変更などを会話形式でおこなえるようにしたものである。

5. 結論

KUIPNET が開発されてから3年が経過したが、現在は各 Host の資源の強化を除いてほぼ安定し、恒常的に運転され、実用化されている。これらのトランプの計測結果からも、各 Host 計算機の資源を共有して、高度な情報処理をおこなううえで有効かつ重要な手段となり、欠かせぬものともなっている。

謝辞

KUIPNET の開発にあたり、坂井研究室の多くの方々の協力に感謝いたします。

参考文献

- (1) 坂井, 金出, 藤原: 異なるプロトコルを持つ計算機ネットワークの相互接続, 情報処理学会第9回コンピュータネットワーク研究会資料.
- (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, Vol. 15, No. 12, PP. 948-954, 1974.
- (3) 坂井, 田畑, 北沢, 大西: KUIPNETの情報交換制御方式, 信学論(D), Vol. 58-D, No. 11, 1975.
- (4) 北沢: "Development of an In-house Computer Network KUIPNET" 博士学位論文, (申請中), 1976.
- (5) 坂井, 林: ホスト計算機のネットワーク向モニタ, 第15回情報処理学会全国大会, 1974.
- (6) 坂井, 林: ネットワーク向モニタNOSのTELNETシステム, 第16回情報処理学会全国大会, 1975.
- (7) 坂井, 北沢: KUIPNETの解説, KUIPNETマニュアル1, 1976.
- (8) 坂井, 木村: NEAC 2200/250 NOSオペレーティングシステム, KUIPNETマニュアル2, 1975.