衝突を回避する簡易バス型ローカル・エリア・ネットワーク

滑川英世*, 青木正夫**, 岸上利秋*

(昭和60年9月6日受理)

A Simple Bus Local Area Network using Collision-Free Access Control

Hideyo NAMEKAWA*, Masao AOKI** and Toshiaki KISHIGAMI*

Abstract — This paper describes an implementation of a simple bus local area network which employs distributed collision-free access control scheme.

The control scheme uses control-wire in addition to data bus to schedule the transmissions of each stations.

The stations with RS-232C interface can be connected easily to the network by using network controllers.

The transmission rate of the network is 50 Kilo bits/sec.

Details on the design of hardware and software for the network controllers are discussed.

1. まえがき

ローカル・エリア・ネットワークのトポロジーにはバス型とリング型があり，バス型では*Ethernet*に代表されるCSMA／CD方式，リング型ではトークン・パッシング方式が主流となっている。そのうち，バス型はタップが受動的であることが最大の長所である反面，各ステーションの競合をどう解決するかという問題がある。CSMA／CD方式では，それをキャリア検出と衝突検出，そして衝突した場合の再試行法に頼っており，最大スループットや応答時間などの面であまりよい特性は得られない。これに対してラウンド・ロビン・スケジューリングを行なう方式が多数提案されており，性能面や強靱さで注目すべき性質をもつ方式がいくつかある。[3] そのうち代表的な方式はトークン・バスであり標準化もかなり進んでいるが，欠点としてはプロトコルが複雑なことがある。また，他の方式については実際にインプリメントされた例があまりなく，理論的な解析しかなされていないものが多い。

1979年に*K.P. Eswarn*, *V.C. Hamacher*, *G. S. Shedler*らによって提案された方式[1]は，制御線を用いることによってアクセスのスケジューリングを行なうものである。これは，簡単なアルゴリズムでラウンド・ロビン・スケジューリングを実現できるなどいくつかの長所をもつが，まだインプリメントに関する報告はなされていない。そこで，この方式の有効性と実用性を確かめるため，このアクセス制御方式（以下，*The Control-Wire*と呼ぶ。）を用いた簡易ネットワークを構築し，併せて研究室用のコンピュータ資源の共有化をはかることを考えた。設計にあたってはつぎのようなことを目標とした。

- アクセス方式として*The Control-Wire*を採用し，その理論を正確に実現する。

* 茨城大学工学部情報工学科（日立市中成沢町）
Department of Information Science, Faculty of Engineering, Ibaraki University, Hitachi 316, Japan

** 茨城大学大学院工学研究科情報工学専攻（日立市中成沢町）
Graduate Student, Department of Information Science, Faculty of Engineering, Ibaraki University, Hitachi 316, Japan

- 階層構造を明確にしてそれぞれをモジュール化し，ソフトウェア，ハードウェア両面の汎用性，拡張性，わかり易さを追求する。
- できる限り多くの機能をネットワーク・コントローラに任せ，ホスト・コンピュータの負担を軽くする。
- ホスト・コンピュータを，ハードウェアの特別な改造なしに手軽に接続できるようにする。
- 通信速度はある程度実用的な水準を目指す。
- データは透過性（*transparency*）を保てるようにする。
- ネットワーク・コントローラは，ハードウェアをできる限りコンパクトにまとめ，ユニバーサル・ボード1枚程度とする。
- アプリケーションはとりあえず基本的なものに限定し，後で拡張する。

現在，このネットワークにはBASIC MASTER LEVEL-Ⅲが2台接続されている。他に数台のコンピュータ（ミニ・コンピュータを含む。）を接続する作業が進行中ではあるが，現時点ではアプリケーションも基本的なものだけであるため，本稿ではネットワーク・コントローラの説明を中心にネットワーク・システムの構築例について述べる。

2. 制御線を用いるマルチ・アクセス制御

V.C. Hamacher らは(1)において *The Control-Wire* 方式の正当性を証明し，また(2)において待ち行列システムによる性能解析を行なっている。しかし，そこで解析されたアクセス方式は，提案されている2つのアルゴリズムA1とA2のうち公平性が失なわれる危険性のあるA1の方であるため，ここではA2についてはシミュレーションによって性能を確認した。その結果の一部をFig. 1 に示す。*Ethernet* と比べると非常によい特性をもつことがわかる。本ネットワークで採用したアクセス・アルゴリズムはこのA2であり，すべてのステーションに公平なアクセスが保証されている。

このアクセス方式の長所はつぎのような点にある。

- 完全に非同期で，かつ完全に分散されたアルゴリズムである。
- アルゴリズムが簡単である。
- 制御線上の信号はレベル信号のため，回路が簡単でよい。
- ラウンド・ロビン・スケジューリングを実現しているため，応答時間の予測が可能である。

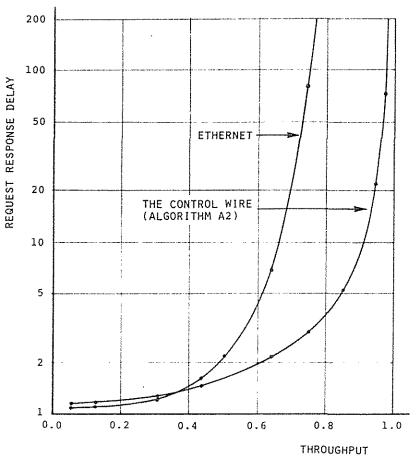

Fig. 1 Delay-throughput characteristics for Ethernet and The control wire.

2.1 衝突回避のためのハードウェア機構

まず，必要なハードウェアの構成をFig. 2 に示す。図中の記号の意味は以下のとおりである。

B（J）：　バス上の信号

P（J）：　ポートJの上位にあるすべてのポートのSEND F／FのOR信号

S（J）：　ポートJのSEND F／Fの値

また，ポートという用語は(1)で使われているものであり，本ネットワークではネットワーク・コントローラと呼んでいる部分を指している。まず，ネットワーク・バスはデータの送受信を行なう双方向性の信号線であり，各ポートは受動タップを通してバスにアクセスする。制御線は一方向性であり，ポートJは上位からの制御線上の信号をP（J）として受け取り，S（J）とのORを下位に送る形で接続される。このように制御線は能動素子を通して接続されるので，信頼性を保つために何らかの対策を必要とするが，制御線上の信号はF／Fの単なるOR信号であって非同期であるため，本ネットワークでは故障したポートを切り離すのに，リレーによるバイパスを設けるという簡単な方法をとった。

2.2 衝突回避のためのアクセス・アルゴリズム

まず，*V.C. Hamacher* らによって提案された2つのアルゴリズム，A1とA2について説明する。これら

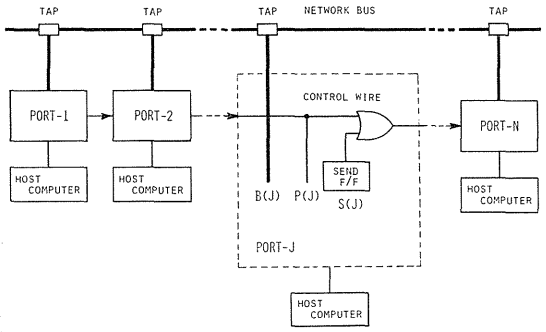

Fig. 2　Bus network and ports.

のアルゴリズムはポートＪにおける送信権を得るまでの手続きとして記述される。記述に使用される信号線の伝搬遅延時間に関する記号は，

Ｔ：　バスの端から端までの信号伝搬遅延時間

Ｒ（Ｊ）：　制御線のポートJから最下位ポートまでの伝搬遅延時間

であり，Ｒ（１）＞Ｔと仮定されている。以下，提案されたアルゴリズムを(1)からそのまま引用し，簡単に説明する。

アルゴリズムＡ１

＊　Ｓ（Ｊ）を１にセットする。
＊　Ｒ（Ｊ）＋Ｔの時間待つ。
＊　バスがアイドルで，かつＰ（Ｊ）＝０となるのを待って，直ちにパケットの送信を開始し，同時にＳ（Ｊ）を０にリセットする。

この簡単な手順で衝突が回避される。しかし，一般にアルゴリズムＡ１は全ポートに公平なアクセスを保証することができず，上位ポートにアクセス権を独占される危険がある。

アルゴリズムＡ２

＊　２Ｔの間バスがアイドルとなっているのが検知されるまで待つ。
＊　Ｓ（Ｊ）を１にセットする。
＊　Ｒ（Ｊ）＋Ｔの時間待つ。
＊　バスがアイドルで，かつＰ（Ｊ）＝０となるのを待って，直ちにパケットの送信を開始し，同時にＳ（Ｊ）を０にリセットする。

すぐわかるように，アルゴリズムＡ２はアルゴリズムＡ１に２Ｔの間バスがアイドルになるのを待つ手順を加えたものである。これによって，若干オーバーヘッドは増えるが，全ポートに公平なアクセスが保証され，ラウンド・ロビン・スケジューリングを実現できる。

アルゴリズムＡ１とＡ２のこのような特性から，本ネットワークではアルゴリズムＡ２を採用している。なお，より詳しいアルゴリズムの説明と各アルゴリズムの特性の解析，また衝突を回避することの証明などについては(1)を参照されたい。

つぎに，本ネットワークにおけるデータ・パケットの送信手順を示して，アルゴリズムＡ２のインプリメンテーションについて簡単に説明する。ポートＪにおけるデータ・パケットの送信手順を流れ図にすると，Fig. 3 のようになる。この手順は，アクノリッジ・パケットの到着待ちと再送などの判断を行なう手続きをアルゴリズムＡ２に加えたものとなっている。各ポートにアクノリッジを管理する機能をもたせたのは，ホスト・コンピュータの負担を軽くするためである。

また，アルゴりズムＡ２の部分についても，Ｒ（Ｊ）をＲ（＝Ｒ（１））に変更してある。これは，各ポートごとに異なる値を使うのはいろいろな面で望ましくないためであり，多少無駄は出るが最大値であるＲ（１）の値を使ってインプリメントや保守を容易にした。この変更は，アルゴリズムの正常な働きに影響しない。

3.　ローカル・エリア・ネットワーク・システム

ネットワーク・システムの設計方針は，階層構造を明確にするという点においている。まず，階層構造を説明してから，各階層ごとに機能と主な仕様を説明する。

3.1　システム構成の概要

階層名を下位階層から順に挙げると，まずネットワーク・コントローラに

- 物理リンク階層
- 論理リンク階層
- パケッティング階層

があり，ホスト・コンピュータには

- コントローラ・リンク階層
- ユーザ・アプリケーション階層

があり，全部で５階層からなっている（Fig. 9 参照）。ホスト・コンピュータ上の２つの階層を実現するソフトウェアを，ここでは通信ソフトウェアと呼び，RS-232 Cを通してネットワーク・コントローラと通信する。ま

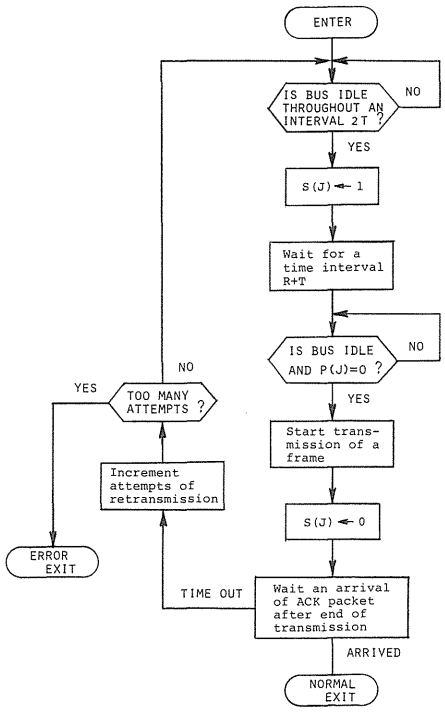

Fig. 3 Protocol for transmitting data packets.

た，ネットワーク・コントローラはCPUにMC68B09を用い，論理リンク階層から上をソフトウェアで実現することによってハードウェアをコンパクトにしている。

3.2 各階層の機能とその仕様

下位階層から順に述べる。

● **物理リンク階層**

すべてハードウェアでインプリメントしてあり，媒体アクセスの管理を行なう。以下，簡単に機能を説明する。

ACIA（6850）により，バイト単位のデータとネットワーク・バス上のデータの直並列変換を行ない，ネットワーク・バスへのアクセスを行なう。また，制御線とSEND F／Fを直接管理して，アクセス・アルゴリズムの大部分を実行する。具体的には，論理リンク階層が物理リンク階層内のF／Fをセットすることによってアルゴリズムの開始を物理リンク階層に指示し，その後送信を開始してS（J）を0にするまでのアルゴリズムを，物理リンク階層がすべて実行する。したがって，この部分は論理リンク階層と密接な関係をもつ。

つぎに物理リンク階層の仕様をまとめる。

- 転送速度…50kbps
- 最大ケーブル長…300m
- ネットワーク・バスおよび制御線媒体…ツイスト・ペア線
- ネットワーク・バス変調方式…単流NRZ

まず，転送速度は直並列変換を行なうACIAの最大能力によって決定された。最大ケーブル長はアクセス・アルゴリズムのパラメータRとTの決定によるもので，送受信の能力による制限ではない。媒体と変調方式は手軽であることにより選んだ。

● **論理リンク階層**

すべてソフトウェアでインプリメントしてあり，送信側と受信側の2つのプロセスをもつ。プロセスは，パケッティング階層のプロセスとともに，コルーチン制御によって並行して動く。送信側プロセスをPUTNETといい，受信側プロセスをGETNETという（Fig. 9 参照）。PUTNETは，Fig. 3 の送信手順を物理リンク階層と協力して実行し，送信バッファ中のデータ・パケットを送信する。また同じアクセス・アルゴリズムを使ってアクノリッジ・パケットの送信を行なう。GETNETは，物理リンク階層が受信したバイト単位のデータからパケットを構成し，自分宛てのデータ・パケットならば受信バッファに格納する。自分宛てのアクノリッジ・パケットならばPUTNETに報告する。このように，論理リンク階層はパケットの送受信，アクノリッジの管理，再送制御などを行なう。

以下に論理リンク階層の仕様をまとめる。

- アクセス方式… *The Control-Wire*（アルゴリズムA2）
- 送信結果の確認…アクノリッジの受け取り
- 最大ノード数…16
- データ・パケット長…8～520バイト（データ部0～512バイト）
- アクノリッジ・パケット長…5バイト
- 誤り検出方式…Check Sum（1バイト）

最大ノード数については，アクノリッジ・パケット受信待ちの際のタイムアウト値に関係して決定され，実際はアドレス・フォーマット上での制約がある。254まで拡張が可能である。パケット長はRAMの容量から決定した。

つぎに，ネットワーク・バス上のパケットのフォーマットをFig. 4 に示す。（b）のアクノリッジ・パケット

におけるACKコードとは，受信した結果を示すコードである。また(c)のパケットは，それまでのデータ(いくつかのパケットに分けられている。)が無効になったことを知らせるパケットである。このパケットだけがポストアンブルをもつ理由は，このパケットの処理には時間がかかるため，その時間を得る必要があるからである。

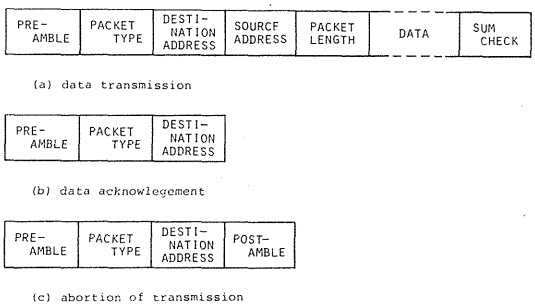

Fig. 4 Packet formats for data link layer.

そして，このフォーマットにおけるTYPEフィールドのフォーマットをFig. 5 に示す。TYPEフラグが0ならデータ・パケットを示し，1なら他の2つのパケットを示す。ENDフラグが1なら，そのデータ・パケットが一連のパケットのうち最後のデータが入っているものであることを示し，パケッティング階層でパケットをもとのデータに組み立てる際に使われる。ABORTフラグは，アクノリッジ・パケットとデータが無効になったことを知らせるパケットとを区別するためのものである。

● 送信バッファ，受信バッファについて

送信バッファは3つ，受信バッファは4つあり，それぞれリング・バッファを構成する。そして，論理リンク階層とパケッティング階層間のパケットの授受に使用される。各バッファの大きさは521バイト（データ部512バイト）である。

● パケッティング階層

送信側と受信側の2つのプロセスをもち，論理リンク階層のプロセスとともに並列に動く。送信側プロセスをGETHOSTといい，受信側プロセスをPUTHOSTという（Fig. 9 参照)。GETHOSTは，ホスト・コンピュータからRS-232Cを通してデータを受信し，それをパケットに分解して送信バッファに格納する。PUTHOSTは，受信バッファに格納されているパケットを1つのデータに組み立てて，RS-232Cを通してホスト・コンピュータに送信する。この階層によって，ホスト・コンピュータはデータをパケットに分解したりもと通りに組み立てたりする必要がなくなる。

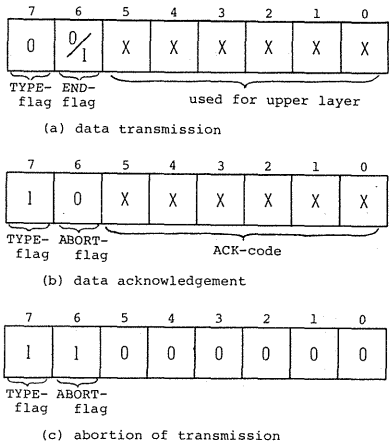

Fig. 5 Binary formats of packet type field in packets.

以下に，パケッティング階層とホスト・コンピュータ間のコマンドをすべて列記し，意味を説明する。コマンドはビット7が1の1バイト・コードである。

- SOT…データ送信開始。(後にアドレスとデータが続く。)
- EOT…データ送信終了。
- ABORT…データ送信中止。(そのデータは無効になる。)
- NONCH…ビット7が1のデータの前につける。(コマンドとデータを区別するため)
- SCC…送信成功をホスト・コンピュータに知らせる。
- ERR…送信失敗をホスト・コンピュータに知らせる。(後にエラー・コードが続く。)
- RSTART…ホスト・コンピュータがPUTHOSTに対し，データの送信を許可する。
- RSTOP…ホスト・コンピュータがPUTHOSTに対し，データの送信を禁止する。

このように，パケッティング階層とホスト・コンピュータのコントローラ・リンク階層はコマンドによって通信を行なうため，その通信フォーマットはFig. 6 のようになっている。なお，(a)のデータ通信におけるステー

ション・アドレスとは，送信時（コントローラ・リンク階層からパケッティング階層へ）はデスティネーション・アドレスを意味し，受信時（パケッティング階層からコントローラ・リンク階層へ）はソース・アドレスを意味する。データ部の長さには制限がない。また，(b)のエラー・コードはERRコマンドのみにつき，エラーの種類を知らせるコードである。

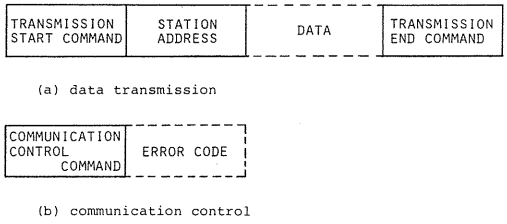

Fig. 6 Message formats for communicating between host computer and network controller.

● **パケッティング階層とコントローラ・リンク階層間の通信について**

ホスト・コンピュータとネットワーク・コントローラ間の通信には，ホスト・コンピュータを改造なしに接続できるよう，RS-232Cを用いている。これによって速度の面では多少マイナスとなるが，手軽に接続できる簡易ネットワークという目的に沿うものとなる。ネットワーク・コントローラが対応できるボー・レートは，1200，400，4800，9600，19200ボーである。

なお，少し補足すると，ミニ・コンピュータを接続する作業も行なっているが，この場合は，コンピュータのバスに直接接続してDMAによって送受信を行なうように設計したため，専用ネットワーク・コントローラとなっている。

● **コントローラ・リンク階層**

コントローラ・リンク階層の役割は，パケッティング階層との通信に必要なプロトコルをすべて受けもつことによって，ユーザ・アプリケーション階層がより手軽に，しかもトランスペアレントで何の制限もない通信を行なえるようにすることである。しかしホスト・コンピュータのOSによっては，モジュール化，階層化のむずかしいものもあり，そのようなものではこの階層は上位階層のユーザ・アプリケーション層に含まれることとなる。

現在，ネットワークに接続されているホスト・コンピュータは，BASIC MASTER LEVEL-Ⅲが2台であり，BASICと機械語でインプリメントしてあるためにユーザ・アプリケーション層との明確な区別はない。

● **ユーザ・アプリケーション階層**

最上位階層であり，いろいろな機能が考えられるが，現在，LEVEL-Ⅲにインプリメントしてあるのはつぎのような機能である。

- フロッピー・ディスク・ファイルの送信。
- キーボードからのメッセージの送信。
- 受信データのフロッピー・ディスクへの格納。
- 受信データのプリンタへの出力。
- 全ステーションの名前を登録し，その名前でステーションを指定できる機能。

この階層では，わずらわしい低レベルの通信プロトコルを考える必要がないため，容易にアプリケーション・ソフトウェアを作成することができる。

4. ネットワーク・コントローラ

4.1 ハードウェア構成

Fig. 7にネットワーク・コントローラのブロック図を示す。使用した集積回路はつぎのとおりである。

- CPU……MC68B09
- ROM……6116P 2個
- ACIA……6850 2個
- TTL IC…21個

その他，リレー，トランジスタ，ディップ・スイッチなどを使用している。

つぎに実際の回路についてであるが，CPU，ROM，RAMなどとそれに付随する回路は一般的であるため，物理リンク・レジスタ，キャリア・センス及び制御線回路，バッファリング回路の部分のみを説明する。

Fig. 8にその回路図を示す。図において左端の信号は，

- D_0～D_3…データ・バスの下位4ビット。
- R／$\overline{W}$……リード／ライト信号。
- E……68B09のシステム・クロック信号。
- E・R／$\overline{W}$…データ・バス上の信号衝突を回避するためのR／W信号。
- R_xD……6850の受信データ端子。
- T_xD……6850の送信データ端子。

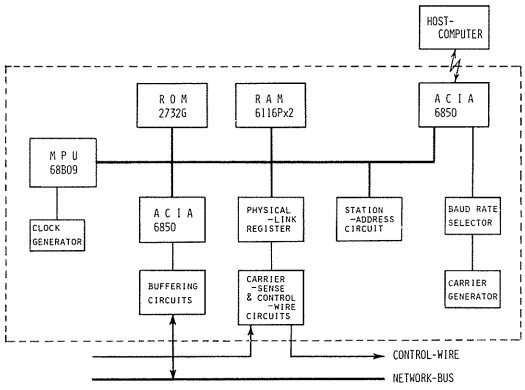

Fig. 7 Functional block diagram for network controller.

を示す。この回路とACIA（6850）とを合わせた部分が物理リンク階層であり，前節で簡単に説明した物理リンク階層の機能はこの回路で実行する。

まず，物理リンク・レジスタであるが，これは物理リンク階層と論理リンク階層のインターフェイスとなる部分である。回路では，左上のスリー・ステート・バッファを通してデータ・バスと接続され，アドレスが割り付けられることによって物理リンク・レジスタとしてアクセスされる。各ビットの役割はつぎのとおりである。

- D_0……論理リンク階層が1を書き込むことによってFig. 8左下のF／Fがセットされ，それによってこの回路が起動する。物理リンク階層が送信を開始するとこのF／Fはクリアされるため，論理リンク階層はこのビットを読むことによって送信開始を知ることができる。
- D_1……アルゴリズムの“P（J）＝0，かつバスがアイドルになるのを待っている”部分まで進んでいることを示すフラグ。送信が始まると

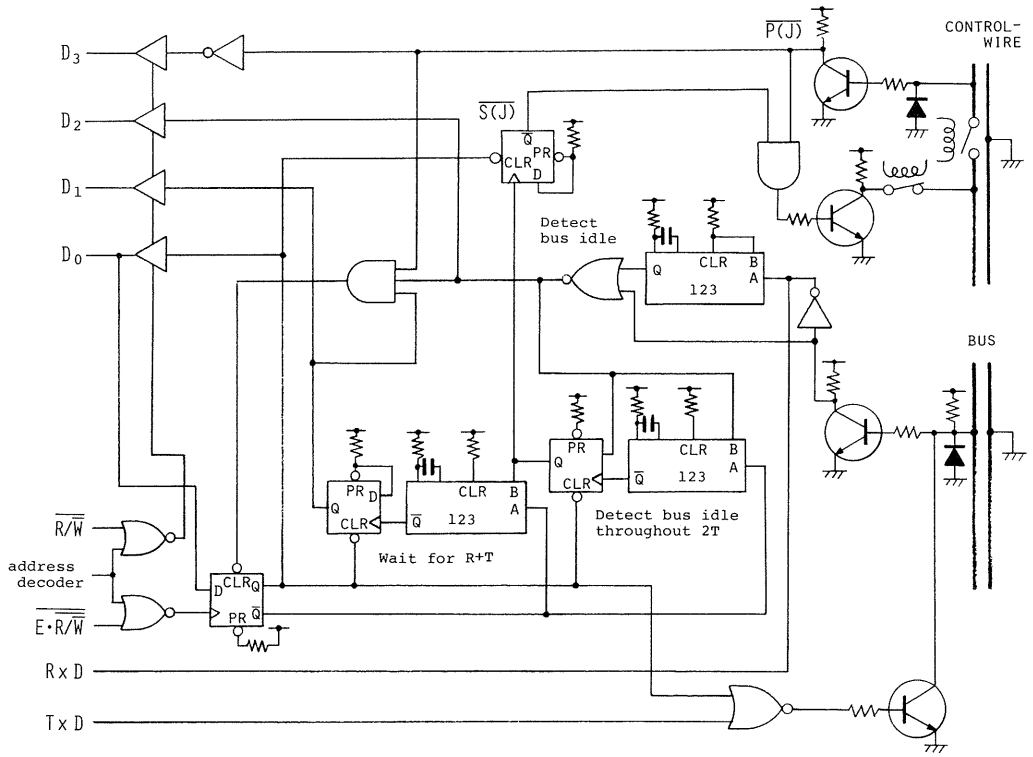

Fig. 8 Multi-access control circuits.

クリアされる。

- D_2……バスがアイドルであることを示すフラグ。アルゴリズムを実行していないときも，アイドルであるかどうかを示している。
- D_3……P（J）の現在の値を示すフラグ。D_2と同様に，アルゴリズムの実行に関係なく値を示す。
- D_4～D_7…使用しない。

実際に論理リンク階層でアルゴリズム実行のために必要なのはD_0だけであるが，他のフラグもネットワークをモニタするなどの用途に使うことができる。

つぎに，実際にアルゴリズムを実行する部分について説明する。この部分は，Fig. 7 のブロック図でキャリア・センスおよび制御線回路と呼んでいる部分であり，回路の大部分を占める。

簡単に原理を説明する。まず，論理リンク階層が左下のF／Fをセットすると，F／Fの出力が回路を起動するのと同時に，T_xDをマスクして送信を禁止する。この後すぐに，論理リンク階層はACIAに送信を指示すため，T_xDには送信信号が常に存在することになり，マスクをはずせばただちにネットワーク・バスにキャリアを送ることができる。回路は，ワン・ショットによる時間測定などによってアルゴリズムを実行し，送信開始条件がそろったら左下のF／FをクリアすることによってT_xDのマスクをはずし，S（J）を 0 にクリアする。論理リンク階層は，このF／Fを調べることによって送信開始したかどうかを知ることができる。

アルゴリズムを実行し，送信開始条件を調べる回路は，さらに 3 つの部分に分かれる。まず，ネットワーク・バスがアイドルであるかどうかを調べる回路であり，ワン・ショットとNORからなる。ワンショットは，ネットワーク・バス上のキャリアによって起動され，出力パルスの長さは400 μsec（バス上の 2 バイト分の長さ）である。

つぎが 2 Tの間アイドルであるかどうかを調べる回路であり，ワン・ショットとD　F／Fからなる。ワン・ショットの出力パルスは 4 μsecとしたので 2 Tは 4 μsecとなり，最大ネットワーク長として決めた300 m（2 Tは 3 μsecとなる。）と比較して多少余裕のあるものとなる。2 Tの間アイドルであることを検知すると，このF／Fの出力はS（J）を 1 にセットし，また，つぎのR＋Tの時間をはかる回路を起動する。

この回路もワン・ショットとF／Fからなり，ワン・ショットの出力パルスは 4 μsecとしてある。このF／Fがセットされ，P（J）＝ 0 ，かつバス・アイドルと

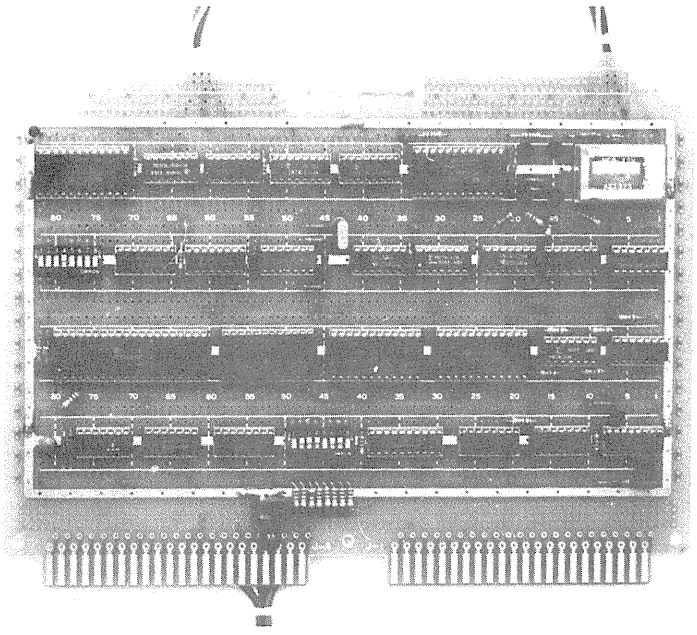

Photo. 1　Network Controller circuit board.

なったときが送信を開始すべき時点であり，ただちに左下のF／Fをクリアする。

残りは，Fig. 7のブロック図におけるバッファリング回路に対応し，Fig. 8ではトランジスタなどによって媒体にアクセスする部分にあたる。

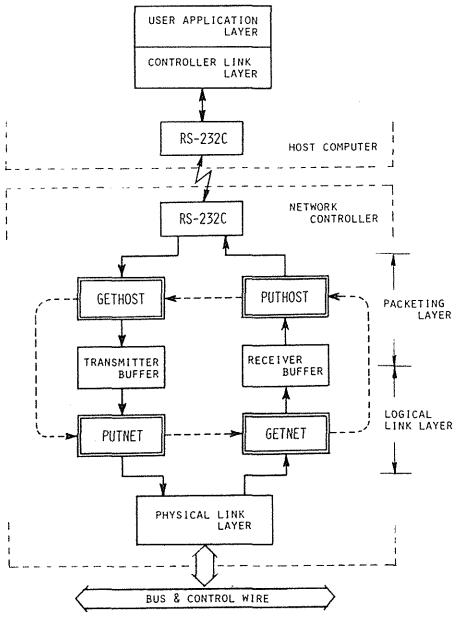

Fig. 9 Logical structure for network system.

4.2 ソフトウェア構成

Fig. 9にネットワークの論理的な構成図を示す。ネットワーク・コントローラにおいてソフトウェアで実現している階層は，パケッティング階層と論理リンク階層であり，それぞれ2つのプロセスからなる。以下で，それぞれのプロセスについて流れ図を示し，簡単に説明する。

● GETHOST

パケッティング階層の送信側を受けもつプロセスである。Fig.10に流れ図を示す。GETHOSTがコントローラ・リンク階層から受け取るのは，SOTコマンドで始まるデータ・フレームか，またはRSTARTかRSTOPコマンドのみである。

RSTART，またはRSTOPコマンドを受け取ったら，PUTHOSTのデータ送信をそれぞれ許可，または禁止する。

SOTコマンドを受け取ったならば，送信バッファ

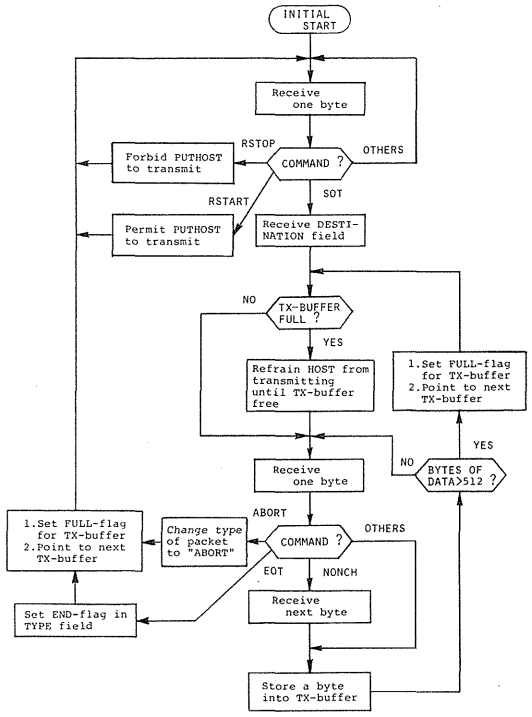

Fig.10 Flow chart for GETHOST.

（TX-buffer）にデータを格納していく。バッファのデータ部が満たされたならば，そこまでを1パケットとし，つぎの送信バッファに移って同じ処理を繰り返す。なお，送信バッファのFULLフラグというのは，そのバッファにパケットが格納されているかどうかを示すフラグである。最後にEOTコマンドが来れば，TYPEフィールドのENDフラグに1をセットし，ABORTコマンドが来れば，パケットのTYPEフィールドのTYPEフラグとABORTフラグを1にセットする。そして，送信バッファのポインタをつぎにすすめ，はじめにもどる。

● PUTHOST

パケッティング階層の受信側を受けもつプロセスである。Fig.11に流れ図を示す。PUTHOSTが送信するのは，SOTコマンドではじまるデータ・フレームか，または送信結果を示すSCCかERRコマンドである。

SCCとERRコマンドは，PUTNETが送信結果を報告してきたときに送信する。

もし受信バッファ（RX-buffer）にパケットが格納されていれば（FULLなら），SOTコマンドを送って

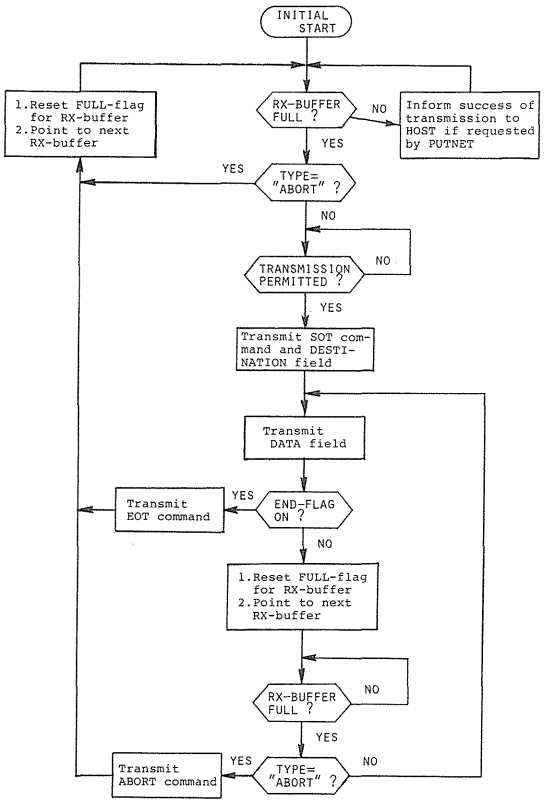

Fig.11 Flow chart for PUTHOST.

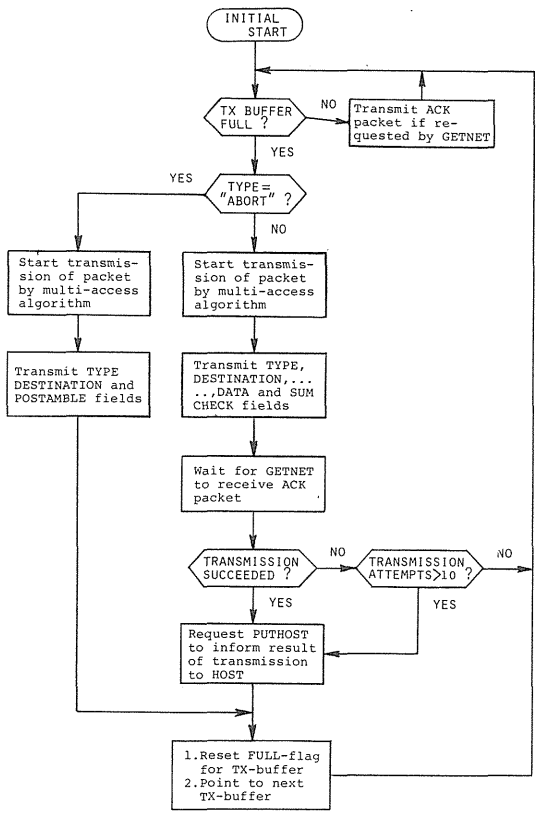

Fig.12 Flow chart for PUTNET.

データの送信を開始する。もちろん，データのビット7が1なら，NONCHコマンドをつけて送る。ENDフラグのセットされているパケットのデータまで送ったならば，EOTコマンドを送信しておわる。途中でタイプがABORTであるパケットになったならば，ABORTを送信しておわる。

● PUTNET

論理リンク階層の送信側を受けもつプロセスである。Fig.12に流れ図を示す。送信バッファにパケットが格納されていれば，データ・パケット，またはABORTタイプのパケットをネットワーク・バスに送信する。もしGETNETからアクノリッジ・パケットの送信を依頼されていれば，それを送信する。どのタイプのパケットを送信する場合も，物理リンクの助けによってアクセス・アルゴリズムを実行してから送信する。

なお，データ・パケットを送信した結果は1パケットごとにホスト・コンピュータに報告するのではなく，ENDフラグのセットされたパケットまでを全部送信してからPUTHOSTにホスト・コンピュータへの報告を依頼する。このことは流れ図には書いていない。

● GETNET

論理リンク階層の受信側を受けもつプロセスである。Fig. 13に流れ図を示す。GETNETはネットワーク上のデータをすべて読み，パケットだったら，そのデスティネーションを調べる。そして自ステーションへのパケットでないなら，そのパケットは読み捨てる。

もし自ステーションへのアクノリッジ・パケットならば，PUTNETに報告する。自ステーションへのデータ・パケットならばソース・アドレスを調べ，もし他ステーションを受信中（まだENDフラグのセットされたパケットが来ていない）なら，受信中であることを示すアクノリッジ・パケットの送信をPUTNETに依頼して，そのパケットを捨てる。これは，複数のステーションからのパケットを，パケッティング階層が混合してデータに組み立ててしまう危険をなくすためである。この部分も流れ図には書いていない。

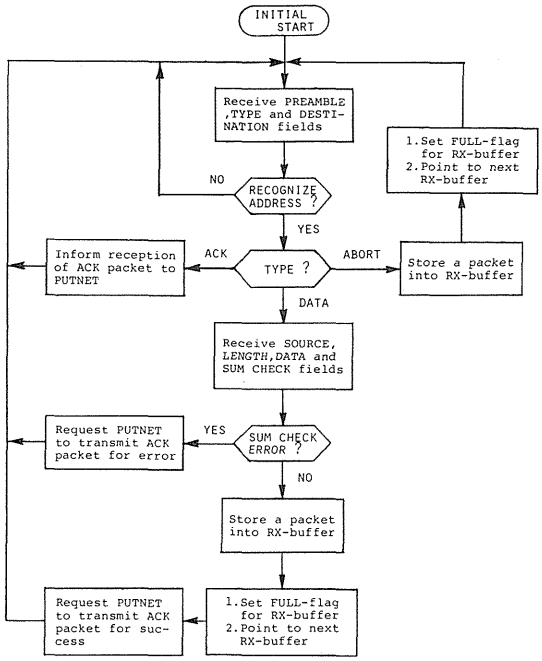

Fig.13　Flow chart for GETNET.

もし取り込むべきデータ・パケットならば，サム・チェックを調べ，結果を示すアクノリッジ・パケットの送信をPUTNETに依頼する。エラーでないならば，受信バッファに格納し，ポインタをつぎの受信バッファにすすめる。

それから，パケットのタイプがABORTなら，そのまま受信バッファに格納し，ポインタをつぎの受信バッファにすすめる。

5. ホスト・コンピュータの通信ソフトウェア

ホスト・コンピュータとして接続してあるのは，現在BASIC MASTER LEVEL - Ⅲ だけであるので，そのプログラムについて簡単に説明する。

ホスト・コンピュータにはコントローラ・リンク階層とユーザ・アプリケーション階層の2階層があるが，LEVEL - Ⅲ ではすべての機能をBASIC言語と機械語サブルーチン5つでインプリメントしたため，2つの階層の明確な区別はしていない。プログラムの機能は以下のとおりである。

- キーボードから入力したメッセージを送信。
- フロッピー・ディスケット上のシーケンシャル・ファイル，ASCIIセーブされたBASICプログラムを送信。
- 受信は，LEVEL - Ⅲ BASICのRS - 232C割込み機能を使うことにより常に可能。
- LEVEL - Ⅲ BASICのkey割込み機能を使うことにより，送信の許可，禁止を常にコントローラに指示することが可能。
- 受信したデータは約8Kバイトのバッファに格納し，フロッピー・ディスクへのセーブとプリンタへの出力が可能。
- 各ステーションをアドレスではなく名前で呼ぶため，ステーション・アドレスと名前をフロッピー・ディスクに登録する。
- BASICを基本に作成したため，処理が遅く，コントローラとの通信速度は最大2400ボーである。

6. あとがき

現在はまだ接続されているステーション数が少ないため，性能についての評価が十分にできず，*The Control-Wire*の有効性を確かめるまでには至っていない。しかし，いくつかのテストを行なうことによりアクセス・アルゴリズムが正しく動作し，衝突の回避を行なっていることは確認できた。また，ネットワーク・コントローラ全体も正常に動作していることが確認できたので，今後はステーション数を増やすことにより負荷を高め，限界での動作と能力を調べることも必要となろう。さらに，ホスト・コンピュータの通信ソフトウェアについてもまだ実用的なものとは言えないので，さらに機能の充実をはかる必要がある。

謝　辞

共に設計，製作に携わった，現㈱富士通の広川誠君に感謝します。

参考文献

(1) K.P. Eswaran, V.C. Hamacher, and G. S. Shedler, "Collision-free access control for computer communication bus networks," IEEE Trans. Software Eng., vol.SE - 7 ,

no.6, Nov. 1981.

(2) V.C. Hamacher, G.S. Shedler, "Collision-free local area bus network performance analysis," IBM J.Res.Develop., vol. 25, no.6, Nov. 1981.

(3) M. Fine, F.A. Tobagi, "Demand assignment multiple access schemes in broadcast bus local area networks," IEEE Trans. Comput., vol.C-33, no.12, Dec. 1984.

(4) 小畑正貴, 田中敏幸, 山元賢治, "Z80パソコン／マイコン用ローカルネット・コントローラの製作", インターフェイス, CQ出版社, 1983. 8, No.75, pp.200－216.

(5) 阿江忠, "ローカル・ネットワーク技術の基礎と実際", CQ出版社 (1983).