Acknowledging Ethernet System の設計

Design of Acknowledging Ethernet System

田丸 喜一郎, 所 眞理雄
Kiichiro TAMARU and Mario TOKORO
慶応義塾大学 工学部, 〒223 横浜市
Faculty of Engineering, Keio University, Yokohama 223

本論文は，ローカル・コンピュータ・ネットワークに適した低価格で信頼性の高い通信方式 Acknowledging Ethernetを提案する。これは，最低位のプロトコルにアクノレッジ機構を付加したEthernetである。アクノレッジ・パケットを衝突させることなく直ちに送信者に返すアルゴリズムが示される。このアルゴリズムを採用することにより，実効転送率および応答時間をアクノレッジ機構を持たない場合のそれぞれ2倍以上に改善できることが計算機シミュレーションによって示される。また，インタフェース・ハードウェアとプロトコルの設計が示され，Acknowledging Ethernetの性能価格比の高さ，そしてホスト・オペレーティング・システムにおける負荷の低さが結論される。

1. はじめに

近年，建物内あるいは敷地内に分散している各種の計算機を接続し，資源の共有やメッセージ交換を行なおうとする動きがある。このような計算機ネットワークはローカル・コンピュータ・ネットワークと呼ばれる。ローカル・コンピュータ・ネットワークはリモート・ネットワーク（ARPAなど一般に公衆回線を使用する）とマルチプロセッサ（BCC 500, C.mmpなど）の間に位置し，低価格で信頼性が高く，しかも転送容量が大きく，ホスト計算機の負荷が小さいことが要求されている。

ローカル・コンピュータ・ネットワークに適した通信方式としてリング（DCS）方式[2,3]とEthernet方式[4]が知られている。これらはいずれもビット直列転送によるパケット交換方式である。DCSでは，リング上を巡環している"トークン"を捕えたステーションがパケットの送信を開始する一種のラウンド・ロビン方式を転送権決定に用いることにより，複数パケットの同時転送（パケットの衝突）を防いでいる。またパケットを受信したステーションは，受取ったパケットの最後尾にACK／NAKを印して送信ステーションに送るので，パケットが環を一巡することにより送信ステーションが受信状態を知ることができる。

Ethernetは転送権の決定を一意的に行なうことはせず，同時にいくつかのステーションが送信を開始する可能性を認める一方，衝突を検出して衝突が続けて起こらないように再転送を制御する衝突制御機構を各ステーションが備えている。ACK／NAKは一般のパケットと同様に転送され，このための特別な機構を持たない。

DCSは1つの送信機が1つの受信機のみに送信するため，Ethernetよりも高い転送容量を持つことができる。一方あるステーションから他のステーションに送られるパケットは多数の送受信機（リピータ）を通過するため（アク

ティブな伝送路），信頼性の点ではEthernet（パッシブな伝送路）の方が優れている。また，転送権の取得に関しても，トークンによる集中型の制御を持つDCSに比べ，制御が完全に分散されているEthernetの方が信頼性が高い。このようにEthernetはローカル・コンピュータ・ネットワークとして大変優れた方式である。しかし，EthernetにはDCSのようなパケットの受信に対する確認の手段が最低位のプロトコルに備わっていないために，ホスト計算機がACK/NAKパケットを作って送信ステーションに送り返さねばならない。このことは，

(1) ホスト計算機の負荷を増大させる
(2) 応答時間が長くなる
(3) 受信確認のパケットが他のパケットと衝突を起こして転送効率を悪化させる

原因となる。

本論文では以上の問題を解決するために，低価格，信頼性を維持しつつ受信ステーションのインタフェースが自動的にアクノレッジ・パケットを送信する一方法を提案する。このようなEthernet方式をAcknowledging Ethernetと名づける。以下，衝突を起こさずにアクノレッジ・パケットを返送するアルゴリズム，Acknowledging Ethernet方式の性能シミュレーション，インタフェース・ハードウェア並びにプロトコルの設計，性能価格比の評価について述べる。

2. アクノレッジ・アルゴリズム

Ethernetにおける転送の単位はパケットである。メッセージはパケットの形で転送される。メッセージの長さが決められた最大パケット長より長い場合には，メッセージはいくつかのパケットに分割され転送される。パケットを受信したステーションは一般に送信ステーションにアクノレッジ・パケットを返送する。1つのメッセージがいくつかのパケットに分割されて転送される場合には，2通りのアクノレッジの方法がある。

(1) 1つのメッセージを構成する一連のパケットを受信した後にアクノレッジ・パケットを転送する（メッセージ毎のアクノレッジ方式）。
(2) 1つのパケットを受信する毎にアクノレッジ・パケットを転送する（パケット毎のアクノレッジ方式）。

(1)の方法は通信路の使用頻度は少ないが，(2)は転送時のエラーの検出，エラーの回復を敏速に行なえる。また(2)に加え，送信ステーションが送信終了後直ちにアクノレッジ・パケットを受信できることが保証されれば（パケット毎の瞬時アクノレッジ方式），即座に転送状態が以下のいずれであるか知ることができる。

(1) ACKパケットを受信することによって転送が成功したこと。
(2) NAKパケットを受信することによって転送が不成功であったこと。
(3) 応答がなかったこと（タイム・アウト）によって，相手が故障または存在しないこと。

従って信頼性の点から，パケット毎の瞬時アクノレッジ方式を採用すべきである。また，応答時間が短縮され，この結果システム全体の性能を改善する。しかしながら，パケット毎の瞬時アクノレッジ方式を実現するためには，アクノレッジ・パケットが他のパケットと衝突しないことが保証されなければならない。

データ・パケットを受信した後，アクノレッジ・パケットを他のパケットと衝突することなく直ちに転送する方法として以下の方法が考えられる。

(1) データ・パケットを転送するケーブルとは別のケーブルをアクノレッジ・パケットを転送するために用いる方法。
(2) データ・パケットとアクノレッジ・パケットを異なった周波数で転送する方法。
(3) アクノレッジ・パケットの転送をデータ・パケットの転送に対して優先させる方法。

(1)，(2)の方法は原理的には非常に単純であるが，両者とも2組の送受信機が必要であり，そのためハードウェア価格が増加し，信頼性が減少する。また(1)は2本のケーブルを必要とし，(2)は変調方式を制限する。これに比べ，(3)の方式は余分なケーブルや送受信機を必要とせず，価格と信頼性を損なわない。

アクノレッジ・パケットをデータパケットの転送に優先させる方法を実現するために，我々は基本待ち時間（Basic Waiting Time：BWT）を導入した。基本待ち時間とは，データ・パケットの転送後直ちに新たなパケットの転送が開

始されたかをすべてのステーションが知ることができる時間であり，これはケーブルの長さによって定まる最大伝搬時間の2倍として定義できる。以下にこの基本待ち時間を用いたアクノレッジ・アルゴリズムを示す。また，その流れ図を図1に示す。

[I] データ・パケットの送信

(1) Etherがサイレント，すなわち誰も使っていない状態になるまで待つ。Etherがサイレントになったらステップ(2)へ。

(2) 基本待ち時間を待つ。その後もEtherがサイレントであればステップ(3)へ。Etherがサイレントでなければステップ(1)へ。

(3) データ・パケットを送信する。データ・パケットが他のパケットと衝突した場合にはステップ(4)へ。衝突せずに送信が終了すればステップ(5)へ。

(4) 衝突制御アルゴリズムに従って再転送間隔を決定する。そしてその時間を待った後ステップ(1)へ。

(5) アクノレッジ・パケットの受信を待つ。送信後基本待ち時間を過ぎてもACK/NAKを受信しなかったならタイム・アウトとなる。

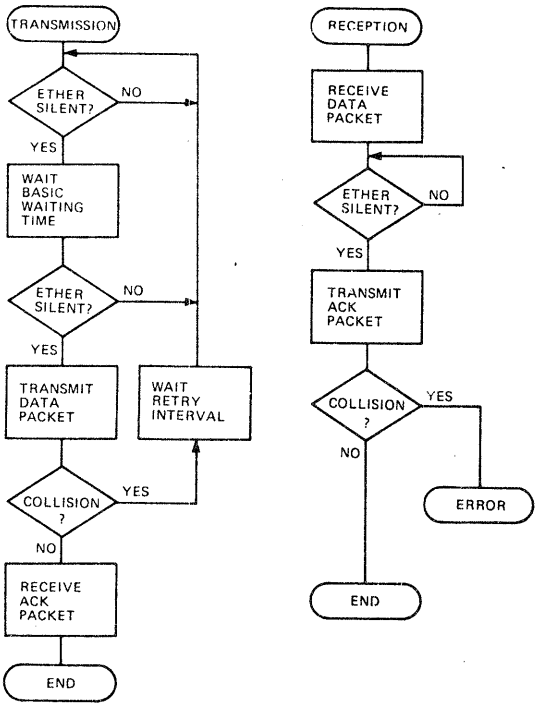

図1 アクノレッジ・アルゴリズム

[II] データ・パケットの受信

(1) データ・パケットを受信する。受信終了後(Etherがサイレントになった後)ステップ(2)へ。

(2) 直ちにアクノレッジ・パケットを送信する。アクノレッジ・パケットが他のパケットと衝突した場合は異常終了となる。

以上のアルゴリズムを要約すると，データ・パケットの送信時にはEtherがサイレントになった後さらに基本待ち時間待ってから送信するのに対して，アクノレッジ・パケット送信時にはEtherがサイレントになった後直ちに送信することによって，アクノレッジ・パケットの転送を他のデータ・パケットの転送に優先させるものである。ここで，基本待ち時間はデータ・パケットの転送時間と比較すると非常に短いため(ケーブル長1km，転送レート1Mbps，4Kbits/packetのとき1/600以下)，応答時間の短縮，そして転送効率の改善が期待できる。

図2はデータ・パケットとアクノレッジ・パケットの転送の様子を示す。ステーションAがステーションCに対してデータ・パケットを転送している間にステーションBがデータ・パケットの転送を要求したとする。ステーションCはデータ・パケットの受信終了後直ちにステーションAに対してアクノレッジ・パケットを転送する。一方ステーションBはステーションAからステーションCへのデータ・パケットの転送終了後，さらに基本待ち時間待った後，Etherがサイレントであるか否かを調べる。その時点ではステーションCからステーションAへのアクノレッジ・パケットによってEtherが使われている。そのためステーションBは再びEtherがサイレントになるまで待つ。このようにして，アクノレッジ・パケットは衝突を起こすことなくステーションAに転送される。ステーションBはこのアクノレッジ・パケットの転送後，再び基本待ち時間待ち，データ・パケットの送信を開始する。

ここに述べた方式以外にも，アクノレッジ・パケットの転送をデータ・パケットに優先する方式が考えられる。例えば，次のようなものである。

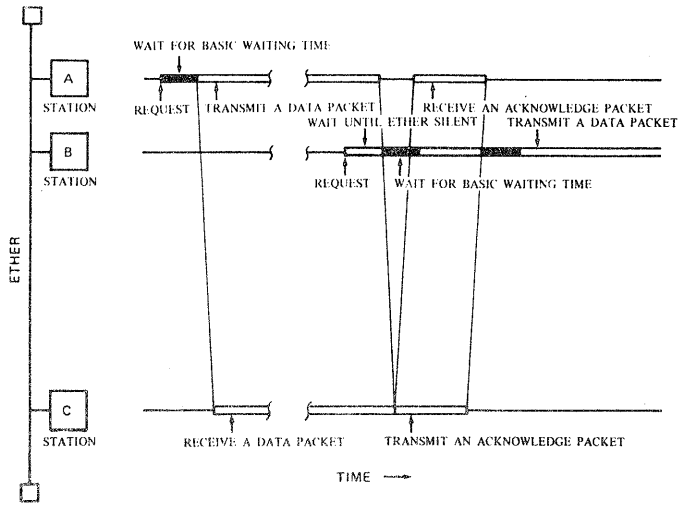

図2 アクノレッジ・パケットがデータ・パケットに優先して転送される様子

(1) データ・パケットもアクノレッジ・パケットも区別なく，Etherがサイレントになったら直ちに送信を開始できる。もしも衝突が起こったら，アクノレッジ・パケットを転送しようとしたステーションは直ちにそのアクノレッジ・パケットを再転送することができる。一方データ・パケットを転送しようとしたステーションは最低でも基本待ち時間を待つように再試行の間隔（retry interval）を決定し，その時間待った後にもう一度転送を試みる。

(2) 前回の転送がデータ・パケットの転送であったかアクノレッジ・パケットの転送であったかを常に監視しておく。アクノレッジ・パケットを転送しようとするステーションはEtherがサイレントであれば直ちにそのパケットを転送できる。データ・パケットを転送しようとするステーションは，前回の転送がアクノレッジ・パケットの転送であったならEtherがサイレントになった後直ちにデータ・パケットを送信できる。前回がデータ・パケットの転送であったならEtherがサイレントになった後さらに基本待ち時間を待って，このときEtherがサイレントであったなら転送を開始できる。サイレントでなかったら，今送られているパケットの種類に従ってもう一度方針を決定する。

(1)の方法は，衝突の検出に要する時間が短い場合には，先に述べた方法よりも転送効率が多少よい。しかし，異常が発生した場合にシステムをデッドロックに導く可能性がある（例えば2つ以上のステーションがアクノレッジ・パケットを転送したとき）。そしてこれを回避するための代価はほんの少しの転送効率の向上に見合わない。

(2)の方法は，先に述べた方法を転送効率の面から改良しているが，これもやはり常にEtherの状態を監視しなければならない点や，前回の転送が衝突を発生していた場合の処置などが大変困難で，性能価格比として優れていない。（送信しているステーション以外のステーションが，Ether上で衝突が起こった事を検出するのは大変むずかしい。）

このような検討から，先に述べたアクノレッジ・アルゴリズムをAcknowledging Ethernet Systemに採用しようと考えている。

3. 性能評価

Acknowledging Ethernet の性能を評価するために計算機シミュレーションを行なった。シミュレーションは，Acknowledging Ethernet とアクノレッジ機構を持たない Ethernet に対し応答時間と実行転送容量についてなされ，最終的には実効転送容量に対する応答時間が求められ，両者の比較がなされた。ここで応答時間とは，ホスト計算機により Ethernet のインタフェースに対してデータ・パケットの送信要求が出されてから，そのデータ・パケットが送信され，これに対するアクノレッジ・パケットをインタフェースが受信するまでの時間である。また，実行転送容量とは単位時間に転送されたデータ・パケットのビット数である。衝突を起こしたデータ・パケットやアクノレッジ・パケットは実効転送容量に含まれない。

衝突が生じた際に用いられる衝突制御アルゴリズムは Ethernet の性能に大きく影響する。シミュレーションでは単純で有効な次に示す2種のアルゴリズムを用いた。

(A) Binary Exponential Backoff [2]. これは衝突の回数により再試行の間隔を決定するアルゴリズムであり，衝突が起こる度に再試行間隔を2倍にする。

(B) 転送が成功した最新のパケットの受信ステーションのアドレスと自分のアドレスとの差で再試行間隔を決定するアルゴリズムである。例えば最新のパケットの受信ステーションのアドレスが10のとき，アドレス15のステーションの再試行間隔は5τ，アドレス20のステーションの再試行間隔は10τである(τは定数)。

(A)はオリジナルな Ethernet に使われたものである。(B)は常に Ether 上のパケット転送をすべてのステーションが監視していなければならないが，アドレス付けおよびτを適当に選ぶことにより，衝突の相手が変わらなければ必ず1度の再試行で転送が成功する効率のよいアルゴリズムである。このアルゴリズムはデッドロックを起こす危険があるため実現されることは無いと思われるが，ここでは性能の比較のために採用した。

その他，シミュレーションには以下の事項を仮定した。

(1) パケット毎にアクノレッジを返す。

(2) インタフェースは，1つのデータ・パケットの送信を終了し，それに対するアクノレッジ・パケットを受信するまで他のデータ・パケットを送信しない。しかしデータ・パケットの受信およびこれに対するアクノレッジ・パケットの送信は行なう。

(3) 応答時間を比較するため，データ・パケットの受信からそれに対するアクノレッジ・パケットの送信までの時間は0とする。

(4) データ・パケットの送信要求間隔は，データ・パケットの受信に影響されない。また，1つのデータ・パケットの送信要求が出されたときに前のデータ・パケットの送信が完了していないときには，その送信の完了後直ちに要求が出される。この送信要求間隔は，平均mに対して分散$m/3$の正規分布を持つ乱数とする。

シミュレーションで用いたパラメータを表1に示す。ケーブル長，最大転送容量，プロセッサ(ホスト計算機)の数はローカル・コンピュータ・ネットワークにおける一般的な値を選んだ。また，データ・パケット長，アクノレッジ・パケット長，送信要求間隔は典型的な値を選んだ。

表1 シミュレーションに用いたパラメータ

ケーブル長	1 Km
基本待ち時間	10 μsec
最大転送容量	1, 8 Mbps
データ・パケット長	4 Kbits†
アクノレッジ・パケット長	128 bits††
プロセッサの数	2, 4, 8, 16, 32, 64, 128
送信要求間隔	平均 0.1 sec
タイム・アウト	1 sec
シミュレーション時間	10 sec

† パケット当りの転送時間は 1 Mbpsのとき 4 msec, 8 Mbpsのとき 0.5 msec。

†† パケット当りの転送時間は 1 Mbpsのとき 0.1 msec, 8 Mbpsのとき 0.016 msec.

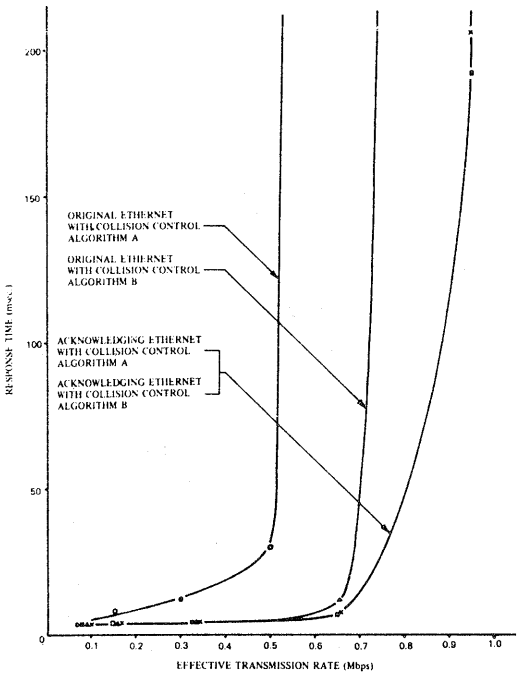

図3 実効転送容量に対する応答時間
(1 Mbpsのとき)

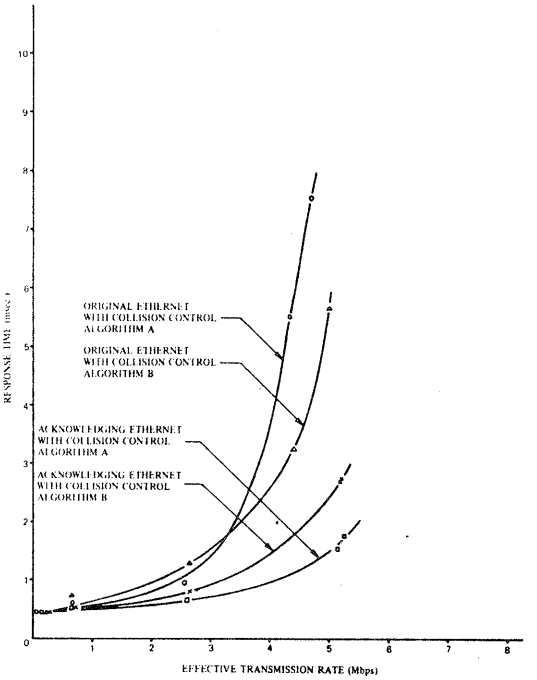

図4 実効転送容量に対する応答時間
(8 Mbpsのとき)

図3および図4に，アクノレッジ機構をもたないEthernet (Original Ethernet) に衝突制御アルゴリズム (A) と (B) を適用したものと，Acknowledging Ethernetに衝突制御アルゴリズム (A) と (B) を適用したもののそれぞれについて，実効転送容量に対する応答時間を示した。図3は最大転送容量を1 Mbpsとした場合であり，図4は8 Mbpsの場合である。これらは，プロセッサ数に対する応答時間，およびプロセッサ数に対する実効転送量のシミュレーション結果から求められたものである。

実効転送容量が増加するに従って応答時間は長くなっているが，アクノレッジ機構を持ったものは応答時間の増大がゆるやかである。最大転送容量の半分の実効転送容量を得るときの応答時間は，Original Ethernetの (A) とAcknowledging Ethernetの (A) を比べると4倍以上改善されており，応答時間から実際にパケットの転送に要する時間を引いた場合（応答のオーバヘッド時間）は7倍以上改善されている。また，Original Ethernetでは最大転送容量の約50%までしか実効転送容量として利用できないのに対し，Acknowledging Ethernetでは70%以上を利用することができる。

さらには，Original Ethernetにおいては (A) よりも (B) の方が高い性能を示すのに対し，Acknowledging Ethernetにおいては逆に (A) の方が高い性能を示す。従って，アクノレッジ機構を持つことにより，簡単な衝突制御アルゴリズムを用いた場合にも大変高い性能が得られることがわかる。

4. インタフェース・ハードウェアの設計

インタフェース・ハードウェアは，ローカル・コンピュータ・ネットワークに対する要求から，低価格で信頼性が高く，しかも接続に伴なうホスト計算機の負荷が軽いことが必要である。Acknowledging Ethernetのためのインタフェースは以上の要求を満たすように設計され，以下のような特徴を持つ。

(1) 自己診断機能，テスト機能，さらに故障

に際して物理的にインタフェースをEtherケーブルから切り離す機構を持ち，信頼性が高い。

(2) 最低位プロトコルおよびエラー回復を含めたより高位のプロトコルの実行に必要な各種の機能を備えているため，ホスト計算機の負荷が軽い。

(3) (2)の機能をマイクロプロセッサを用いたインテリジェント・インタフェースとして実現しているため，価格が低い。

(4) ホスト計算機の差異を吸収する部分を分割して構成したので拡張性が高い。

4.1 インタフェース・ハードウェアの構成と機能

インタフェース・ハードウェアは，送受信機，Acknowledging Ethernet インタフェース，ホスト・インタフェースから構成される。図5にその構成を示す。

[I] 送受信機

送受信機は，Etherケーブルと密着して固定されるアナログ部分である。その主な機能は信号レベルの変換であり，Acknowledging Ethernet インタフェースからの論理レベルの信号を増幅してEtherケーブルに送信する送信機，Etherケーブルの信号を受信して論理レベルの信号に変換する受信機により構成されている。また，故障の際に，アナログの検出回路あるいはAcknowledging Ethernet インタフェースからの指令によってEtherケーブルから物理的に切り離す保護機構を備えている。送受信機の詳細は別の機会に発表する予定である。

[II] Acknowledging Ethernet インタフェース

Acknowledging Ethernet インタフェースは，最低位プロトコルおよびエラー回復を含んだより高位のプロトコルによるパケット転送のための各種の機能を持ったディジタル部分である。それらの機能を以下に列挙する。

(1) 変調と復調

Acknowledging Ethernet では，データは位相変調されてEtherケーブルに送信される。変調および復調はそれぞれ，

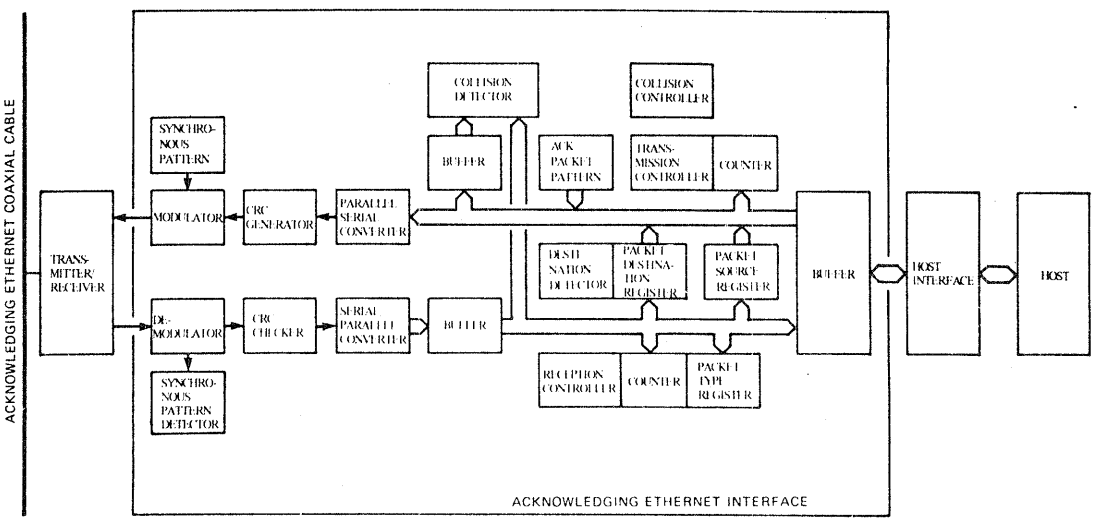

図5 インタフェース・ハードウェアの構成

ディジタル信号を位相変調する機能，位相変調された信号をディジタル信号に変換する機能である。

(2) 同期検出
Ethernetケーブル上のパケットを検出する機能である。パケットの先頭に付けられた8ビットの同期パターンにより検出を行なっている。

(3) エラー検出
パケットのEthernetケーブル上での雑音などによる転送エラーを検出する。パケットの送信時に付けられたCRCを受信時に検査し，エラー検出を行なっている。

(4) 直並列変換
Ethernetケーブル上ではパケットは直列転送されるのに対して，インタフェース内ではバイト単位に取り扱う。そのための変換を行なう機能である。

(5) 衝突検出
パケットの送信時にEthernetケーブル上の信号を受信して，送信しているデータと論理レベルで比較して衝突検出を行なう。また，受信時のエラー検出によっても衝突検出を行なっている。

(6) アドレス検出
パケットの受信アドレスを8ビットのステーション物理アドレスと24ビットのプロセス名で検出する。プロセス名を含めたアドレス検出をインタフェース内で行なっているため，ホスト計算機の負荷が軽減されている。

(7) アクノレッジ・パケットの生成と送信
データ・パケットを受信するたびに，受信状態に対応したアクノレッジ・パケットを直ちに生成しそれを送信する機能である。この機能によってパケット毎の瞬時アクノレッジ方式が実現される。

(8) 送信制御と受信制御
パケットの送信時および受信時におけるAcknowledging Ethernetインタフェース内部の各機能のタイミングを調整し，その動作を制御する機能である。

(9) 衝突制御
データ・パケットが衝突した際の再転送間隔を決定する衝突アルゴリズムを実行する機能である。

(10) エラー回復
パケットの転送におけるエラーを，インタフェースがパケットを再転送することによって回復する機能である。

(11) バッファ管理
インタフェース内部に，送信用に1つ受信用に2つのバッファを持ち送受信の効率化を図っている。これらのバッファを管理する機能である。

(12) 診断
インタフェースおよびホスト計算機の故障を検出する機能である。インタフェースの故障検出は，自分に対するパケットを転送することによって行なう。また，他のインタフェースあるいはホスト計算機が正常に動作しているかどうかをテスト用のパケットを転送して検出する。

Acknowledging Ethernetインタフェースは以上の機能を実現する機構により構成されるが，衝突制御，エラー回復，バッファ管理，診断の機能はマイクロプロセッサにより実現される。

[III] ホスト・インタフェース

ホスト・インタフェースは，ホスト計算機の差異を吸収して，ホスト計算機とAcknowledging Ethernetインタフェースを接続するディジタル部分で，その構造・機能はホスト計算機によって異なる。これには，Acknowledging Ethernetインタフェース内のバッファとホスト計算機内のバッファを結ぶDMA機能が含まれる。

4.2 インタフェースの動作

パケットの送信および受信におけるインタフェースの動作を以下に示す。

[I] パケットの送信

(1) ホスト計算機がパケットの送信を要求すると，ホスト・インタフェースはDMA機構によってホスト計算機内のバッファにあるパケットをAcknowledging Ethernetインタフェース内のバッファに転送する。また，ホスト計算機からの送信コマンドをAcknowledging Ethernetインタフ

ェースに送る。

(2) Acknowledging Ethernetインタフェースは，送信コマンドを受けるとパケット長をカウンタにセットして送信制御機構に対して指令を出す。

(3) 送信制御機構は，送信が可能になると送信機をEtherケーブルに接続して送信を開始する。

(4) 送信制御機構は，データをバッファから取り出し，並直列変換し，CRCを付加し，変調機でパケットの先頭に同期パターンを付加して送信機に送る。

(5) 送信中に，送信データとEtherケーブルから受信したデータを比較して衝突検出を行なう。衝突を検出した場合には，送信を中断して衝突制御機構により再転送間隔を決定し，再送信を行なう。

(6) パケットの送信を終了すると，Acknowledging Ethernetインタフェースは送信機をEtherケーブルから切り離し，対応するアクノレッジ・パケットの受信を待つ。

(7) アクノレッジ・パケットを受信し，送信が正常あるいは回復不可能なエラーである場合には，ホスト計算機に割込み送信終了情報を送る。また，回復可能なエラーであれば再送信を行なう。

[II] パケットの受信

(1) Etherケーブル上の同期パターンを検出すると，復調機は受信制御機構に指令を出しパケットの取り込みを開始する。

(2) 受信制御機構は，パケットのCRCを検査した後データを直並列変換してインタフェース内のバッファに入力する。

(3) パケットの受信アドレスがステーション・アドレスと一致すると，バッファへの入力を続ける。アドレスが一致しない場合には，バッファへの入力を中断し再び同期パターンの検出を行なう。

(4) バッファへの入力と同時に，パケットの種類とパケット長をレジスタとカウンタにセットする。

(5) 受信したパケットがデータ・パケットであれば，受信終了後受信状態に応じたアクノレッジ・パケットを生成し，送信制御機構に対してアクノレッジ・パケットの送信を指令する。

(6) 送信制御機構は，前述の送信手順に従って直ちにアクノレッジ・パケットを送信する。

(7) パケットの受信が正常であった場合には，Acknowledging Ethernetインタフェースはホスト計算機に割込み受信を知らせる。エラーがあった場合にはパケットを破棄する。

(8) ホスト計算機の要求によって，ホスト・インタフェースは，DMA機構でAcknowledging Ethernetインタフェース内のバッファからホスト計算機にパケットを転送する。

5. プロトコル

プロトコルは，ネットワークに接続されているホスト計算機内のプロセスが他のホスト計算機内のプロセスと通信するための規約である。Acknowledging Ethernetでは，プロトコルの処理を機能的に分割できるように階層的な4レベルのプロトコルを設定した。図6にその構成を示す。Acknowledging Ethernetのプロトコルは，低位のプロトコルでプロセス名を含めたアドレス検出およびEtherケーブル上でのパケットの送受信に対するエラー回復を行なうので，ホスト・オペレーティング・システムの負荷が軽減できることが特徴となっている。

5.1 物理レベル・プロトコル

物理レベル・プロトコルは，物理的な通信ラインの制御に関する最低位のプロトコルで，Etherケーブル上でのパケットの送受信がその主な機能である。これらの機能は，インタフェース・ハードウェアで処理される。

(1) パケットの構成

図7にパケットの構成を示す。パケットはCRCで区切られたブロックにより構成されるが，最初の2ブロックがヘッダー部，第3ブロック以後がデータ部である。各フィールドは以下の意味を持つ。

(a) SYNC

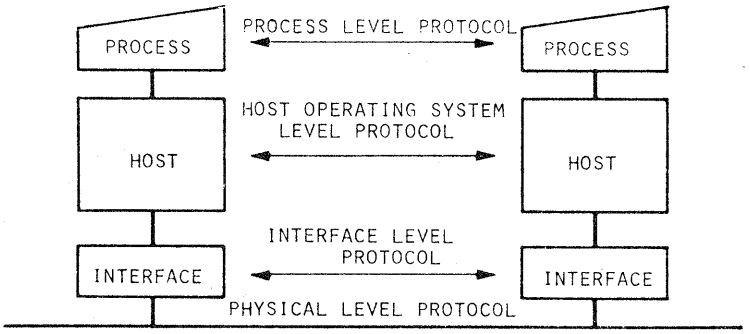

図6　プロトコルの構成

同期検出のための同期パターン(8ビット)。

(b) STATION NUMBER
送信ステーションあるいは受信ステーションのステーション物理アドレス（8ビット）。

(c) PROCESS NAME
送信プロセスあるいは受信プロセスの名前（24ビット）。

(d) TYPE
パケットの種類（16ビット）。

(e) LENGTH
バイト単位で示したパケット長（16ビット）。

(f) SEQ
データ・パケットとアクノレッジ・パケットの対応を付けるためのパケット番号（8ビット）。データ・パケットを受信したステーションは，返送するアクノレッジ・パケットのSEQフィールドに，受信したデータ・パケットのSEQフィールドの値をセットして送信する。

(g) DATA
データ部。アクノレッジ・パケットでは一般にこのフィールドはない。

(2) パケットの種類
パケットには以下の種類がある。

(a) データ・パケット
データ転送用のパケット

(b) アクノレッジ・パケット
データ・パケットに対する応答用のパケットで，以下の受信状態をTYPEフィールドに示す。

(i) 正常に受信した。

(ii) ステーション物理アドレスが一致したにもかかわらずプロセスが存在しなかった。

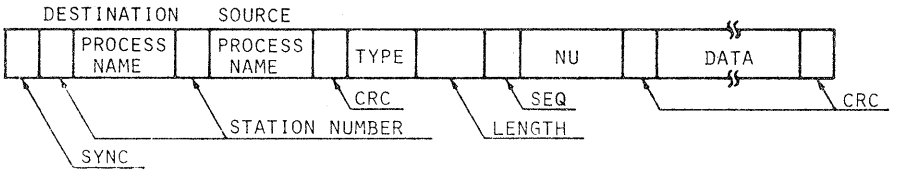

図7　物理レベル・プロトコルにおけるパケットの構成

(iii) LENGTH フィールドに書かれたパケット長と受信したパケット長が一致しなかった。
(iv) CRC エラーが発見された。
(v) パケットの構成が無効であった。
(vi) インタフェース内のバッファが満杯だった。
(vii) ホスト計算機が故障している。

(c) ブロード・キャスト・データ・パケット
複数のステーションに対して送信されるデータ転送用のパケット。

(d) テスト用パケット
他のステーションのインタフェースあるいはホスト計算機が正常に動作しているかどうかをテストするために用いるパケット。

(3) プロセスの識別方法
送信プロセスあるいは受信プロセスの識別は，ステーション物理アドレスとプロセス名で行なう。これを物理レベル・プロトコルで吸収するため，インタフェースのアドレス検出機構はプロセス名を格納する連想メモリを備えている。連想メモリの内容はホスト・オペレーティング・システムで書き換え可能である。ステーション物理アドレスは，ネットワーク内では唯一に定まるものであるが，アドレス0はブロード・キャスト・データ・パケットのステーション物理アドレスとして用いる。これを用いることによって，複数の同種プロセスに対して同時にパケットを転送することが可能である。また，プロセス名0，1は，それぞれインタフェースおよびホスト・オペレーティング・システムに対するプロセス名として用いる。

(4) パケット送受信に対する操作
物理レベル・プロトコルにおけるパケットの送受信に対する操作を以下に示す。

(a) データ・パケットの受信
受信したデータ・パケットのヘッダー部にエラーがある場合には，衝突したパケットであるとして無視する。その他の場合には，受信状態に対応するアクノレッジ・パケットを転送する。受信したデータ・パケットにエラーがある場合にはそのパケットを破棄する。

(b) ブロード・キャスト・データ・パケットの受信
これは複数のステーションに対して送信されるデータ・パケットであるためアクノレッジ・パケットは転送しない。受信したブロード・キャスト・データ・パケットにエラーがある場合には，そのパケットを破棄する。

(c) アクノレッジ・パケットの受信
アクノレッジ・パケットは転送しない。

(d) データ・パケット，ブロード・キャスト・データ・パケットの送信
送信中のエラーは，送信データとEtherケーブルからの受信データとの比較およびCRC検査により検出する。エラーを検出した場合は送信を中断する。エラーを生じた場合の処理は高位のプロトコルであるインタフェース間プロトコルにまかされる。

(e) アクノレッジ・パケットの送信
送信中にエラーを検出した場合は送信を中断し再転送は行なわない。また，そのアクノレッジ・パケットに対応するデータ・パケットは破棄する。

5.2. インタフェース間プロトコル

インタフェース間プロトコルはインタフェース間のパケットの送受信に関するプロトコルで，Etherケーブル上でのパケット転送におけるエラー回復がその主な機能である。これらの機能は，インタフェースのファームウェア（ソフトウェア）で処理される。

(1) パケットの構成
図8にパケットの構成を示す。同期パターンおよびCRCがないことを除くと基本的に物理レベル・プロトコルにおけるパケットと，構成および各フィールドの意味は同じである。

(2) 送信中のエラーが最低位プロトコルで発生した場合の操作

(a) ヘッダー部で検出された場合
衝突であると解釈する。衝突制御機構により再試行間隔を決定し，その間隔を待った後に再転送を行なう。

(b) データ部で検出された場合

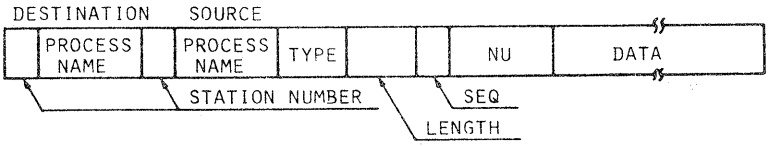

図8　インタフェース間プロトコルにおけるパケットの構成

Ethernetケーブルの雑音あるいは故障であるとして再転送を行なう。このエラーが繰り返される場合には異常終了する。

(3) アクノレッジ・パケットを受信したときの操作

アクノレッジ・パケットに対して次の2点を検査する。

(a) アクノレッジ・パケットの送信プロセスが，送信したデータ・パケットの受信プロセスと一致するか。

(b) アクノレッジ・パケットの番号が，送信したデータ・パケットの番号と一致するか。

この2点が一致しない場合には送信エラーとして再転送を行なう。一致した場合にはアクノレッジ・パケットの内容により以下の操作を行なう。

(a) 正常を示す場合

データ・パケットの送信を正常終了する。

(b) プロセスの不在またはホスト計算機の故障を示す場合

回復不可能のエラーとして異常終了する。

(c) その他のエラーを示す場合

回復可能なエラーとしてデータ・パケットを再転送する。

また，アクノレッジ・パケットを受信できなかった場合は，再転送を行なう。同じデータ・パケットに対して再転送が繰り返される場合には，故障として異常終了する。

5.3 ホスト・オペレーティング・システム間プロトコル

ホスト・オペレーティング・システム間プロトコルは，ホスト・オペレーティング・システム間のパケットの送受信に関するプロトコルである。プロセスからのメッセージは，パケットに分割され転送される。パケットの転送に関しては，プロセス名を含めたアドレス検出，パケット転送時のエラー回復を低位のプロトコルが処理するため，ホスト・オペレーティング・システム間プロトコルは，メッセージを構成するパケットの転送における通番制御を行なうだけである。そのため，ホスト・オペレーティング・システム間プロトコルは簡素化されており，ホスト・オペレーティング・システムの負荷は軽い。このプロトコルは，実際にはホスト計算機とインタフェース間の以下の手続きに従って実行される。

(1) インタフェースに対するコマンド

主なコマンドには以下のものがある。

(a) 送信要求

パケットの送信を要求する。

(b) 送信中断

送信要求したパケットの送信を中断する。

(c) 受信要求

受信パケットをホスト計算機に転送させる。

(d) 受信中断

ホスト計算機への転送を中断させる。

(e) 初期化

インタフェースを初期化する。インタフェースの診断を行なった後，送受信機をEthernetケーブルと接続する。

(f) 終了

送受信機をEthernetケーブルから切り離し，インタフェースの機能を停止する。

(g) プロセス名の登録

受信プロセス名をインタフェースのアドレス検出機構に登録する。

(h) プロセス名の抹消
受信プロセス名を抹消する。

(2) ホスト計算機への割込要因
主な割込要因には以下のものがある。

(a) 送信終了
送信を依頼されたパケットの送信を終了した。

(b) 受信
他のステーションからのパケットを受信した。

(c) 故障
Ethernetケーブルあるいはインタフェースの故障を発見した。

(d) 無効コマンド
インタフェースに対するコマンドが解釈できなかった。

(e) その他
初期化コマンド，終了コマンドの実行を終了した。

5.4 プロセス間プロトコル

プロセス間プロトコルは，プロセス間のデータの授受に関する最高位のプロトコルである。このレベルのプロトコルは，通信方式あるいはハードウェアに依存しないプロトコルである。プロセス間プロトコルは，その用途やオペレーティング・システムにより異なる。

6. 性能価格比の評価

ここでは，Acknowledging Ethernetの他にアクノレッジ機構を持たない二つのEthernetを考え，4章でのインタフェース・ハードウェアの設計および5章でのプロトコルの設計をもとにAcknowledging Ethernetの性能価格比を評価する。

Ethernetを実現するためのハードウェアは，アナログ部とディジタル部に大別することができる。アナログ部は同軸ケーブルおよび送受信機，ディジタル部はEthernetインタフェースおよびホスト・インタフェースである。アナログ部の機能はアクノレッジ機能を持つEthernetでも持たないものでも同一であり，従って価格も同じである。一方，ディジタル部はアクノレッジ機能を持つ場合と持たない場合，またどの程度のインテリジェンスを含ませるかによって価格が異なる。

Ethernetの実現に必要な機能をできる限りホスト計算機のソフトウェアに実行させ，インタフェース・ハードウェアの機能を最小限におさえる場合，そのディジタル部は，衝突検出機構，DMA機構，直並列変換機構により実現される。これは約70個のIC（MSI,SSI）で実現でき，価格は約12万円である。しかしながら，衝突検出アルゴリズム，CRCの生成および検査をホスト計算機で実行しなければならないため，ホスト・オペレーティング・システムの負荷は非常に重い。

インタフェースをインテリジェント化して，衝突制御機構，パケット用のバッファ・メモリー，CRC機構を持たせることによりホスト・オペレーティング・システムの負荷を軽減することができる。この場合，これらの機能はマイクロプロセッサおよびMSIによって容易に実現でき，ICの合計は約80個（LSI, MSI,SSI）で，価格は約15万円に見積もられる。

Acknowledging Ethernetでは，さらに，アクノレッジ機構としてデータ・パケットの受信に対してアクノレッジ・パケットを直ちに送信する機能が必要である。この場合，約85個のIC（LSI, MSI, SSI）により実現でき，価格は約17万円である。

表2は上記の3種のインタフェース・ハードウェアのアナログ部，ディジタル部，およびホスト・オペレーティング・システムの負荷をまとめたものである。Acknowledging Ethernetのハードウェア価格は，その機能が増えた分だけ増加しているが，衝突制御機構，パケット・バッファ，CRC機構を備えたものに対してはその差はわずかである。さらに，ホスト・オペレーティング・システムの負荷が非常に軽減されることや，シミュレーション結果からわかるように性能が大幅に向上することを考慮すると，性能価格比は非常に優れていると言える。また，このディジタル部の大部分をLSI 1チップに収めることも可能であり，価格の低下が期待できることから，今後さらに性能価格比が良くなると予想される。

表2　ハードウェア価格とオペレーティング・システムの負荷の比較

	アナログ部の価格	ディジタル部の価格	オペレーティング・システムの負荷
Original Ethernetのインタフェース	1.5～3万円	約12万円 (IC 70個)	重い
衝突制御機構，バッファ，CRCを付加したインタフェース	1.5～3万円	約15万円 (IC 80個)	軽い
Acknowledging Ethernetのインタフェース	1.5～3万円	約17万円 (IC 85個)	非常に軽い

V. おわりに

本論文では，ローカル・コンピュータ・ネットワークに適した通信方式 Acknowledging Ethernet を提案した。これは，ハードウェア・レベルのアクノレッジ機構を Ethernet に付加したもので，エラー検出，エラー回復の能力が優れており，システムの信頼性を高いものにする。

Acknowledging Ethernet は性能の面でもアクノレッジ機構を持たないものよりも優れていることが計算機シミュレーションにより確認された。最大転送容量の半分の実効転送容量のときの応答時間は 1/4 に減少し，また最大実効転送容量は約 1.5 倍に増加している。

インタフェース・ハードウェアとプロトコルの設計がなされ，この結果から Acknowledging Ethernet のインタフェース・ハードウェアの性能価格比の高さ，ホスト・オペレーティング・システムの負荷の小ささが示された。

ローカル・コンピュータ・ネットワークは今後，建物内や敷地内のホスト計算機を接続するためだけでなく，パーソナル・コンピュータやインテリジェント・ターミナルの普及にともない，これを接続する手段としてますます重要度を増すと考えられる。その中で，Acknowledging Ethernet は低価格で信頼性の高い通信方式として期待できる。また，Acknowledging Ethernet は，ローカル・コンピュータ・ネットワークにおける通信方式としてだけでなく，分散処理システムやマルチプロセッサ・システムなどのプロセッサ間通信方式としても有効であると考えている。

Acknowledging Ethernet は学内の Univac 1106, Burroughs B5700, Facom 230/38S, NOVA, PDP/11 などを接続するローカル・コンピュータ・ネットワークの通信方式として採用される予定である。

最後に，日頃から懇切な御指導を頂いている慶応大学教授 相磯秀夫博士，インタフェース・ハードウェアの設計に御協力頂いた相磯研究室の中川徹氏をはじめとする研究室員の皆様に心から感謝する。また，各種の御援助を頂いた富士通株式会社 井上頼昭部長，和田治社員に感謝する。

参考文献

[1] M. Tokoro and K. Tamaru, "Acknowledging Ethernet," Proc. COMPCON FALL, Sept. 1977.

[2] D. J. Farber, et. al., "The Distributed Computer System," Proc. COMPCON SPRING, March 1977.

[3] P. V. Mockapetris, et. al., "On the Design of Local Network Interfaces," Proc. IFIP, Aug. 1977.

[4] R. M. Metcalfe and D. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Comm. ACM, Vol. 19, No. 7, July 1976.