Osaka University Knowledge Archive

Title	機能階層形多重通信プロセッサによる分散ネットワークの高性能化に関する研究
Author(s)	寺田, 松昭
Citation	大阪大学, 1992, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3060227
rights	
Note	

Osaka University Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

Osaka University

機能階層形多重通信プロセッサによる
分散ネットワークの高性能化に関する研究

平成4年1月

寺田松昭

機能階層形多重通信プロセッサによる
分散ネットワークの高性能化に関する研究

平成４年１月

寺田　松昭

内容梗概

オフィスや工場において、ＬＡＮ（Local Area Network）などの高速パケット伝送路で複数の計算機を相互に接続した分散ネットワークが広く用いられている。近年、分散ネットワークは大規模化が進み、計算機間でやりとりされるデータ量が増大して、計算機での通信制御負荷とデータの転送遅延時間が共に増大してきた。このため、計算機とネットワークとのインタフェース部分におけるデータ転送スループットの向上、データ転送遅延時間の短縮、計算機の通信制御負荷低減が強く求められている。

これらの要求を満たすためには、伝送路であるＬＡＮ（産業用ではデータウェイと呼ぶこともある）の伝送速度を向上することと、計算機におけるデータ転送処理を高速化することとが必要である。ＬＡＮの伝送速度は、光ファイバ通信技術の適用などにより、１～２Ｍｂｐｓから１０～１００Ｍｂｐｓへ１桁～２桁高速化されてきている。一方、データ転送処理は、従来どおり計算機のソフトウェアによってパケット単位に行われており、高速化のあい路となっている。計算機とネットワークとの間のデータ転送処理を高速化するには、パケット単位の処理を計算機本体のソフトウェアで実行するのではなく、計算機と伝送路との間に高性能のＦＥＰ(Front End Processor)を置いて高速に実行する方式が考えられる。ところが、１０Ｍｂｐｓを超える高速パケット伝送路を対象としたＦＥＰの報告は少なく、その効果の評価は十分でない。

本論文では、ホスト計算機と伝送路との間に、データ転送のための通信プロトコル処理を専用に行う機能階層形多重通信プロセッサを置くことにより、従来は、ホスト計算機で実行していた通信プロトコル処理の大部分をＦＥＰで高速に実行させるような機能階層形多重通信プロセッサ方式を提案し、その性能評価を行っている。この方式により、通信プロトコル処理の高速化とホスト計算機の負荷削減を図ることができる事を示す。

具体的には、提案方式では、階層化された通信プロトコルのうち、第１層から第４層までの処理を、２階層に分け、機能階層毎に高速ビットスライスマイクロプロッサ（語長３２ビットのマイクロ命令を２００ｎｓで実行する）でパイプライン処理する専用の多重通信プロセッサを開発し、その効果を明らかにしている。トランスポートプロセ

ッサの構成方法としては、マルチプロセッサ・シングルコマンド処理方式とシングルプロセッサ・マルチコマンド処理方式の２つの方式を提案し、比較検討している。提案方式の評価を行うために、伝送路であるデータウェイの伝送速度が１０Ｍｂｐｓで、トランスポートプロセッサ２台、データリンクプロセッサ１台からなる機能階層形多重通信プロセッサを試作した。その結果、データ転送スループットが約３倍に、応答時間が約１／２に、計算機負荷がデータ長８バイトから１０ｋバイトの範囲で、４０％～８０％削減でき、提案している機能階層形多重通信プロセッサ方式が通信プロトコル処理高速化の要請に十分応えうることを明らかにしている。

今後の高速伝送路時代に対応するためには、プロトコル処理の高速化が益々重要になる。そこで、本論文では、プロトコル処理をさらに高速化するため、正常処理をハードウェアで実行するようにした高性能プロトコルプロセッサ方式を提案している。提案方式は、今後のマルチベンダによるネットワーク環境を考慮し、標準プロトコルであるＯＳＩ（Open Systems Interconnection）に基くプロトコル方式を実行できるようにしている。提案方式の効果は、実験的に評価し、従来方式にくらべ７～１２倍高速化できることを明らかにしている。

本論文の構成は、次のとおりである。まず、第２章において、第１層と第２層のプロトコルを実行するデータリンクプロセッサと第３層と第４層のプロトコルを実行するトランスポートプロセッサとから構成される機能階層形多重通信プロセッサ方式によるＦＥＰを提案する。また試作装置による性能向上効果について評価する。試作装置は、機能分散と負荷分散とにより性能を向上するために、トランスポートプロセッサを複数台のマイクロプロセッサで、データリンクプロセッサを１台のマイクロプロセッサで構成した点に特徴がある。第３章では、第２章で提案した方式を実用化した機能階層形多重通信プロセッサ方式について述べる。実用版では、ハード量を削減するために、トランスポートプロセッサを一台にすると共に、プロセッサの利用効率を向上するためにソフトウェアでコマンドを多重処理する方式にしている。第４章では、さらにプロトコルの高速処理を行うために、正常処理をハードウェアで実行する方式を提案し、その評価を行う。

関連発表論文

[1] M.Terada, J.Kashio, K.Yokota, Y.Hori, H.Fushimi: " A Network Operating System for High Speed Optical Fiber Loop Transmission System ", PP.641-646, Proc. of 5th International Conf. on Computer Communication (Atlanta), 1980

[2] 寺田　松昭、関　高明、樫尾　次郎、堀　雄太郎、「高速パケット伝送路用前置処理装置の一構成法」、情報処理学会論文誌、Vol.23、No.6、PP.707−715、昭和57年11月

[3] 寺田　松昭、高木　悟、安元　精一、中西　宏明、「機能階層形通信プロセッサによる計算機間メッセージ転送スループットの向上方式について」、情報処理学会「ローカルエリアネットワーク」シンポジウム、PP.163−170、昭和58年9月

[4] 福澤　淳二、寺田　松昭、高木　悟、溝河　貞生、末木　雅夫、「マルチ仮想回線動的チャネル割当方式による高性能LANアダプタの一方式」、情報処理学会「LAN／マルチメディアの応用と分散処理」シンポジウム、PP.69−76、昭和59年10月

[5] 寺田　松昭、高木　悟、樫尾　次郎、安元　精一、伏見　仁志、中西　宏明、「高速パケット伝送路用コマンド多重処理形前置処理装置」、情報処理学会論文誌、Vol.26、No.2、PP.219−227、昭和60年3月

[6] 福澤　淳二、寺田　松昭、溝河　貞生、「高性能MACブリッジ構成方式の提案と評価」、情報処理学会論文誌、Vol.31、No.8、PP.1260−1268、平成2年8月

[7] O.Takada, M.Tsukakoshi, M.Terada, M.Yamaga, " An FDDI Bridge for the Super Backbone LAN ", Proc. of The 15th Conf. on Local Computer Networks (Minneapolis), PP.399-404, 1990

[8] T.Hirata, S.Matsui, T.Yokoyama, M.Mizutani, M.Terada, " A High Speed Protocol Processor to Boost Gateway Performance " , Proc. of IEEE Global Telecommunications Conference (San Diego), PP.1426-1430, 1990

[９] 松井　進、平田　哲彦、横山　達也、水谷　美加、寺田　松昭、「専用ハードウェア化による通信プロトコル処理高速化の一方式」、情報処理学会論文誌、Ｖｏｌ．３２、Ｎｏ．２、ＰＰ．２７２－２７９、平成３年２月

[10] 平田　哲彦、横山　達也、水谷　美加、寺田　松昭、「高速ＬＡＮ用通信制御装置の一構成法」、電子情報通信学会情報ネットワーク研究会、ＩＮ９０－９８、ＰＰ．１３－１８、平成３年３月

[11] M.Terada, T.Yokoyama, T.Hirata, S.Matsui," A High Speed Protocol Processor to Execute OSI ", Proc. of IEEE INFOCOM'91 (San Francisco), PP.944-949, 1991

[12] 平田　哲彦、横山　達也、水谷　美加、寺田　松昭、三巻　達夫、「プロトコル高速処理装置により高性能化をはかったＬＡＮ用通信制御装置の一構成法」、情報処理学会論文誌、Ｖｏｌ．３３、Ｎｏ．２、平成４年２月（掲載決定）

[13] M.Tsukakoshi, O. Takada, T. Murakami, M.Terada, M. Yamaga, " Large-scale and High-speed Interconnection of Multiple FDDIs using ATM-based Backbone LAN" , Proc. of IEEE INFOCOM'92 (to appear)

機能階層形多重通信プロセッサによる分散ネットワークの高性能化に関する研究

目　次

第１章　序　論 …… 1
　１．１　本研究の目的 …… 1
　　１．１．１　分散ネットワークと課題 …… 1
　　１．１．２　従来の研究 …… 2
　　１．１．３　本研究の目的とアプローチ …… 3
　１．２　本論文の概要 …… 4
第２章　機能階層形多重通信プロセッサ方式による高性能化 …… 6
　２．１　緒言 …… 6
　２．２　データ転送処理高速化の課題 …… 7
　　２．２．１　計算機制御システムの特徴 …… 7
　　２．２．２　従来方式の問題点 …… 9
　　２．２．３　高速化の方策 …… 9
　２．３　機能階層形多重通信プロセッサ方式の提案 …… 14
　　２．３．１　要求条件と解決方式 …… 14
　　２．３．２　ＦＥＰ－ステーション間インタフェース …… 17
　　２．３．３　ＦＥＰ－計算機間インタフェース …… 18
　　２．３．４　ＦＥＰの構成例 …… 19
　２．４　実験的評価 …… 23
　　２．４．１　評価の対象と目的 …… 23
　　２．４．２　評価方法 …… 23
　　２．４．３　結果 …… 28
　２．５　結言 …… 35
第３章　機能階層形多重通信プロセッサにおける
　　コマンド多重処理方式 …… 36
　３．１　緒言 …… 36
　３．２　機能階層形多重通信プロセッサ方式実用化の課題 …… 36
　　３．２．１　ＦＥＰの位置付け …… 36
　　３．２．２　従来方式の問題点 …… 37
　　３．２．３　改良形ＦＥＰ（ＦＥＰⅡ） …… 37
　３．３　コマンド多重処理方式の開発 …… 40
　　３．３．１　要求条件とＦＥＰの方式提案 …… 40
　　３．３．２　ＦＥＰⅡのソフトウェア処理方式 …… 44

３．３．３　ＦＥＰⅡの性能評価 ・・・・・・・・・・・・・・・・・・・・・・・４５
３．３．４　ＦＥＰⅡの方式評価 ・・・・・・・・・・・・・・・・・・・・・・・４７
３．４　適用評価 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・５２
３．４．１　評価の対象と目的 ・・・・・・・・・・・・・・・・・・・・・・・・５２
３．４．２　評価方法 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・５２
３．４．３　結果 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・５８
３．５　結言 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・５８
第４章　プロトコル処理ハード化による高速化の提案 ・・・・・・・・６２
４．１　緒言 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・６２
４．２　プロトコル処理高速化の課題 ・・・・・・・・・・・・・・・・・・・・・６２
４．２．１　ＯＳＩプロトコル ・・・・・・・・・・・・・・・・・・・・・・・・６２
４．２．２　プロトコル処理高速化の課題 ・・・・・・・・・・・・・・・・・６２
４．２．３　高速化の方策 ・・・・・・・・・・・・・・・・・・・・・・・・・・・６４
４．３　プロトコル処理ハード化方式の提案 ・・・・・・・・・・・・・・・６５
４．３．１　要求条件 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・６５
４．３．２　構成方式の検討 ・・・・・・・・・・・・・・・・・・・・・・・・・６６
４．３．３　プロトタイプの開発 ・・・・・・・・・・・・・・・・・・・・・・・６８
４．３．４　ＬＡＮアダプタへの適用 ・・・・・・・・・・・・・・・・・・・６８
４．４　実験的評価 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・７４
４．４．１　評価の目的 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・７４
４．４．２　評価の方法 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・７４
４．４．３　結果 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・７４
４．４．４　ＬＡＮアダプタへの適用効果 ・・・・・・・・・・・・・・・・７４
４．５　結言 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・７７

第５章　結論 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・７８

謝辞

参考文献

第１章　序論

１．１　本研究の目的

１．１．１　分散ネットワークと課題

計算機により、鉄鋼プラントや化学プラントを制御するシステムは、計算機制御システムと呼ばれる。近年、計算機制御システムは、大規模化、トータルシステム化、分散化が進んでおり、ホスト計算機と多数の端末計算機を高速データウェイ[1]などの高速パケット通信ネットワークによって相互に接続した分散ネットワーク構成が広く採用されている[2][3]。計算機制御システムの大規模化に伴い、分散ネットワークには、①計算機間のデータ転送スループットの向上、②応答時間の短縮、③計算機負荷の削減など、性能の向上が強く求められている[4]。

また１９７０年代に開発が始まったＬＡＮ(Local Area Network)[5]は、その後急速に進展し、近年、ＬＡＮを高速パケット通信ネットワークとして、サーバとなる計算機とクライエントとなる計算機／ワークステーション／パソコンを相互に接続し、クライエントからサーバの資源の共有を図るサーバ／クライエントシステムとよばれる新しい分散ネットワーク形態が急速に普及してきた。このような分散ネットワークにおいても、規模の増大や扱うデータ量の増大に伴い、①ファイル転送スループットの向上や②応答時間の短縮が強く求められている。

分散ネットワークの性能を向上するためには、①計算機間を接続するための伝送路であるＬＡＮやデータウェイの伝送速度を向上することと、②計算機におけるデータ転送処理を高速化することとが必要である。

データウェイの伝送速度は、光ファイバ通信技術の適用により、１～２Ｍｂｐｓから１０～３２Ｍｂｐｓへと約１桁高速化された[4][6]。一方、データ転送処理は、相変わらず計算機のソフトウェアによって、データウェイの転送単位であるパケット毎に行われている。このため、①計算機でのデータ転送処理時間が、データウェイのパケット転送時間に比べて大きく、伝送速度の向上に見合う計算機間での実効転送ス

ループットが得られない、②計算機のデータ転送処理に係る負荷が大きい、という問題があった。

１．１．２　従来の研究

上記課題に対して、従来下記の研究が行われてきた。

計算機におけるパケット単位の処理を高速化する方式として、外付けのハードウェアで実現する方法がある。ハードウェアで行う方式には、専用論理回路を組む方式[3]と専用通信プロセッサを計算機と伝送路との間に設ける方式[10][11][12][13]とがある。通信プロセッサの研究開発は、１９７０年代半ばころから活発となり、主に多種の通信手順を持つ多数の回線を制御する目的で研究開発が行われた。これらは、大型計算機のフロントエンドプロセッサとして位置付けられるものであり、回線速度も比較的低速（一般には、４８ｋｂｐｓ以下）で、サポートプロトコル範囲も２層以下（物理層とデータリンク層）に限られていた。

１９８０年代になると、ＬＡＮが出現し、パソコンやワークステーション向けに、１０Ｍｂｐｓクラスの伝送速度を対象としたＬＡＮアダプタが開発されるようになった[7][8]。これは、一本の高速回線を対象としている点で従来の通信プロセッサとは異なるものであった。しかし、その機能範囲は２層以下（物理層とデータリンク層）であり、１６ビットの汎用マイクロプロセッサでデータ転送処理を行うので、性能的にも制限されていた。研究開発の中心は、データリンク層の下位半分に位置付けられ伝送路の共用制御を司るＬＡＮの媒体アクセス方式にあった。

制御用ネットワークのように、ミリ秒オーダの応答時間が要求される分散システムにおいて、データウェイのような１０～３２Mbpsの高速パケット通信ネットワークを対象とし、１層から４層までのプロトコルを高速に実行する通信プロセッサの報告は少なく、その効果の評価は十分行われていない。

近年、ＦＤＤＩ（Fiber Distributed Data Interface）の標準化が進み計算機に直結できるＬＡＮの速度が１００Ｍｂｐｓに向上した[9]。しかし、ＬＡＮの速度向上にもかかわらず、プロトコル処理オーバヘッドなどによって、エンドツーエンドでの実効スループットが向上しないということが大きな問題になってきた。この原因は、プロトコ

ルの処理を計算機のプログラムやＬＡＮアダプタの中のマイクロプロセッサでパケット毎に処理していることによる。このため、エンドツーエンドの実効スループットを向上するための研究が活発になってきた。ＬＡＮ環境でのトランスポートサービスのスループットが詳細に測定され報告されている[36][45]。高速ネットワークのための新しい軽量プロトコルが提案されている[37]。そのような軽量プロトコルアーキテクチャに基づく第３層と第４層のプロトコルを実行するＶＬＳＩの方式が提案されている[38]。このように、新しいプロトコルを導入して、実効スループットを向上するアプローチがある。しかしながら、ＩＳＯで標準化が進んでいるプロトコル体系であるＯＳＩ（Open Systems Interconnection:開放型システム間相互接続）［40］は、マルチベンダによるネットワーク環境の増加に伴い、着実に普及することが予想でき、ＯＳＩのプロトコル処理を高速化することも益々重要になると考えられる。このアプローチの例としては、パケット交換網のための標準プロトコルであるＸ．２５の第２層と第３層を実行するＶＬＳＩの試作をした例が報告されている[39]。またＯＳＩのような多層構造プロトコルの性能評価をした例が報告されている［46］［47］［48］。

１．１．３　本研究の目的とアプローチ

本研究の目的は、計算機間での実効データ転送スループットを向上するための分散ネットワークの高性能化方式について論ずることにある。

計算機におけるデータ転送処理を高速化する方法としては、パケット単位の処理を計算機のファームウェアで行う方法と専用の外付けハードウェアで行う方法とが考えられる。両法を計算機負荷の観点から比較すると、後者のほうが性能の改善度合いは高い。そこで、本研究では、ホスト計算機と伝送路との間に、データ転送処理を専門に実行する機能階層形多重通信プロセッサを置くことにより、従来は、ホスト計算機で実行していた通信プロトコル処理の大部分を通信プロセッサで高速に実行させるようにした。この方式により、通信プロトコル処理の高速化とホスト計算機の負荷削減を図ることができる事を示す。

次に本研究では、ＬＡＮアダプタに適用することを意図して、第４層以下のプロトコルの高速処理方式を検討する。高速ネットワーク対応の新プロトコルを考案して高速化を図る方法もあるが、本研究ではマルチベンダ環境でのネットワーク化を重視し、プロトコルとしてＯ

ＳＩを対象にして高速処理方式を検討する。ＯＳＩの高速処理には、プロトコル処理アルゴリズムに工夫をするアプローチもあるが、それでは効果が少ないと考え、本研究ではプロトコル処理のハード化を検討する。

１．２　本論文の概要

本論文は、機能階層形多重通信プロセッサによる分散ネットワークの高性能化について述べている。本論文の構成は、次のとおりである。

第２章では、通信プロセッサを第１層と第２層のプロトコルを実行するデータリンクプロセッサと第３層と第４層のプロトコルを実行するトランスポートプロセッサとから構成する機能階層形多重通信プロセッサの提案と性能向上度の評価を行う[16][20][21][22]。提案した方式は、ホスト計算機と伝送路との間に、データ転送のための通信プロトコルを専用に処理する機能階層形多重通信プロセッサを置くことにより、従来は、ホスト計算機で実行していた通信プロトコル処理の大部分を通信プロセッサで高速に実行させるようにした点に特徴がある。この方式により、通信プロトコル処理の高速化とホスト計算機の負荷削減を図ることができる事を示す。

具体的には、階層化された通信プロトコルのうち、第１層から第４層までの処理を、２階層に分け、機能階層毎に高速ビットスライスマイクロプロッサ（語長３２ビットのマイクロ命令を２００ｎｓで実行する）でパイプライン処理する専用の多重通信プロセッサを開発してその効果を明らかにしている。トランスポートプロセッサの構成方法としては、マルチプロセッサ・シングルコマンド処理方式（前置処理装置：ＦＥＰ－Ｉ）を提案し検討する。効果の評価は、パケット通信ネットワークであるデータウェイの伝送速度が１０Ｍｂｐｓで、トランスポートプロセッサを２台のマイクロプロセッサで、データリンクプロセッサを１台のマイクロプロセッサで構成した機能階層形多重通信プロセッサを試作して実験的に行い、トランスポートプロセッサを用いない場合に比べてデータ転送スループットが３倍に、応答時間が１／２に、計算機負荷削減がデータ長８バイトから８ｋバイトの範囲で、４０％～８０％になることを明らかにする。この結果より、機能階層形多重通信プロセッサ方式が通信プロトコル処理高速化の要請に応えうることを結論づける。

第３章では、第２章で提案した方式（ＦＥＰ－Ｉ）のハード量を削減するために、トランスポートプロセッサを一台にして、ソフトウェアでコマンドを多重処理する方式について述べる[23][24][25][26]。

具体的には、コマンド多重処理形トランスポートプロセッサ（前置処理装置：ＦＥＰ－Ⅱ）を提案する。提案するＦＥＰ－Ⅱは、１台の高速マイクロコンピュータで多数のコマンドを多重に処理するので、少ないハードウェアで、高い性能を得ることが期待できる。提案方式の効果は、３２ビットスーパミニコンと高速データウェイとの接続に適用し、シミュレーションにより評価する。評価の結果、提案したＦＥＰ構成方式は、以下の効果があることを明らかにする。(1) ハード量は、基板２枚（第２章で提案したマルチプロセッサ方式の２／３）で済む、(2) 性能は、データウェイの伝送速度との比に換算して約０．１４（第２章で提案したマルチプロセッサ方式のプロセッサ２台のときの性能に相当）である、(3) ＦＥＰを適用した３２ビットスーパミニコンは、従来の１６ビット機種に比べ、平均２０００バイトのメッセージ転送性能において、スループットで約３倍、応答時間で約１／２の性能向上効果がある

第４章では、プロトコル処理ハード化による高速化方式の提案について述べる[27][28][29][30][31][32]。データウェイの伝送速度の向上は、めざましくやがて１００Ｍｂｐｓの時代も到来すると考えられる。本論文の主眼は、高性能通信プロセッサによる高性能化であるが、１００Ｍｂｐｓ時代には、単にプロセッサの高性能化だけでなく、プロトコル処理のハード化を考える必要がある。第４章では、プロトコル処理ハード化の課題を整理し、ＯＳＩのレイヤ２、レイヤ３、レイヤ４のプロトコルを対象に、プロトコル処理のハード化による高速化方式を提案する。

第５章では、本論文で得られた結果のまとめと今後の課題について述べる。

第２章　機能階層形多重通信プロセッサ方式による高性能化

２．１　緒言

近年、計算機制御システムの大規模化が進み、データウェイで接続された計算機間のデータ転送性能の向上が強く求められている[4]。性能を向上するためには、計算機におけるデータ転送処理を高速化することが必要である。従来、データ転送処理は計算機のソフトウェアによって、データウェイの転送単位であるパケット毎に行われている。このため、①計算機でのデータ転送処理時間が、データウェイのパケット転送時間に比べて大きく、伝送速度の向上に見合う計算機間での実効転送スループットが得られない、②計算機のデータ転送処理に係る負荷が大きい、という問題があった。

計算機におけるパケット単位の処理負荷をオフロードする方式として通信プロセッサの研究開発が、１９７０年代半ばころから、主に多種の通信手順を持つ多数の回線を制御する目的で行われた[10][11]。これらは、大型計算機のフロントエンドプロセッサとして位置付けられるものであり、回線速度も比較的低速（一般には、４８ｋｂｐｓ以下）で、サポートプロトコル範囲も２層以下に限られていた。１９８０年代になると、ＬＡＮ[5]が出現し、パソコンやワークステーション向けに、１０Ｍｂｐｓクラスの伝送速度を対象としたＬＡＮアダプタが開発されるようになった[7][8]。ＬＡＮアダプタの機能範囲は２層以下であり、１６ビットの汎用マイクロプロセッサでデータ転送処理を行うので、性能に限界があった。

計算機制御システムにおける分散ネットワーク[14][15]のように、ミリ秒オーダの応答時間が要求される環境において、１０～３２Mbpsの高速パケット通信ネットワークを対象とし、１層から４層までのプロトコルを高速に実行する通信プロセッサの報告は少なく、その効果の評価は十分行われていない。

本章では、計算機と伝送路との間に、トランスポートプロセッサ（以下前置処理装置：ＦＥＰと呼ぶ）とデータリンクプロセッサ（データウェイの場合は、ステーションに相当）とから構成する機能階層形多重通信プロセッサを置くことによってデータ転送性能を向上する方式の提案と性能向上度の実験的評価とを行っている。

(1) 高いデータ転送スループットが、ミリ秒オーダの厳しい応答時間の制約下で得られるように、ＦＥＰを、複数の高速マイクロコンピュータ（Ｉ

ＯＰ）と１台のマルチプレクサとで構成している。

(2) ＩＯＰのメモリを少なくするために、ＩＯＰには、１パケット分のバッファだけをもたせ、計算機とＩＯＰとの間の転送単位は、パケットにしている。

(3) 計算機の負荷を低減するため、メッセージのセグメンティング／リアセンブリは、ＩＯＰが計算機のメモリに置かれたメッセージを直接アクセスすることによって行う。

(4) 上記提案方式の性能向上度を、ＩＯＰ２台から成る試作装置によって実験的に評価し、ＦＥＰを挿入することによって、ステーションのみの場合に比し、データ転送スループットが３倍に、応答時間が１／２に、計算機負荷削減がデータ長８バイトから１０ｋバイトの範囲で４０％～８０％になることを明らかにしている。

２．２　データ転送処理高速化の課題

計算機制御システムにおける計算機間データ転送の性能を定義するとともに高速化に当たっての問題点と解決法を論じる。

２．２．１　計算機制御システムの特徴

計算機制御システムの中で、図２．１に示すようなデータウェイに代表される高速パケット通信ネットワークによって、複数の計算機が接続されている例[17][18][19]をとりあげ、以下の議論を進める。データウェイは、光ファイバ／同軸ケーブルとステーションとで構成されている[4]。ステーション相互間では、パケットと呼ばれる最大５００バイト程度の可変長データブロックが、１０～３２Mbpsでビットシリアルに伝送される。計算機間のデータ転送は、主にホスト計算機と１０～２０台の端末計算機との間で行われる。近年、計算機制御システムは、大規模化が進み、下記性能の改善が強く求められている。

(1) データ転送スループット
(2) 応答時間
(3) 計算機負荷

これらの述語は、次のように定義する。

(1) 単位時間に転送できるデータ量
(2) メッセージの送信要求がユーザ・プログラムから発行されてから、送信を完了し、ユーザ・プログラムに制御が戻るまでの時間
(3) データ転送処理プログラムが動作している時間と計算機の動作時間との比

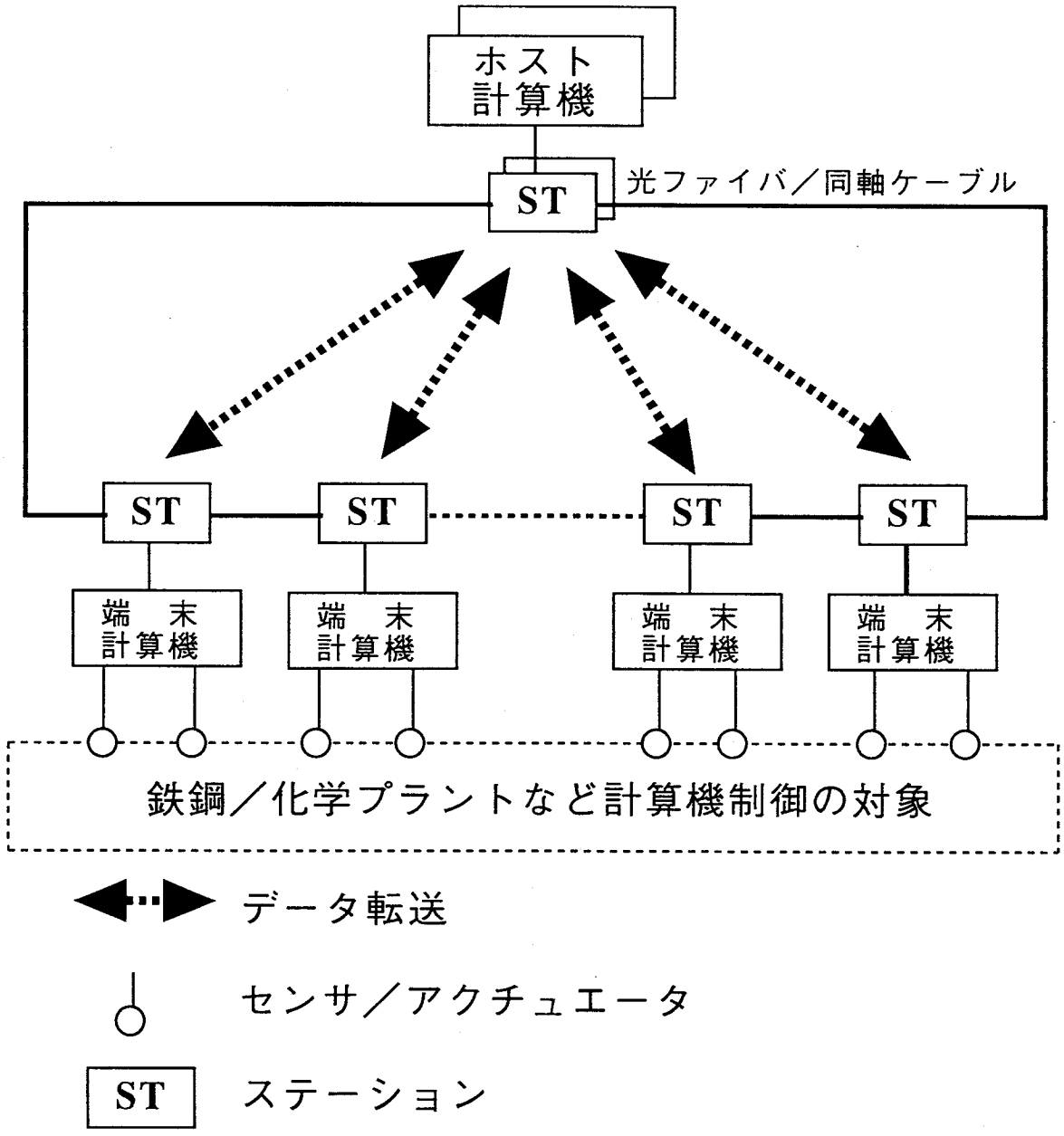


図２．１　高速パケット通信ネットワークを用いた
計算機制御システム

２．２．２　従来方式の問題点

計算機間で転送される情報内容はメッセージといわれている。メッセージの長さは不定のため、計算機間のデータ転送では、メッセージをセグメントに分割し、各セグメントをパケットとして伝送路に送出する。

従来は、計算機とデータウェイのステーションとが、図２．２のように直接接続され、データ転送処理は、計算機のソフトウェアによって、パケット単位に行われていた（図２．３）。すなわち、データ転送処理は、次の四つの制御により実施されていた。

(A) ユーザプログラムとのインタフェース制御
(B) メッセージのセグメンティング／リアセンブリ
(C) 伝送路に時分割多重化されたパケットを複数の論理的な伝送路（論理チャネル）に分ける制御
(D) ステーションとのインタフェース制御

このように、多くの処理をすべて計算機のソフトウェアで行っているため、計算機間のデータ転送性能は、伝送路の伝送速度が向上したにもかかわらずほとんど向上していない。

２．２．３　高速化の方策

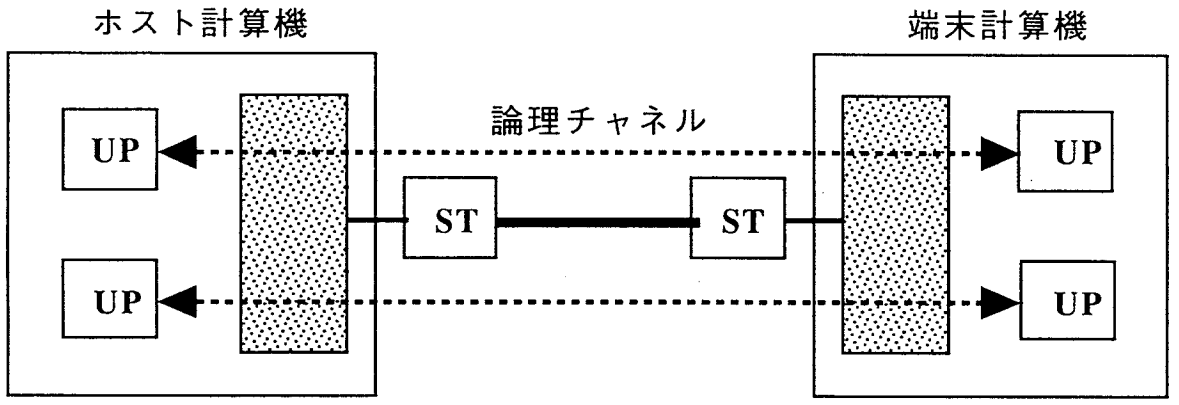
データ転送処理を高速化する方法としては、下記の２つが考えられる。

(1) ファームウェア化
(2) ハードウェア化

ファームウェア化は、計算機のデータ転送処理すべて（２．２．２節のＡ～Ｄ）にわたって高速化を行うことができるが、計算機負荷の低減効果が少ない。

ハードウェア化は、パケット単位の処理（２．２．２節Ｂ～Ｄ）を、外づけのハードウェアで行う方法で、専用の論理回路を組む方式と前置処理装置（ＦＥＰ）を設ける方式とがある。専用回路は、高速性は、大であるが、論理チャネルの数が多くとれないなど融通性に問題がある。これらに比し、ＦＥＰを設ける方式は、計算機負荷低減効果が大きく、融通性に富む。

本論文では、ＦＥＰ方式を取り上げる。ＦＥＰの導入（図２．４）によって、計算機は、図２．５に示すようにパケット単位の処理から解放され、メッセージ単位の処理のみ行えばよく、計算機負荷の大幅な低減が期待できる。さらに、

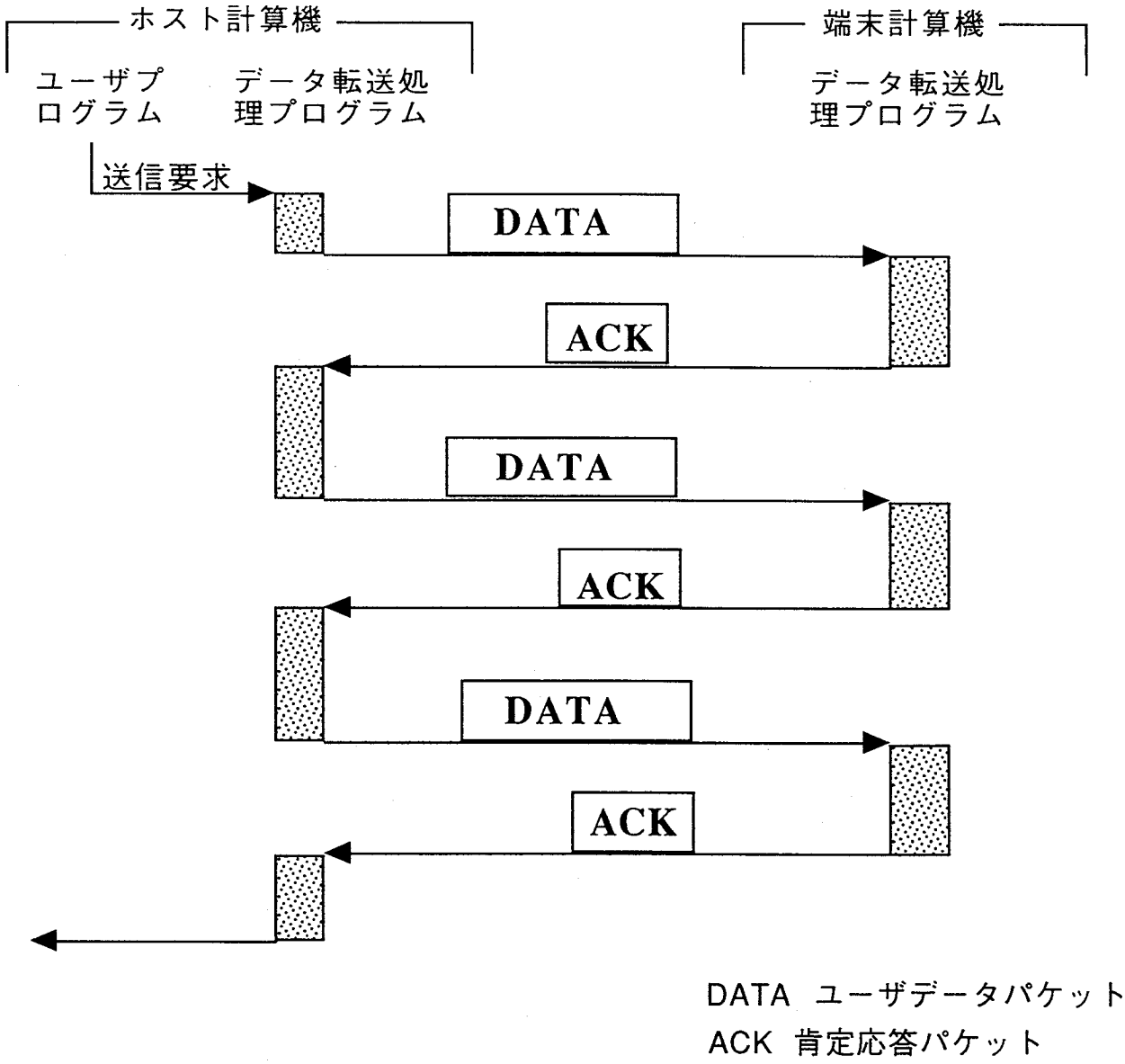


高速パケット伝送路　　データ転送処理プログラム

UP　ユーザプログラム

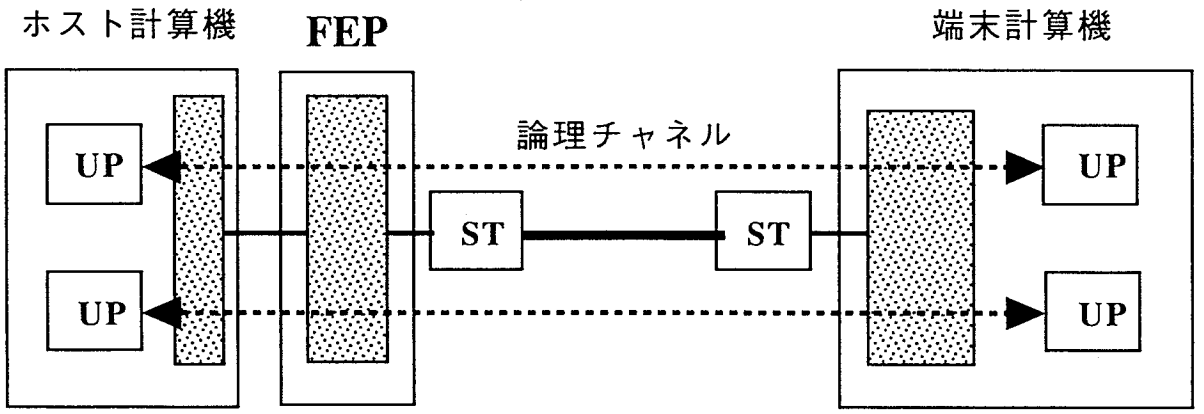
ST　ステーション

図２．２　高速パケット伝送路と論理チャネル



論理チャネルレベルのデータ転送モデルを示している

図２．３　従来システムのデータ転送モデル



データ転送処理プログラム

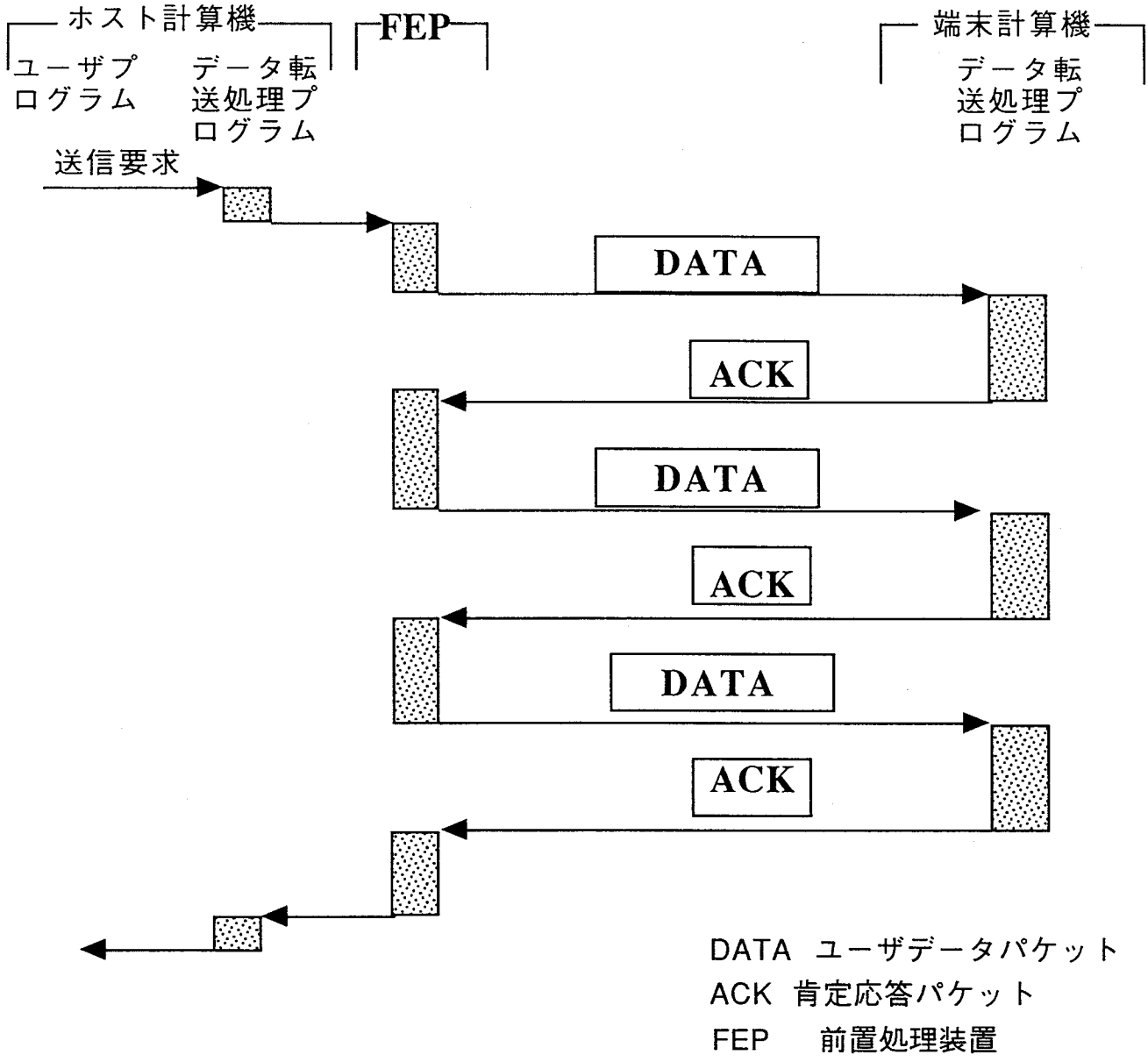
高速パケット伝送路

FEP　前置処理装置

UP　ユーザプログラム

ST　ステーション

図２．４　前置処理装置（FEP）を用いた計算機間接続



論理チャネルレベルのデータ転送モデルを示している

図２．５　FEP適用時のデータ転送モデル

ＦＥＰを高性能なプロセッサで構成すれば、ＦＥＰで行うパケット毎の処理を、計算機で行っていたよりも高速に行えるので、データ転送スループット、応答時間の改善も期待できる。

２．３　機能階層形多重通信プロセッサ方式の提案

高速パケット通信ネットワークに適した前置処理装置（ＦＥＰ）が具備すべき条件を明らかにし、その解決方法について論ずる。

２．３．１　要求条件と解決方式

(1) 要求条件

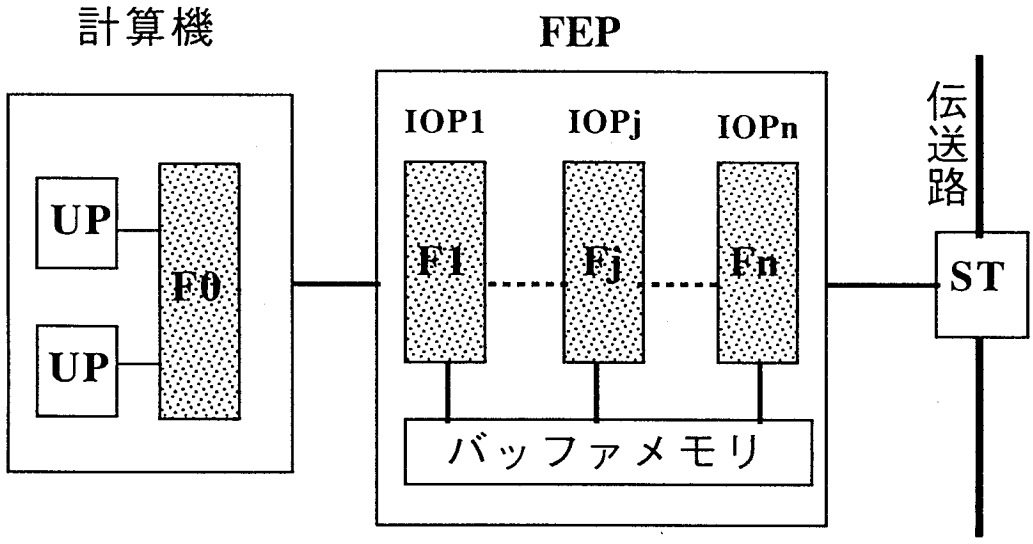
高速パケット通信ネットワークを用いた計算機制御システムにおいて使用されるＦＥＰには、下記の条件が必要である。

(a) １０～３２Ｍｂｐｓの伝送速度に適応できる高いデータ転送スループット
(b) 応答時間が数ミリ秒という高速応答性
(c) パケットが多重化された高速伝送路とのインタフェース機能
(d) メッセージのセグメンティング／リアセンブリおよび伝送路のパケットを論理チャネルに分ける処理をＦＥＰが行えるようにする計算機インタフェース

(2) データ転送スループットの向上

データ転送スループットは、平均パケット長と、パケット処理能力の積で定義され、パケット処理能力は、パケット当たりの処理量（プログラムのダイナミックステップ数）が一定ならば、ＦＥＰの命令実行能力で定まる。したがって、１０～３２Ｍｂｐｓの高速パケット伝送を可能にするには、ＦＥＰに十分な処理能力を持たせる必要がある。そこで、提案方式では、ＦＥＰを複数のプロセッサ（以下ＩＯＰ:Input/Output Processorと称す）で構成することにした。複数ＩＯＰによるＦＥＰの構成方式としては、機能分散方式（図２．６）と負荷分散方式（図２．７）とが考えられる。

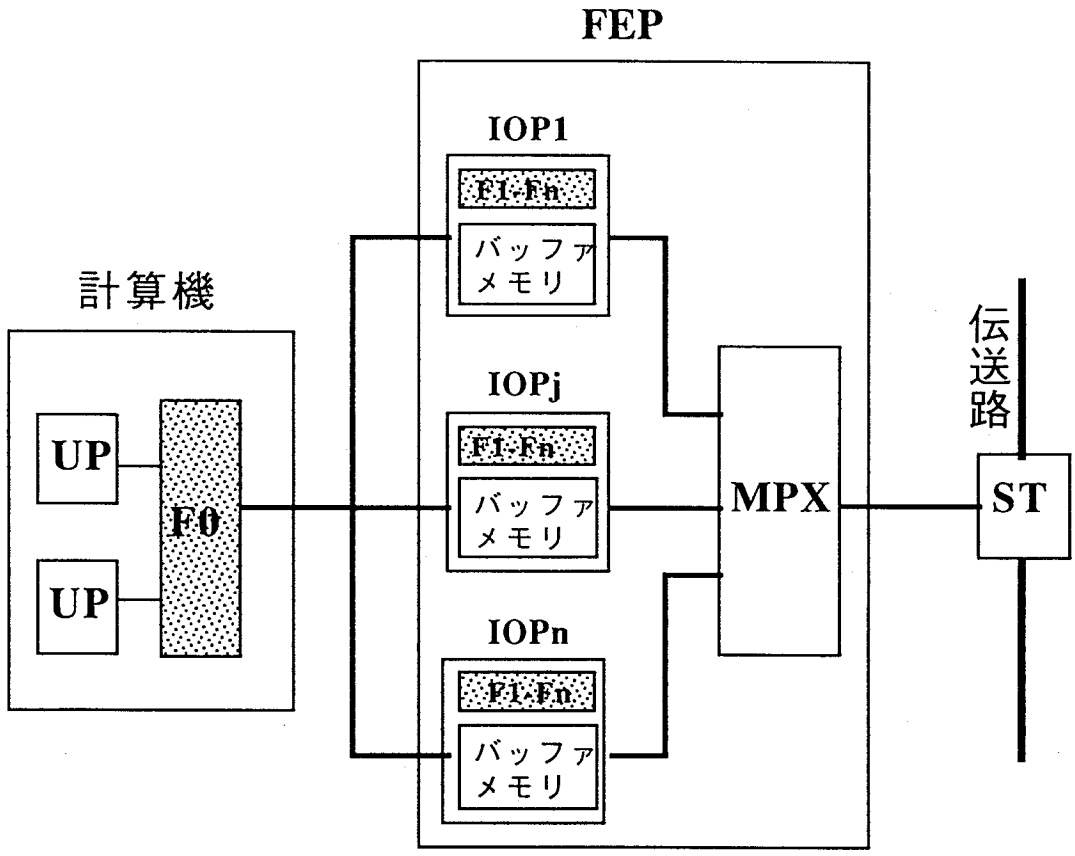
(a) 機能分散方式
各ＩＯＰは、固有の機能を分担する。計算機からのデータは、まずＩＯＰ１に渡され、以降ＩＯＰ２，ＩＯＰ３，・・・と渡され、最後にステーションに渡される。



データ転送処理プログラム

UP　ユーザプログラム
ST　ステーション
FEP　前置処理装置
IOP　入出力プロセッサ

図２．６　機能分散方式によるFEPの構成例



UP　ユーザプログラム
ST　ステーション
FEP　前置処理装置
IOP　入出力プロセッサ
MPX　　STとのインタフェース制御
　とパケットの分配／統合を実行

データ転送処理プログラム

図２．７　負荷分散方式によるFEPの構成例

(b) 負荷分散方式

各ＩＯＰは、同一の機能を備えている。計算機からのデータは、各ＩＯＰに分配され、各ＩＯＰでは、これらのデータを一時、ＩＯＰのバッファメモリに蓄えて、所定の処理を行った後、ステーションへ渡す。このとき、各ＩＯＰを対等にするため、マルチプレクサ：ＭＰＸと呼ぶ装置を介してステーションに接続する。

両方式を比較すると、応答時間特性は、Ｍ／Ｍ／Ｓ待ち行列モデルを仮定する限り、負荷分散方式が優れていることがよく知られている。一方、プログラムと制御テーブルを格納するためのメモリは、機能分散方式が少なくて済む。計算機制御システムでは、応答時間に対する要求が厳しいので、負荷分散方式を採用し、ＩＯＰのメモリ容量を削減する工夫を行うことにした。

(3) 応答時間の短縮

応答時間は、ＦＥＰの処理時間とデータをＩＯＰのバッファメモリに転送する時間とを短縮することによって改善される。一つのパケットに関する処理は、並列に行えないので、ＦＥＰの処理時間を短縮するには、ＩＯＰ単体の命令実行速度を大きくする（すなわち、高速プロセッサを使う）必要がある。バッファメモリへの転送時間を少なくするには、転送回数を減らすこと、バッファメモリの読みだし／書き込みを高速にする（すなわち高速メモリを使う）ことが必要になる。

高速プロセッサと高速メモリは、実装密度が低く、高価なので、少なくともバッファメモリは、極力少なくて済む方式が望ましい。応答時間を短縮し、メモリを減らすため、次の方針を立てた。

(a) ＦＥＰ内でのデータ蓄積は、ＩＯＰのみにとどめ、ＭＰＸでは行わない。
(b) ＩＯＰのバッファメモリは、１パケット分のみを準備する。
(c) 計算機のメモリを各ＩＯＰの共有メモリとして用いる。

２．３．２　ＦＥＰ－ステーション間インターフェース

高速パケット通信ネットワークの特徴は、同一伝送路上を、異なる論理チャネルのパケットが、時分割多重で高速に伝送されていることである。これら高速伝送されているパケットを複数のＩＯＰで高速に処理し、データ転送スループット、応答時間を改善するには、ＦＥＰとステーションおよびＦＥＰと計算機のインターフェースの実現方法が重要な課題である。

ＦＥＰとステーションとは、マルチプレクサ（ＭＰＸ）によって接続され、

受信パケットは、ＭＰＸによってＩＯＰに分配される。ＭＰＸには、ＩＯＰ複数台分以上の高いスループットとＩＯＰの負荷に片寄りを生じさせないパケット分配方式とが求められる。

ＭＰＸは、上記目的を満たすため、パケットヘッダの参照、論理チャネルへの振り分けをすることなく、パケットをＩＯＰに分配する。すなわち、ＭＰＸは、各ＩＯＰのバッファメモリにパケットが蓄積されている（使用中）か否かを保持しておき、バッファメモリが使用中でないＩＯＰにパケットを渡す。バッファメモリの使用状況は、ＩＯＰからＭＰＸに対して、専用の信号線によって知らされる。

この方式によれば、ＭＰＸでのパケットの蓄積は不要であるばかりでなく、分配アルゴリズムが単純なので、分配処理をハードウェア化しやすく、高速化できる。

２．３．３　ＦＥＰ－計算機間インターフェース

データ転送性能を向上するためには、ＦＥＰと計算機とのインターフェースを単純にするとともに、計算機の処理をできるだけＦＦＰに移すことが重要となる。とくに、高速パケット伝送路用ＦＥＰでは、メッセージのセグメンティング／リアセンブリとパケットを論理チャネルに振り分ける処理が、データ転送処理プログラムの大きな部分を占めるので、これらの処理をＦＥＰでどのように行うかが課題となる。とりわけ、複数のマイクロコンピュータでの処理方法が中心課題となる。

計算機とＦＥＰとのインターフェースの動作は、大きく送信と受信とに分けられ、各々は、計算機からＩＯＰへの起動とＩＯＰから計算機への終了報告とからなる。

送信時には、計算機のデータ転送処理プログラムが、使用中でないＩＯＰを１台選択してメッセージの送信を要求し、このＩＯＰを使用中の状態にする（起動）。起動されたＩＯＰは、計算機メモリのメッセージを読みだして、最初のセグメントを送信し、送信を終わると、計算機にこのＩＯＰが空状態になったことを知らせる。メッセージの残りの部分は、セグメント毎に受信側から返送されてくる応答を受信ＩＯＰが、計算機メモリのメッセージを読みだして送る。最後のセグメントを送信したＩＯＰが、計算機に対して送信完了の割り込みを入れる（終了報告）。

受信時には、受信要求を計算機のテーブル（該当論理チャネル制御テーブル）にセットしておく（ＩＯＰは起動しない）。やがて、伝送路からこの論理チャネルあてのパケットがＭＰＸに届く、ＭＰＸは、１台のＩＯＰにパケットを分

配する。パケットを受け取ったＩＯＰは、パケット・ヘッダに定義されている論理チャネル制御テーブル（計算機のメモリにある）より、受信パケットのストア・アドレスを取だし、そこへパケットをストアする（パケットのリアセンブリ）。最終セグメントのパケットをストアしたＩＯＰが計算機に対して受信完了割り込みを入れる（終了報告）。

このようなＩＯＰの連携動作を可能にするためには、論理チャネル制御テーブルを計算機のメモリを介して、あるＩＯＰから別のＩＯＰへリレーできるようにする必要がある。すなわち、マスタ計算機のメモリに置き、セグメンティング／リアセンブリを行うＩＯＰがそれを自分のバッファメモリに一時的にコピーして使い、必要な部分を変更して再び計算機のメモリに返すのである。これを制御テーブルリレー方式と呼ぶ。以下では、この方式を図２．８にもとづき、詳しく説明する。

(a)ＩＯＰ１は、パケットｉをメッセージ・バッファにストアし終わるとＡＤＤＲ（ｉ）をＡＤＤＲ（ｊ）にして、制御テーブル（ｘ）を計算機の制御テーブル（ｙ）に返却する。
(b) 次に、パケットｊをＩＯＰ２が受けると、テーブル（ｙ）をＩＯＰ２の制御テーブル（ｚ）にコピーして、パケットｊのストア・アドレスであるＡＤＤＲ（ｊ）を知る。
(c) パケットｊをメッセージ・バッファのＡＤＤＲ（ｊ）にストアするとＡＤＤＲ（ｊ）をＡＤＤＲ（ｋ）にしてテーブル（ｚ）をテーブル（ｙ）に返却する。

この方式によれば、メッセージおよび主要な制御テーブルを計算機のメモリに配置し、ＩＯＰには、パケットバッファとわずかな制御テーブルを置くだけで済む。しかも、セグメンティング／リアセンブリをＩＯＰが行える。

２．３．４　ＦＥＰの構成例

提案したＦＥＰのシステム構成をミニコンへの適用を例にし、従来の構成（図２．９）と対比させて、図２．１０に示す。ＦＥＰを、複数のマイクロコンピュータ（ＩＯＰ）と１台のマルチプレクサ（ＭＰＸ）とから構成する。各ＩＯＰを、マイクロプロセッサ、プログラムメモリ、バッファメモリによって構成し、計算機の入出力バスに接続する。各ＩＯＰを集約して、ステーションに接続するためＭＰＸを、ＩＯＰとステーションとの間に入れる。ＭＰＸには、パケット蓄積バッファを設けない。

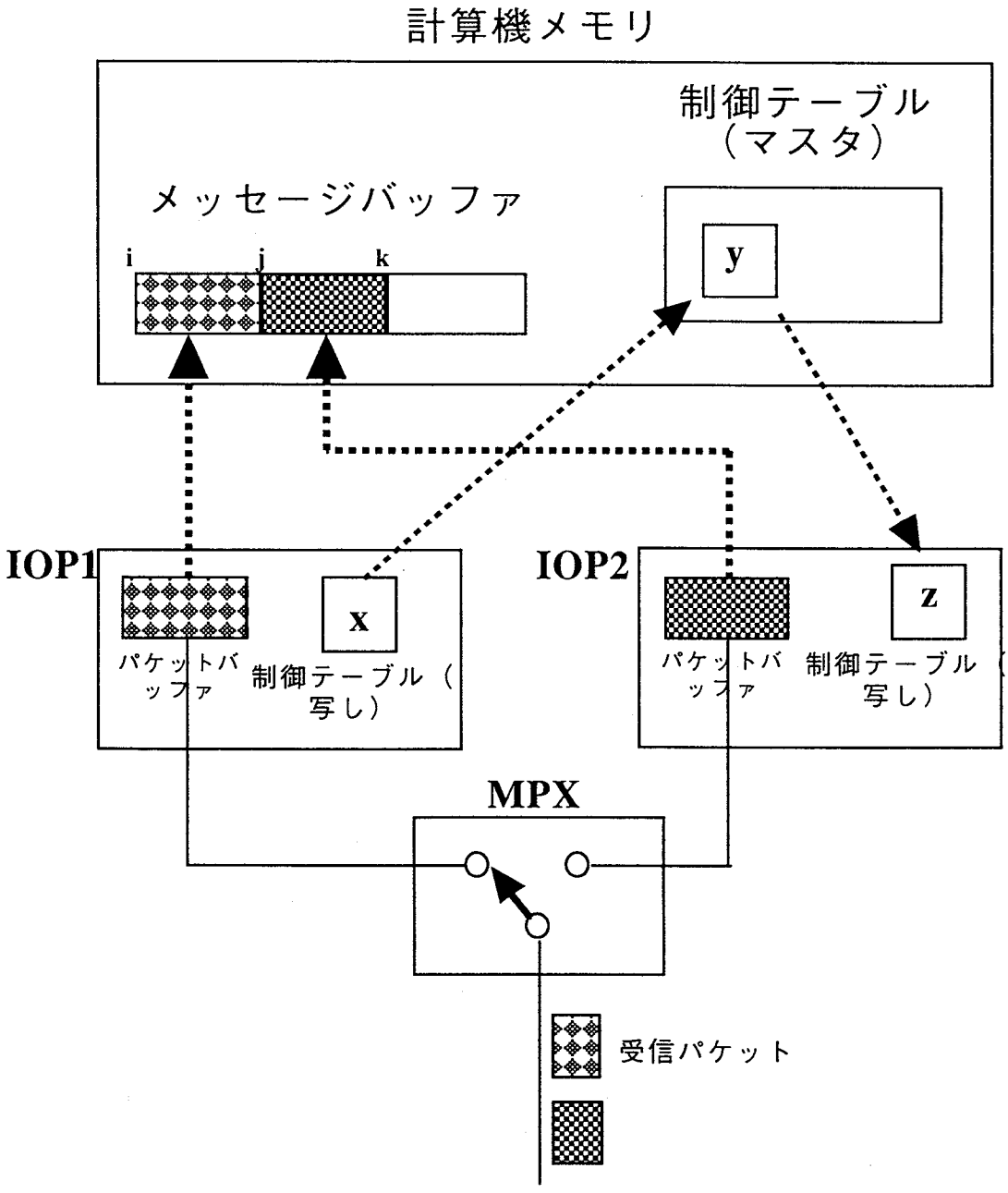


図２．８　制御テーブルリレー方式

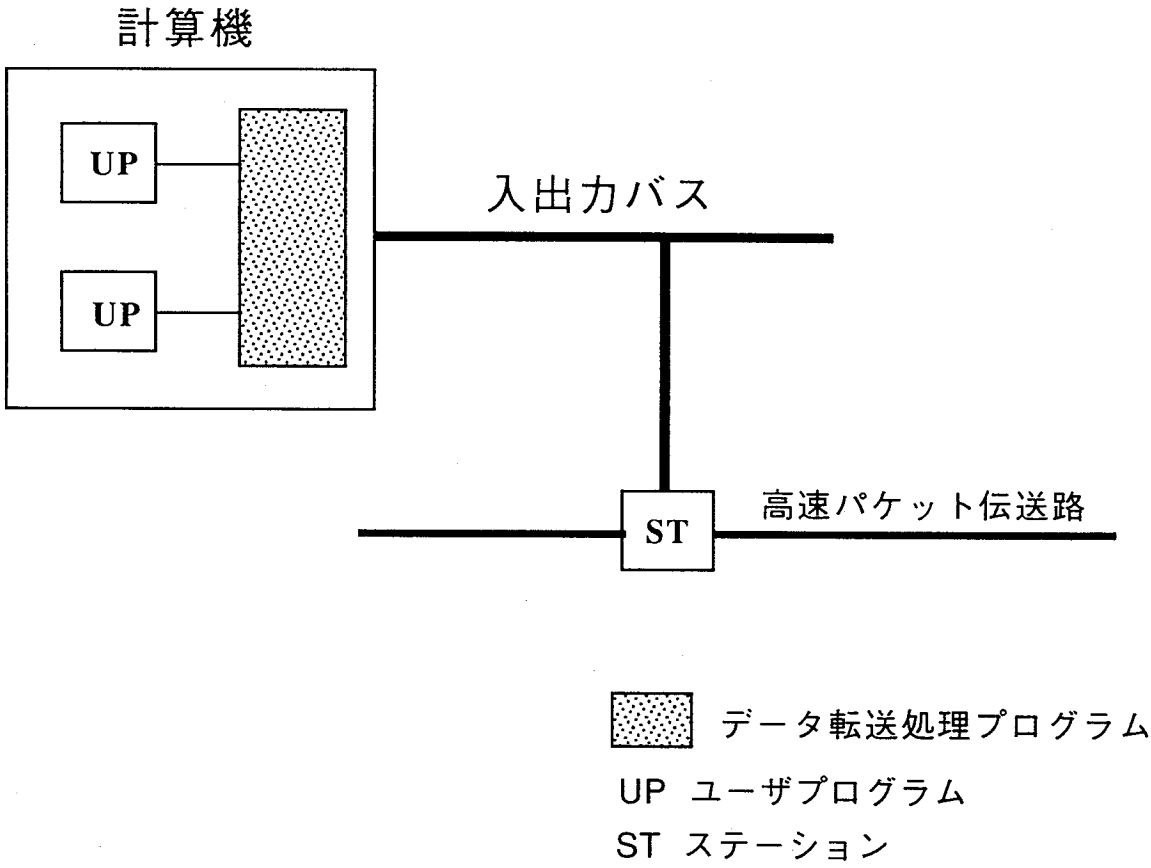
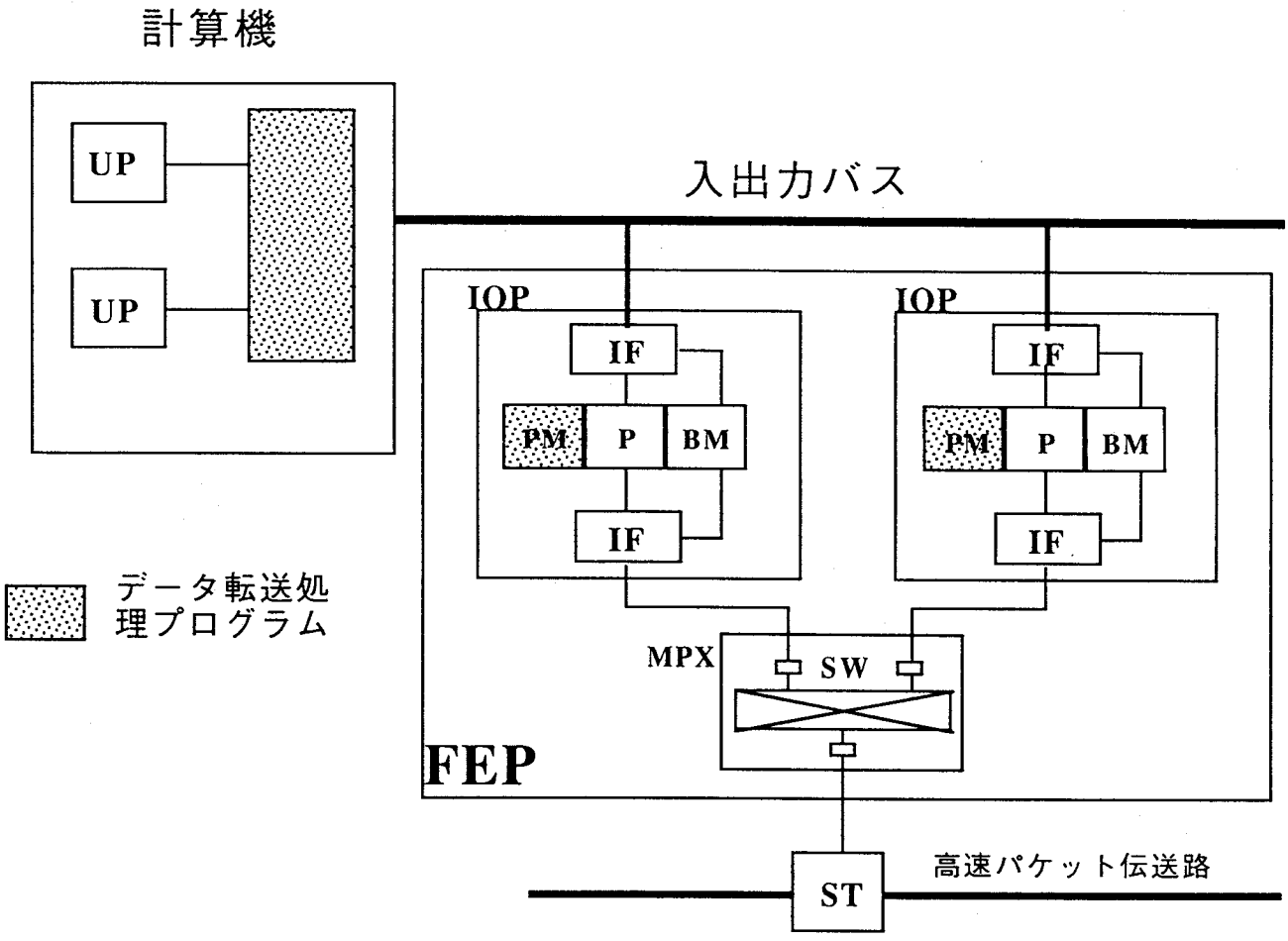


図2.9　従来の計算機とステーションの接続形態



IOP　マイクロコンピュータ
PM　プログラムメモリ
P　プロセッサ
BM　バッファメモリ
IF　インタフェース回路
SW　スイッチ

UP　ユーザプログラム
ST　ステーション
FEP　前置処理装置
MPX　マルチプレクサ

図２．１０　提案前置処理装置のシステム構成

本方式によれば、各マイクロコンピュータのメモリを少なくできるので、高速プロセッサ、高速バッファメモリを使うことができ、高速応答を得やすい。複数のマイクロコンピュータによる並列処理方式のため高いデータ転送性能が期待できる。さらに、ＦＥＰでメッセージのセグメンティング／リアセンブリを行うので、計算機は、メッセージ単位にＦＥＰの起動と終了の処理を行うだけで済み、計算機負荷の大幅な削減が期待できる。

２．４　実験的評価

２．４．１　評価の対象と目的

高速パケット伝送路を用いた計算機間のデータ転送をとりあげ、ＦＥＰの導入効果を、スループット、応答時間、計算機負荷の三つの観点から評価する。

２．４．２　評価方法

３章で提案した方式に従ってＦＥＰを製作し、ＦＥＰ付システムと従来システムとで性能を実測し、比較する。

ＦＥＰは、２台のＩＯＰとＭＰＸとから構成し、計算機の入出力バスとステーション（ＳＴ）との間に挿入した。ＭＰＸは、すべてハードロジックで構成した。ＩＯＰは、ビットスライス・マイクロプロセッサを用い、語長３２ビットのマイクロ命令を２００ｎｓで実行できるようにした。ＩＯＰの仕様を表２．１に示す。ＩＯＰのマイクロプログラムは、約３６００ステップで、従来、計算機のソフトウェアで行われていたデータ転送処理のうち、論理チャネルへの振り分け、セグメンティング／アセンブリ、およびステーションの制御を行う。輻輳制御、エラーリカバリは、上位レベルで行うものとし、本評価の対象外とする。

実測には、計算機２台を光データウェイで接続したシステム（図２．１１）を用いた。光データウェイは、伝送速度が１０Ｍｂｐｓで、パケットサイズが５１２バイトの高速パケット伝送路であり、ステーションを介して、ＦＥＰ／計算機と伝送路とを接続する。データウェイの仕様を表２．２に示す。データリンクの制御は、このステーションによって行われる。ステーションの仕様を表２．３に示す。計算機は、語長１６ビットのミニコンで、性能は、０．５MIPSである。

測定は、テストプログラムにより行った。テストプログラムは、計算機間デ

表２．１　　ＩＯＰの仕様

項番	項目		仕様	備考
1	プログラムメモリ	語長	３２ビット	
		容量	４０００語	
		サイクルタイム	４０ｎｓ	
2	データメモリ	語長	１６ビット	制御テーブル、パケットバッファとして600語を使用している
		容量	１０００語	
		サイクルタイム	５５ｎｓ	
3	演算部	語長	１６ビット	AMD社の2900シリーズを使用している 4ビット×4=16ビット
		レジスタ	１６個	
		速度	２００ｎｓ／命令	
		命令数	７０	
4	マイクロプログラム		３６００ステップ	

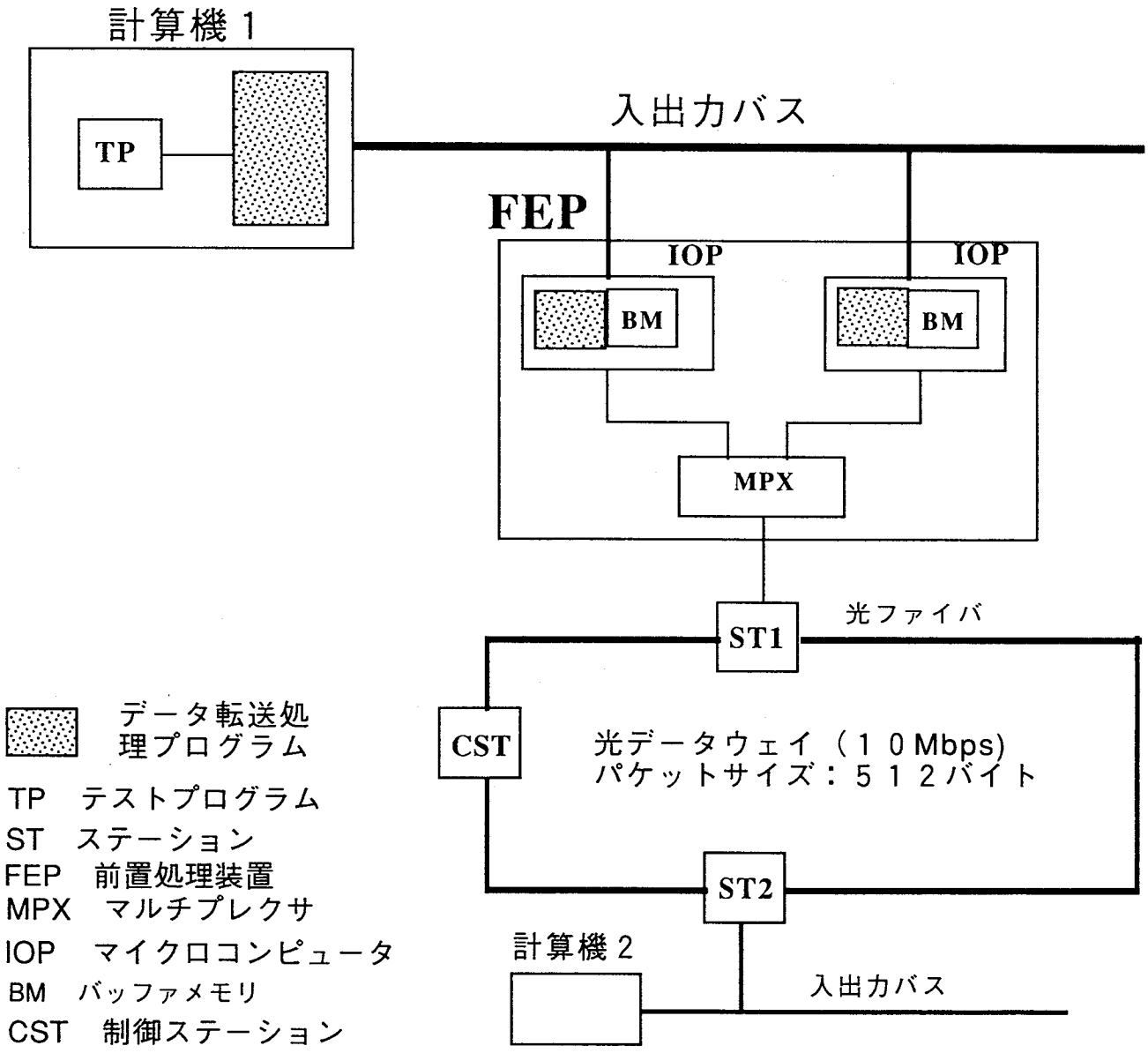


図２．１１　実測システムの構成

表２．２　データウェイの仕様

項　目	仕　　様
伝送速度	１０Ｍｂｐｓ
ネットワーク形態	ループ
ステーション数	最大２５５台／ループ
ステーション間距離	最大２ｋｍ
伝送媒体	光ファイバケーブル
パケット長	最大５１２バイト（可変長）
実効転送速度	最大９００ｋバイト／秒 （３２ステーション、１０ｋｍ／ループ）
誤り制御	ＣＲＣチェック フレーム合理性チェック
障害対策	制御ステーションの二重化 ステーション障害時のバイパス ハードウェア自動診断 リモート診断 統計ＲＡＳ情報トレース 伝送品質の監視 ループ使用率の監視

表２．３　　ステーションの仕様

項番	項目		仕様	備考
1	プログラムメモリ	語長	３２ビット	
		容量	２０００語	
		サイクルタイム	４０ｎｓ	
2	データメモリ	語長	１６ビット	
		容量	４０００語	
		サイクルタイム	５５ｎｓ	
3	演算部	語長	１６ビット	AMD社の2900シリーズを使用している 4ビットX4=16ビット
		レジスタ	１６個	
		速度	２００ｎｓ／命令	
		命令数	７０	
4	マイクロプログラム		２０００ステップ	

ータ転送処理プログラムを用いて、計算機１のメモリ上のデータを計算機２のメモリに転送する動作を繰り返し行う。テストプログラムからの転送要求を受けて、データ転送処理プログラムは、計算機１からＳＴ$_2$に、データと指令（計算機２のメモリアドレス、データの長さなど）を送る。ＳＴ$_2$は、この指令に基づき、ＤＭＡ方式で計算機２のメモリに書き込む。転送要求のあったデータ長がパケットサイズより大きい場合は、何回かに分けてＳＴ$_2$に書き込み要求を行う（セグメンティング）。１回の書き込み要求ごとに、計算機２のメモリへのデータ書き込みと応答パケットの返送をＳＴ$_2$が行う。従来システムとＦＥＰ付システムのタイムチャートを図２．１２、図２．１３に示す。各々の処理時間（ｔｉｊ）を表２．４に示す。測定条件は、下記とする。

(a) 測定時は、テストプログラムだけが動いており、他の業務処理プログラムは全く動いていない。
(b) スループットの測定は、二つのテストプログラムを同時に走らせて行い（多重度２）計算機負荷と応答時間の測定は、多重度１で行う。
(c) 性能は数分間同じ動作を繰返し、その間の平均値で算出する。

２．４．３　結果

(1) データ転送スループット

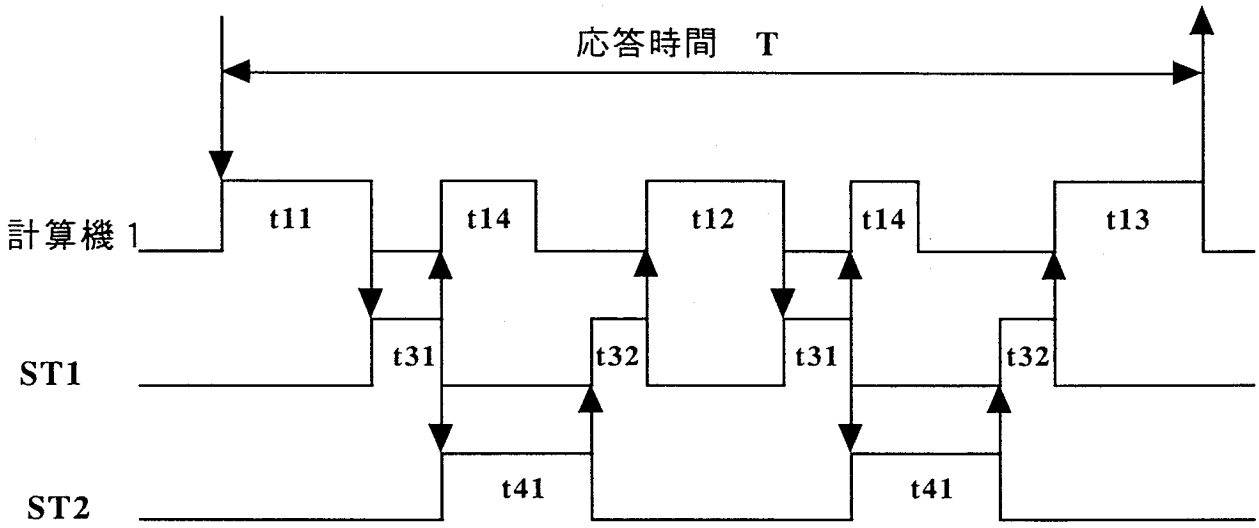
単位時間あたり、計算機１から計算機２に転送できたデータ量でスループットを定義する。この定義のもとに、種々のデータ長でスループットを測定し、比較したのが図２．１４である。これにより、データ長が１ｋバイトのとき２．５倍、８ｋバイトのとき３．５倍スループットが向上することがわかった。

(2) 応答時間

応答時間として図２．１２のＴおよび図２．１３のＴ’をとり、種々のデータ長で測定した結果を図２．１５に示す。サイクルタイム５５ｎｓの高速データバッファメモリ、５ＭＩＰＳの高速マイクロコンピュータを用いたこと、マルチプレクサでのパケットの蓄積を不要にしたことにより、応答時間は、データ長６０バイトで１／２に改善できた（データ長が長くなると改善度は少なくなるが、通常応答時間が最も問題になるのはデータ長の短いところである）。

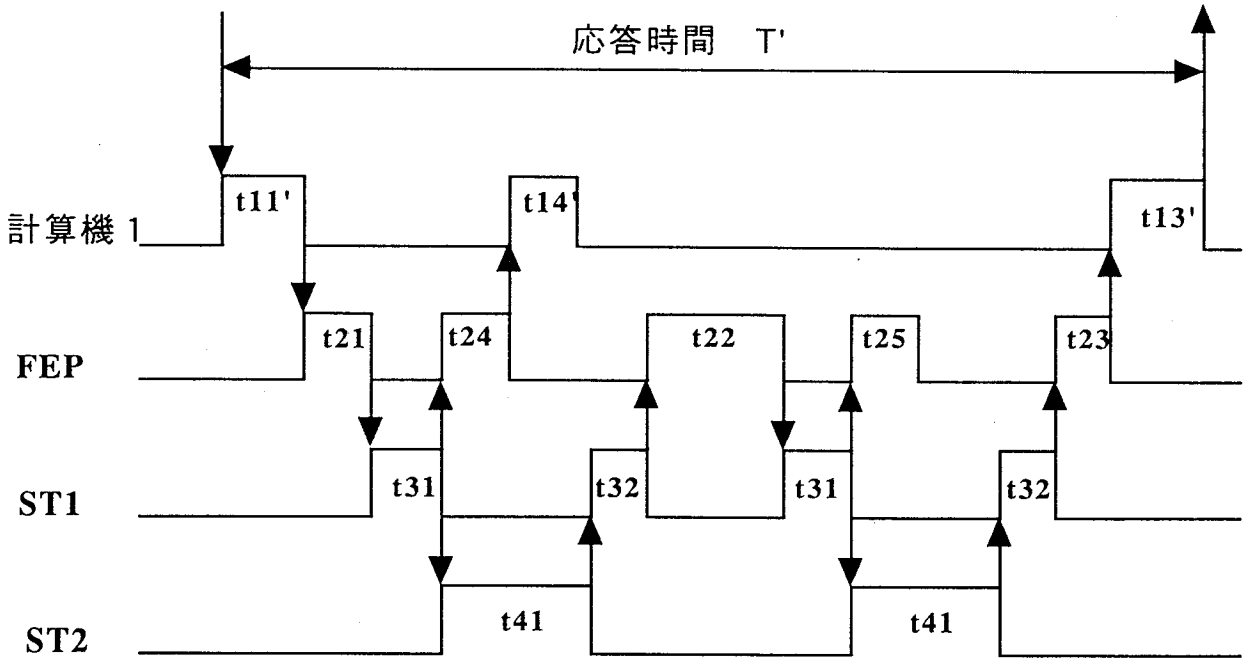
(3) 計算機負荷

データ長を変えて、テストプログラムを繰返し走らせて計算機負荷を測定し、ＦＥＰ付システムと従来システムの差をプロットしたのが図２．１６である。これより、データ長が８バイトと短いところでも４０％計算機負荷が減り、８ｋバイトという長いデータの場合は、８０％負荷が減ることがわかった。



tij　処理時間（詳細は、本文参照）
FEP　前置処理装置
（２セグメントの例、セグメント数はデータ長により変わる）

図２.１２　従来システム（FEPのない場合）の
データ転送タイムチャート



tij　処理時間（詳細は、本文参照）

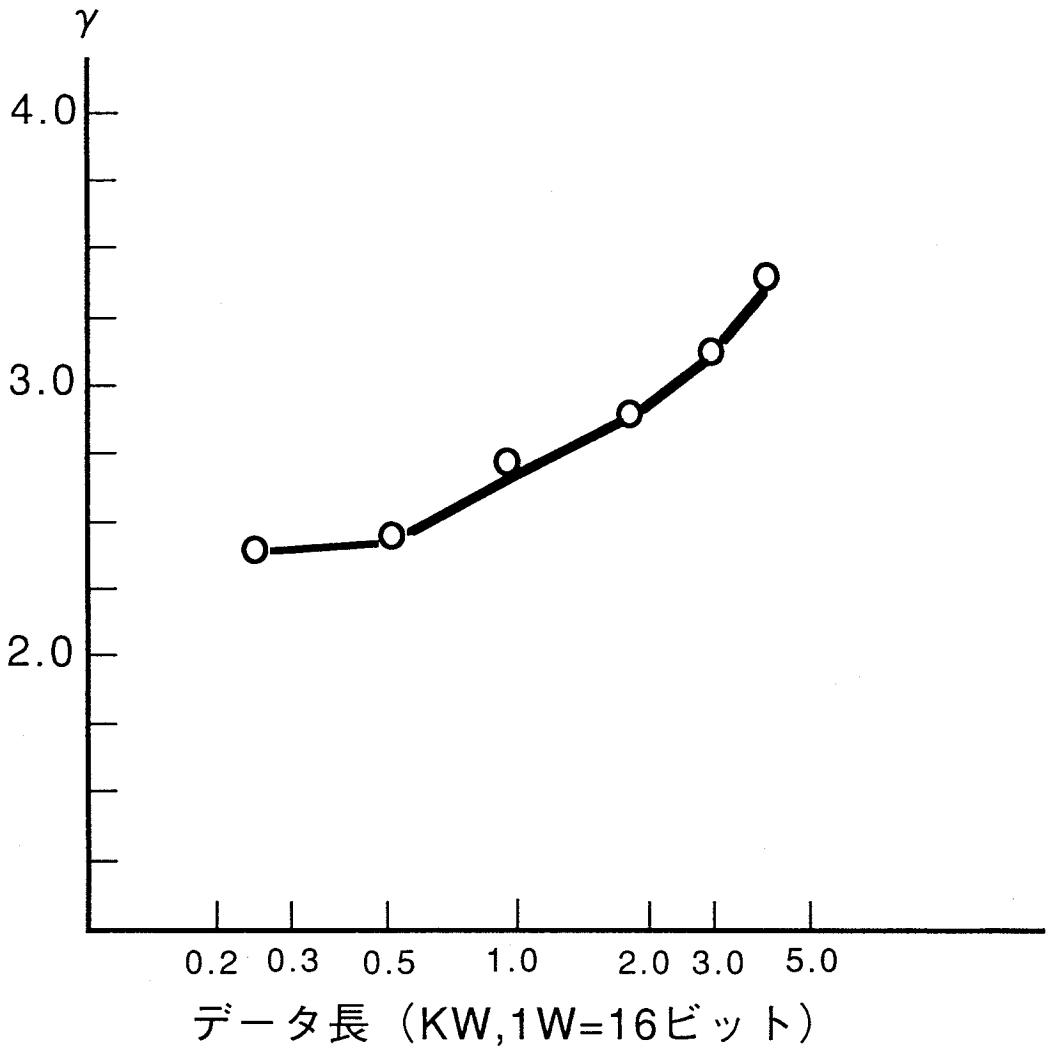
FEP　前置処理装置

（２セグメントの例、セグメント数はデータ長により変わる）

図２．１３　FEP付きシステムのデータ転送タイムチャート

表２．４　処理時間ｔijの内容

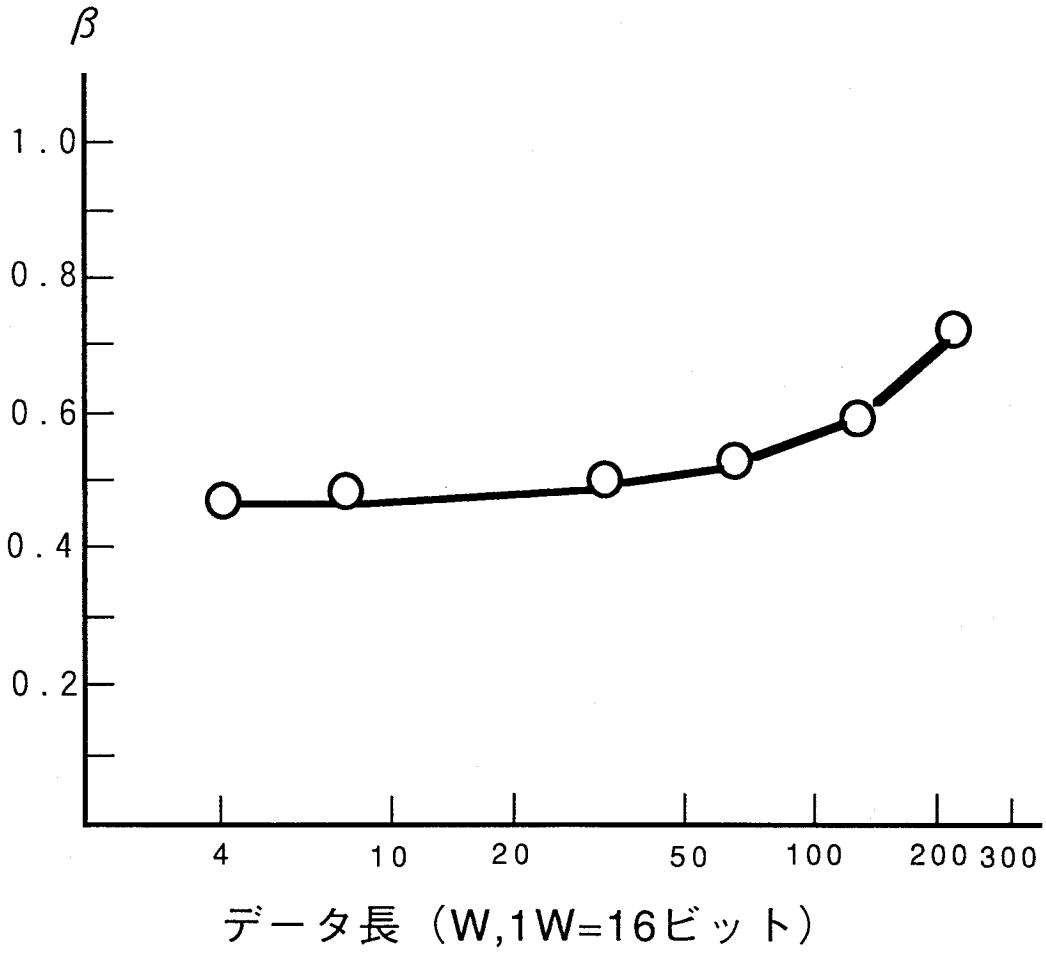
ｔij	処　理　内　容
t_{11}	テストプログラムから要求を受けて必要なヘッダを作成し、ステーションに送信要求を行う。
t_{12}	相手ＳＴからのレスポンスを受け、次のセグメントの送信要求を行う。
t_{13}	最後のセグメントに対するレスポンスを受けテストプログラムを再起動する。
t_{14}	パケット送信後処理。
t_{11}'	テストプログラムから要求を受けてＦＥＰに送信要求を行う。
t_{13}'	ＦＥＰからの割り込みを受け、テストプログラムを再起動。
t_{14}'	ＩＯＰのＢＵＳＹを解除。
t_{21}	ヘッダを作成し、ステーションに送信要求を行う。
t_{22}	相手ＳＴからのレスポンスを受け、次のセグメントの送信要求を行う。
t_{23}	最後のセグメントに対するレスポンスを受け、計算機に割込みを発生する。
t_{24}	ＩＯＰのＢＵＳＹを解除。
t_{25}	パケット送信後処理。
t_{31}	パケットの送信（ST_1）
t_{32}	パケットの受信（ST_1）
t_{41}	メモリへの書き込みを行い応答パケットを送信（ST_2）



γ＝γ１／γ０
γ０　：FEPのないシステムのスループット
γ１：FEP付きシステムのスループット

条件：２つのテストプログラムを同時に走らせ単位時間に計算機１から計算機２に転送できた合計データ量で比較した。

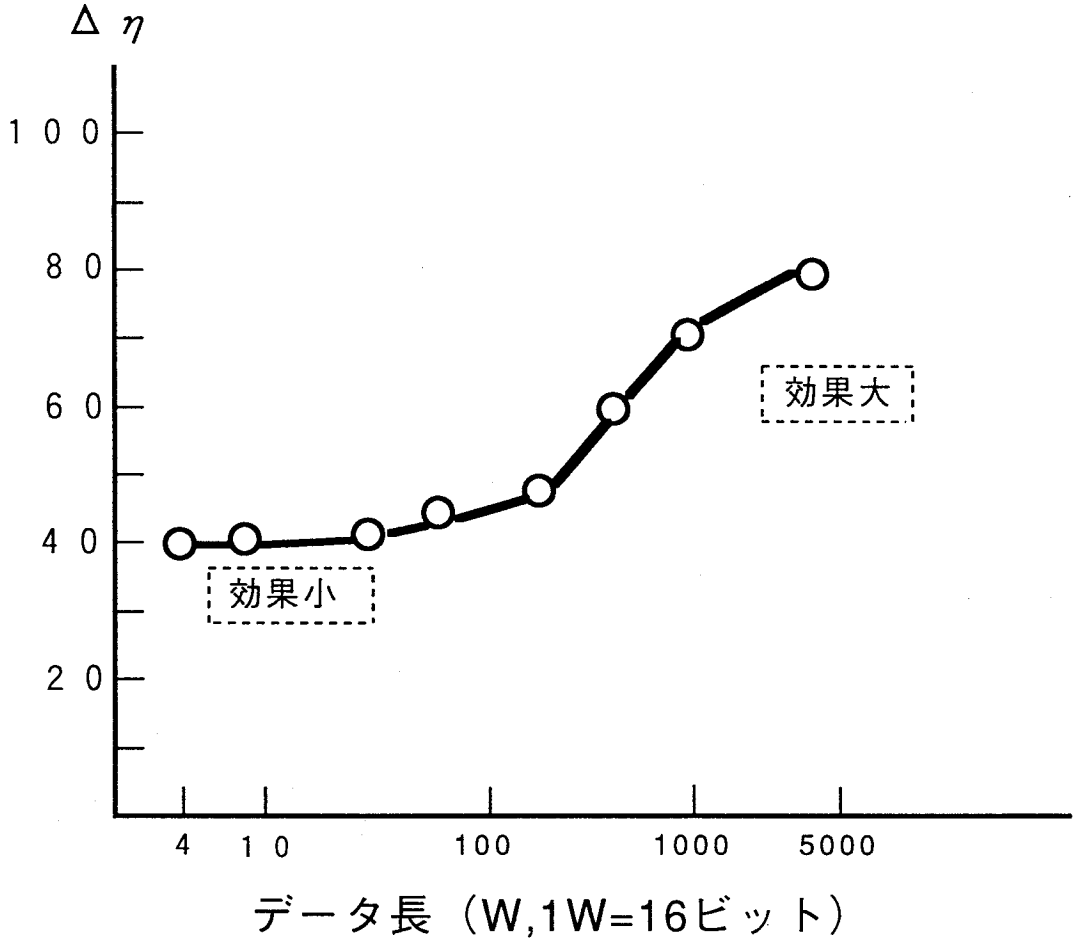
図２．１４　スループット比



$\beta = \beta 1 / \beta 0$
$\beta 0$：FEPのないシステムの平均応答時間
$\beta 1$：FEP付きシステムの平均応答時間

条件：１つのテストプログラム動かしたときの平均応答時間（図２.１２、図２.１３のTおよびT')を測定する。

図２.１５　応答時間比（FEPなしのときを１）



Δη＝η0－η1
η0　：FEPのないシステムの計算機１の負荷
η1：FEP付きシステムの計算機１の負荷

条件：１つのテストプログラム動かしたとき
　　の計算機１の負荷を測定する。

図２.１６　計算機負荷低減効果

２．５　結言

高速パケット通信ネットワークによって複数の計算機を接続した計算機制御システムを対象に、計算機間のデータ転送スループット、応答時間、計算機負荷などの性能を向上する方式を検討し、マルチマイクロコンピュータ構成のトランスポートプロセッサ（前置処理装置処理：ＦＥＰ）とデータウェイステーションとから構成する機能階層形多重通信プロセッサ方式を提案した。提案方式による性能向上度を把握するため、マイコン２台から成るＦＥＰを試作し、伝送速度１０Ｍｂｐｓの高速パケット通信ネットワークに適用した。　実験の結果、計算機間データ転送性能は、ＦＥＰが無いときに比べて、次のように改善されることが明らかになった。

(1) データ転送スループットが３倍になる（データ長が４ｋＢのとき）。
(2) 応答時間が１／２になる（データ長が６０Ｂのとき）。
(3) 計算機負荷が４０％（データ長が８Ｂのとき）～８０％（データ長が８ｋＢのとき）削減できる。

提案方式は、計算機制御システムにおいて、データウェイを用いた場合について論じた。これ以外のシステムでも高速パケット通信ネットワークと計算機との接続の場合には、提案方式が性能向上に役立つものと考えられる。

第３章　機能階層形多重通信プロセッサにおけるコマンド多重処理方式

３．１　緒言

近年計算機制御システムの大規模化が進み、計算機間のデータ転送性能の向上が強く求められている[4]。第２章では、トランスポートプロセッサ（前置処理装置：ＦＥＰ）とデータリンクプロセッサ（データウェイのステーション）とから構成する機能階層形多重通信プロセッサによって、データ転送性能を向上する方式をとりあげ、ＦＥＰの方式提案（ＦＥＰＩ）と性能向上度合の実験的評価とを行った。本章では、ＦＥＰＩで問題になったハード量を削減するため、新しいＦＥＰの方式提案（ＦＥＰII）と実測およびシミュレーションによる性能評価とを行っている。

(1) ハード量削減のため、ＦＥＰを１台の高速マイクロコンピュータで構成し、ソフトウェア方式の工夫によって多数のコマンドを同時に処理する。
(2) 性能を向上するため、ＦＥＰのソフトウェア構造をコンパクトなスケジューラのもとで動くマルチタスク構造とし、処理の並列化を行う。
(3) この方式に基づき、ビットスライスマイクロプロセッサを用いて、ＦＥＰを開発し、性能をシミュレ-ションで求め、マイコン1台でも十分な性能が得られることを確認している。

３．２　機能階層形多重通信プロセッサ方式実用化の課題

３．２．１　ＦＥＰの位置付け

計算機制御システムの中で、データウェイによって、複数の計算機が接続されている例[17][18][19]をとりあげ、以下の議論を進める。データウェイは、光ファイバ／同軸ケーブルとステーションとで構成されている。ステーション相互間では、パケットと呼ばれる５００～１５００バイト程度の可変長データブロックが１０～３２Ｍｂｐｓでビットシリアルに伝送される。

計算機間のデータ転送は、主にホスト計算機と１０～２０台の端末計算機との間で行われる。すなわち、計算機のユーザプログラム相互間には、論理チャネルと呼ばれる仮想的な伝送路が定義され、論理チャネルを介して、ユーザプ

ログラム間でメッセージがやりとりされる。ＦＥＰは、図３．１に示すように、ホスト計算機とステーションとの間にあって、従来ホスト計算機で行っていたパケット単位のデータ転送処理をホスト計算機に代わって高速に実行する。

ホスト計算機とＦＥＰのインタフェースは、仮想的な伝送路である論理チャネルである。各論理チャネル上でのデータ転送手順は、図３．２に示すように、パケット毎に応答をとる方式である。ホスト計算機からは、メッセージ単位にＦＥＰを起動し、パケットへの分割（セグメンティング）／編集（アセンブリ）は、ＦＥＰで行う。ＦＥＰは、プロトコルのレイアで表現すれば、トランスポートレベルの処理まで行う。

３．２．２　従来方式の問題点

ＦＥＰの処理は、コマンドが発行されている多数の論理チャネルに対して並列に行われる。この結果、ＦＥＰの負荷は重く、メモリも多く必要になる。これを解決するため、第２章では、ＦＥＰを複数のマイクロプロセッサで構成する方式を提案し、効果を評価した。この方式は、ステーションとＦＥＰのマイクロプロセッサとの間に、マルチプレクサ（ＭＰＸ）と呼ぶ装置を挿入することによって、高速に受信パケットをＦＥＰのプロセッサに分配し、各プロセッサで並列に処理する方式である（以下、この方式のＦＥＰをＦＥＰＩと呼ぶ）。しかし、ＦＥＰＩには、実用化する上で次の問題があることがわかってきた。

(1) 処理を高速化するため、マイクロプロセッサとしては、集積度の低いバイポーラ技術を用いたビット・スライス・マイクロプロセッサを使う必要がある。このため、マイクロプロセッサ一台が基板（30.5×42.5 cm）一枚のハード量となる。
(2) マルチプレクサが、マイクロプロセッサ２～６台とインタフェースするために基板一枚のハード量となり、最小構成でもプロセッサと合わせて基板二枚になる。
(3) 個々のマイクロプロセッサは、同時には、一つのコマンドのみ処理する方式にしているため、パケットの送信中、マイクロプロセッサがアイドル状態になっている。

このように、ＦＥＰＩでは、性能向上効果はあるが実用化するには、ハード量が大きくなりすぎるという問題があった。

３．２．３　改良形ＦＥＰ（ＦＥＰⅡ）

本章では、前記ＦＥＰＩの問題点を解決した新しい方式のＦＥＰを提案する

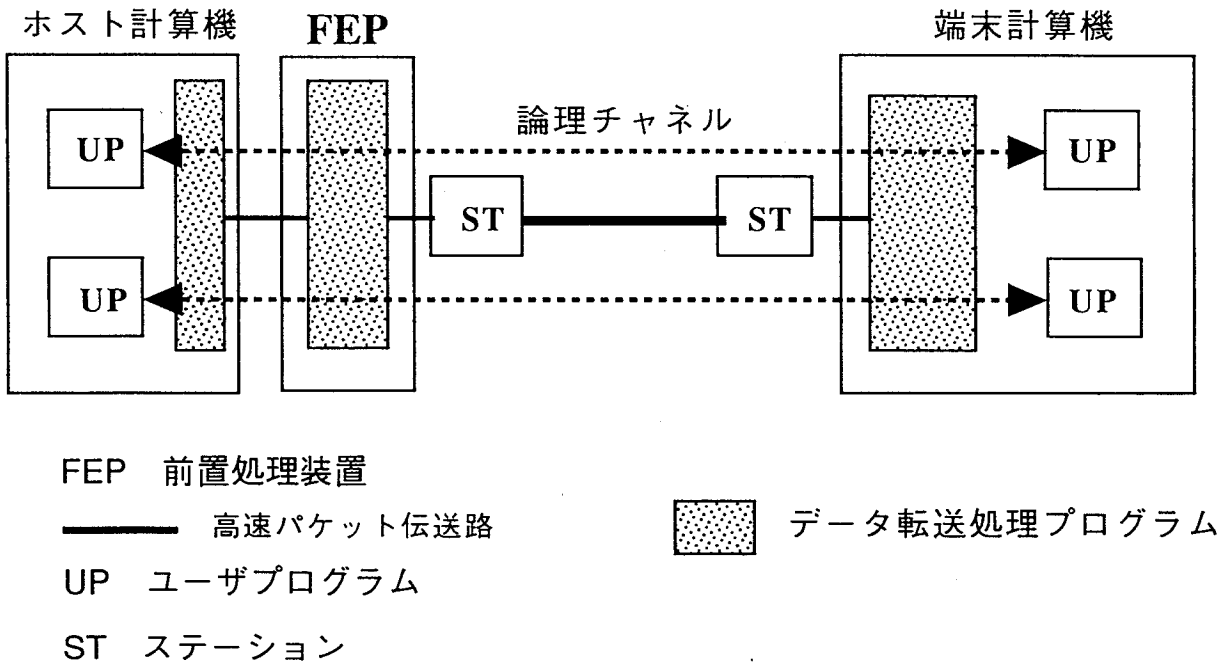
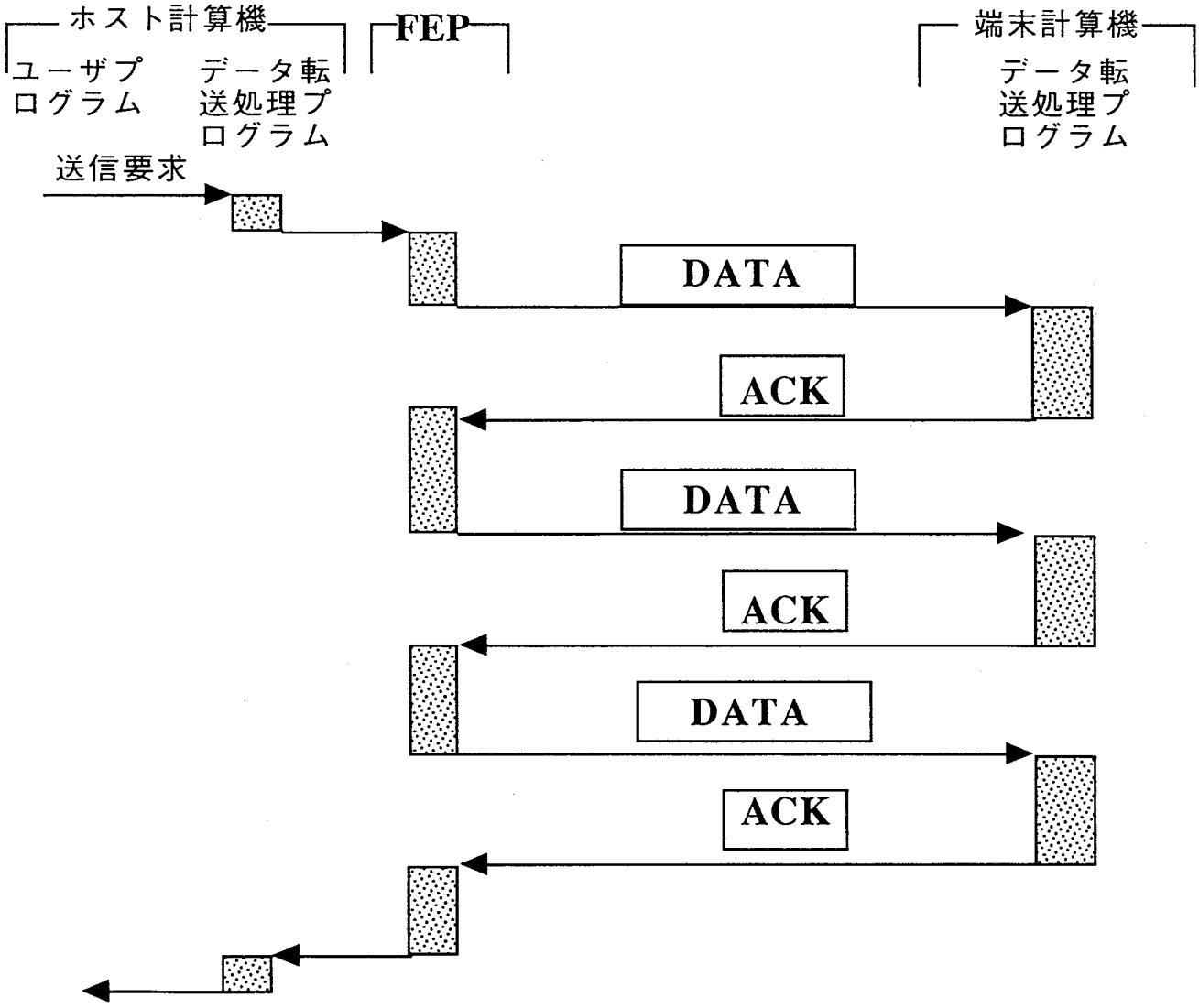


図３．１　前置処理装置（FEP）の位置付け



DATA　ユーザデータパケット

ACK　肯定応答パケット

FEP　前置処理装置

論理チャネルレベルのデータ転送モデルを示している

図３．２　FEP適用時のデータ転送モデル

（これをＦＥＰⅡと呼ぶ）。ＦＥＰⅡでは、下記の方針により、ハード量を減らす工夫をする。

(1) マイクロプロセッサはステーション対応に一台として、マルチプレクサ（ＭＰＸ）を不要にする。
(2) マイクロプロセッサは、同時に複数のコマンドを処理できるようにして、アイドルタイムを極力減らし、一台である程度の性能を確保する。

上記改良により、マルチプレクサ（ＭＰＸ）が不要になり、マイクロプロセッサが一台ですむので、ハード量の低減が期待できる。

３．３　コマンド多重処理方式の開発

本節では、高速パケット伝送路に適したコストパフォーマンスのよい前置処理装置（ＦＥＰⅡ）のコマンド多重方式について論じる。

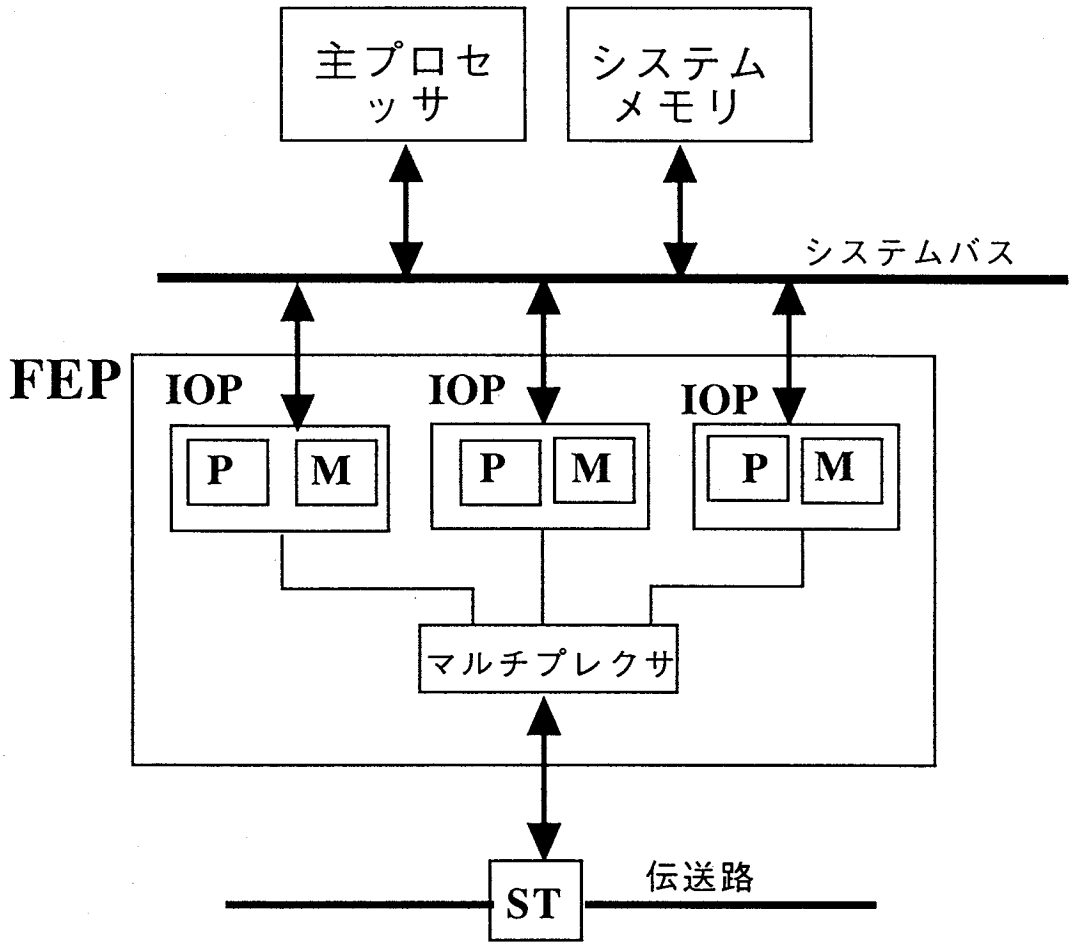
３．３．１　要求条件とＦＥＰの方式提案

高速パケット伝送路を用いた計算機制御システムにおいて使用されるＦＥＰには、下記の要求条件がある。

(1) １０～３２Ｍｂｐｓの伝送速度に適応できる高いデータ転送スループット
(2) 応答時間が数ミリ秒という高速応答性
(3) 制御用計算機のハード規模に見合うコンパクトなハードウェア

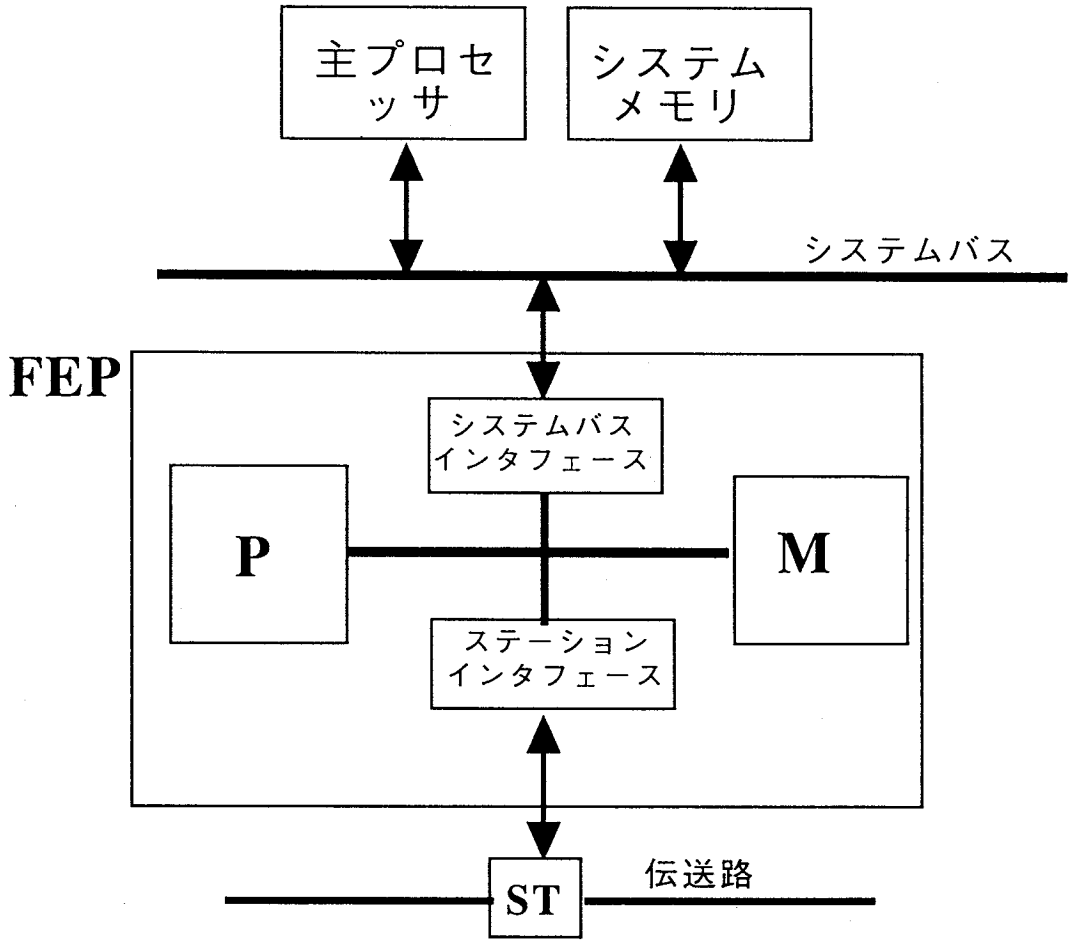
前記要求条件を満たすＦＥＰ構成方式としては、マルチプロセッサ・シングル・コマンド処理方式とシングルプロセッサ・マルチコマンド処理方式とが考えられる。前者は、高性能化には適すがハード量が増大するという欠点があり、後者は、ハード量を削減できるが、高性能化に弱点がある。本章では、後者のシングルプロセッサ・マルチコマンド処理方式を採用し、コマンド多重処理などのソフトウェア技術を駆使して高性能化を図ったＦＥＰ（ＦＥＰⅡ）を提案する。

ＦＥＰⅡの構成を、従来のマルチプロセッサ方式のＦＥＰ（ＦＥＰＩ）（図３．３）と対比して、図３．４に示す。ＦＥＰⅡでは、マイクロプロセッサを一台にし、ステーションとマイクロプロセッサを１：１に対応させて、マルチプレクサ（ＭＰＸ）を不要にしている。ＦＥＰⅡのプロセッサ仕様を、ＦＥＰＩのプロセッサ（ＩＯＰ）と対比させて、表３．１に示す。ＦＥＰⅡでは、基



FEP　前置処理装置
ST　ステーション
IOP　マイクロコンピュータ
M　メモリ
P　プロセッサ

図３．３　マルチプロセッサ方式によるFEPの構成



FEP　前置処理装置
ST　ステーション
M　メモリ
P　プロセッサ

図３．４　シングルプロセッサ方式によるFEPの構成

表３．１　ＦＥＰに使用したプロセッサボードの比較

項番	項目		ＦＥＰ－ＩのＩＯＰ	ＦＥＰ－Ⅱ
1	プログラムメモリ	語長	３２ビット	←
		容量	４０００語	８０００語
		サイクルタイム	４０ｎｓ	５０ｎｓ
2	データメモリ	語長	１６ビット	←
		容量	１０００語	１６０００語
		サイクルタイム	５５ｎｓ	←
3	演算部	語長	１６ビット	← (AMD社の2900シリーズ 4ビット×4=16ビット)
		レジスタ	１６個	
		速度	２００ｎｓ／命令	
		命令数	７０	
4	ハード量（プラグイン枚数）		1	2

板２枚に、①一命令２００ｎｓの高速マイクロプロセッサ、②プログラムメモリ：８ｋＷ（ＦＥＰＩの２倍）、③データメモリ：１６ｋＷ（ＦＥＰＩの１６倍）④ステーションとのインタフェース回路、⑤システムバスとのインタフェース回路をすべて収容している。

上記方式により、ハード量は、基板２枚にできる見通しが得られたが、コマンド多重処理ソフトウェア方式をどうするか、そして、それによってどの程度の性能が得られるかが問題である。次に、これらの点について検討する。

３．３．２　ＦＥＰⅡのソフトウェア処理方式

ＦＥＰⅡでは、ハード量を減らすため、マイクロプロセッサを１台にしたため、高度なソフトウェア処理によって、プロセッサの能力を十分に活用する必要がある。

(1) コマンドの多重受付

ＦＥＰがトランスポートレベルの処理まで行うためホストからＦＥＰへは論理チャネル対応に下記のコマンドを多重に発行できるようにする。

・ＲＥＣＶ：論理チャネルを指定し、メッセージの到着を待つ。

・ＳＥＮＤ：論理チャネルを指定し、メッセージを送信する。

ここで、メッセージの長さは、伝送路のパケット長を越えていてもよいものとする。コマンドは、論理チャネル対応に多重に受け付けるが、同一論理チャネルには、一つのコマンドのみ受け付ける。

(2) コマンドの並列処理

コマンド並列処理とは、ホスト計算機から受け付けたコマンドをＦＥＰ内部で並列に処理することである。送信（ＳＥＮＤ）を例にとると、各コマンドはまずプロセッサのソフトによる起動処理を受け、送信データがステーションを経由して伝送路に送出され、相手計算機に送られる。やがて相手計算機から応答が返される。応答が、ステーションで受信されると、コマンドは、再びプロセッサによる継続処理（セグメンティングがある場合）または終了処理を受ける。

ＦＥＰⅡでは、次のようにして、コマンドの並列処理を実行している。

・プロセッサの処理を細分化し、各々の処理をタスクとして、コンパクトなスケジューラで制御するマルチタスク構造とする。

・各コマンドは、コマンド処理テーブルに格納された形で、順次タスクの処理を受けていけるように各タスクの前に待ち行列を作って待たせておく。

・各タスクＴｉは、起動されると自タスクの前の処理待ち行列Ｑｉを調べ、処理待ちコマンドがあれば処理を行い、そのコマンドを次のタスクＴｊの待ち行列Ｑｊにつけ、タスクＴｊを起動する。
・スケジューラは起動されたタスクを次々とディスパッチしていき、プロセッサ資源を、処理を必要としているタスクにきめ細かく配分する。

(3) メモリの節約

コマンドの多重処理を行うには、多数のコマンド処理テーブルが必要になる。しかし、ハード量を減らすには、メモリを節約する必要がある。コマンド処理テーブルのメモリ容量を削減するため下記の方策をとった。

・コマンド処理テーブルの動的割当
コマンドがすべての論理回線に同時に発行される可能性は小さいことに着目し、コマンド処理テーブルを動的に論理チャネルに割り当てる方式にした。
・パケットバッファリング
コマンドの処理には、ユーザデータのバッファリングが必要になる。とりわけ、ＦＥＰでセグメンティング／アセンブリを行う場合、メッセージバッファをＦＥＰで持つと、ＦＥＰのメモリ量が増大する。そこで、ＦＥＰⅡでは、メッセージバッファはホストプロセッサの主メモリに置き、ＦＥＰではパケットバッファのみ持つことにした。セグメンティング／アセンブリは、ＦＥＰが主プロセッサのメモリを直接アクセスすることによって行う。
・パケットバッファとコマンド処理テーブルは分離しておき、送受信時のみ、パケットバッファを動的に確保する。

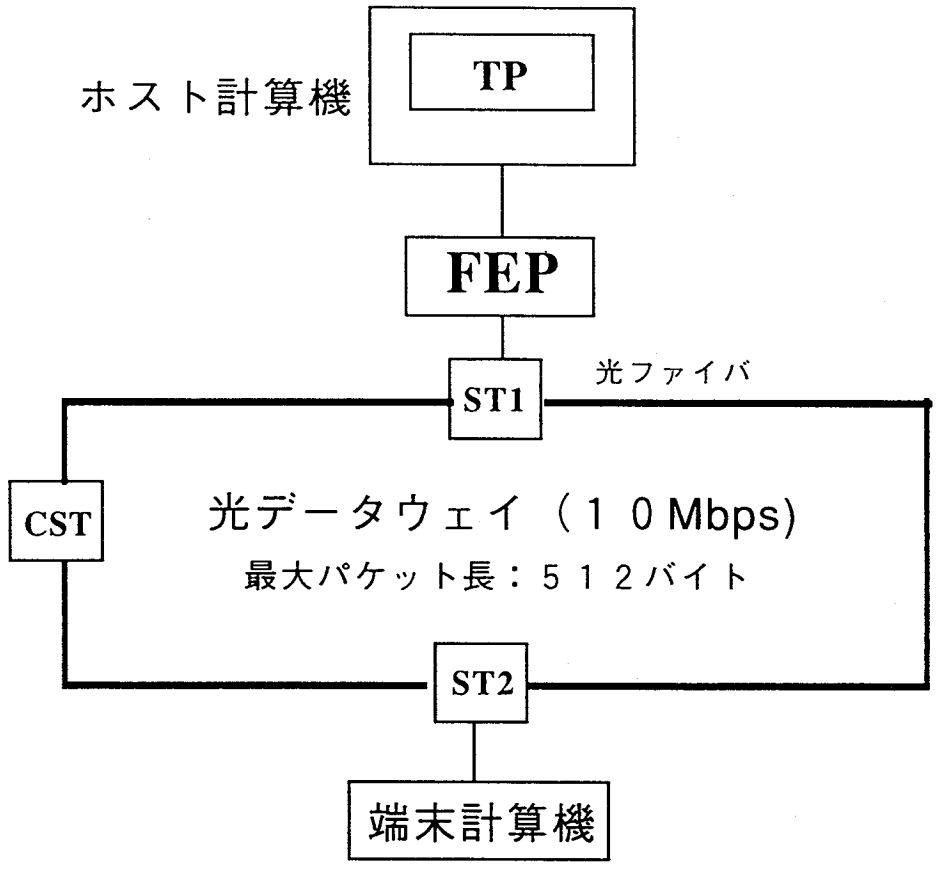
３．３．３　ＦＥＰⅡの性能評価

ＦＥＰⅡで採用したシングルプロセッサ・マルチコマンド処理方式では、コマンド多重処理による高性能化が重要な課題である。ＦＥＰⅡでは、前節で述べたソフトウェア処理方式によって、プロセッサの処理能力を最大限に引き出し、高性能化をはかっている。本節では、メッセージ転送スループットを例に、ＦＥＰＩとＦＥＰⅡの性能をシミュレーションにより求め、比較する。

(1) シミュレーションモデル

計算機間のメッセージ転送を想定する。メッセージは、計算機内のユーザプログラムから別の計算機のユーザプログラムに渡される。ここでは、図３．５に示すように、一台の計算機に付加されたＦＥＰに着目して行う。

メッセージは、伝送路に送出されるとき、ＦＥＰによってパケットに分割さ



FEP　前置処理装置
ST　ステーション
TP　テストプログラム
CST　制御ステーション

図３.５　性能比較のためのモデルシステム構成

れ、パケット毎に相手計算機から応答（ＡＣＫ）を受け取るものとする。この分割されたメッセージの断片をセグメントと呼ぶ。

メッセージ転送処理のタイムチャートを図３．６に、各処理時間を定義と共に表３．２に示す。処理時間は、１パケット（５１２バイト）の伝送時間を単位として示している（また処理時間は、ＦＥＰＩとＦＥＰⅡの具体ハード、ソフトの処理時間を根拠としている）。なお、相手計算機でメッセージパケットを受信後、応答パケットを返送するまでの時間ｔ51は、ＦＥＰの性能と直接関係しないので０としている。同様に、ｔ32、ｔ42も０としている。

ＦＥＰＩでは、プロセッサがステーションに送信要求した後も、ステーションからの送信終了割り込みを待ってアイドル状態となり解放されないことに注意する必要がある（シングルコマンド処理方式）。これに対して、ＦＥＰⅡでは、ステーションに送信要求を行った後、プロセッサは、次のコマンドを処理できる（コマンドの多重処理）。

(2) シミュレーションの条件

シミュレーションは、ＧＰＳＳを用いて下記の条件で行った。

・メッセージ送信要求は、ポアッソン到着とする。
・メッセージの長さは、平均２０００バイトの指数分布とする。
・パケットの最大長は、５１２バイトとする。
・伝送速度は、１０Ｍｂｐｓとする。

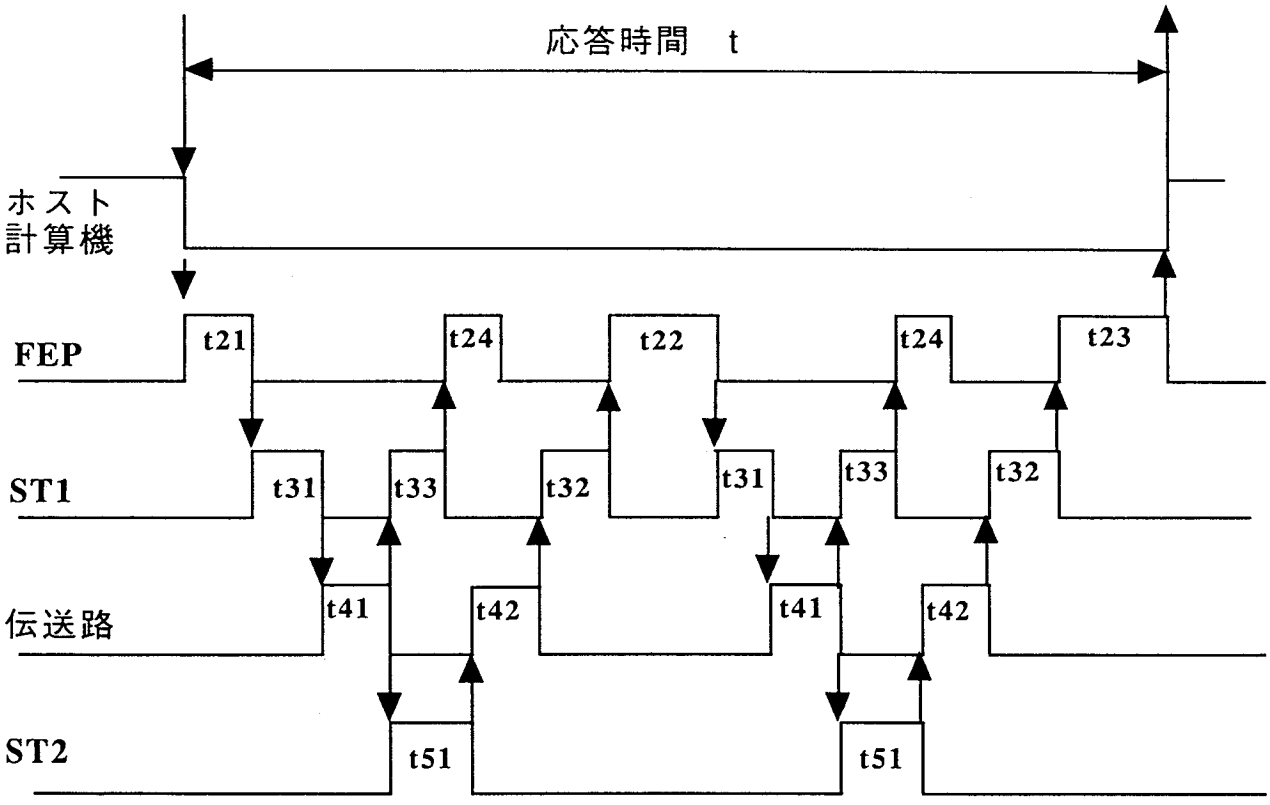
(3) シミュレーション結果

単位時間あたり、計算機から伝送路へ転送できたデータ量を、メッセージ転送スループット、計算機がＦＥＰに送信要求してから、送信が終了して、計算機に終了を通知するまでの時間（図３．６のｔ）を応答時間として、スループットと応答時間の関係を図３．７に示す。この結果、表３．２の処理時間のもとでは、ＦＥＰⅡの性能は、ＦＥＰＩの２プロセッサとほぼ同等であることがわかった。

３．３．４　ＦＥＰⅡの方式評価

シングルプロセッサ多重コマンド処理方式のＦＥＰⅡを提案し、ＦＥＰＩと対比しつつ、方式や性能を述べてきた。本節では、これまでの検討結果を整理し、ＦＥＰⅡの方式をコストパフォーマンスを中心に評価する。

ＦＥＰＩとＦＥＰⅡの二つの方式を表３．３に整理して示す。ＦＥＰＩのマルチプロセッサ方式では、プロセッサの数を、１、２、３と増すことによって性能を向上することができ、３台では最大０．２５程度のスループットが得ら



tij　処理時間（詳細は、本文参照）

FEP　前置処理装置

（２セグメントの例、セグメント数はデータ長により変わる）

図３.６　性能比較モデルタイムチャート

表３．２　性能比較モデルで用いた処理時間

tij	ＦＥＰ　Ⅰ	ＦＥＰ　Ⅱ	処　理　内　容
t_{21}	３．１＊	４．８＊	ホスト計算機から起動され、最初のセグメントをステーションに送信するまでの時間
t_{22}	４．８＊	５．９＊	端末計算機からの応答を受けて、次のセグメントをステーションに送信するまでの時間
t_{23}	２．１	２．２	端末計算機からの最終セグメントに対する応答を受けて、ホスト計算機に終了を通知するまでの時間
t_{24}	０．６	０．３	ステーションからの送信完了割り込みの処理時間
t_{31} t_{33}	２．２	２．２	１パケットの送信に要するステーションの処理時間
t_{41}	１．０＊	１．０＊	１メッセージパケット（最大５１２バイト）の伝送時間
t_{32}	０	０	ステーションにおける応答パケット受信処理時間
t_{42}	０	０	応答パケット（約６０バイト）の伝送時間
t_{51}	０	０	受信ステーション側でメッセージパケットを受信して、応答を返すまでの時間

＊データ長により異なる。この数値は、５１２バイトのときのもの。
各処理時間ｔij は、　１パケット（５１２バイト）の伝送時間（t_{41}）を単位として表している。

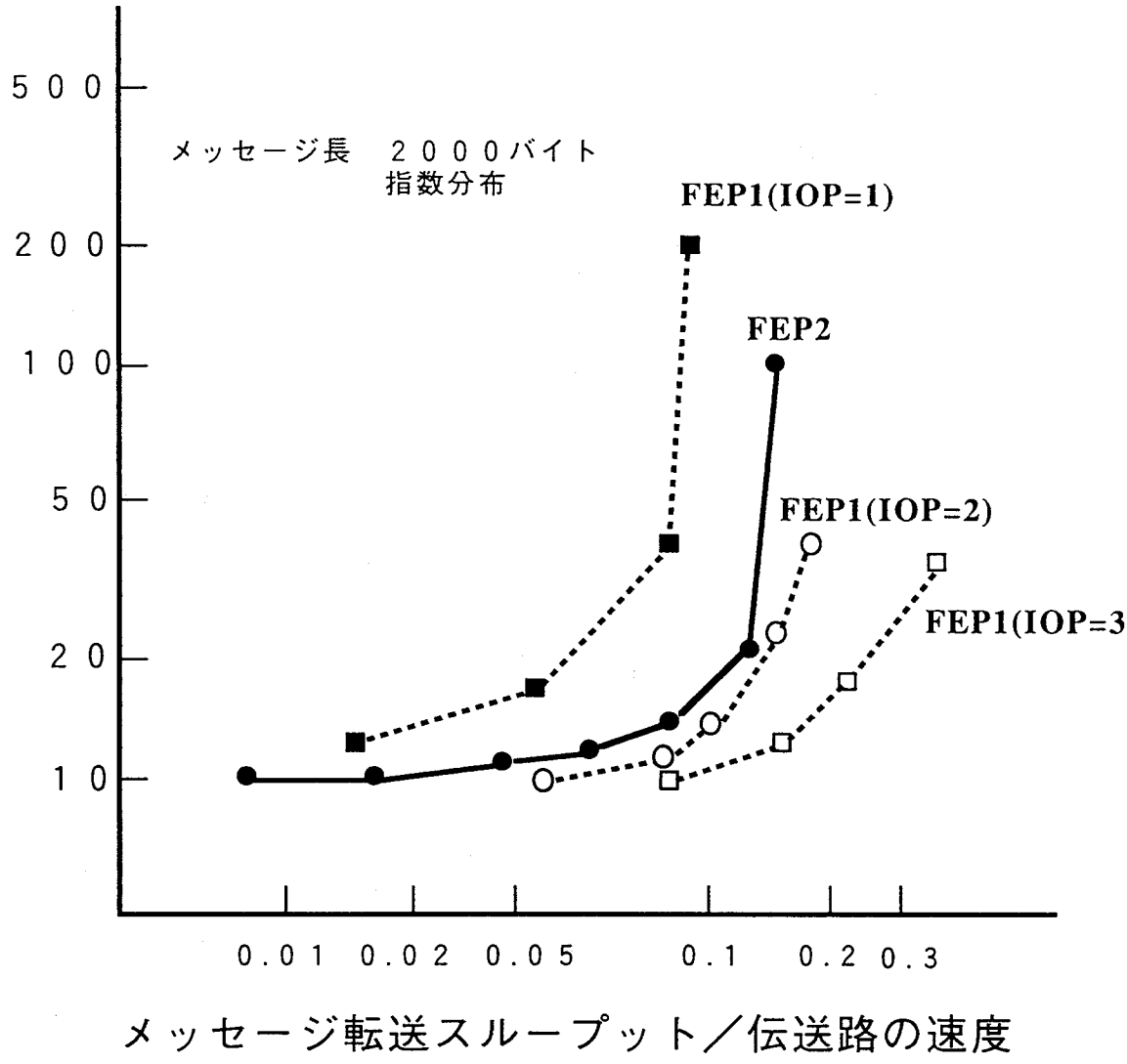


図3.7　FEPの性能比較（シミュレーション）

表３．３　ＦＥＰの方式比較

<table>
<tr><th>項番</th><th>項　　目</th><th colspan="2">ＦＥＰ－ＩのＩＯＰ</th><th>ＦＥＰ－Ⅱ</th></tr>
<tr><td>1</td><td>プロセッサ数</td><td colspan="2">n</td><td>1</td></tr>
<tr><td>2</td><td>同時処理数
（１プロセッサあたり）</td><td colspan="2">1</td><td>m</td></tr>
<tr><td>3</td><td>マルチプレクサ</td><td colspan="2">要</td><td>不要</td></tr>
<tr><td>4</td><td>制御の複雑さ</td><td colspan="2">ハード複雑</td><td>ソフト複雑</td></tr>
<tr><td>5</td><td>ハード量［基板枚数］</td><td colspan="2">ｎ＋１</td><td>２</td></tr>
<tr><td>6</td><td>プログラム規模
（ステップ）</td><td colspan="2">３６００</td><td>８０００</td></tr>
<tr><td rowspan="3">7</td><td rowspan="3">性能（伝送速度との比）</td><td>ｎ＝１</td><td>０．０５</td><td rowspan="3">０．１４</td></tr>
<tr><td>ｎ＝２</td><td>０．１５</td></tr>
<tr><td>ｎ＝３</td><td>０．２５</td></tr>
</table>

れる。これに比してシングルプロセッサ方式であるＦＥＰⅡでは、約０．１４であり、これ以上の性能が必要な場合には、ステーション＋ＦＥＰⅡをもう一式置く必要がある。ハード量と性能の関係を図３．８に示す。これより０．０５～０．１４においては、ＦＥＰⅡがＦＥＰＩよりコストパフォーマンスがよいことがわかる。

以上の検討から、シングルプロセッサ方式によってハード量を基板２枚に収め、しかも、２台のプロセッサによるマルチプロセッサ方式に相当する高性能をソフトウェアのコマンド多重処理により得られることがわかった。

３．４　適用評価

３．４．１　評価の対象と目的

ＦＥＰⅡを制御用３２ビットスーパミニコンに適用し、高速パケット伝送路である１０Ｍｂｐｓ光データウェイ用前置処理装置として機能させ、その効果を評価する。

３．４．２　評価方法

３．３節で提案した方式に従ってＦＥＰを製作し、ＦＥＰの各部処理時間をマイクロプログラムのステップ数から算出する。こうして得られた処理時間をもとに、シミュレーションにより、応答時間、スループット、主プロセッサ利用率を求める。同じ事を、ＦＥＰを適用していない１６ビット制御用ミニコンの場合に行い比較する。

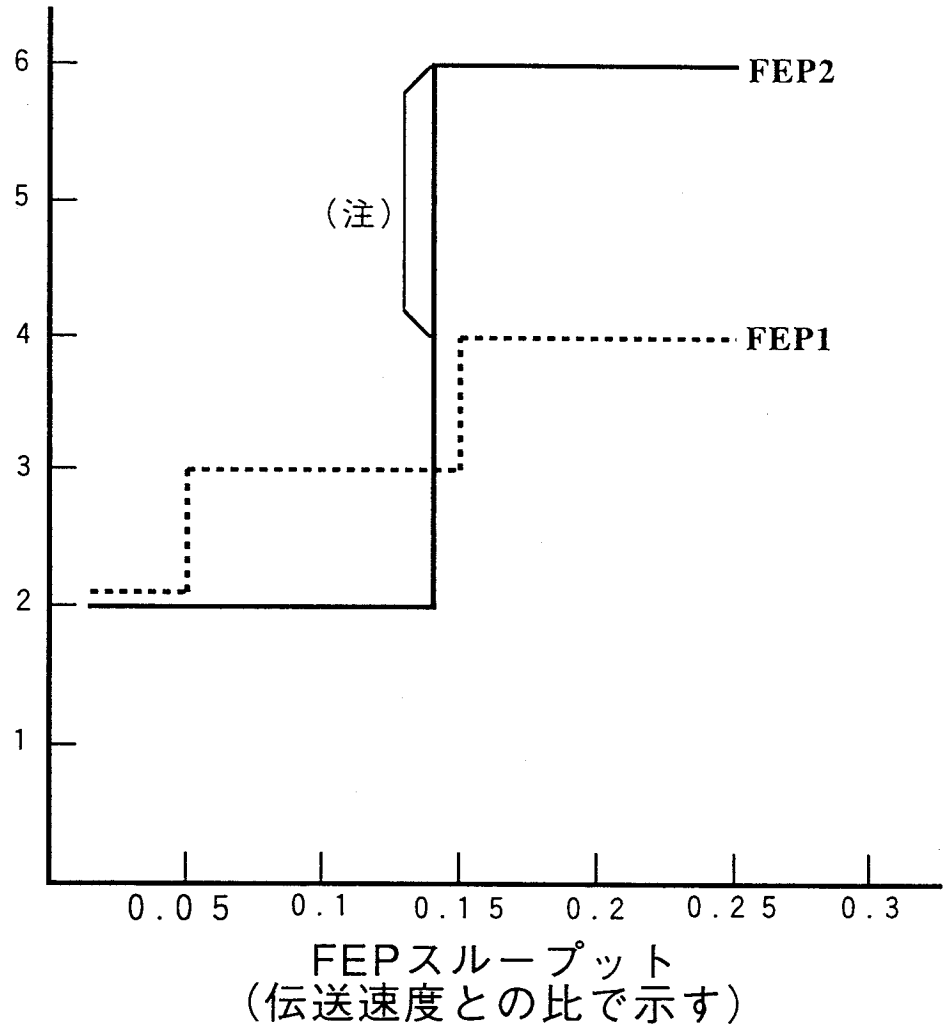
（１）性能評価システムの構成

２台の計算機が高速データウェイで接続された図３．９のようなシステムを考える。ＦＥＰⅡは、ホスト計算機にＤＷＰとして接続されている。データの転送は、ホスト計算機のユーザプログラムからの送信要求の発行によって行われる。パケット長を超えるメッセージの送信が指令された場合、ＦＥＰは、セグメンティングを行う。１セグメントごとに、相手計算機からの応答を待つ。データ転送タイムチャートを従来システムの場合（図３．１０）と対比させて図３．１１に示す。各々の処理時間（ｔｉｊ）は、表３．４のとおりである。

（２）性能の定義

評価で用いる性能は、下記のごとく定義する。

応答時間：ユーザプログラムからの送信要求が発行された時点から、データ



（注）FEP2は、２台目を設置するとき、ステーションがもう１台必要になる。
図では、ステーションのハード量を基板２枚と仮定して示してある。

図３.８　コストパフォーマンスの比較

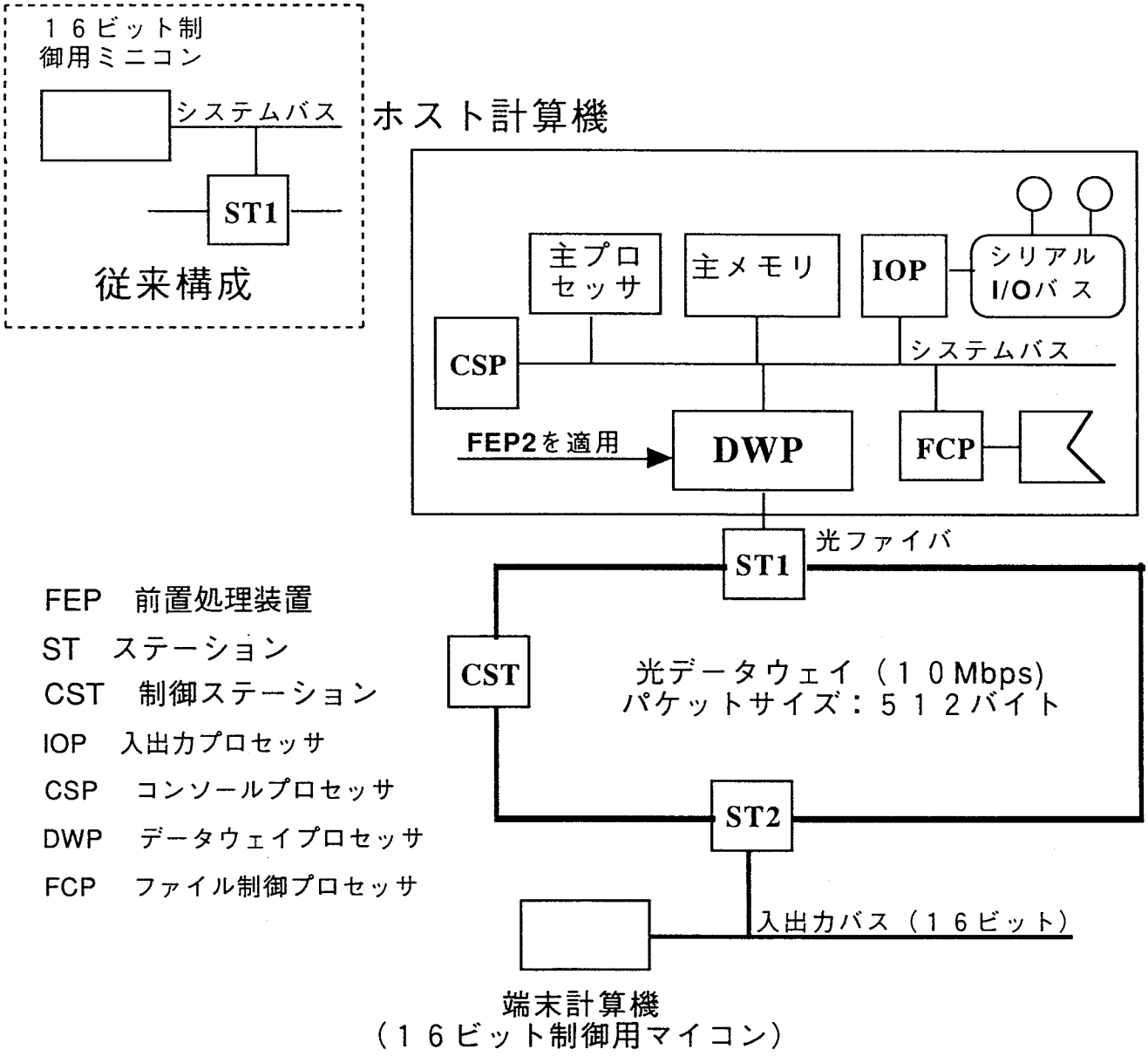
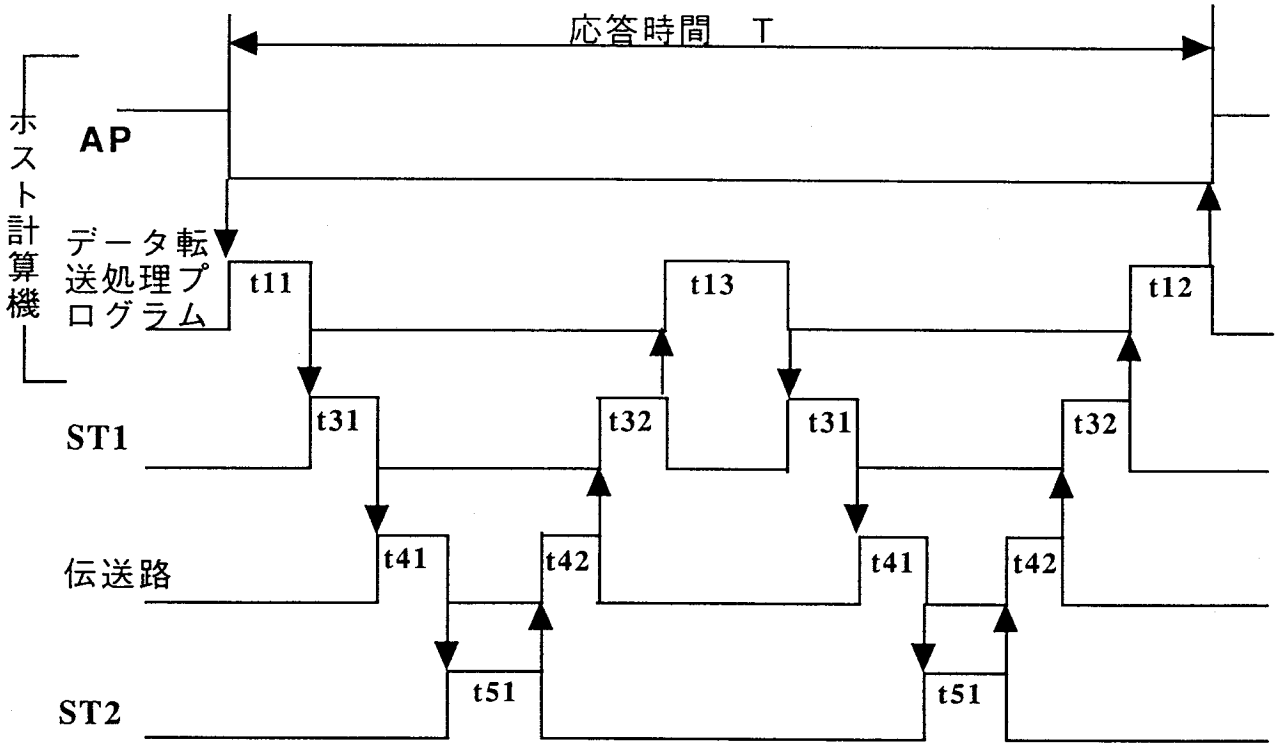


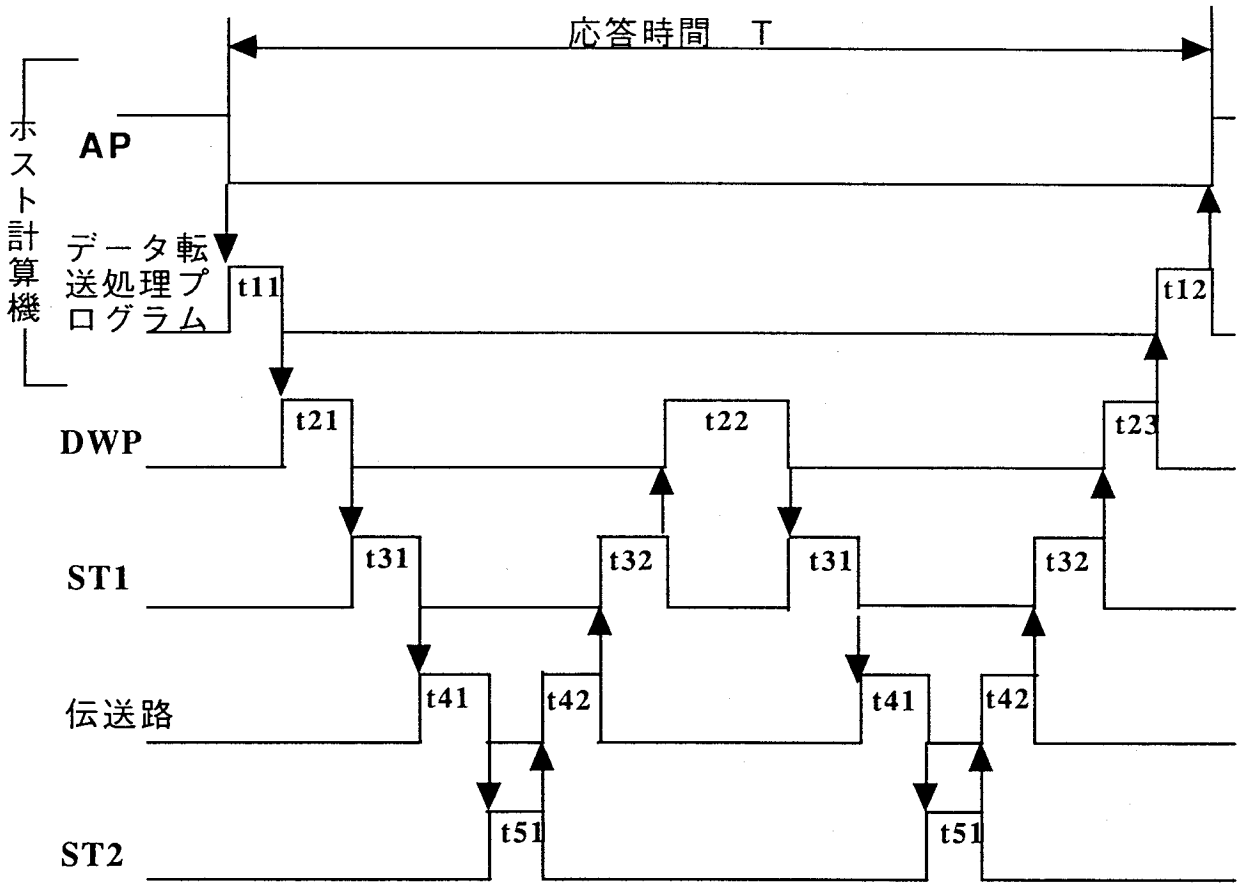
図３．９　適用評価システムの構成



tij　処理時間（詳細は、本文参照）

（２セグメントの例、セグメント数はデータ長により変わる）

図３．１０　適用評価モデルタイムチャート
（従来システム）



DWP　データウェイプロセッサ

tij　　処理時間（詳細は、本文参照）

（２セグメントの例、セグメント数はデータ長により変わる）

図３．１１　適用評価モデルタイムチャート
（FEP2適用システム）

表３．４　適用評価モデルで用いた処理時間

tij	従来システム	FEP Ⅱ適用システム	処理内容
t_{11}	１０	１．８	ユーザプログラムからの送信要求を受けステーション（従来システム）／DWP（FEP Ⅱ適用システム）を起動するまでの時間
t_{12}	６．９	１．８	ステーション／ＤＷＰからの終了通知を受け、ユーザプログラムを再起動するまでの時間
t_{13}	１５	－	従来システムにおいて、応答パケットを受け、次のセグメントをステーションに送信するまでの時間
t_{21}	－	４．８＊	ホスト計算機から起動され、最初のセグメントをステーションに送信するまでの時間
t_{22}	－	５．９＊	端末計算機からの応答を受けて、次のセグメントをステーションに送信するまでの時間
t_{23}	－	２．２	端末計算機からの最終セグメントに対する応答を受けて、ホスト計算機に終了を通知するまでの時間
t_{31}	２．２	２．２	１パケットの送信に要するステーションの処理時間
t_{32}	０	０	ステーションにおける応答パケット受信処理時間
t_{41}	１．０＊	１．０＊	１メッセージパケット（最大５１２バイト）の伝送時間
t_{42}	０	０	応答パケット（約６０バイト）の伝送時間
t_{51}	０	０	受信ステーション側でメッセージパケットを受信して、応答を返すまでの時間

＊データ長により異なる。この数値は、５１２バイトのときのもの。
各処理時間ｔij は、　１パケット（５１２バイト）の伝送時間（t_{41}）を単位として表している。

転送をすべて終了し、再びユーザプログラムに制御が戻るまでの時間（図３．９のＴ）

・データ転送スループット：単位時間に、ユーザプログラムからＤＷＰを経て、伝送路へ送出されるデータ量（送信要求は、複数発行されるものとする）

・プロセッサ利用率：主プロセッサの全稼働時間のうち、データ転送処理に費やされた時間の割合

（３）シミュレーションの条件

メッセージ送信要求は、ポアッソン到着とする。メッセージの長さは、平均２０００バイトの指数分布とする。相手計算機から応答が返ってくるまでの時間（t_{32}、t_{42}、t_{51}）は、０とする。

３．４．３　結果

表３．４の処理時間ｔｉｊを用いてＧＰＳＳによるシミュレーションを行い、図３．１２、図３．１３に示す結果を得た。スループットは伝送速度との比で表して、最大０．１４である（従来比約３倍）。応答時間は、メッセージ伝送時間（メッセージ長／伝送速度）との比で表して１０である（従来比約１／２）。主プロセッサの利用率は、１５％以下である。

３．５　結言

高速パケットネットワークによって複数の計算機を接続した計算機制御システムを対象に、計算機間のデータ転送スループット、応答時間、計算機負荷などの性能を向上する方式を検討し、コマンド多重処理形トランスポートプロセッサ（前置処理装置：ＦＥＰ）を提案した。提案したＦＥＰは、１台の高速マイクロコンピュータで多数のコマンドを多重に処理するので、少ないハードウェアで、高い性能を得ることが期待できる。

提案方式の効果は、３２ビットスーパミニコンと高速データウェイとの接続に適用し、シミュレーションにより評価した。評価の結果、提案したＦＥＰ構成方式は、下記の効果があることが明らかになった。

(1) ハード量は、基板２枚（第２章で提案したマルチプロセッサ方式の２／３）で済む

(2) 性能は、データウェイの伝送速度との比に換算して約０．１４（第２章で提案したマルチプロセッサ方式のプロセッサ２台のときの性能に相当）である

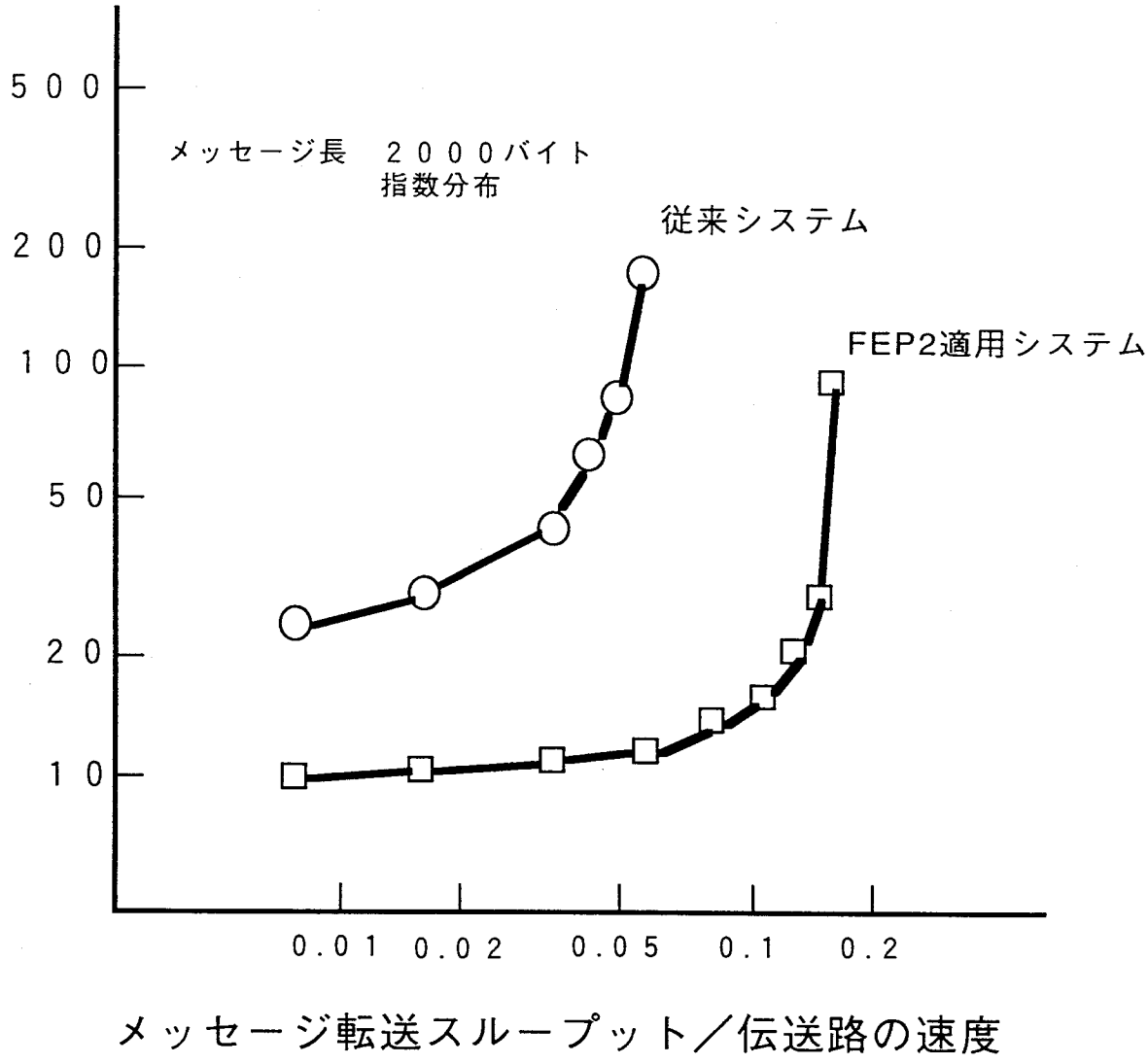


図3.12　メッセージ転送スループット
（シミュレーション）

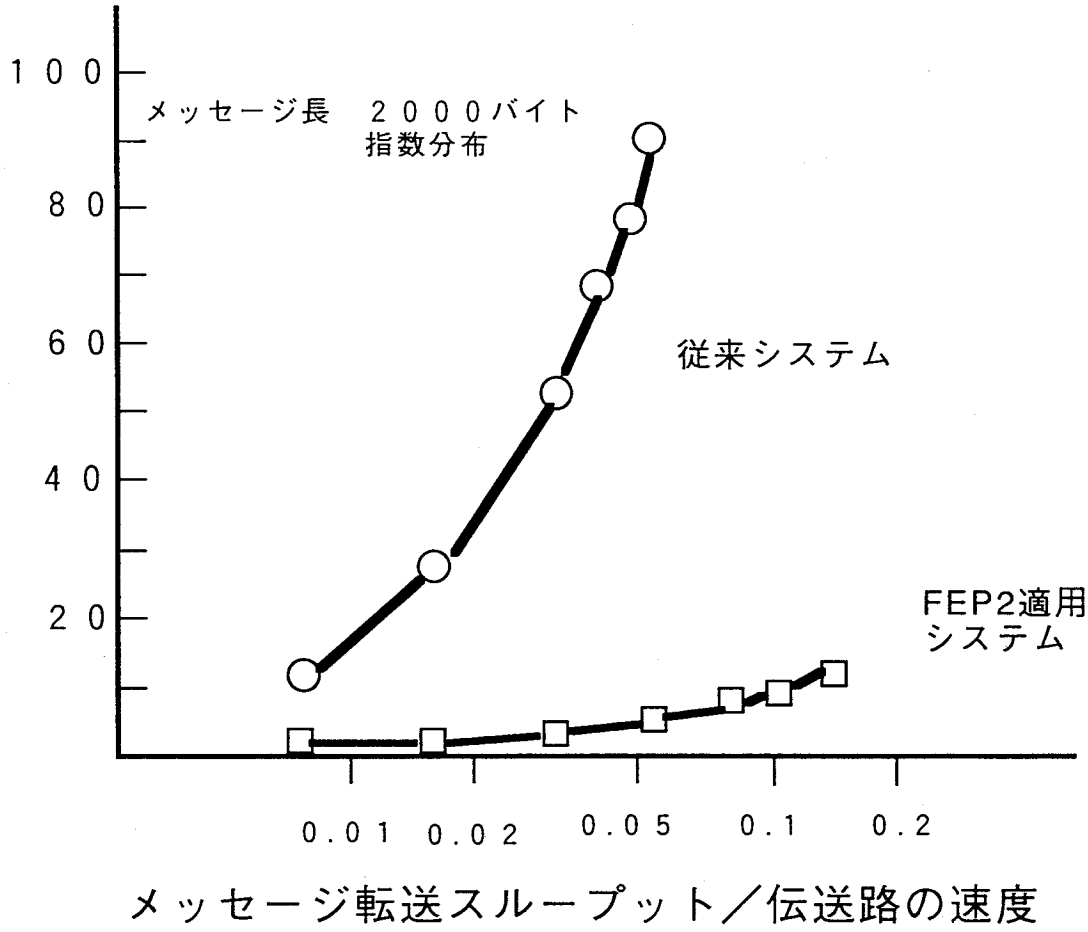


図3.13　主プロセッサ利用率（シミュレーション）

(3) ＦＥＰを適用した３２ビットスーパミニコンは、従来の１６ビット機種に比べ、平均２０００バイトのメッセージ転送性能において、スループットで約３倍、応答時間で約１／２の性能向上効果がある

第４章　プロトコル処理ハード化による高速化の提案

４．１　緒言

計算機におけるパケット単位の処理を高速化する方式として、専用通信プロセッサを計算機と伝送路との間に設ける方式を提案し、高速化の効果があることを前章までに述べてきた。近年、ＬＡＮの伝送速度は、光ファイバ通信技術の進展により、１００Ｍｂｐｓへと更に高速化される方向にある。また、プロトコルの標準化が進み、ＯＳＩが広く普及しようとしている。この結果、たとえばＬＡＮとして伝送速度１００ＭｂｐｓのＦＤＤＩ[9]を使い、プロトコルとしてＯＳＩ[40]を用いるようなオープンなネットワーク環境では、ＯＳＩが従来の制御専用プロトコルに比べて複雑なため、データ転送処理を通信プロセッサにオフロードするだけでは十分な性能を確保できない恐れがある。このため、従来通信プロセッサのプログラムによって行われていたデータ転送処理のうち、プロトコルに係る処理をハードウェアで実行することを提案する。

４．２　プロトコル処理高速化の課題

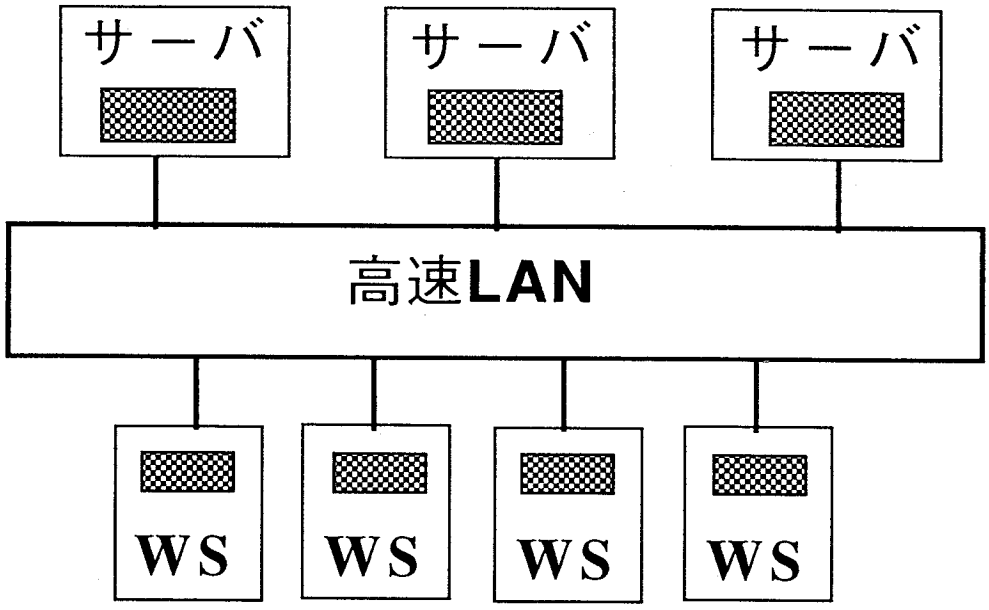
４．２．１　ＯＳＩプロトコル

ＯＳＩは、７層構造のアーキテクチャを持っている[40]。各層には、多くのプロトコル種別（コネクションオリエンテッド／コネクションレス）、プロトコルオプション／プロトコルクラスが定義されている。本章では、第２層から第４層のプロトコルを高速に実行することを考える。この理由は、図４．１に示すような分散ネットワークシステムにおいて、これらの層がＬＡＮアダプタで実行されるからである。プロトコルスタックもＬＡＮ環境を考慮し、第２層にＬＬＣクラス１［41］、第３層にＣＬＮＰ［42］、第４層にＴＰ４［43］［44］を選定している。

４．２．２　プロトコル処理高速化の課題

ＯＳＩの高速実行には、２つのアプローチがある。第１のアプローチは、ＲＩＳＣ技術に基づく高速マイクロプロセッサを用いる方法であり、第２のアプローチは、ＶＬＳＩ技術に基づく専用のプロトコル処理ハードウェアを開発する方法である［38］［39］。

前者は、多様なプロトコル／プロトコルオプションに対する柔軟性に利点が



LANアダプタ

WS　ワークステーション

図４.１　高速**LAN**を用いた分散システムの構成例

あるが、ハードウェア規模がＬＡＮアダプタに実装するには大きくなりすぎるという欠点がある。後者は、高速化と低コスト化に有利である反面、柔軟性に欠ける。本論文では、後者のアプローチを採る。その理由は、ＬＡＮアダプタに適用するには、ハードウェアサイズとコストが重要な要件であるからである。

ＯＳＩを実行する専用ハードウェアを開発するには、下記の問題を克服する必要がある。

(1) 各層のプロトコル処理が複雑

ＯＳＩのトランスポート層クラス４は、エンドツーエンドのトランスポートコネクションにおけるエラー回復責任を持っており、プロトコルが複雑になっているため、ハードウェアによる実行は困難である。多くのパラメータや付加機能について、コネクション設定時に通信する相手と取決めを行う必要がある。またデータ転送フェーズでは、ウィンドウフロー制御、シーケンス番号制御、セグメンティング／リアセンブリなどを実行する必要がある。ＯＳＩのプロトコルヘッダは、可変長でビット毎に意味付けされており、４バイトバウンダリに整頓されていないため、ハードウェアによる処理を困難にしている。

(2) 層間のインタフェースオーバヘッド

ＯＳＩは７層構造のため、層間のインタフェース制御オーバヘッド（たとえば、データの移動、パラメータの受渡し、プロトコルヘッダの受渡しなど）が発生する。したがって、多層構造のプロトコル処理に適した処理方式が必要である。

(3) タイマ制御が複雑

各トランスポートコネクションに３種類のタイマが使われる。したがって、同時処理可能なコネクション数を１２８～２５６に設定しようとすると、３８４～７６８のタイマ制御機構が必要になる。

４．２．３　高速化の方策

ＯＳＩのハード化にまつわる前記課題を克服するため、以下の方針を設定した。

(1) 正常データ転送処理のみ専用ハードウェア化

ＯＳＩの全階層を実行する専用ハードウェアができれば、高速化には有効であるが、実際には柔軟性に乏しく、ハード量が非現実的なほど大きくなると予

想される。そこで、本論文では下記の点を考慮して選定された機能のみをハード化することにした。

・プロトコル処理においては、正常データ転送処理の高速化ニーズがコネクションの設定／開放処理より強い。
・プロトコルにおけるプロファイルの違いや各層のプロトコルの特徴は、コネクション設定時やデータ転送の異常処理に多く現れる。
・伝送路への光ファイバの適用、ディジタル化などにより、伝送路のビット誤り率は、低減化される方向にある。つまり、データ転送フェーズでの異常処理の頻度は少なくなる傾向にある。

そこで、プロトコル処理を、①正常データ転送、②異常処理、③コネクション設定／開放処理、の３つの部分に分け、正常データ転送処理のみを専用ハードウェアで高速実行し、残りの部分は汎用のマイクロプロセッサで処理する。

(2) 各層対応に専用ハードウェアを設置

ＯＳＩの各層におけるクラス／オプションを組み合わせて７層全体でセットにしたものは機能標準と呼ばれる。層毎にプロトコル処理を実行する専用ハードウェアを開発することによって、多様な機能標準にも専用ハードウェアの組合せで柔軟に対応できるようにした。

(3) 多層プロトコルの処理を複数の専用ハードウェアでパイプライン処理

層間のインタフェースオーバヘッドを削減するため、ＦＩＦＯによるハードウェアインタフェース機構を導入する。このようにして、ＦＩＦＯで接続された各階層の専用ハードウェアがパイプラインで動作する。

これらの方策によって、下記の効果が期待できる。
・とりわけ高速化が要求される正常データ転送処理は、専用ハードウェアによって高速に処理できる。
・コネクション設定／開放制御や異常処理は、汎用マイクロプロセッサにより柔軟に処理できる。
・ハードウェア化を正常データ転送処理に限定することによって、専用ハードウェアの規模を削減でき、ＯＳＩの各層を１チップのＶＬＳＩにできる。

４．３　プロトコル処理ハード化方式の提案

４．３．１　要求条件

(1) 高速

ＯＳＩのような階層型の通信プロトコルを前提とすると、１００Ｍｂｐｓから１Ｇｂｐｓの伝送速度の向上に対応できる高速処理が求められる。

(2) コンパクトなハードウェア

ＬＡＮアダプタに適用できるようなコンパクトなハードウェアが必要である。

(3) 融通性

ＯＳＩの４層以下に適用することを考えると、各層に定義されているプロトコルクラスや各種のオプションに柔軟に対応できる融通性が求められる。

４．３．２　構成方式の検討

高性能プロトコル処理プロセッサの構成方式を検討する。

(1) 各層に対応する専用ハードウェア

各層のプロトコル処理は、独立に並行して処理できるので、各層対応に専用ハードウェア（ＦＥＵ：Front End Unit）を設けた。プロトコル処理をＦＥＵにより、並列処理するかパイプライン処理するかは、大きな方式上の選択になる。

・並列処理方式：１フレームの処理を複数のＦＥＵが同時に処理する方式である。下位の層でエラーが起きると、既に実行した他の層の処理をバックトラックする必要がある。

・パイプライン処理：１フレームの処理は、同時には１つのＦＥＵのみが処理する。フレームは、次々とＦＥＵをリレーされる方式である。下位層でエラーがあれば、そのフレームは次の層のＦＥＵにはリレーされない。

２つの方式とも、スループットに大きな差は無いと考えられる。１つのフレームに着目したときの遅延時間は、並列処理方式がパイプライン処理方式に比べて、１／ｎ（ｎはパイプラインの段数で、各段の処理時間が等しいとしたとき）にできる。本論文では、制御が簡単なパイプライン方式を採用した。

(2) バックエンドマイクロプロセッサ

専用ハードウェアはデータ転送の正常処理のみ実行するので、残りの処理（コククションの設定／開放、データ転送の異常処理など）は、汎用のマイクロプロセッサ（ＢＥＵ：Back End Unit）で処理する。高性能化する上で、送

受信フレームの分配方式が重要である。

・ＢＥＵで分配：一度フレームをＢＥＵで受けて分析し、正常データ転送処理でＦＥＵにより処理できると判定したもののみをＦＥＵに渡す方式。
・ＦＥＵで分配：まずＦＥＵにフレームを受け、自分で処理できないものをＢＥＵに渡す方式。

ＢＥＵは汎用マイクロプロセッサであるので、判定に時間がかかる。一方、ＦＥＵは、ハードウェアで構成するので判定処理が高速に行える。本論文では、ＦＥＵにより判定する方式を採用した。

(3) ユーザデータ転送処理とプロトコルヘッダ処理の分離

フレームは、一般にプロトコルヘッダ部とユーザデータ部とから構成されている。フレームは、一つの層から隣接する次の層へ渡される。層間のインタフェースオーバヘッドを削減することは、高速化にとって重要である。提案方式では、下記の方法を採った。

・各ユニットからアクセス可能な共通メモリを用意する。
・送受信フレームは、この共通メモリに一度だけバッファする。層間では、フレームを移動させない。
・各ユニットは、この共通メモリにアクセスすることによって、プロトコル処理を行う。
・ＦＥＵは、フレームのプロトコルヘッダ部のみを処理し、ユーザデータ部には関与しない。
・層間のインタフェースは、ＦＥＵ間に設けたＦＩＦＯ (First In First Out) 回路を経由して行う。
・ＦＥＵとＢＥＵの間もＦＩＦＯ回路で連絡する。

(4) バウンダリ制御回路

一般にプロトコルヘッダのフォーマットは、４バイト境界を意識して設計されていないため、共通メモリに蓄えられたフレームのプロトコルヘッダは、４バイト境界に整頓されていない。それゆえ、もし共通メモリにある情報に４バイト境界を越えてアクセスすると、余分のメモリサイクルを要する。プロトコルヘッダにアクセスするときの余分のメモリサイクルを不要にするため、バウンダリ制御回路を設けた。この回路を用いることによって、１～４バイトのデータであれば、４バイト境界に整頓されていなくても、１メモリサイクルでアクセスできる。

(5) プロトコルヘッダ事前準備

プロトコルヘッダの情報には、送信相手の識別子、順序番号、長さ、コネクション識別子、フロー制御などがある。このうち、トランスポートのコネクションが設定された後は、いつも固定情報で済むものも多い。そこで、トランスポートコネクション対応に送信バッファを割り当て、コネクション設定後これらの固定情報を予めバッファに固定的にセットしておく方式とした。こうすることによって、データ転送フェーズでは順序番号や長さなどフレーム毎に変化する情報だけセットすれば済み、高性能化できる。

以上の検討結果を踏まえて提案した高性能プロトコルプロセッサのハードウェア構成を図４．２に示す。

４．３．３　プロトタイプの開発

ＯＳＩの２層から４層を実行するための高性能プロトコルプロセッサのプロトタイプを開発した。このプロトタイプの仕様を表４．１に示す。このプロセッサの詳細なプロトコルサポート範囲を表４．２（ＣＬＮＰ）と表４．３（ＴＰ４）に示す。

この実験モデルでは、ＭＣ６８０２０（２５ＭＨｚ）をＢＥＵとして用いている。各ＦＥＵの内部構造を図４．３に示す。ＦＥＵは、ＡＬＵ、ＦＩＦＯ、ＤＭＡ回路、ヘッダの符号化／複合化回路、コネクションの状態を保持するためのレジスタファイルおよび高速シーケンサから構成している。この汎用シーケンサは、マイクロプログラム機能を持ち、５０ｎｓ／命令の高速で動作する。このマイクロプログラムの変更により、多様なプロトコルに対応できる。

４．３．４　ＬＡＮアダプタへの適用

図４．２は、提案しているプロトコルプロセッサのＬＡＮアダプタへの適用方法も示している。ＬＡＮアダプタは、プロトコルの１層から４層を実行する。アダプタは、ホストインタフェース、共通メモリ、ＭＡＣ－ＬＳＩ、ＭＣＵ（ＭＡＣ－ＬＳＩ制御ユニット）、プロトコルプロセッサから構成している。

伝送路からＭＡＣ－ＬＳＩを介して受信したフレームは、共通メモリにストアされ、プロトコルプロセッサが処理する。処理されたフレームは、ホストインタフェースを経由してホスト計算機の主メモリに転送される。送信の場合は、この逆に処理が行われる。

提案したプロトコルプロセッサは、ＭＣＵを変更することによって、容易に各種ＭＡＣ－ＬＳＩに対応することができる。

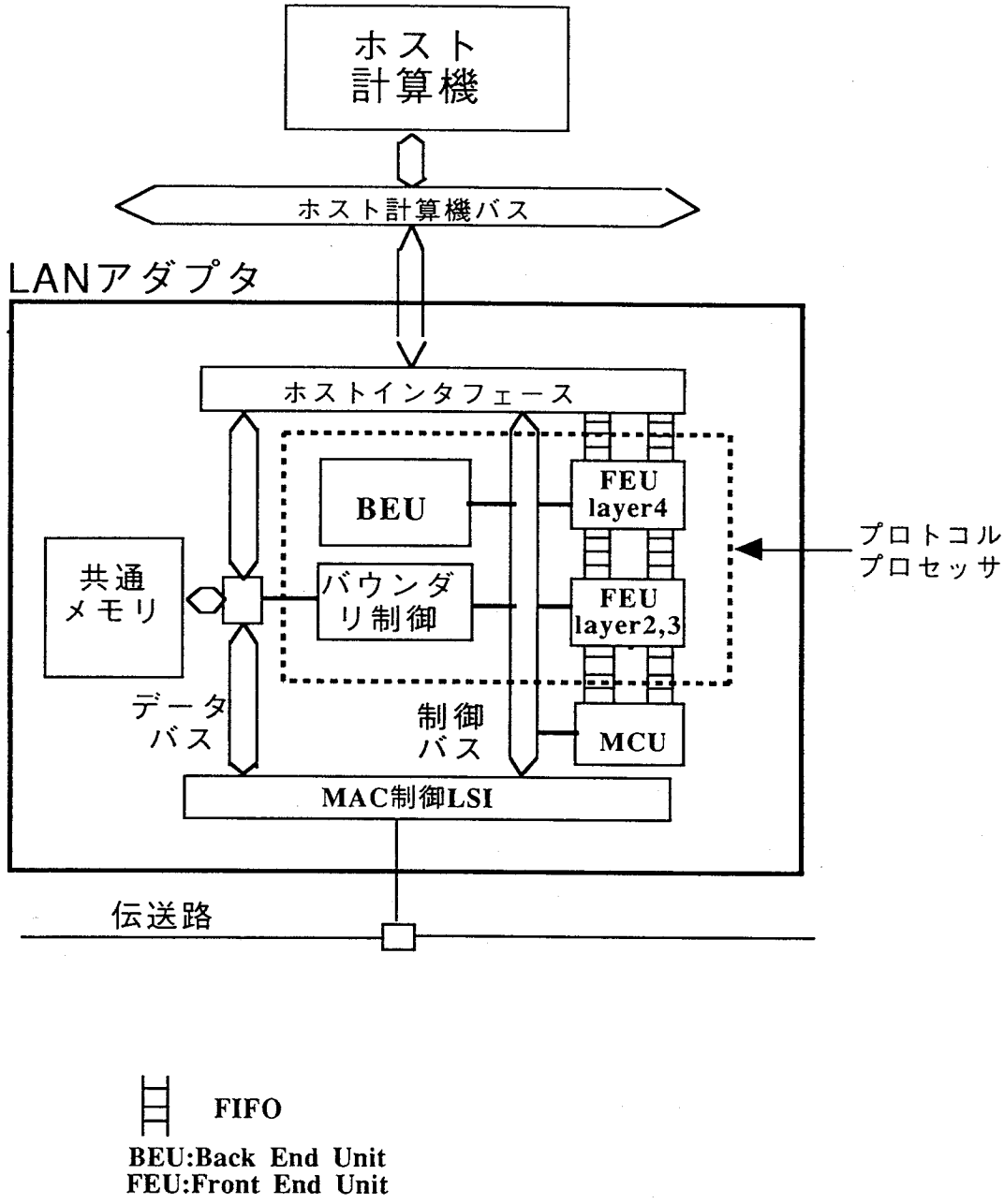


図4.2　高速プロトコルプロセッサの構成

表４．１　プロトタイプ高速プロトコルプロセッサの仕様

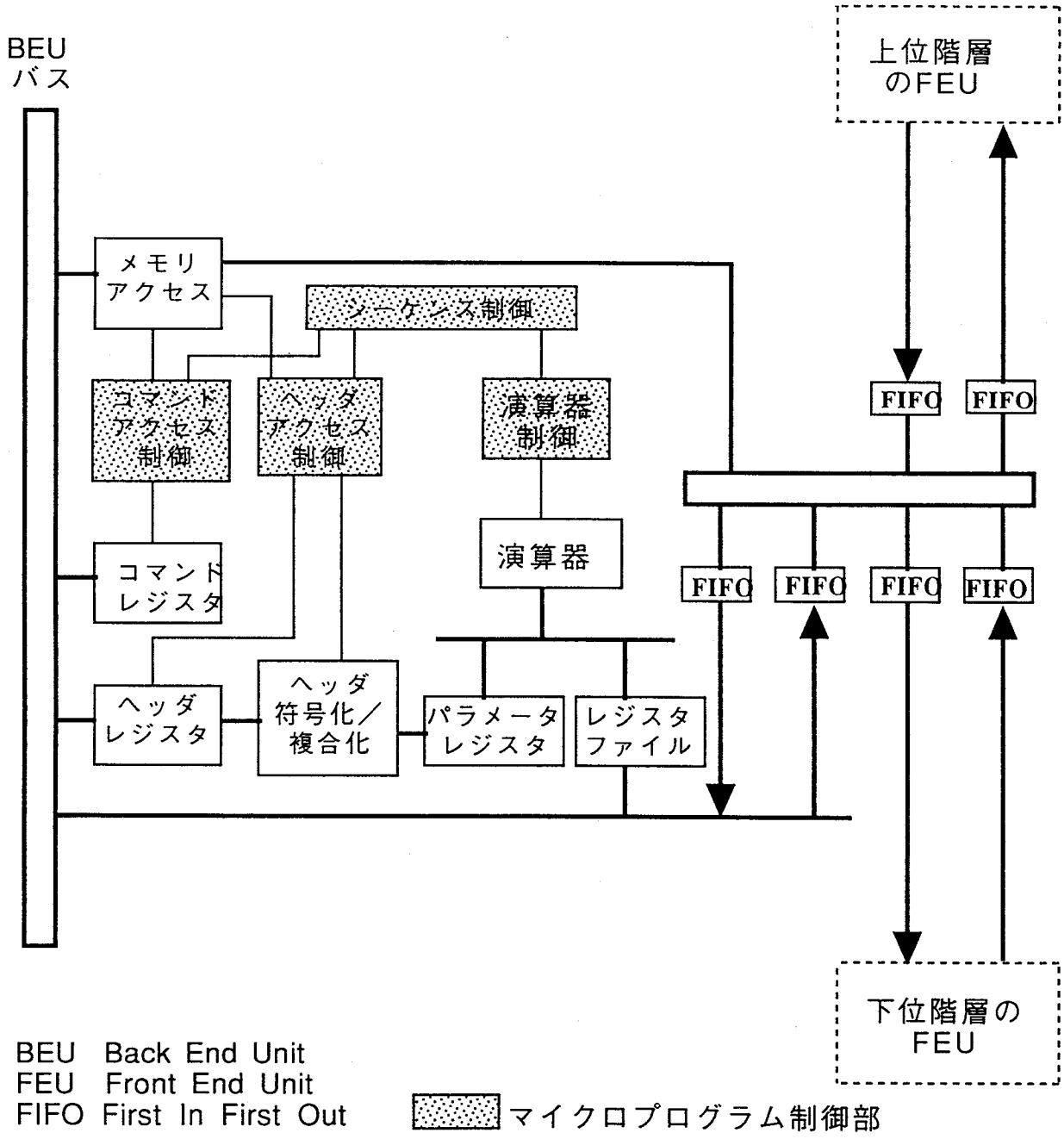
項　目		仕　様	備　考
プロトコル	第４層	ＩＳＯ８０７３	ＴＰ４
	第３層	ＩＳＯ８４７３	ＣＬＮＰ
	第２層	ＩＳＯ８８０２／２	ＬＬＣクラス１
ＢＥＵ (Back End Unit)		ＭＣ６８０２０	２５ＭＨｚ
共通メモリ		２５６ｋＢ	バッファメモリ
内部バス		ＩＥＥＥ１０１４	３２ビット
バウンダリ制御回路		4ﾊﾞｲﾄｱｸｾｽ/1ﾒﾓﾘｻｲｸﾙ	
ＦＥＵ	制御方式	マイクロプログラム	５１２ステップ
	シーケンサ	５０　ｎｓ／命令	
	演算器（ＡＬＵ）	１６ビット	
	層間インタフェース	ＦＩＦＯ	１０２４エントリ
	ＢＥＵインタフェース	ＦＩＦＯ	１０２４エントリ
	ＤＭＡ性能	４ＭＢ／ｓ	ヘッダの読み／書き
	ハードウェアサイズ	３００×３２０　ｍｍ	シングルボード

表４．２　プロトタイププロトコルプロセッサのＣＬＮＰサポート範囲

項番	項　　目	サポート
1	ＰＤＵの作成／分解	○
2	ヘッダフォーマットの分析	○
3	ＰＤＵの寿命制御機能	○
4	ＰＤＵのルート機能	×
5	ＰＤＵの分割／組立て	×
6	ＰＤＵの廃棄	○
7	エラー報告	○
8	ＰＤＵヘッダエラーの検出	○
9	パッディング	○
１０	セキュリティ	×
１１	ソースルーティング	×
１２	ルートの記録	×
１３	サービス品質の維持	×
１４	優先データ	×

表４．３　プロトタイププロトコルプロセッサのＴＰ４サポート範囲

項番	プロトコル機構	サポート
1	ネットワークコネクションへの割当て	×
2	ＴＰＤＵの転送	○
3	分割と組立て	×
4	連結と分離	×
5	コネクションの確立	○
6	コネクション確立の拒否	○
7	正常解放	○
8	ＴＰＤＵのﾄﾗﾝｽﾎﾟｰﾄｺﾈｸｼｮﾝへの関連付け	○
9	ＤＴ　ＴＰＤＵの番号付け	○
１０	優先データ転送	×
１１	ＴＰＤＵの確認までの保持	○
１２	多重化と逆多重化	×
１３	明示的フロー制御	○
１４	検査和（サムチェック）	×
１５	凍結レファランス	○
１６	タイムアウト時の再送	○
１７	再順序付け	×
１８	無活動監視制御	○
１９	プロトコル誤りの扱い	○
２０	分流と合流	×



BEU Back End Unit
FEU Front End Unit
FIFO First In First Out

マイクロプログラム制御部

図4.3 FEUの内部構造

４．４　実験的評価

４．４．１　評価の目的

このプロセッサで得られる処理速度の改善度合いを測定するため、第２層から第４層の処理時間を従来方式と実験的に比較した。さらに、この結果にもとづき、この高速プロトコルプロセッサをＬＡＮアダプタに適用した場合の処理時間を推定した。

４．４．２　評価の方法

実験的な高速プロセッサの性能は、次のように測定した。ＦＥＵに関する性能は実際に測定器により計測し、ＢＥＵのソフトウェアに関する性能はダイナミックステップを積算し、これに平均命令実行時間を乗ずることで求めた。ＭＡＣ層を実装しないで擬似ＭＡＣプログラムをＢＥＵ上に作成し、内部ループを形成して処理時間を測定した。

比較のために、正常プロトコル処理時間と擬似ＭＡＣプログラム処理時間を、ＭＣ６８０２０（２０ＭＨｚ）を用いたワークステーション上で測定した。比較に用いたデータ転送シーケンスを図４．４に示す。

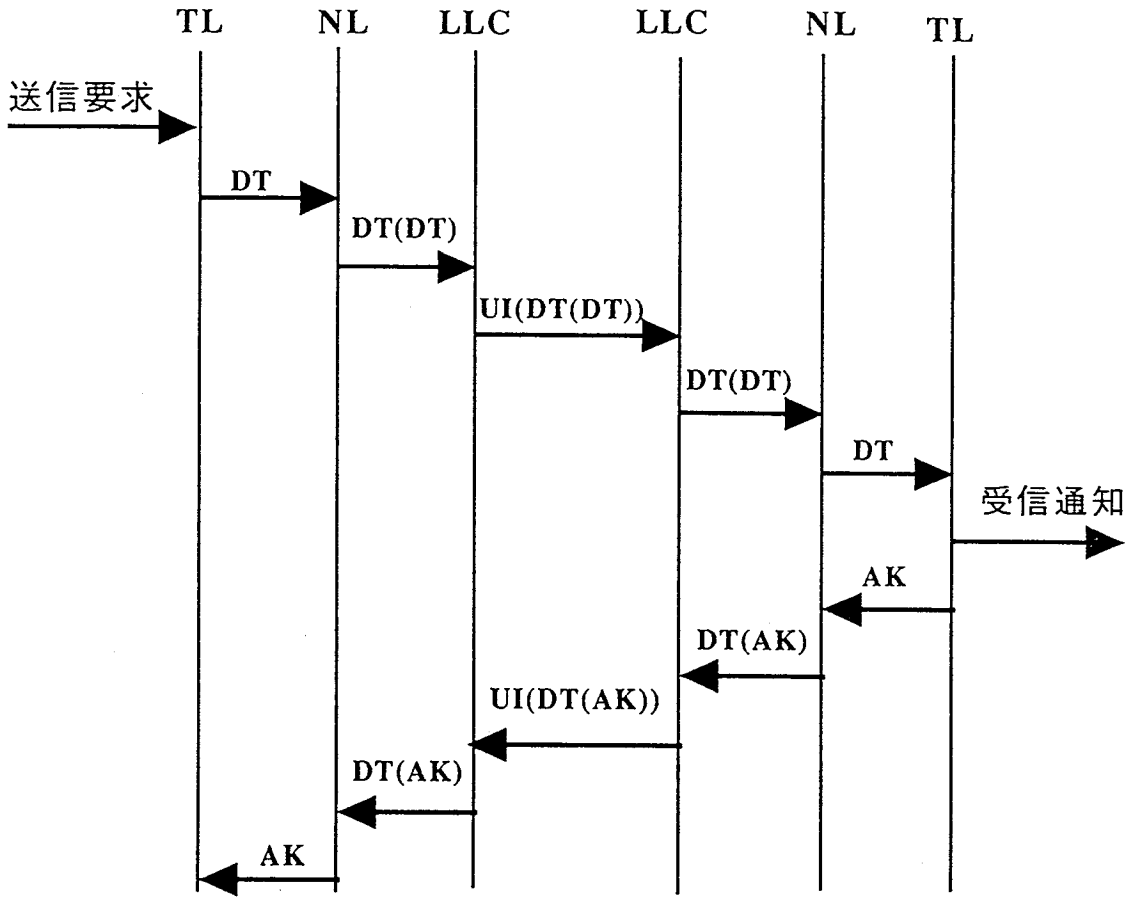
４．４．３　結果

比較結果を表４．４に示す。高速プロトコルプロセッサを適用した実験システムは従来方式による場合に比べ、正常データ転送において処理性能が７～１２倍向上した。

４．４．４　ＬＡＮアダプタへの適用効果

送信／受信に対するＬＡＮアダプタの総処理時間を各部の処理時間から計算した。得られた処理時間をＦＤＤＩ（１００Ｍｂｐｓ）上のフレーム伝送時間で正規化し、ＤＭＡスループット４、８、１６ＭＢ／ｓをパラメータとして表４．５（フレームサイズが４０００バイトの場合）と表４．６（フレームサイズが３００バイトの場合）に示した。

これらの結果は、ＬＡＮアダプタにおけるプロトコル処理に起因する性能上のボトルネックが提案した高速プロトコルプロセッサによって解消できることを示している。



LLC　論理リンク制御層
NL　ネットワーク層
TL　トランスポート層

図4.4　データ転送シーケンス

表４．４　プロトコル処理時間の比較

項 番	処理内容	従来方式	プロトタイプとの比
1	ＤＴフレームの送信	1	１／１４
2	ＡＫフレームの受信	1	１／１１
3	ＤＴ送信＆ＡＫ受信	1	１／７
4	送信側（項番１＋項番２）	1	１／１２
5	受信側（項番３）	1	１／７

表４．５　ＦＤＤＩ用ＬＡＮアダプタの推定性能
（フレームサイズ＝４０００バイト）
（処理時間は、フレーム伝送時間で正規化した値）

処 理 内 容	送 信 側			受 信 側		
ＤＭＡスループット　(バイト/秒)	４Ｍ	８Ｍ	１６Ｍ	４Ｍ	８Ｍ	１６Ｍ
プロトコルプロセッサ	０．６	０．６	０．６	０．５	０．５	０．５
ＭＡＣ制御ユニット（ＭＣＵ）	０．７	０．７	０．７	０．７	０．７	０．７
ホストインタフェース	３．８	２．２	１．４	４．１	２．５	１．８

送信側とは表４．４の項番４を、受信側とは表４．４の項番５の処理を意味する。

表４．６　ＦＤＤＩ用ＬＡＮアダプタの推定性能
（フレームサイズ＝３００バイト）
（処理時間は、フレーム伝送時間で正規化した値）

処 理 内 容	送 信 側			受 信 側		
ＤＭＡスループット　(バイト/秒)	４Ｍ	８Ｍ	１６Ｍ	４Ｍ	８Ｍ	１６Ｍ
プロトコルプロセッサ	８．０	８．０	８．０	７．１	７．１	７．１
ＭＡＣ制御ユニット（ＭＣＵ）	９．６	９．６	９．６	９．６	９．６	９．６
ホストインタフェース	１２	１０	９．５	１６	１５	１４

送信側とは表４．４の項番４を、受信側とは表４．４の項番５の処理を意味する。

４．５　結言

本論文では、ＬＡＮアダプタに適した高速ＯＳＩプロセッサを提案した。この提案の効果を定量的に評価するために、ＯＳＩの第２層から第４層のプロトコル処理を実行できるプロトタイプを作成し、その性能を測定した。

実験システムにおける正常データ転送のプロトコル処理時間は、従来方式による場合に比べて、送信処理で１／１２受信処理で１／７にできることを明らかにした。

これらの結果から、提案した高速プロトコルプロセッサを適用することによって、ＬＡＮアダプタの性能を改善できることを結論づけた。

第５章　結　　論

本論文は、機能階層形多重通信プロセッサによる分散ネットワークの高性能化について論じた。得られた結論を下記にまとめる。

高速パケット通信ネットワークによって複数の計算機を接続した分散ネットワークを対象に、計算機間のデータ転送スループット、応答時間、計算機負荷などの性能を向上する方式を検討し、通信プロセッサを第１層と第２層のプロトコルを実行するデータリンクプロセッサと第３層と第４層のプロトコルを実行するトランスポートプロセッサとから構成する機能階層形多重通信プロセッサ方式の提案と性能向上度の評価を行った。

提案した方式は、ホスト計算機と伝送路との間に、データ転送のための通信プロトコルを専用に処理する機能階層形多重通信プロセッサを置くことにより、従来は、ホスト計算機で実行していた通信プロトコル処理の大部分を通信プロセッサで高速に実行させるようにした点に特徴がある。この方式により、通信プロトコル処理の高速化とホスト計算機の負荷削減を図ることができる事を示した。

具体的には、階層化された通信プロトコルのうち、第１層から第４層までの処理を、２階層に分け、機能階層毎に高速ビットスライスマイクロプロッサ（語長３２ビットのマイクロ命令を２００ｎｓで実行する）でパイプライン処理する専用の多重通信プロセッサを開発してその効果を明らかにした。トランスポートプロセッサの構成方法としては、マルチプロセッサ・シングルコマンド処理方式とシングルプロセッサ・マルチコマンド処理方式の２つの方式を提案し検討した。

マルチプロセッサ・シングルコマンド処理方式については、マルチマイクロコンピュータ構成のトランスポートプロセッサ（前置処理装置：ＦＥＰ）とデータリンクプロセッサとから構成する機能階層形多重通信プロセッサ方式を提案した。提案方式による性能向上度を把握するため、マイコン２台から成るＦＥＰを試作し、伝送速度１０Ｍｂｐｓの高速パケット通信ネットワークに適用した。実験の結果、計算機間データ転送性能は、ＦＥＰが無いときに比べて、次のように改善されることを明らかにした。

(1) データ転送スループットが３倍になる（データ長が４ｋＢのとき）。
(2) 応答時間が１／２になる（データ長が６０Ｂのとき）。
(3) 計算機負荷が４０％（データ長が８Ｂのとき）～８０％（データ長が８ｋＢのとき）削減できる。

シングルプロセッサ・マルチコマンド処理方式については、ＦＥＰの構成方式としてコマンド多重処理形トランスポートプロセッサを提案した。提案したＦＥＰは、１台の高速マイクロコンピュータで多数のコマンドを多重に処理するので、少ないハードウェアで、高い性能を得ることが期待できる。提案方式の効果は、３２ビットスーパミニコンと高速データウェイとの接続に適用し、シミュレーションにより評価した。提案したＦＥＰ構成方式は、下記の効果があることを明らかにした。

(1) ハード量を、３０．５ｃｍ×４２．５ｃｍの基板２枚に収めることができる。

(2) 性能は、データウェイの伝送速度との比に換算して約０．１４である。

(3) ＦＥＰを適用した３２ビットスーパミニコンは、従来の１６ビット機種に比べ、平均２０００バイトのメッセージ転送性能において、スループットで約３倍、応答時間で約１／２の性能向上効果がある。

これらの結果より、機能階層形多重通信プロセッサ方式が通信プロトコル処理高速化の要請に応えうることを結論づけた。

提案方式は、計算機制御システムにおいて、高速パケット通信ネットワークとしてデータウェイを用いた場合について論じた。これ以外のシステムでも、高速パケット通信ネットワークと計算機との接続の場合には、提案した機能階層形多重通信プロセッサ方式がデータ転送性能の向上に役立つものと考えられる。

データウェイなどの高速パケット通信ネットワークの伝送速度は、光通信技術の進歩に伴い今後も更に高速化されるもの予想される。この結果、本研究で取り上げた分散ネットワークにおけるデータ転送処理高速化の要求は、ますます増大するものと考えられる。１００Ｍｂｐｓから１Ｇｂｐｓの高速パケット通信ネットワークが実用化される時代にあっては、これらの伝送速度に見合うパケット単位のプロトコル処理時間は、マイクロ秒オーダになる。従って、今後とも更に高速化の研究を続ける必要がある。

本研究では、独自のプロトコルでシステムを構築してきたが、その後プロトコルの標準化が進み、ＩＳＯで規定されたＯＳＩプロトコルが次第に普及してきている。また、本研究では、通信プロセッサに数枚のボードを要しているが、最近の半導体技術の進歩は、日進月歩であり、高度な機能を持った各種のＶＬＳＩが開発されるようになってきている。このような背景から、１００Ｍｂｐｓ以上の高速光ＬＡＮを用いた分散ネットワークにおいて、ＯＳＩプロトコル

をハードウェアによって高速に処理するＶＬＳＩの開発が、今後の重要な課題になると思われる。このような背景から本論文では、単にプロセッサの高性能化だけでなく、プロトコル処理のハード化を考える必要があることを指摘すると共に、プロトコル処理ハード化の課題を整理し、ＯＳＩのレイヤ２、レイヤ３、レイヤ４のプロトコルを対象に、プロトコル処理のハード化による高性能化方式を提案し、今後の研究の道標とした。

謝　　辞

本論文をまとめるにあたって本論文の細部にわたり検討を加え修正のご指導をいただいた大阪大学基礎工学部情報工学科宮原秀夫教授に心から感謝します。終始ご指導いただいた大阪大学基礎工学部情報工学科嵩忠雄教授、橋本昭洋教授、都倉信樹教授、岡山大学工学部情報工学科岡本卓爾教授に感謝の意を表します。

本研究を進めるに当たり、応用面からの貴重な助言をいただいた（株）日立製作所元大みか工場桑原洋部長（現在同社常務取締役）、平井浩二副部長（現在同社情報通信事業部副事業部長）、同社大みか工場井手寿之副工場長、伏見仁志部長、中西宏明部長、安元精一部長に深く感謝します。本研究の遂行を工場側から支援していただいた大みか工場の堀雄太郎主任技師、林慶治郎主任技師、岡田政和主任技師、溝河貞夫主任技師、元大みか工場の佐藤寛之氏（現在同社情報事業本部主任技師）に感謝します。

本研究の機会を与えていただいた（株）日立製作所システム開発研究所元所長三浦武雄博士（現在同社副社長）、前所長川崎淳博士（現在日立ソフトウェアエンジニアリング（株）専務）、堂免信義所長、大町一彦企画室長に感謝します。本研究遂行のきっかけを与えていただくと共に、その後の研究遂行、論文化、そして本論文の執筆にあたって終始親切なるご指導と督励をいただいた当研究所主管研究長三巻達夫博士に深く感謝します。また、研究途上において筆者の最初の学術論文を書く際に、懇切丁寧に御指導いただいた元当研究所主管研究員大成幹彦博士に感謝します。

本研究の遂行を直接ご指導いただいた当研究所主管研究員桎尾次郎博士、元当研究所平子叔男部長、元当研究所津田順二部長（現在同社システム開発本部副技師長）に心からお礼申し上げます。研究遂行を支援していただいた元当研究所の横田耕治氏、高木悟氏、関高明氏、黒木正彦氏、また当研究所第４部の福澤淳二研究員、松井進研究員、平田哲彦研究員、横山達也氏、水谷美加嬢、また学位論文申請にあたり事務処理面で支援していただいた木下由加里嬢に感謝の意を表します。

以上のほかにも、多くの方々の暖かい励ましと多大な御援助御協力をいただきました。これらの方々を含め、ここに深く感謝の意を表します。

参考文献

[1] 上谷　晃弘、「データハイウェイの現状と将来」：情報処理、Vol.18、No. 1、pp.76-87、1977

[2] Y.Matsumoto, O. Sasaki, T, Sumi: " A Distributed Processing System and Its Application to Industrial Contorol ", Proc. of National Computer Conference '78, pp.1273-1279 ,1978

[3] Jensen E. D. : " The Honeywell Exprimental Distributed Processor - An Overview ", IEEE COMUTER, 11, 1, pp.28-39 ,1978

[4] M. Takahashi, M. Yanaka, H. Fushimi, M. Maeda, J. Kashio, M. Mitsuoka: " Optical Fiber Data Freeway System - A Loop Network for Distributed Computer Control ", COMPCON'81, Spring, pp.458-463, 1981

[5] David D. Clark, Kenneth T. Pogran, David P. Reed: " An Introduction to Local Area Networks ", Proceedings of The IEEE, 66, 11, pp.1497-1516, 1978

[6] 石坂　充弘、中塚　茂雄、覚埜　高音、市橋　立機、松永　宏：「高速ループシステム（ＬＯＯＰ－３）」、情報処理学会第２１回全国大会予講集、pp.662-656、1980

[7] Robert M. Metcalfe, David R. Boggs: " Ethernet: Distributed Packet Switching for Local Computer Networks", CACM, Vol. 19, No. 7, pp. 395-404, 1976

[8] M.Stark, A, Kornhauser, and D. Van-Mierop: " A High Functionality VLSI LAN Controller for CSMA/CD Network ", IEEE COMPCON'83 Spring, pp.510-517, 1983

[9] Floyd E. Ross: " An Overview of FDDI: The Fiber Distributed Data Interface " , IEEE Journal on Selected Areas in Communications, Vol. 7, No. 7, pp. 1043-1051, Sept. 1989

[10] 八木　驍：「通信制御の動向」、情報処理、Vol. 20、No. 1、pp.3-21、1979

[11] 橋本　昭洋、山下　正秀：「通信制御プロセッサ」、電子通信学会誌、Vol. 62、No. 11、pp.1296-1303、1979

[12] F. E. Heart, S. M. Ornstein, W. R. Crowther, W. B. Barker: " A New Minicomputer/Multiprocessor for the ARPA Network ", Proc. of National Computer Conference '73, pp.529-537, 1973

[13] 福原　美三、小山　謙二：「マルチプロセッサ構成による機能階層形通信プロセッサの検討」、昭和５５年度電子通信学会総合全国大会、論文

番号1414、1980
[14] 森田　和夫、桑原　洋、井手　寿之：「制御用データ伝送システムと周辺装置」、電気学会雑誌、Vol. 96、No. 8、pp.699-704, 1976
[15] 桑原　洋、平子　叔男：「制御用コンピュータのネットワークシステム」電気学会雑誌、Vol. 98、No. 3、pp.199-203, 1978
[16] 樫尾　次郎、寺田　松昭、黒木　正彦、平井　浩二、伏見　仁志、今井　真澄：「ＨＩＤＩＣ　８０ネットワークシステム（ＤＰＣＳ）の開発」、情報処理学会第１７回全国大会、論文番号260、1976
[17] 桑原　洋、平井　浩二、平子　叔男、寺田　松昭：「分散形制御用計算機システム」、昭和５３年電気四学会連合大会、論文番号212、pp.647-650、1978
[18] 平子　叔男、平井　浩二、寺田　松昭：「制御用分散処理システム」、情報処理、Vol.20、No.4、pp.346-349, 1979
[19] 三巻　達夫、寺田　松昭：「インハウスネットワーク」、計測と制御、Vol.19、No.1、pp.103-109, 1980
[20] M.Terada, J.Kashio, K.Yokota, Y.Hori, H.Fushimi: " A Network Operating System for High Speed Optical Fiber Loop Transmission System", ICCC'80, pp.641-646, 1980
[21] 寺田　松昭、関　高明、樫尾　次郎、堀　雄太郎：「高速パケット伝送路を対象とした通信プロセッサの検討」、情報処理学会第２３回（昭和５６年後期）全国大会、論文番号 5D-7、1981
[22] 寺田　松昭、関　高明、樫尾　次郎、堀　雄太郎：「高速パケット伝送路用前置処理装置の一構成法」、情報処理学会論文誌、Vol. 23、No.6、pp.707-715, 1982
[23] 寺田　松昭、高木　悟、中西　宏明、伏見　仁志：「ローカルネットワークにおけるコンピュータインタフェース高速化の検討」、情報処理学会第２６回（昭和５８年前期）全国大会、論文番号 5G-8、1983
[24] 寺田　松昭、高木　悟、安元　精一、中西　宏明：「機能階層形通信プロセッサによる計算機間メッセージ転送スループットの向上方式について」、情報処理学会「ローカルエリアネットワーク」シンポジウム、pp. 163-170、1983
[25] 福澤　淳二、寺田　松昭、高木　悟、溝河　貞夫、末木　雅夫：「マルチ仮想回線動的チャネル割当方式による高性能ＬＡＮアダプタの一方式」「ＬＡＮ／マルチメディアの応用と分散処理」シンポジウム、pp.69-76、1984
[26] 寺田　松昭、高木　悟、樫尾　次郎、安元　精一、伏見　志、中西　宏明：「高速パケット伝送路用コマンド多重処理形前置処理装置」、情報処理学会論文誌、Vol. 26、No. 2、pp.219-227, 1985

[27] 水谷　美加、松井　進、寺田　松昭：「プロトコル処理高速化方式の提案とＯＳＩプロトコルへの適用」、情報処理学会第４０回（平成２年前期）全国大会、論文番号５Ｎ－３、１９９０

[28] 横山　達也、松井　進、平田　哲彦、水谷　美加、寺田　松昭：「通信プロトコル高速処理プロセッサの方式提案」、情報処理学会第４１回（平成２年後期）全国大会、論文番号４Ｑ－７、１９９０

[29] T.Hirata, S.Matsui, T.Yokoyama, M.Mizutani, M.Terada: " A High Speed Protocol Processor to Boost Gateway Performance", Proc. of IEEE Globecom'90, PP.1426-1430, 1990

[30] 松井　進、平田　哲彦、横山　達也、水谷　美加、寺田　松昭：「専用ハードウェア化による通信プロトコル処理高速化の一方式」、情報処理学会論文誌、Vol. 32、No.2、 pp.272-279, 1991

[31] 平田　哲彦、横山　達也、水谷　美加、寺田　松昭：「高速ＬＡＮ用通信制御装置の一構成法」、電子情報通信学会　情報ネットワーク研究会、IN90-98、 pp.13-18、 1991

[32] M.Terada, T.Yokoyama, T.Hirata, S.Matsui: " A High Speed Protocol Processor to Execute OSI", Proc. of IEEE INFOCOM'91, pp.944-949, 1991

[33] 横山　達也、平田　哲彦、水谷　美加、寺田　松昭：「プロトコル処理専用プロセッサのアーキテクチャに関する一考察」、情報処理学会第４２回（平成３年前期）全国大会、論文番号５Ｔ－６、１９９１

[34] 平田　哲彦、横山　達也、水谷　美加、寺田　松昭：「高速ＬＡＮ通信制御装置の構成法に関する一考察」、情報処理学会第４２回（平成３年前期）全国大会、論文番号５Ｔ－７、１９９１

[35] 水谷　美加、平田　哲彦、横山　達也、寺田　松昭：「高速ＬＡＮ通信制御装置の性能に関する一考察」、情報処理学会第４２回（平成３年前期）全国大会、論文番号５Ｔ－８、１９９１

[36] Liba Svobodova: " Measured Performance of Transport Service in LANs", Computer Networks and ISDN Systems, Vol. 18, pp.31-45, 1989/90

[37] David R. Cheriton and Carey L. Williamson:" VMTP as the Transport Layer for High-Performance Distributed Systems " , IEEE Communications Magazine, pp.37-44, June 1989

[38] Greg Chesson: " Protocol Engine Design", USENIX '87, pp.313-319, 1987

[39] H. Ichikawa, H. Yamada, T. Akaike, S. Kanno, and M, Aoki: " Protocol Control VLSI for Broadband Packet Communications ", Globecom '88, pp.1494-1498, 1988

[40] ISO8072, Information Processing Systems - Open Systems Interconnection - Basic Reference Model, 1984

[41] ISO8802-2, Information Processing Systems - Local Area Networks - Part 2: Logical Link Control, 1987

[42] ISO8473, Information Processing Systems, Open Systems Interconnection, Data Communications Protocol for Providing Connection less - Mode Network Service ,1986

[43] ISO8072, Information Processing Systems - Open Systems Interconnection -Connection Oriented Transport Service Definition, 1986

[44] ISO8073, Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification, 1986

[45] Bernd W. Meister: "A Performance Study of ISO Transport Protocol", IEEE Trans. on Computers, Vol. 40, No. 3, pp.253-262, 1991

[46] 斎藤　忠夫、白畑　厚志：「多層構造プロトコルをもつネットワークの伝送効率の評価」、信学論(D-I), Vol. J72-D-I, No. 7, pp.562-569, 1989

[47] 稲井　寛、横平　徳美、村田　正幸、宮原　秀夫：「階層型通信プロトコルの性能評価」、電子情報通信学会論文誌 B-I、Vol. J73-B-I、No.5、pp.505-515、1990

[48] 斎藤　忠夫、福井　俊之：「多層構造プロトコルをもつネットワークの伝送効率の評価式」、電子情報通信学会、論文誌 D-I、Vol. J73-D-I、4、pp.424-431、1990

[49] 平井　浩二、林　利弘、神内　俊郎、寺田　松昭：「ＨＩＤＩＣ　８０マルチシステムをサポートするソフトウェア」、昭和５１年電気学会全国大会、論文番号１１４８、１９７６

[50] 横田　耕治、寺田　松昭、堀　雄太郎：「ＨＩＤＩＣ　８０ネットワーク・システム（ＤＰＣＳ）におけるリソース排他制御システムについて」昭和５４年電気学会全国大会、論文番号１１８４、１９７９

[51] 寺田　松昭、堀　雄太郎、高木　悟：「異機種ネットワークに関する一考察ーーＨＩＤＩＣ　８０シリーズによるＴＳＳ端末エミュレーターー」、昭和５４年度　情報処理学会第２０回全国大会、論文番号３Ｇ－１、１９７９

[52] 寺田　松昭、中西　宏明、樫尾　次郎、伏見　仁：「ＨＩＤＩＣ　シリーズ　ネットワークアーキテクチャ」、昭和５７年電気学会全国大会、論文番号１２４４、１９８２

[53] 寺田　松昭、樫尾　次郎、安元　精一、中西　宏明：「低コストローカ

ルエリアネットワーク（μ－ΣＮＥＴ）の設計思想」、情報処理学会第２８回（昭和５９年前期）全国大会、論文番号１Ｄ－５、１９８４
[54] 寺田　松昭、福澤　淳二、高木　悟、溝河　貞生、大貫　健：「低コストローカルエリアネットワーク（μ－ΣＮＥＴ）用通信制御ボード（μＮＣＰ）の開発」、情報処理学会第２８回（昭和５９年前期）全国大会、論文番号１Ｄ－６、１９８４
[55] 寺田　松昭：「ブロードバンドＬＡＮにおける双方向ＣＡＴＶ」、情報処理学会　ニューメディア／マルチメディアと分散処理－現状と将来の課題－シンポジウム、pp. 13-19、1985
[56] 松井　進、横山　達也、柳生　和男、寺田　松昭：「通信制御アダプタの性能に関する一考察」、情報処理学会　マルチメディア通信と分散処理研究会、27-5、pp. 1-8、1985
[57] 高橋　康弘、高田　治、寺田　松昭、村橋　誠一、檜山　邦夫：「ＬＳＩ化Σネットワークの開発（１）－システム構成－」、情報処理学会第３１回（昭和６０年後期）全国大会、論文番号1Q-7、1985
[58] 高橋　康弘、高田　治、檜山　邦夫、寺田　松昭：多元情報光ＬＡＮにおけるハイブリッド交換アークテクチャに関する一考察、電子通信学会情報ネットワーク研究会、ＩＮ８５－１１５、１９８５
[59] 寺田　松昭、、伏見　仁、安元　精一、高橋　正弘、古寺　博：「超高速光ＬＡＮを応用した分散処理システムの制御方式」、情報処理学会第３０回（昭和６０年前期）全国大会、論文番号１Ｐ－６、１９８５
[60] 寺田　松昭、柳生　和男、平田　哲彦、森藤　素良：「マルチメディアブロードバンドＬＡＮの方式提案」、情報処理学会第３２回（昭和６１年前期）全国大会、論文番号４Ｄ－３、１９８６
[61] 柳生　和男、平田　哲彦、寺田　松昭、西沢　隆彦：「マルチメディアブロードバンドＬＡＮのデータ通信方式」、情報処理学会第３２回（昭和６１年前期）全国大会、論文番号７Ｄ－１、１９８６
[62] 福澤　淳二、寺田　松昭、林　慶治郎：「ＬＡＮベース分散処理用ネットワークソフトウェアの一考察」、情報処理学会第３３回（昭和６１年後期）全国大会、論文番号３Ｕ－４、１９８６
[63] 松井　進、柳生　和男、寺田　松昭：「ＬＡＮ－ＷＡＮ相互接続網におけるネットワークレイヤプロトコルの提案」、情報処理学会第３３回（昭和６１年後期）全国大会、論文番号４Ｕ－１１、１９８６
[64] 木名瀬　敏彰、寺田　松昭、大塩　剛至、岡田　重雄：「パソコン通信用ゲートウェイに関する一考察」、情報処理学会第３４回（昭和６２年前期）全国大会、論文番号１Ｙ－８、１９８７
[65] 寺田　松昭：「音声・データ統合ＬＡＮの応用サービス」、昭和６２年電気・情報関連学会連合大会、論文番号３３－４、１９８７

[66] 寺田　松昭、柳生　和男、森藤　素良：「マルチメディアブロードバンドＬＡＮ技術のマルチメディアサービスシステムへの適用方式」、情報処理学会第３４回（昭和６２年前期）全国大会、論文番号5Y-2、1987

[67] 樫尾　次郎、寺田　松昭、星　徹：「地域、構内ネットワークの展望、電子情報通信学会誌」、Vol. 70、No. 11、pp.1103-1111, 1987

[68] 寺田　松昭：「ＬＡＮ、ＭＡＮへのＡＴＭの応用」、昭和６３年電気・情報関連学会連合大会、論文番号２２－６、１９８８

[69] 山内　雪路、滝安　美弘、田中　捷樹、寺田松昭：「ＡＴＭ技術を用いた高速基幹ＬＡＮの検討」、電子情報通信学会　情報ネットワーク研究会、ＩＮ８８－１２１、pp. 25-29、1989

[70] 樫尾　次郎、寺田　松昭、天田　栄一：「広帯域ＬＡＮの技術動向と国際標準化」、電子情報通信学会論文誌、Vol. J72-B-I、No. 11、PP.896-903、1989

[71] 福澤　淳二、寺田　松昭、溝河　貞生：「高性能ＭＡＣブリッジ構成構成方式の提案と評価」、情報処理学会論文誌、Vol.31、No.8、pp.1260-1267, 1990

[72] 塚越　雅人、高田　治、寺田　松昭：「ＬＡＮ間ブリッジ接続に関する一考察」、情報処理学会第４０回（平成２年前期）全国大会、論文番号３Ｎ－８、１９９０

[73] O.Takada, M.Tsukakoshi, M.Terada, M.Yamaga: " An FDDI Bridge for the Super Backbone LAN", Proc. of Localnet'90, pp.399-404, 1990

[74] 重差　秀彦、滝安　美弘、天田　栄一、寺田　松昭、畠山　靖彦、山鹿光弘：マルチメディア高速基幹ＬＡＮ　Σ－６００の開発」、電子情報通信学会春季全国大会、論文番号 B-638、1991

[75] 塚越　雅人、高田　治、寺田　松昭、山鹿　光弘：「マルチメディア高速基幹ＬＡＮ　Σ－６００のＦＤＤＩブリッジ方式」、電子情報通信学会春季全国大会、論文番号 B-640、1991

[76] E. Amada, T. Morita, M. Terada, M. Yamaga, M. Miyamoto: "Architecture and technology for Next Generation Multimedia Private Networks ", Proc. of 6th Telecom'91 Forum, 1991

[77] 平田　哲彦、横山　達也、水谷　美加、寺田　松昭、三巻　達夫：「プロトコル高速処理装置により高性能化をはかったＬＡＮ用通信制御装置の一構成法」、情報処理学会論文誌、Vol.33、No.2、平成４年２月（掲載決定）

[78] M.Tsukakoshi, O. Takada, T. Murakami, M.Terada, M. Yamaga:" Large-scale and High-speed Interconnection of Multiple FDDIs using ATM-based Backbone LAN" , Proc. of IEEE INFOCOM'92 (to appear)