

100 Mb/s 光トークンリングを用いたプロセッサ間結合システム†

星 子 隆 幸^{††} 魚 住 栄 市^{††} 小 柳 津 育 郎^{††}

複数のプロセッサ、記憶装置、周辺装置等との高速通信を行うネットワークへの適用をねらいとした米国規格協会 (ANSI) の高速 LAN の新基準が提案されているが、筆者らはこの高速 LAN と類似の適用領域を持ち、100メガビット/秒 (Mb/s) の伝送速度を有する光トークンリング方式のネットワークを導入したプロセッサ間結合システムの検討を進め、このほどネットワークノードを構成するプロセッサ間結合装置 (PCU) を開発した。本装置は、プロセッサをチャンネルを介してリングに接続し、HDLC 手順を完全群接続——ネットワーク内の任意のノード間の接続——系向きに拡張したデータリンク制御手順を用いてリング上の任意のプロセッサとのデータ転送制御を行う。本論文では、高速のプロセッサ間通信に適した PCU の機能・構成条件を明らかにする。PCU にこの機能・構成条件を適用することにより、完全群接続系で一般に採用されている通信方式と比べて、プロセッサ間通信制御に要するソフトウェア処理を 1/2 以下に軽減できる。また、メッセージ通過時間を IEEE 802 トークンリング方式と比較して、特に高トラヒック、長リングになるほど大幅に短縮できる。さらに、システム復旧の高速化に有効なリングの障害箇所検出・切離しの自動化アルゴリズムを提案する。

1. ま え が き

近年、計算機による処理量増大、システムの段階的成長、高信頼化等に対する要望を多数台の疎結合マルチプロセッサ構成による負荷分散で解決し、ソフトウェアの開発費削減・生産性向上さらにはシステム処理能力向上に対する要望をプロセッサの分散化 (機能分散) により解決していく傾向が顕著である。

機能分散化の例としては、

- ・マルチメディアに対応するための通信処理の専用プロセッサへの分離。
 - ・アーキテクチャの異なる計算機上に構築されたソフトウェアやデータベースをジョブ転送またはファイル転送により有効利用するための異機種間接続。
- などのシステム構成があげられる。

一方、計算機の所要処理能力や記憶データおよび処理データ量の飛躍的な増大等により、複数の計算機からなるシステムを高速のネットワーク接続構成とする需要も顕在化してきている¹⁾。

本論文は、同一ビル内または隣接ビル間に分散設置された数十台程度のプロセッサを完全群接続し、任意のプロセッサ間で高速通信を行うシステムを主対象として、該システムにおける高速性・経済性に優れたプ

ロセッサ間結合方式並びにプロセッサ間結合装置 (以下 PCU と称す) の機能・構成条件について提案するものである。まず次章でシステム適用領域の明確化および基本的プロセッサ間結合方式とその選定理由の明確化を行う。次に 3 章で高速性に優れたプロセッサ間通信実現のためのソフトウェアインタフェース並びにデータリンク制御機能の提案と高速化の定量的評価を行う。最後に 4 章で伝送路系障害検出時、自動的に障害箇所の切離し・システム復旧が可能な障害処理方式を提案する。

2. システム構成条件

本論文で対象とするシステム構築にあたって要求される条件として、

- (1) データ通信サービスを行うセンタシステムの処理能力拡大に柔軟に対応可能なシステム拡張性
- (2) 通信処理に要するソフトウェアオーバーヘッドが少なく、ハードウェアの転送能力向上を経済的に実現可能な通信制御インタフェース
- (3) システムの長時間連続運転の実現とシステム障害時の高速リカバリ等があげられる。

これらの要求条件を満足する PCU の方式構成条件とその理由/背景を表 1 に示す²⁾⁻⁷⁾。本表に示す方式構成条件より、図 1 に示すシステム構成を対象に PCU の具備すべき機能条件の明確化とその評価を行う。

† Processor to Processor Communication System Using a 100 Mb/s Optical Token Ring by TAKAYUKI HOSHIKO, EIICHI UOZUMI and IKURO OYAIZU (Processing Equipment Section, NTT Electrical Communication Laboratories).

†† NTT 電気通信研究所処理装置研究室

表 1 PCU 方式構成条件
Table 1 Design concept of PCU.

項目	諸元	理由 / 背景
プロセッサ間結合方式	チャンネル結合	①チャンネル結合は通信オーバーヘッドが小さく、高速で標準的なチャンネルインタフェース (I/O インタフェース) が提供される。 ②回線インタフェースは通信オーバーヘッドが大きく、高速性に欠ける。 ③メモリ結合は性能レンジやアーキテクチャの異なる多数のプロセッサ間接続が困難で、柔軟なシステム拡張性に欠ける。
伝送方式/伝送路	光ファイバケーブル (ビットシリアル)	所要伝送速度 100 Mb/s と PCU-PCU 間接続距離 1km 以上の設計目標を経済的に実現可能。
ネットワークポロジ	リング	100 Mb/s 程度の高速伝送を数十台の PCU 間で光を用いて信頼性良くかつ経済的に実現するにはバス、スターは不向き。
アクセス方式	トークンパッシング	①トークンパッシングは高速、高信頼度で拡張性に富む。 ②コスト、拡張性の点で周波数分割多重方式は不適。 ③チャンネル多重はイメージ情報やファイル転送等の高速・大容量転送に不向き。 ④レジスタ挿入方式は遅延時間が増加し、送受信制御が複雑化する。 ⑤信頼度ネック解消の観点から、集中制御方式は不向き。
同期方式	独立同期	①集中制御ノードを必要とせず、高いシステム信頼性が確保可能。 ②中継時のジッタ累積がない。 ③システム収容可能ノード数制約の緩和。
伝送路符号	8B10B	①8B10B 符号変換は、バイトオリエントなデータに対し比較的簡単な符号変換回路で実現でき、障害回復等に用いる特殊パターンの設定も容易。 ②所要伝送速度の 2 倍の帯域を必要とする符号は、伝送路、送受信回路等がコスト高になる。特に 100 Mb/s 程度の伝送域では、経済性および信頼性に優れた LED 適用が比較的容易なため、2 倍の帯域を必要とする符号は不適。 ③RZ, NRZ 等はタイミング抽出が困難で、直流成分変動大による AGC 回路負担が大きい。 ④伝送系で用いられる mB1C 符号は、直流成分を抑えるためのスクランブラが必要で、符号変換回路の規模が増大する。

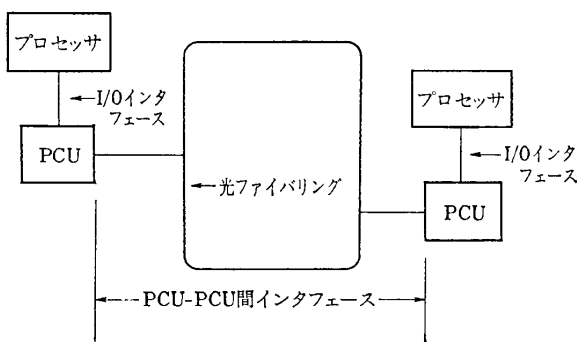


図 1 システム構成概要
Fig. 1 System configuration.

3. プロセッサ間通信方式

高速のプロセッサ間通信実現のためには、伝送効率を向上させるだけでなく、伝送路トラヒック制御の改善、すなわち通信制御に要するオーバーヘッドを削減し、ハードウェア・ソフトウェア処理双方の高速化が

不可欠である。本章では、通信制御に要するソフトウェア処理高速化に有効な PCU ハードサポート機能について述べ、次に上記ハードサポート機能との親和性に優れ、簡素で高速性に富んだデータリンク制御機能について述べる。

3.1 通信制御に関するソフトウェアインタフェース

3.1.1 通信高速化のポイント

リング等の共通伝送路を介したプロセッサ間通信におけるデータ転送は、通信制御プログラムが管理する送受信バッファ間でいったんデータ転送を行い、割込み等を契機に同バッファより各ユーザ領域へ移送する方式が一般的である。この方式は、高トラヒック、長データ転送の環境下では送受信バッファ管理や割込み処理、データ移送等の通信処理に要するステップ数、性能上のネック要因として無視し得なくなる。

一方、1対1チャンネル結合のプロセッサ間通信で採

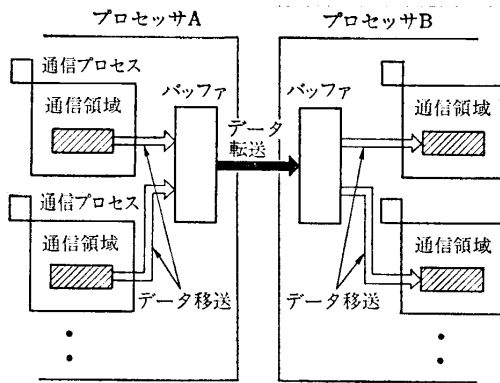


図 2 メッセージバッファ間通信方式
Fig. 2 Transmission through the data buffer.

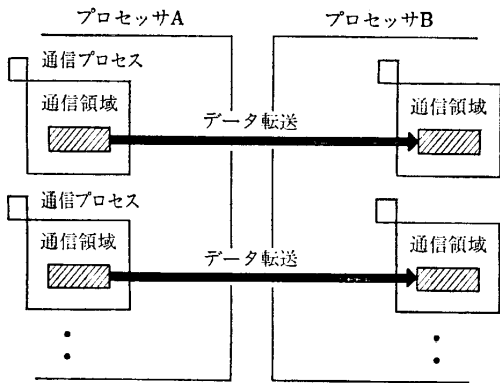


図 3 通信プロセス間直接転送方式
Fig. 3 Direct transmission between each transmission process.

用されている複数サブチャネルを用いた多重通信方式では、個々の通信プロセスごとにサブチャネルを用いて通信パスを特定できるため、送受信バッファを個々の通信プロセス側で用意し、該バッファ間でデータの直接転送が可能である。さらに、従来通信制御プログラムで行っていた送受信バッファのフィックス/フロート制御も、同一送受信バッファを連続して使用することが可能なため、フィックス化、フロート化を最適な契機で実施することが可能となる⁸⁾。このため、従来のメッセージバッファ間通信方式と比べて、通信処理に要するダイナミックステップ数を 1/2 以下に低減可能である⁹⁾。

従来のメッセージバッファ間通信および個々の通信プロセス間直接通信の方式イメージをそれぞれ図 2、図 3 に示す。

本稿で対象とする完全群接続されたシステムで、上記通信プロセス間直接通信方式と同等の通信効率を確保するためには、この通信効率化を可能としている諸

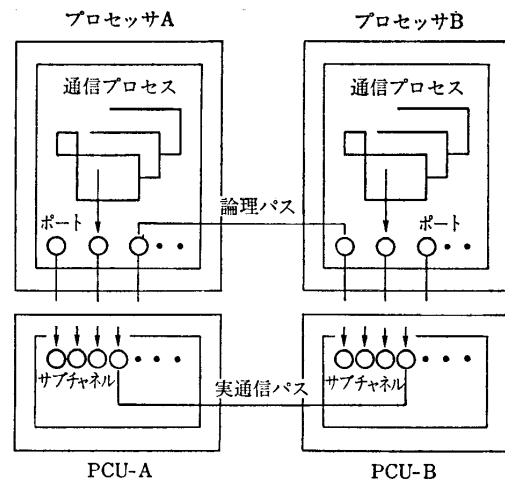


図 4 通信の基本概念
Fig. 4 Principle of processor-to-processor communications.

機能——複数サブチャネルを持ち、各サブチャネル対応にチャンネルを早期解放して、アテンション割込みを伴わずに相手からの通信起動を待ち合わせることでできる多重通信機能——を PCU でもサポートするとともに、複数の相手と通信を行う PCU 特有の問題として、通信相手の PCU アドレスとサブチャネル番号を、現チャンネルプログラム仕様下で効率的に指定する通信先アドレス指定法を確立することが重要となる。

3.1.2 通信先アドレス指定法

(I) 通信の基本概念

PCU を介したプロセッサ間通信は、図 4 に示す基本概念の下で実施されるものとする。

(1) 論理バス：個々の通信プロセスは、論理バスを指定して通信を行う。このため、通信に先立って論理バスを開設する。

(2) ポート：各プロセッサ内に、論理バスを開設するための窓口としてポートを設定する。論理バスはポート名の対で意識される。

ポート名：PCU アドレス + ポート番号

(3) 実通信バス：PCU-PCU 間のデータ転送は、実通信バスを指定して行われる。

実通信バス：送受信双方の PCU アドレス
+ PCU サブチャネル番号

(II) 通信パスの設定管理

複数サブチャネルを用いたプロセッサ間通信では、ユーザプログラムが意識する論理バスを通信制御プログラムが実通信バスにマッピングし、この実通信バスを構成するサブチャネル単位の多重通信を行う。1 対

表 2 完全群接続系でのアドレス指定に関する問題点とその対処法
Table 2 Problems of destination addressing in the ring network.

項番	問題点	対処法	評価/理由
1	既存のチャンネル仕様では、チャンネル配下のデバイスとして自側 PCU のサブチャンネルを指定できるのみで、通信相手の PCU アドレスとサブチャンネル番号の指定ができない。	案A: 通信を伴うコマンド発行時、相手アドレスは該コマンドのデータでその都度指定する(データグラム方式) 案B: ①各 PCU サブチャンネル対応に通信相手の PCU アドレスとサブチャンネル番号を記憶し管理するパス管理テーブル (CTBL) を設置。 ②通信の前処理として、専用コマンドにより CTBL を設定し実通信パス確立。 ③通信を伴うコマンド発行時、自側 PCU サブチャンネル番号のみを指定。相手アドレスは CTBL 検索により PCU ハードが自動設定(バーチャルサーキット方式)。	案B: better (理由) ・1対1通信と同等の通信効率実現可能 ・1対1通信との親和性が高く通信制御プログラムの一部流用も可能
2	リングに接続される全 PCU 相互の通信が可能のため、実通信パス設定が複雑。特にシステム更改、障害時等のパス再設定処理が複雑化し影響範囲も大きい。	案A: ソフトであらかじめ余裕を持った実通信パスの設定を行い、システム更改、障害等の影響を最小限にとどめる。 案B: 通信相手 PCU の指定を行えば送受信双方の PCU サブチャンネルの対応関係は、PCU ハードが自動選定。	案B: better (理由) ・ソフト論理の簡素化 ・システムジェネレーション簡素化 ・トラヒックに即応した最適パス設定の容易化

1 通信の場合は、チャンネル間アダプタのサブチャンネルをデバイスアドレスで選択指定することにより、一意に実通信パスが決定される。しかし、多数台の完全群接続の場合は 2 個の PCU サブチャンネル間で実通信パスを設定するためには、互いに通信相手の PCU アドレスと PCU サブチャンネル番号を持つことが必要となる。表 2 に完全群接続系での通信先アドレス指定に関する問題点の所在とその解決案の比較を示す。表 2 の比較から、本稿で提案する実通信パス設定方式の動作を図 5 に示す。

3.2 データリンク制御機能

3.2.1 基本設計

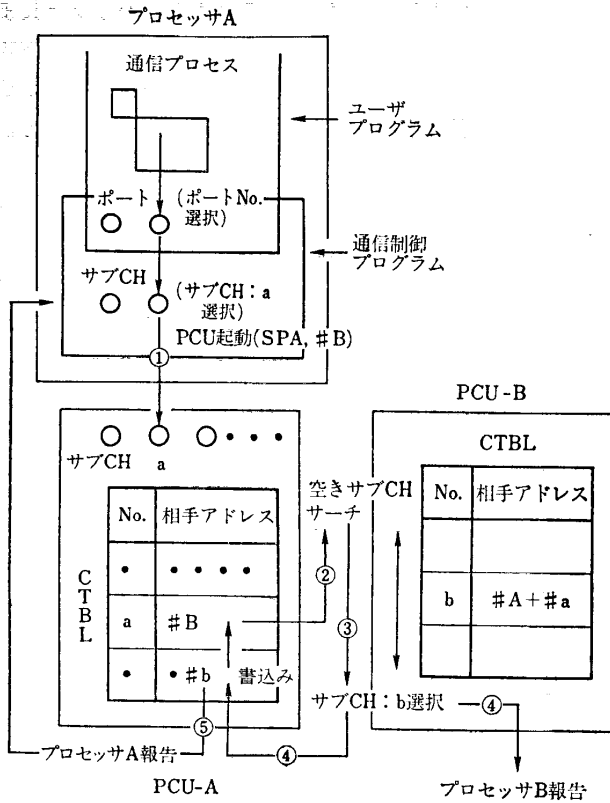
データリンクレベルプロトコル設定にあたり、大別して HDLC 手順を高速の完全群接続されたチャンネル結合プロセッサ間通信用に拡張する方向と、IEEE 802 トークンリング方式¹⁰⁾または ANSI のファイバ分散型データインタフェース (FDDI)¹¹⁾等の LAN 標準仕様に準拠し、データリンクレイヤの一部(例えば LLC)または上位レイヤで高速のプロセッサ間通信用の機能追加を行う方向とが考えられる。本システムにおいては、前者の方向が以下の理由により適合性が高いと考える。

(1) IEEE 802 標準仕様は、主として会話型で動作する端末装置を計算機に接続する 4 Mビット/秒程度の低速のフロントエンド型ネットワークを指向して

おり、高速の領域に適用するには伝送効率、フレームの信頼性、ハードウェア量/コスト等に問題がある。

(2) ANSI の FDDI 規格は、IEEE 802 標準仕様を高速領域に適用する場合の問題点と考えられる差分マンチェスタ方式の伝送路符号、従属同期方式、送信権移動契機等が改善され、応答方式も信頼性上一定の改善が見られる。しかし、チャンネル結合のプロセッサ間通信用に適用するには、送信権制御が複雑であり、機能の簡素化、応答の信頼性のより一層の改善等が望まれる。

(3) チャンネル結合されたプロセッサ間通信用においては、送受信双方のプログラムが発行する通信用コマンド間で同期をとってデータ転送が実施される(後述の図 7 の動作例参照)。通信回線を介した転送のように、チャンネル転送速度に比べて伝送器の速度が遅く、信頼性にも欠ける点があれば、PCU-PCU 間の通信とプロセッサ-PCU 間の通信を明確に分離した方がよい。しかし、本システムのように比較的短距離で信頼性の高い高速の伝送路を持つ場合、PCU-PCU 間の伝送路遅延をプロセッサ-PCU 間のチャンネル転送速度と比べて無視できるほど小さいと見ることができ(3.2.4 項参照)ため、PCU-PCU 間は単にプログラムが発行する通信用コマンドとデータを伝達するだけの手順を用意し、送受信双方の通信・同期制御は直接上記の通信用コマンドレベルで実施する方が簡潔で



[実通信パス設定手順]

- ①通信制御プログラムが PCU-B との実通信パス設定用に選択した PCU-A のサブチャンネル(a)に対して通信パス設定コマンド (SPA と呼称) により、相手 PCU アドレス情報 #B を転送
- ②サブチャンネル(a)に対応する CTBL エントリの相手アドレス域に #B を設定し、上記 SPA を PCU-B に転送
- ③SPA を受信した PCU-B は、CTBL 内で相手アドレスが確立していない任意のサブチャンネル (b) を選択し、b の相手アドレス域に #A+#a を設定
- ④PCU-B のサブチャンネル(b)に対応する相手アドレスが確立したことを PCU-B の上位プロセッサ並びに PCU-A に報告
- ⑤PCU-A は PCU-B からの上記報告を基に、サブチャンネル (a) の相手アドレスとして #b を追加設定し、一連の SPA コマンド処理が完了したことを PCU-A の上位プロセッサに報告

図 5 案 B による実通信パス設定動作例

Fig. 5 Example of communication path setting.

伝送効率、ハードウェア量の面からも優れている。

3.2.2 送信権制御方式

送信権制御方式設定にあたっては、送信権獲得の契機と送信権放棄の契機に着目する必要があるが、均等の送信機会を与えられたプロセッサ間的高速通信の環境下では、トークンフレーム受信即送信権獲得とするのが効率的である。したがって、後者のみに着目した場合、送信権制御法として以下の3案が考えられる。

案A：送信待ち状態にある通信フレームをすべて送出し終わった時点で送信権放棄

案B：送出した1フレーム受信時点で送信権放棄

案C：1フレーム送出時点で直ちに送信権放棄

このうち、以下の理由により、案Cが高速のプロセッサ間通信への適用を主目的とした送信権制御方式としては最も優れている。

(1) 案Aは特定の PCU に長時間送信権が占有されることがあり、送信権監視等の制御が複雑化するとともに伝送速度 (100 Mb/s) とチャンネル-PCU間転送速度 (~24 Mb/s) との性能差を考えると、後者の転送速度が性能上のネック要因となり、伝送速度の効率的利用によるシステムスループット向上が実現できない。

(2) 間欠障害対策として、後述の再送をサポートするためには、ビジーまたは無応答検出時点で送信フレームが直ちに送出できる状態に保持されていることが望ましい。案Aでこれをサポートすると、送信バッファ等のハードウェア量が大きくなる。

(3) 案Bは優先制御の付加機能を実現する上で有効な手法であるが、本システムのようにリング伝送速度と対チャンネル転送速度の差が大きい場合、次のフレーム送出要求までに送信権を獲得できないケースは少ないと考えてよく、優先制御の必要性は小さい。

(4) 案Bと案Cのメッセージ通過時間をランザクション、リング長、メッセージ長をパラメータとして比較すると図6に示す結果が得られる (VL: リング伝送速度, VI: チャンネル-PCU 間転送速度, L: リング長, T_{pcu} : PCU 内処理時間, D: PCU 内伝送遅延, B: 転送メッセージ長; 算出式の詳細は付録参照)。全般的に案Cの方が案Bよりメッセージ通過時間が短く、リング長が長く、メッセージ長が短いほどその傾向が顕著となる¹²⁾。

3.2.3 応答方式

データリンクレベルでの受信フレームに対する応答の返し方として、以下の3案が考えられる。

案A：フレーム末尾の受信インディケータビットを受信側が ON にすることにより応答を返す。

案B：受信側が送信権獲得時、応答フレームにて応答を返す。

案C：フレーム送出時、送信側で当該フレームのうしろに応答用のスペースを確保してから、送信権を放棄し、受信側はこのスペースに回答フレームを挿入することにより応答を返す (送信

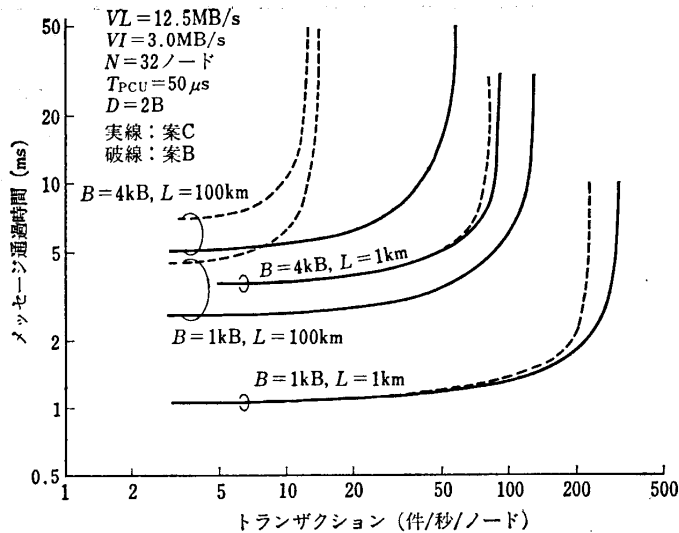


図 6 送信権制御方式とメッセージ通過時間

Fig. 6 Message elapse time comparison among token control features.

権不要)¹³⁾.

上記 3 案のうち、以下の理由により 100 Mb/s 程度の高速伝送への適用を主目的とした応答方式としては、案 C が最も優れている。

(1) 案 A は、受信インディケータビットの ON 制御を 100 Mb/s の伝送速度に同期して高速に行うことが必要となり、ハード実現上の負担が大きい (コスト/ハードウェア量増大)。また、受信インディケータビットを FCS の対象とすることが困難で、応答の信頼性に問題がある。

(2) FCS エラー、バッファビジー等によりフレーム受信ができなかった場合、高信頼度化、ソフトウェアオーバーヘッド削減等の観点から、データリンクレベルで送信側から該フレームを再送することが望まれる。案 B は送信側で再送の要否を判断できるまでの時間が長く、送信バッファの利用効率が悪くなる。

(3) トラヒックが集中した場合にも何らかの応答を返すことが望ましいが、案 B の場合たくさんの応答フレームを記憶する手段が必要となる。

(4) 案 C は応答領域追加により実質的なフレーム長増大となり、その分伝送効率の低下をもたらす。しかし、伝送路ネックがシステム性能に影響するトラヒック条件は極めてまれなので、その影響は無視できる (3.2.4 項参照)。

3.2.4 分割転送方式

PCU-PCU 間のデータ転送方式には、大別してプログラムが指定するデータ長を制限するが、PCU 内

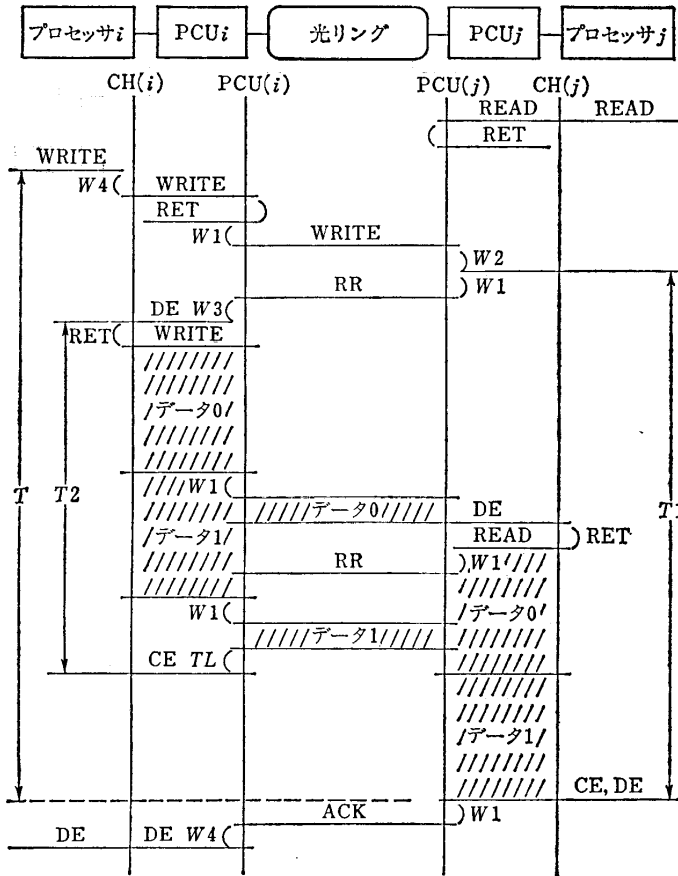
に十分長いバッファを用意し、バッファ長までのデータを一括して送信する一括転送方式と、プログラムが指定するデータ長に制限を付けずハードウェアが一定長にデータを分割して送る分割転送方式とがある。

前者は、データの欠送対策が不要でバッファ管理、再送制御および冗送対策も前者と比べて簡単なため、広く採用されている方式である。しかし、チャンネル-PCU 間と PCU-PCU 間の転送速度差に着目すれば、以下に示すように後者の方がメッセージ通過時間を短縮することができる。また、プロセッサ間通信の転送データ長に関するソフトウェア制限が統一されていない現状を考えると、前者の方式はデータ長分割のための新たなプログラム上の手順追加が必要になる場合がある。

このため、高速で汎用性の高いプロセッサ間通信に適用する観点から、後者の分割転送方式が前者の一括転送方式より優れている。

【メッセージ通過時間の比較】分割転送方式によるデータ転送動作例を図 7 に示す (各 PCU は送受信バッファを複数個有し、空きバッファには先行制御によりデータを格納することができるものとする)。

分割転送の場合、仮にチャンネル-PCU 間転送速度を 3 メガバイト/秒としても、リング上の伝送速度の 1/4 以下なので、受信側 PCU からの RR 応答受信時にまだ送信データが送信バッファに格納されていないケース (I/O インタフェースネック) が多く、この場合、伝送路上のフレーム転送と受信側 PCU からのチャンネルへのデータ転送が送信側 PCU のチャンネルからのデータ受信動作に隠れてしまう。このため、全メッセージ通過時間 (図中の T に相当) の中で伝送路上の遅延時間と受信側 PCU のチャンネルへのデータ転送時間が占める部分は、最後の 1 フレームを送信側 PCU が伝送路に送出し始めてから、該フレームが受信側 PCU からチャンネルへ転送され終わるまでだけである。一方、一括転送の場合、全データのチャンネルから送信側 PCU への転送、伝送路上の遅延、受信側 PCU からチャンネルへの転送が完全に直列になるため、全メッセージ通過時間の中にはこれらの時間がすべて加算されてしまう。両者のメッセージ通過時間をトランザクション量と分割数 (分割数 1 = 一括転送) とをパラメータとして比較すると、図 8 に示す結果が得られる (算出式の詳細は付録参照)。ト



RET : コマンド再試行要求
 DE : コマンド動作終了 (CH 再結合要求)
 CE : データ転送動作終了 (→CH を解放)
 RR : データ受信準備完了
 ACK : データ受信完了
 T : メッセージ通過時間
 T1 : 受信バッファ保留時間
 T2 : 送信動作によるチャネル占有時間
 W1 : トークン待ち時間
 W2 : 受信バッファ空き待ち時間
 W3 : チャネル再結合待ち時間
 W4 : 起動/報告待ち時間

図 7 データ転送動作例
 Fig. 7 Example of data transmission.

ランザクションが高い領域では、伝送路遅延が大きくなって伝送路ネックとなる傾向にあり、分割転送と一括転送とのメッセージ通過時間に逆転現象が見られる。しかし、分割転送を必要とする長データ(キロバイトオーダー)転送を、この逆転現象が見られるトランザクション量以上の高トラヒック下で実行するシステムは極めてまれと考えてよく、このようなシステムは伝送路の転送速度を一層速くして、伝送路ネックを解消するのが本筋である。したがって、図 8 のシステムを例にとると、分割転送の方が分割数 2 で約 30~40%、分割数 4 で約 60~70% メッセージ通過時間が短縮され、これがより長データになるほど、差はより顕著になる。

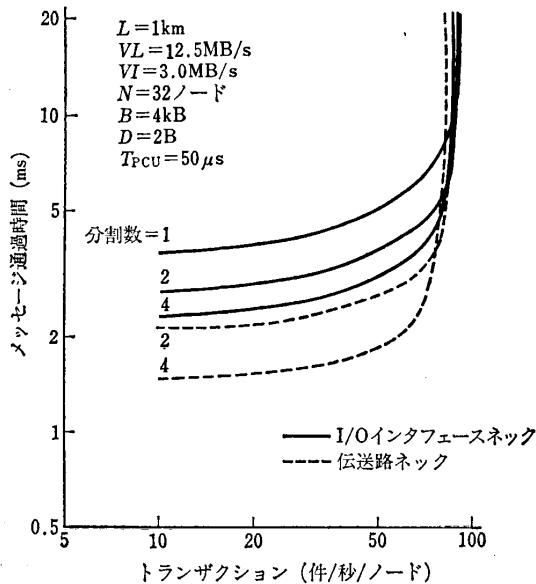


図 8 分割数とメッセージ通過時間
 Fig. 8 Message elapse time comparison among message partitioning features.

3.2.5 フレーム構成

以上のデータリンク制御手順は、HDLC フレームを図 9 に示すように完全群接続系向きに拡張した簡潔なフレーム構成により実現できる。制御コマンドで情報フレーム、応答フレーム (ACCEPT/BUSY)、トークンフレームのフレーム種別を指定する。

4. 障害処理方式

4.1 基本方針

伝送路、PCU のリピータ部の障害等により、送信権消滅またはクロック異常が検出された場合、システムがハングアップしてしまうため早急な障害箇所の切

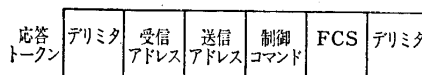
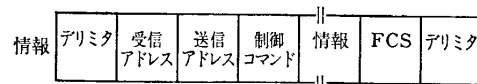


図 9 フレーム構成
 Fig. 9 Frame format.

離し、システム復旧が要求される。このため、これらの障害回復をノードのハードウェア機能により自動的にを行う方式が幾つか提案され実用化されている。

本システムにおいても、上記障害に対して、PCU-PCU 間での自動的な回復手段を設けることとし、以下に具体的手法の提案を行う。

4.2 送信権消滅対策

送信権はトークンフレームによりリング上を巡回するため、ビット耐力が必ずしも十分でない。このため、間欠障害による送信権消滅対策として、送信権消滅検出ノードがトークンフレームを生成し、送信権を復活する方式が提案されている。しかし、この方式では固定障害による送信権消滅に対しては効果がない。

固定障害による送信権消滅にも対応するために、以下の2段階で送信権回復を行うのが望ましい。その具体的手順を図10に基づいて説明する。

〈障害発生状況〉

PCU-A~F で構成されるリングにおいて PCU-A と PCU-F との間の両伝送路異常。

〈PCU 機能条件〉

①各 PCU は送信権監視用タイマ (T_x & T_y) を有し、以下の条件を満足する条件をタイマ値が設定される。

$$T_x = \alpha + \beta \cdot \gamma \quad (\alpha: \text{トークンフレーム最大受信間隔}, \beta > \text{フレームのリング一周時間}, \gamma: \text{PCU アドレス})$$

$$T_y > \max. T_x$$

②トークンフレーム検出 PCU は T_x , T_y をリセット & スタートする。

③各 PCU は現用パスに対してのみクロック送出/抽出を行う。

【第1段階; 間欠障害に対する送信権回復】 T_x オーバフロー検出 PCU (T_x 設定条件より、複数 PCU が同時に T_x オーバフローを検出することはない) がトークンフレームを生成し、該フレームの一巡チェックを行う (T_x のみリセット & スタート)。トークンフレームが戻ってくれば送信権回復とみなして通常動作を継続する。戻ってこなければそのまま放置する。

【第2段階; リング接続構成変更による送信権回復】

(1) A以外の PCU は、A~E での T_x オーバフローにより最低1回のトークンフレームを受信する。このため、 T_y オーバフローを検出するのはAだけである。 T_y オーバフロー時、Aは現用パスに対し

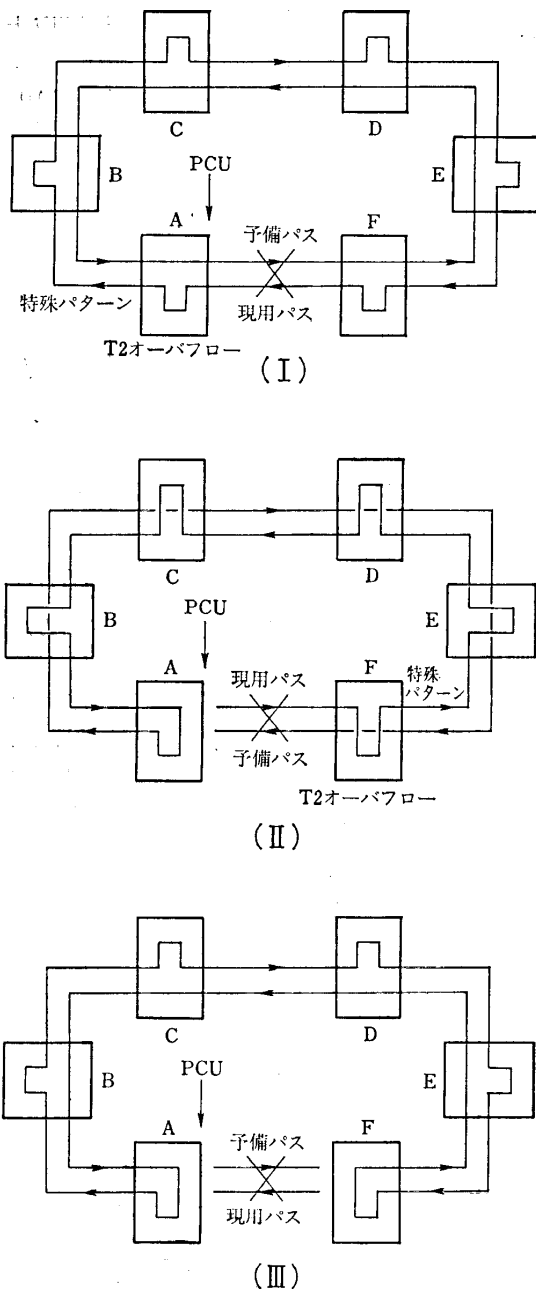


図10 送信権自動回復方式

Fig. 10 Automatic recovery from token missing.

て特殊パターンを送出する (図10-I 参照)。

(2) 特殊パターンを受信した B~F は現用パスを予備系に切り替えて T_x , T_y をリセット & スタートする。Aは特殊パターン送出後、同様にパス切替えを行って T_x , T_y をリセットする (図10-II 参照)。

(3) Aは自 PCU 宛のトークンフレームを送出し T_x , T_y をスタートする。上記自 PCU 宛トークンフレームを受信しない場合は両パス障害とみなしてルー

バック状態に切り替わる。

(4) (1)と同様の理由により、今度はFでT_yオーバーフローを検出する。このため、上記(1)~(3)の動作をFからの特殊パターンを契機に実行する。

(5) AとF両者がループバック状態に切り替わると、その後最初にTxオーバーフローを検出したPCUが発行する自PCU宛トークンフレームが送信元で受信されるので、送信権回復とみなして通常動作を再開する(図10-III参照)。

4.3 クロック異常対策

伝送路またはPCUのリピータ部の障害、あるいは送受信プロセッサ間の速度偏差が所定の値以上になった場合、クロック異常が検出されることがある。このクロック異常は必ず送信権消滅を伴うため、経済性の観点からはクロック異常対策を前記送信権消滅対策と一本化する案が考えられる。しかし、障害検出・システム復旧の即応性の観点からはPCUにクロック異常検出に伴って作動する4.2節と同様のリング接続構成変更機能をサポートするのが望ましい。いずれを採用するかは、PCUに対するトータルの要求条件を考慮して決定するのが肝要である。

5. むすび

完全群接続された任意のプロセッサ間の通信を高速かつ経済的に実現するための方式条件を明らかにした。通信制御に関するソフトウェアインタフェースについては、従来のチャンネル間アダプタ(CTCA)等を介した1対1のプロセッサ間のチャンネル結合を完全群接続されたプロセッサ間の通信システム用に拡張したものであり、現在広く使用されているプロセッサ間通信インタフェースに合わせた点に特徴がある。また、データリンクレベル以下のリングアーキテクチャについては高速の光ファイバ通信向きに簡素化したプロトコルを提案し、そのプロトコルの優位性を示している。

上記方式条件に沿ったプロセッサ間結合システムは、任意のプロセッサ間で高速通信を指向した汎用性の高い方式として広く実用化に供し得るものと期待できる。

謝辞 本検討にあたりご指導、ご助言をいただいたNTT電気通信研究所の橋本昭洋知能処理研究部長、松永俊雄情報処理研究部主幹研究員並びに関係各位に深謝します。

参 考 文 献

- 1) Joshi, S. and Lyer, V.: New Standards for Local Networks Push Upper Limits for Light-wave Data, *Data Communications*, pp. 127-138 (July 1984).
- 2) Burke, R. G.: Fiber Optics in Computing Systems, Proc. of the IEEE International Symposium on Circuits and Systems, pp. 176-178 (Apr. 1977).
- 3) Stuck, B. W.: Calculating the Maximum Mean Data Rate in Local Area Networks, *Computer*, Vol. 16, No. 5, pp. 72-76 (1983).
- 4) Widmer, A. X. and Franaszek, P. A.: A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code, *IBM J. Res. Dev.*, Vol. 27, pp. 440-451 (1983).
- 5) Bux, W.: Local-Area Subnetworks: A Performance Comparison, *IEEE Trans. Commun.*, Vol. COM-29, No. 10, pp. 1465-1473 (1981).
- 6) Strole, N. C.: A Local Communications Network Based on Interconnected Token-Access Rings: Tutorial, *IBM J. Res. Dev.*, Vol. 27, No. 5 (1983).
- 7) 田崎他: リング型ローカルエリアネットワークにおける同期方式の検討, 情処分散処理ネットワーク研究会, IN83-68, pp. 31-36 (1983).
- 8) 中野, 森: 疎結合計算機システムにおける高速計算機間通信方式, 情処第32回計算機アーキテクチャ研究会 (1981).
- 9) 星子他: 共通バスを介したプロセッサ間通信効率化の検討, 情処第24回全国大会, 6H-6 (1982).
- 10) IEEE Project 802, Local Area Network Standards Draft IEEE Standard 802.5 Token Ring Access Method and Physical Layer Specifications, Working Draft Aug. 5 (1983).
- 11) ANSI, Draft Proposed, FDDI Token Ring Media Access Control, June 1 (1984).
- 12) 魚住: トークンリングの送信権制御法に関する一検討, 情処第27回全国大会, 3J-1 (1983).
- 13) Hatta, H. and Yasue, K.: Technical Considerations and Implementations on an Optical Token Loop, IFIP WG 6.4, Sept. (1983).
- 14) 橋田: ポーリング制御における待合せモデルの解析, 昭和43年度信学会全国大会, p. 129 (1968).

付録 メッセージ通過時間算出条件

PCUを介したプロセッサ間通信は図7に示すとおりであり、トークン待ち時間 W_1 は制限式多重待ち行列($K=1$)モデルにて表すことができる¹⁴⁾。一方、受信バッファ空き待ち時間 W_2 及びチャンネル再結合待ち時間 W_3 はM/M/1の待ち行列モデルにて近似する。したがって、送信権制御方式案B, 案Cそれぞれ

表 A 平均メッセージ通過時間算出式
Table A Average message elapse time.

項番	項目	記号	送信権管理方式(案B)		送信権管理方式(案C)	
			I/O インタフェースネック	伝送路ネック	I/O インタフェースネック	伝送路ネック
1	平均メッセージ通過時間	T	$W4 + T_{pcu} + W1 + Ts + TL/2 + W2 + T1$	同左	同左	同左
2	受信バッファ保留時間	$T1$	$2T_{pcu} + W1 + Ts + W3 + T2 + Tcp$	同左	同左	同左
3	送信動作によるチャネル占有時間	$T2$	$2T_{pcu} + P \cdot Tcp + W1 + Td + TL$ $2T_{pcu} + Tcp + W1 + Td + TL$ $+ (2W1 + Ts + Td + TL)(P-1)$		案B(I/Oインタフェースネック)と同じ	案B(伝送路ネック)と同じ
4	トークン待ち時間	$W1$	$\bar{F}^2/2F + \alpha \cdot \bar{F}1^2/2(1 - \alpha \cdot F1)$; $F1 = F + h$ $F = N \cdot U + (N-1)\rho \cdot h$ $\rho = \alpha \cdot N \cdot U / (1 - \alpha \cdot N \cdot h)$ $\bar{F}^2, \bar{F}1^2$: 2乗平均 $\alpha = (2 + 2P)\lambda$	同左	同左	同左
5	受信バッファ空き待ち時間	$W2$	$\lambda \cdot T1 \cdot T1 / (1 - \lambda \cdot T1)$	同左	同左	同左
6	チャネル再結合待ち時間	$W3$	$\lambda \cdot T2 \cdot T2 / (1 - \lambda \cdot T2)$	同左	同左	同左
7	起動/報告待ち時間	$W4$	$\lambda \cdot T2 \cdot T2$	同左	同左	同左
8	平均フレーム送出時間	h	$(2Ts + (Td + Ts)P) / (2 + 2P)$	同左	同左	同左
9	平均歩行時間	U	TL/N	同左	同左	同左
10	リング一周時間	TL	$5L + D \cdot N/VL$	同左	同左	同左
11	ショートフレーム送出時間	Ts	$A/VL + TL$; A : ショートフレーム長=45バイト	同左	A/VL ; $A=43$ バイト	同左
12	データフレーム送出時間	Td	$(B/P + A)/VL + TL$	同左	$(B/P + A)/VL$	同左
13	データブロックのI/Oインタフェース転送速度	Tcp	$B/P/Vi$	同左	同左	同左

λ : トラヒック (件/秒/ノード), N : ノード数, L : リング長 (km), D : 遅延バイト数/ノード (2バイトに設定), P : 分割数, B : 転送データ長/コマンド, T_{pcu} : PCU 内処理時間 (50 μ s に設定)

れのメッセージ通過時間は、各待ち時間を独立事象とみなすことにより表Aに示す算出式で求めることができる。なお、本表で $P=1$ は分割転送無しの場合である。

(昭和60年6月13日受付)

(昭和60年12月19日採録)

星子 隆幸 (第27巻第1号参照)

魚住 栄市 (第27巻第1号参照)

小柳津育郎 (第27巻第1号参照)