

複数リンクを用いた高効率データ転送方式の提案

平田 俊明[†] 近藤 毅[†] 宮崎 聡[†]

2つの装置間で複数のリンクを用いてデータ転送を行う方式には、複数のリンクに順次データ単位を割り当てるマルチリンク方式と1つのデータ単位を分割し、これをおのおのリンクに割り当て、受信側でこれを組み立てるパラレルリンク方式がある。このうち、パラレルリンク方式では、データ転送の並列性と逐次的なソフトウェア処理のトレードオフにより転送時間を最小にするデータ単位の分割数、すなわち、リンク上の転送データ長が決まる。これに対して、従来のパラレルリンク方式では、1つのデータ単位を単純にリンク数に分割し、これをおのおのリンクに割り当てる方式であり、データ転送時間を最小化するための考慮がなされていない。本論文では、パラレルリンク方式において、データ転送時間の最小化を考慮したデータ転送方式を提案する。提案方式では、使用できるリンク数を有限とし、1つのリンク上の転送データ長の上限を考慮する。また、各リンクのデータ転送速度は、すべて同一の場合と速度が異なる場合の両方についてデータ転送方式を提案する。提案方式を典型的なデータ転送システムのモデル上で定量的に評価した結果、提案方式の有効性が確認できた。

A Study of High Efficient Data Transfer Method Using Several Links

TOSHIAKI HIRATA,[†] TAKESHI KONDO[†] and SATOSHI MIYAZAKI[†]

There are two methods of enabling data transfer between two nodes using several links: the multi-link method and the parallel-link method. In the former, the data are sent to the links one by one. In the latter method, data are divided into several transfer units and sent by several links at the sending node and reassembled at the receiving node. In this method, there is an optimum data division size that minimizes data transfer time because of trade-off between data transfer time in the link and data processing time at the node. This paper proposes a method for optimal data transfer for the parallel-link method and evaluates it using a typical data transfer model.

1. 緒言

2つの装置間のデータ転送性能を向上させる方法の1つに、装置間のデータ転送を複数のリンクを用いて行う方法がある。複数のリンクを用いたデータ転送方式には、複数のリンクに順次データ単位（ユーザが転送しようとする1つのデータの単位をデータ単位とよびリンク上を転送する1つのデータの単位を単にデータとよぶ）を割り当てる方式^{1),2)}と、1つのデータ単位を分割し、これをおのおのリンクに割り当て、受信側でこれを組み立てる方式³⁾がある。前者の方式をマルチリンク方式、後者の方式をパラレルリンク方式とよぶ。マルチリンク方式は、データを連続的に転送する場合、すなわち、各種のアプリケーションでリンクを多重化して使用するようなスループット性能を重

視する場合に有効な方法である。マルチリンク方式は、OSI (Open Systems Interconnection) 第2層に位置付けられるマルチリンク手順として規定されており¹⁾、待ち行列の構成やデータの分配アルゴリズムについて論理解析やシミュレーションを用いた評価が多数行われている^{2),4)~6)}。一方、パラレルリンク方式は、個々のデータの転送時間、すなわち、応答時間性能に注目した場合有効である。例えば、ホスト計算機に直結する端末との入出力やプロセス制御用計算機の制御機器との入出力のように応答時間を重視するアプリケーションが専用リンクを使用する場合である。また、パラレルリンク方式は、マルチリンク方式に比べて以下の点で有利である³⁾。

- (1) パラレルリンク方式では、データ単位を送信側と受信側で正しい順序で受け渡せる。
- (2) 1つのリンク上の転送データ長に制限がある場合でもパラレルリンク方式では、受信処理が短時間に

[†] (株)日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

行える。

上記(1)は自明である。上記(2)は次の理由による。マルチリンク方式では、あるリンクに対して1つのデータ単位を分割して順次送信し、受信側で組み立てる必要がある。このとき、送信データ長が同一の場合、受信側では、最初の分割データを受け取ってから、ある一定の間データの組立てを行うことができない。そして、しばらくした後、到着したデータを複数のリンク上で一斉に組み立てなければならず、受信側のCPUが一時的に混雑する。それに対して、パラレルリンク方式では、同一サイズのデータを送信した場合でも、一定時間間隔でデータを受信することができるため、一時的に受信側のCPUが混雑することを防げる。

本論文では、複数のリンクを用いたデータ転送方式として、パラレルリンク方式を対象とする。2つの装置間のデータ転送時間は、装置のCPU上のソフトウェア処理時間とリンク上の転送時間の和になる。CPUは1つと仮定すれば、ソフトウェア処理は、逐次処理であり、リンク上の転送処理は、複数リンクによる並列処理が可能である。データ転送を複数のリンクに分散して行う場合、データ転送の並列性とソフトウェア処理オーバーヘッド(データの分割数を多くするほどオーバーヘッドが大きくなる)のトレードオフにより装置間の転送時間を最小とするデータ単位の分割数、すなわち、各リンク上の転送データ長が決まる。従来のパラレルリンク方式では、1つのデータ単位をリンクの数に分割し、これをおおののリンクに割り当てる方式であり³⁾、データ転送時間の最小化をはかるための考慮がなされていない。本論文では、複数のリンクを用いたパラレルリンク方式において、データ転送時間の最小化を考慮したデータ転送方式を提案し、これを典型的なデータ転送システムのモデル上で定量的に評価する。提案方式は、使用できるリンク数を有限とし、1つのリンク上の転送データ長の上限を考慮したものとす。また、リンクのデータ転送速度は、すべて同一の場合と、速度が異なる場合の両方について、データ転送方式を提案する。なお、本論文では2つの装置間のデータ転送性能を向上させることを目的としたデータ転送方式を議論するものであり、本来誤り率の低い伝送路が対象になること、また議論を複雑にしないため、誤り制御については考えないこととする。以下、2章では本論文で対象とするパラレルリンク方式を示し、3章で高効率データ転送方式を提案す

る。4章で提案方式の定量的な評価を行う。

2. パラレルリンク方式

パラレルリンク方式では、図1に示すように2つの装置間を N 本のリンクで結び、これらのリンクによりデータを転送する。LC(Link Controller)は、各リンク上のデータ転送制御を行う。PLC(Parallel Link Controller)は、図2に示すように上位レイヤから送信要求のあった1つのデータ単位を分割し、これを配下のLC(リンク)に分配し、受信側でこれを組み立て1つのデータ単位として上位レイヤに渡す。PLCでは、上位からのデータ単位を要求順に転送し、送受信ノード間でデータ単位の順序性が保証されるようにする。すなわち、1つのデータ単位の転送で N 本のリンクを占有することになる。各リンク上で転送可能なデータ長には上限があるものとするが、PLCは上位レイヤに対して、転送可能なデータ長の制限を設けない。すなわち、PLCは上位レイヤに対して、最大データ長の制約のない1本の仮想的な高速リンクを提供する。

パラレルリンク方式では、受信側のデータ組み立ては、データ分割の際、順序番号の入ったヘッダと最終

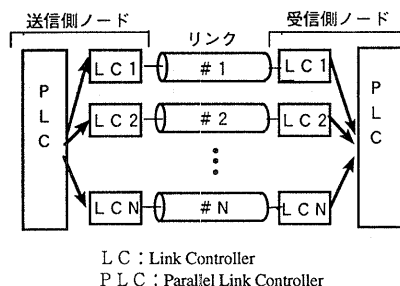


図1 システム構成

Fig. 1 Configuration of parallel link method.

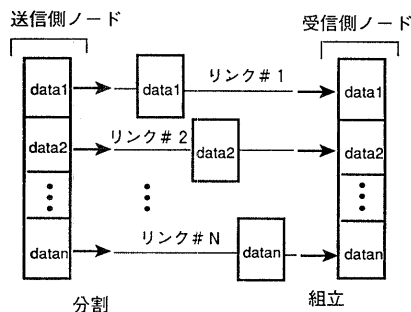


図2 パラレルリンク方式

Fig. 2 Parallel link method.

データか否かを示すビット (X. 25 の Mbit 相当) を挿入することにより実現できる. この方式では, 受信側では順番どおりにデータを接続していき, 最終データを受信した時点で, 上位レイヤにデータ単位を渡せばよい. このほか, データ組立の方法として, 送信側と受信側で次にデータを送受信すべきリンクをあらかじめ認識しておく, つまり, あらかじめリンクに順序番号をふる方法がある. この方式は, データ転送を開始するにあたって, 各リンクのコネクション確立用のデータ送受信でリンクの番号を取り決めることによって実現できる. この方式では, 送受信データ中に最終データか否かを示すビットは必要であるが, 順序番号の入ったヘッダは不要である.

3. 高効率データ転送方式

3.1 従来方式の問題点

2つの装置間のデータ転送時間は, 装置の CPU 上のソフトウェア処理時間とリンク上の転送時間の和になる. ここで, ソフトウェア処理時間とは, PLC が行うデータ分割/組立処理, データのリンクへの割り当てや, LC が行うプロトコル処理, バッファの確保/解放, リンクの起動処理等をいう. CPU は1つと仮定すれば, ソフトウェア処理は逐次処理であり, リンクの転送処理は複数リンクによる並列処理が可能である.

パラレルリンク方式では, データ転送の並列性から, 分割数を増やすほどリンク上の転送時間は短くなるが, データ送受信に必要なソフトウェア処理のオーバーヘッドは多くなる. このため, データ転送の並列性とソフトウェア処理オーバーヘッドのトレードオフにより転送時間を最小とするデータ単位の分割数, すなわち, 各リンク上の転送データ長が決まる.

従来のパラレルリンク方式では, 1つのデータ単位を単純にリンク数に分割し, これをおのおののリンクに割り当てる方式であり, データ転送時間の最小化を図るための考慮がされていない.

3.2 提案方式 (その1)

N 本のリンクのデータ転送速度がすべて同一の場合について, 高効率データ転送方式を示す. パラレルリンク方式では, 3.1節で示したように転送時間を最小にするデータ単位の分割数および各リンク上の転送データ長が存在する. L を上位レイヤから転送要求のあったデータ長, V を各リンクの転送速度, T_p を各リンクごとのソフトウェア処理時間 (LCの処理時間

+PLCの処理時間の一部), T_h をリンク対応のハードウェア処理時間 (データ長には非依存), n をデータ分割数, T_c を長さ L のデータの転送時間, l_{max} を各リンク上で転送可能な最大データ長とする. このとき, T_c を最小にするデータ分割数 n は問題1の解として与えられる.

問題1

$$\min_{n \in I} T_c = q(L/(Vn) + T_h) + nT_p \quad (1. a)$$

$$\text{subj. to } 0 < L/n \leq l_{max} \quad (1. b)$$

ここで, q は n/N を切り上げた整数値, I は自然数の集合である. T_c はソフトウェア処理時間 nT_p , リンク対応のハードウェア処理時間 qT_h およびリンク上の転送処理時間 $q(L/(Vn))$ の和で表される. ハードウェア処理時間およびリンク上の転送処理時間は, $n > N$ の場合, リンクをラップラウンドで使用するため, ラップラウンドの回数 q を乗じてある.

問題1から具体的なデータ転送アルゴリズムを導く. まず, その前に問題1において使用可能なリンク数 N および最大データ長 l_{max} を考慮しない問題2を考える.

問題2

$$\min_n T_c = L/(Vn) + T_h + nT_p \quad (2)$$

問題2の解は, 式(2)の目的関数は微分可能関数であるので, 以下の最適性条件式を満たす⁹⁾.

$$dT_c/dn = -L/(Vn^2) + T_p = 0 \quad (3)$$

式(3)を n について解くと,

$$n = \sqrt{L/(T_p V)} \quad (4)$$

となる. 本来 n は自然数であるので, 式(4)で求めた値を T_c が小さくなる方に n を切り上げまたは切り捨てる. これを n' とする. これより, 最適なデータ分割長 L_0 (KByte)は,

$$L_0 = L/n' \quad (5)$$

となる.

次に, 上記 L_0 を用いた具体的な転送アルゴリズムを以下に示す.

(1) $L_0 < l_{max}$ の場合

(a) $L \leq L_0 \times N$

この場合は, 問題1において $q=1$ であり, 制約条件式(1. b)を満たしているため, 問題1は問題2に置き換えられる. したがってデータ転送アルゴリズムは, 図3に示すように, 上位レイヤから転送要求のあったデータ単位をデータ長 L_0 に分割し, 先頭データから順次, 空きリンクに割り当てる方式となる.

(b) $L_0 \times N < L \leq l_{\max} \times N$

この場合、問題1は問題3に置き換えられる。

問題3

$$\min_n T_c = q(L/(Vn) + T_h) + nT_p \quad (6. a)$$

$$\text{subj. to } n \geq N \quad (6. b)$$

$$0 < L/n \leq l_{\max} \quad (6. c)$$

この場合は制約条件式(6. c)は明らかに満たしているため、制約条件式(6. b)のみを考慮すればよい。問題3の解は、付録Aに示すように $n=N$ となる。したがって、データ転送アルゴリズムは、図4に示すように、上位レイヤから転送要求のあったデータ単位を N 分割し、先頭データから順次、空きリンクに割り当てる方式となる。

(c) $L \geq l_{\max} \times N$

この場合は問題3において制約条件式(6. b)は明らかに満たしているため、制約条件式(6. c)のみを考慮すればよい。このとき、問題3は解析的に解を求めることが困難である。よって、付録Bに示すように、近似的に $n=L/l_{\max}$ を解とする。したがって、データ転送アルゴリズムは、図5に示すように、上位レイヤから転送要求のあったデータ単位を l_{\max} に分割し、先頭データから順次、空きリンクに割り当て、データ分割数が N を超える分については、転送が終了したリンクに順次割り当てる方式となる。

(2) $L_0 \geq l_{\max}$ の場合

この場合、問題1を直接解く必要があり解析的に解

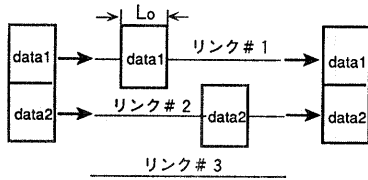


図3 $L < L_0 \times N$ で $N=3$ の場合
Fig. 3 Data transfer algorithm in the case of $L < L_0 \times N$ and $N=3$.

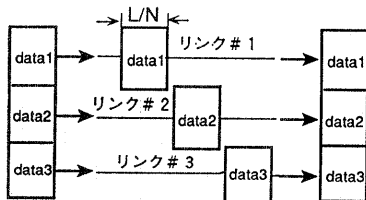


図4 $L_0 \times N < L \leq l_{\max} \times N$ で $N=3$ の場合
Fig. 4 Data transfer algorithm in the case of $L_0 \times N < L \leq l_{\max} \times N$ and $N=3$.

を求めることは困難である。よって、付録Cに示すように近似的に $n=L/l_{\max}$ を解とする。したがって、データ転送アルゴリズムは、(1)-(c)と同様に上位レイヤから転送要求のあったデータ単位を l_{\max} に分割し、先頭データから順次、空きリンクに割り当て、データ分割数が N を超える場合は、その分については転送が終了したリンクに順次割り当てる方式となる。

3.3 提案方式 (その2)

N 本のリンクのデータ転送速度が異なる場合について、高効率データ転送方式を示す。

$v=(v_1, \dots, v_N)$ (bps) を各リンクの転送速度とする。 v_1, \dots, v_N は転送速度の順とする。

$l=(l_1, \dots, l_n)$ (KByte) を各リンクの転送データ長とする。 l_1, \dots, l_n は、速度 v_1 のリンクから順に対応させ、 N を超える分は速度 v_1 のリンクから再度対応させる。各リンクの最大転送データ長は各リンク同一とし、 l_{\max} とする。このとき、長さ L のデータの転送時間 T_c を最小とする l, n は問題4の解として与えられる。

問題4

$$\min_{n \in I, l} q(f(l) + T_h) + nT_p \quad (7. a)$$

subj. to

$$f(l) = \min_l \max_{\substack{i \in \{1, \dots, n\} \\ j = \text{mod}(N, i-1) + 1}} (l_i/v_j) \quad (7. b)$$

subj. to

$$0 < l_i \leq l_{\max} \quad (i=1, \dots, n) \quad (7. c)$$

$$\sum_{i=1}^n l_i = L \quad (7. d)$$

ここで、 $\text{mod}(N, i)$ は i を N で割ったときの余りである。式(7. a)は式(1. a)と同様の意味である。 $f(l)$ はリンク上の転送処理時間である。これは、制約条件(7. c), (7. d)のもとで転送時間が最大のリンクの転

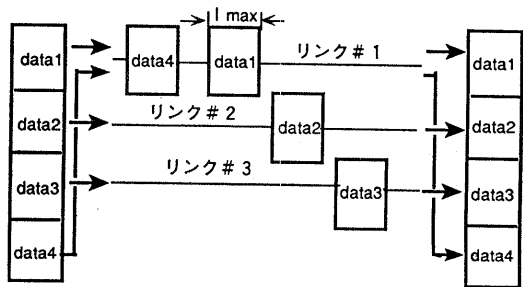


図5 $L > l_{\max} \times N$ で $N=3$ の場合
Fig. 5 Data transfer algorithm in the case of $L > l_{\max} \times N$ and $N=3$.

送時間を最小とする l に対する目的関数値である。このことは、転送時間が最大のリンクの転送時間が全体のリンクの転送時間を決めることを意味する。なお、問題4では $n > N$ の場合、問題を簡単にするため、近似的に(7. b)~(7. d)では N を超える分のリンクも存在するとし、(7. a)でリンクの転送時間にラップラウンド回数 $q (=n/N)$ を乗ずることとした。

問題4の解法として、本論文では n を1から順に与え、それぞれの n に対して、式(7. b)~(7. d)を解き、 T_c が最小となる n を求める方式を採用する。パラレルリンク方式の実際の適用では n は大きくても10~20程度と考えられるため、この方式でも実用上差し支えない。式(7. b)~(7. d)は、問題5と等価である⁹⁾。

問題5

$$f(l) = \min_{l, \varepsilon} \varepsilon \quad (8. a)$$

subj. to

$$l_i/v_j \leq \varepsilon \quad (8. b)$$

$$0 < l_i \leq l_{\max} \quad (8. c)$$

$$i=1, \dots, n \quad j = \text{mod}(N, i-1) + 1$$

$$\sum_{i=1}^n l_i = L \quad (8. d)$$

問題5は線形計画問題であり、容易に解くことができる。なお、問題5を解くにあたり、与えられた n に対し変数 v の要素 v_j と変数 l の要素 l_i とが1対1に対応づくように変数 v を $v_j = v_{\text{mod}(N, i-1) + 1}$ で表される変数 $v' = (v'_1, \dots, v'_n)$ に置き換える。

以上より、データ転送アルゴリズムをまとめると以下のようなになる。

(1) 問題4を解き、 T_c を最小とする l, n を求める。

(2) 上位からの転送データを先頭から長さ l_1, \dots, l_n に分割し、転送速度 v_1, \dots, v_n のリンクに順次割り当てる。転送データ数が N を超える分については対応するリンクのデータ転送が終了しだい、転送データを割り当てる。

本データ転送方式を実現する場合、データ転送ごと問題4を解くことは処理オーバーヘッドの点から好ましくない。そこで、 L に対する最適 l, n をあらかじめ計算し、その結果をテーブル化しておき、データ単位の転送時にこのテーブルを検索する方法が考えられる。テーブル化すべき L の範囲は、使用する上位プログラムから予測し、もし、テーブルに記載の範囲外のデータ長のデータを転送するときは、例えば、以

下のような簡略化したアルゴリズムを用いることができる。

$$(a) \quad L \leq l_{\max} \times N$$

上位から転送要求のあったデータ単位を n 分割し、先頭データから順次、空きリンクに割り当てる。

$$(b) \quad L > l_{\max} \times N$$

上位から転送要求のあったデータ単位を l_{\max} に分割し、先頭データから順次、空きリンクに割り当てる。データ分割数が N を超える分については転送が終了したリンクに順次割り当てる。

4. 転送時間の評価

本論文で提案したデータ転送方式の有効性をまず1つのデータ単位の転送時間に注目して定量的に評価し、次に連続転送を行う場合の評価を定性的に行う。

4.1 リンクの転送速度が同一の場合

(1) 前提条件

2つの通信制御処理装置間を複数のリンクで接続するデータ転送システムのモデルを考える。リンク数 $N=5$ 、リンク上で転送可能な最大データ長 $l_{\max}=4$ Kbyte とする。また、CPUの処理能力を1 MIPS、1つの分割データの転送に必要なソフトウェア処理を4 Kstep (通信制御処理装置では標準的な値) とすると、ソフトウェア処理時間 $T_p=4$ msec となる。なお、データ分割が発生しない場合はその分の処理時間を差し引き $T_p=2$ msec とする。リンク対応のハードウェア処理時間は、ソフトウェア処理時間の約1/10とし、 $T_h=0.5$ msec とする。

(2) 計算結果

(1)の前提条件に対しリンクの転送速度が $V=1$ Mbps, $V=0.2$ Mbps, $V=6$ Mbps の各ケースについて3.2節で示したデータ転送アルゴリズムを適用した結果をそれぞれ表1, 表2, 表3および図6, 図7, 図8に示す。これらには、比較のため、従来のパラレルリンク方式(データ単位を単純にリンク数に分割する方式)でのデータ転送時間とシングルリンクでのデータ転送時間を併記する。

パラレルリンク方式は、全レンジの転送データ長(データ単位の長さ)に対してシングルリンクおよび従来方式と比較して同等か有利である。シングルリンクに対しては分割数が1のときは同等であるがそれ以外では分割数が多くなるほど有利となる。従来方式に対しては $L \leq L_0 \times N$ の範囲で有利である。 $L > L_0 \times N$ の範囲では、データ単位は、 N 分割または l_{\max}

で分割されるため同等になる. このようにパラレルリンク方式はシングルリンク, 従来方式の持つ欠点を解決した有効な方式であるといえる.

次にパラレルリンク方式が有効となるレンジをリンクの転送速度との関連で考察する. 上述のようにパラレルリンク方式は分割数が2以上で $L \leq L_0 \times N$ の範

表 1 データ転送時間の評価 (リンクの転送速度均等, 転送速度=1Mbps)

Table 1 Evaluation of data transfer time (in the case of same link speed, transfer speed =1Mbps).

転送データ長	提案方式		従来方式の転送時間 (msec)	シングルリンクでの転送時間 (msec)
	分割数	転送時間 (msec)		
1KB	1	10.3	22.1	10.3
2KB	2	16.3	23.6	18.1
4KB	3	22.9	26.7	33.7
8KB	4	32.1	33.3	69.5
16KB	5	45.5	45.5	141.0
32KB	8	95.5	95.5	284.0

表 2 データ転送時間の評価 (リンクの転送速度均等, 転送速度=0.2Mbps)

Table 2 Evaluation of data transfer time (in the case of same link speed, transfer speed =0.2Mbps).

転送データ長	提案方式		従来方式の転送時間 (msec)	シングルリンクでの転送時間 (msec)
	分割数	転送時間 (msec)		
1KB	3	26.2	28.7	43.5
2KB	5	36.9	36.9	84.4
4KB	5	53.3	53.3	166.3
8KB	5	86.0	86.0	332.7
16KB	5	151.6	151.6	665.4
32KB	8	197.8	197.8	1330.7

表 3 データ転送時間の評価 (リンクの転送速度不均等, 転送速度=6Mbps)

Table 3 Evaluation of data transfer time (in the case of different link speed, transfer speed =6Mbps).

転送データ長	提案方式		従来方式の転送時間 (msec)	シングルリンクでの転送時間 (msec)
	分割数	転送時間 (msec)		
1KB	1	3.9	20.7	3.9
2KB	1	5.2	21.1	5.2
4KB	1	8.0	21.6	8.0
8KB	2	14.0	22.7	43.2
16KB	4	22.0	24.9	86.4
32KB	8	43.7	43.7	172.7

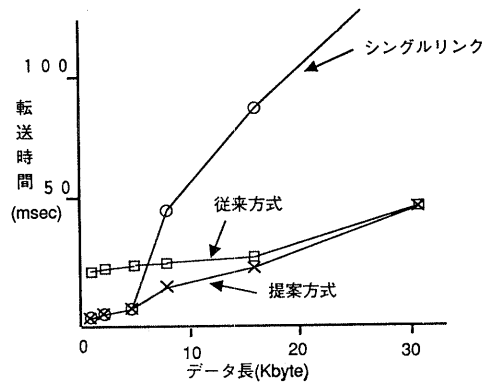


図 6 リンクの転送速度均等の場合のデータ転送時間 (転送速度=1Mbps)

Fig. 6 Data transfer time in the case of same link speed, transfer speed=1Mbps.

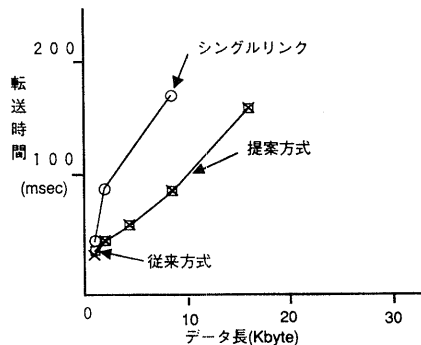


図 7 リンクの転送速度均等の場合のデータ転送時間 (転送速度=0.2Mbps)

Fig. 7 Data transfer time in the case of same link speed, transfer speed=0.2Mbps.

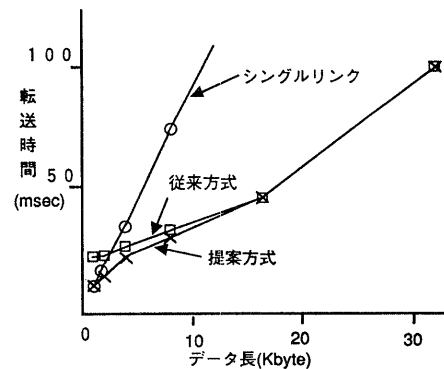


図 8 リンクの転送速度均等の場合のデータ転送時間 (転送速度=6Mbps)

Fig. 8 Data transfer time in the case of same link speed, transfer speed=6Mbps.

冊でシングルリンク、従来方式に対して有効である。本例ではこのレンジに属する分割数2の場合を考える。この場合、リンクの転送速度が0.2Mbpsのとき $L=0.5$ KB, 1Mbpsのとき $L=2$ KB, 6Mbpsのとき $L=8$ KBに対応する。このようにリンクの転送速度が速いほどパラレルリンク方式が有効な転送データ長（データ単位の長さ）は長くなる（本例ではリンクの転送速度が5倍になるとデータ長は4倍ほど長くなる）。また、特定のデータ長に対し、最適なデータ分割数とリンクの転送速度との関係を考察する。 $L=2$ KBの場合、0.2Mbpsのとき最適分割数は5, 1Mbpsのとき最適分割数は2, 6Mbpsのとき最適分割数は1となる。このようにリンクの転送速度が速いほど最適な分割数は小さくなる（本例ではリンクの転送速度が5倍になると最適な分割数は約半分となる）。

以上からパラレルリンク方式の性能を引き出そうとする場合、リンクの転送速度が与えられていれば、対象とするシステムで最も性能を引き出したいデータ長のレンジからリンクの本数を決定することになる。

4.2 リンクのデータ転送速度が異なる場合

(1) 前提条件

リンクの転送速度を $v_1=1.5$ Mbps, $v_2=1.25$ Mbps, $v_3=1$ Mbps, $v_4=0.75$ Mbps, $v_5=0.5$ Mbps とする。他の条件は4.1節と同様とする。

(2) 計算結果

(1)の前提条件に対して、3.3節で示したデータ転送アルゴリズムを適用した結果を表4および図9に示す。これらには、比較のため、従来のパラレルリンク方式（データ単位を単純にリンク数に分割する方式）でのデータ転送時間とシングルリンクでのデータ転送時間（速度 v_1 のリンクを用いた場合）を併記する。本ケースでは提案方式は、従来方式、シングルリンクに対して、すべての場合で有利である。リンクの転送速度が異なる場合はケースが多岐に渡るため詳細な定

表4 データ転送時間の評価（リンクの転送速度不均等）
Table 4 Evaluation of data transfer time (in the case of different link speed).

転送データ長	提案方式		従来方式の転送時間 (msec)	シングルリンクでの転送時間 (msec)
	分割数	転送時間 (msec)		
1KB	1	7.71	23.6	7.71
2KB	2	14.2	26.7	12.9
4KB	2	20.0	33.0	23.3
8KB	3	29.2	45.5	44.2
16KB	4	44.3	70.5	85.8

量的考察は割愛するが、提案方式は速度のパラッキが大きいほど従来方式に対してより高い効果があると考えられる。

4.3 複数データの連続転送を行う場合の考察

複数データを連続転送する場合の評価指標としてスループット（全転送データ長/すべてのデータの転送に要した時間）を用いる。

(1) 従来方式との比較

複数のデータ単位を連続転送するために要する時間は従来方式、提案方式とも転送時間の和になる。よって、スループットも提案方式が従来方式より有利となることは4.1, 4.2節の考察から明らかである。

(2) マルチリンク方式との比較

各リンクで転送可能なデータ長に制限がないとき、各データの分割、組立て処理のない分、複数データ単位の連続転送ではマルチリンク方式の方が有利である。しかし、一般にはリンク上で転送可能なデータ長には制限があり、マルチリンク方式でも上位からの転送データ単位がこの制限を越える場合は1つのリンク上を分割したデータを順次転送する必要がある。この場合、マルチリンク方式は本提案のパラレルリンク方式と同等かもしくは緒言で述べた理由から劣る場合がある。

本提案方式は本来の適用の目的とする応答時間を重視するアプリケーションでの有効性は明らかであるが（この場合、マルチリンク方式はシングルリンクの効果しかない）、上述のようにリンク上で転送可能なデータ長に制限があり、データ分割が発生する場合（ファイル転送、画像転送等比較的長いデータを扱う場合）ではスループットが重要なアプリケーションに対しても有効に働く。

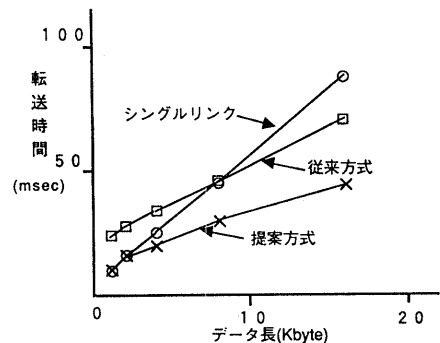


図9 リンクの転送速度不均等の場合のデータ転送時間
Fig. 9 Data transfer time in the case of different link speed.

5. 結 言

本論文では、複数のリンクを用いたデータ転送方式として、パラレルリンク方式を対象とし、データ転送時間の最小化を考慮したデータ転送方式を提案した。データ転送方式は、リンクのデータ転送速度が同一の場合と異なる場合のそれぞれについて提案した。また、提案方式を典型的なデータ転送システムのモデル上で定量的な評価を行った。この結果、提案方式は、従来方式に対して、リンクの転送速度が同一の場合で最大 50%、リンクの転送速度が異なる場合で 60%~35% の転送時間の改善が図れることを示した。

本論文で提案した高効率データ転送方式は、CPU上のソフトウェア処理時間とリンク上の転送時間のトレードオフに意味のある範囲で有効に働く。ソフトウェア処理時間がリンクの転送時間に対して極端に大きい場合は複数のリンクを用いること自体に意味がないし、その逆の場合は、データ単位を単純にリンク数に分割するだけで十分である。したがって、本論文で提案した高効率データ転送方式を適用しその効果を得ようとする場合は、対象とするシステムに対する事前評価が必要である。しかし、本提案方式はシングルリンクやデータ単位を単純にリンク数に分割する方式に比べてどのような場合でも劣ることはなく、また適用例でも示したように、一般的なデータ転送システムでは大部分の場合、本提案方式が有効な範囲のものであり、本提案方式は広く適用できるものである。また、連続転送の場合のマルチリンク方式との比較ではリンク上で転送可能なデータ長の制限によりデータ分割が発生する場合、本方式は有効である。

本論文で示したデータ転送方式は、通信ネットワークにおけるデータ転送、計算機の周辺装置との入出力におけるデータ転送等、特に応答時間を重視するアプリケーションを中心に幅広い範囲に適用できる。

謝辞 日頃ご指導いただき、本研究においては有益なご討論、ご助言をいただいた日立製作所システム開発研究所および同ソフトウェア開発本部の関係者の方々に感謝致します。

参 考 文 献

- 1) 高橋 修: 3.3 データリンク層, 情報処理, Vol. 26, No. 4, pp. 315-320 (1985).
- 2) 西園敏弘: 異速度回線からなるマルチリンクパケット伝送方式の評価, 信学論, Vol. J6 a-B, No. 12, pp. 1647-1655 (1986).

- 3) 篠崎郁生, ほか: ネットワーク間接続のソフトウェアの開発, 大学内ネットワークの相互接続の諸問題シンポジウム, pp. 51-57 (1990).
- 4) 西園敏弘, ほか: パケット交換におけるマルチリンク制御, 情報処理学会分散処理システム研究会資料, DP 16-4 (1982).
- 5) 吉田, ほか: M/M/2 待ち行列系の順序待ち特性, 信学論 (B), Vol. J66-B, No. 7, pp. 926-927 (1983).
- 6) 西園敏弘, ほか: マルチリンクパケット伝送方式の解析, 信学論(B), Vol. J67-B, No. 1, pp. 47-53 (1984).
- 7) 西園敏弘, ほか: ランダム分配によるマルチリンクパケット伝送方式の解析, 信学論(B), Vol. J67-B, No. 10, pp. 1148-1154 (1984).
- 8) 西園敏弘, ほか: マルチリンクパケット伝送方式の評価, 信学論(B), Vol. J68-B, No. 10, pp. 1112-1118 (1985).
- 9) 志水清孝, 相吉英太郎: 数理計画法, 昭晃堂 (1984).

付 録 A

$n=N$ の場合の T_c の値は, $q=1$ であるため,

$$T_c = L/(VN) + T_h + NT_p \quad (9)$$

となる。 $n > N$ に対しては, $q \geq 2$ であり, T_c の値は,

$$T_c = q(L/(Vn) + T_h) + nT_p \quad (10. a)$$

$$(q-1)N < n \leq qN \quad (10. b)$$

である。式(9), (10)の各項を比較すると, $n > N$ に対して, 式(10)の値は式(9)の値より常に大きい。よって, 問題3の解は $n=N$ である。

付 録 B

定義から

$$q-1 < n/N \leq q \quad (11)$$

式(11)から $q \geq 2$ の場合, T_c は

$$n/N(L/(Vn) + T_h) + nT_p \leq T_c$$

$$< (n/N+1)(L/(Vn) + T_h) + nT_p \quad (12)$$

である。 T_c の最小値は式(12)の左側の不等式の左辺から

$$L/(NV) + nT_h/N + nT_p \quad (13)$$

となり, n の正勾配の一次関数である。

一方, T_c の最大値は式(12)の右側の不等式の右辺は微分可能関数であることから

$$dT_c/dn = -L/(Vn^2) + T_p + T_h/N = 0 \quad (14)$$

なる最適性条件を満たす。式(14)を n について解くと,

$$n = \sqrt{L/V(T_h/N + T_p)} \quad (15)$$

となる。これより, T_c の最大値はある最小値を持つ

下に凸な関数であることがわかる。また、ここでは $n > N$, $N \leq L/l_{\max}$ であるため、以下がいえる。

(1) $L/l_{\max} \geq \sqrt{L/V(T_h/N + T_p)}$ の場合

T_c の最大値, 最小値とも単調増加であるため, 制約式(6. c)から $n = L/l_{\max}$ が問題3の解となる。

(2) $L/l_{\max} < \sqrt{L/V(T_h/N + T_p)}$ の場合

この場合, T_c を最小とする n は

$$L/l_{\max} \leq n \leq \sqrt{L/V(T_h/N + T_p)} \quad (16)$$

を満たす。ここでは正確な解を求めることは困難であること, また実現容易性のため場合分けを減らす意味で, $n = L/l_{\max}$ を近似的に問題3の解とする。

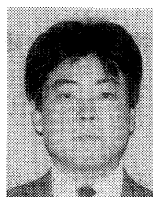
(1), (2)から $n = L/l_{\max}$ を近似的に問題3の解とする。

付録 C

L_0 に対する n を n' とすると, $L_0 \geq l_{\max}$, $L_0 = L/n'$ から $n' \leq L/l_{\max}$ である。よって, $n \geq L/l_{\max}$ から T_c は $n \leq N$ ($n' \leq N$ の場合) で単調増加であるため $n = L/l_{\max}$ が問題1の解である。また, $n > N$ では付録Bの結果から $n = L/l_{\max}$ を問題1の近似解とみなす。

(平成4年9月22日受付)

(平成6年2月17日採録)



平田 俊明 (正会員)

昭和36年生。昭和58年慶應義塾大学工学部計測工学科卒業。昭和60年同大学院修士課程修了。同年(株)日立製作所入社。現在, 同社システム開発研究所勤務。平成4年から慶應義塾大学環境情報学部非常勤講師兼任。計算機通信ソフトウェアに関する研究開発に従事。電子情報通信学会会員。



近藤 毅 (正会員)

昭和39年生。昭和61年千葉大学理学部物理学科卒業。昭和63年同大学院修士課程修了。同年(株)日立製作所入社。以来, 計算機用通信ソフトウェアの性能評価およびその設計, 無線LANシステムなどの研究開発に従事。現在, 同社システム開発研究所勤務。電子情報通信学会, 日本物理学会各会員。



宮崎 聡 (正会員)

昭和30年生。昭和53年京都大学工学部電気工学第二科卒業。昭和55年同大学院修士課程修了。同年(株)日立製作所入社。現在, 同社システム開発研究所にてネットワーク管理, 運用, 設計支援技術に関する研究開発に従事。工学博士。電子情報通信学会, IEEE, ACM 各会員。