

超並列相互結合網におけるバッファ制御方式の改良

沼 寿隆[†] 坂井 修一^{††} 朴 泰祐[†]

[†] 筑波大学 電子・情報工学系 ^{††} 東京大学 工学系研究科 電気工学専攻

相互結合網の転送性能を向上させる方法として、(1) ネットポロジを工夫する、(2) 網内のバッファ容量を増やす、(3) 適応ルーティングを行なう、(4) virtual cut-through 方式を用いる、などがある。本稿では、これらとは直交する方法として、バッファの制御方式を改良することを提案し、シミュレーション評価によってこれを検証する。改良は、メッセージの出力順序を FIFO ではなく出力できる順番にする、バッファを多ポート化する、という2点に関して、virtual cut-through の結合網の上で行なった。シミュレーションの結果、これらの改良によって結合網の転送性能が大幅に向上することが確認された。

Advanced Buffer Control Schemes in Massively Parallel Interconnection Networks

Hisataka NUMA[†] Shuichi SAKAI^{††} Taisuke BOKU[†]

[†] Institute of Information Sciences and Electronics, University of Tsukuba

^{††} Department of Electrical Engineering, The University of Tokyo

So as to improve performance of interconnection networks, the followings have been considered: (1) to optimize network topologies, (2) to enlarge buffer in the network, (3) to adopt adaptive routing and (4) to adopt virtual cut-through routing. Orthogonal to these ideas, improvements of the buffer controlling methods are here proposed and examined by simulations. The improvements are: to output a transferable packet first, not in FIFO order, and to provide multiple output port for a buffer. These improvements are designed on the virtual cut-through network and the simulation results show that they considerably gain performance of interconnection networks.

1 はじめに

並列計算機の構成を考える上で、相互結合網は必要不可欠な要素であり、システム全体の処理性能は相互結合網の転送性能に大きく左右される。ネットワーク・トポロジは転送性能に最も大きな影響を与える要素であり、これまでに数多くが提案され、実用化されている。また、メッセージの転送方式やルーティング・アルゴリズムは、ネットワーク・トポロジと同様に相互結合網の転送性能を決定する上で重要な要素であり、これらを改良することによって転送性能の向上を図ることができる。

現在までに様々な結合網が提案され、実現可能な規模及び利用可能な技術に沿って実用化されてきた。

中でもメッシュ網を基本とする結合網は、現在並列計算機で最も普遍的に実装されており、特に近接作用を中心とした計算に有効であることが知られている。他方、ハイバクロスバ網、MDCE 網といったランダム転送に強い結合網も研究・実装されている。これらは、直径が小さく通信チャネルが多いため、複雑な転送パターンであっても高い転送性能を保つことができる。後者の研究例として、ハイバクロスバ網に virtual channel を用いた適応ルーティングを導入したり、virtual cut-through 方式を導入することによって、同結合網の転送性能を向上させることができると分かっている [1] [2]。

しかし、これまでに提案された方式はそのほとん

どがスイッチのバッファ制御に FIFO (first-in first-out) 方式を用いていたため、バッファが溢れやすく、チャネルの利用効率が悪いという欠点があった。そこで本稿では、上記に述べた方法とは直交する方法として、バッファの制御方式を改良することを提案し、シミュレーション評価によってこれを検証する。その改良は、メッセージの出力順序を FIFO ではなく出力できる順番にする、バッファを多ポート化する、という2点に関して、virtual cut-through の結合網の上で行なう。なお、シミュレーション評価は、2次元メッシュ網と3次元ハイパクロスバ網に適用した場合について行なう。

2 パケット交換スイッチにおけるバッファ制御方式の改良

まず最初に、virtual cut-through 方式における従来のバッファ制御方式であった FIFO 方式の問題点について述べ、次にこの FIFO 方式を改良した2つの制御方式を提案する。

2.1 FIFO 方式の問題点

ここでは、図1に示すように入力ポート側にバッファを持つスイッチを仮定する。virtual cut-through 方式 [3] では、メッセージは衝突が起これなければ wormhole 式に転送され、メッセージ同士の衝突などにより転送をブロックされた場合には、そのスイッチに設けられているバッファに一時的にストアされる。このようなストア動作を行なうことによって、メッセージの後続部分が他のメッセージと衝突することを防ぎ、wormhole 方式に比べて、全体としてのブロック率を低減させることができる。ただし、入力ポートのバッファは FIFO 式に動作するため、そのポートのバッファ中に複数のメッセージが存在する場合、最も先にストアされたメッセージだけが出力権を持つことになる。つまり、図2に示すように出力できるのはバッファの先頭にあるメッセージだけであり、それ以外のメッセージは要求している出力ポートが空いていたとしても先頭のメッセージを追い越して行くことはできない。

実際にスイッチ上でパケット同士の衝突が起これると、ブロックされた方のメッセージは入力ポートのバッファにストアされ、出力ポートが利用可能になるまで待機することになる。このようなブロックが続くとバッファに空きがなくなり、それが前段のスイッチにまで影響を及ぼす。また、出力ポートの利

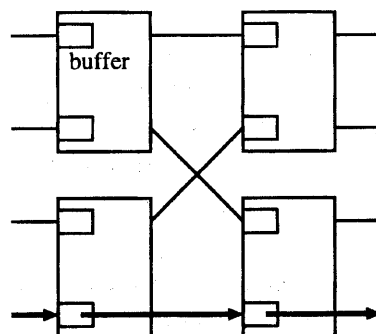


図1: パケット交換スイッチ

用効率が悪いために、何も出力されずに遊んでしまうポートが多くなる。そのため、バッファの制御方式次第で転送性能が著しく悪化してしまう。

2.2 バイパス制御の導入

前節の欠点を解消するため、以下で FIFO 方式を改良した制御方式を提案する。

図3に示すように、バッファにストアされている先頭のメッセージが出力できない場合でも、空いている出力ポートを要求している後続メッセージをバッファから取り出せるようにする。つまり、後ろのメッセージが前のメッセージを追い越すことを可能にするバイパス制御を導入する。この方式を bypass-single 方式（以下 single 方式と省略）と呼ぶことにする。この方式を採用することによって、FIFO 方式に比べて出力ポートの利用率を大幅に改善できると考えられる。

2.3 バッファの多ポート化

FIFO 方式や single 方式では一つのバッファからは一つのメッセージしか取り出せない。そこで、次にこの点を改良した方式を提案する。先ほどのバイパス制御に加えて、図4に示すように、バッファからの出力を多重化し、一つのバッファから複数のメッセージを出力可能にする。この方式を bypass-multi 方式（以下 multi 方式と省略）と呼ぶことにする。例えば、あるスイッチの特定のバッファにだけメッセージが複数ストアされている状況を仮定する。すると、もしそれぞれのメッセージの要求している出力ポートが異なっていれば、それらが同時に流れていくことが可能になる。これにより、single 方式以上の性能

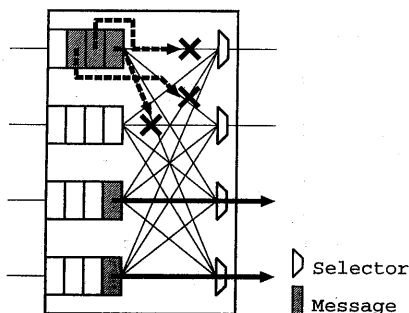


図 2: FIFO 方式

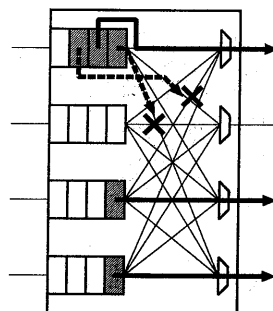


図 3: bypass-single 方式

向上が期待できる。

また, multi 方式は集中制御型の構造化バッファとは異なり, バッファの入力ストリームが一つで済むため, これに比べて実現が容易であると思われる。

3 評価

3.1 シミュレーション環境と仮定

第2節で提案した2つの制御方式の性能評価を, 計算機シミュレーションによって行なう。シミュレーションには汎用相互結合網シミュレータ生成システム INSPIRE [4] を用いる。INSPIRE はシミュレーション・エンジンの都合上メッセージの調停に FGFS (first-generated first-service) 方式を採用しているため, 複数のメッセージによって出力ポートの競合が起こった場合には, それらの中で最も先に生成されたものが優先される。

評価は, メッセージ長を固定してバッファ容量を変化させた場合と, バッファ容量を固定してメッセージ長を変化させた場合の2通りについて行なう。

シミュレーションの条件を以下に示す。

- システムは 1024PU 構成。
- メッセージの転送方式は virtual cut-through 方式。virtual cut-through 方式には複数のモデルがあるが, 今回は次のようなモデルを仮定する。
 - ー メッセージが次のノードへ進むためには, そのノードのバッファにメッセージ全体をストアさせるだけの空き容量がなければならない。
 - ー バッファにストアしている際中でも, 出力ポートが空けば出力を開始する。
- 転送パターンは一様ランダム転送。

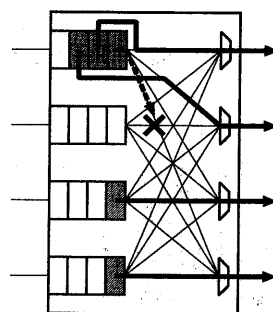


図 4: bypass-multi 方式

- PU に到着したメッセージは明示的な受信処理無しに, 適宜受信されるものとする。また, 送信処理と受信処理を同時にできるものとする。
- チャネルのバンド幅を 1[flit/clock] に正規化し, メッセージ長を flit 単位で表わす。
- シミュレーション時間は, 10,000[clock]。

以上の条件の下, 2次元メッシュ網 (32×32) 及び 3次元ハイパクロスバ網 (8×8×16) に FIFO, single, multi の3通りの制御方式を適用し, 性能評価を行なう。なお, ルーティングは次元オーダーによる固定ルーティングとする。

評価指標には, 正規化スループットと平均レイテンシを用いる。1PU 当たりのメッセージ受信量を最大理論スループットを 1 として正規化したものが正規化スループットであり, 平均レイテンシはメッセージが生成されてから受信 PU に到着するまでの時間を表わす。

3.2 バッファ容量の影響

メッセージ長を固定した場合の評価結果を図5及

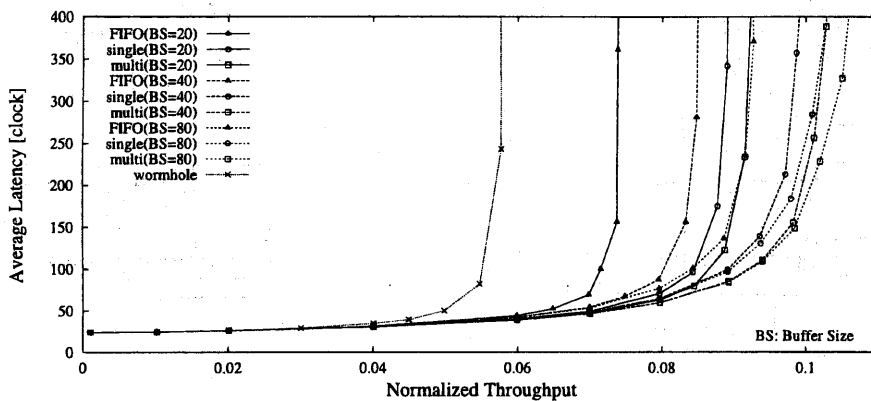


図 5: メッシュ網における転送性能 (メッセージ長を固定)

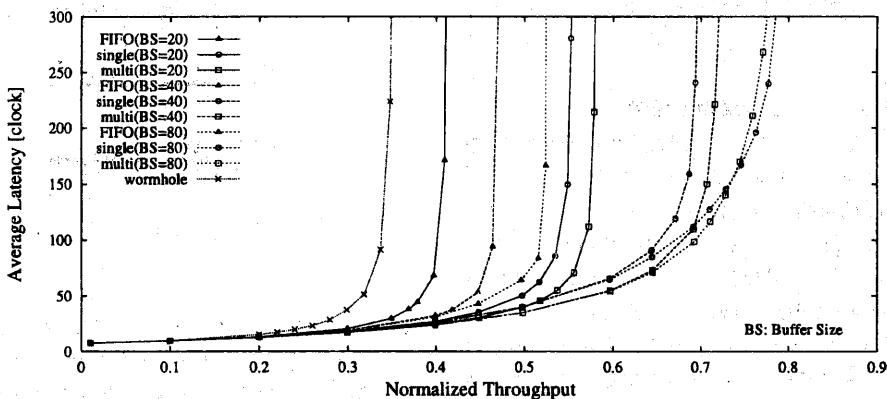


図 6: ハイパクロスバ網における転送性能 (メッセージ長を固定)

表 1: メッシュ網における評価結果

	BS=20	BS=40	BS=80
bypass-single	25%	27%	24%
bypass-multi	30%	32%	29%

(BS: Buffer Size)

表 2: ハイパクロスバ網における評価結果

	BS=20	BS=40	BS=80
bypass-single	30%	45%	48%
bypass-multi	37%	51%	47%

(BS: Buffer Size)

び図6に示す。メッセージ長は10[flit]に固定し、バッファ容量を20[flit], 40[flit], 80[flit]の3通りに変化させている。グラフの横軸が正規化スループット、縦軸が平均レイテンシである。また、プロットされた点が‘△’のものはFIFO方式, ‘○’はsingle方式, ‘□’はmulti方式をそれぞれ表わしている。‘()’の中の値はバッファ容量 (BS: Buffer Size) を表わし、そのバッファ容量によって線種を分けている。比較のため, wormhole方式 (‘wormhole’) の結果も載せている。さらに、これらの結果をまとめたものを表1及び表2に示す。これらの表は、各々のバッファ容量において、FIFO方式と比較した場合のsingle方式とmulti方式の最大スループットの増加割合を%で

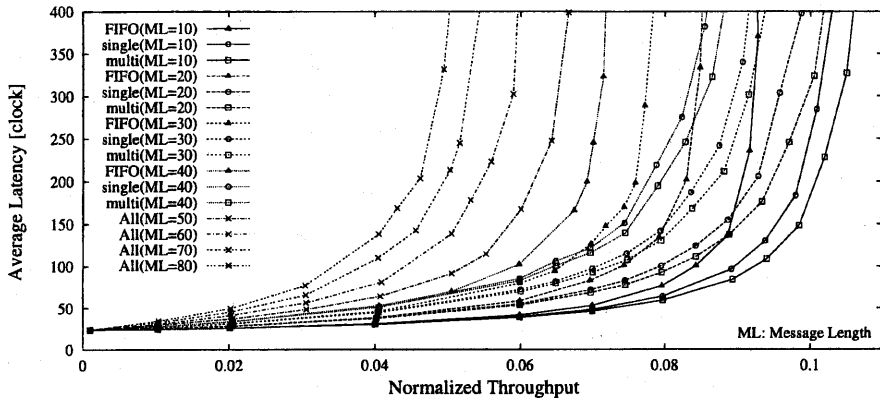


図 7: メッシュ網における転送性能 (バッファ容量を固定)

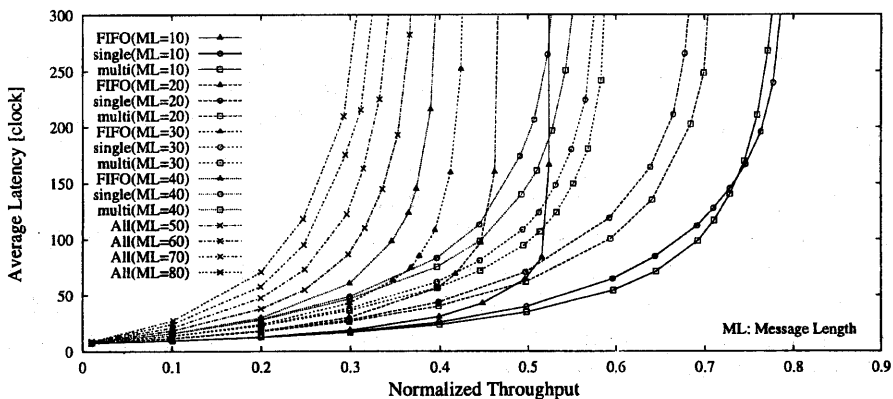


図 8: ハイバクロスバ網における転送性能 (バッファ容量を固定)

表わしている。

全体の傾向として、バッファ容量がメッセージ長の数倍あれば性能が大幅に向上することが分かる。また、ネットワークが空いている状態では制御方式の違いによる性能差は無いが、網が混雑してくると、FIFO 方式に比べて single 方式と multi 方式では転送性能が大幅に向上している。しかし、single 方式と multi 方式には性能差があまり無い。この 2 つの方式の性能が微差である理由を次のように考える。例えば、スイッチの特定のバッファにだけメッセージが溜り、それ以外のバッファにはメッセージが無いという状態であれば、複数出力可能である利点が活き、multi 方式が single 方式よりも性能が大幅に向上

するはずである。しかし、ランダム転送ではこのような状態は起こりにくいため、2 方式の性能にあまり差が無かったのではないかと考察される。

また、ハイバクロスバ網においてバッファ容量が 80[flit] の場合だけ、他の場合と結果が異なっている。つまり、single 方式の性能が multi 方式のそれを逆転している。このような現象の原因については目下検討中であるが、この現象はメッシュ網では起こらず、ハイバクロスバ網において、メッセージ長に対してバッファ容量が大きい場合に起こっている。したがって、一般にメッセージ長に対してバッファ容量が大きい場合にこの現象が起こるという分けではなく、ハイバクロスバ網ではこのような現象が起こる

ということである。

以上より、従来の制御方式である FIFO 方式に比べて大幅な性能向上が得られることが分かり、ここで述べた改良が有効であることが分かった。また、クロック当たりのスループットという点で、single 方式より multi 方式の方が僅かに性能が良いという結果が得られたが、この2方式については、ハードウェアの複雑さの点から、動作周波数に差が生じることが考えられる。しかし、multi 方式は1つのバッファから複数のメッセージを出力させる必要があるため、バッファの出力スループットを物理リンクのその数倍用意しなければならず、single 方式に比べ実装が難しいと考えられる。よって、この程度の性能差であれば、single 方式を採用するのが妥当と言える。一方、FIFO 方式と single 方式の比較については、詳細な回路設計をした上でのさらなる検討が必要となるが、性能向上の大きさから考えてみると、single 方式は有望であると思われる。

3.3 メッセージ長の影響

次に、バッファ容量を 80[flit] に固定し、メッセージ長を変化させてみた。メッセージ長は 10[flit] から 80[flit] まで 10[flit] 刻みに変化させている。その結果を図 7 及び図 8 に示す。グラフの横軸、縦軸、及び点種は先のグラフと同じだが、今度は「()」の中の値がメッセージ長 (ML: Message Length) を示している。また、グラフ中の「All」は、FIFO 方式、single 方式、multi 方式の結果をまとめて表わしたものである。これは、メッセージ長が 40[flit]、つまりバッファ容量の半分を越えると、バッファにストアできるメッセージは高々1つであり、その場合はパイパス及び複数出力が起らないので3つの制御方式の結果が同じになるからである。

メッセージ長が長くなると性能は低下しているが、それでも single 方式と multi 方式の FIFO 方式に対する優位性は変わらない。しかし、メッセージ長がバッファ容量の半分を越えると一気に性能は低下し、single 方式と multi 方式の性能は FIFO 方式と同じになる。そのため、設計する場合にはこのことに十分留意しなければならない。例えば、メッセージ長が長い場合にパケット分割等が必要になるが、ここに挙げた結合網では、同一ソース及び同一デスティネーションでのパケット追い越しは発生しないため、実現は比較的容易であろう。

4 おわりに

本稿では、相互結合網の転送性能を向上させる方法として、従来のバッファ制御方式である FIFO 方式を改良した bypass-single 方式と bypass-multi 方式を提案した。そして、それらをメッシュ網とハイパクロスバ網に適用し、計算機シミュレーションによって性能評価を行なった。その結果、提案した2方式を適用した場合、FIFO 方式に比べて大幅な性能向上を得ることが分かった。一方、multi 方式と single 方式の性能差はわずかであった。この結果から、提案した2方式のどちらが適当かは、ハードウェアコストを加味した評価を行なってみないと判断できないが、single 方式の方がコストパフォーマンスの点で有利であると思われる。いずれにしても、FIFO 方式に対するこれらの優位性は確認できた。

今回はバッファ容量を段階的に変化させていき、その場合の評価を行なったが、最適なバッファ容量で結合網を構成するために、バッファの利用率に関する評価も必要であると思われる。また、ハードウェアコストを考慮した評価を行なうことも今後の課題である。

謝辞

本研究において、貴重な御助言を頂いた板倉憲一氏をはじめとする筑波大学アーキテクチャ研究室の方々には心から感謝します。なお、本研究の一部は文部省科学研究費（基盤(C)09680323）の補助を受けている。

参考文献

- [1] 朴 泰祐 他「ハイパクロスバ・ネットワークにおける転送性能向上のための手法とその評価」並列処理シンポジウム JSPP'94, 1994.
- [2] 曾根 猛 他「ハイパクロスバ・ネットワークにおける virtual channel の動的選択による適応ルーティング」並列処理シンポジウム JSPP'95, 1995.
- [3] P.Kermani and L.Kleinrock: *Virtual Cut-Through: A New Computer Communication Switching Technique*. Computer Network, Vol.3, No.4, pp.267-286, 1979.
- [4] 原田 智紀 他「並列処理ネットワークのための性能評価用シミュレータ生成系 INSPIRE」情報処理学会研究報告 Vol.95, No.80, pp.65-72, 1995.