

2 次元メッシュ上のマルチキャスト通信における 使用仮想チャネル数の削減

笹 河 良 介^{†1} 佐 藤 真 平^{†1,†2} 吉 瀬 謙 二^{†1}

プロセス技術の進展により、数十から数百ものコアが搭載されるメニーコアプロセッサが実現可能になると考えられる。メニーコアプロセッサにおいて、コア間の通信は Network on Chip (NoC) と呼ばれる通信路を用いておこなわれる。Network on Chip においてもマルチキャスト通信に関する研究が行われている。一般的にマルチキャスト通信では、パケットが到着する受信ノードの順番はルーティングにより決定される。しかし、パケットの到着順をシステムやプログラマが自由に決定できるようにする場合、宛先数に比例して仮想チャネルを用意する必要がある。本稿では、パケットのノード到着順を自由に決定できるマルチキャスト通信における、必要仮想チャネル数を削減できるルーティングを提案する。提案手法により、3 つの受信ノードに対するマルチキャスト通信において、3 本必要であった仮想チャネルを 2 本に削減できることを示す。

Reducing required virtual channels for multicast in 2D Mesh Networks

RYOSUKE SASAKAWA,^{†1} SHIMPEI SATO^{†1,†2}
and KENJI KISE^{†1}

^{†1} 東京工業大学 大学院情報理工学研究所

Graduate School of Information Science and Engineering, Tokyo Institute of Technology

^{†2} 日本学術振興会 特別研究員 (DC2)

Research Fellow of the Japan Society for the Promotion of Science (DC2)

1. はじめに

プロセス技術の進展により、1 チップに複数のコアが集積されるようになった。設計プロセスは依然として微細化傾向にあり、数十から数百ものコアが搭載されるメニーコアプロセッサが実現可能になると考えられる。メニーコアプロセッサにおいて、コア間の通信は Network on Chip (NoC) と呼ばれる通信路を用いておこなわれる¹⁾。

Network on Chip に限らず、マルチキャスト通信に関する研究は従来から行われており²⁾⁻⁵⁾、特に Network on Chip に関しては、2 次元メッシュ上でのマルチキャスト通信に関する研究が多い⁶⁾⁻⁹⁾。1 つのパケットに対して複数の宛先を設けるマルチキャスト通信でも、ユニキャスト通信と同様に、デッドロックが起こらないようなルーティングやフロー制御が求められる。

一般的なマルチキャスト通信において、パケットが到着する受信ノードの順番は、ルーティングにより決定される。しかし、パケットの到着順をシステムやプログラマが自由に決定できるようにする場合、宛先数に比例した仮想チャネルを用意する必要がある。

本稿では、パケットのノード到着順を自由に決定できるマルチキャスト通信において、必要仮想チャネル数を削減できるルーティングを提案する。提案手法により、3 つの受信ノードに対するマルチキャスト通信において、3 本必要であった仮想チャネルを 2 本に削減できることを示す。

2. 2 次元メッシュ上のマルチキャスト通信

2 次元メッシュ上のマルチキャスト通信として、The dual-path multicast routing algorithm および The multi-path multicast routing algorithm⁹⁾ がある。これは図 1 および図 2 に示されているように、送信ノードからの相対位置によって受信ノード群を複数のノード群に分ける。そして、各ノード群に対し 1 つのパケットを用い、適切なルーティングを行い、マルチキャスト通信を実現する。以上のように、各ノード群に対しそれぞれパケットを用意することによって、複数の仮想チャネルを用いずに、デッドロックフリーなルーティングを実現している。

3. 任意のノード順によるマルチキャスト通信

2 章で示したように、The dual-path multicast routing algorithm および The multi-path multicast routing algorithm は、複数の仮想チャネルを用いずに、デッドロックフリーな

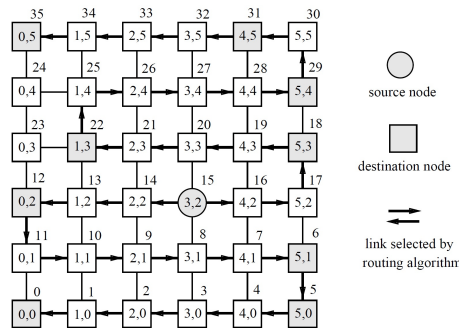


図 1 dual-path multicast の例

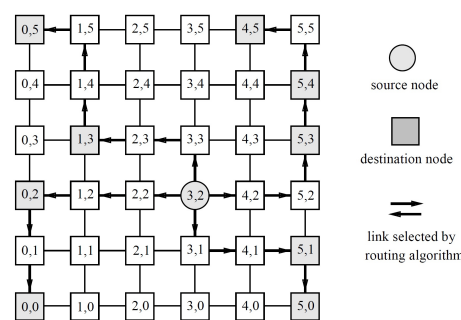


図 2 multi-path multicast の例

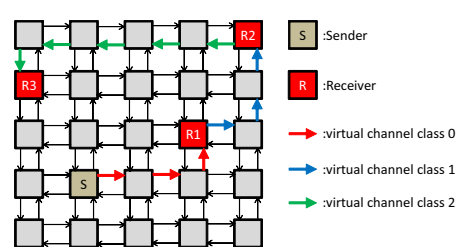


図 3 受信ノードが 3 ノードある場合の例

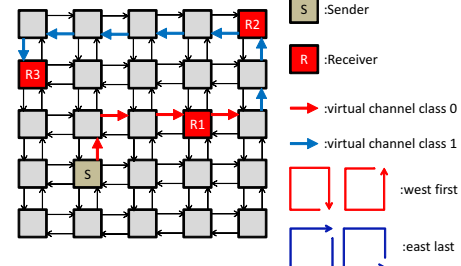


図 4 提案するルーティング方法

ルーティングを実現している。しかし、各受信ノードのパケット到着順はアルゴリズムによって決められており、各受信ノードのパケット到着順をシステムやプログラマが自由に決定することはできない。そこで、各受信ノードのパケット到着順を自由に決定できる、つまり任意のノード順でマルチキャストできる通信を考える。

各受信ノードを任意のノード順で、かつデッドロックフリーに通信できるマルチキャスト通信の実現方法としては、受信ノードが最大 n ノードある場合に仮想チャンネルを n 本用意する方法がある。

受信ノードが最大 3 ノードであり、仮想チャンネルを 3 本用意した場合の例を図 3 に示す。まず、3 本ある仮想チャンネルを 3 つのクラス、class 0、class 1、および class 2 に分ける。そして図 3 にあるように、送信ノード (S) から 1 つ目の受信ノード (R1) までのルーティ

ングに class 0 のチャンネルを利用する。1 つ目の受信ノードから 2 つ目の受信ノード (R2) までのルーティングは class 1 のチャンネルを利用する。同様に 2 つ目の受信ノードから 3 つ目の受信ノード (R3) までのルーティングを class 2 のチャンネルを利用する方法である。図においては各ルーティングには XY 次元順ルーティングを用いている。

図において、受信ノードを通過する前後で、ルーティングが Y 次元方向から X 次元方向へと変化している。これは本来 XY 次元順ルーティングでは許可されていないルーティングである。仮想チャンネルを分けていない場合、このルーティングによりデッドロックが起こる可能性がある。したがって、各受信ノードに到着する度に使用する仮想チャンネルのクラスを変えることで、このようなルーティングをすることができる。よって、受信ノードが最大 n ノードある場合には仮想チャンネルを n 本用意する必要がある。

デッドロックを回避するために仮想チャンネルをクラスに分ける、という考え方は、文献 10) で提案されている The Static Algorithm にも示されている。

4. 提案手法

各受信ノードを任意のノード順で、かつデッドロックフリーであるマルチキャスト通信を実現するためには、 n ノードある受信ノードに対し、仮想チャンネルを n 本用意する必要がある。本稿では、受信ノードが 3 ノードある場合でも、2 本の仮想チャンネルで実現できるルーティングを提案する。

4.1 ルーティング方法

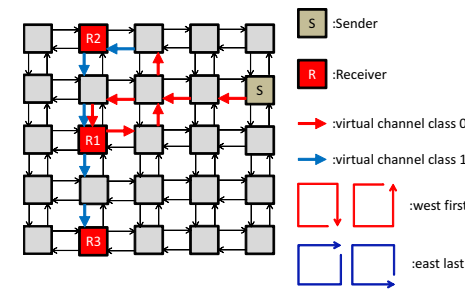
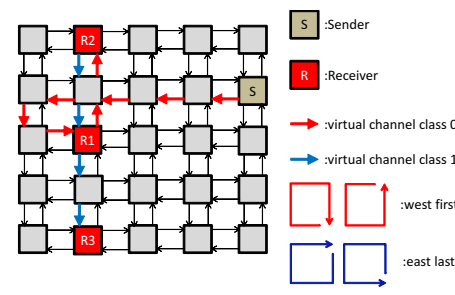
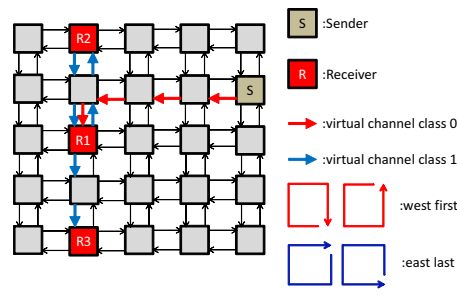
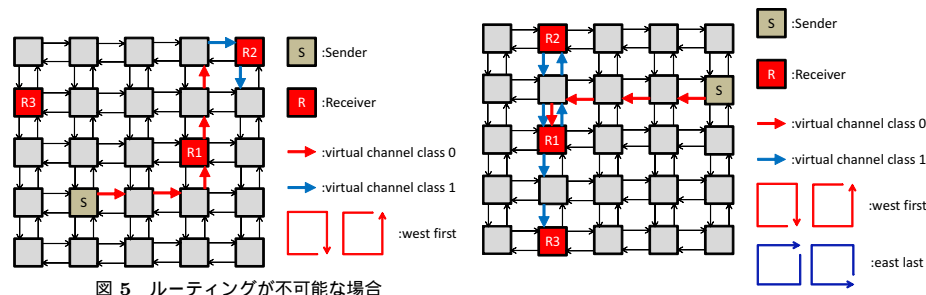
受信ノードが 3 ノードある場合でも、2 本の仮想チャンネルで実現できるルーティング方法を示す。

具体的に図 4 を使って説明する。仮想チャンネルを 2 本用意し、class 0 および class 1 の 2 つのクラスに分ける。今回は class 0 でのルーティングを west first、class 1 でのルーティングを east last とする。図のように、1 つのクラスのチャンネルで複数の受信ノードを通過するようなルーティングを行うことによって、仮想チャンネル 2 本のみでのルーティングを可能にする。

4.2 受信ノードの配置に対するルーティングの分析

受信ノードが 3 ノードある場合、全受信ノードの配置は、

- 送信ノード (S) から見て、1 つ目の受信ノード (R1) が、{ 東、西、南、北、北西、北東、南西、南東 } の合計 8 方向
- 1 つ目の受信ノード (R1) から見て、2 つ目の受信ノード (R2) が 8 方向



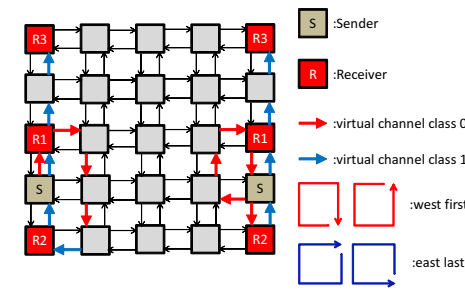
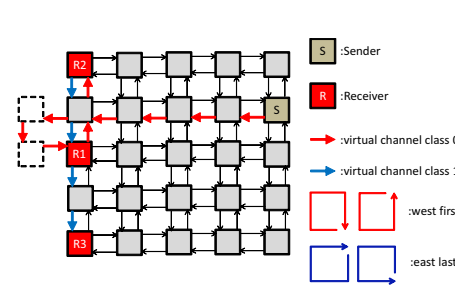
• 2 つ目の受信ノード (R2) から見て、3 つ目の受信ノード (R3) が 8 方向つまり、 $8 \times 8 \times 8 = 512$ 通りの配置が存在する。全ての配置において 4.1 節のようなルーティングが可能かどうかを分析する。

まず、仮想チャネルの各クラスのルーティング方法の組み合わせによっては、ルーティングが不可能であることが分かった。例えば、class 0 および class 1 のルーティング方法を共に west first とした場合、図 5 のように各受信ノードの位置関係が (北東, 北東, 南西) となっている配置に対しては仮想チャネル 2 本ではルーティング不可能である。

しかし、class 0 でのルーティングを west first, class 1 でのルーティングを east last とした組み合わせを採用した場合は、ルーティング不可能な配置は存在せず、全ての配置に対応できていることを確認した。

次に、class 0 でのルーティングを west first, class 1 でのルーティングを east last といった組み合わせを採用した場合でも、各通信が最少ホップ数で実現できない、つまり minimal routing ができない配置が存在することが分かった。例えば、図 6 のようなルーティングでは、R2 の前後で、北方向から南方向への折り返しが発生している。同クラスのチャネル内で、このルーティングが存在してしまうとデッドロックが起きてしまう。したがって、図 7 や図 8 のように、2 ホップ分余分なルーティングが必要となってくる。class 0 でのルーティングを west first, class 1 でのルーティングを east last とした組み合わせを採用した場合、ルーティングは可能だが、各通信が最少ホップ数ではない配置は合計 30 個存在した。

さらに、各ノードが端のノードであった場合、図 7 のようなルーティングでは、図 9 のように 5×5 のノードから外にはみ出てしまう。よって、図 8 のような、例え各ノードが端のノードであっても外にはみ出すことのないルーティングが必要である。今回、各通信が最



少ホップ数で実現できない配置となる 30 個全てについて分析した結果、全ての配置において、各ノードが端のノードであっても外にはみ出さないルーティングが可能であることが分かった。

ただし特別な場合として、各受信ノードの位置関係が (北, 南, 北) (南, 北, 南) とした縦一列、または (東, 西, 東) (西, 東, 西) とした横一列、となる場合は、図 10 のように必ずどちらか片方にはみ出てしまう。

しかし、両方にはみ出ることは無いので、各ノードが端のノードであってもルーティングは可能である。

以上の分析により、class 0 でのルーティングを west first, class 1 でのルーティングを east last とした組み合わせを採用した場合に、図 11 のように分析でき、全配置となる 512 配置全てでルーティングが存在することが確認できた。

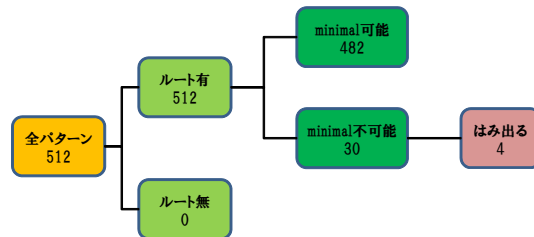


図 11 全 512 配置の分類
(class 0 に west first, class 1 に east last を採用した場合)

以上の結果から、提案するルーティング方法で、仮想チャネル 2 本のみでのマルチキャストが可能であることが示された。

4.3 パケットの送受信の順番

任意のノード順でマルチキャストを行う場合、同じ送受信ノードであっても、パケットが送信された順番とそのパケットが受信される順番が前後する可能性がある。例えば、送信ノード (S) から受信ノード (R3) へのユニキャストと、送信ノード (S) から受信ノード (R1, R2, R3) へのマルチキャストの場合は、ユニキャストが後に送信されても、マルチキャストのパケットより早く受信ノード (R3) に届く可能性がある。この場合、ユニキャストのルーティングをマルチキャストのルーティングと同じものを用いれば、パケットが送信される順番と受信される順番は一致する。しかし、あらゆるノード同士の通信で、パケットが送信される順番と受信される順番が一致するようなルーティングが可能かどうかの検討は今後の課題となる。

5. ま と め

プロセス技術の進展により、数十から数百ものコアが搭載されるメニーコアプロセッサが実現可能になると考えられる。メニーコアプロセッサにおいて、コア間の通信は Network on Chip (NoC) と呼ばれる通信路を用いておこなわれる。

Network on Chip においてもマルチキャスト通信に関する研究が行われている。一般的にマルチキャスト通信では、パケットが到着する受信ノードの順番はルーティングにより決定される。しかし、パケットの到着順をシステムやプログラマが自由に決定できるようにする場合、宛先数に比例して仮想チャネルを用意する必要がある。

本稿では、パケットのノード到着順を自由に決定できるマルチキャスト通信における、必

要仮想チャネル数を削減できるルーティングを提案した。提案手法により、3 つの受信ノードに対するマルチキャスト通信において、3 本必要であった仮想チャネルを 2 本に削減できることを示した。

参 考 文 献

- 1) Dally, W.J. and Towles, B.: Route packets, not wires: on-chip interconnection networks, *Proceedings of the Design Automation Conference*, pp.684–689 (2001).
- 2) Lin, X. and Ni, L.M.: Deadlock-free multicast wormhole routing in multicomputer networks, *Proceedings of the 18th annual international symposium on Computer architecture*, ISCA '91, New York, NY, USA, ACM, pp.116–125 (online), DOI:http://doi.acm.org/10.1145/115952.115965 (1991).
- 3) Lin, X., McKinley, P.K. and Esfahanian, A.-H.: Adaptive Multicast Wormhole Routing in 2D Mesh Multicomputers, *PARLE*, pp.228–241 (1993).
- 4) Lin, X., McKinley, P.K. and Ni, L.M.: Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers, *IEEE Trans. Parallel Distrib. Syst.*, Vol.5, pp.793–804 (online), DOI:http://dx.doi.org/10.1109/71.298203 (1994).
- 5) Duato, J.: A Theory of Deadlock-Free Adaptive Multicast Routing in Wormhole Networks, *IEEE Trans. Parallel Distrib. Syst.*, Vol.6, pp.976–987 (online), DOI:10.1109/71.466634 (1995).
- 6) Samman, F.A., Hollstein, T. and Glesner, M.: Multicast parallel pipeline router architecture for network-on-chip, *Proceedings of the conference on Design, automation and test in Europe*, DATE '08, New York, NY, USA, ACM, pp.1396–1401 (online), DOI:http://doi.acm.org/10.1145/1403375.1403714 (2008).
- 7) Carara, E.A. and Moraes, F.G.: Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh Networks-on-Chip, *Proceedings of the 2008 IEEE Computer Society Annual Symposium on VLSI*, Washington, DC, USA, IEEE Computer Society, pp.341–346 (online), DOI:10.1109/ISVLSI.2008.18 (2008).
- 8) Samman, F.A., Hollstein, T. and Glesner, M.: Adaptive and deadlock-free tree-based multicast routing for networks-on-chip, *IEEE Trans. Very Large Scale Integr. Syst.*, Vol.18, pp.1067–1080 (online), DOI:http://dx.doi.org/10.1109/TVLSI.2009.2019758 (2010).
- 9) Lin, X., McKinley, P.K. and Ni, L.M.: Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers, *IEEE Trans. Parallel Distrib. Syst.*, Vol.5, No.8, pp.793–804 (1994).
- 10) Dally, W.J. and Aoki, H.: Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, *IEEE Trans. Parallel Distrib. Syst.*, Vol.4, pp.466–475 (online), DOI:http://dx.doi.org/10.1109/71.219761 (1993).