

適応型ルーティングにおける Output Selection Function に関する研究

鯉 淵 道 紘[†] 舟 橋 啓^{††}
上 樂 明 也[†] 天 野 英 晴[†]

相互結合網におけるルーティングは、経路が固定されている deterministic (oblivious) routing に代わって、動的に経路を選択できる adaptive routing が利用されるようになってきている。adaptive routing では出力チャネルの選択アルゴリズムである output selection function が性能に影響を与えるが、従来の output selection function は特定のトポロジやトラフィックパターンによって性能が大きく変化してしまう問題がある。本研究ではトラフィックパターンやトポロジ、ルーティングアルゴリズムに依存しない output selection function である load-dependent selection function と load-dependent selection function の機能をシンプルにした LRU (Least Recently Used) selection function を提案し、シミュレーションにより評価を行った。その結果、提案した output selection function は評価したすべてのトラフィックパターンにおいて、既存の方法に比べ高い性能を示すことが分かった。

The Impact of Output Selection Function on Adaptive Routing

MICHIHIRO KOIBUCHI,[†] AKIRA FUNAHASHI,^{††} AKIYA JOURAKU[†]
and HIDEHARU AMANO[†]

Recently, adaptive routing algorithms which achieve a high performance compared with deterministic ones are commonly used in large scale parallel machines. Although their performance is influenced with an output selection function or output channel selection algorithm, only a few studies depending on specific topologies or specific traffic patterns have been done. In this paper, we propose two output selection functions called load-dependent selection function and LRU (Least Recently Used) selection function, which are not based on a specific topology nor traffic pattern. Result of simulations shows that proposed output selection functions achieve better performance than traditional ones with each traffic pattern.

1. はじめに

近年、大規模並列計算機の研究、開発はますますさかんとなり、すでに数千ノードの規模を持つシステムが商用化されている。このような大規模並列計算機では、ノード間を結ぶ結合網およびルーティングアルゴリズムがシステムの構成、性能に大きな影響を与える。大規模並列計算機では、通常、転送容量と耐故障性の点から、出発地から目的地への最短経路が複数存在する結合網を用いる。また、最短経路が複数存在しない結合網でも、迂回を許せば複数の経路を持たせることができるものが多い。このような結合網では、あ

る経路が混雑したら、別の経路を使ってパケットを転送することにより、故障や混雑を回避することができる。このように、場合によって経路を選んでルーティングする方法を適応型ルーティング (adaptive routing) と呼ぶ。adaptive routing は経路を動的に選んでルーティングするため、結合網中のすべての故障箇所を把握する必要はなく、故障箇所をその場で迂回することが可能である。さらに空いているチャンネルを有効に用いることで、結合網の性能を最大限に引き出すことができ、また、局所的な混雑を回避する点においても deterministic routing に比べて有利である。多くの deadlock free adaptive routing が提案されており^{1)~4)}、一部の手法はすでに実際の並列計算機上でも用いられている。

adaptive routing は複数のパケット進路を選択することができるため、どの出力チャネルを選択する

[†] 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

^{††} 三重大学工学部

Faculty of Technology, Mie University

かを決定するアルゴリズム, すなわち output selection function が性能に影響を与える. output selection function を適切に設定すればチャンネルの使用率を均一にし, hotspot の原因となるパケット間の複雑な cyclic dependency を減らすことが可能であり, 相互結合網の転送性能を最大限に利用することができる. また, output selection function は adaptive routing の自由度が高くなるほど出力チャンネルの選択肢が増えるため, 相互結合網の性能をより引き出すことができる.

しかし, adaptive routing 自体の研究が広く行われているのに対し, 従来用いられている output selection function は, 塞がっていないチャンネルをランダムに選択する等の非常に単純な方法が, 特定の結合網のトポロジやトラフィックパターンに対する方法^{5),6)}に限られている.

自由度の高い adaptive routing が多数提案されている現在, adaptive routing の研究成果をより引き出すことができる output selection function の研究が相互結合網の性能向上のポイントになる可能性がある.

そこで, 本論文ではトラフィックパターンやトポロジ, ルーティングアルゴリズムに依存しない output selection function である load-dependent selection function, および, その性能を保持しつつ, ハードウェア量を抑えた LRU selection function を提案し, 評価する.

以降, 2 章では output selection function について紹介し, 3 章ではネットワークの状況を判断して出力チャンネルを選ぶ output selection function である load-dependent selection function を提案する. さらに 4 章では, load-dependent selection function をシンプルにした LRU (Least Recently Used) selection function を提案する. そして, 5 章で評価を行い, 6 章で結論を述べる.

2. Adaptive routing における output selection function

adaptive routing では, 出発地から目的地までに存在する複数経路を動的に選択してルーティングを行うため, 多くのパケットが途中経路において複数の出力チャンネルを選択することが可能となる. 出力チャンネルの選択は, 一般的にはそのチャンネルの状態に依存する. たとえば, 一方のチャンネルが使用されている場合には, 利用されていないチャンネルを優先的に利用する. しかし, 複数のチャンネルが使用されておらず, どちらも利用可能だった場合, output selection function を用いて利用すべき出力チャンネルを決定する. これらの選択

アルゴリズムをルーティングポリシーと名付けている場合もある^{7),8)}.

理想的な output selection function とは複雑な cyclic dependency の発生を抑え, すべてのノードでのチャンネルの使用率を均一にすることである. この条件が満足されれば, 結合網の全チャンネルが有効に利用され, またパケットの流れがスムーズになり, その結果, 結合網の性能を最大限に引き出すことができる.

今まで wormhole routing⁹⁾を用いた adaptive routing について多くの研究がなされてきたが, deterministic routing (oblivious routing) とのレイテンシ, スループットの比較に終わっているものが多く, output selection function がパフォーマンスに与える影響についての研究はあまり多くない.

既存の output selection function の中で最も単純な方法は, random selection function である. この方法は, 出力可能なチャンネルが複数の方向にある場合, ランダムにチャンネルを選択する. ランダムに出力チャンネルを選ぶことによりトラフィックをある程度分散させることができる.

次に, 基本的に k-ary n-cube を対象とした方法として dimension order selection function が提案されている. この方法は出力可能なチャンネルが複数の方向にある場合, その中で次元の一番低いチャンネルを選択する selection function である. たとえば 2 次元 torus において, x, y 方向ともに空いているチャンネルがある場合, x 方向を選択する. 一方, zigzag selection function⁷⁾では, 出力可能なチャンネルが複数の方向にある場合, その中で残っているホップ数が最大の次元の出力を選択する. つまり, mesh 等の結合網ではなるべく結合網の中心に向かって斜めにルーティングを行う selection function である. たとえば 2 次元 torus において, $s(x_s, y_s)$ から $d(x_d, y_d)$ にパケットを送る場合, x, y 方向ともに空いているチャンネルがあれば $|x_d - x_s|$ と $|y_d - y_s|$ の値のうち大きい方の次元方向を選択する.

ほかにも Badr らにより mesh において理論的に最適な selection function が提案されている⁷⁾. しかし, この理論は全チャンネルが均等に利用されるという仮定の下にのみ成立する. したがって, 実際に並列アプリケーションを実行した場合の性能向上には疑問が残る.

このように既存の output selection function は単純なもの以外は, 特定のトポロジやトラフィックパターンを対象とする点が問題である.

3. Load-dependent selection function

これまでに紹介してきた output selection function は、ネットワークの状況をまったく考慮に入れていないため、混雑している可能性の高い方向にパケットを転送してしまうことがある。現実にはネットワーク中に hot spot が発生する一方で空いているチャンネルが多数存在するようなケースが生じる。そのため hot spot の生成、通過をなるべく避け、全体のチャンネル利用率を均一にすることが output selection function の重要な役目であることはいうまでもない。そこでネットワークの状況を反映して出力チャンネルを選択する output selection function である、load-dependent selection function を提案する¹⁰⁾。

ネットワークの状況を把握する方法はいろいろ考えられるが、ネットワーク中のすべてのノードにおける混雑状況をクロックごとにどこかで集計して把握することは困難である。そこで load-dependent selection function では各ノードが各自の物理リンクをいつ利用したかという情報がある一定期間保存していることでネットワークの混雑状況をある程度把握する。

このために、ルータ内の各物理リンクの出力ごとにカウンタを1つ設け、flit が通過するごとにカウンタアップする。つまり、あるパケットが到着したときにカウンタの値が0だった場合、そのパケットの一番最後の flit (tail flit) の転送が終了した時点でのカウンタの値はそのパケット長となる。また、クロックごとに flit の出力要求がない方向のカウンタは0でなければ、デクリメントを行う。そして、ルーティングの際、出力可能な方向が2つ以上ある場合、各出力可能な物理リンクのカウンタを比べ、値が一番小さい出力方向を選択しパケットを転送する。load-dependent selection function の例として2次元 torus の場合を図1、図2に示す。図1ではカウンタ値が120と64でどちらも選択可能なので64の出力方向を選択している。片方の出力チャンネルがすでに他のパケットにより塞がっている場合は、利用可能なもう一方の出力チャンネルを選択する(図2)。

各次元のサイズが偶数の2次元 torus の場合、最短型の adaptive routing において load-dependent selection function はネットワークの状況に応じて出力チャンネルを最大4方向の中から選択することができる(図3)。

また、load-dependent selection function は worm-hole routing においてブロックされたパケットが同時に複数のチャンネルを占有し、停滞した場合、カウンタ

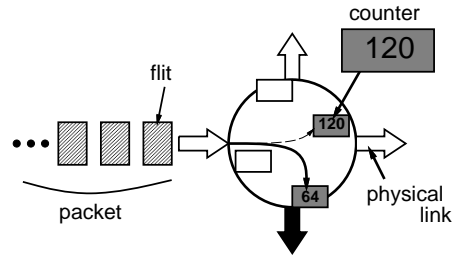


図1 2次元 torus での load-dependent selection function (2つの出力チャンネルが選択可能な場合)

Fig. 1 Load-dependent selection function on 2D torus (when two output channels are available).

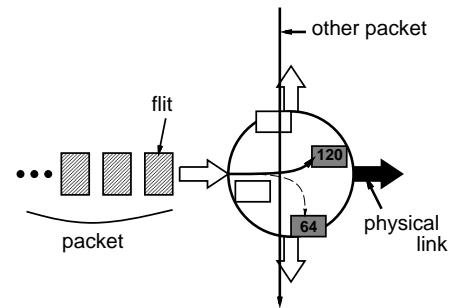


図2 2次元 torus での load-dependent selection function (片方の出力チャンネルが塞がっている場合)

Fig. 2 Load-dependent selection function on 2D torus (when one output channel is busy).

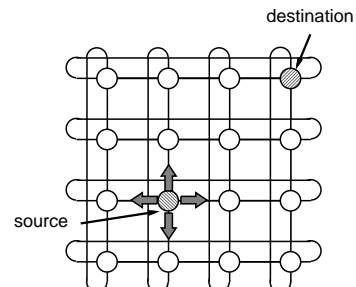


図3 各次元のサイズが偶数の2次元 torus における選択可能な出力方向 (minimal adaptive routing)

Fig. 3 Possible output directions on 2D even array-size torus (minimal adaptive routing).

操作を行わない。したがって、このパケットが存在するルータでその混雑情報を保持することができる。

パケット長がネットワークの直径に比べ大きい場合が多いことを考えると通過フリット数を通してネットワークの状況を把握する方式は1つの有効な方法である。

ところで、現在、相互結合網のスループットを上げるために、ルータはよりシンプルで高クロックに耐えられるものが必要である。実装されている並列計算機

上で wormhole routing が多く利用されているのはその典型である^{11),12)}。load-dependent selection function は実装上の容易さを考慮しており、ルータ内の各方向の物理リンクに対して1つカウンタを用意することにより実現している。また、load-dependent selection function は結合網のトポロジを選ばないため、すべての結合網に対して適用できる¹³⁾。

簡単に load-dependent selection function の特徴をまとめると

- (1) 以前パケットを送出したチャンネルをなるべく利用しない。
- (2) 一定時間経過したルーティング情報は消滅する。
- (3) 物理チャンネルごとにカウンタが必要。

(1) は hot spot の生成を抑え、(2) はすでに目的地に到着し、ネットワークに存在しないパケットや、ルータから十分離れているパケットの情報を除外し、そのノード近辺の混雑状況を保持する働きを持つ。

4. LRU selection function

load-dependent selection function はネットワークの混雑状況に応じて出力チャンネルを選択できる点で従来の方法と異なっている。しかし、パケット長が大きい場合、load-dependent selection function の通過フリット数を記録するカウンタは大容量なものになってしまい、ハードウェア量が無視できないレベルになる可能性がある。

そこで本章では load-dependent selection function の機能を大幅に削減することによりハードウェア量を抑え、なおかつ性能をほぼ維持することができる output selection function である LRU (Least Recently Used) selection function を提案する。

LRU selection function のアルゴリズムは各ノードにおいて使用されずにいた時間の最も長い出力チャンネルを選択する、というシンプルなものである。LRU selection function は利用されていないチャンネルを優先的に選択するため、load-dependent selection function と同様にトラフィックパターンによらずトラフィックを分散させる効果が期待できる。

load-dependent selection function では各出力チャンネルを一定時間内に通過したフリット数に応じて出力チャンネルを選択したのに対し、LRU selection function は各出力チャンネルを最後に通過したパケットの経過時間のみに応じて出力チャンネルを選択する。

load-dependent selection function と LRU selection function の主な違いは以下の2点である。

- load-dependent は LRU と違い、パケット長を考

慮して出力チャンネルを選択する。

- load-dependent はパケットがブロックされて停滞した場合の混雑情報が記録されない(カウンタ値が変わらない)。これに対し LRU では停滞している方向の優先度を最も低く設定する。

LRU selection function のアイデアは仮想記憶におけるページの追い出しアルゴリズムである LRU (Least Recently Used) と同様である。LRU はメモリアクセスの局所性を最大限利用できるためヒット率を上げることができる。これに対して LRU selection function はトラフィックの局所性をできるだけ排除し、分散させるため、使用されずにいた時間の最も長いチャンネルを優先的に使用する。

5. 評価

提案した LRU selection function、および load-dependent selection function についてフリットレベルシミュレータを C++ で記述し、評価を行った。一般的な deterministic routing (e-cube routing¹⁴⁾) と fully adaptive routing (Duato's protocol¹⁾) との比較を行ったうえで Duato's protocol における5つの output channel selection function (dimension order, random, zigzag, load-dependent, LRU) について比較、検討を行う。

5.1 シミュレーション条件

以下、シミュレーションに用いた条件を示す。

シミュレーション条件

- 実行時間: 50,000 clk
(初めの 5,000 clk は評価せず)
- バーチャルチャンネル数: 3 (Duato's protocol), 2 (e-cube)
- ネットワーク: 2D-torus
- ネットワークサイズ: 16×16 (256 ノード), または 32×32 (1,024 ノード)
- パケットが 1 hop 移動に要する時間: 最低 3 clk (ルーティング, ルータ内のクロスバ移動, ルータ間の移動に各 1 clk)
- パケット長: 128, または 256, 64 を混在
- トラフィックパターン
 - uniform traffic
 - matrix transpose traffic
 - bit reversal traffic

シミュレーションで初めの 5,000 クロックはネットワークが安定せず、想定した負荷に達していないと考

えられるため評価は行ななかった．また，パケット長を考慮した出力チャンネルの選択を行う load-dependent と，考慮しない LRU との性能差を調べるために，複数の長さのパケットが混在する場合と，すべてのパケットが均一の長さの場合の 2 パターンを用いた．

トラフィックは uniform traffic に加え，確率モデルのシミュレーションでよく用いられる 2 つのトラフィックパターン (matrix transpose , bit reversal) を用いた．matrix transpose traffic はネットワークサイズを k とすると，座標が (x, y) のノードは $(k - y - 1, k - x - 1)$ のノードにパケットを転送するトラフィックパターンである．また，bit reversal traffic は，各ノード番号を 2 進数で表記しノード番号が $(a_0, a_1, \dots, a_{n-2}, a_{n-1})$ のノードは自分のノード番号のビット列を逆順に並べた $(a_{n-1}, a_{n-2}, \dots, a_1, a_0)$ のノードにパケットを転送するトラフィックパターン

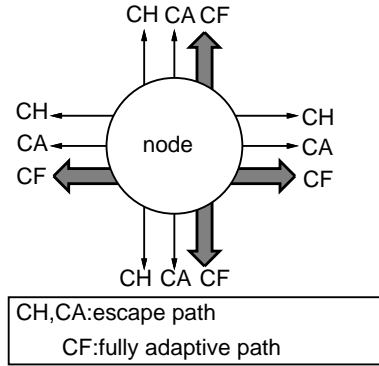


図 4 2次元 torus における Duato's protocol でのバーチャルチャンネル
Fig. 4 Virtual channels required by Duato's protocol on 2D torus.

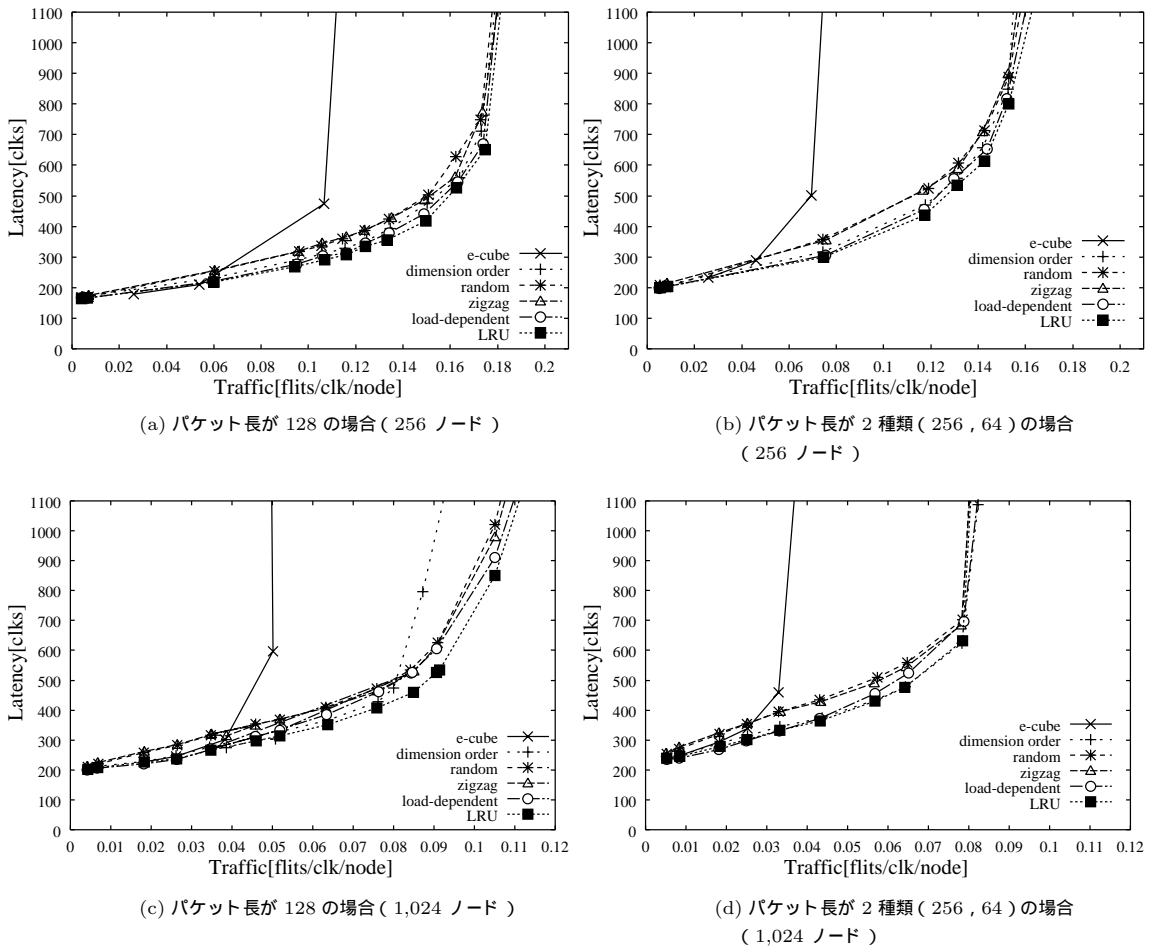


図 5 Uniform traffic
Fig. 5 Uniform traffic.

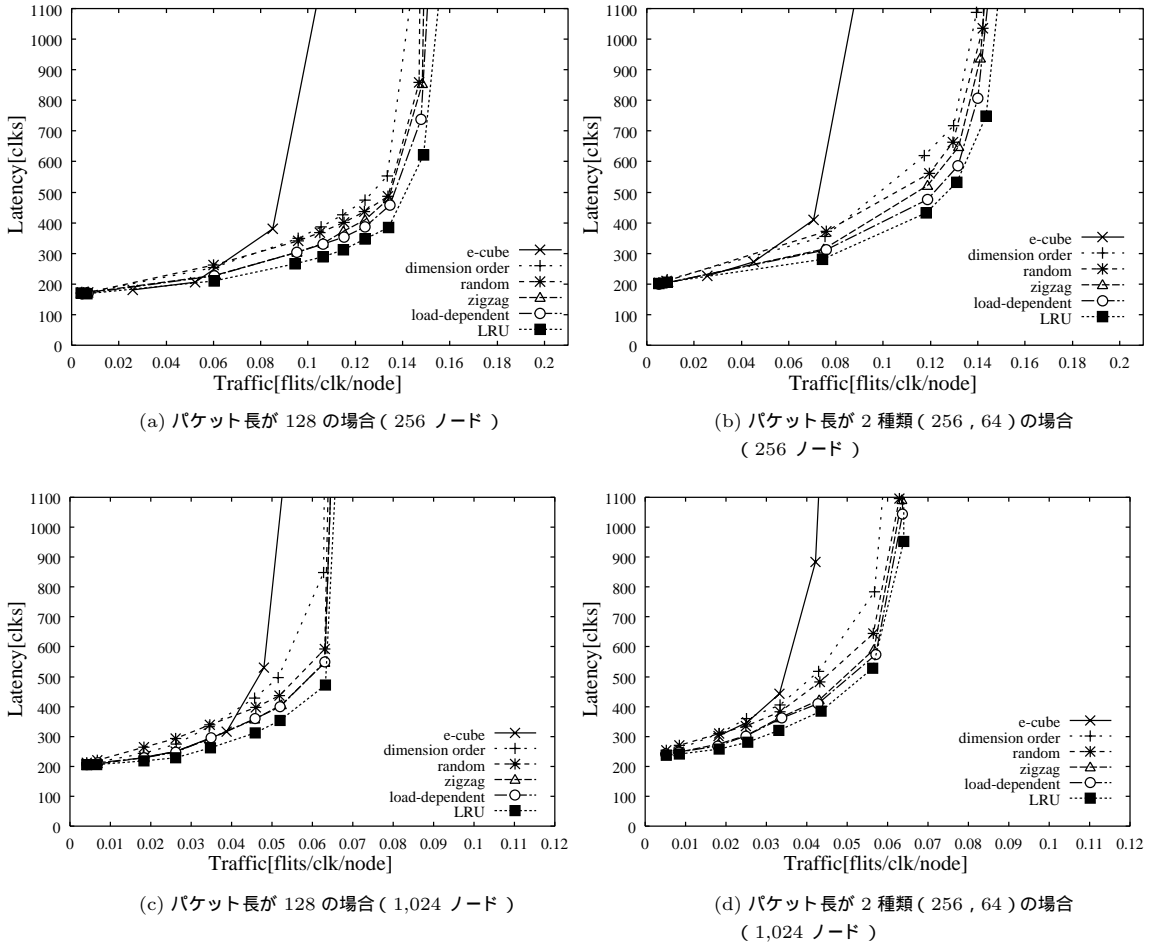


図 6 Matrix transpose traffic
Fig. 6 Matrix transpose traffic.

である。

5.1.1 ルーティングアルゴリズム

シミュレーションで使用したルーティングアルゴリズムについて、パーチャルチャネルの使い方等を以下に述べる。

e-cube routing

2次元 torus における e-cube routing とはパケットが x 次元を必要 hop 数移動した後、 y 次元を移動する最小型の deterministic routing である。torus の場合、デッドロックフリーを保証するために各次元に wraparound channel が存在する経路と存在しない経路の 2 つが必要になる。したがって各ノード間に 2 本以上のパーチャルチャネルが必要である。

Duato's protocol

Duato's protocol はネットワーク全体にわたる循環のない経路 (escape path) を e-cube routing で用意し、escape path と、escape path により循環が切断

されたチャネル使用制限のない経路 (fully adaptive path) を動的に切り替えてルーティングを行う fully adaptive routing である。2次元 torus では各ノード間に 3 本以上のパーチャルチャネルが必要になる (図 4)。図 4 は 2 本のパーチャルチャネル (CA, CH) を e-cube routing に用い escape path とし、1 本のパーチャルチャネル (CF) を fully adaptive path に用いて Duato's protocol を実現した例である。

5.2 Uniform traffic

uniform traffic での評価を図 5 に示す。横軸はトラフィック負荷、縦軸はレイテンシを表している。トラフィック負荷は、全ノードが毎クロックに 1 flit 受信する場合を 1.00 としており、レイテンシは出発地のプロセッサがパケットを生成してから、目的地のプロセッサがパケットの最後の flit を受け取るまでの時間を表す。また、uniform traffic では各パケットの目的地ノードはランダムで決定されており、等確率に分散

されている。

図 5 (a), 図 5 (b) はネットワークサイズが 256 ノードであり, それぞれパケット長が 128 の場合, パケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す. また, 図 5 (c), 図 5 (d) はネットワークサイズが 1,024 ノードであり, それぞれパケット長が 128 の場合, パケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す.

図 5 の各図において, deterministic routing でのレイテンシも比較のため載せてあるが, adaptive routing との差は明白であり, 特にトラフィック負荷が高い場合には adaptive routing と deterministic routing の性能の差が顕著になる. Duato's protocol と deterministic routing はどちらも最短経路しか通らないルーティングのため hop 数は基本的には変わらないが, Duato's protocol を用いたほうがパーチャルチャネルを効率良く利用するため, 高い性能を示す.

5 つの output selection function の中では LRU, load-dependent がつねにレイテンシが低く, 若干高い性能を示しているが, さほど差は出ていない. これは uniform traffic では output selection function に特別なものを利用しなくてもトラフィック自体が十分に分散されており, 各ノードのチャネルが均等に利用されるため output selection function でトラフィックを分散する必要が高くないことに起因する. しかし, 局所的にトラフィックの集中が発生し, output selection function による差が若干表れたと考えられる. この傾向はノード数が増えるほど強くなっている.

また, 複数の長さのパケットが存在する場合, load-dependent は LRU と違い, パケット長を考慮して出力チャネルを選択できるため有利になるはずだが, 図 5 (b), 図 5 (d) より, LRU が load-dependent よりも高い性能を示した.

これは load-dependent, LRU とともに全体のトラフィックを完全に把握して出力チャネルを選択しているわけではないので, トラフィック自体が十分に拡散されている場合, load-dependent の厳密な方法よりも, シンプルなアルゴリズムの LRU を用いておおまかにトラフィックを振り分ける方法の方が有効であることを示している. また, これは load-dependent は LRU と違い, ネットワークがブロックされて詰まった場合の混雑情報を考慮しないことが影響している.

5.3 Matrix transpose traffic

次に matrix transpose traffic での評価を図 6 に示す. 図 6 (a), 図 6 (b) はネットワークサイズが 256 ノードであり, それぞれパケット長が 128 の場合, パ

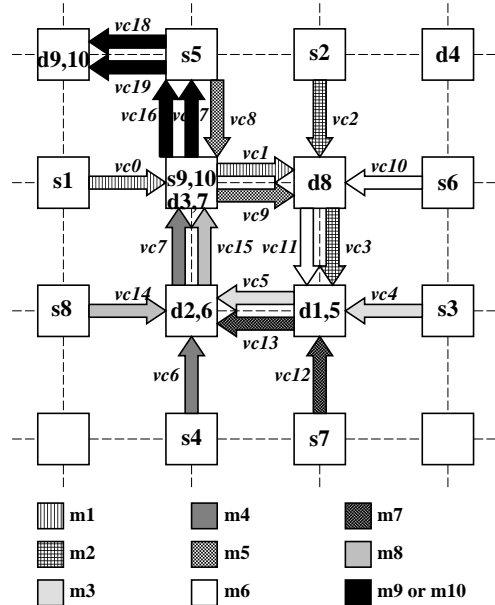


図 7 cyclic dependency の発生
Fig. 7 Generation of cyclic dependency.

ケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す. また, 図 6 (c), 図 6 (d) はネットワークサイズが 1,024 ノードであり, それぞれパケット長が 128 の場合, パケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す.

一般的に uniform traffic では均一にトラフィックが分散されるのに対し, matrix transpose traffic は中央付近に偏る. よって同条件下では各 selection function と uniform traffic に比べレイテンシが大きくなる.

図 6 の各図において, deterministic routing でのレイテンシも比較のため載せてあるが, uniform traffic の場合と同様に adaptive routing との差は明白である.

5 つの output selection function を比較すると, load-dependent, LRU のレイテンシが常に低いことが分かる. また, トラフィック負荷が高くなると dimension order のレイテンシが急激に高くなるが, 他は緩やかな曲線を描いてレイテンシが上昇する. これはパーチャルチャネルを効率良く利用する Duato's protocol でも偏りがあるようなトラフィックではパーチャルチャネルを利用する順番がパケット間の複雑な cyclic dependency の発生に影響を与え, レイテンシに影響を与えることを示している. adaptive routing における cyclic dependency とはデッドロックを起こしてはいないが, パケット間の相互依存が発生してしまっている状況をいう (図 7). cyclic dependency が

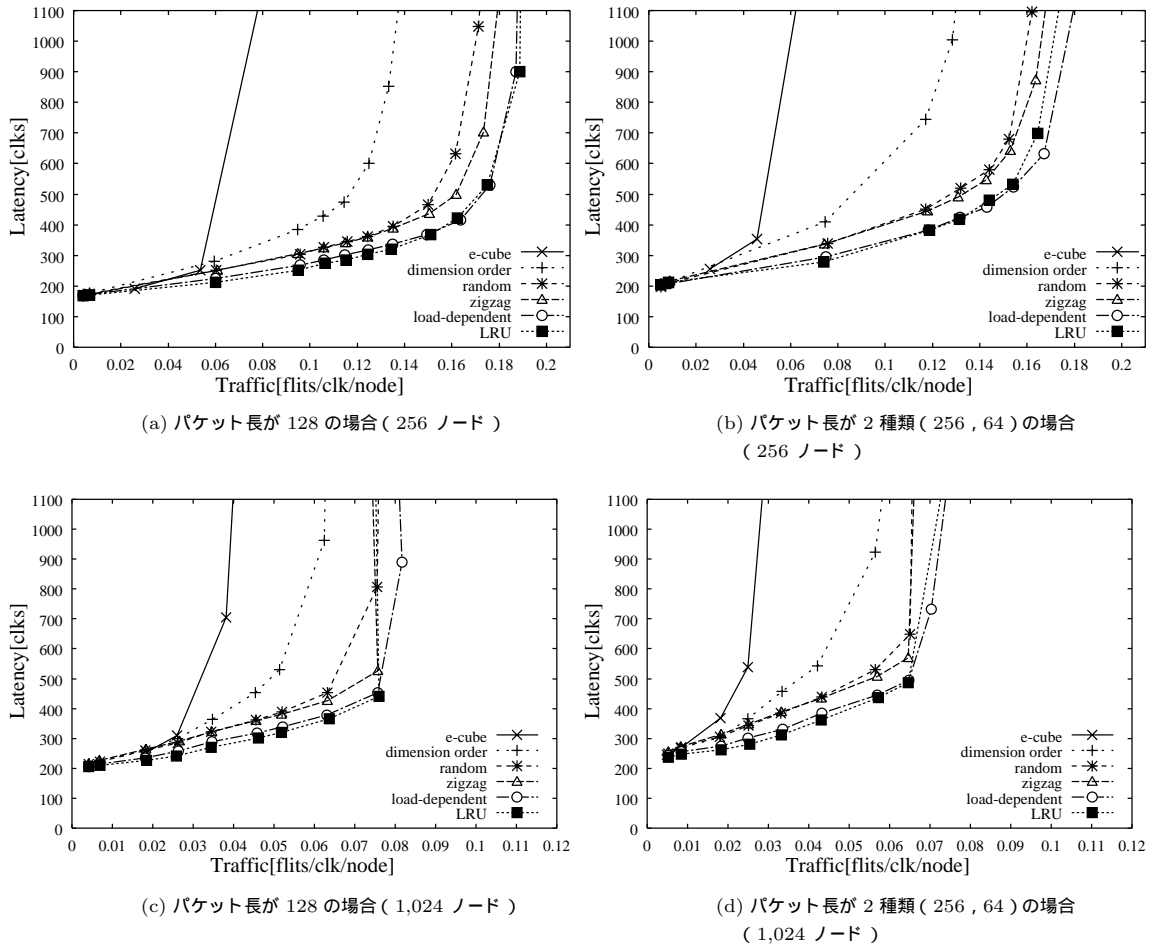


図 8 Bit reversal traffic
Fig. 8 Bit reversal traffic.

発生すると、あるパケットが動くまで他のパケットが動けないため複数のパケットが 1 つずつ順番に移動せざるをえない状況が発生する。このため、レイテンシが大幅に増え、hotspot の原因となる。

図 6 の各図より 5 つの output selection function の中では uniform traffic と同様に load-dependent と LRU が高い性能を示している。これは load-dependent と LRU はトラフィックパターンによらず、パケットがノードに到着した時点でのネットワークの負荷に応じて出力チャネルを選択するので、安定して高い性能を示すことを表している。特にこの傾向はノード数が多く、また、複数のパケット長が存在する場合ほど強い。

LRU と load-dependent を比べると、図 6 (a)、図 6 (b)、図 6 (c)、図 6 (d) とともに LRU が load-dependent よりも高い性能を示した。これは load-dependent はネットワークの状況を各出力チャネルを一定時間内

に通過したフリット数を基に判断する。したがって、load-dependent ではパケットが停滞した場合の混雑情報を考慮しないことが影響している。

5.4 Bit reversal traffic

次に、bit reversal traffic での評価を図 8 に示す。図 8 (a)、図 8 (b) はネットワークサイズが 256 ノードであり、それぞれパケット長が 128 の場合、パケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す。また、図 8 (c)、図 8 (d) はネットワークサイズが 1,024 ノードであり、それぞれパケット長が 128 の場合、パケット長が 256 と 64 のパケットがランダムに混ざっている場合を表す。

一般的に bit reversal traffic は matrix transpose traffic のような極端な偏りは発生しないが、局所的な範囲で小規模な偏りが多数発生する特徴を持つ。そのため一般的には同条件下において各 selection function は uniform traffic と matrix transpose traffic の中間

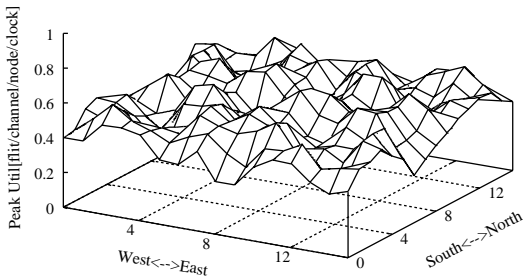


図9 ピークチャネル利用率の分布 (bit reversal , dimension order)

Fig. 9 Peak channel util. (bit reversal, dimension orser).

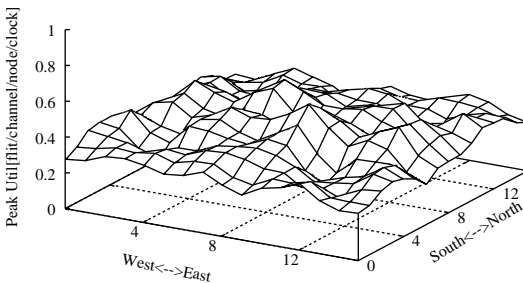


図10 ピークチャネル利用率の分布 (bit reversal , load-dependent)

Fig. 10 Peak channel util. (bit reversal, load-dependent).

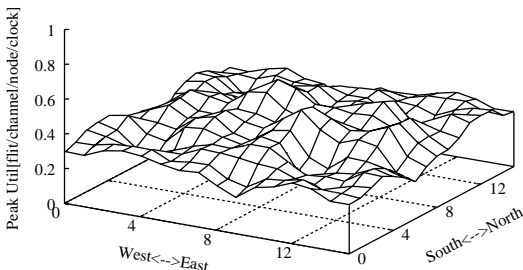


図11 ピークチャネル利用率の分布 (bit reversal , LRU)

Fig. 11 Peak channel util. (bit reversal, LRU).

の性能を示す。

図8の各図に示すように、明らかに adaptive routing の方が deterministic routing に比べて高い性能を示している。

5つの output selection function を比較すると、dimension order が最も低いトラフィック負荷でネットワークの飽和を引き起こし、一方 load-dependent が最も高い性能を示している。また、同じトラフィック負荷の場合、LRU, load-dependent のレイテンシが random に比べ10%以上低い。同じ Duato's protocol を用いているにもかかわらず、output selection function による性能差が著しいことが分かる。図8(c)にいたつ

ては dimension order は deterministic routing の約2倍のスループットを持つが、LRU, load-dependent は deterministic routing の約3倍のスループットを持つ。このように bit reversal traffic では matrix transpose traffic 以上に selection function による性能差が生じる。これは selection function が局所的な範囲の traffic の偏りを効果的に改善できることを示している。

5.5 ピークチャネル利用率の分布

次に各 output selection function において、各ノードのピークチャネル利用率の分布について調査する。

bit reversal traffic での評価である図8(a)において、トラフィック負荷が0.18のときのピークチャネル利用率の分布を図9、図10、図11に示す。

縦軸はノードごとのピークチャネル利用率を表す。ピークチャネル利用率とは、各ノードごとに1,000 clk ごとのチャネル利用率を集計し、5,000 ~ 50,000 clk 中での最高値を表す。また、横軸はノードの位置を表す。output selection function がトラフィックを適切に分散させていれば、各ノードのピークチャネル利用率をより低く、均一にできるはずである。

図9、図10、図11より、LRU, load-dependent は dimension order と比べ、明らかにピークチャネル利用率を低く、より均一にしている。よって LRU, load-dependent output selection function の目標であるトラフィックの分散に成功したことを示している。LRU, load-dependent の差はほとんど見られず、両者とも、bit reversal traffic においてトラフィックを分散する能力を同程度持っていることが分かった。

6. ま と め

adaptive routing において複数ある出力チャネルを選択するアルゴリズムである output selection function に、ネットワークの混雑状況により出力チャネルを選択する新たな output selection function (load-dependent selection function, LRU selection function) を提案し、評価を行った。

LRU selection function, load-dependent selection function はトラフィックパターンや結合網のトポロジに依存せず、つねに最適に近い output selection function を提供する。シミュレーションの結果より、すべてのトラフィックパターンにおいて従来の output selection function より高い性能を実現することが分かった。

LRU と load-dependent は、ほぼ同じ性能を示したため、output selection function としてはハードウェア量の小さい LRU が適しているといえる。

今後は、本研究室で実装中である相互結合網の命令レベルシミュレータを用いて実際のアプリケーション上での評価を行う予定である。

参 考 文 献

- 1) Duato, J.: A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks, *IEEE Trans. Parallel and Distributed Systems*, Vol.6, No.10 (1995).
- 2) Glass, C.J. and Ni, L.M.: Maximally Fully Adaptive Routing in 2D Meshes, *Proc. ISCA92*, pp.278-287 (1992).
- 3) Chien, A.A. and Kim, J.J.: Planar-Adaptive Routing: Low-cost Adaptive Networks for Multiprocessors, *Proc. ISCA92*, pp.268-277 (1992).
- 4) Dally, W.J. and Aoki, H.: Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, *IEEE Trans. Parallel and Distributed Systems*, Vol.4, No.4, pp.466-475 (1993).
- 5) Feng, W.-C. and Shin, K.G.: Impact of Selection Functions on Routing Algorithm Performance in Multicomputer Networks, *Proc. 11th Annual Conference on Supercomputing* (1997).
- 6) Schwiebert, L. and Bell, R.: The Impact of Output Selection Function Choice on the Performance of Adaptive Wormhole Routing, *Proc. International Conference on Parallel and Distributed Computing Systems*, pp.539-544 (1997).
- 7) Badr, S. and Podar, P.: An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies, *IEEE Trans. Comput.*, Vol.38, No.10, pp.1362-1371 (1989).
- 8) Wu, J.: An Optimal Routing Policy for Mesh-Connected Topologies, *Proc. International Conference on Parallel Processing*, Vol.1, pp.267-270 (1996).
- 9) Ni, L.M. and McKinley, P.K.: A Survey of Wormhole Routing Techniques in Direct Networks, *IEEE Trans. Comput.* (1993).
- 10) 鯉淵道紘, 舟橋 啓, 上樂明也, 天野英晴: 適応型ルーティングにおける output selection function, 並列処理シンポジウム JSPP'2000 予稿集, pp.181-188 (2000).
- 11) Baverton, OR and Intel Corporation, S.S.D.: *Paragon XP/S Product Overview* (1991).
- 12) Oed, W.: *The Cray Research Massively Par-*

allel Processing System: Cray T3D, Cray Research (1993).

- 13) 鯉淵道紘, 舟橋 啓, 上樂明也, 若林正樹, 天野英晴: Irregular Network における Adaptive Routing の提案, 信学技報, CPSY2000-44, pp.33-40, 電子情報通信学会 (2000).
- 14) Dally, W.J. and Seitz, C.L.: Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, *IEEE Trans. Comput.*, Vol.36, No.5, pp.547-553 (1987).

(平成 12 年 9 月 14 日受付)

(平成 12 年 12 月 1 日採録)



鯉淵 道紘

平成 12 年慶應義塾大学理工学部情報工学科卒業。現在、同大学大学院理工学研究科修士課程に在学中。計算機アーキテクチャに関する研究に従事。



舟橋 啓

平成 7 年慶應義塾大学理工学部電気工学科卒業。平成 12 年同大学大学院博士課程修了。現在、三重大学理工学部情報工学科助手。工学博士。計算機アーキテクチャに関する研究に従事。



上樂 明也

平成 10 年慶應義塾大学理工学部電気工学科卒業。平成 12 年同大学大学院修士課程修了。計算機アーキテクチャに関する研究に従事。



天野 英晴 (正会員)

昭和 56 年慶應義塾大学工学部電気工学科卒業。昭和 61 年同大学大学院理工学研究科電気工学専攻修士課程修了。現在、慶應義塾大学理工学部情報工学科助教授。工学博士。計算機アーキテクチャの研究に従事。