

# チャネルの使用条件に基づいた アウトプットセレクションファンクションに関する研究

杉山 伸 悟<sup>†</sup>, 舟橋 啓<sup>†</sup>

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固定ルーティングに代って動的に経路を選択できる適応型ルーティングが利用されるようになってきている。適応型ルーティングでは出力チャネルの選択アルゴリズムであるアウトプットセレクションファンクションが性能に影響を与える。しかし、従来のアウトプットセレクションファンクションでは自由度の高い適応型ルーティングに対し適切な判断に基づいて出力チャネルを決定できない、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、などの問題がある。本研究ではトラフィックパターンやトポロジに依存しないアウトプットセレクションファンクションである CCB (Channel Characteristic Based) セレクションファンクションを提案し、シミュレーションにより評価を行った。CCB は既存の研究と異なり、ルーティングアルゴリズムによってバーチャルチャネルの特徴 (使用条件) が各チャネルごとに異なる点に着目し、その性質を利用してパケットの転送効率を向上させる。シミュレーションの結果、CCB は評価したすべてのトラフィックパターンにおいて、既存の方法より高い性能を示すことが分かった。

## An Output Selection Function Based on the Characteristic of Virtual Channel

SHINGO SUGIYAMA<sup>†</sup> and AKIRA FUNAHASHI<sup>†</sup>

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 based on specific topologies or specific traffic patterns have been done. In this paper, we propose a novel output selection function called CCB (channel characteristic based) selection function, which doesn't depend on a specific topology nor traffic pattern. CCB is proposed by paying attention on a causal relationship between the characteristic of each virtual channel and the network performance. Result of simulations shows that proposed output selection function improves the performance compared with 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どとなり、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。一般的に並列計算機で用いられている結合網は出発地から目的地への最短経路が複数存在する。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。結合網は転送容量

を上げるため、もしくは耐故障性を持たせるために出発地から目的地への最短経路を複数持つようにするのが望ましい。また、最短経路が複数存在しない場合でも、多くの結合網は迂回を許すことにより、複数の経路を持たせることができる。

経路を決定する際に、出発地ノードから目的地ノードが決まれば必ず同じ経路を通る固定ルーティング (deterministic routing) がある。固定ルーティングは実装が容易という利点を持つ一方、ある経路が混雑したり、故障で使えなくなったりした場合、その経路を迂回することができないといった欠点がある。それに対し、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

<sup>†</sup> 三重大学工学部情報工学科

Department of Information Technology, Mie University  
現在、静岡日電ビジネス株式会社ソリューションシステム部  
Presently with NEC Shizuoka Business Corporation

routing)と呼ぶ。適応型ルーティング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。さらに適応型ルーティングは、空いている経路や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固定ルーティングに比べて有利である。適応型ルーティングではパケットの経路が動的に変化するため、デッドロックを起こす可能性がある。そのため、デッドロックフリーであることを保証しなければならない。さらに、適応型ルーティングでは出力チャンネルの選択アルゴリズムであるアウトプットセレクションファンクションがパフォーマンスに影響を与えるにもかかわらず、既存のアウトプットセレクションファンクションではトラフィックパターンに依存してしまい、思うような効果を得ることができていない。そこでトラフィックパターンやトポロジに依存しないCCB(Channel Characteristic Based)セレクションファンクションを提案し、評価を行う。

本研究では、Duato's protocolを用いた方法でデッドロックフリーな適応型ルーティングを実現し、シミュレーションによる評価を行う。

CCBではバーチャルチャンネルの状況を調べ、既存のアウトプットセレクションファンクションではあまり使われていなかったチャンネルを優先的に使うことで、今まで偏りがみられたネットワーク資源の使用を均一化しようとしていることが最大の特徴である。この特徴により、大きな性能向上を実現できた。

以降、2章では既存のアウトプットセレクションファンクションについて紹介する。3章でCCBを提案、およびそのアルゴリズムを説明し、4章で評価を行う。最後に5章で結論を述べる。

## 2. アウトプットセレクションファンクション

### 2.1 既存のアウトプットセレクションファンクション

適応型ルーティングでは、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、2つのチャンネルのうち、一方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が、両方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には、アウトプットセレクションファンクションを用いて利用すべき出力チャネ

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<sup>1),2)</sup>。

理想的なアウトプットセレクションファンクションとは、各物理チャンネルを時分割で共有しているバーチャルチャンネルの利用数をなるべく抑え、かつすべてのノードでのチャンネルの使用率を均一にするものである。この条件が満足されれば、パケットの流れがスムーズになり、かつ、結合網の全チャンネルを有効に利用できる。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。

今までワームホールルーティング<sup>3)</sup>を用いた適応型ルーティングについて多くの研究がなされてきたが、固定ルーティングとのレイテンシ、スループットの比較に終わっているものが多く、アウトプットセレクションファンクション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。

既存のアウトプットセレクションファンクションの中で最も単純な方法は、Randomセレクションファンクションである。この方法は、出力可能なチャンネルが複数の次元(物理チャンネル)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セレクションファンクションが提案されている。この方法は出力可能なチャンネルが複数の次元にある場合、その中で次元の一番低いチャンネルを選択する方法である。たとえば2次元トラスにおいて、 $x, y$ 次元ともに空いているチャンネルがある場合、 $x$ 次元を選択する。一方、Zigzagセレクションファンクション<sup>1)</sup>では、出力可能なチャンネルが複数の次元にある場合、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、メッシュなど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方法である。たとえば2次元トラスにおいて、 $s(x_s, y_s)$ から $d(x_d, y_d)$ にパケットを送る場合、 $x, y$ 次元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を選択する。

しかし、これらのアウトプットセレクションファンクションは、ネットワークの状態をまったく考慮に入れておらず、混雑している可能性の高い次元にパケットを転送してしまうことがある。そのため、性能がトラフィックパターンやトポロジに依存してしまう問題がある。

一方、ネットワークの状態を反映する機能を持たせようとしたものもある。Martinezらは出力可能なチャンネルが複数の次元にある場合、最も最近使われていない出力バーチャルチャンネルを選択するLeast Recently

Used ( LRU ) セレクションファンクションを提案した<sup>4)</sup>。トラフィックを分散させる効果を狙ったものだが、パーチャルチャネルレベルで出力チャネルを選択するだけで、物理的なチャネルレベルでの考慮がなされていない。そのため、物理リンクに時分割で共有するパーチャルチャネルが複数ある場合、混雑している物理チャネルに最も最近使われていないパーチャルチャネルがあると、混雑している物理チャネルへパケットを転送してしまう。その結果、物理チャネルレベルでのトラフィックの分散がうまく行われず、パケットのレイテンシが大きくなってしまい性能が出ない。

Martinez らはほかにも、過去一定期間内に最も使われていない出力チャネルを選択する Least Frequency Used ( LFU ) などを提案している<sup>4)</sup>。しかし、いずれもパーチャルチャネルにのみ着目しているため、LRU と同様、各物理チャネルに複数のパーチャルチャネルが存在する場合、トラフィックが必ずしも分散されず、思うような性能が得られない。

我々も各ノードでトラフィックを分散させるため、各ノードが 1 クロックの転送で送ることのできる単位であるフリットをカウントし、トラフィックを把握する LD ( Load-dependent ) セレクションファンクションを提案した<sup>5)</sup>。しかし、その考え方の視点が物理チャネルのみであったために、ネットワークの混雑を正確に把握することができなかった。

ほかにも Badr らによりメッシュにおいて理論的に最適なセレクションファンクションが提案されている<sup>1)</sup>。しかし、この理論は全チャネルが均等に利用されるといった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。

## 2.2 既存のアウトプットセレクションファンクションの評価

今まで紹介した既存のアウトプットセレクションファンクションについて、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。評価に用いた結合網、ルーティングアルゴリズムおよびパラメータなどについて簡単に説明する。

結合網は図 1 に示すような 2 次元トーラスを用いた。トーラスは各次元のノードをリングで結合した構造を持ち、結合網中に端点を持たない構成となる。なお、ネットワークサイズは  $16 \times 16$  で評価を行った。シミュレータに与えるトラフィックパターンには、偏りのあるトラフィックパターンとして Bit reversal traffic を用いた。Bit reversal traffic については 4 章で詳しく述べる。また、各アウトプットセレクションファンク

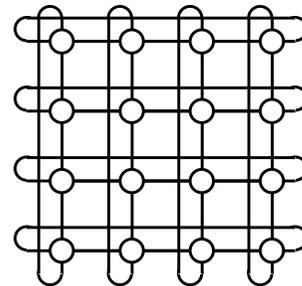


図 1 2次元トーラス  
Fig. 1 2D torus.

ションは 3.1.1 項に述べられている「パーチャルチャネルの中で最も使用条件が厳しいものを優先」というセレクションファンクションを併用し、ネットワークレイテンシ、スループットの 2 つの指標により評価を行った。これら 2 つについても 4 章で詳しく説明する。ルーティングアルゴリズムには、\*-channel と呼ばれる Duato's protocol<sup>6)</sup>を使用した。ここで簡単に\*-channel を  $n$  次元トーラスに適用した例について説明する。

\*-channel はネットワーク全体にわたる循環のない経路 ( escape path ) を用意し、escape path と、escape path により循環が切断された、チャネル使用条件のない経路 ( fully adaptive path ) を動的に切り替えてルーティングを行うデッドロックフリーなアルゴリズムである。 $n$  次元トーラスの場合、escape path には e-cube ルーティング<sup>7)</sup>を用いる。e-cube ルーティングではデッドロックフリーを保証するために最低次元から順にルーティングを行う。また、トーラスでは各次元のノードはリング状に結合されているため、各次元の端点を接続するラップアラウンドチャネルを通過する際にチャネルの切換えを行う必要がある。そのため e-cube ルーティングでは各ノード間に 2 本 ( CH, CA ) のパーチャルチャネルを必要とし、その結果 \*-channel には各ノード間に 3 本 ( CH, CA, CF ) 以上のパーチャルチャネルが必要になる。そのためシミュレータでも 3 本のパーチャルチャネルを使用する ( 図 2 )。

また、\*-channel で用いられるパーチャルチャネルは使用条件がそれぞれ異なる。CH は現在のノードから各次元における目的ノードに到達するまでにラップアラウンドチャネルを使用することがない場合に利用可能なチャネルで、CA はこの条件がないチャネルである。これらは最短経路かつ一定の次元順 ( 低次元から ) で用いられるためデッドロックフリーであり、escape path となる。これに対して CF は最短経路を守ること以外の使用条件がない fully adaptive path であり、

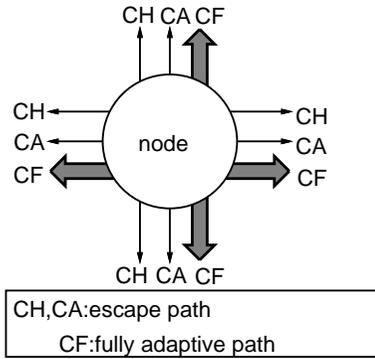


図2 2次元トーラスにおける\*-channelでのバーチャルチャネル  
Fig. 2 Virtual channels required by \*-channel on 2D torus.

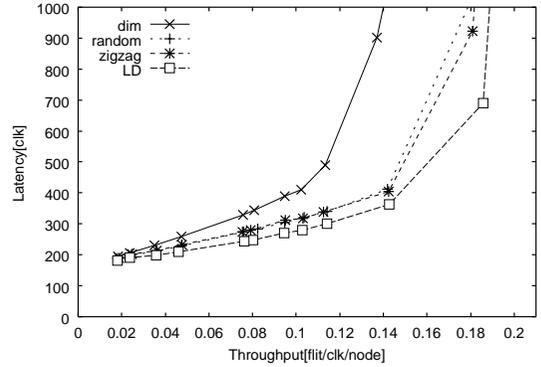
CFはどんな場合においても使用可能である。チャネルの使用条件はCF, CA, CHの順に厳しくなっていく。

具体的にルーティング例をあげると、あるノードに到着したパケットが2次元以上のルーティングの経路が残されている場合(つまり、2次元トーラスなどで  $x, y$  のどちらの次元にもルーティングが可能な場合), 最低次元にはバーチャルチャネル3本(CF, CA, CH)のうちどれかを使用してルーティングが可能だが、その他の次元にルーティングを行う場合にはCFチャネルのみが使用可能である。なお、本論文では  $n$  次元トーラスにおいて最低次元とは  $x$  次元のことを指すこととする。

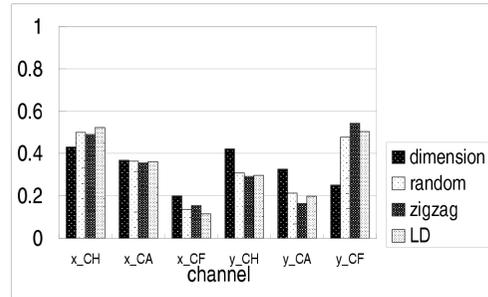
ここで、図3(a), (b)にシミュレータによる評価を示す。

図3(a)に目を向けると、LDが最も良い性能を示していることが分かる。また、図3(b)は、各アウトプットセレクションファンクションごとのそれぞれのバーチャルチャネルの使用率である。このグラフより、各バーチャルチャネルの使用率には傾向があることが分かる。この傾向は、各バーチャルチャネルはそれぞれ使用条件が違いため、同一に扱うべきではないことを表し、それはまた、それぞれのバーチャルチャネルの特徴を表している。

図3の2つのグラフを見比べると、性能とチャネルの使用率の間には関係があることが分かる。たとえば、LDのような高い性能を示すアウトプットセレクションファンクションほど、 $x$ 次元のCFチャネル( $x_{CF}$ )の使用率が少なく、 $y$ 次元のCFチャネル( $y_{CF}$ )の使用率が高いということである。既存のアウトプットセレクションファンクションの研究では、各バーチャルチャネルの特徴の違いに着目した例はなく、ここに性能向上の鍵があると思われる。そこで本研究では、



(a) 16 × 16 2次元トーラス (256 ノード) Bit reversal



(b) 16 × 16 2次元トーラスチャネル分布

図3 性能とチャネルの分布の関係

Fig. 3 Simulation results of existing output selection functions.

各バーチャルチャネルの特徴を考慮したアウトプットセレクションファンクションを提案する。

### 3. Channel Characteristic Based セレクションファンクション

2章では、既存のアウトプットセレクションファンクションがそれぞれ問題をかかえていることを述べた。その中で少し触れたが、これまで紹介してきたLD以外のアウトプットセレクションファンクションは、ネットワークの状況をまったく考慮に入れておらず、ネットワークがどのような状況であっても同じアウトプットセレクションファンクションによりパケットの転送を行っていた。しかし、現実にはネットワーク中にホットスポットが発生する一方で空いているチャネルが多数存在するようなケースが生じる。ホットスポットの生成、通過をなるべく避けることがアウトプットセレクションファンクションの重要な役目であることはいうまでもない。これを解決するために、ネットワークの状況を反映しつつ、チャネルの特徴(使用条件)を考慮してネットワーク中の資源を有効活用する、Channel Characteristic Based (CCB) セレクションファンク

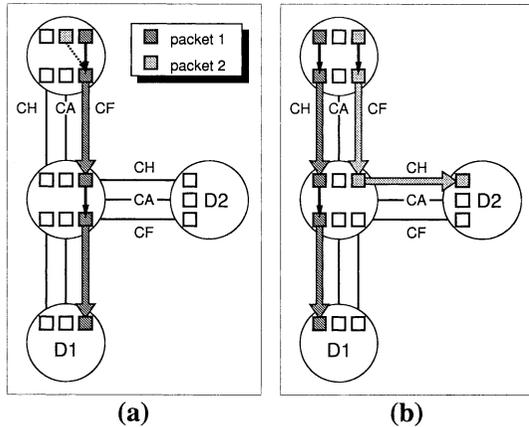


図4 ルーティング例  
Fig. 4 Routing example.

ションを提案する。

CCBは、ネットワークのチャンネルの状態によりチャンネルを選択し、そのバーチャルチャンネルの目的にあった使用率に制御する特徴を持つ。

### 3.1 CCBの提案

CCBはその性能向上をなしとげるために以下の4つの点を考慮する。

- (1) バーチャルチャンネルの中で最も使用条件が厳しいものを優先。
- (2) 最低次元におけるCFチャンネルの使用を抑制。
- (3) 高次元優先。
- (4) 周辺の状況の情報収集。

#### 3.1.1 バーチャルチャンネルの中で最も使用条件が厳しいものを優先

2.2節ではCH, CA, CFの3つのチャンネルの使用条件がそれぞれ異なることでバーチャルチャンネルごとに特徴が違うことを述べた。CCBでは、同一チャンネル内に出力しようとするバーチャルチャンネルが2つ以上あった場合、使用条件の厳しいものから使用し、使用条件の緩いバーチャルチャンネルしか選択できないパケットの転送効率を上げる。

使用条件の厳しいものを優先することによりパケットの転送効率が向上する根拠を示すために、図4にルーティング例を示す。図4(a)は最も使用条件が厳しいものを優先しなかった場合の例で、図4(b)はここで提案する最も使用条件が厳しいものを優先した場合の例である。それぞれのパケット(パケット1, パケット2)の目的地をD1, D2とすると、パケット1はD1に行くためには、 $y$ 次元だけを通ればよい。それに対してパケット2がD2に行くためには、 $x$ 次元と $y$ 次元を通らなくてはならない。また、パケット2が

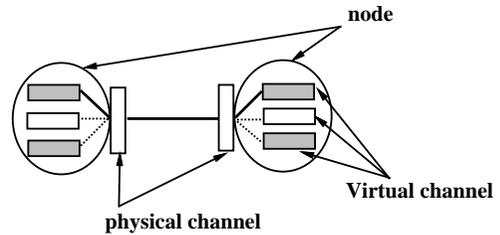


図5 単一ノードに複数のパケットが到着した例  
Fig. 5 An example of some packets arrived to a node.

$y$ 次元を通るためには、\*-channelの制約から $y$ -CFチャンネルしか使うことができない。

使用条件が厳しいチャンネルを優先しなかった場合(図4(a)),パケット1はCH, CAチャンネルがフリーな状態であるにもかかわらず、CFチャンネルを使用することになる。その結果パケット2は進むことができず、ノード内に停滞してしまう。逆に、最も使用条件が厳しいものを優先することでパケットの停滞を避けることが可能となる(図4(b))。

#### 3.1.2 最低次元におけるCFチャンネル使用を抑制

図3より、性能と $x$ -CFチャンネルの使用率の間には相関関係があり、 $x$ -CFチャンネルの使用を抑えているほど性能が高いことが観測された。この観測された現象を分析する。

まず、2次元( $x$ 次元と $y$ 次元)でのルーティングで、 $x$ -CFチャンネルがいつ使われているのかを分析する。3.1.1項で述べたように、最も使用条件の厳しいチャンネルから使用していく場合、 $x$ -CFチャンネルが使われるときは、確実に $x$ -CHか $x$ -CAチャンネルが使用されている。このような場合、図5のように同一物理チャンネルの中にパケットが複数存在する。

ここで、同一物理チャンネルに複数パケットが存在することがパフォーマンスの低下につながる原因を考察する。同一物理チャンネルに複数のパケットが存在する場合、1本の物理チャンネル上で複数のパケットをタイムシェアリングにより転送しなければならず、その結果物理チャンネル上にパケットが1つしかない場合に比べて各パケットごとの物理チャンネルの使用効率が低下し、性能は低下する。

したがって、 $x$ -CFチャンネルの使用を抑制することで同一チャンネルでのマルチプレクスの発生を抑えることが性能向上の鍵となることが予想される。

ここでは2次元でのルーティングを例にとって説明したが、ルーティングが多次元化しても同様であり、 $y, z$ 次元においてのルーティングなら $y$ 次元、 $x, y, z$ 次元のルーティングなら $x$ 次元というように、最低

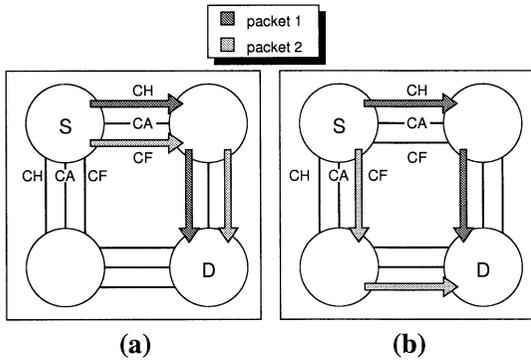


図6 2つのアウトプットセレクションファンクションの比較  
Fig. 6 Comparison between two output selection functions.

次元 (\*-channel の制約でパーチャルチャンネルが3本使用可能な次元)のCFチャンネルの使用を抑えることで性能の低下を防ぐ。

3.1.3 高次元優先

また、図3より、性能は  $y_{CF}$  チャンネルの使用率とも相関関係があり、性能が高いほど  $y_{CF}$  チャンネルの使用が多いことも観測できた。ここでも観測された現象について分析する。

2次元トラスでのルーティングにおいて  $y_{CF}$  チャンネルが使用される時は2つの場合が存在する。1つは  $x$  次元の残りホップが0になり、 $y$  次元のみにしか行けなくなった場合、もう1つは、 $x$  次元のホップが残っている状態で  $y$  次元にルーティングをしたときである。ただし、前者は  $y_{CH}$ ,  $y_{CA}$  チャンネルが使用中であったときに限り使用されるので、3.1.2 項の理由より性能の低下を招く。したがって、後者の  $x$  次元のホップが残っている状態で  $y$  次元にルーティングをしたときに  $y_{CF}$  チャンネルを使うことが性能向上の要因であることが分かる。

では、なぜ  $x$  次元のホップが残っている状態において  $y_{CF}$  チャンネルを使用すること、すなわち高次元チャンネルを優先的に使うことが性能向上に結び付くのだろうか。以下にその理由を考察する。

考察の際に、対照的な例である Dimension order のような低次元 (パーチャルチャンネルが3本使える次元) 優先のアウトプットセレクションファンクション (図6(a)) を例にあげて、高次元チャンネルを優先するアウトプットセレクションファンクション (図6(b)) と比較する。

図6(a)では、低次元を優先しているためパケットの流れが  $x$  次元に集中してしまい、パーチャルチャンネルのマルチプレクスが増加している。それに対し、

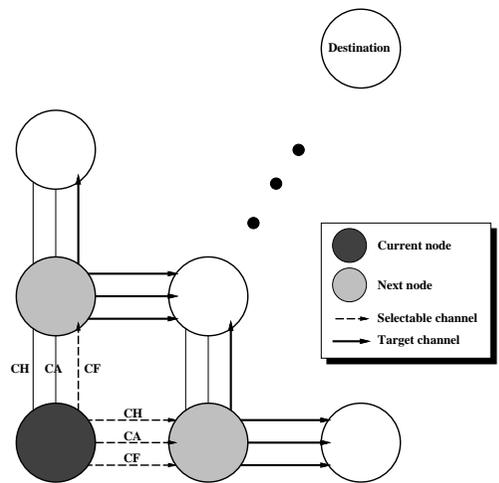


図7 周辺のノードの情報も調べた例  
Fig. 7 Collecting informations of neighboring nodes.

図6(b)ではパケットの流れは分散し、ネットワーク資源の利用を均一化している。以上の理由により、高次元チャンネルを優先的に使用することで性能は向上すると考えられる。

CCBでは例にあげた  $y_{CF}$  だけでなく、パーチャルチャンネルCFのみを使う高次元チャンネルを優先的に使用することでパフォーマンスの向上を図る。

3.1.4 周辺の状況の情報収集

LDは各ノードにそれぞれフリットが通過したとき、その数をカウントすることでネットワークの混雑状況を把握していた<sup>5)</sup>。しかしながら、各ノードは自分の情報を保持するだけで正確なネットワーク全体の混雑状況を把握することは不可能だった。そこで、CCBでは各パーチャルチャンネルがフリーであるか、ビジーであるかという情報を基として、隣りのノードの情報をあわせて持つことでより正確な混雑状況を把握する。

厳密には、隣りのノードとは、パケットが進む可能性のある次元の1歩先のノードである。周辺の状況の情報収集の例を図7に示す。図7はパケットが  $+x$  方向、 $+y$  方向に進もうとしている例で、この場合ではそれぞれ  $+x$  方向の1歩先のノード、 $+y$  方向の1歩先のノードのフリーなパーチャルチャンネルの本数の合計を調べる。目的地ノードの位置によって1歩先のノードのどの方向の情報を集めるかが変化するが、この例では  $x$  次元の3本(CH, CA, CF)のパーチャルチャンネル、 $y$  次元の1本(CF)のパーチャルチャンネルを調べている(図7中の太線の矢印で示されるチャンネル)。

ルーティングを行う際、1歩先のパーチャルチャンネルの情報を得ることにより時間を浪費しないために、それぞれのノードは1クロック前の各パーチャルチャ

ネルの状態を保持しておき、その値をルーティング時に1歩先のノードに渡すものとする。ハードウェア化に必要とするものは、わずか数ビットのバッファとそれらの転送用の専用線だけなので、莫大なハードウェアを要求しない。

1歩進んだノードの情報を得ることで、どれだけ正しい情報を得て、性能上昇が得ることができるか、その比較のために4章で周辺の情報収集を行わないS-CCBセレクションファンクションの評価も加えて評価する。S-CCBは1歩先の情報を収集しない点を除けばCCBと同一である。

### 3.2 CCBのアルゴリズム

ここでは、CCBの出力チャンネル決定のアルゴリズムについて説明する。CCBは、出力可能な次元が1つのみの場合には使われない。そのため、このアルゴリズムは、2次元以上の方向に出力チャンネルがあった場合を想定している。以下にCCBとS-CCBのアルゴリズムを示す。

#### 設計1 CCBのアルゴリズム

- (1) パケットを出力することのできる次元が1次元のみの場合、その次元を選択する。
- (2) パケットを出力することのできる次元が2次元以上の場合、それぞれの次元で使用されていないバーチャルチャンネルの本数を調べる。
  - (a) パケットが到着したノードの最低次元のバーチャルチャンネルの空きが1本、もしくは空いていない場合は、空いている次元のうちで最も高い次元を選択する。
  - (b) パケットが到着したノードの最低次元のバーチャルチャンネルの空きが2本、3本の場合は、1歩先のノードのバーチャルチャンネルの情報を使用し、空いているバーチャルチャンネルの合計が最も多い次元を選択する。もし、最も多い次元が複数ある場合は、高次元を優先する。
- (3) 選択した次元の中で最も使用条件の厳しいバーチャルチャンネルを選択する。

アルゴリズム中の(2).(a)は3.1.2項および3.1.3項を、(2).(b)は3.1.4項を、そして(3)は3.1.1項をそれぞれ適用している。

S-CCBのアルゴリズムは、CCBのアルゴリズムをシンプルにしたものである。それは、CCBのアルゴリズムの2.(b)のように1歩進んだノードの状況を見ることをしないものである。以下に、S-CCBのアルゴリズムを示す。

#### 設計2 S-CCBのアルゴリズム

- (1) パケットを出力することのできる次元が1次元のみの場合、その次元を選択する。
- (2) パケットを出力することのできる次元が2次元以上の場合、それぞれの次元で使用されていないバーチャルチャンネルの本数を調べる。
  - (a) パケットが到着したノードの最低次元のバーチャルチャンネルの空きが3本の場合、その次元を選択する。
  - (b) パケットが到着したノードの最低次元のバーチャルチャンネルの2本以下の場合、空いている次元の中で最も高い次元を選択する。
- (3) 選択した次元の中で最も使用条件の厳しいバーチャルチャンネルを選択する。

## 4. 評価

本章では、提案したCCB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。アウトプットセレクションファンクションはDuato's protocol<sup>6)</sup>上で実装した。あわせて、6つのアウトプットセレクションファンクション(Dimension order, Random, Zigzag, LD, S-CCB, CCB)について比較、検討を行う。

ネットワークサイズ、バーチャルチャンネルの本数、パケット長はパラメータを変更することで設定し、それぞれのノードは、プロセッサ、リクエストキュー、ルータにより構成され、各ノードはルータに接続された双方向チャンネルにより隣りのノードと接続されている。また、ルータは、チャンネルバッファ、クロスバ、リンクコントローラ、制御回路から構成されていると想定する。

### 4.1 シミュレーション条件

本シミュレータでは目的地ノードは、以下のトラフィックパターンにより決定する。

- Uniform  
すべての目的地ノードはランダムに決定され、均一に分散される。
- Bit reversal  
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 $(a_{n-1}, a_{n-2}, \dots, a_1, a_0)$ へパケットを送る。
- Matrix transpose  
ノード番号が $(x, y)$ のノードは $(k - y - 1, k - x - 1)$ ( $k$ は各次元内のノード数)のノードにパケットを送る。

本シミュレータでは、2つの指標により評価を行った。

- ネットワークレイテンシ：あるノード  $p$  がパケットの最初のフリットを入力バッファに挿入した時刻を  $t_0$ 、目的地ノードである  $q$  がパケットの最後のフリットを受け取った時刻を  $t_1$  とする。ここで、 $T_{lat}(p, q) = t_1 - t_0$  をネットワークレイテンシと呼び、ネットワークの性能を測る指標とする。

表 1 シミュレーション条件  
Table 1 Simulation parameters.

実行時間	50,000 clk
ネットワーク	2次元トラス, 3次元トラス
ネットワークサイズ	$32 \times 32$ (1024 ノード), $8 \times 8 \times 8$ (512 ノード)
バーチャルチャンネル数	3
パケット長	128 フリット
フロー制御	ワームホール
フリットを送るのに かかる時間	3 クロック
トラフィックパターン	Uniform Matrix transpose Bit reversal

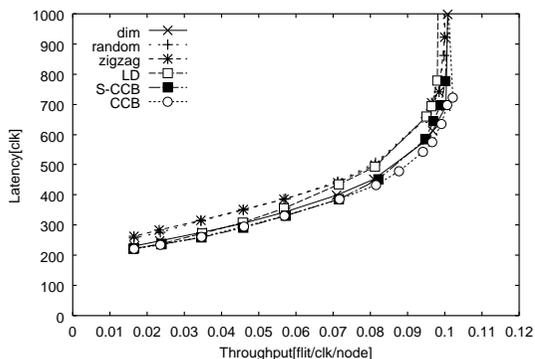
- スループット：ここでは、ネットワークのスループットを各ノードのルータがパケット中のフリットを次のノードのルータに転送する確率と定義する。すなわち、各ルータが毎クロック、フリットを転送するならば、スループットは 1.00 である。リクエストキューがあふれた場合は、シミュレーションは停止する。

シミュレーションに用いた条件を表 1 に示す。また、フリットを送るのにかかる時間は、1 クロック目はルーティング、2 クロック目はフリットを入力チャネルから出力チャネルまでクロスバを通して転送する、3 クロック目は、次のノードまでフリットを転送するために 3 クロック必要とする。

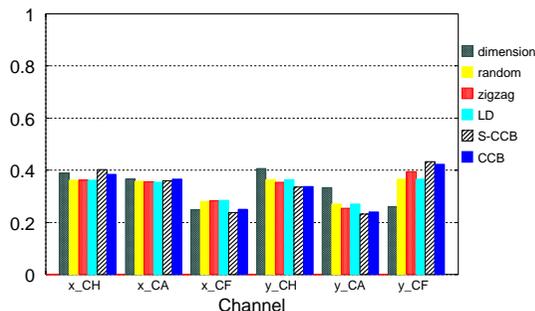
シミュレーションで初めの 5,000 クロックはネットワークが安定せず、想定した負荷に達していないと考えられるため評価の対象外とした。

#### 4.2 Uniform traffic

Uniform traffic での評価を図 8 と、図 9 に示す。各図 (a) の横軸はスループット、縦軸はレイテンシを表



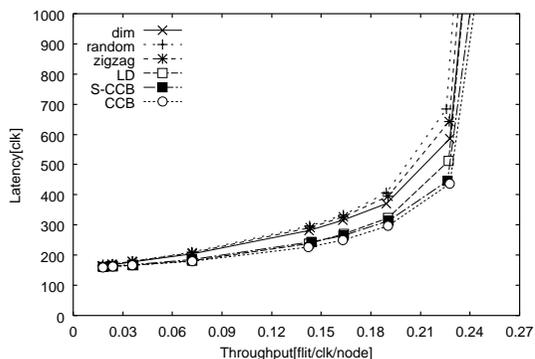
(a)  $32 \times 32$  2D torus ( 1024 ノード )



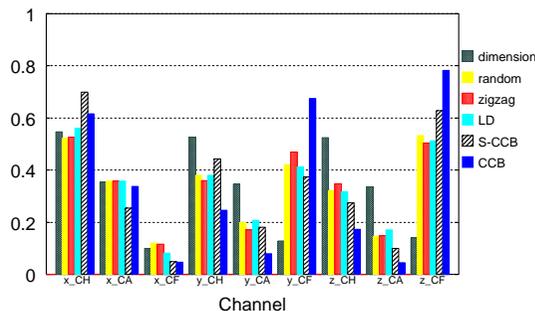
(b)  $32 \times 32$  2D torus チャンネル分布

図 8 Uniform traffic  $32 \times 32$  2次元トラス

Fig. 8 Uniform traffic ( $32 \times 32$  2D torus).



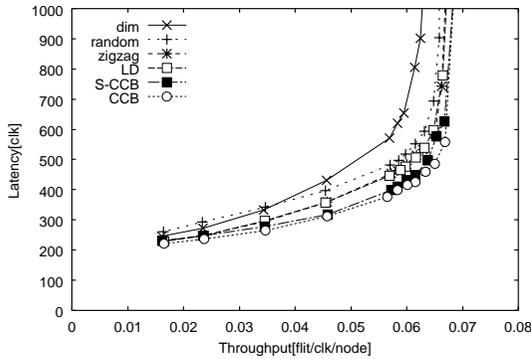
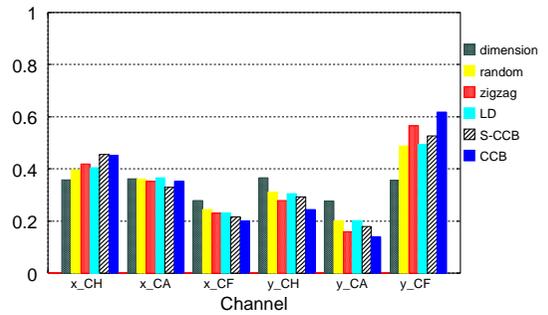
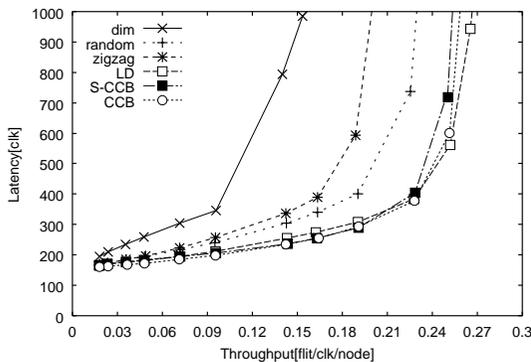
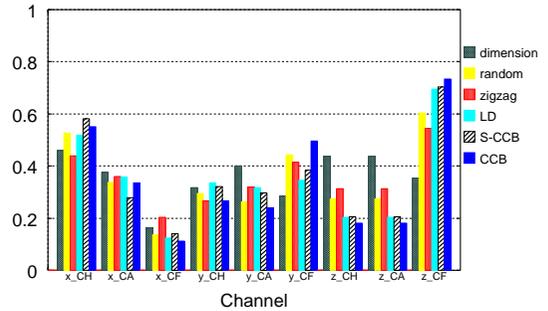
(a)  $8 \times 8 \times 8$  3D torus ( 512 ノード )



(b)  $8 \times 8 \times 8$  3D torus チャンネル分布

図 9 Uniform traffic  $8 \times 8 \times 8$  3次元トラス

Fig. 9 Uniform traffic ( $8 \times 8 \times 8$  3D torus).

(a)  $32 \times 32$  2D torus (1024 ノード)(b)  $32 \times 32$  2D torus チャンネル分布図 10 Matrix transpose traffic  $32 \times 32$  次元トーラスFig. 10 Matrix transpose traffic ( $32 \times 32$  2D torus).(a)  $8 \times 8 \times 8$  3D torus (512 ノード)(b)  $8 \times 8 \times 8$  3D torus チャンネル分布図 11 Matrix transpose traffic  $8 \times 8 \times 8$  次元トーラスFig. 11 Matrix transpose traffic ( $8 \times 8 \times 8$  3D torus).

している。各図 (b) はそのチャンネルの使用状況を示す。チャンネルの使用状況は、各アウトプットセレクションファンクションにおけるネットワークが飽和する直前の負荷におけるグラフで、縦軸は各次元でパケットが選択したパーチャルチャンネルの割合を示し、横軸はそのパーチャルチャンネルを示している。

図 8 (a), (b) は 1024 ノード 2 次元トーラスでの評価、図 9 (a), (b) は 512 ノード 3 次元トーラスでの評価である。

図 8, 図 9 とともに、CCB が最も高い性能を示している。しかしながら、Uniform traffic ではすべてのノードにおいてパケットが等確率で生成されるため、特に選択するチャンネルが 2 つしかない 2 次元トーラスでは、アウトプットセレクションファンクションごとにおけるチャンネルの分布に大きな偏りが見られない。その結果として図 8 (a) のグラフでは性能の劇的な差が生まれなかった。それに対して 3 次元トーラスでは、チャンネルの分布に差が付き、ネットワークが飽和

する直前の点において、 $x_{CF}$  チャンネルの使用を抑え、 $y_{CF}$ ,  $z_{CF}$  チャンネルを多く使うことで、CCB はレイテンシをかなり低く抑えることができている。

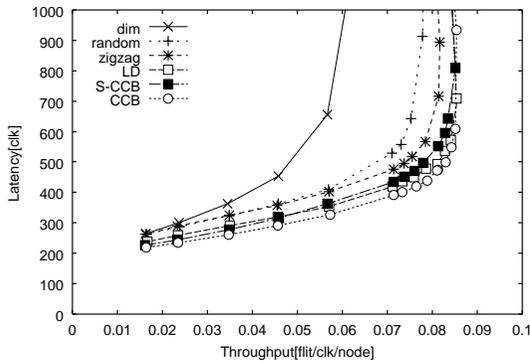
また、CCB と S-CCB を比較すると、すべての場合において、CCB が性能面で上回っている。これは、周囲の情報を得たことにより、より正確なルーティングを可能にしていることを表している。

#### 4.3 Matrix transpose traffic

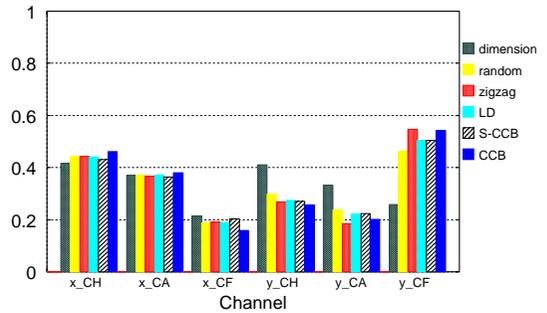
次に、Matrix transpose での 1024 ノード 2 次元トーラスでの評価を図 10 に、512 ノード 3 次元トーラスでの評価を図 11 に示す。

図 10, 図 11 とともに、Dimension order が最も低いスループットでネットワークの飽和を引き起こし、CCB が高い性能を示している。

同じ Duato's protocol を用いているにもかかわらず、アウトプットセレクションファンクションによる性能差が Uniform traffic に比べ著しい。このことは偏りが発生するトラフィックパターンにおいて、各ア



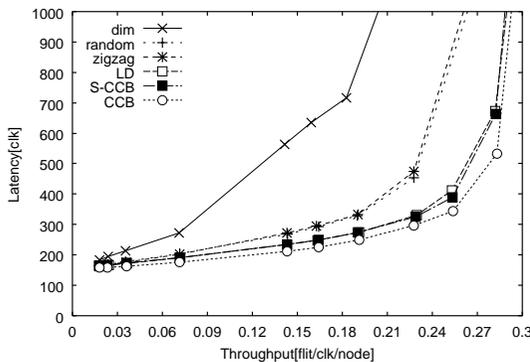
(a) 32 × 32 2D torus (1024 ノード)



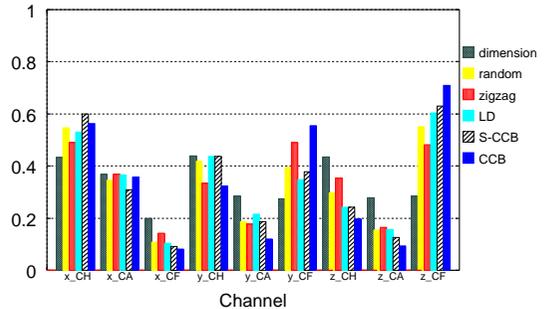
(b) 32 × 32 2D torus チャンネル分布

図 12 Bit reversal traffic 32 × 32 2 次元トーラス

Fig. 12 Bit reversal traffic (32 × 32 2D torus).



(a) 8 × 8 × 8 3D torus (512 ノード)



(b) 8 × 8 × 8 3D torus チャンネル分布

図 13 Bit reversal traffic 8 × 8 × 8 3 次元トーラス

Fig. 13 Bit reversal traffic (8 × 8 × 8 3D torus).

アウトプットセレクションファンクションがどれだけ有効かを表している。

他のトラフィックパターンでの評価と異なる点としては、図 11 で LD が最も高い性能を示した点がある。これは偏りのあるトラフィックでさらに多次元化したことにより、高負荷時に LD の利点が大きく働いたためであろう。

チャンネルの使用状況を見てみると、特に高い性能を示した CCB と LD では、最低次元である  $x$  次元の CF チャンネルの使用が抑えられており、最高次元である  $z$  次元の CF チャンネルの使用が多いことが図 11 (b) に示されている。これは、偏りのあるトラフィックにおいて、パーチャルチャンネル単位で制御することにより、存在するネットワーク資源を有効に使うことができていたことを表している。

また、S-CCB と CCB を比較すると、Uniform traffic のときと同様、すべての点において CCB が上回っている。これは、データの流れに偏りのある Matrix

transpose traffic においても CCB が有効なことを表している。

#### 4.4 Bit reversal traffic

次に、Bit reversal traffic での 1024 ノード 2 次元トーラスでの評価を図 12 に、512 ノード 3 次元トーラスでの評価を図 13 に示す。

Bit reversal traffic ではトーラス内に極端な偏りは発生しないが、小規模の偏りが多数発生しやすくなる。そのため、Matrix transpose traffic と同様にアウトプットセレクションファンクションの動きが Uniform traffic のときに比べより顕著なものとなる。

図 12, 図 13 とともに Dimension order が最も低いスループットでネットワークの飽和を引き起こす一方で、CCB が最も高い性能を示している。チャンネルの使用状況を見てみると、S-CCB, CCB は 3 次元トーラス上での Uniform traffic の評価と同様に、最低次元である  $x$ -CF チャンネルの使用が抑制され、高次元である  $y$ -CF,  $z$ -CF チャンネルの使用率が高いことが分

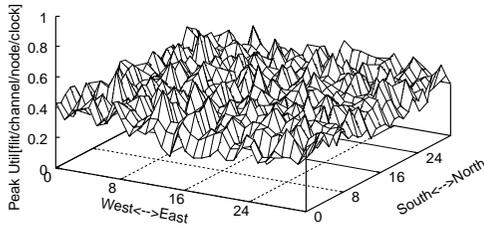


図 14 Dimension order

Fig. 14 Peak channel util. for dimension order.

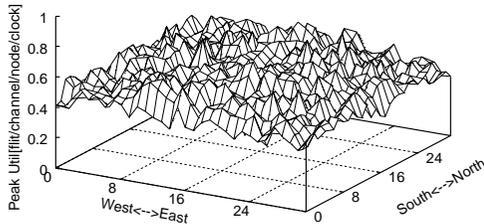


図 15 Random

Fig. 15 Peak channel util. for random.

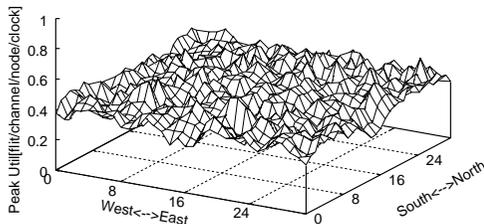


図 16 CCB

Fig. 16 Peak channel util. for CCB.

かる。

また、S-CCB と CCB を比べると、Bit reversal traffic のときもまたすべての点において CCB が上回っている。これは Matrix transpose traffic のときと同様、ネットワーク上でデータの流りに偏りのある状態において CCB が有効なことを表している。

以上のように CCB は、アウトプットセレクションファンクションに必要な機能を果たし、Bit reversal traffic において、既存のものに比べ大幅な性能向上を果たしている。

#### 4.5 物理チャンネル利用率

次に各物理チャンネルの利用率について評価を行う。32×32 次元トラス上でトラフィックパターンに Bit reversal を用いて 3 つのアウトプットセレクションファンクション (Dimension order, Random, CCB) でのチャンネルの利用率を求めた (その性能評価は図 12 (a) に示される)。図 14, 図 15, 図 16 は各アウトプットセレクションファンクションでの物理チャンネルの利用率を示している。

図 14, 15, 16 における  $z$  軸はピークチャンネル利用率を示し、 $x$  軸と  $y$  軸はネットワーク中のそれぞれのノードの相対的な位置を示している。また、ピークチャンネル利用率とは、スループットを 1000 クロックごとに計算し、ネットワークが飽和する直前のときの各物理チャンネルの利用率を指すものである。アウトプットセレクションファンクションがネットワークトラフィックを均一にできた場合、それぞれのノードでのピークチャンネル利用率は下がり、均一化されるといえる。

図 14, 15, 16 を見ると、CCB は他のセレクションファンクションと比べるとピークチャンネル利用率が均一に近づいていることが分かる。これは、ネットワーク資源を有効に使用していることを示しており、性能向上の一因を担っていることが分かる。

## 5. まとめ

今回はネットワークの状態により、出力方向とそのバーチャルチャンネルを選択するアウトプットセレクションファンクションである、CCB を提案した。CCB はそれぞれのバーチャルチャンネルの特徴とその性能の關係に注意を払うことで、使用されているバーチャルチャンネルの偏りをできるだけ均一に近づけ、ネットワークの持つ資源を有効に使うことが特徴である。シミュレーションの結果より、CCB はすべてのトラフィックパターンにおいて従来のアウトプットセレクションファンクションより高い性能を実現することが分かった。

また、周辺ノードの状況を把握しない S-CCB とのパフォーマンスの比較により、周辺の状況を把握することが与える影響についても調べた。シミュレーション評価より、特にトラフィックの偏りが大きい場合には CCB の優位性が明らかになった。周辺ノードの状況を把握することはハードウェアコスト、およびルーティングにかかるクロック数の増加を招くが、CCB ではそれらを低減するために 1 クロック前の情報を利用するなどの工夫を行った。以上のことから、S-CCB よりも CCB を利用した方が有利であると考えられる。

今後は様々な種類のパケット長を混合した場合や、トラフィックパターンの変化、インストラクションレベルシミュレータを用いるなど、今回評価をとっていない条件で評価を行う予定である。

## 参考文献

- 1) 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).
- 2) Wu, J.: An Optimal Routing Policy for Mesh-Connected Topologies, *Proc. International Conference on Parallel Processing*, Vol.1, pp.267–270 (1996).
  - 3) Ni, L.M. and McKinley, P.K.: A Survey of Wormhole Routing Techniques in Direct Networks, *IEEE Trans. Comput.*, No.2, pp.62–76 (1993).
  - 4) Martinez, J.C., Silla, F. and Duato, J.: On the Influence of the Selection Function on the Performance of Networks of Workstations, *Proc. ISHPC*, pp.292–300 (2000).
  - 5) 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).
  - 6) 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, pp.1055–1067 (1995).
  - 7) 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 13 年 3 月 29 日受付)

(平成 13 年 12 月 18 日採録)



杉山 伸悟

昭和 53 年生 . 平成 13 年三重大学工学部情報工学科卒業 . 現在 , 静岡日電ビジネス株式会社勤務 . 並列計算機における相互結合網 , ルーティングアルゴリズムに関する研究に従事 .



舟橋 啓 (正会員)

昭和 46 年生 . 平成 7 年慶應義塾大学工学部電気工学科卒業 , 平成 12 年同大学大学院理工学研究科計算機科学専攻後期博士課程修了 . 工学博士 . 現在 , 三重大学工学部助手 . 並列計算機における相互結合網 , ルーティングアルゴリズムに関する研究に従事 . IEEE , IEEE-CS 各会員 .