情報処理学会論文誌　コンピューティングシステム　Vol.5　No.3　112–123 (May 2012)

予測を用いたNoC向け混雑回避機構

谷口 将一[1,a)]　松谷 宏紀[1]　山崎 信行[1]

受付日 2011年10月7日, 採録日 2012年1月30日

概要：メニーコアプロセッサにおいては，スケーラビリティの高い通信網として Network-on-Chip（NoC）が広範囲に利用されている．NoC ではハードウェアコストを抑えるため，単純な固定型ルーティングアルゴリズムがよく用いられる．しかし，トラフィックパターンに偏りがあると固定型ルーティングは混雑を回避することができず，ネットワークの性能が低下しやすい．本論文では，周辺の混雑を予測することでネットワーク性能を向上させることができる低コストなオンチップルータを提案する．提案ルータは現在と未来の両方の混雑情報をルータ間で互いに交換し合う．受信した混雑情報と自身がローカルに持っている混雑情報から，周辺の混雑情報を生成することで，混雑を予測し回避する．シミュレーションによる評価の結果，提案ルータは，近傍もしくは広域の混雑情報を用いて混雑を回避する既存の手法より，スループットとレイテンシを改善することができた．RTL による実装と論理合成を行ったところ，提案ルータの面積オーバヘッドはたかだか 5.1%であり，各ルータポートごとの配線のオーバヘッドはたかだか 3 本であった．

キーワード：相互結合網

A Prediction-based Congestion Avoidance Scheme for NoCs

Masakazu Taniguchi[1,a)]　Hiroki Matsutani[1]　Nobuyuki Yamasaki[1]

Received: October 7, 2011, Accepted: January 30, 2012

Abstract: Network-on-Chips (NoCs) have been widely used as scalable interconnection networks for many-core processors. Although NoCs typically employ simple deterministic routing algorithms to reduce the hardware complexity, such deterministic algorithms cannot avoid traffic congestion when traffic patterns have localities, resulting in a lower network performance. In this paper, we propose an on-chip router with Predictor for Regional Congestion (PRC) in order to improve the network throughput with modest hardware overhead. The proposed routers exchange their past and predicted future congestion information with each other. They synthesize their regional congestion information based on the local and received information in order to predict and avoid congestion. The simulation results show that the proposed routers improve the throughput and the latency compared to congestion-aware routers that exploit local or global congestion information. The RTL design of the proposed router shows that the area overhead is only 5.1% and additional wires for each router port is only three.

Keywords: interconnection network

1. はじめに

半導体技術の進歩にともない，1 チップに多数のコアを搭載することが可能となった [1], [2], [3]．現在，このようなメニーコアプロセッサにおける通信網として Network-on-Chip（NoC）[4] が広範囲に利用されている．NoC は従来のオンチップバスよりも高いスケーラビリティが期待できるものの，バスと比べると実装面積が大きくなりがちである．したがって，NoC を構成するオンチップルータのハードウェアコストを抑えつつ，ネットワークの性能を向上させることはつねに重要な研究テーマである．

結合網におけるルーティングアルゴリズムは，一般的に，

[1] 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa 223–8522, Japan
[a)] masakazu@ny.ics.keio.ac.jp

© 2012 Information Processing Society of Japan

固定型ルーティング，ランダム型ルーティング，適応型ルーティングに分類される [5]．固定型ルーティングでは，送信元と宛先の組ごとに使用する経路が固定されている．一方，ランダム型ルーティングではランダムに，適応型ルーティングではネットワークの状態に応じて動的に経路が変化する．現状のメニーコアプロセッサにおける NoC では，実装の容易さ，面積コストの小ささから単純な固定型ルーティングアルゴリズムが用いられる [2], [3]．しかし，このような単純な固定型ルーティングはトラフィックパターンに偏りがあると混雑を回避することができず，ネットワーク性能が低下しやすい．一方，ランダム型ルーティングおよび適応型ルーティングはインオーダ転送を保証できないものの，動的に経路を変更するため負荷を分散させやすく，トラフィックパターンの偏りに強い．ランダム型ルーティングはネットワークの状態を考慮しないため，その性能向上には限界があるが，適応型ルーティングはネットワークの状態を考慮し，比較的空いている経路を使用するため様々なトラフィックに適応可能である．このように，ルーティングアルゴリズムによる性能向上とその複雑さ，実装コストの間にはトレードオフが存在する．

そこで，本論文では，少ないハードウェアオーバヘッドで様々なトラフィックに適応可能な適応型ルータを実現することを目的とする．そのために，周辺の混雑を予測することでネットワーク性能を向上させることができる低コストなオンチップルータ（PRC：On-chip Adaptive Router with Predictor for Regional Congestion）を提案する．PRC ルータは現在と未来の両方の混雑情報をルータ間で互いに交換し合い，受信した混雑情報と自身がローカルに持っている混雑情報から周辺の混雑情報を生成し，この情報をもとにルーティングを行うことで混雑を回避する．

本論文の構成は次のとおりである．まず，2 章で適応型ルーティング向けの既存の混雑回避手法について調査し，それらのトレードオフについて述べる．次に，3 章で PRC ルータを提案し，4 章でその実装の詳細について述べる．5 章で設計および実装したルータの評価を行い，6 章で本論文をまとめる．

2. 適応型ルーティング向け混雑回避手法

本研究の目的は，少量のハードウェアオーバヘッドで効率的に適応型ルーティングを実現することである．デッドロックフリーな適応型ルーティングアルゴリズムとして Turn Model [6]，Odd Even Turn Model [7]，Duato's protocol [8] など様々なアルゴリズムが提案されており，本論文でもこれらの利用を前提とする．適応型ルーティングでは，これらのルーティングアルゴリズムによって得られたデッドロックフリーなルーティング経路の中からネットワークの混雑状況に応じて経路を動的に選択することで混雑を回避する．このための混雑回避手法は，近傍ルータ（ローカル）の混雑情報のみを用いる手法とネットワーク全体のグローバルな情報を用いる手法に大別できる．

2.1 ローカルな情報のみ用いる手法

ここでは，混雑回避のために 1 ホップ先までの混雑情報を利用する手法を「ローカルな情報のみ用いる手法」とする [9], [10], [11], [12]．これらは，クレジットベースのフロー制御によって，隣接ルータの情報（入力バッファの利用率など）が利用可能であることを想定している．適応型ルーティングにおける多くの Output Selection Function（OSF）はこの手法に属する．以下にローカルな情報のみを用いる混雑回避手法の例をあげる．

ランダム選択機構　出力可能なポートが複数存在する場合，その中からランダムにポートを選択する [9]．最も単純な機構ではあるが，パケットの経路が固定されないため，ある程度負荷を分散させることができる．

空き VC 数最大出力ポート選択機構　空の仮想チャネルの数を各出力ポートごとに集計し，その数が最も多かった出力ポートを選択する [9]．パケットが比較的空いている経路を通ることができるため性能が向上する．また，ルータ間に新たに配線を追加する必要はなく，比較的ハードウェアオーバヘッドは小さい．

空きバッファ数最大仮想チャネル選択機構　各仮想チャネルについて空きバッファの数を数え，最も多かった仮想チャネルを選択する [10]．比較的空いている仮想チャネルを選ぶことができるため，性能が向上する．また，クレジットベースのフロー制御を用いている場合はルータ間の配線を追加する必要はなく，比較的ハードウェアオーバヘッドは小さい．

これらの手法は各ルータが自分自身で持っている情報のみ使用するため，ハードウェアオーバヘッドは小さいが，ローカルな混雑情報のみを利用するため，その性能向上には限界がある．

2.2 グローバルな情報を用いる手法

グローバルな情報を用いる手法として 2008 年に Regional Congestion Awareness（RCA）[13] が提案されている．適応型ルーティングでは混雑情報の精度がネットワークの性能に影響するが，RCA は混雑情報を伝播し集約することで，ネットワークの広域の混雑情報を利用可能にしている [13]．各ルータは隣接ルータからの混雑情報を受信し，その情報と自身の持っている情報を合成してさらに他の隣接ルータに伝播させる．この情報伝達をパケットフローの上流方向に再帰的に行うことで広域の混雑情報が利用可能となる．広域の混雑情報の利用により，様々なトラフィックにおいて性能が向上する．ただし，混雑情報の伝播および集約には多数の加算器やシフタが必要であったり，物理チャネルあたり 8 本または 16 本のルータ間のワイヤの追

© 2012 Information Processing Society of Japan

加が必要であったりするなど，ハードウェアオーバヘッドは大きい．また，遠方の情報を使用する場合，その情報伝達には時間がかかる．よって，その情報が使用されるころには情報発信元の混雑状況は変わっていて，その情報がすでに役に立たない可能性がある．なお，RCA のほかにも，グローバルな情報を用いる混雑回避手法は，2008 年以降いくつか提案されている [14], [15], [16]．これらの関連研究と提案手法の違いについては 5.3 節で議論する．

2.3　性能とオーバヘッドのトレードオフ

適応型ルーティングの性能とそのオーバヘッドの間にはトレードオフの関係がある．ローカルな情報のみを用いる手法はハードウェアオーバヘッドは小さいが，その性能向上には限界がある．一方，RCA のようにグローバルな情報を用いる手法は様々なトラフィックに対応できるが，ハードウェアオーバヘッドは大きく，また，いかに速くグローバルな混雑情報を伝播させるか（情報の新しさ）に検討の余地がある．

そこで，本論文では，RCA ほど広範囲でない中程度の距離（2 ホップ程度）の混雑情報を用い，ハードウェアコストを抑えつつネットワーク性能の向上を狙うことを考える．さらに，混雑情報の新しさと精度を向上させるため，予測や情報の先送りなどの手法を導入する．

3.　予測を用いた混雑回避機構

前章で述べたように，ルーティング手法のネットワーク性能とそのハードウェアコストにはトレードオフの関係がある．そこで，本論文では周辺の混雑を予測するオンチップ適応型ルータ（On-chip Adaptive Router with Predictor for Regional Congestion：PRC）を提案する．PRC は，中程度の距離（本論文の実装では 2 ホップ）の混雑状況の情報を使用して混雑を回避し，さらに予測と情報の先送りを用いて，混雑情報の精度を向上させる．

PRC は投機ルータ [17] のような既存のルータへの僅かな変更のみで実装できる軽量ルータである．PRC および投機ルータのパイプライン構成は同じで，RC（Route Computation），VSA（Virtual Channel Allocation と Speculative Switch Allocation）および ST（Switch Traversal）の 3 段構成である．また，PRC は 2D-Mesh で最短適応型ルーティングを行うことを前提としている．デッドロック回避の手法には，様々なルーティングアルゴリズムが利用可能であるが，今回は Turn Model [6] より West-First ルーティングを用いる．以降，PRC の設計について，例を示しながら順を追って説明する．ここで，座標は最北西を原点 $(0,0)$ と定義し，座標 (x,y) に位置するルータを $R(x,y)$ と表記する．また，$R(x1,y1)$ と $R(x2,y2)$ の間のチャネルを $C(R(x1,y1),R(x2,y2))$ と表記する．また，混雑していないチャネルを実線で，混雑しているチャネルを破線で示す．

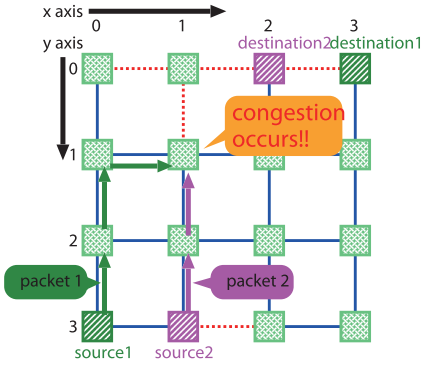


図 1　パケットの衝突
Fig. 1　Packet conflict.

3.1　パケットの衝突

本研究ではパケットの衝突を回避することにより，ハードウェアコストを抑えつつネットワーク性能の向上を図る．図 1 にパケットの衝突の様子を示す．この例では，source1 から destination1 に，source2 から destination2 にそれぞれパケットが送信されており，$R(1,1)$ において，パケットの衝突が起こっている．もし source2 から送られたパケットが $R(1,2)$ において，東方向にルーティングされていればこの衝突は回避できるが，$R(1,2)$ は，source1 から送られたパケットが $R(1,1)$ に到着することを知らないため，この衝突を回避することはできない．このような衝突は従来手法のようにパケットフローの上流に混雑情報を伝播させるだけでは回避することはできない．これは，パケットフローの上流にのみ情報を伝播させる場合，実際に混雑が生じた後でしか混雑を回避できないためである．

3.2　Ahead Information の生成

そこで，パケットを送信するという情報をパケットフローの下流に，実際のパケットに先行して送信することを提案する．各ルータはパケットを受信し，そのパケットをどちらに送信するのかを決めると，つまり RC ステージが完了すると，すぐさま実際のパケットの送信に先行して，パケットを送信するという情報 1 ビットをパケットフローの下流に先送りする．この情報を **Ahead Information** と定義する．この情報はパケットがそのルータから抜けきるまで，流したままの状態となる．

ここで，Ahead Information を送信するタイミングには，1) RC ステージの後，2) RC ステージと同時の 2 種類が考えられ，1) の場合は実際にパケットが送信される 2 サイクル前，2) の場合は 3 サイクル前となる．1) の場合，Ahead Information を受信したルータ（ルータ X とする）が周囲の混雑情報を合成し，自ノード（ルータ X）におけるルーティング（混雑回避）に反映させるまでにさらに 1 サイクルかかる．つまり，ルータ X において，混雑情報をルーティングに反映させる前に RC ステージを通過したパケットに関しては，Ahead Information を受信しているにもか

© 2012 Information Processing Society of Japan

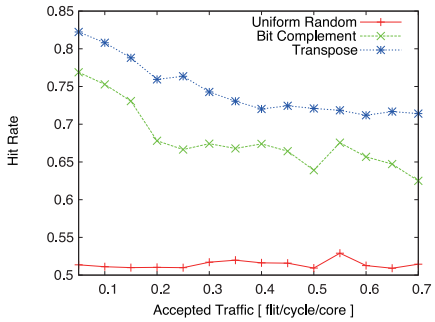


図 2　予測ヒット率
Fig. 2　Hit rate of prediction.

かわらずそれを混雑回避に活かせない．

一方，本論文では予測を用いることで 2) の方式を実現する．これによって，ルータ X は隣接ノードのパケットが RC ステージを通過したと同時にその旨を示す Ahead Information を受信でき，同サイクルにルータ X に入力されたパケットのルーティングにこの Ahead Information を反映できるようになる．**図 2** に各トラフィックにおける予測の精度を示す．評価環境は，トポロジは 4×4 2D-Mesh，仮想チャネル数は 2，バッファサイズは 4 フリット，バースト性は平均 4 パケット，パケットサイズは 5 フリットとした．予測の精度は 51〜82%程度と比較的高く，2) の方式によって良好な結果が得られている．

ここで使用する予測アルゴリズムについては 3.5 節で解説する．

3.3　Predicted Transmitted Information の生成

Ahead Information を受信したルータは，そのパケットがどちらに送信されるのかを予測し，その予測情報と，自身がローカルに持っている情報を合成して 1 ホップ先までの未来の混雑情報を生成する．この未来の混雑情報を **Predicted Transmitted Information** と定義する．Predicted Transmitted Information は，各ビットが各出力ポートの混雑状況に対応する 4 ビットのベクトル情報である．この 1 ホップ先までの未来の混雑情報を，さらに隣接ルータに送信する．Predicted Transmitted Information を受信した隣接ルータは 2 ホップ先までの未来の混雑情報を知ることができる．

3.4　各情報の伝播のタイミング

表 1 にパケットを受け取ったルータの状態と，そのルータが発信した混雑情報，および，その混雑情報を用いて生成された情報が伝播するタイミングを示す．

ただし，RC は Route Computation，VSA は Virtual Channel Allocation と Switch Allocation，ST は Switch Traversal，LT は Link Traversal，AI LT は Ahead Information の Link Traversal，make PTI は Predicted Transmitted Information を生成している状態，PTI LT は Predicted Transmitted Informatio の Link Traversal，make RCI は Regional Congestion Information を生成している状態を示す．また，**図 3** に以上の情報を用いた混雑回避の例を示す．表 1 と図 3 の Cycle n はそれぞれ対応している．

表 1　動作のタイミング
Table 1　Timing of operations.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4
router	RC	VSA	ST	LT
(a)	AI LT	make PTI	PTI LT	make RCI
(b)		AI LT	make PTI	PTI LT
(c)			AI LT	make PTI
(d)				AI LT

各サイクルでの，各ルータの状態，各情報の伝播について順に説明する．

Cycle 1　$R(0,1)$ は $R(0,2)$ からパケットを受信し，そのパケットを東側にルーティングすると予測すると，すぐさま $R(1,1)$ に 1 ビットの Ahead Information(a) を伝播させる．

Cycle 2　$R(1,1)$ は $R(0,1)$ からの Ahead Information(a) を受け取り，そのパケットが東側の $R(2,1)$ にルーティングされると予測している（Prediction (a)）．この $R(1,1)$ がローカルに持っていた情報と，予測した情報を合わせたものが Predicted Transmitted Information(a) である．一方，$R(0,1)$ は RC ステージを終えて，本当にそのパケットが東側にルーティングされることが分かり，$R(1,1)$ に 1 ビットの Ahead Information(b) を伝播させている．

Cycle 3　$R(1,1)$ は Cycle 2 で生成した Predicted Transmitted Information(a) を $R(1,2)$ に伝播している．これと同時に，$R(1,1)$ は Cycle 2 で受け取った Ahead Information(b) とローカルに持っていた情報を合わせ，さらに新しい Predicted Transmitted Information(b) を生成する．また，$R(0,1)$ はパケットの送信が終わるまで，Ahead Information を送信し続ける（Ahead Information(c)）．

Cycle 4　$R(1,2)$ は Cycle 3 で受信した Predicted Transmitted Information(a) を用いて，ルーティングを行うことができる．Predicted Transmitted Information(a) には，Cycle 2 で $R(1,1)$ が予測した $C(R(1,1), R(2,1))$ が混雑するという情報が含まれているため，$R(1,2)$ は $R(1,3)$ から受け取った packet 2 を，北側ではなく東側にルーティングできる．PRC はこのようにしてパケットの衝突を回避できる．また，これと同時に，新たな Predicted Transmitted Information(b) の伝播，Prediction(c) による Predicted Transmitted Information(c) の生成，Ahead Information(d) の伝播が行われる．

© 2012 Information Processing Society of Japan

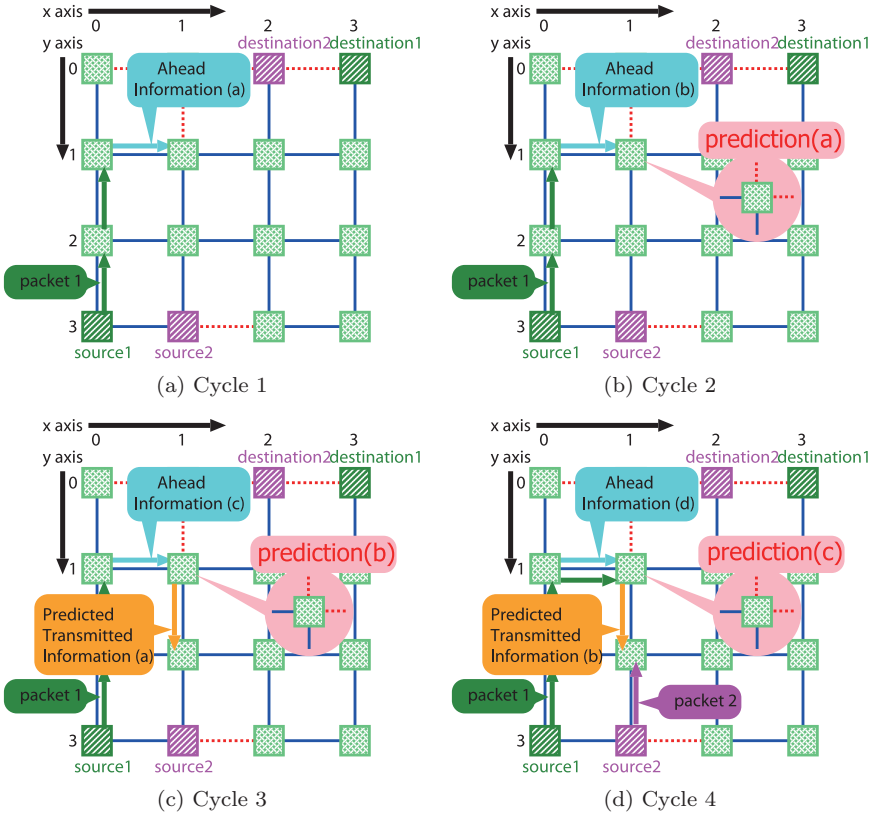


図 3　動作例
Fig. 3　Operation example.

3.5　出力ポートの予測

出力ポートの予測は各入力チャネルごとに行う．2 回連続で同じ方向にルーティングを行った場合，次回以降も同じ方向にルーティングすると予測する．これはプロセッサにおける 2 ビット予測器と同じ発想で，多量のパケットが同じ方向にルーティングされる中，たった 1 つのパケットが異なる方向にルーティングされたとしても，以降のパケットは多量のパケットと同じ方向にルーティングされるという思想に基づく．ここでは，3 ビット予測器と同じ発想で 3 回の同じ方向へのルーティングにより予測の方向が切り替わる予測器を用いなかった．これは，プロセッサの場合，予測の方向は分岐するか否かの 2 方向であるが，本研究で前提としている最短適応型ルーティングを行う 2D-mesh の NoC の場合，全部で 4 方向へのルーティングがありうる．この場合，予測の方向が極端に切り替わりにくくなってしまう．そこで，本論文では 2 回の同じ方向へのルーティングで予測方向が切り替わる予測器を採用した．

ハードウェアコスト削減のため，Ahead Information の生成と，Predicted Transmitted Information の生成には，同一の予測アルゴリズムを用い，同一の予測器を共有して使用する．

3.6　使用する混雑情報のホップ数

提案手法は 3 ホップ以上先の混雑情報を使用するように拡張可能であるが，本論文の実装で使用する混雑情報を 2 ホップまでに制限したのは以下の理由による．まず，物理チャネルあたりの配線コストに関して，たとえば，3 ホップの混雑情報を交換する場合，Predicted Transmitted Information に関して，1 ホップ目のために 3 ビット，2 ホップ目のために 9 ビット，Ahead Information に関して 1 ビットの，合計 13 ビット必要であり，4 ホップの場合は 3 ホップ目のための 27 ビットを足して，40 ビット必要であり，範囲を広げるに従い必要なコストは大幅に増加するためである．また，範囲を広げてもその混雑状況が遠方のノードに伝わるためにレイテンシがかかるため，遅れて届くこのような混雑情報が混雑回避の役に立つとは限らず，実際，5.1 節の評価で示すとおり，提案手法のほうが RCA（8 ホップ先の混雑状況まで使用）よりスループットが高くなる．

4.　オンチップルータへの実装

前章で説明した PRC 機構を一般的なオンチップルータ上に実装した．図 4 に PRC のアーキテクチャを示す．破線で囲った部分が本研究で追加したハードウェアである．

4.1　追加したハードウェア

4.1.1　Route Predictor

入力物理チャネルごとに予測器が用意されており，それぞれのルーティング方向を予測する．各予測器は各入力物理チャネルごとのルーティング方向を監視し，3.5 節で述

© 2012 Information Processing Society of Japan

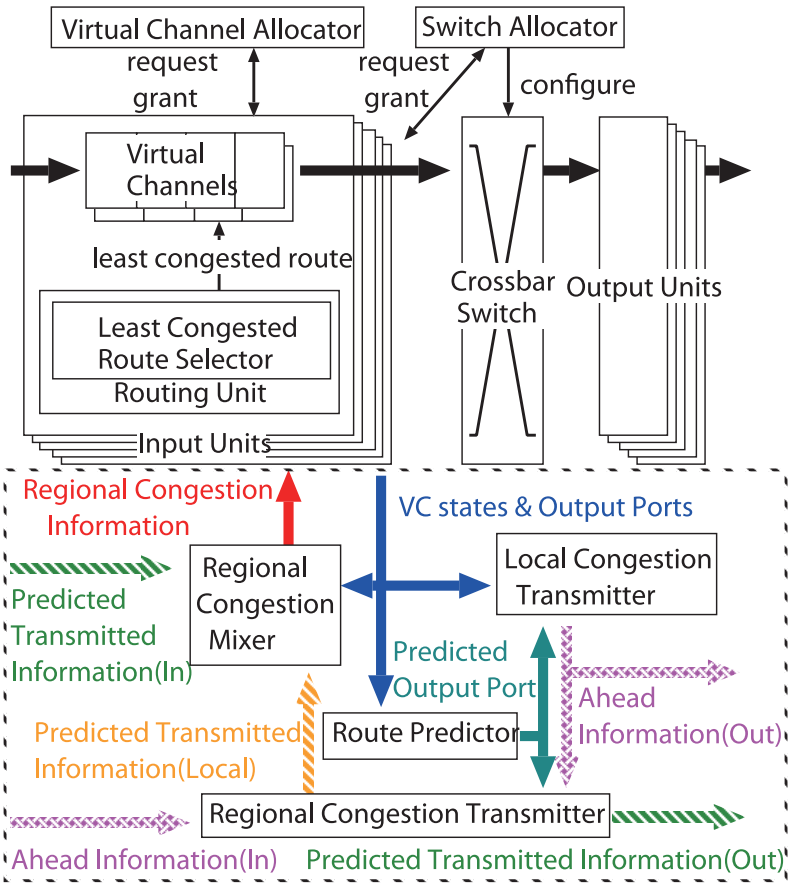


図 4　PRC のアーキテクチャ
Fig. 4　PRC architecture.

べたとおりのルーティング方向の予測を行う．各予測器は 2 つの 3 ビットレジスタを持っており，1 つは前回のルーティングの方向を保持し，もう 1 つは現在の予測方向を保持する．前者のレジスタと今回のルーティングの方向が同じ方向であれば，後者のレジスタが更新される．

4.1.2　Local Congestion Transmitter

このハードウェアは各入力仮想チャネルの状態とそれらがパケットを出力するポートを監視し，Ahead Information(0) を生成する．このために，Local Congestion Transmitter は東西南北それぞれの未来の混雑状況を示す 4 ビットのベクトル情報を保持する．Local Congestion Transmitter はパケットを保持している全仮想チャネルを調べ，それらがパケットを出力するポートに対応するビットを 1 にする．ただし，RC ステージの入力仮想チャネルについては Route Predictor の予測情報を用いて，対応するビットを 1 にする．この 4 ビットの情報は，それぞれ 1 ビットずつ Ahead Information(0) として対応する隣接ルータに送信されると同時に，ローカルの Regional Congestion Transmitter にも送信される．

4.1.3　Regional Congestion Transmitter

このハードウェアは隣接ルータの Local Congestion Transmitter から受信した Ahead Information(In) とローカルの Ahead Information(Out) を合成して，Predicted Transmitted Information(Out)，(Local) を生成する．

Predicted Transmitted Information は隣接ルータと，自ルータ両方で使用するため，両方に送信される．どちらも同じ情報であるが，図 4 上では，隣接ルータに送信するものを Predicted Transmitted Information(Out)，自ノードで使用するものを Predicted Transmitted Information(Local) と表記している．

Predicted Transmitted Information の生成のためには，まず，Ahead Information(In) を用いて未来の混雑情報を 4 ビットのベクトルとして生成する．Ahead Information(In) の各入力チャネルに対応するビットが 1 であれば，Route Predictor の出力を用いてその出力ポートを予測し，対応するビットを 1 にする．この未来の混雑情報 4 ビットと，ローカルの東西南北それぞれ 1 ビットの Ahead Information(Out) を合わせた 4 ビットとの論理和をとることで，Predicted Transmitted Information(Out)，(Local) を生成する．

4.1.4　Regional Congestion Mixer

このハードウェアは各ルータが得られる 3 種類の情報，すなわち，ローカルの情報，Ahead Information(In)，Predicted Transmitted Information(In)，それぞれについて，

© 2012 Information Processing Society of Japan

4 出力ポート × 3 経路の計 12 経路の混雑度をそれぞれ数値化する．3 種類それぞれの数値を合計することで，その経路の最終的な混雑度を求める．各情報の数値化は以下のように行う．

Ahead Information(In)　Ahead Information はパケットフローの上流から下流への，これからパケットを送信するという情報であり，隣接ルータの Local Congestion Transmitter から受信する．各入力チャネルごとに予測を用いて出力ポートを判断する必要があるが，その判断はローカルの Regional Congestion Transmitter がすでに行っており，Predicted Transmitted Information(Local) として出力されている．そのため，Regional Congestion Mixer はこれを受信するだけでよい．各出力ポートの混雑度を 1 ビットで得ることができるため，その 0 または 1 の値をそのまま，その出力ポートを使用する経路の混雑度とする．

Predicted Transmitted Information(In)　隣接ルータの Regional Congestion Transmitter からの入力チャネルあたり 3 ビットの情報であり，4 出力ポート × 3 経路の計 12 経路の混雑情報である．パケットフローの下流から上流への情報であり，12 経路それぞれに対応するビットの 0 または 1 をそのまま，その経路の混雑度とする．

Local Information　ローカルに持っている情報であり，各出力ポートを通る 3 経路の混雑度をそれぞれ使用中の仮想チャネル数とする．

上記の混雑度を各経路ごとに，それぞれ合計したものを混雑度として出力する．Routing Unit 内の Least Congested Route Selector はこの各経路ごとの混雑度を受け取り，使用可能な経路のうち最も混雑度の低い経路への出力ポートを選択する．

4.2　仮想チャネルの使用方法と各チャネルごとの配線数

本研究の実装ではスループットを最大化するために，すべての仮想チャネルを均等に使用した．この場合，仮想チャネル間で通信パターンに差が出にくいため，先に示した Ahead Information および，Predicted Transmitted Information は，仮想チャネルごとではなく，物理チャネルごとに用意した．そのため，本研究の実装では物理チャネルあたり Ahead Information の 1 ビットと Predicted Transmitted Information の 3 ビットの合計 4 ビットの配線が必要となる．

一方で，仮想チャネルごとに違うタイプのメッセージを流す用途では仮想チャネルごとに通信パターンに違いが生じる．このような場合において，仮想チャネルごとに混雑情報を生成・伝播させようとすると，仮想チャネル数の分だけリンクの追加配線数も増加する．

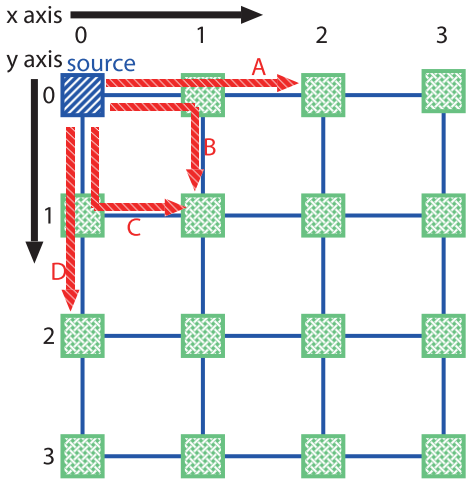


図 5　2 ホップ先までの経路
Fig. 5　Routes within two hops away.

4.3　ハードウェアコストの削減

PRC は 2 ホップ先までの未来の混雑情報を考慮してルーティングを行う．ある送信元ノードから宛先ノードの最短経路において，2 ホップ先の中継点までは 4 種類の最短経路がある．たとえば，送信元ノードから見て宛先ノードが (X+,Y+) 方向にある場合，**図 5** に示すように，X+方向に転送してもう 1 度 X+方向に転送する場合（X+→X+と表記（A）），X+→Y+（B），Y+→X+（C），Y+→Y+（D）の 4 種類がある．しかし，たとえば，宛先が $R(1,1)$ であった場合は，A と D は使用不可能である．最大 4 経路のうち，どの経路が使用可能なのかを判断し，さらにその中から最も混雑度合いの低い経路を選択するためには，大きなハードウェアコストがかかる．

そこで本論文では，4 種類の経路の混雑情報のうち，ターンを含む 2 種類の経路の混雑情報のみを利用する．つまり，図 5 においては，B と C のみの混雑情報を使用する．このことにより，宛先が同じ軸上にないとき，比較すべき経路の数はつねに 2 経路となり，ハードウェアを簡素化できる．さらに Predicted Transmitted Information に関して，直進方向の情報を伝播する必要がなくなるため，実際に送信するビット数を 3 ビットから 2 ビットに削減できる．

図 5 の B と C のようなターンを含む経路の中から混雑度合いの低い方を選んでいく実装は，2D-Mesh で最短適応型ルーティングを行う場合は，性能を低下させることなく，むしろ向上させる．これは文献 [11] で示されているようにパケットが宛先と同じ軸上に届くのを遅らせる方が，より多くの代替経路を利用できる可能性が高く，混雑を回避しやすくなるためである．たとえば，**図 6** では，A の経路をとったパケットはすでに 1 方向しかルーティングを行うことができず，混雑を回避することができないが，B の経路をとったパケットは 2 方向のルーティングを行うことができ，混雑を回避することができる．

図 7 に直進方向の情報を利用する場合と利用しない場

© 2012 Information Processing Society of Japan

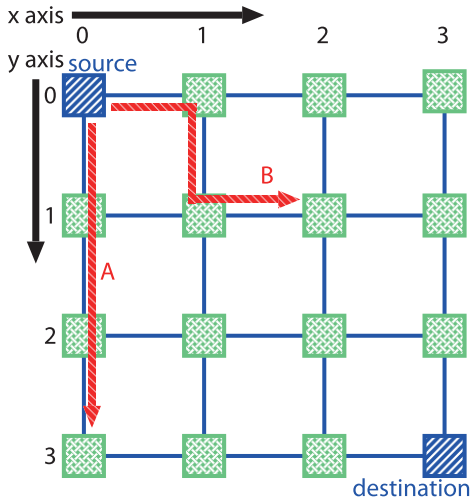


図 6　代替経路の可能性
Fig. 6　Possibility of alternative routes.

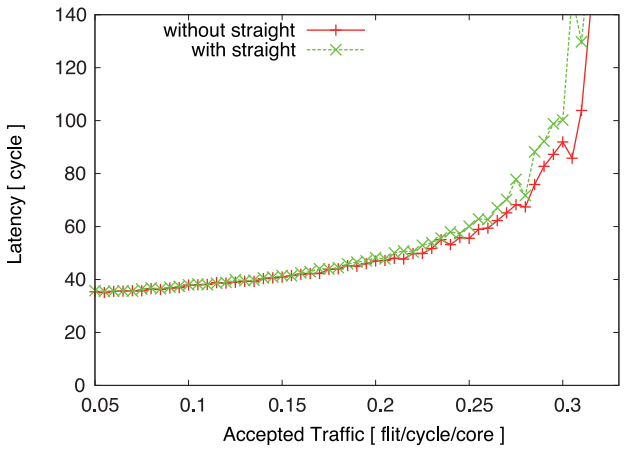


図 7　直進方向の情報
Fig. 7　Information of straight.

合のレイテンシを示す．評価環境は，仮想チャネル数は 2，バッファサイズは 4 フリット，ネットワークサイズは 4×4，トラフィックパターンは Uniform Random，バースト性は平均 4 パケット連続とした．直進方向の情報を無視し，ターンを含む経路を多く通る実装とした方が低レイテンシとなっていることが分かる．

また，表 4 に示すように，直進方向の情報を利用する場合の実装面積は，利用しない場合の面積に比べて 2.2%多い．よって，本研究では，ターンを含む経路の混雑情報のみを利用する実装とした．

5.　評価および考察

5.1　スループット性能

Cadence 社の NC-Verilog を用いて，Verilog HDL で実装した PRC ルータと既存のルータのシミュレーションによる評価を行った．比較を行ったのは以下の 4 種類の手法である．

DOR　次元順ルーティング（固定ルーティング）

表 2　評価環境
Table 2　Evaluation environment.

トポロジ	2D-Mesh
ネットワークサイズ	4×4，8×8
仮想チャネル数	2
バッファサイズ	4 フリット
フリットサイズ	128 ビット
パケットサイズ	5 フリット
バースト性	平均 4 パケット連続，なし
トラフィックパターン	Transpose [5], Uniform Random [5], Bit Complement [5]

Local　空き VC 数最大出力ポート選択機構（ローカルの情報のみ使用）[10]

PRC　Predictor with Regional Congestion（中距離の情報の使用）

RCA　Regional Congestion Awareness（グローバルな情報の使用）[13]

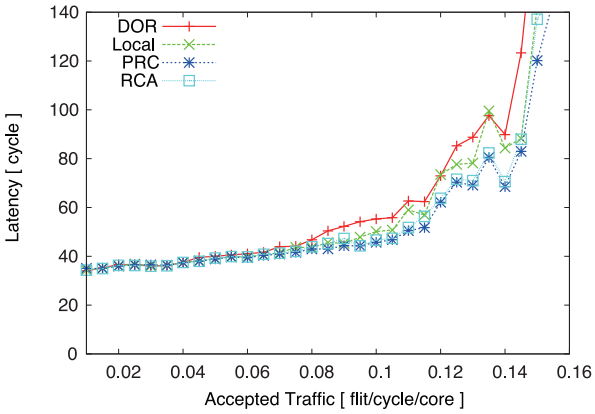
投機的スイッチアロケーションを行う 3 段パイプラインルータに上記の 4 種類の手法を実装し，シミュレーションを行った．ただし，RCA は次ホップの進行方向によって仮想チャネルをグループ化して使い分ける特殊なルータアーキテクチャをベースラインとしている．そのため RCA の評価には，通常の投機ルータに RCA の混雑情報を伝播させる機構を実装したものを用いた．評価に用いたパラメータは**表 2** に示すとおりである．シミュレーションサイクルは 100,000 サイクルとし，そのうち初めの 1 割の期間をウォームアップ期間，終わりの 1 割の期間をドレインの期間として無視し，それらの期間を除いた 80,000 サイクルを評価に用いた．ヘッダフリットがネットワークの入り口のキューに入ってから，テイルフリットが宛先ノードに到着するまでの時間を通信レイテンシとした．バーストトラフィックの生成には，Markov-modulated process（MMP）[*1] を用いた方法 [5] を使用した．また，本研究のシミュレーションでは，スループットを最大化するため，仮想チャネルを均等にランダムに使用している．

5.1.1　4×4 ネットワークにおけるバーストトラフィック

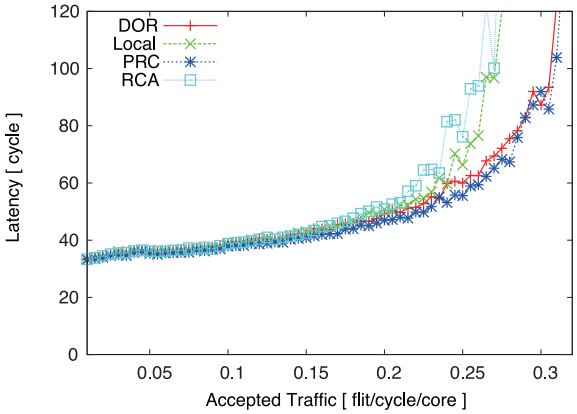
図 8 (b)，図 8 (a)，図 8 (c) に表 2 のとおりのパラメータを用いたシミュレーションの結果を示す．Transpose は負荷の偏ったトラフィックであり，固定型ルーティングである DOR は負荷を分散させることができず，最も高レイテンシとなっている．一方，適応型ルーティングは比較的低レイテンシとなっており，本論文で提案した PRC ルータは最も低レイテンシであった．これは，混雑情報のパケッ

*1　MMP による手法では，まず，パケットを送信する状態（ON 状態）と送信しない状態（OFF 状態）を定義し，OFF 状態から ON 状態への遷移確率を α，ON 状態から OFF 状態への遷移確率を β とする．本評価では，連続パケット数の平均が 4 パケットになるように α と β を与えている．

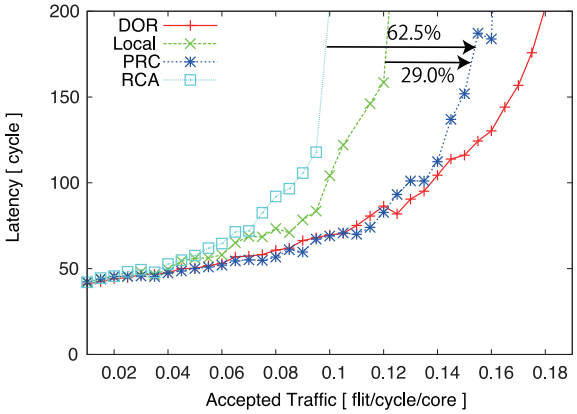
© 2012 Information Processing Society of Japan



(a) Transpose

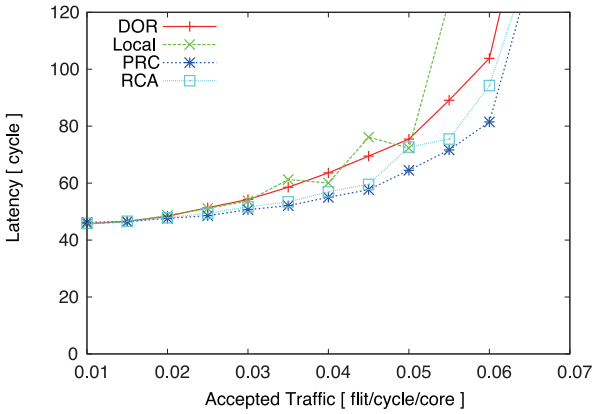


(b) Uniform Random

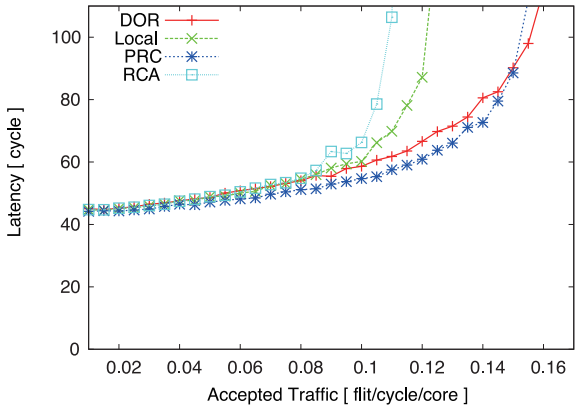


(c) Bit Complement

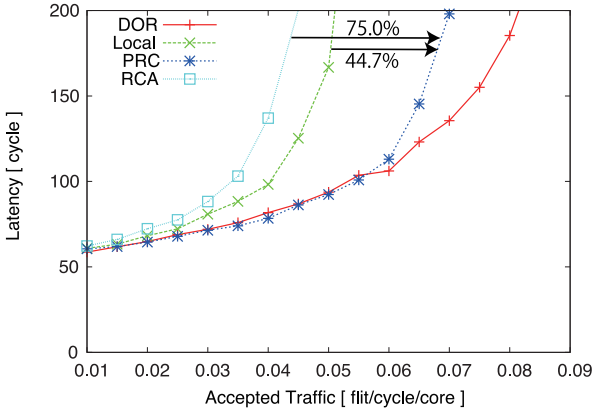
図 8　スループットとレイテンシ（4 × 4，バースト）
Fig. 8　Throughput and latency (4 × 4, burst).



(a) Transpose



(b) Uniform Random



(c) Bit Complement

図 9　スループットとレイテンシ（8 × 8，バースト）
Fig. 9　Throughput and latency (8 × 8, burst).

トフローの下流方向への伝播によるものである．

Uniform Random は初めから負荷が分散されているため，DOR にとって有利なトラフィックパターンであり，適応型ルーティングを用いてもこれ以上負荷分散できず，性能も向上しにくく，Local と RCA のスループットは DOR と比べて低下している．一方で PRC のスループットはあまり低下せず，DOR に匹敵するスループットを保っている．

Bit Complement は，DOR を行うと完全に負荷が分散するトラフィックパターンであり，適応型ルーティングにとっては不利な結果となる．そのため，DOR のスループットが最も高く，Local と RCA の性能はかなり低下した．一方で，PRC のスループットはあまり低下せず，Local，RCA と比べてそれぞれ 29.0%，62.5%向上した．

なお，バッファサイズを 8 フリットに変更した場合の評価も行ったが，傾向に差は生じなかった．

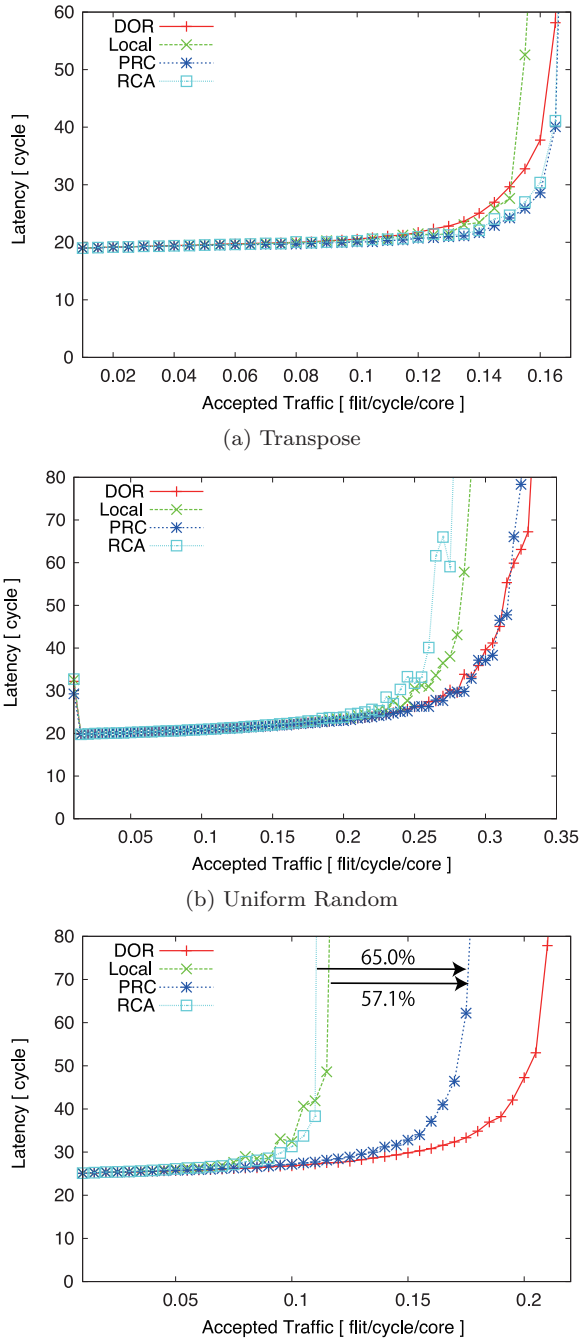
5.1.2　8 × 8 ネットワークにおけるバーストトラフィック

ネットワークサイズを 8×8 に変更した場合の評価を行った．**図 9** (b)，図 9 (a)，図 9 (c) にその結果を示す．ネットワークサイズを大きくしても，各手法の性能の傾向に大きな変化はなく，PRC が有用であることを示した．

© 2012 Information Processing Society of Japan

5.1.3　4×4 ネットワークにおける非バーストトラフィック

PRC が非バーストトラフィックでも性能を向上させることができること示すため，つねに一定の確率で一様にパケットを挿入した場合の評価を行った．**図 10** (b)，図 10 (a)，図 10 (c) にその結果を示す．本論文で提案した PRC は予測機構を使用しているため，非バーストトラフィックでは予測の的中率が低下し，性能が向上しにくい．しかし，非バーストトラフィックを与えた場合でも性能は向上し，既存の適応型ルーティング手法と比較するとすべてのトラフィックにおいて，レイテンシ，スループットが改善された．



(a) Transpose

(b) Uniform Random

(c) Bit Complement

図 10　スループットとレイテンシ（4 × 4，非バースト）
Fig. 10　Throughput and latency (4 × 4, non-burst).

5.2　ハードウェア量

5.2.1　論理合成の結果

Synopsys 社の Design Compiler を用いて TSMC 130 nm プロセスで論理合成を行った．その際，NoC 全体ではなく，単一の 5 ポートルータを切り出して合成した．なお，本研究で使用したアービタはマトリックスアービタ [5] である．**表 3** に最大動作周波数を，**表 4** に目標動作周波数を 333 MHz とした場合の実装面積を示す．括弧内は DOR との差を示したものである．最大動作周波数の低下は 14.8%に抑えられ，実装面積の増加も DOR と比べてどちらも約 5.1%に抑えられた．これらは各種情報の生成および合成に用いる演算が単純であること，2 つの予測情報の生成に同一の Route Predictor を共有して使用したこと，Ahead Information と Transmitted Information を同一のハードウェアで生成したことによる．また，ルータ間に追加した配線は，Ahead Information のための 1 ビットと Predicted Transmitted Information のための 2 ビットの，チャネルあたり合計 3 ビットのみであり，配線のオーバヘッドは小さい．

5.2.2　RCA との比較

RCA のベースラインは特殊であるため，ルータ全体ではなく RCA，PRC それぞれの混雑情報を伝播させる部分のみについて，必要となるゲート数を比較した．

RCA のパラメータは，RCA Fanin [13] を想定し，ルータ間の追加のワイヤは 8 本とし，シフタには 8 ビットバレルシフタを用い，ローカルの混雑情報の生成には空き仮想チャネル数最大出力ポート選択機構 [9] を用いた．PRC のパラメータおよび RCA の他のパラメータは，表 2 のとおりとした．

論理合成結果より，RCA の混雑情報を伝播させる部分のハードウェアのゲート数は 3,111 個であったのに対し，PRC は 1,317 個であった．RCA は多数の 8 ビット加算器を必要とするが，PRC は少量の 1 ビット加算器と 2 ビッ

表 3　最大動作周波数
Table 3　Maximum frequency.

	最大動作周波数 [MHz]
DOR	492（± 0.0%）
Local	483（− 1.9%）
PRC without straight	446（− 9.4%）
PRC with straight	458（− 6.9%）

表 4　実装面積
Table 4　Implementation area.

	実装面積 [μm^2]	面積の平方根 [μm]
DOR	479,784（± 0.0%）	693（± 0.0%）
Local	480,363（+ 0.1%）	693（+ 0.0%）
PRC without straight	504,070（+ 5.1%）	710（+ 2.5%）
PRC with straight	515,407（+ 7.4%）	718（+ 3.6%）

© 2012 Information Processing Society of Japan

ト加算器があればよい．RCA，PRC の加算器に使用されているゲート数の合計はそれぞれ 1,495 個，120 個である．また，RCA は多数のシフタも必要であり，そのゲート数の合計が 1,296 個である．これらの理由により RCA のゲート数は増加している．また，RCA，PRC のルータ間に追加で必要なワイヤの数はそれぞれ 8 本，3 本となっており，PRC の配線のオーバヘッドは小さい．

5.3　他のグローバルな情報を用いる手法との比較

本節では，他のグローバルな情報を用いる混雑回避手法 [14], [15], [16] と提案手法との違いについて議論する．

文献 [16] では，複数のルータで 1 つのクラスタを構成し，隣接クラスタ間で混雑情報を交換する．各ルータではこのようにして得た non-local な混雑情報と local な混雑情報を用いて混雑しない経路を選択する．この手法では，agent network と呼ばれる混雑情報の交換を行うための専用ネットワークに加え，cluster agent と呼ばれる専用コントローラをクラスタごとに 1 つ設ける必要がある．一方，提案手法はリンクあたりの配線オーバヘッドは 3 本，ルータの面積オーバヘッドは 1.3%と非常に低コストで実現できる．

文献 [15] では，高次元なオフチップネットワーク向けに，近傍の混雑情報を用いて適応型ルーティングを行う手法をいくつか提案している．その中でも Piggyback ルーティングでは，同一グループ内のルータと混雑情報を交換し，適応型ルーティングの経路選択に利用する．混雑情報の交換には，リンクの状態をパケットにピギーバックする方法，もしくは，利用していないチャネルを用いてブロードキャストする方法が紹介されている．一方，提案手法は NoC 向けという特徴を活かし，リンクあたりたかだか 3 本の専用線を用いることで混雑情報を低レイテンシで伝播させることができる．

文献 [14] では token と呼ばれる各ルータの入力バッファの利用率を示すメッセージを近傍ルータと交換する．このように，これらの関連研究 [13], [14], [15], [16] は現在のネットワークの状態を近傍ルータへ伝播させることで混雑情報の共有，混雑回避を行う．一方，我々の提案手法では，現在の情報に加え，予測に基づいた Ahead Information を用いることで実際に混雑が生じる前に混雑を検出し，混雑を事前に回避することができる．

6.　まとめ

本研究では，ハードウェアコストを抑えながら NoC 上でのロードバランシングおよびパケットの衝突を回避するために，周辺の混雑を予測するオンチップ適応型ルータ（On-chip Adaptive Router with Predictor for Regional Congestion：PRC）を設計および実装した．PRC は混雑状況を隣接ルータに伝え，その情報を用いて混雑を回避したルーティングを行う．さらに各入力チャネルごとに出力ポートの予測を行い，この情報をパケットフローの上流だけでなく下流にも送信することでパケット衝突の回避を行う．

各種パラメータを変更しながらシミュレーションによる評価を行った結果，PRC ルータは適応型ルーティングを行うと性能の上がりやすい Transpose において，すべてのパラメータで最も低レイテンシとなった．その一方で，適応型ルーティングと相性の悪い Uniform Random および Bit Complement においても，すべてのパラメータで性能をあまり低下させることはなく，適応型ルーティングの中で最も良い性能となった．

また，オーバヘッドについては，面積の増加は 5.1%，動作周波数の低下は 9.4%，ルータ間の配線の追加は物理チャネルあたり 3 本のみであった．

よって，PRC は実装面積を低く抑える必要がある場合に，混雑を回避しスループットおよびレイテンシを向上させるのに有用であるといえる．

謝辞　本研究は科学技術振興機構 CREST の支援によるものであることを記し，謝意を表す．また，本研究の一部は文部科学省グローバル COE プログラム「環境共生・安全システムデザインの先導拠点」によるものであることを記し，謝意を表す．

参考文献

[1] Ainsworth, T.W. and Pinkston, T.M.: Characterizing the Cell EIB On-Chip Network, *IEEE Micro*, Vol.27, No.5, pp.6–14 (2007).

[2] Wentzlaff, D., Griffin, P., Hoffmann, H., Bao, L., Edwards, B., Ramey, C., Mattina, M., Miao, C.-C., Brown III, J.F. and Agarwal, A.: On-Chip Interconnection Architecture of the Tile Processor, *IEEE Micro*, Vol.27, No.5, pp.15–31 (2007).

[3] Howard, J. et al.: A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS, *Proc. International Solid-State Circuits Conference* (*ISSCC'10*), pp.108–109 (2010).

[4] Dally, W.J. and Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks, *Proc. Design Automation Conference* (*DAC'01*), pp.684–689 (2001).

[5] Dally, W.J. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).

[6] Glass, C.J. and Ni, L.M.: The Turn Model for Adaptive Routing, *Proc. International Symposium on Computer Architecture*, pp.278–287 (1992).

[7] Chiu, G.-M.: The Odd-Even Turn Model for Adaptive Routing, *IEEE Trans. Parallel and Distributed Systems*, Vol.11, No.7, pp.729–738 (2000).

[8] Duato, J.: A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, *IEEE Trans. Parallel and Distributed Systems*, Vol.4, No.12, pp.1320–1331 (1993).

[9] Dally, W.J. and Aoki, H.: Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, *IEEE Trans. Prallel and Distributed Systems*, Vol.4, No.4, pp.466–475 (1993).

[10] Kim, J., Park, D., Theocharides, T., Vijaykrishnan, N.

© 2012 Information Processing Society of Japan

and Das, C.R.: A Low Latency Router Supporting Adaptivity for On-Chip Interconnects, *International Conference on Design Automation*, pp.559–564 (2005).
[11] Badr, H.G. and Podar, S.: 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).
[12] Martinez, J.C., Silla, F., Lopez, P. and Duato, J.: On the Influence of the Selection Function on the Performance of Networks of Workstations, *Proc. International Symposium on High Performance Computing*, pp.292–300 (2000).
[13] Gratz, P., Grot, B. and Keckler, S.W.: Regional Congestion Awareness for Load Balance in Networks-on-Chip, *Proc. International Symposium on High-Performance Computer Architecture* (*HPCA'08*), pp.203–214 (2008).
[14] Kumar, A., Peh, L.-S. and Jha, N.K.: Token Flow Control, *Proc. International Symposium on Microarchitecture* (*MICRO'08*), pp.342–353 (2008).
[15] Jiang, N., Kim, J. and Dally, W.J.: Indirect Adaptive Routing on Large Scale Interconnection Networks, *Proc. International Symposium on Computer Architecture* (*ISCA'09*), pp.220–231 (2009).
[16] Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J. and Tenhunen, H.: Efficient Congestion-Aware Selection Method for On-Chip Networks, *Proc. International Workshop on Reconfigurable Communication-centric Systems-on-Chip* (*ReCoSoC'11*), pp.1–4 (2011).
[17] Peh, L.-S. and Dally, W.J.: A Delay Model and Speculative Architecture for Pipelined Routers, *Proc. International Symposium on High-Performance Computer Architecture* (*HPCA'01*), pp.255–266 (2001).

谷口 将一　（学生会員）

2011 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院修士課程に在籍．オンチップネットワークの研究に従事．

松谷 宏紀　（正会員）

2004 年慶應義塾大学環境情報学部卒業．2008 年同大学大学院理工学研究科開放環境科学専攻博士課程修了．博士（工学）．現在，慶應義塾大学理工学部情報工学科専任講師．2009 年度より 2010 年度まで日本学術振興会特別研究員 SPD．計算機アーキテクチャ，オンチップネットワークの研究に従事．

山崎 信行　（正会員）

1991 年慶應義塾大学理工学部物理学科卒業．1996 年同大学大学院理工学研究科計算機科学専攻博士課程修了．博士（工学）．同年電子技術総合研究所入所．1998 年 10 月慶應義塾大学理工学部情報工学科助手．同専任講師を経て，2004 年 4 月より同助教授（現，准教授）．リアルタイムシステム，プロセッサアーキテクチャ，並列分散処理，システム LSI，ロボティクス等の研究に従事．日本ロボット学会，IEEE 各会員．

© 2012 Information Processing Society of Japan