Vol.2009-ARC-184 No.31
2009/8/6

予測ルータによる低遅延 Fat Tree ネットワーク

館下智明[†1] 秋岡明香[†1] 吉永努[†1]
松谷宏紀[†2] 鯉渕道紘[†3]

並列計算機においてはノード間の通信遅延が処理能力に影響を与えるため，低遅延なネットワークが求められている．そこで，本論文ではルーティングに予測を用いることにより，Fat Tree ネットワークの通信遅延を削減する試みについて報告する．予測方法として，Tree の上（ルート）方向に向かうメッセージのみ予測した場合と，上下方向のメッセージを予測する場合について考察する．また，システムエリア/オンチップ・ネットワークへの応用についても考察する．シミュレータを用いた実験の結果，Fat Tree トポロジにおいて予測ルータを使用した場合，従来に比べて約 20 %ほど遅延を小さくできることがわかった．

Prediction router for low latency Fat Tree network

Tomoaki Tateshita,[†1] Sayaka Akioka,[†1]
Tsutomu Yoshinaga,[†1] Hiroki Matsutani[†2]
and Michihiro Koibuchi[†3]

Low latency networks are required in a parallel computer because the delay of communication between the nodes influences processing performance. In this paper, we report on the method of reducing the delay of the communication on the Fat Tree network when the prediction is used for routing. The paper considers the prediction procedure when only messages toward the upside direction direction of upside of Tree (root) are predicted and when the messages in the vertical direction are predicted. Moreover, it is applicable for the system area/on chip network. Our results showed that it is able to reduce the delay about 20% compared with the past when the prediction router is used in the Fat Tree topology as a result of the experiment with the simulator.

1. はじめに

近年，並列計算機はその高性能化に伴い，コア数，ネットワーク規模の増大が進んで来ており，コア間の通信遅延が並列計算機の処理能力に与える影響が大きくなってきている．コア間の通信には Interconnection Network が広く用いられるため，通信遅延の小さいルータの開発が望まれている．

これまでの研究によって，予測機構を持つ低遅延ルータを用いることによりオンチップネットワーク（OCN）において遅延を削減できることがわかっており[1)]，予測のヒット率は予測アルゴリズムやネットワーク構造に依存することがわかっている[2)]．本研究では，PC クラスタなどに用いられる Fat Tree ネットワーク[3)] に対して予測ルータを導入し，通信遅延を削減する試みについて報告する．オンチップネットワークとシステムエリアネットワーク（SAN）の 2 種類の実装を想定してシミュレーションを設定し，それぞれについて通常のルータを使用した場合と予測ルータを使用した場合の遅延を比較する．システムエリアネットワークではフリットのサイズを変えた場合の遅延の比較も行う．またオンチップネットワークを想定したシミュレーションでは，ノード数を変えて 2 種類の予測アルゴリズムの予測のヒット率の比較も行う．

本論文では，2 章で予測ルータについて説明し，3 章でシリアル通信について説明する．4 章でシミュレータでの評価結果を示し，5 章において関連研究をまとめた後，6 章でまとめとする．

2. ルータの構造

2.1 予測ルータ

図 1 に予測ルータの構造を示す．通常のルータでは入力バッファにメッセージが届いてからアービトレーションを行うが，予測ルータではバッファにフリットが無いとき，予測器を用いて次に来るフリットが使うと予測される出力ポートへアービトレーションを行う．

通常のルータでは，入力バッファにフリットが到着すると図 2 の normal に示すように，

†1 電気通信大学
The University of Electro-Comunicasions
†2 東京大学
The University of Tokyo
†3 国立情報学研究所
National Institute of Informatics

Vol.2009-ARC-184 No.31
2009/8/6

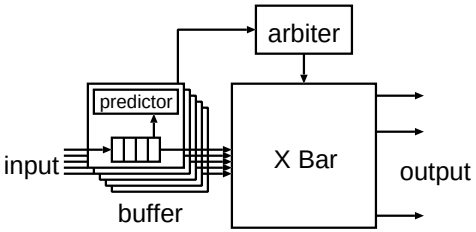

図 1 予測ルータの構造
Fig. 1 Architecture of prediction router

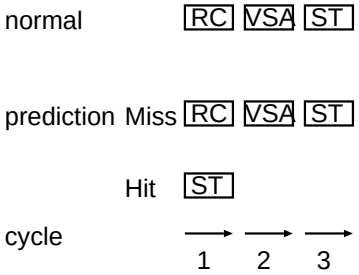

図 2 通常のルータと予測ルータのパイプライン構造
Fig. 2 Pipelines for a normal router and a prediction router

3 段のパイプラインを経て出力する．まず，宛先アドレスから出力ポートを計算する RC（Routing computation）を行い，次に RC の結果をもとに出力仮想チャネルの割り当てとクロスバスイッチの設定を行う VSA（Virtual channel/Switch allocation）を実行する．そのうえで，クロスバスイッチを通過する ST（Switch traversal）を経て出力ポートからフリットを出力する．

予測ルータでは，予測が正しい場合，入力したメッセージは RC と VSA を省略して ST を実行することができるため（図 2 の prediction Hit），予測成功率に応じて通信遅延を削減することができる．また，予測がはずれた場合は予測ルータも通常のルータと同様に，RC，VSA，ST の 3 段のパイプラインを経て転送を行う（図 2 の prediction Miss）．ただし，予測失敗時に間違った出力ポートへフリットが送られてしまう場合がある．その対策として，オンチップネットワークでは ST と平行して RC を実行することによって予測失敗を検出し，予測ミスしたメッセージが出力ポートから出力されることを防止する[4]．

2.2 予測アルゴリズム

予測ルータは予測の成功率によって通信遅延が左右されるので，予測成功率の高い予測アルゴリズムが求められる．ここでは本論文で使用する予測アルゴリズムについて述べる．

Static Straight（SS）ではフリットは同一次元上を直進すると予測する．Fat Tree においては下の階層から来たフリットは上の階層へ出力されると予測し，上の階層から来たパケットは下の階層を目指すと予測を行う．図 3 の middle のように，本論文で検討する Fat Tree は上向き，下向きのポート数が等しいので，それぞれのポートが 1 対 1 に対応するポートを予測出力ポートとする．また，図 3 の top のように，例外的に最上位の階層では折り返すように予測を行う．

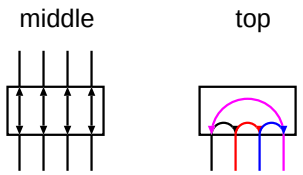

図 3 SS の Fat Tree における予測の仕方
Fig. 3 Method of prediction in Fat Tree of SS

Up Priority（UP）は up*/down* ルーティングにおいて，上の階層を目指すと予測し，下の階層を目指す予測は行わない．Fat Tree においては SS は UP を含んでいるといえる．

3. シリアル通信

3.1 Ser/Des 変換

システムエリアネットワークではシリアル通信を想定する．ルータ間のデータ送受信をシリアル通信で行う場合，通信データのシリアライズとデシリアライズ変換を行う必要がある．それぞれの変換処理にはある程度の時間がかかるが，予測ルータで予測が成功した場合にシリアルデータのままルータを通過できるようにすれば，予測スイッチングを用いることで，メッセージをルータに入力するときのシリアライズ遅延を通信時間から削減することができる．

3.2 ヒントビット

2.1 節において，ST を実行している間に RC を並列に実行することによって予測ミスを検出することを述べた．本論文ではオンチップネットワークはパラレル転送を想定しているため，入力ポートにおいて各フリットを完全にバッファリングした後，RC と ST を実行する．しかし，3.1 節で述べたようにメッセージがシリアルデータをそのままルータを通過する場合には RC と並列実行する予測ミス検出がルータからメッセージ出力を開始するまでに間に合わなくなる．

そこで，ヒントビットを用いて予測ミスメッセージの出力を抑制する方法を説明する．ヒントビットは，メッセージの送信元ノードから宛先ノードへ進む方向を示し，メッセージの先頭に付け加える（図 4）．例えば，Fat Tree トポロジで up*/down* ルーティングを用いる場合，送信元ノードと宛先ノードがわかれば最短経路は Tree をどこまで up 方向に上がればいいかがわかる（Least common ancestor routing）．2 階層のスイッチで構成する Fat

Vol.2009-ARC-184 No.31
2009/8/6

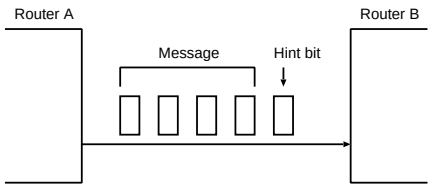

図 4 ヒントビット
Fig. 4 Hint bit

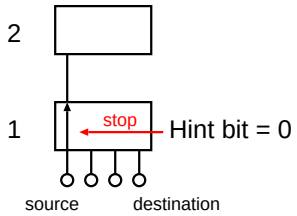

図 5 ヒントビットによる制御
Fig. 5 Control by Hint bit

Tree の場合を例にとると，1 段上がるだけでいいときはヒントビットを 0 とし，2 段上がる必要があるときはヒントビットを 1 とする．このとき階層 1 のルータで上向きのポートを予測しているのにヒントビットが 0 の信号が入力された場合は予測が外れていると判断することができる（図 5）．ヒントビットを用いて予測ミスを検出する機構を用いると，RC と ST を並列に実行するよりも，予測ミスパケットの kill 機構が簡略化されるなど，ルータの構造を単純にできる可能性があるが，本論文では議論が発散するため扱わないこととする．また本論文のシミュレーションでは，ヒントビットはシステムエリアネットワークでのみ用い，オンチップネットワークでは用いない．

4. 評価結果

フリットレベルのシミュレータである booksim シミュレータ[5] を用いてシミュレーションを行い，予測ルータによる遅延削減の効果を調べる．なお，オンチップネットワークとシステムエリアネットワークの 2 種類のネットワークを想定してシミュレーション条件を設定した．LAN のシミュレーションでは Quadrics 社の QsNet^{III}[6] を条件設定のモデルに用いたシミュレーション条件を設定した．メッセージの通信パターンには uniform random を用いた．uniform パターンはメッセージごとに宛先ノードをランダムに選択する．

4.1 オンチップネットワーク

オンチップネットワークを想定したシミュレーションでは，Fat Tree トポロジにおいて通常のルータと予測ルータを比較する．

Fat Tree の構成を示す方法として，ここでは（p，q，r）Fat Tree と表す．p は 1 つのスイッチが有する上向きのリンク数，q は下向きのリンク数で，r は階層数を表す．よって，（4，4，3）Fat Tree は上向きに 4 本，下向きに 4 本の計 8 本のリンクを持つルータからなる 3 階層の Fat Tree を表す（図 6）．

表 1 OCN のシミュレーション条件
Table 1 Simulation parameter of OCN

	Case 1
Topology	(4,4,3) Fat Tree
Traffic	uniform
Routing	up*/down*
Switching	Wormhole
Channel	2VC
Pipeline	[RC][VSA][ST]
flit/packet	5 flit
Input buffer	4 flit FIFO

表 2 SAN のシミュレーション条件
Table 2 Simulation parameter of SAN

	Case 2
Topology	(16, 16, 2) Fat Tree
Traffic	uniform
Routing	up*/down*
Switching	Store and forward or Virtual cut through
Channell	4VC
Pipeline	[Des][RC][VSA][ST][Ser][LT]
Packet size	256 byte
Input buffer	256 byte

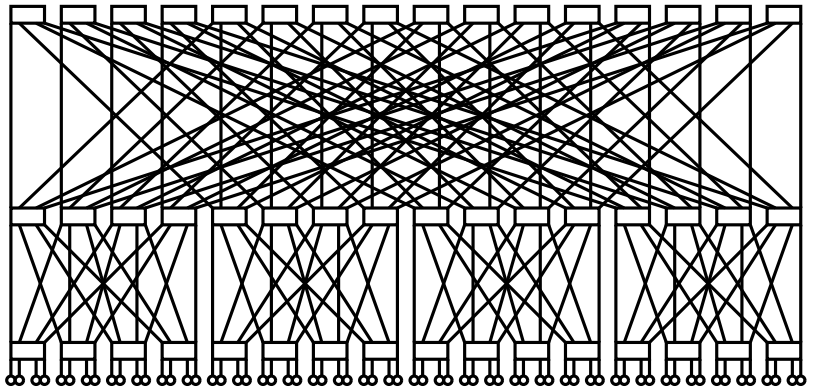

図 6 （4，4，3）Fat Tree
Fig. 6 （4，4，3）Fat Tree

4.1.1 ルータの構成

ここで予測ルータと通常のルータの構成について説明する．まず共通点として，ルータはワームホールスイッチングを使用し，入力ポートごとに 2 本の仮想チャネルを持つ．それぞれの仮想チャネルは 4-flit 分のバッファを持つ．また，パケットは 5-flit と仮定し，ヘッダがブロックされると複数のルータにまたがって停止する．

4.1.2 遅延の評価

図 7 に，シミュレーションによって得た平均遅延を示す．FatTree-normal は通常のルータを使用した場合の遅延のグラフで，FatTree-pred は予測ルータを用いた場合の遅延のグ

ラフである．予測アルゴリズムは SS を用いる．2.2. 節で説明したように，通信パターンによらずそれぞれのポートが 1 対 1 に対応するポートを予測するので，予測による出力ポートのアービトレーションでは衝突は起きない．予測が当たれば RC と VSA を省略して，ST のみの 1 サイクルで転送できる．

通常の Fat Tree と予測ルータの Fat Tree のグラフを比較すると，負荷が小さいときでも 20 %ほど，予測ルータを用いた Fat Tree の方が遅延が小さいことがわかる．これは，ルータ当たりのホップ遅延が必要なパイプライン段数の削減によって小さくなること，またメッセージ当たりの遅延低下によってネットワークの飽和スループットが向上することによる．

4.1.3 ヒット率の比較

図 8 にノード数を変えた場合の SS と UP の平均ヒット率を示す．ネットワークサイズの変更に当たっては，ルータのポート数はそのままに，Fat Tree の階層数を変化させた（4，4，r）Fat Tree を用いた．他のメッセージとの衝突を防ぐため，Injection rate は Zero load，すなわちネットワーク上にたったひとつしかパケットが存在しない無負荷の状態としている．

UP と SS ともにネットワークサイズが大きくなるにつれてヒット率が上昇し，SS は 60 %，UP は 45 %程度で飽和していることがわかる．SS は Tree の上下方向に進むメッセージに対して予測するため，予測を行う機会は UP の 2 倍以上あるが，Fat Tree では下向きポートの予測は上向きポートの予測に比べて当たりにくいためヒット率は 2 倍にはならない．また文献 2) では（1，4，r）Fat Tree および（2，4，r）Fat Tree について予測のヒット率の比較を行っており，(1，4，r）Fat Tree は約 40 %，(2，4，r）Fat Tree は約 35 %でほぼ一定の値となっている．文献 2) では上向きポートを異なる経路として扱っているため，上向きポートが増えると予測のヒット率が低下しているが，本論文では上の階層へ向かうメッセージはどの上向きポートでも予測成功としているのでヒット率が高くなり，より遅延を削減することができる．

4.1.4 パイプライン段数の比較

図 9 に予測ルータのパイプライン段数を変えた場合の平均遅延を示す．予測アルゴリズムは SS を使用している．それぞれ 3 段，5 段，7 段のパイプラインだが，予測が成功した場合は 1 段で転送ができるとする．

パイプライン段数を増やすとルータで複雑な処理を行ったり，動作周波数を上げることができるようになるが，図 9 のように平均遅延が大きくなる．しかし，負荷が小さいときはパイプラン段数が増えても遅延の増加が小さく，予測ルータによる遅延削減の効果はパイプライン段数が多いほど大きいことがわかる．

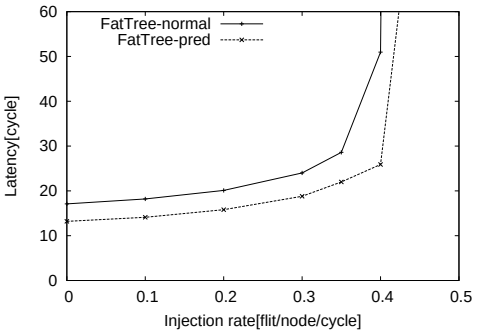

図 7 OCN における遅延
Fig. 7 delay in OCN

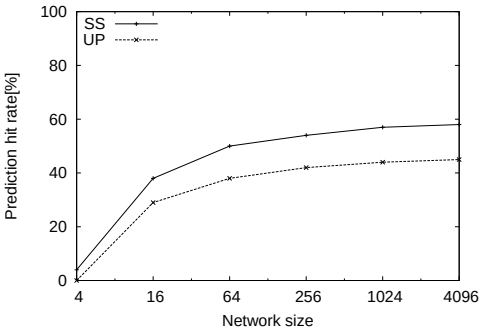

図 8 OCN におけるヒット率
Fig. 8 hitrate in OCN

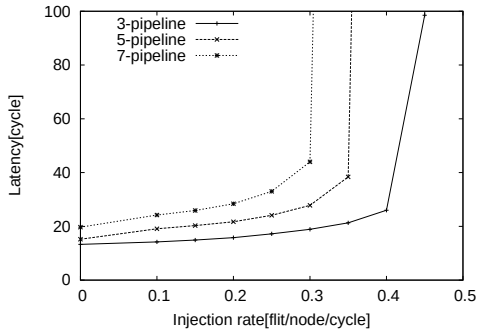

図 9 パイプラン段数を変えた遅延
Fig. 9 latency when pipeline is changed

以上より，通信負荷，ノード数，パイプライン段数の 3 つの条件を変えた場合に，予測ルータを用いることによって遅延を削減できていることを確認した．

4.2 システムエリアネットワーク

$QsNet^{III}$ をモデルとして，システムエリアネットワークを想定したシミュレーションでは，Fat Tree トポロジで比較を行う．表 2 にシミュレーション条件を示す．

4.2.1 ルータの構成

ルータ内の動作周波数を 312 MHz とし，リンクは 6.25 GHz で動作するチャネルが 4

チャネルずつ双方向にあると仮定する．よってリンクは 25 Gbit/sec となるが，リンクを通すときには 8b10b 変換を行うため実際のリンクスループットはやや小さくなり，ルータの 1 サイクル（3.2 nsec）当たり 8 byte の情報を転送することができる．パケットは 256 byte としているので，リンクでパケットを全て転送するためには 32 サイクルかかる．ルータはリンクごとに 4 本の仮想チャネルを持ち，それぞれの仮想チャネルはパケットサイズと同じ 256 byte のバッファを持っているとする．

予測アルゴリズムは UP を使用する．よって 2 階層 Fat Tree では予測を行うのは最初のルータの一回きりである．ヒントビットを用いることによって予測のヒット/ミスを判定して転送ミスをすることなく遅延を削減することができる．パイプラインは Des，RC，VSA，ST，Ser，LT の 6 段とする．Des はデシリアライズ，Ser はシリアライズ変換を表しており，LT は Link Traversal でリンクの移動時間を表している．

4.2.2 遅延の評価

図 10 に通信負荷を変化させたときの平均遅延を示す．スイッチング方式はストアアンドフォワードとし，パケットが全てバッファリングされてからルータ内の動作を行うとする．そのためリンクの移動時間である LT が 32 サイクル，ルータのパイプラン動作が 5 サイクルで，リンクの移動が遅延の大部分である．予測がヒットした場合はパケットが全て到着するのを待たずに転送を行うとする．シミュレーションより，負荷が 0.15（flit/node/cycle）を超えると予測のありなしに関わらず遅延が急激に大きくなっているが，それまでは予測スイッチングにより一貫して 20 %ほど遅延を小さくできることがわかる．

4.2.3 OCN との比較

4.1.2 節で評価した SS 予測を用いたオンチップネットワークのシミュレーションでも予測ルータは通常のルータに比べて 20 %ほど遅延を削減した．オンチップネットワークでは 3 階層の Fat Tree であり，SS 予測を用いているので最大で 5 回，予測の機会がある．それに比べて 2 階層の Fat Tree で UP 予測の場合は最初のルータでの 1 回しか予測の機会が存在しないにもかかわらず，同程度遅延を削減している．

以上の理由としては 2 つの原因が考えられる．まず，SAN ではデシリアライズとシリアライズを行っており，予測成功時にはその両方を省略するため，オンチップネットワークに比べて予測ルータの遅延削減効果が大きくなる．また，ストアアンドフォワードでフロー制御をしているため，予測ルータを用いない場合にはパケットが全て届くまで待つ必要がある．そのため，予測ルーティングを行った場合とそれ以外の差がさらに大きくなっている．

もうひとつの原因としては，予測の成功率のばらつきが考えれられる．Fat Tree では上の階層へパケットを送るときは複数あるポートのうちどれを使っても問題はないが，下の階層へ送るときは唯一のポートを選択することになる．そのため，予測の成功率で考えると下向きよりも上向きのほうが予測の成功率が高くなりやすい．よって，SS では最大で 5 回の予測の機会があっても下の階層への予測は成功率が低いため遅延の削減にはつながりにくくなっている．

4.2.4 フリットサイズ

4.2.2 節ではフリットをパケットサイズとしており，ストアアンドフォワードで転送をしていたが，ここでは同じサイズのパケットを複数のフリットに分けてカットスルーで制御した場合について評価する．

図 11 では横軸をフリットサイズとしており，フロー制御を行うフリットのサイズを表している．パケットのサイズは同じとしているので，フリットサイズが小さければ小さいほど，1 つのパケットをより多くのフリットに分割できる．またバッファリングはフリット単位で行い，デシリアライズなどはフリットがバッファに溜まってから行う．たとえば，フリットサイズが 16 byte のときは LT が 2 サイクル，32 byte のときは LT が 4 サイクルとなる．そのため，フリットサイズが小さいほど細かいフロー制御ができるので，より遅延を小さくすることができている．

予測ルータを使用した場合，フリットが 8 byte のときで 8.3 %遅延を削減している．フリットが 256 byte のときには 23.2 %，遅延を小さくできており，フリットサイズが大きくなるにつれて予測ルータによる遅延削減の割合が大きくなっている．これは予測が成功したときにはフリットが全て届くのを待たずに次のルータへ信号を送り出すためである．

5. 関 連 研 究

予測ルータの他にも，チップ内ネットワークを対象とした低遅延ルータに関して盛んに研究が行われている．

本論文では，3 段パイプライン構成のルータを基に予測機構を適用する議論を進めてきた．一方で，RC と SA をオーバラップさせることで，ルータのパイプラインステージを 1 ステージ分削減する投機ルータに関する様々な議論も行われている[5]．ただし，SA は RC 完了後に実行する必要があるため，RC と SA を同一サイクルで実行することは効率が悪い．

そこで，1 ホップ手前のルータに次ホップのルータの RC を予め実行する手法である Next routing computation，NRC 機構を併用する手法が議論されている．NRC の結果は次ホップのルータで使われ，自ルータの SA に影響を与えないため，NRC と SA を並列に実行で

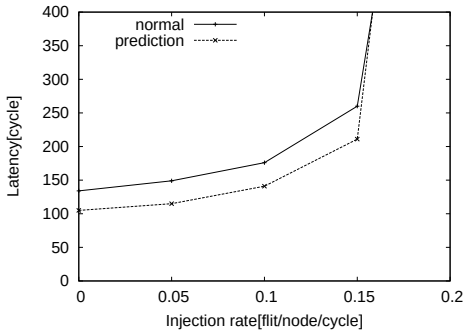

図 10 SAN における遅延
Fig. 10 delay in SAN

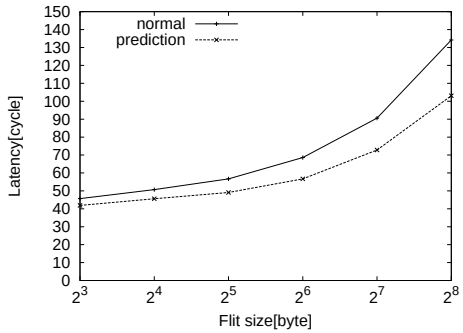

図 11 SAN におけるフリットサイズを変えたときの無負荷遅延
Fig. 11 zero load latency when the flit size is changed in SAN

きる．このように次ホップの経路を計算するルーティングをルックアヘッドルーティングと呼ぶ．

さらに，SA と ST をオーバラップさせることで 1 サイクル転送も可能となるが，1 サイクルに多くの処理を詰め込むことになるので，動作周波数の大幅な低下を招きやすい[5]．また，隣接ルータ間で連携することでパケット処理の遅延を抑える方法も提案されている．Express VC は，仮想的に非隣接ルータ間でバイパス経路を構成することにより中継ルータにおける所要パイプライン段数を削減する[7]．しかし，局所性を持つ通信パターンに対しては低遅延化の効果が小さい．

これらに比べて，予測ルータは動作周波数の低下が 6 %と小さく[4]，かつ，あらゆる入出力ポート対の通信に対しても遅延を削減できる点で極めて有効といえる．さらに，我々の評価結果より[4]，通信局所性の強い並列アプリケーションの場合，トポロジの規則性を活用することで 90 %の予測成功率を達成できることが報告されている．また，本論文では kill 機構やヒントビットを用いて予測ミスによるメッセージ出力を防止しているが，予測ミスによって間違った出力ポートへメッセージが出力される場合でも遅延が削減できることがわかっている[8]．

6. ま と め

本研究では Fat Tree トポロジにおいて予測ルータを用いて，オンチップネットワークとシステムエリアネットワークの 2 種類のネットワーク条件において遅延と予測ヒット率の評価を行った．評価結果より，それぞれのネットワーク条件で約 20 %の遅延削減ができること，ルータのパイプライン処理やリンクの移動に時間がかかるネットワークほど予測ルータの効果が大きくなることを確認した．また（4，4，r）Fat Tree において，上向きポートを同一経路として扱うことにって SS の予測ヒット率を 60 %近くまで上げられることがわかった．

今後の課題としては予測ルータの予測成功率を高めるための予測アルゴリズムの研究やネットワーク構造の研究を行っていく予定である．

謝辞 本研究は，一部科学研究費補助金基盤研究（C）課題番号 19500040，及び平成 21 年度国立情報学研究所・提案型共同研究の援助による．

参 考 文 献

1) 松谷 宏紀，鯉渕 道紘，天野 英晴，吉永 努：“低遅延オンチップネットワークのための予測ルータの評価”，第 7 回先進的計算基盤システムシンポジウム（SACSIS‘09）論文集，pp.209-218，May 2009.
2) 鯉渕 道紘，吉永 努，村上 弘和，松谷 宏紀，天野 英晴：“予測機構を持つルータを用いた低遅延チップ内ネットワークに関する研究”，情報処理学会論文誌コンピューティングシステム，Vol.1，No.2，pp.59-69,（2008）.
3) C. E. Leiserson，“Fat-trees：Universal networks for hardware-efficient supercomputing”，IEEE Transactions on Computers，vol.34，No.10，pp.892-901，1985.
4) H. Matsutani，M. Koibuchi，H. Amano and T. Yoshinaga：“Prediction Router: Yet Another Low Latency On-Chip Router Architecture”，Proceedings of the International Symposium on High-Performance Computer Architecture（HPCA‘09），pp.367-368，Feb 2009.
5) W.J. Dally and B. Towles：“Principles and Practices of Interconnection Networks”，Morgan Kaufman Publishers,（2003）.
6) Roweth，D. Jones，T：“QsNetIII an Adaptively Routed Network for High Performance Computing”，Proceedings of IEEE Symposium on High Performance Interconnects（HOTI‘08）. pp.157–164 (2008).
7) A. kumar，L.-S. Peh，P. Kundu and N. K. Jha：“Express Virtual Channels：Towards the Ideal Interconnection Fabric”，Proceedings of the ISCA‘07，pp.150-161,（2007）.
8) 吉永 努，村上 弘和，鯉渕 道紘：“2-D トーラスネットワークにおける動的通信予測の効果”，先進的計算基盤システムシンポジウム（SACSIS‘07），pp.219-226，May 2007.