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

鯉渕道紘[†] 吉永努[††] 村上弘和[††]
松谷宏紀[†††] 天野英晴[†††]

チップ内ネットワークはパケット処理を行うルータを多数用いることで，高スケーラビリティ，高スループットを実現している．しかし，ルータ構造はルーティング計算，仮想チャネル，クロスバなどの複雑な内部処理を行うため，リピータバッファを用いた従来のバス構造に比べて転送遅延が増大する．そこで，本稿では，(1) チップ内ネットワークにおいてパケットの転送遅延を削減するために，予測機構を持つルータを適用することを提案し，(2) 遅延，スループット，ハードウェア量，消費エネルギーの側面から複数の予測アルゴリズムを用いたシミュレーション評価を行う．評価結果より，単純な予測アルゴリズムを持つ予測機構の導入により，ハードウェア量は 20%，エネルギーは 26% の増加となるが，既存のワームホールネットワークに比べ最大 32%，無負荷パケットの遅延を削減することがわかった．また，単純な予測アルゴリズムを用いることで，予測が 100%成功する理想的なネットワークの遅延に比べて 7.4%の増加に留めることができることが分かった．

A Low-Latency Network-on-Chip using Predictive Routers

Michihiro Koibuchi,[†] Tsutomu Yosinaga,[††]
Hirokazu Murakami,[††] Hiroki Matsutani[†††]
and Hideharu Amano[†††]

Network-on-chip achieves both high scalability and high throughput, by using a large number of packet routers. However, every router performs internal complicated operations, such as routing computation, virtual-channel and crossbar allocation, that increase the packet latency, compared with traditional bus structure with repeater buffers. In this paper, (1) we propose to apply predictive routers into network-on-chip in order to reduce the packet transfer latency, and (2) we evaluate its latency, throughput, the amount of hardware, and energy. Evaluation results show that simple prediction algorithms reduce by up to 32% the unloaded packet latency, compared with a conventional wormhole network, although the prediction mechanism increases by 20%, and 26% the amount of hardware and energy, respectively. The simple prediction algorithms increase only by 7.4% the packet latency comapred with that of an ideal packet network where all predictions succeed.

1. はじめに

半導体技術の進歩によって単一チップ上にプロセッサやメモリ，I/O など複数の設計モジュールをタイル状に実装できるようになり，このようなタイル同士の結合にチップ内ネットワーク (Network-on-Chip: NoC) が用いられるようになった．チップ内ネットワークのルータは，高い動作周波数，高スループットを実現するためにパケット処理を複数に細分化するパイプライン方式を採用している．そして，パケットはルーティングテーブル計算，出力ポートの設定，アービタ，クロスバ転送などの複数のステージを経て入力ポートから出力ポートへ転送される．

WAN (Wide Area Network) や LAN などのチップ間通信と異なり，リンク長が mm オーダと極めて短いため，パケットのルータ遅延がネットワークの転送遅延の多くを占めることになる．よって，チップ内ネットワークは従来のリピータバッファを用いたバスに比べて，ホップ毎にデータ転送遅延が増大してしまう問題がある．チップ内通信では一般的にチップ間通信に比べて，ノード，通信粒度 (データサイズ，頻度) ともに小さく，細かくなるため通信遅延に対する要求

† 国立情報学研究所/総合研究大学院大学/JST
National Institute of Informatics/The Graduate University for Advanced Studies/JST
†† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications
††† 慶應義塾大学大学院 理工学研究科
Graduate School of Science and Technology, Keio University

が厳しい．そのため，通信遅延の削減がチップ内ネットワークの研究領域の大きな課題となっている[1]．さらに，Intel 80 コア NoC チップ[2] に代表されるように，コア数は増加する一方であり，2012 年にはチップ内に RAW アーキテクチャのタイルが 1,024 個配置可能という見積りもある[3]．つまり，コア数の増大に伴い，ルータ数も増加し，通信遅延の問題がより顕著になる傾向にある．

そこで，本稿では数十〜数百コアシステムを想定したチップ内ネットワークにおいて，パケットの遅延を隠蔽するために予測ルータ[4] を適用することを提案し，総合的な評価により有効性を明らかにする．

予測ルータは，各入力ポートが次に到着するパケットの出力ポートを事前に予測し，出力ポート，アービタ，クロスバなどのルータ内パイプライン処理をパケットの到着前に事前に投機的に実行する．予測ルータは，パケットが入力ポートに到着した直後に出力ポートにただちに転送される．したがって，予測が成功した場合 (すなわち，ルータが予測した出力ポートとパケットの適切な出力ポートが一致した場合)，パケット転送におけるルータ遅延を劇的に削減することができる．また，予測が外れた場合には並列に実行している通常のパイプライン処理によりパケット転送が行われることになるため，既存のルータのパイプライン処理と同じ遅延となる[4]．

メニーコアに代表される多数のノード (コア) を持つシステムでは強い規則性を持つ 2 次元トーラス，メッシュ，あるいはツリー系のトポロジでコア間を接続することが多い．このような場合には，トポロジとルーティングの規則性をパケットの出力方向の予測に用いることができるため高い予測成功率を達成することができる．

以降，2 章において関連研究，3 章において予測ルータについて述べる．4 章においてノード数と予測成功率の関係を明らかにし，5 章においてシミュレーションを用いて，予測ルータを用いたチップ内ネットワークの遅延，スループット，ハードウェア量，エネルギーについての評価結果を示し，6 章において結論を述べる．

2. 関 連 研 究

並列分散システムの相互結合網におけるルータの通信遅延を削減する研究は近年盛んに行われている．

投機ルータ[5] は，クロスバの調停と出力仮想チャネル割り当て等のルータ内パイプラインステージを同時に，投機的に実行することでルータ当たりのパケット転送遅延を削減する．しかし，パケットが到着した後，そのパイプライン処理を開始するためルーティング計算のステージ等の遅延を削減することが困難である．次章以降で述べる通り，投機ルータは予測ルータと併用することが可能であり，相反する技術ではない．

Preferred Path[6] はデータパスを変更することで遅延を削減することができるが，ソースルーティングを対象とし，さらにルータアーキテクチャが特化される．また，Preferred Path はクロスバを通過するデータパスと，それを迂回するデータパスの両方が必要となる．一方，予測ルータは予測の成否に関わらず，パケットは同一のクロスバを通過するデータパスにより転送される．

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]．したがって，局所性を持つ通信パターンに対する低遅延化効果は有さない．マッドポストマンスイッチングは，パケットヘッダの受信完了を待たずにパケット出力を開始する．よって，ビットシリアルリンクを使用した場合，すなわち，チップ間通信におけるパケットヘッダの逐次受信遅延の削減を念頭においた技術である．また，予測ルータは，動的通信予測による柔軟な設定ができる点でマッドポストマンスイッチングと異なる．

3. チップ内ネットワークにおける予測ルータ

3.1 パケットのパイプライン処理

予測ルータは，入出力ポート，制御回路，ならびにクロスバユニットにより構成されるパイプラインルータにおいて，予測器を制御回路に加えた構成を取る。

パイプラインルータにおけるパケット処理は 4 つのプリミティブに分けることができる．ここでは，説明のために 4 段パイプライン処理を採用している典型的なルータ[8] を用いて予測ルータの仕組みを簡単に説明する．この 4 段パイプラインルータは，(1) 入力ポートにおいてパケットのヘッダから出力ポート情報を解読，あるいはルータの制御ユニットから出力ポート情報を獲得し (Routing Computation: RC)，(2) 出力ポートを設定し (Virtual-Channel Allocation: VA)，(3) 出力ポートへのクロスバの設定を行い (Switch Allocation:SA)，(4) パケット転送 (Switch Transfer: ST) することで，パケットは出力ポートへ転送される．この様子を図 1 に示す．

予測ルータでは，RC ステージにおいて，同時に予測機構 (predictor) によりあらかじめ設定された出力ポートへパケットを転送する (Predictive Switch Transfer: PST)．予測が成功した場合には，パケットの先頭部

分 (ヘッダフリット) はそのまま出力ポートに転送される．チップ内予測ルータでは予測が失敗した場合には，間違った出力ポートに転送されたフリットを削除し，上記 4 ステージの通常のルータのパイプライン処理を行う．

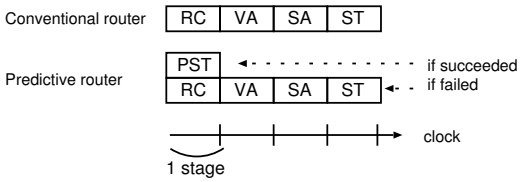

図 1 ルータのパイプライン処理の例

我々は，チップ間通信を対象とし，シリアル転送，あるいはナローチャネル転送においてパケットのヘッダフリットのルーティング情報が到達次第，パイプライン処理を開始するルータ[4] について議論してきた．予測ミスを検出した場合にも一部のビット転送はすでに行われているため，予測ミスパケットの隣接ルータへの伝搬を止めることは困難となる．同様の問題はマッドポストマンスイッチングにおいても指摘されている．そこで予測ミス発生時に一部のフリットが隣接ルータに伝搬する問題に対処するパケットの処理方法，予測ミスを最小限に抑える賢い予測アルゴリズムについて検討を進めてきた[4]．

一方，本論文が対象とするチップ内ルータは，パラレル転送，つまりファットチャネル転送を行っているため，入力ポートにおいて各フリットを一度，完全にバッファリングした後，RC/PST 処理を行う．そのため，予測ミスが生じた場合，自身のルータの出力ポート，あるいは隣接ルータの入力ポートにバッファリングしている間に削除することが可能となる．よって，いずれの削除方法においても，チップ内ネットワークでは予測ミスパケットが複数のルータに伝搬し，他のパケットの進行をブロックする状況は生じない．また，軽量化，予測ミスパケットの伝搬防止を含めたルータアーキテクチャに特化した詳細な議論，定量的な評価は[9] にまとめている．

3.2 予測アルゴリズム

入力ポートにおいて，次に到着するパケットの出力ポートを予測するアルゴリズムは，予測ルータを用いたチップ内ネットワークの性能に大きく影響を与える．もっとも単純な予測アルゴリズムは静的に出力ポートを予測する方法である．静的直進予測アルゴリズム (SS) は入力パケットが常に同一次元を直進すると予測する．すなわち，2 次元トーラスにおいて北の入力ポートに到着したパケットは南の出力ポートへ，東からは西へといった具合である．ランダム予測アルゴリズム (Rand) はランダムに予測出力ポートを選択する．

一方，直前ポート予測アルゴリズム (LP) は，入力パケットが 1 つ前のパケットと同一の出力ポートを選択すると動的に予測する．したがって，通信履歴は入力ポート毎に 1 つ分が必要となる．より洗練された方法としては，パターンマッチ予測アルゴリズム (SPM) がある．SPM は文献[10] で提案されたパターンマッチングに基づくユニバーサル予測アルゴリズムに，系列の長さ制限等の制約条件を付けたものである[4]．過去の通信履歴から繰り返しパターンを検索することによって，並列プログラムが持つ通信の規則性を抽出しやすい利点を持つ．

4. 予測成功率

本章では典型的なチップ内ネットワークのトポロジであるトーラス，Fat ツリーにおける予測成功率を経路の分散から求める．

4.1 トーラス

4.1.1 経路数

トーラストポロジは対称性を持つため，次元順ルーティングにおいてルータ間チャネルを通過する経路数 (出発地-目的地対数) は均一となる．ここで，各ノード (コア) は 1 つのルータと 1 つのプロセッシングエレメント (PE) で構成されているとする．

図 2 に示したように，次元順ルーティングを用いた k-ary 1-cube トーラス (k は奇数) におけるルータ間チャネル上の経路数 T_{1d}，その中で直進する経路数 $T_{1d_{ss}}$ はそれぞれ式 1,2 のようになる．

$$T_{1d} = 1 + 2 + \ldots + (\frac{k}{2} - \frac{1}{2}) = \sum_{i=1}^{\frac{k}{2}-\frac{1}{2}} i \qquad (1)$$

$$T_{1d_{ss}} = 0+1+\ldots+(\frac{k}{2}-\frac{3}{2}) = \sum_{i=1}^{\frac{k}{2}-\frac{1}{2}} (i-1) = \sum_{i=1}^{\frac{k}{2}-\frac{3}{2}} i \qquad (2)$$

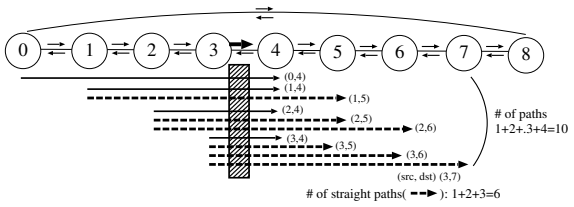

図 2 1 次元トーラスにおける経路の分布 (k は奇数)

チップ内ネットワークでは，2 次元/3 次元トーラス

が対象となる[11] ため，次に n 次元トーラスへ拡張を行う．次元順ルーティングの規則性を利用して，あるルータへの入力チャネルを通過する経路数 T と，その中で直進する経路数 T_{ss} はそれぞれ式 3, 4 のようになる．

$$T = k^{n-1} \sum_{i=1}^{\frac{k}{2}-\frac{1}{2}} i \tag{3}$$

$$T_{ss} = k^{n-1} \sum_{i=1}^{\frac{k}{2}-\frac{3}{2}} i \tag{4}$$

次元順ルーティングでは，あるルータにおいて i 次元入力チャネルから j 次元出力チャネル (i≤j) へ通過する経路数 $T_{(i,j)}$ は，そのルータにおいて $1, \ldots, (j-1)$ 次元方向の移動が完了した経路の集合となるため式 5 となる．また，i 次元入力チャネルからそのルータに連結している PE を目的地とする経路数 $T_{PE}(i)$ は式 6 となる．

$$T_{(i,j)} = \left(\frac{k}{2} - \frac{1}{2}\right)^2 k^{n+i-j-1} \tag{5}$$

$$T_{PE}(i) = \left(\frac{k}{2} - \frac{1}{2}\right) k^{i-1} \tag{6}$$

4.1.2 予測成功率

各 PE はポアソン分布に従ってパケットを独立に生成，注入し，目的地はランダム (ユニフォームトラフィック) と仮定する．ユニフォームトラフィックは，局所性を持たないため予測が難しいアクセスパターンといえる．この場合，予測成功率は，ルータにおいて入力チャネルを通過する経路の中で予測した出力チャネルを通過する経路の割合となる．直進予測の成功率 (P_{ss})，ランダム予測の成功率 (P_{rad}) を表 1 に示す．

直前ポート予測の予測成功率 P_{lp} は，ある入力チャネルを通過するパケットが 2 回連続で 1 つの出力チャネルに転送される確率となる．よって，経路の分布から直前ポート予測の成功率は表 1 のようになる．

ポアソン分布に従って各 PE が独立にパケットを注入するため，次に到着するパケットの出力チャネルを，1 つ前のパケットの出力チャネルと同じと予測する LP と，履歴の中からパターンマッチングを取る SPM を用いた場合の予測成功率は同じとなる．同様にして，PE からの入力チャネルポートにおける予測成功率 ($P_{\{ss,rad,lp,spm\}_{PE}}$) を表 2 に示す．なお，PE からの入力チャネルポートにおける直進予測は，次元順ルーティングの特性から第 1 次元方向出力チャネルを予測するものとする．

最後に次元内ノード数 k が偶数の場合を求める．偶数の場合，$\frac{k}{2}$ ホップ離れたノードへの経路が 2 つ存在するが，この経路を半分ずつ使いチャネル利用率が均一と仮定する．この場合，各チャネルを通過する経路数が次式のように異なる．

表 1 i 次元入力チャネルポートにおける予測成功率

P_{ss}	$\frac{T_{ss}}{T}$
P_{rad}	$\frac{1}{n}\sum_{j=1}^{n}\frac{1}{2(n-j+1)}$
$P_{lp,spm}$	$(\frac{T_{ss}}{T})^2 + (\frac{T_{PE}(i)}{T})^2 + 2\sum_{j=i+1}^{n}(\frac{T_{(i,j)}}{T})^2$

表 2 入力ローカルチャネルポートにおける予測成功率

$P_{ss_{PE}}$	$\frac{T_{PE}(n)}{k^n-1}$
$P_{rad_{PE}}$	$\frac{1}{2n}$
$P_{(lp,spm)_{PE}}$	$\frac{2\sum_{i=1}^{n} T_{PE}(i)^2}{(k^n-1)^2}$

$$T = k^{n-1} \sum_{i=1}^{\frac{k}{2}} (i - \frac{1}{2}) = k^{n-1} (\sum_{i=1}^{\frac{k}{2}} i - \frac{k}{4}) \tag{7}$$

$$T_{ss} = k^{n-1} \sum_{i=1}^{\frac{k}{2}-1} (i - \frac{1}{2}) = k^{n-1} (\sum_{i=1}^{\frac{k}{2}-1} i - \frac{k}{4} + \frac{1}{2}) \tag{8}$$

偶数の場合における各予測アルゴリズムの予測成功率は，(式 7，8 を用いた) 表 1，2 となる．

4.2 Fat ツリー

Fat ツリーにおける直進予測 (SS) は下位チャネルからの入力ポートにおいては 1 つの上位チャネルを，上位チャネルからの入力ポートにおいては，1 つの下位チャネルを予測するものとする．また，出発地においてパケットの経路を定めることで静的に経路分散しているものとする．

4.2.1 経 路 数

トーラスの場合と異なり，Fat ツリーは経路数の分布がチャネルの階層毎に異なる．ここでは各ルータの階層数 i を PE からの距離と定義し，PE を便宜的に階層 0 とする．

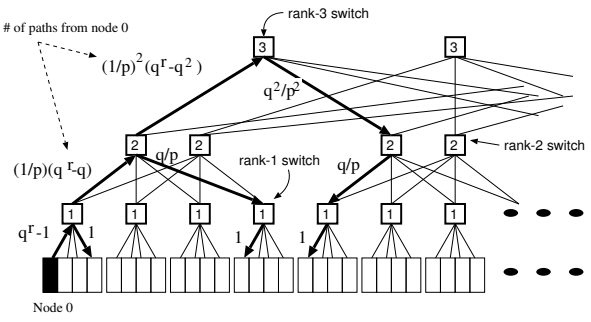

図 3 Fat ツリー (p=2,q=4,r=3)

表 3 階層 i ルータにおける入力下位チャネルポートにおける予測成功率

P_{ss}	$\frac{T_{up(i)}}{T_{up(i-1)}}$
P_{rad}	$\frac{1}{p+q-1}$
$P_{lp,spm}$	$p(\frac{T_{up(i)}}{T_{up(i-1)}})^2 + (q-1)(\frac{T_{dw_{dw}(i)}}{T_{up(i-1)}})^2$

図 3 に 1 つの出発地からの経路数を示したが，階層 i ルータから上位チャネルを通過する経路数 $T_{up}(i)$ は，上位リンク (チャネル) 数 p，下位リンク数 q，階層数 r を用いて式 9 のようになる．

$$T_{up}(i) = (\frac{q}{p})^i(q^r - q^i) = \frac{q^{i+r} - q^{2i}}{p^i} \quad (9)$$

階層 i ルータにおいて，入力下位チャネルから出力下位チャネルへ通過する経路数 $T_{dw_{dw}(i)}$ は Fat ツリーの規則性から式 10 のようになる．

$$T_{dw_{dw}(i)} = \frac{q^{i-1}}{p^{i-1}} \quad (10)$$

4.2.2 予測成功率

トーラスの場合と同様にして，経路数の分布からユニフォームトラフィックにおける予測成功率を表 3 に示す．なお，ルータにおいて上位チャネルを通過する経路の中で，1 つの下位チャネルを通過する経路の割合は $\frac{1}{q}$ と均一になる．

5. 評　　価

本章では，予測成功率，無負荷状態のネットワークにおけるパケット遅延，スループット，電力，ハードウェア量について，予測ルータと既存の投機ルータ，ワームホールネットワークとの比較をシミュレーションにより行う．

5.1 予測成功率

表 1，2 からユニフォームトラフィックにおける次元順ルーティングのパケットの予測成功率を算出した．その各ネットワークサイズにおける結果を図 4 に示す．

本研究の対象は，数十〜数百コアのチップ内ネットワークであるが，参考のため 1,000 コア規模までの結果を示し，傾向が分かりやすいようにした．

図には含めていないが，5.3 節で示す 2 次元チップ内ネットワークシミュレーションにおいて測定した予測成功率との誤差は 3.3%と小さかった．図 4 より，直進予測 (SS) と直前ポート予測 (LP) はコア数が多くなるにつれて予測成功率が向上していき，500 ノードを越える辺で徐々に一定に近づいていくことが分かる．また，近隣通信が多い場合は直進予測は成功率が低くなることが報告されているが[4]，局所性のないユニフォームトラフィックでは数十ノードから数百ノードのすべてのサイズにおいて直進予測が直前ポート予測に比べて高い予測成功率となった．これは，次元順ルーティングの特徴である，パケットが次元内で必要ホップ数直進した後，他の次元に転送されることに起因する．

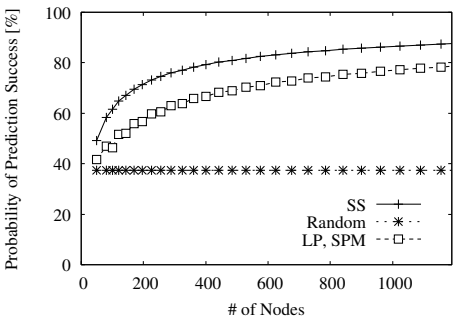

図 4 k-ary 2-cube トーラスにおける予測成功率

同様にしてツリー系のトポロジにおける予測成功率を図 5，6 に示す．“LP,SPM with Up only” は，ルート以外のルータにおいて入力ダウンチャネルポートのみ予測を行った場合を示している．ツリー (1,4,r) は Fat ツリー (2,4,r) に比べて上位リンク数が少ないため，各ノード対の経路数が少なくなる．その結果，予測成功率が高くなっている．トーラスの場合と同様の理由で，直進予測 (SS)，直前予測 (LP) の予測成功率がランダム予測成功率に比べ，最大 65%増と極めて高い．ツリー系のトポロジではノード数が増加するにつれて，直進予測，直前予測の予測成功率は向上していき，64 ノード以上はほぼ一定となった．よって，数十から数百コアという本議論の対象サイズにおいて，予測機構をほぼ最大限利用することができるといえる．

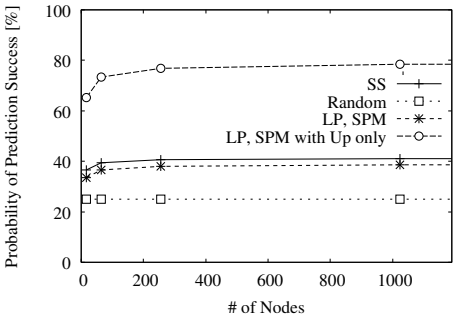

図 5 ツリー (1,4,r) における予測成功率

5.2 無負荷状態におけるパケットの転送遅延

無負荷状態のワームホールネットワークにおけるパケット転送遅延は式 11 で求まる[12]．T_r, T_a, T_s, d, h

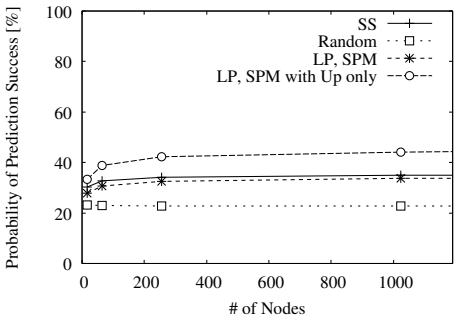

図 6 Fat ツリー (2,4,r) における予測成功率

はそれぞれルータにおけるルーティング，調停，スイッチング遅延，ホップ数，ヘッダを表す．

$$Lat = T_{LinkProp}(d+1) + (T_r + T_a + T_s)d + \frac{PacketSize + hd}{\text{LinkBW}} \quad (11)$$

予測が成功した場合には $T_r = 0, T_a = 0$ とし，前章での予測成功率を用いて，ユニフォームトラフィックにおけるパケットの平均転送遅延を結果を図 7， 8 に示す．

本研究の対象は，数十〜数百コアのチップ内ネットワークであるが，参考のため 1,000 コア規模までの結果を示した．

その他のパラメータは，チップ内ネットワーク向けに軽量化した 3 段パイプラインルータ[4] から抽出した ($T_r = 1, T_a = 1, T_s = 1$)．パケット長はチップ間通信の場合に比べて短い 16 フリットとした．

また，ケーススタディとして Intel 80 コアのチップ内ネットワークの 5 サイクルルータ (調停 2 サイクル) を想定した結果を図 9 に示すが，3 サイクルルータの場合と比べて各予測方式の遅延削減効果の傾向に違いはない．

各図において，横軸はノード数，縦軸は無負荷状態のネットワークにおけるパケット遅延を表している．“Conv” は予測を行わない基となるワームホールネットワーク，“Ideal” は予測が 100%成功すると仮定した理想的な予測ルータネットワーク，“Spec” は，予測は行わないが，2 章で述べた投機実行により 2 サイクル転送する場合，“SS+Spec” は直進予測を用いた予測ルータを用い，かつ，予測が失敗した場合，投機実行により 2 サイクル転送するという予測ルータと投機ルータを組み合わせた場合の結果を各々表す．

図 4， 7， 9 より，予測成功率が高い場合ほど遅延が小さくなることが分かる．つまり，3 つの予測アルゴリズムの中で予測成功率が高い直進予測が最も遅延が小さく，ランダム予測が最も遅延が大きい．

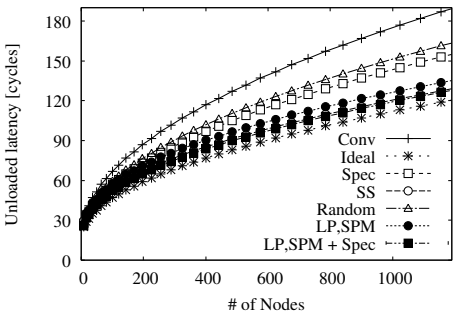

図 7 無負荷状態のネットワークにおけるパケット遅延 (k-ary 2-cube トーラス)

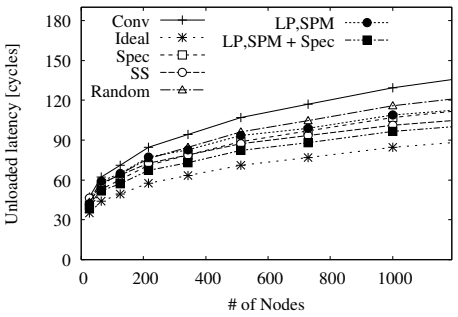

図 8 無負荷状態のネットワークにおけるパケット遅延 (k-ary 3-cube トーラス)

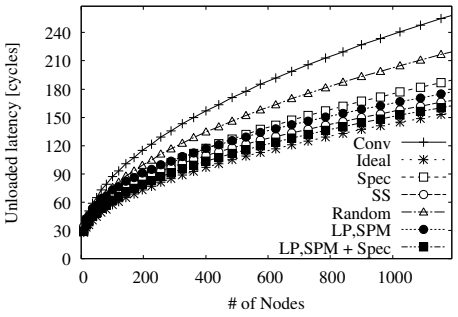

図 9 無負荷状態のネットワークにおけるパケット遅延 (k-ary 2-cube トーラス，Intel 80 コア NoC モデル)

また，500 ノード規模やそれ以上と大きくなるほど予測機構を導入したことによる遅延削減の効果が大きいといえる．また，数十ノードから数百ノードの範囲内において，元のワームホールルータに比べて予測ルータを用いることで最大 32%遅延を削減できていることが分かる．また，直進予測 (SS)，直前ポート予測 (LP) を用いた予測ルータは，予測が 100%成功したと仮定した理想的な場合に比べて，遅延が 7.4%増えるに留まり，数十ノードから数百ノードの範囲内では極めて有効であることが分かる．

次に，無負荷状態のツリー系のトポロジにおけるパケットの平均転送遅延を図 10，11 に示す．パケットの目的地はランダムに選択し，均一な分布となるよう

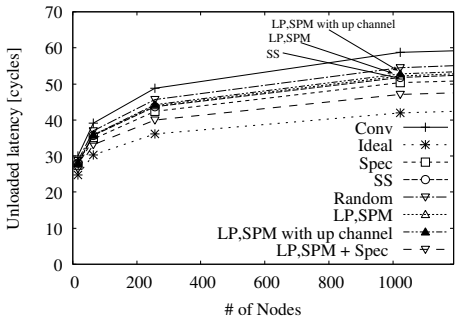

図 10 無負荷状態のネットワークにおけるパケット遅延 (ツリー (1,4,r))

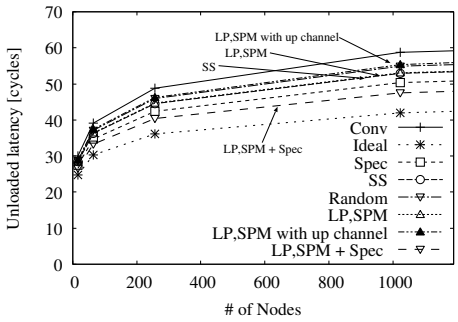

図 11 無負荷状態のネットワークにおけるパケット遅延 (Fat ツリー (2,4,r))

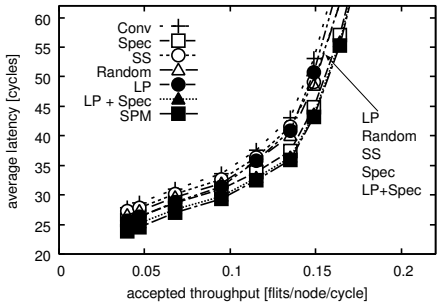

図 12 スループットとレイテンシ (2D トーラス，64 PEs, LU 分解)

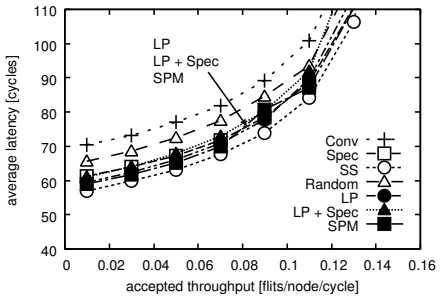

図 13 スループットとレイテンシ (2D トーラス，256 PEs, ユニフォーム)

にした．その他のパラメータはトーラスの場合と同様である．これらの結果より，各ルータにおいて入力ダウンチャネルにのみ予測ルータを設置することで一定の遅延削減効果が得られていることがわかる．これは入力アップチャネルからのパケットの出力チャネルの予測が難しいことに起因する．ツリー系のトポロジでは予測成功率が低いため，予測ルータは投機ルータに比べてパケット遅延が大きくなっている．よって，ツリー系のトポロジにはより賢い予測アルゴリズムが必要であるといえる．

5.3 スループット

スループット評価のために，C++で記述されたフリットレベルシミュレータを用いて評価を行った．予測成功率の評価環境と同様となるように，2 次元トーラスにおける 2 本の仮想チャネルを用いた次元順ルーティングを採用した．また，各ノードは 1 つのルータと PE を持つものとし，64，256 ノード構成規模とした．通信パターンとしては，ユニフォームトラフィックと NAS Parallel Benchmark（NPB）プログラム LU 分解から得られた通信パターンを用いた．

本シミュレータはフリット単位で処理をする一方，LU 分解のトラフィックトレースにおいて各パケットは，生成クロック，出発地，目的地，データサイズ (x Byte) の 4 項目で構成されている．そのため，本シミュレータにおいて LU 分解の評価を行うために 16Byte/フリットの可変長パケットとして扱った．一方，各パケットは 16 フリット (内ヘッダ 1 フリット) とした．

また，ネットワーク内において，特に空間的，時間的局所性を持つ LU 分解のトラフィックパターンではパケットの衝突が発生する．そのため，前節での無負荷状態のネットワークにおける評価と異なり，パケットの衝突が，本評価の平均遅延に影響している．

図 12，13 は，平均遅延と受信トラフィックの関係を示している．両図のように可変長 (LU 分解)，固定長パケットのいずれにおいても予測ルータは遅延を削減することができている．さらに，各パケットがネットワークに留まる時間が短くなることにより，パケットの衝突の発生が抑えられ，その結果，スループットについても約 22%改善した．LU 分解においては，SPM が直進予測 (SS) に比べて高いスループット，低遅延を達成していることが分かる．これは，パケットのホップ数が少ない局所性を持つトラフィックでは，ルータ内を直進するパケットの割合がへることに起因する．また，トラフィックの空間的，時間的局所性に応じて予測出力ポートを動的に選択する SPM はユニフォームトラフィック，LU 分解の両方において安定した性能を達成していることが分かった．

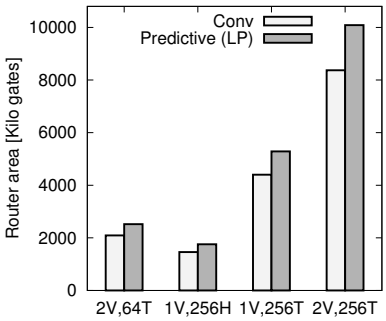

図 14 16 × 16 2 次元トーラスネットワークのルータの総ハードウェア量

表 4 5-ポート仮想チャネルルータと NI の面積の内訳 (Kilo gate)

	Conv. ルータ	予測ルータ (LP)
Xbar & Arb	2.448 (7.5%)	2.698 (6.8%)
Channels	29.894 (91.4%)	36.070 (91.5%)
Misc	0.360 (1.10%)	0.634 (1.6%)
Total	32.701	39.401
NI	6.757	6.757

5.4 ハードウェア量

ネットワークのハードウェア量は，PE のネットワークインタフェース (NI) とルータが大部分を占める．評価のために 3 ステージ (RC,VC/SA, ST) のパイプライン処理を行うワームホールルータを設計した．これは，我々がこれまで議論，評価してきたルータ[11] に予測機構を追加した構成を取り，VC と SA は統合され 1 サイクルで実行することができる．フリットは 64 ビット幅，各入力仮想チャネルは 4 フリットを格納できる FIFO バッファを持つ．ルーティング実装は，ソースルーティング法を仮定し，ネットワークサイズに大きさが依存するルーティングテーブルは持たない．また，仮想チャネル数は 2 本とした．ネットワークインタフェース (NI) は，ハードウェア量を抑えるために，フリットをスイッチングするために必要となる 2 フリットの FIFO バッファを持つ設計とした．そして，Synopsys Design Compiler バージョン Y-2006.06-SP2, ASPLA 90nm スタンダードセルライブラリを用いてチップ内ネットワークルータ，NI を合成した．また，合成条件は面積優先とした．

図 14 は，2 本の仮想チャネルを持つ 64 ノード 2D トーラス，256 ノード H-Tree，256 ノード 2D トーラス，2 本の仮想チャネルを持つ 256 ノード 2D トーラスネットワークにおけるルータの合成結果である．これらの図において，“Conv” は，基となるワームホールルータを表し，“Predictive (LP)” は LP 予測アルゴリズムを用いた予測ルータを表す．図 14 より，予測機構付きネットワークがトーラス，ツリー系トポロジにおいて 20%のハードウェア量の増加で実現できることが分かった．

5.5 消費エネルギー

前節で実装した予測ルータの消費電力を求めるために，(1) Design Compiler でルータ回路を合成し，(2) Verilog-XL でシミュレーションを行い Switching Activity Interchange Format（SAIF）を生成し，(3) Power Compiler でこの SAIF をもとに消費電力を解析した．解析には，動作電圧 1.0V の 90nm CMOS プロセスを使用した．

パケットストリームは，最大リンクバンド幅の 30%の利用率となるようにした．各ヘッダフリットは目的地アドレスを含み，データフリットはランダムなデータが入ったペイロードとした．ハードウェア量の評価に用いた予測ルータにおいて，予測の成否に関わらず予測に基づくパイプライン (1 サイクル) と通常のパイプライン (3 サイクル) の両方を実施し，ルータ外の物理チャネルには片方のみにフリットを転送する．そのため，図 15 に示したように，ルータにおいて 1 ビット転送するために必要となる消費エネルギーは予測ルータを用いる場合，26%増加する．ただし，現状では，Alpha 21364 マイクロプロセッサの電力に占めるネットワークの割合は 20%，MIT RAW プロセッサでは 36%とチップ全体の電力に占めるネットワーク自体の割合は支配的ではない[13]．よって，このエネルギーの増分は致命的にはならない可能性が高い．

さらに，パイプラインの段数が深いルータの場合，電力削減のため，予測が成功した場合における通常のパイプライン転送を途中で中止する機構を追加する等の検討が重要になるが，本評価では軽量な 3 段パイプラインルータであるため，この中止機構は搭載していない．なお，予測ルータの導入により，ホップ数，経路が変更されることはないため，元のワームホールルータと予測ルータにおける 1 ビットあたりの出発地から目的地までの転送エネルギー比はルーティングアルゴリズムによらず一定となる．

6. ま と め

本稿では，予測機構を持つルータを用いて投機的にパケットのパイプライン処理を実行することで，パケットの遅延を削減するチップ内ネットワークについて様々な側面から評価を行った．評価結果より，チップ内ネットワークの典型的なトポロジであるトーラス．Fat ツリーにおいて単純な予測アルゴリズムを用いることで既存のワームホールネットワークに比べて

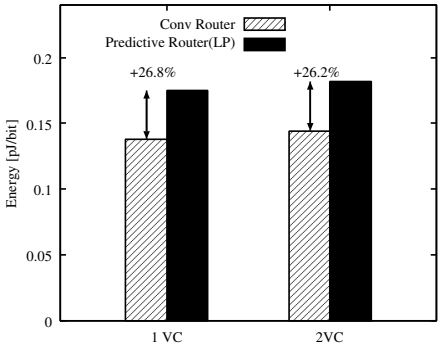

図 15 予測ルータの電力比較

32%のパケット遅延の削減，また，予測が 100%成功した理想的なネットワークの遅延に比べて 7.4%の増加に留まることが分かった．トレードオフとして予測機構の導入によるハードウェア量の増加は 20%，エネルギーの増加は 26%であった．ただし，現状ではこれらの点について，チップ全体に占めるネットワークの割合は支配的とはなっていない[13] ため，致命的にはならないと考えられる．また，コア数が増加するにつれて，予測機構を持つルータのパケット転送遅延の削減効果が大きくなることが分かった．

今後は，適応型ルーティング向け予測ルータの拡張を行う．さらに，予測ミスが生じた場合にも経路変更することでそのままパケットを目的地まで配送する仕組みについても提案する予定である．

謝辞 本研究の一部は，科学技術振興機構「JST」の戦略的創造研究推進事業「CREST」における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「ULP-HPC:次世代テクノロジのモデル化・最適化による超低消費電力ハイパフォーマンスコンピューティング」により行った．

参 考 文 献

1) Final report. Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems, available at http://www.ece.ucdavis.edu/õcin06/, December 2006.
2) S. Vangal, et al. An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS. In *Proceedings of the International Solid-State Circuits Conference*, February 2007.
3) T. M. Pinkston and J. Shin. Trends Toward On-Chip Networked Microsystems . *International Journal of High Performance Computing and Networking*, Vol. 3, No. 1, pp. 3–18, 2005.
4) 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) 論文集, pp. 219–226, May 2007.
5) L.S. Peh and W. J. Dally. A Delay Model and Speculative Architecture for Pipelined Routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 256–266, January 2001.
6) G. Michelogiannakis, D. Pnevmatikatos, and M. Katevenis. Approaching Ideal NoC Latency with Pre-Configured Routes. In *Proceedings of the ACM/IEEE International Symposium on Networks-on-Chip*, May 2007.
7) A. Kumar, L.S. Peh, P. Kundu, and N. K. Jha. Express Virtual Channels: Towards the Ideal Interconnection Fabric. In *Proceedings of the 34th International Symposium on Computer Architecture (ISCA)*, June 2007.
8) W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
9) 松谷宏紀, 鯉渕道紘, 天野英晴, 吉永努. 予測機構を持った低遅延オンチップルータアーキテクチャ. 情報処理学会技術研究報告 [計算機アーキテクチャ], May 2008 (to appear).
10) P. Jacquet, W. Szpankowski, and I. Apostol. A universal predictor based on pattern matching. *IEEE Trans. Info. Theory*, pp. 1462–1472, 2002.
11) 松谷宏紀, 鯉渕道紘, 天野英晴. Network-on-Chip における Fat H-Tree トポロジに関する研究. 情報処理学会論文誌コンピューティングシステム, Vol. 48, No. SIG 13(ACS19), pp. 178–192, August 2007.
12) J. L. Henessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach Fourth Edition, Appendix E: Interconnection Networks*. Morgan Kaufmann, 2006.
13) V. Soteriou and L.S. Peh. Exploring the Design Space of Self-Regulating Power-Aware On/Off Interconnection Networks. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 3, pp. 393–408, March 2007.