Vol.2012-ARC-201 No.7
2012/8/1

テクニカルノート

TSVを用いる間接NoCの評価

飯尾 亮介[1]　平木 敬[1]

概要：シングルスレッド性能向上の鈍化によるメニーコアへの転換が行われつつある現在，パフォーマンススケーリングを得るためにNoCに関して多くの研究がなされている．特に，半導体製造技術の向上により可能となったシリコン貫通電極（TSV）による三次元積層を用いた3D-NoCが注目されている．3D-NoCがもたらす利点の1つとして，リンクの長さを短くしレイテンシを小さくすることが可能である点が挙げられる．本論文では，その利点をMultistage Interconnection Network (MIN) に適用し，k-ary n-tree及びDragonflyについてのパフォーマンス及び資源消費に対する評価を行った．

Evaluation of Indirect NoC using TSV

Ryosuke Iio[1]　Kei Hiraki[1]

Abstract: Recently, the improvement of single thread performance has decreased. This has led to focus on multi thread performance on many cores. To obtain performance scaling, a lot of research have been performed. Due to the improvement of semiconductor manufacturing, 3D stacking chips and 3D-NoC can be designed and developed using Through Silicon Via (TSV). One of the advantages produced by it is to have the ability of reducing the length of links and the latency. In this paper, by adopting this advantage to Multistage Interconnection Network (MIN), we evaluated k-ary n-tree and Dragonfly in terms of performance and resource consumption.

1. はじめに

現在，シングルスレッド性能向上の鈍化によるメニーコアへの転換が行われつつあり，このような環境においては，相互接続アーキテクチャが性能に与える影響が大きくなってくる．レイテンシやバンド幅に対するスケーリングを得るためネットワークオンチップ (NoC) が提案され，多くの研究がなされてきた．

他方，半導体製造技術の向上により，シリコンチップを縦に重ねその間を導線や無線技術で接続する三次元半導体を製造することが可能になりつつある．その中でも特に，シリコン貫通電極 (TSV) を利用した三次元積層チップは次世代半導体製造技術として有力なものと考えられており，NoCに適用した際の研究が行われてきた．

[1] 東京大学
The University of Tokyo

これらを組み合わせた，TSVを用いる3D-NoCは以下のような利点を持つ．

まず，二次元メッシュから三次元メッシュのように，NoCのトポロジ次元を自然に増やすことが可能となる．二次元チップ上に$4 \times 4 \times 4$の三次元メッシュを実装しようとすると，図1のように3次元目の接続に無理が生じてしまうが，3次元目のリンクをTSVとして3枚のチップを用いる実装にすると，概念的配置と実際の配置が良く一致する．

次に，レイテンシの軽減が可能である点が挙げられる．先に挙げた利点とかぶる部分があるが，図1において，3次元目のリンクの長さが1,2次元目のリンクの3倍程度となっている．動作周波数を上げるために1,2次元目のリンクと同程度の長さになるようにリピータを挟むと，レイテンシもそのまま3倍になってしまう．一方TSVを用いた場合には例えば8層をまたがるTSVリンクを利用しても二次元上のリンクより十分に短くなり，大幅にレイテンシ

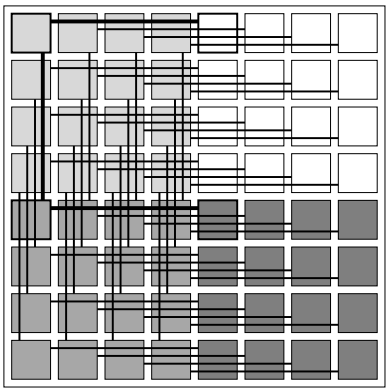


図 1 二次元チップにおける三次元メッシュの配置例

を削減できる場合がある．

本論文では，間接ネットワーク，特に k-ary n-tree 及び，それをグループ内トポロジとして利用した Dragonfly についての評価を行う．これらは Multistage Interconnection Network (MIN) と呼ばれる間接ネットワークに属し，多くのリンクからなっていることが多い．特にリンクとルータを H-tree をベースとして構築する場合，ルート付近のステージのリンクが非常に長くなり，レイテンシの悪化やターンアラウンドタイムを隠蔽するための大きなクレジットバッファ及びフリットバッファを用意する必要が出てくるという問題がある．このような問題に起因する性能の劣化は，先述のように TSV を用いることで軽減することが可能である．

本論文の構成は以下のようになっている．2 章で関連研究について述べ，3 章では，評価を行った NoC デザインについて要約を述べる．次の 4 章では実験及び評価手法について述べる．5 章においては，実験の結果とそれらに関する考察を述べる．最後に 6 章で結論と今後の課題について述べる．

2. 関連研究

Zia らは，本論文と同様 TSV を用いた NoC についての評価を行っている [12]．対象としているトポロジは flattened butterfly，fat tree，tree，3D mesh，及び論文中で新たに提案されている CNOC である．フロアプランには Kannan らの提案した階層的フロアプランアルゴリズム [4] を用いており，評価を行っている最大のノード数が大きいという特徴を持っている．しかしながら，消費電力や TSV の数など資源消費に関する評価は詳細に行われているにもかかわらず，パフォーマンスに関する評価は行われていない．

fat-tree は [7] において提案された間接トポロジであり，後 Ohring らによってリンクの数やルータの次数によって一般化された形式が示された [8]．そのうち，各ステージごとに等しいスイッチの数を持ち最上位レベル以外のルータが等しい次数であるものを k-ary n-tree と呼ぶ．Gomez らは，k-ary n-tree における静的ルーティングを提案した [2]．この手法を用いることで，任意のノード間のパケット送信をインオーダにできるだけでなく，負荷分散を静的に行うことにより，パフォーマンスの向上も得られている．しかし，TSV を用いた NoC という文脈での実験が行われておらず，本論文ではそれを行うことにより，更なる評価を与える．

Reinemo は [10] において，Dragonfly[6] と fat-tree に対して比較評価を行っている．評価は大規模なコンピュータクラスタの文脈で行われていないという問題があった．比較的小規模なチップ内のインターコネクションとして用いた場合は精査されていない．そのため小規模なケースにおいても同様の傾向を示すかを評価することは重要である．

これらの論文では，さらに我々の知る限り他の論文においても，NoC における Dragonfly の評価およびそれとの fat-tree の比較は行われていない．本論文は，TSV を利用した Dragonfly の実装および評価，比較を行っている点において特徴的である．

3. NoC デザイン

本節では，評価対象とした k-ary n-tree 及び Dragonfly のトポロジやルーティングに関する性質及びアルゴリズムの要約を行い，チップ上への物理的な設計に関して述べる．

表 1 は，本評価において対象としたネットワーク構成を示している．k-ary n-tree のルータ次数に関しては，前者は最上位のルータ，後者はそれ以外のルータの値を示している．Dragonfly のルータ次数およびルータ数に関しては，前者はグループ内リンクのみを持つルータ，後者はグループ間リンクを持つルータの値を示している．

以下の説明において特に断りがない限り，線は双方向リンクを，円はルータを，矩形はノードをそれぞれ表す．

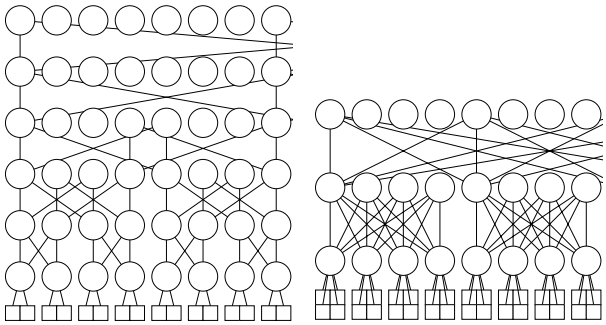
3.1 k-ary n-tree

k-ary n-tree はマルチステージネットワークの一種である．n 段のレベルを持ち最上位レベルを除く各レベルのルータは，上位レベル下位レベルそれぞれに対して k 本，合計 $2k$ 本のリンクを持つ．すなわち，ルータは nk^{n-1} 個，双方向リンクは nk^n 本，ノード数は k^n となる．

図 2(a) は 2-ary 6-tree，図 2(b) は 4-ary 3-tree の概念図を示している．ただし，リンクの数が非常に多いため，一部のリンクは表示していない．本論文においては，ルー

トポロジ	ルーティング	ルータ次数	ルータ数
2-ary 6-tree	Adaptive NCA	2/4	32/160
2-ary 6-tree	Deterministic	2/4	32/160
4-ary 3-tree	Adaptive NCA	4/8	16/32
4-ary 3-tree	Deterministic	4/8	16/32
Dragonfly 2-4	Deterministic Min	4/5	96/32
Dragonfly 4-2	Deterministic Min	8/7	16/16

表 1 対象 NoC 構成



(a) 2-ary 6-tree (b) 4-ary 3-tree

図 2 *k*-ary *n*-tree

ティングアルゴリズムとして Adaptive Nearest Common Ancestor (ANCA) 及び Gomez らによって提案された静的ルーティング [2] を用いた．

ANCA は，2 つの段階からなる最短適応ルーティングである．最初の段階においては，送信元ノードと送信先ノードが共通して持つルータ（以下，共通祖先ルータと呼ぶ）まで木を辿っていく．一般に共通祖先ルータは複数あり，この段階に適応ルーティングを行う余地がある．本論文ではクレジットベースのパケット混雑判定を行い，よりクレジットが多く残っている側のリンクへとルーティングを行う．次の段階においては，共通祖先ルータから送信先ノードまで木を逆方向に辿っていく．この段階では辿れるパスが 1 つだけに固定されるため，適応ルーティングを行うことはできない．

Gomez らによって [2] で提案された *k*-ary *n*-tree に対する静的ルーティングを用いると，共通祖先ルータへのパス及び共通祖先ルータからのパスを静的に負荷分散させることにより，静的ルーティングの利点であるパケットのインオーダ配送を可能としつつ，一部のワークロードにおいては ANCA を越える性能を得ることが可能となる．例として，2-ary 3-tree において各リンクを通る可能性のある送信先ノード番号を示したものが図 3.1 である．送信元ノードから共通祖先ルータまで辿っていく段階では，同じステージにあるリンクの間で完全に負荷分散がなされている．一方，共通祖先ルータから送信先ノードまで辿っていく段階では，全体で見ても一つのパスしか存在しない．このため，異なるノードに送信されているパケット同士の衝突が発生しないようになっている．

3.2 *k*-ary *n*-tree Dragonfly

Dragonfly[6] はルータ，グループ内ネットワーク及びグループ間ネットワークの 3 つのレベルからなる階層的ネットワークである．グループ内ネットワーク及びグループ間ネットワークのトポロジにはある程度の任意性があり，[6] においては，グループ内ネットワークとして flattened butterfly[5] を用いている．このグループ内ネットワークをマルチステージネットワークにすると，上位レベルのルータやリンクを簡潔にしホップ数を減らしたものである．本論文においては *k*-ary *n*-tree をグループ内ネットワークとして選択した．1 グループをチップ 1 枚に配置し，グループ間ネットワークはグループ間を単純に接続するだけとした．図 3.2 にグループ間接続のアウトラインを示す．

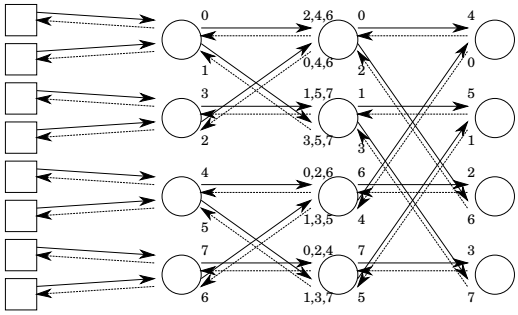


図 3 2-ary 3-tree における静的ルーティング [2]

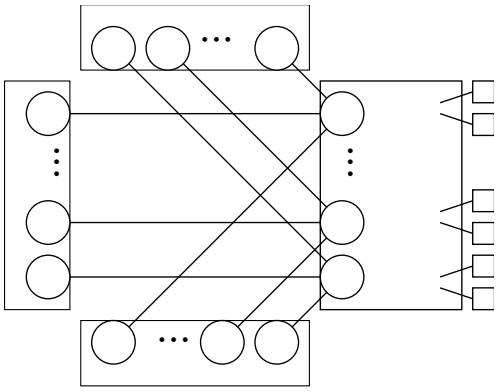
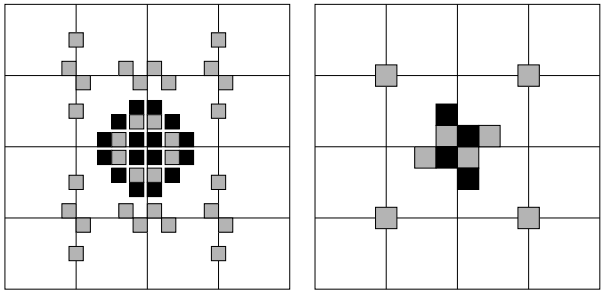


図 4 *k*-ary *n*-tree Dragonfly の概念図

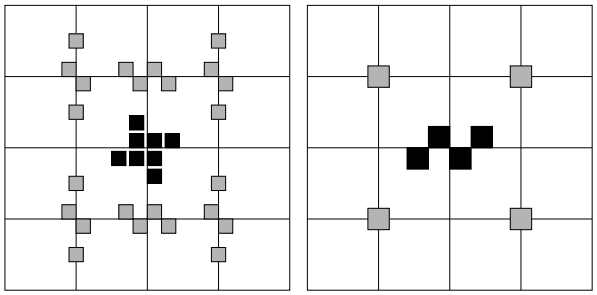
Dragonfly の適応ルーティングとして [6] では，Universal Globally Adaptive Load-balanced (UGAL) routing [11] をベースとしたものが用いられる．Dragonfly における UGAL は，送信元グループと送信先グループの間にランダムに選択した中間グループを挟むルーティングである．最短ルーティングと UGAL との選択には，ローカルのキュー情報を用いるもの (UGAL-L) と全てのグローバルチャンネルのキュー情報を用いるもの (UGAL-G) がある．

k-ary *n*-tree をグループ内トポロジとする本論文の Dragonfly における静的ルーティングのアルゴリズムは以下のようになっている．ルーティングに循環した依存関係が生じないため，このアルゴリズムはデッドロックフリーであり，同時にライブロックフリーである．

```
if 現在のルータが送信先ノードの
       属するグループに属している then
   k−ary n−treeと同様
else
   if 現在のルータが最上位レベル then
      グループ間リンクを通って
      送信先ノードの属するグループへ
   else
      k−ary n−treeと同様
   end if
end if
```



(a) 2-ary 6-tree (b) 4-ary 3-tree



(c) 2-ary 4-tree Dragonfly (d) 4-ary 2-tree Dragonfly

図 5 1 層あたりフロアプラン

3.3 フロアプラン

4 層 16 ノードの 64 ノードとし，ノードは 4x4 のタイル状に配置する．リンクはタイルの境界を通し，ルータもリンクに接続するために境界に面するように配置するものとする．今回評価するトポロジはすべてツリー構造に近い構造となっているため，ルータ及びリンクは H-tree によって配置することが可能である．本論文においては手動でフロアプランニングを行う．図 5(a) から図 5(d) は，今回評価を行う対象に関してフロアプランを行った結果を示している．色が少し濃くなっている矩形がルータを示し，黒いルータは他の層のルータと TSV で接続されているリンクを持つことを示している．

4. 評価

前述の 6 つの構成に対し，ワークロードとして合成トラフィックを用いたフリットベースシミュレーションによるパフォーマンス評価及び，ルータとリンクによる面積及び消費電力の静的な推定を行う．本章では，それぞれの目的及び評価の各種パラメータや環境についての詳細を述べる．

4.1 パフォーマンス評価

高次数ルータを用いたマルチステージ間接ネットワークを用いる利点として，少ないルータを用いてより多くのノードを低ホップ数，すなわち低レイテンシで接続することが可能であることが挙げられる．これは，NoC のトポロジとして，実装が容易なメッシュをはじめとした直接ネットワークではなく間接ネットワークを選択する理由として重要なものである．それゆえ，本論文ではパフォーマンスの評価基準としてレイテンシを用いる．

実験方法としては，ノードあたりパケット流入率に対するレイテンシの変化を測定する．測定には，cycle-accurate シミュレータである booksim[1] をベースとしてトポロジおよびルーティングアルゴリズムの追加を行ったものを用いる．ルータのモデルにはパイプライン化された投機的ルータ [9] を用いる．本論文ではタイル状に配置されたノードの一辺の長さのリンクを 1 サイクルで通過可能とする．それ以上の長さのリンクに関しては，長さが一辺以下の長さになるような数のリピータを挿入しその分レイテンシが大きくなると想定する．また，TSV を用いた垂直方向のリンクの長さは長くとも32.00 µm と，水平方向のリンクの長さ単位である1.5 mm に比べて十分短い [12] ため，垂直方向リンクによるレイテンシへの影響は考慮しない．これらの値は [12] において想定されている値である．表 2 に想定したパラメータをまとめる．

以上の想定の元，uniform random，transpose，bit complement の 3 種の合成トラフィックパターンを用いて評価を行う．パケットの送信元ノード番号を Src，送信先ノード番号を Dst とした際の送信先を決定する計算式を表 3 に示す．なお，その際は 1 パケット 1 フリットと仮定する．

ノード数	1 層あたりノード数	ルータアーキテクチャ
64	16	3 ステージ [9]

VC 数	入力バッファ数	出力バッファ数	フリット幅
2	16 flit	0 flit	128 bit

表 2 想定パラメータ

トラフィックパターン	送信先決定式
uniform random	Dst = Random((0, *# of nodes*))
transpose	$\mathrm{Dst}_x = \mathrm{Src}_y$, $\mathrm{Dst}_y = \mathrm{Src}_x$
bit complement	Dst = -Src **and** (*# of nodes* - 1)

表 3 トラフィックパターン

4.2 面積・消費電力

チップ上に構築される NoC において，面積や電力のようなリソースの消費量を考慮することは重要である．本論文においては，ルータの面積及び，パケットが流れていない状態の消費電力を計算する．値の算出には Orion 2.0[3] によって得られた推測値を用いる．Orion 2.0 はルータとリンクの面積と消費電力を算出することが可能である．ルータに関しては仮想チャネルアロケータ，スイッチアロケータ，クロスバ，バッファに消費される面積が算出できる．また，消費電力においてはそれらの消費電力にクロック分配に必要な電力を加えたものを算出することが可能である．本論文においては，ルータに対するフリットの流入率を 0%とすることで，パケットが流れていない状態の消費電力を静的に算出する．その際，リンクに関してはパケットが流れない場合には電力を消費しないものとする．その

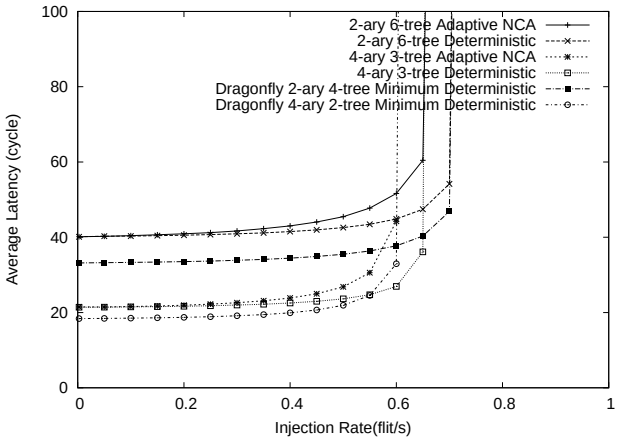


図 6 uniform random における平均レイテンシ

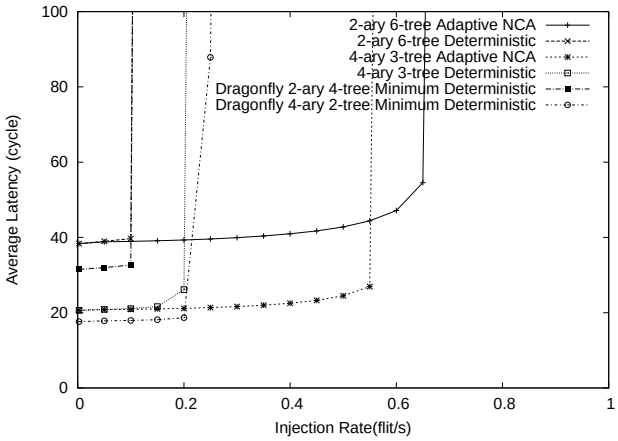


図 7 transpose における平均レイテンシ

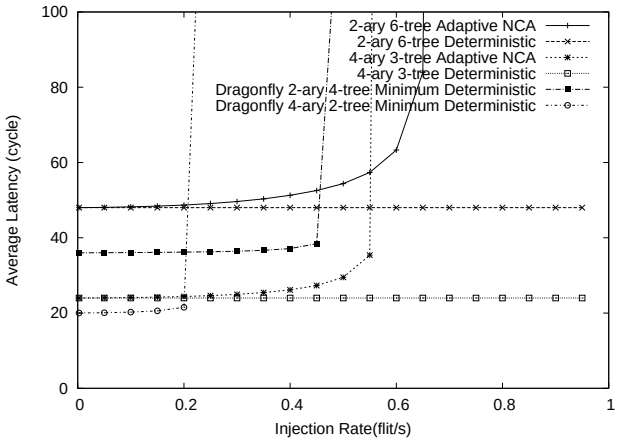


図 8 bit complement における平均レイテンシ

際のパラメータは 4.1 で表 2 に示した通りである．

5. 結果・考察

図 6 から図 8 はそれぞれ，uniform random トラフィック，transpose トラフィック，bit complement トラフィックにおいて，パケットが送信元ノードによってネットワークに注入されてから送信先ノードによってネットワークから取り除かれるまでに要したサイクル数の遷移を示している．X 軸は各サイクルにおいて各ノードでパケットが注入される確率を，Y 軸はその流入率における平均レイテンシをそれぞれ示している．グラフが急激に立ち上がった流入率がネットワークの飽和が発生した点である．言い換えると，スループットが頭打ちになる点である．2 つのネットワーク構成を比較した際，常に一方のグラフが他方のグラフの右下にある場合には，右下側のトポロジが明確に優れていると言える．

全体的な傾向としては，ネットワークの利用率が低いうちは Dragonfly の方が低いレイテンシを示している．しかし，Dragonfly は k-ary n-tree に比べて早く飽和し，レイテンシが大きくなってしまっている．また，bit complement における静的ルーティング k-ary n-tree はどのパケット流入率でも一定のレイテンシを示している．この性質は k や n の値に依存しないと推測される．なぜならば，各ステージのリンクの数が同時にネットワークに注入されうるパケット数以上であり，特に祖先ルータへのパスで衝突が発生しないからである．

適応ルーティングと静的ルーティングでの比較においても，明確な違いが見て取れる．ANCA は 3 種類全てのトラフィックパターンにおいてほぼ同じレイテンシを示しているのに対し，静的ルーティングはパターンによって極端な性能差が生じていることがわかる．

次に，図 9 は各トポロジ構成においてネットワークを構成するルータによって占有される面積の総和を示している．下から順にバッファ，クロスバ，仮想チャネルアロケータ，スイッチアロケータがそれぞれ占有する面積である．ルータの次数が大きくなった際，最も大きく影響を受けるのはクロスバの面積であることがわかる．異なるトポロジ種の間で比較してみると，Dragonfly は同じアリティの k-ary n-tree と比較し，面積の観点ではそれほど減っていないかむしろ増加する場合もあることがわかる．増加しているのは 2-ary の場合であり，ルータ数の減少よりもグループ間を接続しているルータの次数が増加した方が面積に与える影響が大きいことを示している．4-ary の場合はグループ

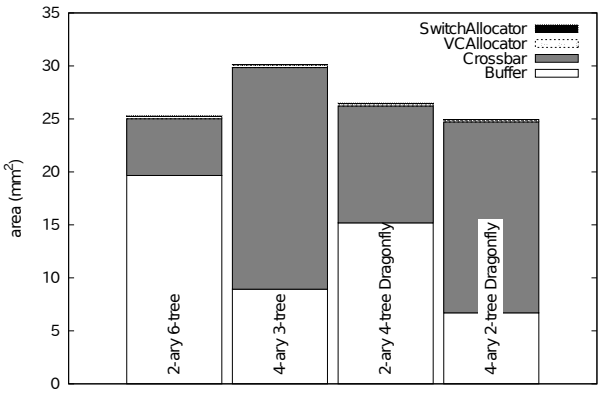


図 9 ルータ面積

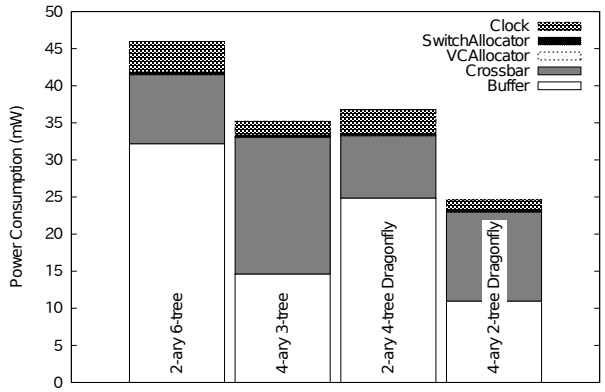


図 10 パケット流入が無い状態のネットワークの消費電力

間を接続するルータの次数は減少しており，それがグラフにも現れている．

最後に，図 10 は各トポロジ構成においてパケットが流れていない状態の消費電力を示している．下から順にバッファ，クロスバ，仮想チャネルアロケータ，スイッチアロケータ，クロックがそれぞれ消費する電力である．実際にパケットが流れるとアロケータが動作するようになり，このグラフとは異なった結果が得られると予想されるが，待機電力を減らすという観点から図 10 のようなデータは計測が必要である．面積における結果とは異なり，消費電力に関しては Dragonfly にすることによって k-ary n-tree から単純に減少していることがわかる．この結果はクロスバやアロケータに比べ，バッファの方が待機電力に与える影響が大きいということを示している．また，2-ary においてはクロックの消費電力が全体に対して占める割合が多いことから，ルータの数がクロック分配に要する待機電力に比較的大きな影響を与えることがわかる．

6. おわりに

本論文では，k-ary n-tree 及び，それをグループ間トポロジとして利用した Dragonfly についての合成トラフィックによるパフォーマンス評価を行い，さらに面積消費・電力消費に関する評価を行った．ネットワークに対するパケット量が少ないときには，Dragonfly が k-ary n-tree に対して良い性能を示した．これは Dragonfly の持つ，いくらかのステージが省略されているという性質によるものである．一方，多くのパケットが存在しているときには，逆に k-ary n-tree が Dragonfly に対して良い性能を示した．これは Dragonfly は k-ary n-tree の上位レベルのルータとリンクをより少ないリンクで置き換えた，と見ることが可能であることからも推察できる通りである．

また，ルーティング間の比較により，少なくとも k-ary n-tree においては，静的ルーティングと適用ルーティングの選択は典型トラフィックパターンによって決定されるべきであるということもわかった．トラフィックに特定のパターンが存在していない場合，均一トラフィックとみなすことができる．即ちそのような場合には，図 6 が示すように静的ルーティングを用いると性能向上が得られる可能性がある．一方，特定のトラフィックパターンが支配的である場合には，そのトラフィックによって得られるパスが重なってしまうと性能の低下が大きくなってしまう．

今後の課題としては，資源評価に関してより正確に行う必要がある点が挙げられる．本論文では，ルータのみに対して，パケットがネットワークを流れない状態，すなわち待機電力のみを消費電力の指標としたが，ワークロードを実行した際のリンクまで含めた消費電力についても評価する必要がある．

参考文献

[1] W.J. Dally and B. Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[2] C. Gomez, F. Gilabert, M.E. Gomez, P. Lopez, and J. Duato. Deterministic versus adaptive routing in fat-trees. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–8. IEEE, 2007.

[3] A.B. Kahng, B. Li, L.S. Peh, and K. Samadi. Orion 2.0: A power-area simulator for interconnection networks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, (99):1–5, 2012.

[4] S. Kannan and G.S. Rose. A hierarchical 3-d floorplanning algorithm for many-core cmp networks. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 1211–1214. IEEE, 2011.

[5] J. Kim, W.J. Dally, and D. Abts. Flattened butterfly: a cost-efficient topology for high-radix networks. *COMPUTER ARCHITECTURE NEWS*, 35(2):126, 2007.

[6] John Kim, Wiliam J. Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ISCA '08, pages 77–88, Washington, DC, USA, 2008. IEEE Computer Society.

[7] Charles E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Trans. Comput.*, 34(10):892–901, October 1985.

[8] Sabine R. Öhring, Maximilian Ibel, Sajal K. Das, and Mohan J. Kumar. On generalized fat trees. In *Proceedings of the 9th International Symposium on Parallel Processing*, IPPS '95, pages 37–, Washington, DC, USA, 1995. IEEE Computer Society.

[9] L.S. Peh and W.J. Dally. A delay model and speculative architecture for pipelined routers. In *High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on*, pages 255–266. IEEE, 2001.

[10] Sven-Arne Reinemo. Topologies: Fat-trees and dragonflies - a perspective on topologies, 2012.

[11] A. Singh. *Load-balanced routing in interconnection networks*. PhD thesis, Stanford University, 2005.

[12] A. Zia, S. Kannan, H. Jonathan Chao, and G.S. Rose. 3d noc for many-core processors. *Microelectronics Journal*, 2011.