相互結合網 RDT における adaptive routing

上樂 明也　　舟橋 啓　　西村 克信　　天野 英晴

慶応義塾大学理工学部
〒 223 横浜市港北区日吉 3-14-1
Tel:045-560-1063 Fax:045-560-1064
email: jouraku@aa.cs.keio.ac.jp

相互結合網 RDT は再帰構造を持つ Torus の重なりから構成されており、超並列計算機のプロセッサ間接続を行うために優れた性質を多く持っている。本研究では、Duato による必要十分条件を用いた方法と Turn モデルを用いた方法の 2 つのアプローチにより RDT 上でデッドロックフリーな adaptive routing を実現する方法を示す。Duato による必要十分条件を用いたルーティングはネットワーク中の利用されていないチャネルを最大限に活用することが出来るため高い通過率を実現することが出来、Turn model を用いたルーティングは、ランクの使用順が自由、最短経路を通る必要がない、各次元内で逆方向に方向転換が可能であるなど高い自由度を実現することが出来る。本研究では更に、シミュレーションによる性能評価を行った。

相互結合網, 適応型ルーティング, デッドロック回避, RDT

Adaptive routing on the Recursive Diagonal Torus

A. Jouraku　　A. Funahashi　　K. Nishimura　　H. Amano

Keio University
3-14-1 Hiyoshi, Kouhoku, Yokohama, Japan 223
Tel:+81-045-560-1063 Fax:+81-045-560-1064
email: jouraku@aa.cs.keio.ac.jp

Recursive Diagonal Torus or RDT consisting of recursively structured tori is an interconnection network for massively parallel computers. In this paper, we proposed two adaptive routing algorithms on the RDT. By using Duato's necessary and sufficient condition, we proposed an adaptive rouing algorithm on the RDT. Since channels are used efficiently with this algorithm, performance can be improved. We also proposed a new adaptiave routing algorithm by using Turn model. By using this algorithm, ranks can be used in the free turn and it doesn't have to use the minimal path. It can exploit higher flexibility than the above algorithm since it can use vectors in the reverse direction. In this paper, we also evaluated the behavior of the adaptive algorithm by simulatiuon.

Interconnection Network, Adaptive Routing, Deadlock Avoidance, RDT

1 はじめに

RDT(Recursive Diagonal Torus)[7]は, 比較的小さいdegreeで小さな直径と高いランダム転送能力を実現する相互結合網であり, 文部省重点領域研究で開発が進んでいる超並列計算機テストベッド JUMP-1[5] 用にルータチップが開発されている [10].

RDTは, ツリー, ハイパーキューブのエミュレーションが容易である等, 超並列網として優れた特徴を持つ一方, メッシュを内蔵していることから現在のメッシュ/トーラス結合並列計算機上で開発されたアルゴリズムの移植が容易である. さらに, ルートを複数持つ階層構造を利用して分散共有メモリを実現する方法の検討が行なわれている [9][13].

RDTのルーティングは, 出発地から目的地までの経路のベクトルを分解するベクトルルーティングを基本とし[7], e-cubeルーティング [1] を利用して, デッドロックを回避する方法が提案され [8], 実際にルータチップでも使われている. しかし, この方法は, デッドロックを防ぐために, ランクを使用する順序およびベクトルを適用する順序が決っているため, 効率が悪い上, 混雑や故障の迂回を行なうことができない. 一方, 経路の選択に自由度を持つルーティング法として Floating Vector Routing[8] も提案されているが, この方法ではデッドロックの可能性がある.

そこで, 本研究では, 近年, 急速に研究が進んだ adaptive routing を RDT に適用し, 効率が良く, 故障や混雑の迂回が可能なルーティング法を提案, 評価する.

2 RDTの構成

RDTの定義と基本的なルーティング法について, 本論文での議論に必要な範囲を以下にまとめる. 詳細な定義については [12] を参照されたい.

RDTは, 二次元トーラスを基本とする. 簡単のため基本トーラスは $N_{x0} \times N_{y0}$ ($N_{x0} = N_{y0} = c(2n)^{\gamma}$, c, n, γ は正の整数) であるとし以下のように番号付けられているとする.

定義 1 : 基本トーラス

(0,0)	(1,0)	(2,0)	$\cdots$	(N_{x0}-1,0)
(0,1)	(1,1)	(2,1)	$\cdots$	(N_{x0}-1,1)
(0,2)	(1,2)	(2,2)	$\cdots$	(N_{x0}-1,2)
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
(0,N_{y0}-1)	(1,N_{y0}-1)	(2,N_{y0}-1)	$\cdots$	(N_{x0}-1,N_{y0}-1)

上に示すノードの二次元アレイ上で, ノード (x, y) が四方の隣接ノード *(node (mod($x \pm 1, N_{x0}$), y) (x, mod* $(y \pm 1, N_{y0})$ *))* との間に結合リンクを持つ構造を基本トーラスまたはランク *0* トーラスと呼ぶ. □

一般に, トーラス構造に対してバイパスリンクを定める場合, 最も効果的なのは対角線方向である. 今, 各ノード (x, y) が, $(x \pm n, y \pm n)$ と結ぶ 4 本の付加リンクを持つとすると, この付加リンクは新たなトーラス状の結合網を形成する. このトーラス状結合網は, 基本トーラスに対し 45 度傾き, グリッドサイズは$\sqrt{2}$n 倍になっている.

このトーラス状結合網をランク 1 トーラスと呼び, 正の整数 n を基数と呼ぶ. さらに, ランク 1 トーラス上に同様な方法で付加リンクを付け加え, ランク 2 トーラスを形成する. 以上の操作を再帰的に繰り返し, 次々とランクの高いトーラスを形成していく.

我々の提案した RDT(Recursive Diagonal Torus) は, 以上の方法により再帰的に構成したトーラス状結合網の組合せにより構成される. ここで, 偶数ランクのトーラスが基本トーラスと同様の 2 次元正方隣接格子になるのに対し, 奇数ランクのトーラスは縦横比が 2:1 で循環ループが螺旋状になる. 一般的に, トーラスは, 基本トーラス同様の正方隣接格子で単純な循環構造を持つ結合網を指すが, ここでは, 螺旋状の循環ループを持つ奇数ランクの結合網も一括して定義し, 共に区別なく「トーラス」と呼ぶことにする.

定義 1に示した基本トーラスに対し, 再帰的に上位トーラスを形成することにより, RDT (Recursive Diagonal Torus) を定義する.

定義 2 : RDT
ランク *0* トーラス *(基本トーラス)* 上に定義 *2* に基づき再帰的にトーラスを形成したとき, 各ノードが, 基本トーラス及び基本トーラスをランク *0* として順に形成した他の *m* 個の上位トーラスを形成するためのリンクを持つ結合網を **RDT(n,R,m)** と呼ぶ. ここで, *n* は基数, *R* は最大ランク数である. また, *m* を多重度と呼ぶ. □

以上の定義では, 各ノードは他のノードと違ったランクの上位トーラスを持つことができる点に注意されたい. 以下, ノードがあるトーラスを形成するために必要なリンクを持つことを, 「トーラスを持つ」と表現する. RDT(n,R,m) は基本トーラスと m 個の上位ランクのトーラスを持つため, degree は 4(m+1) となる.

実際に 1 万ノード程度のサイズを念頭においた場合 , n=2 , m=1 に取るのが有利であり, 現在,JUMP-1 用に図 1に示す構成の RDT, RDT(2,4,1)/αについて検討を進めている.

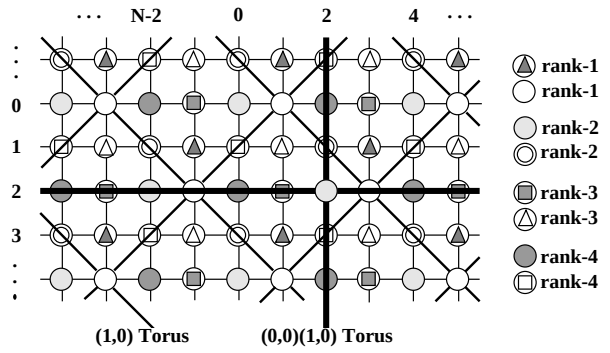

図 1: RDT(2,4,1)/αの構成

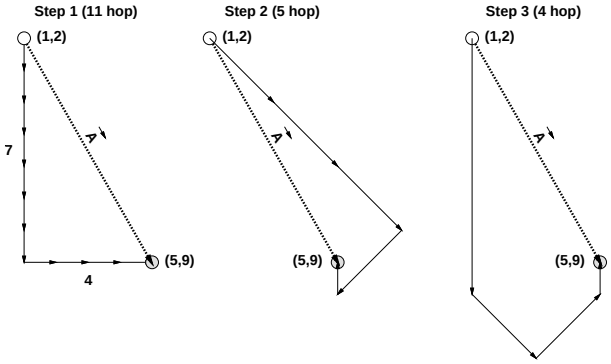

図 2: ベクトル ルーティングの例

n=2 の場合, あるランク上には独立なトーラスが 8 つ形成される. RDT(2,4,1)/αでは, 基本トーラス上に形成される 8 つのランク 1 トーラスのうち, 2 つをそのままランク 1 トーラスを形成するのに用い, ランク 2, 3, 4 を形成するために, それぞれ 2 つずつを用いる. すなわち, ランク 1 から 4 まで各ランクのトーラスを持つノードの数は等しい.

この構成では図 1に示すように, あるノードが持っていない上位トーラスは隣接ノードのどれかが持っている. すなわち, 基本トーラス (ランク 0-トーラス) 上の 1 ステップの移動で全ての上位トーラスが利用可能である.

3 RDT 上でのルーティングアルゴリズム

3.1 ベクトルルーティング

ベクトルルーティングは, 出発値から目的値までの経路を, 各ランクのトーラスの一辺を単位ベクトルとするベクトルの合成によって表現することにより, 使用するトーラスのランクとリンクの方向を求める方法である.

ここでは紙面の都合により直観的なモデルのみを示すが, 詳細な定義については [12] を参照されたい.

例として, (1,2) から (5,9) へのルーティングのベクトル分解を $n = 2$ とした場合について図 2に示す.

図 2の Step 1 では (1,2) から (5,9) までの $\vec{A}$をランク 0 の単位ベクトルであらわしている. 順に Step 2, Step3 とベクトル分解を繰り返すと,$\vec{A}$は Step 1 では 11 hop かかっていたルーティングが 4 hop で可能となる.

ベクトルルーティング は操作が簡単で, 利用するベクトルの順番を変えることにより, 代替経路が得られるため, 混雑や故障箇所の迂回が可能である. しかし, ベクトルルーティング は deterministic routing のため, パケットがノードを出発するときに既にルーティングは決定している. そのため, 各ノードが結合網中の全ての状況を把握していなければならない.

3.2 RDT 上での adaptive routing

deterministic routing に対し ,adaptive routing は, 経路を動的に選んでルーティングするため, 混雑や故障を発見したときにその場で回避することが可能である. また, 空いている経路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, 局所的な混雑を回避する点においても有効である. 耐故障性の点でも, 各ノードが送信しようとした経路が故障していたら, その場で他の経路を選択できるため deterministic routing に対し有利である.

しかし, たとえ adaptive routing が行えたとしても, デッドロックを起こす可能性がある. そこで ,2 つのアプローチから RDT 上でのデッドロックフリーな adaptive routing を提案した [11]. 基本的にはベクトルルーティングで求めたベクトルの使用する順序を変更することにより代替経路を求めるが, 最短経路を使用する必要がない場合は, 新たにベクトルを利用していくことも可能である.

4 Duato's protocol

4.1 k-ary n-cube

今まで提案されてきたデッドロックフリーな adaptive routing は, 何らかの方法で循環を断ち切ることによって, デッドロックを起こさない様にしてきた. しかし Duato は, 循環を含む経路に対してもデッドロックを起こさない adaptive routing の必要十分条件を示した [2].

Duato の条件は,

a. 結合網全体に渡る循環の無い逃げ道 (escape path) を用意する

b. 逃げ道と, 逃げ道により循環が切断されてデッドロックが起きなくなる他の経路によって結合網中のどのノード間でもパケットが送れるようにする

の 2 点を満足できれば a.,b. の経路を adaptive routing で選択することにより, デッドロックしない adaptive routing が可能になる.

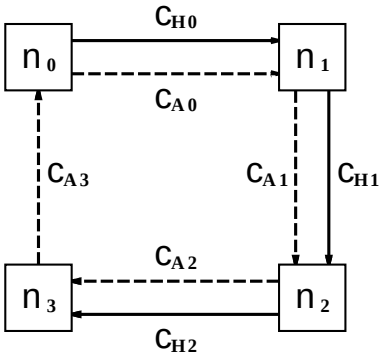

図 3: デッドロックフリーの証明

Duato は, 以下の手順により,k-ary n-cube での adaptive routing を実現した.

1. はじめに単方向リング (図 3) のネットワークについて考える. パケットが存在しているノードより目的地のノード番号が大きい場合は, チャネル C_H を使い, 低い場合は C_A を使用することで escape path が保証される.escape path が全てのノード間に存在するため, このネットワークはデッドロックフリーである.

2. 次に, ネットワークを双方向リングに拡張する. しかし,Duato の方法では, ルーティングは常に最短経路を通る (minimal routing) から, ルーティングの途中でそれまで使用していたリンクと逆方向のリンクを使用することはありえず, 逆方向のリンクは全く別のネットワークとして分離して考えることができる. よって, 双方向リングのネットワークでもデッドロックフリーである.

3. 次に, ネットワークを多次元のものに拡張する. e-cube routing[1] と同様に, 次元の使用順を固定すると, 結局は双方向 (単方向) リングを決まった順に使用するだけになる. よって, ルーティングのときに使用されるチャネル間の依存関係は各次元で独立であるため, デッドロックフリーである.

   以上により,escape path C_1 が用意された.

4. ここで, バーチャルチャネル $C_F(Fully\ \ adaptive)$ を新たに用意し, そのチャネルでは次元の使用順を無視して移動可能とする. これにより, デッドロックフリー な adaptive routing が可能となる.

ただしこのルーティングでは,2. を満足させるために, 常に最短経路を通るという,minimal routing を守らなければならない.

4.2 RDT

上に述べた k-ary n-cube での adaptive routing を RDT 用に拡張する.

1. k-ary n-cube での 3. までと同じ手順で escape path C_1 を用意する. この escape path は, 単一トーラスでのもので, ランク間の移動を想定していない.

2. 次に, ランクの使用順を固定する. 数字をランク, X or Y を次元とすると,
   $Xrank3 \rightarrow Yrank3 \rightarrow Xrank2 \rightarrow Yrank2 \rightarrow Xrank1 \rightarrow Yrank1$
   (ランク 3 の X 方向から順に Y 方向, ランク 2 の X 方向$\cdots$) という順に channel を使う. ランク間の移動には特別なチャネル C_B を使う. e-cube routing と同様の考え方により, このチャネル集合は, デッドロックフリーである. これにより RDT 用の escape path C_1 が各ランクに用意された.

3. 新たに用意されたバーチャルチャネルでは,
   $Xrank3 \rightarrow Yrank3 \rightarrow Xrank2 \cdots$
   の順を無視してチャネルを使用可能とする.

   バーチャルチャネル C_{Fn} を各ランクの C_1 ごとに用意する (図 4). さらに, 使用するランクの順を C_1 と同様に制限する.

   使用するランク順を固定することにより,e-cube routing と同様, このチャネル集合はデッドロックフリーとなる.

   ただし, ランクの使用順は
   $rank3 \rightarrow rank2 \rightarrow rank1 \cdots$と固定である.

C_{Fn}も C_1同様, 常に最短経路を通るという minimal routing を守らなければならない. また, 使用するランク順は固定となる. このルーティング法でルーティングが可能なベクトル集合と可能でないものを図 5に示す.

図 5(a) のベクトルはあるルーティングの例である. これにベクトルの入れ換えを行ったものが (b),(c) のベク

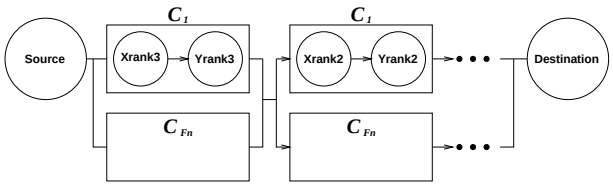

図 4: RDT でのチャネル集合

トルである. (c) のベクトルのような, ランクの使用順を自由に決めるようなルーティングは不可能である.

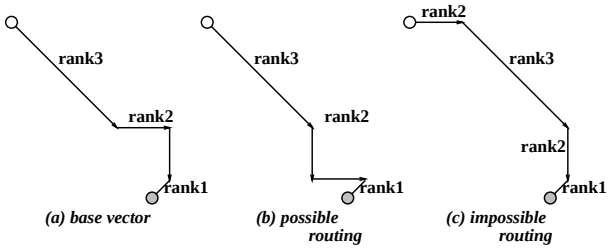

図 5: ルーティング例

5 Turn モデル

RDT(2,4,1)/αではノードが持っていない上位ランクを利用するためには, 基本トーラスを移動しなければならない. RDT(2,4,1)/αに Duato の条件を用いた方法を適用することは当然可能だが, ランクを決められた順番に使用しなければならない. そこで,Turn モデルによる方法を提案する.

5.1 2 次元メッシュでの Turn モデル

デッドロックが生じるのは, 結合網内のバッファが論理的に循環構造を作ってしまうためである. Glass らによる Turn モデル [4] は, 循環を生じないように, パケットがルーティング中に方向を変えるパターンを制限する方法である.

ここでもっとも単純な 2 次元メッシュでの Turn モデルを考える. この場合, 可能な循環構造を, 図 6(a) に示す. ここで,Turn モデルに従い, 各循環の進路変更を 1 つずつ禁止し, デッドロックを防いだ場合を図 6(b) に示す. このルーティング方法は north-last 法と呼ばれる.

5.2 RDT での Turn モデル

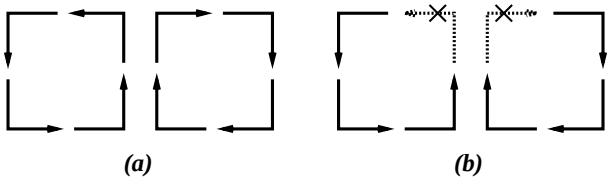

図 6: Turn モデル

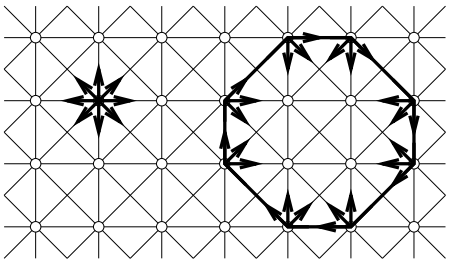

図 7: RDT での循環構造

先ほど述べた 2 次元メッシュでの Turn モデルを RDT でのモデルに拡張する. まず,RDT での可能な循環構造について考える.RDT では 8 方向に曲がることができるため, 図 7のような循環構造を示す.8 角形ではなく,3 角形,4 角形などの循環構造もとるが, 全てこの 8 角形の循環構造で示せるので省略する.

図 8: RDT での Turn モデル

次に, 図 6と同様に, 循環を生じないようにルーティングの方向を制限する .north-last 法を用いるので, 循環構造の右上 (もしくは左上) となりうるベクトルの使用を禁止する. その結果, 図 8で示すようなベクトルの集合となる.

以上の制限では不足している部分がある. すなわち, 循環構造は右上 (もしくは左上) の角がない場合にも生ずる (図 9). よって, 図 9で点線で示したベクトルの使用も禁止する. 以上により,RDT での循環構造は存在しなくなる. 新しく求めた RDT での Turn モデルを図 10に示す. このモデルは RDT(2,4,1)/αでも変更することなく利用することができる.

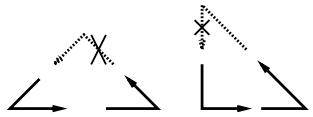

図 9: 特殊な場合の循環構造

図 10: RDT での Turn モデル 2

RDT(2,4,1)/αでは 2 次元メッシュと異なり, 図 1に示すようにランク間の移動がある. 例えば、ランク 1 の北東または北西方向のベクトルを使用した後にランク 3 の北東または北西方向のベクトルを使用する必要があるときには、ランク 0 のベクトルを利用して移動しなければならないために禁止されるターンを行わざるを得なくなる。このようなときにはチャネルを切替えればよいが、それが可能でない場合には一度プロセッサに落として再ルーティングを行う。一般的な 2 次元トーラスでの Turn モデルは, 図 6(b) を見てもわかるように, 曲がるパターンが 8 つあるうち,2 つを禁止する. つまり$\frac{1}{4}$の進路変更が使用不可となる. RDT での Turn モデルでは,24 パターン中 7 つを禁止するため, 2 次元トーラスのときより若干効率が悪くなる. しかし,Duato の必要十分条件を用いた方法では不可能だった RDT(2,4,1)/α でのランクの使用順を自由に決めれる点や, 同じ次元内で逆方向に進むことを可能にしており,adaptive routing の自由度としては高くなっている.

6 評価

ここでは, 提案された adaptive routing についてシミュレーションを実行し評価を取った.

シミュレータは C++で記述され, deterministic routing と adaptive routing の比較を行った. 以下, シミュレーションに用いた条件を示す.

シミュレーション条件

- パケット長: 8 flit
- 実行時間: 10000 clk(初めの 1000clk は評価せず)
- バーチャルチャネル数: 4 (CF, CH, CA, CB)
- ネットワークサイズ: 16 × 16
- パケットは 1 clk で 1 hop 移動
- トラフィックパターン
  - uniform traffic
  - nonuniform traffic (bit reversal)

6.1 Duato's protocol

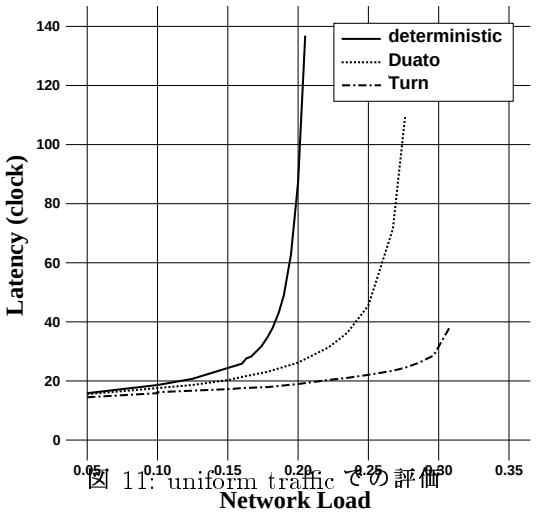

図 11: uniform traffic での評価

はじめに,uniform traffic での評価を図 11に示す. 横軸はネットワークの負荷, 縦軸はレイテンシを表している. ネットワークの負荷は, 全ノードが毎クロックに 1 flit 生成する場合を 1.00 としており, レイテンシは source node のプロセッサがパケットを生成してから, destination node のプロセッサがパケットの最後の flit を受け取るまでの時間を表す.

uniform traffic では, 各パケットの destination はランダムで決定されており, 等確率に分散されている.

deterministic routing でのレイテンシも比較のためのせてあるが, 負荷が低いときのレイテンシしか評価が取れなかったため, レイテンシが極端に高くなるときの負荷はわからなかった. 負荷が低いときには,Duato's protocol を用いたほうが若干性能が高い. しかし, 一般的に adaptive routing は負荷が高いときの方が deterministic

routing との性能の差が顕著になる. 今回はそのような評価は得ることが出来なかった.

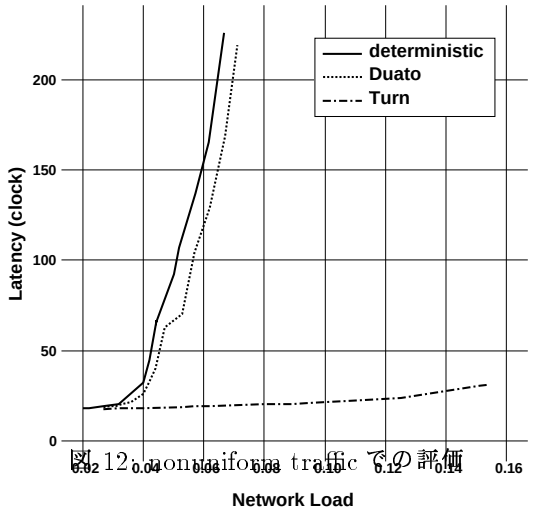

図 12: nonuniform traffic での評価

次に, nonuniform traffic での評価を図 12に示す. bit reversal は, ノード番号を 2 進数で表記し, $a_0, a_1, \cdots, a_{n-2}, a_{n-1}$ のノードは $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ のノードにパケットを送るようなトラフィックパターンのことである.

図 12に示すように, Duato's protocol の方が deterministic routing に比べて高い性能を示しており, ネットワークの飽和も deterministic routing に比べて高い負荷において発生する.

Duato's protocol と deterministic routing はどちらも最短経路しか通らないルーティングのため hop 数は基本的には変わらないが, Duato's protocol を用いたほうがバーチャルチャネルを効率良く利用するため, 高い性能を示す.

7 まとめ

adaptive routing を相互結合網 RDT に 2 つのアプローチから適用し, 新しいルーティング法を提案した.

Duato's protocol を用いて,RDT(2,4,1)/α用の adaptive routing を提案した. このルーティング法は, ネットワーク中の利用されていないチャネルを最大限に活用することが出来, 高い通過率を示すことが予想される. しかし, RDT(2,4,1)/αでは使用するランクの順番を変えることが出来ず, また元来最短経路を通ることしか許していないため,adaptive routing としての自由度は低い.

更に,Turn モデルを用いた adaptive routing を提案した. このルーティング法は, ランクの使用順は自由, 最短経路を通る必要がない, 各次元内で逆方向に方向転換が可能という点で Duato's protocol よりも高い自由度を示す. しかし,north-last 法を使用しているので, 北に進むのは最後にしなければならないといった制限はある.

提案されたルーティング法についてシミュレーションによる評価を行った. Duato's protocol は, nonuniform traffic のときは deterministic routing に比べて, 高い性能を示した. 一方,uniform traffic の場合は 負荷が低いときしか deterministic routing の評価が得られなかったため, それほどの差は見られなかった.

既に deterministic routing を用いた RDT(2,4,1)/αのルータチップは実装されており, 今後は adaptive routing を用いたチップの実装を想定したシミュレーションを行い, Duato's protocol, Turn model 両方のルーティングの評価を行う予定である.

参考文献

[1] W.J.Dally, C.L.Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. on Comput.*, vol. 36 no. 5, pp. 547-553, May. 1987.

[2] J. Duato , "A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks" In *Proc. of the International Conference on Parallel Processing.*, vol.I, pp.142-149, 1994.

[3] J. Duato , "A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks" In *IEEE Trans. on Parallel and Distributed Systems, Vol.6, No.10*, 1995.

[4] C.J.Glass and L.M.Ni, "Maximally Fully Adaptive Routing in 2D Meshes," In *Proc. of ISCA92.* pp.278-287, 1992.

[5] T.Tanaka, et, al. "The Massively Parallel Processing System JUMP-1," Ohmsha, 1996.

[6] P.Merlin, P.Schweitzer, Deadlock Avoidance in Store-and-Forward Networks-I : Store-and-Forward Deadlock, *IEEE Computer* Vol.28, No.3, pp.345-354, (1980).

[7] Y. Yang, H. Amano, H. Shibamura, and T.Sueyoshi. Recursive diagonal torus: An interconnection network for massively parallel computers. In *Proc. of IEEE SPDP*, 1993.

[8] Y. Yang, H. Amano, Message Transfer Algorithms on the RDT. In *IEICE Trans. on Information and Systems, Vol.E79-D, No.2*, 1996.

[9] T.Kudoh, et.al, Hierarchical bit-map directory schemes on RDT for a massively parallel processor JUMP-1 *Proc. of ICPP*, 1995.

[10] H.Nishi, K.Nishimura, K.Anjo, H.Amano, T.Kudoh, The JUMP-1 router chip *Proc. of International Phoenix conference on computers and communications*, 1996.

[11] A.Funahashi, T.Hanawa, T.Kudoh, H.Amano, "Adaptive routing on the Recursive Diagonal Torus." *Proc. of International Symposium on High Performance Computing* , pp.171-182 Nov. 1997

[12] 楊、天野、柴村、末吉、超並列計算機向き結合網 :RDT, 電子情報通信学会論文誌, *D-I Vol.J78-D-I*, 1995.

[13] 相互結合網 RDT 上で階層マルチキャストによるメモリコヒーレンシ維持手法, 情報処理学会論文誌, *37*巻 *7*号, 1996.