1 はじめに

近年，PC/WS の性能向上により Network of workstations(NOWs) [1],[2],[3],[4] などの高速なネットワークで接続された PC や WS で並列処理を行う研究が盛んに行なわれている．これらは同規模の並列計算機よりもコストが安い点が特徴で，将来に渡って高性能計算の主力マシンの一角を占めると予想される．

通常これらのネットワークには wiring flexibility，scalability の要求からトポロジフリーである irregular network が用いられる．irregular network はトポロジフリーのため，従来の並列計算機で用いられているメッシュやトーラスなどの固定トポロジに比べてデッドロックの回避が複雑なものとなっている．このため，irregular network におけるルーティングアルゴリズムは，デッドロック回避のために厳しいチャネル使用制限を課さざるを得なくなっている．

up*/down* routing[1] は，チャネル間の cyclic dependency を除去することによりデッドロックフリーを保証した irregular network における代表的なルーティングアルゴリズムであるが，効率的にネットワークのバンド幅を利用することができない場合が多い．これは，課せられたチャネル使用制限のために，非最短経路を通るパケットが多くなり無駄なトラフィックとレイテンシが増加する，ルートノード付近にトラフィックが偏りやすくなるためにホットスポットが発生しやすくなる，などの問題を本質的に抱えているからである．

そこで本論文では，up*/down* routing における後者の問題を緩和することを目的としたルーティングアルゴリズム Left-first turn routing (L-turn routing) を提案する．L-turn routing は，up*/down* routing と同様に，有向グラフの構築をベースとしているが，up*/down* routing とは対照的に，リンクに対する論理的な方向の割当てを spanning-tree の水平方向を基準として行なうことにより水平方向におけるルーティングの自由度を高めてルートノード付近におけるトラフィックの偏りを緩和することを目的としている．

以降，第 2 章では様々な irregular network の形態を隠蔽するモデルについて述べ，第 3 章では irregular network における代表的なルーティングアルゴリズムである up*/down* routing について述べる．そして，第 4 章では L-turn routing の提案を行い，第 5 章にてシミュレーション結果を提示，考察を行う．また，第 6 章で結論と今後の課題を述べる．

2 Network Model

irregular network のルーティングはプロセッシングエレメントと switch の接続，switch 同士の接続等，様々な接続パターンを考えなければならない．しかし，接続パターンに応じてそれぞれルーティングアルゴリズムを考えるのは得策ではない．

例えば，WS をネットワークで接続してクラスタコンピューティングを行う場合, 通常は各 WS はネットワークを構成する switch と接続される. この場合，パケットは目的地の WS に接続された switch まで到達できれば，後はデッドロックせずに目的地のプロセッサに到着することができる．よって，ルーティングアルゴリズムは switch 間のルーティングにのみ焦点を当て検討することができる．

switch 間の interconnection network I は corresponding graph G で以下のように表せる．

$$I = G(N, C)$$

ここで，N は switch の集合，C は switch 間の bidirectional link を表している．これにより，図 1 の irregular network は図 2 のように表すことができる．

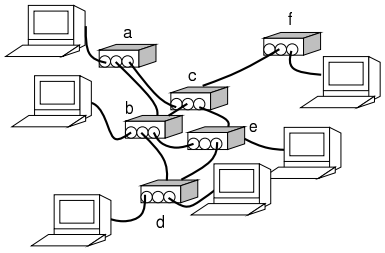

図 1: switch-based interconnection network

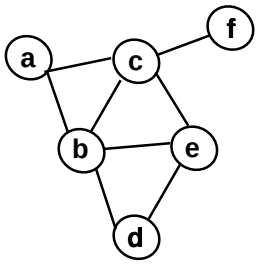

図 2: corresponding graph G

図 2.b のようにネットワークを一般化し，ルーティングアルゴリズムを検討することによりルーティングアルゴリズムは switch 間接続にとどまらず，直接網にも適用することができる．

3 Up*/Down* Routing

本章では irregular network における代表的なルーティングアルゴリズムである up*/down* routing について述べる．

up*/down* routing は irregular topology だけでなく regular topology にも適用可能な deadlock-free partially

adaptive routing であり，Autonet[1] や Myrinet[5] などのネットワークにおいて既に利用されている．

up*/down* routing は，全てのリンクに *up* または *down* の方向が割当てられた有向グラフを必要とするため，最初にその基礎となる spanning-tree を構築する必要がある．spanning-tree の構築は，breadth-first search (BFS) または depth-first search (DFS) のいずれかの方針に基づいて行なわれるが，ここでは Autonet で利用されている BFS に基づいた spanning-tree である propagation order spanning-tree (POST)[6] の概念をベースとした構築方法を簡単に説明する．この概念に基づいた spanning-tree の構築アルゴリズムは以下のようになる．

1. ネットワークの中から任意に spanning-tree の root node を選択する
2. root node は，全ての隣接ノードに接続要求メッセージを送信し，要求を受諾したノードを root node の子ノードとして spanning-tree に付け加える
3. あるノードの子ノードとなったノードは，隣接ノードに接続要求メッセージを送信し，要求を受諾したノード (既に spanning-tree に含まれているノードは要求を拒否する) を自身の子ノードとして spanning-tree に付け加える．
4. 全ノードが spanning-tree に含まれるまで 3 の作業を繰り返す

spanning-tree を構築した後，ネットワーク上の全てのリンクに対して以下に示す規則に基づいて up または down の方向を割当て，有向グラフを構築する．

まず，up 方向を次の 2 つの条件のいずれかを満たすリンクに対して割当てる．

1. 移動先のノードが移動元のノードよりもルートノードに近い
2. 移動先のノードと移動元のノードのルートノードからの深さが同一であり，移動先のノード ID の方が移動元のノード ID よりも小さい

次に，残りの全てのリンクに対して down 方向を割当てる．

以上の作業により，例として図 3 のような有向グラフが構築される．

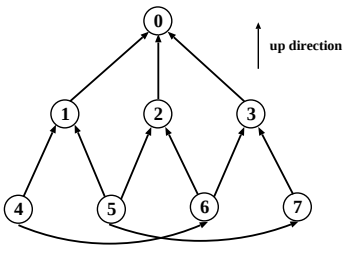

図 3: BFS spanning-tree に基づいた有向グラフ

deadlock-free と任意のノード間の経路を保証するために，全ての経路は必ず 0 回以上 up 方向に移動した後に 0 回以上 down 方向に移動する，という条件を守る必要がある．この条件により，down 方向から up 方向への移動を行なうことはできなくなるので，チャネル間の cyclic dependency が除去され deadlock-free が保証される．条件を守る限りは経路を自由に選択できるが，常に最短経路を選択することはできないので， up*/down* routing は partially adaptive routing となる．例えば，図 3 においてノード 5 からノード 0 へパケットを転送する場合には，ノード 1 またはノード 2 を経由してノード 0 まで到達することができるので全ての最短経路を選択することができるが，ノード 1 からノード 6 へパケットを転送する場合には，down 方向から up 方向への移動を含むノード 4 を経由する最短経路は選択することができず，ノード 0 とノード 2 またはノード 3 を経由する非最短経路しか選択することができない．

4 L-turn routing

up*/down* routing では，図 4 のように，spanning-tree を上下の方向から捉え，各リンクに対する論理的な方向を定めるための基準値をルートノードからの子孫方向の距離 (深さ) とし，隣接ノード間の基準値を比較することにより up または down の方向を割当てている．

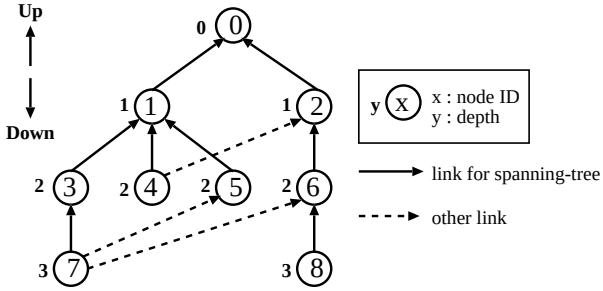

図 4: up*/down* routing における有向グラフ

しかし，up*/down* routing では down 方向から up 方向へのルーティングを禁止しているため，本質的にルートノード付近にトラフィックが偏りやすくなるという問題を抱えている．

そこで我々は，spanning-tree を左右の方向から捉え，論理的な方向を定めるための基準値をルートノードからの水平方向の距離 (幅) とすることにより，ツリーの水平方向におけるルーティングの自由度を高めてルートノード付近のトラフィックの偏りを解消することを目的とした L-turn routing の提案を行なう．

以降，まず L-turn routing の定義に必要となる左右の論理的な方向を持つ有向グラフ (L-R directed graph) を

構築する手順を説明し，その後 L-turn routing の定義を述べる．

4.1 L-R directed graph の作成

L-turn routing は，up*/down* routing と同様にその定義のために有向グラフの構築を必要とする．L-turn routing が用いる有向グラフである L-R directed graph は，up*/down* routing 向けの有向グラフとは対照的に左右の論理的な方向を持つ．

L-R directed graph の構築の手順は以下の通りである．

1. BFS spanning-tree の構築

第 3 章において説明した POST などの手法により BFS spanning-tree の構築を行なう．

2. 基準値の割当て

1 で構築した spanning-tree に対して前順走査を行ない，各ノードに対して走査を行なった順に図 5(図 4 と同じネットワーク) のように昇順で番号を割当てる．図 5 が示すように，この作業により各ノードに割当てられる番号は水平方向におけるルートノードからの距離を示す値となる．

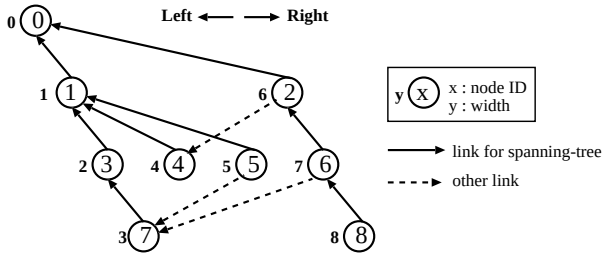

図 5: L-R directed graph

3. リンクに対する論理的な方向の割当て

各ノードに割当てた基準値を用いて全てのリンクに対して *left* または *right* の方向を割当てる．まず，以下の条件を満たすリンクに対して left 方向を割当てる．

- 移動先のノードの基準値が移動元のノードの基準値よりも小さい

L-R directed graph では，各ノードに割当てられる基準値が重複することがないので，up*/down* routing のようにタイブレークを考慮する必要が無い．

次に，それ以外のリンクに対して right 方向を割当てる．

以上の手順により，L-R directed graph が構築される．

4.2 L-turn routing algorithm

deadlock-free と任意のノード間の経路を保証するために L-turn routing は，以下の条件を守るものとする．

> 全ての経路は必ず 0 回以上 left 方向に移動した後に 0 回以上 right 方向に移動する

これにより，right 方向に移動した後に left 方向に移動することができなくなるので，up*/down* routing と同様にチャネル間の cyclic dependency が除去され deadlock-free が保証される．

図 5 に示すように，L-R directed graph において spanning-tree を構成する全てのリンクは，left 方向に進むとルートノードに近づき，right 方向に進むとルートノードから遠ざかるという特性を備えている (これは前順走査による基準値の割当てを行なっているためである)．この特性により，最初に left 方向のみに必要なだけ進み続けることにより必ず目的ノードを子孫に持つノードに到達することができ，その後は right 方向のみに進み続けることにより任意の目的ノードに到達することができる．従って L-turn routing では，up*/down* routing と同様に任意のノード間の経路が保証される．

次に，図 5 を用いて L-turn routing のルーティング例をいくつか示す．

まず，ノード 3 からノード 8 にパケットを転送する場合， L-turn routing では，ルートノード 0 を経由せずに，ノード 7 と 6 を経由する最短経路により到達することができる．一方，up*/down* routing では，down 方向から up 方向への移動が必要なために上記の最短経路を選択することができず，ルートノードを経由する非最短経路しか選択ができない．

次に，ノード 1 からノード 6 にパケットを転送する場合，L-turn routing では，ルートノードを経由する 1,0,2,6 の経路に加えて，ルートノードを経由しない 1,4,2,6 および 1,3,7,6 の経路も選択できる．これに対して up*/down* routing ではルートノードを経由する経路しか選択できない．

最後に，ノード 7 からノード 2 にパケットを転送する場合， up*/down* routing は，最短経路 7,6,2 を選択できるが，L-turn routing では right 方向から left 方向への移動が必要なために選択することはできず，7,3,1,0,2 もしくは 7,3,1,4,2 の非最短経路しか選択ができない．しかしそれでもルートノードを経由しない経路を選択することが可能となっている．

上記のように L-turn routing は partially adaptive routing となる．

5 評価

本章では，コンピュータシミュレーションによる L-turn routing の性能評価の結果を示す．

シミュレーションは，C++言語により実装されたフリットレベル相互結合網シミュレータを用いて行ない，比較のために up*/down* routing の評価を同時に行なった．

5.1 ネットワーク構成

ネットワークは，相互に接続されたスイッチおよびスイッチに接続されたプロセッシングエレメントの集合から構成されている．各スイッチは 6 つのポートを持ち，他のスイッチとの接続のために 4 つのポートを使用し，残りの 2 つのポートをプロセッシングエレメントとの接続に用いている．

シミュレーションは，16, 64 switches から構成された irregular topology と 2D トーラスに対して行なった．

irregular topology に関しては，付加リンク数を最大数および最大数の $\frac{3}{4}$ と変えた場合の評価を行ない，それぞれ 20 の異なるトポロジをランダムに生成して評価を行なった．

5.2 シミュレーション条件

実行したシミュレーションにおける条件を表 1 に示す．

表 1: シミュレーション条件

パケット長	128 flit
バーチャルチャネル数	1
ルーティング	1 clock
クロスバ間移動	1 clock
リンク間移動	1 clock
実行時間	2,000,000 clk
トラフィックパターン	uniform

シミュレーションにおける最初の 10,000 クロックについては，ネットワークの状態が安定していないと考えられるために評価の対象外としている．

評価を行なった up*/down* routing と L-turn routing は共に，選択可能な経路の中で最短となる経路のみを選択するようにしている．これは，無駄なトラフィックの増加によりパケットの衝突が増加して性能が低下することを防ぐためである．

5.3 Irregular network における評価

5.3.1 16 switches irregular network

まず，16 スイッチの irregular network において付加リンク数を 24 本 とした場合の結果を図 6 と図 7 に，32 本とした場合の結果を図 8 と図 9 に示す．図 6 と図 8 はランダムに生成した各トポロジのスループットを表し，図 7 と図 9 はそのスループット達成時におけるレイテンシを表している．

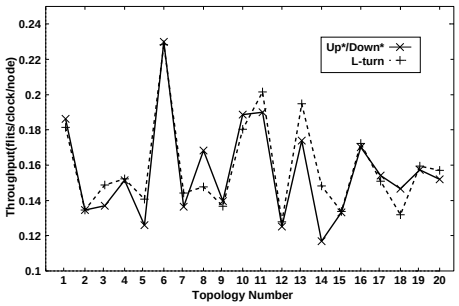

図 6: Throughput for 16 switches irregular networks with 24 link

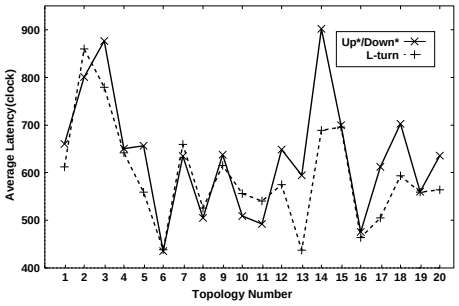

図 7: Average latency for 16 switches irregular networks with 24 link

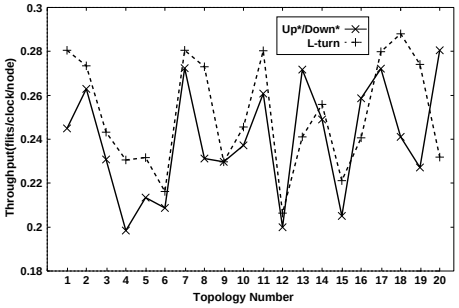

図 8: Throughput for 16 switches irregular networks with 32 link

図 6 と図 7 において，スループットおよびレイテンシの最大値は最小値の約 2 倍となっており，ネットワークの性能はトポロジの構造に大きく依存していることが分かる．図 8 と図 9 より，32 link の場合においても同様の傾向が表れていることが分かる．32 link の場合には，各トポロジ間の性能差は若干小さくなっているが，これはリンク数の増加に伴なってルーティングの自由度が高くなり全体的に効率の良いルーティングが可能となっているためと考えられる．

2 つのルーティングアルゴリズムの性能差もまた同様にトポロジの構造に依存している．例えば，図 6 における 8 番のトポロジでは，up*/down* routing のスループット

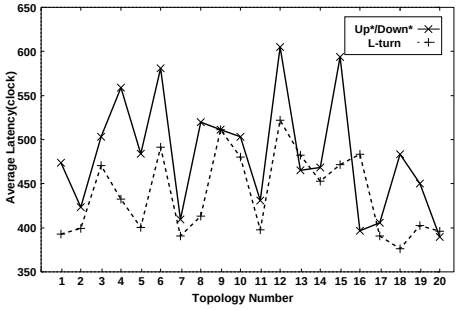

図 9: Average Latency for 16 switches irregular networks with 32 link

は L-turn routing よりも約 14% 高いが，14 番のトポロジではこれとは対照的に L-turn routing の方が約 26% 高くなっている．これは，禁止される経路の割合および分布がトポロジの構造に依って up*/down* routing と L-turn routing では異なるために性能に影響を与えるルーティングの自由度に大きな差が生じているためと考えられる．この傾向は 32 link の場合により顕著なものとなっているが，これはリンク数が増加したことにより各ルーティングアルゴリズムにおける禁止経路の割合と分布がトポロジの構造に依存する度合が大きくなっているためと考えられる．

以下に示す表 2 は，上の図で示したそれぞれのリンク数における 2 つのルーティングアルゴリズムのスループットとレイテンシの平均値をまとめたものである．表中の括弧内の値はそれぞれの標準偏差を表している．

表 2: 16 switches irregular networks における Throughput[flits/clock/node] と Latency[clock] の平均値

24 link	Avg. Throughput(SD)	Avg. Latency(SD)
Up*/Down*	0.1559(0.0445)	634.5(194.3)
L-turn	0.1586(0.0353)	593.5(145.5)
32 link	Avg. Throughput(SD)	Avg. Latency(SD)
Up*/Down*	0.2398(0.0463)	482.8(95.5)
L-turn	0.2511(0.0387)	437.9(71.2)

表 2 より，24 link の場合には L-turn routing の方が若干高いものの，平均的には 2 つのルーティングアルゴリズムの性能はほぼ同一であるといえる．しかし，標準偏差の値より L-turn routing の方が若干安定した性能を示しているといえる．

32 link の場合も同様であるが，24 link の場合よりも若干スループットの差が大きくなっている．

5.3.2 64 switches irregular network

次に，64 スイッチの irregular network において付加リンク数を 96 本 とした場合の結果を図 10 と図 11 に，128 本とした場合の結果を図 12 と図 13 に先と同様に示す．

これらの図より 64 スイッチのネットワークにおいても同様にネットワークの性能はトポロジの構造に大きく依

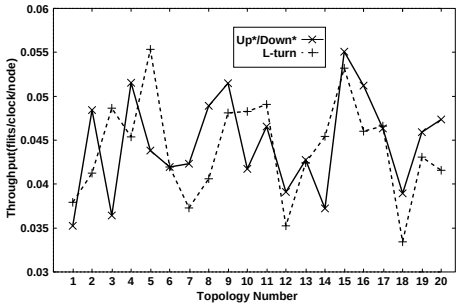

図 10: Throughput for 64 switches irregular networks with 96 link

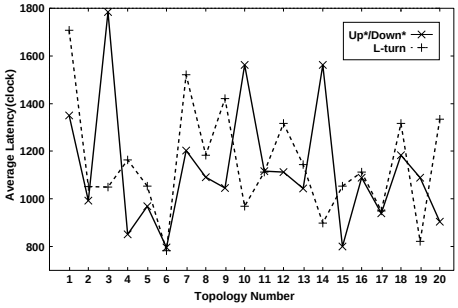

図 11: Average latency for 64 switches irregular networks with 96 link

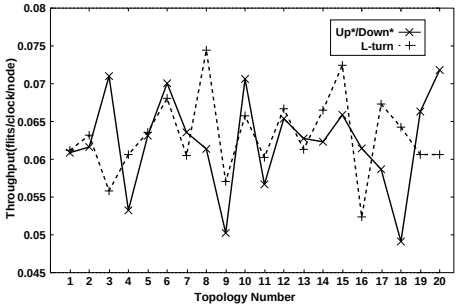

図 12: Throughput for 64 switches irregular networks with 128 link

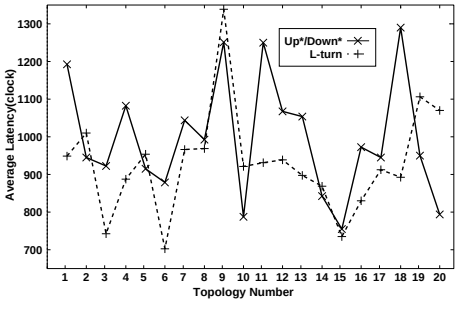

図 13: Average latency for 64 switches irregular networks with 128 link

存していることが分かる．しかし，2 つのルーティングアルゴリズムの性能差がトポロジの構造に依存する度合はより大きくなっている．これはリンク数が 16 switches のネットワークに比べて 4 倍増加しているためであると考えられる．

64 スイッチのネットワークにおける各ルーティングアルゴリズムのスループットとレイテンシの平均値を以下の表 3 に示す．

表 3: 64 switches irregular networks における Throughput[flits/clock/node] と Latency[clock] の平均値

96 link	Avg. Throughput(SD)	Avg. Latency(SD)
Up*/Down*	0.0446(0.0093)	1124.5(405.8)
L-turn	0.0440(0.0075)	1148.1(315.1)
128 link	Avg. Throughput(SD)	Avg. Latency(SD)
Up*/Down*	0.0623(0.0092)	996.7(245.5)
L-turn	0.0631(0.0067)	931.2(191.3)

表 3 より，96 link の場合には，up*/down* routing が，128 link の場合には，L-turn routing がそれぞれ若干他より優れているが，やはり平均的な性能はほぼ同等であるといえる．同様に．標準偏差の値より L-turn routing の方が若干安定した性能を示していることが分かる．

5.4 2D torus における評価

次に regular network である 2D torus におけるシミュレーション結果を図 14 と図 15 に示す．これらの図はそれぞれ 16, 64 スイッチにおけるスループットとレイテンシの関係を示している．

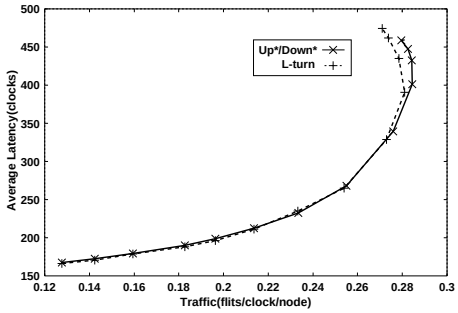

図 14: Average latency versus traffic for 16 switches 2D torus

まず，図 14 より 16 スイッチにおいては，若干の差はあるものの 2 つのルーティングアルゴリズムの性能はほぼ同等であるといえる．一方，64 スイッチの場合には図 15 より，スループットに関しては up*/down* routing の方が若干高いがレイテンシに関しては L-turn routing の方が低い値を維持している．

以下に示す表 4 は 2D torus における各ルーティングアルゴリズムにより選択可能な経路の総数をそれぞれ表し

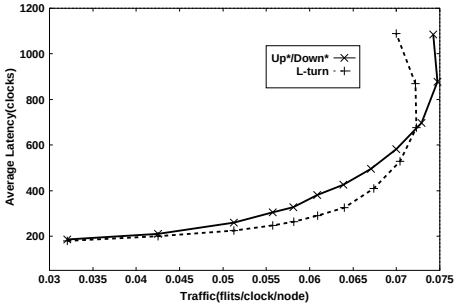

図 15: Average latency versus traffic for 64 switches 2D torus

たものであり，この値はルーティングの自由度に関する目安となる．

表 4: 各ルーティングアルゴリズムの 2D torus における経路の総数

16 switches	経路総数
Up*/Down*	520
L-turn	520
64 switches	経路総数
Up*/Down*	33568
L-turn	17482

この表よりまず，16 スイッチにおいては 2 つのルーティングアルゴリズムの間には経路数に関して差が無いことが分かる．これは図 14 の結果に適合しているといえる．これに対して 64 スイッチにおいては up*/down* routing の方が L-turn routing よりも 2 倍ほど高い値を示しており，ルーティングの自由度がより高いものであることが分かる．一般にルーティングの自由度が高いほど性能が高くなると言われているが，それにもかかわらず 2 つのルーティングの性能差が少ないのは，選択可能な経路がある部分 (ルートノード付近など) に集中してしまい非最短経路型である up*/down* routing ではかえってトラフィックの偏りがひどくなってしまっているためではないかと考えられる．2 つのルーティングアルゴリズムのスループットの差が小さく up*/down* routing のレイテンシが L-turn routing に比べて高くなっているのはこのことが原因となっているのではないかと考えられる．

6 結論

Irregular network におけるルーティングアルゴリズムである L-turn routing の提案と評価を行なった．L-turn routing は，既存の up*/down* routing と同様にチャネル間の cyclic dependency を無くすことにより deadlock-free を実現するが，リンクに対する論理的な方向の割当てを spanning-tree の水平方向を基準として行なうことにより水平方向におけるルーティングの自由度を高めてルートノード付近におけるトラフィックの偏りを緩和することを目的としている．

シミュレーションにより up*/down* routing と L-turn routing の評価を行なった結果，irregular network における両者の性能はトポロジの構造に大きく依存し，平均的にはほぼ同等であるが若干 L-turn routing の方が安定した性能を示すことが分かった．また，regular network である 2D torus においても両者の性能はほぼ同等であることが分かった．

今後は本研究で得られた結果を元に，トポロジの構造に性能が依存する度合を小さくするためのルーティングアルゴリズムの設計やトポロジに応じて最適なルーティングアルゴリズムを適応的に選択する手法などの検討および評価を行なっていく予定である．

参考文献

[1] M.D. Schroeder et al. Autonet: A high-speed, self-configuring local area network using point-to-point links. *Technical Report SRC research report 59,DEC*, Apr. 1990.

[2] R. W. Horst. Tnet: A reliable system area network. In *IEEE Micro*, Feb. 1995.

[3] F.Silla and J.Duato. Efficient Adaptive Routing in Networks of Workstations with Irregular Topology. In *proc. of CANPC'97*, Feb. 1997.

[4] 西 宏章, 多昌 鷹治, 工藤知宏, and 天野英晴. 仮想チャネルキャッシュを持つネットワークルータの構成と性能. In *JSPP'99* 論文集, 1999.

[5] N.J. Boden et al. Myrinet - A gigabit per second local area network. *IEEE Micro, vol. 15*, Feb. 1995.

[6] M.D. Schroeder T.L. Rodeheffer. Automatic reconfiguration in Autonet. *Technical Report SRC research report 77,DEC*, Sep. 1991.