Vol. 42 No. SIG 9(HPS 3) 情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム Aug. 2001

L-turn Routing：Irregular Network における Adaptive Routing

鯉渕道紘[†] 舟橋啓[††]
上樂明也[†] 天野英晴[†]

近年，Personal computer を高速なネットワークで結び，並列処理を行う研究がさかんに行われている．通常ネットワークは wiring flexibility，scalability の要求からトポロジに制限のない irregular network が用いられるが，既存の irregular network のルーティングアルゴリズムは，チャネル使用制限が厳しく，リンクを有効に使うことができない．そこで本稿では各リンクを効果的に使うことを念頭においた adaptive routing である Left-up-first turn routing（L-turn routing）を提案する．L-turn routing は irregular network を directed-graph に見立てる際に工夫を施すことにより，デッドロックを防ぐために課すパケットの転送禁止ターンの分散を実現する点が特徴であり，高いトラフィック分散能力を持つ．シミュレーション結果より，L-turn routing は様々なトポロジにおいて高い性能を示すことが分かった．

L-turn Routing: An Adaptive Routing in Irregular Networks

Michihiro Koibuchi,[†] Akira Funahashi,[††] Akiya Jouraku[†]
and Hideharu Amano[†]

Network-based parallel processing using commodity personal computers has been widely developed. Since such systems require high degree of flexibility and scalability of wiring, a high-speed network with an irregular topology is often needed. In traditional routing algorithms for irregular networks, available paths are considerably restricted in order to avoid deadlocks. In this paper, we propose a novel routing algorithm called Left-up-first turn routing (L-turn routing), which makes a better traffic balancing in irregular networks by building a specific directed-graph. Result of simulations shows that L-turn routing achieves better performance than traditional ones with each topology.

1. はじめに

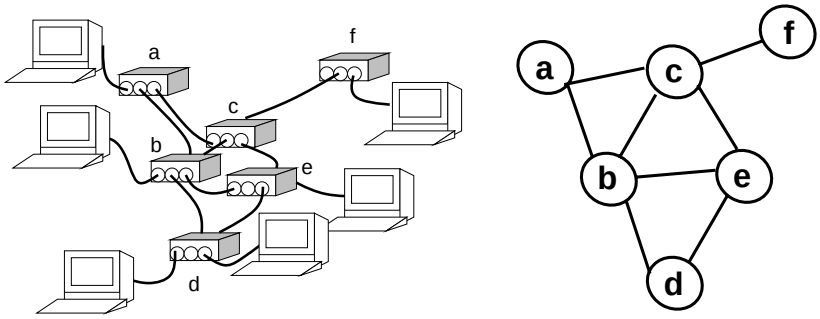
近年，Personal computer（PC）/Workstation（WS）の性能向上により高速なネットワークで接続された PC や WS を用いて並列処理を行う研究がさかんに行われている[1)~5)]．これらは同規模の並列計算機よりもコストの点で有利であり，将来にわたって高性能計算の主力マシンの一角を占めると予想される．

現在，これらのシステムは専用のクラスタが用いられることが多いが，高速なネットワークを用いて机上に配置された PC や WS を接続し，専用のクラスタシステムと同様の性能を実現することを可能にするシステムも提案されている[5)]．通常これらのネットワークには wiring flexibility，scalability の要求から irregular network が用いられるため，irregular network 上のルーティングアルゴリズムに関する研究がさかんに行われている[1),6)~8)]．しかし，irregular network はトポロジフリーのためルーティングアルゴリズムに対する要求が厳しく，従来の並列計算機で用いられている固定トポロジに比べて，性能上不利な点が多い[9)]．

また，irregularity がデッドロックの除去を困難なものにしており，既存のルーティングアルゴリズムはいずれもデッドロックフリーを実現するため，厳しいチャネル使用制限を課している．そのため有効に使えるリンクが限られてしまい，トラフィックが偏る問題が生じる．

そこで本稿では高いトラフィック分散能力を持つルーティングアルゴリズムである Left-up-first turn routing（L-turn routing）を提案する．L-turn routing は irregular network を directed-graph に見立てる際に

† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
†† 三重大学工学部
Faculty of Technology, Mie University



(a) Switch-based interconnection network (b) Corresponding graph G

図 1 ネットワークモデル
Fig. 1 Network model.

工夫を施すことにより，デッドロックを防ぐために課すパケットの転送禁止ターンの分散を実現する点が特徴である．

これにより L-turn routing はネットワーク全体のリンクを有効に使うことができ，ツリーの root 近辺のノードにトラフィックが偏る問題も解決できる．

以降，2 章では irregular network の形態を隠蔽するネットワークモデルについて述べ，3 章では irregular network における既存の up/down based routing について述べる．そして，4 章では L-turn routing の提案を行い，5 章にてシミュレーション結果を提示，考察を行う．また，6 章で今後の課題と結論を述べる．

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 どうしの接続など，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．

たとえば，専用のクラスタシステムと同様の並列処理を行うために分散配置されている Personal computer（PC）/Workstation（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 PC（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 PC（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** (a) の irregular network は図 1 (b) のように表すことができる．

図 1 (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. 既存のルーティングアルゴリズム

irregular network におけるルーティングアルゴリズムはここ数年来，様々な提案がなされており[1),2),6),7),10)]，up*/down* routing などはすでに実装されている[1)]．

本章では irregular network における代表的な 4 つのルーティングアルゴリズムを説明する．

3.1 Primitive Up/Down Routing

primitive up/down routing は単純な方法だが，irregular network のルーティングアルゴリズムの基礎となるものである．

まず，ネットワークを以下の規則に基づき論理的なツリーに見立て（spanning tree），その後ツリーの階層構造を利用したルーティングを行う．

spanning tree の構築は BFS（Breadth-First Search），または DFS（Depth-First Search）のいずれかの方針に基づいて行われるが，本稿では一般的である BFS を基にした spanning tree の構築方法を説明する．

(1) ネットワークの中から任意にツリーの root node を選ぶ．
(2) root node と隣接するノードをツリーに付け加える．
(3) ツリーの各ノードの隣接ノード（ツリーに含まれていないもの）をツリーに付け加えていき，

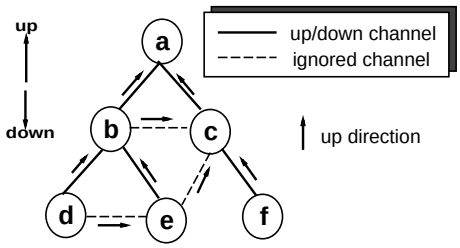


図 2 全域木
Fig. 2 A spanning tree.

全ノードがツリーに入るまでこの作業を繰り返す．

上記のアルゴリズムにより，ツリー上の root node 以外のノードは親ノードをただ 1 つ持つようになっている．また，親ノードと子ノード以外へのリンクはツリーを作成する際に無視される．

図 2 は図 1 (a) のネットワークに上記のアルゴリズムを適用した場合の一例である．破線のリンクは実際には存在するが，ツリーには含まれない．

root node 方向へのチャネルを up channel，root node から遠ざかる方向へのチャネルを down channel と呼ぶ．

ルーティングアルゴリズムは単純で，パケットの目的地ノードが現在地ノードの子孫ノードでない場合，up channel を用いて親ノードへパケットを転送し，目的地ノードが子孫ノードの場合，down channel を用いてそのノードを含むサブツリーにパケットを転送する．つまり，ルーティングは非最短型の determinisitic routing であり，一部の実在するリンクを利用することができない．

パケットは up channel を用いて root node 方向に必要 hop 数（0 を含む）移動した後，down channel を用いて root node から遠ざかる方向に必要 hop 数（0 を含む）移動する．したがって，up channel と down channel の間に循環構造をとることがないためデッドロックフリーが保証される．

3.2 Up*/Down* Routing in Autonet

Autonet で用いられている up*/down* routing [1] は primitive up/down routing で利用できなかったチャネルを up channel，または down channel に固定し，ルーティングに利用する点が primitive up/down routing と異なる．また，tree level が等しいノード間のリンクでは Node ID の小さい方から大きい方へのチャネルを down channel，逆を up channel とする．

パケットは primitive up/down routing と同様に必ず up channel を用いて root node 方向に必要 hop 数（0 を含む）移動した後，down channel を用いて root node から遠ざかる方向に必要 hop 数（0 を含む）移動する．

Autonet で用いられている up*/down* routing は up channel と down channel の間に循環構造をとることがないのでデッドロックフリーが保証される．

図 2 においてノード **a** からノード **e** にパケットを転送する場合，primitive up/down routing では，**a**→**b**→**e** と経路が 1 つなのに対し，Autonet で用いられている up*/down* routing では，それに加え **a**→**c**→**e** の経路も選択可能である．

Autonet で用いられている up*/down* routing は primitive up/down routing に比べてチャネルの使用制限が緩い点で有利である．

一般的に Autonet で用いられているものを up*/down* routing と呼ぶ場合が多く，本稿でも，以後特に指定しない場合，Autonet で用いられているものを up*/down* routing と呼ぶことにする．

3.3 Prefix Routing

Wu らにより提案された prefix routing [7] は primitive up/down routing で利用できなかったチャネルを cross channel として，ショートカットに用いることができる点が特徴である．

prefix routing ではツリーを作成する際，各ノードとチャネルにラベルを付け，すべてのチャネルを directed-graph に組み込み，ルーティングに利用する．

ノード v のラベル $L_n(v)$，ノード u からノード v へのチャネルのラベル $L_c(u,v)$ は以下の規則に基づいて生成される．

root node のラベルを $L_n(r) = 1$ とし，ノード $L_n(v)$ の k 番目の子ノード u のラベル を $L_n(u) = L_n(v) \parallel k$ とする．$\parallel$ は concatenation operation である．ツリーに含まれるチャネルに対しては，ノード v がノード u に比べ root node に近い場合，$L_c(v,u) = L_n(u)$，$L_c(u,v) = e$ とし，それ以外のチャネルに対しては，$L_c(v,u) = L_n(u)$，$L_c(u,v) = L_n(v)$ とする．e は空ラベルである（**図 3**）．

ルーティングは，目的地ノードのラベルの prefix を持つチャネルがあればそのチャネルを選択し，なければ，ラベル 'e' のチャネル（up channel）を選択する．これにより，各パケットは primitive up/down routing の経路をたかだか 1 回ショートカットすることが可能である．これは，パケットがショートカットを行うことが可能となる条件が，

(1) 現在地ノードが目的地ノードと子孫関係にない，
(2) ショートカット先のノードが目的地ノードまた

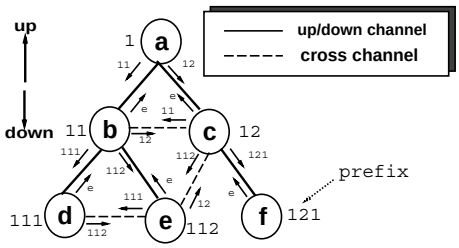


図 3 prefix routing に用いられる directed-graph
Fig. 3 Directed-graph for prefix routing.

は目的地ノードの先祖ノードとなる，
の 2 つであるために，一度ショートカットを行うとその後は primitive up/down routing と同様に，必ず down channel のみをたどって目的地ノードに到達するため (1) の条件が成立しなくなるからである．

たとえば図 2 においてノード **b(11)** からノード **f(121)** にパケットを転送する場合，primitive up/down routing では，**b(11)→a(1)→c(12)→f(121)** と 3 hop 必要だが，prefix routing では **b(11)→c(12)→f(121)** と 2 hop で到着することができる．

prefix routing ではラベルを調べるだけでルーティングができるので，他の up/down based routing に比べルーティングテーブルが小さいという特徴も持つ．また，prefix routing は cross channel を使用した場合でも以後 down channel のみを用いて転送するため，up channel と down channel の間に循環構造をとることがなく，デッドロックフリーが保証される．

3.4 Minimal Routing

Silla らは既存のルーティングアルゴリズムを escape path に使い，各リンクに fully adaptive なバーチャルチャネルを追加する minimal routing を提案し[6]，cyclic dependency を除去することなしに，デッドロックフリーを実現した．以下にルーティングアルゴリズムを簡単に述べる．

まず，各リンク間に 2 本のバーチャルチャネルもしくは物理チャネル（original channel，new channel）を用意する．そして，ルーティングは原則として new channel を用いて行う．new channel の使用条件は，escape path による経路と等しい，もしくはより短い経路をとることのみである．ただし，各ノードにおいて選択可能な new channel がすべて使用されている場合，original channel に切り換えてルーティングを行う．original channel では既存のルーティングアルゴリズムを用いる．一度 original channel を使用したパケットは二度と new channel を使用することはできない．

original channel は，既存のルーティングアルゴリズムを用いているためデッドロックフリーが保証されており，ネットワーク全体にわたる escape path になる．

minimal routing はパケットが new channel を使用した場合最短経路をとることが多いため効率が良く，既存の非最短型のルーティングアルゴリズムに比べて大きな性能向上を実現することができる．

3.5 既存のアルゴリズムの問題点

up/down based routing ではすべてのチャネルを up channel と down channel に分け，up channel と down channel の間に厳しい制限を設けデッドロックフリーを実現した．したがって，up/down based routing ではすべてのリンクを効果的に使うことが難しく，root node にトラフィックが偏る，チャネルが空いていても最短経路をとりにくい，などの問題点がある．

minimal rouitng はネットワークの各物理リンクを二重にする，もしくは各リンクに 2 本のバーチャルチャネルの導入が必要になり，また，original channel に用いるルーティングアルゴリズムの性能に左右されてしまう[6]．

4. L-turn Routing

本章では，前章で述べた問題点を解決するため，irregular network を directed-graph に見立てる際に工夫（L-R directed-graph）をし，新たに導入した論理的方向に基づいた left/right routing，および left/right routing のトラフィック分散能力を強化した Left-up-first turn routing（L-turn routing）を提案する．

以降，まず irregular network をツリー（L-R tree），および L-R directed-graph に見立てる手順を説明し，その後 left/right routing，および L-turn routing を提案する．

4.1 L-R Tree の作成

L-R tree は従来の up/down based routing に用いられている垂直方向（up/down）の概念のみを持つ一次元的な spanning tree に対し，垂直方向（up/down）と水平方向（left/right）の概念を持つ二次元的な spanning-tree である．

以下で，まず L-R tree における重要な概念についての定義を行い，その後 L-R tree および L-R directed-graph の定義を行う．

定義 1（Node ID） spanning tree に対して root node から Breadth First Search（BFS）を行い，ノードを訪問した順に昇順で各ノードに 0 以上の整数を割り当てる．この値を各ノードの *Node ID* とする．

定義 2（width） spanning tree に対して root

node から前順走査を行い，ノードを訪問した順に昇順で各ノードに 0 以上の整数を割り当てる．この値を各ノードの *width*とする．

定義 3（depth） spanning tree における，各ノードと root node との最短距離を各ノードの *depth*とする．

上記の定義における spanning tree は，文献 1) などに示される手法に基づいて構成する．

次に，二次元の概念の導入のために各チャネルに割り当てられる論理的な方向，horizontal direction と vertical direction を定義する．

定義 4（horizontal direction） 接続先のノードの width が接続元のノードの width より小さいチャネルの horizontal direction を *left direction*とし，接続先のノードの width が接続元のノードの width より大きいチャネルの horizontal direction を *right direction*とする．

定義 5（vertical direction） 接続元のノードの depth に比べ接続先のノードの depth が小さいチャネルの vertical direction を *up direction*とし，接続元のノードの depth に比べ接続先のノードの depth が大きいチャネルの vertical direction を *down direction*とする．depth の値が等しいノード間の各チャネルに関しては，left direction を持つチャネルに対しては up direction，right direction を持つチャネルに対しては down direction をそれぞれ vertical direction として割り当てる．

定義 4 と定義 5 よりすべてのチャネルは水平方向と垂直方向からなる二次元の論理的方向を持つ．

定義 6（L-R tree, L-R directed-graph） spanning tree を構成する全チャネルに対して horizontal direction および vertical direction を割り当てることにより構築した二次元の論理的方向を持つ有向グラフを *L-R tree*と呼び，irregular network における spanning tree 構成チャネル以外の全チャネルに対して horizontal direction および vertical direction を割り当て，これらのチャネルを L-R tree に付加して構築した二次元の論理的方向を持つ有向グラフを *L-R directed-graph*と呼ぶ．

図 4 は従来の up/down based routing で用いられる directed-graph の一例であり，**図 5** は図 4 の各ノードに width を割り当てることにより作成された L-R directed-graph である．

図 5 において直線の矢印は up direction を表し，点線の矢印は left direction を示している．

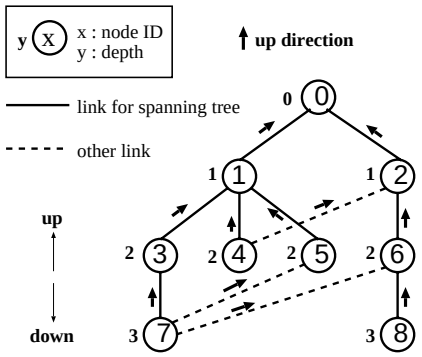


図 4 up/down directed-graph
Fig. 4 An up/down directed-graph.

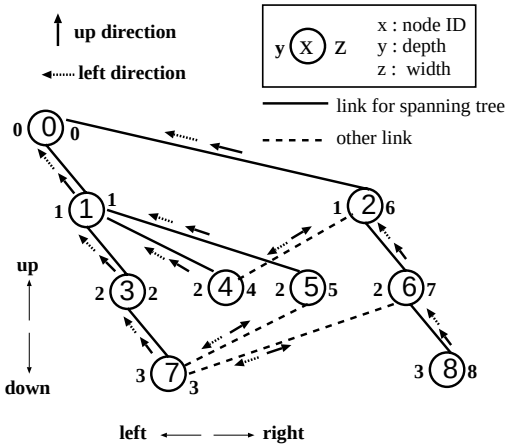


図 5 L-R directed-graph
Fig. 5 A L-R directed-graph.

なお，図 4 および図 5 において，各リンクは bidirectional であり，実線のリンクは spanning tree を構成しているリンクを，破線のリンクはそれ以外のリンクを表している．

また，ノードの円内に書かれた数字は Node ID，左側に書かれた数字は depth をそれぞれ示している．depth は垂直方向における root node からの最短距離を示す値であり，同階層間のノードには同じ値が割り当てられる．一方，図 5 のノードの右に書かれた数字は width を示しており，これは，定義 2 に基づいて前順走査を図 4 の spanning tree に対して行って割り当てたものである．width は水平方向における root node からの最短距離を示す値となる．

各ノードに対する width の割り当て方により，1 つの irregular network に対して複数の L-R directed-graph をマップすることが可能である．

4.2 Left/right Routing Algorithm

本節では前節で述べた L-R directed-graph 上での基本的なルーティングアルゴリズムである left/right routing を提案する．

まず，以下のように horizontal direction を用いて left channel，right channel を定義する．

定義 7（left channel，right channel） L-R directed-graph において，right direction を持つチャネルを *right channel*と呼ぶ．また，L-R directed-graph において，left direction を持つチャネルを *left channel*と呼ぶ．

たとえば，図 5 において，ID **2** のノードから ID **6** のノードへのチャネルは right channel であり，ID **2** のノードから ID **0**，**4** のノードへの各チャネルは left channel である．

left/right routing は left channel，right channel を用いて以下のように表される．

定義 8（left/right routing） left/right routing は L-R directed-graph において以下の条件を満たすルーティングアルゴリズムである．

- right channel を使用した後，left channel を使用してはならない．

left/right routing は上記の条件を守れば，いかなる経路も選択可能であり，非最短型の適応型ルーティングである．

定理 1 left/right routing はデッドロックフリーである．

Proof: right channel を使っているパケットは left channel を使うことができない．したがって left channel と right channel の間の cyclic dependency が切断されており left/right routing はデッドロックフリーである． ∎

定理 2 left/right routing は，primitive up/down routing において選択可能なすべての経路を選択することができる．

Proof: L-R tree において left channel による移動を行うと垂直方向において root node に近づき，right channel による移動を行うと垂直方向において root node から遠ざかる．L-R tree に含まれるチャネルはすべて spanning tree を構成するチャネルであるので，以上より L-R tree を構成する left channel と right channel はそれぞれ primitive up/down routing が用いる up channel および down channel と同じ特性を持つことがいえる．このため，L-R tree を構成する left channel を用いて必要 hop 数移動を行ってから right channel を用いて必要 hop 数移動を行うことにより，禁止ターンを行わずに primitive up/down routing と同じ経路を選択することが可能となる．

したがって，left/right routing は，primitive up/down routing において選択可能なすべての経路を選択することができる． ∎

定理 3 left/right routing は，prefix routing において選択可能なすべての経路を選択することができる．

Proof: prefix routing においてパケットは cross channel をたかだか 1 回使用して prmitive up/down routing の経路をショートカットし（3.3 節参照），cross channel の使用前には up direction のチャネルのみを 0 回以上使用し，使用後には down direction のチャネルのみを 0 回以上使用する．

定理 2 より，primitive up/down routing における up direction のチャネルは L-R directed-graph において left channel であり，down direction のチャネルは right channel である．したがって，left/right routing において prefix routing における cross channel に相当する channel が left channel，right channel のどちらであっても，cross channel を使用してショートカットを行った場合に left/right routing において禁止されているターンを行うことはない．

したがって，left/right routing は，prefix routing において選択可能なすべての経路を選択することができる． ∎

left/right routing はチャネルの使用制限を up/down based routing と違い，水平方向に課しているが，定理 2，定理 3 より任意のノード間における経路が保証されている．

なお，left/right routing においてライブロックを防ぐためにはパケットがつねに primitive up/down routing と同じ経路か，より短い経路をとるようにルーティングテーブルを作り，その中から出力チャネルの選択を行うようにすればよい．各ノードではパケットのヘッダが到着すると，パケットが到着した入力ポートと目的地ノードの組合せをインデックスとしてノードごとにあるルーティングテーブルを引くことにより出力ポートを決定し，ルーティングを行う．この手続きは up*/down* routing [1)] とほぼ同様である．

ルーティングテーブル作成はツリー作成後に root node に集められたトポロジ情報（コネクション情報）を基に各ノードがルーティングアルゴリズムの条件を満たす各目的ノードまでの最短経路を調べる，などの手法に基づいて行われる [1)]．

ルーティングテーブル作成アルゴリズムはほかにもあるが，いずれにせよルーティングテーブルは任意のノード間の connectivity を保証するよう作成される．

なお，ルーティングテーブルに目的地ノードまでの経路が複数あり，両者とも選択可能な場合，output selection function により出力チャネルを決定する[11)〜13)]．

left/right routing では up/down based routing で禁止されている down direction から up direction へのパケット転送の一部が可能になり，かつ prefix routing の経路をも選択可能である．

たとえば図 5 においてノード ID **3** からノード ID **8** へパケットを転送するような場合，prefix routing や up*/down* routing では **3→1→0→2→6→8** と root node を経由し，かつ 5 hop 必要だが，left/right routing では **3→7→6→8** と 3 hop で到着することができる．さらに，left/right routing は **3→1→0→2→6→8** の経路も選択可能である．

次に，ノード ID **7** からノード ID **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，left/right routing では right direction から left direction 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 root node を経由しない経路を選択することが可能となっており，root node へのトラフィックの集中を緩和している．

4.3 L-turn Routing Algorithm

本節では left/right routing における禁止ターンの配置を分散させることにより，より均等なトラフィックの分散を実現する Left-up-first turn routing（L-turn routing）を提案する．

まず，以下のようにチャネルの定義を行う．

定義 9（channels for L-turn routing）

- L-R directed-graph において up direction を持つ left channel を *left-up channel*，down direction を持つ left channel を *left-down channel* とする．
- L-R directed-graph において，up direction を持つ right channel を *right-up channel*，down direction を持つ right channel を *right-down channel* とする．

なお，定義 5 より depth の値が等しいノード間の right channel は right-down，left channel は left-up channel となる（**図 6**）．

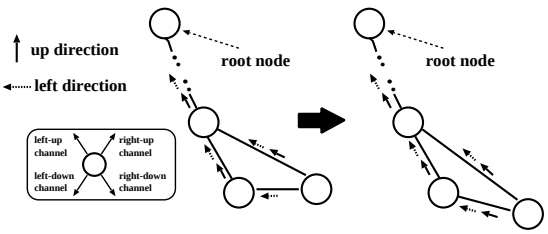


図 6 root node と等距離にあるノードの位置関係
Fig. 6 A position between nodes at equal distances.

定義 9 より，L-R directed graph 上のチャネルはすべて left-up，left-down，right-up，right-down の 4 種類のチャネルに分類される．

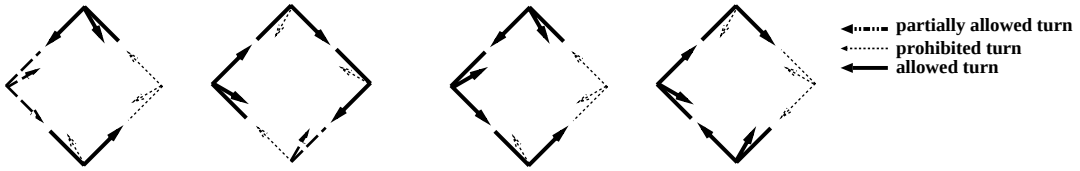
定義 10（L-turn routing） L-turn routing は L-R directed-graph において以下の条件を満たすルーティングアルゴリズムである．

- left-up channel 以外の channel を使用した後に，left-up channel を使用してはならない．
- right-up channel から left-down channel へのターンを禁止する．
- left-up channel を含まず，かつ right-up channel から left-down channel へのターンを含まない循環が L-R directed-graph 上に存在する場合，その循環を構成している left-down channel から right direction を持つチャネルへのターンを禁止する．

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な循環構造をとるためである．L-turn routing では定義 10 により，循環構造を作らないように L-R directed-graph 上のルーティングにおけるターン方向を**図 7** (a) のように制限している．図 7 において薄色の破線はルーティングが禁止されているターンを示しており，実線はルーティングが許可されているターンを示している．また，色の濃い破線は L-R directed-graph において循環を構成する可能性がある場合にルーティングを禁止する必要があるターンを表している．

定義 10 において，はじめの 2 条件のみでは**図 8** のような，left-up channel を含まず，かつ right-up channel から left-down channel へのターンを含まない循環構造を防ぐことができない．

そこで最後の条件により，ルーティングテーブル作成時に，探索を行いこれらの循環が検出された場合には循環を構成している left-down channel から right-up channel へのターン，もしくは left-down channel から right-down channel へのターンを禁止する．これにより，L-R directed-graph において left-down channel から right-up channel のターン，もしくは left-down



(a) Left-up-first turn routing (b) Left/right routing

図 7 L-turn rouitng と left/right routing
Fig. 7 L-turn routing and left/right routing.

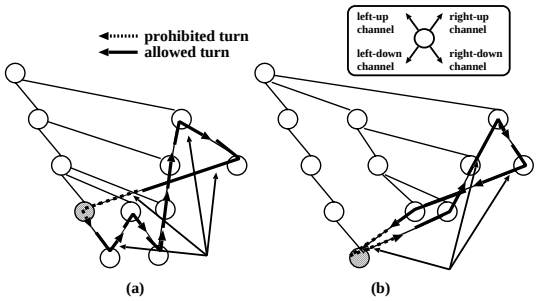


図 8 left-down channel から right 方向へのターンを禁止する場合
Fig. 8 Cases in which turns from left-down channel to right direction are prohibited.

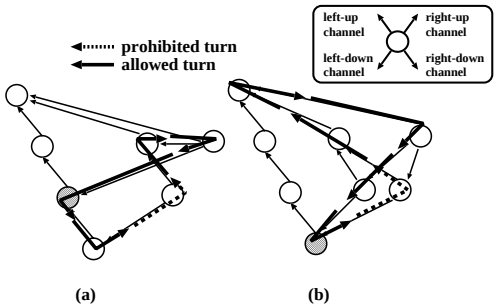


図 9 left-down channel から right 方向へのターンが可能な場合
Fig. 9 Cases in which turns from left-down channel to right direction are allowed.

channel から right-down channel へのターンの一部が禁止される可能性がある．たとえば，図 8 の各々の斜線のノードにおける上記のターンは上記の循環構造を構成するため禁止されるが，**図 9** においては上記の循環構造が存在しないので斜線のノード上の各ターンは許可される．

デッドロックを防ぐために必要となる禁止ターンの組合せは，L-turn routing における組合せ以外のパターンを考えることが可能ではあるが，connectivity の保証と禁止ターンの分散を両立させることは難しい．

なお，left/right routing の場合は図 7 (b) のようにパケットの禁止ターンが右端に偏っており，トラフィックの偏りを招きやすい．一方，L-turn routing の場合，場合によっては禁止ターンが増えるが，図 7 (a) のようにルーティングできないターンが分散しているためトラフィックが偏りにくく，パケットがスムーズに流れることが期待できる．

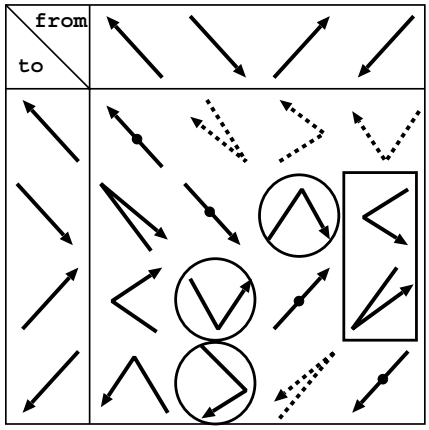


図 10 L-R directed-graph における全ターン
Fig. 10 All turns in L-R directed-graph.

定理 4 left-up channel を含まず，かつ right-up channel から left-down channel へのターンを含まない循環には left-down channel から right-down channel，もしくは right-up channel へのターンが含まれる．

Proof: L-R directed-graph 上で形成されうるすべてのターンは**図 10** に示す 16 通りのターンである．図 10 の横軸はターンする前のチャネル，縦軸はターン後のチャネルを表しており，破線のターンは L-turn routing においてつねに禁止されるターンである．

16 通りのターンから同方向へのターン，left-up channel を含むターン，および right-up channel から left-down channel へのターンを除いたターンは，図 10 の円に囲まれた 3 通りのターン（right-up→right-down, right-down→right-up, right-down→left-down）と四角で囲まれた 2 通りのターン（left-down→right-down, left-down→right-up）の計 5 通りである．

前者の 3 つのターンの horizontal direction を考えると，right-up channel から right-down channel へのターン，right-down channel から right-up channel へのターンはいずれも right direction へ進むものであり，また right-down channel から left-down channel へのターンは right direction から left direction へのターンである．したがって，前者の 3 つのターンのみでは left direction から right direction へターンできないため循環を構成することができない．

よって left-up channel を含まず，かつ right-up channel から left-down channel へのターンを含まない循環には left-down channel から right-down channel もしくは right-up channel へのターンが含まれる．∎

定理 5 L-turn routing はデッドロックフリーである．

Proof: L-R directed-graph において L-turn routing は定義 10 より，left-up 方向への channel とその他の channel との間の cyclic dependency が切断されている．また，図 8 のように left-up channel を含まず，かつ right-up channel から left-down channel へのターンを含まない特殊な循環が存在する．これは循環を構成する left-down channel から right-down channel または left-down channel から right-up channel へのターンを禁止することで，定理 4 より cyclic dependency が切断される．したがって，すべての cyclic dependency が切断されているので L-turn routing はデッドロックフリーである．∎

定理 6 L-turn routing は，primitive up/down routing において選択可能なすべての経路を選択することができる．

Proof: L-R tree において left-up channel による移動を行うと垂直方向において root node に近づき，right-down channel による移動を行うと垂直方向において root node から遠ざかる．L-R tree に含まれるチャネルはすべて spanning tree を構成するチャネルであるので，以上より L-R tree を構成する left-up channel と right-down channel はそれぞれ primitive up/down routing が用いる up channel および down channel と同じ特性を持つことがいえる．このため，L-R tree を構成する left-up channel を用いて必要 hop 数移動を行ってから right-down channel を用いて必要 hop 数移動を行うことにより，禁止ターンを行わずに primitive up/down routing と同じ経路を選択することが可能となる．

したがって，L-turn routing は，primitive up/down routing において選択可能なすべての経路を選択することができる．∎

なお，ルーティングテーブルの作成は，left-down channel から right-up channel へのターンまたは left-down channel から right-down channel へのターンが存在するノードごとに循環を構成する可能性があるかどうかを調べる手続きが追加される点を除いては left/right routing の場合とほぼ同様の方法で実現される．

L-turn routing の特徴をまとめると以下のようになる．

目的地ノードまでの経路を保証：

L-turn routing はチャネルの使用制限が特殊であるが，任意のノード間の経路を保証する．

パケットの平均 hop 数の抑制と高い adaptivity：

L-turn routing は L-R directed-graph を用いることにより，primitive up/down routing の経路を含む様々な経路の中から出力経路を選択できるため，チャネルを効果的に使うことができ，スループットに大きな影響を与えるパケットの平均 hop 数を抑えることができる．

様々なトポロジへの高い適応性：

left/right routing では，ツリーの root node 付近にトラフィックが偏りやすいという up/down routing の問題点[9]を解決するためにトラフィックを left，right 方向に分散するようにした．

L-turn routing では，さらに left/right routing に比べ様々なトポロジに対し安定した性能が得られるようにルーティング禁止ターンを分散させている（図 7 (a)）．これにより，様々なトポロジの irregular network に対し安定した高い性能が期待できる．

最近，up*/down* routing をもとにトラフィックの偏りを防ぐ試み[8),14)]も行われているが，前者はルーティングテーブルの作成手続きが複雑である，後者は各ノードにおいて大量のバッファを必要とする，などの問題がある．

5. 評　　価

提案した L-turn routing，left/right routing，そして既存の primitive up/down routing，prefix routing，up*/down* routing，minimal routing の各ルーティングアルゴリズムを C++ で記述したフリットレベルシミュレータを用いて評価を行った．シミュレーションに用いた条件を**表 1** に示す．

シミュレーションではじめの 5,000 クロックはネットワークが安定せず，想定した負荷に達していないと

考えられるため無視して評価を行った．なお，1 ノードにつき 1 プロセッシングエレメントと仮定した．ネットワークのトポロジについては，irregular network での性能評価としてランダムに 9 ノードを接続したもの（**図 11** (a)）と 16 ノードをランダムに接続した 3 種類のトポロジ（図 11 (b)，(c)，(d)）を用いた．さらに，比較のため典型的な regular network である 4×4 2D torus，6 × 6 2D mesh の評価を行った．

表 1 シミュレーションパラメータ
Table 1 Simulation parameters.

Simulation time	50,000 clocks (ignore the first 5,000 clocks)
Network	Irregular network, 4×4 2D torus or 6×6 2D mesh
Network size	9, 16 or 36 nodes
The number of virtual channels	1 (primitive up/down, prefix, up*/down*, left/right, L-turn) 2 (ones with minimal routing)
Packet length	128 flits
Switching tech.	wormhole
Traffic pattern	uniform
Flit transfer time	3 clocks

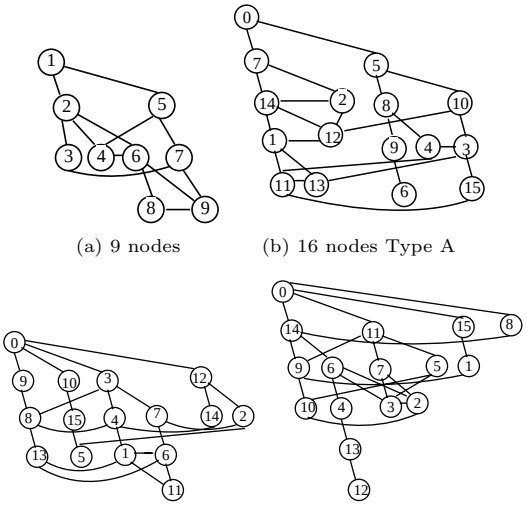


図 11 シミュレーションに用いた Network
Fig. 11 Networks in simulations.

ルーティングアルゴリズムについては primitive up/down，prefix，up*/down*，left/right，L-turn についてそのルーティングアルゴリズム単体での評価（バーチャルチャネル 1 本）と minimal routing を併用したとき（バーチャルチャネル 2 本）の評価を行った．

また，irregular network では各パケットがネットワークにかける負担を抑えることが性能向上につながる[3]．よって，up*/down* routing，left/right routing，L-turn routing については選択可能な経路の中から最短のものを選択するようにした．

5.1 9 Nodes Irregular Network

9 nodes irregular network での評価を**図 12** に示す．図 12 (a) はリンク間のバーチャルチャネルが 1 本のとき，図 12 (b) はリンク間のバーチャルチャネルが 2 本で，各アルゴリズムに minimal routing を併用したときのものである．また，横軸は受信トラフィック，縦軸はレイテンシを表している．受信トラフィックは，全ノードが毎クロックに 1 flit 受信する場合を 1.00 としており，レイテンシは出発地ノードのプロセッサがパケットを生成してから，目的地ノード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 では各パケットの目的地ノードはランダムに決定されており，等確率に分散されている．

図 12 (a) より，L-turn routing が最も高い性能を示した．また，図 12 (b) より minimal routing を併用した場合，各ルーティングアルゴリズムの性能差が

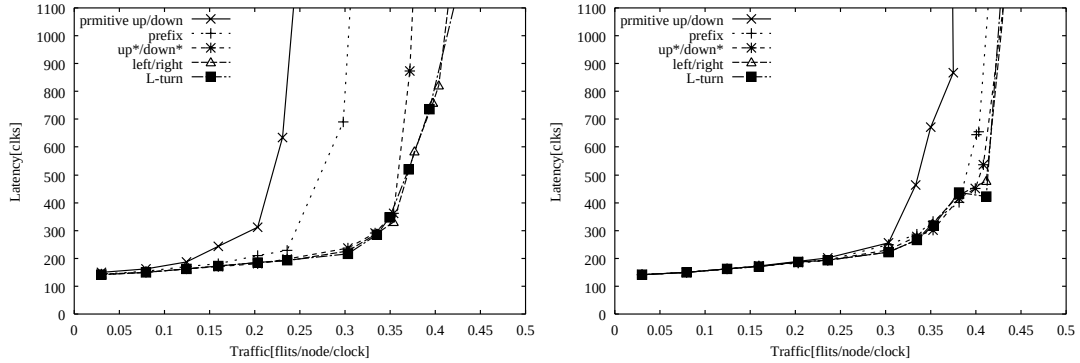


図 12 9 nodes irregular network
Fig. 12 9 nodes irregular network.

縮まっていることが分かる．これは多くのパケットが minimal routing を行うチャネル（new channel）のみを利用し，original channel 上で各ルーティングアルゴリズムを利用するパケットが減少しているからである．しかし minimal routing を併用した場合にも original channel 上で高いトラフィック分散能力を持つ L-turn routing が一番高い性能を示した．

また，平均 hop 数の評価を**表 2** に示す．バーチャルチャネルの本数が 2 本になると，どのルーティングアルゴリズムでも最短経路を選択可能な new channel を使用するため平均 hop 数の差が小さくなるが，original channel の平均 hop 数の差がパフォーマンスに影響を及ぼしていることが分かる．

5.2 16 Nodes Irregular Networks

16 nodes irregular networks での評価を**図 13** に示す．

図 13 (a)，(c)，(e) はリンク間のバーチャルチャネルが 1 本のとき，図 13 (b)，(d)，(f) はリンク間のバーチャルチャネルが 2 本で，minimal routing を各アルゴリズムと併用したときのものである．

また，図 13 において (a)，(b) は図 11 (c) の Type A，(c)，(d) は図 11 (d) の Type B，(e)，(f) は図 11 (e) の Type C のトポロジでの評価である．

図 13 より同条件下では left/right routing，L-turn routing とも既存のものよりも安定して高い性能を示していることが分かる．

次に，平均 hop 数の評価を**表 3** に示す．L-turn routing，left/right routing，up*/down* routing ともほぼ同じ平均 hop 数を示したため，ルーティングアルゴリズムの性能差は利用可能な経路の分布状況などによるパケットの衝突の頻度の違いなどにより生じたと考えられる．

5.3 16 Nodes 2D Torus

4 × 4 2D torus での評価を**図 14** に示す．

図 14 (a) はリンク間のバーチャルチャネルが 1 本のとき，図 14 (b) はリンク間のバーチャルチャネルが 2 本で，minimal routing を各アルゴリズムと併用したときのものである．

図 14 (a) より，ノード数が 16 の規則的なトポロジにおいても L-turn routing はパケットの転送が禁止されているターンが分散されておりネットワーク全体を効果的に使うことができ，高い性能を示していることが分かる．また，図 12 (a) と比較すると，ノード数が増加したときに各ルーティングアルゴリズムの差は顕著となる．これはノード数が増加するにともない各ノード間のリンクの本数が増加し，禁止ターンが分散している L-turn routing はトラフィックの分散を効果的に行うことができたためである．

このことより，L-turn routing はノード数が多い結合網においてより有利であることがいえる．

また，図 14 より，minimal routing を併用しないときの L-turn routing はバーチャルチャネルが 1 本にもかかわらず，バーチャルチャネルを 2 本用いて minimal routing と併用している他のルーティングアルゴリズムと同等の性能を示していることが分かる．

次に，平均 hop 数の評価を**表 4** に示す．表 4 に示すとおり，left/right routing，L-turn routing の平均 hop 数が最も少ない．表 4 より，irregular network のルーティングアルゴリズムは非最短型であるため，平均 hop 数を抑えることが性能向上に結びついている

表 2 9 ノードにおける平均 hop 数
Table 2 Average number of hops in 9 nodes.

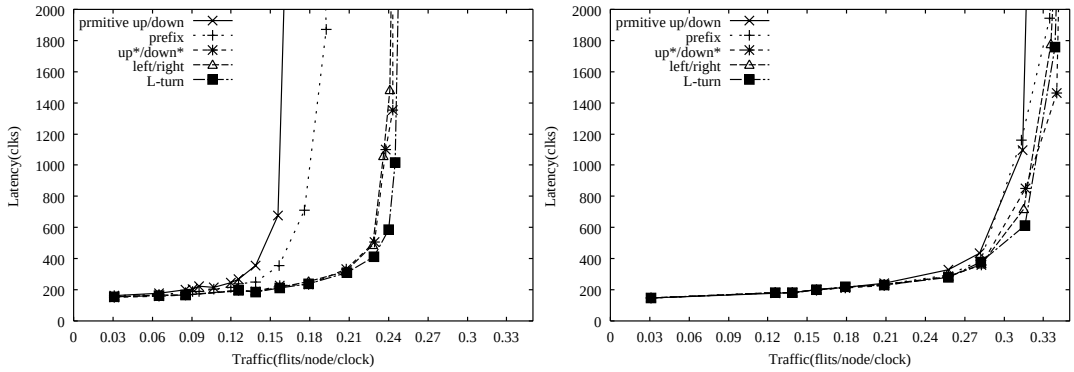
	Vch × 1	Vch × 2
primitive up/down	2.73	1.95
prefix	2.08	1.85
up*/down*	1.93	1.84
left/right	1.91	1.82
L-turn	1.91	1.81

表 4 16 nodes torus における平均 hop 数
Table 4 Average number of hops in 16 nodes torus.

	Vch × 1	Vch × 2
primitive up/down	3.33	2.26
prefix	2.50	2.17
up*/down*	2.22	2.16
left/right	2.13	2.12
L-turn	2.15	2.13

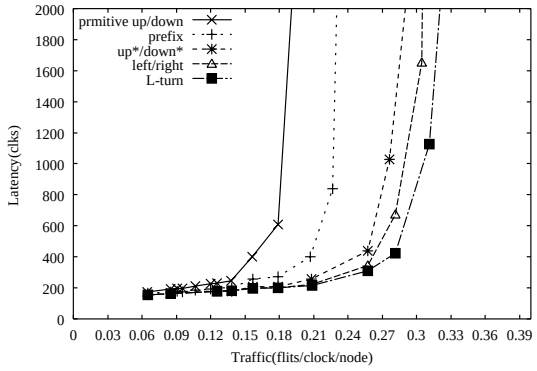
表 3 16 nodes irregular networks における平均 hop 数
Table 3 Average number of hops in 16 nodes irregular networks.

	Type A		Type B		Type C	
	Vch×1	Vch×2	Vch×1	Vch×2	Vch×1	Vch×2
primitive up/down	3.87	2.84	3.47	2.63	3.59	2.82
prefix	3.43	2.76	2.96	2.55	2.97	2.69
up*/down*	2.87	2.67	2.66	2.51	2.79	2.66
left/right	2.98	2.75	2.66	2.51	2.73	2.67
L-turn	2.81	2.81	2.66	2.51	2.75	2.67

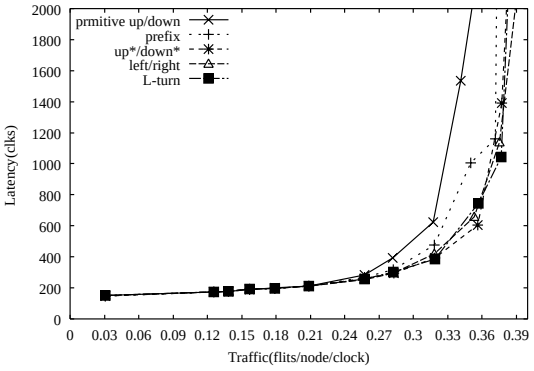


(a) 1 virtual channel in Type A

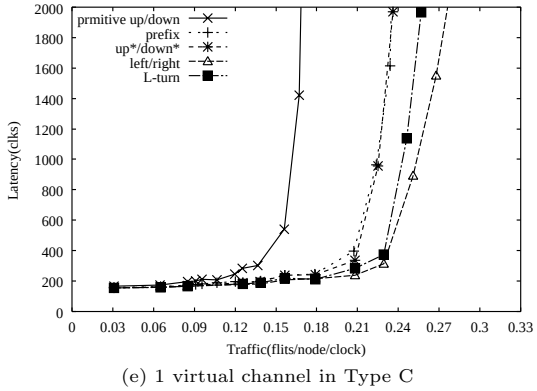
(b) 2 virtual channels in Type A



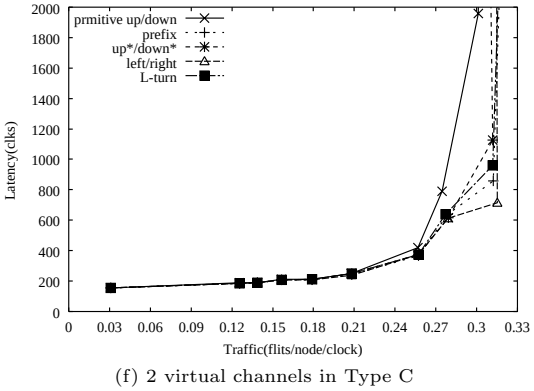
(c) 1 virtual channel in Type B



(d) 2 virtual channels in Type B



(e) 1 virtual channel in Type C



(f) 2 virtual channels in Type C

図 13 16 nodes irregular networks
Fig. 13 16 nodes irregular networks.

ことが分かる．

5.4 36 Nodes 2D Mesh

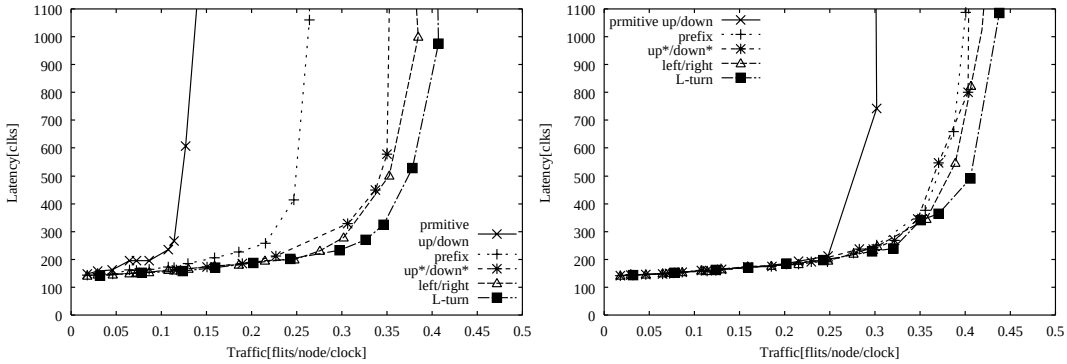
6×6 2D mesh での評価を**図 15**，**図 16** に示す．

図 15 (a)，図 16 (a) はリンク間のバーチャルチャネルが 1 本のとき，図 15 (b)，図 16 (b) はリンク間のバーチャルチャネルが 2 本で，minimal routing を各アルゴリズムと併用したときのものである．

root node を図 15 の場合，mesh の端のノード $(0,0)$，図 16 ではノード $(2,2)$ に決め，MDST（minimal-depth spanning tree）[1] を基にした directed-graph を用いて各ルーティングアルゴリズムの評価を行った．

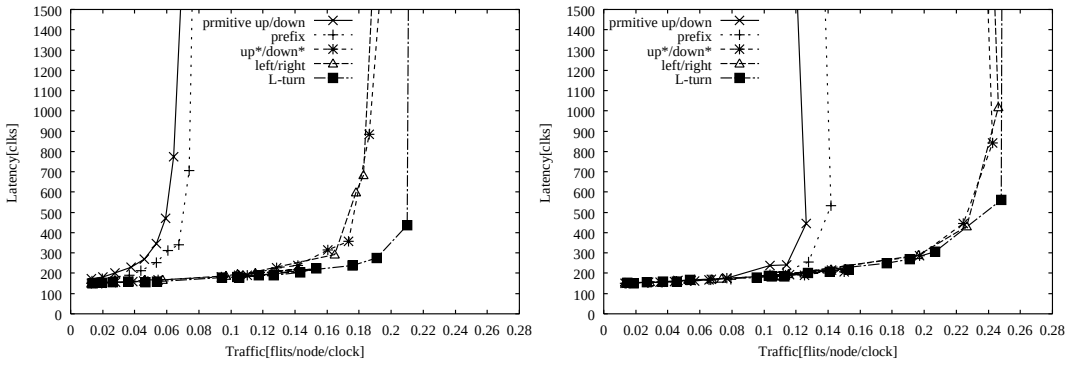
図 15 (a)，図 16 (a) において，L-turn routing と left/right routing，up*/down* routing の性能差が著しい．これは L-turn routing はパケットの転送禁止ターンが分散しているため，様々な directed-graph に対し安定した性能を発揮することを示しいる．

また，図 15，図 16 より L-turn routing はバーチャ



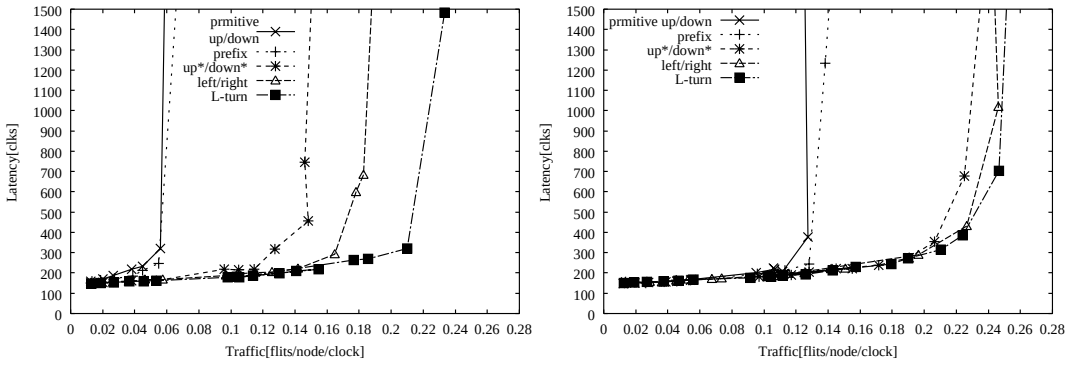
(a) 1 virtual channel (b) 2 virtual channels with minimal routing

図 14 16 nodes 2D torus
Fig. 14 16 nodes 2D torus.



(a) 1 virtual channel (b) 2 virtual channels with minimal routing

図 15 36 nodes 2D mesh（root node が (0,0) の場合）
Fig. 15 36 nodes 2D mesh (when root node is (0,0)).



(a) 1 virtual channel (b) 2 virtual channels with minimal routing

図 16 36 nodes 2D mesh（root node が (2,2) の場合）
Fig. 16 36 nodes 2D mesh (when root node is (2,2)).

ルチャネルが 1 本にもかかわらず，minimal routing を併用したその他のルーティングアルゴリズムとほぼ同等の性能を示していることが分かる．

次に，平均 hop 数の評価を**表 5** に示す．表 5 に示すとおり，left/right routing，L-turn routing の平均 hop 数が最も少なく，mesh においても安定した性能を示している．

5.5 物理チャネル利用率の分布

次に各ルーティングアルゴリズムにおいて，各ノードの物理チャネル利用率の分布について調査する．

4×4 torus における uniform traffic での評価である図 14 において，各ルーティングアルゴリズムのネットワークが飽和する直前時の各物理チャネル利用率の分布を**図 17** に示す．root node の位置は座標 $(0, 0)$ である．

縦軸はノードごとのピークチャネル利用率を表す．ピークチャネル利用率とは，各ノードごとに 1,000 clk ごとのチャネル利用率を集計し，5,000〜50,000 clk の中での最高値を表す．また，横軸はノードの位置を表す．ルーティングアルゴリズムがトラフィックを適切に分散させていれば，飽和直前時における各ノードのピークチャネル利用率を高く，均一にできるはずである．つまり，ネットワークの性能を最大限引き出しているはずである．

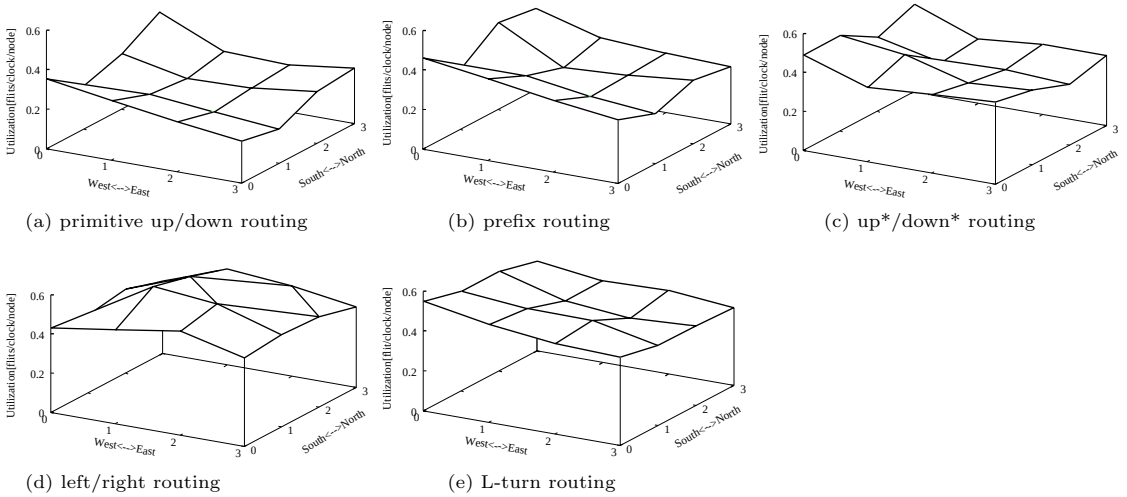
torus における uniform traffic では，トポロジが均一かつ，トラフィックが均等に分散されるため，ネットワークを効果的に使うことが比較的容易な条件である．しかし，図 17 より，L-turn routing 以外のルーティングアルゴリズムは大きくトラフィックが偏ってしまっていることが分かる．primitive up/down routing，prefix routing，up*/down* routing では root node 付近にトラフィックが偏り，left/right routing では root node 付近にトラフィックが偏らないが，別のノードに偏りが生じた．一方，L-turn routing は偏りを完全になくしたとはいいきれないが，大きなトラフィックの偏りを抑えており，また，ピークチャネル利用率がかなり高い状況でもネットワークが飽和していない．その結果高い性能につながったと考えられる．

L-turn routing ではトラフィックを分散させるために，up*/down* routing，left/right routing と違い，ノードの位置関係を 2 次元のグラフで考えている．ネットワークをさらに 3 次元，4 次元と高次元なグラフでとらえることも可能である．しかし，各ノード間の connectivity を保証するには禁止ターンを減らす必要があり，またデッドロックフリーを実現するためには禁止ターンが一定数必要である．このトレードオフを複雑な高次元なグラフ上で解決し，実用に耐えうる範囲の手順で実現することは，2 次元の場合よりも難しい問題である．ただ，今後ネットワークを多次元のグラフに展開し，そのうえで効果的な手法が登場し，高い性能を持ったルーティングアルゴリズムが登場する可能性はある．

図 17 より L-turn routing は，目標の 1 つである他のルーティングアルゴリズムよりもトラフィックの高い分散効果があることが確認された．

表 5 36 ノードにおける平均 hop 数
Table 5 Average number of hops in 36 nodes.

	root node is (0,0)		root node is (2,2)	
	Vch×1	Vch×2	Vch×1	Vch×2
primitive U/D	6.51	4.57	4.73	4.24
prefix	5.73	4.35	4.55	4.14
up*/down*	4.06	3.97	4.04	4.04
left/right	4.02	4.00	4.02	4.00
L-turn	4.00	3.99	4.04	3.99



(a) primitive up/down routing (b) prefix routing (c) up*/down* routing

(d) left/right routing (e) L-turn routing

図 17 ピークチャネル利用率
Fig. 17 Peak channel utilization.

6. ま と め

irregular network において高いトラフィック分散能力を持つ adaptive routing である L-turn routing，および L-turn routing に必要な L-R tree の提案を行った．

L-turn routing は irregular network を 2 次元の論理的な方向を持つ L-R directed-graph に見立て，パケットの論理的なターン方向の一部を制限することにより，全ノードへの経路の保証とデッドロックフリーを実現する．さらに，L-turn routing はトラフィックの偏りを抑えるため，ネットワーク全体をより効果的に利用することができる．

シミュレーションの結果より，L-turn routing は今回評価を行ったすべてのトポロジにおいて既存のルーティングアルゴリズムよりも高い性能を示すことが分かった．

今後は L-turn routing の自由度を十分に生かせるような L-R tree の構成法，および L-turn routing における出力チャネルの選択法（output selection function）[11),12)]を検討し，さらなる性能向上をはかる予定である．また今回は 36 ノードまでの評価を行ったが，より多いノード数や様々なトポロジでの評価も行う予定である．

参 考 文 献

1) Schroeder, M.D., et al.: Autonet: A high-speed, selfconfiguring local area network using point-to-point links, Technical Report SRC research report 59, DEC (1990).
2) Boden, N.J., et al.: Myrinet: A Gigabit-per-Second Local Area Network, *IEEE Micro*, Vol.15, No.1, pp.29–35 (1995).
3) Silla, F. and Duato, J.: Efficient Adaptive Routing in Networks of Workstations with Irregular Topology, *Proc. Workshop Comm. and Architectural Support for Network-Based Parallel Computing*, pp.46–60 (1997).
4) Nishi, H., Tasho, K., Yamamoto, J., Kudou, T. and Amano, H.: A Local Area System Network RHiNet-1/SW: A Network for High Performance Parallel Computing, *High Performance Distributed Computing*, pp.296–297 (2000).
5) Kudoh, T., Nishimura, S., Yamamoto, J., Nishi, H., Tatebe, O. and Amano, H.: RHiNET: A network for high performance parallel computing using locally distributed computing, *Proc. Innovative Architecture for Future Generation High-Performance Processors and Systems*, pp.69–73 (1999).
6) Silla, F. and Duato, J.: High-Performance Routing in Networks of Workstations with Irregular Topology, *IEEE Trans. parallel and distributed systems*, Vol.11, No.7, pp.699–719 (2000).
7) Wu, J. and Sheng, L.: Deadlock-Free Routing in Irregular Networks Using Prefix Routing, DIMACS Technical Report 99-19 (1999).
8) Sancho, J.C., Robles, A. and Duato, J.: A Flexible Routing Scheme for Networks of Workstations, *Proc. 2000 International Symposium on High Performance Computing*, pp.260–267 (2000).
9) Silla, F. and Duato, J.: Is it Worth the Flexibility Provided by Irregular Topologies in Networks of Workstations?, *Proc. Workshop Comm. and Architectural Support for Network-Based Parallel Computing*, pp.47–61 (1999).
10) Silla, F. and Duato, J.: On the Use of Virtual Channles in Networks of Workstations with Irregular Topology, *IEEE Trans. Parallel and Distributed Systems*, Vol.11, No.8, pp.813–828 (2000).
11) 鯉渕道紘，舟橋　啓，上樂明也，天野英晴：適応型ルーティングにおける output selection function に関する研究，情報処理学会論文誌，Vol.42, No.4, pp.704–713 (2001).
12) Martinez, J.C., Silla, F., Lopez, P. and Duato, J.: On the Influence of the Selection Function on the Performance of Networks of Workstations, *Proc. 2000 International Symposium on High Performance Computing*, pp.292–300 (2000).
13) Schwiebert, L.: A Performance Evaluation of Fully Adaptive WormholeRouting including Selection Function Choice, *IEEE International Performance, Computing, and Communications Conference*, pp.117–123 (2000).
14) Flich, J., Lopez, P., Malumbres, M.P., Duato, J. and Rokicki, T.: Combining In-Transit Buffers with Optimized Routing Schemes to Boost the Performance of Networks with Source Routing, *Proc. 2000 International Symposium on High Performance Computing*, pp.300–309 (2000).

(平成 13 年 2 月 15 日受付)
(平成 13 年 6 月 7 日採録)

鯉渕　道紘

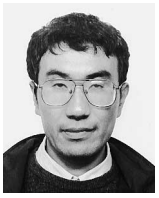
平成 12 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院理工学研究科修士課程に在学中．計算機アーキテクチャに関する研究に従事．

舟橋　　啓（正会員）

平成 7 年慶應義塾大学理工学部電気工学科卒業．平成 12 年同大学大学院博士課程修了．現在，三重大学工学部情報工学科助手．工学博士．計算機アーキテクチャに関する研究に従事．

上樂　明也

平成 10 年慶應義塾大学理工学部電気工学科卒業．平成 12 年同大学大学院修士課程修了．現在，同大学院理工学研究科研究生．計算機アーキテクチャに関する研究に従事．



天野　英晴（正会員）

昭和 56 年慶應義塾大学工学部電気工学科卒業．昭和 61 年同大学大学院工学研究科電気工学専攻博士課程修了．現在，慶應義塾大学理工学部情報工学科教授．工学博士．計算機アーキテクチャの研究に従事．