2次元 Turn モデルに基づくイレギュラーネットワーク向けルーティングアルゴリズムの設計と評価

上 樂 明 也† 鯉 渕 道 紘† 天 野 英 晴†

大規模 PC クラスタにおける PC 間の相互結合には，通常，トポロジに制限のないイレギュラーネットワークが用いられる．しかし，イレギュラーネットワークにおける代表的なデッドロックフリールーティングアルゴリズムである up*/down* ルーティングは，1 次元有向グラフを基に Turn モデルを適用するため，パケットの転送禁止ターンが偏り，効率的にネットワークバンド幅を利用することが難しい．本稿では，up*/down* ルーティングにおける問題点を改善するため，既存の有向グラフを 2 次元に拡張する．そして，2 次元有向グラフに対して Turn モデルを適用することにより，パケット転送禁止ターンを分散する効率的なデッドロックフリールーティングアルゴリズムを設計する手法を示す．シミュレーション結果より，提案したルーティングアルゴリズムは up*/down* ルーティングに比べ，最大 96% のスループット向上を実現することがわかった．

The Design and Evaluation of Routing Algorithms Based on 2D Turn Model for Irregular Networks

Akiya Jouraku,† Michihiro Koibuchi† and Hideharu Amano†

Irregular networks are usually used to connect personal computers in massively PC clusters. Up*/down* routing algorithm, which is a typical deadlock-free routing algorithm in irregular networks, tends to make traffic unbalancing because it is based on a one-dimensional directed-graph. In order to solve the traffic unbalancing caused by up*/down* routing algorithm for irregular networks, two-dimensional directed graph is introduced, and novel routing algorithms based on two-dimensional turn model are proposed. The proposed routing algorithms improve traffic balancing by distributing prohibited turns of packet transfer over the network. Simulation results demonstrate that proposed algorithms improve throughput up to 96%.

1. はじめに

PC(パーソナルコンピュータ) および WS(ワークステーション) などのコモディティコンポーネントを Myrinet[1] に代表される高速なネットワーク —SAN (システムエリアネットワーク)— で接続することにより構築される高性能クラスタシステムはコストパフォーマンスに優れた大規模並列/分散コンピューティング環境として注目されている[1)2)3)]．

SAN は，point-to-point リンクにより相互結合される高速なスイッチ群から構成され，分散配置されている PC/WS をユーザが容易に接続できることを重要視している．そのため，結合方式としてはトポロジに制限がないイレギュラーネットワークが用いられることが多い．しかし，イレギュラーネットワークでは，並列計算機で用いられるトーラスなどのレギュラーネットワークに比べて経路保証とデッドロックフリーの実現が難しいため，ルーティングアルゴリズムの設計がより複雑になる．このため，既存のルーティングアルゴリズムのほとんどは，この問題を解決するためにトポロジ上へスパニングツリーのマッピングを行ない，ツリー構造が持つ結合性および非循環性の特性を利用している．これらの中で最も代表的なルーティングアルゴリズムである up*/down* ルーティング[4)] は，ネットワークを 1 次元の方向 (up/down) を持つ有向グラフに見立て，単純な 1 次元の Turn モデル[5)] を適用することによりデッドロックフリーを実現している．しかし，1 次元の Turn モデルでは禁止ターンの偏りが大きくなるため，トラフィックの分散の実現が困難になり，効率的にネットワークのバンド幅を利用することが難しくなってしまう．

我々は，up*/down* ルーティングにおける上記の問題点を改善するために，2 次元の方向 (up/down と left/right) を導入したルーティングアルゴリズム[6)] を提案した．しかし，この提案では循環構造除去に必要な禁止ターン集合の選択法が確立されていなかったため，ルーティングアルゴリズムの一例を示すにとどまった．

そこで，本稿では，Turn モデルの手法[5)] に基づき，循環構造除去のための経路の分散を考慮した禁止ターン集合の決定手法を提案する．これにより，形成可能なデッドロックフリールーティングアルゴリズムが全部で 4 つ存在することを明らかにする．更に，任意のトポロジを選択可能なフリットレベルシミュレータを用いることにより，提案したルーティングアルゴリズ

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

ムの評価を様々なパターンのトポロジにおいて行う．

2. up*/down* ルーティング

up*/down* ルーティングは，イレギュラーネットワーク向けの適応型ルーティングであり，Autonet[4] や Myrinet[1] などで実用化されている．

up*/down* ルーティングは，トポロジ上のすべてのチャネルに *up* または *down* の方向を割当てた有向グラフを必要とする．そのため，最初に，スイッチをノードとしたスパニングツリーを構築する必要がある．代表的なスパニングツリーの構築方法は，Autonet で用いられている BFS(Breadth-First Search) に基づく方法[4] である．

有向グラフの構築は，BFS スパニングツリーを構築した後，ネットワークの全チャネルに対して，次のように方向を割当てることによって行なわれる．

1. up 方向を次の 2 つの条件のいずれかを満たすチャネルに対して割当てる．
   a. 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
   b. 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 が移動元のノード ID よりも小さい
2. 残りの全てのチャネルに対して down 方向を割当てる．

up*/down* ルーティングは，デッドロックフリーと任意のノード間の経路を保証するために，次のような 1 次元の Turn モデルを適用している．

まず，構築した有向グラフには up, down の 2 つの方向のみが存在するので，パケット転送時に発生するターンは，up→down および down→up の 2 パターンとなり，これらのターンの連鎖により 1 パターンの循環構造が形成されうる．従って，全てのパケットは必ず 0 回以上 up 方向に必要なだけ移動した後に 0 回以上 down 方向に移動して目的ノードまで到達する，という単純な制限により，down 方向から up 方向へのターンを行なうことができなくなるため，チャネル間の循環依存が除去されデッドロックフリーが保証される．

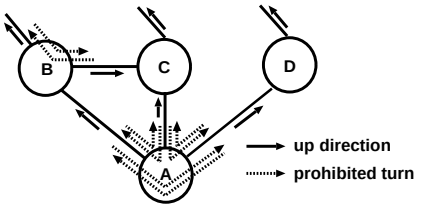

図 1 up*/down* ルーティングにおける禁止ターンのペア
Fig. 1 Pairs of prohibited turns in up*/down* routing

up*/down* ルーティングでは，up 方向から down 方向へのターンを禁止すると任意のノード間の経路が保証されないため，禁止ターンの選択に関しては自由度が無い．このため，図 1 のように，禁止ターンが形成される 2 つのリンク間において，互いに反対の方向に向かう禁止ターンのペアが必ず形成されてしまうという欠点を持つ．図 1 において，ノード B に 1 つ，ノード A には 3 つの禁止ターンのペアがそれぞれ形成されている．このような禁止ターンの偏りにより，ネットワーク内のトラフィックに偏りが生じ，ネットワークバンド幅の有効利用が困難になるという問題が発生する．

3. 2 次元 Turn モデルベースルーティングアルゴリズム

up*/down* ルーティングでは，パケット転送時に発生するターンが 2 パターンのみであるため，禁止ターンに偏りが生じやすい．そこで，我々はこの問題点を改善するために，up*/down* ルーティングの 1 次元有向グラフを拡張して 2 次元化することで，禁止ターンの偏りを減らす方式を提案した[6]．しかし，この方法は，2 次元有向グラフ上でデッドロックフリールーティングが可能な禁止ターンの選択方法のうち 1 つを示したに過ぎず，他の禁止ターンの選択方法およびルーティングアルゴリズムの可能性が残されていた．

そこで，本稿では H/V グラフと呼ばれる 2 次元有向グラフを導入し，2 次元の Turn モデルを適用してデッドロックフリーを実現するための禁止ターンの選択をシステマティックに行ない，可能なルーティングアルゴリズムをすべて導出する手法を示す．

3.1 H/V グラフの構築

3.1.1 各ノードへの depth の割当て

最初に，up*/down* ルーティングと同様に，BFS スパニングツリーを構築し，各ノードに対して *depth* を割当てる．depth は，各ノードのルートノードからの垂直方向における最短距離を示し，各チャネルの *vertical direction*(*up* および *down*) の決定に用いられる．

例として，9 ノードのイレギュラーネットワークに対する depth の割当てを 図 2(a) に示す．図 2(a) において，各リンクは互いに反対の方向を持つ 2 つの単方向チャネルから構成され，実線と破線はそれぞれスパニングツリーを構成するリンク (tree link) とそれ以外のリンク (other link) を示している．

3.1.2 各ノードへの horizontal spread および 2 次元座標の割当て

次に，2 次元有向グラフを構築するために，depth に加えて各ノードに対し *horizontal spread* を割当て，*horizontal direction*(*left* および *right*) の概念を導入する．

horizontal spread は，構築したスパニングツリー上でルートノードを起点とした前順走査を行なったときの訪問順序であり，走査における訪問順にしたがって 0 から始まる昇順の値が各ノードに割当てられる．

horizontal spread は，図 2(b) に示すように，直観的には，スパニングツリー上の水平方向における座標を表すものであり，各チャネルの horizontal direction および，同じ depth を持つノード間の vertical direction の決定に用いられる．

これにより，各ノードに対して horizontal spread (h) と depth (d) から成る一意の 2 次元座標 (h, d) を割当てることが可能となる．

例として，図 2(a) のネットワークに対する horizontal spread および 2 次元座標の割当ては，図 2(b) のようになる．

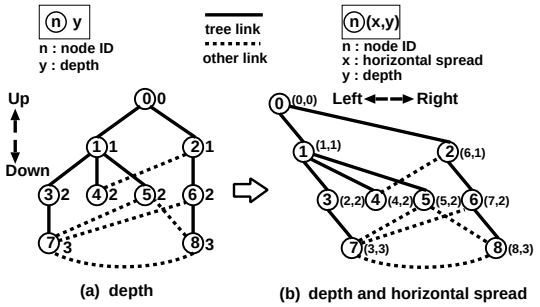

図 2 depth と horizontal spread の割当て
Fig. 2 Assignment depth and horizontal spread

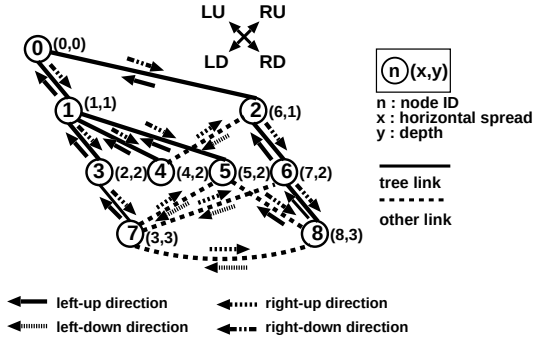

図 3 H/V グラフ
Fig. 3 H/V Graph

3.1.3 各チャネルへの方向の割当て

最後に，各ノードに割当てられた 2 次元座標を基に，各チャネルに対する horizontal direction と vertical direction の割当てを行ない，H/V グラフの構築に必要となる H/V direction の割当てを行なう．

まず，次のようにして horizontal direction を各チャネルに割当てる．座標 (x_s, y_s) から座標 (x_d, y_d) に向かうチャネルにおいて，

(1) $x_s > x_d$, ならば *left* 方向を割当て，
(2) $x_s < x_d$, ならば *right* 方向 を割当てる．

次に，同様にして vertical direction を各チャネルに割当てる．

(1) $(y_s > y_d) \vee ((y_s = y_d) \wedge (x_s < x_d))$, ならば *up* 方向を割当て，
(2) $(y_s < y_d) \vee ((y_s = y_d) \wedge (x_s > x_d))$, ならば *down* 方向を割当てる．

そして，各チャネルに対して 4 つの方向から成る H/V direction を割当てる．各チャネルの H/V direction は，horizontal direction (h) と vertical direction (v) の組み合わせ $HV(h, v)$ により次のように定める．

(1) $HV(left, up)$ に対し *left-up(LU)* 方向 を割当て，
(2) $HV(left, down)$ に対し *left-down(LD)* 方向を割当て，
(3) $HV(right, up)$ に対し *right-up(RU)* 方向 を割当て，
(4) $HV(right, down)$ に対し *right-down(RD)* 方向 を割当る．

本稿では以降 H/V direction *dir* を持つチャネルを *dir* チャネルと呼ぶ．

各チャネルに対して H/V direction が割当てられることにより，2 次元有向グラフである H/V グラフが構築される．例として，図 2 におけるネットワークの H/V グラフは，図 3 に示すものになる．

H/V グラフにおいて，スパニングツリーを構成するチャネルのみから成る部分グラフを特に，H/V ツリーと呼ぶ．

3.2 2 次元 Turn モデルベースルーティングアルゴリズムの設計と定義

H/V グラフに Turn モデルを適用し，トラフィックの分散を実現するデッドロックフリールーティングアルゴリズムを設計する方法を示す．

Turn モデルの適用手順[5] は，次のようになる．

(1) パケット転送時に形成可能なターンを列挙する．
(2) 列挙されたターンの連鎖により形成される循環構造の全パターンを列挙する．
(3) 列挙された循環構造の除去に最低限必要となる禁止ターンを選択する．

我々は，これに次の 2 つの手順を追加することにより，H/V グラフにおけるトラフィックの分散を実現する．

(1) 循環構造除去において，分散を考慮した禁止ターンの選択を行なう．
(2) 特定の循環構造を探索により検出するアルゴリズムを適用して不要な禁止ターンを除去する．

3.2.1 準　　備

最初に準備として，本節で用いられる表記を次に示す．

まず，ノード到着時のパケット転送方向 p_dir とノード通過後のパケット転送方向 n_dir により形成されるターンを T_{p_dir,n_dir} と表す．次に，ターン T_i を伴なうパケット転送後にターン T_j を伴なう転送が可能である場合のターンの連鎖を $C(T_i, T_j)$ と表す．そして，H/V グラフにおいて，$\{C(T_i, T_j) \mid j = (i+1) \bmod n, i = 0, 1, ..., n-1\}$ を形成する n 個のターンの集合 $\{T_0, T_1, ..., T_{n-1}\}$, により循環構造が形成される場合，その循環構造を $L(T_0, T_1, ..., T_{n-1})$ と表す．

3.2.2 ターンの列挙

H/V グラフにおいて，ある H/V direction へ移動した後，その他の H/V direction へ移動した際に形成可能なすべてのターンを図 4 に示す．図 4 より，H/V グラフでは 4 つの H/V direction が存在するため，形成可能なターンは全部で 12 パターンとなる．

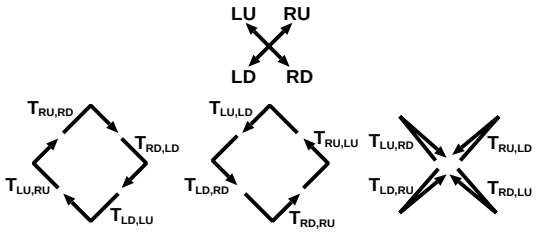

図 4 H/V グラフにおいて形成可能なターン
Fig. 4 All possible turns in H/V graph

3.2.3 循環構造の列挙と禁止ターンの選択

図 4 に示したターンの連鎖により形成されうるすべての循環構造の列挙とその除去に必要な禁止ターンの

選択を次の手順で行なう．

まず，H/V グラフ内の任意の 2 つのノードが，ツリーに属さないリンク (other link) により直接接続されている場合を考える．

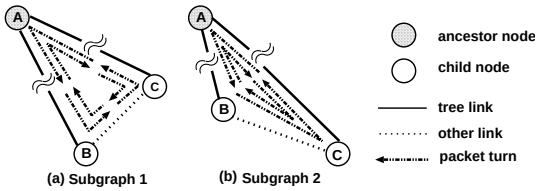

図 5 H/V グラフにおける 4 つの基本的な循環構造
Fig. 5 Four possible cycles in H/V graph

図 5 は，H/V グラフにおける 2 パターンの部分グラフを表わしており，図 5(a),(b) において，ノード B とノード C は other link により直接接続され，ノード A がそれらの祖先ノードとなっている．2 つの部分グラフの違いは，ノード B とノード C の垂直方向における相対位置の違いである．このような 2 つの子ノードは互いに，共通の祖先ノードを経由して 2 つ以上の tree link を辿ることにより到達可能であるので，このような other link が存在する場合には，これらの tree link と other link を通る 2 つの循環構造が常に形成される．図 5(a),(b) において，各部分グラフには，そのような 2 つの循環構造がそれぞれ形成されている．図 5(a) における循環構造は，循環 $L_a(T_{LU,RD}, T_{RD,RU}, T_{RU,LU})$, および $L'_a(T_{LU,RD}, T_{RD,LD}, T_{LD,LU})$ であり，図 5(b) における循環構造は，循環 $L_b(T_{LU,RD}, T_{RD,LU})$, および $L'_b(T_{LU,RD}, T_{RD,LU})$ となるが，循環 L_b と L'_b は論理的に同一である．

これらの循環構造をすべて破るために，各循環構造内の 1 つのターンを次のポリシーに基づいて禁止する．

(1) ターン $T_{LU,RD}$ を禁止しない．
(2) 可能な限り，選択した禁止ターンの組合せにより図 1 のような禁止ターンの偏りが発生しないようにする．

H/V グラフにおいて，任意のノード間の経路を保証するためには，(1) 任意のノードから LU 方向の tree channel を 0 回以上用いて任意の目的地ノードの祖先ノードに到達可能であり，かつ，(2) 祖先ノードに到達後に RD 方向の tree channel を 0 回以上用いて任意の目的地ノードに到達可能である，という 2 つの条件が必要がなので, ターン $T_{LU,RD}$ を禁止することはできない．

これらのポリシーを考慮すると，循環 L_a および L'_a を破るために禁止するターンの集合は，$\{T_{RU,LU}, T_{LD,LU}\}$ または $\{T_{RD,RU}, T_{RD,LD}\}$ となり，循環 L_b を破るための禁止ターンは $T_{RD,LU}$ となる．図 6(a),(b) に，これらの選択により禁止ターンの分散が実現されていることを示す．図 6(c) に示すように，ターン $T_{RD,LU}$ を禁止すると偏りが発生してしまうが，循環 L_b を破るためにはこれ以外に選択肢が無いので，この場合はやむをえない．

このことから，図 5 に示したすべての循環構造を破るために禁止するターン集合は，

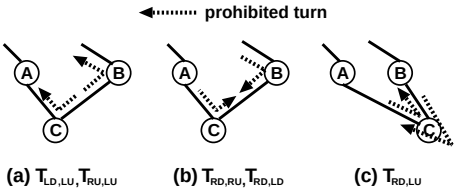

図 6 H/V グラフにおける禁止ターン
Fig. 6 Prohibited turns in H/V graph

$$P_1 = \{T_{RU,LU}, T_{LD,LU}, T_{RD,LU}\}$$
$$P_2 = \{T_{RD,RU}, T_{RD,LD}, T_{RD,LU}\}$$

のいずれかとなる．

次に，禁止ターン集合 P_1 または P_2 に属さないターンの連鎖により形成される循環構造の列挙を行なう．誌面の都合上，ここでは禁止ターン集合として P_1 を選んだ場合の手順を示すが，P_2 を選択した場合についても同様にして考えることができる．

まず，禁止ターン集合 P_1 に属するターンを含まない循環を検出するために，残りの 9 パターンのターンを $Q_1 = \{T_{LU,n\_dir} \mid n\_dir \in \{LD, RU, RD\}\}$ および $Q'_1 = \{T_{p\_dir,n\_dir} \mid p\_dir, n\_dir \in \{LD, RU, RD\}, p\_dir \neq n\_dir\}\}$ の 2 種類に分類する．

定理 1 ターン集合 Q_1 に属するターンを含む循環構造には，禁止ターン集合 P_1 に属するターンが必ず含まれる．□

証明 ターン集合 Q_1 に属するターン T_x を含み，かつ禁止ターン集合 P_1 に属するターンを含まない循環構造が形成可能であると仮定する．T_x は LU 方向 からその他の H/V direction へのターンであるので，このとき，ターン T_x の直前に連鎖して循環を形成するターンは，ターン集合 $\{T_{p\_dir,LU} \mid p\_dir \in \{LD, RU, RD\}\}$ に属するものでなければならない．しかし，このターン集合は，禁止ターン集合 P_1 と同一であるため先の仮定に矛盾する．ゆえに，ターン集合 Q_1 に属するターンを含む循環構造には，禁止ターン集合 P_1 に属するターンが必ず含まれる．□

定理 1 より，禁止ターン集合 P_1 に属する禁止ターンにより循環構造が破れるため，ターン集合 Q_1 に属するターンを禁止する必要がなくなる．これにより，LU 方向 を伴なうターンを含むすべての循環構造が除去される．このため，禁止ターン集合 P_1 を選択した場合に形成可能な循環構造は，ターン集合 Q'_1 に属する LU 方向 を伴なわないターンのみで構成されるものに絞られる．そこで，そのような循環構造を識別するために，ターン集合におけるターン間の依存関係を示す TDG(Turn Dependency Graph) を導入する．TDG D は，$D = G(V, E)$ で表わされ，V は形成可能なターン集合を表し，E は V に属する 2 つのターン間で形成可能なターンの連鎖の集合を表す．

図 7 に，ターン集合 Q'_1 における TDG を示す．

図 7 において，各頂点は，ターン集合 Q'_1 に属するターンの 1 つを表し，各頂点間を結ぶ矢印は 2 つのターン間の連鎖を表している．図 7 より，Q'_1 に属するターンによって形成されるすべての循環構造は，破線で表される 4 つの循環構造のいずれか 1 つを必ず含むことがわかる．これら 4 つの循環構造は以下の通りとなる．

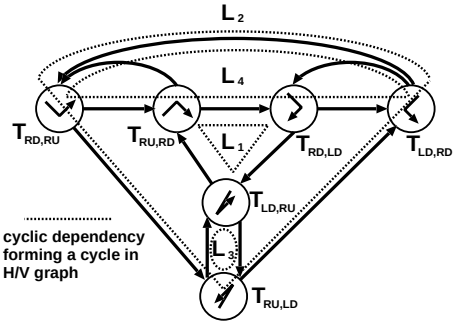

図 7 ターン集合 Q_1' における Turn Dependency Graph
Fig. 7 Turn Dependency Graph for the turn set Q_1'

(1) $L_1(T_{RU,RD}, T_{RD,LD}, T_{LD,RU})$
(2) $L_2(T_{RD,RU}, T_{RU,LD}, T_{LD,RD})$
(3) $L_3(T_{LD,RU}, T_{RU,LD})$
(4) $L_4(T_{RD,RU}, T_{RU,RD}, T_{RD,LD}, T_{LD,RD})$

先の選択ポリシーに基づいて 4 つの循環構造を破るための禁止ターン集合を選択すると，次に示す P_1' または P_1'' のいずれかとなる．

$$P_1' = \{T_{LD,RU}, T_{LD,RD}\}$$
$$P_1'' = \{T_{RU,LD}, T_{RU,RD}\}$$

図 8 に上記の 4 つの循環構造とそれらを破るための禁止ターン集合 P_1' および P_1'' に属するターンをそれぞれを示す．

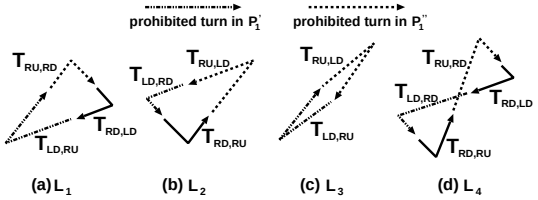

図 8 ターン集合 Q_1' により形成可能な 4 つの循環構造
Fig. 8 Four possible cycles formed by the turn set Q_1'

先に選択した禁止ターン集合 P_1 と合わせて，最終的に禁止ターン集合は，$P_1 + P_1'$ または $P_1 + P_1''$ のいずれかとなる．禁止ターン集合 P_1 により LU 方向を伴なうターンを含む循環構造が破れ，禁止ターン集合 P_1' または P_1'' によりその他のターンを含む循環構造が破れるので，H/V グラフにおいて形成可能な循環構造がすべて破れ，デッドロックフリーであることが保証される．

禁止ターン集合として P_1 の代わりに P_2 を選択した場合には，同様の手順によって禁止ターン集合 $P_2' = \{T_{LD,RU}, T_{LU,RU}\}$ または $P_2'' = \{T_{RU,LD}, T_{LU,LD}\}$ が選択され，すべての循環構造を破るために必要な禁止ターン集合は $P_2 + P_2'$ または $P_2 + P_2''$ となる．

3.2.4 循環構造検出アルゴリズムによる冗長禁止ターンの削減

誌面の都合上，ここでは禁止ターン集合として $P_1 + P_1'$ を選択した場合について述べるが，その他のターン集合を選択した場合も同様にして考えることができる．

禁止ターン集合 P_1' に属する 2 つの禁止ターンは図 8 の 4 つの循環構造を破るために必要である．しかし，これら 2 つのターンを含む循環構造には，図 9 のように，禁止ターン集合 P_1 に属するターンを一緒に含むものも存在する．

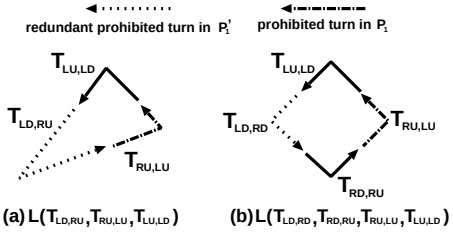

図 9 冗長な禁止ターン
Fig. 9 Redundant prohibited turns

図 9 における 2 つの循環構造には，禁止ターン集合 P_1 および禁止ターン集合 P_1' に属する禁止ターンがそれぞれ 1 つずつ含まれている，このような場合，P_1' に属するターンを禁止せずとも P_1 に属する禁止ターンにより循環構造は破れる．そのため，禁止ターン集合 P_1' に属するターンをすべて禁止してしまうと冗長な禁止ターンが発生し，ルーティングの自由度が低下してしまう．

そこで，H/V グラフにおける 図 8 の 4 つの循環構造を検出する探索アルゴリズムを提案する．そして，禁止ターン集合 P_1' に属するターンを，検出された循環構造に含まれている場合にのみ禁止することにより，冗長な禁止ターンを削減する．

循環構造の検出は，H/V グラフにおいて，次の 2 つの条件のいずれかを満たす各ノードをそれぞれ起点として深さ優先探索を行なうことにより行なわれる．

(a). 1 つ以上の RU チャネルおよび RD チャネルが接続されている (禁止ターン集合 P_1' に属するターン $T_{LD,RD}$ を形成).
(b). 2 つ以上の RU チャネルが接続されている (禁止ターン集合 P_1' に属するターン $T_{LD,RU}$ を形成).

探索において，隣接ノードの訪問に利用される出力チャネルは，次の条件を満たす場合に選択可能であるとし，利用後に利用済のマークをつける．

1. LU チャネルではない (P_1 に含まれる禁止ターンを形成しない),
2. 利用済マークがついてない,
3. 以前の探索において禁止された，ターン集合 P_1' に属するいずれかのターンを形成しない

探索の手順を次に示す．

手順 1: 条件 (a) に該当するノードを起点とした探索

まず，起点ノードから出る RD チャネルのいずれか 1 つを選び，到達可能な隣接ノードを訪問する．以降，到達先のノードにて選択可能な出力チャネルがある限り深さ優先探索により訪問を続け，訪問先のノードにて選択可能な出力チャネルが存在しなければ直前のノードに戻って探索を続ける．探索により LD チャネルを通って起点ノードに戻ってきたならば循環構造が検出されたことになり，その LD チャネルと出発時に利用した RD チャネルの間に形成されるターン $T_{LD,RD}$ を禁止する．検出される循環構造は，ターン $T_{LD,RD}$ を含み 禁止ターン集合 P_1 に属するターンを含まないので，循環 L_2 または L_4 のいずれかとなる．

探索は選択可能な出力チャネルが無くなるまで続ける．この作業は，起点ノードに存在する各 RD チャネルに対して順に行なわれる．

手順 2: 条件 (b) に該当するノードを起点とした探索

この探索は，次の点を除いて条件 (a) における探索とほぼ同様に行なわれる．

1. 起点ノードからの最初の訪問には RU チャネルを用いる．
2. 循環構造検出時には，ターン $T_{LD,RU}$ が禁止される．

この探索により検出される循環構造は，ターン $T_{LD,RU}$ を含み禁止ターン集合 P_1 に属するターンを含まないので，循環 L_1 または L_3 のいずれかとなる．

図 10 に，この探索アルゴリズムにより検出される循環構造の例を示す．

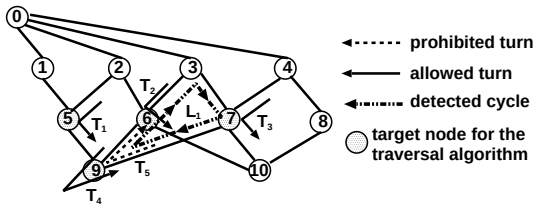

図 10 探索アルゴリズムにより検出された循環構造
Fig. 10 A detected cycle by the traversal algorithm

図 10 において，5 つのターン T_1,...,T_5 は，ターン集合 P_1' に属するターンであり，ノード 5,6,7,9 が探索の起点となっている．図 10 より，探索アルゴリズムにより検出される循環構造は，循環 L_1 だけであるため，5 つのターンのうち，循環 L_1 に含まれるターン T_5 のみを禁止すればよいことがわかる．

ノード数を n，ノードあたりのリンク数を l とすると，探索アルゴリズムの計算量は $O(n^2 * l)$ となる．

3.2.5 ルーティングアルゴリズムの定義

最後に，選択可能な 4 通りの禁止ターン集合に基づく 4 つのルーティングアルゴリズムを次のように定義する．

まず，ターン集合 P_1 を禁止する 2 つのルーティングアルゴリズムを定義する．

ターン集合 P_1 を禁止することにより，LU 方向に向かうすべてのターンが禁止されるため，LU 方向に目的地ノードがある場合には最初に LU 方向に移動する必要がある．そこで，このようなルーティングアルゴリズムをまとめて **L-turn (Left-up first turn) ルーティング**と呼び，H/V グラフにおいて，ターン集合 P_1 に属するターンをすべて禁止し，ターン集合 P_1' に属するターンのうち循環構造検出アルゴリズムにより検出された循環に含まれるものだけを禁止するルーティングアルゴリズムを **L-turn/α** と呼ぶ．同様に，ターン集合 P_1 に属するターンをすべて禁止し，ターン集合 P_1'' に属するターンのうち循環構造検出アルゴリズムにより検出された循環に含まれるものだけを禁止するルーティングアルゴリズムを **L-turn/β** と呼ぶ．なお，L-turn/α ルーティングは，過去に我々が提案した L-turn routing[6] と同じものとなる．

同様にして，ターン集合 P_2 を禁止するルーティングアルゴリズムを **R-turn(Right-down last turn) ルーティング**と呼び，ターン集合 P_2 に属するターンをすべて禁止し，ターン集合 P_2'(P_2'') に属するターンのうち循環構造検出アルゴリズムにより検出された循環に含まれるものだけを禁止するルーティングアルゴリズムをそれぞれ **R-turn/α**(**R-turn/β**) と呼ぶ．

定理 2 L-turn/α, L-turn/β および R-turn/α, R-turn/β はデッドロックフリーである □

証明 H/V グラフにて形成可能なすべての循環構造は，各ルーティングアルゴリズムにおいて選択している禁止ターン集合により破られる．ゆえに，L-turn/α, L-turn/β および R-turn/α, R-turn/β はデッドロックフリーである □

定理 3 L-turn/α, L-turn/β および R-turn/α, R-turn/β では任意のノード間の経路が保証される．□

証明 H/V ツリーに属するチャネルの方向は，LU 方向または RD 方向のみであり，RD 方向から LU 方向へのターンは禁止されているので，H/V ツリー内で形成可能なターンは $T_{LU,RD}$ のみとなる．$T_{LU,RD}$ は禁止されていないので H/V ツリーにおいては任意のノード間でのパケット転送が保証される．ゆえに，L-turn/α,L-turn/β および R-turn/α, R-turn/β では任意のノード間の経路が保証される．□

L-turn および R-turn ルーティング (以降，L/R-turn ルーティングと略す) では，禁止ターンを行なわない限り任意の非最短経路を選択することが可能である．しかし，ライブロックフリーの保証や任意の非最短経路を許した場合に発生しやすくなるホットスポット形成の防止のため，各ルーティングアルゴリズムでは，任意のノード間における選択可能な経路のうち最短となるものだけを選択するものとする．

各ルーティングアルゴリズムにおける全ノード間の経路は，H/V グラフの全ノードにおいてダイクストラのアルゴリズムを適用することにより求められる．経路探索においては，(1) 禁止されたチャネル間の移動はできない，(2) 各辺の重みはすべて等しい，という条件を守るものとしている．

最短経路が複数存在する場合には理論的にはすべての経路が選択可能でああるが，(1) 複数経路選択の可否，(2) 経路選択が行なわれるタイミング，(3) 経路選択ポリシー，などは，up*/down*ルーティングと同様に対象とするスイッチの実装に依存する．

up*/down* ルーティングが適用されているネットワーク[1)4)] に対して，理論的には，提案した各ルーティングアルゴリズムを適用することが可能である．

図 11 に，L-turn/α および up*/down* ルーティングの経路例を示す．図 11 は，16 ノード構成の H/V グラフにおけるノード 11 からノード 10 へのパケット転送において，各ルーティングアルゴリズムにより選択可能なすべての経路を示している．図 11 において，各ルーティングアルゴリズムは，共に 4 通りの経路を持つことがわかる．しかし，up*/down* ルーティングの経路はすべて 5 ホップを要し，ルートノードを必ず通らなくてはならないのに対し，L-turn/α の経路はすべて，3 ホップを要するだけであり，また，ルートノードを通る必要が無く，経路も分散されている．

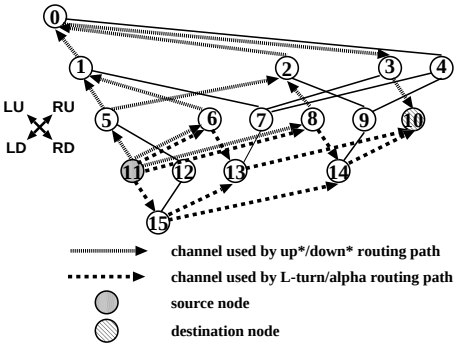

図 11 up*/down* および L-turn/α ルーティングの経路例
Fig. 11 An expmale of paths of up*/down* and L-turn/α routing

4. 関 連 研 究

本稿で提案した L/R-turn ルーティングは，2 次元有向グラフに対する Turn モデルの適用により，仮想チャネルを付加せずともトラフィックの分散を実現し，スループットを向上させている．

これに対し，複数の仮想チャネルを用いることにより上記の問題解決を図るルーティングアルゴリズムに関する研究[7)8)9)] も行なわれている．これらの手法では，仮想チャネルを用いて複数の仮想ネットワークを必要数用意し，(1)up*/down* ルーティングもしくは L/R-turn ルーティングなどの仮想チャネルを必要としないルーティングを併用し，必要に応じてパケット転送中に仮想ネットワークを切り換える[7)8)]，(2) ノード対ごとに利用する仮想ネットワークを振り分ける[9)]，などにより最短経路を実現している．これにより，パケットの衝突やネットワークバンド幅の占有時間が減少し，トラフィックの分散が実現される．

現在，InfiniBand[3)] などの仮想チャネル (レーン) の利用が可能な SAN と Myrinet[1)] のように仮想チャネルの切り替えができない SAN との両方が存在しており，仮想チャネルの有無に依存しない L/R-turn ルーティングは，両者のルーティングに適用可能な数少ない研究であるといえる．

5. 評　　価

シミュレーションによる各 L/R-turn ルーティングおよび比較のための up*/down* ルーティングの評価を以下に示す．

5.1 評価用シミュレータ

我々は，評価用のフリットレベルシミュレータを，C++ 言語を用いて実装した．このシミュレータは，任意のトポロジにより結合されたスイッチおよびそれに接続されるホスト PC 間のパケット転送動作をフリットレベルでシミュレートする．シミュレータ上では，スイッチは，チャネルバッファ，クロスバ，リンクコントローラ，制御回路により構成され，パケットスイッチング方式，ルーティングアルゴリズム，ポート数などを指定できる．また，ホスト PC は，パケットの生成および送受信を行ない，パケットサイズ，トラフィックパターンなどを指定できる．

5.2 評 価 条 件

評価は，まず 64 ノードから成るイレギュラーネットワークを対象として行なった．このイレギュラーネットワークは，異なる 10 パターンのトポロジを，同一ノード間にリンクを 2 本以上接続しない，という制約を課した上でランダムに生成したものである．

一方，実際の SAN では規則性や階層性がある程度見られるため，レギュラーネットワークにおける評価も重要となる．最近は Myrinet-Clos[10)] などの多数のリンクと中間スイッチを用いる階層接続網が用いられる傾向にあるが，これらのコストの大きい結合網では，up*/down*ルーティングと L/R-turn ルーティングの双方でルーティングは最適となり，差は生じない．そこで，より低コストの実装時に用いられる 8×8 の 2 次元トーラスを，レギュラーネットワークの代表として評価対象とした．

各トポロジにおけるルートノードは，crossing path ☆が最小となるものを選択し．crossing path が等しい場合には，各ノード間の最短距離の平均が最小であるものを選択した．各ノードは，8 ポートを持つ 1 つのスイッチとそれに接続される 4 つのホスト PC から成り，スイッチの残りの 4 ポートは他のスイッチとの接続に用いられる．

各ルーティングアルゴリズムは，分散ルーティング方式で実装し，選択可能な経路のうちホップ数が最小となる経路のみを選択可能としている．また，経路上の各スイッチにおいて，選択可能な出力ポートが複数存在する場合には，ポート番号が最小となるものを選択するようにしている[4)]．

各シミュレーションは $1,000,000$ クロック実行され，最初の $50,000$ クロックを評価の対象外とした．パケット長は 128 フリットで，トラフィックパターンには *uniform* と非一様である *bit-reversal* をそれぞれ用いた．スイッチング方式はバーチャルカットスルーを用い，スイッチ間のフリット転送には，3 クロックを要するものとした．

5.3 イレギュラーネットワークの評価

表 1 10 パターンの 64 ノードイレギュラーネットワークにおける各性能値の平均値
Table 1 The average value of the performance metrics on 10 irregular networks with 64 nodes

	PT(SDPT)	PPT	スループット	
			Uni.	Bit.
up*/down*	3.019(3.669)	96.6	0.0452	0.0477
L-turn/α	2.875(2.225)	18.9	0.0576	0.0707
L-turn/β	2.990(2.269)	20.6	0.0572	0.0711
R-turn/α	2.882(2.238)	19.4	0.0468	0.0485
R-turn/β	2.900(2.272)	20.8	0.0471	0.0472

表 1 は，10 パターンの 64 ノードイレギュラーネットワークにおける各ルーティングアルゴリズムの，(1) ノードあたりの禁止ターン数 (*PT*) およびその標準偏差 (*SDPT*)，(2) ペア禁止ターン数 (*PPT*)，(3) uniform および bit-reversal トラフィックにおけるスループット，の平均値をそれぞれ表している．*SDPT* と *PPT* は，ネットワークにおける禁止ターンの分散の度合であり，数値が小さいほどより均等に分散していることを示す．スループットは，受信トラフィック (フリット/クロック/ホスト *PC*) の最大値であり，全ホスト PC が毎クロックに 1 フリット受信する場合を 1.00 としている．

☆ 1 つのチャネルを通るノード対の経路数の最大値を示す

表 1 より，L/R-turn ルーティングは，up*/down* ルーティングよりも小さな PT, $SDPT$ および PPT を実現している．特に PPT は，up/down* ルーティングの約 20% にまで減少している．これより，L/R-turn ルーティングでは，より均等な禁止ターンの分散と禁止ターン数の削減を実現していることがわかる．

次に，各トラフィックにおけるスループットを見ると，up*/down* ルーティングに比べて各 L-turn ルーティングの性能向上が大きく，uniform トラフィックにおいて約 28%，bit-reversal トラフィックにおいて約 49% の性能向上を達成している．一方，禁止ターン分散の度合はほぼ同じであるにもかかわらず，各 R-turn ルーティングのスループットは L-turn ルーティングに比べて劣り，up*/down* ルーティングと同程度となっている．この理由として，R-turn ルーティングにおける次の 2 つの要因が考えられる．(1) ルートノードに近づく LU 方向へのターンが，ターン $T_{RD,LU}$ を除いてすべて許可されている，(2) ルートノードから離れる RD 方向から他の方向へのターンが全て禁止されている．この 2 点より，R-turn ルーティングではルートノード方向へトラフィックが集中しやすくなり，スループットが低くなったものと考えられる．

5.4 2 次元トーラスの評価

表 2 8 × 8 2 次元トーラスにおける各性能値
Table 2 The performance metrics on 8x8 2D Torus

	PT(SDPT)	PPT	スループット	
			Uni.	Bit.
up*/down*	2.500(2.264)	80	0.0455	0.0414
L-turn/α	2.640(1.789)	17	0.0771	0.0812
L-turn/β	2.625(2.012)	20	0.0691	0.0800
R-turn/α	2.641(1.789)	17	0.0507	0.0513
R-turn/β	2.625(2.012)	20	0.0491	0.0513

表 2 は，8 × 8 2 次元トーラス上の各トラフィックにおける各ルーティングアルゴリズムの，(1)PT および $SDPT$，(2)PPT，(3) 2 つのトラフィックにおけるスループット，をそれぞれ表している．

表 2 より，L/R-turn ルーティングの PT は，up*/down* ルーティングに比べて大きいものの，$SDPT$ および PPT は，イレギュラーネットワークの場合と同様に小さくなっている．これより，L/R-turn ルーティングは，2 次元トーラスにおいても，より均等な禁止ターンの分散を実現していることがわかる．

次に，スループットを見ると，2 つの L-turn ルーティングがその他のルーティングアルゴリズムよりも大きなスループットを達成していることがわかる．特に，L-turn/α は，up*/down* ルーティングに対し，uniform トラフィックにおいて約 70%，bit-reversal トラフィックにおいて約 96% の性能向上を達成している．一方，2 つの R-turn ルーティングのスループットは，up*/down* ルーティングと比べると bit-reversal においては約 24% の性能向上を達成しているが，L-turn ルーティング に比べると劣っている．

以上の結果より，L-turn ルーティングにおいて，禁止ターンの分散を実現することにより高スループットが達成されることが確認された．

6. ま と め

本稿では，up*/down* ルーティングにおいて禁止ターンが偏る問題を改善するために，2 次元有向グラフである H/V グラフを構築し，Turn モデルを適用して禁止ターンの分散を実現するデッドロックフリールーティングアルゴリズムを設計する方法を示した．

提案した方法により，形成可能なデッドロックフリールーティングアルゴリズムが全部で 4 つ存在することが明らかになった．これらのルーティングアルゴリズムは，それぞれ 2 つの L-turn ルーティングと R-turn ルーティングに分類され，分散を考慮した禁止ターンの選択と循環構造検出アルゴリズムの適用により，禁止ターンの分散と不要な禁止ターンの除去を実現した．

シミュレーションの結果，L-turn ルーティングおよび R-turn ルーティングは，禁止ターンの分散を実現し，up*/down* ルーティングよりも高いスループットを示すことがわかった．特に，L-turn ルーティングは，最大で 96%のスループット向上を示すことがわかった．

参 考 文 献

1) N.J.Boden et al.: Myrinet: A Gigabit-per-Second Local Area Network, *IEEE Micro*, Vol. 15, No. 1, pp. 29–35 (1995).
2) T.Kudoh and et.al: RHiNET: A network for high performance parallel computing using locally distributed computing, *Proc. IWIA*, pp. 69–73 (1999).
3) I.T.Association: InfiniBand architecture. Specification, *available from the InfiniBand Trade Association* (2001).
4) M. D. Schroeder and et al.: Autonet: A high-speed, selfconfiguring local area network using point-to-point links, *Technical Report SRC research report 59,DEC* (1990).
5) C.J.Glass and L.M.Ni: Maximally Fully Adaptive Routing in 2D Meshes, *Proc. ISCA*, pp. 278–287 (1992).
6) 鯉渕 道紘, 舟橋 啓 , 上樂 明也, 天野 英晴 : L-turn Routing: Irregular Network における Adaprive Routing, 情処論文誌 HPS, Vol. 43, No. Sig 9(Hps3), pp. 119–134 (2001).
7) F.Silla and J.Duato: High-Performance Routing in Networks of Workstations with Irregular Toporogy, *IEEE Trans. on PDS*, Vol. 11, No. 7, pp. 699–719 (2000).
8) M.Koibuchi and et.al: Deterministic Routing Techniques by Dividing into Sub-Networks in Irregular Networks, *the IASTED International Conference on NPDPA*, pp. 143–148 (2002).
9) T.Skeie, O.Lysne and I.Theiss: Layered Shortest Path (LASH) Routing in Irregular System Area Networks, *Proc. IPDPS*, pp. 162–169 (2002).
10) C.L.Seitz: Recent Advances in Cluster Networks, *available from the Myricom, Inc.* (2001).