耐故障性を考慮した k-ary n-cube 用適応デッドロック回復ルーティング

吉永 努[†] 細越洋行[†,☆] 曽和将容[†]

並列分散計算機用の k-ary n-cube ネットワークを対象として，故障チャネル/ノードへの耐性を考慮した適応型デッドロック回復ルーティングについて述べる．k-ary n-cube の規則性を活用したルーティングを不定型ネットワークにも対応可能にすることにより，任意の故障チャネルに対してデッドロックフリーなルーティングアルゴリズムを提案する．このアルゴリズムは，物理チャネルあたり最低 2 本の仮想チャネルを使用して構築でき，通信経路が故障チャネルと関係ない場合の完全適応ルーティングと故障チャネル付近での最短経路の迂回をサポートすることを特徴とする．ハードウェア記述言語を使用した 2 次元トーラス用ルータの設計と $k = 10$ における通信シミュレーション結果を示し，提案するルーティングアルゴリズムの有効性について議論する．

Fault-Tolerant Adaptive Deadlock-Recovery Routing for k-ary n-cube Networks

Tsutomu Yoshinaga,[†] Hiroyuki Hosogoshi[†,☆] and Masahiro Sowa[†]

This paper describes a fault-tolerant, adaptive deadlock-recovery routing algorithm for k-ary n-cube networks of parallel and distributed computers. We integrate regular and irregular network routing algorithms in order to tolerate arbitrary number/shape of channel and node faults with guaranteeing deadlock freedom. Proposing algorithm can be implemented with two virtual channels per physical channel in a minimum case. It supports fully adaptive routing on a network that does not include faulty channels and provides minimal misrouting paths around faults. We show a router design for 2D torus and communication performance in a case of $k = 10$, then discuss its validity with comparing to several other algorithms.

1. はじめに

近年の高性能計算を目的とする並列分散計算機には，数千の計算ノードをネットワーク結合するものも登場し，大規模化が進んでいる．このような超並列計算機においては，一部のネットワーク故障がシステム全体の運用に支障をきたさない耐故障性が重要となる．QsNet [14] や Myrinet [13] は複数パスルーティングをサポートしており，故障に関係する経路を選択しないようにする耐故障性を有する．これらのクラスタ用ネットワークでは，k-ary n-tree のような木構造トポロジが用いられることが多く，その場合複数ある最短経路から故障チャネルのみを使用不可に設定すればよく，故障部位の遠回りによる迂回は必要ない．

k-ary n-cube ネットワークは，各次元方向に k ノードを持つ n 次元のメッシュ/トーラスの総称であり，超並列計算機の相互結合網として広く利用されてきた．これらの中には，Alpha 21364 [12] や BlueGene/L [1] のように適応ルーティングを採用するものもあるが，その多くが Duato のプロトコル[9] に基づく最短経路選択を行うものであり，ネットワークの耐故障性をサポートするものは少ない．k-ary n-cube で耐故障性を持たせるためには，故障チャネルの迂回が必要になり，最短経路のみでルーティングを完了することはできない．これまで，ターンモデル[10] に基づく非最短経路ルーティングや仮想ネットワークを導入してブロック故障をデッドロックせずに迂回するアルゴリズム[4] などが提案されている．これらのほとんどが，基本的に k-ary n-cube の定型性に基づくデッドロック回避型のアルゴリズムである．そのため，少ない仮想ネットワークしかサポートしないと耐故障能力に乏しく，柔

† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications
☆ 現在，キヤノン株式会社
Presently with Canon Inc.

軟性を上げるためには多くの仮想チャネルを装備しなければならない．また，故障ブロック形状の制限から故障していないノード/チャネルが利用不可能になったり，最短となる迂回経路の選択が困難になったりする場合がある．

本論文では，不定型ネットワーク用のデッドロック回復に専用の仮想チャネルを設け，定型ネットワークルーティングに基づく適応ルーティング用の仮想チャネルと組み合わせることで故障部位の回避をサポートする耐故障ルーティングアルゴリズムを提案する．具体的には，デッドロック回復用と故障部位回避用の 2 種類のルーティング表を導入することにより，k-ary n-cube の次元数 n やノード数 k^n に非依存な適応型の耐故障ルーティングアルゴリズムが物理チャネルあたり最低 2 本の仮想チャネルで構成できることを示す．また，2 次元トーラス用ルータのハードウェア記述言語による設計と $k = 10$ のときのシミュレーション結果を 3 つのデッドロック回避型ルータと比較し，ハードウェアコストや通信性能について議論する．

2. ルーティングアルゴリズム

2.1 耐故障ルーティング

我々のルーティングアルゴリズムは，ネットワークを構成するチャネルとノードの故障に対する耐性を持つものを対象とする．このうちノード故障については，その故障ノードに接続したすべてのチャネルを利用不可にすることで対応する．ネットワークの耐故障性は，故障の検知とネットワークの再構成によって実現される．故障の検知は，一般にチャネルの信号レベルでの活線判定や，隣接ノード間でのメッセージ受信応答確認または定期的なメッセージ交換によって行われる．ネットワークの再構成は，検知した故障チャネルやノードを取り除き，部分的に k-ary n-cube トポロジの定型性を失ったネットワークでデッドロックすることなしに通信処理が可能なように設定し直すことである．同様に，故障チャネルの復旧を検知した場合にもネットワークの再構成を行うものとする．

ブロック故障モデルでは，発生位置が未知である複数の故障に対応するために，故障形状を定型トポロジのルーティングに合致するものとする必要がある．このため，故障ブロックに無故障ノードが含まれてしまったり，故障ブロック自体の認識が複雑化してしまったりする[20]．これに対して，任意の故障チャネル/ノードを取り除いたネットワークを不定型トポロジととらえ，それに対応するルーティングアルゴリズムをサポートすれば，故障領域のブロック化は不要となり無故障ノードを無駄にすることもない．この場合，元々のネットワークの定型性と故障による不定型性に対するルーティングをいかに効率的に組み合わせるかが鍵となる．不定型トポロジ用のアルゴリズムでは，ネットワーク再構成フェーズにおいて故障チャネルを除いた有効なトポロジを反映するルーティング表を各ノードで計算することが必要となる．定型ネットワーク用の耐故障ルーティングには，大域的な故障情報を利用しないアルゴリズムも存在するが，ノード故障を扱うためには故障ノード情報を全ノードが知る方が安全である．なぜならば，故障ノード宛のメッセージを出力する無故障ノードがあると不都合が生じると考えられるためである．

不定型ネットワーク用のアルゴリズムとしては，up*/down*ルーティング[18] や L-Turn ルーティング[11] のように仮想チャネルなしに実装可能なものが提案されており，比較的低コストで定型トポロジ用のルーティングと組み合わせることが可能である．本論文では，up*/down*ルーティングによるデッドロック回復をサポートした Detour-UD ルーティングを提案し，その実現方法に関して述べる．Anjan らによって提案されたデッドロック回復方式のルーティングアルゴリズム Disha[2),3)] は，デッドロック回避方式に比べて仮想チャネルに対する制約が少ないというメリットを有するが，これまで耐故障性については十分に検討されてこなかった．我々が Detour-UD で主張する新規性は，Disha と異なり故障情報に基づいて定型および不定型ネットワーク用の 2 つのルーティング表を導入すること，それにより柔軟な故障部位の迂回を可能とすることである．

なお，故障ノードによって失われるタスクや動的チャネル故障によって壊れたメッセージは，上位のソフトウェアによって回復されることを前提とし，Detour-UD では扱わない．また，多くのブロック故障モデルにおける仮定と同様に，故障チャネル/ノードは隣接した無故障ノードによって認識され，無故障チャネルを通じて故障情報の交換ができるものとする．

2.2 Detour-UD による適応ルーティング

図 1 に Detour-UD ルーティングアルゴリズムを示す．Detour-UD ルーティングでは，ネットワークを故障領域と無故障領域に分割したルーティングを行う．故障領域は故障チャネルから一定距離以内の領域とし，それ以外の領域は無故障領域と定義する．**図 2** に，k-ary 2-cube の例を示す．ここでノード A と B 間のチャネルが故障していると仮定する．故障領域距離を 1 とした場合，故障チャネルに接続した 2 ノード

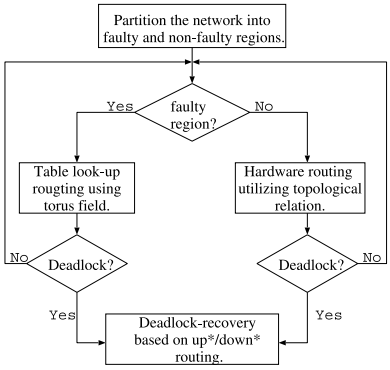


図 1 Detour-UD ルーティングアルゴリズムの流れ図
Fig. 1 A flow chart of the Detour-UD routing algorithm.

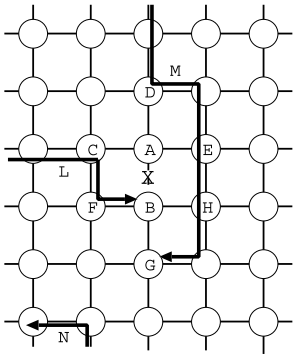


図 2 迂回ルーティング例
Fig. 2 An example of misrouting.

A，B が故障領域となる．また，故障領域距離を 2 とした場合には，A，B に加えて C〜H の 6 ノードも故障領域とする．

無故障領域のメッセージは，k-ary n-cube の定型性に基づいて最短経路を完全適応ルーティングする．ネットワークトポロジの定型性を前提としたルーティングでは，メッセージの宛先ノード番地と現在のノード番地の相対関係を元に出力チャネルをハードウェアで決定する．一方故障領域のメッセージも現在位置から宛先ノードまでの最短経路を適応ルーティング可能にするため，k-ary n-cube から故障チャネルを除いたルーティング表を利用する．無故障領域または故障領域のメッセージについて，デッドロックを検出した場合にはじめて up*/down*ルーティングによるデッドロック回復を行う．一度デッドロック回復を始めたメッセージは宛先ノードまで up*/down*ルーティングによって配送する．デッドロック回復経路もルーティング表で管理する．

図 2 の例では，無故障領域のメッセージ N は初めルーティング表を参照せずに最短経路を適応ルーティングし，デッドロックを検出したらルーティング表を参照してデッドロック回復を行う．また，メッセージ L と M は故障領域に到達した時点からルーティング表に基づいた故障迂回用の適応ルーティングを開始する．すなわち，ノード B を宛先とするメッセージ L がノード C に到達すると，ルータ C は最短経路となるノード F への出力を指示する．同様にノード G を宛先とするメッセージ M は故障領域のルータ D または A で左右いずれかへの迂回を行う．このように，ルーティング表によって複数の出力チャネル候補が示されていれば故障領域でも適応ルーティングを行ってよい．またメッセージが故障領域から無故障領域に出れば，ルーティング表参照を行わない適応ルーティングを再開する．

一般に，ルーティング表の参照による経路選択は，それを必要としないハードウェア経路選択よりもルーティング表のアクセス調停などによって 1 ホップあたりの時間が長くなりがちである．そこで，Detour-UD ではルーティング表の参照をデッドロック回復と故障領域内のメッセージに限定する．そうすることで，無故障領域での通信遅延を抑えながら故障領域での効率的な迂回を実現する．

2.3 ルーティング表による経路選択

ルーティング表は，宛先ノード番地に対してメッセージをどのチャネルに出力すべきかの対応表である．適応型デッドロック回復ルーティングでは，デッドロックが検出されるまでは自由にメッセージの経路選択を行ってよく，潜在的なデッドロック検出後にデッドロック回復を行う．そこで，Detour-UD ルーティングではデッドロック前に参照する k-ary n-cube 用のルーティング表とデッドロック回復で参照する up*/down* ルーティング用の 2 つを使用する．

図 3 に，図 2 におけるノード E のルーティング表の一部を示す．宛先ノード番地 A〜H に対して torus 表と up/down 表の 2 つを持ち，それぞれネットワークを 2 次元トーラスと幅優先探索（以降 BFS と略す）スパニングツリー[18] と考えた場合の北東西南順での最短経路上の出力チャネル候補を 1 にセットしている．**図 4** にノード C を根とした場合の BFS スパニングツリーの一部を示す．たとえば，図 2 でノード A と C は E から見て西方向にあるので，torus 表は 0010（西出力チャネルのみ 1）となっている．図 4 からも，ノード E から（西出力チャネルを使用して）A と C は up 方向のみで到達可能であり up/down 表の内容も同様に 0010 となる．これに対して，ノード E から B の 2 次元トーラス上での最短経路は故障チャネルの

	torus	up/down
A:	0010	0010
B:	0001	0010
C:	0010	0010
D:	1010	0010
E:	0000	0000
F:	0011	0010
G:	0001	0001

図 3 ルーティング表（ノード E）
Fig. 3 The routing table on node E.

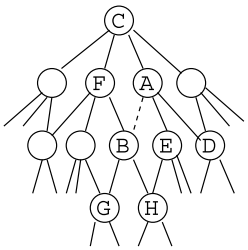


図 4 BFS スパニングツリー
Fig. 4 A BFS spanning tree.

関係でノード H（南）方向に限られるため B 番地の torus 表は 0001 となるが，図 4 の BFS スパニングツリーから E-H-B の経路は down 後 up が必要な禁止された経路であるため，E から A 経由で C まで up 転送してから F 経由で down することになる．したがって B 番地の up/down 表は 0010（西方向）を示している．

2.4 仮想チャネルの使用法

Detour-UD ルーティングでは，k-ary n-cube ネットワークの物理チャネルあたり 2 本以上の仮想チャネルを使用する．このうち 1 本の仮想チャネルは up*/down* ルーティングによるデッドロック回復の専用とし UD-VC と呼ぶことにする．残りの仮想チャネルは完全適応ルーティングに自由に使用でき FA-VC と呼ぶ．無故障領域のメッセージは FA-VC を優先して使用することで，適応ルーティングによる輻輳の回避が可能となる．故障領域のメッセージも torus ルーティング表に基づいて FA-VC を優先して使用することで，ネットワークの定型性を利用した最短経路による故障チャネルの迂回と適応ルーティングの柔軟性を活用できる．

FA-VC を使用した適応ルーティングでは，デッドロック検出をサポートする必要がある．これは，FA-VC を使用して転送中のメッセージのブロッキング時間のタイムアウト機能などにより実装する．デッドロック検出されたメッセージは，up/down ルーティング表を参照してデッドロック回復経路を決定し，宛先ノードまで UD-VC を使用して転送する．up*/down* ルーティングは任意のトポロジに対して非サイクリックな経路の存在を保証しており，どのノードからデッドロック回復を開始してもかまわない．デッドロック回復経路を UD-VC で提供することにより，ワームホール方式を含む種々のフォロー制御方式で並列デッドロック回復可能である．

動的な故障に対応するためには，ルーティング表とネットワークトポロジの一時的な不整合への対策が必要となる．Detour-UD は UD-VC によるデッドロック回復に依存するため，UD-VC 自体でのデッドロックは防止しなければならない．これには，ネットワーク再構成フェーズにおいて，デッドロック回復中のメッセージを破棄/再送によって回復することや，UD-VC を多重化する方法が考えられる．

Detour-UD の仮想チャネルに対する要件は，従来の適応型デッドロック回避方式ルーティングが 4 本以上を必要とするのに対して低コストである[4),5),7),19)]．一般に，デッドロック回避方式では仮想チャネルの使用条件を厳しく分類することでデッドロック自体の発生を防止する．これに対して，デッドロック回復方式ではデッドロック回復専用の仮想チャネルが存在すれば，残りの仮想チャネルの使用に制限はない．複数の FA-VC を実装した場合には，メッセージがそれらを制約なしに使用することができるため，デッドロック回避方式に比べてルーティングの自由度を高くできる．

2.5 Detour-UD の k-ary n-cube での正当性

k-ary n-cube に対する Detour-UD ルーティングアルゴリズムが，物理チャネルあたり 2 本の仮想チャネルで構成できることを述べる．

定義 1：相互結合網 k-ary n-cube を $I = (N, C)$ と表す．ただし N と C は，それぞれ I を構成するノード，およびチャネル集合とする．

定義 2：I から故障ノードと故障チャネルを除き，連結した $I' = (N', C')$ を作成する（ネットワーク再構成）．ただし $N' \subseteq N$，$C' \subseteq C$ とする．

定義 3：I' を故障領域 I'_f と無故障領域 I'_n に分割する．ただし，$I' = I'_f \cup I'_n = (N'_f, C'_f) \cup (N'_n, C'_n)$，$N'_f \cap N'_n = C'_f \cap C'_n = \emptyset$ とする．

定義 4：物理チャネルあたり FA-VC と UD-VC の 2 本の仮想チャネルを利用して $C' = C_{fa} \cup C_{ud}$ と分割する．ただし，C_{fa} と C_{ud} はそれぞれ FA-VC，UD-VC の仮想チャネル集合であり $C_{fa} \cap C_{ud} = \emptyset$ である．

定義 5：I' に対する Detour-UD のルーティング関数を $R = R_1 \cup R_2 \cup R_3$ とする．ここで，R_1 は I'_n 上の最短経路完全適応ルーティング関数であり，x を送信元，y を宛先，p を宛先 1 ホップ手前のノード，$c_{j=0..i}$ を経路上の仮想チャネルとする

と，$R_1(x,y,p) = (c_0,c_1,...,c_i), \forall x,p \in N'_n, \forall y \in N', c_{j=0..i} \in C_{fa}$ と定義する．R_2 は I'_f 上の非最短経路部分適応ルーティング関数であり，$R_2(x,y,p) = (c_0,c_1,...,c_k), \forall x,p \in N'_f, \forall y \in N', c_{j=0..k} \in C_{fa}$ となる．また，R_3 は I' 上の up*/down*ルーティング関数であり，$R_3(x,y) = (c_0,c_1,...,c_l), \forall x,y \in N', c_{j=0..l} \in C_{ud}$ とする．

仮定 1：ネットワーク再構成処理時に，ルーティング関数 R_3 により仮想チャネル UD-VC を使用してデッドロック回復中のメッセージは破棄する．

補題 1：R_3 は連結，かつデッドロックフリーである．

証明：定義 5 から，R_3 は I' 上の任意のノード対 (x,y) 間の通信経路を仮想チャネル部分集合 C_{ud} により提供するため連結である．また，R_3 で提供される通信経路は C_{ud} に属する仮想チャネルのみで構成され，C_{fa} に属する仮想チャネルへの依存を持たない．したがって，C_{ud} のみで形成される R_3 のチャネル依存グラフがサイクルを持たないことを示せばよい．ここで，up*/down*ルーティング関数は任意のネットワークトポロジに対して非サイクリックな経路しか許さないため，k-ary n-cube 上の R_3 も次数 k や次元数 n に依存せずに仮定 1 のもとでデッドロックフリーとなる．

定理 1：Detour-UD は，k-ary n-cube の次数 k や次元数 n に依存せずに物理チャネルあたり FA-VC と UD-VC の最少 2 本の仮想チャネルでデッドロックフリー，かつネットワークの無故障領域では完全適応ルーティング可能である．

証明：補題 1 より，Detour-UD のルーティング関数 $R(x,y)$ には，連結でデッドロックフリーなルーティングサブ関数 $R_3(x,y) = R(x,y) \cap C_{ud}$ が存在する．このことは，I' 上で転送中のメッセージを任意のノードから R_3 によってデッドロック回復可能なことを示している．したがって，故障領域内外でルーティングサブ関数 R_1 と R_2 を併用することにかかわらず $R(x,y)$ 自体がデッドロックフリーとなる．また，定義 4 と 5 から，R_1 はネットワークの次数 k や次元数 n に依存せずに物理チャネルあたり 1 本の FA-VC を装備することにより，無故障領域での完全適応ルーティング機能を提供する．

3. ネットワーク再構成

ネットワーク再構成処理では，故障または復旧チャネル/ノードの位置に基づいて新しい BFS スパニングツリーの根ノードを決定し，各ノードのルーティング表を更新する．各ノードは，故障情報と根ノード ID が分かればダイクストラのアルゴリズムなどを用いて自身のルーティング表を計算することができる．

Autonet は，ネットワーク再構成時にすべての通信中のメッセージを破棄する手法を実装している[17),18)]．これに対して，Detour-UD ではネットワーク再構成時にデッドロック回復中のメッセージのみ破棄する手法を提案する．それにより，破棄/再送を必要とするメッセージ数を少なく抑えることが期待できる．

根ノードの決定には，2 通りの方法が考えられる．1 つは静的に根ノード候補の優先順位を決めておき，無故障ノードの中から 1 つを選択する方法である．故障発生位置が根ノードと離れている場合，既存の根ノードを再利用すればルーティング表更新への影響を小さくすることができる．また，故障チャネル/ノードの復旧を検出した場合にも根ノードを変える必要はない．2 つ目は，故障領域の 1 つのノードを根としてネットワークを再構成する方法である．こちらは，故障領域から根ノード候補へネットワーク再構成を依頼するメッセージの到達性が問題となる場合に有効である．いずれにしても，既存の根ノードが故障した場合には新しい根ノードを決定する必要がある．

3.1 ネットワーク再構成手続き

はじめに，単一の故障領域に対するネットワーク再構成手順について考察する．なお，ネットワーク再構成処理中に新たな故障が発生することは考えないものとする[15)]．

(1) 故障チャネル/ノードを検出すると，周りの無故障ノード間で隣接通信を行い，故障領域を特定する．その通信手順は，長方形故障ブロックモデルに準ずるものでよい[4)]．

(2) 故障領域内のノードはデッドロック回復以外の通常の通信処理を中断し，1 つの正常なノードを根ノード候補とする．**図 5** の例では，故障領域の左上頂点に相当するノードを根としている．

(3) 根ノード候補は自身のルーティング表を計算し，up/down ルーティング表の down チャネル方向に新しい根ノード ID と故障位置情報を引数としてネットワーク再構成メッセージをブロードキャストする．

(4) 根ノードからネットワーク再構成メッセージを受信したノードは，通常の通信処理を一時停止して各ノードのルーティング表を計算する．

(5) 新しい BFS スパニングツリーで葉ノード（up チャネルのみを持つノード）までルーティング表を更新すればネットワーク再構成が完了するので，up 方向に通信再開を通知する．

次に，2 つ以上の独立した故障領域が別々の根ノー

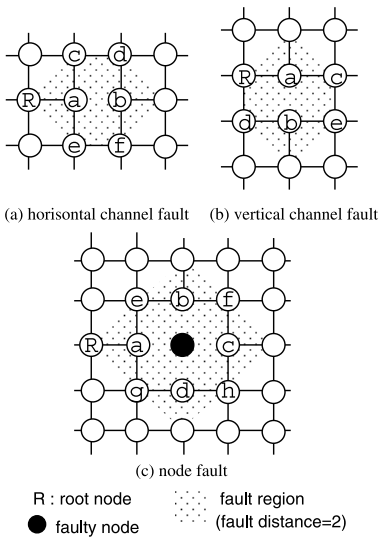


図 5 BFS スパニングツリーの根ノード
Fig. 5 Location of a root node for the BFS spanning tree.

ド候補によってネットワーク再構成を行う場合を考える．最も簡単な方法は，全ノードのルーティング表更新の同期をとることである．すなわち，上記手順 (5) を up 方向のルーティング表更新の完了通知とし，根ノードでリダクション処理を行ってから改めて通信再開を down 方向にブロードキャストすればよい．いずれかのノードが同期をとる前に複数のネットワーク再構成メッセージを受信した場合には，優先順位の高い一方の根ノード候補に新しい故障リストを送ってネットワーク再構成をやり直し，もう一方の根ノード候補による再構成処理はキャンセルするようにする．

3.2 ネットワーク再構成手続きの妥当性

前節に示した手続きにより，ネットワーク再構成が行えることをルーティング表の更新前と新旧混在時点での通信に着目して考察する．また，Detour-UD のネットワーク再構成で特徴的な 2.5 節で示した仮定 1 の妥当性についても述べる．

(1) ルーティング表の更新前の通信

故障領域の特定は，その故障に対応したルーティング表への更新前であっても無故障チャネルを使用した隣接ノード間通信を繰り返すことで実現できる．たとえば，図 5 (a) の例では，ノード a-b 間のチャネルが故障するとこれらのノード間での直接的な最短経路通信は行えないが，隣接ノード c〜f を介して検出した故障情報のやりとりは可能である．この隣接ノード間通信のためには，最低限 UD-VC が利用できればよい．ここで，ネットワーク再構成処理開始前に UD-VC を使用してデッドロック回復を始めたメッセージは，故障チャネルによっていずれかのデッドロック回復メッセージがブロックされない限り，それ自体でデッドロックすることはない．故障チャネルによってブロックされたデッドロック回復中のメッセージ D を保持するルータは，故障チャネルを検出したルータにほかならないので，それ自身がネットワーク再構成モードになる．したがって，D を破棄することによって UD-VC を空にすれば隣接ノード間での通信路は確保される．また，隣接ノード間通信は up または down 方向に 1 ホップするだけであるため，up*/down*ルーティングの制約に違反しない．

(2) 新旧ルーティング表混在状況化での通信

故障情報の共有後，仮決めした根ノード候補は，故障チャネルを除いた新しいルーティング表に従ってネットワーク再構成メッセージをブロードキャストする．このネットワーク再構成メッセージの通信路についても，(1) に述べたと同様の理由で最低限 UD-VC は確保できれる．このとき，受信側ノードは古いルーティング表を持っているが，それぞれネットワーク再構成を行ってルーティング表を更新した後に葉ノード方向にネットワーク再構成メッセージをフォーワードすれば，その到達性に問題は発生しない．

なお，故障チャネル/ノードが復旧した場合については，ルーティング表が更新されるまで復旧したチャネルを使用しないだけであり通信に問題はない．

(3) 仮定 1 の妥当性

ネットワーク再構成処理時に，破棄するメッセージをデッドロック回復中のメッセージに限定することは，デッドロック回復以外の通常のメッセージがネットワーク再構成処理中そして処理後に FA-VC に残っていてもよいことを意味する．Detour-UD において FA-VC を使用するルーティング関数 R_1 と R_2 は，ともにネットワークの定型性を前提とした経路選択を行うが，ネットワーク再構成処理によって仮に中継ノードから宛先までの経路が変更されたりなくなったとしても，デッドロック回復によって宛先までの到達性が保証される．

4. 2D トーラス用ルータ

我々は，ルーティングアルゴリズムによるハードウェアコストと通信性能を比較する目的で，Detour-UD を含む 4 つのアルゴリズムについて，2D トーラス用のルータを Verilog-HDL により設計した．

4.1 Detour-UD ルータの構成

図 6 に 2D トーラス用 Detour-UD ルータの構成を示す．Detour-UD ルータは，東西南北 4 つのネット

表 1 2D トーラス用ルータの比較
Table 1 A comparison of 2D torus routers.

ルータ		DM-Order	Duato	Detour-NF	Detour-UD
適応ルーティング		×	○	○	○
耐故障性		×	×	△	○
ルーティング表（bits）		−	−	−	8×100
最少 VC 数/物理チャネル		2	3	4	2
VC4 本構成時	非適応 VC	4	2	2	0
	完全適応 VC	0	2	1	3
	部分適応 VC	0	0	1	1
	クロック（MHz）	79	55	52	48
	面積（セル数）	11,196	18,549	13,207	20,098
	FF 数	5,205	5,360	5,325	6,341

VC: Virtual Channel, FF: Flip-Flop

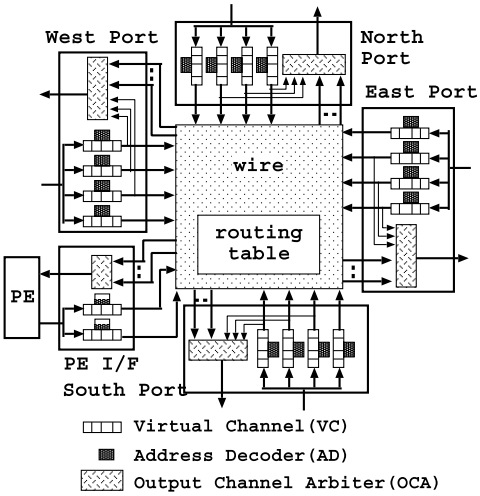


図 6 2D トーラス用 Detour-UD の構成
Fig. 6 Hardware organization of the Detour-UD router for 2D tori.

ワークポートとローカルプロセッサとのインタフェース（PE I/F），およびルーティング表からなる．図 6 中央の wire は，各ハードウェアブロック間の結線を簡略化して表している．

4 つのネットワークポートは，それぞれ 4 本の仮想チャネル（VC）とメッセージヘッダのアドレスデコーダ（AD），および出力チャネル調停回路（OCA）によって構成される．4 本の VC のうち，3 本を完全適応ルーティング用の FA-VC，1 本をデッドロック回復用の UD-VC としている．Detour-UD は最低 1 本の FA-VC と 1 本の UD-VC で構成可能であるが，他のルーティングアルゴリズムとの比較で VC 数を統一するため，本章では物理チャネルあたり 4 本の VC 構成について議論する．なお，一般にデッドロック回復ルータでは FA-VC を増やすことにより通信性能を向上させることができる[16]．任意のノードからのデッドロック回復をサポートするため，FA-VC はメッセージの 180 度ターン（入力チャネルと同一方向に出力する）が可能なように OCA へのデータパスを設けている．

PE I/F は 2 本の VC と OCA からなり，180 度ターンがないことを除き，ネットワークポートと同様の構成となっている．ルーティング表は，全ハードウェアブロックの仮想チャネルを 2 グループに分け，各グループの VC から同時に参照できるよう 2 ポート読み出しメモリとして実装した．

4.2 ルーティングアルゴリズムの特徴比較

本節では，**表 1** に示す 4 つの 2D トーラス用ルータについて，それぞれの特徴ついて比較する．また，各ルーティングアルゴリズムを実装するための最少 VC 数は異なるが，物理チャネルあたり 4 本の VC に統一した場合の内訳とハードウェアコストを示す．なお，Detour-UD を除く 3 つはどれもデッドロック回避型のルーティングアルゴリズムに基づく．

DM-Order X-Y 次元順に固定ルーティングする．

Duato Duato のプロトコルに基づいて最短経路を完全適応ルーティングする[9]．

Detour-NF Duato のプロトコルによる完全適応ルーティングとターンモデルに基づく Negative-First ルーティングによる故障チャネル/ノードの迂回をサポートする[19]．

Detour-UD 完全適応ルーティングと up*/down* ルーティングによるデッドロック回復をサポートし，ルーティング表を用いて故障チャネル/ノードの迂回に対応する．

DM-Order と Duato は，実際の並列計算機に最もよく採用されてきた最短経路ルーティングであるが，耐故障性は考慮していない．DM-Order は，最低 2 本の非適応ルーティング用 VC（以降，非適応 VC）で構成可能であるが，非適応 VC 数を増やすことにより通信スループットを改善できる[6]．Duato は，デッド

ロック回避のために非適応 VC を 2 本必要とし，それに完全適応ルーティング用 VC（完全適応 VC）を追加した構成となる．したがって，物理チャネルあたり 4 本の VC 構成では完全適応 VC が 2 本となる．Detour-NF は，Duato の最小要件である 2 本の非適応 VC と 1 本の完全適応 VC に非最短経路をサポートする Negative-First ルーティング用の部分適応 VC を 1 本追加した構成となる．この非適応 VC により，2D トーラスに対して最低 1 つの故障チャネル/ノードの迂回を可能とするが，故障位置によっては 2 つ以上の同時故障に対応できない．Detour-UD は，3 本の完全適応 VC とデッドロック回復のための 1 本の部分適応ルーティング用 VC（部分適応 VC）で構成し，任意数の故障に対応する．

4.3 ハードウェアコスト比較

表 1 の 4 つのルータは，無衝突時にメッセージヘッダを 1 ホップあたり (1) ラッチ (2) アドレスデコード (3) 出力調停（2 クロック）(4) データ転送の 5 クロックで実行するように設計した．ただし，Detour-UD のみデッドロック回復と故障領域でルーティング表を参照するときはアクセス調停と遅延でさらに 5 クロック必要となる．また，ルータ間物理チャネルは 32 ビット双方向とし，各 VC 容量は 32 ビット× 8 フリットとした．なお，Detour-UD に必要なルーティング表の容量は，図 3 の形式で 8 ビット ×100 語とした．

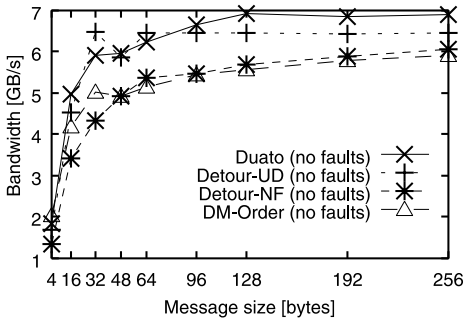
表 1 中のクロック速度，面積，フリップフロップ（FF）数は，各ルータの Verilog-HDL 設計を Synopsys FPGA-Compiler II により以下の条件で論理合成した結果である．

ターゲットデバイス Altera APEX II 自動選択
最適化 速度優先（High Effort）
I/O パッド 挿入

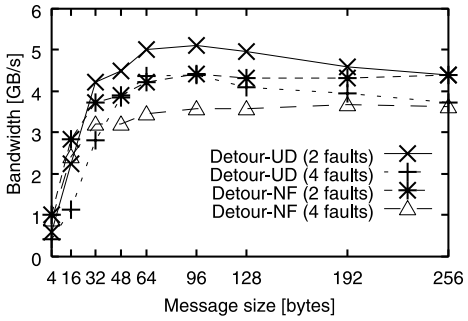
表 1 から，完全/部分適応 VC 数が増えるに従い，ロジックが複雑化し，クロック速度が小さくなることが分かる．同時に FPGA 上の回路面積も大きくなる．FF 数に関しては，ルーティング表の必要な Detour-UD の値が大きくなっている．したがって，回路の複雑さと適応ルーティングや耐故障性のための自由度はトレードオフの関係にあり，目的とする並列分散システムに応じて機能を選択すべきであるといえる．

5. シミュレーション結果

表 1 に示した 4 つのルータについて，Verilog-HDL シミュレータによる通信性能評価を行った．ネットワークはサイズ 10 × 10 の 100 ノード 2D トーラスとし，各ルータの動作周波数を表 1 に示したクロック速度，



(a) 無故障時



(b) 有故障時

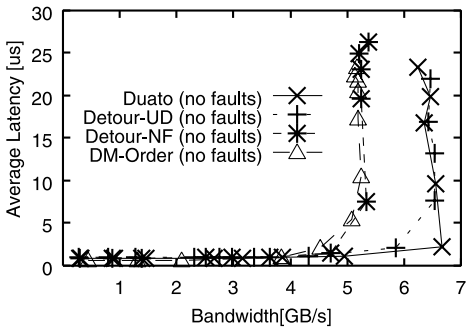
図 7 行列交換通信のバンド幅
Fig. 7 Bandwidth for a matix-transpose traffic.

ルータ間のデータ転送遅延を 1 クロック以内と仮定した．また，シミュレーション開始からネットワーク全体で 3000 番目までの到着メッセージをウォームアップとして除き，それに続く 4000 メッセージが宛先ノードに到着するまでの間を評価対象とした．

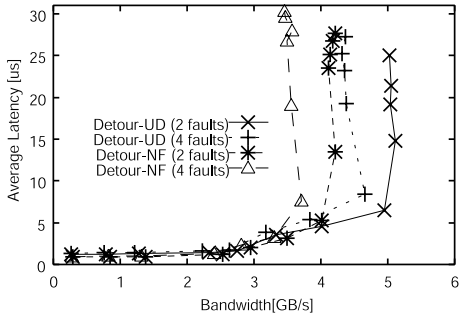
5.1 通信性能

DM-Order と Duato については無故障状態のみでの評価を，また Detour-NF と Detour-UD については無故障状態とネットワーク中に 2 つまたは 4 つのノードが静的に故障している場合についてのバンド幅とメッセージあたりの平均レイテンシを評価した．また，Detour-UD のデッドロック検出には，メッセージのブロッキング待ち時間を 128 クロックとし，故障領域距離は 2 に設定した．

図 7 に，行列交換通信においてメッセージサイズを変化させた場合のネットワーク全体の通信バンド幅を示す．行列交換通信は，番地 (i,j) と (j,i) のノード間でメッセージ交換を繰り返す通信パターンであり，ネットワーク上の通信に偏りが発生する．無故障時のバンド幅は，高い順に Duato，Detour-UD，Detour-NF，DM-Order となっている．Detour-UD は，完全適応 VC 数が最も多くルーティングの自由度は高いがクロック速度の関係で Duato よりも低バンド幅となっ

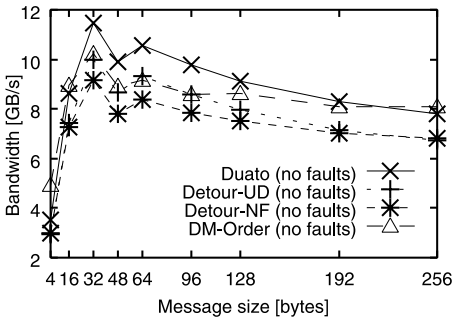


(a) 無故障時

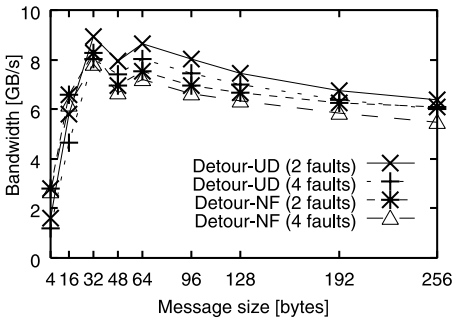


(b) 有故障時

図 8 行列交換通信の平均レイテンシ
Fig. 8 Average latency for a matix-transpose traffic.



(a) 無故障時



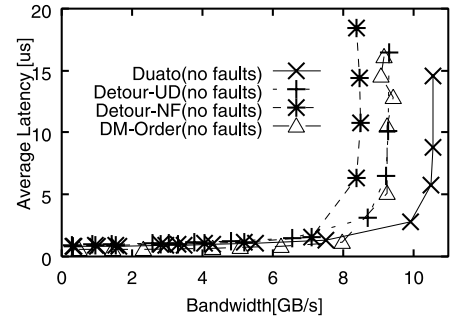
(b) 有故障時

図 9 ランダム通信のバンド幅
Fig. 9 Bandwidth for a random traffic.

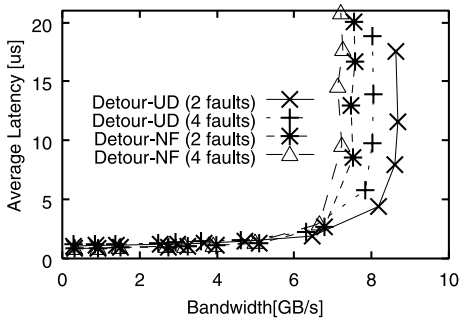
た．Detour-NF と DM-Order が同等なバンド幅を示した理由も，ルーティング自由度とクロック速度の影響が相殺された結果である．図 7 (b) から，Detour-UD と Detour-NF は故障ノード数が増えるとそれぞれバンド幅が低下するが，小さなメッセージサイズの場合を除き前者の方が高バンド幅であることが分かる．これは，Detour-NF が 1 本の部分適応 VC しか故障の迂回に利用できないのに対して，Detour-UD はトーラスルーティング表を利用して 3 本の完全適応 VC を故障の迂回に利用できることによる．有故障時の Detour-UD が，小さなメッセージサイズに対してバンド幅が低い理由は，故障領域内でルーティング表を参照する割合が増え，1 ホップあたりの遅延時間が増えるためである．また，メッセージサイズが大きくなったときに Detour-UD のバンド幅が Detour-NF に漸近するように低下する主な理由はデッドロック回復によるオーバヘッドと考えられる．

図 8 に，行列交換通信パターンのメッセージサイズを 64 バイト（16 フリット）に固定し，メッセージの送信間隔を変化させたときの平均レイテンシを示す．送信間隔が大きくネットワークが空いている（横軸のバンド幅値が小さい）状態ではルーティングアルゴリズム間の違いは小さいが，送信間隔を短くするとネットワークが飽和し，レイテンシが増加する．行列交換通信では，ルーティングの自由度が大きく，クロック速度が高速なほどネットワークの飽和容量は大きな値を示し，平均レイテンシも小さくなる．Detour-UD の 4 ノード故障時のレイテンシ特性は，Detour-NF の 2 ノード故障よりもネットワーク飽和容量が大きく低レイテンシとなっており，良好な耐故障性能を示している．

図 9 に，ランダム通信のバンド幅を示す．バンド幅の値は図 7 よりも高いが，メッセージサイズが大きくなると低下する傾向が見られる．バンド幅が高くなった理由は，メッセージがネットワークにより一様に分散するためである．また，メッセージサイズが大きくなった場合のバンド幅低下は，我々の設計したルータの仮想チャネル（VC）容量が 8 フリット分と小さく，ワームホール通信におけるブロッキングが起きやすくなることや，VC 切替え時にバブルサイクルが発生することなどによる．無故障時のルーティングアルゴリズムによる違いを観察すると，クロック速度の遅い Detour-UD と Detour-NF が低バンド幅となっているが適応ルーティングが効果的であった 32〜96 バイトメッセージについては，ルーティング自由度の大きい Detour-UD が DM-Order とのクロック差を相殺して

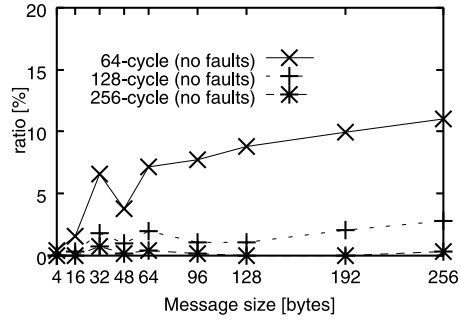


(a) 無故障時

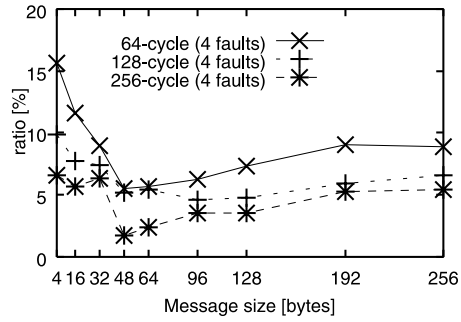


(b) 有故障時

図 10 ランダム通信の平均レイテンシ
Fig. 10 Average latency for a random traffic.



(a) 無故障時



(b) 4 ノード故障時

図 11 行列交換通信におけるデッドロック回復メッセージの割合
Fig. 11 The ratio of deadlock recovery messages for the matix-transpose traffic.

いる．

有故障時については，Detour-UD，Detour-NF ともに故障ノード数の増加に従ってバンド幅が低下する傾向は同じであるが，その低下率は行列交換通信よりも小さい．したがって，ランダム通信のようなユニフォーム通信パターンよりも非ユニフォーム通信に対する故障の影響が大きいことが分かる．

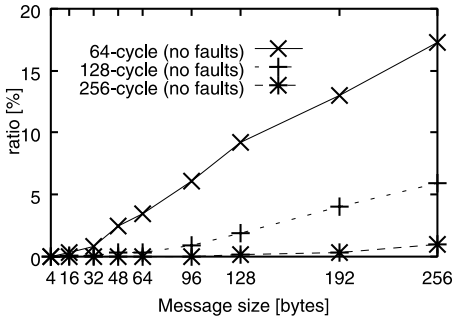
図 10 に，ランダム通信に対するメッセージあたりの平均レイテンシを示す．無故障時に Detour-UD と DM-Order のレイテンシ特性が近似しているのは，ルーティング自由度とクロック速度の影響が相殺した結果である．Detour-UD と Duato の差も主にクロック速度によるもので，仮にクロック速度が同じならばほぼ同一のレイテンシ特性を表す．有故障時の Detour-UD と Detour-NF の特性は，故障ノード数が増加するとネットワーク飽和容量が小さくなりレイテンシが増加するが，行列交換通信と比較して故障ノード数による影響は小さくなっている．

5.2 デッドロック回復の影響

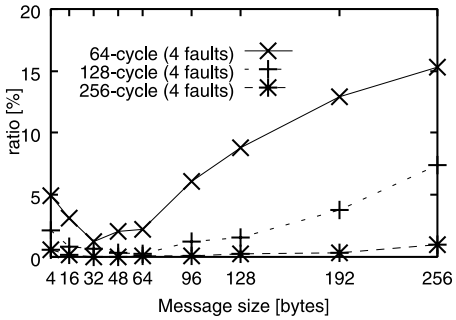
デッドロック回復ルータは，デッドロック回避方式に比べてデッドロック検出とデッドロック回復の実装方法が通信性能に影響しやすい．そこで前節に示した Detour-UD の評価におけるデッドロック検出のタイミングについて，メッセージのブロッキング待ち時間を 64，128，256 クロックと変えた場合のデッドロック回復メッセージ数の割合と通信性能への影響を調査した．**図 11**，**図 12** に無故障時と 4 ノード故障時の Detour-UD のバンド幅評価におけるデッドロック回復メッセージ数の割合を示す．

これらのグラフから，デッドロック検出までのメッセージのブロッキング待ち時間を短くするとデッドロック回復メッセージが増加することが分かる．また，無故障時にはメッセージサイズが大きくなるとデッドロック回復が増える傾向にあるが，この理由はブロッキング時間が長くなるためである．それに対して，4 ノード故障時にはネットワークが飽和しない小さなメッセージに対するデッドロック回復の割合が高く，ネットワークが飽和するメッセージサイズになるとサイズとともに増加する．これは，小さなメッセージが故障領域でホットスポットを形成しやすいためと考えられる．

デッドロック回復の頻度がネットワーク全体のバンド幅へ及ぼす影響については，無故障時には実験した範囲内で行列交換通信とランダム通信パターンのどちらにもほとんど現れなかった．ただし，4 ノード故障



(a) 無故障時



(b) 4 ノード故障時

図 12 ランダム通信におけるデッドロック回復メッセージの割合
Fig. 12 The ratio of deadlock recovery messages for the random traffic.

時に 64 クロックでデッドロック回復を開始する場合については，行列交換通信において 10%程度のバンド幅低下が見られた．したがって，デッドロック回復のオーバヘッドは有故障状況下，およびネットワークが飽和していない状況でデッドロック回復の頻度が高い場合に現れやすいことが分かる．このため，デッドロック回復用の仮想チャネルの使用頻度があまり高くならないような適切なデッドロック検出が重要といえる．

6. おわりに

本論文では，k-ary n-cube ネットワークを対象とした耐故障性を有するデッドロック回復ルーティングアルゴリズムを提案し，2 次元トーラス用の Detour-UD ルータの設計と通信性能を示した．Detour-UD ルータは，完全適応ルーティング用の VC と up*/down* ルーティングによるデッドロック回復用の VC を装備することにより，任意数/任意形状のチャネル/ノード故障に対応可能である．また，無故障領域でのネットワークトポロジの定型性を利用したルーティングと故障領域でのルーティング表を用いた最短経路による迂回をサポートすることで，通信性能を大きく損なうことなしに柔軟な耐故障性を提供する．ただし，4 章で議論したようにルータのハードウェアコストとチップの動作速度はルーティング自由度とトレードオフの関係にあり，目的とする並列分散システムに応じたルーティング機能選択が重要である．

今後の課題として，k-ary n-cube の次数や次元数を変えた場合の Detour-UD の評価や，動的な故障に対応するための効率的なネットワーク再構成アルゴリズムの設計などがあげられる．

謝辞 本研究におけるハードウェア設計は，東京大学大規模集積システム設計教育研究センターを通し，日本ケイデンスおよびシノプシス株式会社の協力で行われたものである．

本研究は，一部科学研究費補助金 基盤研究（C）(2) 課題番号 15500033 の援助による．

参 考 文 献

1) An overview of the BlueGene/L supercomputer, SC2002 Technical Paper (2002).
2) Anjan, K.V. and Pinkston, T.M.: An Efficient, Fully Adaptive Deadlock Recovery Scheme: DISHA, *Proc. 22nd ISCA*, pp.201–210 (1995).
3) Anjan, K.V., Pinkston, T.M. and Duato, J.: Generalized Theory for Deadlock-Free Adaptive Wormhole Routing and its Application to Disha Concurrent, *Proc. IPPS '96*, pp.815–821 (1996).
4) Boppana, R.V. and Chalasani, S.: Fault-Tolerant Wormhole Routing Algorithms for Mesh Networks, *IEEE Trans. Comput.*, Vol.44, No.7, pp.848–864 (1995).
5) Chien, A.A. and Kim, J.H.: Planer-adaptive routing: Low-cost adaptive networks for multiprocessors, *Proc. 19th ISCA*, pp.268–277 (1992).
6) Dally, W.J.: Virtual-Channel Flow Control, *Proc. 17th ISCA*, pp.60–68 (1990).
7) Duato, J.: A Theory of Fault-Tolerant Routing in Wormhole Networks, *IEEE Trans. Parallel and Distributed Systems*, Vol.8, No.8, pp.790–802 (1997).
8) Duato, J., Yalamanchili, S. and Ni, L.: *Interconnection Networks—An Engineering Approach*, p.515, IEEE Computer Society Press (1997).
9) Duato, J.: A New Theory of Deadlock-Free Adaptive Routing in Wormhole Network, *IEEE Trans. Parallel and Distributed Systems*, Vol.4, No.12, pp.1320–1331 (1993).
10) Glass, C.J. and Ni, L.M., The Turn Model for Adaptive Routing, *Proc. 19th ISCA*, pp.278–287 (1992).

11) 上楽明也，鯉渕道紘，天野英晴：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，情報処理学会論文誌：コンピューティングシステム，Vol.44, No.SIG 11(ACS3), pp.157–168 (2003).
12) Mukherjee, S.S., Bannon, P., Lang, S., Spink, A. and Webb, D.: The Alpha 21364 Network Architecture, *IEEE Micro*, Vol.22, No.1, pp.26–35 (2002).
13) http://www.myri.com/
14) Petrini, F., Feng, W., Hoisie, A., Coll, S. and Frachtenberg, E.: The Quadrics Network: High-Performance Clustering Technology, *IEEE Micro*, Vol.22, No.1, pp.46–57 (2002).
15) Pinkston, T.M., Pang, R. and Duato, J.: Deadlock-Free Dynamic Reconfiguration Scheme for Increased Network Dependability, *IEEE Trans. Parallel and Distributed Systems*, Vol.14, No.8, pp.780–794 (1997).
16) Pinkston, T.M. and Warnakulasuriya, S.: On Deadlocks in Interconnection Networks, *Proc. 24th ISCA*, pp.38–49 (1997).
17) Rodeheffer, T.L. and Schroeder, M.D.: Automatic Reconfiguration in Autonet, *Proc. ACM Symposium on Operating Systems Principles* (*SOSP'91*), pp.183–197 (1991).
18) Schroeder, M.D., Birrel, A.D., Burrows, M., Murray, H., Needham, R.M., Rodeheffer, T.L., Satterthwaite, E.H. and Thacker, C.P.: Autonet: A High-Speed, Self-Configurable Local Area Network using Point-to-Point Links, *IEEE J. Selected Areas Commun.*, Vol.9, No.8, pp.1318–1335 (1991).
19) Yoshinaga, T., Hosogoshi, H. and Sowa, M.: Design and Evaluation of a Fault-Tolerant Adaptive Router for Parallel Computers, *Proc. 6th IWIA*, pp.100–107 (2003).
20) Wang, D.: A Rectilinear-Monotone Polygonal Fault Block Model for Fault-Tolerant Minimal Routing, *IEEE Trans. Comput.*, Vol.52, No.3, pp.310–320 (2003).

(平成 16 年 1 月 29 日受付)
(平成 16 年 5 月 9 日採録)

吉永　努（正会員）

1986 年宇都宮大学工学部情報工学科卒業．1988 年同大学大学院修士課程修了．同年より宇都宮大学工学部助手．2000 年より電気通信大学大学院情報システム学研究科助教授．博士（工学）．1997 年から翌年にかけて電子技術総合研究所・客員研究員．分散並列処理，クラスタ・コンピューティング等に興味を持つ．電子情報通信学会，IEEE 各会員．

細越　洋行

2002 年電気通信大学電気通信学部卒業．2004 年同大学大学院情報システム学研究科博士前期課程修了．分散並列計算機アーキテクチャ，高信頼システム等に興味を持つ．現在，キヤノン株式会社．

曽和　将容（正会員）

1974 年名古屋大学大学院博士課程修了，工学博士．同年群馬大学工学部助手．助教授をへて 1986 年名古屋工業大学工学部教授．1993 年から電気通信大学大学院情報システム学研究科教授．並列プロセッサ等の研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM 各会員．