多次元構造を持つMIN

塙 敏博[†,☆] 朱 笑岩[†,☆☆]
亀井貴之[†,☆☆☆] 天野英晴[†]

本論文では，多次元構造を持つMIN，nD-MINを提案する．nD-MINでは，Generalized Cube網のクロスリンクを多次元に拡張して，さらに循環リンクを付加することにより，特定のメモリモジュールに対する転送ステップ数を減らし，アクセスの局所性を利用することができる．ルーティングの方法として，アルゴリズムの簡単な平面ルーティングと，多次元構造を最大限に利用することのできる次元自由ルーティングを提案した．確率モデルによるシミュレーションの結果，次元自由ルーティングを用いた場合，ランダム転送においても，通常のMINより優れたアクセスレイテンシを実現することができることを示した．さらに，アクセスに局所性が存在する場合は，アクセスレイテンシを最大24%改善できることが分かった．また，故障や混雑を迂回するルーティング方法についても提案し，大きな性能の低下なしに迂回が可能であることが分かった．

MIN with Multiple-dimensional Structure

Toshihiro Hanawa,[†,☆] Xiaoyan Zhu,[†,☆☆] Takayuki Kamei[†,☆☆☆]
and Hideharu Amano[†]

In this paper, we propose a novel MIN (Multistage Interconnection Network) structure called the multi-dimensional MIN (nD-MIN). Unlike the conventional MINs, the nD-MIN can make the best use of access locality by its multi-dimensional structure and feedback lines. According to the probabilistic simulation results, the latency is 24% better than that of traditional MINs under the traffic with communication locality, by using the dimensional free routing. Even without communication locality, the latency of the nD-MIN is better than that of traditional MINs if the size is appropriately matched. Faulty or congested elements or links can be bypassed with a simple modification of the routing algorithm.

1. はじめに

大規模な並列計算機を構成する際に，PU（Processing Unit）間あるいはPU-メモリ間の相互結合方式は重要な位置を占めており，並列計算機の規模，応用問題の性質などに応じた最適な結合網を求める研究が積極的に進められている．その中で，多段結合網**MIN**（Multistage Interconnection Network）はスケーラビリティが高く，転送容量が大きい利点を持つため，PUと共有メモリモジュールを結合する方法として，古くから研究されてきたが[1]，実際のマシン上で実装された例は最近少なくなっている．

これは，MINの持つ構造上の問題点が1つの原因となっている．すなわち，従来のMINでは，PUとメモリモジュールの間で交換されるパケットは，必ず同じ数のステージを通って転送される．この構造は，同時に複数のメモリモジュールから複数のPUに大量にデータを転送するには便利であるが，PUがメモリモジュールに対して行うアクセスに局所性が存在する場合，その利点を活かすことができない．システムが大規模になると，1つのシステム上で多数のジョブやプロセスが動作するようになり，すべてのPUがすべてのメモリモジュールに対して一様にアクセスを発生することは考えにくい．そこで，MINの特徴を保持したまま，特定のメモリモジュールに対するアクセスを効率良く行うことのできる構成が有利になる．

本論文では，局所性を利用することが可能で大規模

† 慶應義塾大学理工学部
Faculty Science and Technology, Keio University
☆ 現在，東京工科大学情報工学科
Presently with Department of Information Technology, Tokyo Engineering University
☆☆ 現在，三菱マテリアル株式会社サイバースペース研究所
Presently with Cyberspace Research Laboratory, Mitsubishi Materials Corporation
☆☆☆ 現在，株式会社東芝システムLSI技術研究所
Presently with System ULSI Engineering Laboratory, Toshiba Corporation

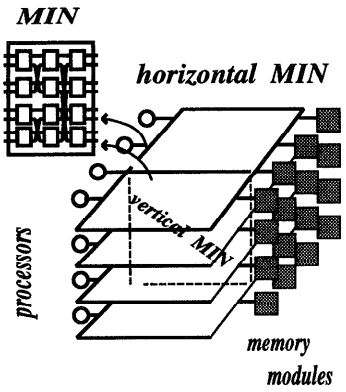


図1 3次元 MIN の構成原理
Fig. 1 The first step to 3D-MIN.

並列システムに向く n 次元 MIN（nD-MIN）を提案する．この MIN は，循環構造と多次元構造を持つことで，MIN の特徴を保ったまま，局所性を生かすことができる．さらに，そのレイテンシについて確率モデルを用いて評価を行い，故障や混雑の迂回法について検討を行う．

2. 多次元 MIN の提案

2.1 nD-MIN の構成原理

nD-MIN は，MIN のステージ間の接続を多次元方向に拡張することにより構成する．ここでは，基本となる MIN に図 1 中に示す Banyan 網（Generalized Cube 網）[2] を用いる．この網は，1 つのスイッチングエレメントが，出力と同一番号の端子に対して接続されたリンク（平行リンク）と 2^S（S はステージ番号）離れた番号の端子に対して接続されたリンク（クロスリンク）を持ち，直接網のハイパキューブに相当する多段網を構成する．

nD-MIN は，このクロスリンクを多次元方向に対しても設けることにより構成する．図 1 に 3 次元 MIN の構成原理を示す．この構成では，PU は，自分と同一の水平および垂直平面上のメモリモジュールしかアクセスできない．そこで，Generalized Cube 網に図 2 に示すように循環リンクを付加する．この構造は，直接網の中の循環網に属する Circular Banyan 網[3] のトポロジを MIN に利用したものと考えることができる．

図 2 の構造を基本とした 3D-MIN を図 3 に示す．この構造は直接網である CB[2] [4),5)] を MIN に応用したものと考えることもできる．循環リンクを持つことによりハードウェア量は増加するが，任意のメモリモジュールにアクセスすることができる．言い換えると，3 次元 MIN では水平，垂直のそれぞれの方向で同一平面上のメモリモジュールに対しては直接アクセス可能である．他のメモリモジュールに対しては，最初に垂直方向の MIN を用いて目的地と同一垂直平面上にパケットを送り，いったん循環リンクを用いて出発地に戻った後，水平方向の MIN を用いて目的地にパケットを送ることによりアクセスすることができる．

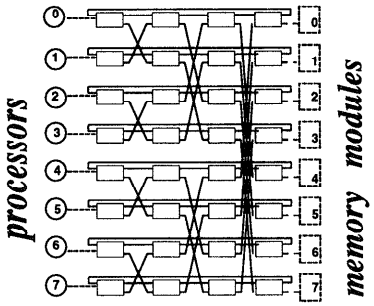


図2 循環構造を付加した網
Fig. 2 The MIN with circular links.

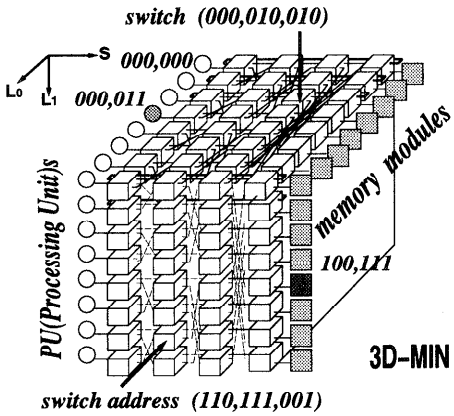


図3 3次元 MIN の構成
Fig. 3 The structure of 3D-MIN.

図 3 に示す 3D-MIN において，PU 番号，メモリ番号はそれぞれ 1 次元少ない 2 次元の座標 $P(X_0, X_1)$, $M(X_0, X_1)$ で表すことができ，スイッチングエレメント番号はこれにステージ方向を合わせた 3 次元の座標 $E(X_0, X_1, s)$ で表すことができる．エレメント間の接続は，$E(X_0, X_1, s+1)$ に対する平行リンクと X_0, X_1 方向にそれぞれ 2^S 離れたエレメントに対するクロスリンクによって接続される．

以下，nD-MIN の接続方法を定義する．

2.2 nD-MIN の接続

まず，以下のように PU，メモリ，スイッチングエレメントに番号付けを行う．ここで，次元数を n とし，網のホップ数（入力側と出力側の距離=ステージ数）を S とする．j 次元におけるプロセッサ数（=メモリ数）を $L_j = 2^{l_j}$ $(0 \leq j \leq n-2, l_j \geq 1)$ で表す．

プロセッサ数を N とすると，$N=\prod_{j=0}^{j=n-2} L_j$ で表すことができる．このとき $S=\max(l_j)+1$ である．ここで，すべての $L_j(0\leq j\leq n-2)$ が等しいときを，正方 nD-MIN と呼ぶことにする．

PU 番号

$P(X_0,X_1,\ldots,X_{n-2})$

メモリ番号

$M(X_0,X_1,\ldots,X_{n-2})$

ここで，$X_0,X_1,\ldots,X_{n-2}$ は各次元の座標 $(0\leq X_j\leq n-2)$ である．

ここでは，PU からメモリへの単方向の MIN を考える．したがって，PU 番号は MIN に対する入力ラベル（出発地番号），メモリ番号は出力ラベル（目的地番号）となる．

スイッチングエレメント番号

$E(X_0,X_1,\ldots,X_{n-2},s)$

ここで，$X_0,X_1,\ldots,X_{n-2}$ は各次元の座標 $(0\leq X_j\leq L_j)$ であり，s はステージ番号 $(0\leq s\leq S-1)$ である．

上記の番号付けについて，座標 X_j は l_j 桁の 2 進数

$X_j=x_{j(l_j-1)}x_{j(l_j-2)}\cdots x_{j0}(0\leq j\leq n-2)$

で表される．

nD-MIN の接続は以下のように定義される．

$E(X_0,X_1,\ldots X_{n-2},s)$ を 1 本の平行リンクと，$n-1$ 本のクロスリンクにより，以下のスイッチングエレメントと接続する．

平行リンク（pl）

$E(X_0,X_1,\ldots X_{n-2},(s+1)\ mod\ S)$

クロスリンク（$cl_j(0\leq j\leq n-1)$）

$cl_j: E(X_0,\ldots,rev(X_j,s),\ldots X_{n-2},s+1)$

ただし，$(s\leq S-2)$ であり，$rev(X_j,s)$ は X を 2 進数で表した場合の s bit 目を反転する関数である．すなわち，

$$\mathrm{rev}(X_j,s)=x_{j(l_j-1)}\cdots\overline{x_{js}}\cdots x_{j0}\qquad(0\leq s\leq l_j-1)$$

$(S-1)$ ステージすなわち最終ステージのリンクは，循環平行リンクおよびメモリモジュールに接続される出力リンクになる．

上記のスイッチングエレメント間の接続に加え，第 0 ステージの入力にプロセッサ P，最終ステージの出力にメモリ M を接続すると，単方向の nD-MIN を構成することができる．すなわち，入力リンク inl，出力リンク $outl$ は，

inl: $P(X_0,X_1,\ldots X_{n-2})\rightarrow E(X_0,X_1,\ldots X_{n-2},0)$

outl: $E(X_0,X_1,\ldots X_{n-2},S-1)$

$\rightarrow M(X_0,X_1,\ldots X_{n-2})$

のリンクである．

2.3 ルーティング

nD-MIN のルーティングアルゴリズムは Generalized Cube 同様，排他的論理和で構成されるタグを用いる．ただし，nD-MIN は多次元構成であるので，それぞれ対応する次元どうしの番号のビット単位の排他的論理和をとる必要がある．

ルーティングタグ

$$\begin{aligned}T(T_0,T_1,\cdots,T_{n-2})&=P(P_0,\cdots,P_{n-2})\oplus M(M_0,\cdots,M_{n-2})\\&=T(P_0\oplus M_0,\cdots,P_{n-2}\oplus M_{n-2})\end{aligned}$$

$$\begin{aligned}P_j\oplus M_j\equiv T_j&=p_{j(l_j-1)}\oplus m_{j(l_j-1)}\cdots p_{ji}\oplus m_{ji}\\&\quad\cdots p_{j0}\oplus m_{j0}\\&\equiv t_{j(l_j-1)}\cdots t_{ji}\cdots t_{j0}\end{aligned}$$

2.3.1 平面ルーティング

さて，このルーティングタグを用いて可能な，最も基本的なルーティングは，各次元の平面についてのみ転送を行う平面ルーティングである．以下，このルーティング法を C 言語に似た表記法により示す．

平面ルーティングアルゴリズム

```
if (T == 0){
  /* タグが 0 ならすべて平行リンク */
  for (i = 0; i < S − 1; i ++) Use pl;
}else
  for (j = 0; j < n − 1; j ++) {
    if (T_j ≠ 0) {
      /* j 次元のタグが 0 でない */
      for (i = 0; i < S − 1; i ++)
        if (t_ji == 0) {
          /* i bit 目が 0 なら平行リンク */
          Use pl;
        } else {
          /* 1 ならクロスリンク */
          Use cl_j;
        }
      if (j ≠ n − 2) {
        /* 端まで来て，全次元をチェック
           し終わっていなければ循環リンク */
        Use pl;
      }
    }
  }
Use outl ;
```

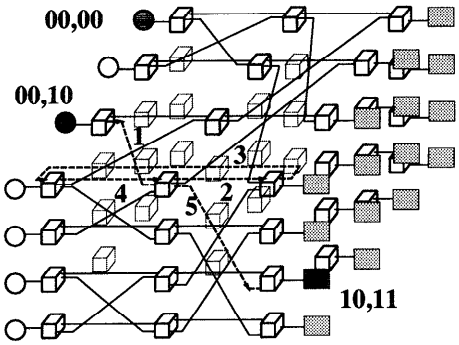


図4 3D-MIN におけるルーティングの例
Fig. 4 Routing examples in the 3D-MIN.

この方法は，下の次元から順にタグが0でないかどうかチェックし，0でなければタグの下のビットから順に判断し，0であれば平行リンク，1であればその次元のクロスリンクに転送する方法である．ただし，j 次元でのタグ（T_j）が0のときは，その次元でのルーティングは省略できる．したがって，パケットは一度同じ平面上でルーティングを始めると，その平面からはずれることはない．パケットは $T_j \neq 0$ の次元数に等しい回数だけ MIN の入口から出口までを通過して転送される．すなわち $T_j \neq 0$ の (次元数 − 1) 回だけ循環リンクを用いる．

2.3.2 次元自由ルーティング

平面ルーティングは実装が容易であるが，パケットが MIN を1回通過するだけで転送可能なメモリモジュールは，同一平面上に制限される．たとえば，3D-MIN では図4に示すように，プロセッサ $P(00,10)$ からメモリモジュール $M(10,11)$ まで転送を行う場合，1-2-3-4-5 の順番でルーティングする必要があり，MIN を2回通過する必要がある．ところが，転送する次元を途中で変えることができれば，1-5 の順でリンクを利用すれば，1回で到着することができる．このように途中で利用する次元を変えることのできるルーティング法である次元自由ルーティングを提案する．

次元自由ルーティングアルゴリズム

```
if (T == 0){
  /* タグが0ならすべて平行リンク */
  for (i = 0; i < S − 1; i ++) Use pl;
} else
  while (T ≠ 0) {
    /* タグが0でなければ */
    for (i = 0; i < S − 1; i ++) {
      for (j = 0; j < n − 1; j ++){
        /* j 次元のタグについて */
        if (t_ji == 1) {
          /* i bit 目が1ならクロスリンク */
          Use cl_j ;
          /* 0にリセット */
          t_ji = 0;
          break;
        }
      }
      if (j == n − 1) {
        /* i bit 目を全次元チェックして
           すべて0だったら平行リンク */
        Use pl ;
      }
    }
    if (T ≠ 0) {
      /* タグが0にならなければ循環リンク */
      Use pl;
    }
  }
Use outl ;
```

この方法では，ステージ i において，すべての次元のタグの i bit 目をチェックする．ここで，1になっているタグがあればその次元のクロスリンクにパケットを転送し，すべての次元のタグが0ならば平行リンクにパケットを転送する．パケットが出口に到着したときにすべてのタグが0になっていなければ循環リンクを通してパケットを戻し，ルーティングを繰り返す．この方法は，並列に n 次元のタグをチェックするハードウェアが必要になるが，最短距離のルーティングが可能である．

表1 に平面ルーティングと，次元自由ルーティングの通過するエレメント数の平均（平均距離）を示す．次元数が大きくなると1回の巡回に要するステップは減るが，巡回数が増える．このため，平面ルーティングでは次元によらずほぼ一定を保ち，次元自由ルーティングでは，次元数が大きくなると，少し減る傾向にある．これは次元自由ルーティングは，次元数が増えるほど，可能なルーティングの範囲が増えるためである．

次元自由ルーティングを用いることにより，平面ルーティングに比べ，最小9%，最大32%程度平均距離を小さくすることができる．平面ルーティングの平均距離は，正方 nD-MIN のときよりも正方でない場合の方が大きくなる一方，次元自由ルーティングの平均距離は，正方 nD-MIN でないときの方が小さくなる．これは，次元自由ルーティングでは，サイズが合わない場合に存在する冗長なステージを使ってルーティング

表1 各ルーティング法における平均距離
Table 1 Average distance of two routing methods.

PU数	3D			4D			5D		
	構成	平面	自由	構成	平面	自由	構成	平面	自由
64	(8,8)	7.06	6.31	(4,4,4)	6.80	5.95	(4,4,2,2)	7.55	4.31
128	(16,8)	9.10	7.19	(8,4,4)	9.53	6.50	(4,4,4,2)	8.27	5.95
256	(16,16)	9.39	8.42	(8,8,4)	10.02	7.38	(4,4,4,4)	9.01	7.65
512	(32,16)	11.45	9.47	(8,8,8)	10.51	8.82	(8,4,4,4)	12.51	8.25
1024	(32,32)	11.63	10.58	(16,8,8)	13.44	9.92	(8,8,4,4)	13.00	8.80

を行うことが可能なためである．

2.4 ハードウェアコスト

次に nD-MIN のハードウェアコストについて評価する．プロセッサ数（メモリモジュール数）を P，MIN の次元を n とし，各次元方向のサイズを等しいとすると，ステージ数 S は

$$\frac{\log_2 P}{n-1}+1$$

となる．したがってエレメント数は，

$$P\times\left(\frac{\log_2 P}{n-1}+1\right)$$

となる．一般の MIN におけるエレメント数は，2×2 のスイッチングエレメントを用いると，

$$\frac{P}{2}\times\log_2 P$$

であるので，エレメント数に関しては，nD-MIN は次元数が増えるほど，一般の MIN に比べて有利であることが分かる．

しかし，nD-MIN はその分入出力が増えるため，クロスポイント数で評価すると，

$$P\times\left(\frac{\log_2 P}{n-1}+1\right)\times n^2$$

となって，一般の MIN における各エレメントのクロスポイント数がつねに4である MIN に比べて不利であることが分かる．しかし，最近の LSI 技術の発達を考えると，クロスポイント数が多くても LSI のピンネックにならない範囲では一概に不利とはいえず，エレメントの配置，リンクの配線のしやすさ等も含め検討する必要がある．

3. シミュレーションによる性能評価

理論的に求めることのできる転送ステップ数は，ネットワーク内部でのパケットの衝突による遅れが考慮されていないため，実際的ではない．そこで，確率モデルに基づきやや実際的なシミュレーションを行う．

3.1 評価条件

ここでは，ローカルメモリまたはキャッシュを持つプロセッサが nD-MIN を介してメモリモジュールと接続されているシステムを想定する．命令およびスタックはローカルメモリまたはキャッシュに格納されていると考え，各プロセッサは，以下のアクセスを各クロックでデータに対して発生するとした．

- n_{br}：ブロックアクセス．読み出しまたは同期変数へのアクセス．各プロセッサはデータが到着するまで次のアクセスを発生することができない．
- n_{nb}：ノンブロッキングアクセス．書き込みアクセスで，プロセッサはアクセスを発生してすぐ次の計算またはアクセスに移ることができる．

ここでは，評価データが発表されているスイッチ結合型並列計算機 SNAIL[6]の評価に基づき，$n_{br}=0.01$，$n_{nb}=0.04$ に設定した．プロセッサ数は $(2^6,2^7,2^8,2^9,2^{10})$ とし，プロセッサの構成は表1に示したとおりである．100000クロックをシミュレーションした．なお，ネットワークが安定するまでの10000クロックは，シミュレーション結果から除いた．

スイッチのモデルは，エレメント内部にパケット格納用バッファを持つ一般的な MIN であり，パケットは1クロックでエレメント間を転送される．読み出しデータ用に，メモリモジュールから PU へ向かう帰還用の MIN を別に持つものとする☆．スイッチングエレメントは，入力リンクに対して FIFO を，出力リンクに対してパケット1個分格納できるバッファを持つ．次のステージのエレメントの FIFO が溢れた場合，パケットは出力リンクのバッファで待ち状態になる．ここではシミュレーション時間とメモリの関係で FIFO で格納できるパケットの最大数は3とした．パケットの優先制御は以下のように行った．

- 平行リンクからのパケットに最高の優先権を与え，順に次元数の低いリンクから来たパケットほど高い優先権を与える．

☆ このことによりハードウェア量は大きくなるが，Illinois 大の Cedar，IBM の RP3 等これまで実装された並列計算機の多くは帰還用の MIN を持つ．我々は帰還用 MIN が必要なく，高速なアクセスが可能な，SSS 型 MIN を提案している[7]が，まだ一般的とはいえないため，ここでは多くの例に従い，帰還用 MIN を別途用意することにした．

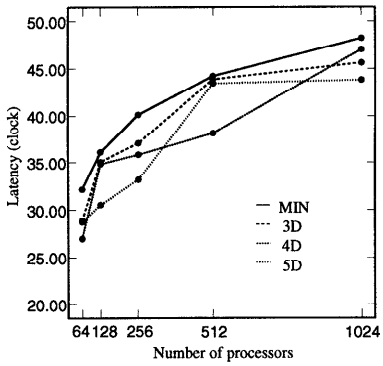


図5 レイテンシ（局所性なし）
Fig. 5 Latency without locality.

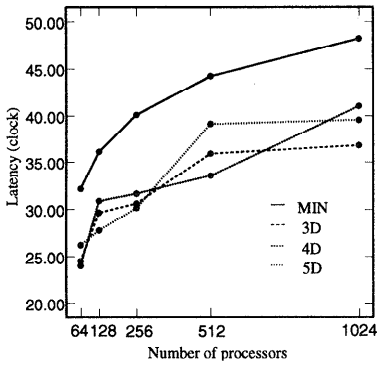


図6 レイテンシ（局所性あり）
Fig. 6 Latency with locality.

- 衝突して，1回待たされたパケットはそのつど優先権を上げ，次回の転送に有利になるようにする．

また，メモリのアクセス時間は3クロックとし，メモリモジュール内はスイッチングエレメント同様パケット3個分のFIFOを持つ．

3.2 アクセスレイテンシ

ここでは，プロセッサが発生した読み出し（ブロッキングアクセス）要求に対応したパケットが戻ってくるまでのクロック数をアクセスレイテンシと考え，アクセスが局所性を持っている場合と持っていない場合について，測定を行った．

各プロセッサがまったくランダムにメモリモジュールに対してアクセスを行った場合に関して，次元自由ルーティング法を用い，プロセッサ数64から1024まで変えた場合のレイテンシを**図5**に示す．

図5によると，局所性のない場合でも，nD-MINは通常の2次元MINに比べ，レイテンシが小さいことが分かる．特に，サイズが合う場合（3Dのときの64，256，1024，4Dのときの64，512，5Dのときの1024）は，2次元MINに比べ明らかに低いレイテンシを実現している．これは，nが4以上の場合，次元自由ルーティングの利用により，nD-MINの方が平均距離が小さい点に加え，2次元MINではパケットどうしの衝突によって遅延が増加するのに対し，nD-MINはより交換能力の高いスイッチングエレメントを用いているため混雑時にも性能が低下しないためと考えられる．また，いずれの次元の場合も，プロセッサ数が次元に合う場合を境に階段上にレイテンシが増加している．

次に，アクセスに局所性のある場合のレイテンシを測定した．これは，アクセスの50%がそれぞれ次元数の一致するメモリモジュールに対して行われると仮定した．この場合，通常の2次元MINに対し，アクセスの局所性を与えると，かえってホットスポットを形成して性能が低下するため，通常のMINに対するアクセスは均一アクセスを用いた．この場合，**図6**に示すように，局所性を生かすことができるnD-MINがさらに有利になり，最大24%改善される．

この結果において，プロセッサ数によって，次元数とレイテンシの大小関係が逆転しているが，サイズが合わない場合に，無駄なスイッチングエレメントを多数通過することになり，レイテンシがあまり減少しないためである．

4. その他の特性と他のMINとの比較

4.1 パーティショニング

nD-MINは，Generalized Cubeにリンクを追加した構造をとるため，各平面のステージ方向に関してパーティショニングが可能である[2]．**図7**に示すように，各列は2^m単位でパーティションにまとめることができ，各パーティション内の通信は他のパーティションに出ることはない．

この性質はそれぞれの平面に対して有効であるため，nD-MINは各次元方向に2^mのクローズドパーティションを形成することができる．このパーティションは，独立した複数のジョブを実行する場合等に有効である．

4.2 故障および混雑の回避

nD-MINは，循環構造を持つため特定のスイッチングエレメントあるいはリンクの故障あるいは混雑を迂回することができる．

最も基本的な迂回法として，まず，nD-MINの任意の平面上での迂回を考える．同一平面上では，すべてのアクセスは，故障あるいは混雑がなければ循環ループを経由しないで目的とするメモリモジュールに到着する．プロセッサに直接接続されたスイッチングエ

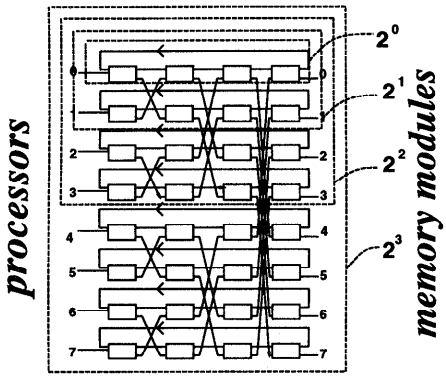


図7 パーティショニング
Fig. 7 Partitioning.

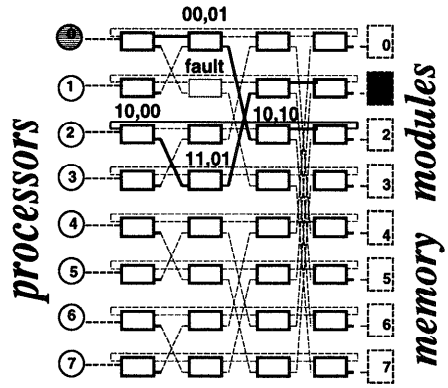


図8 迂回の様子
Fig. 8 Example of bypassing.

レメントが故障した場合，または，メモリに直結したスイッチングエレメントが故障した場合は，そのプロセッサまたはメモリは，システムから切り離される☆．上記のエレメントが混雑した場合は，単純にバッファが空くまで待つことになる．

同一平面上での迂回ルーティング

スイッチングエレメント $E(F, fs)$ の入力リンクまたはバッファが故障したり混雑している場合（ただし F の2進数表示を $F = f_{l-1} \ldots f_0$ とする），そのスイッチングエレメントにデータを転送するエレメント $E(F, fs-1)$，$E(rev(F, fs-1), fs-1)$ は，

(1) fs ステージでは，故障あるいは混雑しているスイッチングエレメントに接続されていない方のリンクを用いてパケットを送る．

(2) $fs+1$ ステージで，クロスリンクを用いた転送を1回行う．$fs+1$ ステージ以降では任意のルーティングを行う．

(3) 循環ループを経由して，2巡目は平面ルーティングに基づいたルーティングを用いる．

上記の手法で故障または混雑しているエレメントを迂回できるのは以下の理由による．

迂回ルーティングの正当性

迂回ルーティングで1巡目に到着する最終ステージのスイッチングエレメント $E(X, n-2)$ のラベル X は，$fs+1$ ステージでクロスリンクを用いることから以下のように表すことができる．

$$X = x_{n-1} \ldots x_{fs+2} \overline{f_{fs+1}}\ \overline{f_{fs}} f_{fs-1} \ldots f_0$$

となる．ただし，$x_{n-1} \ldots x_{fs+2}$ は任意の2進数である．

ここで，循環リンクを用いて2巡目のルーティングを行うため，2巡目の出発地のエレメントのラベルは，$E(X, 0)$ となる．基本ルーティング方法に従ってルーティングを行う場合，出発地の $fs+1$ 桁目のラベルが $\overline{f_{fs+1}}$ であることから，2巡目の通過スイッチングエレメントのラベルは，$fs+1$ ステージまでは1巡目と一致することはない．すなわち，2巡目のパスは故障あるいは混雑したスイッチングエレメント $E(F, fs)$ を通ることはない．

図8 に，PU0がメモリモジュール1に対してパケットを送ったとき，故障したスイッチングエレメント（fault印）を迂回する様子を示す．故障エレメントを避けた次のステージでクロスリンクを用いているため，2巡目のパスは2巡目のパスと故障あるいは混雑したエレメントの存在するステージまでは，異なったパーティション内を進む．このため2巡目のパスが再び故障あるいは混雑したエレメントを通ることはない．

上記の迂回ルーティングは，nD-MIN のすべての次元の平面で可能であり，1巡目だけでなく，2巡目以降に対しても適用することができる．したがって，nD-MIN は，基本ルーティングアルゴリズムに対して，故障あるいは混雑が起きた場合，巡回数を1増やすだけで，迂回可能なパスが n 本存在することが分かる．各平面は，エレメントが2個故障した場合，通過できないエレメントが生じ利用することができなくなる．したがって，nD-MIN は $2n-1$ 個のエレメント故障に対処できることが分かる．

表2 にステージ1のエレメントが1個故障が生じた場合の，平均距離の増加を示す．表2を表1中の平面ルーティングの平均距離と比較すると，故障による平均距離の増加はわずかである．上記の性質は，故障回避だけでなく，混雑を回避する適用型ルーティングを用いる場合にも有利である．

上記の議論は簡単のために平面ルーティングに関し

☆ ほとんどの耐故障性MINにおいて，同様の仮定をしている[8]．

表2 迂回した場合の平均距離
Table 2 Average distance bypassing for faulty element.

	3D	4D	5D
64	7.08	6.80	7.55
128	9.11	9.53	8.28
256	9.40	10.02	9.01
512	11.45	10.51	12.51
1024	11.63	13.44	13.00

て行ったが，次元自由ルーティングについても同様の方法が適用可能である．

4.3 他の研究との比較

他にもループ構造，3次元構造を持ったMINが提案されている．Kumarらによる ASEN[9]は，循環ループを持つ MIN であるが，この循環ループは縦方向であり，主に故障回避のために用いられている．

また，Leaらの Multi-Log_2N ネットワーク[10]，塙らによる PBSF[11]は3次元構造を持った MIN であるが，この MIN では3次元的な接続は1次元的に並んだPUとメモリを結んでおり，結合網の通過率を向上させるために用いられている．したがって，同一数の PU-メモリを接続する場合は，3D-MIN に比べてはるかに大量のハードウェアを必要とする．また，nD-MIN の主たる目的である局所的な利用ができなくなる．

村田らによる MDX[12]は，他次元方向にスイッチを持つ点で nD-MIN と共通点がある．しかし MDX はそれぞれの方向の接続は単段接続でり，多段接続の MIN 構造を持つ nD-MIN の方がサイズに対する拡張性，パケット転送の柔軟性において優れていると考えられる．しかし，この点に関しては今後より詳細な評価と比較が必要である．

5. ま と め

本論文では，局所性を利用することのできる MIN である nD-MIN を提案し，転送レイテンシについて簡単な確率モデルによる解析を行った．その結果，ランダム転送においても従来の MIN より有利であり，アクセスに局所性がある場合はレイテンシを 24% 改善できることが分かった．また，エレメントの故障あるいは混雑に対して迂回するルーティングを示し，大きな性能の低下なしに迂回が可能であることが分かった．

今後，基本的な性能に関してより実際的な条件の下で，シミュレーションを行い評価していく一方，適応型ルーティングによる混雑の回避の効果についても具体的に評価を行っていく予定である．また，今回の評価は，MIN は 2，nD-MIN は次元数に応じたクロスポイント数で行っており，公平さを欠く面がある．LSI のピン数等を仮定して，同一コストを仮定して評価する必要がある．

参 考 文 献

1) Broomell, G. and Heath, J.: Classification Categories and Historical Development of Circuit Switching Topologies, *ACM Computing Surveys*, Vol.15, No.2, pp.93–133 (1983).
2) Siegel, H.J. and Smith, S.D.: Study of multistage SIMD interconnection networks, *International Symposium on Computer Architecture*, pp.223–229 (1978).
3) 児玉祐悦，坂井修一，山口喜教：高並列計算機 EM-4 とその並列性能評価，電子情報通信学会論文誌，Vol.J75-D-I, No.8, pp.607–614 (1992).
4) 横田隆史ほか：超並列計算機 RWC-1 の相互結合網，情報処理学会研究報告，93-ARC-101, pp.25–32 (1993).
5) 横田，松岡，岡本，広野，坂井：超並列向け相互結合網 MDCE の提案と評価，情報処理学会論文誌，Vol.36, No.7, pp.1600–1609 (1995).
6) 笹原正司，寺田 純，大和純一，塙 敏博，天野英晴：SSS 型 MIN に基づくマルチプロセッサ SNAIL，情報処理学会論文誌，Vol.36, No.7, pp.1640–1651 (1995).
7) Amano, H., Zhou, L. and Gaye, K.: SSS (Simple Serial Synchronized)-MIN: A novel multi stage interconnection architecture for multiprocessors, *Proc. IFIP 12th World Computer Congress*, Vol.I, pp.571–577 (1992).
8) Adams, III, G.B., Agrawal, D.P. and Siegel, H.J.: Fault-Tolerant Multistage Interconnection Networks, *Computer*, Vol.20, pp.12–27 (1987).
9) Kumar, V. and Reddy, R.: Augmented Shuffle-Exchange Multistage Interconnection Networks, *Computer*, Vol.20, pp.30–40 (1987).
10) Lea, C.: Multi-Log_2N Networks and Their Applications in Highspeed Electronic and Photonic Switching Systems, *IEEE Trans. Comm.*, Vol.38, No.10, pp.1740–1749 (1990).
11) 塙 敏博，天野英晴：多重出力可能な MIN の性能評価，情報処理学会論文誌，Vol.36, No.7, pp.1630–1639 (1995).
12) Murata, A., Boku, T. and Amano, H.: The MDX: A Class of Networks for Large Scale Multiprocessors, *IEICE Trans. Inf. and Syst.*, Vol.E79-D, No.8, pp.1116–1123 (1996).

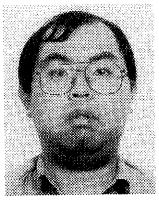
(平成 9 年 11 月 4 日受付)
(平成 10 年 4 月 3 日採録)

塙　敏博（正会員）

平成5年慶應義塾大学理工学部電気工学科卒業．平成10年同大学大学院理工学研究科計算機科学専攻博士課程修了．現在，東京工科大学情報工学科講師．工学博士．並列計算機の相互結合網の解析に興味を持つ．平成8～10年日本学術振興会特別研究員．

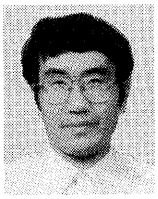
朱　笑岩

平成6年中国沈陽工業大学計算機学院卒業．平成9年慶應義塾大学大学院理工学研究科計算機科学専攻修士課程修了．現在，三菱マテリアル（株）サイバースペース研究所．



亀井　貴之

平成7年慶應義塾大学理工学部電気工学科卒業．平成9年同大学大学院理工学研究科計算機科学専攻修士課程修了．現在，(株）東芝システムLSI技術研究所．



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助教授．工学博士．計算機アーキテクチャの研究に従事．