ハイパーキューブ網の有向グラフ上に構成されるデッドロックフリーなルーティング方式

大宅伊久雄† 小山法孝† 和宇慶康†

並列処理マシンの有力な接続網であるハイパーキューブ網では，ルーティング方式として最下位ビット優先法（e-cube 法）が利用されている．また，一般的なデッドロック防止法としては構造化バッファプール法や仮想チャネル法が知られている．本論文では，ハイパーキューブ網の有向グラフ化によりノード集合全体に順序関係を導入し，デッドロックフリーを保証する新たなルーティング方式（K-法）を提案する．ここでは経路として最短かつ半順方向パスを選択するルーティング関数が，下位次元パスの同形写像を用い，次元数nに関し帰納的に決められる．K-法の通信性能に関連する特性を考察し，（1）経路分布における不均衡性，（2）通信用バッファの効率的な構成法（バーチャル n-キュ，1キュ），（3）パケットフロー制御の優先度決定法（巡回，到着順）について述べる．そしてランダムトラフィックのパケット通信における通信性能(スループット，平均遅延）をシミュレーションにより定量的に調べ，通信手順として1キュ構成の到着順方式が適していること，キュの深さはnの近傍で限界値に収束することを示す．さらに，規則的および不規則的な通信パターンにおいて，e-cube 法と比較し，K-法がより優れた通信スループットを達成することを明らかにする．また，K-法のルーティング方式としての位置付けと今後の課題について報告する．

A Deadlock-Free Routing Algorithm Constructed on a Directed Graph for a Hypercube Network

Ikuo Oyake†, Noritaka Koyama† and Yasushi Wauke†

In order to achieve the deadlock-free property of a routing algorithm, structured buffer pool or virtual channel method has been widely utilized for switching networks. The e-cube routing function for a hypercube network is a well-known example. This paper proposes K-routing method for a hypercube network. The approach to avoid a deadlock occurrence is based on an ordered set of the network nodes, which is derived from the directed graph. The routing function is defined by induction with the use of isomorphic mapping of a directed path. This method allows efficient buffer configuration on each node and realizes flexible packet flow control. Simulation shows that K-method provides better communication throughput than e-cube in packet switching with random traffic.

1. はじめに

最近，いろいろな応用分野で並列処理マシンの利用が予想以上の進展を見せている．アプリケーションプログラムの並列実行において，処理と通信のバランスを確保することは重要である．その意味で並列処理マシンの相互接続網に関する技術[1)~5)]は常に古くて新しいテーマと思われる．筆者らは，科学技術計算用の可視化処理[6)]にハイパーキューブ網の並列処理マシンを開発した[7)~9)]．ここでは，16台のノードを幾何計算処理とピクセル処理の2グループに分割し，大量の表示データをパケット転送する機能パイプラインの並列モデルを設定した．このような通信形態では，パケット通信における高いスループットが要求される．さらに，アプリケーションのチューニング時にはパケット長やバッファ数の調整が有効な手段となる．そこで，パケット通信のルーティング法を決めるに当たり，パケットを蓄積，転送する通信用バッファの構成法について着目した．

ハイパーキューブ網[10),11)]のルーティング方式として，一般には最下位ビット優先法（e-cube 法）が利用されている．また，ルーティングにおけるデッドロック問題を防止するために，構造化バッファプール法[12),13)]や仮想チャネル法[14)~17)]が知られている．これらの方式はデッドロックの対象となる資源（バッファ，チャネル）に順序関係を導入することで資源間にループが形成されることを回避している．すなわち，e-cube 法では送信チャネルの次元が増加方向に選択される．また，その前提として各チャネルごとにキュ（バッ

† 沖電気工業株式会社研究開発本部マルチメディア研究所
Media Laboratory, R&D Group, Oki Electric Industry Co., Ltd.

ファ）を確保する必要がある．構造化バッファプール法では，ホップ数（転送ステップ）により使用するバッファカウンタがインクリメントされるので，各ノードに次元数分のバッファを必要とする．結果的に従来方式は，デッドロックフリーを保証するため各ノードに最小限次元数分のバッファをもち，個々のバッファの使用はチャネル番号やホップ数に従属した制限を受けることになる．

本論文では，ハイパーキューブ網のパケット通信において，より効果的なバッファ構成を可能とするルーティング方式（K-法）を提案する．この方式は，仮想チャネル法と同様に，ルーティングを制限する考えに基づく．相違は，チャネルではなく，ノード集合に順序関係を導入することでデッドロックを回避する点にある．2章では，可視化処理マシン（次元数 $n=4$）に実装したルーティング法[8]を n 次元ハイパーキューブ網のルーティング方式として一般化する．あわせてデッドロックフリーな理論的根拠を明確にする．3章においては，従来法（e-cube 法）との比較という視点から，通信性能に関わる本方式の特性を考察する．そしてマイナス要因としてルーティングの不均衡性を調べ，またプラス要因として効率的なバッファ構成法（バーチャル n-キュ，1キュ）と送受信パケット優先権の決定方式（巡回，到着順方式）について述べる．さらに4章では，これらの特性が通信性能に及ぼす影響をシミュレーションにより定量的に評価し，本方式の効果的な利用法について結論を述べる．

2. ルーティング方式

ハイパーキューブ網のプロセッサと通信チャネルを，それぞれグラフ理論[18]のノードとリンクに対応させる．通信用バッファは，仮想チャネル法ではリンクに属するが，K-法ではノード側に置く．その意味で構造化バッファ法のバッファグラフに相当する．

本章では提案するルーティング方式を以下の展開で説明する．2.1 節では有向ハイパーキューブと同形写像を定義し，ノードに順序関係を与える．この順序はハミルトンパス（または Gray Code）でも定義される．2.2 節では同形写像を用いてデッドロックフリーを保証する経路を定義し，2.3 節にその経路計算アルゴリズムを示す．

2.1 ノード集合の順序付け

通信用バッファ間のデッドロックを防止するため，従来法より強い条件として，ノード集合に順序関係を導入する．そのためハイパーキューブ網の各リンクに方向を付与し有向グラフを構成する．この方向はデータ転送を片側に制限するものではない．この有向グラフを以下有向ハイパーキューブと呼ぶ．これは，ハイパーキューブと同様に2進表記を用い，次元に関し帰納的に定められる．

ここで，以下に使用する記法を説明する．n 次元の有向ハイパーキューブを $H[n]$ とし，そのノードアドレスを $X_nX_{n-1}\cdots X_i\cdots X_1$ で表す．$H[n]$ のなかで，アドレスの第 i ビットが a であるノード全体で張られる $n-1$ 次元部分ハイパーキューブを $sH[n,(i,a)]$ とする．また，ベクトル $I_k=((i_k,a_k),\cdots(i_j,a_j),\cdots,(i_1,a_1))$，$i_k\langle\cdots\langle i_j\cdots\langle i_1$ に対し，アドレスの第 i_j ビットが a_j であるノード全体で張られる $n-k$ 次元部分ハイパーキューブを $sH[n,I_k]$ とする．また，集合 S から集合 T への写像 $y=f(x)$ を $f:S\Rightarrow T, f(x)=y$ と記す．

定義 1)　有向ハイパーキューブ $H[n]$

（1）　$H[1]$は1つの有向線分で，その始点をアドレス0，終点をアドレス1とする．

（2）　$H[n]$, $n\geqq 2$ は以下のように定義する．$H[n-1]$ のコピーを2つ用意する（それぞれ H, H' とする）．両者の等しいアドレスのノード間を線分で結ぶ．H のノードにはアドレスの左端に0をつけ（$0X_{n-1}\cdots X_1$），H' には1をつけ（$1X_{n-1}\cdots X_1$），それらを新アドレスとする．最後に H と H' を結ぶ線分に方向をつける．これは新アドレスに関し，1が偶数個あるノードが始点となるように方向を選ぶ．このようにして $H[n]$ が定義される．

図1に $H[2]$ から $H[3]$ への合成過程を示す．図の $H[3]$ は，下位の2次元部分ハイパーキューブを6個包含する．これらの部分ハイパーキューブは，それぞれ定義1に従う $H[2]$ と有向グラフとして同形となる．一般に次のことがいえる．$H[n]$ は $2n$ 個の $n-1$ 次元部分ハイパーキューブを包含する．これ

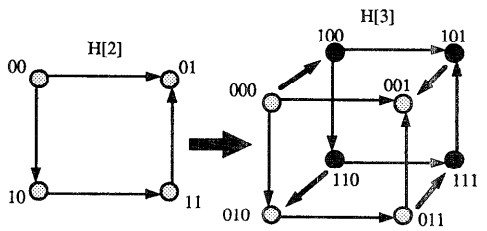


図 1　有向ハイパーキューブ
Fig. 1　Directed hypercube.

らは，i $(1 \leqq i \leqq n)$ ビット目が定数 a $(=0, 1)$ の $X_n \cdots X_{i+1} a X_{i-1} \cdots X_1$ なるノード群で張られる．この部分ハイパーキューブを $sH[n, (i, a)]$ と表す．これは以下に示す写像により，$H[n-1]$ と有向グラフとして同形となる．

（1） $im[n-1, (i, 0)] : H[n-1] \Rightarrow sH[n, (i, 0)]$,

$im[n-1, (i, 0)](X_{n-1} \cdots X_{i+1} X_i X_{i-1} \cdots X_1)$

$= X_{n-1} \cdots X_{i+1} X_i 0 X_{i-1} \cdots X_1$.

（2） $im[n-1, (i, 1)] : H[n-1] \Rightarrow sH[n, (i, 1)]$,

ⅰ）$i \neq n$ のとき，

$im[n-1, (i, 1)](X_{n-1} \cdots X_{i+1} X_i X_{i-1} \cdots X_1)$

$= X_{n-1} \cdots X_{i+1} \overline{X}_i 1 X_{i-1} \cdots X_1$.

ⅱ）$i = n$ のとき，

$im[n-1, (i, 1)](X_{n-1} \cdots X_{i+1} X_i X_{i-1} \cdots X_1)$

$= 1 X_{n-1} \cdots X_{i+1} X_i X_{i-1} \cdots X_1$.

この同形写像 im は，i ビット目に 0 または 1 を挿入，それ以上のビットを左シフトし，$a=1$ かつ $i \neq n$ のとき X_i を反転することを意味する．また，im の表記はノード対応で示されているが，これを有向ハイパーキューブの部分有向グラフ dS の対応 $(im[\cdots](dS) = dT)$ に自然に拡張することができる．

$H[n]$ 内の任意の $n-k$ $(1 \leqq k < n)$ 次元部分ハイパーキューブは，上記 sH の第2パラメータである (i, a) をベクトル指定 $Ik = ((i_k, a_k), \cdots\cdots, (i_1, a_1)), i_k < \cdots i_2 < i_1$ に拡張し，$sH[n, I_k]$ と表現する．さらに $H[n-k]$ との同形写像は，上記 im のパラメータ (i, a) を同じ I_k 指定に置き換えることにより，以下のように帰納的に表現できる．

$im[n-k, I_k] : H[n-k] \Rightarrow sH[n, I_k]$,

$im[n-k, I_k](dS) =$

$im[n-(k-1), I_{k-1}](im[n-k, (i_k, a_k)](dS))$.

次に $H[n]$ が順序集合となることを示す．$H[n]$ を最下位ビットにより，$sH[n, (1, 0)]$ と $sH[n, (1, 1)]$ に分割する（**図2**参照）．ここで2集合間を連結するリンクに注目する．$H[1]$ の定義によりリンクはすべて同一方向 $(X_n \cdots X_2 0 \rightarrow X_n \cdots X_2 1)$ に向かっている．この方向より，集合間に順序関係が付けられる．この最下位ビット分割を，同形関係を考慮しながら，最上位ビットまで再帰的に適用することにより，ノードの順序列が得られる．このノード列は，$H[n]$ の順方向ハミルトンパスとなる．ハミルトンパスとはすべてのノードを一回通るパスであり，順方向とはパスの向きがリンク方向と一致することを意味する．3次元順方向ハミルトンパスは，000→100→110→010→011→111→101→001 である．

以上から，次の命題が成立する．

命題 1）有向ハイパーキューブのノード集合全体は順序集合となる．

順方向ハミルトンパスのノード順序は，以下に定義する Gray Code によっても与えられる．

定義 2）n ビットの Gray Code[19] は以下のように帰納的に定義される．

（1） $\mathrm{Gray}(n) = (G_1, G_2, \cdots, G_{2^n-1}, G_{2^n})$ とする．

（2） $\mathrm{Gray}(n+1) = (G_1 0, G_2 0, \cdots, G_{2^n-1} 0, G_{2^n} 0,$

$G_{2^n} 1, G_{2^n-1} 1, \cdots, G_2 1, G_1 1)$.

2.2 ルーティング関数

経路をすべて順方向パスから選択できれば，デッドロックの要因となるバッファ間のループ形成は回避される．しかし，最下位ビット分割の集合 $sH[n, (1, 1)]$ から $sH[n, (1, 0)]$ への経路を順方向パスで結ぶことはできない．そのため，順方向パスに関する両端リンクでの条件を緩めたパスを以下に定義する．

定義 3）半順方向パス

パスの始点と終点の両端リンクを除き，すべての中継リンクでパスの向きがリンク方向と一致するパスを半順方向パスと呼ぶ．

半順方向パスの例を**図3**に示す．順方向パスは半順

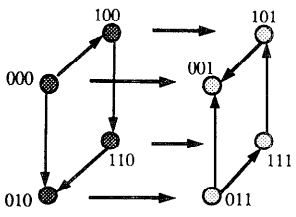


図2 $H[3]$ 最下位ビット分割
Fig. 2 Decomposition $H[3]$ by LSB.

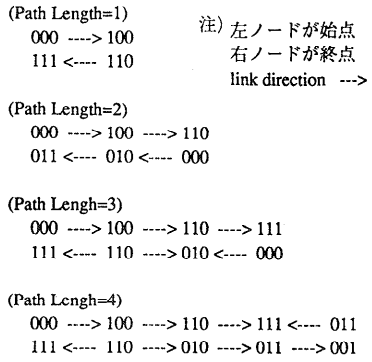


図3 $H[3]$ 内の半順方向パスの例
Fig. 3 Examples of semi-oriented path in $H[3]$.

方向パスの範疇に含まれる．経路選択を半順方向パスにひろげると，任意ノード間を最短経路で結ぶことができる（以下のルーティング関数で例を示す）．順方向から半順方向への条件の緩和は，通信用バッファの使い方に以下の制約を課すことで，デッドロックフリーを維持できる．すなわち，経路の両端ノードではパケットを通信用バッファに格納せず，別のバッファ（入出力バッファ—3.3節で説明）を使うという制限を設けることにする．通信用バッファは，中継ノードでのパケット格納領域として扱う．

次に，n 次元ハイパーキューブ網のルーティング関数を定式化する．ルーティング関数とは，始点ノードと終点ノードに対し，その間のパスを対応させるものである．はじめに，距離 n のノード間の経路を決める関数を定義する．ここで半順方向かつ最短パスの集合を $\{dP\}$ と表記する．

定義 4) $r[n]: H[n] \Rightarrow \{dP\}$,

（1）$r[1](0)=0\rightarrow 1, r[1](1)=1\leftarrow 0$

（2）$r[2](00)=00\rightarrow 10\rightarrow 11$,
$r[2](10)=10\rightarrow 11\rightarrow 01$,
$r[2](11)=11\rightarrow 01\leftarrow 00$,
$r[2](01)=01\leftarrow 00\rightarrow 10$.

（3）$r[n](V), n\geqq 3$ は，$H[n]$ の最下位ビット分割を2回行った部分集合に分けて，下位次元パスの同形写像とパスの結合（＋＋）を利用し，以下のように帰納的に定義する．

$dP=r[n-1](X_{n-1}\cdots X_20)$ とおく．

$r[n](X_{n-1}\cdots X_200)=im[n-1,(1,0)](dP)++$
$\bar{X}_{n-2}\cdots\bar{X}_210\rightarrow\bar{X}_{n-2}\cdots\bar{X}_211$,

$r[n](X_{n-1}\cdots X_210)=X_{n-2}\cdots X_210\rightarrow X_{n-2}\cdots X_211$
$++im[n-1,(1,1)](dP)$,

$r[n](X_{n-1}\cdots X_211)=im[n-1,(1,1)](dP)++$
$\bar{X}_{n-2}\cdots\bar{X}_201\leftarrow\bar{X}_{n-2}\cdots\bar{X}_200$,

$r[n](X_{n-1}\cdots X_201)=X_{n-2}\cdots X_201\leftarrow X_{n-2}\cdots X_200$
$++im[n-1,(1,0)](dP)$.

パス結合＋＋は単に，一致する終点と始点を重ねる操作を意味する．$r[2](X_20)$ は順方向パスであり，帰納法定義による $r[n](X_{n-1}\cdots X_2X_10)$ も，結合される長さ1のパスが順方向であることから順方向パスとなる．また，$r[n](X_{n-1}\cdots X_211)$ と $r[n](X_{n-1}\cdots X_201)$ は，それぞれ第 n ステップと第1ステップでリンク方向と逆になるので半順方向パスとなる．経路は定義より最短パスである．

定義4を用いて任意ノード間 (V, V') のルーティング関数 $R(V, V')$ を求める．V, V' の張る最小部分ハイパーキューブを $sH[n, I_k]$ とすると，$R(V, V')$ は次のようになる．

定義 5) $R: H[n]\times H[n]\Rightarrow\{dP\}$,

（1）V, V' の距離が n のとき，$R(V, V')=r[n](V)$,

（2）その他は，$R(V, V')=im[n-k, I_k]$
$(r[n-k](im^{-1}[n-k, I_k](V)))$.

この関数で決まる経路は，$r[n-k]$ の同形写像を用いることから半順方向パスかつ最短パスとなる．

2.3 経路計算アルゴリズム

前節では，同形写像を使用してノード間の経路を定義した．ここでは順方向ハミルトンパスによるノード集合の順序関係を直接的に利用した経路の計算アルゴリズムを示す．計算アルゴリズムとは，送信中パケットのカレントノードとその終点ノードから，次の転送ノードを算出する方法である．ノード間の順序を$\rangle\rangle$または$\langle\langle$で表し，$V\rangle\rangle V'$ とは V から V' への順方向パスがあることを意味する．任意の2ノードが与えられたとき，順序比較は定義2の Gray Code を利用して計算できる．

アルゴリズム）

パケットのカレントノードを $Vc=C_n\cdots C_i\cdots C_1$，送信先終点ノードを $Vd=D_n\cdots D_i\cdots D_1$ とする．

$Eor=E_n\cdots E_i\cdots E_1$, $E_i=C_i\oplus D_i$（$\oplus$ は排他的論理和），

$B(Vc, Vd)=\{m\mid E_m=1\}$,

$A(Vc, Vd)=\{m\in B(Vc, Vd)\mid Vc\rangle\rangle C_n\cdots\bar{C}_m\cdots C_1\}$

とおく．$B(Vc, Vd)$ は両ノードのアドレスの異なるビット番号（または次元）を示し，$A(Vc, Vd)$ はそのなかで順方向のリンクで繋がる隣接ノードへの次元を示す．

次の送信ノード $Vnext$ を以下のように選択する．

$Vc'=C_n\cdots\bar{C}_{\min B(Vc,Vd)}\cdots C_1$,

$Vc''=C_n\cdots\bar{C}_{\max A(Vc,Vd)}\cdots C_1$ とおくと，

（1）$Vc'\rangle\rangle Vd$ のとき，$Vnext=Vc'$,

（2）$Vc'\langle\langle Vd$ のとき，$Vnext=Vc''$,

（3）$Vc'=Vd$ のとき，$Vnext=Vd$.

以上のように選択したノード列は定義5のルーティング関数で決められる経路と同一になる．

3. 方式の特性に関する考察

e-cube 法と対比させ，K-法の基本的な特性を考察し，あわせて実装上の留意点を明確にする．

3.1 ルーティングの空間的特性

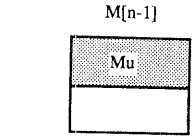
K-法で定義したルーティングの空間的特性を調べ

M[2]

	00	10	11	01
00	1	1	1	0
10	0	1	1	1
11	1	0	1	1
01	1	1	0	1

M[3]

	000	100	110	010	011	111	101	001
000	1	1	1	0	0	1	0	0
100	0	1	1	1	1	0	0	0
110	0	0	1	0	0	1	1	1
010	0	0	0	1	1	1	1	0
011	0	1	0	0	1	1	1	0
111	1	0	0	0	0	1	1	1
101	0	1	1	1	0	0	1	0
001	1	1	1	0	0	0	0	1



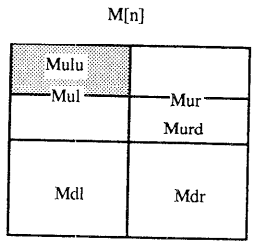


図 4 パスマトリックス $M[n]$ の構成
Fig. 4 Structure of path matrix $M[n]$.

るために，網内の各ノードにおいて経路と交差する回数（経路の始点と終点を含む）を求め，これらを経路の分布値と定義する．K-法の基本関数である $r[n]$ で定義される経路について，その分布値を以下に算出する．

分布マトリックス $M[n]$ を定義する．マトリックスの行と列は，有向ハミルトンパスの順序によるノード列 $(Vi, i=1, 2, \cdots, 2^n)$ を選ぶ．要素 $M[n](i, j)$ は，ノード Vi から距離 n の経路がノード Vj を通過するとき1，それ以外は0とする．このとき，ノード Vk の分布値は以下の式で定義される．

$$D[n](k)=\Sigma M[n](i, k),\ i=1, 2, \cdots, 2^n.$$

$M[2], M[3]$ を**図4**に示す．一般に，$M[n]$ は $r[n]$ の帰納的定義から $M[n-1]$ により決まる．以下にこの関係をみる．図4に示すように，$M[n]$ の行と列をそれぞれ上下 (u, d) と左右 (l, r) に2等分し，それぞれの部分マトリックスを Mul, Mur, Mdl, Mdr とする．さらに，Mul の上半分を $Mulu$, Mur の下半分を $Murd$ とする．また，$M[n-1]$ の上半分を $Mu[n-1]$ とすると，次の関係が成り立つ．

$$Mulu=Mu[n-1],$$
$$Murd(p', q)=Mulu(p, q),\ p'=2^{n-2}-p+1,$$
$$Mul=Mdr,\ Mur=Mdl.$$

また，$Mu[n-1]$ の上半分の分布値を $d[n-1]$ とする．

$$d[n-1](k)=\Sigma Mu[n-1](i, k),\ i=1, 2, \cdots, 2^{n-1}.$$

$D[n](k)$ は以下のように帰納的に求められる．

（1）$n=2$ のとき，

$$d[2]=(1, 2, 2, 1),\ D[2]=(3, 3, 3, 3).$$

（2）$n \geqq 2$ のとき，

$$d[n]=(d[n-1], d[n-1])+(E0[n-2], E1[n-2], E1[n-2], E0[n-2]),$$
$$D[n]=2(d[n-1], d[n-1])+(E1[n-1], E1[n-1]).$$

ここで $E0[n]$, $E1[n]$ は，それぞれ要素が0，1の 2^n 次元ベクトルである．

以上の式で表現される距離 n の経路分布は，ノード列 (Vi) を2等分した部分集合間で分布値が同一パターンとなること，またその部分集合をさらに2等分した部分集合間で分布値が対称パターンとなることを示す．これは $r[n]$ が最下位ビット分割を2度行った部分集合で同形なパスを用い定義されていることと符合する．また，次元数が上がるに従い，この4分割集合内で分布値の不均衡性が線形に増加する．一方，e-cube 法の距離 n の経路分布は各ノードで同じ値 $(n+1)$ となる．K-法のこの特性は，例えば全対全通信のように通信相手が網内で均等に存在する規則的な通信パターンにおいて，パケット転送の負荷がノードごとにアンバランスとなることを意味する．通信性能の観点から，この特性は明らかにマイナス要因と考えられる．これに関する定量的な評価は，次に述べる2つのプラス要因を加味して，4章にて規則的な通信と不規則的な通信の両者において行うことにする．

3.2 通信用バッファの構成法

図 5-(1)に e-cube 法を利用した場合のバッファ構成例を示す．この例ではバッファは各送信チャネルに固定され，深さ1の n-キュを構成している．K-法においては，ノードの通信用バッファ間にデッドロックフリーが保証されているので，通信用バッファの構成はこのようにチャネルに従属させる必要はない．このことは，各ノードに設けられた m 個のバッファをそのノードに共通な資源として扱えることを意味する．例えば n-キュの変形として，バッファを n 個の header を先頭とするリスト構造の共通要素とするバーチャル n-キュ構成が考えられる（図5-(2)）．この構成では，中継ノードに到着したパケットに対し，空きバッファに関する動的な割り当てが可能となる利点を持つ．K-法のバッファ構成上の注意点は，半順方向パスの選択で述べたように，通信用バッファとは別に入出力バッファを設け，送信元（始点）のパケットは出力バッファから直接ポートに転送し，受信

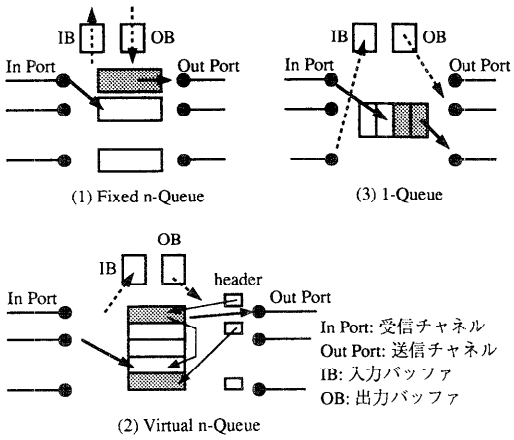


図 5 バッファ構成法
Fig. 5 Buffer configuration.

先（終点）のパケットはポートから直接入力バッファに転送する手順が必要となる．

K-法ではバーチャル n-キュ構成をさらに単純化して，深さ m の1キュ構成（図5-(3)）を採用することが考えられる．この構成は FIFO メモリを使い簡単に実装できる．バッファ構成法の選択は，通信手順と関連するので次節でこの点を考察する．また，バッファ数 m に関しては，得られる通信性能の観点から，次章で定量的に評価することにする．

3.3 通信手順

ハイパーキューブ網の通信方式は，実現に要するハードウェア量から判断して，現実的な1ポート通信を前提とする．1ポート通信では，各ノードにおいて同時に最大1送信と1受信機能が並行動作する．隣接ノード間での通信リンク確立は，e-cube 法と同様に，簡易な Request-Ack 型を適用する．Request 処理では，通信用バッファと出力バッファ内の送信待ちパケットから次に送信するパケットを選択し，その送信チャネルに送信要求（Request）を送出する．また Ack 処理では，受信バッファの空き状態を確認した上で，送信要求があるチャネルから受信相手を選択し，相手ノードに応答（Ack）を返送する．

以上の手順において，パケットのフロー制御に関する方針，すなわち送受信パケットの優先権決定方法を定めておく必要がある．e-cube 法では通常チャネルに対する巡回方式（Round Robin）が使われる．K-法においてもバッファ構成をバーチャル n-キュとして巡回方式を採用することができる．K-法ではさらに1キュ構成による到着順方式（First In First Out）の適用が可能となる．1ポート通信の前提では，e-cube 法への到着順方式の適用は明らかにデッドロック状態を生起し不可能である．K-法での注意点として，通信用バッファと1ポート通信の送信権にまたがるループ形成の問題を回避するため，通信用バッファがフルのノードでは，出力バッファより通信用バッファのパケットを優先する必要がある．

K-法のルーティング処理について述べる．2.2節のルーティング関数 R は，始点と終点のペア (V, V') から再帰的な演算により経路が定まる．そのため，各ノードでは前処理として経路をパスのエクスチェンジ置換列としてテーブル化し，パケットの発生ノードで置換列をパケット header に埋め込む方法がある．この場合中継ノードでのルーティング処理が簡単になる反面，高次元化に対しテーブルが大きくなる欠点をもつ．別の方法として，2.3節のアルゴリズムを直接実装することが考えられる．パケット header には終点ノードのアドレスが持たれ，転送経路の各ノードで次送信ノードを計算する．このとき，Gray Code の順序比較演算をハードウェア化しておくと経路計算が高速化される．

以上述べた K-法の巡回，到着順方式，および e-cube 法の定量的評価を次章で行う．

4. 通信性能に関する総合評価

K-法の原形を実装した可視化処理マシン（$n=4$, $m=1$）では，その固定的な実装法から，前章で述べた通信手順やバッファ構成の選択肢を定量的に評価することは困難である．また，経路分布の不均衡性の影響はできるだけ高い次元で評価する必要がある．これらの理由から，以下に述べるシミュレータを構築した．

4.1 シミュレータの内容

n（$3 \leqq n \leqq 8$）次元ハイパーキューブ網のパケット通信につき，シミュレーションにより通信スループットとパケット別遅延時間の平均値を求める．通信パターンとしては規則的な全対全通信と不規則的なグループ間通信を扱う．グループ間通信とは，可視化処理の分散サーバ型通信パラダイム[7]にみられ，ハイパーキューブ H を2つの部分集合 H_1 と H_2 に分割し，H_1 の全ノードから H_2 の全ノードにパケット転送する形態を意味する．ここでは H_1 と H_2 のノード数の比を 1:1, 3:1, 7:1 に変化させる．各ノードでの発生パケットの送信順は規則性がなくランダムとし（ランダムトラフィック），全域的な最適スケジュリング[19)~21)]

は考えない．シミュレーションでは，通信パターンに従って送信パケット群をはじめに用意し，乱数によって送信順を決める．ランダムトラフィックは発生パケットの内容（例えば可視化処理では3次元ポリゴンが投影される画面座標の位置）により相手先ノード（送信範囲内で動的に変化）が決まる通信形態にみられる．

シミュレータは，全ノード同一サイクルで隣接ノード間のパケット送受信を行う同期制御方式で構成した．通信スループットと平均遅延時間の算出法を以下に説明する．一般に，全パケットを転送するに要する通信時間 T は，

$$T=(TS+TP)\times(LN/(ML\times LA))$$

TS ：通信リンク設定時間
TP ：データ転送時間
LN ：全パケットの延転送時間
LA ：リンク稼働率
ML ：1サイクルの最大動作可能総リンク数
（1ポート通信では $ML=2^n$）

で表現される．K-法（巡回，到着順方式）と e-cube 法（巡回方式）では，通信リンク設定に同じ Request-Ack 型の手順を使うことから TS を同一値とし，$TS+TP=1$ として総サイクル数を求める．そして上式よりリンク稼働率 LA を算出する．このリンク稼働率は通信スループットに比例した値となる．一方，平均遅延時間は，全パケットの転送を個別に監視し，各パケットが相手ノードに到着するまでのサイクル数（中継ノードでの待ちサイクル数を含む）を求め，全パケットに対しその平均値を算出する．リンク稼働率と平均遅延時間は，それぞれネットワーク負荷率を変数として求める．この負荷率とは，送信元ノードにおいて，通信用バッファと別に構成した出力バッファ（FIFO で構成）の先頭にパケットが存在する割合（サイクルごとの出現率）として定義し，20, 40, 60, 80, 100% において算出する．

4.2 シミュレーション結果とその評価

結果は，ネットワーク負荷率を横軸に，リンク稼働率（例は**図 6, 7, 8, 9**）と遅延時間（例は**図 10, 11**）をそれぞれ縦軸にプロットする．K-法の巡回と到着順方式について，3,…,8 次元かつ4通信パターンの全組み合わせにおいて評価するが，プロット図に関しては特徴的なものを選んで示す．各図ではバッファ数をパラメータとした性能値を示す．また，比較の基準として e-cube 法（$m=n$）の場合が併記されている．

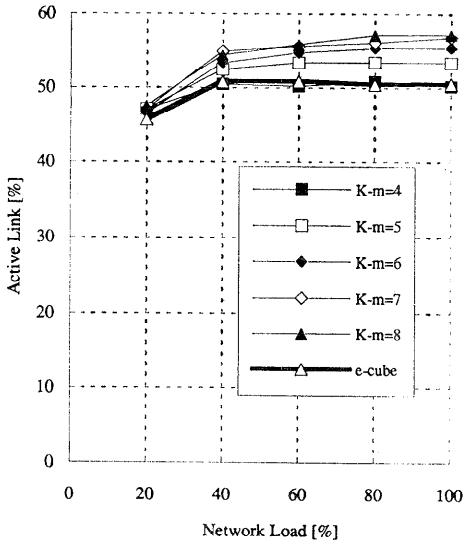


図 6 K-巡回方式のリンク稼働率（$n=6$, 全対全）
Fig. 6 Active link ratio by K-RR ($n=6$, all to all).

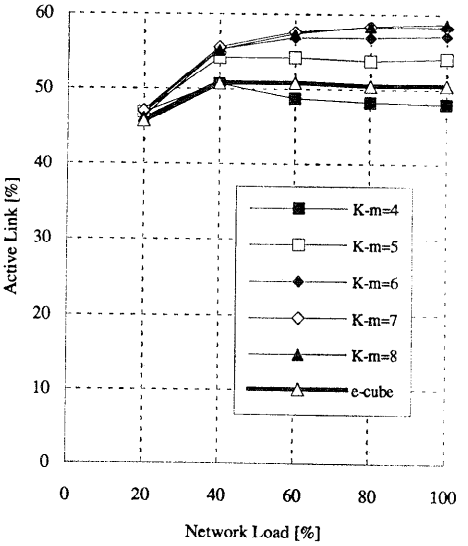


図 7 K-到着順方式のリンク稼働率（$n=6$, 全対全）
Fig. 7 Active link ratio by K-FIFO ($n=6$, all to all).

（1） リンク稼働率

巡回と到着順方式ともに，バッファ数 m が増えるにしたがい稼働率は上昇する．6次元までは，両方式は $m=n$ の近傍で同程度に収束する（図6，7）．7，8次元での限界値への収束は到着順の方が早い（図8，9）．この収束は，バッファ資源のフルが原因で

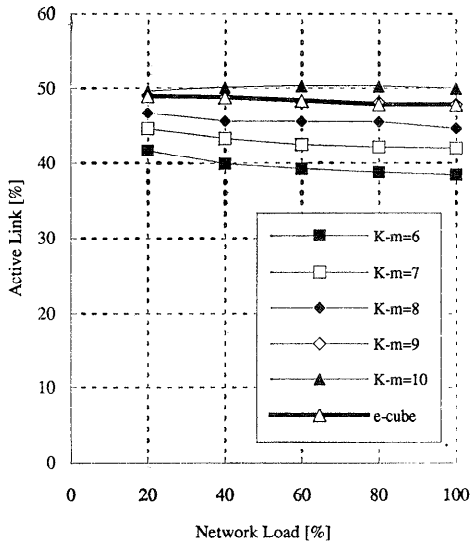


図 8 K-巡回方式のリンク稼働率 (n=8, 全対全)
Fig. 8 Active link ratio by K-RR (n=8, all to all).

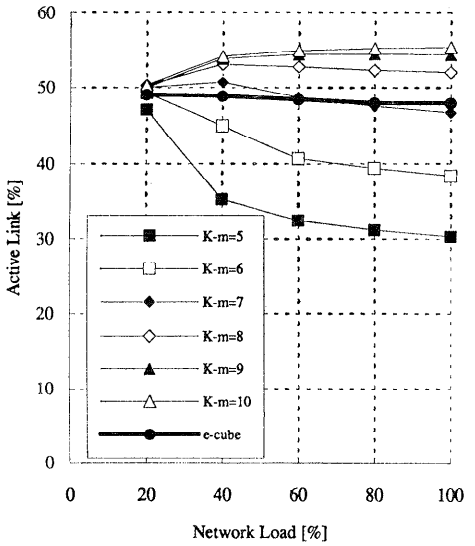


図 9 K-到着順方式のリンク稼働率 (n=8, 全対全)
Fig. 9 Active link ratio by K-FIFO (n=8, all to all).

通信リンク確立が不成功になる場合が解消されることによる．**表 1** に 4 通信パターンでの収束値 m を示す．ここでは稼働率の高い方を数値の網かけで示している．

（2） 遅延時間

さまざまな負荷率において，平均的に少ない遅延時

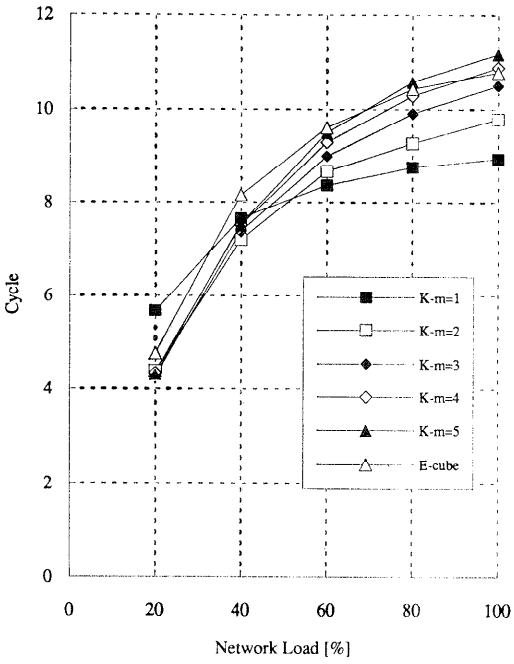


図 10 K-巡回方式の遅延時間 (n=5, 全対全)
Fig. 10 Delay by K-RR (n=5, all to all).

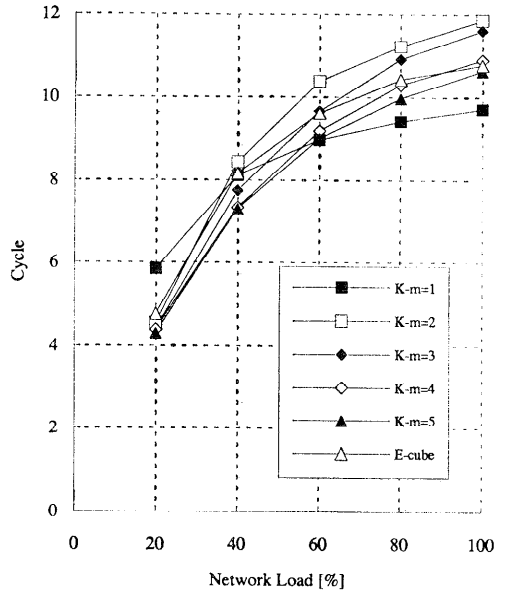


図 11 K-到着順方式の遅延時間 (n=5, 全対全)
Fig. 11 Delay by K-FIFO (n=5, all to all).

表 1 リンク稼働率に最適なバッファ数
Table 1 Number of buffers optimized for active link ratio.

		3	4	5	6	7	8
all:all	FIFO	3	4	6	7	8	9
	RR	3	4	8	9	9	11
group (1:1)	FIFO	2	3	4	5	6	8
	RR	2	3	4	5	8	11
group (3:1)	FIFO	3	3	4	4	7	9
	RR	3	3	4	7	9	9
group (7:1)	FIFO	3	5	7	8	9	10
	RR	2	3	5	8	8	11

表 2 遅延時間に最適なバッファ数
Table 2 Number of buffers optimized for delay.

		3	4	5	6	7	8
all:all	FIFO	1	1	5	7	8	9
	RR	1	1	1	2	2	2
group (1:1)	FIFO	2	2	2	4	4	5
	RR	1	1	2	2	2	2
group (3:1)	FIFO	1	1	1	1	1	1
	RR	1	1	1	2	2	2
group (7:1)	FIFO	1	1	1	1	1	1
	RR	1	1	1	1	2	2

間を与えるバッファ数 m を**表 2**に示す．同様に数値の網かけで遅延時間が良い方を示している．巡回方式では，負荷が重いところ（60% 以上）で m とともに遅延時間は増加する（図 10）．一方，負荷が軽いところ（20%）では m の増加とともに遅延時間は減少する（特に $m=1$ から 2）．また，到着順方式では，5 次元以上の全対全と 1：1 の通信パターンで異なった傾向がみられ，負荷率 60% 以上で，遅延時間は $m=1$ から一度は増加するが，途中から逆に減少し $m=n$ の近傍で収束する（図 11）．遅延時間に関するこれらの現象を以下に解析する．

一般に，距離 k の通信における遅延時間 D は，k ホップ（転送ステップ）の転送があることから，

$D(k)$＝1 ホップの平均遅延時間×k

となり，1ホップの平均遅延時間 Dh を分解すると，

Dh＝（1ホップの平均送信数）×
（1送信に要する平均サイクル数）

となる．ここで1ホップの平均送信数（Ms）とは，当該パケットがあるノードに到着後次のノードに転送されるまでに，他の優先パケット（巡回，到着順方式で異なる）の送信を含めトータル何回の送信オペレーションが行われたかの平均値である．Ms はノードにおける送信待ちパケットの競合度を表す指標とみなせる．また，1送信に要する平均サイクル数（Mc）には，隣接ノードの受信権を獲得するのに待たされるサイクル数も含まれ（$Mc \geqq 1$），Mc の逆数（$1/Mc$）は

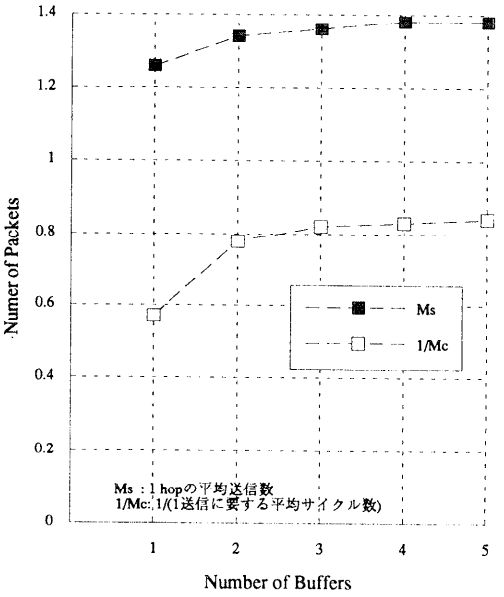


図 12 遅延時間の評価（負荷20%，巡回）
Fig. 12 Delay evaluation by K-RR (load 20%).

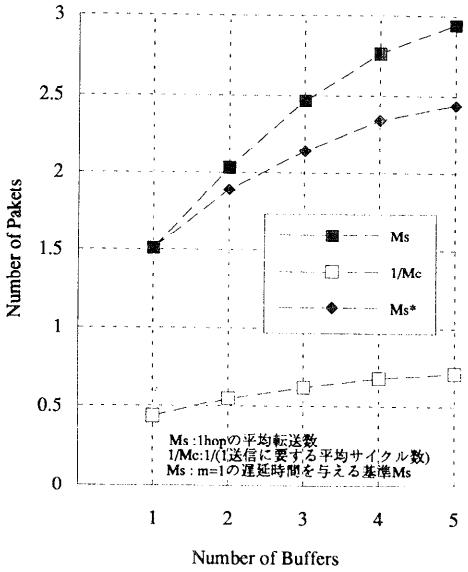


図 13 遅延時間の評価（負荷 80%，巡回）
Fig. 13 Delay evaluation by K-RR (load 80%).

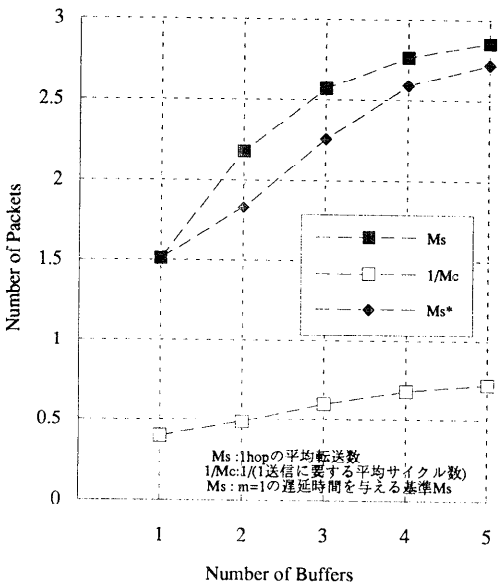


図 14　遅延時間の評価（負荷 80%，到着順）
Fig. 14　Delay evaluation by K-FIFO (load 80%).

1サイクルで成立する送信数となり，通信速度を表す指標とみなせる．この *Ms*, 1/*Mc* 値を，$n=5$ の全対全通信について，シミュレーションで求めた結果を**図 12, 13, 14** に示す．いずれも全体の平均遅延時間の傾向を代表する距離 $k=3$ について算出している．図 12 は，負荷率 20% の巡回方式でバッファ数 *m* を横軸に，*Ms* と 1/*Mc* の送信数を縦軸にプロットしている（補間した点線は変化率を表す）．$m=1$ から 2 へのバッファ数増加に対し，バッファフルの状況が緩和され 1/*Mc* 値が急激に向上するのに比べ，パケットの競合度を表す *Ms* の増加は負荷が軽いことから穏やかな増加に留まっており，この結果，*Ms* と *Mc* の積に比例する遅延時間は $m=1$, 2 の間で減少することになる．

図 13, 14 では負荷が重い場合（80%）に，巡回と到着順方式のそれぞれの *Ms*, 1/*Mc* 値を示す．1/*Mc* 値は両方式ともほぼ同じ傾向を示す．*Ms* 値の変化率をみるため，それぞれの 1/*Mc* の変化に応じて $m=1$ と同じ遅延時間を与える基準 ***Ms**** を図に併記し，実際の *Ms* 値との差分を比較する．巡回方式では差分が単調に増加している．すなわち，資源利用の分散を方針にパケットを分配する巡回方式は，*m* とともに単調にパケットの競合度を増加させている．一方，到着順方式では，$m=2$ で差分が最大になり以降は減少する．すなわち，*m* が小さい範囲で競合度が敏感に立ち上がり，$m=n$ に近づくに従い定常状態に収束する特性をもっているといえる．

表 3　固定バッファ数でのリンク稼働率の比較評価
Table 3　Comparison of active link ratio for a fixed number of buffers.

		3	4	5	6	7	8
all:all	FIFO	●	●	●	●	●	●
	RR						
	e-cube						
group (1:1)	FIFO	●	●	●	●	●	●
	RR						
	e-cube						
group (3:1)	FIFO	●	●	●	●	●	
	RR						●
	e-cube						
group (7:1)	FIFO	●	●	●	●	●	
	RR						●
	e-cube						

表 4　固定バッファ数での遅延時間の比較評価
Table 4　Comparison of delay for a fixed number of buffers.

		3	4	5	6	7	8
all:all	FIFO	●	●	●	●	●	
	RR						
	e-cube						●
group (1:1)	FIFO	●	●	●	●	●	●
	RR						
	e-cube						
group (3:1)	FIFO		●	●	●		
	RR					●	●
	e-cube	●					
group (7:1)	FIFO						
	RR					●	●
	e-cube	●	●	●	●		

（3）　総合評価

以上のことより，通信スループットを優先する立場からは，バッファ数 *m* を *n* の近傍に設定することが有効といえる．$m=n+1$ における K-法の巡回と到着順方式，および e-cube 法との比較を**表 3**（リンク稼働率），**表 4**（遅延時間）に示す．各次元につき，最も良い値を与える方式にマークが付けられている．これにより全体的に到着順方式の有効性が明らかになる．

K-法は経路分布に不均衡性を有するが，バッファ構成の柔軟性から通信スループットに優れた特性を提供する．**図 15, 16** に到着順と e-cube 法の稼働率の差異が 1：1，3：1 のパターンについて示されている．3：1，7：1 の通信パターンでは，特に高次元で網内の通信に偏りが生じ，特定のリンクが隘路になる．このとき到着順より巡回方式がわずかに良い稼働率を示すことが結果として現われている（表 3）．

また，K-法においてはパケットの種別に応じてバッ

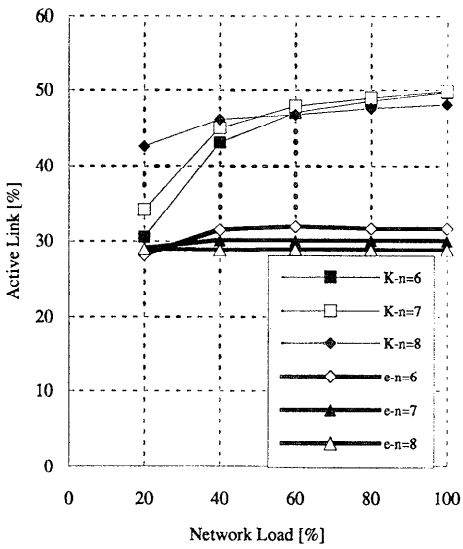


図 15 K-到着順方式のリンク稼働率 (1 : 1)
Fig. 15 Active link ratio by K-FIFO (1 : 1).

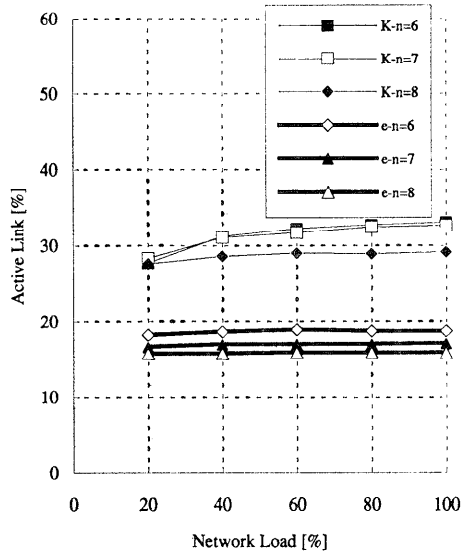


図 16 K-到着順方式のリンク稼働率 (3 : 1)
Fig. 16 Active link ratio by K-FIFO (3 : 1).

ファを別々に構成し，巡回と到着順方式を同一網内で併用することが考えられる．例えば，即時性を要求するパケットはバッファ数 $m=2$ の巡回方式で，スループットを要求するパケットは到着順方式で扱うことができる．この併用方式は K-法の特徴といえる．

K-法のより一般的な利用として，ブロードキャスト通信への適用を次に考察する．前述の可視化処理並列マシンにおいても，プログラム（数 MByte）のダウンロードや，共有データ（数 KByte）の分配にブロードキャスト通信を行った．以下に大量データと少量データに分けて，効率的な通信方法を示す．前提としてルートノードはゼロアドレス（00…000）とする．

大量データの場合は，ゼロノードを始点とする順方向ハミルトンパスに沿って，大量のパケットをパイプライン的に転送する．このとき，必要なバッファ数は $m=2$ であり，パケットサイズを各ノードの総バッファ量の半分に設定することができる．また，少量データについて，転送は有向ハイパーキューブの最下位ビット分割と逆の順序となる．すなわち，ルートノードから n 次元方向（10…000）に転送し，次に 2 ノード（00…000 と 10…000）から $n-1$ 次元方向に，そして最後に 2^{n-1} ノードから 1 次元方向に転送する．このパスはすべて順方向であり，必要なバッファ数は $m=1$ となる．

K-法はバッファ構成に自由度をもつが，その経路選択は始点と終点から静的に決まる oblivious[22] な方式に分類される．hot spot を回避する負荷分散や耐故障性への展開には，adaptive な経路選択への拡張が必要となる．K-法でのアプローチはバッファ構成の工夫にあると思われる．例えば，各ノードに 2 種類のバッファ（正のバッファと負のバッファ）を設け，正のバッファは有向ハイパーキューブのリンク方向を対応させ，負のバッファにはすべてのリンク方向を逆にしたものを対応させる．負のバッファを使う経路は，有向ハミルトンパスが逆転した順序，あるいは Gray Code の順序を反転したものから定義される．この結果，2 つの対称性をもつ経路を設定することができる．さらに，各ノードの正負のバッファ間でパケットを移行，または swap する操作を定義すると，経路を順方向に自由に選択することが可能となり，かつデッドロックフリーが保証される．これらの具体的な展開と評価については今後の課題と考える．

5. おわりに

ハイパーキューブという既存の接続網に対し，デッドロックフリーな新ルーティング方式（K-法）を提案した．K-法は距離 n の経路に不均衡性を有するが，ノードを単位としたデッドロックが防止できる特性を利用し，バッファ数の選択が自由であること，e-cube 法と同様な巡回方式以外に，1 キュのバッファ構成による到着順方式が可能であることを示した．そして到

着順方式により，優れた通信スループットが達成できることを示した．ランダムトラフィックのパケット通信は，可視化処理以外に他のマルチメディア処理やデータベース処理の並列モデルに現れる．K-法はこれら入出力データの通信網に適したルーティング方式と思われる．

K-法の今後の展開として，前述の adaptive な経路選択への拡張以外に，遅延時間の短縮を図る Virtual Cut-Through 法[23]の適用がある．また，半順方向パスを用いた経路では，次元数が上がるとともに双方向チャネル内でのパケット間衝突が減少するという特性を利用し，隣接ノード間に半2重通信の適用が考えられる．これらも今後の重要な課題と思われる．

謝辞 本研究開発の大部分は，通産省工業技術院大型プロジェクト「科学技術用高速計算システムの研究開発」の一環として，沖電気工業(株)が新エネルギー・産業技術総合開発機構（NEDO）から委託を受けて実施したものです．本研究に貴重な御意見をいただいたプロジェクト関係者に深く感謝します．

参 考 文 献

1) Hockey, W. R. and Jesshope, R. C.: *Parallel Computers 2*, Adam Hilger, Bristol and Philadelphia (1988).
2) 富田ほか：並列処理マシン，p. 236，オーム社，東京 (1989).
3) 奥川：並列計算機アーキテクチャ，p. 175，コロナ社，東京 (1991).
4) Lipovski, J. G. and Malek, M.: *Parallel Computing Theory and Comparisons*, p. 381, John Wiley & Sons, New York (1987).
5) Stone, S. H.: *High-Performance Computer Architecture*, p. 411, Addison-Wesley, Massachusetts (1987).
6) DeFanti, A. T., Brown, D. M. and McCormick, H. B.: Visualization, *IEEE Comput.*, pp. 12-25 (Aug. 1989).
7) 大宅ほか：科学技術計算の可視化処理における並列画像生成方式とその評価，情報処理学会論文誌，Vol. 33, No. 7, pp. 906-919 (1992).
8) 小山ほか：高速ハイパーキューブ網と制御方式，並列シンポジウム JSPP '89 (1989).
9) 和宇慶ほか：ハイパーキューブネットワーク制御とその評価，信学会，CPSY 91-5, pp. 9-16 (1991).
10) Seitz, L. C.: The Cosmic Cube, *CACM*, Vol. 28, No. 1, pp. 22-33 (1985).
11) 阿部：ハイパーキューブ・マルチプロセッサ，*bit*, Vol. 21, No. 5, pp. 640-651 (1989).
12) Merlin, M. P. et al.: Deadlock Avoidance in Store-and-Forward Network-1: Store-and-Forward Deadlock, *IEEE Trans. Commun.*, Vol. COM-28, No. 3, pp. 345-354 (1980).
13) 堀江ほか：並列計算機 CAP-2 のルーティング・コントローラ，情報処理学会計算機アーキテクチャ研究会，Vol. 83, No. 38 (1990).
14) Dally, J. W. and Seitz, L. C.: Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, *IEEE Trans. Comput.*, Vol. C-36, No. 5, pp. 547-553 (1987).
15) Dally, J. W. and Seitz, L. C.: The Torus Routing Chip, *Distributed Computing*, pp. 187-196, Springer-Verlag (1986).
16) Dally, J. W.: *A VLSI Architecture for Concurrent Data Structures*, p. 243, Kluwer Academic Publishers, Boston (1987).
17) Dally, J. W.: Virtual-Channel Flow Control, *Proc. of the 17th Int'l Symp. on Computer Architecture*, pp. 60-68 (May 1990).
18) 前田ほか：グラフ理論の基礎，p. 214，オーム社，東京 (1978).
19) Johnsson, S. L. and Ho, C.: Optimum Broadcasting and Personalized Communication in Hypercubes, *IEEE Trans. Comput.*, Vol. 38, No. 9, pp. 1249-1268 (1989).
20) 武：超立方体形ネットワークに於ける全点対全点通信の最適ルーティング法，第 35 回情報処理学会全国大会論文集，pp. 151-152 (1987).
21) 堀江ほか：トーラスネットワークにおける最適全対全通信方式，並列処理シンポジウム JSPP '92, pp. 187-194 (1992).
22) Talia, D.: Message-Routing Systems for Transputer-Based Multicomputers, IEEE MICRO (Jun. 1993).
23) Kermani, P. and Kleinrock, L.: Virtual Cut-Through: A New Computer Communication Switching Technique, *Computer Networks 3*, pp. 267-286, North-Holland Publishing Company (1979).

(平成6年 4 月 27 日受付)
(平成6年12月 5 日採録)

大宅伊久雄（正会員）

昭和 22 年生．昭和 44 年京都大学理学部数学科卒業．同年沖電気工業(株)入社．CAD/CAM，グラフィックス，分散処理，並列処理の研究開発に従事．現在，マルチメディア研究所に勤務．ACM 会員．

小山　法孝（正会員）

昭和36年生．昭和58年千葉大学理学部数学科卒業．昭和63年筑波大学大学院博士課程数学研究科単位取得退学．同年沖電気工業(株)入社．グラフィックス，並列処理，組込み用マイクロプロセッサのソフトウェア開発環境，の研究開発に従事．現在，マルチメディア研究所に勤務．

和宇慶　康（正会員）

昭和32年生．昭和55年琉球大学理工学部電気工学科卒業．昭和58年名古屋工業大学大学院電子工学修士課程修了．同年沖電気工業(株)入社．3次元グラフィックス，並列処理，マルチメディア処理の研究開発に従事．現在，マルチメディア研究所に勤務．電子情報通信学会会員．