社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
CPSY94-49 (1994-07)

分散共有メモリ型超並列計算機におけるディレクトリ方式と相互結合網について

秋山知之 小池帆平 田中英彦

東京大学工学系研究科
〒113 東京都文京区本郷 7-3-1

あらまし

分散メモリ型大規模並列計算機で共有メモリを実現する場合、キャッシュコンシステンシの管理にはディレクトリベースのプロトコルを用いる必要がある。このプロトコルは1対1通信によるコンシステンシの管理を基本とするが、ノード数の増加と共にパケット数、ディレクトリサイズの増加が問題となる。1対1通信の代わりにマルチキャストを用いることによってこの問題は解決することができる。

本論文では低コストでマルチキャストを実現する、多重トーラス構造を持つ相互結合網をまとめてn-RTN(n-Recursive Torus Network)と定義し、n-RTNの一つとして実際の使用形態を考慮した相互結合網RSOT(Recursive Orthogonal Torus)を提案する。このネットワークはHypercubeより小さな次数で小さな平均距離を実現する。また、メッシュを内蔵しており、メッシュアルゴリズムの適用を容易にする。

和文キーワード

超並列処理、キャッシュディレクトリ、相互結合網、マルチキャスト

Cache Directory and Interconnection Network for Large Scale Distributed Shared-Memory Multiprocessors

Tomoyuki Akiyama, Hanpei Koike, Hidehiko Tanaka

Faculty of Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku Tokyo 113, Japan

Abstract

In large scale distributed shared-memory computers, directory-based cache coherence protocol is generally used. The protocol does not rely on broadcast but 1-to-1 communication. When the number of processing elements grows to the level of 10,000, it is possible that one data set is shared by so many processing elements that using 1-to-1 communication is no longer a proper way because of the increase of packets and the amount of directory. A solution of the problem is multicasting packets to all nodes that are considered possible to share a data. The amount of the directory is minimized because the directory should have only a pointer to the area which contains the data sharing nodes.

In this paper, a category of interconnection networks, n-Recursive Torus Network(n-RTN) is first defined, which supports low-cost treeform multicasting for keeping cache coherence. And then, we propose a network, called RSOT(Recursive Orthogonal Torus), which is one of the networks classified into the category and the most suitable one for multicasting use.

The mean internode distance of RSOT is smaller than that of Hypercube , while the degree is also smaller. RSOT contains mesh connection so that mesh algorithms can be applied without any modification.

英文 key words

massively parallel processing, cache directory, interconnection network, multicast

1 はじめに

大規模並列計算機では、データの分散が進み、結果としてネットワークを介した遠隔メモリアクセスが増加することになる。遠隔メモリアクセスはローカルメモリアクセスと比較してレイテンシが大きく、処理のボトルネックとなる。また、遠隔メモリアクセスのレイテンシ隠蔽の目的で、遠隔メモリデータのキャッシングを行なうと、コンシステンシの管理を行なう必要がでてくるため、ネットワークに大きな負荷がかかる。ネットワークを介してコンシステンシを管理する場合、キャッシュはディレクトリ方式をとる必要があるが、大規模並列計算機ではコピーを持つノードをフルマップで管理することは不可能であるため、不完全な形のディレクトリを保持せざるを得ない。不完全な形のディレクトリ情報から、コピー保持の可能性のあるノードすべてに対してコンシステンシ管理パケットを送ることが必要になってくる。本論文では、ネットワークでパケットの分割、結合を行なう木構造のマルチキャストをサポートすることにより、コンシステンシ管理を低コストで行なう方式について検討する。

2 ディレクトリ方式

大規模並列計算機ではコピーを持つノードをフルマップで管理することはサイズ的に不可能であり、何らかの方法で圧縮した形でディレクトリ情報を持つ必要がある。

$\mathbf{Dir_iNB}$ 特定のプログラムではデータの共有が必要なノード数は限られるため [1][2]、データを共有するノードに対するポインタの保持をi個までに限定し、それ以上の共有が生じた時は、どれか1つのポインタをinvalidateする [3]。Dir_iNBは、データ共有ノード数が大きい場合、各ノードがその他のノードをinvalidateし合い、スラッシング状態になるため、性能が大きく低下する。

$\mathbf{Dir_iB}$ Dir_iNBと同様に、ポインタをi個に限定するが、それ以上の共有が生じた場合は、全ノードに対してブロードキャストを行なう方法 [3]。Dir_iBは、ブロードキャストによるトラヒックの増加が問題となる。

distributed 1つのブロックに対するディレクトリを各ノードに分散させて持つ。すなわち、各ノードはデータを共有するノードに対するポインタを保持し、データを共有するノード全体でリストが構成される [4]。ディレクトリ参照のコストが非常に大きいのが問題である。

階層化マップ [7] 図1のような階層化された構造を持つネットワークを前提とする。送信ノード(図のA)に対して通信距離の小さいノードに関しては、共有状態を細かく管理し、通信距離が大きくなるにつれて共有状態は粗く、つまり部分木を1単位として管理する。この方法は階層化マップの中でも近傍詳細型という方式で、その他に遠隔詳細型、同一マップ型がある [9]。

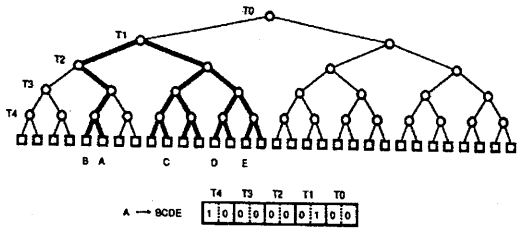

図 1: 階層化マップ

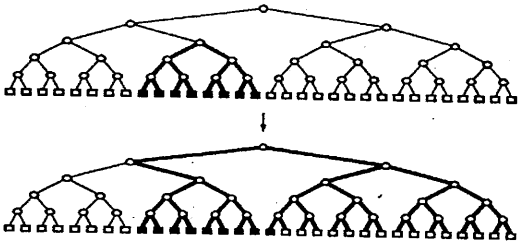

図 2: マルチキャスト範囲の拡大(階層化マップ)

管理の単位となる部分木は通信距離と共に急激に大きくなり、コピーを持たないノードへの送出パケットが多くなるため(図2)、協調動作するスレッドを近傍プロセッサにスケジューリングすることが必須である。

Coarse Vector コピーブロックを持つノードが一定領域以内に収まっている場合は、その情報をフルマップとして持ち、一定領域を越えた時は、数ノードをまとめてマップの1ビットに対応させ、マップサイズを小さく抑える [11]。コピーブロックを持たないノードに対して、コンシステンシ管理のパケットが送られるため、通信量の増加が問題となる。しかし、本論文で提案する相互結合網(n-RTN)は、木構造のマルチキャストをサポートしているため、コンシステンシ管理を部分木ごとに行なうことにより、パケットの増加を極力抑えることができる。

Sectored Block メモリブロックサイズを大きくし、ディレクトリサイズを小さくする方式 [10]。コピーを持つノード情報をブロック単位に持たせる一方、ブロックをサブブロックに分割し、サブブロック単位に属性を持たせることにより、ブロックサイズを大きくしたことによるinvalidateプロトコルにおけるキャッシュミスの増大を抑えることができる。また、データ転送単位をサブブロック単位とすることにより、参照可能になるまでのレイテンシを小さくし、また、ネットワーク負荷を抑える。サブブロック毎にownerを示すポインタを持つ必要がある。ブロックサイズの大きさには限界があるため、マルチキャストを用いる方式と組み合わせる必要が

ある。

3 n-Recursive Torus Network

相互結合網が木構造のマルチキャストをサポートすることは、前節に述べたとおり、超並列計算機のキャッシュコンシステンシ管理において、非常に重要である。超並列計算機の相互結合網にはその他に、次数がノード構成に関わらず一定であること、平均距離が短いことが要求される。また、メッシュアルゴリズムを効率良く実行できることが望ましい。これらの条件を満たす相互結合網 n-Recursive Torus Network(n-RTN) を定義する。

3.1 定義

多重トーラス構造を持つネットワーク n-RTN を以下のように定義する。

$mod(x, N_i)$ を $mod_i(x)$ と表現する。
x_i方向のノード数を N_iとする $(i = 1\sim n)$。

基本トーラス (level 0 トーラス)

$$(x_1, x_2, \cdots, x_n) \longleftrightarrow (mod_1(x_1+1), x_2, \cdots, x_n)$$
$$(x_1, x_2, \cdots, x_n) \longleftrightarrow (x_1, mod_2(x_2+1), \cdots, x_n)$$
$$\cdots\cdots\cdots\cdots$$
$$(x_1, x_2, \cdots, x_n) \longleftrightarrow (x_1, x_2, \cdots, mod_n(x_n+1))$$

ただし、$(x_1, x_2, \cdots, x_n) \in R_0$

$R_0 = \{(x_1, x_2, \cdots, x_n) | 0 \le x_i \le N_i - 1, i = 1\sim n\}$
$\longleftrightarrow$ は 2 ノード間にリンクが張られることを表す。

上位トーラス (level j トーラス)

$$(x_1, x_2, \cdots, x_n) \longleftrightarrow (mod_1(x_1 + s_{i1}^{(j)}), mod_2(x_2 + s_{i2}^{(j)}), \cdots, mod_n(x_n + s_{in}^{(j)})) \quad (i = 1\sim n)$$

ただし、$(x_1, x_2, \cdots, x_n) \in R_j$

$$(x_1, x_2, \cdots, x_n) \in R_j \Rightarrow (mod_1(x_1 + s_{i1}^{(j)}), mod_2(x_2 + s_{i2}^{(j)}), \cdots, mod_n(x_n + s_{in}^{(j)})) \in R_j \quad (i = 1\sim n)$$

各ノードが各レベルのトーラスと結合されることが望ましいが、次数が増え、リンクあたりのデータ幅に制限を受けるため、各ノードはレベル 0 トーラスと少数の上位トーラスと結合して用いる。1 ノードに結合している上位トーラスの数を多重度 m と定義する。
R_jは level j トーラスと結合されるノードの集合を表す。

3.2 n-RTN におけるブロードキャスト

level j の基底 $\mathbf{s}_1^{(j)}, \mathbf{s}_2^{(j)}, \cdots, \mathbf{s}_n^{(j)}$ (図 3) の 1 次結合で level j+1 の基底は次のように表現できる。

$$\mathbf{s}_i^{(j+1)} = u_{i1}^{(j)}\mathbf{s}_1^{(j)} + u_{i2}^{(j)}\mathbf{s}_2^{(j)} + \cdots + u_{in}^{(j)}\mathbf{s}_n^{(j)}$$

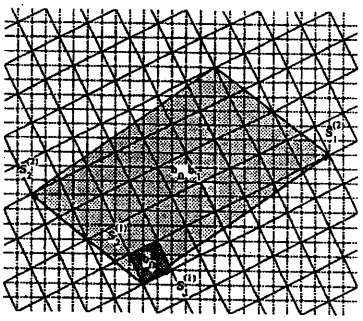

図 3: 基底ベクトルによる上位トーラスの規定

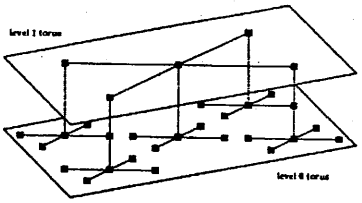

図 4: n-RTN による木構造の構成

n-RTN は図 4に示すように、各レベルのトーラスを用いて、木構造を構成し、この木構造を用いてマルチキャストを実現する。$|\mathbf{u}_1^{(j)}\mathbf{u}_2^{(j)}\cdots\mathbf{u}_n^{(j)}| = b_j \quad (j = 0\sim L-1)$ は、この木構造の level j トーラスにおける分岐数を表す。L はネットワークが持つトーラスの最大レベルとする。

その他に以下のようなパラメタを定義する。
b_j : level j の分岐数
t_j : level j の分岐ホップ数 (分岐に必要なホップ数)
c_j : level j の分岐係数 (1 ホップあたりの分岐数)
h : level 間移動に要するホップ数

c_jは以下のように表せる。

$$c_j = \sqrt[t_j+h]{b_j}$$

n-RTN を用いて再帰的ブロードキャストが可能である条件は以下のように表現できる。ただし、再帰的ブロードキャストが可能とは、図 4の木構造を用いて、全ノードにパケットを送ることが可能であり、また、各ノードに重複してパケットが届くことがないということである。つまり、木構造の葉の部分が 1 対 1 にノードに対応し、葉全体の集合とノード全体の集合が一致するということである。

再帰的ブロードキャストが可能な条件

$|\mathbf{u}_1\mathbf{u}_2\cdots\mathbf{u}_n| = b_L$を満たすある $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n$ に対して $\mathbf{s}_i = u_{i1}\mathbf{s}_1^{(L)} + u_{i2}\mathbf{s}_2^{(L)} + \cdots + u_{in}\mathbf{s}_n^{(L)} \quad (i = 1\sim n)$ のそれぞれが level 0 トーラスの基底と同じ方向を持つ
$\Leftrightarrow$ 再帰的ブロードキャスト可能

$\mathbf{e}_i(x_i$方向単位ベクトル) と同じ方向を持つものが $\mathbf{s}_{i'}$ のとき、$N_i = |\mathbf{s}_{i'}|$ となる。

(説明) level 0 トーラスにおいて分岐数 b_0の分岐パターンが存在し、ある分岐中心ノードからその分岐パターンに従って分岐した結果生じる領域を A_0 とすると、A_0を level 1 トーラスの各基底ベクトルの間隔で、重複部分がなく埋め込むことが可能である。したがって、level 1 トーラスにおいて分岐数 b_1のある分岐パターンにより、領域 A_0を分岐させると、ノード数 b_0b_1の領域 A_1を作ることができ、A_1を level 2 トーラスの各基底ベクトルの間隔で重複部分がなく、埋め込むことも可能となる。したがって、各 level で分岐を繰り返し、level L トーラスでの分岐の結果生じた領域 A_Lは level L+1 トーラスの基底ベクトルの間隔で、重複のない埋め込みが可能であることが、再帰的に示される。level L+1 トーラスの基底ベクトルが基本トーラスの基底ベクトルの方向を持っていれば、level L+1 トーラスの基底ベクトルの長さで折り返してリングを作ることにより、領域 A_Lをそのまま埋め込みが可能なトーラスが形成される。

$$S_j = (\mathbf{s}_1^{(j)}, \mathbf{s}_2^{(j)}, \cdots, \mathbf{s}_n^{(j)})$$

$$U_j = (\mathbf{u}_1^{(j)}, \mathbf{u}_2^{(j)}, \cdots, \mathbf{u}_n^{(j)})$$

とおくと、

$$S_{j+1} = S_jU_j = S_0U_0U_1\cdots U_j = U_0U_1\cdots U_j$$

となり、再帰的ブロードキャストが可能な条件は次のように書き換えることができる。

$U_0U_1\cdots U_L$の列ベクトルを交換すると対角行列になる

⇔ 再帰的ブロードキャスト可能

したがって、level 1 トーラスの基底を適当に選んだ時、その基底によって決まる U_0の逆行列 $U = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n)$ が求まれば、

$$U_2 = U_4 = \cdots = U_0 \quad (1)$$

$$U_1 = U_3 = \cdots = (\alpha_1\mathbf{u}_1, \alpha_2\mathbf{u}_2, \cdots, \alpha_n\mathbf{u}_n) \quad (2)$$

と置くと、

$$U_0U_1\cdots U_{2i-1} = \begin{pmatrix} \alpha_1^i & & & 0 \\ & \alpha_2^i & & \\ & & \ddots & \\ 0 & & & \alpha_n^i \end{pmatrix} (i = 1, 2, \cdots)$$

となり、$L = 2i - 1$ に対して、再帰的ブロードキャストが可能となる。

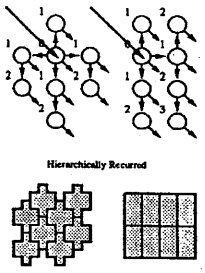

図 5: マルチキャストパターンの比較 (左:RDT)

(1) では $U_2, U_4, \cdots$をすべて U_0としたが、これらは同じである必要はない。ただし、このとき (2) でそれぞれに対する逆行列を用いる必要がある。また、(2) において、列ベクトルを交換しても差し支えない。

対角化が可能な行列 A を適当に選び、A の固有ベクトルを $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$とすると、$P = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n)$ に関する基底の変換 $(P^{-1}AP)$ により A は対角行列になる。

したがって、

$$U_0 = P^{-1}$$

$$U_1, U_2, \cdots, U_{L-1} = A$$

$$U_L = P$$

と定めることにより、

$$U_0U_1\cdots U_L = PA^{L-1}P^{-1} =$$

$$PAP^{-1}PAP^{-1}\cdots PAP^{-1} = \text{対角行列}$$

となり、再帰的ブロードキャストが可能となる。

4 n-RTN の具体例

前節では、木構造のマルチキャストをサポートする超並列処理用の相互結合網 n-RTN を、一般化した形で定義したが、この節ではその範疇で最適なネットワークを検討する。

4.1 Recursive Diagonal Torus

Recursive Diagonal Torus(RDT) [5] [6] は、n-RTN の中で、$\mathbf{u}_1^{(j)} = (k, k), \mathbf{u}_2^{(j)} = (-k, k)$ $(j = 0 \sim L-1)$

を満たすものと考えることができる。kを基数と呼び、通常 $k = 2$ のものが用いられる。$b_j = 2k^2 = 8$ である。基数 k、最大レベル L、多重度 m の RDT を RDT(k,L,m) と表現する。

RDT におけるマルチキャストパターンは図 5の左上のようになる。左下は、2 階層分のマルチキャストを行なったときの領域を示す。

4.2 2-RTN5

基底は、$\mathbf{u}_1^{(j)} = (1, 2), \mathbf{u}_2^{(j)} = (-2, 1)$ $(j = 0, 2, \cdots)$

$\mathbf{u}_1^{(j)} = (1,-2), \mathbf{u}_2^{(j)} = (2,1)$ $(j = 1,3,\cdots)$
によって決定する。

$$b_j = \begin{vmatrix} 1 & -2 \\ 2 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 2 \\ -2 & 1 \end{vmatrix} = 5$$

$t_j = 1$ であり、$h = 1$ のとき $c_j = 2.24$(RDT では $c_j = 2$)となる。

その他に、

$\mathbf{u}_1^{(j)} = (2,3), \mathbf{u}_2^{(j)} = (-3,2)$ $(j = 0,2,\cdots)$
$\mathbf{u}_1^{(j)} = (2,-3), \mathbf{u}_2^{(j)} = (3,2)$ $(j = 1,3,\cdots)$
$b_j = 13, t_j = 2, c_j = 2.35$
で定義される構成 (2-RTN13) や、

$$U_j = \begin{pmatrix} 1 & 2 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & 0 \end{pmatrix} (j = 0,2,\cdots)$$

$$U_j = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 2 & -2 \\ -1 & 1 & 0 \end{pmatrix} (j = 1,3,\cdots)$$

$b_j = 24$ で定義される構成 (3-RTN24) なども存在する。

4.3 マルチキャストパターンに対する制約

RDT を含めて n-RTN は基本トーラスを内蔵しており、メッシュアルゴリズムを効率良く実行できる。メッシュアルゴリズムを使うために矩形あるいは直方体の形にパーティショニングを行なった際、収束判定などの目的でパーティション内のグローバル共有変数を用いる場合、RDT のようなマルチキャストパターンを持ったネットワークでは、コンシステンシ管理パケットがパーティション外に発行されることは避けられない。これは通信量の増加を招き、処理速度の低下につながる。

また、プロセッサ資源の管理に buddy system を使う場合を考えると、RDT のようなマルチキャストパターンに合わせて、パーティショニングを行なうことは、OS の負荷を大きくする原因となる。

このようなことから、マルチキャストパターンは矩形、直方体であることが要求される。しかし、図 5を見てわかるとおり、マルチキャストパターンを矩形、直方体にすることは、一般にマルチキャストのレイテンシを増加させることになる(図で、左右共に $b = 8$ であるが、左の RDT では $t = 2$ であるのに対して右は $t = 3$)。ディレクトリ方式のキャッシュでは、メモリトランザクションの単一性と順次性の保証のために、トランザクションはホームの存在するノードにまず送られ、そこで調停がなされる。順次性の保証のためにはメモリトランザクションやキャッシュの操作ごとに Acknowledge (Ack) を処理の要求元に返す必要がある。n-RTN で木構造のマルチキャストを用いてこのような操作を行なう場合、送信元では 1 つのパケットを、木構造の下の階層に行くにしたがって分割するという方式を用いることにより、すべて 1 対 1 通信で行なうよりも、パケット数を少なくすることが可能となり、処理性能の向上に大きな効果がある [8]。Ack を回収する

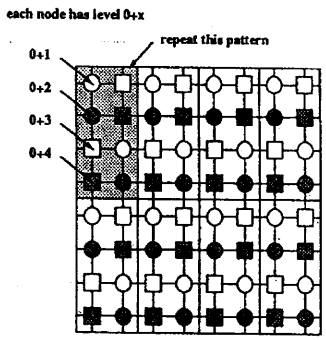

図 6: RSOT の構成 (レベル 0 トーラス)

時には、逆に分割が行なわれたノードで結合が行なわれるが、そのためには、結合の行なわれるノードのルータに、現時点で回収された Ack の数を格納するテーブルを保持する必要がある。マルチキャストのレイテンシが増加すると、この情報の保持時間が長くなるため、テーブルサイズを大きくしなければならず、ルータのハードウェア量の増大を招く。

4.4 Recursive Orthogonal Torus

4.3 節では、マルチキャストパターンを矩形にすることの重要性、また、一般にそうした場合、マルチキャストのレイテンシが増加し、Ack テーブルのサイズが増加することを述べた。ここで提案する相互結合網 Recursive Orthogonal Torus(RSOT) は、RDT とほぼ同等の平均距離、スループットを持ち、マルチキャストのレイテンシも RDT と同じ程度に抑え、なおかつ、マルチキャストパターンが矩形であるという特徴を持つ。

この相互結合網も RDT と同様に、n-RTN の 1 つとして定義することができる。基底は $\mathbf{u}_1^{(j)} = (0,h), \mathbf{u}_2^{(j)} = (-k,0)$ $(j = 0 \sim L-1)$ によって決定する。以下では $h = 4, k = 2$ に限定して考える。RSOT も RDT と同様に、h、kおよび最大レベル L、多重度 m を用いて RSOT(h,k,L,m) と表現することにする。

$b_j = hk = 8$ である。

$$U_j = \begin{pmatrix} 0 & -2 \\ 4 & 0 \end{pmatrix} (j = 0 \sim L-1)$$

$S_j = U_0 U_1 \cdots U_{j-1}$ より、

$$S_1 = \begin{pmatrix} 0 & -2 \\ 4 & 0 \end{pmatrix}, S_2 = \begin{pmatrix} -8 & 0 \\ 0 & -8 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 0 & 16 \\ -32 & 0 \end{pmatrix}, S_4 = \begin{pmatrix} 64 & 0 \\ 0 & 64 \end{pmatrix}$$

が求まる。5 レベル構成の RSOT を図 6、7、8に示す。多重度 m は 1 であり、次数 (ノードあたりリンク数) は 8 である。ノード構成が変わるとレベルの構成は変わるが、次数は変化させる必要はなく、リンクの結合パターンの変更のみで対応できるため、スケーラブルである。これは n-RTN すべてに当てはまる。

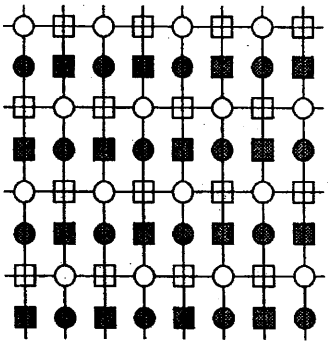

図 7: RSOT の構成 (レベル 1 トーラス)

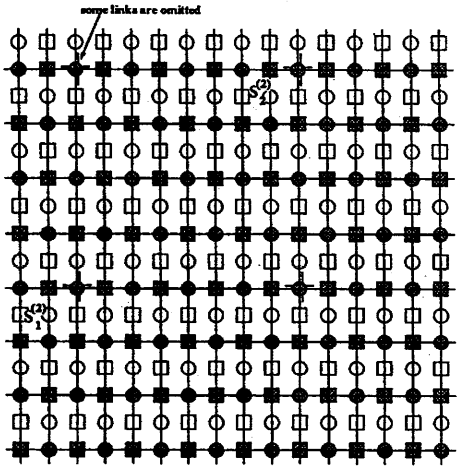

図 8: RSOT の構成 (レベル 2 トーラス)

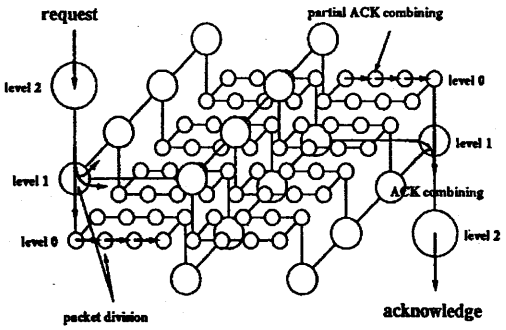

図 9: RSOT におけるマルチキャスト

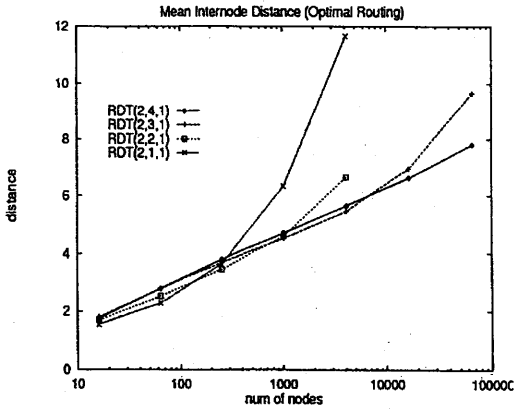

図 10: RDT(2,L,1) の平均距離 (L=1～4)

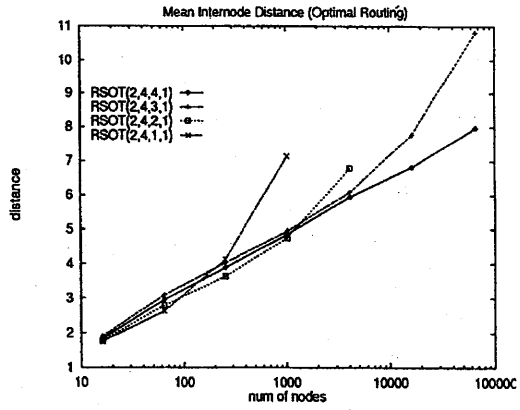

図 11: RSOT(2,4,L,1) の平均距離 (L=1～4)

RSOT におけるマルチキャストを図 9に示す。メモリトランザクションは図のように分割を繰り返しながら各ノードにマルチキャストされる。RSOT はその ACK の回収方式に特徴がある。RSOT のようにマルチキャストパターンを矩形にした場合、そのレイテンシが問題となるが、RSOT では ACK パケットを要求パケットとは異なる木構造で回収する方法をとることによってこの問題を解決する。このような方法をとると、ACK の結合ノードに各 ACK が返ってくるタイミングが理想的には等しくなり、必要とする ACK テーブルのサイズを小さくすることが可能となる。また、隣のノードから ACK が届く時間と、自分のノードから ACK が発生する時間も理想的には等しいため、ルータのバッファに結合可能な ACK パケットが存在すればバッファ中で結合してしまうという方法 (partial ACK combining) をとることにより、ネットワーク負荷を最小限に抑えることが可能である。

4.5 性能評価

次に RDT と RSOT について、相互結合網の基本的特性の比較を行なう。

RDT(2,L,1)(L=1～4) の平均距離を図 10に示す。RSOT(2,4,

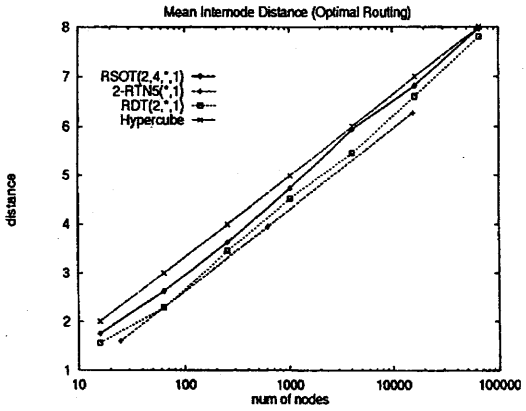

図 12: 平均距離の比較

L,1)(L=1～4) の平均距離を図 11に示す。ただし、ここでの平均距離は最適ルーチングを用いた場合の各ノード間距離の単純平均である。通常はノード構成に合わせて、平均距離を最小にする L を選択する。図 12に RDT(2,L,1)、RSOT(2,4,L,1)、2-RTN5(L,1)(ともに最適構成、次数 8)、Hypercube(65536 ノードで次数 16) の平均距離を示す。RSOT の RDT に対する平均距離の増加は 0.2～0.3 にとどまっていることがわかる。

RDT(2,2,1)、RSOT(2,4,1)、2-torus に対して、スループットの測定を行なった。ただし、ノード構成は 16 × 16 で、フロー制御方式は virtual cut through とした。また、各ネットワークの 1 ノードあたりのリンク数を、プロセッサへのリンクを含めて RDT、RSOT では 9 とし、2-torus では 5 と考え、リンクの幅をそれに合わせて 5:9 であると仮定し、パケット長を RDT、RSOT では 9、2-torus では 5 とした。またバッファサイズを RDT、RSOT では 10、2-torus では 18 とした。ランダム通信を行なった結果を図 13に示す。正規化を行なったにも関わらずランダム通信では 16 × 16 という少数ノード構成でありながら、2-torus と RDT、RSOT の間には大きなスループットの違いが見られる。ランダム通信のスループットに関しても RDT と RSOT の違いは少ないといえる。全ノードを 1/4 のパーティションに分割し、パーティション内でランダム通信を行ないスループットを測定した (図 14)。2-torus が RDT、RSOT に比較して高いスループットを示しているのは、1/4 パーティションに分割することにより、RDT、RSOT で、高いレベルのトーラスの使用が不可能となる状況が生じたためである。この場合においても、RDT と RSOT の差はあまりない。パーティション内のランダム通信では、マルチキャストパターンによる違いは当然見られないが、パーティション内のマルチキャストでは RSOT の優位性ははっきりと現れるであろう。

図 15は RDT(2,4,1) と RSOT(2,4,4,1) がブロードキャストにかかるホップ数を示したものである。RSOT では、図 9に示した方式を用いることにより RDT と同等のホップ数でブロードキャストを可能にしている。

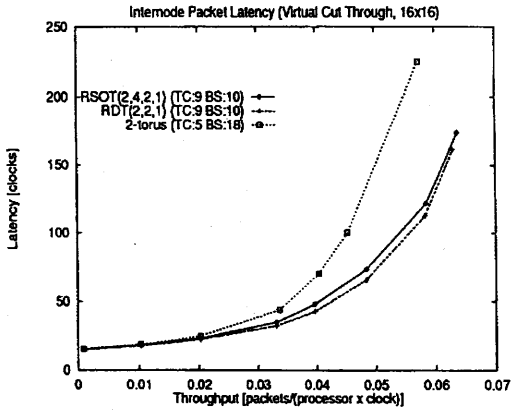

図 13: 通信特性 (ランダム通信)

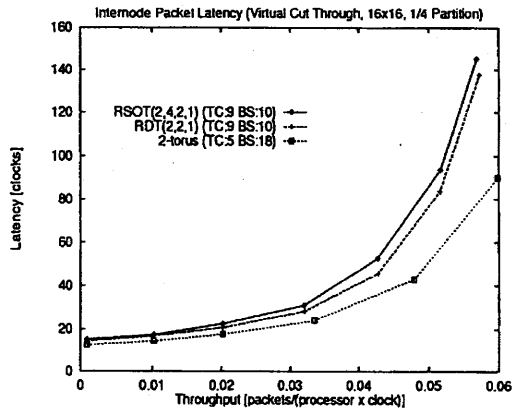

図 14: 通信特性 (1/4 パーティション通信)

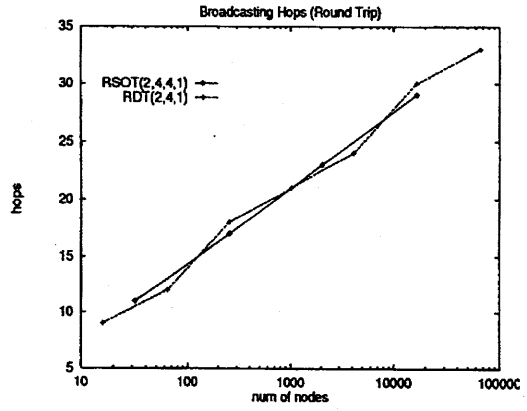

図 15: ブロードキャストホップ数 (Ack の回収を含む)

5 おわりに

超並列計算機で効率的なディレクトリベースのキャッシュを実現するためにはネットワークでパケットの分割、結合を行なう木構造のマルチキャストをサポートすることが必要であると考え、それを多重トーラス構造のネットワーク (n-RTN) で実現する方法を示した。また、n-RTN の一つとして実際の使用形態を考慮したネットワーク RSOT を提案し、その基本性能の評価をした。

謝辞

本研究の一部は、文部省科学研究費 (重点領域研究 (1) 課題番号 04234130 「超並列原理に基づく情報処理基本体系」) による。適切な助言を頂いた東京大学理学部の松本先生、およびその他の関係者に感謝します。

参考文献

[1] W.Weber, A.Gupta. ASPLOS-III proc. 243-256

[2] S.J.Eggers, R.H.Katz. 15th ISCA proc. 373-382

[3] A.Agarwal, R.Simoni, J.Hennesy, M.Horowitz. 15th ISCA proc. 280-289

[4] D.B.Gustavson. IEEE Micro. Feb.1992 10-22

[5] 楊愚魯, 天野英晴, 柴村英智, 末吉敏則. 信学技報 CPSY93-26. 105-112

[6] 楊愚魯, 天野英晴. 情処研報 92-ARC-96. 141-147

[7] 松本尚, 平木敬. JSPP'93 proc. 245-252

[8] 佐藤充, 三吉貴史, 松本尚, 平木敬, 田中英彦. シミュレーションを用いた疑似フルマップの定量的評価. SWoPP'94 ARC.

[9] 工藤知宏, 松本尚, 平木敬, 西村克信, 楊愚魯, 天野英晴. 超並列計算機でのコヒーレンシ維持のためのマルチキャスト手法の検討. SWoPP'94 ARC.

[10] B.W.O'Krafka, A.R.Newton. 17th ISCA proc. 138-147

[11] D.Lenoski, J.Laudon, K.Gharachorloo, A.Gupta, J.Hennesy. 17th ISCA proc. 148-159