

## 2次元 Mesh・Torus ネットワーク上での 最適全対全通信アルゴリズムの評価

高上 治之<sup>†</sup> 矢崎 俊志<sup>††</sup> 安島 雄一郎<sup>†††</sup>  
清水 俊幸<sup>†††</sup> 石畑 宏明<sup>†</sup>

筆者らは Mesh・Torus ネットワーク上での全対全通信アルゴリズム A2AT を提案した。本論文では、A2AT の通信性能をフリットレベルのネットワークシミュレータを用いて評価した結果について報告する。現実的なモデルである、物理チャネルあたりのバーチャルチャネル数を 2 とした場合、予測値に対し平均約 1.09 倍の通信時間であり、既存の全対全通信アルゴリズムと比較して、約 12.3%~48.0%通信時間が低減され、ネットワークサイズが大きくなるほど優位であった。通信の開始時刻は各ノードでばらつきがある場合でも、ノード内でローカルな送受信の待ち合わせを行うことにより、各ノードでのわずかなタイミングのずれが全体の通信性能に影響を与えないことを示した。各ノードからの送信数を増やした場合は、送信数 1 のときと比べ、Mesh では平均約 18.8%、Torus では平均約 41.2%通信時間が低減された。

### Evaluation of Optimal All-to-All Communication Algorithm on 2-dimensional Mesh Network and Torus Network

HARUYUKI TAKAUE,<sup>†</sup> SYUNJI YAZAKI,<sup>††</sup> YUICHIRO AJIMA,<sup>†††</sup>  
TOSHIYUKI SHIMIZU<sup>†††</sup> and HIROAKI ISHIHATA<sup>†</sup>

In this paper, we evaluate the performance of all-to-all communication algorithm for torus and mesh network, A2AT, by using a flit level simulator. Under a practical condition that use 2 virtual channels, A2AT achieved 1.09 times of analytical prediction time and cut 12.3% to 48.0% of communication time when existing algorithm was used. We shows that the difference of start time of communication occurred in each node have little effect to communication performance. When a number of concurrent message transfer was set to more than 1, A2AT for mesh reduced by 18.8% and torus reduced by 41.2% in case of that was set to 1.

#### 1. はじめに

ネットワークへの通信負荷が高い通信パターンとして全対全通信がある。全対全通信とは、全てのノードが他の全てのノードに対して、それぞれ異なった内容のメッセージを送信する通信パターンである。行列の転置、FFT など多くのアプリケーションで頻繁に使用される。

近年の大規模並列計算機では、ノード数の増加に伴い、Mesh・Torus などのネットワークポロジが用いられる事が多くなってきた。実際に、Cray XT5<sup>1)</sup>

のように 3 次元 Torus を採用しているものや、Red Storm<sup>2)</sup> のように、 $z$  次元のみを Torus とした 3 次元 Mesh を採用しているものがある。このようなネットワークを持つシステムでは、通信をする際、メッセージの衝突が起きやすい。この影響で一部がホットスポットとなり、通信性能の低下につながる。そのため、Mesh・Torus といったバイセクションバンド幅の小さなネットワークでは、バンド幅を最大限に引き出す通信アルゴリズムの実現が重要となる。

Mesh・Torus のように、ノードが複数のリンクをもつ場合には、複数の通信コントローラを並行動作させ、空いているリンクを効率的に使用することで通信性能を向上することができる。最近の並列計算機は、1 つのノードに複数の通信コントローラを持ち、複数のリンクから同時にメッセージを送信できるものが増えてきている。Bruck ら<sup>3)</sup>、Tipparaju ら<sup>4)</sup>、Ajima ら<sup>5)</sup> は、そのようなシステムについて報告している。

<sup>†</sup> 東京工科大学

Tokyo UnTiversity of Technology

<sup>††</sup> 電気通信大学 情報基盤センター

Information Technology Center, The University of Electro-Communications

<sup>†††</sup> 富士通株式会社

FUJITSU, LIMITED.

筆者らは 2次元の Mesh および Torus ネットワークにおいて、複数のメッセージを同時送信可能な通信コントローラをもつシステム向けに、全対全通信を最適に実行するアルゴリズム A2AT<sup>6)</sup> を提案した。A2AT では、従来方式のように各ノードが通信 Phase 毎に全体での同期をとる必要はない。個々のノードは複数のメッセージを並行して送信するだけでよい。ネットワーク上での競合の影響を考慮するにあたり、ネットワーク中の全リンクについて、メッセージは公平にリンクのバンド幅を共有して流れるものと仮定した。

アービトレーションの違いや、バッファリングなどの影響は通信性能にも影響を与える。A2AT は、遠くからきたパケットほど優先度をあげて、各ノードでの送信完了時間を公平とするようなアービトレーション<sup>7)</sup> を基に考案した。現在の多くのシステムではルータ内のメッセージをラウンドロビンで選択し、各ポートで公平に出力する実装が主流である。このようなルータを縦続接続したネットワークでは、各ノードが同時にメッセージを送った場合、あとから合流したメッセージの方が優先度が高くなる。アルゴリズムの提案時には、このようなモデルの違いによる影響については評価されていなかった。また、ルータ上でのパケットのバッファリングの影響も考慮しておらず、この点においても現実のマシン上での通信性能との乖離が懸念される。

本論文では、多くのシステムで採用されているルータ内でのラウンドロビンによるアービトレーションの影響、バッファリングの影響も考慮したシミュレーションモデルによる A2AT の性能評価を行う。2 章で対象とするネットワーク、3 章で A2AT について述べる。4 章では A2AT と既存のアルゴリズムとの通信性能を比較し、5 章でまとめる。

## 2. 全対全通信

### 2.1 対象とするネットワーク

A2AT では、2次元格子状に接続された各ルータにノードを 1 つ接続する構成の 2次元 Mesh・Torus ネットワークを対象とする。ルータ間は双方向のリンクで接続されている。Mesh・Torus を構成する全リンクは同一のバンド幅を持ち、各リンクは同時に双方向の通信を行うことが可能なものとする。

各ノードは、図 1 のように、送信メッセージが格納された 1 つの送信キューに対し、複数の通信コントローラを持つ。ノードとルータは双方向の通信路で接続され、ルータ間リンクのバンド幅に対して十分に大きいバンド幅を持つものとする。各ノードは、全対全

通信に必要な  $(N - 1)$  回の送信を FIFO 順に並行動作する通信コントローラに割り当てていく。

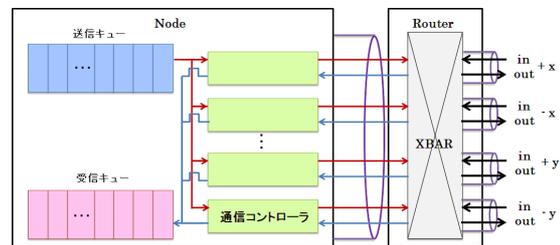


図 1 Node 内, Router 内の構成

メッセージの中継は、ワームホール方式、またはパッチャルカットスルー方式を想定している。ルーティングは Dimension-order ルーティングとする。この方式は、 $XY$  平面上にある 2次元 Mesh・Torus ネットワークにおいて、まずメッセージを送信元ノードから  $X$  軸方向に送り、次に  $Y$  軸方向に送る方式である。

### 2.2 既存の全対全通信アルゴリズム (A2AND)

直接法による全対全通信アルゴリズムとして、MPICH<sup>8)</sup> に使用されている SimpleSpread 法 (以降、A2A) がある。A2A は、図 2 のように宛先を 1 個ずつ順に変えて送信する。ここで、 $N$  は全ノード数、

```

for  $i = 1$  to  $N - 1$  do
    send( $(myid + i) \bmod N$ );
end for
    
```

図 2 A2A の送信順

$myid$  は並列処理における自分の Rank,  $send(t)$  は、ノード  $t$  へのメッセージ送信を示す。

通信ネットワークが Mesh や Torus である場合は、1次元で順に送受信位置を移動させていく A2A の他に、 $d$ 次元座標上の相対位置を順次移動させる方法もあろう。この方法を以降、A2AND と呼ぶ。 $x$  方向のサイズが  $N_x$ ,  $y$  方向のサイズが  $N_y$  の 2次元 Mesh・Torus では、送信先のノードの相対位置を 2次元座標  $(x, y)$  で示す。各ノードは  $0 \leq x \leq N_x - 1, 0 \leq y \leq N_y - 1$  の範囲に図 3 のようにメッセージを送る。

## 3. 全対全通信アルゴリズム (A2AT)

A2AT では、ノードに接続されたリンクを有効活用することを考え、1つのノードから行われる同一方向への複数の通信を重ねないように、通信をスケジューリングする。既存のアルゴリズム A2A, A2AND では、

```

for  $y = 0$  to  $N_y - 1$  do
  for  $x = 0$  to  $N_x - 1$  do
    if  $x \neq 0$  and  $y \neq 0$  then
      send( $(myidx + x) \bmod N_x, (myidy + y) \bmod N_y$ );
    end if
  end for
end for
end for

```

図 3 A2AND の送信順

同一方向への送信が連続し、その方向のリンクのバンド幅がネックとなって全体のバンド幅を有効に利用できない。空いているリンクを利用するようスケジューリングすることで通信性能の向上ができる。通信コントローラを複数用いるメリットを活かすため、全対全通信に必要な  $(N - 1)$  回の送信を FIFO 順に並行動作する通信コントローラに割り当てていく。並行動作する通信コントローラの数、以降  $NCT$  とする。

### 3.1 奇数サイズの正方形 2 次元 Mesh

本論文では、正方形 2 次元 Mesh および Torus ネットワーク上での A2AT の性能評価を行う。一辺のサイズ  $K$  が奇数サイズの正方形 2 次元 Mesh・Torus では図 4 に示すように 2 step で全対全通信を行う。Step 1 の処理を図 5 に、Step 2 の処理を図 6 に示す。send( $i, i$ ) は、send( $(myidx + i) \bmod N_x, (myidy + i) \bmod N_y$ ) を示す。

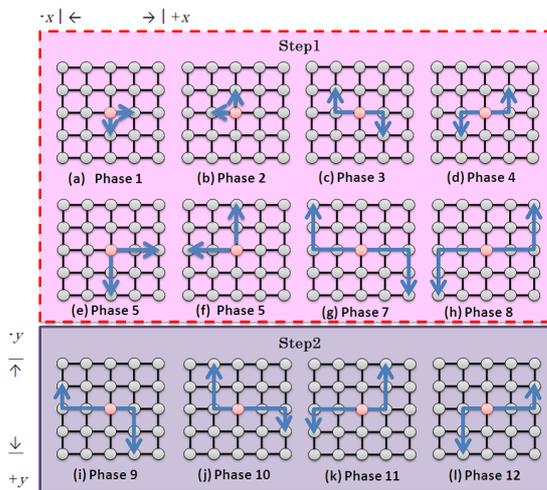


図 4 奇数サイズ正方形 2 次元 Mesh

Step 1 では、まず送信元ノードは  $x$  方向および、 $y$  方向の軸上にあるノード、および対角線上にあるノード

### Step 1

```

for  $i = 1$  to  $\frac{K-1}{2}$  do
  send( $i, 0$ ); send( $0, i$ ); {(a)(e)}
  send( $-i, 0$ ); send( $0, -i$ ); {(b)(f)}
  send( $i, i$ ); send( $-i, -i$ ); {(c)(g)}
  send( $i, -i$ ); send( $-i, i$ ); {(d)(h)}
end for

```

図 5 A2AT Step 1 送信順

に対して送信を行う。図 4(a)~(h) のように、自ノードの隣接ノードを処理した後、順に遠いノードへの処理を行う。軸上にあるノードの処理は、図 4(a), (b), (e), (f) のように、 $x$  軸上にあるノードへのメッセージの送信と  $y$  軸上にあるノードへのメッセージの送信を組み合わせるで行う。組み合わせる送信するメッセージの通信を以降、Phase と呼ぶ。全ノードが同時にこの操作を行う事により、各 Phase ですべてのリンクを使用し、送信を行うことができる。

対角線上にあるノードに対しても、図 4(c), (d), (g), (h) のように送信を行う。

### Step 2

```

for  $i = 1$  to  $\frac{K-1}{2}$  do
  for  $j = 1$  to  $i$  do
    send( $i, j$ ); send( $-j, -i$ ); {(i)}
    send( $j, i$ ); send( $-i, -j$ ); {(j)}
    send( $i, -j$ ); send( $-j, i$ ); {(k)}
    send( $j, -i$ ); send( $-i, j$ ); {(l)}
  end for
end for

```

図 6 A2AT Step 2 送信順

Step 2 では、Step 1 で送信が完了した以外の位置にあるノードに対して、図 4(i)~(l) のように、 $x$  方向、 $y$  方向の各リンクの負荷バランスが均一になるよう、ともに最大でホップ数分のメッセージが流れるように組み合わせる送信を行う。

### 3.2 偶数サイズの正方形 2 次元 Mesh

一辺のサイズ  $K$  が偶数である 2 次元 Mesh では、このネットワーク内にある最大の奇数サイズのネットワークについて、Step 1, Step 2 と順に処理し、その後、図 7 に示す、Step 3 で余った各行の処理を順に行う。

奇数サイズの時と同様の考えで、図 7(m),(n) のよ

Step 3

```

for  $i = 1$  to  $\frac{K}{2} - 1$  do
     $send(\frac{K}{2}, i); send(-i, \frac{K}{2}); \{(m)\}$ 
     $send(\frac{K}{2}, -i); send(i, \frac{K}{2}); \{(n)\}$ 
end for
 $send(\frac{K}{2}, 0); send(0, \frac{K}{2}); \{(o)\}$ 
 $send(\frac{K}{2}, \frac{K}{2}); \{(p)\}$ 
    
```

図 7 A2AT Step 3 送信順

うに、 $x$  方向、 $y$  方向の負荷バランスを均一とし、各リンクに最大でホップ数個のメッセージが流れるように組み合わせて送信を行う。次に、図 7(o) に示すように、水平垂直軸上にあるノードの処理を行う。最後に、対角線にあるノードの処理を行う。対角線にあるノードの処理は、 $NCT$  が 1 となるが、 $x$  方向、 $y$  方向のすべてのリンクを使用し、送信を行うことができる。Step 3 においても、常にすべてのリンクを使用して送信を行っている。

3.3 2次元 Torus への拡張

2次元 Torus ネットワークについても 2次元 Mesh ネットワークと同様に考えることができる。奇数サイズの正方形 2次元 Mesh ネットワークのアルゴリズムを、 $NCT$  を 4 とし、 $+x$  方向、 $+y$  方向、 $-x$  方向、 $-y$  方向の 4 方向を組み合わせて同時に送信を行う。これにより、常に 4 方向のリンクを使用し、各リンクに流れるメッセージ数が均一となる。

偶数サイズでは、ちょうど中間距離にあるノードへの送信には、2 通りの経路がある。このときは送信先に応じて、逆方向のリンクを使用し、 $+x$  方向、 $+y$  方向、 $-x$  方向、 $-y$  方向の各リンクに流れるメッセージ数がホップ数個となるように送信を行う。

3.4 A2AT による全対全通信時間

A2AT による全対全通信時間を通信のバンド幅のみに着目して、解析的に求める。A2AT は、ネットワーク中の全リンクについて、メッセージは公平にリンクのバンド幅を共有して流れると仮定した。一辺のサイズ  $N$  が奇数である 2次元 Mesh において、各ノードは自分から見て、 $x$  方向に  $i$ 、 $y$  方向に  $j$  の位置にあるノードを相対位置  $(i, j)$  で表す。すべてのノードが自分の位置から  $(i, j)$  の位置にあるノードへメッセージを送信したとき、 $x$  方向の各リンクに流れる最大のメッセージ数は  $i$  個となり、 $y$  方向の各リンクに流れる最大のメッセージ数は  $j$  個となる。各リンクのバンド幅は、各リンクに同時に流れるメッセージ数により公平に等分される。各ノードが享受できるバンド幅は

経路の最少のバンド幅で律速されるので、各ノードは、 $1/\max(i, j)$  のバンド幅でメッセージを送信することが出来る。バンド幅のみに着目し、レイテンシを無視したときの各 Phase での通信時間は  $\max(i, j)$  となる。このようにして、各 Phase での通信時間の総和を取ることで、解析的に全対全通信時間を算出することができる。 $NCT$  を 1 としたときの予測値は、 $TV_{NCT1} (= N(N+1)(N-1)/3)$  となる。

$NCT$  を 2 としたとき、Mesh での予測値は、理論的な下限の通信時間 (LowerBound)  $LB_{Mesh}$  と一致する<sup>6)</sup>。Torus では、各 Phase で各ノードが享受できるバンド幅は  $NCT$  を 1 とした場合と同等である。全通信にかかる Phase 数は  $NCT$  を 1 としたときの  $1/2$  となるため、予測値  $TV_{NCT2}$  は  $TV_{NCT1}$  の  $1/2$  となる。 $NCT$  を 4 としたとき、Torus では理論的な下限の通信時間  $LB_{Torus}$  となる。A2AT の予測値による通信時間を、表 1 にまとめる。

表 1 A2AT の解析的に求めた予測値

	Mesh	Torus
$NCT = 1$	$TV_{NCT1} = N(N+1)(N-1)/3$	
$NCT = 2$	$LB_{Mesh} = N(N+1)(N-1)/4$	$TV_{NCT2} = N(N+1)(N-1)/6$
$NCT = 4$	—	$LB_{Torus} = N(N+1)(N-1)/8$

Mesh では  $NCT$  を 2、Torus では  $NCT$  を 4 とした場合に、常に全方向のリンクを使用するように通信をスケジューリングしている。 $NCT$  が少なく、全方向のリンクを使用するに満たない状況でも、既存のアルゴリズムと異なり同一方向への通信が連続しないため、リンクを効率的に使用している。

4. アルゴリズムの性能評価

4.1 Booksim

Booksim は、Stanford 大学の Dally らによって作られたサイクルベースのフリットレベルのシミュレータである<sup>9)</sup>。Booksim では、他のネットワークシミュレータ同様、特定の通信パターンを流し、定常状態のネットワークでスループットやレイテンシを測定する。Booksim は、実際のシステムに多く使われているハードウェアモデルを抽象化しており、シミュレーションによる結果が、現実マシン上での通信性能に近い精度が期待できる。本論文では、A2AT の全対全通信時間を計測し、解析的に求めた通信時間と比較、通信性能評価を行う。メッセージ単位でルーティングを行い、フリット単位でフロー制御を行う。メッセージがノー

ドからネットワークに流れるまでに1サイクル、ノードの受信処理に1サイクルかかる。また、ルータを1つ通過するのも1サイクルかかる。

本アルゴリズムの評価を行うため、Booksim に以下の拡張を行った。

- 通信パターンの追加：全対全通信アルゴリズムである、A2A, A2AND, A2AT の追加。
- 全対全通信の通信時間：全対全通信の最後のメッセージで各ノードの送信を停止させ、最後のメッセージの到着でシミュレーションを停止させることで全対全通信時間を測定する。
- ローカル同期モード：各ノードは各 Phase で他ノードからのメッセージの受信完了後、目的のメッセージの送信を開始する。ノード内での送信と受信の同期のため、オーバーヘッドが小さい。
- Wait モード：各ノードでの全対全通信の開始時刻をランダムにばらつかせる。
- 同時送信数の増加：ルータからノードへのポートを増やすことで、同時送信数  $NCT$  を増やせるように実装した。

#### 4.2 実験方法

提案した全対全通信アルゴリズム A2AT について、既存の全対全通信アルゴリズム A2AND とのシミュレーション結果から得られた通信時間を比較した。シミュレーションの条件を表 2 に示す。

表 2 シミュレーション条件

ネットワーク	k-ary 2-cube/mesh(k = 4 ~ 17) .
通信パターン	A2AND, A2AT .
パケット長	1 パケット, 100 フリット .
メッセージ長	1 メッセージ, 1 パケット .
スイッチング	ワームホール方式, バーチャルカットスルー方式 .
バッファリング	フリット単位でバッファリング .
アービトレーション	ラウンドロビンによる パケットごとのアービトレーション .
ルーティング	Dimension-order ルーティング .

#### 4.3 シミュレーション結果

##### 4.3.1 VC が宛先ノード数個ある場合

A2AT の通信性能を確認するため、アルゴリズム考察時に想定した状況に近い形である各ノードがバーチャルチャネル (VC) を宛先ノード数個持ち、バッファサイズを小さくしてシミュレーションを行った。この方法では、各ノードは宛先ごとに異なった VC を使いメッセージを送信する。各ルータでのアービトレーションはすべての VC に対して公平に行われる。よって、メッセージの通過ホップ数に依存せず、全メッセー

ジで公平なアービトレーションとなる。

バッファリングの影響をなくすため、バッファサイズを最小限である 2 とした。A2AT は偶数サイズの場合、中間距離にあるノードへの送信は、送信先に応じて逆方向のリンクを使用して送信を行うことを想定している。Booksim では、動的に空いている VC を優先して使うため、中間距離にあるノードへの送信は本来の動作とは異なる。この場合の結果を図 8 に示す。横軸はネットワークの 1 辺のサイズ、縦軸は  $NCT$  を 1 としたときの予測値  $TV_{NCT1}$  に対する比率である。

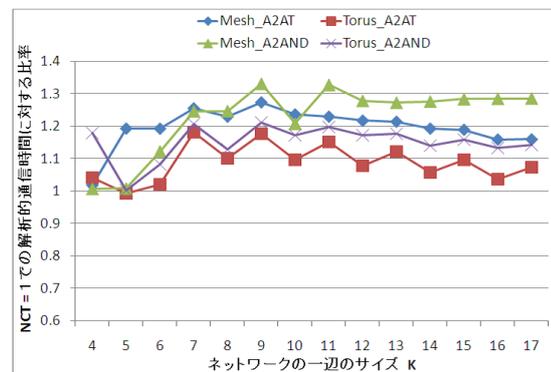


図 8 VC が宛先ノード数個ある場合 (VC = 宛先ノード数個, バッファサイズ = 2,  $NCT = 1$ )

A2AT, A2AND, いずれもネットワークサイズによらず、予測値に対する比率の変化は小さかった。A2AT と A2AND の解析的に求めた予測値は同じである<sup>6)</sup>。シミュレーションにおいても、A2AND と A2AT はほぼ同様の結果が得られた。Torus では予測値  $TV_{NCT1}$  と比較し、平均約 1.09 倍の時間で通信が完了している。予測値より長くなる理由として、ルータを通過するのに時間がかかりレイテンシが伸びた点があげられる。

Mesh では予測値と比較し、平均約 1.20 倍の時間で通信が完了している。Mesh では、メッセージの中継によりレイテンシが伸びた点に加え、右行きと左行きのメッセージでホップ数が異なることによる影響があげられる。A2AT では、各 Phase の通信が同時刻に送信を行った場合、同時刻で終わることを想定している。ホップ数が異なるために、各 Phase における各ノードでの受信開始時刻、受信完了時刻にばらつきが生じた。受信完了時刻が異なるため、送信が早く終

フロー制御情報の伝達に 1 サイクルかかるので、バッファサイズが 1 ではバイライン動作ができなくなるため、バッファサイズを 2 としている。

わったノードでは、次の Phase の通信を行うため、通信が遅れている Phase のメッセージと一部オーバーラップした。この影響で、各 Phase での受信完了時刻のばらつきは大きくなり、予測値よりも通信に時間がかかった。

#### 4.3.2 VC を 2 個とした場合

各ノードが VC を 2 個持つ場合についてシミュレーションを行った。Torus の場合、通信のデッドロックを回避するために 2 つのバーチャルチャネル (VC) を用いる。通信が境界線である dateline を超えるものと超えないものとで別々の VC を使用するように、VC を切り替える。これは、実際のシステムに多く使われている構成である。Mesh の場合、VC が 1 つでもデッドロックが発生しないが、Torus と条件を合わせるために VC を 2 つとし、動的に空いている VC を優先して使う。この結果を図 9 に示す。

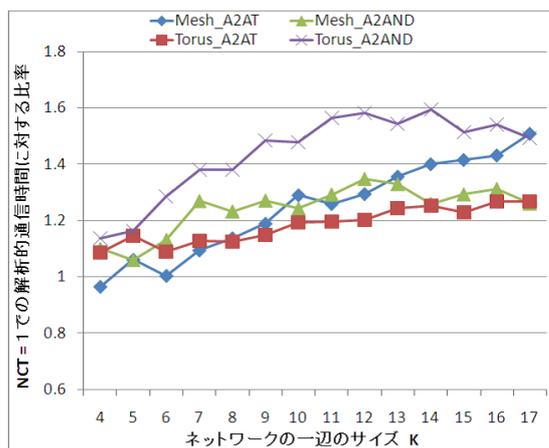


図 9 VC を 2 個とした場合 (VC = 2, パッファサイズ = 20, NCT = 1)

Torus では予測値と比較し、平均約 1.19 倍の時間で通信が完了している。VC を 2 個とした場合、VC を宛先ノード数個持つ場合のような全メッセージに対して公平なアービトレーションは行われない。同時に同サイズのメッセージを同じバンド幅で送信した場合でも、各ノードで送信完了時刻にばらつきが発生する。そのため、通信時間が長くなった。A2AT は、ネットワークサイズが大きくなると、Mesh ネットワーク上でメッセージを送った場合、A2AND よりも通信が遅い結果となった。

不公平なアービトレーションの影響による、各ノードでの通信時間のばらつきを抑えるために、ローカルに送受信の同期を取る、ローカル同期モードでシミュレーションを行った。図 10 に示すように、A2AT は

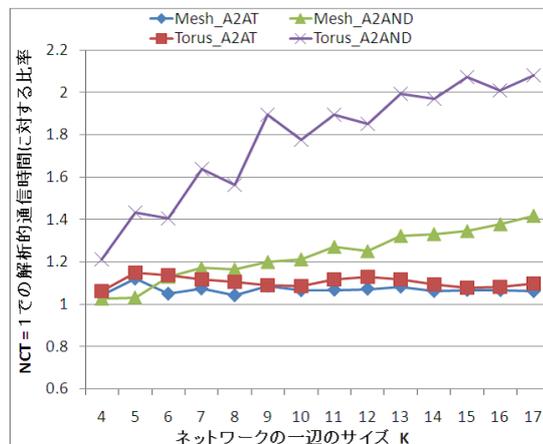


図 10 VC を 2 個とした場合 (VC = 2, パッファサイズ = 20, NCT = 1, ローカル同期モード)

A2AND に比べ Mesh では平均約 12.5%通信時間が低減され、Torus では平均約 36.0%通信時間が低減された。予測値である  $TV_{NCT1}$  と比較して、平均約 1.11 倍の時間で通信が完了している。ローカルに送受信の同期を取ったことにより、異なった Phase 間でのメッセージのオーバーラップが少なくなった。各 Phase の各ノードでの送受信完了時刻のばらつきが小さくなり、通信性能が低下しなかった。

ローカル同期モードで通信を行った場合、A2AT は、A2AND と比較し約 12.3% ~ 48.0%通信時間が低減され、ネットワークサイズが大きくなるほど、優位であった。A2AND では、同一方向への通信が連続する性質上、同一方向のメッセージの重なりが多くなる。ネットワークサイズが大きくなるほど、経路上での重なりも多くなり、不公平なアービトレーションの影響も大きくなる。そのため、ローカル同期モードでも同期による待ち時間が長くなり、通信時間が長くなった。

#### 4.3.3 通信開始時刻がばらついた場合

A2AT 考案時には、全ノードで同時刻に通信を開始する前提で設計したが、実際のシステムでは、通信の開始時刻が各ノードでばらつきが発生する。柴村ら<sup>10)</sup>により、A2AT に対しパケットペーシングを適用した性能評価が行われているが、「A2AT での通信はわずかなタイミングのずれに敏感であり、全体の通信性能の低下につながる」と報告されている。A2AT での通信開始時刻が各ノードでばらついた際の影響を調べるため、各ノードで 1 パケット送信するのにかかる通信時間に相当する時間内で全対全通信開始時刻をランダムにばらつかせた状況で、シミュレーションを行った。結果を、図 11 に示す。

A2AT では、Mesh・Torus とともに全ノードでばらつ

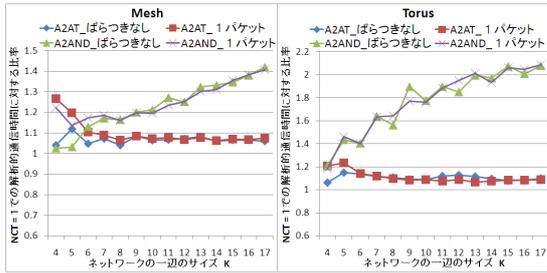


図 11 通信開始時刻をばらつかせた状況での全対全通信性能  
(VC = 2, バッファサイズ = 20, NCT = 1, ローカル同期モード, Wait モード)

きのある場合とない場合とで通信性能はほぼ同等の結果が得られた。予測値と比較して, Mesh・Torusとも平均約 1.10 倍の通信性能であった。A2AT は, ローカル同期を取った場合, 各ノードでの通信開始時刻がばらついた場合でも, 同時刻に通信が開始した場合と同等の通信性能であり, わずかなタイミングのずれは通信性能に影響を与えない。

それに対し A2AND では, 通信開始時刻がばらつきがある場合でも, 同時刻に通信が開始した場合でも, ネットワークサイズが大きくなるほど, 予測値よりも通信時間が長くなっている。4.3.2 節においても述べたが, 同一方向への通信が連続する性質上, 経路上での重なりも多くなり, 不公平なアービトレーションの影響も大きくなるためである。

図 11 をみると, ネットワークの一辺のサイズが 11, 13 では, ばらつきがない場合の方が通信に時間がかかっている。図 12 より, サイズが 11, 13 では, ばらつきがない場合の方が各 Phase での受信完了時刻の差の平均値が大きいことがわかる。そのため, 全対全通信の開始時刻がばらついた場合よりも, 不公平なアービトレーションの影響が大きくなっていると考えられる。

A2AT では, ネットワークの一辺のサイズが 4, 5 の場合, ばらつきがない場合とばらつきがあった場合とで通信性能に差がみられる。ばらつきがあった場合, 予測値と比較して Mesh・Torus では平均約 1.23 倍の通信性能である。この理由を調べるために, Torus において各 Phase での受信完了時刻の最も早く受信が完了したノードと, 最も遅く受信が完了したノードとの差を取り, 全対全通信にかかる全 Phase での平均値を調べた。結果を図 12 に示す。縦軸は全通信時間に対する各 Phase での平均ずれ時間の比率である。

ローカルに同期を取らなかった場合では, ネットワークサイズが大きくなるにつれ, 各 Phase での受信完了時刻の差も広がっていく傾向にあり, 各 Phase での

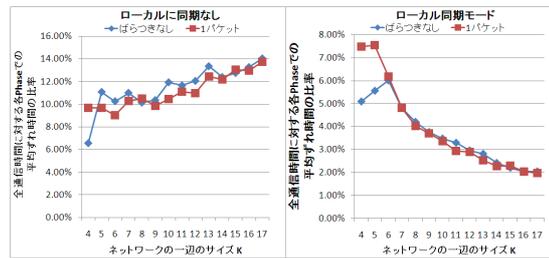


図 12 各 Phase での受信完了時刻の差  
(VC = 2, バッファサイズ = 20, NCT = 1, Wait モード)

通信性能が低下している。ローカル同期モードでは, ネットワークサイズが大きいところでは, 各ノードの各 Phase での受信完了時刻の差は, 平均約 3.6% の割合であり, 各ノードのばらつきが小さい。ネットワークサイズが小さいところをみると, 全対全通信の開始時刻がばらついた場合は, ばらつきがない場合と比べ, 受信完了時刻の差が占める比率が高いのがわかる。そのため, 全対全通信の開始時刻がばらついた場合は, ばらつきがない場合と比べ通信に時間がかかる理由である。

#### 4.3.4 NCT を増やした場合

1 パケットを 100 フリット, 1 メッセージを 1 パケット, バッファサイズは 200 フリット分とし, NCT を 2 とした場合についてもシミュレーションを行った。結果を図 13 に示す。

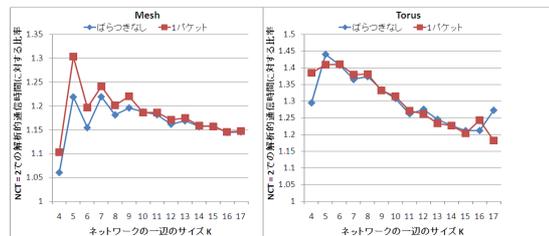


図 13 NCT を 2 とした場合での全対全通信性能  
(VC = 2, バッファサイズ = 200, NCT = 2, ローカル同期モード, Wait モード)

A2AT が NCT を 2 としたときの予測値と比較して, Mesh では平均して約 1.18 倍, Torus では平均して約 1.30 倍の時間で通信が完了している。Mesh においては, NCT を 2 とした時の予測値は, 理論的下限の通信時間  $LB_{Mesh}$  に対しての比率である。NCT を 2 としたときも, 通信開始時刻のばらつきの影響はない。

NCT を 1 としたときと比べ, Mesh では平均約 18.8%, Torus では平均約 41.2%通信時間が低減された。NCT を増やしたことで, 複数の通信コントロー

ラを用いるメリットを活かし、リンクを効率的に利用しているため、通信性能が向上している。 $NCT$  を 2 とした場合においても、わずかなタイミングのずれに対して、ローカルに同期を取れば通信性能の低下は起きない。

Torus では  $NCT$  を 4 とした場合、常に 4 方向のリンクを使用するようスケジューリングしている。 $NCT$  を 4 とした場合のシミュレーション結果を図 14 に示す。

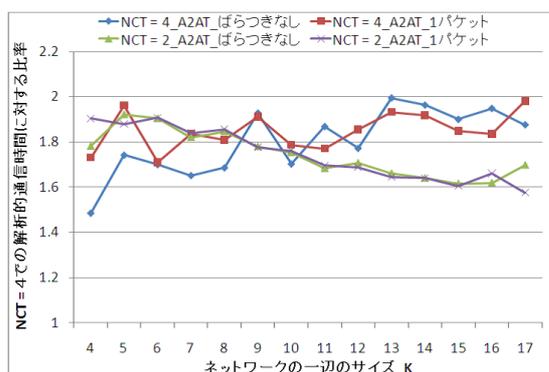


図 14  $NCT$  を 4 とした場合での全対全通信性能 (VC = 2, バッファサイズ = 200,  $NCT$  = 2, 4, ローカル同期モード, Wait モード)

$NCT$  を 4 としたときは、 $NCT$  を 2 としたときの通信時間と比較して、一辺のサイズ  $K$  が 11 以上のところでは  $NCT$  を 2 としたときよりも、平均約 14.7%通信時間が長くなっており、通信性能の向上が得られていない。一部のパケットの最後尾のフリットが宛先が空いているにもかかわらず、宛先ノードに到達できないケースがみられた。この影響により、待ち時間は長くなり各 Phase でのばらつきが大きくなり、通信性能が低下している。ネットワークサイズが小さいところでは、通信時間が低減されているケースもみられる。しかし、 $NCT$  の増加に従った通信性能の向上は得られず、原因を究明する必要がある。

#### 4.3.5 長方形の場合

長方形 2 次元 Mesh・Torus ネットワークでも正方形の場合と同様に、Mesh では  $NCT$  を 2, Torus では  $NCT$  を 4 とした場合に、常に全方向のリンクを使用するように通信をスケジューリングできる<sup>6)</sup>。

長方形の場合、まずネットワークに含まれる最大の正方形奇数サイズについてメッセージを送信する。次に、偶数サイズの場合と同様の考え方で余った各行について、メッセージを送信する。前節に示したように、一辺が偶数サイズの場合でも奇数サイズの場合とほぼ

同等の通信性能が発揮できている。このことから余った行や列への通信も同様に効率よく行えることが予想される。長方形の場合も正方形の場合と同等の結果が得られることが期待できる。

## 5. ま と め

以前提案した全対全通信アルゴリズム A2AT については、フリットレベルのネットワークシミュレータである Booksim を用いて通信性能評価を行った。

アルゴリズム考案時に想定した状況に近いケースでは、解析的に求めた予測値に近い値が得られた。Torus では予測値  $TV_{NCT1}$  に対して、平均約 1.09 倍の通信時間となった。Mesh では右行きと左行きのメッセージでホップ数が異なる影響で、Phase がそろいづらくなり、平均約 1.20 倍の通信時間となった。

VC を 2 個持つ、実際のシステムに多く使われている構成では、不公平なアービトレーションとなるため、VC を宛先ノード数個持つ場合と比べ、通信時間は長くなった。各ノードでの通信終了時間のばらつきを抑えるため、ローカル同期モードでシミュレーションを行った結果、予測値と比較し、平均約 1.11 倍の時間で通信が完了した。A2AND では同一方向への通信が連続するため、競合の影響による待ち時間が長くなり、ネットワークサイズが大きくなるほど、A2AT の優位性が高かった。

通信の開始時刻が各ノードでばらついた状況においても、ばらつきがない場合とほぼ同等の通信性能が得られた。予測値と比較し、Mesh・Torus とともに平均約 1.10 倍の通信時間となった。

ローカル同期モードでシミュレーションを行った結果、ネットワークサイズが大きいところでは、各 Phase での受信完了時刻のばらつきは、予測値に対して平均約 3.6%の割合であった。各 Phase での通信性能が低下しておらず、わずかなタイミングのずれが全体での通信性能にも影響を与えないことを示した。

$NCT$  を 2 とした場合は、通信の開始時刻が各ノードでばらついた状況においても、ばらつきがない場合とほぼ同等の通信性能が得られた。予測値と比較し、Mesh では平均約 1.18 倍、Torus では平均約 1.30 倍の通信時間となった。 $NCT$  を 2 とした場合においても、ローカルに同期を取ることで、ネットワークサイズによらず、わずかなタイミングのずれによる影響を受けず、安定した時間で通信を行うことができることを示した。

今後は、 $NCT$  を 4 としたときの通信性能低下の原因を究明すること、長方形 Mesh・Torus での評価、

A2AT を MPI ライブラリとして実装し、実機での性能測定が課題となる。

謝辞 本研究の一部は、九州大学情報基盤センターの研究用計算機システムを利用して行われました。

ゴリズムのシミュレーション評価, 情報処理学会研究報告, Vol. 2010-HPC-126, No. 14, pp. 1-9 (20100727).

#### 参 考 文 献

- 1) Worley, P. H., Barrett, R. F. and Kuehn, J. A.: Early Evaluation of the Cray XT5, *Proc. of the 51st Cray User Group Conference*, pp.1-12 (2009).
- 2) Hoisie, A., Johnson, G., J.Kerbyson, D., Lang, M. and Pakin, S.: A Performance Comparison Through Benchmarking and Modeling of Three Leading Supercomputers: Blue Gene/L, Red Storm, and Purple, *SC '06: Proc. of the 2006 ACM/IEEE conference on Supercomputing*, IEEE Computer Society, p. 3 (2006).
- 3) Bruck, J., Ho, C.-T., Kipnis, S. and Weathersby, D.: Efficient algorithms for all-to-all communications in multi-port message-passing systems, *SPAA '94: Proc. 6th annual ACM symposium on Parallel algorithms and architectures*, New York, NY, USA, ACM, pp. 298-309 (1994).
- 4) Tipparaju, V. and Nieplocha, J.: Optimizing All-to-All Collective Communication by Exploiting Concurrency in Modern Networks, *SC '05: Proc. 2005 ACM/IEEE conference on Supercomputing*, Washington, DC, USA, IEEE Computer Society, p. 46 (2005).
- 5) Ajima, Y., Sumimoto, S. and Shimizu, T.: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, *Computer*, Vol. 42, No. 11, pp. 36-40 (2009).
- 6) 高上治之, 矢崎俊志, 安島雄一郎, 清水俊幸, 石畑宏明: 2次元 Mesh ネットワーク・Torus ネットワーク上での最適全対全通信アルゴリズム, *情報処理学会論文誌 コンピューティングシステム*, Vol. 3, No. 2, pp. 88-98 (2010).
- 7) Abts, D. and Weisser, D.: Age-based packet arbitration in large-radix k-ary n-cubes, *SC '07: Proc. 2007 ACM/IEEE conference on Supercomputing*, New York, NY, USA, ACM, pp. 1-11 (2007).
- 8) : MPICH . <http://www.mcs.anl.gov/research/projects/mpi/>.
- 9) Dally, W. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003).
- 10) 柴村英智, 三輪英樹, 薄田竜太郎, 平尾智也, 安島雄一郎, 三吉郁夫, 清水俊幸, 石畑宏明, 井上弘士: パケットベージングを用いた最適全対全通信アル