計算機アーキテクチャ 97−16
（ 1992. 12. 4 ）

超並列計算機のための相互結合網シミュレータの開発

柴村 英智† 久我 守弘‡ 末吉 敏則†‡

†九州工業大学情報工学部
知能情報工学科

‡九州工業大学
マイクロ化総合技術センター

sibamura@mickey.ai.kyutech.ac.jp

超並列計算機の実現に向けて様々な相互結合網の調査を遂行するための相互結合網シミュレータについて述べる．開発したシミュレータは，相互結合網に関する諸特性をネットワーク記述言語と呼ぶ仕様記述を用いて詳細に明記するため汎用性が高く，所望する相互結合網の性能評価に大変有効である．また，実際の並列プログラムから取得した通信パターンを基にシミュレーションを行うことができるため，より具体的な性能を把握することが可能である．本稿では，超並列計算機のための相互結合網の構成について述べた後，本シミュレータを用いてピラミッド網とトーラス網のシミュレーションによる通信性能の評価を行う．

Development of an Interconnection Network Simulator for Massively Parallel Computer

Hidetomo Shibamura† Morihiro Kuga‡ Toshinori Sueyoshi†‡

† Department of Artificial Intelligence
Kyushu Institute of Technology
Iizuka, 820 Japan

‡ Center for Microelectronic Systems
Kyushu Institute of Technology
Iizuka, 820 Japan

sibamura@mickey.ai.kyutech.ac.jp

This paper presents an interconnection network simulator to investigate the performance of various interconnection networks toward realization of massively parallel computer. Our developed interconnection network simulator provides efficient features to evaluate an interconnection network which is desired using network description language which specify the characteristics of interconnection network in detail, and besides can grasp the performance of interconnection network due to simulate with some communication patterns which obtained from actual parallel programs definitely. Pyramid and torus network are simulated using this simulator and are evaluated with the communication performance respectively.

1. はじめに

現在，数千台以上のプロセッサ要素から構成される超並列計算機に関した研究が盛んに行われている．また並列計算機の重要な構成要素のひとつである相互結合網については，多種多様のトポロジーが提案かつ研究されている．しかし，現在に至るまで開発されてきた並列計算機の多くは特定のトポロジーしか採用しておらず，今後，様々なトポロジーを持つ並列計算機の開発が期待できる．相互結合網を特徴づける要素としてトポロジー，ルーティング方式，フロー制御方式が挙げられる[1]．これらの要素が適切に設定されていない場合においては，高い並列処理効率を得ることができない．超並列計算機の様に大規模な相互結合網の設計が必要な場合には，上記の要素に加えハードウェア性能，実現性を総合的に熟慮したうえでこれらの適切な選択がさらに重要となる．従って，相互結合網の性能を把握するために大規模なシミュレーションが必要である．

相互結合網の性能評価において並列処理効率を左右する要因のひとつに，プロセッサ要素間の通信パターンが挙げられる．通信パターンは実行する並列プログラムの時間的局所性および物理的局所性により変化し，通信遅延時間に影響を及ぼすため，相互結合網の能力を一意に定められない．最近では相互結合網の性能評価のため，理論解析やシミュレーションが行われているが，様々な通信パターンを踏まえた性能評価は，ほとんど行われていない[2~5]．

本研究では静的網を対象とし，様々な相互結合網の能力を調査するために相互結合網シミュレータの開発を行った．本シミュレータはネットワーク記述言語と呼ぶ仕様記述によって，相互結合網の諸特性を明記するため，汎用性が非常に高い設計になっている．また，実際の並列プログラムの通信パターンを用いてシミュレーションを行える特徴を備えており，相互結合網の詳細な性能評価を遂行することができた．

本稿では，この相互結合網シミュレータの概要，ネットワーク記述言語，通信パターンの取得方法およびシミュレーションを行った相互結合網の性能評価について述べる．以下，第2章では開発した相互結合網シミュレータの仕様，構成および情報収集機能について述べる．また，第3章では，相互結合網に関した諸特性を明示するネットワーク記述言語の仕様について述べる．第4章では相互結合網の詳細な性能を評価するために，実際の並列プログラムの通信パターン取得について述べ，さらに5章ではピラミッド網およびトーラス網に関して，本シミュレータを用いて行った性能評価について述べる．

2. 相互結合網シミュレータ

2.1 要件

相互結合網には大きく分けて動的網と静的網があり，シミュレーションおよび性能評価は両方について行うことが望ましい．静的網は動的網と異り，データを転送する際の経路を決定するルーティング方式を十分考慮しなければならない．本研究ではルーティング・アルゴリズムの変化に対する性能調査も行えるよう，シミュレーションの対象とする相互結合網は静的網を前提とした．また，MIMD型の並列計算機を意識してハードウェアの諸機構を非同期動作モード，分散制御方式およびパケット交換方式とした．

シミュレータを用いて相互結合網の性能を調査する場合，相互結合網を特徴づける要素の選択を，容易に行えることが望まれる．また，より具体的な性能を得るためにハードウェアの仕様も変更できることが望ましい．そこで，シミュレータの開発にあたり以下のような要求を満す必要がある．

(1) トポロジー：大規模な相互結合網では様々な結合形態が考えられるため，意図するノード間を柔軟かつ容易に接続および変更できること．

(2) ルーティング方式：ルーティング方式には様々なルーティング・アルゴリズムをシミュレートできるようにし，かつ容易に変更できること．また，ノードあるいはネットワークに関する情報をルーティングの最に必要とするアルゴリズムにも考慮して，これらの情報を得る機構を実現すること．

(3) フロー制御方式：代表的なフロー制御方式であるstore and forward[1], wormhole[1], virtual cut-through[1], virtual channel[6]を使用できること．また，これらの実現に伴うバッファ長，パケット長，フリット長をそれぞれ指定できること．

(4) ネットワークのハードウェア仕様：ネットワークの転送バンド幅，動作周波数をそれぞれ指定できること．

(5) ノードのハードウェア仕様：ノードにおけるネットワーク・インタフェース内部でのルーティング制御，および，パケットやフリットの操作に伴う処理遅延時間を指定できること．

一方，ノード間の通信性能は並列プログラムの実行性能に大きな影響を与える．特に数値計算の場合，複数のノードから相互結合網を介してデータを授受し，計算結果を再び相互結合網へ送出する．そのためネットワークおよび中継ノードにおける遅延に焦点が置かれ，遅延減少のための様々な研究が行われている．

Johnson[2]は相互結合網のシミュレーションにおいて個々のプロセッサの振舞いをアプリケーション・モデルとして枠組みを定義し，通信パターンを理論的にモデル化して並列計算機の性能評価を行っている．しかし実際の通信パターンには一連の物理的・時間的局所性が含まれるものの，並列プログラムの異によりこれらの局所性は時事刻々と変化する．そのため，並列プログラムから得られる実際の通信パターンを用いてシミュレーションを行うことが，より詳細な相互結合網の性能評価に重要である．よって，相互結合網へ投入する通信パターンは以下のような要求を満す必要がある．

(6) 通信パターンの使用：並列プログラムの通信パターンは並列計算機の実行性能に大きな影響を与える．この影響を調べるために，並列プログラムを実行する最に得られる通信パターンを取込み，詳細な相互結合網の性能評価を行えること．

(7) メッセージの生成：(6)で通信パターンを使用しない場合，シミュレータがメッセージを生成できるようにする．その際，メッセージの発生間隔，距離別の発生割合，メッセージ長の制限範囲をそれぞれ指定できること．

相互結合網の性能評価を行うために，シミュレータは次の機構を備える必要がある．

(8) 情報収集機構：相互結合網の性能を評価するにあたり，有効なシミュレーション情報を収集できること．

2.2 相互結合網シミュレータの構成

2.1節で述べた要件をすべて満足するように本シミュレータを開発した．特に，(1)~(5),(7)を満すために，本シミュレータでは相互結合網に関する様々な情報を明記するためのネットワーク記述言語を用意した．すなわちネットワーク記述言語で書かれたネットワーク記述ファイルを変更することにより，上記(1)トポロジーに関して柔軟に対応できる．すなわち，任意のトポロジーを持つ相互結合網のシミュレーションを行うことができる．(2)ルーティング方式に関しては，現状ではシミュレータ実現の容易なことからdeterministic routingを用いて静的なルーティングを行った．なおルーティング方式に関するネットワーク記述言語向きの表記仕様を検討中である．(3)フロー制御方式に関してはstore and forward, wormhole, virtual cut-throughを現在サポートしており，ネットワーク記述言語を用いて選択することができる．また，(4)ネットワークおよび(5)ノードのハードウェア仕様も詳細に記述可能であるため，所望の相互結合網の実現に大変効果がある．

シミュレータの構成を図1に示す．本シミュレータは大きく分けて，ネットワーク記述言語解析部およびシミュレータ・エンジンの2つから構成される．ネットワーク記述言語解析部は，ネットワーク記述言語によって記述されたネットワーク記述ファイルから相互結合網に関する情報を抽出し，シミュレーションを行うために必要な情報を生成する．シ

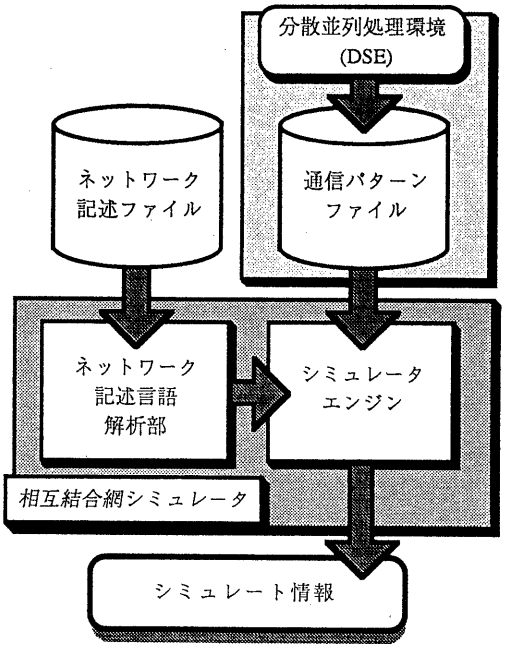


図1. 相互結合網シミュレータの構成および実行環境

ミュレータ・エンジンでは，4章で述べる分散並列処理システムから取得した通信パターン・ファイルとネットワーク記述言語解析部からの情報を基にシミュレーションを行う．また，通信パターン・ファイルを使用せず，メッセージをメッセージの距離（メッセージの送信元から送信先までの距離）別にランダムに発生させてシミュレーションを行うことも可能である．そして，最終的に次節で述べる様々なシミュレート情報を出力する．

2.3 相互結合網シミュレータの情報収集機能

相互結合網の性能を評価するために収集する情報は以下に挙げる要求を満す必要がある．

(1) ネットワークにおける遅延は相互結合網の性能を大きく左右するため，あらゆる要素の遅延時間を収集できること．

(2) 相互結合網が持つ転送能力を知るために，相互結合網へ投入されるメッセージ数および相互結合網を通過したメッセージ数を収集できること．

(3) メッセージの転送におけるネットワークの負荷状態を把握できること．

これらの要求を満すために，本シミュレータは次に示す情報を収集する．

- ネットワーク全体におけるメッセージ・パケットの平均遅延時間
- 距離別ごとのメッセージおよびパケットの平均遅延時間
- 単位時間あたりにネットワークへ注入されるメッセージ長の総計およびメッセージの総数
- 単位時間あたりにノードへ到着したメッセージ長の総計およびメッセージの総数
- ネットワークへ残留しているメッセージ長の総計およびメッセージの総数
- 各ノードにおける衝突頻度

これらの情報を用いることにより，相互結合網の詳細な性能の解析を行うことができる．

3. ネットワーク記述言語

3.1 言語仕様

本シミュレータでは対象とする相互結合網に関し

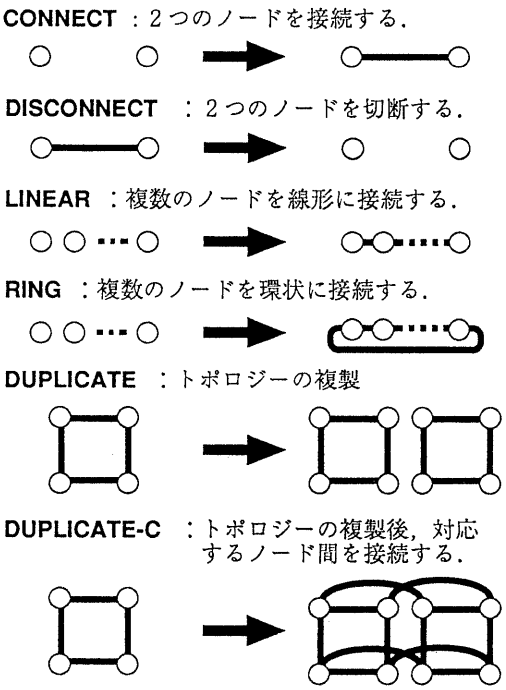


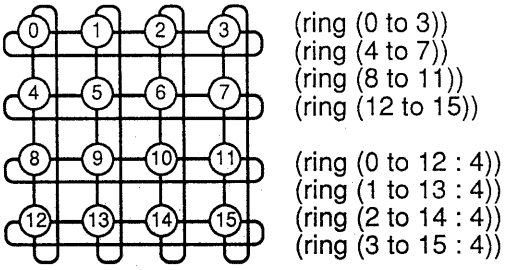
図2. ネットワーク記述言語におけるノード接続命令

た情報を，ネットワーク記述言語を用いて明記する．相互結合網の仕様のドキュメント化にも利用できる様に表記方法は単純かつ明確に行えるように配慮した．

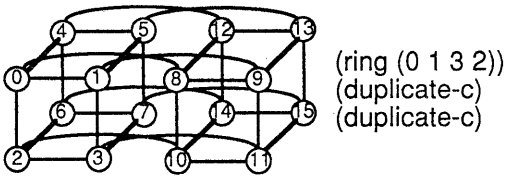
ネットワーク記述言語は大きく分けてネットワークに関する情報，ノードに関する情報，メッセージに関する情報の3つから構成される．

(1) ネットワークに関する情報

- トポロジー：図2に示すノード間の接続命令を用いてノード間の結合形態を指定する．また，数値のインクリメント機能，トポロジーの複製機能を備えることにより，複雑な形態も極めてシンプルなスタイルで表現できる．これらの接続命令を用いることにより容易にトポロジーを構築・変更することが可能である．
- ルータの構成方式：フロー制御方式を指定する．
  例） (FlowControlMethod StoreAndForward)
- ネットワークのハードウェア仕様：ネットワークの転送バンド幅，動作周波数を指定する．



(a) トーラス網



(b) ハイパーキューブ網

図3. ネットワーク記述言語による相互結合網の構築例

例） (NetworkBandWidth 32 bits)
(NetworkFrequency 25 MHz)

(2) ノードに関する情報

・ノードのハードウェア仕様：ノードの内部処理遅延，バッファ長を指定する．

例） (NodeHardwareDelay 10 μsec)
(NodeBufferSize 32 bytes)

(3) メッセージに関する情報

・メッセージの発生間隔：ノードから発生するメッセージの最低発生間隔を指定する．

例） (MessageCreationInterval 10 μsec)

・メッセージの発生割合：ノードから発生するメッセージの割合を各距離別に指定する．

例） (MessageCreationRate Hop3 0.0001 %)

・メッセージ長の制限：ノードから発生するメッセージ長の範囲を指定する．

例） (MessageLength 16 1024)

なお，時間に関する指定は1nsec単位で指定することができる．

3.2 ネットワーク記述例

図3にノード接続命令によるトーラス網とハイパーキューブ網の構築例およびネットワーク記述言語による表記を示す．図3 (a)のトーラス網では(ring (0 to 3))によりノード0，1，2，3が環状に接続され，以下水平方向の接続を行う．同様に(ring (0 to 12 : 4))によりノード0，4，8，12が環状に接続され，最終的にトーラス網が構築できる．図3 (b)のハイパーキューブ網では(ring (0 1 3 2))によりノード0，1，3，2が環状に接続される．さらに次の(duplicate-c)により環状接続されたノード0，1，3，2が複製され対応するノード同士が接続される．すなわち左半分の立方体が構築される．引続き(duplicate-c)を行うことによりハイパーキューブ網が構築できる．

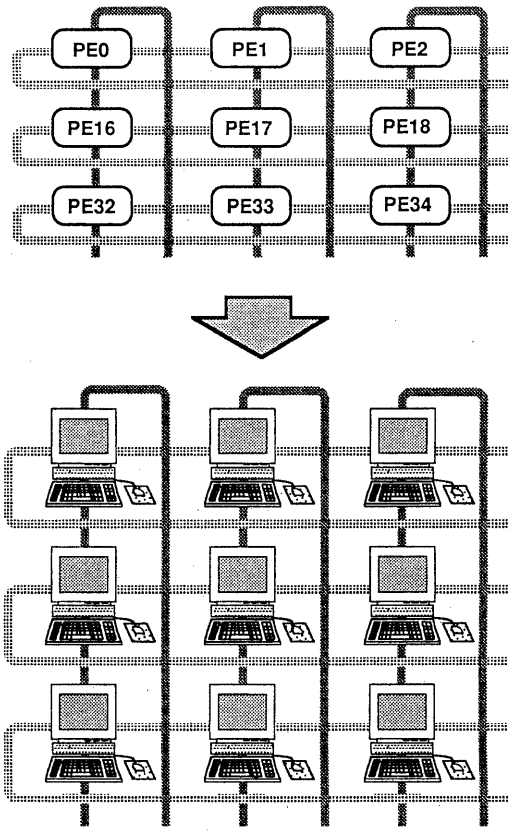


図4. DSE：分散並列処理システム

4. 通信パターンの生成

様々なアプリケーションの通信パターンを取得するために，我々の研究室で構築したDSE (*Distributed Supercomputing Environment*)と呼ぶ分散並列処理システムを使用した[7][8]．DSEシステムは，複数のワークステーションを通信ネットワークで接続した分散

処理環境上に構築されており，分散共有メモリ型の構成をとる．図4に示すように，DSEシステムは用意された接続ファイルを基に各ワークステーションをプロセッサ要素として割付け，それぞれを接続して並列処理環境を実現するため，意図する相互結合網と同等の結合形態を容易に実現することが可能である．また，各プロセッサごとに入出力メッセージの履歴を保存できるため，実際の並列プログラムの実行に伴なう通信パターンをはじめとした各種情報を取得することが可能である．

本シミュレータで使用する通信パターンを得るためにSun microsystems社のSPARC station 2を16台使用し，その上でDSEシステムを使用し通信パターンを生成した．各々のマシンを仮想化することにより1台あたり16個のプロセッサ要素を割り付けた．この環境上で並列プログラムを実行させ，得られた通信パターンを，最終的に256台のプロセッサ要素を持つ並列計算機上での通信パターンとした．

5. シミュレーションによる性能評価

本シミュレータを用いて，ピラミッド網およびトーラス網についてシミュレーションを行った．前者は通信パターンを使用せずシミュレータによりメッセージを発生させた．後者は8×8の行列演算についてDSEによりアクセスパターンを生成し用いた．

5.1 ピラミッド網

図5に示す最下層部に8×8のメッシュ構造を持つ2次元ピラミッド構成の相互結合網について性能評価を行った．ネットワークの性能パラメータは以下のように設定した．

- フロー制御方式：store and forward
- 転送バンド幅：32 bits
- 動作周波数：25 MHz
- ハードウェア遅延：0 μsec.
- メッセージの発生割合：0.01 % (100nsecにつき1回程度発生)
- メッセージ長：32 bytes固定長

このシミュレーションでは通信パターンを用いず，メッセージの発生をランダムに行い，メッセージの発生間隔を密(200nsec)から疎(30000nsec)に変化

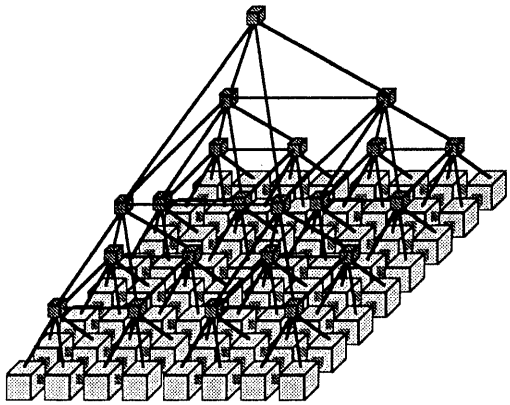
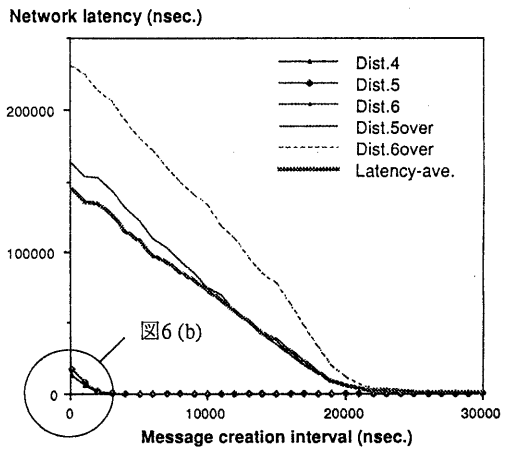
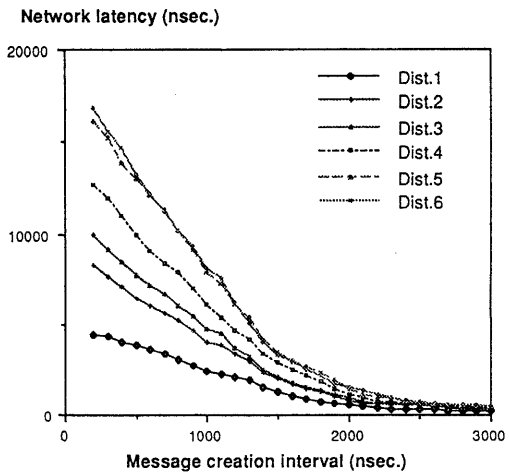


図5. 2次元ピラミッドの構成

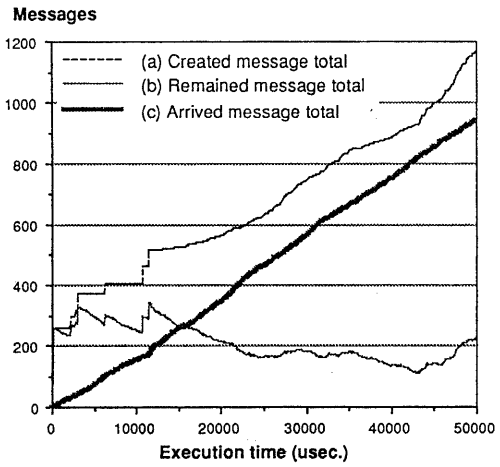


(a) 遠距離通信性能

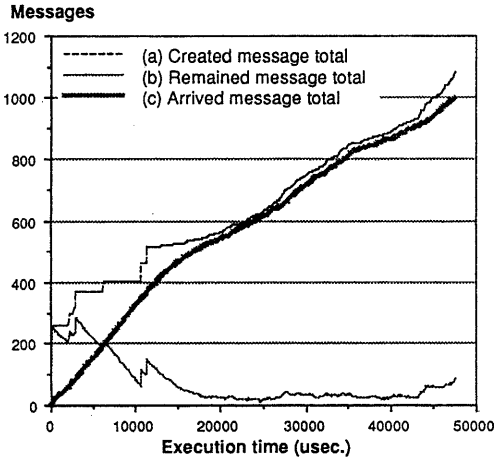


(b) 最下層部における通信性能

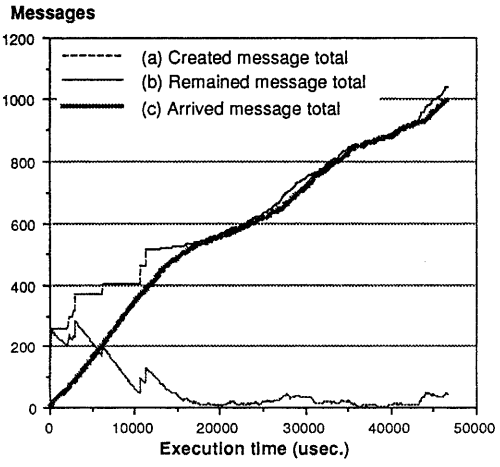
図6. ピラミッド網の通信性能



(a) パケット長8バイト



(b) パケット長16バイト



(c) パケット長32バイト

図7. トーラス網におけるメッセージ数の変化

させた．評価指標としては，最初の1,000個のメッセージが到着するまでの遅延時間について評価した．また，メッセージに関しては送信元から送信先までの距離(1~6)ごとに測定した．

距離が4以上のメッセージについて遅延時間を調べたものを図6(a)に示す．これはピラミッド網内において遠距離ノード間の通信遅延を表わしている．この図から通信遅延が非常に大きなもの(Dist.5over, Dist.6over)，逆に非常に小さなもの(Dist.4, Dist.5, Dist.6)があることがわかる．遅延が大きなものは最下層のメッシュ網から上層のノードを経由してメッセージが転送される．上層部へメッセージが上昇するほどメッセージの衝突が頻繁に発生するようになるため，通信遅延が非常に大きくなることが言える．また通信遅延が非常に小さなものは上層部のノードを経由せず，メッシュ網内でメッセージの転送が行われる場合である．メッシュ網内だけでメッセージの転送が行われる場合の遅延を図6(b)に示す．この図から距離に比例して遅延時間が大きくなっていることがわかる．

以上のことからこのピラミッド網においては，最下層部のメッシュ網の通信性能が非常に高いという結果が得られた．

5.2 トーラス網

次に16×16のノードを持つトーラス網について性能評価を行った．ネットワークの性能パラメータを以下に示す．

・フロー制御方式：store and forward

・転送バンド幅：32 bits

・動作周波数：25 MHz

・ハードウェア遅延：100 μsec.

このシミュレーションでは8×8の行列演算の積を5回行う並列プログラムから，通信パターンを取得しそれを用いた．通信パターンの特徴としては，まず，行列の各要素の値を共有メモリへ参照しに行くため，単位時間当りに大量のメッセージが発生する．その後演算が開始されるとメッセージの発生は比較的穏やかになり，ほぼ一定量の通信を行う．演算後，結果を共有メモリへ格納するため再び大量のメッセージが発生する．5回の積を求めるのに必要なメッセージ数は，通信パターンからおよそ9200個必要であった．

評価指標としては1,000個のメッセージが到着するまで，(a)時間経過と共に発生したメッセージの総数，(b)ネットワーク内に残留しているメッセージの総数，(c)ノードへ到着したメッセージの総数を表している．また，パケット長を8バイト固定長，16バイト固定長，32バイト固定長と変化させた各々のパケット長における測定結果をそれぞれ図7 (a)(b)(c)に示す．パケット長が8バイトの時(図7 (a))，メッセージの大きさに対してパケット長が小さすぎるため，経由ノードでの遅延が積算されてネットワーク内にメッセージが大量に残留することがわかる．また，図7 (b)(c)から，ある程度パケット長が大きくなるとメッセージの残留はかなり減少することがわかる．パケット長を増加させた場合，メッセージの残留は減少するが，パケット内に有効な情報が少ししか存在しないため，断片化が問題となる．

6. おわりに

本稿では超並列計算機のための相互結合網シミュレータの概要とシミュレータを用いた性能評価について述べた．本シミュレータを用いることにより，目的とする相互結合網の性能を容易に把握することができ，並列計算機の研究，開発に大きな指針を与えるであろう．現在，より詳細なシミュレーションを行うため，ネットワーク記述言語の拡張，情報収集機能の向上を行っている．

謝辞

シミュレーションに必要な通信パターンをDSEを用いて取得していただいた本学情報工学研究科院生の手塚忠則氏ならびに知能情報工学科学部生の岡雅樹氏に感謝の意を表す．また，シミュレータの開発にあたり，日頃から貴重な御助言，御討論いただく本学情報工学部有田・末吉研究室の諸氏に感謝する．なお，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04235103「超並列ハードウェア・アーキテクチャの研究」)の補助を受けたことを付記する．

参考文献

[1] William J. Dally : Network and Processor Architecture for Message-Driven Computers, *VLSI and PARALLEL COMPUTATION, Edited by Robert Suaya and Graham Birtwistle*, pp. 140-222 (Chapter 3), 1990.

[2] William J. Dally : Performance Analysis of *k*-ary *n*-cube Interconnection Networks, *IEEE Transactions on Computer.*, vol. 39, no. 6, pp. 775-785, June 1990.

[3] Kirk L. Johnson : The Impact of Communication Locality on Large-Scale Multiprocessor Performance, *Proceedings of 19th Annual International Symposium on Computer Architecture*, pp.392-402, May 1992.

[4] 柴村 英智，久我 守弘，末吉 敏則：超並列計算機のためのネットワークシミュレータ，平成4年度電気関係学会九州支部連合大会論文集 1009, October 1992.

[5] 船越 誠司，久我 守弘，末吉 敏則：ハードウェア記述言語による相互結合網のモデル化と性能評価，第45回情報処理学会全国大会講演論文集 7L-05, October 1992.

[6] William J. Dally : Virtual-Channel Flow Control, *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp.192-205, March 1990.

[7] B. O. Apduhan, T. Sueyoshi, Y. Namiuti, T. Tezuka and I. Arita : Experiments of A Reconfigurable Multiprocessor Simulation on A Distributed Environment, *Proceedings of 1992 IEEE International Phoenix Conference on Computers and Communication*, pp.539-546, April 1992.

[8] T. Tezuka, K. Ryokai, B. O. Apduhan and T. Sueyoshi : Implementation and Evaluation of a Distributed Supercomputing Environment on a Cluster of Workstations, *Proceedings of 1992 International Conference on Parallel And Distributed Systems*, December 1992.