

Title	優先順位に基づく仮想チャネルフロー制御に関する研究
Author(s)	三浦, 康之
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1253
Rights	
Description	Supervisor:堀口 進, 情報科学研究科, 修士

修 士 論 文

**優先順位に基づく
仮想チャネルフロー制御に関する研究**

指導教官 堀口 進 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

三浦 康之

1999 年 2 月 15 日

目次

1 序論	7
1.1 研究の背景と目的	7
1.2 本論文の構成	9
2 ワームホールルーティングおよび仮想チャネル	10
2.1 はじめに	10
2.2 ワームホールルーティング	10
2.3 仮想チャネル	13
2.4 仮想チャネルのフロー制御	16
2.5 従来の仮想チャネルフロー制御法	17
2.6 まとめ	18
3 優先順位に基づく仮想チャネルフロー制御法	19
3.1 はじめに	19
3.2 従来の仮想チャネルフロー制御の問題点	19
3.3 優先順位による仮想チャネルフロー制御	22
3.4 まとめ	25
4 仮想チャネルフロー制御システム	26
4.1 はじめに	26
4.2 システム構成	26
4.2.1 ハードウェア	26
4.2.2 通信の手順	28
4.3 シミュレータの構成	31
4.4 まとめ	32

5	二次元格子結合網における性能評価	33
5.1	はじめに	33
5.2	二次元格子結合網	33
5.3	ランダム通信による性能評価	35
5.3.1	実験方法	35
5.3.2	平均通信時間および平均スループット	36
5.3.3	考察	44
5.4	FFT による性能評価	48
5.4.1	FFT のマッピング	48
5.4.2	実験の諸条件	50
5.4.3	通信に要する時間および実行時間の比較	50
5.5	適応型ルーティング	53
5.5.1	ルーティング法	53
5.5.2	ランダム通信の平均転送時間および平均スループット	56
5.6	まとめ	57
6	階層型相互結合網 TESH における性能評価	58
6.1	はじめに	58
6.2	TESH のネットワーク構成	58
6.3	仮想チャネルフロー制御システム	60
6.3.1	ルーティングアルゴリズム	60
6.3.2	デッドロックフリーの保証	62
6.3.3	階層型相互結合網のための仮想チャネルフロー制御	63
6.4	ランダム通信による性能評価	63
6.4.1	評価方法	63
6.4.2	平均通信時間および平均スループット	63
6.4.3	考察	66
6.5	FFT による性能評価	71
6.5.1	マッピング手法	71
6.5.2	通信に要する時間の比較	71
6.5.3	実行時間の比較	73
6.6	四方向通信による性能評価	75
6.6.1	通信方法	75

6.6.2	通信時間および実行時間	77
6.7	まとめ	78
7	結論	80
	謝辞	82
	参考文献	83
	研究業績	85

目 次

2.1	ストアアンドフォワード	12
2.2	ワームホールルーティング	12
2.3	パケットのブロック	13
2.4	仮想チャネルの利用によるブロックの回避	14
2.5	デッドロックの例	15
2.6	仮想チャネルによるデッドロックの回避	15
2.7	仮想チャネルの物理的構成	16
2.8	ラウンドロビンのフロー制御	17
3.1	ネットワーク性能の低下の要因	20
3.2	ラウンドロビンにより発生する空白	21
3.3	占有法のフロー制御	23
3.4	占有法による空白の抑制	24
4.1	システム構成	27
4.2	通信の手順	30
5.1	格子結合網	34
5.2	実験に用いる二次元メッシュ	36
5.3	ランダム通信の平均転送時間（パケット長 16）	37
5.4	ランダム通信の平均スループット（パケット長 16）	38
5.5	ランダム通信の平均転送時間（パケット長 32）	39
5.6	ランダム通信の平均スループット（パケット長 32）	40
5.7	ランダム通信の平均転送時間（パケット長 64）	41
5.8	ランダム通信の平均スループット（パケット長 64）	42
5.9	転送先が集中した通信の平均転送時間	43
5.10	リンクの稼働率	44

5.11 フリットが転送されない原因	45
5.12 パケットが存在しないために転送を行なわないリンクの数	46
5.13 フリット間に空白があるために転送が行なえないリンクの数	46
5.14 前方のパケットにブロックされるために転送が行なえないリンクの数	47
5.15 FFT の通信パターン	49
5.16 FFT における各手法の平均通信時間	51
5.17 FFT の実行時間	52
5.18 Turn モデル	53
5.19 north last 法によるランダム通信の平均通信時間	56
5.20 north last 法によるランダム通信の平均スループット	57
6.1 基本モジュールの構成	59
6.2 TESH(2,2,0) の構成	60
6.3 TESH におけるランダム通信の平均転送時間	64
6.4 TESH におけるランダム通信の平均スループット	65
6.5 リンクの稼働率	67
6.6 パケットが存在しないために転送を行なわないリンクの数	67
6.7 フリット間に空白があるために転送が行なえないリンクの数	68
6.8 前方のパケットにブロックされるために転送が行なえないリンクの数	68
6.9 TESH におけるランダム通信の転送時間分布 (チャンネル数 4)	69
6.10 TESH におけるランダム通信の転送時間分布 (チャンネル数 8、パケット生 成率 1.0×10^{-3})	70
6.11 TESH におけるランダム通信の転送時間分布 (チャンネル数 8、パケット生 成率 1.2×10^{-3})	70
6.12 FFT における各手法の平均通信時間	73
6.13 FFT の実行時間の比較	74
6.14 四方向通信の BM 間通信	76
6.15 四方向通信の平均通信時間	77
6.16 四方向通信の実行時間	78

表 目 次

4.1	シミュレーションに用いたパラメータ	32
5.1	実験の種類および条件	35
5.2	転送が行なえないリンク数の平均値	45
5.3	FFT のシミュレーションのための諸条件	50
6.1	デッドロックフリーを保証するために必要なチャネル数	62
6.2	転送が行なえないリンク数の平均値	66
6.3	FFT のシミュレーションのための諸条件	72

第 1 章

序論

1.1 研究の背景と目的

計算機の性能を大幅に向上するための手段として、プロセッサを多数結合して並列に計算を行う並列計算機が注目されている。特に、動画像処理やシミュレーションなどのような、大量のデータを扱い膨大な量の計算を行う必要のある分野では並列計算機は重要な存在となっている。

並列計算機では、複数の PE(Processing Element) が並列に計算を行った結果を互いにやり取りしながら一連の処理を行っている。この時、通信の部分がボトルネックとなり十分な性能向上を得られないことが多い。したがって、PE 間の通信時間を削減することが並列計算機の性能向上のための重要なファクターとなる。通信時間を削減するための方法としては、メッセージ転送の前後にかかる時間を削減する方法と、メッセージ転送時間そのものを削減する方法の二つに大きく分けられる。転送前後の時間を削減する手法として、清水 [1] らは、キャッシュメモリやメッセージ用のバッファとネットワークの間で直接メッセージをやりとりする方法を提案している。並列計算機 FLASH[2] は、転送前後の処理の重要性を考慮してプロトコル処理専用のプロセッサを設けることにより柔軟な前後処理を行える設計となっている。また、RWC-1[3] のノードプロセッサである RICA は、通信と演算を融合することにより、通信の高速化を目指している。

ネットワーク上のメッセージ転送を改善するための研究としては、転送方式の改良やルーティングの改善などが挙げられる。転送方式としては、ストアアンドフォワード、ワームホールルーティング [4]、バーチャルカットスルー [5] などが存在する。

ストアアンドフォワードは、各 PE にパケット全体を保持できる数のバッファを持ち、いったんパケット全体を受け取ってから次の PE へ転送する方式である。バーチャルカッ

トスルーも同様に、各 PE にパケット全体を保持できる数のバッファを持つが、ストアアンドフォワードと異なり、パケット全体を受け取るまで次の PE への転送を待たずに転送を行なえる。これに対しワームホールルーティングは、各 PE にパケット全体を保持できる数のバッファを持たずに、フリットをパイプライン状に転送する手法である。これらの手法のうち、ストアアンドフォワードは転送に時間がかかるため、通常はワームホールルーティングまたはバーチャルカットスルーが用いられる。また、転送するパケットのサイズが大きいシステムは、バーチャルカットスルーを用いるとハードウェア的なコストが大きいので、ワームホールルーティングが用いられる。

ワームホールルーティングを用いた場合、パケットの衝突によるブロッキングが多くなるという問題がある。そこで、PE の間に複数の仮想的な転送経路（仮想チャンネル）を設けることによって、ブロッキングを防止する。仮想チャンネルは、複数の仮想的な配線で一つの物理的な配線を共有することにより、あたかも配線が複数存在するように利用する手法である。仮想チャンネルは、Dally ら [6] によって提案され、ブロッキング回避やデッドロック回避などのために多くの並列計算機に実装されている。デッドロック回避の例としては、転送が行われるごとに次元数を上げてゆくことでチャンネルの循環を防ぐ構造化バッファ法 [7] や、メッシュやトーラスなどにおいて、PE 番号が増加する方向に向かうパケットと減少するパケットで別のチャンネルを使用する double Y-Channel ルーティング [8] などがある。

また仮想チャンネルは、ワームホールルーティングとともに用いるとコスト対効果が良いという特徴がある。それは、以下の理由による。

- ワームホールルーティングでは必要となるバッファサイズが小さいため、仮想チャンネルを付加するために必要なコストが少なくて済む。
- ワームホールルーティングはパケット同士によるブロッキングが頻繁に起こるため、仮想チャンネルを付加することによる性能の向上が大きい。

仮想チャンネルでは複数の仮想チャンネルから一つを選択して物理リンクを通すため、チャンネルの優先順位をどのように決定するかというルール決め方（仮想チャンネルフロー制御法）によりパケットの流れが異なる。そのため、従来いくつかの研究がなされている。例えば、Rexford ら [9] は、リアルタイムパケットの処理を行うために優先順位制御を行うルータの設計を行っており、Song ら [10] は、優先順位の異なるパケット同士で流量を調整する throttle policy および優先順位の高いパケットがリンクを占有する preempt policy によりリアルタイム性の向上を図っている。また戸田ら [11] は、パケット先頭の優先順位情報をパケットの後方に送ることでリアルタイム性の向上を行っている。

しかしながら、これらはいずれもリアルタイム性を目指した研究であり、それ以外については、現状では十分に検討されているとは言えない。例を挙げると、Song ら [10] の研究ではリアルタイム以外の通信は単純なラウンドロビンでフロー制御を行っている。また、並列計算機 JUIMP-1[12] でも、仮想チャネルのフロー制御にラウンドロビンを採用している。実際、ワームホールルーティングでは、ラウンドロビンにより仮想チャネルフロー制御を行うと、調停が行われる全パケットに大きな遅れが公平に発生する。また、階層型相互結合網のようにトラフィックが不均一になるような場合についても、不均一性を考慮したフロー制御についての研究はなされていない。

以上のことを踏まえて本研究では、メッセージに適切な優先順位を設けた仮想チャネルフロー制御方式を提案し、その性能を評価することを目的とする。まず、ワームホールルーティングと仮想チャネルに関して、現状における問題点を考察する。次に、従来法に代わる仮想チャネルフロー制御手法を提案し、二次元格子網によりその有用性を確かめる。さらに、階層型相互結合網の特性を生かした仮想チャネルフロー制御手法を提案し、階層型相互結合網の一種である TESH(Tori connected mESHes)[13] を対象とした実験により、その有用性を示す。

1.2 本論文の構成

本論文の構成は次の通りである。第2章では、ワームホールルーティングと仮想チャネルについて説明し、その構造および問題点について述べる。第3章では、まず優先順位を持ったフロー制御法を説明し、第4章でシミュレーションに用いたシステムの構成を説明する。第5章では、格子結合における各仮想チャネルフロー制御手法の性能を評価する。第6章では、階層型相互結合網である TESH における仮想チャネルフロー制御法について説明し、さらにその性能を評価する。第7章は結論である。

第 2 章

ワームホールルーティングおよび仮想チャネル

2.1 はじめに

本章では、ワームホールルーティングと仮想チャネルについて説明し、その構造および問題点について述べる。

パケット転送方式の一つであるワームホールルーティングは、ストアアンドフォワードやバーチャルカットスルーのような他の転送方式に対していくつかの利点を持った方式である。反面、パケット同士の衝突によるブロッキングが多くなるために仮想チャネルの付加が行われる。

本章では、まず 2.2 でパケット転送方式の一つであるワームホールルーティングの手法と、他の転送方式と比較した利点について説明する。次に、2.3 で、仮想チャネルの使用目的とワームホールルーティングとの関係について説明した後、2.4 で仮想チャネルフロー制御が必要な理由を、仮想チャネルの機構と合わせて述べ、2.5 で従来の仮想チャネルフロー制御法とその問題点を述べる。

2.2 ワームホールルーティング

多くの並列計算機でメッセージ通信をする時には、クロックに同期して 1 フリットごとに転送を行なってゆく。フリットとは 1 クロックの転送で送ることのできる単位で、8～64bits 程度の大きさである。ノード間のパケット転送方式は、ワームホールルーティング、ストアアンドフォワード、バーチャルカットスルーの 3 つに大別される。

ストアアンドフォワードとは、図2.1で示されるように、各ノードにパケット全体を保持できるバッファを持ち、各ノードがパケット全体を受け取ってから次のノードへ送信する方式である。バーチャルカットスルーはストアアンドフォワードと同様に、パケット全体を保持できるバッファを各ノードに持っているが、パケット全体の到着を待つことなく次のノードに転送できるという方式である。

ワームホールルーティング [4] は、図2.2で示されるように、パケットをより小さな単位のリットに分割し、リットをパイプライン状に転送する手法で、並列計算機上でメッセージを転送する際の手法として広く用いられている。ワームホールルーティングには、以下のような特長がある。

- パケット全体を保存するためのバッファが必要ないため、ストアアンドフォワードやバーチャルカットスルーに比べて少ないバッファサイズで実現できる
- パケット長が大きい場合、転送時間がネットワーク距離に依存しにくくなる

上記の特徴から、並列計算機のメッセージ送信には従来のストアアンドフォワード方式に代わって頻繁に使用されている。

以下、ストアアンドフォワードおよびワームホールルーティングの転送時間に関して議論する。 T_c をクロックのサイクルタイム、 D をネットワーク距離、 L をパケット長、 W をバンド幅とすると、従来のストアアンドフォワードの転送時間 T_{SF} は次式で示される。

$$T_{SF} = T_c \left(D \times \frac{L}{W} \right) \quad (2.1)$$

これに対し、ワームホールルーティングの転送時間 T_{WH} は次式で示される。

$$T_{WH} = T_c \left(D + \frac{L}{W} \right) \quad (2.2)$$

ここで、 L_h は、ヘッダ長である。

このようにワームホールルーティングでは、パケット長が長くなるに従って転送時間がネットワーク上の距離に依存しにくくなる。

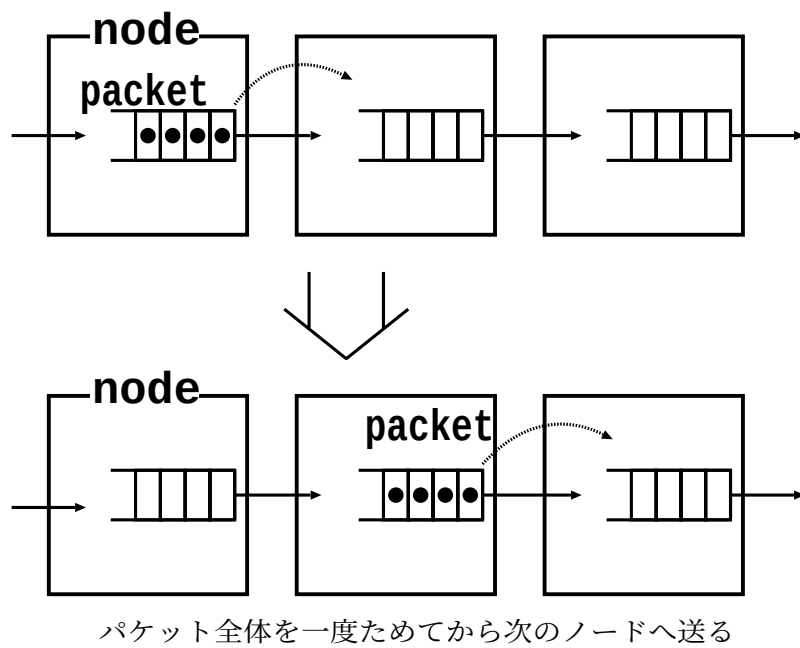


図 2.1: ストアアンドフォワード

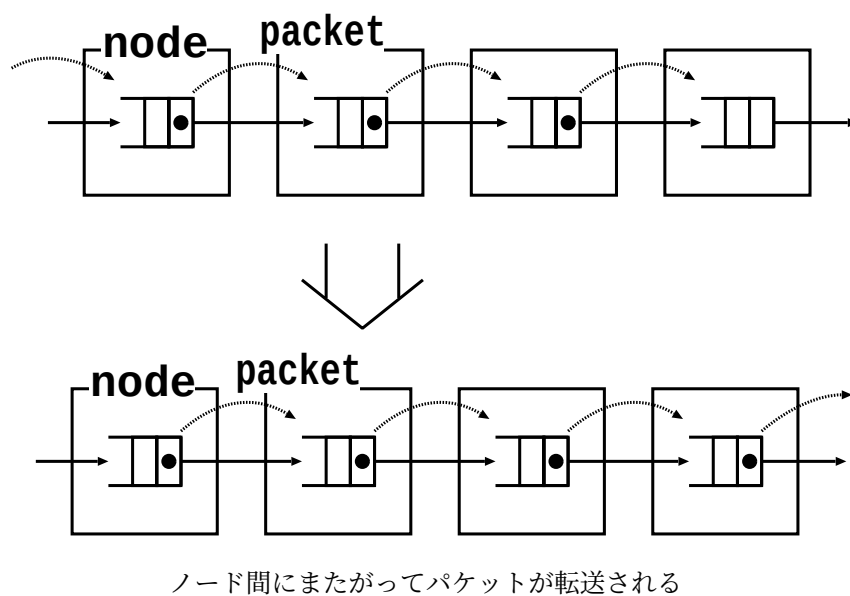


図 2.2: ワームホールルーティング

2.3 仮想チャネル

ワームホールルーティングの欠点として、パケット同士の衝突によるブロッキングが多いという点があげられる。ワームホールルーティングでは、パケットの先頭がブロックされると、そのパケットが複数のバッファを占有したまま停止してしまうため、図 2.3 のように、停止したパケット A が他のパケット B をブロックしてしまうということが起こりやすい。そこで、図 2.4 のようにバッファを用意して仮想的な経路を設ける。これを仮想チャネルと呼ぶ。このようにすると、パケット B は仮想チャネルによりブロックされることなく先に進むことができる。

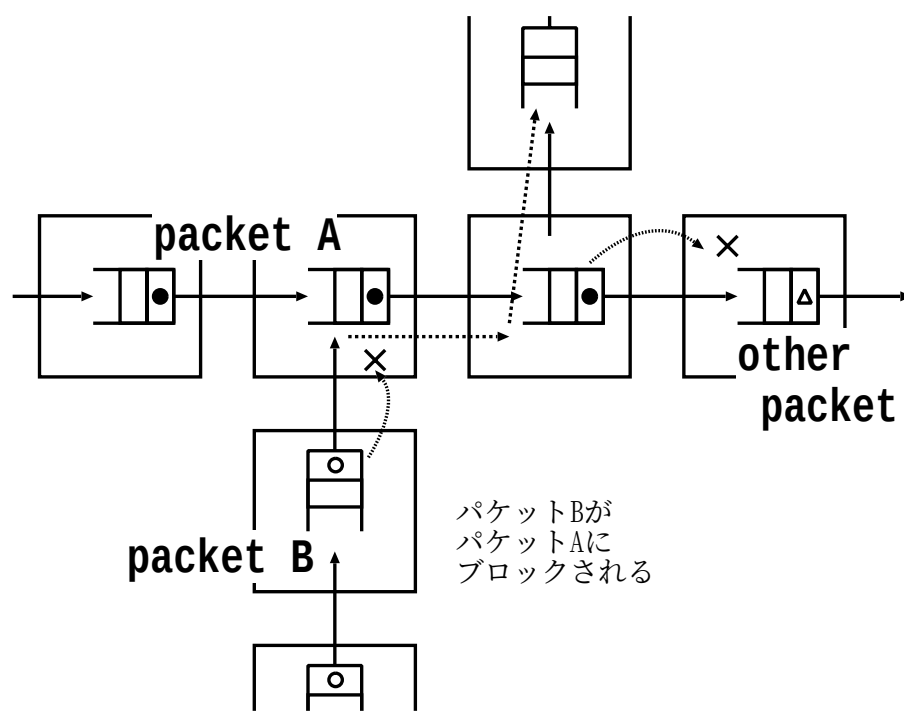


図 2.3: パケットのブロック

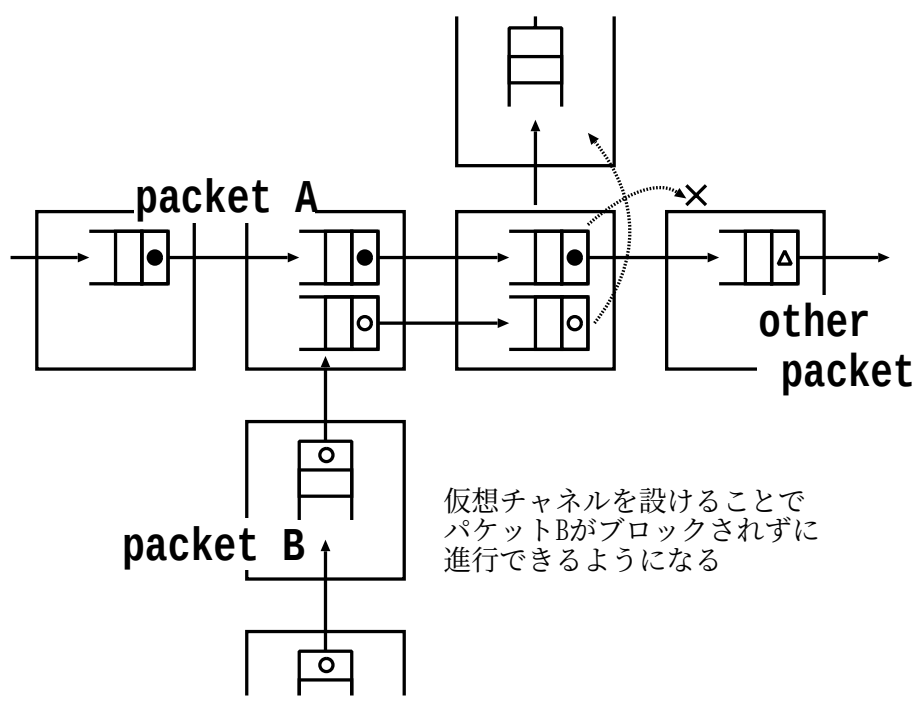


図 2.4: 仮想チャネルの利用によるブロックの回避

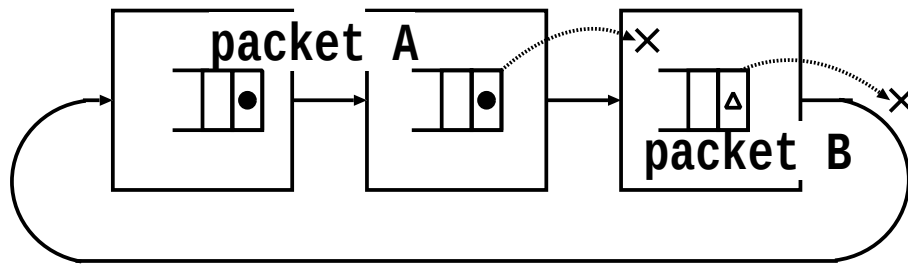


図 2.5: デッドロックの例

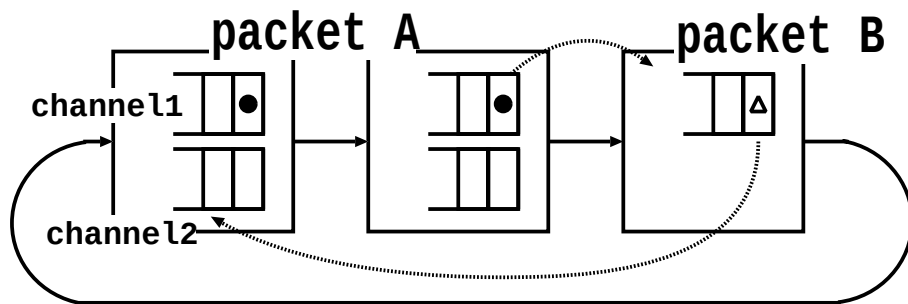


図 2.6: 仮想チャネルによるデッドロックの回避

仮想チャネルは、デッドロック回避のためにも用いられる。デッドロックは、例えば図 2.5 のようにパケット同士がブロックし合う時に発生する。特に、ブロックの頻度が多いワームホールルーティングでは頻繁にデッドロックが発生する。そこで、性能の低下を防ぐためにデッドロックが発生しないルーティング法を選ぶ必要がある。デッドロックを起こさないためには、パケットを格納するバッファが巡回しないようにすれば良い。そのために、turn モデルを考えて循環を防ぐ方法 [14] などがあるが、他にも図 2.6 のように、一つの物理リンクに複数のチャネルを設け、巡回が生じないようにチャネルを選択するというルールを設けて転送を行なうという方法がある。後者の方法の例としては、転送が行われるごとに次元数を上げてゆくことでチャネルの循環を防ぐ構造化バッファ法 [7]、メッシュやトーラスなどにおいて、PE 番号が増加する方向に向かうパケットと減少するパケットで別のチャネルを使用する double Y-Channel ルーティング [8] などが挙げられる。

仮想チャネルは、原理的にはワームホールルーティング、ストアアンドフォワード、バーナカルカットスルーのいずれにも用いることができるが、ワームホールルーティングに用いるのが最も効果的である。それは次のような理由による。第一に、ワームホールルーティ

ングは、必要となるバッファサイズが小さく、仮想チャネルを付加するために必要なコストが少なく済むことである。ストアアンドフォワードやバーチャルカットスルーで仮想チャネルを付加するためには、最大パケット長と同じ数のバッファを用意しなければならない。一方、ワームホールルーティングでは、パケット全体を格納するバッファを必要としない。したがって、他の2つの方法よりも少ないコスト消費で仮想チャネルを付加することができる。

第二に、ワームホールルーティングは、仮想チャネルを付加することによる性能の向上が大きいことが挙げられる。前述のように、ワームホールルーティングでは一つのノードでパケット全体を保持することができないためブロッキングが起こりやすい。その分、仮想チャネルを利用したブロック回避の機会も多くなり、性能向上が期待できるからである。

2.4 仮想チャネルのフロー制御

仮想チャネルの実装例を図 2.7 に示す。実装のためには、チャネルの数と同数のバッファとハンドシェイク線、リンクを共有するためのマルチプレクサを必要とする。

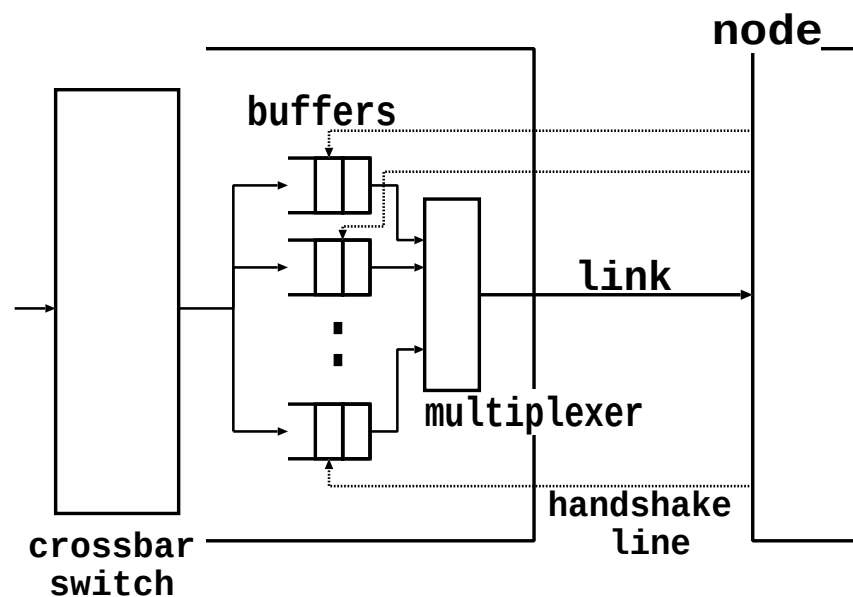


図 2.7: 仮想チャネルの物理的構成

図 2.7 に示したように、仮想チャネルは実際には一つしか持たないリンクを複数のチャネルで共有して、あたかもリンクが複数あるかのように振舞う。この時、どのようなルー

ルに従って仮想チャネルを物理リンクに割り当てるかによってパケットの流れが大きく異なり、システム全体の性能に大きく影響する。物理リンクの割り当てに関しては、リアルタイム処理を目的としたシステムを中心に研究が行われている。

各メッセージに優先順位を設けて仮想チャネルのフロー制御を行なうという手法を最初に提案したのは、Mutka[15]である。以来、リアルタイム性を求めたフロー制御に関する研究が盛んに行われている。[9][10][11]。

しかし、リアルタイム性を必要としないシステムにおいては、フロー制御に関してはあまり考慮されていない。例を挙げると、Song ら [10] の研究では通信の種類をリアルタイム通信と、非リアルタイム通信に分け、非リアルタイム通信はラウンドロビンでフロー制御を行っている。また、並列計算機 JUIMP-1[12] でも、仮想チャネルのフロー制御にラウンドロビンを採用している。

2.5 従来の仮想チャネルフロー制御法

従来の仮想チャネルフロー制御法であるラウンドロビンは、パケットを1フリットずつ異なる仮想チャネルから選択して、物理リンクを通過させてゆく手法である。図2.8のように2つのチャネルに1つずつパケットがある場合、最初に片方のチャネル（この場合はチャネル1）から1フリット通過させる。次に、もう片方のチャネル（この場合チャネル2）から1フリット通過させるといった順に制御を行う。

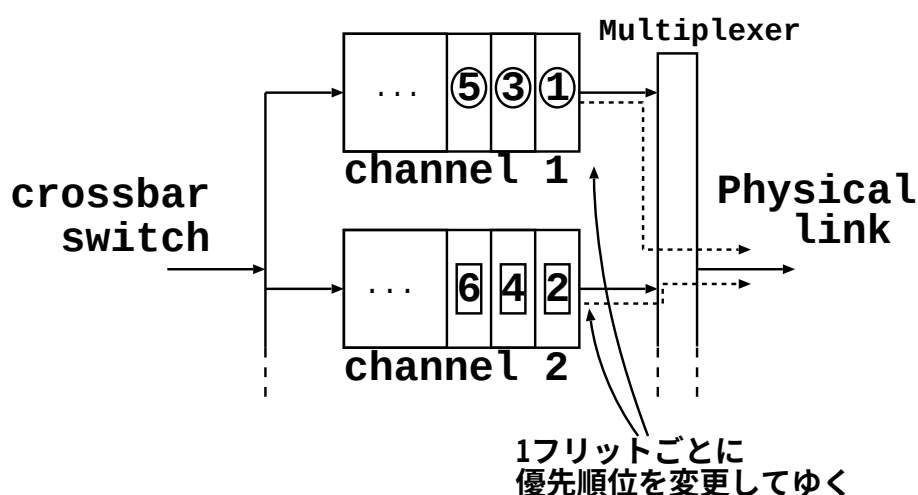


図 2.8: ラウンドロビンのフロー制御

ラウンドロビンのような単純な方法によって仮想チャネルフロー制御を行うことには、二つの問題が考えられる。第一に、ラウンドロビンという手法そのものがワームホールルーティングに適したものではない可能性がある。

第二に、階層型相互結合網のような不均一なネットワークで転送が行なわれる場合、ネットワークの特性を考慮する必要があるという点が挙げられる。例えば、低階層の中での通信と高階層での通信では、異なるポリシーに基づいて異なる優先順位で制御することにより、性能の向上が得られる可能性がある。

2.6 まとめ

本章では、ワームホールルーティングと仮想チャネルについて、その目的および特徴を説明した。さらに、仮想チャネルのフロー制御について説明した。最後に従来型の仮想チャネルフロー制御を説明し、その問題として、ワームホールルーティングの特性を考慮した仮想チャネルフロー制御についてはあまり検討されていないという点、不均一な転送が行なわれるケースについて検討されていないという点の二点を挙げた。

次章では、第一の課題であるワームホールルーティングの特性を考慮した仮想チャネルフロー制御について、具体的にどのような問題が発生するかについて説明し、その解決法を示す。

第 3 章

優先順位に基づく仮想チャネルフロー制御法

3.1 はじめに

本章では、優先順位に基づいた仮想チャネルフロー制御法である占有法を提案する。2章では、従来手法の問題点として、以下の二点が考えられることを述べた。

- 手法そのものがワームホールルーティングに適したものではない可能性がある。
- 不均一なネットワークで転送が行なわれる場合、ネットワークの特性を考慮する必要がある

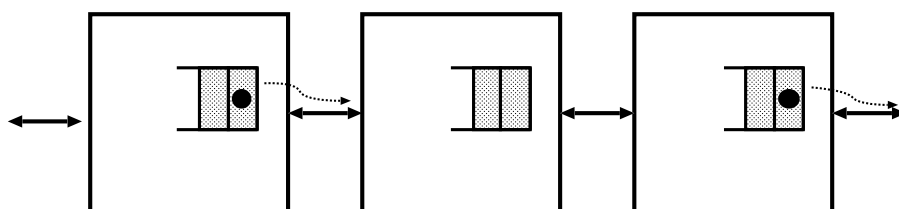
うち本章では、前者について考察を進める。まず 3.2 で、ワームホールルーティングにおいて従来のラウンドロビンで制御を行なうと効率が悪化する理由について述べる。次に 3.3 で、その問題点を解決するための手法を提案し、その手法によって問題点が解消される理由を述べる。

3.2 従来の仮想チャネルフロー制御の問題点

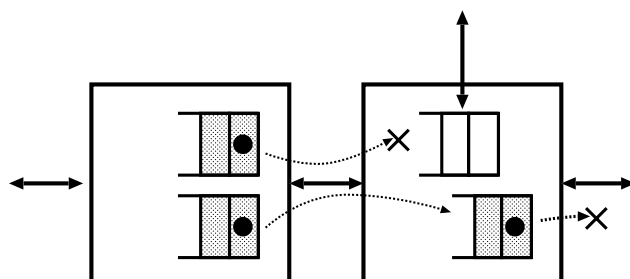
ネットワーク全体の転送効率が低下する原因として、図 3.1 に示される二つを挙げることができる。一つ目は、バッファは予約されているがフリットが到着していないために転送が行なえず、転送の効率が低下するケースである。このようなケースでは図 3.1(1) のように、予約されているが中にフリットがないために転送が行えないバッファが増えるため、転送の効率が低下する。

二つ目は、一つのリンクに複数のフリットを格納できる時に発生するケースである。図 3.1(2) のように、前方のフリットが仮想チャネルのアービトレーションなどの理由で停止している時に後続のフリットが到着した場合、後続のフリットは前方のフリットの転送待ちにより停止しなければならないために、無駄な転送となる。

ワームホールルーティングでラウンドロビンにより制御を行うと、これら二つのうち前者が発生する。図 3.2 のようにパケット同士が衝突した場合、ラウンドロビンでは 1 フリットごとに順番にリンクを利用するため、フリットとフリットの間に空白ができやすい。この状態でパケットが先に進むと、パケット同士の衝突の起こらない所では図 3.1(1) のように、フリット間の空きのために転送の効率が悪くなる場合がある。



**(1) フリットの間空きがあるため
転送が行えない**



(2) 無駄な転送をしてしまう

図 3.1: ネットワーク性能の低下の要因

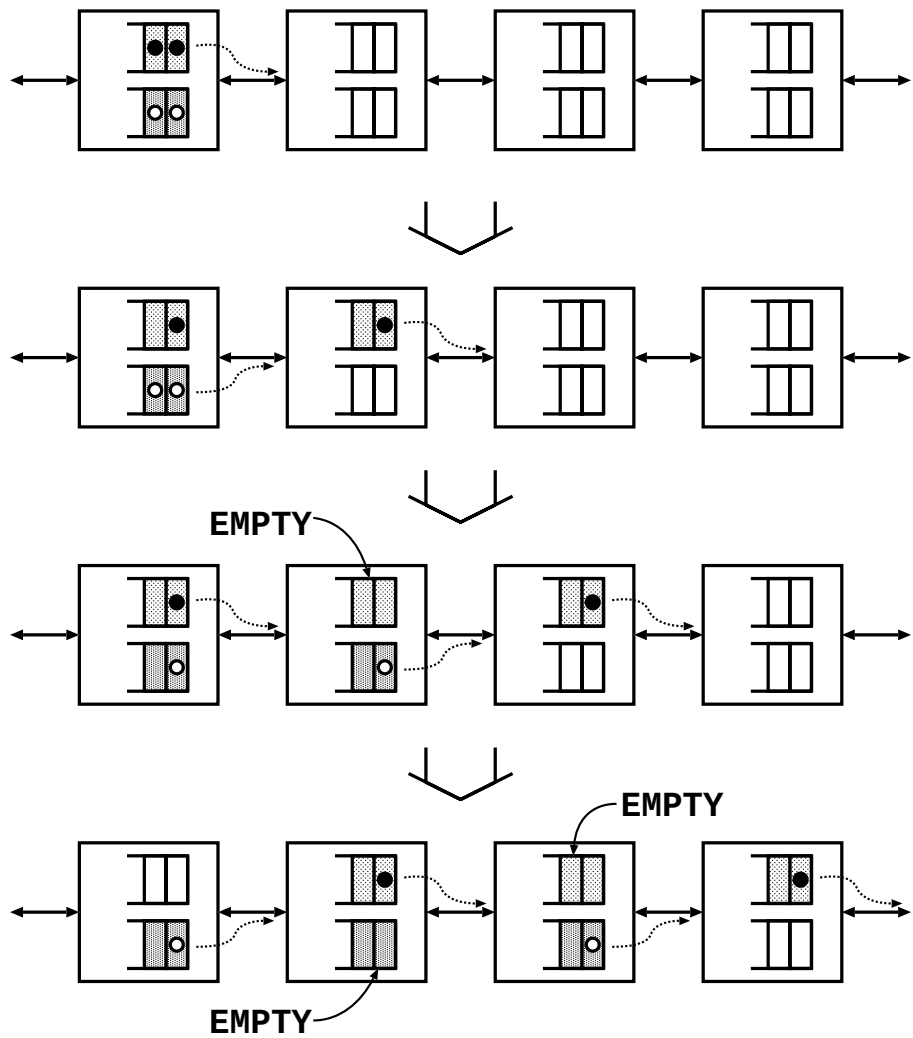


図 3.2: ラウンドロビンにより発生する空白

3.3 優先順位による仮想チャネルフロー制御

3.2で述べた問題点を解決するため、各チャネルに優先順位を設けて特定のチャネルが優先的にリンクを利用する手法を提案する。本章で提案する占有法は、パケットの到着した時刻の早い順に優先順位を割り当てることにより、一つのパケットができる限り集中的にリンクを利用する手法である。占有法は、以下のルールに従って仮想チャネルのアービトレーションを行なう。

1. 各メッセージはそれぞれ優先順位値を持つ
2. 新たに到着したメッセージの優先順位値を、(現在チャネルに存在するパケット数) + 1 に設定する。
3. 優先順位値の小さい順にチャネルがリンクの使用権を得て、メッセージを通過させる。
4. あるメッセージが通過しきった時、通過しきったメッセージより優先順位の値の大きい各メッセージの優先順位値から 1 差し引く。

占有法のフロー制御を図 3.3 に示す。占有法では、1 つのメッセージ全体が通過するまで優先順位を変更しない。そのため、特定の仮想チャネルがリンクをほぼ占有することになる。もちろん、優先順位の高いパケットがブロッキングにより先に進めない場合、優先順位の低いパケットがリンクを使用することになるので、単独のパケットによる完全な占有状態とはならないが、ほぼそれに近い状態になる。

占有法によりフロー制御を行なうと、二つのパケットが衝突した場合に、図 3.4 のようにフロー制御が行われる。この場合、各パケットに優先順位があるため、片方のパケットがフリットの間に空白を作ることなくリンクを通過することができる。そのため、ラウンドロビンに比べてフリットの間に空白ができにくい。すると、図 3.1(1) のような状態は発生しにくくなり、リンクの利用効率の向上が期待できる。

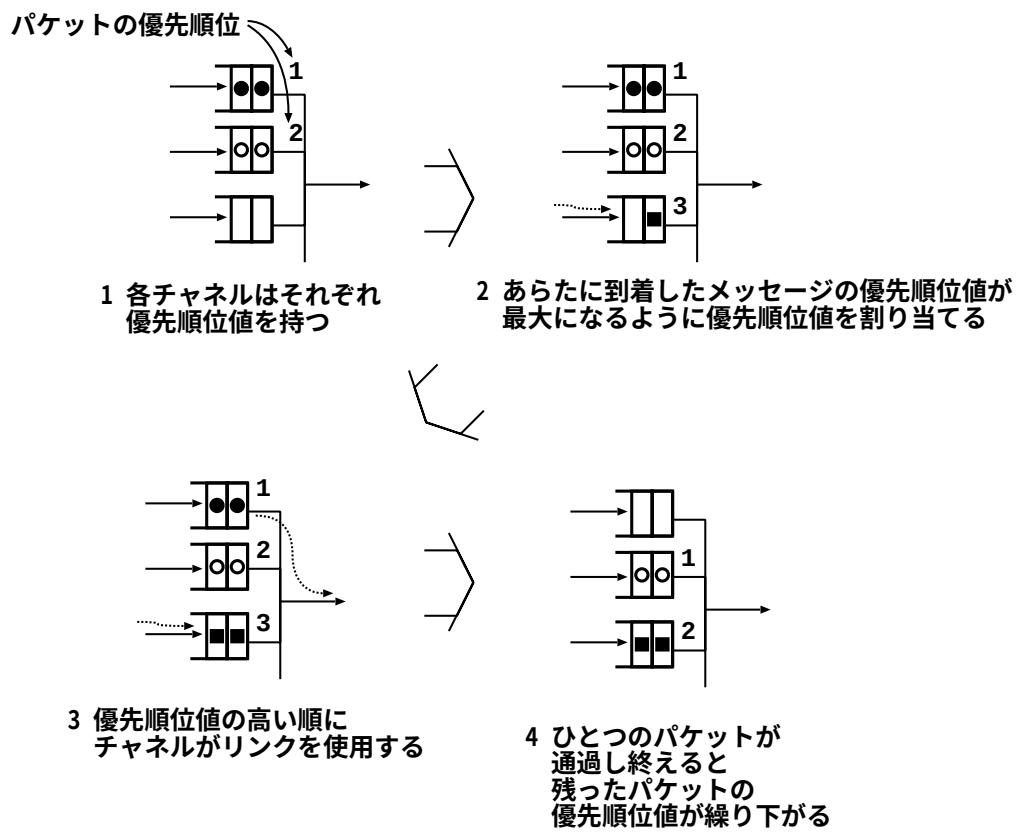


図 3.3: 占有法のフロー制御

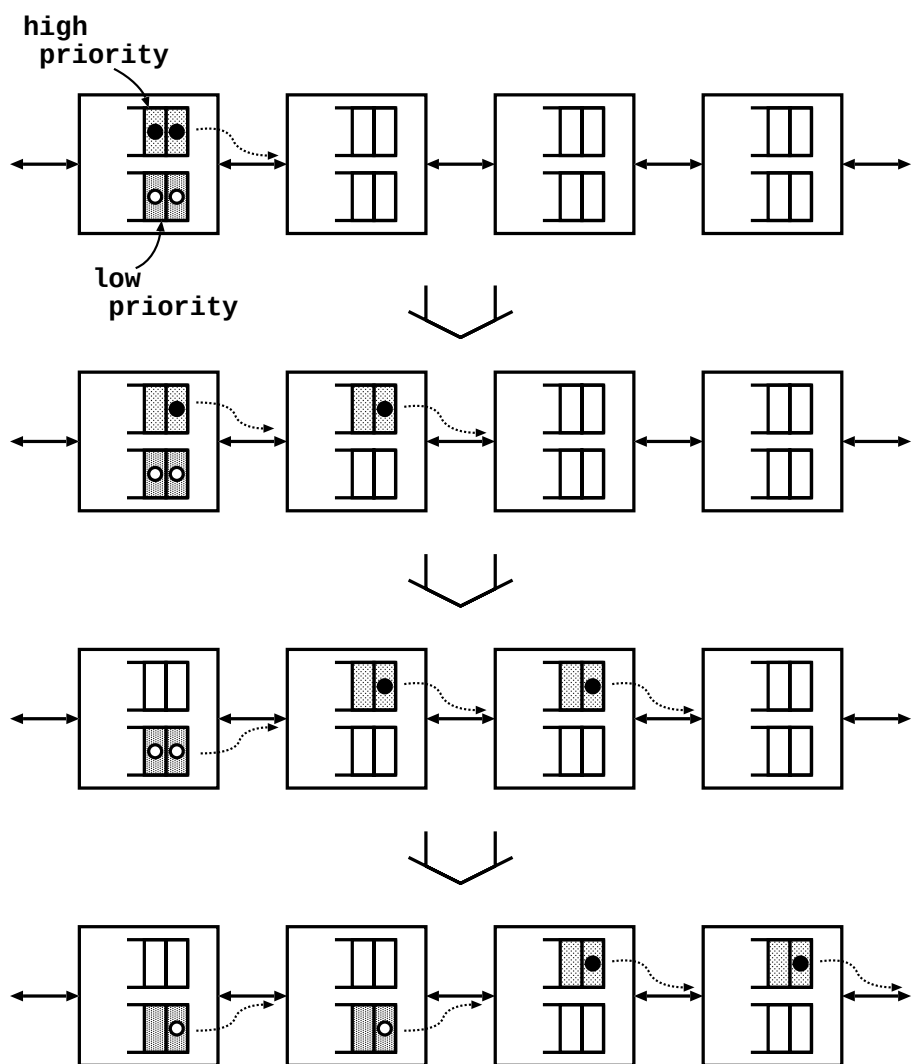


図 3.4: 占有法による空白の抑制

3.4 まとめ

本章では、2章で述べた問題点に基づいて、優先順位に基づいた仮想チャネルフロー制御法である占有法を提案した。まず、ラウンドロビンがワームホールルーティングに適したものでない理由として、フリットの間空白がでやすくなることを指摘した。次に、このような問題点を解決する手法として、できるだけ一つのパケットがリンクを占有して使用する占有法を提案し、占有法の制御法と、占有法が性能向上につながる理由について説明した。

次章では、本手法で実験を行うために想定した並列計算機のアーキテクチャと、そのシミュレータの概要を説明する。

第 4 章

仮想チャネルフロー制御システム

4.1 はじめに

本章では、3章で述べた仮想チャネルフロー制御法の性能を実験により評価するための並列計算機システムの構成について説明する。設計するシステムは通常の計算機システムに比べ、仮想チャネルフロー制御法の違いによる通信性能の違いを評価する目的に特化したシステムとしている。

最初に、4.2でシミュレーションに用いるためのシステムの構成および通信手順を説明し、4.3で、とシミュレータの概要およびシミュレーションに用いたパラメータについて述べる。

4.2 システム構成

4.2.1 ハードウェア

3章で述べた仮想チャネルフロー制御法の性能を実験により評価するための並列計算機システムの設計を行なう。設計の指針としては、ワームホールルーティングで転送する際の仮想チャネルフロー制御の性能を測定することが可能な設計とする必要がある。したがって、パケット転送方式はワームホールルーティングとし、ブロッキングやデッドロックを回避するための仮想チャネルを各リンクごとに設ける。

想定するアーキテクチャを図 4.1に示す。特徴としては以下のものが挙げられる。

- 各 PE は、ノードプロセッサ、ルータ、ネットワークインターフェイスからなる。

- PE 間の通信はルータを介して行なう。
- ネットワークインターフェイスは送信用と受信用があり、一回に一つずつのメッセージの送信と受信を同時に行うことができる。
- ルータの各リンクには、ブロッキング回避のための仮想チャネルを複数設ける。
- メッセージ転送の方法はワームホールルーティングを用いている。

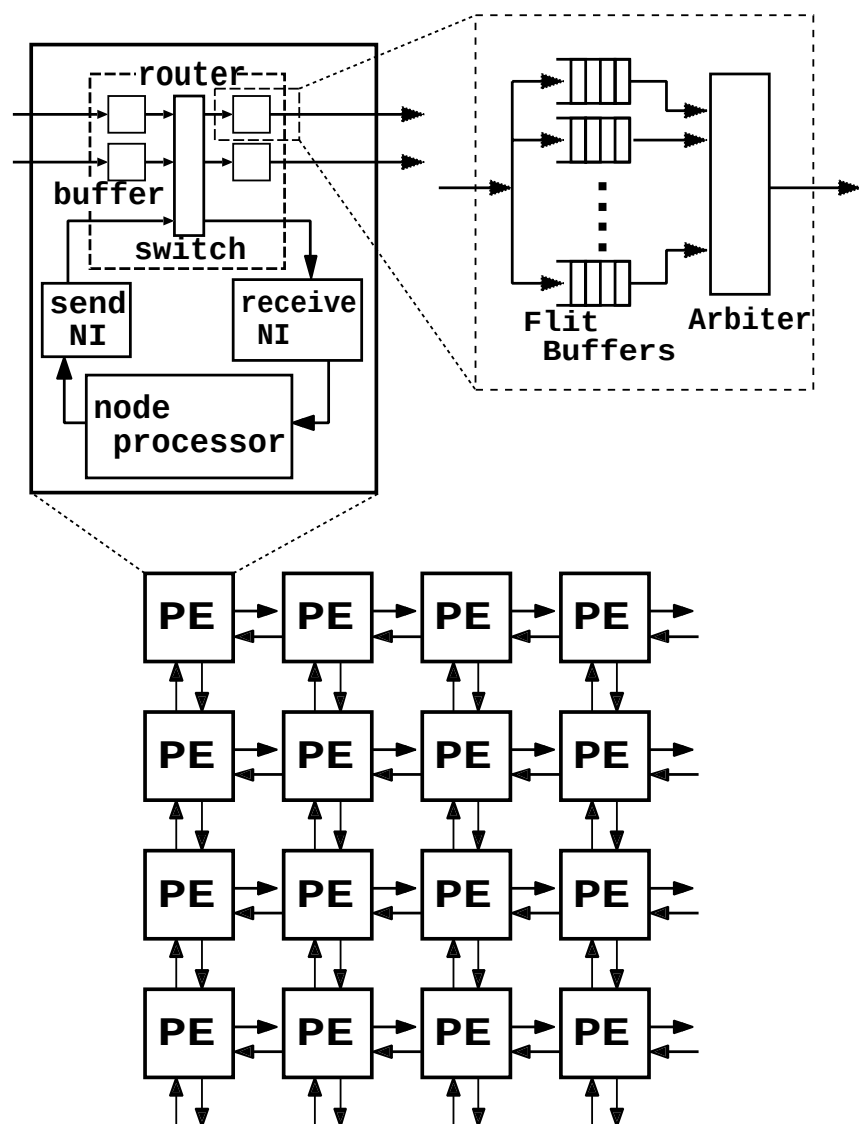


図 4.1: システム構成

以下に各ユニットの役割について述べる。

1. ノードプロセッサ

演算などの命令を実行する通常のプロセッサである。送信要求や受信許可は、このノードプロセッサを介して行われる。

2. 送信用 NI

送信するデータをパケットにする、受信側に転送要求を発するなど、送信のための準備を行う部分である。これらの処理はノードプロセッサの処理とは別に実行できる。

3. 受信用 NI

ノードプロセッサのデータ受信要求に従い、送信用 NI からデータを受け取り、受け取ったデータを格納する部分。

4. ルータ

本システムの中でも、仮想チャネルフロー制御の性能比較を行うための重要な部分である。ワームホールルーティングにより実際にパケットの転送を行っている。ルータの各リンクは、仮想チャネルを複数設けた構造になっている。

4.2.2 通信の手順

ノード間の通信は、多くの場合

1. 送信側から受信側へ転送要求
2. 送信バッファ獲得
3. 受信バッファ獲得要求
4. 受信バッファ獲得要求受理
5. プロセッサから送信バッファへのメッセージ転送
6. 送信バッファから受信バッファへのメッセージ転送
7. 受信バッファからプロセッサ側へのメッセージ転送
8. 受信バッファ解放

といった手順に従って行なわれる [16]。本システムは、通信性能の評価を行う関係上、これにいくつかの修正を加えた手順に従って通信を行っている。具体的には以下の点に修正を加えている。

- ネットワークが混雑した状態での通信性能についての評価を可能とするために、受信バッファは複数の送信バッファからの送信要求を受け付けることも可能な構成とする。そのため、送信バッファ獲得後に受信バッファの獲得は行わず、受信バッファは転送要求の受理のみを行う。
- 受信側のノードプロセッサは、必要な時に必要なデータを受け取る必要がある。そのため受信側のノードプロセッサが受信用 NI ヘデータ受取り要求を発して、データ受け取り要求がある時に受信用 NI からノードプロセッサへデータが引き渡されるという手順を加える。

通信の手順を図 4.2 に示す。まず、送信側のノードプロセッサから送信用 NI に対してメッセージ転送要求を出す。次に送信用 NI は、転送するデータを格納するためのバッファを確保し、受信側に対して転送要求を出す。受信側では、受信用 NI が受信可能であることを確認して送信側に受理信号を送る。その後に送信側と受信側の間で、ルータを介した転送が行なわれる。

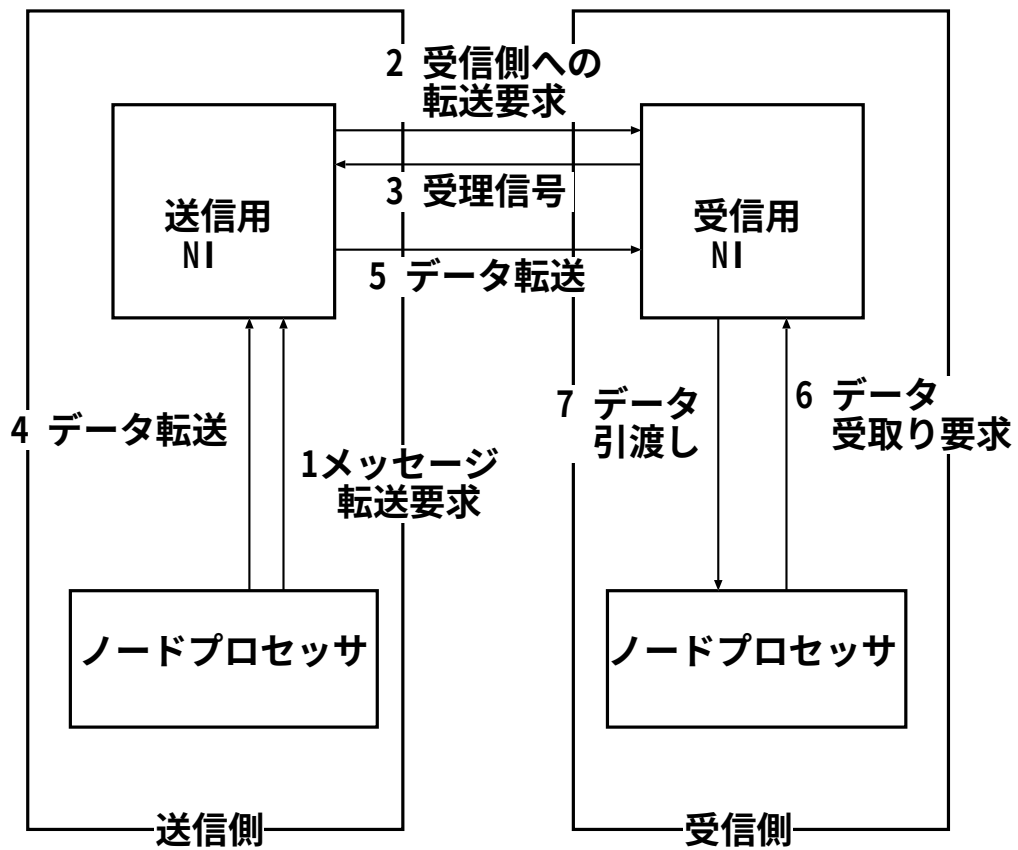


図 4.2: 通信の手順

4.3 シミュレータの構成

前節のハードウェア構成に基づいてソフトウェアによるシミュレータを構築した。以下に、本シミュレータにおいて各ユニットに与えられた機能について説明する。今回用いるシミュレータは、通信性能の評価に特化するため、4.2で述べた設計といくつか異なる点がある。

1. ノードプロセッサ

「送信要求」「データ受取り要求」「命令の実行」の各機能を持つ。演算の種類等の違いは通信性能を評価する上で本質的ではないため、すべて「命令の実行」という形でひとくくりにしている。

2. 送信用 NI

ノードプロセッサの送信要求を受けてパケットを生成し、送信時間と送信元 PE に関する情報と共に転送処理を行う。「送信バッファ獲得」は、パケットの転送にもノードプロセッサにおける命令の実行にも影響しないため省略されている。また、単純化のためノードプロセッサから送信用 NI へのデータ転送は省略している。

3. 受信用 NI

受け取ったパケットから、送信時間と送信元に関する情報を獲得して、転送時間を記録する。なお、「受信バッファ獲得」は、シミュレーションに影響しないため省略されている。

4. ルータ

ヘッダの情報を基に、実際にワームホールルーティングによりパケットの転送を行う。

なお、シミュレーションに用いる各パラメータは表 4.1の通りに設定した。

表 4.1: シミュレーションに用いたパラメータ

チャンネル数	4
パケットヘッダのサイズ	6 flits
仮想チャンネル 1 つあたりのバッファサイズ	1flit
転送要求を出すために要する時間	6 cycles
送信バッファのセットアップ時間	9 cycles
受信側への転送要求に要する時間	1 cycles
ルータの受取り側バッファの転送時間	1 cycle/flit
ルータの送り側バッファの転送時間	1 cycle/flit

4.4 まとめ

本章では、仮想チャンネルフロー制御法の性能を評価するための並列計算機システムの構成について、その目的、ハードウェア構成、各ユニットの機能および通信手順を説明し、シミュレータにおいて各ユニットに与えられた機能およびシミュレーションに用いたパラメータについて述べた。

次章では、二次元格子網上における各手法の性能比較を行なう。

第 5 章

二次元格子結合網における性能評価

5.1 はじめに

格子結合網にはメッシュ網とトーラス網があり、シミュレーションや方程式の解法などの分野で重要な役割を持つ結合網である。本章では、それらのうち二次元メッシュにおける各手法の性能比較を行う。性能比較は、4章で述べたシミュレータを用いて行い、ランダム通信および FFT の性能を評価する。まずはじめに 5.2 において、二次元格子結合網の種類について説明し、5.3 でランダム通信の性能評価を行う。次に 5.4 で FFT を実行した場合の性能評価を行い、5.5 で適応ルーティングを行った場合のランダム通信の性能を評価する。

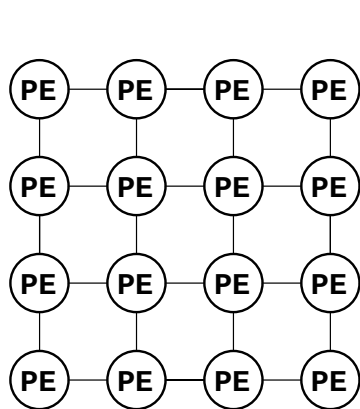
ランダム通信に関しては、平均の転送時間および平均スループットを測定し、評価する。また、FFT の性能は FFT の通信のパターンを模した転送を行ったときの平均転送時間および各 PE の実行時間により評価する。

5.2 二次元格子結合網

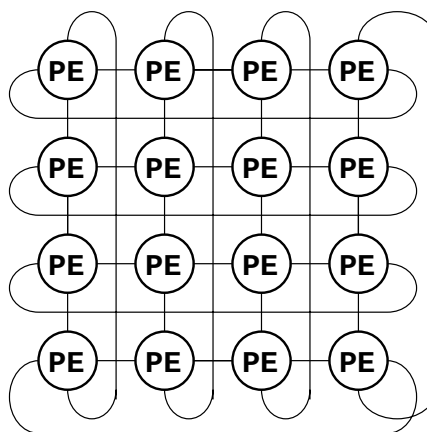
格子結合網は図 5.1 のように PE を格子状に並べた結合網である。図 5.1(a) のように上下左右の端のノード同士を結合しないものをメッシュ網、5.1(b) のように結合したものをトーラス網とよぶ。格子結合網は二次元のものとは限らず、5.1(c) のような三次元以上の結合網も存在する。このようにいくつかの次元を持つ格子結合網のうち、トーラス網のことを、一般にはハイパートーラスもしくは k -ary n -cube とよぶ。

格子結合網は構成が単純であり、シミュレーションや偏微分方程式の解法などの用途に向いているといった特長があるため、並列計算では重要な結合網である。本章では、これ

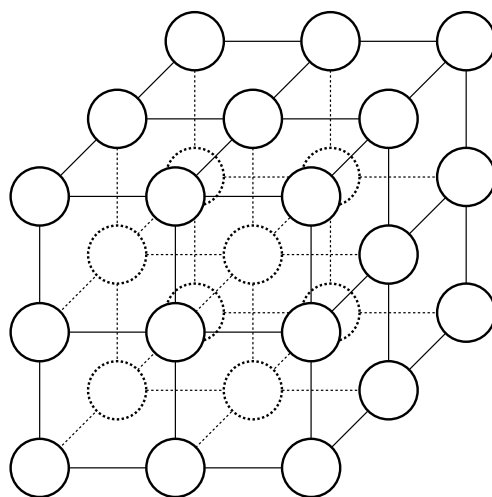
らのうち最も基本的な格子結合網である二次元メッシュ（図 5.1 (a)）を取り扱う。



(a) 二次元メッシュ



(b) 二次元トーラス



(c) 三次元メッシュ

図 5.1: 格子結合網

5.3 ランダム通信による性能評価

5.3.1 実験方法

4章で述べたシミュレータを用いて性能評価を行う。メッシュサイズは図 5.2 に示すように 16×16 で、リンクは同時に双方向の通信を行えるものとしている。仮想チャネル数は 4 とし、一定の長さを持つパケットを、ある確率で各 PE がランダムに送信する。シミュレーションは 20000 サイクルまで実行した。

評価基準は平均転送時間および平均スループットである。平均転送時間は、ノードプロセッサが送信側 NI に対してパケットの送信要求を行った時から、パケットが目的の PE へ到着するまでの時間の平均値である。平均転送時間の単位はサイクルとしている。平均スループットは、シミュレーションを停止するまでの間に目的地に到着するフリット数を、シミュレーションの実行時間で割った値である。平均スループットの単位は flits/cycle となる。

表 5.1 に、実験の種類および条件を示す。通信パターンとして、単純に全ての PE 同士でランダムに通信を行うものと、転送先のみを一部の PE に限定し（本実験では PE0～PE15 に限定している）、ランダムなタイミングで通信を行なうものの 2 種類について行っている。後者の場合、転送先付近ではトラフィックが非常に混雑する。

表 5.1: 実験の種類および条件

	実験の種類	パケット長
(1)	ランダム通信	16,32,64
(2)	転送先が集中した通信	16

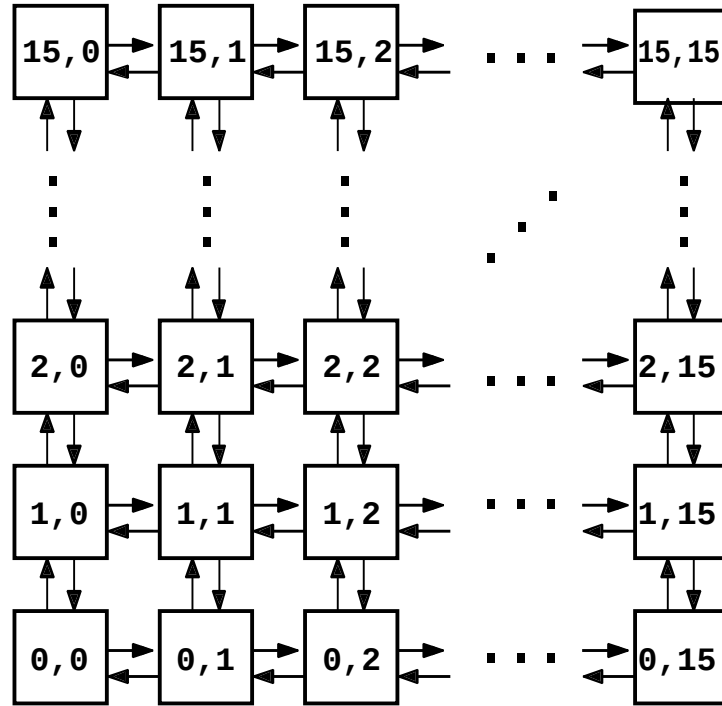


図 5.2: 実験に用いる二次元メッシュ

5.3.2 平均通信時間および平均スループット

ランダム通信におけるラウンドロビンおよび占有法のパケット生成率に対する平均転送時間および平均スループットを図 5.3～5.8 に示す。パケット長をいずれにした場合も、占有法の平均転送時間は短かく、平均スループットは大きくなっている。ただし、パケット長が大きい場合は、小さい場合に比べて性能向上の割合が若干大きい。

占有法は図 3.4 に示すように、フリット間の空きをできにくくすることによりネットワークの効率を上げる手法であるため、パケット長が長い場合のようにフリット間の空きが多くなるケースでは性能の向上率が高くなることが理由と思われる。

転送先が集中した通信の、パケット生成率に対する平均転送時間を図 5.9 に示す。結果が示すように、平均転送時間で大幅な性能向上が得られている。このように経路の途中が混雑するケースでは、フリット間の空白の少ない占有法が、より良い結果を示す。

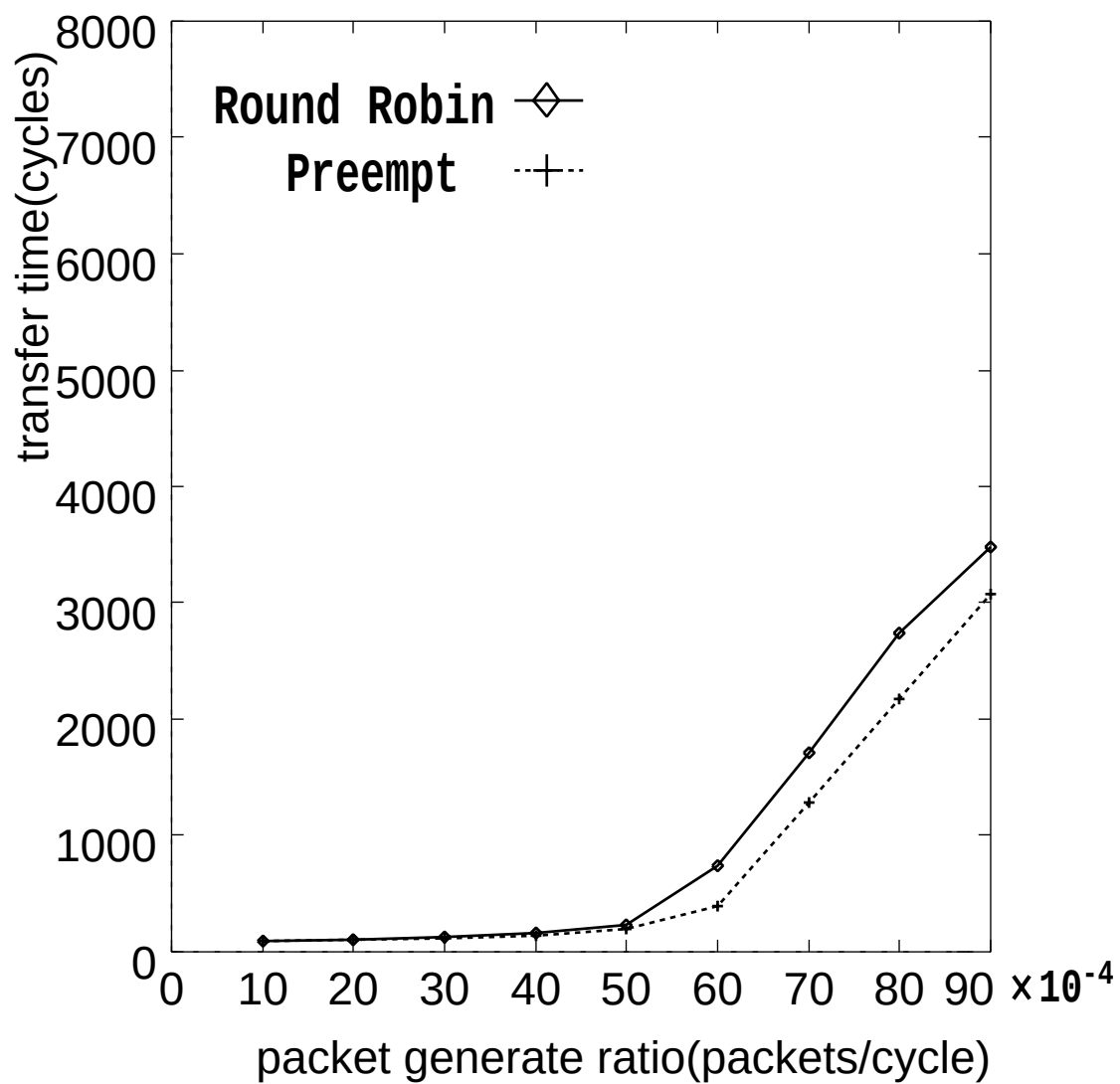


図 5.3: ランダム通信の平均転送時間 (パケット長 16)

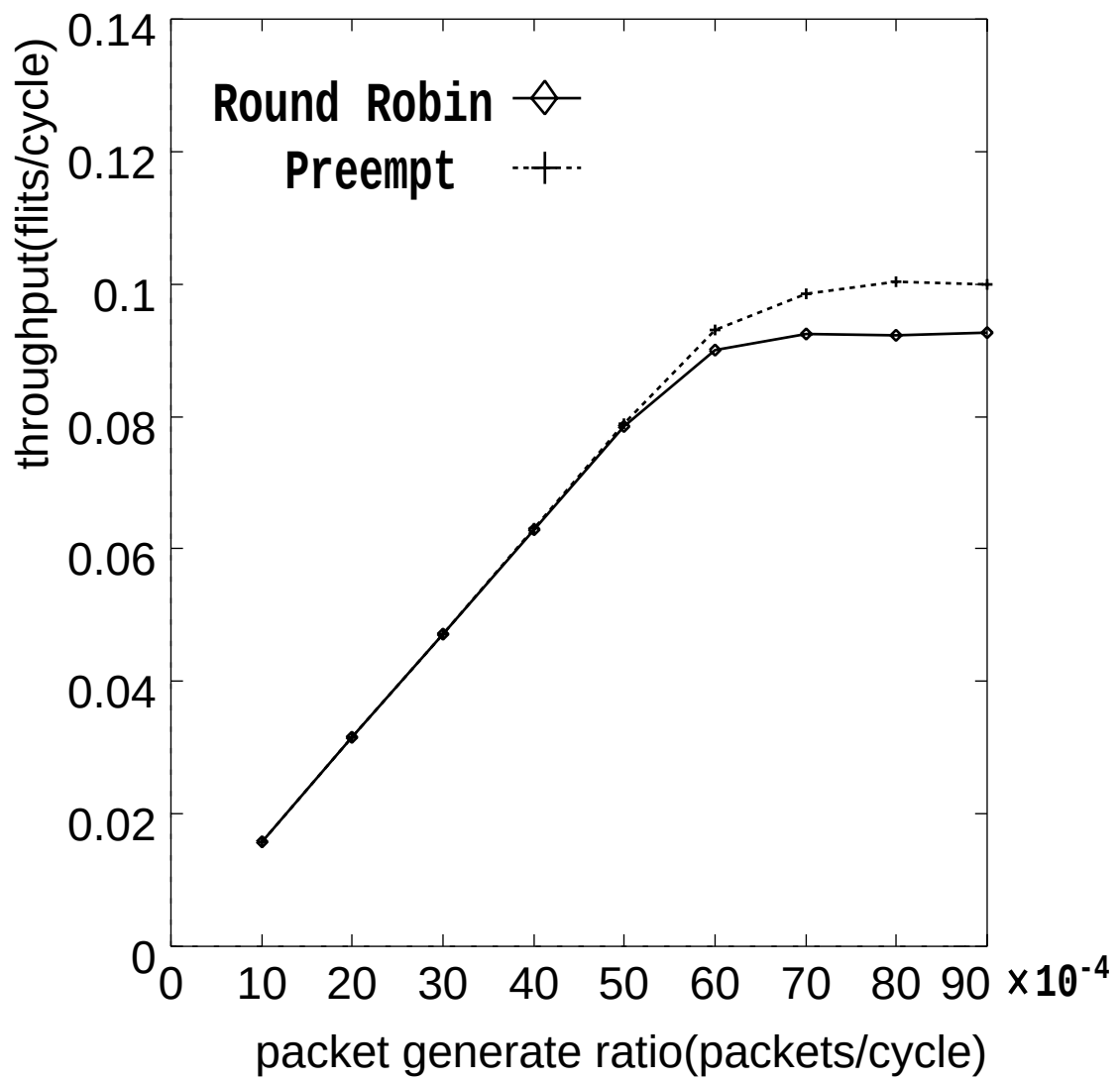


図 5.4: ランダム通信の平均スループット (パケット長 16)

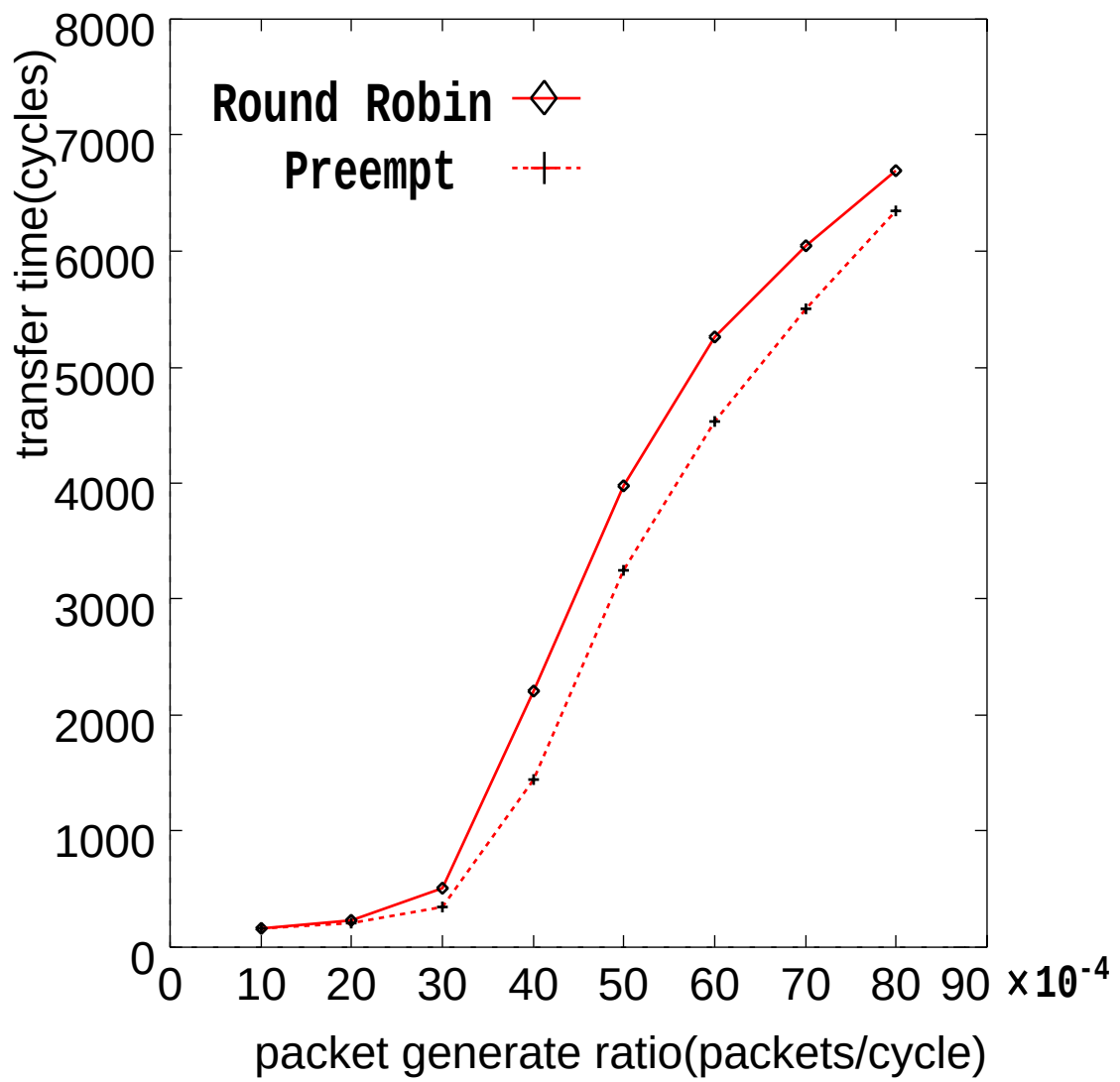


図 5.5: ランダム通信の平均転送時間 (パケット長 32)

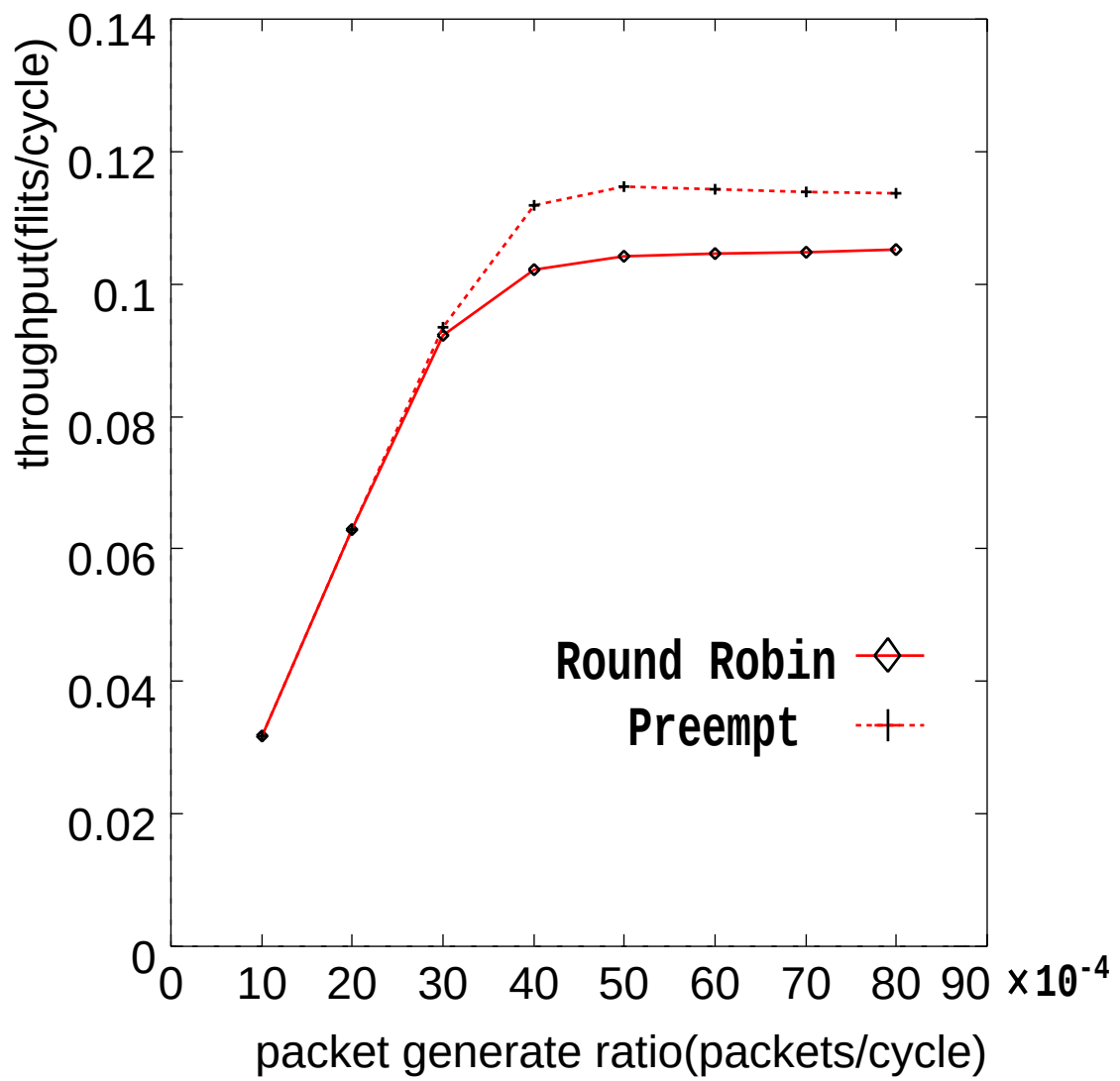


図 5.6: ランダム通信の平均スループット (パケット長 32)

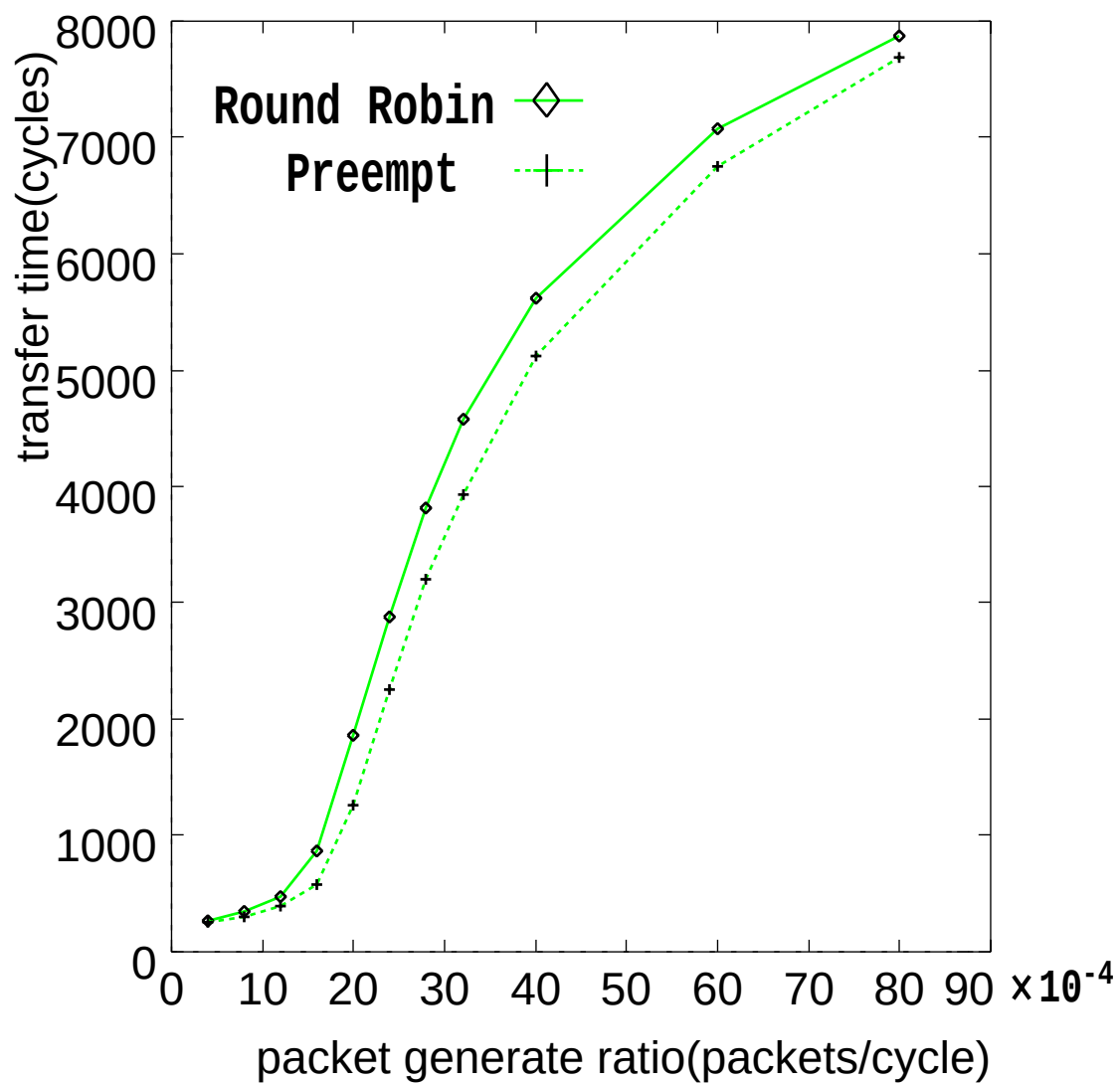


図 5.7: ランダム通信の平均転送時間 (パケット長 64)

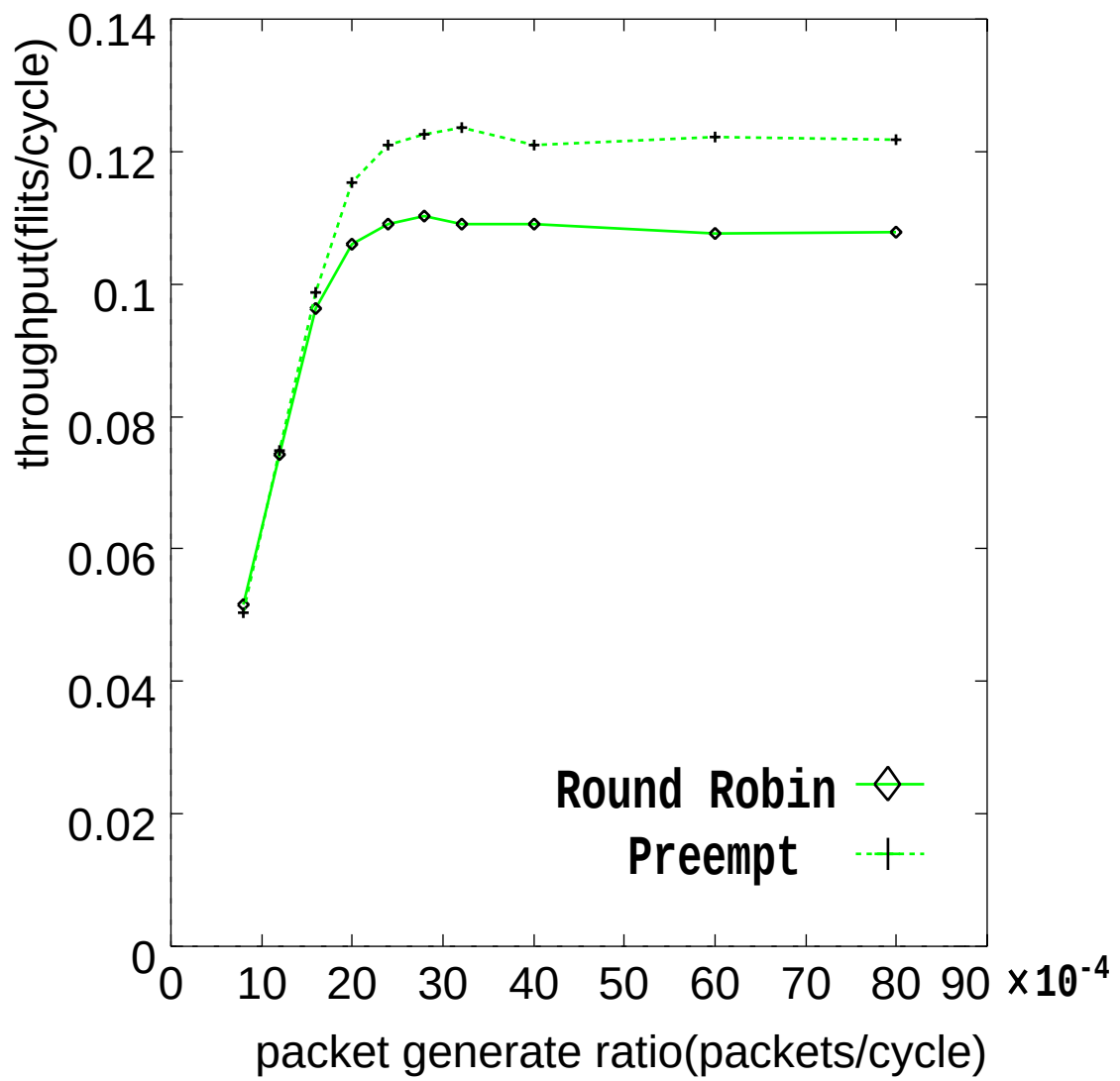


図 5.8: ランダム通信の平均スループット (パケット長 64)

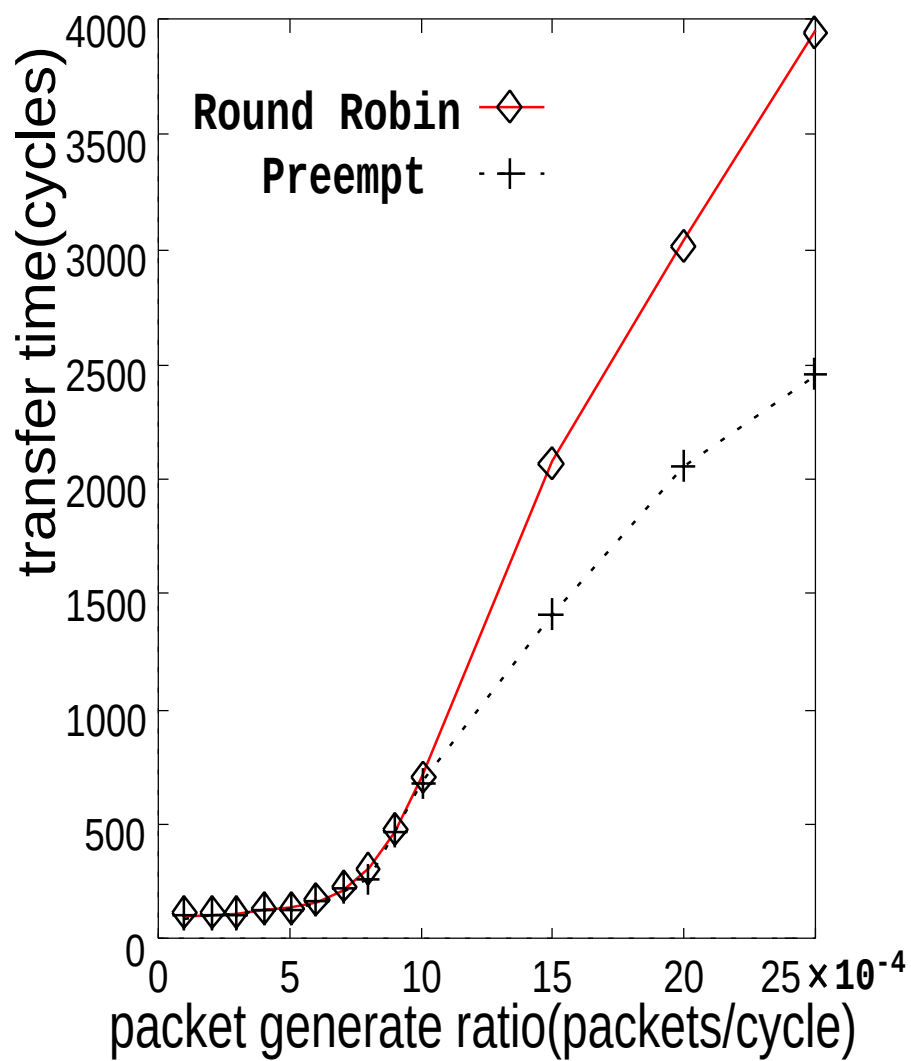


図 5.9: 転送先が集中した通信の平均転送時間

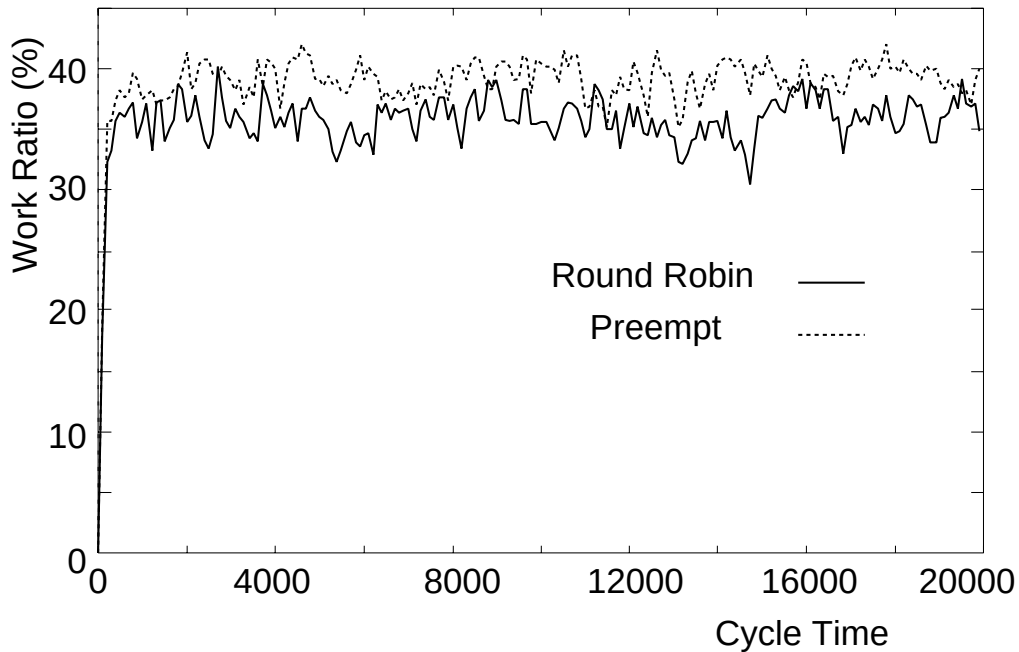


図 5.10: リンクの稼働率

5.3.3 考察

平均スループットにおいて占有法が良い結果を示す原因を考察する。まず、リンクの稼働率（実際に転送が行なわれているリンクの割合）を評価する。パケット長 16 のメッセージを、パケット生成率 8.0×10^{-3} でランダムに転送し、各時間のリンクの稼働率を 20000 サイクルまでプロットした。実験結果を図 5.10 に示す。占有法の平均稼働率は 39.84 % となり、ラウンドロビンの平均稼働率である 36.66 % を上回っている。

このような現象の起きる理由を考察する。あるリンクがフリットを転送できない理由として、以下の点が挙げられる。

1. パケットが存在していないために転送が行なわれない。
2. バッファは予約されている（つまりパケットは存在する）が、フリット間に空白があるためにフリットがバッファに存在せず、転送が行なえない。
3. 入力側のバッファがいっぱいになってしまっているために、フリットがブロックされてしまう。

以上の 3 つの現象を図に表わすと図 5.11 のようになる。ランダム転送中に、これらが各時間に平均していくつのリンクで発生しているかをリンクの稼働率の実験と同じ方法

でプロットした結果を図 5.12～5.14 に示す。また、それぞれの平均値を表 5.2 に示す。なお、リンク数は全部で 960 個である。結果、パケットが存在しないために転送が行えないケースは、二つの手法でほとんど変わらないにもかかわらず、フリット間に空白があるため転送が行えないケースと、前方のパケットにブロックされるケースが占有法において減少していることが分かる。これらの結果から、占有法ではフリット間の空白の発生を抑えることで性能向上を実現していると言える。

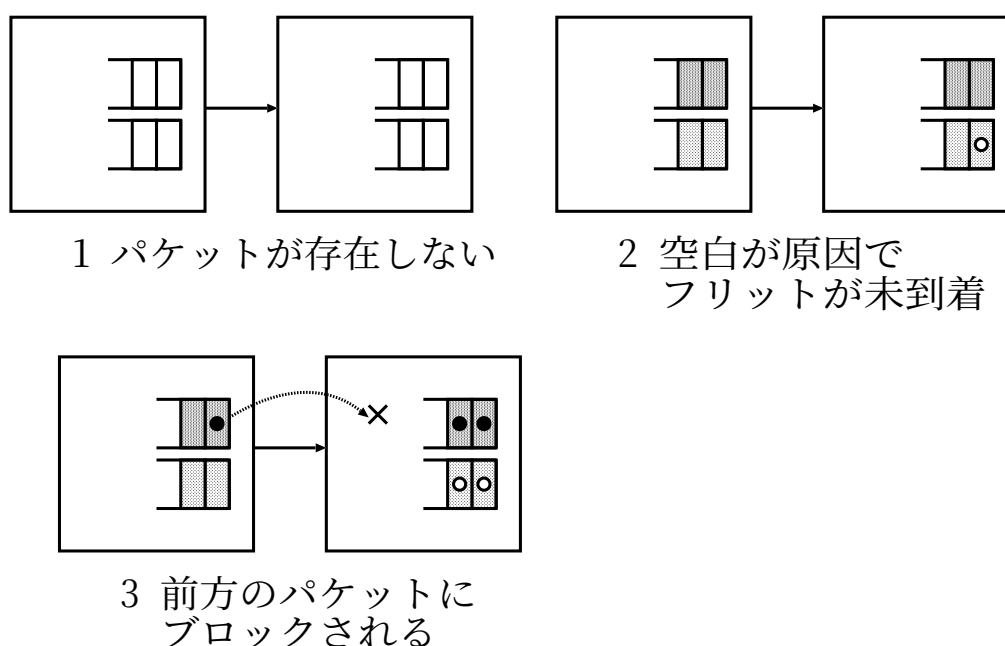


図 5.11: フリットが転送されない原因

表 5.2: 転送が行えないリンク数の平均値

転送が行えない 原因	リンク数の平均	
	ラウンドロビン	占有法
パケットが存在しない	125.28	125.48
フリット間に空白がある	153.27	124.07
前方のパケットにブロックされる	352.67	311.17

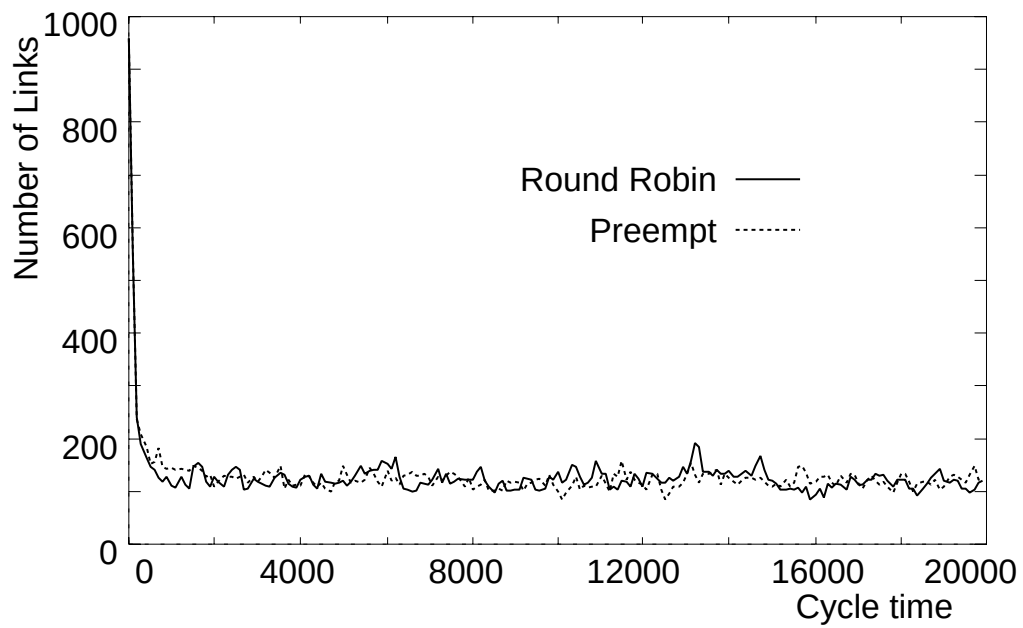


図 5.12: パケットが存在しないために転送を行なわないリンクの数

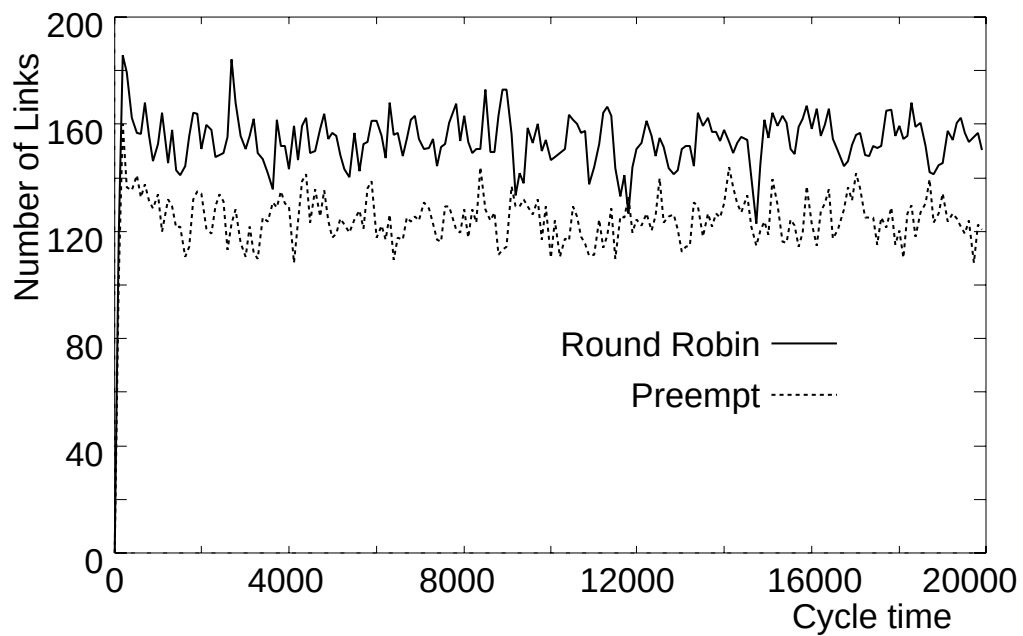


図 5.13: フリット間に空白があるために転送が行えないリンクの数

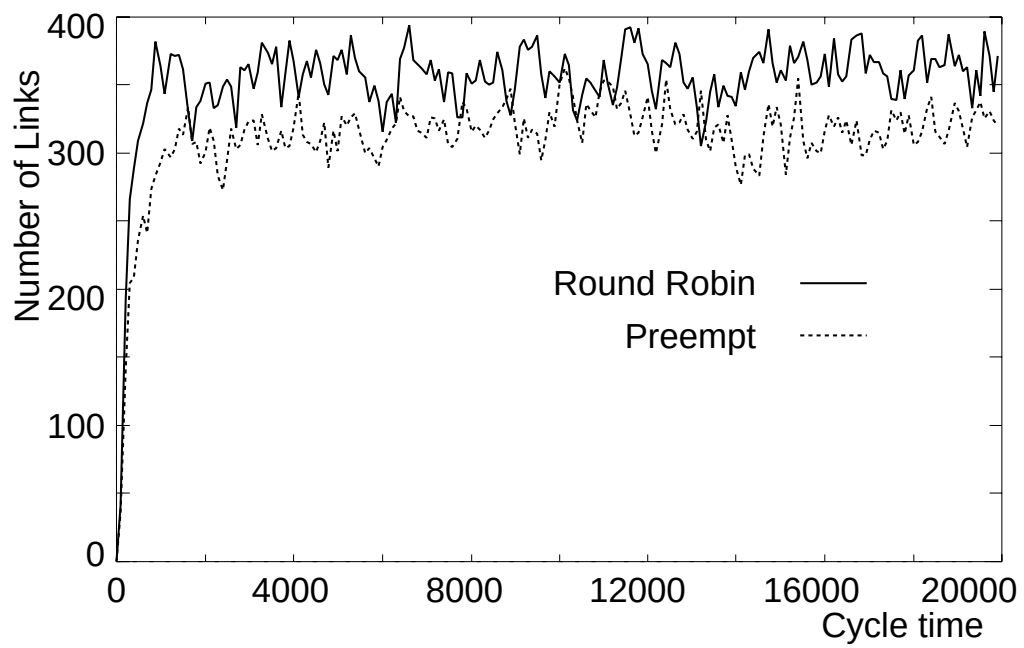


図 5.14: 前方のパケットにブロックされるために転送が行えないリンクの数

5.4 FFT による性能評価

5.4.1 FFT のマッピング

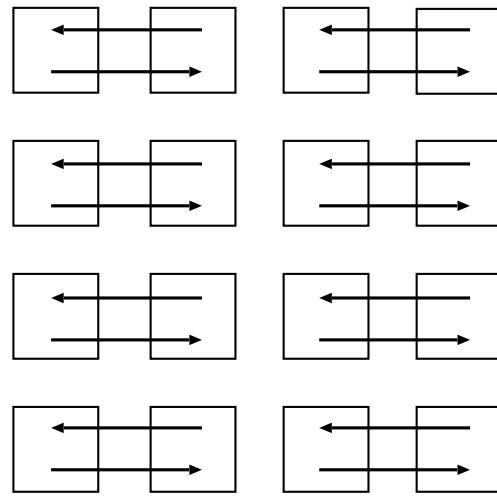
より現実の通信パターンに近い評価を行なうために、シミュレータ上で FFT をトレースして性能を評価する。まず、 256×2^n 個のデータを以下のように二次元格子上にマッピングする。なお、ここではデータ番号および PE 番号を 2 進表記で表現している。

1. データに 0 から 256×2^n までの番号をつける。この時につけた番号を 2 進表記すると、 $b_{n+7} \dots b_3 b_2 b_1 b_0$ と表現される。
2. データ番号 $b_{n+7} \dots b_3 b_2 b_1 b_0$ を、 $PE(b_7 b_6 b_5 b_4, b_3 b_2 b_1 b_0)$ に割り当てる。

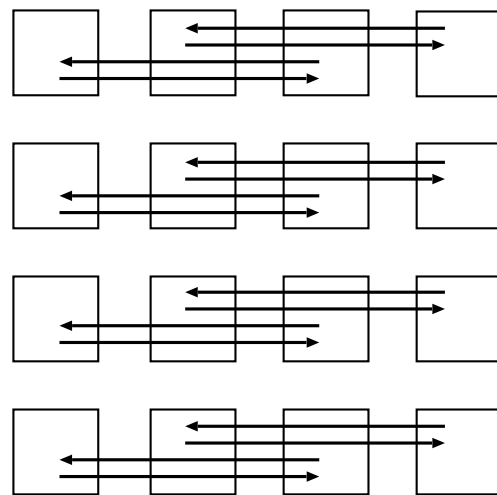
すると、 $PE(0,0)$ にはデータ番号 $256 \times i (i = 0, 1, 2, \dots)$ 、 $PE(0,1)$ にはデータ番号 $256 \times i + 1 (i = 0, 1, 2, \dots)$ 、という具合にマッピングされる。

16×16 メッシュ上にこのようにマッピングした場合、通信は全部で 8 回行なわれる。その際の各 PE の j 回目の転送先は、PE 番号を 2 進表記して $PE(b_7 b_6 b_5 b_4, b_3 b_2 b_1 b_0)$ とあらわした場合、 b_j を反転した番号の PE となる。例えば、一回目の転送先は $PE(b_7 b_6 b_5 b_4, b_3 b_2 b_1 \bar{b}_0)$ 二回目の転送先は $PE(b_7 b_6 b_5 b_4, b_3 b_2 \bar{b}_1 b_0)$ となる。

FFT の、最初の 2 回の通信パターンを図 5.15 に示す。 256×2^n 個のデータを扱う場合、FFT では全部で $n + 8$ ステージのバタフライ演算を行なうことになるが、上のようにマッピングした場合、最初の 8 ステージ以降は同一 PE 上にあるデータ同士でのやり取りになるため、PE 間通信を必要としない。



一回目



二回目

図 5.15: FFT の通信パターン

5.4.2 実験の諸条件

本シミュレーションでは、実験の条件を表 5.3 のように設定している。FFT の各ステージ間では、データ数と同数のバタフライ演算の他に、転送先の計算と通信のセットアップを一回ずつ行なうことになる。

表 5.3 のパラメータを用いると、通信と通信の間の演算のための時間 T は

$$T = T_t + T_s + n \times T_b \quad (5.1)$$

1 メッセージのメッセージ長 L は

$$L = n \times L_d \quad (5.2)$$

となる。

表 5.3: FFT のシミュレーションのための諸条件

条件	記号	値
バタフライ演算一回に要する時間	T_b	220 cycles
通信のためのセットアップ時間	T_s	120 cycles
転送先の計算に要する時間	T_t	120 cycles
データ一つあたりのメッセージ長	L_d	16 flits

5.4.3 通信に要する時間および実行時間の比較

各手法について、一回の通信に要した時間を図 5.16 に、FFT の実行に要した時間を図 5.17 に示す。図中の折れ線は各 PE の実行時間の平均値を示し、縦線の上端は各 PE の実行時間の最大値、下端は最小値を示している。

図 5.16 に示すように、占有法はラウンドロビンに比べて平均通信時間が改善されているが、図 5.17 のように、実行時間はほとんど変化がない。このようなことが起こるのは、占有法では転送時間の最大値が変化せずに平均転送時間を短縮しているためと考えられる。転送時間の長いメッセージが一部だけ存在すると、FFT のように「受信」→「計算」→「送信」のプロセスを繰り返すアプリケーションでは十分な性能向上が得られない。

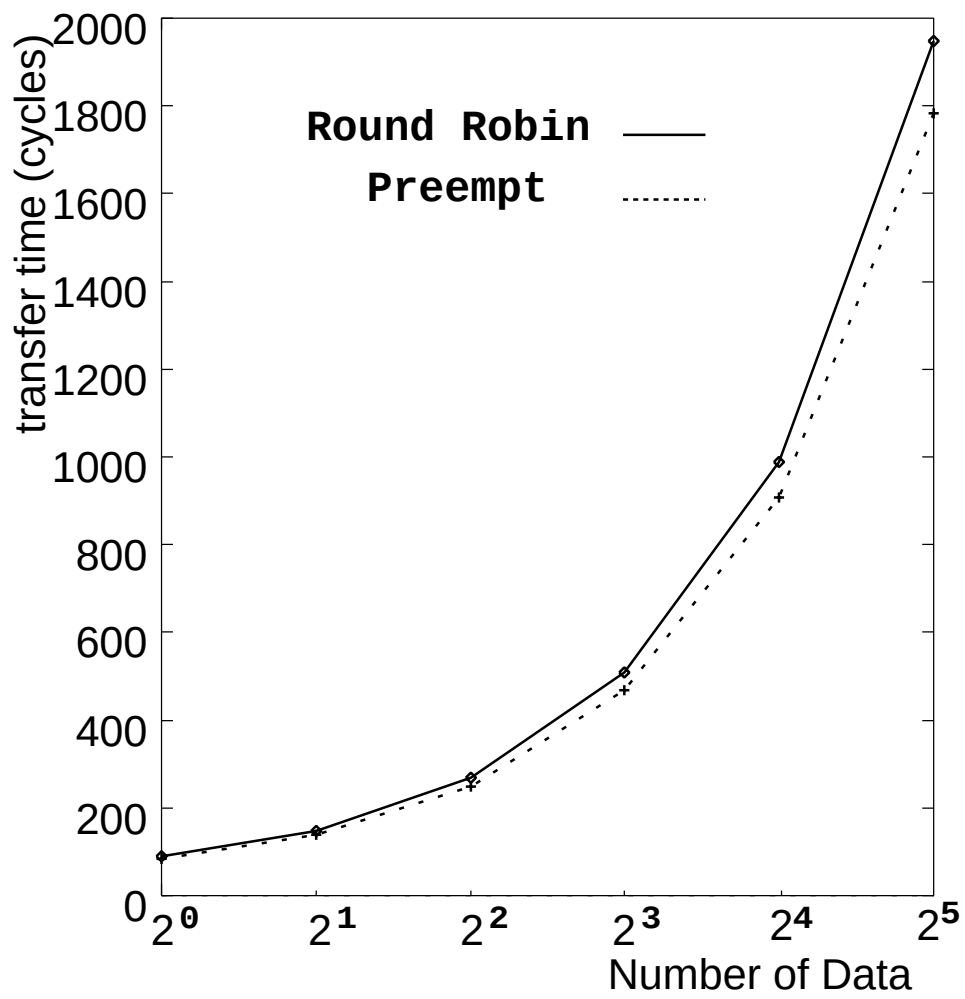


図 5.16: FFT における各手法の平均通信時間

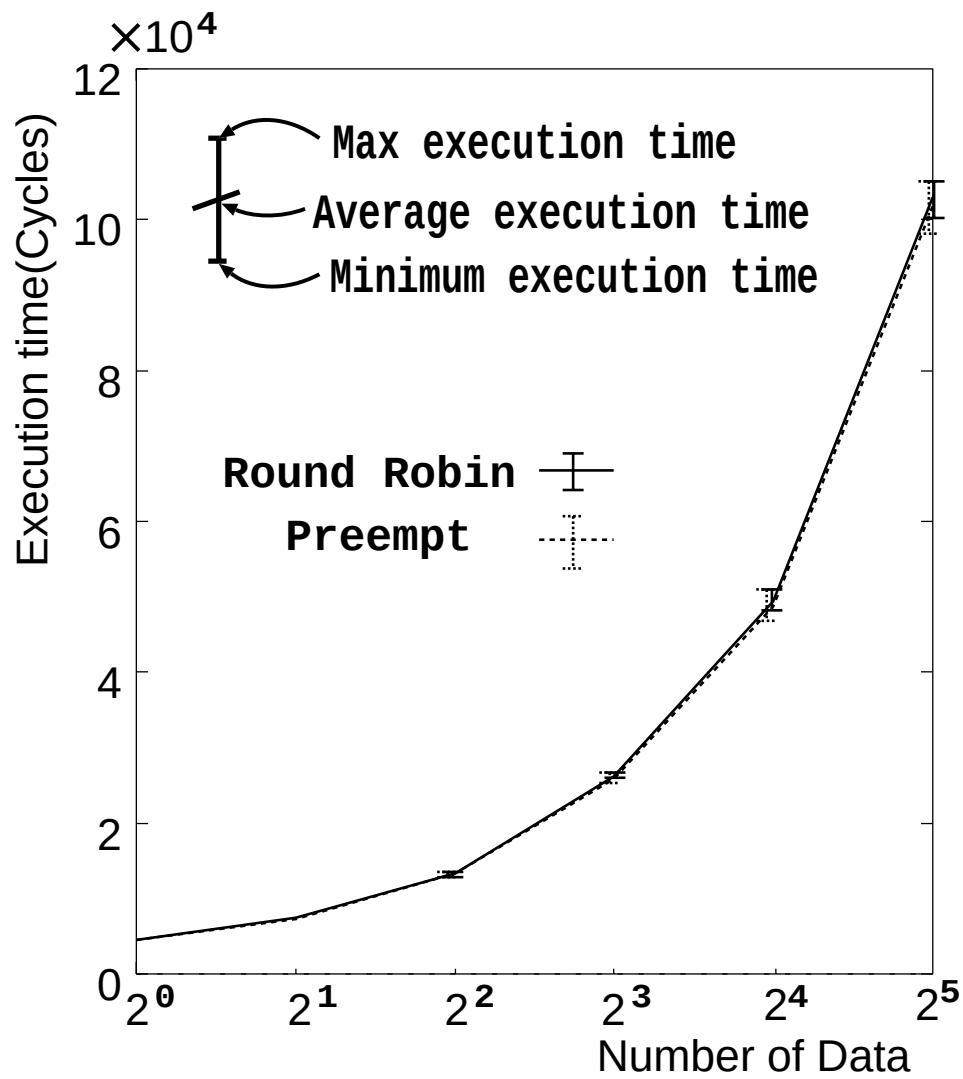


図 5.17: FFT の実行時間

5.5 適応型ルーティング

5.5.1 ルーティング法

適応型ルーティングとして、本章では Turn モデルによる方法に基づいてルーティングアルゴリズムを決定し、実験を行なう。Turn モデルは、Glass ら [14] により提案された方法で、循環が生じないようにパケットの方向転換のパターンを制限する方法である。

たとえば、図 5.18(a) のようなパケットの方向転換を許してしまうとパケットの循環が生じる。そこで、図 5.18(b) のようにパケットの転送方向に制限を加えることによりパケットの巡回を防ぐ。Turn モデルによる方法にはいくつかの種類があるが、図 5.18(b) のような Turn モデルによる方法を、特に north last 法と呼ぶ。本章では、この north last 法により実験を行なう。

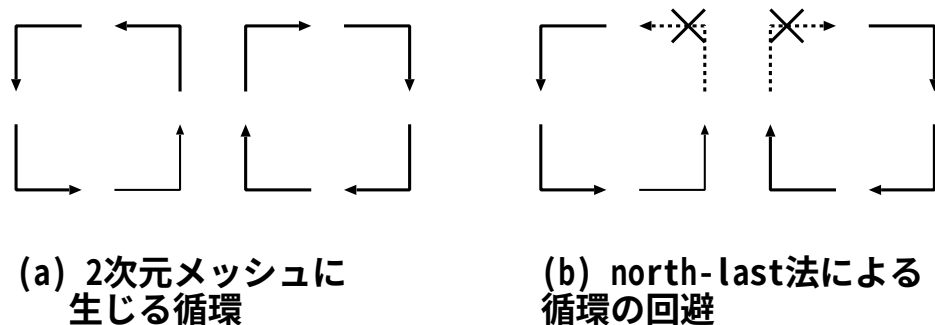


図 5.18: Turn モデル

以下に、実験に用いた north last 法のルーティングアルゴリズムを示す。ルーティングタグは、 $t[y] = s[y]$ 、 $t[x] = s[x]$ 、 $t[link] = \text{NULL}$ と定める。north last 法では、上方向の転送が最後の転送にならないといけないので、目的ノードが現在位置の真上にある時以外は上方向への転送は行わない。目的ノードが現在位置の真下にある時は、ネットワークの混雑状況を見つつ「下」→「右」→「左」の順に選択する。目的ノードが現在位置より左にある場合は（真左以外の時も）、やはりネットワークの混雑状況を見つつ「左」→「下」の順にリンクを選択する。目的ノードが現在位置より右にある場合は、「右」→「下」と選択する。Uターンは禁止とする。また、左リンクから右下隅のノードに入るとデッドロックになるため、目的ノードが現在位置より左にある時の右方向への転送も禁止とする。目的ノードが現在位置より右にある時の左方向への転送も、同様の理由により禁止とする。

```

Routing Algorithm for 2D-mesh(north last):
Routing();
source;s[y],s[x]; destination;d[y],d[x];
tag;t[y],t[x],t[link];

while((t[y] != d[y]) or (t[x] != t[y])) do

    if t[x] = d[x],
        if t[y] < d[y], move packet to upper node;t[y] = t[y] + 1;
        t[link] = upper;endif;
        if t[y] > d[y],
            if lower node is not congested & t[link] != lower,
                move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
            elseif right node is not congested & t[link] != right,
                move packet to right node;t[x] = t[x] + 1;t[link] = right;
            elseif left node is not congested & t[link] != left,
                move packet to left node;t[x] = t[x] - 1;t[link] = left;
            elseif t[link] != lower,
                move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
            elseif t[link] != right,
                move packet to right node;t[x] = t[x] + 1;t[link] = right;
            else
                move packet to left node;t[x] = t[x] - 1;t[link] = left;
            endif;
        endif;
    endif;

    if t[x] > d[x],
        if left node is not congested & t[link] != left,
            move packet to left node;t[x] = t[x] - 1;t[link] = left;
        elseif lower node is not congested & t[link] != lower,
            move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
        elseif t[link] != left,

```



```

        move packet to left node;t[x] = t[x] - 1;t[link] = left;
    else
        move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
    endif;
endif;

if t[x] < d[x],
    if right node is not congested & t[link] != right,
        move packet to right node;t[x] = t[x] - 1;t[link] = right;
    elseif lower node is not congested & t[link] != lower,
        move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
    elseif t[link] != right,
        move packet to right node;t[x] = t[x] - 1;t[link] = right;
    else
        move packet to lower node;t[y] = t[y] - 1;t[link] = lower;
    endif;
endif;

endwhile;
end.

```

5.5.2 ランダム通信の平均転送時間および平均スループット

図 5.19にランダム通信の平均転送時間を、図 5.20に平均スループットを、それぞれ示す。north last 法では、混雑しているリンクを迂回しつつ目的のノードを目指すため、迂回することによってかえって性能が悪化する場合がある。今回の実験で、パケット生成率が 5.0×10^{-3} を超えるあたりから平均スループットが低下しているのはそのためと思われる。ラウンドロビンと占有法を比較した場合、占有法がやや性能の低下が起こりにくくなっている。また、平均転送時間についても、占有法の方が短い時間で転送が行なわれている。

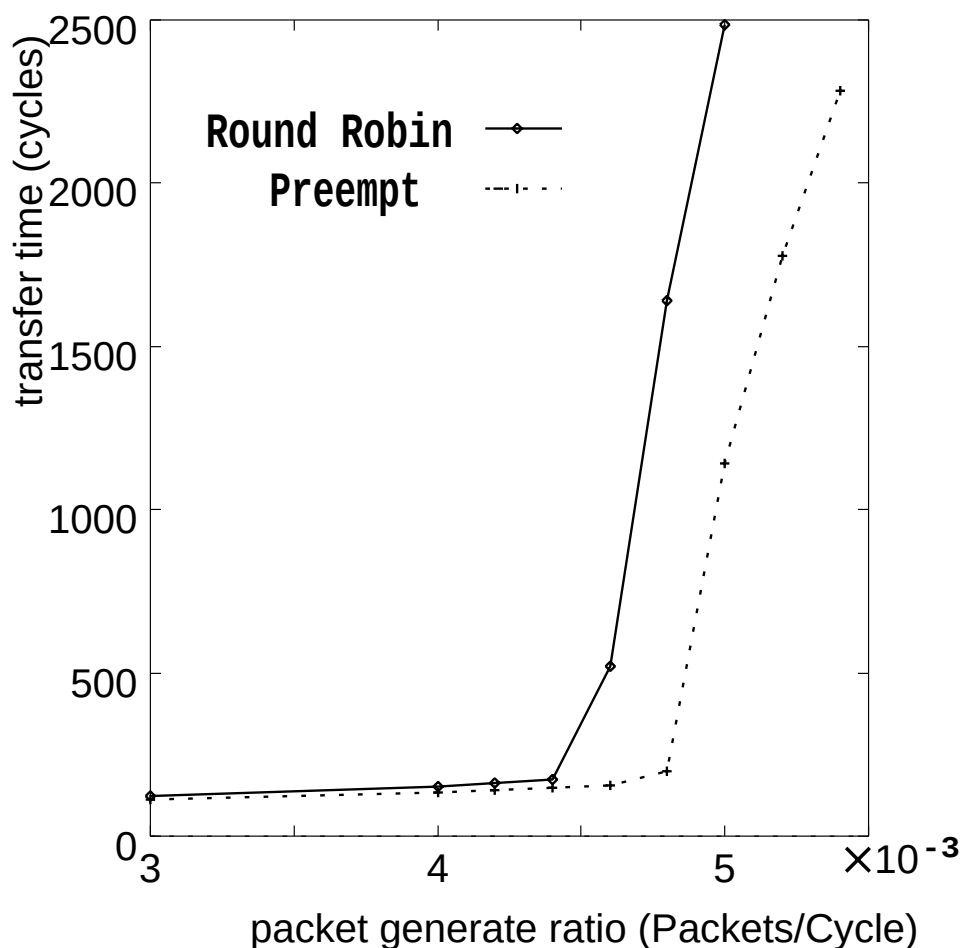


図 5.19: north last 法によるランダム通信の平均通信時間

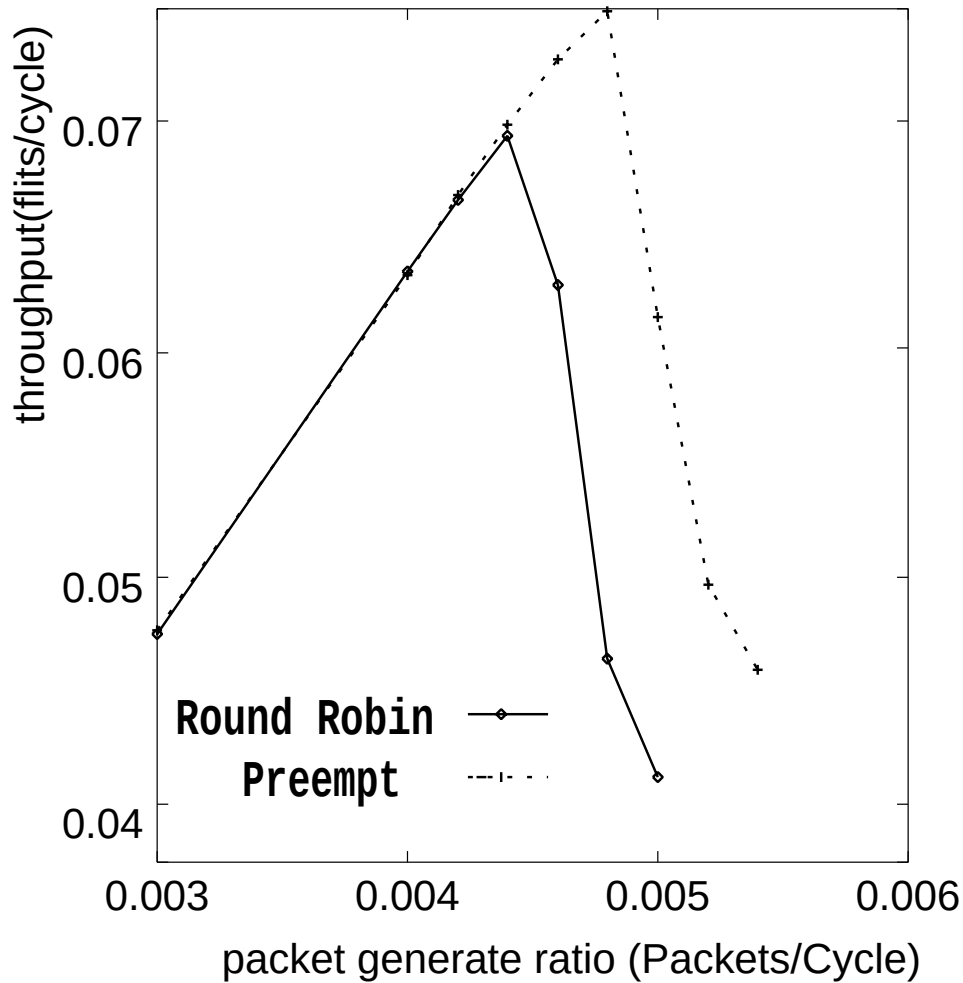


図 5.20: north last 法によるランダム通信の平均スループット

5.6 まとめ

本章では、二次元格子網上における各手法の性能比較を行った。その結果ランダム通信では、パケット長がいずれの場合についても、平均転送時間は占有法の方が短く、平均スループットは占有法の方が大きくなり、双方とも占有法が良好な結果を示した。一方、FFT を実行した結果、平均転送時間は改善されたが、実行時間はほとんど改善されなかった。適応ルーティングである north last 法で実験を行った結果、パケット生成率を上げてゆくと両者の手法とも大きな性能低下が起きたが、占有法では若干性能の低下が起こりにくくなることが分かった。

次の第 6 章では、階層型相互結合網である TESH での性能評価を行う。

第 6 章

階層型相互結合網 TESH における性能評価

6.1 はじめに

本章では、階層型相互結合網である TESH における性能比較を行う。階層型相互結合網はその他の結合網と異なり、最下位レベルネットワークである基本モジュール (BM) 内のみで行なわれる通信と基本モジュール間で行なわれる通信で通信性能が異なる。そのため、BM 間のリンクにトラフィックが集中するなどの特徴を有する一方、BM 間通信と BM 内通信で異なる優先順位を設けることにより、性能の向上が得られる可能性がある。そこで、BM 間通信を優先して転送する階層型結合網用占有法を新たに提案し、性能を評価する。

評価基準は、ランダム通信の平均転送時間および平均スループット、FFT と四方向通信の平均パケット転送時間および実行時間である。まず 6.2 で TESH のネットワーク構成を説明し、6.3 で TESH のルーティング法およびデッドロック回避の方法を説明し、TESH に用いるための、階層型結合網用占有法のフロー制御を説明する。次に 6.4 でランダム通信の性能、6.5 で FFT における性能、さらに 6.6 で、四方向通信の性能を評価する。

6.2 TESH のネットワーク構成

TESH は、Jain[13] らによって提案された階層型相互結合網で、三次元 VLSI/ULSI への実装を考慮した結合網である。

図 6.1 に、TESH の最下位レベルネットワークである基本モジュール (BM) の構造の一

例を示す。基本モジュールは、図のように各 PE をメッシュで結合した構造となっており、メッシュ構造の四端のリンクを利用して上位レベルネットワークを構成する。

TESH(m, L, q) ネットワークは、次のように定義される。

- 基本モジュールは $2^m \times 2^m$ メッシュにより構成される。
- L レベルの階層構造により構成される。
- 各レベルにつき 2^q 本の BM 間リンクを持つ。

本章の実験では、図 6.2 に示すような TESH(2, 2, 0) を対象とした実験を行なう。

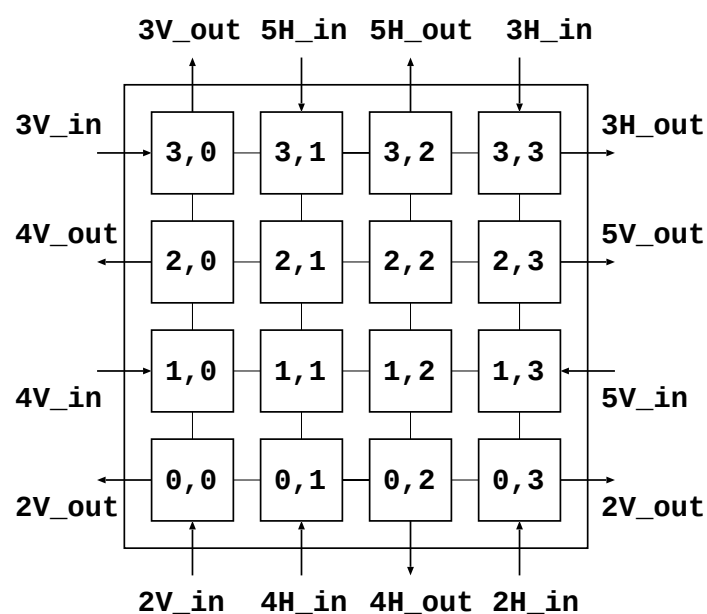


図 6.1: 基本モジュールの構成

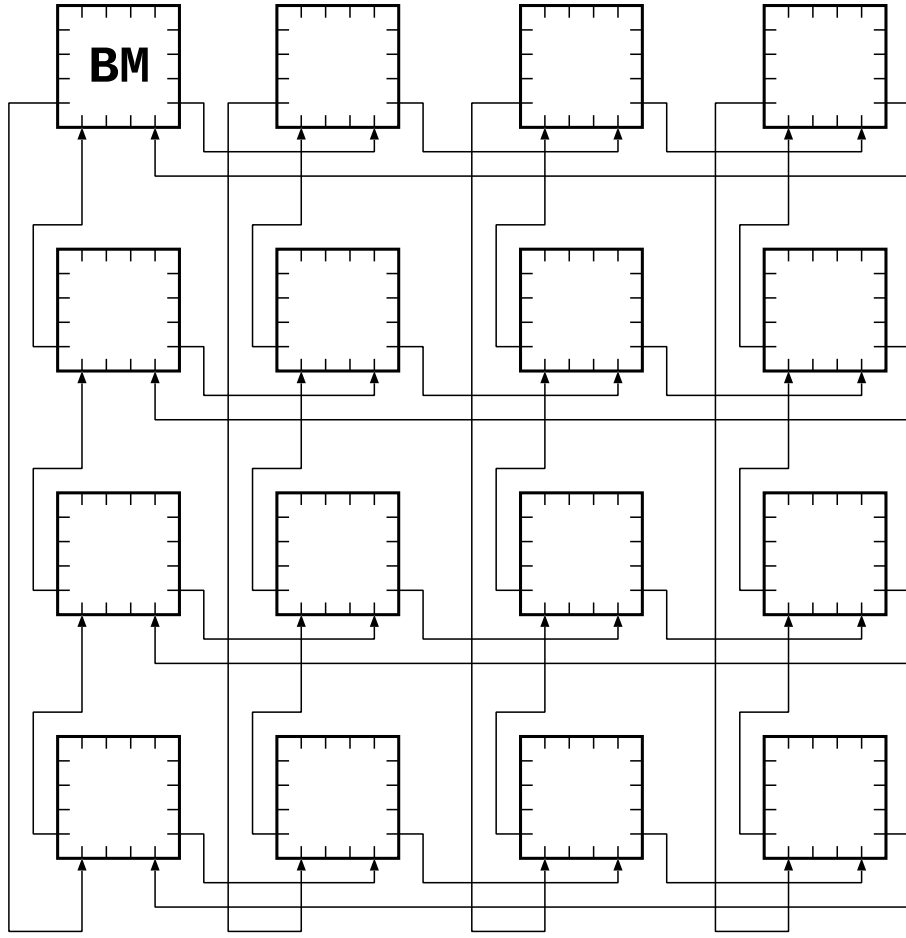


図 6.2: TESH(2,2,0) の構成

6.3 仮想チャネルフロー制御システム

6.3.1 ルーティングアルゴリズム

ルーティングアルゴリズムに関しては、[13] に記述されているものを用いる。まず、基本モジュールのサイズが 4×4 ($m = 2$)となるネットワークのノード番号を次のように表記する。

$$Address = (a_{2L-1}a_{2L-2})(a_{2L-3}a_{2L-4})\dots(a_3a_2)(a_1a_0) \quad (6.1)$$

$$= a_{n-1}a_{n-2}\dots a_3a_2a_1a_0 \quad (6.2)$$

ここで、 n は次元数をあらわし $n = 2L$ である。また a_n は 0 から 3 までを取る数であ

る。このように表記すると、 (a_1a_0) が、基本モジュール上のノードの位置、 (a_3a_2) がレベル2 ネットワーク上の基本モジュールの位置に対応する。

以上の方法により転送元ノード $S = s_{n-1}s_{n-2}\dots s_1s_0$ および $D = d_{n-1}d_{n-2}\dots d_1d_0$ を表記し、以下のようにしてルーティングタグ $T = t_{n-1}t_{n-2}\dots t_1t_0$ を定義する。

$$t_i = \begin{cases} (d_i - s_i) \bmod 2^m & , \text{for } 2 \leq i \leq n-1 \\ d_i & , \text{for } i = 0, 1 \end{cases} \quad (6.3)$$

このようにして定義したルーティングタグを用いて、次に示すルーティングアルゴリズムに従って転送を行なう。

Routing Algorithm for a Level-L Network:

```
Routing();
source; s[n-1], s[n-2], ..., s[0]; destination; d[n-1], d[n-2], ..., d[0];
tag; t[n-1], t[n-2], ..., t[0];

    for i = n-1:2;
while(t[i] != 0) do
    if i is even number, outlet_node = horizontal outlet of Level-(i/2+1); endif;
    if i is odd number, outlet_node = vertical outlet of Level-(i/2+1); endif;
    send packet to next BM; t[i] = t[i] mod 2^m;
endwhile;
    endfor;
BM_tag(t[1], t[0]) = receiving node address - destination(d[1], d[0])
while(t[1] != 0) do
    if t[1] > 0 move packet to upper node; t[1] = t[1] - 1; endif;
    if t[1] < 0 move packet to lower node; t[1] = t[1] + 1; endif;
endwhile;
while(t[0] != 0) do
    if t[0] > 0 move packet to right node; t[0] = t[0] - 1; endif;
    if t[0] < 0 move packet to left node; t[0] = t[0] + 1; endif;
endwhile;
end.
```

6.3.2 デッドロックフリーの保証

前項で示したルーティングアルゴリズムは、デッドロックフリーが保証されていないため、デッドロック回避について検討する必要がある。デッドロック回避のための方法にはさまざまなものが検討されているが、本研究では仮想チャネルを利用した方法 [7][17] を用いる。

転送されるチャネルに巡回が起らないようにルーティングが行なわれればデッドロックは回避される [18]。本研究では、巡回が起らないようなチャネルの選択をすることによりデッドロックを回避する。前項のルーティングアルゴリズムを TESH(2,2,0) に適用すると、以下に示す 3 つの順序で転送が行なわれる。

1. 転送元ノードから BM 間リンクに到達するまでの BM 内転送
2. BM 間転送
3. BM 間リンクから、転送先ノードまでの BM 内転送

したがって、これらそれぞれに対して別々にチャネルを割当てる。上記 1. と 3. はメッシュにおける転送と同じなので必要なチャネルの数は一つである。上記 2. については、トーラス網における e-cube ルーティング [19] と同じなので、二つのチャネルが必要になる。

BM 内転送と BM 間転送は、ほとんど別のリンクで行なわれるため、ほとんどのリンクは、必要なチャネル数は 2 である。ただし、BM 内のノード (0,0) と (0,3) を結ぶリンクのみは BM 間転送が行なわれるため、さらにチャネルを一つ加えなければならない。従って、デッドロックフリーを保証するために必要なチャネルの数は表 6.1 になる。

表 6.1: デッドロックフリーを保証するために必要なチャネル数

BM 間リンク	2 個
ノード (0,0) と (0,3) を結ぶ BM 内リンク	3 個
上記以外の BM 内リンク	2 個

本実験では、上に示した固定チャネル以外にも、いくつかのフリーチャネルを用意している。フリーチャネルは、上記 1.~3. のいずれの転送にも用いることができるチャネルである。これらのチャネルは、バッファが完全に空の状態の時にのみ用いるという条件付きで使用する限り、デッドロックフリーは保証される [20]。

6.3.3 階層型相互結合網のための仮想チャネルフロー制御

TESH に限らず、多くの階層型相互結合網は、通信の局所性を利用してコストパフォーマンスを得ることを目的としている [21]。したがって BM 間通信の通信性能がある程度犠牲にされるが、アプリケーションによっては BM 内通信と BM 間通信のこのような隔差が性能に大きく影響する可能性が考えられる。そこで、BM 間通信を行なうパケットの優先順位を高く設定した仮想チャネルフロー制御について検討する。

この手法は、BM 内通信用のパケット同士（あるいは BM 間通信用のパケット同士）は、占有法によって制御を行なうが、BM 内通信用と BM 間通信用のパケットが衝突した時は BM 間通信用パケットを優先的に通過させるという方法である。

6.4 ランダム通信による性能評価

6.4.1 評価方法

一定の長さを持つパケットをある確率で各 PE がランダムに送信するランダム通信により性能評価を行なう。使用するネットワークは TESH(2,2,0) とする。仮想チャネル数は 4 とし、デッドロックフリーのために必要なチャネル以外はすべてフリーチャネルとしている。シミュレーションは 20000 サイクルまで実行した。評価基準は、平均転送時間および平均スループットである。なお、パケット長は 16 としている。通信パターンは、単純に全ての PE 同士でランダムに通信を行うもののみについて評価している。

6.4.2 平均通信時間および平均スループット

ランダム通信におけるラウンドロビン、占有法および階層型結合網用占有法のパケット生成率に対する平均転送時間および平均スループットを図 6.3, 図 6.4 に示す。パケット長をいずれにした場合も、占有法が良い結果を示している。性能向上の割合はメッシュの場合よりも大きく、およそ 20 % 程度の性能向上となっている。階層型結合網用占有法に関しては、占有法と比べてほとんど性能の変化は認められなかった。

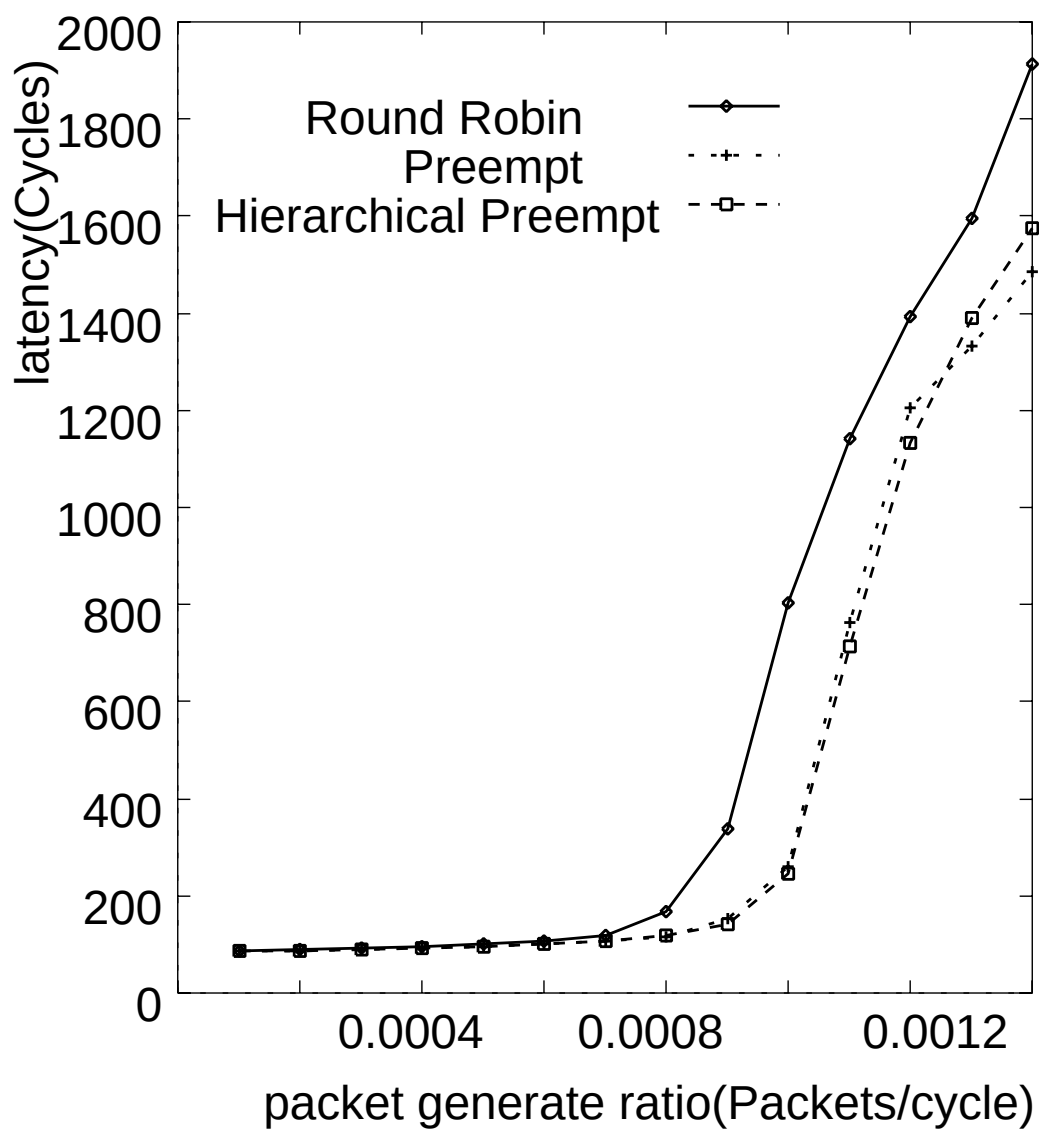


図 6.3: TESH におけるランダム通信の平均転送時間

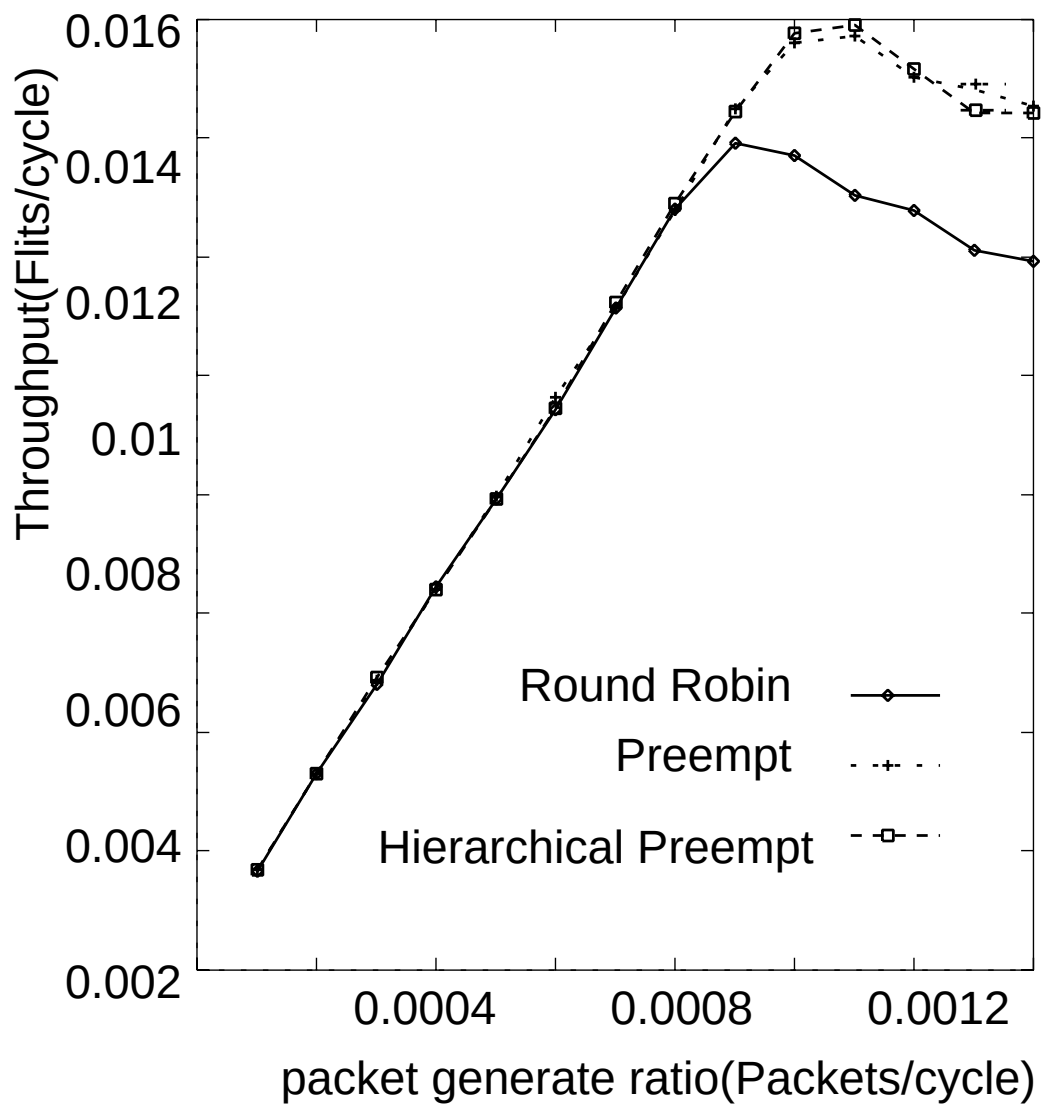


図 6.4: TESH におけるランダム通信の平均スループット

6.4.3 考察

5.3.3の議論と同じ方法で、占有法の性能が向上する理由を、特にトラフィックが混雑する BM 間リンクの状態を調査することにより考察する。全部で 32 個存在する BM リンクの稼働率を図 6.5 に示す。平均すると、占有法の稼働率は 52.45 % となり、ラウンドロビンの 46.25 % を上回っている。

次に、フリットの転送が行われない 3 つの理由について考察する。すでに 5.3.3 で、リンクがフリットを転送できない理由として、以下の 3 点を挙げた。

1. パケットが存在していないために転送が行なわれない。
2. バッファは予約されている（つまりパケットは存在する）が、フリット間に空白があるためにフリットがバッファに存在せず、転送が行なえない。
3. 入力側のバッファがいっぱいになってしまっているために、フリットがブロックされてしまう。

これらについて、各時間に平均していくつのリンクで発生しているかをプロットする。結果を図 6.6～6.8 に示す。また、それぞれの平均値を表 6.2 に示す。これらの結果は、メッシュに関する実験と同傾向を示しているが、パケットが存在しないために転送が行われないケースが、他のケースに比べて相対的に少なくなっている。前者は両手法とも同程度の数なのに対して、後者はラウンドロビンの方が数多く発生するため、後者が多い方が占有法の性能向上が大きくなる。そのため、TESH のランダム通信は、メッシュのランダム通信より通信性能向上の割合が大きくなるものと思われる。

表 6.2: 転送が行なえないリンク数の平均値

転送が行なえない 原因	リンク数の平均	
	ラウンドロビン	占有法
パケットが存在しない	2.20	2.64
フリット間に空白がある	6.38	4.42
前方のパケットにブロックされる	9.26	6.40

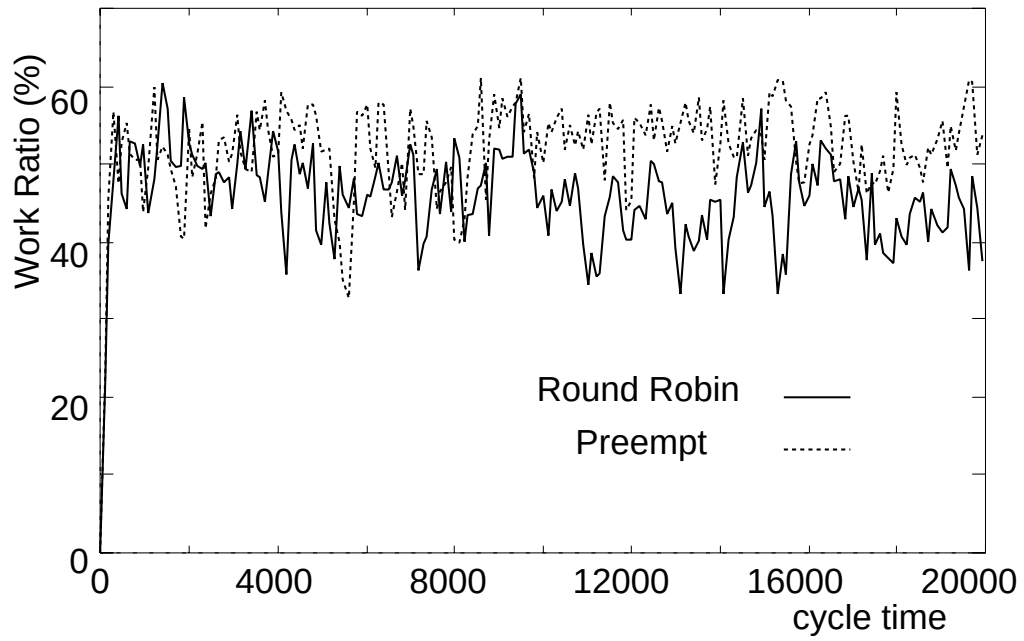


図 6.5: リンクの稼働率

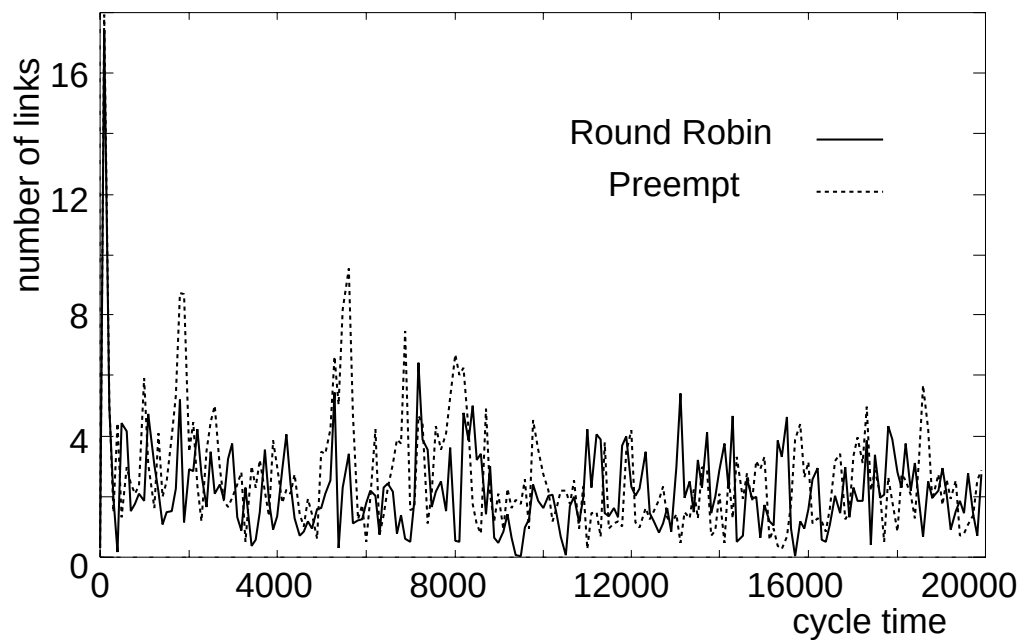


図 6.6: パケットが存在しないために転送を行わないリンクの数

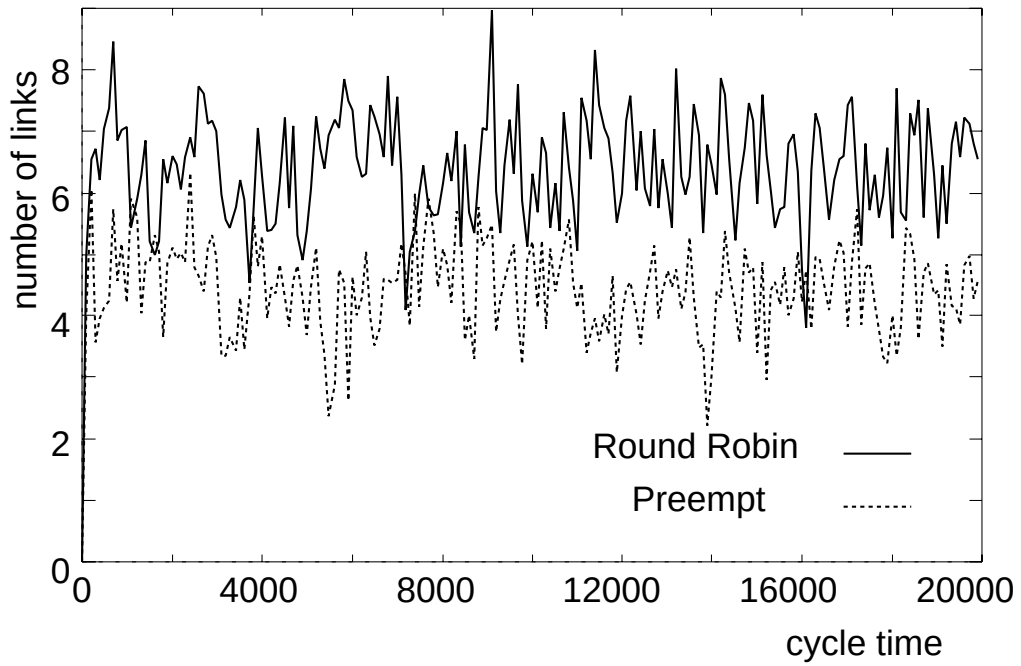


図 6.7: フリット間に空白があるために転送が行えないリンクの数

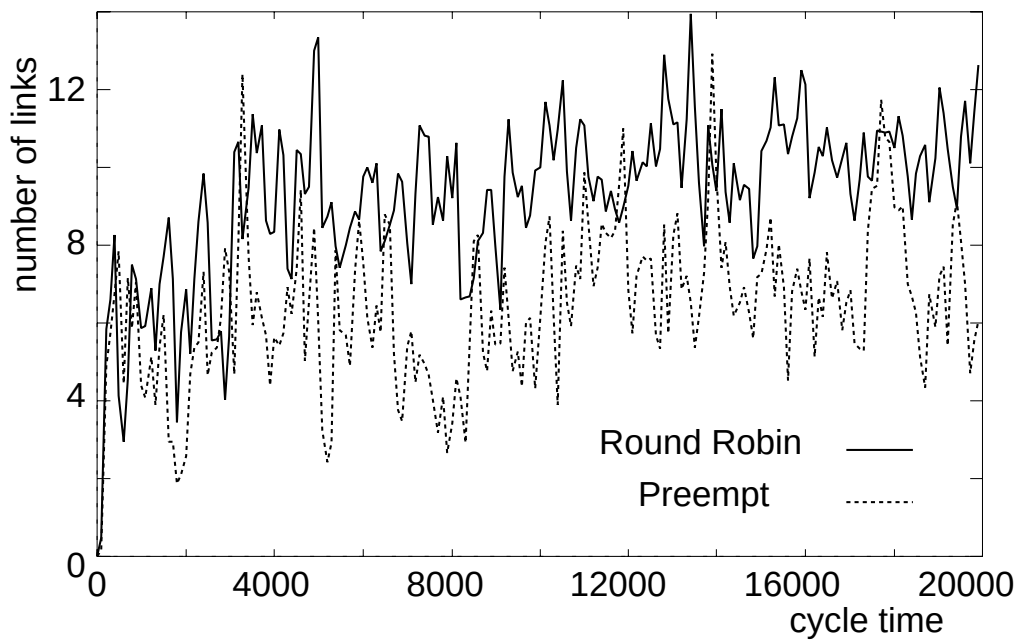


図 6.8: 前方のパケットにブロックされるために転送が行えないリンクの数

階層型結合網用占有法が、占有法に比べてほとんど性能に変化が見られない理由を考察するために、両者の転送時間の分布を測定する。転送時間分布は、パケット生成率 1.0×10^{-3} の時について、20000 サイクルまで実行した結果をプロットした。

図 6.9 に、測定した結果を示す。図に示すように、両者にはほとんど差が見られない。次に、同様の実験を、仮想チャネルの数を 8 に増やして行なった結果を図 6.10 に示す。チャネル数が 4 の場合と違い両者に差が見られる。パケット生成率が 1.0×10^{-3} の時には、階層型結合網用占有法の転送時間のばらつきが若干抑えられているが、図 6.11 のようにパケット生成率が高くなると、必ずしも転送時間のばらつきが抑えられているとは言い難くなる。

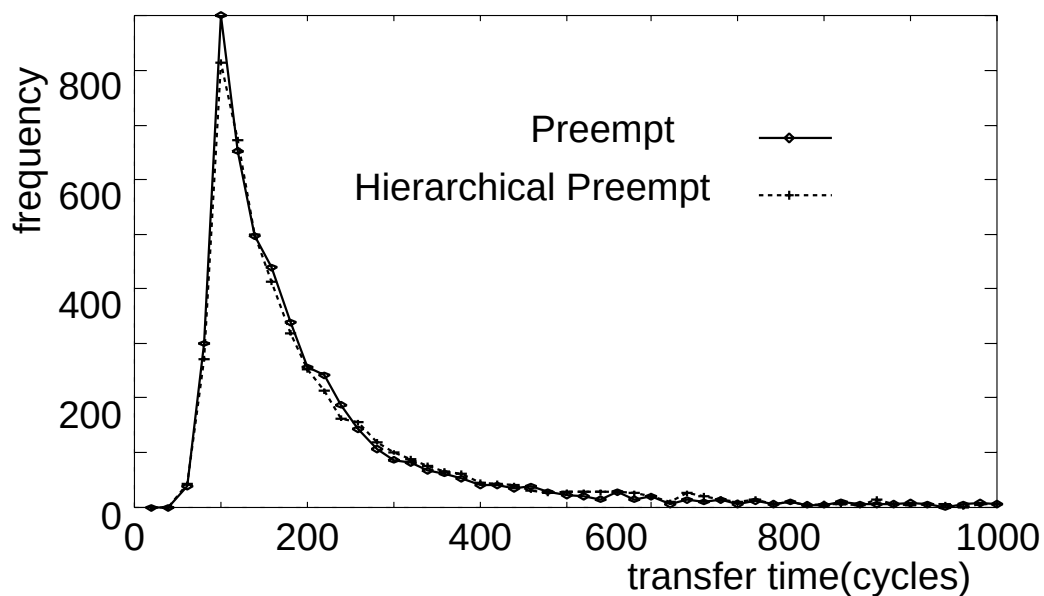


図 6.9: TESH におけるランダム通信の転送時間分布（チャネル数 4）

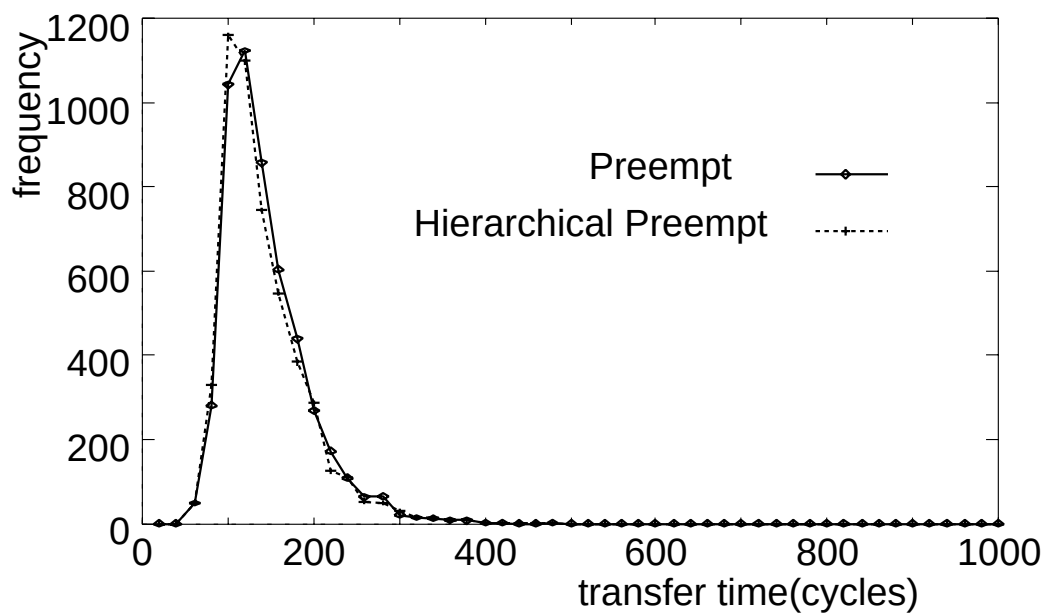


図 6.10: TESH におけるランダム通信の転送時間分布（チャンネル数 8、パケット生成率 1.0×10^{-3} ）

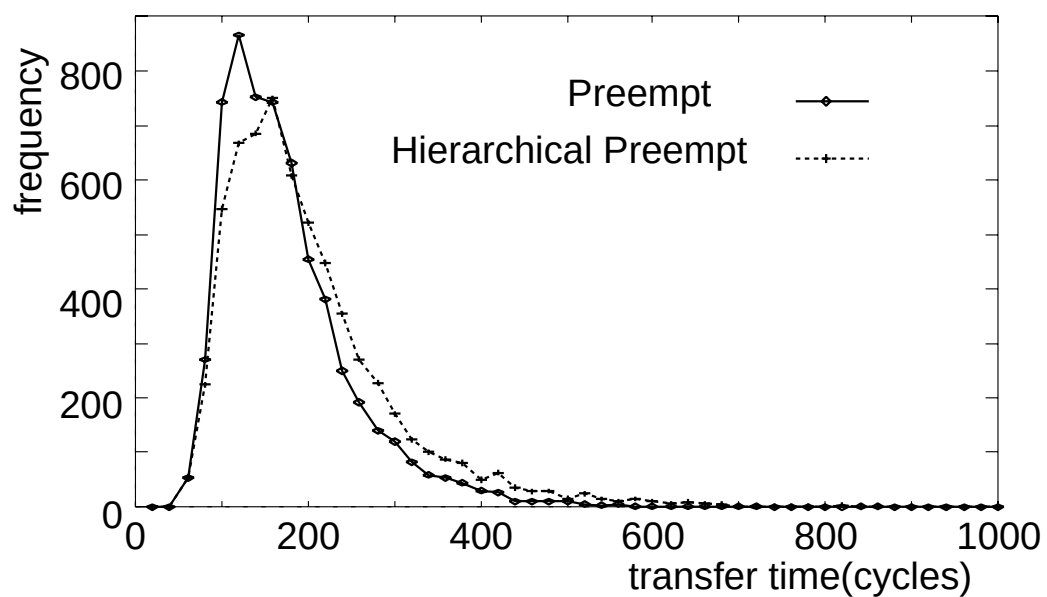


図 6.11: TESH におけるランダム通信の転送時間分布（チャンネル数 8、パケット生成率 1.2×10^{-3} ）

6.5 FFT による性能評価

6.5.1 マッピング手法

シミュレータ上で FFT をトレースして性能を評価する。まず、 256×2^n 個のデータを以下のように TESH(2,2,0) にマッピングする。なお、ここではデータ番号および PE 番号を 2 進表記で表現している。

1. データに 0 から 256×2^n までの番号をつける。この時につけた番号を 2 進表記すると、 $b_{n+7} \dots b_3 b_2 b_1 b_0$ と表現される。
2. データ番号 $b_{n+7} \dots b_3 b_2 b_1 b_0$ を、 $PE(b_7 b_6, b_5 b_4)(b_3 b_2, b_1 b_0)$ に割り当てる。

すると、 $PE(0,0)(0,0)$ にはデータ番号 $256 \times i (i = 0, 1, 2, \dots)$ 、 $PE(0,0)(0,1)$ にはデータ番号 $256 \times i + 1 (i = 0, 1, 2, \dots)$ 、という具合にマッピングされる。

16×16 メッシュ上にこのようにマッピングした場合、通信は全部で 8 回行なわれる。その際の各 PE の i 回目の転送先は、PE 番号を 2 進表記して $PE(b_7 b_6, b_5 b_4)(b_3 b_2, b_1 b_0)$ とあらわした場合、 b_i を反転した番号の PE となる。例えば、一回目の転送先は $PE(b_7 b_6, b_5 b_4)(b_3 b_2, b_1 \overline{b_0})$ 、二回目の転送先は $PE(b_7 b_6, b_5 b_4)(b_3 b_2, \overline{b_1} b_0)$ となる。上のようにマッピングした場合、最初の 4 回の転送は基本モジュール内のみでの通信となる。基本モジュール間の通信は、後の 4 回の通信のみである。また、 256×2^n 個のデータを扱う場合、FFT では全部で $n + 8$ ステージのバタフライ演算を行なうことになるが、この場合最初の 8 ステージ以降は同一 PE 上にあるデータ同士でのやり取りになる。

本シミュレーションの実験の条件は、5.4.2 と同様、表 6.3 のように設定している。表 6.3 のパラメータを用いると、通信と通信の間の演算のための時間 T は 5.4.2 と同様

$$T = T_t + T_s + n \times T_b \quad (6.4)$$

1 メッセージのメッセージ長 L は

$$L = n \times L_d \quad (6.5)$$

となる。

6.5.2 通信に要する時間の比較

各手法について、一回の通信に要した時間を図 6.12 に示す。図に示すように、占有法はラウンドロビンに比べて平均通信時間が改善されている。階層型結合網用占有法について

表 6.3: FFT のシミュレーションのための諸条件

条件	記号	値
バタフライ演算一回に要する時間	T_b	220 cycles
通信のためのセットアップ時間	T_s	120 cycles
転送先の計算に要する時間	T_t	120 cycles
データ一つあたりのメッセージ長	L_d	16 flits

は、ラウンドロビンよりは平均通信時間は短くなっているが、占有法より悪化している。これは、BM 内通信を行っているパケットに優先順位の高い BM 間通信のパケットが割り込むために、パケットの流れが乱され、前者のパケットのフリット間に空白ができてしまうためと推測される。

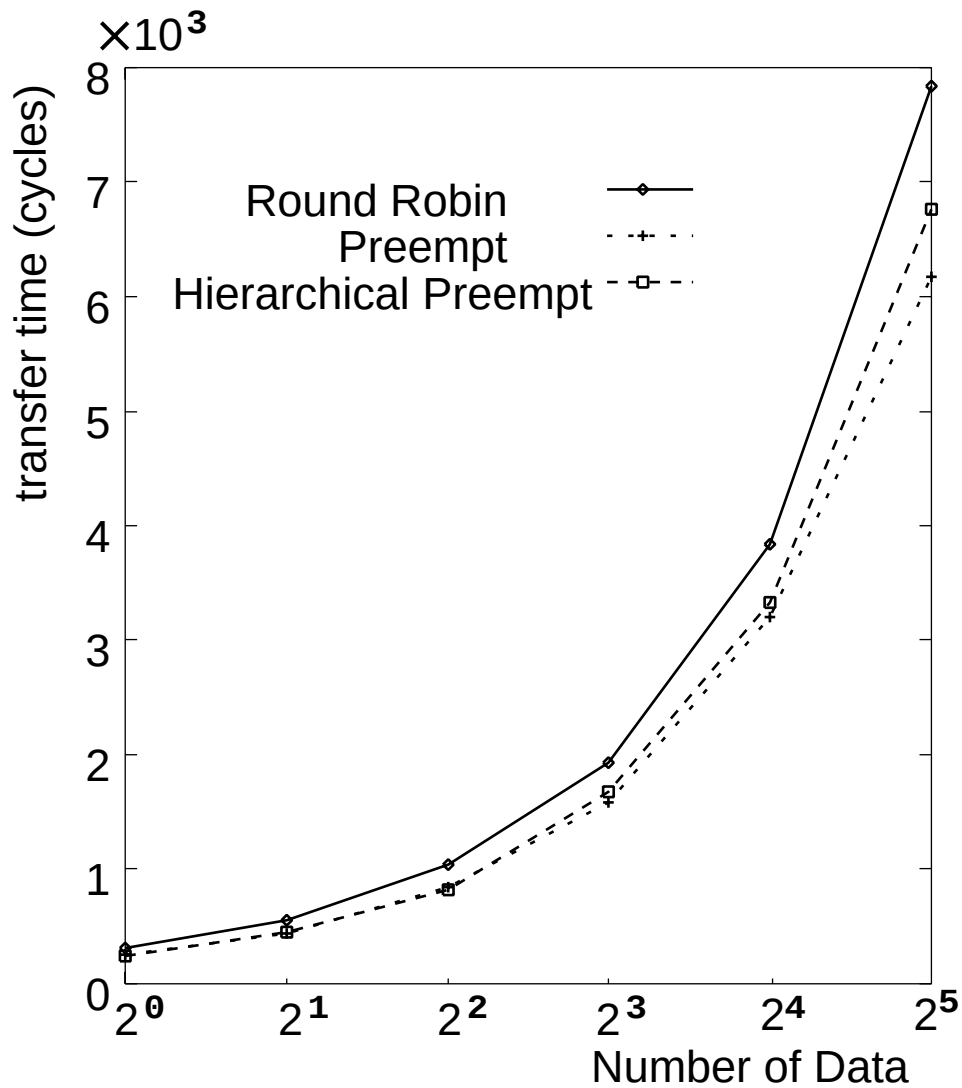


図 6.12: FFT における各手法の平均通信時間

6.5.3 実行時間の比較

各手法について、FFT の実行に要した時間を図 5.17 に示す。図中の折れ線は各 PE の実行時間の平均値を示し、縦線の上端は各 PE の実行時間の最大値、下端は最小値を示している。

TESH(2,2,0) 上で実行した場合、ラウンドロビンに比べて占有法や階層型結合網用占有法の実行時間は改善されている。このことから、占有法はメッシュのような均一な結合網

よりも、階層型結合網に有利な手法であることが考えられる。占有法と階層型結合網用占有法を比較した場合、占有法の実行時間の方が短いのは、6.5.2に示したように占有法の転送時間が短いのが原因と思われる。

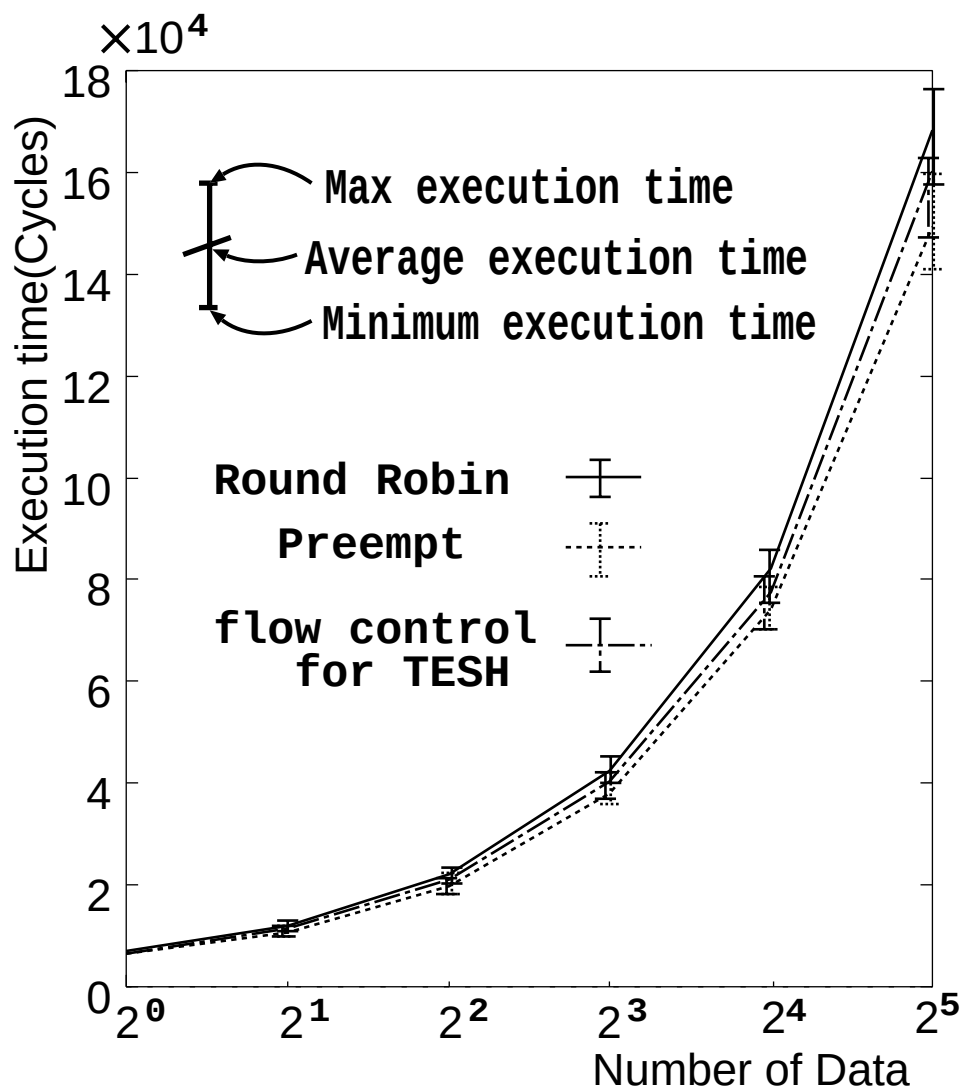


図 6.13: FFT の実行時間の比較

6.6 四方向通信による性能評価

6.6.1 通信方法

各 PE の周囲四方向の PE に対する通信を行ない、その通信性能を評価する。通信元の PE 番号を $(a_3, a_2)(a_1, a_0)$ と表現し、各 PE の通信先は、以下のルールに従うものとする。

$$dest_u = \begin{cases} ((a_3 + 1) \bmod 4, a_2)(0, a_0) & , \text{ for } a_1 = 3 \\ (a_3, a_2)(a_1 + 1, a_0) & , \text{ for others} \end{cases} \quad (6.6)$$

$$dest_d = \begin{cases} ((a_3 - 1) \bmod 4, a_2)(3, a_0) & , \text{ for } a_1 = 0 \\ (a_3, a_2)(a_1 - 1, a_0) & , \text{ for others} \end{cases} \quad (6.7)$$

$$dest_l = \begin{cases} (a_3, (a_2 - 1) \bmod 4)(a_1, 3) & , \text{ for } a_0 = 0 \\ (a_3, a_2)(a_1, a_0 - 1) & , \text{ for others} \end{cases} \quad (6.8)$$

$$dest_r = \begin{cases} (a_3, (a_2 + 1) \bmod 4)(a_1, 0) & , \text{ for } a_0 = 3 \\ (a_3, a_2)(a_1, a_0 + 1) & , \text{ for others} \end{cases} \quad (6.9)$$

ここで、 $dest_u$ 、 $dest_d$ 、 $dest_l$ 、 $dest_r$ はそれぞれ、上方向、下方向、左方向、右方向の転送先を表わしている。

この定義に従うと、基本モジュール内では、それぞれに隣接した PE との通信となる。また、基本モジュール間の通信は、図 6.14 のようになる。

通信は、各 PE について「送信」→「受信」→「送信」→「受信」→… という手順に従って行われる。送信は、上下左右の四つすべてに対して行なわれ、各 PE が隣の四つの PE からパケットを受信し終えた後に次の送信を行う。したがって、「上方向へ送信」「下方向へ送信」「左方向へ送信」「右方向へ送信」「四つのパケットを受信」の動作をもって 1 ステップとし、1 ステップの動作がすべてを終えた後に次のステップに進むことになる。

実験は、上で示したステップを複数回実行し、パケットの平均転送時間および実行終了までの時間で評価した。なお、転送するパケットのパケット長は 64 フリットとした。

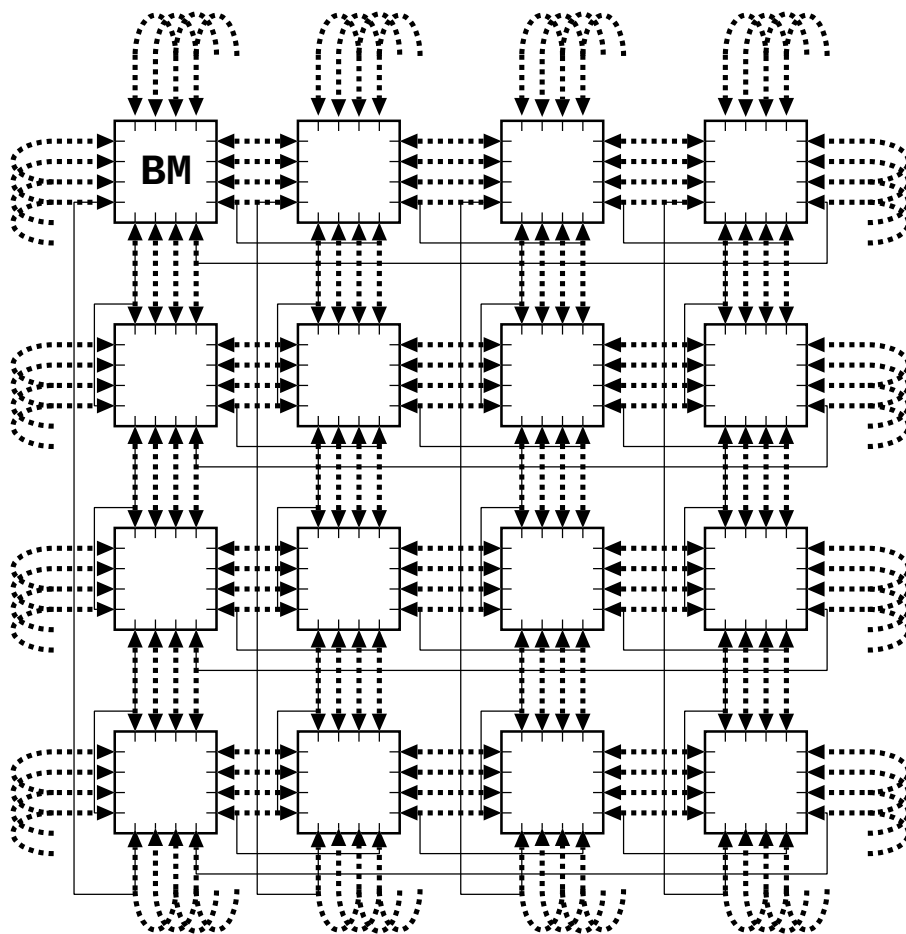


図 6.14: 四方向通信の BM 間通信

6.6.2 通信時間および実行時間

四方向通信の平均通信時間および実行時間を図 6.15、6.16に示す。この場合もやはり 6.5.2 および 6.5.3で示した FFT の実験と同じ傾向を示している。階層型結合網用占有法が占有法に比べて良い結果を示さない理由は、通信時間の悪化と、十分に通信時間のばらつきを抑えることができていないことが原因と思われる。

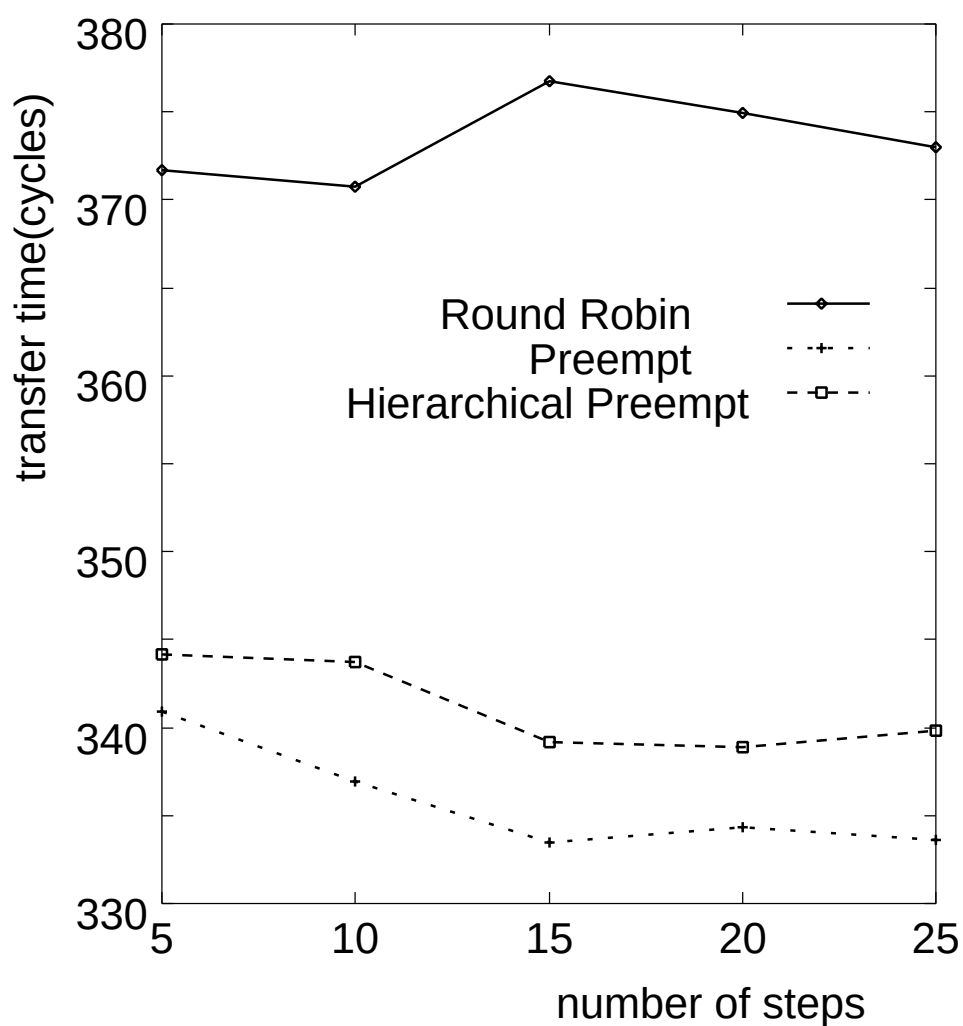


図 6.15: 四方向通信の平均通信時間

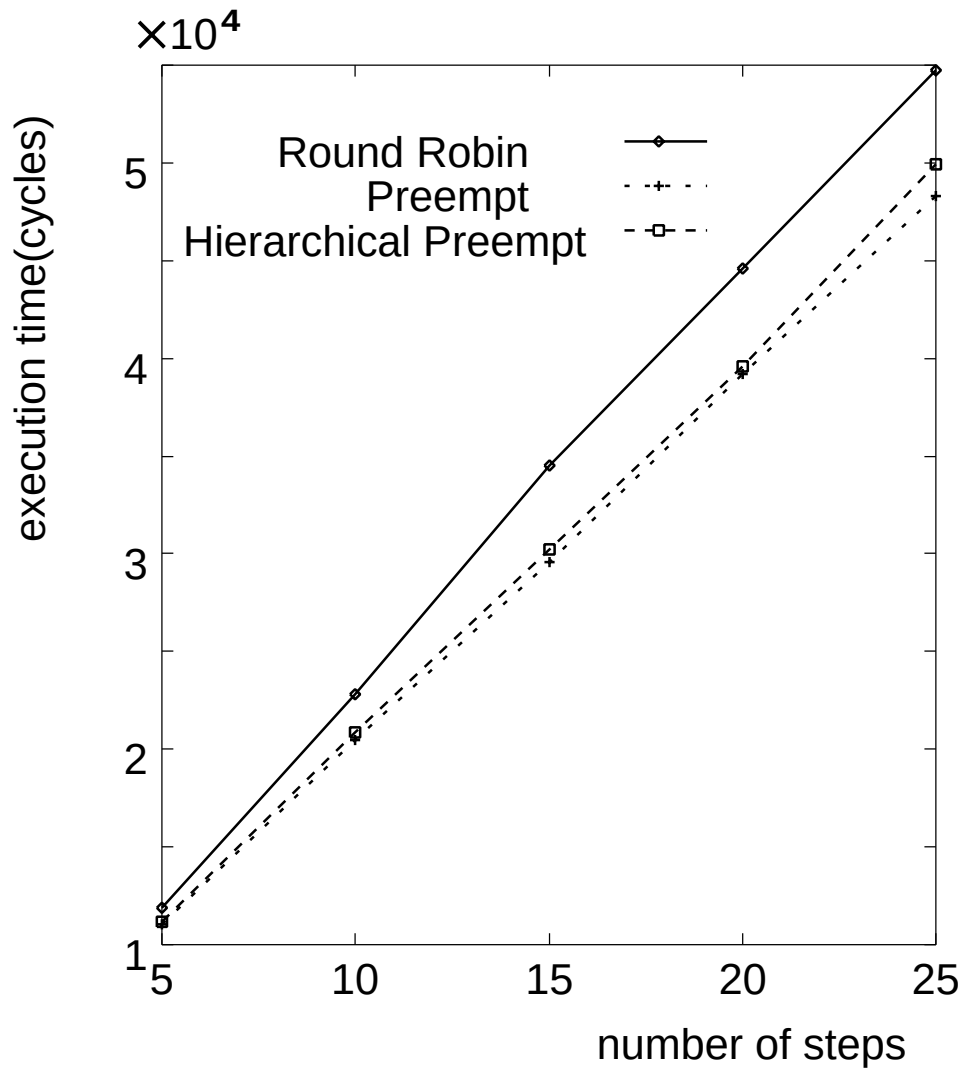


図 6.16: 四方向通信の実行時間

6.7 まとめ

本章では、階層型相互結合網である TESH における仮想チャネルフロー制御法について説明し、その性能を評価した。はじめに、TESH のネットワーク構成を説明し、ルーティング法、デッドロック回避の方法を述べ、階層型結合網用占有法のフロー制御を説明した。

実験としては、まずランダム通信の平均通信時間および平均スループットを評価した。

その結果占有法は、メッシュの場合よりも大きな性能の向上が得られたが、階層型結合網用占有法は性能に変化はなかった。そのため階層型結合網用占有法の転送時間分布を測定した所、占有法とほとんど変わらないという結果になった。

FFT や四方向通信に関しても性能評価を行った。その結果、占有法では実行時間が改善されたが、階層型結合網用占有法ではかえって平均通信時間の悪化を招くため、FFT や四方向通信に関しても良い結果が得られないことが分かった。階層型結合網用占有法に関しては、今後は平均通信時間のばらつきを抑えられる方法を検討する必要があると思われる。

第 7 章

結論

並列計算機上でメッセージを転送する際の効率的な手法として、仮想チャネルを用いたワームホールルーティングが広く用いられている。

ワームホールルーティングは、転送時間が短くて済むということからストアアンドフォワード方式に代わって使用されるようになってきた。また、パケット長が長い場合、バーチャルカットスルーに比べてハードウェアコストが少なく済むという利点がある。反面、ブロッキングが多いという欠点を持つため、仮想チャネルを設けてブロッキングを回避する。

仮想チャネルは、ブロック回避やデッドロック回避などのために多くの並列計算機に実装されている。仮想チャネルでは複数の仮想チャネルから一つを選択して物理リンクを通すため、チャネルの優先順位をどのように決定するかというルール決め方、つまり仮想チャネルフロー制御法によりパケットの流れが異なる。現状では仮想チャネルフロー制御法についてはあまり検討されてはおらず、ラウンドロビンのようなごく単純な手法により物理リンクのフロー制御を行っている。特にワームホールルーティングの場合、ラウンドロビンにより仮想チャネルフロー制御を行うと、調停が行われる全パケットに公平に最大の遅れが発生するといった問題がある。また、階層型相互結合網のような不均一な結合網の場合にも全パケットが同じルールにより転送を行っているなど、まだ十分に検討されていない点がいくつか存在する。

そこで本論文では、メッセージに適切な優先順位を設けた仮想チャネルフロー制御方式を提案し、その性能評価を行った。まず第 1 章では、研究の背景と目的について述べ、第 2 章で、ワームホールルーティングと仮想チャネルの使用目的、利点・欠点、構造および従来型の仮想チャネルフロー制御法について述べた。第 3 章では、ワームホールルーティングにおいて従来の方法でフロー制御を行う場合、具体的に何が問題になるかについて

述べたのち、優先順位を持ったフロー制御法として占有法を提案した。第4章ではシミュレーションに用いたシステムのハードウェア構成を説明し、シミュレータ上にどのように実装したかについて述べた。

第5章では、格子結合における各仮想チャネルフロー制御手法の性能評価を行った。性能比較は4章で述べたシミュレータを用いて行ない、ランダム通信およびFFTの性能を評価した。まず、ランダム通信の平均転送時間および平均スループットの評価を行った。その結果、平均転送時間は占有法の方が短く、平均スループットは占有法の方が大きくなり、双方とも占有法が良好な結果を示した。次に、FFTを実行した場合の性能評価を行った。その結果、平均通信時間は占有法の方が短くなるが、FFTの実行時間は双方ともほぼ同じになった。最後に、適応ルーティングとしてnorth last法を行った場合のランダム通信の性能を評価した。その結果、パケット生成率を上げてゆくと両者の手法とも大きな性能低下が起きたが、占有法では若干性能の低下が起こりにくくなることが分かった。

第6章では、階層型相互結合網であるTESHにおける仮想チャネルフロー制御法について説明し、さらにその性能を評価した。はじめに、TESHのネットワーク構成を説明し、ルーティング法、デッドロック回避の方法を述べ、階層型結合網用占有法のフロー制御を説明した。実験は、まずランダム通信の平均通信時間および平均スループットを評価した。その結果占有法は、メッシュの場合よりも大きな性能の向上が得られたが、階層型結合網用占有法は性能に変化はなかった。階層型結合網用占有法の転送時間分布を測定した所、占有法とほとんど変わらないという結果になった。FFTや四方向通信に関しても性能評価を行った。その結果、占有法では実行時間が改善されたが、階層型結合網用占有法では、かえって平均通信時間の悪化を招くため、FFTや四方向通信に関しても良い結果が得られないことが明らかになった。階層型結合網用占有法に関しては、今後は平均通信時間のばらつきを抑えられる方法を検討する必要があると思われる。

今後の課題として、第一に階層型相互結合網において平均通信時間のばらつきを抑えられる方法を検討することと、TESHにおける効率的な適応ルーティングの方法を検討することが挙げられる。

謝辞

本研究を進めるにあたり、熱心に御指導頂いた北陸先端科学技術大学院大学 堀口 進 教授、阿部 亨 助教授に深く感謝いたします。

副テーマで御指導頂きました金子峰雄助教授に感謝いたします。

研究の進め方等で、日頃より御指導して下さいました山森一人助手、林亮子助手に深く感謝いたします。

最後に、有意義なアドバイスを下さったマルチメディア統合システム講座の皆様に深く感謝いたします。

参考文献

- [1] 清水, 堀江, 石畑. 高速メッセージハンドリング機構. 情処学論, Vol. 34, No. 4, pp. 638–647, 1993.
- [2] J.Kuskin et.al. The stanford flash multiprocessor. In *ISCA94*, pp. 302–313, 1994.
- [3] 坂井, 岡本, 松岡, 廣野, 児玉, 佐藤, 横田. 超並列計算機 rwc-1 の基本構想. In *JSPP93*, pp. 87–94, 1993.
- [4] L.M.Ni and P.K.McKinley. A survey of wormhole routing techniques in direct networks. *Proc of the IEEE*, Vol. 81, No. 2, pp. 62–76, 1993.
- [5] P.Kermani and L.Kleinrock. Virtual cut-through: A new computer communication switching techniques. *Computer Networks*, Vol. 3, No. 4, pp. 267–286, 1979.
- [6] W.J.Dally. Virtual-channel flow control. *IEEE Trans on Parallel and Distributed Systems*, Vol. 3, No. 2, 1992.
- [7] M.P.Merlin and J.P.Schweitzer. Deadlock avoidance in store-and-forward networks - 1: Store and forward deadlock. *IEEE Trans. on Comm.*, Vol. COM-28, No. 3, pp. 345–354, 1980.
- [8] D.H.Linder and J.C.Harden. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *IEEE Trans. on Computers*, Vol. C-40, No. 1, pp. 2–12, 1991.
- [9] J.Rexford, J.Hall, and K.G.Shin. A router architecture for real-time point-to-point networks. In *ISCA '93*, pp. 237–246, 1996.
- [10] H.Song, B.Kwon, and H.Yoon. Throttle and preempt: A new flow control for real-time communication in wormhole networks. In *ICPP*, pp. 198–202, 1997.

- [11] 戸田, 西田, 高橋, Michell, 山口. 優先度先送り方式による実時間相互結合網用ルータチップの実現と性能. 情処学論, Vol. 36, No. 7, pp. 1619–1629, 1995.
- [12] 天野, 西, 西村, 工藤. 超並列計算機プロトタイプ JUMP-1 のルータチップ. 信学技報 (cpsy-94-98), 電子情報通信学会, 1995.
- [13] V.K.Jain, T.Ghirmai, and S.Horiguchi. Tesh:a new hierarchical interconnection network for massively parallel computing. *IEICE Transactions*, Vol. E80-D, No. 9, pp. 837–846, 1997.
- [14] C.J.Glass and L.M.Ni. Maximally fully adaptive routing in 2d meshes. In *ISCA92*, pp. 278–287, 1992.
- [15] Jong-Pyng Li and Matt W. Mutka. Real-time virtual channel flow control. In *International Phoenix Conference on Computers and Communications*, pp. 97–103, 1994.
- [16] 富田眞治. 並列コンピュータ工学. 昭晃堂, 1996.
- [17] 堀江, 石畑, 池坂. 並列計算機 ap1000 における相互結合網のルーティング方式. 信学論 D-I, Vol. J75-D-I, No. 8, pp. 600–606, 1992.
- [18] 天野英晴. 並列コンピュータ. 昭晃堂, 1995.
- [19] W.J.Dally and C.L.Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. on Computers*, Vol. C-36, No. 5, pp. 547–553, 1987.
- [20] J.Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Trans. on Parallel and Distributed Systems*, Vol. 4, No. 12, pp. 1320–1331, 1993.
- [21] Y.R.Potlapalli. Trend in interconnection networks topologies: Hierarchical networks. In *International Conf. on Parallel Processing Workshop*, pp. 24–29, 1995.

研究業績

- [1] 三浦 康之, 阿部 亨, 堀口 進,
“ワームホールルーティングにおける仮想チャネルのフロー制御”,
第 57 回情報処理学会全国大会, Vol.1, pp.48-49, 1998.
- [2] 三浦 康之, 阿部 亨, 堀口 進,
“ワームホールルーティングにおける仮想チャネルフロー制御”,
情報処理学会第 74 回 HPC 研究会, HPC74-11, pp.59-64, 1998.