バンク型マルチポートメモリを用いたスイッチアーキテクチャ

小林一彦† 藤井崇之††† 弘中哲夫†† 小出哲士††† マタウッシュ ハンス ユルゲン†††

† 広島市立大学大学院 情報科学研究科 情報工学専攻
†† 広島市立大学 情報科学部 情報工学科
††† 広島大学ナノデバイス・システム研究センター

近年，インターネットの急速な普及によりネットワークはインフラとして欠かせないものになってきており，その高速化には目覚しいものがある．現在，このような状況に伴うトラヒックの増加に対応できる大容量スイッチが求められている．しかし，クロスバ等を用いた既存スイッチではスイッチファブリック内のパスを取り合う際に発生するブロッキングによる性能低下を完全に防ぐことはできない．また，スイッチに入力されるトラヒックに応じて各インタフェースに適切なバッファサイズを割り当てることも困難である．本稿ではこれらの問題を解決するバンク型マルチポートメモリを用いたスイッチアーキテクチャの提案を行い，そのシミュレーション評価を行う．

Switch Architecture with Banked Multi-port Memory

Kazuhiko KOBAYASHI† Takayuki FUJII††† Tetsuo HIRONAKA††
Tetsushi KOIDE††† Hans Jürgen Mattausch†††

†Graduate School of Information Sciences, Hiroshima City University
††Faculty of Information Sciences, Hiroshima City University
†††Research Center for Nanodevices and Systems, Hiroshima University

In recent years, network infrastructure for high-speed network is becoming more important with the rapid spread of the Internet. And especially there is a big demand in the high-speed network switches to deal with the network requirements. But network switches using switch fabrics such as crossbar, cannot avoid blocking caused by routing races that results in performance loss. Moreover it is difficult to change the buffer-size of each interface adaptively by the amount of traffic inputted. In this paper, we propose the switch architecture with the banked multi-port memory, which can solve these problems, and show some evaluation results by software simulation.

1　はじめに

2001年末における国内のインターネット利用者数は約6,000万人と推計されており，2005年には8,720万人に達することが予測されている[1]．近年，このようなインターネットの急速な普及によりネットワークはインフラとして欠かせないものになってきており，その高速化には目覚しいものがある．現在，このような状況に伴うトラヒックの増加に対応できる大容量スイッチが求められている．しかし，クロスバ等を用いた既存スイッチではスイッチファブリック内のパスを取り合う際に発生するブロッキングによる性能低下を完全に防ぐことはできない．また，スイッチに入力されるトラヒックに応じて各インタフェースに適切なバッファサイズを割り当てることも困難である．本稿ではこれらの問題を解決するバンク型マルチポートメモリを用いたスイッチアーキテクチャの提案を行い，そのシミュレーション評価を行う．

2　既存スイッチアーキテクチャ

2.1　入出力バッファ型スイッチ

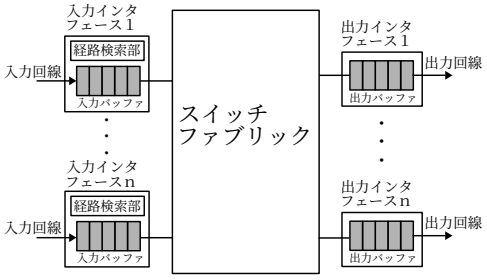

図 1: 入出力バッファ型スイッチの構成図

共有バス型スイッチや共有メモリ型スイッチではバスの動作速度やメモリのアクセス速度がボトルネックとなってしまう[2][3]ため，これらの方法で大容量スイッチを実現することは難しい．これらの問題を解決することができる大容量化に適したスイッチアーキテクチャとして図1に示すようなスイッチファブリックにクロスバを用いた入出力バッファ

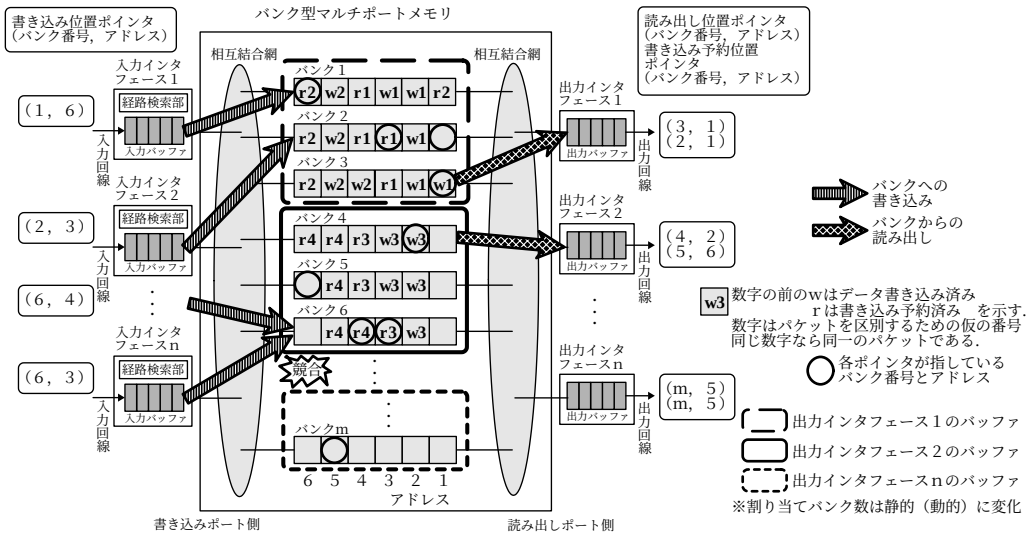

図 2: 提案スイッチの構成図と動作例

型スイッチがある [3]．このスイッチではスイッチに入力されたパケットは入力インタフェースの経路検索部で出力先インタフェースが決定され，スイッチファブリックにより転送される．入力バッファは経路検索やスイッチファブリック内のパスを取り合う際に発生するブロッキングなどのための待ちに使用され，出力バッファはスイッチファブリックと出力回線との速度差を埋めるために使用される．

2.2 既存スイッチアーキテクチャの問題

クロスバを用いた入出力バッファ型スイッチは大容量化に適しているが，いくつか問題点もある．

(a) 出力先が競合するとブロッキングが発生してしまう [4]

(b) 入出力バッファは各インタフェース毎に固定されているため，他のインタフェースでバッファ容量が余っていても容量が不足しているインタフェースへ空きバッファ容量を割り当てることはできない

上記の問題を解決するため, バンク型マルチポートメモリを用いたスイッチアーキテクチャを用いることを提案する. 提案アーキテクチャでは，出力バッファとして使用するバンクを 1 出力インタフェース当たり複数個割り当てることで問題 (a) を解決し，各出力インタフェース毎に割り当てるバンク数を静的，あるいは動的に変化させることで問題 (b) を解決する．次節でその詳細について述べる．

3 バンク型マルチポートメモリを用いたスイッチアーキテクチャ

3.1 基本アーキテクチャ

本スイッチアーキテクチャでは，複数個の 1 ポートメモリからなるバンクで構成されたマルチポートメモリをパケットのスイッチングに使用する．その構成を図 2に示す.

本スイッチアーキテクチャではスイッチに入力されたパケットは入力インタフェースで経路検索を行った後に，バンクでバッファリングされ，出力インタフェースへ転送される．バンクは既存スイッチの出力バッファとして使用し，相互結合網と出力回線の伝送速度差を埋める．出力インタフェースに設けられている出力バッファは相互結合網と出力回線の伝送速度差を埋めるために必要な容量を各出力インタフェースに設ける．

3.1.1 バンクへの書き込み・読み出しの制御方法

次にバンクへの書き込み，読み出しの制御方法について説明する．提案スイッチでは，

- 各入力インタフェースに存在する書き込み位置ポインタ
- 各出力インタフェースに存在する書き込み予約位置ポインタ
- 各出力インタフェースに存在する読み出し位置ポインタ

の 3 種類のポインタを用いて書き込み，読み出しするバンク番号とバンク内アドレスを決定する．各ポインタは図 3に示すように割り当てられているバンク内を□内の数字の順に移動し，y までくると 1 に戻る．以下にパケットの書き込み，読み出しをする際の手順をまとめる．

1. 入力インタフェースからバンクにパケットの書き込み要求があった時，該当出力インタフェースの書き込み予約位置ポインタを参照してパケットの書き込み先頭位置を決定し，そのパケット長に応じて書き込み予約位置ポインタを進める．
2. 同時に入力インタフェースの書き込み位置ポインタをパケットの書き込み先頭位置に設定す

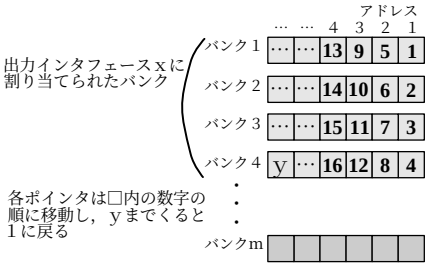

図 3: バンクへの書き込み，読み出しの制御方法

る．入力インタフェースはこの書き込み位置ポインタを参照して，パケットの書き込み位置を決定し，書き込みが行われるごとにポインタを１つずつ進めていく．

3. バンクからのパケットの読み出しには出力インタフェースの読み出し位置ポインタを用いる．出力インタフェースは読み出し位置ポインタが示すアドレスのデータを参照し，パケットが書き込まれていれば読み出しが最優先で行われる．読み出しが行われると，読み出し位置ポインタを１つ進める．

図2に具体的な動作例を示す．この例では各出力インタフェースに3バンク割り当てているので，2つの入力インタフェースから出力インタフェース1に転送されるパケットを書き込むと同時に出力インタフェース1へパケットを読み出すことができている．提案スイッチではこのようにしてブロッキングを防止する．しかし，このサイクルで新たなパケット4の書き込み予約が行われ，出力インタフェース2に割り当てられているバンク6で書き込み競合が発生している．このような場合には，該当出力インタフェースの読み出し位置ポインタからより近い箇所への書き込みを優先する．すなわち，この例では入力インタフェースｎの書き込みが優先される．このような競合に敗れた書き込み要求は１サイクル後に再び同様の要求をすることになる．この例では本バンク制御の特性により次サイクルではバンク5・アドレス2から読み出しを行うと同時に，バンク4・アドレス4とバンク6・アドレス4に書き込みを行うことになり，同一バンクへのアクセス競合はなくなる．

3.2 出力インタフェースへの動的なバンク割り当て手法

本アーキテクチャの特徴として，バンク型マルチポートメモリを用いることでトラヒックの状況に応じて出力インタフェースにバンクを動的に割り当てることができるという利点がある．バンクを静的に割り当てる方法は，例えばどの出力インタフェースにトラヒックが集中しやすいかなどの情報があらかじめわかっていれば，トラヒックの集中することが予想される出力インタフェースにバンクを多く割り当てておくことでパケット損失を緩和することができ有効である．しかしながら，例えばある程度短い時間ごとにトラヒックの集中する出力インタフェースが変化するが，トータルでは均一なトラヒックである場合には，静的なバンク割り当てでは対応することができない．このような場合にはバンクを動的に割り当てる方法が有効である．

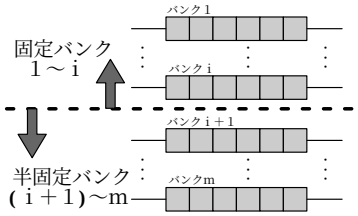

図 4: 動的なバンク割り当ての基本

動的なバンク割り当て手法では，図4のように

- 各出力インタフェースに固定で割り当てるバンク（固定バンク）
- トラヒックに応じて各出力インタフェースに自由に，かつ動的に割り当てるバンク（半固定バンク）

という2種類のバンクを用意しておく．固定バンクを用意しておくのは最低限の性能を確保するためである．本稿ではバンク割り当て・解放の自由度および性能を考慮し，バンク統合型，バンク分離型という2つの動的なバンク割り当て手法を提案する．

3.2.1 バンク統合型動的なバンク割り当て手法

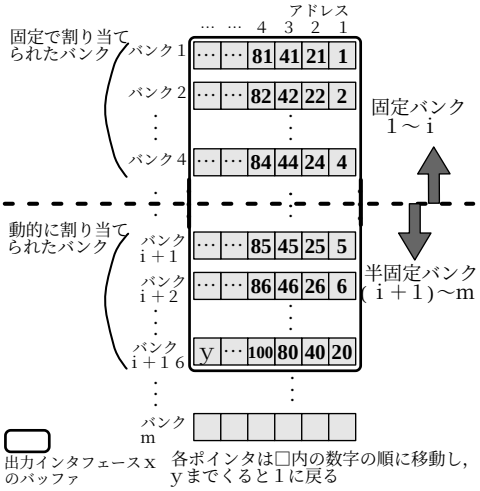

図 5: バンク統合型動的なバンク割り当て手法

バンク統合型動的なバンク割り当て手法は図5のようにポインタを移動させる方法である．この手法では常にバンク使用率を監視し，その値に応じてバンクの割り当て・解放を行う．バンクの割り当てとは割り当てられたバンクへパケットの書き込み予約を行えるようになることを意味し，バンクの解放とは解放されたバンクへのパケットの書き込み予約が行えなくなることを意味する．つまり，バンクの解放を決定してからすぐに解放されたバンクを他の出力インタフェースに割り当てることはできず，バンクからデータがすべて読み出されるのを待たなければならない．

この手法の特徴は静的なバンク割り当て手法と比べて，入力インタフェースからバンクへ同時に書き込みを行える数が多くなるということである．その反面，バンクが動的に割り当てられてから割り当てられたバンク全体が有効に使用されるまでの時間が長くなると考えられる．同様に，バンクの解放を決定してからデータの読み出しが完全に終わるまでの時間も長くなり，半固定バンクの有効使用率が低くなってしまうことが考えられる．

統合型におけるバンク使用率の定義
統合型におけるバンク使用率とは該当出力インタフェースに割り当てられた固定バンクのみの使用率を意味している．すなわち，動的に割り当てられたバンクがあったとしてもそのバンクはここでいうバンク使用率には関係しない．

3.2.2 バンク分離型動的なバンク割り当て手法

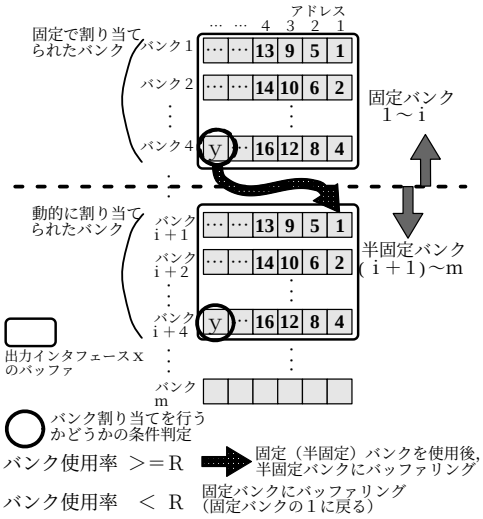

図 6: バンク分離型動的なバンク割り当て手法

バンク分離型動的なバンク割り当て手法は図 6のようにポインタを移動させる方法である．この手法ではバンクの割り当ては図 6に示したように行われる．バンクの解放は動的に割り当てられたバンクの○のところに読み出しポインタが来た時に解放が自動的に行われるため，判定の必要はない．またバンクの割り当て回数に制限はない．

この手法の特徴としては入力インタフェースからバンクへ同時に書き込みを行える数が多くはならないが，バンク統合型の欠点である半固定バンクの使用効率の悪さが改善できることである．

分離型におけるバンク使用率の定義
統合型のバンク使用率とは異なり分離型におけるバンク使用率は割り当てられた固定バンク，半固定バンクすべてのバンク使用率のことを示す．

4 シミュレーション評価

4.1 シミュレーションモデル・評価項目

図 1で示した入出力バッファ型スイッチのスイッチファブリックに共有バス・クロスバ・多段接続網（オメガ網）を用いたものを既存スイッチのシミュレーションモデルとする．提案スイッチのシミュレーションモデルは 3 節で記述した通りである．また評価項目として，スループット・パケット損失率・内部遅延時間を用いた．

4.2 シミュレーション条件

表 1: シミュレーション条件（メモリ容量を除く）

シミュレーションサイクル数	200000 [cycle]
入出力ポート数	32 [port]
入出力回線の伝送速度	32 [bit/cycle]
入力負荷	100 [%]
パケット長	1～12000 [bit] でランダム
パケットの経路検索による遅延	なし
バス（配線）の幅	32 [bit]

表 2: シミュレーション条件（メモリ容量）

既存スイッチ		
1 エントリあたりの入力バッファ容量		
120[Kbit] * 32[port]	=	3840[Kbit]
1 エントリあたりの出力バッファ容量		
120[Kbit] * 32[port]	=	3840[Kbit]
	計	7680[Kbit]
提案スイッチ		
1 エントリあたりの入力バッファ容量		
120[Kbit] * 32[port]	=	3840[Kbit]
1 エントリあたりの出力バッファ容量		
12[Kbit] * 32[port]	=	384[Kbit]
総バンク容量	=	3456[Kbit]
（1 バンクあたりのバンク容量は総バンク容量をバンク数で割った値）		
	計	7680[Kbit]

表 1と表 2にシミュレーション条件を示す．スイッチの純粋な性能比較をするためにパケットの経路検索による遅延はなしとした．表 2に示した通り，公平な比較になるよう既存スイッチと提案スイッチの総メモリ容量は等しくなるように条件設定をした．

また，以下に示す 2 種類のトラヒックを用いて評価を行った．

- 均一なトラヒック

均一なトラヒックは入力されるパケットの出力先インタフェースが均一となるトラヒックである．均一なトラヒックではパケット生成の際に乱数を用いてランダムにパケットの出力先インタフェースを決定する．

- 偏りがあるトラヒック

偏りがあるトラヒックを用いるのはブロッキングが発生しやすい状況での性能比較を行うためである．偏りがあるトラヒックは入力されるパケットの出力先インタフェースがトータルでは均一だが，5000 サイクルごとにパケットの集中する 1 つの出力インタフェースが異なるトラヒックである．このトラヒックでは出力の偏っていないインタフェースを出力先とするパケットの発生確率を 1 とすると，出力の偏っているインタフェースを出力先とするパケッ

トの発生確率が n 倍（以下ではこの n のことを偏りの度合いと呼ぶ）になるよう偏らせた．評価では n が 4 と 8 の場合の 2 通りの偏りがあるトラヒックを用いた．

4.3 提案スイッチ内での比較

4.3.1 バンク数の違いによる比較

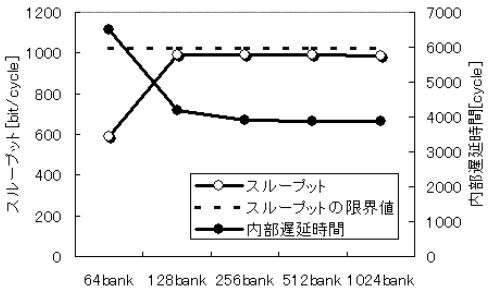

図 7: 静的なバンク割り当て手法のバンク数による性能差（均一なトラヒック）

提案スイッチのバンク数の違いによる比較を静的なバンク割り当て手法を用いて評価した．用いたトラヒックは均一なトラヒックと偏りの度合いが 4 と 8 の偏りがあるトラヒックの計 3 種類である．図 7 のグラフに均一なトラヒックのスループット，内部遅延時間を示す．紙面の都合上割愛したが，他の偏りがある 2 種類のトラヒックを用いた場合も値こそ違うもののほぼ同様の傾向を示した．この結果から 128 バンク（1 出力インタフェース当たり 4 バンク割り当て）あれば性能確保できることがわかった．動的なバンク割り当て手法では性能を確保するためのバンクとして 128 バンクを固定で割り当てるバンクとして用意し，＋ 128 バンクを動的に割り当てるバンクとして用意する．つまり，提案方式はこれ以降，すべて 256 バンクの評価結果を用いることにする．

4.3.2 静的なバンク割り当て手法とバンク統合型・分離型動的なバンク割り当て手法の比較

バンク統合型・分離型動的なバンク割り当て手法のバンク割り当て・解放の条件の違いによる性能比較を偏りの度合いが 4 の偏りのあるトラヒックを用いて行った．また，統合型の今回のシミュレーション評価では制御の簡単化を考えて，すでに動的に割り当てられているバンクが存在している場合，それ以上さらに動的にバンクを割り当てることをしないという限定の下で評価を行った．

図 8に統合型の様々なバンク割り当て・解放の条件下でのスループットと遅延時間を示す（パケット損失率の評価結果は紙面の都合上割愛した）．図 8にある動的に割り当てるバンク数とは図 5でいうと動的に割り当てられたバンクのバンク数のことである．評価結果からはスループット，内部遅延時間ともにバンク割り当て・解放の条件にあまり関係なくほぼ一定の値を示していることがわかる．しかしながら，公平性など様々な観点からみる時にこの評価結果だけでバンク割り当て・解放の条件にまったく関係ないと言うことはできない．

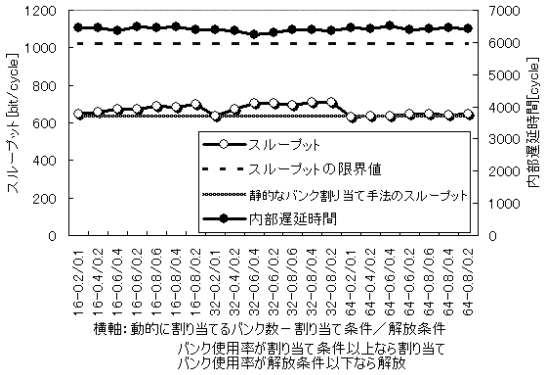

図 8: バンク統合型動的なバンク割り当て手法のバンク割り当て・解放の条件の違いによるスループットと内部遅延時間の比較

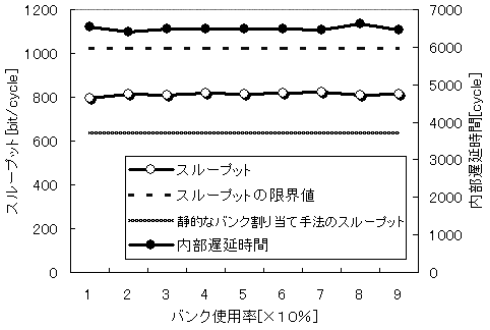

図 9: バンク分離型動的なバンク割り当て手法のバンク割り当ての条件の違いによるスループットと内部遅延時間の比較

次に分離型における性能比較のためのバンク割り当ての条件としてバンク使用率は 10 %～ 90 %まで 10 %刻みで評価した．図 9にその評価結果のスループットと内部遅延時間を示す（パケット損失率の評価結果は紙面の都合上割愛した）．評価結果からは統合型と同様にスループット，内部遅延時間ともにバンク割り当て・解放の条件にあまり関係なくほぼ一定の値を示していることがわかる．統合型の結果は静的なバンク割り当て手法と同じか少し良いかといった結果であったが，分離型は静的なバンク割り当て手法と比べて優れた結果を示している．これは 3.2 節で説明した通り，統合型では半固定バンクの有効使用率が低いことに起因していると考えられる．パケット損失率のグラフは紙面の都合上割愛したが，分離型と比べて統合型のパケット損失率が高くなっていることからもこのことがわかる．それに比べ，分離型では動的に割り当てられたバンクがより有効に使用されており，それが 2 つの動的なバンク割り当て手法の差となって現れている．ただし，統合型でもバンクを細かく割り当てて 1 回以上バンクを動的に割り当てることを許せば，この問題点を改善できるかもしれないという余地を残している．

4.4 既存スイッチと提案スイッチの比較

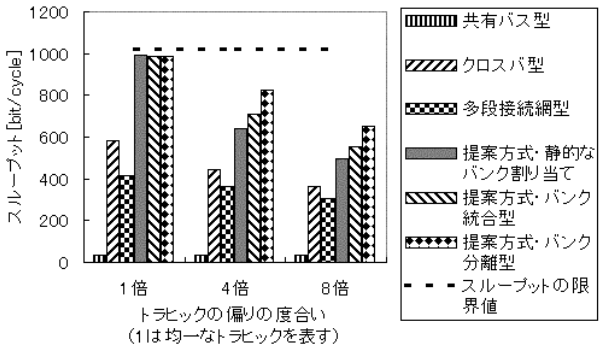

図 10: 既存・提案スイッチのスループット比較

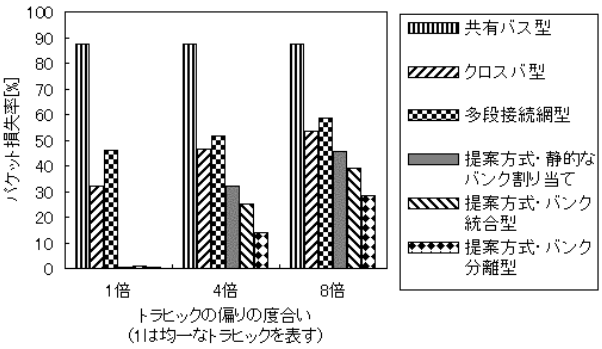

図 11: 既存・提案スイッチのパケット損失率比較

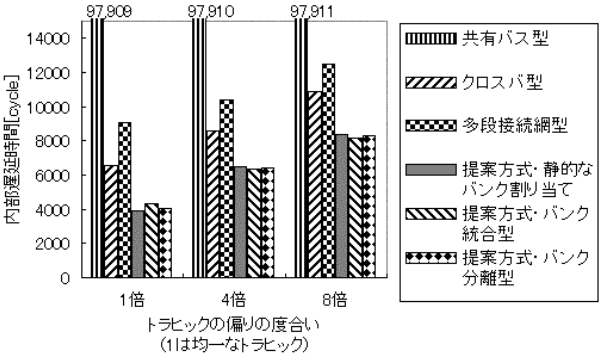

図 12: 既存・提案スイッチの内部遅延時間比較

既存スイッチと提案スイッチの比較を均一なトラヒックと偏りの度合いが 4 と 8 の偏りがあるトラヒックの計 3 種類で評価した．動的なバンク割り当て手法のバンク割り当て・解放の条件として統合型は図 8の 32-0.8/0.2 を用い，分離型は 70% を用いた．スループット比較を図 10に，パケット損失率比較を図 11に，内部遅延時間比較を図 12に示す．図 10，図 11，図 12において静的なバンク割り当て手法とクロスバを用いた既存スイッチを比較すると，入力トラヒックに関わらず，すべての項目において静的なバンク割り当て手法が優れた結果を残していることがわかる．このような結果になるのは静的なバンク割り当て手法ではクロスバを用いた既存スイッチとの最大の違いであるブロッキングの防止（軽減）が実現できているためである．その結果，同時にバッファでの待ち時間も減るため内部遅延時間の減少を実現でき，バッファ使用率も低下するためパケット損失率の低下も達成することができる．

次に，図 10，図 11，図 12において静的なバンク割り当て手法とバンク分離型動的なバンク割り当て手法を比較してみると，分離型が少なくとも同程度か優れた値を示していることがわかる．これは動的なバンク割り当て手法のトラヒック状況に応じて迅速にバンクを動的に割り当てることができるという利点によるものである．しかし，図 12において内部遅延時間は分離型も静的なバンク割り当て手法と同程度である．分離型ではバンクを動的に割り当てることでパケット損失を減少させることができると同時に，より多くのパケットを転送することが可能になるためさらなるスループットの向上にもつながる．しかしながら，バンクを動的に割り当てることでパケットの待ち行列も長くなってしまうため，必ずしも内部遅延時間の減少にはつながらない．

5 まとめ

本稿では既存スイッチの問題を解決することができるバンク型マルチポートメモリを用いたスイッチアーキテクチャの提案とシミュレーション評価を行った．シミュレーション評価により分離型動的なバンク割り当て手法を用いることでクロスバを用いた既存スイッチと比べて最大で約 85% のスループット向上を実現した．同時に，パケット損失率，内部遅延時間の低下も実現できることがわかった．今後の課題として，より実トラヒックに近いトラヒックを用いてのシミュレーション評価， VOQ を実装したクロスバスイッチとのシミュレーション比較や既存スイッチと提案スイッチのハードウェア量評価，提案スイッチにおける QoS[5] の実現方法を検討することなどが挙げられる．

謝辞 本研究の一部は半導体理工学研究センターとの共同研究 “大きなランダムアクセスバンド幅を持つスーパコンパクト・マルチポートメモリ，及びそれを用いたシステム・オン・チップ／パッケージ向け高性能アプリケーション” による．

参考文献

[1] “平成 14 年版情報通信白書”，総務省
[2] 齊藤，相田，青木，西村，新保:“多段接続網を用いた大容量 IP データグラムスイッチング方式”，信学論，B，Vol.J83-B，No.11，pp.1545-1553，2000.
[3] 森脇，豊田，高瀬，遠藤，三村:“次世代大規模 IP ネットワーク向けノードシステム”，信学技法 SSE2000-223，pp.79-84，2001.
[4] 天野英晴:“並列コンピュータ”，昭晃堂，1996.
[5] 阪田史郎:“インターネットにおける QoS 制御”，信学誌, Vol.85, No.10,pp.749-755, 2002.