

ダイナミックリコンフィギュレーションによる 高速パケット転送処理回路の小型化

A study of dynamic reconfiguration in high-speed packet processing

山田 博希† 甲斐 英則† 片山 勝†
Hiroki Yamada Hidenori Kai Masaru Katayama

1. はじめに

高速パケット転送処理回路の小型化を狙いとし、マルチコンテキスト型リコンフィギュラブルデバイスを用いて分割したタスクの回路記述データを動的に切り替えるダイナミックリコンフィギュレーションについて述べている。リアルタイム系のパケット処理においてダイナミックリコンフィギュレーションを行う際に課題となるスループット劣化について検討し、軽減する方法を提案、評価する。

2. 背景

IT化が進む中、インターネットの普及によるトラフィックの急激な増大と同時にユーザ要求の多様化、高度化などから、ネットワークおよびネットワークに接続されたさまざまな機器に対して、より高速なパケット処理機能と同時に機能変更の柔軟性に対する要求が高まっている。

このような高速性と機能変更の柔軟性の要求を両立できる技術としてリコンフィギュラブル技術のネットワーク機器への適用が検討されており、アイピーフレックス社のDAPDNA-2など、数十ギガビットクラスの高速度転送処理が可能で且つ回路記述データを基に瞬時に回路変更できる高性能なリコンフィギュラブルデバイスも登場している。

今後、高速性と機能変更の柔軟性が特に望まれるパケット処理機能の例としては、ファイアウォールやTCP/IPSec処理等のオフロードなどが考えられる。これらの機能を持つパケット転送処理回路をリコンフィギュラブルデバイスで構成しようとした時、リコンフィギュラブルデバイスの機能集積規模はASICに比べて低くデバイスが複数個必要になる等からパケット転送処理回路の実装規模が大きくなるという問題がある。

一方、小型化の方法として、必要に応じて回路を動的に変更するダイナミックリコンフィギュレーションが有効と考えられる。例えば、パケット毎に異なる処理(ex. 上り/下りパケット、IPv4/v6パケット)に対応してリコンフィギュレーションすれば、回路規模の削減が可能である。

3. 高速パケット転送処理

本稿で論ずるパケット処理はルータやサーバの回線インタフェースボードなどで行うパケット識別・転送処理やファイアウォールなど高機能パケット処理である。これらの機能を持つボードでは、複数のマルチコンテキスト型のリコンフィギュラブルデバイス(以降、エレメントと記す)を搭載し、各エレメントの内部に備えた複数面のコンフィギュレーションメモリを切り替えることにより、瞬時に機能を変更することが可能となる。リコンフィギュラブルデバイスの高速性とパケット処理の柔軟性(ヘッダの抽出位置の変更、高位レイヤ情報の処理、など)を生かして、各エレメント間は、パケットインタフェース(インライン型)で多段シリアル接続され、データフロー型のパイプライン処理を行うことを前提とする(図1参照)。ここで、本稿でのパケット転送処理回路の小型化では、ボードに搭載されるエレメントの個数をダイナミックリコンフィギュレーションにより削減することを狙っている。

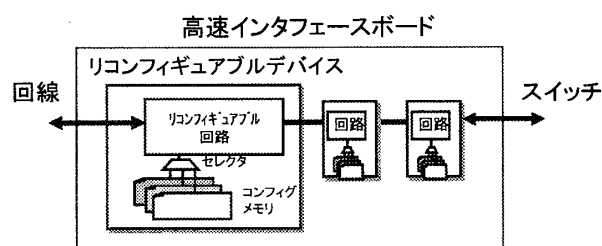


図1 ルータの高速インタフェースボードの例

4. ダイナミックリコンフィギュレーション

パケット転送処理回路の小型化のために行うダイナミックリコンフィギュレーションとしては、図2に示すようにタスクの分割の仕方に応じてブランチタスク型とシリアルタスク型が考えられる。ブランチタスク型は、パケット毎の分類結果などから所要のタスクに対応するコンフィギュレーションデータを選択し回路を再構成することにより分岐構造のタスク処理を構成する方法である。一方、シリアルタスク型は、フィードバックループを用い再入力される度に回路を再構成することにより連続したタスク処理を構成する方法である。ブランチタスク型は、タスクの分割数、シリアルタスク型は、ループ回数に対応して回路の共用による小型化が可能であり、例えば、図2において、Task A~Task Dを計4つのエレメントで構成していたと仮定すると、ダイナミックリコンフィギュレーションを行うことにより、1つのエレメントで構成できることになる。

† (社) 電子情報通信学会, IEICE

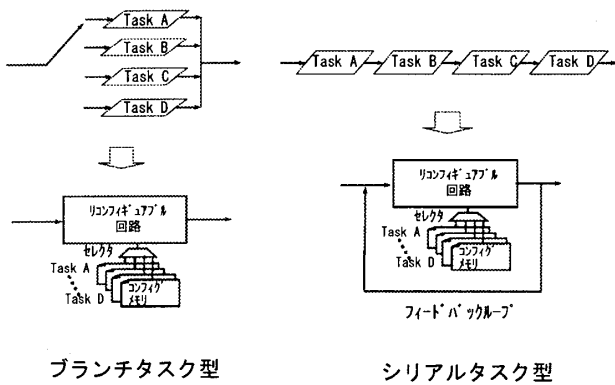


図2 タスク分割による回路の小型化

5. スループットの劣化

リコンフィギュレーションする際の回路の切り替え自体については、1クロック (nsec オーダ) で可能なものが現在入手できる。しかしながら、回路変更時にパケット紛失・データ化けを起こさない(無瞬断)ためには、回路へのパケットの入力を止めるとともに回路中の処理中パケットの出力を待ってリコンフィギュレーションを行う必要がある。回路中のパケットが全て出力するまでの時間は、タスクの処理遅延(レイテンシ)分必要であり、エレメント規模にもよるが数百クロック以上が想定できる。従って、リコンフィギュレーションによるスループットへの影響については、リコンフィギュレーションの頻度とともにレイテンシが重要なファクタである。また、フィードバックループを用いたシリアルタスク処理の場合、回路を時分割利用しているためループの回数に比例してスループットの劣化が発生する。

5.1 ブランチタスク型

(1) ブランチタスク型のスループット劣化

図3にブランチタスク型の実現例を示す。リコンフィギュラブル回路、切替判定回路及びコンフィギュレーションメモリで構成され、切替判定回路からの指示により、必要に応じて入力パケット毎にコンフィギュレーションメモリを選択し所要の回路にリコンフィギュレーションする。

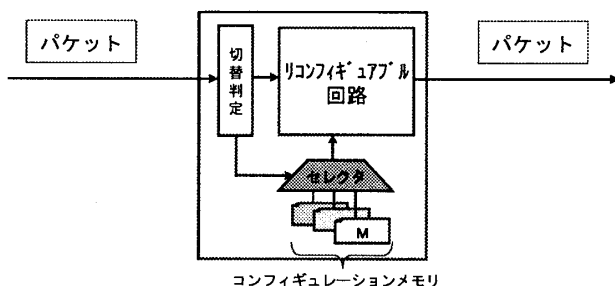


図3 ブランチタスク型の実現例

リコンフィギュレーションを無瞬断で行うため参考文献[1]で述べた方式(以降、シングルバッファ方式と記す)を図4に示す。無瞬断リコンフィギュレーションを行うために、バッファを用いて空きタイミング(=エレメントのレイテンシ)を生成する。ここで、シングルバッファ方式におけるスループット劣化率 α_1 は、平均パケット長を K (クロック)、エレメントのレイテンシを D (クロック)、タスクの分岐数(=コンフィギュレーションメモリ面数)を M とした場合、(式1)で表すことができる。ここで、タスク毎の分岐確率はタスクによらず全て等しいと仮定している。

$$\alpha_1 = \frac{1}{M} + \frac{M-1}{M} \left(\frac{K}{K+D} \right) \cdots \text{(式1)}$$

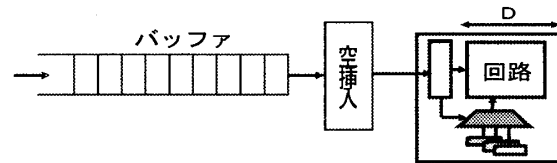


図4 シングルバッファ方式

(2) マルチバッファ方式と改善効果

マルチバッファ方式を図5に示す。本方式は、タスクの分岐数に対応した M 面のバッファを並列に2面(A/B面)設け、一定時間(フレーム)間隔毎に同一処理となるパケットを集めることによりリコンフィギュレーション回数を削減する方式である。ここで、本方式におけるスループット劣化率 α_2 は、フレームを $\beta \times D$ とした場合、同一フレーム内のリコンフィギュア回数は最大 $M-1$ 回のため、(式2)で表すことができる。

$$\alpha_2 = \frac{\beta}{\beta + (M-1)} \cdots \text{(式2)}$$

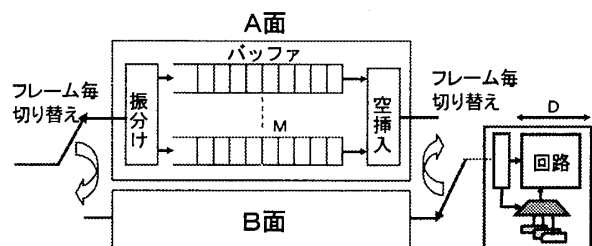
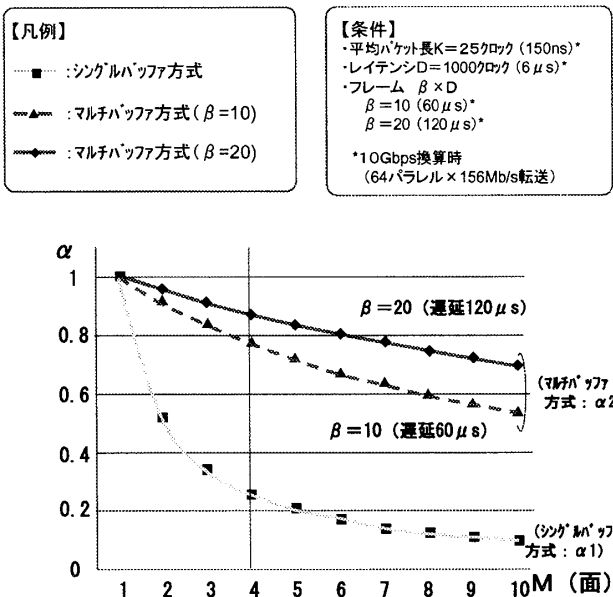


図5 マルチバッファ方式

図6にシングル方式及びマルチバッファ方式のスループット劣化率 α_1 、 α_2 を示す。シングルバッファ方式は、 M の増加によりスループットは急激に劣化する($M=4$ で0.25に劣化)が、マルチバッファ方式は、劣化の影響を大きく改善できる($M=4, \beta=10$ で0.77、 $\beta=20$ で0.87)。ここで、マルチバッファ方式では、フレームやレイテンシに比例して必要なバッファ量と遅延時間が増加するが、バッファ容量は、エレメントの規模に比べると十分小さく(例えば、 $M=4, \beta=20, D=1000$ クロックで0.7Mbyte $\times 2$)、遅延時間も10Gbps程度の高速回線では影響は小さい($\beta=20$ で120 μ s 参考: ネットワーク遅延: QoS クラス 0=100ms)。

図6 ブランチタスク型のスループット劣化率 (α)

5.2 シリアルタスク型

フィードバックループを用いたシリアルタスク型の実現例として、ループドパイプラインリコンフィギュレーション方式[2]を図7に示す。エレメント数を S 、ループ回数 (=コンフィギュレーションメモリ数) が M で、タスク分割数は、 $S \times M$ である。また、スループット劣化を補うために内部の転送処理速度を回線速度よりアップ可能な構成としている。なお、 $1/M$ がエレメント数の削減率となる。ここで、複数回のループを経て一連のタスク処理が行われるパケットの処理単位を $\gamma \times D$ (クロック) とおくと、スループットの劣化率 α は、(式3)で表わされる。すなわち、回線速度と同一のスループットを保障するために必要な内部転送速度のアップ率は $1/\alpha$ である。

$$\alpha = \gamma / M(\gamma + S) \quad (\text{式3})$$

ここで、リコンフィギュレーションをエレメント一つ分のレイテンシに相当する空きを用いてパイプライン動作で順次行う (パイプラインリコンフィギュレーション: 図8参照) とすると、挿入する空き時間を短縮できるためスループット劣化を改善できる。すなわち、(式3)において $S=1$ と等価となる。

図9に、スループット劣化特性を示す。図9より、 $\gamma=15$ でパイプラインリコンフィギュレーション時 ($S=1$) の場合、スループット劣化率は、ほぼ $1/M$ の特性 (ループを用いた時分割処理による劣化分) になることがわかる。 γ は、前述のブランチタスク処理における β に相当しており、値 ($\gamma=15$) においても、バッファ量、遅延時間に与える影響は少ない。

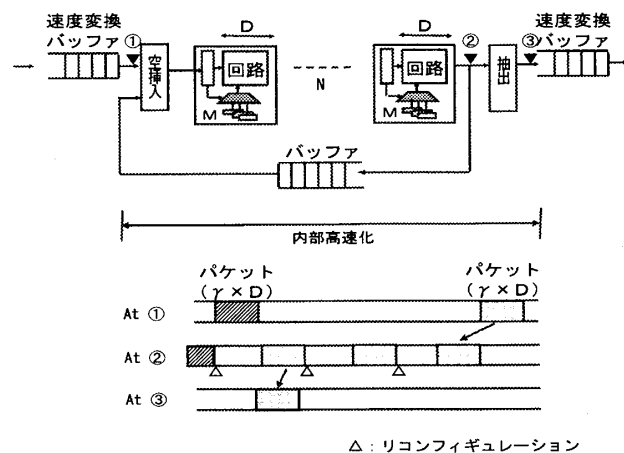


図7 ループドパイプラインリコンフィギュレーション方式

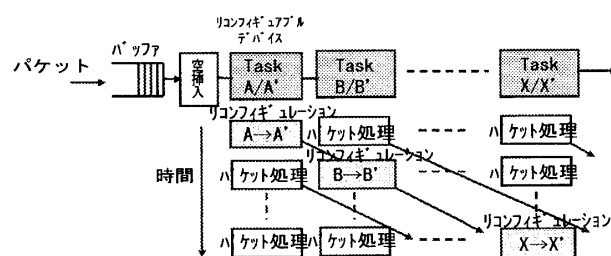
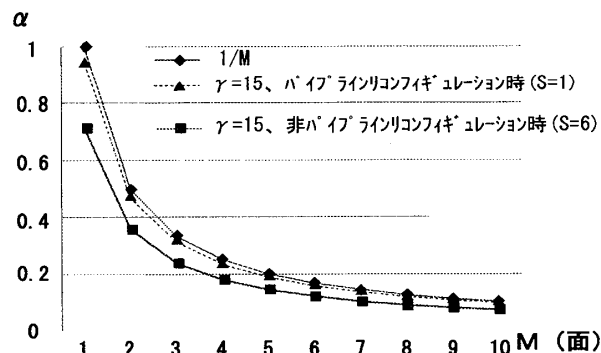


図8 パイプラインリコンフィギュレーション

図9 シリアルタスク型のスループット劣化率 (α)

6. おわりに

ダイナミックリコンフィギュレーションを用いた高速パケット転送処理回路の小型化に関する検討を行った。タスクの分割の仕方に応じて、ブランチタスク型とシリアルタスク型に分類し、それぞれについてリコンフィギュレーションによるスループット劣化特性を明らかにするとともに軽減法を提案し、その改善効果を示した。

文献

- [1] 甲斐、山田: “パケット転送処理における無断断リコンフィギュレーション法の検討” 電子情報通信学会/サティ大会 B-6-150 (2003)
- [2] H. Yamada, H. Kai and J. Yoshida, “Using re-configurable technology in high-speed packet processing” ICNO4, February 2004.