

## 新世代キャッシュ搭載ネットワークプロセッサにおける キャッシュアルゴリズムの提案

八巻 隼人† 西 宏章‡

### 1. 結論

近年, Twitter や Skype といった大量の小さなパケットにより構成される豊かな web サービス・コンテンツが増加し, ネットワークトラフィックが爆発的に増加している. このようなサービス・コンテンツは今後も増加することが予想されており, これら膨大なトラフィックがバックボーンネットワークルータに集中すると, ルータのパケット処理能力が限界を超える可能性が否めない. このような問題に対し, キャッシュ機構を持つネットワークプロセッサ(NP)の研究が行われてきた[1,2]. 既存のキャッシュ搭載 NP は, パケット処理において比較的执行時間を要するルーティングテーブル検索に特化し, 宛先 IP アドレスをキーとしてそのテーブル検索結果をキャッシュ, 以降のパケットに適用することで処理の効率化を行っている. しかし, 近年のルータはルーティングテーブル検索のみならず, カプセリングなど, ボトルネックとなりうる様々な処理が要求される. そこで我々は, 5 タプル情報(Src IP・Port, Dst IP・Port, Protocol)を用いてより柔軟に処理に対応する新世代キャッシュ搭載ネットワークプロセッサ P-Gear を提案している[3]. 本ポスターでは P-Gear におけるキャッシュミス率を改善するアルゴリズムの提案を行う.

### 2. P-Gear

P-Gear は, 5 タプルにより定義される同一セッションのパケットが NP において同一方法で処理されるという事実を基に, セッション情報をキーにその処理結果をキャッシュし, 以降のパケットに適用する. 図 1 に P-Gear の構成を示す. 到着したパケットは Analysis-Engine において必要な内部情報列が抽出され, C-Engine によりキャッシュエントリの参照が行われる. エントリに一致した場合には R-Engine において, キャッシュ内の処理結果を適用することで PU を介さず処理を行う. このパイ

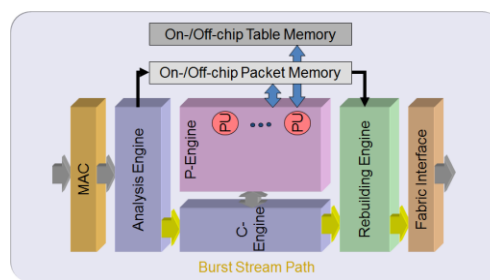


図 1 P-Gear の構成

ライン構造を Burst-Stream Path と呼び, ワイヤレートで処理を行う. 一方でキャッシュミスした場合には, P-Engine において集積化された PU 群により処理を行う. 従って, PU の負担を減らすためにはキャッシュミス率の削減が不可欠であり, キャッシュアルゴリズムを改善する重要性が高い.

### 3. キャッシュアルゴリズム

P-Gear はセッション毎にエントリを持つため, 多数パケットで構成されるセッションに対するエントリの割合が高いことが望ましく, 少数パケットで構成されるセッションに対するエントリの価値は低い. 特に 1 パケットのみのセッションは, キャッシュヒットすることがなく, 登録する必要がない. 図 2 に示すように, トラフィック中ではこのようなセッションが全体の 7 割を占めており, キャッシュミス率に与える影響が大きいと考えられる. そこで, 少数パケットからなるセッションのエントリを排除するアルゴリズムについて, キャッシュ登録・キャッシュ追い出しの観点からそれぞれ提案する.

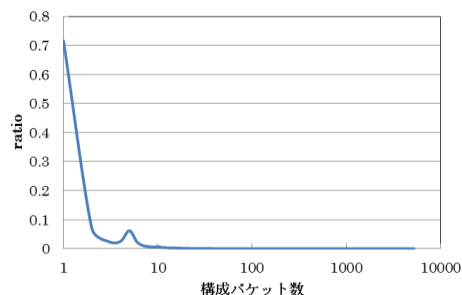


図 2 セッションの構成パケット数の分布

† 慶應義塾大学大学院理工学研究科

‡ 慶應義塾大学システムデザイン工学科

### 3.1 キャッシュ登録アルゴリズム

キャッシュ登録アルゴリズムでは、特定のセッションに対しエントリ登録を拒否することを事前に決定する。ここではいくつかある登録アルゴリズムの中から、特に DNS に焦点を当てた DNS-Aware Cache について述べる。DNS は 1 パケットの要求セッションと応答セッションで構成されるため、キャッシュ登録は無駄である。従い DNS で用いられる Port53 を持つパケットに対しキャッシュ登録をしないことでエントリの有効活用が可能である。図 3 に示すように、DNS-Aware Cache はいずれのエントリサイズにおいてもミス率を改善でき、2,048 エントリの時に最大でミス率の 7% を改善できる。

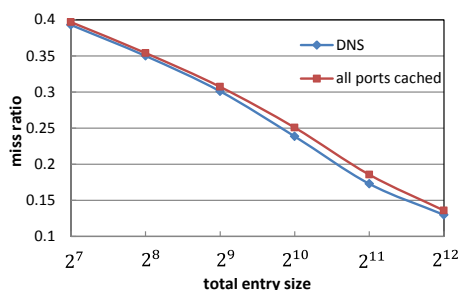


図 3 DNS-Aware Cache におけるキャッシュミス率の比較

### 3.2 キャッシュ追い出しアルゴリズム

P-Gear のキャッシュは 4-way セットアソシアティブで構成される。追い出しアルゴリズムでは、このキャッシュラインが溢れた時に削除すべきエントリを決定する。従来、追い出しアルゴリズムでは一般的に Least Recently Used (LRU) が用いられてきた。しかしながら、図 4 に示すように LRU ではエントリが追い出されるまでに最低でも way 数回のエントリ入れ替えが必要となる。P-Gear において、少数パケットで構成されるセッションは早く追い出されるべきであり、LRU は最適ではない。そこで、Last Storing Cache ではエントリ登録時の初期位置を最も優先度が低い位置に設定することで、少数パケットによるセッションのエントリを追い出す。エントリが参照された場合には優先度が最も高い位置へと送る。また、Elevator Cache も同様にエントリ登録の初期位置を最も優先度が低い位置に設定するが、参照があった場合には、1 つ上の優先度の位置へと送る。シミュレーションにおいて図 5 に示すように、Last Storing Cache, Elevator

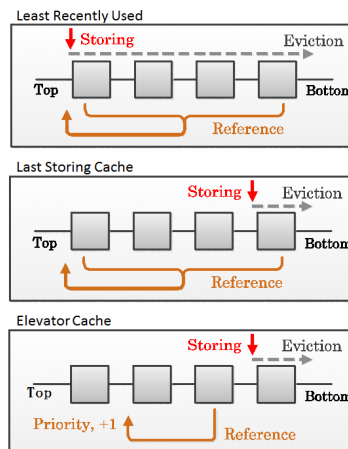


図 4 各追い出しアルゴリズムの構造

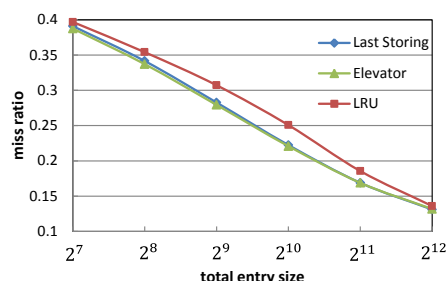


図 5 各追い出しアルゴリズムにおけるキャッシュミス率の比較

Cache 共に 1,024 エントリの時に LRU に比べ最大でミス率の 12% を改善できる。僅かながら Elevator Cache のほうが P-Gear において効果的である。

## 4. 結論

新世代キャッシュ搭載ネットワークプロセッサ P-Gear に対し、少数パケットからなるセッションに着目したキャッシュアルゴリズムを提案した。本手法により 7% 以上のミス率改善が可能である。

### 参考文献

- [1] Tzi Cker Chiueh and Prashant Pradhan, "High-Performance IP Routing TableLookup Using CPU Caching," In Proceedings of INFOCOM'99, pp.1421-1428, 1999.
- [2] Bryan Talbot, Timothy Sherwood and Bill Lin. "IP Caching for Terabit SpeedRouters," In Proceedings of Global Communication Conference (Globecom'99), pp.1565-1569, 1999.
- [3] 西宏章, 奥野通貴, "次世代インターネットを支えるネットワークプロセッサアーキテクチャ P-Gear の提案," 電子情報通信学会 CS 研究会 CS2003-96, pp. 81-84, Sep. 2003.