M-15 データ駆動型最長一致検索機構の超高速パケット分類処理への応用
High-Speed Packet Classification Using Data-Driven LPM Mechanism

森川 大智† 岩田 誠†
Daichi Morikawa Makoto Iwata

1. はじめに

多様な情報通信サービスに柔軟に対応可能なネットワーク装置の実現には，プログラム可能な超高速ネットワークプロセッサ(NPU)が必須となってきている．筆者らも,パイプライン処理能力に優れたデータ駆動型マルチプロセッサ[1]をパケット・ルータ向きに最適化した，データ駆動型ネットワークプロセッサ(DDNP)の構成法を検討している．既に，NPU のボトルネックとされている IP アドレス検索処理を 50M 検索/秒で実現できる機構も提案している[2].

本稿では，この検索機構を応用して DDNP のパイプライン処理能力を発揮させれば，超高速にパケット分類処理を実現できることを明らかにする．

2. パケット分類処理

2. 1 パケット分類

パケット分類は，入力されるパケットの参照すべきフィールド（キーフィールド）と，各キーフィールドが満たすべき条件を記述したルール（フィルタ）を照合し，すべてのキーフィールドがフィルタの条件を満たすようなフィルタ群を，フィルタの順序つき集合であるフィルタセットから選出する関数である．本稿では，選出フィルタ群からフィルタを 1 つ取り出す場合には，フィルタセットにおいてルールが記述順に順序付けられていることを想定しており，ルールを検索結果として出力する．

2. 2 照合法

パケット分類において，ユーザが定義可能な照合法は以下の 3 通りである．

(a) 完全一致照合, (b)プレフィックス照合, (c) 領域照合

これらの照合法において，完全一致照合は照合するキーフィールドと同一の長さのプレフィックスとの照合とみなせるため，プレフィックス照合の特殊な場合として取り扱える．また，領域照合については，たとえば，$8 \leqq x \leqq 11$ は[8, 11] = [1000, 1011] = prefix(10*), $x \leqq 1023$ は[, 1023] = [, 0000001111111111] = prefix(000000*)として表現できる．さらに $1023 < x$ は [1024,] = [0000010000000000,] = prefix(000001*), prefix(00001*), (0001*), (001*), (01*), (1*)として表現可能であり，これもプレフィックス照合として取り扱える．以上より，各フィルタとの照合を全てプレフィックス照合とみなし，IP アドレス検索と同様の最長一致処理を適用すれば，パケット分類処理が実現できる．

3.パケット分類のパイプライン型実現法

3. 1 概要

n 個のキーフィールドを対象とした n 次元のパケット分類では，原則，キーフィールド毎に並列にフィルタ照合を行い，最後に各照合結果を統合すれば，ターンアラウンド時間が最短となる.しかし，次元数の増加に伴い探索空間が膨大となり，単純に同時並列処理を行うと膨大な記憶空間が必要となる.このため，各キーフィールドの部分照合を並列に実行し，徐々に探索空間を縮小しながら検索する再帰的分類方式が提案されている[3]. この方式でも 1 回の照合に必要なデータが冗長になり，記憶空間の利用効率が悪い．

本稿に提案する方式は，各フィールドに対する照合を流れ作業的にパイプライン並列に順次実行し，高速化を図る方式である.したがって，検索が進むにつれて探索範囲を徐々に限定でき，記憶空間を最小限に抑えられる．さらに，各フィールドに対するプレフィックス照合をレベル圧縮トライ(LC-Trie)木検索処理[4]により実現することによって，これらの処理を DDNP の IP アドレス検索機構[2]を用いて高速化すると同時に，メモリ参照回数を最小限に抑えている．

ただし，フィルタセットの大規模化に伴って，フィルタに偏りがあると，検索木がアンバランスになってしまう．そこで，各フィルタ条件の先頭数ビットを基に検索木を分割する手法[5]を導入している．分割した検索木は図 1 に示すように Index Jump (I-J) テーブルに格納され，サブツリーおよび各次元のバランスを改善できる．

3. 2 分類アルゴリズム

分割時に使用した先頭ビットは，各検索木を示す I-J テーブルのアドレスとして使用し，パケット分類時には，まず入力パケットのキーフィールドから同テーブルのアドレスを計算し，グループを選択，そこから得られる検索木に対して最長一致検索処理を行う．すなわち，Index Jump 部とツリー検索部の 2 つのモジュールで構成し,ツリー検索部は，LC-Trie 検索部と Prefix 検査部から構成する（図 2）．各モジュールはメモリアクセスを伴う処理単位で構成され，これらのパイプライン並列処理によってメモリアクセス遅延を隠蔽し，メモリアクセスレートと等価な最大検索レートを実現している．

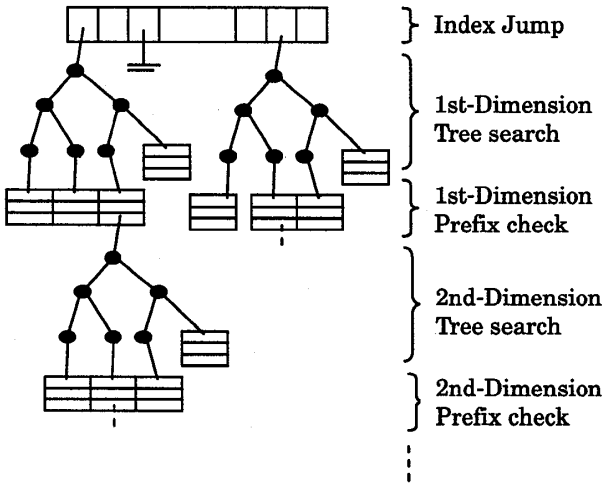

図 1 検索データの階層構造

† 高知工科大学大学院 基盤工学専攻 情報システム工学コース

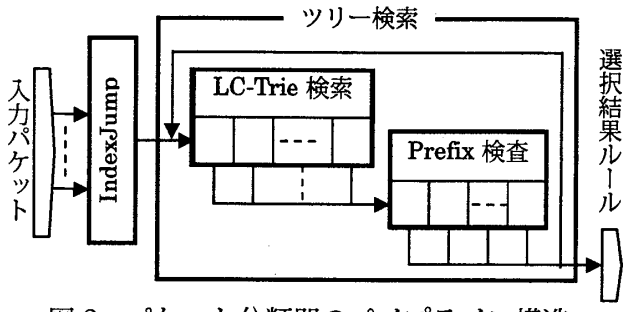

図 2　パケット分類器のパイプライン構造

(a) Index Jump 部：入力パケットのキーフィールドを基に本テーブルを参照し，該当する検索木へのポインタを出力する．Index Jump に使用されるアドレスは図 3 に示すように，各キーフィールドの先頭 x ビットを取り出し，これらのビットを連結した値である．

(b) ツリー検索部： Index Jump 部において選択された検索木を検索する部分であり，LC-Trie 検索アルゴリズムを用いている．これによって，検索木を深さ方向に圧縮して，平均メモリアクセス回数を極小化できる．図 2に示すように，本モジュールは LC-Trie 木を探索する LC-Trie 検索部と，木探索の結果を検査する Prefix 検査部からなる．これらをパイプライン並列に次元数だけ繰り返して，検索処理を行う．

3.3　Index 選択

I-J テーブルのエントリ数は，選択された各フィールドのビット長 x_i ($1 \leq i \leq n$)により決まり，$O(2^{x_i})$で線形に増加する．そのため，適切なビット選択が必要である．

平均的に分散したフィルタセットの場合，選択ビット数にかかわらず平均的に I-J テーブルに割り当てられる．しかし，実際のフィルタセットではかなりの偏りが予想される．実際に，Mae-West のルーティングテーブル[6]を基に 4 次元のフィルタセットを作成し，先頭 2 つのキーフィールドを宛先／送り元 IP アドレスとし，残りを宛先／送り元ポートと仮定して，同様の操作を行ったところ，I-J テーブルのエントリ数が増加しても，フィルタ群は特定エントリに集中した．このことから，フィルタセットの傾向に対し，ヒューリスティックに I-J テーブルに必要最小限のリソース消費で実現可能なビット数を設定する必要がある．

現時点では，生成したフィルタセットの各アドレスフィールドに 4 ビット，各ポートフィールドに 2 ビットを適用し I-J テーブルを生成すると，ルール群に対して，最大サイズとなる検索木でも，分割前の約 10 分の 1 となり，実用的な検索木が生成できることを確認した．

また，ローカルサイトやホームゲートウェイなどの比較的小規模な網においては大多数のフィルタの選択ビットがすべて同じ値を示すことが多い.これでは I-J テーブル内の有効データ数が少なくなり，Index Jump の効果が薄れる.このような場合には，偏りのあるフィールド条件を 2 つのサブフィールド条件に分割する，つまり次元を（N+1）に拡張すれば，Index Jump テーブルの分散を大きくできると考えられる．

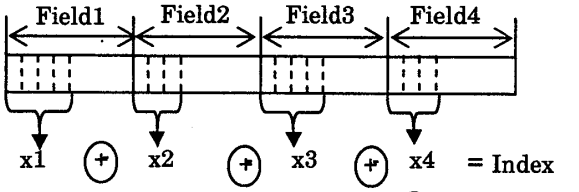

図 3　Index 生成式

4．考察

本手法において，1 入力パケットにつきメモリアクセス回数は，Index Jump 部で 1 回，ツリー検索部で L 回必要である．ただし，LC-Trie 木の深さは O（log log n）で増加するため，L=log log n である．キーフィールドの数を M，フィルタの各条件を木構造に写像するために必要なノード数を n_i ($1 < i < L$)とすると，パケット分類処理に必要なメモリアクセス回数 N は最悪の場合，$N = 1 + \Sigma_{1<i<M}(\log\log n_i)$であり，検索時間は $O(M \cdot \log\log n)$となる．

本手法の評価を行うにあたり，Mae-West のルーティングテーブルから作成した 4 次元のフィルタセット 110K エントリを対象に，現在設計中の DDNP 上で動作させたときの見積もり結果を表 1 に示す．

表 1. 提案手法の性能見積もり結果

I-J テーブル容量	2	256	4096
分割フィルタ群	78846	14861	6882
分類レート	11M	12M	12M

分割フィルタ群：FilterSet/IndexJumpEntrie

分類レート: Packets/sec

上記の結果から，フィルタセットをバランス良くグループに分割できるため，平均的には約 24Gbps 程度の IPv4 パケットストリームを分類可能なことを確認した．

5．むすび

本稿では,データ駆動型プロセッサの特徴であるパイプライン並列処理の徹底的な活用によって，少ない記憶量で高速にパケット分類処理を実現する手法を提案した．本手法では，フィルタセットをヒューリスティックに分割し，1 検索木あたりのデータ量を極小化することで，巨大なルール群に対しても，効率的なパケット分類が可能である．また，IPv6 のようにキーフィールド長が長いものに関しても，キーフィールドを分割し，本手法を適用することにより，同等の性能が得られると考える．今後は試作中の評価システムを用いて各種の実証的性能評価を行う予定である．

参考文献

[1]H. Terada, et al, "DDMP's: Self-timed super-pipelined data-driven multimedia processors," Proc.of the IEEE, 87(2), 1999.

[2] D. Morikawa, et al, "Superpipelined IP-Address Lookups on a Data-Driven Network Processor", PDCS 2001.

[3] Pankaj Gupta, et al., "Algorithms for Packet Classification", IEEE Network, 15(2) , 24-32, 2001.

[4] S. Nilsson and G. Larlsson, IP-address lookup using LC-tries, IEEE J. on Selected Areas in Comm., 17(6), 1999.

[5] Thomas Y. C. Woo, "A Modular Approach to Packet Classification: Algorithms and Results", INFOCOM 2000

[6] IPMA Project - http://www.merit.edu/ipma/