

低コスト・低消費電力 TCAM における 効率的なルーティングテーブル管理法

阿多 信吾[†] 黄 恵聖^{††} 山本 耕次^{†††} 井上 一成^{†††} 村田 正幸^{††}

[†] 大阪市立大学 大学院工学研究科 〒 558-8585 大阪市住吉区杉本 3-3-138

^{††} 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{†††} 株式会社ルネサステクノロジ 〒 664-0005 兵庫県伊丹市瑞原 4-1

E-mail: [†]ata@info.eng.osaka-cu.ac.jp, ^{††}{h-hwang,murata}@ist.osaka-u.ac.jp,

^{†††}{yamamoto.koji4,inoue.kazunari}@renesas.com

あらまし 現在 IP ルータで高速アドレス検索に使用されている TCAM (Ternary CAM) には、消費電力、コスト、および容量などの制約があり、これらを解決するための新しい TCAM ハードウェアが求められている。本稿では低コスト・低消費電力を実現する新しい TCAM アーキテクチャにおいて効率的にルーティングテーブル情報を格納するための手法について検討する。

キーワード IP ルータ, アドレス検索, TCAM, 電力消費, ビット位置選択

Managment of Routing Table in TCAM for Reducing Cost and Power Consumption

Shingo ATA[†], Haesung HWANG^{††}, Koji YAMAMOTO^{†††},

Kazunari INOUE^{†††}, and Masayuki MURATA^{††}

[†] Graduate School of Engineering, Osaka City University

^{††} Graduate School of Information Science and Technology, Osaka University

^{†††} Renesas Technology Corporation

E-mail: [†]ata@info.eng.osaka-cu.ac.jp, ^{††}{h-hwang,murata}@ist.osaka-u.ac.jp,

^{†††}{yamamoto.koji4,inoue.kazunari}@renesas.com

Abstract TCAM (Ternary Content Addressable Memory) is widely used for high speed address lookup. However, TCAM has a potential problem on hardware cost and energy consumption, which limits to deploy large amount of capacity in IP routers. In this paper, we propose a new TCAM-based hardware architecture which reduces significantly power consumption. The proposed architecture consists of multiple sub TCAMs and a indexing table. We also propose a method to select appropriate bit positions used for indexing based on the statistical characteristics of routing table. Our numerical results show that the proposed method can divide the whole routing table nearly equal to sub TCAMs.

Key words IP Router, Address lookup, Ternary Content Addressable Memory (TCAM), Power consumption, Bit position selection

1. はじめに

現在、インターネットトラフィックを中継処理するルータでは、TCAM (Ternary Content Addressable Memory) が幅広く用いられている。具体的には、パケットの転送処理を行うためのルーティングテーブル検索、パケットをポリシーに基づき通過・

棄却を行うアクセス制御、トラフィック統計情報を取得するためのフロー情報の格納などである。

ルーティングテーブル検索は、宛先ネットワークと出力ポートを関連づけたルーティングテーブルをルータに構築し、パケット到着ごとにパケットの宛先アドレスがテーブルのどのエントリーに該当するかを検索する。アクセス制御では、ACL (Access

Control List) と呼ばれる制御ルールを記述し、パケットがどのルールに該当するかを検索し、該当するルールで指定された制御法にもとづき、パケットを通過、棄却、あるいはレート制御などを行う。また、トラフィック統計情報の計算では、関連性の高いパケットをフローとしてまとめ、フローごとに到着バイト数、パケット数などをカウントする。このため、パケット到着ごとにそのパケットがどのフローに属するかを検索する。いずれの場合においても、膨大な候補の中から内容に合致するものを高速に検索する必要があり、高速にメモリ内容を検索できる TCAM が適している。

インターネットの発達、および利用形態の変化に伴い、これらの処理は今後ますます必要となっている。ルーティングテーブルは、2007 年 12 月現在 25 万エントリに迫る勢いで、かつ直近 5 年間で倍増しており、今後も加速度的に増加すると考えられる [1]。また、アクセス制御についても全世界的なセキュリティ意識の向上により、「不要なトラフィックを排除」することから「必要最小限のトラフィックのみを通過」させるなど、より強化されたアクセス制御が用いられつつある。近年の報告ではサイトにおけるアクセス制御リスト (ACL) のルール数はおよそ 5,000 程度であるといわれている [2] が、今後ますます増加するものと思われる。さらに、異種トラフィックが混在するインターネットにおいて、アプリケーション種別に応じたトラフィック制御を実現するためには、フローの識別および制御が求められるなど、今後、TCAM への需要はますます高まるものと考えられる。

しかしながら、一方で TCAM の容量を今後も増加させていくことは容易ではない。一般的に TCAM のもつ問題点として、(1) 消費電力が大きい、(2) コストが高い、ということがあげられる。TCAM は検索時、すべての格納された内容に対して比較を行うことから、全ゲートへの通放電が発生する。このため、ランダムアクセスメモリ (RAM) と比較して格段に電力消費が激しい。また、RAM と比べて製品の稼働率に対する制約が厳しく、半導体の歩留まりが悪いなどの理由によりコストを下げるのが容易ではない。特に消費電力については、ルータの消費電力が 2010 年には国内総発電量の 0.4~1.7% にまで達する [3] など社会問題化してきており、これまで以上の省電力化が求められている近年において、消費電力の激しい TCAM を安易に容量増強することができなくなりつつある。

このため、TCAM の消費電力低減・低コスト化に関する研究がなされている。[4], [5] では、ルーティングテーブル全体を複数のサブグループに分割し、それぞれごとに TCAM を割り当て、検索は該当する TCAM のみにアクセスすることによって、電力消費を下げる方法が提案されている。また、[6] では、プレフィックス内の k ビットを用いたハッシュ関数によりテーブルを分割する方法が提案されている。さらに、[7], [8] では、プレフィックスの特定の範囲をインデックスとした分割が行われている。一方、[9] などでは、プレフィックスの特性に基づいた集約を行うことで、テーブルサイズを減少させる手法が考えられている。

しかしこれらの方法には以下で述べる問題点がある。まず、

既存の TCAM を変更せずに外部回路、あるいはソフトウェアによる制御を行う方式の場合、検索途中で外部 I/O への入出力が発生するため、速度およびコスト上の問題点が発生する。また、ソフトウェアが複雑になることにより、前処理や更新などのオーバーヘッドも発生する。さらに、いずれの分割手法も分割に使用するインデックスは全てのデバイスにおいて共通であるため、必ずしもすべてのサブグループに対して最適な検索がなされているとは限らない。

一方我々の研究グループでは、低コスト・低消費電力を実現する新しい TCAM アーキテクチャについて検討を行っている。さらなる電力消費、コストの効率化を実現するため、次世代 TCAM ではネットワーク特性を考慮に入れることを検討する。具体的には、TCAM 全体を複数の「サブテーブル」に分割し、配列インデックスによる分岐を行う。検索は、まずインデックスによって検索すべきサブテーブルを求め、サブテーブル内を通常の TCAM と同様に検索する。これにより、比較演算は単一サブテーブルのみに行われることから、比較に必要な電力はサブテーブル数の逆数で抑えられることになる。

ただし本 TCAM は、消費電力およびコストに決定的な影響を持つサブテーブルの使用効率が、インデックスの決定方法によって大きく変化する。そこで本稿では、IP ルータのルーティングテーブル管理を対象とし、次世代 TCAM における効率的なルーティングテーブルの保持方法について検討する。特に、サブグループにおけるインデックスの決定法がコスト等に大きく影響することから、コストを低減させるために、テーブル全体を可能な限り均一に分散させる、インデックス作成法について検討する。[6] などの方法とは異なり、各サブグループへのインデックス生成を個別に最適化することで、より均一化された検索が可能となる。そして、分割数による提案方式の効果について数値評価により明らかにする。その結果、ルーティングテーブル全体を分割し、テーブルの特性にもとづいたインデックスにより、テーブルをほぼ均一に分散させ、より消費電力を小さくするテーブル構築が可能であることを明らかにする。

以下、2. で次世代 TCAM の構成について述べる。次に 3. では、インデックス値によるテーブル分割とその概要について説明し、4. においてデバイスの多段構成による均一化手法を述べる。さらに 5. では、提案手法である、テーブルの統計的性質を考慮したインデックス決定法について述べる。そして 6. で提案方式の有効性について数値評価を用いて考察し、最後に 7. でまとめと今後の課題について述べる。

2. 低コスト・低消費電力を実現する次世代 TCAM の構成

ここでは、本稿で対象とする低コスト・低消費電力を実現する次世代 TCAM の概要について説明する。図 1 に、対象とする TCAM デバイスの概要を示す。この TCAM は、TCAM 全体をまず 256 で分割したサブテーブルを用い、これらを並列に並べることによって構成する。この 256 個のサブテーブルのいずれと比較すべきかを、その前段階にあるインデックステーブルにより決定する。このインデックステーブルは 256 エント

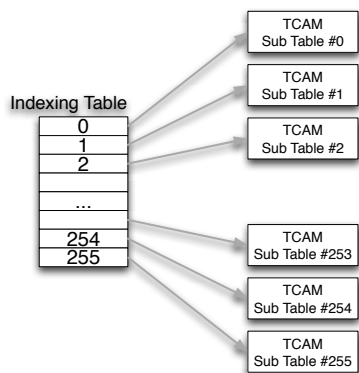


図1 TCAM 概要

りより構成され、それぞれの要素はサブテーブルへのポインタを保持する。IPv4 の場合、アドレス検索は最長 32 ビットのプレフィックスを用いて行われる。インデックステーブルへの検索はそのうち 8 ビットを用いて行われ、サブテーブルの検索は残りの 24 ビットについて行われる。

たとえば、インデックステーブルへの検索を先頭 8 ビットを用いて行う場合、ルーティングテーブルは各要素であるプレフィックスの先頭 8 ビットの値によって 256 分割され、サブテーブルに格納される。ただし、Longest Prefix Matching を実現するため、サブテーブル内はプレフィックスが短いものがアドレス上位になるように格納する。

検索は、パケットが到着するたびに、パケットの宛先アドレス (32 ビット) の先頭 8 ビットにより、残りの 24 ビットに対する検索を行うサブテーブルを決定し、サブテーブル内を通常の TCAM と同様内容一致検索を行う。

ただし、インデックステーブルの検索に用いるビットは必ずしも先頭から 8 ビットである必要はない。本デバイスでは、インデックステーブルとして用いるビットを Index Mask として入力で与え、任意のビットを 8 ビット分組み合わせることによってインデックステーブルの検索を行うことができる。これは、先頭 8 ビットをインデックスとして用いた場合、著しい不均衡が生じるため、任意ビットをインデックスとして用いることで不均衡を解消させることを目指したものである。先頭 8 ビットのインデックスによる不均衡については次節で説明する。

3. インデックス値によるテーブルの分割

すでに述べたとおり、本稿で対象とする次世代 TCAM では、インデックス値の決定の方法がサブテーブルへの格納に大きな影響を与える。ここではまず、先頭 8 ビットによるサブテーブルへの分割について説明する。本稿では、実ルーティングテーブルのデータとして、2006 年 11 月に取得した RIPE のルーティングテーブル (RouteViews より取得) を用いる。本テーブルの総エントリ数は 198,852 である。このテーブルに対して、先頭 8 ビットによるインデックスを行った場合の、各サブテーブルに格納されるプレフィックス数の分布を図 2 に示す。図で示すとおり、格納されるプレフィックス数は、サブテーブル間で大きく異なっており、プレフィックスが全く存在しないサ

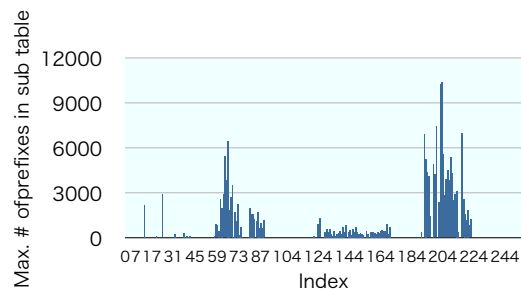


図2 Partition by first 8 bits

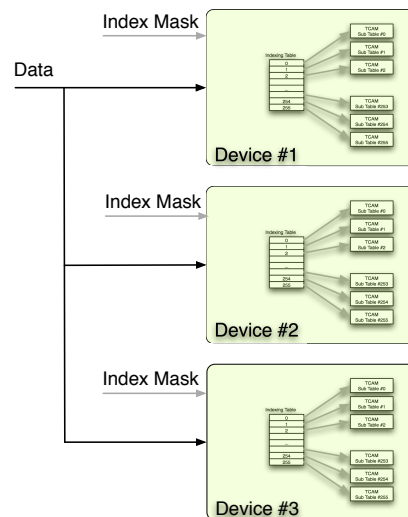


図3 マルチデバイス構成による TCAM の概要

ブテーブルが存在する一方で、プレフィックス数が 10,000 を超えるサブテーブルがあることがわかる。このうち、プレフィックス数が 0 となる部分は、インデックスにより示された先頭 8 ビットに該当する領域が IANA 予約領域 (たとえばクラス E など) や未使用領域、黎明期への Class A 割り当て (実際ほとんどグローバルとして用いられていない) などが要因として考えられる。一方、10,000 を超える領域は、複数の RIR (特に RIPE、APNIC) が、CIDR により小さいネットワーク規模 (/30, /29, /28) のアドレスを細かく割り当てているからであると考えられる。

5. では、このサブテーブル間のばらつきをさらに軽減させるため、任意ビットによるインデックスにおける、ルーティングテーブルの特性を考慮したインデックス法について提案する。

4. TCAM デバイスの並列化

本稿では、次世代 TCAM デバイスを複数並列化し、各デバイスにルーティングテーブルを分割して格納することを考える。図 3 に概要を示す。ただし、Longest Prefix Matching を保持するためには、プレフィックスの短いエントリはより上位のデバイスに格納するべきである。

テーブルの分割方法として、以下の 3 種類を考える。

(1) **均等分割 (Equal Partition):** エントリ全体をデバイス数で割り、均等に割り当てる。もっとも単純な方法であ

り、デバイスごとの使用率を一定にすることができる。しかしながら、サブテーブル間では不均衡が発生しやすいため、必ずしも最良の方法ではない。

(2) **プレフィックス長による分割 (Prefix-length based Partition):** プレフィックスの長さによって、格納するデバイスを決定する方法である。たとえば、デバイス #1 には /8~/10 のプレフィックスを格納し、デバイス #2 には /11~/13 を格納する、というように、デバイスごとに格納するプレフィックスの長さを指定する。この分割法はもっとも実装が容易であり、またプレフィックスの追加削除も簡単である。しかし一方で、デバイスごとに利用率に差が生じる可能性があるという問題がある。

(3) **偏向分割 (Biased Partition):** 均等分割の問題点は、プレフィックス長の短いものを格納する時に、どうしてもサブテーブルの不均衡が発生することである。これは、インデックスとして "*" (don't care) が存在するビット位置を含めることができないため、プレフィックス長が /8 であるプレフィックスが含まれるデバイスのインデックスは必ず先頭 8 ビットにならざるを得ない。先頭 8 ビットによるインデックスは、サブテーブル間で大きな不均衡を生じさせるため、その影響を可能な限り軽減させる必要がある。そこで偏向分割では、より短いプレフィックスには多くの TCAM を割り当てることによってサブテーブルの不均衡を軽減させる。

以上によって分割されたプレフィックスは、さらに各デバイスのインデックスによりサブテーブルに分割される。このサブテーブル間での不均衡を小さくするためには、各デバイスのインデックスを適正に決定することが重要となる。そのためのインデックス決定法を次節で述べる。

5. テーブルの統計的性質を考慮したインデックス決定法

5.1 基本アイデア

インデックス決定の目的は、サブテーブルによる分割を可能な限り均衡化させるためには、検索ビットのうちどのビットをインデックスとして用いるのがよいか、というのを決定することである。ここでは、インデックスで用いる先頭ビットと 2 ビット目以降について分けて考える。

先頭ビットの選択は、そのビットをインデックスと使用することで、テーブル全体が 2 等分されるのが望ましい。これはすなわち、そのビットの位置が 0 であるプレフィックス数と、1 であるプレフィックス数がほぼ同じであるということに他ならない。したがって、各ビット位置ごとにその値が 0, 1 であるプレフィックス数をそれぞれカウントし、その差がもっとも 0 に近いものを選ぶのが望ましい。

次に、2 ビット目について考える。先頭ビットと連結することによって 00, 01, 10, 11 の 4 種類の値を取り得るが、それぞれについてプレフィックスが 25% ずつ分配されることが望ましい。このとき、2 ビット目だけを考えると、やはり先頭ビットと同様に、そのビット位置の値が 0, 1 であるプレフィックス数がほぼ等しいビットが適当であるといえる。

次に、先頭ビットとの相関を考える。先頭ビットとの相関が強いほど、00, 11 が多く現れることから、インデックスとしては不適切である。逆に、負の相関が強ければ、01, 10 ばかりが出るので、やはり不適切である。結局、もっともよく分散させるためには、先頭ビットとの相関性がもっとも弱いビット位置である、すなわち、 $A=B$ となる割合が 50% であり、 $A \neq B$ となるのがやはり 50% となるがもっとも適当であるといえる。

以上より、インデックスの 2 ビット目として適当なビットの位置は、

- その位置の値が 0, 1 であるプレフィックスの個数がほぼ 50% ずつ

- 先頭ビットとの相関がもっとも弱い

という 2 つの条件を満足することになる。これらの条件はどちらかが優先度が高いものではなく、ともに満たすものが望ましい。そこで、本稿では相乗平均を利用し、順位付けを行う。3 ビット目以降についても同様に求める。

5.2 インデックスビット選択アルゴリズム

以下に具体的なアルゴリズムを示す。ここで $\text{bit}[j]$ は、プレフィックスの j ビット目の値 (0, 1, *) を表す。

(1) **初期設定:** すべてのエントリについて、"*" を含まないビット位置を候補とし、その集合を J とする。

(2) **先頭ビットの決定:** 候補のビットそれぞれについて ($j \in J$)、ビット j が "0", "1" であるプレフィックス数 p_j^0, p_j^1 をカウントする

(3) $P_j = |(p_j^0 - p_j^1)|$ をそれぞれ計算し、 $P_i = \min_{j \in J} P_j$ となるビット位置 i を求め、インデックスの先頭ビットとする。そして、 i を J から取り除く

(4) **2 ビット目以降の決定:** 候補ビット $j (j \in J)$ について、同様に $P_j = |(p_j^0 - p_j^1)|$ を計算する。

(5) ビット i との相関性を求めるため、 $\text{bit}[i] == \text{bit}[j]$ となるプレフィックスの数を求め、これを $e_{i,j}$ とする。

(6) $C_{i,j} = |e_{i,j} - M/2|$ を計算する。ただし、 M はルーティングテーブルのプレフィックス総数を表す。

(7) 相乗平均 $Q_{i,j} = \sqrt{P_j C_{i,j}}$ を計算し、 $Q_{i,j}$ が最も小さいものから順に 2 ビット目、3 ビット目と選択する。ただし、相乗平均は積 $P_j C_{i,j}$ で代用できる。

6. 数値評価

本節では、提案するインデックス生成法について、実ルーティングテーブルを用いた数値評価を行い、その結果を示す。数値評価に用いたデータは、RouteViews プロジェクトより取得した、RIPE ルーティングテーブルで、2006 年 9 月に取得したものである。総プレフィックス数は 198,852 である。また、デバイスに対するルーティングテーブルの分割は均等分割 (Equal Partition) を用いる。

評価指標として、サブテーブルにおける最大プレフィックス数 N_p を用いる。これは、TCAM サブテーブルの最大エントリ数に該当する。すなわち、検索の最終段階における同時比較数を表すことから、この数が大きいほど消費電力も大きく、コストも高いことを意味する。したがって、 N_p を可能な限り最

表 1 Effect of Indexing: 3 Devices

Dev.	Prefix len	Selected bit pos	N_p
#1	/8 - /22	1, 2, 3, 4, 5, 6, 7, 8	2,562
#2	/22 - /24	9, 14, 16, 17, 19, 20, 22, 23	359
#3	/24 - /32	11, 13, 14, 17, 18, 19, 20, 21	326

小とすることが望ましい。ただし同程度の N_p であれば、外部 I/O 数の少ない方がコストが軽減できるため、デバイス数が少ない方が望ましい。

6.1 2分割の場合

はじめに、提案方式による効果を確認するため、2つのデバイスを用いた場合の結果を示す。この場合、デバイスにはそれぞれ 99,942 個のプレフィックスが格納される。デバイス #1 には、プレフィックス長が /8 から /24 までのエントリが格納され、デバイス #2 は /24 から /32 までのプレフィックスが対象となる。提案方式によって選択されたビットはそれぞれ

- #1: 1, 2, 3, 4, 5, 6, 7, 8
- #2: 13, 14, 16, 17, 19, 20, 22, 23

となった。このとき、各サブテーブルに格納されたプレフィックス数の最大値および分散はそれぞれ、#1 で Max 3,390、Var = 178,915、#2 では Max = 467、Var = 410.3 となった。この結果より、デバイス #2 での効果は非常に高く、プレフィックスがほぼ均一化してサブテーブルに格納されていることがわかる。その一方で、デバイス #1 ではサブテーブル間のばらつきが高く、最大も非常に大きな値である。この理由はすでに述べたとおりデバイス #1 のインデックスビットが先頭 8 ビットにせざるを得ないためである。

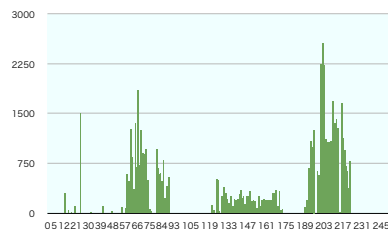
6.2 3分割の場合

次に 3 分割の場合について示す。3 分割を行った場合のデバイスごとの対象プレフィックス長、選択ビット、サブテーブル内の最大プレフィックス数、およびその分散を表 1 に、また各デバイスごとのサブテーブル内のプレフィックス数分布を図 4 に示す。2 分割の場合と同様、デバイス #1 をのぞき、ほぼ均等に分布できていることが分かる。

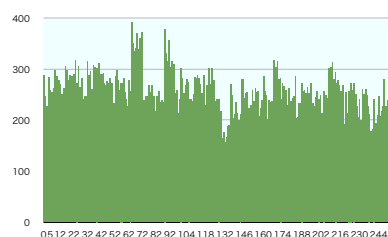
6.3 分割数による効果

次に、デバイス数を増やすことでどの程度効果があるかを調べるため、デバイス数と最大プレフィックス数の関係を図 5 に示す。ここで、5(a) はデバイス #1 を含めたすべてのデバイスにおける最大プレフィックス数、5(b) はデバイス #1 をのぞいた場合の最大プレフィックス数をそれぞれ示したものである。これらの図より、最大プレフィックス数は、デバイス #1 の影響が最も大きいことがわかる。一方、デバイス #1 をのぞいた場合、最大プレフィックス数の変化は 10 デバイス程度で収束し、それ以降デバイスを増やしても、あまりプレフィックス数には変化がなく、むしろばらつきが大きくなっている。

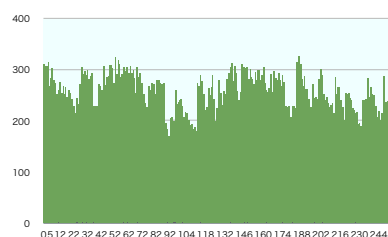
これを調べるため、図 6 に、デバイス数が 5, 10, 15, 20 の時のそれぞれのデバイスごとの最大プレフィックス数の分布を示す。この図より、デバイス数が 5, 10 の場合では比較的デバイス間のプレフィックス数はデバイス #1 をのぞきほぼ安定しているのに対し、15, 20 デバイスの場合はデバイス間のばらつ



(a) #1

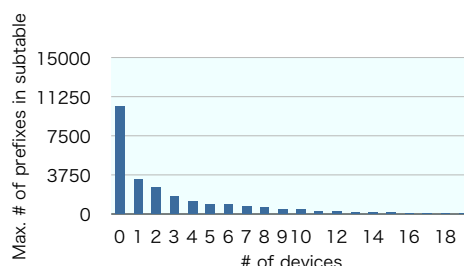


(b) #2

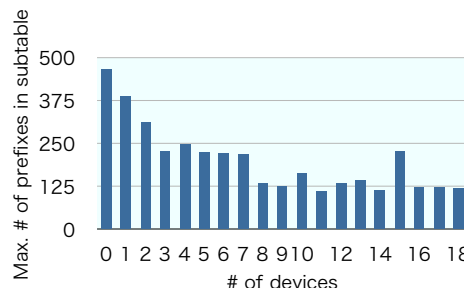


(c) #3

図 4 Distributions of # of prefixes in sub table



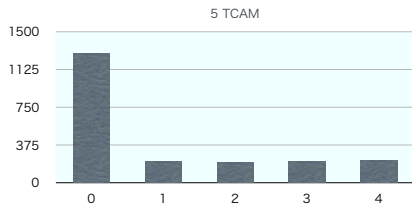
(a) Include Device #1



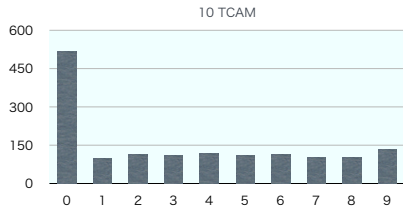
(b) Except Device #1

図 5 Appropriate number of devices

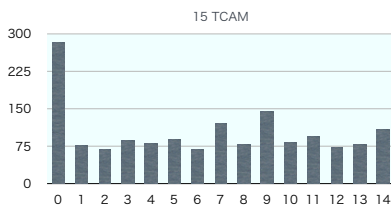
きが大きいことがわかる。これは、デバイスごとに格納するプレフィックス数が少なくなるため、細かい CIDR に細分化して割り当てられた特定の領域について、プレフィックス全部を単



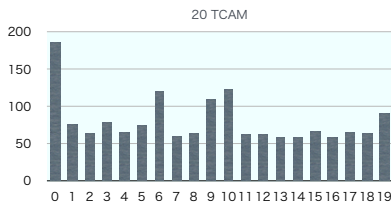
(a) 5 TCAMs



(b) 10 TCAMs



(c) 15 TCAMs



(d) 20 TCAMs

図6 Distribution of max. # of prefixes among TCAM devices

一デバイスで格納することができないためである。

6.4 デバイス#1の細分化

以上の結果より、デバイス#1のプレフィックス数が結果に大きな影響を与えることがわかった。すでに述べたとおり、これは $1/8$ のプレフィックスが存在するデバイスでは、インデックスが先頭8ビット以外を取り得ないためである。そこで、 $1/8$ のプレフィックスによる影響をできるだけ小さくするため、短いプレフィックス長のエントリについては、さらに分割して細分化することを考え、残りを均等分割する。図7に、プレフィックス長の短いものから格納したプレフィックスの数によって、最大プレフィックス長 N_p がどのように変化するかを示す。この図より、12,000程度のプレフィックスを格納するのであれば、先頭8ビットによるインデックスを用いても、最大プレフィックス数は200程度に抑えることが可能である。6.3の結果より、残りを9あるいは10デバイスで均等分割すれば、やはり最大プレフィックス数は200程度になるので、全体を考えた場合でも最大プレフィックス数を200程度に抑えることが可能となる。

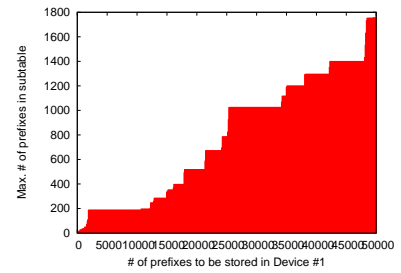


図7 Impact of short prefixes

7. まとめと今後の課題

次世代 TCAM において、プレフィックスをサブテーブルに均等配置するためにはインデックスの決定方法が重要となる。本稿では、ルーティングテーブルの特性に基づくインデックス決定法の提案を行った。数値評価により、提案方式によって得られたインデックスを用いて分割することで、大幅なエントリ数削減が可能であることがわかった。今後の課題として、一度構築されたテーブルについて、BGP 更新に対してどの程度安定性があるかを調査する必要がある。また、より詳細なコスト比較についても検討すべきであると考えられる。

謝 辞

本研究の遂行において、多大なるご協力をいただいた、大阪大学サイバーメディアセンター准教授、長谷川剛博士に深謝申し上げます。

文 献

- [1] G. Houston, "BGP reports." <http://bgp.potaroo.net/>.
- [2] D. E. Taylor and J. S. Turner, "ClassBench: A packet classification benchmark," *Tech. Rep. WUCSE2004-28, Department of Computer Science & Engineering, Washington University*, April 2004.
- [3] 小笠原 敦, "情報通信のエネルギー問題 —求められる通信インフラの省電力化—," 文部科学省「科学技術動向」, June 2006.
- [4] R. Panigrahy and S. Sharma, "Reducing TCAM power consumption and increasing throughput," in *Proceedings of Hot Interconnects (HotI 2002)*, p. 107, August 2002.
- [5] R. Panigrahy, S. Sharma, M. J. Akhbarizadeh, and M. Nourani, "A TCAM-based parallel architecture for high-speed packet forwarding," *IEEE Transactions on Computers*, vol. 56, pp. 58–72, January 2007.
- [6] F. Zane, G. Naligar, and A. Basu, "CoolCAMs: Power-efficient TCAMs for forwarding engines," in *Proceedings of IEEE INFOCOM 2003*, vol. 1, pp. 42–52, April 2003.
- [7] K. Zheng, C. Hu, H. Lu, and B. Liu, "An ultra high throughput and power efficient TCAM-based IP lookup engine," in *Proceedings of IEEE INFOCOM 2004*, pp. 1984–1994, April 2004.
- [8] K. Zheng, C. Hu, H. Lu, and B. Liu, "A TCAM-based distributed parallel IP lookup scheme and performance analysis," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 863–875, April 2006.
- [9] V. C. Ravikumar, R. N. Mahapatra, and L. N. Bhuyan, "EaseCAM: An energy and storage efficient TCAM-based router architecture for IP lookup," *IEEE Transactions on Computer*, vol. 54, pp. 521–533, May 2005.