

# Routing/ARP/ACL/QoS ごとの テーブル分離パケット処理キャッシュ

長田 大樹<sup>1,a)</sup> 田中 京介<sup>1</sup> 八巻 隼人<sup>1</sup> 三輪 忍<sup>1</sup> 本多 弘樹<sup>1</sup> 五島 正裕<sup>2</sup>

**概要:** ルータのパケット処理を高スループット化、省電力化する技術として、パケット処理におけるテーブル検索の結果をキャッシュするパケット処理キャッシュ (Packet Processing Cache: PPC) が提案されている。PPC はパケットの 5 タプル (送信元/宛先 IP アドレス, 送信元/宛先ポート番号, プロトコル番号) をキーとして、複数テーブルの検索結果を 1 エントリにまとめて格納できる一方で、ルーティングテーブル等の 5 タプルの全てをキーとして用いないテーブルに対しては、キャッシュの効率が悪い。そこで本稿では、Routing/ARP/ACL/QoS テーブルを個別にキャッシュするテーブル分離 PPC を提案する。実ネットワークトレースを用いた評価では、2 階層の PPC において、テーブル分離 PPC は従来の PPC に対してテーブル検索スループットを 1.47 倍に向上し、消費電力を 24%削減できることを示した。

## Table-Separate Packet Processing Caches for Routing/ARP/ACL/QoS

### 1. はじめに

近年、インターネットサービスの成長は目覚ましく、全世界の IP トラフィック総量は 2022 年までに 2017 年の約 3 倍に増加することが予測されている [1][2]。通信トラフィック総量の増加に伴い、インターネットルータではパケット処理スループットの向上が課題となっている。また、パケット処理に伴い電力を消費することから、インターネットルータの消費電力の増大も懸念されている。以上のことから、今後、インターネットルータにはパケット処理の高スループット化と省電力化を同時に達成することが求められる。

インターネットルータにおいて、パケット処理スループットと消費電力を決定づける処理は主にテーブル検索である [3]。ルータは到着したパケットの転送に必要な情報 (転送先や転送優先度、転送可否など) を得るために、ルータ内のメモリに格納されたテーブルを検索する。一般的なルータではルーティングテーブル、ARP (Address Reso-

lution Protocol) テーブル、ACL (Access Control List)、QoS (Quality of Service) テーブルなどの複数テーブルを検索する。

近年のルータは高い検索機能を持つ TCAM (Ternary Content Addressable Memory) にテーブルを格納し、検索を高速化している。しかし、現在の TCAM の検索性能では、最小サイズのパケットが連続してルータに到着するワーストケースにおいて、100Gbps 程度のスループットに留まり、将来的な回線速度である 400Gbps や 1Tbps に対して検索性能が不足している。また、TCAM の 1 アクセスあたりの消費電力は、SRAM などの一般的に使用されるメモリと比べて極めて大きく、TCAM はルータ消費電力全体の約 40%を占める [4]。これらのことから、パケット処理の高スループット化と省電力化の達成には、TCAM のみによらない新たなテーブル検索アーキテクチャが期待されている。

そのための手法の一つとして、**パケット処理キャッシュ (PPC: Packet Processing Cache)** が注目されている。PPC では TCAM に加え、高速かつ省電力な SRAM をテーブル検索のキャッシュとして使用する。インターネットルータのテーブル検索においては、同一の TCAM エントリを参照するパケット群が短時間に連続して到着する傾向がある [5]。そのため、PPC による高いヒット率が期待で

<sup>1</sup> 電気通信大学  
The University of Electro-Communications, Chofu, Tokyo  
182-8585, Japan

<sup>2</sup> 国立情報学研究所  
National Institute of Informatics, Chiyoda, Tokyo  
101-8430, Japan

a) osada@hpc.is.uec.ac.jp

き、PPC を用いることで TCAM アクセス回数を大幅に削減できる。

PPC を用いたテーブル検索では、PPC のアクセス遅延と消費電力、またミス率が性能を左右する。これらは、いずれも PPC の容量に依るが、トレードオフの関係にあるため、PPC の容量効率を高め、出来る限り小規模な PPC を用いて高いヒット率を達成できることが望ましい。

そこで本研究では、Routing/ARP/ACL/QoS の検索結果を一つのエン트리としてキャッシュする従来の PPC に対し、各テーブルの検索結果を個別にキャッシュするテーブル分離 PPC を提案し、PPC の容量効率の向上を目指す。提案手法では、テーブル毎の PPC で必要な情報だけをキーに用いるため、より効率的に PPC のエントリを作成することが可能となる。各テーブルの PPC の容量比率を適切に決定し、従来 PPC よりも高スループットかつ省電力なテーブル検索を実現する。

本稿は以下のように構成されている。第 2 章では、テーブル検索処理の改善手法の一つであるパケット処理キャッシュについて述べる。第 3 章で本研究の着眼点を述べ、第 4 章で提案手法について述べる。第 5 章では評価方法を述べ、第 6 章に評価結果を示す。第 7 章で関連研究について述べ、第 8 章に結論をまとめる。

## 2. パケット処理キャッシュ

### 2.1 パケット処理キャッシュの概要

PPC はテーブル検索結果を、TCAM よりも小容量ゆえ、高速かつ省電力なメモリにキャッシュする。この際、全てのテーブルのテーブル検索結果は、パケットの送信元 IP アドレス、宛先 IP アドレス、送信元ポート番号、宛先ポート番号、プロトコル番号からなる 5 タプルで一意に決定されるフローごとにまとめて一つのエン트리としてキャッシュする。図 1 に示すように、ルータ内の各テーブルは 5 タプルのいずれか、または全てをキーとして検索される。したがって、PPC がテーブル検索結果をフロー単位でキャッシュすることで、同一フローの後続パケットは TCAM にアクセスすることなく、PPC よりテーブル検索結果を得ることが可能となる。

図 2 に PPC のテーブル検索処理の流れを示す。まず、テーブル検索時に、パケットの 5 タプルのハッシュ値を計算し、それをインデックスとして PPC を参照する。その結果、同一フローのエントリが PPC 内に存在した場合 (PPC ヒット) には、PPC からテーブル検索結果を得る。同一フローのエントリが PPC 内に存在しない場合 (PPC ミス) には、TCAM にアクセスし、テーブル検索を行い、TCAM で得られたテーブル検索結果とフロー情報を PPC にキャッシュする。

PPC にはタグとして 5 タプル、データとしてテーブル検索結果が格納される。それぞれ、5 タプルは 13byte、テ

Dst IP	Next Hop IP	Output I/F	Next Hop IP	MAC address
2.2.2.2/24	3.3.3.3	1	3.3.3.3	1234.5678.9abc
1.1.1.1/24	6.6.6.6	4	6.6.6.6	4567.1234.7777
⋮	⋮	⋮	⋮	⋮

(a). Routing Table

(b). ARP Table

Src IP	Dst IP	Src Port	Dst Port	Protocol	Result
*	*	*	*	6	permit
1.1.1.1	*	*	*	36	deny
⋮	⋮	⋮	⋮	⋮	⋮
*	*	*	*	*	deny

(c). ACL

Src IP	Dst IP	Src Port	Dst Port	Protocol	Result
*	*	*	*	6	0
1.1.1.1	*	*	*	23	2
⋮	⋮	⋮	⋮	⋮	⋮

(d). QoS Table

図 1 ルータ内の各テーブルの構成

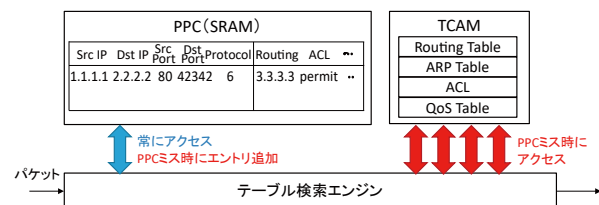


図 2 PPC のテーブル検索処理の流れ

ブル検索結果は、出力インタフェース情報 1byte, MAC アドレス 12byte, フィルタリング情報 1byte, QoS 情報 1byte の計 15byte により構成される。従って、PPC は 1 エントリあたり 28byte となる。

PPC を用いたテーブル検索では、PPC ミスに伴い TCAM アクセスが生じることから、PPC ミスを削減することがスループット向上と省電力化に重要な要素となる。一方で、キャッシュミスを削減するために PPC の容量を増やすと、SRAM のアクセス遅延や消費電力が増大してしまう。

### 2.2 多階層パケット処理キャッシュ

近年の PPC に関する研究では、一般的なマイクロプロセッサのキャッシュに習い、PPC を階層化した多階層 PPC が提案されており、これにより PPC の高スループット化と省電力化が実現されている [6]。多階層 PPC では、容量の異なる複数の PPC を用いて、前段に小容量で高速な PPC を、後段に大容量な PPC を配置することで、高速な PPC アクセスと低い PPC ミス率の両立が可能となっている。

田中らは、PPC を多階層化する、多階層 PPC を提案している [6]。2 階層の場合、32KB の L1PPC と 512KB エントリの L2PPC により、従来の 1 階層の PPC と比べてスループットを 190.4% 向上させ、平均消費電力を 27.3% 削

表 1 サーバ・クライアント通信のフロー例

送信元 IP アドレス	宛先 IP アドレス	送信元 ポート番号	宛先 ポート番号	プロトコル 番号
1.1.1.1	2.2.2.2	443	55346	6
1.1.1.1	2.2.2.2	443	55347	6
1.1.1.1	2.2.2.2	443	55349	6
1.1.1.1	2.2.2.2	443	55351	6

減できることを示している。また、3階層の場合は、8KBのL1PPC、128KBのL2PPC、4MBのL3PPCにより、1Tbpsのテーブル検索処理性能を達成でき、スループットを254.1%改善し、消費電力を27.8%削減できることを示している。

PPCの多階層化はテーブル検索の高スループット化、省電力化に有効なアプローチではあるが、論文[6]でも示されているように、L1PPCやL3PPCにおいて十分なヒット率を得られず、後段のL2PPCやTCAMがボトルネックとなることがある。更なるテーブル検索性能の向上には、PPCの容量効率の向上や初期参照ミスの削減など、階層化以外のアプローチによってPPCミスを削減することも必要となる。

### 3. 本研究の着眼点

PPCは複数テーブルの検索結果をまとめて1エンタリに格納できる一方で、ルーティングテーブルやARPテーブル単体から見るとキャッシュの効率が悪い。これは、本来ルーティングテーブルやARPテーブルの検索に用いられるキーがIPアドレスのみであるのに対し、PPCはフロー単位でテーブル検索結果をキャッシュするためである。

実際の通信におけるフローの例(表1)を用いて説明する。インターネットにおいては表1に示すように、短命なエフェメラルポートが通信に用いられることが多い。この際、ポート番号のみが異なる複数フローの packets が転送され、PPCではフロー数分のエンタリが生成される。しかしながら、ルーティングテーブル単体で見た場合には、宛先IPアドレス2.2.2.2をキーとした1エンタリがあれば十分である。このようなPPCにおける容量効率の悪さは、容量性ミスや偽の初期参照ミスを増加させる一因となる。

図3は、5タプルをキーとする従来のPPCと、宛先IPアドレスのみをキーとするルーティングテーブルのキャッシュの、容量に対するミス率を測定したグラフである。この図を見てもわかるように、ルーティングテーブルのキャッシュはPPCと同程度のミス率を1/10~1/100程度の容量で達成できる上に、最小で1%以下までミス率を削減可能となる。

### 4. 提案手法

前章の着眼点をもとに、本論文ではテーブル毎に個別の

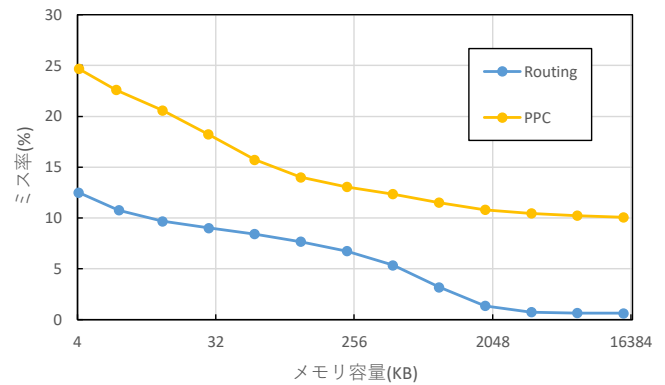


図 3 各テーブルのキャッシュミス率

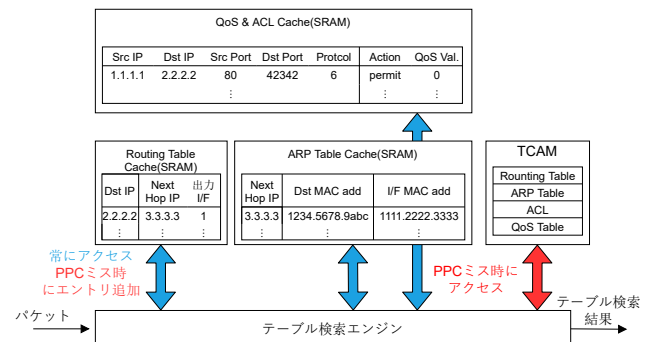


図 4 テーブル分離 PPC の概略図

PPCを用意したテーブル分離パケット処理キャッシュを提案する。

本章では、提案手法のコンセプトと具体的なPPCの構成について示す。

#### 4.1 コンセプト

図4にテーブル分離パケット処理キャッシュの概略図を示す。テーブル分離PPCでは、テーブル毎にPPCを個別に用意することで、主にルーティングテーブルとARPテーブルに対する容量効率の改善が期待できる。

テーブル分離PPCは、従来PPCをテーブルごとに分離させた構成となるため、同じ総容量で見た場合に各テーブルに対するPPCの容量は従来PPCよりも小さくなる。一方で、前章で示したようにテーブルによっては容量効率が10倍~100倍程度向上する。従って、各テーブルに対し、PPCの容量を適切に設定することで、従来PPCと同程度の総容量であっても総体的により高いPPCヒット数を達成することが期待できる。

#### 4.2 構成

各テーブルのPPCの構成を図5に示す。ルーティングテーブルに対してはキーに宛先IPアドレス4byte、バリューにネクストホップIPアドレス4byteと出力インタフェース情報1byteの計5byte、ARPテーブルに対してはキーにネクストホップIPアドレス4byte、バリューに

Key(4byte)		Value(5byte)		
Dst IP (4byte)	Next Hop IP (4byte)	Output I/F (1byte)		
1.1.1.1	3.3.3.3	1		
⋮	⋮	⋮		

(a). Routing Table Cache

Key(4byte)		Value(12byte)	
Next Hop IP (4byte)	Dst MAC Address (6byte)	I/F MAC Address (6byte)	
3.3.3.3	1234.5678.9abc	1111.2222.3333	
⋮	⋮	⋮	

(b). ARP Table Cache

Key(13byte)					Value(2byte)	
Src IP (4byte)	Dst IP (4byte)	Src Port (2byte)	Dst Port (2byte)	Protocol (1byte)	Action (1byte)	QoS Val. (1byte)
1.1.1.1	2.2.2.2	80	42342	6	Permit	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

(c). QoS & ACL Cache

図 5 各テーブルの PPC の構成

2つの MAC アドレス 12byte (ルータ自身の出力ポートの MAC アドレスと、その出力ポートと繋がっているルータの MAC アドレス), ACL と QoS テーブルに対しては検索キーが同一であるため 2 テーブルをまとめ、キーに 5 タプル 13byte, バリューに ACL の結果である転送可否, QoS テーブルの結果である転送優先度の計 2byte となる。

ARP テーブルに対する PPC は、ルータが備えるポートと同数のエントリがあれば、全ての検索結果を保持でき、PPC ミスはエントリ数と同じ回数の初期参照ミスが起こるのみである。そのため、ARP テーブルに対する PPC のミス率は (ルータのポート数)/(処理パケット数) となる。ここで、ルータの持つポート数は近年のハイエンドルータで数十から百程度であり、それに対してルータが処理するパケット数は億を超える。従って、ARP テーブルに対する PPC のミス率は、他テーブルに対する PPC のミス率と比べて非常に小さい値となりほぼ無視できる。

## 5. 評価方法

### 5.1 評価目的

提案手法において達成可能なテーブル検索のスループットおよび消費電力を従来 PPC と比較し、提案手法の有効性を明らかにする。

### 5.2 評価環境

評価には PPC のシミュレータと実ネットワークのパケットトレースを用いた。以下でそれぞれについて詳述する。

PPC のシミュレータは関連研究 [6] で作成されたシミュレータを使用した。このシミュレータは Go 言語で記述されており、PPC のキャッシュ構成をパラメータとし、入力として与えたパケットトレースが PPC を用いて処理された際の、PPC アクセス数および TCAM アクセス数、それぞれのアクセス率を計測することができる。

表 2 評価した PPC の構成

パラメータ	値
容量	8KB - 128KB(L1) 256KB - 2MB(L2)
連想度	4-way (ARP 以外) full associative (ARP)
置換アルゴリズム	LRU

本論文の評価においては、WIDE MAWI Working Group-Traffic Archive[7] の提供する実ネットワークのパケットトレースを PPC シミュレータへの入力トラフィックとして用いた。このパケットトレースは、太平洋を横断する 1Gbps リンク (WIDE samplepoint-F) で 2017 年に取得された、総パケット数 52,840,127 の 15 分間のトレースである。

### 5.3 評価モデル

本論文では、ルーティングテーブルに対する PPC と QoS&ACL に対する PPC の容量比率を 1:3, 1:4 とした 2 パターンと、ルーティングテーブルに対する PPC の容量を 32KB に固定した計 3 パターンで評価する。ARP テーブルに対する PPC は前述したように 48 エントリとする。

表 2 に、本評価における PPC の構成をまとめた。ARP テーブルの PPC のみフルアソシアティブ方式、それ以外はセットアソシアティブ方式 (4way) とし、追い出しアルゴリズムは LRU (Least Recently Used) とした。PPC の総容量は 8KB から 2MB の範囲としている。

2.2 節で紹介した多階層 PPC を考慮し、1 階層 PPC と 2 階層 PPC それぞれを実装した。2 階層 PPC においては、両階層を従来の PPC で構成する従来アーキテクチャと、L1PPC のみをテーブル分離 PPC で構成するアーキテクチャ、両階層をテーブル分離 PPC (ARP テーブルのキャッシュは L1 のみ) で構成するアーキテクチャの 3 種を実装し、比較した。2 階層 PPC の場合の容量は、L1PPC を 8KB-128KB, L2PPC を 256KB-2MB の範囲内で組み合わせ、インクルーシブな構成とした。

### 5.4 スループットと消費電力のモデル式

テーブル分離 PPC を用いたテーブル検索におけるスループットと消費電力については、田中らにより提案されている PPC における消費電力、スループットのモデル式 [6] を拡張し、それぞれの値を算出する。従来の PPC のスループットと消費電力の評価には、田中らによる先行研究のモデル式を使用した。

まず、1 階層のテーブル分離 PPC によるテーブル検索スループット  $T$  の計算式を以下に示す。

$$T = \min \left( \frac{l}{d_{Rtg}}, \frac{l}{d_{ARP}}, \frac{l}{d_{QoS\&ACL}}, \frac{l}{d_{TCAM} \cdot m_{avg}} \right) \quad (1)$$

$$m_{avg} = (m_{Rtg} + m_{ARP} + 2m_{QoS\&ACL}) / 4 \quad (2)$$

ここで  $d_{Rtg}$ ,  $d_{ARP}$ ,  $d_{QoS&ACL}$ ,  $d_{TCAM}$  はそれぞれ各テーブル分離 PPC および TCAM のサイクルタイムを表す。また,  $m_{avg}$  は (2) 式で定義されるテーブル分離 PPC 全体の重み付き平均ミス率を表す。  $m_{Rtg}$ ,  $m_{ARP}$ ,  $m_{QoS&ACL}$  はテーブル分離 PPC それぞれのミス率を表す。  $l$  は 1 パケットの平均サイズを表しており, 本研究では最悪ケースを想定し, パケット長の最小値である 64byte とした。

2 階層のテーブル分離 PPC におけるテーブル検索スループットは, (1) 式, (2) 式を拡張し, 以下により求められる。

$$T = \min \left( \frac{l}{d_{Rtg}^{L1}}, \frac{l}{d_{ARP}^{L1}}, \frac{l}{d_{QoS&ACL}^{L1}}, \frac{l}{d_{Rtg}^{L2} \cdot m_{Rtg}^{L1}}, \frac{l}{d_{QoS&ACL}^{L2} \cdot m_{QoS&ACL}^{L1}}, \frac{l}{d_{TCAM} \cdot m_{avg}} \right) \quad (3)$$

$$m_{avg} = \left( m_{Rtg}^{L1} \cdot m_{Rtg}^{L2} + m_{ARP}^{L1} + 2m_{QoS&ACL}^{L1} \cdot m_{QoS&ACL}^{L2} \right) / 4 \quad (4)$$

ここで各変数の上付き添字は L1PPC, L2PPC の別を示す。

次に, PPC における平均消費電力  $P$  のモデル式を式 (5) に示す [6]。

$$P = E \cdot n + S \quad (5)$$

$E$  はメモリのパケットあたりの動的消費エネルギーを表し,  $S$  はメモリの静的消費電力を表している。また,  $n$  は 1 秒あたりのパケット数を表している。ここで, 1 階層のテーブル分離 PPC での  $E$  と  $S$  は, それぞれ以下の式で表せる。

$$E = E_{Rtg} + E_{ARP} + E_{ACL&QoS} + 4E_{TCAM} \cdot m_{avg} \quad (6)$$

$$S = S_{Rtg} + S_{ARP} + S_{ACL&QoS} + S_{TCAM} \quad (7)$$

$E_{Rtg}$ ,  $E_{ARP}$ ,  $E_{ACL&QoS}$ ,  $E_{TCAM}$  はそれぞれテーブル分離 PPC と TCAM の動的消費エネルギーを表している。また, 同様に  $S_{Rtg}$ ,  $S_{ARP}$ ,  $S_{ACL&QoS}$ ,  $S_{TCAM}$  はそれぞれの静的消費電力を表している。

2 階層のテーブル分離 PPC における消費電力は上式を拡張し, 以下となる。

$$E = E_{Rtg}^{L1} + E_{ARP}^{L1} + E_{ACL&QoS}^{L1} + E_{Rtg}^{L2} \cdot m_{Rtg}^{L1} + E_{QoS&ACL}^{L2} \cdot m_{QoS&ACL}^{L1} + 4E_{TCAM} \cdot m_{avg} \quad (8)$$

$$S = S_{Rtg}^{L1} + S_{ARP}^{L1} + S_{ACL&QoS}^{L1} + S_{Rtg}^{L2} + S_{QoS&ACL}^{L2} + S_{TCAM} \quad (9)$$

スループットと同様に, 各変数の上付き添字は L1PPC, L2PPC の別を示す。

#### 5.4.1 メモリのパラメータ

スループットおよび消費電力の算出において, PPC に使用されるメモリのサイクルタイムや動的消費エネルギー, 静的消費電力は, CACTI6.5[8] のシミュレーション結果を

表 3 CACTI6.5 の設定値

設定項目	設定値
半導体プロセス	32nm
ポート数	1(read)+1(write)
メモリトランジスタ	ITRS-HP(L1) ITRS-LSTP(L2)

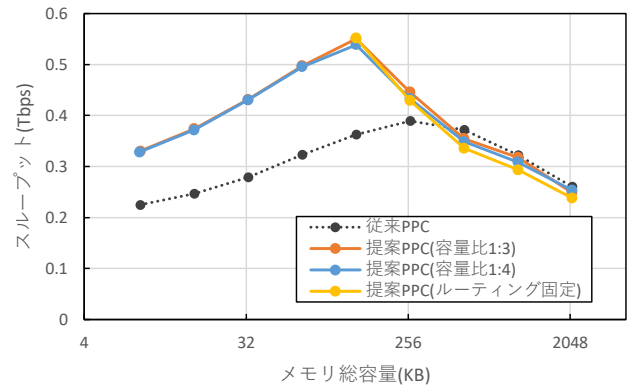


図 6 1 階層 PPC におけるスループットの評価結果

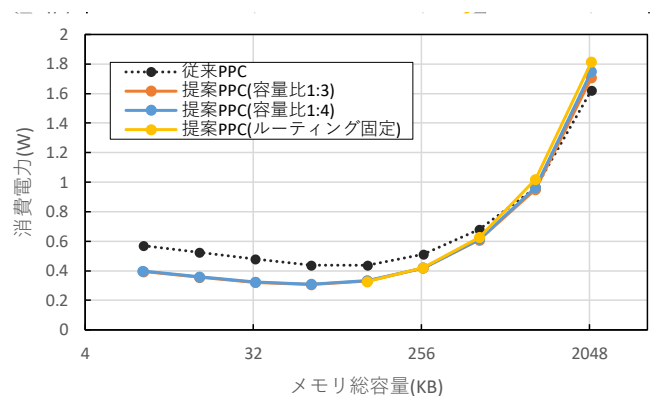


図 7 1 階層 PPC における消費電力の評価結果

用いた。CACTI の設定値を表 3 に示す。TCAM のパラメータは, ルータで一般的に用いられる 20MbitTCAM を想定し, 消費電力は容量に比例すると仮定して論文 [9] の値を定数倍することで見積もった。パケット数  $n$  は, 1Gbps のリンクである WIDE トレースにおける 1 秒あたりの平均パケット数を元に, スループットの増大に比例してパケット数が増加すると仮定して計算を行った。

## 6. 評価結果

1 階層 PPC の評価結果を図 6 から図 8 に示す。

PPC の総容量が 256KB 以下においては, テーブル分離 PPC は従来 PPC よりもテーブル検索を高スループット化, 省電力化できることがわかる。これは PPC を分離することによって, PPC ミス率が下がり, TCAM アクセス回数が減ったことによるものだと考えられる。テーブル分離 PPC と従来 PPC の双方において高スループット, 低消費電力を達成する総容量 128KB で比較した場合, 従来 PPC は 363Gbps, 0.44W であるのに対し, テーブル分離

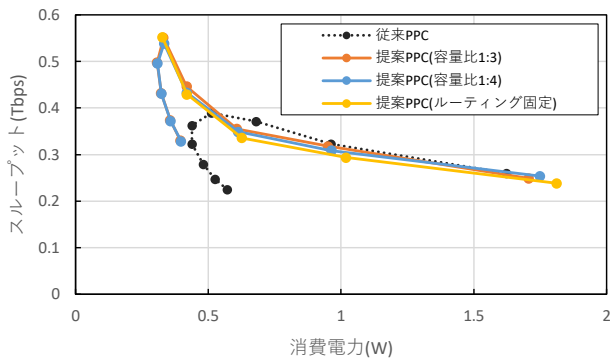


図 8 1 階層 PPC のスループットと消費電力

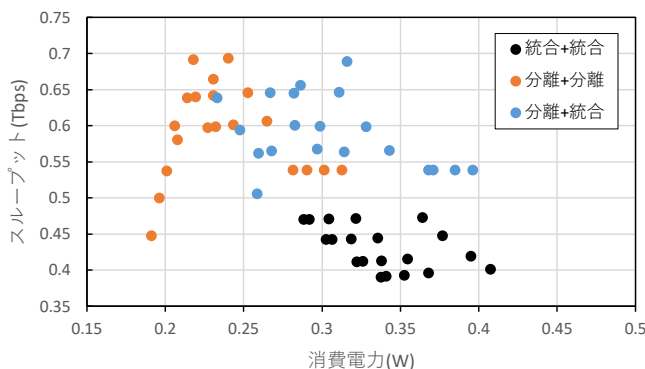


図 9 2 階層 PPC のスループットと消費電力

PPC は 552Gbps, 0.33W を実現している. すなわちテーブル分離 PPC により, スループットを 1.52 倍に向上し, 消費電力を 25%削減できることを示した.

各テーブルに対する PPC の容量比率によるテーブル検索性能への影響については, 本論文で評価した 3 パターンにおいては大きな違いは見られなかった. これは, 本論文で評価した PPC の総容量では, 3 パターンの間に大きな容量差が表れないためだと考えられる. 256KB 以上の総容量における結果を見ると, ルーティングテーブルに対する PPC の容量は 32KB に固定するよりも, より大容量とした方がスループット, 消費電力共に改善されることがわかる.

次に, 2 階層 PPC での評価結果を図 9 に示す. 図中のラベルにおいて, “統合+統合” は両階層を従来 PPC で構成するアーキテクチャを, “分離+分離” は両階層をテーブル分離 PPC で構成するアーキテクチャを, “分離+統合” は L1PPC をテーブル分離 PPC で, L2PPC を従来 PPC で構成するアーキテクチャを示す. 図 9 より, 2 階層 PPC においては, L1PPC と L2PPC をどちらもテーブル分離 PPC により構成することで, テーブル検索のスループットと消費電力を改善していることがわかる. これは L1PPC, L2PPC どちらにおいてもテーブル分離 PPC により容量効率が向上するためだと考えられる.

従来 PPC の最適な構成である 8KB の L1PPC, 2MB の L2PPC におけるテーブル検索性能 470Gbps, 0.29W に対

し, 32KB の L1PPC, 1MB の L2PPC のテーブル分離 PPC におけるテーブル検索性能は 691Gbps, 0.22W となった. すなわち, テーブル分離 PPC によりスループットが 1.47 倍に向上され, 消費電力が 24%削減された.

## 7. 関連研究

八巻らは, パケット処理キャッシュのエントリ置換アルゴリズムに着目し, パケット数の多いフローのエントリを長期間保持し, パケット数の少ないフローのエントリを迅速に追い出すエントリ置換アルゴリズムとして ELC(Elevator Cache) が提案している [10].

エントリ置換アルゴリズムとして一般的に用いられる LRU では新たなエントリを MRU(Most Recently Used) へ挿入するが, 1 度も参照されないエントリを追い出すのに最低 (ウェイト数) - 1 サイクルを要する. そのため, パケット処理キャッシュでは 1 パケットのみからなるフローのエントリを速やかに追い出すことができない問題がある. 実ネットワークにおけるトラフィック量の約 95% は 10 パケット以下からなるフローで構成されており, また, その多くが 1 パケットのみからなるフローである.

ELC は新たに挿入するエントリを LRU(Least Recently Used) へ挿入し, 参照されるごとに MRU 側のエントリを置換するエントリ置換アルゴリズムである. これにより, キャッシュヒットの可能性が高い多数のパケットで構成されるフローのエントリを優先的にキャッシュ内に残し, 少数のパケットで構成されるフローのエントリを迅速に追い出すことを可能にする.

ELC を用いることによって, パケット処理キャッシュのミス率を LRU と比べて平均 11.1%削減した.

## 8. おわりに

インターネットルータにおいて, パケット処理におけるテーブル検索を高スループット化, 省電力化するパケット処理キャッシュ (PPC) が提案されている. PPC は, パケットの 5 タプルをキーとして, Routing/ARP/ACL/QoS テーブルの検索結果を 1 エントリにまとめてキャッシュすることで, テーブル検索を効率化している. 一方で, ルーティングテーブル等の 5 タプル全てを検索キーとして用いないテーブルの, 同一の検索キーと結果が複数格納されてしまう点で, 容量効率が悪い.

そこで, 本論文では, これらのテーブルに対して個々の PPC を用意するテーブル分離 PPC を提案した. テーブル分離 PPC により, 主にルーティングテーブル, ARP テーブルに対するキャッシュ効率が向上する.

PPC シミュレータおよび実ネットワークパケットトレースを用いた評価では, 1 階層 PPC において, テーブル分離 PPC は従来 PPC に対して, スループットを 1.52 倍に向上し, 消費電力を 25%削減できることを示した. また, 2 階

層 PPCにおいても、両階層をテーブル分離 PPCにより構成することで、従来 PPC に対して、スループットを 1.47 倍に向上し、消費電力を 24%削減できることを示した。

**謝辞** 本研究は 2020 年度国立情報学研究所公募型共同研究 (20S1201), JSPS 科研費 JP18K18022 の助成を受けている。

## 参考文献

- [1] Cisco: Cisco Annual Internet Report (2018-2023) White Paper, Cisco (online), available from <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html> (accessed 2020-10-05).
- [2] Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017-2022, Cisco (online), available from <https://cyrekdigital.com/pl/blog/content-marketing-trendy-na-rok-2019/white-paper-c11-741490.pdf> (accessed 2020-10-05).
- [3] Eatherton, W., Varghese, G. and Dittia, Z.: Tree bitmap: hardware/software IP lookups with incremental updates, *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2, pp. 97-122 (2004).
- [4] Nawa, M., Okuda, K., Ata, S., Kuroda, Y., Yano, Y., Iwamoto, H., Inoue, K. and Oka, I.: Energy-efficient high-speed search engine using a multi-dimensional TCAM architecture with parallel pipelined subdivided structure, *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, pp. 309-314 (2016).
- [5] Gamage, S. and Pasqual, A.: High performance parallel packet classification architecture with popular rule caching, *2012 18th IEEE International Conference on Networks (ICON)*, IEEE, pp. 52-57 (2012).
- [6] Tanaka, K., Yamaki, H., Miwa, S. and Honda, H.: Evaluating architecture-level optimization in packet processing caches, *Computer Networks*, Vol. 181, p. 107550 (2020).
- [7] WIDE MAWI WorkingGroup: MAWI Working Group Traffic Archive, , available from <http://mawi.wide.ad.jp/mawi/> (accessed 2018-11-04).
- [8] Muralimanohar, N., Balasubramonian, R. and Jouppi, N.: Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0, *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, IEEE, pp. 3-14 (2007).
- [9] Pagiamtzis, K. and Sheikholeslami, A.: Content-addressable memory (CAM) circuits and architectures: A tutorial and survey, *IEEE journal of solid-state circuits*, Vol. 41, No. 3, pp. 712-727 (2006).
- [10] Yamaki, H. and Nishi, H.: Line replacement algorithm for L1-scale packet processing cache, *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, pp. 12-17 (2016).