トラフィックの時間的局所性を利用したネットワークプロセッサ・アクセラレータ

奥野通貴[†] 西宏章[††]

インターネットの普及にともない，ルータにおいて次世代の回線速度 100 Gbps（Giga bit per second）への対応が懸念されている．本稿では，ネットワークトラフィックの時間的局所性を利用して，パケット処理を高速化するネットワークプロセッサ（NP）アクセラレータを提案する．NP アクセラレータは，パケット処理そのものをキャッシュする Header Learning Cache（HLC）と，同一パケット処理によるプロセッサ資源の浪費を抑止する Cache Miss Handler（CMH）と呼ぶ機構を持つ．実トレースを利用した仮想 100 Gbps シミュレーションにより，アクセス網バックボーンルータで約 83%以上，広域網で約 55%以上の HLC ヒット率を確認した．特にアクセス網においては本方式により，20 Gbps 相当のプロセッサと組み合わせ，100 Gbps のパケット処理速度を期待できる．

Network Processor Accelerator Using Temporal Locality of Traffic

Michitaka Okuno[†] and Hiroaki Nishi[††]

The “Network Processor (NP) Accelerator” that is described in this paper is a novel architecture to accelerate packet processing throughput by using temporal locality of network traffic. The NP Accelerator includes two key components “Header Learning Cache (HLC)” to learn packet processing itself and “Cache Miss Handler (CMH)” to restrain dispatching same flow packets to the packet processors. We confirmed the access-edge backbone router traces showed approximately 83–98% HLC hit rate in simulation. Especially at the access-edge backbone router, the NP accelerator can achieve 100-Gbps packet processing throughput with 20-Gbps packet processors.

1. はじめに

近年のインターネットトラフィック増加にともない，通信インフラであるイーサネットを利用するバックボーンルータには，パケット高速転送能力と様々なサービスへの適応能力が要求されている．ハイエンドバックボーンルータでは，2002 年に登場した 10 Gbps 回線をサポートするために，パケット処理用プロセッサとして専用開発した ASIC（Application-Specific Integrated Circuit），もしくは，プログラム修整により開発後にも機能変更・拡張が可能なネットワークプロセッサ（Network Processor: NP）を利用している．

2003 年までに発表されている代表的なハイエンド NP を**表 1** に示す．これらの NP は，内蔵するパケット処理プロセッサ（Processing Unit: PU）を**図 1** に示すようにマルチプロセッサ化，マルチスレッド化し，VLIW（Very Long Instruction Word）の利用，処理のパイプライン化等を組み合わせ，パケットを並列処理することにより 10〜40 Gbps の高い処理速度を実現している[1),2)]．

ここで，約 4 年ごとに 10 倍高速化されてきたイーサネットの標準化動向から予想すると，ハイエンド NP に要求されるパケット処理速度は，2007 年頃に現在比 5〜10 倍にあたる 100 Gbps に達する．従来方式を延長し，PU 資源数の増加，または動作周波数の向上により 100 Gbps を達成する方法は，従来比約 5〜10 倍のダイサイズ増加，消費電力上昇を招き，事実上，実現が困難であると考えられる．

そこで，本稿ではトラフィックの特性を利用し，PU で処理するパケット数を削減して NP のスループットを向上させる方式，ネットワークプロセッサ（NP）アクセラレータを提案する．また，実トレースデータに基づく 100 Gbps の回線速度を想定したシミュレーションにより，現実的なチップサイズの NP アクセラレータで，特にアクセス網バックボーンルータにおいて 100 Gbps のパケット処理を期待できることを示す．

† 日立製作所中央研究所
Central Research Laboratory, Hitachi, Ltd.
†† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

表 1 代表的なハイエンド NP
Table 1 List of high-end network processors.

Product (Company)	処理速度 周波数	消費電力 プロセス	アーキテクチャ概要
X10q (Xelerated)	40 Gbps 200 MHz	11 W 0.13 um	200 ステージパイプライン，4way-VLIW
NP-1c (EZchip)	20 Gbps 250 MHz	15 W 0.13 um	処理を 4 ステージ化，64PU 配分
NPE10 (IMC)	20 Gbps 333 MHz	15 W 0.13 um	64PU 高速 DMA 相互結合
iPP (Silicon Acc.)	20 Gbps 333 MHz	14.5 W 0.13 um	32PU（256 マルチスレッド）
CGN16100 (Cognigine)	20 Gbps 200 MHz	18 W 0.18 um	6way-VLIW，16PU（64 マルチスレッド）
IXP2800 (Intel)	10 Gbps 1400 MHz	25 W 0.13 um	16PU（128 マルチスレッド）
np7510 (AMCC)	10 Gbps 500 MHz	10 W 0.18 um	6PU（144 マルチスレッド）
APP750 (Agere)	10 Gbps 266 MHz	12 W 0.16 um	処理をパイプライン化 4way-VLIW

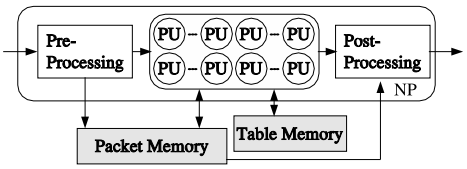

図 1 従来 NP の構成
Fig. 1 Structure of general network processor.

2. トラフィックの時間的局所性とその利用

ネットワークを流れるデータは，複数のパケットの連なりとしてネットワークを通過する．このため，文献 3)，4) で示されるように，短時間に同一ヘッダを持つパケットが多数出現しやすい特性，すなわち，時間的局所性を持つ．トラフィックの時間的局所性は，ネットワークの中心部に近付くほど大量の異なるパケットが混ざり合うため薄れてしまうが，末端近くのアクセス網やユーザ数が限定される LAN（Local Area Network）環境では非常に強く表れると考えられる．

時間的局所性の利用にはキャッシュが有効であるため，これまでも，宛先 IP（Internet Protocol）アドレスに対応する次ホップ情報をキャッシュし，ルーティングテーブル検索を高速化する研究は数多く行われてきた．たとえば，ゲートウェイネットワークアドレスをフルアソシアティブキャッシュに蓄える手法[5]，IP アドレスキャッシュ[6]，通常のプロセッサのキャッシュを利用する手法[7),8)]等があげられる．

本稿では，検索等の処理結果だけをキャッシュするのではなく，PU 処理そのもの，すなわち処理結果を再現するための情報をキャッシュし，PU で処理するパケット数を削減して NP 全体の処理速度を向上させる NP アクセラレータを提案する．また，同一ヘッダを持つパケットが連続してキャッシュミスを起こして PU 資源を浪費しないための仕組みを持つ Cache Miss Handler（CMH）と呼ぶ機構を提案する．

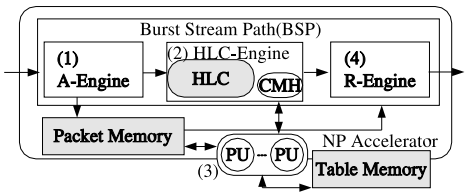

図 2 NP アクセラレータの構成
Fig. 2 Structure of NP accelerator.

3. ネットワークプロセッサアクセラレータ

3.1 NP アクセラレータの動作原理

NP アクセラレータは，トラフィックの時間的局所性を利用するために，Header Learning Cache（HLC）と呼ぶキャッシュメモリを備え，次の手法により，PU で処理するパケット数を削減して NP 全体でのスループット向上を図る．

(1) HLC 未登録パケットは PU 群で処理☆を行い，処理結果をビット列処理専用ハードウェア☆☆で再現するための情報を HLC に登録（学習）．
(2) HLC 登録済パケットは PU 処理をバイパスし，HLC から情報を引き出し，ビット列処理専用ハードウェアで高速処理．

HLC ヒット率を h，ビット列処理専用ハードウェア処理速度を s とすると，PU 群全体に要求される処理速度は $s(1-h)$ と表現できる．たとえば，専用ハードウェアを 320 bit 幅，333 MHz 動作とする．このとき，80%以上の HLC ヒット率が得られれば，現在実現可能な 20 Gbps 相当の PU 群で，NP アクセラレータ全体で処理速度 100 Gbps を実現できる．

3.2 NP アクセラレータのパケット処理概要

図 2 に NP アクセラレータの構成を示す．NP アクセラレータは，複雑なプログラム処理をする PU 群と Burst Stream Path（BSP）と呼ぶビット列処理専用パイプラインハードウェア，各種の情報を保持するテーブルメモリ，受信パケットを保持するパケットメモリに

☆ たとえば，ルーティングテーブル検索や，MPLS（Multiprotocol Label Switching），VLAN（Virtual LAN）等のラベル情報の処理，IPv4，IPv6 アドレス変換・カプセリング処理，フィルタリング処理等

☆☆ パケット処理は，途中経過は複雑でも最終的には検索結果等を参照してパケットに情報追加・削除・置換・変更する等のビット列操作に帰着できる．

より構成する．パケット処理の流れを順に説明する．

手順 1：BSP の先頭に位置する機能ブロック Analysis-Engine（A-Engine）でパケットの解析を行い，処理を簡易化するためにトークンと呼ぶ情報列を生成する．トークンは，固有情報（U-Info: 32 bit）☆1，解析情報（A-Info: 16 bit）☆2，抽出情報（E-Info: 384 bit）☆3の 3 情報（合計 432 bit）により構成する．入力パケットは内蔵の 4 MByte 程度の大容量パケットメモリに保存する．A-Engine は，432 bit 幅程度の比較器，バレルシフタ，マスク論理を多段に接続した構成を持ち，比較，シフト，マスク量をプログラムにより変更することで，パケットの任意部分を解析，抽出する．

手順 2：BSP の中間に位置する機能ブロック HLC-Engine で，トークンから生成したアクセスキーで HLC を参照する．HLC にヒットしたトークンは，HLC に登録されている処理結果を含んだトークンに置き換えられ，手順 3 をスキップし，手順 4 へ進む．

手順 3：HLC にミスしたトークンを PU 群でプログラム処理し処理情報（P-Info: 80 bit）☆4を追加する．アドレス変換やカプセリング，MPLS/VLAN 等のヘッダ追加・削除・置換等をともなう場合は，その改変情報を E-Info に上書きする．処理後のトークンは HLC に登録し，R-Engine へ渡す．

手順 4：BSP の最後尾に位置する R-Engine でトークンの U-Info を参照してパケットメモリから処理前のパケットを取り出す．そして A/P-Info を参照し，多段に接続した 320〜512 bit のバレルシフタ，マスク論理と演算器によりパケットに対しビット列操作を行い，送信パケットを組み立てなおす．このとき，アドレス変換や新規ヘッダの追加等が必要であれば E-Info に該当情報があるため，A/P-Info に従い，E-Info をヘッダの適切な位置に付加・置換等する．必要な修整を終えると，パケットを外部へ転送する．

3.3 NP アクセラレータでの PU 群の位置付け

PU 群は NP アクセラレータに内蔵する方式と，外付けにして既存の NP を利用する方式がある．外付けの NP を利用するメリットは，PU 群分の面積を，HLC やパケット保持用のメモリに使えること，PU 群全体での処理能力を後から変更できることである．デメリットとしては，パケットの送受信を NP アクセラレータと外付け NP 間で行うため，内蔵する場合に比べて HLC への登録が遅れ，同一ヘッダを持つパケットによる HLC ミスが発生する確率が高くなることが考えられる．

☆1 整理番号，パケットメモリへのポインタ等．
☆2 各レイヤでのパケット種別，エラー判定等．
☆3 IP アドレスや TCP ポート等のヘッダ部分の抽出情報．
☆4 ルーティングテーブル検索結果等のパケットヘッダに元々含まれない処理結果と Rebuilding-Engine（R-Engine）と呼ぶ機能ブロックで PU 処理を再現するための命令相当の情報．

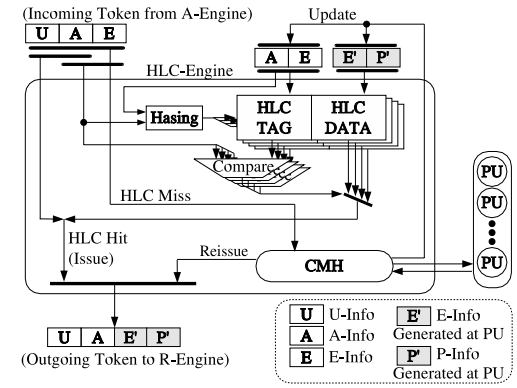

図 3 HLC-Engine の構成とトークンの流れ
Fig. 3 Structure of HLC-Engine and token flow.

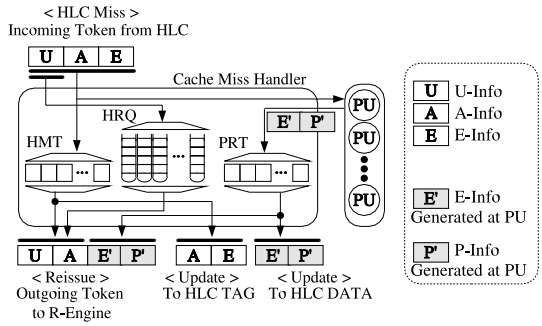

図 4 CMH の構成とトークンの流れ
Fig. 4 Structure of CMH and token flow.

4. Header Learning Cache Engine

NP アクセラレータでパケット処理速度を向上させる鍵となる HLC-Engine の構成とトークンの流れを**図 3**，**図 4** を用いて説明する．HLC-Engine はキャッシュ本体の HLC と，Cache Miss Handler（CMH）で構成する．

4.1 Header Learning Cache（HLC）

HLC は，PU での処理そのものを学習するためのキャッシュである．HLC は，タグとして処理前のトークンの A/E-Info を，データとして処理後のトークンの E/P-Info を記録する．HLC 参照には，A/E-Info 全体もしくは一部を縮約したアクセスキーを用いる．HLC タグに登録されている A/E-Info と参照に利用した A/E-Info が一致すると HLC ヒットとなる．ヒットすると PU 処理をバイパスし，参照に利用したトークンの U/A-Info に HLC データに登録されている E/P-

Info を結合したトークンを R-Engine に渡す（図 3 中の HLC Hit（Issue））．HLC タグに不一致の場合，HLC ミスとなり，元のトークンをそのまま CMH に渡す（図 3 中の HLC Miss）．HLC ミスが発生しても，停止要求がない限り後続のトークンは HLC 参照を続ける．

4.2 Cache Miss Handler（CMH）

CMH は，HLC ミスしたトークンを PU 群に渡して処理させ，処理済みトークンを R-Engine に渡す機能，および，HLC に登録する機能を持つ．また，PU 資源の浪費☆1と HLC ヒット率の低下☆2を防ぐため，同一の A/E-Info を持つトークンの PU 割当てを防ぐ機能を持つ．図 4 を用いて，CMH 処理を説明する．

(1) HLC ミスしたトークンの A/E-Info で，HLC ミスをおこしたトークンを管理するテーブルである HLC Miss Table（HMT）を調べる．A/E-Info が同一のトークンが登録されていなければ，当該トークンを HMT に登録し，PU に当該トークン処理を要求する．

(2) HMT に A/E-Info が同一のトークンがすでに登録されていれば，PU に処理要求を出さず U-Info だけを Header Reissue Queue（HRQ）と呼ぶキュー☆3に登録する．この動作により PU 資源の浪費を防ぐ．

(3) PU は，当該トークンに対応するパケットを処理し，トークンの E/P-Info を生成して，処理結果を一時的に記録する Processing Result Table（PRT）と呼ぶテーブルに登録する☆4．

(4) PRT が更新されると，対応する HMT の A/E-Info で HLC を参照，HLC タグの更新をし，PRT の E/P-Info で HLC データを更新する（図 3，4 中の Update）．該当 HLC エントリが埋まっている場合は上書きする．

(5) 同時に，まず HMT エントリの U/A-Info，次に対応する HRQ エントリ群の U-Info を順に取り出し，PRT の E/P フィールドを結合して R-Engine に送出する（図 3，4 中の Reissue）．Reissue により，HLC ヒット率の低下分をカバーする．Reissue した HRQ エントリは順次削除する．

(6) Reissue/Update 終了後，対応する HMT，PRT のエントリを削除し，当該トークンの処理を終了する．

☆1 PU 処理中トークンと A/E-Info が同一のトークンを PU 処理しても同一結果が得られるため，PU 資源の浪費となる．

☆2 PU 処理中トークンと A/E-Info が同一のトークンは，PU でのトークン処理完了まで HLC ミスとなる．

☆3 HMT の各エントリに対応した FIFO（First In First Out）構造の HRQ がある．HRQ の構成については 5.5 節で考察する．

☆4 HMT の各エントリに対応した PRT エントリがある．

表 2 トレース採取サイト一覧
Table 2 Site list.

KEY	SITE	匿名化	link (bps)
WIDE	WIDE trans-Pacific line B	suffix のみ	100 M
HCRL	Hitachi Central Lab	なし	100 M
A-IV	Univ. of Auckland uplink	あり	155 M
IPLS	Abilene Indianapolis router	あり	2.5 G

KEY	DATE	#of packets
WIDE	2003/2/27 全日	26,313,826
HCRL	2002/Apr-Jul 上位 34 日間	11,668,988
A-IV	2001/3/27 全日	30,000,000
IPLS	2002/8/14 全日	20,000,000

WIDE: http://tracer.csl.sony.co.jp/mawi/
HCRL: 日立製作所中央研究所
A-IV: http://wand.cs.waikato.ac.nz/wand/wits/auck/4/
IPLS: http://pma.nlanr.net/PMA/,
http://www.internet2.edu/resources/AbileneMap.pdf

なお，R-Engine に渡すトークンは，パケットの順序性維持のため，HLC ヒットして Issue されたトークンより CMH からの Reissue トークンを優先する．

5. NP アクセラレータの評価

NP アクセラレータは，ルーティングテーブル情報以外にも，ポリシー情報やフィルタリング情報等様々な情報を HLC にキャッシュし，R-Engine に当該パケットの組立指示や廃棄処理指示等を出せるが，ここでは，評価として広く利用されているルーティングテーブル検索処理に焦点を絞った．本章では，**表 2** に示す特徴のある 4 種類のサイト☆5から取得した実トレースを用いて，NP アクセラレータの HLC，CMH に関する評価を，機能レベルシミュレーションにより実施した．

5.1 トラフィックの扱いについて

表 2 に示すサイトの回線速度は，IPLS の 2.5 Gbps を除き，100 Mbps または 155 Mbps と，想定する 100 Gbps の回線速度に対して非常に小さい．そこで，仮想 100 Gbps 回線として扱うために，次の 2 つのケースを想定してシミュレーションを行った．

(1) ネットワークの末端がトレース採取時より 40〜1000 倍高速になると仮定し，採取トレースのパケット到着時間を 40〜1000 倍速くして

☆5 WIDE は，ルートネームサーバ等様々なサービスと，NSPIXP（Network Service Provider Internet eXchange Point）等のエクスチェンジポイントを含む広域網の代表で，国外向け回線のトレースである．HCRL は，1000 人程度の規模を持つ研究開発部門のインターネット向けパケットのトレースで，アクセス網（小規模サイト）の代表である．A-VI は，Auckland 大学（3000 人のスタッフと，27000 人の学生が所属）および地域プロバイダのトレースで，アクセス網（中規模サイト）の代表である．IPLS は，米国全土を網羅する Abilene Internet2 バックボーンにおける Indianapolis と Kansas City 間ネットワークのトレースで，広域網の代表である．

表 3 検討した HLC パラメータ
Table 3 HLC parameters.

アソシアティビティ	1/4way（置換は LRU 方式）
総エントリ数	256 エントリから 256 K エントリまでの 2 の冪乗
アクセスキー生成方式	RAW 方式/CRC 方式

100 Gbps 相当にする．以下，このトレースを「単純圧縮」（図中では CMP）と表現する．

(2) ネットワークの末端がトレース採取時より 10 倍高速になると仮定し，採取トレースのパケット到着時間を 10 倍速くする．さらに採取トレースを分割し☆1，それぞれの先頭を揃えて集線し 100 Gbps 相当にする．以下，このトレースを「混合圧縮」（図中では MRG）と表現する．

5.2 HLC の構成に関する評価

表 3 に示すパラメータ範囲で HLC の構成を変え，HLC ヒット率を調べた．本評価における HLC は，キャッシュミスすると即座に処理済みトークンを HLC に登録する理想的なモデルである．

比較のため，HLC 参照アクセスキーの生成には従来どおり IP アドレスをそのまま利用する[4),5),7)]RAW 方式☆2と，CRC 演算の剰余を利用する☆3CRC 方式☆4を用いた．いずれも，IP アドレスを含む E-Info を利用するため，IP アドレスが匿名化されていないトレースを用いた方が正確な評価になると考え，HCRL のトレースを利用した．

図 5 に結果を示す．横軸は HLC の総エントリ数，縦軸は HLC ヒット率を示す．調査した HLC エントリ全範囲で，単純圧縮では 97.7%以上，混合圧縮でも 89.4%以上の高い HLC ヒット率を示した．これは，HCRL が非常に時間的局所性の高いトラフィックであり，512 エントリ以上あれば 10 Gbps 相当の PU 群と組み合わせて 100 Gbps を実現できることを示している．HLC のタグとデータに必要なメモリ量は，1 エントリあたり約 1 Kbit☆5なので，以降の評価では実装を想定し HLC のエントリ数を 4 K（約 500 KByte）とする．

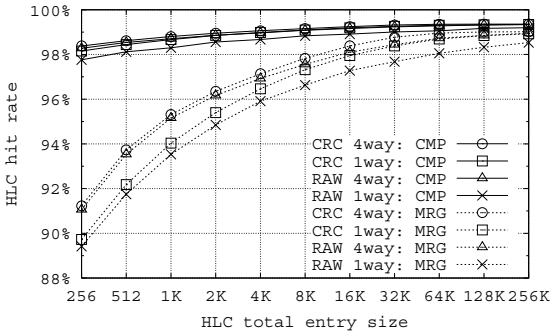

図 5 HCRL における HLC ヒット率
Fig. 5 HLC hit rate at HCRL.

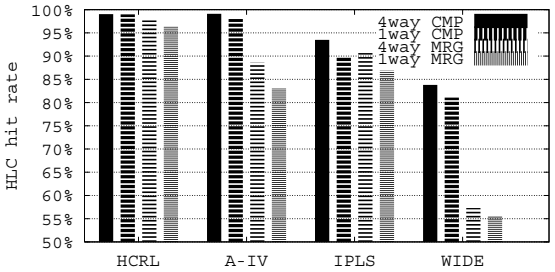

図 6 各サイトでの HLC ヒット率
Fig. 6 HLC hit rate at each site.

アソシアティビティは 4way が，アクセスキー生成方式は CRC 方式がつねに高い HLC ヒット率を示した．特に 1way の CRC 方式は，単純圧縮では 1 K エントリ以上，混合圧縮では 32 K エントリ以上で RAW 方式の 4way と同等の HLC ヒット率を示したことから，高い攪拌能力を期待できることが分かった．

5.3 各サイトでの HLC ヒット率の評価

HLC 参照アクセスキーの生成方式を CRC 方式，エントリ数を 4 K に固定し，5.2 節と同じ条件で，表 2 の全サイトにおける HLC ヒット率を調べた．結果を**図 6** に示す．棒グラフは，左から順にサイトごとの 4way-単純圧縮，1way-単純圧縮，4way-混合圧縮，1way-混

☆1 A-IV は 24 時間分のデータを 64 分割，IPLS は 4 分割した．HCRL は 34 日分のデータからアクセスの多い 9–11 時，13–15 時，15–17 時のデータを抜き出し，上位 100 位分を利用した．WIDE は 15 分間隔で採取されている 96 本のデータをそのままマージしたため，換算後は 96 Gbps 相当である．

☆2 QoS（Quality of Service）の観点から，パケット群をフローとして扱うために，送信元，宛先の両 IP アドレスを利用した．すなわち，HLC のエントリサイズが 2^{2N+A}（A は 0 または 1）であるとき，送信元 IP アドレスの下位から N ビット，宛先 IP アドレスの下位から N+A ビットを連結して HLC 参照アドレスとした．

☆3 CRC 演算には次の式を用いた．たとえば（128 K:17,3）は，128 K エントリの HLC で用いた CRC 演算の剰余の生成多項式が $x^{17} + x^3 + 1$ であることを表す．生成多項式 = (32:4,1) (64:5,2) (128:7,3) (256:8,6,3,2) (512:9,5) (1 K:10,7) (2 K:11,2) (4 K:12,3) (8 K13,4,3,1) (16 K:14,5) (32 K:15,1) (64 K:16,5,3,1) (128 K:17,3) (256 K:18,3)

☆4 CRC 演算は 64B66B の符合化にも利用されるように撹拌能力に優れているため，HLC でエントリ競合を起こしにくくする利点があると予想される．また，トークンの A/E-Info の内容に依存せず，同一のアルゴリズムを用いて $\log_2$（bit 幅）段の XOR 論理で構成できるため，ハードウェア化が容易である利点もある．

☆5 タグは A/E-Info（16+384=400 bit），データは E/P-Info（384+80=464 bit）．8 bit ごとに 1 bit のパリティを付加し，(400+464) bit×9 bit/8 bit=972 bit．

合圧縮での HLC ヒット率を示す．

幅広いユーザが利用する広域網の HLC ヒット率は，IPLS では単純圧縮でも混合圧縮でも 86〜93%と高めだったが，WIDE では，単純圧縮データで 81〜83%，混合圧縮では 55〜57%程度と低めだった．WIDE の混合圧縮で 100 Gbps を実現するには，PU 群に現在のハイエンド NP の処理速度を上回る 45 Gbps 以上の処理速度が必要とされ，4 K エントリを超える大容量の HLC 実装が必要である．つまり，広域網で 4 K エントリの HLC を利用する場合，PU 群の処理速度を 2 倍程度引き上げることは期待できるが，100 Gbps の処理速度の実現は困難と考えられる．

一方で，ユーザ規模がある程度限られるアクセス網の HCRL や A-IV では単純圧縮で 98〜99%，混合圧縮でも 83〜97%の高い HLC ヒット率が得られた．これより，NP アクセラレータは，企業やデータセンタ，アクセス網のバックボーン等，ユーザ数は限定されるが大量のデータを高速に送受信する必要がある部分のルータに適用することにより，現在のハイエンド NP 相当の 20 Gbps の PU 群を組み合わせて 100 Gbps の処理速度を期待できる．

5.4 HLC，HRQ によるトークン処理数削減効果

次に，PU での処理時間をパラメータとし，HLC と CMH 内部の Header Reissue Queue（HRQ）により PU で扱うトークンのピーク数をどの程度削減できるか調べた．HLC は，4way，4 K エントリ，CRC 方式アクセスキー生成とし，CMH 内部の HLC Miss Table（HMT）と HRQ のエントリ数は無制限とした．トレースには HCRL と WIDE を用いた．

図 7 上段に HCRL の結果を，図 7 下段に WIDE の結果を示す．「NO HLC」は HLC がない場合，「ONLY HLC」は HLC だけある場合，「HLC+HRQ」は HLC と HRQ がある場合を示す．横軸は PU 群でのトークン 1 つあたりの処理時間，縦軸は PU 群で同時処理されるトークンの数を示す．左側の図は，PU 群を NP アクセラレータに組み込む場合を想定し，処理時間を 100〜800 サイクル☆とした．右側の図は，PU 群を外付けにする場合を想定し，処理時間を 800〜12800 サイクル☆☆とした．

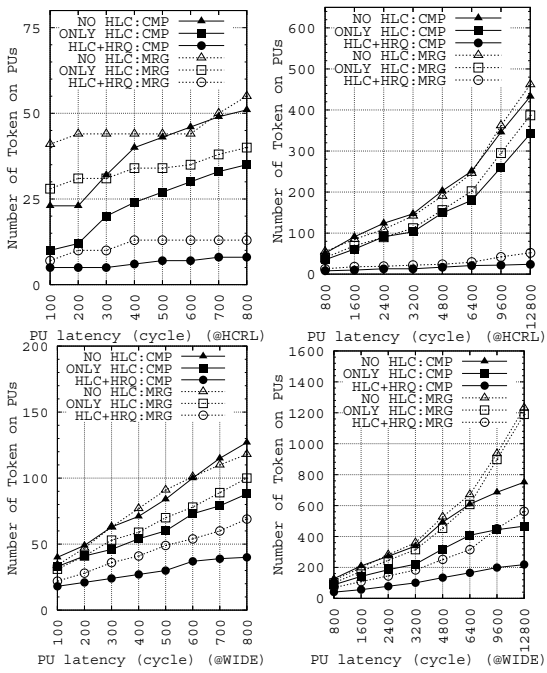

図 7 PU 群で同時処理するトークン数（HCRL，WIDE）
Fig. 7 The number of simultaneously processed token on PUs at HCRL and WIDE.

図 7 の結果より，HLC だけでは PU 群での処理時間が増えるにつれ，PU 群で処理するトークン数削減が困難になるのが分かる．これは，HLC 登録までにトークンの処理が終了しない可能性が高くなるためと考えられる．しかし，HRQ を追加すると，HLC だけを持つケースからさらに 2 割〜7 割程度 PU 群で処理するトークン数を削減できている．同一ヘッダのパケットが連続するバースト性のあるトラフィックがあるケースや，PU での処理時間が長いケースでは，PU 群で処理するトークン数の削減に HRQ が効果的であると結論づけられる．

次に，4 進のパトリシアツリーを利用したルーティングテーブル検索で，参照回数に応じて PU の処理時間を変化させて評価を行った．1 サイクルを 3 ns とし，1 回のテーブル検索（メモリ参照）に 30 サイクル，メモリ参照以外のオーバヘッドを PU 群を組み込む場合は 30 サイクル，PU 群を外付けにする場合は 3600 サ

☆ 文献 9) から，IPv4 パケットのルーティングテーブル検索処理のメモリ参照以外のオーバヘッドを 30 サイクルとした．32 bit 幅，ランダムアクセス時間 25 ns の FCRAM（Fast-Cycle RAM）をテーブルメモリとし，1 エントリに 64 bit を使う．4 進のパトリシアツリーを利用すると 1 回のテーブル検索には 256 bit が必要．256 bit の取得には 2 回 FCRAM にアクセスし 30 サイクルかかるとした（333 MHz 動作では 25 ns は 9 サイクルに相当．バーストアクセス分で 3 サイクル，その他のオーバヘッド 3 サイクルを加え 15 サイクル．2 回分で 30 サイクル）．最短約 100 サイクル（30 + 30 × (2〜3 回参照)）を想定した．

☆☆ 文献 10) によると，Intel 社の IXP2800 では，IPv4 パケットの先頭が到着し，最後尾が転送されるまでの平均レイテンシは約 10.8 μs である．333 MHz の NP アクセレラータを想定すると 10.8 μs は 3600 サイクルに相当する．より複雑な処理を実施することも想定し，12800 サイクルまでを調べた．

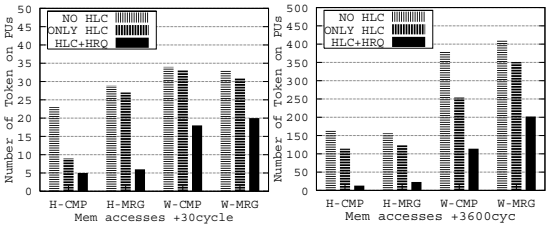

図 8 PU 群で同時処理するトークン数（H:HCRL,W:WIDE）：ルーティングテーブルを利用

Fig. 8 The number of simultaneously processed token on PUs at HCRL and WIDE: Routing Table was used.

イクルとした．

図 8 左は PU 群を組み込む場合，図 8 右は PU 群を外付けにする場合の結果であり，縦軸に PU 群で同時に処理されるトークン数を示す．左から，HCRL の単純圧縮で HLC なし，HLC のみ，HLC と HRQ ありの場合，次の 3 つが HCRL の混合圧縮の場合，同様に WIDE の単純圧縮，混合圧縮と続く．

ルーティングテーブル検索の平均回数は，HCRL が 3.75 回（約 112 サイクル），WIDE が 7.31 回（約 219 サイクル）であった．図 7 と比較した場合，大きな差は発生せず，HRQ が HCRL のようなトラフィックの局所性が強く表れやすいアクセス網において特に有効であることが示された．

5.5 HMT，HRQ に関する評価

5.4 節後半のルーティングテーブルを参照するシミュレーションで，CMH を構成する HLC Miss Table（HMT）と Header Reissue Queue（HRQ）に必要なエントリ数および HRQ に適する構成を考察した．**図 9** 上は PU 群を組み込んだ場合，図 9 下は PU 群を外付けにした場合の HMT，HRQ の利用状況を示す．横軸は，HLC ミストークンを割り当てる HMT のエントリ番号を示すと同時に，複数ある HRQ の ID 番号も示している．なお，ID 番号の小さい空きエントリから順にトークンを割り当てるものとした．縦軸は，各 ID の HRQ において，ピーク時に必要となるエントリ数を示す．

PU 群を組み込む場合，最大 19 エントリの HMT（WIDE-MRG），最大 21 エントリの HRQ（HCRL-MRG），最大総計 21 エントリの HRQ（HCRL-MRG）が利用された．PU 群を外付けにする場合，最大 202 エントリの HMT（WIDE-MRG），最大 137 エントリの HRQ（WIDE-CMP），最大総計 149 エントリの HRQ（WIDEL-MRG）が利用された．

HRQ は，(HRQ の ID の数) × (最も利用された HRQ のエントリ数) を実装する方法もある．しかし

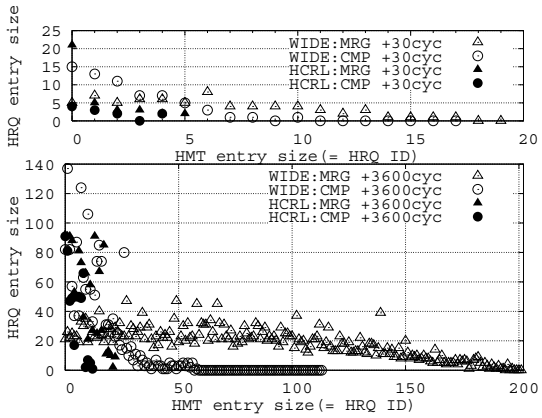

図 9 HMT，HRQ エントリ数（HCRL，WIDE）

Fig. 9 The number of HMT and HRQ entries on HCRL and WIDE.

表 4 HLC，CMH のメモリ量見積もり

Table 4 HLC, CMH memory estimation.

HLC	TAG:	450 bit/entry	225 KB (4 K entry)
	DATA:	522 bit/entry	261 KB (4 K entry)
CMH	HMT:	432 bit/entry	1.7 KB (32 entry)
（PU 処理時間	HRQ:	32 bit/entry	128 B (32 entry)
250 cyc 程度）	PRT:	464 bit/entry	1.8 KB (32 entry)
CMH	HMT:	432 bit/entry	13.5 KB (256 entry)
（PU 処理時間	HRQ:	32 bit/entry	1 KB (256 entry)
3800 cyc 程度）	PRT:	464 bit/entry	14.5 KB (256 entry)

ながら，図 9 から各 HRQ に必要なエントリ数にはばらつきがあるため，メモリ量を節約するならば，HRQ はフルアソシアティブ構成が適していると考えられる．HMT，HRQ の実装量は，PU 群の実装位置，また，PU 群の処理時間をどの程度と想定するかにより異なる．PU 群組み込み（処理時間 250 サイクル程度），外付け（処理時間 3800 サイクル程度）のケースそれぞれで，HMT は 32，256 エントリ，HRQ は 32，256 エントリ程度あれば実用上問題ないと考えられる．

5.6 HLC，CMH のハードウェア量に関する考察

NP アクセラレータに搭載する HLC☆，CMH のハードウェア量見積りを**表 4** に示す．NP アクセラレータは，HLC および CMH として約 500 KByte のメモリ，BSP 論理，PU 群として従来の NP 相当の論理を集積して実現する．ダイサイズ増加は，0.13 um プロセスを利用したとき BSP 論理として約 22 mm^2 ☆☆，

☆ 8 bit ごとに 1 bit パリティを想定．

☆☆ 機能的には表 1 の EZchip 社 NP-1c の TOPperse（12PU）および TOPmodify（8PU）に相当．NP-1c のダイサイズを 100 mm^2 強と仮定した．NP-1c は 4 MByte の組み込み DRAM に約 30 mm^2 を利用しているため，組み込みの 64PU は約 70 mm^2 を占めると考えた．20PU は約 22 mm^2 に相当．

HLC と CMH に約 10 mm^2 ☆，合計約 32 mm^2 程度と見積もっている．また，NP アクセラレータは，メモリ（HLC と CMH）参照により本来 PU 群で実施されるパケット処理を削減するため，チップ全体での消費電力は，同等の処理を行う従来方式の NP より少なくなることが期待される．

6. むすび

ネットワークトラフィックに存在する時間的局所性を活用し，プロセッサでの処理パケット数を削減して NP 全体のパケット処理速度を向上させる NP アクセラレータを提案した．NP アクセラレータは，Burst Stream Path（BSP）と呼ぶ専用パイプラインハードウェアを利用して，プロセッサ処理そのものを Header Learning Cache（HLC）と呼ぶキャッシュに記録し，再利用することでプロセッサ処理を削減する．また，Cache Miss Handler（CMH）に備える Header Reissue Queue（HRQ）と呼ぶ機構により，HLC 登録までに HLC ミスする同一ヘッダを持つパケットでプロセッサ資源が浪費されることを防ぐ．

アクセス網，広域網の実トレースを単純圧縮および混合圧縮して生成した仮想 100 Gbps トレースを用い NP アクセラレータを評価した．4 K エントリ，4way，CRC 演算を利用したアクセスキー生成方式の HLC により，アクセス網では，単純圧縮で約 98%以上，混合圧縮でも約 83%以上の HLC ヒット率が得られた．つまり，約 5 倍のスループット向上を期待できるため，現在のハイエンド NP 相当の 20 Gbps の処理速度を持つプロセッサ群と 320 bit 幅 333 MHz 動作の BSP を組み合わせて 100 Gbps の処理速度を期待できる．広域網では，4 K エントリの HLC では，100 Gbps の実現は困難であるが，2 倍程度のスループット向上を期待できる．

以上より，NP アクセラレータは企業やデータセンタ，アクセス網のバックボーン等，ユーザ規模が数万程度に限られ，大量のデータを 100 Gbps レートで転送する必要のあるルータに利用すると特に有効である．

参考文献

1) Crowley, P., Franklin, M., Hadimioglu, H. and Onufryk, P.: *Network Processor Design — Issues and Practices Volume 1*, Morgan Kaufmann (2002).
2) Gwennap, L. and Wheeler, B.: *A Guide to NETWORK PROCESSORS Fourth Edition*, The Linley Group (2002).
3) Jain, R.: Characteristics of Destination Address Locality in Computer Networks: A Comparison of Caching Schemes, *Computer Networks and ISDN Systems*, Vol.18, pp.243–254 (1990).
4) Chvets, I. and MacGregor, M.: Multi-zone Caches for Accelerating IP Routing Table Lookups, *High Performance Switching and Routing* (*HPSR*) *2002*, pp.121–126 (2002).
5) Feldmeier, D.: Improving Gateway Performance with a Routing Table Cache, *INFOCOM'88*, pp.298–307 (1988).
6) Talbot, B., Sherwood, T. and Lin, B.: IP Caching for Terabit Speed Routers, *IEEE GLOBECOM 99*, Vol.2, pp.1565–1570 (1999).
7) Chiueh, T.C. and Pradhan, P.: High-Performance IP Routing Table Lookup Using CPU Caching, *INFOCOM'99*, pp.1421–1428 (1999).
8) Chiueh, T.C. and Pradhan, P.: Cache Memory Design for Network Processors, *IEEE High Performance Computer Architecture Conference* (*HPCA*) *2000* (2000).
9) Partridge, C., Carvey, P., Burgess, E., Castineyra, I. and Winterble, S.: A Fifty Gigabit Per Second IP Router, *IEEE/ACM Trans. Networking*, Vol.6 (1998).
10) Meng, D., Gunturi, R. and Castelino, M.: IXP2800 Intel Network Processor IP Forwarding Benchmark Full Disclosure Report for OC192-POS, *Network Processing Forum* (2003).
11) International Technology Roadmap for Semiconductors: 2002 Update ITRS (2002).

(平成 15 年 10 月 6 日受付)
(平成 16 年 2 月 13 日採録)

奥野 通貴

1998 年慶應義塾大学大学院理工学研究科修士課程修了．同年（株）日立製作所入社，エンタープライズサーバ事業部勤務を経て 2003 年より同社中央研究所勤務．

☆ 文献 11) より，0.13 um プロセスの 6T-SRAM（6 Transistor Static Random Access Memory）の密度は，2.37 M Tr/mm^2.
(2.37 M Tr/mm^2)/(6Tr×8 bit)=50.6 KByte/mm^2.

西　宏章（正会員）

1999年慶應義塾大学大学院理工学研究科後期博士課程修了．同年技術研究組合新情報処理開発機構，2002年（株）日立製作所中央研究所勤務を経て，2003年より慶應義塾大学理工学部システムデザイン工学科助手，工学博士．