通信の時間的局所性を利用した広帯域低消費電力向けネットワークプロセッサアーキテクチャに関する研究

2007年度

奥野 通貴

論文要旨

2000年代初頭にかけて，多数の簡易なプロセッサを集積してパケット処理を高スループットかつプログラマブルに実現するネットワークプロセッサ (NP：Network Processor) がネットワーク転送装置向けに登場した．次世代ハイエンド NP では，現在より 10 倍高速な半二重 100Gbps 回線の処理が求められるが，内蔵プロセッサ数を増加させることでスループットを向上させる従来の手法では，チップサイズや消費電力が要求性能に比例して増大し問題となる．

本研究では，この問題を解決するアーキテクチャとして，短時間のうちに同一のヘッダ情報を持つパケットがネットワーク上に多数出現しやすいという通信の時間的局所性を利用するキャッシュ型ネットワークプロセッサ CBNP (Cache-based Network Processor) を提案した．CBNP は，パケットの解析と組立処理を行なう BSP (Burst Stream Path) と呼ぶメインパス中にプロセスラーニングキャッシュ (PLC：Process Learning Cache) と呼ぶ専用のメモリと通常の NP に相当する複数のプロセッシングエレメント (PE：Processing Element) を備える．そして，パケット処理のボトルネックとなる各種の検索処理 (テーブルルックアップ) やカプセリングヘッダ生成等を初回のパケットのみ PE で実施し，その結果と，結果の適用方法を PLC に記録する．後続の同一とみなせるヘッダを持つパケットには，PLC の内容を BSP 中で再利用することで内蔵プロセッサ数を増加させなくても高スループット化を実現できる．また，CBNP はキャッシュミスハンドラ (CMH：Cache Miss Handler) と呼ぶ機構を備え，PLC 登録処理中に到着する PLC 未登録の後続パケットを保持して PLC 登録後に結果を適用することで消費電力の高い PE の利用を抑止しつつ，PLC アクセスをノンブロッキング化する．

CBNP の機能を確認するために，ハードウェア記述言語 Verilog HDL で CBNP を記述して FPGA ハードウェアエミュレータを構成し，複数のサイトで採取された実ネットワークトラフィックトレースをワイヤレートで与えてスループット評価を行った．この評価より，CBNP は，4K エントリの PLC と 1K エントリの CMH，及びアクセス網では回線速度の 1 割程度，コア網でも 4 割程度を処理できる PE を内蔵するだけで，回線速度と同じパケット処理スループットを実現できることを確認した．更に，多数の PE を集積する従来型の NP と，CBNP のそれぞれの場合について，100Gbps 回線処理時の主要構成要素の面積と消費電力の見積もりを行い，CBNP が従来型の NP に比べて半分以下の面積及び消費電力で実現できる見通しを得た．これらの評価を通じて，CBNP は従来型の多数の PE を集積してスループットを確保するネットワークプロセッサよりも消費電力効率の良いアーキテクチャとして期待できる．

Abstract

Network processors (NPs) are programmable devices that use many simple built-in processors to optimize packet processing on network equipment. Although next generation NPs will require even higher throughput, conventional NP architecture, in which the number of built-in processors is increased, would lead to excessive die size and power-consumption problems.

In response to these problems, I have proposed an alternative NP architecture called a " cache-based network processor " (CBNP) that realizes higher packet-processing throughput with low power consumption by using the locality of network traffic. Since data is divided into several packets and is transmitted in network, many packets that have the same header information tend to appear in a short period. The CBNP is equipped with a memory called a " process-learning cache " (PLC) and a relatively small number of built-in processors as its main logic. Only bottleneck processes such as table lookup of the first packet in every packet flow are processed by the built-in processors, and the results such as lookups are recorded in the PLC with commands for using them. The following packets that have the same header information are processed by using PLC contents without built-in processors. The CBNP also has a cache-miss handler that holds the following packets that arrive during PLC registration and then applies the contents of PLC in order to realize non-blocking PLC access.

A hardware emulator for the CBNP was developed to evaluate its packet-processing throughput with real network-traffic traces. The evaluation results demonstrate that CBNP can achieve a wire-rate packet processing when fitted with a 4K-entry PLC, 1K-entry CMH, and built-in processors that could provide about 10% of the line speed for the access network and 40% of the line speed for the core network. Moreover, according to an estimation of gate-level area and power consumption, the CBNP is expected to have an area and power consumption below half that of a conventional NP. We therefore expect that the CBNP is an applicable NP architecture for future network equipment requiring high throughput and low-power consumption.

目 次

表 目 次

図 目 次

第1章　緒論

1.1　本研究の目的

インターネットトラフィックは増加の一途を辿っており，基幹ネットワークでは，これらの大量のパケットを高速に交換するためのネットワーク転送装置 (ルータ，レイヤ 3 スイッチ) が利用されている．現在の代表的なハイエンドルータは筐体あたり 10Gbps 回線を 32 本程度交換可能であり，ルータ内のパケット処理に利用されるネットワークプロセッサは，半二重 10Gbps〜40Gbps 程度のスループットを備えている．早ければ 2009 年末に 100Gbit Ethernet の登場が予想されるため，ネットワークプロセッサには，最低でも半二重 100Gbps のスループットが要求されることになる．しかしながら，従来のネットワークプロセッサのアーキテクチャを踏襲した場合，LSI ダイサイズ及び消費電力の点で問題点が生じうる．

従来型のネットワークプロセッサの大多数は，個々のパケットが独立に処理可能であることを利用し，内部に大量のプロセッシングエレメント (PE) を集積したり，マルチスレッド機構を搭載したり，または，その両者を実現し，パケットレベルの並列処理を行なうことで高スループット化を実現したきた [2, 3, 4]．最先端のネットワークプロセッサは，32 個程度の PE，0.13μm 程度の CMOS プロセス，300MHz 程度の動作周波数で，半二重 20Gbps 程度のパケット処理スループットを 15W 程度の消費電力で実現している．従来のアーキテクチャの延長線上で，近い将来必要となる半二重 100Gbps のパケット処理を実現するためには，並列度の更なる向上が必要となる．

従来は，半導体のスケーリング則 [5] を利用して半導体を微細化し，集積度を向上させることで得られたトランジスタによって回路を並列化し，スループットを増加させてきた．同時に，周波数は大きく向上させず電源電圧も小さくすることで消費電力の急激な増加を防ぐことができた．しかしながら，0.13μm 未満の半導体プロセスでは，これまで無視できていたリーク電流による電力消費が無視できなくなってきており，半導体のスケーリング則が少なくとも消費電力の面で成立しなくなりつつある．このため，トランジスタレベルの改善を実現するための様々な研究がなされているが，本研究では，アーキテクチャレベルでのアプローチによって問題を解決することを目指した．すなわち，従来の PE 数を増加させる (並列度を向上させる) 以外のアプローチで消費電力を増加させずに高スループット (広帯域) を実現するためのネットワークプロセッサアーキテクチャを確立することを目的とした．

そこで，本研究では，短時間のうちに同一のヘッダ情報を持つパケットがネットワーク上に多数出現しやすいという通信の時間的局所性に着目した．そして，パケット処理のボトルネックとなる各種の検索処理 (テーブルルックアップ) やカプセリングヘッダ生成等を初回のパケットのみ PE で実施し，その結果と，結果の適用方法を記録し，後続の同一とみなせるヘッダを持つパケットに記録内容を適用することで，消費電力増加の原因となる PE 数を増やさずにスループットを向上させるネットワークプロセッサアーキテクチャCache-based Network Processor (CBNP) を提案する．尚，処理結果と適用方法を生成，記録中にも後続のパケットがネットワークから次々と到着するため，CBNP では，それらの後続パケットの処理を滞らせないための仕組みも合わせて提供する．まず，CBNP のアーキテクチャと構成を説明し，実際のネットワークトラフィックトレー

スを利用した評価により，CBNP が高スループットのパケット処理を実現できることを確認する．また，CBNP と従来型ネットワークプロセッサの LSI ダイサイズと消費電力を見積もり，CBNP が従来型ネットワークプロセッサより，小サイズ，低消費電力で実現できることを確認する．

1.2 本論文の構成

本論文は 6 章構成となっており，本章以下，第 2 章では，背景となるネットワークトラフィックとルータ，ネットワークプロセッサについて紹介し，次世代ネットワークプロセッサの高スループット化への課題を明らかにする．そして，その解決策として通信の時間的局所性を利用する方法を述べ，関連研究について紹介する．

第 3 章では，通信の時間的局所性を利用したネットワークプロセッサアーキテクチャCBNP (Cache-based Network Processor) を提案する．特に通信の時間的局所性を利用し，ネットワークから次々と到着するパケットを滞らせることなく従来型ネットワークプロセッサよりも少ない PE 群で処理するための PLC (Process Learning Cache) と CMH (Cache Miss Handler) と呼ぶ機構を中心に説明する．また，予備評価を通じて CBNP の有効性を示す．

第 4 章では，CBNP のプロトタイプを紹介する．そして，実際のネットワークトラフィックを利用し，条件的に最も厳しいワイヤレート下における評価を通じて，CBNP が高スループットのパケット転送処理を実現できることを示す．

第 5 章では，CBNP と従来型ネットワークプロセッサのコスト比較として，いくつかの前提条件におけるダイサイズと消費電力の比較を行なう．その結果を元に，CBNP のコスト優位性について論じる．

最後に第 6 章で，ネットワーク機器において近年懸念されている消費電力の増加問題の視点を交えながら，CBNP の将来展望と結論を述べる．

また，付録として，CBNP のプロトタイプで実装した各種トークンのフォーマットの詳細，設定レジスタの詳細，ネットワークプロセッサで参照しうるレイヤ 2〜4 の代表的なプロトコルのヘッダフォーマット，NP 周辺の主要な I/O インタフェースを掲載する．

第2章　背景，関連研究

本章では，本研究の背景となるネットワークトラフィックとルータ，ネットワークプロセッサ (Network Processor: NP) について紹介する．また，本研究に関連するネットワーク通信の局所性を利用した研究についても紹介する．

2.1　ネットワークトラフィックの増加

1990 年代後半以降，一般個人や民間企業のインターネット利用が加速度的に増えた結果，ネットワークのトラフィック量は増加の一途を辿っている．ネットワークのトラフィック量の増加に関する予測として，通信容量 (通信帯域) は半年間で 2 倍 (10 年で 100 万倍に相当) の成長を遂げるというギルダーの法則 [6] や，ネットワークの性能は 1 年で 2 倍 (10 年で 1000 倍に相当) になるというビルジョイの法則が知られているが，いずれも 18ヶ月で半導体の集積度は 2 倍になるというムーアの法則 [7] を超えた飛躍的な増加量である．ネットワークのトラフィック量の増加は，経路数の増加と回線速度の高速化の両方によってもたらされている．

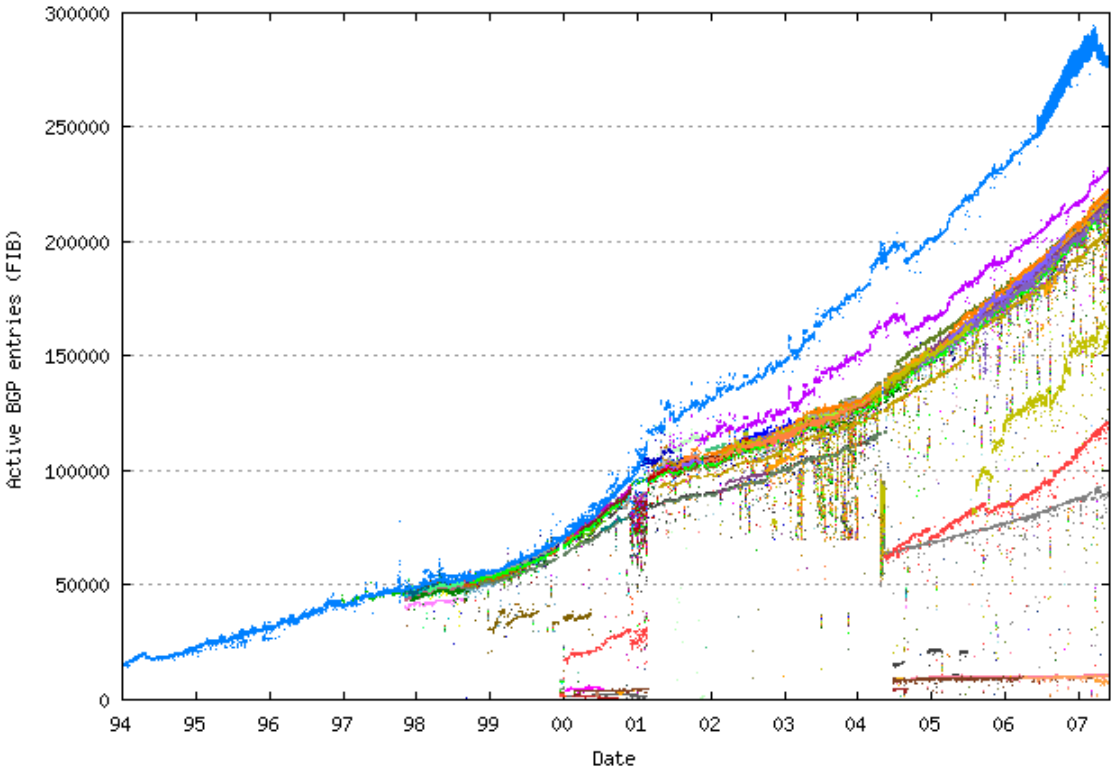

図 2.1: BGP テーブルエントリ数の増加 [1]

まず，経路数の増加に関して説明する．インターネットは，同一ルーティングポリシーの配下にある IP (Internet Protocol) ネットワーク (やルータ) の集合で構成される AS (Autonomous System：自律システム) の集合体である．AS 間のルーティング (経路決定) には，一般に BGP (Border Gateway Protocol) と呼ばれるルーティングプロトコルが利用される．このため，BGP ルーティングで利用される BGP テーブルのエントリ数の増加は，経路数の増加を表現する指標のひとつとなる．ここで，図 2.1に BGP テーブルのエントリ数の増加の様子を示す [1]．このグラフでは，1994 年から 2007 年までの期間における複数の AS の有効な BGP テーブルエントリ数を示している．2001 年頃までは BGP テーブルエントリ数は指数関数的に成長し，インターネット接続性の維持への脅威となっていた．これは，従来の IP アドレス割り当てがクラスフルアドレッシングを利用していたことに一因がある．32bit の IP アドレスはネットワーク部とホスト部に分割されるが，クラスフルアドレッシングでは，IP アドレスのネットワーク部を 8/16/24bit の固定ブロック単位で区切る．このため，固定ブロック長とは異なる bit 数での経路集約ができず，登録すべき BGP テーブルエントリ数も増加しやすかった．そこで IP アドレスのネットワーク部の長さ (プレフィックス長) を自由に設定できる CIDR (Classless Inter-Domain Routing) が広く利用されるようになった．CIDR では複数の経路を適切に集約することができるため，CIDR 利用が一般化した 2001 年以降は BGP テーブルのエントリ数は線形に近い増加にとどまっている．最もエントリ数の多い AS では，2007 年上半期に 27 万エントリに達しており，1998 年と比較すると 10 年間で 5.4 倍程度の増加に相当する．

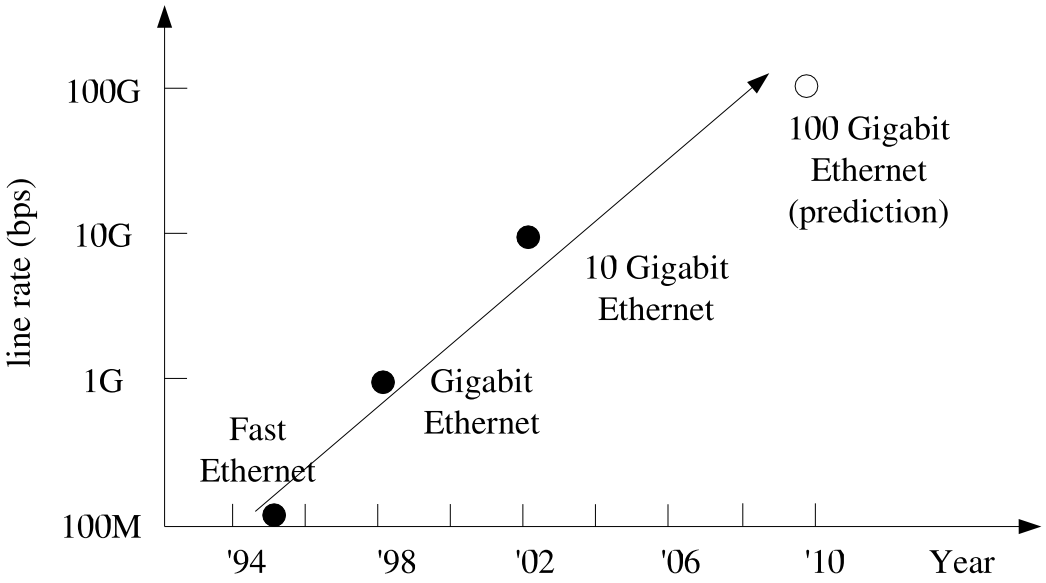

図 2.2: Ethernet の回線速度の向上

次に回線速度の高速化について説明する．1990 年代始め頃は，インターネットを利用する個人や民間企業等のエンドユーザの間では，ダイヤルアップや ISDN による数十 Kbps (bit per second) 程度の低速な回線が利用されていた．このため，ネットワークのトラフィック量も今日と比較すると非常に少なかった．ところが，日本では 2000 年代以降，数十 Mbps 程度の ADSL (Asymmetric Digital Subscriber Line) が安価に提供され始め，2007 年現在では 100Mbps の FTTH (Fiber To The Home) の普及も進行している．このように，末端ユーザの回線速度が 2〜3 桁向上したことにより，ネットワークのトラフィック量が増加している．また，基幹インフラにおいても従来の SONET に

加え，低コストで大容量通信が可能な Ethernet [8]が普及したこともネットワークのトラフィック量の増加に影響を与えてきたと考えられる．特に Ethernet は，扱いやすさや安価なインタフェース，素早い技術革新の導入等の理由により，標準化が順調に進み，図 2.2に示すように，1995 年の Fast Ethernet (100Mbps)，1998 年の Gigabit Ethernet (1Gbps)，2002 年の 10Gb Ethernet (10Gbps) と約 4 年毎に 10 倍ずつ高速化されてきた．10Gbps 強の回線速度は技術的障壁が高く，標準化が遅れていたが，2006 年の夏に HSSG (Higher Speed Study Group) が IEEE802.3 標準化委員会の中に発足し，2009 年中の標準化完了を目指した 100Gb Ethernet (100Gbps) の標準化活動が始まっている．

このように，経路数の増加と回線速度の高速化を考慮すると，最近の 10 年間では，ギルダーの法則の 100 万倍とまでは達しないものの，ビルジョイの法則に匹敵するネットワークのトラフィック量増加があるものと想像できる．

2.2 ネットワーク転送装置：ルータ

こうした大量のネットワークトラフィックを適切な宛先へ転送するために，基幹ネットワークでは，ネットワーク転送装置 (ルータ，レイヤ 3 スイッチ) が利用されている．一般的な IP 転送を行なうルータは，複数のネットワークインタフェースを備えており，受信したパケットのレイヤ 2 ヘッダが自インタフェース宛の宛先 MAC アドレスを持っていればパケットを取り込む．そして，受信したパケットのレイヤ 3 ヘッダに記録されている宛先 IP アドレスを利用して，ルータが備えるルーティングテーブルを参照し，出力インタフェース番号とネクストホップアドレス (目的地ではなく，自ルータに接続されている次の機器のインタフェースに対応する IP アドレス) を獲得する．そして，ネクストホップアドレスに対応する MAC アドレスでパケットの宛先 MAC アドレスを書換える．また，送信元 MAC アドレスを自身の出力インタフェースの MAC アドレスに書換えて，選択した出力インタフェースからパケットを送信する．また，パケットが壊れていないかの検査や，パケットがネットワーク上に永遠に存在し続けないようにパケットのエイジング，必要であればパケットヘッダの修正等も行なう．更に，ハイエンドルータでは，前記のような単純なパケット転送だけでなく，高度な機能として，パケットヘッダの調査結果と設定用のテーブル内容に基づいた優先度制御やフィルタリング等も行なう．

インターネット黎明期のルータでは，高々十数本の 10Mbps 程度の回線を介したパケット交換を行なうだけの能力があれば十分であったため，通常のプロセッサでルータを構成し，ソフトウェア処理によってパケット処理を実現することが可能であった．ところが，回線速度が向上しトラフィック量が増加するにつれ，通常のプロセッサでは処理が困難になり，ルータ専用ハードウェアが必要となった．現在の代表的なハイエンドルータの 1 筐体あたりの交換容量，回線数，出荷年を表 2.1に示す．これらのルータは，筐体あたり 10Gbps Ethernet 回線であれば 32 本程度を収容でき，パケットのワイヤレート処理が可能である．Ethernet の標準化動向を考慮すると，2010 年頃に 100Gbps 回線を 32 本程度収容可能なハイエンドルータの登場が望まれる．

ここで，ルータの内部構成について簡単に説明する．図 2.3にルータの構成概念図を示す．ルータは，コントロールプレーンとデータプレーンのふたつによって構成される．コントロールプレーンは，ルーティングテーブルや ACL[注 1)]等の各種テーブル情報の作成や更新，また，例外処理等，複雑ではあるが回線速度より低速な制御用の処理を行なう．このため，通常のサーバ等に利用され

[注 1)]ACL: Access Control List の略称．ホストやネットワーク層デバイスがネットワークに対して提供しているサービスポート番号やデーモンのリスト．通常，上り回線と下り回線両方に対する制御を行なうためのもので，ファイアウォールに類似した利用方法がある．

表 2.1: 代表的なハイエンドルータ

ベンダ	機種名	交換容量	回線数	出荷年
Juniper [9, 10]	T640	半二重 640 Gbps	10 Gb Ethernet × 32 or OC-192 (10Gbps) × 32 等	2002 年 4 月
Cisco [11, 12]	CRS-1 8 スロットモデル	半二重 640 Gbps	OC-768 (40 Gbps) × 8 or 10 Gb Ethernet × 64 (over subscribe) 等	2004 年 5 月
Cisco [11, 13]	CRS-1 16 スロットモデル	半二重 1280 Gbps	OC-768 (40 Gbps) × 16 or 10 Gb Ethernet × 128 (over subscribe) 等	2004 年 5 月
AlaxalA [14]	AX7800 シリーズ AX7816R	半二重 640 Gbps	10 Gb Ethernet × 32 or OC-192 (10 Gbps) × 32 等	2004 年 10 月

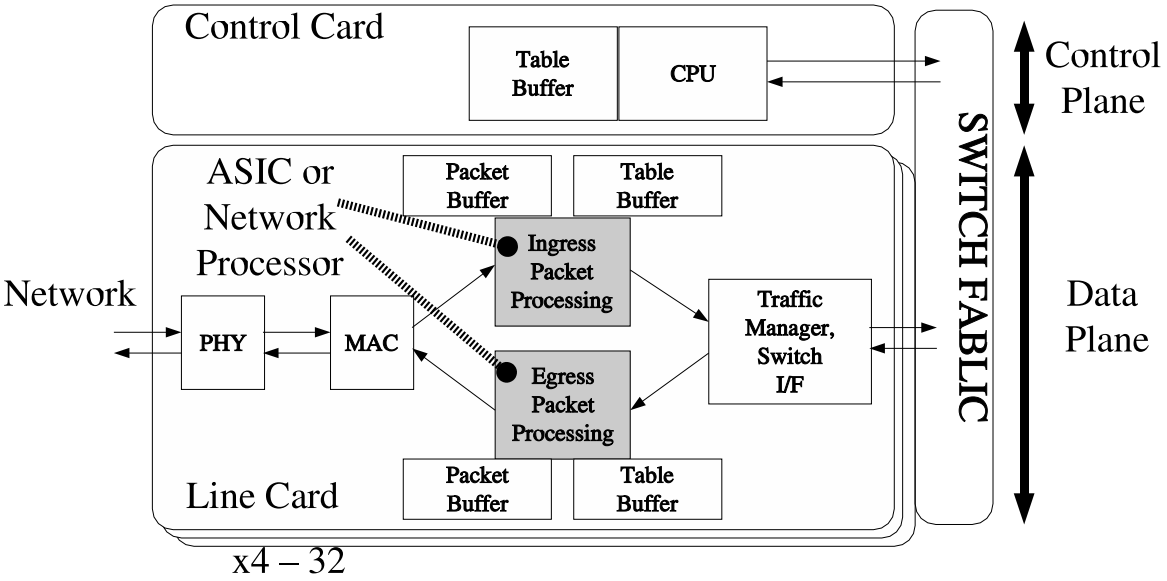

図 2.3: ルータの構成概念図

る CPU (Central Processing Unit)が利用されることが多い．一方で，データプレーンはネットワークインタフェースを備える複数のラインカード間でスイッチファブリックを介してデータ転送 (パケット転送) 処理を行なう．データ転送処理はネットワーク回線からの入力処理を扱う Ingress と，ネットワーク回線への出力処理を扱う Egress に分けて考えることができる．片方向だけの処理を回線の半二重処理，両方向の処理を回線の全二重処理と呼ぶ．データ転送処理では，パケットの解析，経路検索，転送，コピー，ヘッダ修整，廃棄等，比較的単純ではあるが回線速度に達する高速な処理が要求される．パケット転送処理機能のルータ中での実装位置は，ネットワークインタフェースとなる PHY (物理層用コントローラ) と MAC (メディアアクセスコントローラ)，及びルータ内でパケット交換を行なうスイッチファブリックの間であり，図 2.3中の灰色部分に相当する．通常の CPU を利用したソフトウェア処理では近年の 10Gbps のような高速回線に対応できないため，専用開発した ASIC (Application Specific Integrated Circuit)が利用されることが多い．ただし，ASIC は一度 LSI 化してしまうとその後の修正が事実上不可能であるため，通信規格が変

わって処理方法の変更が発生した場合に対応することができない．そこで，高速回線処理に対応しつつプログラマブルにパケット処理を実現したいという要求から 2000 年台初頭にパケット処理に特化したプログラマブルなデバイス，ネットワークプロセッサが登場し，一部のルータでは利用されはじめている．

2.3 ネットワークプロセッサの機能

ネットワークプロセッサは，レイヤ 3 (ネットワーク層) からレイヤ 7(アプリケーション層) のパケット処理を行なう．ここでレイヤとは，国際標準化機構によって制定された OSI (Open Systems Interconnection) 参照モデルの 7 層 (レイヤ) 構造を指す．尚，レイヤ 3 以上の処理以外にレイヤ 2 に相当する MAC アドレスに基づく転送や，レイヤ 2.5 に相当する VLAN (Virtual Local Area Network) や MPLS (Multi Protocol Label Switching) の処理等を扱うこともある．MPLS は，当初は転送の高速化のために利用されていたが，現在では転送経路を決定するための外側ラベルと VPN (Virtual Private Network) 番号の識別に使う内側ラベルをスタックする等の方法で IP-VPN を実現するために利用されることが多い．本節では，ネットワークプロセッサの具体的な処理に関して説明する．各処理を効率良く実現するために，ネットワークプロセッサは通常のプロセッサと異なるハードウェアや命令を追加している．

2.3.1 クラシフィケーション

受信パケットに対して適切な転送処理や優先度処理を実施するためには，受信パケットを解析して適切な情報を抽出し，受信パケットを所定のグループ (フロー) に分類する必要がある．この分類操作をクラシフィケーションと呼ぶ．代表的なクラシフィケーションとしては，レイヤ 3 で行なう IP ルーティングのためのクラシフィケーションや，レイヤ 4 で行なう優先度を考慮した DiffServ(注 2)転送等のためのクラシフィケーション (5-tuple)，単純なパケットフィルタリングのためのクラシフィケーション，また，レイヤ 7 で行なう複雑なパケットフィルタリングのためのクラシフィケーション等がある．ネットワークプロセッサは，クラシフィケーションのために，特定のフィールドを切り出すためのビット列操作やパターンマッチング用のハードウェアと命令を備えることが多い．

2.3.2 テーブルルックアップ

クラシフィケーションの種類によって各種のテーブルメモリを検索し，宛先の決定や，優先度の決定，処理方法の決定等が必要である．このテーブルメモリ検索操作をテーブルルックアップと呼ぶ．尚，用いるクラシフィケーションの種類によって，完全一致 (exact match)，プレフィックスマッチ (prefix match)，レンジマッチ (range match)，ワイルドカード (wild card)，正規表現 (regular expression) 等，様々なルックアップ方法がある．例えば，レイヤ 3 の単純な IP ルーティングの場合，どの宛先へパケットを転送すれば良いか決定する経路検索を行なうために，受信パケットの宛先 IP アドレスをサーチキーとしてルーティングテーブルのルックアップを行なう．一般に，ルー

(注 2)DiffServ (Differentiated Services) は，複数の優先クラス間で相対的な転送性能差をつけることで，トラフィックの優先制御を実現する．IPv4 では TOS (Type of Service) フィールド，IPv6 では Traffic Class フィールドに DSCP (Diffserv Code Point) と呼ぶ優先情報識別子を割り当て，各ルータでは，DSCP を参照して優先制御を行なう．

ティングテーブルは，宛先 IP アドレス，ネットワークマスク，出力ポート識別子とネクストホップアドレス等を各エントリに持つ．出力ポート識別子とは，ルータ装置が複数持つインタフェースのうちどのインタフェースを出力として利用するか区別するためのものである．また，ネクストホップアドレスとは，当該パケットが次に通過すべきルータのインタフェースが持つ IP アドレスである．ネクストホップアドレスは，パケットの持つ宛先 MAC アドレスを次のルータのインタフェースの持つ MAC アドレスに付け変えるために利用する．IP アドレスと MAC アドレスとの対応付けは，ARP (Address Resolution Protocol) を利用することで実現できる．現在の IP ルーティングでは CIDR (Classless Inter-Domain Routing)が利用されているため，最長一致検索 (LPM: Longest Prefix Match)，すなわち，ルーティングテーブル中の宛先 IP アドレスのうち最も長く一致するエントリをルックアップにより検索し，そのエントリの持つ出力ポート識別子とネクストホップアドレスを経路検索結果として出力する．一致するものが無ければ，デフォルトルートの出力ポート識別子とネクストホップアドレスを出力する．

その他のバリエーションとして，レイヤ 3 の IP マルチキャスト通信の場合，宛先 IP アドレスだけでなく，送信元 IP アドレスも同時に参照してルックアップを行なう．また，レイヤ 4 の DiffServ 転送等の場合，5-tuple と呼ぶ 5 種類の情報 (レイヤ 3 ヘッダ中の送信元 IP アドレス，宛先 IP アドレス，プロトコル番号，そしてレイヤ 4 中の送信元ポート番号，宛先ポート番号) を参照してルックアップを行なう．また，レイヤ 4 のパケットフィルタリングの場合，レイヤ 3 ヘッダ中の送信元 IP アドレス，宛先 IP アドレス，プロトコル番号 (TCP/UDP の区別)，そしてレイヤ 4 中の宛先 TCP/UDP ポート番号を参照してルックアップを行なう．

ネットワークプロセッサは，クラシフィケーションの種類に応じたテーブルルックアップを高速に行なうための機能が必要である．例えば，Patricia Trie や Binary Trie，Multibit Trie 等の木状の経路検索アルゴリズムと通常の DRAM や SRAM インタフェースを備えたり，LPM 検索に適した TCAM (Ternary Content Addressable Memory) の制御アルゴリズムと TCAM インタフェースを備えたりする必要がある．TCAM とは，データワードを表現するビット列を 1，0，X (Don’t Care) の 3 値で構成するメモリである．通常の RAM (Random Access Memory) と異なり，あるデータワードを指定するとそのデータワードが見つかった場所のアドレスを返す．データワードを表現するビット列を 1，0 の 2 値で構成する CAM は，TCAM と区別するために BCAM (Binary CAM) と呼ばれることがある．

表 2.2: 回線速度とパケット処理レート

回線種類	回線速度	半二重パケット処理レート
Ethernet (10Base-T)	10 Mbps	15 Kpps
Fast Ethernet (100Base-T)	100 Mbps	150 Kpps
Gigabit Ethernet (GbE)	1 Gbps	1.5 Mpps
10Gigabit Ethernet (10GbE)	10 Gbps	15 Mpps
100Gigabit Ethernet (100GbE)	100 Gbps	150 Mpps
OC-3	156 Mbps	0.39 Mpps
OC-12	625 Mbps	1.56 Mpps
OC-48	2.5 Gbps	6.25 Mpps
OC-192	10 Gbps	25 Mpps
OC-768	40 Gbps	100 Mpps

クラシフィケーション性能やテーブルルックアップ性能は，1秒間に何パケットの処理を行なえるかを意味する pps (packet per second) や sps (search per second) という単位で表現される．また，テーブルルックアップは，ネットワークプロセッサの中でもボトルネックになりやすい処理である．パケットは可変長であるため，最も小さいサイズのパケットが最も高い性能を要求する．回線の種類に応じた半二重パケット処理レートを表 2.2に示す(注 3)．特に，5-tuple のような複雑なクラシフィケーションとルックアップを行なう場合，要求される処理レートを実現するのが非常に困難となりうる．

2.3.3 エラー検出

ノイズ等の影響で，ネットワーク伝送中にデータの一部が誤った値に変わってしまう bit error が発生する可能性がある．このため，送信側でエラー検出用情報をパケットに付与し，受信側でエラー検出を行なうための操作がネットワークの各レイヤにおいて定義されている．CRC (Cyclic Redundancy Check) やチェックサム等がその代表例である．例えば，レイヤ 2 では，Ethernet の場合，FCS (Frame Check Sequence) と呼ぶ 4Byte のフィールドがパケット末尾にある．送信側では，FCS 部分を除くパケット (フレーム) 全体を CRC の生成多項式(注 4) で割り算し，その余りを FCS とする．受信側では，同じアルゴリズムで FCS を計算し，パケットに付与された FCS と不一致であればエラーとして検出する．レイヤ 3 では，IP の場合，IP ヘッダの中に CS (Header Check Sum) と呼ぶ 2Byte のフィールドがある．送信側では，CS フィールドを除く IP ヘッダの各 Byte の 1 の補数を加算し，最終結果を bit 反転して CS とする．受信側では，同じアルゴリズムで CS を計算し，IP ヘッダの CS と不一致であればエラーとして検出する．レイヤ 4 まで扱う場合には TCP や UDP ヘッダの CS に関しても同様である．この他，IPv4 の TTL (Time To Live) や IPv6 の Hop Limit フィールドの値の有効性検査等も行なう．すなわち，TTL や Hop Limit がゼロとなり期限切れ (expire) になれば該当パケットを廃棄し，ICMP (Internet Control Message Protocol) パケットによって送信元に通知する．エラー検出をソフトウェアだけで実現する場合，処理時間が比較的長くなりうる．一方で，専用のエラー検出ハードウェアを用いれば非常に高速かつ容易に機能を実現できる．このため，ネットワークプロセッサでは，専用のエラー検出ハードウェアを組み込み，当該ハードウェアを利用するための命令を追加しておくことが多い．ネットワークプロセッサはこのようなエラー検出の仕組みを搭載し，エラーパケットを廃棄する．

尚，ネットワークプロセッサは個々のパケットのエラー検出・廃棄までは行なうが，エラー訂正まではしないのが一般的である．これは，パケットのエラーを訂正するための冗長な情報が付加されていないこと，仮に付加されていても，エラー検出よりもはるかに大きなオーバーヘッドが発生するので現実的ではないことが理由として挙げられる．一般にエラー訂正は，パケットの最終受信先の上位レイヤのアプリケーションによって行なわれる．

(注 3)Ethernet のパケット処理レートの算出方法は次の通り．Ethernet の最小パケット長は 64Byte である．また，パケットの先頭には，8Byte のプリアンブルが存在する．プリアンブルとパケットを 1 塊としたとき，各塊間には最低 12Byte の IFG (Inter Frame Gap) が存在するため，合計 84Byte (672bit) 単位で最小長のパケットが存在する．GbE の場合，1 秒間に 1Gbit 分のパケット，すなわち，1Gbit/672bit ＝ 1.488 M 個　1.5M 個のパケットが存在する．よって，表 2.2中では，1.5Mpps とした．また，現在，100Gigabit Ethernet は存在しないが，存在した場合の値を記載した．

(注 4)Ethernet 用の CRC 生成多項式は，$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$

2.3.4 カプセリング

あるプロトコルで通信するネットワーク上で，異なるプロトコルのパケットで通信するための仕組みを提供するのがカプセリングである．例えば，IPv4 で通信するネットワーク上で IPv6 の通信をする場合，IPv6 パケットを IPv4 パケットでカプセリングして通信し，再び IPv6 ネットワークに戻った時にデカプセリング (カプセリングを解除) する．別の例として，コア網で利用されることの多い MPLS では，IP パケットを MPLS パケットでカプセリングし，シムヘッダと呼ばれる 4Byte の中に収められる 20bit のラベル識別子を利用して通信を行なう．そして，再び，MPLS ネットワークを出る時にデカプセリングする．このようなカプセリングを実現するためには，パケットの中のしかるべきレイヤ位置に適切なカプセリングヘッダを生成して挿入したり，削除したりする操作が必要である．異なるネットワークの境界におかれるルータのネットワークプロセッサでは，このようなカプセリングの機能が必要である．カプセリングのうち，新規ヘッダの生成は，クラシフィケーションとテーブルルックアップの結果を利用し，演算命令で加工が必要であるため，ネットワークプロセッサの中でもボトルネックになりやすい処理である．一方で，新規ヘッダの挿入や，置換，ヘッダの削除はビット列操作やアライン用のハードウェアと命令を備えることで高速化しやすい．

2.3.5 フラグメンテーション

リンク上で 1 回の転送で送信できるパケットの最大長を MTU (Maximum Transmission Unit) と呼び，次の転送先の MTU が送信元の MTU より小さい場合，送信元では，パケットを指定の MTU の大きさに分割してから送信する．この分割操作をフラグメンテーション，最後に分割後のパケット受け取った宛先のホストが分割時に付与された番号に従って元のパケットに組み立て直すことをリアセンブリと呼ぶ．フラグメンテーションは，パケットのペイロード部分を分割し，分割後のペイロードそれぞれに新規にヘッダを付与するため，必然的にスループットが低下する．また，フラグメンテーションをソフトウェアだけで実現する場合，処理時間が比較的長くなりうる．そこで，ネットワークプロセッサは，高速なフラグメンテーション処理を実現するために，ビット列操作やアライン用のハードウェアと命令を備えることが多い．

2.3.6 フィルタリング

レイヤ 4 以上のパケットフィルタリングを行なう場合，ネットワークプロセッサは，テーブルルックアップの結果，廃棄するように指定されたパケットを廃棄する．パケットの廃棄自体は，ネットワークプロセッサ中で当該パケットが利用している資源から当該パケットを外部へ転送せずに廃棄するだけで実現できる．フィルタリングでボトルネックとなるのは，テーブルルックアップ操作そのものである．

2.3.7 ヘッダ修正

これまでに説明したパケットのエラー検出や，カプセリング，フラグメンテーションでは，パケットのヘッダ修正が必要である．この他，パケットがネットワーク上で永遠に存在し続けないためのパケットエージング (TTL や Hop Limit の減算) にもヘッダ修正が必要である．MPLS 網の中の転送では，シムヘッダ中のラベル修正が必要である．また，DiffServ 等で特定のネットワーク

の中で優先度制御を実現するためには，ヘッダの中の優先度ラベルの修正等が必要である．ネットワークプロセッサでは，上記のようなパケットのヘッダ修正処理のためのハードウェアと命令が必要である．多くのヘッダ修正機能は，これまでの説明で登場したビット列操作とアライン用のハードウェアと命令で実現できる．一部の特殊なヘッダ修正には，通常の演算命令で対応するか，専用のハードウェアと命令の追加で対応する．

2.3.8 トラフィックマネジメント

ネットワークで伝送できるパケット量は，回線速度によって制限される．複数の回線から一つの回線に伝送が集中する (輻輳する) 場合や，高速回線から低速回線へ伝送する場合，入力量が出力量を上回る．前者は，ルータ装置の各ラインカードにネットワークプロセッサを搭載し，複数のラインカードの入力トラフィックが一つのラインカードの出力へ集中する場合に発生する．後者は，あるラインカードのネットワークプロセッサは 1 本の高速回線 (例：10Gbps) を接続し，別のラインカードのネットワークプロセッサは複数の低速ネットワーク回線 (例：1Gbps) を接続し，高速回線側から低速回線側へ転送する場合に発生する．これらの場合，全てのパケットを転送できないため，何らかの廃棄制御を行なう必要がある．このような制御を総称してトラフィックマネジメントと呼ぶ．トラフィックマネジメントにはいくつかの方法がある．

- **キューイング (Queueing)：** パケットに何らかのルールに基づいて優先度を導入する．複数の優先度対応の FIFO キューを用意し，所定のルールに基づいて FIFO キューから取りだす操作を行なうことで，限られた回線容量を有効に利用する．FIFO キューからの主な取りだしルールは，プライオリティキューイング (PQ: Priority Queueing)，重みつきラウンドロビン (WRR: Weighted Round Robin)，重みつきフェアキューイング (WFQ: Weighted Fair Queueing) 等がある．PQ は，優先度の高いキューのパケットを常に優先して取りだす．実装は容易だが，低優先度のパケットが全く出力できない状態 (スタベーション) が発生しうる．WRR では，低優先度パケットのスタベーションを避けるために，各優先度キューのパケットを優先度と平均パケット長に応じて転送する．WFQ では，WRR よりもパケット長をより正確に反映して優先度毎の転送を行なう．

- **パケット廃棄 (Packet Discard)：** ネットワークプロセッサの備えるパケットメモリの容量は有限であるため，何らかの輻輳が発生して，ネットワークプロセッサの外部にパケットを送信できない状態が続くと，やがてパケットメモリからパケットが溢れ，パケットを廃棄しなければならなくなる．パケット廃棄は，なるべく悪影響の少ないやり方で実現することが望ましい．廃棄方法としては，Tail Drop，RED (Random Early Detection, Random Early Discard) 等が利用されることが多い．Tail Drop は，パケットメモリが一杯になった場合に，後続のパケットを廃棄する単純な方法である．RED は，保持しているフロー毎の平均キュー長 (　パケット数) に応じて確率的にパケットを廃棄する方法であり，TCP のパケットフローに対しては，Tail Drop よりも効率良く働く．

- **ポリシング (Policing)：** 設定したトラフィック量や帯域を超過した場合に，超過したパケットにマーク付けを行なったり，超過パケットを廃棄したりする操作である．ポリシングによって，各パケットフローの利用帯域の上限を制限できる．音声や動画等のリアルタイムの通信の制御に適する．その特性上，一般的には入力側で制御を行なう．

- **シェーピング (Shaping)：** 設定したトラフィック量や帯域を超過した場合に，一旦バッファリングし，設定した帯域となるようにスケジューリングし直して送信する操作である．バッファリングによりスケジュールするための遅延が発生するため，遅延要求が低くても一定の転送容量を確保したい通信の制御に適する．その特性上，一般的には出力側で制御を行なう．

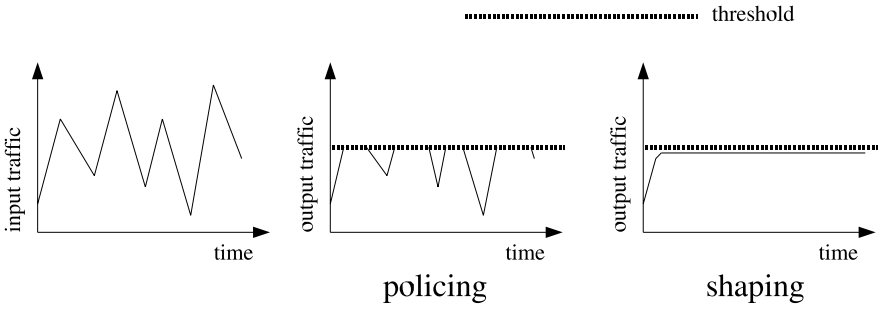

図 2.4: ポリシングとシェーピング

尚，比較的長い時間のトラフィックマネジメントを実現するためには，大量のパケットを保持するための大容量パケットメモリと各フローを保持するための大量の制御用 FIFO を必要とする．例えば，1 秒間という比較的長い時間の中で輻輳制御を実現するには，10Gbps 回線の場合，少なくとも 10Gbit (1.25GByte) のパケットメモリを必要とする．このため，トラフィックマネジメント機能はネットワークプロセッサとは別にトラフィックマネージャと呼ばれる専用の ASIC で実現されることも多い．ネットワークプロセッサにトラフィックマネジメント機能を内蔵する場合，数十マイクロ秒〜数ミリ秒程度の時間の中で輻輳制御を行なうのが一般的である．

2.3.9 ネットワークプロセッサへの期待

以上，ネットワークプロセッサの具体的な処理に関して説明した．これらの処理を効率良く実現するために，ネットワークプロセッサは，通常のプロセッサとは若干異なったハードウェアと命令を備えている．具体的には，パターンマッチングや，経路検索等のテーブルルックアップ，ビット列操作，アライン等に適した専用ハードウェアと，それらの専用ハードウェアを利用するための命令等である．逆に通常のプロセッサが備える，論理アドレスを物理アドレスに変換するためのメモリ管理機構 MMU (Memory Management Unit) や，浮動小数点演算を行なうための FPU (Floating Point Unit) 等はネットワーク処理では不要であるため備えていない．また，ネットワークプロセッサは，通常のプロセッサ同様，プログラマビリティを備えているため，従来の ASIC と比較して，パケット処理の高速化のためのハードウェア構成の変更や，より高度なネットワークサービスの導入に対し，柔軟に素早く対応できる利点を持っている．

ネットワークプロセッサは，産業界が主導となって開発して来たために非公開な情報も多く，学術的な体系化が十分に行なわれていないのが現状である．こうした中，奈良先端科学技術大学院大学の河合らは，ネットワークプロセッサに関して，主に学術機関から発表された多数の論文を調査し，アーキテクチャ，性能評価，プログラミング支援環境，アプリケーションの 4 分野に関してまとめている [15, 16]．河合らは，近年，ネットワークプロセッサが注目されるようになった背景として次のような点を挙げている．

- ルータのパケット転送能力向上や新機能追加を短期間で実現するため．

近年のハイエンドルータでは，パケット処理の高速化を実現するために，図 2.3で示したように ASIC を用いたパケット処理プロセッサを各ラインカードのネットワークインタフェース近傍に配置する構成を採ることが多い [2]．しかし，10Gbps を超えるような高速ネットワークをサポートするためには，高性能 ASIC チップの開発に莫大な費用と時間がかかる．そのため，機能変更のためにソフトウェアによるプログラムが可能で，並列化による性能向上が容易なネットワークプロセッサを用いることで，短期間で安価な製品開発が可能になることが期待されている．

- 高度なネットワーク技術の構築が望まれるようになったため．

  アクティブネットワーク技術 [17, 18] に代表されるように，クライアントやサーバ等のエンドノードだけでなく，ルータのような中継ノードにも一定のサービス処理能力を持たせ，より動的なネットワークサービスを実現するために，ネットワークプロセッサを応用することが期待されている．

- ゲートウェイやサーバホスト等のエンドノードの性能向上のため．

  回線速度の向上に伴い，エンドノードにおけるネットワーク処理の負荷が高まってきており，プロセッサ能力を本来の演算処理に割けなくなり始めている．そこで，ネットワークプロセッサにより，エンドノードのネットワーク処理をオフロードすることが期待されている．例えば，TCP/IP 処理のオフロードが検討されている [19, 20, 21, 22].

ネットワークプロセッサは，この他にも，ネットワークストレージサービスのためのシステムへの応用 [23, 24] やネットワークセキュリティのためのパケットフィルタリング機能として DDoS (Distributed Denial of Service) 攻撃を軽減するためのフィルタリング機能 [25, 26]，アクセス制御リスト (ACL) を用いたパケット分類機能への応用 [27]，レイヤ 7 補助機能として HTTP における URL ベースのスイッチで必要となるパターンマッチング機能や IP ルーティングにおける radix-tree 検索機能，セキュリティ機能で利用される MD5 アルゴリズム等への応用 [28, 29] 等も期待されている．

以上のように，ネットワークプロセッサには様々な利用形態が期待されているが，本論文では，プログラマビリティを備えたままルータのパケット転送能力向上を実現するためのネットワークプロセッサアーキテクチャに焦点を絞って議論する．

2.4 ネットワークプロセッサのバリエーション

ネットワークから到着するパケットは互いに独立に処理が可能であるため，各種の並列処理によってパケット処理のスループットを向上させることができる．このため，多くのネットワークプロセッサはパケット処理に特化した簡易な PE (Processing Element) を多数集積し，PE のパイプライン化，PE のマルチプロセッサ化，PE のマルチスレッド化といった並列処理技術を利用して，パケット処理のスループットを向上させている [2, 3, 4]．2007 年現在のネットワークプロセッサのパケット処理スループットは 1.2Gbps (例:半二重 OC-12 回線 2 本分)〜40Gbps (例：半二重 OC-768 回線 1 本，半二重 10Gbit Ethernet 回線 4 本) の範囲に相当する．尚，1Gbps (Gigabit Ethernet 回線) 以下のパケット処理スループットを持つネットワークプロセッサは，複雑な処理を低速で行なうことを目的としたコミュニケーションプロセッサとして位置付けることができ [4]，本論文の議論の対象からは除くこととする．

ここでは，高いパケット転送能力を持つ現在の代表的なハイエンドネットワークプロセッサとその構成を紹介する．各ネットワークプロセッサが内蔵する PE は，パケット処理エンジン，パケットプロセッサ，または，特殊な名称で呼ばれることが多いが，本論文中では特殊な名称の場合を除いて PE と呼んで説明する．

2.4.1 Silicon Access 社 iPP

iPP は，32 個の PE を集積したネットワークプロセッサであり，Silicon Access 社によって開発された．iPP のブロックダイアグラムを図 2.5に示す．パケット用インタフェースとして SPI-4.2 を 2 基備え，全二重の 10Gbps 回線処理 (半二重 20Gbps 相当) が可能である．TSMC 社の 0.13μm プロセスで製造され，動作周波数 300MHz，消費電力 (typical) 14.5W (max は 18.5W) である．2002 年 6 月にサンプル出荷が始まった [4].

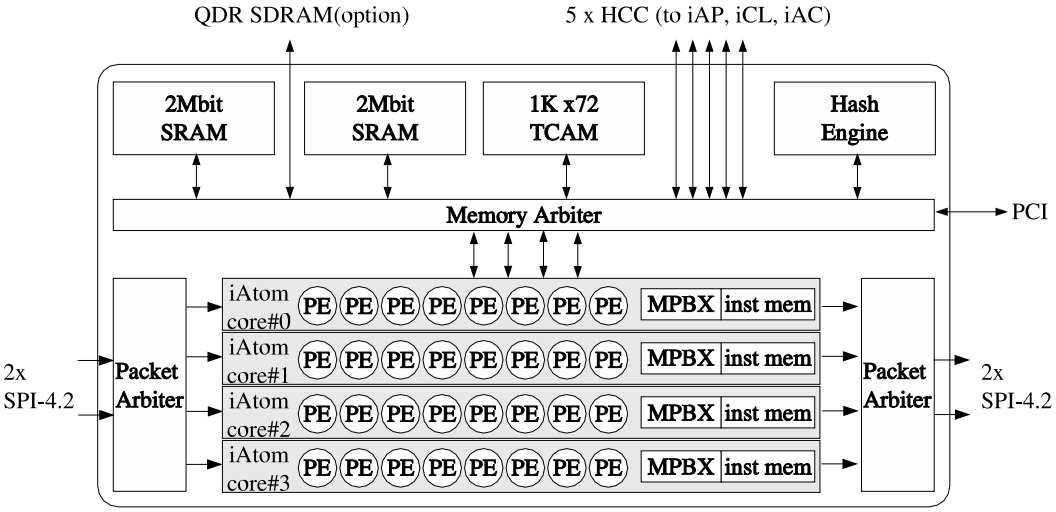

図 2.5: Silicon Access 社 iPP のブロックダイアグラム

iPP の各 PE は，パケット処理に特化した 8way のマルチスレッドプロセッサであるため，32 個の PE により，iPP 全体では 256way のマルチスレッド処理が可能である．iPP は，8 個の PE で iAtom core と呼ぶクラスタを構成し，iAtom core を 4 個集積した構成である．各 iAtom core は，4K 個の命令を保持できる命令メモリを内蔵している．ただし，分岐命令は命令メモリには保持されず，MPBX (Massively parallel branch accelerator) と呼ぶ補助ユニットに格納される．MPBX は，後述の内蔵 TCAM を利用してプログラム中の長い CASE 文の処理を 1 サイクルで実現することで処理の高速化を図っている．また，パケットヘッダの解析や挿入，ハッシングや順番制御を高速化するためのハードウェアエンジンも搭載している．この他，iPP は，2Mbit の SRAM を 2 個集積し，MPLS ラベルテーブルを始めとする各種のテーブルデータを格納するためのスクラッチメモリとして利用する．また，72bit 幅 1K エントリ，または 144bit 幅 512 エントリに構成できる TCAM を内蔵し，MAC/VLAN/IP ルックアップ，ポート再割当等に利用する．Silicon Access 社では，iFlow チップセットの 1 構成要素として iPP を位置付けており，iPP に，検索エンジンである iAP (address processor) と iCL (classifier)，状態制御やポリシングを行なうための iAC (account chip) と 5 系統の HCC (High-speed Coprocessor Channel) を介して接続して全二重 10Gbps 回線の処理に利用する．

iPP では，シングルイメージのプログラミングモデルが利用されている．スレッド切り替えは

ハードウェアで処理され，プログラマは意識する必要が無い．また，パケットは，パケットアービタにより空いている PE に自動的に割り当てられる．

2.4.2 AMCC 社 nP37x0，nP7510

nP37x0 (nP3700，nP3710) は，PE を 3 個集積したネットワークプロセッサであり，AMCC 社によって開発された．同社にとっては第二世代の全二重 OC-48 回線 (半二重 5Gbps 相当) 向けのネットワークプロセッサであり，パケットインタフェースとして，nP3700 は SPI-3/Utopia-3 を 2 基，nP3710 は GMII を 4 基備えている．また，いずれも SPI-4.2 を 1 基備えている．nP37x0 のブロックダイアグラムを図 2.6に示す．0.13μm プロセスで製造され，動作周波数は 700MHz，消費電力 (推測値) は 14W である．2004 年の第一四半期にサンプル出荷が始まった [4].

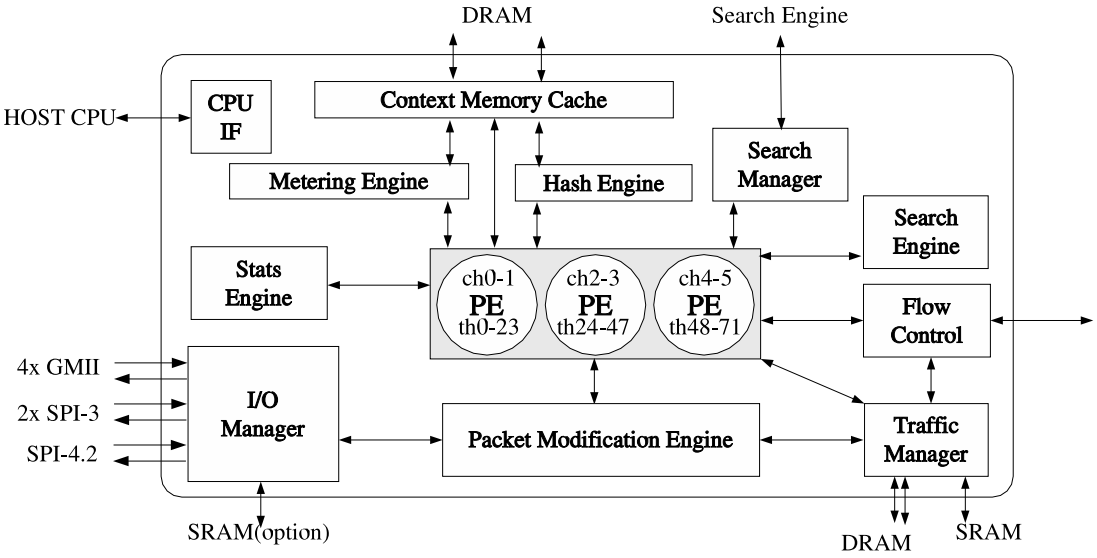

図 2.6: AMCC 社 nP37x0 のブロックダイアグラム

nP37x0 の PE は nPcore と呼ばれる 64bit レジスタを持つパケット処理専用プロセッサであり，32bit RISC 命令に，拡張分岐命令，ビット列操作命令が加えられている．各 PE は独立に動作する．各 PE は 2 本のチャネルと呼ぶ機構を持ち，各チャネルで上り回線用 6 スレッド，下り回線用 6 スレッド，合計 12 スレッドの処理が可能である．あるスレッドの処理がストールすると自動的かつ即座に他のスレッドに処理が切り替えられる．nP37x0 は 3PE を内蔵するため，全体では 72 スレッドの処理が可能である．入力パケットは自動的に空きスレッドに割り当てられ，各スレッドは同一の命令コードを利用してパケット処理を進める．nP37x0 では，シングルイメージのプログラミングモデルが利用され，プログラマからは PE 数やスレッド数が意識される事はない．パケットのクラシフィケーションやスケジューリング等の機能は，内蔵する専用ハードウェアを利用することでワイヤスピードでのパケット処理を実現している．

また，nP37x0 の他，PE を 6 個集積したネットワークプロセッサ nP7510 も開発されている．nP7510 は，パケット用インタフェースとして自社専用の ViX-v3 と呼ぶバスを 1 基備えており，半二重の 10Gbps 回線処理が可能である．全二重 10Gbps 回線の処理のためには，上り回線と下り回線それぞれに nP7510 を 1 個ずつ必要とする．また，標準の SPI インタフェースに接続するためには，インタフェース変換用の FPGA が必要となる．nP7510 は，0.18μm プロセスで製造され，動

作周波数は 350MHz，消費電力 (推測値) は 13W である．2002 年の第一四半期にサンプル出荷が始まった [4].

2.4.3 Agere 社 APP550，APP750

APP550 は，専用 VLIW (Very Long Instruction Word)エンジンを集積したネットワークプロセッサであり，Agere 社によって開発された．APP550 のブロックダイアグラムを図 2.7に示す．パケット用インタフェースとして SPI-3 を 2 基備え，全二重の 2.5Gbps 回線処理 (半二重 5Gbps 相当) が可能である．TSMC 社の 0.13μm プロセスで製造され，コア電圧は 1.2V，動作周波数は 266MHz, 消費電力 (typical) は 11W である．2003 年 4 月にサンプル出荷が始まった [4].

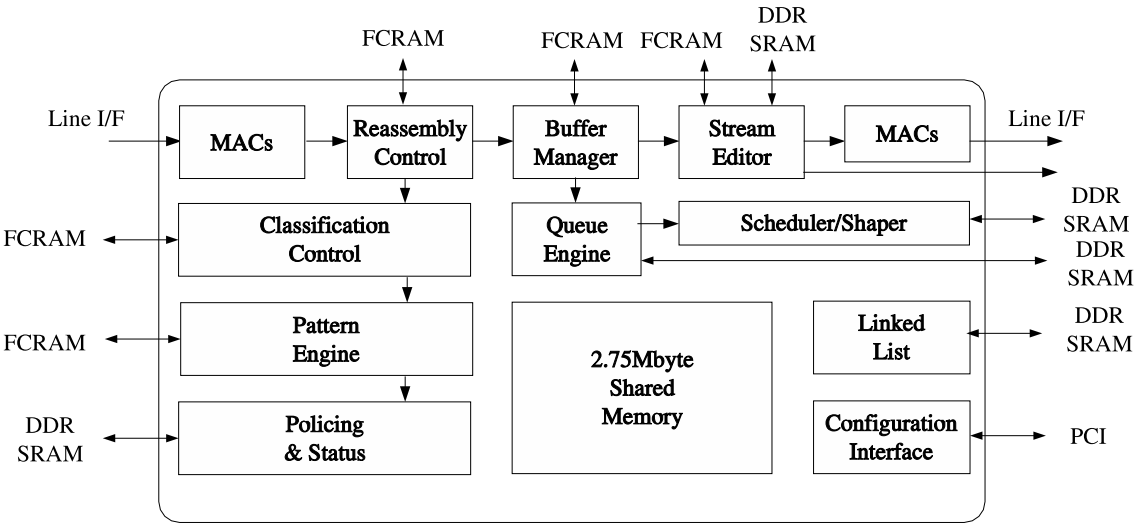

図 2.7: Agere 社 AP550 のブロックダイアグラム

APP550 は，パケット処理を粗粒度のパイプラインステージに分割して行なう．上流のクラシフィケーションステージでは，最大 64 パケットまで同時に解析を行なうことができる．その他の処理ステージは，4 命令同時実行可能な VLIW プロセッサになっており，帯域制御のためのパケットの廃棄処理や，パケットヘッダの修正処理等を行なう．尚，パケットの本体は，外付けの FCRAM に保持され，内部のパイプラインを流れるのはパケット処理用のコンテキストだけである．APP550 は，5 系統の 18bit 幅 DDR SRAM インタフェースと 6 系統の FCRAM インタフェースを備えているが，処理フロー数が少ないシステムに利用する場合は，外付け DDR SRAM を利用せず，2.75MByte の内蔵共有メモリだけで済ますこともできる．

また，APP550 の上位互換ネットワークプロセッサ APP750 も開発されている．APP750 は，7 命令同時実行可能な VLIW プロセッサになっており，パケット用インタフェースとして SPI-4.2 を 1 基備え，半二重の 10Gbps 回線処理が可能である．全二重 10Gbps 回線の処理のためには，上り回線と下り回線それぞれに APP750 を 1 個ずつ必要とする．APP750 は，Agere 社の 0.16μm プロセスで製造され，動作周波数は 266MHz，消費電力 (推測値) は 12W である．2003 年 6 月にサンプル出荷が始まった [4].

2.4.4 Intel 社 IXP2800

IXP2800 は，PE を 16 個集積したネットワークプロセッサであり，Intel 社によって開発された．IXP2800 のブロックダイアグラムを図 2.8に示す．パケット用インタフェースとして SPI-4.2 を 1 基備え，半二重の 10Gbps 回線処理が可能である．このため，全二重 10Gbps 回線の処理のためには，上り回線と下り回線それぞれに IXP2800 を 1 個ずつ必要とする．Intel の 0.13μm プロセスで製造され，動作周波数は 1.4GHz，消費電力 (typical) は 25.5W である．2003 年の第 1 四半期にサンプル出荷が始まった [4, 30, 31, 32].

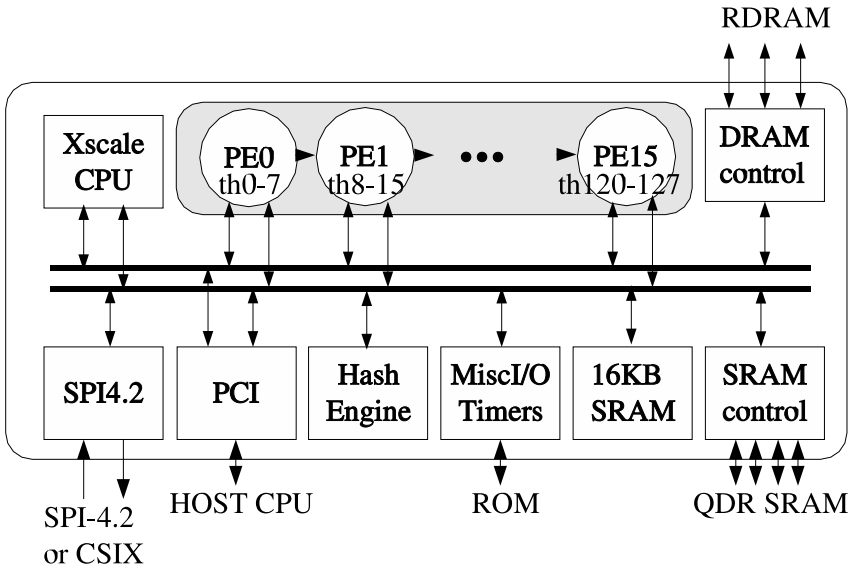

図 2.8: Intel 社 IXP2800 のブロックダイアグラム

IXP2800 の各 PE は，8way のマルチスレッドプロセッサであり，16 個の PE があるため，IXP2800 全体では 128way のマルチスレッド処理が可能である．また，各 PE は，データ移動とビット列操作に最適化した 32bit の RISC 風命令セットを利用し，レジスタファイルと 32bit の ALU，16KByte (4Byte 命令 4K 個) の命令メモリを持つ．スレッド切り替えはゼロサイクルで実行可能である．各スレッドは 32 個の汎用レジスタと 32 個の転送レジスタを持つ．汎用レジスタは，他のスレッドからもアクセス可能である．また，CRC エンジンや PE 間のコヒーレンス維持処理のための分散キャッシュとして動作する小規模 CAM，パケットヘッダのバイト単位のアライメント処理機構等も備える．各パケットの IP アドレスルックアップのために，3 回以上のハッシュ操作が必要である．ハッシュエンジンは 48bit，64bit，128bit オペレーションを利用し，IPv6 を含む様々なルックアップをアシストする．この他，XScale プロセッサ (700MHz，命令キャッシュ32KB，データキャッシュ32KB) を集積し，PCI64 (66MHz) を利用して制御プレーンの処理のためにホストプロセッサとの通信に利用している．

IXP2800 は，非常に柔軟な SMP (symmetric multiprocessor) 構成であり，内蔵 PE 群をパイプライン的に接続して利用することも，並列プロセッサ的に利用することもできる．パイプラインモデルを利用する場合，プログラマは，パケット処理を各パイプラインステージで処理すべき単位に分割しなければならず，この分割が適切でないと処理の重いステージで性能が律速されることになる．並列処理モデルを利用する場合，シングルイメージのプログラミングモデルとして利用できるが，パケット処理命令を各パケットエンジンの命令メモリに納まるサイズ (命令 4K 個以内)

にしなければならないため，複雑な処理ができなくなる．融合型のモデルも利用可能である．例えば，DiffServ を行う場合，まず，8PE をパラレルモデルで利用し，続いて，7PE をパイプラインモデルで利用している．

2.4.5 Xelerated 社 X10q

X10q は，200 個の PE を集積したネットワークプロセッサであり，Xelerated 社によって開発された．X10q のブロックダイアグラムを図 2.9に示す．パケット用インタフェースとして SPI-4.2 を 4 基備え，2 系統の全二重の 10Gbps 回線処理 (半二重 40Gbps 相当) が可能である．TSMC 社の 0.13μm プロセスで製造され，動作周波数 200MHz，コア電圧 1.2V，消費電力 (typical)9.5W (最大 11W) である．2003 年 2 月にサンプル出荷が始まった [4, 33, 34].

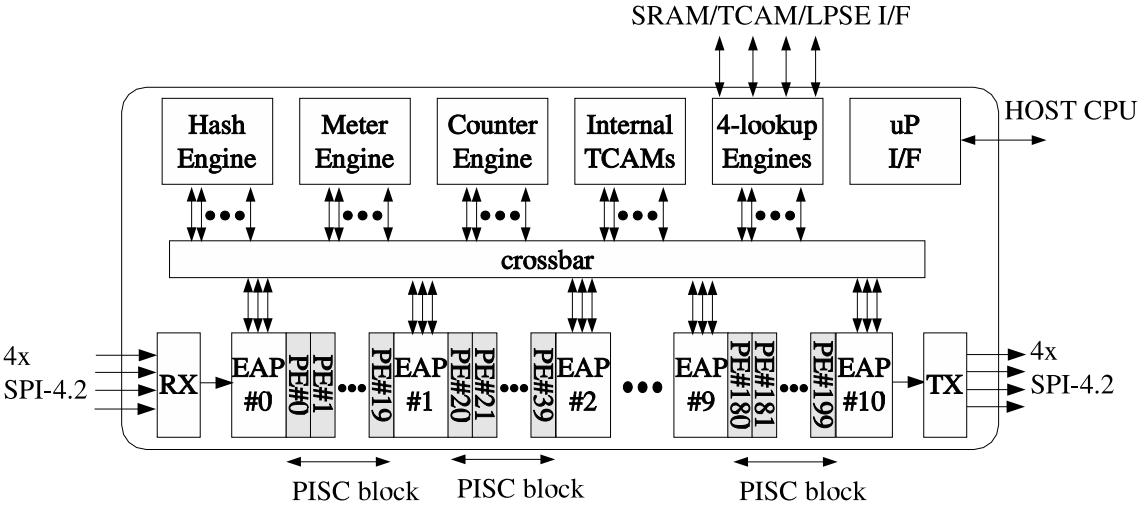

図 2.9: Xelerated 社 X10q のブロックダイアグラム

X10q の PE は，PISC (Packet Instruction Set Computer)プロセッサと呼ばれ，20 個の PE で PISC ブロックを構成している．また，PISC ブロックの前後には，各種の共有資源へアクセスするための EAP (Engine Access Point) と呼ぶ機構を備えている．X10q 全体で 10 個の PISC ブロックと 11 個の EAP を交互にパイプライン状に結合した構成を採っている．X10q の基本ユニットである PISC プロセッサは，ALU 処理，移動処理，分岐処理，オフセット処理の 4 種類の機能ユニット，44Byte のローカルパケットバッファとレジスタファイルを持っている．複数のインストラクションワード (IW) が各 PISC プロセッサに関連付けられ，各 PISC プロセッサは，Row Instruction Pointer (RIP) によって，各パケットで実行する命令を選択する．各 IW は，4 命令の VLIW (Very Long Instruction Word) 命令であり，PISC プロセッサが備える 4 種類の機能ユニットの動作を決定する．各 PISC プロセッサは，パケットあたり 1 回だけ VLIW 命令を実行し，結果と実行コンテキスト (レジスタとパケットバッファの中身) を次の PISC プロセッサ渡すことで，パケット処理を連続的に行っていく．また，X10q は，共有資源としてハッシュエンジン，ポリシング用の 8K 個のメーター，統計収集用の 32K 個の 48bit カウンタ，144bit 幅 512 エントリの 5 個の TCAM，4 個のルックアップエンジンを内蔵している．これらの資源は，EAP からのみ内蔵クロスバを経由してアクセス可能で，パケットあたり最大 800 操作と 28 メモリアクセスが可能である．その間，パケットは EAP 内の FIFO で待機している．典型的な操作内容は，パケットのクラシフィケーショ

ン，修正，カプセリング，ポリシングである．また，EAP は，4 個のルックアップエンジン経由で外部メモリにもアクセス可能である．EAP は検索結果を PISC プロセッサの RIP へ直接転送可能であり，この時 N-way の分岐命令を生成する．

X10q では，シングルイメージのプログラミングモデルが利用される．プログラマは，VLIW 形式の命令を利用し，RISC 風のアセンブラを記述する．1 個の PISC プロセッサには，1 個の VLIW 命令を割当て，処理が不要な場合は NOP (no operation) を挿入する．パイプラインストールは発生せず，全てのパケット処理は同一のサイクル数で出力される．

2.4.6 EZchip 社 NP-1c

NP-1c は，64 個の PE を集積したネットワークプロセッサであり，EZchip 社によって開発された．NP-1c のブロックダイアグラムを図 2.10に示す．パケット用インタフェースとして XGMII を 2 基備え，全二重の 10Gbps 回線処理 (半二重 20Gbps 相当) が可能である．インタフェースは SPI-4.2 と CSIX とすることもできる．IBM の 0.13μm プロセス (Cu11) で製造され，動作周波数は 240MHz，コア電圧 1.5V，最大消費電力 15W である．2003 年 6 月にサンプル出荷が始まった [4, 35, 36, 37].

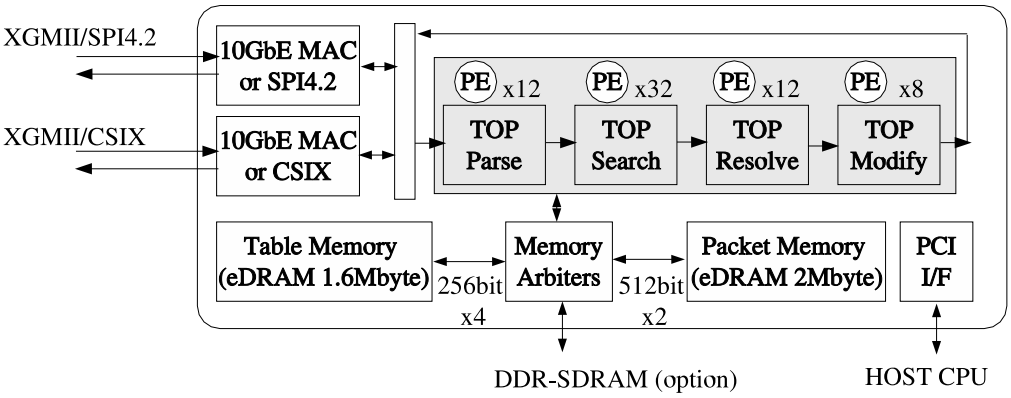

図 2.10: EZchip 社 NP-1c のブロックダイアグラム

NP-1c は，TOPcore (Task Optimized Processors) と呼ぶ技術に基づき，特定の処理に特化した PE を利用してパケット処理を行なう．具体的には，パケット処理を TOPparse，TOPsearch，TOPresolve，TOPmodify の 4 処理 (4 ステージ) に分解し，各処理にそれぞれ 12 個，32 個，12 個，8 個の専用 PE を割り当てている．パケットは後段ステージへとパイプライン的に渡されながら，各 PE で処理されていく．各ステージの PE は，そのステージに適した構成になっている．TOPparse ステージの PE は，パケットヘッダの解析，フィールド抽出，search ステージ用検索キー作成，パケットペイロードからの特定文字列検索等を行なう．TOPsearch ステージの PE は，TOPparse ステージで生成した検索キーを利用して，ルーティングテーブルや，QoS 処理用のクラシフィケーションテーブル，ポリシーテーブルのルックアップ (検索) を行なう．TOPresolve ステージの PE は，TOPsearch ステージのルックアップ結果に基づき，パケットの転送や，QoS 処理の決定，各種テーブルやセッション状態の更新を行なう．最後に TOPmodify ステージの PE は，パケットに対してヘッダの一部を上書きしたり，追加，削除等の更新作業を行なう．

NP-1c は，4MByte の 250MHz 動作の組込み DRAM を内蔵しており，パケットバッファとテーブルメモリに分割して利用している．パケットバッファは，512bit 幅 2 バンク構成で 250Gbps のスループットを持ち，LAN トラフィックをバッファするのに十分な 2MByte の容量を持つ．WAN

トラフィックを扱う場合には，大容量の外部メモリをパケットバッファとして利用可能な QX-1 などの外付けのトラフィックマネージャが必要である．テーブルメモリは，256bit 幅 4 バンク構成であり，こちらも 250Gbps のスループットを持ち，ACL のような小さなテーブルを保持するのに十分な 1.6MByte の容量を持つ．ルーティングテーブルのような大規模なテーブルとして利用する場合，最もよく利用されるエントリだけを保持し，それ以外のエントリは最大 256MByte までの外部メモリ (DDR SDRAM) に保持して利用する．

NP-1c は，シングルイメージのプログラミングモデルを利用する．NP-1c の各ステージに複数ある PE は，ソフトウェアからは，1 個の PE のように見える．各ステージにはスケジューラがあり，実行待ちの PE に自動的にパケットを割り当て同時に複数のパケットを処理させることができる．基本的に，各ステージでは同一の命令を実行するが，各 PE は独立した 8KByte 命令メモリ (4Byte 命令 2K 個格納) を備えているため，各 PE で異なる命令を実行することも可能である．この場合，パケットの種類により特定の PE で実行されることになる．いずれにしても，NP-1c は，4 ステージ処理のため，パケットあたり最大 8K 個の命令を利用できる．各ステージでは，処理後のパケット順番が乱れうるため，出口ポート毎に到着順と同じになるように並べ直す仕組みを組みこんでいる．

2.4.7 Internet Machines 社 NPE10

NPE10 は，64 個の PE を集積したネットワークプロセッサであり，Internet Machines 社によって開発された．NPE10 のブロックダイアグラムを図 2.11に示す．パケット用インタフェースとして SPI-4.2 を 2 基備え，全二重の 10Gbps 回線処理 (半二重 20Gbps 相当) が可能である．TSMC 社の 0.13μm プロセスで製造され，動作周波数 333MHz，コア電圧 1.0V，消費電力 (typical)18W である．2003 年 1 月にサンプル出荷が始まった [4].

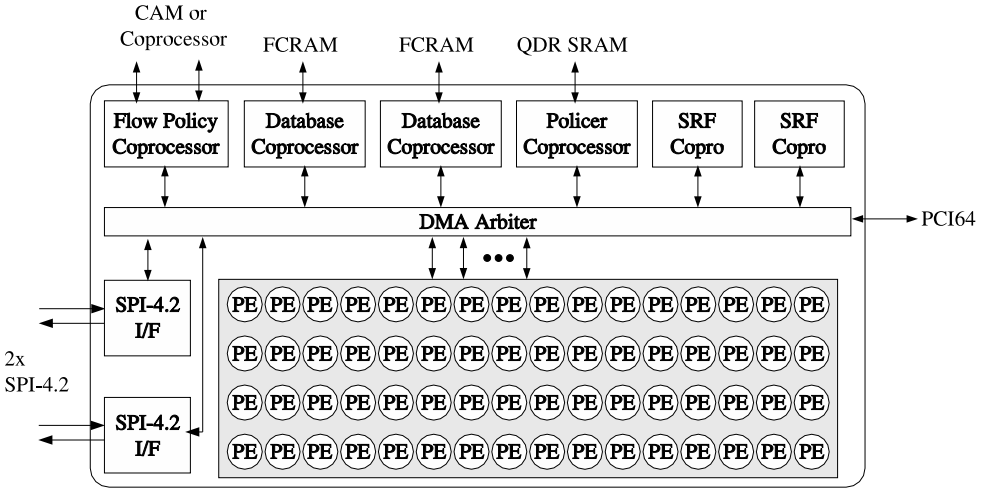

図 2.11: Internet Machines 社 NPE10 のブロックダイアグラム

NPE10 の各 PE は，ARC Cores 社からライセンスを受けたシンプルな 1 命令発行の 32bitRISC プロセッサであるが，パケット処理に特化した拡張を行っている．例えば，各 PE は，通常の 32 本のレジスタに加え，256Byte の拡張レジスタを持ち，パケットヘッダを高速に扱うことができるようになっている．各 PE は，4KByte の命令メモリ (4Byte 命令 1K 個格納) を備えている．ま

た，各 PE の処理を補助するための専用コプロセッサとして FPC (Flow Policy Coprocessor)，2 個の DC (Database Coprocessor)，PC (Policer Coprocessor)，2 個の SRF (Splice/Reorder/Fragmentation Coprocessor) を集積しており，各 PE は，これらの資源に対し，内蔵 DMA アービタ経由でアクセス可能である．

各コプロセッサは入力キューとスケジューラを備え，同時に複数のリクエストを受けつけ，処理し，結果を要求元の PE や他のコプロセッサへ直接転送する．FPC は，2 本の外部インタフェースを備え，最大 2 個のコプロセッサや，TCAM，SRAM 等を接続し，経路検索に利用する．各インタフェースは，最大 144bit までのキー長で 100Msps (search per second) の検索に対応可能である．DC は，外付け FCRAM をデータベースメモリとして利用することで最大 2M エントリ (64MByte) のデータベースを保持できる．PC は，外付け QDR SRAM をコンテキストメモリとして利用することで最大 256K エントリのコンテキストを保持でき，srTCM，trTCM，デュアルリーキーバケツのアルゴリズムに基づくポリシングを実現できる．SRF は，パケットヘッダの追加やパケット分割，パケット順番変更等を行なうことができる．

NPE10 は，シングルイメージのプログラミングモデルを利用している．入力されたパケットは一旦内蔵メモリに保持された後，実行待ちの PE に割り当てられ，処理される．また，各コプロセッサは各 PE の命令に既述しておくことで利用できる．

2.4.8 Cisco 社 Toaster3

Toaster3 は，PE として 16 個の Toaster Micro Controller (TMC) と呼ぶパケット処理エンジンを集積したネットワークプロセッサであり，Cisco 社によって開発された．Toaster3 のブロックダイアグラムを図 2.12に示す．専用の 10Gbps 回線パケット用インタフェースを備え，半二重の 10Gbps 回線処理が可能である．このため，全二重 10Gbps 回線の処理のためには，上り回線と下り回線それぞれに Toaster3 を最低 1 個ずつ必要とする．富士通の 0.18μm プロセスで製造され，動作周波数 240MHz，最大消費電力 14W である．また，LSI ダイサイズは 259mm^2 (一辺長 16.1mm) で 1030 万ゲートを集積している．2001 年 6 月に開発され，Cisco 社のエッジルータ 10000 シリーズの後継機に利用された [4, 11, 38].

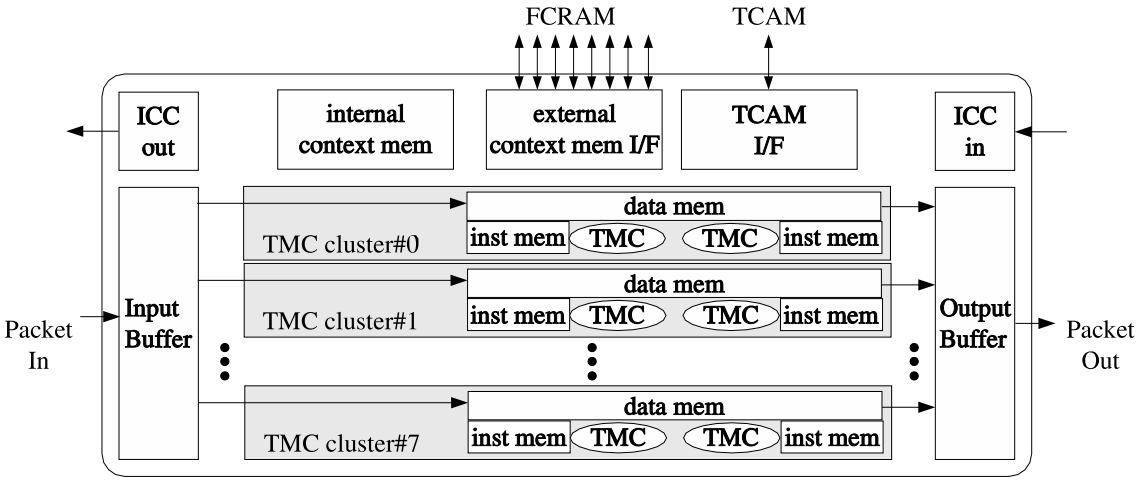

図 2.12: Cisco 社 Toaster3 のブロックダイアグラム

Toaster3 は，2 個の TMC でクラスタを構成し，そのクラスタ 8 個を並列に配置した構成である．

入力パケットは，8 クラスタのいずれかに割当てられる．各クラスタは，データメモリ (8KByte の SRAM) を介してパケットを受信し，2 個の TMC を利用してパケット処理を行なう．各 TMC は，74bit 幅の命令を 2K 個格納できる命令メモリ，2 個の 32bit ALU，16 個の汎用レジスタを持つ，5 段パイプラインプロセッサである．各 TMC は，独立した命令メモリを備えているため，それぞれが独立したパケット処理を実施できる．クラスタ内の 2 個の TMC は，データメモリの他に制御レジスタユニット，メモリ管理ユニットを共有している．また，各クラスタは，クラスタ外部へのインタフェースを備え，TCAM インタフェースと内部コンテキストメモリ，外部コンテキストメモリインタフェース (FCRAM インタフェース)，Toaster3 チップ間通信チャネル (ICC: Inter-chip Communication) ユニット，ホストプロセッサインタフェースへアクセスが可能である．

尚，Toaster3 の前身となる Toaster2 では，Toaster3 と同様 16 個の TMC に相当するプロセッサを内蔵していたが，クラスタを 2 個ではなく 4 個のプロセッサで構成していた．そして，クラスタ内の 4 個のプロセッサが 1 つの命令メモリを共有していたため，1 クラスタ内では同じ命令コードを使わざるを得ず，独立したパケット処理が困難であった．

2.4.9 Cisco 社 SPP(Silicon Packet Processor)

SPP (Silicon Packet Processor) は，192 個の PE を集積したネットワークプロセッサであり，Cisco 社によって開発された．SPP のブロックダイアグラムを図 2.13に示す．専用の半二重 40Gbps 回線パケット用インタフェースを備えており，全二重 40Gbps 回線処理には上り回線と下り回線それぞれに SPP を 1 個ずつ必要とする．IBM の 0.13μm プロセス (Cu11) で製造され，動作周波数 250MHz，消費電力 35W である．また，LSI ダイサイズは 335mm^2 (一辺長 18.3mm) で 3800 万ゲート (1 億 8500 万トランジスタ) を集積している．2002 年 12 月にサンプルが製造され，2004 年 5 月発表の Cisco 社のハイエンドルータ CRS-1 用に利用されている [4, 11, 39].

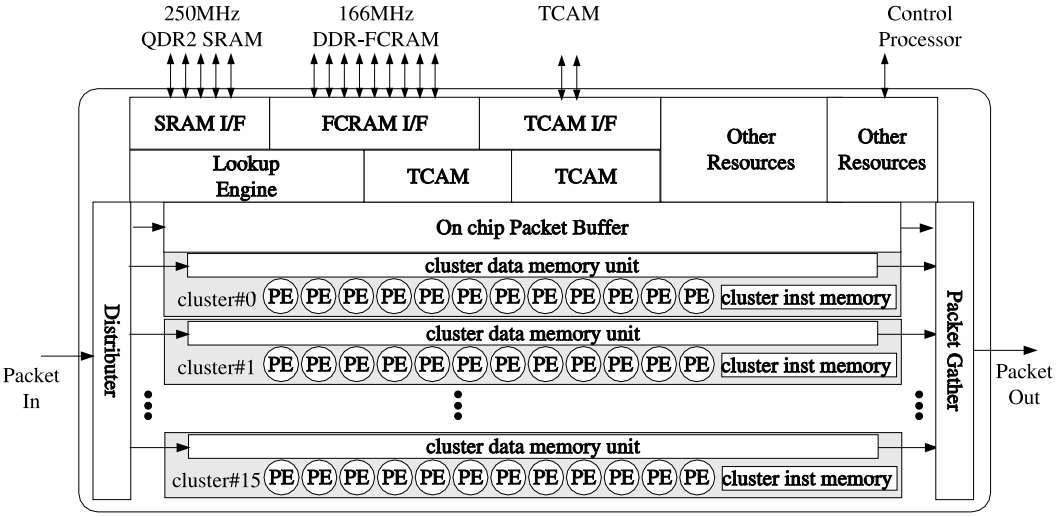

図 2.13: Cisco 社 SPP のブロックダイアグラム

SPP は，12 個の PE でクラスタを構成し，そのクラスタを 16 個並列に配置した構成である．合計 192 個の PE を備えているが，パケット処理に利用されるのは 188 個と発表されている．クラスタ内の PE はカスケード構造を持つクロスバで接続されており，各 PE で 1 パケットの全処理を実施する．各パケットは，パケット先頭 100Byte までがいずれかのクラスタに送信され，クラスタ内

でデータメモリユニットを介して，いずれかの PE に割り当てられる．パケットの残りの部分はオンチップのパケットバッファに保持される．このパケット分散はフローとは無関係に行なわれる．各 PE でパケット処理後，オンチップパケットバッファに保持しておいたパケットの残りの部分と結合し，受信順番と同じ順番に並べ直してから出力する．各 PE は，Tensilica 社の Xtensa 32bit RISC プロセッサを拡張した 5 段パイプラインプロセッサであり，パケット処理用に小容量の命令キャッシュメモリとデータメモリ，DMA (Direct Memory Access) エンジン，nonaligned operation 命令の追加等が行なわれている．

SPP は，各 PE の共有資源として，ACL，Netflow(注 5)，ポリシング等の用途向けに 512K エントリの TCAM，IPv4，IPv6，マルチキャスト，MPLS 用の 2M エントリのルックアップメモリ，64bit の 1M エントリのステータスカウンタ等を内蔵している．また，QDR2 SRAM インタフェース，FCRAM インタフェースを持ち，QDR2 SRAM は，ポリシング状態，クラシフィケーション結果等の保持，FCRAM はルックアップテーブルの保持に利用している．

2.5 従来型ネットワークプロセッサの分類

2.4節で説明した代表的なハイエンドネットワークプロセッサのアーキテクチャは様々であるが，多数の PE による並列処理を利用するものが大多数を占める．大きく分類するとロードバランス処理型とパイプライン処理型，もしくは，両者の融合型に分けて考えることができる．

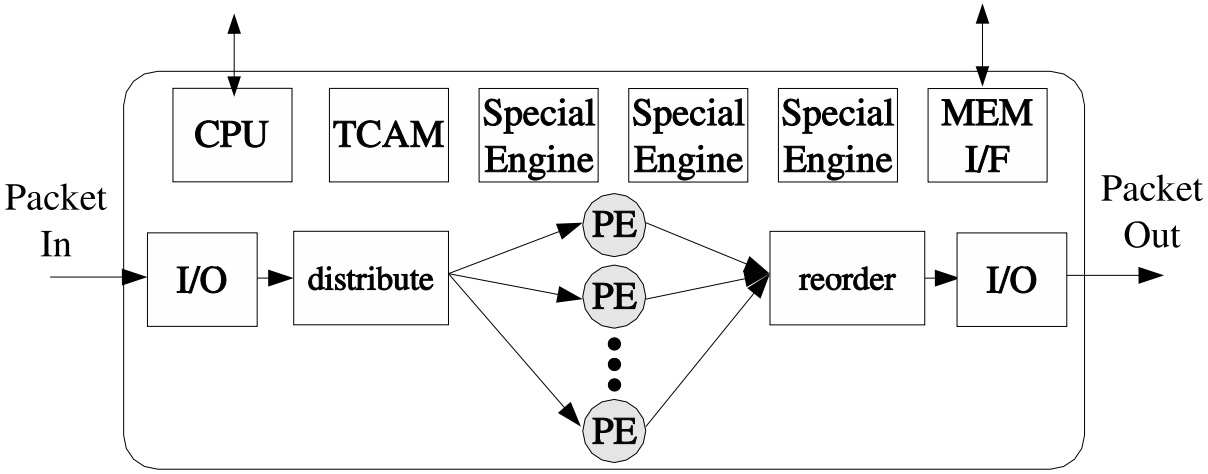

図 2.14: ロードバランス処理型ネットワークプロセッサの概念図

まず，ロードバランス処理型の概念図を図 2.14に示す．ロードバランス処理型は，多数の PE を並列に接続する形に集積し，PE 毎にパケット処理を実施するモデルである．入力パケットを処理待ちの PE，もしくはフロー毎に指定した PE に割り当てて処理を進める．ここで，フローとは指定したヘッダ情報が同じパケット群を意味する．どのレイヤまでパケット処理を行なうかによって指定の種類が異なる．例えば，単純な IP フォワーディングでは，レイヤ 3 の宛先 IP アドレスフィールドが同じであれば同一フローとみなすことができる．回線速度を R としたとき，N 個の

(注 5)Netflow：Cisco 社によって開発された IP トラフィック情報を収集するためのネットワークプロトコル．複数のバージョンが存在し，広く利用されているのは Version 5 であり，送信元 IP アドレス，宛先 IP アドレス，送信元 TCP ポート番号，宛先 TCP ポート番号，プロトコル番号の 5-tuple 情報でパケットをフローとして区別し，情報収集する．

PE を利用することで，1 個の PE あたりに要求される処理速度を R/N にできるという考え方に基づいている．各 PE は，パケット処理を最初から最後まで全て実施するための命令を備えていることが必要である．Cisco 社の SPP や Toaster3，Silicon Access 社の iPP，AMCC 社の nP7510，nP37x0 等が代表例として挙げられる．尚，パケット処理では，各種のテーブルルックアップのためにメモリアクセスが必要である．しかし，一旦メモリアクセスが始まると PE の演算器は事実上停止状態になってしまう．そこでメモリアクセス中も後続のパケット処理に PE の演算器資源等を有効利用するために，PE にマルチスレッド技術を適用するものもある．Silicon Access 社の iPP はその代表例である．

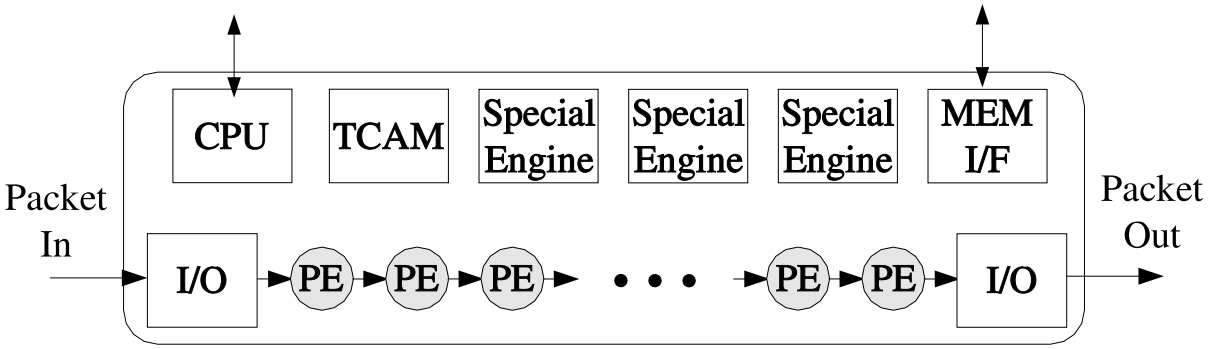

図 2.15: パイプライン処理型ネットワークプロセッサの概念図

次に，パイプライン処理型の概念図を図 2.15に示す．パイプライン処理型は多数の PE を縦列に接続する形で集積する．そしてパケット処理を細分化し，その処理の一部ずつを各 PE に割り当てるモデルである．パケットの各処理が一定時間で終わるように分割できることが理想である．各 PE は，回線速度 R と同じ処理速度で動作することが求められるが，各 PE は，該当処理部分の命令だけを保持すればよい．Xelerated 社の X10q，Agere 社の APP750 や AP550 等が代表例である．

また，ロードバランス処理型とパイプライン処理型の融合型の場合は，パケット処理を大きな複数の処理に分割し，各処理を対応する処理ステージで実行する．各ステージに複数の PE を配置し，ステージ内ではパケットの並列処理，ステージ間ではパイプライン処理を利用する．EZchip 社の NP-1c や Intel 社の IXP2800 が代表例である．

2.6 次世代ネットワークプロセッサへの課題

表 2.3に，2.4節で説明した代表的なハイエンドネットワークプロセッサの諸パラメータ一覧を示す．全二重処理を行なうネットワークプロセッサもあるが，ここでのスループットは，半二重換算値としてある．これらのハイエンドネットワークプロセッサは，32 個程度の PE，0.13μm 程度の CMOS プロセス，300MHz 程度の動作周波数で，半二重 20Gbps 程度のパケット処理スループットを 15W 程度の消費電力で実現していることがわかる．

ここで，IEEE802.3 標準化委員会では 100Gb Ethernet の標準化を 2009 年中に完了させることを目標としているため，2010 年頃には，少なくとも半二重 100Gbps 回線の処理が可能なハイエンドネットワークプロセッサが求められる．従来のネットワークプロセッサの概念の延長線上で考えれば，集積する PE 数を増加して更なる並列化を利用するか，動作周波数を上げることでスループットを向上させることになる．

表 2.3: 代表的なハイエンドネットワークプロセッサ

製品名 (ベンダ)	スループット 動作周波数	プロセス 消費電力	タイプ 内蔵 PE 数，特徴	出荷年
SPP	40Gbps	IBM 0.13μm (Cu11)	ロードバランス型	Dec 2002
(Cisco)	250MHz	35W	188PE	sampled
NPE10	20Gbps	TSMC 0.13μm	ロードバランス型	Jan 2003
(Internet Machines)	333MHz	18W	64PE	sampled
iPP	20Gbps	TSMC 0.13μm	ロードバランス型	Jul 2002
(Sillicon Access)	300MHz	14.5W (max18.5W)	32PE (256 スレッド)	sampled
nP7510	10Gbps	0.18μm	ロードバランス型	1Q 2002
(AMCC)	350MHz	13W (推測値)	6PE (144 スレッド)	sampled
nP37x0	5Gbps	0.13μm	ロードバランス型	1Q 2004
(AMCC)	700MHz	14W (推測値)	3PE (72 スレッド)	sampled
Toaster3	10Gbps	Fujitsu 0.18μm	ロードバランス型	Jun 2001
(Cisco)	240MHz	14W	16PE	sampled
NP-1c	20Gbps	IBM 0.13μm (Cu11)	融合型	Jun 2003
(EZchip)	240MHz	max 15W	64PE, 4 ステージに配置	sampled
IXP2800	10Gbps	Intel 0.13μm	融合型	1Q 2003
(Intel)	1.4GHz	25.5W	16PE (128 スレッド)	sampled
X10q	40Gbps	TSMC 0.13μm	パイプライン型	Feb 2003
(Xelerated)	200MHz	9.5W (max11W)	200PE, 4way VLIW	sampled
APP750	10Gbps	Agere 0.16μm	パイプライン型	Jul 2003
(Agere)	266MHz	12W (推測値)	1PE (7way VLIW)	sampled
APP550	5Gbps	TSMC 0.13μm	パイプライン型	Apr 2003
(Agere)	266MHz	11W	1PE (4way VLIW)	sampled

ここで，2.1式に，消費電力と各種パラメータとの関係を示す．2.1式の A は動作しているトランジスタ数，T は全トランジスタ数，c は静電容量，f は動作周波数，v は電源電圧，l はリーク電流である．消費電力は 2.1式に示すように 1 項目と 2 項目の和となっており，1 項目が動的な消費電力，すなわちトランジスタのスイッチング時にソースからドレインに流れる電流によってもたらされる消費電力，2 項目が静的な消費電力，すなわちトランジスタのゲートリーク電流によってもたらされる消費電力である．

$$power = Acfv^2 + Tlv \tag{2.1}$$

0.13μm プロセスまでは，動的消費電力が静的消費電力より数桁大きく，事実上，2.1式の 2 項目は無視できたため，半導体のスケーリング則の恩恵を得ることができた．半導体のスケーリング則とは，LSI 素子の寸法を $1/K$ に微細化し ($K > 1$)，電界が一定となるようにゲート酸化膜厚や接合深さ，電源電圧等を $1/K$ にスケーリングすれば，消費電力密度が変わらず，集積度が K^2 倍に向上する (すなわち，消費電力は $1/K^2$ となる) という法則である．スケーリング則の下では，同じチップ面積に K^2 倍の素子を集積させても消費電力が変わらない．このため，スケーリング則

が利用できれば，プロセス微細化に伴う集積度向上によって得られたトランジスタで回路を並列化し，スループットを増加させながら，周波数を大きく向上させず電源電圧を小さくすることで，消費電力の急激な増加を防ぐことができる．

しかしながら，0.13μm 未満の半導体プロセスでは，これまで無視できていた 2.1式の 2 項目の静的な消費電力が無視できなくなり，少なくとも消費電力面では，これまでのスケーリング則がなりたたなくなりつつある．例えば，90nm プロセスでは，ゲートリーク電流によって，動的な消費電力の 30% に相当する静的消費電力が生じるとの報告がある [40]．更に 65nm プロセスにおいては静的消費電力が動的消費電力を上回る可能性が高く，従来通りの半導体のスケーリング則を期待することができない．近年では，この問題に対し，SOI (Silicon on Insulator)[5] や，スリープトランジスタ [41, 42, 43]，マルチスレッショルドやダイナミックスレッショルド [44, 45]，歪みシリコン [46] 等の技術を利用して，トランジスタレベルで取り組む有用な研究が盛んである．しかしながら，本研究では，アーキテクチャレベルでのアプローチによって問題を解決することを目指した．すなわち，半導体のスケーリング則に依存せずに，従来の PE 数を増加させる (並列度を向上させる) 以外のアプローチで消費電力を増加させずに高スループット (広帯域) を実現するためのネットワークプロセッサアーキテクチャを確立することを目指した．

そこで注目したいのがネットワーク通信における局所性である．現在，主流となっている IP 通信の場合，データは，64Byte から 1518Byte のパケット複数個に分割されてネットワーク上で送信される．厳密には，IP パケットをペイロードとしたフレームを Ethernet で送信するが，本論文ではフレームとパケットを同義として扱う．パケットは宛先までの経路上に存在するルータ装置を介して転送されるが，各ルータのラインカード上のパケット処理部 (ネットワークプロセッサ) では，同一フローのパケット，すなわち，同一ヘッダとみなせるパケットが短時間に大量に出現しやすい．この現象をネットワーク通信の時間的局所性と呼ぶことにする．ヘッダが同一のパケットは，施すべき処理やその処理結果も同じになる．そこで，通信履歴を採取し，処理結果を後続の同一ヘッダを持つパケットにも適用する仕組みを提供できれば，PE 数を増加させずにパケット処理スループットを増加させることが可能になる．本論文では，ネットワークトラフィックの時間的局所性を利用したネットワークプロセッサアーキテクチャを第 3 章において提案する．

2.7 関連研究

ネットワークの局所性を利用した研究としては，パケット処理のボトルネック処理である経路検索 (ルーティングテーブルのルックアップ) の高速化を行なう手法に関して，古くから研究が行なわれている [47, 48, 49, 50]．本節では，関連研究として，これらの研究の代表例を紹介する．

2.7.1 High-Performance IP Routing Lookup Using CPU Caching

ニューヨーク州立大学の Chiueh らは，ネットワーク上では宛先 IP アドレスが同じパケットが短時間に繰り返し出現しやすいという時間的局所性を指摘した．そして，IP アドレスをプロセッサの仮想アドレスとして扱うことで，通常のプロセッサのキャッシュを利用し，ソフトウェアベースで経路検索を高速化する手法を提案した [51].

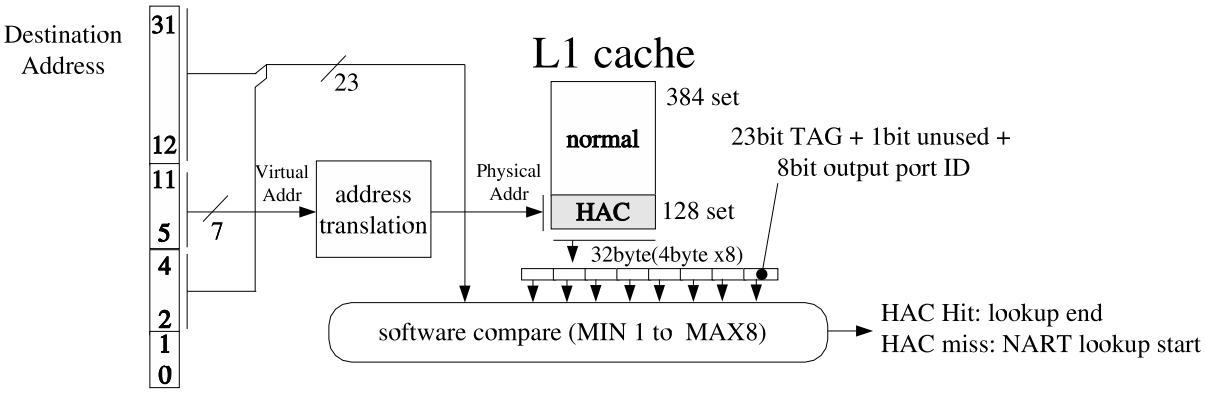

図 2.16: HAC の構造 (32Byte エントリ 512 セット，16Kbyte の L1 キャッシュ利用例)

Chiueh らの提案した経路検索アルゴリズムは，Destination Host Address Cache (HAC) と，Destination Network Address Routing table (NART) によって構成される．図 2.16に HAC の構造を示す(注 6)．32Byte のキャッシュラインを 512 セット持つ 16Kbyte の L1 キャッシュのうち，1/4 にあたる 128 エントリを HAC として利用する．各キャッシュラインには 8 個の HAC エントリ (32bit) が格納され，各エントリは 23bit のタグと 8bit の出力ポート識別子，1bit の未使用領域で構成される．入力パケットの宛先アドレスの 11bit 目から 5bit 目までを仮想アドレスとして扱い，アドレス変換した値を物理アドレスとして L1 キャッシュの HAC 部分の中から 1 セットを選択する．そして，選択されたキャッシュラインに含まれる 8 個の HAC エントリの 23bit を，元の宛先アドレスの 31bit 目から 12bit 目及び 4bit 目から 2bit 目を結合した値と一致するかどうか，ソフトウェアにより順次検査する．一致した時点で HAC ヒットとなり出力ポート識別子が得られる．キャッシュライン中には 8 個のエントリがあるため，一致するエントリ位置によって HAC ヒット時間が異なることに注意されたい．尚，いずれのエントリにも一致しない場合は，NART を利用して経路検索処理を続ける．NART では，3 階層のテーブルが利用され，上位階層から順次一致検査を行なう仕組みになっている．16KByte の L1 キャッシュと 1Mbyte の L2 キャッシュを備える 500MHz の Alpha プロセッサで HAC と NART 構造による経路検索性能を計測したところ，最善のケースで 5.69 サイクル (11.38ns) 間隔，すなわち，87.87Mpps の経路検索性能を得られたとしている．

(注 6)文献により宛先 IPv4 アドレスの MSB と LSB の表示方法が異なっている．本論文では，誤解を避けるため，宛先 IPv4 アドレスの MSB (Most Significant Bit) を 31，LSB (Least Significant Bit) を 0 に統一して図示することとする．

また，HAC の入力部にシフタを組み合わせ，個別の宛先アドレス毎ではなく，ある程度の宛先アドレスレンジをまとめて格納する格納する HARC (Host Address Range Cache)，更にプログラマブルハッシュエンジンも組み合わせた IHARC (Intelligen HARC) を利用することで，更なるヒット率改善が望めるとしている [52].

2.7.2 Routing Lookups in Hardware at Memory Access Speeds

米国スタンフォード大学の Gupta らは，ルータ装置のボトルネックの原因として CIDR による IP ルーティングで利用されている最長一致検索 (LPM: Longest Prefix Match) のメカニズムを指摘し，解決策として，メモリをパイプライン状に配置しメモリアクセス毎に経路検索を可能とするアルゴリズムを提案した [53]. 図 2.17に，Gupta らが提案した経路検索アルゴリズム DIR-24-8-BASIC の構造を示す.

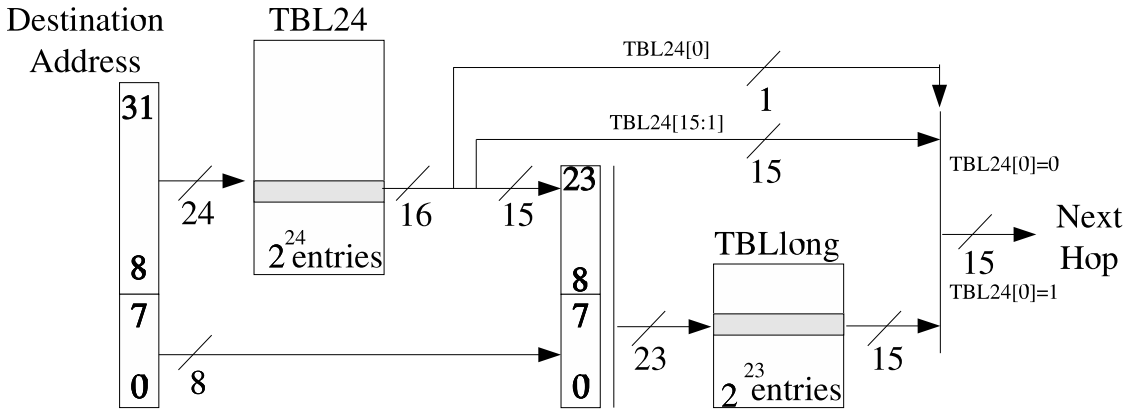

図 2.17: DIR-24-8-BASIC アーキテクチャ

DIR-24-8-BASIC では，TBL24 と呼ぶ第一のテーブルと TBLlong と呼ぶ第二の 2 種類のテーブルを共に DRAM に格納する．TBL24 は，2^{24} エントリを備え，プレフィックス長が 24bit 以下の IPv4 の全経路情報を収める．各エントリには 1bit の識別子と 15bit の結果を収める．結果の 15bit は，識別子が 0 の場合，次ホップ情報を示し，識別子が 1 の場合，TBLlong へのポインタを示す．TBLlong は，プレフィックス長が 24bit より大きい IPv4 の全経路情報を収める．この 2 種類のテーブルを利用して，次のように次ホップの経路を決定する．まず，IPv4 アドレスの上位 24bit を利用して TBL24 をアクセスする．プレフィックス長が 24 以下ならば，TBL24 から読みだした値が次ホップの経路である．プレフィックス長が 25 以上であれば TBL24 から読みだした 15bit の結果と元の IPv4 アドレスの下位 8bit を結合した 23bit で TBLlong をアクセスし次ホップの経路を決定する．ネットワークトラフィックの局所性 (特にプレフィックス長の分布に関する空間的局所性) により，多くの場合，プレフィックス長は 24 以下であることが期待できるので 1 回のメモリ参照で経路を決定することができる．また，DRAM を利用し，メモリアクセス速度を 50ns とすると，プレフィックス長が 24 以下の場合は，20×10^6 回の経路検索 (= 20Mpps: Mega packet per second)が可能であるとしている．

しかしながら，この手法は論文中での指摘もあるように，TBL24 や TBLlong の更新が問題となる．例えば，プレフィックス長が 8bit に対応するエントリは TBL24 上に 2^{16} エントリも存在するため，このエントリ全てを更新することは著しい性能低下を引き起こす.

2.7.3 IP Caching for Terabit Speed Routers

カリフォルニア大学サンディエゴ校の Bryan Talbot らは，ネットワーク上では宛先 IP アドレスが同じパケットが短時間に繰り返し出現しやすいという時間的局所性を指摘し，IP アドレスキャッシュを利用して経路検索を高速化する手法を提案した [54].

IP アドレスキャッシュの特徴は二つあり，一つ目は IPv4 アドレスを二分割してインデックスとタグとして利用することである．二つ目は，キャッシュラインサイズが 1Byte (1 エントリ) ということである．ネットワークを次々と流れるあるパケットのネクストホップ情報に対して，次のパケットのネクストホップ情報が直接関係を持つことは無い，すなわち空間的局所性が存在しないため，キャッシュラインに複数のネクストホップ情報を格納しても意味がないため，キャッシュラインには 1 個の結果だけしか入れる必要がないとしている．通常のプロセッサのキャッシュ同様，インデックスでキャッシュメモリにアクセスし，該当エントリのタグと IPv4 アドレス内のタグを比較して一致すればキャッシュヒットとなる．9 個のサイトから採取した実ネットワークトレースを利用した調査の結果，図 2.18に示すように，IPv4 アドレスの上位をタグ，下位をインデックスとするのが良いことを示した．また，キャッシュエントリ数を 4K〜256K に変化させ，ダイレクトマップ方式，2 ウェイと 4 ウェイのセットアソシアティブ方式のキャッシュ構成を調査したところ，4 ウェイ・セットアソシアティブ方式であれば，4K エントリのキャッシュで 91%以上のヒット率に達していることを示した．

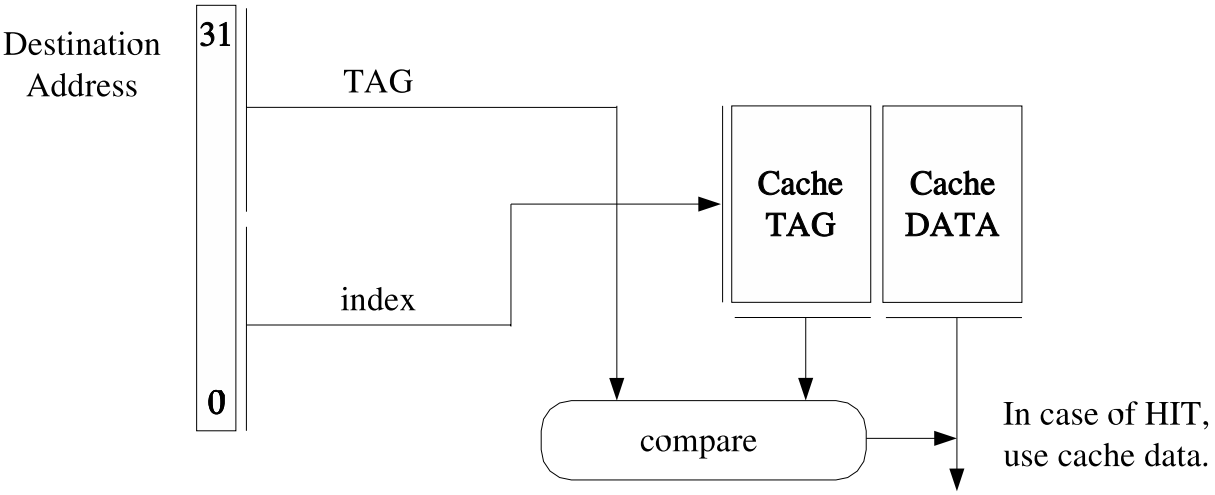

図 2.18: キャッシュのイメージ

DRAM を利用した経路検索では，スタンフォード大学の Gupta らの手法 [53] を用いても DRAM のサイクルタイム (ここでは 55ns) 間隔でしか結果が出ない．しかしながら，4K エントリ，2 ウェイ，2ns サイクルタイム，1ns レイテンシの L1 キャッシュを DRAM と組み合わせて利用すると最悪ケースのサイトでも 11.11ns 間隔に改善できることを示した．また，1M エントリ，4 ウェイ，2ns サイクルタイム，10ns レイテンシの L2 キャッシュも組み合わせると最悪ケースのサイトでも 3.45ns 間隔に改善できることを示した．以上から，ネットワークの時間的局所性を利用した IP キャッシュにより，4ns 以下，すなわち 250Mpps の経路検索性能を得られるため，Ethernet の平均パケット長 (約 500Byte) を仮定すると半二重 1Tbps の処理速度を達成できるとした．

2.7.4 Multi-zone Caches for Accelerating IP Routing Table Lookups

アルバータ大学の Chvets らは，IP トラフィックに存在する局所性について考察を行ない，プレフィックス長によって複数の領域に分割したキャッシュ(マルチゾーンキャッシュ) によってキャッシュヒット率 (＝経路検索性能) を向上させる手法を提案した [55].

Chvets らは，ネットワークの局所性として時間的局所性と空間的局所性を指摘している．時間的局所性とは，短時間に同一の宛先アドレスが繰り返し出現しやすい現象を指し，これは，通信トラフィックが，パケットの連なりとしてネットワークを通過するために存在するとしている．また，空間的局所性とは，同一レンジの宛先アドレス参照が出現しやすい現象を指し，これは，ネットワークアドレス (プレフィックス，アドレスの上位部分) が同じでホストアドレス (アドレスの下位部分) が異なる IP アドレスが同一のサブネットに属しているため，サブネット内のあるアドレスがアクセスされれば，同一サブネット内の別のアドレスもアクセスされやすいために存在しているとしている．そして，2 種類のサイトのトレースを調査し，250ms 以内に約 80%のパケットで以前に出現した宛先アドレスが再参照されるという強い時間的局所性と，約 95%の宛先アドレス参照はプレフィックス長が 11bit から 24bit の間という，アドレス空間全体の 44%に存在しているという強い空間的局所性を確認した．

そして，時間的局所性と空間的局所性の両方を効率的に利用するために，キャッシュエントリの大半をゾーン 1 としてネットワークアドレスのうち頻繁に出現しやすいものに，残りのキャッシュエントリをゾーン 2 としてそれ以外のネットワークアドレス用に割り当てるマルチゾーンキャッシュを提案した．図 2.19にマルチゾーンキャッシュのエントリ割合イメージを示す.

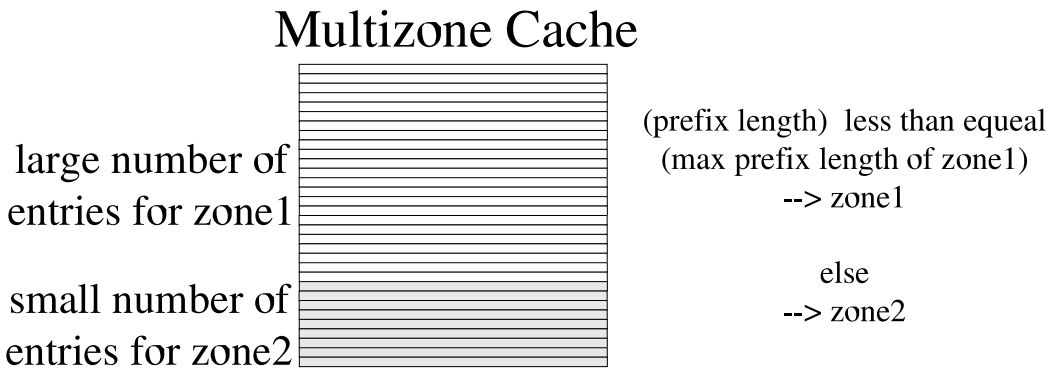

図 2.19: マルチゾーンキャッシュのエントリ割合イメージ

マルチゾーンキャッシュでは，指定したプレフィックス長以下のエントリはゾーン 1 に，それ以外はゾーン 2 に記録する．尚，キャッシュの各エントリは，10bit の出力ポート識別子とプレフィックス (プレフィックス長に依存) を格納するものとしている．マルチゾーンキャッシュの効力を確認するために，プレフィックス長を 18，20，22bit として，2 種類のゾーンを作った場合と，1 種類のゾーンだけの通常のキャッシュとを比較した．いずれのプレフィックス長でもゾーン 1 とゾーン 2 の最適な割り合いは異なるものの，適切なゾーン分割を行なうと通常のキャッシュよりもキャッシュミス率を約半分程度にできるという結果を得ている．また，Kasnavi らはマルチゾーンキャッシュをパイプライン化して高速化する研究を行なっている [56].

2.7.5 Novel IP Address Lookup Algorithm for Inexpensive Hardware Implementation

フィンランドの情報技術研究所の Kari は，SRAM と FPGA を組み合わせたハードウェアにより，スタンフォード大学の Gupta らと同様，プレフィックス長の分布に関する空間的局所性を利用した IP の経路検索を高速化するアルゴリズムを提案した [57]．Kari は，従来のソフトウェアによる手法 [58, 59, 60, 51] では，プロセッサのキャッシュを利用するが，キャッシュミス時のペナルティが大きすぎること，また，Gigabit 回線を備えるルータに対してプロセッサの I/O 性能が不足すること等を指摘した．また，ASIC を開発する手法は開発コストがかかりすぎることを指摘した．そこで，開発コストをかけず高速な経路検索を実現するために，SRAM と FPGA を組み合わせ，それらのデバイスに適したアルゴリズムを提案した．図 2.20に，Kari が提案した経路検索アルゴリズム CS Trie (Compact Stride Multibit Trie) の構造を示す．

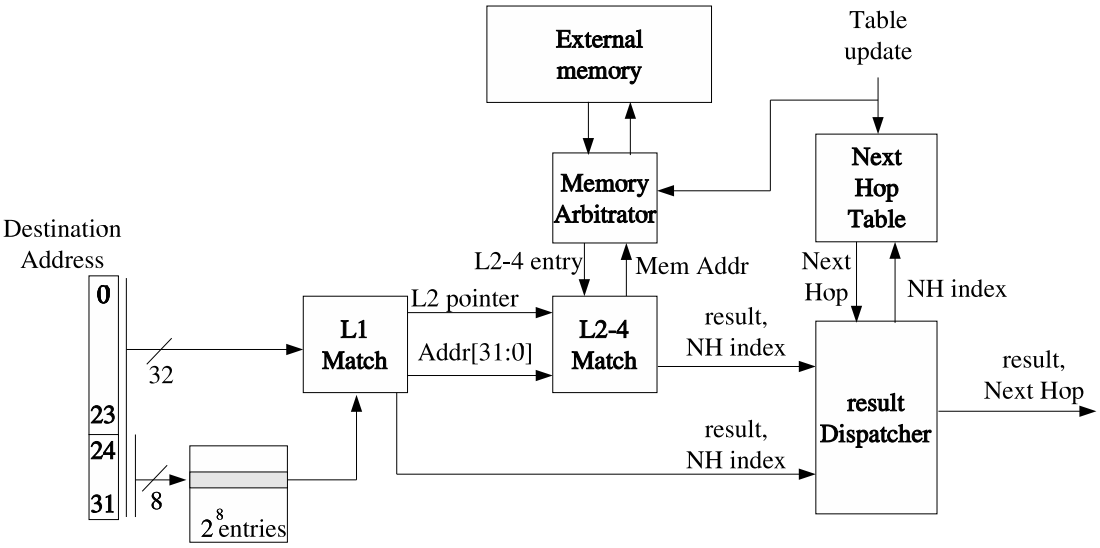

図 2.20: CS Trie based address match system

CS Trie では，経路検索のためのメモリ参照回数の削減，並列化やパイプライン化による高速化，メモリ量を極力削減すること等を実現することを目指している．まず，IPv4 アドレスを 8bit 単位で分割し，4 階層の trie(順序付き木状データ構造の一種) を構成する．この分割したデータ構造を Compact Stride Multibit Trie と呼んでいる．図 2.20に示すように，各階層の処理をパイプライン化し，また，8bit 分割により第一階層となるルートノード情報は，FPGA 上に搭載可能な 256 エントリの小規模なメモリに格納することができるとしている．ルートノードで目的のネクストホップのアドレスが見つかればそこで処理を終了し，見つからなければ後続のパイプラインで外付けの高速メモリ (SRAM) を利用して，第二階層から必要に応じて第四階層までの検索処理を行なう．通常の SRAM と FPGA を利用した CS Trie では，60Mpps から 130Mpps の経路検索性能を達成できるであろうとしている．

2.8 第 2 章のまとめ

本章では，本研究の背景となるネットワークトラフィックの増加，ネットワークトラフィックを扱うルータ装置，ルータ装置で用いられるネットワークプロセッサの処理内容と要求機能，また，従来型のハイエンドネットワークプロセッサの代表例のアーキテクチャを示した．大多数のネッ

トワークプロセッサは，多数の簡易プロセッサ (PE) を集積したオンチップマルチプロセッサ (マルチコアプロセッサ) であり，パケット処理スループットを向上させるために，PE 数を増加させるアプローチを採っていることを示した．このアプローチは，半導体のスケーリング則の恩恵に基づき，半二重 10Gbps〜40Gbps 回線処理までは順調に進んで来た．しかしながら，次世代ではトランジスタのゲートリーク電流により，消費電力面で半導体のスケーリング則が崩れるため，従来の PE 数を増加させるアプローチは，特に消費電力面で不利となる可能性について述べた．

そこで，次世代の半二重 100Gbps 以上の回線処理のために，ネットワーク通信における時間的局所性を利用することに言及した．ネットワーク上には，同一フローのパケット，すなわち，同一ヘッダとみなせるパケットが短時間に大量に出現しやすく，ヘッダが同一のパケットは，施すべき処理やその処理結果も同じになる．そこで，通信履歴を採取し，処理結果を後続の同一ヘッダを持つパケットにも適用するキャッシュの仕組みを提供できれば，PE 数を増加させずにパケット処理スループットを増加させることが可能になる．従来から，ネットワーク通信における局所性に関する研究は行なわれており，これらのうち代表的な研究に関して調査し，これらが経路検索性能を向上させるための手法であることを明らかにした．第 3 章では，従来研究とは異なるキャッシュ手法でネットワークプロセッサのパケット処理スループットを向上させるためのアーキテクチャを提案する．

第3章　Cache-based Network Processor (CBNP) の提案と構成

本章では，通信の時間的局所性を利用した CBNP と呼ぶネットワークプロセッサアーキテクチャを提案する．そして，CBNP におけるトークンと呼ぶ内部情報，CBNP の内部構成，特に時間的局所性を利用するためのメカニズムについて説明し，予備評価結果から CBNP の有効性を示す．

3.1　CBNP のアーキテクチャ

ネットワークプロセッサは，第 2 章で説明した様々なパケット処理を高速化するために専用のハードウェアや命令の追加を行なっている．更に内蔵 PE 数の増加やマルチスレッドの導入等の並列処理技術により，現在のハイエンドネットワークプロセッサは半二重 10Gbps〜40Gbps のパケット処理スループットを達成している．しかしながら，0.13μm 以下の半導体プロセスではリーク電流の影響が無視できなくなり，スケーリング則が崩れ始めているため，今後，並列処理技術の導入のみによってスループットを向上させる手法は，消費電力の大きな増加を招いてしまう可能性がある．そこで，消費電力増加を抑止しつつ，スループットを向上させるアーキテクチャとして，パケット処理の中でボトルネックとなる処理の結果を記録し，再利用する手法が考えられる．通常のパケット転送処理の中でボトルネックになりやすいのは，2.3節で紹介したテーブルルックアップのような複雑なメモリ参照のための処理や，カプセリングのような参照結果を元に新規のヘッダを生成する処理等である．

幸いなことに，2.6節や 2.7節で紹介したようにネットワーク通信には短時間に同一とみなせるヘッダ情報を持つパケット (クラシフィケーションした結果が同一であるパケット) が多数出現するという時間的局所性が存在する．本論文では，この通信の時間的局所性を有効に利用することで，PE 数を増加させずにスループットを増加させるためのネットワークプロセッサアーキテクチャ，Cache-based Network Processor (CBNP) を提案する．CBNP は，PE での処理結果と処理結果適用方法 (Process) を併せて記録する Process Learning Cache (PLC) を備え，後続の同一とみなすパケットには PE を利用せず，PLC の内容を適用することで高スループット化を実現する．また，あるパケットを PE で処理中に同一とみなすパケットが到着する場合，PLC の目的のエントリが更新されるまで該当パケットを保持しつつ後続の別のパケットの処理を進めることで消費電力の高い PE の利用を抑止し，尚且つ，PLC をノンブロッキング化する仕組みを Cache Miss Handler (CMH) として提供する [61, 62, 63, 64, 65, 66, 67].

図 3.1に CBNP のブロックダイアグラムを示す．CBNP は，PLC と CMH を組み込んだパイプライン処理型のネットワークプロセッサである．パイプライン部分を Burst Stream Path (BSP) と呼び，パケットを解析しトークンと呼ぶ内部情報列を生成する A-Engine，PLC を参照してトークンに処理結果と処理結果適用方法を付与 (場合によっては置換) する C-Engine，トークンの内容を適用する R-Engine によって構成する．また，CBNP は，BSP と並列動作する P-Engine と呼ぶ PE 群

を集積した機能ブロックを備える．

CBNPは，回線速度対応の高速処理を実現するために，各部で利用する命令のレベルが異なる．特に高速性を要求されるBSPでは，ビット列操作やフィールド検出，専用ハードウェア処理のON/OFF等の比較的簡易なマイクロコードレベルのプログラマビリティのみをA-EngineとR-Engineに与える．C-Engineはキャッシュ用の機能ブロックであり，受動的に命令を利用する．例えば，P-Engineや外部の制御CPU等によってPLCのエントリを消去する命令を受けつける．P-Engineは，BSPに比べてプログラマブルであり，回線速度達成の妨げ(ボトルネック)となりうる複雑な処理を行なう．例えば，BSPから与えられた情報(トークン)を操作して各種のテーブルメモリを参照するためのアクセスキーを作って，テーブルメモリから必要なデータを検索する．テーブルメモリの種類には，経路検索のためのルーティングテーブルやアクセス制限や優先度制御のためのACL (Access Control List)等がある．また，P-Engineは新規ヘッダ生成等も行ない，PLCに格納する置換用のトークンを生成する．

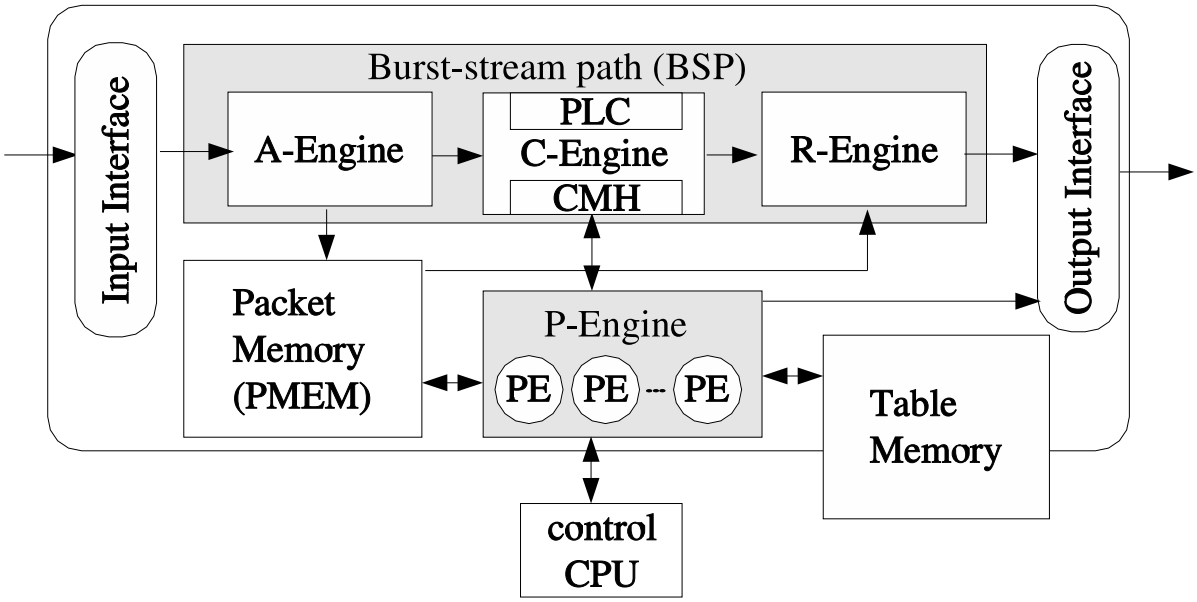

図 3.1: Cache-based Network Processorのブロックダイアグラム

CBNPの最大スループットは，BSPの最大スループットに相当し，これを ***th(BSP)*** と表現する．また，CBNPのキャッシュヒット率を ***h*** と表現する．CBNPで ***th(BSP)*** のスループットを維持するためには，3.1式で示すようにキャッシュミス率分を補う ***th(BSP)×(1-h)*** 以上のスループットのP-Engineを集積すれば良い．

$$\textbf{\textit{P-Engine}に必要なスループット} \geqq \textit{th(BSP)}\times\textit{(1-h)} \tag{3.1}$$

尚，CBNPはPLCとCMHを備えており，CBNPのキャッシュヒット率は，PLCのヒット率とCMH (Cache Miss Handler)のCMT (Cache Miss Table)のヒット率の和を意味する．PLCとCMHの詳細に関しては，3.3.2節にて説明する．

3.2 CBNPが利用する内部情報：トークン

CBNP では，受信パケットからトークンと呼ぶ内部情報を生成し，パケット処理に利用する．トークンとして，U-Info，A-Info，E-Info，R-Info，P-Info の 5 種類の情報を定義する．ここでは各情報の詳細について説明する．表 3.1には，トークンの各情報と CBNP を構成する 4 種類の Engine (A-Engine，C-Engine，P-Engine，R-Engine) との関係 (生成，記録，利用) を示す．

表 3.1: トークンと各 Engine の関係

名称	Engine 部位				意味
	A	C	P	R	
U-Info	生成	記録 (/利用)	(利用)	利用	パケット毎に与える固有の情報
A-Info	生成	記録/利用	利用	利用	パケットの解析情報
E-Info	生成	記録/利用	利用	–	パケットから抽出した情報
R-Info	–	記録	生成	利用	パケットヘッダの置換情報や追加情報
P-Info	–	記録	生成	利用	処理結果の一部，及びその処理結果適用方法

3.2.1 U-Info：固有情報

U-Info (Unique Information) とは，パケット毎に固有の情報であり，20bit〜30bit 程度を想定している．U-Info は，CBNP 内におけるパケットの整理番号や，パケットを保持するためのパケットバッファアドレス，エラー情報や例外情報を記録する．

U-Info は A-Engine で生成し，C-Engine で PLC ミスした際には CMH に記録する．また，R-Engine で元のパケットを参照するために利用する．BSP で想定外のパケットの場合には内蔵 PE が外部の制御 CPU で直接パケット処理をするために P-Engine でも利用する．場合によっては，C-Engine で PLC のエントリをクリアするために利用することもある．

3.2.2 A-Info：解析情報

A-Info (Analyzed Information) とは，パケットの解析情報であり，10bit〜30bit 程度を想定している．A-Info は，ネットワーク通信の各レイヤにおけるプロトコルを必要な分だけ解析した情報である．CBNP の振舞を決定するコントロールレジスタと A-Engine に格納するマイクロコード (命令) により，どのレイヤのどのようなプロトコルまで解析するかを変化させることができる．

例えば，単純なパケットフォワーディング (パケット転送) を設定している場合は，レイヤ 2 のヘッダ情報を調べ，レイヤ 3 のプロトコルを解析した情報を A-Info とすればよい．このとき，様々なレイヤ 3 プロトコルに対処するならば，レイヤ 2 のプロトコルフィールドをそのまま A-Info として取りだせばよいし，現在のインターネット通信の主流のプロトコル (IPv4，または IPv6．この他，VLAN やコア網転送で利用される MPLS 等も主要対象．) のいずれかをターゲットとするならば，対象とするプロトコルであるか否かという情報を A-Info とすればよい．優先度を考慮したパケットフォワーディングを設定している場合は，レイヤ 3 だけでなくレイヤ 4 のプロトコルも合わせて解析し，A-Info とすればよい．

A-Info は A-Engine で生成し，C-Engine の PLC タグの一部として記録する．そして，後続パケットのトークンが PLC に存在するか否かのヒット判定に利用する．また，P-Engine でのテーブル

ルックアップやヘッダ生成処理等や，R-Engine で適用する処理を判定するためにも利用する．

3.2.3 E-Info：抽出情報

E-Info (Extracted Information)とは，パケットのクラシフィケーションを行なうことで得られる抽出情報であり，300bit〜400bit 程度を想定している．E-Info は，A-Info に従ってパケットヘッダの必要なフィールドを抽出することで生成する．いくつかの代表例を表 3.2に示す．単純な IP フォワーディングでは，IPv4 ヘッダの宛先 IP アドレス 32bit だけを抽出すればよい．優先度を考慮したフォワーディングを行なう場合，いくつかの手段が考えられるが，例えば，送信元アドレス及び宛先アドレスの組み合わせでパケットフローを分類して優先度を考慮する方法が挙げられる．IP フォワーディングの場合，IPv4 ヘッダの宛先 IP アドレス 32bit と送信元 IP アドレス 32bit の合計 64bit を抽出すればよい．また，5-tuple と呼ばれる 5 種類の情報を抽出してパケットフローを分類して優先度を考慮する方法もある．この 5 種類の情報とは，一般的には，IPv4 ヘッダの宛先 IP アドレスと送信元 IP アドレス，上位レイヤ (この場合レイヤ 4) のプロトコル(注 1)，上位レイヤの宛先ポート番号と送信元ポート番号である．尚，いずれの場合も，IPv6 の場合は IP アドレス部分は 32bit ではなく 128bit となる．この他，MPLS のシムヘッダの情報や，VLAN のタグヘッダの情報を抽出するのにも利用する．

多数のユーザを収容するインターネットで優先度を考慮したパケットフォワーディングを行なうことを想定すると，5-tuple のような参照箇所の多い設定ができることが望ましい．また，設定対象外のパケット場合は，E-Info として必要な部位だけを抽出するのではなく，パケットのヘッダの先頭部分を E-Info が許容する長さの範囲内でそのまま全て抽出することもできる．

E-Info は A-Engine で生成し，C-Engine の PLC タグの一部として記録する．そして，後続パケットのトークンが PLC に存在するか否かのヒット判定に利用する．また，P-Engine でのテーブルルックアップやヘッダ生成処理等にも利用する．

表 3.2: 代表的な E-Info 抽出内容

設定内容	抽出フィールド	必要 bit 数
IP フォワーディング	IPv4A (DST)	32
IPv6 フォワーディング	IPv6A (DST)	128
優先度考慮 IP 転送	IPv4A (SRC, DST)	32
優先度考慮 IPv6 転送	IPv6A (SRC, DST)	256
5-tuple(IPv4)	IPv4A (SRC, DST), Protocol，Port (SRC, DST)	104
5-tuple(IPv6)	IPv6A (SRC, DST), Protocol，Port (SRC, DST)	296

IPv4A: IPv4 アドレス
IPv6A: IPv6 アドレス
DST: Destination (宛先)
SRC: Source (送信元)

(注 1)プロトコルは，A-Info の一部で代用することも可能である．

3.2.4 R-Info：置換情報

R-Info (Replaced Information) とは，パケットヘッダの置換情報やカプセリング化用の追加情報であり，PLC が記録する処理結果の一部である．300bit〜400bit 程度を想定している．R-Info には，例えば，通常の IP フォワーディングの場合，置換用の宛先/送信元 MAC (Media Access Control) アドレスを入れておく．また，利用プロトコルが異なるネットワーク境界でのフォワーディングの場合，IPv4 から IPv6 ヘッダの置換 (またはその逆) やカプセル化，VLAN や MPLS のカプセリング化や，MPLS のシムヘッダ情報の置換等に用いる．

R-Info は P-Engine で生成し，C-Engine の PLC データの一部として記録する．そして，R-Engine でパケットを修正するためのデータとして利用する．

3.2.5 P-Info：処理情報

P-Info (Processing Information) とは，パケットの処理情報であり，PLC が記録する処理結果の一部と処理の適用方法である．20bit〜100bit 程度を想定している．P-Info は，パケットの処理結果として，ルーティングテーブルや ACL 等の各種のテーブルルックアップ結果を含む．また，パケットの処理結果適用方法として，R-Engine で利用するパケット修正や廃棄処理命令等の指示子を含む．例えば，ヘッダに対するビット列操作 (操作種類 (追加または削除，置換)，操作位置，操作長) の方法，CRC 演算やチェックサム演算の有無，パケット廃棄の有無，優先度処理の指定等である．ビット列操作の際に利用する追加ヘッダや置換ヘッダは，R-Info の中に収められている．

P-Info は P-Engine で生成し，C-Engine の PLC データの一部として記録する．そして，R-Engine でパケットを修正するための命令やデータの一部として利用する．

3.3 CBNP 各部の構成

CBNP の主要構成要素である BSP は，A-Engine，C-Engine，R-Engine と呼ぶ三種類の機能ブロックから構成する．まず，これらの機能ブロックに関してトークンとの関係を交えて説明を行ない，ついで，複雑なプログラム処理を行なう P-Engine について説明する．

3.3.1 A-Engine の構成

A-Engine は，BSP の最上流部に位置し，受信パケットを解析して U-Info，A-Info，E-Info を含むトークンを生成するための機能ブロックである．第 2 章で紹介したネットワークプロセッサの機能に当てはめると，パケットのクラシフィケーションとエラー検出を行なう部位に相当する．図 3.2に A-Engine のブロックダイアグラムを示す．

A-Engine は，パケットを受信するとパケットの先頭部分を BSP 処理用にコピーし，パケット全体をパケットメモリへ格納する．パケットの先頭部分は最大 192byte 程度を想定しているが，これは，一般にネットワークプロセッサで解析を行なうレイヤ 2 からレイヤ 7 の一般的なヘッダ情報を含むのに十分な大きさである．レイヤ 2 からレイヤ 7 までは全部で 6 レイヤあるが，実際には，レイヤ 2，レイヤ 2.5，レイヤ 3，レイヤ 4 の 4 レイヤと，レイヤ 5 以上はレイヤ 7 までをひとまとめにした 1 レイヤの合計 5 レイヤに分類して扱うことが多い．現在のインターネットでは，レイヤ 7 を除けば，各レイヤで利用されるプロトコルはある程度の数に限定される．そこで，主

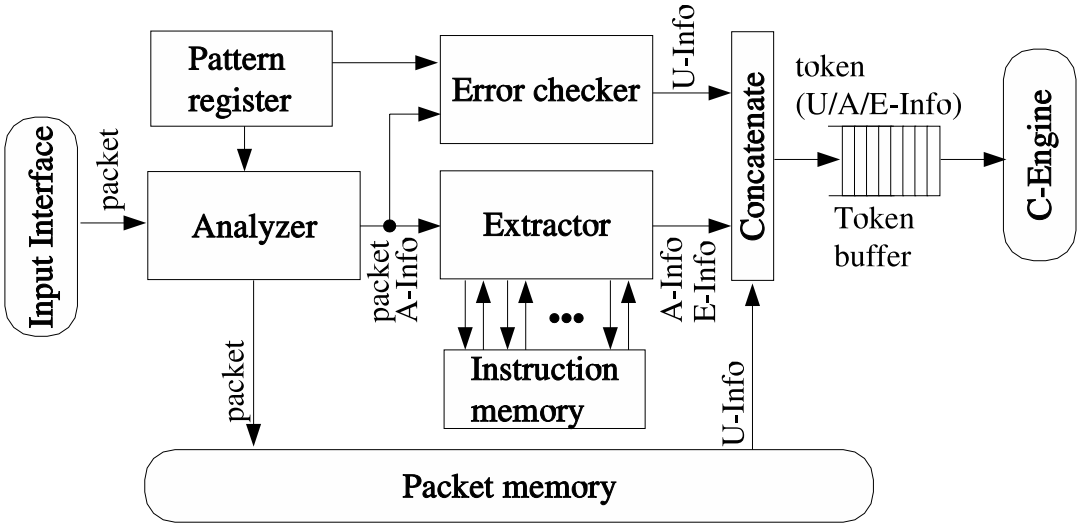

図 3.2: A-Engine のブロックダイアグラム

要なプロトコルのタイプフィールドの価やその位置を予め A-Engine 内のパターンレジスタに用意しておくことが有効である．また，パケットの先頭部分から指定した適切な部位を抽出するための命令列を用意しておく．

A-Eingne は，まず，解析専用パイプライン (Analyzer) を利用し，入力パケットに対してレイヤ 2 から最大でレイヤ 7 までの解析を行なう．いずれのレイヤまでの解析するかはパターンレジスタの設定による．また，Analyzer の各ステージでは，同じくパターンレジスタの設定に従って入力パケットが解析対象のプロトコルか否かを判定し，その判定結果を A-Info として生成する．

次に，抽出専用パイプライン (Extractor) において，A-Info と Extractor が備える命令メモリの命令を利用し，パケットの先頭部分から適切な部位の抽出を行なう．Extractor は 8 ステージのパイプライン構成で，パケットの先頭部分を 4 分割して入力する．そして，各ステージは 4 分割した各ラインから，1 フィールドずつ抽出を行なう．具体的には，各ステージには 1byte 粒度で動作可能なバレルシフタとセレクタが配置してあり，命令は各ラインを何 byte ずらしてどのフィールドを選択するかという処理に利用する．尚，どのレイヤまでを抽出するかによって用意する命令が異なる．各ラインが 8 ステージを通過するので，1 パケットあたり，最大 32 フィールドの抽出が可能である．抽出した結果を E-Info とする．

パケットの先頭部分は，Extractor 通過と同時に Error Checker にも渡される．Error Checker は，Analyzer の解析で未対応のプロトコルであった場合，エラーとして U-Info の一部として登録する．また，IPv4 の Header Check Sum や TTL の検査等も行なう．尚，パケット本体はオンチップのパケットメモリに記録し，パケット記録アドレスやパケット長を U-Info の一部として生成する．

最後に，U-Info，A-Info，E-Info を結合し，C-Engine アクセス用のトークンを生成し，トークンバッファに積んで A-Engine での処理を終了する．

3.3.2 C-Engine の構成

C-Engine は，BSP の中央部に位置し，A-Engine から受信したトークンを処理結果や処理方法を含むトークンに置き換えて R-Engine へ渡す処理を担う，CBNP の中心的役割を果たす機能ブロッ

クである．第 2 章で紹介したネットワークプロセッサの機能に当てはめると，パケットのボトルネック処理 (テーブルルックアップやカプセリングヘッダ生成等) の結果を即座に取り出す部位に相当する．図 3.3に C-Engine のブロックダイアグラムを示す．

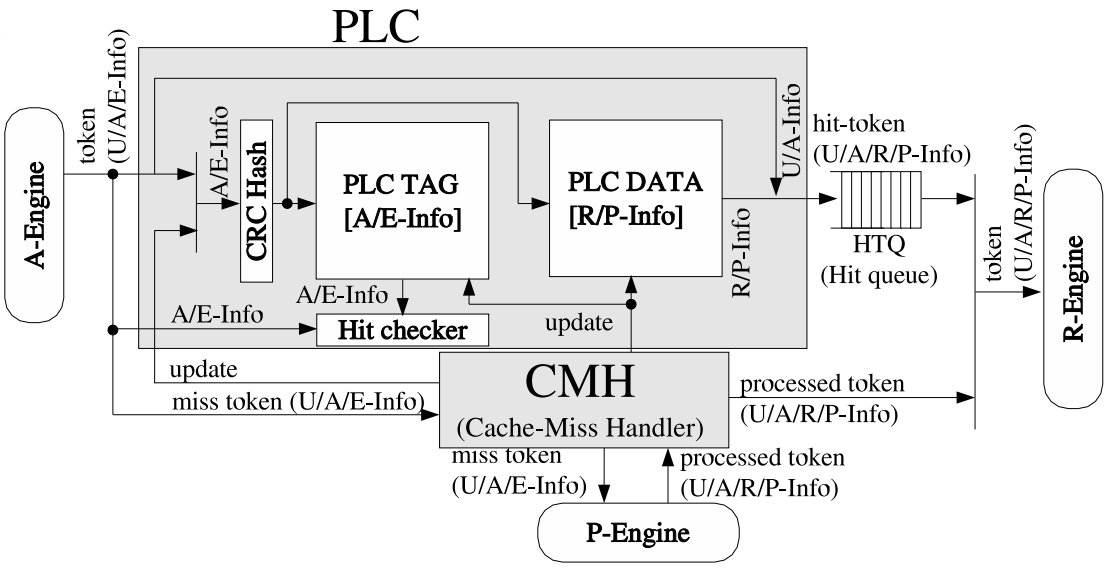

図 3.3: C-Engine のブロックダイアグラム

C-Engine は，Process Learning Cache (PLC) と呼ぶパケット処理結果や処理方法を記録するキャッシュメモリ機構と Cache-Miss Handler (CMH) と呼ぶ PLC に未登録のトークンを処理する機構を備える．また，CMH 経由で P-Engine を接続しており，C-Engine だけで処理ができない場合，補助的に P-Engine を利用する．PLC は HTQ (Hit Queue) と呼ぶ FIFO キュー経由で R-Engine と接続されている．HTQ は，後述の PLC ヒットしたトークンを一時的に保持し，CMH から PLC ミスして処理されたトークンが発行されていないタイミングで保持しているトークンを R-Engine へ送信する．R-Engine 側から観測すると，HTQ の存在により，A-Engine から送信されてくるトークンが PLC にヒットしてもミスしても，常時，C-Engine から連続的にトークンを受信できるようになり，CBNP のスループット低下を防ぐことができる．以下に，PLC と CMH に関してそれぞれの詳細を説明する．

PLC (Process Learning Cache) の役割と構成

PLC は，パケット処理のうちボトルネックとなる処理の結果，特に各種のルックアップやカプセリングヘッダ生成等の結果，更に，BSP の R-Engine における処理結果の適用方法を記録するためのキャッシュメモリである．PLC は，最初は全エントリが空の状態であるが，A-Engine からトークンを受信し始めると次々と登録されていく．登録されるのは，パケットフローの先頭のトークンの P-Engine での処理結果 (R/P-Info) である．尚，この登録は，トークンを受信してから CMH 経由で P-Engine にトークンが渡され，P-Engine での処理が完了してからであるため，その分のタイムラグがある．この点に関しては，CMH の項で追加説明するため，ここでは省略する．PLC に登録後は，後続の同一ヘッダとみなせるヘッダを持つパケットのトークン (A/E-Info が同一) には，P-Engine を一切利用せずに，PLC の内容をそのまま利用できる (R/P-Info を即座に渡す)．通信の

時間的局所性の存在が強く期待できる場合，大多数のパケットのトークンに対し，PLC から直接 R/P-Info を与えることができるため，P-Engine(通常のネットワークプロセッサの PE 集積部分) のスループットが低くても CBNP は高スループットのパケット処理を実現できる．

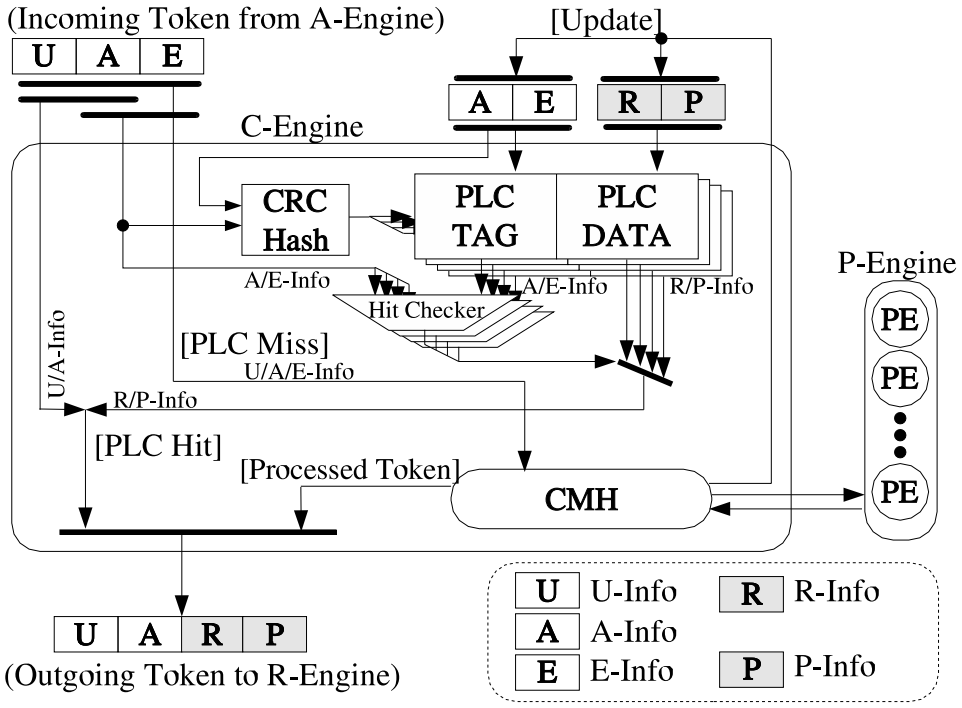

図 3.4: PLC のブロックダイアグラム

PLC は，通常のキャッシュメモリと同様，図 3.4に示すように，タグメモリとデータメモリで構成する．複数ウェイ構成の場合，リプレースウェイを決定するための LRU メモリも備える．また，ダイレクトマップ方式 (割当てエントリを一意に決定) やセットアソシアティブ方式 (割当てエントリをセット中のウェイのいずれかに決定)，フルアソシアティブ方式 (割当てエントリを自由に決定) 等の構成をとることができ，図 3.4の例では，4 ウェイのセットアソシアティブ方式の場合を示している (LRU メモリは省略している)．PLC の格納内容は，表 3.3に示す通りである．

表 3.3: PLC の格納内容

項目	格納内容
PLC TAG	トークンの A/E-Info と Valid bit
PLC DATA	トークンの R/P-Info
	複数の情報 (例：A-Info と E-Info) はスラッシュで区切って表記 (A/E-Info) する．

PLC TAG が有効な状態 (Valid bit が 1) において，入力トークンの A/E-Info が，PLC TAG の A/E-Info と一致すれば「PLC ヒット」，不一致であれば「PLC ミス」と呼ぶ．PLC のアクセス方法は以下の通りである．

1. トークンの A/E-Info を CRC ハッシングし，PLC TAG を参照する．PLC ヒットなら 2 へ，PLC ミスなら 3 へ進む

2. 該当する PLC DATA を読み出し，トークンの E-Info を R-Info に置き換え，P-Info を付加して HTQ 経由で R-Engine へ送信し終了

3. CMH へ PLC ミスしたトークンを渡す．当該トークンの処理が CMH で完了したら，PLC TAG を A/E-Info で，PLC DATA を R/P-Info で更新する．また，PLC ミスを起こしていたトークンは CMH でトークンの E-Info を R-Info に置き換え，P-Info を付加して R-Engine へ送信し終了 (詳細は次に示す「CMH の役割と構成」を参照)

PLC は，パケット処理結果や処理方法を記録するため，通常のキャッシュと異なる点を幾つか備えている．その違いは，タグ格納内容，ハッシング方法，キャッシュライン格納量，アクセス種類である．それぞれに関して説明する．

- タグ格納内容： 通常のプロセッサのキャッシュでは，アクセスアドレスのうちキャッシュをインデックスする下位数ビットを除いた上位ビットをタグとして利用する．PLC では，トークンの A/E-Info をタグとして利用する．

- ハッシング方法： ハッシング方法とは，PLC へのトークン (データ) 格納位置 (インデックス) を決定するための方法である．通常のプロセッサのキャッシュでは，データがどのアドレスに存在するかによって区別しており，目的のデータへのアクセスアドレスの下位数ビットをそのままインデックスとして利用している．PLC では，個々のパケットをトークンとして表現しており，同一フローに属するパケットからは解析結果 (A-Info) と抽出内容 (E-Info) が同じトークンが生成される仕組みである．そこで，トークンの A/E-Info が同じであれば PLC で同一のインデックスを利用する必要がある．しかしながら，A/E-Info は，単純なアドレスではなく，解析した情報と，パケットから抽出した複数のフィールドを結合したものであり， 大きさが 400bit 強に達するため，従来のように単純な一部のビットだけを利用することが極めて困難である．単純な IP フォワーディングでは，従来のキャッシュの研究のように，宛先 IP アドレスだけを利用しその一部の bit をインデックスとして利用することも可能であるが，5-tuple のような 5 フィールドの情報を抽出して優先度や宛先，処理方法を決定する場合には扱いが極めて困難である．そこで PLC では，A/E-Info 全体に対し CRC (Cyclic Redundancy Check，巡回冗長検査) ハッシングを行ない，得られた剰余をインデックスとして利用する．CRC は，連続する誤りを検出するための誤り検出符号の一種であり，十分な撹拌能力を持つため，誤り検出だけでなく，データ検索のハッシングキーとしても利用できる．また，入力データビット列から任意の長さのキーを生成することができるため，400bit 強の非常に長いビット列である A/E-Info から，十数 bit 前後の比較的短いビット列である PLC のインデックスを生成するために適している．CRC 回路は，複数段の XOR ゲートで容易に構成することができる利点もある．表 3.4に CRC ハッシングの演算多項式の例を挙げておく．

- キャッシュライン格納量： 通常のプロセッサのキャッシュでは，データの空間的局所性を利用するために，キャッシュタグが示すキャッシュデータ位置には複数のデータをキャッシュラインとして格納している．通常，キャッシュラインは複数データで構成されており，アクセスアドレスの最下位数ビットを利用して個々のデータを区別する．キャッシュラインに複数のデータを含んでいるのは，通常のプログラムでは，あるアドレスを始点にアドレスが連続する複数のデータを扱うことが多いため，連続したデータを同一タグで参照するキャッシュデータとして登録した方がキャッシュヒット率が向上するためである．PLC でも，宛先

表 3.4: CRC 多項演算式の例

剰余のビット数	メモリのエントリ数	CRC 多項演算式
8	256	$X^8+X^6+X^3+X^2+1$
9	512	X^9+X^5+1
10	1K	$X^{10}+X^7+1$
11	2K	$X^{11}+X^2+1$
12	4K	$X^{12}+X^3+1$
13	8K	$X^{13}+X^4+X^3+X^1+1$
14	16K	$X^{14}+X^5+1$
15	32K	$X^{15}+X^1+1$
16	64K	$X^{16}+X^5+X^3+X^1+1$
17	128K	$X^{17}+X^3+1$
18	256K	$X^{18}+X^3+1$

IP アドレスだけをインデックスに利用するのであれば，最下位数ビットが異なる IP アドレスは同一の宛先ネットワークである可能性が高いので，若干の空間的局所性がありヒット率向上が期待できる．しかしながら，PLC では 5-tuple のような複雑なトークン内容にも対応するため，トークンの A/E-Info を CRC ハッシングしてアクセスする．このため，従来のプロセッサで見られるようなアドレスの空間的局所性を直接利用することができない．よって，PLC のデータはキャッシュラインの概念を持たず，持ったとしても 1 データのみで構成するキャッシュラインとなる．

- アクセス種類 (リードオンリー)：通常のプロセッサのキャッシュは，データの読み出し，書き込みの両方を行なう．PLC は，先行トークンの処理結果を後続の同一フローのトークン (同一 A/E-Info を持つトークン) に適用するためのキャッシュであるため，読み出しを行なうだけであり，動作としては命令キャッシュに近い．このため，データキャッシュで問題となる書き戻し問題 (ライトスルーやライトバックポリシーの選択，そのための処理方法) が発生しない．ただし，後述するように，ルーティングテーブルとの間で情報を同期させるための PLC エントリ廃棄動作が存在する．

時間の経過と共に PLC には有効なエントリが増えていき，やがて，あるフローのトークンにとって空きエントリが無い状態が発生する．この場合，通常のプロセッサのキャッシュと同様に，PLC エントリのリプレースを行なってエントリを確保する．すなわち，PLC の有効なエントリと同一のエントリに割り当てられる後続の別フローのトークンが来た場合，PLC が既に古いフローに割当て済みであれば，古いフローのエントリを新しいフローで上書きする．上書きされた後は，古いフローは再び PLC ミスを起こし，新しいフローが PLC ヒットするようになる．これら新旧のフローが継続的に存在する場合，PLC の同一エントリは両フローで絶えず更新され続ける．これは，通常のプロセッサのキャッシュで，異なるアドレスのデータが同一のキャッシュエントリに割り当てられるために発生するピンポン現象に類似する．尚，あるフローがある PLC エントリを更新する前に別のフローが同一の PLC エントリを要求することもありうる．この場合，最終的に PLC に登録されるのは後続のフローである．尚，古いフローも新しいフローも，後述の CMH の異なる CMT エントリに割り当てられ，PLC 登録されるまでは CMH でキャッシュヒットと同等の扱いを受けることができる．CMT で同一のエントリに割り当てられるフローが続出しない限り

は，後続のトークン群は，ノンブロッキングで PLC アクセスを続けることができるため，CBNP は高いスループットを維持できる．

通常，PLC に一旦エントリが登録されると，当該エントリは新しいフローで上書きされるか，後述のルーティングテーブル更新が発生するまで存在し続ける．セットアソシアティブ方式の PLC の場合，新しいフローで上書きされる場合は，LRU 等のアルゴリズムによって上書きされるセットのエントリが決まるが，既に不要なエントリであることがわかるのであれば，不要なエントリを消去した方が効率が良い．クラシフィケーションの種類によっては，不要なエントリを明示的に削除する方法を採ることも可能である．例えば，5-tuple の場合，TCP フローを意識することが可能であり，TCP の FIN bit が 1，かつ ACK bit が 0 であれば，当該 TCP フローの最後のパケットであるので，PLC にエントリが存在する場合，当該エントリを明示的に消去することができる．

CMH (Cache Miss Handler) の役割と構成

PLC の項で触れたように，PLC への R/P-Info 登録には，P-Engine での処理が完了するまでのタイムラグがある．ネットワークのパケット転送処理においては，このタイムラグを放置すると致命的な性能低下を引き起こしうる．何故なら，PLC 登録が完了するまで，ネットワークから次々とやってくるパケットの処理が停止してしまうからである．このため，PLC 参照をノンブロッキング化するための仕組みが重要となる．PLC 参照のノンブロッキング化とは，PLC の hit under miss 処理と miss under miss 処理を意味する．

- **hit under miss 処理：**先行トークンで PLC ミス発生中に後続トークンの PLC ヒットを許し，PLC ミス処理のレイテンシの一部を PLC ヒット処理で隠蔽すること
- **miss under miss 処理：**先行トークンで PLC ミス発生中に後続トークンの PLC ミスを許し，複数の PLC ミス処理をオーバーラップさせて進めることで PLC ミス処理のレイテンシの一部を隠蔽すること

CMH は，PLC ミスしたトークンを連続的に受信して PLC 参照をノンブロッキング化する機能，受信したトークンを P-Engine に処理を割り当てる機能，また，P-Engine から処理済のトークンを受信し PLC を更新する機能を持つ．図 3.5に CMH のブロックダイアグラムを示す．また，表 3.5に CMH 内の機能部位と格納するトークンの内容一覧を示す．

表 3.5: CMH 内の機能部位と格納内容

機能部位	格納内容	備考
CMT (Cache Miss Table)	A/E-Info	PLC ミス中の全フローを管理するテーブル． フローの先頭トークンのみ P-Engine で処理．
CMQ (Cache Miss Queue)	U-Info	同一フロー中の各トークンを管理する FIFO キュー． 各 CMQ は CMT の各エントリに対応．
EWQ (Ex. Waiting Queue)	U/A/E-Info	P-Engine 処理待ちのトークンを管理する FIFO キュー
PRQ (Processing Result Queue)	R/P-Info	P-Engine 処理済みトークンを管理する FIFO キュー

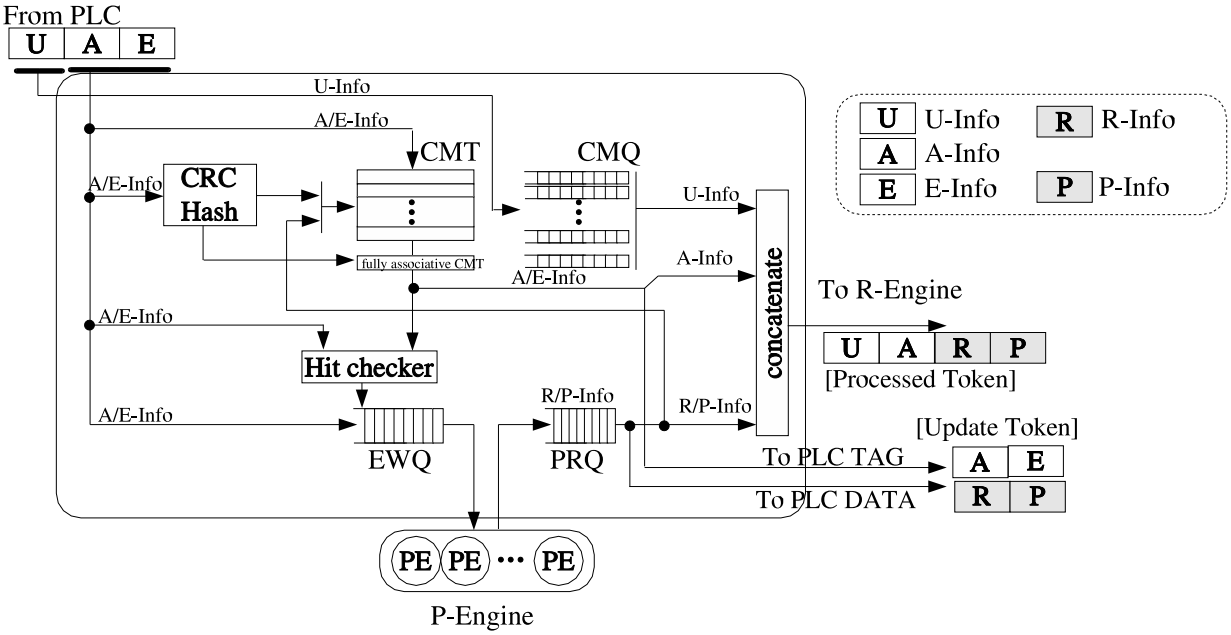

図 3.5: CMH のブロックダイアグラム

CMH は，CMT (Cache-Miss Table) と呼ぶテーブルで PLC ミス中のフローを管理するために，トークンの A/E-Info を記録する．また，CMQ (Cache-Miss Queue) と呼ぶ FIFO キュー群によって，PLC ミスした同一フロー中の個別のトークンを管理するために，トークンの U-Info を記録する．CMQ は，CMT のエントリと同じ数だけ存在し，各 CMQ が各 CMT エントリに対応する．また CMH は，CMT で管理中のフローにつき 1 個のトークンの U/A/E-Info を P-Engine で処理させるために，P-Engine 処理待ち用の FIFO キュー EWQ (Execution Waiting Queue) を備える．更に，P-Engine からの処理済トークンの R/P-Info を保持しておく PRQ (Processing Result Queue) と呼ぶ FIFO キューも備える．

CMT はフルアソシアティブのキャッシュであることが望ましいが，フルアソシアティブキャッシュは，CAM (Content Addressable Memory) を利用せねばならず，一般的なメモリを利用したキャッシュより登録から次回の検査までにかかるレイテンシが長くなるため，高速動作には向かない．また，面積も大きいため，エントリ数を多くとることができない．そこで，PLC と同様にダイレクトマップキャッシュか，セットアソシアティブキャッシュを利用するのが現実的である．CMT でのミス率を低下させるためには，セットアソシアティブキャッシュの方が望ましい．尚，注意しなければならないのは，CMT ではキャッシュウェイのリプレース動作を禁止せねばならないことである．これは，CMT が PLC ミス中のトークンを P-Engine での処理が終わるまで保持しておかないと，P-Engine で処理後のトークンとの辻褄あわせができなくなるためである．そこで，既に全ウェイを利用中のセットにトークンが割り当てられることになった場合に備えて，CMT に小規模のフルアソシアティブエントリを追加することが望ましい．図 3.5では，ダイレクトマップ型の CMT に 1 エントリのフルアソシアティブ CMT エントリを追加した例を示している．尚，CMT のヒット率を向上させるためには，2〜8 ウェイのセットアソシアティブ型の CMT に 8〜16 エントリ程度フルアソシアティブ CMT エントリを追加するような構成が現実的である．PLC 同様，A/E-Info 全体を利用して CMT を検査する必要があるため，CRC ハッシングを利用している．

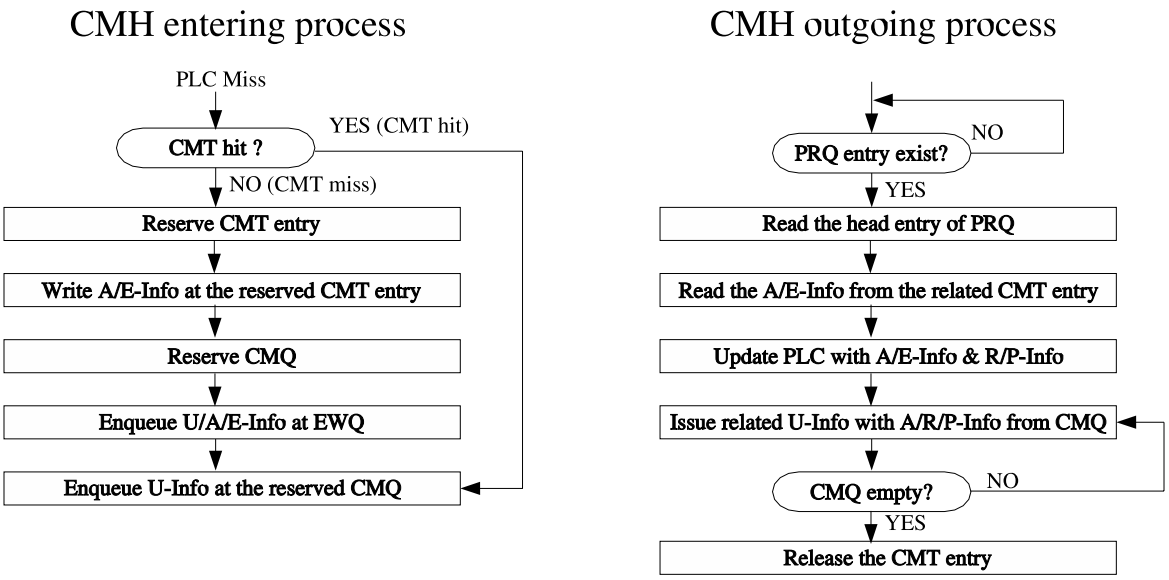

図 3.6: CMH 入出力処理のフローチャート

図 3.6に，トークンの CMH へ入出力処理のフローチャートを示す．CMH 入力処理 (図 3.6の左側) では，CMT でヒットチェックを行ない，受信 PLC ミストークンが CMH 管理中のトークンと同一フロー (A/E-Info が同一) か否かを検査する．同一でなければ，CMT ミスとなり，CMT のエントリを確保し，トークンの A/E-Info を登録する．そして，CMT のエントリに対応する CMQ を確保し，トークンの U-Info を書き込む．更に，P-Engine で当該トークンの処理を行なうために，EWQ にトークンの U/A/E-Info を書き込んで CMH 入力処理を終了する．また，CMT 検査で，トークンが同一であれば，CMT ヒットとなり，対応する CMQ の末尾に U-Info を書き込んで CMH 入力処理を終了する．

CMH 出力処理 (図 3.6の右側) では，PRQ に P-Engine からの処理済トークンがあるかチェックする．PRQ に有効なエントリが存在すれば，先頭からひとつ取り出し，対応する CMT エントリの A/E-Info を読み出す．そして，PLC のアクセスキーを CRC 演算により生成し，PLC タグを A/E-Info で，また，PLC データを R/P-Info で更新する．更に，対応する CMQ に保持されている U-Info，CMT の A-Info，PRQ の R/P-Info を結合して処理済トークンとして R-Engine に発行する．当該 CMQ が空になるまで発行を繰り返したら，CMT の当該エントリを解放して CMH 出力処理を終了する．

現在のネットワーク通信で広く利用される TCP/IP 通信では，トランスミッション制御プロトコルとして，TCP (Transmission Control Protocol) や UDP (User Datagram Protocol) が主に利用される．特に UDP ではパケットの送信順番と到着順番が変わってしまった場合に，順番を元通りにする仕組みが提供されていないため，ネットワーク層以下でパケット順番が入れ変わることは望ましくない．TCP ではパケット順番を元通りにするための機能が盛り込まれているが，順番整列操作は受信ホストにおいて CPU パワーの浪費と遅延増加に繋がるため，やはり，ネットワーク層以下でパケット順番が入れ変わることは望ましくない．CBNP の C-Engine では，次の操作によって同一フロー内のパケットの順序維持を保証する．

- PLC ミスした同一フローに属するパケットのトークンを同一の CMQ で FIFO 管理する
- CMQ のトークン発行を HTQ より優先的に行なう．(PLC 更新直後に PLC ヒットした後続パ

ケットのトークンが，CMQ に溜まっている先行パケットのトークンを追い抜かないようにする)

ルーティングテーブルの更新と PLC エントリの廃棄

ネットワーク上で，現在利用されている経路が削除されたり，新たな経路が追加されたりすると，ルータは BGP4 や RIP 等に代表されるルーティングプロトコルを利用して経路情報を互いに交換し，自身の持つルーティングテーブルを更新する．ここで，更新された経路に関する情報が PLC に残っていると，以後の対応する宛先へのパケット転送処理が誤ったものになってしまう．このため，ルーティングテーブルが更新された場合，PLC のエントリも併せて更新するか廃棄する必要がある．ここでは，廃棄するものとして説明する．尚，ハイエンドルータでは，制御プレーンを司る CPU がルーティングテーブルの本体を備え，データプレーンを司る各ラインカードのネットワークプロセッサは，ルーティングテーブルの情報のうち，転送処理に特化した情報だけを抽出した高速アクセスが可能なフォワーディングテーブルを備える．このため，正確には，ルーティングテーブルの更新があると，各ラインカードのフォワーディングテーブルが更新される．それを受けて，PLC の所望のエントリを廃棄する．

先に述べたように，PLC はトークンの A/E-Info 全体に対して CRC ハッシングを行なってインデックス付けしている．このため，クラシフィケーションの種類，すなわち，E-Info の抽出内容によっては，ある宛先 IP アドレスに対する PLC エントリを一意に見つけ出すことは困難である．このため，PLC を更新，または廃棄する方法として次の二通りの方法が考えられる．

1. ルーティングテーブルの更新があった場合，PLC の全エントリをフラッシュ(廃棄) する．
2. ルーティングテーブルに PLC の関連するエントリのインデックス番号を記録しておき，ルーティングテーブルの更新があった場合，PLC の関連するエントリのみをフラッシュ(廃棄) する．尚，ある宛先を含む PLC エントリが複数個存在する可能性があるため，ルーティングテーブルには，PLC の該当エントリ一つと，次の PLC のエントリのためのポインタを記録する．ふたつ目以降の PLC の該当エントリ一つと次のポインタは，別のメモリ上に記録し，最後のエントリはポインタ無しとする．

ルーティングテーブルは，利用するルーティングプロトコルによって更新の頻度が異なる．例えば，小規模ネットワーク向けの RIP (Routing Information Protocol) では，30 秒に 1 回，経路情報の交換が行なわれる．また，AS 間のような大規模ネットワーク向けの BGP では，経路が変化したときに経路情報の交換が行なわれる．CERNET BGP VIEW Global Internet[68] で採取された BGP テーブルの更新ログを調査すると，少ない場合は 1 秒に十数回程度であるが，多い場合には，数千回のオーダで更新がなされることがわかる(注 2)．更新頻度が十分低い場合は問題ないが，最悪の状態では，PLC の更新が CBNP の性能に悪影響を与えうる．

ここで，333MHz 動作の CBNP で，PLC に 2 サイクルピッチでアクセスする場合，PLC アクセスは最大 166M 回/秒行なわれる．仮に，10K 回/秒のオーダでルーティングテーブルが更新される場合，16.6K 回毎に PLC の更新が必要となる．方法 1 の場合，4K エントリの PLC では 16.6K 回に対し 4K 回と約 25%の時間相当の更新が必要となり，性能へ大きな悪影響を与えてしまうと考えられる．一方，方法 2 の場合，当該ルーティングテーブルエントリに関連する PLC エントリ

(注 2)例えば，2002 年 5 月 1 日の 19 時 15 分 42 秒には，2577 回/秒の更新があった．また，2004 年 12 月 21 日の 15 時 57 分 29 秒には，1217 回/秒の更新があった．

数によって性能への影響が変わる．全 PLC エントリに関連づけられていた場合は方法 1 と変わらないため，大きな悪影響を与えてしまう．しかしながら，唯ひとつの PLC エントリしか関連していなかった場合は 16.6K 回に対し 1 回と 0.006%の時間相当の更新でしかないため，性能への悪影響はほとんどない．100 エントリが関連していた場合でも，0.06%と性能への悪影響は非常に少ない．通常，ルーティングテーブルが 10K 回/秒のオーダで永続的に更新し続けられることは稀な状態であると考えられる．平均して 100 回/秒オーダ程度であるとすれば，PLC 更新の性能への悪影響は無視できるレベルと考えられる．以上より，理想的には方法 2 の PLC エントリ廃棄方法を実装するのが望ましい．

3.3.3 R-Engine の構成

R-Engine は，BSP の最下流部に位置し，受信したトークンの U-Info と A-Info，P-Info，及び設定レジスタの内容に従い，元のパケットを読み出し，必要であれば R-Info の内容でヘッダ修正を施し，出力するパケットを再構築するための機能ブロックである．第 2 章で紹介したネットワークプロセッサの機能に当てはめると，パケットのフィルタリングやヘッダ修正，フラグメンテーション，場合によってはトラフィックマネジメントを行なう部位に相当する．図 3.7に R-Engine のブロックダイアグラムを示す．

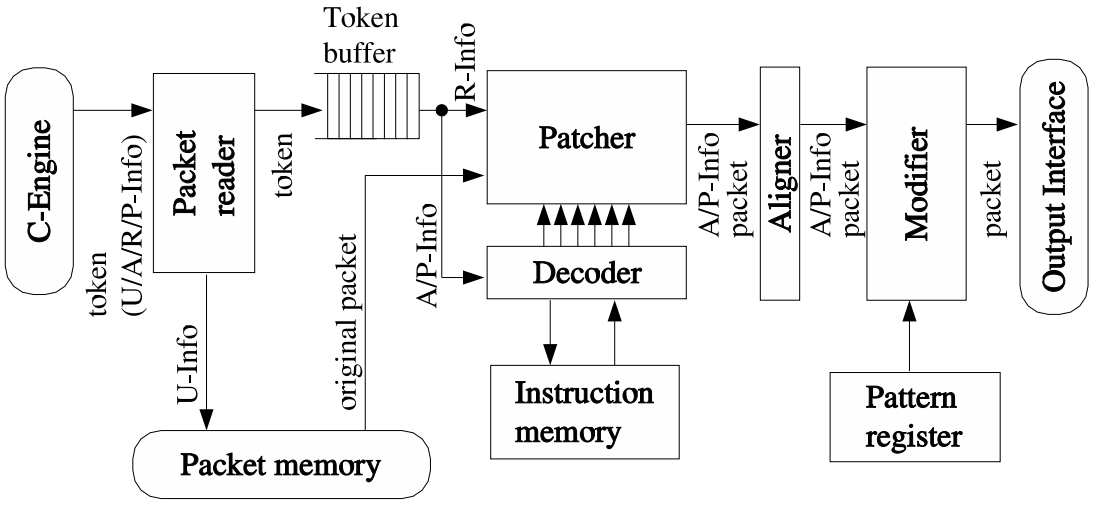

図 3.7: R-Engine のブロックダイアグラム

R-Engine は，C-Engine からの入力トークンをトークンバッファに保持し，同時にトークンの U-Info を参照して元のパケットをパケットメモリから読み出す．そして，パケットヘッダの追加，削除，置換を行なう場合，Patcher と呼ぶ byte 粒度のセレクタとバレルシフタを備える専用パイプラインを利用してパケットヘッダを修正する．具体的には，追加，置換の場合は，A/P-Info の内容を利用し，命令を命令メモリから対応する移動位置，移動量を示す命令を読み出し，R-Info に含まれている情報を追加，または置換する．この操作により，宛先/ 送信元 MAC アドレスの置換，IPv6 カプセリング，IPv4 カプセリング，MPLS カプセリング等の各種のカプセリング処理や，VLAN や MPLS ラベルの付替等を行なうことができる．また，削除の場合は，A/P-Info の内容を利用し，命令を命令メモリから対応する移動位置，削除量を示す命令を読み出し，パケットから対象部位を削除する．例えば，デカプセリング処理を行なうことができる．ヘッダの追加，置換，

削除処理が不要である場合は，Patcher を素通りさせ，Aligner へパケットをそのまま渡す．

Aligner では，ヘッダ修正によってパケット長が変った場合に，出力パケット長を調整する．最後に，Modifier と呼ぶ機能ブロックで A/P-Info とパターンレジスタの設定値を参照し，対象となるフィールド位置に適切な修正処理を行なう．例えば，IPv4 パケットの TTL (Time to Live) フィールドの更新やチェックサムフィールドの再計算等がこの処理に相当する．最後に，処理が終ったパケットを出力する．

3.3.4 P-Engine の構成

P-Engine は，受信したトークンから R-Info と P-Info を生成するのに利用するための機能ブロックである．第 2 章で紹介したネットワークプロセッサの機能に当てはめると，パケットのボトルネック処理 (テーブルルックアップやカプセリングヘッダ生成等) を行なう部位に相当する．P-Engine は，R/P-Info の生成には，やや複雑なプログラム処理や各種のテーブルルックアップ等が必要であり，PE を多数集積した従来型のネットワークプロセッサのアーキテクチャをそのまま適用できる．そのため，特定のアーキテクチャに限定する必要はない．

P-Engine が受信するトークンには A/E-Info が含まれているため，プロトコル情報は A-Info として，また，処理に必要な情報は E-Info として抽出されている．そこで，A/E-Info を元に，指定されているレイヤで必要なテーブルルックアップを行ない，その結果を P-Info として生成する．また，IPv4 から IPv6 アドレスへの変換や，IPv6 パケットを IPv4 パケットでカプセリングする (その逆もある)，MPLS や VLAN のヘッダを生成する，付け替える，といった処理の場合は，新規ヘッダ情報として R-Info を生成する．また，A-Engine で解析できなかった場合は，E-Info として収められているパケットの先頭部分 (パケットヘッダそのもの) に対して直接プログラム処理を行なって R-Info と P-Info を生成する．この場合，トークンを CMH へ返却しても，CMH は PLC へ登録しない．

一般に，PE の命令メモリは小規模であるため，想定内のパケット処理 (トークン生成) 用命令を保持できる程度と考えられる．そのため，例外的なパケットを処理するための命令そのものを全て置くことは非現実的であり，また，そのような例外的なパケットは数自体が少ないため，外部の CPU へ処理を転送するなどの処置も考えられる．

3.4 PLC に関する予備評価

3.1節で示したように，CBNP の最大スループット ***th(BSP)*** を維持するためには，キャッシュミス率分を補う ***th(BSP)*×*(1-h)*** 以上のスループットの P-Engine を集積すれば良い．この P-Engine には現在のネットワークプロセッサのアーキテクチャを利用することができる．第 2 章で示したように，現在のハイエンドネットワークプロセッサは半二重 10Gbps〜40Gbps 程度のスループットを実現している．近い将来必要とされる半二重 100Gbps 回線の処理を実現するためには，式 3.2，3.3より，***h* ≧ *0.9〜0.6*** となり，90%〜60%以上のキャッシュヒット率 (PLC ヒット率と CMH の CMT ヒット率の和) を実現できれば良いと考えられる．

$$th(BSP) = 100\ Gbps \tag{3.2}$$

$$th(BSP) \times (1-h) \leqq 10\ Gbps \sim 40\ Gbps \tag{3.3}$$

そこで，実際のネットワークトラフィックトレースを利用して，PLC が 90%〜60%以上のキャッシュヒット率を達成できるか確認するための予備評価をサイクルベースでパケットを取り込んで処理するソフトウェアシミュレータを構築して行なった．このソフトウェアシミュレータは，設定した PLC のエントリ数や連想度，ハッシング方法から与えたネットワークトラフィックトレースにおける PLC ヒット率を算出する．また，与えたネットワークトラフィックトレースに記載されている各パケットのタイムスタンプと P-Engine の処理時間とを考慮して，PLC にミスした P-Engine で処理すべきパケット (トークン) 数を算出することで，CMH の CMT に必要な資源量を評価する．

3.4.1 時間的局所性に関する予備評価

ソフトウェアシミュレータでダイレクトマップ (1 ウェイ・セットアソシアティブ)，4K エントリ，CRC 方式の PLC を実装し，表 3.6に示す 17 個のサイトのネットワークトラフィックトレースを利用して，PLC ヒット率の予備評価を行なった．結果を図 3.8に示す．いずれのサイトも，中規模〜大規模レベルのインターネットバックボーンである．明確な区分けは難しいが，多くはアクセス網であり，AIX，IPLS，MRA，TXS，WIDE はコア網と考えることができる．図 3.8に示すように，多くのアクセス網では 93%以上の高い PLC ヒット率が得られている．コア網を含めても 76.8%〜99.8%の範囲の PLC ヒット率が得られているため，33.2Gbps 以上の P-Engine を集積する CBNP は，半二重 100Gbps 回線の処理を実現できると考えられる．この P-Engine に必要とされるスループットは，現在のハイエンドネットワークプロセッサで実現可能な値である．

表 3.6: トレース採取サイト一覧

KEY	SITE	回線速度	利用パケット数	採取日
A2 [69]	University of Waikato	156 Mbps	17,795,023	2000/1/28
ADV [70]	Advanced Networks and Services	156 Mbps	235,240	2003/2/20
ANL [71]	Argonne National Lab. to STARTAP	156 Mbps	1,010,904	2003/2/20
APN [72]	APAN one-armed router at STARTAP	156 Mbps	339,812	2003/3/15
BELL [71]	Bell Labs	100 Mbps	4,047,317	2002/5/22
BUF [73]	University of Buffalo	156 Mbps	174,514	2003/3/15
BWY [74]	Columbia Univ.(BroadWaY)	156 Mbps	561,586	2003/3/15
COS [75]	Colorado State Univ.	156 Mbps	932,720	2003/3/15
MEM [76]	University of Memphis	156 Mbps	156,096	2003/3/15
ODU [71]	Old Dominion Univ.	156 Mbps	264,792	2003/3/15
OSU [71]	Ohio State Univ.	156 Mbps	354,676	2003/3/15
TXG [77]	Texas GigoPOP	625 Mbps	451,432	2003/3/15
AIX [78]	NASA Ames to MAE-West	625 Mbps	272,769	2003/3/15
IPLS [79]	Abilene IPLS router instrumentation	2.5 Gbps	20,000,001	2002/8/14
MRA [80]	Merit Abilene	625 Mbps	3,289,947	2003/2/23
TXS [71]	Texas Univ. GigaPOP at Rice Univ.	156 Mbps	17,207	2003/3/15
WIDE [81]	WIDE backbone trans-Pacific line B	100 Mbps	26,313,826	2003/2/27

3.4.2 PLC のエントリ数とヒット率に関する予備評価

ソフトウェアシミュレータでいくつかのパラメータを変化させながら，PLC のエントリ数とヒット率の相関関係を調査した [61, 62, 63]．日立製作所中央研究所 (HCRL) のネットワークトラフィックトレースを利用し，パラメータとして，表 3.7に示すように，PLC のメモリ構成の連想度を 1 ウェイと 4 ウェイの 2 通り，PLC のハッシング方法をアドレス情報をそのまま利用する RAW 方式とアドレス情報に CRC 演算の剰余を利用する CRC 方式の 2 通り，トレースのアグリゲーション方法をあり，なしの 2 通りとし，PLC のエントリ数を 256〜256K エントリまで変えながら PLC のヒット率を調査した．尚，PLC にミスした場合，次のサイクルには PLC に結果が登録され，次

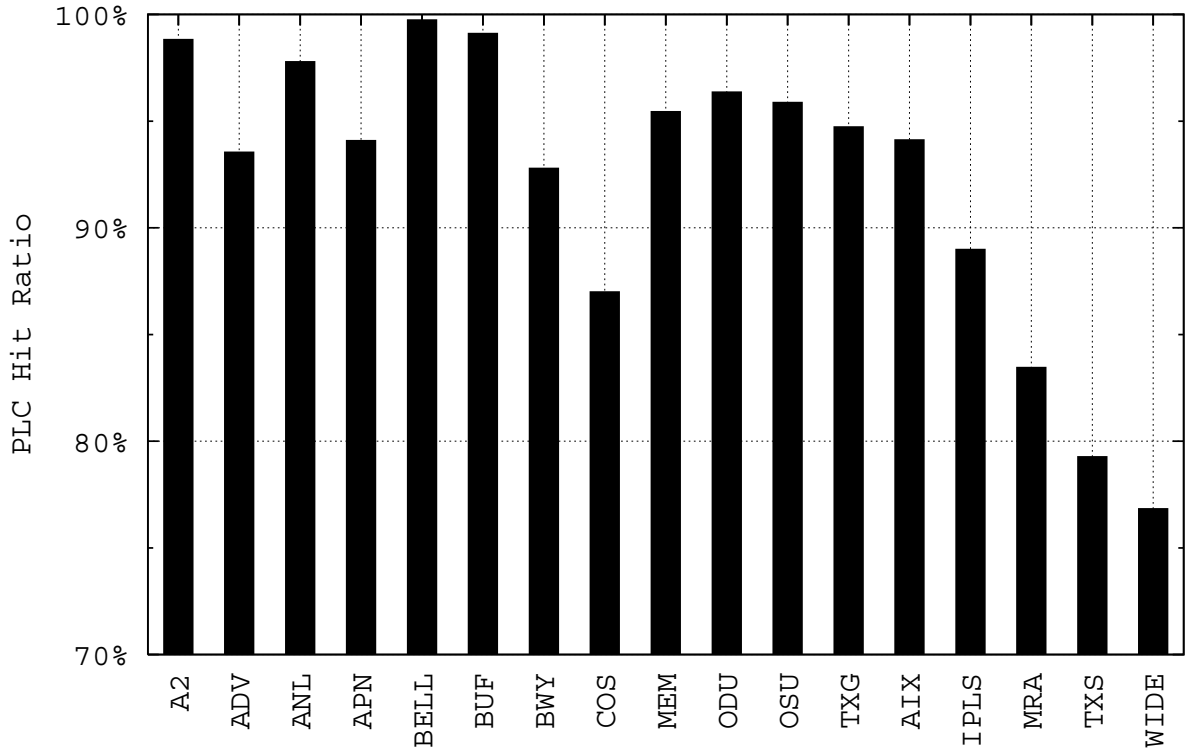

図 3.8: 17 サイトでの 1 ウェイ, 4K エントリの PLC ヒット率 (SRC/DST IPv4 アドレス利用)

のパケットからすぐに PLC の内容を利用できるものとした．実際には，PLC ミスが発生したら，CMH 経由で P-Engine に当該パケットのトークンを渡して複数サイクルかけて処理し，その後に PLC に登録することになる．しかしながら，CBNP のアーキテクチャでは，PLC ミス直後の同一ミスは CMH で保持し，後続のパケット処理をブロックしないように適用する仕組みを提供するため，PLC ミスが発生したら，次のサイクルから PLC の内容を利用できるというシミュレーションを行なっても問題ない．

表 3.7: 予備評価パラメータ

パラメータ	選択肢	概略
連想度	1 ウェイ	ダイレクトマップ方式
	4 ウェイ	リプレースアルゴリズムは LRU
ハッシング方法 (アクセスキー生成方法)	RAW	宛先/送信元 IP アドレスペアの下位ビットをそのまま利用
	CRC	宛先/送信元 IP アドレスペア全体を CRC ハッシングし利用
アグリゲーション	なし	採取トレースのパケット間隔だけを縮めて利用
	あり	採取トレースを 100 本マージして利用

ここで，ハッシング方法について補足説明を行なう．CBNP の PLC は，クラシフィケーションされたフロー毎に，パケットの処理結果と処理の適用方法を記録し，後続の同一とみなすことができるパケット (クラシフィケーションした結果が同一であるパケット) に適用する仕組みを提供する．この点で，宛先 IP アドレス毎に出力ポート識別子 (もしくはネクストホップアドレス等) のみを記録し，

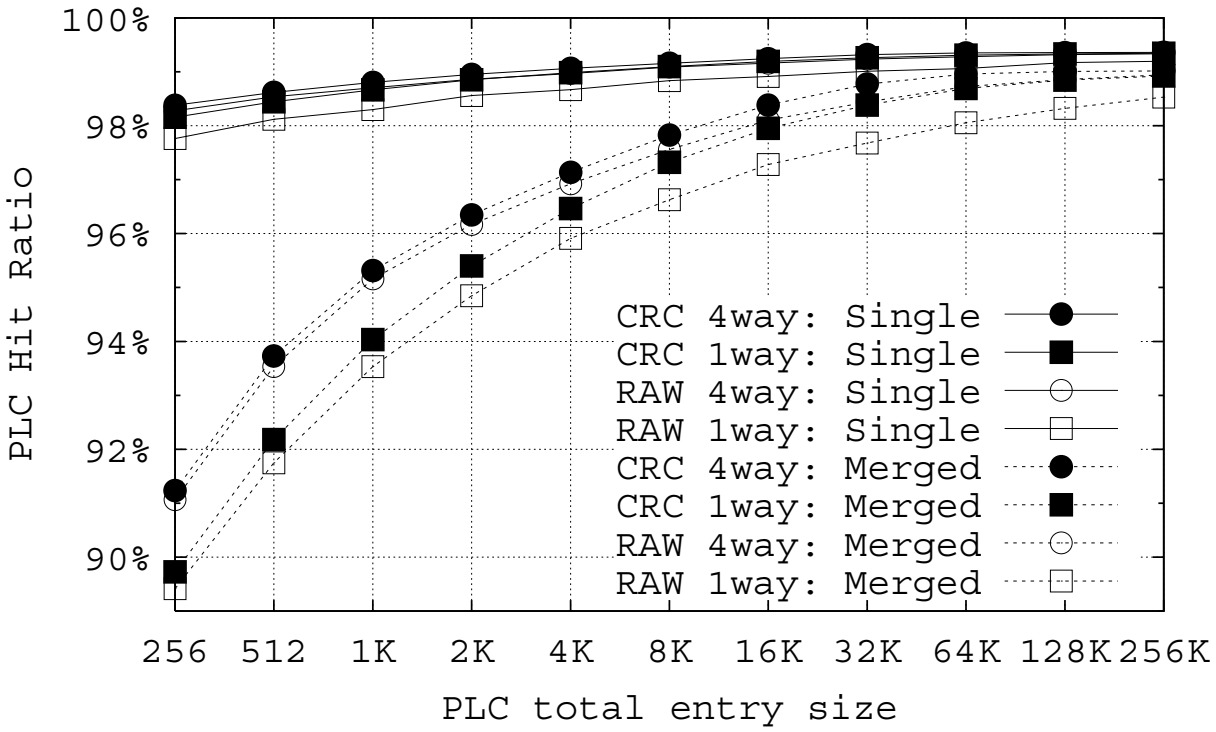

図 3.9: HCRL での PLC ヒット率 (SRC/DST IPv4 アドレス利用)

後続の同一宛先 IP アドレスのパケットに適用する従来の研究 [51, 52, 53, 54, 55, 56, 57, 58, 59, 60] とは異なる．この予備評価では，宛先 IP アドレスと送信元 IP アドレスの両方を利用するクラシフィケーションでパケットフローを識別し，PLC にアクセスすることとした．そこで，RAW 方式では，PLC のエントリ数が 2^{2N+A} である場合 ($A = 0$ or 1)，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 PLC を参照するためのアクセスキーとした．また，CRC 方式では，送信元 IP アドレスと宛先 IP アドレスを連結したもの全体に対して，表 3.4の多項演算式を利用して CRC 演算を行なって得られた剰余を PLC を参照するためのアクセスキーとした．

この予備評価で利用したトレースの採取元である HCRL は 1000 人規模の研究開発機関であり，ネットワークの区分ではアクセス網に相当する．このため，時間的局所性も非常に強く現われやすいと考えられる．同じトレースで時間的局所性の存在を弱めた評価，すなわち，大量のパケットフローが行き交うコア網，もしくは，将来，ネットワーク利用者や利用アプリケーションが増加した場合のアクセス網をエミュレートした評価も行なおうと考え，HCRL のトレースを分割した後，アグリゲーション (マージ，多重化) した評価も行なった．具体的には，HCRL の 34 日分のデータを 2 時間毎に分割した中から，パケット数の多かった上位 100 位分を集め，100 トレースを 1 トレースにアグリゲーションした評価を行なった．

図 3.9に結果を示す．横軸が PLC エントリ数，縦軸が PLC ヒット率を示し，凡例の Single はアグリゲーションなし，Merged はアグリゲーションありを示す．宛先/送信元 IP アドレスをアクセスキーに利用した場合でも，調査した PLC エントリ数の全範囲において，アグリゲーションなしの場合 97.7%以上の非常に高いヒット率を確認することができた．また，トレース 100 本を多重化したアグリゲーションありの場合でも 89.4%以上の高い PLC ヒット率を確認することができた．

これは，利用したネットワークトラフィックが非常に高い時間的局所性を備えていることを意味している．PLCエントリ数は多い程ヒット率が向上するが，アグリゲーションありの場合のヒット率向上がフラットになりつつある2K〜8K程度あれば良いと考えられる．また，いずれの場合も，現在の技術で実現可能な半二重10GbpsのスループットのP-Engineを集積すれば，CBNPは半二重100Gbps回線の処理を実現できる見込みである．

ここで，PLCの1エントリの容量は，トークンの各情報のサイズをA-Info[20bit]，E-Info[384bit]，R-Info[384bit]，P-Info[80bit]とすると，タグはA/E-Infoを格納するので404bit，データはR/P-Infoを格納するので464bitとなり，タグとデータを合わせると868bitとなる．これを約1Kbit程度と考えると，先の2K〜8Kエントリの PLCは2M〜8Mbit (250KByte〜1MByte)となる．0.13μm以下のCMOSプロセスを利用するLSIのオンチップに集積することを考えると，4Kエントリ(500KByte)程度が妥当な数字と考えられる．連想度は，通常のプロセッサのキャッシュ同様，4ウェイの方が1ウェイよりも高いヒット率を示した．エントリ数が少ない程，同一エントリに割り当てが競合するという，いわゆるコンフリクトミスが発生しやすいので，連想度の効果が高く得られていることがわかる．また，ハッシング方法はRAW方式よりもCRC方式の方が常に高いヒット率を示した．ただし，4ウェイと1ウェイの連想度差程の効果は得られていない．しかしながら，RAW方式は連想度によるヒット率差がエントリ数を増加させてもほとんど変わらないのに対し，CRC方式は連想度によるヒット率差がエントリ数の増加に従い小さくなっている．これは，CRC方式の撹拌能力の現われであると考えることもできる．

以上，予備評価からは，PLCは，4ウェイ・セットアソシアティブ，4Kエントリで，CRCハッシングを利用するのが妥当であると考えられる．

3.4.3 各サイトにおけるPLCのヒット率に関する予備評価

PLCのエントリ数とヒット率に関する予備評価から得た4ウェイ・セットアソシアティブ，4Kエントリで，CRCハッシングのPLCのヒット率に関する追加調査をソフトウェアシミュレータを利用し行なった[61, 62, 63]．追加調査には，表3.8に示す4つのサイトのネットワークトラフィックトレースを利用した．この調査でも，簡易なQoS制御等を行なうことを想定し，宛先IPアドレスと送信元IPアドレスの両方を利用することとした．すなわち，PLCには，宛先IPアドレスと送信元IPアドレスの両方を利用してCRCハッシングを行ない，アクセスすることとした．尚，パラメータとして，トレースのアグリゲーション方法をあり，なしの2通りを用いた．トレースをアグリゲーションした評価は，パケットフロー数が増加するであろう将来のコア網をエミュレートした評価とも考えている．

表3.8: トレース採取サイト一覧

type	KEY	SITE	回線速度	採取日
アクセス網	HCRL	Hitachi Central Research Laboratory	100 Mbps	2002/Apr-Jul
	A-IV[82]	University of Auckland uplink	156 Mbps	2001/3/27
コア網	IPLS[79]	Abilene IPLS router instrumentation	2.5 Gbps	2002/8/14
	WIDE[81]	WIDE backbone trans-Pacific line B	100 Mbps	2003/2/27

まず，採取トレースを利用した4つのサイトに関して簡単に説明する．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コア網の代表で，国外向け回線のトレースである．HCRLは，先の予

備評価で利用したトレースであり，アクセス網 (小規模サイト) の代表である．A-IV は， Auckland 大学 (3000 人のスタッフと，27000 人の学生が所属) 及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，コア網の代表である．尚，評価においてトレースをアグリゲーションして利用する場合，アグリゲーション後の回線速度が等しくなるように，WIDE，HCRL は約 100 本 (WIDE は 96 本)，A-IV は 64 本，IPLS は 4 本のトレースファイルをマージして利用した．

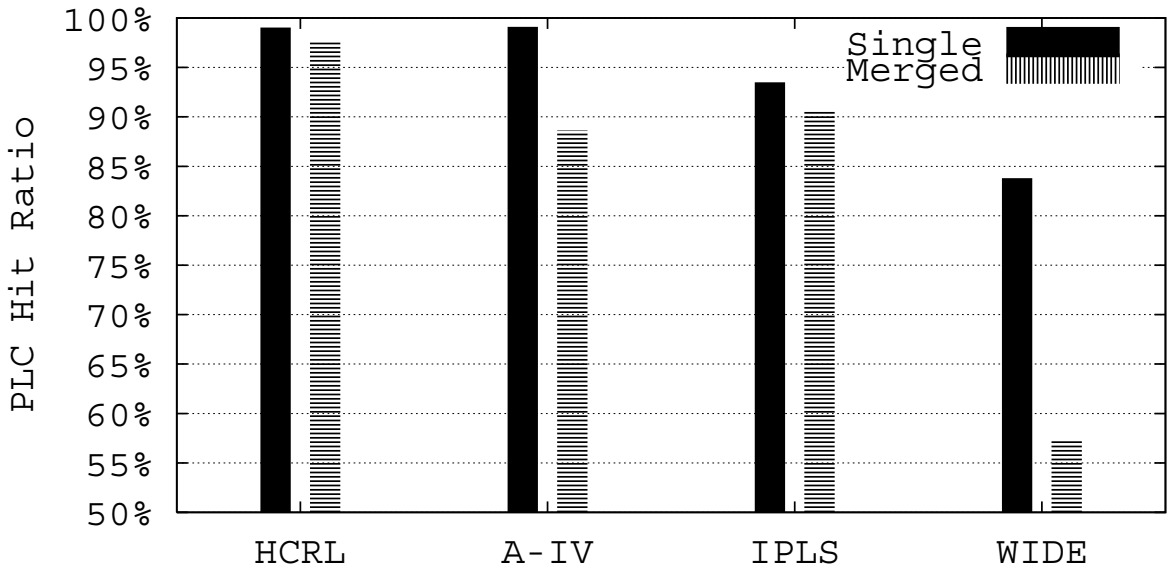

図 3.10: 4 ウェイ，4K エントリの PLC ヒット率 (SRC/DST IPv4 アドレス利用)

図 3.10に，各サイトにおける PLC ヒット率の結果を示す．アグリゲーションなしの場合，コア網，アクセス網ともに 83.8%以上の PLC ヒット率を示した．これは，CBNP に半二重 20Gbps の P-Engine を集積すれば半二重 100Gbps 回線のパケット処理を実現できることを意味する．また，アグリゲーションした場合，コア網の WIDE では，PLC ヒット率が 55.53%と 60%を下まわった．これは 100 回線アグリゲーションによって時間的局所性がほとんど無くなってしまったためであると考えられる．この場合，半二重 40Gbps のスループットの P-Engine を集積しても，CBNP のスループットは半二重 100Gbps に若干届かない．アクセス網では依然として 88.63%以上の高い PLC ヒット率を示し，CBNP に半二重 20Gbps の P-Engine を集積すれば半二重 100Gbps 回線のパケット処理を十分に実現できる．

以上より，4 ウェイ・セットアソシアティブ，4K エントリで，CRC ハッシングの PLC を持つ CBNP では，現在のアクセス網やコア網どちらにおいても半二重 20Gbps の P-Engine を集積すれば半二重 100Gbps 回線のパケット処理を実現できる見込みであると考えられる．ただし，コア網において，極端なアグリゲーションが行なわれる状況では，PLC が効果的に動作できない可能性があることも確認できた．

3.5 CMHに関する予備評価

CMHは，PLCミスを起こしたトークンを受信し，CMT (Cache Miss Table) を利用して同一フローに属するパケットのトークンが重複しないようにP-Engineへ割り当てる．また，同一フローに属するパケットのトークンはCMTの各エントリに対応するCMQ (Cache Miss Queue) へFIFO順に記録する．ここでは，CMTとCMQに必要なエントリ数について考察する．

3.5.1 CMTのエントリ数に関する予備評価

あるパケットのトークンがPLCミスを起こした場合，CMHのCMTでもミスを起こせば，P-EngineのPEで処理するが，一般にパケット処理で想定されている処理サイクル数は様々である．例えば，半二重20Gbps回線のパケット処理が可能なEZchip社のNP-1cでは，検索処理 (TOPsearchステージ) に32PEを用い，並列処理によって，PEあたり256サイクルの処理時間を確保するとされている [4]．この予備評価では，200サイクル〜2400サイクル程度をPEでの妥当な処理時間と考えて議論を進める．

同一フローに属するパケットだけが連続する場合，CMTのエントリ数は，P-Engineでの1トークンの処理時間分，すなわち，200サイクル〜2400サイクル程度の間に受信するパケットの数と同じだけ必要である．例えば，100Gbit Ethernetを想定すると，最小長のパケットは6.72nsに1個ずつ到着する．333MHz動作のCBNPを想定すると，最短で2.24サイクルに1パケット到着するので，200サイクル〜2400サイクルの間には，89個〜1072個のパケットが到着することとなる．先のPLCの予備評価より，ネットワークトラフィックには十分な時間的局所性があるため，この間に到着するパケットが全て異なるフローである可能性は低い．しかしながら，PLCをノンブロッキングとするためには，CMTのエントリ数は，全てが異なるフローのパケットである最悪の場合を考慮したエントリ数，すなわち，1000エントリ程度が望ましい．

また，CMTは，PLCミスを起こした全てのトークンを連続的に受信することでPLCをノンブロッキングにする役割を担う必要があるため，理想的にはフルアソシアティブメモリで構成することが望ましい．ただし，フルアソシアティブメモリはCAMに代表されるように，全エントリにタグの比較器を設ける必要があり，面積及び消費電力の増大を招く．特にCBNPでのタグは，トークンのA/E-Infoであり，400bit前後と非常に大きいため影響が大きい可能性が高い．そこで，面積と消費電力の増大を防ぐため，セットアソシアティブメモリを利用する手段も有力である．ただし，あるウェイが利用しつくされた場合，CMT内の他のエントリが空いていてもCMTを利用できないため，後続トークンのPLC参照をブロッキングしてしまう可能性がある．このようなケースでのブロッキングを緩和するために少数のフルアソシアティブエントリを設けるのが理想的である．一般にある程度の高速動作を前堤とするならば，4ウェイ程度が目安と考えられる．以上より，4ウェイ・セットアソシアティブ，1024エントリ (256エントリ/ウェイ) に加え，8〜16エントリ程度のフルアソシアティブエントリを持つCMTが，実現可能な実装例として考えられる．

予備評価として，CMTにどの程度のエントリ数が必要となるか調べるために，実際のネットワークトラフィックとして，PLCの予備評価でも利用したHCRLとWIDEのトレースを用いてソフトウェアシミュレーションを行なった．パケットフローは送信元IPアドレスと宛先IPアドレス (IPv4) で区別した．PLCは，4ウェイ・セットアソシアティブ，4Kエントリで，CRCハッシングとし，PEでの処理レイテンシを200サイクル〜3200サイクルの範囲で評価した．パケット長は，トレースに記録されているサイズそのままとし，パケット間隔を詰めることで100Gbpsの仮想Ethernet回線のワイヤレート処理を333MHz (1サイクルは3ns) で行なうことを想定した．

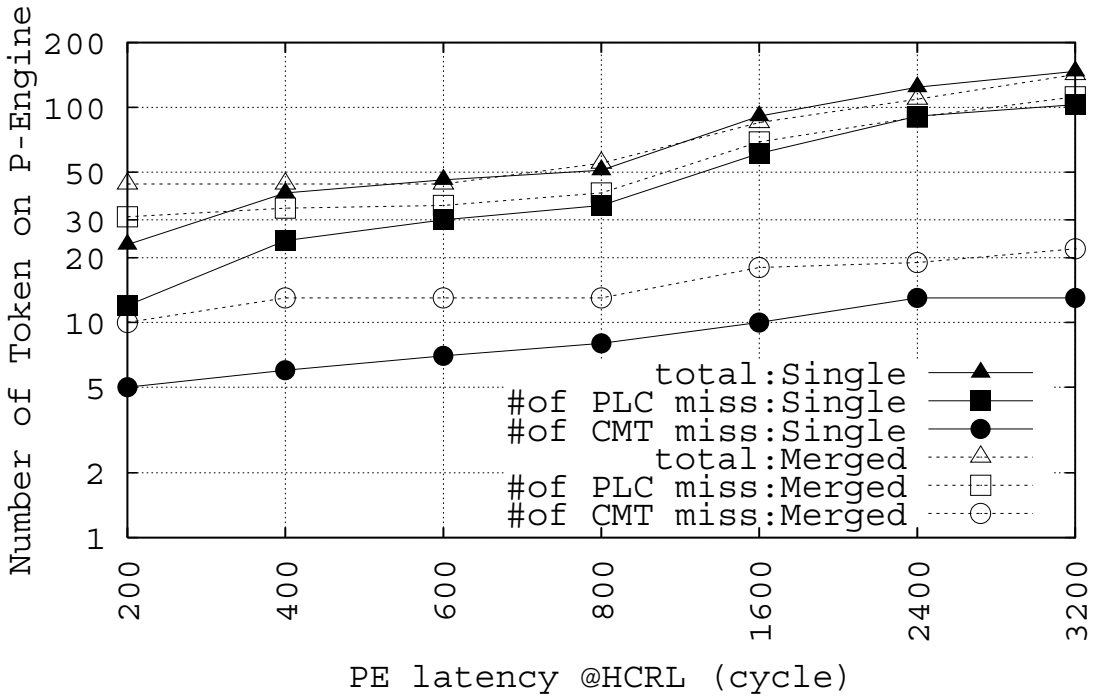

図 3.11: PE レイテンシと P-Engine 処理トークン数の関係 (HCRL トレース，仮想 100Gbps 回線, 333MHz 動作，SRC/DST IPv4 アドレス利用)

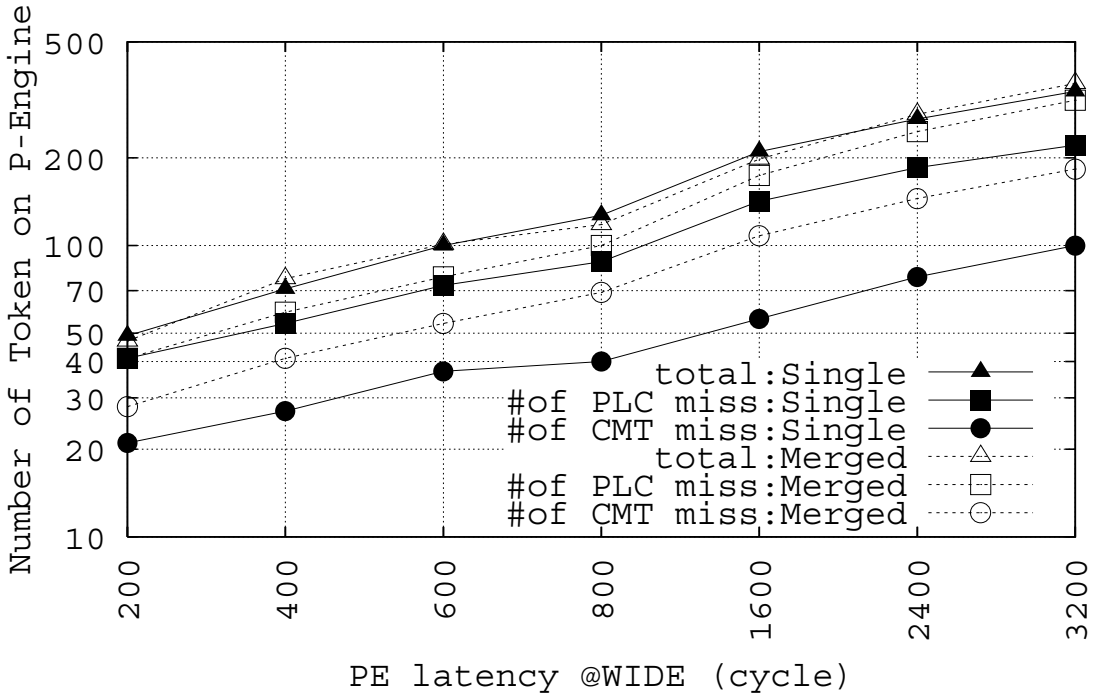

図 3.12: PE レイテンシと P-Engine 処理トークン数の関係 (WIDE トレース，仮想 100Gbps 回線, 333MHz 動作，SRC/DST IPv4 アドレス利用)

図 3.11に HCRL の結果，図 3.12に WIDE の結果を示す．横軸は PE のレイテンシ，縦軸は PLC ミスを起こして P-Engine で同時に処理を進めるトークン数 (パケット数) である．凡例の Single はアグリゲーションなし，Merged はアグリゲーションありを示す．また，total は PLC 参照を行なった総トークン数，# of PLC miss は PLC ミスしたトークン数，# of CMT は CMT ミスしたトークン数である．総アクセス数は，アグリゲーションあり/なしに関係なく同様の値となっているが，PLC ミス数，CMT ミス数は，アグリゲーションありの方がネットワークトラフィックの時間的局所性が弱められるため，増加している．ここで，PLC ミス数は CMT が同時に処理するトークン数を意味している．仮に CMH で CMQ を備えない場合，CMT でこの全てのトークンを管理することになる．また，CMT ミス数は，CMT が同時に処理するトークンのうち，異なるフローのトークンの数である．同じ ミスのトークンは，CMT では同一エントリを利用し，CMT に関連づけられた CMQ で管理する．このため，CMT ミス数は，CMT が管理しなければならないトークンの最大数を意味する．

コア網である WIDE の方がネットワーク通信の時間的局所性が HCRL より低く，また，平均パケット長も HCRL より小さいため，全ての領域でトークン数が多くなっているのがわかる．ここで，CMT に必要なエントリ数は，これらのグラフでは CMT ミス数に相当する．グラフからは，WIDE の場合でも 200 サイクル〜2400 サイクル程度とすれば，アグリゲーションなしで 21〜78 エントリ，アグリゲーションありで 28〜145 エントリである．このため，見積もった 1000 エントリは事実上，フルアソシアティブ構成の CMT であれば十分なエントリ数であると考えられる．また，4 ウェイ・セットアソシアティブの CMT (256 セット/ウェイ) と 8〜16 エントリ程度の CMT でも，CMT が利用する CRC ハッシングによって同一セットのコンフリクトミスが十分少なくなると考えれば，やはり十分なエントリ数であると考えられる．

3.5.2 CMQ のエントリ数に関する予備評価

CMQ は，同一フローに属するパケットのトークンを CMH 内で FIFO 順に保持する．同一フローに属するパケットだけが連続する場合，CMQ の深さは，CMT のエントリ数の考察同様，P-Engine での 1 トークンの処理時間分，すなわち，200 サイクル〜2400 サイクル程度の間に受信するパケットの数と同じだけ必要である．例えば，100Gbit Ethernet の場合，最小長のパケットは 6.72ns に 1 個ずつ到着する．333MHz 動作の CBNP を想定すると，2.24 サイクルに 1 パケットとなり，200 サイクル〜2400 サイクルの間には，89 個〜1072 個のパケットを受信することとなる．しかしながら，インターネット上では唯一つだけのフローが流れ続けることはないため，CMQ の深さを 89〜1072 とするのは過剰である．200 サイクル〜2400 サイクルの間に 10，または 20 フロー程度が存在すると仮定すると，CMQ の深さは，それぞれ，9〜108 または，5〜54 程度で十分と考えられる．

ここで，CMT のエントリ数の予備評価の結果から考えれば，CMQ のエントリ数が最大となるケースは，所定のサイクル数の間に 1 トークンしか出現しないフローが大量に出現し，大量のトークンが出現するフローが一つだけ出現するケースである．このとき，CMQ の最大エントリ数は，図 3.12の PLC ミス数から CMH ミス数を減じて 1 を加えたものとなる．すると，200 サイクル〜2400 サイクルの場合，14〜102 エントリとなる．また，大量のトークンが出現するフローが CMT ミス数の 10%とすると 5.3〜7.7 エントリ，5%とすると 7.5〜13.6 エントリとなる．以上を考慮すると，CMQ には 16 エントリ程度を用意すれば実用上は十分であると考えられる．

3.6 第 3 章のまとめ

ネットワーク通信における時間的局所性を有効に利用することで，PE 数を増加させずにスループットを増加させるためのネットワークプロセッサアーキテクチャ，Cache-based Network Processor (CBNP) を提案し，内部処理に利用するトークン，及び各部の詳細に関して説明した．CBNP は，PE での処理結果と処理結果適用方法 (Process) を併せて記録する Process Learning Cache (PLC) を備え，後続の同一とみなすパケットには PE を利用せず，PLC の内容を適用することで高スループット化を実現する．また，あるパケットを PE で処理中に同一とみなすパケットが到着する場合，PLC の目的のエントリが更新されるまで該当パケットを保持しつつ後続の別のパケットの処理を進めることで消費電力の高い PE の利用を抑止し，尚且つ，PLC をノンブロッキング化する仕組みを Cache Miss Handler (CMH) として提供する．CBNP は，PLC と CMH の CMT (Cache Miss Table) に未登録分を補うスループットを持つ P-Engine (PE の集合体) を集積することで，ワイヤレートのパケット処理を実現する．

PLC の CRC ハッシングの効果とヒット率及びメモリ量に関する予備評価結果からは，4K エントリ，4 ウェイ・セットアソシアティブの PLC を利用すれば，100Gbps 回線の処理を実現するために，現在の技術で実現可能な 10Gbps〜40Gbps 程度の P-Engine を集積するだけでよい可能性を示した．また，CMH の CMT は 1K エントリ，各 CMT に対応する CMQ (Cache Miss Queue) は 16 エントリ程度あれば実用上十分である可能性を示した．

第4章　CBNPのプロトタイプを利用した評価

CBNP (Cache-based Network Processor)の動作検証とワイヤレートのパケット転送処理を実現できることを検証するために，FPGA で構成したテストベッド上に CBNP のプロトタイプを実装し，コア網である WIDE の実ネットワークトレースを利用してキャッシュヒット率とスループットを評価した．本章では，まず，評価方法について説明し，次いで，実装した CBNP のプロトタイプの構成，利用トレースについて示し，最後に考察する．

4.1　評価方法

CBNP の有効性を検証するためには，実ネットワーク環境下において実動作するハードウェアを利用するのが理想的である．この目標に可能な限り近づけるために，FPGA で構成したテストベッド上に，ハードウェア記述言語 Verilog HDL で設計した CBNP のプロトタイプをマッピングし利用した．そして，tcpdump(注 1)形式で採取されている実際のネットワークトラフィックデータを，Linux PC から tcpreplay と呼ぶコマンドを利用して FPGA テストベッドに送信し，CBNP にパケット転送処理をさせ，キャッシュヒット率とスループットを計測した．尚，CBNP のキャッシュヒット率とは，PLC ヒット率と CMH ヒット率の総和を意味する．スループットとは，パケットの総出力 Byte 数を総入力 Byte 数で割った値を意味する．評価の際には，CBNP でキャッシュミスした際にパケットを処理する P-Engine のスループットを CBNP 内部のメインパスである BSP (Burst Stream Path) に対して 10%〜40%の間で変化させた．

尚，全ての評価は，CBNP に対する負荷が最も高くなるワイヤレート状態で行なった．ワイヤレートとは，最大回線速度の状態でパケット処理することである．ワイヤレートでパケット入力する場合，Ethernet では，パケットとパケットの間には最小のインターフレームギャップ (12Byte) とプリアンブル (8Byte) だけが挿入された状態で送信することになる．更に，ワイヤレート状態の中でも最高負荷状態での評価をするために，パケット長を最小長に変化させた場合の評価や 2 回線のデータをマージした場合の評価も行なった．2 回線マージの評価は，パケットフロー数が増加するであろう将来のコア網をエミュレートした評価とも考えている．いずれの場合も，CBNP を高機能なルータ装置でパケット転送処理に利用することを想定し，QoS を考慮した 5-tuple によるクラシフィケーションとルーティング処理を行なうことにした．すなわち，パケットから宛先 IP アドレスだけを抽出するのではなく，宛先 IP アドレス，送信元 IP アドレス，プロトコル，宛先ポート番号，送信元ポート番号の 5 情報を抽出してトークンを生成し，CBNP のキャッシュである C-Engine の PLC を参照することとした．PLC 未登録のフローは PLC ミスを起こし，CMH 経由で P-Engine でルーティング処理，5-tuple による ACL 処理を行なってトークンの R/P-Info を生成して PLC を更新する．尚，今回の実験では PLC と CMH によって CBNP が回線速度の処理を実現できることを実証する点に焦点を絞っている．このため，P-Engine はルーティング処理や

(注 1)tcpdump とは，コマンドライン上で利用する一般的なコンピュータネットワーク調査ツールであり，ネットワーク上を流れる TCP/IP 等のパケット情報を採取することができる．

ACL 処理そのものを実施せず，PLC ミス受信後，所定の時間が経過したら予め準備しておいた対応する R/P-Info を返却する単純なモデルを実装した．また，3.3.2節ではルーティングテーブルの更新と PLC エントリの廃棄に関して議論したが，議論の方法 2 の手法によれば，ほとんど性能への影響を無視できることから，本実験ではルーティングテーブルの更新操作は行なわない単純なモデルとして評価した．

4.2 CBNPプロトタイプの構成

FPGA テストベッドの全体像を図 4.1に示す．FPGA テストベッドの主要構成要素は，FPGA を 5 個，パケット保持用のメモリを 2 枚，ネットワークインタフェース用の Gigabit Ethernet を 2 本である．FPGA には，Virtex-II Pro と呼ばれるシリーズ [83] の xc2vp70 を利用した．xc2vp70 は，4 入力 1 出力の LUT(Look Up Table) とフリップフロップのペアを 2 個備えたスライスと呼ぶ基本ブロックを 33088 個備えている．また，スライス内の LUT とフリップフロップ，及び，桁上げ用論理を合計 74448 個，BlockRAM と呼ぶメモリブロックを 5904Kbit 分備えている．

FPGA テストベッドの概観を図 4.1に示す．CBNP のプロトタイプをハードウェア記述言語 Verilog HDL で設計し，論理分割を行なって，このテストベッド上の各 FPGA に Xilinx 社の CAD ツール ISE の Project Navigator 7.1.04i を利用してマッピングした．CBNP のプロトタイプは，33MHz のシステムクロックで動作し，内部の最大回線処理速度は 2Gbps 相当である．また，トークンを構成する各情報の bit 数は，U-Info[22 bit]，A-Info[12 bit]，E-Info[336 bit]，R-Info[336 bit]，P-Info[30 bit] である．このため，A-Engine が生成する U/A/E-Info を含むトークンは 370bit，C-Engine で置換後の U/A/R/P-Info を含むトークンは 400bit となる．

表 4.1: 各 FPGA の機能

	機能名	機能
FPGA1	IFA (Input Flow Accelerator)	入力パケットの一次保持，ワイヤレート生成
FPGA2	A-Engine	パケット解析，U/A/E-Info 生成
FPGA3	C-Engine (PLC)	U/A/E-Info を U/A/R/P-Info に置換
FPGA4	C-Engine (CMH) & P-Engine	キャッシュミス処理と R/P-Info 生成
FPGA5	PMEM & R-Engine	パケット保持と U/A/R/P-Info 適用

各 FPGA にマッピングした機能を表 4.1と図 4.2に示す．5 個の FPGA のうち，1 個の FPGA (FPGA1) で CBNP と外部入出力とのインタフェースを行ない，残りの 4 個の FPGA (FPGA2〜FPGA5) に CBNP の各機能を割り振った．まず，FPGA に割り振った各機能について説明する．

1. **FPGA1 (IFA)：**
   FPGA1 には，Gigabit Ethernet 用の MAC (Media Access Control) 論理を 2 組割当て，外部と CBNP との通信に利用した．また，IFA (Input Flow Accelerator) と呼ぶ論理を割当て，外部入力を一旦保持してからワイヤレートで CBNP へ送信する機能を持たせた．ネットワークトラフィックデータの再現に利用する tcpreplay コマンドは，Linux PC からソフトウェアで送信処理をするため，50Mbps〜100Mbps 程度のスループットしか出すことができない．この状況下で CBNP に対する負荷が最も高くなるワイヤレート状態の入力を実現するために，IFA には，ある程度の量のパケットを一旦溜め込んでからワイヤレートで CBNP へ入力する機能を持たせた．具体的には，IFA は，2 個の外付けの 256MByte の DDR-SDRAM を FIFO

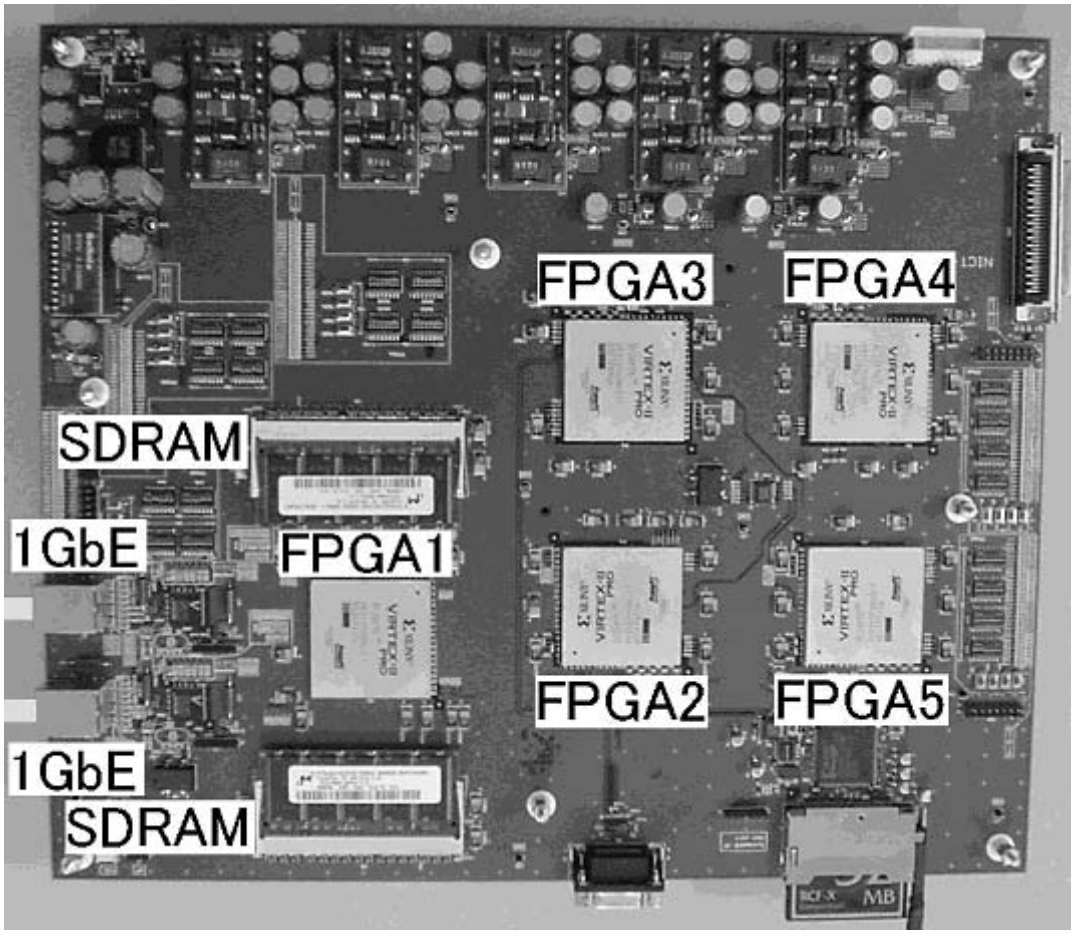

図 4.1: FPGA テストベッドの概観

(First In First Out) キューとして制御する．まず，片方の FIFO キューが一杯になるまでパケットを保持し，一杯になったところで，溜まっているパケット群全てをワイヤレートで CBNP に送信する．送信中は，Gigabit Ethernet 側からの受信は他方の FIFO キューに切り替え，こちらも一杯になったら先と同様に溜まっているパケット群全てをワイヤレートで CBNP に送信する．この操作を繰り返すことで CBNP が常にワイヤレートでパケットを受信できる状態を作りだす．また，パケット長が最小のとき，最小間隔でパケット処理をしなければならないため，CBNP にかかる負荷は最大となる．この高負荷環境を作りだすために，IFA にはパケット長を変化させる機能も持たせた．パケット長最小モードでは，FIFO キューにパケットを書き込む際に最小長 (Ethernet の場合，64Byte) 分だけを書き込むことで実現する．また，IFA には，2 本の Ethernet 回線をアグリゲート (マージ) する機能も持たせた．アグリゲートモードでは，互いに無関係なトレースがマージされた状態で CBNP にパケット送信されるため，局所性の存在を弱めた評価を行なうことができる．

2. **FPGA2 (A-Engine)：**
   FPGA2 には，A-Engine の機能を割り当てた．A-Engine には，レイヤ 2，レイヤ 2.5，レイヤ 3，レイヤ 4 解析用のパイプラインを実装し，5-tuple 抽出用のマイクロコードを収めた．その他のマイクロコードを収めることで，5-tuple 以外のクラシフィケーション結果を抽出することも可能である．A-Engine は，パケット本体を FPGA5 が持つ PMEM (Packet Memory) に

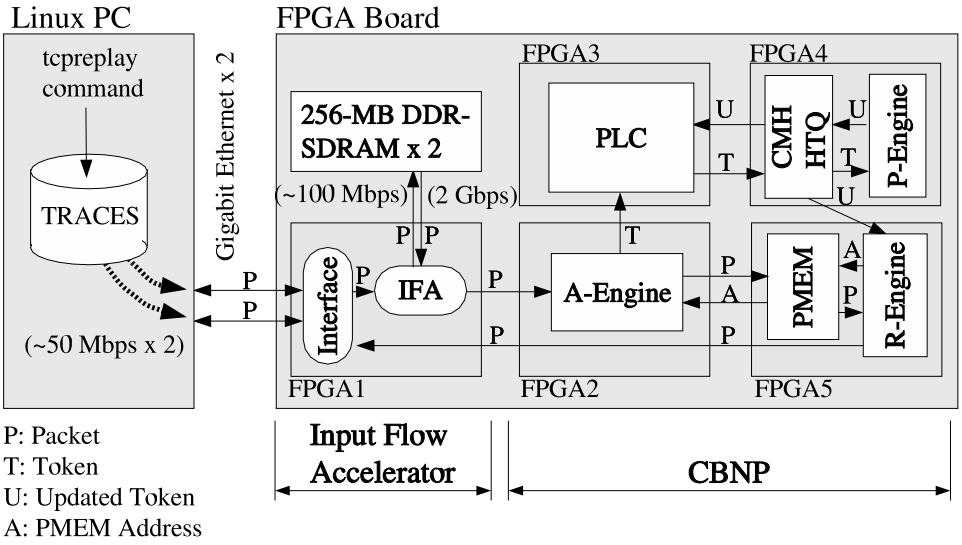

図 4.2: FPGA で構成した CBNP のプロトタイプ

書き込み，PMEM の書込みアドレスを U-Info の一部として受け取る．最終的に U/A/E-Info を含むトークンをパケットあたり 1 個生成し，PLC に渡す．

3. **FPGA3 (C-Engine (PLC))：**
   FPGA3 には，C-Engine のうち PLC (Process Learning Cache) の機能を割り当てた．PLC の諸元を表 4.2に示す．3 章の予備評価結果に基づき，PLC のエントリ数は 4096 とし，4 ウェイ・セットアソシアティブとした (1024 エントリ/ウェイ)．PLC のアクセスキー生成には CRC ハッシングを利用する．リプレースアルゴリズムには擬似 LRU を用いた．キャッシュラインサイズは，通常のプロセッサでいうところの 1 に相当し，対応する R/P-Info が唯一つ格納される．FPGA3 が FPGA4 に送信するトークンは，PLC ヒットであれば U/A/R/P-Info であり，PLC ミスであれば U/A/E-Info である．

表 4.2: PLC の諸元

連想度	4 ウェイ・セットアソシアティブ
リプレースアルゴリズム	擬似 LRU
アクセスキー生成方法	CRC ハッシング (生成多項式：$X^{10}+X^7+1$)
エントリ数	4096 (1 ウェイあたり 1024)
キャッシュラインサイズ	1 (対応する R/P-Info を 1 個のみ)
タグメモリ容量	175 KByte (A/E-Info，valid bit 格納．349bit × 4K)
データメモリ容量	183 KByte (R/P-Info 格納，365bit × 4K)
LRU メモリ容量	3 Kbit (3bit × 1K)

4. **FPGA4 (C-Engine (CMH) と P-Engine)：**
   FPGA4 には，C-Engine のうち CMH (Cache Miss Handler) と HTQ (Hit Queue)，P-Engine の機能を割り当てた．CMH と HTQ の諸元を表 4.3 に示す．

   CMH の CMT (Cache Miss Table) のエントリ数は 1024 とし，物量を抑えるために，フルア

表 4.3: CMH と HTQ の諸元

CMT	連想度	4 ウェイ・セットアソシアティブ + 小規模フルアソシアティブ
	アクセスキー生成方法	CRC ハッシング (生成多項式：$X^8+X^6+X^3+X^2+1$)
	エントリ数	1024 (256 エントリ/ウェイ) + 8(フルアソシアティブ)
	CMT エントリ容量	44.0 KByte (A/E-Info，valid bit 格納．349bit × 1K+8)
CMQ	本数	1024 + 8 (CMT の各エントリに対応)
	CMQ あたりのエントリ数	16
	CMQ 総エントリ容量	44.4 KByte (U-Info 格納．22bit × 16 × 1032)
EWQ	エントリ数	256
	EWQ エントリ容量	11.6 KByte (U/A/E-Info 格納．370bit × 256)
PRQ	エントリ数	256
	PRQ エントリ容量	11.4 KByte (P/R-Info 格納．366bit × 256)
HTQ	エントリ数	128
	HTQ エントリ容量	6.25 KByte (U/A/P/R-Info 格納．400bit × 128)

ソシアティブ構成ではなく PLC と同様の 4 ウェイ・セットアソシアティブ構成とした．また，後続の PLC ミスが既に利用されている CMT の同一セットを要求する場合に備え，8 エントリの小規模フルアソシアティブ CMT エントリを設けた．CMT のエントリは，P-Engine で対応する R/P-Info が生成され，PLC 更新と CMH 保持トークンへの適用が完了すると解放される．このため，通常のキャッシュに必要なリプレースアルゴリズムは不要である．また，各 CMQ の深さは第 3 章の考察に基づき，16 エントリとした．

P-Engine は，プログラマブルなマルチプロセッサアレイと経路検索用のルーティングテーブルを備える部分である．しかしながら，今回の試験では，CBNP のキャッシュヒット率とスループットを調査するのが主眼であるため，P-Engine は A/E-Info を受信すると予め設定したタイミングで予め決めておいた R/P-Info を返却するだけの仕組みとした．この R/P-Info を返却するまでの時間を調整することで BSP (Burst Stream Path) に対する P-Engine のスループットを表現することができる．FPGA4 は，FPGA3 で PLC ヒットしている場合，HTQ に U/A/R/P-Info を含むトークンを受信する．PLC ミスしている場合，CMH に U/A/E-Info を含むトークンを受信しキャッシュミス処理を開始する．キャッシュミス処理が完了すると，FPGA3 には PLC 更新のためのトークンを，また，FPGA4 には U/A/R/P-Info に置き換え済みのトークンを送信する．

5. **FPGA5 (PMEM (Packet Memory) と R-Engine)：**
FPGA5 には，パケットメモリと R-Engine の機能を割り当てた．今回の試験では，5-tuple によるルーティング処理を行なうだけである．このため，元のパケットが持つ送信元/宛先 MAC アドレスを，トークンの R-Info にある送信元/宛先 MAC アドレスに付け変えるためのマイクロコードを R-Engine に収めた．また，パケットメモリは 256Byte の区画を 2048 個備え，最大で 512KByte 分のメモリを実現している．このパケットメモリは，A-Engine から受信したパケットを 256Byte 毎に分割して格納し，先頭の区画を示すアドレスを U-Info として返却する．また，R-Engine からの要求に従い，トークンの U-Info に示される区画を先頭にパケットを取りだす．

最後に，表 4.4に各 FPGA の機能と内部資源利用率を示す．また，パケット生成環境である Linux PC の諸元を表 4.5に示す．Linux には特殊な設定は一切加えておらず，デフォルトの状態で利用した．また，二つの Gigabit Ethernet のうち一つは PC のマザーボードに備え付けのものを利用し，もう一つはネットワークインタフェースカードをマザーボードの PCI バスに接続して利用した．

表 4.4: 各 FPGA の機能と内部資源利用率

	Function	SLICE	LUT	RAM
FPGA1	IFA	40%	29%	27%
FPGA2	A-Engine	48%	39%	10%
FPGA3	C-Engine (PLC)	19%	10%	56%
FPGA4	C-Engine (CMH) & P-Engine	43%	34%	47%
FPGA5	PMEM & R-Engine	6%	5%	72%

表 4.5: パケット生成環境 (Linux PC) の諸元

CPU	Pentium4 (3.4GHz)
Memory	2-G bytes
Network	Gigabit Ethernet x 2
OS	Debian GNU/Linux 3.0
tcpreplay command	version 2.3.3

4.3 利用トレース

CBNP の評価に利用する実ネットワークトラフィックデータとして，WIDE (Widely Integrated Distributed Environment [81]) で公開提供されているバックボーンルータにおける 24 時間トレース (サンプルポイント B) を選択した．これらのトレースの採取日と本論文中での呼称 (WIDE1, WIDE2，WIDE3) の関係を表 4.6に示す．このサンプルポイントでは，100Mbps 回線が利用されている．

表 4.6: WIDE トレース

呼称	採取日
WIDE1	2003 年 2 月 27 日全日 (15 分間 ×96 本)
WIDE2	2005 年 1 月 7 日全日 (15 分間 ×96 本)
WIDE3	2005 年 9 月 22 日全日 (15 分間 ×96 本)

各トレースは，15 分間毎に 1 ファイルにされており，96 ファイルで 24 時間分となっている．CBNP の評価の前に，これらのトレースの内容を調査した．その結果をまとめたのが，図 4.3，4.4，4.5であり，それぞれ，採取パケット数，平均スループット，平均パケット長を示している．各図の各点が 1 ファイル，すなわち，15 分間トレースを示している．トレース採取ポイントの回線速度は 100Mbps であるため，図 4.4より，WIDE トレースの平均スループット (この場合は，回線への平均的な負荷のかかり具合) は 20%〜30%程度でしかないことがわかる．今回の試験において

は，FPGA1 に実装した IFA の機能によって，入力パターンを変えることなく，ワイヤレート状態，すなわち，入力間隔を最小間隔まで狭めて回線負荷 100%に相当した状態で CBNP へ入力する．

評価には，他の時間帯より比較的パケット数が少なく，平均スループットも低めであった 06:00-06:59 (15 分間のトレース 4 本分)，及び他の時間帯より比較的パケット数が多く，平均スループットも高めであった 15:00-15:59 (15 分間のトレース 4 本分) と 22:00-22:59 (15 分間のトレース 4 本分) の時間帯のトレースを用いた．参考までに，WIDE2 の 22:00-22:59 の時間帯のトレースは他の時間帯と比べて際立った特徴を示していることが図 4.3と図 4.5からわかる．すなわち，他の時間帯よりも 5 倍近い量のパケットが出現しており，平均パケット長が 1/4〜1/5 程度の 100Byte 程度となっている．これは，明らかに通常の通信パターンとは異なる傾向であり，なんらかのアタックが行なわれていた可能性もある．CBNP の評価では，通常の通信パターンに加え，このような特殊なパターンでも期待通り高いスループットを実現できるかがひとつのポイントとなる．

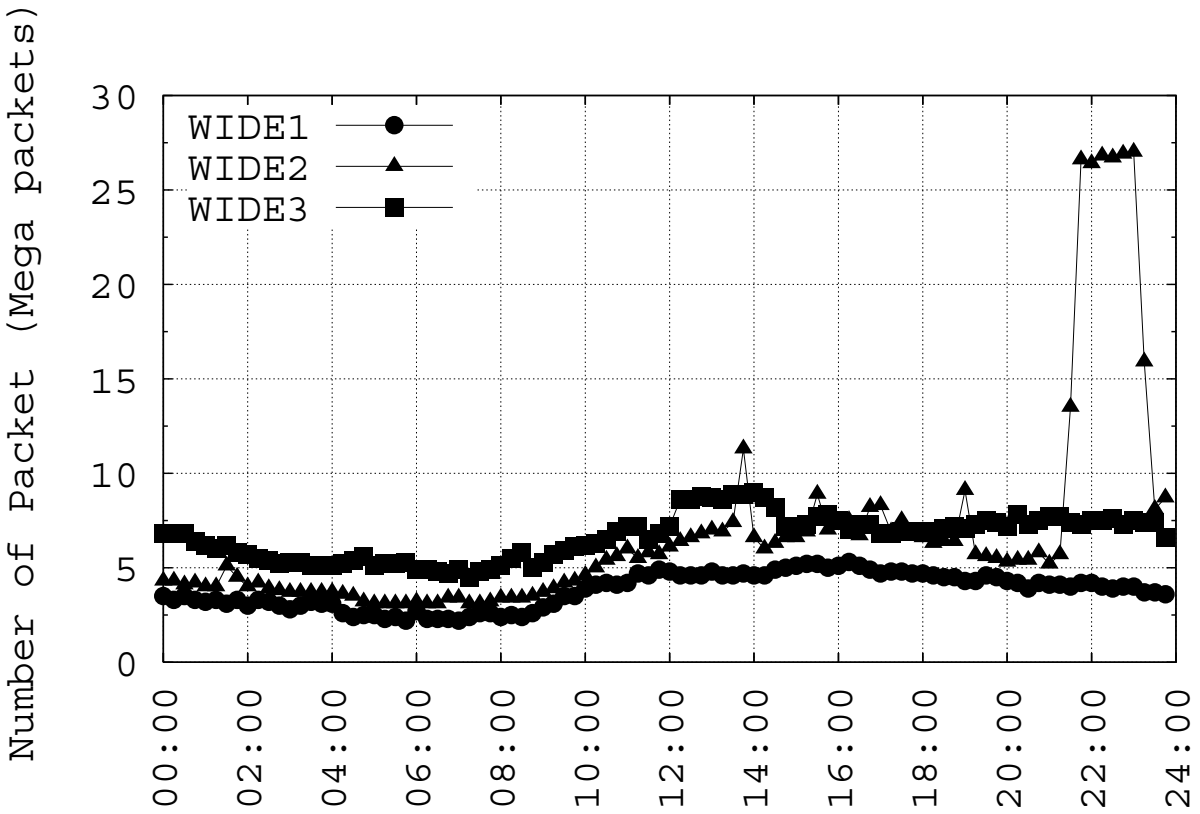

図 4.3: WIDE トレースのパケット数

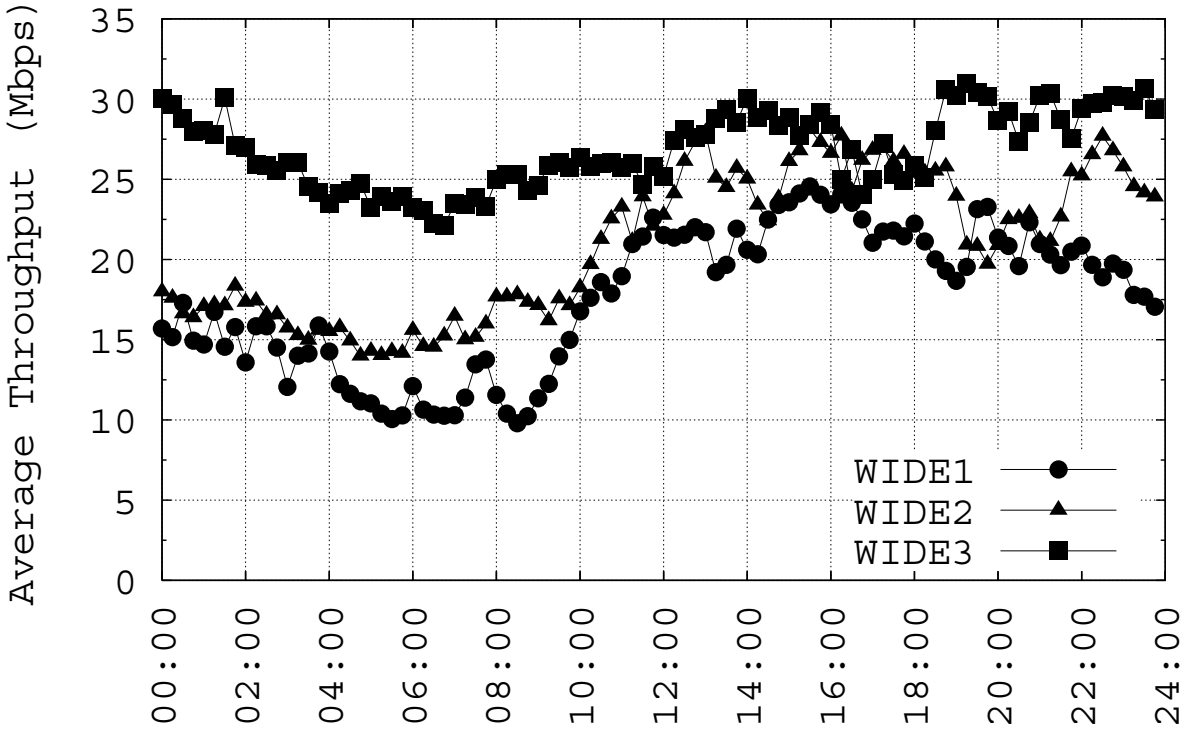

図 4.4: WIDE トレースの平均スループット (Mbps)

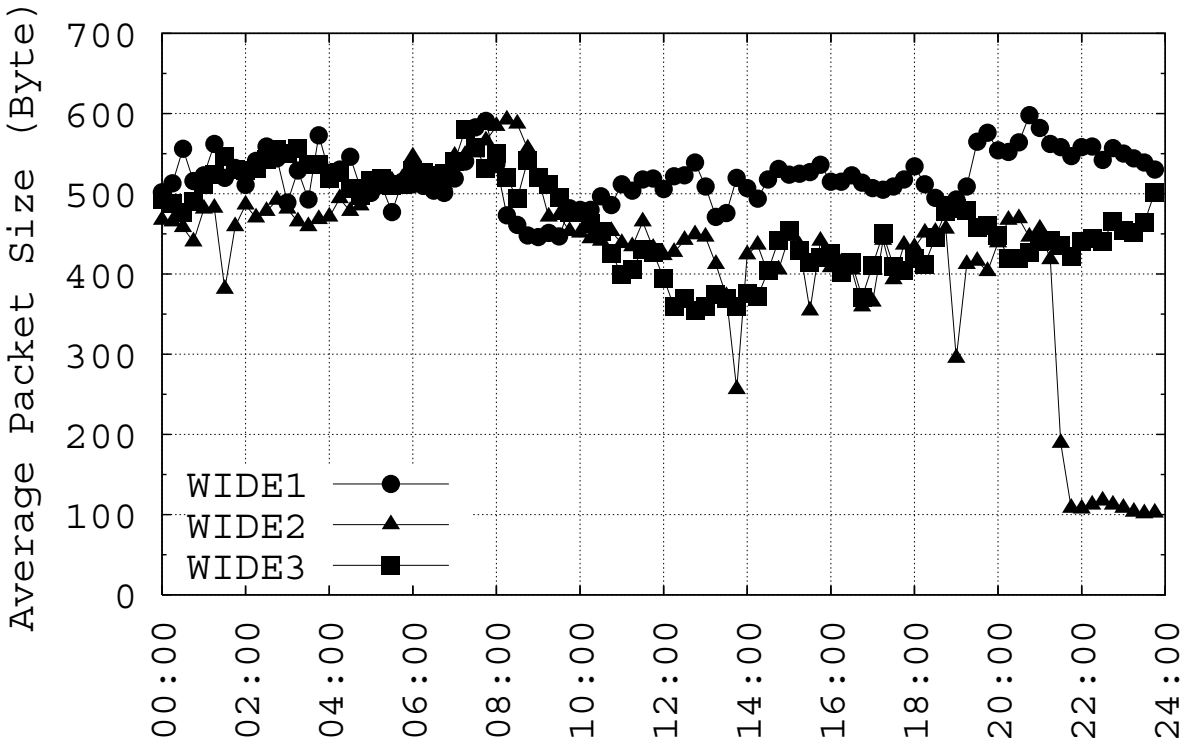

図 4.5: WIDE トレースの平均パケット長 (Byte)

4.4 評価結果

CBNP を表 4.7に示す 4 条件下で評価を行なった．それぞれの評価結果と考察についてそれぞれ述べる．

表 4.7: CBNP 評価条件

項番	パケット長	アグリゲーション	コメント
#1	最小長	なし	負荷最大のワイヤレート状態における評価
#2	実パケット長	なし	実環境下でのワイヤレートにおける評価
#3	最小長	あり (2 回線)	#1 において局所性を弱めた評価
#4	実パケット長	あり (2 回線)	#2 において局所性を弱めた評価

4.4.1 最小パケット長での評価

まず，表 4.7中の項番#1 に示す，最小パケット長 (64Byte)，アグリゲーションなしの状態，すなわち，負荷最大のワイヤレート状態における評価結果を図 4.6に示す．左側のグラフの横軸は，回線速度に対する P-Engine (従来型ネットワークプロセッサの場合は集積する全 PE) のスループット比を示す．縦軸は，P-Engine のスループットに対する CBNP のキャッシュヒット率 (PLC ヒット率と CMT ヒット率の和) を黒塗りのポイントで，PLC ヒット率を白抜のポイントで示す．また，右側のグラフの横軸は，左側のグラフと同じである．縦軸は，P-Engine のスループットに対する CBNP のスループットを黒塗りのポイントで示す．また，白抜のポイントで，PLC 及び CMH を動作させず P-Engine のみによるパケット処理のスループット (凡例の NO PLC/CMH)，すなわち，通常のネットワークプロセッサをエミュレートした場合のスループットを示す．上段のグラフは 06:00-06:59，中段のグラフは 15:00-15:59，下段のグラフは 22:00-22:59 のトレースの結果をそれぞれ示す．

WIDE2 の 22:00-22:59 のトレースを除くと，いずれのトレースも同様の傾向を示した．P-Engine のスループットが 10%〜30%と低い領域では，CBNP は従来型のネットワークプロセッサよりは 2.5 倍以上高いスループットを実現できている．しかしながら，100%のスループットを達成するまでには至っていない (100% 未達分のパケットは A-Engine で廃棄される)．キャッシュヒット率もこの領域では安定せず低い値となっている．これは，入力したパケットのうち A-Engine で廃棄されたパケットも PLC ミスを起こしたパケット (トークン) としてキャッシュヒット率の計算をしているためである．キャッシュヒット率の内訳の大部分は PLC によるものであり，CMH (の CMT) は最大 5%程度を占めるのみであった．しかしながら，CMH (の CMT (Cache Miss Table) や CMQ (Cache Miss Queue)) の存在がスループットに与える影響はキャッシュヒット率として観測される値以上に大きいと考えられる．なぜなら，CMH の CMT や CMQ が存在することにより，PLC ミスが発生している最中でも後続のパケットのトークンを停止させずに PLC にアクセスさせることができるためである．

100%のスループットを達成するには，従来型ネットワークプロセッサでは，回線速度と同じスループットを備える P-Engine，すなわち，PE 群が必要である．一方で，CBNP は，キャッシュミス率を補うだけのスループットを備えた P-Engine を備えれば良い．例えば，今回の評価で最もキャッシュヒット率の低かったのは WIDE3 のトレースであり，そのキャッシュヒット率は 65.5%程度である．この場合，従来型ネットワークプロセッサであれば 100%のスループットの P-Engine (PE

群) が必要であるが，CBNP は 34.5%以上のスループットを持つ P-Engine を備えれば良い．このため，図 4.6からもわかるように，40% のスループットを持つ P-Engine を備えている CBNP の場合に 100%のスループットを達成できている．100Gbps のスループットを想定すると，この場合，P-Engine は 40Gbps のスループットが必要であるが，この値は第 2 章の従来例で示したように現在の半導体技術で実現できている値である．

また，他のトレースとは異なる傾向を示した WIDE2 の 22:00-22:59 のトレースは，95.3%という非常に高いキャッシュヒット率を示している．これは，非常に高い時間的局所性が存在していたことを意味しており，CBNP は BSP に対してわずか 10%のスループットを持つ P-Engine を備えるだけで 100%のスループットを達成できる．このトレースは，意図的になんらかのアタックが行なわれていた可能性があるが，今回のように十分な時間的局所性を備えていれば，CBNP では，アタックパケット以外の通常のパケットもアタックパケットに邪魔されること無く処理することができる利点を持つと考えられる．また，高いキャッシュヒット率，すなわち，強い時間的局所性は WIDE のようなコア網より少ない数のパケットフローが流れるアクセス網で出現しやい傾向にあるため，CBNP のアーキテクチャは，アクセス網のルータ装置に搭載する場合，より効果的であると考えられる．

4.4.2 実パケット長での評価

次に，表 4.7中の項番#2 に示す，実パケット長 (64Byte〜1518Byte)，アグリゲーションなしの状態，すなわち，実環境下における負荷最大のワイヤレート状態の評価結果を図 4.7に示す．各グラフの各軸やプロットの意味は項番#1 のグラフと同じである．

図 4.7の結果から，CBNP は BSP に対してわずか 10%のスループットを持つ P-Engine を備えるだけで 100%のスループットを達成できることがわかる．これは，回線速度に対して 10%のスループットを持つ従来型ネットワークプロセッサの 2.85 倍近いスループットを CBNP が持つことを意味する．実パケット長の場合，図 4.5から明らかなように，平均パケット長は 400Byte〜500Byte 程度である．このため，パケットの到着間隔が最小パケット長 (64Byte) の場合に比べて 6 倍〜8 倍程度に長くなったため，P-Engine に要求されるスループットが 1/6〜1/8 程度になり，CBNP は BSP に対してわずか 10%のスループットを持つ P-Engine を備えるだけで 100%のスループットを達成できたと考えることができる．

また，他のトレースとは異なる傾向を示した WIDE2 の 22:00-22:59 のトレースでは，興味深い結果が得られている．このトレースが含むパケット長は最小パケット長に十分近い値であるため，通常のネットワークプロセッサの場合は，ほぼ P-Engine のスループットそのままの総スループットしか得られていない．すなわち，100%のスループットを通常のネットワークプロセッサで得る場合には，100%分の P-Engine が必要になると考えられる．一方で，CBNP の場合，他のトレースと同様に 10%のスループットを持つ P-Engine を備えれば 100%のスループットを達成できている．これは WIDE2 の 22:00-22:59 のトレースが 95% もの高いキャッシュヒット率 (時間的局所性) を備えているためであり，CBNP の特性を十分に発揮できた例であると言える．

4.4.3 2 回線アグリゲーションしたパケットトレースでの評価

将来のネットワークトラフィックパターンを予測するのは困難であるが，特性そのものは現在と傾向はかわらず，ネットワークサービス数やユーザ数が増えて，パケットフロー数が現在より

も増加する可能性が高いと考える．そこで，最後の評価として，表 4.7中の項番#3 と#4 の最小パケット長 (64Byte) 及び実パケット長 (64Byte〜1518Byte) におけるアグリゲーションありの状態の評価結果を図 4.8と図 4.9にそれぞれ示す．各グラフの各軸やプロットの意味は項番#1 のグラフと同じである．尚，X:00-X:29 のトレースと X:30-X:59 のトレース (X ＝ 6，15，22) を FPGA テストベッドの二つの Gigabit Ethernet 回線からそれぞれ同時に送信し，FPGA1 に実装した IFA により，数パケット単位でマージすることで 2 回線のアグリゲーションを実現した．

CBNP のスループットが最低となったのは，図 4.8に示すパケット長を最小化した WIDE3 トレースの場合であり，BSP に対し 40%のスループットを持つ P-Engine を備えた場合でもキャッシュヒット率は 57%程度と，図 4.6の場合に比べて 13.2%低下している．このため，CBNP で 100%のスループットを達成するためには，43%以上 (例えば 50%) の P-Engine を集積する必要がある．ただし，図 4.9からは，実パケット長であれば BSP に対して 10%のスループットの P-Engine であっても，CBNP で 100%のスループットを達成できる結果が得られている．

アグリゲーションすることによって，ネットワークトラフィックの時間的局所性が弱められることはわかるが，近い将来においても，このネットワークトラフィックの時間的局所性の存在が皆無になることは考えにくい．また，最小パケット長のパケットだけが出現することも，ユーザがある程度の大きさのデータを送受信するためにネットワークを利用することを考慮すると考えにくい．このため，これまでの評価を元に考えると，CBNP は，回線速度に対して 40%程度のスループットを持つ P-Engine を集積することで，十分なヘッドルームを持ちつつネットワークトラフィックを効率的に扱うことが可能であると考えられる．また，100Gbps 回線時代にコア網での利用に難点が出るとしても，時間的局所性が強く存在しうるアクセス網では十分利用可能であると考えられる．

4.5 第 4 章のまとめ

FPGA テストベッド上に CBNP のプロトタイプを実装し，コア網である WIDE の実ネットワークトラフィックトレースを利用したワイヤレートにおける評価を行なった．尚，CBNP を高機能なルータ装置に利用することを想定し，単純な宛先アドレスによるルーティング処理ではなく，QoS を考慮した 5-tuple によるパケットフローの分類を行ない，ルーティング処理とした．

まず，ワイヤレート処理の中でも最大負荷となる最小パケット長の評価を行なった．この場合，CBNP は，従来型ネットワークプロセッサの概ね 2.5 倍高いスループットを実現でき，キャッシュミス率に相当する回線速度の 34.5%以上のスループットを備える P-Engine を集積することで 100%のスループットを実現できるという結果が得られた．

次に，実際のネットワークトラフィックをワイヤレート化した実パケット長 (平均 400Byte〜500Byte) の評価を行なった．この場合，パケットの到着間隔が最小パケット長 (64Byte) の場合に比べて 6 倍〜8 倍程度に長くなるため，P-Engine に要求されるスループットも 1/6〜1/8 になり，CBNP は最小パケット長の場合よりも少ない回線速度比 10%のスループットの P-Engine を集積するだけで回線速度対応のワイヤレート処理を実現できることがわかった．一方で，従来型のネットワークプロセッサは，回線速度に対し 30%のスループットを持てば実パケット長においては回線速度対応のワイヤレート処理を実現できるが，最小パケット長に近いパケットが多数出現する状態には対応できない．CBNP では，最小パケット長に近いパケットが多数出現しても，回線速度比 10%のスループットの P-Engine を集積するだけで 100%のスループットを実現できるという結果が得られた．

最後に，2 回線をアグリゲーションした評価からは，時間的局所性が弱められた結果，特に最小パケット長だけのワイヤレート処理の場合に CBNP の P-Engine に必要なスループットが増加してしまうことを確認できた．しかしながら，実パケット長の場合，CBNP は，回線速度比 10%の P-Engine を集積するだけで依然として 100%のスループットを実現でき，従来型ネットワークプロセッサよりも有利であるという結果が得られた．

これらの評価より，ネットワーク通信が強い時間的局所性を示しやすいことが確認できた．そして，CBNP はキャッシュミス率に相当するスループットを持つ P-Engine を集積することにより，処理負荷が高くなる最小長に近いパケットでも回線速度のワイヤレートパケット処理が可能になることを確認できた．また，実パケット長のパケット処理においても従来型ネットワークプロセッサより十分な処理のヘッドルームを備える点で CBNP の優位性が示された．また，C-Engine の PLC と CMH の両機構が期待通りに動作することを確認できた．

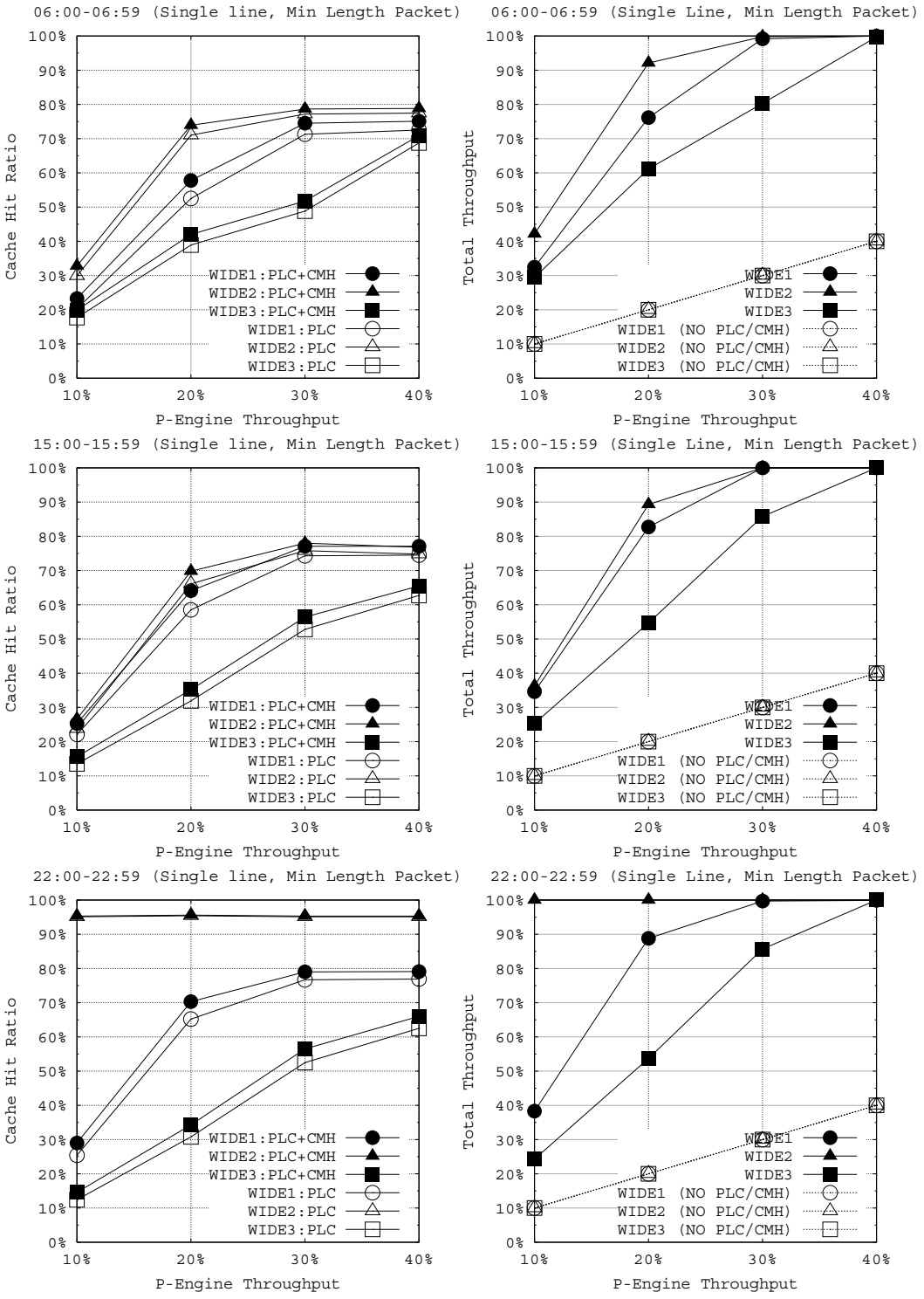

図 4.6: 最小パケット長ワイヤレート回線におけるキャッシュヒット率とスループットの関係

両グラフの横軸は回線速度に対する P-Engine (従来型 NP の場合は集積する全 PE) のスループット比．左側のグラフの縦軸の黒塗りポイントは P-Engine のスループットに対する PLC ヒット率と CMT ヒット率の和，白抜きポイントは PLC ヒット率．右側のグラフの縦軸は P-Engine のスループットに対する CBNP のスループット．尚，凡例の NO PLC/CMH は通常の NP をエミュレートした場合のスループット．上段は 6 時台，中段は 15 時台，下段は 22 時台のトレースの結果．

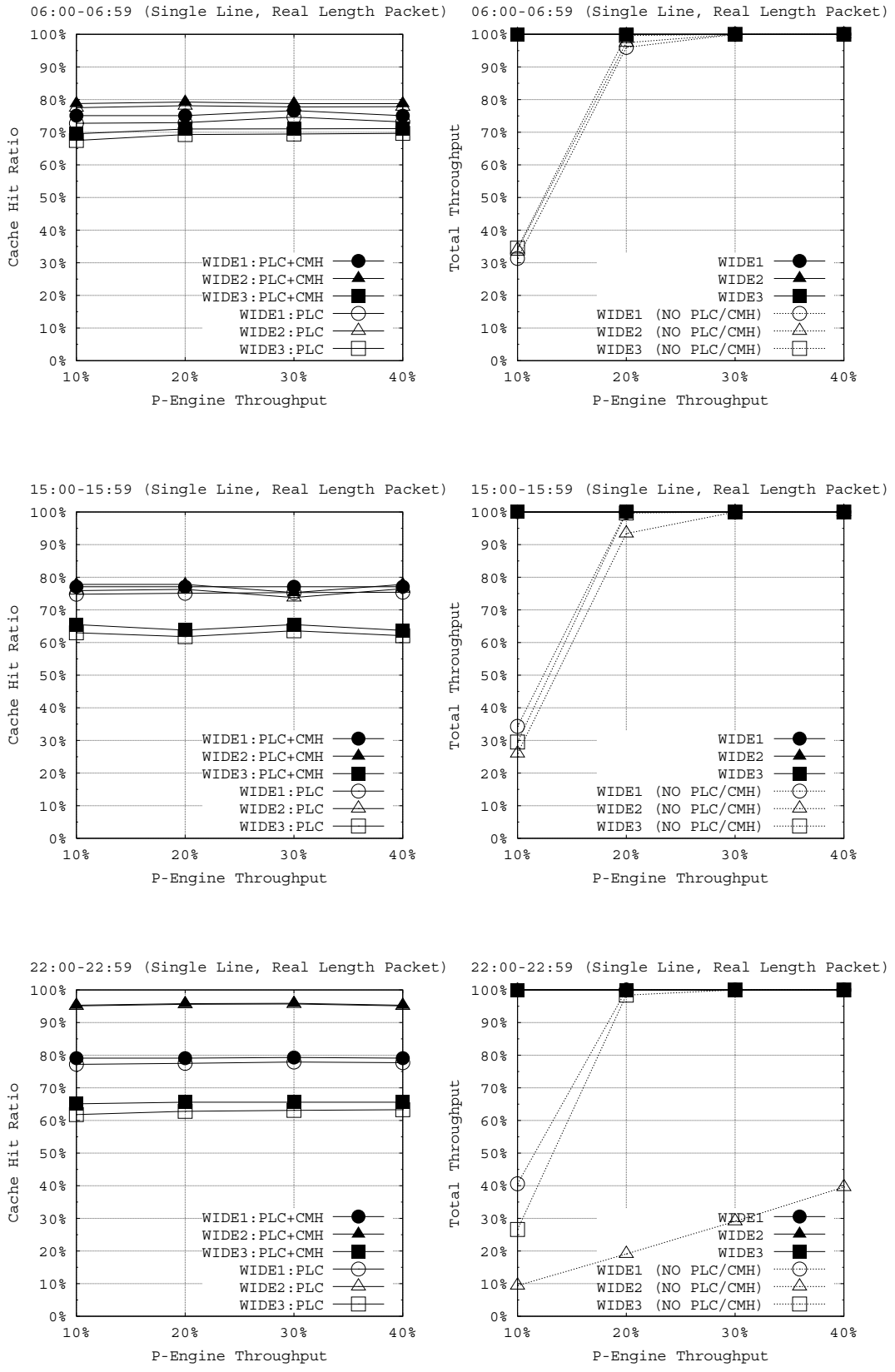

図 4.7: 実パケット長ワイヤレート回線におけるキャッシュヒット率とスループットの関係

グラフの読み方は図 4.6と同じ.

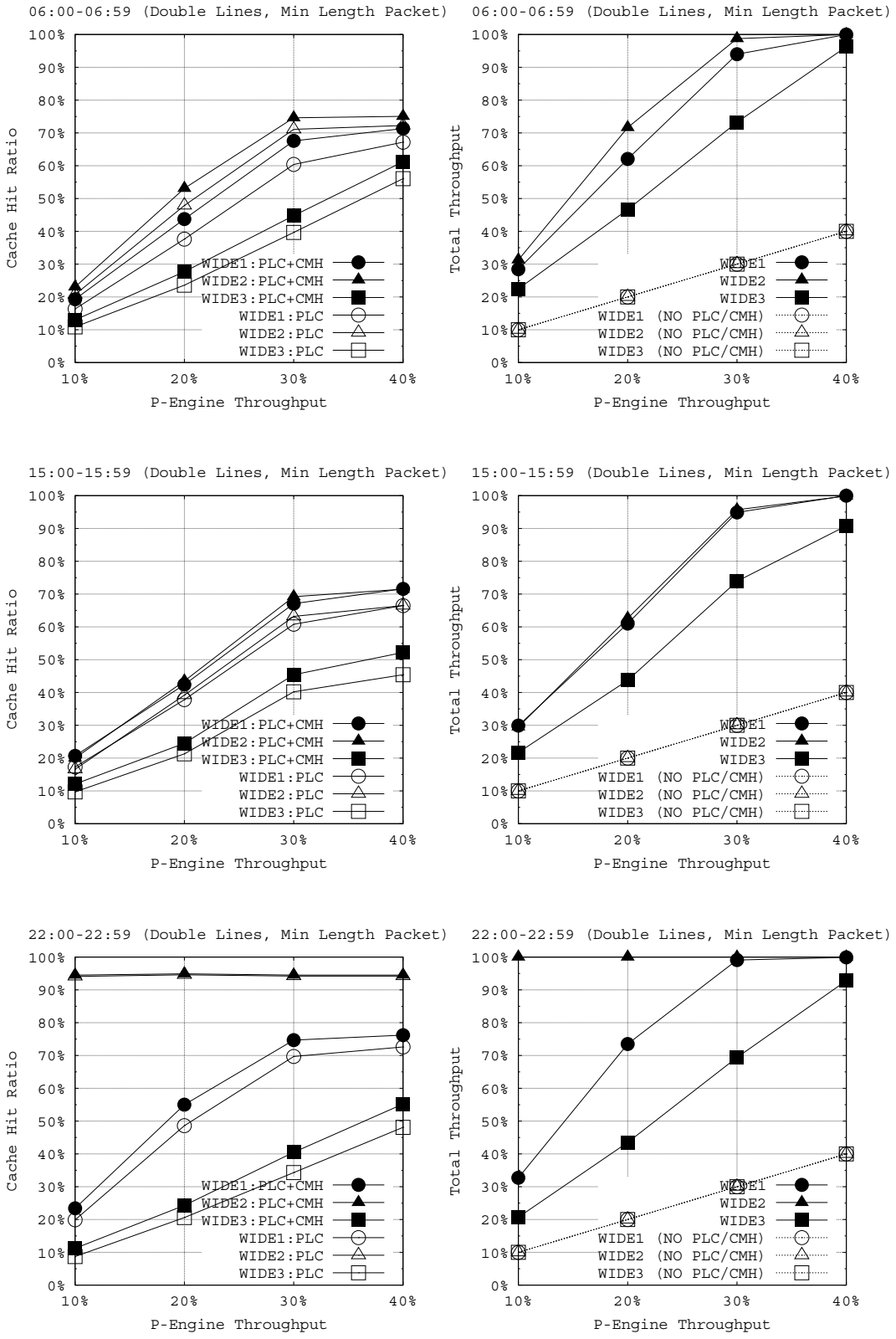

図 4.8: 2 回線アグリゲーショントレースで最小パケット長ワイヤレート回線におけるキャッシュヒット率とスループットの関係

グラフの読み方は図 4.6と同じ.

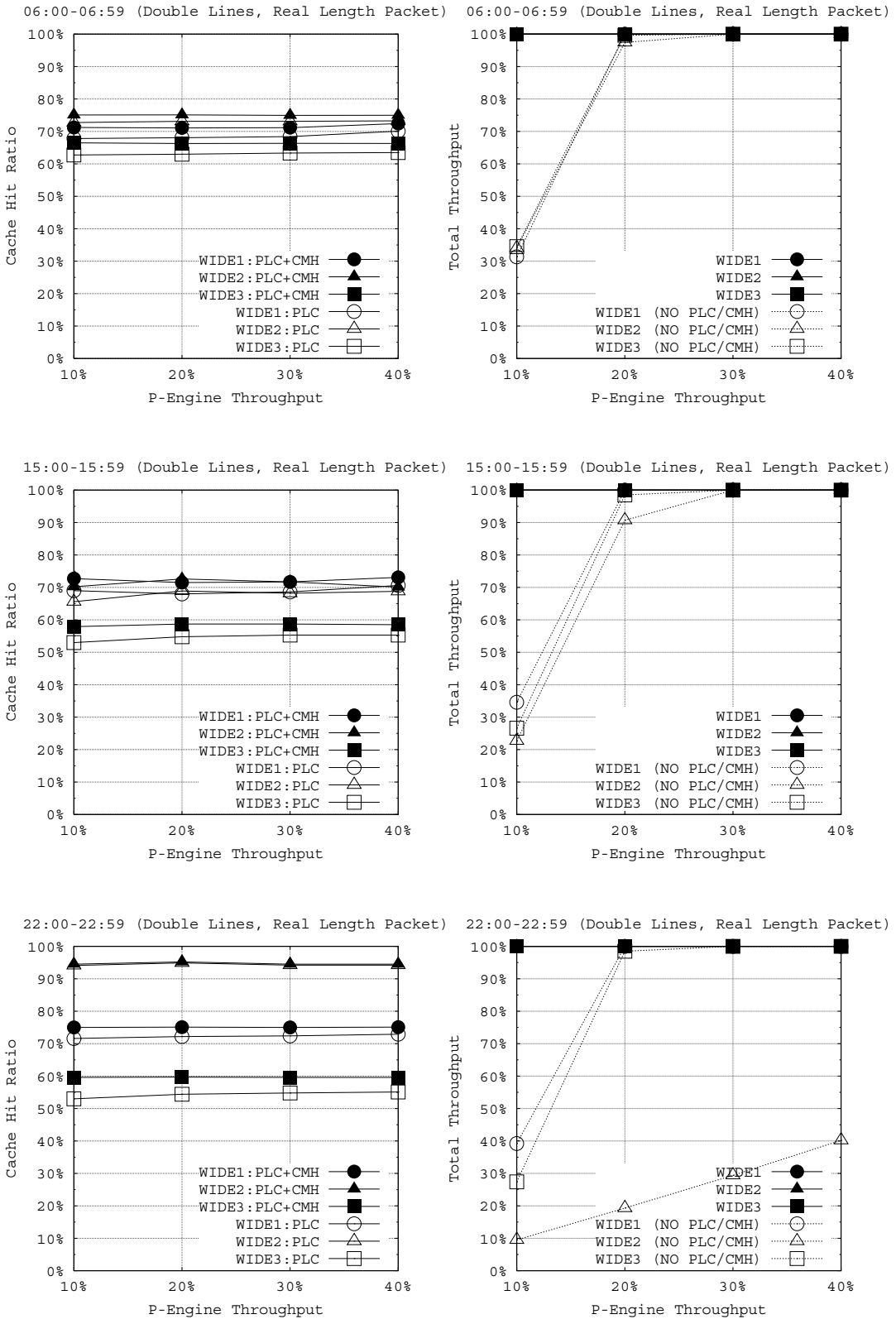

図 4.9: 2 回線アグリゲーショントレースで実パケット長ワイヤレート回線におけるキャッシュヒット率とスループットの関係

グラフの読み方は図 4.6と同じ.

第5章　CBNPと従来型ネットワークプロセッサのコスト比較

本章では，CBNPと従来型ネットワークプロセッサのコスト比較に関して考察する．コストは様々な定義が可能であり，また，様々な要因が絡み合うため絶対値としての評価は困難である．本論文中では，コストとしてダイサイズと消費電力を定義する．ダイサイズとは，半導体チップの面積であり，シリコンウェハから製造できるチップ数を決定する切断面の大きさのことを指す．ダイサイズが小さい程，1枚のシリコンウェハから製造できるチップ数が増加し，また，シリコンウェハ上の不純物などによる不良部分を含む確率が低下するため，歩留りが向上する．このため，ダイサイズはLSIの製造原価に大きな影響を及ぼす．すなわち，小さいダイサイズほど望ましく，製造原価，ひいては販売価格を低減させることが可能となる．また，消費電力とは，文字通り，半導体が消費する電力である．半導体の消費電力は，ほとんどが熱に変換されるので，消費電力が高いということは，発熱量が高いことを意味する．半導体デバイスの温度が各半導体のworst条件として規定された温度を超えると，トランジスタの動作を保証できなくなり，誤動作を引き起こす．また更なる発熱が続くと，ボードやそのほかの部品の過熱や変形，場合によっては溶融による致命的なシステムダウンを引き起こす．このため，半導体の温度がworst条件を超えないようにヒートシンクや空冷用のファン，水冷ジャケット等を利用した発熱対策がなされる．このようなシステム冷却対策によるコスト増加を削減するためにも，半導体の消費電力は小さい程良い．

本章の評価では，半二重100Gbps回線向けのNPと，半二重1Gbps〜100Gbps回線向けの様々なNPをターゲットとする．手順としては，まず，両ターゲットにおけるCBNPと従来型ネットワークプロセッサの両者について，論理ゲート数とメモリ量を見積る．次に，得られた値を元に，両者のコア部分のダイサイズと最大消費電力を推測する．得られた結果から，CBNPの従来型ネットワークプロセッサに対する優位性について考察する．

5.1　100Gbps回線NP向けのダイサイズと消費電力見積り

5.1.1　比較の前提条件

まず，半二重100Gbps回線向けのCBNPと従来型ネットワークプロセッサの両NPについて検討，考察を行なう．比較の前堤として，両NPを第2章で紹介した代表的なハイエンドプロセッサと同等の0.13μmのCMOS技術で動作周波数333MHz，電源電圧1.2Vとして見積りを行なう．両NPはパケット処理用のプログラマブルなPE(CBNPではP-Engineに相当)とオンチップのパケットメモリ，メインインタフェースを備えるもとのした．メインインタフェースとは，ここではネットワーク回線及び装置内のスイッチファブリックに接続するためのインタフェース論理と物理ポートを指す．CBNPは，これらに加えてBSP (Burst Stream Path，すなわち，A-Engine/C-Engine/R-Engine)を備える．

5.1.2 CBNP の BSP のゲート数とメモリ量

まず，第 4 章で設計した CBNP のプロトタイプを元に半二重 100Gbps 回線用の CBNP を再設計し，Hitachi 0.13μm 用の CMOS ライブラリを利用して Synopsys 社の論理合成ツールで論理合成した．再設計の際には，トークンを構成する各情報の bit 数は，U-Info[28 bit]，A-Info[20 bit]，E-Info[384 bit]，R-Info[384 bit]，P-Info[80 bit] とした．このため，A-Engine が生成する U/A/E-Info を含むトークンは 432bit，C-Engine で置換後の U/A/R/P-Info を含むトークンは 512bit となる．また，BSP のバス幅を 384bit とし 333MHz 動作時に 100Gbps 回線に十分な帯域を確保した．表 5.1に論理合成した BSP 各部のゲート数を示す．BSP 全体で約 1.1M ゲートとの見積り結果である．

表 5.1: BSP のゲート数

Block name	Sub block name	# of gates
A-Engine	L2, L3, and L4 Packet Analyzer	59K
	Error checker	71K
	8-stage Header Extractor	173K
	Others	21K
C-Engine	PLC path/control	188K
	CMH path/control	209K
	Others	49K
R-Engine	Packet Reader	10K
	Decoder, Patcher	171K
	Aligner	27K
	Packet Modifier	117K
	Others	12K
Total		1108K

また，BSP では PLC や CMH，トークンバッファや命令メモリ等にオンチップの SRAM を利用する．利用箇所により，1 ポートまたは 2 ポートの SRAM を使いわけた．表 5.2に BSP 各部のメモリ量を示す．PLC は，4 ウェイ・セットアソシアティブで総エントリ数 4096，リプレースアルゴリズムは擬似 LRU とした．CMH の CMT は，4 ウェイ・セットアソシアティブで総エントリ数 1024，各 CMT に対応する CMQ は深さ 32，EWQ と PRQ は共に深さ 256 とした．また，A-Engine と R-Engine のトークンバッファ，C-Engine の HTQ は深さはいずれも 128 とした．この他，A-Engine には 16bit 幅で 1024 エントリ及び 128 エントリの命令メモリを 8 個ずつ，R-Engine には 128bit 幅で 1024 エントリのメモリを 1 個用意した．単純積算すると BSP で利用するメモリ総量は，約 740KByte となる．

5.1.3 プロセッシングエレメントのゲート数とメモリ量

ネットワークプロセッサで利用する PE は，通常のプロセッサが備える MMU(Memory Management Unit) や FPU(Floating Point Unit) 等の複雑な機能ブロックは不要である．その代わりにパケット処理に適した論理演算，ビット列操作，分岐命令，ロード ストア命令等を備えていれば良い．その結果，各 PE は非常にシンプルな 32bit の RISC プロセッサとすることができる．ここでは，既存のネットワークプロセッサ設計事例から各 PE の論理ゲート数，メモリ量を推測する．

表 5.2: BSP で利用するオンチップ SRAM

Block	Use	Type	Bit	Word	No.	KBytes
A-Engine	Inst memory1	1 port RAM	16	1K	8	16.00
	Inst memory2	1 port RAM	16	128	8	2.00
	Token buffer	2 port RAM	432	128	1	6.75
C-Engine	PLC TAG	1 port RAM	460	1K	4	230.00
	PLC LRU	1 port RAM	4	1K	1	0.50
	PLC Data	1 port RAM	522	1K	4	261.00
	CMH CMT	2 port RAM	405	256	4	50.63
	CMH CMQ	2 port RAM	28	32	1024	112.00
	CMH EWQ	2 port RAM	460	256	1	14.38
	CMH PRQ	2 port RAM	464	256	1	14.50
	HTQ	2 port RAM	512	128	1	8.00
R-Engine	Inst memory	1 port RAM	128	1K	1	16.00
	Token buffer	2 port RAM	512	128	1	8.00

第 2 章で紹介した Cisco 社の半二重 10Gbps スループットのネットワークプロセッサ Toaster3 は，比較的シンプルな PE を 16 個備え，10.3M ゲートであるとされている [38]．このゲート数は約 400KByte 相当のオンチップ SRAM も含んでいるため，16 個の PE 部分の総論理ゲート数を約 5M ゲートと推測する．すなわち，1PE あたり 300K ゲートと推測し，これを本議論の PE の論理ゲート量とした．また，第 2 章で示した各社のネットワークプロセッサが備える PE のメモリ構成を参考に考え，ここでは各 PE は，1 ポート SRAM による命令メモリを 8KByte，2 ポート SRAM によるデータメモリを 2KByte 備えると仮定した．

ここで，半二重 100Gbps 回線を処理するために必要な PE 数は，従来型のネットワークプロセッサの場合 160 個と考える ((16 個/10Gbps)×10 倍)．一方，CBNP では第 4 章の結果からキャッシュミス率分を補うために回線速度の 40%のスループットを持つ P-Engine を備えれば良いので，必要な PE 数は 64 個と考える ((16 個/10Gbps)×4 倍)．それぞれの場合の PE に関する総論理ゲート数とメモリ量の関係を表 5.3に示す．

表 5.3: PE 数と論理ゲート数及びメモリ量

	PE 数	総論理ゲート数	総命令メモリ量	総データメモリ量
従来型 NP(100Gbps 相当)	160	48 M ゲート	1280KByte	320KByte
CBNP 用 P-Engine(40Gbps 相当)	64	19.2M ゲート	512KByte	128KByte

5.1.4 コア部分のダイサイズ見積り

これまでの前提条件の元，0.13μmCMOS プロセスにおける半二重 100Gbps 回線向けの従来型ネットワークプロセッサと CBNP のコア部分のダイサイズの見積もりを行なった．SRAM の面積算出には，日立の 0.13μm の CMOS ライブラリの値を参考にした [65, 67]．尚，ここでコアとは，PE 部分と BSP 部分の総和を指す．ネットワークプロセッサ上でパケットを処理中に一時的にパ

ケット本体を確保しておくためのパケットメモリは，従来型ネットワークプロセッサでも CBNP でも共通に必要となるため，比較評価項目からは除外している．同様の理由により，外部のネットワークやスイッチファブリックとのインタフェース部分も比較評価項目からは除外している．表 5.4に見積もり結果を示す．

表 5.4: 0.13μm CMOS プロセスにおける 100Gbps NP コアのダイサイズ見積もり (単位: mm^2)

Block		CBNP with 40Gbps P-Engine (64 PEs)	従来型 NP (160 PEs)
BSP	Logic	11.08	–
	PLC/CMH memory	34.23	–
	Other memory	3.09	–
PE array	Logic	192.00	480.00
	Memory	28.30	70.76
Core Total		268.7	550.76

表 5.4より，CBNP の BSP は PE 部分と比較し，十分小さなサイズで実現が可能であることが見込まれる．この結果，CBNP のコア部分のダイサイズは 268.7mm^2，従来型ネットワークプロセッサのコア部分のダイサイズは 550.76mm^2 となり，CBNP は従来型ネットワークプロセッサの 48.8%のダイサイズで実現できる見込みである．

5.1.5 コア部分の最大消費電力見積り

これまでの前提条件の元，0.13μmCMOS プロセス，動作周波数 333MHz，電源電圧 1.2V における半二重 100Gbps 回線向けの従来型ネットワークプロセッサと CBNP のコア部分の最大消費電力を見積もった．確度の高い消費電力見積もりは非常に困難であるため，今回の見積もりでは TSMC 社の一般講演で利用された資料 [84] から，0.15μm プロセスにおいて 1.5V 電源では 9.1nW/MHz が基本消費電力であることを参考にした．消費電力は，第 2 章で示した 2.1式からもわかるように電源電圧の 2 乗に比例するため，1.2V 電源の場合，5.82nW/MHz になると考え，この値を 0.13μm プロセスにおける 1.2V 電源の消費電力と仮定した．また，SRAM の最大消費電力は日立の 0.13μm の SRAM ライブラリの値を参考とした．これらの仮定に基づいた最大消費電力の見積もり結果を表 5.5に示す．

CBNP の BSP は PE 部分と比較し，十分小さな消費電力で実現が可能であることが見込まれる．この結果，CBNP のコア部分の最大消費電力は 50.65W，従来型ネットワークプロセッサのコア部分の最大消費電力は 113.3W となり，CBNP は従来型ネットワークプロセッサの 44.7%の消費電力で実現できる見込みである．CBNP では，BSP で消費することが見込まれるのは 5.33W であり，これはコア全体の消費電力のうち 10.5%でしかない．このため，CBNP は，従来型ネットワークプロセッサに比べて電力効率の非常に高いアーキテクチャであることがわかる．

尚，本論文で示したの消費電力は第 2 章の 10Gbps 扱のハイエンドネットワークプロセッサと比較すると，やや大きめの値となっている．これは本論文で想定した論理の活性化率が大きかったこと，また，本論文ではクロックゲーティング等の低消費電力化技法を考慮していないことが主な要因と考えられる．しかしながら，これらを考慮しても，従来型ネットワークプロセッサと

表 5.5: 0.13μm CMOS，333MHz，1.2V，100Gbps NP コアの最大消費電力見積もり (単位: W)

Block		CBNP with 40Gbps P-Engine (64 PEs)	従来型 NP (160 PEs)
BSP	Logic	2.20	–
	PLC/CMH memory	2.50	–
	Other memory	0.63	–
PE array	Logic	38.13	95.32
	Memory	7.19	17.98
Core Total		50.65	113.3

CBNP の消費電力の比率自体は変わらず，CBNP は，従来型ネットワークプロセッサに比べて電力効率の非常に高いアーキテクチャであると考えられる．

5.2 様々な回線速度向けのダイサイズと消費電力見積り

第 5.1 節では，100Gbps 回線向けのダイサイズと最大消費電力を見積もった．本節では，様々な回線速度において同様の比較を行ない，CBNP が優位な範囲に関して考察する．

比較する回線速度としては，半二重 1Gbps，2.5Gbps，10Gbps，20Gbps，40Gbps，80Gbps，100Gbps の 7 種類とした．CBNP は，第 5.1 節での検討同様，回線速度に対して 40%に相当する P-Engine を集積することとした．また，CBNP の BSP は 100Gbps 回線向けと同一の論理を利用し，動作周波数だけを回線速度に応じて低下させることとした．基本動作周波数を 333MHz としているため，例えば，10Gbps 回線向けの場合，33MHz とした．プログラマブルなパケット処理を行なう PE 部分は，CBNP と従来型 NP いずれも動作周波数は 333MHz のままとし，代わりに PE 数を必要最低限の数に削減することとした．基本 PE 数は，従来型 NP の場合，100Gbps 回線相当で 160 個としたため，例えば，10Gbps 回線向けの場合，16 個とした．PE 数に端数が出る場合は，切り上げた個数とした．以上の前堤の元，様々な回線速度向けの NP のコア部分のダイサイズと最大消費電力の見積もり結果を表 5.6に示す．また，回線速度 (スループット) とコア部分のダイサイズとの関係を図 5.1に，回線速度 (スループット) とコア部分の最大消費電力との関係を図 5.2示す．

表 5.6: 様々な回線速度向けの NP のダイサイズと最大消費電力見積もり

スループット (Gbps)	CBNP(回線速度比 40%の P-Engine 集積)				従来型 NP		
	PE 数	BSP 周波数	ダイサイズ	消費電力	PE 数	ダイサイズ	消費電力
1	1	3.3 MHz	51.8 mm^2	0.76 W	2	6.88 mm^2	1.42 W
2.5	2	8.3 MHz	55.3 mm^2	1.55 W	4	13.8 mm^2	2.83 W
10	7	33 MHz	72.5 mm^2	5.49 W	16	55.1 mm^2	11.33 W
20	13	66 MHz	93.2 mm^2	10.26 W	32	110.2 mm^2	22.66 W
40	26	133 MHz	137.9 mm^2	20.54 W	64	220.3 mm^2	45.32 W
80	52	266 MHz	227.4 mm^2	41.08 W	128	440.6 mm^2	90.64 W
100	64	333 MHz	268.7 mm^2	50.65 W	160	550.8 mm^2	113.3 W

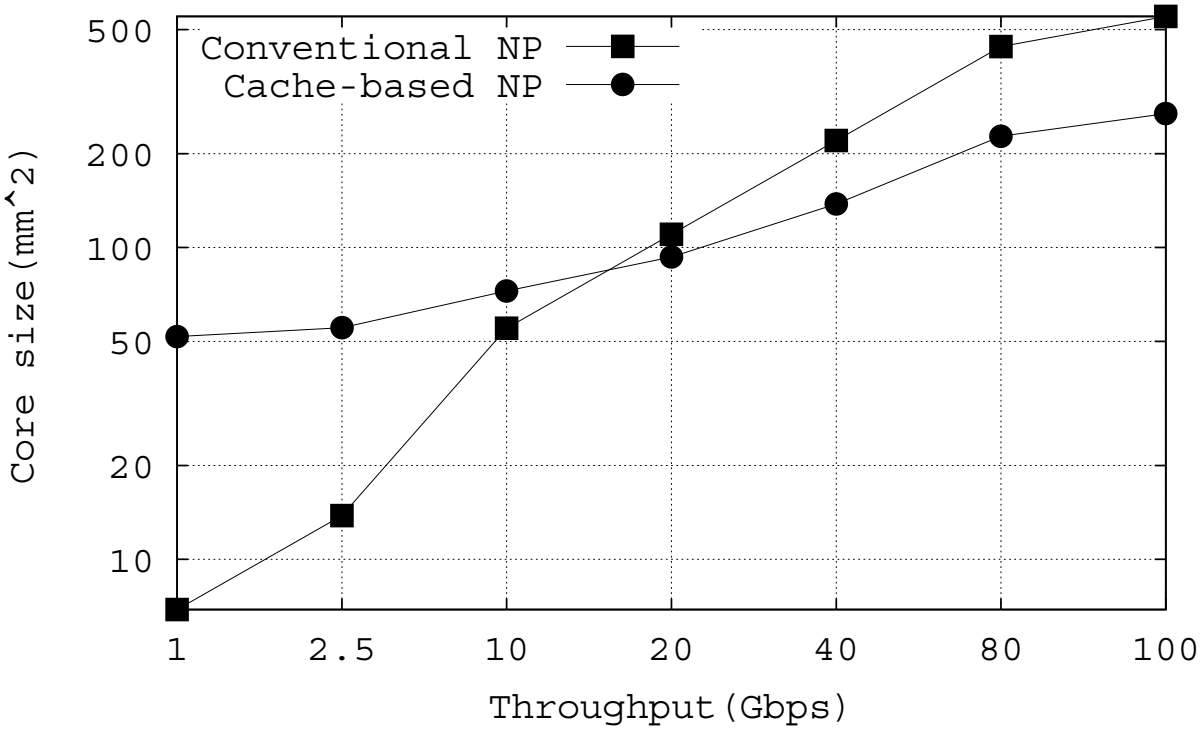

図 5.1: 回線速度 (スループット) に対するコア部分のダイサイズ

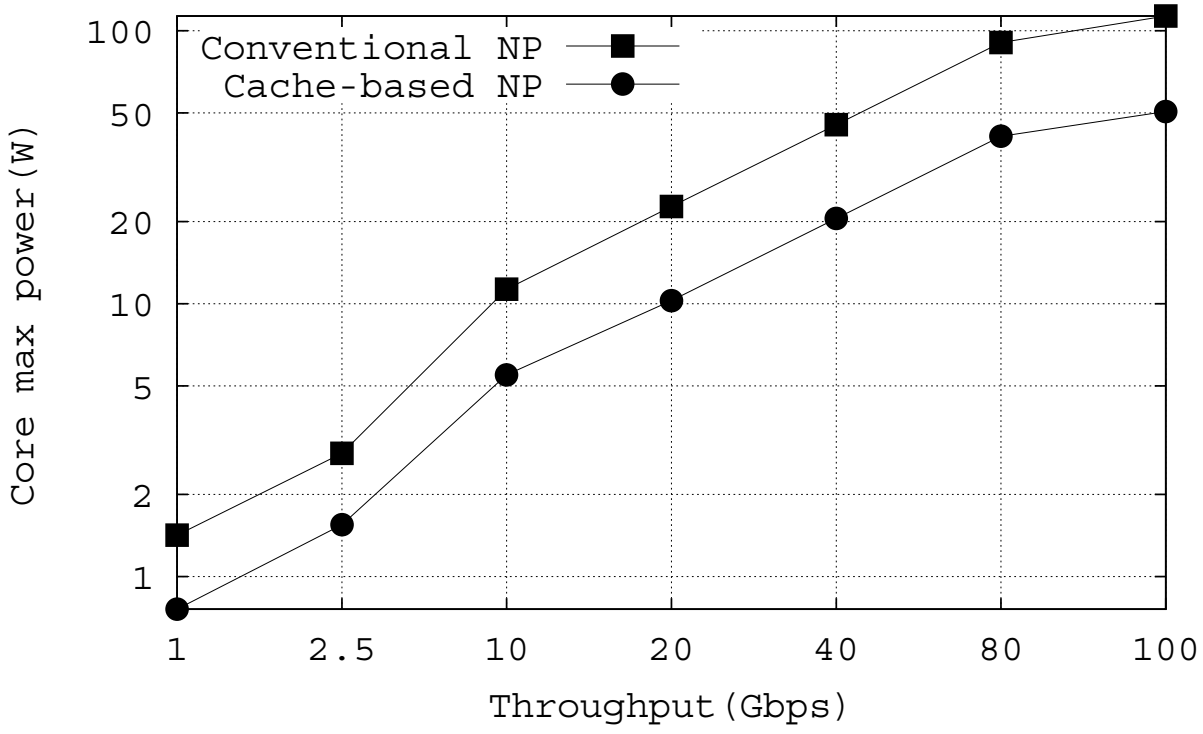

図 5.2: 回線速度 (スループット) に対するコア部分の最大消費電力

表 5.6，図 5.1より，20Gbps 以上の回線速度をターゲットとする場合，CBNP は，従来型 NP よりもコア部分のダイサイズを小さくできることがわかる．10Gbps 回線以下の領域では，CBNP のダイサイズが従来型 NP よりも大きくなっているが，10Gbps の場合でも 72.5mm^2 と，LSI の歩留まりを悪化させるような巨大なサイズではないため，大きな問題になることはないと考えられる．また，消費電力の観点から表 5.6，図 5.2を見ると，CBNP は，全回線速度領域において従来型 NP の 45%〜55%程度の消費電力で動作可能な見込みである．

5.3 第 5 章のまとめ

本章では，CBNP と従来型ネットワークプロセッサの両者について，論理ゲート数とメモリ量を見積り，両者のコア部分のダイサイズと最大消費電力を比較した．尚，第 4 章の評価結果に基づき，CBNP が集積する P-Engine は回線速度比 40%のスループットを備えるものとした．

その結果，CBNP のコア部分 (BSP と P-Engine) は，従来型ネットワークプロセッサのコア部分と比較して，ダイサイズは 48.8%，最大消費電力は 44.7%で実現できる見込みである．更に，CBNP では，BSP で消費することが見込まれるのは 5.33W であり，これはコア全体の消費電力のうち 10.5%でしかない．このため，CBNP は，従来型ネットワークプロセッサに比べて電力効率の非常に高いアーキテクチャであると言える．

また，各種の回線速度に対応する両アーキテクチャのコア部分のダイサイズと消費電力を比較した．20Gbps 以上の回線速度をターゲットとする場合，CBNP は，従来型 NP よりもコア部分のダイサイズを小さくできることがわかる．10Gbps 回線以下の領域では，CBNP のダイサイズが従来型 NP よりも大きくなっているが，10Gbps の場合でも 72.5mm^2 と，LSI の歩留まりを悪化させるような巨大なサイズではないため，大きな問題になることはないと考えられる．また，CBNP は，全回線速度領域において従来型 NP の 45%〜55%程度の消費電力で動作可能な見込みである．以上より，CBNP は従来型ネットワークプロセッサに対し，ダイサイズ，消費電力の両面で優位であると結論づけられる．

第6章　結論

本研究では，短時間のうちに同一のヘッダ情報を持つパケットがネットワーク上に多数出現しやすいという通信の時間的局所性を利用し，パケット処理の結果と処理結果の適用方法を併せて記録し，後続の同一フローパケットに適用するネットワークプロセッサアーキテクチャCache-based Network Processor (CBNP) を提案した．そして，CBNP は，多数のプロセッシングエレメント (PE) を集積する従来型のネットワークプロセッサに比べて半分未満の面積と消費電力でパケット処理が可能な効率の良いアーキテクチャであることを明らかにした．

従来型のネットワークプロセッサでは，一般に，内蔵する PE 数を増加させ，パケット処理の並列度を上げることにより広帯域化 (高スループット化) を実現してきた．この手法は，0.13μm の半導体プロセスまでは半導体のスケーリング則の恩恵により比較的スムーズに消費電力を大幅に上げること無く広帯域化を実現できた．これは，ゲートリーク電流をほとんど無視できたため，動的な消費電力の削減のみに注力すればよかったためである．ところが，0.13μm 以下の微細な半導体プロセスでは，リーク電流による静的な消費電力が無視できなくなってきている．すなわち，消費電力面で半導体のスケーリング則が成立しなくなりつつある．このため，従来手法の延長線上で PE 数を増加させてスループットを向上させる手法は，消費電力効率の面で問題が生じうる．

本研究で提案した CBNP は，ネットワーク通信に存在する時間的局所性を利用し，PE 数を増加させずにパケット処理の広帯域化を図るアーキテクチャである．CBNP はパイプライン処理型のネットワークプロセッサに分類できる．CBNP のパイプライン部分を Burst Stream Path (BSP) と呼び，BSP は回線速度対応の高速処理を実現するために，必要最小限のプログラマビリティのみを備える．回線速度達成の妨げとなる複雑な処理，すなわち，経路検索等の各種テーブルルックアップや新規ヘッダ生成等の処理は BSP と並列動作するプログラマブルな PE 群を集積した P-Engine と呼ぶ機能ブロックで実施する．BSP はパケットを解析しトークンと呼ぶ内部情報列を生成する A-Engine，PLC を参照してトークンに処理結果と処理結果適用方法を付与 (場合によっては置換) する C-Engine，トークンの内容を適用する R-Engine によって構成する．

CBNP は，P-Engine での処理結果と結果の適用方法 (Process) を併せて記録する Process Learning Cache (PLC) を C-Engine に備える．そして，PLC 登録済みパケットと同一とみなせる後続のトークン (パケット) には PE を利用せず，PLC の内容を適用することで高スループット化を実現する．また，Cache Miss Handler (CMH) と呼ぶ機構により，PLC 登録が完了するまでに発生する同一のキャッシミス (PLC ミス) を PE に割り当てず次々と取り込み，PLC 登録の際に取り込んでおいたトークンに PLC 登録内容を順次適用する．すなわち，CMH は消費電力の高い PE の利用を抑止し，尚且つ，PLC をノンブロッキング化する仕組みを提供する．CBNP は，PLC と CMH でキャッシュミスしたトークン (パケット) の処理に必要なスループットの P-Engine を集積することで，回線速度対応のパケット処理を実現できる．

本論文の第 3 章では，提案した CBNP アーキテクチャのうち PLC と CMH に関する予備評価をソフトウェアシミュレータを構築して行ない，PLC と CMH の基本パラメータに関して考察した．第 4 章では第 3 章で得られた考察結果を利用し，FPGA で構成したテストベッドを用いて CBNP プロトタイプを構築して CBNP の有効性を検証した．この CBNP プロトタイプは，4K エントリ，4 ウェイ・セットアソシアティブ，CRC ハッシングの PLC と，1K エントリ，4 ウェイ・セットアソシアティブ及び 8 エントリのフルアソシアティブの CMT (Cache Miss Table)，各 CMT エントリに対応する 16 エントリの CMQ (Cache Miss Queue) を備え，P-Engine のスループットを BSP に対して 10%〜40%と変化させて評価した．この評価では，実際のハイエンドルータでのパケット転送処理を想定し，5-tuple (宛先/送信元 IP アドレス，プロトコル，宛先/送信元ポート番号) によるクラシフィケーションでフローを分類し，コア網である WIDE の実ネットワークトレースを利用してキャッシュヒット率 (PLC と CMH のヒット率の総和) と CBNP のスループットを調査した．尚，最も厳しいワイヤレート下で CBNP を評価するために，WIDE のネットワークトレースのパケット間隔を最小化したり，パケット長を最小化したり，将来のネットワークを想定して 2 回線をマージ (アグリゲーション) してフロー数を増やしたりした．

実パケット長のままのワイヤレート評価では，多数のフローが出現するために通信の時間的局所性が弱まりやすいコア網であっても，回線速度比わずか 10% の P-Engine を集積するだけで，CBNP は回線速度対応のパケット転送処理が可能であった．また，2 回線をアグリゲーションし，尚且つ，パケット長を最小化した非常に厳しい条件下でも，回線速度比 40%相当の P-Engine の集積することでほとんどの場合，CBNP は回線速度対応のパケット転送処理が可能であった．将来のネットワークトラフィックの傾向を予測することは困難であるが，通信のフロー数が爆発的に増加することはあっても，最小パケット長だけが存在する通信となることは考えにくい．このため CBNP は，現在の技術で実現可能な 10Gbps〜40Gbps 程度のを集積することで 2010 年代に要求される半二重 100Gbps 回線のパケット転送処理に十分対応できる見通しである．

これまでサーバやストレージの分野では，電力問題が話題に挙がることがたびたびあったが，最近ではネットワーク機器が深刻な電力不足をもたらすという懸念が急浮上している．近年，ISP (Internet Service Provider)が処理するネットワークトラフィック量は第 2 章でも説明したように大幅に増加し続けており，ルータやスイッチの消費電力が増え続けている．このまま増加すれば，2010 年にルータやスイッチが消費する電力は 2000 年の 2 倍以上に達するという試算がある．例えば三井情報開発株式会社総合研究所が 2002 年に行なった予測では，省エネ対策を施さない場合，2000 年に 349 億 kWh であった消費電力が 2010 年には 450 億 kWh も増加し，2 倍以上の 799 億 kWh となっている [85]．消費電力増加に伴い空調設備の増強が必要であり，ネットワーク機器やネットワーク機器室の維持費も増加してしまう．このため，今後のネットワーク機器は大容量高速転送や高機能化の実現だけでなく，低消費電力化が強く要求される可能性が高い．

本論文の第 5 章では，CBNP が集積する P-Engine を回線速度の 40%分と仮定して，従来型ネットワークプロセッサとの面積比及び消費電力比を見積もった．その結果，CBNP のコア部分 (BSP と P-Engine) は，従来型ネットワークプロセッサのコア部分と比較し，48.8%の面積，44.7%の最大消費電力で実現できる見込みである．このため，CBNP は従来型ネットワークプロセッサに対し，ダイサイズ，消費電力の両面で優位である．また，CBNP アーキテクチャは，次世代の 100Gbps 級回線以外，すなわち，現在既に利用されている 10Gbps 回線や 40Gbps 回線の処理の置き換えにも有効である可能性を示した．特に，同一ヘッダを持つとみなせるパケットが短時間に大量に出現するという通信の時間的局所性が強く現われる場合，CBNP は非常に有効に機能することが確認できている．パケットフロー数が相対的に少なく，時間的局所性が強く現われやすいアクセス

網，またはコア網の中でもアクセス網に近い部分におけるでは，CBNP の方が低消費電力で所望のスループットを実現できる見込みであり，消費電力削減にも効果的であると考えられる．また，CBNP の PLC や CMH (の CMT) で高いキャッシュヒット率を得られる場合，CBNP の PE 群集積部である P-Engine の利用頻度が下がる．場合によっては，P-Engine 内の未使用 PE のクロックを停止することで，更なる電力削減効果も期待できる．

尚，短時間に出現するパケットフロー数が CBNP の PLC や CMH (の CMT) のエントリ数よりもはるかに大きくなって，通信の時間的局所性が弱くなるような場合には CBNP による処理が難しい．例えば，世界中で極少数しか存在しないような真のコア網では，時間的局所性が弱まるような情況が発生する可能性があり，CBNP にとって不利な情況となりうる．このような真のコア網では複雑な処理は一切行なわず，簡単なルックアップと転送だけを繰り返すような手法を用いる等，適材適所で各種の技術を適用するのが良いと考えられる．CBNP は，アクセス網や，アクセス網寄りのコア網等のネットワーク領域に特に適しており，低消費電力かつ高スループットでパケット転送処理を行なうのに有効なアーキテクチャとして期待できる．

謝辞

本論文の主査を快く引き受けてくださり，また，本論文の礎となるプロセッサやキャッシュをはじめとするコンピュータアーキテクチャの基礎から応用まで深いレベルで絶えずご指導いただいた慶應義塾大学理工学部情報工学科 天野 英晴教授に深く感謝いたします。

本論文の副査を快く引き受けてくださり，数多くの有益なアドバイスで論文の品質を向上させてくださった慶應義塾大学理工学部情報工学科 寺岡 文男教授，同情報工学科 河野 健二准教授，同システムデザイン工学科 西 宏章准教授にも深く感謝いたします。特に西准教授には，本研究の開始時より，共同研究者として数多くの有益なアイデアを与えていただきました．ここに，深く感謝の意を表します．

本論文の予備審査で数多くの有益な議論を交わし貴重なアドバイスをしていただいた慶應義塾大学理工学部情報工学科 山崎 信行准教授にも深く感謝いたします．

本研究の一部は平成 16 年度から 17 年度までの，(独) 情報通信研究機構の委託研究「テラビット級スーパーネットワークの研究開発」の一環として実施されました．

本研究の機会と指針を与え，様々な面から絶えず研究環境を調えてくださった (株) 日立製作所中央研究所 ネットワークシステム研究部 西村 信治部長にも感謝いたします．

本研究開始時に一緒に議論していただいた慶應義塾大学理工学部システムデザイン工学科 西研究室の石田 慎一氏，谷 紘太氏にもお世話になりました．感謝の意を表します．

CBNP プロトタイプの開発に際し，筆者と共に深夜まで論理設計，論理検証，実機検証，性能評価を行なっていただいた日立情報通信エンジニアリング株式会社の高田 順一氏にも感謝いたします．また，CBNP プロトタイプを実装するための FPGA テストベッドの設計，製造に際し，東京エレクトロン デバイス株式会社の森川 健一氏，他お世話になった多くの方々にも感謝いたします．

筆者は，1998 年に (株) 日立製作所入社後，エンタープライズサーバ事業部にてスーパーテクニカルサーバ SR8000 用プロセッサ設計開発への参画，米国 IBM 社とのサーバ共同開発プロジェクトへの参画を経て，2003 年夏に (株) 日立製作所中央研究所へ異動し，本研究に着手しました．異動の際に様々な面で不慣れな中，親切に支援していただいた同ネットワークシステム研究部の皆様と，現在，同プラットフォームシステム研究部で共に研究を進めている豊田 英弘氏にも感謝いたします．また，筆者が仕事の傍らで博士論文を執筆することを容認し，温かく見守ってくださった同プラットフォームシステム研究部の皆様にも感謝いたします．

そして最後に，これまでずっと支えてくれた両親と妻恵に感謝します。

2007 年 7 月 日立製作所中央研究所にて

参考文献

[1] BGP Routing Table Analysis Reports. http://bgp.potaroo.net.

[2] P. Crowley, M.A. Franklin, H. Hadimioglu, and P.Z. Onufryk. *Network Processor Design –Issues and Practices Volume 1–*. Morgan Kaufmann, 2002.

[3] Douglas E. Comer. *Network Systems Design using Network Processors (IXP1200 Version)*. Pearson Prentice Hall, 2004.

[4] Linley Gwennap and Bob Wheeler. *A Guide to NETWORK PROCESSORS Sixth Ed.* The Linley Group, 2004.

[5] 法元 寛. 低消費電力 LSI を実現する SOI デバイス技術. 沖テクニカルレビュー, Vol. 71, No. 3, Jul. 2004.

[6] George Gilder. *TELECOSM:How Inifinite Bandwidth will Revolutionize Our World*. Free Press, 2000.

[7] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, No. 19, Apr. 1965.

[8] IEEE802.3_task_force. http://grouper.ieee.org/groups/802/3/10G_study/public.

[9] Juniper Networks. http://www.juniper.net/products_and_services/t_series_core_platforms, 2004.

[10] Juniper Networks. *T640 Routing Node and TX Matrix Platform: Architecture (White Paper)*.

[11] Cisco Systems. http://www.cisco.com.

[12] Cisco Systems. *Cisco CRS-1 8-Slot Single-Shelf System(Data Sheet)*, 2004.

[13] Cisco Systems. *Cisco CRS-1 16-Slot Single-Shelf System(Data Sheet)*, 2005.

[14] AlaxalA Networks Corporation. http://www.alaxala.com/jp/support/manual/index.html.

[15] 河合栄治, 門林雄基, 山口英. ネットワークプロセッサ技術に関するサーベイ. 電子情報通信学会 IA 研究会, pp. 55–60, May. 2003.

[16] 河合栄治, 門林雄基, 山口英. ネットワークプロセッサ技術の研究開発動向. 情報処理学会論文誌 コンピューティングシステム, Vol. 45, No. SIG1(ACS4), pp. 54–65, Jan. 2004.

[17] D. L. Tennenhouse and J.M. Smith and W.D. Sincoskie and D.J. Wetherall and G.J. Minden. A Survey of Active Network Research. *IEEE Commnuications Magazine*, Vol. 35, No. 1, pp. 80–86, 1997.

[18] 山本幹. アクティブネットワークの技術動向. 電子情報通信学会論文誌 B, Vol. J84-B, No. 8, pp. 1401–1412, 2001.

[19] 長田孝彦, 東海林敏夫, 山下博之, 塩川鎮雄. マルチメディアコンテンツ転送向け高性能 TCP/IP 通信ボードの構成と評価. 情報処理学会論文誌, Vol. 39, No. 2, pp. 347–355, 1998.

[20] 西川博昭, 青木一浩. プロトコル多重処理のデータ駆動型実現法とその実験的検討. 電子情報通信学会論文誌 D-I, Vol. 85-D-I, No. 7, pp. 635–643, 2002.

[21] U. Nordqvist and D. Liu. Packet Classification and Termination in a Protocol Processor. In *Proceedings of the 2nd Workshop on Network Processors*, pp. 88–99, 2003.

[22] P. Pradhan and W. Xu and I. Nair and S. Sahu. Efficient and Faithful Performance Modeling for Network-Processor Based System Design. In *Proceedings of the 2nd Workshop on Network Processors*, pp. 125–132, 2003.

[23] 上村哲也, 熊谷幸夫. Active Network Storage のネットワーク性能の評価. 電子情報通信学会論文誌 D-I, Vol. 85-D-I, No. 9, pp. 841–849, 2002.

[24] C.J. Georgiou, V. Salapura, and M. Denneau. A Programmable Scalable Platform for Next Generation Networking. In *Proceedings of the 2nd Workshop on Network Processors*, pp. 1–9, 2003.

[25] R. Thomas, B. Mark., T. Johnson, and J. Croall. High-speed Legitimacy-based High-performance DDoS Filtering with Network Processors: A Case Study and Implementation on the Intel IXP 1200. In *Proceedings of the 2nd Workshop on Network Processors*, pp. 133–147, 2003.

[26] R. Thomas, B. Mark., T. Johnson, and J. Croall. NetBouncer: Client-legitimacy-based High-performance DDoS Filtering. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition(DISCEX III)*, 2003.

[27] M.E. Kounavis and A. Kumar and H. Vin and R. Yavatkar and A.T. Campbell. Directions in Packet Classification for Network Processors. In *Proceedings of the 2nd Workshop on Network Processors*, pp. 148–157, 2003.

[28] G. Memik and W.H. Mangione-Smith. A Flexible Accelerator for Layer7 Networking Applications. In *Proceedings of the 39th Design Automation Conference (DAC'02)*, pp. 646–651, 2002.

[29] G. Memik, S.O. Memik, and W.H. Mangione-Smith. Design and Analysis of a Layer Seven Network Processor Accelerator Using Reconfigurable Logic. In *Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02)*, 2002.

[30] Intel Corporation. `http://www.intel.com`.

[31] Intel Corporation. *Next Generation Network Processor Technologies – Enabling Cost Effective Solutions for 2.5 Gbps to 40 Gbps Network Services (White Paper)*, Oct. 2001.

[32] Intel Corporation. *Internet Exchange Architecture Network Processors: Flexible, Wire-Speed Processing from the Customer Premises to the Network Core (White Paper)*, 2002.

[33] Xelerated, Inc. `http://www.xelerated.com`.

[34] Gary Lidington. *Programming A Data Flow Processor – How to reduce the programming complexities typically associated with network processors, while maintaining the efficiency and control required for real-time, data-driven applications (White Paper)*. Xelareted, Inc., Sep. 2003.

[35] EZchip Technologies. `http://www.ezchip.com`.

[36] EZchip Technologies. *Network Processor Designs for Next-Generation Networking Equipment (White Paper)*.

[37] EZchip Technologies. *The Role of Memory in NPU System Design (White Paper)*.

[38] Peter N. Glaskowsky. Toaster3 Pops Up at MPF. *Microprocessor Report*, Oct. 16 2002.

[39] Will Eatherton. The Push of Network Processing to the Top of the Pyramid. In *Proceedings of Symoposium on Architectures for Networking and Communications Systems*, Oct. 2005.

[40] B. Doyle, R. Arghavani, D. Barlage, S. Datta, M. Doczy, J. Kavalieros, A. Murthy, and R. Chau. Transistor elements for 30nm Physical Gate Length and Beyond. *Intel Technology Journal*, Vol. 6, No. 2, May. 2002.

[41] Changbo Long and Lei He. Distributed sleep transistors network for power rerduction. In *Proceedings of Design Automation Conference 2003*, pp. 181–186, Jun. 2003.

[42] Anand Ramalingam, Bin Zhang, Anirudh Devgan, and David Z. Pan. Sleep Transistor Sizing Using Timing Criticality and Temporal Currents. In *Proceedings of Asian and South Pacific Design Automation Conference(ASPDAC), Shanghai*, Jan. 2005.

[43] Kaijian Shi and David Howard. Sleep Transistor Design and Implementation – Simple Concepts Yet Challenges To Be Optimum. In *Proceedings of VLSI Design, Automation and Test (VLSI-DAT) 2006*, pp. 1–4, Apr. 2006.

[44] James Kao, Anantha Chandrakasan, and Dimitri Antoniadis. Transistor Sizing Issues and Tool For Multi-Threshold CMOS Technology. In *Proceedings of DAC-34: ACM/IEEE Design Automation Conference*, pp. 409–414, Jun. 1997.

[45] Kaushik Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-Meimand. Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits. *Proceedings of the IEEE*, Vol. 91, pp. 305–327, Feb. 2003.

[46] Intel Corporation. *Accelerating the Convergence of Computing & Communications, Intel 90nm Communications Technology (White Paper)*, 2003.

[47] D.C. Feldmeier. Improving Gateway Performance with a Routing-Table Cache. In *Proceedings of IEEE INFOCOMM '88*, pp. 298–307, Mar. 1988.

[48] Raj Jain. Characteristics of Destination Address Locality in Computer Networks: A Comparison of Caching Schemes. *Computer Networks and ISDN Systems*, Vol. 18, No. 4, pp. 243–254, May. 1990.

[49] X. Chen. Effect of Caching on Routing-Table Lookup in Multimedia Environments. In *Proceedings of IEEE INFOCOMM '91*, Vol. 3, pp. 1228–1236, Apr. 1991.

[50] Craig Partridge, Philip Carvey, Ed Burgess, Ishidro Castineyra, Tom Clarke, Lise Graham, Michael Hathaway, Phil Herman, Allen King, Steve Kohalmi, Tracy Ma, John Mcallen, Trevor Mendez, Walter C. Milliken, Ronald Pettyjohn, John Rokosz, Joshua Seeger, Michael Sollins, Steve Starch, Benjamin Tober, Gregory Troxel, David Waitzman, and Scott Winterble. A 50-Gb/s IP Router. *IEEE/ACM Transactions on Networking*, Vol. 6, No. 3, pp. 237–248, 1998.

[51] Tzi Cker Chiueh and Prashant Pradhan. High-Performance IP Routing Table Lookup Using CPU Caching. In *Proceedings of INFOCOM'99*, pp. 1421–1428, 1999.

[52] Tzi Cker Chiueh and Prashant Pradhan. Cache Memory Design for Network Processors. In *Proceedings of IEEE High Performance Computer Architecture Conference (HPCA) 2000*, pp. 409–419, Jan. 2000.

[53] Pankaj Gupta, Steven Lin, and Nick McKeown. Routing Lookups in Hardware at Memory Access Speeds. In *Proceedings of INFOCOM'98*, pp. 1240–1247, 1998.

[54] Bryan Talbot, Timothy Sherwood, and Bill Lin. IP Caching for Terabit Speed Routers. In *Proceedings of Global Communication Conference (Globecom'99)*, pp. 1565–1569, 1999.

[55] I.L. Chvets and M.H. MacGregor. Multi-zone Caches for Accelerating IP Routing Table Lookups. In *Proceedings of High Performance Switching and Routing (HPSR) 2002*, pp. 121–126, 2002.

[56] Soraya Kasnavi, Paul Berube, Vincent C. Gaudet, and Jose Nelson Amaral. A Multizone Pipelined Caches for IP Routing . In *Proceedings of International IFIP-TC6 networking conference No4, Waterloo, CANADA*, Vol. 3462, pp. 574–585, 2005.

[57] Kari Seppänen. Novel IP Address Lookup Algorithm for Inexpensive Hardware Implementation. *WSEAS Trans. on Commun.*, Vol. 1, No. 1, pp. 76–84, 2002.

[58] Stefan Nilsson and Gunnar Karlsson. IP address lookup using LC-tries. *IEEE Journal on Selected Areas in Communications*, pp. 1083–1092, 1999.

[59] Mikael Degermark, Andrej Brodnik, Svante Carlsson, and Stephen Pink. Small Forwarding Tables for Fast Routing Lookups. In *Proceedings of SIGCOMM*, pp. 3–14, 1997.

[60] Miguel Á. Ruiz-Sánchez and Ernst W. Biersack. Survey and Taxonomy of IP Address Lookup Algorithms. *IEEE Network*, pp. 8–23, 2001.

[61] 奥野通貴, 西宏章. ヘッダラーニングキャッシュとキャッシュミスハンドラを利用したネットワークプロセッサ高速化方式の提案と評価. 電子情報通信学会 CS 研究会 CS2003-168, pp. 61–66, Mar. 2004.

[62] 奥野通貴, 西宏章. 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. *IPSJ Trans. on Advanced Computing System*, Vol. 45, No. SIG 6, pp. 45–53, May. 2004.

[63] Michitaka Okuno and Hiroaki Nishi. Network-Processor Acceleration-Architecture Using Header-Learning Cache and Cache-Miss Handler. In *Proceedings of The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI2004*, Vol. III, pp. 108–113, Jul. 2004.

[64] 奥野通貴, 西宏章. 次世代 IP バックボーンネットワークを支える高スループット・低消費電力パケット処理アーキテクチャ. 電子情報通信学会 IN 研究会 IN2004-89, 第 104 巻, pp. 35–40, Oct. 2004.

[65] Michitaka Okuno, Shinichi Ishida, and Hiroaki Nishi. Low-Power Network-Packet-Processing Architecture Using Process-Learning Cache for High-end Backbone Router. *IEICE Trans. Electron.*, Vol. E88-C, No. 4, pp. 536–543, Apr. 2005.

[66] 奥野通貴, 西村信治, 石田慎一, 西宏章. ネットワークトラフィックの時間的局所性を利用したブロードバンドネットワーク向けキャッシュ型パケット処理技術. 情報処理学会論文誌ブロードバンド・ユビキタス・ネットワークとその応用特集号, Vol. 47, No. 2, pp. 346–354, Feb. 2006.

[67] Michitaka Okuno, Shinji Nishimura, Shinichi Ishida, and Hiroaki Nishi. Cache-based Networkr Processor Architecture: Evaluation with Real Network Traffic. *IEICE Trans. Electron.*, Vol. E89-C, No. 11, pp. 1620–1628, Nov. 2006.

[68] CERNET BGP View Global Internet. http://bgpview.6test.edu.cn/bgp-view/index.shtml.

[69] University of Auckland: Auckland-II trace data. http://wand.cs.waikato.ac.nz/wand/wits/auck/2/.

[70] ADV site information. http://pma.nlanr.net/PMA/Sites/ADV.html.

[71] NLANR: The National Laboratory for Applied Network Research Project. http://moat.nlanr.net.

[72] APN site information. http://pma.nlanr.net/PMA/Sites/APN.html.

[73] BUF site information. http://pma.nlanr.net/PMA/Sites/BUF.html.

[74] BWY site information. http://pma.nlanr.net/PMA/Sites/BWY.html.

[75] COS site information. http://pma.nlanr.net/PMA/Sites/COS.html.

[76] MEM site information. http://pma.nlanr.net/PMA/Sites/MEM.html.

[77] TXG site information. http://pma.nlanr.net/PMA/Sites/TXG.html.

[78] AIX site information. http://pma.nlanr.net/PMA/Sites/AIX.html.

[79] Abilene Indianapolis router. http://pma.nlanr.net/Traces/long/ipls1inde.html.

[80] MRA site information. http://pma.nlanr.net/PMA/Sites/MRA.html.

[81] WIDE:Widely Integrated Distributed Environment. http://tracer.csl.sony.co.jp/mawi/.

[82] University of Auckland: Auckland-IV trace data. http://wand.cs.waikato.ac.nz/wand/wits/auck/4/.

[83] Xilinx Inc. *Virtex-II Pro and Virtex-II Pro X FPGA User Guide, v4.0 edition*, Mar. 2005.

[84] Taiwan Semiconductor Manufacturing Company Ltd. *Enabling Innovation, TSMC Fall 2002 Technology Symposium*, 2002.

[85] 総務省. ユビキタスネット社会の進展と環境に関する調査研究会 報告書 参考資料 2 ICT 機器・インフラの電力消費の予測. http://www.soumu.go.jp/joho_tsusin/policyreports/chousa/yubi_pande/pdf/0503_houkoku_s01.pdf, 2005.

[86] Intel Corporation. *Intel Common System-Level Interface Connecting 10 GbE LANs/WANs to OC-192 MANs (White Paper)*, Aug. 2002.

論文目録

本研究に関する論文

公刊論文

1. 奥野 通貴, 西 宏章: “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.”, 情報処理学会論文誌コンピューティングシステム (ACS), No.6 May. 2004. pp. 45-53.

2. Michitaka Okuno, Shinichi Ishida, Hiroaki Nishi: “Low-Power Network-Packet-Processing Architecture Using Process-Learning Cache for high-End Backbone Router.”, IEICE Transactions on Electronics, Vol. E88-C, No.4, Apr. 2005. pp. 536-543.

3. 奥野 通貴, 西村 信治, 石田 慎一, 西 宏章: “ネットワークトラフィックの時間的局所性を利用したブロードバンドネットワーク向けキャッシュ型パケット処理技術.”, 情報処理学会論文誌ブロードバンド・ユビキタス・ネットワークとその応用特集号, 情報処理学会論文誌, Vol.47, No.2, Feb. 2006. pp. 346-354.

4. Michitaka Okuno, Shinji Nishimura, Shinichi Ishida, Hiroaki Nishi: “Cache-based Network Processor Architecture: Evaluation with Real Network Traffic.”, Special Section on Novel Device Architectures and System-Integration Technologies, IEICE Transactions on Electronics, Vol. E89-C, No.11, Nov. 2006. pp. 1620-1628.

国際会議

1. Michitaka Okuno, Hiroaki Nishi: “Network-Processor Acceleration-Architecture Using Header-Learning Cache and Cache-Miss Handler.”, The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI2004, volume III, Communication and Network Systems, Technologies and Applications. Jul. 2004. pp. 108-113.

研究会ほか

1. 奥野 通貴, 西 宏章: “ヘッダラーニングキャッシュとキャッシュミスハンドラを利用したネットワークプロセッサ高速化方式の提案と評価.”, 電子情報通信学会. CS2003-168. Mar. 2004. pp. 61-66.

2. 奥野 通貴, 西 宏章: “次世代 IP バックボーンネットワークを支える高スループット・低消費電力パケット処理アーキテクチャ.”, 電子情報通信学会. IN2004-89. Vol.104, IN-340, Oct. 2004.

pp. 35-40.

その他の論文

研究会ほか

1. 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. NS2003-121,IN2003-87,CS2003-96. Vol.103, No.310, Sep. 2003. pp. 81-84.

付 録A　CBNPのトークンの実装例

トークンの実装例として，第 4 章で示した CBNP プロトタイプでのトークンの詳細を示す.

表 A.1: トークンの実装例

名称	略称	bit 幅	意味
Unique Information	U-Info	22 bit	パケット毎に与える固有の情報
Analyzed Information	A-Info	12 bit	パケットの解析情報
Extraxted Information	E-Info	336 bit	パケットから抽出した情報
Replaced Information	R-Info	336 bit	パケットヘッダの置換情報や追加情報
Processing Information	P-Info	30 bit	パケット処理結果の一部，及びその処理結果適用方法

表 A.2: U-Info の実装例

bit	bit 幅	名称	略称	意味
[21]	1	TCP_FIN	TCPFIN	TCP の ACK bit が 0 かつ FIN bit が 1 であること を示す．フローの最後のパケットであるため PLC ヒットなら，PLC エントリを消去する PLC ミスなら，PLC への登録を行なわない
[20]	1	L3_IP _FRAG MENTATION	IPFRGMNT	IPv4 のフラグメント bit
[19]	1	PLC_CONTROL	PLCCTLR	ホスト CPU や P-Engine が発行した PLC 制御用トークンである. 制御内容は U-Info[17:0] を利用して通知
[18]	1	ERROR	ERR	パケットにエラーが含まれていることを示す エラー内容は E-Info[31:0] フィールド を利用して通知
[17]	1	REQUEST_TO _P-ENGINE	PEXE	BSP で未サポート種類のパケット であることを示す．P-Engine または 外部 CPU で処理が必要である.
[16:11]	6	reserved	rsv	予約
[10:0]	11	PACKET _MEMORY _ADDRESS	PMEMADDR	パケットを記録したパケットメモリ の先頭アドレスを示す.

表 A.3: A-Info の実装例

bit	bit 幅	名称	略称	意味
[11:10]	2	L4_TYPE	L4TP	Layer4 種類を示す 00: 未判定，もしくは想定外のプロトコル 01: 「L3TP が IPv4 であり L4TP が IPv6」または「L3TP が IPv6 であり L4TP が IPv4」 10: TCP 11: UDP
[9:8]	2	L3_TYPE	L3TP	Layer3 種類を示す 00: 未判定，もしくは想定外のプロトコル 01: ユーザが指定したプロトコルである 10: IPv4 11: IPv6
[7:5]	3	L2INFO_MPLS	L2MPLS	MPLS 種類を示す 0**: MPLS unicast 1**: MPLS multicast *00: shim header 0 個 *01: shim header 1 個 *10: shim header 2 個 *11: shim header 3 個以上
[4:2]	3	L2INFO_VLAN	L2VLAN	タグ VLAN 情報を示す 1**: RIF=1 であり，VLAN tag の長さが判定不能，もしくは，VLAN tag が 4 個以上 000: VLAN tag 0 個 001: VLAN tag 1 個 010: VLAN tag 2 個 011: VLAN tag 3 個以上
[1]	1	INPUT_PORT	IPORT	入力ポート番号 (プロトタイプでは 2 個)
[0]	1	L2INFO_TYPE	L2TP	Ethernet タイプ 0: Ethernet Version2 1: IEEE 802.3 SNAP

表 A.4: P-Info の実装例

bit	bit 幅	名称	略称	意味
[29]	1	PRIORITY	PRIO	パケット処理の優先度を示す. 1 が高優先度，0 が低優先度. 今回のプロトタイプでは，優先度を 2 個に 限定しているため 1bit. 複数レベルの優先度を想定する場合, bit 数を増やして対応.
[28:27]	2	SWITCH _PORT	SPORT	ルータでの出力ポート識別子 対応する宛先を 1，それ以外を 0 とする. マルチキャストの場合，複数の bit を 1 とする. 今回のプロトタイプでは，出力ポートを 2 個に 限定しているため 2bit. N ポートのスイッチを想定すれば，N bit となる
[26]	1	DISCARD	DCD	対応するパケットを廃棄する.
[25:24]	2	NEW _HEADER	NHD	ADD/DELETE 操作を行なったあとの 新しいヘッダに修正が必要であることを示す. 00: 修正不用 01: MPLS ヘッダ用修正必要 10: IPv4 ヘッダ用修正必要 11: IPv6 ヘッダ用修正必要
[23:19]	5	ADD _OFFSET	AOS	ヘッダ追加オペレーションを実施する 先頭 Byte 位置を示す. 0〜31Byte の範囲で指定可能.
[18:13]	6	ADD _SIZE	ASZ	ヘッダ追加オペレーションで追加する 情報の Byte 数．0〜63Byte の範囲で指定可能. (実際には R-Info の最大長 42Byte までである．)
[12]	1	ADD	ADD	1 ならば，ヘッダ追加オペレーション実施 追加位置は AOS で，追加サイズは ASZ で指定.
[11:7]	5	DELETE _OFFSET	DOS	ヘッダ削除オペレーションを実施する 先頭 Byte 位置を示す. 0〜31Byte の範囲で指定可能.
[6:1]	6	DELETE _SIZE	DSZ	ヘッダ削除オペレーションで削除する 情報の Byte 数．0〜63Byte の範囲で指定可能.
[0]	1	DELETE	DEL	1 ならば，ヘッダ削除オペレーション実施 削除位置は DOS で，削除サイズは DSZ で指定.

表 A.5: E-Info の実装例

bit	bit 幅	名称	略称	意味
[335:0]	336	EXTRACTED_INFO	E-Info	パケットから抽出した情報. 情報の並べ方や抽出内容はクラシフィケーション に依存し，コントロールレジスタで設定 例えば，単純な IP フォワーディングでは， 宛先 IP アドレスだけを抽出する. 5-tuple の場合は，宛先/送信元 IP アドレス プロトコル番号，宛先/送信元ポート番号を 抽出する. 有効な情報以外の部分はゼロで埋める.

表 A.6: R-Info の実装例

bit	bit 幅	名称	略称	意味
[335:0]	336	REPLACED_INFO	R-Info	カプセリング化用の追加情報や パケットヘッダへの置換情報. カプセリング用の追加情報の例としては IPv4 ヘッダや IPv6 ヘッダ，MPLS 網のエッジに 位置する Label Edge Router (LER) での 追加用のシムヘッダ等が挙げられる. 置換情報の例としては，ルータがパケットを受信し 当該パケットを次のルータへ転送する際に必要と なる付替用の送信元/宛先 MAC アドレスや， MPLS 網の内部に位置する Label Switch Router (LSR) で付け替えるシムヘッダ内のラベル等が挙げられる， トークン上では E-Info フィールド全体を R-Info で 置換する．有効な情報以外の部分はゼロで埋める.

付 録B　CBNPプロトタイプのコントロールレジスタ実装例

コントロールレジスタの実装例として，第 4 章で示した CBNP プロトタイプでのコントロールレジスタの詳細を示す.

表 B.1: コントロールレジスタの実装例 (前半)

bit	bit 幅	名称	略称	意味
[31:28]	4	reserved	rsv	予約
[27:24]	4	BEHAVIOR_SETTING	BHVSET	CBNP の動作設定 0000: IP(L3) の DIP によりルーティング. 同時に IP_OP 参照 0001: IP(L3) の DIP/SIP により分類し， ルーティング. 同時に IP_OP 参照 0010: IP(L3) で 5-tuple により分類し， ルーティング．もしくは， IPv4/IPv6 のカプセリング処理. 同時に IP_OP 参照 1000: MAC(L2) の DIP によりルーティング. 同時に VLAN_OP，LNUM 参照 1001: MAC(L2) の DIP/SIP により分類し， ルーティング. 同時に VLAN_OP，LNUM 参照 1010: MPLS Label Edge Router 処理 EoMPLS ラベル付加/削除 同時に MPLS_OP 参照 1011: MPLS Label Switch Router 処理 最も外側の MPLS ラベル置換 または，削除 同時に MPLS_OP 参照 上記以外: 予約
[23:22]	2	reserved	rsv	予約

表 B.2: コントロールレジスタの実装例 (後半)

bit	bit 幅	名称	略称	意味
[21]	1	DISABLE_PLC	DISPLC	1 ならば PLC 無効化
[20]	1	IGNORE _MY_MAC	IGMMC	1 ならば自宛以外の MAC でも処理
[19:16]	4	Internet _PROTOCOL _OPERATION	IP_OP	IP 処理の動作設定 0000: IP ヘッダ追加/削除処理無し 1000: IPv4 ヘッダ追加 (エンカプセリング) 1001: IPv4 ヘッダ削除 (デカプセリング) 1010: IPv6 ヘッダ追加 (エンカプセリング) 1011: IPv6 ヘッダ削除 (デカプセリング) 上記以外: 予約
[15:12]	4	VLAN _OPERATION	VLAN_OP	VLAN tag 処理の動作設定 0000: VLAN tag 追加/削除処理無し 0001: VLAN tag 追加 (数は LNUM 参照) 0010: VLAN tag 削除 (数は LNUM 参照) 0011: VLAN tag 置換 (数は 1 個) 0010: IPv6 ヘッダ付加 (エンカプセリング) 上記以外: 予約
[11:8]	4	MPLS _OPERATION	MPLS_OP	EoMPLS 処理の動作設定 0000: EoMPLS ラベル追加 0001: EoMPLS ラベル削除 0010: IP-VPN：MPLS ラベル追加 0011: IP-VPN：MPLS ラベル削除 上記以外: 予約
[7:2]	6	reserved	rsv	予約
[1:0]	2	LABEL _NUMBER	LNUM	追加/削除数を示す 01: 追加/削除数 1 個 10: 追加/削除数 2 個 11: 追加/削除数 3 個

付 録C　代表的なプロトコルのヘッダフォーマット

ネットワークプロセッサがパケット転送処理の際に参照しうるレイヤ 2，レイヤ 2.5，レイヤ 3，レイヤ 4 の代表的なプロトコルのヘッダフォーマットを示す.

C.1　レイヤ 2プロトコルのヘッダフォーマット (Ethernet)

レイヤ 2 の代表的プロトコルとして Ethernet のヘッダフォーマットを図 C.1に示す．Ethernet は DIX 1.0 規格から始まり，現在では 802.3 RAW，802.3 with LLC (Logical Link Control)，802.3 with SNAP (SubNetwork Access Protocol)，Ethernet Ver.2 の 4 種類が用いられている．最も広く利用されているのは，Ethernet Ver.2 と呼ばれるフォーマットである．これらのフォーマットは，DMAC/SMAC に続く 2Byte のフィールドの意味をどのように解釈するかという問題から誕生した.

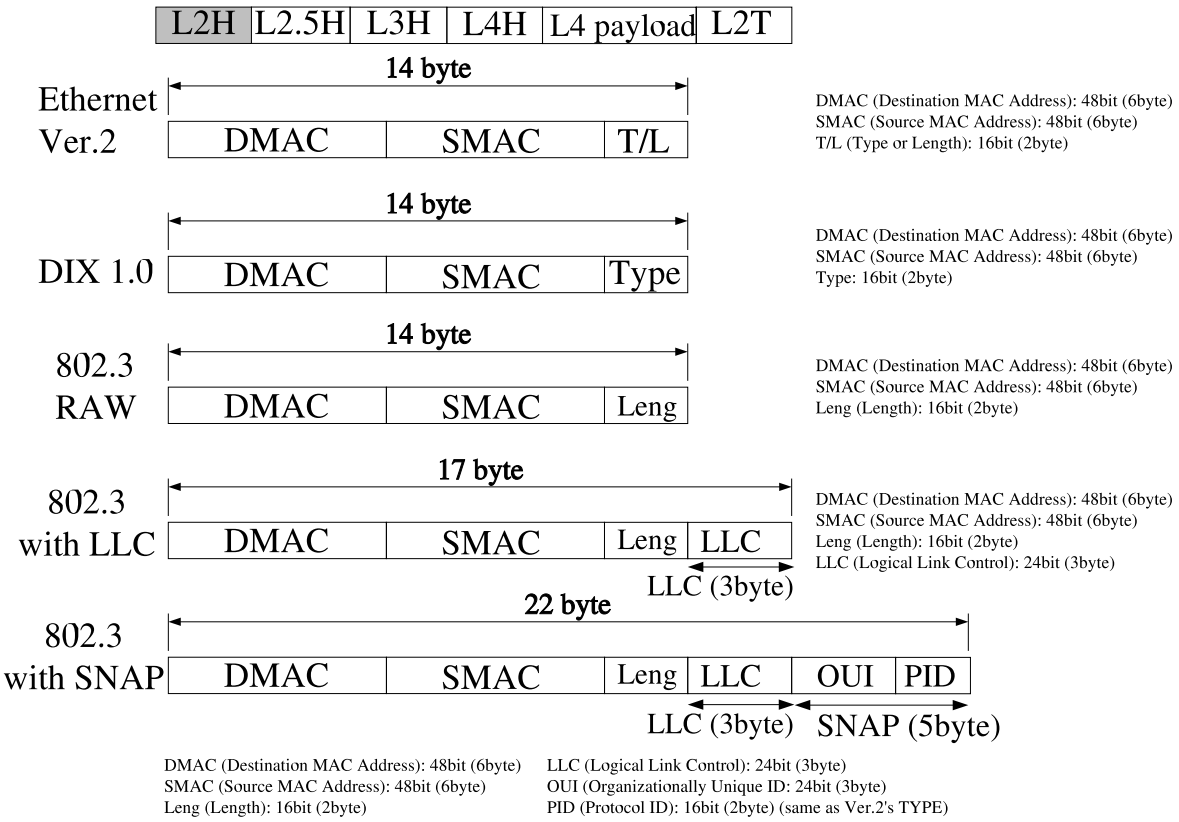

図 C.1: Ethernet (レイヤ 2) のヘッダフォーマット

- **Ethernet Ver.2**: 1997 年の IEE802.3x(全二重規格) 制定時に定められたフォーマットであり，DMAC/SMAC に続く 2Byte のフィールドを Type/Length フィールドとして両解釈する．現在広く利用されている Ethernet フォーマットである．Type/Length フィールドの値が 16 進数 0x0600 (10 進数 1536) 未満の場合は Ethernet のペイロード長 (Length) を示し，16 進数 0x0600 (10 進数 1536) 以上の場合は上位層のプロトコル (Type) を示す．

- **DIX 1.0**: DEC 社，Intel 社，Xerox 社によって規定された Ethernet の最初の規格．Type フィールドは上位層のプロトコルを示す．

- **802.3 RAW**: IEEE802.3 で標準化された Ethernet フォーマット．DIX 1.0 規格では Type として規定されたフィールドがペイロード長を示す Length に変更された．Novell NetWare で利用されており，上位層のプロトコルとして IPX しか利用しないため Type フィールドが必要なかった．

- **802.3 with LLC**: 上位層のプロトコルを区別するために 802.3 RAW に 3Byte の LLC フィールドを追加．LLC フィールドは 1Byte の DSAP (Destination Service Access Point)，1Byte の SSAP (Source Service Access Point)，1Byte の制御情報で構成し，DSAP，SSAP で上位層のプロトコルを区別する．尚，最上位の 1bit は予約されているので区別できるプロトコル数は 2^7 種類 (128 種類) までである．

- **802.3 with SNAP**: 上位層のプロトコルを DIX 1.0 と同等数区別するために 802.3 with LLC に 5Byte の SNAP フィールドを追加した．SNAP フィールドは 3Byte のベンダを区別する ID (OID) と 2Byte の PID で構成し，PID で上位層のプロトコルを区別する．

いずれのフォーマットでも，Ethernet の標準的なペイロード長は 46Byte〜1500Byte (Ethernet のヘッダとトレイラも含めると 64Byte〜1518Byte) である．802.3 with LLC や 802.3 with SNAP では LLC フィールドと SNAP フィールドがペイロード領域の先頭部分を利用しているため，実質的なペイロード長は LLC，SNAP フィールドの分だけ短くなっている．尚，これらのフォーマットは同時に出現しても次のような手順で区別できる．

1. T/L フィールドの値が 16 進数 0x0600 以上であれば，Ethernet Ver.2．No なら次へ．
2. T/L フィールドの次の 3Byte (LLC の位置) が 16 進数 0xAAAA03 であれば，802.3 with SNAP．No なら次へ．
3. T/L フィールドの次の 2Byte が 16 進数 0xFFFF であれば，上位層のプロトコルが IPX であると判断 (先の 2Byte は IPX の Check Sum フィールドに相当) し 802.3 RAW．No なら 802.3 with LLC．

C.2 レイヤ 2.5プロトコルのヘッダフォーマット (MPLS，タグ VLAN)

レイヤ 2.5 の代表的なプロトコルとして MPLS とタグ VLAN のヘッダフォーマットを示す．

まず，図 C.2に MPLS のヘッダフォーマットを示す．L2 の Type フィールドの値が 16 進数 0x8847 であればユニキャスト用の MPLS，16 進数 0x8848 であればマルチキャスト用の MPLS である．MPLS では，Label，EXP (Experimental Use)，S (Bottom of Stack)，TTL の 4 フィールドを含むシムヘッダと呼ぶ 4Byte の識別子を利用する．MPLS ネットワーク内では，ラベルの値を次々と付け替えながらパケット転送 (ラベルスイッチング) する．また，大規模ネットワークにおいて，管理組織が異なる階層化された MPLS ネットワークの間を透過的にラベルスイッチングするために，複数のシムヘッダ (ラベル) をスタック (カプセリング) する機能を持つ．スタックされている場合，最も外側 (L2 ヘッダ側) のラベルを参照して転送を行なう．

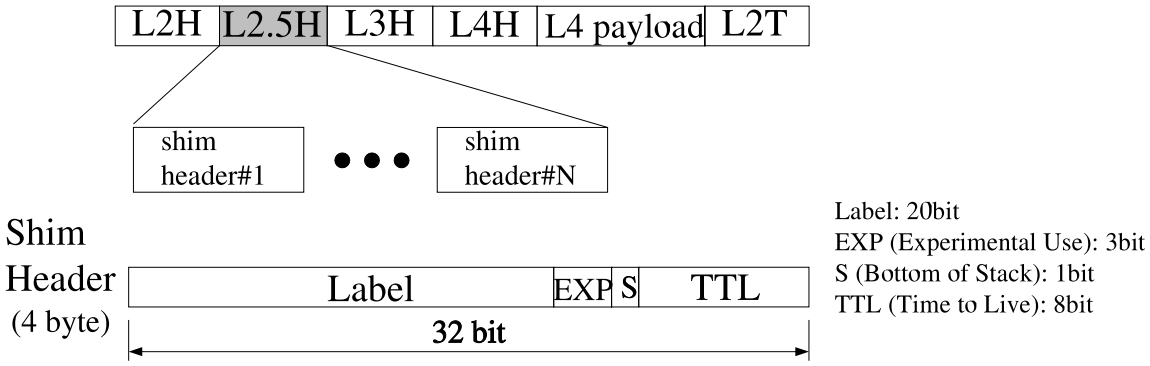

図 C.2: MPLS (レイヤ 2.5) のヘッダフォーマット

- **Label (20bit)**: ラベルスイッチングに利用する宛先識別子である．ラベルから適切な出力インタフェースを検出して出力する．また，ラベルは出力用のラベルに書換える．ラベルの 0〜15 の 16 個は制御用に予約されている．
- **EXP (Experimental Use) (3bit)**: 実験用の予約フィールドである．現在は，DiffServ 用のデータのサービスクラスを識別するフィールドとして利用されることが多い．
- **S (Bottom of Stack) (1bit)**: 1である場合，スタックされている末尾のシムヘッダであることを示す．
- **TTL (Time to Live) (3bit)**: IPv4 の TTL と同じ．利用方法も同じで，ルータを経由する毎に 1 減算して，ゼロになったら当該パケットを廃棄 (expire) する．MPLS ネットワークの入口のエッジルータで，IPv4 の TTL (IPv6 の Hop Limit) フィールドを複写する．また，MPLS ネットワークの出口のエッジルータでは，この TTL を IPv4 の TTL (IPv6 の Hop Limit) フィールドへ複写する．

次に，図 C.3にタグ VLAN のヘッダフォーマットを示す．タグ VLAN は，L2 ヘッダに続いて VLAN Tag と呼ぶ 4Byte の識別子を利用して，公衆網を個々のユーザを分離する VPN (Virtual Private Network) を構築するために用いられる．MPLS と同様に階層的な管理が出来るように，VLAN Tag をスタックすることもできる．

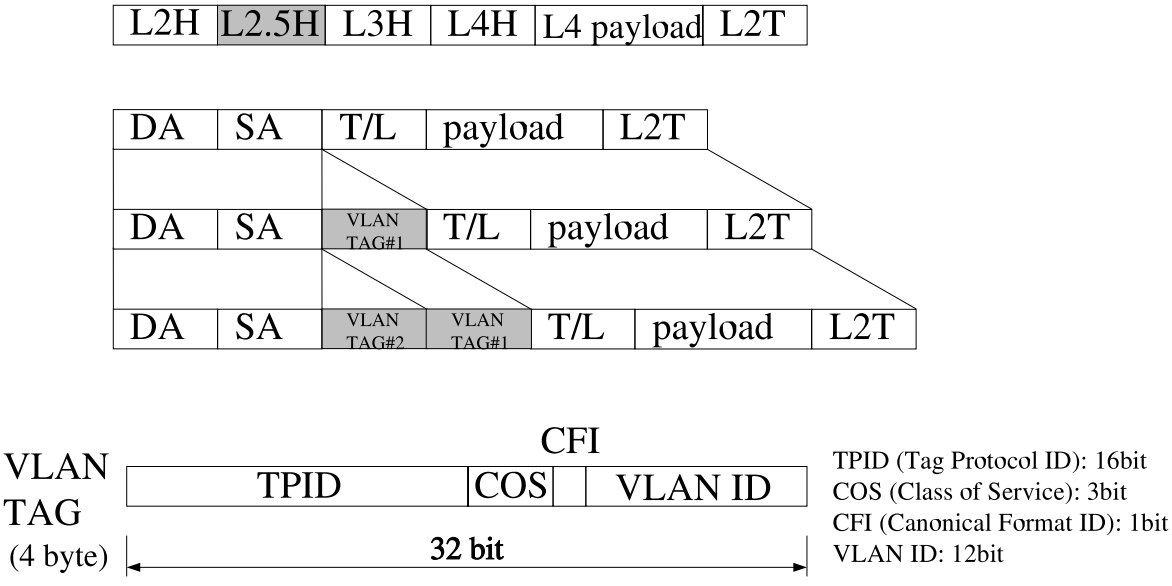

図 C.3: タグ VLAN (レイヤ 2.5) のヘッダフォーマット

- **TPID (16bit)**: TPID が 16 進数 0x8100 であれば VLAN Tag であることを示す．この他，スタック用の TPID 0x9100 (事業者用の VLAN Tag) や TPID 0xA100 (Ethernet over Ethernet 用の VLAN Tag) 等がある．
- **COS (Class of Service) (3bit)**: 0 (低い)〜7 (高い) までの 8 種類の優先度を示す．
- **CFI (Canonical Format ID) (1bit)**: 0 なら Big Endian，1 なら Little Endian のペイロードが続くことを示す．実質的には，0 であれば Ethernet フレームを意味し，1 であればトークンリングフレームを意味する．
- **VLAN ID (12bit)**: VLAN で相互に区別するための 4096 種類表現可能な識別子．ただし，0 は優先度のみを表現する識別子として，また，4095 は管理処理用に予約済．

C.3 レイヤ 3プロトコルのヘッダフォーマット (IPv4, IPv6)

レイヤ 3 の代表的なプロトコルのヘッダフォーマットとして図 C.4に IPv4 ヘッダ，図 C.5に IPv6 ヘッダを示す．

- **IPv4**: 現在広く利用されている IP フォーマットである．L2 の Type フィールドの値が 16 進数 0x0800 であれば IPv4 である．

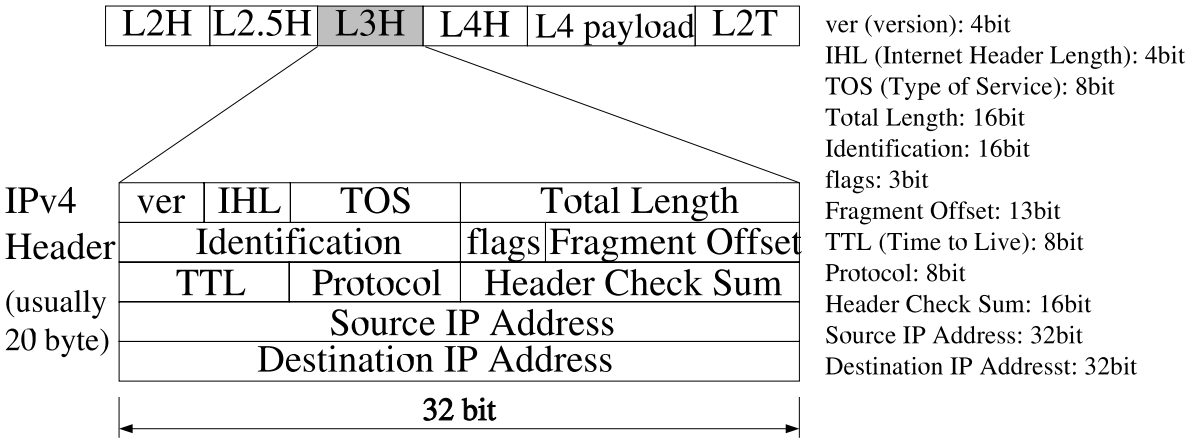

図 C.4: IPv4 (レイヤ 3) のヘッダフォーマット

  - **Ver (4bit)**: Version 番号を示す．IPv4 では 4.
  - **IHL (Internet Header Length) (4bit)**: IP ヘッダ長.
  - **TOS (Type of Service) (8bit)**: サービス品質を定義するフィールド．上位層によって指定される．先頭の 3bit で優先度，次の 3bit で低遅延要求，高スループット要求，高信頼性要求，最後の 2bit は予約である．厳密に使いこなすことは困難であり，DiffServ 等で TOS フィールドを別の定義で利用することもある.
  - **Total Length (16bit)**: IPv4 ヘッダとペイロードの合計長を示す.
  - **Identification (16bit)**: パケット識別子
  - **flags (3bit)**: bit0 は予約．bit1 はパケットのフラグメント禁止を示し，0 なら分割可能，1 なら分割不可能．bit2 が 0 ならフラグメントされた最後のパケット．1 なら途中のパケット.
  - **Fragment Offset (13bit)**: パケットをフラグメントしたデータが元のパケットのどこに位置するかを示す.
  - **TTL (Time to Live) (8bit)**: パケットの寿命を示し，一つのホスト (ルータ) を経由する毎に 1 減算し，ゼロになった時点で当該パケットは廃棄される.
  - **Protocol (8bit)**: 上位層 (レイヤ 4) のプロトコルを示す.
  - **Source IP Address (32bit)**: 送信元 IP アドレス
  - **Destination IP Address (32bit)**: 宛先 IP アドレス

- **IPv6**: IPv4 のアドレス枯渇問題の解消等を狙い，IPv4 の基本的な概念を踏襲しながら，IP アドレスを 32bit から 128bit に拡張したフォーマットである．L2 の Type フィールドの値が 16 進数 0x86DD であれば IPv6 である．

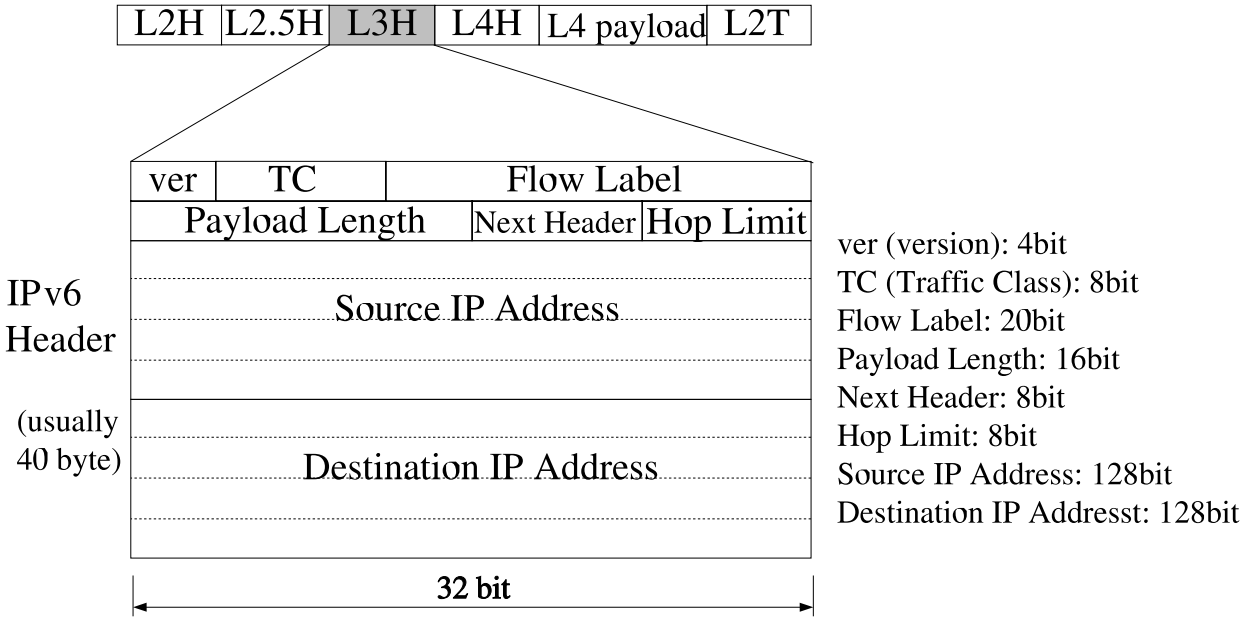

図 C.5: IPv6 (レイヤ 3) のヘッダフォーマット

  - **Ver (4bit)**: Version 番号を示す．IPv6 では 6.
  - **Traffic Class (8bit)**: IPv4 の TOS フィールドに相当.
  - **Flow Lable (20bit)**: フローを識別するために利用.
  - **Payload Length (16bit)**: IPv6 のペイロード長のみを示す．IPv6 ヘッダ長を含まない点で IPv4 の Total Length と異なる．
  - **Identification (16bit)**: パケット識別子
  - **Next Header (8bit)**: IPv4 の Protocol フィールドに相当.
  - **Hop Limit (8bit)**: IPv4 の TTL に相当.
  - **Source IP Address (128bit)**: 送信元 IP アドレス
  - **Destination IP Address (128bit)**: 宛先 IP アドレス

C.4 レイヤ 4プロトコルのヘッダフォーマット (UDP，TCP)

レイヤ 4 の代表的なプロトコルのヘッダフォーマットとして図 C.6に TCP ヘッダ，図 C.7に UDP ヘッダを示す.

- **TCP**: レイヤ 3プロトコルにおいて，上位層を示すフィールドの値が 10 進数 6 であれば TCP である．TCP は信頼性のあるデータ転送を行なうためにデータ破壊や重複，順番の入れ替わり等の問題に対応できるようになっている.

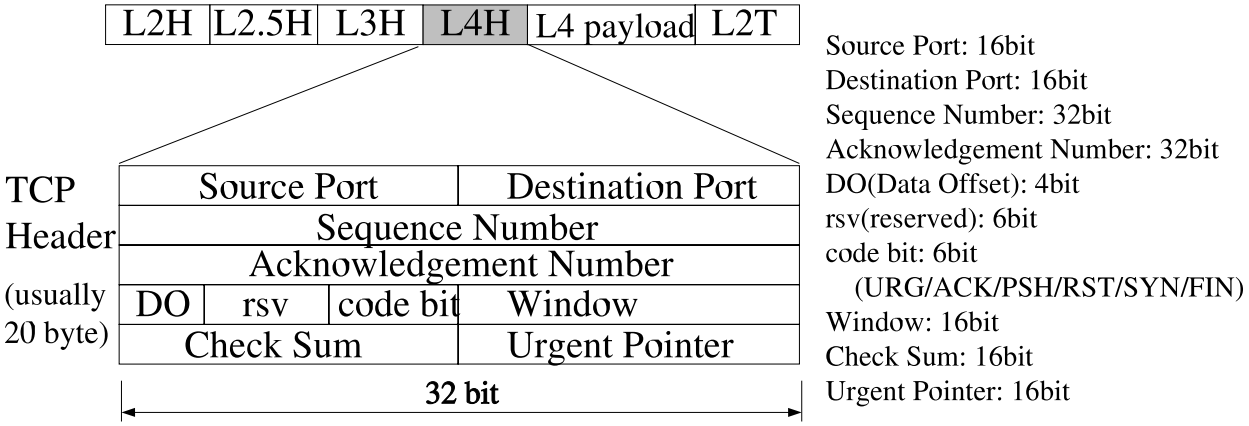

図 C.6: TCP (レイヤ 4) のヘッダフォーマット

  - **Source Port (16bit)**: 送信元ポート番号.
  - **Destination Port (16bit)**: 宛先ポート番号.
  - **Sequence Number (32bit)**: 当該 TCP セグメント (パケット) が，送信される TCP セグメント群全体の中でどの位置であるかを示すシーケンス番号．宛先では，Sequence Number 順にパケットを組み立てることで順番の入れ替わりを防げる.
  - **Acknowledgement Number (32bit)**: 確認応答番号が格納される．次に受信すべきデータの Sequence Number を意味する.
  - **Data Offset (4bit)**: TCP ヘッダのヘッダ長.
  - **Code Bit (6bit)**: フラグとして利用する．URG，ACK，PSH，RST，SYN，FIN の 6 フラグ (各 1bit) があり，それぞれの役割は以下の通り.
    * **URG (Urgent Flag)**: 1 の場合，緊急に処理すべきデータが含まれている．データ位置は Urgent Pointer フィールド (TCP の先頭から何 Byte (オクテット) 目にあるか示す) で定義.
    * **ACK (Acknowledgement Flag)**: 1 の場合，Acknowledgement Number フィールドが有効．コネクションを確立するときの SYN セグメント以外はいつでも 1 になっている必要がある.
    * **PSH (Push Flag)**: 1 の場合，受信データをすぐに上位層のアプリケーションへ渡す．0 ならすぐに渡さずバッファリングしても良い.
    * **RST (Reset Flag)**: 1 の場合，強制的にコネクションが切断される．これは何らかの異常が検出された場合に送信される．例えば，使用しないポート番号に接続要

求が来た場合，接続できないので受信側は RST=1 として送信する．元の送信側では RST=1 受信によりコネクションを切断する．

- **SYN (Synchronize Flag)**: 1 の場合，コネクションの確立要求を示す．SYN=1 のセグメント (パケット) を SYN セグメントと呼ぶ．SYN セグメントには，Sequence Number フィールドに初期シーケンス番号を格納している．
- **FIN (Fin Flag)**: 1 の場合，送信すべきデータがもう無いことを示し，コネクションの切断要求を示す．FIN=1 のセグメント (パケット) を FIN セグメントと呼ぶ．送信すべきデータを全部送信したら，FIN セグメントを送信側と受信側で送り合い，コネクションを切断する．

– **Window (Window Size)(16bit)**: 受信側で受信可能なデータサイズを送信側に通知するのに利用．送信側は Window 以上のデータを送信できない．

– **Check Sum (16bit)**: 送信したデータが壊れていないか確認するために利用．

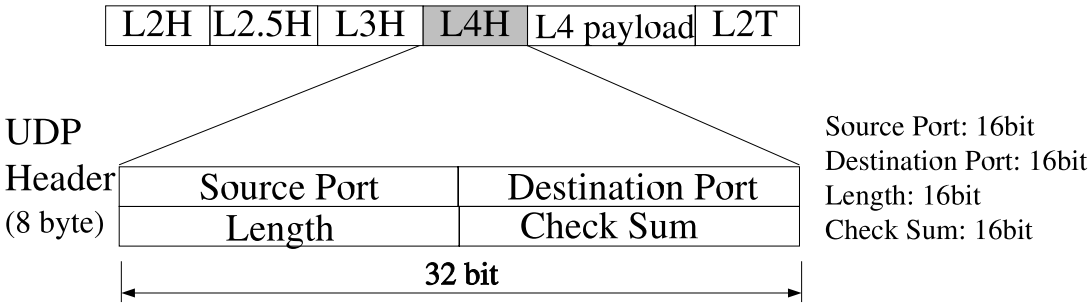

図 C.7: UDP (レイヤ 4) のヘッダフォーマット

- **UDP**: L3 の上位層を示すフィールドの値が 10 進数 17 であれば UDP である．UDP は信頼性より通信速度を重視したプロトコルであり，TCP と異なり送信先にデータが届いたかどうか確認をとらないため処理が容易である．音声や動画等の通信，データ量の小さい通信，ブロードキャストやマルチキャストに利用される．

  – **Source Port (16bit)**: 送信元ポート番号．相手からの返事を必要としない通信なので，送信元ポート番号はなくても良く，その場合は値をゼロとする．

  – **Destination Port (16bit)**: 宛先ポート番号．

  – **Length (16bit)**: UDP ヘッダ長と UDP ペイロード長の合計値．

  – **Check Sum (16bit)**: 送信したデータが壊れていないか確認するために利用．

付 録D　NP周辺の主要なI/Oインタフェース

ネットワークプロセッサで利用される主なI/Oインタフェースは，大きくはEthernetスタイルとSonetスタイルに分類できる．図D.1に典型的なEthernet搭載のラインカード，または，Sonet搭載のラインカードで利用されるI/Oインタフェースの種類と搭載位置を示す．デバイスベンダや装置ベンダが専用(プロプライエタリ)に用意するインタフェースも多いが，他のデバイスとの接続性を向上させるために規格化されている著名なインタフェースを中心に紹介する．

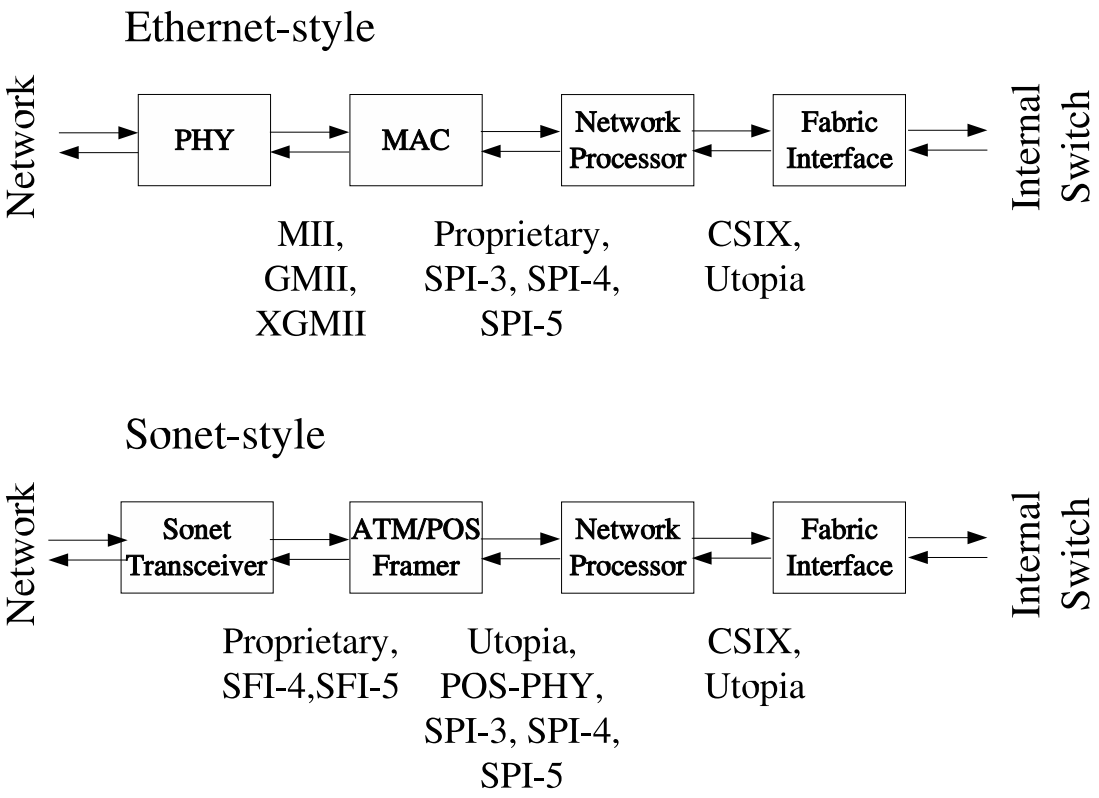

図 D.1: I/Oインタフェースの種類

- **メディアアクセス制御層 (MAC) と物理層 (PHY) の間のインタフェース**
  - **MII**: Media Independent Interfaceの略称．MIIは4bit幅25MHz動作であり，10Mbpsまたは100Mbps Ethernet用のインタフェースである．一般的なMIIは1ポートあたり16ピンが必要であるため，このピン数を減らすための規格としてRMII，SMIIがある．RMII (Reduced MII)は2bit幅50MHz動作であり，MIIよりピン数が半減されている．SMII (Serial MII)はデータ信号と制御信号を組み合わせ，1bit幅125MHz動作で扱って

いる．RMII や SMII は公式な標準インタフェースではないが，多くの PHY ベンダで採用されている．

- **GMII**: Gigabit Media Independent Interface の略称．GMII は，8bit 幅 125MHz 動作であり，Gigabit Ethernet 用のインタフェースである．MII も GMII もクロックとデータ信号を送信用，及び受信用に分離して備えている．GMII にも利用ピン数を削減するための規格である RGMII や SGMII が存在する．また，TBI (10-bit Interface) と呼ばれるインタフェースも存在する．TBI は，8B10B の符号化を搭載している．
- **XGMII**: 10 Gigabit Media Independent Interface の略称．XGMII は，32bit 幅 312.5MHz 動作で 10 Gigabit Ethernet 用のインタフェースである．HSTL (High-speed transceiver logic) 信号を利用している．
- **SFI-4**: SERDES Framer Interface Level4 の略称．SFI-4 は，PHY の中で SerDes (シリアル/パラレル変換を行なう高速シリアルインタフェース) とフレーマを接続するためのインタフェースであり，OC-192 (10Gbps) 回線の接続に利用する．
- **SFI-5**: SERDES Framer Interface Level5 の略称．SFI-5 は，SFI-4 の後継で，2.5Gbps の高速シリアルリンクを 16 リンク並列利用したインタフェースであり，OC-768 (40Gbps) 回線の接続に利用する．

- **ネットワークプロセッサ周辺のインタフェース**

  - **Utopia Level-1**: Universal test and operations PHY interface for ATM の略称．Utopia は，ATM フレーマとネットワークプロセッサの間のインタフェースである．また，ネットワークプロセッサとスイッチファブリックの間のインタフェースとしても利用される．Utopia は 3 レベル (3 バージョン) を定義しており，オリジナルの Level-1 は 8bit 幅 25MHz 動作でクロックとデータ信号を送信用，及び受信用に分離して備えている．最大で OC-3 (156Mbps) までの送受信が可能である．
  - **Utopia Level-2**: Utopia Level-2 は 16bit 幅 50MHz 動作で，OC-12 (625Mbps) までの送受信が可能である．
  - **Utopia Level-3**: Utopia Level-3 は 32bit 幅最大 104MHz 動作で，3.2Gbps までの送受信が可能である．また，Level-1，Level-2 と互換性も備え，OC-12 であれば 8bit で実装可能である．OC-48 (2.5Gbps) の場合は 32bit 全てを利用する．
  - **SPI-3**: Optical Internetworking Forum により制定された System Packet Interface Level 3 の略称であり，OC-48 (2.5Gbps) に利用する．
  - **SPI-4**: Optical Internetworking Forum により制定された System Packet Interface Level 4 の略称であり，OC-192 (10Gbps) SONET や 10Gigabit Ethernet (10Gbps) に利用する．SPI-4.2 (Level 4, Phase 2) が著名．16bit 幅の LVDS (低電圧差動信号) で，311MHz から 322MHz，DDR でラインあたり 622Mbps である [86].
  - **SPI-5**: SPI-4 の後継となる System Packet Interface Level 5 の略称であり，OC-768 (40Gbps) や他の 40Gbps 回線に利用する．16bit 幅のデータラインを踏襲するために，2.5Gbps ～3.125Gbps の高速シリアルリンクを 16 リンク利用する．1.2V の CML (current-mode logic) 信号を利用する．

– **CSIX-L1**: Network Processor Forum により制定された Common Switch Interface Level 1 の略称であり，Utopia にかわるスイッチファブリック用インタフェースである．CSIX-L1 は電気インタフェースと基本フレーム形式を定義している．32/64/96/128bit 幅のデータパスを利用し，最大 250MHz 動作である．例えば，OC-48 (2.5Gbps) は 32bit 幅 100MHz で実装され，OC-192 (10Gbps) は 64bit 幅 200MHz で実装される (フレームヘッダのオーバーヘッドがあるため，必要な回線速度よりも大きな回線容量が必要である). 尚，CSIX-L1 は特定の回線速度のために bit 幅と動作速度の組み合わせを規定していない．このため，同一の回線速度でも異なるベンダによって実装された CSIX-L1 は相互に接続できない可能性があるため，注意が必要である．