平成 18 年 3 月修了
博士（工学）学位論文

(和文題目) データ駆動型ネットワークプロセッサ DDNP の
パケットフィルタリング機構

(英文題目) A Packet Filtering on Data-Driven Network Processor:DDNP

平成 17 年 12 月 16 日

高知工科大学大学院　工学研究科　基盤工学専攻 (基盤工学コース)

学籍番号　1076034

森川　大智

Daichi Morikawa

要 旨

データ駆動型ネットワークプロセッサ DDNP の パケットフィルタリング機構

森川　大智

近年，ネットワークの伝送速度の劇的な向上と，モバイル端末の性能向上に伴い，モバイル・コンピューティングが急激に普及している．さらに将来は，ユーザの周囲に偏在する，モバイル端末，情報家電，センサー等の多くのネットワーク接続可能な機器が相互に協調動作し，ユーザの移動に追従して，周辺環境に応じてダイナミックにネットワークを形成することでより能動的なコンピューティング環境が実現するだろう．そこで，筆者の属する研究室ではエンド端末のプロトコル処理，通信品質保証 (QoS) 制御，ユーザ認証，セキュリティ制御などを担うネットワークプロセッサ (NP) として，低消費電力で，かつ，ソフトウェア実現が可能なデータ駆動アーキテクチャによるデータ駆動型ネットワークプロセッサ (DDNP) を検討している．

本研究では，ネットワーク内の QoS 制御やセキュリティ制御を中央集約的に制御することが困難なネットワーク環境において，各端末が送受信するパケットストリームを検査し，選択的にフィルタリングする機構に着目した．まず，レイヤ 3/4 ヘッダ情報とフィルタリング条件の照合を高速に行うパイプライン並列パケット分類方式を提案する．さらに，ネットワークエンドにおけるヘッダ情報の傾向からフィルタリングルールを最適化することで，ナノプロセッサのオンチップメモリに格納することが出来ることを示す．そして，本提案手法を DDNP を用いてソフトウェア実現するための特徴命令，ないしプロセッサ構成について述べる．最後に，本提案方式を含む，DDNP アーキテクチャのハードウェアレベルのエミュ

レーションを可能とし，性能評価，回路規模の見積もりを行うために評価ボードを試作した．評価ボードを用いた評価の結果，本方式は 6 %程度の回路拡張で約 12MLookup/sec(IPv4) の性能を実現できることを示す．

キーワード　パケットフィルタリング，ネットワークプロセッサ，データ駆動，ファイアウォール

目次

第 1 章　序論　1

第 2 章　パケットフィルタリングとデータ駆動型ネットワークプロセッサ: DDNP　5

2.1　緒言 5

2.2　サラウンディング・コンピューティングの概要 6

2.3　組み込み型パーソナルゲートウェイ 8

2.4　ネットワークセキュリティ確保のためのパケットフィルタリング 10

2.5　結言 14

第 3 章　パイプライン並列パケット分類方式　17

3.1　緒言 17

3.2　パケット分類の定義 18

3.3　既存のパケット分類方式 18

3.3.1　ソフトウェア方式 19

3.3.2　ジオメトリベース方式 20

3.3.3　ヒューリスティック方式 20

3.3.4　ハードウェア方式 21

3.3.5　各方式の分類レートと必要記憶容量比較 22

3.4　ルールの照合法 23

3.5　最長一致検索 25

3.5.1　キャッシュ方式 26

3.5.2　ハードウェアによる完全一致検索方式 26

3.5.3　ツリー方式 26

3.6　検索木の平衡化 31

3.7 分類方式のパイプライン並列化 33
3.8 結言 34

第 4 章 フィルタリングルールの最適化 37

4.1 緒言 37
4.2 フィルタリングルールの傾向 38
4.3 メモリ使用量 41
4.4 結言 43

第 5 章 データ駆動型ネットワークプロセッサ DDNP による実現法 45

5.1 緒言 45
5.2 既存ネットワークプロセッサとデータ駆動型ネットワークプロセッサ:DDNP 46
5.3 データ駆動パケットフォーマット 49
5.4 パケット分類特徴命令 50
5.4.1 LC-Trie 木検索命令 50
5.4.2 プレフィックス検査命令 54
5.5 ナノプロセッサ構成 61
5.6 パケット分類ソフトウェア 62
5.7 結言 64

第 6 章 性能評価 67

6.1 緒言 67
6.2 性能評価システム 67
32bitDDMP チップ 70
FPGA チップ 71
6.3 パケットフィルタリング用回路の回路規模 73
6.4 最長一致検索の性能評価 74

6.5 パケットフィルタリングの性能評価 78
6.5.1 5-tuple パケット分類 78
6.5.2 パーソナルゲートウェイ向けパケットフィルタリング 78
6.6 結言 82

第 7 章 まとめ **85**

謝辞 **89**

参考文献 **91**

付録 A 関連業績 **97**
A.1 査読付き論文誌 97
A.2 国際会議 97
A.3 国内学会 98

第 1 章

序論

米国防総省の ARPAnet に起源を発するインターネットは，1990 年初頭に発明された WWW(World Wide Web) と，1990 年中頃におけるアメリカでの完全商業化をうけて急速に発展した．さらに，同時期におけるパーソナルコンピュータの一般ユーザー層への爆発的な普及と相まって，今日では WWW や電子メールなど原始のものから，リアルタイム系の音声や映像ストリーミング配信，ファイル共有，データベースなどの大容量のデータを取り扱うものや，ネットショッピング，電子商取引などの高度なものまでさまざまなサービスを提供するインフラストラクチャとして，我々の生活におけるあらゆる場面で重要な位置付けを担うまでになっている．さらに近年では，ネットワークの大容量化と平行して，移動体端末によるモバイルコミュニケーションの台頭が著しい．この変化は，これまでのコンピュータのあり方，コンピューティング環境の変化を急速に促すものである．

まず，伝送技術の急速な発展，とりわけ高密度波長分割多重方式 (Dense Wavelength Division Multiplexing:DWDM) などの光伝送技術の発展により，基幹ネットワークの容量は数 Tbps を達成するまでになった．このことは，エンドユーザー間におけるリッチコンテンツの相互交換が容易になる土壌を形成し，それにあわせて新しいサービスが注目されるようになった．例えば，これまでのクライアントモデル/サーバ型のデータ配信モデルとは異なり，Peer-to-Peer 技術によりエンド端末同士を同定し直接的データ配信するモデルが注目を集めている．いずれも，物理的なネットワーク上に仮想ネットワークの一種であるオーバーレイネットワークを構築することで，アプリケーションレベルにおいて既存ネットワークの構造的制約を意識しない方式を採用している．

伝送路に関しては，無線技術の飛躍的進歩によりオフィスネットワークでは

IEEE802.11b[2] に基づく無線 LAN や，より高速化された IEEE802.11a[2] ないし IEEE802.11g[2] に基づく無線 LAN 搭載機が製品化されつつある．このことは，電話網を主体とする階層型インターネット網から，エンド端末同士の無線通信によるアドホック型インターネット網への移行を促す急進的な要因となり得る．アドホック型インターネット網のなかで代表的なマルチホップ無線アドホックネットワークは，基地局やアクセスポイントなどのインフラに依存せず端末のみでネットワークを即座に構成することが可能な技術であり，モバイルコミュニケーションの高度化やユビキタスネットワークの構築手段として注目され，内外で研究開発が活性化している [11][14].

さらに，ネットワークに接続する端末もモバイル端末やセンサー，RFID タグなどの無線デバイスの小型・軽量・低価格化や標準化の進展などにより多様化している．

このように，ネットワーク環境の変化は，近い将来，ユーザが使用する携帯端末，ユーザの周辺環境に偏在する各種デバイスおよびセンサー等が，相互結合することでユビキタス環境が実現する [1]．そのうえ，各々自律的に広大なインターネット上に分散するコンピューティング資源をそれぞれ用途に応じて階層的に利用するネットワークを構成し，ユーザの移動に追従して，周辺環境に分散されたプロセスがダイナミックにネットワーク上を遷移し，環境に応じた機能を提供するサラウンディングコンピューティング環境が実現されるだろう．

これを受けてネットワークノードは，従来のベストエフォート型のトラヒックに加えて，ミッションクリティカルなトラヒックを優先的に扱うなど，異なったサービス品質を提供するためにトラヒックの要求に応じた扱いを行うことが必要となっている．加えて，エンタープライズコンピューティング/ネットワーキングリソースは，より広範囲に多様なサービスを提供するため，悪質な攻撃から保護されなければならない．そのため，ルーティング，ファイアウォール，NAT(Network Address Translation)，帯域監視，帯域制御といったネットワーク品質保証 (Quality of Service:QoS) 機構が不可欠なものとなりつつある．今日では，パケットストリームのレイヤ 3 およびレイヤ 4 のヘッダフィールドを参照するマルチルックアップ処理に加え，負荷調整，ウェブスイッチング，不正進入検知といったネット

ワークアプリケーションにおいてはレイヤ 5 からレイヤ 7 のヘッダフィールドまた，あるいはペイロードによってパケットのより深い解析や分類を行うようになっている．また，サービス拒否攻撃 (以下，DoS 攻撃)，不正侵入等の不正アタックが今日のインターネットにとって深刻な問題となっている．特に，最近ではインターネットからの不正アクセスの危険に加えて，ローカルネットワーク内部からの脅威も増加しており，これまでのネットワーク型のファイアウォールと不正侵入検知装置 (IDS) だけでは対応できなくなっている．各ホスト上で動作する分散型ファイアウォールもあるが，多くはソフトウェアで実現されており，その性能がホストの処理性能に依存するため，複雑化の一途を辿る不正アタックを検知，排除しきれないといった問題もある．すなわち，利用状況によって変化する環境下で利用可能な，コンピューティング資源を選択的に，かつ，ユーザに対して透過的に協調利用できるネットワークを構成するためには，サービスに応じて周辺環境に適切に処理を分散するための帯域，遅延，ジッタ等の要素によるネットワーク QoS，および，安全性，信頼性といったセキュア QoS を満足することが重要な要素となる．

筆者の属する研究室では，エンド端末のプロトコル処理，通信品質保証 (QoS) 制御，ユーザ認証，セキュリティ制御などを担うネットワークプロセッサとして，低消費電力で，かつ，ソフトウェア実現が可能なデータ駆動アーキテクチャに基づく [33] によるデータ駆動型ネットワークプロセッサ (DDNP) を検討している．そして，次世代のネットワークにおけるパケット処理を柔軟に実現するため，データ駆動型ネットワークプロセッサを用いた分散型組み込みパーソナルゲートウェイを提案している [38][39].

本研究では，各種サービスに応じたパケットフィルタリング機構として，パイプライン並列に処理可能な高速パケット分類アルゴリズムを提案し，このアルゴリズムをデータ駆動型プロセッサの特徴であるパイプライン並列処理の徹底的な活用によって高速にソフトウェア実行する方式を提案する．次世代ネットワークにおけるパーソナルゲートウェイなどにも応用しえることを示す.

以下に，本論文の構成，ならびに新しく得られた成果を述べる.

本論文の第 2 章においては，パーソナルゲートウェイの要求条件をネットワーク QoS お

よびセキュア QoS それぞれの観点から述べ，パケットフィルタリングについて示す．

第 3 章においては，ネットワークプロトコル処理におけるレイヤ 3 レイヤ 4 パケットフィルタリングを行うためのパケット分類方式を示す．そして，パケット分類の高速ソフトウェア実現をするために，分類規則の照合にパイプライン並列化した最長一致アルゴリズムを用いる方式を述べる．そして，大規模なルールセットに対しては，ヒューリスティックに分割し，検索空間を限定することで，記憶空間の局所化を行う手法について示す．

第 4 章では，提案パケット分類器のセキュリティシーンにおけるパケットフィルタリングに応用する際のフィルタリング条件を示す．そして，端末に対して流入出するパケットストリームの特性を事前評価することで，効率的なレイヤ 4 パケットフィルタリングを実現できることを示す．さらに，組み込み型パーソナルゲートウェイ上で実現するに当たり，使用メモリ量の縮退，使用プロセッサ数の削減を見込めることを示す．

第 5 章では，本研究の実現基盤となるデータ駆動型ネットワークプロセッサ DDNP について提示する．そして，提案方式をデータ駆動型ネットワークプロセッサ DDNP の機能として実現するにあたり，提案方式の更なる高速化を行うため，データ駆動型ソフト (複合命令) の提案を行い，それらを実装したメモリ参照に特化したナノプロセッサの構成を提案する．

第 6 章では，本研究で提案したパケットフィルタリング機構を実現するための回路規模の評価，および同機構の実証的性能評価を行うために試作した DDNP 評価ボードの構成について述べる．そして，評価ボードを用いた提案方式の性能評価，および提案方式の実装における回路規模の評価を行う．

第 2 章

パケットフィルタリングとデータ駆動型ネットワークプロセッサ: DDNP

2.1 緒言

ネットワークの発展と小型携帯端末の普及に伴い，いわゆる「どこからでもコンピュータを利用できる」という意味でユビキタス・コンピューティングの実現が期待されている．ユビキタスとはラテン語の “ubique ＝あらゆるところで” という形容詞を基にした，「(神のごとく）遍在する」という意味で使われている英語で，ユビキタス・コンピューティングとは Weiser らによって提唱されたもので，ユーザーにとってコンピュータの実体を意識させること無く，「人間の生活環境にコンピュータとネットワークが融合し，ユーザーはその場所や存在を意識することなく利用できるコンピューティング環境」をいう [1].

ユビキタス・コンピューティング環境は，既存のネットワーク上に存在する無数のコンピュータを相互接続し，リソースを共有することで仮想的におおきなコンピューティング環境を構成するグリッド・コンピューティングと無線 LAN や Bluetooth といった無線通信により接続された小型携帯端末によるモバイル・コンピューティングに大別できる．これらのネットワークは既存の階層型ネットワーク網の上に成り立っている．すなわち，端末側の通信手段が無線化されたとしても，それはホットスポットまでのラストワンマイルが無線化されているのみで，その先は既存ネットワークを用いている．将来的には，マルチホップ無線

アドホックネットワークによる，既存ネットワーク網に依存しない局所的なネットワークが無数に形成されるだろう．そして，より個人に特化して局所ネットワークを形成し，個人にあわせたサービスを提供するコンピューティング環境が出来るだろう．筆者はこれをサラウンディング・コンピューティング環境と呼ぶ．

本章では，まず，サラウンディング・コンピューティング環境を実現するためのネットワーク構造の概要について述べる．そして，それを実現するための要素機構としてのパーソナルゲートウェイに要求される機能要素を示す．その上で，セキュア QoS の観点においてパーソナルゲートウェイがフィルタリングすべきデータ，およびそのフィルタリング方法について概説する．

2.2　サラウンディング・コンピューティングの概要

ユビキタス・コンピューティング環境は，いたるところに偏在するいかなるコンピューティング資源を使っても同等のコンピューティング環境が得られることが特徴である．サラウンディング・コンピューティングの特徴は，ユビキタス・コンピューティング環境を，個人ユーザの観点から捉えて，我々の周囲に偏在する携帯電話，家電製品，センサーなどのようなコンピューティングリソースが，自律的に相互通信を行いユーザー個々に応じたサービスを提供するパーソナルエリアネットワークを構築することにある．

この環境は図 2.1 に示すとおり，以下の 3 つの階層構造からなる．

- ネットワーク層

  現在の IP ネットワークトポロジ，あるいはその他の物理的ネットワークトポロジに基づいて構成される．アドホックネットワークにおけるルーティングプロトコル [12][13] もこの層で処理される．

- ミドルウェア層

  現行の静的なネットワーク上に動的なネットワークを実現するフレームワークを提供し，そこに接続する周辺端末を仮想レベルで相互接続する．アプリケーション層は，ミ

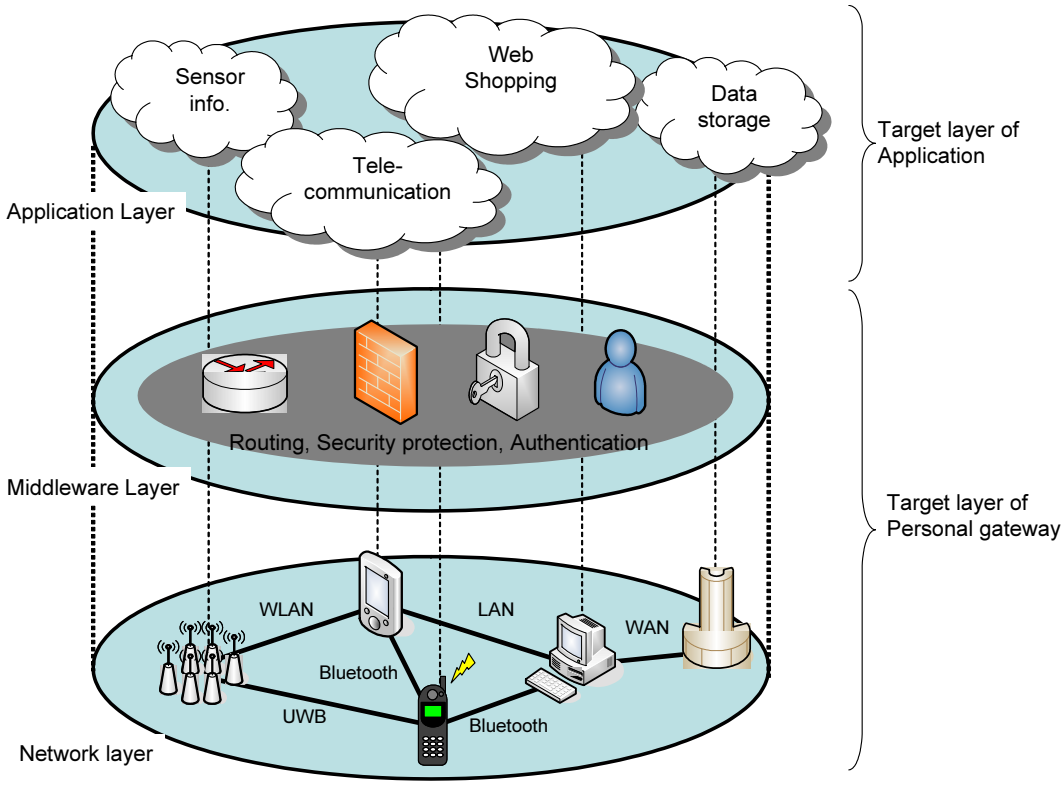

図 2.1　サラウンディング・コンピューティング環境の処理階層

ドルウェア層によって仮想的に結合された周辺端末を用いて種々のサービスを処理するとともに，我々にユーザインターフェイスを提供する．一般に SoftEther などの仮想プライベートネットワーク (VPN) や，gnutella ネットワークや Freenet[10] といった P2P プロトコルによる仮想ネットワークがこの層にあたる．

- アプリケーション層

  ユーザへのサービスを提供する層である．

サラウンディング・コンピューティング環境において，各端末はアプリケーションを実行するための機能を提供するだけではなく，サラウンディング・ネットワークにおけるゲートウェイ機能も兼ねる．言い換えると，すべての周辺端末はパーソナルゲートウェイを内包するといえる．

2.3　組み込み型パーソナルゲートウェイ

パーソナルゲートウェイは，サラウンディングコンピューティング環境におけるエンド端末のネットワークへの出入り口に位置し，エンド端末が用いるネットワーク層，ミドルウェア層，アプリケーション層のネットワーク処理を処理する．以下に，パーソナルゲートウェイにおける 2 つの特徴的機能を示す．

- パケット処理

  パーソナルゲートウェイの基本機能であり，ミドルウェア層で定義されているプロトコル，およびネットワーク層で定義されているプロトコルをオーバーレイ処理しながら，サラウンディングネットワーク上で送受信されるデータを状況に応じて適応的に転送する．ネットワーク層とミドルウェア層のプロトコル設計は独立したものであるが，固定長フィールドで定義されたヘッダを持つデータを 1 単位としたデータ転送という点において，双方の処理における処理過程には類似点が多い．

- セキュリティ保護

  図 2.2 で示すように，ネットワークからの不正アクセス，ネットワーク内部からの不正アクセス，およびエンド端末からネットワークへの意図しない通信を阻止する．ネットワーク層，ミドルウェア層，アプリケーション層にそれぞれ異なった形式のセキュリティ保護がある．主なセキュリティ保護機能として，ネットワーク層およびミドルウェア層におけるパケットフィルタリング，アプリケーション層での認証，暗号化/複合などがある．特に，パケットフィルタリングは不正アクセスを発見，廃棄するための第一歩として重要である．

現行のパーソナルゲートウェイの多くは，ホスト端末上で動作するソフトウェアとして実現されている．しかし，アドホックネットワーク環境下で資料される端末の多くは，(a) 安定した電源を供給できない，(b) 計算に用いる処理資源が限られた，モバイル端末やセンサー類が多数を占めると予想される．このような条件下で，端末本来の処理ではないソフトウェアベースのセキュリティ処理を行うことは厳しい．すでに，モバイルアドホックネット

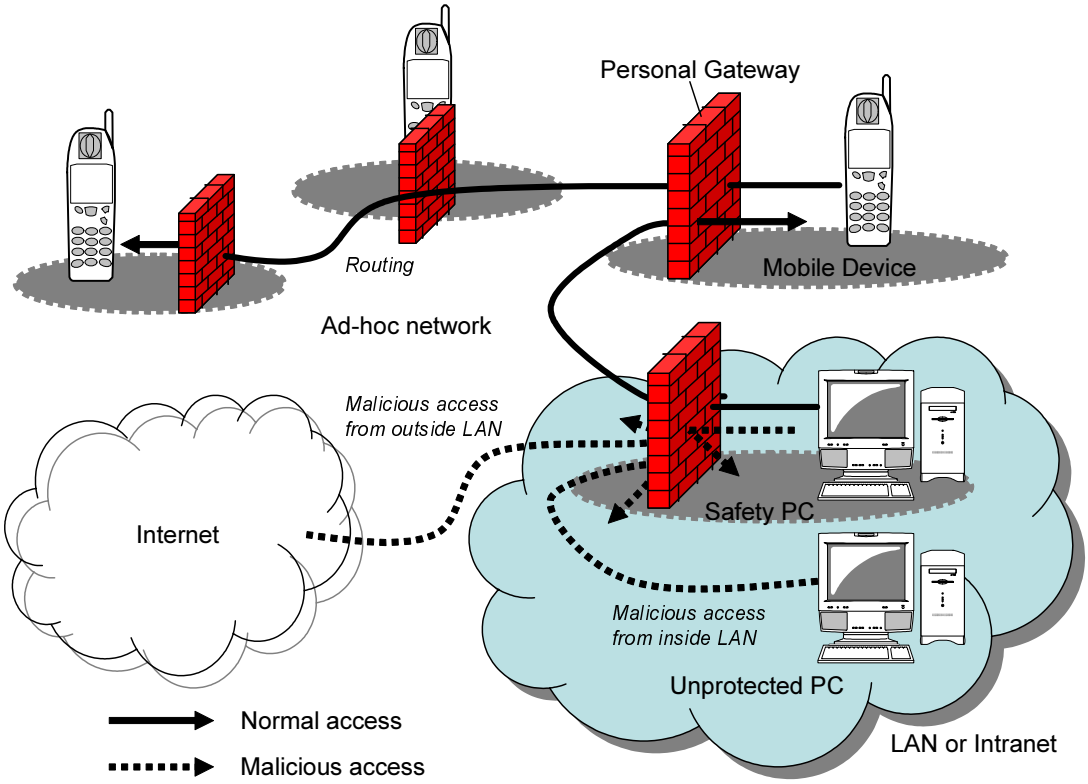

図 2.2　パーソナルゲートウェイと通信の流れ.

ワークに対応したルーティング機構および名前解決プロトコル処理機構を持つハードウェアベースのパーソナルゲートウェイが提案されている [7][8][9]．しかし，これらはいずれもセキュリティ保護について言及していない．そこで，本研究では，筆者の所属する研究室において検討している，パケット処理に加えて，セキュリティ保護を柔軟に実現することを考慮した，データ駆動型原理に基づくデータ駆動型ネットワークプロセッサ（詳細は第 6 章）を用いた組み込み型パーソナルゲートウェイを提案する [38][39].

組み込み型パーソナルゲートウェイのセキュリティ保護に注目すると，組み込み型データ駆動パーソナルゲートウェイのアーキテクチャは図 2.3 で示す構成となる．セキュリティ機能モジュール（図 2.3 中，Security func）は，ネットワーク層，ミドルウェア層，およびアプリケーション層における各プロトコルを対象としたフィルタリング処理を行う．プロトコルスタック処理機能モジュール（図 2.3 中，Protocol processing），あるいは認証モジュール（図 2.3 中，Authentication）は，対象端末の用途，ないしパーソナルゲートウェイを搭載す

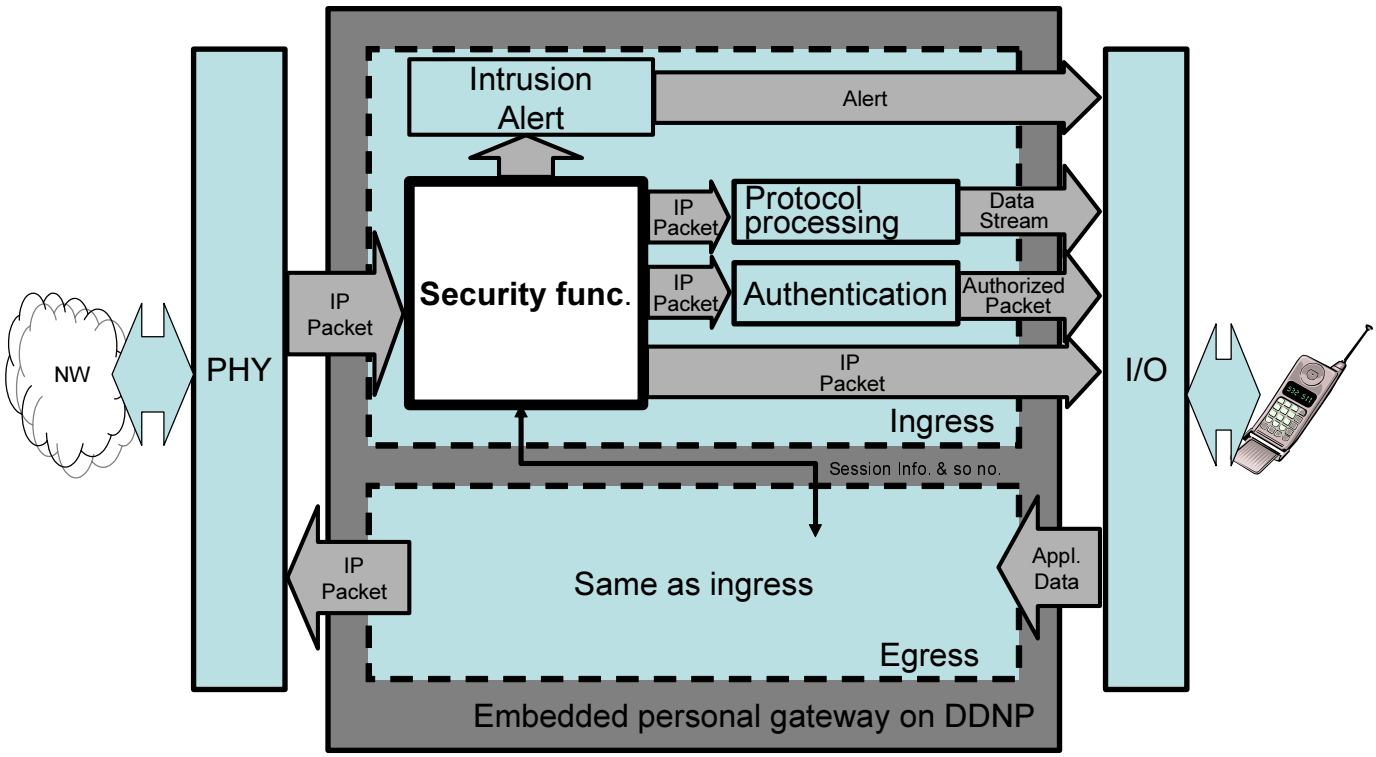

図 2.3　組み込み型パーソナルゲートウェイの機能ブロック図.

るに当たり許容されるハードウェアリソースによって，選択的にセキュリティ機能モジュールの後に付加される．侵入警報（図 2.3 中，Intrusion Alert）とはセキュリティモジュールで検知された悪意のある攻撃を収集して携帯端末に送信する．

特に，フィルタリング処理はプロトコルに応じてさまざまな処理形態があり，プロトコルおよびサービスに応じて適切に割り当てる，あるいは複数の処理を組み合わせる必要がある．

2.4　ネットワークセキュリティ確保のためのパケットフィルタリング

ネットワークを流れるデータは複雑化の一途をたどっており，境界ルータ，家庭内ゲートウェイ，そしてモバイル端末などにおけるネットワーク処理部は，ポリシー・ルーティング，課金などネットワーク QoS に関する要求の他に，DoS 攻撃，不正アクセス，フィッシ

ングなどネットワークセキュリティに関する要求を内包している．すなわち，サービスを柔軟にかつ高速に提供するためには，多様なパケット流を受容したとき，その中から適切なもののみを抽出し，適切な処理に振り分ける機構が重要となる．とりわけ，携帯端末においてはハードウェアリソース，および駆動電源容量の制約から，ネットワークよりデータを受理してから出来る限り早い段階でデータを選別し，処理負荷を軽減することが肝要である．

ネットワークセキュリティの主な装置であるファイアウォールは，インターネットと会社の内部のネットワークの間に設置し，パケットフィルタリングを行うことにより不正なアクセスから会社の内部ネットワークのコンピュータを守ります。企業の場合は，高価なファイアウォールを使用して外部のネットワークから内部のコンピュータを保護する．データの通過を許可・拒否する動作をフィルタリングと呼び，動作する内容によって以下の３つに大別できる．

- パケットフィルタリング

  IP 通信において，宛先や送信元の IP アドレス，ポート番号などを監視し，あらかじめ設定した条件によって，その通信を受け入れる (ACCEPT)，廃棄する (DROP)，拒否する (REJECT) などの動作を行うことで通信を制御する．判断条件には，IP ヘッダにおける宛先 IP，発信元 IP，TCP ヘッダにおける宛先ポート番号，発信元ポート番号，SYN ／ ACK フラグ，IP フラグメントフラグ等が用いられる．図 2.1 は代表的な不正アクセスのフィルタリングに用いられるルールを示したものである．

- ステートフルパケットインスペクション

  IP パケットの内容に応じて，宛先や送信元の IP アドレス，ポート番号などの，接続や遮断といった動作のための条件を動的に変化させ，通信制御を行う．例えば，内部と外部で双方向の通信を行う場合，内部から外部の通信を許可するだけで，その通信への応答に関してのみ，外部からの通信を受け入れる，といった動作を自動的に行う．

- アプリケーションフィルタ

  IP パケットではなく，HTTP や FTP といったアプリケーションプロトコルのレベル

表 2.1　代表的な不正アクセスのフィルタリングに用いられるルールセット.

Illegal Access	IP Src addr	TCP Src port	Protocol	Dir	TCP Flag	ICMP Type	Action
Smurf	not Src IP	*	ICMP	LAN	*	8(0)	Reject
Fraggle	not Src IP	7,13,19,21	UDP,ICMP	LAN	SYN	8(0)	Reject
SYN Flood	not Src IP	*	TCP	LAN	SYN	*	Accept
Land attack	equals Dst IP	*	TCP	LAN	*	*	Reject
IP spoofing	not Src IP	*	*	LAN	*	*	Accept
Open tear	not Src IP	*	UDP	LAN	*	*	Accept
SYN scan	*	*	TCP	WAN	RST	*	Accept
FIN scan	*	*	TCP	WAN	RST	*	Accept

で外部との通信を代替し，通信内容を精査して制御する．例えば，HTTP リクエストのメッセージ・ヘッダ部分に含まれる情報から通信の可否を決定する URL フィルタリングでは，HTTP メッセージは表 2.2 に示すように，複数行から成り立つ一連の文字列を精査する．ここにはリクエストによって取得するべき HTML のロケーションを表すリクエスト URI が含まれる．

図 2.4 は，同組み込み型パーソナルゲートウェイにおけるセキュリティ機能モジュールの機能要素とデータフローの一例を示したものである．

IP De-fragment モジュールは，ネットワークのネットワーク層が IPv4 で運用されているときで，かつ，IP パケットが何らかの理由によりフラグメントされている場合に，元の IP パケットに再構築するものである [44]．IP デフラグメントは，ネットワークの中継地点

表 2.2　HTTP プロトコル GET メソッドのメッセージ例.

メッセージ・ヘッダ	リクエスト・ライン	GET /students/index.html HTTP/1.1
	リクエスト・ヘッダフィールド	Accept: */* Accept-Language: ja Accept-Eucoding: gzip, deflate Referer: http://www.kut.co.jp/link/ User-Agent:Mozilla/4.0 (compatible; MSIE5.01; Windows NT 5.0) Host: www.kut.co.jp
	一般ヘッダフィールド エンティティヘッダフィールド	Pragma: no-cache
	その他	Cookie:Apache=200.199.199.199. 4032966456721350
空行 (CR+LF)		空行 (CR(0x0d)+LF(0x0a) のみの行)
メッセージ・ボディ		

の MTU が，送信元の MTU を下回った場合に，データを細分化して転送することであり，正常な動作である．一方で，その振る舞いを模倣したオーバーラッピングフラグメント攻撃や，タイニーフラグメント攻撃といった DoS 攻撃がある．これらの攻撃は，パケット単体でのパケットフィルタリングを巧みに交わし，デフラグメント後に悪意のあるデータを再現する攻撃である．そのため，本モジュールにおいてあらかじめデフラグメントを行い，後段のフィルタリングモジュールでのフィルタリング抜けを回避する．デフラグメントを通過したパケットストリームの TCP/IP ヘッダ部分は次段の処理にまわされ，ペイロード部分はペイロードバッファに格納される．

Packet Classification は，パケットのレイヤ 3/4 のヘッダ情報を基に，表 2.1 の様なあらかじめ定められた条件と照合し，該当する条件にあったアクションを決定する．また，SPI(Stateful Packet Inspection) は，TCP のコネクション情報を管理し，不正なセッショ

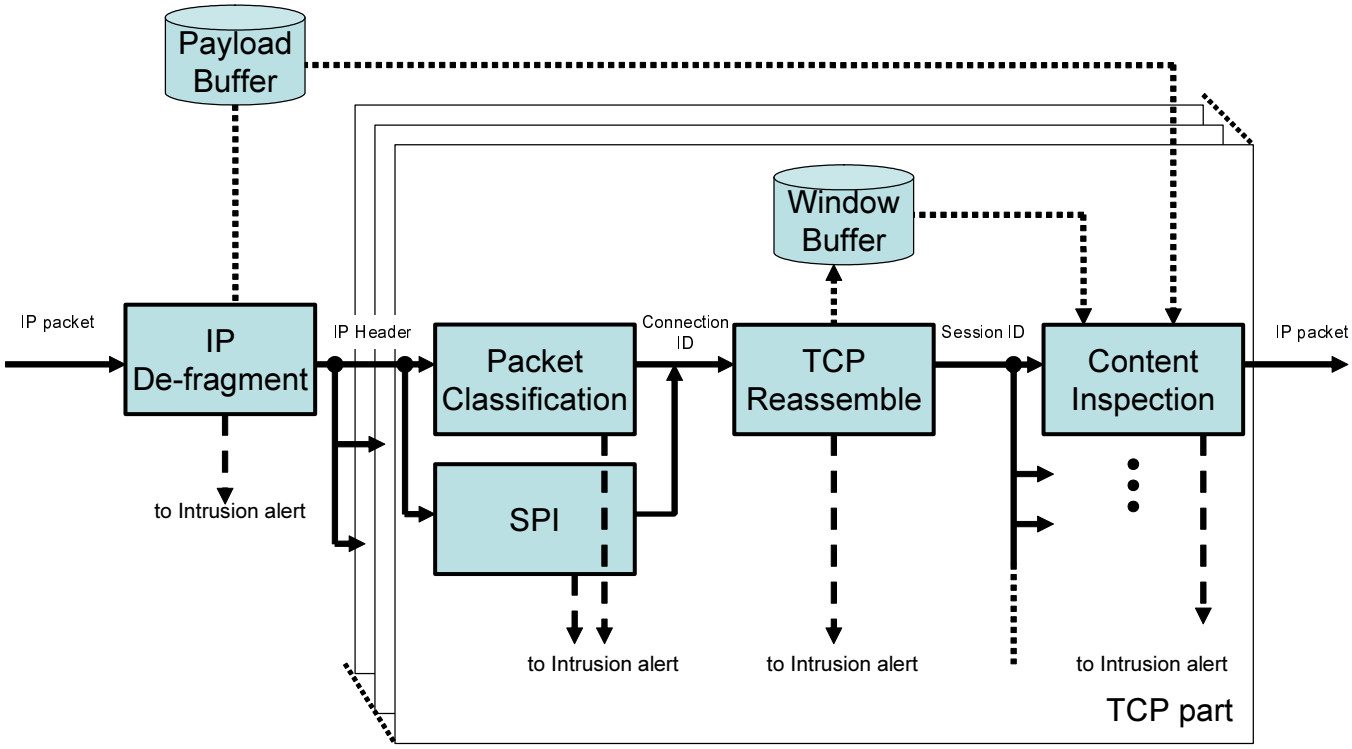

図 2.4　セキュリティモジュールの機能要素とデータフロー.

ンを廃棄する [43]．これらは，それぞれ静的パケットフィルタリング，および動的パケットフィルタリングとして相補的に動作する．

Content Inspection はレイヤ 7 のデータをフィルタリングするものである [45]．本モジュールは，レイヤ 7 のデータに対して文字列検索を行い，フィルタリング条件に一致するものをフィルタリングする．

このような構成において，各モジュール間のメッセージを non-strict データフローにすることで，各モジュールにおける処理遅延，特に処理負荷の高いアプリケーション層のコンテンツフィルタリングの処理遅延を極小化できると考えられる [39].

2.5　結言

本章では，サラウンディング・コンピューティング環境を実現するためのネットワーク構造の概要について述べた．将来的には，マルチホップ無線アドホックネットワークによる，

2.5 結言

既存ネットワーク網に依存しない局所的なネットワークが無数に形成され，より個人に特化して局所ネットワークを形成し，個人にあわせたサービスを提供するコンピューティング環境が出来るだろう．次に，そうした環境を実現するための要素機構としてのパーソナルゲートウェイの必要性と，要求される機能要素を示した．その上で，ネットワークセキュリティとパケットフィルタリングの観点においてパーソナルゲートウェイがフィルタリングすべきデータ，およびそのフィルタリング方法について概説した．また，DDNP を用いたパーソナルゲートウェイを実現するときのデータフローの一例を示し，本研究におけるパケットフィルタリング機構でレイヤ 3/4 パケット分類の位置づけを行った．

第 3 章

パイプライン並列パケット分類方式

3.1 緒言

前章で述べたとおり，レイヤ 3/4 パケットフィルタリングは，OSI 参照モデルのネットワーク層およびトランスポート層におけるパケットのヘッダ情報を参照することで，パケットの通過／廃棄を決定する．この仕組みをパケット分類という．

本章では，パケット分類がルーティングテーブルのルックアップをはじめ，DiffServ による QoS 制御などの前段処理として中心的役割を担っており，ユーザの要求に応じて異なるサービスを提供するために欠かせない処理であることを示し，さまざまなフィルタリング対象にスケーラブルに対応するためには，トレードオフの関係にある処理負荷とメモリ容量のバランスを柔軟に平滑化できる方式が必要であることを述べる．そして，検索空間の増大による分類レートの急激な低下を防ぐため，パイプライン並列処理による実現法が有効であることを示す．このとき，分類レートと記憶空間のトレードオフ，異なるフィールド長や種別に対する柔軟性を得るため，以下の手法を採用する．

- ルールセットの大規模化に伴うルールの偏りに対する検索木のバランスを確保するため，Index Jump テーブル [25] を導入し，検索木を分割する．
- 最長一致検索アルゴリズムとして，メモリ参照数の極小化による検索遅延の削減，かつパイプライン並列化が容易な LC(Level Compress)-Trie 木構造を用いる．

以上のことより，各種サービスに応じたパケット分類をパイプライン並列に処理可能な高速アルゴリズムを提案し，このアルゴリズムをデータ駆動型プロセッサの特徴であるパイプラ

イン並列処理の徹底的な活用によって高速にソフトウェア実行する方式を提案する．

3.2　パケット分類の定義

IP パケットのヘッダに含まれているプロトコル，送信先 IP アドレス，発信元 IP アドレス，その他オプションの情報を読みとって，ユーザーが指定したルールにのっとってパケットを通過／遮断する．

サービス毎の要求は，ルールセットで示される．ルールセット内のルールは，フィールドに応じた規則を示しており，それら総てを Π で表すと，

$$\Pi = (\pi_1, \pi_2, ..., \pi_n)$$

であり，各ルールは，分類に必要なフィールド群と，ポリシーで表される．フィールド群を F，ルールを γ とすると，

$$\gamma = (f, \pi) \ \ \{f \subseteq F \ \ \& \ \ \pi \in \Pi\}$$

となる．このとき，ルールセット C は，

$$C = ((f_1, \pi_1), (f_2, \pi_2), ..., (f_n, \pi_n))$$

$$\bigcup_{i \leq n} f_i = F$$
$$f_i \cap f_j = \phi, \ \ j \ with \ i \neq j$$

である．総てのパケットフィルタリング方式は，この定義に従い分類処理を行う．

3.3　既存のパケット分類方式

n 個のフィールドを対象とした n 次元パケット分類では，ルールセットの大規模化に伴い計算量が激増する．

既存のパケット分類方式は，表 3.1 示すとおり 4 つに大別できる [17]．これらの分類方式を以下の観点から述べ，問題点を明らかにする．

- 分類速度

- 記憶容量
- 扱えるフィールド数
- 実装の容易さ

表 3.1 パケット分類方式の種類.

Category	Algorithm
Basic data structure	Linear search, caching, hierarchical tries, set-prunning tries
Geometry based	Grid-of-tries, AQT, FIS
Heuristic	RFC, hierarchical cuttings, tuple-space search
Hardware only	Ternary CAM, bitmap-intersection

3.3.1 ソフトウェア方式

最も単純な分類アルゴリズムは，各ルールを線形に検索する Linear Search である．本アルゴリズムが必要とする記憶空間は，ルールセットをそのままメモリ構造に写像するため非常に効果的に使用できる．さらに，ルールセットをそのまま用いるため，検索に用いるデータ構造は単純そのものであり，容易に変更可能である．しかし，ルールセットが大規模化すると，分類レートも比例して長くなるという問題点がある．

そこで，分類レートを削減する方式として，一般に Trie 構造が用いられる．図 3.1 は，Hierarchical Tries アルゴリズムにおける検索木の例を示している．本方式は，各ルールを BinaryTree で表現し，各ルールの検索木を逐次検索していく．このとき，分類レートは，ルールセットのサイズに依存せず，分類に用いるフィールド数とフィールドのデータ幅に依存する．また，記憶容量は主にルール数，フィールド数，各フィールドの幅に支配される．

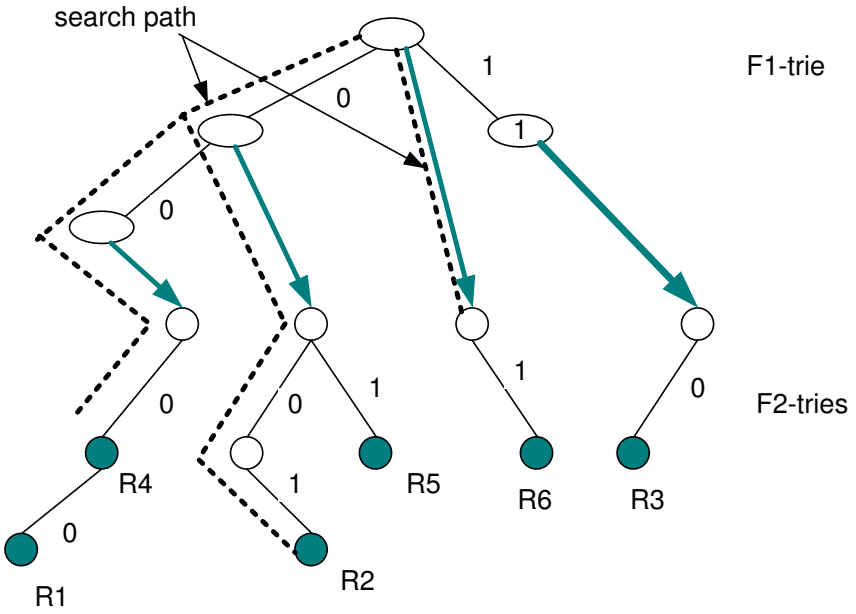

図 3.1　Hierarchical Trie

3.3.2　ジオメトリベース方式

ルールセットを n 次元平面に写像することで, n 次元点位置決定問題として扱う [18][19].. 図 3.2 に示す 2 次元点位置決定問題では，図中の点 P の座標が与えられたとき，垢締め定められた領域のどの領域に P が含まれるかを求める問題である．P の座標をパケットのフィールド値，領域をルールセットの定める値域と対応付けられる．本方式において，分類レートと記憶空間は完全なトレードオフの関係にある．ルール数 n，フィールド数 F のとき，$O(\log n)$ 分類レートを達成する場合，記憶空間は $O(n^F)$ 必要であり，記憶空間を $O(n)$ とした場合は，$O(log^{F-1}\ n)$ の分類レートが必要となる．そのため，ルール数が 1000 でフィールド数が 3 の場合，1GB の記憶容量が必要になる，あいは，100 回のメモリアクセスが必要となり，現実的ではない.

3.3.3　ヒューリスティック方式

図 3.3 に示す，Recursive Flow Classification アルゴリズム [20] は，各フィールドを細かく分割し，分割したデータを基に並列に段階的に検索を進めていく方式である．各フィール

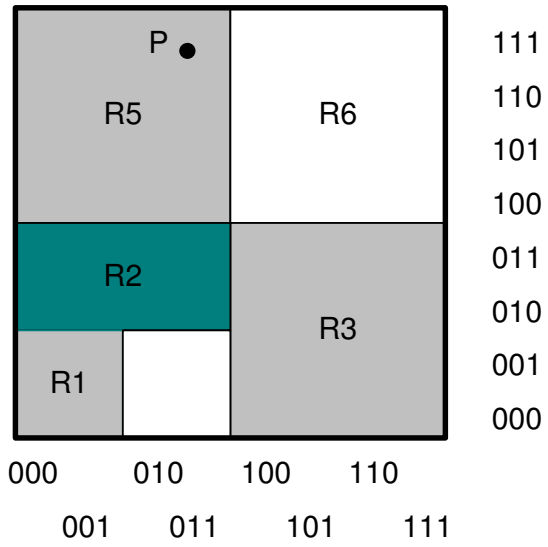

図 3.2　ジオメトリベース

ドは現実のルールセットを基にヒューリスティックに分割する．分類レートは次元数にのみ依存し，n 次元のルールセットにも実現的な性能を有する．ただし，多大な記憶空間を必要とする [21].

3.3.4　ハードウェア方式

Bitmap-intersection[16] は，ハードウェア実装に最適化された方式であり，小規模なルールセットにおいては，高速な分類処理と少ない記憶空間で実現可能な方式である．しかし，ルールセットが大規模化すると，ルールセットのサイズに応じて必要な記憶空間が急増し，メモリバンド幅も線形に増加してしまう．

もうひとつの代表的な方式として，連想メモリを用いたハードウェアデバイスである TCAM(Ternay Content Addressable Memory) がある [22]．TCAM のセルは 0，1，および*を保持することが可能である．*は don't care を表し，ワイルドカードを含むルールの一致処理を TCAM が実現するためにマスクとして用いられる．TCAM は連想メモリを用いることにより，非常に優れた検索性能を実現しており，パケット分類に理想的であるように見える．現在のコア／エッジノード向けのネットワーク装置の多くは，パケットフィルタリングに TCAM を用いて高速化を図っている．しかし，ハードウェア資源の限られたモバ

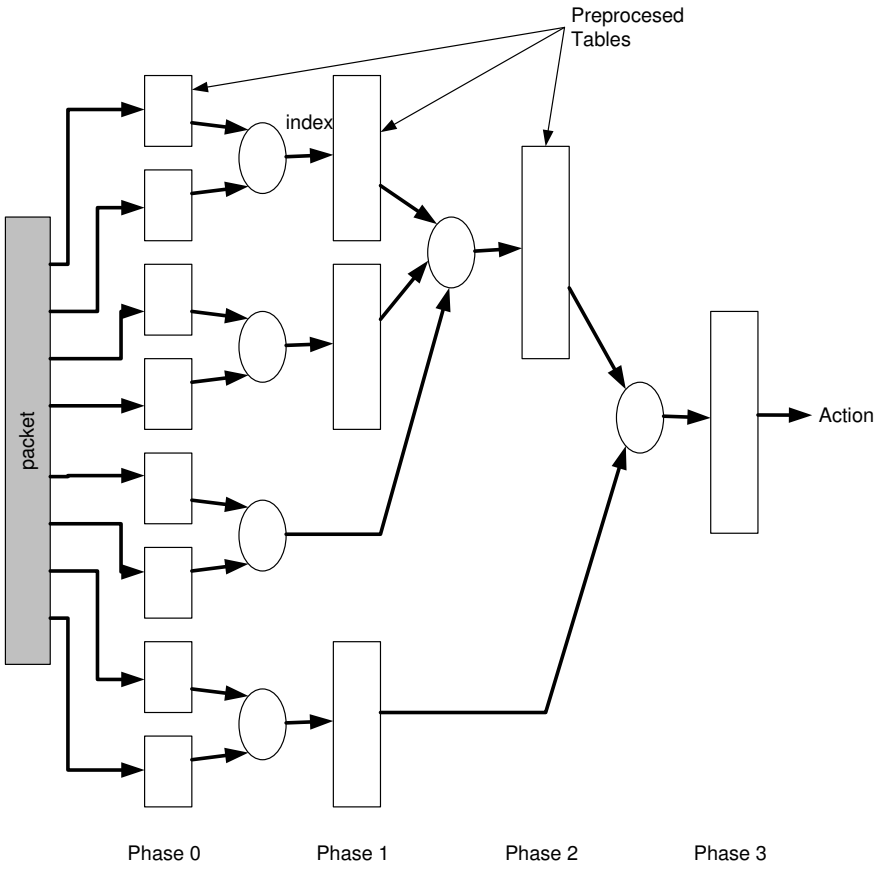

図 3.3　Recursive Flow Classification

イル端末においては，CAM は回路コストが高い上，ルールセットの大規模化に柔軟に対応できない．さらに，単位面積当たりの電力消費が大きいという問題がある．そのため，今後半導体技術の向上により，高性能な TCAM が実現するかもしれないが，依然として特異なデータ構造を操作する効率的なアルゴリズムによる解決法を用いる必要性がある．

3.3.5　各方式の分類レートと必要記憶容量比較

表 3.2 より，いずれの方式も，主に分類レートはルール数に依存し，記憶容量はフィールド数に依存する．扱える次元数が少ないものは，静的な低次レイヤのパケット分類なら性能を達成できるが，今後の複雑化する高次レイヤの分類には対応できない．また，次元数の制約のない分類方式においては，各フィールドの部分照合を並列に実行し, 徐々に探索空間を縮小しながら検索する再帰的分類を行う方式が多く見られる．このとき，分類レートを向上

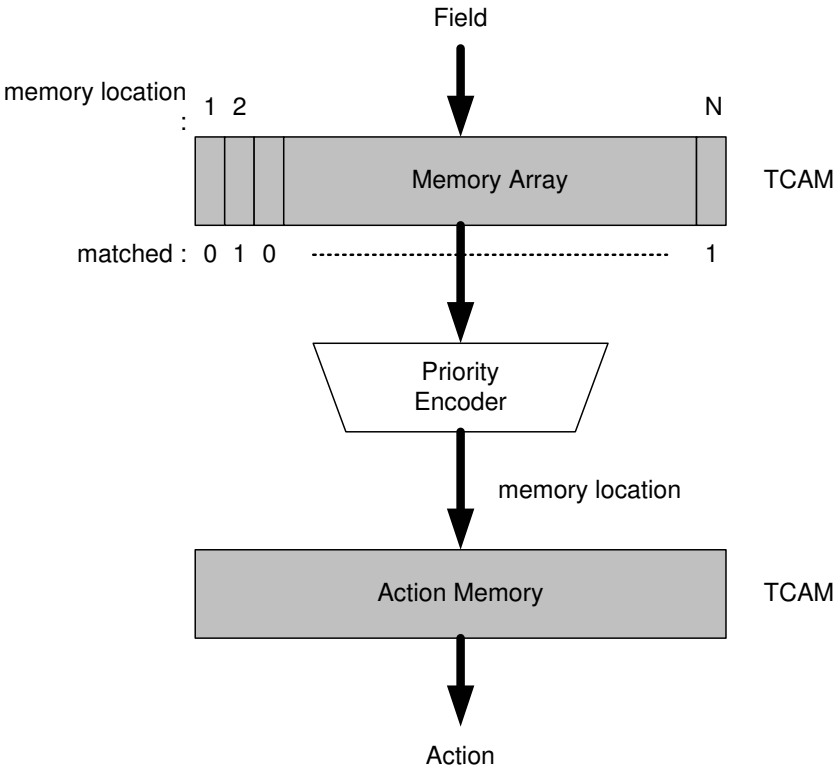

図 3.4　TCAM

させる反面，検索データの記憶量が爆発的に増大する．仮に記憶空間を節約すれば，限定的な次元数の分類しかできなくなる問題点がある．

以上のことより，パケット分類の高速ソフトウェア実現においては，

- パイプライン並列化可能
- 記憶空間の極小化
- 上記の要求を簡便な処理の複合的な利用により実現可能
- 簡単なハードウェア機構により実現可能

が重要になるといえる．

3.4　ルールの照合法

パケット分類において, ユーザが定義可能な照合法は以下の 3 通りである．

表 3.2　各方式の分類レートと必要記憶容量

Algorithm	Time	Storage
Linear search	N	N
Hierarchical trie	W^d	NdW
Set-pruning trie	dW	N^1
Grid-of-tries	W	NW
AQT	aW	NW
FIS-tree	$(l+1)W$	$1 \times N^{(l+1/l)}$
RFC	d	N^d
HI-Cuttings	d	N^d
Tuple-space search	M	N
Ternary CAM	1	N
Bitmap-intersection	$dW + N/memwidth$	dN^2

N-Number of Rules W-Width of dimension

d-Number of dimensions a-Tunable integer parameter

- プレフィックス一致
- 完全一致
- 範囲一致

プレフィックス一致は，IP アドレスのマスク領域の判定に用いられる．

完全一致は，プロトコルの同定や特定フラグビットの状態を表現するために用いられる．完全一致照合は照合するルールと同一の長さのプレフィックスとの照合とみなせるため，プレフィックス照合の特殊な場合として取り扱える．

そして，範囲一致は主に TCP ポート番号を対象として用いられる．例えば，ネットワークから取得した TCP/IP パケットの TCP 宛先ポートを x としたとき，x に対して

表 3.3　範囲指定条件のプレフィックス条件化.

Range	Port#	Binary string	Prefix expression
$135 \leq x \leq 140$	136	0000000010001000	0000000010001*
	137	0000000010001001	
	138	0000000010001010	0000000010001100
	139	0000000010001011	
	140	0000000010001100	
$140 \leq x \leq 144$	141	0000000010001101	0000000010001101
	142	0000000010001110	000000001000111*
	143	0000000010001111	

$136 \leq x \leq 140$ という範囲一致条件は，パケットの TCP 宛先ポートが 136 より大きくて 140 より小さい場合に，パケットはこの条件に一致したといえる．このとき，範囲一致条件 $136 \leq x \leq 140$ は，表 3.3 で示すとおり (00000000100001$*$) および (0000000010001100) というプレフィックスで表現できる．

照合法をプレフィックス一致に限定することにより，アルゴリズムの一元化が可能となり，計算コストの削減につながる．また，照合法の置換により，場合によっては照合データの表現が冗長になるが，照合データを統一することにより，マクロに照合データの圧縮が可能となり，結果として記憶コストの抑制につながる [26].

3.5　最長一致検索

パケット分類における分類条件の照合法としてプレフィックス一致照合をおこなうにあたり，最長一致検索アルゴリズムを適用する．既存の最長一致検索方式は，大きく分けると

- キャッシュによる検索方式
- ハードウェアによる完全一致検索方式

- ツリー構造のデータ構造を用いた検索方式

の 3 つが挙げられる．

3.5.1　キャッシュ方式

高速な汎用プロセッサのキャッシュメモリにルックアップテーブルをコンパクトに格納する方法である [35][36][37]．これらの方法は，平均検索時間としては，マルチギガビットのリンク速度に近づいているが， 検索のキーとしさらに，IP アドレス検索などで使用する場合，現在普及している IPv4 におけるプレフィックス分布を元にキャッシングを行っているため，動的なトラヒックの分布への適応をが困難である．さらに，プレフィックス分布にの変化にも対応する必要があり適応性が低い．

3.5.2　ハードウェアによる完全一致検索方式

ハードウェアによる最長一致検索は，キャッシュ方式同様プレフィックス分布における 24bit 長のプレフィックスが多いということを活用し，プレフィックスの完全一致を行っている．これにより，高速な検索とメモリ要領の削減を達成できるが，プレフィックス分布に依存したプロセッサ構成になってしまうという問題点がある．

3.5.3　ツリー方式

最長一致検索には，ツリー型のデータ構造によりルックアップテーブルを表現し，検査 bit に基づいて親ノードから小ノードへ分岐をし，検索を行う方式がある [27][28][29]．しかし，この検索方法では，ツリーをたどるときの枝分かれ毎に，比較判定と，時間のかかるメモリアクセスが必要なため，検索時間がかかる．また，この方式は汎用プロセッサ上でソフトウェアとして実行されるため，専用ハードウェアによる実現法ほど高速化が望めない．

ルーティングテーブルを Trie 構造に写像すると図 3.5 のようになるが，これではノード数が大きくなり検索にかかる平均深度 (ルートからリーフまでのパスの平均長) が長くなる．

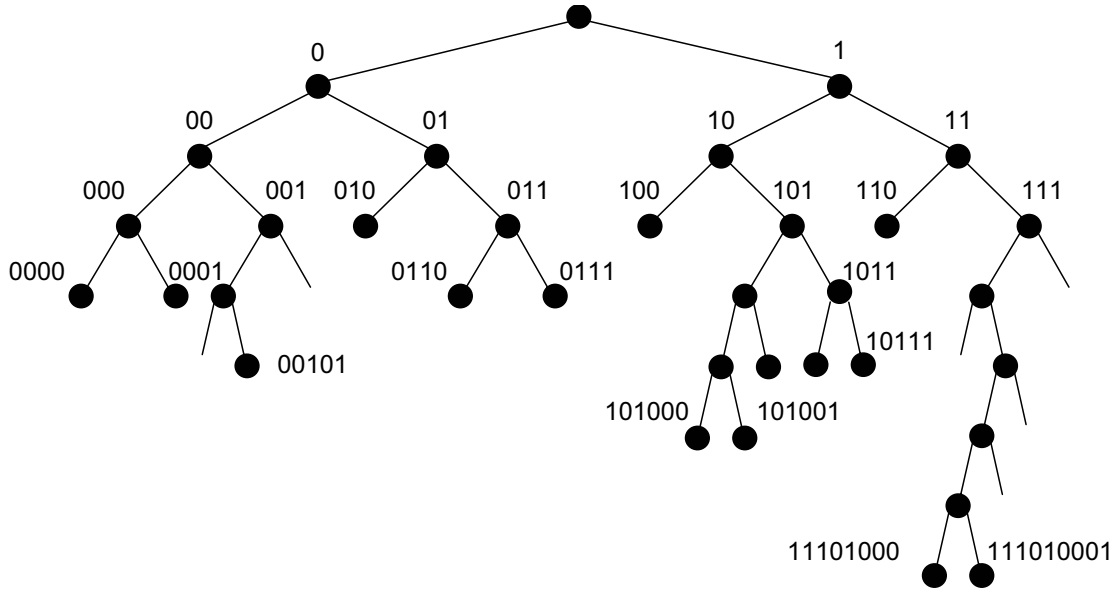

図 3.5　Binary Tree.

この問題を解決するために現在用いられている手法の一つが Path Compress 法であるである．この手法は，Binary Tree における子ノードが一つしかない中間ノードを削除し，パス検索に必要ないビットの長さをノードに保持することでパス長を縮小する方法であり，一般に Patricia Tree と呼ばれる．図 3.5 の Trie を Pass Compress することによって生成された Patricia Tree を図 3.6 に示す．Patricia Tree では，木構造におけるリーフの数を n としたとき，総ノード数は必ず $(2n-1)$ になる．この統計的な特徴はよく知られている．しかし，ツリー構造に写像する対象データの長さが長くなった場合に，Path Compress 法ではデータ数に比例して深さ方向にノード数が増え，平均深度の増加を抑制することができず，メモリアクセス回数，およびメモリ使用量の削減には至らない．

この問題を解決する方法が Level Compress 法である．Level Compress 法は各ノードが 2^k の子ノードを保持する MultiWay ツリー構造をとることでツリーを深さ方向にさらに圧縮する方式であり，

- 検索木の深さはフォワーディングテーブルのエントリ数に対し $O(\log\log n)$ で増加する．
- 検索木の深さはアドレス長と独立しているため，大きなプレフィックスにも対応可能．

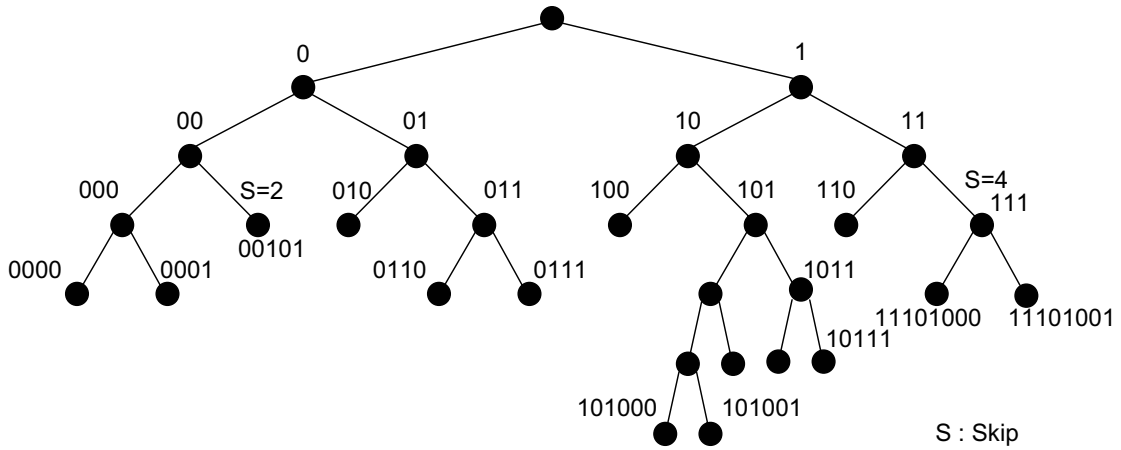

図 3.6　Patricia Tree.

という特徴を持つ．これより得られた LC(Level Compressed)-Trie は平均深度の劇的な削減を可能とし，ツリー検索に必要なメモリアクセス回数の削減およびメモリ使用量の削減が望める．

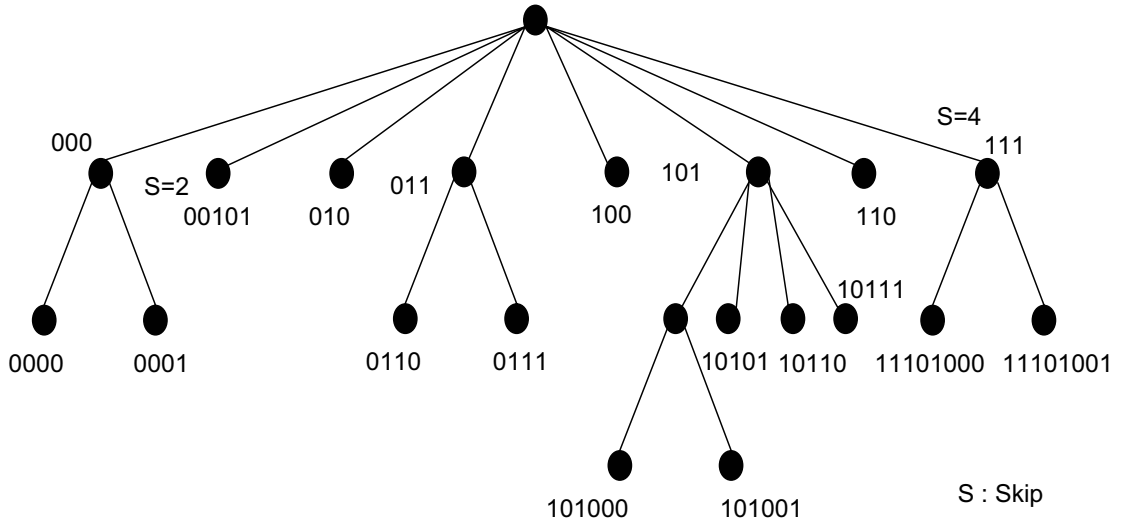

図 3.7　Level Compress Trie.

LC-Trie の各ノードは検索処理を可能とするため，以下のデータを持つ．

- Branch： 値が k のとき，ノードは 2^k 個の子ノードを持つことを示す．また，$k = 0$ のときは，ノードが終端 (リーフ) であることを示す．
- Skip： Path Compress を行った際に省かれた深さ方向のノード数を示す．検索時に検

索 bit を Skip 分だけ飛ばす.

- Pointer： ノードが持つ一つの子ノードへのポインタである．図 3.7 の LC-Trie 木を想定した場合，ポインタはノードが持つ子ノードの左端を指す．リーフの場合は，プレフィックスを格納しているテーブルへのポインタを返す.

図 3.7 のデータ構造を配列表現を表 3.4 に示す．また，そのときの BaseVector を表 3.5 に示す.

表 3.4　LC-Trie 木の配列表現

	Branch	Skip	Pointer
0	3	0	1
1	1	0	9
2	0	2	2
3	0	0	3
4	1	0	11
5	0	0	6
6	2	0	13
7	0	0	12
8	1	4	17
9	0	0	0
10	0	0	1
11	0	0	4
12	0	0	5
13	1	0	19
14	0	0	9
15	0	0	10
16	0	0	11
17	0	0	13
18	0	0	14
19	0	0	7
20	0	0	8

表 3.5　プレフィックスの配列表現

nbr	string
0	0000
1	0001
2	00101
3	010
4	0110
5	0111
6	100
7	101000
8	101001
9	10101
10	10110
11	10111
12	110
13	11101000
14	11101001

3.6　検索木の平衡化

パケット分類のためのルールセットの大規模化に伴って，ルールに偏りがあると，検索木が不均質になる．LC-Trie 木を図 3.8 に示すように Index Jump (Index Jump) テーブルに格納することで，サブツリーおよび各次元のバランスを改善できる．さらに，検索が進むにつれて探索範囲を徐々に限定でき, 記憶空間を最小限に抑えられる.

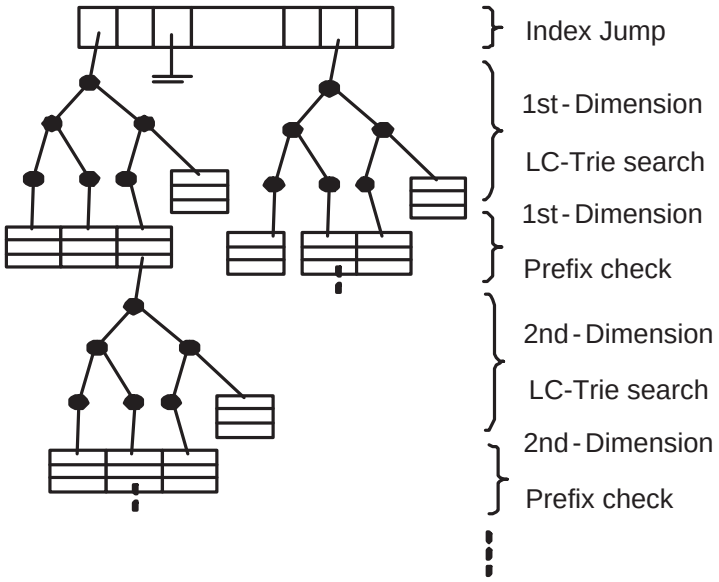

図 3.8　パケット分類処理の分類木構造.

検索木を分割する際，および Index Jump テーブルを参照する際には，図 3.9 に示す計算式を用いて，各キーフィールドの先頭 x ビットを取り出し，これらのビットを連結した値を Index として用いる.

このとき，Index Jump テーブルのエントリ数は，パケット分類に用いるフィールドのビット長 $x_i(1 \leqq i \leqq n)$ により決まり，$O(2x_i)$ で線形に増加するため，適切なビット選択が必要である．平均的に分散したルールセットの場合，検索木は選択ビット数にかかわらず，Index Jump テーブルにほぼ均等に割り当てられる．しかし，実際のルールセットではかなりの偏りが予想される．実際に，Mae-West のルーティングテーブル [30] から 4 次元のルールセットを作成して調べたところ，Index Jump テーブルのエントリ数が増加しても，フィルタ群が特定エントリに集中する傾向が見られた．このことから，ルールセットの分布に応

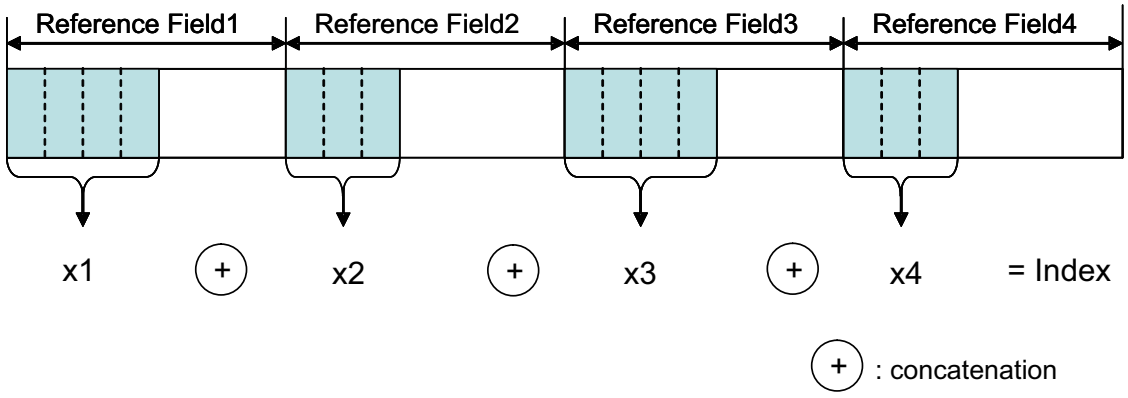

図 3.9　Index Jump に用いる Index の生成法.

じて，Index Jump テーブルに必要最小限のリソース消費で実現可能なビット数をヒューリスティックに設定する必要があると言える．

例えば，生成したルールセットの先頭 2 つのキーフィールドを宛先/送り元 IP アドレスとし，残りを宛先/送り元ポート番号と仮定したとき，各アドレスフィールドに 4 ビット，各ポートフィールドに 2 ビットを適用し Index Jump テーブルを生成する．すると，ルール群に対して，最大サイズとなる検索木でも，分割前の約 10 分の 1 となり，実用的な検索木が生成できることを確認した．また，ローカルサイトやホームゲートウェイなどの比較的小規模な網においては大多数のフィルタの先頭ビットが同じ値を示すことが多い．これでは Index 生成用の選択ビットとして機能せず，Index Jump テーブル内の有効データ数が少なくなり，Index Jump の効果が薄れる．このような場合には，偏りのあるフィールド条件を 2 つのサブフィールド条件に分割する．つまり，次元を $(N+1)$ に拡張することで，Index Jump テーブルによる分散効果を大きくできる．

また，ローカルサイトやホームゲートウェイなどの比較的小規模な網においては大多数のフィルタの選択ビットがすべて同じ値を示すことが多い．これでは Index Jump テーブル内の有効データ数が少なくなり，Index Jump の効果が薄れる．このような場合には，図 3.10 のように，偏りのあるフィールド条件を 2 つのサブフィールド条件に分割する．つまり次元を（N+1）に拡張すれば，Index Jump テーブルの分散を大きくできると考えられる．

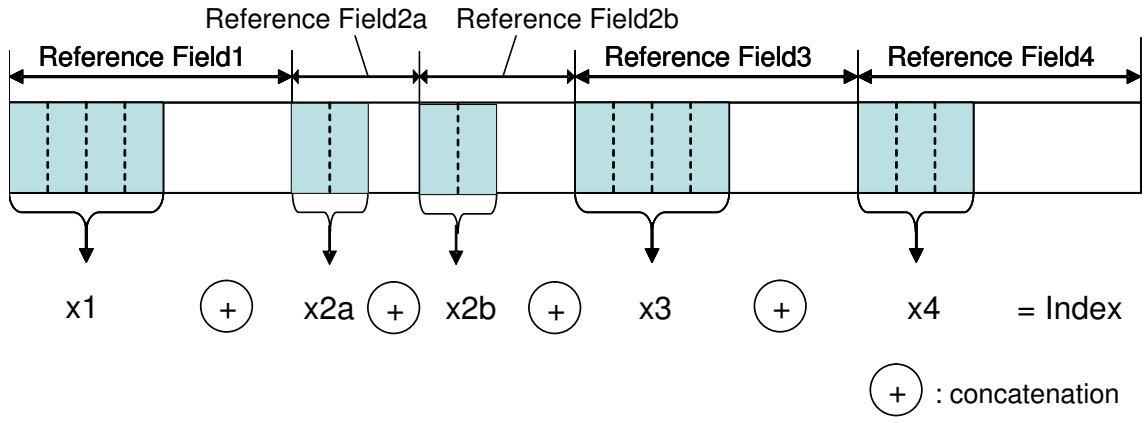

図 3.10　次元数の拡張.

3.7　分類方式のパイプライン並列化

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

1. Index Jump 部入力パケットのフィールドを基に本テーブルを参照し，該当する検索木へのポインタを出力する. Index Jump に使用されるアドレスは前節で述べたように, 各フィールドの先頭 x ビットを取り出し，これらのビットを連結した値である.
2. フィールド検索部 Index Jump 部において選択された検索木を検索する部分であり, LC-Trie 検索アルゴリズムを用いている. これによって，検索木を深さ方向に圧縮して，平均メモリアクセス回数を極小化できる. 図 3.11 に示すように，本モジュールは LC-Trie 木を探索する LC-Trie 検索部と，木探索の結果を検査する Prefix 検査部からなる. これらをパイプライン並列に次元数だけ繰り返して，検索処理を行う.

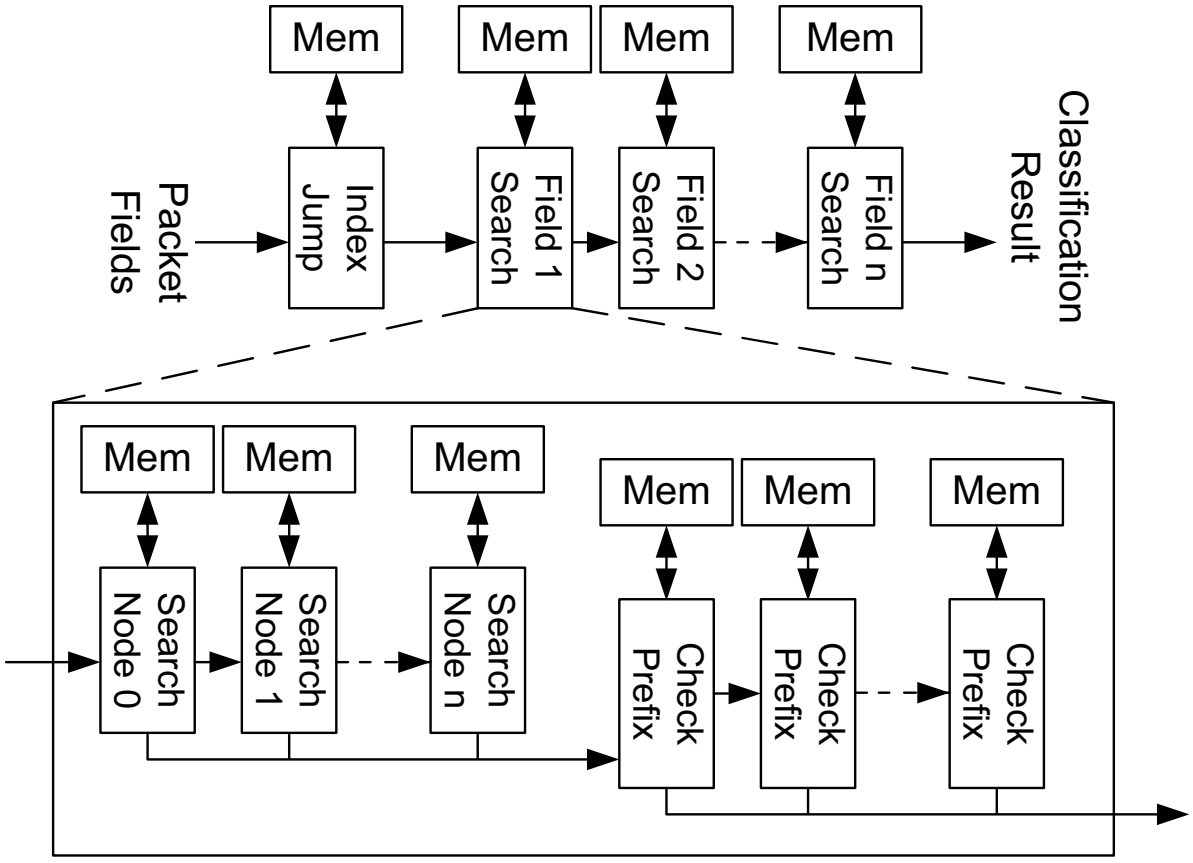

図 3.11　パケット分類処理のパイプライン並列化.

3.8　結言

本章では，パケットフィルタリングにおけるレイヤ 3 およびレイヤ 4 パケットフィルタリングを行うための方式であり，ルーティングテーブルのルックアップをはじめ，DiffServ による QoS 制御などの前段処理として中心的役割を担っており，ユーザの要求に応じて異なるサービスを提供するために欠かせない処理であるパイプライン並列パケット分類方式を提案した.

パケット分類におけるルールの照合法をプレフィックス照合に統一し，最長一致検索アルゴリズムを適用する．最長一致検索アルゴリズムには，メモリ参照回数を極小化し検索遅延を削減可能な LC-Trie 検索法を用いた．そして，さまざまなパケットフィルタリング対象にスケーラブルにより柔軟に対応するためには，トレードオフの関係にある処理負荷とメモ

リ容量のバランスを柔軟に平滑化できる方式が必要であることを述べ，メモリ参照をパイプライン並列に処理することで検索遅延に依存せず検索レートの向上が可能であることに着目し，LC-Trie 検索法をパイプライン並列化したアルゴリズムを提案した．さらに，大規模なルールセットに対しては，Index Jump テーブルを導入し，検索木をヒューリスティックに分割することで, 1 検索木あたりのデータ量を縮退し，検索木の偏りを緩和した．さらに，並列度の向上に伴い，処理に必要な記憶空間が増大することは明らかであり，記憶空間の縮退を考慮する必要がある．この問題に対しては，各フィールドの照合処理を流れ作業的にパイプライン並列に順次実行することにより解決する方法を提案した．

今後の課題としては，今後主流になるであろう IPv6 への対応を考慮し，プロセッサコア構成の最適化および検索アルゴリズムの最適化を行う予定である．

第 4 章

フィルタリングルールの最適化

4.1 緒言

前章で，フィルタリング方式として LC-Trie を用いたパイプライン並列パケットフィルタリング方式を提案した．

同方式はルールの大きさに対してスケーラブルではあるものの，主に数万エントリ超の大規模なルールデータベースを見据えたものであった．一方で，エッジノードや企業ネットワーク，ローカルネットワークにおけるフィルタリングルール数はおよそ数 10 から 2000 程度であることが分かっている [42]．さらに，フィルタリングルールは送信元 IP アドレスおよび宛先 IP アドレスの対，およびとトランスポート層のフィールドであるプロトコル番号や送信元ポート番号ないし宛先ポート番号などで構成されるが，Michael らによれば多くの場合アプリケーションネットワークの数をしのぐ．すなわち，エンド端末におけるパケットフィルタリングのためのに応用する場合，実効的なルール数の観点からルールデータベースを検討することにより，より効果的なフィルタリングが可能になると考えられる．

本章では，ISP やイントラネットで用いられている実際のフィルタリングルールの傾向を考察する．そして，ルールデータベースを本提案方式で用いる LC-Trie 木構造に写像した際のメモリ使用量について述べる．

4.2　フィルタリングルールの傾向

はじめに，今日利用されているトランスポート層のプロトコルは TCP である [53] ことから，以降，エンド端末のフィルタリングルールに出現するトランスポート層のプロトコルを送信元 TCP ポート番号，宛先 TCP ポート番号，あるは TCP ポート番号とする．

Lounavis らによれば，ワイルドカードを含むフィルタリングルールにおける送信元 IP アドレスおよび宛先 IP アドレスのプレフィックス長の分布は図 4.1（文献 [42] から引用）に示すとおり，0，21，23，24，25，28，30，32 という特定プレフィックス長に偏っている．同様に，ワイルドカードを含まないフィルタリングルールにおけるプレフィックス長の分布においても図 4.2（文献 [42] から引用）が示すとおり，特定プレフィックス長に偏っていることが判る．

また，Woo らによるファイアウォールベンダー数社のフィルタリングルールの特徴抽出においてもほぼ同様の傾向が見られた [6]．

このことから，以下のことが特徴として挙げられる．

プレフィックス長：　一般的な IP アドレスのプレフィックスは 0，21，23，24，25，28，30，32 である，

IP アドレス全域指定：　IP アドレスに関するフィールドの約半数がワイルドカードである．

次に，フィルタリングルールにおける TCP ポート番号のフィールドセットのサイズ分布は 4.3（文献 [42] から引用）となっている．さらに，TCP ポート番号のフィールドに関しては，その 5%〜10%が ≥ 1024 であるとしている．これは，unkown ポートに対する一括フィルタリングをしていることを示している．また，[42] が示すとおり，パケットストリームが送受信される方向によって，宛先ポート番号の条件が定義されている場合は，送信元ポート番号はほぼワイルドカードになることが判る．逆向きの場合は，送信元ポート番号の条件が定義されている場合は，宛先ポート番号はほぼワイルドカードになることが判る．したがって，TCP ポート番号に関しては以下の特徴が挙げられる．

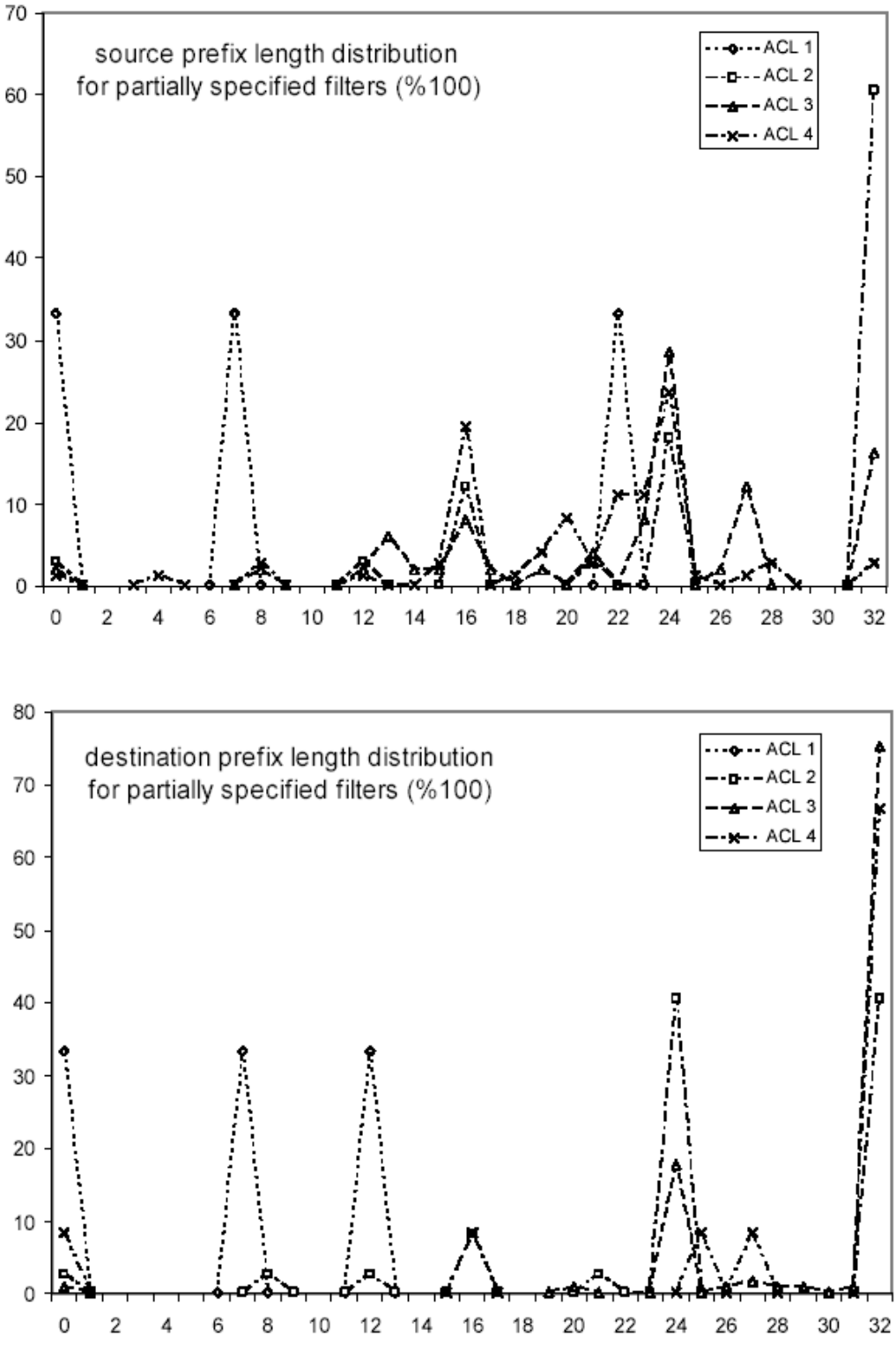

図 4.1　ワイルドカードを含むフィルタリングルールにおけるプレフィックス長の分布.

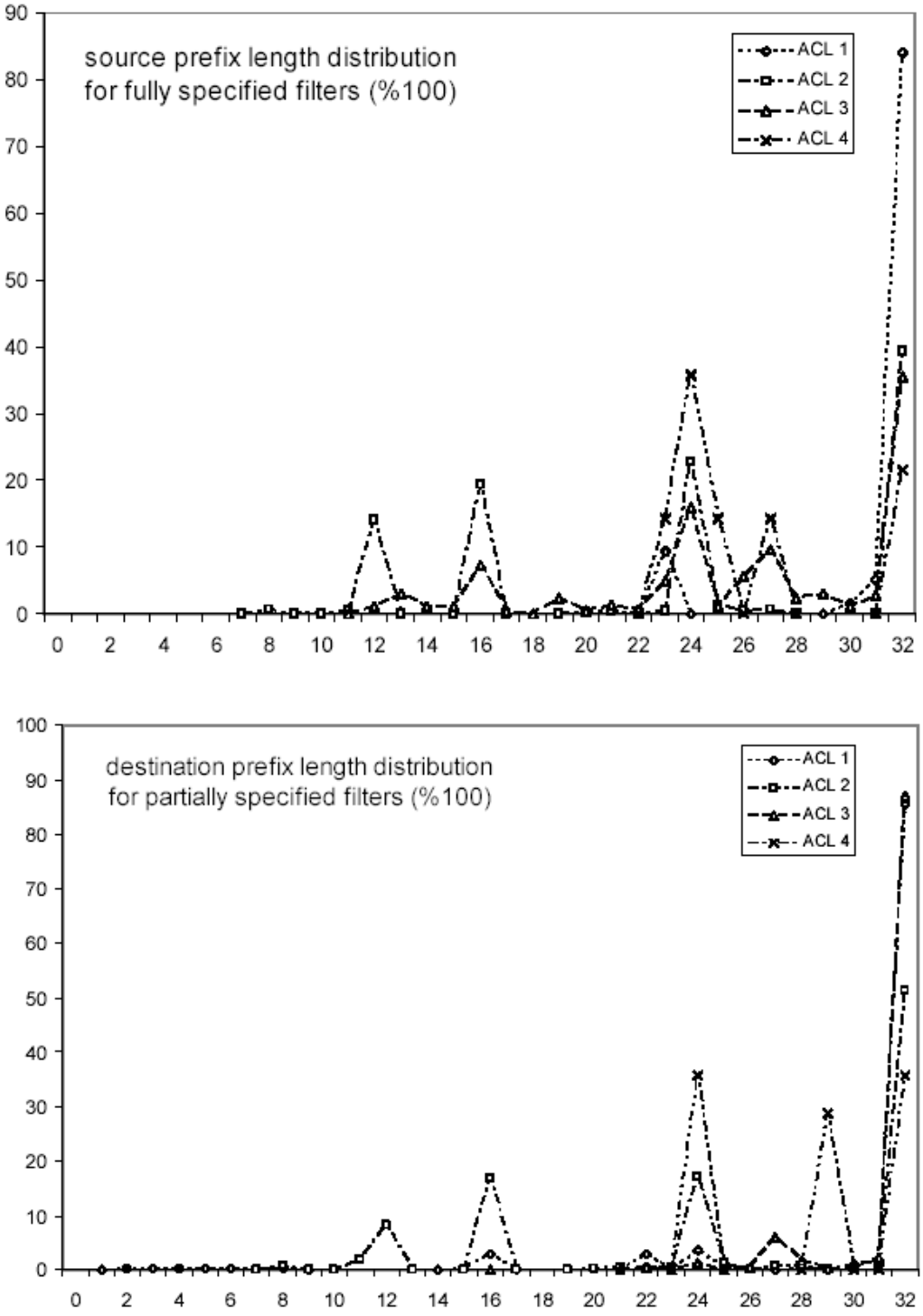

図 4.2　ワイルドカードを含まないフィルタリングルールにおけるプレフィックス長の分布.

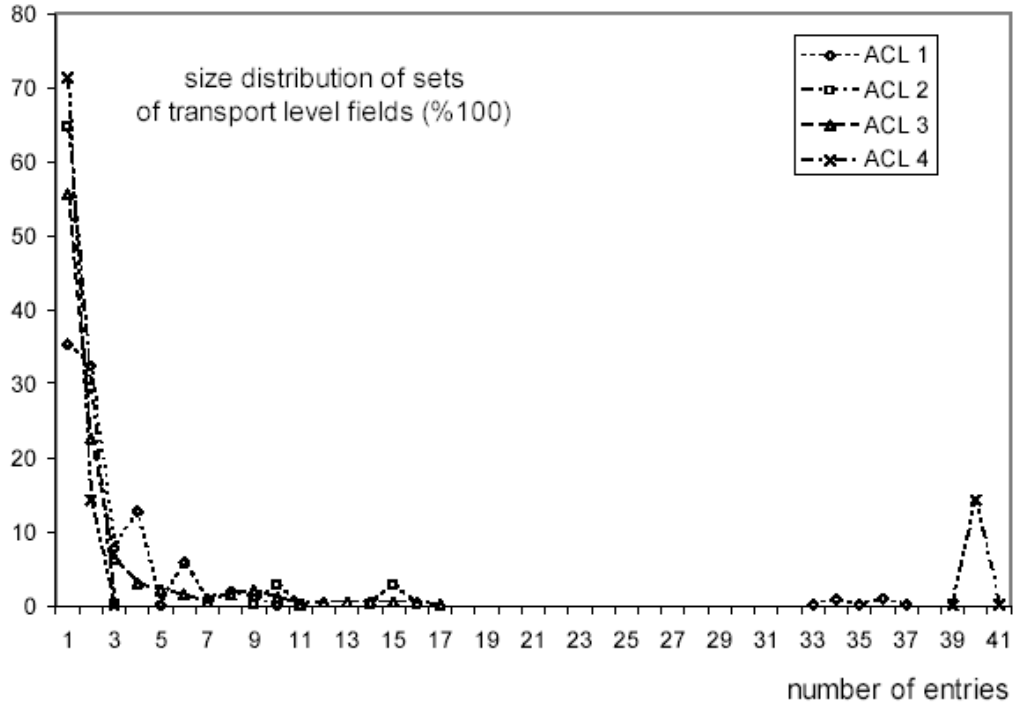

図 4.3　トランスポート層のフィールドセットのサイズ分布.

同一 IP アドレスへの重複エントリ数：　フィルタリングルールに出現する TCP ポート番号フィールドの約 90%が 1-4 個の IP アドレスフィールドに重複している．また，残りの 10%は 5-40 個の IP アドレスフィールドに重複している，

範囲指定の割合：　TCP ポート番号に関するフィールドの全体の約 5%が範囲指定である．

4.3　メモリ使用量

まず，IP アドレスフィールドを LC-Trie に写像したときの，メモリ使用量を測るために，ルール数が 100，500，1000 の場合における，送信元／宛先 IP アドレスがワイルドカードである割合を変化させて，そのときに必要となるメモリ量を求めた．結果，表 4.4 のような結果が得られた.

ワイルドカードがまったくない場合では，ルール数が 100 のとき約 4KByte，ルール数が 500 のとき約 16.5KByte，そして，ルール数が 1,000 のとき約 34.7KByte 必要であること

が判った．このことより，メモリ使用量はルール数の増加率に比例して，

また，すべてがワイルドカードであった場合は約 1.2KByte 必要だった．これは，LC-Trie がマルチウェイトライであることに起因しており，ダミーのリーフが生成されたためである．

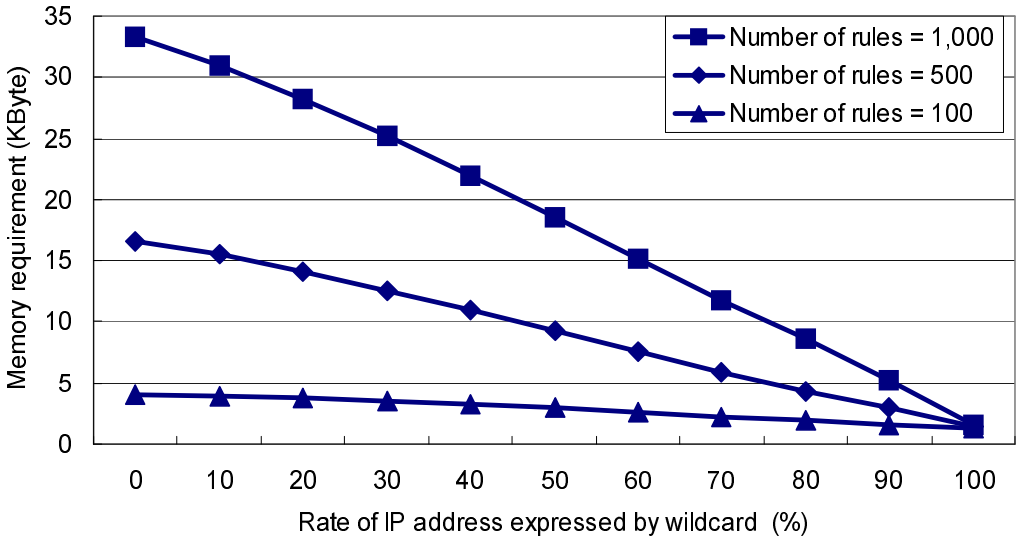

図 4.4　IP アドレスのワイルドカード率毎のメモリ使用量（ルール数 100，500，1000）．

フィルタリングルールの傾向に従い，パーソナルゲートウェイ向きの擬似フィルタリングルールを作成した．また，検索対象の縮退を行い，エンド端末におけるフィルタリングルールを，IP アドレス，および宛先ないし送信元 TCP ポートとした．以上の条件でメモリアクセス数とメモリ消費量を評価した．なお，メモリアクセス数は，最長一致検索における LC-Trie 木構造探索のための SearchTrie 命令によるメモリアクセス数と，プレフィックス検査のための CheckPrefix 命令によるメモリアクセス数の総和である．

図 4.5 は，本提案方式を用いた際のフィルタリングルールの数に対するメモリ消費量と，分割レートである．この結果から，メモリ消費量の増加量はフィルタリングルールのルール数の増加量と比例していることがわかる．ファイアウォールの平均的なルール数は 150 前後であることから，約 8K バイトのメモリ領域があれば十分格納できることが判った．

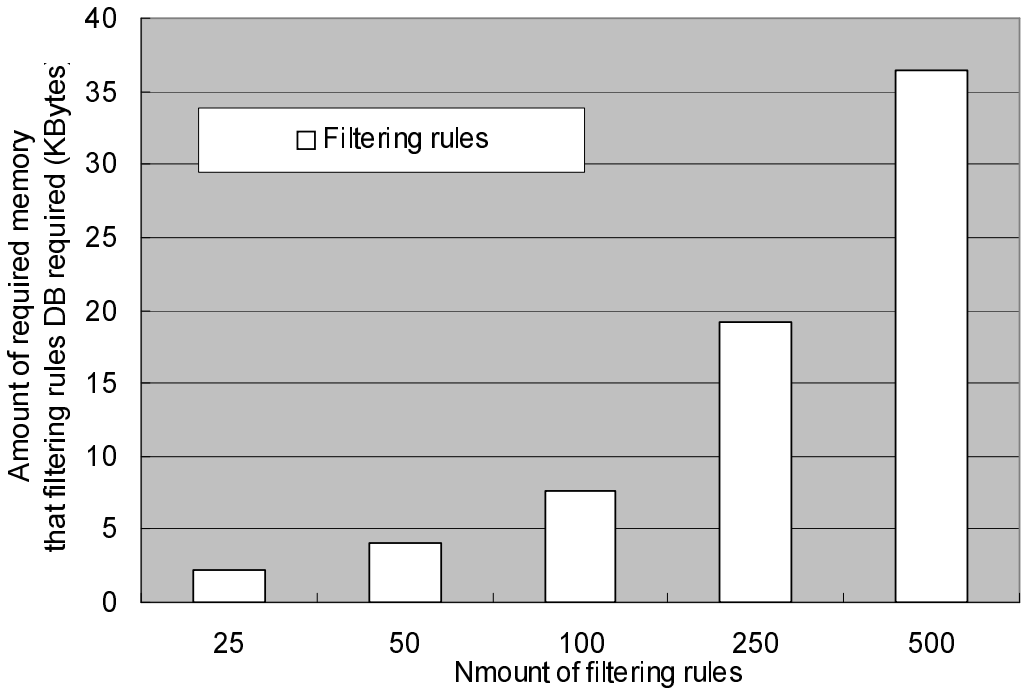

図 4.5 フィルタリングルール別のメモリ使用量と分割レート

4.4 結言

LC-Trie を用いたパイプライン並列パケットフィルタリング方式はルールの大きさに対してスケーラブルではあるものの，主に大規模なルールデータベースを見据えたものであり，また，検索のための Trie 生成あるいは更新の速度は検索処理に対して厳密に考慮していなかった．本章では，エンド端末のパケットフィルタリングに注目して，フィルタリングルールの傾向からルールデータベースの最適化をおこなった．フィルタリングルールの傾向としては，送信元 IP アドレスおよび宛先 IP アドレスのついには，それぞれプレフィックス長に偏りがあることがわかった．すなわち，フィルタリングルールの LC-Trie 木構造への写像の際に，エンド端末におけるフィルタリングルールの傾向を加味することで，LC-Trie 木構造に関するメモリ使用量を縮退することが出来ることがわかった．このことから，特定用途向きにフィルタリングルールを事前最適化することで，パケットフィルタリングの性能向上が見込める．

今後の課題としては，IPv6 を用いたときのパケットフィルタリングのためのフィルタリングルールの傾向を明確にし，IPv6 環境下でのサラウンディング・コンピューティングにおけるフィルタリングルールの最適化を行う必要がある．

第 5 章

データ駆動型ネットワークプロセッサ DDNP による実現法

5.1 緒言

使用状況によって変化する環境下で利用可能なコンピューティング資源を選択的に，かつ，ユーザに対して透過的に協調利用できるネットワークを構成するためには，サービスに応じて周辺環境に適切に処理を分散するためのパケットフローの制御を効果的に実現する必要がある．また，携帯電話や PDA 端末といった携帯端末で形成されるアドホックネットワークのような局所的ネットワークにおいては，そのネットワーク内でのセキュリティポリシーを中央集約的にできないため，各々の端末が分散的に安全性，信頼性を保証する必要がある．すなわち，端末自身がさまざまな階層のプロトコルにおけるヘッダ処理やペイロード処理をネットワーク QoS 制御と，セキュア QoS 制御の両側面から重畳して包括的に処理する必要がある．そのためには，高速な組み込み NP が必要となる．

本章においては，まず，既存 NP，およびデータ駆動型ネットワークプロセッサ:DDNP の特性を示し，DDNP が高速パケット処理性能を柔軟に提供できる NP として期待できることを述べる．次に，提案パケットフィルタリング方式の DDNP による実現方を示す．そして，DDNP アーキテクチャの回路規模および性能評価をハードウェアレベルで行うため，FPGA に最長一致検索用の複合命令 SearchTrie 命令と CheckPrefix 命令，ならびに各種テーブルメモリ参照命令を含む命令セットを備えたプロセッサ (nPE) を実装した性能評価システムの構成を示す．

5.2　既存ネットワークプロセッサとデータ駆動型ネットワークプロセッサ:DDNP

これまでのネットワーク装置は，専用ハードウェアで構成された ASIC(Application Specific Integrated Circuit) か，ソフトウェアを汎用 CPU 上で動作させて実現してきた．ASIC はこれまでの速度重視の性能向上においては劇的な能力を発揮してきた．しかし，今日の複雑多様化するサービスに対応するためには，その開発期間はあまりにもかかりすぎる．ソフトウェアによる実現法は，柔軟性を持ち合わせており新しいプロトコルにも即対応可能である．しかし，ネットワークの光伝送技術は Guilder の法則によると 6 ヵ月ごとに 2 倍に増加するとされてるが，一方で半導体技術は Moore の法則に従えば 18 ヶ月に 2 倍しかならず，汎用 CPU 上で動作するソフトウェア実現法では満足な速度が得られない．こうした問題点において，従来の速度性能と柔軟性のトレードオフの考え方を抜本的に変える新しいタイプの半導体デバイスとしてネットワークプロセッサ (NP) が注目されている [15].

NP は，パケット処理向けに最適化されたプログラミング可能な半導体デバイスである．これにより，ハードウェアの速度性能を維持しつつ，ソフトウェアによりプログラム可能であることであることから，新しいプロトコルにも柔軟に対応可能となる．よって，今後の製品ライフサイクルの短縮と，大幅な開発コストの削減をもたらす．一般にレイヤ 2 からレイヤ 7 ルーティングにおけるタスク (ヘッダ解析，パターンマッチ，ビットフィールド操作，テーブルルックアップ，パケット変形，データ移動) を処理するように設計されている．

現在，各 NP ベンダーはこれらのタスクをパイプライン構成することにより，リンク速度への追従を図っている．例えば，Intel の NP である IXP2800 は図 5.1 に示すように，パケット処理用にパケットエンジン (PE) を 15 個パイプライン接続し，マルチスレッドにより動作する．各 PE は RISC ライクの命令セットと大量のレジスタを保持するカスタムプロセッサコアである．IXP2800 は，1.4GHz という動作周波数を必要とし，消費電力は 14W に上る．

Motorola の C-5 も同様に，Motorola が開発したカスタム 32-bit RISC コアである

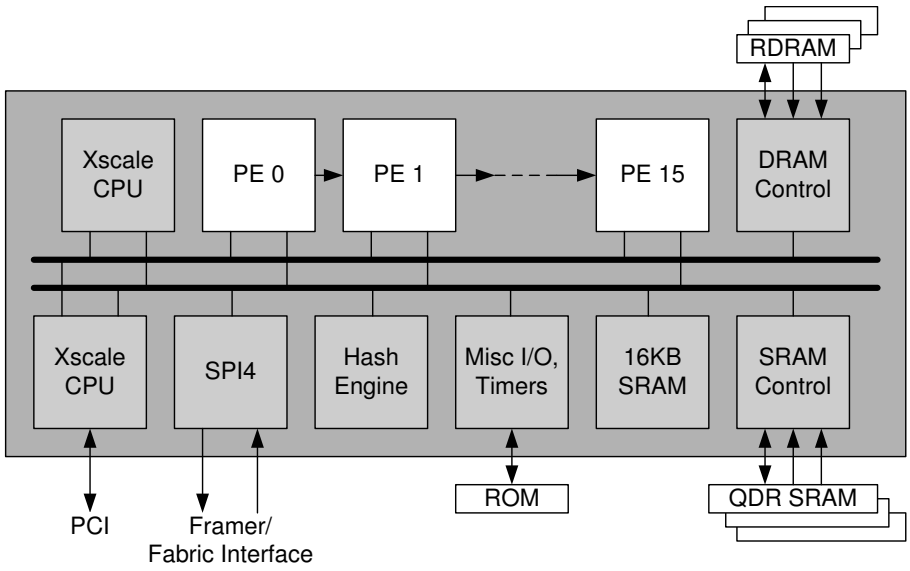

図 5.1　Intel IXP2800 ブロックダイアグラム

Channel Processor(CP) と呼ばれる複数のプロセッサコアを並列に配置することによりトータルで 8Gbps の性能を得ている (図 5.2).

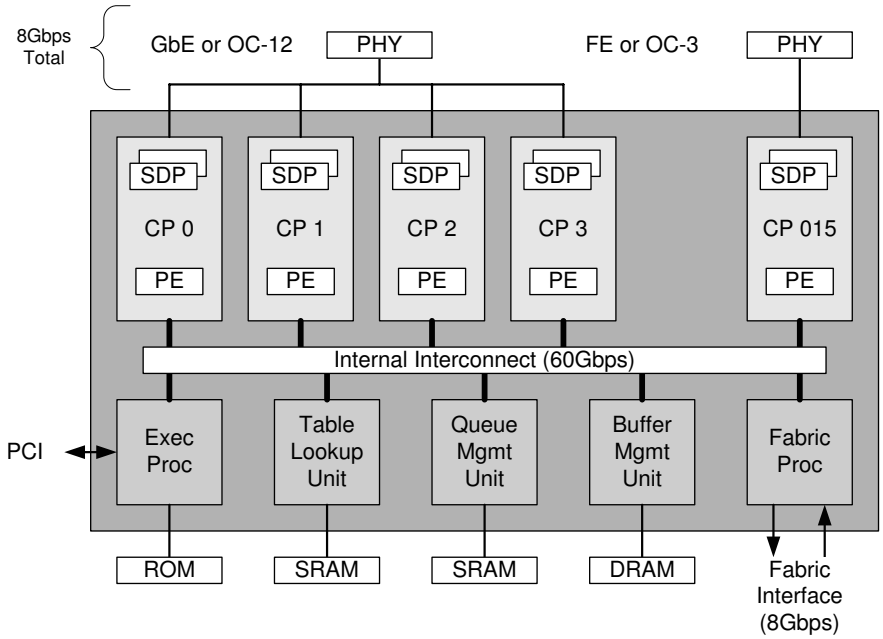

図 5.2　Motorola C-5 ブロックダイアグラム

上記の例のように，既存の NP の多くは，バス接続あるいは相互接続ルータによりパケットエンジンや，各種専用プロセッサを接続している．今後さらに高速化，多様化するであろ

う，ネットワークサービスに柔軟に対応するには，よりパイプライン並列度を高める必要がある．しかし，いずれもクロックベースで動いているため，すべてのプロセッサ間で何らかの協調制御が必要になり，パイプライン並列化の阻害要因となる．さらに，集積度が上がると消費電力は格段に上がってしまう．

NP は雑多なネットワークを潤滑に処理するよう設計されているが，昨今のリンク速度の向上において，NP 単独で多様なサービスに対応することを困難にしている．特に，処理負荷の高いパケット分類や各種テーブル検索処理が高速化の鍵となっている．そのため，現在はパケット分類や各種テーブル検索処理を専用化した高速 LSI を補助的に用いた実現方法を取っているしかし，この方法では，NP の最大の特徴であるプログラマビリティが失われてしまう．

筆者の属する研究室では，ネットワークプロセッサアーキテクチャとして，自己同期型データ駆動型マルチプロセッサ [33] を採用した，データ駆動型アーキテクチャは，自己同期型パイプラインにより自然なパイプライン並列化が可能となるうえ，パイプライン段間の局所的な同期のみ行えばよいので，極少電力化も図れる．これらの性質によって，図 5.3 に示すように，ヘッダ処理や，キューイング機構，その他の演算機構を備えた DDNP にパケット分類に特化した機能回路を持つ nPE を容易に追加実装することが可能となる．

その上，データ駆動原理に基づいており，プロセッサ間の協調制御の必要がない．そのため，プロセッサ内で処理負荷が大きくなり，他の処理が実行できない場合でも，様々な種類のプロセッサを容易にマルチプロセッサシステムとして相互に接続することで負荷分散を可能とする．

よって，上記要件を満たし，高速パケット処理性能を柔軟に提供する NP アーキテクチャとして期待できる．

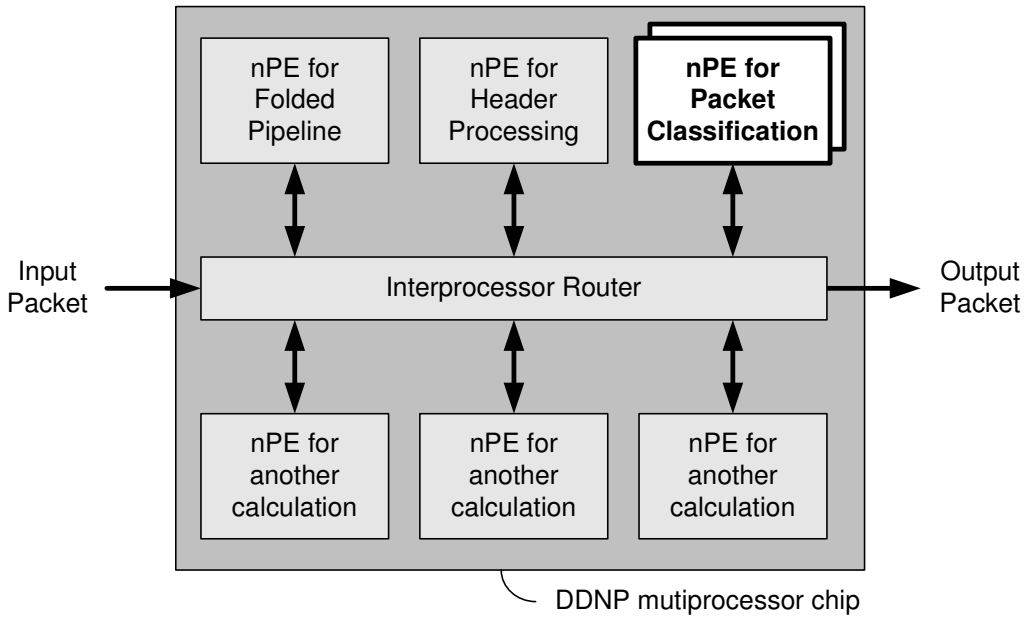

図 5.3　パケット分類用 nPE を持つ DDNP.

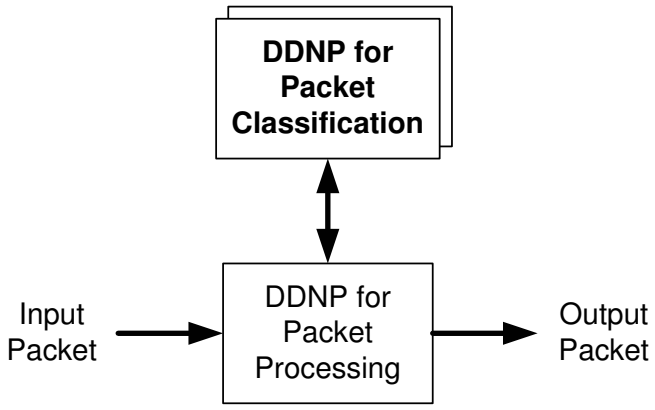

図 5.4　パケット分類機構を持つプロセッサと DDNP の接続.

5.3　データ駆動パケットフォーマット

データ駆動型プロセッサで扱うデータは，データ駆動パケットと呼ばれる形式で表される．パケットフォーマットを図 5.5 に示す．データ駆動パケットは，データ駆動原理を実現するため，演算データにデータの行き先情報，演算コード情報，プログラムのアドレス情報などの制御タグ (図中 Control Data) を付加したデータ形式である．データ部 (図中 DL) は，DDNP では IP パケット処理を対象とするため，4oct(32bit) の精度を採用した．さらに，処理に必要な中間データを保持する領域 (図中 DX) を付加し，1 パケットで処理対象

データと中間データを完結することにより，プロセッサ内を滞留するパケット数の削減を行う．

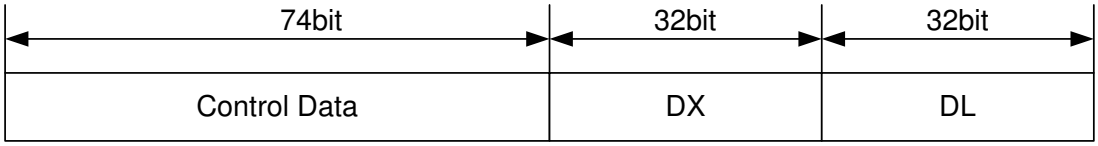

図 5.5　データ駆動型ネットワークプロセッサの基本パケットフォーマット．

5.4　パケット分類特徴命令

本研究では，メモリアクセス回数の削減，およびパイプライン並列による性能向上を達成するために，平均メモリアクセス回数の少ない LC-Trie 法を採用した．しかし，既存の命令セットは単純な処理機能しか有しておらず，検索アルゴリズムをソフトウェア的に実現すると，プログラム全体の命令数およびクリティカルパス長が大きくなり，効率良く実現できない．そこで，LC-Trie 法の特性に基づいて，命令の複合化を行い，プログラム中の命令数を極小化することで，検索レートの向上と検索時間の短縮を図る．

LC-Trie は以下の 2 つの処理ステージで実現される．そこで，各ステージの処理内容に沿って複合化した特徴命令を作成した．

- LC-Trie 木検索
- プレフィックス検査

5.4.1　LC-Trie 木検索命令

LC-Trie 木検索命令 (SearchTrie) は，LC-Trie 木の 1 レベル分の検索を行うための特徴命令である．本命令は，以下の処理をパイプライン並列に行う．

1. テーブル (LC-Trie Table) アドレスの計算
2. ノード情報の取得

3. ノード情報から検索終了したかどうかの判定
4. 検索終了：プレフィックス検査データへのポインタを出力
5. 検索未終了：次ノード取得用ポインタを出力

検索過程のフローを図 5.6 に示す.

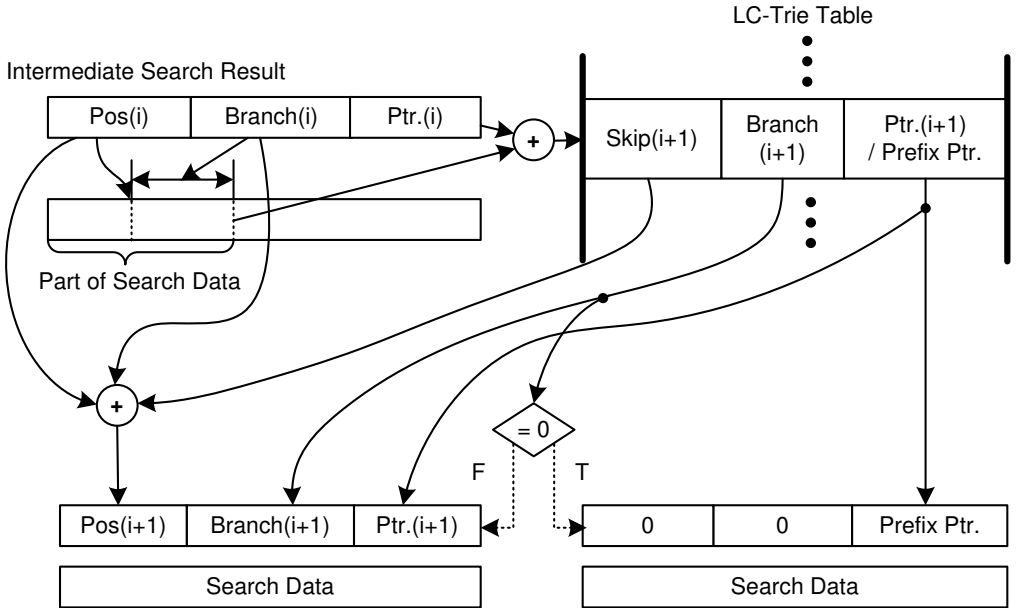

図 5.6　LC-Trie 木検索命令.

本命令は 32 ビットデータ (IPv4 アドレス等) を用いる命令と，32 ビット以上のデータ (IPv6 アドレス等) を扱う命令があり，ソフトウェアにより選択的に使用可能である．以下に各命令の動作を示す.

- SRCHTRI4

  32bit データに対して，内蔵メモリまたは外付けメモリ内の検索テーブルを参照し，検索処理を行う命令である．図 5.7 に示すとおり，1 つの入力パケットを受け，処理結果によって選択出力する.

  入力パケットの DX データ領域は，検索する 32bit のデータが格納される．入力パケットの DL データ領域は，図 5.8 に示すフィールドに分割されて解釈される．LC-Trie は，前章で述べたとおり，ノード情報として Branch，Skip，Pointer を保持しており，DL

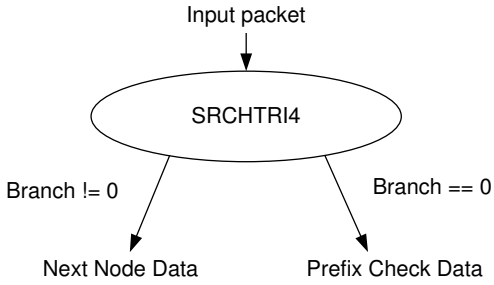

図 5.7　SRCHTRI4 命令.

データ領域の Branch，Pos，Ptr はそれぞれに対応する．ただし，Skip フィールドに関しては，処理の簡略化のため本実現においては現ノードからの省略分を表すのではなく，プレフィックスの先頭から，Skip 後の位置までのビット数を表す.

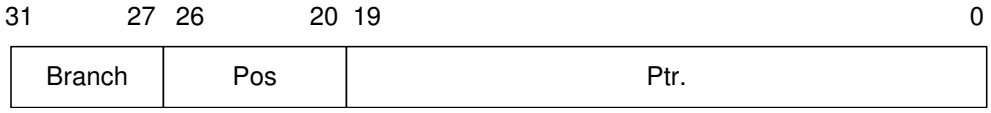

図 5.8　SRCHTRI4 命令における DX データ領域のフィールド解釈.

DL データ領域の Branch，および Pos の値を使用して，DX データ領域から図 5.9 に示すビットフィールドを抜き出し，そのフィールドを整数とみなして値 Val を得る.

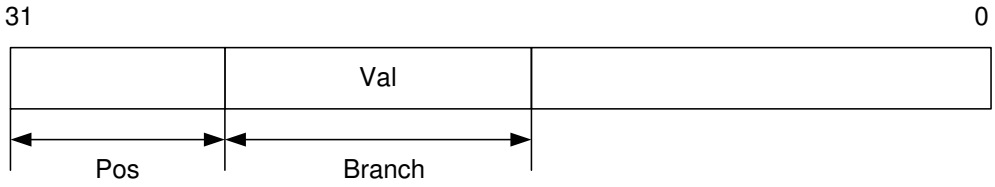

図 5.9　SRCHTRI4 命令における DX データ領域のフィールド解釈.

DX データ領域の Ptr にオフセットに相当する値 Val を加えた値が，次ノードを得るための検索テーブルのアドレスとして使用される．得られたノード情報は，DL データ領域に格納される．その後，DL データ領域の Branch が 0 であれば，LC-Trie 木の検索が終了したとみなす．このとき，DL データはプレフィックス検査を行うためのデータを保持していることになり，プレフィックス検査処理へパケットが出力される．DL データ領域の Branch が 0 で無ければ，さらに子ノードを保持しているということを意

味するため，Branch が 0 になるまで上記の処理を繰り返す．

- SRCHTRI6

  64bit データに対して，内蔵メモリまたは外付けメモリ内の検索テーブルを参照し，検索処理を行う．図 5.10 に示すとおり，2 つの入力パケットを受け，処理結果によって 2 パケットを選択出力する．

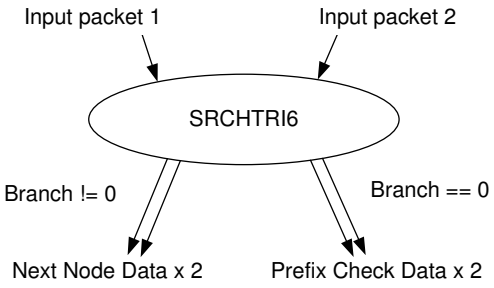

図 5.10　SRCHTRI6 命令．

各入力パケットの DX データ領域には，それぞれ検索する 32bit のデータが格納される．各データは，図 5.11 に示すように結合され，64bit データとして解釈される．

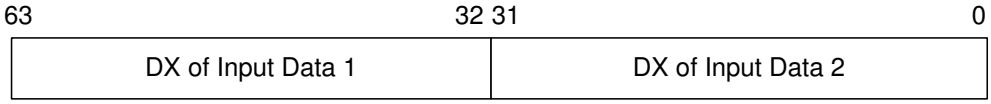

図 5.11　SRCHTRI6 命令における DX データ領域のフィールド解釈．

入力パケット 1 の DL データ領域は，図 5.12 に示すフィールドに分割されて解釈される．LC-Trie は，前章で述べたとおり，ノード情報として Branch，Skip，Pointer を保持しており，DL データ領域の Branch，Pos，Ptr はそれぞれに対応する．ただし，Skip フィールドに関しては，処理の簡略化のため本実現においては現ノードからの省略分を表すのではなく，プレフィックスの先頭から，Skip 後の位置までのビット数を表す．

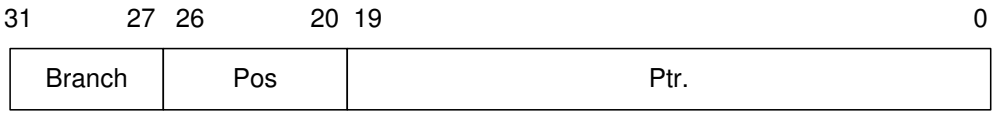

図 5.12　SRCHTRI6 命令における DX データ領域のフィールド解釈．

DL データ領域の Branch，および Pos の値を使用して，DX データ領域から図 5.9 に示すビットフィールドを抜き出し，そのフィールドを整数とみなして値 Val を得る．

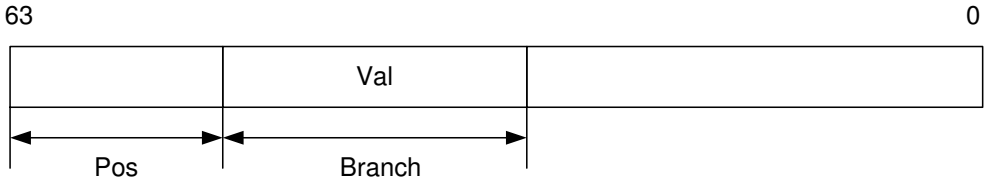

図 5.13　SRCHTRI4 命令における DX データ領域のフィールド解釈.

DX データ領域の Ptr にオフセットに相当する値 Val を加えた値が，次ノードを得るための検索テーブルのアドレスとして使用される．得られたノード情報は，双方のパケットの DL データ領域に格納される．その後，DL データ領域の Branch が 0 であれば，LC-Trie 木の検索が終了したとみなす．このとき，DL データはプレフィックス検査を行うためのデータを保持していることになり，プレフィックス検査処理へ 2 つパケットが出力される．DL データ領域の Branch が 0 で無ければ，Branch が 0 になるまで上記の処理を繰り返す.

5.4.2　プレフィックス検査命令

プレフィックス検査命令 (CheckPrefix) は，SearchTrie 命令によって得られたプレフィックスの正当性を検査し，NextHop 情報を得るためのポインタを出力する命令である (図 5.14)．SearchTrie 命令と同様に，以下の処理をパイプライン並列に行う.

1. プレフィックスと検査データの取得

2. IP アドレスとプレフィックスの比較

3-1. 一致：検索結果の出力

3-2. 不一致：プレフィックス検査用ポインタの出力

検査過程のフローを図 5.14 に示す.

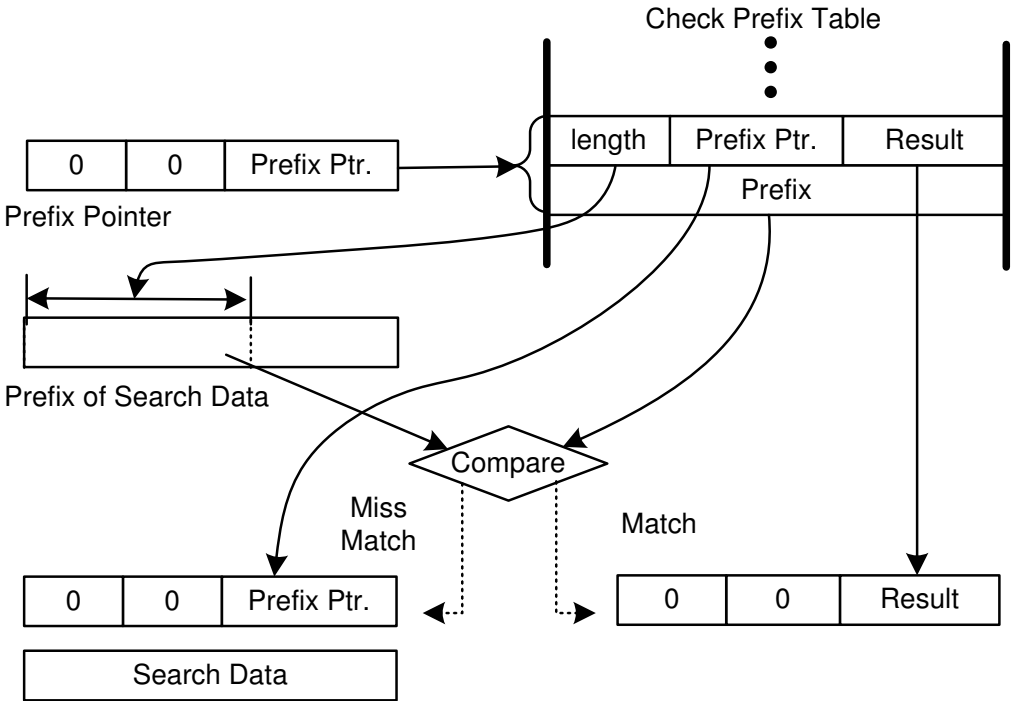

図 5.14　プレフィックス検査命令.

本命令は，SearchTrie 命令同様，32 ビットデータ (IPv4 アドレス等) を用いる命令と，32 ビット以上のデータ (IPv6 アドレス等) を扱う命令があり，ソフトウェアにより選択的に使用可能である.

- CHKPRFX4

  32 ビットデータに対して，内蔵メモリまたは外付けメモリ内のプレフィックス検査用データ 2 ワードを参照し，プレフィックス検査処理を行う．図 5.15 に示すとおり，1 つの入力パケットを受け，検査結果によってパケットを選択出力する.

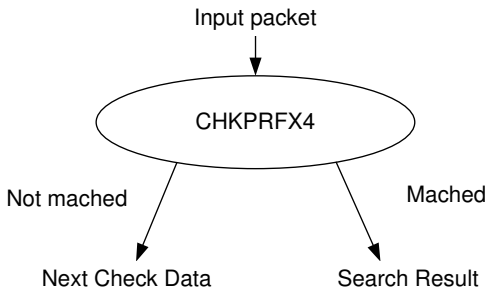

図 5.15　CHKPRFX4 命令.

入力パケットの DX データ領域は，検査する 32bit のデータが格納される．入力パケットの DL データ領域は，メモリ参照用のアドレスとして使用する．まず，DL データ領域の値を基にメモリ参照を行い，プレフィックス検査用データ (図 5.16-(A)) を取得する．同時に，DL データ領域の値に 1 を加算したものでメモリ参照を行い，検査対象とするプレフィックス (図 5.16-(B)) を取得する．

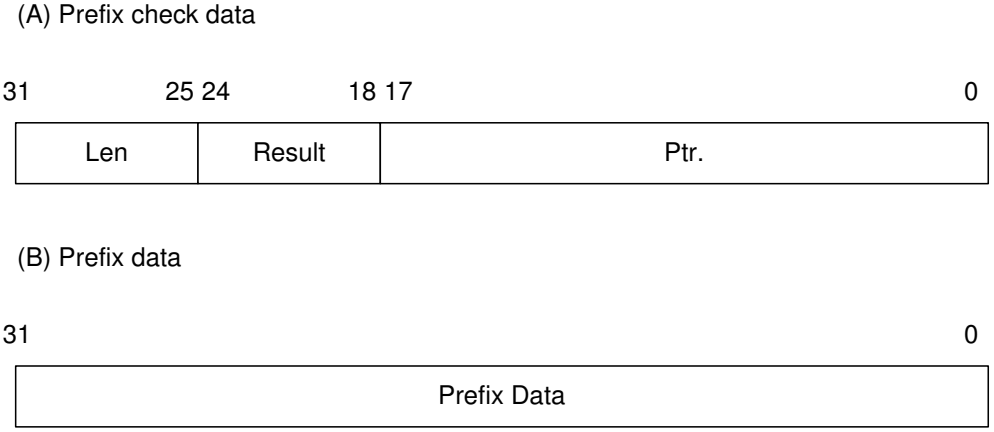

図 5.16　CHKPRFX4 命令で使用するデータフィールドの解釈.

検査データである DX データ領域の上位側 Len ビットと，Prefix Data の同じく上位側 Len ビットとの間で，一致・不一致の比較を行う．

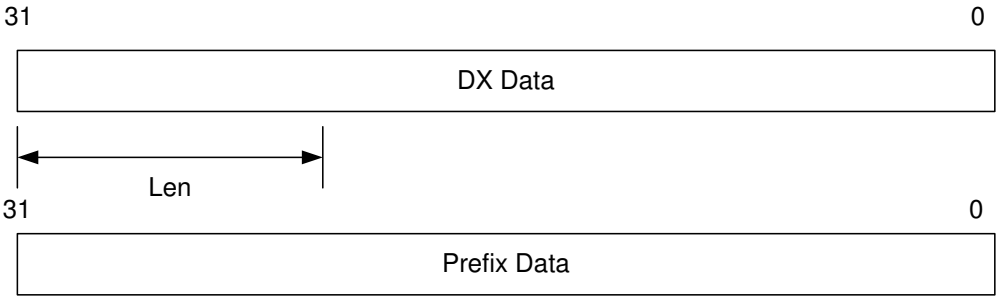

図 5.17　CHKPRFX4 命令におけるプレフィックスの一致・不一致検査.

一致した場合は，プレフィックス検査用データ中の Result を DL データ領域に格納し，検索結果として出力される．不一致の場合は，プレフィックス検査用データ中の Ptr を DL データ領域に格納し，プレフィックス再検査用ポインタとして出力される．

- CHKPRFX6

64 ビットデータに対して，内蔵メモリまたは外付けメモリ内のプレフィックス検査用データ 3 ワードを参照し，プレフィックス検査処理を行う．図 5.18 に示すとおり，1 つの入力パケットを受け，検査結果によってパケットを選択出力する．

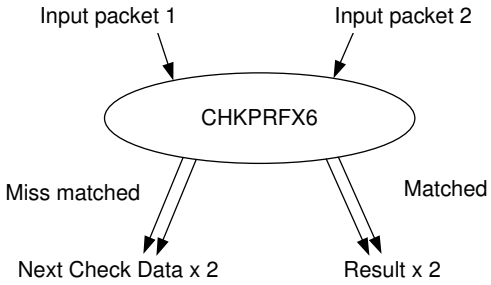

図 5.18　CHKPRFX6 命令.

各入力パケットの DX データ領域には，それぞれ検索する 32bit のデータが格納される．各データは，図 5.19 に示すように結合され，64bit データとして解釈される．

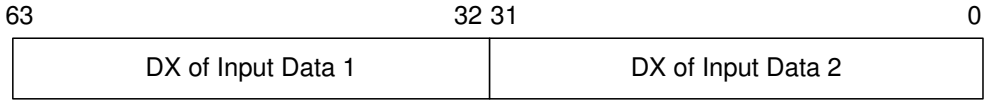

図 5.19　SRCHTRI6 命令における DX データ領域のフィールド解釈.

入力パケット 1 の DL データ領域は，メモリ参照用のアドレスとして使用する．まず，DL データ領域の値を基にメモリ参照を行い，プレフィックス検査用データ (図 5.20-(A)) を取得する．同時に，DL データ領域の値に 1 を加算したものと，2 を加算したものをアドレスとしてメモリ参照を行い，検査対象とするプレフィックスの下位 32 ビット (図 5.20-(B)) と，プレフィックスの上位 32 ビット (図 5.20-(C)) をそれぞれ取得する．

検査データである DX データ領域の上位側 Len ビットと，Prefix Data の同じく上位側 Len ビットとの間で，一致・不一致の比較を行う．

一致した場合は，プレフィックス検査用データ中の Result を DL データ領域に格納し，検索結果として 2 パケットを出力される．不一致の場合は，プレフィックス検査用データ中の Ptr を DL データ領域に格納し，プレフィックス再検査用ポインタとして 2 パ

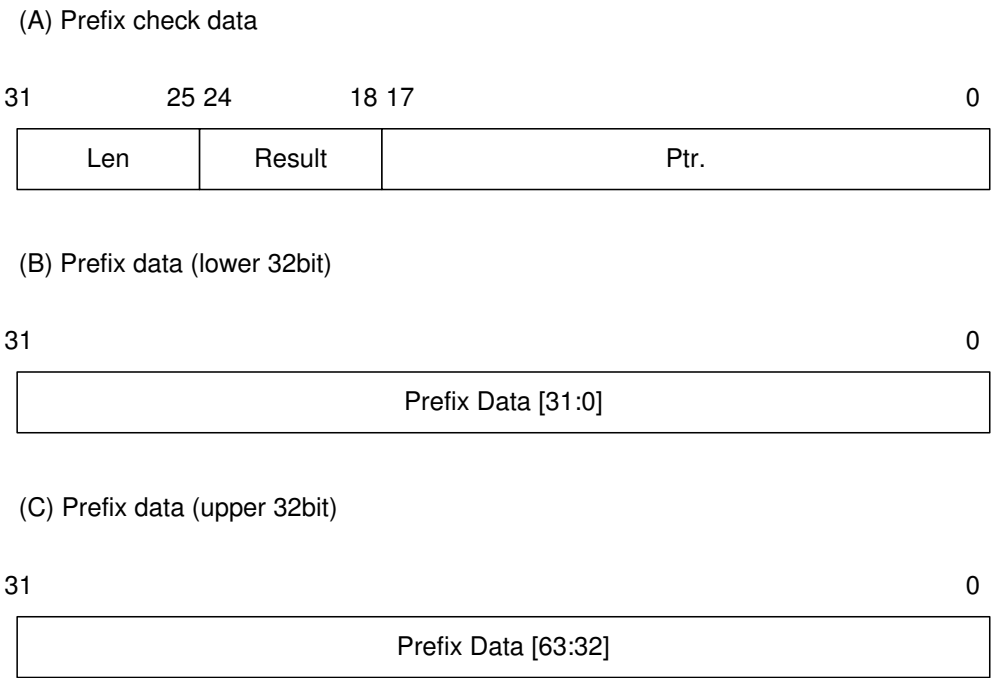

図 5.20　CHKPRFX6 命令で使用するデータフィールドの解釈.

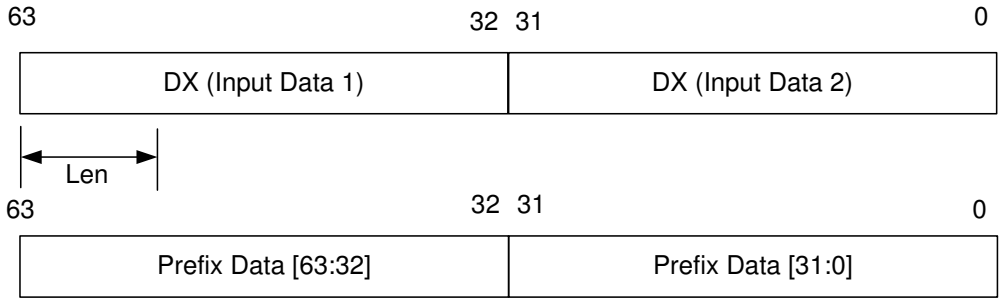

図 5.21　CHKPRFX6 命令におけるプレフィックスの一致・不一致検査.

ケットを出力される.

- CHKPRFXD

  64 ビットデータに対して，内蔵メモリまたは外付けメモリ内のプレフィックス検査用データ 3 ワードを参照し，プレフィックス検査処理を行う．CHKPRFX6 の機能に加え，128 ビットプレフィックス検査に向けた拡張機能を有している．図 5.22 に示すとおり，1 つの入力パケットを受け，検査結果によってパケットを選択出力する．

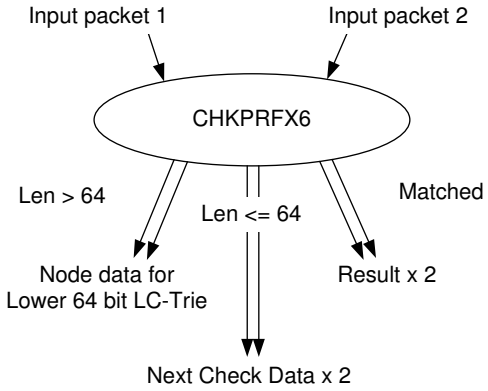

図 5.22 CHKPRFXD 命令.

各入力パケットの DX データ領域には，それぞれ検索する 32bit のデータが格納される．各データは，図 5.23 に示すように結合され，64bit データとして解釈される．

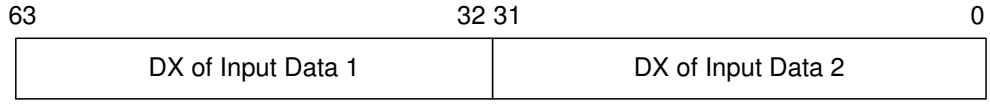

図 5.23 SRCHTRID 命令における DX データ領域のフィールド解釈.

入力パケット 1 の DL データ領域は，メモリ参照用のアドレスとして使用する．まず，DL データ領域の値を基にメモリ参照を行い，プレフィックス検査用データ (図 5.24-(A)) を取得する．同時に，DL データ領域の値に 1 を加算したものと，2 を加算したものをアドレスとしてメモリ参照を行い，検査対象とするプレフィックスの下位 32 ビット (図 5.24-(B)) と，プレフィックスの上位 32 ビット (図 5.24-(C)) をそれぞれ取得する．

検査データである DX データ領域の上位側 Len ビットと，Prefix Data の同じく上位側

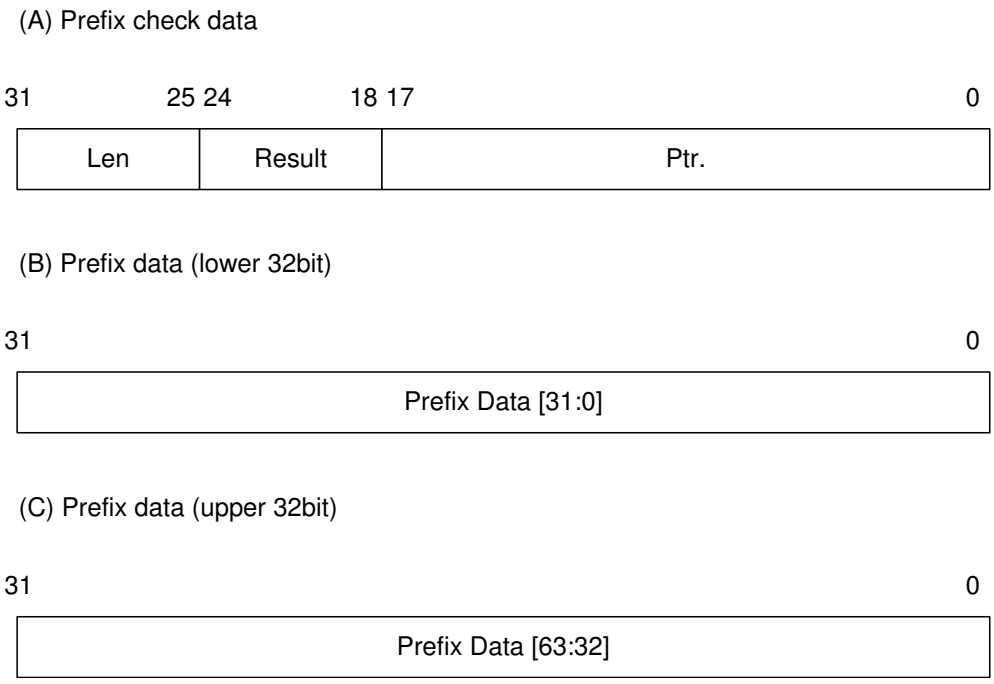

図 5.24　CHKPRFXD 命令で使用するデータフィールドの解釈.

Len ビットとの間で，一致・不一致の比較を行う.

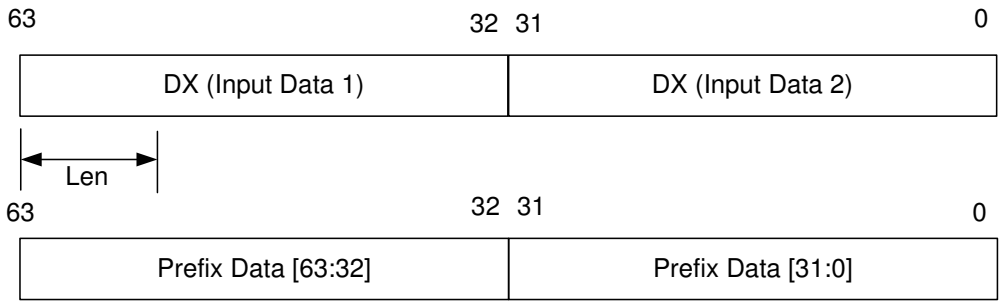

図 5.25　CHKPRFXD 命令におけるプレフィックスの一致・不一致検査.

Len の値は 1 から 128 まで設定可能で，Len が 0 の場合は，128 とみなされる．64 以上が設定されると，64 ビット全ビット比較となる.

一致して，かつ Len $>$ 64 の場合は，プレフィックス検査用データ中の Ptr が DL データ領域に格納され，新たな LC-Trie 検査のためのポインタとして出力される．一致して，且つ Len $\leq$ 64 の場合は，Result の値が DL データ領域に格納され，検索結果として出力される．不一致の場合は，プレフィックス検査用データ中の Ptr の値が DL デー

タ領域に格納され，プレフィックス再検査用ポインタとして出力される．

5.5　ナノプロセッサ構成

ナノプロセッサを構成するに当たり，専用ハードウェア実現法の欠点である柔軟性の欠如，汎用プロセッサによる実現法の欠点である低性能の両者を克服するために，次の要件を満足する nPE を構成する必要がある．

- 必要最低限のハードウェア追加により追加命令を実現でき，既存の命令セットも実行可能なこと．
- メモリアクセスの並列パイプライン化によって，メモリ参照遅延を隠蔽できること．
- ルールセットの分布に応じて，最適なプログラム，およびメモリ割当てを柔軟に設定可能なこと．

本研究で使用するデータ駆動型プロセッサは，整数演算用 nPE，およびテーブル参照用 nPE(TBLnPE) から成るマルチプロセッサ構成である．基本構成は現行のデータ駆動型マルチプロセッサ DDMP と同等であり，整数演算用 nPE は，既存の整数演算用 nPE を使用する．テーブル参照 nPE は，メモリアクセス命令を主体とし，内蔵オンチップメモリ (内部テーブル) と外付けオフチップメモリ (外部テーブル) へのアクセスが可能である．さらに，各テーブル参照用 nPE は独立してアクセス可能なメモリをそれぞれ持ち，計 2 個のメモリを外部テーブルとして持つ．これらのテーブル参照用 nPE はソフトウェアで選択的に使用できる．これより，それぞれのテーブル参照用 nPE に異なったテーブル情報を持つメモリ参照プログラムを実装できる．あるいは，2 つの nPE に同一のテーブル情報を分割配置することも柔軟に行える．本研究においてテーブル参照用 nPE は，最もメモリ参照の頻度が高い最長一致検索処理において使用する．このとき，既存のテーブル参照命令セットを継承しつつ，高速化を図るため後述する最長一致検索用特徴命令を実装する．

また，最長一致検索処理において，プレフィックス検査命令が 1 回の処理においてメモリ参照を 2 回行う必要があるため，処理遅延が大きくなる．そのため，プレフィックス検査を

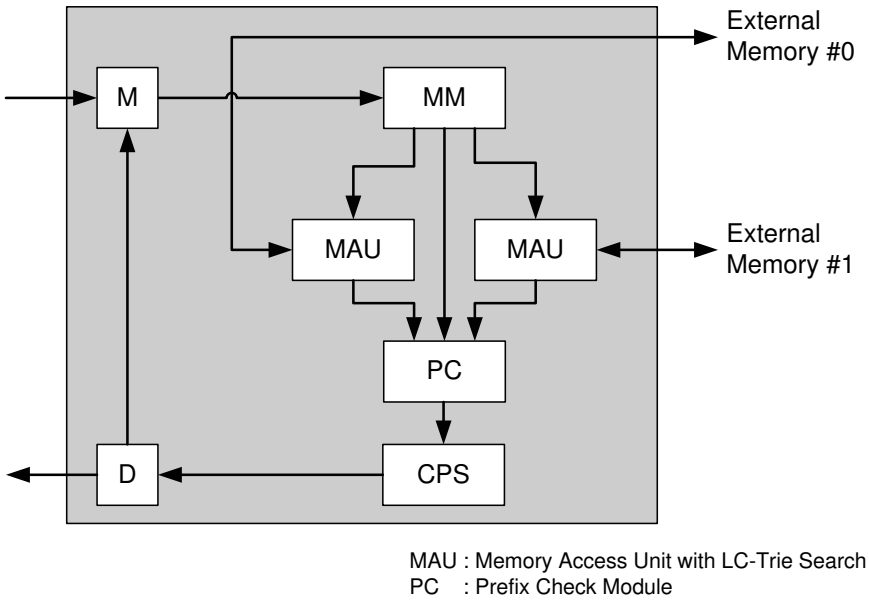

図 5.26　2 ポートメモリ参照機能搭載拡張 TBLnPE.

行う ExtTBLnPE を用意し，LC-Trie 検索命令を行う TBLnPE と協調動作させることで更なる高速化を図る．ExtTBLnPE は，図 5.26 に示すとおり，2 回のメモリ参照を 1 命令サイクルで実行するため，2 つの独立したメモリに並列にアクセスするよう設計している．

なお，TBLnPE と ExtTBLnPE が，同じメモリを共用するため，図 5.27 に示すように，nPE とメモリの間に調停機構を実装する．

本構成において，LC-Trie 検索用データとプレフィックス検査用データを各メモリに適宜分散格納することにより，処理負荷を低減する．

5.6　パケット分類ソフトウェア

パケット分類処理用のデータ駆動プログラムは，前章で述べたように，Index Jump 部および Field Classification 部からなる．入力フィールドより，まず Index Jump テーブルの参照ポインタを得るための Index 計算を行う．Index 計算は単純なビット演算命令の組み合わせにより表現される．その後，フィールド数 n に応じて，SearchTrie 命令および CheckPrefix 命令を用いた最長一致検索処理を n 回行う．

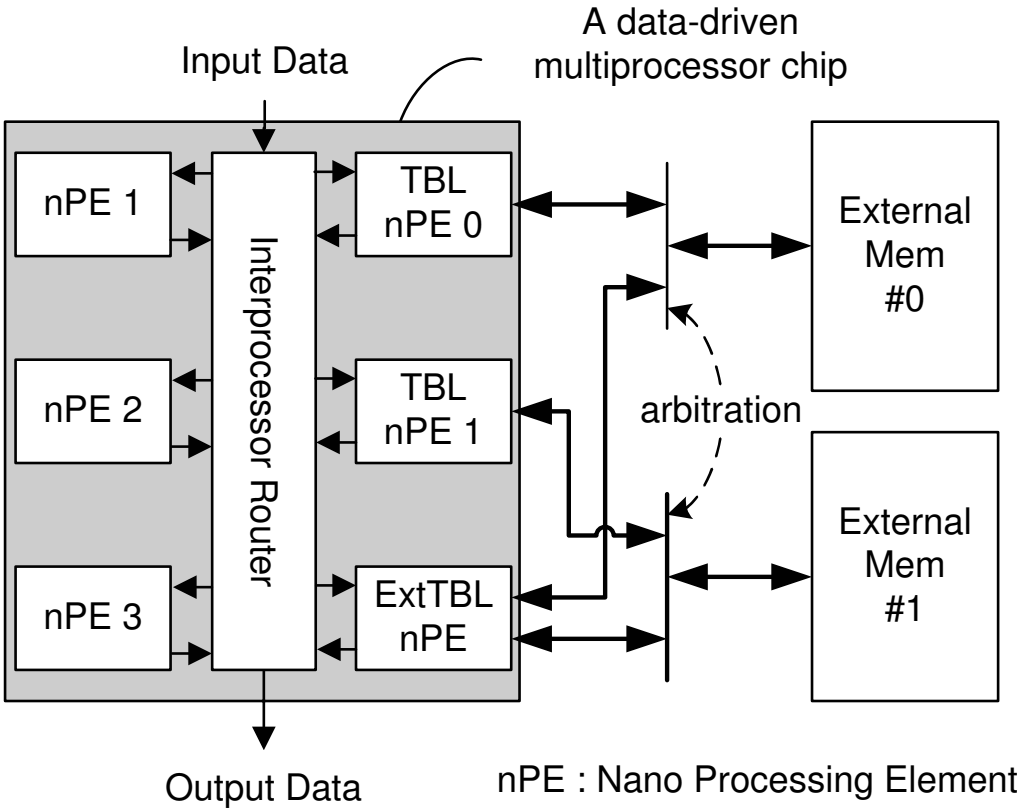

図 5.27　メモリの調停 I/F と分散メモリ構成.

図 5.28 はパケット分類処理のためのデータ駆動プログラムを示している.

図 5.29 は，Field Classification 部における (A)32 ビットデータに対する最長一致検索と，(B)64 ビットデータに対する最長一致検索を示している.

(A) は IPv4 の宛先アドレス (32 ビット) から NextHop を求めるプログラムである．まず，入力された IP アドレスは，LC-Trie 検索用初期データを付加され，1 つのデータ駆動パケットで表現される．次に，LC-Trie 検索用命令 (SRCHTRI4) にて内部テーブル (或いは外部テーブル) に格納されている LC-Trie 木のノード情報を基に，木の探索を行う．そして，探索が終了すると，プレフィックス検査用命令 (CHKPRFX4) を用いて検索の正当性を検査する．プレフィックスが一致すると，テーブル参照命令 (readMem) を用いて NextHop を出力する.

(B) では IPv6 の最長一致検索を行う．ただし，IPv6 のアドレス長は 128 ビットあるが，一方で，データ駆動パケットのデータフィールド長は 32 ビットである．そのため，宛先ア

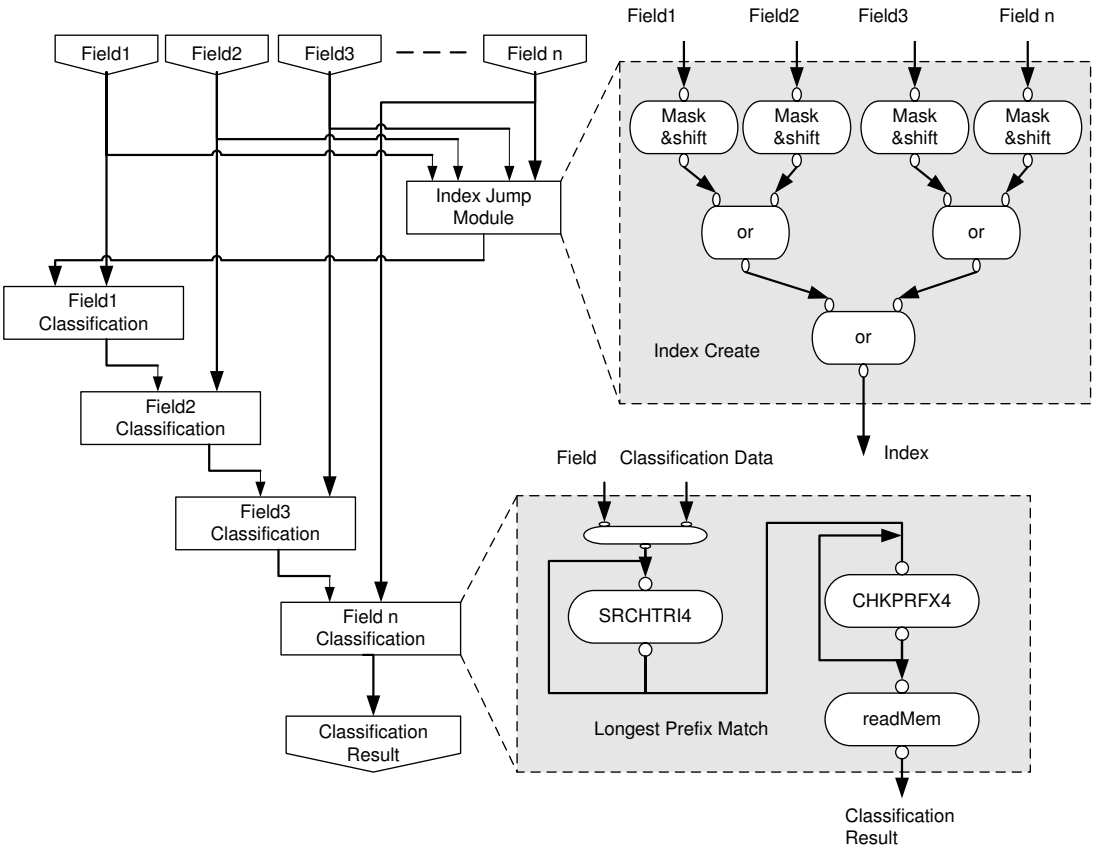

図 5.28　パケット分類処理のデータフローグラフ.

ドレスを 4 分割して，先頭の 2 データパケット (64 ビット) でまず最長一致検索を行い，さらにプレフィックスが長い場合には後半 2 データパケットを適宜呼び出し，継続して検索する.

5.7　結言

本章では，既存 NP，およびデータ駆動型ネットワークプロセッサ:DDNP の特性を示すことで，DDNP の採用するデータ駆動型アーキテクチャにより，柔軟かつ高速なパケット処理機構を提供できる NP として期待できることを示した．次に，提案パケットフィルタリング方式の DDNP による実現方として，データ駆動パケットフォーマット，および最長一致検索用の特徴命令 SearchTrie 命令と CheckPrefix 命令の構成を定義し，それらを用いたソフトウェア構成とソフトウェアを実行するためのプロセッサ構成を示した．さらに，DDNP

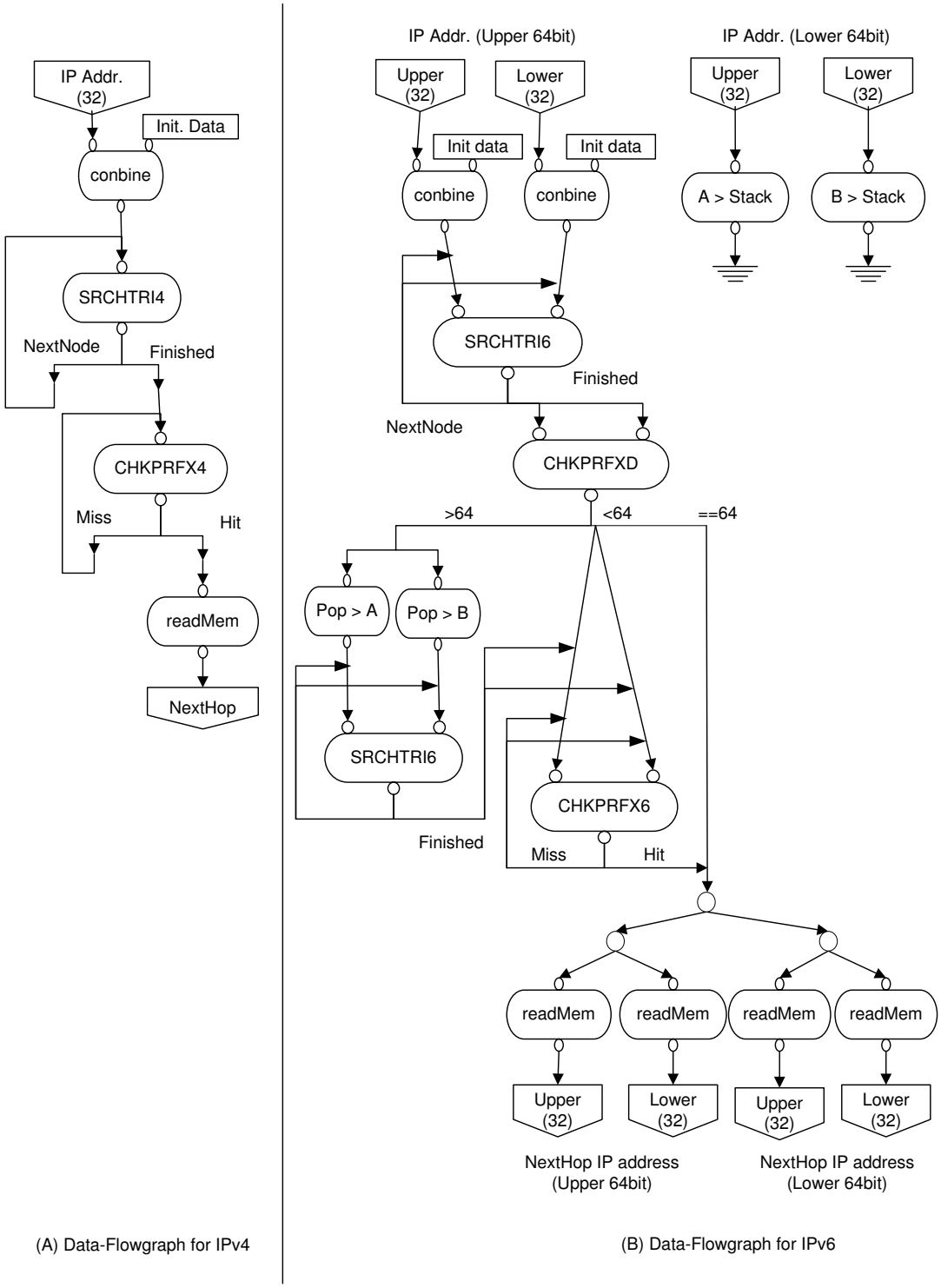

図 5.29　最長一致検索のデータフローグラフ.

アーキテクチャの回路規模および性能評価をハードウェアレベルで行うため，FPGA に最長一致検索用の複合命令 SearchTrie 命令と CheckPrefix 命令，ならびに各種テーブルメモリ参照命令を含む命令セットを備えた nPE を実装した性能評価システムの構成を示した．

今後の課題としては，提案方式を含めたパーソナルゲートウェイの構成を評価するため，ハードウェアベースの性能評価システムにおいて固定化されている nPE 内部の構成およびプロセッサ構成，メモリ構成，命令セット等を任意に変更でき，かつ視覚的に性能見積もりを可能とする開発支援ツールとしてのシミュレータの実現が急がれる．

第 6 章

性能評価

6.1 緒言

筆者の提案したパケットフィルタリング機構を含む，DDNP アーキテクチャの回路規模および性能評価をハードウェアレベルで行うため，FPGA に最長一致検索用の複合命令 SearchTrie 命令と CheckPrefix 命令，ならびに各種テーブルメモリ参照命令を含む命令セットを備えた nPE を実装した評価システムを開発した．

以下本章では，同評価システムを用いて以下の評価を行った．

1. パケット分類用回路の回路規模
2. 最長一致検索の性能評価
3. パケットフィルタリングの性能評価

評価結果より，提案方式の有効性を明らかにする．

6.2 性能評価システム

本研究において，本提案方式の機能検証，および性能見積もりのためのシミュレータを作成した．また，自己同期型パイプラインにより構成された IP パケット向きアーキテクチャに関する各種性能評価を実施するために必要なエミュレーション機能と性能を備えた，DDNP 用評価ボード (図 6.1) を試作した．

本評価ボードの作成に当たり，搭載されるデータ駆動型 VLSI チップは，自己タイミング型パイプラインハードウェア機構によりデータ駆動方式を実現したアーキテクチャ，特に，

IP パケット処理向きアーキテクチャの各種性能評価を効果的に実施するために，次の要件を満たすこととする．

- エミュレーション評価の誤差を極力少なくするために，本研究テーマの基本原理である自己同期型パイプライン機構により実現されていること，
- 多次元パイプライン処理構造を模擬するために，トークンの追い越しを許す動的データ駆動方式に基づき動作すること，および
- 多数のデータ駆動型プロセッサを有機的にパイプライン接続したシステムアーキテクチャの評価を実施するために，多様な接続構造をチップ上に実現する必要がある．したがって，複数のプロセッサが 1 チップ上に集積化され，その処理性能が通常の汎用マイクロプロセッサの性能以上のピーク性能を有していること．

さらに，IP パケット処理向きアーキテクチャについて多用な評価を実施するために，本ボードには，回路構成を自由に再構成可能な FPGA チップが 2 個搭載されていることに加えて，次の要件を満たす，自己同期パイプラインによるデータ駆動型プロセッサがそれぞれ 1 つずつ実装されていること．

- 上記のデータ駆動型 VLSI チップと互換性のあるパケットフォーマットにより，入出力データの授受が可能なこと

以上の要件を満たすデータ駆動型 VLSI チップならびに FPGA チップを搭載したボードは，次の用件を満足していること．

- データ駆動型 VLSI チップと FPGA チップ間の接続には，エミュレーション評価の誤差を極力少なくするために，本研究テーマの基本原理である自己同期型パイプライン機構により実現されていること
- 本ボード上でのソフトウェアの実行状況を観測するための機能を有し，これらの機能がホストコンピュータ経由で制御可能なこと．

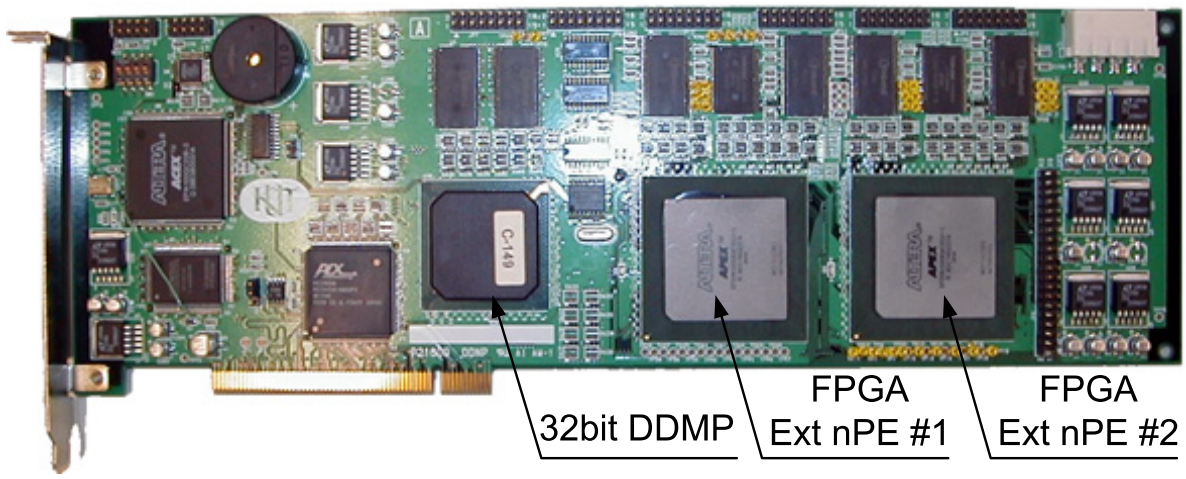

図 6.1　評価用 DDNP ボード.

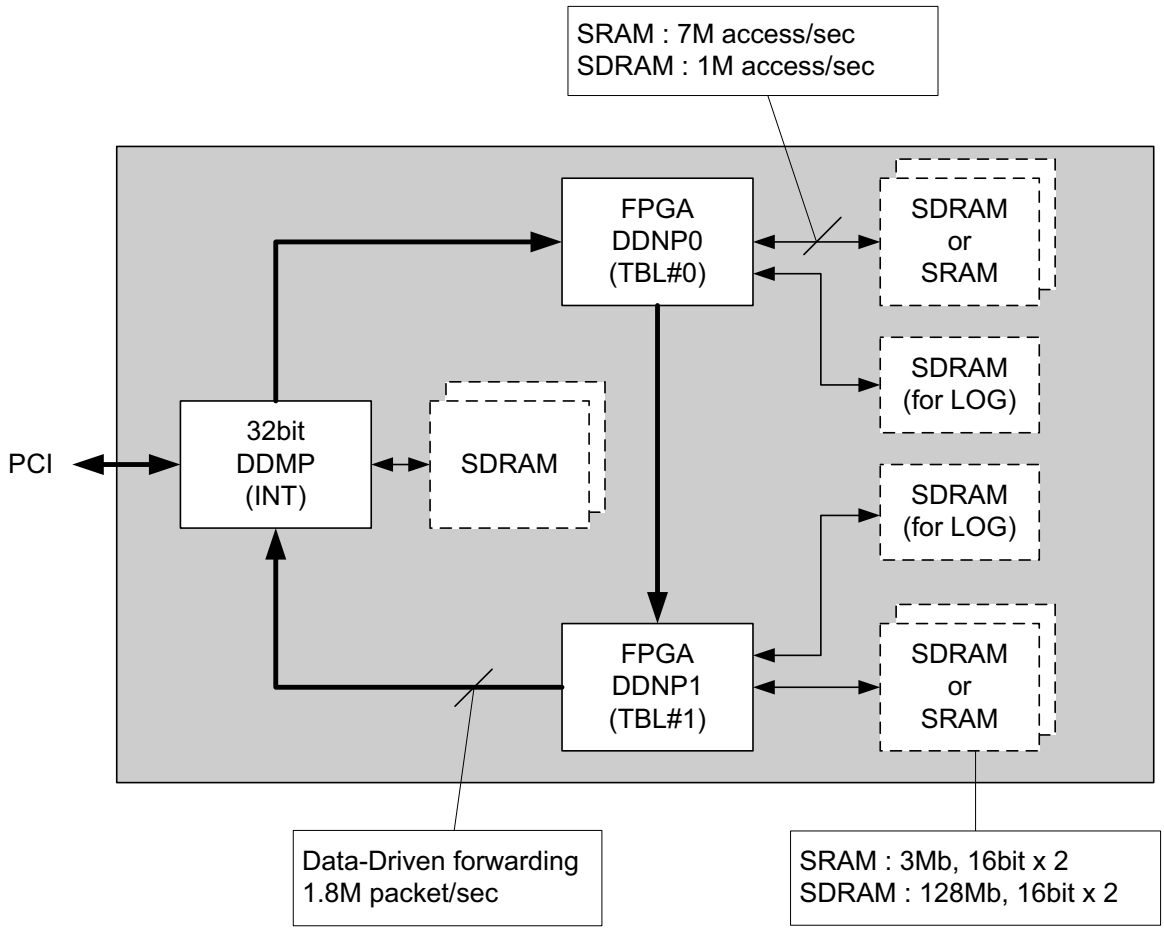

図 6.2　評価用 DDNP ボード構成.

以上の条件を受けて，本評価ボードはシャープ (株) が試作した信号処理用 32bit データ駆動型プロセッサ (DDMP)，および FPGA チップ (APEX DSP DB EP20K 1500E) から構成され，今回新たに提案した特徴命令を含む nPE(Ext nPE#1/#2) を FPGA チップ上に実装した．

各チップの接続関係を図 6.2 に示す．図のように DDMP と各 FPGA は環状に接続されており, 完全な自己同期パイプラインを形成する．FPGA の内部動作速度は約 10MHz, チップ間のデータ転送レートは 1.8M packet/sec である．外付けメモリとして 7M access/sec の SRAM，および 1M access/sec の SDRAM を搭載し，排他利用が可能である．FPGA に実装した nPE は，動作性能を 32bit データ駆動型プロセッサの動作性能に対して等価的にスケーリングして回路実装しているため，測定結果がそのまま比例換算可能である．

32bitDDMP チップ

32bitDDMP は整数演算用 nPE およびその命令セットを保持する．本チップ上では，既存のデータ駆動プログラムを実行する．さらに，本評価ボードのリアルタイムデータ入力用の制御プログラムを実行する．これは，本評価ボードにおいてホストマシンとのデータの授受は PCI 経由で行われるが，PCI インタフェースはデータ転送速度が遅く，本評価ボード内の動作速度に対するリアルタイムデータ入力が困難なため，一時本チップに接続された，SDRAM 内に入力データを保持し，その後バースト的に読み出すことによって，チップ内動作速度等価な入力レートを実現させるために必要である．

本チップは，データ駆動パケットの入出力ポート，ブリッジとの HostI/F ポート，および SDRAM インターフェースポートを持つ．

- DDMP パケットの入出力ポートは，出力側を FPGA0 と，入力側を FPGA1 と接続し，DDMP パケットデータを 15Mpacket/sec の速度で転送する．
- ブリッジとは上り下りあわせて 30Mpacket/sec の転送速度で DDMP パケットデータを転送する．SDRAM とは 32bit 幅，133MHz の転送速度で通信を行う．

- フローグラフパケット，データパケットないし制御パケットはホスト PC から PCI，ブリッジ経由で投入する．フローグラフパケットは 32bit-DDMP 内の PS メモリに保持される．パケットデータは入力パケットとなる．
- SDRAM は 32bit 幅，133MHz の転送速度を持つインターフェースにより接続する．

FPGA チップ

FPGA チップは，既存データ駆動プロセッサにはない機能回路を実現するために用いられる．拡張 TBL と呼ぶ最長一致検索処理を超高速に行うための特長命令セットをサポートしたナノ PE を実装する．本評価ボードでは，FPGA 上に最長一致検索専用ナノ PE を搭載する．ただし，本 FPGA はユーザにより書き換え可能であり，上記の動作を変更することを可能とする．FPGA の書き換えは PCI を通して JTAG により行う．プログラムは FPGA 内の EEPROM に保持される．

- 内部動作速度は約 10M Hz
- チップ間のデータ転送レートは 2.5Mpacket/sec
- 外付けメモリとして SRAM(7M access/sec)，あるいは SDRAM(1M access/sec) を排他使用．

本研究において，FPGA 上に最長一致検索処理を行うための，特徴命令，およびメモリインタフェースを備えた nPE を構成した．本 nPE は，既存のメモリ参照用 nPE である TBLnPE の回路構成を流用して，新規回路を追加実装した．内部構成を図 6.3 に示す．

入力パケットは，まず既存回路内の Entry PS において，本 nPE 内の処理回路を使用するかどうかを決定する．入力パケットが，本 nPE での演算を要する場合は，追加回路に転送される．追加回路は，定数制御 (CST) を行った上で，待ち合わせ (FC) に入る．発火条件がそろうと，HSP0，および HSP1 は，最長一致検索処理を含んだメモリ参照処理を行う．このとき，チップ内メモリを参照する処理 (HSP0) か，チップ外メモリを参照する処理 (HSP1) が選択実行される．HSP1 が参照するオフチップメモリは，SRAM，および

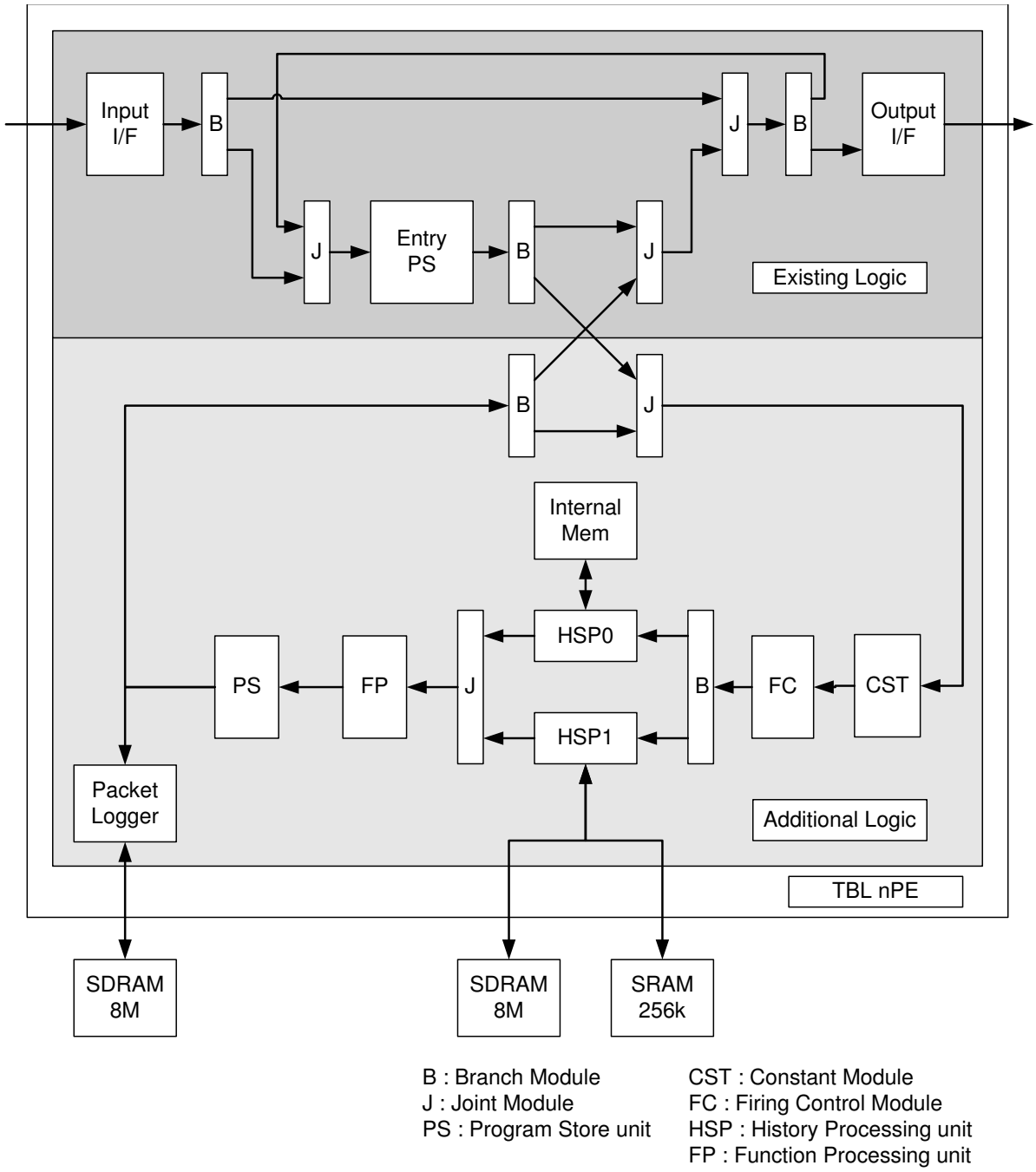

図 6.3　拡張 TBLnPE の内部構成.

表 6.1　機能拡張に伴うゲート数，および増加数.

	Total gates	Increases
Existing DDMP	481K	—
Add IPv4 Inst.	536K	27.5K
Add IPv4 & Pv6 Inst.	549K	34.0K

SDRAM があり，事前にパラメタ設定によって使用メモリが指定されてされている．演算された，パケットは次の処理を決定し (PS)，それによって，本 nPE 内を周回するか，他の nPE へ移動する．

6.3　パケットフィルタリング用回路の回路規模

まず，新たに追加した特徴命令を含む nPE に必要な回路規模を試算した．論理合成に Mentor Graphics[31] 社の Leonard Spectrum を用い，前述の FPGA をターゲットとしてゲートレベルの回路規模を見積もった．なお，論理合成ツールが出力するデータは FPGA の論理セル数となるため，これを論理セル辺りの使用ゲート数を基に換算した（表 6.1）．これより，約 6%程度のゲート数増加で IPv4 用の回路拡張で構成可能であることが判る．さらに，IPv6 用回路は IPv4 用回路に高々 1.4%のゲート数増加で拡張可能であることが明らかとなった．これは，1 チップ LSI として問題なく実現可能な規模である．

つぎに，回路性能評価として最長一致検索における SearchTrie 命令，CheckPrefix 命令，ないしその他基本命令の実行サイクル数，1 検索辺りの使用ノード数，および必要サイクル数を求めた．なお，SearchTrie 命令，および CheckPrefix 命令はそれぞれ，オンチップメモリ参照用とオフチップメモリ参照用があり，オフチップメモリ参照命令の実行サイクルはおおよそオンチップメモリ参照命令の 8 倍かかる．

IPv4 の IP アドレスに対する回路性能は表 6.2 のとおりであった．SearchTrie 命令において，LC-Trie の特徴より最大命令実行回数は 5 となり，平均命令実行回数は 2 となる．こ

のとき，最も命令実行サイクル数の多い CheckPrefix 命令に着目して，オンチップメモリ，およびオフチップメモリを使用した場合の最大スループット，平均スループット，および最小スループットを求めた．結果，表 6.3 の様になった．

同様に，IPv6 の IP アドレスにおける回路性能は表 6.4 に示すとおりであった．ここでも，処理負荷が大きいのはプレフィックス検査であり，検索レートは表 6.5 の様になった．

表 6.2　IPv4 の回路性能.

	Basic Cycle	Node/Loop			Cycle		
		max	avg	min	max	avg	min
Search Trie	1	5	2	1	5	2	1
	8	5	2	1	40	16	8
Check Prefix	2	5	1	1	10	2	2
	16	5	1	1	80	16	16
Basic Instruction	2	2	2	2	2	2	2

upper:use Internal memory lower:use External memory

6.4　最長一致検索の性能評価

パケットフィルタリングの性能を評価する上で，まず分類処理の中核的処理をになる最長一致検索機構について，現行 DDMP[33][34] と提案方式での性能比較を行う．DDMP は，現行の集積化技術を用いると仮定してシミュレーションにより求めた．このとき，プロセッサ内の処理対象データのビット長 32bit，自己タイミング型パイプラインの動作速度 140M データ/秒とした．各拡張テーブル参照用 nPE からアクセスするテーブルメモリの仕様は，オンチップメモリ (アクセス時間:7.1 nsec.)，オフチップメモリ (アクセス時間：10.1 nsec.) とした．また，現行 DDMP，提案方式ともに TBLnPE を 2 つ用いたときの性能を示す．

ルーティングテーブルが，すべてオンチップメモリに格納可能な場合，メモリアクセス遅

表 6.3　Prefix 検査用 nPE の検索レート (IPv4).

	max	avg	min
Internal Memory	16.0	80.0	80.0
External Memory	0.8	10.0	10.0

$$\text{Mpacket/sec} = 160MHz \div (Ncycle/packet)$$

表 6.4　IPv6 の回路性能.

	Basic Cycle	Node/Loop			Cycle		
		max	avg	min	max	avg	min
Search Trie	2	7	3	1	14	6	2
	8	7	3	1	56	24	8
Check Prefix	3	1	1	1	3	3	3
for 32bit	24	1	1	1	24	24	24
Check Prefix	3	7	0.5	0	21	1.5	0
for 64bit	24	7	0.5	0	168	12	0
Basic Instruction	1	3	3	3	3	3	3

upper:use Internal memory lower:use External memory

延は限りなく無視できる．しかし，MaeWest 等の大規模サイトのルーティングテーブルは巨大なため，オンチップメモリにすべて格納しきれない．そこで，最も頻繁にアクセスされる LC-Trie 検索木のレベル 1 のみを 2 つの拡張テーブル参照 nPE のオンチップメモリに分割格納し，レベル 2 からレベル 5 まではオフチップメモリに格納する．

このとき，IP アドレス検索時間は，内部テーブルへのアクセス遅延が外部テーブルへのアクセス遅延を越えないようにルーティングテーブルを分割配置すると，外部テーブルへのメモリアクセス時間に制約される．外部テーブルへのアクセスは LC-Trie 検索時とプレフィックス検査処理の両方で発生するため，検索レートはメモリアクセスレートに支配さ

表 6.5　Prefix 検査用 nPE の検索レート (IPv6).

	max	avg	min
Internal Memory	5.0	20.0	40.0
External Memory	0.6	3.3	5.0

$$\text{Mpacket/sec} = 160MHz \div (Ncycle/packet)$$

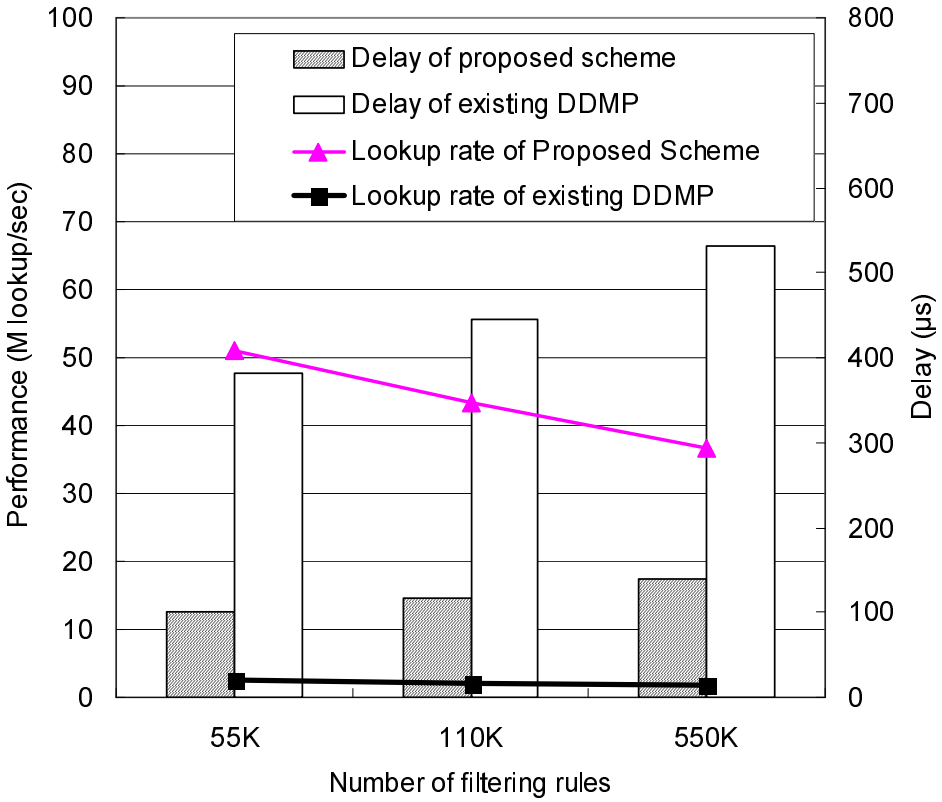

図 6.4　既存 DDMP と提案手法の検索遅延および検索レート（IPv4 アドレス検索)

れる.

まず，IPv4 アドレス最長一致検索に対して，エントリ数が 55K，110K，550K のとき，既存の DDMP で提案方式を実行した場合と専用命令を用いて実行した場合それぞれのスループットと実行遅延を測定したところ，図 6.4 のような結果が得られた．この結果から，専用命令を用いることで検索遅延は約 5 分の 1 程度に削減でき，検索レート 6 倍程度向上す

ることが明らかになった．また，エントリ数 55K のとき検索レートは 51M Lookup/sec となり，最短の TCP ACK パケット (384 ビット) の場合 18Gbps，平均的な IPv4 パケット (2000 ビット) の場合 96Gbps を受理可能なことを示している．

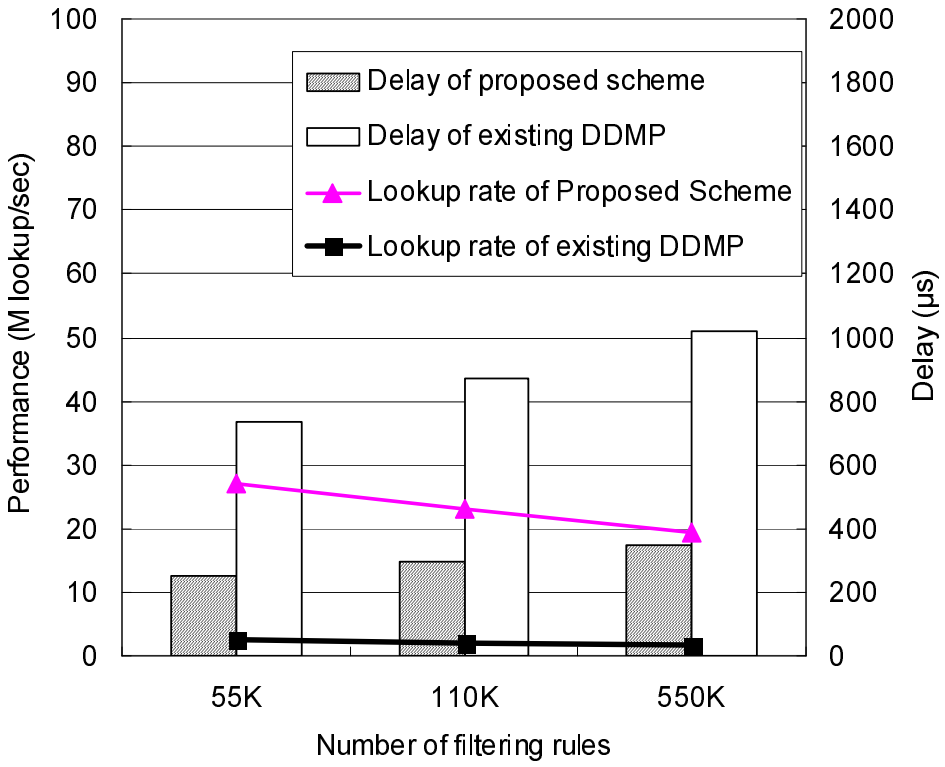

図 6.5　既存 DDMP と提案手法の検索遅延および検索レート (IPv6 アドレス検索)

同様に，IPv6 アドレス最長一致検索に関して評価を行ったところ，図 6.5 のような結果が得られた．その結果，やはり検索遅延は約 5 分の 1 程度に削減でき，検索レートは 6 倍程度向上した．すなわち，検索レートは 27M Lookup/sec であり，最短パケットで 10Gbps，平均長の IPv6 パケットで 54Gbps を受理可能なことが判った．

これは，DDNP が OC-768 の上の SONET/SDH リンクに関する IPv4 を提供可能であることを示唆する．

6.5　パケットフィルタリングの性能評価

パケットフィルタリングの性能評価として，エッジノードにおける大規模フィルタリングルールを対象とした 5-tuple パケット分類，およびパーソナルゲートウェイ向けに最適化したフィルタリングルールを対象としたフィルタリングを行った．

6.5.1　5-tuple パケット分類

まず, 5 次元のルールセットを用いたパケット分類を評価した．ルールセットは MaeWest のルーティングテーブルより擬似的に作成した．また，Index Jump テーブルの Index 生成には，生成したルールセットの先頭 2 つのキーフィールドを宛先/送り元 IP アドレスとし，残りを宛先/送り元ポート番号と仮定したとき，各アドレスフィールドに 4 ビット，各ポートフィールドに 2 ビットを適用する．Index Jump テーブルのサイズが，2，256，4096 の場合について試行した．

結果，図 6.6 に示すように，Index Jupm を用いることにより，分割後のルールセットのエントリ数を大幅に削減できていることが判った．これにより，検索すべき部分検索木が縮退され，検索性能の向上につながっている．分類レートは，12M Lookup/sec であり，平均パケット長 2000bit のとき約 24Gbps 程度の IPv4 パケットストリームを分類可能なことを示唆している．

6.5.2　パーソナルゲートウェイ向けパケットフィルタリング

次に，フィルタリングルールを最適化したときの，フィルタリング性能を評価した．

ネットワークエンド使用されるフィルタリングルールセットの平均的な傾向として，第 4 章で述べた以下の条件を基に擬似フィルタリングルールを作成した．

- プレフィックス長が 0，21，23，24，25，28，30，32，
- IP アドレスの約半分がワイルドカードである，
- ルールセットの約 5%のポート番号が範囲指定である，

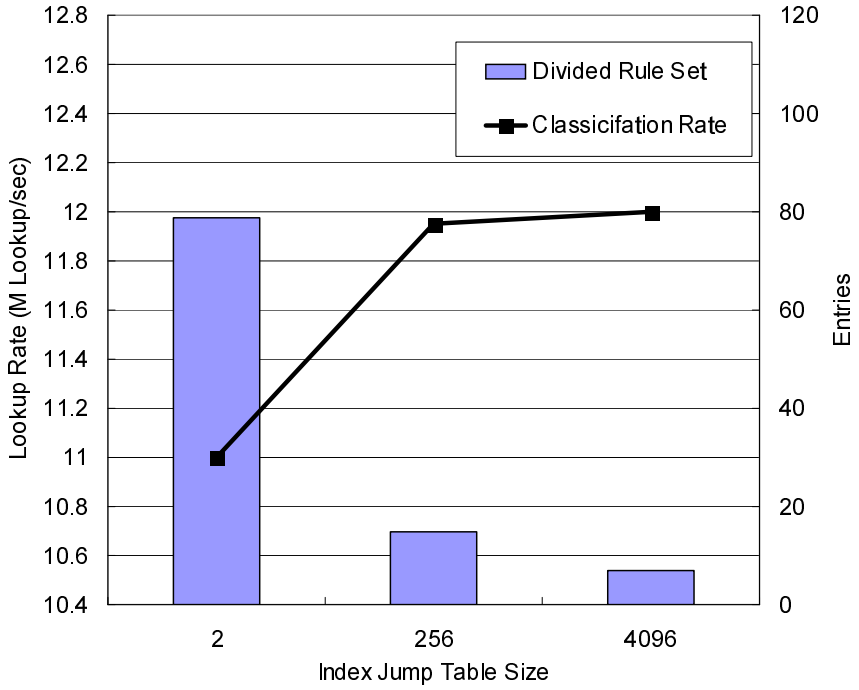

図 6.6　Index Jump テーブルの導入における分割ルールセットのサイズと分類レート

- トランスポート層フィールドの約 90%が 1-4 個の IP アドレスフィールドに重複している．
- 送信元ポート番号と宛先ポート番号は排他的に定義される．

このときの，IP アドレスに関するメモリアクセスは，図 6.7 に示す様に，最大メモリアクセス数はフィルタリングルール数が 25 のとき約 3.2 回，フィルタリングルール数が 50 を超えたときから 7 回で一定した．また，平均メモリアクセス数はフィルタリングルール数 25 からフィルタリングルール数 500 にかけて 3 回から 5 回に増加した．次に，送信元ポート番号に関するメモリアクセスは，図 6.8 に示すよう，最大メモリアクセス数はフィルタリングルール数が 25 のとき約 3 回，フィルタリングルール数が 50 を超えたときから 4 回で一定した．また，平均メモリアクセス数はフィルタリングルール数 25 からフィルタリングルール数 500 にかけて 1.5 回から 2.3 回に微増した．

また，宛先ポート番号に関するメモリアクセスは，送信元ポート番号の定義と図 6.9 に示すよう，最大メモリアクセス数はフィルタリングルール数が 25 のとき約 3 回，フィルタリ

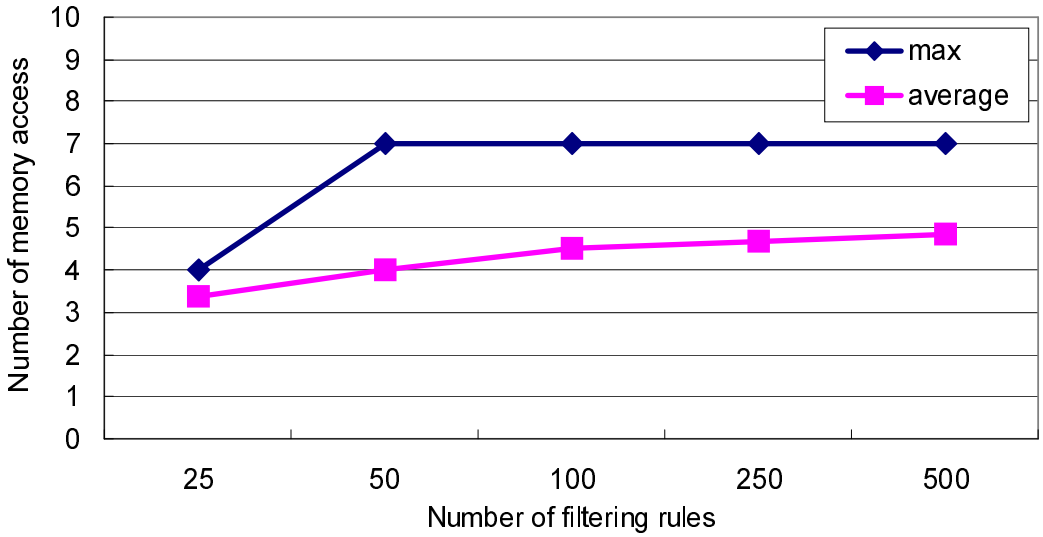

図 6.7　パーソナルゲートウェイ向けフィルタリングルールの IP アドレスに関するメモリアクセス回数.

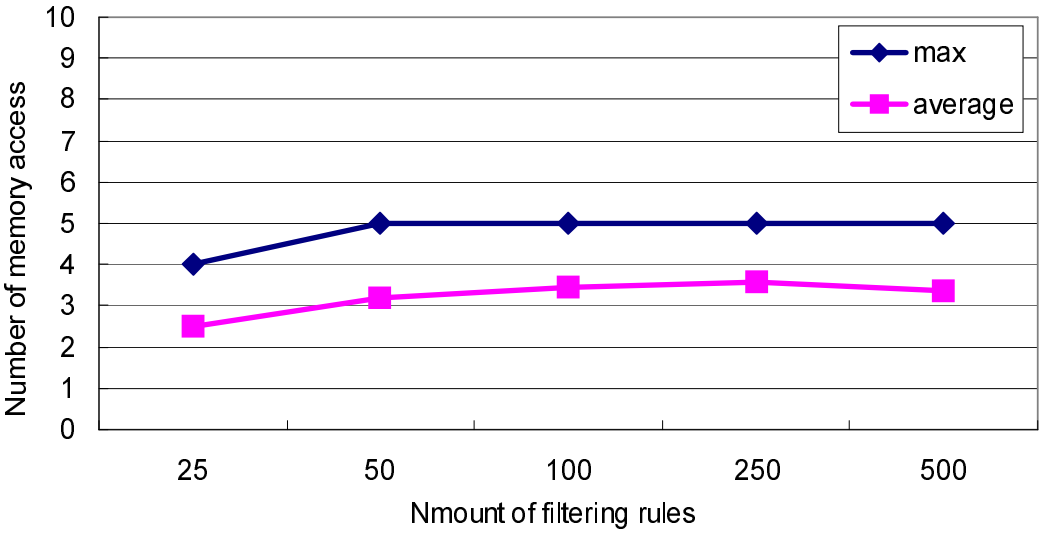

図 6.8　パーソナルゲートウェイ向けフィルタリングルールの宛先 TPC ポートに関するメモリアクセス回数.

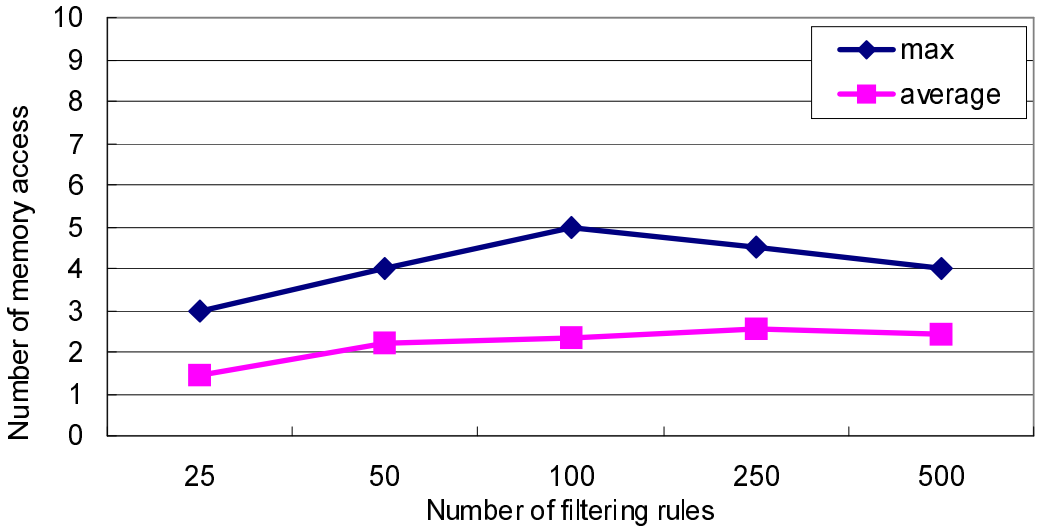

図 6.9　パーソナルゲートウェイ向けフィルタリングルールの送信元 TCP ポートに関するメモリアクセス回数.

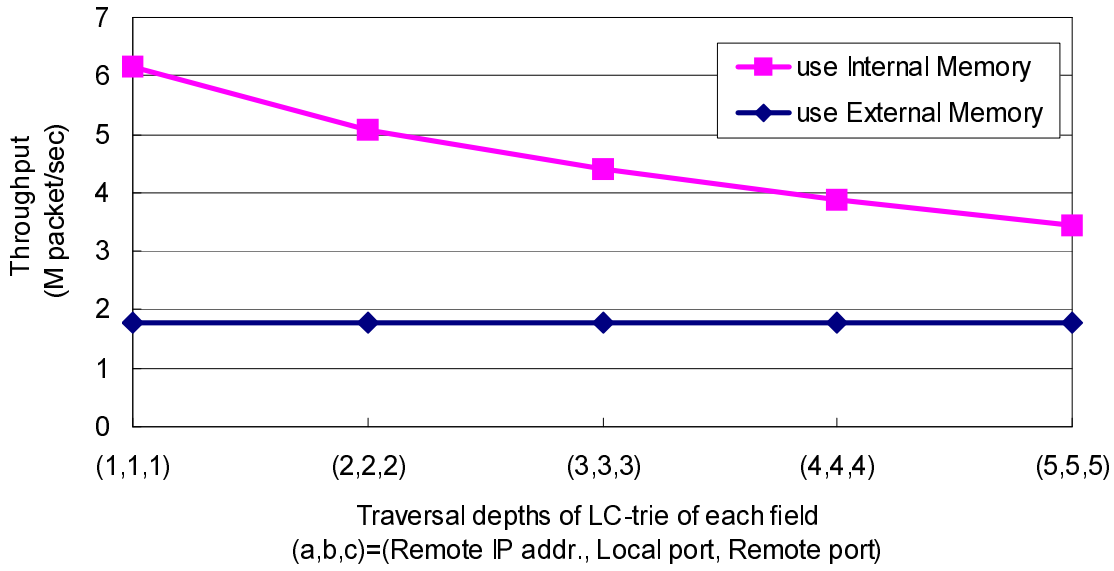

図 6.10　メモリアクセス回数の変化におけるフィルタリング性能

ングルール数が 50 を超えたときから 4 回で一定した．また，平均メモリアクセス数はフィルタリングルール数 25 からフィルタリングルール数 500 にかけて 1.5 回から 2.3 回に微増した．

このメモリアクセス回数を基に，処理速度を見積もったところ図 6.10 に示す結果を得た．オンチップメモリのみを使用すると 1 プロセッサで平均 3.8M Lookup/sec，最悪でも約 3M Lookup/sec の処理を達成できることが判った．これは，TCP パケットの最小サイズである 64 バイトのパケットストリームを受けた場合，約 1.5G bit/sec を達成できることを示す．

6.6　結言

本章では，プロトタイプシステムを用いて，

1. パケット分類用回路の回路規模
2. 最長一致検索の性能評価
3. パケットフィルタリングの性能評価

を行った．

1. において，提案回路の回路規模を試算した．IPv4 用回路は 6%程度の回路増加で構成できることが判り，IPv6 用回路はさらに約 1.4%の回路追加により拡張可能であることが明らかとなり，1 チップ LSI として問題なく実現可能な規模であることが判明した．

2. において，IPv4 アドレス検索に関して，約 51M Lookup/sec の検索レートを達成し，最短の TCP ACK パケットの場合 18Gbps，平均的な IPv4 パケットの場合 96Gbps を受理可能なことを示した．そして，IPv6 アドレス検索においては，検索レートは 27M Lookup/sec であり，最短パケットで 10Gbps，平均長の IPv6 パケットで 54Gbps を受理可能なことを示した．

3. において，Index Jump の導入により，検索空間を大幅に限定することが可能となった．また，5-tuple パケット分類性能として，平均パケットのとき約 24Gbps 程度の IPv4 パケットストリームを分類可能なことが判った．さらに，パーソナルゲートウェイ向けのパ

ケットフィルタリングルールを用いたとき，オンチップメモリのみを使用すると 1 プロセッサで平均 3.8M Lookup/sec，最悪でも約 3M Lookup/sec の処理を達成できることが判った．これは，TCP パケットの最小サイズである 64 バイトのパケットストリームを受けた場合，約 1.5G bit/sec を達成できることを示した．

なお，本研究においては IPv6 に関しては IPv4 と同様の基本ヘッダ情報を用いて同じフィルタリングルールを設定した上で評価を行った．一方で IPv6 には IPv4 にはない拡張ヘッダオプションが存在しており，それらを用いたルーティングが想定されるが，本研究においては考慮していない．したがって，今後の課題として IPv6 における拡張ヘッダオプションを対象としたフィルタリングルールを想定した評価を行う必要がある．

第 7 章

まとめ

本研究では，ネットワーク内の QoS 制御やセキュリティ制御を中央集約的に制御することが困難なネットワーク環境において，各端末が送受信するパケットストリームを検査し，選択的にフィルタリングする機構に着目した．

本論文の第 2 章においては，まず，ユーザの周囲に偏在する，モバイル端末，情報家電，センサー等の多くのネットワーク接続可能な機器が相互に協調動作し，ユーザの移動に追従して，周辺環境に応じてダイナミックにネットワークを形成することでより能動的なコンピューティング環境として，サラウンディング・コンピューティング環境を提起した．さらに，サラウンディング・コンピューティング環境におけるネットワークセキュリティについて示した．パケットフィルタリングについて延べた．本研究の基礎となるデータ駆動型ネットワークプロセッサ DDNP について提示する．そして，次世代の小規模分散コンピューティング環境において必須となりうる組み込み型パーソナルゲートウェイについて述べる．

第 3 章では，ネットワークプロトコル処理におけるレイヤ 3/4 パケットフィルタリングを行うためのパケット分類方式の比較検討をした．そして，パケット分類の高速ソフトウェア実現をするために，分類規則の照合にパイプライン並列化した最長一致アルゴリズムを用いることで高速化を図った，さらに，大規模なルールセットに対しては，ヒューリスティックに分割し，検索空間を限定することで，記憶空間の局所化を行う手法について述べた．

第 4 章では，パケットフィルタリングのセキュリティシーンにおける使用方法を提示し，セキュリティ保証のためのフィルタリング条件を示した．そして，端末に対して流入出するパケットストリームの特性を事前評価した．効率的なレイヤ 4 パケットフィルタリングを実現できることを示した．その結果，エンド端末における平均的なフィルタリングルール数で

ある 150 前後において，メモリ使用量は約 8KByte であった．これは，DDNP の 1 ナノプロセッサがもつオンチップメモリに格納できるサイズである．また，メモリアクセス回数は IP アドレスフィールド，トランスポート層フィールドの総和で平 7.5 回，最悪で 19 回必要であることが判った．

第 5 章では，提案方式をデータ駆動型ネットワークプロセッサ DDNP の機能として実現するにあたり，提案方式の更なる高速化を行うため，データ駆動型ソフト (複合命令) の提案を行い，それらを実装したメモリ参照に特化したナノプロセッサの構成を提案した．

第 6 章では，本研究で提案したパケットフィルタリング機構を実現するための回路規模の評価，および同機構の実証的性能評価を行うために試作した DDNP 評価ボードの構成について述べる．そして，評価ボードを用いた提案方式の性能評価，および提案方式の実装における回路規模の評価をした．特徴命令の追加による回路規模の増加は，IPv4 用で高々 6%であった．さらに，IPv6 用回路は IPv4 用回路に約 1.4%程度のゲート数増加で拡張できることが判った．また，評価の結果，大規模フィルタリングルールにおいて，10 プロセッサで 24G bit/sec の性能を達成できることが判った．また，パーソナルゲートウェイ向けのパケットフィルタリングにおいて，1 プロセッサで 1.5G bit/sec を達成できることを明らかにした．

近い将来，ユーザが使用する携帯端末，ユーザの周辺環境に偏在する各種デバイスおよびセンサー等を，相互結合することでユビキタス環境が実現するだろう．さらに，各端末が自律的に広大なインターネット上に分散するコンピューティング資源をそれぞれ用途に応じて階層的に利用するネットワークを構成し，ユーザの移動に追従して，周辺環境に分散されたプロセスがダイナミックにネットワーク上を遷移し，環境に応じた機能を提供するサラウンディングコンピューティング環境が実現されるだろう．こうした環境において，複雑雑多なプロトコル処理，QoS 制御，ユーザ認証，セキュリティ制御などをハードウェア資源の限られた端末上でソフトウェア実現することが求められる．そのため，筆者の属する研究室では，データ駆動型アーキテクチャによるデータ駆動型ネットワークプロセッサを検討している．本研究におけるパケットフィルタリング機構を含め，データ駆動型ネットワークプロ

セッサが次世代ネットワークにおけるエンド端末のパケットストリームを柔軟に授受する装置として活躍できることを期待して，本論文の結びとしたい．

謝辞

本研究の全過程を通じて，懇切丁寧に御指導，御鞭撻を賜った高知工科大学情報岩田誠教授に心より感謝の意を表する．

本研究の遂行ならびに本論文の作成にあたり，御指導，御助言を頂き，常に温かい御激励を下さった高知工科大学寺田浩詔元副学長に深謝の意を表する．

情報工学一般及び各専門分野に関しご教示頂き，研究全般にも御示唆，御助言を賜った，高知工科大学島村和典教授，清水明宏教授，西谷隆夫教授，および菊池豊助教授には副査として有益なご助言をしていただき深く感謝する．

筆者が高知工科大学に在籍して以来，種々の面でひとかたならぬ御助言，御激励をいただいた，高知工科大学岡田守学科長，坂本明雄教授，篠森敬三教授，木村義政教授，福本昌弘助教授，浜村昌則助教授，任向実助教授，妻鳥貴彦講師，酒居敬一講師に厚く御礼申し上げる．

本研究の実施に伴い，種々の技術的な御討論や有益な御助言を下さった九州大学大森洋一助手に心から御礼申し上げる．

日本テレコム (株) 孔令傑氏，林秀樹氏にはディスカッションなどを通じて最新の価値ある情報を提供していただくとともに研究についてご助言を賜った．ご協力に感謝する．

本研究で使用したデータ駆動型プロセッサ用開発ツール，シミュレータの提供，および評価ボードの試作協力をしていただいたシャープ株式会社の方々，ならびに株式会社ジーニックの関係者各位に深く感謝の意を表す．

高知工科大学岩田研究室ネットワークチームの同士として共に研究に励んだ，志摩浩氏，Ruhui Zhang 氏，白根裕太氏に感謝の意を表す．

筆者が，岩田研究室に在籍して以来，データ駆動プロセッサ，および種々の技術的な議論を行い，また研究室活動における苦楽を共にした三宮秀次氏に深謝する．

末筆ながら，高知工科大学岩田研究室の関係諸氏に深く感謝の意を表す次第である．

参考文献

[1] Mark Weiser, “Ubiquitous Computing,” IEEE Computer *Hot Topics*, August. 1993.

[2] Institute of Electronic and Electrical Engineers (IEEE). Wireless medium access control (MAC) and physical layer (PHY) specifications. Standard 802.11, 1999.

[3] C. Payne and T. Markham,“Architecture and Applications for a Distributed Embedded Firewall,” 17th Annual Computer Security Applications Conference, pp.329-336, December, 2001.

[4] T. K. Lee, S. Yusuf, W. Luk, A. Sloman, E. Lupu and N. Dulay, “Development Framework for Firewall Processors,” ICFPT 2002: IEEE Intl Conference on Field Programmable Technology (FTP),pp.352-355 December 2002.

[5] S. Ioannidis, A, D. K. Steven, M. Bellovin, and J. M. Smith, “Implementing a Distributed Firewall,” In Proc. of the ACM Conference on Computer and Communications Security (CCS), pp.190-199, November, 2000.

[6] Qiu, G. Varghese and S. Suri, “Fast Firewall Implementations for Software and Hardware-Based Routers,” Ninth International Conference on Network Protocols (ICNP’01), pp.0241, June 2001.

[7] D. D. Clark and J. T. Wroclawski, “The Personal Router Whitepaper,” MIT Technical Report, http://www.ana.lcs.mit.edu/papers/PDF/PR_whitepaper_v2.pdf.

[8] P. Faratin, J. Wroclawski, G. Lee and S. Parsons, “The Personal Router: An Agent for Wireless Access,” American Association for Artificail Intelligence, July, 2002.

[9] M. Roussopoulos, P. Maniatis and E. Swierk, “Person-level Routing in the Mobile People Architecture,” Proc. of USITS’99: The 2nd USENIX Symposium on Internet Technologies & Systems, Octorber, 1999.

[10] I. Carke, O. Sandberg, B. Wiley and T. W. Hong, “Freenet: A Distributed Anony-

mous Information Storage and Retrieval System," in Designing Privacy Enhancing Technologies, Lecture Notes in Computer Sience2009, pp.45-66, 2001.

[11] "Mobile Ad-hoc Networks (manet)," http://www.ietf.org/html.charters/manet-charter.html, IETF, June, 2005, work in progress.

[12] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC3561, http://www.ietf.org/rfc/rfc3561.txt.

[13] T. Clausen, P. Jacquet, Project Hipercom, "Optimized Link State Routing Protocol (OLSR)," RFC3561, http://www.ietf.org/rfc/rfc3561.txt.

[14] 南 正輝, 森川博之, 青山友紀, "動的でアドホックなネットワークサービスフレームワークの検討," 協調とモーバイルシンポジウム (DICOMO2000) 論文集. 情報処理学会, pp.13-18. Jun. 2000.

[15] L. Gwannap and B. Wheeler, "A Guide to Network Processors Third Edition," *MicroDesign Resources,* July,2002.

[16] J. Bolaria and B. Wheeler, "A Guide to Classification and Traffic Management Coprocessors Second Edition," *MicroDesign Resources,* May, 2002.

[17] P. Gupta and N. McKeown, "Algorithms for Packet Classification," IEEE Network, 15(2), pp.24-32, 2001.

[18] V. Srinivasan, S. Suri, G. Varghese and M. Waldvogel, "Fast and Scalable Layer Four Switching," *Proc. of ACM Sigcomm,* pp.203-14, September,1998.

[19] M.M.Buddhikot, S.Suri and M.Waldvogle, "Space Decomposition Techniques for Fast Layer-4 Switching," Protocols for High-Speed Networks 1999, pp.25-42, 1999

[20] P. Gupta and N. McKeown, "Packet Classification on Multiple Fields," *Proc. ACM SIGCOMM,* pp.147-60, September, 1999.

[21] V.Srinivasan, S.Suri, and G.Varghese, "Packet Classification Using Tuple Space Search," *In SIGCOMM,* pp.135-146, 1999.

[22] F.shafai, k.J.Schultz G.F.R.Gibson, A.G.Bluschke and D.E.Somppi, "Fully Parallel

30-MHz, 2.5 Mb CAM," *IEEE Journal of Solid-State Circuts,*vol.33, no.11, 1998

[23] T.V.Lakshman and D.Stiliadis "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching," Proc. ACM SIGCOMM '98, pp.203-214, 1998.

[24] 高橋直久, "フィルタ弁別関数の部分計算に基づく実時間パケット分類器," Proc. WIT'99, pp.190-197, 1999

[25] T.Y.C. Woo,"A modular Approach to Packet Classification: Algorithms and Results," *Proc. of IEEE INFOCOM,* vol.3, pp.1203-22, March,2000.

[26] A. Feldmann and S. Muthukrishnan, "Tradeoffs for packet classification," Proc. of IEEE Infocom, vol.3, pp.1193-202, 2000.

[27] S. Nilsson and G. Larlsson,"IP-address Lookup Using LC-Tries," IEEE J. on Sel. AIC,17(6),pp.1083-1092,1999.

[28] M.Degrmark, A.Brodnik, S.Carlsson, and S.Pink, "Small Forwarding Tables for Fast Routing Lookups," *ACM SIGCOMM '97,*27, pp.3-14, 1997.

[29] H.H.Tzeng and T.Przygienda, "On Fast Address-Lookup Algorithms," *IEEE Journal On Selected Areas In Communications,* 17(6), pp.1067-1081, 1999.

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

[31] Mentor Graphics - http://www.mentorg.co.jp/

[32] D. Morikawa, M. Iwata and H. Terada, "Superpipelined IP-Address Lookups on a Data-Driven Network Processor," *PDCS 2001,* pp.431-436, 2001.

[33] H. Terada, S. Miyata and M. Iwata, "DDMP's: Self-Timed Super-Pipelined Data-Driven Multimedia Processors," *Proc. of the IEEE,* 87(2), pp.282-296, 1999.

[34] 岡本俊弥, "データ駆動型メディアプロセッサ," 情報処理学会誌第 39 巻 3 号, pp.208-214, 1999.

[35] M.Waldvogel, G.Varghese, J.Turner, B.Plattner, "Scalable High Speed IP Routing Lookups," *ACM SIGCOMM '97,*pp.26-36, 1997

[36] 村瀬勉, 小林正好, “Longest Prefix Match キャッシュを用いた IP フォワーディングテーブル検索方法の提案,” 電子情報通信学会信学技報, SSE2000-20, pp.73-78, 2000.

[37] 羽賀雅則, 塩本公平, “ポインタキャッシュを用いた最長一致に基づくルーティングテーブル検索法,” 電子情報通信学会信学技報, SSE98-26, pp.7-12, 1999.

[38] M. Iwata, D. Morikawa, R. Zhang, W. Su, Y. Zheng, L. Kong, H. Terada, “Design Concept of An Embedded Data-Driven Firewall Processor,” International Conference on Next Era Information Networking NEINE’04, pp.80-87, Sep. 2004.

[39] D. Morikawa, Y. Shirane, R. Zhang, and M. Iwata, “Architecture of Embedded Data-Driven Personal Gateway Processor,” International Conference on Next Era Information Networking NEINE’05, pp.88-96, Sep. 2005.

[40] A.Wool, “A Quantitative Study of Firewall Configuration Errors,” *Computer*, **vol.37, Issue 6**, pp.62-67, June, 2004,

[41] R. Thomas, B. Mark, T. Johnson and J. Croall, “NetBounce: High-speed Legitimacy-based DDoS Packet Filtering,” Proceedings of the Third DARPA Information Survivability Conference and Exposition (DISCEX III), 2003.

[42] M. E. Lounavis, A. Kumar, H. Vin, R. Yavatkar and A. T. Campbell, “Directions in Packet Classification for Network Procesors,” Proceedings of the 2nd Workshop on Network Processors, pp.148-157, 2003.

[43] R.Zhang, M.Iwata, Y.Shirane, T.Asahiyama, W.Su, Y.Zheng, “High Speed Stateful Packet Inspection in Embedded Data-Driven Firewall,” International Conference on Next Era Information Networking NEINE’04, pp.322-329, Sep. 2004.

[44] Y. Shirane, R. Zhang, D. Morikawa and M. Iwata, “Parallel Implementation of Packet Reassembly on Embedded Personal Gateway,” NEINE’05, Sep. 2005.

[45] R. Zhang, D. Morikawa and M. Iwata, “A Time-Space Efficient Multi-Pattern Matching Scheme for Embedded Personal Gateway Processor,” NEINE’05, Sep. 2005.

[46] M.H.Overmars A.F.Stappen, “Range Searching and Point Location among Fat Objects,”

[47] 林秀樹, 岩田誠, 孔令杰, 米田進, 寺田浩詔, “超広帯域ネットワークにおける分散型マルチプロトコルルータ,” 情報研報 *2001-DSM-22,*pp.1-6, July 2001.

[48] 細美俊彦, 岩田誠, 林秀樹, 寺田浩詔, “自己同期パイプラインによる SLA 制御の実現法,” 電子情報通信学会全国大会, March 2002.

[49] 小林正好, 村瀬勉, 小倉直志, 栗山敦, “マルチギガビット IP フォワーダ用 50Mpps Longest Prefix Match 検索 LSI,” 電子情報通信学会信学技報, IN98-119, 7-12, 1998.

[50] L.Gwannap, ”A Guide to Security Processors”, *MicroDesign Resources,* May, 2002.

[51] 森川大智, 岩田誠, 寺田浩詔, “IP アドレス検索処理の超高速パイプライン実現法,” 第 62 回情報処理学会全国大会, 5T-07, 2001.

[52] P.Gupta, S.Lin, and N.McKeown, “Routing Lookups in Hardware at Memory Access Speeds,” *IEEE INFOCOM ’98,*10B-1, pp.1240-1247, 1998.

[53] 三屋光史朗, 長健二朗, 加藤朗, 村井純, “パケットサイズ分布からみた IP トラフィックの傾向,” インターネットコンファレンス 2000, 論文集, pp.47-56, 2000.

付録 A

関連業績

A.1 査読付き論文誌

1. D. Morikawa, M. Iwata, and H. Terada, “Super-Pipelined Implementation of IP Packet Classification,” Journal of Intelligent Automation and Soft Computing, Vol.10, No.2, pp.175-184, Aug. 2004.

A.2 国際会議

1. D.Morikawa, M.Iwata, H.Hayashi, and H.Terada, “Superpipelined IP-Address Lookups on a Data-Driven Network Processor,” in Proc. of Int. Conf. on Parallel and Distributed Computing and Systems, pp.431-436, Aug. 2001.
2. D. Morikawa, T. Matsumoto, M. Iwata, “Fast Packet Filtering in Data-Driven Embedded Firewall,” NEINE ’ 04, pp. 361-366, Sep. 28, 2004.
3. M. Iwata, D. Morikawa, R. Zhang, W. Su, Y. Zheng, L. Kong, H. Terada, “Design Concept of An Embedded Data-Driven Firewall Processor,” International Conference on Next Era Information Networking NEINE’04, pp.80-87, Sep. 2004.
4. D. Morikawa, Y. Shirane, R. Zhang, and M. Iwata, “Architecture of Embedded Data-Driven Personal Gateway Processor,” International Conference on Next Era Information Networking NEINE’05, pp.88-96, Sep. 2005.
5. Y. Shirane, R. Zhang, D. Morikawa, M. Iwata, “Parallel implementation of Packet

Reassembly on Embedded Personal Gateway," International Conference on Next Era Information Networking NEINE'05, pp.548-553, Sep. 5, 2005.

6. R. Zhang, D. Morikawa, M. Iwata, "A Time-Space Efficient Content Inspection Engine for Embedded Personal Gateway Processor," International Conference on Next Era Information Networking NEINE'05, pp.97-107, Sep. 5, 2005.

A.3　国内学会

1. 森川大智, 大森洋一, 岩田誠, "データ駆動方式による IP アドレスの一実現法," 平成 12 年電気関係学会四国支部連合大会, pp.207, Oct. 2000.
2. 森川大智, 岩田誠, 寺田浩詔, "IP アドレス検索処理の超高速パイプライン実現法," 情処学会第 62 回 (平成 13 年前期) 全国大会講演論文集 (3), pp.395, Mar, 2001.
3. 森川大智, 岩田誠, "データ駆動型最長一致検索機構の超高速パケット分類処理への応用," FIT 2002, pp. 63-64, Sep, 2002.
4. 森川大智, 岩田誠, "データ駆動型ネットワークプロセッサにおける高速パケット分類処理," SWoPP2003, 情報処理学会研究報告 Vol.2003, No.84, pp.37-42, Aug. 2003.
5. 志摩浩, 森川大智, 岩田誠, 11W-CDMA 用ベースバンド処理のデータ駆動型プロセッサによる実現法," 電気関係学会四国支部連合大会, 12-8, Oct. 2002.
6. 宮崎康徳, 志摩浩, 森川大智, 岩田誠, "ターボ符号復号器のストリーム並列処理に関する検討," 電気関係学会四国支部連合大会, 12-9, Oct. 2002.