修士論文 2002年度(平成14年度)

高速なパケット転送のための経路表構築手法に関する研究

慶應義塾大学大学院政策・メディア研究科

頴原桂二郎

修士論文要旨　2002年度 (平成14年度)

高速なパケット転送のための経路表構築手法に関する研究

本研究では、各ルータにおいて経路検索に使用する経路表を整合性を損わずに縮小し、パケット転送時の経路検索を高速化する機構を設計し実装した。

現在、インターネットは急激な成長を続けている。利用者数の増加、利用用途の拡大によってネットワークの規模が拡大すると同時に、一般利用者が利用するアクセスリンクの高速化、それに伴うマルチメディアアプリケーションの普及によって利用者当たりの使用帯域も大幅に増加している。

これら様々な要因により、インターネット上に流れるトラフィックは増加の一途を辿っており、ルータはより多くのトラフィックを制御する必要性に迫られている。ルータはIPパケットごとに宛先IPアドレスから最適なネクストホップを検索する必要があるため、ルータの高速化には経路検索の高速化が必要である。

既存の研究では、経路表から最長一致の経路を検索する機構の高速化と最適化が行われてきた。本研究では、各ルータが持つ経路の量に着目し、検索対象にする経路の数を最小限に抑えることにより経路検索の高速化を実現した。

現在のルータが持つ経路表は、そのルータが動作させているルーティングプロトコルによって得られた経路と静的に設定された経路の全てを含む。しかし、各ルータがパケットを転送する可能性があるネクストホップの数は数台から数十台であり、同じプリフィックスにマッチする複数のプリフィックスが同じネクストホップを示す場合が多くある。このことから、既存のルータは必要以上の経路エントリに対して最長一致の経路検索を行っていると言える。本機構では、経路制御プロトコルや管理者が構築する経路表とは別に、不必要な経路エントリを削除した経路検索専用の経路表をルータが動的に作成する。

本機構の評価としてフルルートを保持するルータ、国内の経路を保持するルータ、AS内部の経路を持つルータの状況を適用した。その結果、本機構適用後の経路表は既存の経路表よりも大幅に縮小され、経路検索に必要な時間も大きく短縮されることが分かった。

キーワード

1 インターネット　2 経路　3 経路検索　4 ルータ　5 高速ネットワーク

慶應義塾大学大学院政策・メディア研究科

頴原 桂二郎

Abstract of Master's Thesis Academic Year 2002

Building Efficient Routing Table for Fast Route Look-up

This research designed and implemented a mechanism to minimize the routing table in each router when searching for a route, and fasten the time needed to search for a route when transfering a packet.

Today, the Internet is rapidly growing. As the scale of the network expand with diversity in it's usage and an increase in users,

Meanwhile, user's access links have fasten and the bandwidth have also increased per user with correspondance with the increase in multi-media applications.

With these factors, there has been a rapid increase in the traffic in the Internet, and the routers are required to control even more traffic. Since the routers must search for the most optimize next hop from the destination field in each packet, and to fasten the process in each router requires to fasten the time to search for each route.

Existing research fasten the time requied to search for a route by a method of longest matching from the routing table. By focusing on the amount of routes handled of each router, this research have succeeded in fasting the search time by minimizing the routes in the routing table.

In the routing table in existing routers, all routes from the routing daemon is included. However, the number of next hop is from a few nodes to tens of nodes, therefore existing routes have unneccessary routes. In this research, we created a dynamic routing table without unneccessary entries.

For evaluation, this system was set on the routers that has full-route, routing infomation for domestic network, and routing infomation for AS. As a result, It was proved that the routing table was scaled down and the time taken for searching the route was drastically shortened.

Key Word

1 Internet 2 Route 3 Routing Table Lookup 4 Router 5 High speed network

Keio University Graduate School of Media and Governance

Keijiro Ehara

目 次

図 目 次

表 目 次

第1章　序論

1.1　インターネットの成長と拡大

現在、インターネットは急激な成長を続けている。また、様々な要因からインターネットに流れるトラフィック量も増加している。

第一に、利用者数の増加があげられる。特にアジア太平洋地域、南アメリカなど、これまでインターネットの普及が遅れていた地域では全年比 1.5 倍以上の速度で利用者数が増加している [12]。第二に、アクセス技術の進歩による一利用者あたりの使用帯域の増加があげられる。これまで一般家庭や小規模オフィスからのインターネット接続は電話線を利用することが多かった。しかし、ADSL や CATV 網によるインターネット接続サービスの普及、FTTH など地域限定で利用できる超高速通信回線の整備などにより、一般利用者からのアクセスリンクが広帯域化した。第三に、アクセスリンクが広帯域化したことで、これまでのメールや WWW を主体とした比較的消費帯域が少ないアプリケーションだけでなく、音楽や映像などマルチメディアコンテンツを閲覧するアプリケーションが頻繁に利用されるようになった。また、P2P アプリケーションによるユーザ間での情報共有も盛んになった。

この他にも、携帯電話など新しい端末がインターネットに接続したり、IP 電話の登場のように既存の通信媒体がインターネットに置き換えられるなど、インターネットの利用方法がより多様化していることも、インターネットに流れるトラフィック量を増加させている。

1.2　インターネット上の経路数とトラフィック量の増加傾向

インターネットは AS(Autonomous System:自律システム) と呼ばれる「単一の経路制御ポリシを持つネットワークの集合」によって構成されている。インターネットの経路制御は AS 内の経路制御と AS 間の経路制御に分離している。AS 内

の経路情報は AS 内だけで管理し、AS 間の経路情報は隣接する AS が互いに経路情報を交換することで世界的に伝搬する。世界的な AS 間の経路情報の総数をフルルートと呼ぶ。ISP は経路集約を目的として、マスクとしてまとめられる形で IP アドレス空間の割り振りを受けており、これによってフルルートの増加は抑制されている。しかし、APNIC[2] によると 2002 年 6 月 30 日現在で 113,294 の経路が存在する。

トラフィック量の増加傾向を示す例として NSPIXP-2(Network Service Provider Internet Exchange Point[9]) における総トラフィック量の変化を図 1.1 に示す。NSPIXP-2 は WIDE プロジェクトによって運営されており、商用インターネットを相互に接続する場合の問題点について実証的な手法を用いて研究をすすめている。NSPIXP-2 には現在 50 以上の共同研究組織が実験に参加している。2002 年 6 月現在、NSPIXP-2 では 10Gbps 近いトラフィックが流れている。前年度 1 月のトラフィック量は 3Gbps 程度であり、18ヵ月で 3 倍以上の増加である。

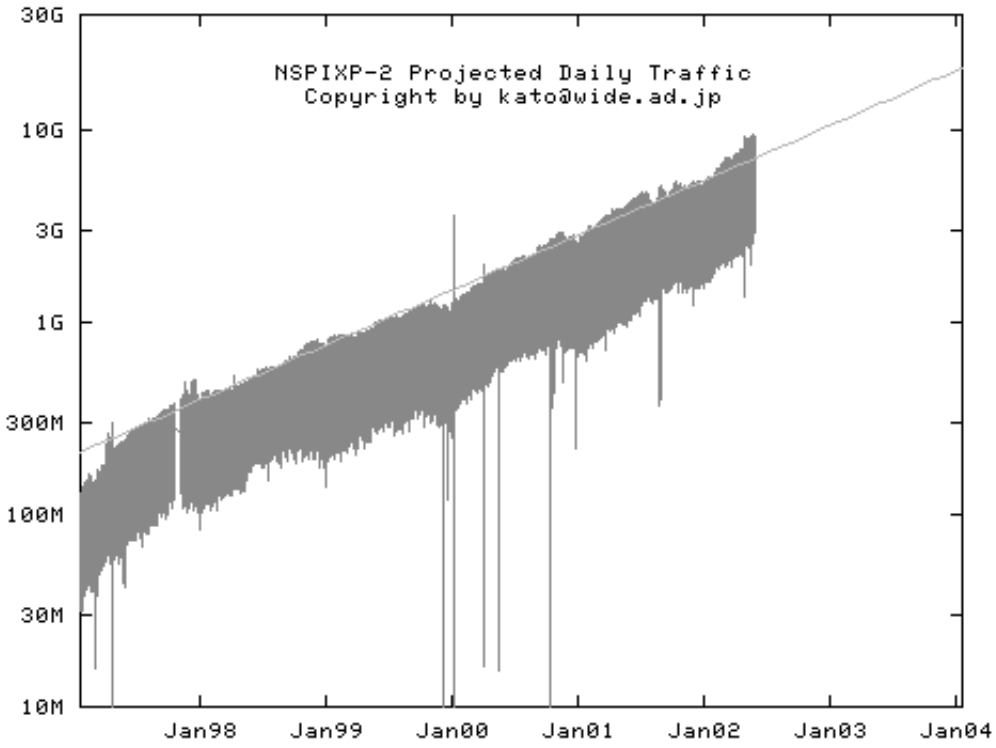

図 1.1: NSPIXP-2 における総トラフィック量の変化

1.3 高速なパケット転送の必要性

このような急激なトラフィック量の増加に際し、ルータにおけるパケット処理能力の向上が強く求められている。ルータはIPパケットごとに宛先IPアドレスへの最短経路を検索し、最適なネクストホップにパケットを転送する。このため、ルータは必要なトラフィックを転送するために十分な帯域を持つネットワークインタフェースと、最小の時間でネクストホップを決定する計算能力が必要である。このうち、ネットワークインタフェースはハードウェアの高速化や多重化技術によって対応されている。ルータの計算能力に関しては、CPUの高速化と共に経路検索手法の効率化が必要である。例えば、64byte長のIPパケットによる100Mbpsのトラフィックを転送するには、195,000pps(packets per second)の速度でIPパケットを転送する必要がある。これは1パケットあたり平均5usで処理する計算である。

1.4 本研究の目的

本研究では、ルータが持つ経路表を各ルータにおいて単純化し、経路検索を高速化することを目的とする。既存の研究では経路検索手法の効率化が考えられてきた。本研究はインターネットの膨大な経路数が経路検索処理に及ぼす影響に着目し、検索対象の経路を単純化し減少させ、この問題を解決する。本研究で提案する機構は経路の検索手法そのものには変更を加えないため、様々な検索技術と併用し使用できる。

1.5 本論文の構成

第2章では、現在のインターネットにおける経路表の特徴を述べる。第3章では、経路検索を高速化する技術について述べる。第4章では、本研究が提案する手法について述べ、第5章ではその実現方法の設計と実装を述べる。第6章では、WIDEプロジェクト[10]のバックボーンネットワークにおいて本機構を適用した結果を述べ、評価とする。第7章では、以上の議論をまとめ結論を述べる。

第2章　現状と問題点

本章では、インターネットにおける経路集約の問題点を述べる。

2.1　経路の集約

2.1.1　インターネットの経路

現在、インターネットの経路はCIDR(Classless Inter-Domain Routing,[3]) と呼ばれる運用方法が用いられている。初期のインターネットではネットワークをクラスA、B、Cと分け、クラスごとにネットワークアドレス部とホストアドレス部が決められていた。

- クラスA
  - ネットワークアドレス部8bit + ホストアドレス部24bit
- クラスB
  - ネットワークアドレス部16bit + ホストアドレス部16bit
- クラスC
  - ネットワークアドレス部24bit + ホストアドレス部8bit

しかし、クラスを用いたアドレス空間の割り当てでは、規定された大きさのIPアドレス空間しか配布できないため、非効率である。CIDRはIPアドレスのネットワークアドレス部とホストアドレス部の長さを任意に決められる考え方で、1993年頃から使用され始めた。CIDRを用いたアドレス空間をプリフィックスと呼び、192.168.0.0/24のように表記する。この場合、ネットワークアドレス部が24bitであり、ホスト部が残りの8bitになる。これを図2.1に示す。

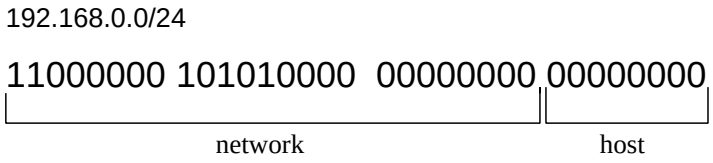

図 2.1: CIDR によるアドレス部の分割

CIDR を用いることで、IP アドレス空間を必要な大きさだけ各団体に割り当てできるようになった。

また、割り当てられたアドレス空間を効率的に利用するため、ルータには可変長サブネットマスク (VLSM: Variable Length Subnet Mask, [1]) の機能が実装されている。この機能では、一つのネットワークをサブネットに分割する際に、サブネットごとに異なる長さのマスク長を使用する方法である。例えば、一つのクラス C のネットワークをサブネットに分割する際に、/27 と/26 のネットワークを同時に設定できる。

しかし、CIDR と VLSM の導入によって、アドレスのネットワーク部とホスト部の割り当てをアドレス番号から識別できなくなった。このため、インターネットの経路はネットワークアドレスとネットマスクの組み合わせ (プリフィックス) で伝搬されている。現在、インターネット上のルータは最適な経路を検索するために最長一致の経路検索を行う。つまり、あるアドレスを含む全てのプリフィックス の中でプリフィックス長が最も長い経路を最適な経路とする。その理由は、同じアドレス空間を示す複数のプリフィックスが存在する可能性があるからである。例えば、あるルータの経路表において 192.168.0.0/24 の経路と 192.168.0.0/27 の経路が同時に存在した場合、よりプリフィックス長が長い 192.168.0.0/27 の経路が優先される。この考え方から、現在 default 経路は 0.0.0.0/0 の経路 (全てのアドレス空間にマッチするプリフィックス長 0 の経路) と考えられている。最長一致の経路検索の問題点は、ネットワークアドレス部固定の経路検索に比べ、ルータに多くの処理を必要とする点である。

2.1.2 プリフィックスを用いた経路の集約

ルータにおける経路検索処理を軽減するため、インターネットでは様々な手法で経路の集約が取り組まれている。連続したプリフィックスは、よりプリフィックス長が短い一つの経路に集約できる。例えば、192.168.0.0/24、192.168.1.0/24、

192.168.2.0/24、192.168.3.0/24 の 4 つの連続したプリフィックスは 192.168.0.0/22 という一つのプリフィックスに集約できる。これを図 2.2 に示す。

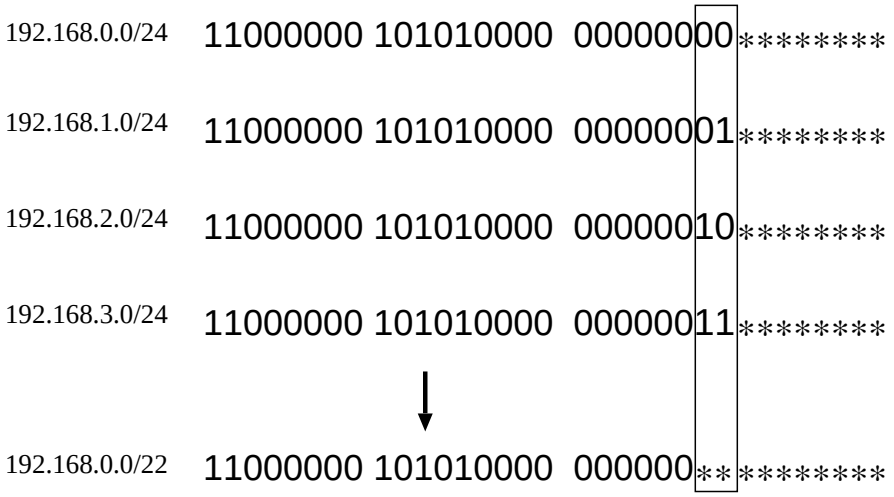

図 2.2: プリフィックスの集約

このような経路の集約は、ある ISP(Internet Service Provider) から外部の ISP へ BGP(Border Gateway Protocol[11]) によって経路情報を広告する際に使用される。BGP における経路の集約方法は 2.2.2 に述べる。

2.2 経路集約の運用技術

2.2.1 IP アドレスの分配方法

現在、IP アドレス空間は IANA(Internet Assigned Numbers Authority) が管理している。IANA は世界の地域ごとに設立されている地域インターネットレジストリ (RIR: Regional Internet Registry) に IP アドレス空間の割り振りを行う。各 RIR は国別インターネットレジストリ (NLR: National Internet Registry) にアドレスに割り振りを行う。アジア太平洋地域の RIR は APNIC(Asia Paciffic Network Information Center) であり、日本の NLR は JPNIC(Japan Network Infomation Center) である。各 ISP はローカルインターネットレジストリ (LIR: Local Internet Registry) となり、NLR から IP アドレスの割り振りを受ける。

一般的に、NLR から ISP に割り当てられるアドレス空間は/16、/19、/22 など

の比較的大きなアドレス空間である。ISP はこの中からエンドユーザに必要なアドレス空間を割り当てる。以上から、ISP から他の ISP に BGP で経路情報を広告する際は NLR から割り振られたプリフィックスだけを広告すればよい。IP アドレスを連続したプリフィックス長で分配することによって経路が集約される様子を図 2.3 に示す。

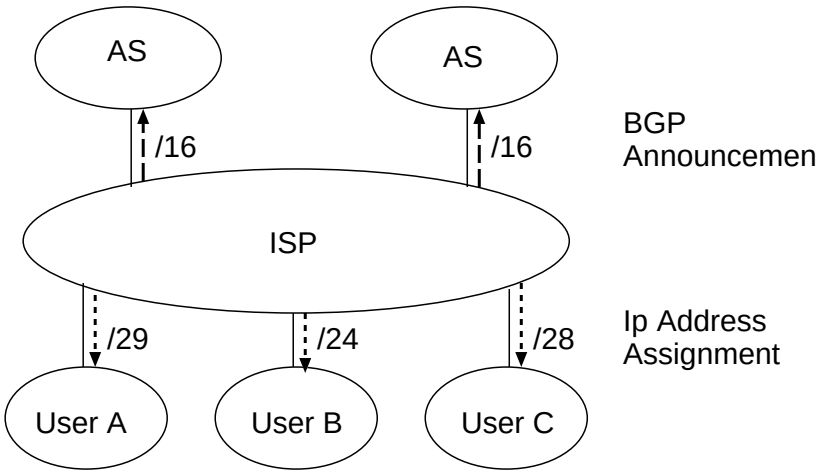

図 2.3: IP アドレスを連続したプリフィックス長で分配

現在割り振られている IP アドレス空間には、エンドユーザが LIR 以外の NIR や RIR から直接割り振られたアドレス空間も存在する。これをプロバイダ非依存アドレス (PI アドレス: Provider-Independent address) と呼ぶ。例えば、日本の場合 1993 年に正式に CIDR の割り振りを開始したため、それ以前からインターネットを利用していた組織はプロバイダ非依存アドレスを持っている場合がある。このような組織はプロバイダと接続した場合も独自のアドレス空間でインターネットに接続するため、プロバイダ側で経路を集約できない場合が多い。

2.2.2 BGP による経路集約

ISP が BGP により外部 ISP に経路を広告する場合、BGP の経路集約機構を用いて複数のプリフィックスをまとめる運用が行われている。例として、ISP が他の ISP にマルチホームした場合を述べる。図 2.4 では、AS1 はトランジット AS である AS2 にマルチホームしている。

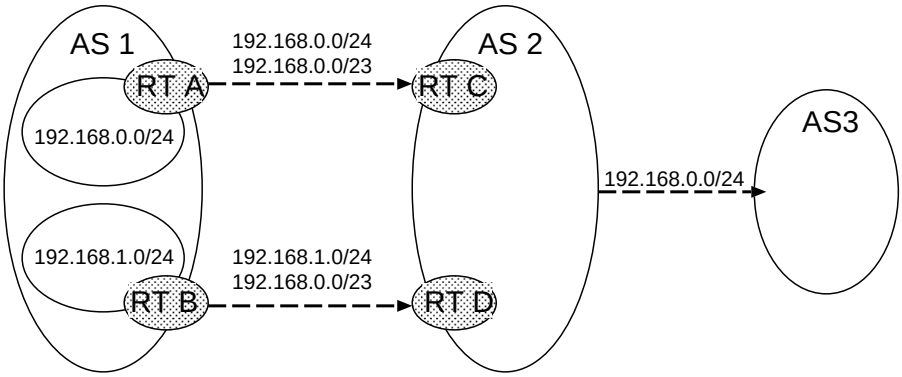

図 2.4: BGP による経路集約

AS1 は 192.168.0.0/24、192.168.1.0/24 の 2 つのプリフィックスを保持している。192.168.0.0/24 はルータ A 付近のネットワークであるため、ルータ A を通して外部 ISP と通信したいとする。また、192.168.1.0/24 はルータ B 付近のネットワークであるため、ルータ B を通して外部 ISP と通信したいとする。この場合、AS1 は AS2 に対してルータ A、ルータ B からこれらのプリフィックスを別々に広告する。また、AS1-2 間の回線のうち一本がダウンした場合に残りの回線を通して AS1 内の全てのプリフィックスが外部 AS と通信できるように、192.168.0.0/23 という集約経路も広告する。AS2 では AS1 から 192.168.0.0/24、192.168.1.0/24、192.168.0.0/23 という 3 つの経路を受け取っているが、AS2 が他の AS に AS1 からの経路を再広告する場合には集約経路である 192.168.0.0/23 だけを広告すればよい。

2.2.3 punching hole 問題

2.2.1 で述べた通り、

ISP は経路集約を目的として連続プリフィックスで IP アドレスの割り当てを受けている。しかし、AS 番号を取得するほど大規模でないネットワークがマルチホームを行うために、割り当てをうけた ISP 以外からもそのプリフィックスを流す場合がある。図 2.5 では、ISP A が 192.168.0.0/20 のアドレス空間を保持している。ユーザ X は ISP A から 192.168.0.0/24 のアドレス空間を割り当てられている。ISP A は集約アドレスである 192.168.0.0/20 だけを他の ISP に広告する。ここで、ユーザ X はマルチホームのために ISP B にも接続したとする。ユーザ X は既に ISP A からアドレスを割り当てられているため、ISP B に 192.168.0.0/24 の経路を

広告する。ISP B は 192.168.0.0/24 のアドレス空間をそれ以上集約できないため、この経路をそのまま他の ISP に広告する。この結果、NIR によって ISP への連続プリフィックスが割り当てられた環境でも各ユーザに分配されたより細かな経路がインターネット上に流れてしまう。

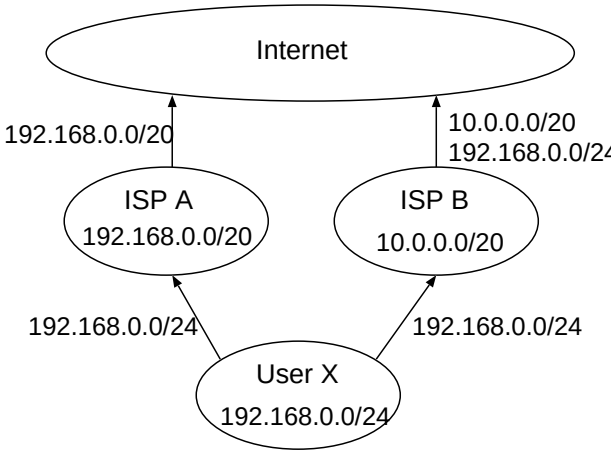

図 2.5: punching hole 問題

図 2.6 は現在の BGP フルルートに含まれるプリフィックス長ごとの経路数を示す。punching hole された経路は/24 のプリフィックス長を持つ経路が多く観測されており、図 2.6 からも /24 の経路が占める割合が非常に多いことが分かる。punching hole の影響でフルルートが増大しているのは経路数を抑制する上で大きな問題である。

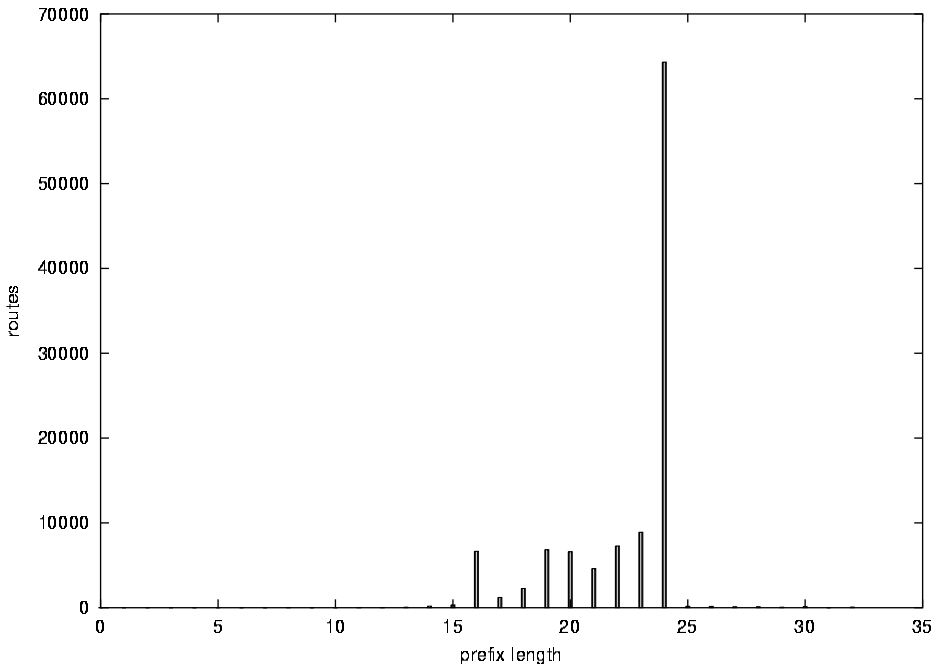

図 2.6: フルルートのプリフィックス長分布

2.3 運用による経路集約の限界

本章で述べた通り、インターネットにおける BGP 経路は IP アドレス割り当て手法や各 AS における BGP 運用によって増大が抑制されている。しかし、CIDR 以前に割り振られた IP アドレス空間が存在するため、ISP ごとに経路を集約する手法には限界がある。また、プリフィックスの連続性と AS 間の接続には関連がないため、BGP の運用の中で集約できる経路数は限られている。更に、punching hole 問題のように ISP に連続プリフィックスで割り当てられたアドレス空間の一部が個別の経路としてインターネットに流れる問題が発生しており、BGP フルルートは緩やかな増加傾向にある。本研究では、このような運用技術による経路の集約によって経路数を減らす手法ではなく、各ルータにおいて経路制御プロトコルから得られた経路を集約し、経路検索の対象になる経路数を減少させるアプローチを用いる。

第3章　関連研究

3.1　経路表のデータ構造

ルータは、IP パケットを転送するために、経路表を利用する。経路表は、宛先アドレスを検索キーとして、パケットが次に転送されるべきルータを決定するためのデータベースである。

ルータは、IP パケットを転送するたびに経路表を検索するため、経路表のパフォーマンスがルータのパフォーマンスに大きく影響する。毎秒数百万のパケットを転送することが求められるルータでは、経路表の検索速度が大きな問題となる。また、経路の追加および削除は、経路探索に比べて極めて頻度が少ないため、追加または削除に要する時間を犠牲にしても、探索速度が向上することが望ましい。

本章では、経路表の検索速度を向上するために行われてきた、経路表データ構造についての先行研究を述べる。

3.2　Radix tree, Trie

経路表のデータ構造として現在最も一般的となっているものが、radix tree または Trie[4] と呼ばれるものである。これは、BSD 系 UNIX カーネルの経路表に採用されている。

radix tree では、検索の各ステージにおいて、桁毎の検索を行うものである。内部データ構造は、ある桁数までにマッチする経路エントリを保有するノードで構成された木となっている。図 3.1 に radix tree 内部のデータ構造を示す。

図中では、radix tree の中間ノードを◯で、保持される経路情報を□で表した。中間ノードは次の桁の bit が 1 か 0 かによって下流のノードに分岐される。それぞれの中間ノードは、該当するプリフィックス長の経路情報を保持できる。

検索は、IP パケットの宛先を検索キーとし、radix tree のルート (トップノード) から中間ノードをたどる形で行われる。マッチする下流ノードを持たない中間ノードまで検索され、検索中の一番深い中間ノードが持つ経路情報が、該当する経路情報となる。

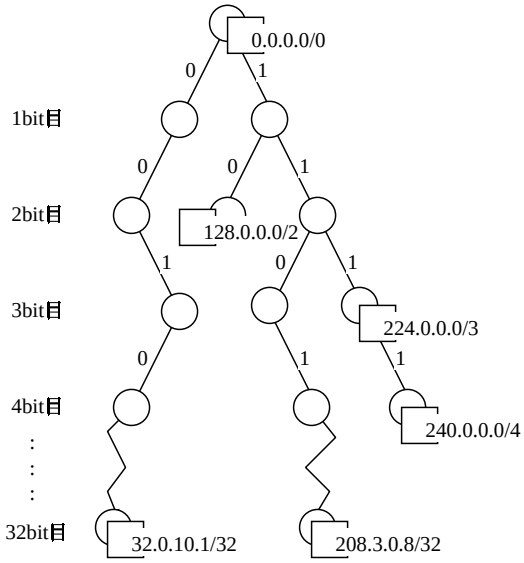

図 3.1: Radix tree の構造

例えば、224.1.0.1 をキーとした検索は、224.1.0.1 を 2 進数で表したもの、11100000 00000001 00000000 00000001 の全ての桁毎に、1 であれば右の子、0 であれば左の子をたどっていく。ルートから右右右左 左左…とたどっていこうとするが、図中では 224.0.0.0/3 の中間ノードから左にたどれないため、この中間ノードがマッチする経路情報を持つことになる。

radix tree での利点は、どのような宛先アドレスに対する検索も、たかだか桁数回の検索回数しか要さないことにある。IPv4 パケットの転送時には、最悪でも宛先アドレスの長さ、つまり 32 回の検索で、該当する経路情報を取得できる。

また、経路表内の経路が集約されるほど、この検索回数も減少する。下流のノードを刈り込むほど、早い段階で検索が終了するからである。

3.3 Patricia tree

Patricia tree[7] は、前述の radix tree の改良版である。radix tree 内部の不要な中間ノードを削除し、検索回数とメモリ使用量を改善したものである。

図 3.1 で挙げた radix tree のデータを patricia tree で表した場合の内部構造を、図 3.2 に示す。

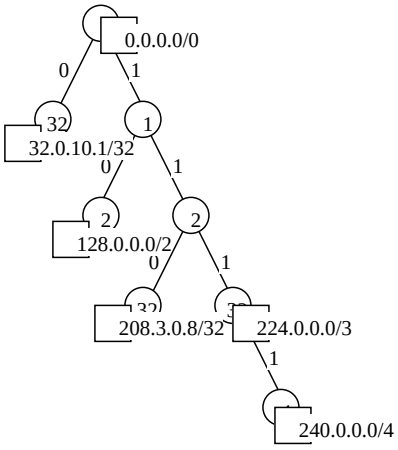

図 3.2: Patricia tree の構造

3.4 Lulea

Lulea[5] と呼ばれる radix tree の改良版では、1bit 毎に検索をする必要がないことが示された。

radix tree では、検索キーとなる宛先アドレスの 1bit 毎に、下流のノードへのポインタが保持される。これは、アドレスの 1bit を、長さが 2 である配列に展開した、と言える (図 3.3)。

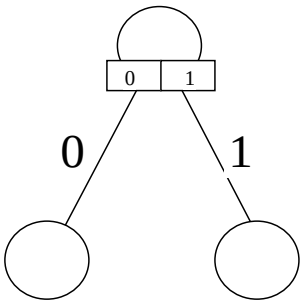

図 3.3: Radix tree の中間ノードの構造

Lulea では、アドレス内の複数の bit を、あらかじめ展開しておく。展開は、取り得る全ての値に対する下流ノードへのポインタを保持する。その値に対する下流ノードが存在しない場合は、現在のノード自身がマッチするエントリとなる。

例えば、2bit 毎に展開しておく場合は、2bit で取り得る値すべて、00 01 10 11

に対して、それぞれ下流ノードを設定する。この様子を、図 3.4 に示す。このようにして、Lulea では検索の段階を減少させる。また、Lulea では展開後の配列をどのようにして圧縮するかについても言及している。

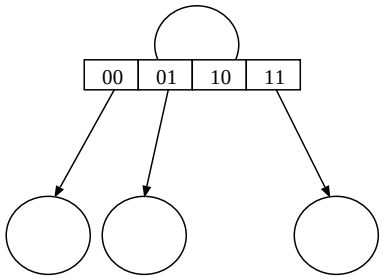

図 3.4: Lulea の中間ノードの構造

3.5 プリフィックス長の二分探索

プリフィックス長の二分探索は、アドレスの長さの二乗根の検索回数しか必要としない方法である。例えば、IPv4 のアドレス長は 32(bit) であるので、$\log_2(32)$、つまり最悪でも 5 回の検索となる。これは、Waldvogel らによって開発された [6]。

プリフィックス長の二分探索では、保持する経路をプリフィックス長毎のハッシュ配列に保持する。また、中間ノードは、その経路にマッチするさらに長い経路があるかどうかを保持する。検索段階では、プリフィックス長のハッシュ配列を検索し、さらに長い経路があればプリフィックス長を長くし、なければプリフィックス長を短くして検索を続ける。

プリフィックス長の二分探索の概念図を図 3.5 に示す。

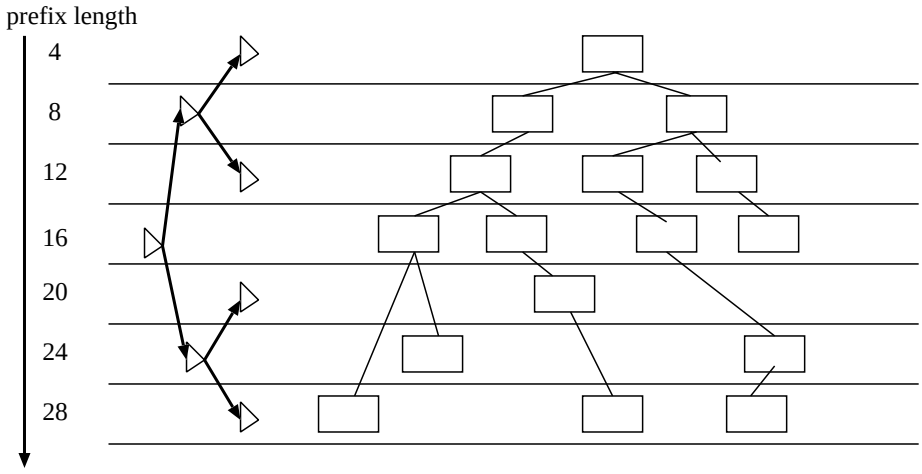

図 3.5: プリフィックス長の二分探索 概念図

第4章　不必要な経路エントリを削除した経路表システムの提案

4.1　概要

本章では不必要な経路エントリを削除した検索用経路表システムについて、その提案と詳細を述べる。

本システムはIPネットワーク上で不連続ネットマスクが事実上使われないものであることを前提とする。この前提のもと、ルータの保有する経路の多くが同一のネクストホップを向いており、集約可能であることに着目した。それらの経路の多くを検索用経路表から省略し、検索すべきエントリ数を削減する手法を詳細に論じる。また、本システムの新規性の要である従来の経路表と検索用経路表の分離についてその有効性を述べる。

また、実際のネットワークで使用されているルータ内の経路表について触れ、本システムを適用したさいの効果予測も述べる。

4.2　本システムの新規性

4.2.1　従来の経路表と検索用経路表の分離

本システムにおける最大の新規性は、従来の経路表と検索用経路表を分離したことにある。従来の経路表を残したまま、検索用に経路エントリ数を削減した経路表を作成することで、パケット転送時の必要処理量を縮小できる。

これまで、UNIXシステムにおける経路検索手法は第3章で触れたように、多く研究されてきた。それらの多くはradix treeに代わる経路表システムや検索アルゴリズムであり、いかに高速に最長一致の経路検索ができるかを追求してきた。

本研究がこれらの先行研究と異なる点は不必要な経路自体を検索対象から外すというところにある。そのため本研究と他の経路検索アルゴリズムは共存可能であり、組み合わせて使用することでさらに経路検索を高速化できる。

4.2.2 従来の経路表との共存

ネットワーク管理者は、BGP や OSPF[8] などの動的経路制御プロトコルの検証を目的として、ルータに経路が正しく設定されているかを調べることがある。そのため検索用経路表とは別に、マスターとなる管理用経路表が必要である。この管理用経路表を検索時に使用する必要はなく、動的経路制御プロトコルなどのユーザプロセスからの経路の追加や削除要求に対応し、それを反映させるためのものである。したがって、この管理用経路表は新たに用意する必要はなく、従来の経路表をそのまま使用すればよい。

また、ユーザに対して検索用経路表を意識させることなく透過的なものとするため、検索用経路表と従来の経路表との間で経路の矛盾があってはならない。矛盾が生じてしまった場合は経路の検証が意味を成さなくなり、ユーザは必要以上のデバッグを強いられることになるため検索用経路表の操作は整合性を第一に考慮したものある必要がある。

4.3 検索用経路表構築アルゴリズム

本システム最大の特徴である検索用経路表は、従来の経路表から同じネクストホップを持つ連続した経路エントリを省いたものである。ここでいう連続した経路エントリとは、ネットマスク長の連続性についてである。例えば表 4.1 で示される経路表を図 4.1 のような一致するビットごとのツリー構造にした場合、連続した経路エントリとは大きな円で表現されるネクストホップを持つノードのうち連続したものをいうこととし、この場合 (A, B), (A, C), (A, D) の 3 つの連続した経路エントリのペアが存在するものとする。

prefix	nexthop
*	a
0	b
10	b
11	a

表 4.1: 経路表の例

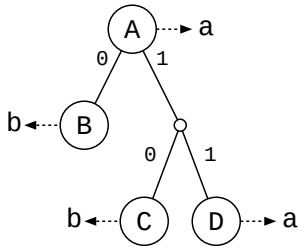

図 4.1: 経路表の例:ツリー構造

なお、本システムによる経路の集約は不連続ネットマスクを用いたプロトコルファミリには適用できない。VLSM を採用した IP 環境において不連続ネットマスクはもはや実用的なものではなく通常使用されないものである。したがって、本システムは不連続ネットマスクを使用しないことが前提である。

4.3.1 経路の追加

検索用経路表は、マスターとなる従来の経路表に変更が加えられるたびに経路表を走査して再構築すればよい。ここで、検索用経路表には同じネクストホップを持つ連続した経路エントリが存在しないという本システムの特徴を考慮すると、経路の追加時だけは処理を大幅に少なくできることがわかる。それは、ツリー構造の経路表の場合、経路の追加は最長一致の経路検索後にされるということに起因している。すなわち、経路を追加する時点での最長一致経路エントリのネクストホップと追加しようとしている経路のネクストホップが一致した場合は追加しなくてもよいということである。ただし、既に存在する経路の上位に経路を追加する場合は、追加する経路と連続となる既存の経路すべてを走査してネクストホップが一致する場合は削除をしなればならない。

この手順で構築された検索用経路表を表 4.2 と図 4.2 に示す。図 4.2 の右側のツリーは patricia tree を用いた場合の、さらにノード数が削減された状態である。

prefix	nexthop
*	a
0	b
10	b

表 4.2: 検索用経路表の例

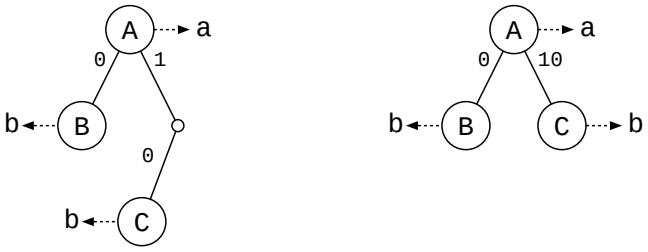

図 4.2: 検索用経路表の例:ツリー構造

4.3.2 経路の削除と変更

経路の追加と異なり、経路の削除と変更については注意が必要である。それは、その対象となる経路が他の下位の経路の集約点となっていた場合、単純に削除や変更をしてしまうと下位の経路にまで影響を及ぼしてしまうことである。このため必ずマスターとなる経路表から下位の経路を走査し、必要であれば検索用経路表に新たに下位の経路を追加しなればならない。さらにそれら下位の経路を加えることで同じネクストホップを持つ連続した経路が生じる場合はそれらを省かなければならない。

このように経路の追加と異なり、必要とされる処理は多い。特に上位の経路であればあるほど下位の経路の集約点となっている場合が多く処理量は増える。

4.4 本システムの利点と欠点

4.4.1 経路表の縮小

経路の集約により、経路表のエントリ数が削減され、規模が縮小する。経路検索全体を通しての実際の効果は経路検索アルゴリズム次第であるものの、一般にエントリ数は少なければ少ないほどより高速に検索できる。

また、本システムは、保有する経路の多くがデフォルトルートのネクストホップと一致する場合にそれらの多くの経路を検索用経路表から省くことが可能なため、検索時には効果を最大限に発揮するといえる。

4.4.2 個別経路毎の統計情報の無効化

BSD の経路表に用意されている統計情報変数のひとつに参照回数を記録するカウンタがある。本システムでは独自に集約された経路表を使用するため、集約された経路ごとの参照回数は記録できるが、本来の個別経路の参照回数を検索時にすべて記録することはできない。従って、これらの統計情報を必要とするシステムでは使用できないか、統計情報を取るための別の手段を用意する必要がある。

4.5 本システムのトポロジごとの効果予測

4.5.1 経路の種類

WIDE プロジェクト (AS2500) のルータは保有経路の違いにより以下の 3 つに大別できる。

- フルルートを保有するルータ
- 国内の経路を保有するルータ
- AS2500 内といくつかの AS 外の経路を保有するルータ

これらのルータそれぞれの場合において例を挙げ、ルータ周辺のトポロジと保有する経路数などの情報とともに本システムを適用した際の効果予測を述べる。

4.5.2 例 1: フルルートを保有するルータの場合

11 万以上の経路数であるフルルートを持つルータとして cisco5.otemachi を例に挙げる。cisco5.otemachi は AS2500 と海外を結ぶルータのひとつであり、周辺の略トポロジを図 4.3 に示す。

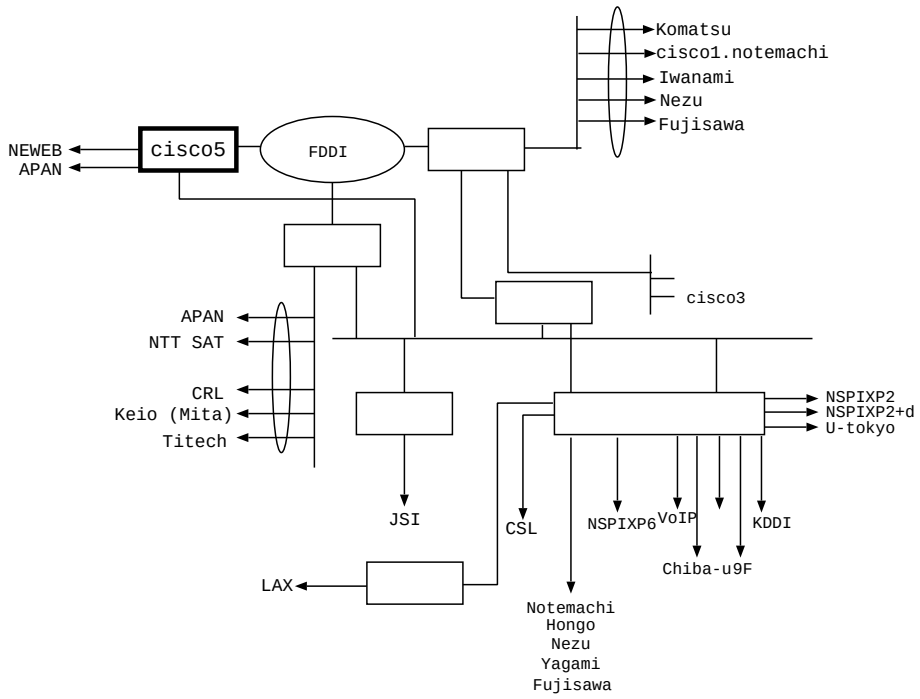

図 4.3: cisco5.otemachi 周辺の略トポロジ

cisco5.otemachi の経路のうち、海外からの多くの経路は左に伸びる矢印の方に向いており、国内の AS 内外の経路の多くは NSPIXP-2 と AS2500 を結ぶ右下のルータに向いている。デフォルトルートは左に伸びる矢印に向いているため、海外からの 9 万近くの経路をすべてデフォルトルートに集約させることが可能であると推測でき、経路数の削減率が期待できる。

cisco5.otemachi の経路表において、BGP 経路、OSPF 経路のほとんどがそれぞれ単一のネクストホップを保持していることを、図 4.4、4.5 に示す。

```
B    208.221.13.0/24 [20/0] via 210.132.94.77, 1w3d
B    206.51.253.0/24 [20/0] via 210.132.94.77, 1w3d
B    205.204.1.0/24 [20/0] via 210.132.94.77, 1w3d
B    204.255.51.0/24 [20/0] via 210.132.94.77, 1w3d
B    204.238.34.0/24 [20/0] via 210.132.94.77, 05:53:58
B    204.17.221.0/24 [20/0] via 210.132.94.77, 1w3d
B    203.238.37.0/24 [20/0] via 210.132.94.77, 1w3d
B    203.34.233.0/24 [20/0] via 210.132.94.77, 1w3d
B    199.0.199.0/24 [20/0] via 210.132.94.77, 1w3d
B    198.17.215.0/24 [20/0] via 210.132.94.77, 1w3d
B    192.68.132.0/24 [20/0] via 203.181.248.207, 2d22h
     170.170.0.0/16 is variably subnetted, 3 subnets, 3 masks
B       170.170.0.0/19 [20/0] via 210.132.94.77, 1d14h
B       170.170.224.0/20 [20/0] via 210.132.94.77, 1d15h
B       170.170.254.0/24 [20/0] via 210.132.94.77, 1d14h
B    216.239.54.0/24 [20/0] via 210.132.94.77, 1w3d
B    216.220.5.0/24 [20/0] via 210.132.94.77, 1w3d
B    216.103.190.0/24 [20/0] via 210.132.94.77, 1w3d
B    213.239.59.0/24 [20/0] via 210.132.94.77, 1w3d
B    213.152.76.0/24 [20/0] via 210.132.94.77, 11:05:48
B    207.254.48.0/24 [20/0] via 210.132.94.77, 2d18h
B    205.152.84.0/24 [20/0] via 210.132.94.77, 1w3d
B    203.220.22.0/24 [20/0] via 210.132.94.77, 5d02h
B    203.171.97.0/24 [20/0] via 210.132.94.77, 1w3d
B    203.1.203.0/24 [20/0] via 210.132.94.77, 4d06h
B    198.205.10.0/24 [20/0] via 210.132.94.77, 13:32:35
B    192.35.226.0/24 [20/0] via 210.132.94.77, 1w3d
     170.171.0.0/16 is variably subnetted, 4 subnets, 2 masks
B       170.171.0.0/16 [20/0] via 210.132.94.77, 1w3d
B       170.171.251.0/24 [20/0] via 210.132.94.77, 1w3d
B       170.171.253.0/24 [20/0] via 210.132.94.77, 1w3d
B       170.171.252.0/24 [20/0] via 210.132.94.77, 1w3d
B    216.155.65.0/24 [20/0] via 210.132.94.77, 2d13h
B    207.138.71.0/24 [20/0] via 210.132.94.77, 1w3d
B    206.32.236.0/24 [20/0] via 210.132.94.77, 1w3d
B    205.236.35.0/24 [20/0] via 210.132.94.77, 1w3d
B    204.138.68.0/24 [20/0] via 210.132.94.77, 1w3d
B    203.87.158.0/24 [20/0] via 210.132.94.77, 01:15:38
B    203.19.218.0/24 [20/0] via 210.132.94.77, 11:24:30
B    202.49.249.0/24 [20/0] via 210.132.94.77, 3d12h
B    199.185.124.0/24 [20/0] via 203.181.248.207, 1d10h
B    198.223.27.0/24 [20/0] via 210.132.94.77, 1d14h
B    193.100.167.0/24 [20/0] via 210.132.94.77, 1w3d
B    192.236.46.0/24 [20/0] via 203.181.248.207, 2d22h
B    192.100.166.0/24 [20/0] via 210.132.94.77, 1w3d
B    192.70.132.0/24 [20/0] via 210.132.94.77, 1w3d
```

図 4.4: cisco5.otemachi の BGP 経路:ネクストホップの多くが 210.132.94.77 である

```
O E1 206.117.185.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.168.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 192.153.106.0/24
           [110/10023] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.138.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 202.249.6.0/24 [110/10121] via 203.178.137.33, 01:00:50, Fddi1/1/0
O E1 206.117.236.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.49.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.19.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.2.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 206.117.87.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 204.80.119.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 204.80.102.0/24
           [110/5071] via 203.178.140.216, 01:00:50, FastEthernet2/0/0
O E1 203.178.143.0/24
           [110/10030] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 206.117.169.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 202.249.37.0/24
           [110/10021] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 192.153.107.0/24
           [110/10023] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 206.117.139.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 206.117.237.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 192.47.167.0/24 [110/10020] via 203.178.137.33, 01:01:00, Fddi1/1/0
O E1 206.117.48.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 206.117.3.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 206.117.18.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
O E1 204.80.103.0/24
           [110/5071] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
     203.178.142.0/24 is variably subnetted, 10 subnets, 5 masks
O       203.178.142.240/29
           [110/81] via 203.178.140.216, 01:01:00, FastEthernet2/0/0
```

図 4.5: cisco5.otemachi の OSPF 経路:ネクストホップの多くが 203.178.140.216 である

4.5.3 例 2: 国内の経路を保有するルータの場合

国内の経路を保有するルータとして gsr1.fujisawa を挙げる。gsr1.fujisawa は国内の AS 内外の 1 万数千の経路を保有するが、それらの多くは隣接する大手町のルータに向いている。この様子を、図 4.6 に示す。

```
B    202.163.96.0/24 [200/12103] via 203.178.136.15, 1d13h
B    199.212.26.0/24 [200/0] via 203.178.136.19, 1w3d
B    199.111.161.0/24 [200/0] via 203.178.136.19, 5d05h
B    192.43.226.0/24 [200/0] via 203.178.136.19, 5d01h
B    207.95.154.0/24 [200/100] via 203.178.136.15, 1w1d
B    192.160.106.0/24 [200/0] via 203.178.136.15, 3w3d
B    209.72.149.0/24 [200/250] via 203.178.136.15, 2d16h
B    194.149.91.0/24 [200/0] via 203.178.136.15, 4d00h
B    192.149.89.0/24 [200/0] via 203.178.136.19, 2d23h
O E1 192.122.183.0/24
          [110/10152] via 203.178.138.227, 00:04:17, GigabitEthernet1/0.4
B    192.104.166.0/24 [200/0] via 203.178.136.19, 6d18h
B    209.118.183.0/24 [200/8515] via 203.178.136.15, 1d23h
B    202.84.142.0/24 [200/0] via 203.178.136.15, 2d17h
B    194.35.241.0/24 [200/0] via 203.178.136.19, 4d20h
     193.1.208.0/26 is subnetted, 1 subnets
B       193.1.208.192 [200/0] via 203.178.136.19, 1w3d
B    192.16.192.0/24 [200/0] via 203.178.136.19, 1w2d
B    136.152.0.0/16 [200/0] via 203.178.136.19, 6d18h
B    211.253.61.0/24 [200/100] via 203.178.136.15, 2w6d
B    208.185.122.0/24 [200/5477] via 203.178.136.15, 1w0d
O E1 206.117.168.0/24
          [110/5061] via 203.178.138.227, 01:02:01, GigabitEthernet1/0.4
B    192.236.63.0/24 [200/0] via 203.178.136.19, 1w0d
B    192.206.29.0/24 [200/0] via 203.178.136.19, 3d18h
B    136.155.0.0/16 [200/0] via 203.178.136.19, 1w0d
     198.169.125.0/28 is subnetted, 1 subnets
B       198.169.125.128 [200/0] via 203.178.136.19, 1w3d
B    136.154.0.0/16 [200/5399] via 203.178.136.15, 11:28:11
B    170.190.0.0/16 [200/5403] via 203.178.136.15, 5d17h
B    136.156.0.0/16 [200/0] via 203.178.136.19, 4d20h
B    210.96.165.0/24 [200/5353] via 203.178.136.15, 1w0d
B    198.172.125.0/24 [200/5399] via 203.178.136.15, 1w0d
B    136.159.0.0/16 [200/0] via 203.178.136.19, 1w3d
B    202.7.219.0/24 [200/5469] via 203.178.136.15, 1w0d
O E1 192.188.106.0/24
          [110/10152] via 203.178.138.227, 01:02:26, GigabitEthernet1/0.4
     136.145.0.0/16 is variably subnetted, 88 subnets, 5 masks
B       136.145.31.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.30.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.57.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.56.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.59.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.58.0/24 [200/0] via 203.178.136.19, 2d05h
B       136.145.61.0/24 [200/0] via 203.178.136.19, 2d05h
```

図 4.6: gsr1.fujisawa の経路表:ネクストホップの多くが 203.178.136.15 または 203.178.136.19 である

4.5.4 例 3: Internal route を持つ場合

BGP を喋らず、AS 内に広告される経路を OSPF によって計算するルータの例として、慶應義塾大学を収容する cisco11.fujisawa を挙げる。AS 内のルータは大半がこのようなルータであることが一般的に言える。

AS 内に広告される経路のほとんどが、ある単一のネクストホップを向いている様子を、図 4.7 に示す。

```
O E1 192.138.29.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.134.29.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.171.12.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.5.166.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.16.166.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 206.117.49.0/24 [110/5059] via 203.178.138.227, 01:03:10, Vlan4
O E1 198.253.177.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 202.209.159.0/24 [110/10010] via 203.178.138.231, 01:03:10, Vlan4
O E1 202.223.142.0/24 [110/10010] via 203.178.138.231, 01:03:10, Vlan4
O E1 192.55.106.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 206.117.19.0/24 [110/5059] via 203.178.138.227, 01:03:10, Vlan4
O E1 202.25.113.0/24 [110/10010] via 203.178.138.231, 01:03:10, Vlan4
O E1 206.117.2.0/24 [110/5059] via 203.178.138.227, 01:03:10, Vlan4
O E1 204.222.221.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.47.46.0/24 [110/10010] via 203.178.138.231, 01:03:10, Vlan4
O E1 204.134.136.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 192.12.29.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 205.68.103.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 198.35.10.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 206.117.87.0/24 [110/5059] via 203.178.138.227, 01:03:10, Vlan4
O E1 204.80.119.0/24 [110/5059] via 203.178.138.227, 01:03:10, Vlan4
O E1 198.180.147.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 198.253.198.0/24 [110/10150] via 203.178.138.227, 01:03:10, Vlan4
O E1 204.80.102.0/24 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 204.37.17.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 203.178.143.0/24 [110/10008] via 203.178.137.93, 01:03:26, Vlan100
O E1 206.117.169.0/24 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 202.249.37.0/24 [110/10009] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.48.242.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 137.128.0.0/16 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.153.107.0/24 [110/10010] via 203.178.138.231, 01:03:26, Vlan4
O E1 206.117.139.0/24 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.41.208.0/24 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 202.242.7.0/24 [110/10010] via 203.178.138.231, 01:03:26, Vlan4
O E1 192.135.122.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 202.44.203.0/24 [110/10010] via 203.178.138.231, 01:03:26, Vlan4
O E1 206.117.237.0/24 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.5.148.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.12.133.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.47.167.0/24 [110/10010] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.16.167.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 137.247.0.0/16 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 198.17.176.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 137.228.0.0/16 [110/5059] via 203.178.138.227, 01:03:26, Vlan4
O E1 198.253.161.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
O E1 192.42.122.0/24 [110/10150] via 203.178.138.227, 01:03:26, Vlan4
```

図 4.7: cisco11.fujisawa の経路表:ネクストホップの多くが 203.178.138.227 である

第5章　本経路表システムの設計と実装

5.1　設計

5.1.1　全体図

4.3BSD Reno 以降の BSD では、ルーティングメッセージを用いてユーザプロセスとカーネルとの間で経路情報をやりとりする。カーネル側のインターフェイスをルーティングソケットと呼び、これを介してメッセージを交換する。ルーティングメッセージにはバージョンという概念がありユーザプロセスはカーネルと同じバージョンのルーティングメッセージを使用しなければならない。ルーティングメッセージについてはヘッダファイル<net/route.h>で定義されており、現時点での NetBSD におけるルーティングメッセージはバージョン 3 である。これらを図 5.1 に示す。

```
#define RTM_VERSION     3       /* Up the ante and ignore older versions */

#define RTM_ADD         0x1     /* Add Route */
#define RTM_DELETE      0x2     /* Delete Route */
#define RTM_CHANGE      0x3     /* Change Metrics or flags */
#define RTM_GET         0x4     /* Report Metrics */
#define RTM_LOSING      0x5     /* Kernel Suspects Partitioning */
#define RTM_REDIRECT    0x6     /* Told to use different route */
#define RTM_MISS        0x7     /* Lookup failed on this address */
#define RTM_LOCK        0x8     /* fix specified metrics */
#define RTM_OLDADD      0x9     /* caused by SIOCADDRT */
#define RTM_OLDDEL      0xa     /* caused by SIOCDELRT */
#define RTM_RESOLVE     0xb     /* req to resolve dst to LL addr */
#define RTM_NEWADDR     0xc     /* address being added to iface */
#define RTM_DELADDR     0xd     /* address being removed from iface */
#define RTM_OIFINFO     0xe     /* Old (pre-1.5) RTM_IFINFO message */
#define RTM_IFINFO      0xf     /* iface/link going up/down etc. */
#define RTM_IFANNOUNCE  0x10    /* iface arrival/departure */
```

図 5.1: ルーティングメッセージの種類

経路制御デーモンなどのユーザプロセスがカーネルに経路を追加したい場合は、RTM_ADD メッセージを、カーネル内の経路を削除したい場合は RTM_DELETE を使用する。

一方、カーネル内における経路情報交換インターフェイスは関数 rtrequest() であり、ルーティングソケットがルーティングメッセージを受け取ると呼ばれる関数が rtrequest() である。関数 rtalloc() は経路を検索し、その結果であるネクストホップとなる値を返す。

これらの関数を変更し、検索用経路表からの検索とその構築をユーザに対して透過的にできるようにする。

5.1.2 経路表の検索

本実装では従来の経路表と検索用経路表の構造を同じとし、関数 rtalloc() 及び関数 rtalloc1() による検索ルーチンを再利用する。このため経路数の違いを純粋に比較できる。また、検索時は検索対象の経路表を切り替えるだけで済むため実装も容易である。

カーネル内部で経路表は図 5.2 に示されるようにプロトコルファミリ毎の

radix_node_head 構造体へのポインタの配列という形で保持されている。したがって、検索用経路表のためにこの変数を二重化し、必要に応じて切り替えて使用することで対応する。

```
struct  radix_node_head *rt_tables[AF_MAX+1];
```

図 5.2: 配列 rt_tables[]

5.1.3 経路表の構築と管理

経路の追加や削除といった処理に必要なことは第 4 章で述べたとおりである。これらを注意し、関数 rtrequest1() 内の配列 rt_tables[] とその値を参照した処理をする部分を検索用経路表を操作できるよう変更する。

5.2 実装環境

表 5.1 に本システムの実装環境を示す。

OS (Kernel)	NetBSD-current 1.6A
Architecture	i386
CPU	Athlon 1000MHz
L1 Cache	128KBytes
L2 Cache	256KBytes
Memory	768MBytes

表 5.1: 実装環境

第6章　評価

6.1　経路エントリ数削減による影響

ここでは、検索用経路表用意することがどのくらい効果あったのかを定量的に評価する。

表6.1に本研究による経路表の縮小具合を示す。経路表A, B, Cはそれぞれ第4章の効果予測の節で説明したルータcisco5.otemachi, gsr1.fujisawa, cisco11.fujisawaの経路表にそれぞれ該当する。

経路表	経路数	適用後経路数	縮小率
A	110120	11309	10.3%
B	16612	739	4.45%
C	2593	700	25.9%

表 6.1: 本システム適用前後の経路数の比較

6.2　Pentium TSC を用いた測定

経路検索にかかる時間的コストを調査するためにPentium TSC(Time Stamp Counter)を用いた。Pentium TSCは64bitのカウンタであり、CPUのサイクルクロック毎に値が1ずつ増加する。この値を参照するためにカーネル内の`ip_input()`, `ip_forward()`, `ip_output()`, `rtalloc()`関数などにカウンタ値を記録するコードを追加した。

関数`ip_input()`はデータリンク層からIPパケットを受けとり、自ホスト宛のパケットかどうかを調べる。自ホスト宛でなければ関数`ip_forward()`を呼ぶ。その中で経路を調べた上で関数`ip_output()`にパケットを渡し、関数`ip_output()`がデータリンク層インターフェイスにパケットを渡すという流れである。

経路表A,B,Cと必要クロックサイクル数のそれぞれの関係を表6.2, 6.3, 6.4に

示す。項目 rtalloc() は関数 `ip_input()` 中で関数 `rtalloc()` が呼ばれてから値を返すまでのクロック数であり、項目 ip_input() は関数 `ip_input()` が呼ばれてから関数 `ip_output()` 中でデータリンク層インターフェイスにパケットを渡すまでのクロックサイクル数である。これらのクロックサイクル数は他のホストからランダムな IP アドレスを宛先とする数千の IP パケットを投げ、経路を検索した結果の平均値である。なお、測定中は他のプロセスも動作しており、CPU キャッシュやページングの影響を受けている。このためルータとしての機能を果たす通常時に比較的近い環境といえる。

	経路数	rtalloc()	ip_input()
適用前	110120	15694	34091
適用後	11309	6562	19989

表 6.2: 平均クロックサイクル数の変化:経路表 A

	経路数	rtalloc()	ip_input()
適用前	16612	11910	27724
適用後	739	5216	20298

表 6.3: 平均クロックサイクル数の変化:経路表 B

	経路数	rtalloc()	ip_input()
適用前	2593	8752	25521
適用後	700	4366	19379

表 6.4: 平均クロックサイクル数の変化:経路表 C

OS (Kernel)	NetBSD-current 1.6A
Architecture	i386
CPU	Dual AMD Athlon Palomino MP 1.6GHz
L1 Cache	128KBytes
L2 Cache	256KBytes
Memory	1024MBytes

表 6.5: 測定環境

測定環境を表 6.2 に示す。CPU 動作クロックは 1.6GHz であるため 1600 クロックサイクルで約 1us に相当する。これらの結果を時間に換算するとそれぞれ約 6us〜約 8us だけ高速に処理できるようになったといえる。これは数マイクロ秒のオーダーであり、ping コマンドなどからはその結果を見ることは難しいが、IP 層で必要な処理が従来に比べ 59%〜76%の時間で済ませられており、本研究の有効性が示される。

6.3 新規性について

本研究は独自の手法をもちいて経路を集約し、それをもとにした経路表を検索時に使用する。これにより、意味合いは多少異なる集約ではあるものの経路制御プロトコルなどの運用技術だけでは不可能であった経路の集約をも可能とした。

また、カーネル内に検索用と管理用の 2 種類の経路表を保持することも独自の手法である。経路表を二重化するオーバーヘッドはメモリ使用量の増加という面と経路表構築にかかる時間としてあらわれるものの、検索時間の短縮度合いによる効果は大きい。

第7章　結論

本章では、本研究のまとめと今後の課題を述べる。

7.1　まとめ

クラスという概念が事実上消失したインターネットアーキテクチャにおいて、不連続ネットマスクを用いることはもはや現実的ではなく、実用的なものではない。本研究ではここに着目し、経路制御プロトコルの使用する経路情報をもとに、ルータとなるシステムがパケット転送時に使用できる経路エントリ数を削減した、検索用経路表を構築することに成功した。

本システムで用いた検索用経路表の構築方法は、IP アドレスをビット列とみなしてツリー状の経路表構造を持たせたときに、経路表内にネクストホップを同じとする連続した経路エントリを省略するというシンプルなものである。また、経路を追加する際には、検索用経路表の構築工程を大幅に削減できることについても述べるとともに、経路の削除や変更の際の処理の複雑性についても議論した。

評価には WIDE プロジェクト (AS2500) 内のルータのうち、フルルート、国内経路、AS 内経路の三つの異なる経路セットを保有する、3 種類のルータから実際に抽出した経路を使用した。この実際に使用されている経路のそれぞれについて、本システムによる検索用経路表を作成した結果、経路数をもとの経路数に比べ 4.45%から 25.9%の大きさにまで削減することができた。検索対象の経路数の削減による、経路検索に必要とされる時間の減少を、CPU のクロックサイクル数をもとにした測定で述べるとともに、測定環境において、IP 層で必要とされる処理がそれぞれの場合において 59%から 76%の処理量となったことについて言明した。

7.2 今後の課題

本研究に関する今後の課題として 2 つ挙げる。ひとつは、radix tree 以外の他の経路検索手法を適用し、さらなる経路検索の高速化を目指すことである。もうひとつは、より積極的な経路の集約である。

本システムは検索時の経路数を削減するものであり、経路表構造を直接変更するものではない。そのため他の経路検索手法との親和性が高い。さらなる経路検索を目指して有効な手法を取り入れていくことは楽しみである。

また、2 つめの積極的な経路の集約は、例えば検索用経路表内でネットマスク長が 24 の連続した経路を複数集約し、より短かいネットマスク長の経路とすることである。BGP などの経路制御プロトコルからの AS パス属性をもとにした動的な集約も考えられる。

これらの手法については今後の研究課題とし、実現していく予定である。

謝辞

本研究を進めるにあたり御指導をいただきました慶應義塾大学環境情報学部教授の村井純博士に深い感謝の意を表します。また、絶えず御助言を頂きました同学部の中村修博士、楠本博之博士、東京大学大学院情報理工学系研究科助教授 江崎浩博士に感謝いたします。

また、本論文の執筆に当たりまして常に励ましと御協力を下さった慶應義塾大学環境情報学部徳田・村井・楠本・中村研究会の諸兄に感謝致します。

参考文献

[1] P. Almquist and F. Kastenholz. Towards Requirements for IP Routers. RFC 1716, IETF, November 1994.

[2] APNIC. Apnic homepage. WWW page. http://www.apnic.net.

[3] S. Fuller, T. Li, J. Yu, and K. Varadhan. Classless inter-domain routing (CIDR): an address assignment and aggregation strategy. RFC 1519, IETF, September 1993.

[4] Donald E. Knuth. *The Art of Computer Programming, vol 3.* 1973.

[5] S. Carlsson M. Degermark, A. Brodnik and S. Pink. Small forwarding tables for fast routing lookups. In *Proceedings of SIGCOMM '97*, 1997.

[6] J. Turner M. Waldvogel, G. Varghese and B. Plattner. Scalable high speed ip routing lookups. In *Proceedings of SIGCOMM '97*, September 1997.

[7] D. R. Morrison. Patricia – practical algorithm to retrieve information coded in alphanumeric. In *Journal of the ACM, 15(4)*, October 1968.

[8] J. Moy. Ospf version 2. RFC 2328, IETF, April 1998.

[9] WIDE Project. Nspixp homepage. WWW page. http://nspixp.sfc.wide.ad.jp/.

[10] WIDE Project. Wide project homepage. WWW page. http://www.wide.ad.jp/.

[11] T. Li Y. Rekhter. A border gateway protocol4. RFC 1771, IETF, March 1995.

[12] 財団法人インターネット協会. インターネット白書 2001, June 2001.