修士論文 2001 年度 (平成13年度)

ドメイン間経路制御における動的な障害検出機構についての研究

慶應義塾大学大学院政策・メディア研究科

長橋 賢吾

修士論文要旨 2001年度(平成13年度)

ドメイン間経路制御における動的な障害検出機構についての研究

論文要旨

現在、インターネットにおける経路情報の交換には、大きくわけてドメイン内経路制御およびドメイン間経路制御の二種類が利用されている。その中で、ドメイン間経路制御は、多くの問題が指摘されているが、深刻な問題の一つとしてブラックホール問題がある。

ブラックホール問題は、本来、経路を広告すべきAS(自律システム)が経路を広告せず、他のASが誤って経路を広告してしまい、結果的にその経路の到達性が失われる問題を指し、現在、インターネット上で頻繁に発生している問題である。

本論文では、ABIC(Automatically BGP Integrity Check)という手法を用いて、BGPによる経路の更新が発生する度に、データベースマネージャーに問い合わせ、更新メッセージに含まれるAS情報と、データベース中に存在するAS情報を比較することによって、正しいASによる経路広告であるかをチェックする。

これによって経路のブラックホールを動的に検出することが可能になり、このシステムを設計・実装し、実際のネットワーク上で運用・評価した結果、運用に耐えられる十分なパフォーマンスを得ることができた。

キーワード

1. ドメイン間経路制御　2. ブラックホール　3. 自律システム　4. プレフィックス　5.BGP

慶應義塾大学大学院政策・メディア研究科
長橋 賢吾

Abstract of Master's Thesis Adademic Year 2001

Research into the Automatically Detection of BlackHole Routes in the Interdomain Routing Environtment

Summary

The exchanges of routing information being used in the Internet are mainly the Interiordomain routing and the Interdomain routing. Of the two, there are many problems being pointed out in the Interdomain routing, with the "BlackHole Route Problem" being the most critical. This "BlackHole Route Problem" can often be seen in the current Internet, caused when the appropriate AS (Autonomous System) does not announce its route while other AS accidentaly announces its own route, causing the route to lose reachability.

In this paper, we propose ABIC (Automatically BGP Integrity Check) architecture, which querys to the database manager everytime the route is updated in BGP, to see if the AS information in the update packet from the BGP match the AS information in the database or not. By applying ABIC architecture, it can dynamically detect the BlackHole routes. After designing and implementing ABIC architecture and applying it to the Internet, our evaluation showed enough performance to operation in the Internet.

Key Word

1.Interdomain Routing 2. BlackHole 3. AS 4. Prefix 5. BGP

Keio University Graduate School of Media and Governance
Kengo Nagahashi

目次

図 目 次

表 目 次

第1章　はじめに

本章では、まず、1.1 節で、本論文の目的について述べる。その後、1.2 節で、本論文に関する構成について述べ、1.3 節で、本論文中で使用される語句について定義する。

1.1　本論文の目的

本論文では、インターネットにおける経路制御のうちドメイン間経路制御に絞り、その中で深刻な問題の一つであるブラックホール問題に焦点をあて、その解決を図ることを目的とする。この目的を達成することによって、これまでは実現されていなかった動的にブラックホール経路を検出することが可能になる。

1.2　本論文中の構成

2 章では、本論文における背景について述べる。背景としては、現在のインターネット上の経路制御機構について 2.1 節で述べ、その後、ドメイン内経路制御、およびドメイン間経路制御について議論する。その後、ドメイン間経路制御に対象を絞り、現在どのような点が問題になっているかについて、2.2 節において議論する。その中で、とくに問題となっている経路のブラックホールについて 2.2.3 小節で取り上げ、その発生原理、現状の対策などについて議論する。

3 章では、これまでのドメイン間経路制御に関する関連研究について、2.2 節で指摘した問題点を踏まえて述べる。具体的には、3.1 節では、“BGP Route Flap Damping” と呼ばれる変動している経路に対してペナルティを与え、一定の閾値を越えた場合、経路を抑制する手法について、解説、議論する。3.2 節では、BGP で定義されている “Community” 属性について解説、議論する。3.3 節では、S-BGP(Secure BGP) と呼ばれる BGP プロトコルにセキュリティ拡張を施したプロトコルについて解説・議論する。

4 章では、本論文で提案するアーキテクチャー ABIC(Automatically BGP Integrity Check) について述べる。まず、4.1 節で、現状のドメイン間経路制御がどのような規模で運用されている

かなどについて検証し、その要件を整理する。その後、4.3 節では、上記のアーキテクチャーについて他との比較をしながらその有効性について議論する。5 章では、4 章で提案したアーキテクチャーに関する設計および実装について議論する。

5.1 節では、データベースマネージャーおよび BGP ルータ間でのメッセージ交換、データベースについて議論する。

5.2 節では、実際の実装について具体的に議論する。6 章では、5 章で、設計・実装した本手法についての評価をおこなう。7 章では、提案したアーキテクチャーについての結論について述べ、残された研究課題など今後のロードマップについて議論する。

1.3 本論文中の語句ついて

本論文中で用いる用語を表 1.1 に定義する。

表 1.1: 用語の定義

用語	本論文中での意味
経路制御	インターネット上においてルータ同士で到達性情報 (経路) を交換する仕組み。ルーティングと同義。
プレフィックス	133.27/16 のように、IP アドレス中のネットワーク部分とそのネットワーク部のビット長 (プレフィックス長) を組み合わせたもの。本論文では、経路と同義として扱う。
自律システム (AS)	一つの管理ポリシーによって管理されているネットワークの集合体。
AS 番号	IR(Internet Registry) から割り当てられる一意な自律システムの番号。
ドメイン	経路制御ドメイン、本論文では AS と同義として、定義する。
ドメイン内経路制御	AS 内での経路制御。IGP (Interior Gateway Protocol) と同義。
ドメイン間経路制御	AS 間での経路制御。EGP (Exterior Gateway Protocol) と同義。また、本論文中では BGP4+とも同義の語句として用いる。
BGP4+	ドメイン間経路制御で一般に用いられている経路制御プロトコル。
ポリシーデータベース	ある AS に関する管理ポリシーを記述してあるデータベース。
BGP ルータ	ドメイン間経路制御プロトコルである BGP4+を使って経路制御するルータ。
UPDATE	BGP-4 UPDATE Message の略。
ピア	BGP-4 での接続相手先、近隣 (neighbor) と同義
広告	BGP ルータが他のピアに UPDATE を送出すること、アナウンスと同義
オリジン AS	あるプレフィックスを生成 (オリジネート) する AS

第2章　背景

2.1　インターネットにおける経路制御

インターネットにおける経路制御は、以下の二つに大きく分別することができる。

- ドメイン内経路制御
- ドメイン間経路制御

ここでのドメインとは、自律システム (AS) を指し、AS 内における経路経路が、前者のドメイン内経路制御 AS 間での経路制御が後者のドメイン間経路制御にあたる。図 2.1 は、この関係について図示している。

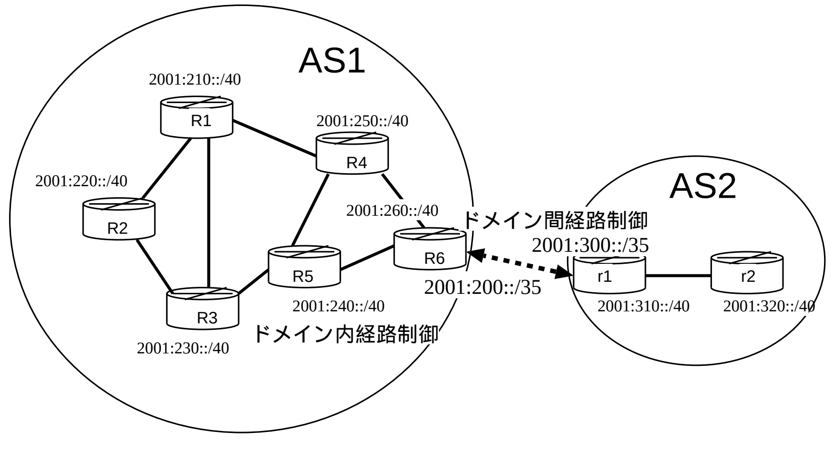


図 2.1: ドメイン内経路制御とドメイン間経路制御

ドメイン内

AS1 は、R1 から R6 まで 6 台のルータで構成され、その中で 2001:210::/40 などの経路情報を独自のルーティングポリシーによって、どのルータが経路情報を交換している。
AS2 では、r1 と r2 が 2001:310::/40、2001:320::/40 という経路を AS1 のポリシーとは関係なく、AS2 の独自のポリシーで交換している。
このようにドメイン内部での独自の経路ポリシーに基づいて経路を交換する仕組みがドメイン内経路制御である。

ドメイン間

一方、AS1 での R6 と AS2 での r1 は、それぞれのドメインで集約した経路をお互いに交換している。このようにドメイン内での経路を集約して、他の AS と経路を交換するのがドメイン間経路制御である。
図 2.1 の例では、AS1、AS2 ともに一つのルータがドメイン間経路制御をおこなっているが、実際には、冗長性を持たせるため、複数のルータでドメイン間経路制御している場合の方が圧倒的に多い。
本章では、2.1.1 節で、ドメイン内経路制御、そして 2.1.2 節においてドメイン間経路制御の仕組みについて述べる。

2.1.1 ドメイン内経路制御

ドメイン内経路制御とは、前述のように一つの AS 内でルータがお互いの経路情報を交換する仕組みである。本節では、このドメイン内経路制御について述べる。ドメイン内経路制御には、ルータをホップした数を計算し、もっともホップのすくない経路を採用するアルゴリズムをもつ RIP(Routing Information Protocol) [1] が存在するが、これは

- 最大ホップ数が、15 ホップまでで、それよりホップ数の多いネットワークでは利用することができない
- 原則として 30 秒に一度しか経路のリフレッシュをしないので、経路情報の収束が遅い

などの理由により、非常に小規模なネットワークを除いて、現在、その運用はおこなわれていない。それにかわって、OSPF(Open Shortest Path First)[2] や IS-IS[3] に代表されるリンクス

テート方式によるドメイン内経路制御をおこなう方式が一般的である。本小節では、リンクステート方式によるドメイン内経路制御に焦点をあて、その特徴そして、どのようにして経路を交換するかについて述べる。
リンクステート方式によるドメイン内経路制御は、おもに以下の要素から構成される。

- ドメイン内のルーティングエリアのトポロジーに関して Link-State-DataBase(LSDB) をもち、それを生成、維持する
- 経路の交換に関しては、LSDB に自身のトポロジー情報を LSA(Link State Advertisement) として保持する
- ルータは、その LSA をエリア上のすべてのルータに対して、フラッディングする
- 各ルータは、その LSA を受信し、各ルータが保持しているトポロジーに反映させる

つまり、ドメイン内経路制御に参加しているすべてのルータは、おなじトポロジー情報を記述したデータベースをもち、もし、あるルータのトポロジーが変化した場合は、それをフラッディング (ブロードキャスト) し、そのエリアに参加しているすべてのルータに、変更情報を通知し、そのデータベースを同期させる。
この場合、あるルータが誤って自身のもつべき経路とはまったく関係のない経路を広告しても、それがLSA として送信され、他のルータにも容易に注入することが可能である。OSPF では、こうした攻撃を防ぐためにパスワード認証をしているが、これだけでは不当な LSA の注入を避けることはできない。

2.1.2　ドメイン間経路制御

次にドメイン間経路制御に関して、以下の 3 項目について述べる。

- ドメイン間経路制御の概要
- ドメイン間経路制御の経緯
- BGP4+プロトコルについて

ドメイン間経路制御の概要

ドメイン間経路制御は、2.1.1 節で述べたエリア内のすべてのトポロジー情報を共有しているリンクステート方式とは異なり、LSDB などは保持せず、更新情報 (BGP UPDATE メッセージ)

をドメイン間のルータで交換する。
また、現在のドメイン間経路制御の事実上の標準であるBGPでは、パス・ベクター方式を採用している。これは、UPDATEメッセージ中に、AS-PATHと呼ばれるパスが含まれており、これは

- 経路を生成したAS(オリジンAS)から、ルータを経由する毎に、そのAS番号をAS-PATHに追加する
- もし、AS-PATHに、同じAS番号が存在していたら、BGPはそれをループとみなし、そのUPDATEを無効にする

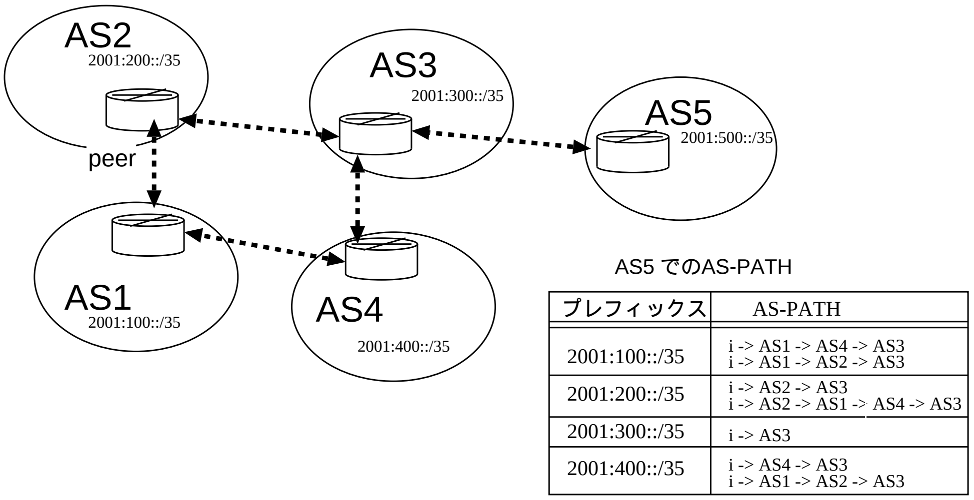
という機能を提供する。これによって、ループを検出することが可能となる。

ドメイン間経路制御の経緯

ドメイン間経路制御は、インターネットの初期段階では、EGP(Exterier Gateway Protocol)と呼ばれるゲートウェイプロトコルを使用していた。実際、アメリカ合衆国において初期のインターネット網であるNFSNETでは、はじめEGPが利用されていた。しかし、EGPは、柔軟な経路制御ポリシーが記述できない、ルーティングループが発生するなどがあり、これにかわるプロトコルとして、BGP(Border Gateway Protocol)が登場した。。当初、交換する経路は、クラスAやクラスBなどクラスに基づいたアドレスによる経路交換であったが、クラスBアドレスブロックの枯渇などにより、1993年頃からクラスという概念を排除したCIDR(Classless Inter-Domain Routing)[4]が導入されることになった。CIDRに対応したBGPが、BGP4+[5]であり、現在ではBGP4+をベースにIPv6やIPXなどのプレフィックスもサポートしたMultiprotocol Extensions[6]が、幅広く用いられている。なお、本論文では、今後BGPは、上記のBGP4+Multiprotocol Extensionsを指す。

BGP4+プロトコルについて

次に、BGPがどのようなプロトコルであるか、実際に例をあげて説明する。図2.2は、典型的なBGP経路交換を示している。
図2.2では、一つのASにつき、通常，一つのAS内部に1台のルータのみ存在してBGP経路交換する場合は少ない。しかし、BGPの概念を説明しやすくするため、図2.2では、一つのASに一台のみBGPルータが存在すると仮定する。



AS5 でのAS-PATH

プレフィックス	AS-PATH
2001:100::/35	i -> AS1 -> AS4 -> AS3 i -> AS1 -> AS2 -> AS3
2001:200::/35	i -> AS2 -> AS3 i -> AS2 -> AS1 -> AS4 -> AS3
2001:300::/35	i -> AS3
2001:400::/35	i -> AS4 -> AS3 i -> AS1 -> AS2 -> AS3

図 2.2: ドメイン間経路制御

まず、AS1 と AS2 との経路交換について考える。AS1 と AS2 はお互いに経路を交換するために、TCP ポート 179 番で、近隣ルータ (これをピアと呼ぶ) と TCP コネクションを確立する。また、BGP は、TCP に似た状態遷移 (Finate State Machine) を持つ。つまり、コネクションの状態を幾つかの段階に分け、その状態を管理する。具体的に、以下の 5 段階がある。

- Idle

  ピアからのコネクションが来るまで待機。すべてはこの状態から始まる。

- Connect

  ピアからコネクションがきて、TCP コネクションが確立を待機している状態

- Active

  TCP コネクションが確立して、ピアからの応答を待っている状態。

- OpenSent

  Open メッセージを送信し、ピアからの Open メッセージを待っている状態。

- OpenConfirm

  ピアからの keepalive/notify メッセージを待つ状態。これらのメッセージを受信すると、Established になる。

- Established

  ピアとのセッションが確立した状態。UPDATE メッセージの交換を始める。

状態遷移が、Established になったとき、BGP は、UPDATE を交換することによって、経路の追加、取り消し (WITHDRAW) を行っている。UPDATE は、図 2.3 のように以下の三つの要素から構成される。

取り消し (WITHDRAW) このフィールドには、もはや到達性がなくなった IPv4 プレフィックスのリストがはいる

到達性情報 (NLRI) 到達性のある IPv4 プレフィックスのリスト。IPv6 の場合は、パス属性中の”Multi-Protocol Reachable NLRI” に NLRI がはいる

パス属性 経路交換に必要な各種属性 (AS パス、ネクストホップ、ローカルプリファレンス、コミュニティ属性、MED 等) が入っており、同一の同一の NLRI が複数のピアから注入されたときに、どの経路を優先するか判断するため上記の属性が利用される

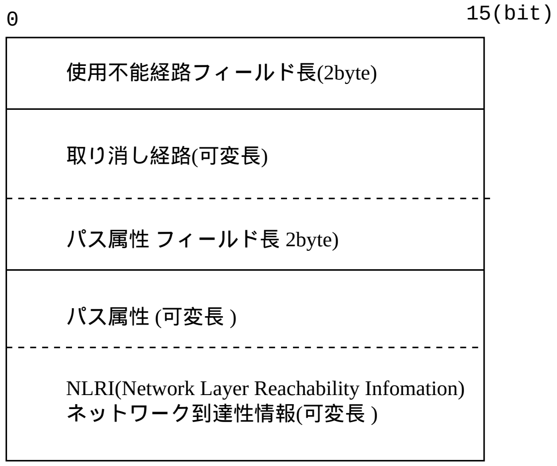


図 2.3: BGP UPDATE メッセージ

AS1 と AS2 の場合では、AS1 と AS2 がピアの確立に成功すると、AS1 は、NLRI として自身で生成している経路 2001:100::/35 を AS2 に BGP UPDATE として送出する。これと同様に AS2 も NLRI として 2001:200::/35 を AS1 に対して送出する。AS1 は、2001:200::/35 というプレフィックスは、AS2 以外からは広告されていないので、これを最良の経路と判断して、自身の経路表に注入する。AS2 も、2001:100::/35 に関して同様に振る舞う。

次に、図 2.2 において、AS5 に焦点をあてる。AS5 は、AS3 のみとピアを確立しており、すべてのプレフィックスは、AS3(AS3 をトランジット AS と呼ぶ) を経由することになる。このとき、AS1 が生成する経路 2001:100::/35 は、

1. AS1 → AS4 → AS3
2. AS1 → AS2 → AS3

の二通りのAS-PATHが存在する。この場合、AS5が、どちらのAS-PATHを自身の経路表に注入するかは、UPDATEメッセージ中のパス属性中の情報に依存する。たとえば、パス属性中のMED(Multi Exit Disc)が、AS4からのUPDATEには、50で、AS3からのUPDATEでは、100の場合、MEDの値が低いAS4のUPDATEが最良の経路としてAS5の経路表に注入される。このように、BGPは、ドメイン間でどのASあるいはプレフィックスを優先するか、あるいは拒否するかというポリシールーティングに適した経路制御機構であるといえる。
以上の議論をまとめると、ドメイン間経路制御およびBGPは、以下の特徴をもつ。

- BGPは、パス・ベクトル型アルゴリズムを用いており、ループを発生させないように設計されている
- インターネットの黎明期には、ドメイン間経路制御プロトコルには、EGPが用いられていたが、現在はCIDRに対応したBGP4+Multiporotocl Extensionsが標準である。
- BGPルータ同士は、ピアとよばれるTCPセッションを確立し、UPDATEメッセージというメッセージを交換することによって経路交換をおこなう。
- 異なるASパスで、おなじ経路情報を受信した場合、UPDATE中のパス属性が参照され、その中で最適な経路が選択される。

これらを踏まえて、次節では、現在のドメイン間経路制御に関する問題点について議論する。

2.2 ドメイン間経路制御の問題点

現在のドメイン間経路制御の問題点として、以下の3点をあげることができる。

- 経路情報の収束
- AS番号の不足
- ブラックホール経路の発生

各問題について、次節以降で議論する。

2.2.1 経路情報の収束

経路情報の収束とは、ある AS が広告した経路情報が、インターネット上で振動せず、安定して伝播することを指す。小規模なネットワーク上であれば、考えられない問題であるが、ドメイン間経路制御というインターネット全体の経路制御でもあるドメイン間経路制御では、非常に深刻な問題である。

ドメイン間経路制御は、しばしば “デフォルトフリーゾーン (default free zone)” と呼ばれ、ドメイン内経路制御で頻繁に用いるデフォルト経路が原則的に存在しない。つまり、BGP ルータは、インターネット上のすべての経路を持つ必要がある。現在、この経路 (フル・ルート (full route) と呼ぶ) の数は、IPv4 では,104,000 経路を越えており、今後もさらに増えることが予想される。つまり、104,000 もの経路情報は、膨大な量のルータのメモリを消費するばかりか、ひとたびある AS の BGP ルータで、経路の取り消し・更新がおこなわれると、それがあらゆるルータに伝播することになる。IX(Internet Exchange) 上に接続される BGP ルータの UPDATE メッセージの数は、15 分ごとに数千回との指摘 [7] もあり、ドメイン間経路制御においては、その接続ルータ数、経路数が増えるにしたがって、その収束までの時間は、それに比例するということができる。

では、なぜこのように非常に多くの経路が存在するのであろうか。一つは新規に AS およびプレフィックスが割り当てられそれが default free zone に注入されることが考えられる、もうひとつは、R. Govidan が指摘しているマルチホームの問題がある [8]。CIDR 導入以降、インターネットにおける経路 (プレフィックス) は、最大限に上流サイトで集約して default free zone に注入するように推奨されてきた。しかし、上位 ISP へのコネクティビティを 2 つ以上もつマルチホーム組織では、最適な経路を選択するため、上位 ISP からフル・ルートを受信し、なおかつ上位 ISP 経由で default free zone に自身のプレフィックスを注入する。これによってマルチホーム組織は、最適な経路を受信することが可能になるが、インターネット全体に与えるインパクトは大きい。つまり、こうしたマルチホーム組織に関する UPDATE も default free zone に注入され、その数は莫大なものになる。G. Labovits らの調査によると [9]、コア・インターネットバックボーンの経路のうち、25 パーセントがマルチホーム組織のものであるとの報告もあり、マルチホーム問題をふくめ経路情報の収束は大きな問題となっている。

2.2.2 AS 番号の管理

2.1.2 小節で述べたように、ドメイン間経路制御では、AS 番号を付与した AS-PATH をピア同士で交換することによって、経路情報を交換する。ここで重要になるのが、重複なく AS 番号を割り振るかということである。これは、IP アドレスと同様、AS 番号に重複があっては、同じ AS

番号を保有するまったく別のドメインが、全く違う経路を広告するなどケースが考えられ、インターネット全体に与える影響は多大である。

ゆえに、現在AS番号の割り当てはIPアドレス同様にインターネット資源割り当て機関であるICANN(かつてのIANA)を頂点に、RIR(Regional Internet Registry)と呼ばれる世界に3ヶ所存在するインターネット資源割り当て機関(ヨーロッパ・アフリカ地域はRIPE、アジア・環太平洋地域は、APNICそしてアメリカ地域はARIN)が、AS番号の割り当て業務を担当している。
現在のAS番号の割り当てについて述べたが、IPアドレス同様、AS番号に関してもも、その枯渇が指摘されている。現在、AS番号のフィールド長は、2バイトであり、原理的に65,536個のASが存在可能であるが、プライベートASとして、64512 ~ 65535まで予約されているので、グローバルインターネットで使用することができるAS数は、64,512個である。そのうち現在では、すでに20,000以上AS番号が割り当てられており、現状のままの割り当てを続ければ、AS番号が枯渇する恐れもある。また、現在、IPアドレスでは、IPv4からIPv6への移行が始まりつつあるが、IPv6になってもAS番号は、IPv4と共通であり、参入障壁が低いIPv6において、AS番号を取得するという動きもあり、さらにAS番号の割り当て数が増加することも考えることができる。
こうしたAS番号の枯渇を防ぐために、インターネットの標準化団体であるIETF(Internet Engineering Task Force)では、AS番号フィールド長を4バイトにする提案が現在、IETFのIDR(Inter Domain Routing)分科会において議論中である[10]。

2.2.3 ブラックホール

次にドメイン間経路制御中の問題で、本論文において中心的に扱うブラックホール問題について、述べる。

正常なドメイン間経路制御

ブラックホール現象とは、どのような現象であるかについて議論する前に、正常なドメイン間経路制御の状態について述べる。
図2.4は、正常なドメイン間経路制御について示している。ここでの要点は、以下である。

- AS5は、2001:500::/35という経路を生成、その到達性情報(NLRI)を近隣ASに広告している。

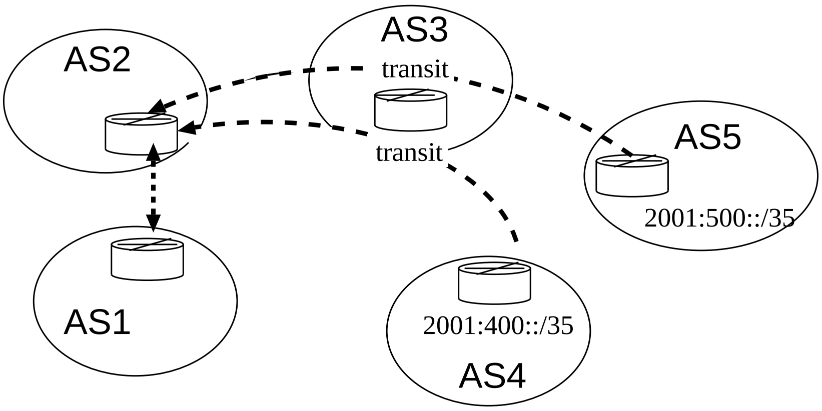


図 2.4: 正常なドメイン間経路制御

- AS4 は、2001:400::/35 という経路を同様に生成し、その到達性情報を近隣 AS に広告している。
- AS3 は、AS5 および AS4 の経路をトランジットし、AS2 および AS1 に広告する。
- AS1 および AS2 は、AS5 からの経路は AS3 経由で受け取る。

ブラックホール経路の発生

次に、先述した正常な経路交換から、どのようにしてブラックホールが発生するかについて述べる。

図 2.5 は、ブラックホール経路が発生している状況を示している。すなわち、

- プレフィックス 2001:600::/35 は、AS6 が本来は広告すべてきはずのプレフィックスである
- しかし、設定ミスなどで、AS4 が 2001:600::/35 をこの場合、誤って広告している
- AS3 は、2 種類の同じ NLRI(2001:600::/35) を含む UPDATE を受信する
  1. オリジン AS=AS6 からの正しい UPDATE (AS-PATH = AS6 → AS5)
  2. AS4 からの誤ったオリジン AS を含む UPDATE (AS-PATH = AS4)

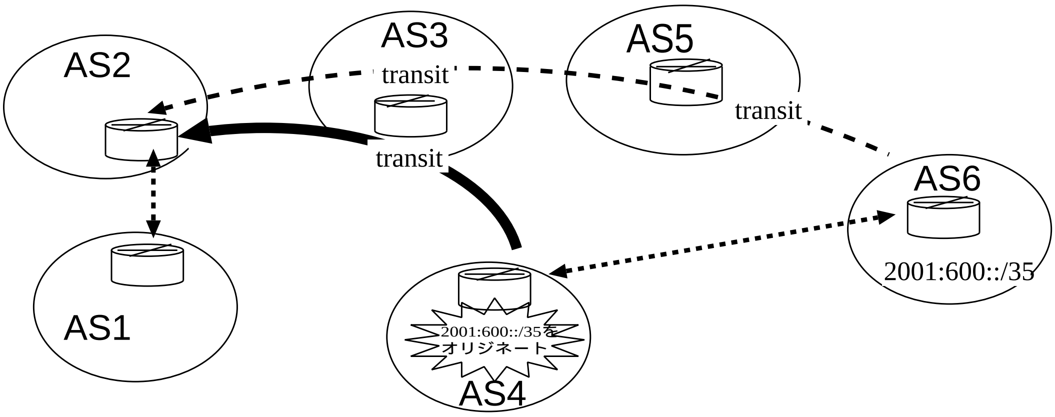


図 2.5: ブラックホール経路の発生

- AS3 は、上記の UPDATE を比較し、AS-PATH の短い AS4 の UPDATE を最良経路として採用し、それを AS2 に UPDATE する
- AS2 は、AS3 からの UPDATE を受信し、最良経路として採用し、それを AS1 に UPDATE する
- AS1 もこれと同様に、NLRI=2001:600::/35 のオリジン AS は AS3 とし、経路表に注入する
- 結果的に、AS3、AS2、AS1 から NLRI=2001:600::/35 への到達性が失われる

このように、本来経路を生成すべき AS でない AS が、本来経路を生成すべき AS にかわって経路をオリジネートし、到達性が失われる現象を経路のブラックホール化と呼ぶ。

ブラックホール問題発生の原因

では、なぜ上記のようなブラックホール経路問題が発生するのか、ここではその原因について検討する。前述では、設定ミスと仮定したが、それを含めて他に考えられる原因およびその根拠として、X. Zhao 達は、幾つかの原因を指摘している [11]。

BGP を用いないマルチホーム 2 つの AS 間に専用回線が存在しており、その回線間は、BGP ではなく、ドメイン内経路制御プロトコル (static,OSPF,RIP など) などを用いて、プレフィッ

クス情報を交換する。これがBGPに注入され、AS外に広告すると、本来広告すべきではないASが、そのプレフィックスを広告することになる。

誤った設定 設定を誤った結果、他のプレフィックスをアナウンスしてしまう現象をさす。彼らの調査では、2001年4月10日のBGPの経路情報を観測 解析したところ、277のMOAS(Multiple Origin AS)のうち、5,532件が設定誤りによって発生したと指摘している。

現時点でのブラックホール対策

先述した経路のブラックホール問題であるが、現状では、こうした問題が発生した際、

1. ブラックホール経路をオリジネートしているASを、RRDB(Routing Registry Database)から検索する
2. 検索の結果、そのASに関する管理者情報が記載されているので、その管理者情報に電子メールなどで連絡して、状況を伝える

という手段が一般的である。しかし、L.Gaoが指摘しているように[12]、RRDBは、すべてのASあるいは経路に関する情報が記載されているとはかぎらない、かつ更新されるタイミングが遅いので、一時的にルータの経路フィルターを変更するなどの対策をとるのが現在では一般的となっている。

そこで本研究では、このブラックホール問題に対象を絞り、どのようにしてこうした現象自動的に検出し、抑制するかということについてこれから議論する。

第3章　関連研究

前章では、現在のドメイン間経路制御に関する概要と、そこで発生している問題、とくにブラックホールを対象として、その問題点について議論した。本章では、これに関連して現在ドメイン間経路制御では、どのような研究が行われているかについて述べる。

3.1 BGP Route Flap Damping

BGP Route Flap Damping[13] は、ルート・フラッピングと呼ばれる同一のプレフィックスが、UPDATE と WITHDRAWN を繰り返すことによって膨大なネットワークトラフィックを増加させる現象を抑制することを目的に考案された仕組みである。
BGP Route Flap Damping では、プレフィックスを well-behaved と ill-behaved の二つに分類する。well-behaved とは、ある継続した期間において高い安定性を示しているプレフィックスを指す。一方、ill-behaved とは、短い時間で頻繁に UPDATE/WITHDRAW が生じる状態を示している。BGP Route Flap Damping では、この ill-behaved な状態がある程度将来的に継続するものであるとし、そのプレフィックスが安定するまで、その広告を抑制するメカニズムを提供する。将来にわたって ill-behaved な状態が継続することを判断するために、BGP Route Flap Damping ではプレフィックス情報の履歴を蓄積する。この履歴とは、ある期間にプレフィックスがフラップした履歴であり、BGP Route Flap Damping ではフラップする度にペナルティが増加する。このペナルティがあらかじめ与えられた閾値を越えると、プレフィックスは抑制される。なお、経路が抑制されても、ペナルティはその経路が依然としてフラップし続けていれば与え続けられる。また、プレフィックスの抑制を解除し、再び広告しはじめる場合に関しても、上記と同様な基準が適用されるが、再広告の場合は、そのペナルティが指数関数的に減少するアルゴリズムを採用している。以下は、BGP Route Flap Damping においてプレフィックスを抑制するために利用するパラメータである。

- penalty フラップするたびにプレフィックスに割り当てられる値
- half-life-time ペナルティが半分減るのに経過しなければいけない時間

- suppress limit ペナルティがこの値を越えたら、そのプレフィックスは抑制される

- reuse limit ペナルティがこの値よりも小さければ、抑制していたプレフィックスは再び広告をはじめることができる

- history entry フラップ情報を蓄積するために用いる。

図3.1は、BGP Route Flap Dampingの遷移について示している。

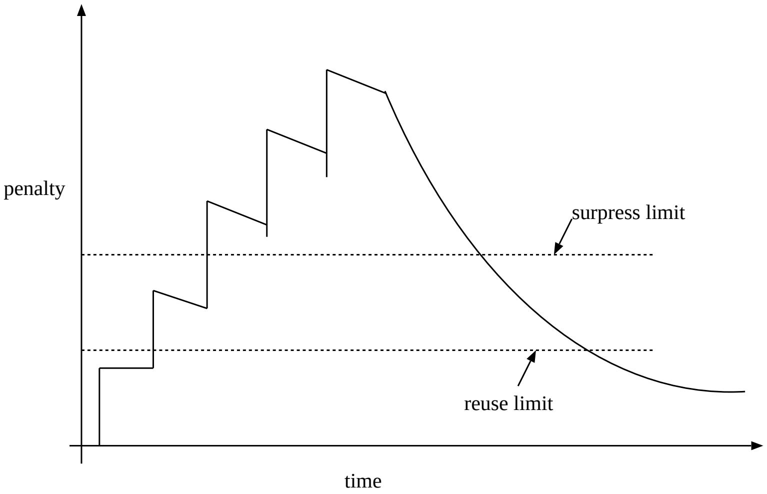


図 3.1: BGP Route Flap Damping の遷移

ここでは、プレフィックスがフラッピングする度にペナルティが付与される一方で、フラッピングがとまると、指数関数的にそのペナルティは減少する。

BGP Route Flap Dampingは、本来であれば各ASは、自身の持つプレフィックスを集約してAS外に広告すべきであるが、2.2.1節で述べたように、マルチホームなどの影響により、必ずしも集約したプレフィックスをAS外に広告していないのが現状であり、これがBGPの頻繁なフラッピングを生む大きな原因となっている。こうした頻繁なフラッピングに対して、ルートダンピングは、フラッピングを検出し、抑制することは可能であるが、そのフラッピングしているプレフィックスが本来広告されるべきもので、定常な状態であるかなどについては、検出することはできない。

3.2 BGP Community

ドメイン間経路制御は、何度か述べたようにポリシールーティングと呼ばれ、各ドメインのルーティングポリシーにしたがって経路交換をおこなう。このルーティングポリシーの一つの要素として利用されているのが、BGP Community 属性[14]である。
コミュニティ属性とは、ある共通の特性を共有する、宛先のグループと定義することができる。コミュニティ属性は、一つのネットワークや一つのASに制限されておらず、物理的な限界はない。たとえば、あるISPに特有のコミュニティ属性を定義する、このコミュニティはBGPで経路交換しているすべてのBGPルータで共有可能である。つまり、コミュニティ属性はプレフィックスやAS番号などとは異なり、論理的に各コミュニティ属性を判断するので、ルーティングポリシーを簡略化するため、しばしば用いられる。すなわち、BGPルータは、UPDATE中に含まれるコミュニティ属性を参照し、どのプレフィックスを受け取り、どれを近隣のBGPルータに渡すかを判断する。
コミュニティ属性は、UPDATEメッセージ中に含まれる4バイトのオプション属性で、0x00000000から0x0000FFFFおよび0xFFFF0000から0xFFFFFFFFまでの範囲のコミュニティは予約されている。これらのコミュニティはwell-knownコミュニティとして、以下の属性を持つ。

NO_EXPORT (0xFFFFFF01) BGPルータが、このコミュニティ値を受け取ったら、ASコンフェデレーション内部および外部ピアに広告することができない。

NO_ADVERTISE (0xFFFFFF01) BGPルータが、このコミュニティ値を受け取ったら、他のBGPピアに、このUPDATEを広告してはならない。

NO_EXPORT_SUBCONFED(0xFFFFFF03) BGPルータが、このコミュニティ値を受け取ったら、外部のピアには、広告してはならない。ASコンフェデレーションで定義した内部のピアのみに広告することができる。

well-knownコニュニティ属性のほかに、プライベートなコミュニティ属性を、特別な使用目的のために使用することがある。多くのISPでは、コミュニティ属性のはじめの2バイトをAS番号に使い、最後の2バイトでそのASに関連する値を定義するというものである。たとえば、“export-group”というプライベートコミュニティを定義したいISPは(AS65423)、65423:1というコミュニティを定義することができる。これは、AS65423が1というコミュニティ属性を定義することを意味し、1であればその経路を広告するなど、各ISP内で定義し、ポリシールーティングをおこなう。

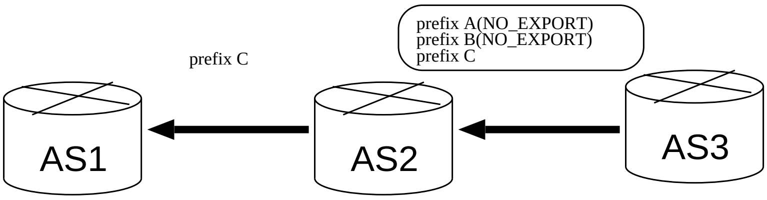


図 3.2: コミュニティ属性の利用

図 3.2 は、コミュニティ属性の利用例をあらわしている。AS-3 は、AS-2 に対し、いくつか NO_EXPORT コミュニティを付与して広告している。AS-2 は、プレフィックス Z のみ受信し、それを AS-1 に広告する。

3.3 S-BGP

S-BGP (Secure BGP)[15] は、現在の BGP にセキュリティ機能を拡張した BGP である。S-BGP は、セキュアな BGP の運用を目的に設計されたものであるが、そのセキュアな BGP の運用とは以下の状態をさす。

1. BGP スピーカーは、ピアから送出された BGP UPDATE を受信し、その情報を変更することなく、その他のピアに送出する。また、送出する情報は、以前に送出した古い情報でなく、その UPDATE から送られた最新の情報を指定したピアに送出する。
2. UPDATE は、指定した AS もしくは BGP スピーカーのもとに送出する。
3. UPDATE は、その AS において認証され、正しければそのルーティング情報を BGP スピーカに送出する。
4. ある AS から生成する経路は、その経路を所有すべき機関によって、認証され、認証が成功することによってはじめて UPDATE として送出することが可能である。
5. UPDATE 中に含まれる各組織の IP アドレススペースは、ICANN を頂点とする階層構造に添って割り当てられる。
6. UPDATE 中に取消し経路があった場合、これに関しても認証をおこなう。
7. BGP スピーカーは UPDATE をおくり、ピアはその UPDATE を正しく受取り、ローカルルーティングポリシーに添って正しく処理する。

上記のセキュアな BGP の要件を満たすべく S-BGP では、1.PKI、2.Attestations 3.IPsec の三つのセキュリティ技術を導入することによって、その要件の実現を図っている。

PKI(Public Key Infrastructure) PKIは、UPDATE中のAS番号、IPアドレス、ルータIDを認証するために用いる。PKIの構造としては、図3.3のように、IPアドレスでは、ICANNを頂点に階層構造をとり、各組織に割り当てられたIPアドレススペースを認証する。

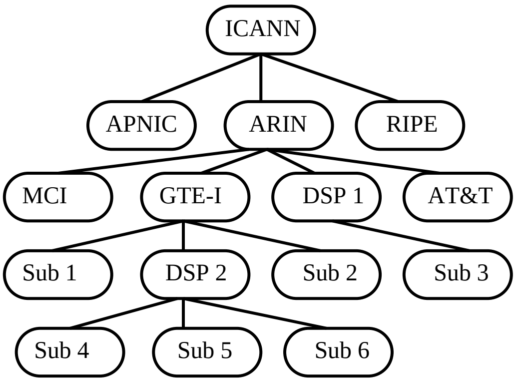


図3.3: IPアドレススペースにおけるPKIストラクチャー

attestations attestation (宣誓)は、上述のPKI機構から生成した証明書と鍵を利用して、その署名と検証にあたる。具体的には、UPDATE中のパス属性に以下の2種類のattestationを定義し、IPアドレススペースが正しいか、オリジンASが正しいかなどについて検証する。

- Route Attestations (RAs)
  それぞれのAS(あるいはそのASを代表するルータ)が発行するattestationであり、経路に関する検証をおこなう。
- Address Attestations (AAs)
  BGPでアナウンスするアドレススペースに関するattestationであり、そのアドレススペースを所有するASが署名・検証する。これによって、同じプレフィックスを複数のASが広告することを回避することができる。

図3.4は、BGP UPDATEメッセージ中に含まれるRoute Attestationsについて図示したものである。

IPsec S-BGPでは、BGPスピーカーとピアとのTCPセッションを第三者によって盗聴されな

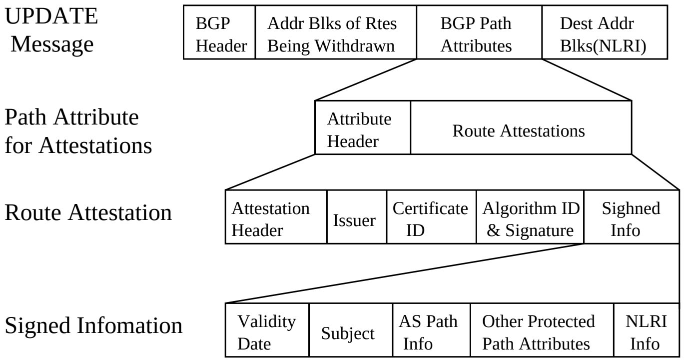


図 3.4: BGP UPDATE 中の Route Attestations

いように IPsec の ESP(Encapsulating Security Payload) を用いる。

以上、S-BGP のシステム構成について述べたが、このシステムがすべての BGP に導入されれば、2.2.3 節で指摘したブラックホール問題は解決することが可能であるが、問題はその普及である。普及のためのポイントとして、論文では、

1. BGP パス属性に、新たな属性を追加するだけなので、既存の BGP には影響を与えない
2. CPU は、Pentium200Mhz 程度で十分耐えることができ、証明書を蓄積するハードディスク容量も 32M バイトあれば十分であることから多くの BGP ルータで十分なパフォーマンスを得ることができる

などの理由をあげている。

第4章　ABICアーキテクチャーについて

本章では、2.2.3 小節において指摘したブラックホール問題について、その問題を解決するために必要な要件を整理し、動的なブラックホール検出システムである ABIC
(Automatically BGP Integrity Check) について他の手法と比較しながら、議論する。

4.1　必要な要件

本節では、ブラックホール経路を動的に検出するために、どのような要素が必要であるかという要件について整理する。
2.2.3 小節で述べたように、ブラックホール経路の本質的な原因は、本来は経路を広告すべきでない AS が、何らかの原因で誤って経路を広告してしまうことであり、

1. 本来、広告すべき AS
2. 誤って広告している AS

の両方をマークし、比較すれば、動的に検出することが可能である。また、RFC2791[7] で指摘しているように、ドメイン間経路制御は、ドメイン内経路制御にくらべて、その規模性および安全性が重要になる。以上より、動的にブラックホール経路を検出するために必要な要件は、

- 本来、広告すべき AS とブラックホール経路を広告している AS との整合性
- 安全性
- 規模性

ということができる。次節では、それぞれの要素について具体的に検討する。

4.1.1　オリジン AS 情報の整合性

前述のように、ブラックホール経路を検出するためには、UPADATE メッセージ中に含まれる経路が、本来どの AS からオリジネートされるかべきかについて正確にトラックしてあるデー

タベースが必要である。

こうした機能を実現するデータベースとして、現在IRR(Internet Routing Registry) とDNS(Domain Name System) が存在する。本節では、この2種類について、長所および短所について検討する。

IRR(Internet Routing Registry) を用いる方法

IRR は、世界中に存在するドメイン間経路制御に関する各種情報を蓄積し、それをデータベースとして提供することによって、ドメイン間経路制御の障害検出に役立てるべく成立した。この原型ができたのが1995 年であり、当初は、アメリカ合衆国のMerit Networks、ANS(現在のUUnet)、InternetMCI(現在のCable and Wireless)、オランダのRIPE Network Coordination Centre(NCC) とカナダのCAnet(現在のBell Canada) が中心となって、これらの組織が管理しているAS や経路などを登録している。

登録するデータベースは、大きくわけて、

1. 誰がその経路およびAS 番号を登録したかを記述する登録者オブジェクト(MAINTAINTER オブジェクト)
2. AS に関するポリシー記述したAS オブジェクト(AUT-NUM オブジェクト)
3. AS から生成される経路に関するポリシーを記述した経路オブジェクト(ROUTE オブジェクト)

が存在する。

図4.1 は、IRR に登録してある経路オブジェクトの例である。route は、経路オブジェクトのキーであり、その属性として、descr(経路に関する説明)、origin(経路をオリジンするAS、記載必須)、mnt-by(登録者オブジェクト、だれがこのオブジェクトを更新する権限があるのかを示す)、changed(更新者のメールアドレスと、更新日) そしてsource(メンテナンスしているデータベース名) が記載されている。

これをブラックホール経路検出のために必要なデータベースとして用いるためには、一意に識別可能なroute 属性をキーにして、そこからorigin 属性を取得する方法が考えられる。

```
route:         204.70.2.80/28
descr:         ROUTE-AA
origin:        AS653561
mnt-by:        CW
changed:       aai@aa.net 20011007
source:        AA
```

図 4.1: IRR データベース中の ROUTE オブジェクト

DNS を用いる方法

DNS(Domain Name System)[16, 17] は、ルート (.) を頂点として、階層的にホスト名から IP アドレス、IP アドレスからホスト名を変換する仕組みである。“DNS-based NLRI origin AS verification in BGP” [18] は、この DNS の仕組みを拡張し、どの経路 (プレフィックス) が、どの AS に属しているかを DNS を通じて把握することが可能になる。

図 4.2 は、DNS に新たな AS レコードを追加したゾーンファイル例を示している。これは以下のように説明することができる。

1. bgp.in-addr.arpa. ゾーンのルートは、205/8 を arin.net にゾーン委譲する (1)。
2. arin.net は、AS リソースレコードより、AS 番号は 65535 をもち、/8 のプレフィックス長を保持している。
3. arin.net は、同時に digex.net. に、205.0/16 および verio.net. に 205.1/16 を割り当て、ゾーンを委譲している。 (2)
4. arin.net から、0.205.bgp.in-addr.arpa. ゾーンを委譲された digex.net は、 AS リソースレコードにより、AS 番号は 2548 で、205.0/16 を保持している。 (3)
5. arin.net から、1.205.bgp.in-addr.arpa. ゾーンを委譲された verio.net. は、AS 番号 1914 で、プレフィックス 205.1/16 を割り当てられている。 (4)
6. verio.net は、AS777 に、205.1.0/24、AS2914 に、205.1.1/24 を割り当てている。

上記の DNS の新たなリソースレコードをに定義し、そのなかにプレフィックス情報、AS 情報を格納することによって、IRR では実現できていない階層的にかつ分散的に上記の情報を管理することが可能になる。

両手法の比較

ここでは、前述したIRRおよびDNSの二つの手法について両者の短所と長所を列挙しつつ、どちらがオリジンAS情報の整合性を保持するデータベースとしてふさわしいか議論する。
まず、IRRの長所として、以下の3点をあげることができる。

- 現在、多くのISPが登録し、利用されている
- whoisで簡単にアクセスすることができる
- IRRの先述したコア組織に関しては、データをミラーしているので、一つのDBサーバにアクセスできなくても、他を利用することができる

また、IRRの問題点としては、以下が存在する

1. IRRにデータを登録・変更してから反映されるまでに時間がかかる
   IRRは複数のデータベースと同期を取っているため、登録したデータが反映されるまで、時間がかかる。

2. 認証プロセスが脆弱
   現在、多くのIRRでは、認証にメールのFromによるマッチングあるいはパスワードによって認証しているが、これでは第3者によってデータを改竄されてしまう可能性も否定できない。それに対応するため、PGP(Pretty Good Privacy)を使った、オブジェクトの更新方法へ移行しつつあるが、まだ普及していない点で、セキュリティには問題点がある。

3. BGP-4で交換されるすべての経路情報が登録されていない
   BGP-4には、プライベートピアと呼ばれる商用に多く用いられる公開されていないピアが数多く存在する。こうした情報は、IRRには記載されておらず、すべての経路が登録されているとはいえない。

つぎに、DNSの場合について、まず長所を検討する。

- 規模性
  現在、インターネット全体で機能しているDNSの枠組みを利用できるので規模性はあるということができる。

- 階層構造
  前述の規模性を維持できる原因のひとつとして、DNS がルートを頂点として階層構造であることがある。AS リソースレコードでも、“bgp.in-addr.arpa.” を頂点に、階層構造になっているため、ある特定のサーバに負荷が集中しない。

同様に、DNS の短所は、以下の点である。

1. AS リソースレコード
   DNS を用いた手法を実現させるためには、AS リソースレコードを新たに定義しなくてはならないが、これは非常に多くのネームサーバを改変する必要がある。
2. “bgp.in-addr.arpa.” の普及
   AS リソースレコードと同様に、原則として bgp.in-addr.arpa も BGP-4 で経路を広告している AS すべて登録する必要があり、その普及もまだなにも進められていない。

以上、IRR の長所・短所、DNS の長所・短所について検討した。IRR は、データベース反映が遅いなど集中管理システムのもつデメリットがいくつかあるが、現在機能しているシステムである。一方、DNS 自身は、その規模性はいうまでもないが、AS リソースレコードおよび bgp.in-addr.arpa. の普及は全くされていない状況であり、それを用いるのは、現実的ではないといえる。そこで、本研究では、オリジン AS データベースとして IRR を用いるが、IRR でもっとも問題になるのが短所の 3. で指摘した、すべての経路情報が IRR に登録されていないという点である。もしも、ほとんどの経路が IRR に登録されていなければ、“正しい” 経路が管理されているデータベースからほど遠いものになり、その信頼性は限りなく低くなってしまう。そこで、ここではすべての経路情報であるフル・ルートが、どの程度の割合で IRR に登録されているか計測をし、果たして IRR を整合性をとるためのふさわしいデータベースであるか検証する。

フル・ルート

本研究は、IPv6 での利用を仮定しているが、IPv6 で問題となってくるのは、現時点で IPv6 での IRR が存在しないということである。また、フル・ルートの数も 300 を越える程度であり、まだ発展段階に過ぎない。そこで、ここでは IPv4 の環境で測定した。現時点 (2001 年 12 月 29 日) での、BGP で交換している経路 (フル・ルート) は、1,103,973 経路であり、そのプレフィックス長ごとの分布を、表 4.1 に示す。

BGP 経路のうち半数にあたる 56%が/24 のプレフィックス長を持っていることが、表 4.1 から

表 4.1: プレフィックス長ごとの BGP 経路の分布

prefix 長	prefix 数	prefix 数/全 prefix 率
/8	19	0.02%
/9	5	0.00%
/10	7	0.01%
/11	11	0.01%
/12	37	0.04%
/13	91	0.09%
/14	226	0.22%
/15	384	0.37%
/16	6994	6.73%
/17	1341	1.29%
/18	2434	2.34%
/19	7116	6.84%
/20	6205	5.97%
/21	4635	4.46%
/22	7043	6.77%
/23	8646	8.32%
/24	58775	56.53%
/28	1	0.00%
/32	3	0.00%

わかる。

IRR-DB

BGP の経路数およびプレフィックス長の分布について述べたところで、次に IRR の計測に移る。IRR は、世界各地に存在し、現在は 44 もの IRR データベースから構成されている。これからのデータベースは FTP サイト ftp.ra.net で公開されており、1 日に 1 度、同期をとっている。ここでは、44 の IRR データベースのうち、もっともファイルサイズが大きい RA プロジェクトによる IRR データベース (RA.DB) およびヨーロッパ・アフリカの地域レジストリ (Local Internet Registry) である RIPE による IRR データベース (RIPE.DB) の 2 種類を分析対象とする。前述のように、IRR データベース中には、AS オブジェクト、AS-SET オブジェクト、ROUTE オブジェクト、MAINTAINER オブジェクトが存在するが、そのなかで図 4.1 に示した ROUTE オブジェクトに注目し、そのキーである route 属性を抽出し、その総数およびプレフィックス長の分

布を RA.DB および RIPE.DB それぞれ分析した。

RA.DB

はじめに、RA.DB についての分析結果を示す。RA.DB に登録されている ROUTE オブジェクトの総数は、76,083 件あり、そのプレフィックス長ごとの分布を表 4.2 に示す。

表 4.2: RA.DB における ROUTE オブジェクト分布

prefix 長	prefix 数	prefix 数/全 prefix 率
/8	25	0.03%
/9	5	0.01%
/10	3	0.00%
/11	3	0.00%
/12	25	0.03%
/13	67	0.09%
/14	162	0.21%
/15	260	0.34%
/16	6234	8.19%
/17	897	1.18%
/18	1688	2.22%
/19	4705	6.18%
/20	4228	5.56%
/21	3091	4.06%
/22	4453	5.85%
/23	5356	7.04%
/24	43966	57.79%
/25	192	0.25%
/26	152	0.20%
/27	165	0.22%
/28	175	0.23%
/29	109	0.14%
/30	67	0.09%
/31	1	0.00%
/32	53	0.07%

RIPE.DB

つぎに、RIPE.DB についての分析結果を RA.DB と同様に示す。RIPE.DB に登録されている ROUTE オブジェクトの総数は、247,790 件であり、そのプレフィックス長ごとの分布を表 4.3 に示す。

表 4.3: RIPE.DB における ROUTE オブジェクト分布

prefix 長	prefix 数	prefix 数/全 prefix 率
/8	4	0.02%
/9	1	0.00%
/10	1	0.00%
/11	3	0.01%
/12	2	0.01%
/13	12	0.05%
/14	50	0.20%
/15	149	0.60%
/16	1965	7.93%
/17	280	1.13%
/18	515	2.08%
/19	2506	10.11%
/20	1265	5.10%
/21	831	3.35%
/22	1190	4.80%
/23	1712	6.91%
/24	13530	54.58%
/25	117	0.47%
/26	101	0.41%
/27	54	0.22%
/28	51	0.21%
/29	179	0.72%
/30	257	1.04%
/32	14	0.06%

BGP 経路と RA.DB および RIPE.DB(1)

次に、BGP 経路中に、どの程度の割合で RA.DB 中のプレフィックスおよび RIPE.DB 中のプレフィックスが含まれているか、すなわちフル・ルート中のプレフィックスが、どの程度それぞ

れのIRRデータベースに登録されているかについて分析した。分析にあたって、IRRプレフィックスとBGP経路との一致基準として、すべてプレフィックス+プレフィックス長による完全一致であり、プレフィックスのみで一致したものは含まない。このRA.DBに関しての、結果を、表4.4に示す。

表 4.4: BGP経路中におけるRA.DBプレフィックスの分布

prefix長	RA.DB/BGP経路	RA.DB/BGP経路
/8	13 / 19	68.42%
/9	3 / 5	60.00%
/10	2 / 7	28.57%
/11	3 / 11	27.27%
/12	18 / 37	48.65%
/13	46 / 91	50.55%
/14	106 / 226	46.90%
/15	150 / 384	39.06%
/16	5039 / 6994	72.05%
/17	599 / 1341	44.67%
/18	1137 / 2434	46.71%
/19	3519 / 7116	49.45%
/20	2494 / 6205	40.19%
/21	1805 / 4635	38.94%
/22	2429 / 7043	34.49%
/23	2703 / 8646	31.26%
/24	21201 / 58775	36.07%

BGP経路中にRA.DBに登録されていたプレフィックス数は、41,267件存在し、BGP経路全体の39%にあたる。

またRIPE.DBについては、表4.5にRA.DBと同様にプレフィックス長ごとの分布を示す。

BGP経路中にRIPE.DBに登録されていたプレフィックス数は、14,030件存在し、それはBGP経路全体の13%に相当する。

RA.DBおよびRIPE.DBで重複しているプレフィックス

つぎに、IRRデータベース(ここではRA.DBとRIPE.DB)登録されているBGP経路のなかで、ひとつのプレフィックスが、2つのIRRデータベースにROUTEオブジェクトとして登録されていないかについて分析する。つまり、ここでは、RA.DBに登録されながらも、かつRIPE.DB

表 4.5: BGP 経路中における RIPE.DB プレフィックスの分布

prefix 長	RIPE.DB/BGP 経路	RUPE.DB/BGP 経路
/8	3 / 19	15.79%
/9	1 / 5	20.00%
/10	1 / 7	14.29%
/11	3 / 11	27.27%
/12	2 / 37	5.41%
/13	11 / 91	12.09%
/14	48 / 226	21.24%
/15	126 / 384	32.81%
/16	1762 / 6994	25.19%
/17	235 / 1341	17.52%
/18	410 / 2434	16.84%
/19	2200 / 7116	30.92%
/20	973 / 6205	15.68%
/21	553 / 4635	11.93%
/22	816 / 7043	11.59%
/23	1061 / 8646	12.27%
/24	5825 / 58775	9.91%

にも登録されている、いわば二重登録している ROUTE オブジェクトについて、その件数について分析する。一致の判別は BGP 経路と RA.DB およびプレフィックスの判別同様、すべて完全一致であり、それをプレフィックス長ごとの分布示しているのが、表 4.6 である。

この結果より、合計重複プレフィックス数は、1,690 件である、つまり IRR データベースに登録されている BGP 経路 41,267 件のうち 4%にあたる 1,690 件が二重に登録されていることになる。

BGP 経路と RA.DB および RIPE.DB(2)

上記の結果より、RA.DB と RIPE.DB は、幾つか重複するものの異なる DB をもっていることがわかった。つまり、BGP 経路中の IRR データベース全体 (この場合は、RA.DB および RIPE.DB) の登録率は、

$$IRR - DB \text{ 登録率} = \frac{RA.DBprefix \text{ 数} + RIPE.DBprefix \text{ 数} - \text{重複 } prefix \text{ 数}}{BGP \text{ 経路数}}$$

であらわすことができる。表 4.7 は、この IRR-DB 登録率をプレフィックス長ごとに示している。

表 4.6: RA.DB、RIPE.DB での重複プレフィックス分布

prefix 長	重複 prefix	重複 prefix/全重複 prefix 率
/8	1	0.06%
/14	2	0.12%
/15	4	0.24%
/16	224	13.25%
/17	20	1.18%
/18	38	2.25%
/19	262	15.50%
/20	86	5.09%
/21	77	4.56%
/22	100	5.92%
/23	98	5.80%
/24	778	46.04%

BGP 経路として BGP が広告しているプレフィックスのうち、IRR データベース全体において ROUTE オブジェクトとして登録している件数は、53,667 件であり、BGP 経路の 51%にあたる。

IRR データベース登録率に関するまとめ

以上までの計測結果をまとめると、以下のことが言える。

1. 個々の IRR データベースでの登録率は、それほど高いとはいえない
2. しかし、それぞれの IRR データベースが、幾つか重複しながらも、別々のプレフィックスが登録されている
3. そのプレフィックスを集約し、重複するプレフィックスを除けば、IRR データベース登録率は向上すると考えることができる
4. 現在の IRR では、DNS のように再帰的にクエリーを転送する仕組みがないので、これを導入するなどして、どの IRR データベースにどのプレフィックスが登録されているかを検索できる仕組みを導入するなどすれば、オリジン AS の整合性をとるデータベースとして IRR を利用することが可能である

表4.7: IRR-DB登録率

prefix長	BGP経路	RA.DB	RIPE.DB	重複数	IRR総数	IRR-DB登録率
/8	19	13	3	1	15	78.95%
/9	5	3	1	0	4	80.00%
/10	7	2	1	0	3	42.86%
/11	11	3	3	0	6	54.55%
/12	37	18	2	0	20	54.05%
/13	91	46	11	0	57	62.64%
/14	226	106	48	2	152	67.26%
/15	384	150	126	4	272	70.83%
/16	6994	5039	1762	224	6577	94.04%
/17	1341	599	235	20	814	60.70%
/18	2434	1137	410	38	1509	62.00%
/19	7116	3519	2200	262	5457	76.69%
/20	6205	2494	973	86	3381	54.49%
/21	4635	1805	553	77	2281	49.21%
/22	7043	2429	816	100	3145	44.65%
/23	8646	2703	1061	98	3666	42.40%
/24	58775	21201	5825	778	26248	44.66%
/28	1	0	0	0	0	0.00%
/32	3	0	0	0	0	0.00%

4.1.2 セキュリティ

つぎに、動的にブラックホール経路検出において必要なセキュリティーについての要件を整理する。ここでのセキュリティとは、

- オリジンASデータベースが改竄されず、正しい情報が格納されている

を指す。前者のデータベースに関しては、第3者によって改竄できない仕組みが必要である。

4.1.3 規模性

最後に、動的にブラックホール経路検出で必要な最後の要素である規模性について検討する。ここでの規模性とは、提案するシステムが、現在のBGP-4環境に適応するかどうかを意味する。そこで現在のBGP-4環境がどのようになっているか、表4.8の環境において計測した。

表 4.8: 計測環境

期間	2001 年 11 月 20 日 16:00 ~ 2001 年 11 月 22 日 16:00
測定環境	NSPIXP6(IPv6 native IX) 上の eBGP ルータ
測定マシン環境	Pentium III 500Mhz 256MB メモリ OS NetBSD 1.5.1
接続環境	ファーストイーサネット (100base-TX)
ピア数	13 ピア

測定マシンは、13 の組織とピアを張っており、ピアからの UPDAT がある毎に、その情報をファイルに記録した。表 4.9 は、それらの UPDATE 数について解析したものである。

表 4.9: UPDATE 数

UPDATE 送出数	44,677
ユニークプレフィックス数	606
全送出 UPDATE におけるユニークプレフィックスの割合	1.4 パーセント
UPDATE 受信数	23,671
ユニークプレフィックス数	302
全受信 UPDATE におけるユニークプレフィックスの割合	1.2 パーセント

ユニークプレフィックスの割合とは、すべてのUPDATEメッセージ中に含まれているプレフィックスのうちユニーク (一意) な数を指す。つまり、UPDATE 送出の場合は、ユニークなのは 1.4 パーセントで、残りの 98.6 パーセントは、UPDATE メッセージに 2 度以上おなじプレフィックスが出現していることを意味する。これは、3.1 節で指摘したルート ・フラッピングがおきていることを意味する。RFC2791[7] によれば、IPv4 における大規模な IX では、1 分間に数千回ものフラッピングが起きていると指摘しているが、IPv6 の場合は、規模が IPv4 に比べて小さいこともあり、フラッピング自体は存在しているものの、その数はすくないといえる。

ただ、フラッピングは、同じ経路が UPDATE ・WITHDROWN を繰り返しているだけであり、そのたびにデータベースに問い合わせるのは、ルータのオーバーヘッドになる。故に、こうしたオーバーヘッドを軽減するため、一度問い合わせた経路は、BGP ルータにキャッシュする仕組みも必要である。

図 4.3 は、1 時間ごとの送出 ・受信 UPDATE を示している。これによると、1 時間にごとの最大の UPDATE 数は、送出が約 8,700 回、受信が、2,100 回であり、この原因としては、他 AS のルータがダウンした、経路を変更したなどが考えられる。ここからいえることは、現実の BGP-4 での環境では、ピークで送出は 8,700 回、受信で 2,100 回の UPDATE を処理することができれば、提案する手法が、現在の BGP-4 環境でも適用することが可能になる。

4.2 ABIC(Automaticlly BGP Integrity Check)

4.1 節では、動的にブラックホール経路を検出する機構のために必要な要件を、(1) データベース (2) 安全性、(3) 規模性という側面から整理した。これを踏まえて、本節では、この要件を満たす動的なブラックホール経路検出である ABIC(Automatically BGP Integrity Check) の概要について、その他の既存の手法と比較しながら、その有用性について検討する。

4.2.1 前提条件

まず、ABIC が扱う範囲を前提として定義する。本論文では、以下を前提としておき、それらに関しての議論は取り扱わない。

- データベースにはすべてのデータが正しく格納されている
  4.1.2 小節で述べた、セキュリティの要件がここでは満たされていると仮定し、セキュリティに関しての議論はおこなわない。

4.2.2 ABIC について

上記の前提をもとに、ABIC アーキテクチャーについて議論する。図 4.4 は、正常な場合の ABIC 動作図である。

これは、以下のようにして UPDATE 中のオリジン AS と DB 中のオリジン AS との整合性をチェックする。

1. BGP ルータ (AS2) は、UPDATE メッセージを受信する
2. UPDATE メッセージ中に含まれる AS-PATH(AS4 → AS3) のうち、そのオリジン AS および NLRI(2001:500::/35) をマークする
3. BGP ルータは、マークしたプレフィックスをキー (2001:500:::/35) にしてリモートのデーターベースマネージャーにクエリーを送信する
4. DB は、BGP ルータからのキーをもとに、自身のデータベースを検索する
5. 検索し、そこからオリジン AS(AS5) を抽出し、その値を BGP ルータに返す
6. BGP ルータは、DB から返されたオリジン AS(AS5) と、UPDATE メッセージに含まれるオリジン AS(AS4) とでその比較をする

7. 両者が一致していれば、それは正しい経路とみなす
8. 一致していなければ、ブラックホール経路として、経路表への注入を抑制する

つまり、図4.4の例では、DB中のオリジンASは、AS5であるにもかかわらず、UPDATE中のオリジンASは、AS4である、ABICはこれを検出し、AS4からの経路広告をブラックホール経路をみなし、経路表への注入を抑制する。
図4.5に、図4.4を抽象化したABICのアルゴリズムを示す。

4.3 他の手法との比較

ここでは、4.2節で議論したABICアーキテクチャーについて、3章で紹介した関連研究と、どの点でことなるか比較検討することにする。

BGP Route Flap Dampingとの比較

BGP Route Flap Dampingは、前述のようにフラッピングしている経路にペナルティを与え、閾値を越えると、その経路を抑制する仕組みである。
ABICとの共通点としては、個々のルータで運用可能という点である、つまりすべてのBGPルータにRoute Flap DampingあるいはABICを導入しなくても、個々のルータレベルでの運用ができる。
また、相違点であるが、 Route Flap Dampingは、フラッピングしている経路は検出できるが、それがブラックホール経路であることは検出できない。

BGPコミュニティ属性との比較

ブラックホール経路を検出するために、BGPコミュニティ属性に、オリジンASを設定し、それを各ルータでそれを判断する方法が考えられる。
ABICの共通点としては、BGPコミュニティ属性を用いれば、ブラックホール経路を検出ずることが可能である。しかし、これを実現するためには、すべてのBGPルータに拡張コミュニティを定義し、オリジンASを設定するようにしなければならない。

S-BGP との比較

ABIC と S-BGP との共通点は、ABIC がリモートデータベースから、S-BGP が PKI を利用した CRL から正しい経路を検出することができる点である。また、 Route Flap Damping のように箇々のルータレベルで運用が可能な点も共通している。
ABIC との相違点としては、ABIC では、DB からの問い合わせを cache することによって、UPDATE メッセージのフラッピングにも耐えられるが、S-BGP では、その点に関しては考慮されていない点で異なる。

```
;(1)
$ORIGIN     bgp.in-addr.arpa.
@           SOA     (...)
0           AS      0 0
205         NS      ns0.arin.net.

; arin.net (2)
$ORIGIN     205.bgp.in-addr.arpa.
@           SOA     (...)
            AS      65535 8
0           NS      ns0.digex.net.
1           NS      ns0.verio.net.

; digex.net  (3)
$ORIGIN     0.205.bgp.in-addr.arpa.
@           SOA     (...)
            AS      2548 16

; verio.net  (4)
$ORIGIN     1.205.bgp.in-addr.arpa.
@           SOA     (...)
            AS      2914 16
0           AS      777  24
1           AS      2914 24
```

図 4.2: AS リソースレコードを追加した DNS

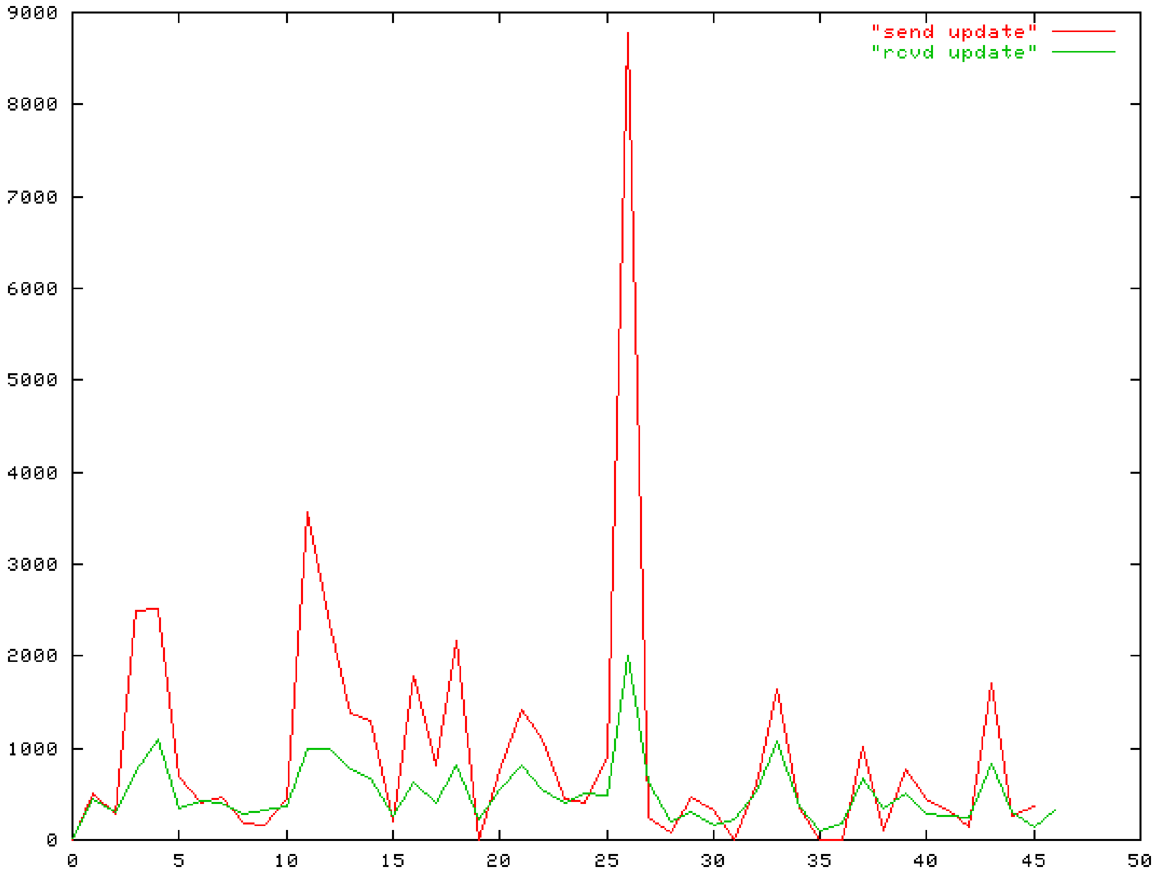
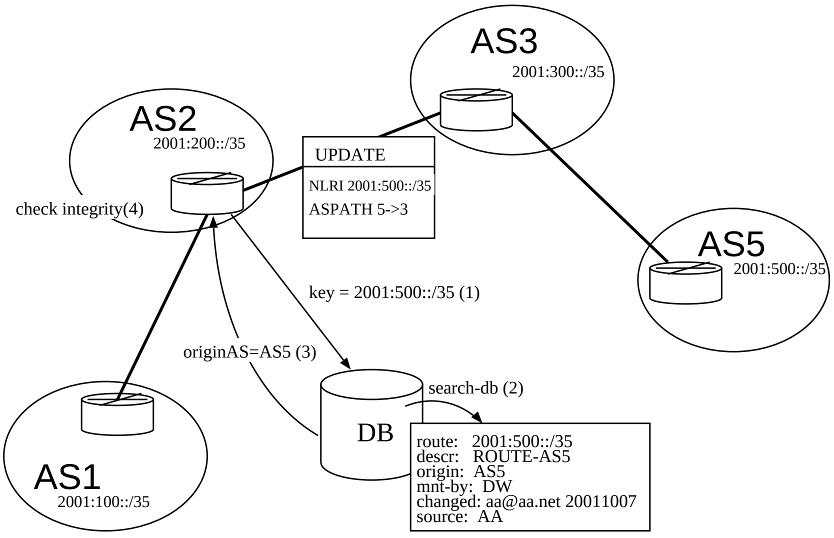


図 4.3: 1 時間ごとの送出・受信 UPDATE 数



route: 2001:500::/35
descr: ROUTE-AS5
origin: AS5
mnt-by: DW
changed: aa@aa.net 20011007
source: AA

図 4.4: ABIC アーキテクチャー

```
while (1)
{
  if (bgp-update-comes)
  {
    oringin_as   = head_of_as_path(update.path_attr.as_path);
    key_prefix   = update.nlri.prefix;

     if ((db_origin_as =  is_cache_exsist (key_prefix)) == NULL)
       {
        db_origin_as = search_registry_db (db_server, key_prefix);
        inject_cache(key_prefix, db_origin_as);
        }

    if (db_origin_as !=  origin_as)
       goto filterd:
  }
}
```

図 4.5: ABIC-アルゴリズム

第5章　ABICの設計・実装

4章では、動的にブラックホール経路を検出する仕組みとしてABICアーキテクチャー(Automaticaly BGP Integrity Check)を提案した。ここでは、このアーキテクチャーについて4章での要件を踏まえつつ、より具体的な設計および実装について述べる。

5.1　設計

4章での、議論に基づき本節では、ABICシステムの設計について述べる。具体的には、BGPルータ/DB間でのメッセージ交換の仕組み、データベースの構造、BGPルータでのキャッシュについて検討する。

5.1.1　BGPルータ/DB間でのメッセージ交換

ABICアーキテクチャーを実現するためには、BGPルータとDB間で、以下のプロトコルが必要となる。

- BGPルータとDBマネージャーが、常に通信することが可能なように定期的に生存確認をする
- BGPルータから、DBマネージャーへプレフィックスをキーにして問い合わせをする
- DBマネージャーは、BGPルータからのクエリーを受け取り、DBを検索する。その検索結果を、BGPルータに返す

これらを実現するために、以下のメッセージを定義し、それぞれのメッセージについて説明する。なお、いづれのメッセージもネットワーク層はIP(バージョンについてはIPv4およびIPv6)で、トランスポート層は、TCP (ポート番号は2107番)を使用して、メッセージ交換をおこなう。

- QUERYメッセージ
- REPLYメッセージ

- KEEPALIVE メッセージ

KEEPALIVE メッセージ

KEEPALIVE メッセージは、BGP ルータと DB マネージャーとの間で、TCP コネクションを保持し続けるために用いるメッセージである。KEEPALIVE メッセージは、QUERY メッセージとは関係なく 30 秒に一度 BGP ルータが送出し、DB マネージャは KEEPALIVE メッセージを受信したら、それを BGP ルータに返送する。
BGP ルータは、DB マネージャーからの KEEPALIVE を受信すると、BGP ルータが保持している KEEPALIVE フラグを立てる。KEEPALIVE フラグが立っていない状態では、BGP ルータと DB マネージャーとの通信路が確立しないとみなし、QUERY メッセージを送信することはできない。

図 5.1 に、KEEPALIVE メッセージのパケットフォーマットを示す。

図 5.1: KEEPALIVE メッセージ

1 オクテットの MSG_TYPE は、どの種類のメッセージかを識別するために定義し、KEEPALIVE メッセージのメッセージタイプの値は、3 である。残りの、3 オクテットは、0 が標準値として埋め込まれており、このフィールドは原則的に参照されない。

QUERY メッセージ

QUERY メッセージは、BGP ルータがピアから UPDATE を受け取ったときに送出されるメッセージである。
図 5.2 に、QUERY メッセージのパケットフォーマットを示している。

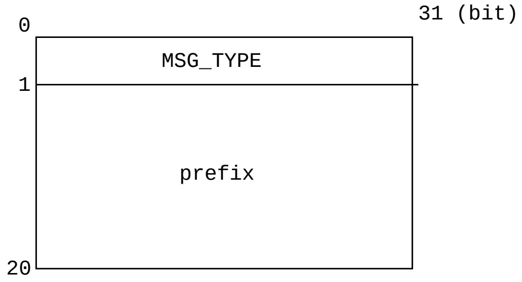


図 5.2: QUERY メッセージ

0 オクテット目の QUERY メッセージの MSG_TYPE 値は、1 である。
1 オクテットから 6 オクテット目に設定される IPv4 プレフィックスは、IPv4 アドレスプレフィックスが設定される。プレフィックスには、IPv4 および IPv6 のいずれかを指定することがでいる。たとえば IPv4 の場合では、203.178.139.51/24 のように IPv4 アドレス中の 32 オクテットのうち、オクテットとオクテットを”.” で区切り、最後にプレフィックス長として、”/” を記述する。
6 オクテットから 24 オクテットまでの IPv6 の場合は、RFC2373[19] に従う。
QUERY メッセージにおいては、IPv4 プレフィックスと IPv6 プレフィックスとが一つのパケットに同時に記述することはできない。また、一つの QUERY メッセージには、一つのプレフィックスのみである。

REPLY メッセージ

最後に、REPLY メッセージについて述べる。REPLY メッセージは、DB マネージャーがデータベースを検索し、キーに完全一致するプレフィックスがあれば、データベースからオリジン AS 情報を抽出し、BGP ルータに返す。
5.3 は、REPLY メッセージのパケットフォーマットについて示している。

0 オクテット目の、REPLY メッセージの MSG_TYPE は、2 である。
1 オクテットから 5 オクテットまでは、DB 検索から得られたオリジン AS 番号が設定される。

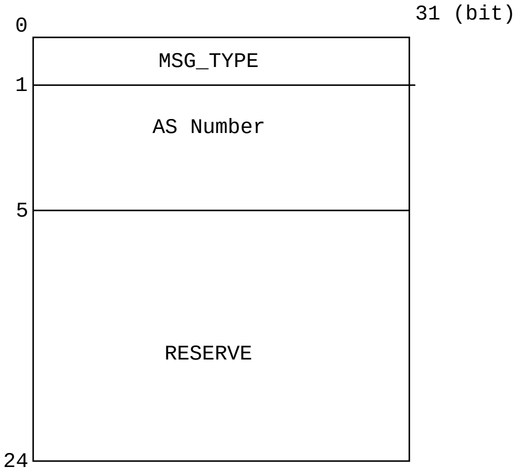


図 5.3: REPLY メッセージ

5.1.2 データベースマネージャーについて

通信プロトコルの設計に続いて、ABIC を実現するために、どのようなデータベースを設計すべきであるか議論する。ABIC では、データベースに関しては、IRR を用いることを 4.1.1 節で検討した。よって、データベースのスキーマなどについては、IRR の記述言語である RPSL[20] を利用する。

検索方法

DB を検索する際に問題となるのが、検索キーとプレフィックス長との関係である。考えられる検索方法としては、以下の 2 通りがある。

完全一致検索

プレフィックスおよびプレフィックス長が完全に一致する route オブジェクトのみを検索結果として格納する方法である。

再帰検索

上記のようにプレフィックス長が完全に一致しなくてもデータベースを再帰的に検索し、最も一致するレコードを検索結果として格納する方法である。

上記にあげた二つの方法であるが、再帰検索の場合、最も一致するオブジェクトが、正しいオリジン AS であるか保証できないという点でその信頼性は低い。よって、検索方法として完全一致による検索を採用する。

5.1.3 BGP ルータでのキャッシュについて

設計において、もう一つ議論しなければならないのが BGP ルータが持つキャッシュである。4.1.3 節で、議論したように、現在の BGP では非常に多くの割合でフラッピングが発生する。そうしたフラッピングに対して、データベースのアクセスを減らすべく BGP ルータでのキャッシュを ABIC の必要要件に加えた。本節では、このキャッシュに関するより具体的な設計について議論する。

キャッシュの動作

4.2.2 小節でも議論したが、ここではより詳しいキャッシュの動作について述べる。以下は、ABIC 全体の振舞いからキャッシュの部分を抽出したものであり、図 5.4 は、その振る舞いを図示したものである。

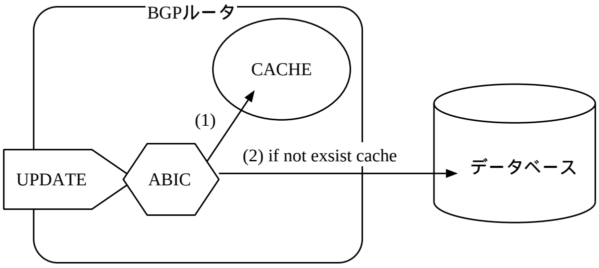


図 5.4: キャッシュの振舞い

1. BGP ルータが UPDATE を受信する
2. BGP ルータは、 UPDATE メッセージに含まれる NLRI を抽出する
3. BGP ルータは、自身のメモリに蓄積されるキャッシュをプレフィックスをキーに検索する
4. キャッシュ中にプレフィックスと完全一致しかつ生存期間が有効な レコードがあれば、そのオリジン AS を返す
5. キャッシュが存在しない場合、データベースマネージャーに QUERY メッセージを送信する

6. データーベースマネージャーからオリジン AS が返されたら、BGP ルータはその整合性をチェックの後、プレフィックスをキーにして、データベースに記載されているオリジン AS および REPLY メッセージを受信した時刻を新たなキャッシュとして挿入する

図 5.5 は、キャッシュの構造である。

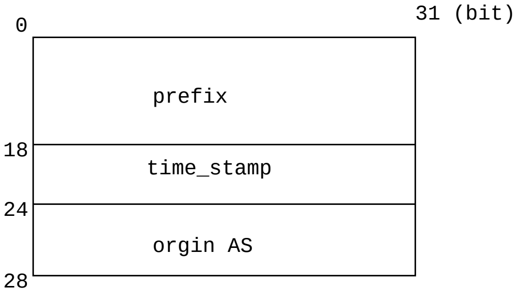


図 5.5: キャッシュの構造

キャッシュの生存期間

次にキャッシュをどのように更新するかについて述べる。一般に、プレフィックスとそのオリジン AS が変わるということはそれほど頻繁におこらない。よって、キャッシュの更新は、数秒に 1 度のオーダーでつねにリフレッシュする必要はない。そこで、本研究ではキャッシュの生存期間を 6 時間と定め、データベースマネージャーからオリジン AS 情報を受信した時刻をキャッシュ中に挿入し、キャッシュを検索したとき、6 時間経過していなければ、そのキャッシュを採用し、6 時間経過していれば、そのキャッシュを無効と判断して、データーベースマネージャーに再度検索する。

5.2 実装

本節では、4 章ならびに 5.1 節において議論した ABIC アーキテクチャーの実装について述べる。

5.2.1 実装環境

ABIC の実装環境を、表 5.1 に示す。

表 5.1: 実装環境

BGP デーモン	zebra-0.92a
IRR	YARD-0.1a[21]

BGP デーモンとして、zebra パッケージを選択した理由としては、現在 GPL ライセンスでソースを公開している数少ないソフトウェアであるとともに、そのパフォーマンスも優れており、幾つかの IX におけるルータとしても運用実績があるので、ABIC を実装する上で最適であるからである。

また、IRR として YARD を選択した理由として、YARD は、RPSL に完全に準拠していないが、データベースの構築あるいはその加工が容易であるという点で、ABIC のデータベースマネージャーを実装するために最適であり、これを採用した。

5.2.2 BGP ルータの実装

本節では、BGP ルータの実装について起動時の処理、UPDATE メッセージを受信したときの処理に分離して ABIC の実装について述べる。

起動時の処理

ABIC の起動は、設定ファイルで、ABIC を有効にする必要がある。図 5.6 は、zebra bgpd 設定ファイルの抜粋である。この例では、データベースマネージャーとして、“rr.sfc.wide.ad.jp” に BGP ルータは、QUERY メッセージを送出する。また、データベースマネージャーは、この場合は DNS に登録してあるホストであるが、IPv4 あるいは IPv6 アドレスを指定することも可能である。

BGP ルータは、設定ファイル中の “bgp registry” ステートメントを読み込むと、bgp_registry_config 構造体を初期化し、この構造体の一つである enabled が 1 になり、ABIC が有効になる。bgp_registry_config 構造体を図 5.7 に示す。

続いて、ABIC はつぎにデータベースマネージャーへ TCP 2107 番ポートへ接続する。接続が成功した場合は、続いて keepalive メッセージの送出に移るが、接続が失敗した場合は、警告メッセージを出力するとともに、30 秒に 1 回づつ再接続を試みる。

keepalive メッセージは、30 秒に一度づつ送出され、データベースマネージャーからの keepalive を受信した時点で、bgp_registry_config 構造体の is_keepalive に 1 を代入する。また、いままで成功していた keepalive が失敗した場合は、is_keepalive に 0 を代入する。

```
router bgp 65443
 bgp registry rr.sfc.wide.ad.jp
 neighbor fe80::200:e2ff:fe34:fddc remote-as 65442
 neighbor fe80::200:e2ff:fe34:fddc interface fxp0
!
 address-family ipv6
 network 2001:200::/35
 network 2001:210::/35
 neighbor fe80::200:e2ff:fe34:fddc activate
 exit-address-family
!
```

図 5.6: zebra-bgpd 設定ファイル例 (抜粋)

UPDATE時の処理

zebraでは、ピアからのパケットを受信すると、それがどの種類のBGPパケットかを判別し、別スレッドでそれを処理する関数を実行する。BGP UPDATEの場合は、bgp_update(sturct peer *, sturct prefix* , struct attr * , afi_t , safi_t, type, sub_type, sturct prefix_rd * , u_char *tag) という関数が実行され、AS-PATHのループチェックや、ポリシーフィルターの適用、nexthopの計算などをおこなうが、この最後にABICの機能を追加した。具体的には、bgp_registry_config構造体のenabledが1であり、かつ同構造体のis_keepaliveが1の場合、以下のABICに関する処理を実行する。

- attr(パス属性)構造体中のaspath構造体から、オリジンASを抽出し、それをbgp_registry_config構造体のas構造体にコピーする。
  オリジンAS抽出方法としては、aspath構造体のデータ部分は、スペース区切りでAS-PATHが格納されているため、データの先頭から、最初のスペースまでがオリジンASとして抽出する。

- 関数rr_send_query (struct bgp_info *, struct prefix *) に、UPDATEに含まれるパス属性情報 (bgp_info構造体) およびNLRI (prefix構造体) を引数として、実行する。

```
struct bgp_registry_config
{
  int sock ;                  /* DBへ接続するためのソケットディスクリプタ */
  int enabled ;               /* ABICが有効であるかどうか */
  struct stream *ibuf;        /* 入力バッファー */
  struct stream_fifo *obuf;   /* 出力バッファー */
  struct stream *host;        /* DBマネージャーのホスト名（FQDN）*/
  struct stream *as;          /* オリジンAS */
  struct stream *rr_as;       /* DBから得たオリジンAS */
  struct hash *cache;         /* キャッシュ */
  struct addrinfo *res;       /* getaddrinfo  */
  u_int32_t is_keepalive;     /* keepalive フラグ */
};
```

図 5.7: bgp_registry_config 構造体

  - rr_send_query 関数は、まずメモリ上にハッシュテーブル形式で格納しているキャッシュを、プレフィックスをキーにして探索する。

  - ハッシュテーブルに該当するエントリーがある場合、rr_as 構造体にキャッシュしてあるオリジン AS が代入され、BGP UPDATE 中のオリジン AS と比較する。

  - キャッシュに該当するエントリーが無い場合、データベースマネージャーに、BGP UPDATE 中のプレフィックス (prefix 構造体) をキーにして問い合わせ、その結果を rr_as 構造体に代入するとともに、キャッシュのハッシュテーブルに代入する。
    なお、データベースマネージャーに問い合わせ、15 秒待っても、返答が無い場合はタイムアウトとみなし、ABIC ルーチンは、キャンセルされる。

- rr_as 構造体の値が NULL でなければ、BGP UPDATE 中のオリジン AS(as 構造体) との比較をする。rr_as 構造体のデータ部分（`rr_as->data`）と as 構造体のデータ部分（`as->data`）を match_orgin_as(struct as *, struct rr_as *) 関数で比較する。

  - 両者の値が完全一致していれば、正しいオリジンAS とみなし、bgp_update 関数に戻り、その他の処理を継続する。

  - 両者の値が一致しない場合は、無効なオリジンAS とみなし、図5.8 のエラーメッセージを出力するとともにそのUPDATE を無効なもととして、フィルターアウトし、そのBPG UPDATE を無効なものとする。

```
[Error] OriginAS = 65441 is incorrect compared with DB OriginAS =  65443
```

図5.8: オリジンAS 不一致時のエラーメッセージ

- rr_as 構造体の値がNULL であれば、ABIC ルーチンはキャンセルされ、次のUPDATE 処理ルーチンに移る。

以上、ABIC アーキテクチャーの具体的な設計・実装について述べたが、次章では、このアーキテクチャーの有効性について評価する。

第6章　評価

本章では、5章で議論したABICアーキテクチャーの実装について、その有効性について評価する。まず、なにをもってしてABICアーキテクチャーが有効であるかという評価方針について議論する。その後、評価する環境について述べ、実際の定量評価をおこなう。

6.1　評価方針

1.1章で述べたように、本研究の目的は、現実のブラックホール問題の解決にある。ABICアーキテクチャーは、この解決を目指すべく提案したものであり、評価としても現実のブラックホール問題を解決できるかどうかが焦点になる。すなわち、現在のドメイン間経路制御の中にABICを導入することによって、ブラックホール問題を解決できることができれば、本研究の目的を達成することができる。つまり、評価としては、現実のドメイン間経路制御の環境を用意し、そこでブラックホール経路生成し、それに対してABICアーキテクチャーがどの程度の適応能力があるかを定量的に評価する。

実際のドメイン間経路制御環境

では、実際のドメイン間経路制御が、どの程度の規模であるか、検証する。ドメイン間経路制御に関して規模を考えなければいけないのは、以下の点である。

- ピアの数
- 処理すべきNLRI(プレフィックス)の数
- 処理すべきフィルターの数
- UPDATEの数

このうちで、ピア、プレフィックス、フィルターは、ABICではその監視対象としていないので考慮する必要はないが、最後のUPDATE数に関しては、ABICはBGP UPDATEごとにその

監視を行うので、その現実の規模に関して考慮する必要がある。
つまり、実際のドメイン間経路制御では、どの程度のUPDATEが生じていて、その環境で評価すればその有効性を示すことができる。実際のUPDATEについては、4.1.3小節で、計測したものを参考にする。計測したUPDATE数(総UPDATE数および1時間あたりの最大UPDATE数)を表6.1に示す。

表6.1: 実際のドメイン間経路制御におけるUPDATE数

UPDATE送出数	44,677
1時間当たりの最大UPDATE送出数	8,785
UPDATE受信数	23,671
1時間当たりの最大UPDATE受信数	2,007

ここでの総UPDATE受信数は、23,671であり、このうち1時間あたり最大で2,007ものUPDATEを受信している。評価指標として、1時間につき2,000程度のブラックホールNLRIを含んだBGP UPDATEを他のピアからABIC対応BGPルータに送出し、ABICがどこまで対応できるかについて評価をする。

6.2 定量評価

6.1節では、ABICアーキテクチャーの評価方針を示したが、本節では、何をもって“対応できる”かについて定義する。
前節の“対応できる”を、本研究の目的に照らし合わせて詳しく分析すると以下の要素であると言える。

ステップ1 2,000ものBGP UPDATEをABICが、損失することなく受け取る

ステップ2 受け取った各UPDATEからプレフィックスとオリジンASを抽出し、キャッシュ中になければデータベースマネージャーに問い合わせる

ステップ3 キャッシュあるいはデータベースマネージャは、ハッシュ・データベースの持つエントリーを検索しその結果を損失なくBGPルータに返す

ステップ4 BGPルータはその結果を受け取り、一致していれば、つぎのBGP処理ステップに進み、一致していなければ、フィルターアウトする

ABICは、上にあげた4つのステップから成立しているのでピアからUPDATEを送出し、どのステップですべてのUPDATEのうち、どれだけパスし、どれだけ損失したのかを評価する。

また、どのような評価環境で、どのようにして評価をするのかについて、6.2.1 小節で触れる。なお、4.1.3 小節では、ルート・フラッピングにより、同じ NLRI による更新・取消しが多いことを指摘し、キャッシュの必要性について述べた。果たして、キャッシュを導入した結果が有効であるかどうかについて、6.2.2 小節では、キャッシュなしの場合、6.2.3 では、キャッシュを導入していない場合の定量評価、次の 6.2.3 小節では、キャッシュを導入した場合の評価について別小節にわけ、その有効性について議論する。

6.2.1　評価環境

ここでは実際の評価環境およびより具体的な評価方法についてのべる。
評価環境としては、図 6.1 のように、BGP ルータ、ABIC 対応 BGP ルータ、IRR(データベースマネージャー) が、一つのファーストイーサネットセグメントに接続されている。

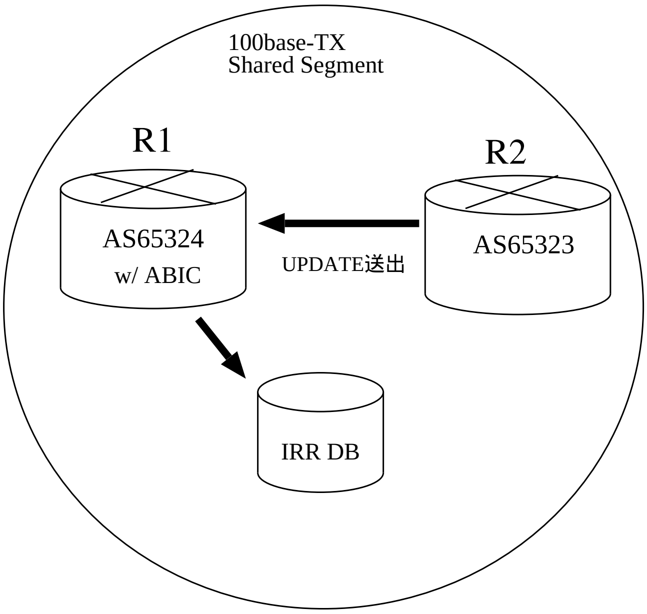


図 6.1: 評価環境のトポロジー

また、評価にあたっては 3 台の同一スペックの PC を用いた、その PC のスペックについて表 6.2 に示す。

次に、R2 から送出する UPDATE について述べる。UPDATE は、データベース中のオリジン AS とは異なる、すなわちブラックホール NLRI プレフィックスを R2 で生成し、送出する。送出する間隔として、10 秒に 1 回 (通常時の UPDATE 間隔) および 1 秒に 1 回 (ルート・フラッピン

表 6.2: 評価マシンのスペック

CPU	Intel Pentium III 800MHz
メモリ	256Mバイト SDRAM
NIC	Intel EtherExpress Pro/100B

グ時のUPDATE間隔)の両方を、それぞれ送出する。R2からのUPDATE送出方法であるが、スクリプトで1秒に1回および10秒に1回、カーネルの経路表に静的に任意の経路を注入・削除する、次にBGPは、zebraデーモンからその注入・削除メッセージを受信し、それをUPDATEとして、ABIC対応BGPルータに送信する。

6.2.2 キャッシュを導入しない場合での定量評価

まず、キャッシュを導入しない場合での定量評価について述べる。キャッシュを導入しない場合とは、DBマネージャーへの問い合わせの前にキャッシュを探索する処理およびDBマネージャーからの結果をキャッシュに加える処理を省いた場合をさす。図6.2に具体的なアルゴリズムを示す。

```
while (1)
{
  if (bgp-update-comes)
  {
    oringin_as   = head_of_as_path(update.path_attr.as_path);
    key_prefix   = update.nlri.prefix;

    db_origin_as = search_registry_db (db_server, key_prefix);
    if (db_origin_as !=  origin_as)
       goto filterd:
  }
}
```

図 6.2: キャッシュを導入しない場合のABICアルゴリズム

この場合についての、ブラックホールNLRIを10秒に一度づつ送信し、それを1時間測定した結果を表6.3に示す。

UPDATEの間隔が10秒あると、BGPルータおよびデータベースマネージャーの負荷、BGP

表 6.3: キャッシュなしでの測定結果 (1)

段階	総 UPDATE 数	通過 UPDATE 数	ロス率
ステップ 1	360	360	0%
ステップ 2	360	360	0%
ステップ 3	360	360	0%
ステップ 4	360	360	0%

ルータとデータベースマネージャー間でのラウンドトリップタイムも結果に影響を与えず、すべて検出することが可能である。

次に、同様に 1 種類のブラックホール NLRI を 1 秒に一度づつ送信し、1 時間測定した結果を表 6.4 に示す。

表 6.4: キャッシュなしでの測定結果 (2)

段階	総 UPDATE 数	通過 UPDATE 数	ロス率
ステップ 1	3600	3600	0%
ステップ 2	3600	3600	0%
ステップ 3	3600	3542	2%
ステップ 4	3532	3183	12%

UPDATE 数を増やすと、ステップ 3 の DB マネージャーによる QUERY 受信およびステップ 4 の DB マネージャーからの REPLY 受信においてロスが生じていることがわかる。

考察

以上にわたり、キャッシュを導入しない場合の測定結果について述べた。ここからわかることはキャッシュを導入しない場合でも、ルート・フラッピングが発生していない状況では、いずれの場合でもロスは生じていない、しかし、ルート・フラッピングに近い環境では、注入する経路の数に関係なくステップ 3 のデータベースマネージャーが BGP ルータから受け取るクエリーを受け取るステップ 3 および UPDATE 中のオリジン AS と DB マネージャーから受け取ったオリジン AS を比較するステップ 4 において、最大で 17%のロスがあることがわかる。この原因とし

て考えることができるのは、データベースマネージャの処理にある。データベースマネージャーの実装では、1つのクエリーを処理するまでの処理において、同時アクセスをさけるためロックをかけているので、あるタイミングで、BGPルータからのQUERYをロック中として、ドロップしてしまったと考えることができる。ただ、これはデータベースマネージャーの実装の問題であるものの、実際のインターネットでは、BGPルータとデータベースマネージャとが同一セグメントで接続されていることはありえなく、BGPルータとデータベースマネージャ間でのラウンドトリップタイムが大きいため、データベースマネージャーからのREPLYが帰ってくる前に、BGPルータが同じプレフィックスで前回とは違うオリジンASでのUPDATEを受け取って結果としてブラックホール経路を検出できなくなるというシナリオも考えられる。

6.2.3 キャッシュを導入した場合での定量評価

次に、ABICにキャッシュを導入した場合での定量評価について検討する。
評価方法は、6.2.1小節で述べた方法と同じであり、1種類の経路を10秒間に1回および1秒間に1回づつピアからUPDATEを送出する。
ただし、キャッシュを導入しない場合では、ステップ3は、データベースマネージャーが受け取ったQUERYの数であったが、ここではキャッシュにアクセスした回数を指す。

1種類のプレフィックスをUPDATEした場合について、10秒間に1回での場合を表6.5に示す。

表6.5: キャッシュありでの測定結果(1)

段階	総UPDATE数	通過UPDATE数	ロス率
ステップ1	360	360	0%
ステップ2	360	360	0%
ステップ3	360	360	0%
ステップ4	360	360	0%

同様に、1秒間の1回、R2からUPDATEを送出した場合についての計測結果を表6.6に示す。

表 6.6: キャッシュなしでの測定結果 (2)

段階	総 UPDATE 数	通過 UPDATE 数	ロス率
ステップ 1	3600	3600	0%
ステップ 2	3600	3600	0%
ステップ 3	3600	3600	0%
ステップ 4	3600	3600	0%

考察

以上の測定結果であるが、1 秒に 1 回問い合わせる場合に限っては、その差は歴然としている。つまり 1 回だけデータベースマネージャーに問い合わせるキャッシュありの場合と、3,600 回のデータベースに問い合わせるキャッシュなしの場合の差であり、ルート・フラッピングが発生するような環境であれば、キャッシュは非常に有効であるということができる。

第7章　結論

7章では、本研究での成果を総括し、その後、今後取り組むべき研究課題について述べる。

7.1　本研究のまとめ

本論文では、2章においてドメイン間経路制御の問題点として、幾つか存在するが、そのなかでブラックホール問題は現在深刻な問題となっており、なぜブラックホール問題が発生する原因について(1)マルチホーム、(2)誤った設定の2つを指摘した。
3章では、こうしたブラックホール問題を解決する関連研究について、(1)BGP Route Flap Damping、(2)BGP コミュニティ属性 、(3)S-BGP の3種類をあげ、その概要およびどの様にしてブラックホール問題を解決できるかについて述べた。
4章では、動的にブラックホール経路を検出するためには、(1)本来、広告すべき AS と(2)誤って広告している AS の両方をマークし、比較することが必要であり、それを実現するための要件として、

1. オリジン AS の整合性
2. セキュリティ
3. 規模性

が必要であると指摘した。
(1)に関しては、整合性をとるためのデータベースとして IRR と DNS をあげ、それぞれの長所・短所を検討し、現在多くの機関で利用されている IRR をオリジン AS の整合性をとるデータベースとして採用することにした。
(2)セキュリティでは、その情報を改竄されず、正しい情報が格納されていることが必要であると指摘した。
(3)規模性に関しては、どの程度の規模性が必要であるか、実際の BGP 環境で測定をし、1時間に UPDATE の送出が8,700回、UPDATE の受信が2,100回あり、それに適応できれば、実際のシステムに適応できることを指摘した。これらの要件をふまえて、4.2節では、動的にブラック

ホール経路を検出することができる ABIC(Automatically BGP Integrity Check) の概要を述べた。ABIC は、BGP ルータが UPDATE を受信すると、そのパス属性中の AS パスからオリジン AS と、NLRI フィールドからプレフィックスをマークする。つぎに、BGP ルータは、プレフィックスをキーにして、データベースに問い合わせ、データベースにそのキーが存在したら、そこからオリジン AS を抽出し、BGP ルータに渡す。BGP ルータは、そのオリジン AS と UPDATE 中のオリジン AS とを比較し、一致していれば正しい経路とみなし、一致していなければブラックホール経路として経路表への注入を抑制することによって動的に検出が可能になることを述べた。つぎに、3 章で指摘した関連研究と ABIC とが、どこが異なるかについて述べた。

5 章では、4 章で提唱した ABIC のより詳細な設計および実装について述べた。ABIC は、BGP ルータとデータベースマネージャーとのメッセージ交換のため、(1)QUERY (BGP ルータ→ DB)、(2)REPLY (DB → BGP ルータ) 、(3)KEEPALIVE (BGP ルータ← → DB) を定義した。

また、BGP においては、その UPDATE メッセージが非常に頻繁におきることを考慮して、一度データベースマネージャーに問い合わせたら、その結果を BGP ルータに蓄積するキャッシュについて述べた。

上記の設計をふまえて、5.2 節では、ABIC の実装について、起動時の処理および UPDATE 時の処理を中心に述べた。

最後に、6 章において実装した ABIC の評価をおこなった。評価の方針としては、ABIC が現実のドメイン間経路制御環境において、どの程度、適応能力ががあるのかを定量的に評価することを目的とした。

そこで、実際のドメイン間経路制御で起こる UPDATE をピアから送出し、どの程度 UPDATE がロスするか、4 つのプロセスにわけて、キャッシュを導入した場合とキャッシュを導入しない場合にわけて、分析した。

キャッシュを導入しない場合においては、10 秒に 1 度 UPDATE をピアから送出した計測では、ロス率は 0%であったが、1 秒に 1 度 UPDATE をピアから送出する場合では、最大で 13%のロスが発生した。一方、キャッシュを導入した場合では、いずれの計測もロス率は 0%であり、ルート・フラッピングのような頻繁に UPDATE が発生する場合に関してはキャッシュが有効であることを示した。

7.2 今後の研究課題

今後の研究課題として大きく分けて(1)ABICの発展・改良、(2)ドメイン内経路制御への応用の2点があると考えられる。それぞれについてその概要、考えられる課題について述べる。

前者のABICの発展・改良とは、本論文では、オリジンASが一致していたら正しい経路、一致していなければブラックホール経路としたが、BGP Dampingで用いるペナルティのような概念を導入することが考えられる。ただ、その導入にあたっては、どのような基準でどのようなペナルティを与えるかなどについて詳しく議論する必要がある。

後者のドメイン内経路制御への応用とは、ABICの流れている経路と正しい経路の比較をドメイン内経路制御に応用することである。実際、ドメイン内経路制御でもあるルータが誤った経路を注入してしまうことがある。それを防ぐために、正しい経路とどのルータがその経路を広告すべきかを管理するデータベースを用意し、各ルータが、経路が更新されたタイミングでそのデータベースにアクセスすることが考えられる。この際、ドメイン間経路制御では、オリジンASとして、だれがその経路を広告すべきであるかを抽象化することが可能であったが、ドメイン内の場合、どのようにこれを抽象化するか、そしてOSPF/IS-ISのような既存のドメイン内経路制御プロトコルとどのように協調動作させるかについて議論する必要がある。

謝辞

本研究を進めるにあたり貴重なアドバイスを頂きました慶應義塾大学環境情報学部教授 村井純博士および同大学 助教授中村修博士に感謝いたします。

また、ブレーンストーミングの段階から多忙にも関わらず、多くの相談に乗って頂いた東京大学大学院情報理工学研究科助教授 江崎浩博士、ならびに、実運用している立場から数多くの有益なアドバイスをしてくださった東京大学情報基盤センター助手 加藤朗氏に感謝いたします。

最後に、こうした研究の場を経済的にバックアップしていただいた私の両親に感謝いたします。

参考文献

[1] G. Malkin. *RIP Version 2.* RFC 2453, November 1998.

[2] J. Moy. *OSPF Version 2.* RFC 2328, April 1998.

[3] D. Oran. *OSI IS-IS Intra-domain Routing Protocol* . RFC 1142, Feburuary 1990.

[4] Fuller K. Varadhan. *Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy.* RFC 1519, September 1993.

[5] Y. Rekhter T. Li. *A Border Gateway Protocol 4 (BGP-4).* RFC 1771, Match 1995.

[6] D. Katz Y. Rekhter T. Bates R. Chandra. *Multiprotocol Extensions for BGP-4.* RFC 2283, February 1998.

[7] J. Yu. *Scalable Routing Design Principles.* RFC 2791, July 2000.

[8] R. Govindan and A. Reddy. *An Analysis of Inter Domain Topology and Route Stablity* . in Proceedings of the IEEE INFOCOM, April 1997.

[9] G.Labovits G. Robert and Farnam Jahanian. *Internet Routing Instability* . IEEE/ACM Transactions on Networking , August 1997.

[10] O. Vohra E. Chen. *BGP support for four-octet AS number space.* draft-ietf-idr-as4bytes work in progress, September 2001.

[11] Xiaoliang Zhao Dan Pei Lan Wang Dan Massey Allison Mankin S. Felix Wu Lixia Zhang. *An Analysis of BGP Multiple Origin AS (MOAS) Conflicts* . ACM SIGCOMM Internet Measurement Workshop , August 2001.

[12] L. Gao J. *Stable Internet Routing Without Global Coordination.* ACM Sigcomm2000, August 2000.

[13] C. Villamizar R. Chandra R. Govindan. *BGP Route Flap Damping.* RFC 2439, November 1998.

[14] R. Chandra P. Traina. *BGP Communities Attribute.* RFC 1997, August 1997.

[15] S. Kent C. Lynn J. Mikkelson K. Seo. *Secure Border Gateway Protocol (S-BGP) - Real World Performance and Deployment Issues* . , August 1998.

[16] P. Mockapetris. DOMAIN NAMES - CONCEPTS AND FACILITIES. RFC 1034, November 1987.

[17] P. Mockapetris. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, November 1987.

[18] Tony Bates Randy Bush Tony Li Yakov Rekhter. *DNS-based NLRI origin AS verification in BGP.* draft-bates-bgp4-nlri-orig-verif work in progress, July 1998.

[19] R. Hinden S. Deering. *IP Version 6 Addressing Architecture.* RFC 2373, July 1998.

[20] C. Alaettinoglu T. Bates E. Gerich D. Karrenberg D. Meyer. *Routing Policy Specification Language (RPSL).* RFC 2280, January 1998.

[21] 長橋 賢吾. 階層的なレジストリ情報の登録 ・更新に関する研究. 2000 年度徳田村井楠本中村研究会卒業論文, 2000.