BGP4 - プロトコルの概要と運用

浅羽登志也((株）インターネットイニシアティブ)

1998 年 12 月 16 日

InternetWeek 98　国立京都国際会館

(社)日本ネットワークインフォメーションセンター編

この著作物は、Internet Week98 における 浅羽登志也氏の講演をもとに当センターが編集を行った文書である。この文書の著作権は、浅羽登志也氏および当センターに帰属しており、当センターの書面による同意なく、この著作物を私的利用の範囲を超えて複製・使用することを禁止します。

c　1998　Toshiya Asaba , Japan Network Information Center

1. 概要
2. BGP 導入の背景
2.1. インターネット全体の構造
2.2. ISP とは
2.3. IX - インターネットエクスチェンジ
2.4. 経路制御とは
2.5. スケーラビリティの問題
3. BGP4 運用技術の基本
3.1. BGP4 の概要
3.2. EBGP と IBGP
3.3. BGP を用いた AS 間経路制御
3.4. AS 外部と AS 内部の階層的経路制御
3.5. パス属性
4. ポリシールーティング
4.1. ポリシールーティングとは？
4.2. ポリシーの例
4.3. ポリシールーティングの実装
4.4. 通過ポリシー
4.5. 複数経路の選択
4.6. マルチホーム下でのロードバランス
4.7. ルーティングレジストリとルートサーバ
5. BGP の運用上の問題
5.1. IBGP フルメッシュ
5.2. Route Flapping と Dampening
5.3. ポリシーの不整合
5.4. 不正な経路情報
5.5. プレフィクス・ベース・フィルタリング
5.6. 経路情報のセキュリティー
6. まとめ
7. 付録: Cisco でのサンプルコンフィギュレーション

1. 概要

ISP 間経路制御の代表的なプロトコルである BGP4 について、その開発の経緯と運用技術の基本について説明し、さらに BGP4 を使用する主な目的であるポリシールーティングと BGP4 運用上の問題について説明します。

2. BGP 導入の背景

2.1. インターネット全体の構造

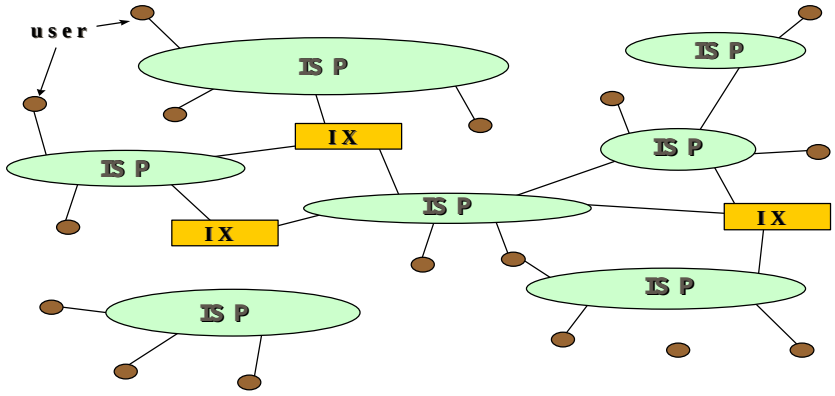


図 2.1.:インターネット

全体の構造現在のインターネットは、インターネットエクスチェンジと呼ばれるを IX を経由、またはダイレクトに ISP 間を接続して、一つの大きなネットワークを構成しています。ユーザはいずれかの ISP を選択して接続することでインターネットに接続します。このようなインターネット全体の構造が今日の姿です。

2.2. ISP とは

・インターネットへのコネクティビティーを提供
・複数の ISP 同士が相互接続して全体を構成
　　“インターネット”
　　相互接続形態
　　IX 経由の接続
　　直接接続
・ユーザはいずれかの ISP 経由でコネクティビティーを得る

今や常識になってしまいましたが、インターネットへのコネクティビティーを提供するのが ISP と呼ばれるもので、Internet Service Provider という言葉の略です。複数の ISP が相互接続して構成されているものがインターネットです。そもそもインターネットの語源

は、ネットワーク同士が相互接続して、という意味ですので、これは言葉の定義どおりになります。相互接続形態としましては、IX を経由して接続したり、またダイレクトにリンクを持ったりといういろいろな形態があります。ユーザはいずれかの ISP を選択してコネクティビティーを得るというわけです。

2.3. IX - インターネットエクスチェンジ

・複数 ISP 間の相互接続を提供するサービス

IX（Internet eXchange)

ISP 同士がトラフィックを交換する場

イーサネット、FDDI、ATM などのマルチアクセス型のデータリンク接続

同じデータリンクメディアを経由して複数 ISP と接続が可能

例

Network Access Point (NAP)

Metropolitan Area Exchange (MAE)

LINX, NSPIXP,JPIX, MEX,HKIX, etc.

このインターネットエクスチェンジというのは何かといいますとこれも今や当たり前のように言われている言葉ですが、４年ぐらいの歴史しか持っていないものです。これは１本１本の線で ISP 間を相互接続すると、ISP が増えれば接続線も増えていき非効率であるため、１ヶ所で何らかの LAN の技術を用いて相互接続性を提供しようというものです。
例えばアメリカでは Network Access Point(NAP)というのが全米に何ヶ所かありますし、今や MCIWORLDCOM になってしまいましたが、もともと MFS というアメリカの新興の電話会社が始めた Metropolitan Area Exchange というサービス、これも機能としては NAP と同様です。しかも MAE West というのは NAP の一部でもありますので、この２つは概念的には全く一緒のもので、その運営の主体がどこにあるかで違うだけです。
上のほうは NSF(アメリカの全米科学財団)が費用を出しており、昔の NSF ネットバックボーンからそのようなネットワークポイントを経由した商用 ISP の接続系へ移行するという目的でつくられたものです。MAE というのは、時期を同じくしてそれを商用で始めてしまったというものです。またアメリカ以外でも LINX（London Internet Exchange）というのがロンドンにありますし、日本でも、NSPIXP ――これがワイドプロジェクトをやっているもので、あと JPIX とか MEX（Media Exchange）など、商用のインターネットエクスチェンジを提供する会社もあります。アジアでも HKIX（Hong Kong IX）や、フィリピンでは PHIX など、各国ごとに１つもしくは２つぐらいの大きなインターネットエクスチェンジが存在しています。

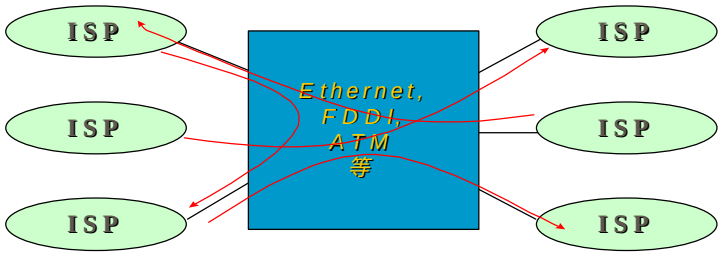


図 2.3.2. IX の概念図

IX の概念としてはこのような形で論理的に１つの大きな LAN のメディアを用意して、ISP をそこにつなぎ込んで、その上で ISP 間のトラフィックの交換を行うというものが IX です。

2.4. 経路制御とは

・インターネットに接続された任意の 2 ユーザ間の、ネットワーク層での接続性の確立
　　・アドレッシング
　　・経路情報の交換
・インターネット上のトラフィック制御
　　・ロードバランス
　　・代替経路の選択
　　・ボトルネックの解消

このようにシンプルなモデルになっていますが、ISP がいくつもあり、その間にリンクがあり、さらに IX を経由した相互接続がある場合、そこでどのようにデータグラムをルーティングするか、それが経路制御と呼ばれる技術によって実現されているものです。ユーザは ISP 経由でインターネットへの接続性を確保するわけですが、任意の２つのユーザを何らかの形で相互に通信できるようにすることが、全体機能としての大きな使命になるわけです。そのために、ユーザに対してどのようなアドレスをつけて、複数の ISP 間で経路情報をどのように交換すればいいかということを、ISP 間の取り決めのもとで制御しながら、インターネット全体でルーティングシステムが動いているということがいえます。
経路制御のもう一つの大きな機能としては、経路情報のやり取りという観点以外にも、結果的にどのようなトラフィックが生まれるかということまで考えなければいけません。経路情報を送るとトラフィックが反対向きに流れてくるというのがインターネットの特徴ですが、往路と復路で違う経路を通ったりすることがありますので、トラフィックがどう流れるかということを忘れがちになってしまいます。

経路情報をこうすればいいと思っていても、実際に動かしてみると思わぬところにトラフィックが集中したり、複数リンクがあっても負荷がアンバランスになってしまうという状態に陥ることがあります。そのような意味でインターネットのルーティングには非常に難しい部分があり、勘と経験に頼る部分がまだまだ大きいといえます。

経路制御の階層

・2 階層の経路制御
    ・ISP の内部、ISP 間
・Interior Gateway (or Routing) Protocol (IGP)
    ・コストに基づく経路選択
    ・OSPF, RIP2
・Exterior Gateway (or Routing) Protocol (EGP)
    ・ポリシーに基づく経路選択
    ・BGP4

次に、経路制御は全体としてどのように行われているのかということですが、非常に大雑把ですが、インターネット全体から見ると経路制御というのは現在２階層で行われています。まず、複数の ISP が接続されたものを一つの固まりとして見た ISP 間で、どういう経路制御をするかという一つの観点。さらに、固まりとして見た ISP の内部に実際はルータやリンクがたくさんあったりするわけですので、その中でどのようにルーティングするかという ISP 内部の問題、そういう２つの階層です。
ISP 間の経路制御をおこなうのがいわゆる Exterior Gateway Protocol と呼ばれているプロトコルで、その代表が BGP4 です。特徴としては、ISP 間経路制御は ISP のポリシーに基づいた経路制御を行わなければいけないと言われています。それに対して ISP 内部のルーティングは、代表的な例としては OSPF とか RIP のバージョン２などを使っています。こちらは例えばリンクのメトリックとかホストなどを用いて行う経路制御になります。

2.5. スケーラビリティの問題

・2 つの問題
    ・アドレス空間の枯渇
    ・経路表（ルーティングテーブル）の爆発
・短期的解決策
    ・CIDR（Class-less Inter-Domain Routing）の推進
    ・プライベートアドレスの活用 (RFC1918)
・長期的解決策
    ・IPv6 (RFC1883)

・アドレス空間の拡張（３２ビット→１２８ビット）
・階層的なアドレス割当と経路制御の推進

こういう形でインターネットの経路制御を２階層で行っていたわけですが、この５年から７、８年ぐらいの間にいくつか問題が発生してきています。一つはアドレス空間の枯渇問題です。ルーティングとアドレッシングは非常に関わりのある問題で、インターネットの規模がまだ小さいときにはルーティングの効率性に基づいてアドレスの割り当てをしていました。しかし、IP バージョン４のアドレス空間は 32 ビット固定長ですのでいつかはなくなるわけで、そのいつかはなくなるという問題がこの７、８年前ぐらいから非常に問題視されるようになってきました。
もう一つの問題は、インターネット上のルータの持っているルーティングテーブルがどんどん大きくなるということです。当然ですが、インターネットが大きくなってユーザ数も大きくなり割り当てられているアドレスが増えてくれば、各ルータが処理しなければいけないルーティングテーブルの数やルータ間の経路情報の量も増えていくわけです。

この２つの問題を同時に解決するために２つの解決策があります。一つは短期的解決策として CIDR というものが導入されましたし、もう一つは長期的な解決策としてアドレス空間を増やすために IPv6 ができたわけです。こういう流れの中でここ５年から 10 年の間にインターネット上のアドレスの割り当てや経路制御の仕方というものが大きく変わってきています。階層的なアドレスの割り当てと階層的な経路制御というものを推進し、それによって次世代にも対応できるインターネットを維持していこうというのが現状の流れです。

CIDR(Classless Inter-Domain Routing)
・目的
・クラスの概念による弊害の払拭
・IPv4 のアドレススペースの有効利用
・経路表のエントリ数の縮少
・階層的アドレス割当
・ビット境界に促したアドレス割当
・経路情報の集成
・アドレスプレフィックス表記
・202.232.68.0 - 202.232.68.63 = 202.232.68.0/26

CIDR の目的について説明します。昔の v4 のアドレスにはクラスという概念があり、そのためアドレスの割り当てが非常に非効率的で、クラスの概念を取り払ってもう少し効率の

いいアドレス割り当てをしようという目的で導入されたのが CIDR です。これはアドレス空間の有効利用を図り、しかも経路、ルーティングテーブルのエントリの数も減らしていこうというものです。
アドレスの割り当てやルーティングを階層的に行おうというものですが、具体的には、昔はバイトバウンダリであったアドレスの階層を 32 ビットのビットバウンダリで行えるようにしたものです。バイトのバウンダリでは、32 ビットを４つの８ビットに分けて 10 進表記すれば、ネットワークアドレスとホストアドレス部分が区別できましたが、CIDR ではビット境界になるため対応できませんで、アドレスの後に、何ビットをネットワークアドレスとして見ているかという情報(例えば/26)を加えるようになっています。
このアドレスとスラッシュのビット長というものをあわせて、プレフィックスという言い方をしています。これは先頭から何ビットまでを着目してアドレッシングやルーティングをするかという、前のほうだけを見る、という意味です。

Classless な経路制御
・VLSM のサポート
　　・インターフェース / 経路表 / 経路制御プロトコル
・Supernet のサポート
　　・アドレス / 経路情報の集成
・“ Glassful ”なアドレス割当と経路制御の概念の排除
　　・all-0 サブネット, all-1 サブネット等
・Classless な経路情報
　　・ネットマスク長の伝播

アドレスの割り当てにおいてこのようなクラスの概念を払拭した場合、経路制御でもクラスの概念を払拭しないといけないということになります。従来であればクラスの概念によりアドレスのネットワーク部分がわかりましたが、CIDR になるとわからなくなります。このため、先頭の何ビットのプレフィックスを見ればよいという情報を経路制御プロトコルの中に組み込んでやる必要が出てくるわけです。
このために、いくつか対応すべきことがあります。まず先ほど説明した、先頭から何ビットをアドレスのネットワーク部分とするかということをビット境界で可変にできる、VLSM(Valuable Length Subnet Mask)、さらに複数の細かいネットワークアドレスをまとめて大きなネットワークアドレスにするという機能(サブネットの逆の概念なので Supernet と呼びます)をサポートしなければいけません。
また、従来あったクラスに基づいた概念をいくつか配置する必要があります。例えば昔は all-0 や all-1 サブネットは使えませんでしたが、現在はどちらも使ってよいということになっていますので、こういったものがルータやホストで扱えないといけません。それから

経路情報にネットマスク長(スラッシュの後ろに書いた数字の部分)をつけ加えなくてはいけなくなってきます。これらの点については、よほど古いマシンや OS を使っていたりしなければ、CIDR への対応は済んでいると思います。

階層的なアドレス割当

階層的なアドレスの割り当てとはどのようにするのか、ということを簡単に説明します。

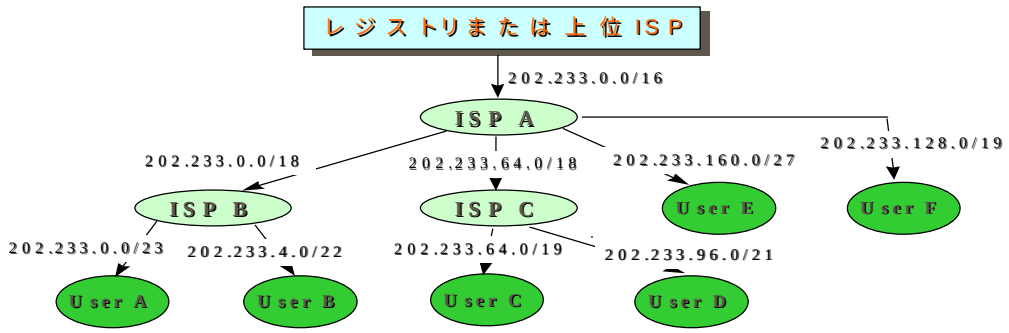


図 2.5.2: 階層的なアドレス割当

図 2.5.2 にレジストリまたは上位 ISP とありますが、これは階層上位にいる ISP は、その下の ISP に対してブロックのアドレスを 1 個割り当て可能ということです。この例では 202.233/16 とありますので、202.233 という先頭 16 ビットを持つアドレスに関しては、ISP A が自分の裁量で割り当ててもよいということを意味しています。その ISP A は/16 をさらに細かく分けて、上からもらったのと同じように自分のカスタマの ISP にブロックでアドレスを渡したりします。ただし、もらったアドレスよりも大きいものは与えられませんから、そのサブセット例えば/18 などを割り当てます。

同じように、また他のカスタマ ISP に/18 を割り当て、あとエンドユーザもいますのでその規模に応じて、/27 で User A に割り当て、/19 で User F に割り当てたりします。エンドユーザにアドレスを割り当てると、実際にそのアドレスを使って接続することになるわけですが、ISP の場合は、アドレスブロックを自分のエンドユーザに対してさらに細かくどんどん割り当てていきます。

202.233.0.0/23 というのはこの/18 のプレフィックスのサブセットになっていますし、同様に 202.233.4.0/22 というのもサブセットになっています。同様に ISP C の場合も自分のもらったブロックの一部を下のユーザに割り当てていきます。このようにして階層的なアドレス割り当てを行います。

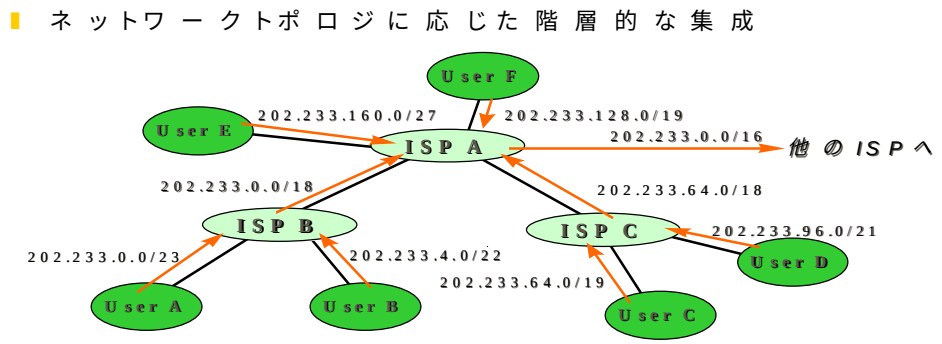


図 2.5.3: 経路情報の集成

ネットワークトポロジに応じた階層的な集成

階層的に割り当てられたアドレスによる経路制御を図 2.5.3 に示します。アドレス割り当てでは上から下にいったのとは逆に、今度は下から上に上がってくるわけです。先ほどとは作画の関係で構成が変わっていますが論理的には同一です。

ISP B から/23 を割り当てられたユーザ A と/22 を割り当てられたユーザ B は、それぞれ ISP B に対して経路情報をアナウンスします。これにより、ISP B とユーザ A、ISP B とユーザ B、もちろんユーザ A とユーザ B 間の接続性が実現されます。ISP B はアナウンスされた２つのプレフィックスを一つにまとめて、/18 の形で上位プロバイダにアナウンスします。ここでアナウンスされるアドレスブロックは、割り当てられたアドレスブロックと１対１に対応している、ということに注意して下さい。このように階層的に割り当てていくと、上位に上ることで経路情報を一つにまとめて見かけ上の数を減らした形でアナウンスできます。

例えば CIDR 対応でない場合２つの経路情報のアナウンスは、そのまま上に上げることになるわけで、数として倍になります。この例では１個が２個になるだけですが、それを世界中やっていると５万が 10 万になって、ルーティングテーブルが爆発することになるわけです。ISP C や ISP A でも同様のことが行われ、最終的に ISP A から６つのエンドユーザに対して１つのアドレスブロックだけをアナウンスするという形になり、全体の経路情報の数を非常に減らしていくことができます。

アドレス利用状況

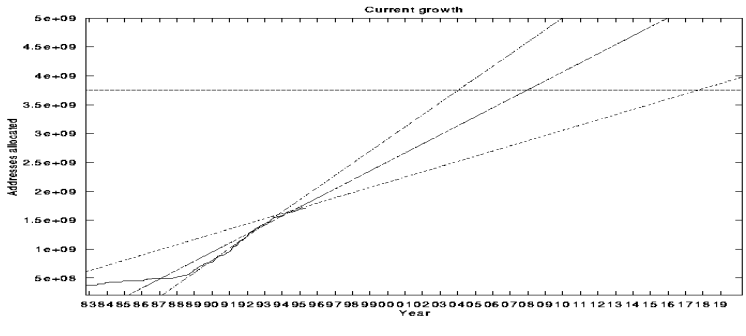


図 2.5.4 アドレス利用状況

図 2.5.4 は、アドレス割り当ての状況を示すグラフです。縦軸が割り当てられたアドレスの数で、昔は徐々に上がっていたものが、いわゆるインターネットブームにより、カーブが急に上を向いたわけです。この傾きを直線で補完すると、大体 2000 年から 2001 年ぐらいのところでアッパーリミットに交わってしまいます。つまり 2001 年ぐらいにはアドレスがなくなってしまうということです。

CIDR が各ベンダのルータで実装され、アドレス割り当ても階層的に行われるように変えられて、消費の傾きが落ちたというのが２つ目の直線です。この場合では 2007 年か 2008 年ぐらいまで IPv4 のアドレスが使えることになります。さらにまた傾きが少し落ちているのは、いわゆるプライベートアドレスの導入で、これによると 2013 年か 2017 年ぐらいまで大丈夫と予測されるわけです。これは IPv4 のアドレスをより長く使えるということですが、重要なのは次のバージョンの IPv6 に移行するまでの時間稼ぎができるということです。昔の状況のままだと今年から来年にかけて全部 v6 に変えなければいけなかったものが、もう少し綿密にインプリメンテーションや仕様の詳細を決めたり、実験をしたりする時間が得られたわけです。

経路表の増大状況

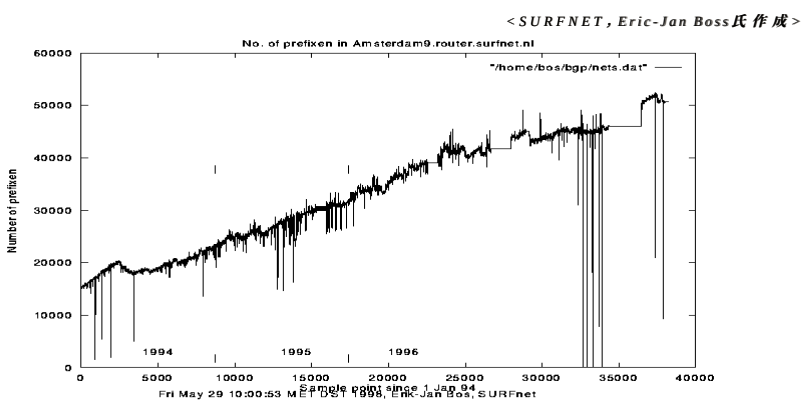


図 2.5.5: 経路表の増大状況

図 2.5.5 は経路表(ルーティングテーブル)の大きさを示すものですが、これも CIDR の効果が良くわかります。横軸はデータを取り始めてからの累積で 1994 年 1 月から始まっています。経路情報が増えてきて、２万経路前後のときに CIDR 化をした効果でテーブルのサイズを減らすことができましたが、1994 年からインターネットが広がって ISP やユーザも増え、実際に落ちてはいますが新しい部分でまた増え始めています。しかし明らかに昔と今で傾きは違うわけです。

これによりルータの方も、経路情報の処理や実際にパケットフォワーディングするときのテーブル・ルックアップなどのオーバーヘッドを抑えることができるようになっています。プライベートアドレスの導入が始まって、さらに傾きは減っていますが、大体５万 2,000 とか５万 3,000 ぐらいの経路数がインターネット全体でやり取りされています。

3. BGP4 運用技術の基本

3.1. BGP4 の概要

ここまでが BGP4 のできてくる背景です。もちろん昔は BGP3 というのがあり、ISP 間で運用されていましたが CIDR には対応していませんでした。それを CIDR に対応させたものが BGP4 です。BGP4 の導入で、初めて CIDR がインターネット全体に使えるようになり、アドレス割り当ての効率化や経路表のサイズ縮小が可能になったわけです。これは大体 1994 年後半から 95 年にかけてですが、一つのエポックメーキングな出来事といえます。

BGP4(Border Gateway Protocol)

- RFC1771
- AS 間経路制御の de-facto 標準プロトコル
  - Autonomous System (AS)
    - 単一の管理主体により、単一経路制御ポリシーのもとで管理・運用される範囲
    - ISP　　AS
    - 現在のインターネットは AS の集合体とみなすことが可能
- CIDR のサポート
    - CIDR の実現に不可欠

その概要を順に説明していきます。

BGP4 は RFC1771 で定義されている AS 間経路制御プロトコルのデファクト・スタンダードで、大きな特徴は CIDR のサポートです。

BGP4 を使う上で重要な概念になってくるのが、AS(Autonomous System)という考え方です。これは、１つの管理主体が１つの経路制御ポリシーで管理している範囲のことで、大体１つの ISP と思っても構いません。もちろん例外はあって、複数の AS を持っている ISP もあります。非常に大きな ISP で世界中にネットワークを持っている場合は、場所によってポリシーを変えたり、全体のルーティングの都合上 AS 番号を分けたほうが効率的な場合がありますので、そのために複数の AS を持ったりしています。例えば UUNET や SPRINTLINK などは複数 AS を持っています。

ISP が AS に対応しており、BGP4 を使うということを考えますと、結局インターネット全体は AS がいくつも集まって、その間で BGP4 を使って経路情報を交換しているネットワークということができます。

特徴

- TCP（ポート 179）を用いる
    - コネクションを張ったルータ間(peer)で 1 対 1 の経路情報の交換
    - 経路情報の交換に信頼性を保証

・RIP 等と異なり、Incremental な情報交換

・16 ビットの AS 番号 (例：IIJ は AS2497)

・Path Vector 方式の経路制御プロトコル

・経路情報に付加されたパス属性（Path Attribute)に基づく経路選択

・ AS Path, Origin, Next Hop, Multi-Exit-Discriminator(MED), Local Preference, etc.

特徴としてはいくつかありますが、まず、信頼性のある通信路上で経路情報を交換するために TCP を用いています。また、２つの BGP スピーカ間で TCP のコネクションを張って、その BGP のコネクションを張ったルータ同士のことを peer という言い方をしますが、その peer 間１対１で経路情報の交換を行います。

RIP では経路情報を、ネットワークの全員にブロードキャストで教えるという乱暴なやり方をとっています。(インターネット上の経路情報は５万経路ぐらいですが、それを例えば 30 秒に１回やり取りするだけで T3 が埋まるといったことになりかねません。) BGP では Incremental な方式、つまり何か変化があったタイミングで経路情報を交換するというやり方をとっています。

AS 番号とは、先ほどの AS を表すためにつけた番号で、現在 BGP では 16 ビットのスペースがとられています。例えば IIJ は AS2497 などと、各 AS に対してユニークに割り当てられています。

Path Vector 方式とは、経路情報に複数のパス属性を付けて、その属性を全部合わせてベストな経路を選択するという方式です。パス属性には例えば AS Path、Origin、Next Hop などがあり、これらが一つの組みになって Path Vector と呼ばれます。

3.2. EBGP と IBGP

・BGP スピーカ (ボーダールータ)

・BGP を用いて経路交換をするルータ等

・EBGP (External BGP)

・異なる AS に属する BGP スピーカ間の BGP セッション

・IBGP (Internal BGP)

・同一 AS 内部の BGP スピーカ間の BGP セッション

・full mesh

・BGP スピーカ間で学んだ経路情報を交換する

・他の IBGP スピーカから学んだ経路は伝播しない

先ほど peer の間で TCP を張って１対１で通信すると説明しましたが、そのときの peer を張る、BGP をしゃべるルータのことを BGP スピーカもしくはボーダールータと呼び、状況に応じて２つの言葉は使い分けられます。
使う場面によって２種類の BGP のセッションが存在します。一つは異なる AS に属する BGP スピーカ同士の BGP のセッションで EBGP と呼ばれます。同じ AS の中でも実際には BGP を使って経路情報の交換をする必要が出てきますので、これは IBGP と呼ばれます。
１つの AS の中でも複数の AS と相互接続している場合には複数のボーダールータがあるわけですが、その複数のボーダールータの間で外とやり取りした経路のパスの情報を交換する必要が出てきます。この場合１つの AS の中にあるすべての BGP スピーカの間でフルメッシュを切るような状態で、IBGP のセッションが張られます。これは各 BGP スピーカが外から学んだ経路情報を、他の BGP スピーカに教えて、全体として情報をシェアする目的で使われています。

ポイントとして、他の IBGP スピーカから学んだ情報は、さらに違う IBGP のスピーカに対しては教えないという大きなルールがあります。最初は私も非常に悩んだのですが、これはなかなか理解しがたい部分です。EBGP で学んだものは当然、IBGP で他の IBGP の peer にまくわけですが、他の IBGP の peer から来た情報をさらに他の IBGP の peer にまくと、おかしなことが起こってしまいます。ですから、こういったことはやらないというのが基本になっています。ただし、それをやろうよという方法も１個提案されて、実際に運用されています。

3.3. BGP を用いた AS 間経路制御

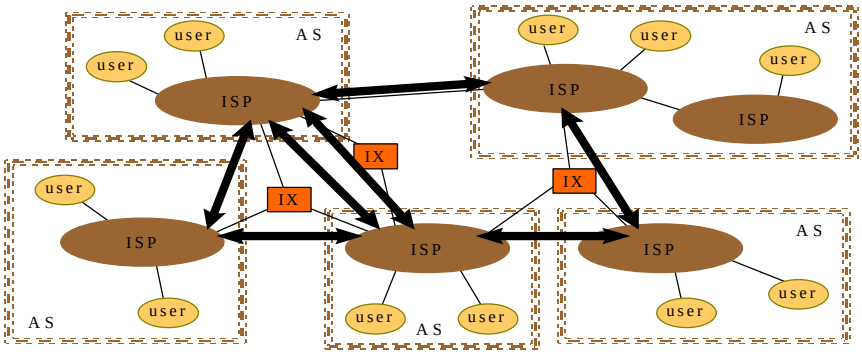


図 3.5: BGP を用いた AS 間経路制御

先ほどインターネットは ISP の集合といいましたが、実は AS の集合であるということになります。また、1 つの ISP が複数の AS を持つ場合もありますが、複数の ISP が 1 つの AS を共有する場合もあります。これまでの説明をまとめたものが図 3.5 で、AS がいくつもいろいろな形で接続されながら、BGP を使って AS 間で経路情報をやり取りしているというイメージが BGP による ISP 間経路制御の全体像です。

3.4. AS 外部と AS 内部の階層的経路制御

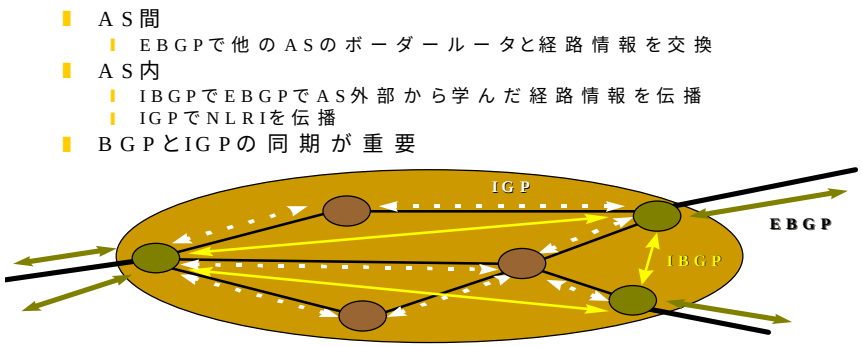


図 3.6: AS 外部と内部の階層的経路制御

AS 外部および AS 内部での BGP のセッションについて説明します。まず、AS の間では EBGP を使って他の AS のボーダールータと経路情報を交換するわけです。例えば図 3.6 では全体が一つの AS で、その内部に BGP スピーカがあります。

AS のボーダーにある BGP スピーカは、他の AS との間に EBGP のセッションを張って経路情報の交換を行います。さらに AS の中では、すべての BGP スピーカの間で IBGP のセッションを張って、外からどういう情報をもらったかということを他の BGP スピーカにも伝えます。IBGP というのはフルメッシュで張る必要があります。これは他の IBGP の peer から聞いた経路情報は、この IBGP の peer には教えないので、フルメッシュでないと全体の間で経路情報のシェアができなくなるからです。

ルータが外から受けた経路情報は IBGP を張っているルータまでは伝わっていますが、BGP スピーカでないルータにはまだ伝わっていないわけです。その場合には AS のポリシーによりますが、EBGP スピーカが OSPL や RIP など何らかの IBGP を使って、外の経路情報を伝えてやる必要があります。この図で NLRI(Network Layer Reachability Information)というのは、要するに経路情報のことです。

ポイントは、こういうやり方の場合には BGP と IGP の同期が必要になるということです。BGP では、あるボーダールータが外から経路情報を受け取って、一つは IBGP でその情報を伝えますが、もう一つは IGP を使ってその情報を中のルータにアナウンスするわけです。ルータ間に複数のルータがかんでいる場合、外から BGP で得られた情報を IBGP で知った時点では、間にそのような経路情報を持っていないルータがあるわけですから、学んだ経路に対する経路というのは確立されていないわけです。

内部のルータは IBGP で学んだのと同じ経路情報を IGP でも受け取るまでは、その経路情報を有効にしません。そうしないと、そのネットワーク相手のパケットを受け取ったときにどちらに投げてよいかわからなくなります。しかし IGP で同じ情報がくれば、コストが

低い方へ投げればよいということがわかるわけです。有効にしないというのは自分のルーティングテーブルにも入れないし、他の AS **の** BGP **スピーカにもアナウンスしないということです。**

3.5. パス属性

・伝播された各経路の属性を示す

　　・複数経路からの経路選択に用いる

　　・ポリシーを表す

・通過型（Transitive)属性と非通過型（Non-Transitive)属性

・必須（Mandatory)属性と任意（Optional)属性

BGP ではいくつかのパス属性の組を用いて経路選択を行いますが、これは、その経路情報がどういう経路でアナウンスされていったかをいろいろな側面から表現するものです。複数経路からの経路選択にはパス属性全体を用いますし、そのパス属性のいくつかは流れてきた経路情報に込められたポリシーを表現しているわけです。

パス属性には通過型と非通過型があり、あと必ず実装しなければいけないものやオプショナルなものもあります。通過型というのは隣からそのパス属性をつけられてきた情報を自分のところで落としたりしてはいけない、というものです。それに対して非通過型は一つの peer の間でしか有効ではなく、その間でのみ重要なので他に出すときには落としてしまうというものです。

Origin 属性

・その経路情報をどこから持ってきたかを表す

・最初に BGP でアナウンスする時に設定される

・必須属性

・可能な値: IGP, EGP, Incomplete

これは、経路情報がどこから生まれてきたかという素性を表すもので、可能な値としては、IGP、EGP そして Incomplete(不明)があります。

例えば、一つの AS 内部でルーティングされている情報を外に出すときには、Origin に IGP がついています。また EGP というのは、他の External Gateway Protocol が生成した経路である、ということを表現しているわけで、他の External Gateway Protocol というのは現在ほとんど使われていませんが、同じ名前で EGP という実際のプロトコルもあったりします。その中で交換されていたものをどこかで BGP に載せかえてきたという場合に、Origin EGP、略して E と表現していますが、そういう値が付きます。

実際には未確認ですが、BGP4 と BGP4 が別のところで動いている場合、片方の BGP4 から Cisco でいう ridistribute のような形を使って持ってくると、Origin EGP が付くのではないかと思います。Incomplete というのは、本当に IGP なのか EGP なのかわからないという場合に使われます。

AS Path 属性

・経路情報が伝播する際に経由した AS の列/組

・ループの検出

・一般的には AS Path の長さが短いほうが選ばれる

　　・ポリシーによる

　　・prepend, stuffing 等の技巧

・必須属性

AS Path 属性には経路情報がどのような AS を経由してきたのかという情報がつけられており、AS 間でのループ検出およびパス選択指標の一つとして使われます。つまり AS Path の長さが短い方が近い、という昔の RIP と同じ方式です。ただし AS Path が短いほうを選べとは RFC には一言も書かれておらず、複数のルータが実装上そうなっているだけです。AS Path が短い方が必ず選択されるというアルゴリズムを実装していないルータでも RFC 違反ではありません。

このような場合、複数の経路で AS Path の長さが同じでも特定のものを選びたいときは、他の AS Path を 1 個増やしてやればよいということで、AS Path prepend または stuffing などという技巧が生まれています。prepend は Cisco の、stuffing は Bay Networks のルータの用語で、同じことをいっています。

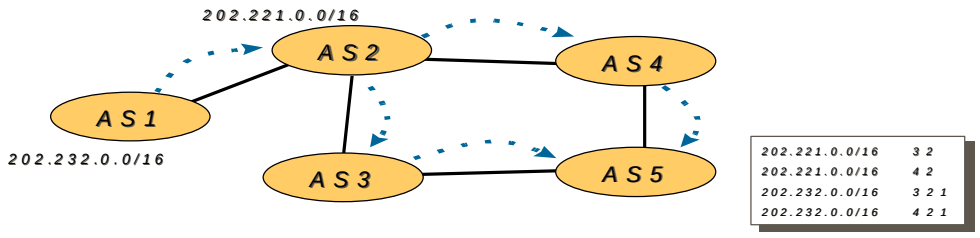


図 3.7.2: AS Path 属性の例

図 3.7.2 のネットワーク構成では、AS1 と AS2 はそれぞれ、202.232.0.0/16 と 202.221.0.0/16 をアナウンス、オリジネートしています。その経路情報が AS5 まで伝わってきたときには、AS5 の経路表は図に示したようなものになっています。

例えば 202.221.0.0/16 では、AS2 からオリジネートされて AS3 から AS5 に伝わってきたので経由している AS としては AS2、AS3 です。これを AS5 から見た順番で書き並べていますので、3 2 という AS Path が一つあります。もう一つの経路は、AS2 から同じプレフ

ィックスがアナウンスされて、AS4、AS5 と伝わってきたので AS Path 属性として、4 2 がついています。同様に 202.232.0.0/16 に関しても、AS2、3、5 と AS2、4、5 という経路がありますので AS Path 属性は２つあります。
このように同じ AS からアナウンスされているプレフィックスでも、複数の経路があるとその分だけ AS Path 属性の違った経路情報が伝わってきます。AS5 では複数の中からどれを選ぶかを決めなくてはなりません。

Next Hop 属性

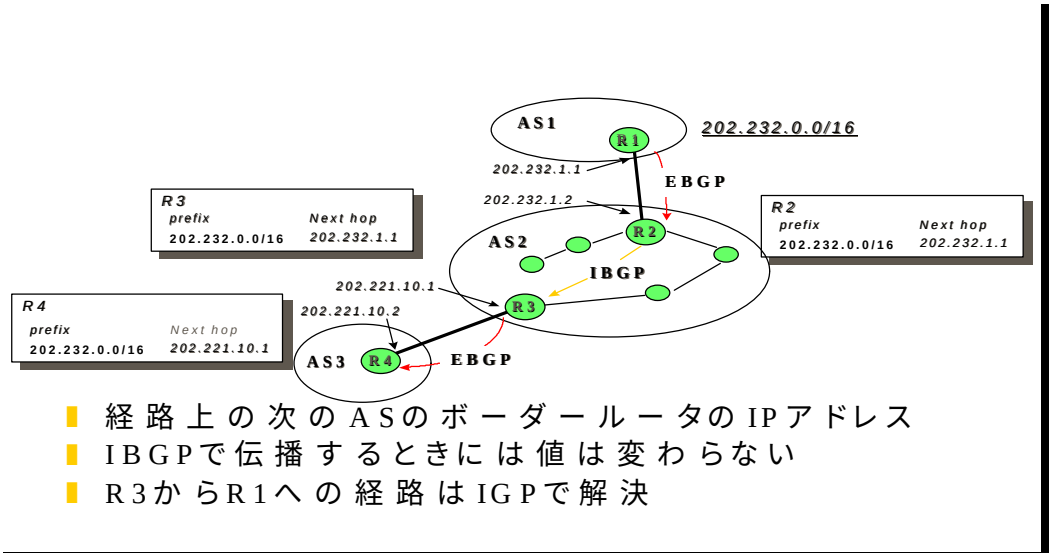


図 3.7.3: Next Hop 属性

経路情報には必ず、Next Hop というものが入りますが、いわゆる IGP、OSP、RIP などでいわれていた Next Hop とは少し感覚が違います。BGP の Next Hop の場合は AS 間の経路制御なので、必ず隣の AS での入り口のアドレスが Next Hop 属性についてきます。
図 3.7.3 は、AS1 から 202.232.0.0/16 が EBGP で伝わって、ルータ R2 から R3 には IBGP で流されて、さらに AS3 のルータ R4 に EBGP で伝わるということを示しています。R1 から R2 に情報が渡ったときに、R2 での Next Hop が R1 のアドレスになるというのは直感的にわかりますが、さらにこれが IBGP で伝わって R3 に伝わったときも、Next Hop は依然変わらず R1 のアドレスがつきます。
隣の AS に行くときには、Next Hop は R3 のアドレスに書きかわって伝わります。このため特に手を加えない限り、１つの AS の中でよその AS から受け取った経路情報に関する Next Hop は、どのルータで見ても同じになります。R3 での Next Hop は 202.232.1.1 になっていますが、これは R3 に直接つながっているルータではありません。この Next Hop のアドレスに対する経路の解決というのは IGP で解決します。すなわちリカーシブに経路表のルックアップを行うことになります。

Multi-Exit Discriminator (MED)

・同一隣接 AS からの複数経路を区別する

・値が小さいほうを優先

　　・IGP のコストを反映させるも可

　　・ロードバランスを考えて設定するも可

・非通過型属性

ここから先は今までのルーティングプロトコル、特に IGP などの OSPF や RIP にはなかったもので、いろいろなポリシーを表すアトリビュートです。一つは MED といわれるもので、同一隣接 AS からの複数経路を区別するためのものです。

MED の値というのは値が小さいほうを優先します。実際にどのように値をつけるかですが、複数リンクで ISP がつながっている状況ではそのリンクのコストをそのまま MED にしてもよいですし、何らかのポリシーで、例えば 100 と 200 というように決めても構いません。複数リンクがある場合はその使い分けをしたくなるわけで、片方のリンクが普段は使わないバックアップならば非常に楽ですが、通常は MED の値を経路情報によってつけかえて投げるということになります。これは、隣接する AS 間だけの複数経路をどう使うかということなので、その先の AS には伝える必要はなく、非通過型の属性になります。

- AS1とAS2の間では、Link1を主に用い、Link2をバックアップとする場合

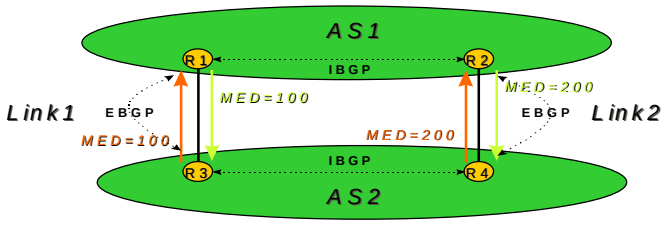


図 3.7.4 は MED が使われる典型的な例です。AS1 と AS2 に対してルータがそれぞれあり、リンクが複数ある場合、２ヶ所で EBGP のセッションを張りますが、この２つのリンクの使い分けを決めるために MED を使います。

例では Link1 がメインで Link2 はバックアップにしています。それぞれアナウンスする経路情報に違う MED の値をつけますが、MED の値は小さいほうが優先されるので Link1 ではお互いに相手に渡すときに 100 とつけ、Link2 はバックアップなので Link1 よりも大きな値で 200 にしています。こうすると AS1 から AS2 に流れるトラフィックに対して、普段は Link1 を通り、Link1 が切れると Link2 に流れる、という制御ができるようになり

ます。

Local Preference 属性
・同一 AS 内部で複数経路の優先度を表すために用いられる
・値が大きいほど優先される
・非通過型属性

これも今までのルーティングプロトコルにはなかったもので、同一 AS 内部での複数経路の優先度を表すために用いられる値です。
ある AS とある AS を結んだときに１個しかパスがないというのは非常に珍しいことで、普通は３つや４つはあります。しかも自分の AS がマルチホームをしていたりしますので、それぞれの AS からもらった経路の中には、同じデスティネーションアドレスが含まれていることになります。
それらの経路のどちらを優先するかは自分のポリシーで決めますが、どちらを優先するかを表すためにこの属性を使います。これは Preference(優先度)ということで値が大きいほうが優先されます。MED と比べると混乱しますが、いわゆるメトリックは小さいほうがいいわけで、Preference は優先度で MED はメトリックと覚えておけばよいと思います。

MED は同じ AS との間に複数のリンクがある場合の制御でしたが、Local Preference というのはもっと一般的に、異なる AS との間に複数リンクがある場合を制御するものです。

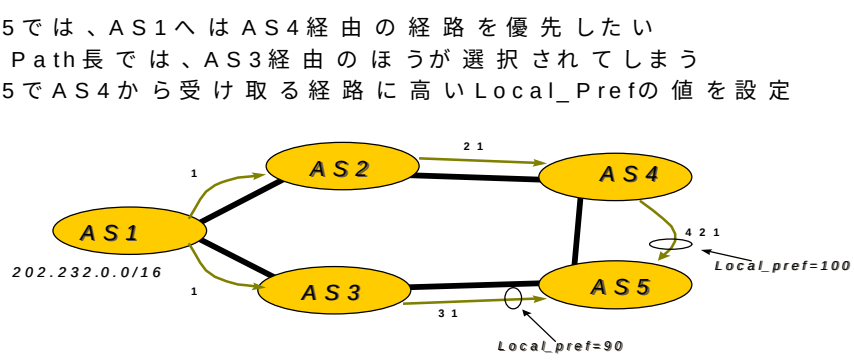


図 3.7.5: Local Preference の例

例えば図 3.7.5 では、AS1 がオリジネートしている 202.232.0.0/16 というネットワークに対する経路情報は複数経路を通って AS5 に伝わっています。AS Path 属性が矢印につけてありますが、AS5 に届いたときに、AS4 経由の場合は AS Path の 4 2 1、つまり AS4、AS2、AS1 という３つの AS があり、また AS3 経由の場合は、3 1、つまり AS3、AS1 ということになります。

AS Path 長としては AS3 経由の方が短くなり、例えば Cisco だとこちらをデフォルトで選択してしまいますが、それだと AS5 のポリシーに合わないということがあります。このような場合は、アナウンスされた経路情報を受け取るときに、Local Preference を違う値に設定して AS Path 長が長いほうを優先するようにします。

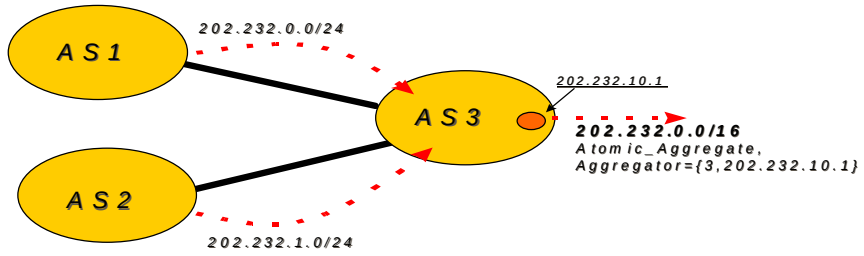
例えば AS4 経由の Local Preference を 100、AS5 経由の Local Preference に 90 という値をつけると AS5 の中では AS4 の経路を優先するという制御ができます。値のつけ方は１つの AS の中で相対的に評価しますので、運用しやすいようなやり方でつけて構いません。AS の運用上のポリシーによりますが、値の余裕をもって 10 飛びでつけるなどということになります。

Atomic Aggregate 属性と Aggregator 属性

- Atomic Aggregate 属性:
- 経路の集成（Aggregate)を行ったときに付加される属性
- 集成の際に細かい経路に付加されていた情報が欠落したことを示す
- 再び細かい経路に分けることはできない

・Aggregator 属性:

・経路の集成を行った BGP スピーカの IP アドレスと、それが属する AS 番号を示す属性



- Atomic Aggregate属性とAggregator属性が設定される

図 3.7.6: 経路の集成

これらの属性は CIDR を使った経路情報の取りまとめを行うもので、英語では Aggregate という言い方をします。

例えば図 3.7.6 では、AS1、2、3 があり、AS3 へは AS1 から 202.232.0.0/24、AS2 から 202.232.1.0/24 という経路があります。AS3 はこの２つの経路情報を一つにまとめて 202.232.0.0/16 という一つの大きなアドレスブロックに対する経路情報を流します。この場合、AS3 はこれらの経路情報をまとめたということを、この経路情報に対して、Atomic Aggregate であり Aggregator は自分であるという情報を込めて、その先の AS に対して伝えていく義務があります。

Atomic Aggregate というのは、Aggregate された経路情報を先の方で分けたりせず、この単位で経路制御することを示すための属性です。Aggregator 属性というのは、経路の取りまとめをだれが行ったかということを、自分の AS 番号と自分のルータのアドレスとを組にして込めてやるもので、そうすると AS が Aggregate したものかが先のほうでわかります。これらの属性がポリシー制御に使われるような場面は実際にはありませんが、プロトコル仕様としては一応このようなものが入っています。

Community 属性

・RFC1997

・経路に色をつける

  ・ポリシーに応じて経路をグループ分けする

  ・一つの経路が複数のグループに属することも可

・32 ビットの整数値

今までの Aggregator までは RFC1771 が出た時点で入っていたものですが、その後、BGP4 の運用から提案されたパス属性がいくつか加えられています。一つはこの Community 属性というもので、RFC1997 になっています。これは昔議論されているときは、ルートカラーなどといわれていたもので、複数の経路で流す経路情報に違う色をつける、といった意味です。これは、単純にいうと経路情報に適当なタグをつけ、そのタグによる制御をあらかじめ遠くの AS に教えておいて、離れた AS との間で何か特殊な制御をできるようにしたものです。

Community 属性の値(共通)

・予約領域

0x00000000 - 0x0000FFFF

0xFFFF0000 - 0xFFFFFFFF

・Well-Known Community:

NO_EXPORT(0xFFFFFF01): AS 外部に出さない

NO_ADVERTISE(0xFFFFF02): 他のルータに出さない

NO_EXPORT_SUBCONFED(0xFFFFFF03): 同盟中の他メンバーAS に出さない

Community 属性は 32 ビット整数値で、いくつかの予約領域があります。また Well-Known Community というのは、すべての AS で共通と思われるものを標準化したものです。経路情報に NO_EXPORT の Community をつけておけば、一つの ISP の中でユーザから受け取る経路情報を自分の AS から先には出したくない場合、ボーダールータが Community の値により落すことができます。受け取ったところ１ヶ所でつけてしまえばよいのですが、Community を使わないでやろうとすると、すべてのルータに対して外部への経路情報アナウンスのフィルタを書くことが必要になり非常に面倒です。
NO_ADVERTISE は BGP で受け取る経路情報を IBGP の peer にも出さずに、ともかく自分の中だけで留めて置くというものです。NO_EXPORT_SUBCONFED は、このあとに説明する AS 同盟(AS Confederation)中の他のメンバーAS に出さないという意味を持っています。すべての BGP の Community 実装においては、これらの Well-Known Community 属性を定義通りに解釈実行しなければいけません。

Community の値(ユーザ定義)

・予約されていない値は、AS 毎に独自の Community を定義できる

・上位 16 ビット: Community を定義した AS 番号

・下位 16 ビット: その AS 内部で用いる Community 番号

・表記法: AS 番号: Community 番号

予約されていない領域は自由に使えるわけですが、AS ごとに独自の Community を定義して AS 内部の運用に使ったり、どこか離れた AS との間で AS 情報を経由したポリシーの伝達手段として使うこともできます。

Community 属性の利用例

・AS 内部での経路のグループ分け

・外部への経路アナウンスのポリシーに応じて Ingress Filter にて Community を定義

・2497:10 顧客の経路

・2497:20　peer の経路
・個別の経路情報ではなく Community の値のみに着目して Egress Filter を設定できる
・例えば、2497:10 を Community 値にもつ経路のみ upstream にアナウンスする等
・他の AS に対するポリシーの伝達

一般にユーザ定義の Community は、上位と下位の 16 ビットずつに分割して、その Community を定義した AS 番号、AS 内部で用いる Community 番号として使用し、表記的には AS 番号と Community 番号との間をコロンで区切ります。これによりいろいろな経路情報をグループ分けすることができます。例えば 2497:10 という Community をつけた経路は自分の顧客の経路、2497:20 というのは peer の経路、すなわち IX 経由で相互接続している相手の AS からもらった経路情報である、というように色分けをします。
特定の Community を持つ経路情報だけに特定のフィルタをかけて、外へのアナウンスや、ナウンスするときに何か他の属性をセットするなどの制御が可能になります。例えば顧客の経路だけを上位プロバイダにアナウンスしたいときには、Community2497:10 を持つ経路情報だけを上位プロバイダにアナウンスするというフィルタを、その上位プロバイダへの出口のところに設定します。
他の AS へ出ていくルータで設定するフィルタを Egress Filter、受け取るところのフィルタを Ingress Filter と呼びます。Community の導入以前は顧客の経路情報を制御するために、ネットワークのプレフィックスについての Egress Filter をひとつずつ書いてマッチするものだけアナウンスし、さらに顧客が増えるたびにフィルタを update する必要がありました。Community をうまく使うと、Egress Filter で Community にマッチするものだけ出すということを一度書いておけば、あとはどこで顧客から情報を受けても、また顧客が増えたとしても受けたところで Community を設定さえすればよいので、運用的には楽になります。

AS 同盟(Confederation)
・RFC1965
・AS 内部を、サブ AS に分割
・サブ AS 間の階層関係、包含関係は無い
・サブ AS では AS 番号にプライベート AS（64512-65535)を用いる
・各サブ AS では独立した IGP の利用が可能
・外部からは一つの AS に見える
・サブ AS 間は、IBGP に近い EBGP
・サブ AS 間で経路を渡すときには Next Hop, MED, Local Preference 等の値は保存
・大きな AS で、IBGP のメッシュを減らすのに役立つ

先ほどでてきました AS 同盟(AS Confederation)の説明で、これも RFC1771 以降の RFC1965 で定義されています。AS が大きくなってしまった場合に AS の中をいくつかに分けたいという状況が出てきますが、これはそのようなときにサブ AS に分割するための属性です。
ただし BGP の場合の Confederation には非常に制限があり、サブ AS 間の階層関係、すなわち１つの AS の中にサブ AS があって、さらにその中にサブ AS があるというのはサポートされていません。また１つの AS が２つのサブ AS に分かれていて、２つのサブ AS に共通部分があるというのもサポートされていません。１つの Confederation があり、その中に一階層のサブ AS がいくつもあるがサブ AS 間は交わりがない、という非常にシンプルな階層構造の場合だけが BGP4 ではサポートされています。
利点として、１つの AS を複数のサブ AS に区切ると、それぞれのサブ AS の中で違う IGP を使ったり、同じ IGP でも違うルーティングドメインを切って分けて運用したり、ということができるようになります。非常に巨大なネットワークでは全体を一つの OSPF で運用しようとすると、リンク、ネットワークや AS External の数も大きくなるため OSPF や update のトラフィックが大きくなります。このため Confederation でいくつかに分けて、それぞれのサブ AS の中でだけで OSPF を使います。サブ AS に分けたとしてもそれは内部の問題であって、外から見ると依然として１つの AS にしか見えません。
サブ AS に分けたあとで、当然サブ AS 間でも BGP の peer を張りますが、このサブ AS 間の peer というのは IBGP に近い EBGP といえます。つまり、ある部分は EBGP だけれども、ある部分は IBGP に近い、という中途半端な形になっています。例えばサブ AS 間で経路情報を渡すときには、EBGP だと Next Hop が変わっていましたが、Confederation 中のサブ AS 間の EBGP では保存されます。また、MED の値は非通過型の属性であるため、EBGP では取り外されるのですが、Confederation 中のサブ AS の間ではそのまま保存されて伝えられます。
もちろん Confederation から外に出るときには、その MED の値は取り外されますし、Next Hop の値も変わりますが、サブ AS 中の EBGP では、これらの情報や Local Preference も保存されたまま伝えられるというようになっています。

ポイントとしては、大きな AS では IBGP のメッシュ、すなわち BGP スピーカの間はフルメッシュを引かなければいけませんが、この数を減らすことができます。

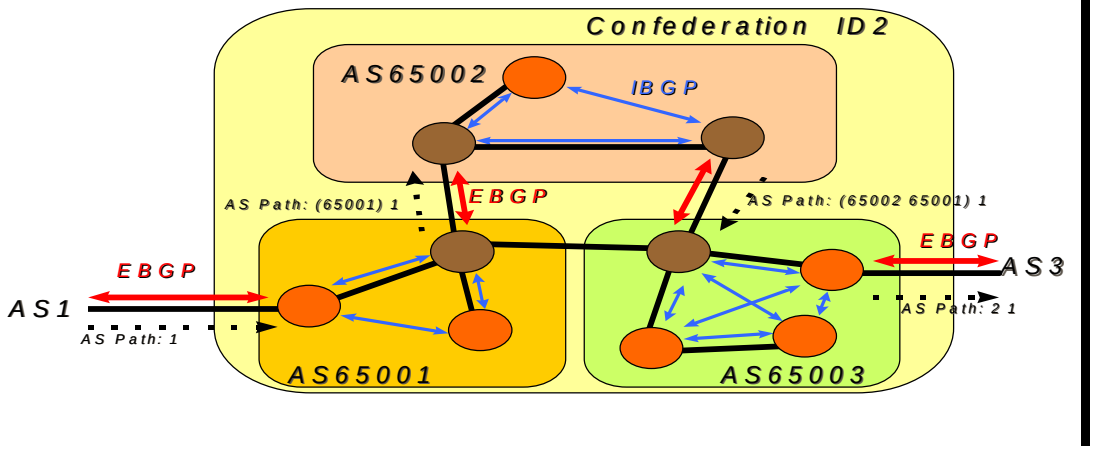


図 3.7.8: AS Confederation の例

図 3.7.8 は AS Confederation の例ですが、いくつかのルータからなる AS は AS1 と AS3 と相互接続しており、そこには EBGP のセッションがあります。その中に Confederation を導入するということで、ルータが３つある AS65001 と AS65002、さらに AS65003 に分けます。

ボーダルータのところでは EBGP、内部のルータ間では IBGP があったのですが、Confederation 中でサブ AS に分けましたので、さらにサブ AS 間でも EBGP のセッションを張る必要が出てきます。以前は AS の内部のボーダールータ間ですべてフルメッシュを張っていましたが、IBGP のフルメッシュというのはサブ AS の中で閉じることができます。

もちろん IGP についてもそれぞれの Confederation の中で閉じるような設定をすることもできます。逆に閉じないような設定も可能ですが、これは運用する AS のポリシーで決めることができます。

この状態で、AS Path:1 という AS Path 情報を持った経路情報が AS1 からやってくると、AS65001 内部のルータ間は IBGP で伝えられて、さらに他のサブ AS に行くときに AS Path:(65001) 1 というようなパス属性がついて出てきます。この 65001 と 1 とは厳密にいうと違うタイプの AS パス属性で、65001 は AS の Confederation 中だけのもので、1 というのは一般的な外とグローバルに使うタイプです。Confederation 中のサブ AS のパスはここでは括弧をつけて表記しています。さらに AS65002 から 65003 へは、AS65002 の AS Path をサブ AS のシーケンスの中に入れて伝わってきますので、AS Path:(65002 65001) 1 となります。さらに AS65003 が隣の AS3 に本当の EBGP を使って伝えるときは、サブ AS の情報シーケンスは取り払って、グローバルに見えている AS 番号(Confederation ID)である 2 をつけるということがなされます。

Route Reflector

・RFC1966

・AS 内部で用いるルートサーバ的イメージ

・BGP スピーカをグループ（クラスタ）に分ける

・リフレクタ

・AS 内の他クラスタのリフレクタと経路情報を交換

・クラスタ内の BGP スピーカに経路情報を供給

・クライアント

・リフレクタから BGP の経路情報をもらう

もう一つさらに Route Reflector というパス属性があり、これも RFC1771 以降の RFC1966 で定義されています。

IBGP では、peer から受けた経路は違う peer に対しては伝えないという大原則がありましたが、IBGP、BGP スピーカが増えていくとメッシュ状に IBGP の peer を張る必要があるので大変になります。そこでどこか一ヶ所に IBGP の経路を配布させようということで考案されたのが、この Route Reflector という概念です。

AS の内部で用いるルートサーバ(経路情報のサーバ)的なイメージですが、BGP スピーカをいくつかのグループに分け、グループの中には１個リフレクタを置いて、他の BGP スピーカはクライアントという形にします。そしてそのリフレクタに経路情報を伝えるとクライアントに同じ経路情報が配られるという仕組みにします。これにより全体の IBGP メッシュ数も減らして、クラスタを分けて運用したりもできます。

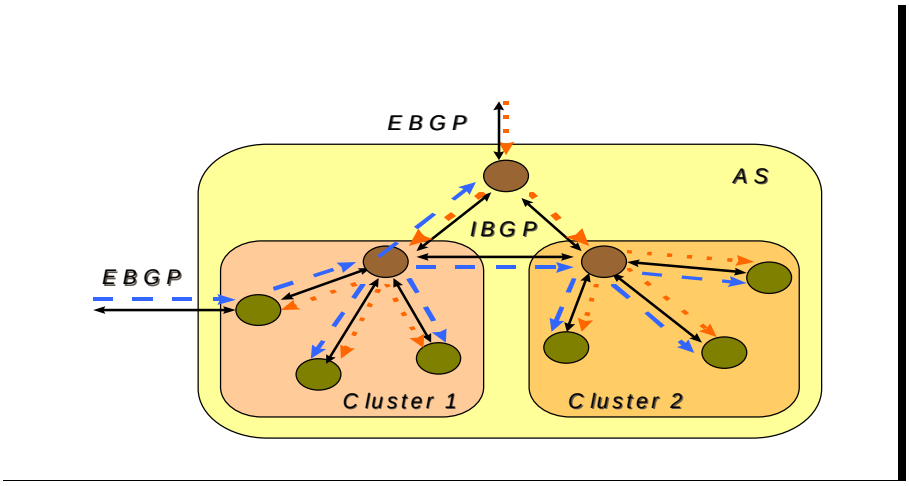


図 3.7.9: Route Reflector の例

図 3.7.9 は Route Reflector の例です。Cluster 1 と Cluster2 中の上の方にあるのが、それぞれの Route Reflector で経路情報を与えると、その下のクライアント全部に同じ情報が配布されるという仕組みになっています。クラスタが違う Route Reflector 間では情報を交換

しなければいけないので IBGP を張ります。また、どちらのクラスタにも属しない BGP スピーカがいる場合にも、同様に Route Reflector と IBGP のメッシュを張る必要があります。
例えば左から EBGP で経路情報が入ってくると、このボーダールータはクライアントなので受け取った情報を自分の Route Reflector に投げます。その Route Reflector はクラスタ中の他のクライアントへ情報配布するのと同時に、他のクラスタの Route Reflector やクラスタに属しない普通の IBGP peer にも配布します。さらに他のクラスタでも Route Reflector がクライアントに配布します。
上から EBGP で入ってきた場合も同様で、経路情報は IBGP でそれぞれのクラスタの Route Reflector に伝播されて、さらにそのクラスタ中でクライアントに対して配布されるということになります。

4. ポリシールーティング

4.1. ポリシールーティングとは？

・ポリシーに基づく経路選択
  ・他の ISP(AS)とどのようにトラフィックをやりとりしたいか
    ・単に近さやコストをもとにした選択ではない
    ・他の ISP との間でどのように経路情報をやり取りするか
    ・個々の目的地ごとに経路を選択する
    ・ BGP のパス属性を用いる
・経路情報のやり取りの制御だけでは実現できないポリシーもある
  ・ネットワークトポロジの再考などが必要

一般的にポリシールーティングとは何か、というのは非常に説明しづらい概念であるという気がします。IBGP として OSPF を使用するのであれば、リンクのコストが決まっていて、コストが最小になる経路を選ぶというわかりやすい説明になります。AS 間の経路制御でポリシーに基づいて経路選択をする、といっても、ではそのポリシーとは何かということになってしまいます。非常に簡単ないい方をすると、結局ポリシーというのは、自分の気持ちです。どのようにして他の ISP とトラフィックのやり取りを実現するかがポリシールーティングであるという言い方が一番いいと思います。もっと乱暴ないい方では、わがままをどうやって実現するかというのが、ポリシールーティングであるということになります。
これは OSPF の例のような、単にコストが小さいということだけを指標にした選択ではない、ということです。ポリシールーティングでは、他の ISP との経路情報交換の制御が必要ですし、隣接の ISP だけではなく色々なネットワークアドレスに対して目的地ごとに経路を選択していくという非常に難しい話になっていくわけです。

現在は、BGP のパス属性を用いてそのようなことを実現しています。これまでに説明しましたパス属性だけですべてのポリシーが実現できるわけではありません。ネットワークトポロジを変えることで対応できる場合もありますが、どのようにしてもできないものもあります。重要なのは、どうすれば自分が今やりたいことができるかを理解した上で隣の AS と話をしたり実際にルータの設定をする、ということがきちんとできるようになっていることだと思います。

4.2. ポリシーの例

・自 AS を通過させてよいかどうか

AS レベルのポリシールーティングということで一番簡単なのは、あるところからきたパケットを自分の AS を通過させて先に届けるかどうかということです。最近では商用インターネットという言葉がもういわれなくなるぐらい当たり前に商用化されていますので、お金を払ってくれるならば通すけれども、そうでなければ通さない、ということが普通にあるわけです。

・顧客 AS は、すべて通過可

相手が顧客であれば、お金を払ってもらっているので自分の AS を通して、さらに upstream にも通すということが普通だと思います。

・非顧客 AS(例えば IX での無償 Peer)

さらに顧客ではない AS、例えば IX 越しに peer するだけの AS というのは、一般的にはお互いにメリットがあるので無償でやっている例が多いと思います。この場合、自分の AS は通しても upstream には通さないというポリシーになることがあると思います。

・隣接 AS や経路上の AS の使い分け

また、隣接 AS とや特定のネットワークに至る経路上の AS を見て、こちらの AS は通りたくないということがあります。例えば、この AS はよく経路情報も落ちるし不安定だから AS Path が短くても選択したくない、という場合です。

・マルチホーム環境下で、料金の高い upstream はできるだけ使いたくない

マルチホームの場合では、こちらのプロバイダはパケット課金で高いのでなるべく使わない、などというポリシーもあると思います。

・自 AS 内のリンクの使い分け

・混んでいるリンクにはなるべくトラフィックを乗せたくない

さらに、混んでいるリンクにはこれ以上トラフィックを乗せたくないので、空いた経路に通したいというような、いろいろなポリシーがあります。

4.3. ポリシールーティングの実装

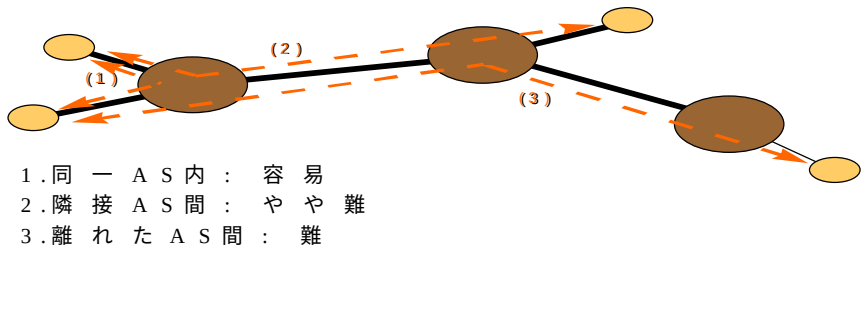


図 4.3: ポリシールーティングの実装

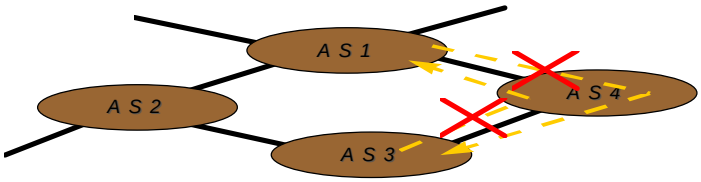
図 4.3 には３つの AS があり、一番左側にあるのが自分の AS です。自分の AS 内の制御というのは、自分でコントロールできるので非常に楽です。隣接 AS 間では、もともとオペレータ同士や経営者同士が何らかのコミュニケーションをとれているわけですから、ポリシー制御の調整は比較的容易です。AS 間のポリシー制御で難しいのは、離れた AS 間の場合に第三者の AS が関係してしまうことで、自分と隣接 AS 間だけの調整ではコネクティビティーを確立できなくなります。

・経路情報のフィルタリング
  ・パス属性に基づいて特定の経路情報をピックアップ
  ・Ingress フィルタ (受信時のフィルタ)
  ・Egress フィルタ (送信時のフィルタ)
・BGP パス属性を操作
  ・Local_pref, MED, AS Path Prepend 等
  ・Community を用いた経路の分類
・経路情報の操作によって、すべてのポリシーを実現できるわけではない

ポリシールーティングの実装で具体的に行うことは、まず経路情報のフィルタリングです。いろいろなパス属性の経路情報を、いろいろなところから BGP で受けるわけですから、そのパス属性の値による Ingress Filtering や外に出すときの Egress Filtering でのポリシー制御、また、フィルタにマッチしたものに特定の AS Path 属性を付けるなどということを指します。
実際にどのように AS Path を操作するかと言いますと、Local_pref や MED を変える、AS Path の短い方が選択されてしまうため Prepend して長くする、運用を楽にするために Community を使って経路を色分けする、というようなことを行います。また、離れた AS に調整をかけるのは非常に難しいのですが、Community 分けをして複数の経路で同じ情報

を出して、遠くの AS に対しては例えば赤色をついたものを選ぶように依頼する、というようなことを行います。しかし、これらの操作をすることだけによって、すべてのポリシーを実現できるわけではないということに注意してください。

4.4. 通過ポリシー



- ＡＳ４は、ＡＳ１とＡＳ３の通信を中継したくない
- ＡＳ_ＰＡＴＨパス属性を用いた経路情報のフィルタリングなどにより実現

図 4.4: 通過ポリシー

・AS4 は、AS1 と AS3 の通信を中継したくない

・AS_PATH パス属性を用いた経路情報のフィルタリングなどにより実現

図 4.4 は通過ポリシーを説明したものです。例えば AS4 では、自分は ISP でも何でもないので、自分と AS1 や自分と AS3 で単に閉じるトラフィックはよいが、AS1 と AS3 の通信を仲介したくない、という場合があります。この場合 AS4 は、例えば AS1 から受け取った経路情報を AS3 には出さないとか、逆に AS3 から受け取った経路情報を AS1 には出さない、というようなフィルタリングを行います。このようなものは実現が比較的簡単なポリシーです。

4.5. 複数経路の選択

さらに、実現したいポリシーとしては、ある AS に対する複数経路の選択というものがあります。

・ある AS へ複数の経路がある場合、どの経路を優先するか？

・AS_PATH の短い経路を優先

例えばオーストラリアあたりの AS に対して複数の経路がある場合、どの経路を優先するのかは、自分の AS で、経路情報を受け取ったところで選択しなければいけないわけです。デフォルトでは AS Path により近い経路が選択されたりするわけで、必要に応じて Local Preference を立てたりする必要があります。

・NEXT_HOP までの IGP 的な距離が短いパスを優先 (Hot Potato)

経路選択には、Next Hop まで一番近い出口から出すというポリシーもあります。アメリカ人たちはこれを Hot Potato ルーティングと呼んでおり、ネットワークに対する複数経路の選択をする場合に、いかに自分の中の少ないリソースを使うかで選ぶというものです。Hot Potato とは、焼きたてのイモは熱いので、できるだけ早く相手に渡してしまいたいという意味をこめています。例えば、Next Hop が２つあって、それらの Next Hop に対する経路

情報が自 AS で OSPF により流れている場合、優先する Next Hop までの経路情報を、例えば External のタイプ１で流してやれば、External の経路もメトリックを見ますので、ルータはトータルコストが小さい方を選択します。

・特定の経路を優先
・顧客からの経路を優先
・特定 IX 経由の経路を優先

また、顧客から来た経路情報や特定 IX 経由の経路などの特定の経路を優先するというようなポリシーもあります。

AS PATH による経路選択

・AS4 での経路選択
  ・経路(1)の AS Path: AS2 AS1
  ・経路(2)の AS Path: AS3 AS2 AS1
  ・通常は AS Path の短い経路(1)が選択される
  ・Ingress Filter で経路(2)の AS Path に高い Local_pref の設定も可
・AS5 は AS4 と異なるポリシーをもてない

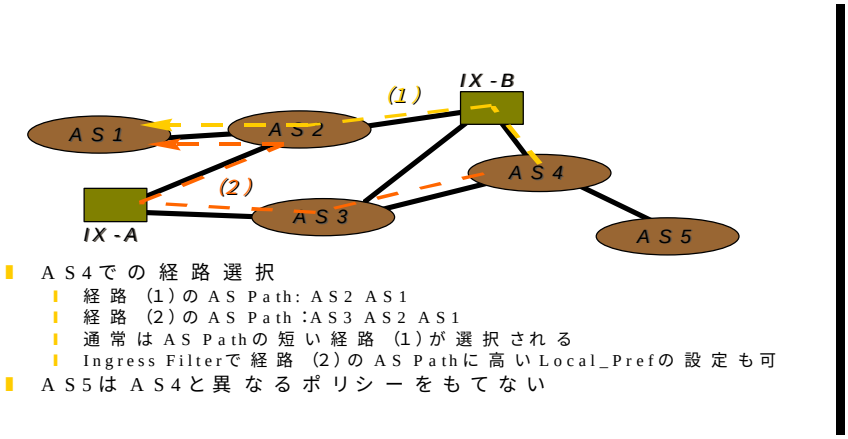


図 4.5.1: AS PATH による経路選択

複数経路選択では、AS Path 属性による方法が一番よく使われます。図 4.5.1 で、AS4 は複数経路で AS1 からの経路情報を受け取っています。AS1、AS2、AS4 という経路についている AS Path 属性は AS2 AS1 です。ここでは便宜上、数字の前に AS というのをつけていますが、実際は show ip bgp をすると、2 1 などと出てくるわけです。

AS4 が、AS3、IX-A 経由で AS2、AS1 という経路を選ぶためには、経路情報として AS3 AS2 AS1 というパス属性がついた経路が来ていなければいけません。通常では短い方が選ばれてしまいますが、IX-B は混んでいるので嫌だということがあるとします。その場合、AS4 が経路情報を受け取る Ingress Filter で、Local Preference を落としてやれば IX-A 経由の方が選択されます。

問題になるのは、AS4 の下に AS5 があった場合で、これは、すべてのポリシーが実現できるわけではないという例になります。AS4 が、AS1 に対して IX-B 越しで AS2 を経由してトラフィックを流したいというポリシーを持っていた場合には、AS5 が違うポリシーを持つことはできません。例えば AS5 が違うポリシーで、AS4 から AS3 へ行き IX-A 越しで AS2 を通って AS1 に行きたいとします。しかし AS5 からのパケットは、AS4 のポリシーにより、IX-B 越しで AS2、AS1 と送られてしまいます。BGP の Path 属性を単純に操作するだけでは、AS5 が AS4 と違うポリシーを持つことはできません。
例えば Cisco では、ポリシールーティングという機構があり、ポリシーフィルタを書いて、経路表の経路情報を無視してパケットフォワーディングする、という恐ろしい設定ができます。BGP 以外にこのようなものを使わないと、今のようなポリシーは実現できないということです。

隣接 AS 間の複数パス
・一方のパスを優先
・MED を利用
・非効率な経路になる場合もあり

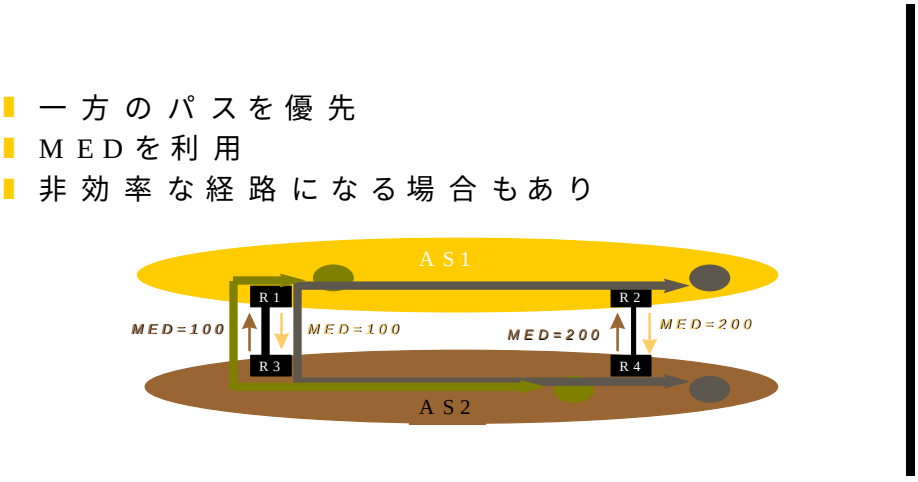


図 4.5.2: 隣接 AS 間の複数パス

複数経路の選択として、隣接 AS の一方のパスを優先したい場合には、MED を使えば簡単にできます。例えば図 4.5.2 で MED=100、MED=200 としておけば、MED の値が小さい方のリンクを使います。しかし、これは必ずしも効率のよいトラフィック交換ではありません。これらのルーターの距離が近い場合にはよいのですが、例えば二つのリンクが東京と大阪だとすると、それぞれの AS 内部で東京と大阪を往復することになります。ですから、MED を使うと非効率的な経路になる場合もあります。

隣接 AS 間の複数パス(Hot Potato)
・最も近い出口から次の AS に渡してしまう

・MED の値を等しくして、Next Hop の解決を IGP で行う
・複数 IX で相互接続している ISP 間で一般的に行われる方法

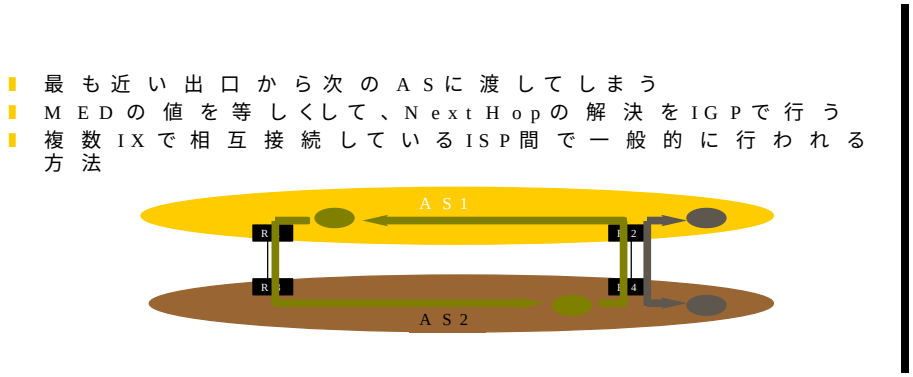


図 4.5.3: 隣接 AS 間の複数パス(Hot Potato)

もう少し効率的なことを考えると、先ほどの Hot Potato ルーティングをするのがよいと言われています。これは最も近い出口から相手の AS にパケットを渡してしまうやり方で、Hot Potato ルーティングというのが、ITU の定義や RFC に書かれているわけではありませんが、みなこのように言っています。

例えば MED では、どうしても特定の相手のアドレスに対してどちらかのリンクを優先するようになりますので、MED ではできません。ですから MED は両方同じにしておいて、Next Hop の解決を IGP で行います。AS1 と AS2 間の経路情報には、同じ経路情報でも Next Hop 属性が違うものが２つあるわけで、これは AS 間の２つのリンクに対応しています。この場合、リカーシブに経路表をルックアップして、特定の経路に対して Next Hop が近いほうを優先するということを行います。

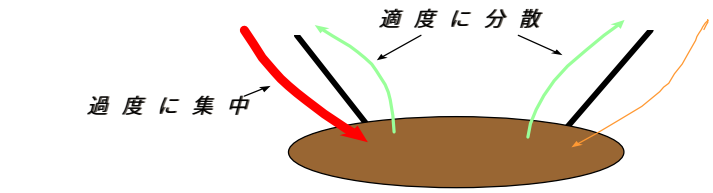
例えば今の例では、それぞれのリンクの情報が IGP に流れていて、その IGP で得たコストが付いた経路情報がテーブル中にあります。この場合、２回引いて経路の解決をするわけですが、BGP での複数経路の選択でも、この IGP のメトリックを使った Next Hop の近さというのが一つの指標に使えます。ルータが IBGP の情報を比べるときに、パス属性も同じ長さで MED も同じというときには Next Hop を比較します。Next Hop を見ると、違うアドレスがついていて、それぞれの Next Hop に対して経路表をルックアップすると、例えばコスト 10 とコスト 11 であるという値が得られます。

そうすると、BGP の経路選択アルゴリズムの一部に使われているコストが低い方を選ぶという機能が有効になり、それぞれのリンク近辺で IBGP により経路をもらっているものは近い出口を選ぶようになります。つまり、OSPF のコスト的にコストが小さい経路を選ぶように制御できるわけです。

２つのリンクをメインとバックアップにするというやり方と、近いほうを使うというやり方のどちらをとるかいうのはそれぞれのポリシーにより決まることですが、例えばアメリカの ISP では、これらのリンクが複数の IX、例えば東海岸や西海岸でつながっている場合

には、MED をつけずに Hot Potato ルーティングするということが共通な約束ごとになっているようです。ただ、必ずしもこれに従わない人ももちろんいます。

4.6. マルチホーム下でのロードバランス



- 他ASに出るトラフィックの調整は容易
  - 自ASで受け取る複数経路間の選択の問題
- 他ASから入ってくるトラフィックの調整は困難
  - 他ASでの複数経路間の選択を制御しなくてはならない
  - MED,AS PATH Prepend, Community等を駆使

図 4.6. マルチホーム下でのロードバランス

マルチホーム下でのロードバランスということですが、他の AS に出るトラフィックの調整というのは自分で制御できるという意味で容易です。しかし、他の AS から入ってくるトラフィックというのは、自分がアナウンスした経路を相手がどう選択するかで変わるわけですから、その調整は非常に難しくなってきます。
これを行うためには、同一 AS とマルチホームしているときは MED が使えますが、違う AS にマルチホームしているときは、AS Path を Prepend してみたり、色々な Community を使ったりという、いろいろな方法を駆使して入ってくるトラフィックをうまくバランスさせなければならないことになります。

経路情報を受け取る AS での選択

自分の AS に入るトラフィック制御のために、相手の AS にどういう経路選択をさせ得るかということを説明します。

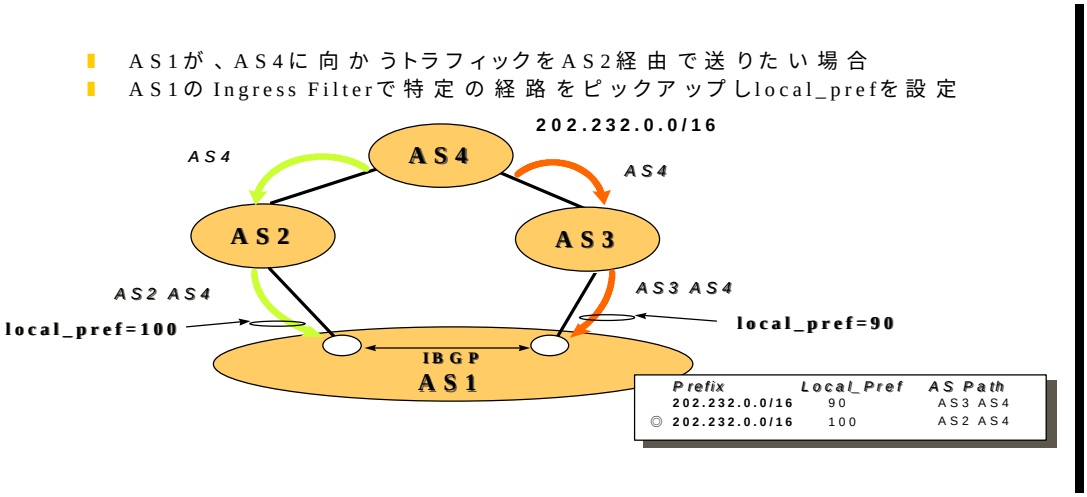


図 4.6.1: 経路情報を受け取る AS での選択

図 4.6.1 は、AS1 と AS4 の間には複数経路があり、AS Path 属性の長さは同じという場合の、AS1 における２つの経路情報間での優先制御です。例えば AS1 の Ingress Filter で、AS Path: AS2 AS4 に対して local_pref=100、AS Path: AS3 AS4 に対しては local_pref=90 とすれば、Local Preference が高い方の経路が選択されます。BGP のテーブルの経路情報では、同じプレフィックスで AS Path 長も同じですが、Local Preference が違っているということで経路制御が行われるわけです。

経路情報をアナウンスする側からの調整(1)

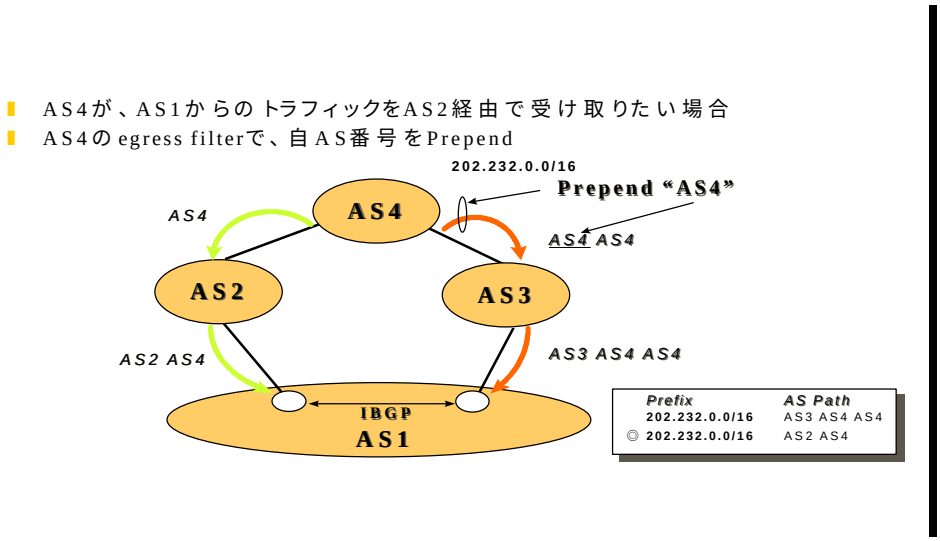


図 4.6.2: 経路情報をアナウンスする側からの調整(1)

図 4.6.2 は、前の例とは逆に、相手から自分に来るトラフィックを制御する方法です。この図では AS4 が自分の AS ですが、先ほどの例とは逆になってしまっています。この場合は自 AS、AS4 の Ingress Filter で何をしようと、相手側の AS1 での経路選択には影響しないわけです。こういう場合の方法の一つに、AS Path Prepend というものがあります。AS Path 長が短い方が優先されるわけですから、優先されたくない方のリンクに対してアナウンスするときに１個 Prepend を余分に AS Path をつけて出します。これにより AS1 側では、受け取った経路情報をみて、短い方へ送るという経路選択がされるわけです。

経路情報をアナウンスする側からの調整(2)

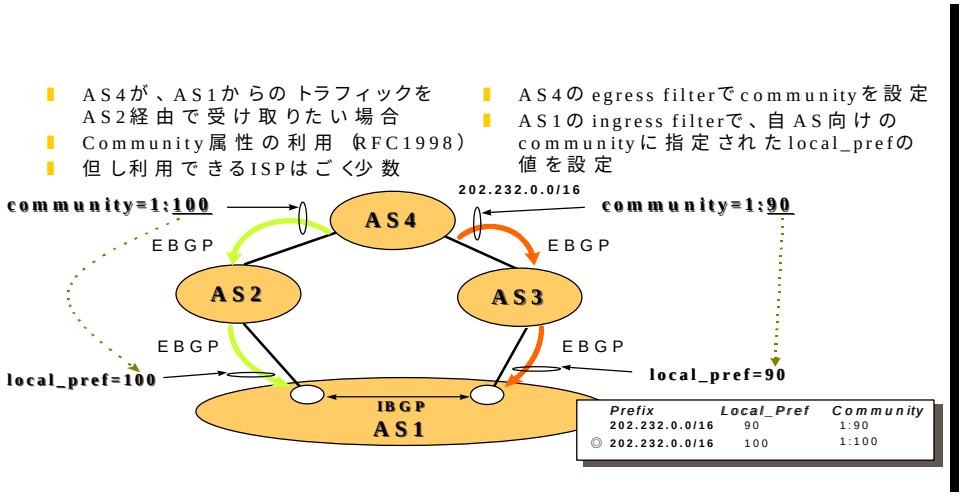


図 4.6.3: 経路情報をアナウンスする側からの調整(2)

もう一つ他の方法として Community を使うものがあります。図 4.6.3 で AS4 が AS1 からのトラフィックを AS2 経由で受け取りたい場合、AS4 が外に出すときに Community を設定します。例えば、AS3 経由の方に community=1:100、AS2 経由の方には community=1:90 というのを流します。community 属性の値ですが、コロンの左側はそれを定義した AS 番号を表しています。つまり、これはあらかじめ AS1 側でそのような Community を使うと表明しており、AS4 ではその値を使って依頼するという形になります。例えば AS1 で、Community 属性の 1:で来たコロンの右側の値を受け取った経路情報の Local Preference に設定するということを行っているとします。その場合、AS4 が経路情報にこのような Community をつけてやれば、受け取った AS1 側では、それに相当する制御をし行い、アナウンスする側からの調整ができることになります。

この方法は当然、相手の AS がサポートしないとだめで、どこの ISP でも使えるわけではありません。例えば MCI はサポートを表明していますし、RFC も書いていたりしていますが、IIJ の場合はまだサポートしていません。

4.7. ルーティングレジストリとルートサーバ

- パケット転送と経路選択のプロセスの分離
- ルーティングレジストリ (RR)
  - 各 AS の経路制御ポリシーのデータベース
- ルートサーバ(RS)
  - 第 2 層エクスチェンジに接続する ISP と BGP で通信する
  - RR に登録されたポリシーをもとに、各 ISP のボーダルータの経路表を計算する
- RR を用いた経路フィルタリング

これまで説明したようないろいろな制御が必要になってくると、ルータの設定の変更や一貫性の保持という点で管理が非常に煩雑になってくるという、運用上の弊害が出てきます。これを解決する方法として、ルーティングレジストリとルートサーバという考え方があります。これは、パケットの転送処理と経路選択のプロセスを分離してしまおうという考え方で、ルーティングレジストリというのは各 AS の経路制御ポリシーを集めたデータベースです。

これを使って、特定のデータベース記述言語により自分の AS のポリシーを登録しておきます。ルートサーバではあからじめレジストリに登録されたポリシーに基づいて、各 ISP から受け取る経路情報を計算して、ISP 向けの経路表を作成します。そうすると、例えば IX などにおいたルートサーバが、IX で交換される経路情報の選択を代行するということができます。

ルーティングレジストリ自身は一般的に皆がアクセスできる経路制御ポリシーのデータベースになっていますので、IX でなくても自分の AS と隣の AS との間での経路制御のために特定のデータベースを使うということができます。

例えば、そのデータベースに前日の午前０時までに update をかければ翌日には、そこからの情報でフィルタを自動的に書きかえて実装するということを取り決めれば、お互いのポリシーをエクスチェンジして、ルータの設定まで済ませてしまうということも可能ではあります。

問題は、そのためにはルーティングレジストリが非常に重要になるということです。この情報が一貫していなかったり最新でなかったりするとルートサーバを使った IX 上の経路情報交換のサービスや、ルーティングレジストリを使ったフィルタの自動更新などというものは到底動かないわけです。

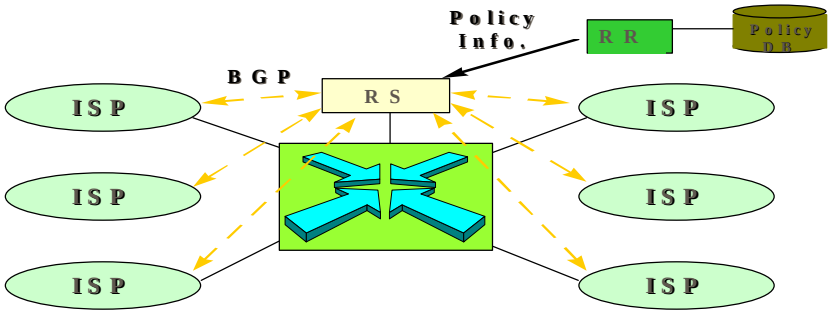


図 4.7: ルートサーバ

ルートサーバを図示すると図 4.7 のようになります。現状はやはり、RR、ルーティングレジストリの情報が最新でなかったり、一貫していないという状況があるため、ルートサーバ、RS を使った経路交換や RR を使ったフィルタの自動更新というものは、なかなか進ま

ない状況にはあります。しかし実際に一部の人たちは、ここを一生懸命やっていますし、このルーティングレジストリの記述言語をどうするかということを何年もかけて ITF でやってきている人もいます。
将来的には、このようなデータベースを使ってポリシーを登録し、自動的にルータを設定するという世界もやってくる可能性はあります。

5. BGP の運用上の問題

実際に ISP のオペレータたちは、このようなポリシー制御をやっているわけですが、そこで今までに出てきた運用上の問題をいくつか説明します。

5.1. IBGP フルメッシュ

・BGP の仕様
  ・ボーダールータ間でのみ IBGP のフルメッシュを張る
  ・外部から学んだ経路は IGP で AS 内部のルータに伝播する
  ・ボーダールータで IBGP と IGP の間で同期を取る
    →フルルートを IGP（例えば OSPF）で流すのは非現実的
    →BGP の経路情報の不安定化につながる→flapping
・IBGP "HACK"
  ・フルルートを持つ必要があるすべてのルータ間で IBGP のフルメッシュを張る
  ・IBGP と IGP との間の同期は取らない
  ・Next Hop の解決は IGP で行う

IBGP のフルメッシュというのは、BGP の仕様上、外の BGP スピーカと自 AS 中のルータすべての間で IBGP コネクションを張って、外部からの経路情報を IGP で AS 内部のルータに伝播しなければいけない、ということです。
さらにボーダールータで IBGP と IGP の間の経路情報を同期させて、同期が終わった時点でルーティングテーブルに挿入し、他の AS にアナウンスしなければいけないということが、仕様上というか教科書的に書いてあったりします。
しかし実際問題としては、インターネット上のフルルートを受けているような場合、現在５万何千経路とあるわけですので、IGP に例えば OSPF で再配布するのかという問題が起きますが、これは現実的ではありません。
例えば Cisco の 7500 クラスでメモリをいっぱい積んでも、BGP でフルルートを受けて OSPF に５万経路というと結構音を上げると思います。これが複数のポイントで外からフルルートをもらって、複数のポイントから OSPF の AS External の経路で流してしまうと、経路情報をハンドルするだけで手一杯で、実際のパケットフォワーディングができなくなるといわれています。ですからこのような教科書的なやり方は、実運用上不可能ということが、BGP を運用し始めたころにもうわかっており実際にはあまり使われていません。
もう一つ、再配布すると BGP の経路情報も不安定になるという問題があります。IGP か IBGP に流れてきて同期をしたら相手に流すということなので、自分の AS のどこかのリンクが落ちたとすると、自分が IBGP で持っている経路情報に対する IGP 的な経路も一度落ちてからまた上がるということになり、flapping というものが起きます。本来 IGP が変化しても BGP には関係ありませんが、IGP と BGP を同期させているために、一度ルーティングテーブルを外してから戻すということになり、その先で peer をしている AS 全部に同

じような update を送るわけです。こういうことが、いろいろなところで起きると、インターネット全体の経路システムが非常に不安定になり、Route Flapping というような現象が起こってきます。
これらの問題の解決方法として IBGP "HACK"というものが使われています。これは何かというと、ポイントは IBGP と IGP の同期はとらないというものです。つまり、BGP で受け取った経路情報は IGP に流さない、そのかわりに例えば、フルルートを持つ必要があるすべてのルータの間で IBGP のフルメッシュを張るという方法をとります。すなわち外の AS と直接通信していないルータでも IBGP を使って AS 内部の BGP スピーカと通信するということです。
こうすると、フルルートの情報は全部 IBGP で配るということをしますので、経路情報を OSPF で配送しなくてもよくなります。つまり、フルルートの情報は BGP で運んで、OSPF では例えば中のリンクの状態だけとか、外とつながるリンクだけの情報を流すことになります。

IBGP フルメッシュとの戦い
これがいわゆる IBGP "HACK"と呼ばれて、今日広く使われている手法です。従来の方法では BGP で受け取ったフルルートを IBGP と OSPF の両方で流すわけですから、BGP スピーカは BGP と OSPF テーブルの両方にフルルートを持っていますが、そういう非効率なことが避けらます。
痛いところとしては、これまでは BGP スピーカが増えると IBGP のメッシュが増えていたのですが、IBGP "HACK"ではフルルートを持たせたいルータが増えると IBGP のフルメッシュの数が増えることになります。BGP の仕様に書かれた方法では現実的には動かないので、IBGP "HACK"でしのいできたのですが、だんだんこのフルメッシュが効いてくるわけです。ルータの数 n が増えていくと、IBGP のメッシュの数が n の２乗のオーダーで増えていきますので、これは非常に問題です。

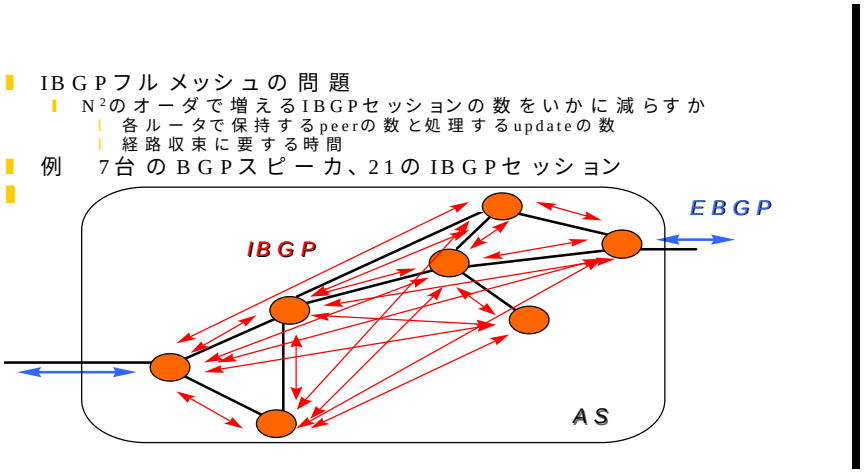


図 5.1.1: IBGP フルメッシュとの戦い

例えば図 5.1.1 の例では、この AS にはルータが７台しかないのですが、外と BGP でしゃべっているのは２台で、中では IBGP "HACK"ですべてのルータでフルメッシュを張っており、21 の IBGP セッションがあります。

Confederation の利用例

- ２つのサブＡＳ
- ２つのＥＢＧＰセッション
- ９のＩＢＧＰセッション

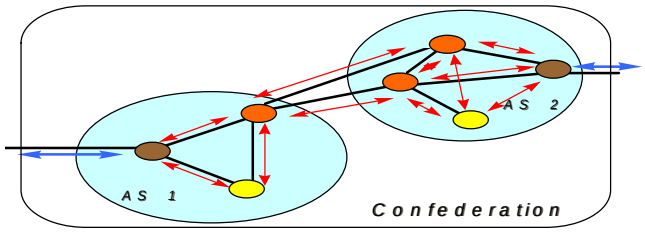


図 5.1.2: Confederation の利用例

これを解決する方法の一つに先ほど説明しました Confederation があります。Confederation を使えば、サブ AS のバウンダリだけで経路情報をやり取りすればよいので、フルメッシュの数を減らすことができます。図 5.1.2 では、先ほどの例で 21 個あった IBGP セッションが、中を２つに分けて Confederation することによって 9 つの IBGP セッションになっています。

Route Reflector の利用例

- ２つのクラスタ
- ３台のリフレクタ、４台のクライント
- ９のＩＢＧＰセッション

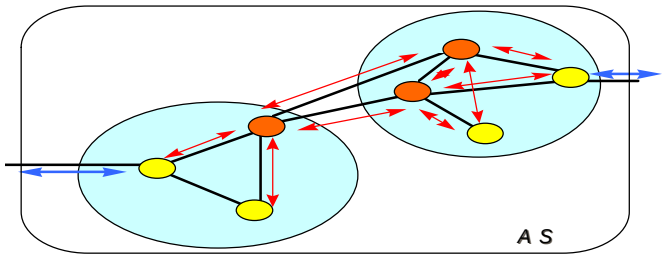


図 5.1.3: Route Reflector の利用例

同様のことは Route Reflector を使ってもできます。Confederation の例では AS の内部にサブ AS がありましたが、図 28 では、Route Reflector のクラスタが 2 つあります。この場合、リフレクタがクラスタ内部の経路情報をクライアントに配ります。さらに、クラスタを超えた IBGP セッションも不要になり、9 つの IBGP セッションに減らすことができ

ます。Confederation と Route Reflector どちらでも、IBGP のセッションを減らすという意味では同じような効果が期待できますが、どちらを使うかは最終的には趣味の問題であると思います。
一つ違うのは、Route Reflector の場合はクラスタに分けていますが、そのクラスタごとに違う IGP のドメインを定義して経路制御するというところまではサポートされていません。Confederation では、サブ AS に分けてそれぞれのサブ AS の中で違う OSPF のドメインで運用することもできます。ですから、絶対的なルータの数が増えて IGP として OSPF を使っている場合、BGP 的にも苦しいけれども OSPF 的にも苦しくなってきたというときには、全体をいくつかのサブ AS に分けてそれぞれのサブ AS の中では異なる OSPF のプロセスを走らせるという方法が現実的であると思います。
また Confederation やサブ AS という方法ではなく、別のグローバルな AS をとってグローバル AS を２つ用いて運用するという方法もあります。

5.2. Route Flapping と Dampening
・不安定な経路情報
　・上がったり落ちたり、属性が変化したりを繰り返す(Route Flapping)
　・大量の UPDATE や WITHDRAW メッセージの処理に、ルータの CPU やメモリ資源が浪費される
　　・インターネット全体の経路制御システムへの影響
・ある程度以上不安定な経路情報は落ちたものとみなす（Dampening)
　・一定の条件を満たすまで再びその経路は採用しない
・時に板ばさみ
　・経路を flap させるほうが悪いのか、勝手に dampening するほうが悪いのか？

Route Flapping という現象が、インターネットが大きくなって相互接続される ISP が増えるに従って問題になってきています。BGP は Incremental なプロトコルで、何か変化があったときのみ update を行うので効率がよいはずなのですが、あまりにも変化を繰り返すような経路があると、そのような update の情報もばらまいてしまいます。しかも、BGP で AS 間の制御をしていますから、インターネット全体の経路情報に対して効いてくることになり、不安定な経路情報が大量に流れていたりすると、インターネット上のすべてのボーダールータ、EBGP スピーカに影響を与えるわけです。
そうするとルータは、本来の役割であるパケットフォワーディングよりも、経路情報 update の処理や経路テーブルのメンテナンスに時間を食われてしまい、どんどんパフォーマンスが落ちていくという弊害が出てきます。
そこで考え出されたのが Dampening という手法です。これは、一定の不安定さを示した経路情報に関しては、ある一定時間、経路を採用しませんというやり方です。これにより

不安定な経路は、AS のバウンダリのところで Dampening されて、update が来ても採用されないという状態になります。
Flap している経路の Dampening に対して顧客からクレームがつくという問題もあり、また、どういうときにその経路情報を Dampening にしてよいかとか、誰がそれをしてよいかという議論もまだ残っています。しかし実際に運用する上では、Dampening を入れざるを得ないという状況もあります。

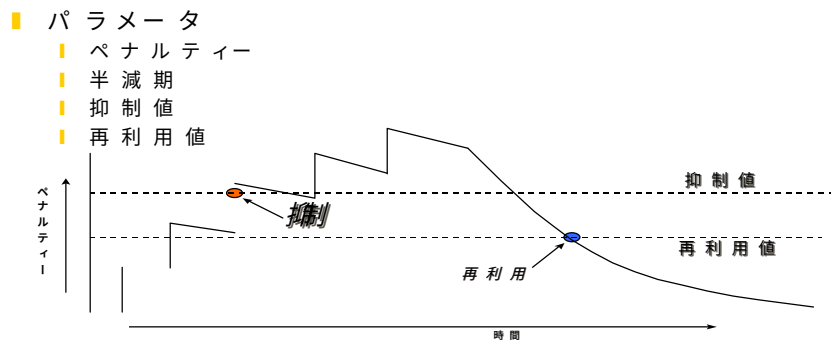


図 5.2: Dampening

図 5.2 に Dampening の例を示します。横軸が時間経過で、Flap する経路にはペナルティと呼ばれる値を与えます。Flap すればするほど特定の経路情報がペナルティの値が増えていくようになっています。また、いったん与えたペナルティが、一定時間経過すると半分になるという半減期というのが定義されています。
ある Flap する経路情報に関してペナルティの値と時間との関係はこのようになり、ペナルティがある値よりも増えたら Dampening するということになります。そのときの値を抑制値(Suppress Limit)と呼んでいます。Flap しないで安定していると、半減期を過ぎるに従ってペナルティの値は減っていきますが、再利用値(Reuse Limit)よりも小さくなれば再びその経路情報を採用します。
ではすべての ISP が、この Dampening をやっているかというと必ずしもそうではありません。経路情報を Flap するのが悪いとはいい切れないからです。ダイナミックルーティングには、落ちて上がってという情報を素早く伝えて、素早く違う経路を選ぶべきという観点もあります。ですから例えばダイヤルアップ接続などの場合、切れているときには落として、つながったときだけ上げたいという欲求は、そんなに間違った欲求ではないように思うわけです。しかし、それをみんながやると、今のルーティングシステムが動かなくなってしまうので、それはやめてくださいといっているのが現状です。
技術的には Dampening をやればよいとしても、このように調整しなければならないことがいくつもあります。また、Dampening に関しても、パラメータがあるわけですから、その条件をアナウンスしながらやるべきかもしれません。つまり、自分の AS での判断、ポリシーによるということになると思います。

5.3. ポリシーの不整合

・自 AS のポリシーと、隣接 AS やその他の AS とのポリシーが擦り合わない場合がある

・隣接 AS 間

・Hot Potato か Cold Potato か

・相手のアナウンスする MED を尊重するか否か

・顧客の経路と peer の経路

・顧客から受け取る経路を最優先する ISP もあるが、それ以外の経路を優先したいときもある

・不整合を解決するために、特例的な設定を増やしていくと設定管理上の問題が発生

ポリシーが必ずしも整合しないという問題もあります。例えば自分がある経路を優先したいけれども、相手は別の経路を優先したいというような場合です。

例えばある AS は Closest Exit から Hot Potato で出したいと考えてもそのリンクは混んでいるから別のリンクを通したい、Hot Potato ではなく Cold Potato をやりたい、というような隣接 AS 間でもポリシーの不整合があります。また MED についても、自分が２つのリンクを使い分けしたいので MED をつけて送るわけですが、有効になるかどうかは、相手側のポリシーによるわけです。

また、顧客の経路と peer の経路で優先度を変える場合ですが、ダイレクトのコネクションと IX 経由のマルチホームである ISP とつながっているような場合には、必ず顧客－顧客連鎖の中のほうを使うというポリシーを Local Preference でちゃんと実装している ISP もあります。しかし、顧客の顧客が自分との間はこちらの IX で peer しているので、それを使いたいということもあります。それを受けるか受けないかというのは AS のポリシーで「IX 経由より、コマーシャルベースで料金をお支払い頂いているほうは 24 時間監視していますので、こちらの方が絶対いい経路です。」ということもあります。

ポリシーの不整合（隣接 AS 間）

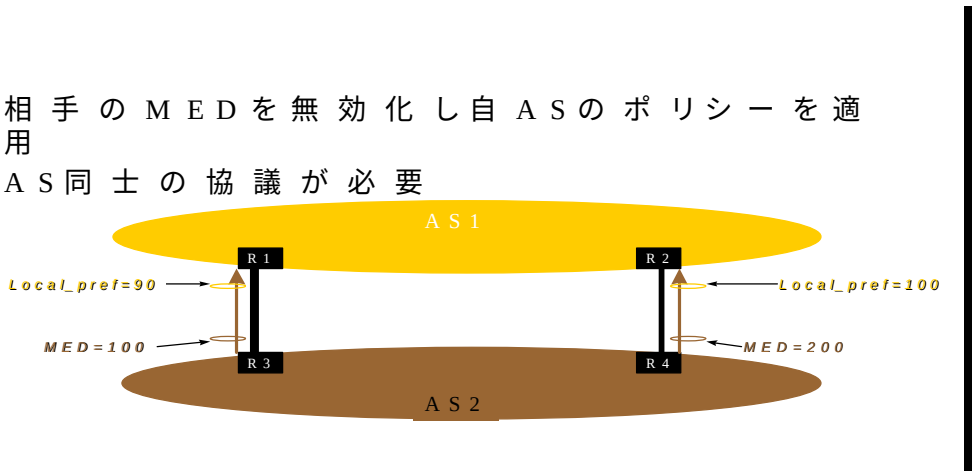


図 5.3.1: ポリシーの不整合(隣接 AS 間)

例えば図 5.3.1 では、AS2 が MED で左側のリンクを優先したいと思っても、AS1 の方は別のリンクを優先したいので Local Preference をあげています。この戦いは Local Preference を立てるほうが勝つわけです。

AS1 では、受け取るときに MED を書きかえてしまうこともできますし、Local Preference のほうが MED よりも強いアトリビュートなので先に評価されて、AS1 のポリシーが優先されてしまいます。この場合、AS2 はどうしようもないわけで、あとは AS 間で協議するしかありません。結局 AS2 と AS1 間のポリシーのすり合わせをまずやらなければいけないということになります。

ポリシーの不整合（顧客か peer か）

もう一つの例ですが、先ほどの顧客か peer かという問題にさらされることが何度かあります。

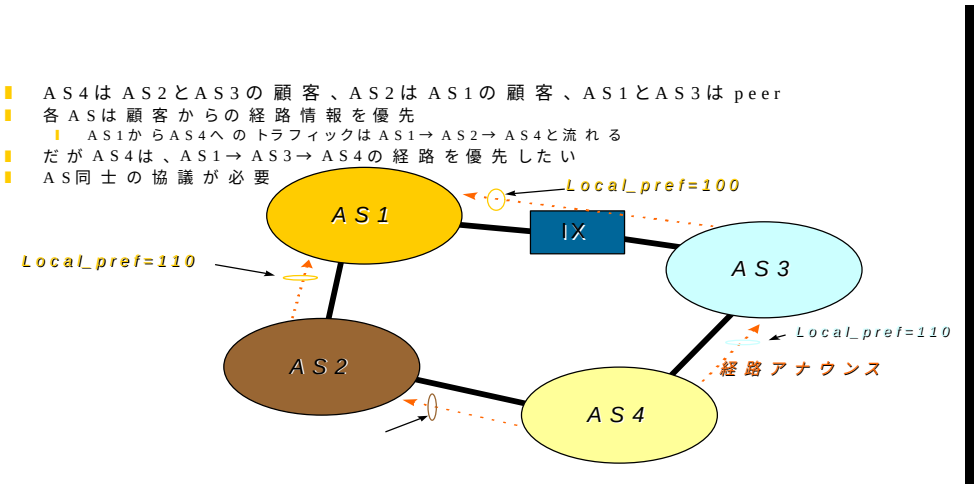


図 5.3.2. ポリシーの不整合

例えば図 5.3.2 で、AS4 は AS2 と AS3 の顧客で AS2 は AS1 の顧客になっており、また、AS1 と AS3 は IX 経由で peer しているとします。AS Path 的には AS4 と AS1 の間は 2Hop ですからどっちも同じなわけですが、各 AS は顧客からの経路情報を優先していますので、AS4－AS2 と AS2－AS1 の顧客連鎖のために AS1 から AS4 のトラフィックは AS1-AS2-AS4 の経路を通ることになります。
AS4 では、AS1-AS3-AS4 の経路を使いたいとしてもどうしようもないわけです。AS1 が例えば Community で Local Preference を設定すれば、Community での制御も可能になりますが、AS1 の方でも IX 越しの方に対してそのような処理はしていないことが多いと思います。

5.4. 不正な経路情報

・不正な経路情報のアナウンスによる事故
　・フルルートを自 AS の origin として流してしまう
　　・その AS がパケットのブラックホールとなる
　・不必要な more specific route を流してしまう
　　・Aggregate よりも more specific にひきずられトラフィックが最適な経路を経由しない
　・IX のセグメントの経路を BGP で他の AS に流してしまう
　　・Next Hop の解決で最適経路を選べない

不正な経路情報というのもたまに攻めてくることがあります。例えば、何かの設定のミスで自分の AS を Origin AS としてフルルートなどを流してしまった場合です。このようなことが起こると、その AS がパケットのブラックホールになってしまい、隣の AS に行こうと思っても、ブラックホールに吸い込まれてしまって行けないというようなことが起きます。それからよくあるのは、CIDR になって Aggregate して経路情報を減らして流しているのに、あるリンクからフィルタの設定ミスで、ばらばらの経路が流れてしまったという場合です。ルータの経路選択というのはベストマッチですので長いプレフィックスの方が必ず勝つわけです。ですから、何かわからないけれどもこっちのリンクは混んでいるなというときに、実は細かい経路が漏れていたということがあります。
また、たまにありますが、IX のセグメントの経路を、そこにつながっている AS が BGP でさらに他の AS に流してしまう場合です。あまり影響はなさそうに思いますが、Next Hop の解決で最適経路を選べないということが発生する場合があります。

IX セグメントの経路アナウンス

不正な経路情報の例として、IX セグメントの経路アナウンスの場合について説明します。

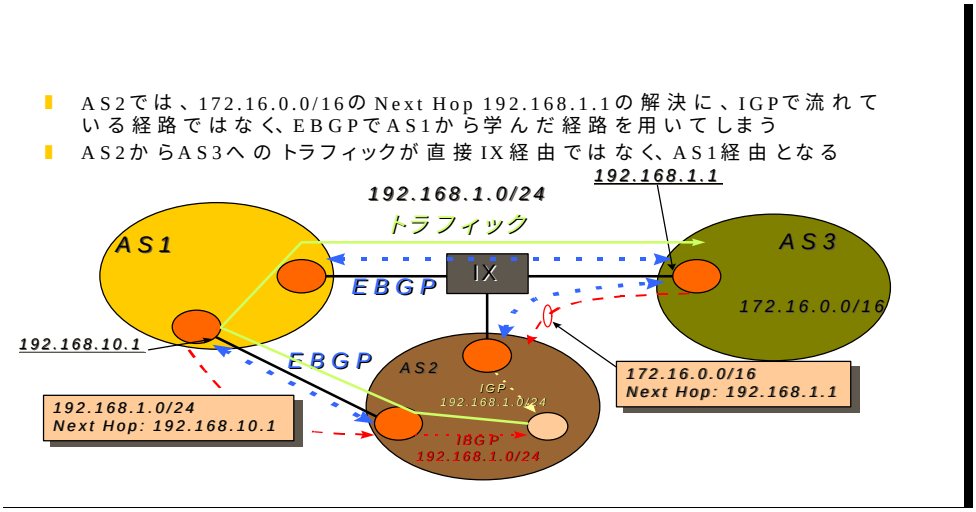


図 5.4.1: IX セグメントの経路アナウンス

AS1、AS2、AS3 が IX 越しに接続していて、AS1-AS3 と AS3-AS2 はそれぞれ peer を張っており、さらに、AS1 と AS2 はダイレクトにリンクを持っています。また、IX のセグメントはアドレス 192.168.1.0/24 を持っています。

この場合、AS3 が 172.16.0.0/16 を EBGP で AS2 に流すと、AS2 では 172.16.0.0/16 というネットワークに対して、Next Hop はルータのアドレス 192.168.1.1 という情報が伝わってきて、中のルータに対してはその情報が IBGP で配られるわけです。一方、IBGP "HACK" をしている場合は Next Hop の解決は例えば OSPF などの IGP を使いますので、ここに流れています。172.16.0.0/16 に対するパケットの Next Hop は 192.168.1.1 であるという IGP で流れている情報によりパケットを流すことができるわけです。

AS1 もこの同様に 192.168.1.0/24 を自分の AS で、IGP で流していますが、何かのはずみで ridistribute か何かしたかによって、BGP でも流してしまったとします。例えば、この場合 AS1 が 192.168.1.0/24 に対して、経路情報をこの自分のルータのアドレスを Next Hop として流してくるわけでが、これは EBGP でやってくると、IBGP で AS2 のルータまで届きます。

そうすると、AS2 のルータには IGP で流れてくる 192.168.1.0 という経路情報と BGP で流れてくる 192.168.1.0 というアドレスの両方が存在することになります。この場合、ルータの実装に依存しますが、Cisco の場合は BGP のほうがアドミニストレーティブ・ディスタンスが低いので、IGP と BGP が流れてきた場合、BGP からの経路を優先してしまうということが起こります。そうすると、AS3 からもらった経路情報 172.16.0.0/16 の Next Hop を解決するときに、BGP でもらった情報を使ってしまい、Next Hop は 192.168.10.1 であるといって、直接流れればいいパケットをわざわざ AS1 に投げてしまうという事故が起こり得ます。

一般的には、不必要な IX のセグメントのアドレスは、自分の中では IGP で流すことはやりますけれども、BGP では他に流してはいけないといわれています。

5.5. プレフィクス・ベース・フィルタリング

・不正な経路アナウンスを防ぐために、プレフィクス単位で Ingress フィルタを設定
　・特に自分が通過を許可している AS からの経路アナウンスに対して
・フィルタの自動生成
　・RR の情報を元にフィルタを生成
　・RR の信頼性は？ (内容、動作)

不正なアドレスを流したり、思わぬ細かい経路を流したり、人の経路情報を自分がオリジネートしたように流してしまうなどの事故が実際にいろいろ起こっています。このような事故を防止するために、最近いろいろな ISP で、プレフィックス・ベース・フィルタリングという方法が使われています。
これは、不正な経路アナウンスを防ぐために、プレフィックス単位でアドレスのアナウンスのフィルタを Ingress Filter に全部書いてしまうというものです。peer を張る AS すべてに対して行うのは大変な作業ですが、先ほどのような事故を完全になくすためには、このようなものが必要になっているのが現状です。
これはもう手作業では不可能なので、ルートレジストリによるフィルタの自動生成が検討されています。ルーティングレジストリのデータベースには、いろいろなポリシー、自 AS のオブジェクト、自分が流すプレフィックスの情報、ネットワークのオブジェクトなどが登録されています。そこで自分と peer する AS からオリジネートするプレフィックスについて、このデータベースを検索すれば全部のリストを作ることができるので、その BGP のフィルタを自動的に update してフィルタを設定しようというものです。
これはスタティックルーティングと良く似ていますが、スタティックルーティングは、スタティックに全部の経路情報をテーブルに登録しますが、これは経路情報が来たら、フィルタに合ったものを経路テーブルに登録する、という点で少し異なります。

5.6. 経路情報のセキュリティー

・いくつかの提案はあるが、未だ発展途上
・アナウンスの信頼性の DNS による認証
　どのプレフィクスをどの AS がアナウンスしてもよいのか
　DNS に AS レコードを追加
　draft-bates-bgp4-nlri-orig-verif-00.txt
・Secure BGP
　IPSEC を用いた peer の認証、メッセージの完全性の保証
　PKI（Public Key Infrastructure)を用いた、経路の生成許可証明と確認
　http://www.net-tech.bbn.com/sbgp/sbgp-index.html

このような事故の反省を踏まえて、プレフィックス・ベース・フィルタリングもそうですが、経路情報自体にもそろそろセキュリティを考えなければいけないといわれています。いくつか提案がありますが、まず一つは DNS に新しくリソースレコードとして AS レコードを追加するというものです。CIDR のプレフィックスを割り当てたときに、AS レコードを使って割り当て情報を DNS 上に登録します。そうするとアナウンスされている経路情報が、どの AS に割り当てられかは、DNS でルックアップできるようになります。そこでルータは、BGP の update を受け取るたびにネームサーバで AS レコードを引いて、オリジネートしている AS が正しいことを確認します。
これについては、DNS-based NLRI origin AS verification in BGP のインターネットドラフトがありますので、興味のある方は読んでみてください。ただ非常に難解です。

また Secure BGP という拡張も考えられています。これは例えば IPSEC の技術を用いて peer の相手の認証をやメッセージの完全性の保証をしたり、Public Key Infrastructure を用いて経路の許可証明を発行したりするものです。これもレファレンスを示すだけにしておきます。

6. まとめ

・BGP4 は AS 間の経路制御の標準プロトコル
　　・経路の選択にはパス属性を用いる
　　・実現できるポリシーは限られている
　　・細かなポリシーの実現にはネットワークトポロジの再考なども必要
・BGP4 を使えばよいというわけではない
　　・使わなくてもよい場合もあれば、使わないほうがよい場合もある
　　・使ったがために運用管理が煩雑になることもある

BGP ができた背景から BGP の概要、パス属性とその使い方、ポリシールーティングとその問題点、そして現状の BGP の問題点について説明しました。
いろいろ問題はありますが、BGP4 というのは AS 間の経路制御を行うためのデファクトのスタンダードです。AS を使った ISP との経路情報交換には、今は BGP4 しかありません。その特徴としてはパス属性というものを用いたりするわけですただし実現できるポリシーは限られていますので、何ができるかできないかを理解した上で、周囲の AS との協議、交渉が必要になります。
マルチホームの場合でも、必ずしも AS 番号を用いた BGP を使用する必要はありません。ですから、マルチホームしたリンクをこう使い分けたいので、これは BGP ではないとできないから BGP を使うために AS 番号をとる、というような考え方が重要です。マルチホームしたから BGP をやらなくてはいけなくて AS 番号が欲しい、ということになると AS 番号は 16 ビットしかありませんので、足りなくなってしまうわけです。
使うと決めた瞬間に、これまでのことを全部理解し、コントロールし、相手の AS と交渉するということが必要になります。使えば楽なるかという、と必ずしもそうではないということも、ご理解頂きたいと思います。

・ポリシールーティングは難しい
　・単なるマルチホームでもトラフィックをうまく複数のリンクに分散することは難しい
　・できる限りシンプルなネットワーク構成が望ましい
・今後も技術開発が必要
　・RR，RS 等の管理技術
　　・RR の情報の UPDATE
・運用技術の確立
　・設定の自動化
　・route flapping 等の問題
・セキュリティ

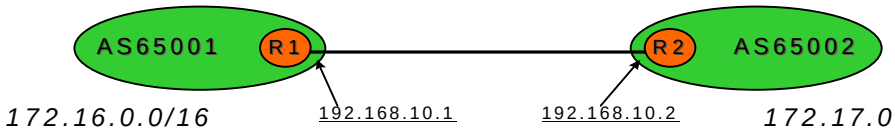
非常に悲しいまとめですけれども、ポリシールーティングとは非常に難しいものです。単なるマルチホームの場合でもトラフィックを複数のリンクにうまく分散することはできないわけです。ですからマルチホームすれば、帯域が増えてリダンダンシーも増えるかというとそうではありません。やみくもに複数リンクを張るよりはシンプルなネットワークトポロジで、シンプルなものを頑丈にしていく方が全体として運用は安定するのではないかと思います。

今後も技術開発が必要で、例えばルーティングサーバやルーティングレジストリというのは、まだまだこれからの技術です。これから問題となるのは、どのようにしてレジストリの情報を更新するか、一貫性を持たせるかということ、またルータの台数が増えていくと全体設定の自動化が必要になります。Route Flapping を今後どう考えて、どう抑制していくかも必要ですし、それから最後にセキュリティという問題もあります。

これだけインターネット中が BGP に頼ってルーティングをしているわけですが、ある意味でこれだけ脆弱なセキュリティでよく動いているな、と驚く場面もあります。ですから、今後このような点もきちんとしなければいけません。そういう意味でやることは、まだまだ一杯あります。

7. 付録: Cisco でのサンプルコンフィギュレーション

Ciscoでの基本設定



R1での設定例

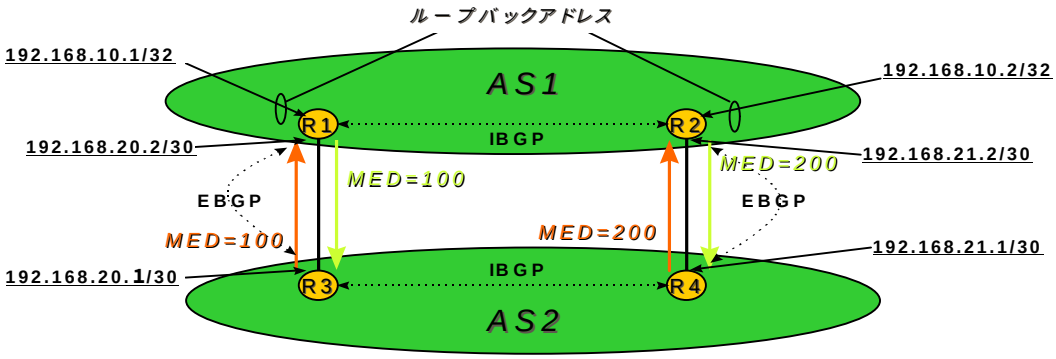
```
router bgp 65001
network 172.16.0.0
neighbor 192.168.10.2 remote-as 65002
```

R2での設定例

```
router bgp 65002
network 172.17.0.0
neighbor 192.168.10.1 remote-as 65001
```

MED

スライド28の例



MED

R1での設定例

```
interface loopback 0
ip address 192.168.10.1 255.255.255.255

router bgp 1
no synchronization
neighbor 192.168.10.2 remote-as 1
neighbor 192.168.10.2 update-source loopback0
neighbor 192.168.20.1 remote-as 2
neighbor 192.168.20.1 route-map MED-OUT out

route-map MED-OUT permit 10
match as-path 10
set metric 100

ip as-path access-list 10 permit ^$
```

R2での設定例

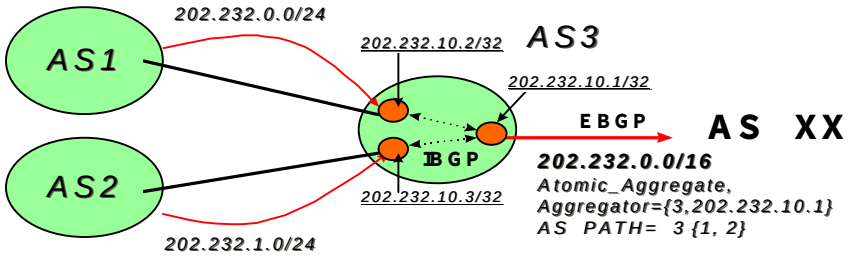
```
interface loopback 0
ip address 192.168.10.2 255.255.255.255

router bgp 1
no synchronization
neighbor 192.168.10.1 remote-as 1
neighbor 192.168.10.1 update-source loopback0
neighbor 192.168.21.1 remote-as 2
neighbor 192.168.21.1 route-map MED-OUT out

route-map MED-OUT permit 10
match as-path 10
set metric 200

ip as-path access-list 10 permit ^$
```

Aggregate

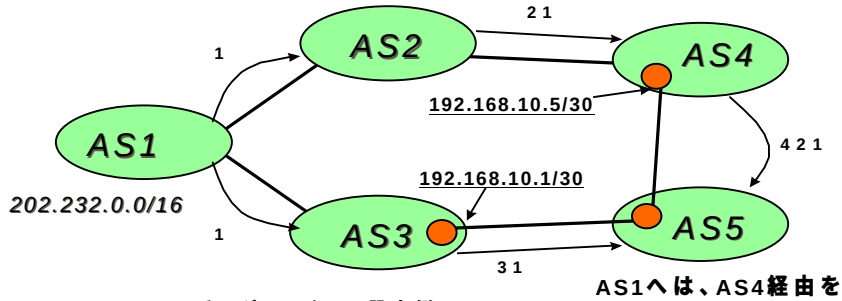


設定例

```
interface loopback 0
ip address 202.232.10.1 255.255.255.255

router bgp 3
no synchronization
network 202.232.10.0
aggregate-address 202.232.0.0 255.255.0.0 as-set summary-only
neighbor 202.232.10.2 remote-as 3
neighbor 202.232.10.2 update-source loopback0
neighbor 202.232.10.3 remote-as 3
neighbor 202.232.10.3 update-source loopback0
neighbor X.X.X.X remote-as XX
```

Local-preference



AS1へは、AS4経由を優先したい

注：ciscoのlocal-preferenceのデフォルト値は100

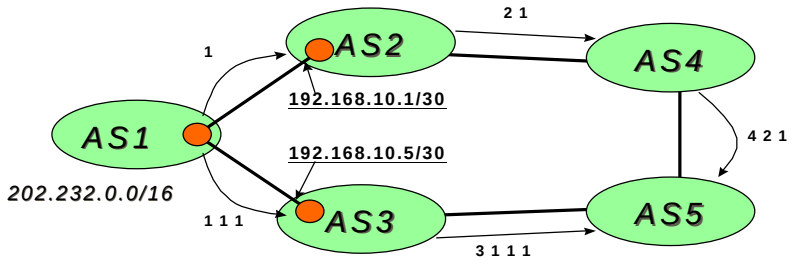
AS5のボーダルータでの設定例

```
router bgp 5
neighbor 192.168.10.1 remote-as 3
neighbor 192.168.10.1 fromAS3 in

ip as-path access-list 10 permit ^3_1$

route-map fromAS3 permit 10
match as-path 10
set local-preference 90
```

AS PREPEND

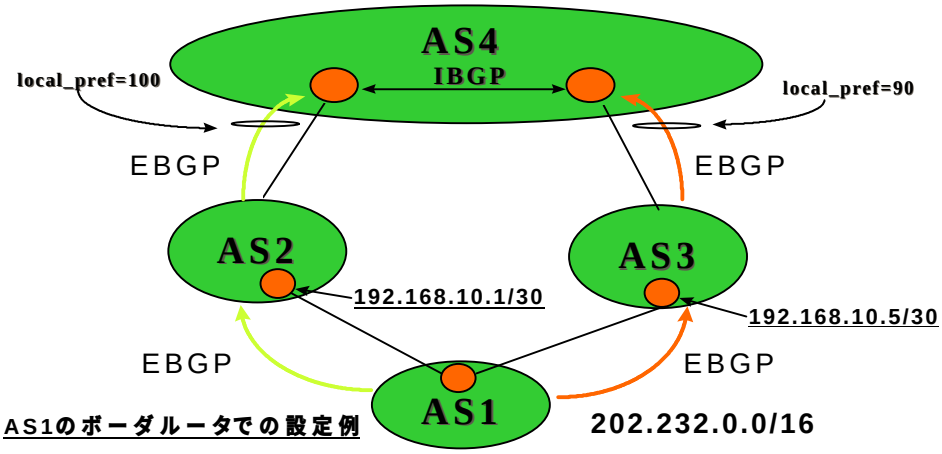


AS1のボーダルータでの設定例

```
router bgp 1
network 202.232.0.0 mask 255.255.0.0
neighbor 192.168.10.1 remote-as 2
neighbor 192.168.10.5 remote-as 3
neighbor 192.168.10.5 route-map PREPEND out

route-map PREPEND permit 10
set as-path prepend 1 1
```

Community



AS1のボーダルータでの設定例

```
ip bgp new-format
access-list 10 202.232.0.0 0.0.255.255

router bgp 1
neighbor 192.168.10.1 remote-as 2
neighbor 192.168.10.1 send-community
neighbor 192.168.10.1 route-map toAS2 out
neighbor 192.168.10.5 remote-as 3
neighbor 192.168.10.5 send-community
neighbor 192.168.10.5 route-map toAS3 out

route-map toAS2 permit 10
match ip address 10
set community 4:100

route-map toAS3 permit 10
match ip address 10
set community 4:90
```