ISPバックボーンネットワークにおける経路制御設計　～実践編～

吉田友哉 yoshida@ocn.ad.jp
NTTコミュニケーションズ㈱

目次

- 全般
- OSPF設計
- BGP設計
- マルチベンダ関連
- セキュリティ関連他

全般

・ネットワーク設計の基本事項
・トポロジー情報と経路情報
・アドレス設計
・N+1設計
・その他

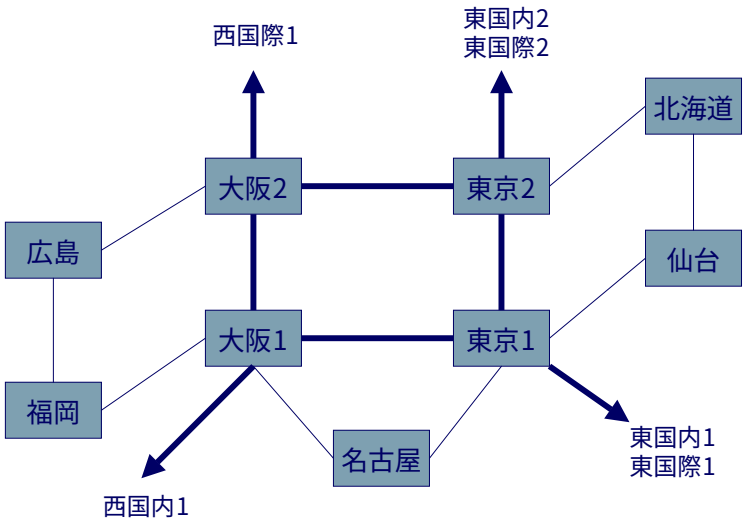
ネットワークの経路制御設計

ネットワークを流れるトラフィックをどうさばくか
→ 必要帯域（ピーク時のトラフィック）の確保と論理設計

- 各POPのトラフィック
  - 地方のPOPのトラフィックは、東京や大阪のメインPOPに接続。あらかじめ設定してある迂回路にて救済
  - そもそもどこがPOPか？（トラフィックの多い地域？）
- 国内ISPとのトラフィック交換
  - 大きなISPとはPrivatePeer、落ちたらIXを利用、もしくはPrivate内で救済。他のISPはIXをメイン。最後は海外トランジットに
- 海外トランジット
  - 均等に複数の上流をうまく使い分ける
  - あるいは、コストの安い上流をメインとし、切れた場合には他に回す等
- 2重故障もある程度考慮にいれて設計するのが望ましい
  - 冗長をとっている2回線とも、という場合にはどうしようもないが、例えば迂回したその先での故障などの場合も考えながら設計することが望ましい

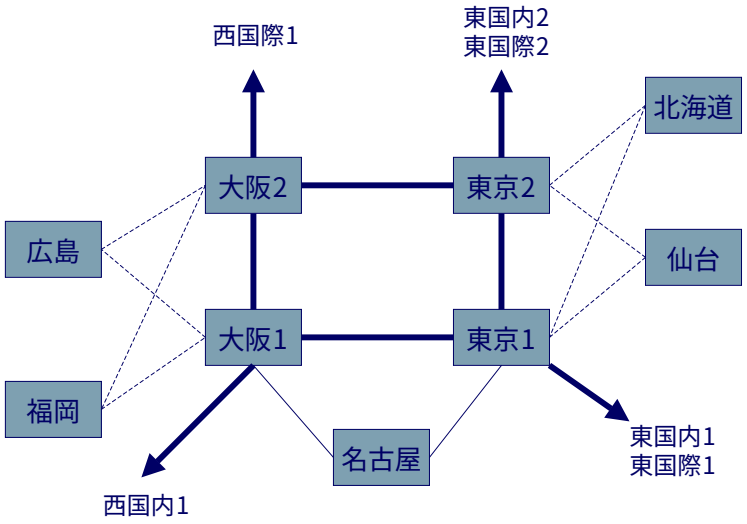
必要な接続性を確保し、トラフィック制御を実施する

ネットワーク設計－例1



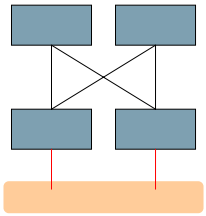
シミュレーションソフトを使って
チェックすることも可能

ネットワーク設計－例2



シミュレーションソフトを使って
チェックすることも可能

ネットワーク設計（基本）

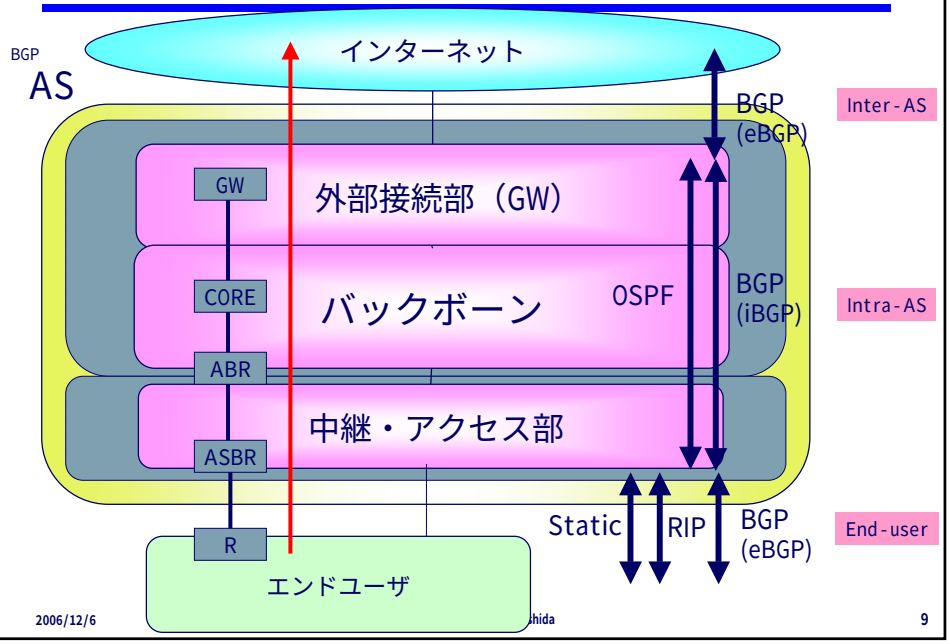


- 信頼性（冗長性の確保）
  - 装置（ノード）、リンクレベルの冗長化、負荷分散
  - ファイバー経路の異経路分散
  - 同機能相当の装置はなるべく分散配備を念頭に
  - 電源系統の分散
- 品質
  - 必要な帯域をきちんと確保する
  - 装置単体、装置間における品質の確保
- 運用性
  - 容易にトラブル対応が可能な、物理的論理的にシンプルな構成
  - 多段構成、HOP数の削減 → 今はルータの性能も上がってきたので、HOP数はそれほど影響しない（実質ナノミリsecオーダレベル）
- 将来性・拡張性
  - 新たなサービスやネットワークの更改等に対応可能なネットワーク

ネットワークの規模・階層的構造

- 中規模・大規模なISPネットワーク
  - 物理ネットワーク
    - 外部から複数の上流経路を受信し、国内のピアも十数以上
    - GWは複数台、それぞれeBGPピア接続
    - 主な地域はPOPになっている
    - COREルータや境界ルータは基本は2重化構成
  - 論理ネットワーク
    - 内部のTopology管理はOSPF、経路情報の管理はBGP（OSPF）
- 階層的構造に沿ったルーティングの設計
  - AS間 [eBGP] — inter-AS
  - AS内 [OSPF/iBGP] — intra-AS
    - 外部接続部（GW）
    - バックボーン
    - 中継・アクセス部
  - エンドユーザ[static/RIP/eBGP] — End-user

階層ルーティングネットワーク全体イメージ



トポロジー情報･経路情報

- トポロジー情報（ネットワークの地図情報）：OSPF
  - バックボーン全体のリンクのつながりを表す情報
  - OSPFのリンクステートデータベース（トポロジカルデータベース）に格納
    - 隣接とLSAを交換し、トポロジカルデータベースを作成
- 経路情報：BGP
  - ユーザの経路情報
    - PAアドレス、上流ISPからの経路情報（フルルート/トランジット経路）
  - 基本はBGPで交換
    - 最近は経路集成はあまり考えなくても大丈夫
  - 以下の場合にはOSPFも有効
    - ユーザ経路を簡単にロードバランスさせたい場合
    - BGPを動かしていないルータから上位に経路情報を渡したい場合

アドレス設計

使用目的別にアドレスを区分け
区分けされた各々のアドレスのaccessabilityを考慮
それらをなるべく経路集成可能なように

- ネットワークの規模が増せば、よりルーティングネットワークに影響を与えるので、なるべく経路は集成可能なように設計する
  - 各POPやABRで集成（例：area-range、 summary-address）
  - ユーザブロックの割り当てプールは連続した割り当てに
- とはいっても、可能な範囲で実施すれば良い
- 可能な範囲で ⇒ 規模相応に
  - ネットワークの規模が大きければ、経路の増大等でルーティングに影響を与えるが、そもそもそのぐらいの大きなネットワークであれば、アドレスもあらかじめある程度豊富に確保可能なはず
  - 逆にネットワークの規模がそれほど大きくなければ、経路も爆発的に増えることもないので、細かく気にしなくても大丈夫

アドレス設計

- 例えば以下ように分類
  - (1) バックボーンアドレス
    - LBアドレス
    - P2Pアドレス、POP間アドレス
    - バックボーンSWセグメントブロック
  - (2) ユーザアドレス
    - ユーザが実際に利用するブロック
  - (3) 外部接続アドレス
    - GWなどで外部と接続する部分のアドレス（実際は (2) に含める）
- セキュリティーの観点
  - Telnet、SSHなどのリモートアクセス範囲の明確化
  - 経路広告の範囲の明確化（DOS対策など）

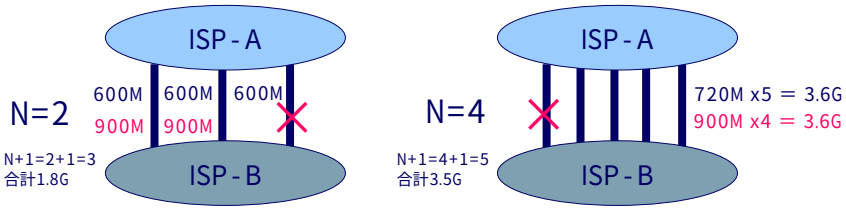
アドレス設計

分類	用途	割り当て	外部への広告	Telnetアクセス
(1) バックボーンアドレス	ループバックアドレス スイッチセグメント point-to-point POP間/POP内セグメント	/32 /27/26 等 /30 /30 等	不要 実際は広告	許可
(2) ユーザアドレス	ダイヤルアッププール DSL、FTTH用プール 常時接続/ハウジング	/24 等 /24 等 /29/28 /24 等	必要	拒否
(3) 外部接続アドレス	プライベートピア・IX接続 トランジット接続 (自ネットワークから相手に払い出す場合には、ユーザアドレスに含める)	/30	不要 実際は広告	拒否

直接ルーティングに必要無いという意味で「不要」。ただし外部からの疎通確認や広告経路が細切れになる等の場合には通常広告する。範囲の明確化自体は必要

N+1設計

- **実際に流れている利用帯域「N」に「＋1」の「N＋1」回線を用意し必要帯域を確保する**
  - **1G～2Gの場合　必要帯域　N＝2　⇒　2＋1＝3本で設計**
  - **3G～4Gの場合　必要帯域　N＝4　⇒　4＋1＝5本で設計**



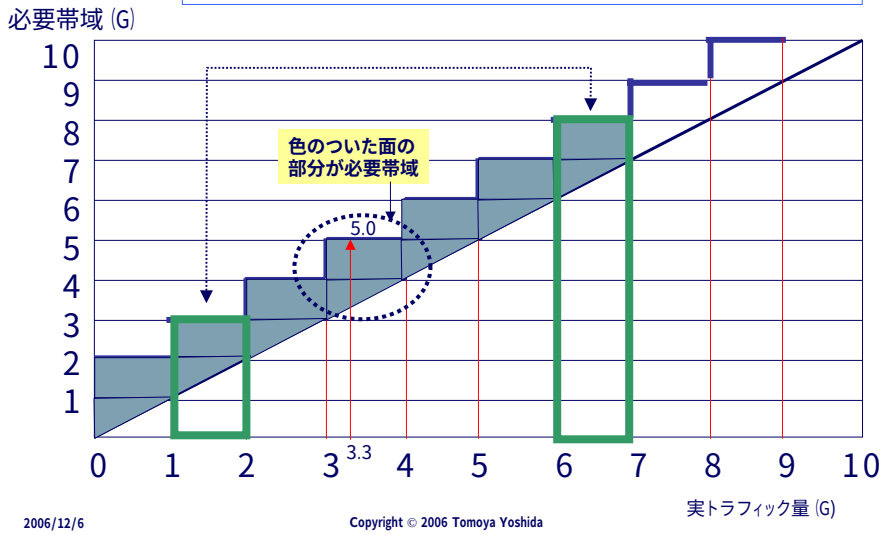
2GE相当のトラフィックに対して、3GEの容量を確保する必要がある
→ 3GEは、2GEの1.5倍の量に相当する

4GE相当のトラフィックに対して、5GEの容量を確保する必要がある
→ 5GEは、4GEの1.25倍の量に相当する

トラフィック量の増加に伴い、回線の有効利用が可能となる。ただしトラフィックのバランシング設計がより複雑になる

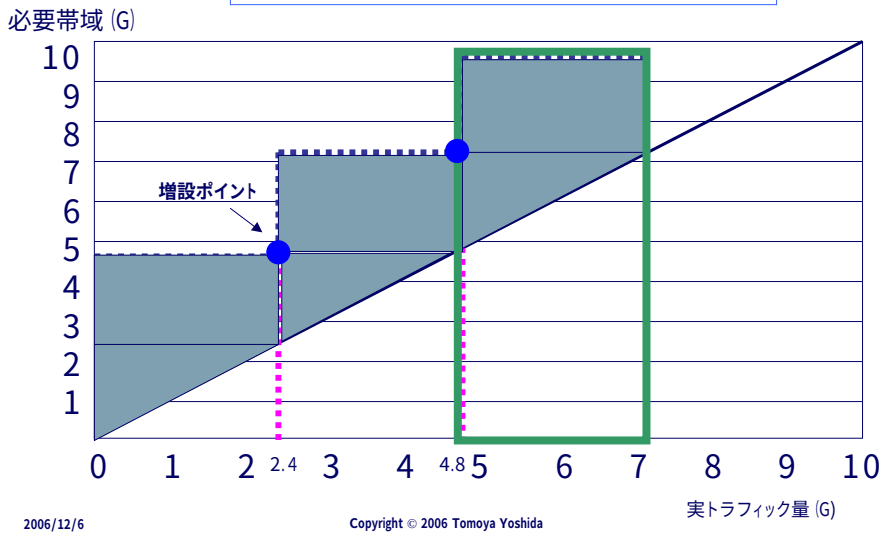
N+1設計　１ＧＥの場合

メリット：　実トラフィック量が増えるほど、効率的に回線が利用可
デメリット：増設ポイントが多い、トラフィック分散設計が大変

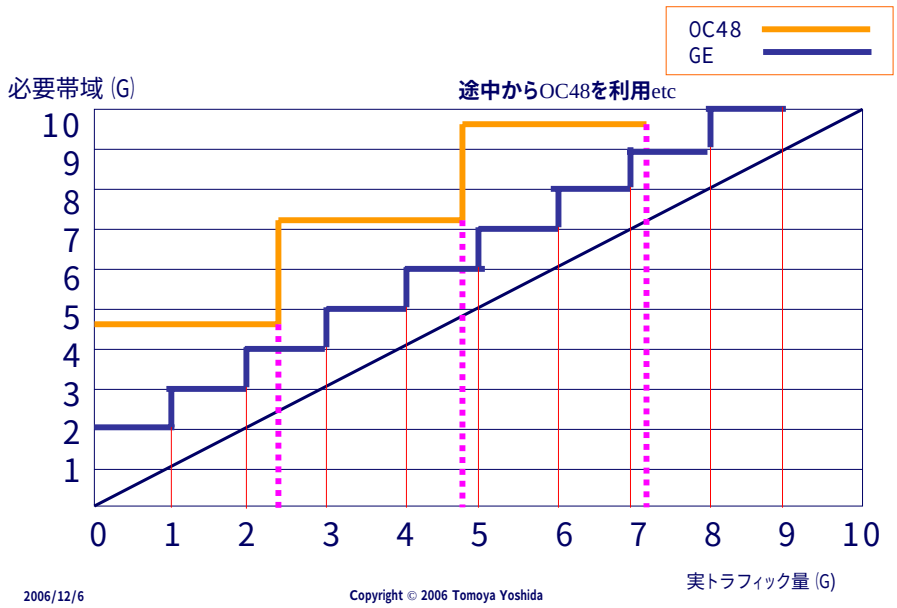


N+1設計　ＯＣ４８の場合

メリット：　増設ポイントが少ない
デメリット：実トラフィック量に比べて必要帯域が多い

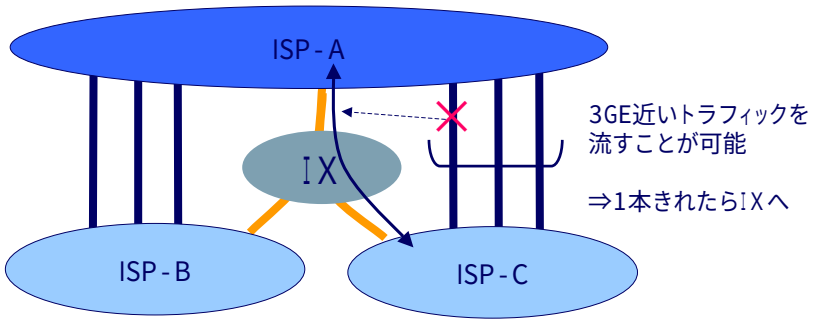


N+1設計　1GE、OC48



回線設計の応用

- IX (Internet Exchange) の回線等を利用し、メイン回線をフルに利用
  - ISP-Aが ISP-B, ISP-C と共にIXで接続していた場合



それぞれ＋1本用意する必要がないので回線の有効利用が見込める

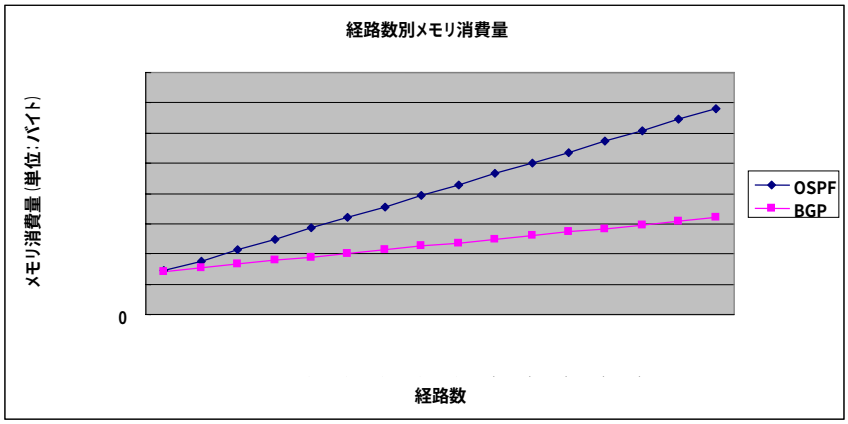
需要予測と回線増設

- 過去から現在までのトラフィック量の伸びのデータをもとに、将来の需要を予測し、プロットした結果を線で結んでみる
- その上で、どの時期までにどのぐらいの帯域を必要とするかを判断
  - 過去3年程度の状況を元に推測するのが良いという説もある
- 実際に回線やファイバーを調達する時間を見込んで、最終的にいつまでに増設の判断をして行動に移さなければならないのか
  メディアの変更を考えるべきなのか（GE x N本 ⇒ 10GE）の判断等
  （例） GEを5本束ねる必要性が出てきた場合、複数ルータで収容する際のルータの収容分散やオペレーション自体も厳しい
  ⇒ 10GEにするか？
  ⇒ 初めから10GE x 2 は厳しい ⇒ OC48 x 4 なら N+1設計で7.2G 等

CPU・メモリ

- 性能が高ければ、それに越したことはない
  - 512M 以上が一般的になってきている
  - どのぐらい必要なのかは、自分のネットワーク環境に近い検証環境をつくってテストする
    - ルーティングエンジンの性能アップで、より効率化されるかも
    - OSPFやBGPの経路数を実網と同じ値、あるいはプラスαの経路でテストを実施

OSPF･BGP メモリ消費量（例）



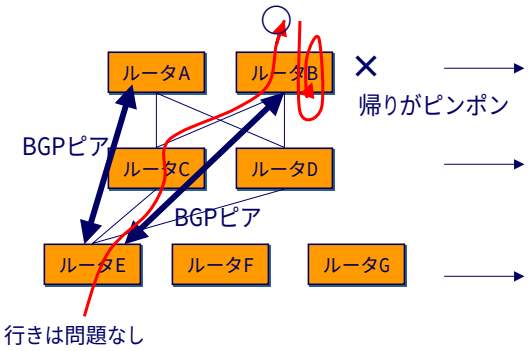
OSや機種によっても、消費量は異なるので、それぞれの組み合わせで
自分にあった環境で検証する必要がある

Loopback（IF/アドレス）

- 装置自体が落ちない限りは生きている仮想インターフェース
  - 通常は/32
- 全ルータに付与するのが望ましい
- OSPFやBGPでは特に重要になってくる
  - OSPFのルータID
    - IDが変わってしまうと、LSAの交換を再度やり直し ⇒ 非常にまずい
  - BGP（iBGP）のピアはloopbackではるのが基本
    - インターフェースでピアをはると、たとえ回線を冗長していても、
      そのインターフェースが落ちると即BGPピアも断になってしまう
    - eBGPから受信した経路のnext-hopにも利用
- ルータへの各種アクセス制御で利用するのが一般的
  - Telnet/SSH access
  - snmp access （MIB、Trap）

論理網と物理網

- ルーティングトポロジーと論理トポロジーの構造は一緒にしておくこと望ましいだろう
  - トラブル時における対応が容易になる
    - このルータが落ちれば、論理的にも落ちる
  - 極端に異なっていると、運用自体が複雑に
    - この場合には、どういう風に経路が流れるんだっけ・・・など



ルータEから下部の経路を
BGPピアにより受信している

ルータCは、OSPFのDefaultに
従ってルーティングしているため、
上位に投げ返してしまう
→ピンポン発生

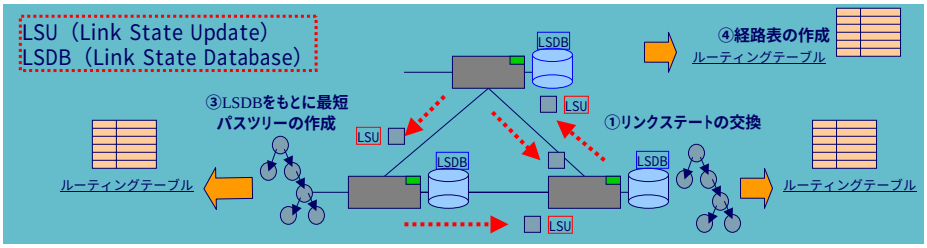
ルータEは、ルータAとルータBに
BGPで経路配信をしている.
上位向けはOSPFのデフォルト

OSPF設計

・エリア設計
・リンクの数
・DR/BDR
・コスト設計
・内部経路・外部経路
・Defaultルートの広告
・経路数
・OSPFの安定性
・その他

OSPFの動きの基本

- OSPFの動き（流れ）
  1. リンクステートパケットを隣接ルータ間で交換
  2. それをもとに、LDSB（トポロジカルデータベース）を各ルータが作成
  3. そのデータベースから、ダイクストラのSPFアルゴリズム（ダイクストラ法）を用いて、自分を頂点とした最短パスツリーを作成
  4. そのツリーをもとに、ルーティングテーブルを作成

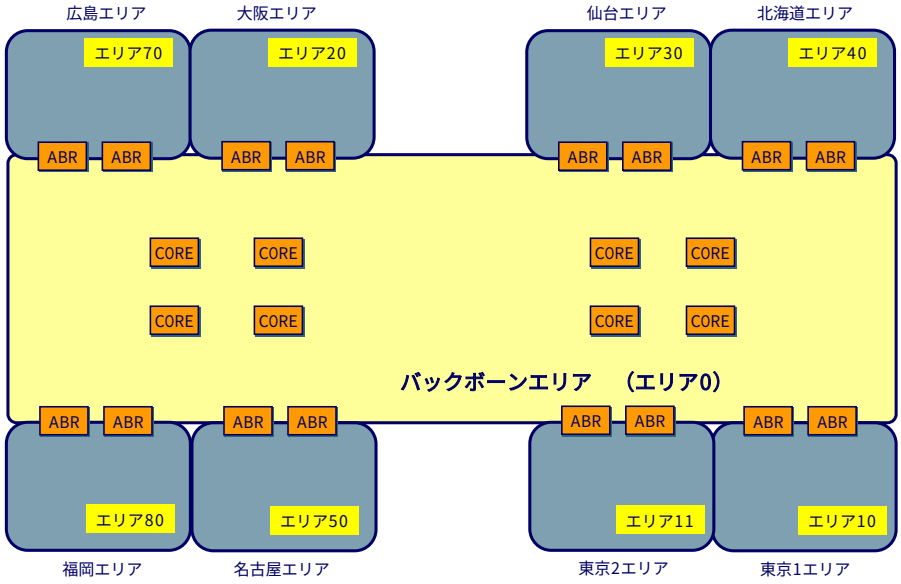


- 自分を頂点としたリンクステート（トポロジカル）データベースをそれぞれのルータが保有しているので、ある個所で障害が発生しても、あらかじめ保持してるLSDBからすぐにそれぞれのルータが再計算可能。収束も非常にはやい
  - RIPなどは、ルーティングテーブルのアップデートを、30秒ごとに隣接へ伝達しているので、その点OSPFは格段に高速化されている

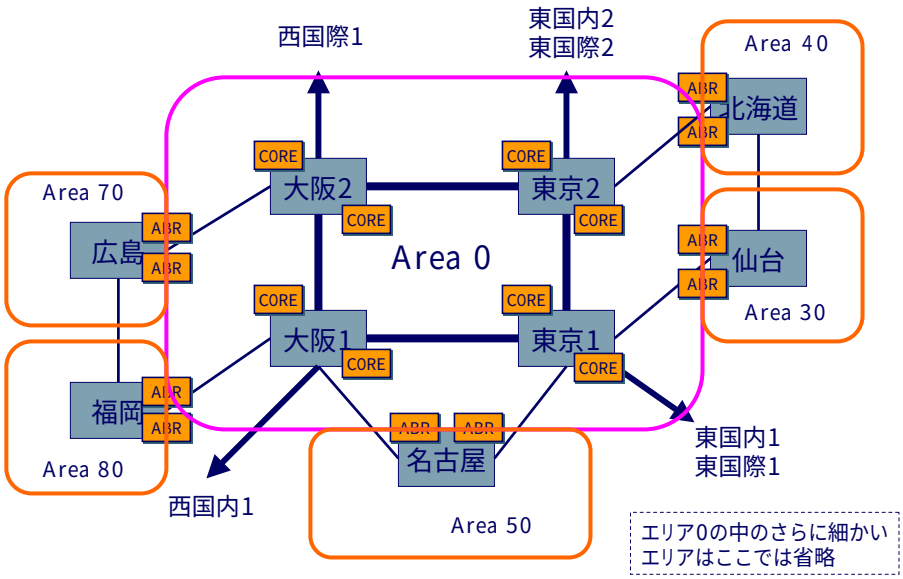
エリア設計

- まずは、エリア0（バックボーンエリア）を中心に考える
  - どこをエリア0にすればよいか？
    - 鉄道を例に考えると、新幹線の走っている主要な駅をエリア0
    - それ以外の、ローカルな路線エリアは、エリア0にぶらさがる各エリアとする
    - ネットワークのコアとなる部分がエリア0となる
    - エリア0以外のエリアは、全てエリア0を介して接続する
    - エリア0に各エリアがぶら下がるようなスター型構成が基本
- むやみにエリアは増やさない
  - エリア0はどんどん肥大化していくので注意が必要
  - エリア分けをする必要がなければ、あえてしない
- 1エリアにはABR（エリア境界ルータ）は2台（以上）
  - ABRが落ちると、そのエリアが全滅という状況は絶対に避ける

エリア設計



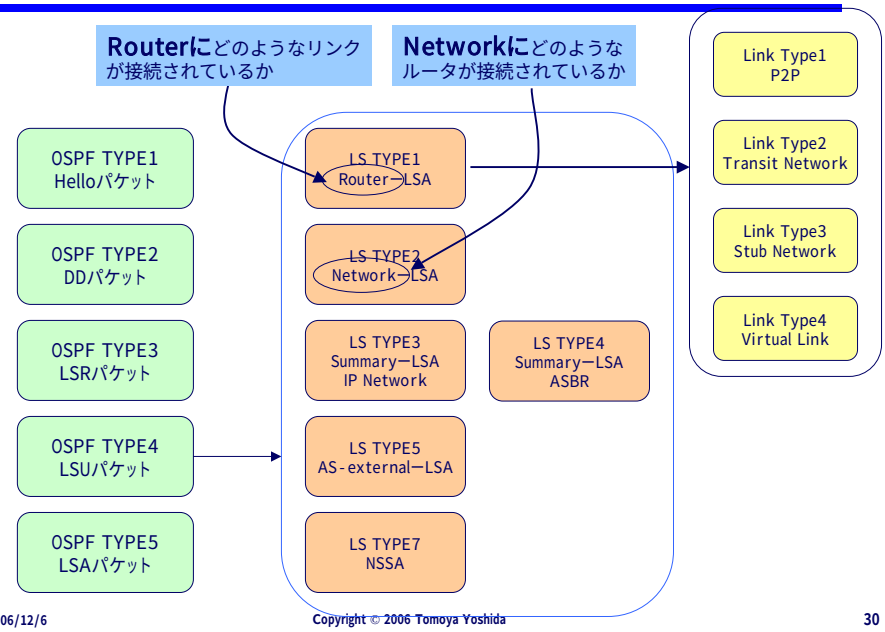
エリア設計



1つのエリアに置けるルータの台数

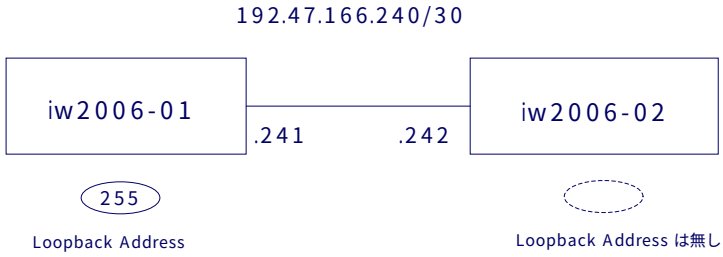
- 一概には言えませんが、、
  - ネットワークのTopologyやリンクの数などにかなり左右される
  - 数十台程度なら、大抵1エリアでおさまるだろう（経験上）
    - ただ、これもあくまで例で、それぞれ事情は異なる
  - OSPFの収束時間が以前に比べて長いと感じている場合
    - エリア分割や、エリア0の台数削減などの対応が必要
  - ルータの性能は侮れない
    - 処理能力の高いルータと低いルータでは、随分と差がある
  - 参考書や文献は、あくまで指標にすぎない（結構古い）
    - Halabi：50台までだろう．60台や70台は避けるべき
    - Moy：1991年に多くて200台といったが、ベンダによっては、350台というところもあるし、50台やそれ以下というところもある
  - 実際には、色々動かしながら試行錯誤していく
  - エリア0の肥大化には注意

OSPFパケットの種類（おさらい）



⑴例⑵ OSPFのLSA

- OSPFの各LSAの実際を見てみましょう
- ⑴例⑵ 以下2台のルータが単純にetherで接続されている場合



トポロジーのバランス

- point-to-pointとSWセグメントをバランスよく
  - むやみにpoint-to-pointのフルメッシュなどを増やすと、LSAが肥大化してしまう可能性もある
    - そのルータにはどのようなリンクがつながっているのか
    - 1つのルータに属する同一エリアのリンク数が多いと、1つのRouter-LSAパケットに含まれるリンクの数が多くなり、肥大化
  - SWセグメントについては、DRがNetwork-LSAを生成
    - ネットワークには、どのルータがつながっているか
    - パケットフォーマットが単純で、DRがそのネットワーク内でneighborとなる各ルータをattachしていく

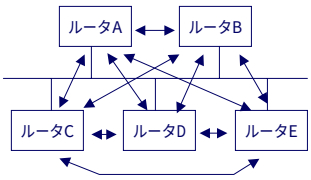
OSPFv3

- 大抵のalgorithm等はOSPFv2を継承。ネットワークの設計も概ね同じと考えてよいだろう（規模相応に）
- OSPFv2との違いは、RFC2740 Section2を参照
  - IPv6のアドレス空間の拡張を考慮して、よりsimpleな設計へ
  - Authentication フィールドの除外 → IP_AH、IP_ESP
  - Neighbor 等はlink-local-addressを適応
  - Link-LSA（type8）の追加 → local-link での flooding
  - Intra-Area-Prefix-LSAの追加
    - Router-LSAs and Network-LSAs を運ぶ
  - LSAの名前の変更
    - Type-3 summary-LSAs → Inter-Area-Prefix-LSAs
    - Type-4 summary-LSAs → Inter-Area-Router-LSAs
  - Link state ID
    - 単純に、同一ルータで生成される複数のLinkStateパケットを区別
    - 例えば、最初のlink-state-id = 0.0.0.1、2番目 = 0.0.0.2 といった感じになっている

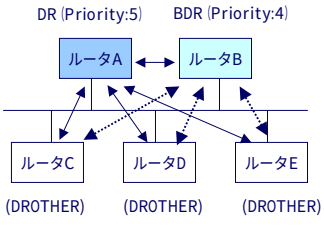
DR/BDR の設計

- DR/BDRは、処理能力の高いルータ、もしくはそれほど仕事をしていないルータにやらせるのが望ましい
- 絶対にDR/BDRにしたくないルータは、Priorityをはじめから0にセットしておく（priority=0の場合、DR/BDRには一切ならない）

DR/BDRがない場合



DR/BDRがある場合



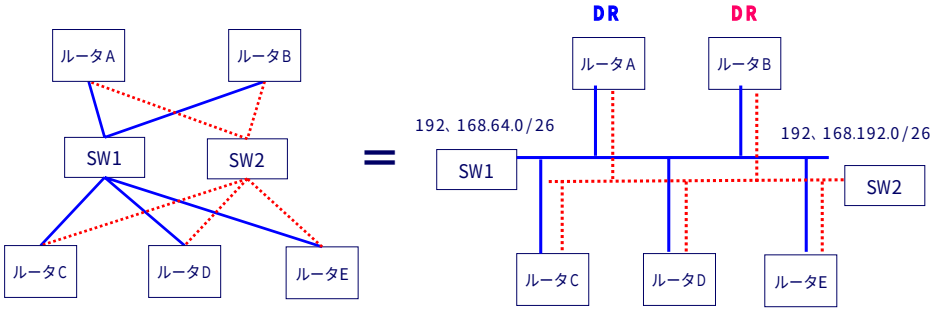
Ciscoの場合には、priority = 1 がデフォルト

Priorityが低くても、最初に立ち上がったものがDRになってしまうので注意

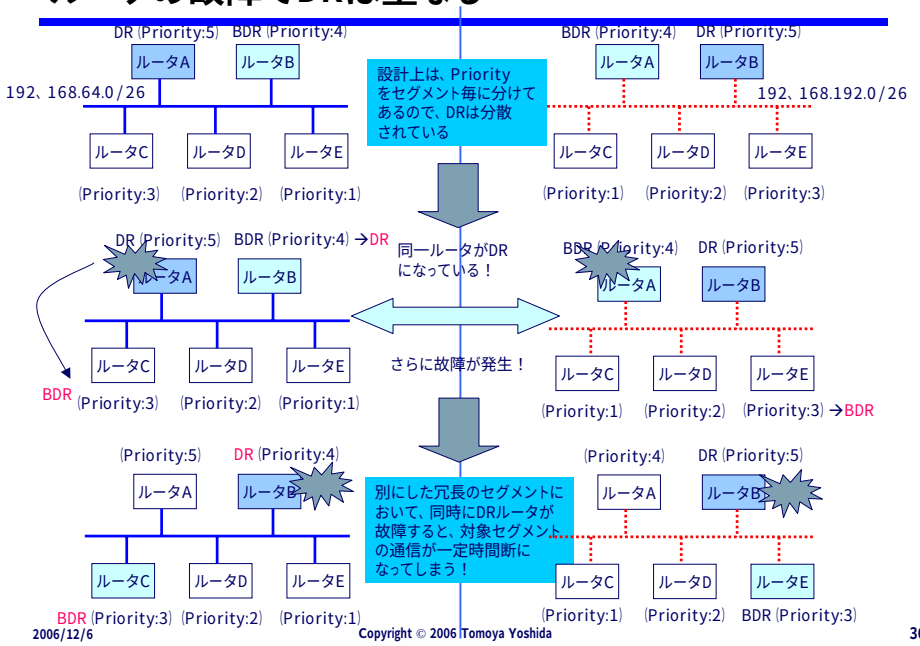
DR/BDR の設計

SW1、SW2 で冗長構成を組んでみる

- DR/BDRを、各々のSWセグメントでうまく付与したい
  - SW1のセグメントでは、ルータAをDR
  - SW2のセグメントでは、ルータBをDR



ルータの故障でDRは重なる



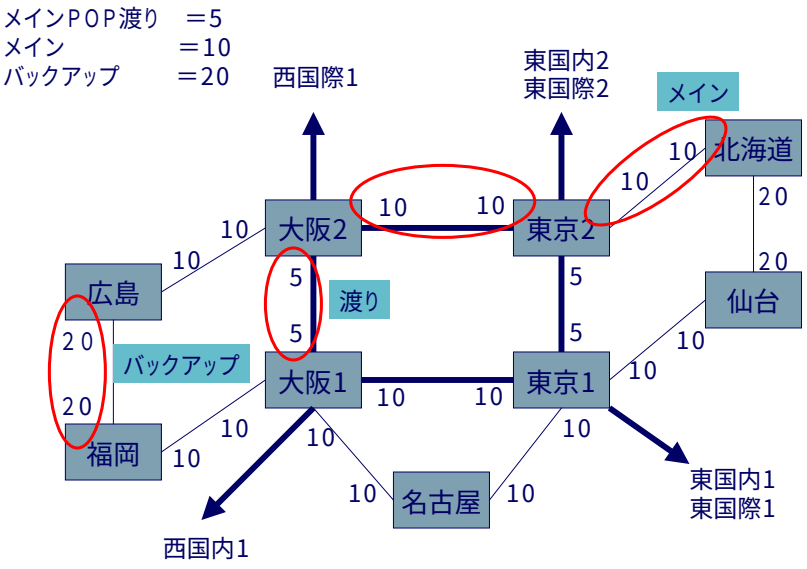
コスト設計（1）

- ネットワークの設計ポリシーを定める
  - どのリンクを通常時にメイン回線として利用するのか
  - イコールコストマルチパス（ECMP） or 0/1
  - 故障時におけるバックアップ（回線断、ルータ断、POP倒壊）
  - あらゆる救済パターンをシミュレーションする
- メイン回線を小さく、バックアップをそれよりも大きな値に
  - あまりにも値がかけ離れていると、ぐるっと回ってしまう
  - 緊急避難時に一時的な迂回を考慮し、微調整が可能なよう各々の値はあらかじめ余裕をもった設計にしておく
- ネットワークのトポロジーが複雑だと非常に設計が難しくなるので、シンプルなトポロジー構成、シンプルなコスト設計を目指すことが望ましい
- ある程度体系的なポリシーを定める
  - 当てはまらない場合には微調整

例）

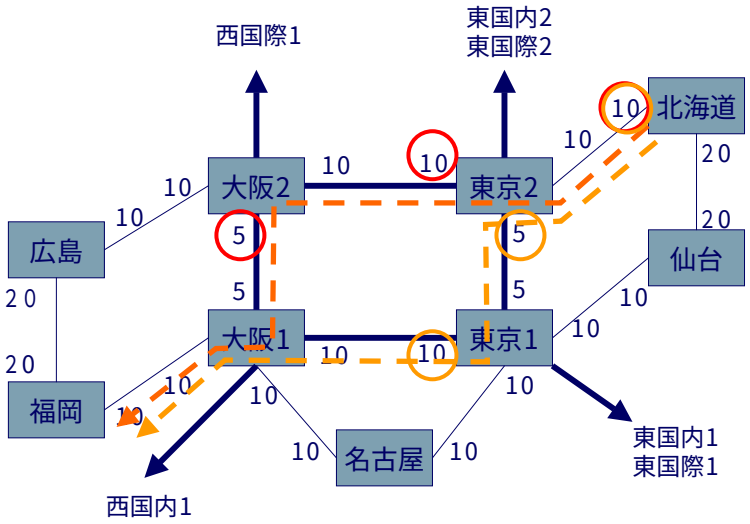
渡り接続回線：	：5
メインの回線：	：10
バックアップの回線：	：20

コスト設計（2）



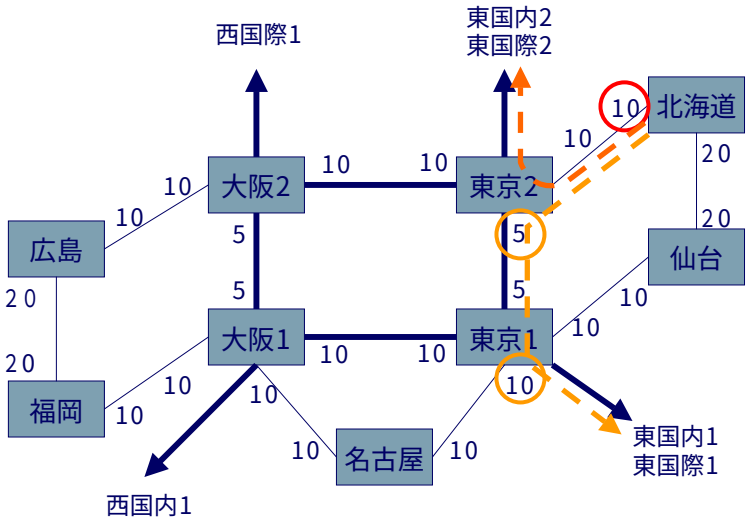
コスト設計 (3)

北海道から福岡への通信
→東京･大阪のスクエア部分は異経路分散、大阪１から福岡へ



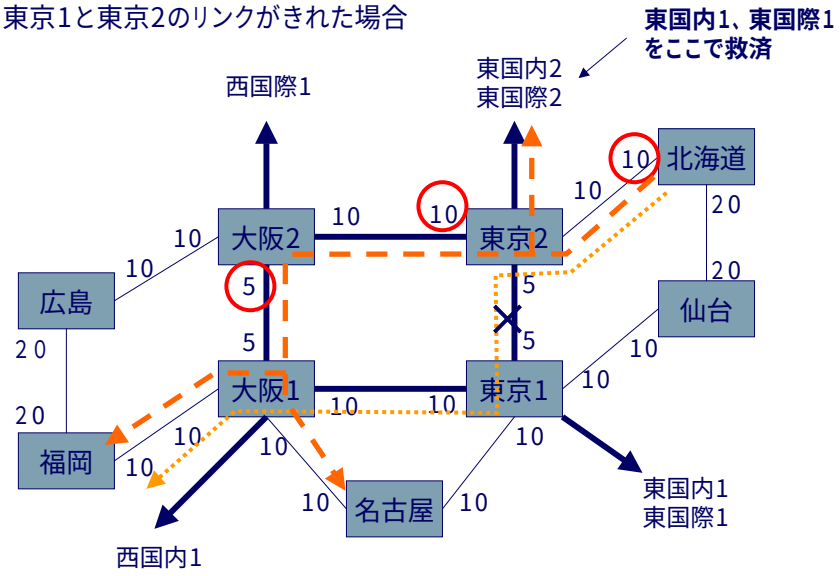
コスト設計 (4)

国際、国内通信は、それぞれ東京１、２より抜けていく



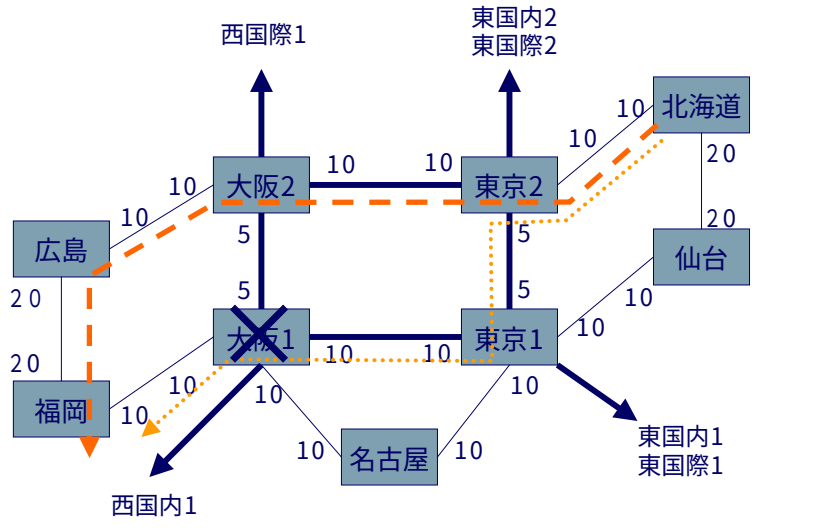
コスト設計 (5)

東京1と東京2のリンクがきれた場合



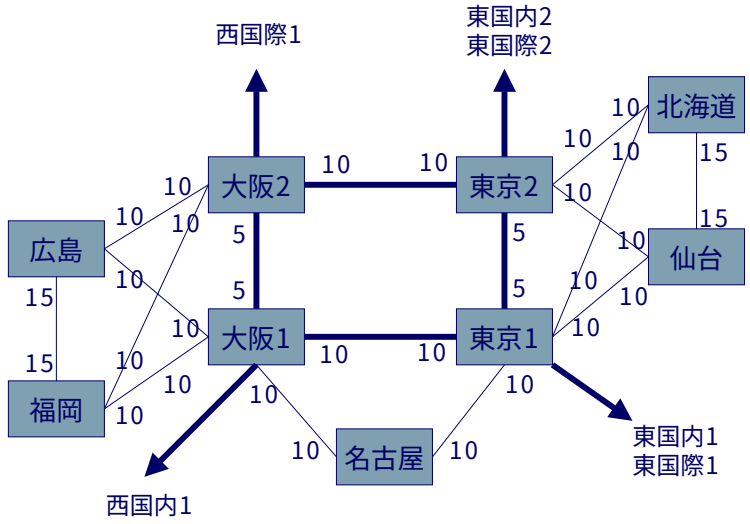
コスト設計 (6)

大阪1が崩壊 → 大阪2から広島経由で福岡へ



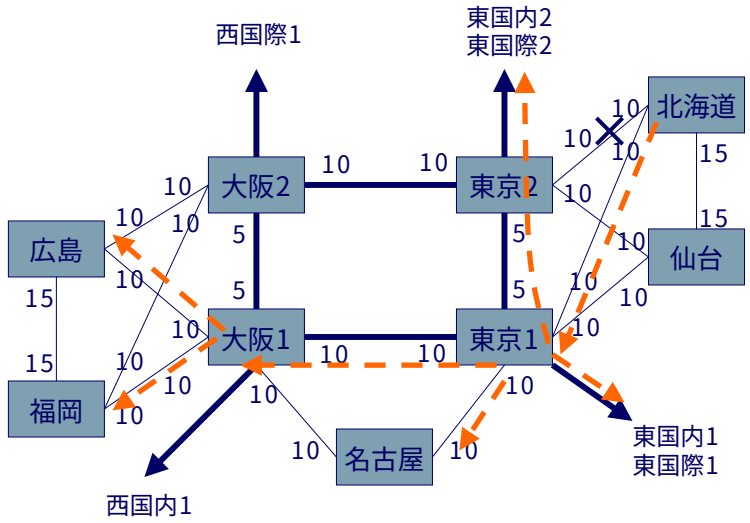
コスト設計（7）

地方のＰＯＰの物理回線を各メインのＰＯＰに冗長化した際の設計例



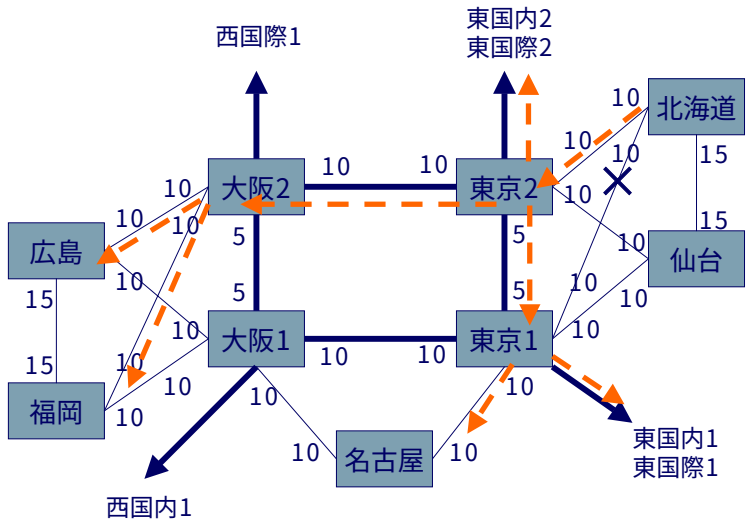
コスト設計（8）

西日本へは、大阪1経由で通信、東京１ー東京2が増加



コスト設計（9）

西日本へは、大阪２経由で通信、東京１ー東京2が増加



コスト設計（まとめ）

- シンプルな設計
  - 役割に応じた値（ポリシー）を決める
  - 行きと帰りはなるべく同じ値に（わざと負荷分散する場合もある）
  - 故障時に複雑な救済経路はとらない（運用性の考慮）
- 単なるコスト設計ではない
  - 物理トポロジーと回線設計が密接に関係
  - BGPのIGPコストに利用される
    - ある回線が切れた場合に、BGPの該当経路へのIGPコスト値が変化するため、exit point も変わる可能性がある
      ⇒BGPのポリシーとも密接に関係してくる
- 想定範囲外の事態が発生した場合
  - その都度見直しを実施するしかない

OSPFの内部経路･外部経路

- 内部経路 (Internal経路)
  - OSPFのトポロジーデータベースを構築し、それをもとに経路計算を実施する
  - 全てがネットワークの地図 (トポロジー情報) 把握することになる為、多くなればなるほど再計算のる際にルータの収束に影響を与える
- 外部経路 (External経路)
  - Internal経路のように、複雑な経路計算はしない
  - 経路に変化があった際にも、OSPFデータベースの再計算を行わないため、負荷は軽い

OSPF内部経路

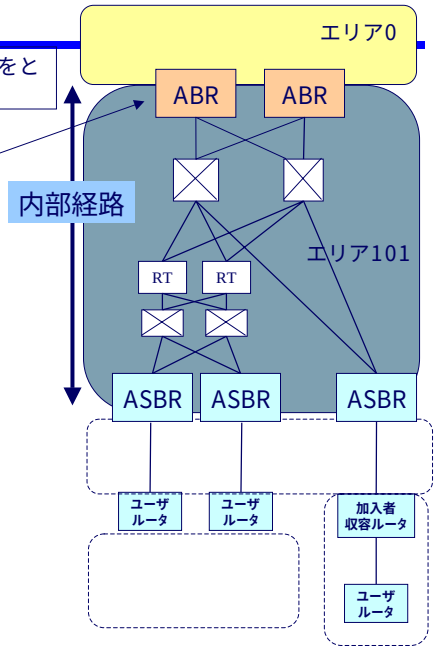
ASBRから上位は、トポロジーの冗長構成をとるためInternal経路である事が必須

■ Ciscoの場合

```
router ospf 100
 area 0 authentication
 area 101 authentication
 network 172.16.32.10 0.0.0.3 area 0
 network 172.16.32.14 0.0.0.3 area 0
 network 10.0.255.129 0.0.0.0 area 101
 network 10.101.1.64 0.0.0.15 area 101
 network 10.101.1.80 0.0.0.15 area 101
```

■ Juniperの場合

```
protocols {
 ospf {
    area 0.0.0.0 {
        interface so-0/1/0.0;
        interface so-1/1/0.0;
    }
    area 0.0.0.101 {
        interface lo0.0;
        interface so-2/1/0.0;
        interface so-2/2/0.0;
    }
  }
}
```



OSPF外部経路

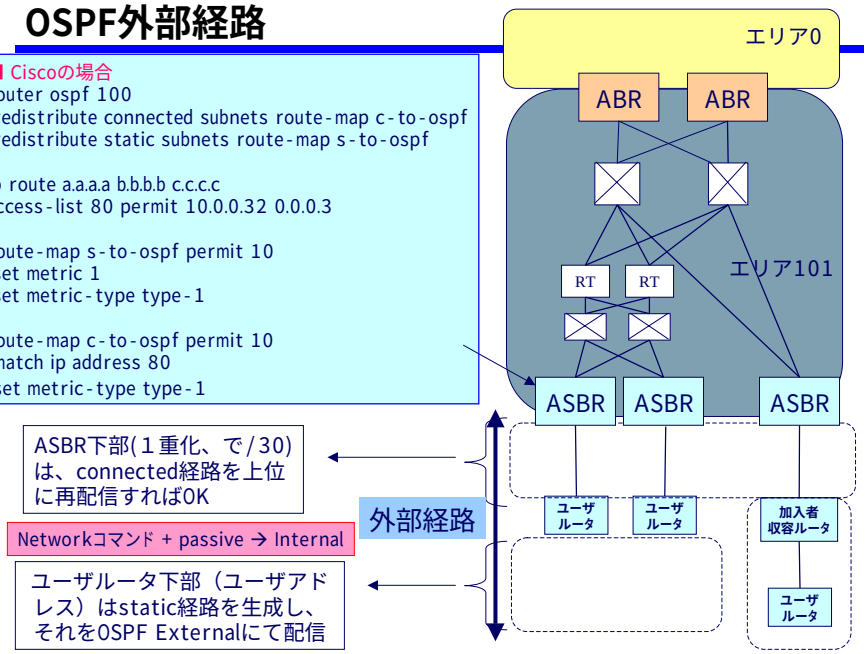
■ Ciscoの場合

```
router ospf 100
 redistribute connected subnets route-map c-to-ospf
 redistribute static subnets route-map s-to-ospf

ip route a.a.a.a b.b.b.b c.c.c.c
access-list 80 permit 10.0.0.32 0.0.0.3

route-map s-to-ospf permit 10
 set metric 1
 set metric-type type-1

route-map c-to-ospf permit 10
 match ip address 80
 set metric-type type-1
```



ASBR下部(１重化、で/30)は、connected経路を上位に再配信すればOK

Networkコマンド + passive → Internal

外部経路

ユーザルータ下部（ユーザアドレス）はstatic経路を生成し、それをOSPF Externalにて配信

OSPF外部経路

- Type 1　- リンクコストと同様に加算される
  - 同じ宛先のType 1外部経路があった場合、途中リンクのコストも加算して、もっとも小さなコストの経路が選ばれる
- Type 2　- とにかく小さな値が選ばれる
  - 同じ宛先のType 2外部経路があった場合、もっとも小さなType 2メトリックの経路が選ばれる
  - 同じType 2メトリックの場合、転送先アドレスまでのコストがもっとも小さな経路が選ばれる（フォワードメトリック）
- 同じ宛先のType 1とType 2の外部経路があった場合、Type 1の経路が選ばれる

OSPFのデフォルトルートの広告

○デフォルトルートの広告とは・・・

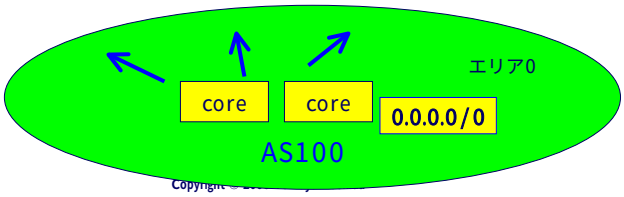
フルルートを保有していないルータが、フルルートを保有しているルータにルーティングできるように設定するもの

★パケット破棄能力にすぐれた中核のルータ等から配信するのが望ましい
→　宛先のない経路に対してのパケットは全てデフォルトに向かってくる！

★BGPのフルルートなどが必要ない部分は、デフォルトルートを活用すべし

■ Ciscoの場合

```
router ospf 100
 default information originate always metric-type 1 metric 5
```



OSPFのデフォルトルートの広告

■ Juniperの場合

```
protocols {
 ospf {
      export DEFAULT-ORIGINATE;
      }
}
policy-options {
   policy-statement DEFAULT-ORIGINATE {
      term 1 {
         from {
            protocol static;
            route-filter 0.0.0.0/0 exact;
         }
         then {
            metric 5;
            external {
               type 1;
            }
            accept;
         }
      }
      term 999 {
         then reject;
      }
   }
}
routing-options {
   static {
      route 0.0.0.0/0 discard;
   }
}
```

→ Protocol、OSPFの部分で、何をexportするのかを定義する．ここでは、「DEFAULT-ORIGINATE」

→ 「DEFAULT-ORIGINATE」の中身を定義

protocol が static で
0.0.0.0/0 に exact match した場合のみ
metric 5、external type-1 で広告

それ以外は、reject

→ Static route の生成
→ discard = null0

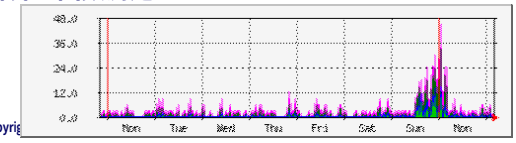
OSPFの安定性

- どの程度の規模まで現状のまま耐えられるか？
  - ルータの機器、メモリ量、CPU、ネットワークのトポロジーなど、色々な要素が関係するため、case-by-caseというのが正直なところ
  - 検証をするにしても、何十台もルータをかき集めて同じ環境を作ってやるのは不可能

  - ある程度経験則を頼りに設計し、実網を監視していくしかない
  - 参考ドキュメント（かなり古い）
    - OSPF Anatom of an Internet Routing Protocol
      - J. Moy (January 1998) RFC著者
    - OSPF DESIGN GUIDE
      - Bassam Halabi (April 1996)
        - インターネット・ルーティング・アーキテクチャーの著者

OSPFの安定性

- LinkStateパケット交換で負荷がけっこうかかる
  - neighborが確立されるのに時間がかかる
  - show (ip) ospf neighbor で見ても、DRとBDRに対して、Statusがしばらくfullにならない等
- 何故か不安定な事象がおこっている
  - Dead timer 値が30秒をかなり下回っていることが多い・・
    - 10秒ごとにHELLOをなげているので、落ちているということになる（別の原因かもしれない）
  - External経路の増大に注意
  - バグの場合もある
  - 疑問に思ったら、ベンダやメーカに問い合わせをしましょう
- 普段からの各エリアにおける状態を確認
  - MIBによる、OSPFの再計算の回数測定
  - MRTG等でグラフ化



不安定事象の解決策

- 機器の性能をUpgradeしてみる
  - バージョンアップやメモリ増設で、劇的に改善される場合もある
  - なるべく、メモリをつんでおくのは悪いことではない
- 1エリアの台数を削減したり、リンクを減らす
  ⇒ LSDBの縮小化
  - 一定の性能のルータを並べている場合には、1台の大容量なルータに集約してしまう、あるいは帯域を太くしてまとめて行く
- 他の方式を検討
  - むやみにOSPFにのっけている人は、BGP化する
    ⇒ static-to-bgp
  - その他
    - Confederation
    - IS-IS化
    - OSPFのプロセスを分ける

MTUサイズ

- DDパケット中の「Interface MTU」サイズ
  - このサイズが一致していない場合、EXCHANGEから先に進まない事象が起こる可能性がある
  - MTU不一致を無視することも可能
    - (cisco) ip ospf mtu-ignore
- きちんとあわせることをお勧めします

その他

- エリアの表記
  - エリア0に関しては、0と表記すれば、自動的に0.0.0.0と解釈されるが、エリア1と書くと、ベンダによっては、
    - Area 0.0.0.1 (ベンダA)
    - Area 1.0.0.0 (ベンダB)

    の2通りの解釈があるので要注意
- ABRで、loopbackはどっちのエリアに属したらよいの？
  - エリアの中にいれておくのがいいでしょう
    - エリア0の孤立時に、通信断になってしまう
    - ＃エリアが分断されるようなケースは絶対させる

OSPF設計まとめ

- エリア設計
  - Area0を中心に設計し、序所に拡大していく
  - 1エリアに配置するABR (エリア境界ルータ) は、2台がよいでしょう
  - 1エリアに何台置けるかは、一概には言えない
    - ルータの性能やそれぞれのネットワークにおける挙動は異なる
    - CPUが落ち着くまでの時間が肥大している場合は、台数削減などを検討する
- リンク数
  - むやみに増やすような設計は避ける
  - point-to-point とSWセグメントをバランスよく配置
- メモリ
  - OSPFはBGPよりも消費量が多いので注意が必要
- DR/BDR
  - DRルータは、相応の負荷がかかるので、そのセグメントにおいて処理の少ないルータや、処理能力の高いルータにやらせるのが望ましい
  - SWセグメントでは、同一ルータが、同じ冗長構成をとっている別SWセグメントのDRを兼任しないようにpriorityの設計をする
    - 運用での修正 (DRがかさなった場合には、interfaceの開閉で対応可能)

OSPF設計まとめ（続き）

- コスト設計
  - 迂回路も含め、どのようにトラフィックをさばくのか、まずはポリシーをしっかりと決めることが大前提
  - あまり複雑な値や経路にはしない
  - 基本は、行きと帰りの経路を一緒にして、運用やトラブル時の対応をなるべく簡易にするのが望ましい
- 経路/経路数
  - 各々のエリアで適切に処理可能な程度の経路数に
  - External経路でも、それなりに数が多くなってくると不安定要因となるので注意
- デフォルトルート
  - デフォルトルートで用が足りる部分は、うまく活用しましょう
  - パケット破棄に強いルータを選定しましょう
- 何かおかしいと思ったら
  - 機器のUpgradeを検討
  - メーカやベンダへ問い合わせる
  - 場合によっては、他の方式を検討
- 運用
  - 日頃から、MIBなどを用いて観測しておく（経路数なども）
- OSPFv3
  - 概ねIPv4と同じと考えればよいが、変更点に注意しながら、規模相応に設計する

BGP設計

- BGP設計の基本事項
- BGPポリシー設計
- iBGP設計
- その他

BGPポリシー設計

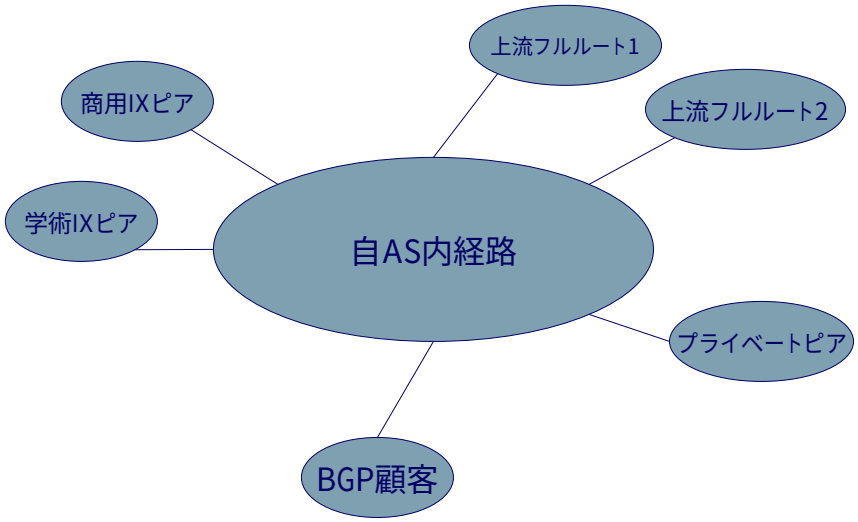
- AS内、AS間において、どのようなポリシーで最適に、且つスケーラブルにBGP経路を配信させるか
  - どの外部ASから何の経路を受信し、どのような優先性を与えるか
    ⇒「受信ポリシー」
  - どのピア先に対して、何の経路を、どのように広告するのか
    ⇒「広告ポリシー」
  - 自AS内経路は、どうやって配信するのか
  - 外部から受信した経路はAS内部にどのように伝播させるのか
    - iBGPをフルメッシュにはるのか？リフレクタの階層構造を用いるのか？
  - AS内全体には、どのようにBGP経路を配信するか
    - COREやGWの必要保有経路は？ABRはフルルート必要？
    - BGPユーザの階層では？
    - 非BGPユーザの階層では？
    - 細かいことを考えずに、全てにフルルートを配信しても問題はない（性能依存）

BGPポリシー設計

- 受信ポリシー
  - 相手から経路を受信する際に、何の経路をどのように受信するのか
    - 複数の上流をどう使い分けるか
    - 国内のピアはどういったポリシーで制御させるのか
      - プライベートを優先？IXと同じ位置付けにする？複数回線で接続されていた場合には？切れた場合にはどこで救済？東西の制御方法は？
    - どういったパスアトリビュートを付与して経路制御をするか
  - 不必要な経路を広告されてきた場合にはどうする？（全体のポリシー）
    - GWを含めたエッジでFilterをかける？
    - Filterに加えて、仮に受信したとしても、該当経路が優先されないようなBGPの制御をきちんと内部でかけるようにする？
- 広告ポリシー
  - 自分の経路やBGP顧客などの経路を配信する際に、何の経路をどういう重み付けで、どういうパスアトリビュートを用いて広告するのか
    - あまり常時使用したくないリンクに対しては、Prependをかませる？
    - Prefixを分けて、回線ごとにトラフィックをさばく？

BGPポリシー設計（受信）

BGPポリシー設計（受信）



BGPポリシー設計（受信）

■以下の接続形態を考える

BGP顧客経路
自AS内広報経路
プライベートピア経路
商用IXピア経路
学術IXピア経路
上流フルルート1
上流フルルート2

基本は、「接続形態に対して、
LOCAL_PREF属性を適用し、それでは
強すぎる場合には、MED属性を用い、
この2つを組み合わせて制御する」

値づけは余裕をもって設計する必要あり
（ルートマップのinstance番号や
OSPFのコスト値などと同じ）
→ 新しい接続形態が増えた場合
→ 値を整理したい場合

```
route-map ebgp-out permi 10
 match as-path 3
 set metric 100

route-map ebgp-out permi 20
 match as-path 4
 set metric 200
...
```


BGPポリシー設計（受信）

接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	110	120	3
商用IXピア経路	300	200	210	220	4
学術IXピア経路	300	300	310	320	5
上流フルルート1	200				6
上流フルルート2	200				6

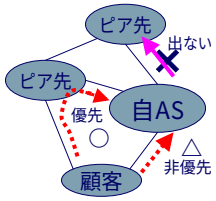
→ 数字には余裕をもって設計
→ ここでの優先順位とは、単純にLOCAL_PREFの値を元とした順位

BGPポリシー設計（受信）

ポイント1： BGP顧客経路は、まず最優先に設定する

接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	110	120	3
商用IXピア経路	300	200	210	220	4
学術IXピア経路	300	300	310	320	5
上流フルルート1	200				6
上流フルルート2	200				6

→ 顧客経路は他のISPなどにちゃんと広報する必要がある
→ もしその顧客が他のISPとマルチホーム接続をしていれば、
ピア経路としても聞こえてくる場合がある
→ その際、仮にピア経由を優先してしまうと、自AS内でベストパス
ではなくなるため、経路がアナウンスされなくなってしまう！



BGPポリシー設計（受信）

ポイント2： BGP顧客の次に、自AS内広報経路は優先させる

接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	110	120	3
商用IXピア経路	300	200	210	220	4
学術IXピア経路	300	300	310	320	5
上流フルルート1	200				6
上流フルルート2	200				6

→ 自AS内経路が、仮に他から流れてきて、Filterにもひっかからなかった
ような場合も想定し、優先させておく必要がある
→ BGP顧客よりも優先度が低いので、顧客から自ASの経路が流れてきた
場合を想定する必要がある。これは、顧客のエッジでフィルタをかける
などの対応をして防ぐ必要がある（顧客経路しか受け取らない）

BGPポリシーは、Filterとの組み合わせで、複合的に考えていく必要がある
→ 一概に上記と同じPriority付けにはならない、ということに注意頂きたい

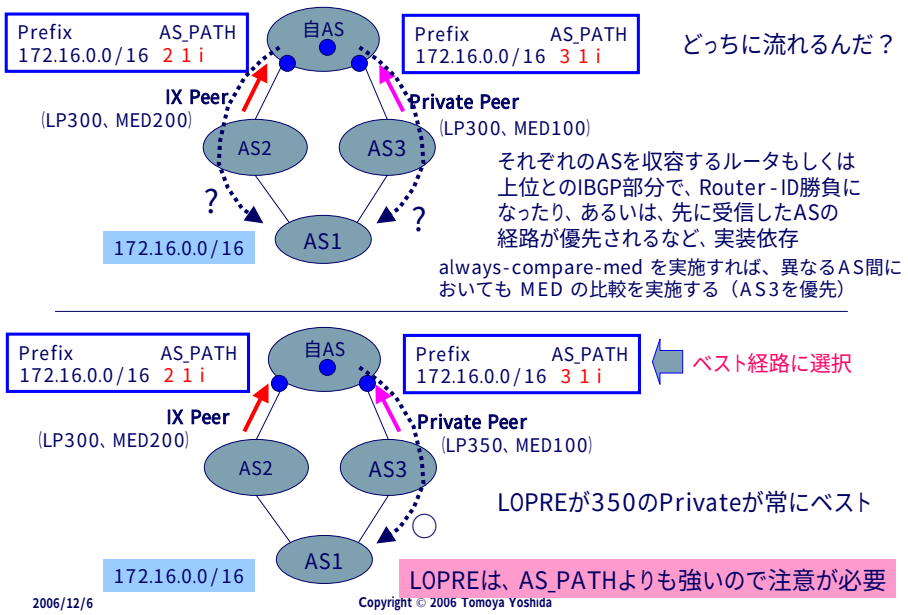
BGPポリシー設計（受信）

ポイント3-1： ピア経路は、LOPREを統一し、MEDで勝負させる

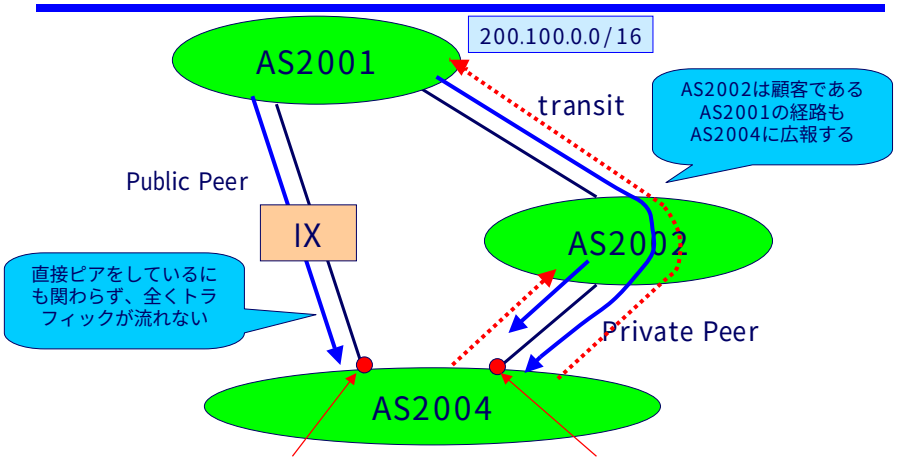
接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	110	120	3
商用IXピア経路	300	200	210	220	4
学術IXピア経路	300	300	310	320	5
上流フルルート1	200				6
上流フルルート2	200				6

→ ピア経由の経路は、基本はAS_PATHによる制御
→ 異なるAS間ではMED比較の対象ではないので、Updateを先に受け取った経路や、Router-IDの大小による比較、ＩＧＰ metric値がもっとも小さいところから抜けていくなどが考えられる
→ プライベートピアを優先されるように、LOPREを高く設定する設定もあり
（例）Local_Preference＝350

BGPポリシー設計（受信）



直接ピアをしているのにトラフィックが流れない例



AS2004での経路の見え方

	Prefix	AS Path	LP	
	200.100.0.0/16	2001	150	
>	200.100.0.0/16	2002 2001	200	◎ベストパス

BGPを使った経路情報の流れ
実際のIPトラヒックの流れ

BGPポリシー設計 (受信)

ポイント3 - 2： Closet Exit で、近いところからルーティング

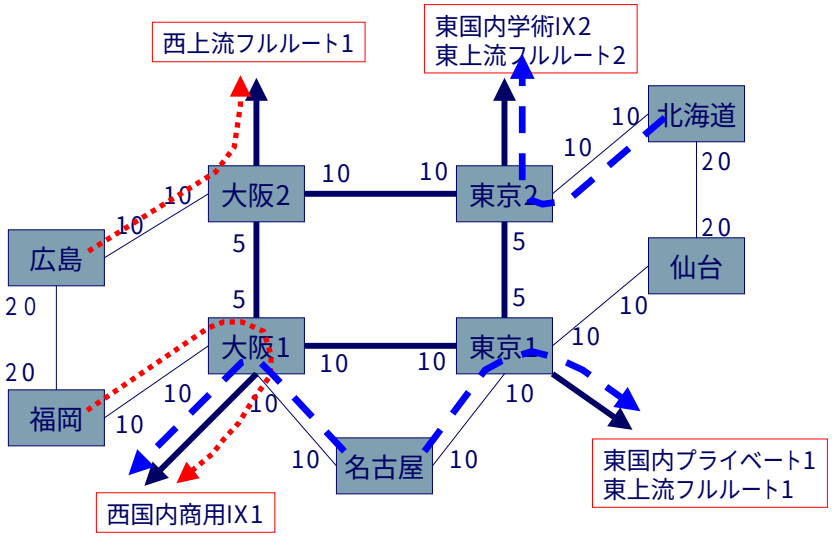
接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	100	100	3
商用IXピア経路	300	100	100	100	3
学術IXピア経路	300	100	100	100	3
上流フルルート1	200				6
上流フルルート2	200				6

→ プライベートやIXなどは区別しない
→ IGPのもっとも近いところからルーティングさせる (IGPの設計が重要になってくる)

ＯＳＰＦでIGPのトポロジーを管理する場合には、ＯＳＰＦのコスト設計が重要になってくる。また同時に、回線設計も重要

BGPポリシー設計（受信）

Closest Exit の場合には、どこに何を収容するのかが非常に重要になってくる
→ 回線収容設計がトラフィックの制御に影響を及ぼす



BGPポリシー設計（受信）

ポイント4： 上流フルルートは、うまく使い分ける

接続形態	LOCAL_PREF	MED1	MED2	MED3	優先順位
BGP顧客経路	500				1
自AS内広報経路	400				2
プライベートピア経路	300	100	110	120	3
商用IXピア経路	300	200	210	220	4
学術IXピア経路	300	300	310	320	5
上流フルルート1	200				6
上流フルルート2	200				6

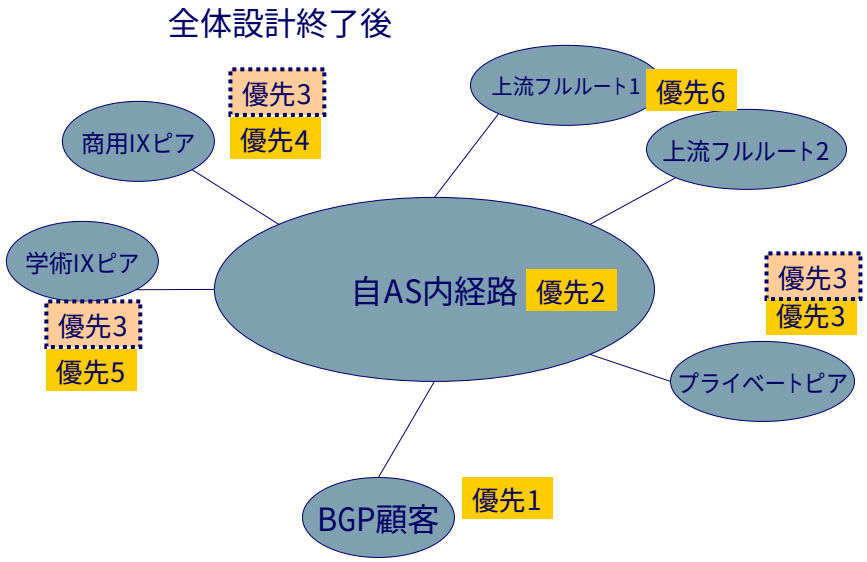
→ もっとも優先度が低いので、何でも良さそうだが、多くの実装で、LOPREのデフォルト値が100になっているため、その値よりも大きくしておくのが望ましいだろう
理由：仮にLOPRE50などで設定していた場合、うっかりミスで、フルルートを他のBGP接続からデフォルトで受信してしまうと、全てがそちらにひっぱりこまれてしまう

→ 使い分けに関しては、AS_PATHにまかせるのが基本．AS-PATH Prepend や、コミュニティを用いて制御する場合も多くある（顧客経路はそれぞれ優先させるなど）
（例）上流1が安い場合には、上流2から受信するときに、Prependを1つかませる

BGPポリシー設計 (受信)

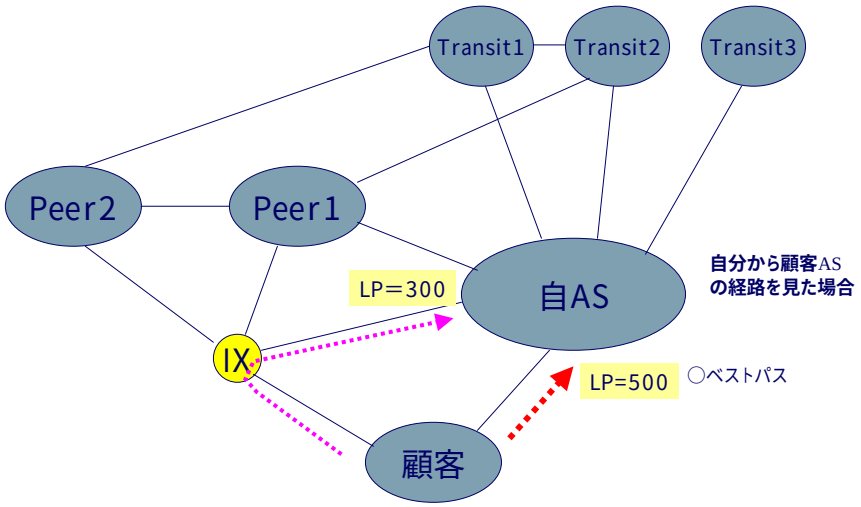
- Closest Exit の注意点
  - IGPメトリックがきいてくるので、OSPFのコスト設計が重要
  - Externalの回線をうまく分散収容する必要がある
    - 同じような位置付けのところに収容すると、ある部分ばかりに引き込まれて偏ってしまう可能性が高い
- 上流の制御
  - 上流が2つ以上ある場合、それぞれのCustomer経路をまずは優先 (近い)
    - 顧客コミュニティーにマッチしたら、優先度を高くして受信　など
    - 大抵上流ISP (Transit ISP) ではコミュニティーがインプリされている
  - それ以外のTransit経路は、例えばコストの安いほうを優先的に利用
    - 完全1:0形態にするなら、LOPREで制御したほうが確実
    - ある程度Topologyに依存させるには、AS_PATH Prependで制御
    - MEDは異なるASでは比較できないので使えない
- 自ASの経路の扱い
  - 自ASの経路を顧客に渡すなどの場合には、顧客の経路が優先となる必要があるため、それよりも優先度を低くするのが望ましいだろう
  - 外部から自分に対して広告されても、Filterではじくなどの仕組みは必要

BGPポリシー設計 (受信)



BGP受信ポリシー確認1

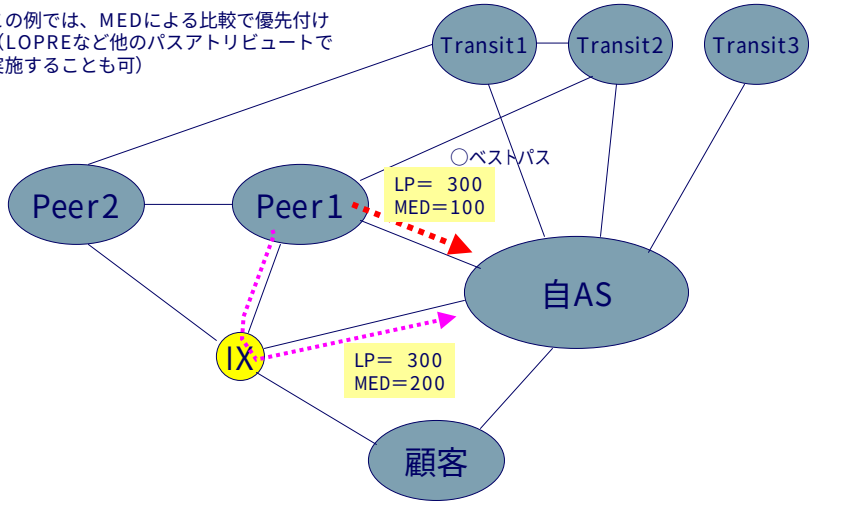
★顧客 かつ ピアの場合は顧客優先、切れたときはIX経由



BGP受信ポリシー確認2

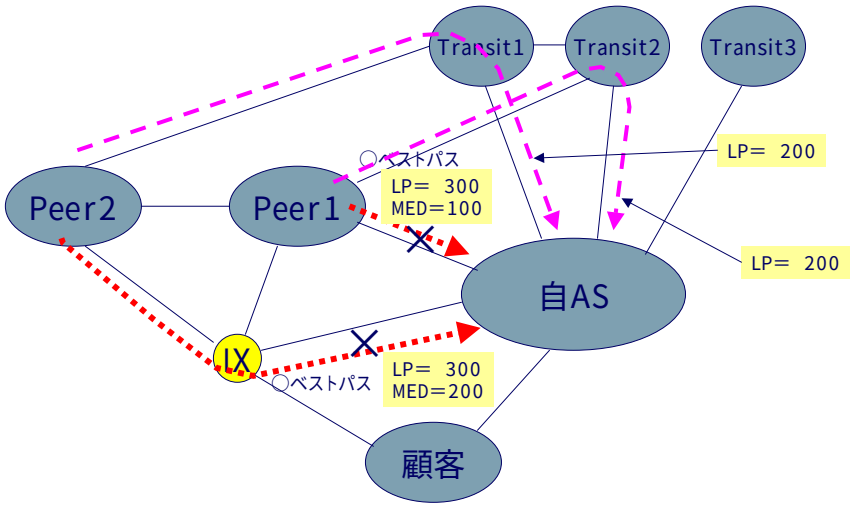
★PrivateピアとIXピアがある場合は、Privateピア優先

この例では、MEDによる比較で優先付け
（LOPREなど他のパスアトリビュートで
実施することも可）



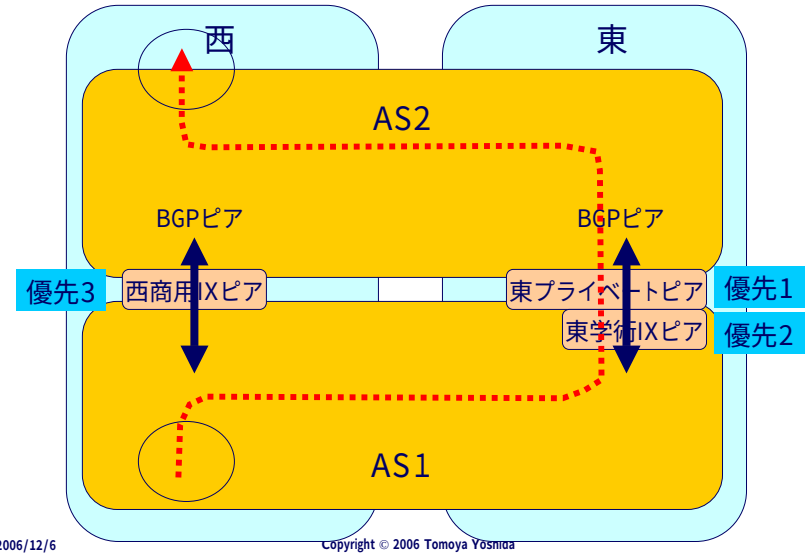
BGP受信ポリシー確認3

★国内ピアが落ちた場合には、(海外) Transitで救済したい



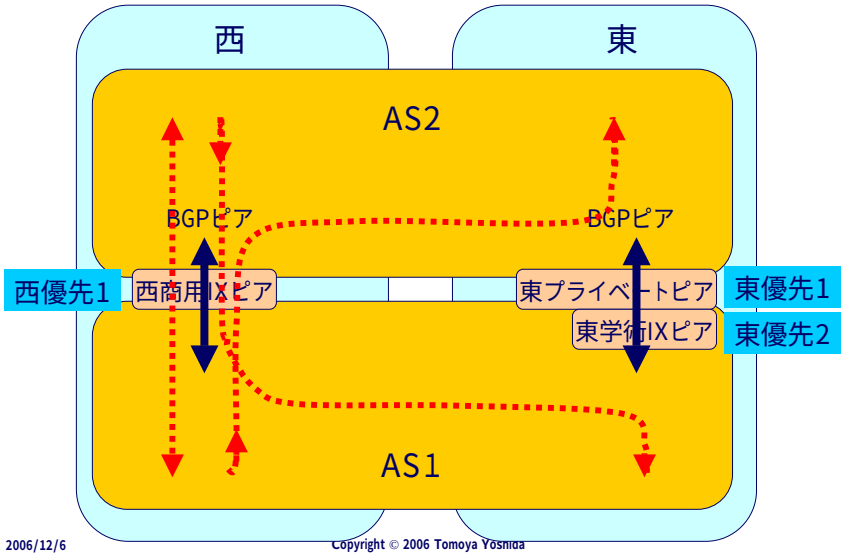
BGPポリシー設計 (さらに)

今までのポリシーだと、折角西でピアをしているのに、わざわざ
東のプライベートを経由して西に戻ってしまう → うまく最適化できない？



経路の最適化

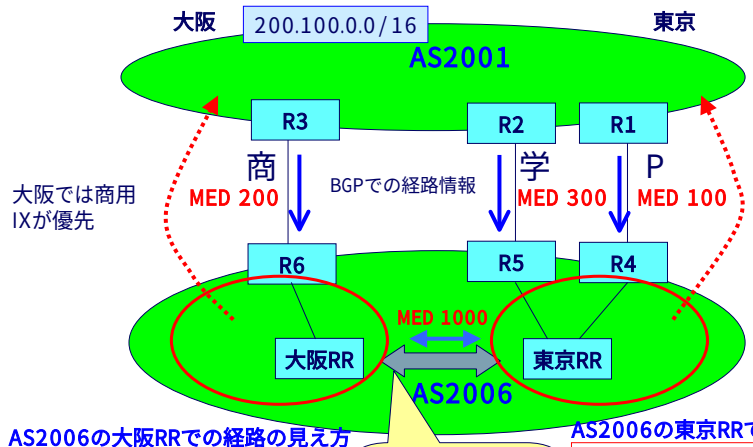
東、西　それぞれ近いところからルーティングするようにしたい



Hot-Potato と Cold-Potato

- Hot-Potato
  - 最も近いところから相手にパケットを出してしまう＝Closet Exit
    - AS1西 ⇒ AS2西
    - AS1東 ⇒ AS2東
- Cold-Potato
  - Hot-Potatoのように近いところからルーティングするのではなく、相手が近いところに出す
    - AS1西 ⇒ AS1東 ⇒ AS2東

Hot-Potatoによる経路制御



AS2006の大阪RRでの経路の見え方

	Prefix	MED	
>	200.100.0.0/16	200	◎ベストパス
	200.100.0.0/16	1000	

相手に経路を互いに渡す際、MEDを1000にして渡す

AS2006の東京RRでの経路の見え方

	Prefix	MED	
>	200.100.0.0/16	100	◎ベストパス
	200.100.0.0/16	300	
	200.100.0.0/16	1000	

Hot-Potatoによる経路制御 (Juniperの例)

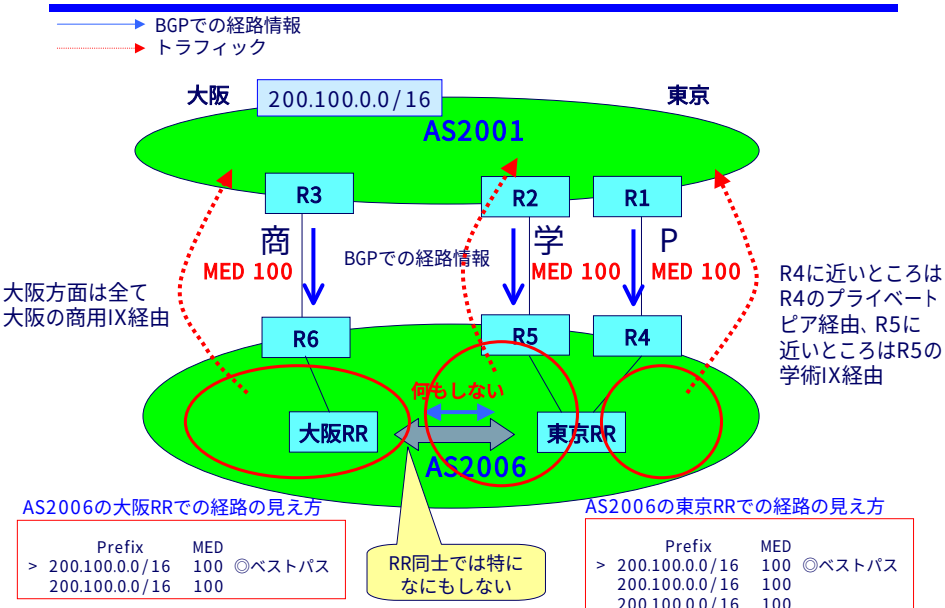
```
protocols {
   bgp {
      group to-RR {
         type internal;
         local-address X.X.X.X;
         peer-as 2006;
         neighbor Y、Y、Y、Y {        ←  Neighborである大阪RRのアドレス
            import HOT_POTATO-IN;     ←  Hot Potato 用Policy
         }
      }
   policy-statement HOT_POTATO-IN {
      term AS2006 {                   ←  対象ISP名
         from as-path AS2006;         ←  対象ISPのAS-Pathの指定
         then {
            metric 1000;              ←  MEDを"1000'に設定
            local-preference 150;     ←  ピアのLocal Preferenceとして設定
            accept;                      書かなければeBGPから受けた時に付加
         }                               されたものがそのまま渡される
      }

      term AS-ALL {                   ←  Hot Potato 以外のISP経路の受信を許可
         from as-path AS-ALL;
         then accept;
      }
      term Other {                    ←  それ以外の経路受信を削除
         then reject;

     as-path AS2006 "(2006 .*)";      ←  対象ISPのAS Numberで始まるAS-Path
     as-path AS-ALL "(.*)";              を指定
```

東京RR
のConfig例

Closet Exit

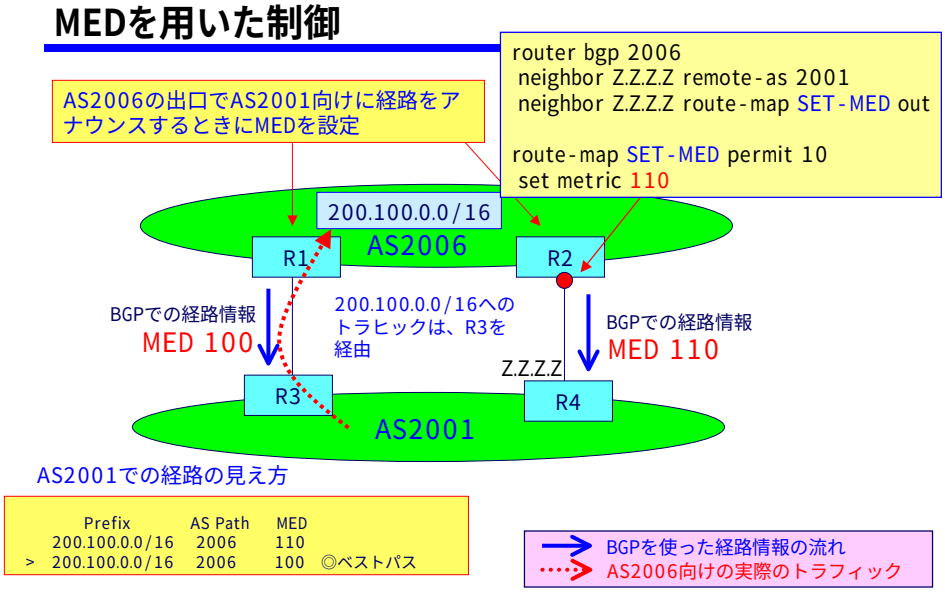


BGPポリシー設計（広告）

BGPポリシー設計（広告）

- 以下の3つのパスアトリビュート・手法を使った制御が基本
  - MED
    - 基本は異なるAS間で比較されないので、隣接AS同士が複数回線で結ばれている場合に有効
  - AS-PATH Prepend
    - 自分のAS-PATHを相手に遠くみせる手法
  - Communityの利用
    - 相手と自分の間で、このCommunityはどうゆう制御をする、ということを事前に取り決めがされている、あるいは公開されているので、相手の優先度を自分主体で調節したりといった柔軟な制御が可能
- 広告経路
  - 上流やピア先には、自分のアドレスとBGP顧客経路を広告
  - BGP顧客には、フルルートを広告
    - 場合によっては、デフォルトルートのみを配信 → お客さん側のBGPルータがメモリ的に厳しいような状況など

MEDを用いた制御



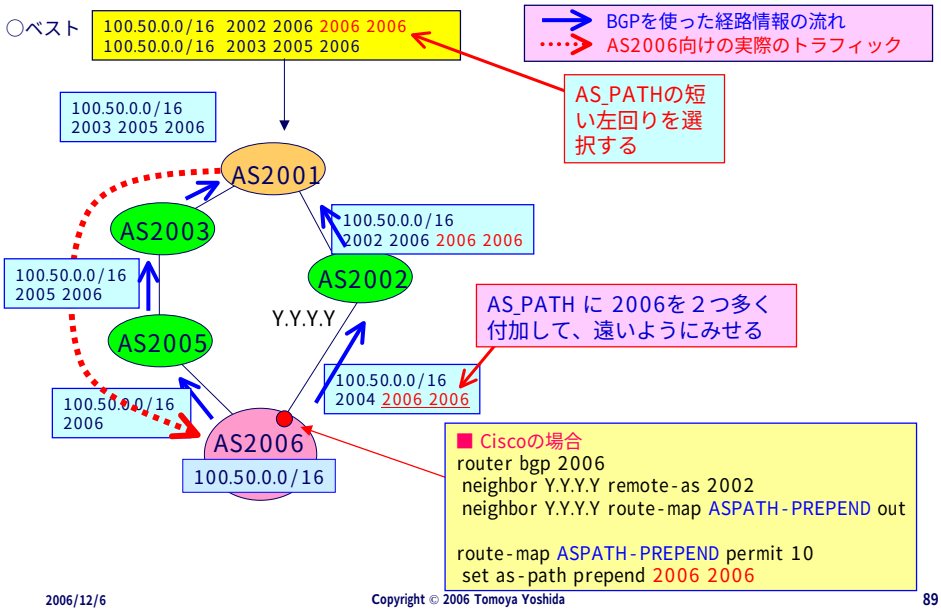
AS2001での経路の見え方

```
      Prefix          AS Path   MED
   200.100.0.0/16     2006      110
>  200.100.0.0/16     2006      100   ◎ベストパス
```

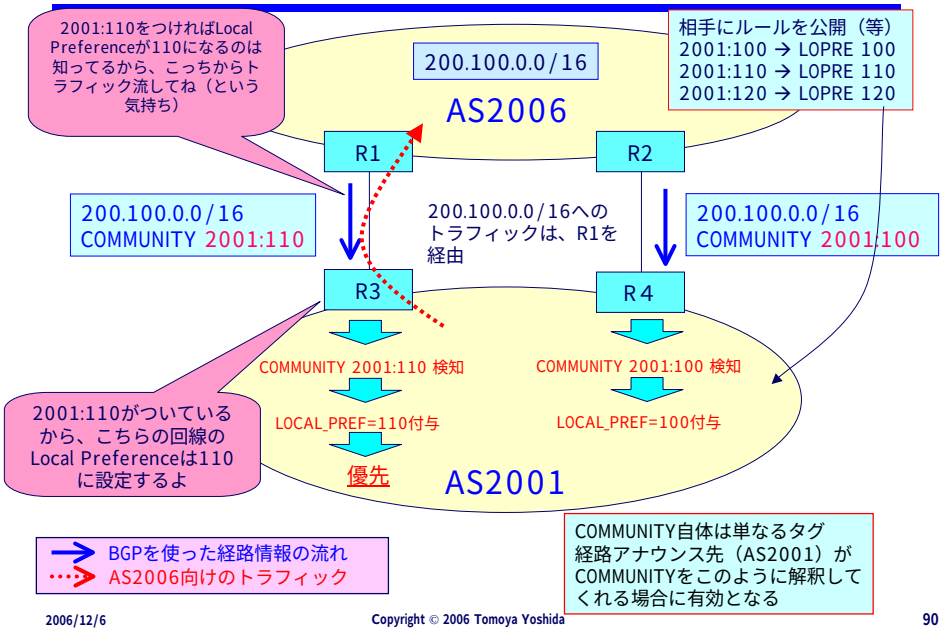
BGPを使った経路情報の流れ
AS2006向けの実際のトラフィック

相手から自分に帰ってくるトラフィックを制御することができる

AS_PATHを用いた制御



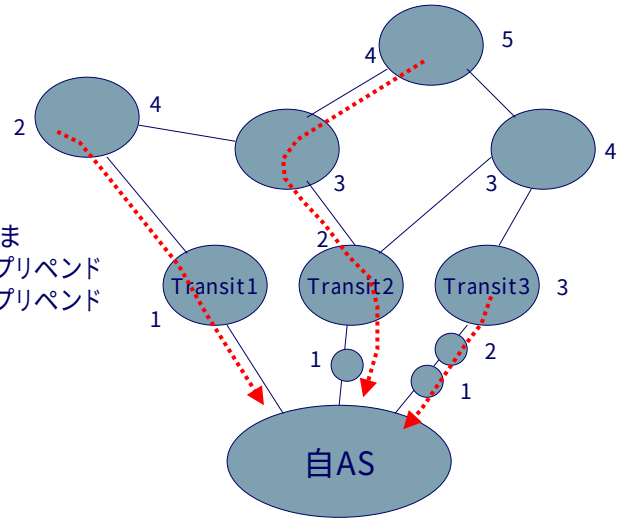
Communityによる戻りのトラフィック制御



BGP広告ポリシー確認1

★海外上流1>2>3 の順序でなるべく使いたい

Transit1はそのまま
Transit2には1つプリペンド
Transit3には2つプリペンド

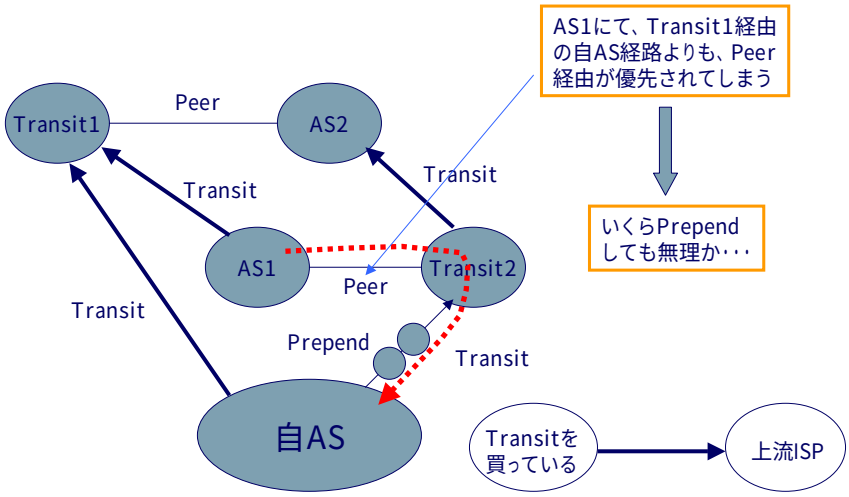


海外上流のトラフィック制御の難しさ

- 上流のその先のTopologyやPeerの関係などなるべく日々把握していく必要がある
  - 上流のTopologyはけっこう変わる
    - 突然急激にトラフィックが変動している。何故？
      - ASが統合されて、既存のTopologyがくずれた・・・
    - よくよく見るとAS-PATHが変わっている
  - でも、Lopreだと強すぎるから、AS-PATH制御になってくる
  - いくらPrependしても、トラフィックが減らない
    - 上のTransit･Peerの関係上無理な場合がある

BGPポリシー設計（広告）

★どうPrependしても、ひっぱりこんでしまう場合



BGP4+

- RFC2858 Multiprotocol Extensions for BGP-4
- RFC2545 IPv6 への Extension
  - Neighbor address は、global or link-local
  - Next-hop-addressは、global＋link-local
  - TransportとしてIPv4を使用することも可能

iBGP設計

iBGP設計

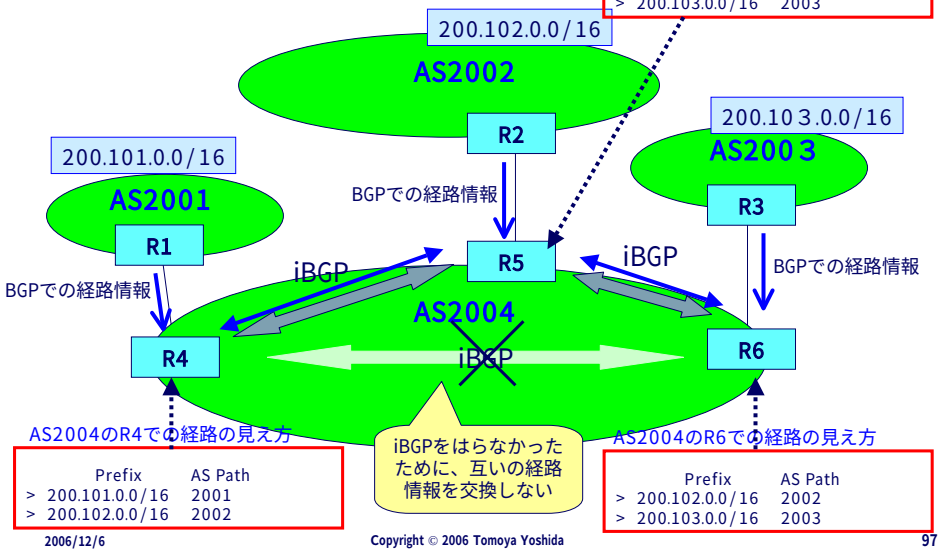
- 全BGPルータが正しくBGP経路情報を保有し、それぞれのルータが正しく経路選択を可能とするように設計する
  - 全く同じ情報を保持する必要があるのとは違う
- BGPの経路は配送すべきところに適切に配送する
  - OSPFのデフォルトルートなどで十分なところはデフォルトでルーティングさせる
  - 内部の細かい経路は必要ないところには配送しないなども可能
    - BGPユーザ向けの階層にはフルルートのみを
    - それ以外の収容ルータ向けには経路を配送しない
- リフレクタ階層構造の利用
  - それほど数が多くなければフルメッシュで十分

BGP経路情報の不一致

iBGPで受信した経路は、他のiBGPピアには渡さない．例えばR5はR4から受信した200.101.0.0/16をR6には広報しない

AS2004のR5での経路の見え方

	Prefix	AS Path
>	200.101.0.0/16	2001
>	200.102.0.0/16	2002
>	200.103.0.0/16	2003



AS2004のR4での経路の見え方

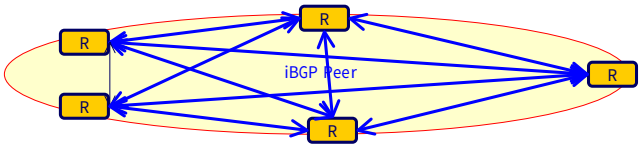
	Prefix	AS Path
>	200.101.0.0/16	2001
>	200.102.0.0/16	2002

AS2004のR6での経路の見え方

	Prefix	AS Path
>	200.102.0.0/16	2002
>	200.103.0.0/16	2003

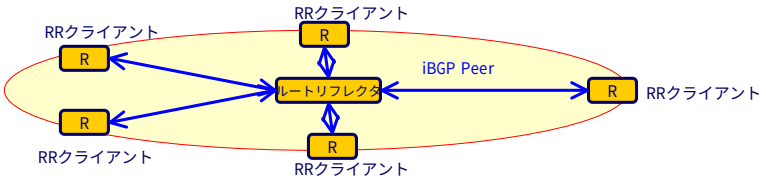
ルートリフレクタ (RR)

●一般的なiBGP Peer



iBGPはフルメッシュでなくてはならない

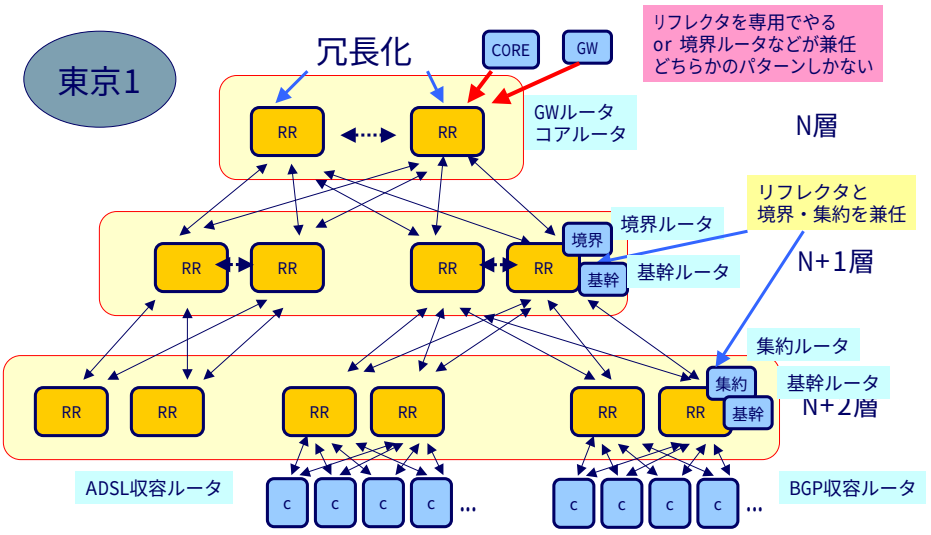
●ルートリフレクタ(RR)を使用したiBGP Peer



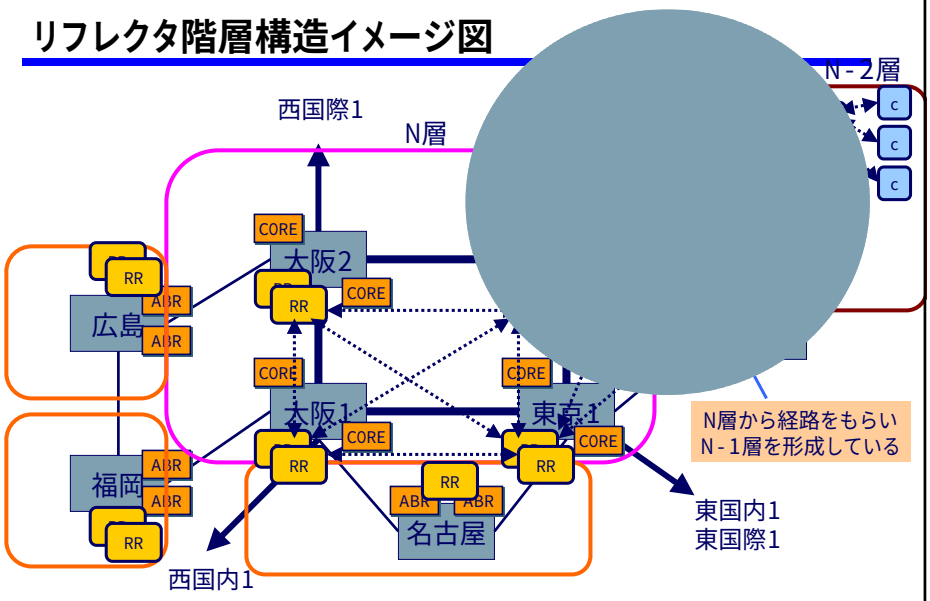
iBGPフルメッシュをルートリフレクタを用いたPeerにより代用

リフレクタ階層構造

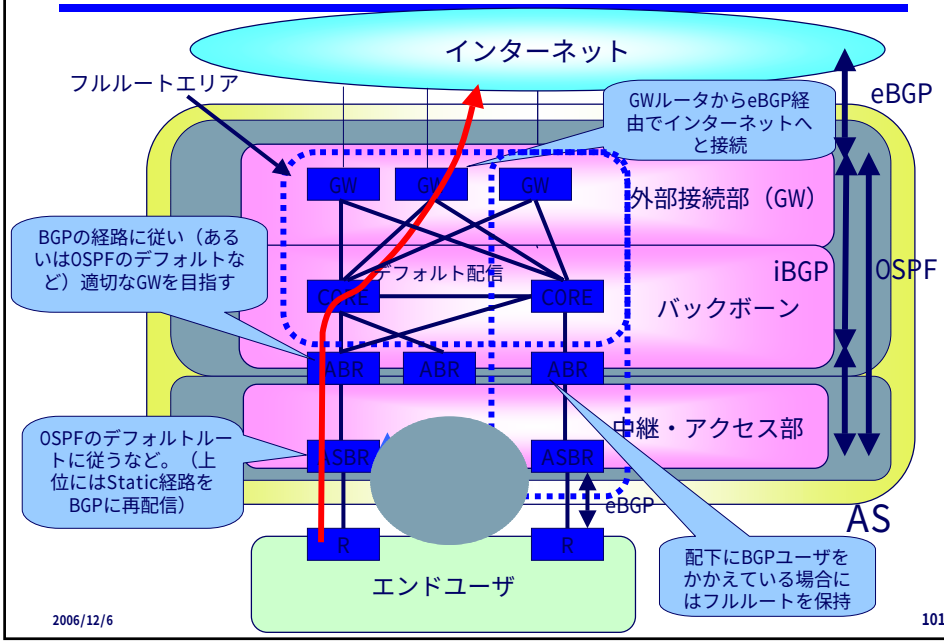
東京1地域を例とするルートリフレクタによるiBGP階層構造
ネットワークの規模により階層は異なる



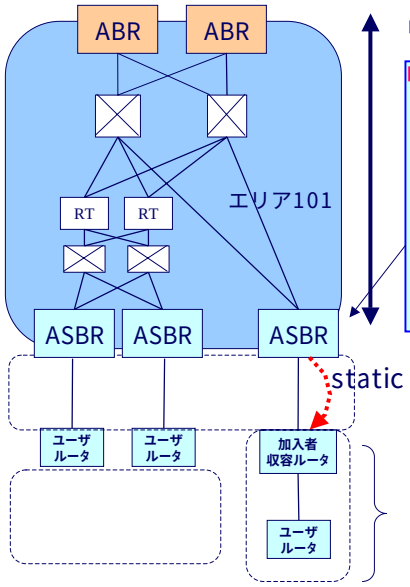
リフレクタ階層構造イメージ図



行きのルーティング



ユーザStatic経路をBGPに再配信



中継・アクセス部

■ Ciscoの例

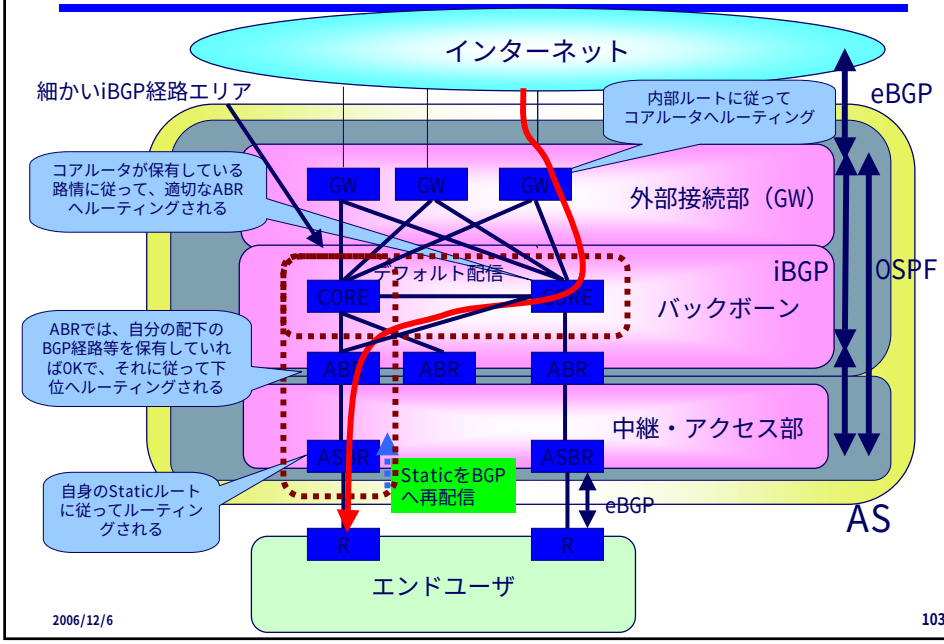
```
router bgp 100
 redistribute static route-map s-to-bgp
 neighbor X.X.X.X remote-as 100
 neighbor X.X.X.X send-community
 neighbor X.X.X.X next-hop-self
 neighbor X.X.X.X update-source loopback 0

ip route a.a.a.a b.b.b.b c.c.c.c

route-map s-to-bgp permit 10
 set community no-export
```

ASBRでユーザアドレスをstaticで記述．それを上位にBGPで再配信．BGPの場合には、no-exportをつけて、GWルータから外にでていかないようにする。内部のiBGPではsend-communityを動作させ、no-exportのCommunity情報がついたものは、内部でのみ伝播する

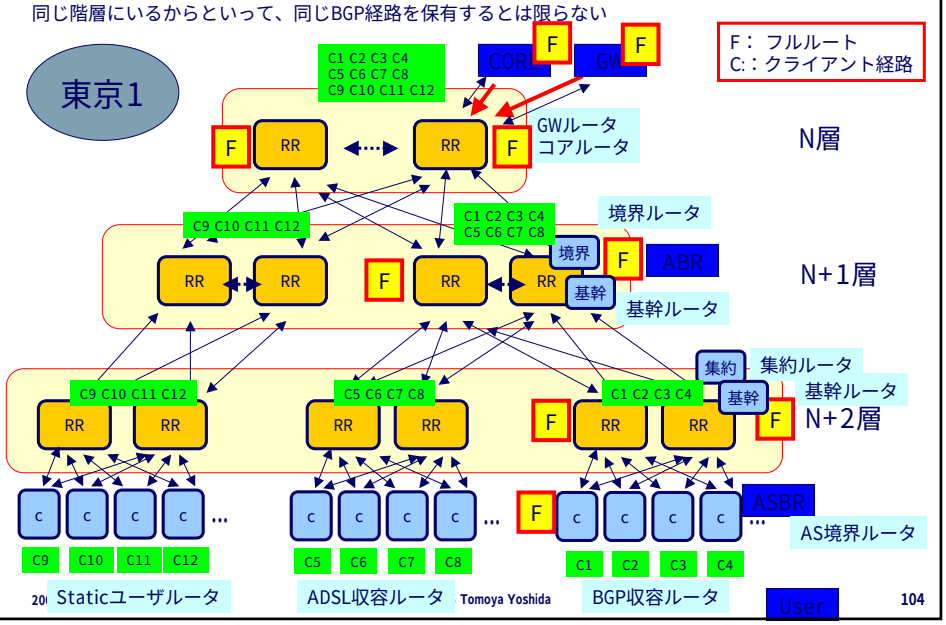
帰りのルーティング



2006/12/6 103

リフレクタ階層構造の経路配信イメージ

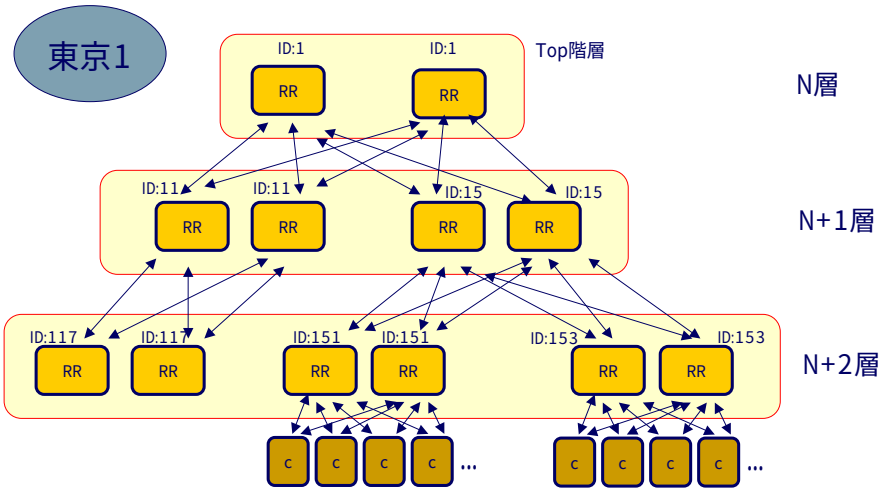
同じ階層にいるからといって、同じBGP経路を保有するとは限らない



Tomoya Yoshida

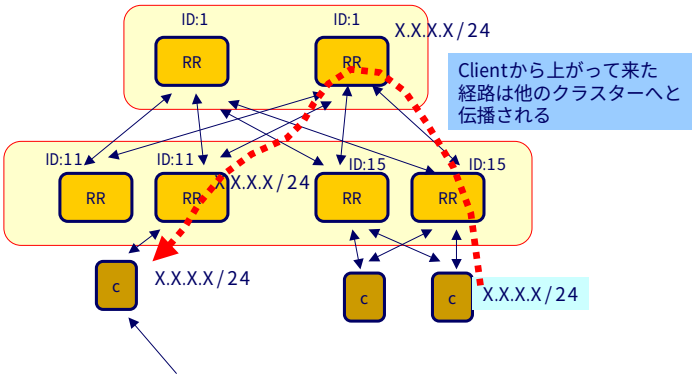
104

リフレクタ階層構造（東京1の例）



東京1地域を例とするルートリフレクタによるiBGP階層構造
1つ前の層からIDが辿れるような付与規則にするとわかりやすい

他のクラスターから経路が伝播される

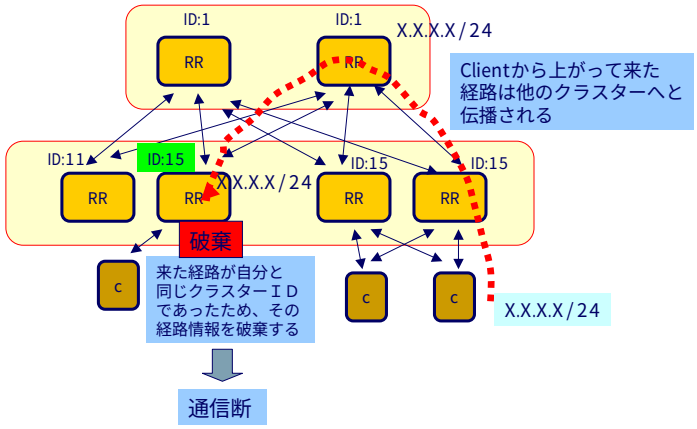


Cluster list: 0.0.0.11 、 0.0.0.1 、 0.0.0.15

リフレクタルータが、また別のリフレクタルータへと経路を配送している. Cluster list は、
辿ってきたクラスターが順に並んでいる. リフレクタが他のリフレクタに配送する場合に、
自分のIDを先頭につけて配送していく （AS_PATH同じようなイメージで、左がもっとも直近）

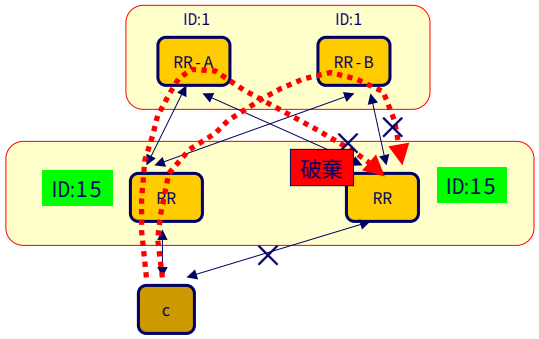
AS_PATH : 4713 2914 701

クラスターIDの設定ミス



クラスターIDが重複してしまったために、自分と同じクラスターIDの経路を他から受信すると、routing loop protection により破棄
（AS_PATHのループディテクションと原理は一緒）

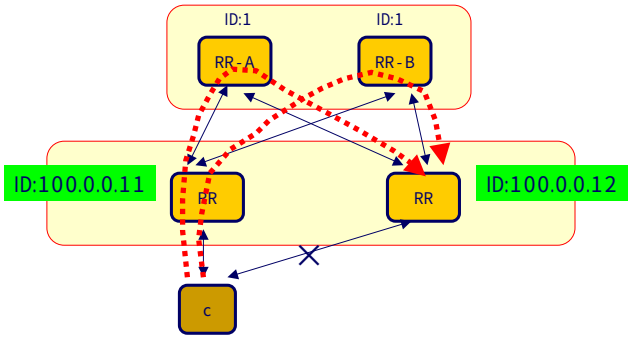
クライアントとのピアが切れた場合（同一ID）



クライアントの片方のピアがきれた場合には、もう一方のリフレクタから上位に配信された経路は、同一IDのため破棄される

ただし、通常各クライアントは、各々両方のリフレクタにピアをはっているので、どちらか一方から経路を受信できる

別のIDを付与した場合



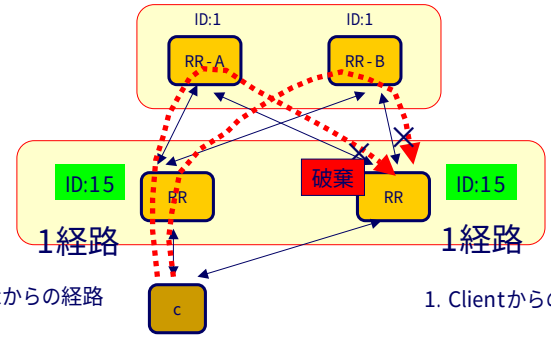
別IDの場合には、クライアントの片方のピアがきれても、
上位リフレクタから経路が配信される（通常状態においても配信される）

RRがパケットフォワーディングも兼ねている場合には、この方法になる

Cluster-id ツリーが増えるので、適応個所には注意したいが、大きな問題はない。BGP経路の伝播が、同一ID適応時とは異なるので注意

経路の見え方（同一IDの場合）

クラスターIDが同一のため、上位リフレクタから反射した経路は
同一IDの下位リフレクタにはわたらない

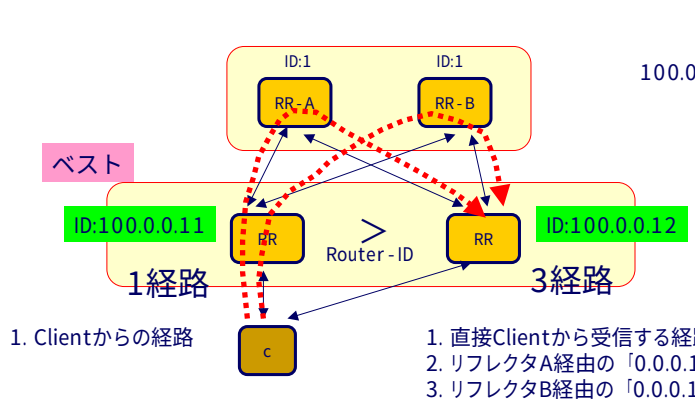


1. Clientからの経路

1. Clientからの経路

経路の見え方（別IDの場合）

IGPコストが同等の場合には、ルータID（リフレクタからの経路受信の場合には Originator ID、それでも一緒の場合にはRouter-ID)が小さいほうをベストに選択．この場合、RR-A、RR-B 共に100.0.0.11からの経路をベストに選ぶ



100.0.0.11 には反射しない

なので1経路しかない

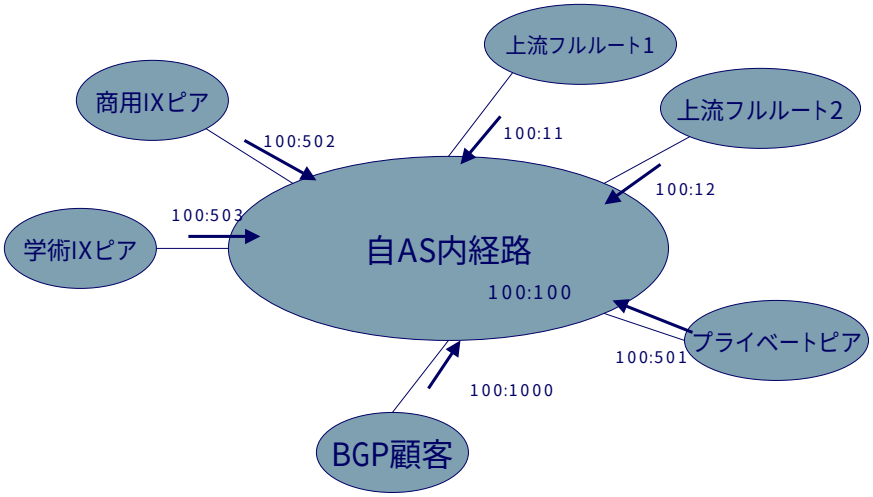
1. Clientからの経路

1. 直接Clientから受信する経路
2. リフレクタA経由の「0.0.0.1 、 100.0.0.11」の経路
3. リフレクタB経由の「0.0.0.1 、 100.0.0.11」の経路

Originator ID：その経路の生成元（GWルータのLBなど）

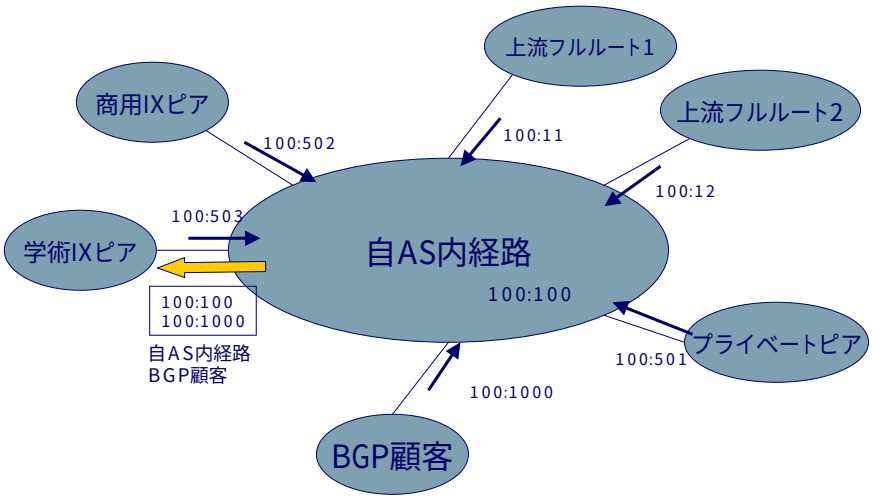
BGPコミュニティの利用（1）

外部からBGP経路受信時、あるいは経路生成時にコミュニティを付加
⇒経路情報に色づけを行い、柔軟な経路制御を可能とする



BGPコミュニティの利用 (2)

付与したコミュニティ値をもとに経路配信を実施
⇒あらかじめ配信ポリシーをコミュニティ値に基づき定義をしておく



iBGP設計のおさらい (1)

- リフレクタの階層化
  - COREを中心とした物理的な階層に沿った論理的な階層化が理想的
    - 経路配送自体も、GWから入ってきたフルルートはCOREを中心に
    - リフレクタがフォワーディングも兼任する場合には注意
  - ID付与時には、わかりやすい数字かループバックアドレスにて設定
  - 何がどのように配信されるのかは、それぞれのネットワークによって異なるので、そのポイントをきちんと押さえておく
- サービスごとにセグメント化を実施し、各セグメント毎に配信経路やルーティング方式を検討する
  - BGPユーザのクラスタ
    - 当然BGPで経路を配信 (フルルートの配信)
    - 他のクラスタの細かい経路を省くことは可能
  - DSLクラスタ
    - 上位にはBGPでクライアント経路を配信
    - 極力フルルートを保有が望ましいが、必須ではない (ポリシー依存)

iBGP設計のおさらい (2)

- BGPコミュニティの利用
  - 経路受信時、生成時に付与する
    - ポリシー毎に複数の値を付与することが可能
  - 配信する際には、コミュニティ値に基づく動的経路制御
  - 注意点
    - 値を誤ると意図しない経路が広告される可能性あり

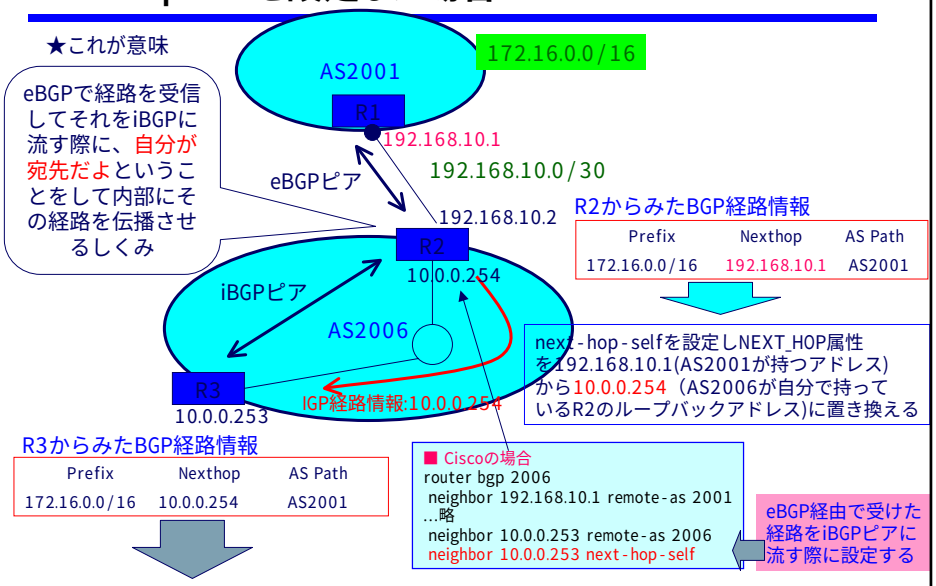
BGP その他

- next-hop-self
- リカーシブルックアップ
- eBGPマルチホップ/マルチパス
- CIDRの広報
- ルートダンプニング

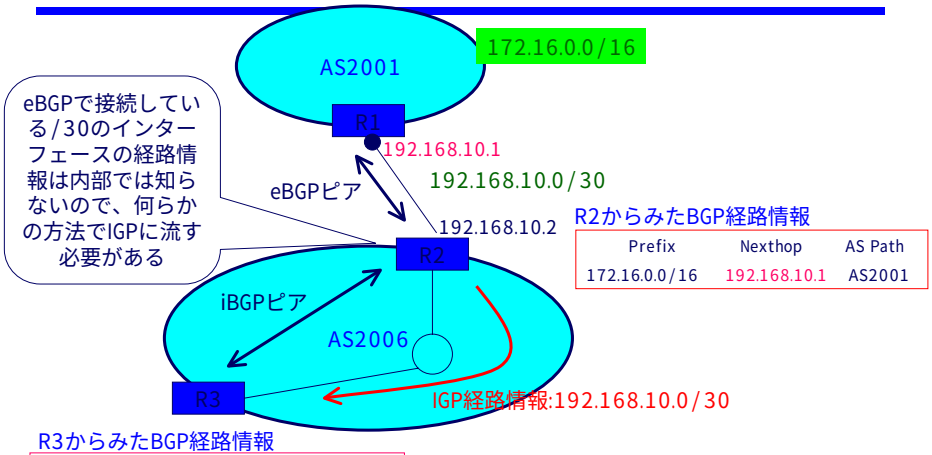
BGPのnext-hopの解決方法

- BGPでは、相手から受信した経路のnext-hopに到達性がなければ、その経路は無効とする（NEXT_HOP属性）
  - eBGPの場合には、受信時に破棄
- 外部経路のNEXT_HOPの解決方法は2つ
  - eBGPから受信する際に、自身のループバックをnext-hopとする
    - iBGPに対して、「next-hop-self」を設定（Ciscoの場合）
    - そのループバックはOSPFなどのIGPでルーティング
  - eBGPピアで使用している/30などのconnectedアドレスをOSPFなどのIGPプロトコルにて広告
    - redistribute connected ←―― better
    - Netwrokコマンド + passive

next-hop-selfを設定した場合



eBGP経路をそのままiBGPに流した場合



eBGPで接続している/30のインターフェースの経路情報は内部では知らないので、何らかの方法でIGPに流す必要がある

R2からみたBGP経路情報

Prefix	Nexthop	AS Path
172.16.0.0/16	192.168.10.1	AS2001

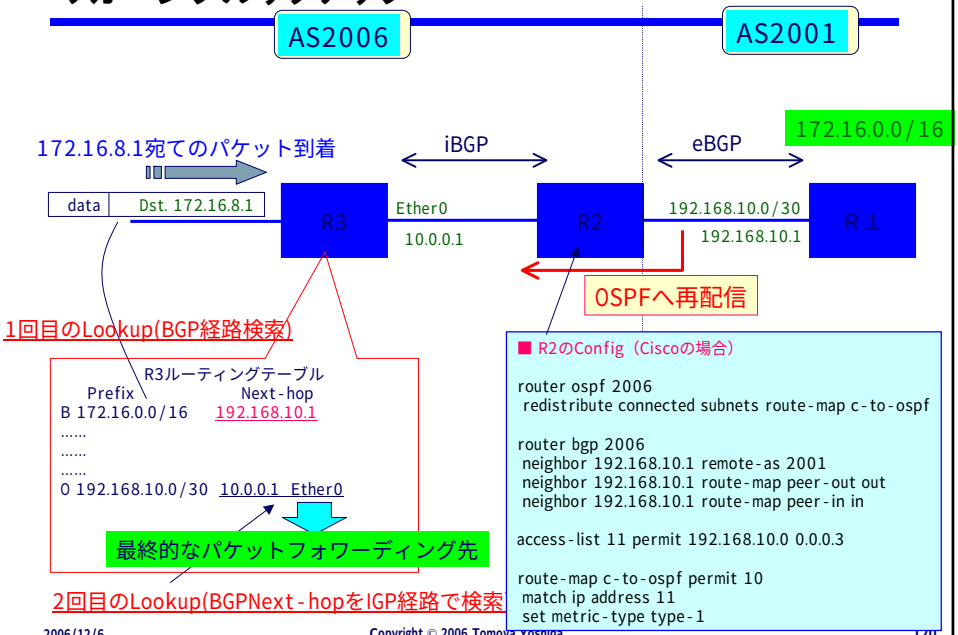
R3からみたBGP経路情報

Prefix	Nexthop	AS Path
172.16.0.0/16	192.168.10.1	AS2001

192.168.10.1に到達するためのルーティング情報がIGPで流れていなくてはならない

AS2001の172.16.0.0/16のNEXT_HOP属性の値は、AS2006から見たときの該当アドレスへのボーダールータのアドレス．この図では、192.168.10.1が出口のアドレス

リカーシブルックアップ



■ R2のConfig（Ciscoの場合）

```
router ospf 2006
 redistribute connected subnets route-map c-to-ospf

router bgp 2006
 neighbor 192.168.10.1 remote-as 2001
 neighbor 192.168.10.1 route-map peer-out out
 neighbor 192.168.10.1 route-map peer-in in

access-list 11 permit 192.168.10.0 0.0.0.3

route-map c-to-ospf permit 10
 match ip address 11
 set metric-type type-1
```

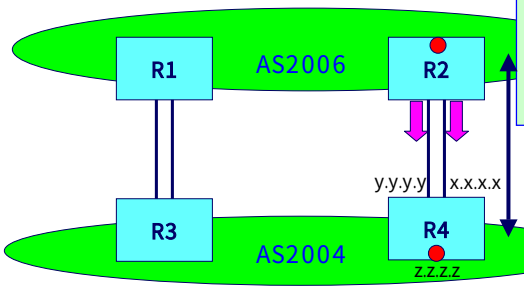
eBGPマルチホップによるロードバランス

同一ルータで外部と複数本でeBGP
ピアをはる場合、eBGPマルチホップ
によりロードバランスが可能

■ Ciscoの場合（R2）
```
router bgp 2006
 neighbor z.z.z.z remote-as 2004
 neighbor z.z.z.z ebgp-multihop 2

ip route z.z.z.z 255.255.255.255 x.x.x.x
ip route z.z.z.z 255.255.255.255 y.y.y.y
```

■ Juniperの場合（R2）
```
protocols {
  bgp {
    group eBGP {
      type external;
        multihop {
            ttl 2;
        }
      peer-as 2004;
      neighbor z.z.z.z;
    }
  }
 }
}
routing-options {
 static {
   route z.z.z.z/32 next-hop [ x.x.x.x y.y.y.y ];
 }
}
```



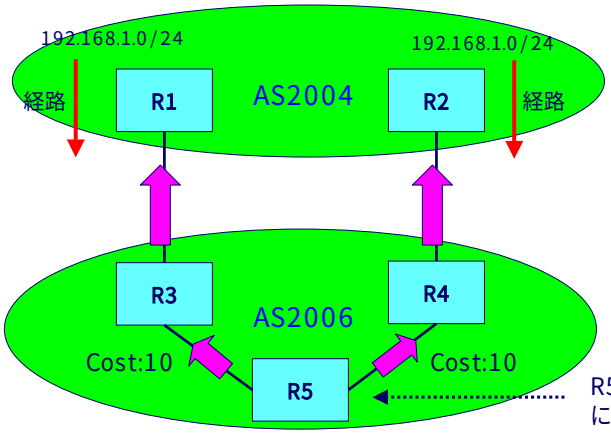
■ループバックアドレスで互いにピアをはる
■相手のループバックに対するルーティングは、Static Route を物理インターフェースに対して設定することにより解決

z.z.z.z → ループバックアドレス

iBGP multipath によるロードバランス

複数のeBGPピアから受信した経路に対して、内部でバランスさせる

■ BGPマルチパスの条件
BGPのマルチパス機能が有効になっていること
経路選択プロセスで、IGPメトリックによる選択をしても決着がつかない場合
※ベンダによって、仕様が異なるので注意



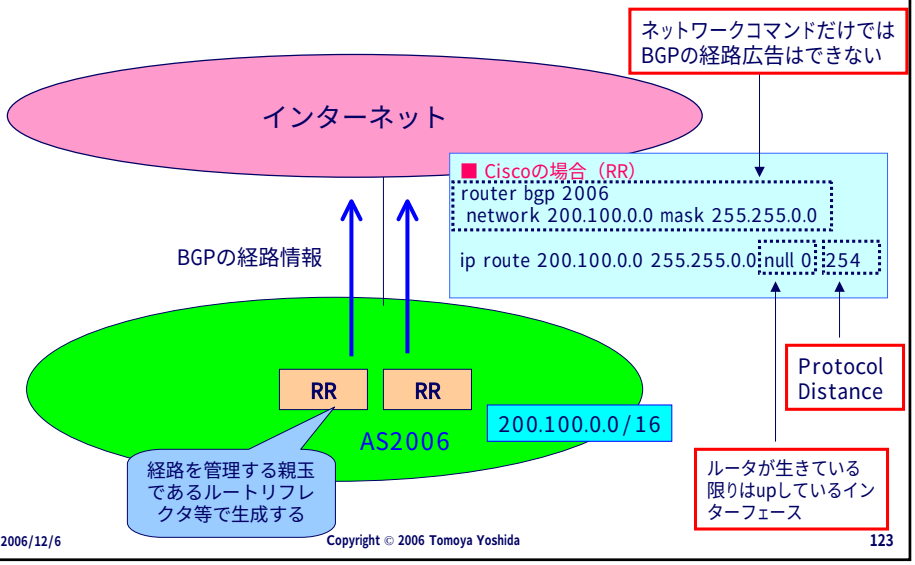
■ Ciscoの場合（R5）
```
router bgp 2006
 maxiumu-path ibgp 2
```

■ Juniperの場合（R5）
```
protocols {
  bgp {
    group iBGP {
      neighbor x.x.x.x {
          multipath;
```

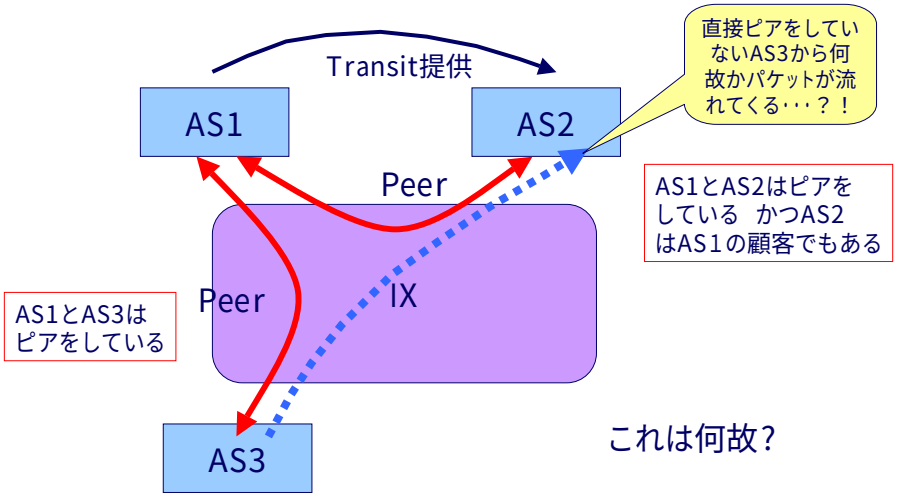
R5にて、マルチパス機能を有効にする. eBGPに対しても可能

PAアドレス（CIDR経路）の広告

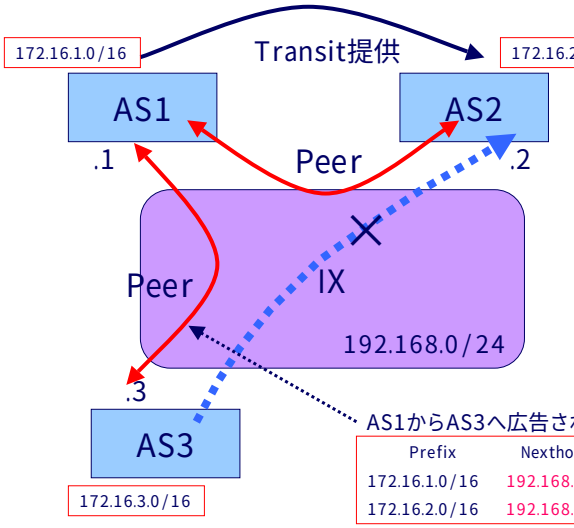
・CIDR経路は「安定して」インターネットに広報されていなくてはならない
・BGPで経路広告する際のIGPは、「static null0」で



next-hop-self つづき



next-hop-self つづき



本当は、マルチアクセスメディア
SWネットワークにおける最適な
next-hopとするためのBGPの実装
→ 直接トラフィックを引き込んで
しまう

だけど

next-hop-self の設定は、
相手に経路広告する際にも
設定しましょう

AS1からAS3へ広告される経路

Prefix	Nexthop	AS Path
172.16.1.0/16	192.168.0.1	AS1
172.16.2.0/16	192.168.0.2	AS1 AS2

AS1がnext-hop-selfの設定をしていなかったのが原因

フラップダンプニング（ルートダンプニング）

回線のup/downなどにより、BGPの
経路がフラップしている場合には、
そのUpdateパケットが頻繁に発生し、
ルータのCPUを無駄に消費してしまう．
それを回避するために、ある閾値を
境に、その経路を抑制するしくみ

Penaltyのカウント方法

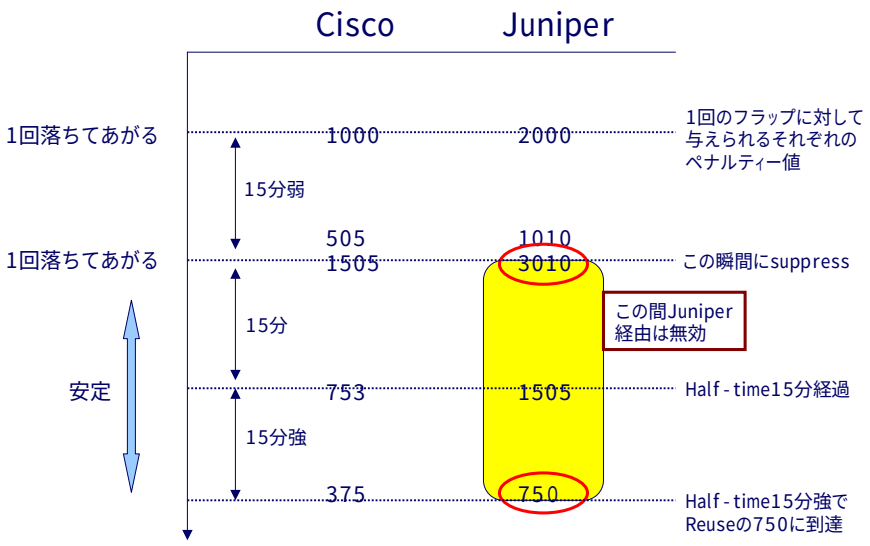
<Cisco>	
Penalty	1000/1Flap
<Juniper>	
* Route is withdrawn	1000
* Route is readvertised	1000
* Route's path attributes change	500

デフォルトのpenalty値

<Cisco>	
half-life:	15 minutes
reuse:	750
suppress:	2000
max-suppress-time:	60 minutes
<Juniper>	
half-life:	15 minutes
reuse:	750
suppress:	3000
max-suppress-time:	60 minutes

1. half-life: 加算されたペナルティー値が半分になるまでの時間
2. reuse: この値までペナルティー値が減れば、再度その経路を広告するという設定値
3. suppress: ペナルティー値の合計がこの値を超えた時点で、制限をかけはじめる
4. max-suppress-time: 制限をかける時間として設定する最大の時間

BGPフラップの例



マルチベンダ関連

マルチベンダ環境

- ベンダの仕様によって、挙動が異なる場合がけっこうある
  - BGPのベストパスセレクションの動作が違う
    - チューニングが必要なときもある
    - 場合によっては、経路選択時に障害も起こりうる
  - 経路表の持ち方が異なる
- ちゃんと事前に検証を行って確認しましょう
  - 実網で判明した場合には、その都度検討
    - OSのVersionUpに伴い、実装変更が発生し、挙動が異なってくる場合もある

BGP Hold-time

- 実装が異なる
  - Juniper → Keepalive：30秒、Holdtime：90秒
  - それ以外 → Keepalive：60秒、Holdtime：180秒
- Hold-timeは、2つのBGPピアの間で異なっていたら、値の小さいほうにあわされるので注意
  - Openメッセージの中にふくまれていて、最初にBGPピアを確立する際のネゴシエーションで決定される

Juniper ←→ Cisco　　の場合には
keep-alive 30秒 / Hole-time90秒　になる

BGPのバージョンは、最初のOPNEメッセージのやり取りの段階で、不一致の場合にはピア自体が張れない
（例えば、バージョン1とバージョン4）
他、OSPFのarea etc...

next-hop-selfの実装

- Cisco
  - 記述しないと有効にならない
    - eBGPから受信した経路をiBGPに流す場合に、「next-hop-self」を記述すると有効
  - iBGPピア同士で書いても、有効にならない
- Juniper
  - 記述しないと有効にならない
    - eBGPから受信した経路をiBGPに流す場合に、「next-hop-self」を記述すると有効（Ciscoと同様）
  - iBGP同士においても、記述すると有効になってしまうので注意
    - ルーティングループを引き起こす可能性がある

send-communityの実装

- Cisco
  - ピア相手に対して「send-community」を記述しないとコミュニティが伝播しない
    - 例えば、no-exportなどの経路を内部で利用し、リフレクタの階層構造を用いて経路配信していた場合、上流向けに対して「send-community」がある箇所ではずれてしまっただけで（outフィルタのポリシーも通過すれば）外部にそのまま流れる
- Juniper
  - デフォルトでコミュニティー情報をわたす
  - 特に設定は必要ない

Route-Refresh メッセージ

- BGPのメッセージType5 = ROUTE_REFLESH
- RFC2918で規定、相手から全BGP経路情報の再送を要求
- BGPのOPNEメッセージのやり取り時に、各々自分がどのタイプが受け入れ可能かを通知する
  - 実際には、「BGP TYPE1 OPENメッセージ」の中の、「Optional Parameters フィールド」の値の中の、「Capability Code」に記述
    Capability Code = 2 : rfc
    Capability Code = 128 : cisco (128以上はベンダ独自使用領域)
  - 最近は、この2種類両方とも実装している、あるいは実装中というベンダが多い
- Juniper、RiverStone はデフォルトでキャッシュ方式を採用している
  - 各ピアから受信した経路をキャッシュしている
  - Ciscoの場合など、「soft-reconfiguration inbound」でキャッシュ
    - 事前にreceive-route を確認してからピアを確立するなどにも使える

BGPのpassiveモードの実装

通常はどちらか一方からのTCP179ポートに対するOPENメッセージによって、コネクションが開設される

Passiveと設定してあると、自分からコネクションをOPNEしようとせず、相手からのコネクション開設を待っている

Passive設定は、JuniperやRiverstoneが対応
『注意』 両方passiveだと、永久にBGPピアが確立しない

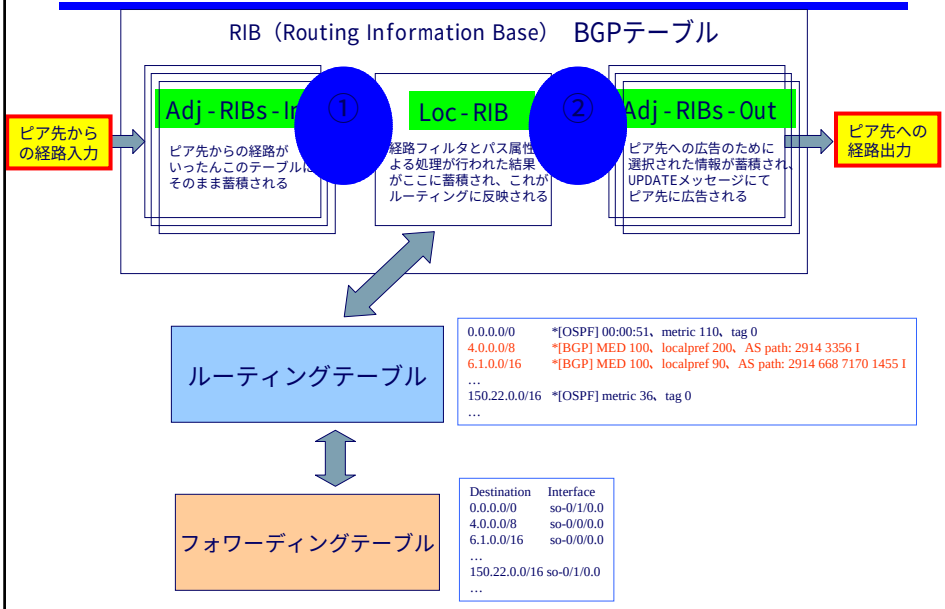
経路管理のされ方 (1)

- ルーティングテーブルのみ：Juniper、RS
  - OSPFもBGPも全て1つのルーティングテーブルで管理されている
  - ルーティングテーブル上でベストではないと、BGPで配信されない
    - 例えばJuniperでは、「advertise-inactive」というコマンドで、OSPFなどBGP以外のプロトコルがベストとなっていても、BGP上で最もベストな経路が配信可能となる
  - BGP以外の経路が配信されてしまう可能性があるので注意
    - Outのpolicy変更は、IPルーティングテーブル全体に適応される
      - match protocol ospf などでマッチしてしまうと、その経路がBGPで配信されてしまう
    - 逆にInのpolicyは、BGPピアに対しては、BGP経路しか受信しないので、BGPの経路に対してのみ適応される → 他のプロトコルの経路を受け取る心配はない

経路管理のされ方 (2)

- ルーティングテーブルとBGPテーブルがある
  ：Cisco、Foundry
  - BGP経路の制御は、BGPテーブルで行われる
  - BGPテーブル上のベスト経路が、ピア先に経路配信される
  - ルーティングテーブルとBGPテーブルの関係
    - BGP経路をピアから受信し、ベストパスを選択する
    - 同時に、そのBGPテーブルでベストとなっている経路を、自身のルーティングテーブルに渡す
    - 渡されたあと，プロトコルディスタンスで、もっとも優先される経路がルーティングテーブルに正式にエントリーされる (OSPFで同じ経路が存在する場合には、BGPテーブルのみでベストパスとしてエントリーされ、ルーティングテーブルにはのらない ← プロトコルディスタンスの差)
  - BGPピアに配信される経路は、BGPテーブルを参照する
    - 通常のルーティングテーブルでベストになっていなくてもOK

BGPのRIB管理と各テーブルの関係

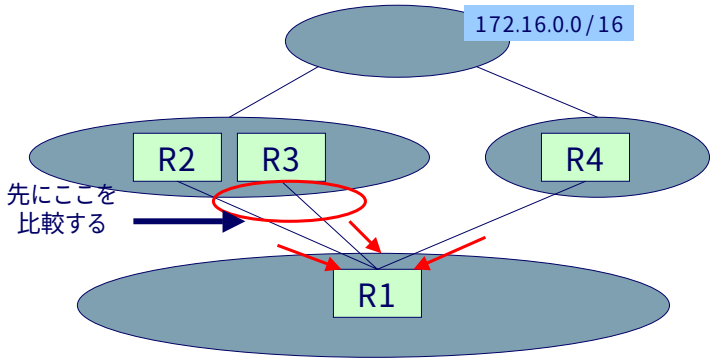


MEDについて

- MED (MULTI_EXIT_DISC)
  - 1つの隣接ASとの間に複数回線がある場合、MEDの値を互いに交換することによって、優先順位をつけることができる
  - 異なるAS間では通常比較の対象にはならない
    - always-compare-med で、異なるAS間でも比較することが可能
  - 値の小さいほうを優先する
  - 2つ以上のASをまたがっては広告されない
    - eBGPピアに対してUpdateを送信する場合には、MED属性は削除される
- MED値がついていない場合には、ベンダーによって解釈が異なる
  - MED = 0 or NULL (もっとも優先される)
  - MED = MAX値 (もっとも値が大きいということは、使われないということ)
  - ベンダによっては、何も値がついていない経路に付与するMED値を変更することが可能

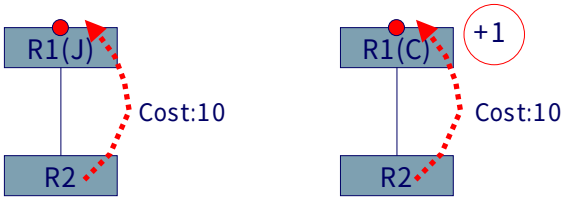
bgp deterministic-med

- BGPピア先から受信した経路のうち、先に同一ASの経路をまず比較して、そのあとに異なるAS間の経路を比較する
  - Ciscoは、デフォルトでは有効になっていない
  - Juniperは、cisco non-deterministic-med を入れると、Ciscoと同様に受信した順に比較するようになる



OSPFのループバックのコスト

- ループバックアドレスの見え方が異なる
  - Cisco:
    - R1がCiscoの場合、R2から見たR1のLoopbackのコストは10+1=11
  - Juniper:
    - R1がJuniperの場合、R2から見たR1のLoopbackのコストは10
- IGPコストで経路選択をしている場合、あるいは、iBGP Multipath などを適応している場合には注意が必要



セキュリティ関連、他

- フィルタリング
- flow monitoring
- BGP Max Prefix、Prefix Limit
- MD5 (Message Digest 5)
- Unicast RPF
- TTL Hack (GTSM)
- IRRと経路フィルタ
- Black Hole ルーティング

セキュリティ設計

- 何を、どのように、何処を、どの程度　守りたいのか
  を明確にする
  - 不要なパケットが外部から来るのを可能な範囲でブロックしたい
  - 過った経路情報がお客さんから来るのをexactにブロックしたい
- それに対する対処を実施する
  - 手法は色々存在するので、その中で適切な対処を行う

フィルタリング

- 2種類、それぞれ2方向 (in/out) のフィルタ
  - 経路フィルタ
    - 外部から自AS内に対して広報されてくる経路をフィルタ (in)
    - 自ASから外部ASに対して広報する際に適応するフィルタ (out)
  - パケットフィルタ
    - 外部から自AS内に対して通過しようとするパケットをフィルタ (in)
    - 自ASから外部ASに対して通過しようとするパケットをフィルタ (out)

経路フィルタ

- In方向 (外部AS→自AS)
  - 共通
    - 自AS経路に加えて、Privateアドレス、マルチキャスト、リンクローカルなど RFC3330で定義された特殊用途アドレス等を必要に応じて遮断
  - 上流・ピア
    - 細かい経路は受け取らない (/24よりも細かいもの など)
    - ピアに対しては、基本はAS_PATHフィルタ and/or Prefix Filterでブロック
    - 異常な経路数に対しては、上限を設けておく (max-prefixなどの複合)
  - 顧客
    - 申告ベースのPrefixのみ (exact-much or 該当Prefix内) を受け取る
- Out方向 (自AS→外部AS)
  - 共通
    - 内部で利用している細かい経路などは、ちゃんとはじくような設定
    - RFC1918な経路を利用している際には、それをはじくフィルタを設定
    - remove-private-AS などの適応など
  - 上流・ピア
    - 自分と顧客経路のみを配信するようなAS_PATHフィルタ
    - コミュニティを利用しての経路広告も可能

パケットフィルタ

- 自分の身は自分で守る
- 相手に出すパケットは、迷惑のかからない程度にフィルタをしておく
- 自分が経路を広報していなければ、パケットはやってこない？

- In方向（外部AS→自AS）
  - ソースがPrivateアドレス、マルチキャストアドレスなどのパケットはフィルタ（uRPFを複合させて、見割り当て空間のフィルタを適応するとなお良い）
  - ソースが自ＡＳアドレスのパケットは注意
- Out方向（自AS→外部AS）
  - 自AS内でちゃんと経路を管理していれば、特段必要ないはず
    - 顧客との接続部分ではじいてしまうなど、入り口の部分ではじくこごも可能
    - プラスαで、予防保全的にFilterを適応しておけば完璧

Flow Monitoring

- PacketをMonitoringすることにより、どの対置からどの対置へPacketが流れているのかを統計的に解析し、ネットワークのデザインにフィードバックする
  - Flowコレクタは、市販のものからFreeのflow-toolsなど様々

Source/Dest IP
Source/Dest Port
Source/Dest AS Number（origin-as or peer-as）
Packet Count、 Byte Count etc..

こういった情報を
UDPのPacketでコレクタ
に向けて送信する

- Netflow
  - Switching 方式として、Ciscoにより開発されたもの
  - Netflow Switching と呼ばれている
- cflowd : Netflowと同じflow export
- sFLOW
  - RFC3176
- IPFIX : IP Flow Information Export
  - IETFの IPFIX-WG にて検討中
  - http://www.ietf.org/html.charters/ipfix-charter.html
  - http://ipfix.doit.wisc.edu/

Netflow

- Netflow
  - 現在 Version 5 が広く使われている
  - Version8は、Version5のFlow情報をaggregateして転送
  - Version9は、特定のFormatに従って必要な情報を抽出してFlowをExportすることが可能

```
interface GigabitEthernet0/0
 ip route-cache flow sampled

ip flow-export source Loopback0
ip flow-export version 5 origin-as
ip flow-export destination 192.168.1.1 9996
ip flow-sampling-mode packet-interval 10000
```

「sample」を書いた場合には、
全てのPacket情報を種族せずに
Intervalをあけて取得する

cflowd

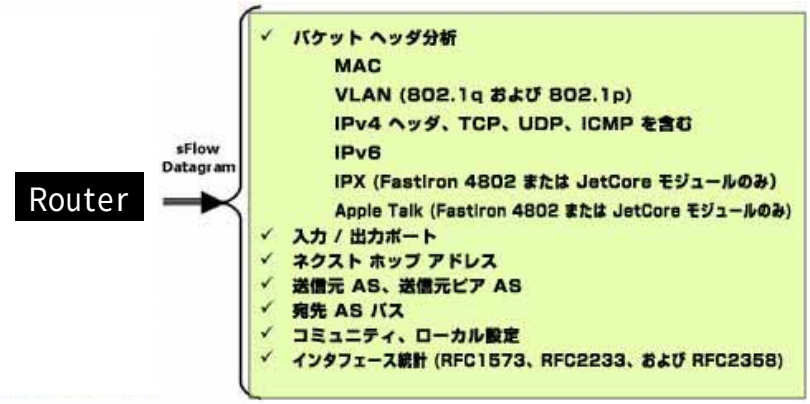
```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                filter {
                    input Cflowd;
                    output Cflowd;
                }
                address 202.249.2.131/24;
            }
        }
    }

firewall {
    filter Cflowd {
        term 999 {
            then {
                sample;
                accept;
            }
        }
    }
}
```

```
forwarding-options {
    sampling {
        input {
            family inet {
                rate 10000;
            }
        }
        output {
            cflowd 192.168.1.1 {
                port 9996;
                engine-id 0;
                version 5;
                autonomous-system-type origin; or peer
            }
        }
    }
}
```

sFLOW

- Sampling Flow
- RFC3176にて定義されているFormatに従ってFlowをExport

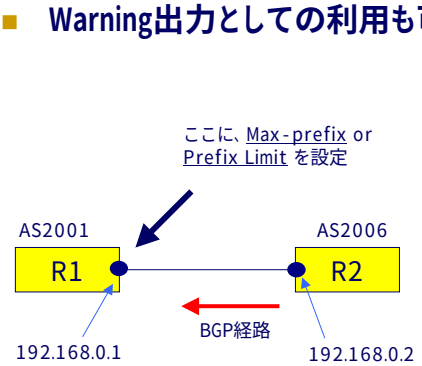


http://www.foundry.ne.jp/technologies/sFlow/definition.html より

（参考）IRS Workshop : http://www.bugest.net/irs/docs_20060922/

BGP Max Prefix、Prefix Limit

- 受信経路数の上限を設定し、想定以上の経路を遮断
  - Peer先等からの経路受信時に設定する
- Warning出力としての利用も可能



Warningを
出力する%
上限値

■ Ciscoの場合（R1）

```
router bgp 2001
 neighbor 192.168.0.2 remote-as 2006
 neighbor 192.168.0.2 max-prefix 1000 50
```

■ Juniperの場合（R1）

```
protocols {
    bgp {
      group iw {
        family-inet {
          unicast {
            prefix-limit {
              maximum 1000
              teardown 50 idle-timeout forever
```

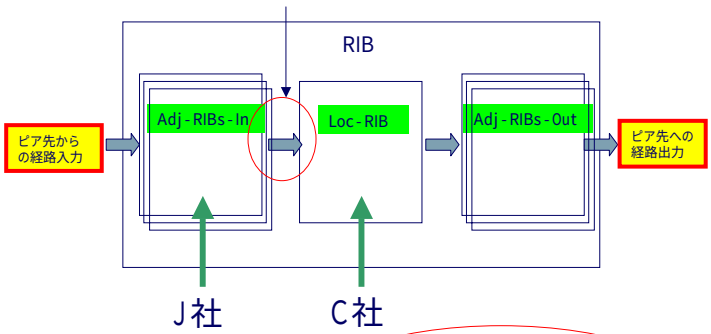
BGP Max Prefix、Prefix Limit

- http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a00801545d5.html
- http://www.juniper.net/techpubs/software/junos/junos81/swconfig81-routing/html/routing-summary41.html
- 適応RIBの違いに注意
  - Cisco
    - → Loc-RIB
  - Juniper
    - → Adj-RIBs-in

```
IW(config-router)#neighbor iw maximum-prefix 200000 ?
  <1-100>         Threshold value (%) at which to generate a warning msg
  restart         Restart bgp connection after limit is exceeded
  warning-only    Only give warning message when limit is exceeded
  <cr>
        デフォルト＝75% で warningが出力される
```

```
IW# set protocols bgp group iw unit 0 family inet unicast prefix-limit ?
Possible completions:
+ apply-groups        Groups from which to inherit configuration data
  maximum             Maximum number of prefixes from a peer (1..4294967295)
➢teardown             Clear peer connection on reaching limit
        ➢<limit-threshold>    Percentage of prefix-limit to start warnings (1..100)
        ➢idle-timeout         Timeout before attempting to restart peer
```


BGP Max Prefix、Prefix Limit



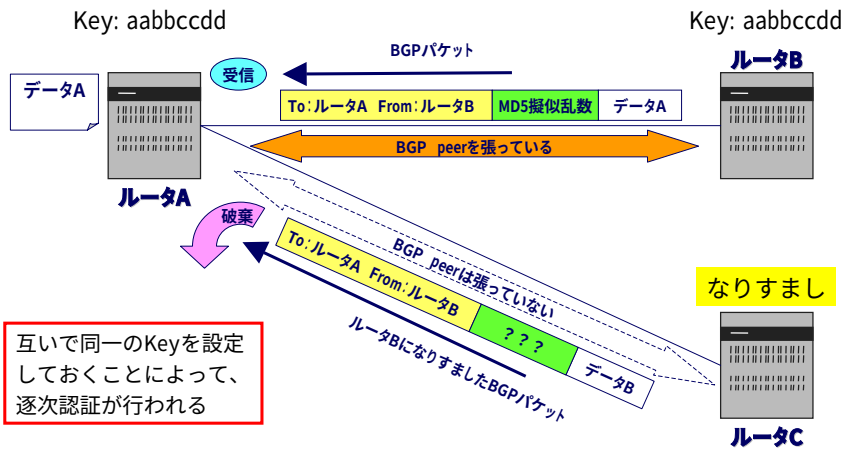
max-prefix以外のFilterを適応している場合には、その該当Filter適応後に、上限値を超えている場合には、limit制限がかかる

➔ Capability Option への拡張をIETFで議論

MD5

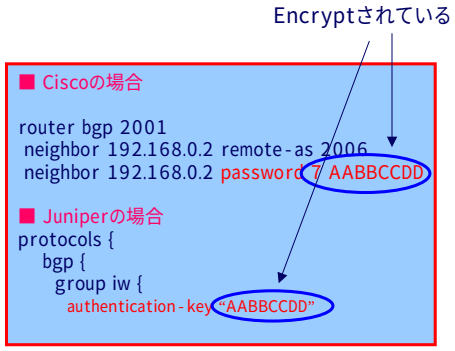
- Message Digest 5
- BGPのPeerに設定することにより、経路交換やPeerの確立時における安全性を向上させる技術の1つ
  - 認証やデジタル署名などに使われるハッシュ関数(一方向要約関数)のひとつ、認証アルゴリズム
  - 両端で同一なキーを設定し、MD5アルゴリズムを用いて変換された128bitの固定長のbit列を両端で比較することで、改ざんされていないか確認
  - MD2、MD4 → MD5
  - 簡潔さ、安全性、速度を重視
    - SHA-1(Secure Hash Algorithm) : 160bit
  - RFC1321

MD5認証イメージ



MD5の設定

- 設定的には特に難しい設定はない
- Keyに使用可能な文字、不可能な文字については、対抗のルータそれぞれ事前に調査の上適応するのが望ましい
  - 特殊用途に予約しているような文字はなるべく使わない
    - お互いの機種が変更になる可能性もあるので



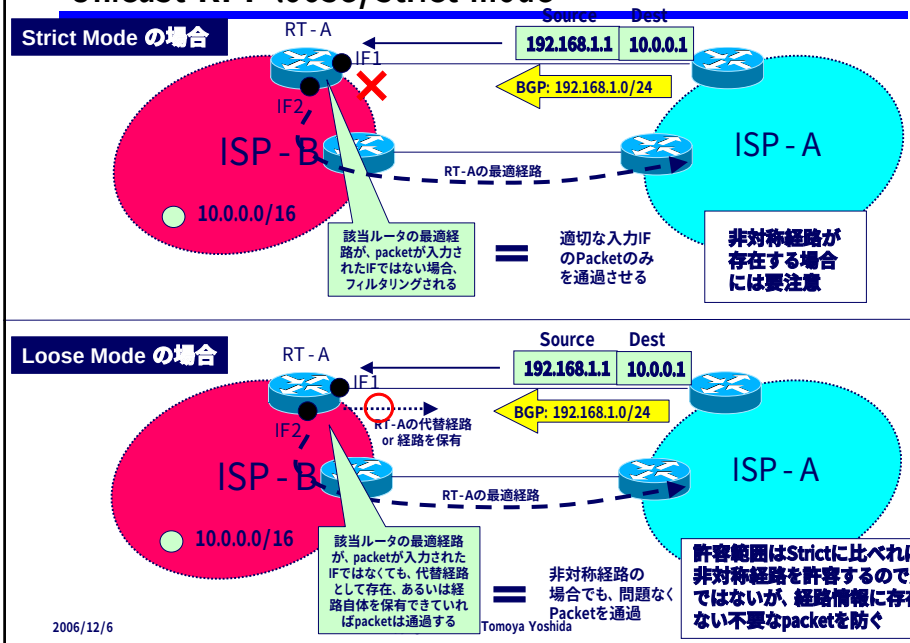
Unicast RPF

「uRPF」＝「unicast Reverse Path Forwarding」

・経路情報を利用した Ingress Filter の手法で、不要なIngress Packetを取り除くことが可能
・既存のDdos攻撃を未然に「動的に」フィルタ可能な技術
・RFC3704 : Ingress-Filtering for Multi-homed Networks

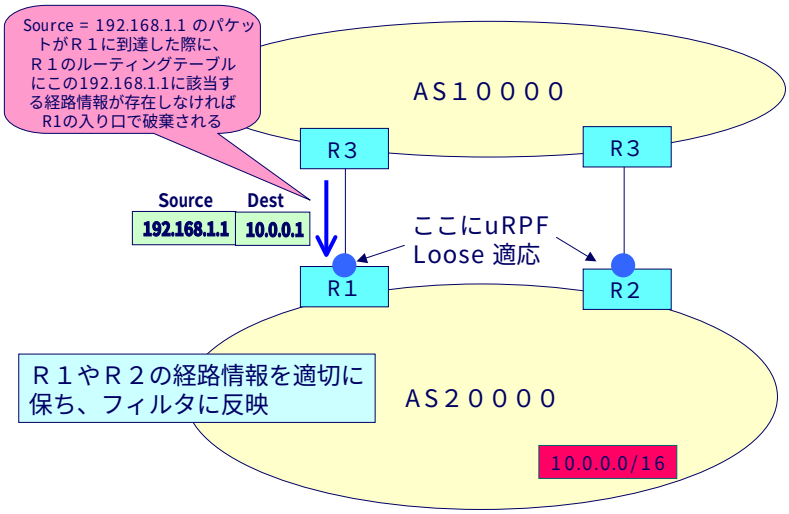
Loose mode	Feasible mode	Strict mode
パケットのソースアドレスがルーティングテーブルにあるかどうかのみを確認し、ルーティングテーブルに存在する場合には通過、存在しない場合には、フィルタされる。 厳密に言うと Reverse Path Forwarding ではなく、また Default経路の処理をどうするかでさらに扱いが分かれる。	パケットのソースアドレスについてFIBではなくRIBを参照する。 パケットを受け取ったインタフェースが経路的にbestでなくとも、代替経路として利用される可能性のあるインタフェースなら、パケットは通過可能。 非対称ルーティングでも、経路がアナウンスされていれば大丈夫だが、適応場所に依存する。	パケットのソースアドレスについてFIBを参照 パケットを受け取ったインタフェースがforwardするべきインタフェースなら、そのパケットは通過可能となる。 Strict Reverse Path Forwarding ではパスの対称性があることを前提としているが、適応場所に依存する (運用技術的には解決可能とされている) 。

Unicast RPF loose/strict mode



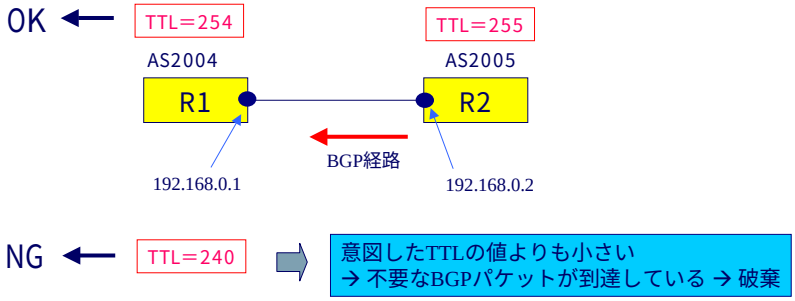
2006/12/6 Tomoya Yoshida

Unicast RPF 適応イメージ (Loose Mode)



TTL Hack （GTSM)

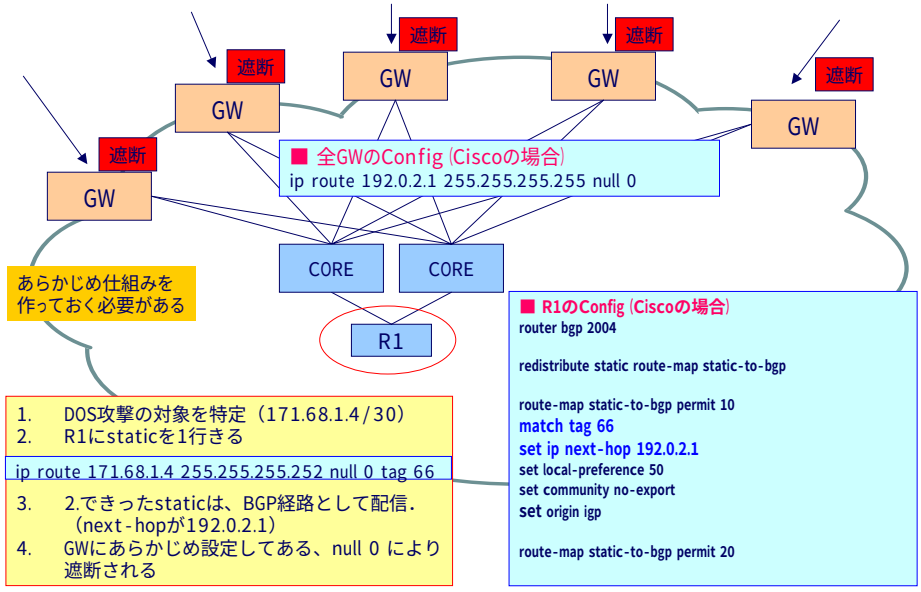
- Generalized TTL Security Mechanism
- RFC3682
- 事前に設定したTTLの値よりも小さな値のPacketをはじく



IRRと経路フィルタ

- Internet Routing Registry
  - BGPの経路情報やASのポリシーを記述したデータベース
  - BGP経路の信憑性確認やコンタクトポイントの検索に有効
    - 何か経路に異常が発生したら、まずIRRの情報を参照するのが一般的
- IRRToolSetを用いたPrefixフィルタの生成
  - http://www.janog.gr.jp/meeting/janog9/pdf/yoshida_janog9.pdf
- IRRの情報から様々なConfigを書くことが可能

Black Hole Routing



1. DOS攻撃の対象を特定（171.68.1.4/30）
2. R1にstaticを1行きる

ip route 171.68.1.4 255.255.255.252 null 0 tag 66

3. 2.できったstaticは、BGP経路として配信.（next-hopが192.0.2.1）
4. GWにあらかじめ設定してある、null 0 により遮断される

ご清聴ありがとうございました

ISPバックボーンネットワークにおける
経路制御設計　～実践編～

吉田友哉 yoshida@ocn.ad.jp
NTTコミュニケーションズ（株）
1E7D 79AD C610 B5F2 A94E　7FF4 F4AC A722 329C 3DE8

参考資料

BGPのベストパス選択一覧表（主な選択）

上から順に経路比較を実施し、ベスト経路が選択

優先度	属性	内容
1	NEXT_HOP	ネクストホップへの到達性があること
2	WEIGHT	Cisco固有のパラメータで，値の大きな経路を優先
3	LOCAL_PREF	Local Pref値の大きな経路を優先
4	LOCAL	Localで生成された経路を優先
5	AS_PATH	AS-PATH長の短い経路を優先
6	ORIGIN	Origin属性が，igp>egp>incompleteの順に優先
7	MED	MED値が小さい経路を優先
8	PEER_TYPE	iBGPよりもeBGP経由で受信した経路を優先
9	IGP METRIC	IGPのMetric値が小さい（近い）パスの経路を優先
10	ROUTER_ID	Router-IDが最も小さい経路を優先

Protocol Distance / Route Preference

CiscoとJuniperにおける、プロトコルディスタンス（ルートプリファレンス）値の違い

■Cisco

プロトコル	Preference値
Connected	0
Static	1
EBGP	20
EIGRP（内部）	90
IGRP	100
OSPF	110
ISIS	115
RIP	120
EIGRP（外部）	170
IBGP	200

■Juniper

プロトコル	Preference値
Connected	0
Static	5
MPLS	7
OSPF internal	10
ISIS level-1 internal	15
ISIS level-2 internal	18
RIP	100
P-to-P	110
OSPF external	150
ISIS level-1 external	160
ISIS level-2 external	165
BGP	170

アルゴリズム毎に整理した代表的なルーティングプロトコル

ディスタンスベクターアルゴリズム（ＲＩＰ）

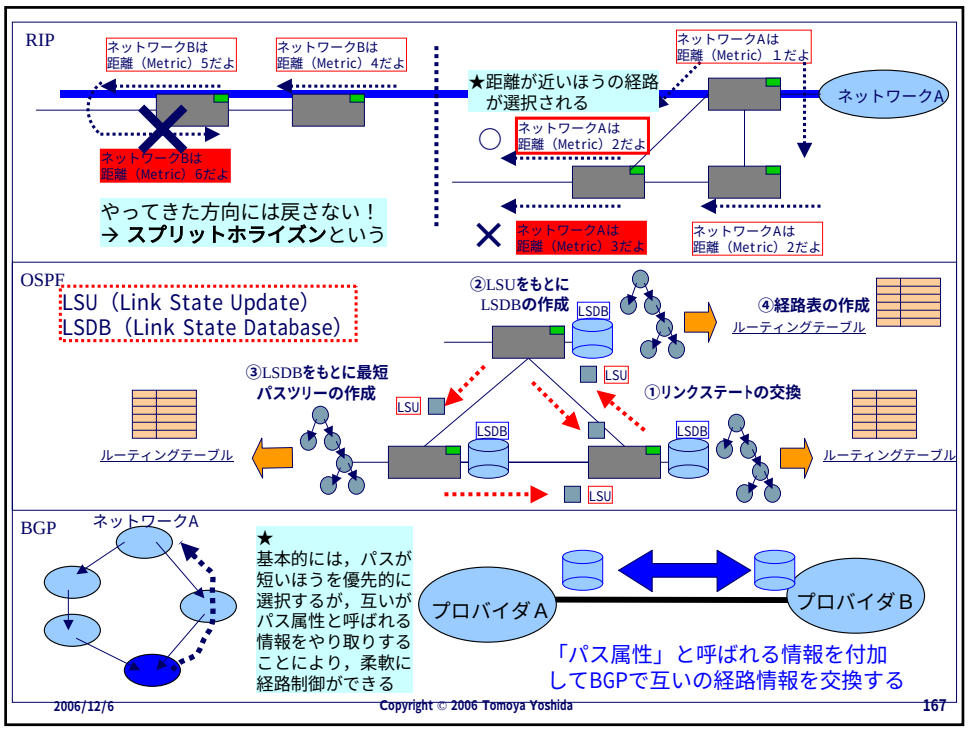
- ホップする数（距離）によって経路が選択される
- 隣接ルータ同士で経路情報を交換することでネットワーク情報を知る
- 他のルータから受信したルーティングテーブルに自分が直接接続している ネットワークを加え，受信したインタフェース以外のインタフェースに流す

リンクステートアルゴリズム：　（ＯＳＰＦ）

- それぞれのルータが自分の接続しているネットワークについての情報等 （リンクのステータス）をネットワーク全体に通知する
- 各ルータで自分を起点とした共通のトポロジーデーターベースを持つ

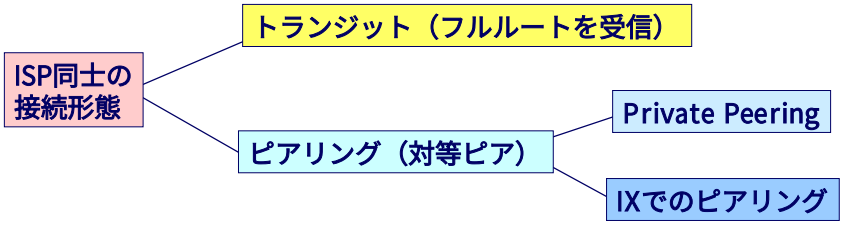
パスベクターアルゴリズム：　（ＢＧＰ）

- 経路情報が伝わっていく際に，経路情報にパス属性と呼ばれる付加情報が ついて伝わる



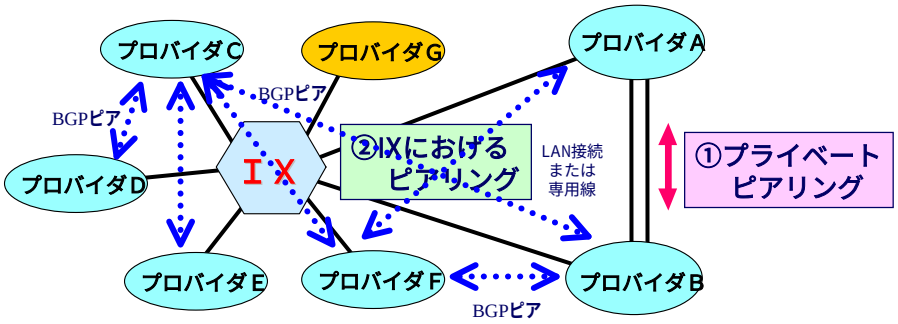
ISPの接続形態

「ピアリング」という言葉には，
「ISP同士が対等な関係でBGPで結ぶ」という狭義な意味と，
「BGPのセッションをはる」という広義な意味が存在する．
どちらの意味かは前後の文脈で判断するのだが、
最近では前者の意味で使われることが多い．
ここで言っているのも前者の意味である．



ISPの接続形態

ISP同士のピアリング接続イメージ図
IXに加入したからといって、全ての相手とピアリングできる
わけではなく、実際には個別にピアリング交渉をしピアをはる

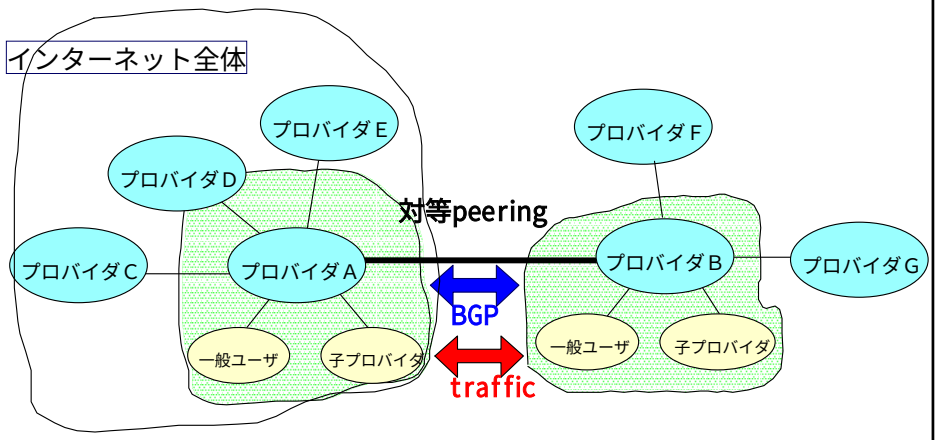


マルチラテラル/バイラテラル

現在ほとんどがバイラテラルモデルを採用して
おり、IX自体はL2SWであるのが一般的

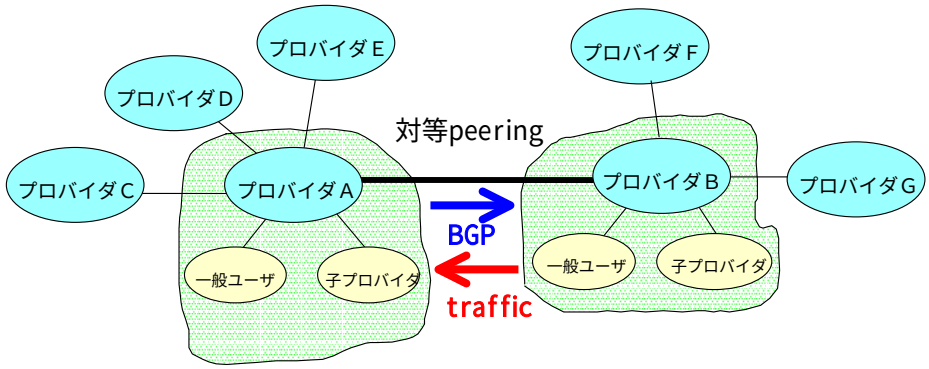
- マルチラテラル
  - IXに加入すると、参加者全員がお互いとピアリング
  - 例：HKIX（香港）．．ルートサーバ等
- バイラテラル
  - IXへの加入と，ピアリングは無関係で、IXとしてはピアリングには関知しない。
  - 実際にはほとんどこのモデル

プロバイダＡとプロバイダＢが対等ピアリングをする場合

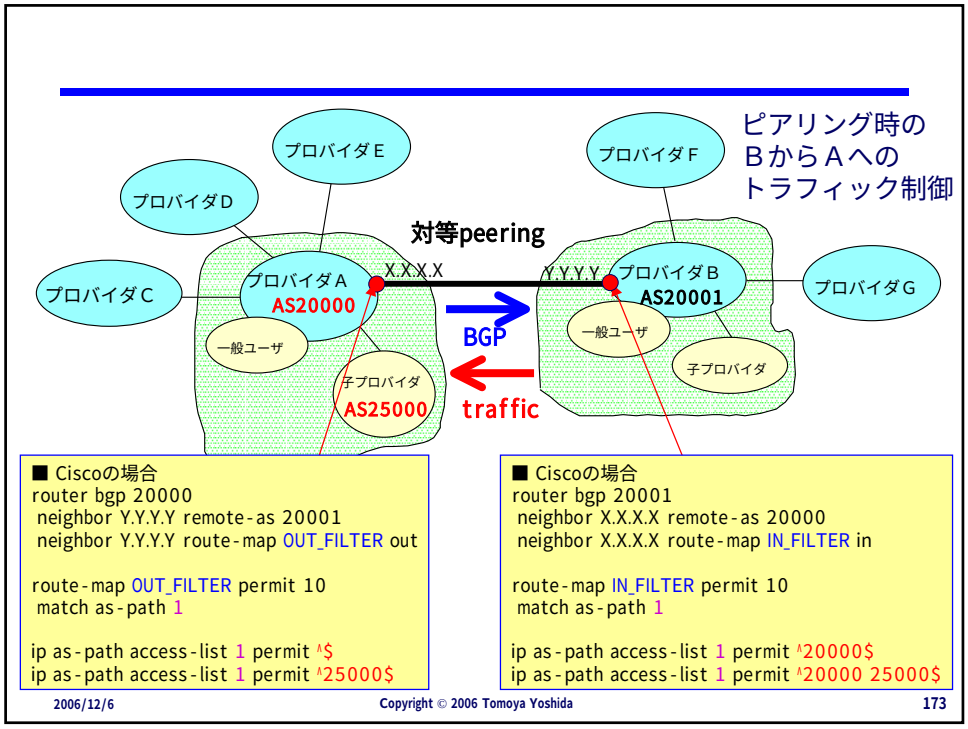


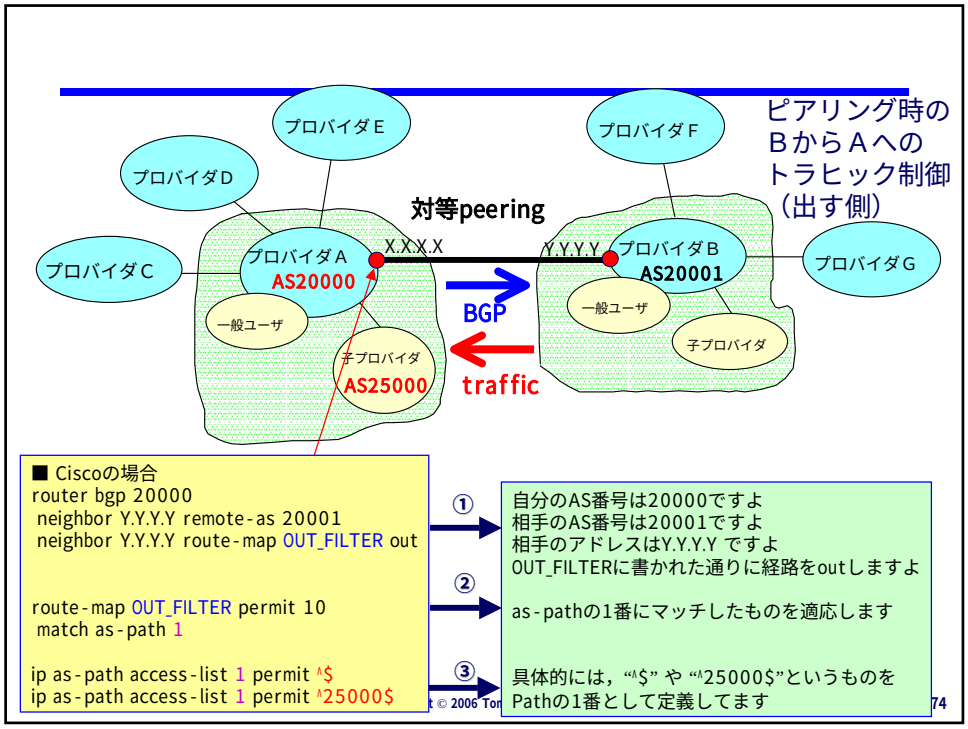
プロバイダＡとプロバイダＢの間の回線のトラヒックは
｛プロバイダＡとそのお客様（一般ユーザと子プロバイダ）｝と
｛プロバイダＢとそのお客様（一般ユーザと子プロバイダ）｝間のものが流れる

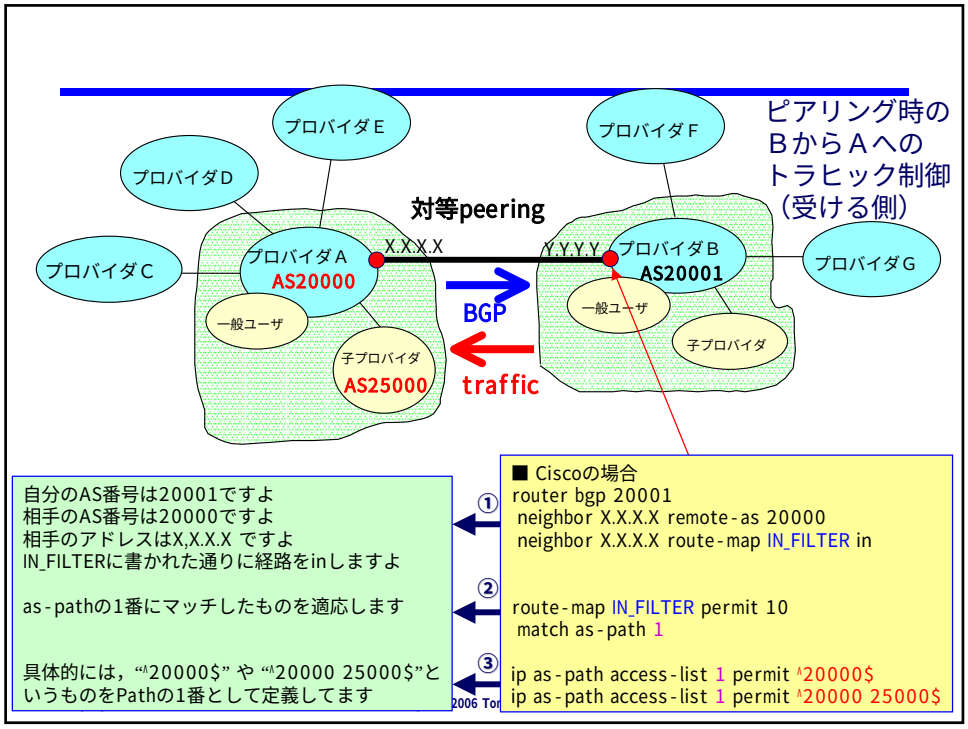
ピアリング時のＢからＡへのトラフィック



・プロバイダＡからは｛プロバイダＡとそのお客様（一般ユーザと子プロバイダ）｝の経路情報を
プロバイダＢに流す。
・プロバイダＢはその経路情報は外（プロバイダＦやプロバイダＧ）には流さない。
・よって、ＢからＡに流れるトラフィックは｛プロバイダＢとそのお客様（一般ユーザと子プロバイダ）｝
から｛プロバイダＡとそのお客様（一般ユーザと子プロバイダ）｝へのトラフィックだけとなる。


ピアリング時の
Bから Aへの
トラフィック制御
プロバイダ E
プロバイダ F
プロバイダ D
対等peering
プロバイダ C
プロバイダ A
AS20000
X.X.X.X
Y.Y.Y.Y
プロバイダ B
AS20001
プロバイダ G
BGP
一般ユーザ
一般ユーザ
子プロバイダ
子プロバイダ
AS25000
traffic
■ Ciscoの場合
router bgp 20000
neighbor Y.Y.Y.Y remote-as 20001
neighbor Y.Y.Y.Y route-map OUT_FILTER out
route-map OUT_FILTER permit 10
match as-path 1
ip as-path access-list 1 permit ^$
ip as-path access-list 1 permit ^25000$
■ Ciscoの場合
router bgp 20001
neighbor X.X.X.X remote-as 20000
neighbor X.X.X.X route-map IN_FILTER in
route-map IN_FILTER permit 10
match as-path 1
ip as-path access-list 1 permit ^20000$
ip as-path access-list 1 permit ^20000 25000$
2006/12/6
Copyright © 2006 Tomoya Yoshida
173


ピアリング時の
Bから Aへの
トラヒック制御
（出す側）
プロバイダ E
プロバイダ F
プロバイダ D
対等peering
プロバイダ C
プロバイダ A
AS20000
X.X.X.X
Y.Y.Y.Y
プロバイダ B
AS20001
プロバイダ G
BGP
一般ユーザ
一般ユーザ
子プロバイダ
子プロバイダ
AS25000
traffic
■ Ciscoの場合
router bgp 20000
neighbor Y.Y.Y.Y remote-as 20001
neighbor Y.Y.Y.Y route-map OUT_FILTER out
route-map OUT_FILTER permit 10
match as-path 1
ip as-path access-list 1 permit ^$
ip as-path access-list 1 permit ^25000$
①
自分のAS番号は20000ですよ
相手のAS番号は20001ですよ
相手のアドレスはY.Y.Y.Y ですよ
OUT_FILTERに書かれた通りに経路をoutしますよ
②
as-pathの1番にマッチしたものを適応します
③
具体的には，“^$” や “^25000$”というものを
Pathの1番として定義してます
74



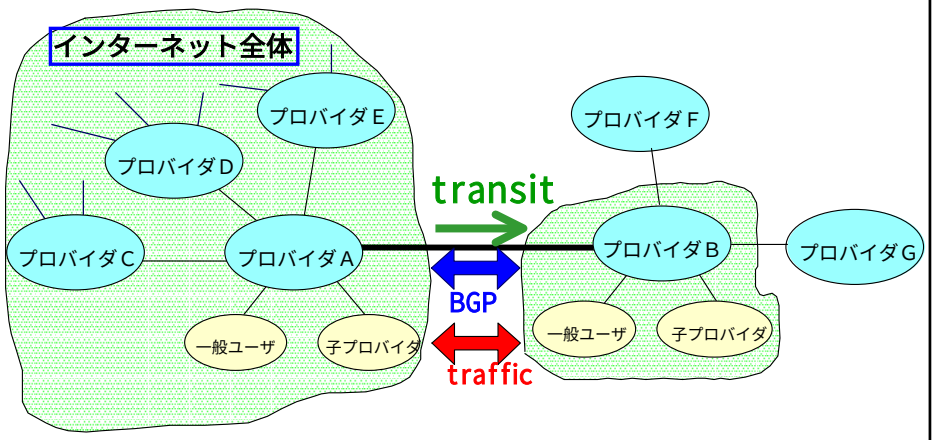
■ Juniperの場合

```
protocols {
 bgp {
  group PEER {
   type external;
   local-address X.X.X.X;
   peer-as 20001;
   export OUT_FILTER;
   neighbor Y.Y.Y.Y;
   }
 }
}
policy-options {
 policy-statement OUT_FILTER {
  term LOCAL_OUT {
   from {
    protocol bgp;
    as-path [ MY-LOCAL AS25000];
    }
   then {
    accept;
   }
  }
 }
 as-path MY-LOCAL "()";
 as-path AS25000 25000;
}
```

■ juniperの場合

```
protocols {
 bgp {
  group PEER {
   type external;
   local-address Y.Y.Y.Y;
   peer-as 20000;
   import IN_FILTER;
   neighbor X.X.X.X;
   }
 }
}
policy-options {
 policy-statement IN_FILTER {
  term AS20000_PEER_IN {
   from {
    protocol bgp;
    as-path [ AS20000-1 AS20000-2];
    }
   then {
    accept;
   }
  }
 }
 as-path AS20000-1 20000;
 as-path AS20000-2 “20000 25000”;
}
```

プロバイダＢがプロバイダＡからトランジットを受ける場合



プロバイダＡとプロバイダＢの間の回線のトラヒックは
｛インターネット全体｝と
｛プロバイダＢとそのお客様（一般ユーザと子プロバイダ）｝間の通信が流れる

プロバイダＢはプロバイダＡのお客様

IXなどでPolicyをまとめたConfig例

■ Ciscoの例

```
router bgp 2004
 neighbor IX1-Main peer-group
 neighbor IX1-Main next-hop-self
 neighbor IX1-Main route-map ix1-main-out
 neighbor IX1-Backup peer-group
 neighbor IX1-Backup next-hop-self
 neighbor IX1-Backup route-map ix1-backup-out
...
 neighbor 192.168.1.10 peer-group IX1-Main
 neighbor 192.168.1.11 peer-group IX1-Backup
 neighbor 192.168.1.12 peer-group IX1-Backup
 neighbor 192.168.1.13 peer-group IX1-Main
 neighbor 192.168.1.14 peer-group IX1-Main
...
ip as-path access-list 10 permit ^$
ip as-path access-list 10 permit ^2008$
ip as-path access-list 10 permit ^2008 2009$
...
route-map ix1-main-out permit 10
 match as-path 10
 set metric 300

route-map ix1-backup-out permit 10
 match as-path 10
 set metric 310
```

■ ポイント1

通常どこのISPに対しても自分から広報する経路は一緒なので、メインとバックアップの2つに分けてグループを作っておく

■ ポイント2

作成したグループを用いて、実際の相手のアドレスに対してポリシーを適応させていく．そのピアをメイン回線として適応するなら、IX1-Main

■ ポイント3

もらう経路はそれぞれ違うので、それは直接相手のネイバーアドレスに対してroute-map を定義する
（例） neighbor192.168.1.10
route-map as-4713-in in