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～

NTTコミュニケーションズ(株)
吉田友哉〈yoshida@ocn.ad.jp〉

本チュートリアルの内容

- 全般　（20分）
- OSPF設計　（40分）
- BGP設計前半　（30分）

- BGP設計後半　（40分）
- マルチベンダ環境（30分）
- その他　（10分）

本チュートリアルの目的

- 実際に，どういった事を考えて経路制御設計を行う必要があるのか，そのポイントを押さえて頂く
- 実際のネットワークに即した形で，具体例や数値，Configなどを見ながら考える
- 自分のネットワークに参考になる部分は是非取り入れて頂く

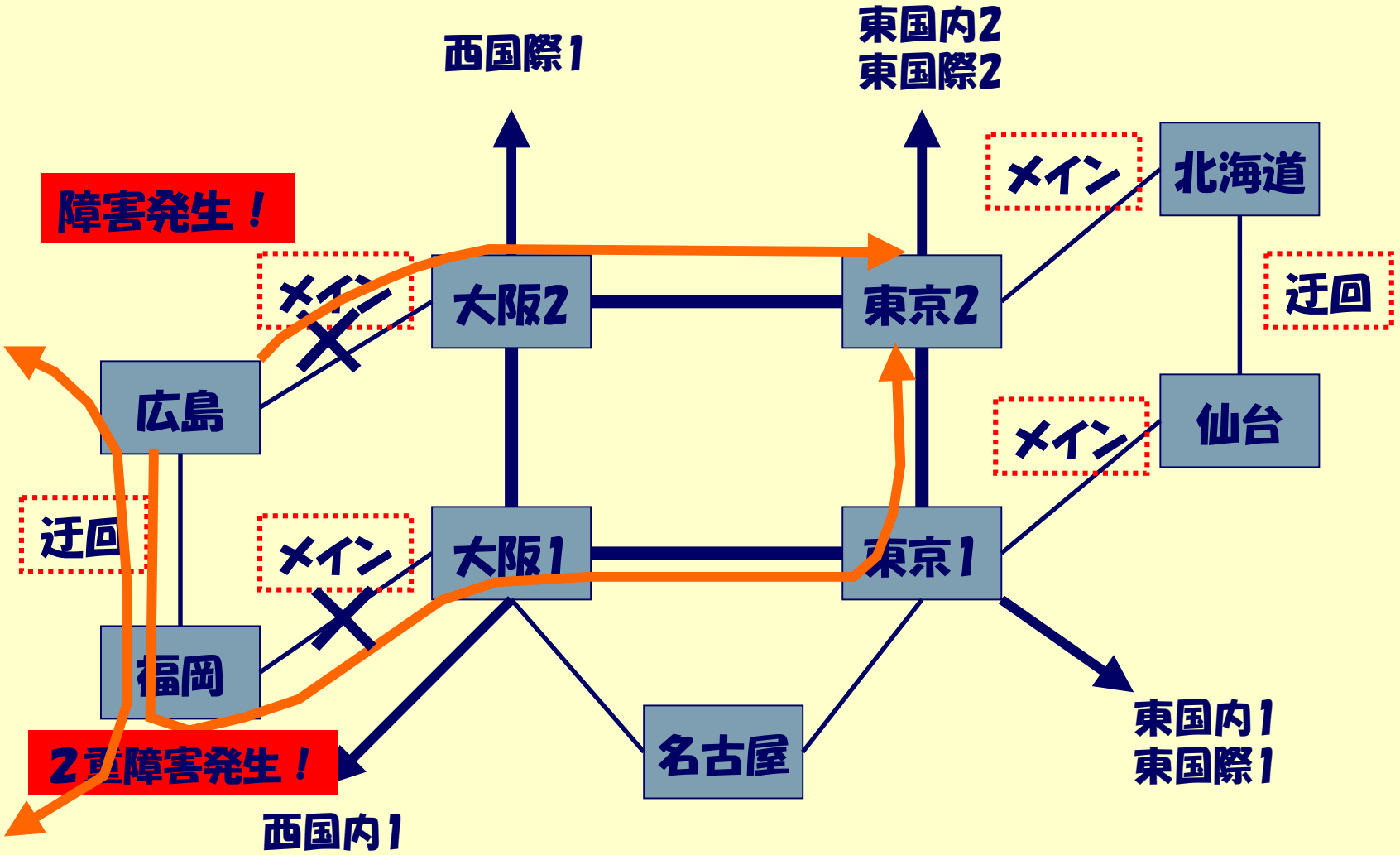
全般

- ネットワーク設計の基本事項
- トポロー情報と経路情報
- アドレス設計
- N+1設計/N+M+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
- 必要な帯域をどうやって確保するのか
  - 各POPのトラフィック
    - 地方のPOPからのトラフィックは，一番近い東京・大阪のメインPOPにもってくる．障害時は，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になるの？
      - トラフィックの多い地域をPOPとして立ち上げていく
  - 国内ISPとのトラフィック交換
    - 大きなISPとはPrivatePeerを基本．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
  - 海外トランジット
    - 均等に2つ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
  - 2重故障もある程度考慮にいれて設計する
    - 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

ネットワーク設計



OSPFやBGPの設計は後述にて

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
  - ビルレベルでの分散
  - 光ファイバーの異経路分散
  - 同一サービスの搭載架の分散，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．ナノミリsecオーダ？
- 将来性・拡張性
  - 新サービス，新たなPOPに対応可能なネットワーク

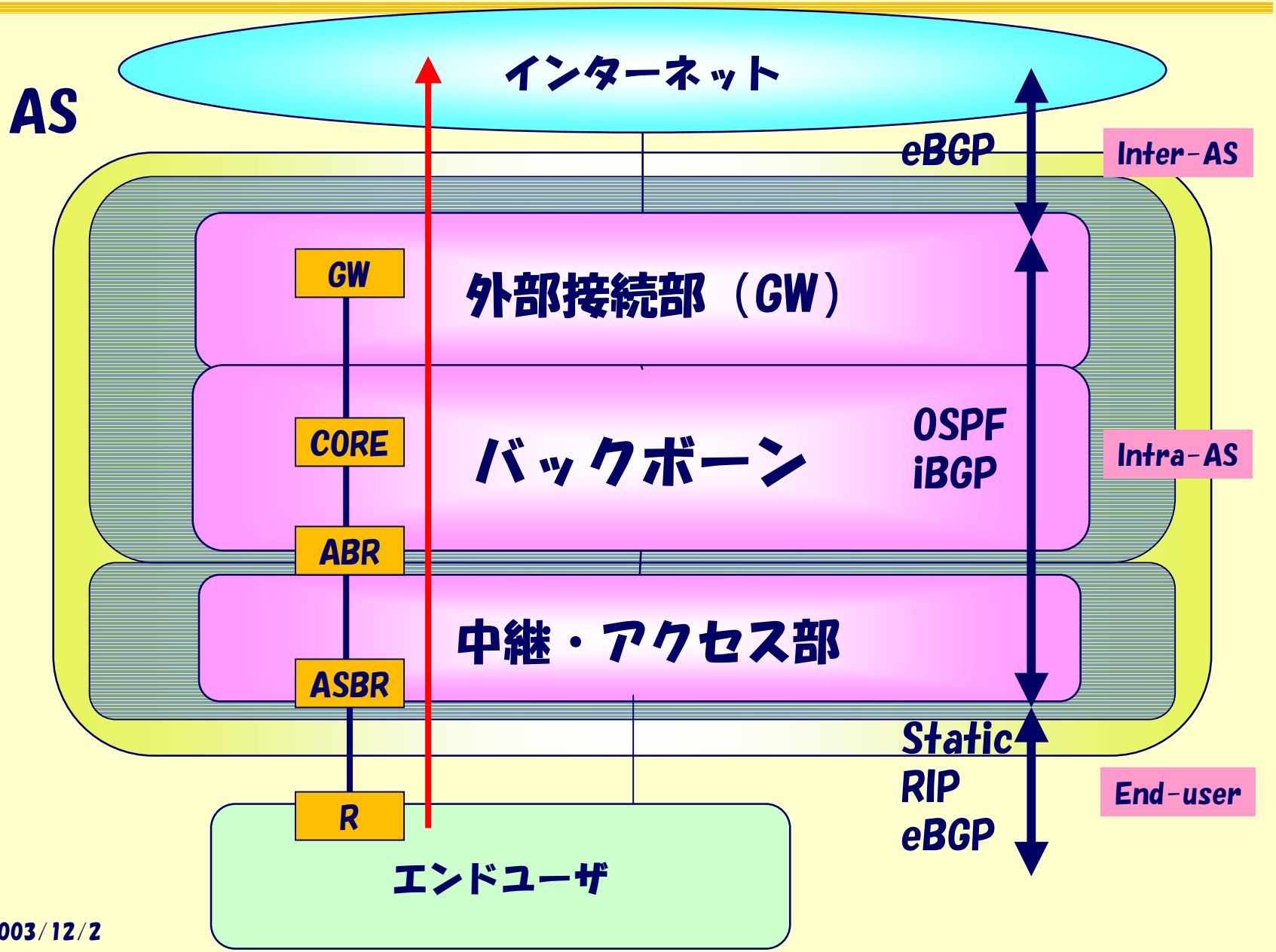
ネットワークの規模・階層的構造

- 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接続を複数本
    - 主要な地域は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
    - IGPはOSPFメイン，EGPはBGP
    - 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]
    - 外部接続部(GW)
    - バックボーン
    - 中継・アクセス部

    } intra-AS
  - 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）
  - 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
    - OSPFでは隣接とLSAを交換し，それに基づいてトポロジカルデータベースを作成する

- 経路情報
  - 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により交換
  - 以下の場合にはOSPFが有効
    - ユーザ経路を簡単にロードバランスさせたい場合
    - 実際に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

- IPアドレスの設計は
  - 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
  - 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，豊富に最初からブロックを確保できないのも事実．現実はけっこう厳しいかも（JPNICおかわり問題？）
    - ちぎって割り当てをせざるをえない
  - できる範囲内でうまく → 最近はそれほど経路が細かくなっても，ルータ自体の負荷はあまりきにしなくてもよいだろう
    - ネットワークの規模が大きくなれば，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 → 規模相応にうまく割り当てが可能となるだろう
    - 逆に規模が小さければ，それほど経路も爆発的に増えることもないので，気にしないでも大丈夫

アドレス設計

- 例えば以下ように分類し，それぞれある程度まとめてアドレスブロックを確保しておく

（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）外部アドレス
  - GWなどで外部と接続する部分のアドレス（実際には（2）に含める）

- セキュリティーの観点
  - Telnet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）

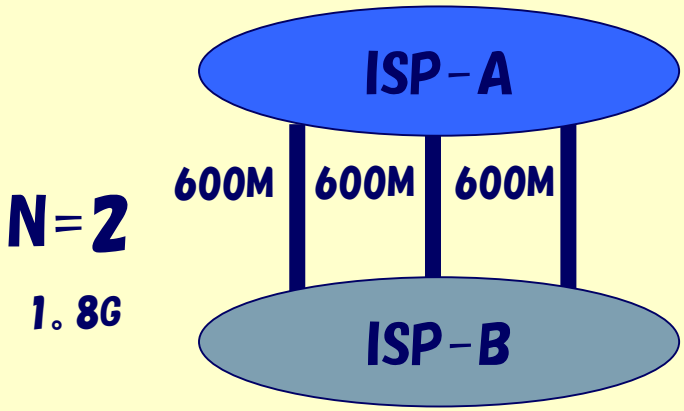
アドレス設計

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 広告	許可
(2)ユーザアドレス	ダイヤルアッププール DSL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)外部アドレス	プライベートピア・IX接続 上流ISP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 広告	拒否

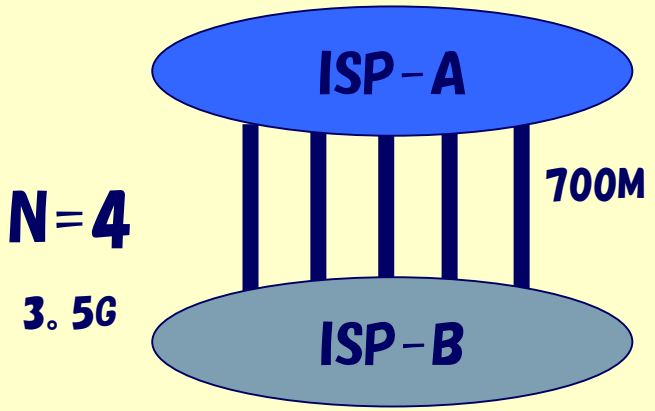
ルーティングに必要無いが，外部からの疎通確認などで実際には広報する，またちゃんとアドレスブロックがまとまっていない場合には，経路広報が細切れになってしまうので，実際にはそこまで細かく分けずに広告するのが一般的．範囲の明確化自体は必要

N+1設計

- 実際に流れている帯域に，+1 の回線本数を用意する
  - N＝1　の場合には，1+1 ＝ 2本で冗長化
  - N＝2　の場合には，2+1 ＝ 3本で冗長化
  - ・・・



100%救済を考えると，2GEのトラフィックに対して，3GE（1.5倍）の容量を確保する必要がある



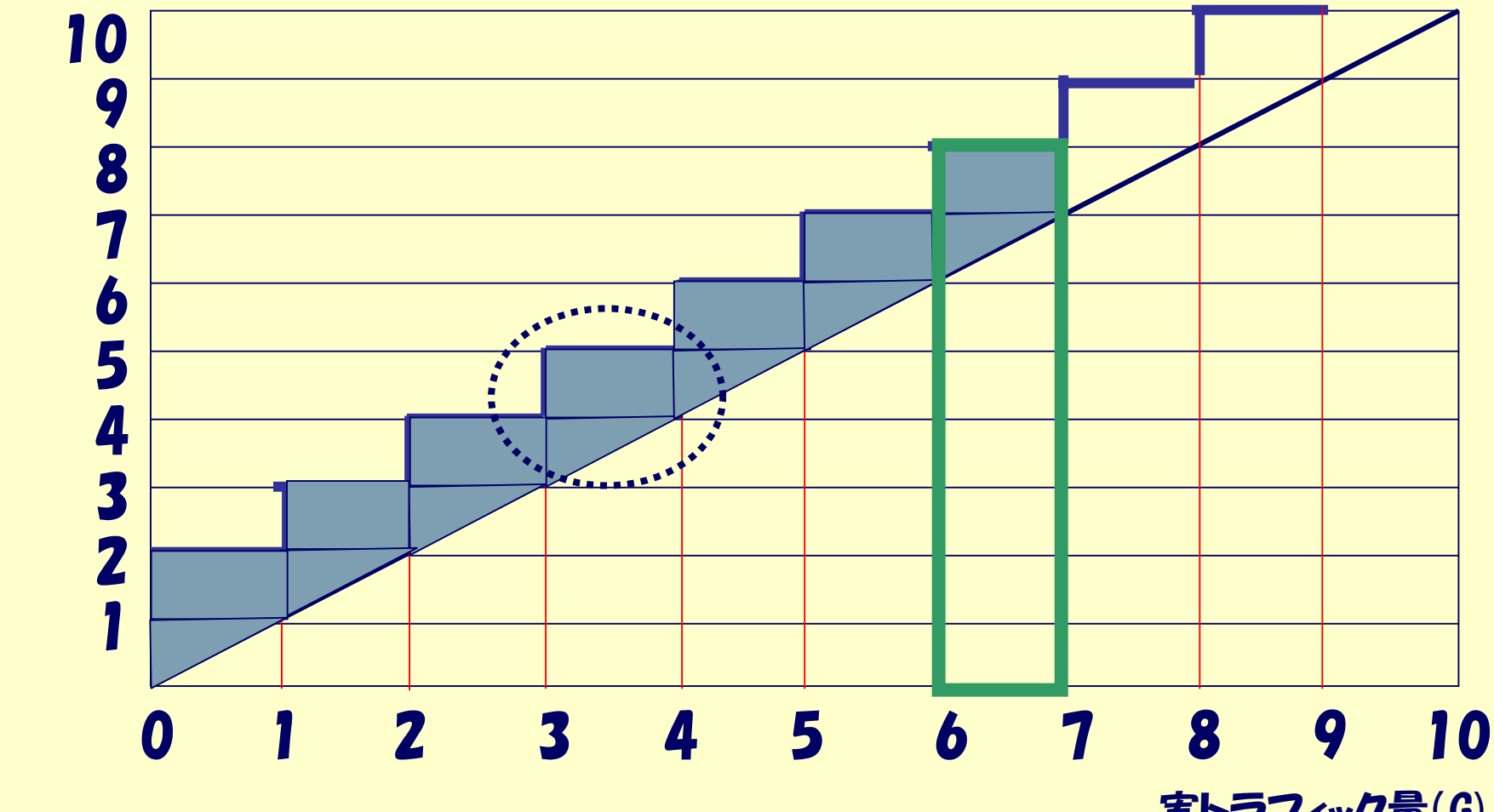
100%救済を考えると，4GEのトラフィックに対して，5GE（1.25倍）の容量を確保する必要がある

トラフィック量が増加するにつれて，回線の有効利用が見込める

N+1設計

GE増設による100%救済設計例

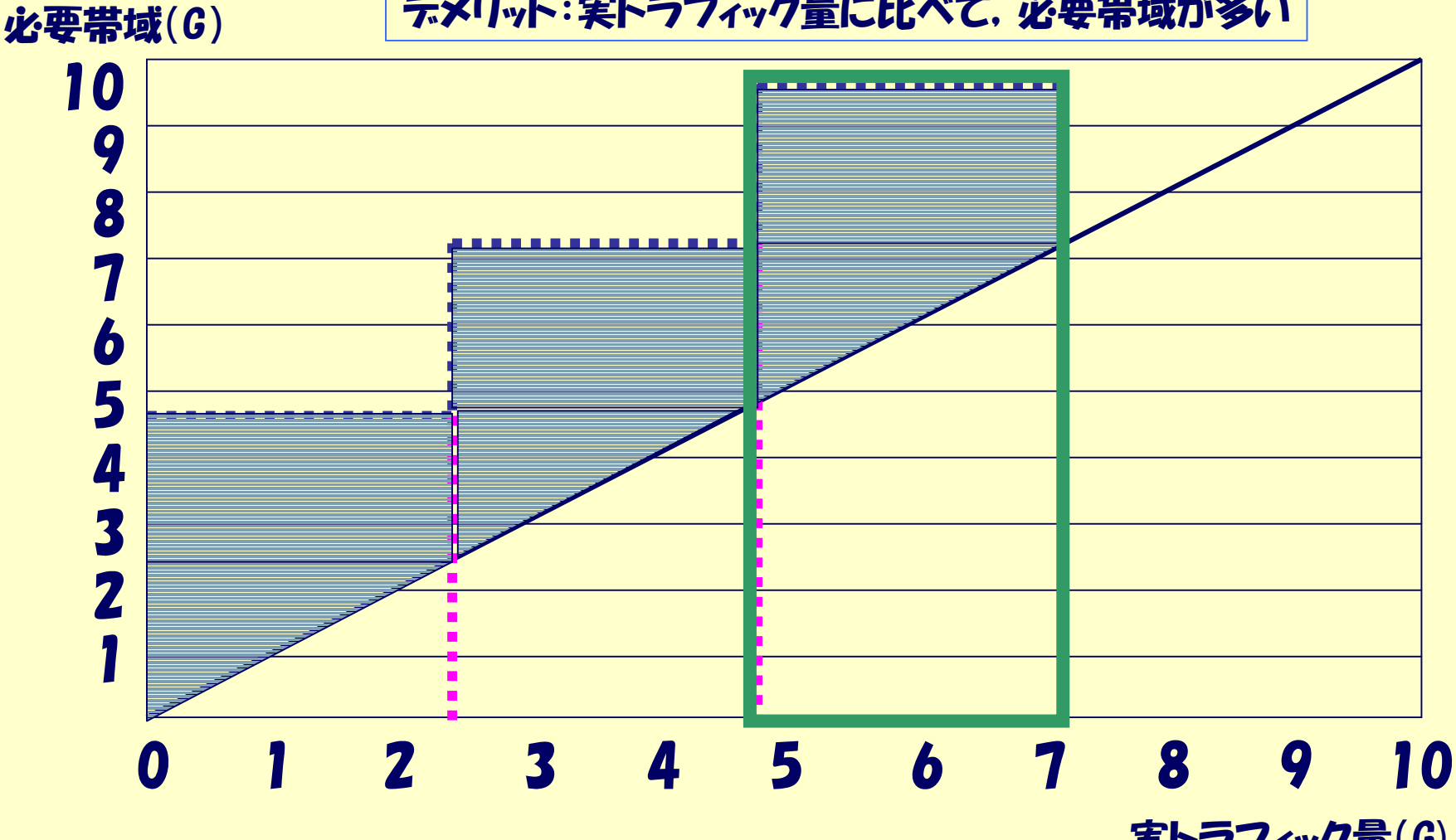
メリット：実トラフィック量が増えるほど，効率的に回線が利用できる
デメリット：増設ポイントが多いため，その都度増設設計が必要



N+1設計

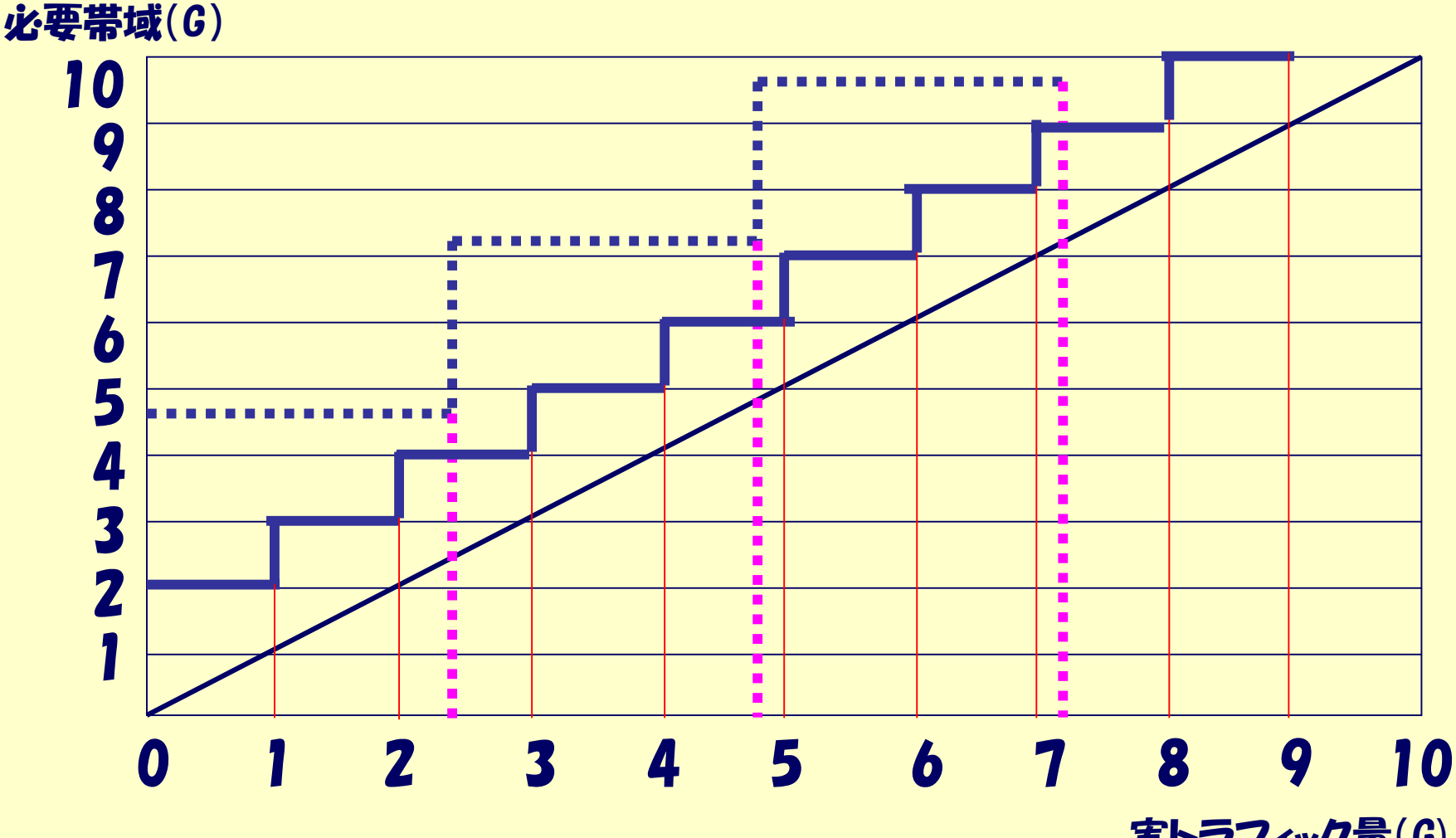
OC48増設による100%救済設計例

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



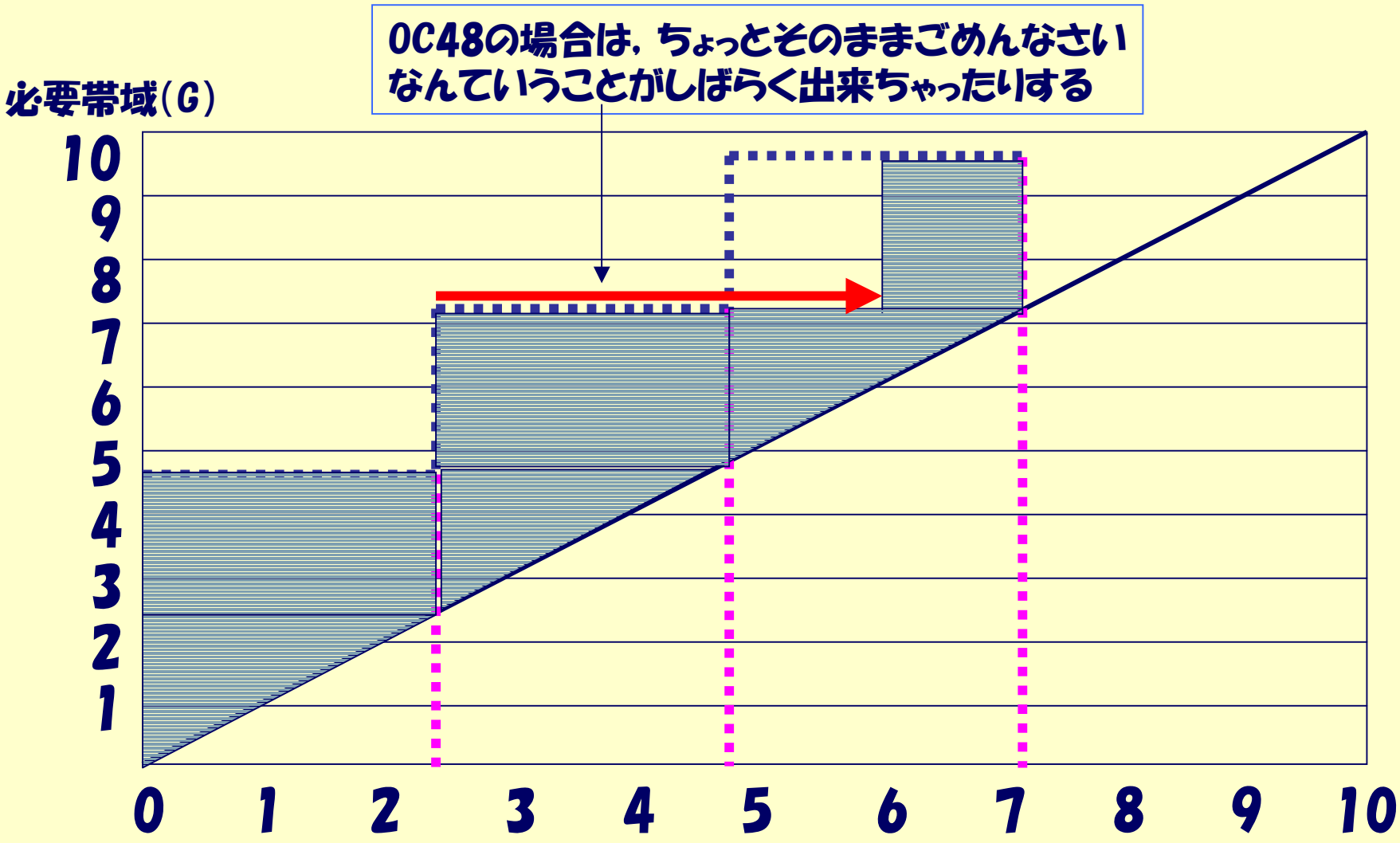
N+1設計

Gigabit Ether と OC48 を重ね合わせると・・・



N+1設計

ちょっと100%救済設計の余裕がない場合だと・・・



N+1設計

- 需要予測
  - 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
  - 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，あるいはメディアの変更を考えるべきなのか(GE → 10GE)の決断をする
    - GEを6本束ねるようになったら，Operationやルータの収容分散自体も厳しい → 10GEにすべき？ でも，用意するなら 10GE x 2 これは厳しい・・・　OC48 x 4 なら 7.2GまでOKか・・・

N+N+1設計

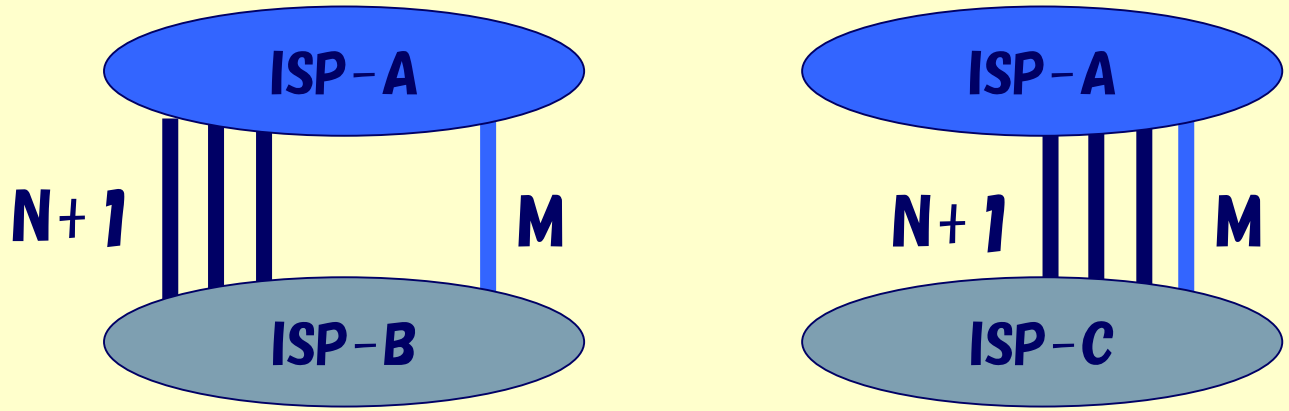
- N+1に加えて，他の接続形態(M)を含めた冗長性の確保
  - N＝1，M＝1　N+M ＝1＋1＋1＝3
  - N＝2，M＝1　N+M ＝ 2＋1＋1＝4
  - ・・・

  例えばPrivateピア(N)のバックアップにIX(M)を利用

  → バックアップ(M)を他のISPと共用させることが可能

  → N+1で100%救済が確保できない場合などに利用できる

  →とはいっても，現実的にはIXの回線って浮かしておく余裕はないかも・・・

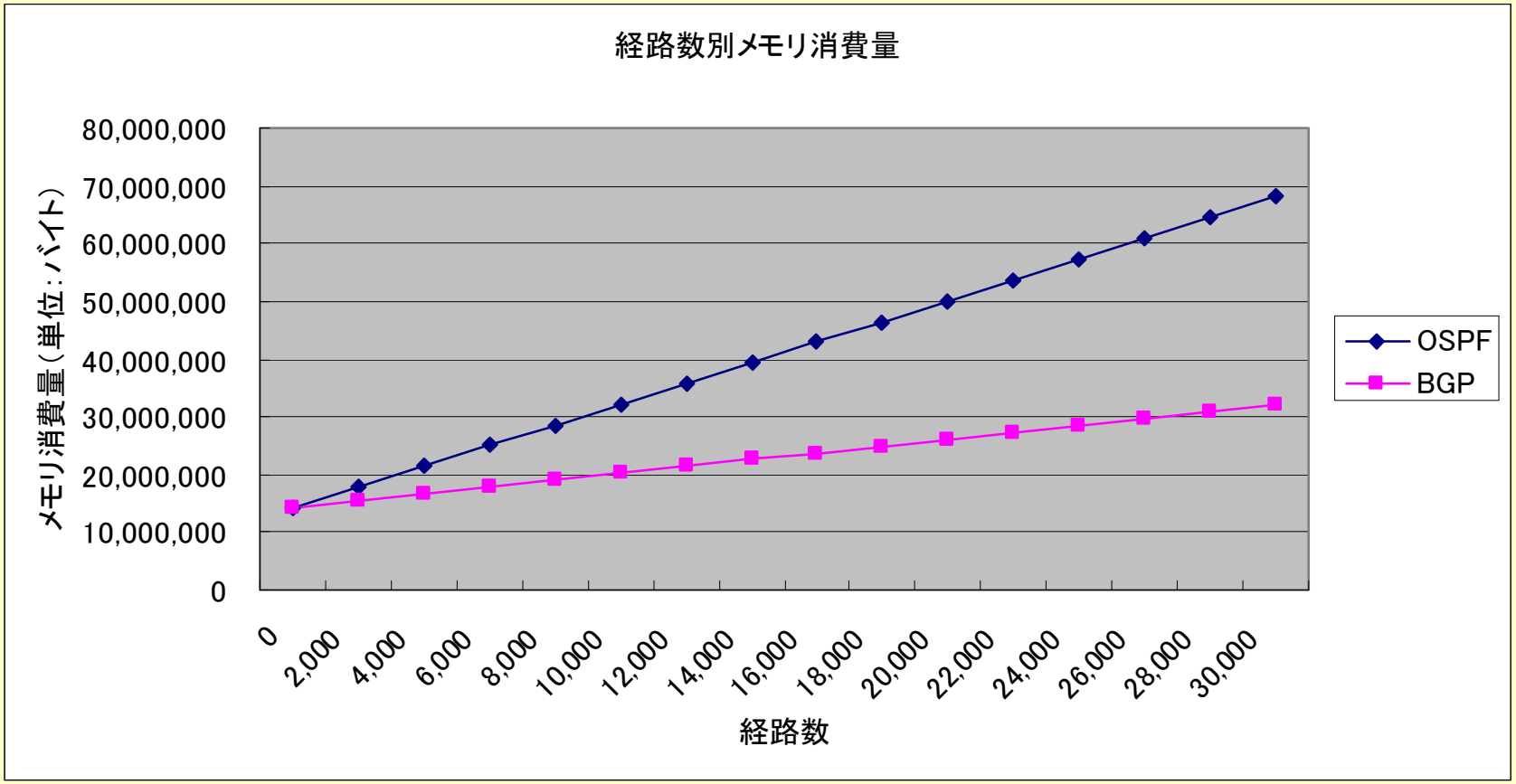


それぞれ＋1本用意する必要がないので，合計7本で済む

CPU・メモリ

- あればあるに越したことはない
  - 256M と 512M ではだいぶ異なる
- 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，あるいは数年先の状態まで考慮してテストする

OSPF・BGP メモリ消費量(例)



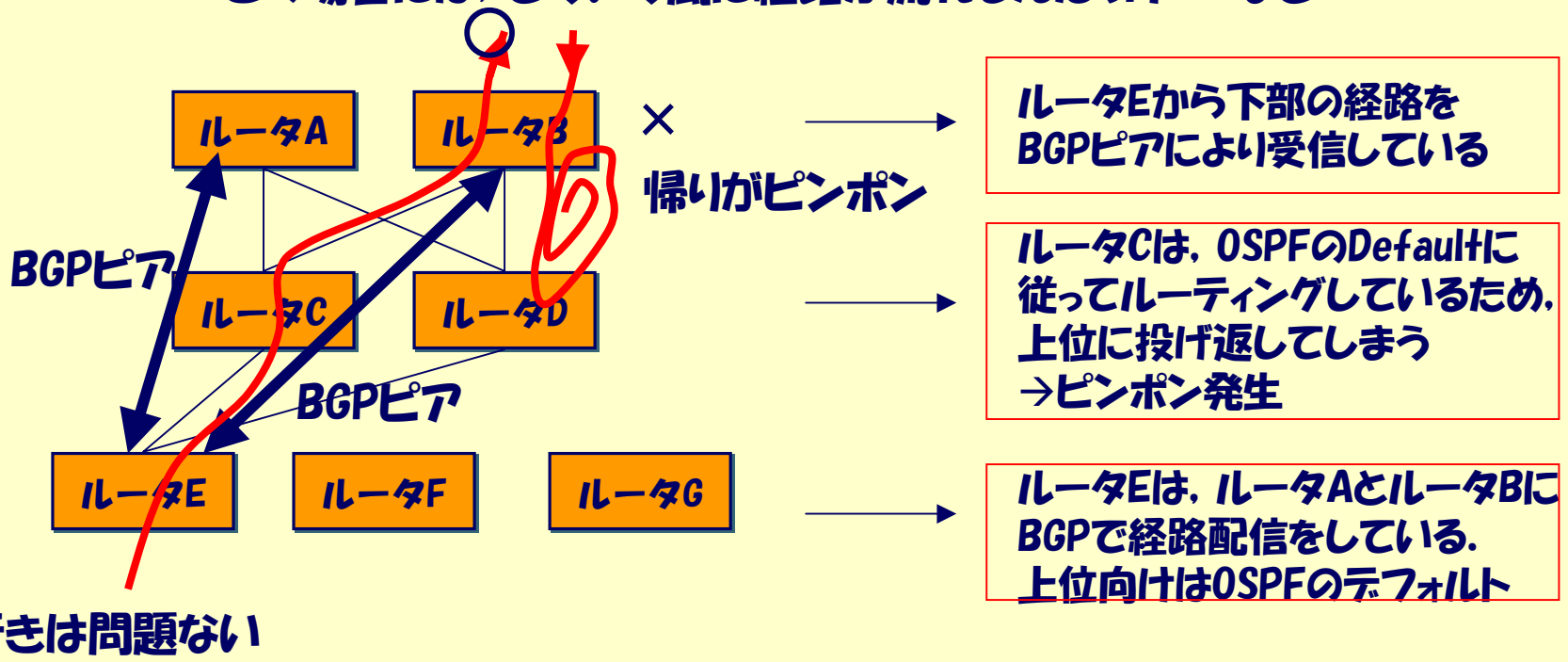
OSや機種によっても，消費量は異なるので，それぞれの組み合わせで自分にあった環境で検証する必要がある

Loopback

- 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 → これは非常にまずい
  - BGP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 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のが望ましいだろう
  - 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



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(おさらい)

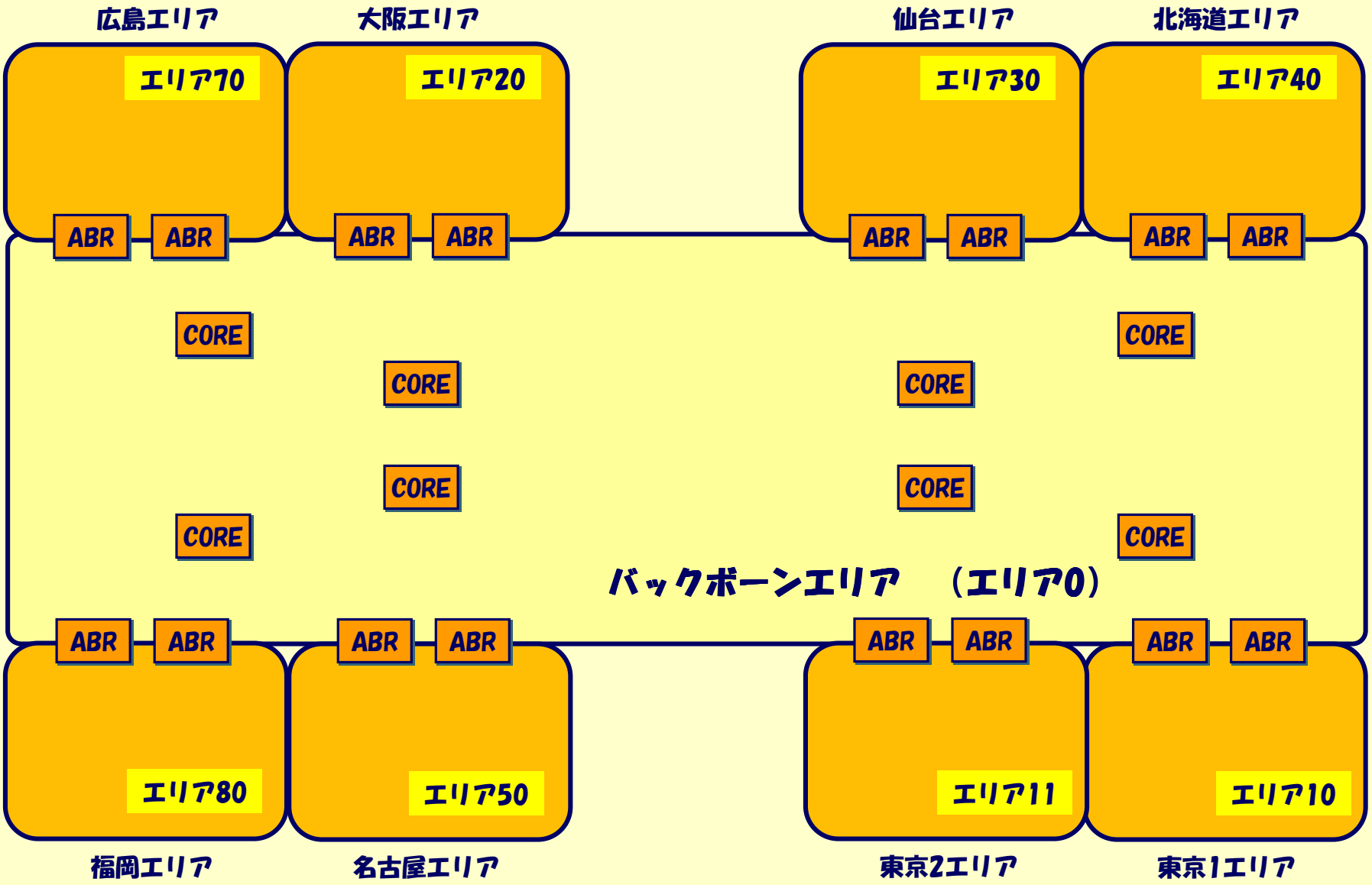
- OSPFの動き(流れ)
  - リンクステートパケットを隣接ルータ間で交換
  - それをもとに，LDSB(トポロジカルデータベース)を各ルータが作成
  - そのデータベースから，ダイクストラのSPFアルゴリズム(ダイクストラ法)を用いて，自分を頂点とした最短パスツリーを作成
  - そのツリーをもとに，ルーティングテーブルを作成

- 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もっているので，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は非常に高速化されている

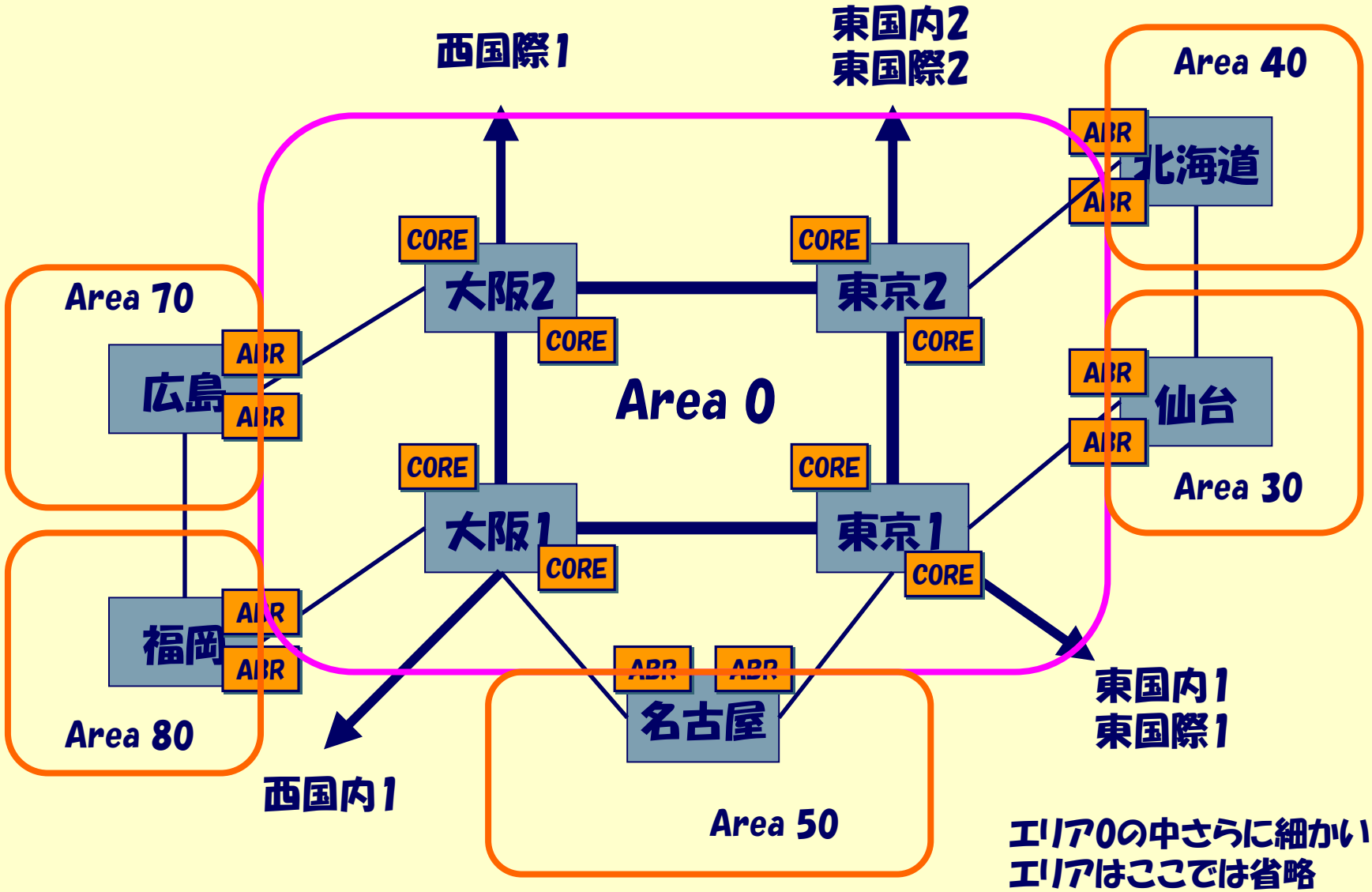
エリア設計

- 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にすればよいのか？
    - 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（京葉線や中央本線など）は，エリア0にぶらさがる各エリアとする
    - エリア0以外のエリアは，全てエリア0を介して接続するこになる
    - エリア0に各エリアがぶら下がるようなスター型の構成になる
    - ネットワークのコアとなる部分がエリア0となる
- 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
- 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・・・

エリア設計



エリア設計



エリア0の中さらに細かい
エリアはここでは省略

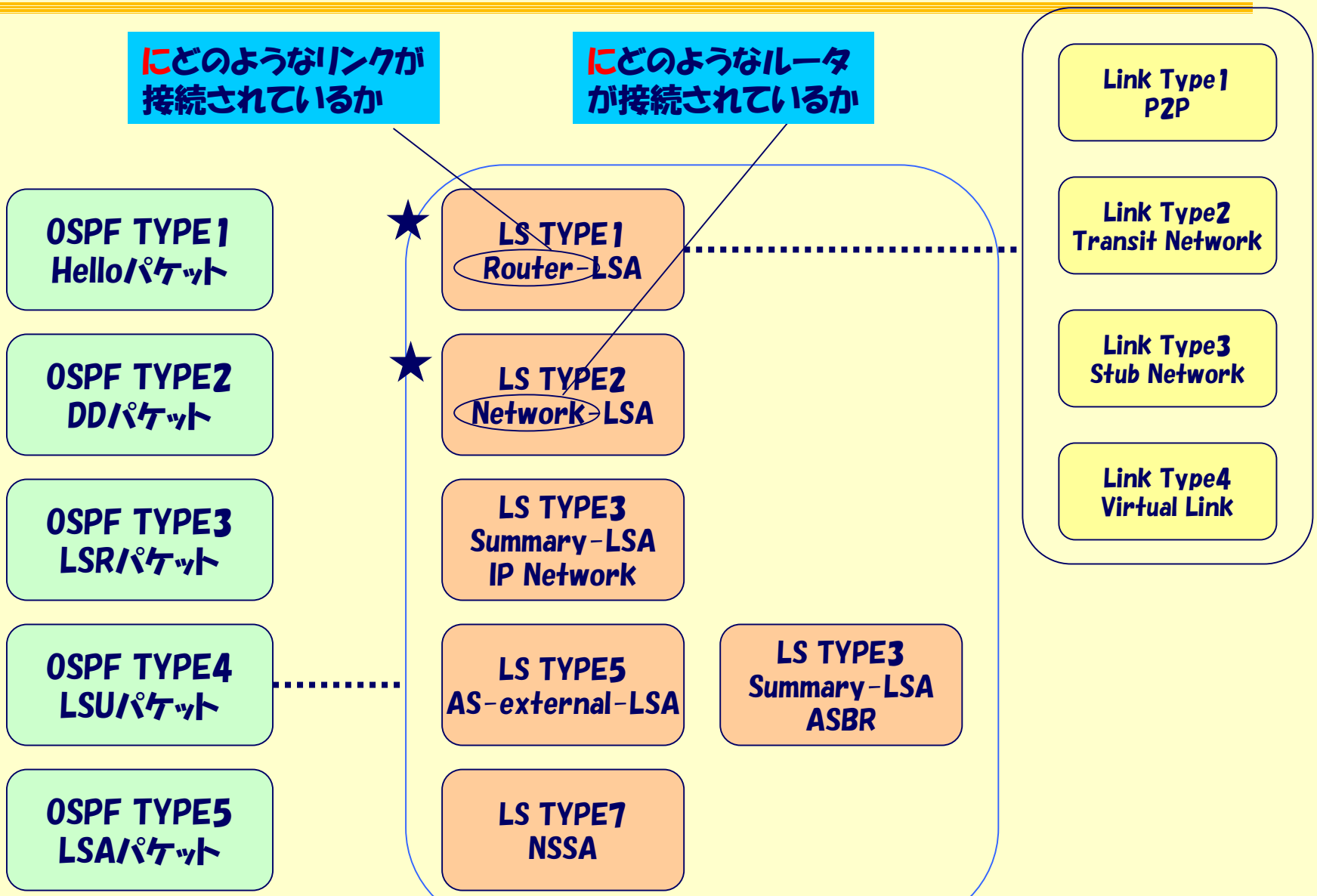
1つのエリアに置けるルータの台数

- 一概には言えません（ほとんど決まり文句・・・）
  - 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，これもあくまで一般論で，それぞれ事情は違う
  - OSPFの収束時間が以前に比べて長いなぁ
    - そろそろエリアを分割，あるいはエリア0の台数を減らすか・・・
  - ルータの性能は侮れない
    - 処理能力の高いルータと，そうではない非力なルータとでは，随分と差がある
  - 参考書や文献（でもあくまで指標にすぎない，実は結構古い）
    - 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
  - エリア0は肥大するので注意

リンクの数

- 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，LSDBが増大してしまう可能性がある
    - 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していくだけ

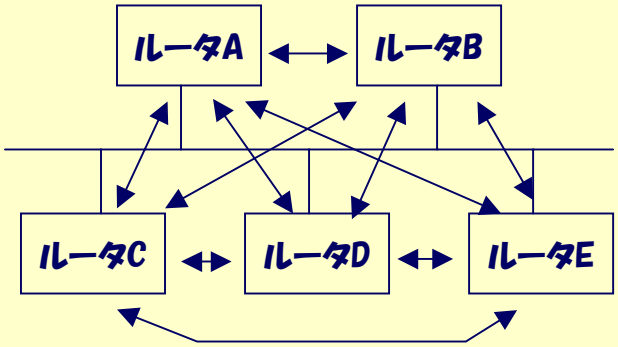
OSPFパケットの種類



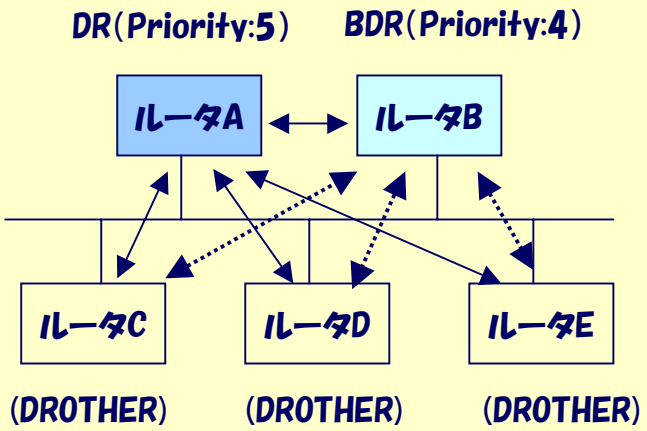
DR/BDR の設計

- 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
- 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

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

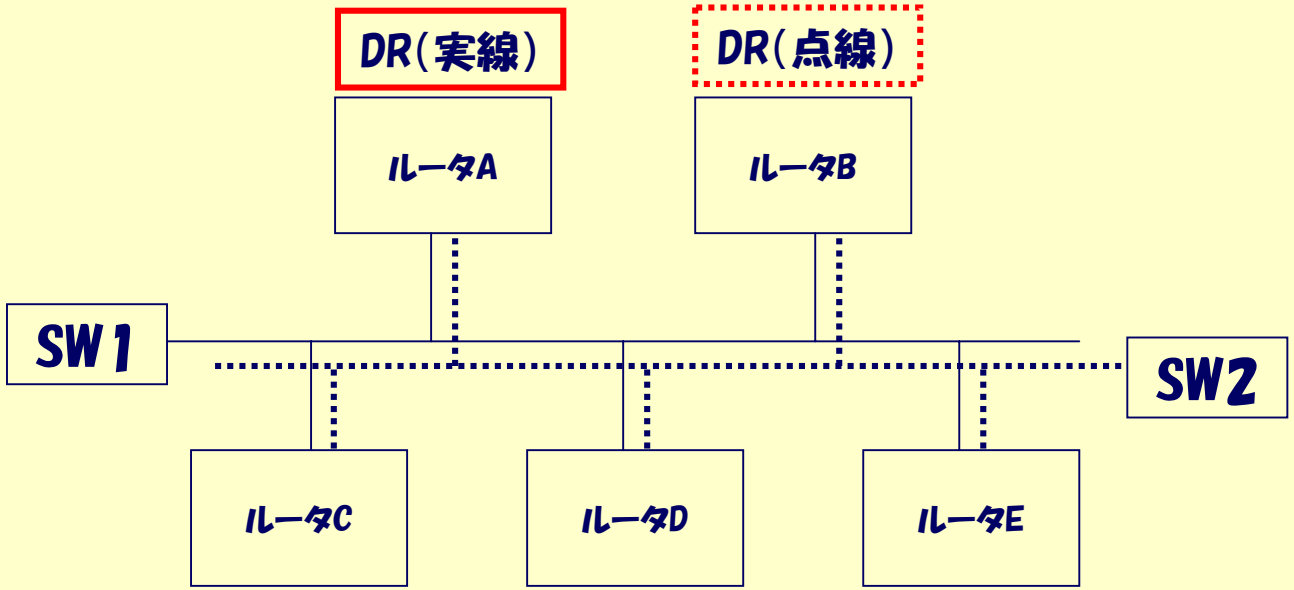
Priorityが低くても，最初に立ち上がったものがDRになってしまうので注意

DR/BDR の設計

SW1とSW2で，2重化の冗長構成をとっている場合

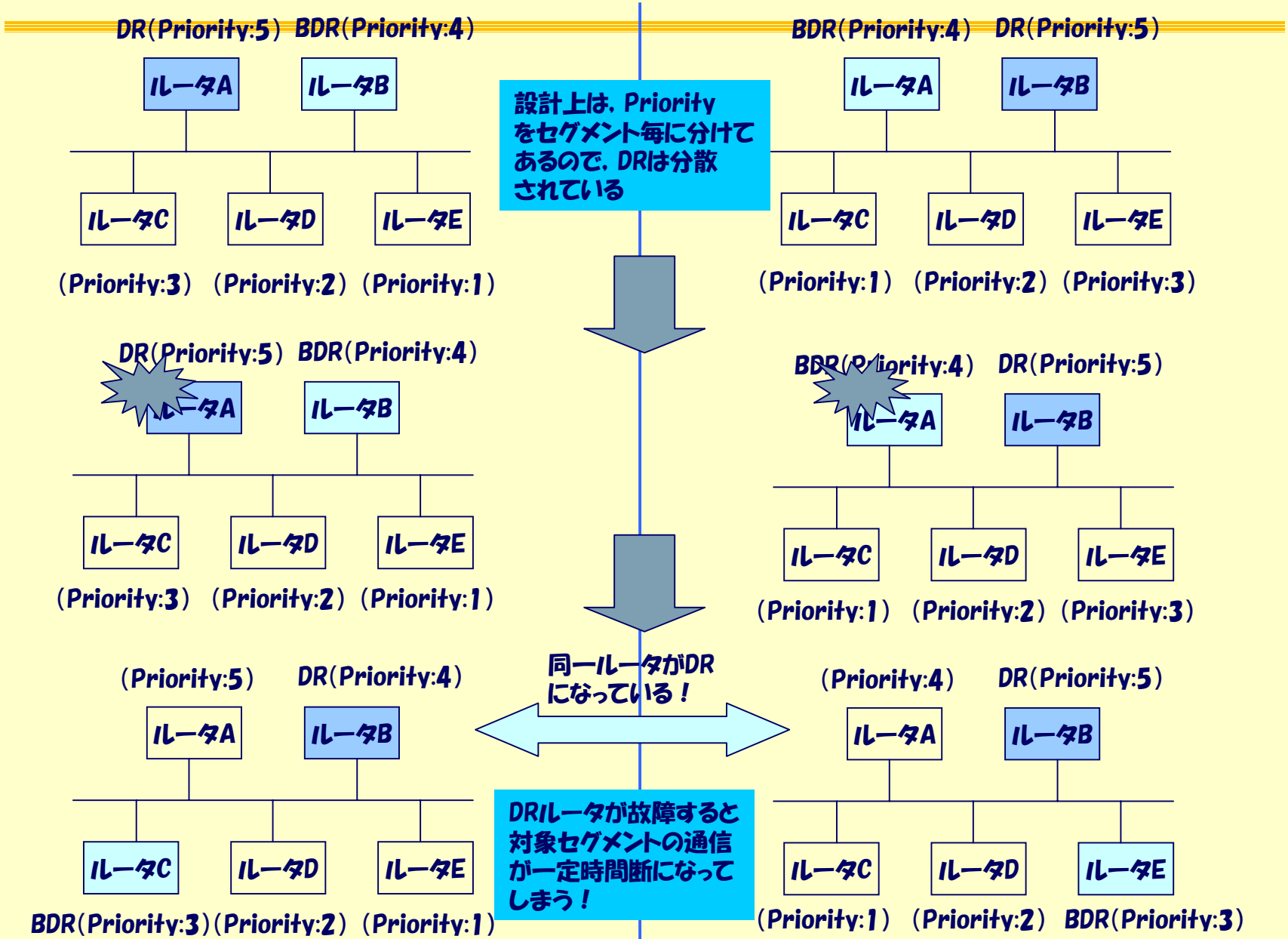
→ DRやBDRをそれぞれのセグメントで分けて付与したい

- → 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

矢印は削りました



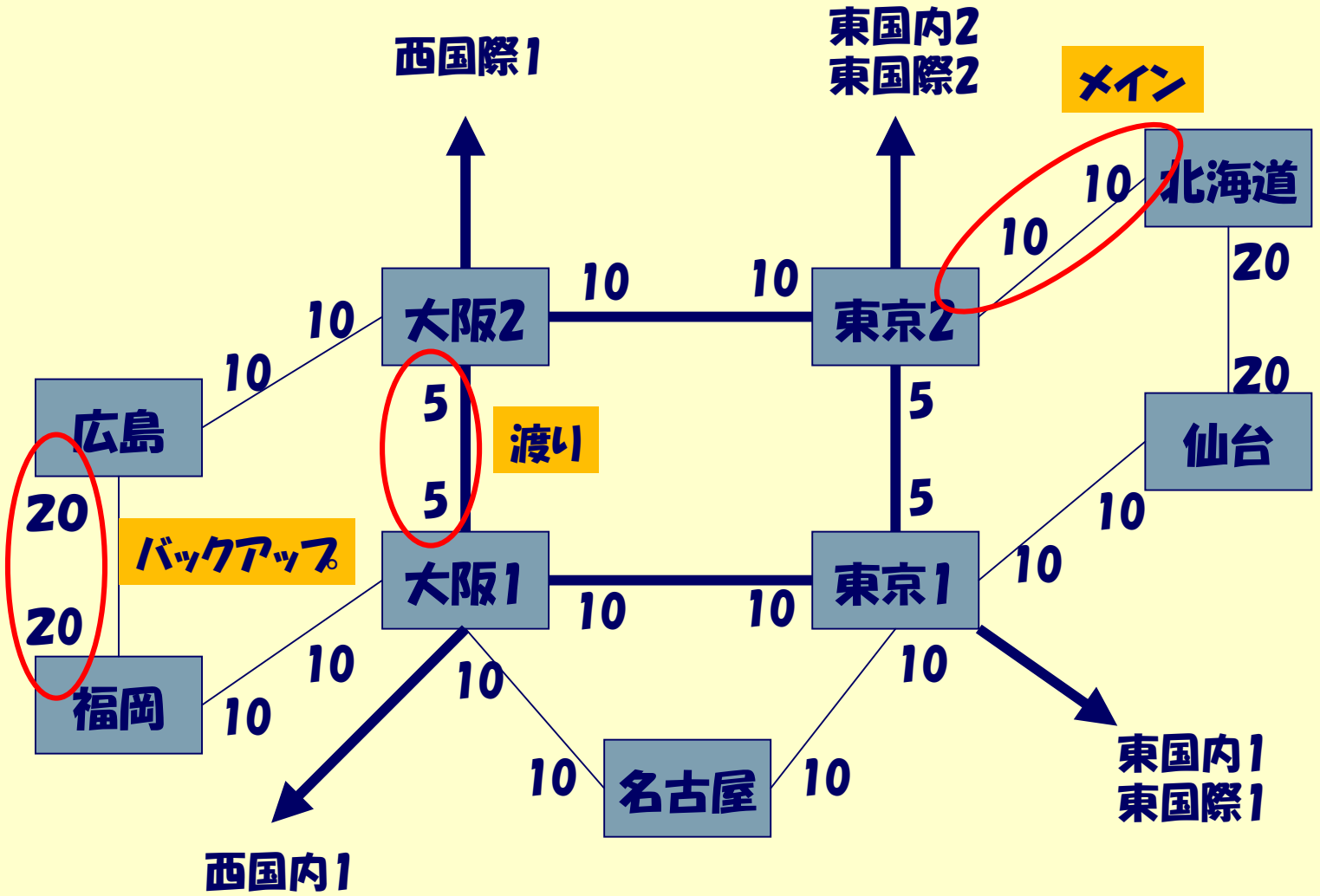
コスト設計

- ネットワークの設計ポリシーが前提（物理トポロジーを含めた）
  - どのリンクを普段メインで使うのか
  - イコールコストマルチパスにするのか，1/0 にするのか
  - あるリンクが落ちた場合には，どこで救済させるのか
    - POPが全断することを想定して，違うPOPで救済させる？
    - あらゆるパターンを想定して考えなければならない
- 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で
  - あまりにも値かけ離れていると，ぐるっと回ってしまう
  - 値は多少余裕のある設計にしておく
    - 緊急避難で，一時的に迂回させる
    - どうしようもない場合に，微妙に調整した場合
- ネットワークのトポロジーが複雑だと，非常に難しくなるので，シンプルな構成で，シンプルなコスト設計が望ましい
- ある程度体系的なポリシーを決めておく
  - 当てはまらない場合には微調整

渡り接続回線：	5
メインの回線：	10
バックアップの回線：	20

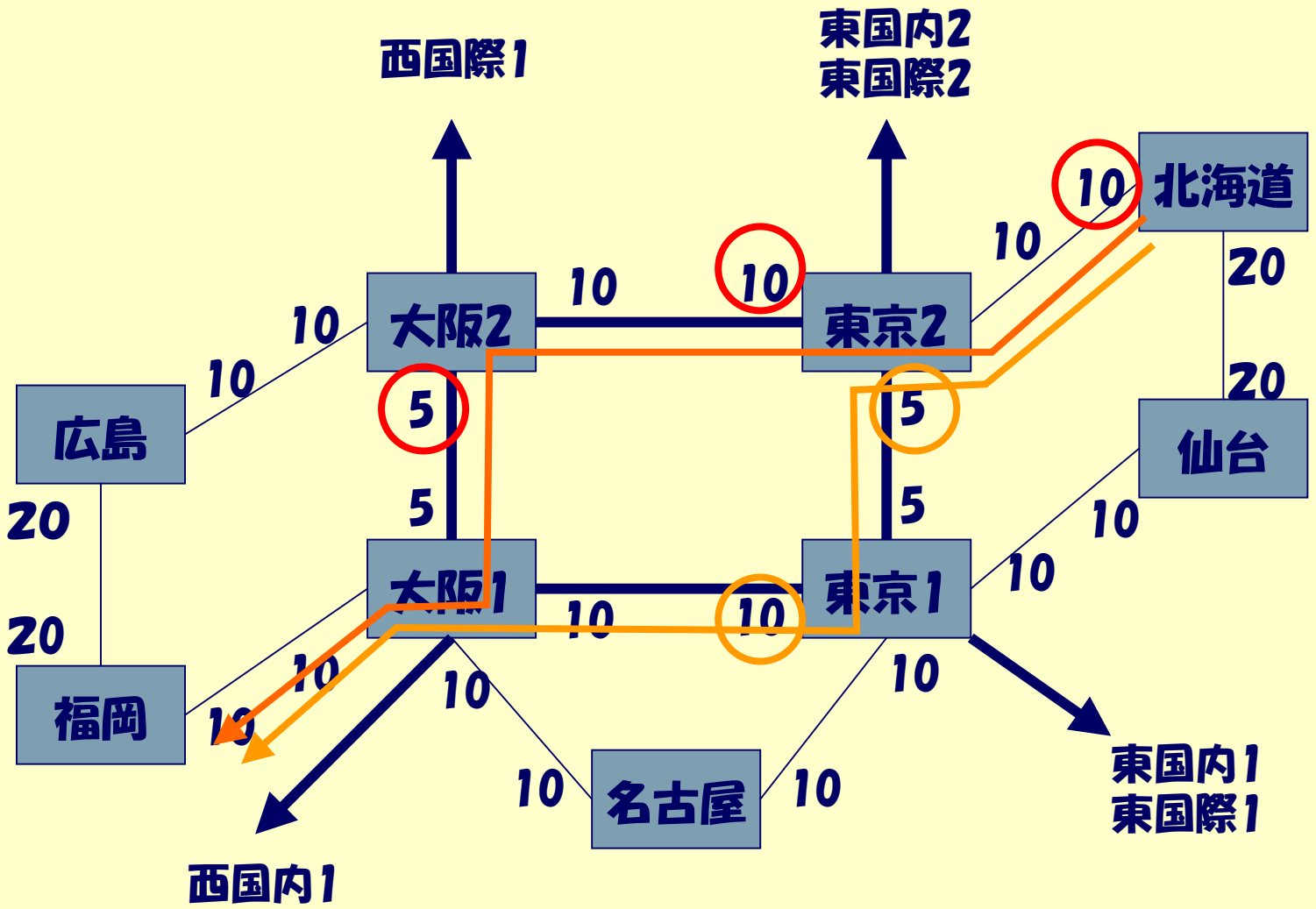
コスト設計

渡りが5，メインが10，バックアップが20



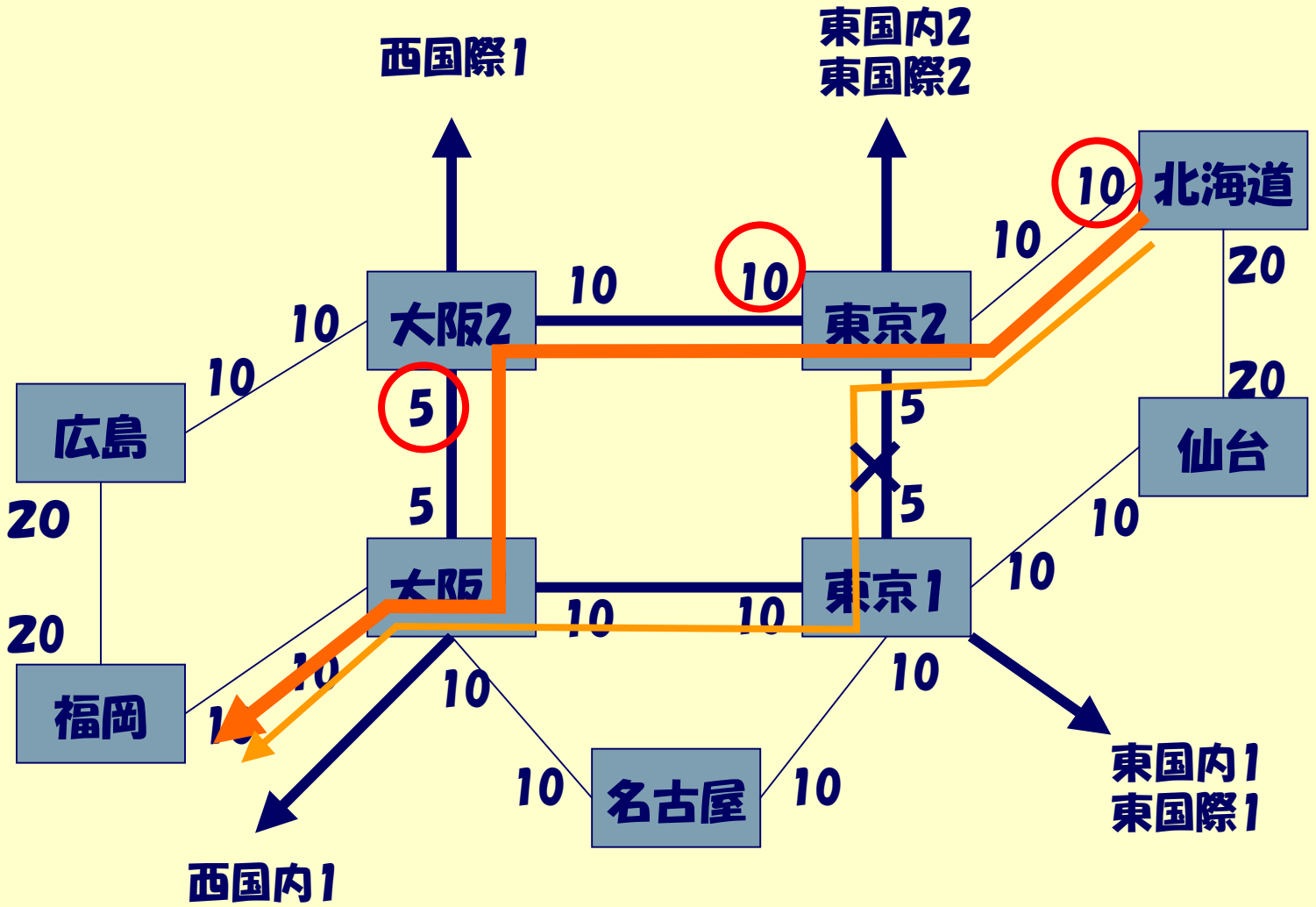
コスト設計

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

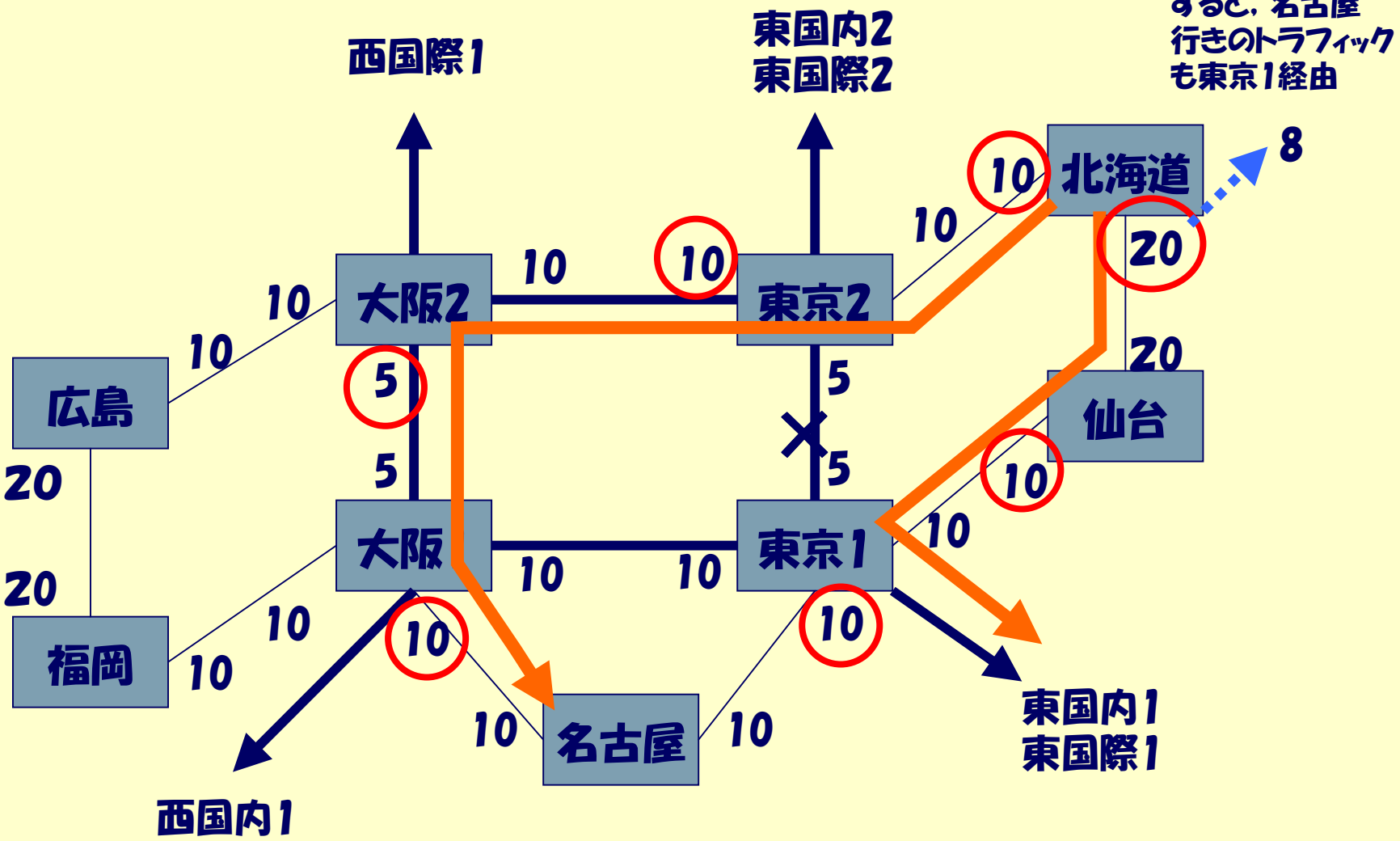
東京1と東京2のリンクがきれた場合 → 全て大阪2経由



コスト設計

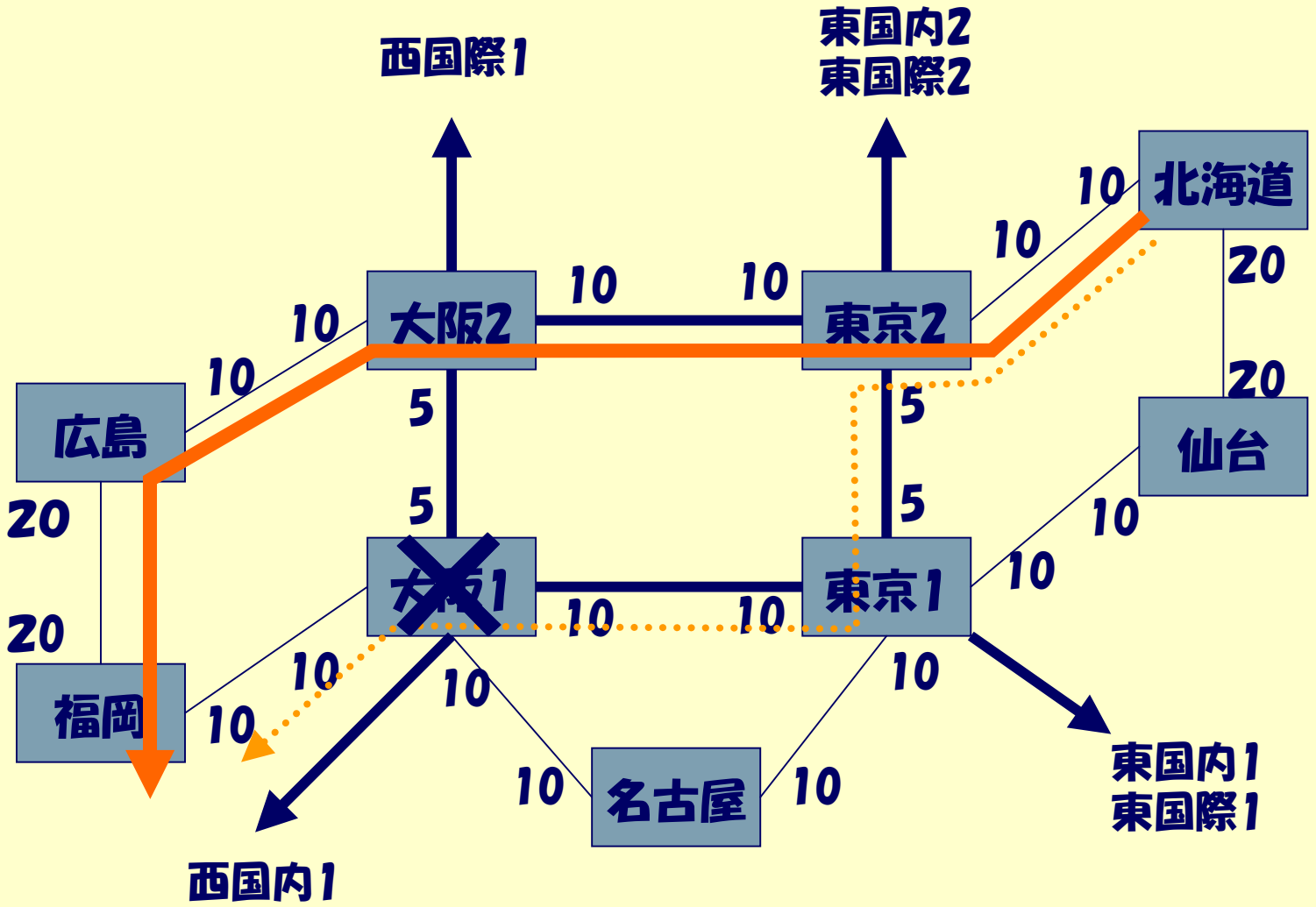
このとき，北海道と仙台のリンクが細い場合などは，
名古屋や西国内へは大阪1経由，東京1や東の国内，国際は仙台経由

これを8などにすると，名古屋行きのトラフィックも東京1経由



コスト設計

大阪1が崩壊 → 大阪2から広島経由で福岡へ



コスト設計(まとめ)

- ポリシー決め
  - 物理構成とトラフィックに基づいて，どこがきれたらどう迂回させるのか
  - 用意できる回線や帯域に依存してしまう場合もあるが・・・
- あまり複雑な設計はしない
  - オペレーションしやすい設計は大切
  - ある場所が故障した際に，あまりに複雑な救済経路にしない
  - 行きと帰りは基本は一緒にする(運用性)
    - わざと行きと帰りの経路をわける場合もあるが
- 思わぬ事態が
  - 設計どおりに実際いかない場合がある
    - 故障時に，想定していたパスとは違うパスに流れ込んでしまった・・・
    - その都度見直し

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をす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，複雑な経路計算は出来ない
  - ただし，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

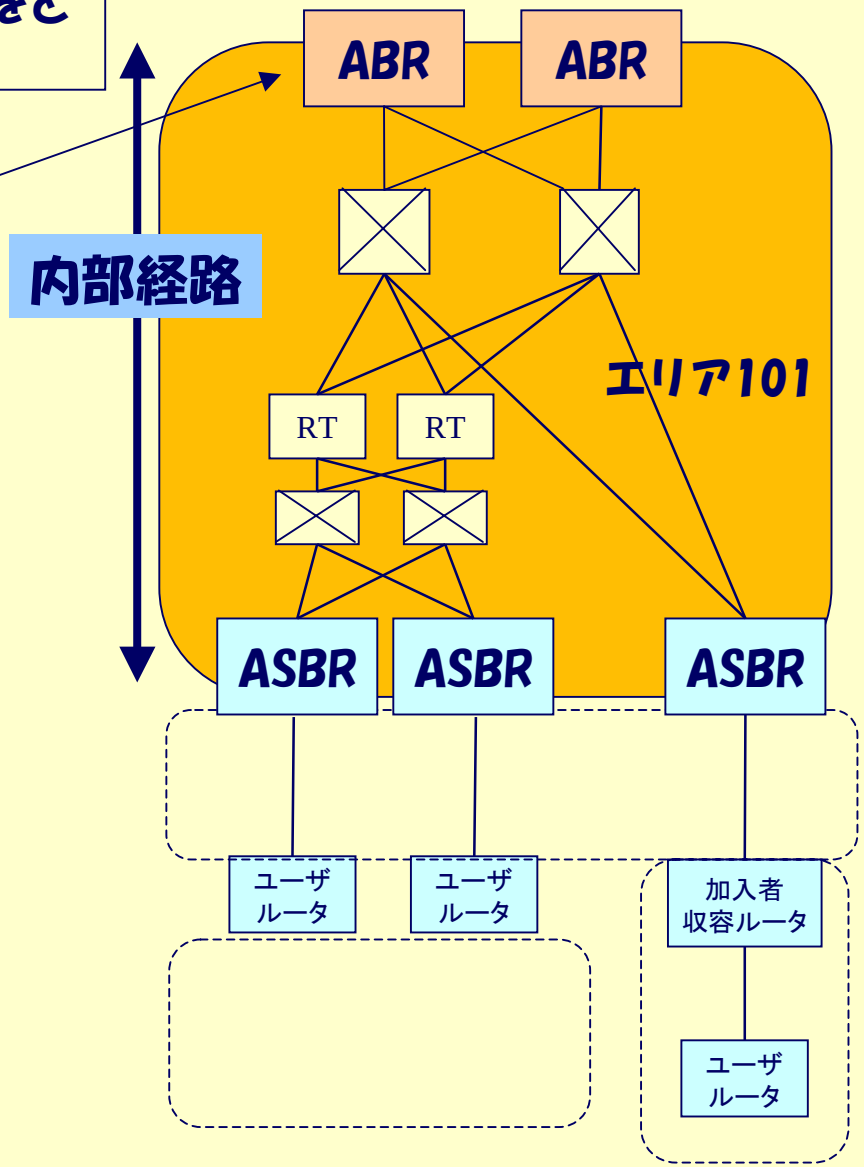
ASBRから上位は，トポロジーの冗長構成をとるためInternal経路である事が必須

■ Ciscoの場合

```
router ospf 2003
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

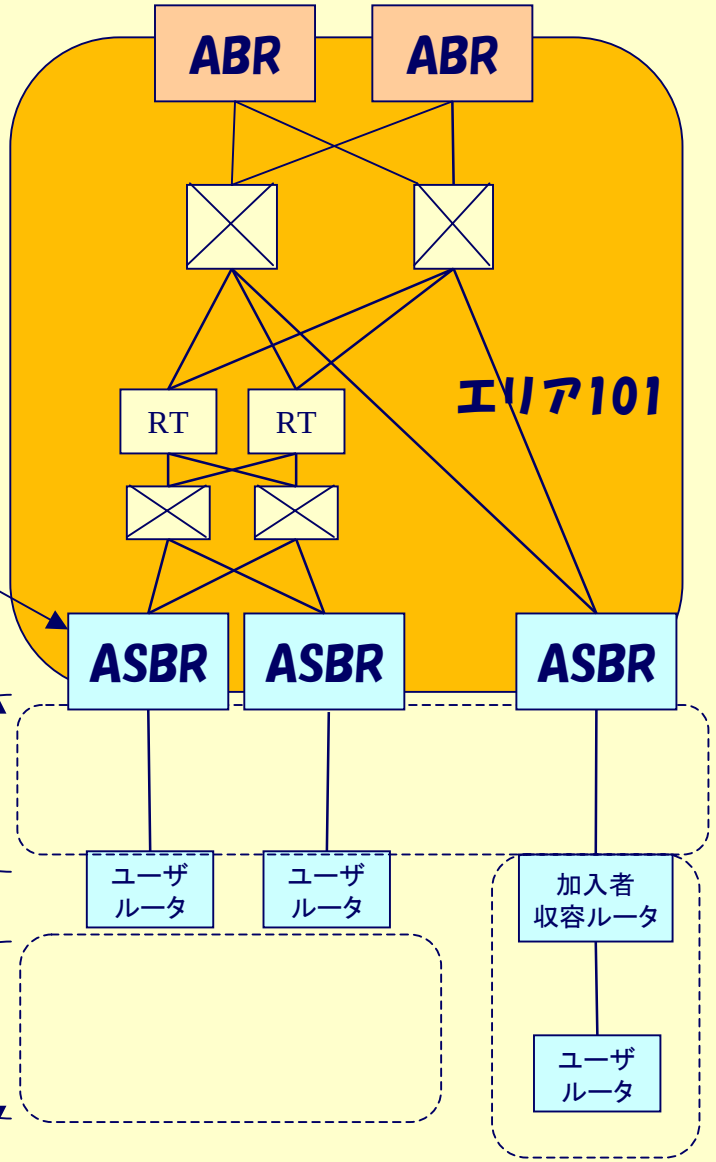
```
protocols {
 ospf {
    area 0.0.0.0 {
        interface so-0/1/0.0;
        interface so-1/1/0.0;
    }
    area 0.0.0.101 {
        interface lo0.0;
        interface so-2/1/0.0;
        interface so-2/2/0.0;
    }
  }
}
```



OSPF外部経路

■ Ciscoの場合

```
router ospf 2003
 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

外部経路

Networkコマンド + passive → Internal

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・・・

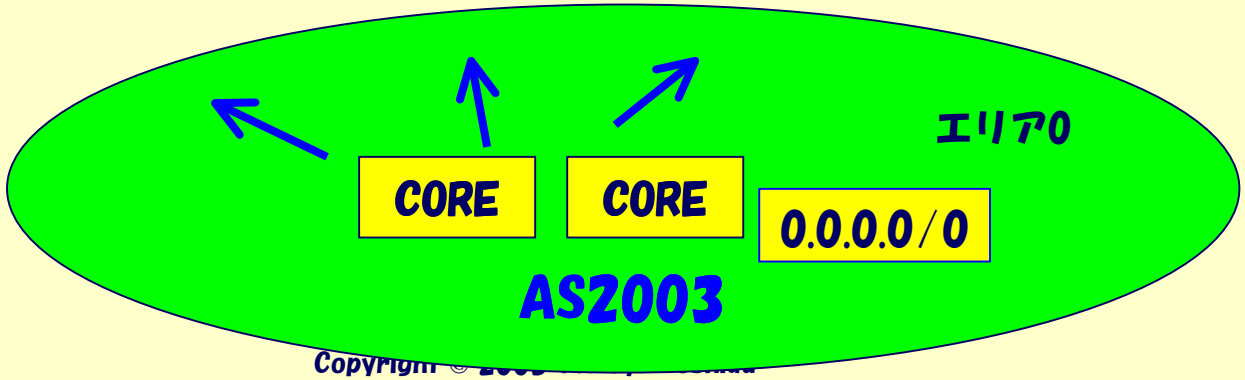
フルルートを保有していないルータが，フルルートを保有しているルータに
ルーティングできるように設定するもの

★パケット破棄能力にすぐれたCORE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 2003
 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

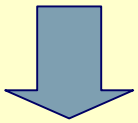
protocol が static で
0.0.0.0/0 に exact match した場合のみ
metric 5, external type-1 で広告

それ以外は，reject

Static route の生成
→ discard = null0

OSPFの安定性

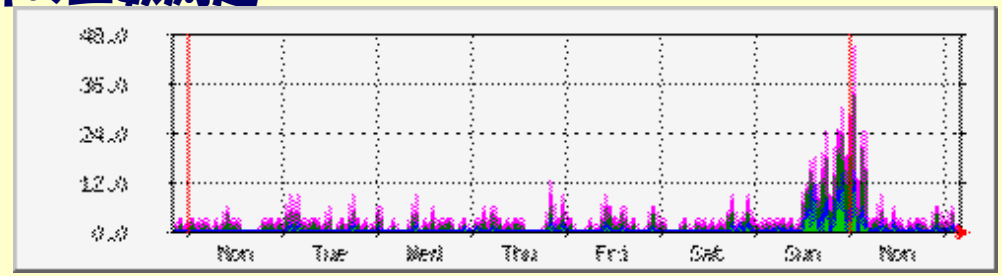
- どの程度の規模まで現状のまま耐えられるか？
  - ルータの機器，メモリ量，CPU，ネットワークのトポロジーなど，色々な要素があるので，ケース・バイ・ケースというのが正直なところ
  - 検証をするにしても，何十台もルータをかき集めて同じ環境を作ってやるのは不可能



  - 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
    - 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・・・
- 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)
  - バグっていう事もよくある
  - 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
- 普段からの確認
  - MIBによる，OSPFの再計算の回数測定
  - MRTGへのグラフ化



危ないと感じたら・・・

- 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　and 帯域を太くしてまとめて行く（序所に）
- 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

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，ちゃんとエリア0.0.0.1 と書くのがよい

- 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

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くようなら，台数を減らしたほうがよいだろう
- リンク数
  - あまりむやみに増やすような設計はさけたい
  - point-to-point とSWセグメントをバランスよく
- メモリ
  - BGPの経路数の約2倍は消費するので，普段から注意が必要
- DR/BDR
  - DRルータは，かなり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基本
  - 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を兼任してしまわないように設計する
    - Priority設計
    - 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

- 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
  - なるべくエリアごとに経路が集成できるようなアドレス設計
  - 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
  - 他の方式を検討するのも価値がある
- 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)

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設計

- AS内，AS間において，どのようなポリシーで，いかに最適に，スケーラブルにBGP経路を配信させるか
  - 外部ASから何の経路を受信するのか．どのような優先性を与えるのか
    - 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内全体に一律配るのではないとすると，じゃぁどこに対して何を配信したらいいのか？
    - 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？

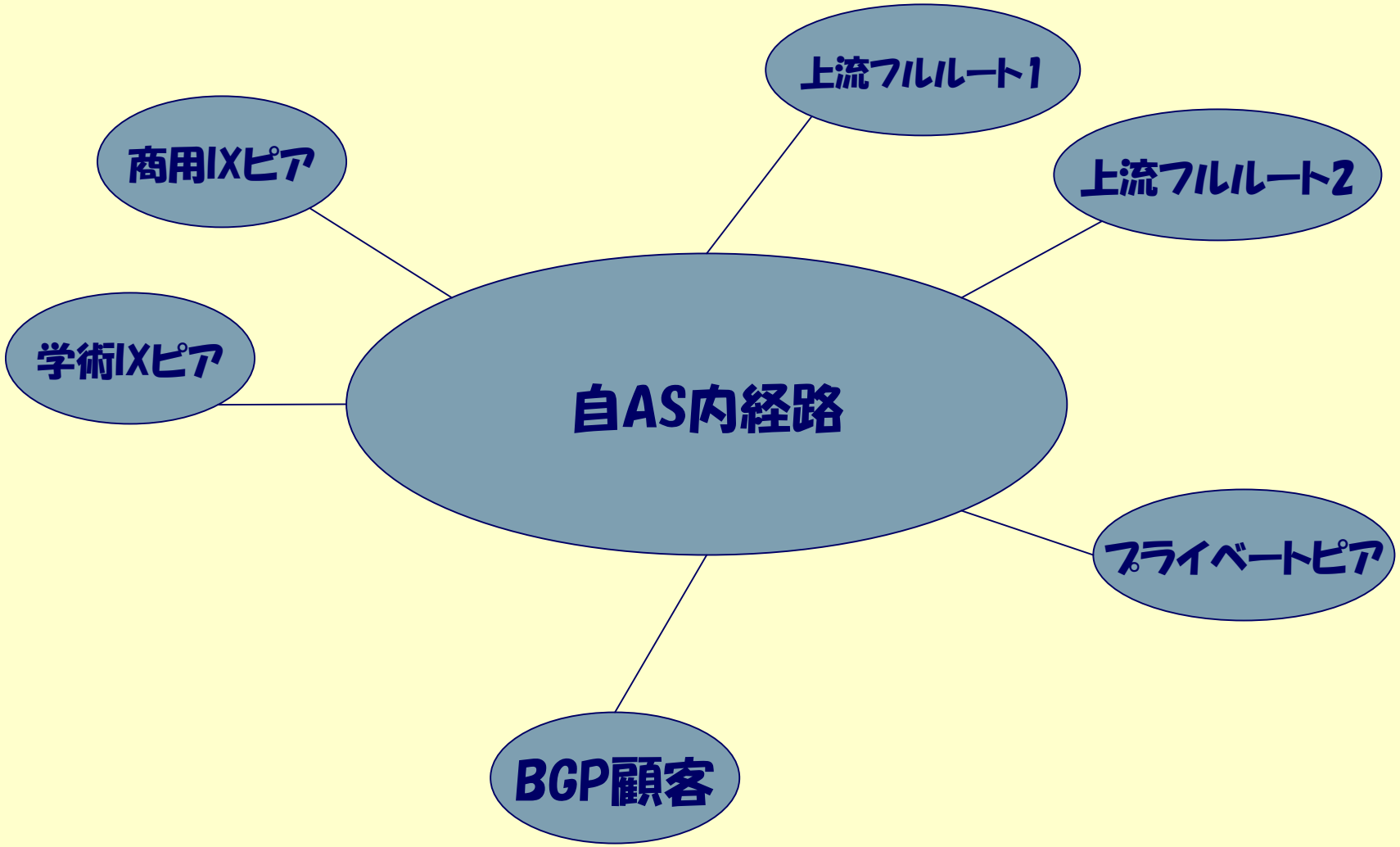
BGPポリシー設計

- 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でFilterをかける？
    - Filterするにはちょっと負荷が気になるので，受信したとしても，該当経路を優先させないように内部で制御をかける？

- 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」

値づけはバッファ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 permit 10
 match as-path 3
 set metric 100

route-map ebgp-out permit 11
 match as-path 4
 set metric 200
...
```

途中にdenyのroute-mapを挿入
したい場合に,数字を書き直さないと駄目

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

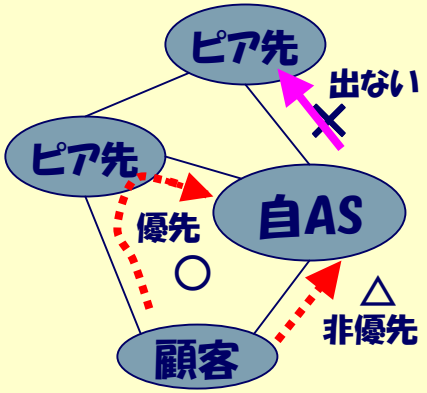
→ 数字には余裕をもって設計
→ ここでの優先順位とは，単純にLOCAL_PREFの値を元とした順位

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

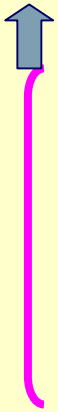
→ 顧客経路は他のISPなどにちゃんと広報する必要がある
→ もしその顧客が他のISPとマルチホーム接続をしていれば，ピア経路としても聞こえてくる場合がある
→ その際，仮にピア経由を優先してしまうと，自AS内でベストパスではなくなるため，経路がアナウンスされなくなってしまう！



BGPポリシー設計(受信)

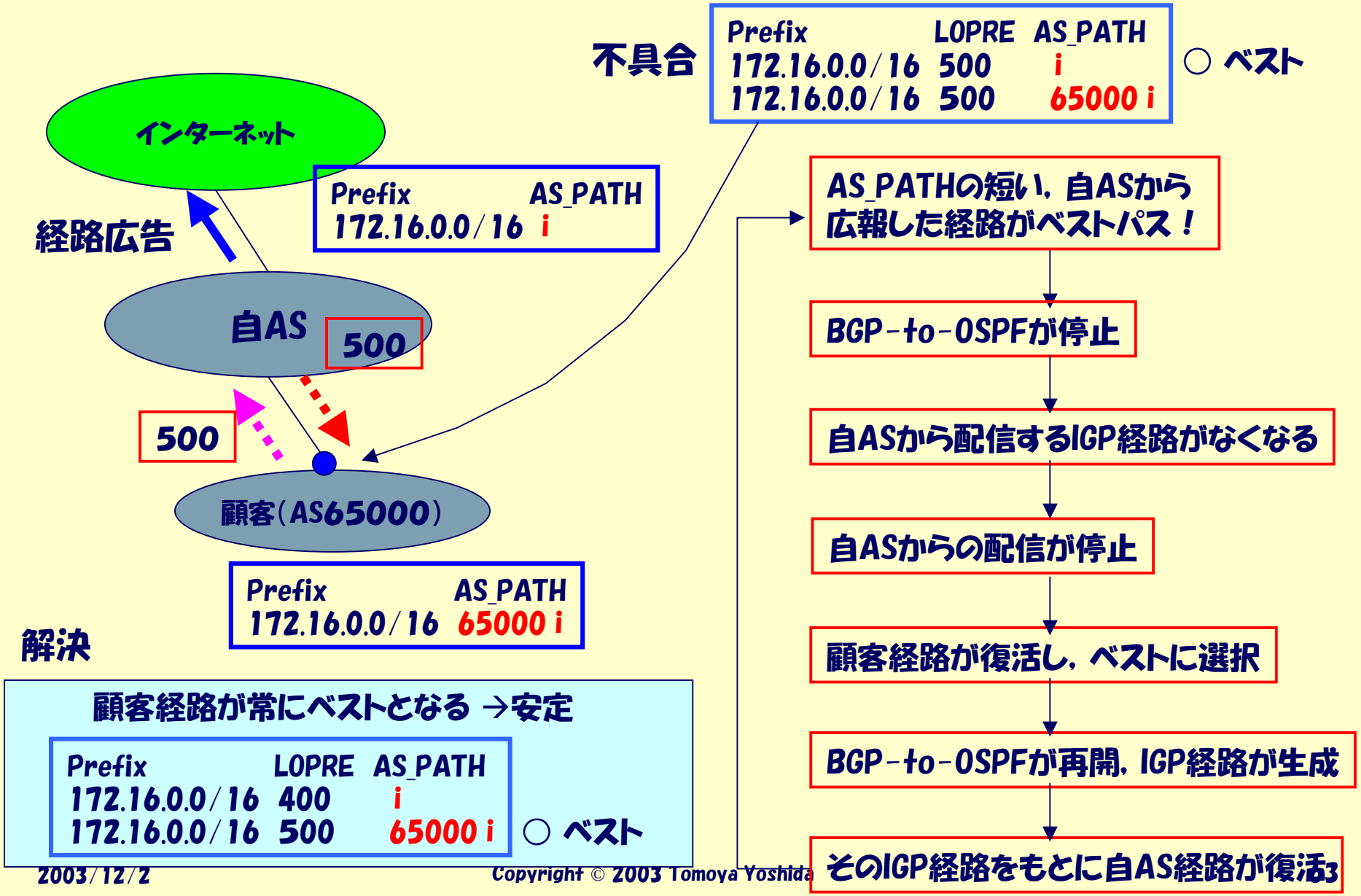
ポイント2: BGP顧客の次に，自AS内広報経路は優先させる

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た場合を想定して，ちゃんと優先させておく必要がある(エッジでFilterする手法も勿論ある)
→ 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)
→ もしも500にした場合には，制御方法によっては不具合が生じる

BGPポリシー設計(受信)



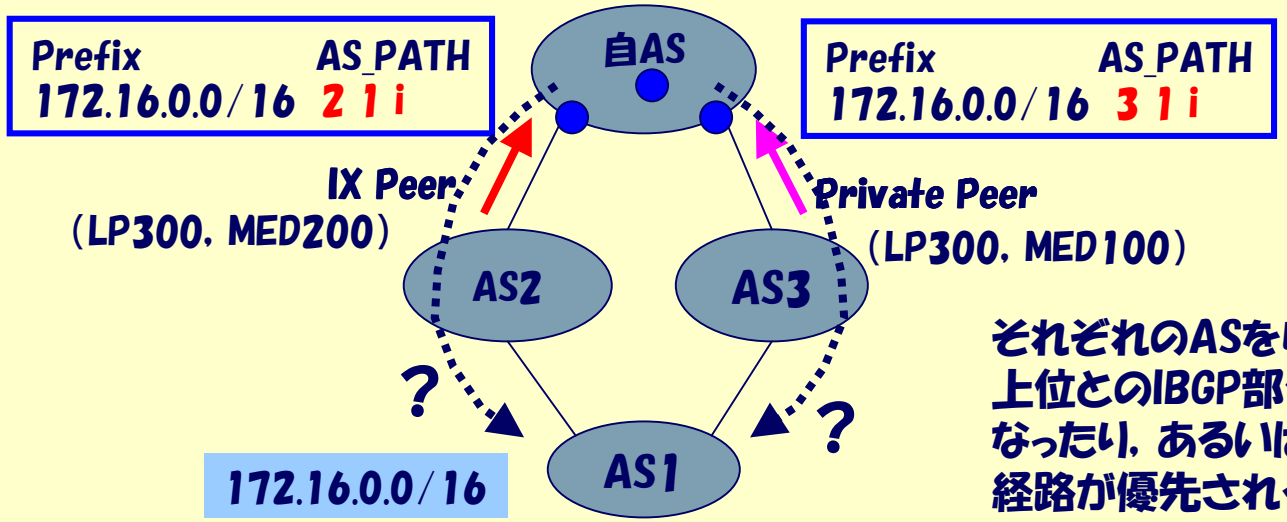
BGPポリシー設計(受信)

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

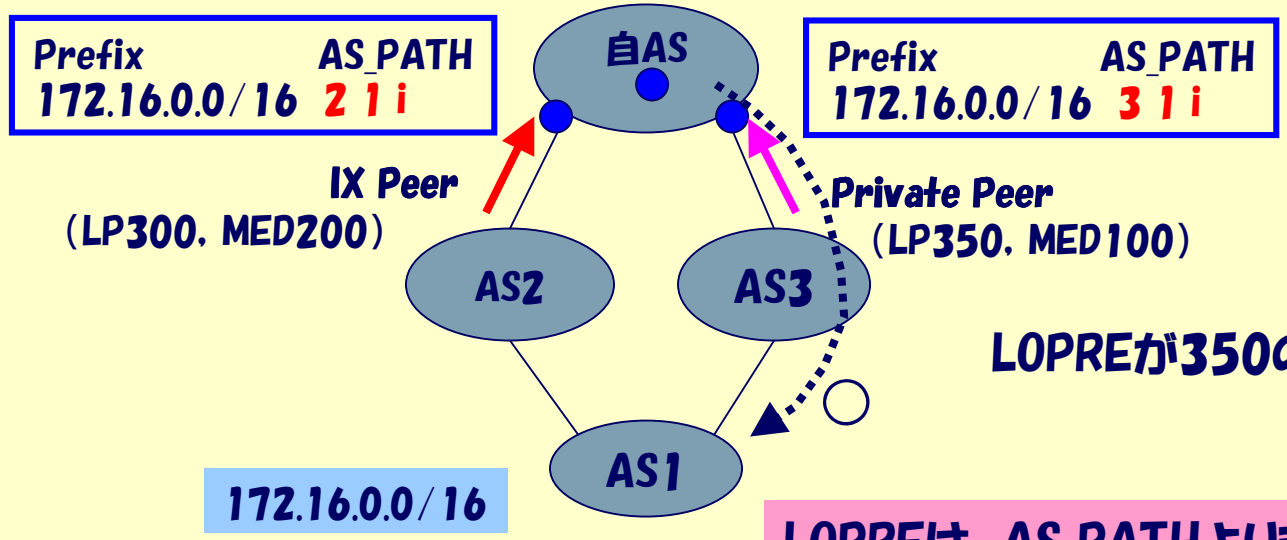
→ ピア経由の経路は，基本はAS_PATHによる制御
→ 異なるAS間ではMED比較の対象ではないので，ID勝負などになる場合もある
→ プライベートピアを優先されるように，LOPREを高く設定する場合もある
　　(例)LOPRE350

BGPポリシー設計(受信)



どっちに流れるんだ？

それぞれのASを収容するルータもしくは上位とのIBGP部分で，Router-ID勝負になったり，あるいは，先に受信したASの経路が優先される　など，まちまちになる

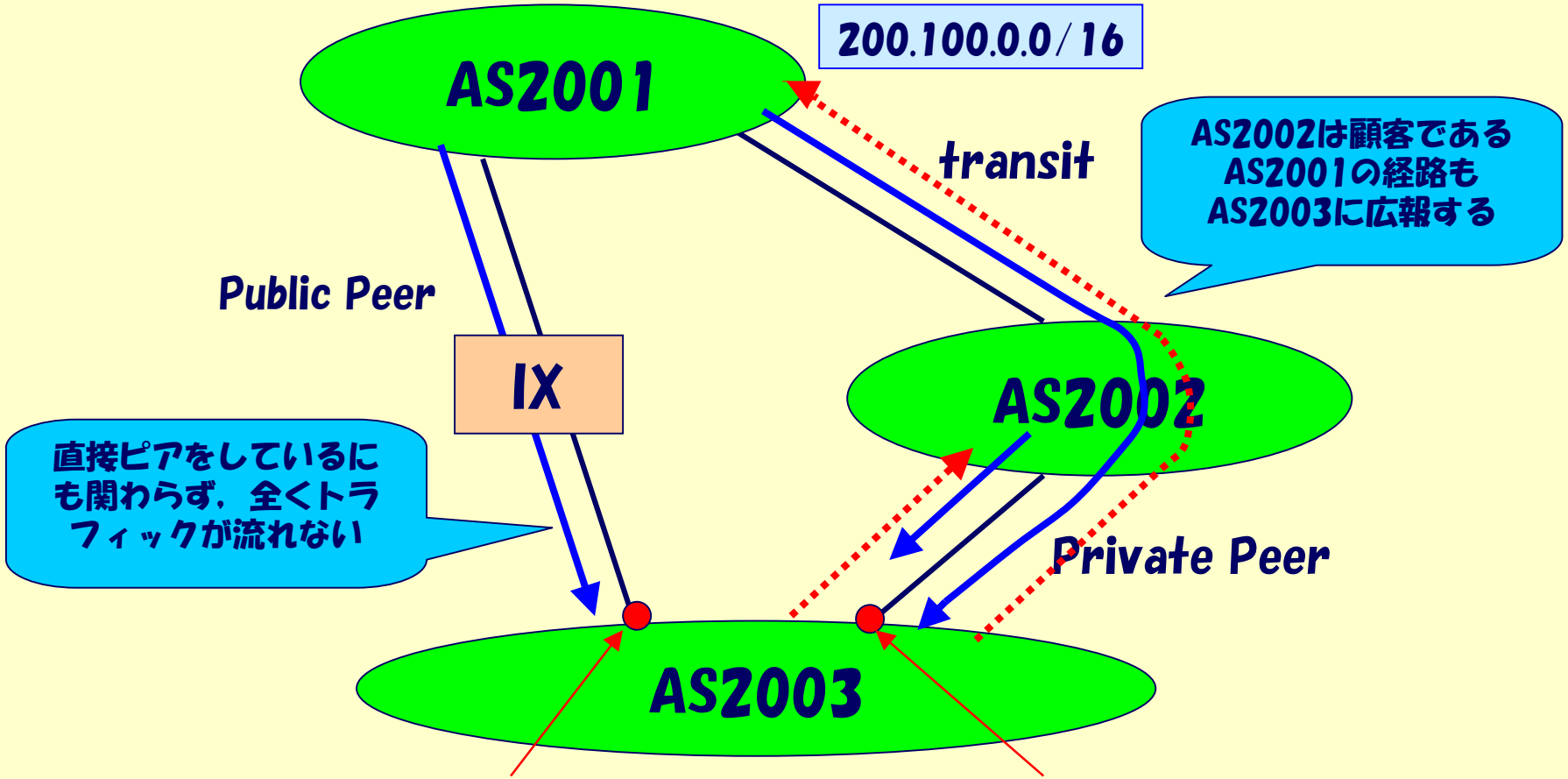


ベスト

LOPREが350のPrivateが常にベスト

LOPREは，AS_PATHよりも強いので注意！

直接ピアをしているのにトラフィックが流れない例



AS2003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

IXなどでPolicyをまとめたConfig例

■ Ciscoの例

```
router bgp 2003
 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

BGPポリシー設計(受信)

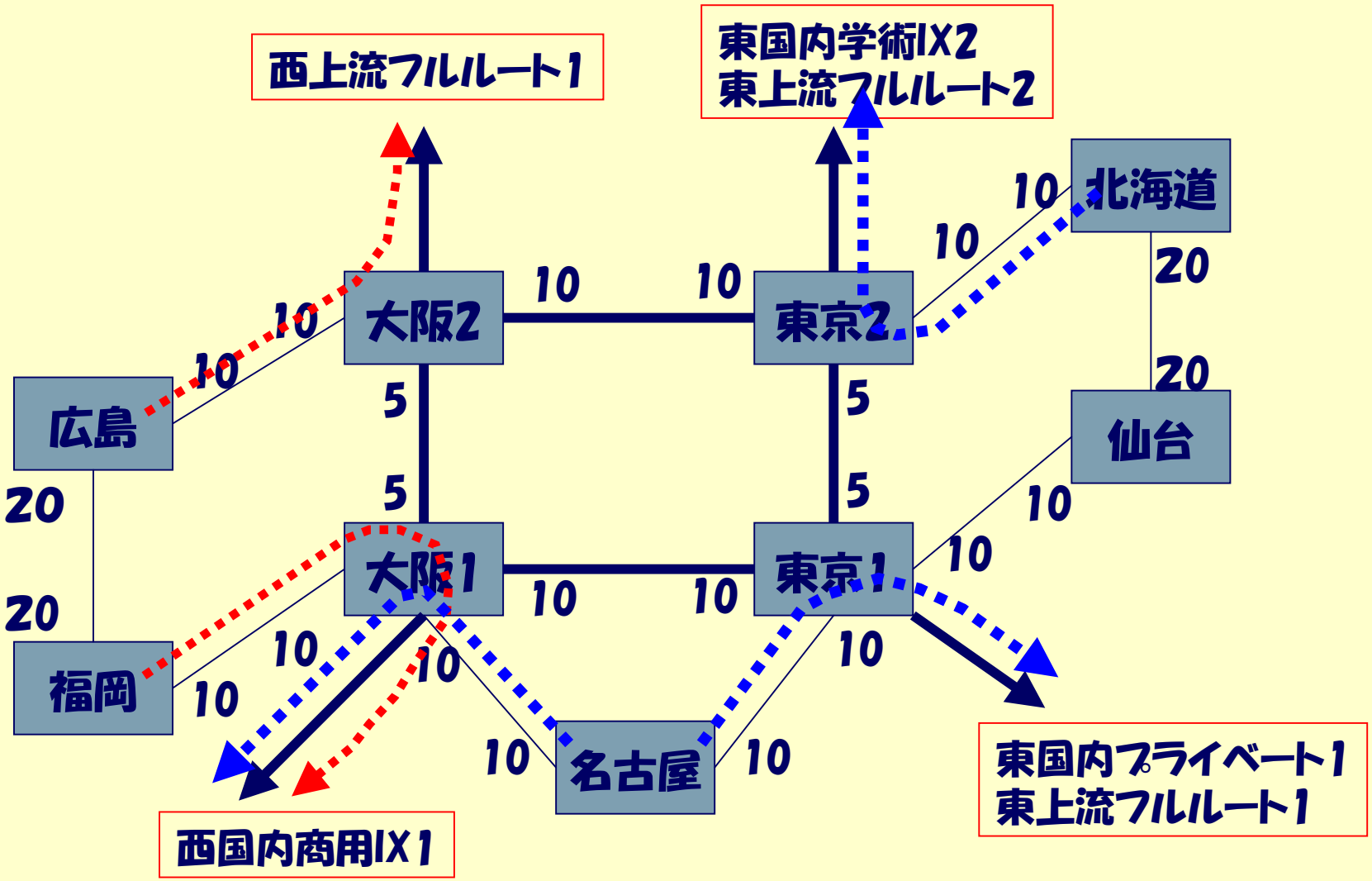
ポイント3-2: Closet Exit で，近いところからルーティング

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)

BGPポリシー設計(受信)

Closet Exit の場合には，どこに何を収容するのかが非常にきいてくる



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

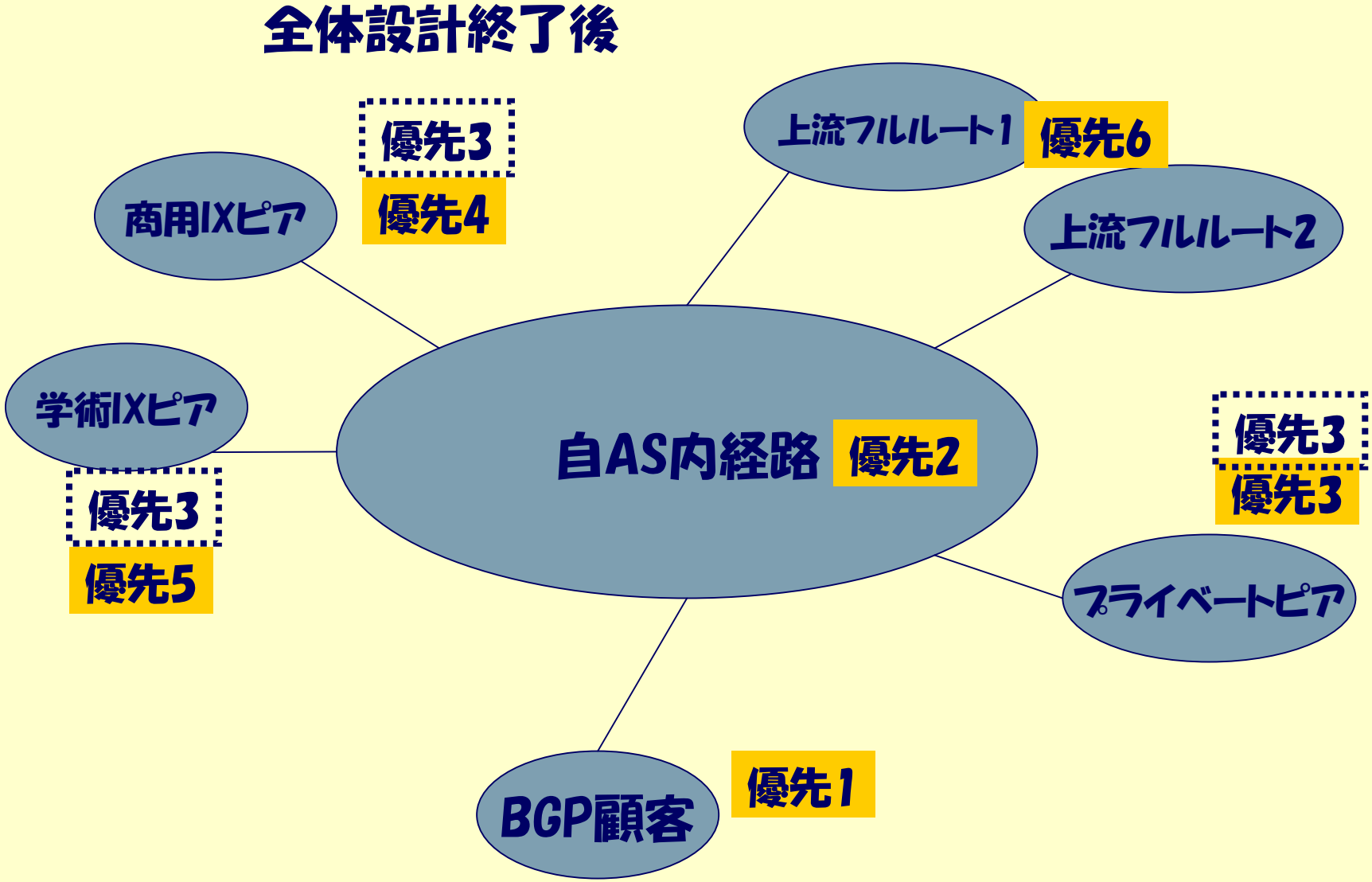
→ もっとも優先度が低いので，何でも良さそうだが，多くの実装で，LOPREのデフォルト値が100になっているため，その値よりも大きくしておくのが望ましいだろう
理由：仮にLOPRE50などで設定していた場合，うっかりミスで，フルルートを他のBGP接続からデフォルトで受信してしまうと，全てがそちらにひっぱりこまれてしまう

→ 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)

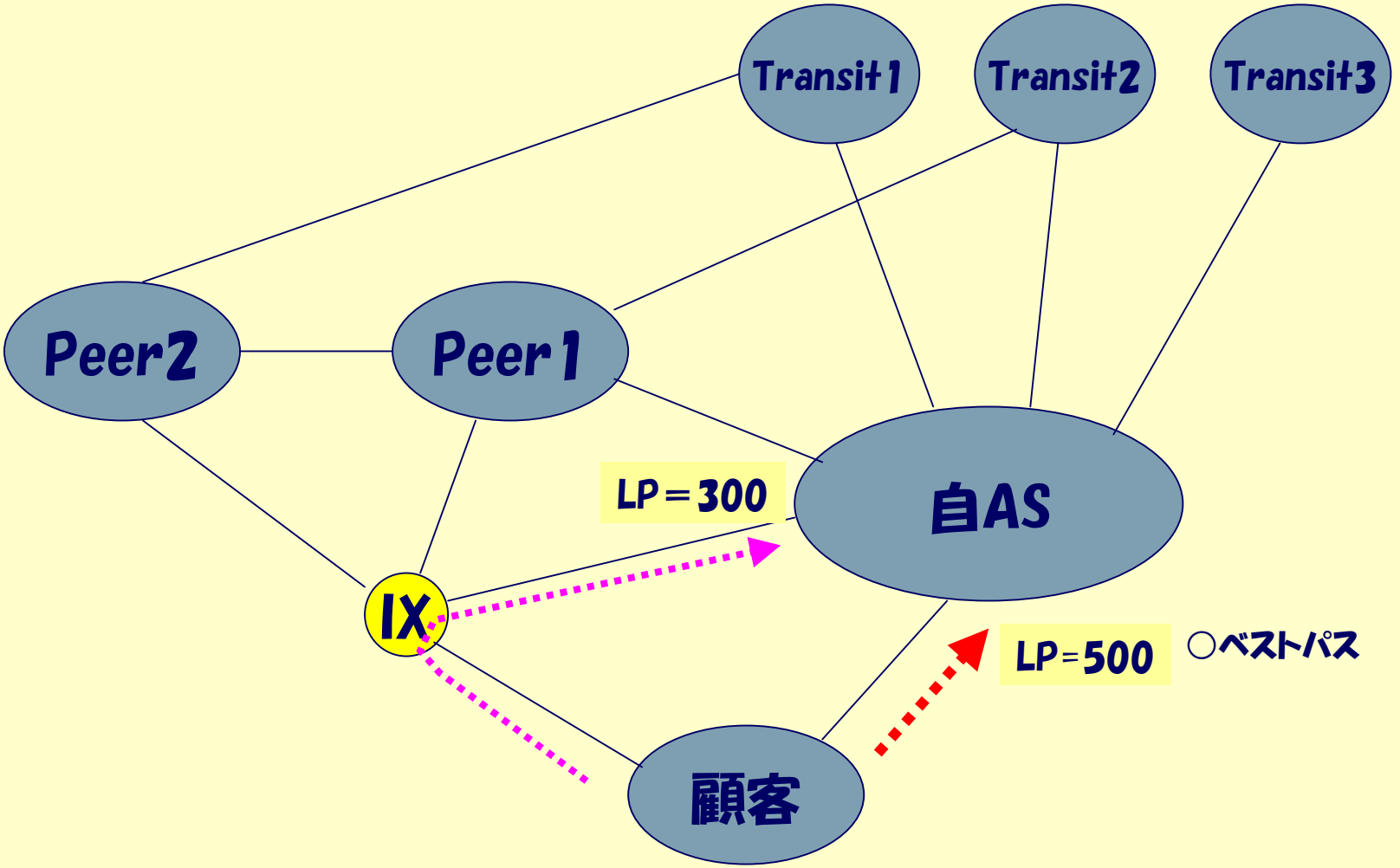
- Closet Exit の注意点
  - IGPメトリックがきいてくるので，OSPFのコスト設計が重要
  - Externalの回線をうまく分散収容する必要がある
    - おなじような位置付けのところに収容すると，ある部分ばかりに引き込まれて偏ってしまう
- 上流の制御
  - 上流が2つ以上ある場合，それぞれのCustomer経路は優先
    - 顧客コミュニティーにマッチしたら，優先度を高くして受信　など
    - 大抵上流ISP(Transit ISP)ではコミュニティーがインプリされている
  - それ以外のTransit経路は，コストの安いほうをとことん使う
    - 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
  - BGPのサービスを顧客に対してしないのであれば，受信ポリシーとして優先順位をつける必要はない．但し，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)



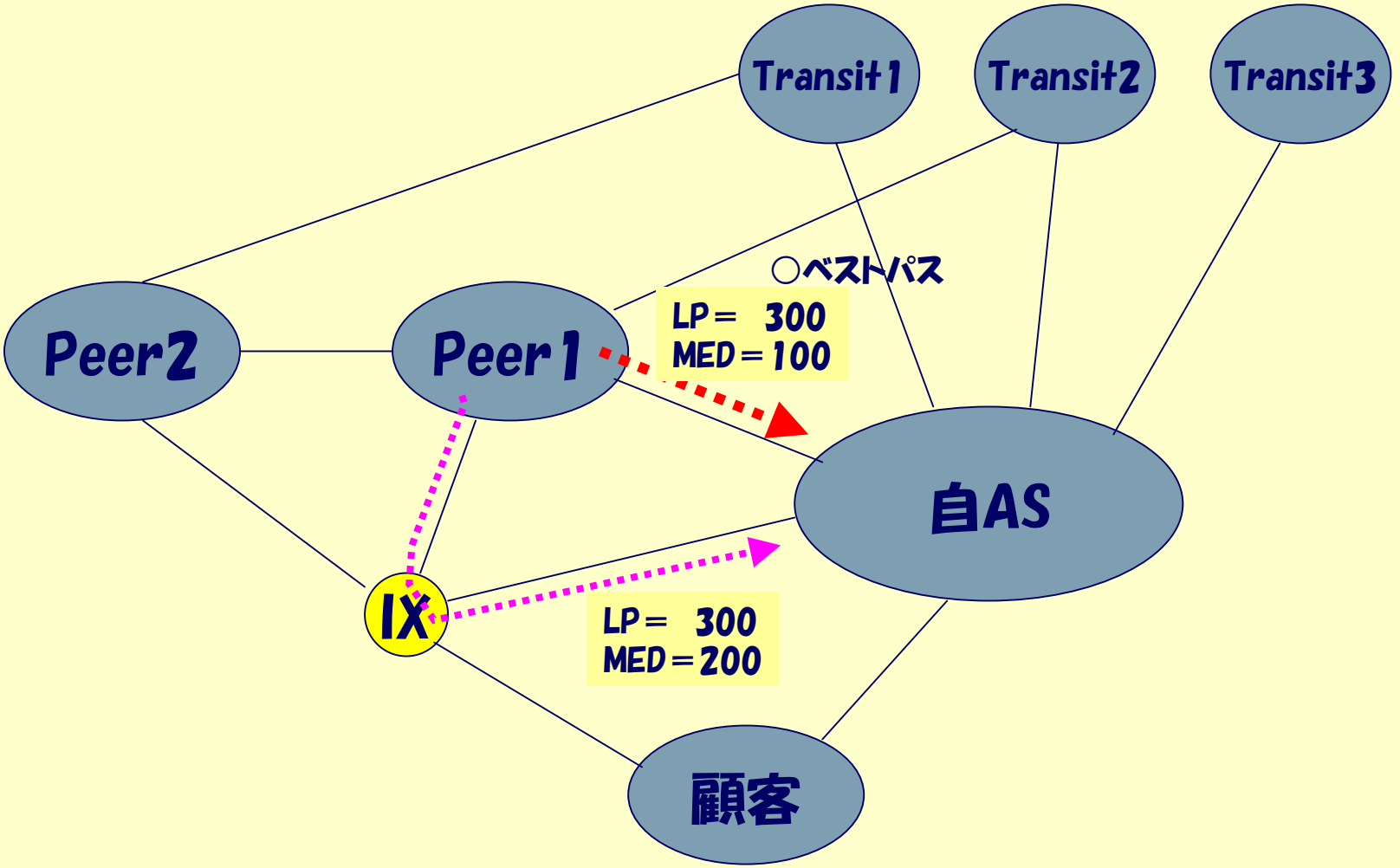
BGP受信ポリシー確認1

★顧客 かつ ピアの場合は顧客優先，切れたときはIX経由



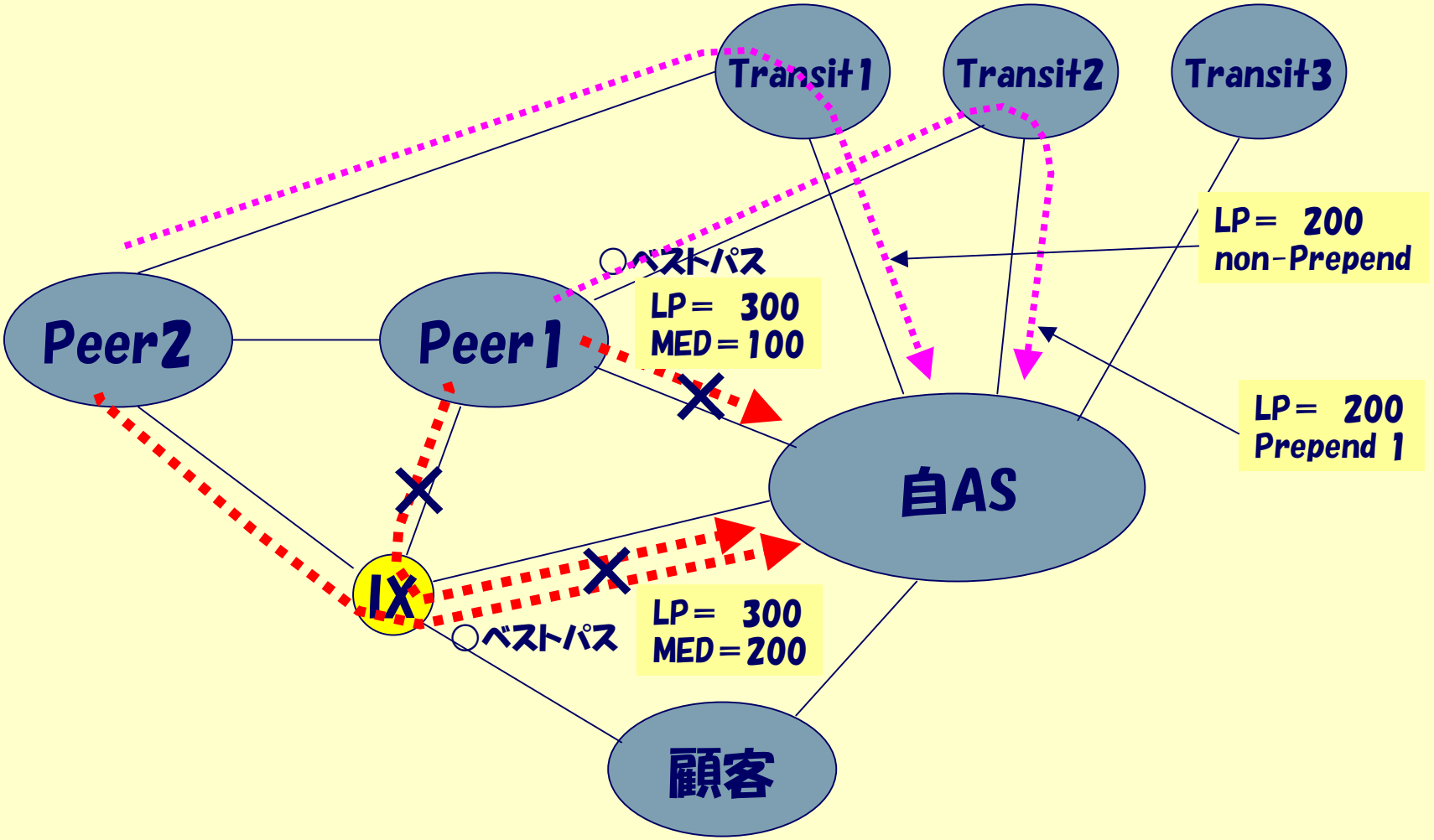
BGP受信ポリシー確認2

★PrivateピアとIXピアがある場合は，Privateピア優先



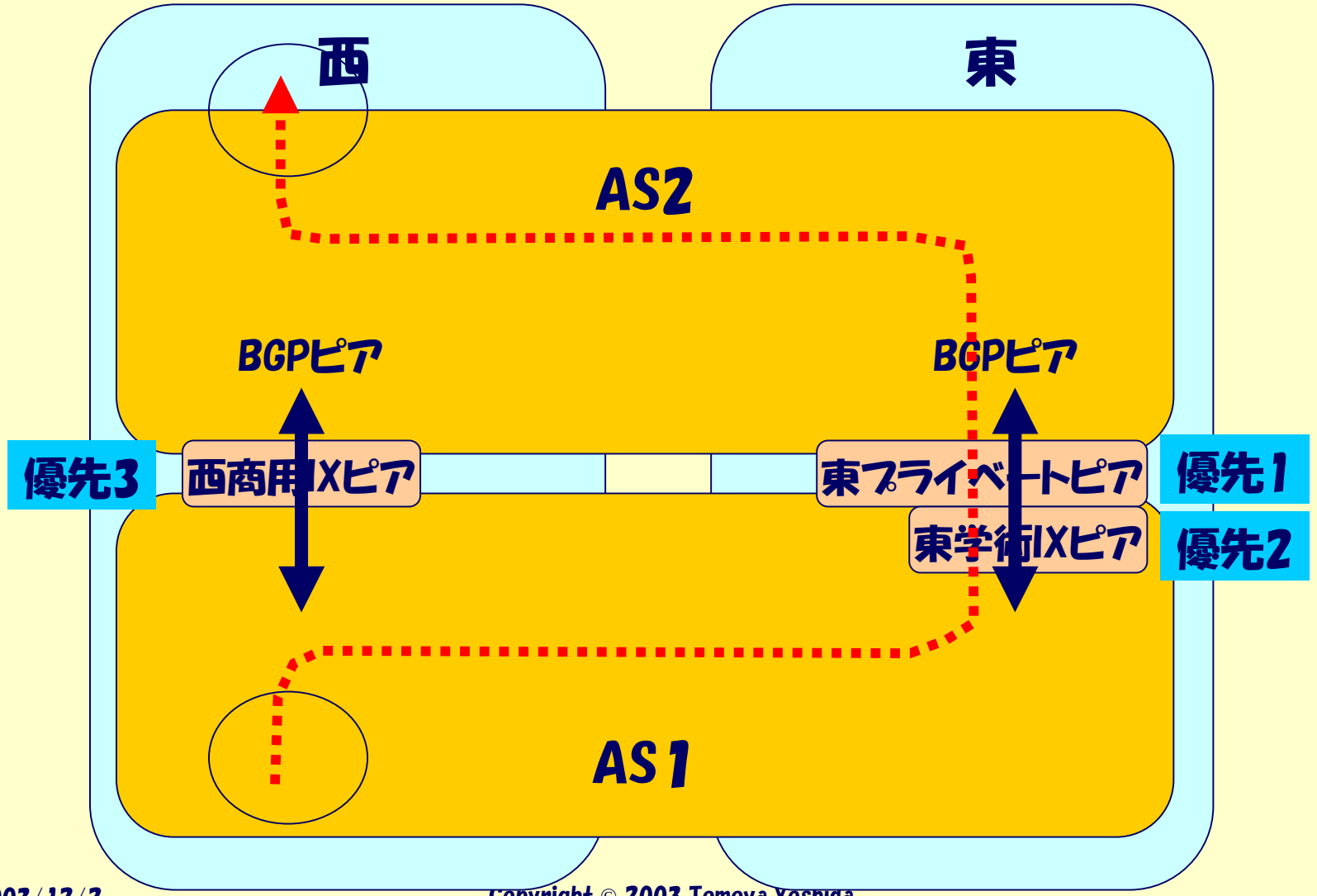
BGP受信ポリシー確認3

★国内ピアが落ちた場合には，（海外）Transitで救済したい



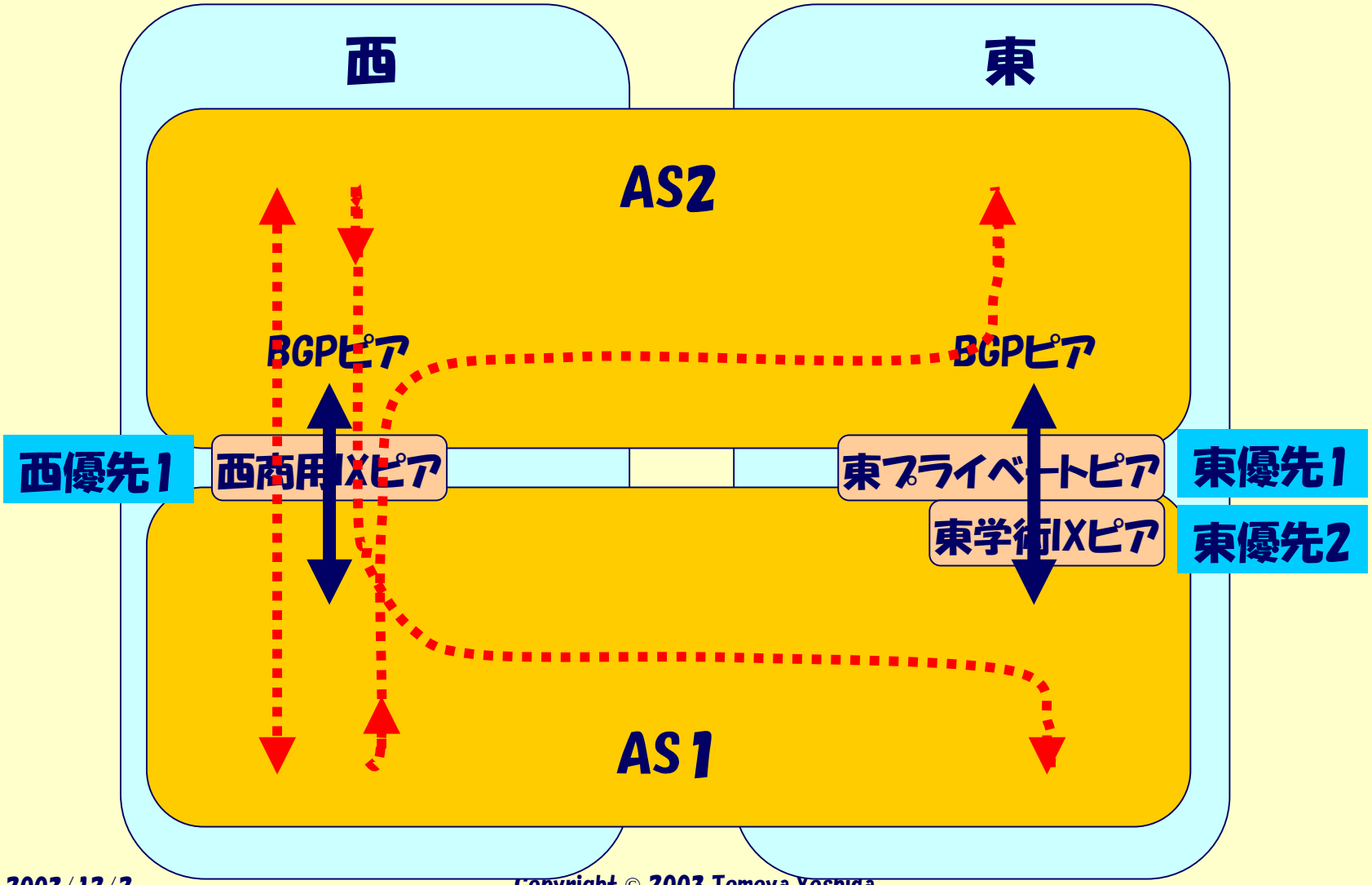
BGPポリシー設計(さらに)

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

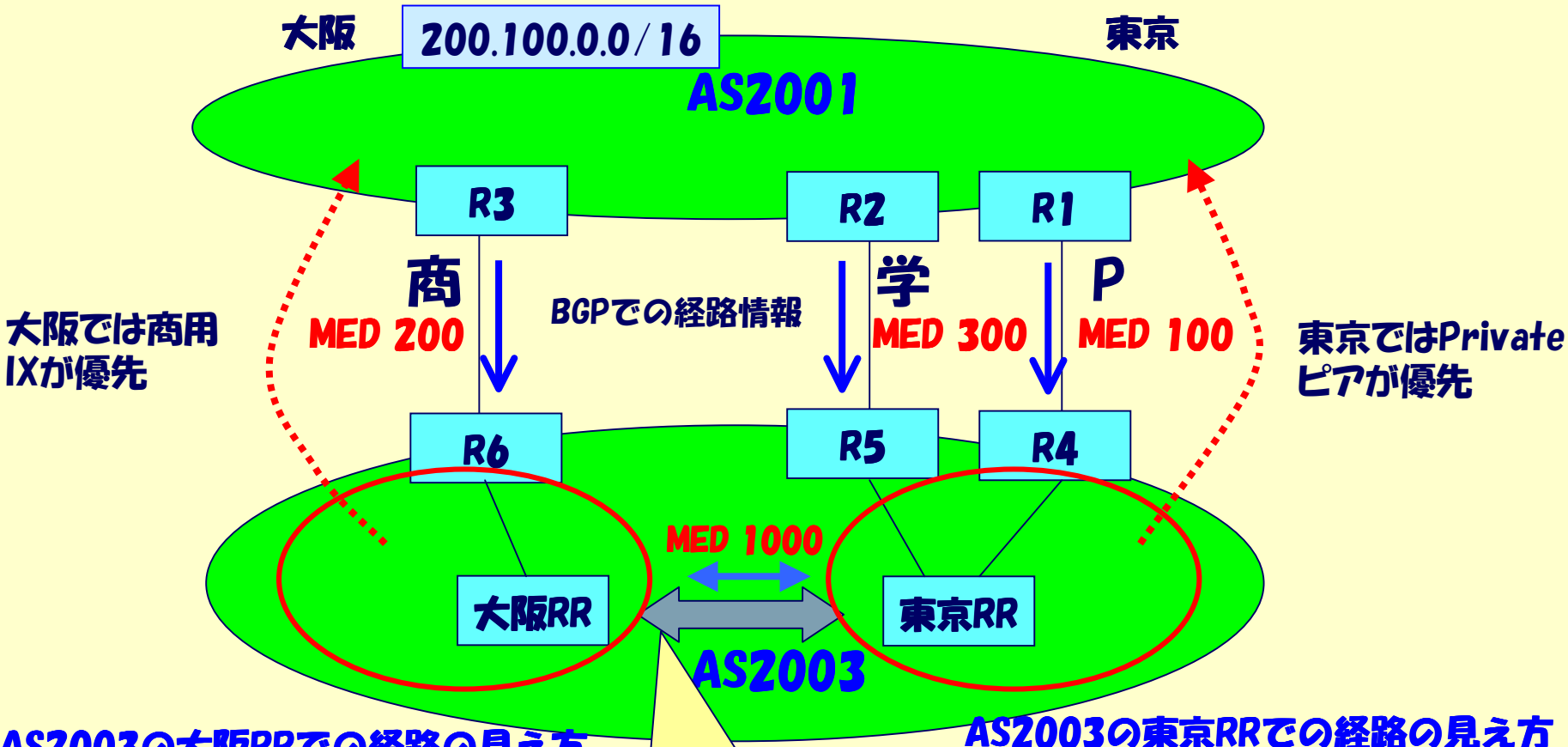


Hot-Potato と Cold-Potato

- Hot-Potato
  - 最も近いところから相手にパケットを出してしまおう
    - AS1西 → AS2西
    - AS1東 → AS2東
  - Closest Exit(とにかく近いところからパケットを出してしまう．通常IGPコストの最も近いところからルーティングさせる手法)
- Cold-Potato
  - Hot-Potatoのように近いところから，というポリシーではなく，例え遠くなったとしても，ポリシーに従ってルーティングさせる方法
    - AS1西 → AS1東 → AS2東 → AS2西
    - AS1東 → AS2東

Hot-Potatoによる経路制御

BGPでの経路情報
トラフィック



AS2003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

AS2003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	

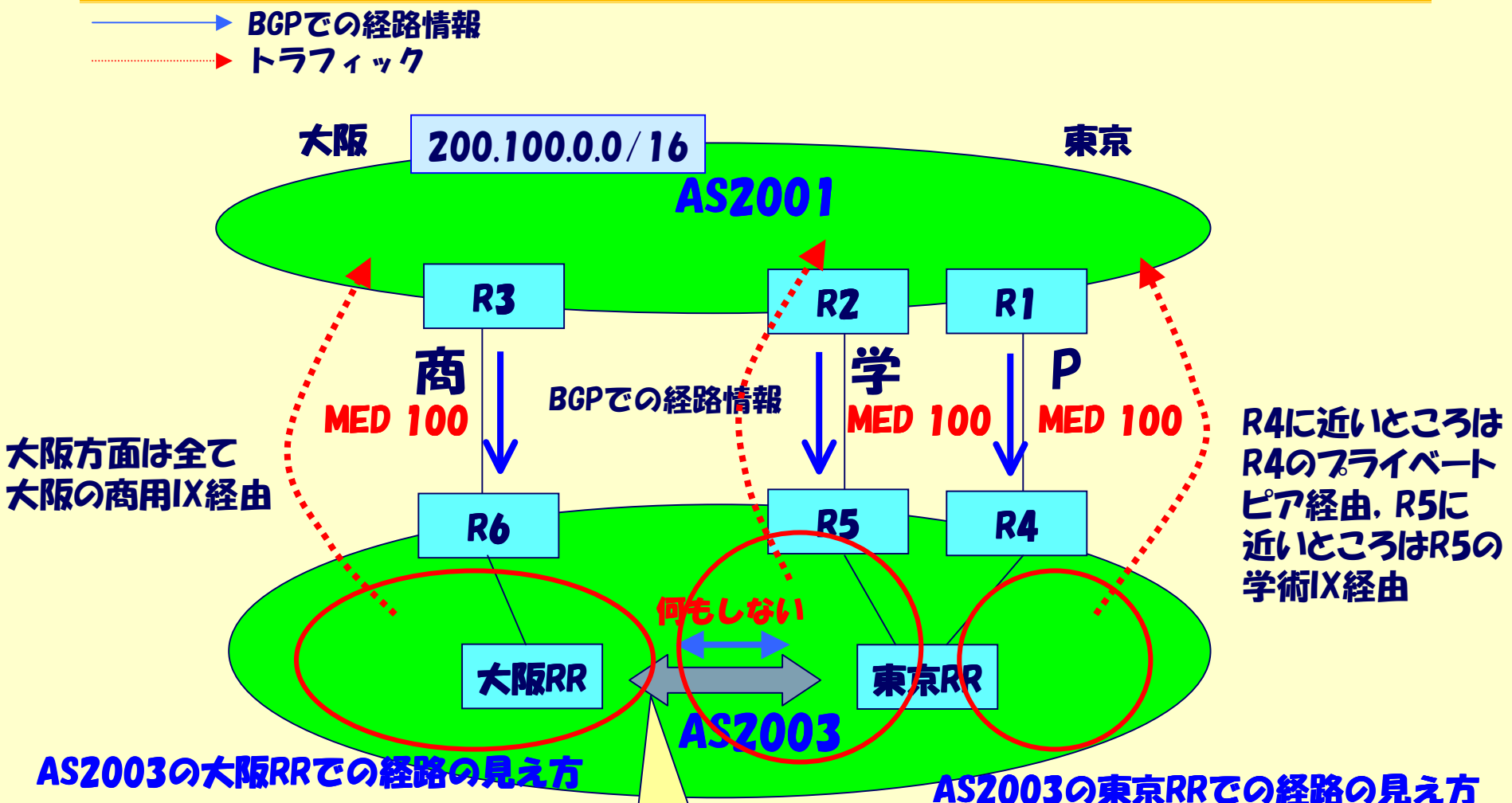
Hot-Potatoによる経路制御(Juniperの設定例)

```
protocols {
    bgp {
        group to-RR {
            type internal;
            local-address X.X.X.X;
            peer-as 2003;
            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 AS2003 {                       ← 対象ISP名
            from as-path AS2003;            ← 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 AS2003 "(2003 .*)";          ← 対象ISPのAS Numberで始まるAS-Path
       as-path AS-ALL "(.*)";                 を指定
```

東京RR

のコンフィグ例

Closet Exit(とにかく近いところから)



BGPポリシー設計（広告）

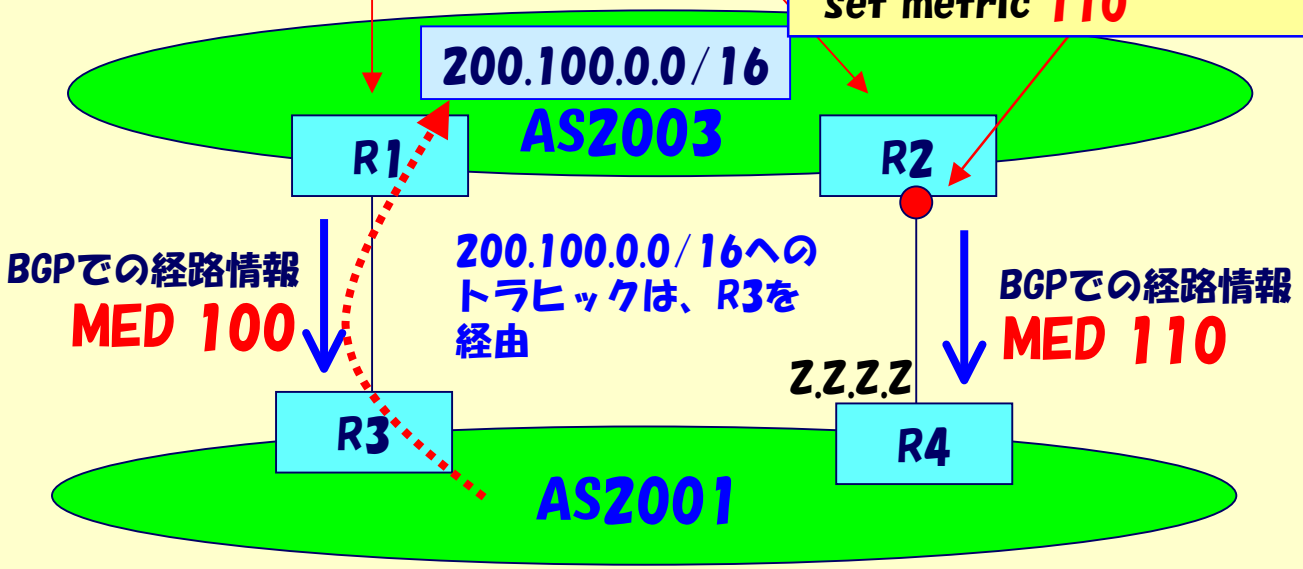
BGPポリシー設計(広告)

- 以下の3つのパスアトリビュート・手法を使って制御するのが基本
  - 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set
    - 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，自分主体で相手のLopreを制御したり，経路を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
    - 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(約3万経路で25M前後は消費する)

MEDを用いた制御

AS2003の出口でAS2001向けに経路をアナウンスするときにMEDを設定

```
router bgp 2003
 neighbor Z.Z.Z.Z remote-as 2001
 neighbor Z.Z.Z.Z route-map SET-MED out

route-map SET-MED permit 10
 set metric 110
```



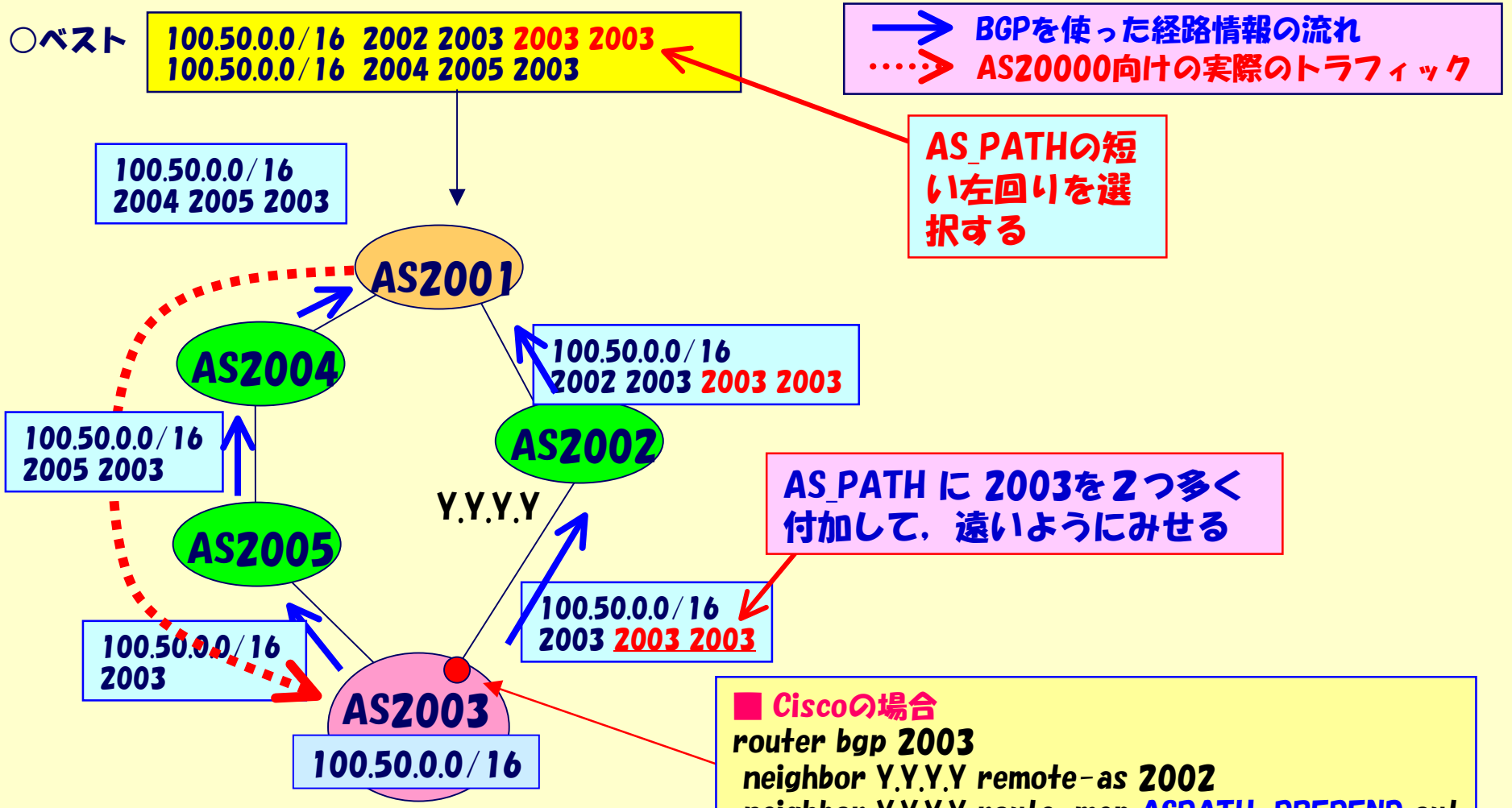
AS2001での経路の見え方

	Prefix	AS Path	MED	
	200.100.0.0/16	2003	110	
>	200.100.0.0/16	2003	100	◎ベストパス

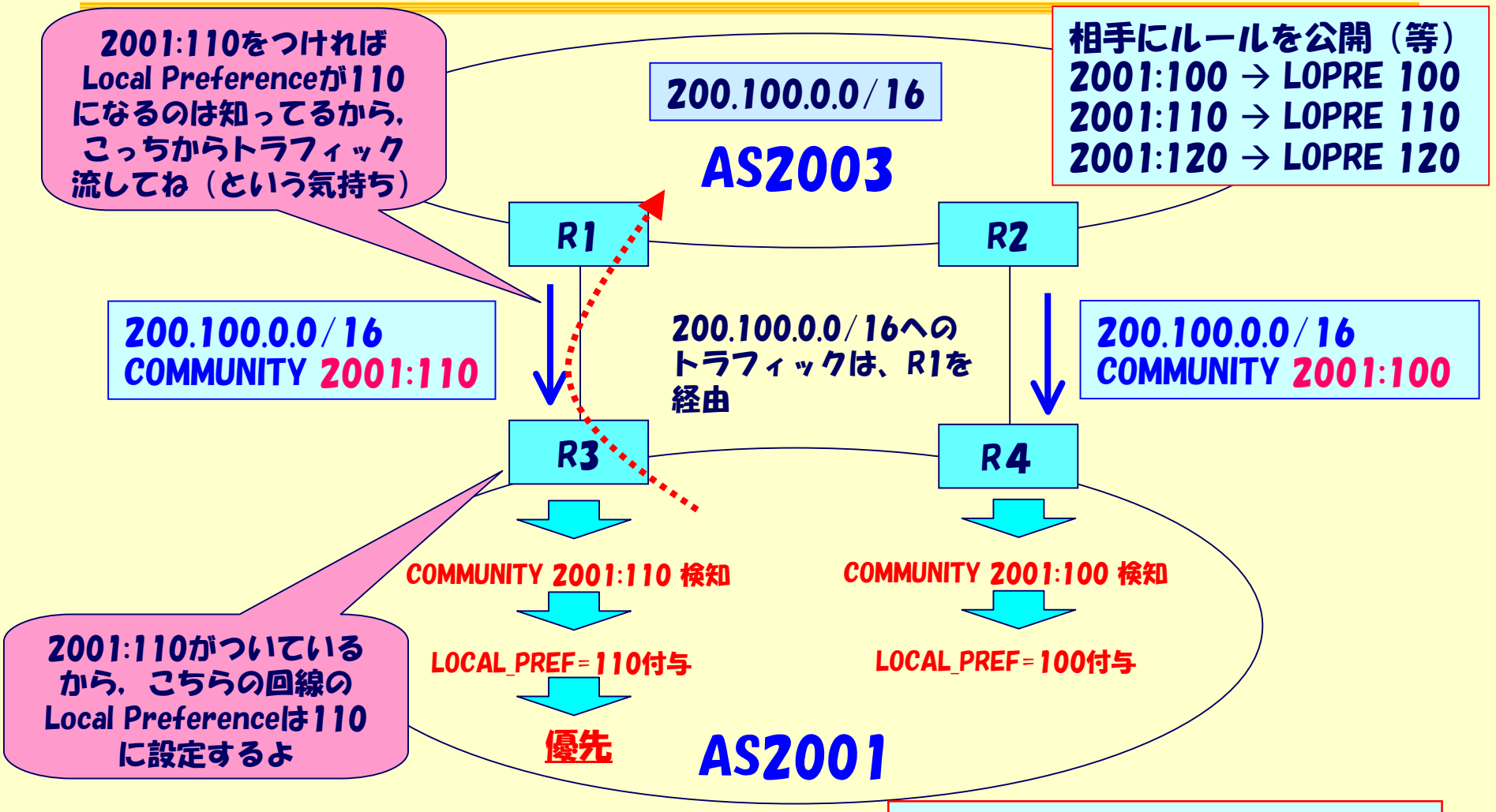
→ BGPを使った経路情報の流れ
⋯→ AS2003向けの実際のトラフィック

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



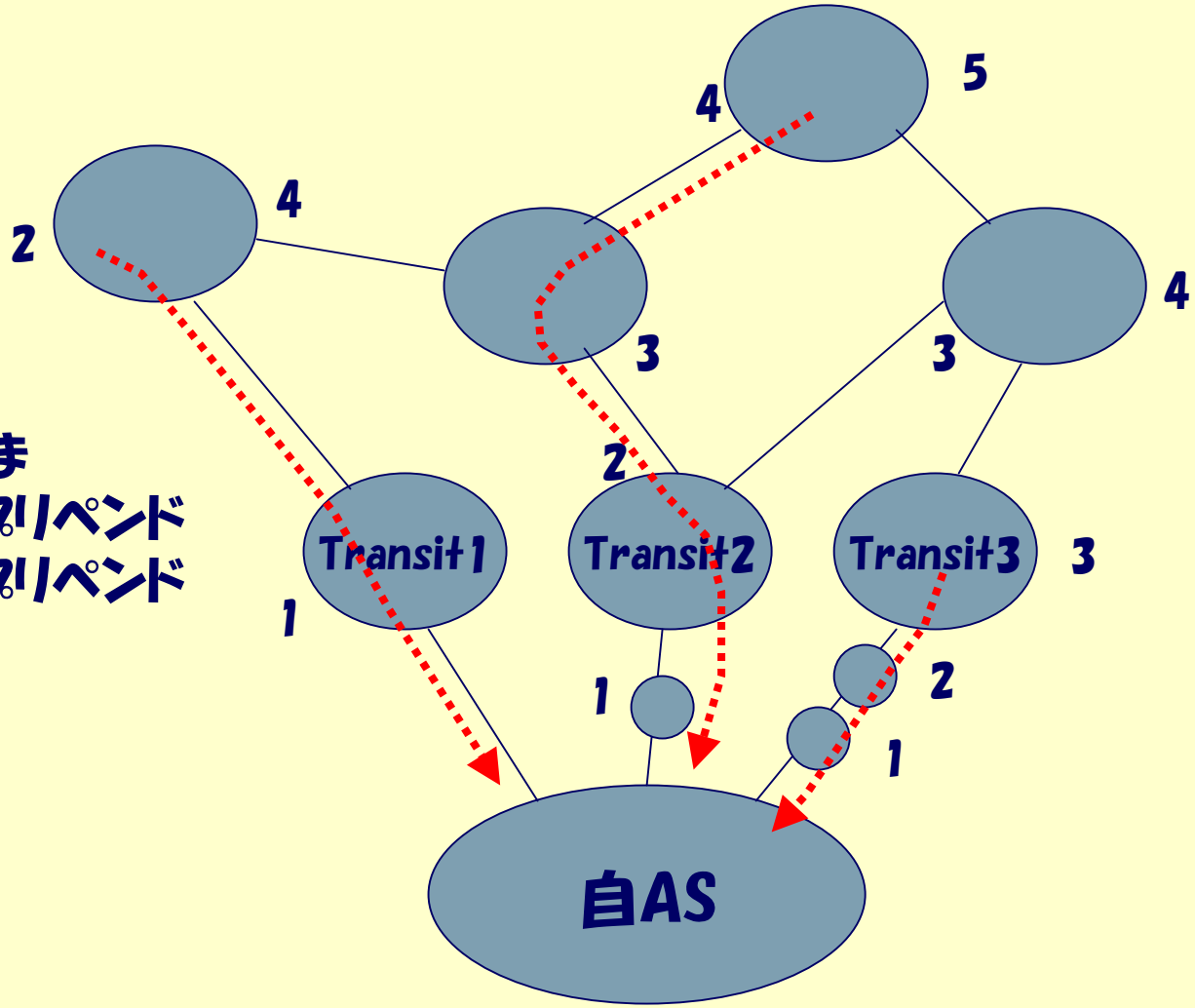
Communityによる戻りのトラフィック制御



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

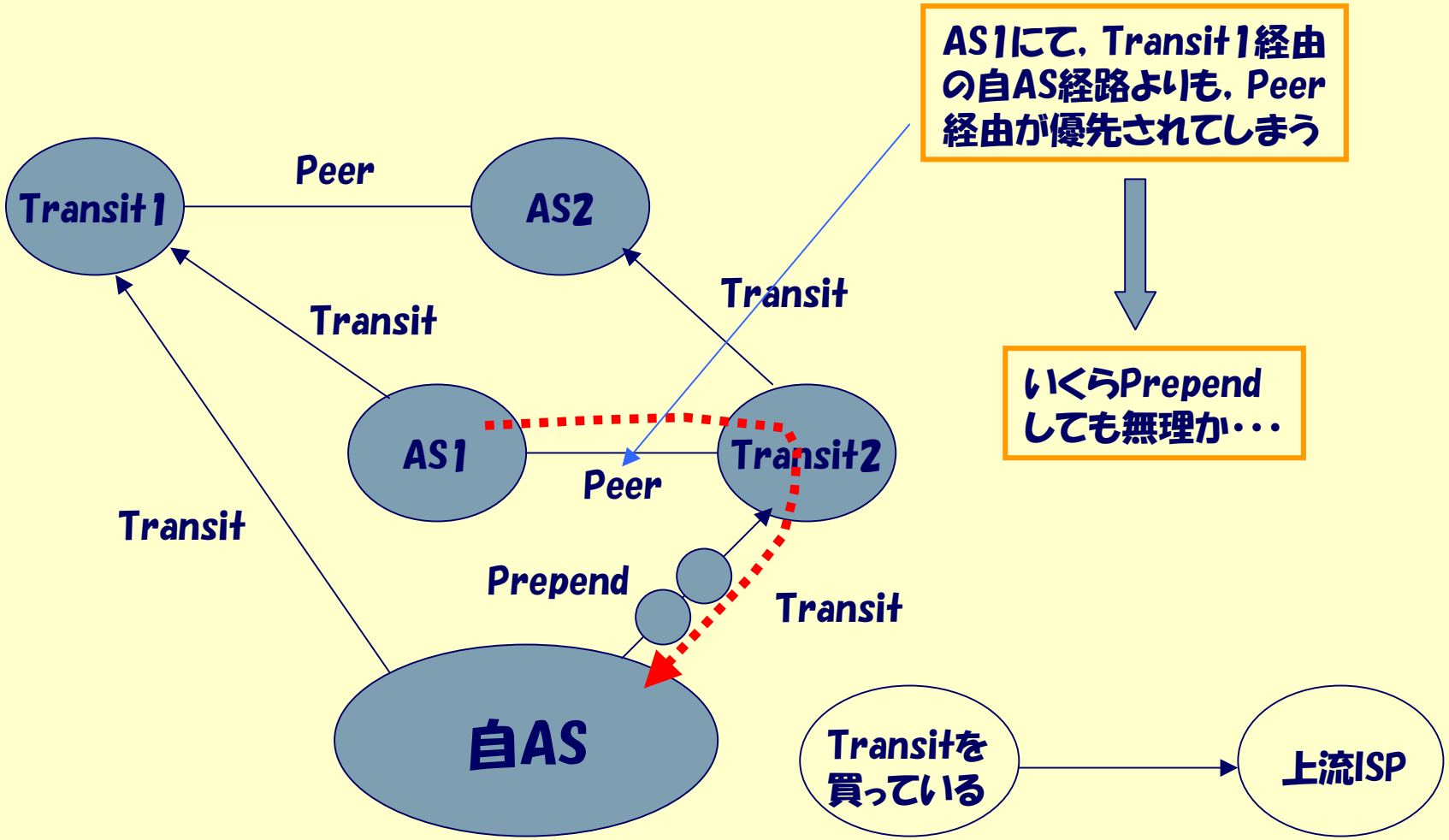


海外上流のトラフィック制御の難しさ

- 上流のその先のTopologyやPeerの関係などをきちんど日々把握している必要がある
  - 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？
    - 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，やっぱりAS-PATH制御？
  - いくらPrependしても，トラフィックがやってくる
    - 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



iBGP設計

iBGP設計

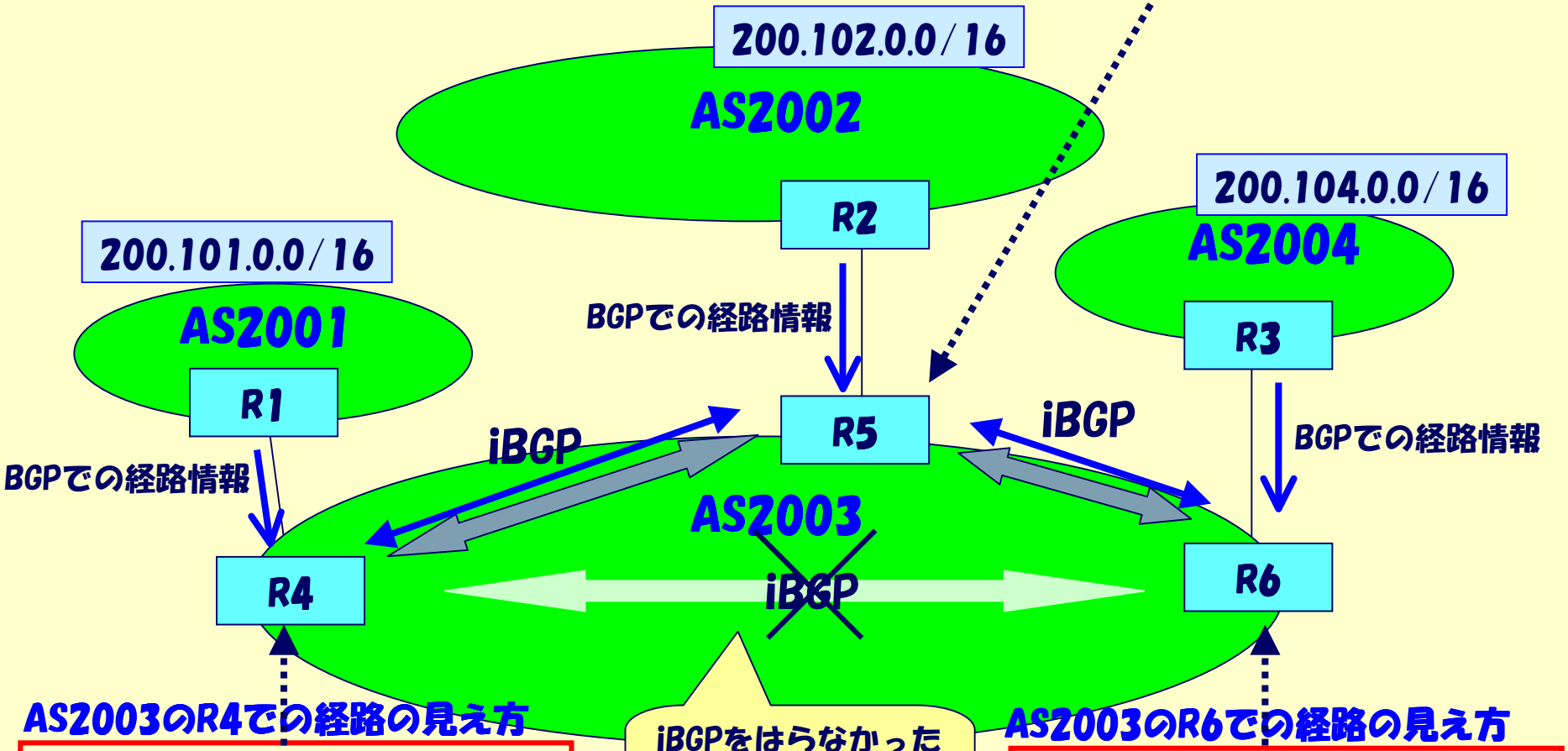
- 全BGPルータが正しくBGP経路情報を保有し，それぞれのルータが正しく経路選択出来るようにする
  - 同じ情報を保持する必要があるのとは違う
- BGPの経路は配送すべきところに適切に配送する
  - OSPFのデフォルトルートなどで十分なところはデフォルトでルーティングさせる
  - 内部の細かい経路は必要ないところには配送しない
    - BGPユーザ向けの階層にはフルルートのみを
    - それ以外の収容ルータ向けには経路を配送しない
- リフレクタ階層構造を利用
  - それほど数が多くなければ，フルメッシュのほうが適当な場合もある

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

AS2003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
>	200.104.0.0/16	2004



AS2003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

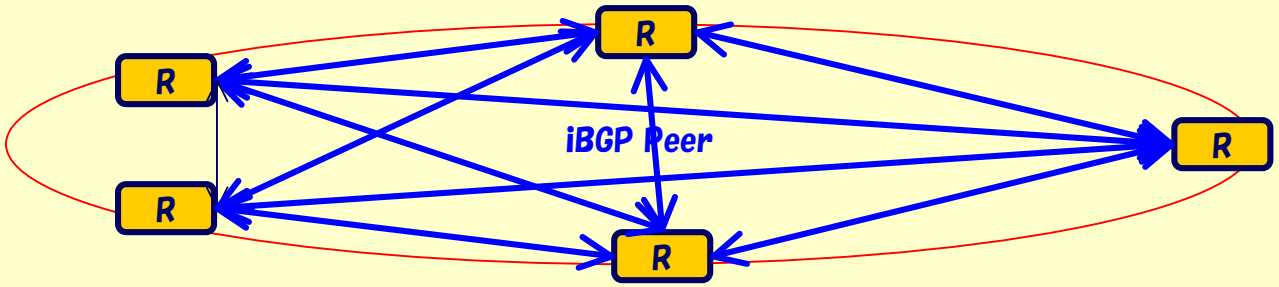
iBGPをはらなかったために，互いの経路情報を交換しない

AS2003のR6での経路の見え方

	Prefix	AS Path
>	200.102.0.0/16	2002
>	200.104.0.0/16	2004

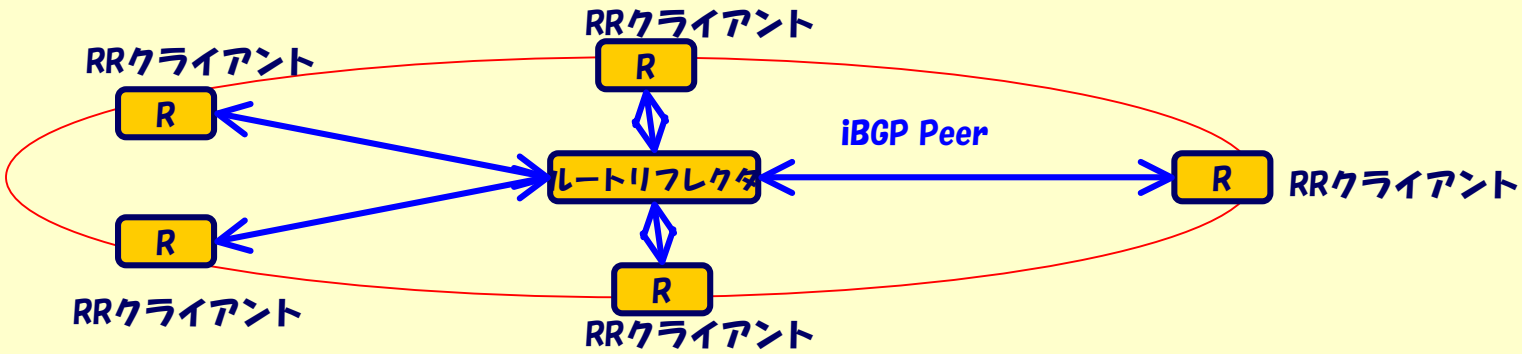
ルートリフレクタ(RR)

●一般的なiBGP Peer



iBGPはフルメッシュでなくてはならない

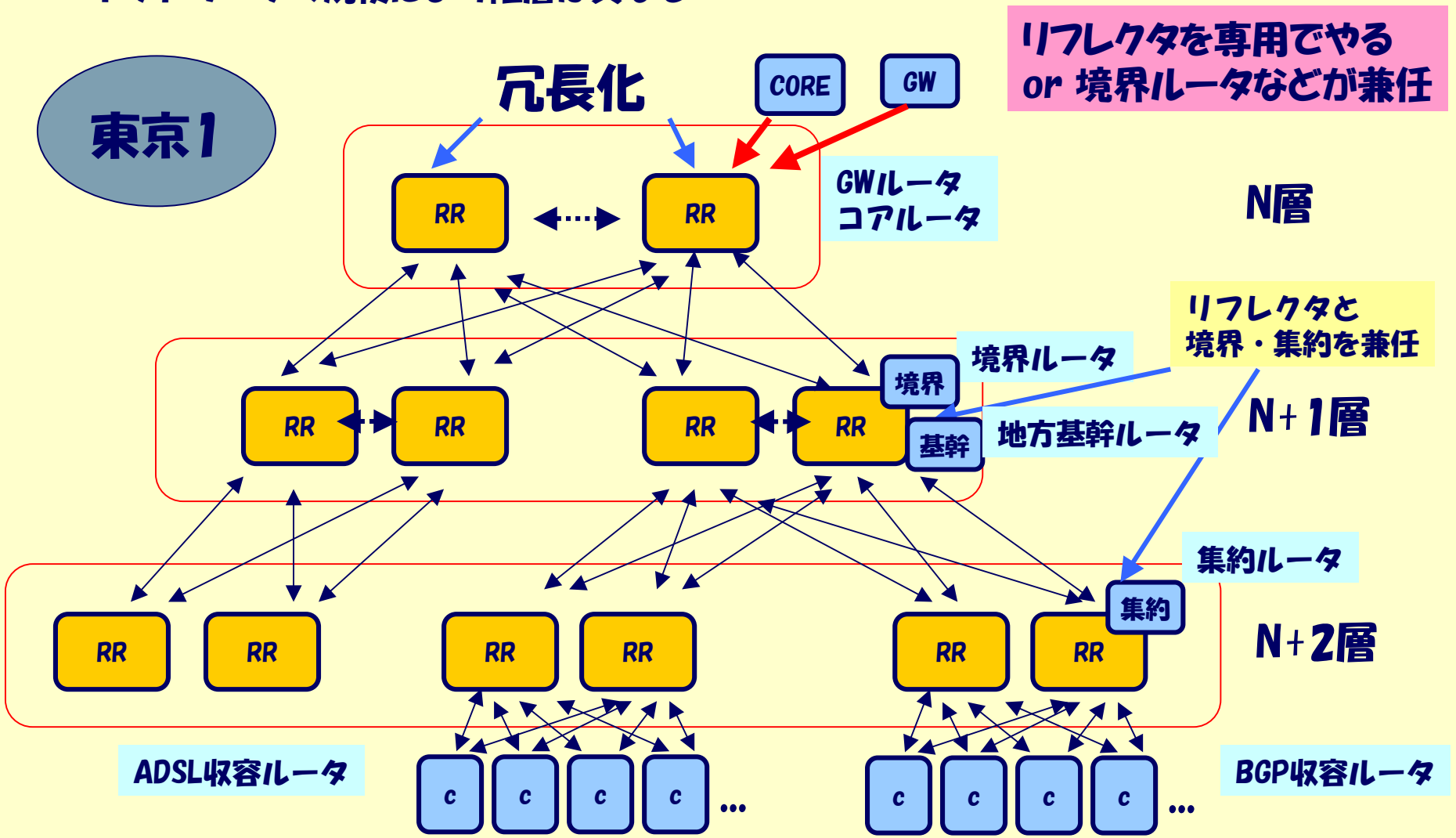
●ルートリフレクタ(RR)を使用したiBGP Peer



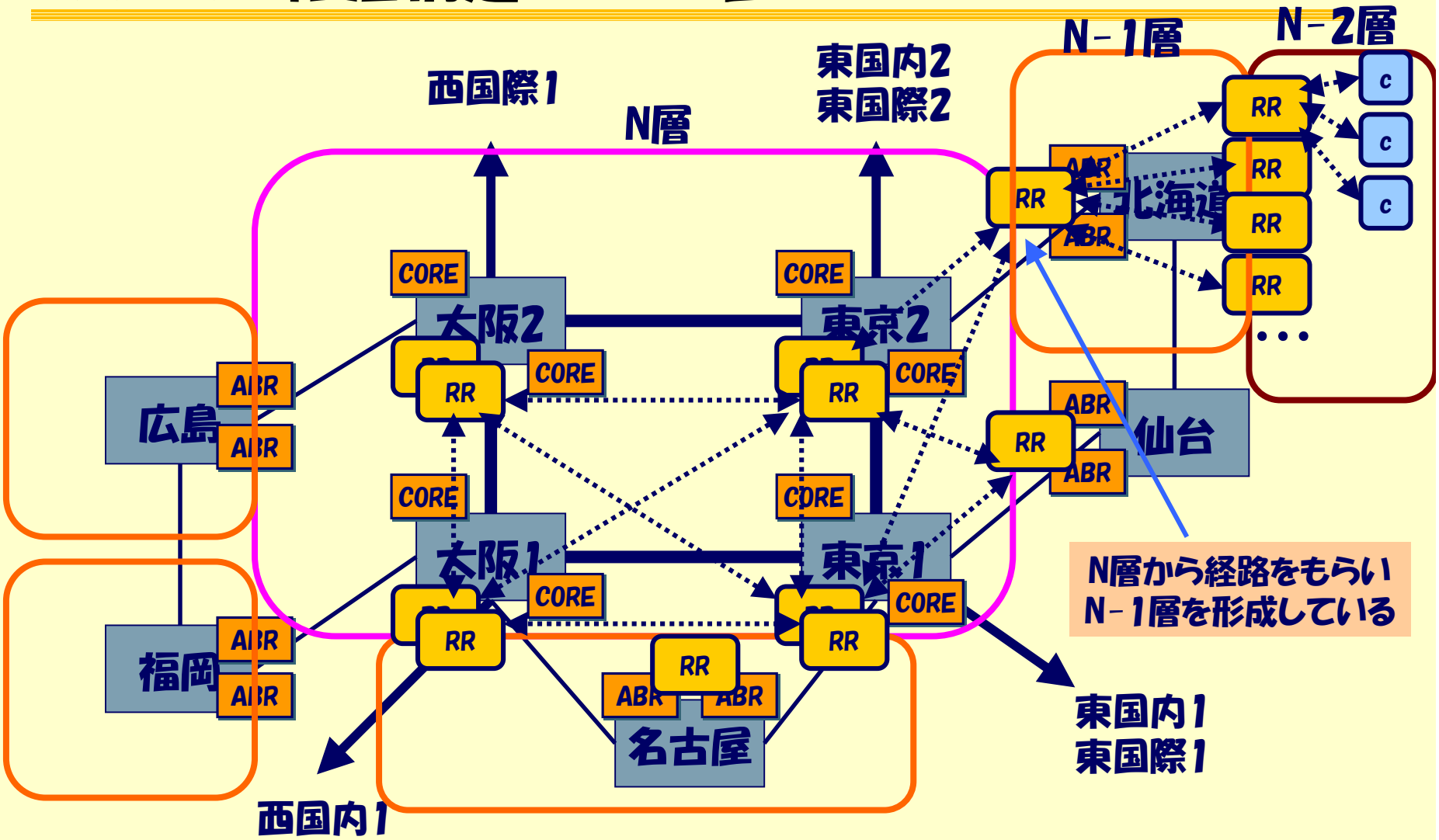
iBGPフルメッシュをルートリフレクタを用いたPeerにより代用

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

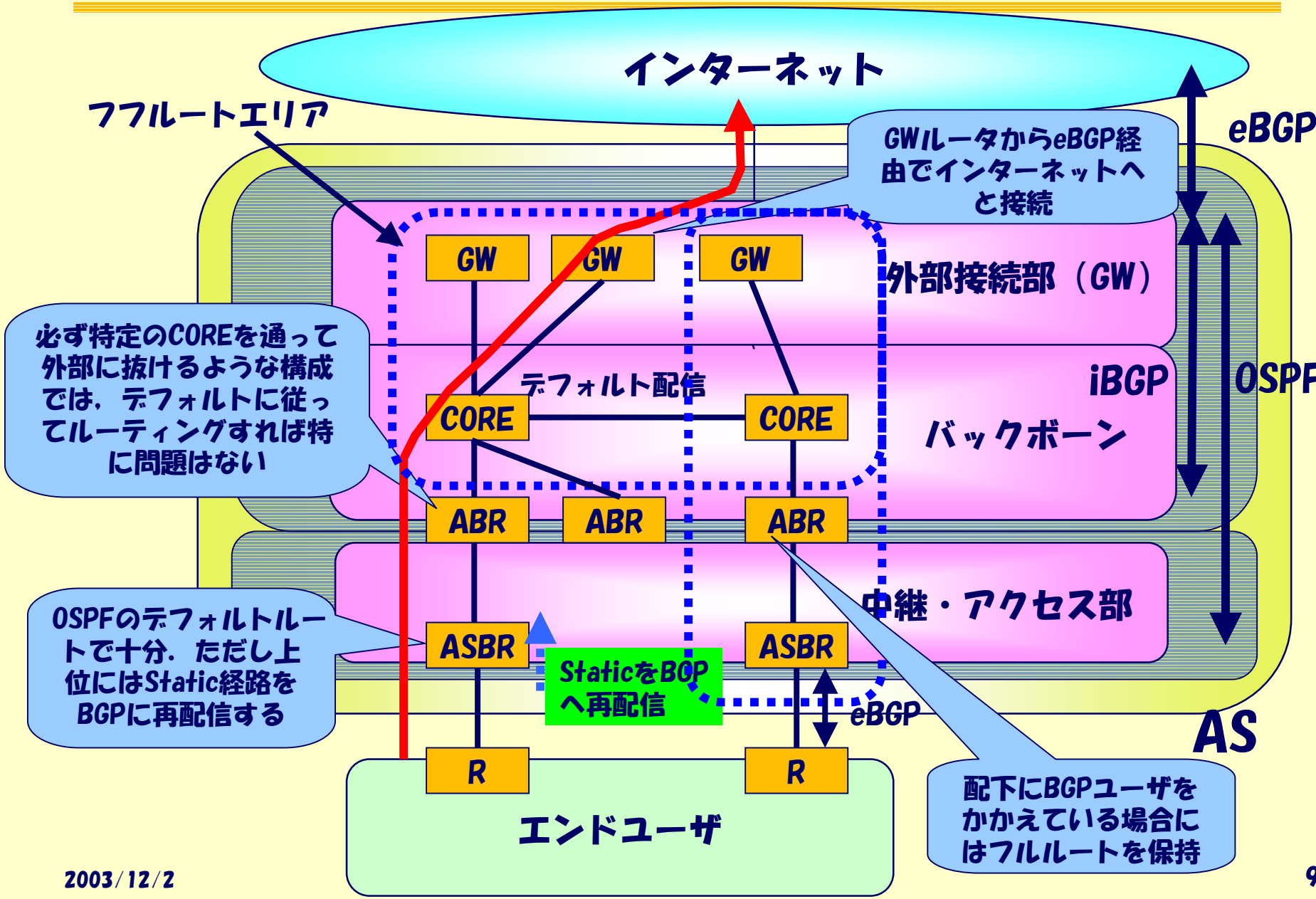


リフレクタ階層構造イメージ図

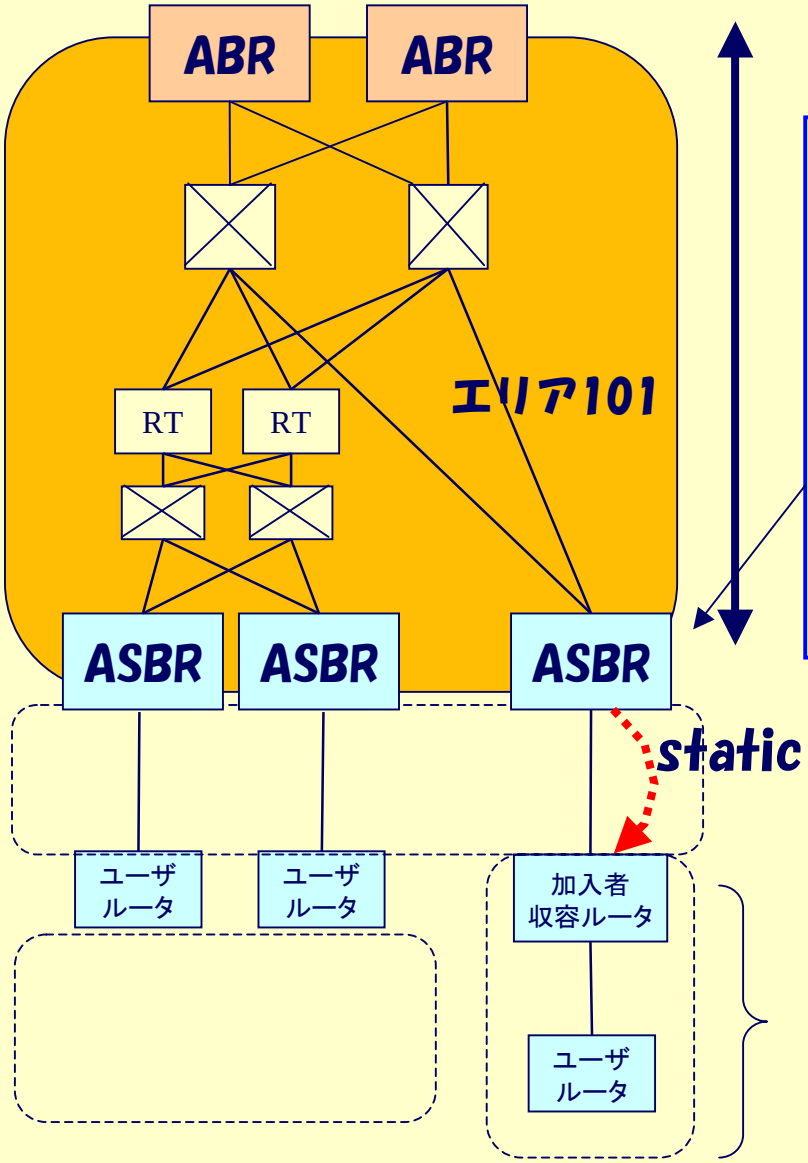


東京1や大阪1などでも，さらに階層化されるが，ここでは省略

行きのルーティング


インターネット
フフルートエリア
GWルータからeBGP経由でインターネットへと接続
eBGP
GW
GW
GW
外部接続部（GW）
必ず特定のCOREを通って外部に抜けるような構成では，デフォルトに従ってルーティングすれば特に問題はない
デフォルト配信
CORE
CORE
iBGP
バックボーン
OSPF
ABR
ABR
ABR
中継・アクセス部
OSPFのデフォルトルートで十分．ただし上位にはStatic経路をBGPに再配信する
ASBR
ASBR
StaticをBGPへ再配信
eBGP
AS
R
R
エンドユーザ
配下にBGPユーザをかかえている場合にはフルルートを保持

ユーザStatic経路をBGPに再配信



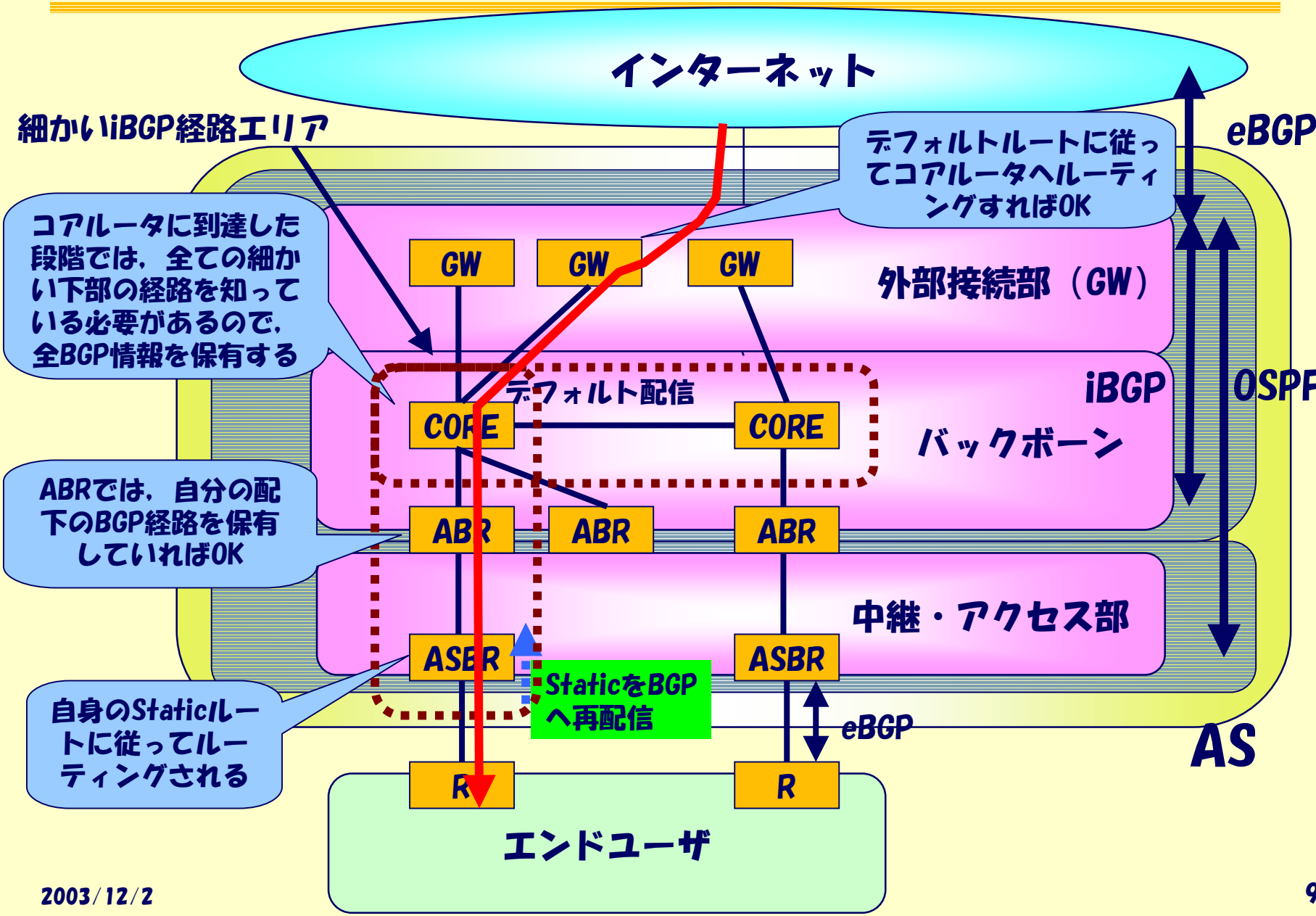
中継・アクセス部

■ Ciscoの例

```
router bgp 2003
 redistribute static route-map s-to-bgp
 neighbor X.X.X.X remote-as 2003
 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
```

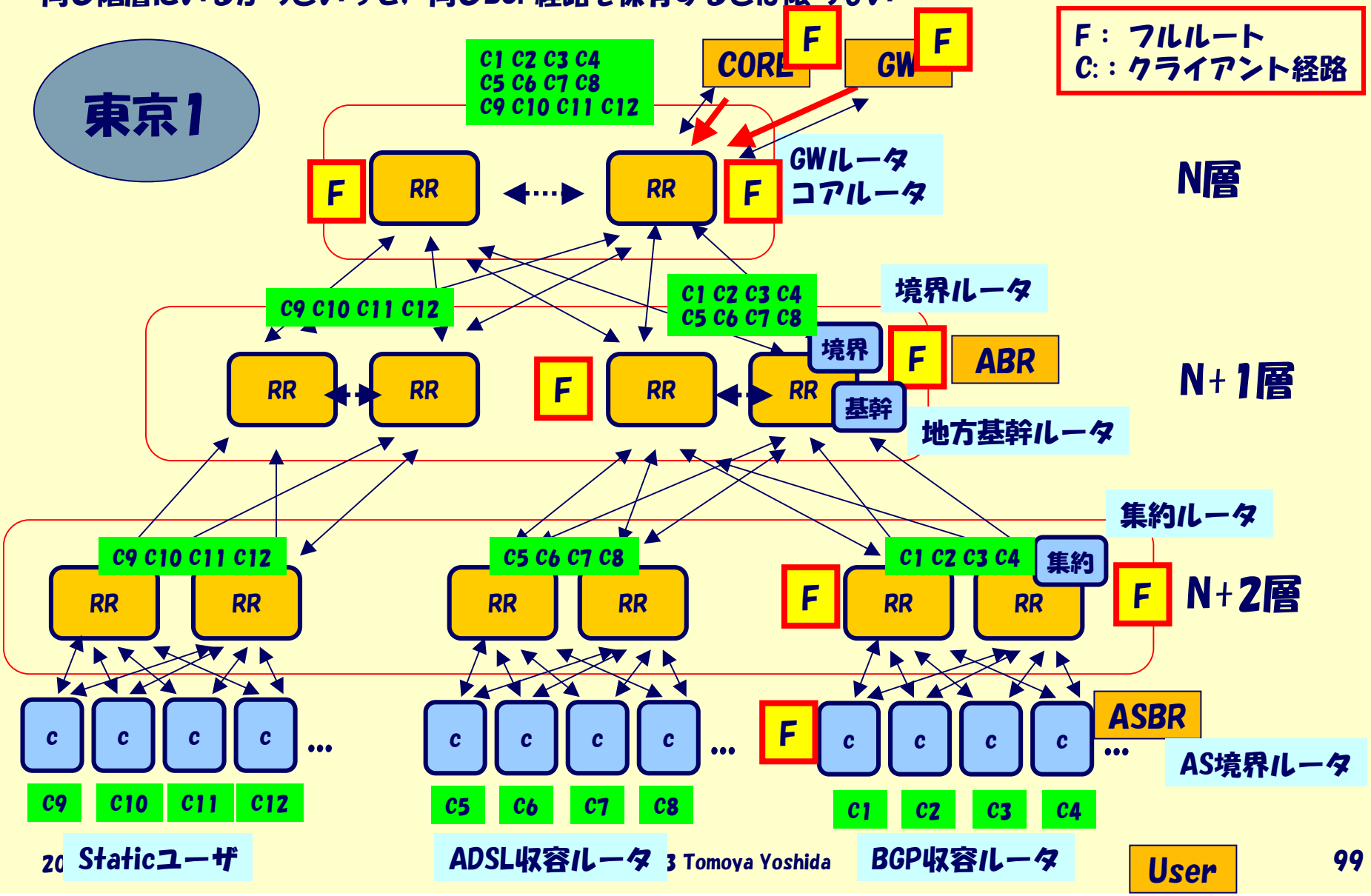
ASBRでユーザアドレスをstaticで記述．それを上位にBGPで再配信．BGPの場合には，no-exportをつけて，GWルータから外にでていかないようにしている．内部のiBGPではsend-communityを動作させ，no-exportのCommunity情報がついたは内部でのみ伝播する

帰りのルーティング

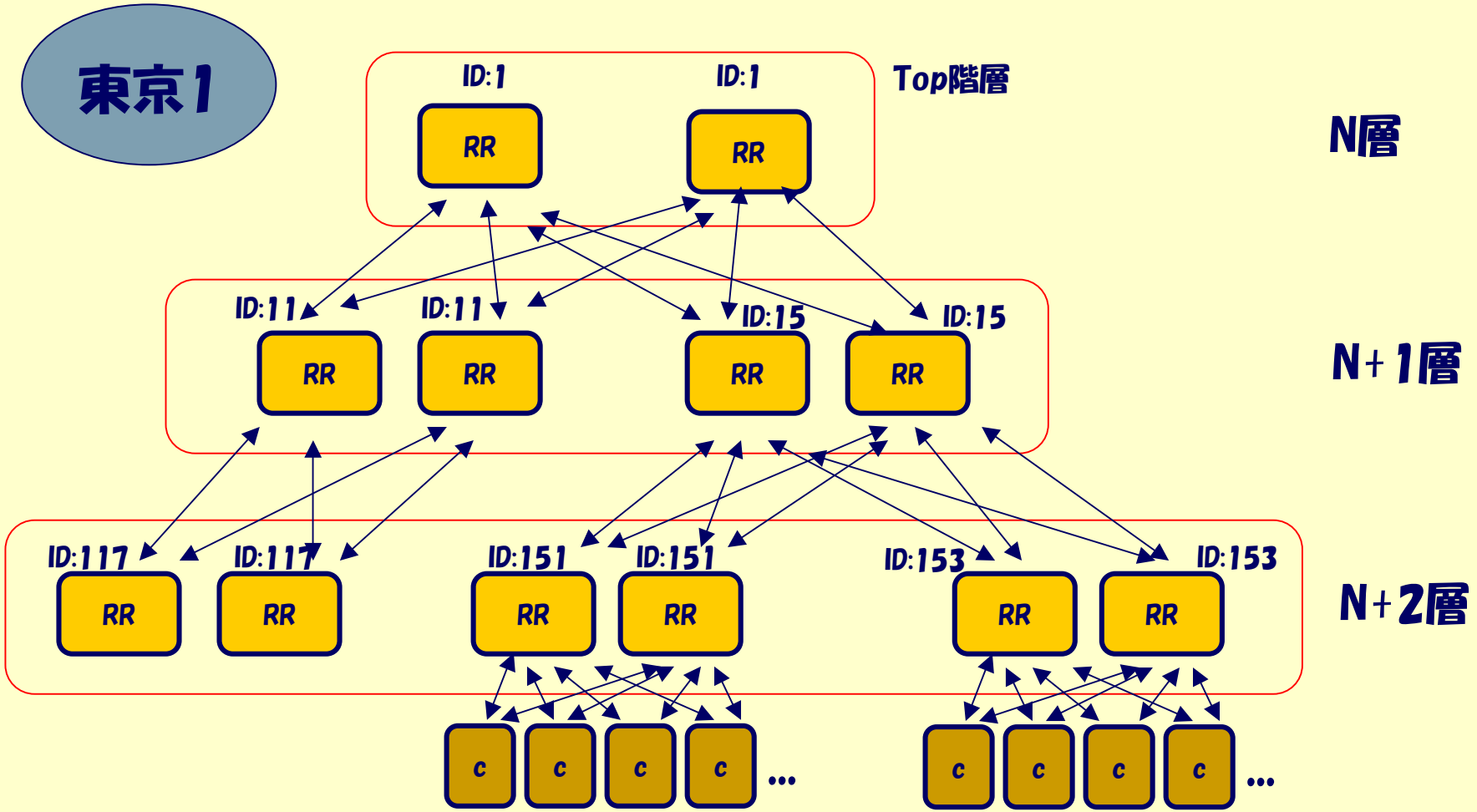

インターネット
細かいiBGP経路エリア
デフォルトルートに従ってコアルータへルーティングすればOK
eBGP
コアルータに到達した段階では，全ての細かい下部の経路を知っている必要があるので，全BGP情報を保有する
GW
GW
GW
外部接続部（GW）
デフォルト配信
iBGP
OSPF
CORE
CORE
バックボーン
ABRでは，自分の配下のBGP経路を保有していればOK
ABR
ABR
ABR
中継・アクセス部
ASBR
ASBR
StaticをBGPへ再配信
eBGP
自身のStaticルートに従ってルーティングされる
AS
R
R
エンドユーザ

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

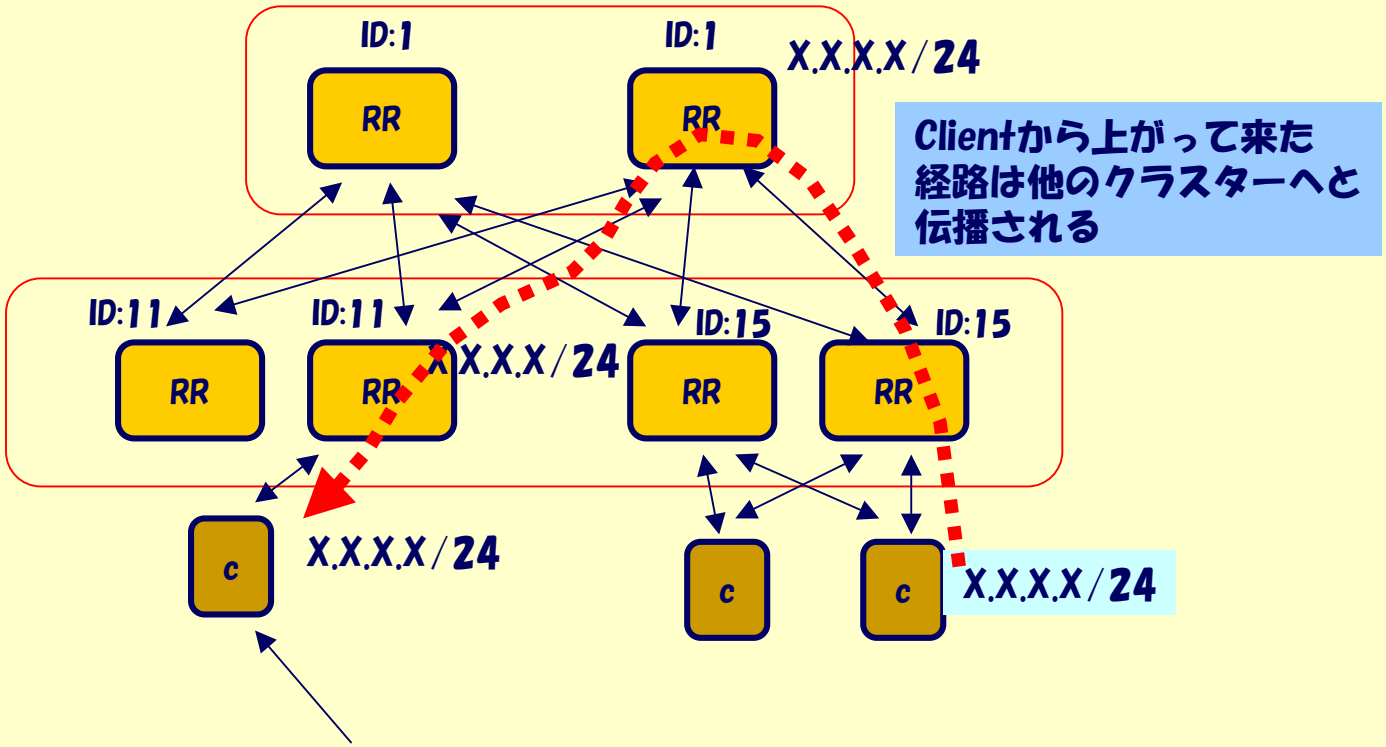


リフレクタ階層構造(東京1の例)



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かもしれない

他のクラスターから経路が伝播される

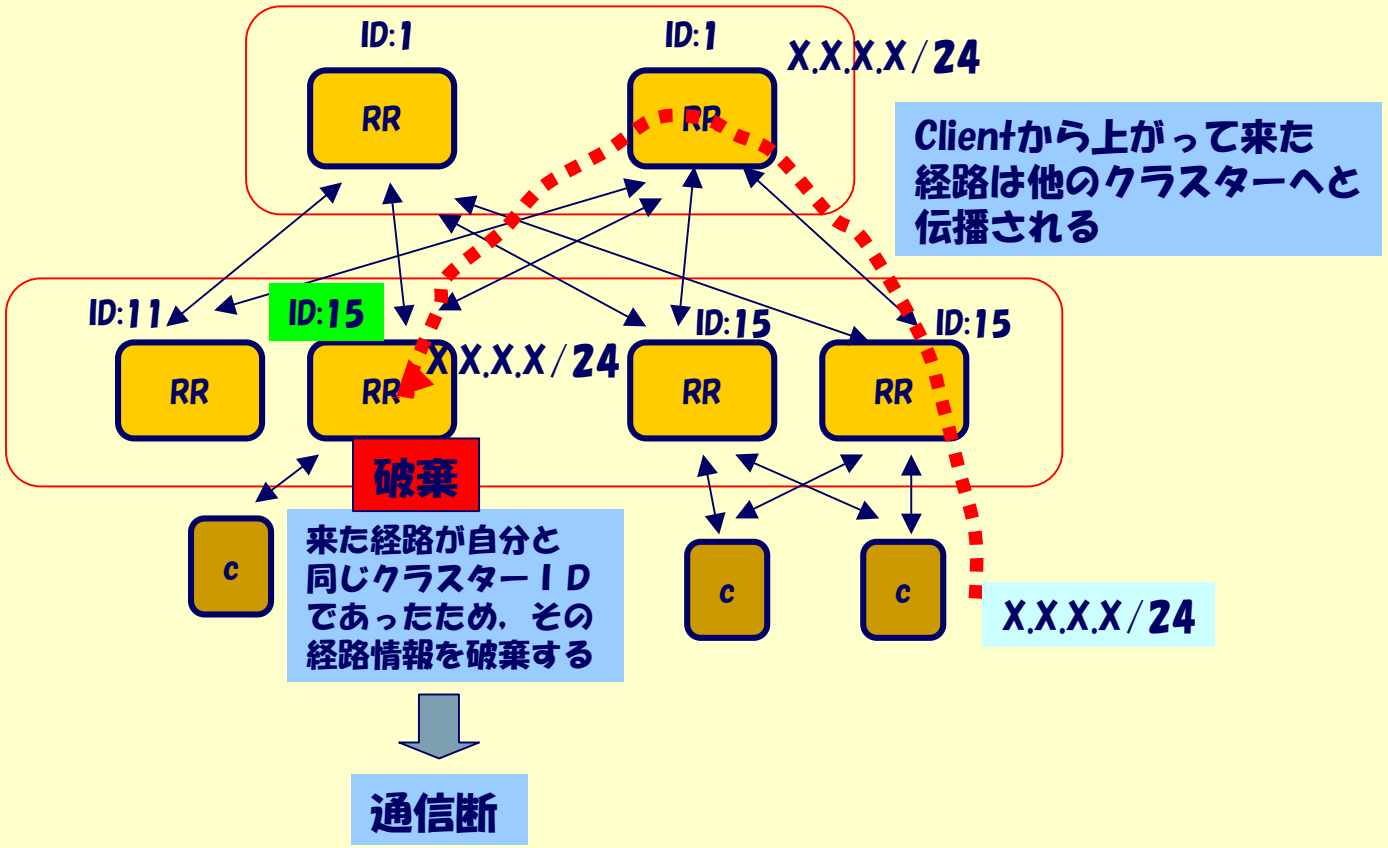


Cluster list: 0.0.0.11 , 0.0.0.1 , 0.0.0.15

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を左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）

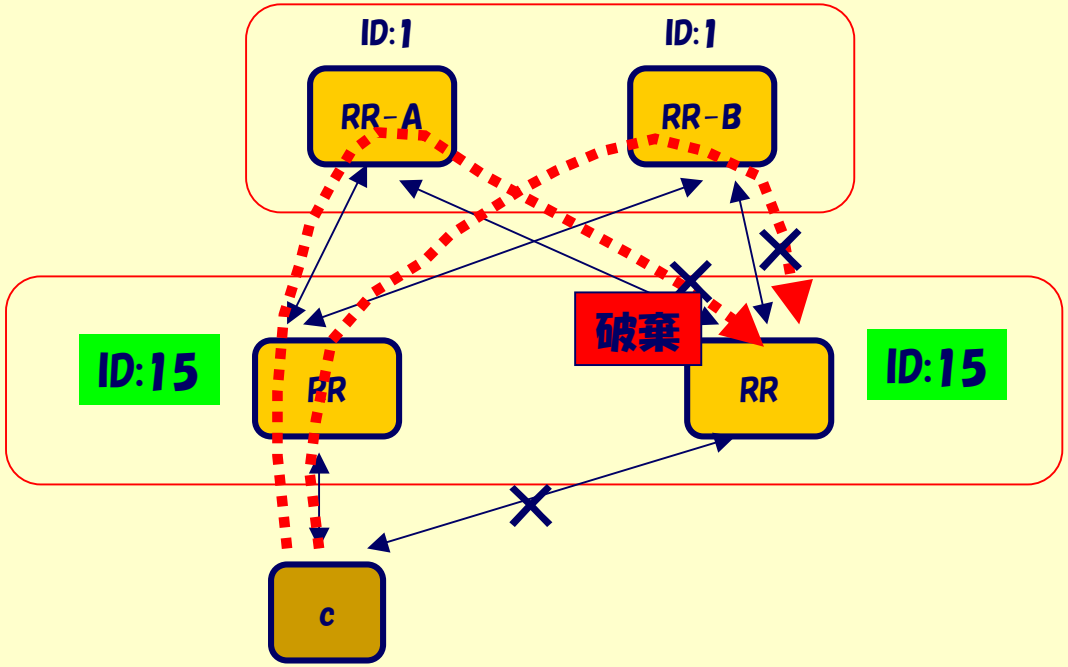
AS_PATH : 4713 2914 701

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，ループを防ぐために破棄してしまう（AS_PATHのループディテクションと原理は一緒）

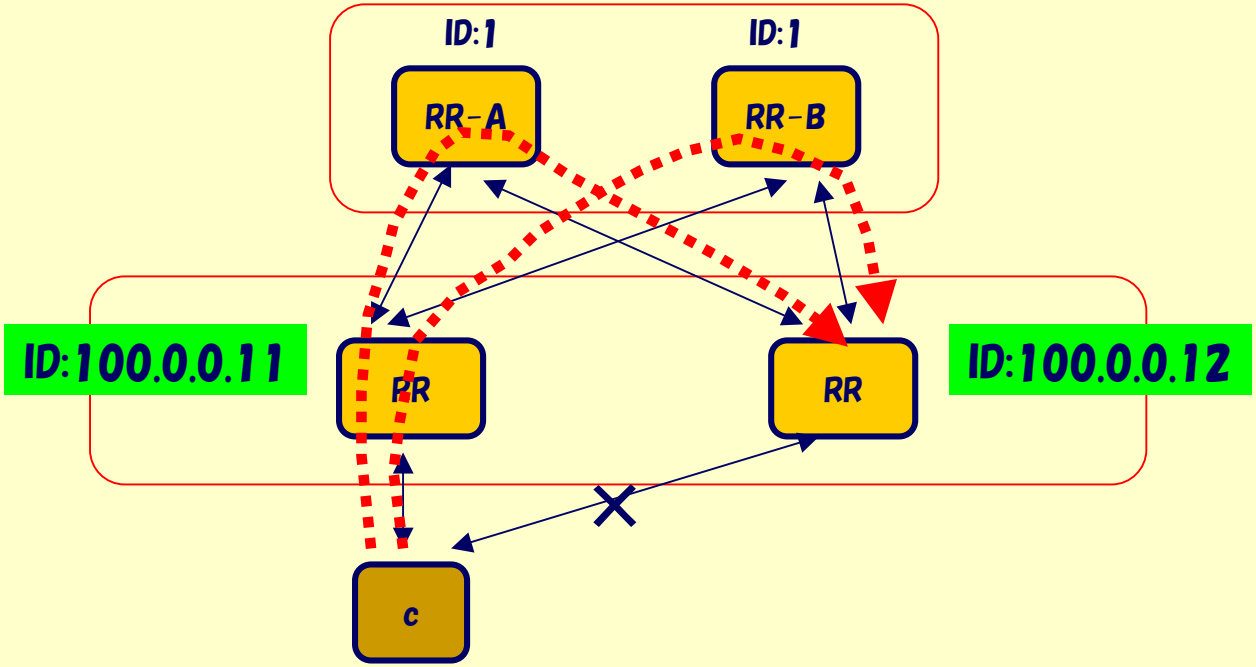
クライアントとのピアが切れた場合(同一ID)



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



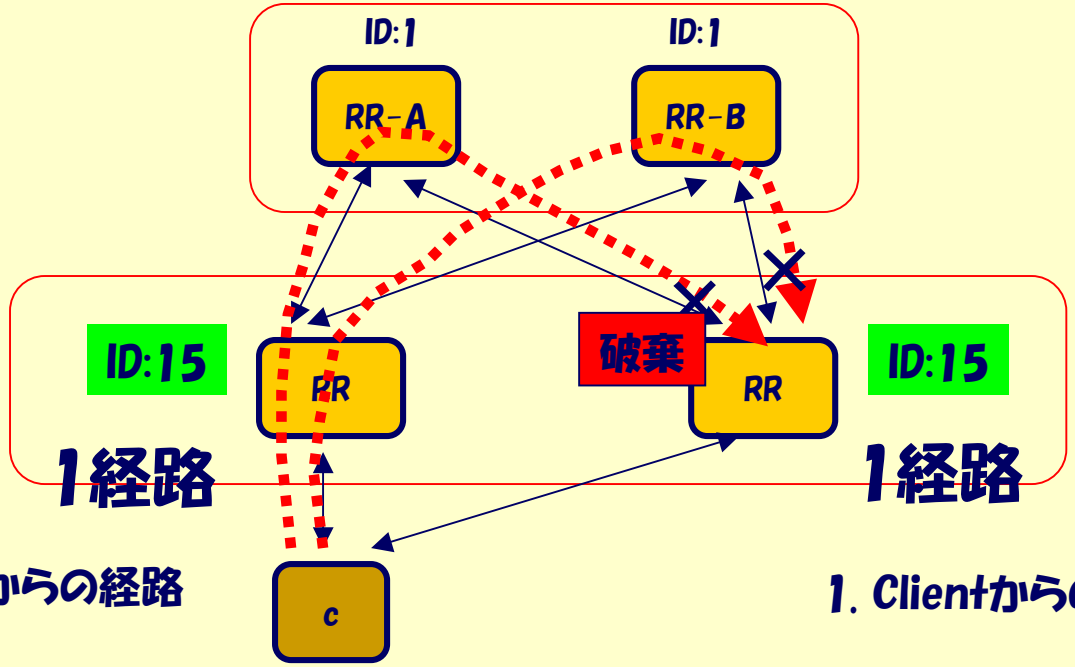
別IDの場合には，クライアントの片方のピアがきれても，
上位リフレクタから経路が配信される．（通常状態においても配信される）

RRがパケットフォワーディングもやっている場合には，この方法がいいだろう

ツリーが増えるので，適応個所には注意したいが，大きな問題はないだろう
BGP経路の伝播が，同一IDとは異なる点にも注意したい

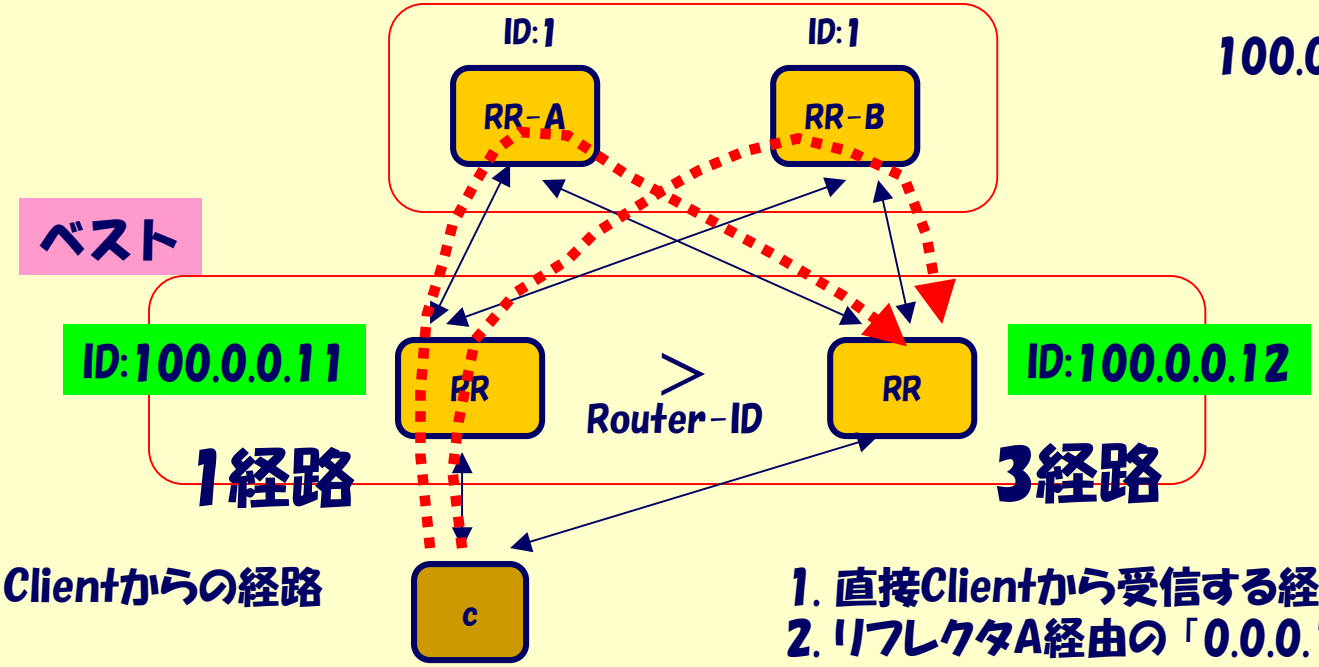
経路の見え方(同一IDの場合)

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



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)

iBGP設計のポイント

- リフレクタの階層化
  - COREを中心とした物理的な階層と同等な階層化が理想的
    - 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を付与する場合に，わかりやすい数字かループバックアドレスに設計する
  - 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，ちゃんと押さえておく必要がある

- サービスごとにクラスター化をし，各クラスターごとに配信経路やルーティング方式を検討する(フォワーディングトポロジーに追従)
  - BGPユーザのクラスタ
    - 当然BGPで経路を配信
    - 他のクラスタの細かい経路まではいらない
  - ADSL専用クラスタ
    - 上位には，BGPでクライアント経路を配信．ルーティングはデフォルトルートに従えばよいので．フルルートを保有する必要はない　など

その他

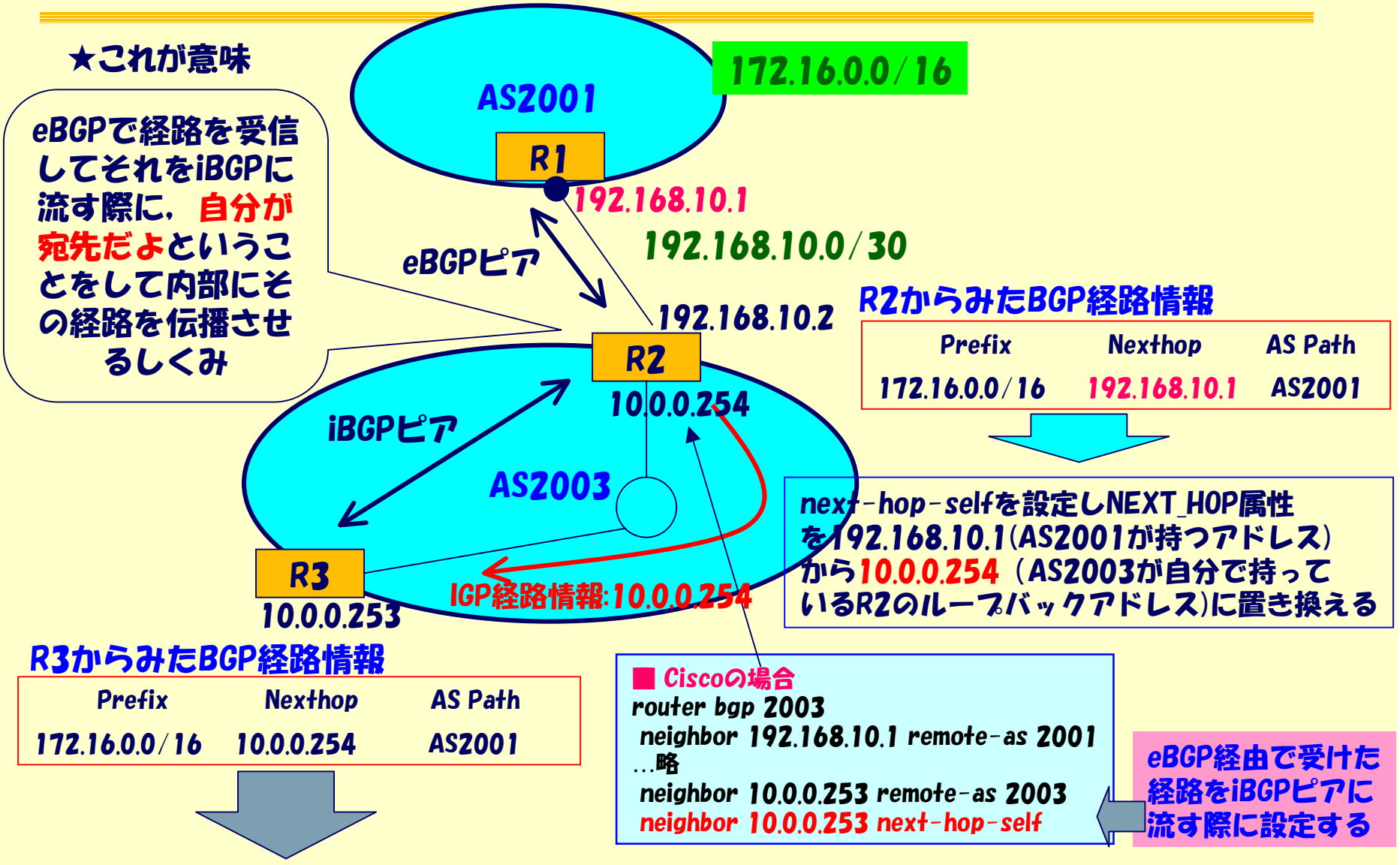
- next-hop-self
- リカーシブルックアップ
- eBGPマルチホップ/マルチパス
- CIDRの広報
- ルートダンプニング

BGPのnext-hopの解決方法

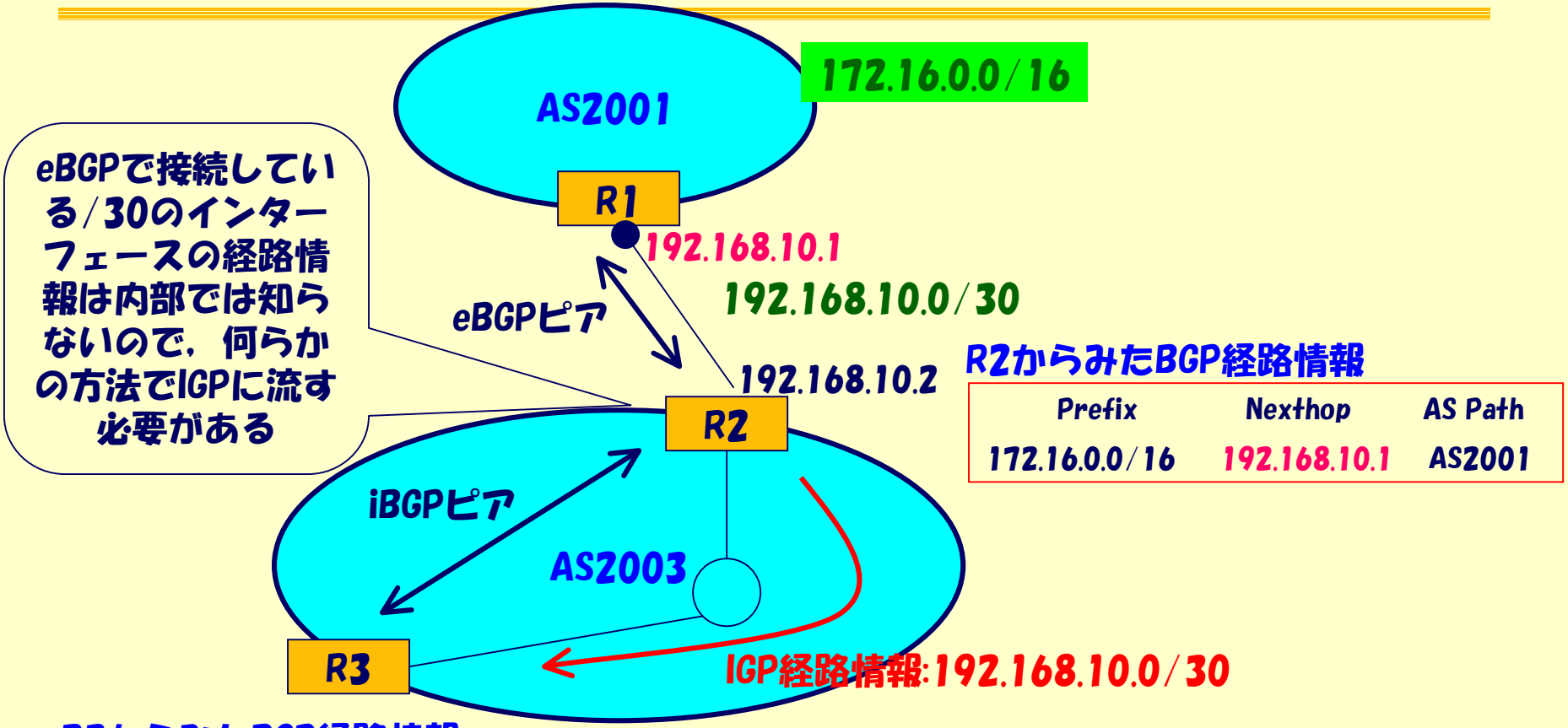
- BGPでは，相手から受信した経路のnext-hopに到達性がなければ，その経路は無効とする(NEXT_HOP属性)
  - eBGPの場合には，受信時に破棄

- 外部経路のNEXT_HOPの解決方法には，2つの方法がある
  - 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，IGPに流す
    - 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

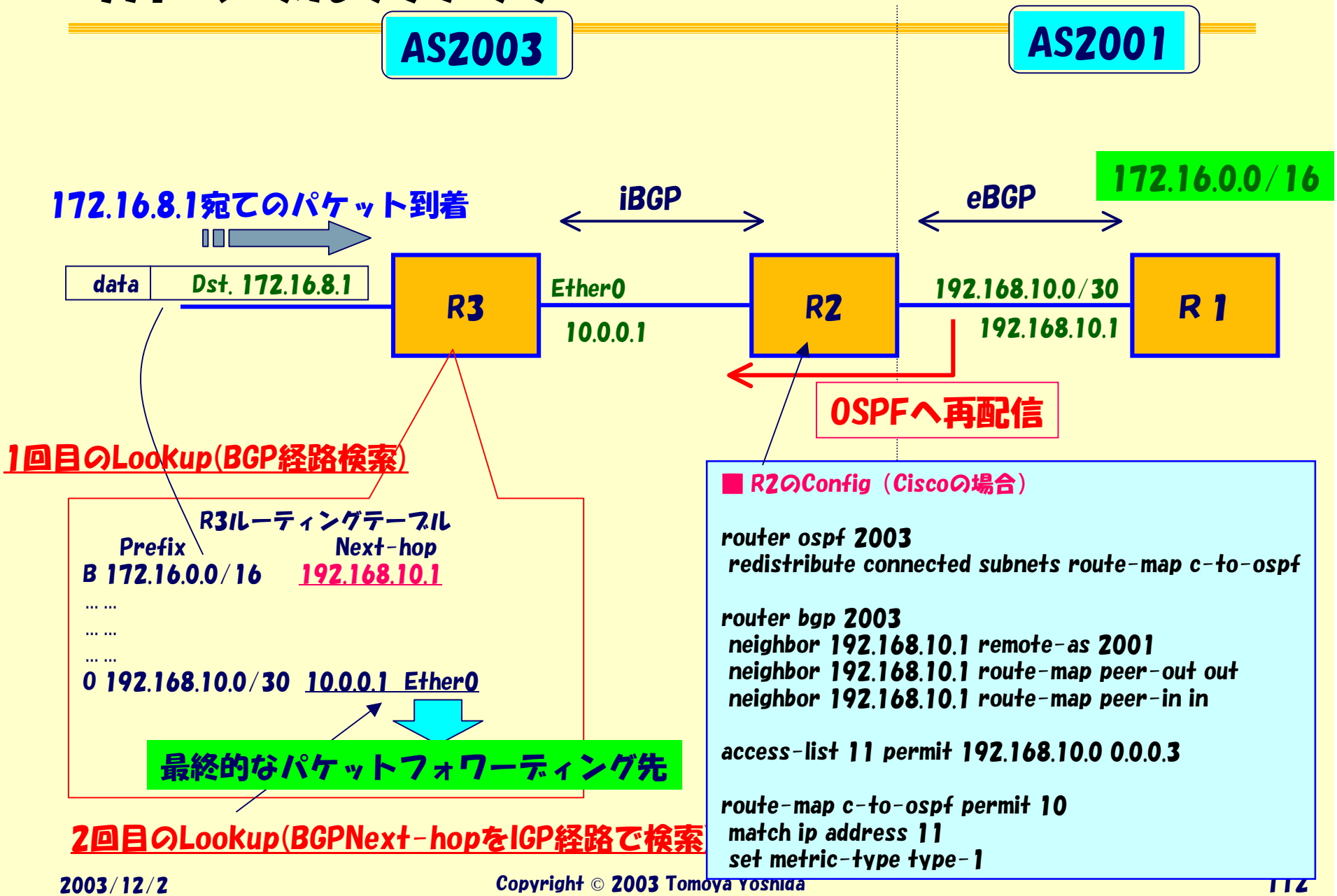
R3からみたBGP経路情報

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、AS2003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



eBGPマルチホップによるロードバランス

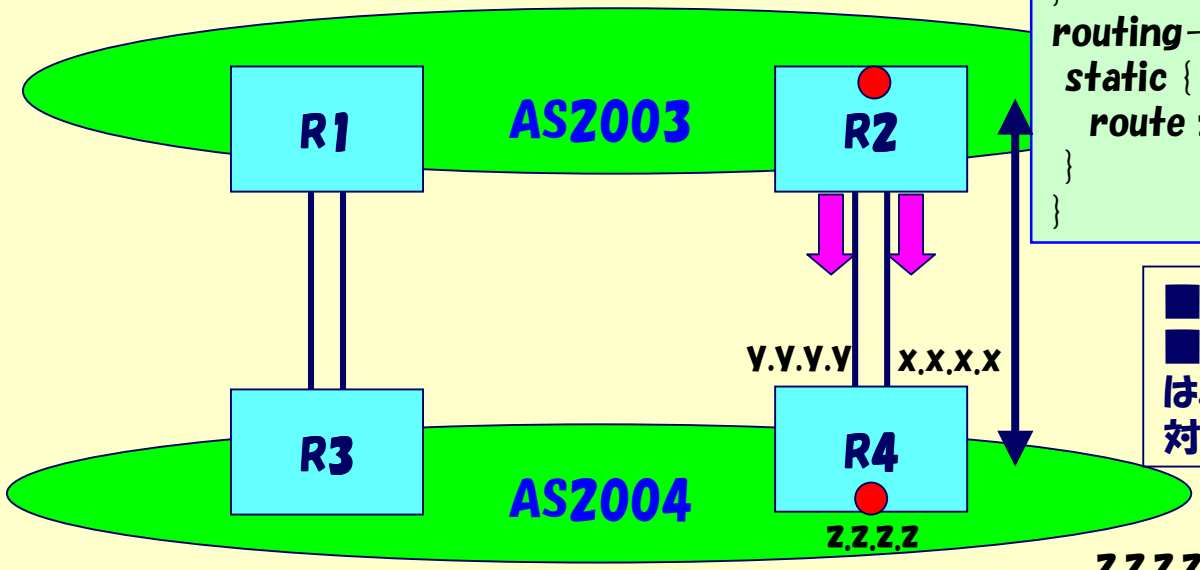
同一ルータで外部と複数本でeBGPピアをはる場合，eBGPマルチホップによりロードバランスが可能

■ Ciscoの場合（R2）

```
router bgp 2003
 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 ];
 }
}
```



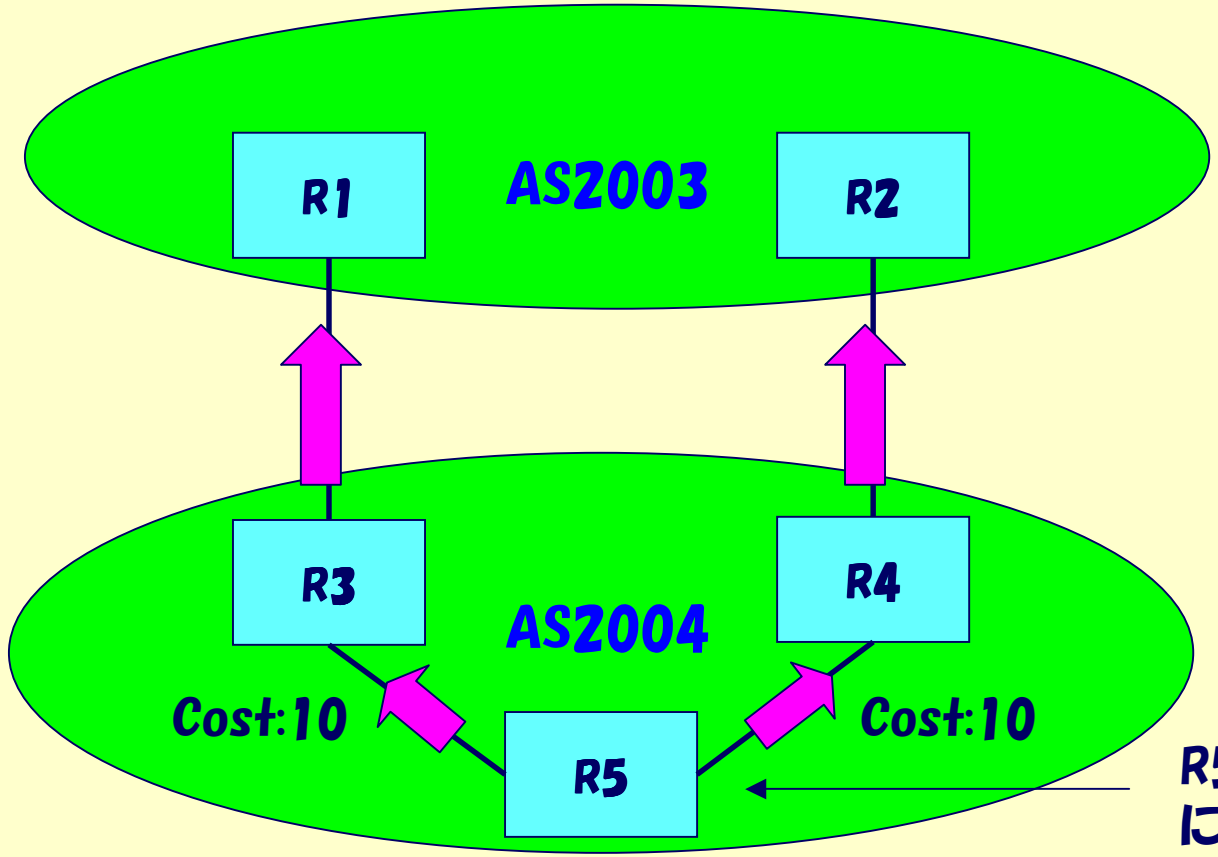
■ループバックアドレスで互いにピアをはる
■相手のループバックに対するルーティングは，Static Route を物理インターフェースに対して設定することにより解決

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 2003
 maxiumu-path ibgp 2
```

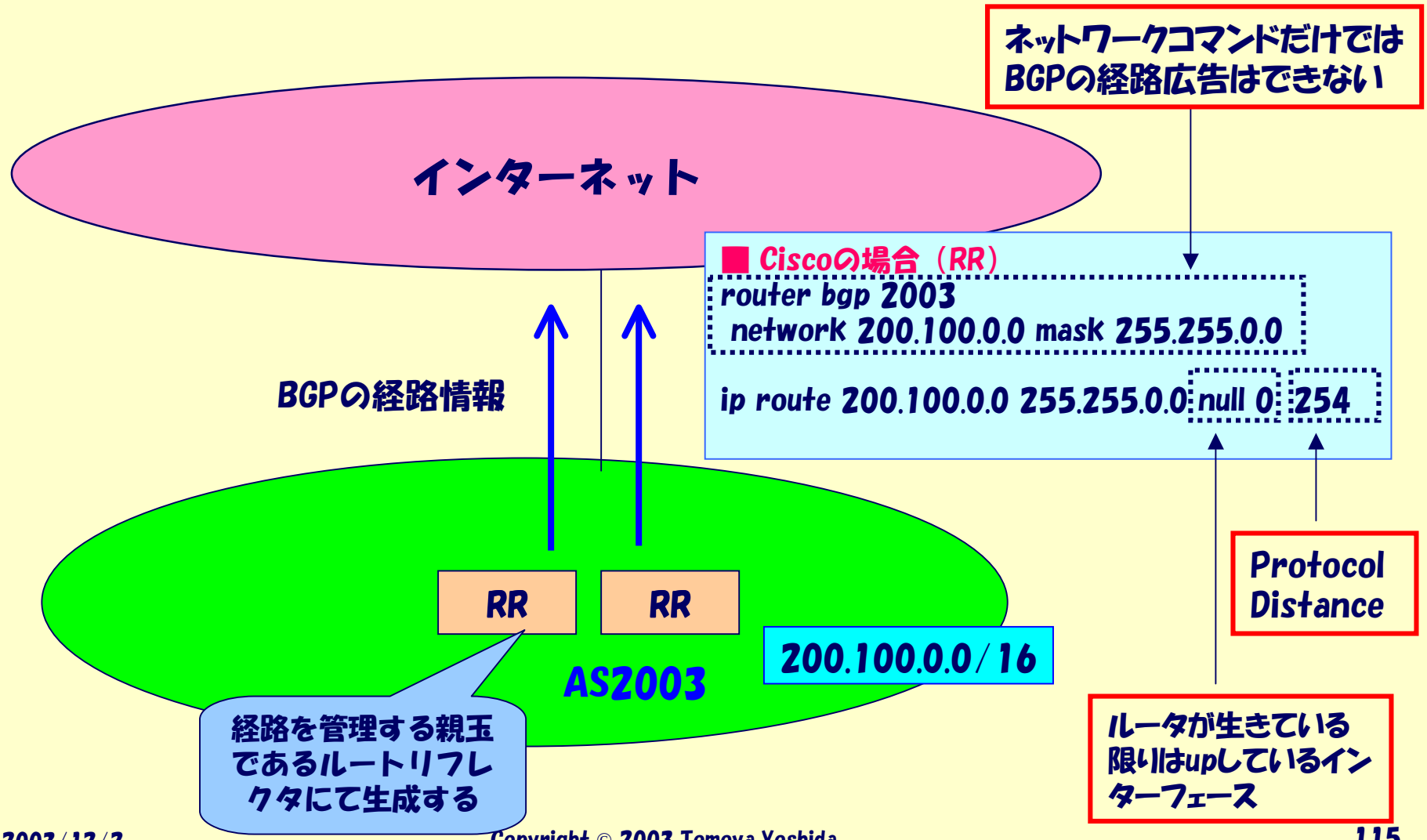
■ Juniperの場合（R5）

```
protocols {
  bgp {
   group iBGP {
      neighbor x.x.x.x {
               multipath;
```

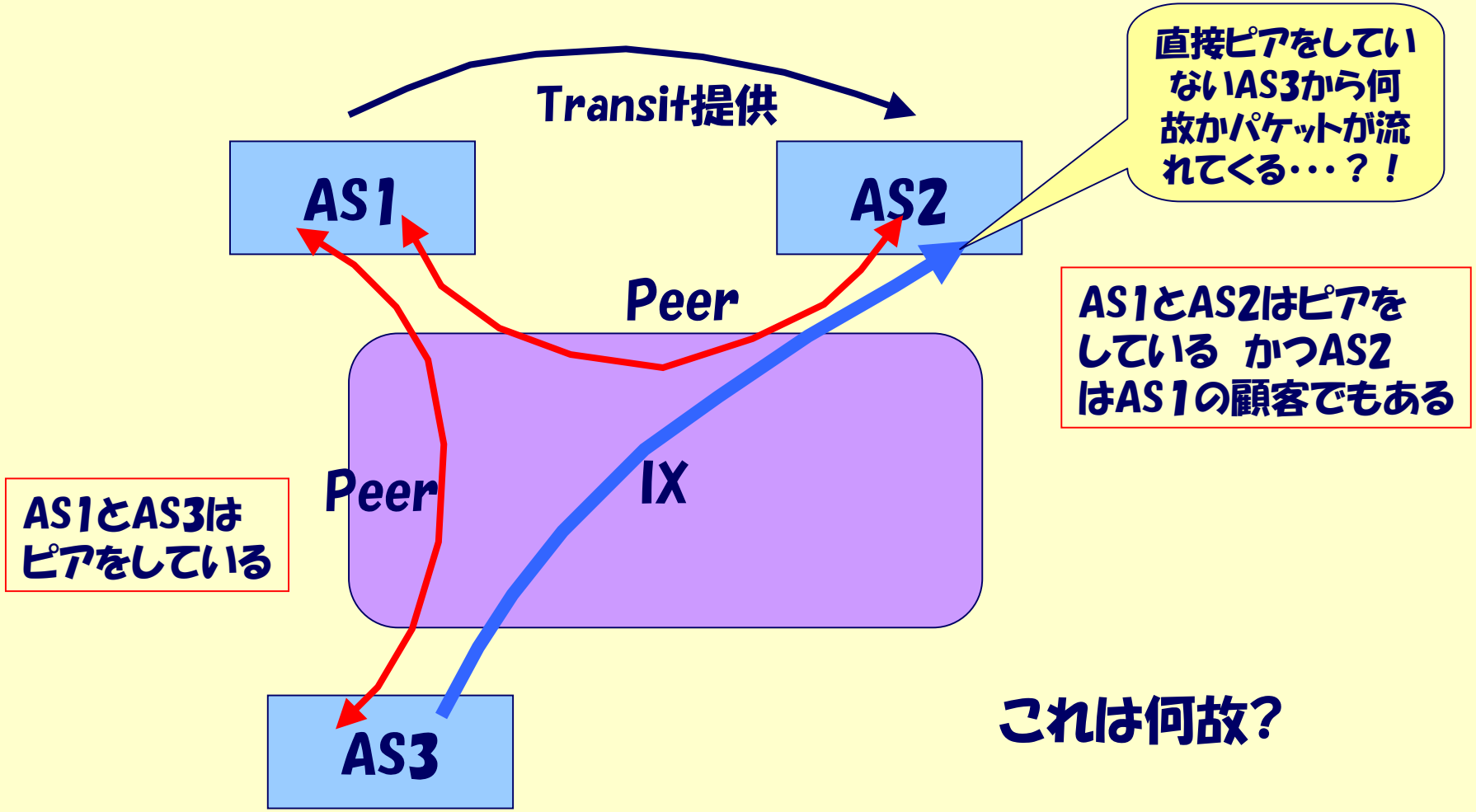
R5にて，マルチパス機能を有効にする．eBGPに対しても可能

PAアドレス(CIDR経路)の広告

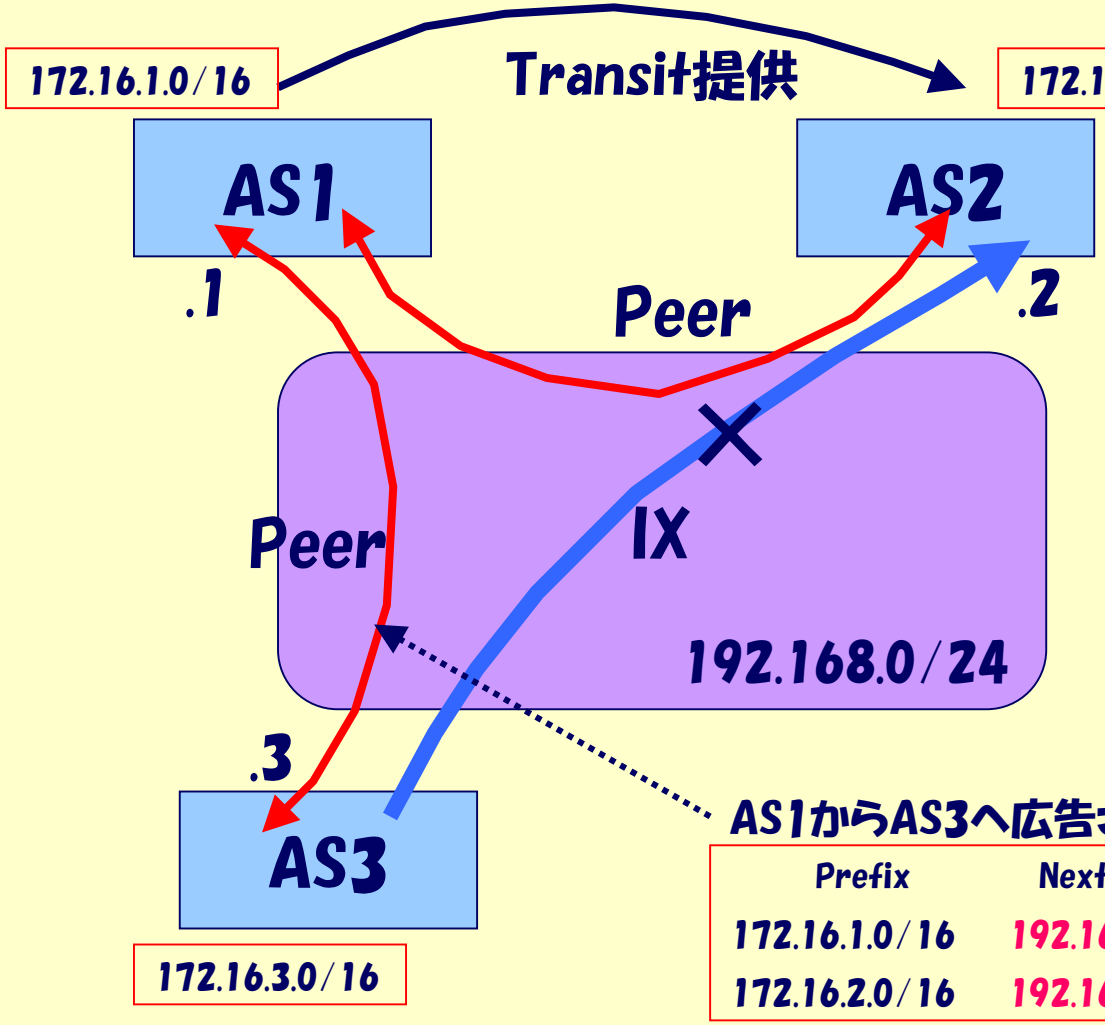
・CIDR経路は「安定して」インターネットに広報されていなくてはならない
・BGPで経路広告する際のIGPは，「static null0」にて行う！



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

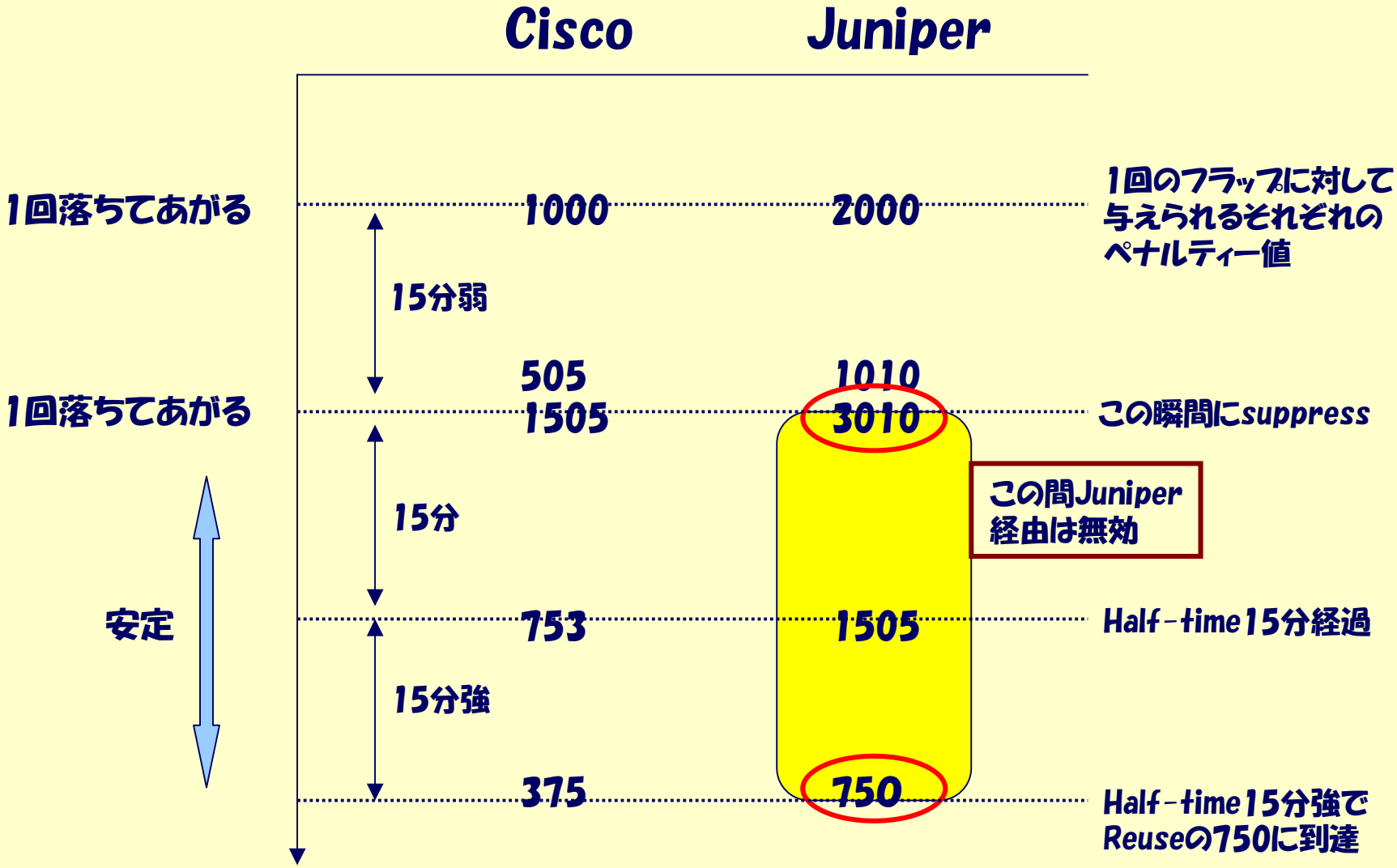
＜Cisco＞	
Penalty	1000/1Flap
＜Juniper＞	
＊Route is withdrawn	1000
＊Route is readvertised	1000
＊Route's path attributes change	500

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



マルチベンダ環境における設計

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
  - など・・・
- ある程度は検証を行って確認しましょう
  - 実網でわかった場合には，その都度検討

BGP Hold-time

- 実装が若干異なる
  - 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）

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
  - ただし，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
  - 対向のピアに対して，「send-community」と記述しないと，ちゃんとコミュニティを伝播してくれない
    - 例えば，no-exportなどの経路を内部で利用していると，上流向けに対して「send-community」がはずれてしまった場合には，外部にもれてしまう
- 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 → メモリを消費する
  - Ciscoの場合など，「soft-reconfiguration inbound」でキャッシュ

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

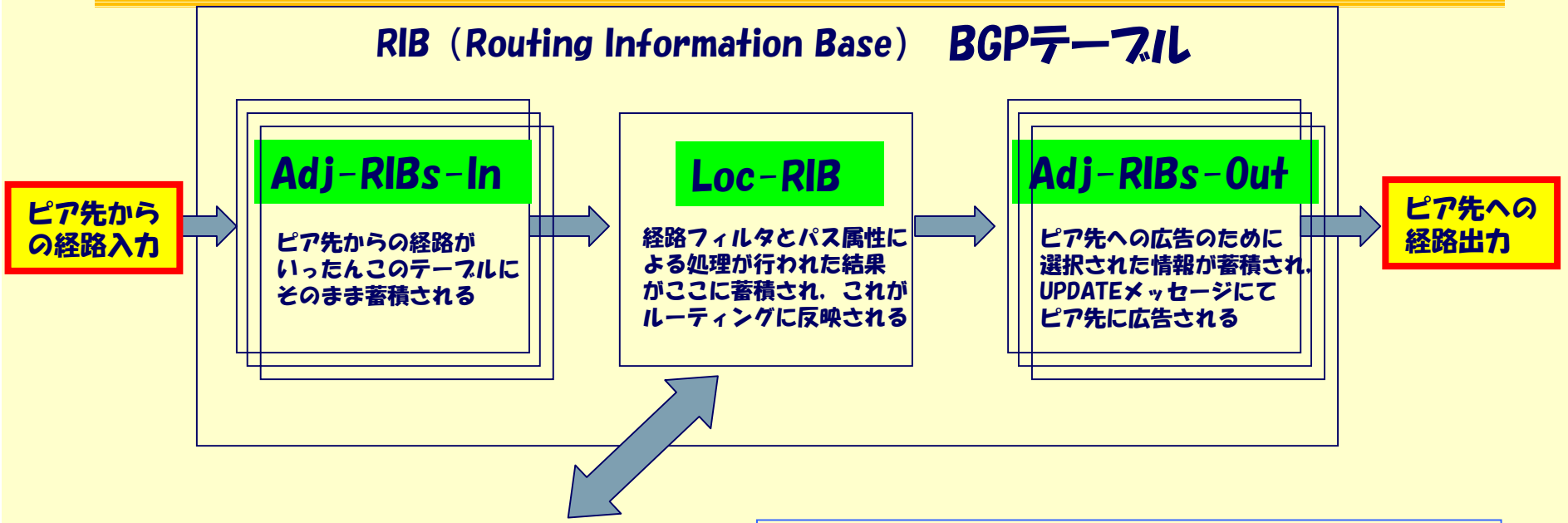
経路管理のされ方(1)

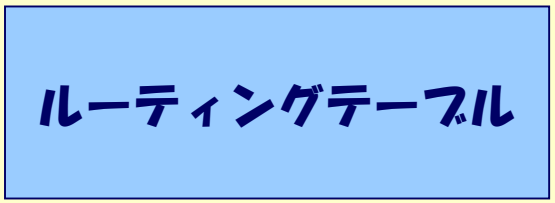
- ルーティングテーブルのみ：Juniper，RS
  - OSPFもBGPも全て1つのルーティングテーブルで管理されている
  - ルーティングテーブル上でベストではないと，BGPにて配信されない
    - 例えばJuniperでは，「advertise-inactive」というコマンドで，OSPFなどBGP以外のプロトコルがベストとなっていても，BGP上で最もベストな経路が配信可能となる
  - BGP以外の経路が配信されてしまう可能性があるので注意
    - Outのpolicy変更は，IPルーティングテーブル全体に適応される
      - match protocol ospf 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






```
0.0.0.0/0          *[OSPF] 00:00:51, metric 110, tag 0
4.0.0.0/8          *[BGP] MED 100, localpref 200, AS path: 2914 3356 I
6.1.0.0/16         *[BGP] MED 100, localpref 90, AS path: 2914 668 7170 1455 I
…
150.22.0.0/16      *[OSPF] metric 36, tag 0
…
```

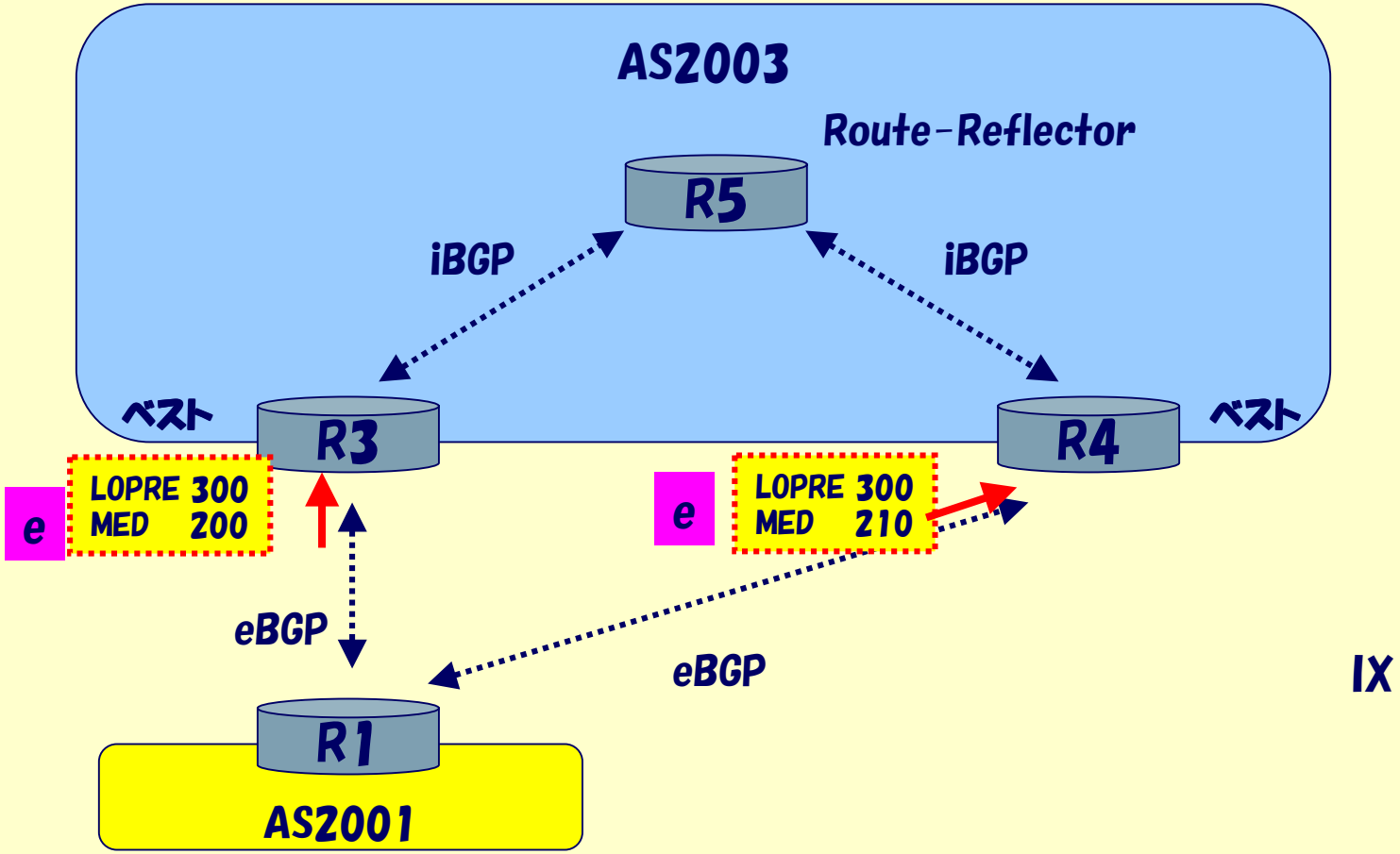

```
Destination        Interface
0.0.0.0/0          so-0/1/0.0
4.0.0.0/8          so-0/0/0.0
6.1.0.0/16         so-0/0/0.0
…
150.22.0.0/16 so-0/1/0.0
…
```

MEDについて

- MED(MULTI_EXIT_DISC)のおさらい
  - 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

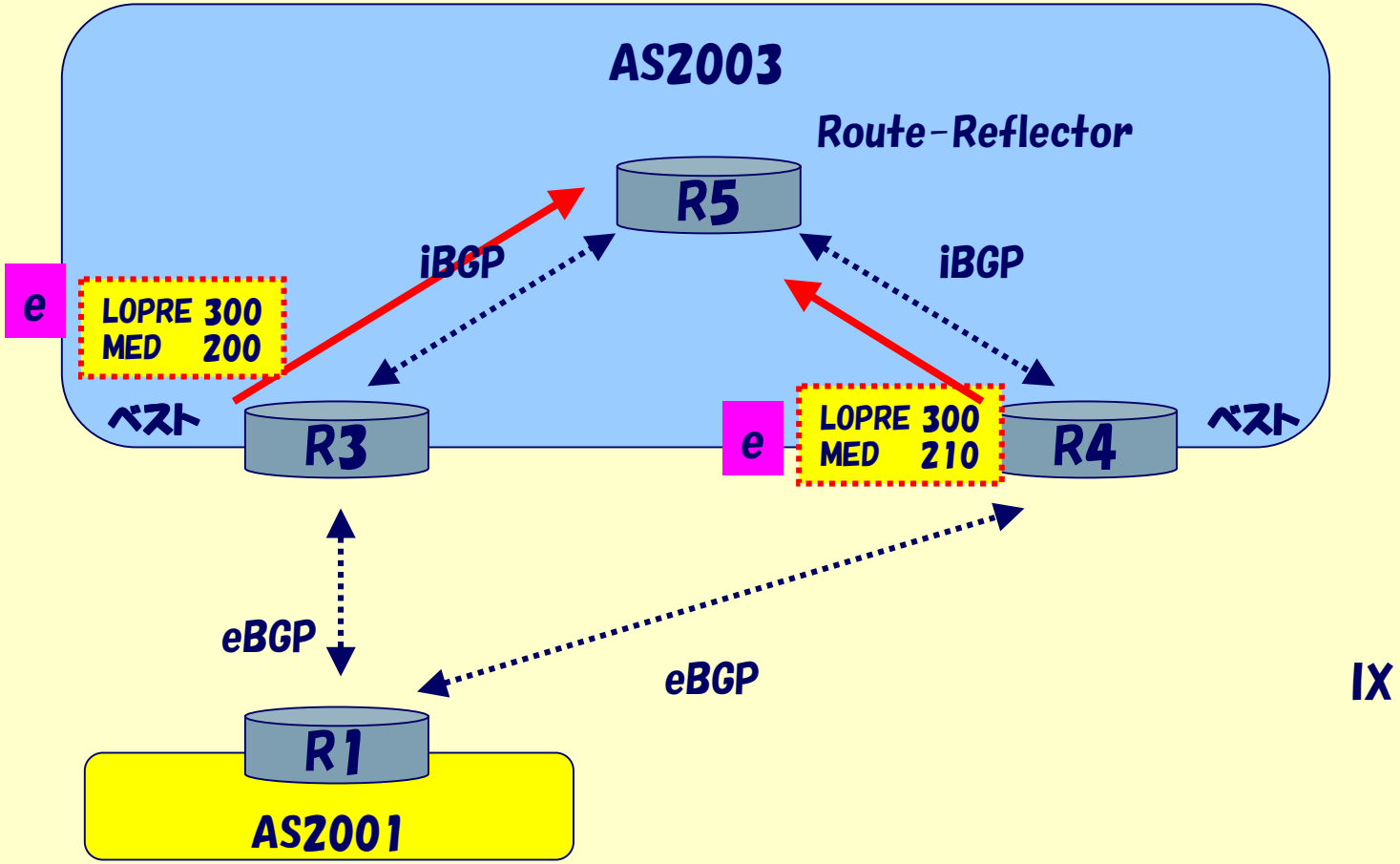
BGP経路比較(MED編)[1]

AS2001から2つのBGPピア経由で経路を受信



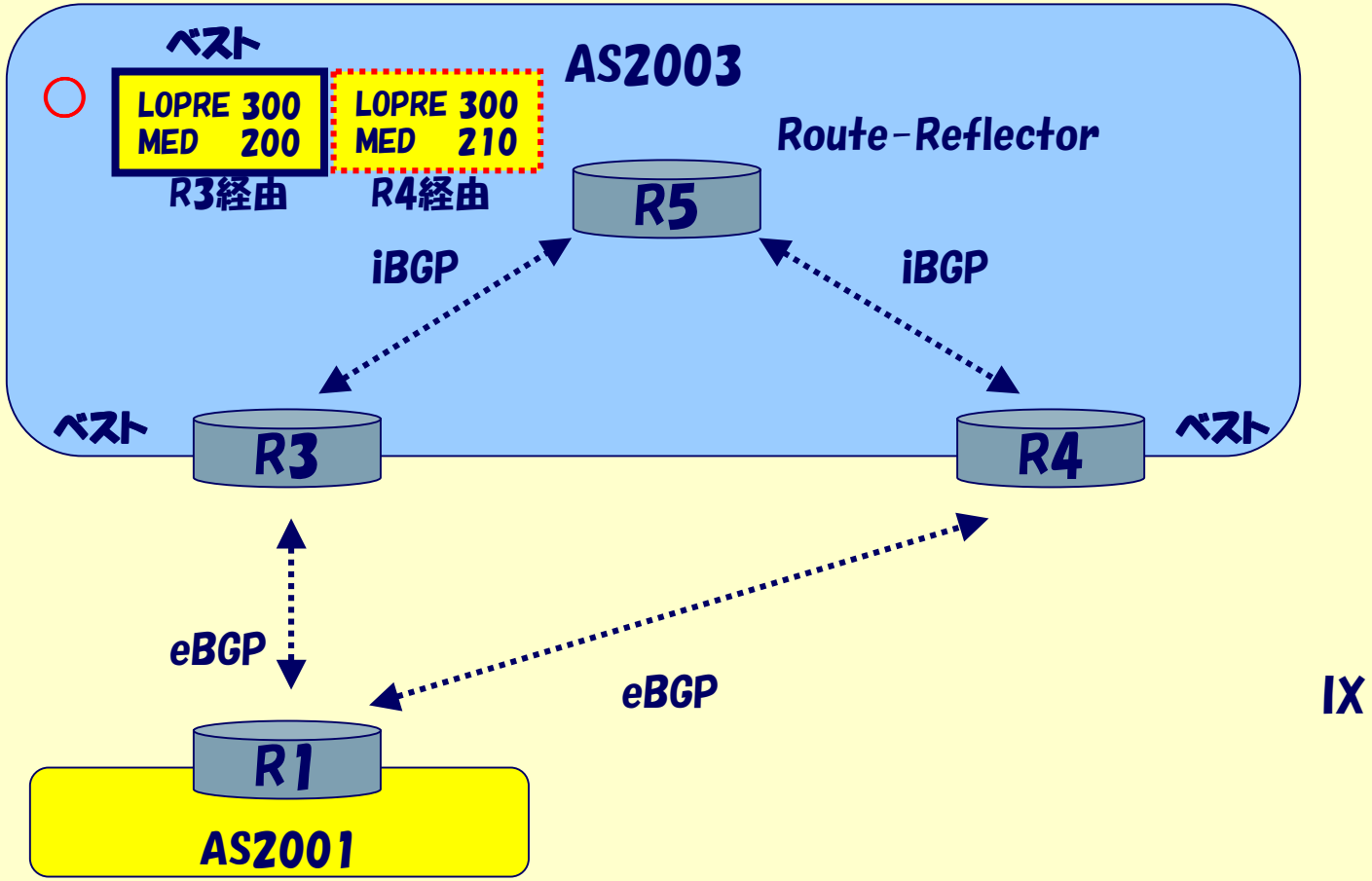
BGP経路比較(MED編)[2]

R3，R4から上位のリフレクタ（R5)へその経路を伝達する



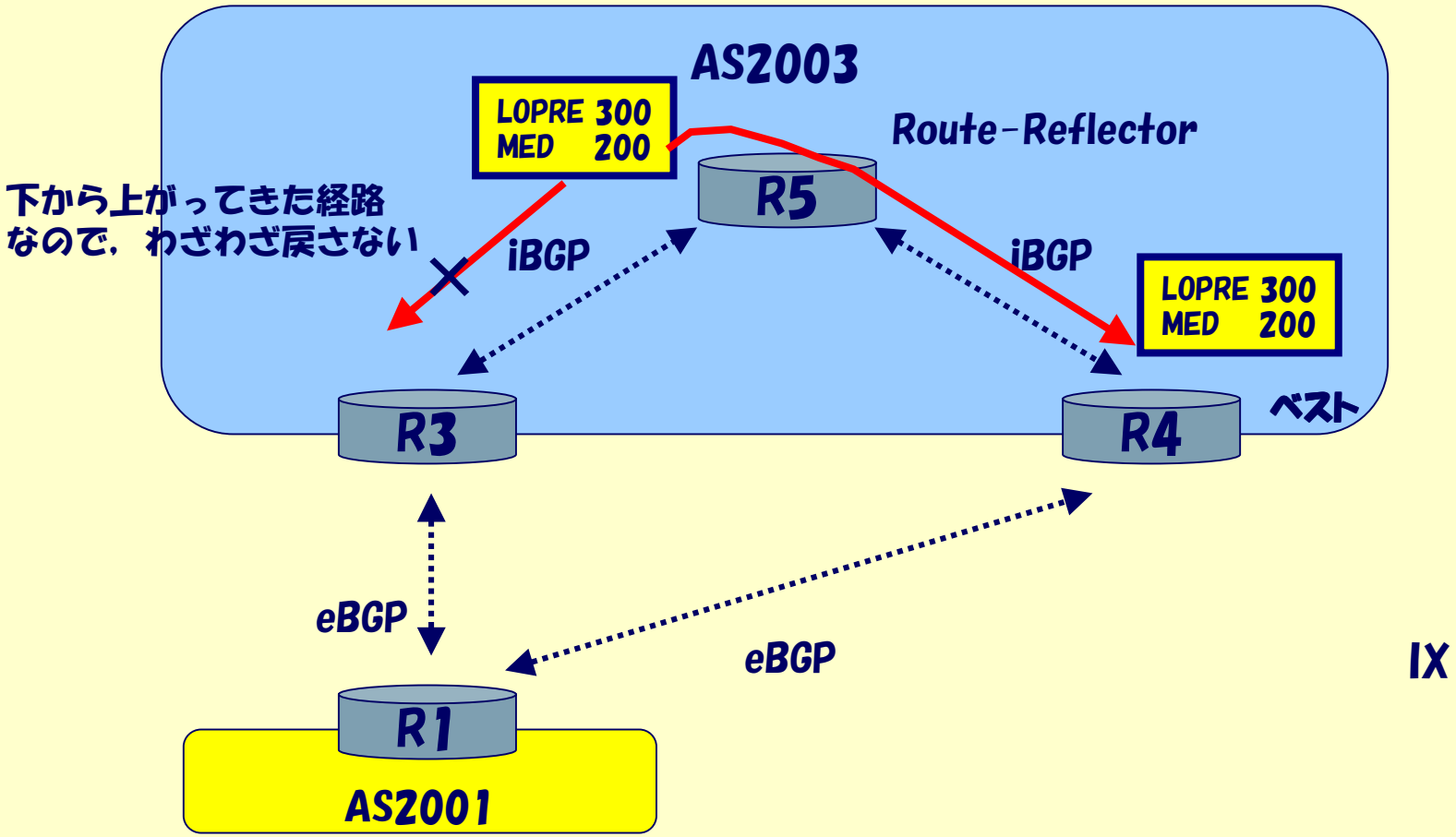
BGP経路比較(MED編)[3]

同一ASの経路なので，MEDの小さいほうをR5ではベストパスに選択



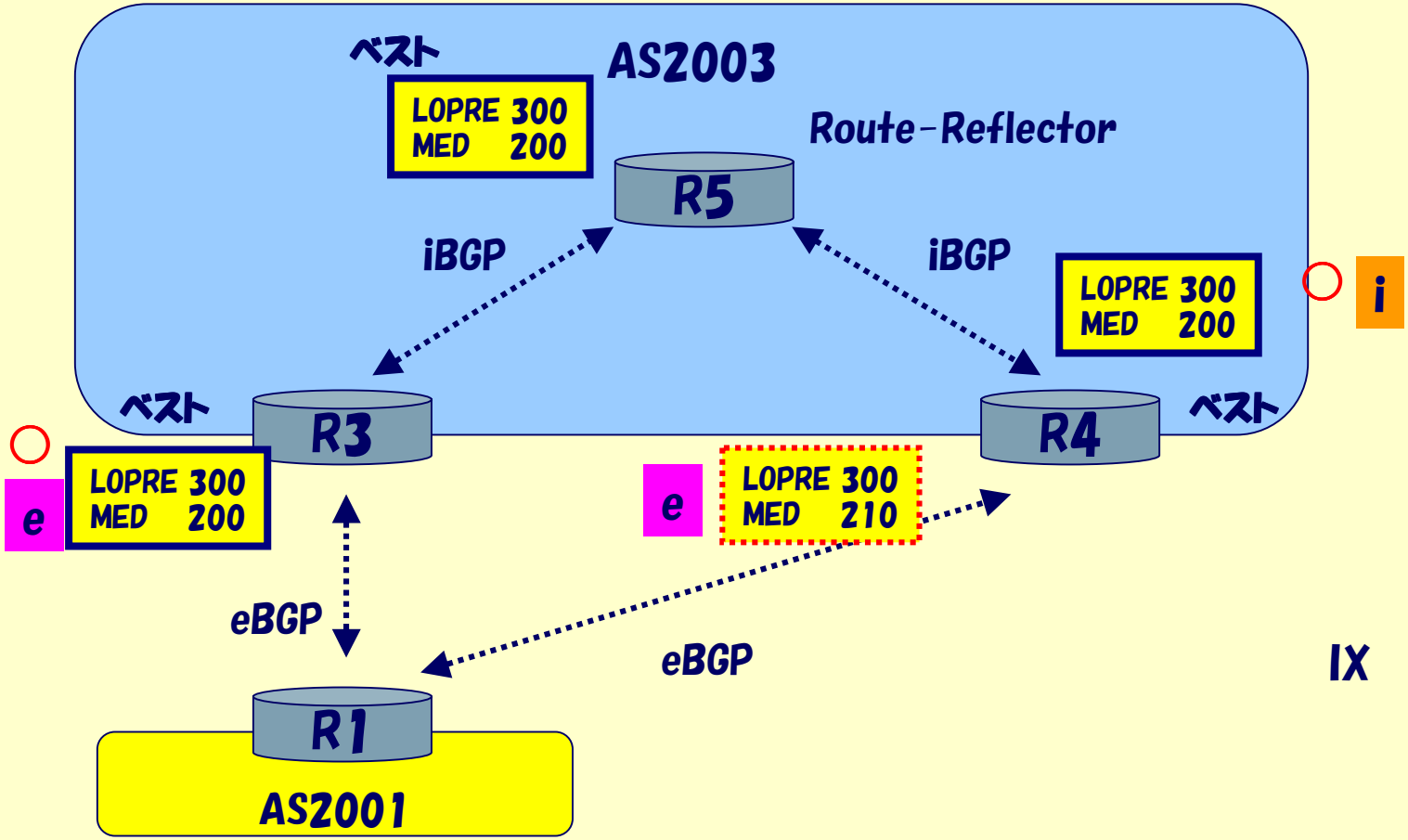
BGP経路比較(MED編)[4]

R3からのベスト経路をR4へ配信



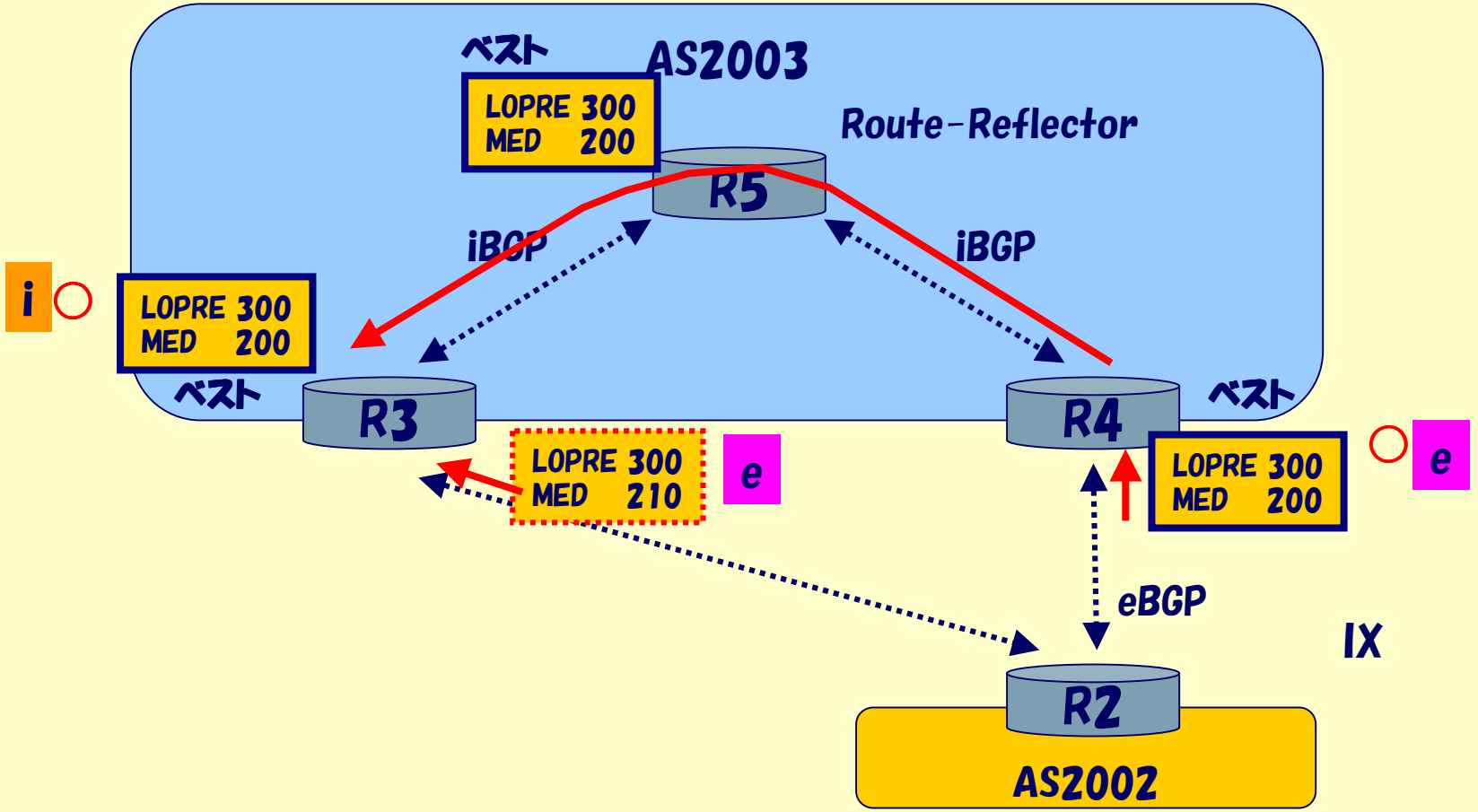
BGP経路比較(MED編)[5]

最終的には，R3経由の経路が伝播して落ち着く



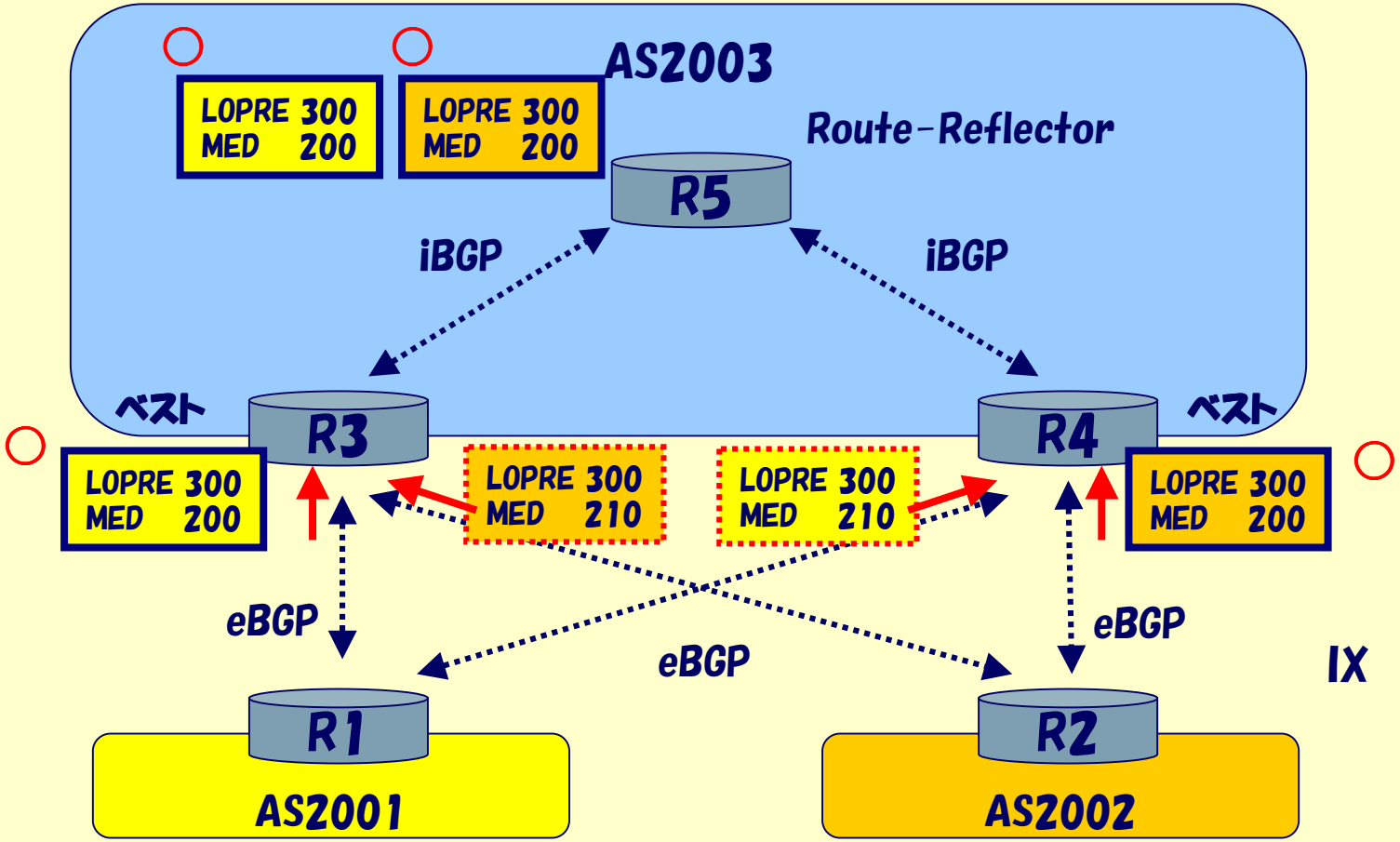
BGP経路比較(MED編)[6]

同様にAS2002の例：この場合は，R4経由の経路がベストになって落ち着く



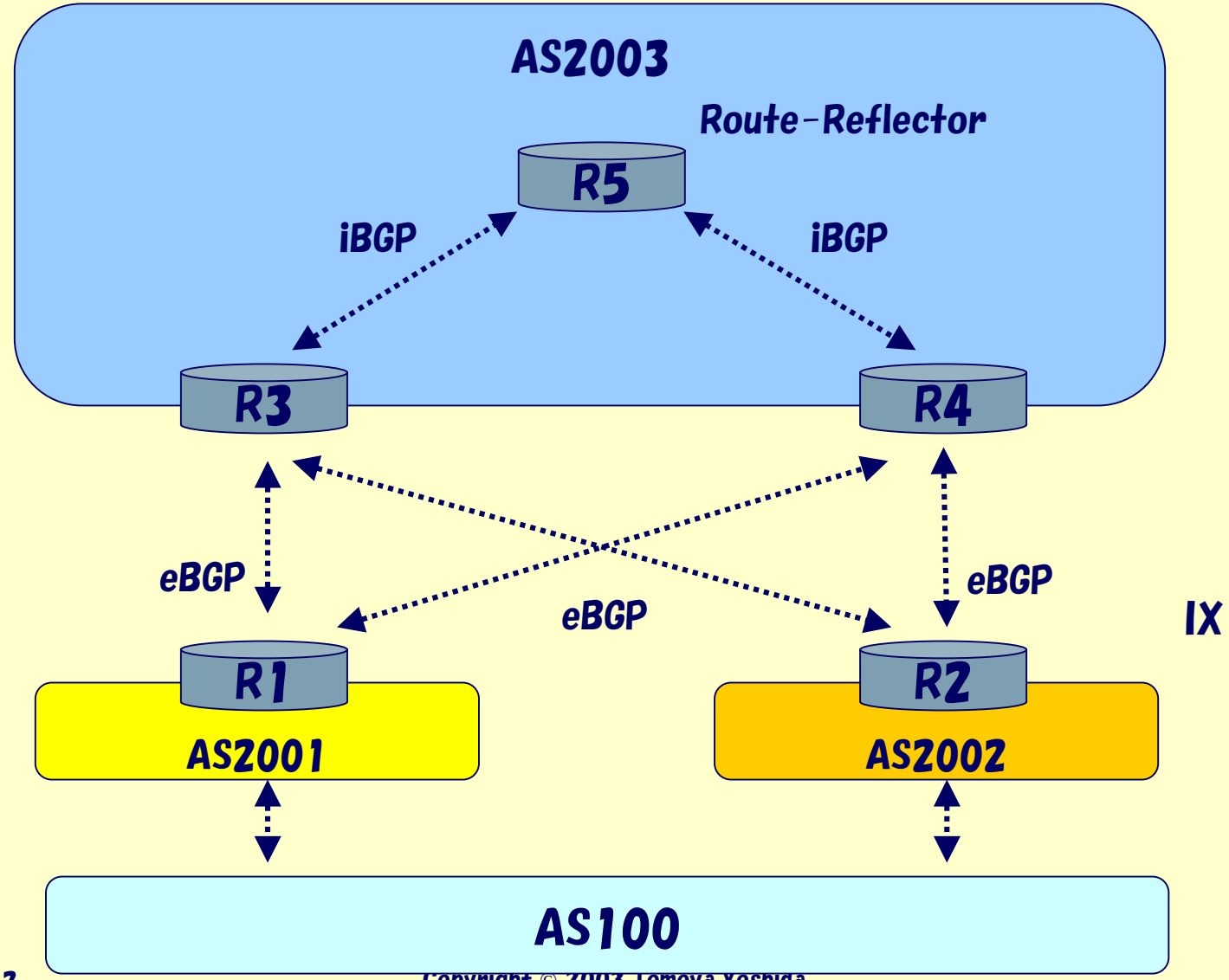
BGP経路比較(MED編)[7]

それぞれ MED200 の経路がベストとなっている(AS2001, AS2002を合体)



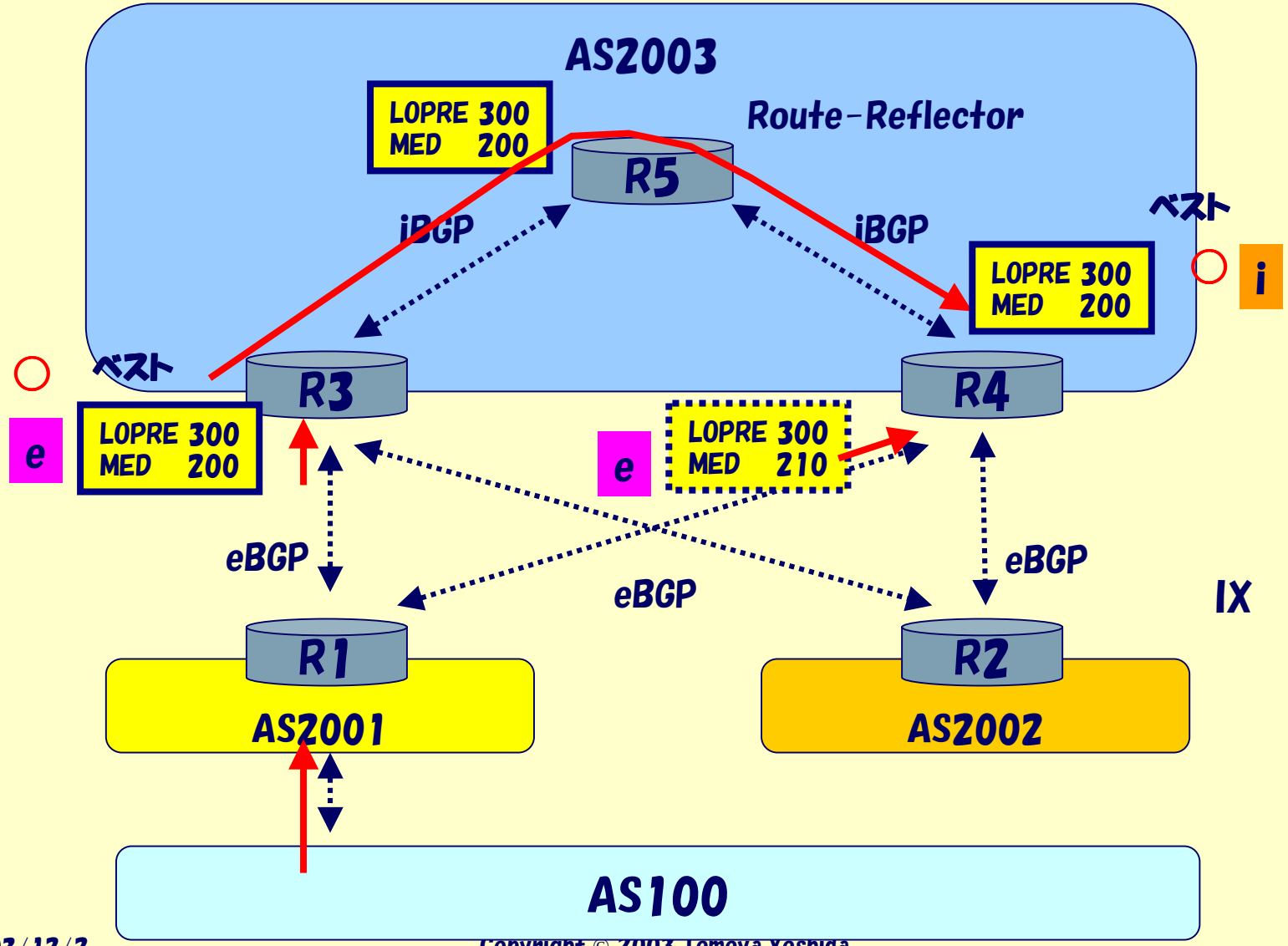
BGP経路比較(MED編)[8]

AS100の経路が，R3とR4から共に広告されてくるようなトポロジーの場合



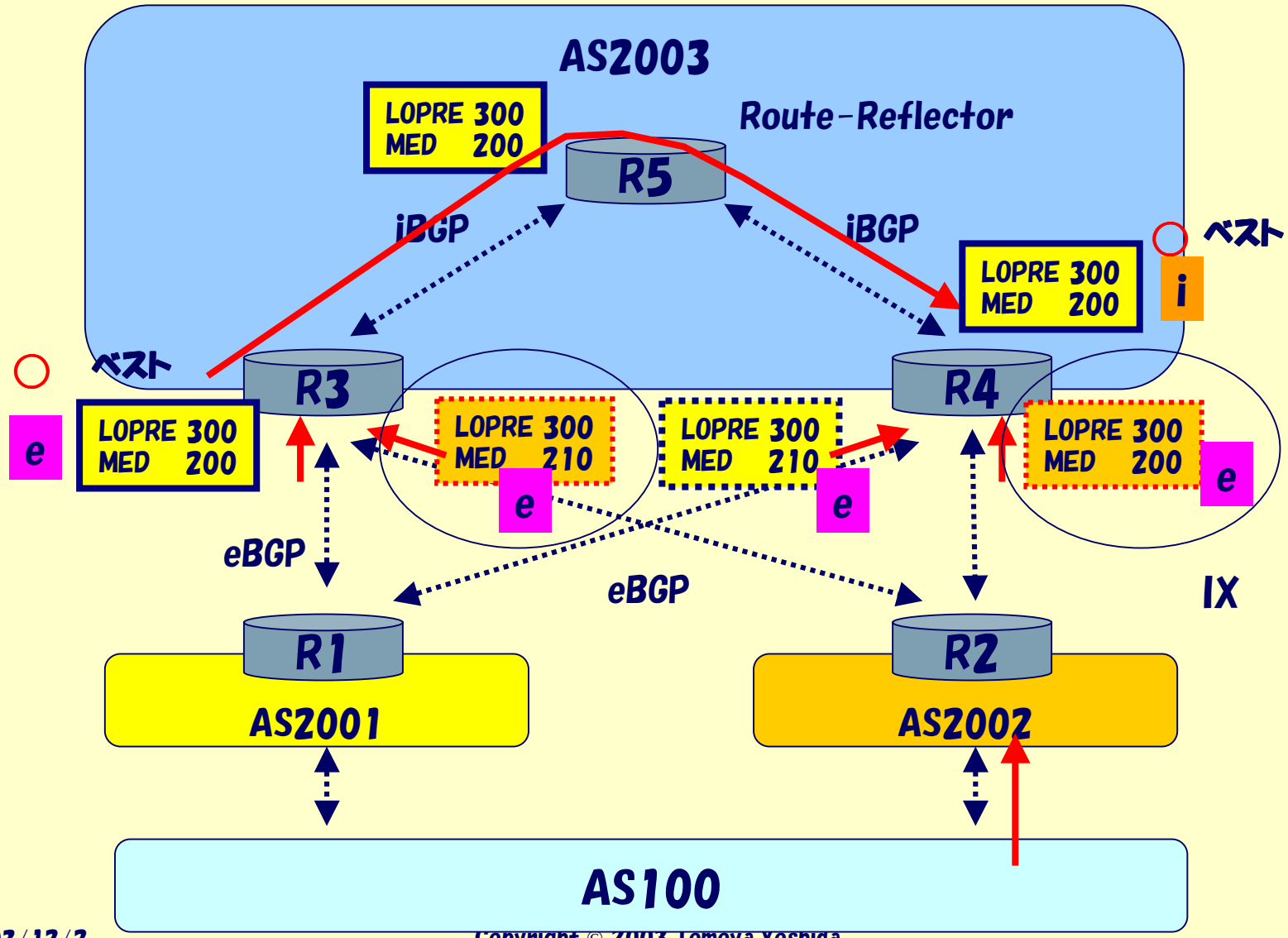
BGP経路比較(MED編)[9]

まず先にAS2001経由でAS100の経路を受信(先に開通したなど)



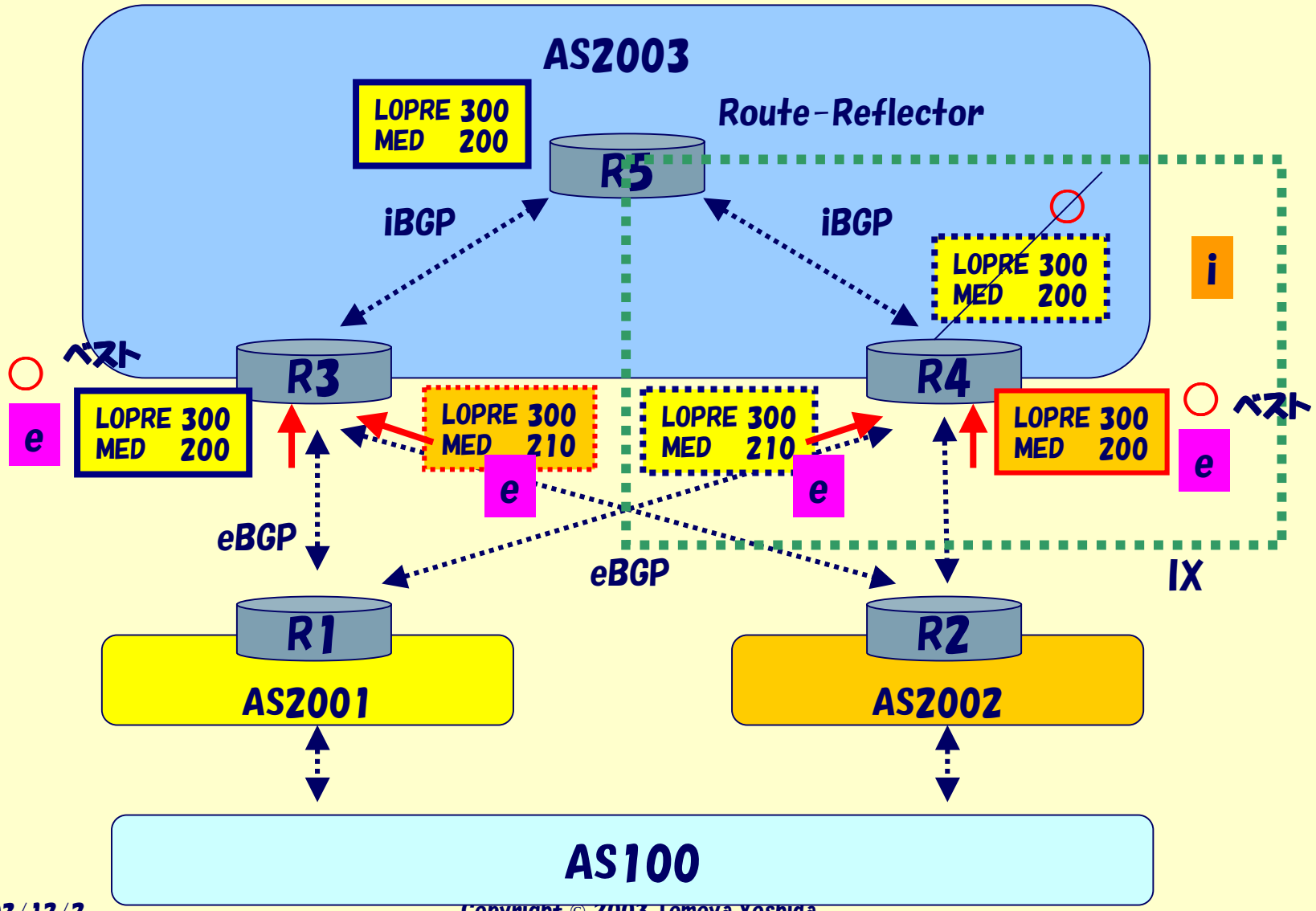
BGP経路比較(MED編)[10]

その後，AS2002経由でもAS100の経路を受信



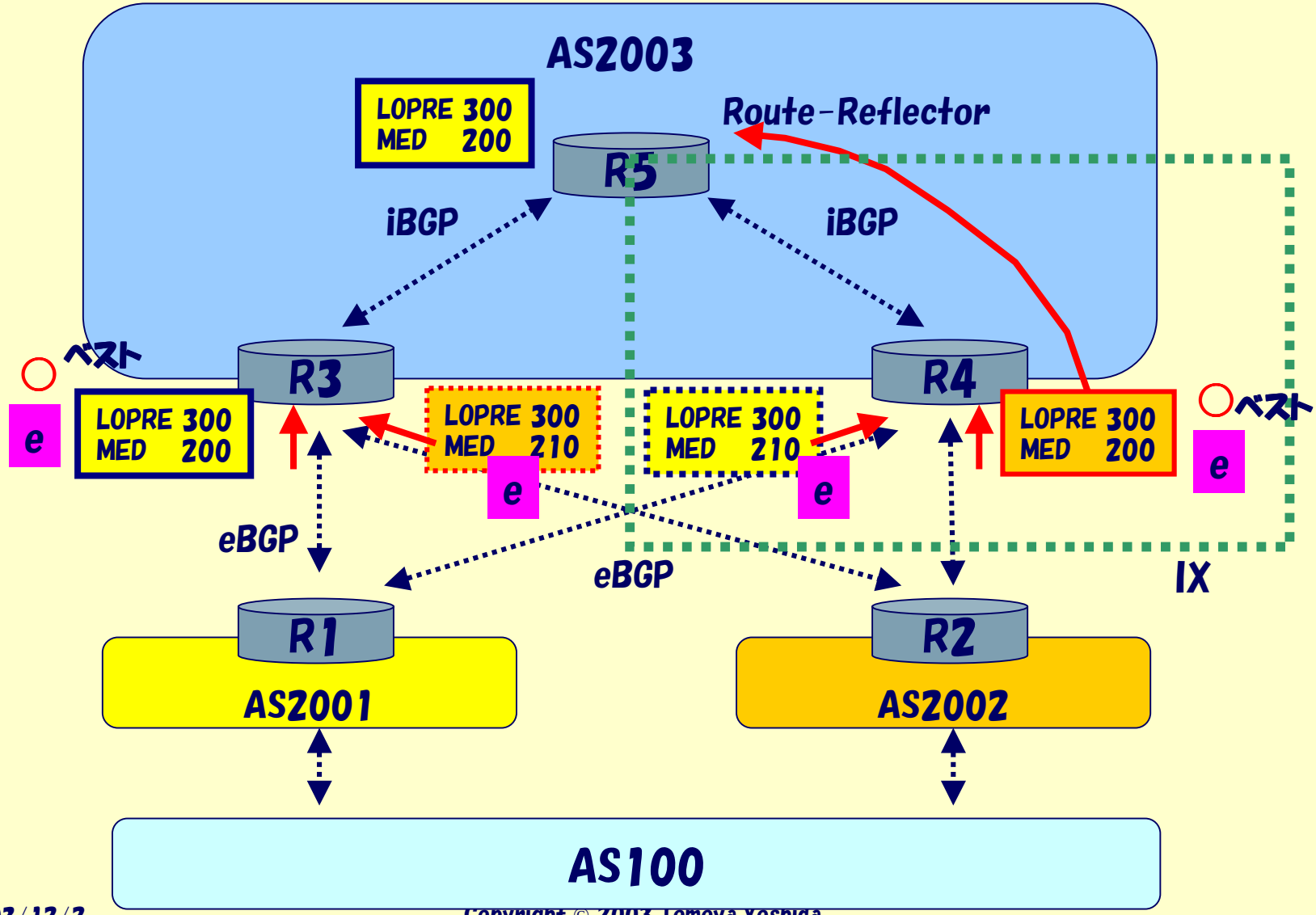
BGP経路比較(MED編)[11]

まずR4では，eBGP経由のAS2002の経路がベストに



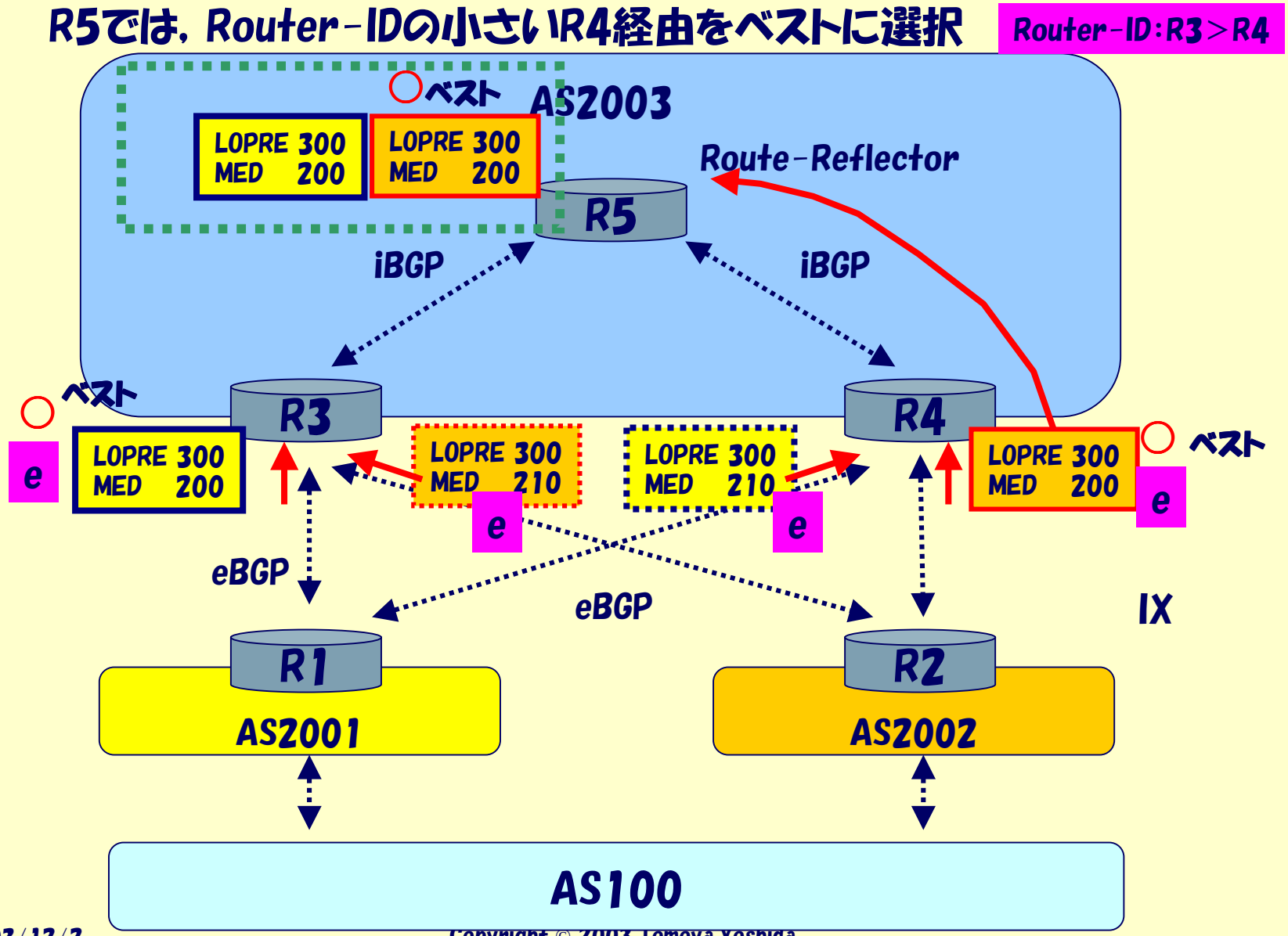
BGP経路比較(MED編)[12]

AS2002経由のベスト経路をR5に伝播



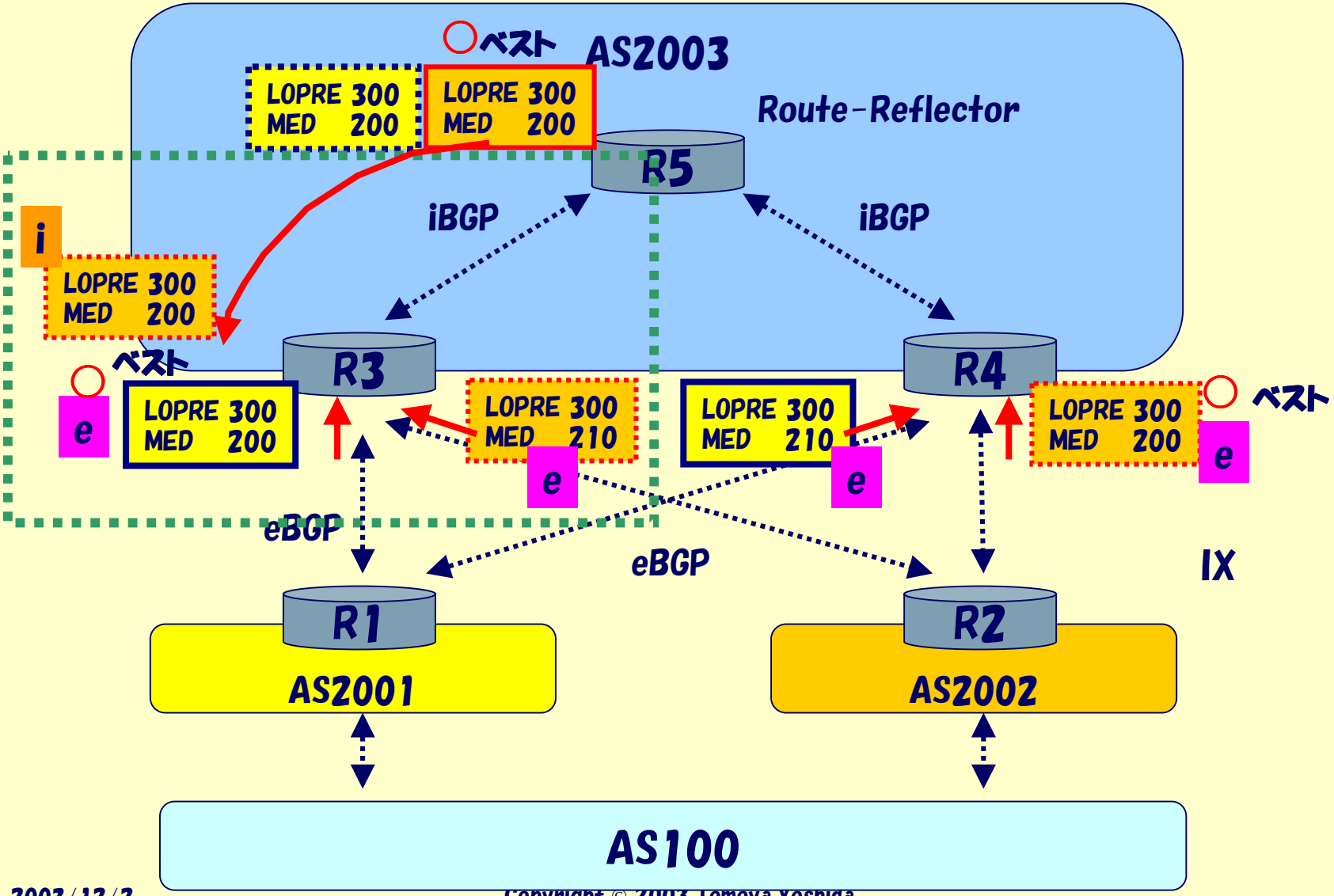
BGP経路比較(MED編)[13]

R5では，Router-IDの小さいR4経由をベストに選択



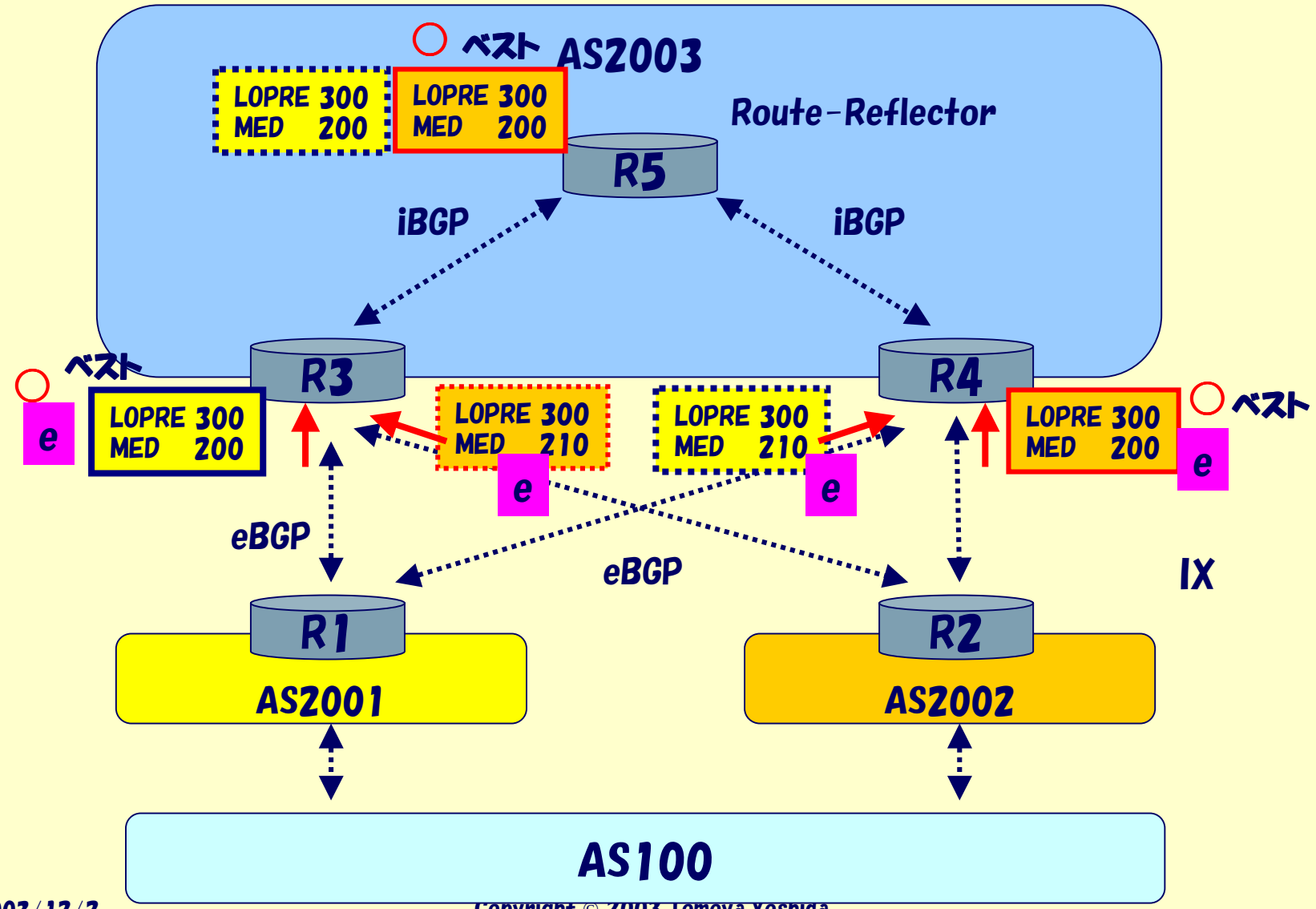
BGP経路比較(MED編)[14]

R3では，Peerタイプで直接AS2001経由のeBGP経路をベストに選択



BGP経路比較(MED編)[15]

R3ではAS2001，R4ではAS2002の経路がそれぞれベストに

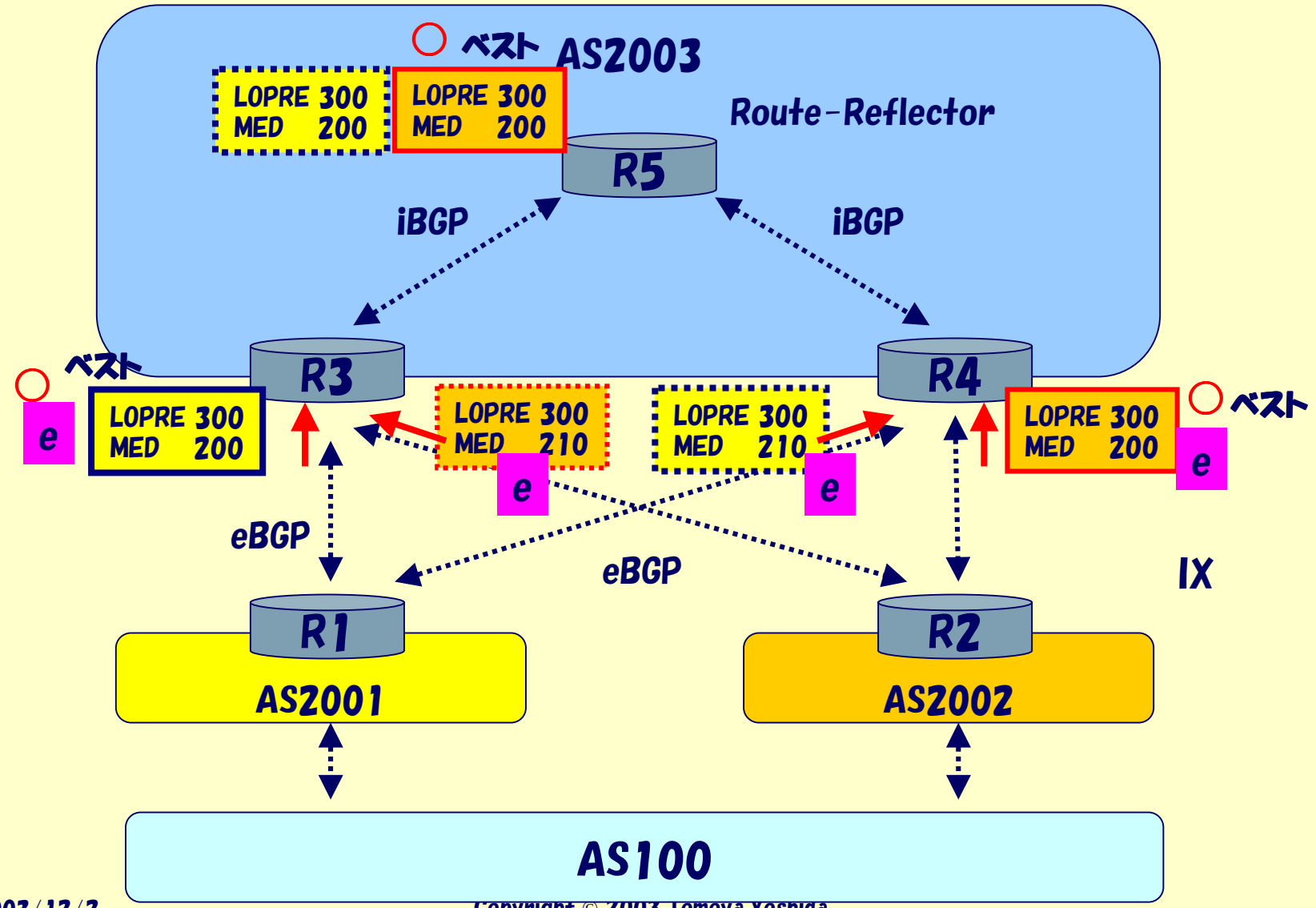


実装の違い

- Cisco
  - eBGPに対して，Router-IDの比較を　しない
  - ルートフラップを考慮した実装らしい
    - 2つのeBGPピアから経路を受信している場合，同一AS_PATHだし，安定して常に広告されている方（先に広告してきた方）を優先的に常に選択していたほうが望ましい
- Juniper
  - eBGPに対して，Router-IDの比較を　する

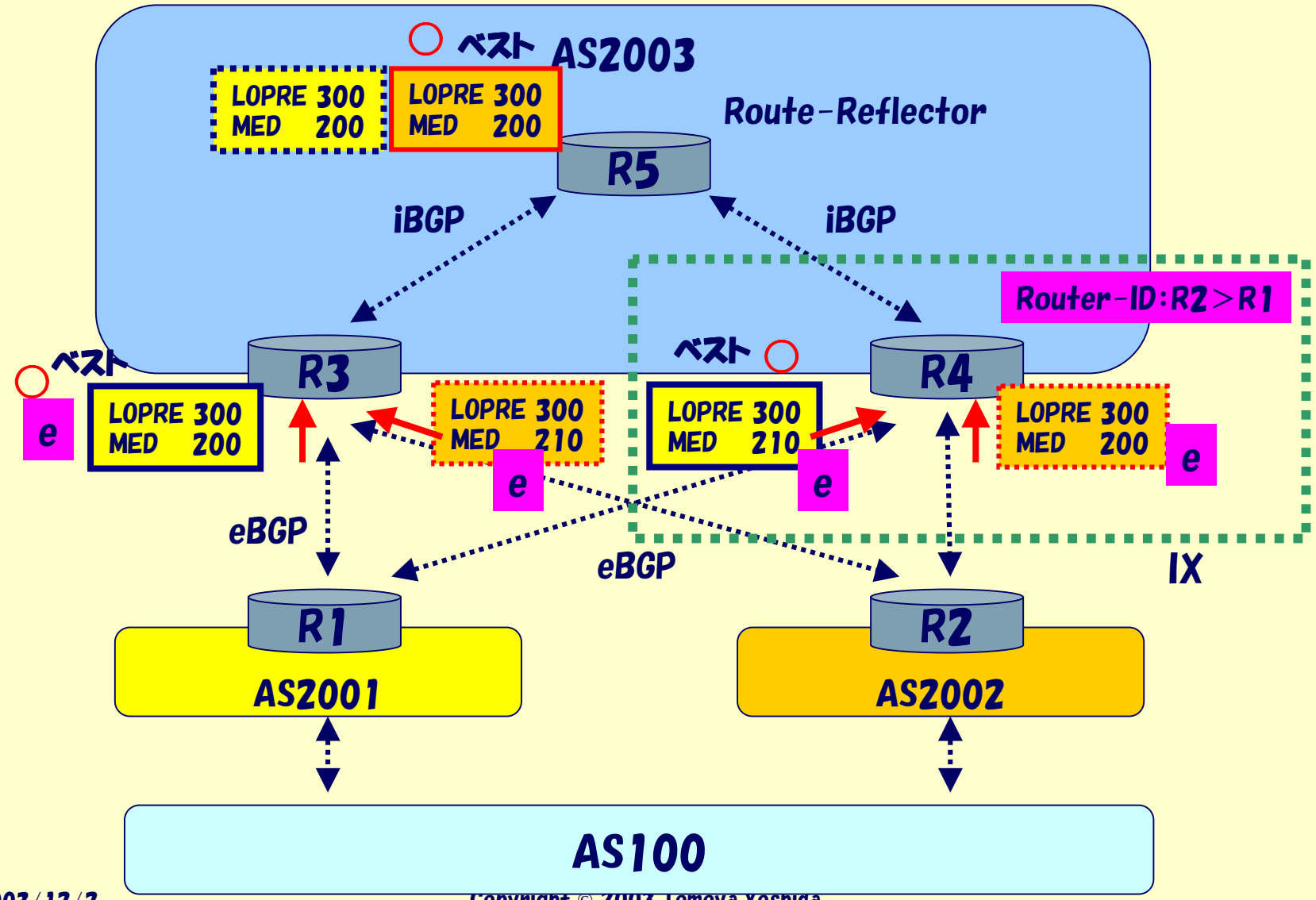
BGP経路比較(MED編):R3/R4 = Cの場合[16]

R3/R4共にeBGPがベストになっているので，このままの状態で落ち着く



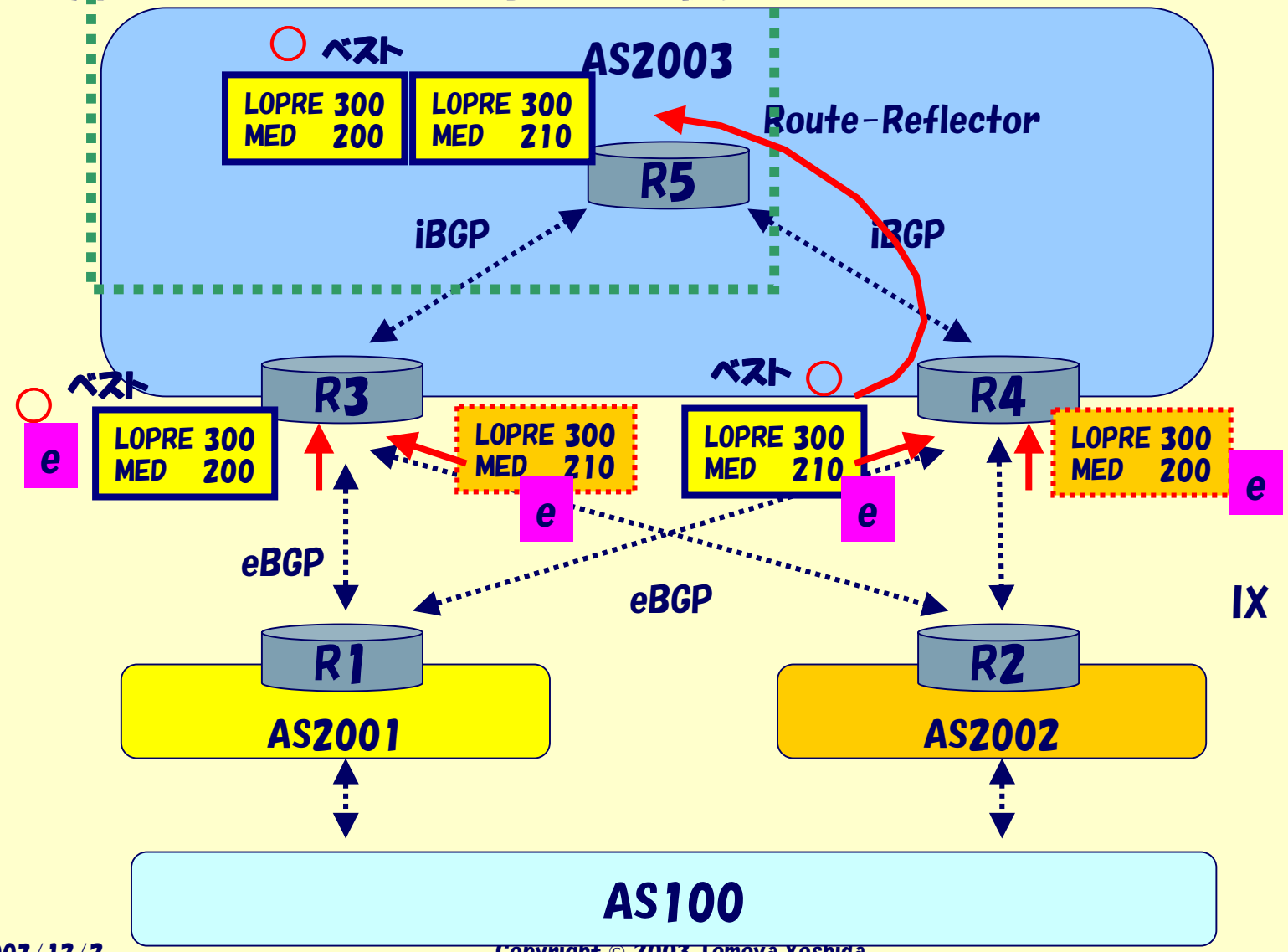
BGP経路比較(MED編):R3/R4 = Jの場合[17]

再度R4で比較をし，Router-IDの小さいR1経由をベストに選択



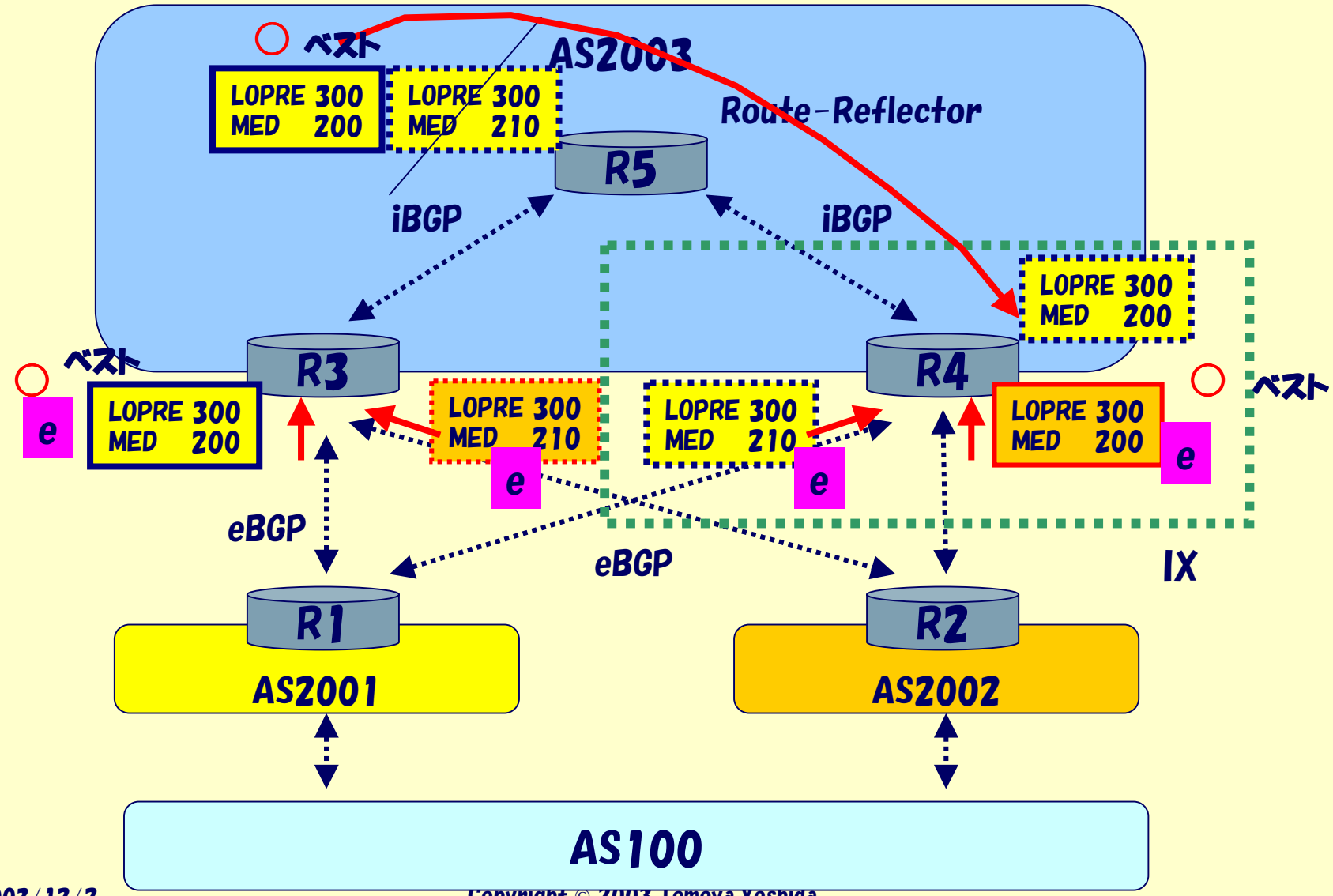
BGP経路比較(MED編):R3/R4 = Jの場合[18]

同一ASなので，MEDの小さいR3経由をベストに



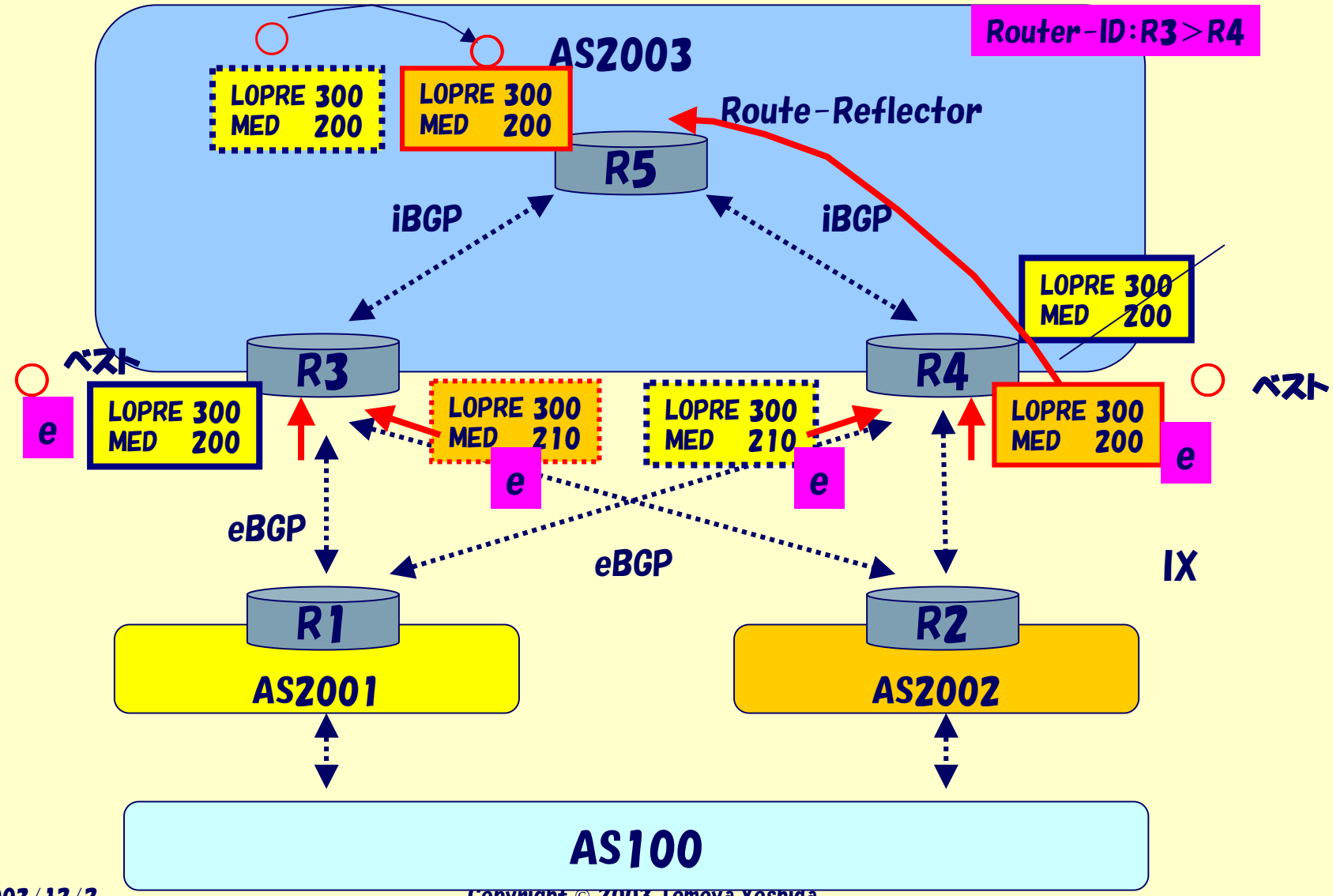
BGP経路比較(MED編):R3/R4 = Jの場合[19]

ベスト経路をR4に配信. R4ではPeerタイプでAS2002の経路が再びベストに



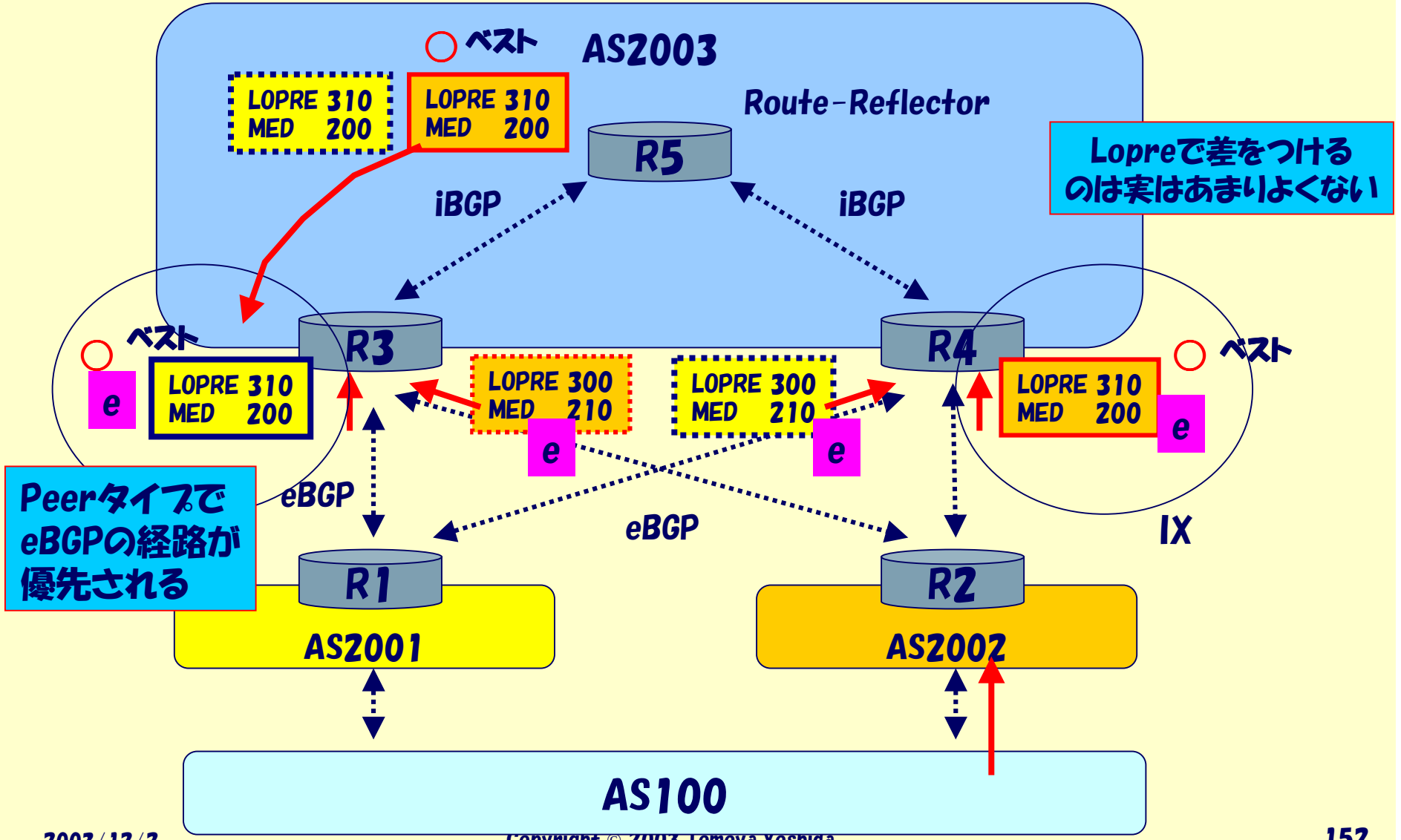
BGP経路比較(MED編):R3/R4 = Jの場合[20]

また元に戻った → 経路情報のフラップが発生している！！



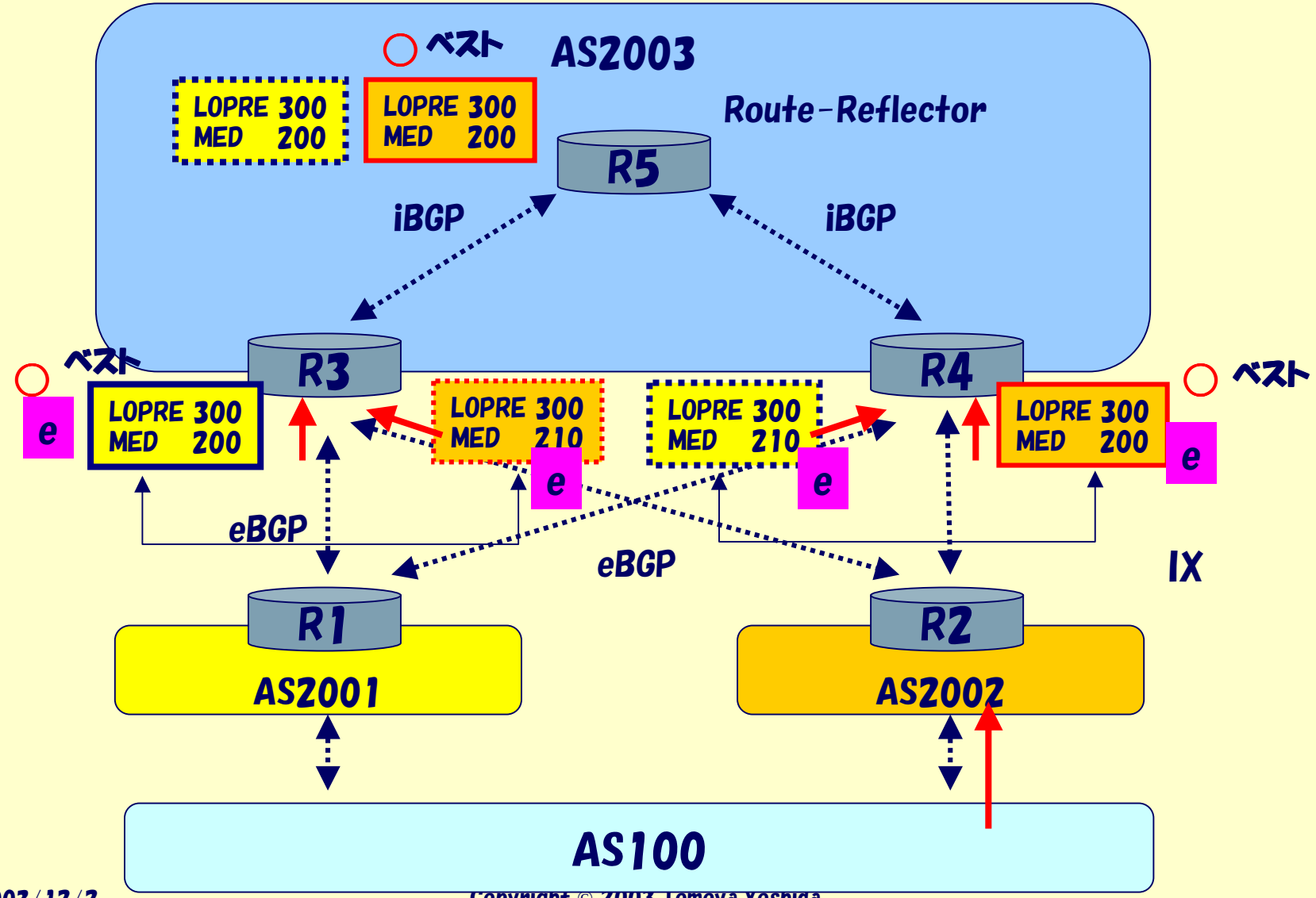
回避策1

明示的に 優先ピアのLOCAL_PREFをあげてしまう(300→310)



回避策2

always-compare-med を使って，異なるAS間でMED比較をさせる

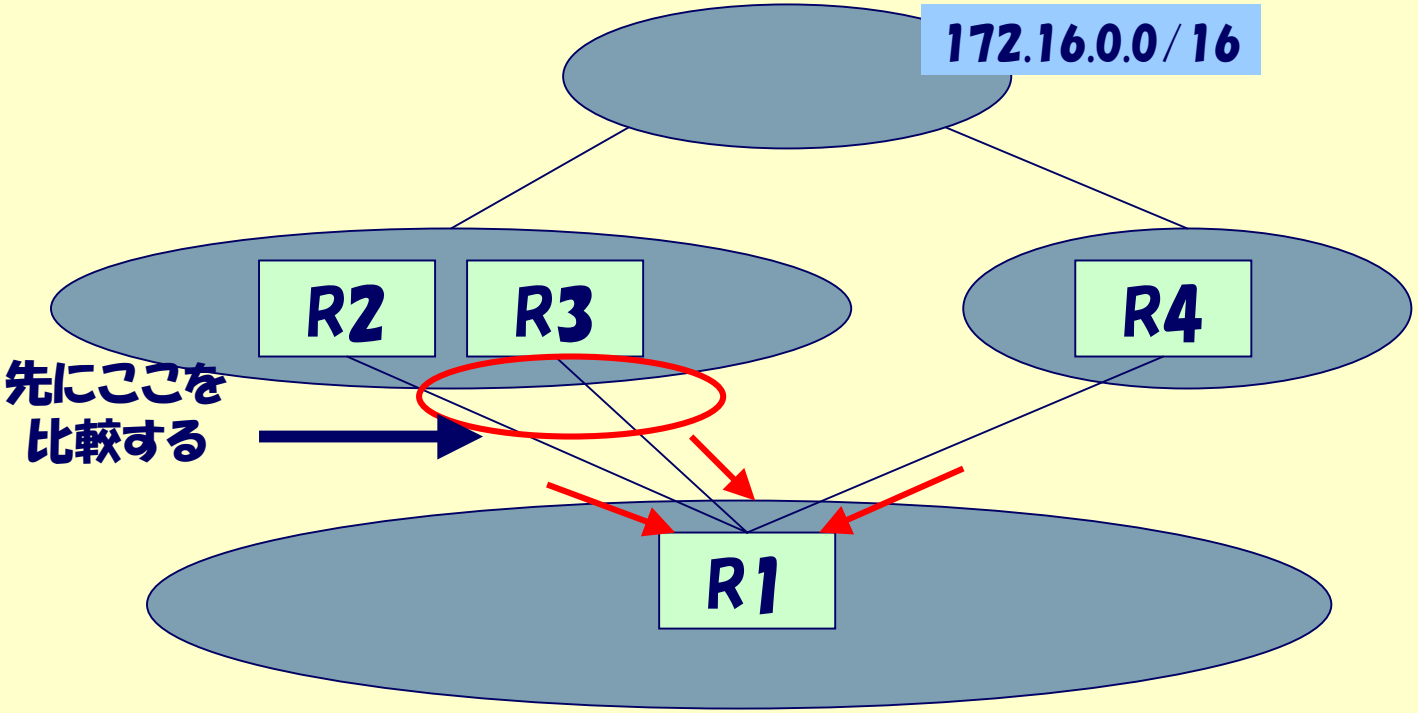


BGP経路周りのトレースの方法

- トレースのポイント
  - 何が起きているのかを把握する
    - tracerouteをして，どういうルーティングをしているか
  - その後、順にログインして，ルーティングテーブル等を追っていく
    - どこをネクストホップにルーティングしようとしているのか
    - 何故下部から経路が配信されてこないのか
    - あるいは，何故違うほうをベストに選択しているか，など
    - 論理トポロジーや物理トポロジーをちゃんと把握した上で調査すること
  - ピンポンしている場合には，大抵片方はデフォルトに従って，もう片方は経路を知っているなどの場合が多い(経験則)
- GWのベストパスは，単にGW自身のベストパスに過ぎない
  - 基本はリフレクタのベスト経路が伝達されているはずなので，パケットの通り道でどのように見えているかを確認しましょう
- こんなことも・・・
  - ちゃんとルーティングテーブルは正しいのに，明後日の方向にパケットをだしている(forwarding-table を clear すると直った　とか)
  - BGP経路がちゃんとピアから流れてこない(ハングっている場合もある)

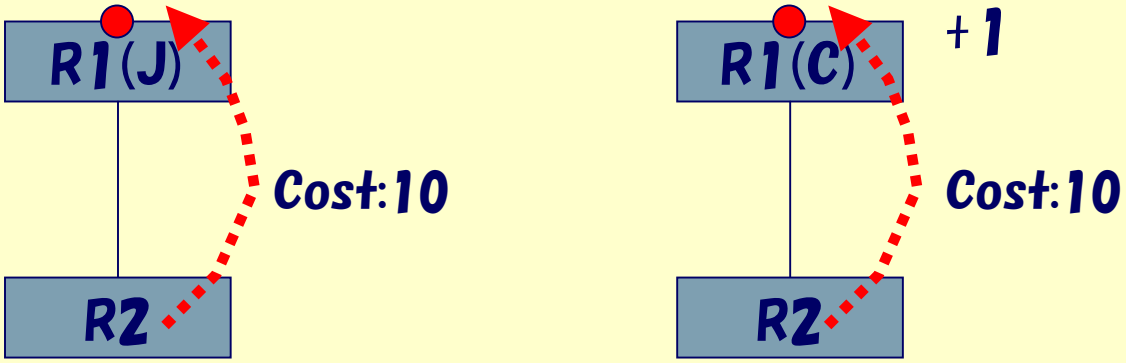
bgp deterministic-med

- 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に見える
  - Juniper：
    - R1がJuniperの場合，R2から見たR1のLoopbackのコストは10のまま
- IGPコストで経路選択をしている場合などは注意が必要



その他

- トラフィック設計
- フィルタリング
  - 経路フィルタ
  - パケットフィルタ
- Black Hole Routing

トラフィックの設計

- In/outでなるべく相殺ができるような収容設計
  - トラフィックの方向をちゃんと把握する
    - Inが多いのか，outが多いのか
  - その上で，同一ルータにどのピア先を一緒に収容すれば効率がよいのかを考えて収容分散設計するなど
    - GWからバックボーン向けの回線の効率化
- 上流やピア先のトラフィックを分析
  - Netflow/cflowd　を用いて測定
  - ピア先のさらにその先のASとのトラフィックが多い → 直接ピア
  - 明らかにおかしなトラフィックが発生しているっぽい
    - mac-accounting　などをやると，staticでむけられているっぽい・・・

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，Privateアドレス，マルチキャスト，リンクローカルなどを遮断
  - 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)
  - 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
    - Privateなどの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
  - 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
    - AS_PATHとPrefix-lengthを組み合わせて，自ASの場合には，細かい経路が出ないように，Prefix-lengthでも制限し，顧客はAS_PATHで制御

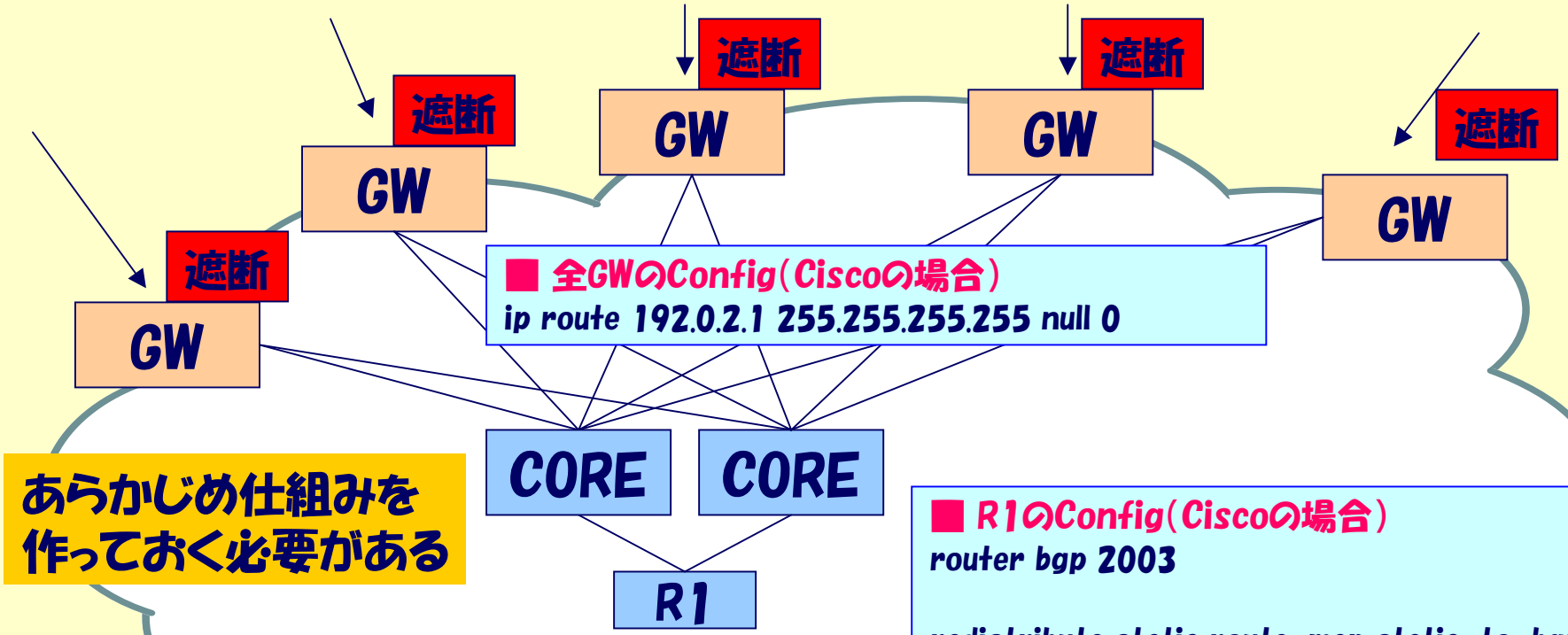
パケットフィルタ

- パケットフィルタを考える前に・・・
  - まず，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
  - やってきたパケットに対して，どういうPolicyを適応するのかを考える
    - ソースアドレスを偽っている場合（スプーフィング）に対して（in）
    - ソースがPrivateアドレスの経路に対して（in）
  - 自分が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
  - 基本は，「自分の身は自分で守る」

- In方向（外部AS→自AS）
  - 共通
    - ソースが自ASアドレス，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チェック）

- 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

Black Hole Routing



あらかじめ仕組みを作っておく必要がある

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

■ R1のConfig(Ciscoの場合)

```
router bgp 2003

redistribute static route-map static-to-bgp

route-map static-to-bgp permit 10
match tag 66
set ip next-hop 192.0.2.1
set local-preference 50
set community no-export
set origin igp

route-map static-to-bgp permit 20
```

ご清聴ありがとうございました

参考資料

参考資料ー1

BGPのベストパス選択一覧表

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

参考資料ー2

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