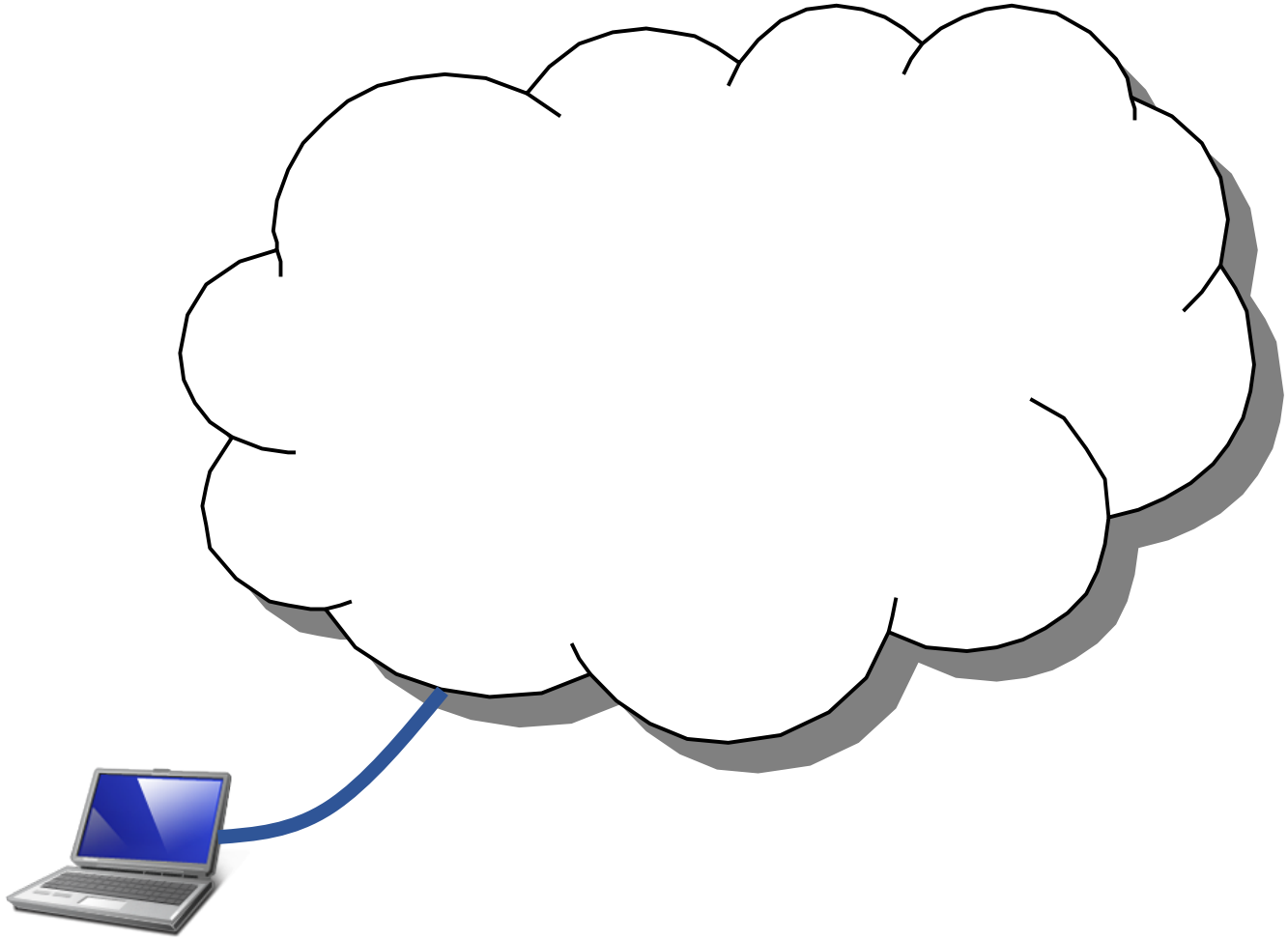
JANOG
BGPチュートリアル
JANOG40 version

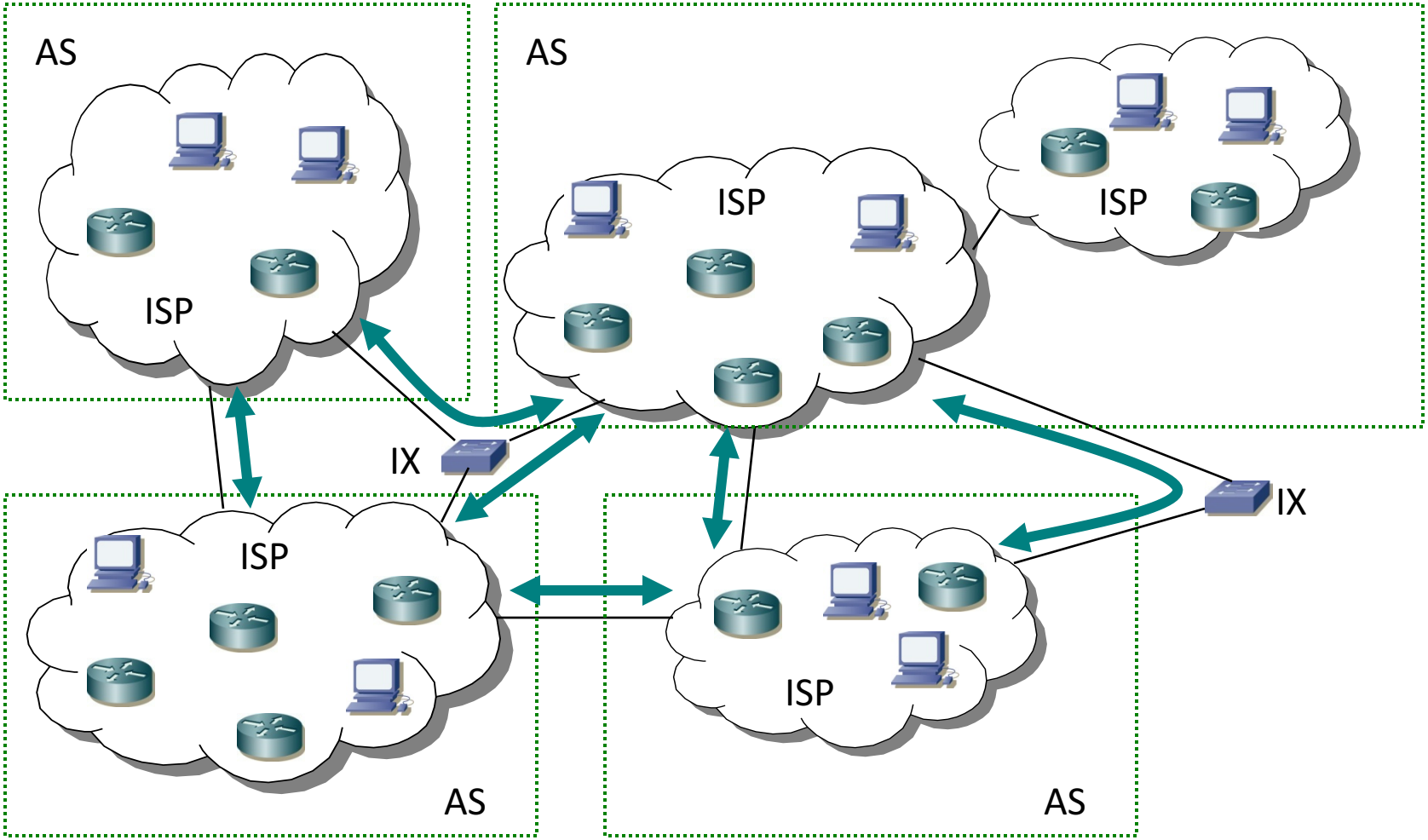
Matsuzaki ‘maz’ Yoshinobu

<maz@iij.ad.jp>

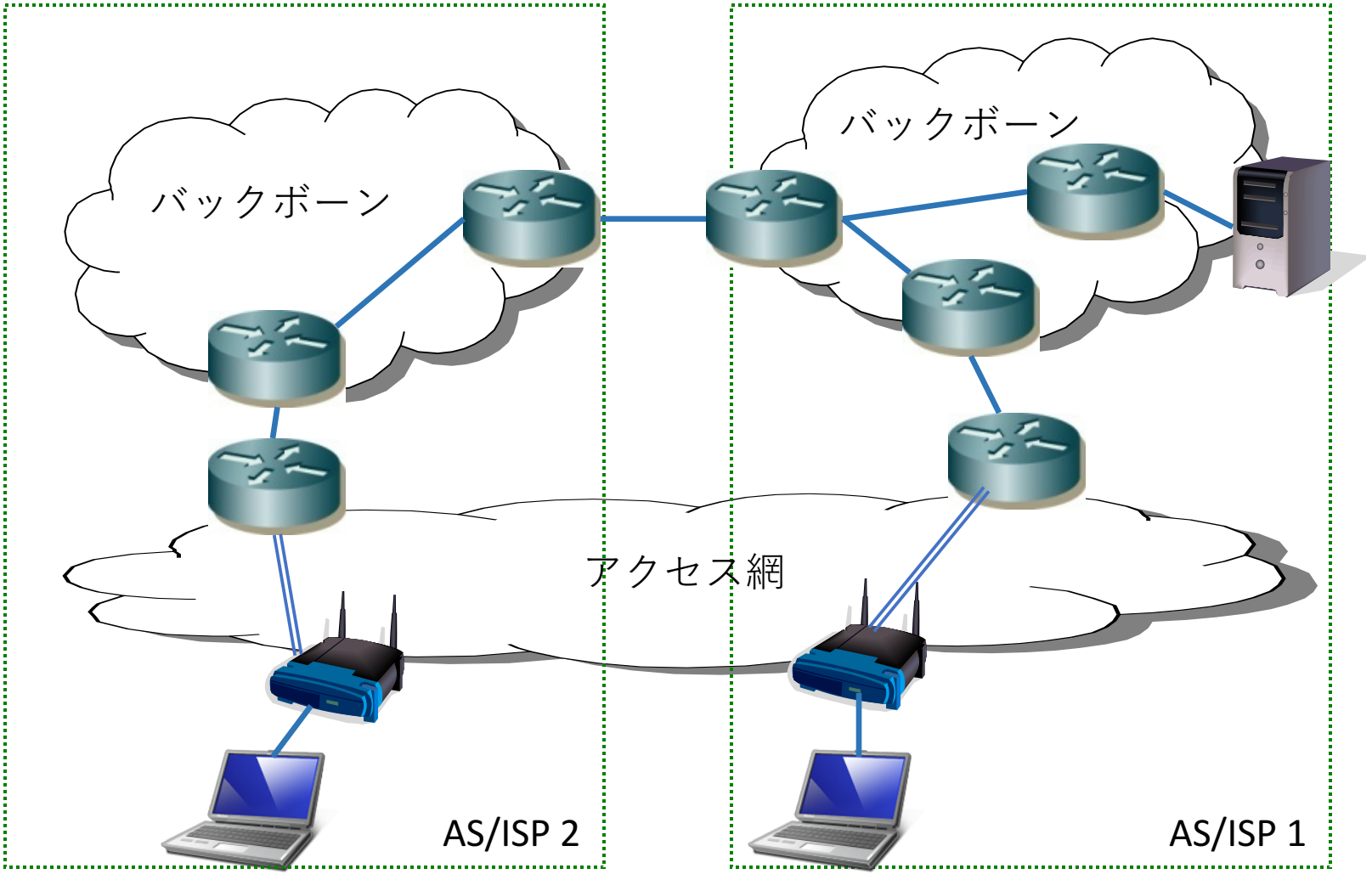
インターネット



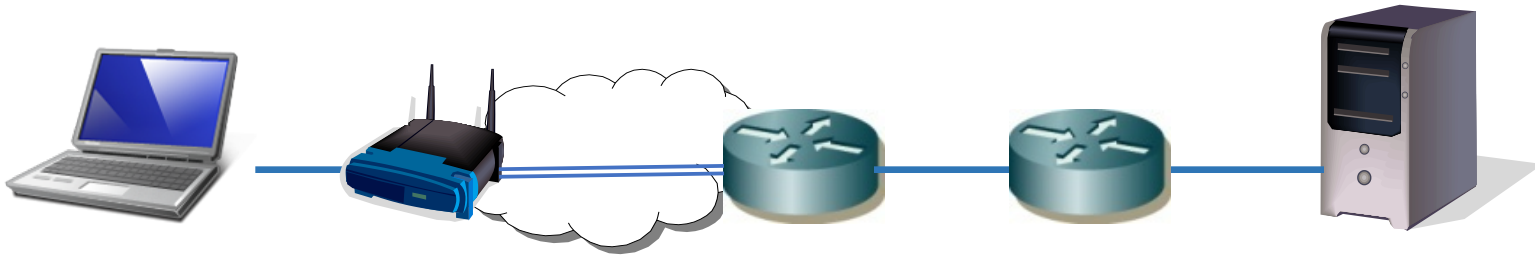
相互接続したネットワーク



アクセス網とバックボーン網



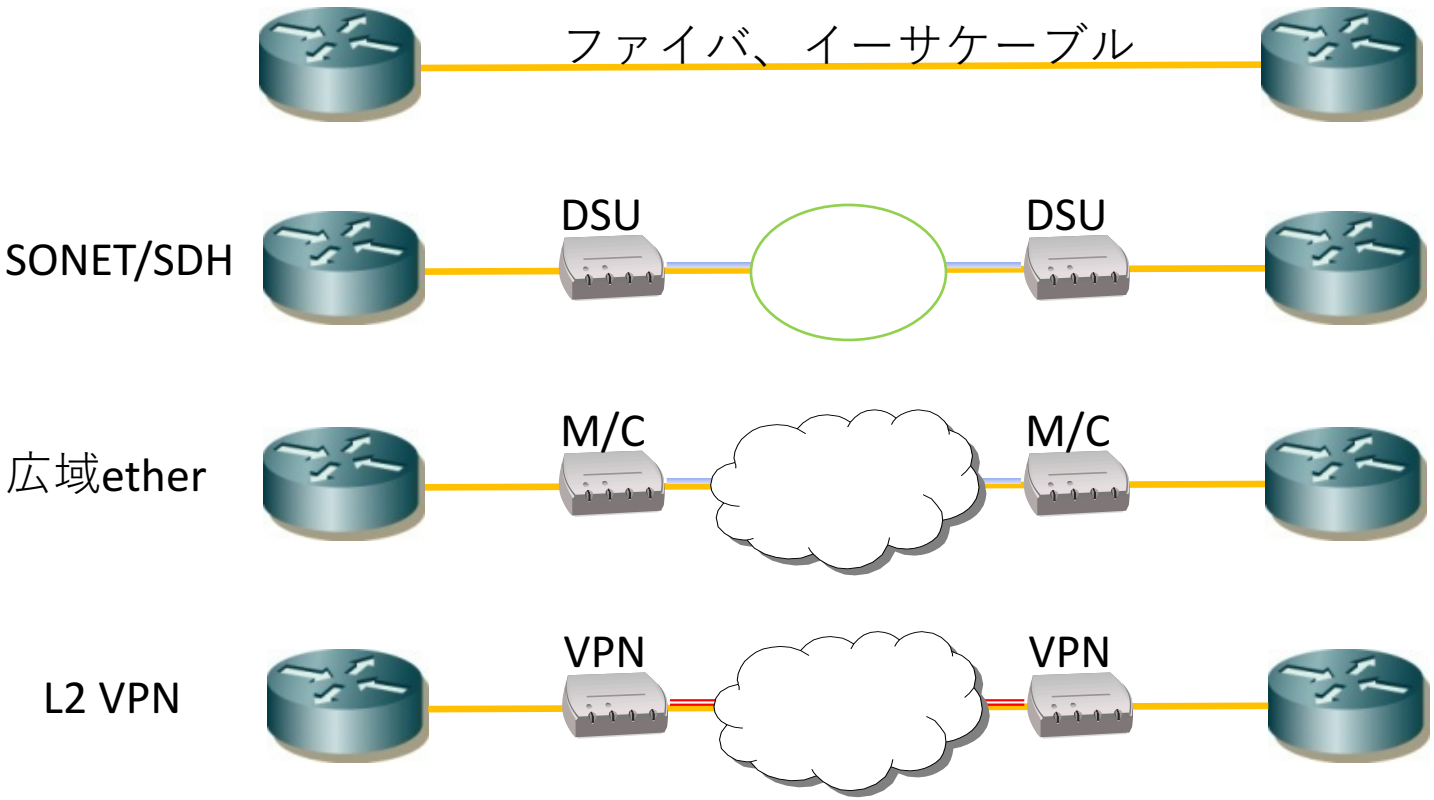
回線、ホスト、ルータ



回線

- IPパケットを転送するための線
  - 専用線、ダークファイバ
  - アクセス網経由の回線(pppoe, ppp)
  - 光ファイバ、イーサケーブル
  - VPNやトンネルプロトコル
- 帯域の保証や到達距離、保守など、メディアやサービスに応じて違いがある
- 実のところ、回線は何が流れてても気にしない
  - IP以外でも良い
  - 独自プロトコルを利用するために利用する人も

2拠点間を結ぶ回線種別



ホスト

- IPで通信したい人たち
  - タブレット、スマートフォン、PC、ゲーム機
- それぞれネットワークに接続するためのインターフェスを持つ
  - イーサネット
  - 無線LAN、無線WAN
  - シリアル、パラレル
- 実はルータになることも可能
  - 例えばテザリングや接続共有

ルータ

- IPパケットを経路表に応じて転送する人たち
  - ブロードバンドルータ
  - エンタープライズ用ルータ
  - バックボーン用ルータ
- 利用できるインタフェースやルーティングプロトコル、学習できる経路数などで違いがある

ルータの違い

- とあるブロードバンドルータ
  - 148,810pps (単純計算で6micro sec/packet)
- とある大きなルータ
  - 770,000,000pps (単純計算で1pico sec/packet)

- 高速化のための努力
  - 専用ハードウェア
  - 分散処理

ネットワーク設計

- 利用可能なネットワークが維持される様に
  - 冗長であること
  - 拡張しやすいこと
  - 運用しやすいこと
- 日々のトラヒックを運びつつも、様々な障害に耐え、増設も素直に行え、運用に過度の負荷をかけない

障害

- 回線は切れる
  - 異経路の確保
- ルータは落ちる
  - 通常時の負荷軽減
  - 迂回路の確保
- データセンタでも停電する
  - 一カ所に依存しない運用

拡張しやすさ、運用しやすさ

- 動くネットワークは誰でも設計できる
  - 障害を考慮しない設計など
- 維持できるネットワークを設計しないと駄目
  - 増強時にも素直に拡張できる
  - トラブル時に混乱しない
  - シンプルで一貫性のあるポリシ
  - 設定変更時に変更箇所が少なくて済むように

設計の制限事項

- 電源
  - 割り振られた電源容量
- 場所
  - 機器を設置するラック数
- 回線
  - 長距離区間を引ける本数、帯域
  - 引き込める回線種別
- ルータや機器
  - ポート数やインタフェース種別
  - サポートしているプロトコル、機能

RFCと実装

- 全ての実装が標準に忠実とは限らない
  - 実装ミス
  - 運用上や性能上の都合
  - 独自の拡張機能
    - 後にRFCとなる場合もある
- 異なる実装の相互接続で問題となりうる
  - OSPFのタイマーとか

標準技術と非標準技術

- 標準技術
  - みんなが使ってるのでメンテナンスされる
  - 他の機器で置き換えられる
- ベンダ特有の非標準技術
  - 痒いところを掻いてくれる(かも)
  - さっさと利用できる
- どれをどう採用するかはネットワークに寄る
  - IIJでは標準技術を重視

機器の評価と検証

- ベンダでも全てを検証しているわけではない
  - 特定機能の組み合わせで発生するバグとか難しい
- 求める機能、性能が利用できるか確かめる
  - カタログスペックなんて当てにならない
  - 自分たちが使うところを集中的に
    - 標準的な構成、機能を利用していると安心感

IPv4アドレス表記

- 32bit長を8bit毎に10進数表記、「.」で繋ぐ
- 192.168.0.1

IPv6アドレス表記

- 128bit長を16bit毎に16進数表記、「:」で繋ぐ
- 2001:0db8:0000:0000:0000:0000:0000:0001
  - 先頭の0を省略 2001:db8:0:0:0:0:0:1
  - 連続の0を圧縮 2001:db8::1
    - ただし、::は一か所だけ (ex: 2001:db8::1:0:1)

ネットワークのプレフィックス表記

- 192.168.0.0/24
  = 192.168.0.0～192.168.0.255
  = 192.168.0.0 mask 255.255.255.0
- 2001:db8::/64
  = 2001:db8:: ～ 2001:db8::ffff:ffff:ffff:ffff

- 連続ネットマスクが前提
  - こんな非連続ネットマスクは表現できない
    - 192.168.0.10 mask 255.255.0.255
  - 複数行での表記になる場合
    - 192.168.0.0～192.168.2.255
    - 192.168.0.0/23, 192.168.2.0/24

クラスレス(Classless)

- クラスの概念はIPv4の過去の遺物なので忘れよう

- 昔はネットワークアドレスの認識に利用
  - IPv4アドレスを見れば、ネットマスクが分かった
  - RIPなどで利用
  - 最近はプロトコルでプレフィックス長を伝播する
  - 今やクラスレスが標準

クラスA　0.0.0.0 ~ 127.255.255.255 → /8
クラスB 128.0.0.0 ~ 191.255.255.255 → /16
クラスC 192.0.0.0 ~ 223.255.255.255 → /24

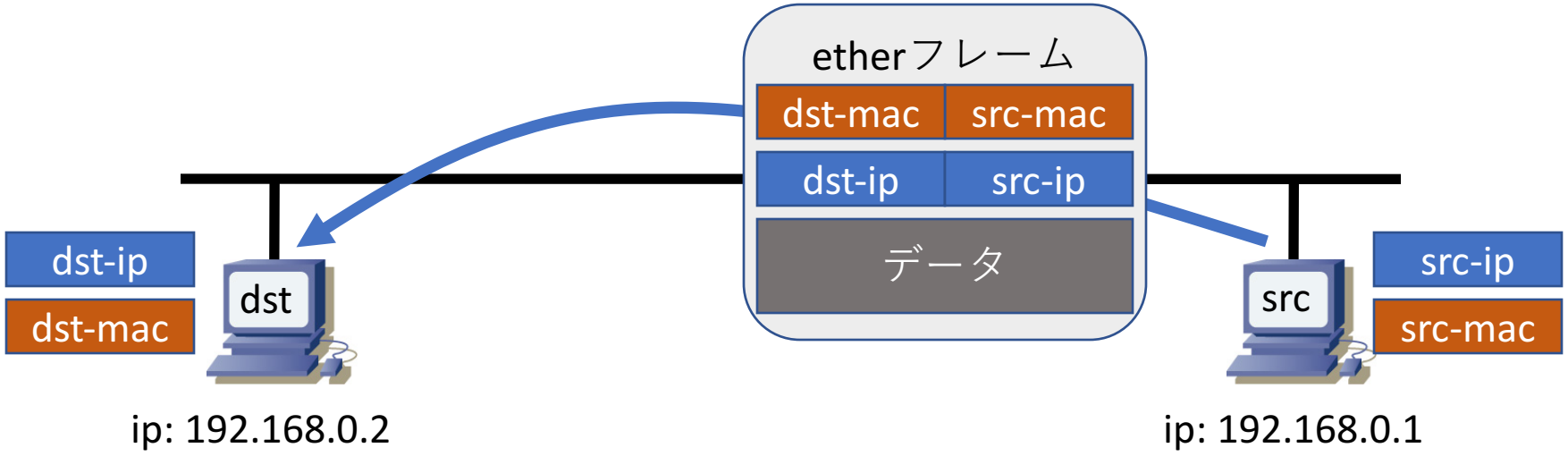
ルーティングとは

- どこを経由してパケットを宛先に届けるか
- 自身に隣接した誰かにパケットを送る
  - もしかすると実際の宛先
  - もしかするとルータ
- 基本的にパケットの宛先IPアドレスをみて判断
  - 特殊な制御をすることも出来るけどお勧めしない

IPv4パケット送信

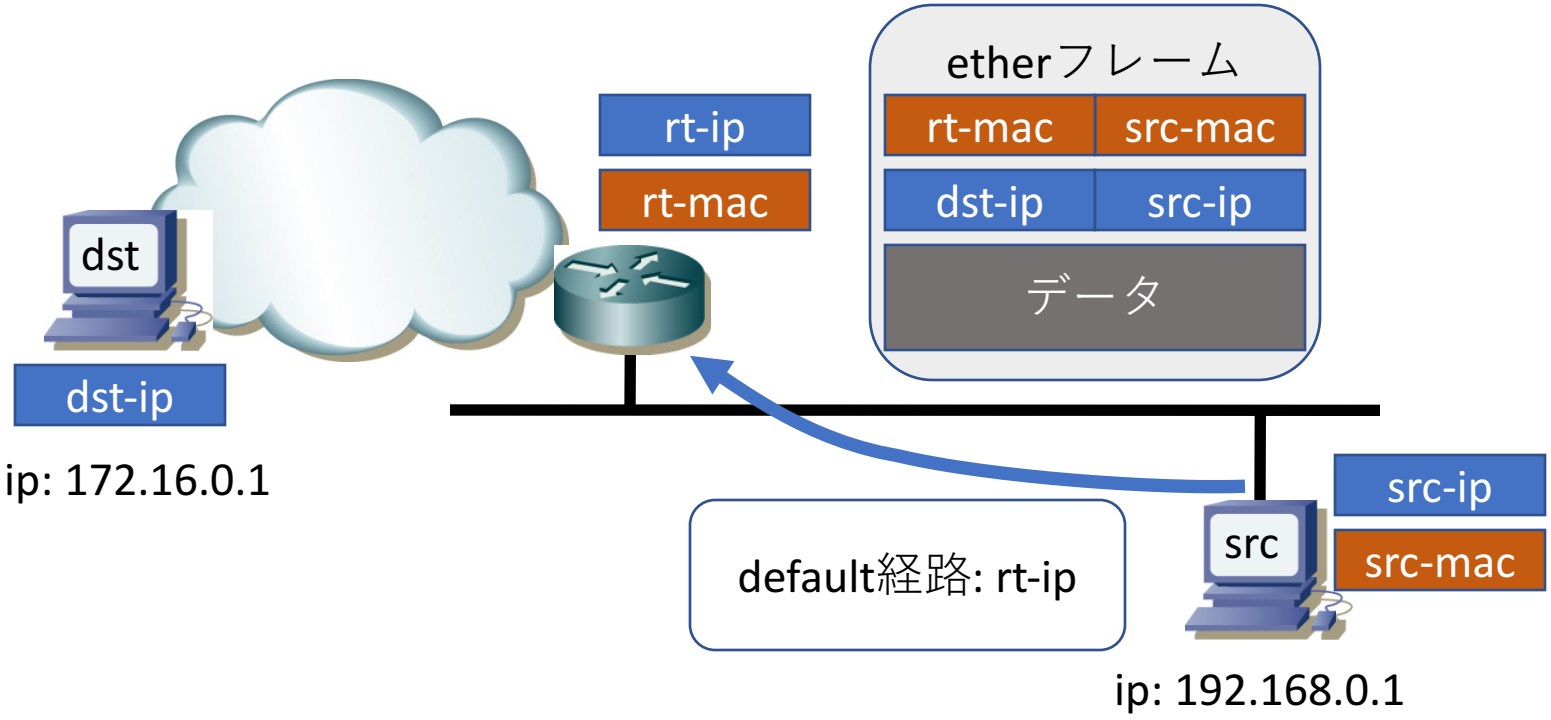
- 同じネットワークに属していれば直接送信

inet 192.168.0.1 netmask 255.255.255.0
↓
192.168.0.0～192.168.0.255が同じセグメント上にある



IPv4パケット送信 2

- 遠くには経路情報に従ってルータに投げる



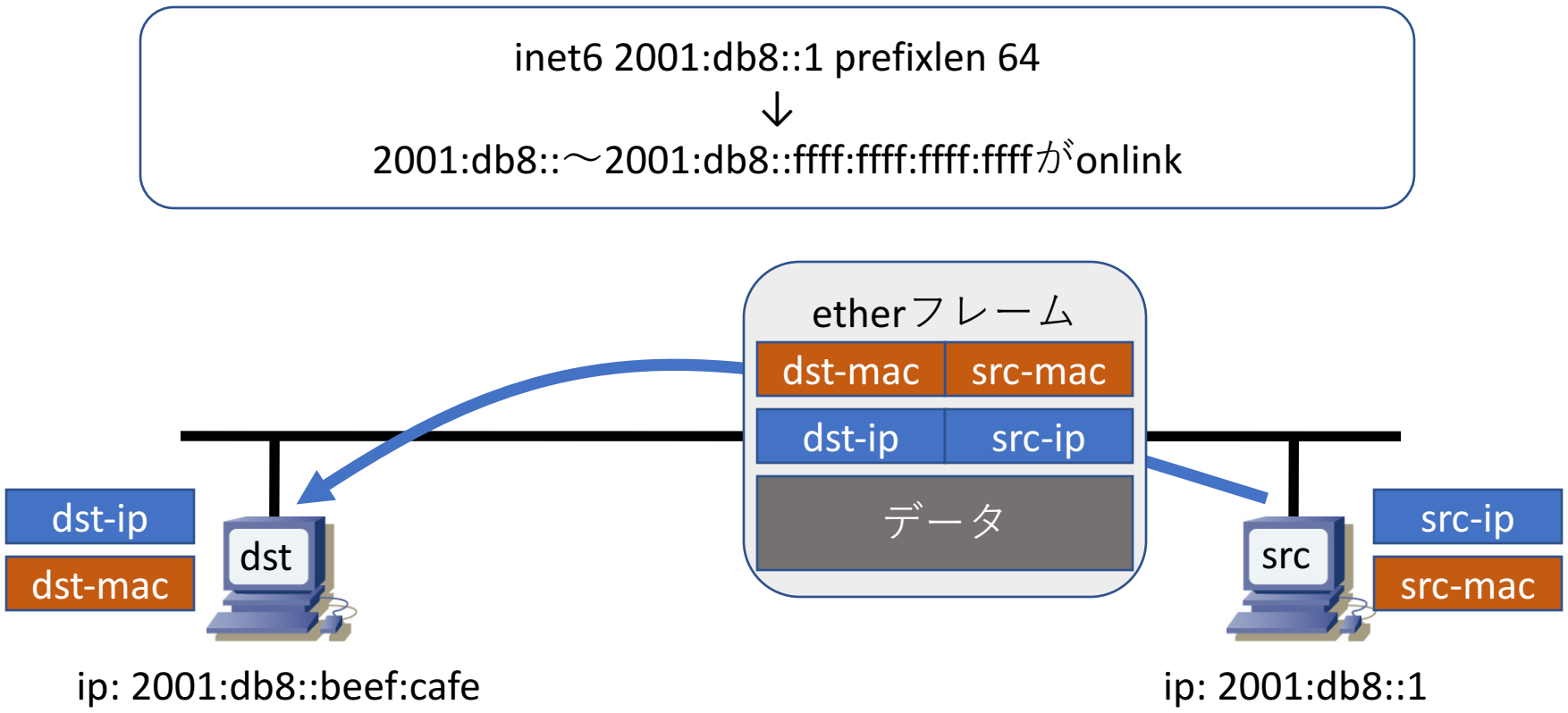
arp (Address Resolution Protocol)

- etherではパケット送信にMACアドレスが必要
  - IPv4アドレスは分かってる (例えばdefaultの向け先)
  - 機器のIPv4アドレスからMACアドレスを知りたい
- arpで解決
  - RFC826

```
arp who-has 192.168.0.2 tell 192.168.0.1
 0x0000:  ffff ffff ffff 0019 bb27 37e0 0806 0001
 0x0010:  0800 0604 0001 0019 bb27 37e0 c0a8 0001
 0x0020:  0000 0000 0000 c0a8 0002
arp reply 192.168.0.2 is-at 00:16:17:61:64:86
 0x0000:  0019 bb27 37e0 0016 1761 6486 0806 0001
 0x0010:  0800 0604 0002 0016 1761 6486 c0a8 0002
 0x0020:  0019 bb27 37e0 c0a8 0001 0000 0000 0000
 0x0030:  0000 0000 0000 0000 0000 0000
```

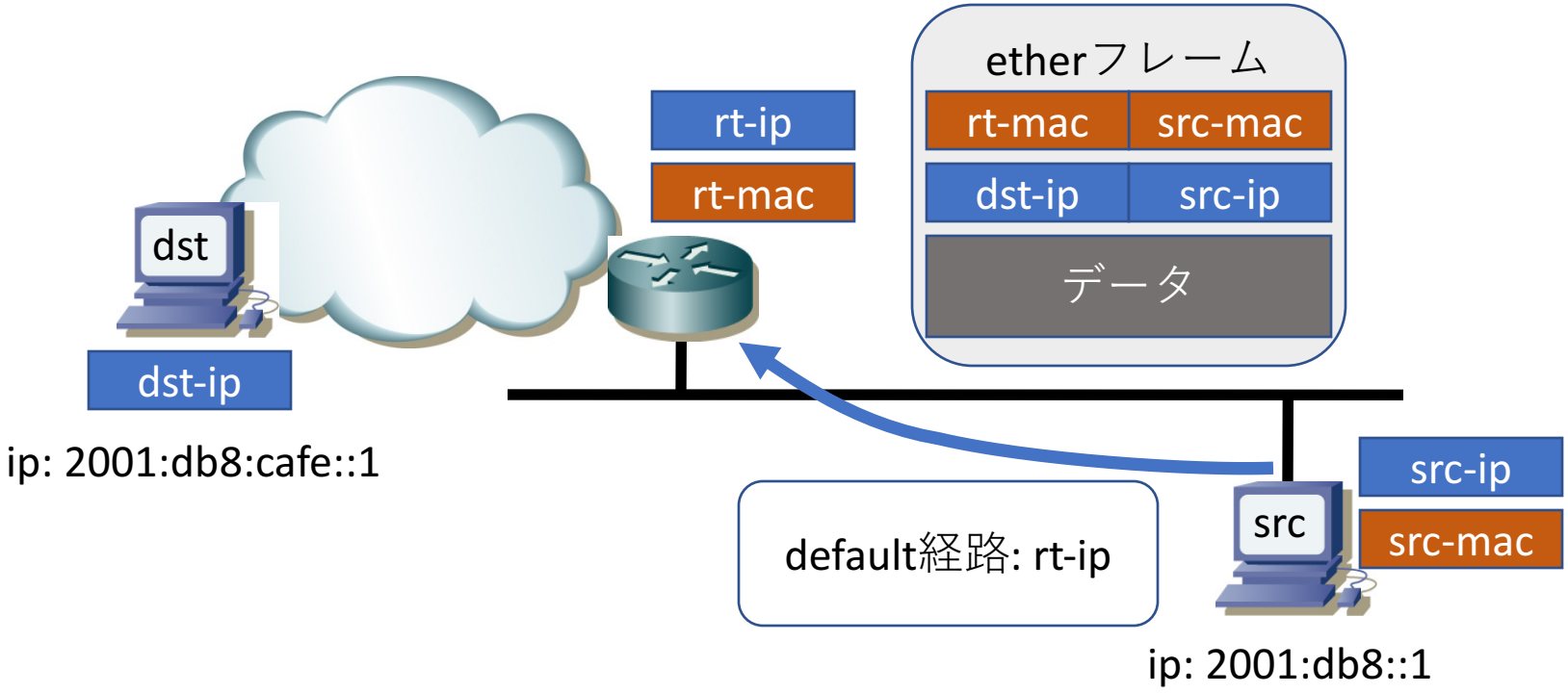
IPv6パケット送信

- 宛先prefixがonlinkだったら直接送信



IPv6パケット送信 2

- 遠くには経路情報に従ってルータに投げる



ndp (Neighbor Discovery Protocol)

- etherではパケット送信にMACアドレスが必要
  - 機器のIPv6アドレスからMACアドレスを知りたい
- ndpで解決
  - RFC4861
    - ICMPv6を利用してMACアドレスを問い合わせる
  - 送り先を未学習ならmulticastアドレス宛て
    - IP: ff02::1:ff00:0000 ～ ff02::1:ffff:ffff
      - 送信先IPアドレスの下位24bitを利用して生成
    - MAC: 33:33:00:00:00:00 ～ 33:33:ff:ff:ff:ff
      - 送信先IPアドレスの下位32bitを利用して生成

ndpでMACアドレス解決

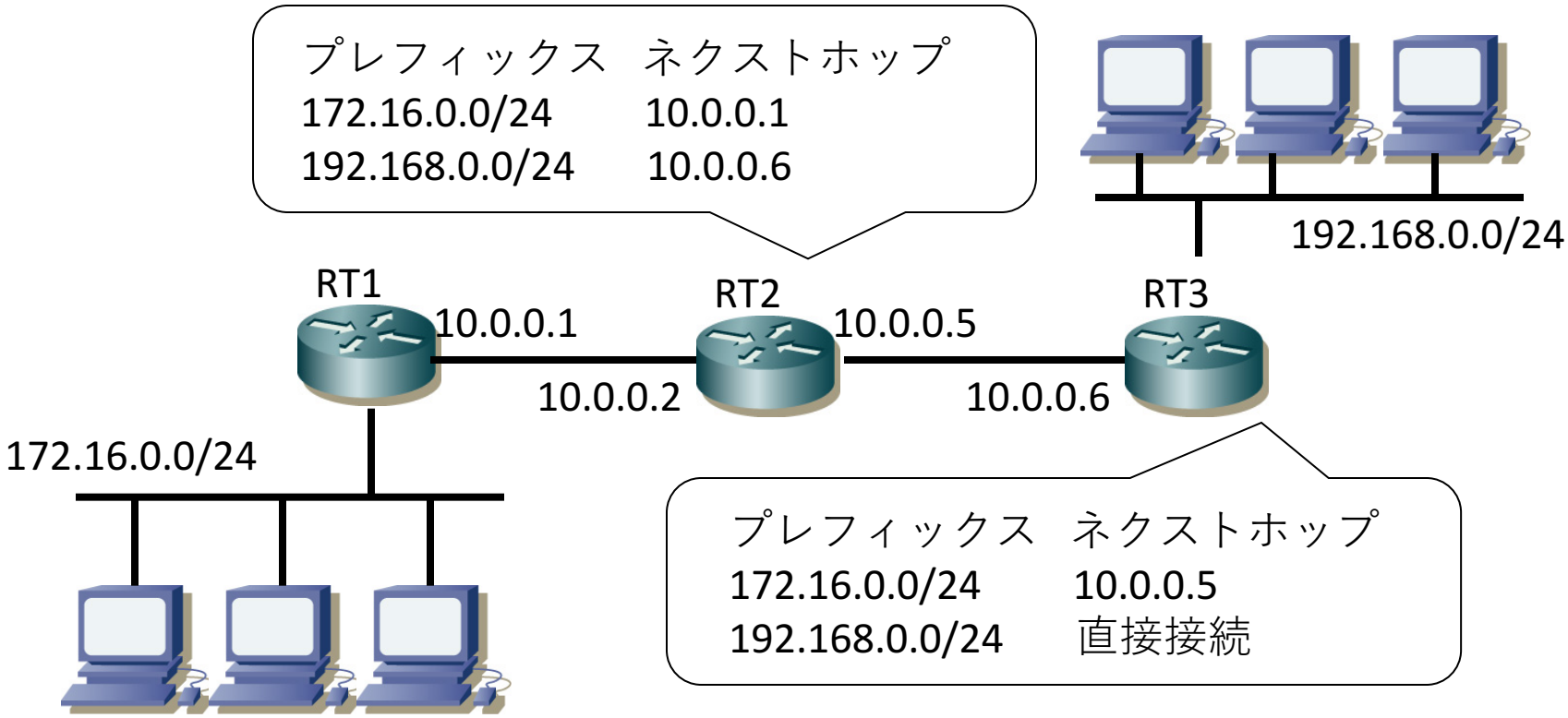
```
IP6 2001:db8::1 > ff02::1:ffef:cafe
 ICMP6, neighbor solicitation, who has 2001:db8::beef:cafe
 source link-address option: 00:19:bb:27:37:e0
        0x0000:  3333 ffef cafe 0019 bb27 37e0 86dd 6000
        0x0010:  0000 0020 3aff 2001 0db8 0000 0000 0000
        0x0020:  0000 0000 0001 ff02 0000 0000 0000 0000
        0x0030:  0001 ffef cafe 8700 9a90 0000 0000 2001
        0x0040:  0db8 0000 0000 0000 0000 beef cafe 0101
        0x0050:  0019 bb27 37e0
IP6 2001:db8::beef:cafe > 2001:db8::1
 ICMP6, neighbor advertisement, tgt is 2001:db8::beef:cafe
 destination link-address option: 00:16:17:61:64:86
        0x0000:  0019 bb27 37e0 0016 1761 6486 86dd 6000
        0x0010:  0000 0020 3aff 2001 0db8 0000 0000 0000
        0x0020:  0000 beef cafe 2001 0db8 0000 0000 0000
        0x0030:  0000 0000 0001 8800 c1fd 6000 0000 2001
        0x0040:  0db8 0000 0000 0000 0000 beef cafe 0201
        0x0050:  0016 1761 6486
```

ちなみにpoint-to-pointリンク

- SDH/SONET/PPPとか
- 回線の先には必ず通信相手が一台だけ
- arp/ndpは利用されない
  - MACアドレス解決が必要ない
- 経路情報に従ってパケットを送出
  - 回線に投げれば相手に届く(はず)

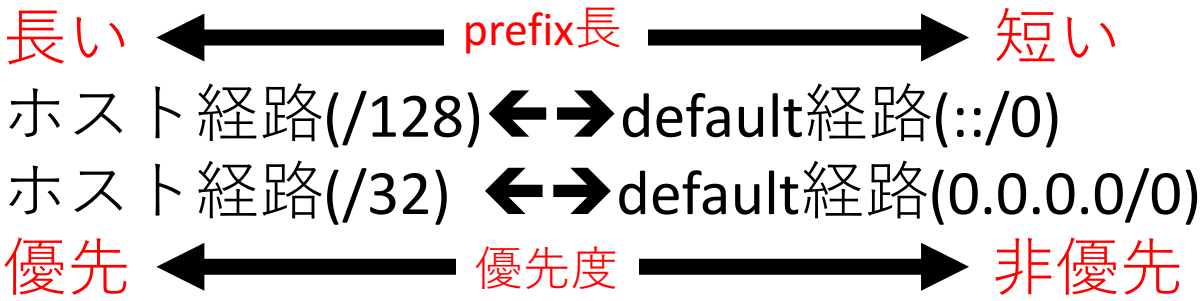
経路情報

- 宛先プレフィックス＋ネクストホップの集合



経路の優先順位

1. prefix長が長い(経路が細かい)ほど優先



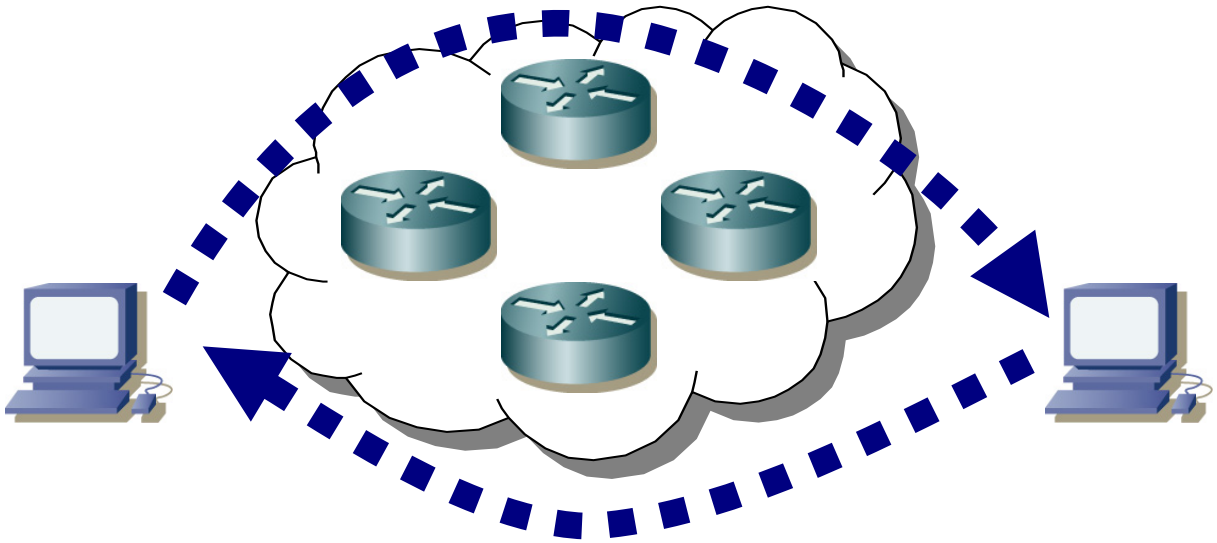
2. 経路種別で優先
   - ① connected経路
   - ② static経路
   - ③ 動的経路(ospf, bgp, etc...)
     - 内訳はベンダ依存

経路の種類

- 静的経路
  - connected経路
    - ルータが直接接続して知っている経路
  - static経路
    - ルータに静的に設定された経路
- 動的経路
  - ルーティングプロトコルで動的に学習した経路
    - OSPFやIS-IS、BGPなどで学習した経路
- これらを組み合わせて適切な経路制御を実現

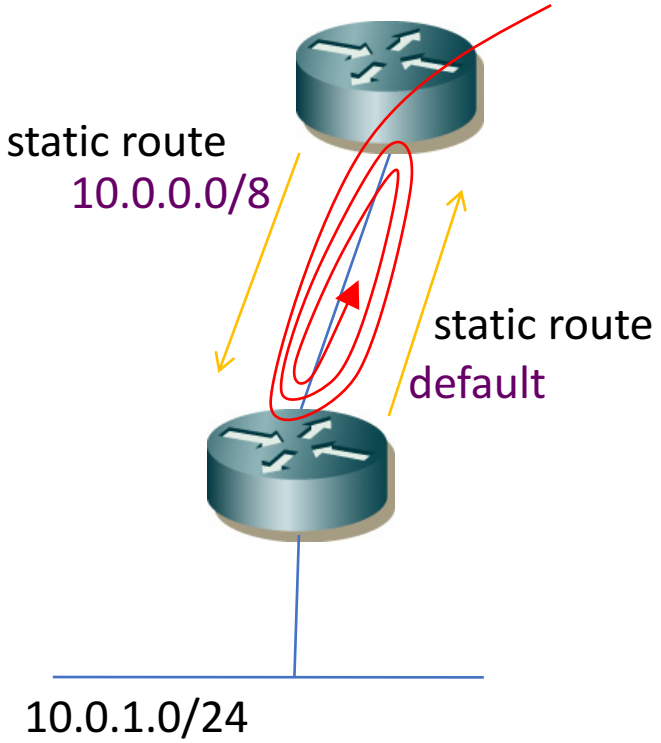
パケットと経路

- 送信元から宛先まで経路に矛盾が無ければ、宛先にパケットが届く
- 双方向で問題が無ければ、相互に通信できる
  - 行きと帰りの経路は違うかもしれない



経路ループ

- 起こしちゃダメ
  - 簡単に回線帯域が埋まる
- 大抵設定/設計ミス
  - 矛盾のあるstatic経路
  - 無茶な設定の動的経路制御



動的経路制御

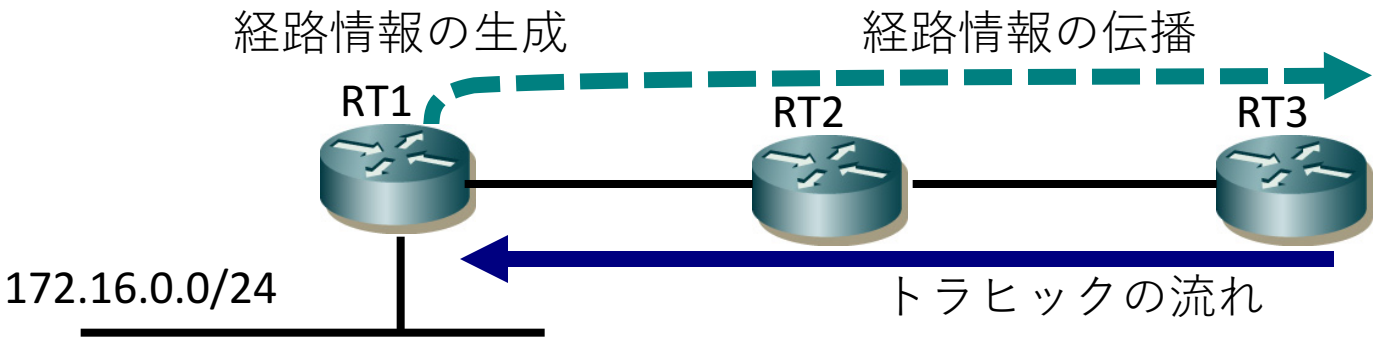
インターネットと動的経路制御

動的経路制御の必要性

- ネットワーク変化を経路情報に反映
  - 自動化 :)
  - ネットワークの拡張が容易
- ISPのバックボーン運用では必須
  - インターネットは変化し続けてる
  - うまく冗長設計すると障害時も綺麗に自動迂回
- 大事なこと
  - プロトコルごとの得手不得手を把握しておく
  - 何を設定しているのか理解しておく

動的経路制御の基本アイディア

- 検知 – ルータがネットワークの変化を検知
- 通知 – 情報を生成し他のルータに伝達
- 構成 – 最適経路で経路テーブルを構成

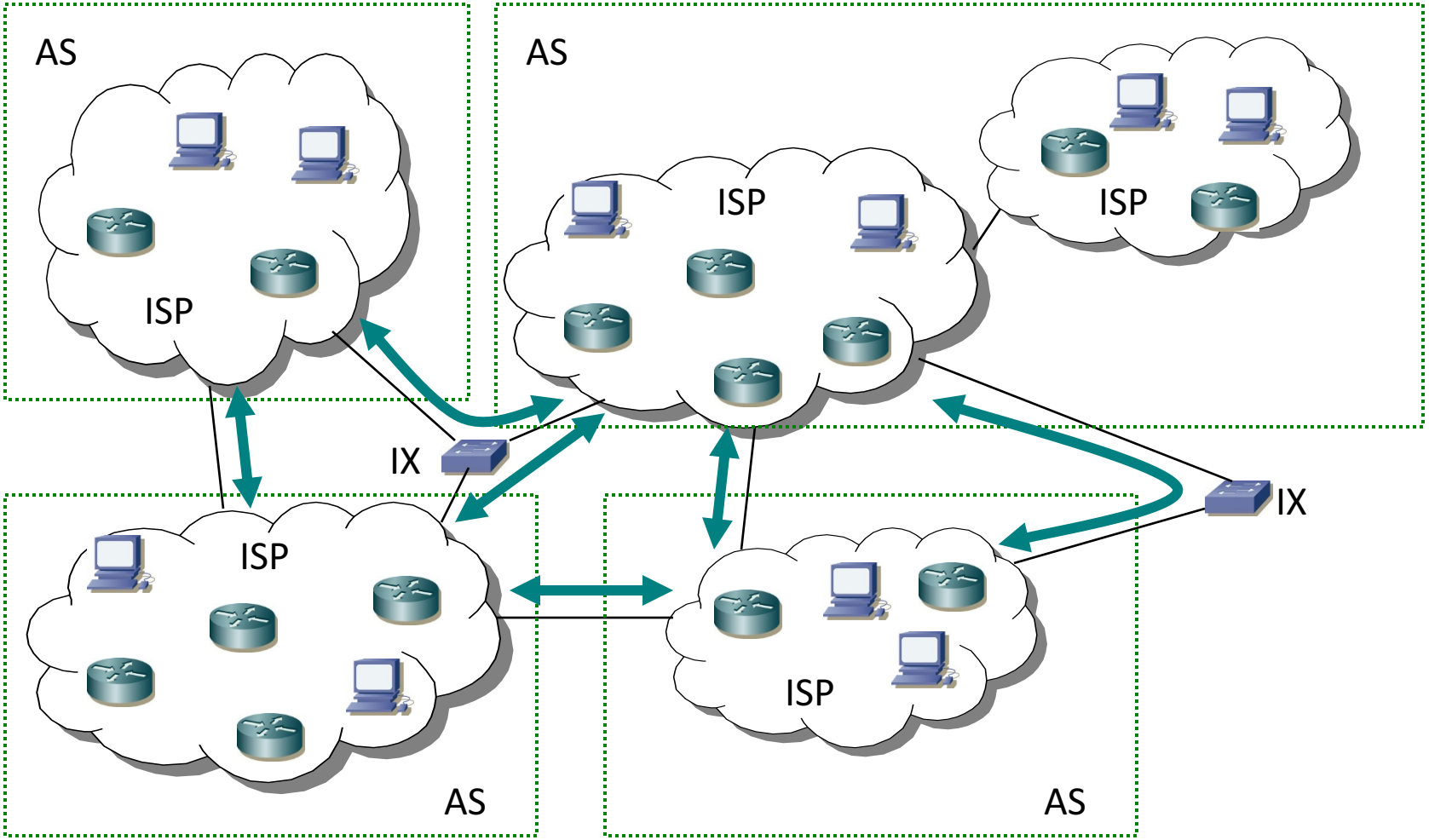


経路情報の伝搬の方向とトラヒックの流れは逆になる

動的経路制御の種類

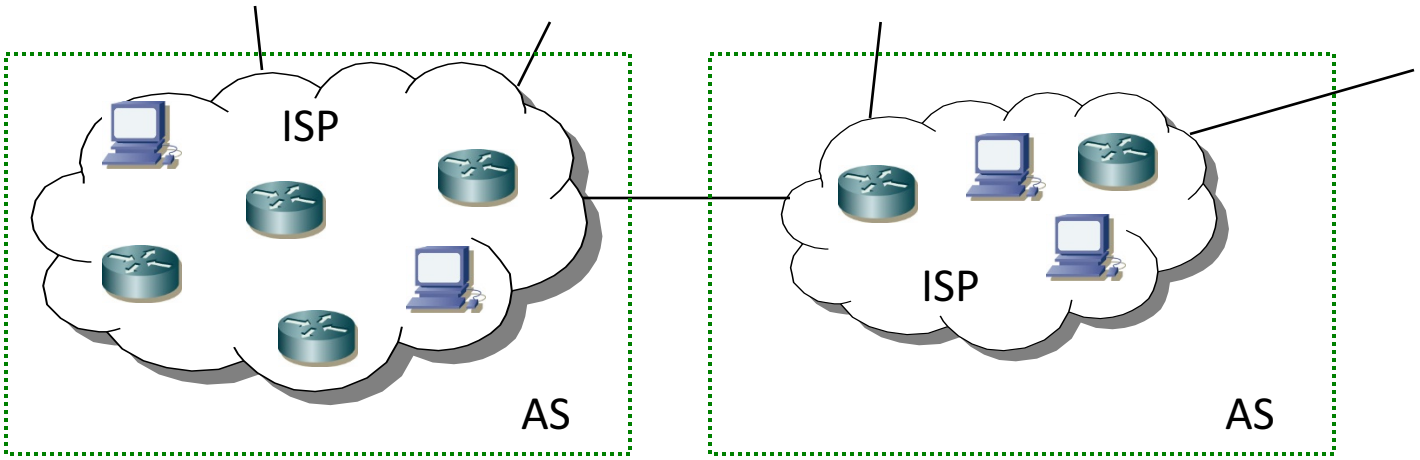
- ディスタンスベクタ(distance vector)
  - RIPなど、距離と方向で運用するプロトコル
- リンクステート(link state)
  - OSPFやIS-ISなど、ルータに繋がっているリンク状態を収集して運用するプロトコル
- パスベクタ(path vector)
  - BGPなど、パス属性と方向で運用するプロトコル

インターネットの構成



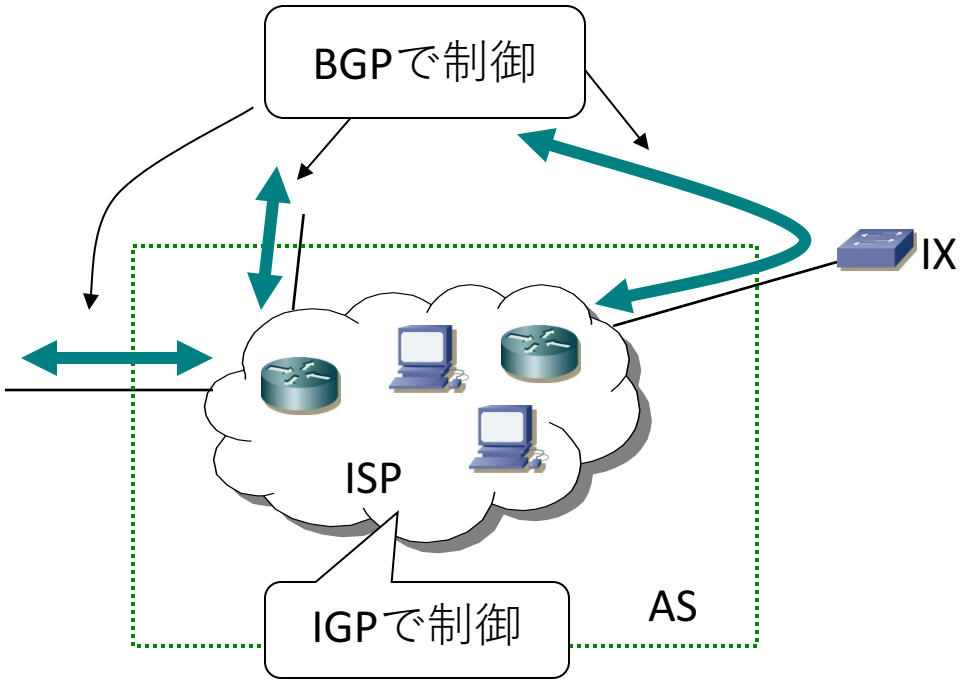
AS

- Autonomous System
- 統一のルーティングポリシのもとで運用されているIPプレフィックスの集まり
- ASの識別子として、インターネットではIRから一意に割り当てられたAS番号を利用する
  - IR: JPNICとかAPNICとかのインターネットレジストリ



IGPとEGP

- IGP
  - OSPF、IS-IS、BGP等
  - AS内
- EGP
  - 事実上BGPのみ
  - AS間



- 最近は網内でトポロジ情報の交換に使うプロトコルをIGPとして認識している場合も多い

ISPでのプロトコルの利用法

- OSPF or IS-IS
  - ネットワークのトポロジ情報
  - 必要最小限の経路で動かす
  - 切断などの障害をいち早く通知、迂回
- BGP
  - その他全ての経路
    - 顧客の経路や他ASからの経路
  - 大規模になっても安心
  - ポリシに基づいて組織間の経路制御が可能

BGP

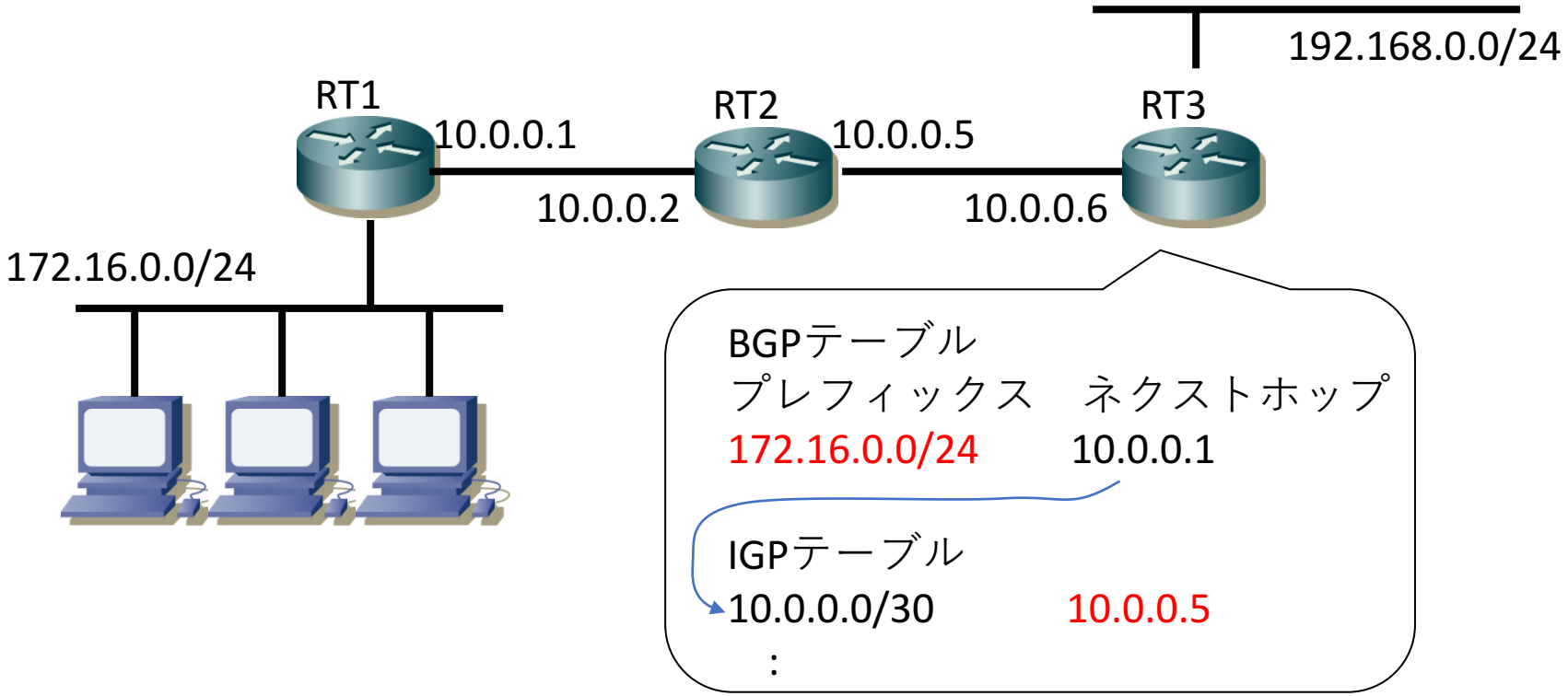
BGP概要

- パスベクタ型プロトコル
  - プレフィックスに付加されたパス属性で経路制御
- AS番号によって組織間、組織内を認識する
- 経路交換にTCPを利用
  - データの到達や再転送はTCP任せ
- 変更があった場合にのみ通知
  - ベスト経路のみを通知する
- 現在のバージョンは４ (BGP4)

BGPの基本アイディア

- 準備
  - 経路交換したいBGPルータとTCPでネイバを構築
  - (ネイバ|ピア|BGPセッション)を張るとも言う
- 通知
  - 最適経路に変更があればUPDATEとしてネイバに広報
  - 受信した経路は幾つかの条件を経て、他のネイバに広報
- 構成
  - 各ルータが受信経路にポリシを適用し、パス情報を元に最適経路を計算して経路情報を構築、パケットを転送

BGPと再帰経路



BGPで学習したネクストホップアドレスをさらに
経路情報で再帰的に探して、ルータが実際に
パケットを送出する隣接ノードを見つけ出す
「172.16.0.0/24宛は10.0.0.5(RT2)にフォワード」

経路優先度

1	NEXT_HOP	NEXT_HOP属性のIPアドレスが到達不可能な経路は無効
2	AS loop	AS Path属性に自身のAS番号が含まれている経路は無効
3	LOCAL_PREF	LOCAL_PREF属性値が大きい経路を優先 (LOCAL_PREF属性が付加されていない場合は、ポリシに依存)
4	AS_PATH	AS_PATH属性に含まれるAS数が少ない経路を優先 (AS_SETタイプは幾つASを含んでも1として数える)
5	ORIGIN	ORIGIN属性の小さい経路を優先 (IGP < EGP < INCOMPLETE)
6	MULTI_EXIT_DISC	同じASからの経路はMED属性値が小さな経路を優先 (MED属性が付加されていない場合は、最小(=0)として扱う)
7	PEER_TYPE	IBGPよりもEBGPで受信した経路が優先
8	NEXT_HOP METRIC	NEXT_HOPへの内部経路コストが小さい経路が優先 (コストが算出できない経路がある場合は、この項目をスキップ)
9	BGP_ID	BGP IDの小さなBGPルータからの経路が優先 (ORIGINATOR_IDがある場合は、これをBGP IDとして扱う)
10	CLUSTER_LIST	CLUSTER_LISTの短い経路が優先
11	PEER_ADDRESS	ピアアドレスの小さなBGPルータからの経路を優先

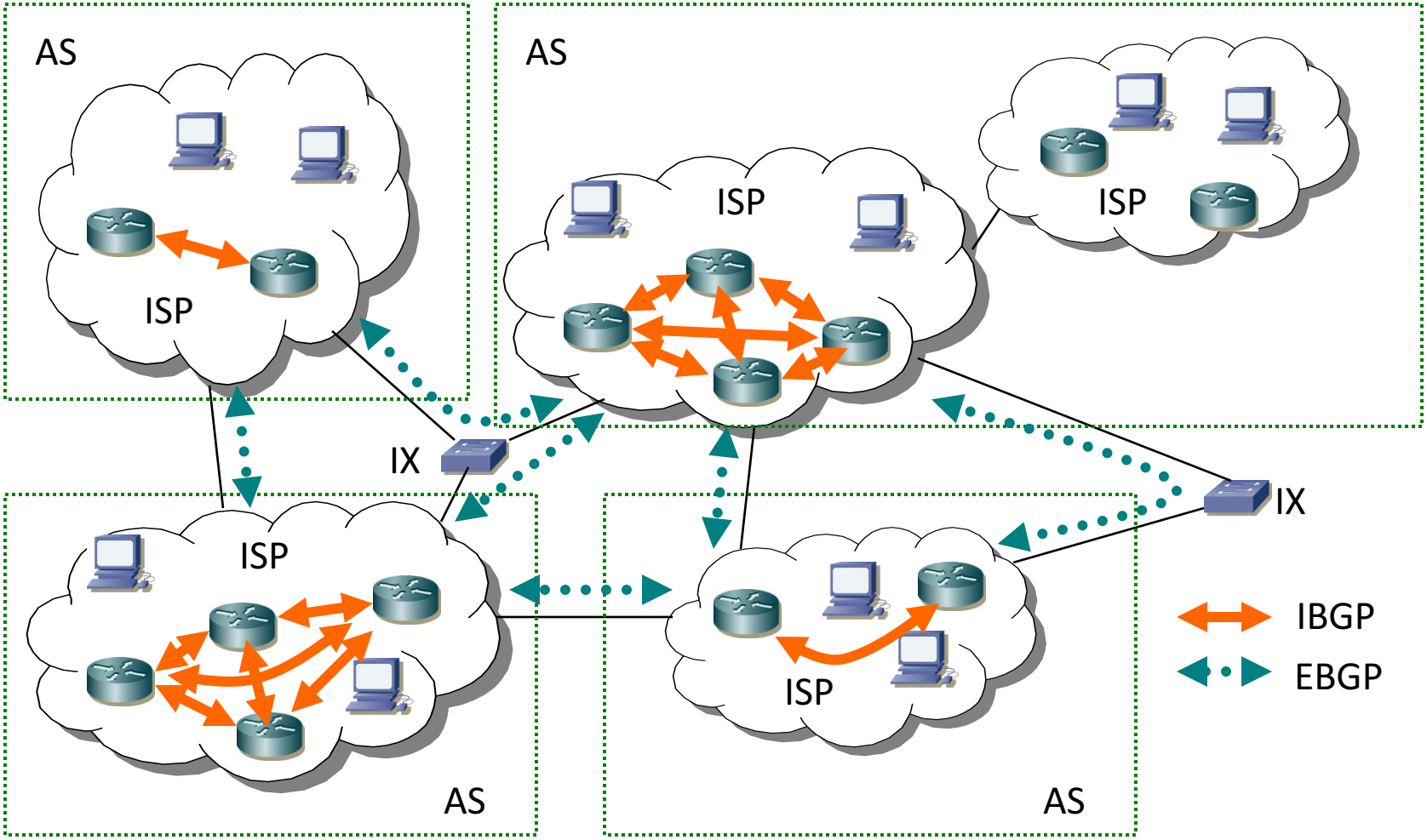
BGP RFCs

- 基本
  - [RFC4271] A Border Gateway Protocol 4 (BGP-4)
- この他にもいっぱい
  - [RFC1997] BGP Communities Attribute
  - [RFC3065] AS Confederations for BGP
  - [RFC4451] BGP MED Considerations
  - [RFC4456] BGP Route Reflection
  - [RFC6286] AS-Wide Unique BGP Identifier for BGP-4
  - [RFC6793] BGP Support for Four-Octet AS Number Space
  - [RFC7606] Codification of AS 0 Processing
  - [RFC8092] BGP Large Communities Attribute
  - [RFC8212] Default EBGP Route Propagation Behavior without Policies

BGP用語

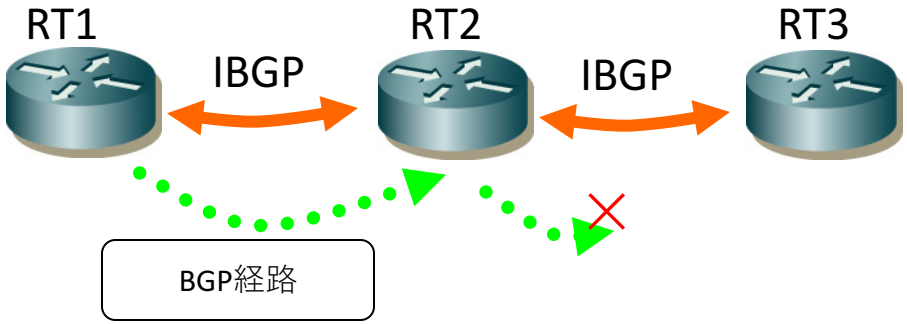
- BGP ID
  - ルータを識別する32bitの数値
    - AS内で一意である必要がある [RFC6286]
  - インタフェースの何れかのIPアドレスから選ばれる
  - 変更が発生しないようにloopbackインタフェースに付与したIPアドレスを利用する場合が多い
- NLRI
  - Network Layer Reachability Information
  - ネットワーク層到達可能性情報
  - prefixで示される宛先のこと

BGPの世界



IBGP(Internal BGP)

- 同じAS内でのBGP接続
- IBGPで受信した経路は他のIBGPルータに広報されない
  - 全ての経路を伝えるには、AS内の全BGPルータがfull-meshでIBGPを張る必要がある

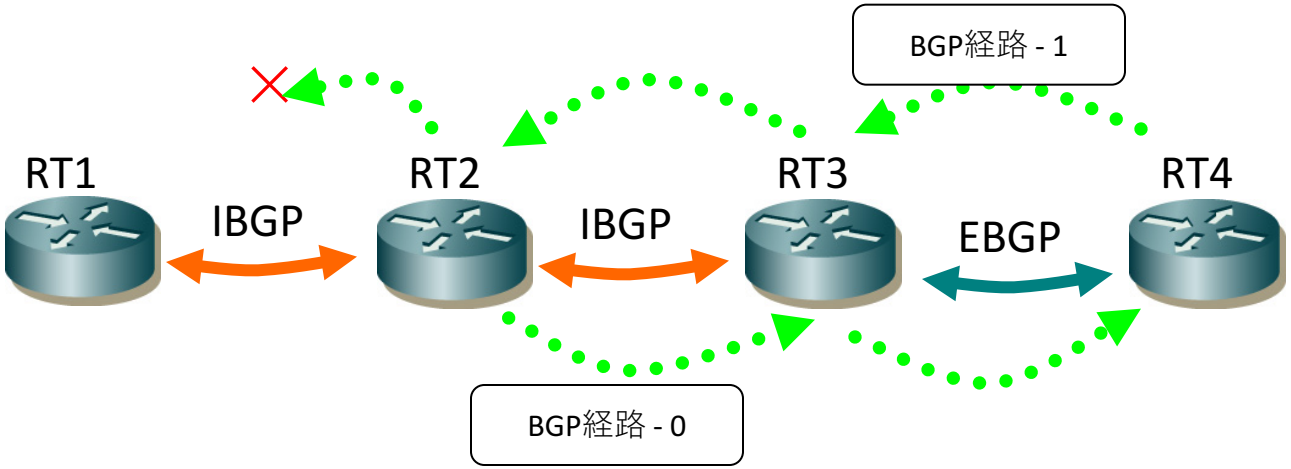


IBGPの基本

- 通常、ループバックインタフェースを利用
  - どれか物理インタフェースが生きてたら到達可能
  - IGPでループバック間の到達性を確保
- 経路情報をそのまま伝える
  - 基本的にパス属性を操作しない
    - MEDやLocal Preference等の優先度、ネクストホップ
  - 下手にいじると経路ループする
- 基本的に全てを広報し、全てを受け取る
  - 特段の理由が無ければ経路フィルタしない

EBGP(External BGP)

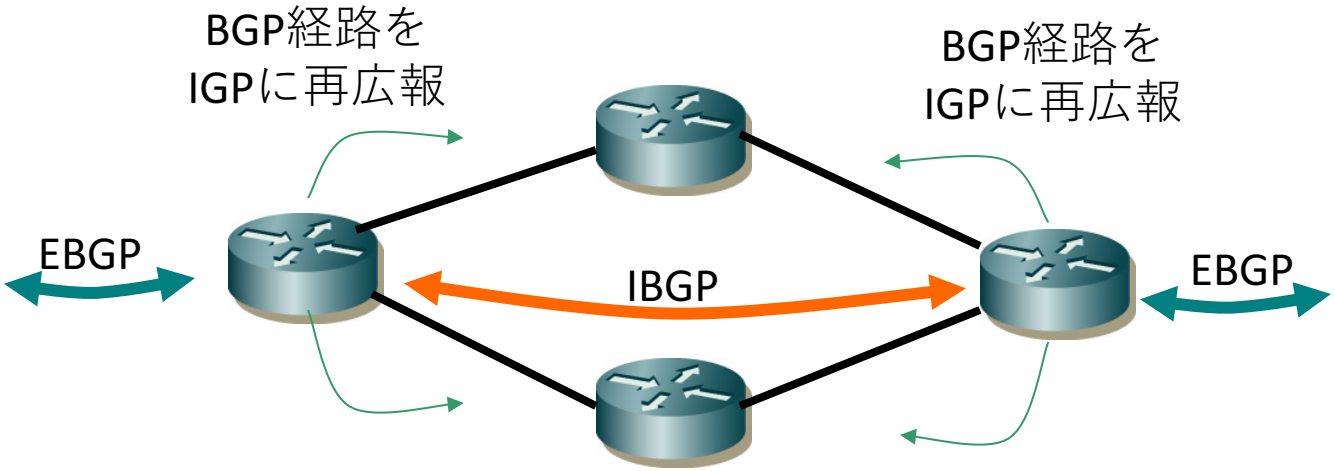
- 異なるASとのBGP接続
- EBGPから受信した経路は、他のBGPルータに広報する
  - IBGPから受信した経路もEBGPには広報する



EBGPの基本

- 通常、物理接続してるインターフェースで張る
- ポリシの実装をするならここ
  - 受信のポリシ
    - 不要な経路のフィルタやタグ付け
    - MEDやlocal preferenceによる優先制御
  - 広報のポリシ
    - 不要な経路のフィルタと必要な経路の広報
    - MEDやprependによる優先制御
- ポリシが違うところは網内でもEBGPにした方が便利
  - Private AS番号の利用など (64512-65534)

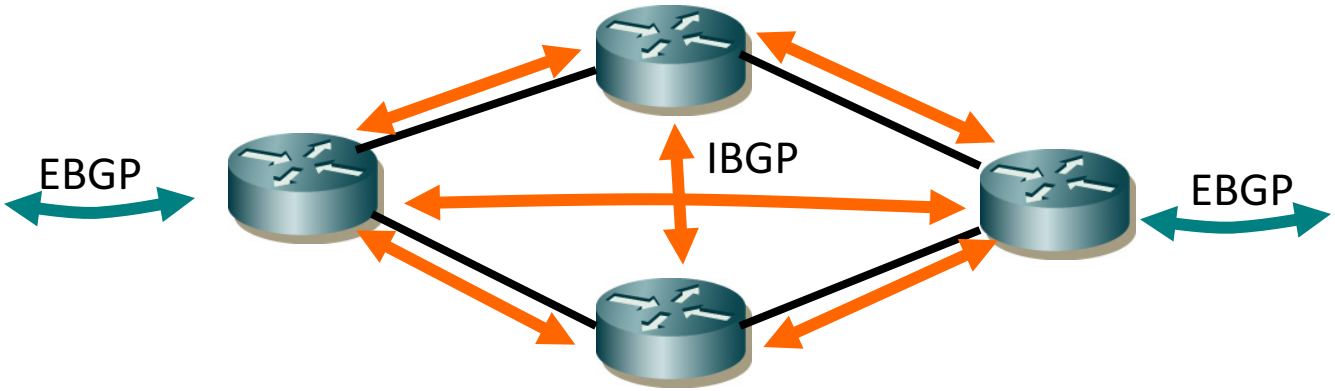
BGPのいにしえのモデル



- EBGPを張るルータのみがBGPルータとなる
- BGP経路をIGP(OSPFやIS-IS)に再広報してAS内部はIGPで経路制御
  - 内部にIGPのみのトラヒック中継ルータが居るため、bgp synchronizationが必要だった

・・・経路数が増大すると破綻

今時のBGPモデル



- 主要なルータは全てBGPルータ
- IGPはトポロジと最低限の経路を運び、BGPでその他の全ての経路を運ぶ

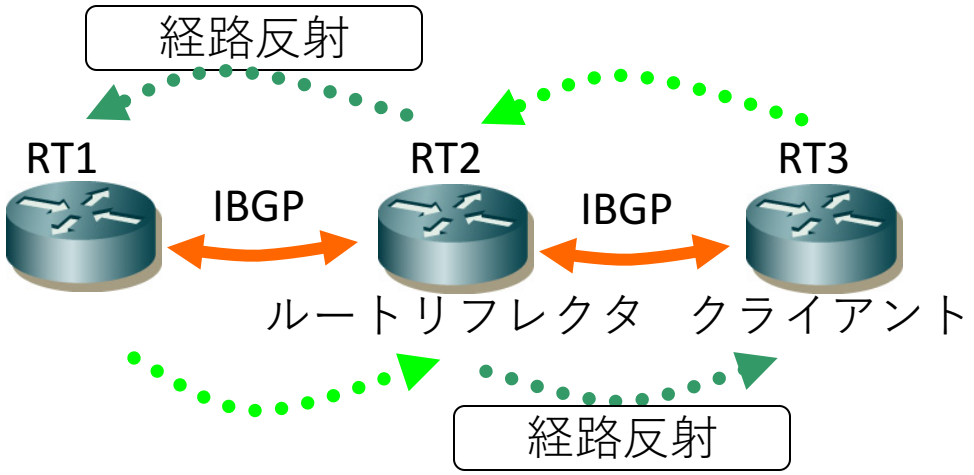
・・・IBGP接続の増大

IBGP full-mesh n*(n-1)/2

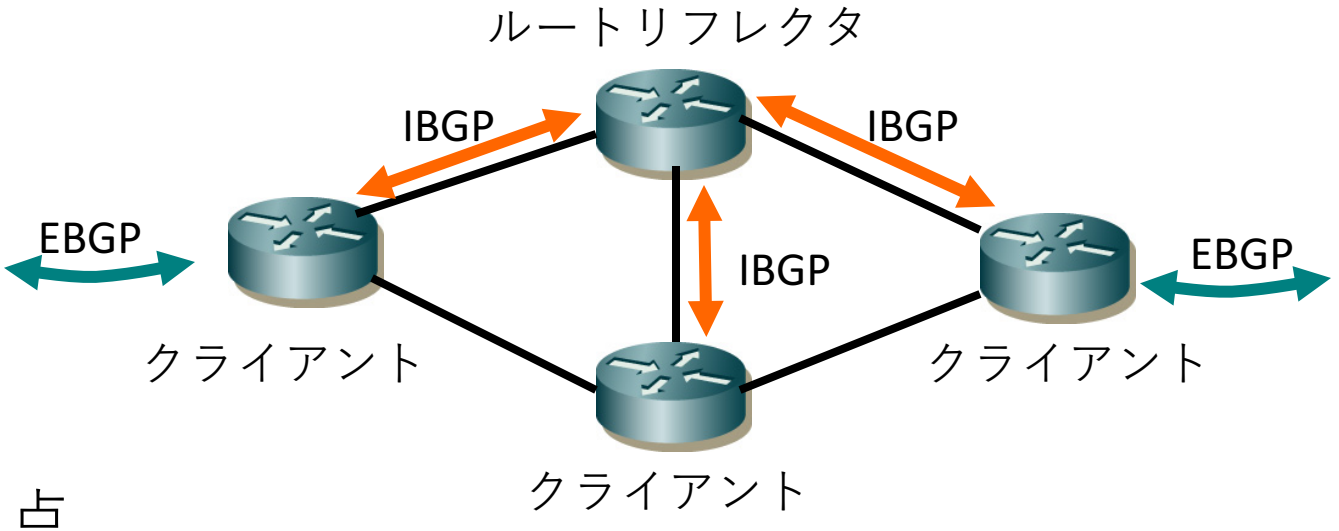
- AS内にBGPルータが増える毎にIBGP接続が増大していく
  - 20台目のBGPルータが接続すると１9接続追加
  - ルータリソースの問題、設定負荷の問題
- 解決策の模索
  - [RFC4456] ルートリフレクタ
  - [RFC3065] コンフェデレーション
  - 気にせずリソースを強大にする
  - ルータを減らす

ルートリフレクタ

- IBGPで受信した経路の転送ルールを変更
- ルートリフレクタの機能
  - BGP接続ごとに設定される
  - クライアント以外のIBGPで受信した経路をクライアントに送信
  - クライアントから受信した経路を他のIBGPルータに送信
- ベスト経路のみを広報するルールは変わらない



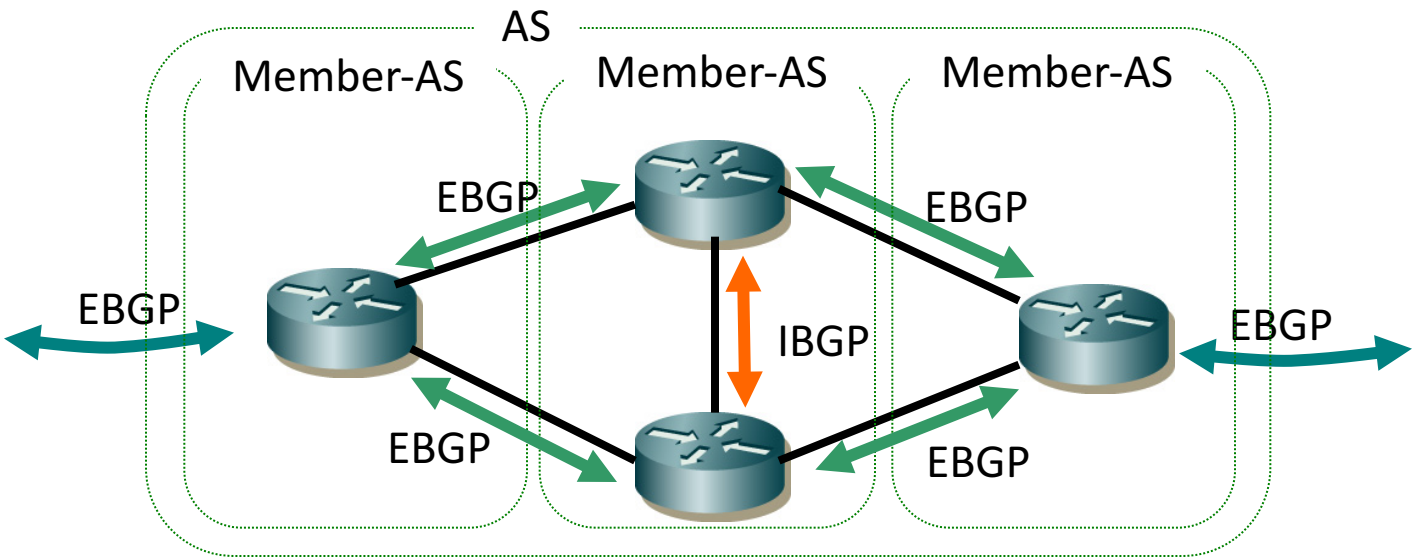
ルートリフレクタの利点と欠点



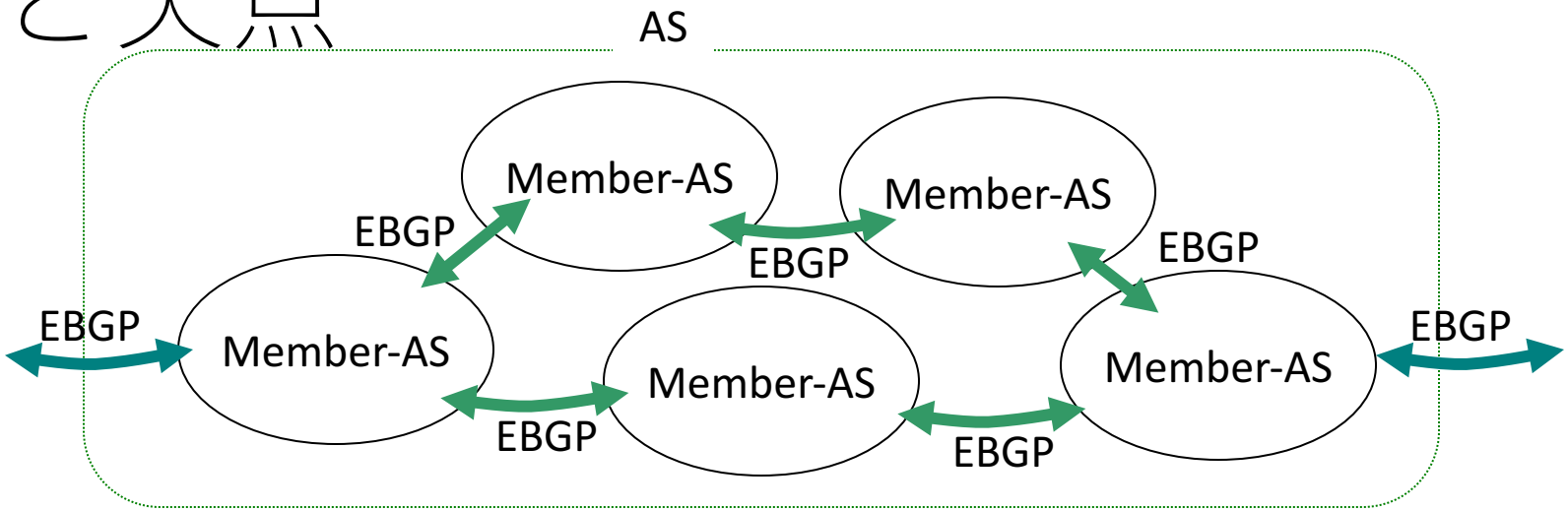
- 利点
  - IBGP接続数が削減できる
  - 比較的容易に導入できる
- 欠点
  - 経路削除時に、UPDATEが増える可能性がある
  - 経路情報が隠蔽されるため最適ではない経路を選ぶ可能性がある
    - リフレクタの階層はできるだけ物理トポロジに合わせるべし！

コンフェデレーション

- 外部からは一つのASのままだが、内部を複数のメンバASで構成する
- メンバAS間のBGP接続はEBGPに似た挙動をする
- メンバASにはプライベートASを使うのが一般的



コンフェデレーションの利点と欠点



- 利点
  - IBGP接続数が削減できる
  - 管理区分を分けられる
- 欠点
  - 経路削除時にUPDATEが増える可能性がある
  - 経路情報が隠蔽されるため最適ではない経路を選ぶかもしれない

BGP運用

相互接続とトラヒック制御

ASの運用

- 到達性の確保
  - 何はともあれ、到達性が重要
  - 大抵、どこかからtransitを購入して保険をかける
- トラヒックの制御
  - BGPは回線の空き具合を気にしない
  - 回線や設備はそんなに柔軟に変えられない
    - ホントは需要に応じて増強するのが一番きれい
  - それでも対処しなきゃいけない事案は出てくる

基本的なお作法

- PAブロックは割り振られたサイズで広報
  - 細かい経路やprivate AS&アドレス等を漏らさない
  - 広報する経路に責任をもつ
- 全ての接続点で一貫した経路広報
  - 相互接続しているASには、どの接続点でも同一の経路を広報
- 何らかトラヒック制御しようとする場合には、事前に相互接続先と相談

経路制御ポリシ

- あった方が運用に一貫性が出て良い
  - 意図しない経路制御を防止できる
- ポリシを考えるもと
  - 提供したい通信、自由度
  - トラヒック制御
  - 自身の経路制御の防御

対外接続

- EBGPで接続
- 他のASと経路交換
  - トランジットしてもらって到達性の確保
  - ピア（相互接続）で独自の接続性の向上
- 接続方法
  - 相互接続に合意
  - 専用回線やIXで接続

IXでEBGP（パブリックピア）

- お互いに同じIXに居る事の確認
- お互いのIPアドレスの通知
  - IXで提供される個別セッションサービスやVLANサービス等を利用する場合、IPアドレスの手配が必要な場合もある
- ネイバの設定

専用回線でEBGP（プライベートピア）

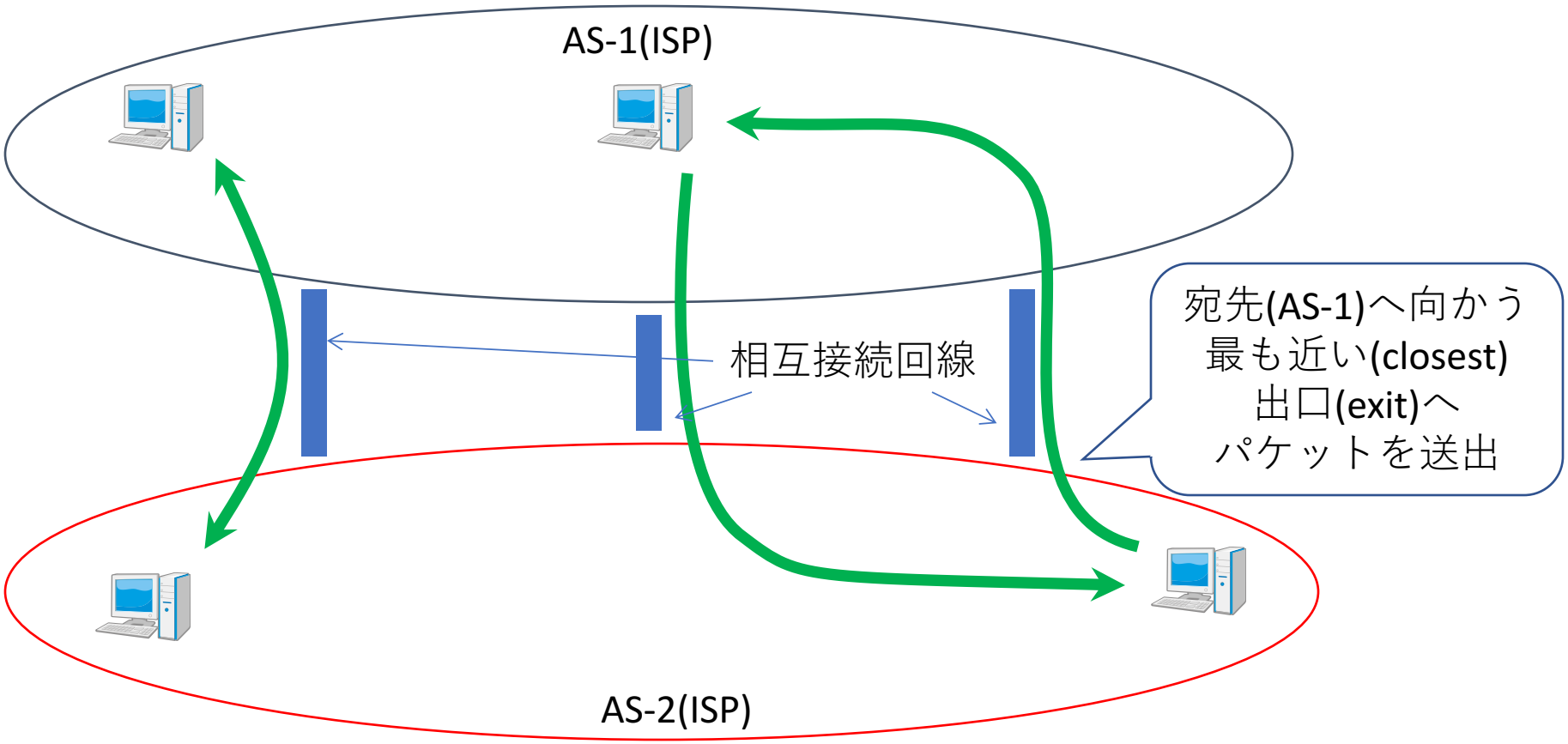
- インタフェースの合意
  - 速度や種別
- 必要に応じて回線手配と費用分担の調整
  - 構内回線や回線サービスなど
- その回線で利用するIPアドレス手配
  - どちらかの組織から持ち出しになる場合が多い
  - /30or/31, /64or/127
- ネイバの設定

あるASと複数拠点で相互接続

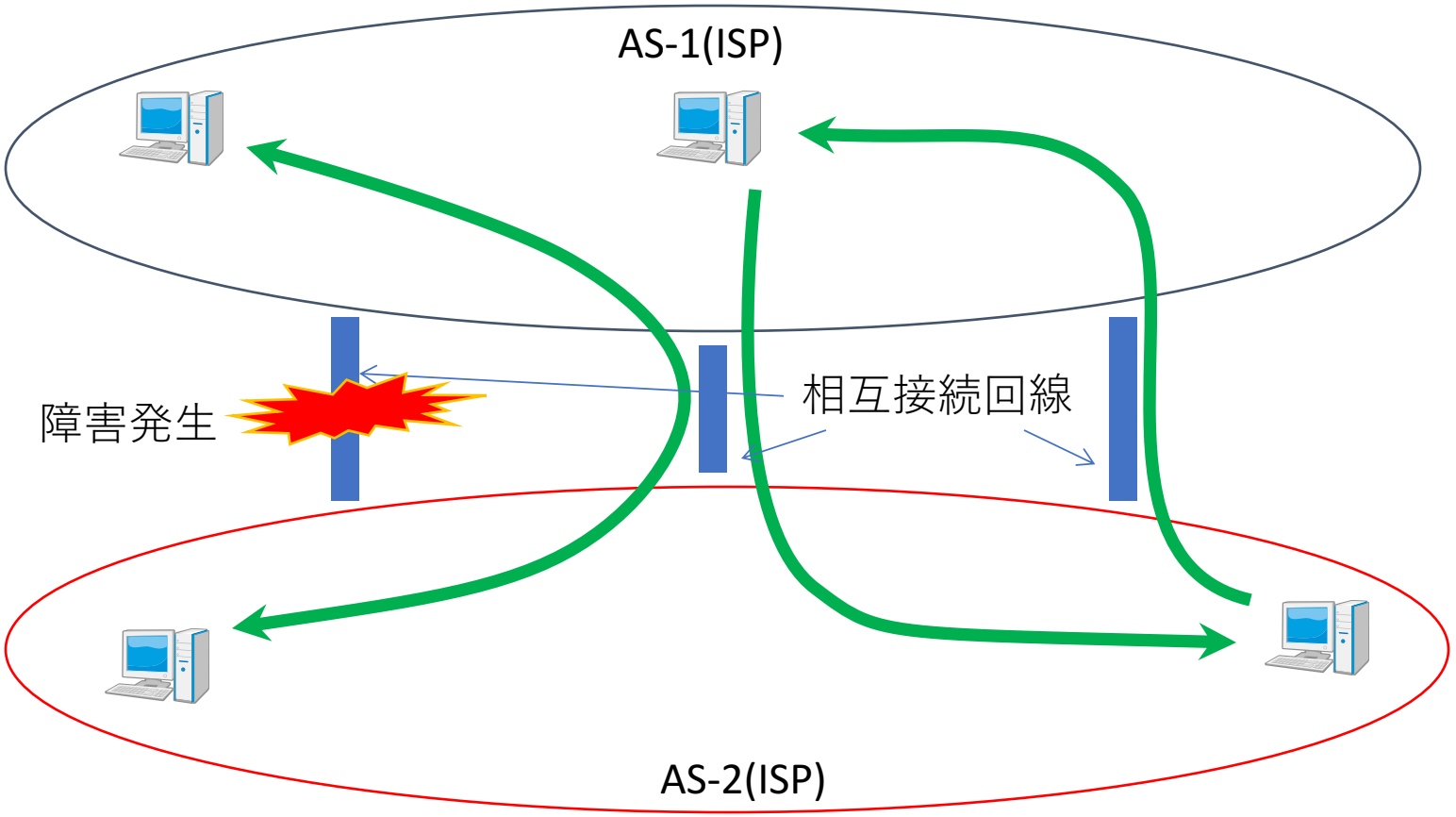
- トラヒック制御を合意しておく必要がある
- **お互いに相手ネットワークの事は分からない**
- **最適な経路を選ぶには、宛先に近いネットワークに素早くパケットを渡せば良い**

  ＝ **closest exit(クローゼスト イグジット)**
  - BGPの素直な利用方法
  - 世界のISPが標準的に採用しているポリシ

closest exit



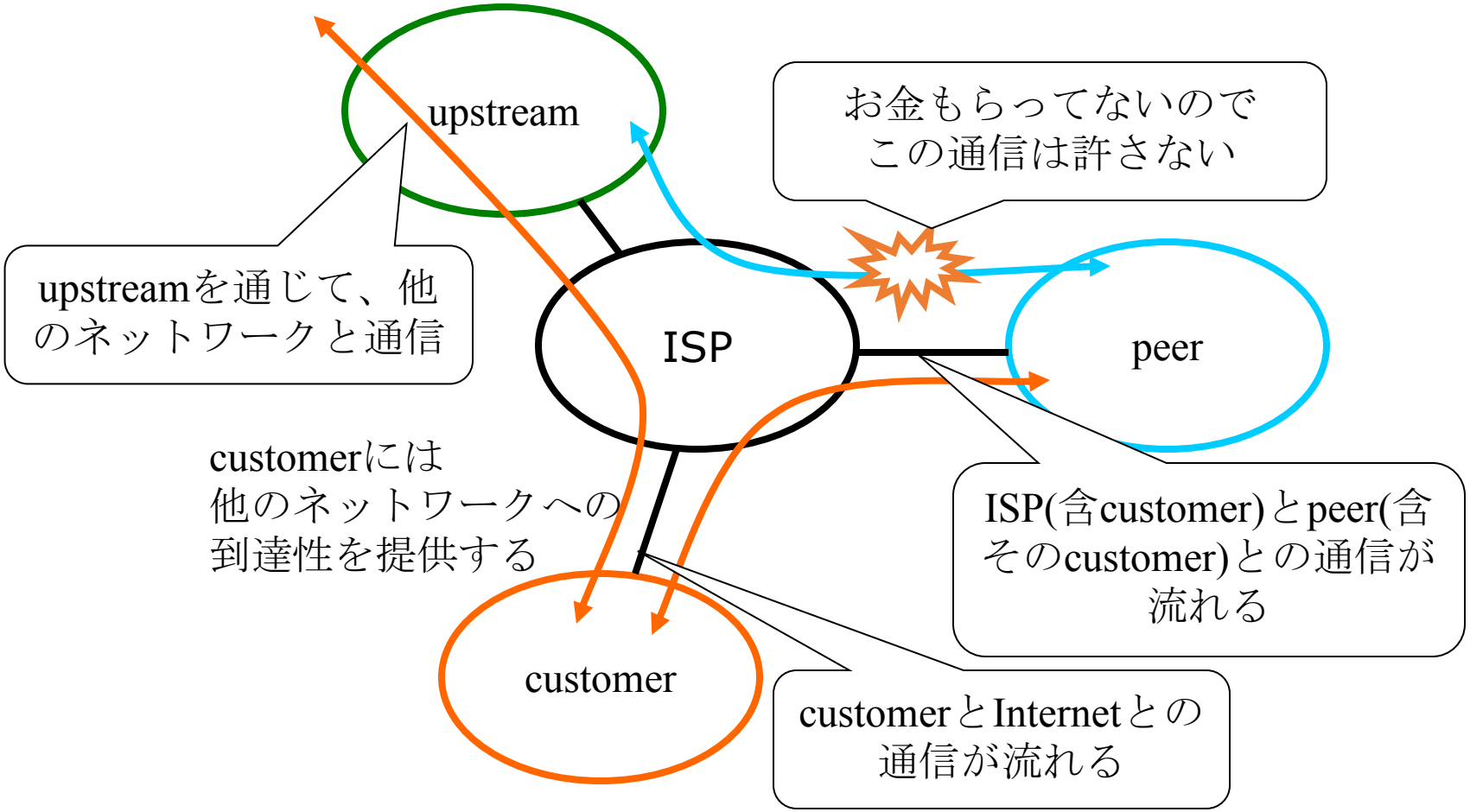
障害発生時のclosest exit



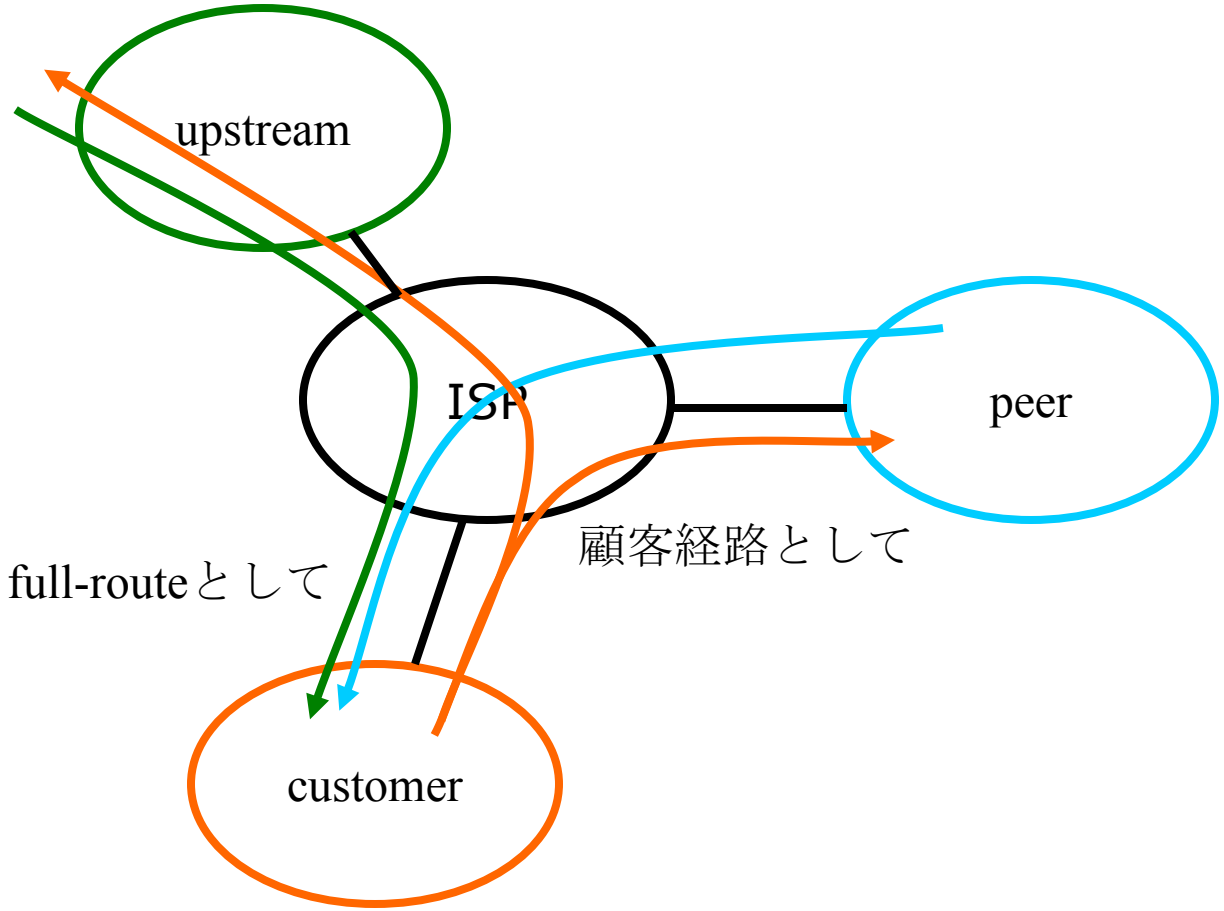
closest exitの特長

- 簡単なポリシで最適な経路を選びうる
  - BGPはclosest exitを前提として設計されている
  - ネクストホップへのIGPメトリックで制御できる
- 相互接続ポイントが増えても、同じ経路制御ポリシのままで運用できる
  - 拡張性に優れる
  - 特別な設計が必要ない

顧客に提供したい通信


upstream
お金もらってないので
この通信は許さない
upstreamを通じて、他
のネットワークと通信
ISP
peer
customerには
他のネットワークへの
到達性を提供する
ISP(含customer)とpeer(含
そのcustomer)との通信が
流れる
customer
customerとInternetとの
通信が流れる

対応する経路広報の流れ



トランジットの実装方法

- 普通はBGP community
  - 顧客経路の受信時にtransit用のTAG付け
    - 顧客からの経路受信時に経路フィルタの併用が必須
  - 外部にはtransit用TAGがついた経路のみを広報
- 小規模なら経路フィルタでも実現可能
  - トランジットする経路をprefixフィルタで管理
  - 外部に広報するときに、このフィルタを適用
    - 顧客から広報されなくてもtransitしてしまうかも

受信経路の基本的な優先制御

- 経路優先度
  - customer ＞ peer　≧ transit
  - ほとんどのASが、LOCAL_PREFを使って実装
- customer経路は優先
  - 顧客にtransitを提供するために優先
    - BGPはベスト経路しか広報しないよね
    - 他から広報された経路が優先されちゃうとtransitできない
- peerとtransitから受信した経路の優先度は低め
  - 少なくともcustomerからの経路よりも低め

LOCAL_PREF

- AS内での経路優先度を示す優先度
- 経路受信時に明示的に設定しておくのが吉

接続相手	設定するLOCAL_PREF例
customer	200
peer	100
upstream	90

- LOCAL_PREFは優先度として強すぎるので、これ以外の細かな制御には向かない
  - 使う場合には相当強い意図と精緻な考察が必要

MED

- 隣接ASとの距離を示す値
  - あるASと複数接続がある場合に、それぞれの優先度を設定
  - eBGPで経路の広報元が値を設定しても良いし、受信側で適当な値を設定しても良い
  - バックアップ経路の指定や、拠点やIXなど狭い範囲での経路選択に利用される場合が多い
- 機器によって実装が違う場合があるので注意
  - 設定してなければ0として扱う (RFC4271)
    - MEDを利用した制御を行うなら、何らの値を明示的に設定するべし

MEDの評価

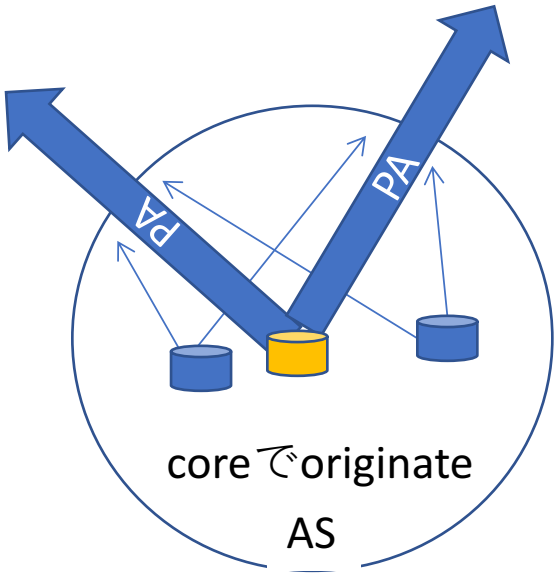
- non-deterministic-med (cisco default)
  - 受信経路の到着順序に従って最適経路を選択する
  - MEDの値が思い通りに評価されないことがあるため、普通使わない
- deterministic-med (juniper default)
  - 同一ASから受信した経路同士を先に比較して、その後再度最適経路を選択する
  - みんな使ってる
- always-compare-med
  - 異なるASから受信した経路でもMEDの値を評価する

受信経路のMED

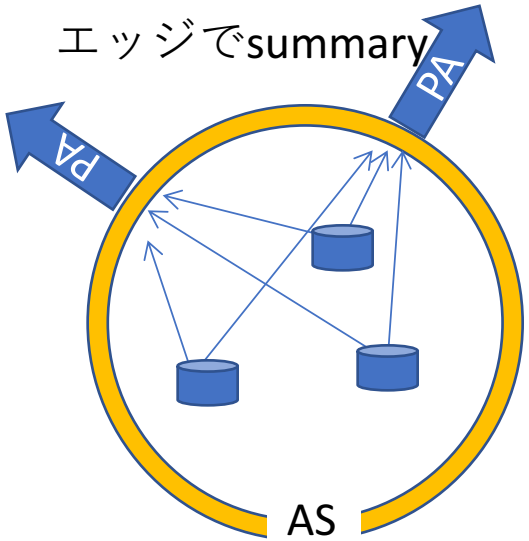
- 受信時に上書き
  - 制御を提供しない場合
  - upstreamやpeerからの経路等
- 受信したMEDをそのまま利用
  - 制御を提供する場合
  - customerやpeerからの経路等

経路の生成

1. 内部のルータでnull向けstatic経路から生成
2. EBGPルータでsummary経路として生成



内部で生成方式



エッジで生成方式

経路の生成：内部で生成方式

- 想定障害
  - 経路生成ルータでの障害や到達性障害
  - ネットワーク分断
  - -> 経路をいかに広報し続けるかが課題
- 対策案
  - 複数台での経路生成
  - 内部ネットワークで頑強な接続性を保持しているルータで経路生成
  - なるべく実ネットワーク利用の近傍で経路生成

経路の生成：エッジで生成方式

- 想定障害
  - 経路生成ルータの孤立
  - ネットワーク分断
  - -> 障害時にうまく広報を止めることが課題
- 対策案
  - 障害時の影響を小さくするため、地域ごとに利用するprefixを分ける
  - 他のIGPとBGPで運ぶ経路情報を棲み分ける
- 他ASと隣接する全EBGPルータで設定が必要
  - 顧客向けやピア収容ルータで忘れないように

customer持ち込みのPI経路生成

- 回線向けのstatic経路から生成
  - 回線が落ちると経路が消える
    - multiple origin ASしている場合には必須機能
  - 回線がflapするとdampeningペナルティがあるかも
- null向けstatic経路から生成
  - customerとの回線が落ちても経路は消えない
  - BGP的には安定

経路制御を守る

- 不正な経路情報の流通を防ぐ
- 意図しない経路制御状態を防ぐ

- BGPはTCPで隣接関係を構築
  - md5で保護
- 経路情報の方があれこれ危ない
  - 色々な経路を送受信する必要がある

経路フィルタ

- トランジットする経路は厳密にprefixフィルタ
  - customerからは広報経路を事前に通知してもらう
  - customerからの経路受信時にフィルタを適用
- ピアとの接続ではできる限りフィルタ
  - 基本的に必要ない経路は受け取らない
    - bogon経路
    - 自身の経路
  - 広報される経路数で異常を検知
    - max-prefix

BGP community

- 経路にタグの様な情報を付加して、これにより様々な処理を実装している
  - transit経路の識別
- 内部処理に利用しているBGP communityを守る必要がある
  - BGP Large communities [RFC8092] を使っている場合も同様
- 経路受信時に削除または上書き

BGP NEXT_HOP

- nexthopの解決用経路は死守すべし
  - 絶対に外部から受け取ってはいけない
  - more specific経路にも注意
  - 全EBGPで確実にprefixフィルタを実装
- BGP NEXT_HOPになりうるIPアドレス
  - 経路を生成しているルータ
  - 相互接続アドレス
    - IX
    - プライベートピア

トラヒック増加対応

- １インタフェースの上限速度がある
  - 今のところ、10GEが標準的
  - 100GEがようやく使われ始めたけどまだ高い
- ISP間、ルータ間は10G以上のトラヒック
  - 実効帯域を何とかして増やしたい
  - しかも、冗長構成は必須
- 次のインタフェースがあんまりない
  - 400Gbps?
  - 中途半端なので、多分100Gを束ねて使う方が楽

link aggregation

- 回線を束ねて、論理的に一つの回線に見せる
  - 複数の回線を束ねられる
  - 束ねられる回線数には実装により、上限あり
- 回線が切れると迂回路に回る
  - 用意した帯域の半分程度しか利用できない
  - 実トラヒック量が許すなら、構成回線が全て切れるまで断と見なさない運用も可能

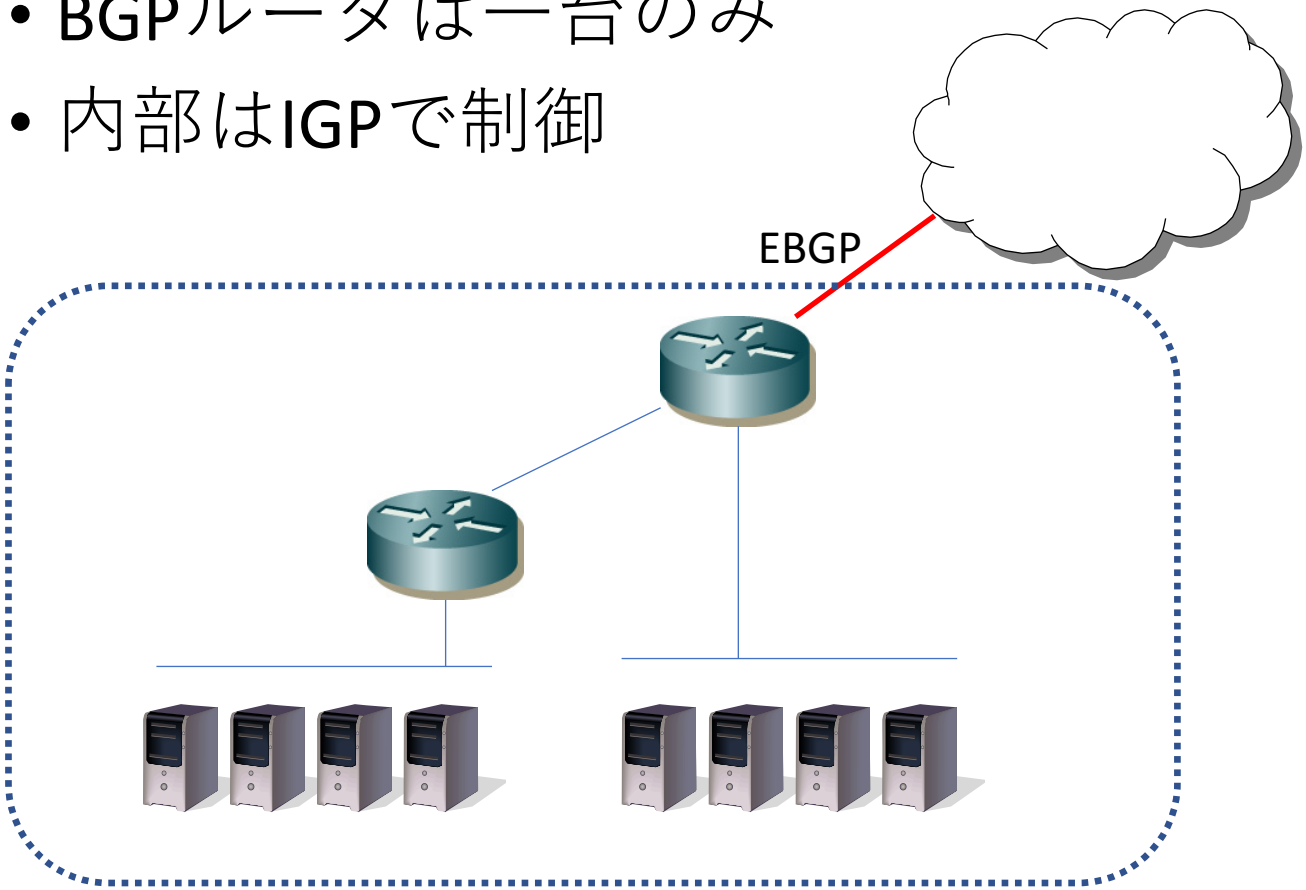
multipath

- OSPF Multipath
  - ISP(AS)内での分散に利用可能
  - 標準技術
- BGP Multipath
  - 非標準技術だが、多くのベンダが採用
  - 構成をきちんと組めば、ISP(AS)間にも有効
- 帯域の利用効率が良い
  - IPアドレスやポート番号にルータ毎のsaltを加えたhashでフォワードする回線を選ぶ場合が多い
    - flowベースで同じ回線を通る
    - 多段にmultipathしてもそこそこ分散するように

BGP構成例

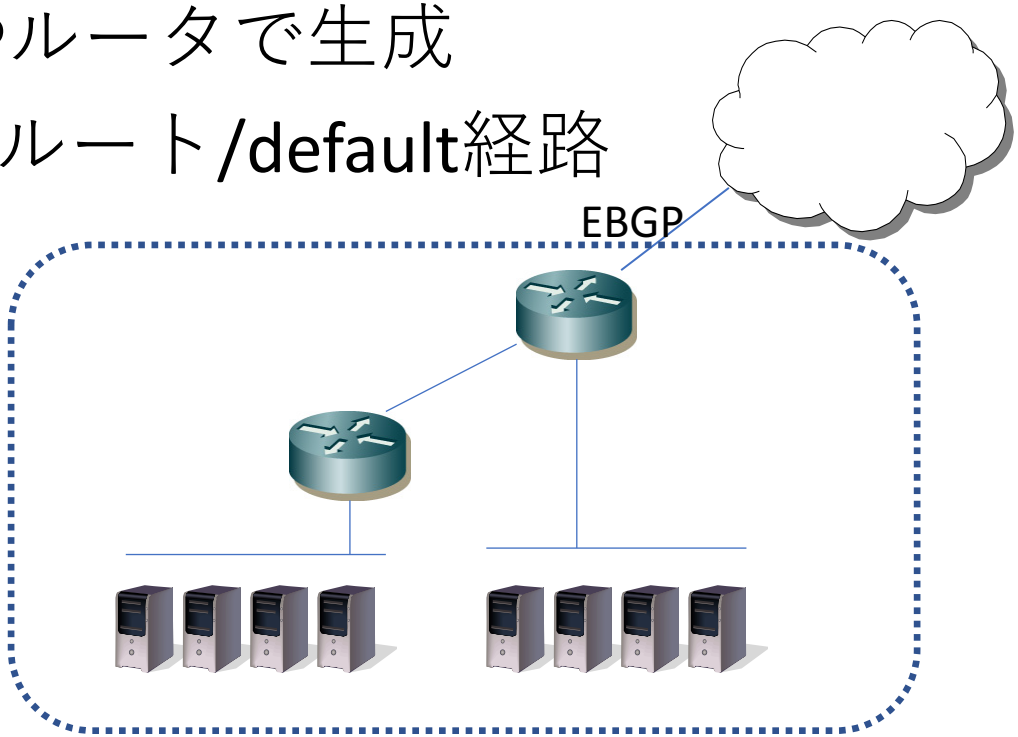
1.上流への接続のみな場合

- BGPルータは一台のみ
- 内部はIGPで制御



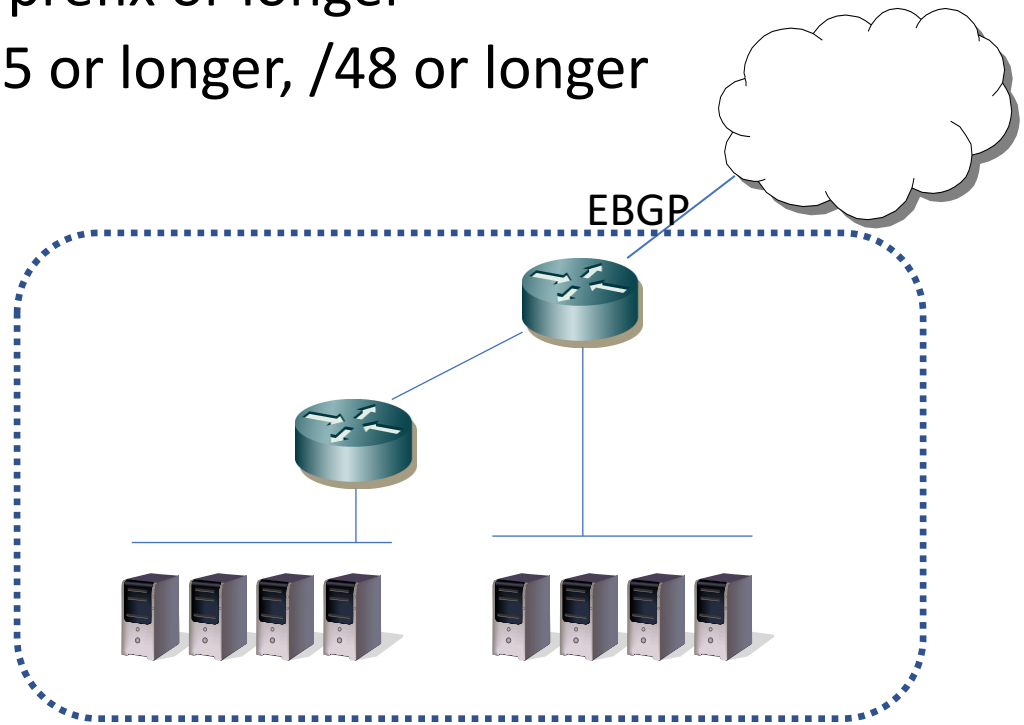
1.上流のみ: 考慮点

- BGPルータが一台なので簡単
- p2pアドレスは上流から割り当て
- 経路生成: EBGPルータで生成
- 受信経路: フルルート/default経路



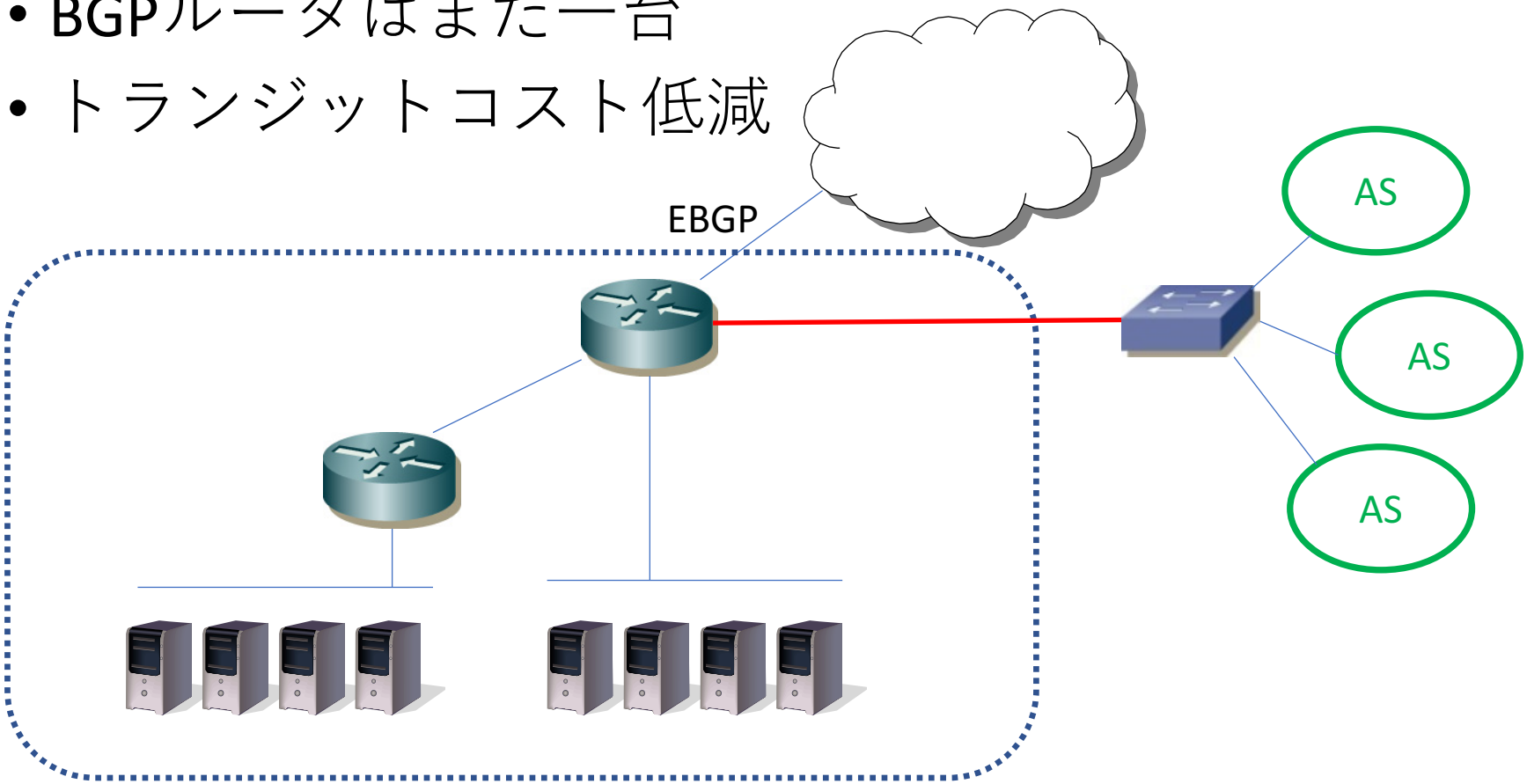
1.上流のみ: 受信経路フィルタ

- 上流からの受信経路フィルタ
  - 自身のprefix or longer
  - private/special prefix or longer
  - 細かい経路 /25 or longer, /48 or longer
- max-prefix?
- max-as-path?



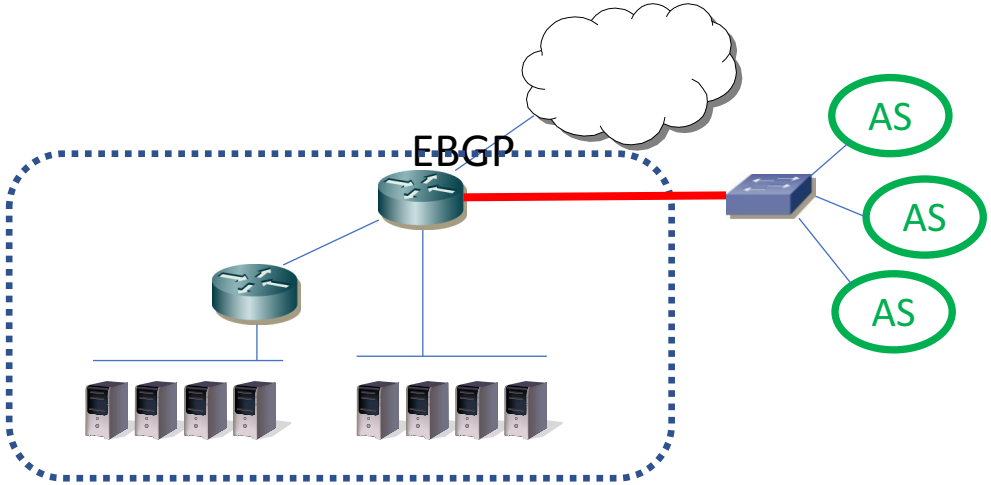
2. IXに接続追加した場合

- BGPルータはまだ一台
- トランジットコスト低減



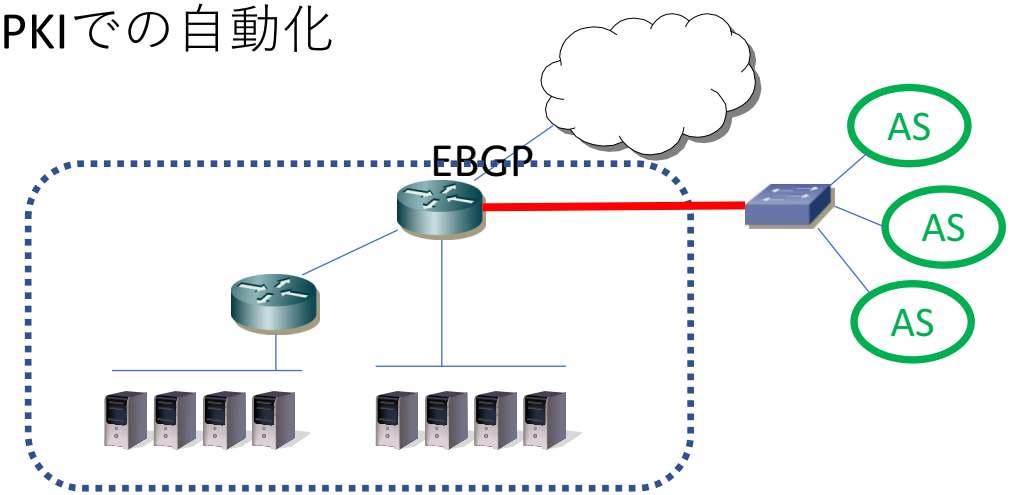
2. IX接続追加: 考慮点

- 個別に他ASと接続交渉が必要
- IXで利用するIPアドレスはIXP事業者から割当て
- 広報する経路は自身のprefixのみ
- 受信する経路は他AS次第



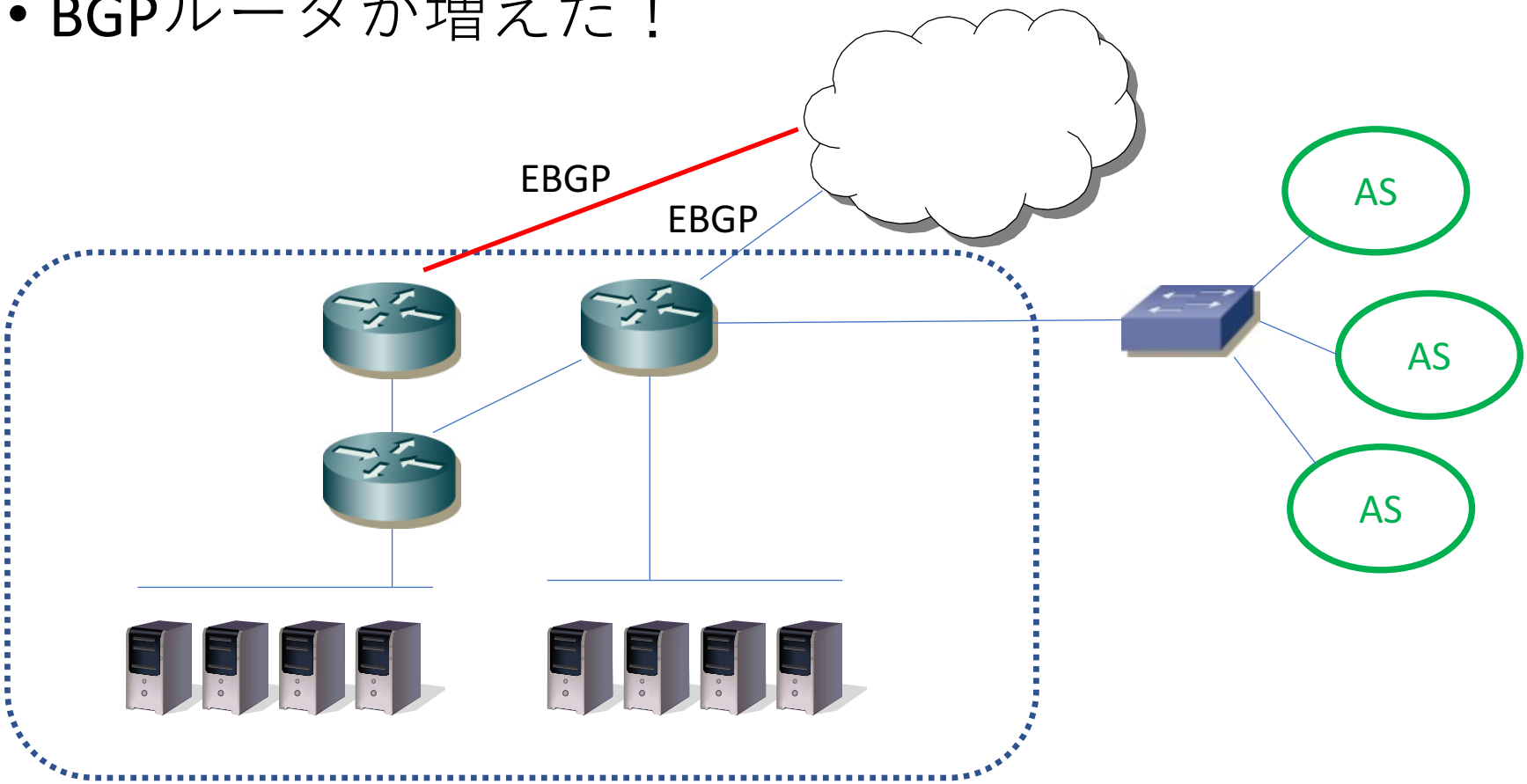
2. IX接続追加: 受信経路フィルタ

- 他ASからの受信経路フィルタ
  - 上流に適用している受信フィルタに加えて
    - 厳密なprefixフィルタ
    - AS_PATHフィルタ
  - でも運用は大変
    - IRRやRPKIでの自動化
    - 諦め?



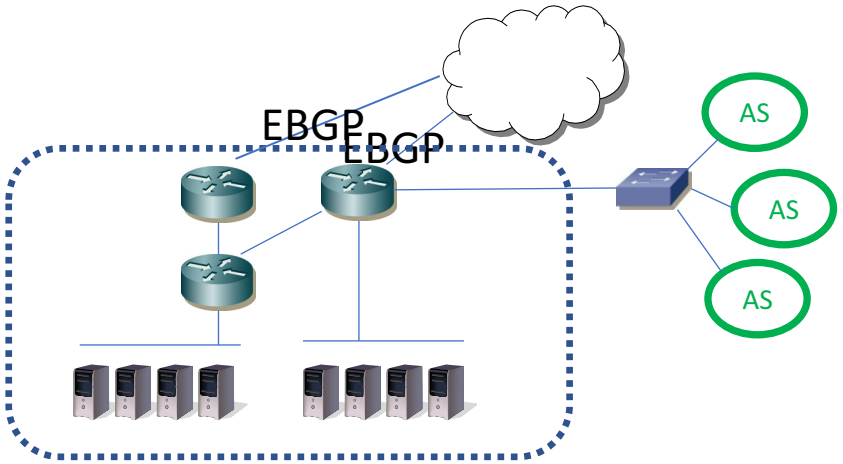
3. 冗長性確保のため上流追加

- BGPルータが増えた！



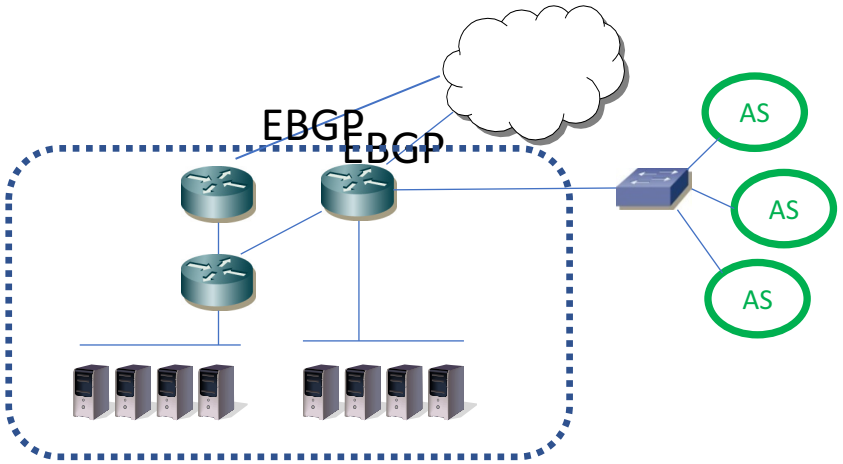
3. 上流追加: 考慮点

- BGP任せの冗長性確保か、がっつり負荷分散か
- 受信した経路のBGP NEXT_HOPをどうするか
- IBGPをどこまで伸ばすか
- 自身の経路広報をどのルータで生成するか



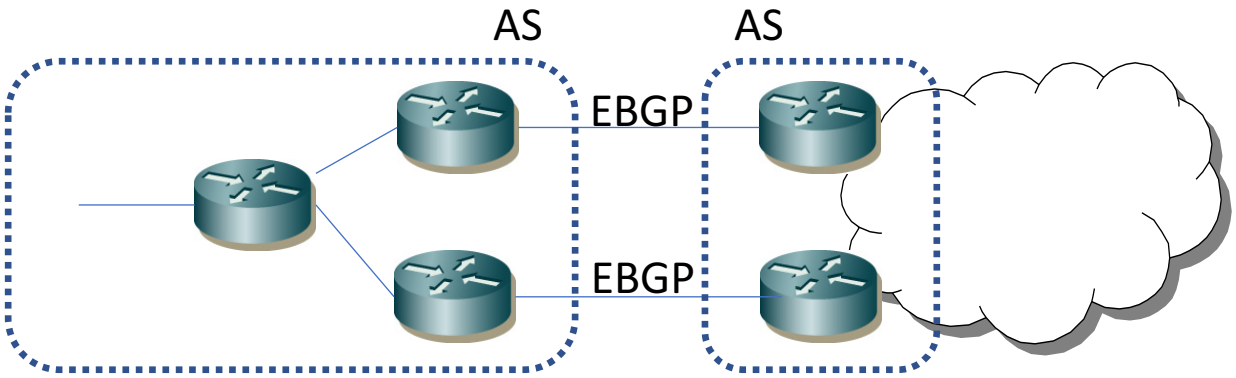
3. 上流追加: 冗長性確保のみ

- 基本的に動的経路制御任せ
- 到達性確保にだけ気をつければ大丈夫
  - 簡単かつシンプル
- 多少気の利いた経路制御を実現するには要検討
  →「IBGPをどこまで伸ばすか」を参照



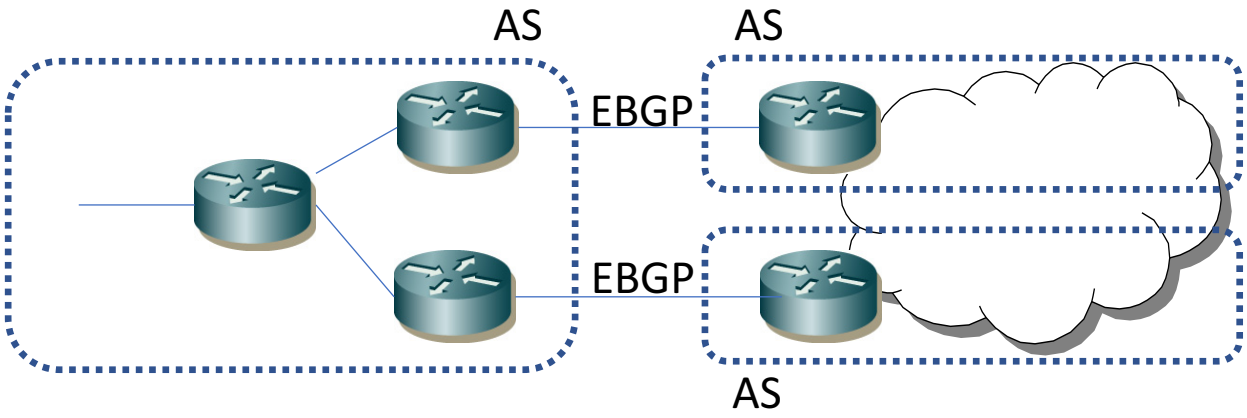
3. 上流追加: 負荷分散

- 上流への回線がほぼ同一視できるなら簡単
  - 同一ASで同一POPへの接続など
  - パケットをどちらの回線に送っても、だいたい一緒
- 双方向でECMP等を使った負荷分散が可能
  - IBGP Multipath, IGP Multipathなどが利用可能



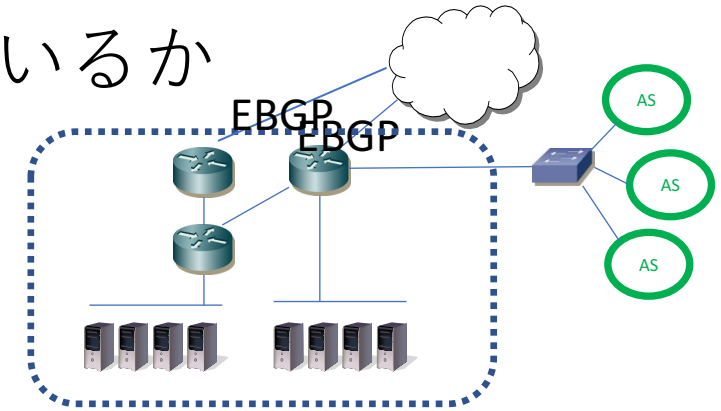
3. 上流追加: 負荷分散

- 同一視できないなら、何らか制御の導入
- トラヒック方向で制御が異なる
  - ネットワーク構成を考え直す必要があるかも
  - 大きなトラヒックを持つ事業者とうまくお話すると、BGP以上に細やかな制御ができる



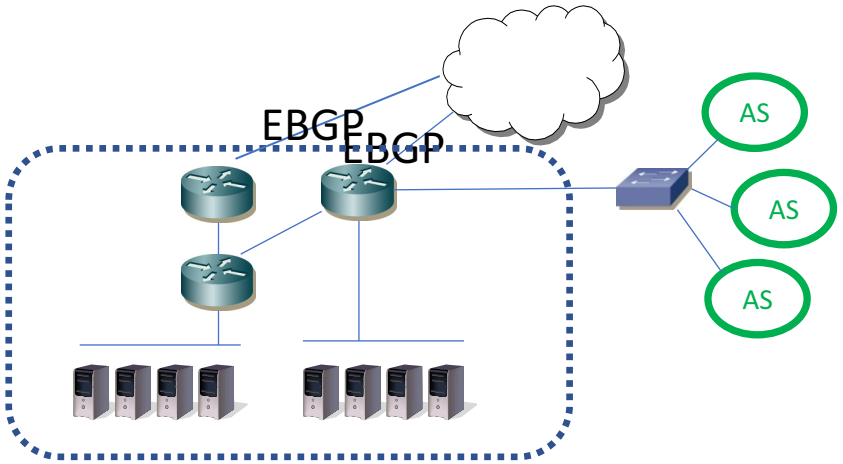
3. 上流追加: 内向き負荷分散

- 内向きのトラヒック制御は難しい
  - 他ネットワークの相互接続関係
  - 何やってもCDN事業者が移ったら一瞬で変動
- できる手段
  - AS_PATH prepend
  - prefix毎にMED(上流が同じASの場合)
- 吸い込む利用者はどこにいるか



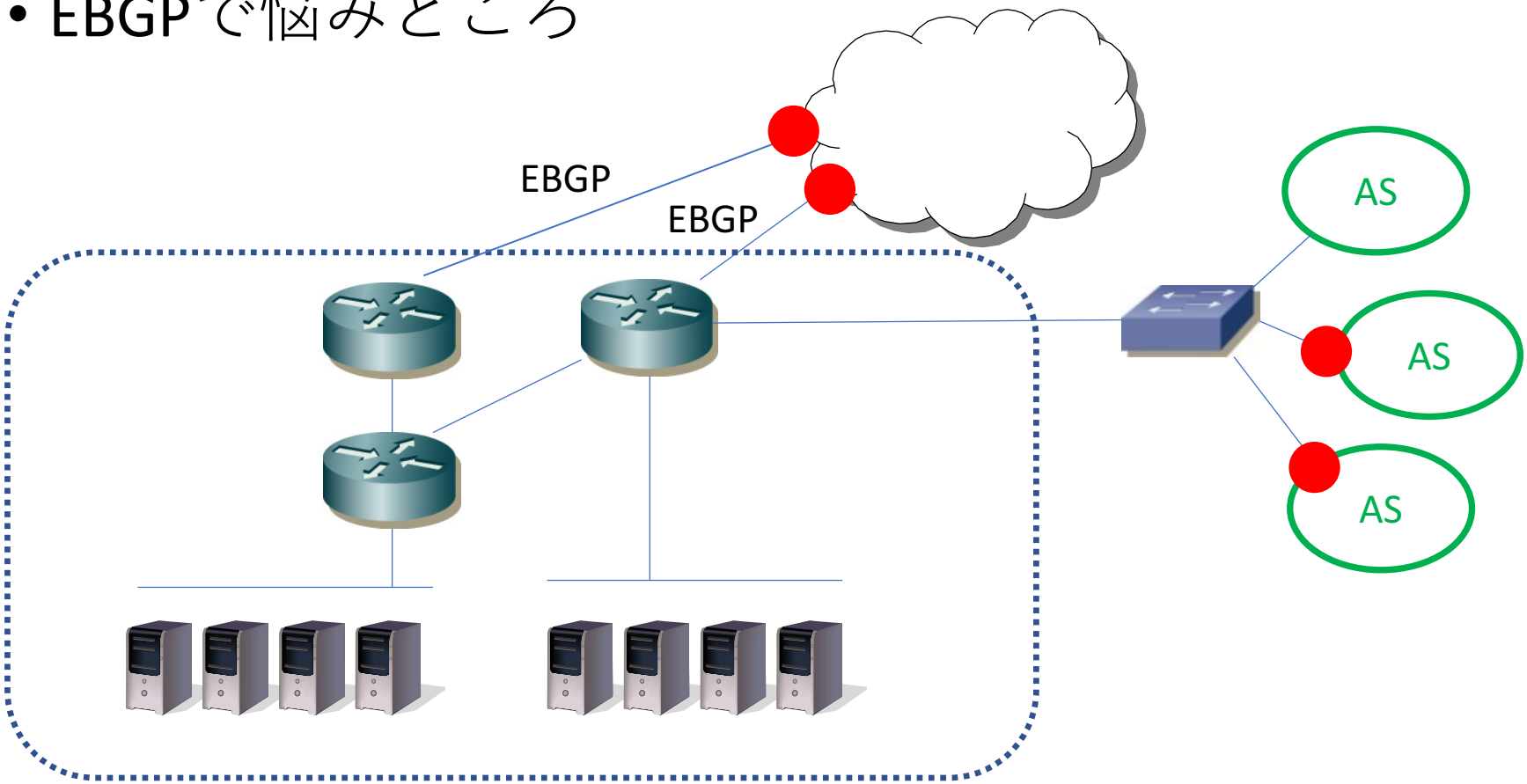
3. 上流追加: 外向き負荷分散

- 経路制御の基本は宛先ベース
  - ECMPで負荷分散
  - 宛先prefix毎に出口を優先制御
- トラヒックを吐いている利用者はどこにいるか



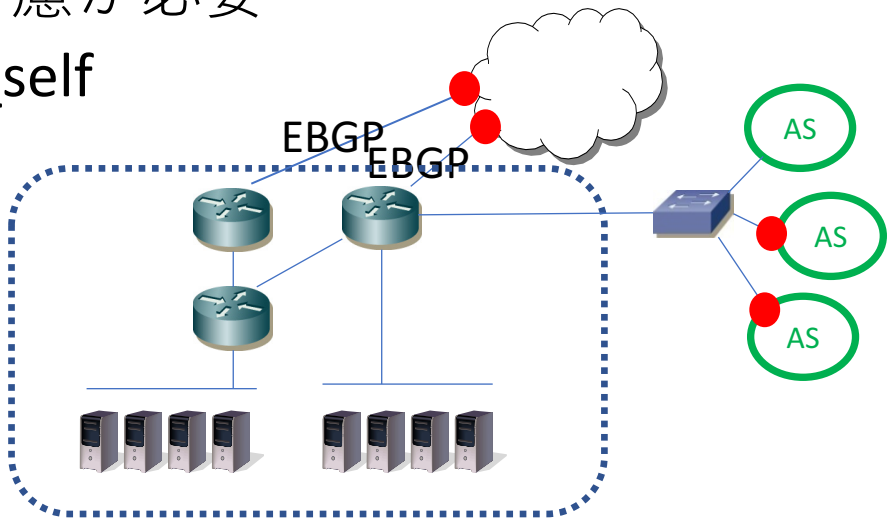
3. 上流追加：BGP NEXT_HOP

- EBGPで悩みどころ

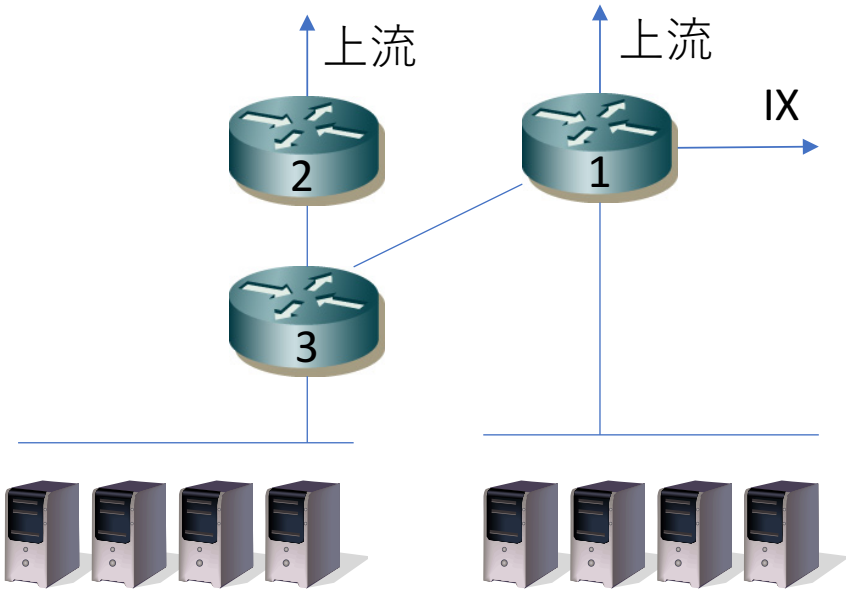


3. 上流追加：BGP NEXT_HOP

- IBGPを一切やらないなら問題なし
- 対外接続リンクのIPアドレスを網内にIGPに乗せて広報したいかどうか
  - 広報するなら、経路フィルタで他のASから細い経路を受信しないなどの考慮が必要
  - 広報しないならnexthop_self
    - IGPコストが変わりうる

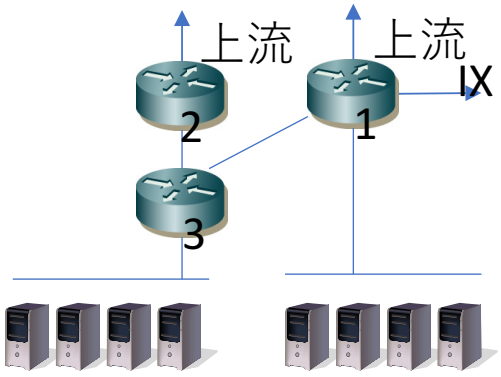


3. 上流追加：IBGPをどこまで



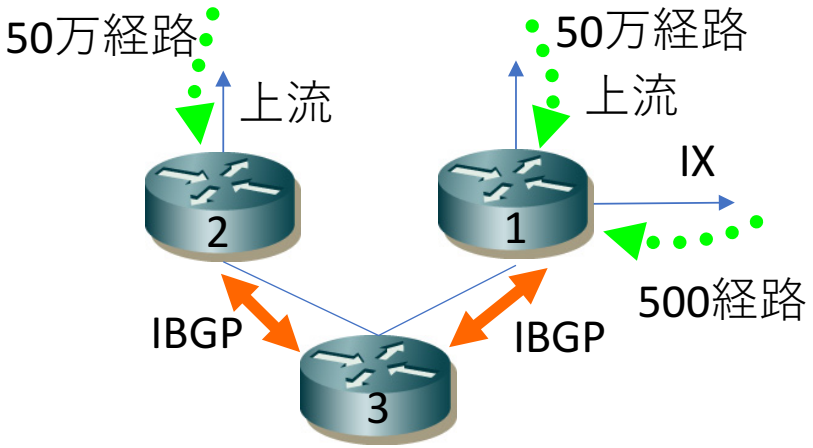
- IXで張ってるピア先には、そちらに流したい
- 上流はなんとなく負荷分散したい

3. 上流追加：IBGPをどこまで



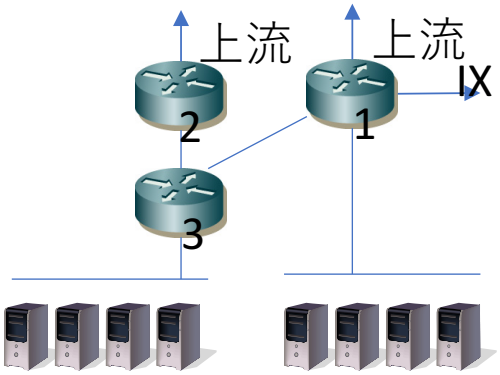
- オプション1: 良い
  - 全ルータでIBGP
  - BGP的に最適経路を選べる
- オプション2: 駄目
  - EBGPしているRT1&2間のみIBGP
  - RT1&2間に直結線を追加する必要がある
  - RT3配下のネットワークはIXを有効利用できない
- オプション3: まだまし
  - RT1&2からはdefaultのみをRT3に広報
  - RT3にはIX経由で受信した経路情報をIBGPで広報

3. 上流追加：BGP経路数



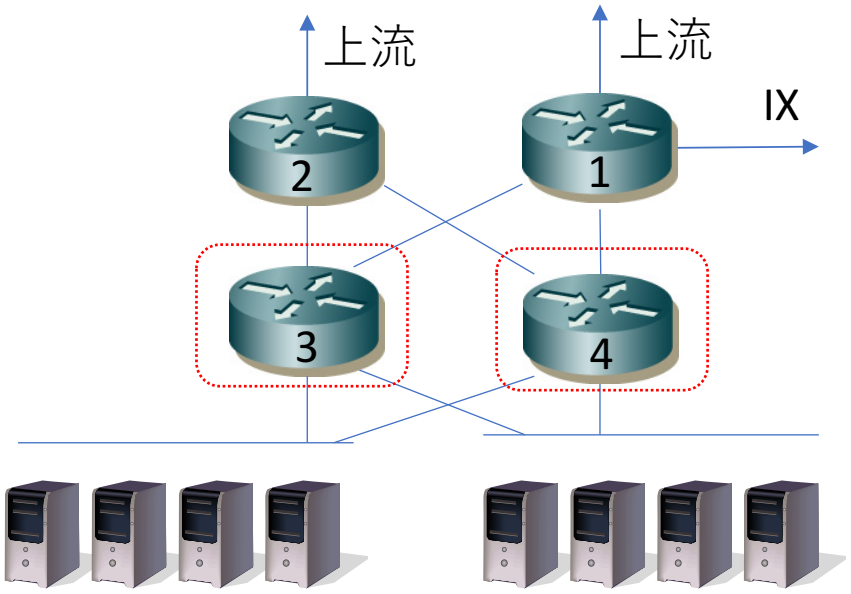
- 何もしないとRT3は約50万経路x2を受信
  - RT1, 2が異なるASに接続している場合、RT1, 2間でIBGPを張ることでRT3への経路広報を多少削減できる
  - でも、迂回性能はちょっと落ちる

3. 上流追加：経路生成



- 想定障害
  - ルータdown
  - 回線down
- オプション1: RT1 / RT3
  - 広報しているルータが死んだら終わり
  - 両方から広報しているとネットワーク分断が怖い
- オプション2: 構成変更
  - RT1に実ネットワークが収容されいてるのが課題
  - これをRT3配下に構成変更
  - もう少し冗長性が欲しくなってくる

3. 上流追加：経路生成

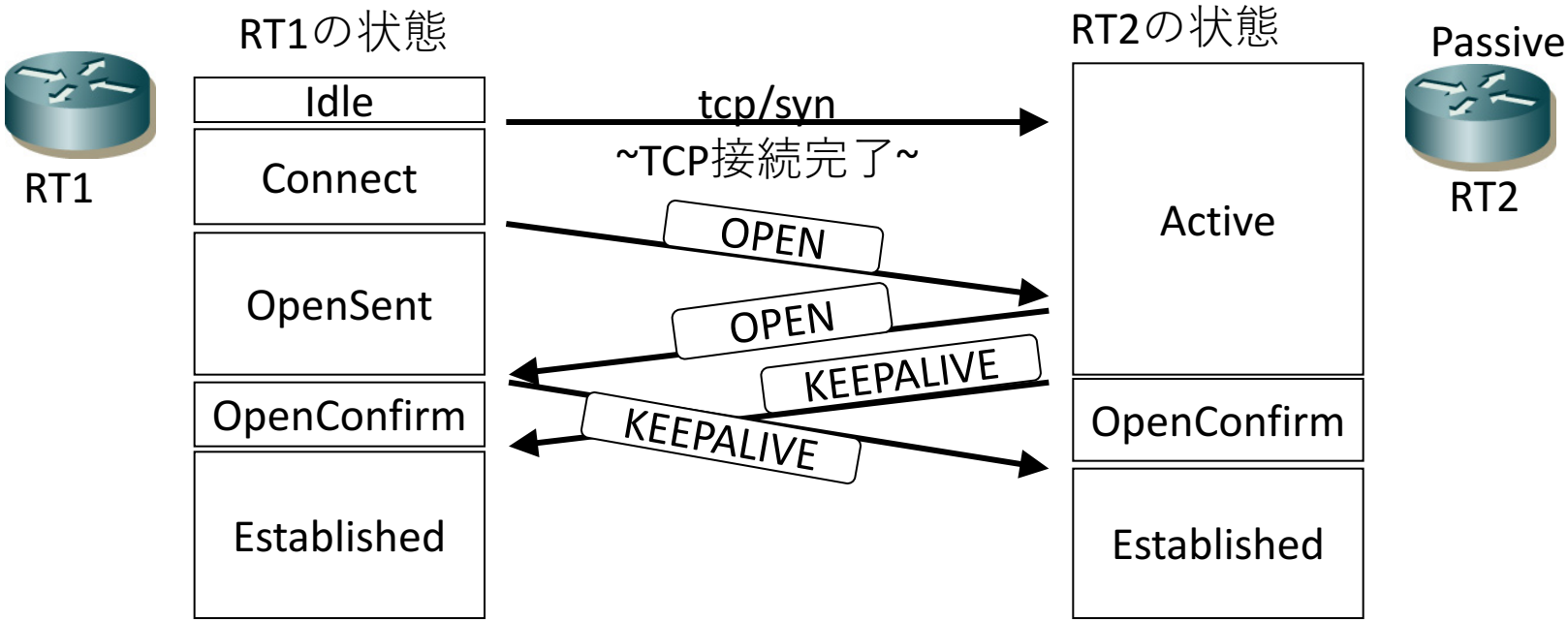


- 全ルータをIBGPで経路交換
- RT3とRT4でPA経路の生成

BGPパケット

BGPのプロトコルパケットの
フォーマットを解説する

BGP接続の確立



Idel – 初期状態

Connect – TCPの接続完了待ち

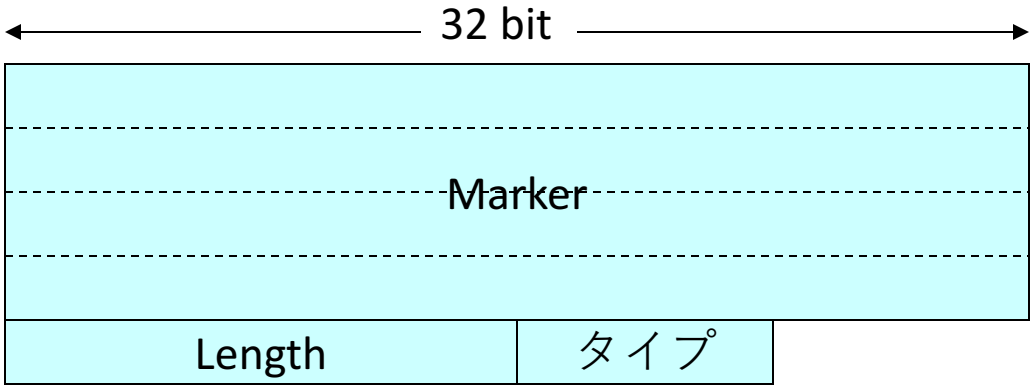
Active – 隣接からのTCP接続を待つ

OpenSent – OPEN送信後、隣接からのOPENを待つ

OpenConfirm – OPEN受信後、隣接からのKEEPALIVEを待つ

Established – BGP接続完了、経路交換の開始

BGP Message header



- Marker(マーカ)
  - 16-octetの全bitが１
  - 過去との互換性のため
- Length
  - 2-octetのメッセージ長
  - 19～4096

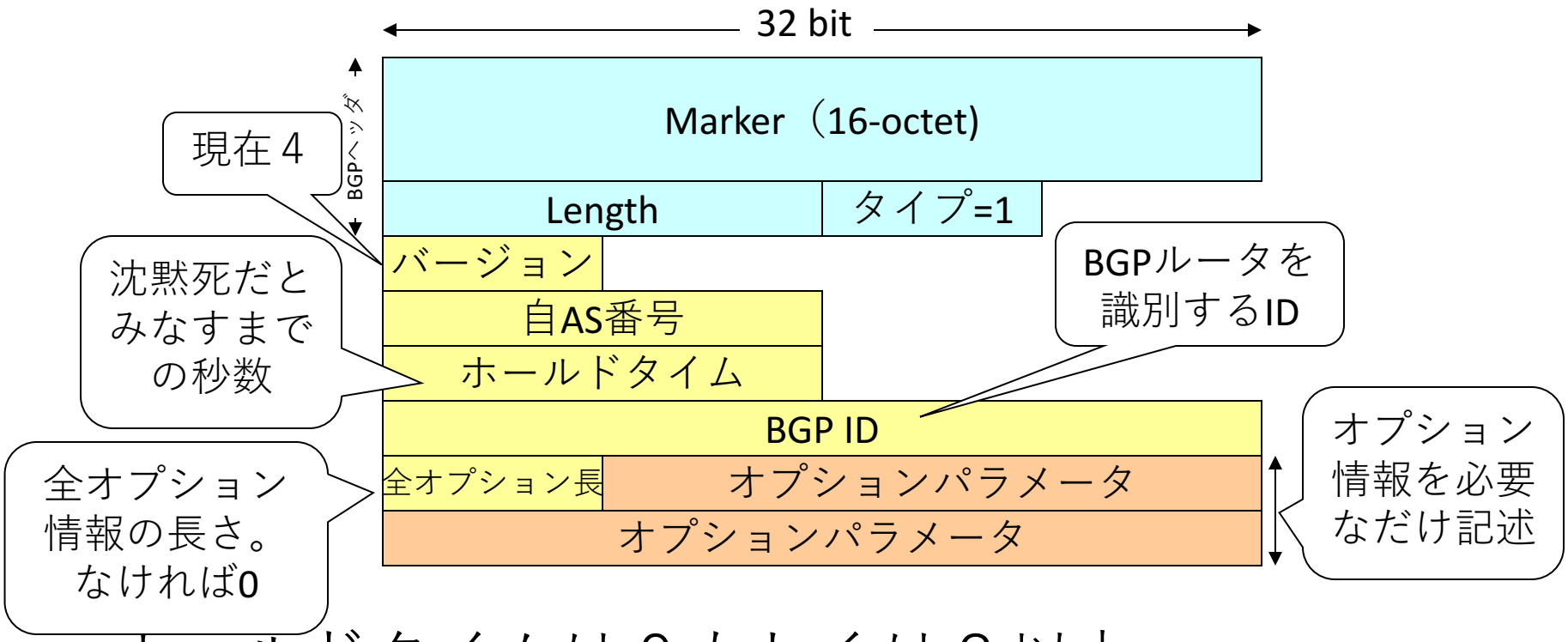
- タイプ（1-octet)
  1. OPEN
  2. UPDATE
  3. NOTIFICATION
  4. KEEPALIVE
  5. ROUTE_REFRESH

タイプ１ OPENメッセージ

- TCP接続が確立後、最初にやりとりされる
- パラメタの交換
  - バージョン、AS番号やBGP ID、ホールドタイム
  - オプションパラメータで各種機能を通知しあう

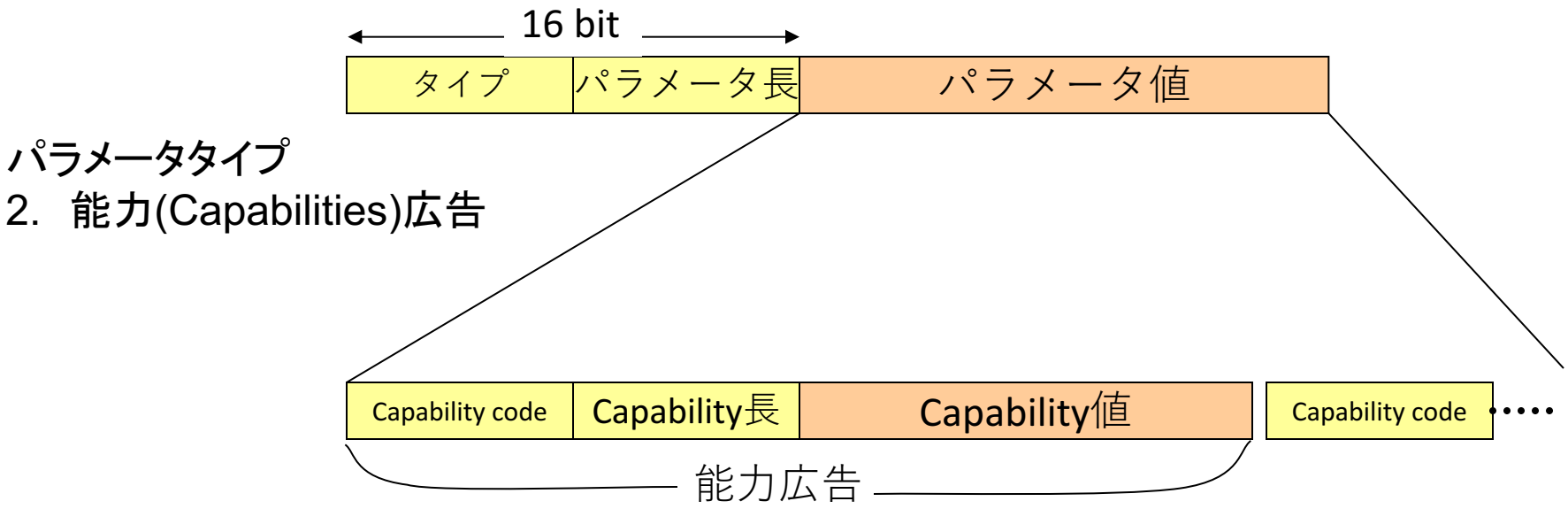
- タイプ4 KEEPALIVEで接続確立

タイプ１ OPENメッセージ



- ホールドタイムは０もしくは３以上
  - 小さな値が採用される
  - ０の場合、セッション維持にKEEPALIVEを利用しない

オプションパラメータフォーマット



- 今のところ能力広告のためだけに利用
  - 利用可能な機能をピア先へ通知する

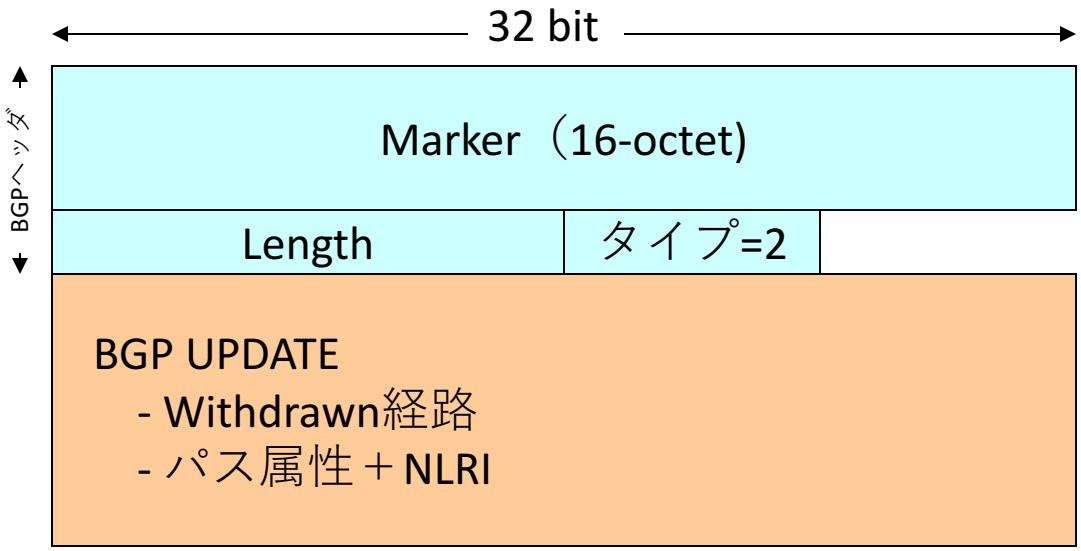
Capabilityコード

1	Multiprotocol Extension	サポートする<AFI, SAFI>の広告
2	Route Refresh	rfc版のRoute Refresh機能広告
3	Cooperative Route Filtering	
4	Multiple routes to a destination	
64	Graceful Restart	
65	Support for 4-octet AS number	
67	Support for Dynamic Capability	
128	Route Refresh(cisco)	Cisco独自のRoute Refresh機能広告

タイプ2 UPDATEメッセージ

- 経路情報を運ぶ
- 一つのメッセージで以下の情報を運べる
  - 複数のWithdrawn(取り消された)経路
  - 同じパス属性を持つ複数のNLRI
    - Withdrawn経路に含まれる経路は、同じメッセージ中でNLRIに含まれてはならない
- 情報の伝播保証はTCP任せ

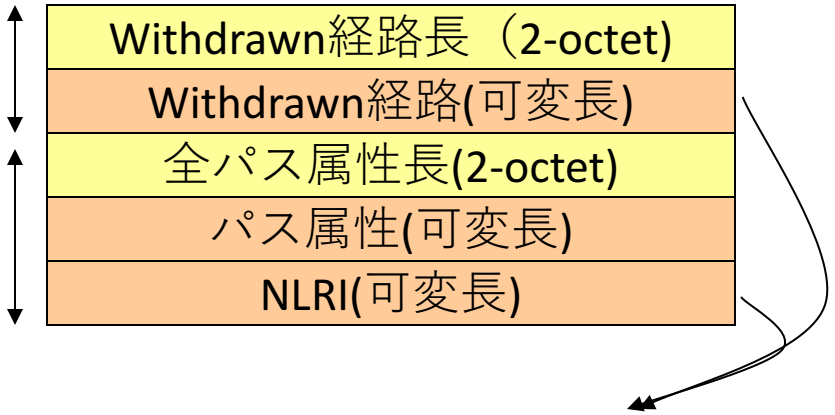
タイプ2 UPDATEメッセージ



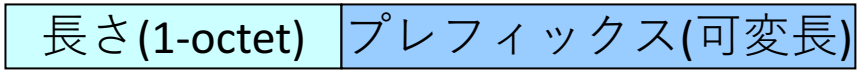
- パス属性が異なるNLRIは、異なるUPDATEメッセージで運ばれる

BGP UPDATEフォーマット

- Withdrawn経路
  - Withdrawnの長さ(2-octet)
  - Withdrawn経路の列挙
- 到達可能経路
  - 全パス属性の長さ(2-octet)
  - パス属性の列挙
  - NLRIの列挙

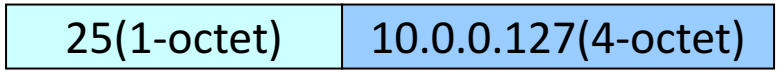


プレフィックスの格納形式



- 例：10.0.0.0/8

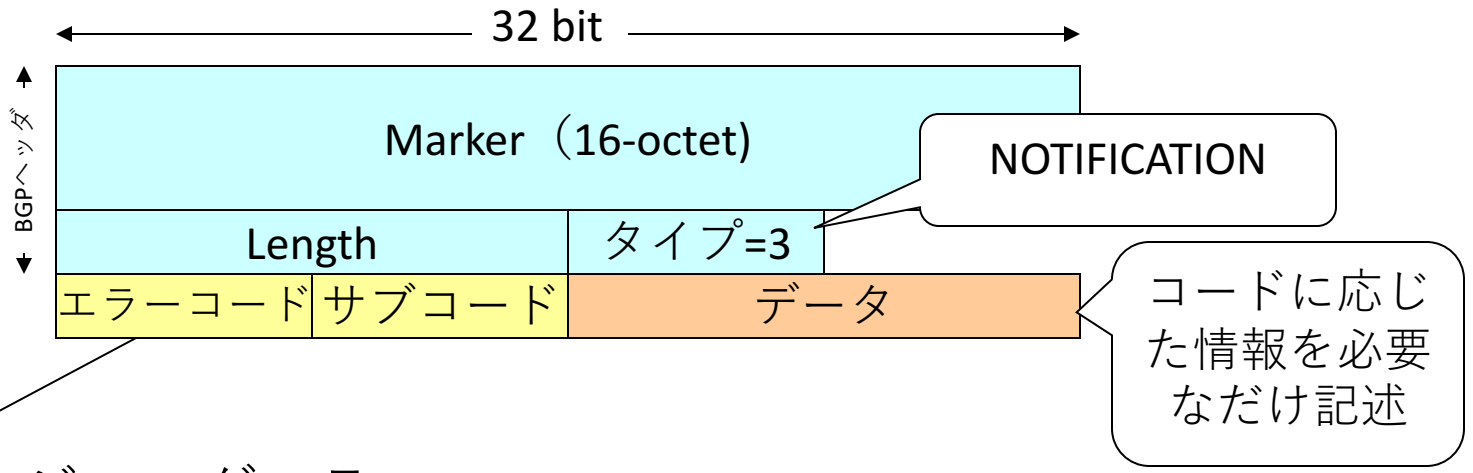
- 例：10.0.0.127/25



タイプ3 NOTIFICATIONメッセージ

- エラーを検出すると送信する
  - 送信後、すぐにBGP接続を切断する
- エラー内容がエラーコードとエラーサブコードで示される
  - 必要であれば、追加のデータも通知される

タイプ3 NOTIFICATIONメッセージ

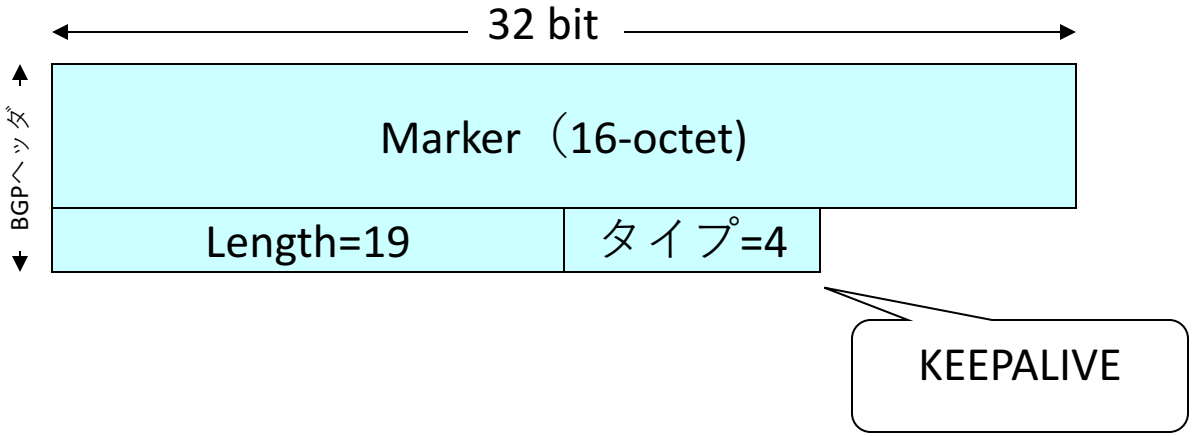


1. メッセージヘッダエラー
2. OPENメッセージエラー
3. UPDATEメッセージエラー
4. HoldTime超過
5. 状態遷移エラー
6. Cease
7. ROUTE-REFRESHエラー

タイプ4 KEEPALIVEメッセージ

- BGP接続を確立させる
- BGP接続を維持する
  - 送信間隔内にUPDATEが無ければ送信
  - 送信間隔はホールドタイムの１/3程度
    - 最小で１秒
  - ホールドタイムが０の場合は送信してはならない

タイプ4 KEEPALIVEメッセージ

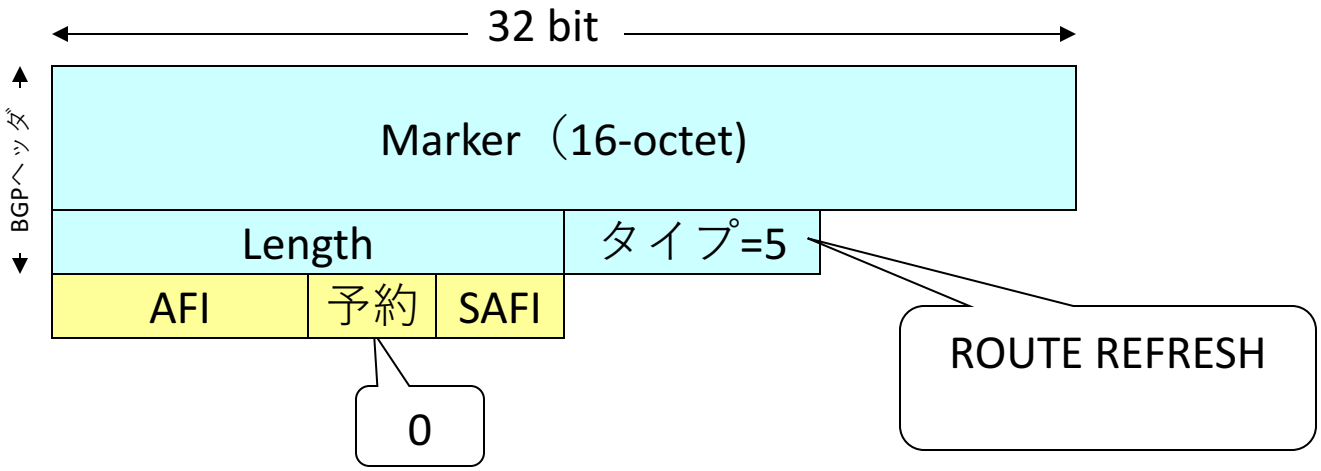


- KEEPALIVEであること以外、何も運ばない
- 最小のBGPメッセージ

タイプ5 ROUTE-REFRESHメッセージ

- 全経路の再広報を依頼する
  - <AFI, SAFI>を指定 (IPv4 unicastなど)
- 受信時、知らない<AFI, SAFI>であれば無視
- メッセージを送信するには、OPENメッセージのCapability広告でROUTE_REFRESH機能が通知されている必要がある

タイプ5 ROUTE-REFRESHメッセージ

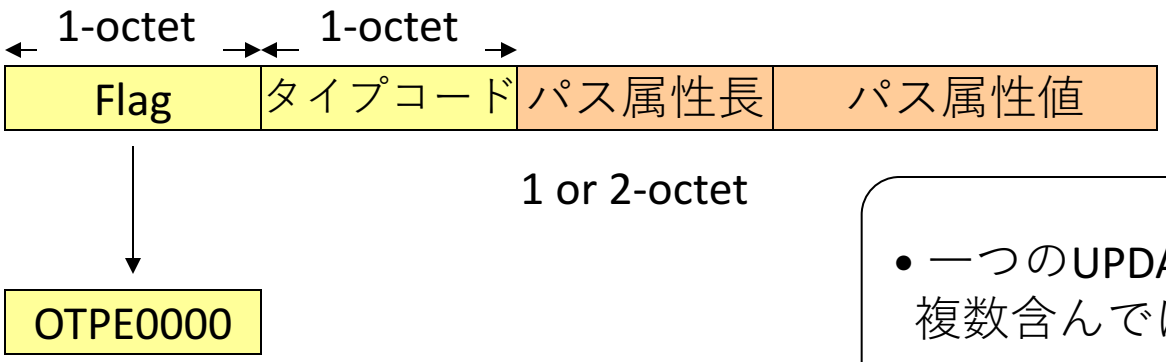


- AFI = Address Famiry Identifier
  - IPv4やIPv6など
- SAFI = Subsequent Address Famiry Identifier
  - UnicastやMulticastなど

パス属性

パス属性の構成と主要なパス属性について解説する

パス属性フォーマット



- 一つのUPDATEに同じパス属性を複数含んではいけない

O bit: Optional(パス属性の種別)
0=Wellknown, 1=optional
T bit: Transitive(パス属性の転送)
0=non-transitive, 1=transitive
P bit: Partial(パス属性の処理)
0=complete, 1=partial
E bit: Extended length
0=パス属性長は1-octet
1=パス属性長は2-octet

- Partial bit
  - オプション属性が、経路が広報されてから経由した全てのルータで解釈されたかどうかを示す
  - 0:全てのルータで解釈された
  - 1:解釈されなかったルータあり

パス属性の４つのカテゴリ

- 周知必須 - well-known mandatory [T]
  - 全てのBGPルータで解釈可能
  - NLRI情報があれば必ずパス属性に含まれる
- 周知任意 - well-known discretionary [T]
  - 全てのBGPルータで解釈可能
  - 必ずしも含まれない
- オプション通知 - Optional transitive [OT]
  - 一部のBGPルータでは解釈できないかもしれない
  - 解釈できなくても、そのまま他のルータに広報する
    - この際、Partial bitを1にセットする
- オプション非通知 - Optional non-transitive [O]
  - 一部のBGPルータでは解釈できないかもしれない
  - 解釈できない場合は、他のルータに広報するとき属性を削除する

ORIGIN属性値

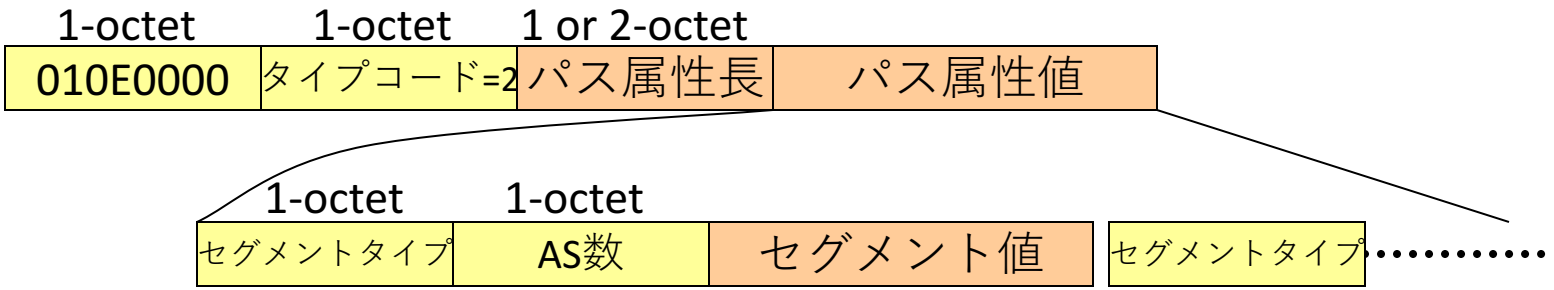
- 周知必須
- NLRIの起源を示す３つのタイプ
- 経路生成元で付加され、その後変更されない

0 – IGP ・・・ AS内部で生成
1 – EGP ・・・ EGP[RFC904]から生成
2 – INCOMPLETE ・・・その他の方法で生成

AS_PATH属性

- 周知必須
- NLRIが通過してきたAS番号のリスト
  - 例えば“10 20 30”
  - 一番右は経路を生成したAS番号
  - 他のASに広報するときに先頭に自AS番号を付加
- 用途に応じてセグメントが用意されている
  - 通常はAS_SEQUENCEを利用する
  - 異なるAS_PATHを集約した場合はAS_SET
  - AS_SETは{}でくくられる表記が多い
    - 例えば”10 20 30 {40 41}”

AS_PATH属性フォーマット



セグメントタイプ

1: AS_SET
UPDATEが経由したAS番号。順序は意味を持たない
異なるAS Pathの経路を集約したときに生成される

2: AS_SEQUENCE
UPDATEが経由したAS番号。順序に意味がある
経由した最新のAS番号はセグメント値の一番左

AS数

octet数ではなく、AS数
つまり、255個のASまで

セグメント値

2-octetのAS番号のリスト

- 新しいセグメントは先頭(左)に付加される
- ふつーはAS_SEQUENCEのみ

AS_PATH属性の処理

- 経路を転送する場合

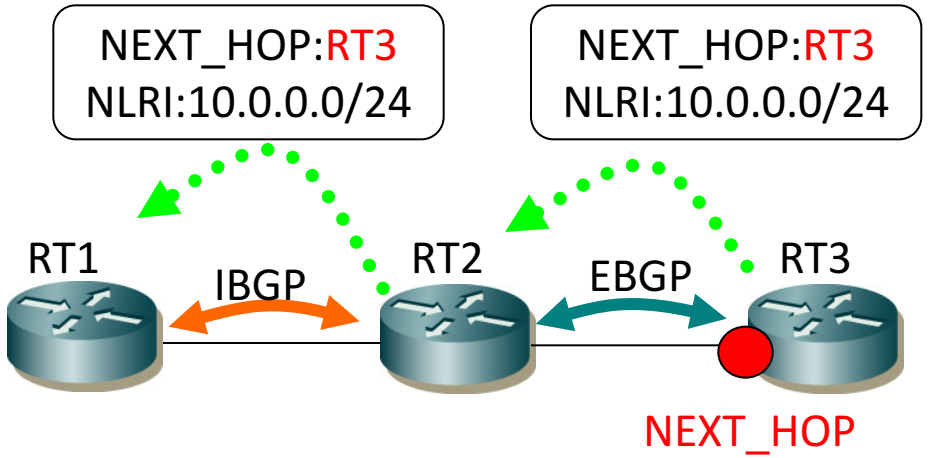
広報先	
IBGP	変更しない
EBGP	自AS番号をAS_SEQUENCEタイプでAS_PATH属性の先頭に付加する

- 経路を生成する場合

広報先	
IBGP	空のAS_PATH属性を生成する
EBGP	AS_SEQUENCEタイプで自AS番号のみのAS_PATH属性を生成する

NEXT_HOP属性

- 周知必須
- NLRIへ到達するためのネクストホップIPアドレス



NEXT_HOP属性の処理

- IBGPに経路を転送するときは
  - 変更しない
  - ただし、設定で自身のIPアドレスに変更することも可能
- IBGPに生成した経路を広報するときは
  - その宛先に到達するためのネクストホップを設定する
  - ただし、自身のIPアドレスを設定することも可能
- EBGPに経路を広報するときは
  - BGP接続に利用している自身のIPアドレスを設定する
  - ただし、宛先のネクストホップがEBGPルータと共通のサブネットに属する場合は、他のルータのIPアドレスや自身の別なインタフェースのIPアドレスを設定することも可能

MULTI_EXIT_DISC(MED)属性

- 周知任意
- 隣接ASとの距離を表す４-octetの数値
  - 小さいほど優先される
  - 付加されていないと最小の０と見なす[RFC4271]
- EBGPで受信したMEDは、他のEBGPにそのまま広報してはならない
- 幾つかの注意点
  - BGP MED Considerations [RFC4451]　など

LOCAL_PREF属性

- 周知
- AS内での優先度を示す4-octetの数値
  - 大きいほど優先される
- IBGPとEBGPで取り扱いが異なる
  - IBGPへの広報では付加されるべき
  - EBGPへの広報では付加してはならない
    - 付加されていた場合は無視
    - コンフェデレーションのSubAS間の場合は例外

COMMUNITIES属性

- オプション通知
- NLRIに32bitの数値で情報を付加する
  - この情報を元に予め実装したポリシ等を適用
- 上位16bitと下位16bitに分けた表記が一般的
  - １０進数で”上位:下位” の様に表記する
  - 自ASでの制御は上位に自AS番号を用い、下位で制御の情報を付加するのが一般的
  - つまり ”asn:nn”

Well-Known-community

- (0xFFFFFF01)　**NO_EXPORT**
  - 他ASに広報しない
  - コンフェデレーション内のメンバASには広報する
- (0xFFFFFF02)　**NO_ADVERTISE**
  - 他BGPルータに広報しない
- (0xFFFFFF03)　**NO_EXPORT_SUBCONFED**
  - 他ASに広報しない
  - コンフェデレーション内でメンバASにも広報しない
- (0xFFFFFF04)　**NOPEER** [RFC3765]
  - 対等ピアには広報しない
  - まだ実装は無さそう

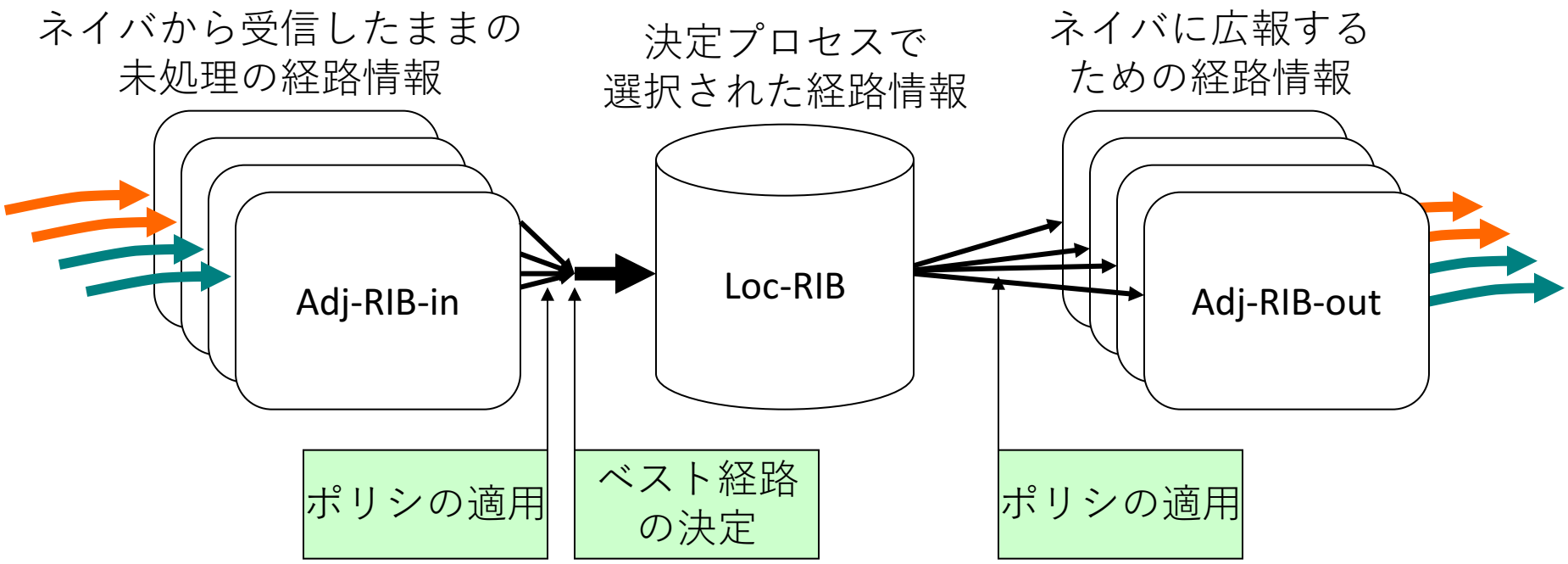
EBGP＆IBGPとパス属性

パス属性	EBGP	IBGP
ORIGIN	必須	必須
AS_PATH	必須	必須
NEXT_HOP	必須	必須
MULTI_EXIT_DISC	任意	任意
LOCAL_PREF	不許可	付加すべき
COMMUNITIES	任意	任意

BGPの経路選択

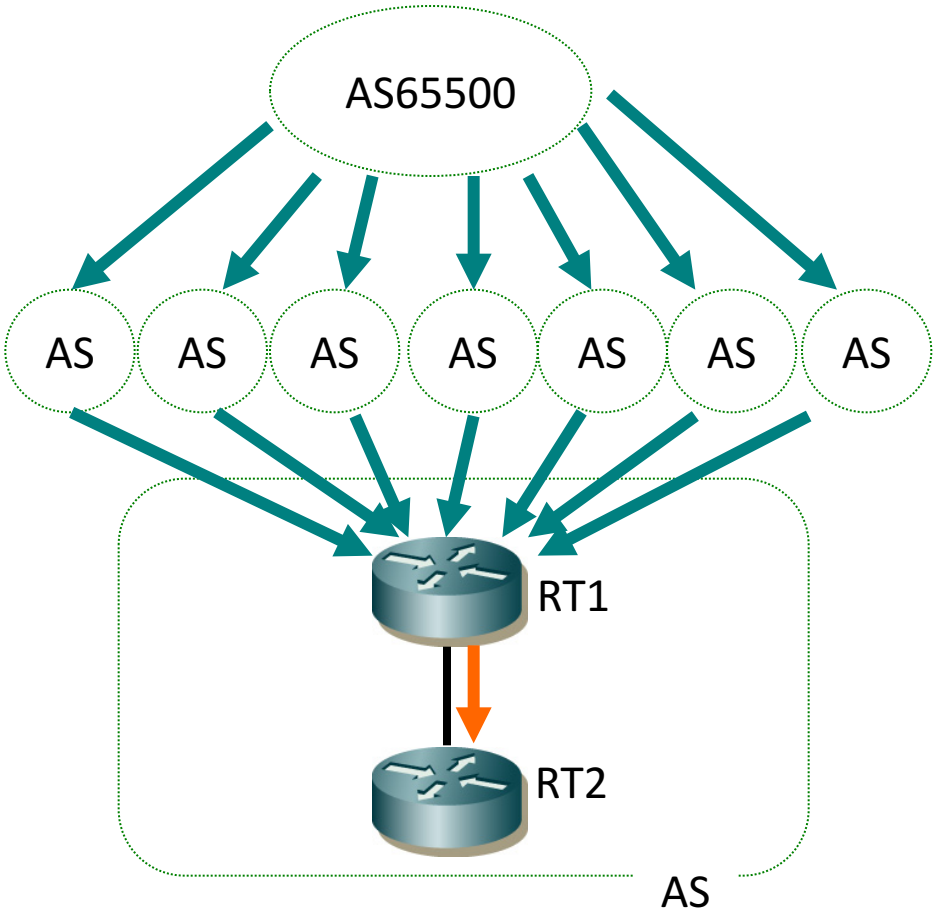
経路処理方法や、経路選択ルールを解説する

BGPの経路処理



- ポリシは設定/実装依存
- 無理なポリシを適用すると、経路ループを引き起こす可能性があるので注意

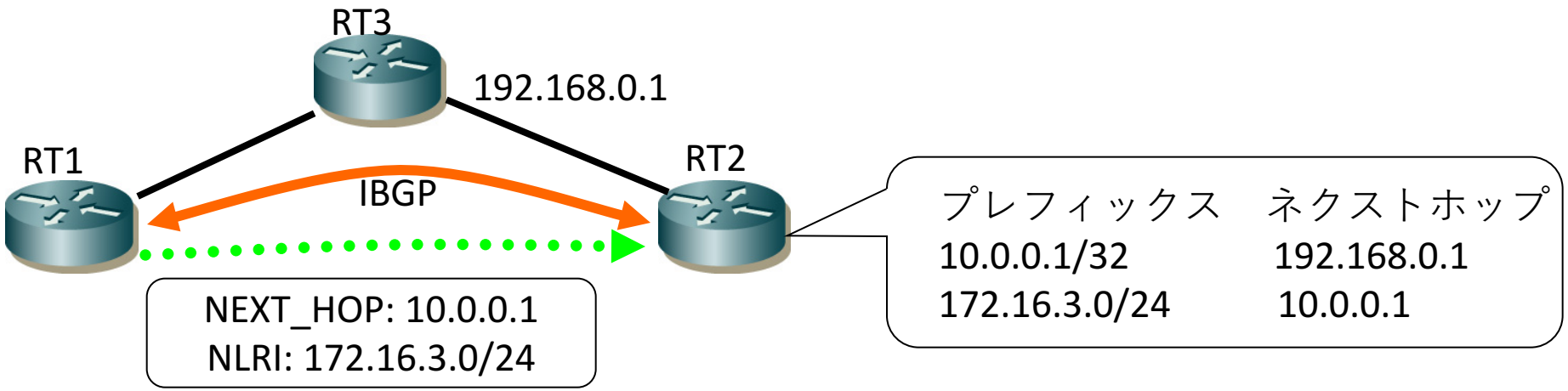
ベスト経路のみを広報



- RT1では７経路見える
  - ただし利用している経路はベストの１つだけ
- RT2へ広報されるのはRT1で選択されたベスト経路のみ
  - 経路に変更があるって最適経路が変わると、それが広報されて上書きされる

NEXT_HOP解決

- NEXT_HOP属性のIPアドレスまで到達可能であること
  - BGPも含めた経路で再帰解決して、最終的にBGPルータの隣接するネクストホップが得られる必要がある[RFC4271]



経路優先度

1	NEXT_HOP	NEXT_HOP属性のIPアドレスが到達不可能な経路は無効
2	AS loop	AS Path属性に自身のAS番号が含まれている経路は無効
3	LOCAL_PREF	LOCAL_PREF属性値が大きい経路を優先 (LOCAL_PREF属性が付加されていない場合は、ポリシに依存)
4	AS_PATH	AS_PATH属性に含まれるAS数が少ない経路を優先 (AS_SETタイプは幾つASを含んでも1として数える)
5	ORIGIN	ORIGIN属性の小さい経路を優先 (IGP < EGP < INCOMPLETE)
6	MULTI_EXIT_DISC	同じASからの経路はMED属性値が小さな経路を優先 (MED属性が付加されていない場合は、最小(=0)として扱う)
7	PEER_TYPE	IBGPよりもEBGPで受信した経路が優先
8	NEXT_HOP METRIC	NEXT_HOPへの内部経路コストが小さい経路が優先 (コストが算出できない経路がある場合は、この項目をスキップ)
9	BGP_ID	BGP IDの小さなBGPルータからの経路が優先 (ORIGINATOR_IDがある場合は、これをBGP IDとして扱う)
10	CLUSTER_LIST	CLUSTER_LISTの短い経路が優先
11	PEER_ADDRESS	ピアアドレスの小さなBGPルータからの経路を優先

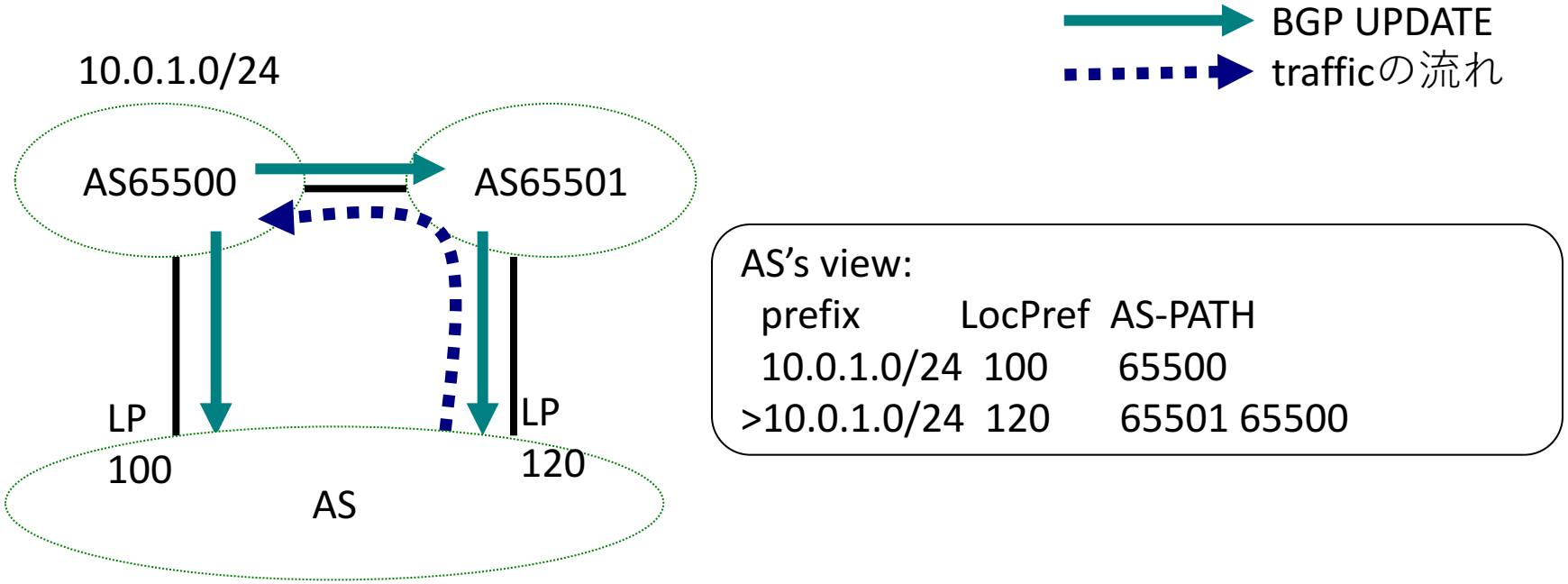
属性値の評価

属性値がどう評価されるかを
解説する

受信経路で重要な属性値

- Local Preference
  - 受信時に設定する
- AS Path
  - 相手ASから広報される
- MED
  - 相手ASから設定されて広報される、もしくは受信時に上書き設定する
- NEXT_HOP Cost
  - AS内部のトポロジに依存する

Local Preference

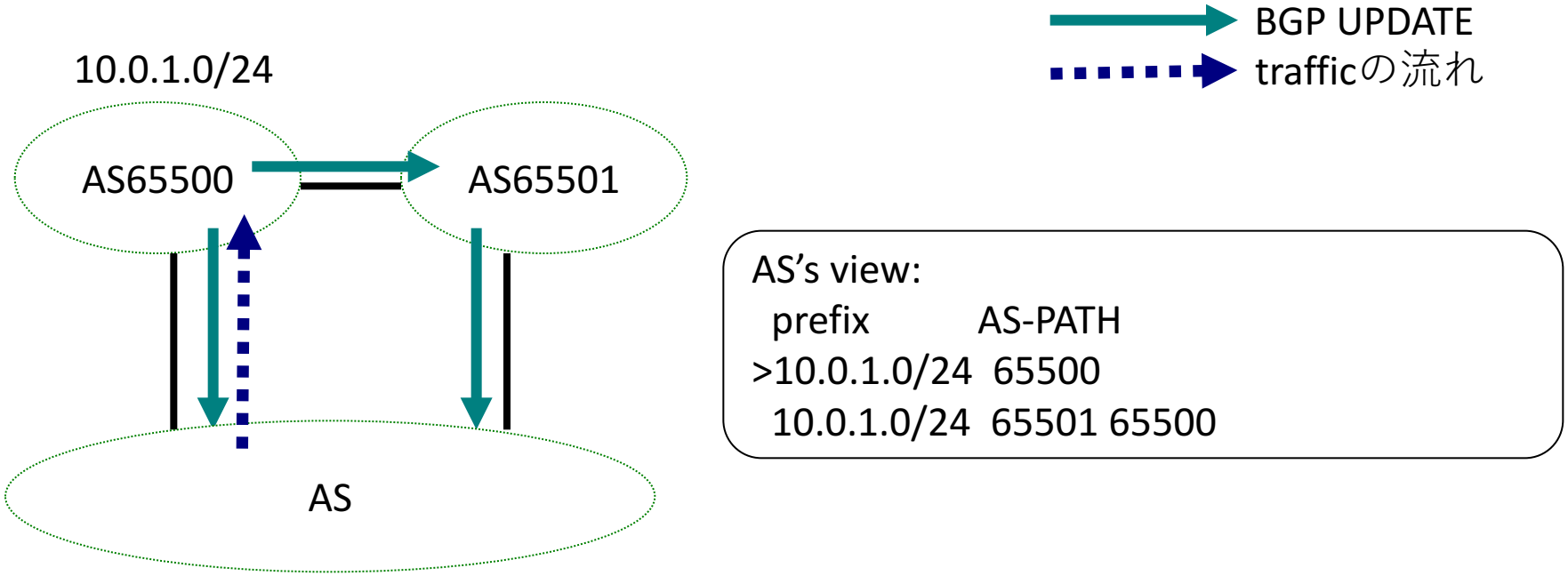


AS's view:

prefix	LocPref	AS-PATH
10.0.1.0/24	100	65500
>10.0.1.0/24	120	65501 65500

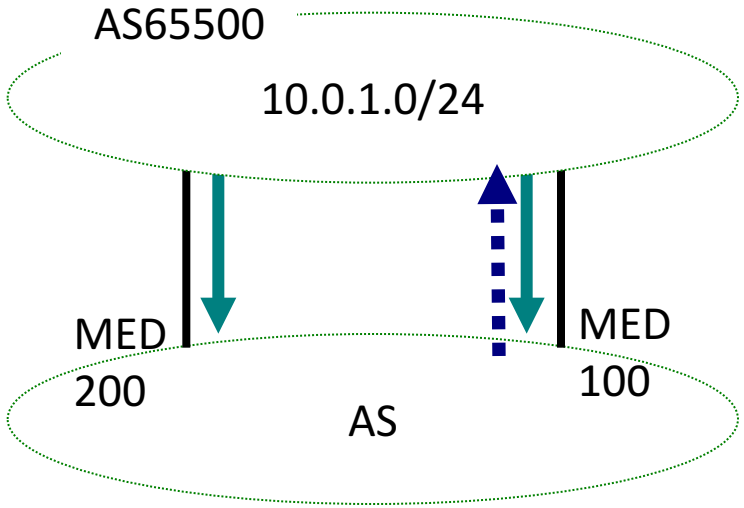
- Local Preferenceの大きな値が優先
- あるAS経由の経路を優先したい場合に有効

AS Path



- AS Path長が短い経路が優先

MED(MULTI_EXIT_DISC)

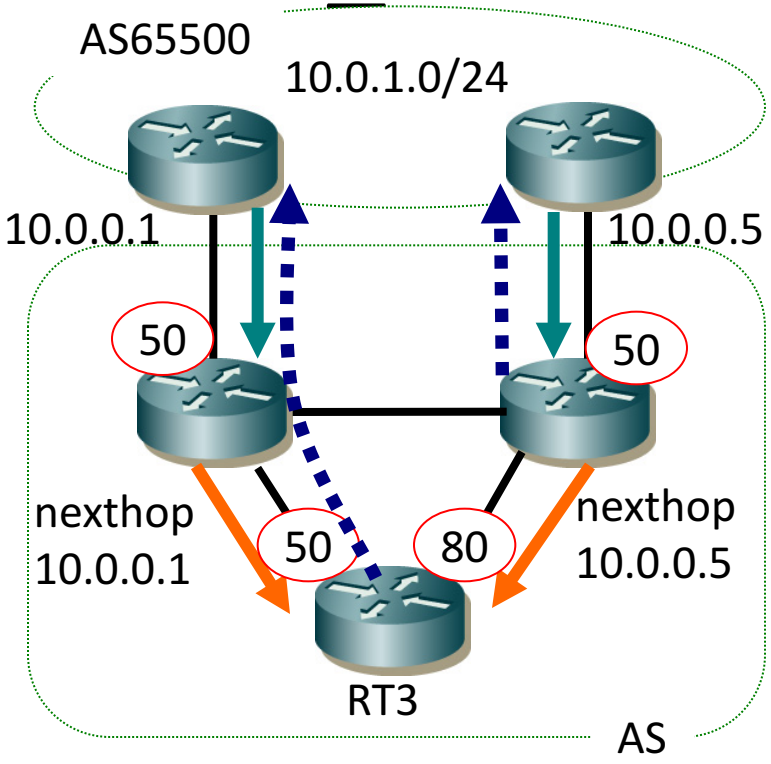


AS's view:

	prefix	MED
>	10.0.1.0/24	100
	10.0.1.0/24	200

- MEDの値が小さい経路が優先
- あるASとの複数接続に優先順位をつけたい場合に有効

NEXT_HOP COST



- NEXT_HOPへのigpコストが小さい経路を優先
- これを利用したのが closest exit

RT3's view:

prefix	nexthop [cost]
>10.0.1.0/24	10.0.0.1 [100]
10.0.1.0/24	10.0.0.5 [130]

他ASへの広報で重要な属性値

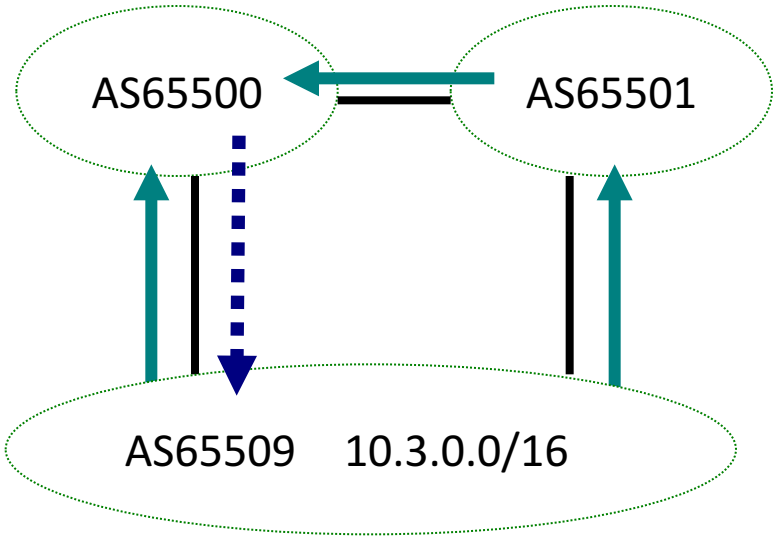
- AS Path
  - prependでAS Path長を伸ばす
- MED
  - 複数接続に優先順位をつける
- Community
  - 広報先ASでの処理を期待する

- 相手とのポリシのすり合わせが重要

AS Path (広報時)

AS65500

prefix	AS-PATH
>10.3.0.0/16	65509
10.3.0.0/16	65501 65509



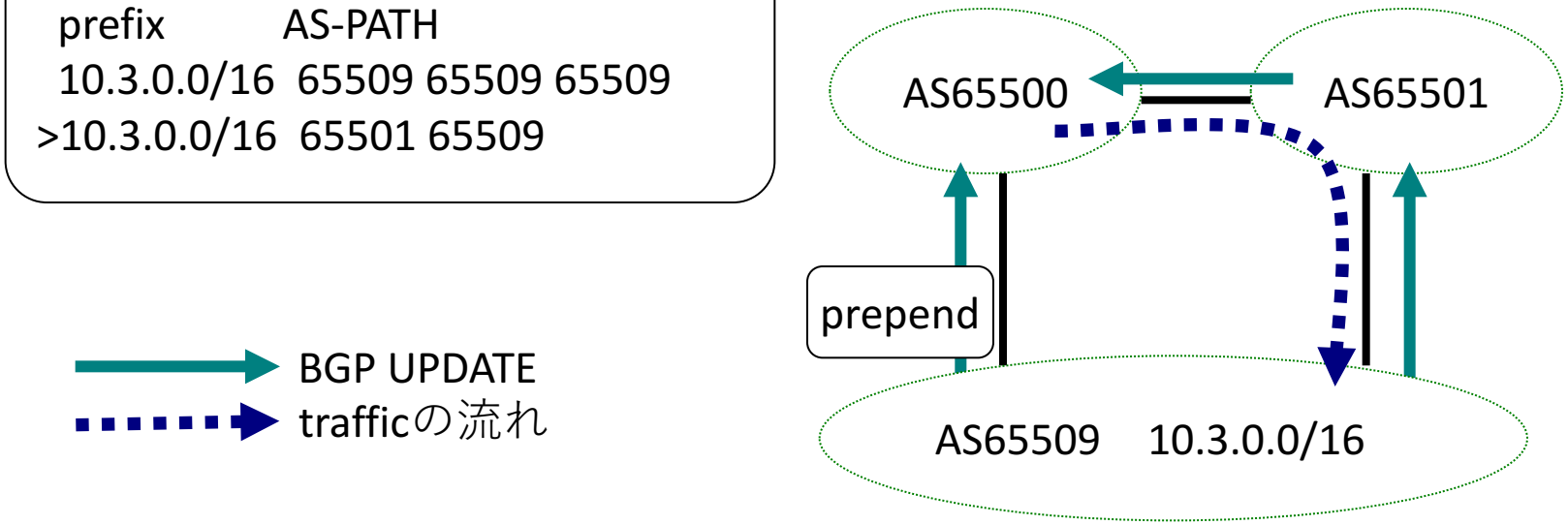
BGP UPDATE
trafficの流れ

- AS Path長が短い経路が優先

AS Path prepend

AS65500

	prefix	AS-PATH
	10.3.0.0/16	65509 65509 65509
>	10.3.0.0/16	65501 65509

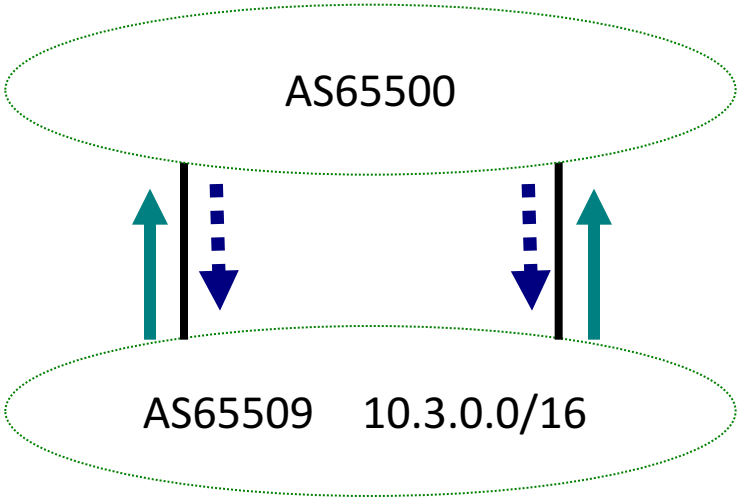


- あるASとの接続リンクを利用したくない場合に、AS Pathを長くして優先度を下げることが出来る

広報通常時

AS65500

prefix	AS-PATH
10.3.0.0/16	65509
10.3.0.0/16	65509



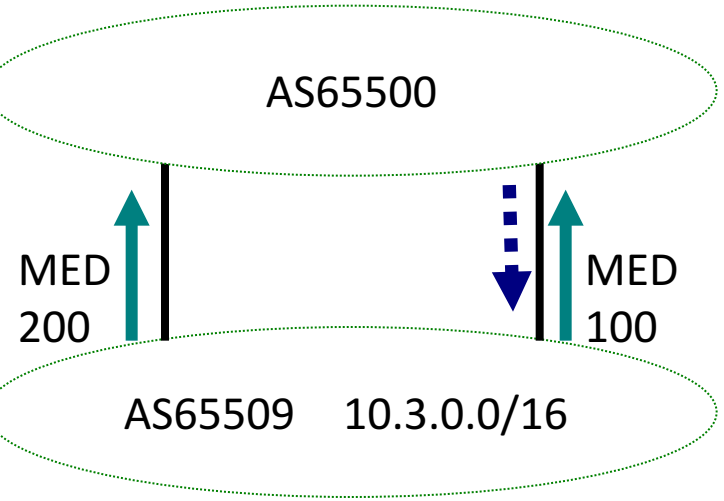
BGP UPDATE
trafficの流れ

- AS65500で特別な制御を行っていなければ、closest exitになるはず
  - トラヒックの分散は相手ASの構成に依存する

MED（広報時）

AS65500

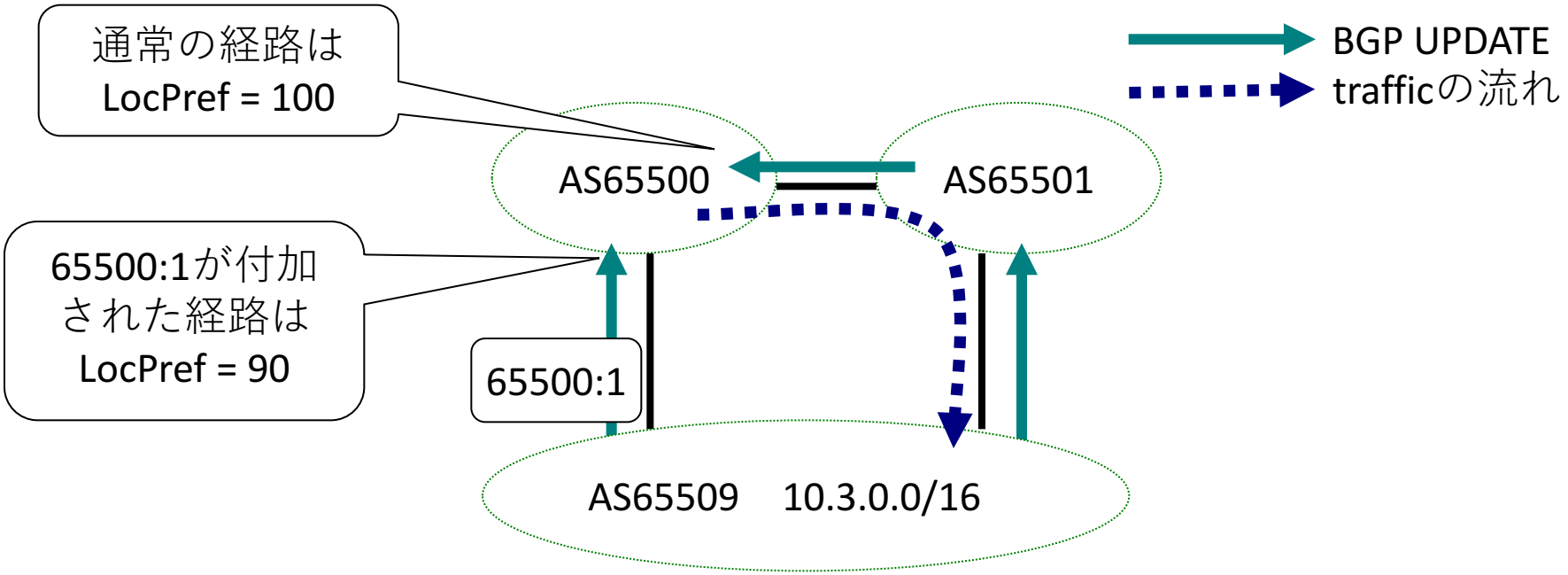
prefix	MED	AS-PATH
>10.3.0.0/16	100	65509
10.3.0.0/16	200	65509



BGP UPDATE
trafficの流れ

- 複数接続に優先順位をつけたい場合
- AS65500でMEDを受け付ける設定になっていれば、小さなMED値の経路が優先される
- MEDを受け付けるかどうかは相手ASのポリシ依存

Community利用例

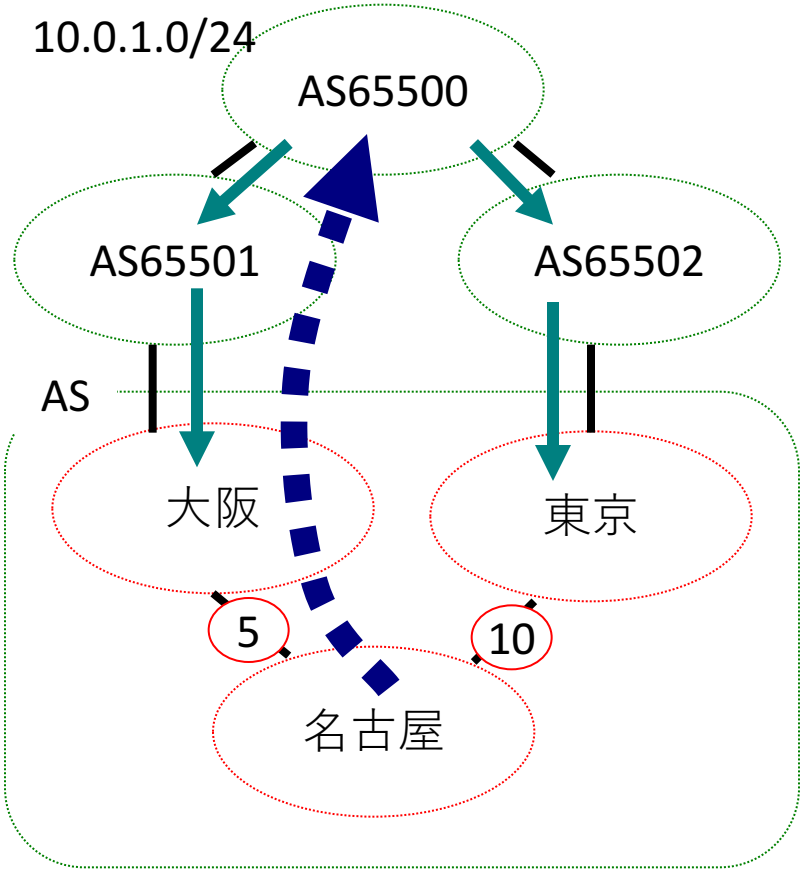


- AS65500がCommunity制御を実装していれば利用できる
- 経路にCommunity情報を付加して、その制御を利用する
- Communityを受け付けるかどうかはASのポリシ依存

BGPのパス選択

OSPFとBGPの関わりなどを
解説する

closest exitとBGP

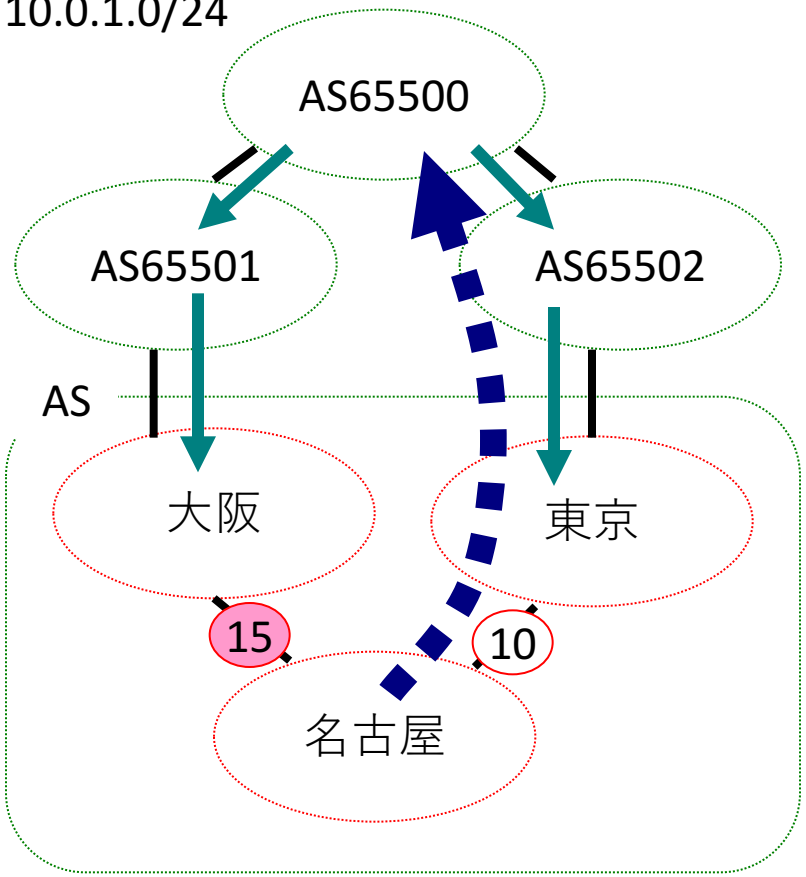


名古屋

prefix	AS-PATH	
>10.0.1.0/24	65501 65500	5
10.0.1.0/24	65502 65500	10

- 名古屋では、65501(大阪)経由を選択中

OSPFのコスト変更



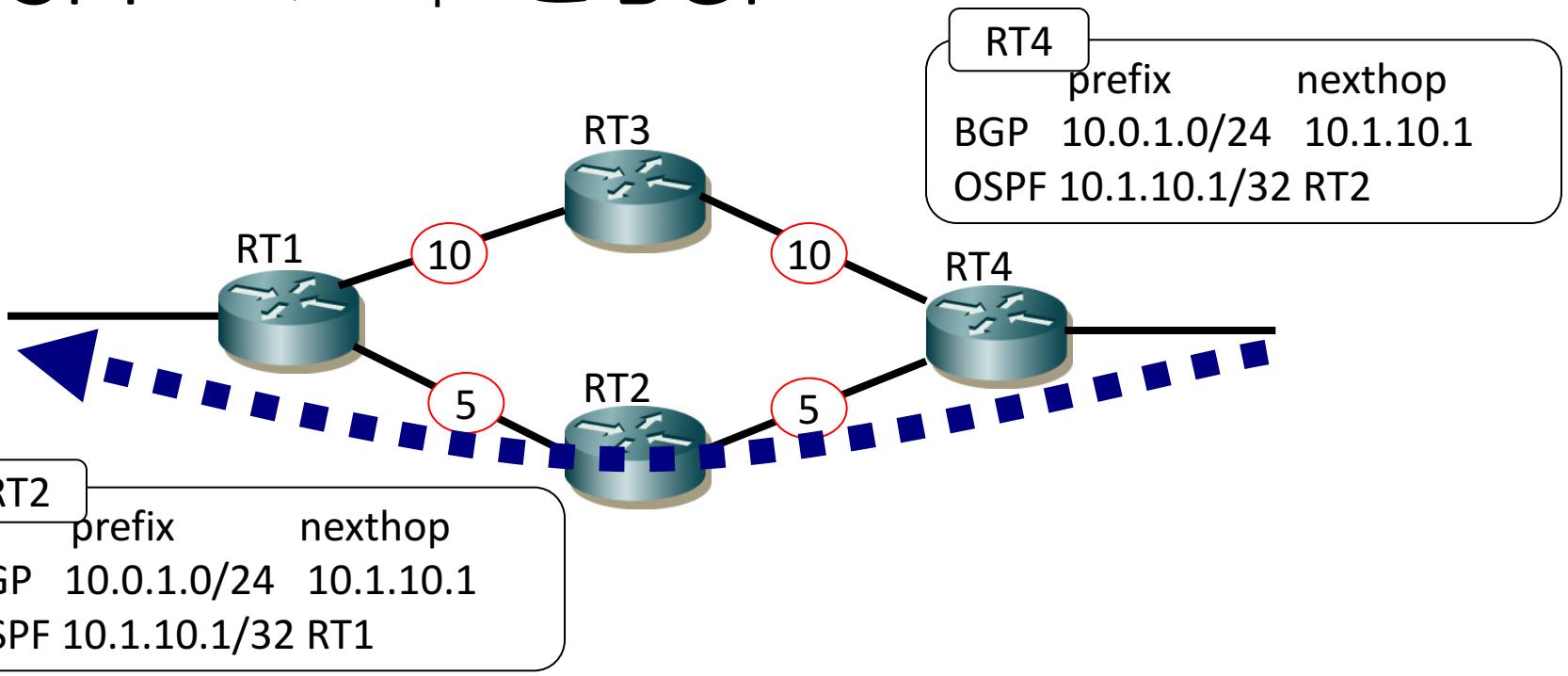
BGP UPDATE
trafficの流れ

名古屋

prefix	AS-PATH	
10.0.1.0/24	65501 65500	15
>10.0.1.0/24	65502 65500	10

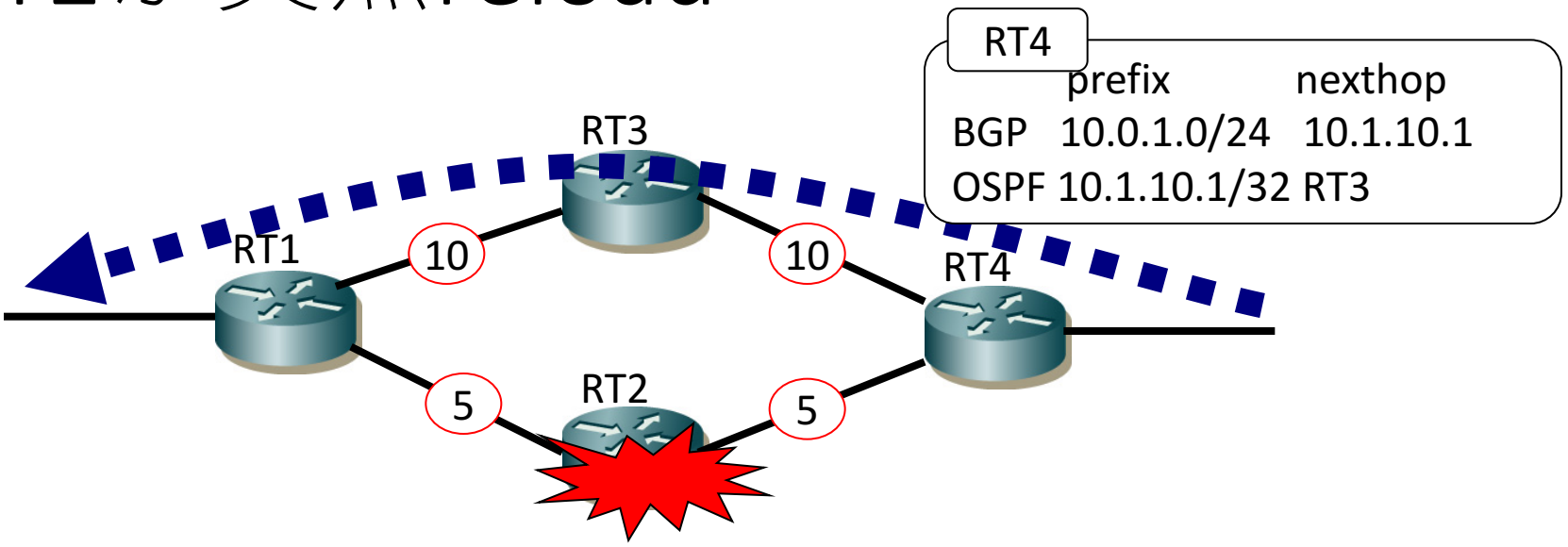
- 名古屋からは65502(東京経由)に更新

OSPFコストとBGP



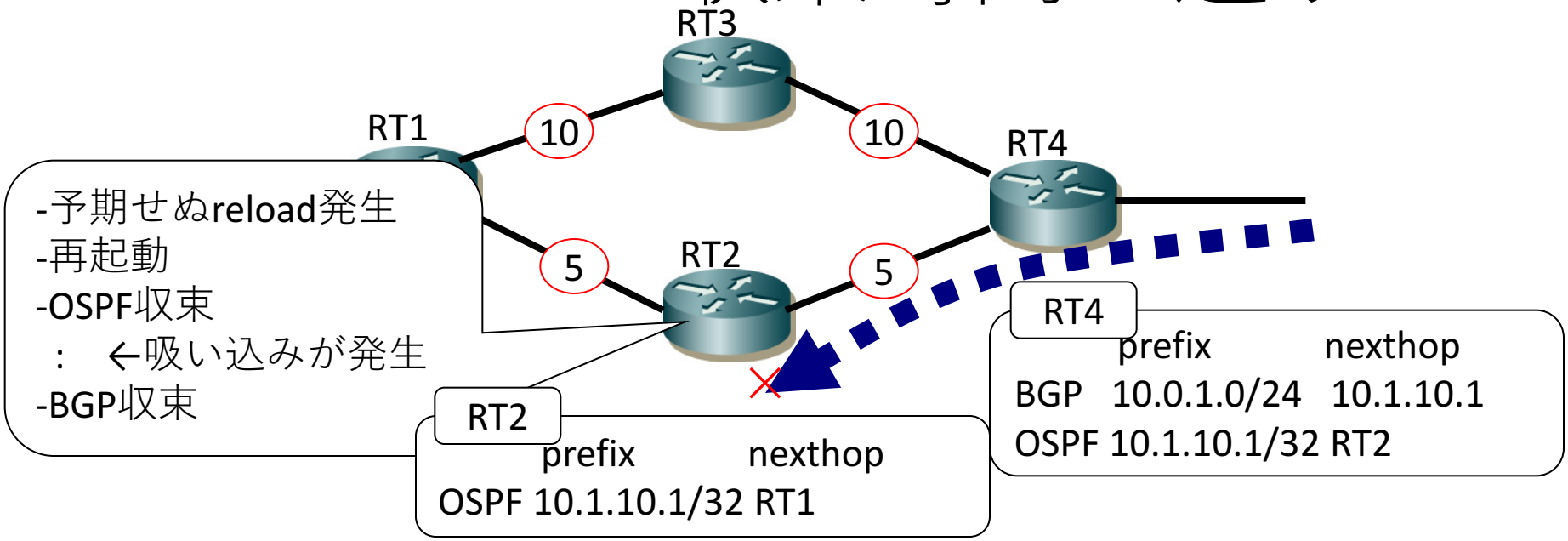
- BGPネクストホップへのOSPFコストが一番小さな経路が選択される

RT2が突然reload



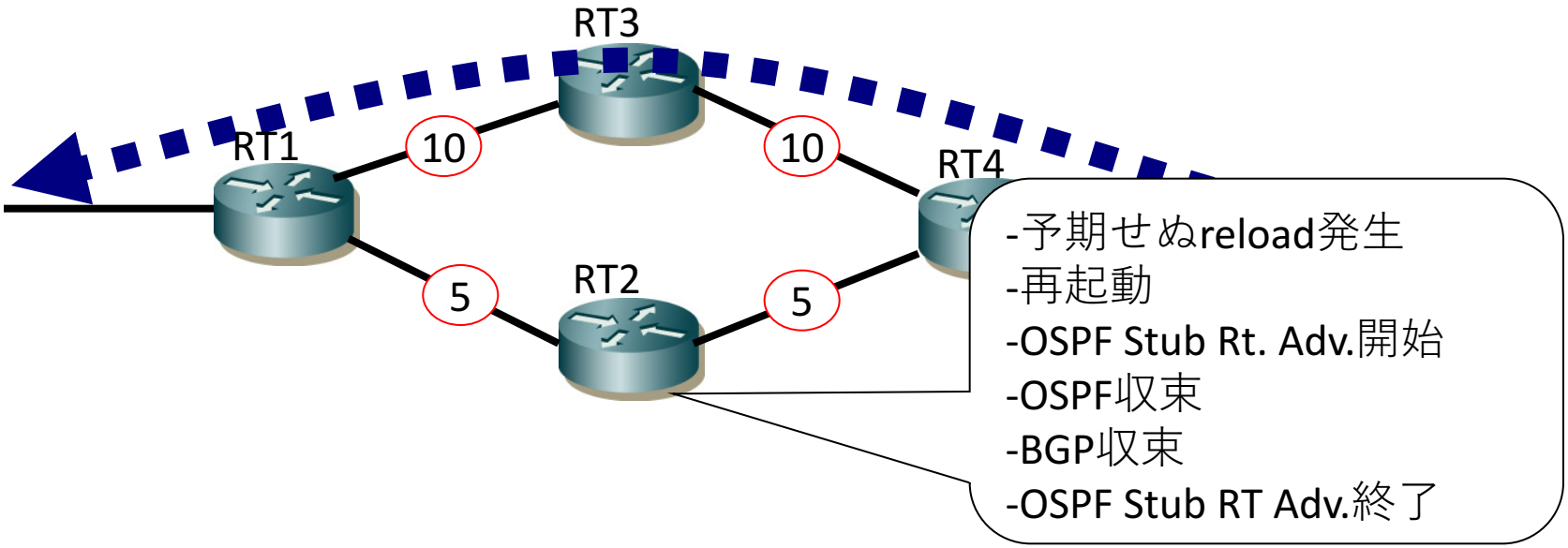
- RT２が再起動・・・
- 他のルータが障害を検出し、OSPF再計算
- トラヒックはRT３を迂回している

OSPFとBGPの収束時間が違う



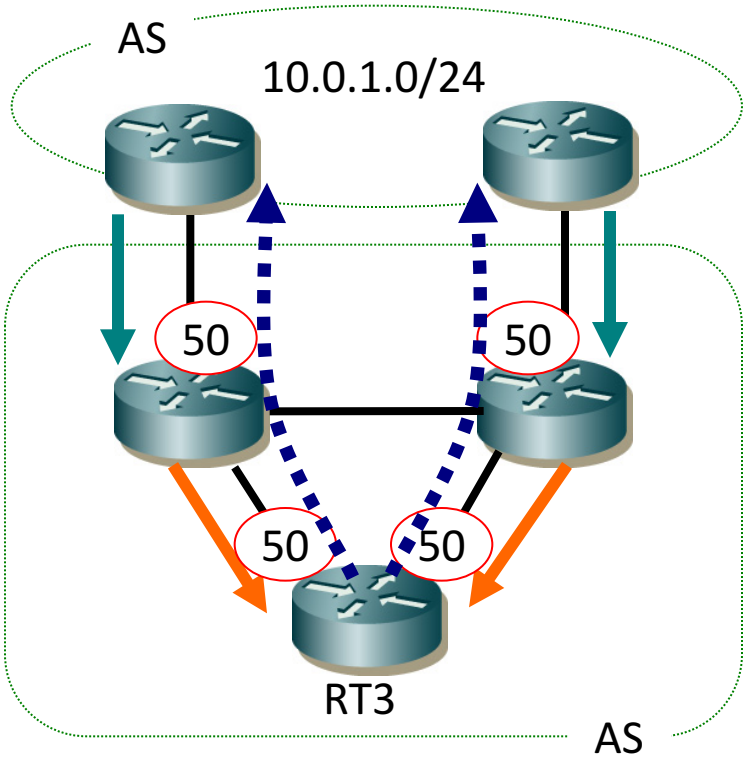
- OSPFは収束したので、RT4ではRT2側を選択
- RT２はまだBGP経路を受信しきっていない
- その間、RT2がトラヒックを破棄してしまう

OSPF StubRouterAdvertisement



- ルータを経由するトラヒックを迂回させる機能
- OSPF起動後に実施して、BGP収束までトラヒックを迂回させる等の利用が考えられる
- 詳しくは[RFC3137]を参照

BGP Multipath



- 複数の経路を有効にできる手法
  - ベンダの実装依存
  - 経路選択で特定の段階まで優先度が一致すればMultipathとして扱う
- RT3でMultipathを使用
  - RT3が他のルータに広報する経路は通常選択される１つのベスト経路のみ

BGP4+

- BGP4のマルチプロトコル(IPv6)対応
  - [RFC2545] [RFC2858]
- OPENメッセージでマルチプロトコル対応を通知
- BGPセッションはIPv4 or IPv6どちらでも可
  - IPv6だとglobal unicast or link-localが選べる
  - IPv6の到達性を保証するには、IPv6でセッションを確立するのがお勧め
- NEXT_HOPは global unicast (+ link-local)
  - プレフィックスと共にMP_REACH_NLRIで運ばれる

BGPの転用

- BGPは、ルータにTCPで情報を通知できる
- パス属性で情報を運ぶ
  - IPv6経路等もパス属性で運ばれる

  ∴パス属性のみでNLRIが無いUPDATEも有効
- 経路を運ぶ以外の目的にも利用されるようになってきた

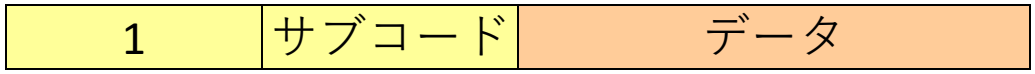
BGP NOTIFICATION メッセージ

BGP NOTIFICATIONメッセージ

1. メッセージヘッダエラー
2. OPENメッセージエラー
3. UPDATEメッセージエラー
4. HoldTime超過
5. 状態遷移エラー
6. Cease
7. ROUTE-REFRESHエラー

コード1: メッセージヘッダエラー

- メッセージヘッダの処理中にエラーを検出



サブコード	エラー内容	データに含まれる内容
0	特定なし	
1	Markerの値が不正	
2	Lengthの値が不正	そのLengthの値
3	解釈できないタイプ	そのタイプの値

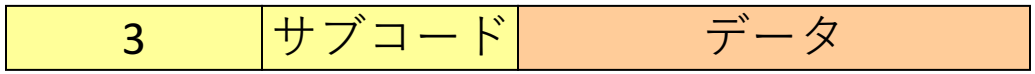
コード2: OPENメッセージエラー

- OPENメッセージの処理中にエラーを検出

サブコード	エラー内容	データに含まれる内容
0	特定なし	
1	バージョン不一致	サポートする最も近いバージョン
2	AS番号でエラー	
3	BGP IDが不正	
4	解釈できないオプション	
5	[Deprecated]	
6	ホールドタイマ値に対応できない	
7	サポートしていないCapability	そのCapabilityコード

コード3: UPDATEメッセージエラー

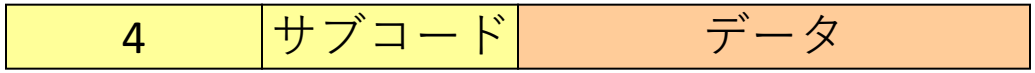
- UPDATEメッセージの処理中にエラーを検出



サブコード	エラー内容	データに含まれる内容
0	特定なし	
1	アトリビュートが不正	
2	周知必須属性が解釈できなかった	エラーを検出した属性値データ
3	周知必須属性が不足している	不足していた属性値のタイプ
4	コードフラグが不正	エラーを検出した属性値データ
5	パス属性値が不正	エラーを検出した属性値データ
6	ORIGIN属性値が不正	エラーを検出した属性値データ
7	[Deprecated]	
8	NEXT_HOP属性値の書式が不正	エラーを検出した属性値データ
9	オプション属性値でエラー	エラーを検出した属性値データ
10	NLRIの書式が不正	
11	AS_PATH属性値が不正	

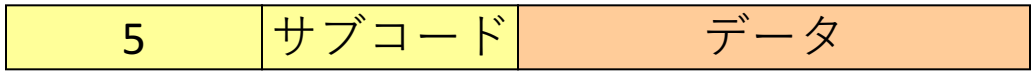
コード4: HoldTimer超過

- HoldTimer期間中に、UPDATEもKEEPALIVEも受信しなかった



コード5: 状態遷移エラー

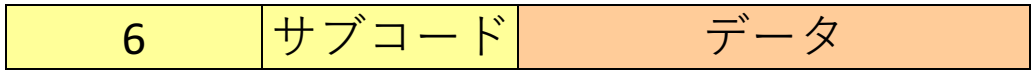
- 予期せぬイベントが発生



サブコード	エラー内容	データに含まれる内容
0	特定なし	
1	Open状態でエラー発生	
2	OpenConfirm状態でエラー発生	
3	Established状態でエラー発生	

コード6: Cease

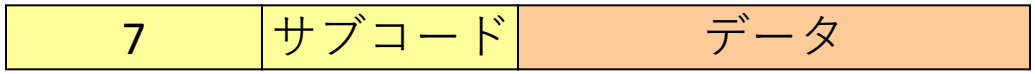
- その他のエラーを検出



サブコード	エラー内容	データに含まれる内容
1	最大受信経路数に到達	AFI, SAFI, prefix上限値
2	Administrative Shutdown	
3	設定削除	
4	Administrative Reset	
5	接続拒否	
6	その他の設定変更	
7	接続競合の解決	
8	リソース不足	

コード7: ROUTE-REFRESHエラー

- Route Refreshでエラーが発生



サブコード	エラー内容	データに含まれる内容
0	特定なし	
1	メッセージ長が不正	

BGPパス属性値
コードタイプ

BGPパス属性値コードタイプ

属性値	タイプ	概要
1 ORIGIN	周知必須	経路の生成情報
2 AS_PATH	周知必須	経路が通過したASの情報
3 NEXT_HOP	周知必須	経路のフォワード先IPアドレス
4 MULTI_EXIT_DISC	オプション非通知	複数出口から経路選定する際の優先度
5 LOCAL_PREF	周知任意	経路のローカル優先度
6 ATOMIC_AGGREGATE	周知任意	BGP経路が途中で集約された情報
7 AGGREGATOR	オプション通知	経路集約を行なったルータ
8 COMMUNITIES	オプション通知	経路に付加するタグ情報

BGPパス属性値コードタイプ続き

属性値	タイプ	概要
9 ORIGINATOR	オプション非通知	クラスタ内での経路生成ルータ
10 CLUSTER_LIST	オプション非通知	経路を反射したクラスタIDのリスト
14 MP_REACH_NLRI	オプション非通知	マルチプロトコルの到達可能経路
15 MP_UNREACH_NLRI	オプション非通知	マルチプロトコルの到達不可能経路
16 EXTENDED COMMUNITIES	オプション通知	拡張されたCOMMUNITIES(主にVPN)
17 AS4_PATH	オプション通知	古い実装で4ByteAS情報を通過させる
18 AS4_AGGREGATOR	オプション通知	古い実装で4ByteAS情報を通過させる
32 LARGE_COMMUNITY	オプション通知	経路に付加するタグ情報