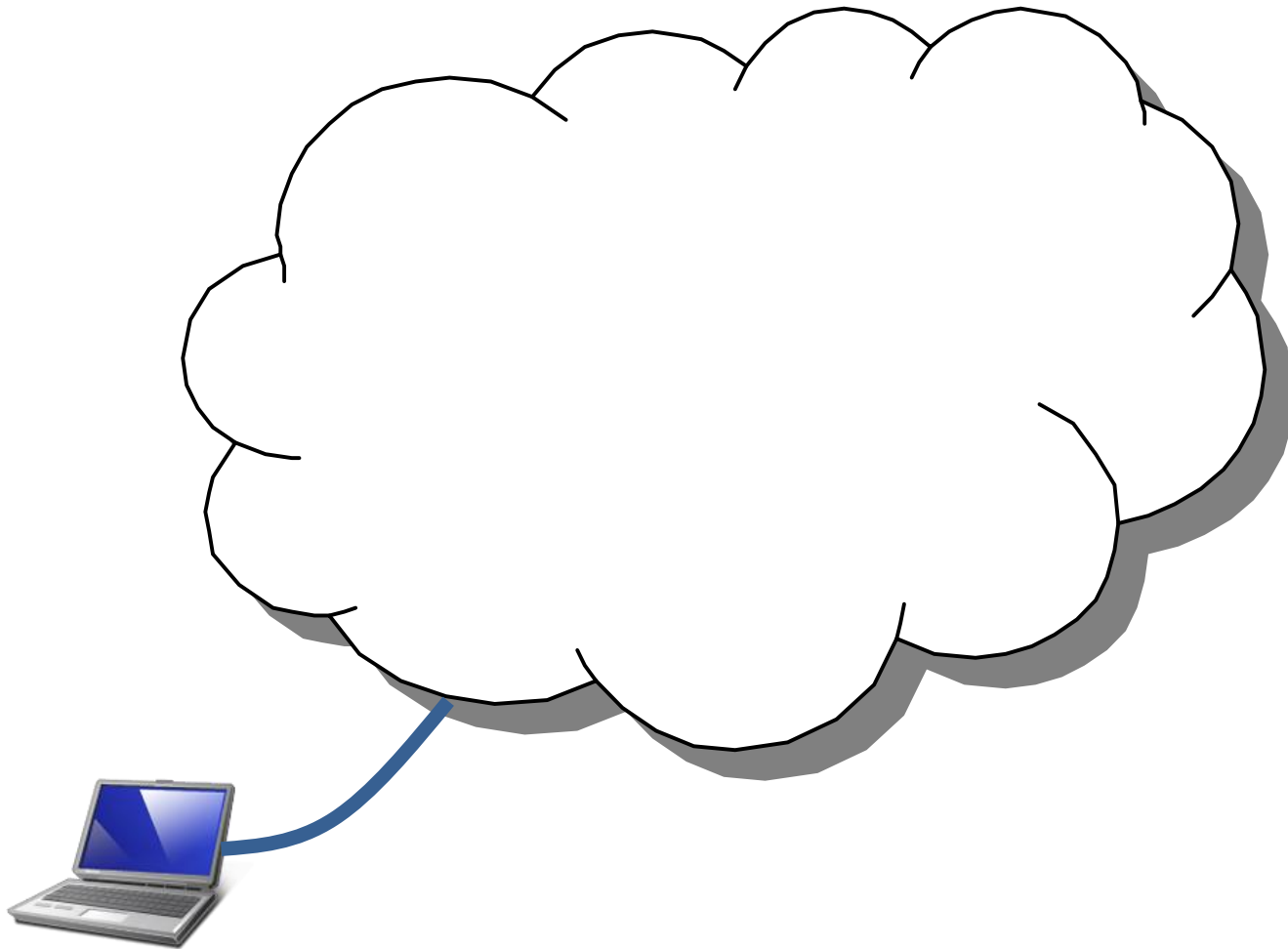


# JANOG routingチュートリアル

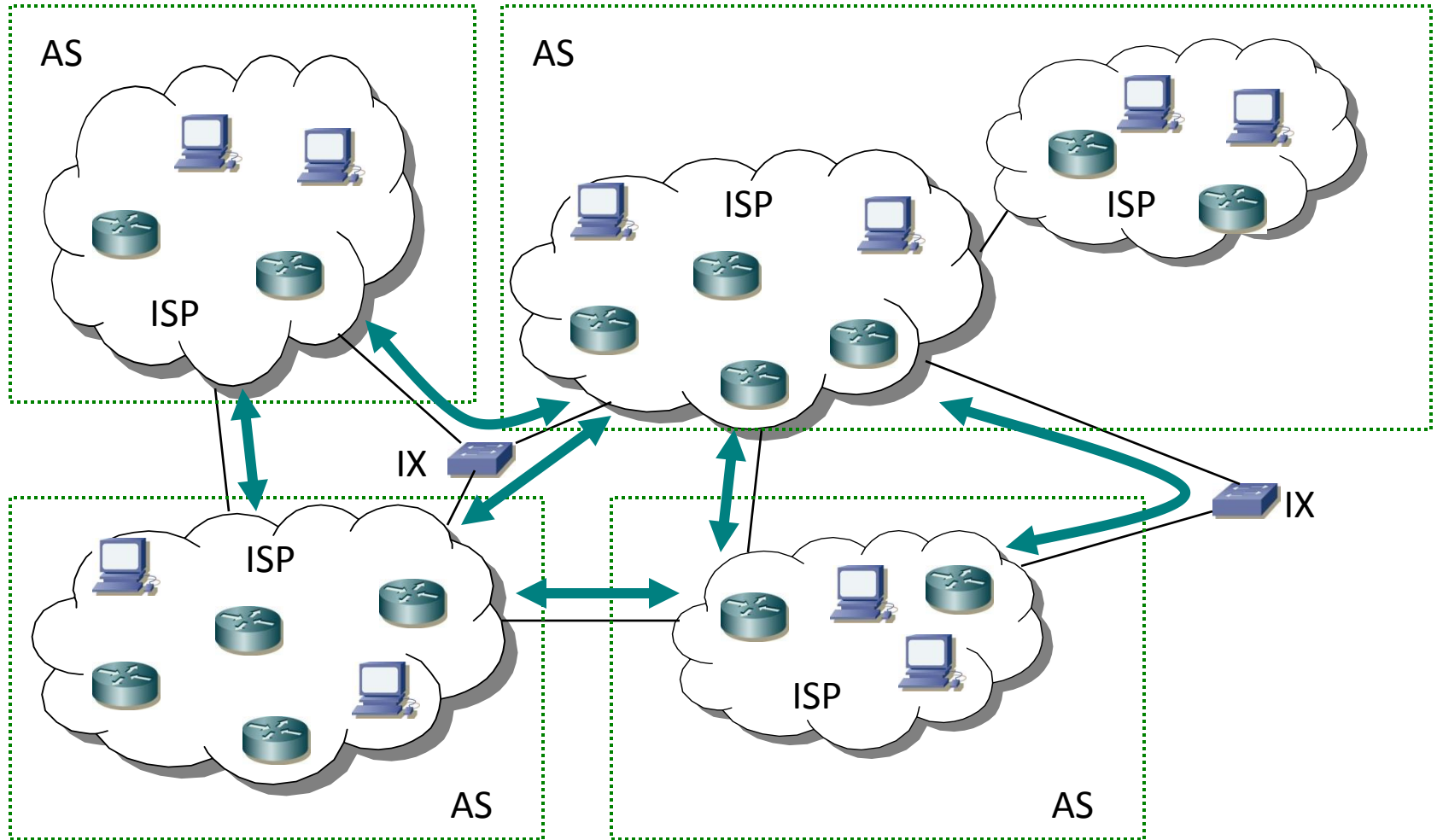
Matsuzaki 'maz' Yoshinobu

<maz@ij.ad.jp>

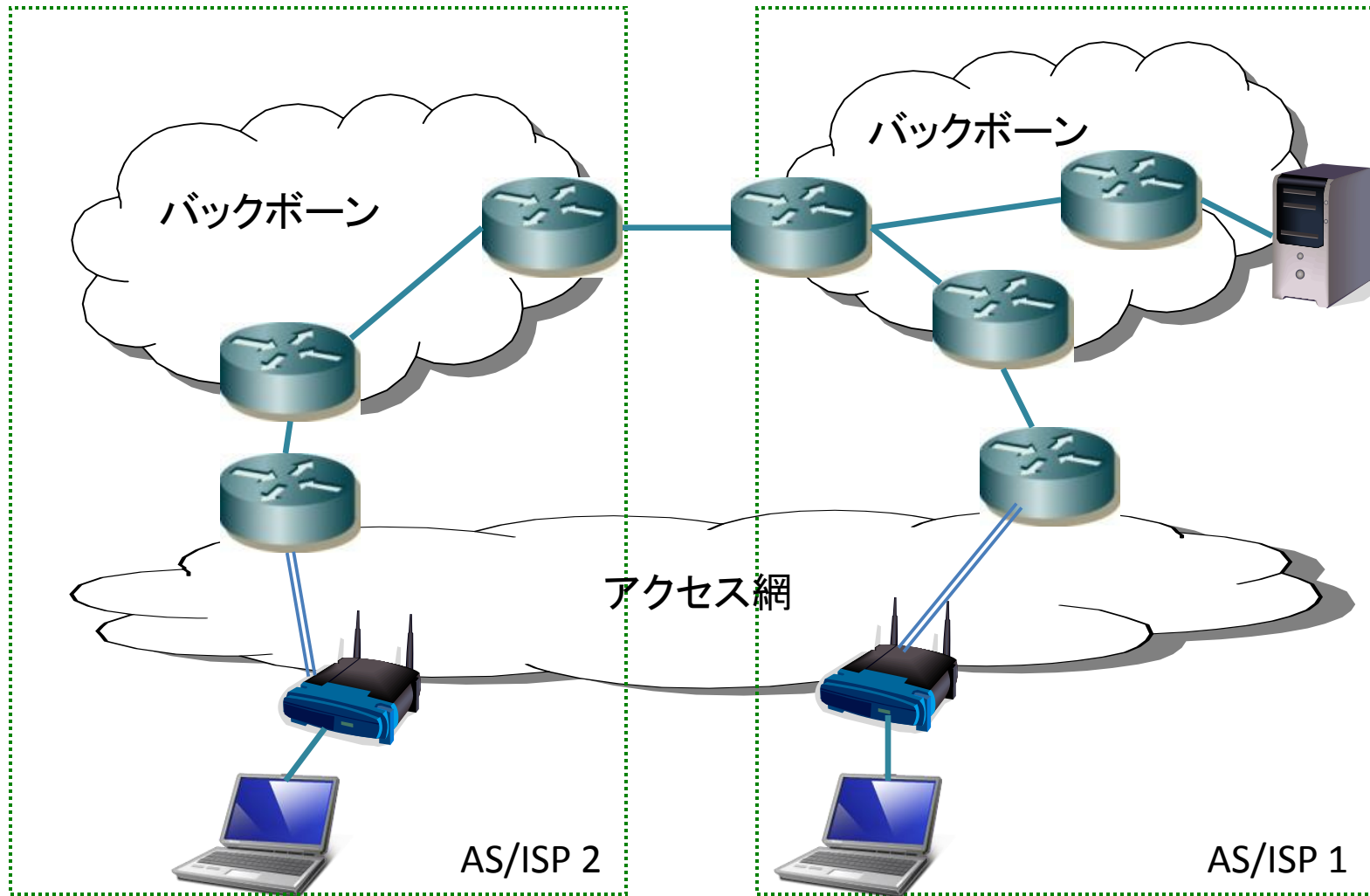
# インターネット



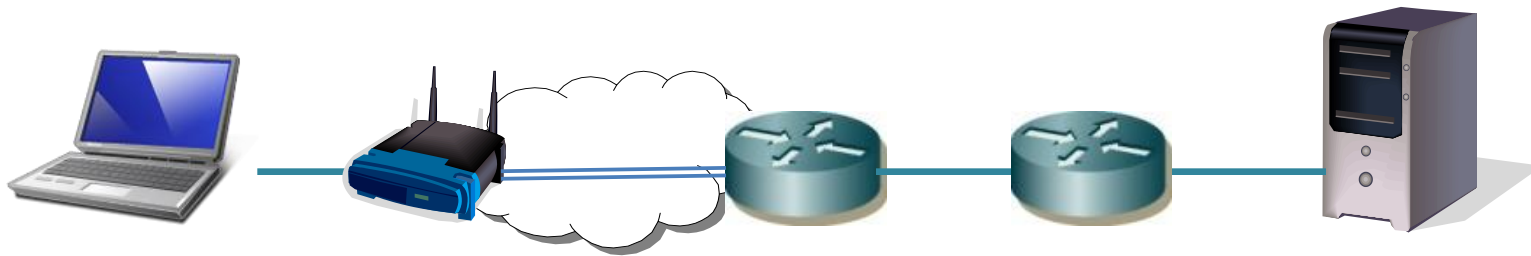
# ネットワーク



# アクセス網とバックボーン網



# ホスト、回線、ルータ



# ホスト

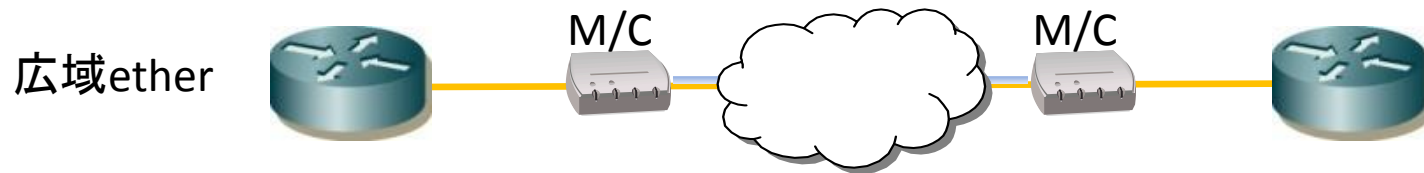
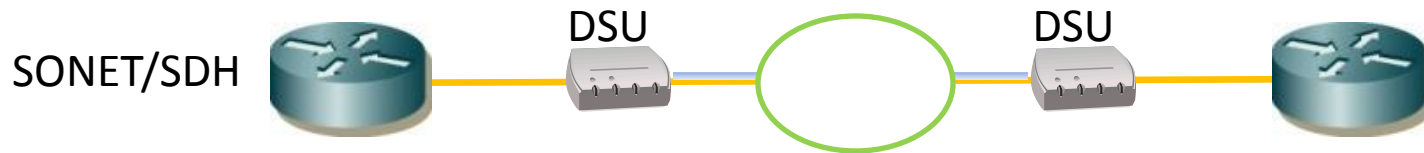
- IPで通信したい人たち
  - PC、ゲーム、PDA、テレビ
- それぞれネットワークに接続するためのインターフェースを持つ
  - イーサネット
  - 無線LAN、無線WAN
  - シリアル、パラレル、USB



# 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
- 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





# 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
- インタフェースや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 (6micro sec/packet)
- とある大きなルータ
  - 770,000,000pps (1pico sec/packet)
- 専用ハードウェアによる高速化

# 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
- 日々のトラフィックを運びつつも、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
  - いろんなパターンが考えられる
- 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ー
  - 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
  - 痒いところを搔いてくれる(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
  - 機能の組み合わせによる場合分けが破綻した
- 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
  - 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
- ルータはパケットの宛先アドレスをみて次の送り先を判断する

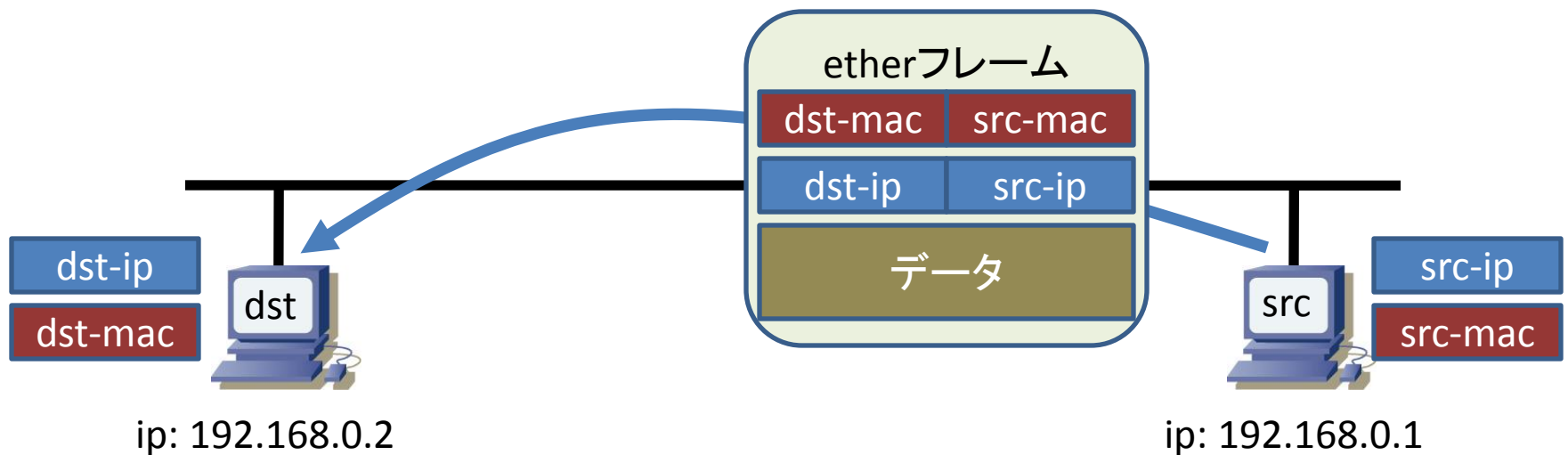
# IPv4パケット送信

- 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

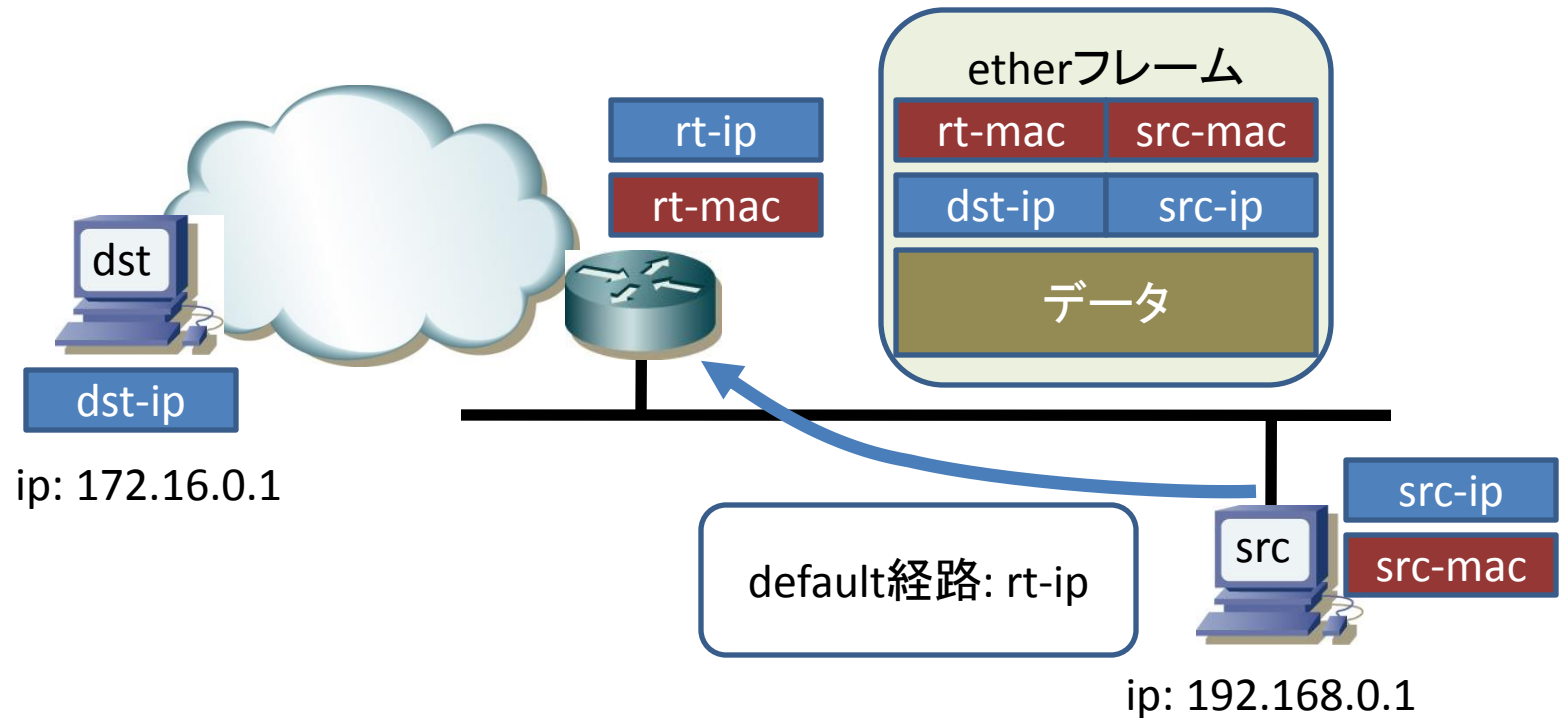


192.168.0.0～192.168.0.255が同じセグメント上にある



# 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)

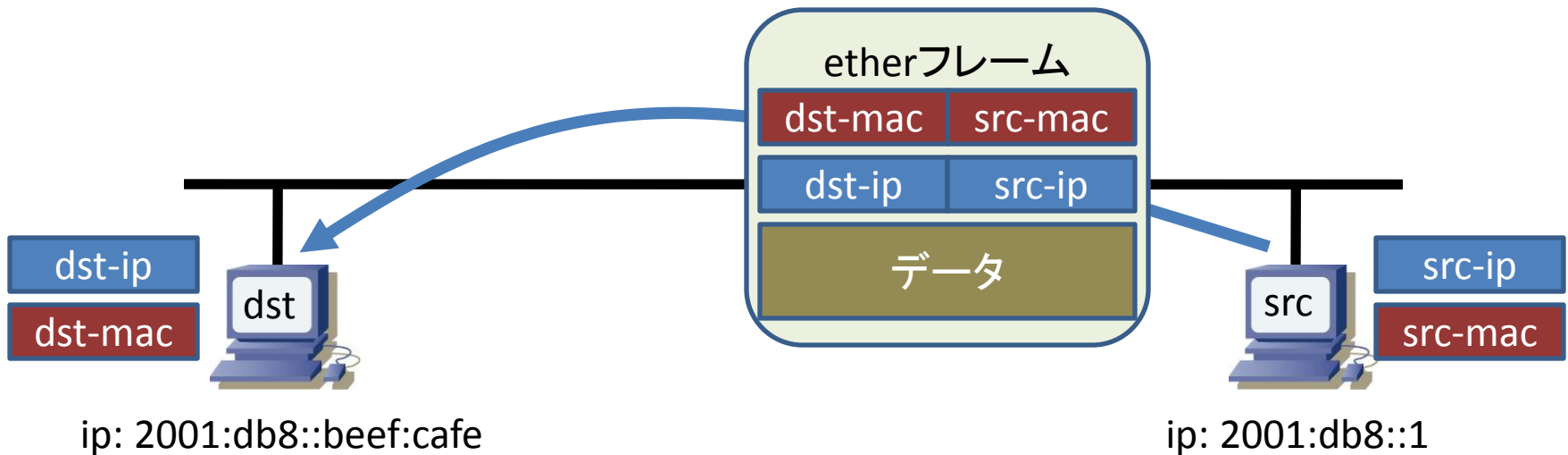
- etherではパケット送信にMACアドレスが必要
  - IPv4アドレスは分かっている (ex. defaultの向け先)
  - 機器のIPv4アドレスからMACアドレスを知りたい
- arpで解決
  - RFC826

```
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

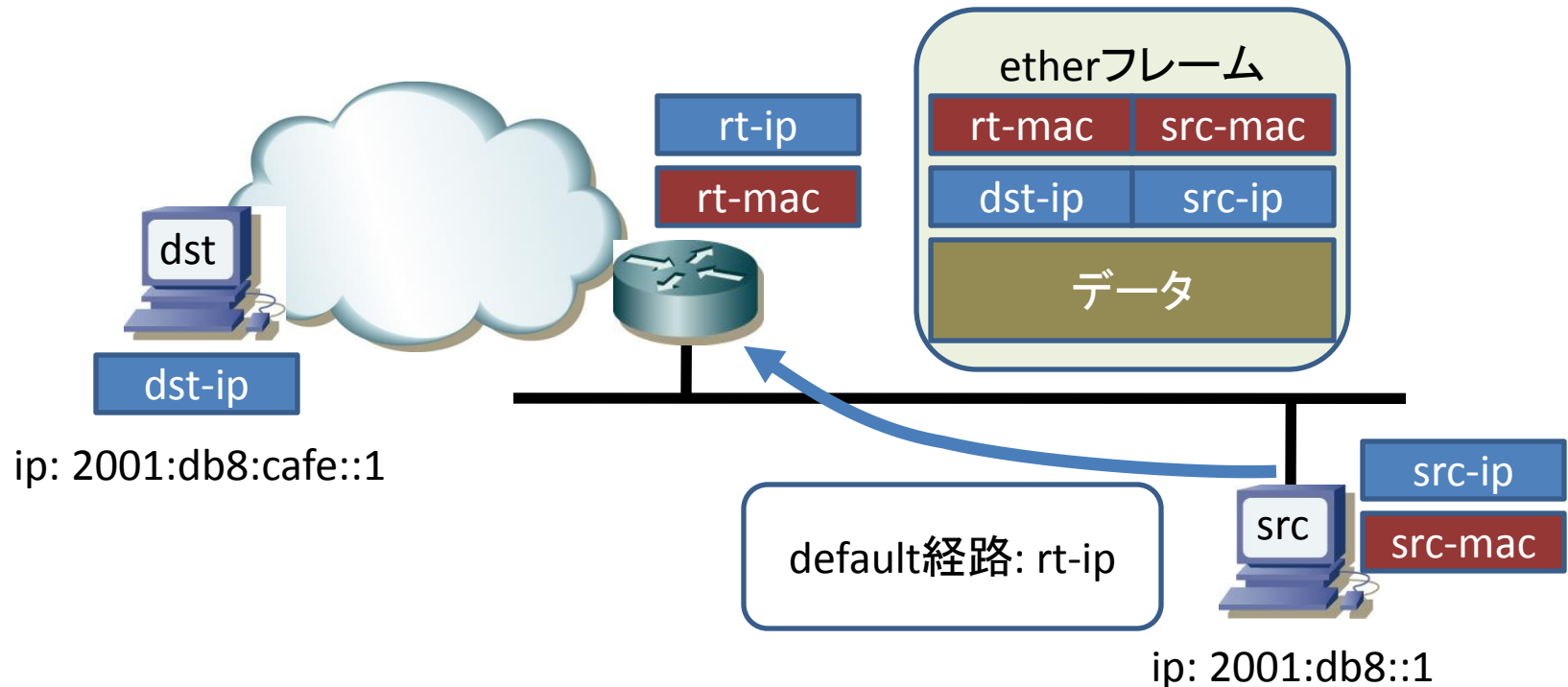
- 同じネットワークに属していれば直接送信

inet6 2001:db8::1 prefixlen 64  
↓  
2001:db8::~2001:db8::ffff:ffff:ffff:ffffが同じセグメント上にある



# 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 ffe0 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 ffe0 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
```

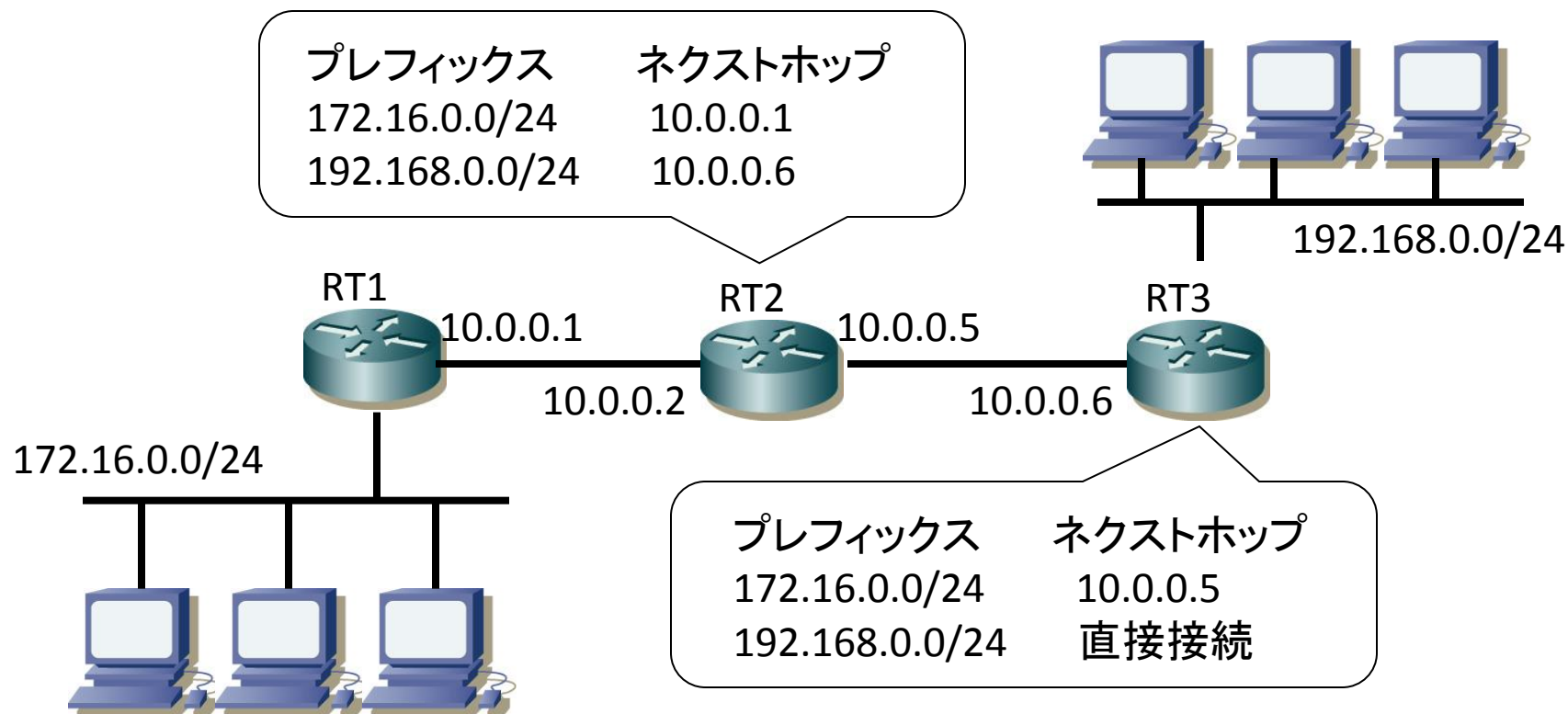
```
0x0000: 0019 bb27 37e0 0016 1761 6486 86dd 6000  
0x0010: 0000 0020 3aff 2001 0db8 0000 0000 0000  
0x0020: 0000 beef cafe 2001 0db8 0000 0000 0000  
0x0030: 0000 0000 0001 8800 c1fd 6000 0000 2001  
0x0040: 0db8 0000 0000 0000 0000 beef cafe 0201  
0x0050: 0016 1761 6486
```

# ちなみにpoint-to-pointリンクの場合

- SDH/SONET/PPPとか
- 回線の先には必ず通信相手が一台だけ
- arp/ndpなどは利用しません
  - 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)

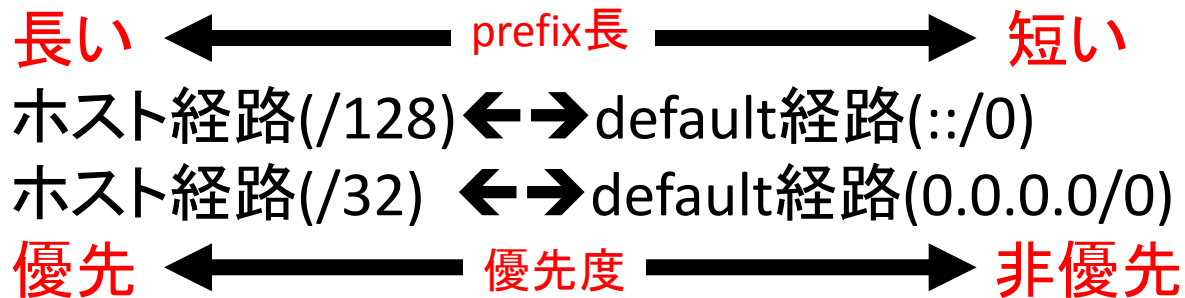
# 経路情報

- 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

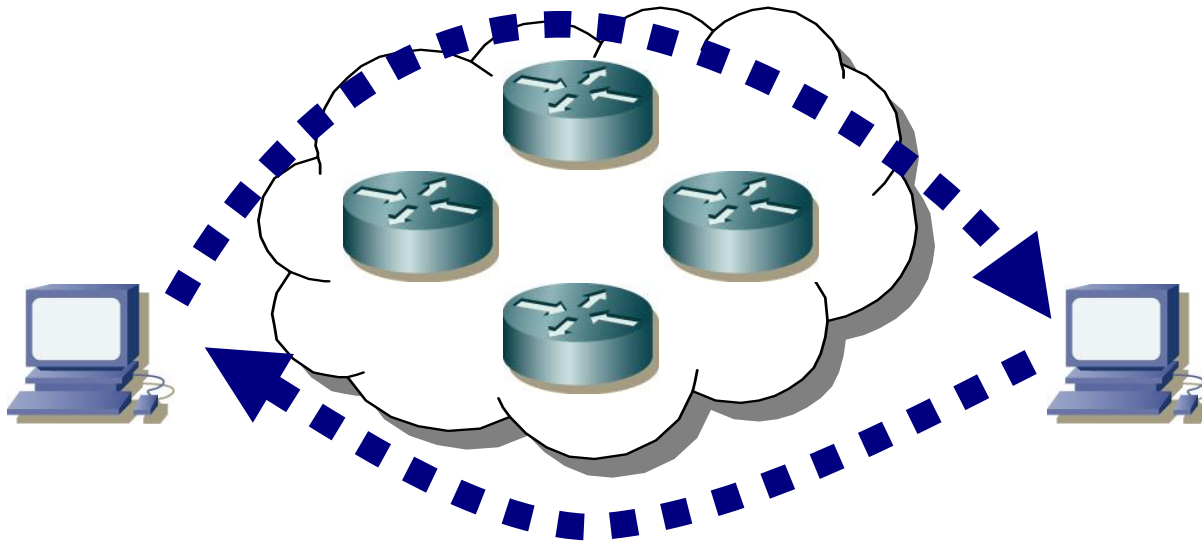


# 経路の種類

- 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

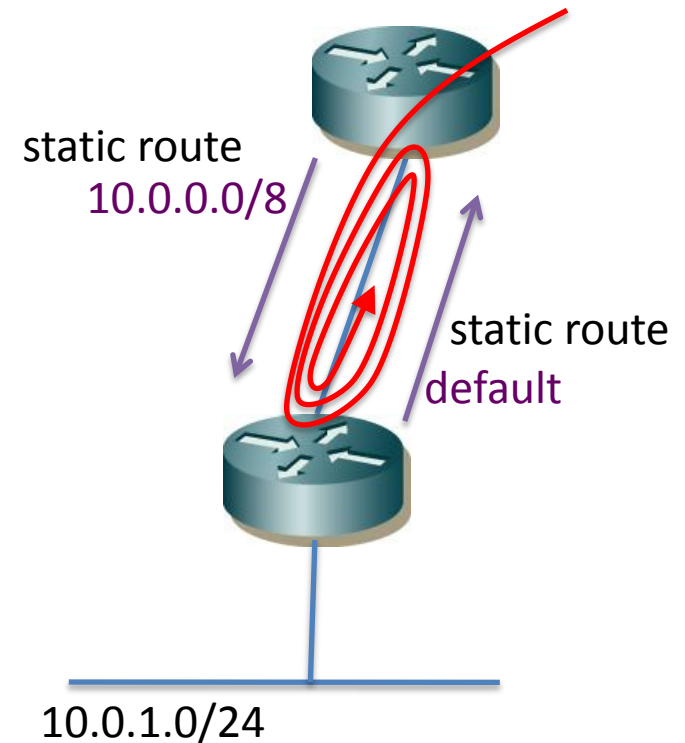
# パケットと経路

- 送信元から宛先まで経路に矛盾が無ければ、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 route
  - 無茶な設定の動的経路制御



# 動的経路制御

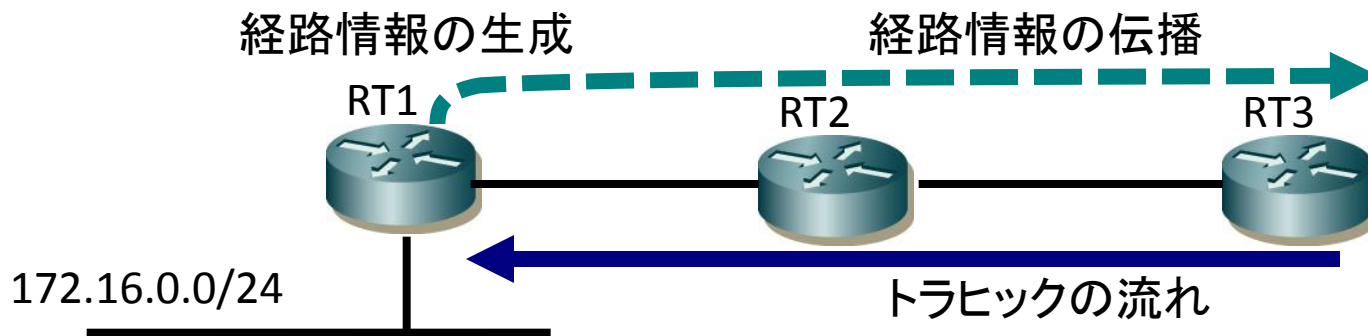
## インターネットと動的経路制御

# 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
  - もちろん事前の設計は必要
- 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
  - 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

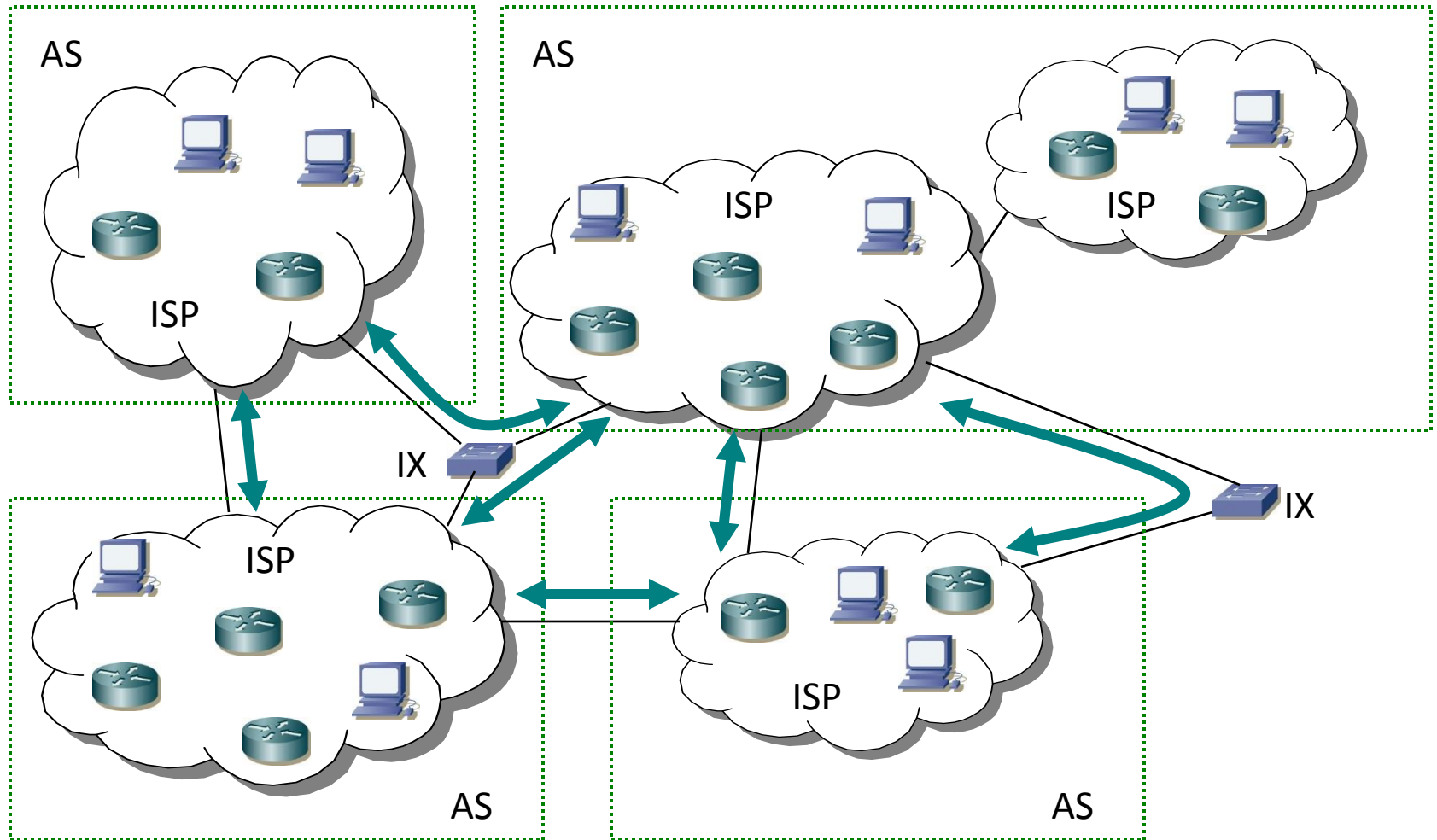


経路情報の伝搬の方向とトラヒックの流れは逆になる

# 動的経路制御の種類

- ディスタンスベクタ(distance vector)
  - RIPなど、距離と方向を扱うプロトコル
- リンクステート(link state)
  - OSPFやIS-ISなど、リンクの状態を収集して管理するプロトコル
- パスベクタ(path vector)
  - BGPなど、パス属性と方向を扱うプロトコル

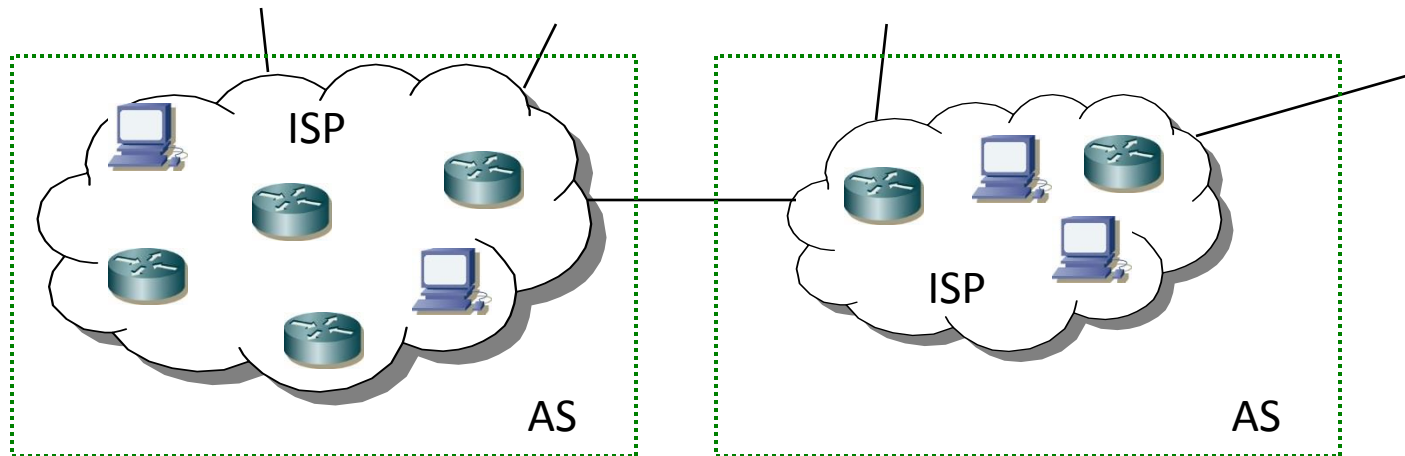
# インターネットの構成





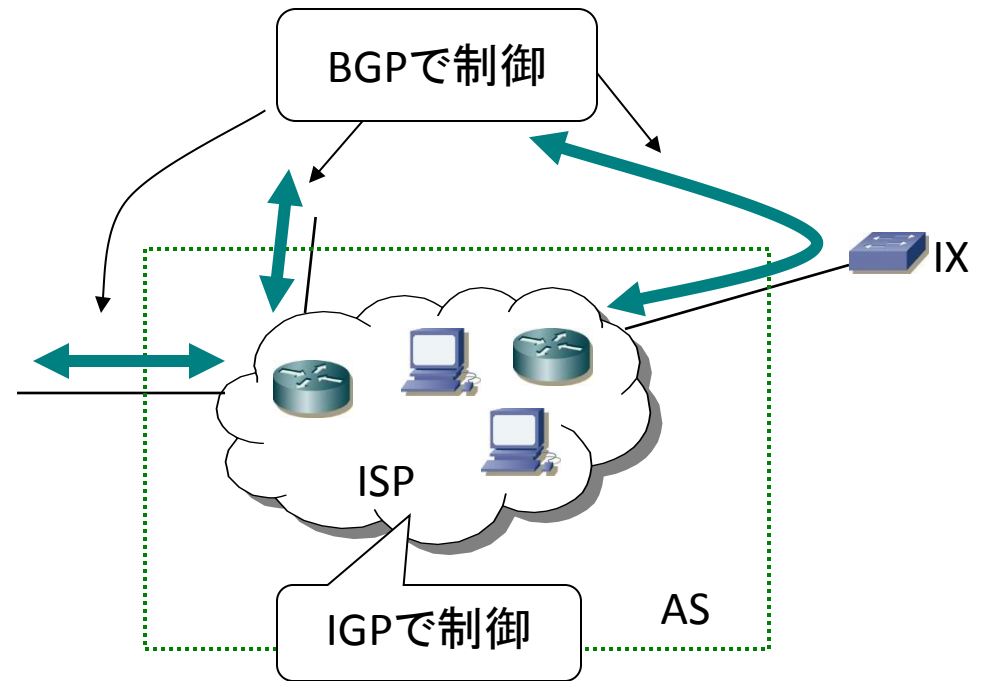
# AS

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



# IGPとEGP

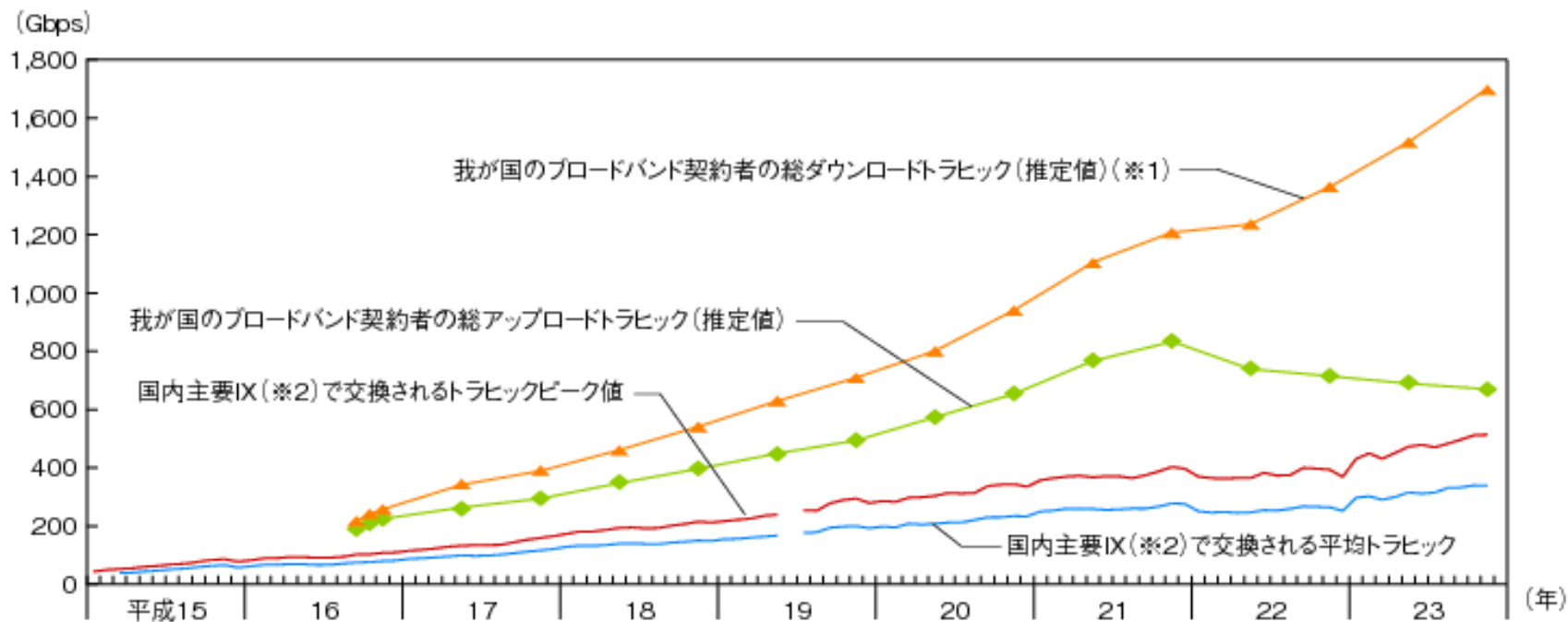
- IGP
  - OSPF、IS-IS、BGP等
  - AS内
- EGP
  - 事実上BGPのみ
  - AS間



# 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

# トラフィック増加への対応



※1 1日の平均トラフィックの月平均。

※2 平成19年6月の国内主要IXで交換されるトラフィックの集計値についてはデータに欠落があったため除外。

総務省:情報通信白書 24年版 より

<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc245320.html>

# 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がようやく出て来たけどまだ高い
- 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

# link aggregation

- 10Gbpsの回線を束ねて、ルータで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

# multipath

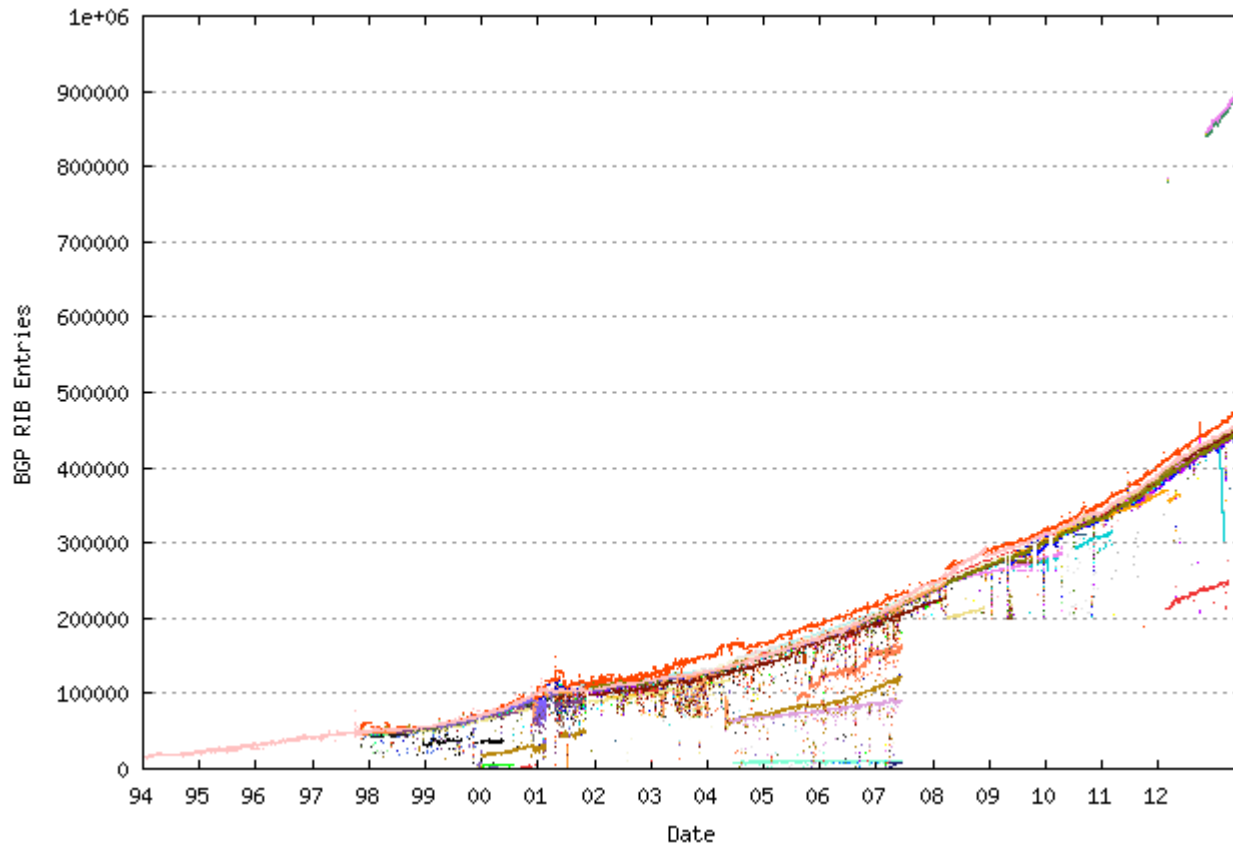
- OSPF Multipath
  - ISP(AS)内には有効
  - 標準技術
- BGP Multipath
  - 非標準技術だが、多くのベンダが採用
  - 構成をきちんと組めば、ISP(AS)間にも有効
- 帯域の利用効率が良い

# より高速なインタフェース

- 100Gbpsインタフェースを備えたルータが市場に出て来たが・・・
  - お値段が高い
  - ポート密度が上がるまで時間がかかる

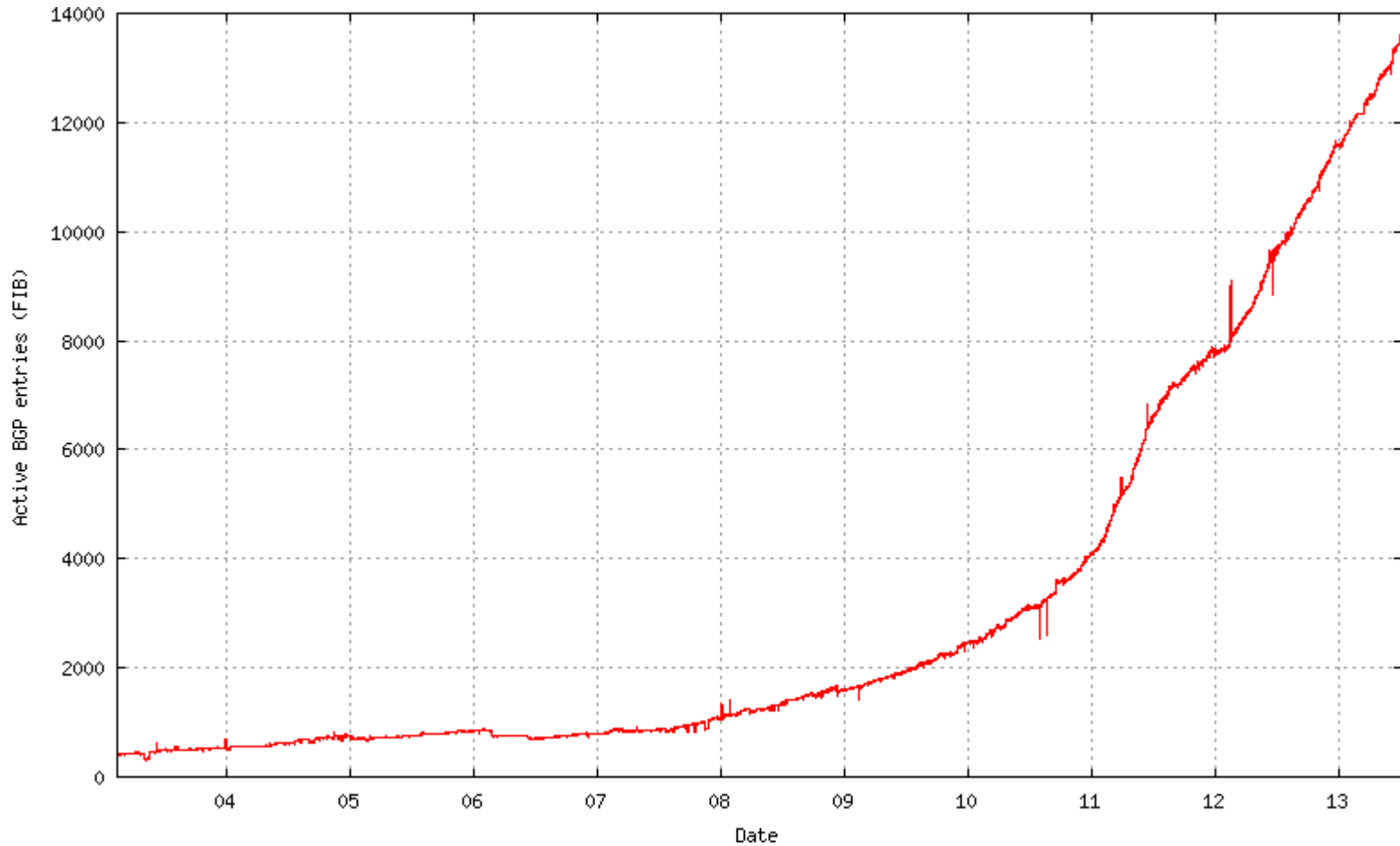


# 経路数増加への対応



<http://bgp.potaroo.net/bgprpts/rva-index.html>

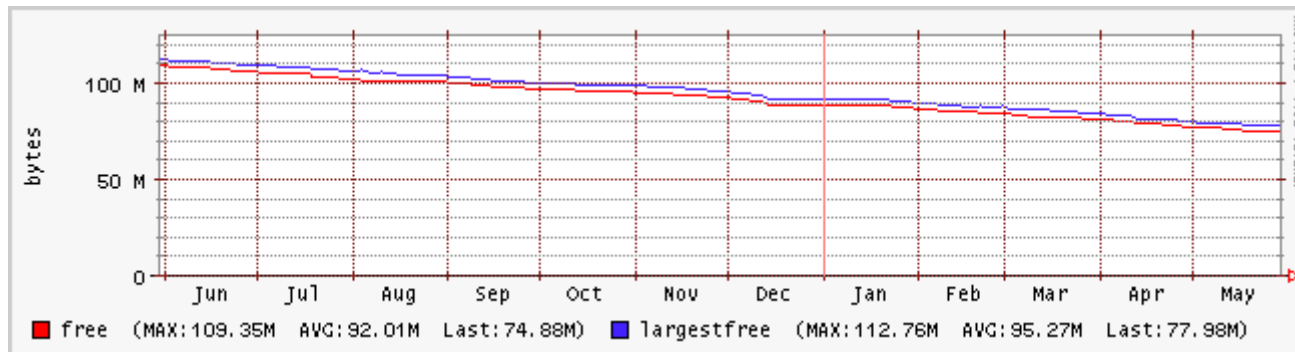
# IPv6経路も増加中



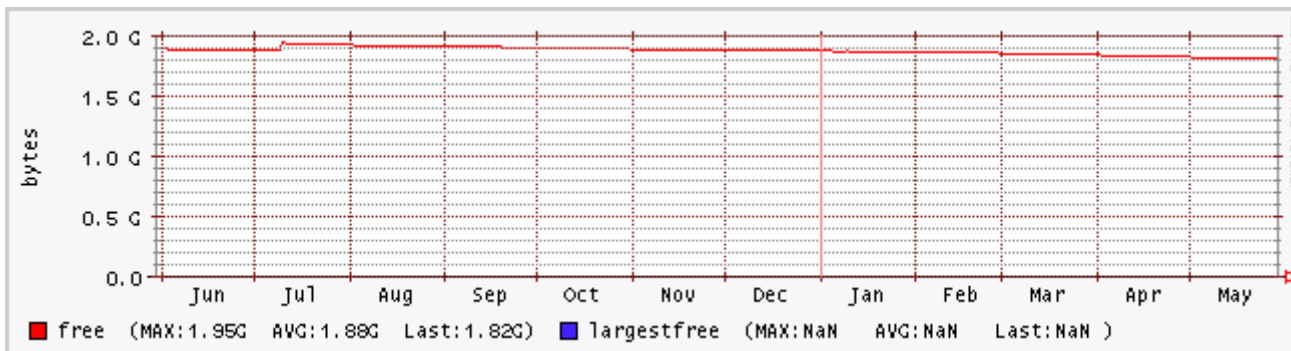
<http://bgp.potaroo.net/index-v6.html>

# 減りゆくルータのメモリ

- 512MBメモリ



- 4GBメモリ



# 経路増加にはグッとくる解決無し

- default経路併用による運用
  - 小規模ルータで経路削り
- 泣きながら増強
  - BGPでトランジット提供している場合

OSPF

# OSPF概要

- リンクステート型
  - 全ルータがネットワークのトポロジ情報を持つ
  - ネットワークに変更があれば通知
- SPFアルゴリズムによる最適経路の選択
  - リンクのコストによる優先付け
  - 同一コストの複数パスによる負荷分散
- エリアによる階層化
  - エリア境界はルータ
- 隣接のルータと情報を交換

# OSPFの基本アイデア

- 準備
  - 隣接した他のOSPFルータと隣接関係を構築
- 通知
  - 各ルータが必要な情報をタイプ別にLSAとして生成
  - 隣接関係のルータにLSAを送信
  - 受信したLSAをさらに他のルータにfloodして網内に伝播
- 構成
  - 各ルータが全LSAを集め、LSDBとして保持
  - 各ルータでLSDBを元にSPF計算して最短パスを求め、経路情報を生成

# OSPF RFCs

- 必読
  - [RFC2328] OSPF Version 2
- この他にもいっぱい
  - [RFC2370] The OSPF Opaque LSA Option
  - [RFC2740] OSPF for IPv6
  - [RFC3101] The OSPF NSSA Option
  - [RFC3137] OSPF Stub Router Advertisement
  - ：



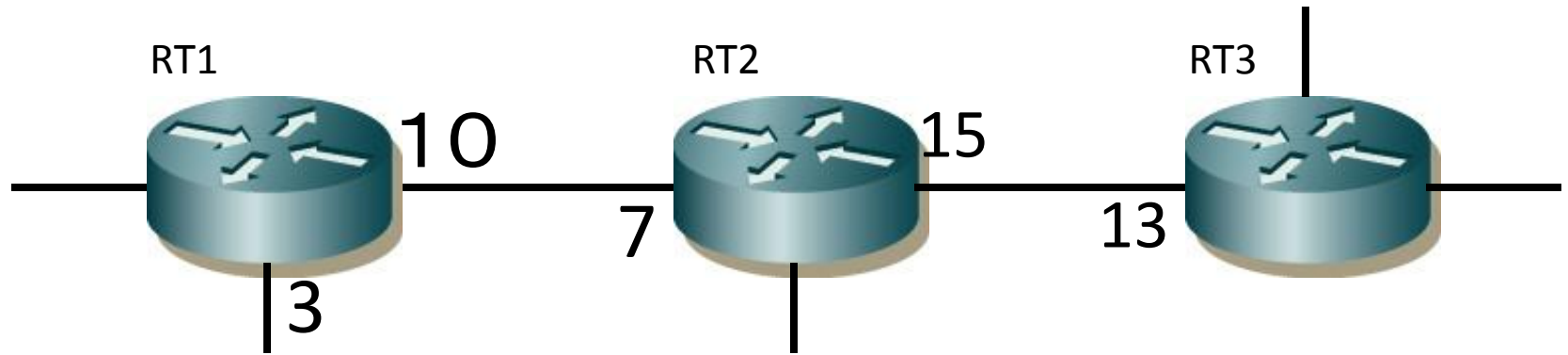
# OSPF用語

- ルータID
  - OSPFのAS内でルータを識別する32bitの数値
  - 特に指定が無い場合、ルータのインタフェースのIPアドレスから選ぶことが多い
  - ルータIDを変更する場合は、OSPFプロセスの再起動が必要なため、実運用では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

# OSPFの経路選択

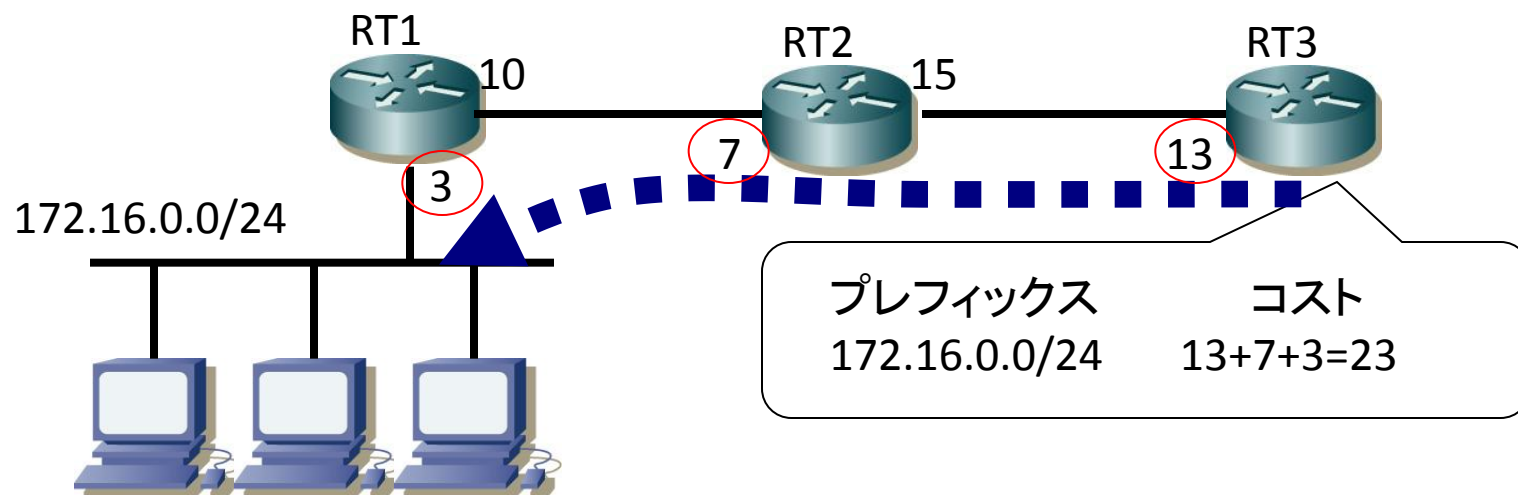
## OSPFとコスト

# リンクコスト(link cost)



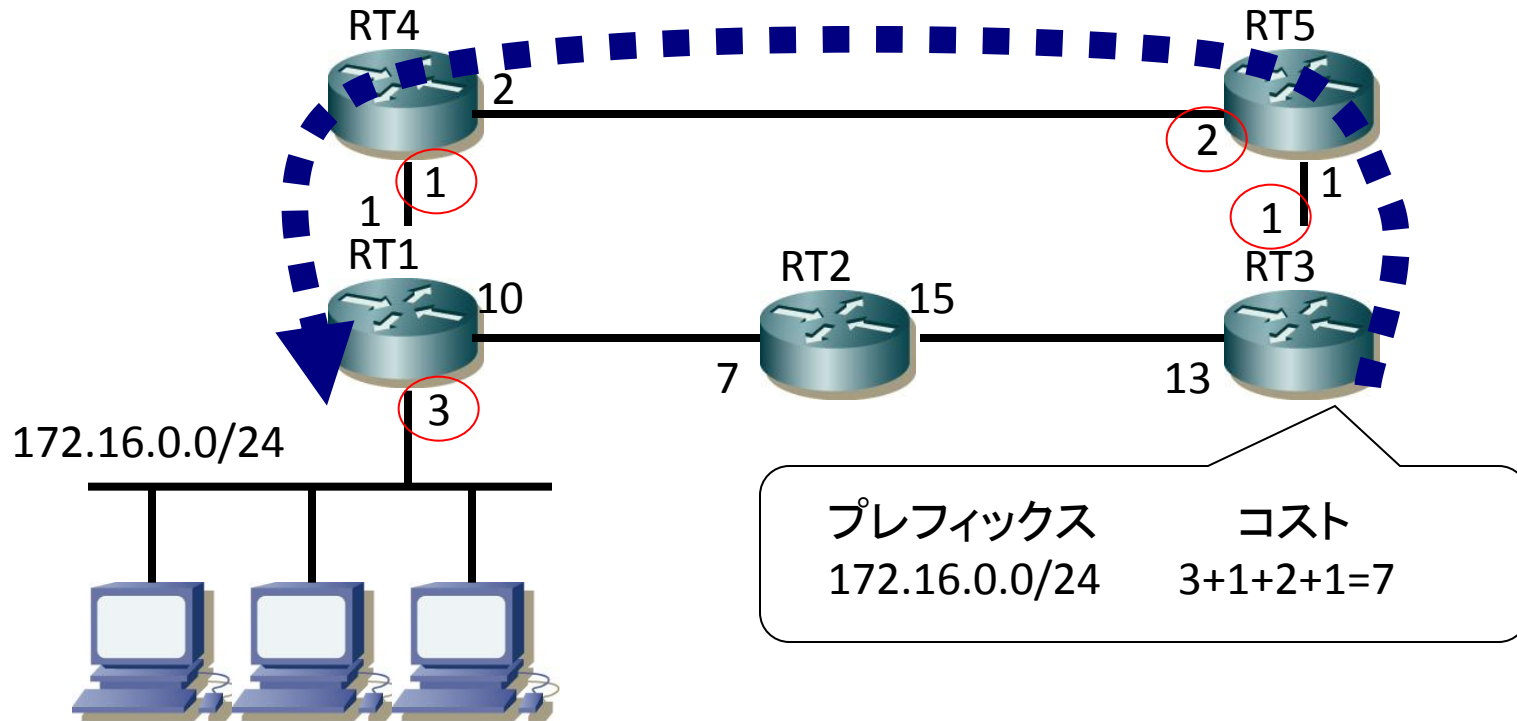
- ルータが、そのインタフェースからパケットを送出するときのコスト(負担)
- 1~65535の整数を管理者が設定する

# コスト(cost)



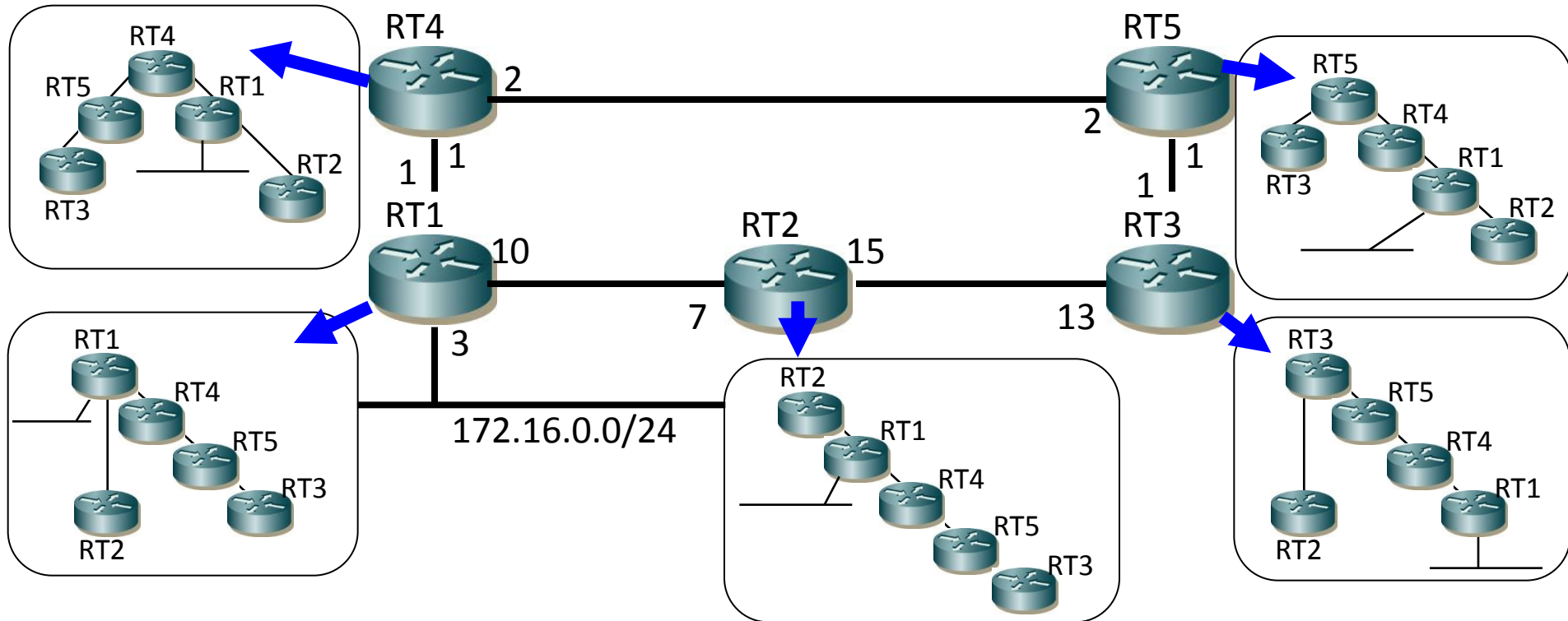
- 宛先までのパスで、パケットが出力されるインタフェースのリンクコストを合計した値

# 小さいコストのパスが優先



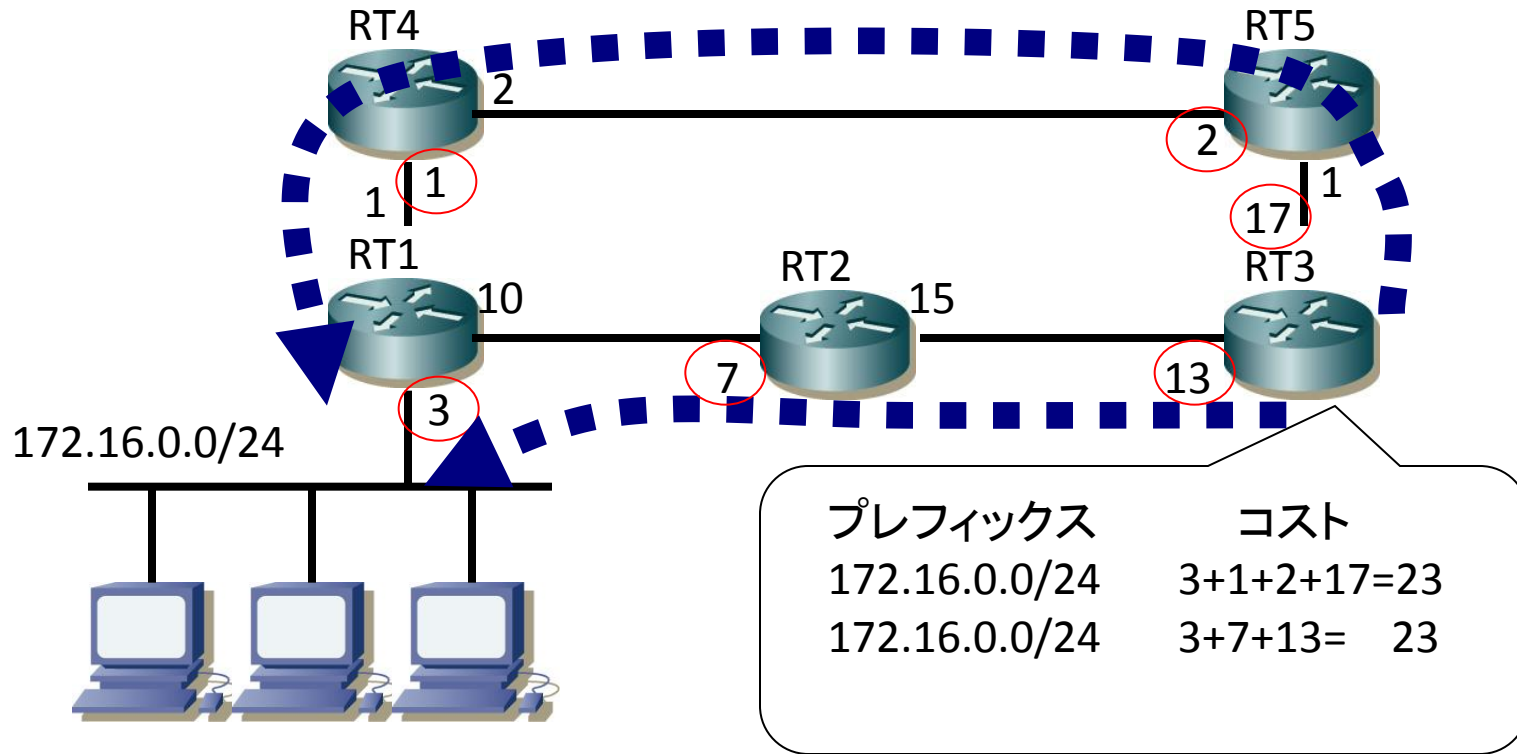
- 最も小さいコストの経路を探索するのが、SPF アルゴリズム

# 各ルータのSPF計算



- 各ルータはSPFで自身を頂点とするツリーを計算する

# 同じコストのパスを複数利用できる



- 同じコストの経路を同時に利用できる
- Equal Cost Multi Path(ECMP)と呼ばれる

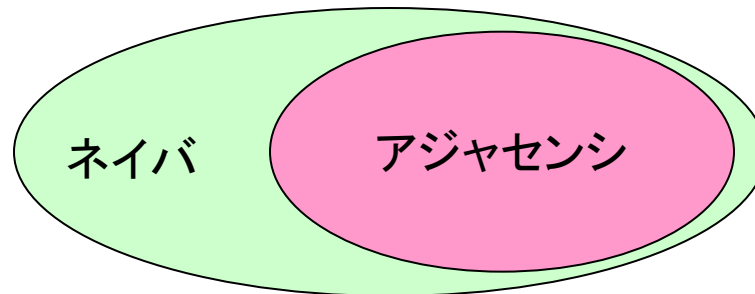
# 隣接関係

## 隣接関係の構築

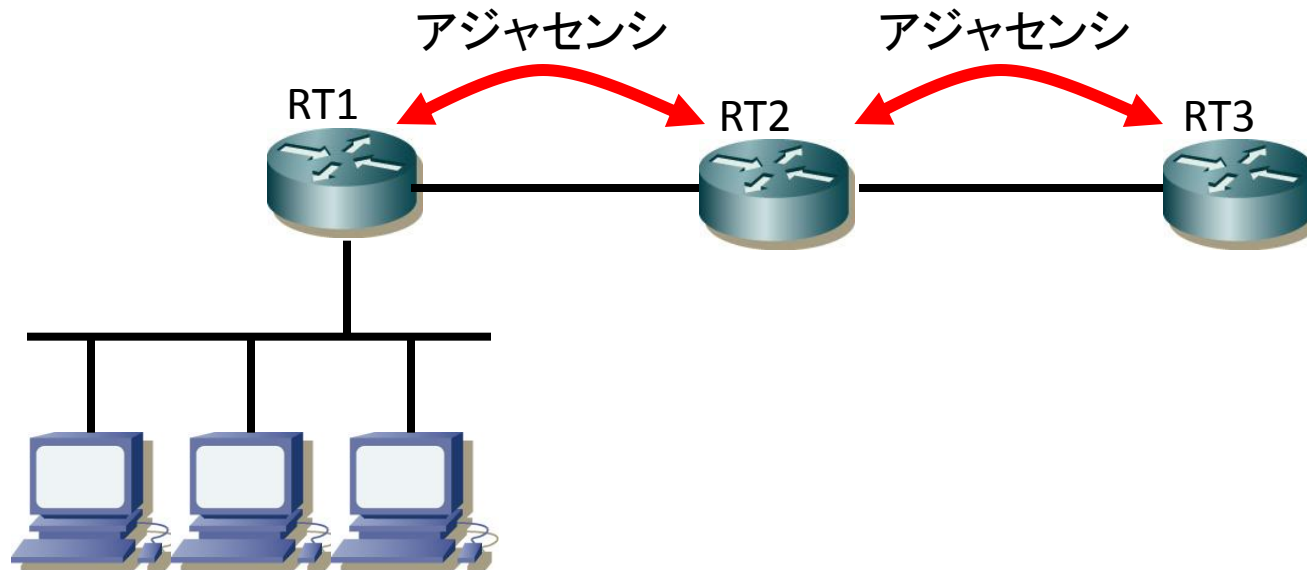


# ネイバとアジャセンシ

- ネイバ(neighbor)
  - 隣接する2台のルータで関係
  - 多くの場合、Helloで自動的に探索、維持される
- アジャセンシ(adjacency)
  - 経路情報を交換するネイバの関係
  - 全ネイバがアジャセンシになるわけではない



# アジャセンシ

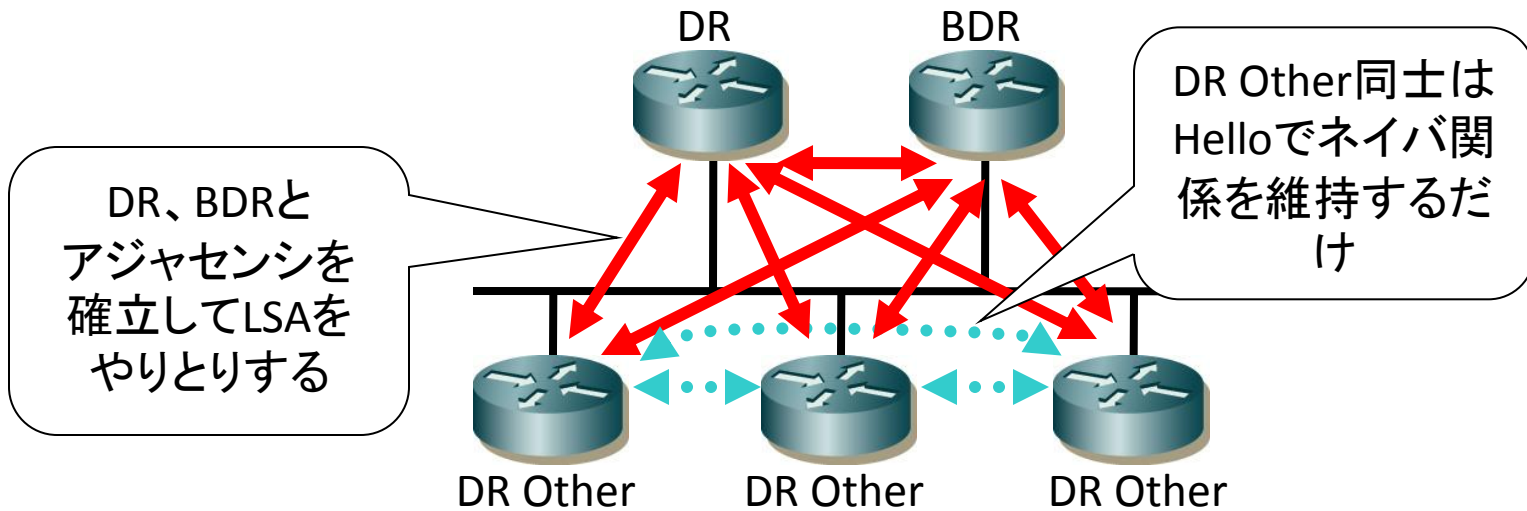


- OSPFで経路交換を行う隣接ルータのこと

# 代表ルータ (DR)

- Designated Routerのこと
  - ブロードキャストやNBMAネットワークで選ばれる
- アジャセンシ数を減らしたい
  - セグメントのルータ数が増えるとアジャセンシ数は猛烈に増加
  - アジャセンシ数が減れば負荷を軽減できる
  - 一つのセグメントでは、選出された代表ルータとだけアジャセンシを確立すればよい
    - 実際にはバックアップのBDRともアジャセンシを確立

# DR、BDRとDR Other

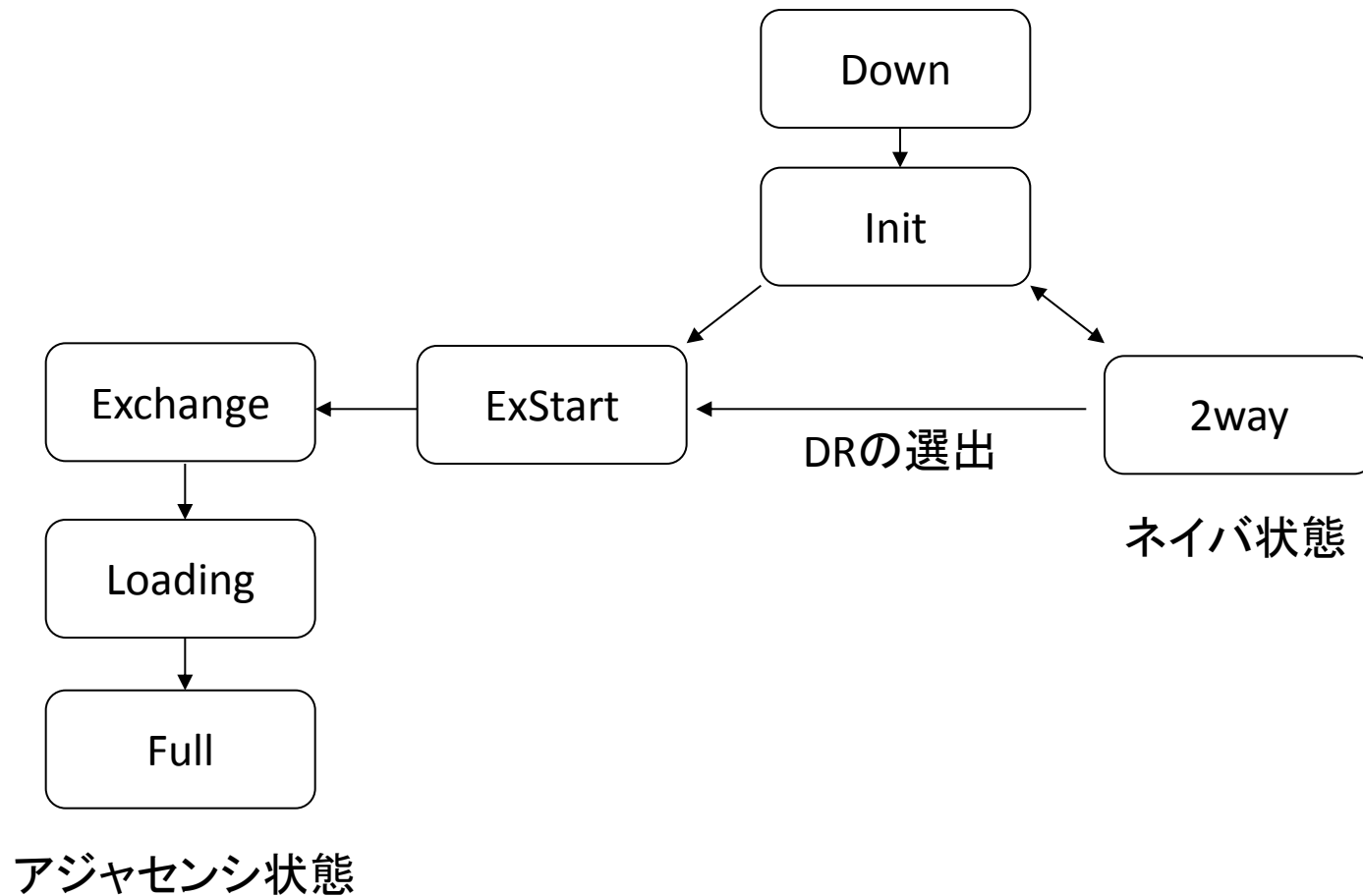


- DR Other(その他のルータ)は、DRとBDRとのみアジャセンシを確立する
- DR Other同士は、アジャセンシを確立せずにネイバ状態(2way)を維持する

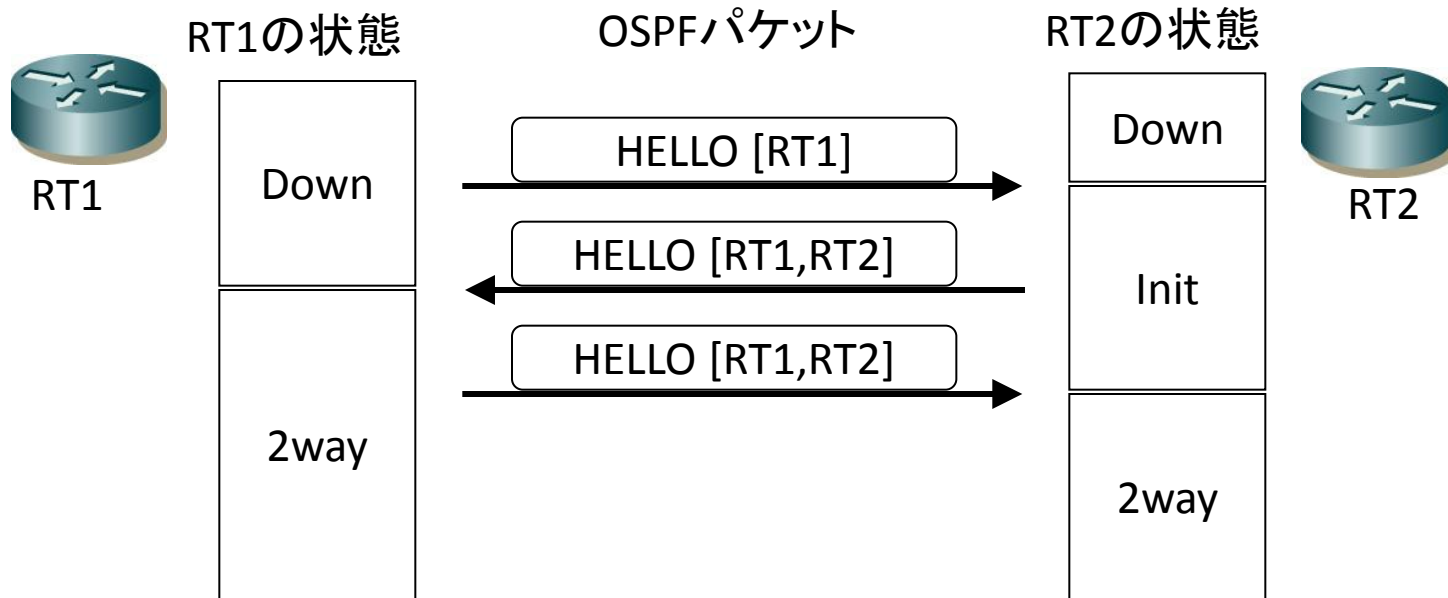
# DR、BDRの選出

- ルータ優先度の高いルータが選出される
  - ただし、既に選出済みの場合は置き換わらない
  - ルータ優先度はHelloで交換される
  - 優先度が同じ場合は、ルータIDの大きな方
- DRが故障したときはBDRがDRへと移行する
  - BDRがDRになるまで、新たなBDRは選ばれない
  - 不要な遷移をできるだけ防ぐため

# 隣接関係の状態遷移



# ネイバの確立まで



Down - Helloを受信していない状態

Init - Helloを受信した状態

2way - 受信したHelloに自ルーターIDが有る状態

# アジャセンシを確立する条件が整っていれば、2wayにならずにExStartに進む

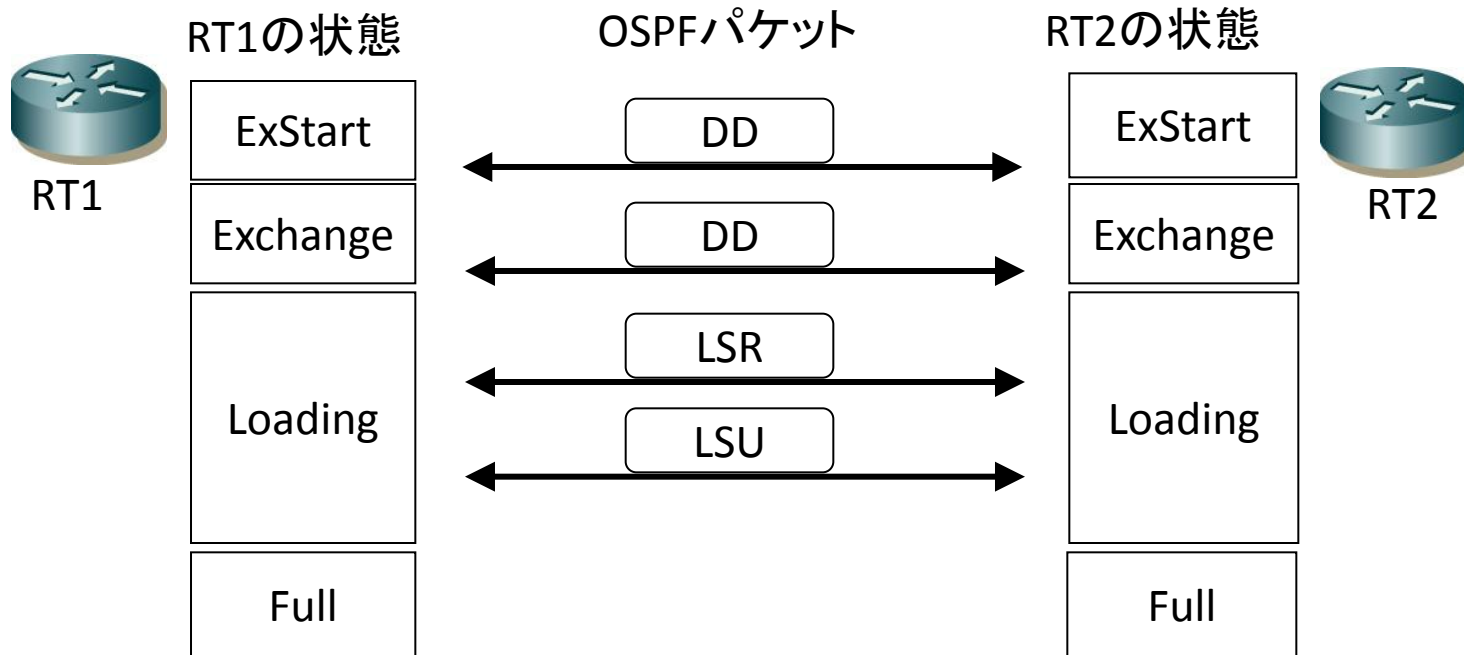
# point-to-pointリンクや、ルーターのどちらかが既にDR、BDRの場合

# ネイバ確立の条件

- Helloパケットに含まれる情報で判別
- 同一でなければならないもの
  - 認証情報
  - エリアID
  - 所属するネットワーク
    - p2pとvirtual接続を除く
  - OptionのE-bit(stubエリアかどうか)
  - hello送信間隔
  - ルータ死亡間隔秒数



# データベースの同期



- ExStart - 同期するマスターを選んでいる状態
- Exchange - お互い保持するLSDBの情報を交換している状態
- Loading - LSDBの差分を交換している状態
- Full - LSDBが同期し、アジャセンシが確立した状態

# データベースの同期条件

- Database Descriptionパケットに含まれる情報で判別
- 同一でなければならないもの
  - インタフェースのMTU
  - OptionのE-bit(stubエリアかどうか)

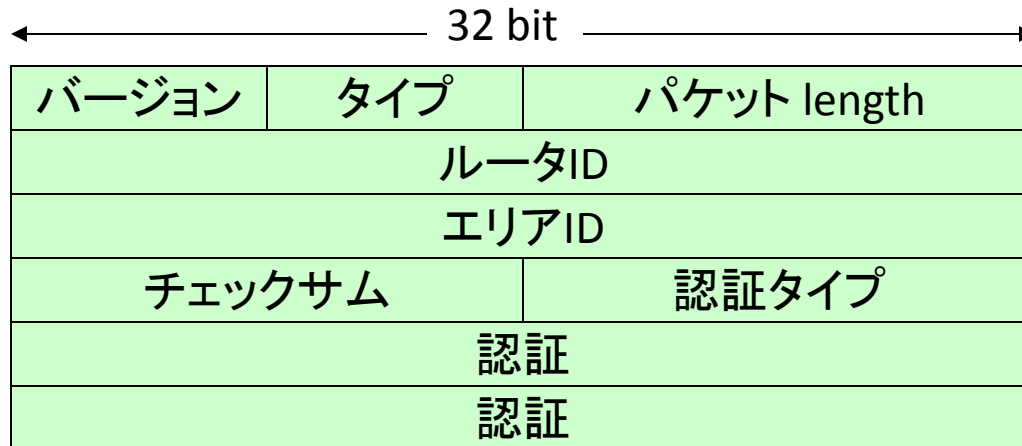
# OSPFパケット

## OSPFのパケットフォーマット

# OSPFパケットの送信先

- ALLSPFRouter[224.0.0.5]
  - 全てのOSPFルータが受信する
- ALLDRRouter[224.0.0.6]
  - DR, BDRのみが受信する
- p2p接続では [224.0.0.5]宛
- ブロードキャストネットワークで、
  - Hello及びDRとBDRからのLS update、LS Ackは[224.0.0.5]宛
  - DROtherからのLS update、LS Ackは[224.0.0.6]宛
- その他
  - ネイバへのunicast宛

# OSPF packet header



## • 24-octetの固定長

タイプ:

- 1 Hello
- 2 Database Description
- 3 Link State Request
- 4 Link State Update
- 5 Link State Acknowledgment

認証タイプ:

- 1 認証なし
- 2 シンプルパスワード認証
- 3 暗号認証

# タイプ1 - Hello

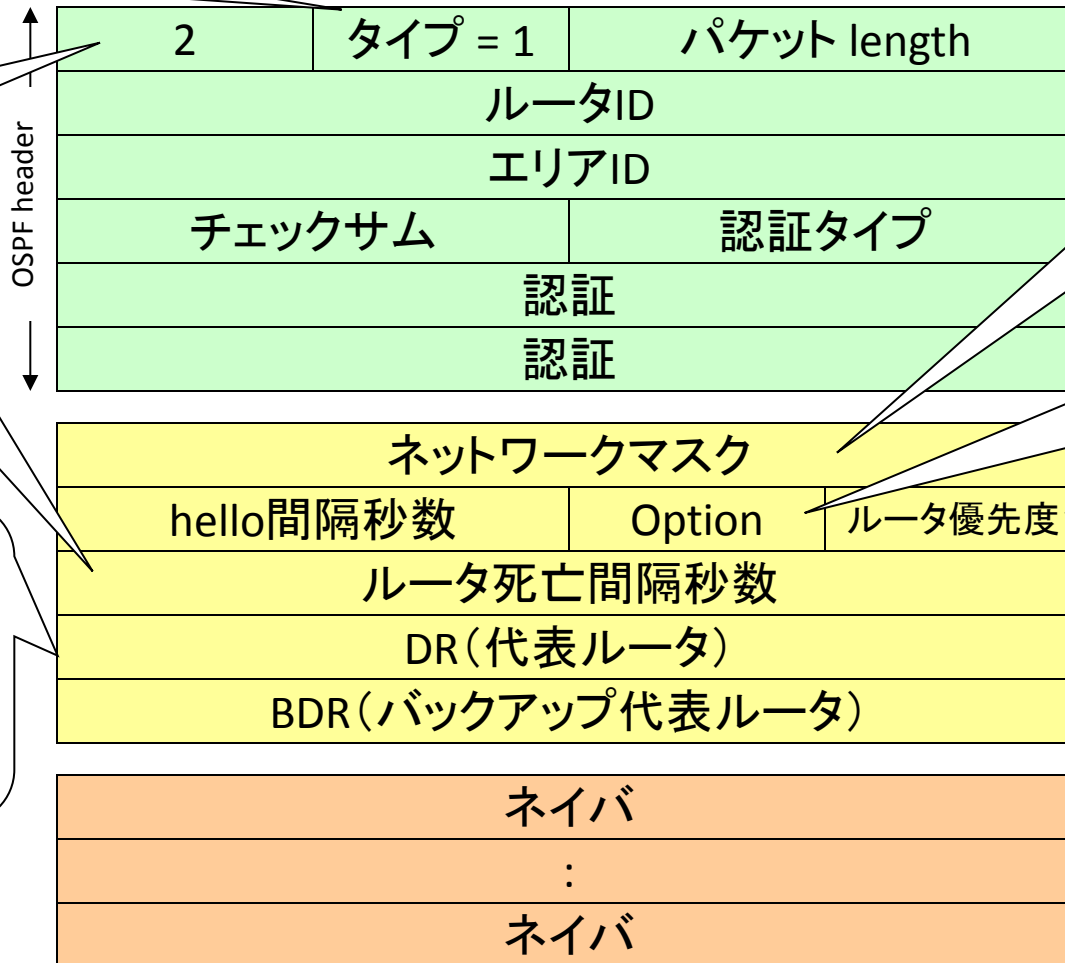
- ネイバの管理
  - ルータID、エリアIDの通知
  - ネイバの検出、維持、生死確認
- ルータ優先度の通知
  - DR、BDRの選出
- Optionフィールドでルータの機能の通知
  - E-bit: 外部経路が扱えるかどうか  
Stubエリアでは0、それ以外は1

# タイプ1 - Hello

タイプ1 =  
Hello

32 bit

OSPF v2



helloを送信するネットワークのネットマスク

応答が無い場合に死亡したと判断するまでの秒数

既に知っていれば、DR,BDRのIPアドレス。知らなければ、0.0.0.0

E-bit:  
外部経路を扱える

DR,BDRの選出に利用

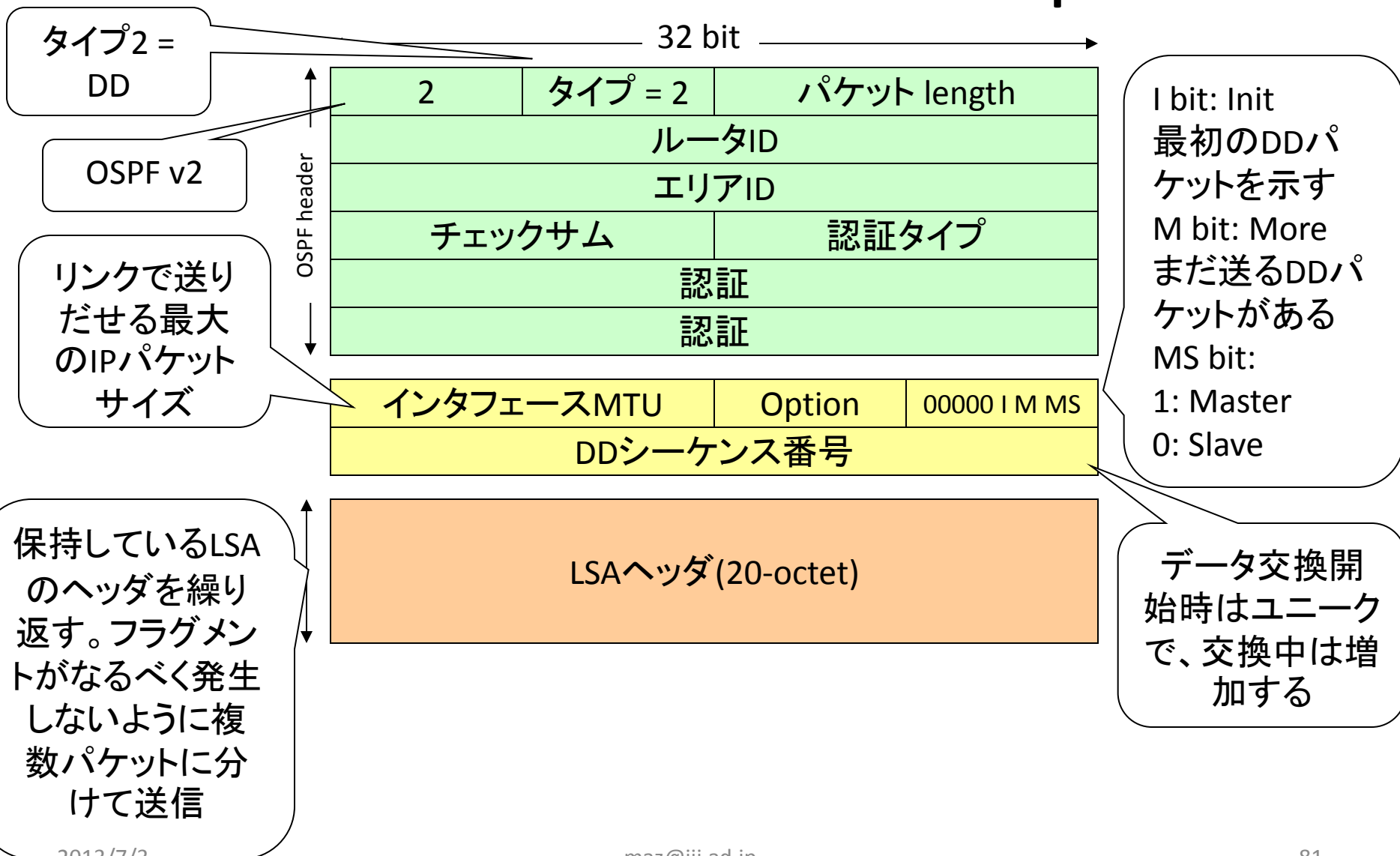
現時点で有効なネイバのルータIDをリストする

# タイプ2 - Database Description

- アジャセンシ確立時に、保持するLSAを通知
  - LSDBの同期をとる
  - 全LSAのヘッダのみを伝える
  - マスタとスレーブになって情報を交換
    - ルータIDの大きい方がマスタ
    - スレーブはマスタのDDシーケンス番号に同期する
- インタフェースのMTUを伝える
- Optionフィールドでルータの機能の通知
  - E-bit: 外部経路が扱えるかどうか(Stubエリアでは0)



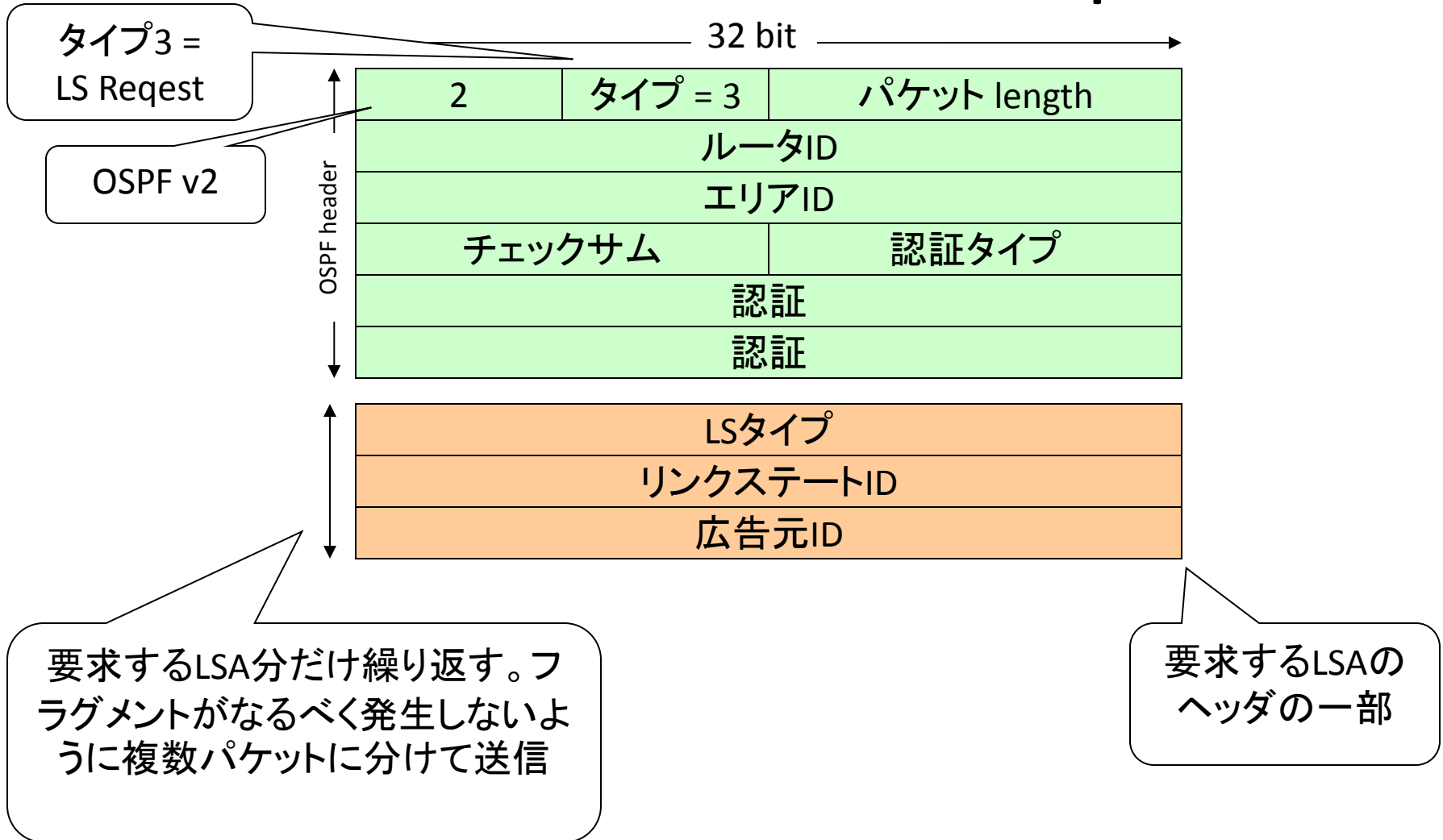
# タイプ2 - Database Description



# タイプ3 - Link State Request

- DDパケットでLSA情報を交換後、差分を埋めるためにLSAを要求する
  - 最新のLSAを要求
  - 保持していないLSAを要求
- LSAが識別できる情報をリストして送信する
  - LSタイプ、リンクステートID、広告元ID

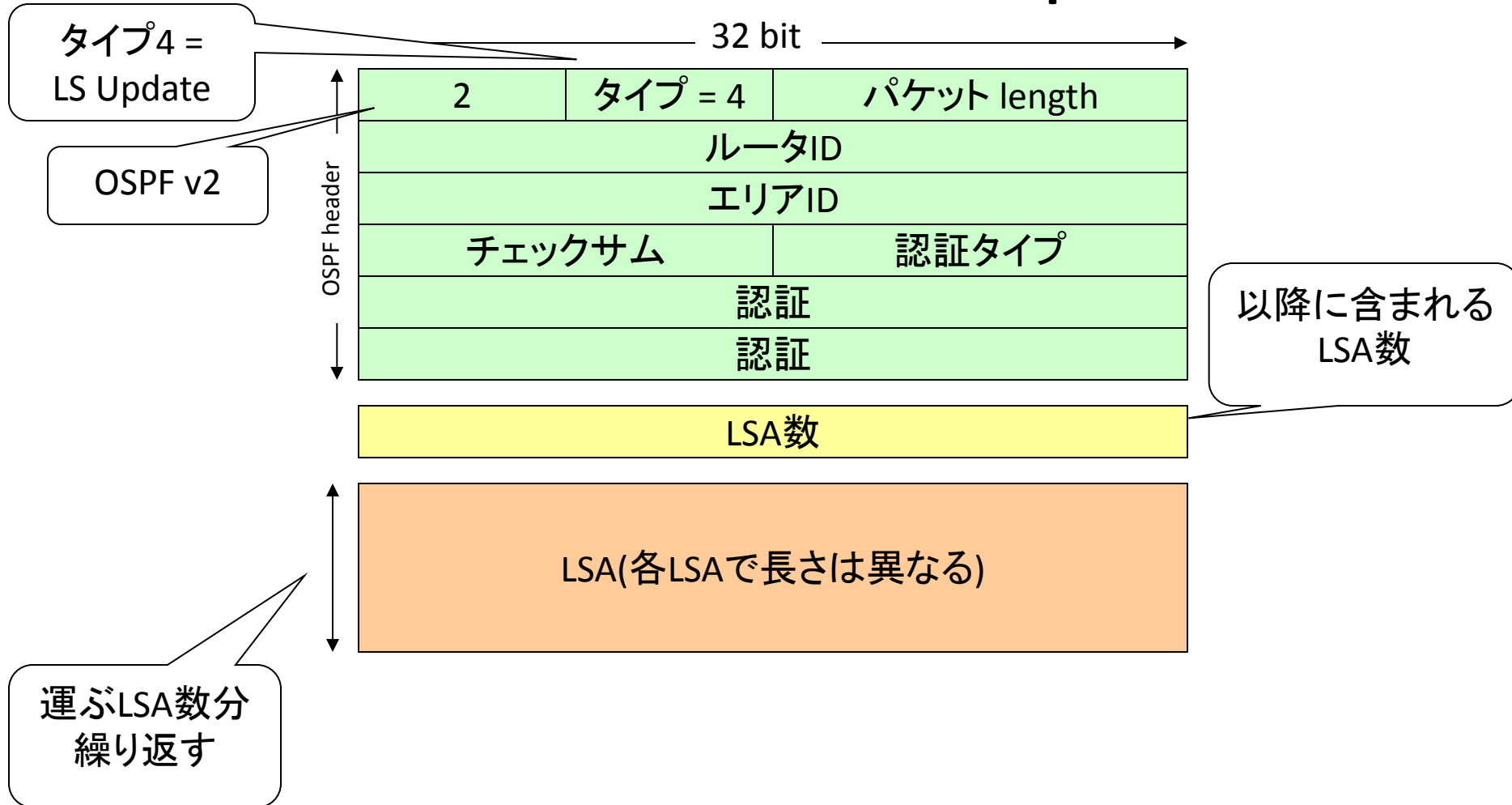
# タイプ3 - Link State Request



# タイプ4 - Link State Update

- 一つ以上のLSAを運ぶ
- 隣接のOSPFルータまで伝播する
  - LSAの転送はHop by Hop
- 受信確認は状態によって異なる
  - Loading中の受信確認は無く、Link State Requestで必要なものが再要求される
  - アジャセンシ確立(Full)後はアジャセンシからLink State Acknowledgmentが返信される

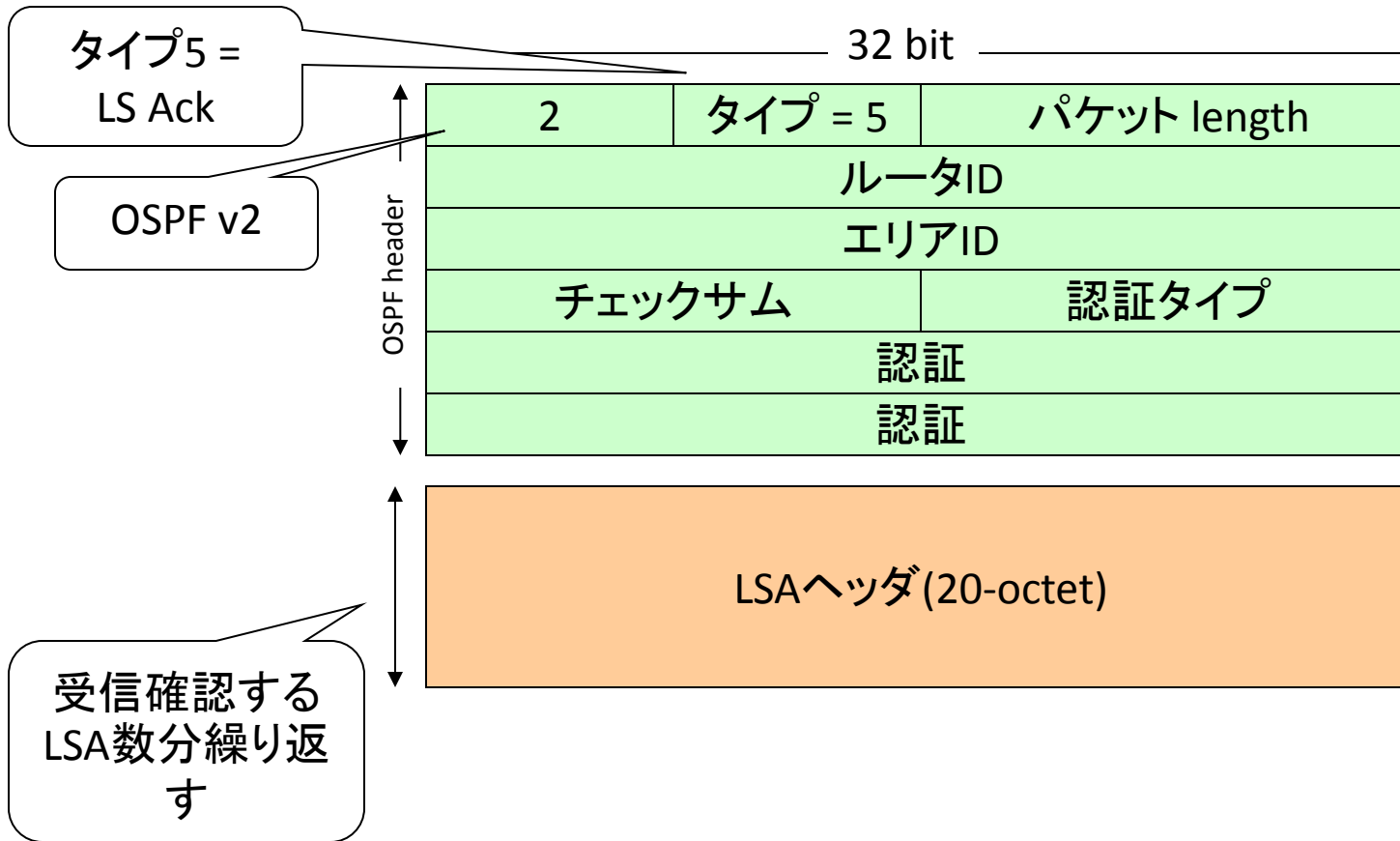
# タイプ4 - Link State Update



# タイプ5 - Link State Acknowledgment

- 受信確認を通知する
  - LSAのヘッダを通知する
  - これで確実にLSAが伝わったことを保証する
- 一つ以上の受信確認を運ぶ
  - アジャセンシ確立(Full)後に利用

# タイプ5 - Link State Acknowledgment



# LSAの伝播と管理

LSAの管理を解説する



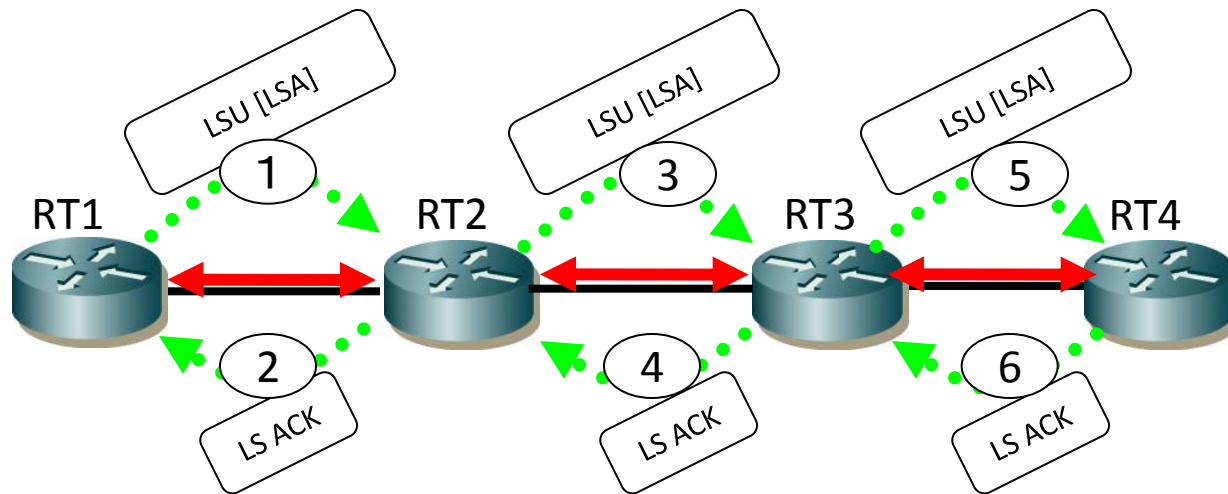
# OSPFの再計算

- トポロジの変化等があれば、新規のLSA生成
  - 接続断や新規ネットワークの接続

- 検知
  - ルータがネットワークの変更を検知
- 通知
  - 新規LSAを生成して伝播
  - LSDBの更新
- 再構成
  - 各ルータでSPFを計算して経路情報を更新

# LSAの伝播

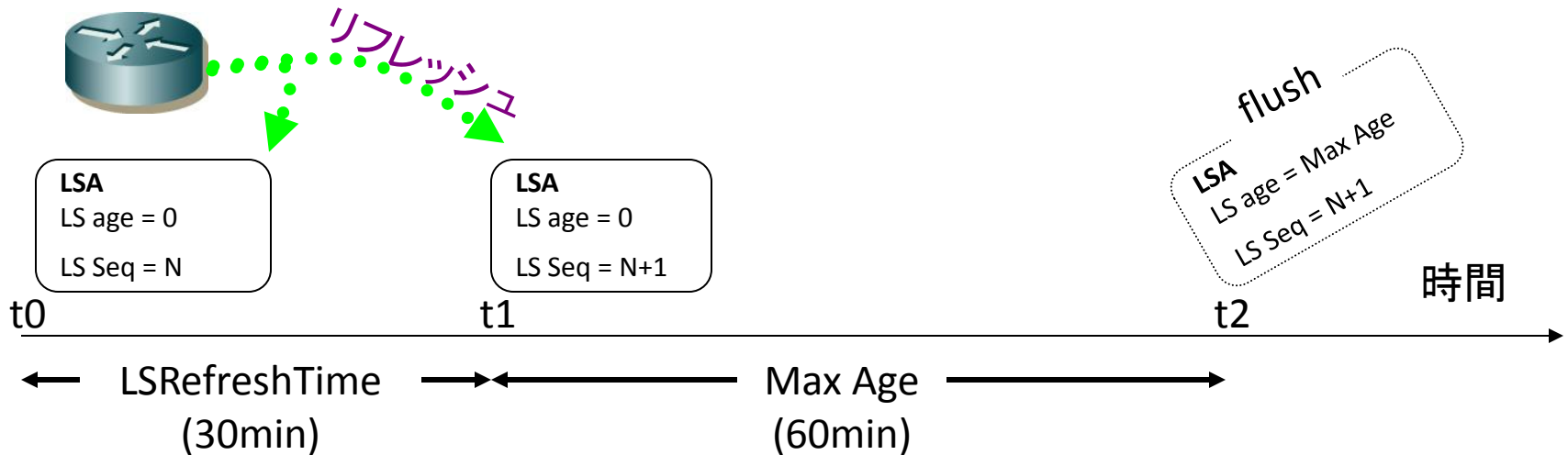
- LSAはLSUパケットでアジャセンシに広報される
  - LSUパケットは複数のLSAを運べる
- 受信したLSAには受信証明としてLS ACKを返信
  - 伝播したことを明確にするため



# LSAのaging

- ルータは保持する全てのLSAに対して、生成されてからの経過秒数を管理する
- Max Age(60分)に達するとLSAが消される
  - LSAは生成元が定期的に取りフレッシュする
  - 幽霊LSAを無くせる

# LSAのリフレッシュ

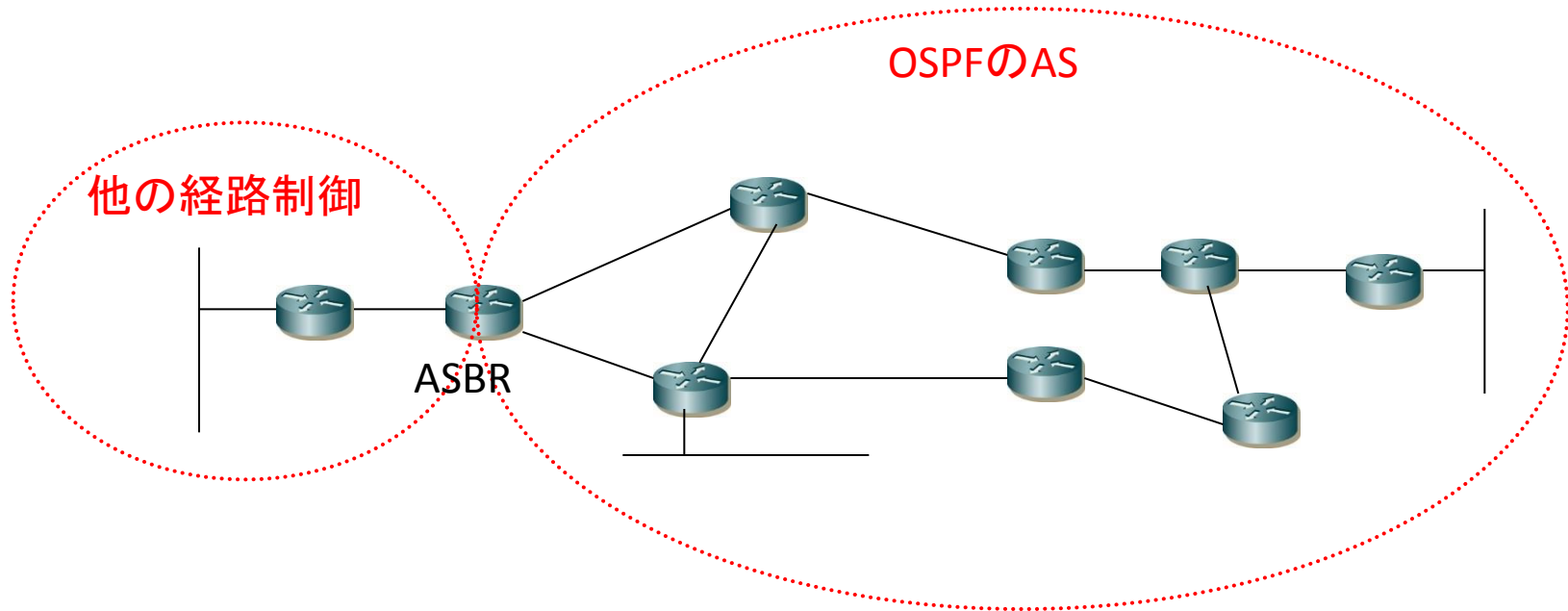


- LSAの生成元はリフレッシュ時間(30分)がくると新規にLSAを生成して広報
  - 変化が無くても生成される
- リフレッシュされないLSAはMaxAge後に利用されなくなる

# OSPFと外部経路

外部経路の扱いを解説する

# OSPFの中と外

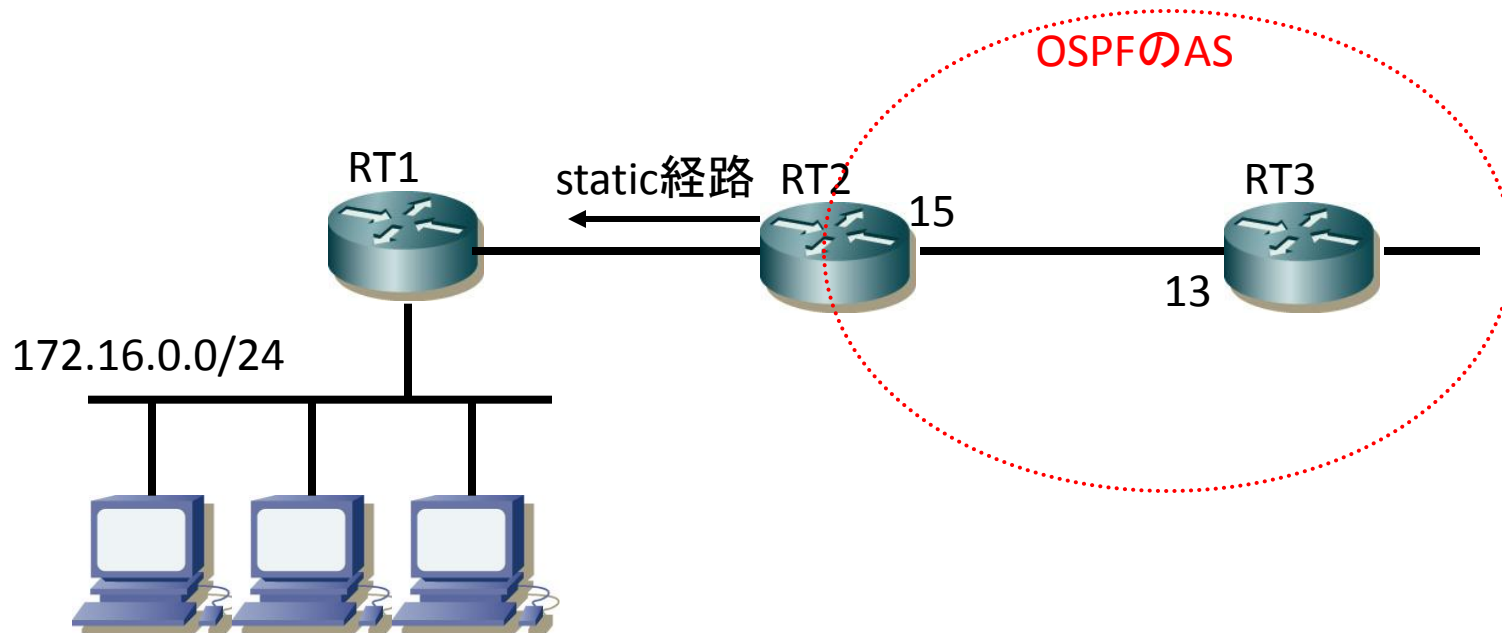


- OSPFで経路制御しているルータのグループがAS
- 他の経路制御との接点になっているルータがASBR

# OSPFのASとASBR

- AS - Autonomous System
  - 共通のプロトコル(OSPF等)で経路情報を交換するルーターのグループ
  - BGPなどでいうASとは概念が異なる
    - インターネットでのASはBGPのASを指す場合が多い
  - ここでは誤解を避けるためOSPFのASと明記する
- ASBR – AS boundary router
  - AS境界ルータ
  - 外部の経路(static等)をAS内に広報するルータ

# 外部経路(external route)



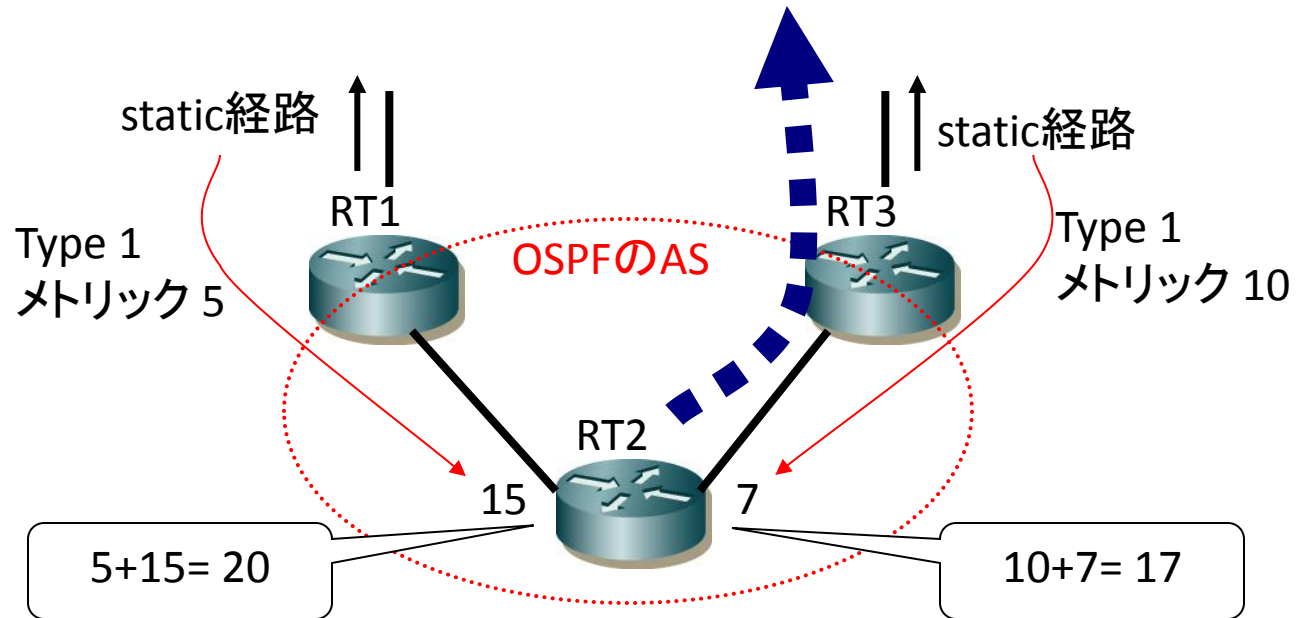
- RT2はstatic経路をOSPFのAS内に広報できる
- 外部経路を広報する際にメトリックを付加できる



# 外部経路メトリック

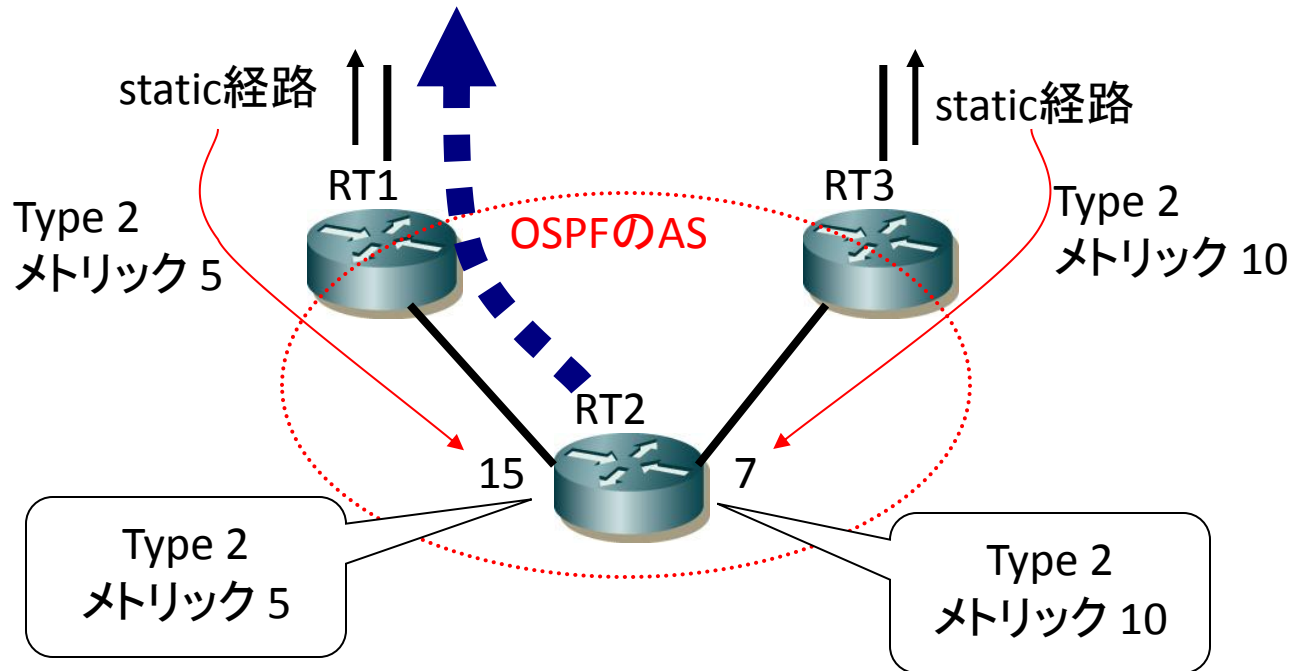
- Type 1 - リンクコストと同様に加算される
  - 同じ宛先のType 1外部経路があった場合、途中リンクのコストも加算して、もっとも小さなコストの経路が選ばれる
- Type 2 - とにかく小さな値が選ばれる
  - 同じ宛先のType 2外部経路があった場合、もっとも小さなType 2メトリックの経路が選ばれる
  - 同じType 2メトリックの場合、転送先アドレスまでのコストがもっとも小さな経路が選ばれる
- 同じ宛先のType 1とType 2の外部経路があった場合、Type 1の経路が選ばれる

# Type 1 外部経路



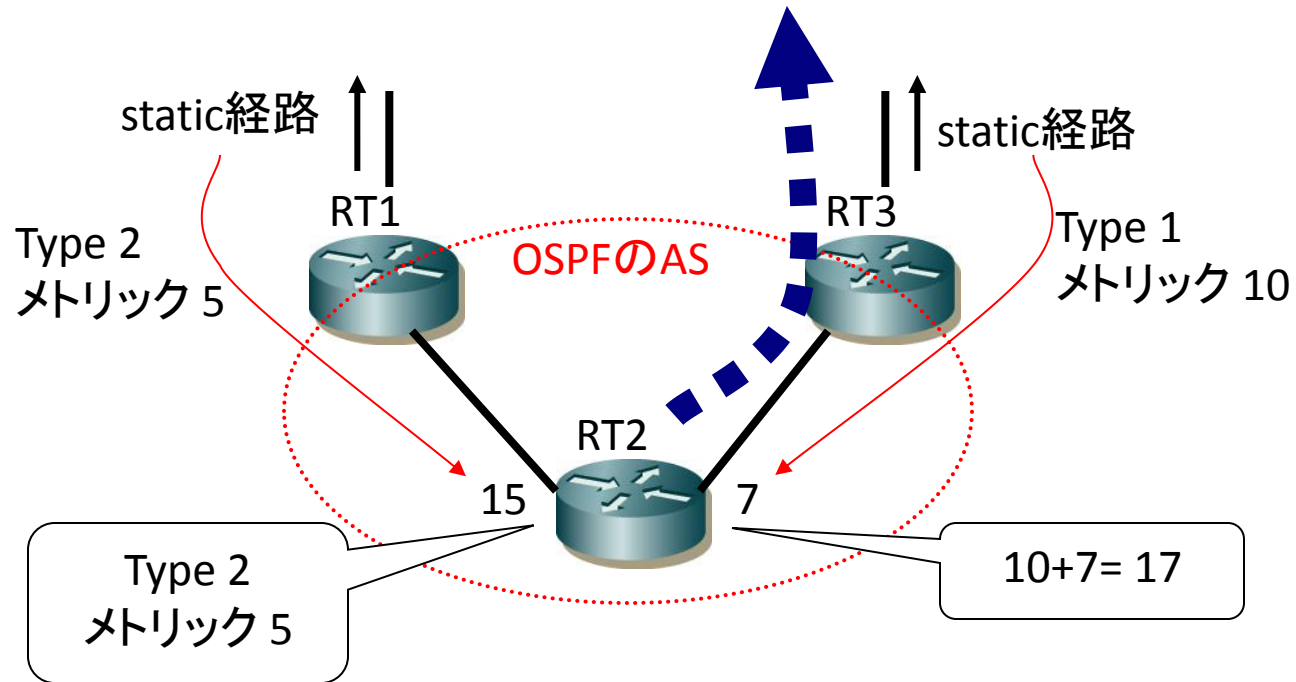
- リンクコストの加算の結果、RT3が広報する外部経路が優先される

# Type 2 外部経路



- 小さなType 2メトリックを持つRT1からの外部経路が優先される

# 外部経路の混在

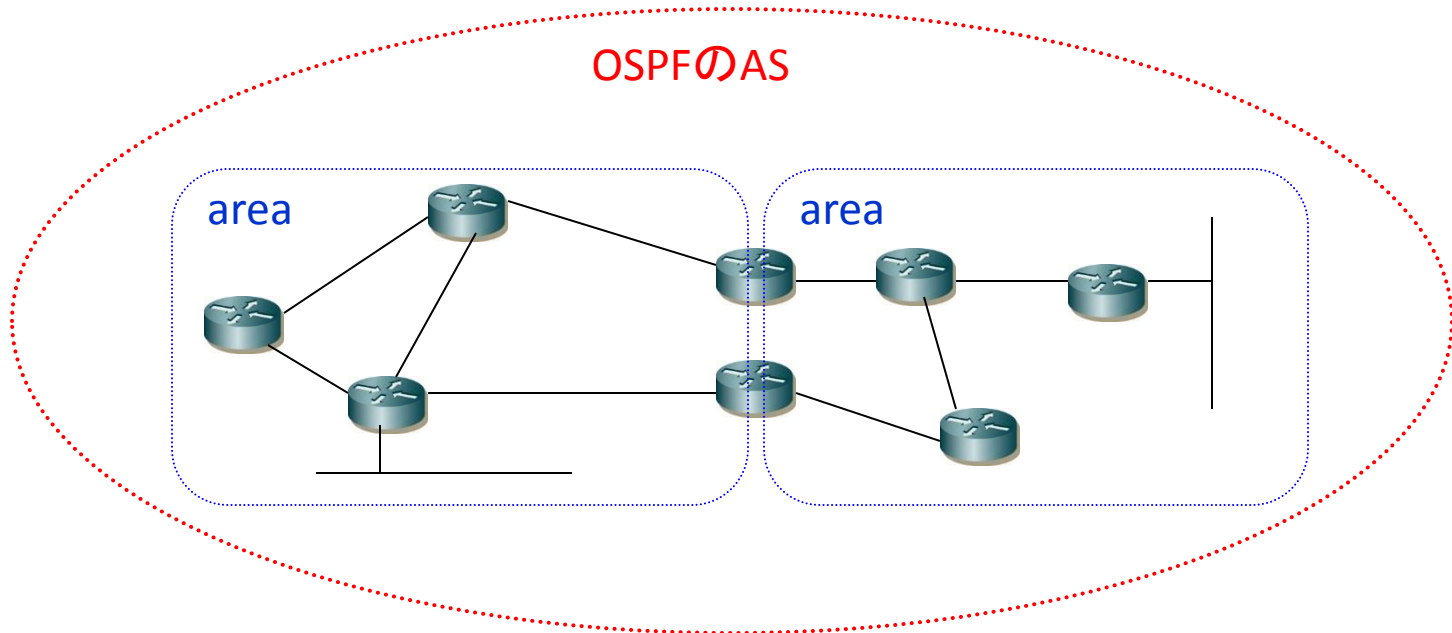


- 常にType 1経路が優先される
- BGPのネクストホップになるアドレスを運ぶならType 1経路
  - closest exitが維持できて便利

# OSPFエリア

OSPFのエリアについて解説する

# エリア(area)



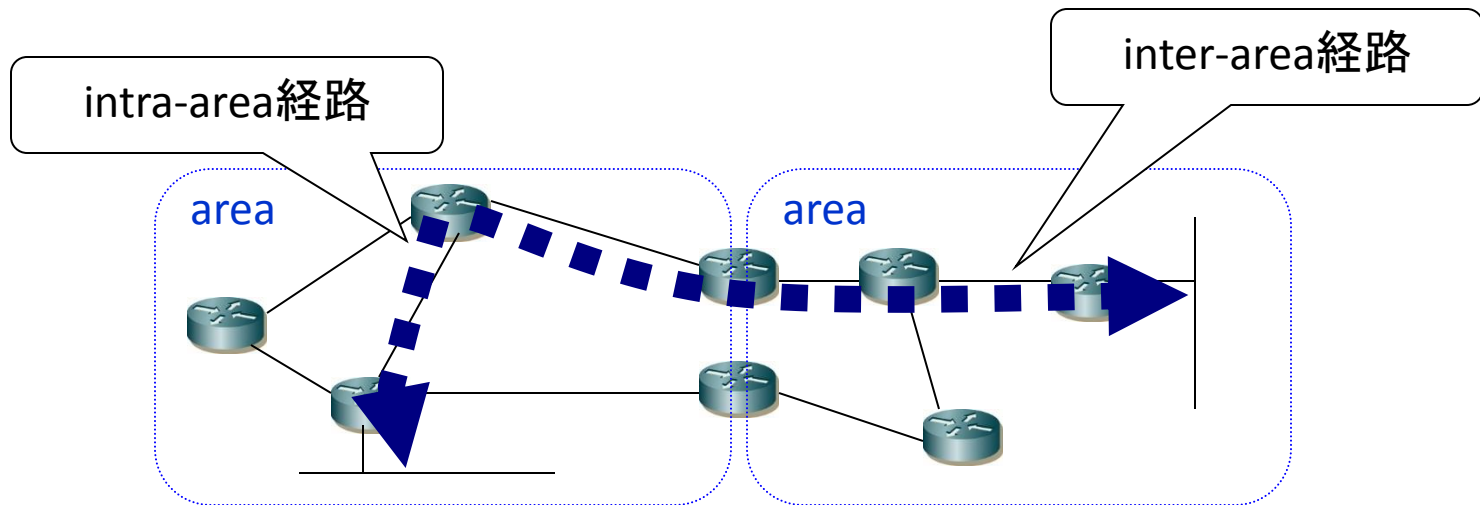
- OSPFでは連続したネットワークのグループを作成できる。これに接続するルータを含めて、エリアと呼ぶ。

# エリアの概要

- ルータがエリアの境界になる
- それぞれのエリアで独立にLSDBが管理され、経路情報が計算される
- あるエリアのトポロジは、他のエリアからは見えない
  - 必要な経路情報のみが伝播する
  - 計算負荷の軽減

# エリア間とエリア内

- 宛先が同じエリアか違うエリアか
  - エリア内経路(intra-area経路)
  - エリア間経路(inter-area経路)
- 同じ宛先については、エリア内経路が優先



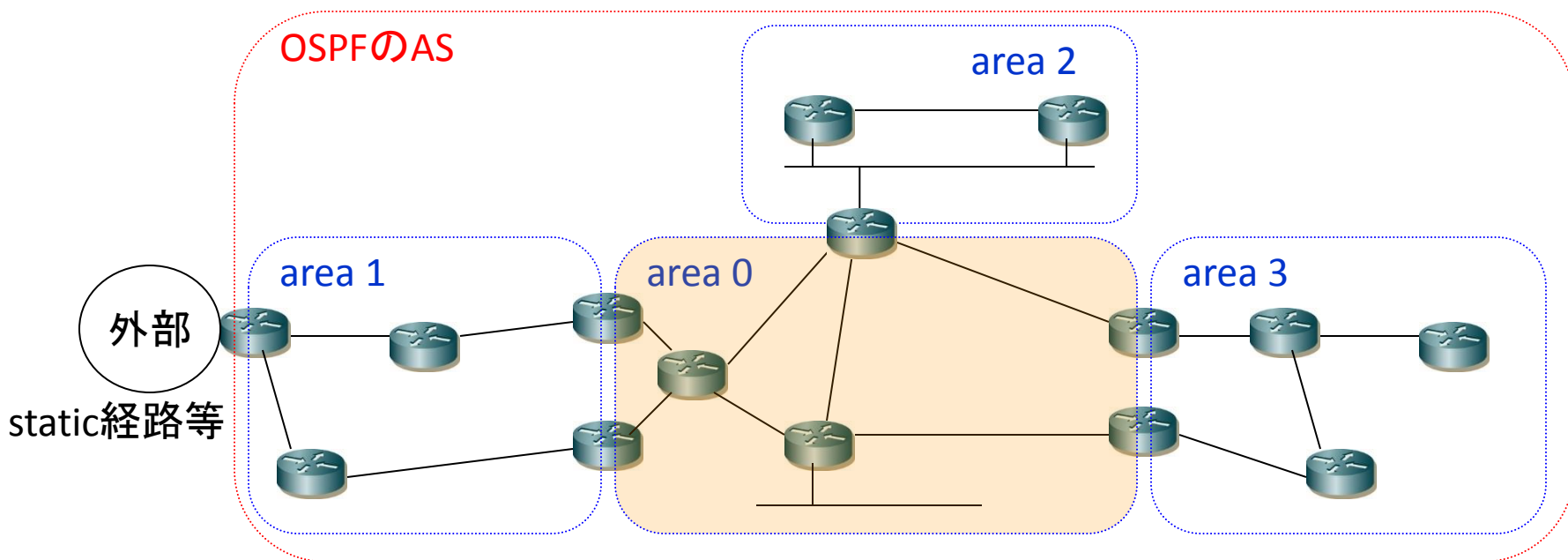


# エリアID(area ID)

- 各エリアを識別する32bitの数値
  - 各エリアに管理者がIDを設定する
- そのまま数字で表記する書式
  - area 0
- IPアドレスの様に8bit毎に区切った書式
  - area 0.0.0.0

# バックボーン エリア(area 0)

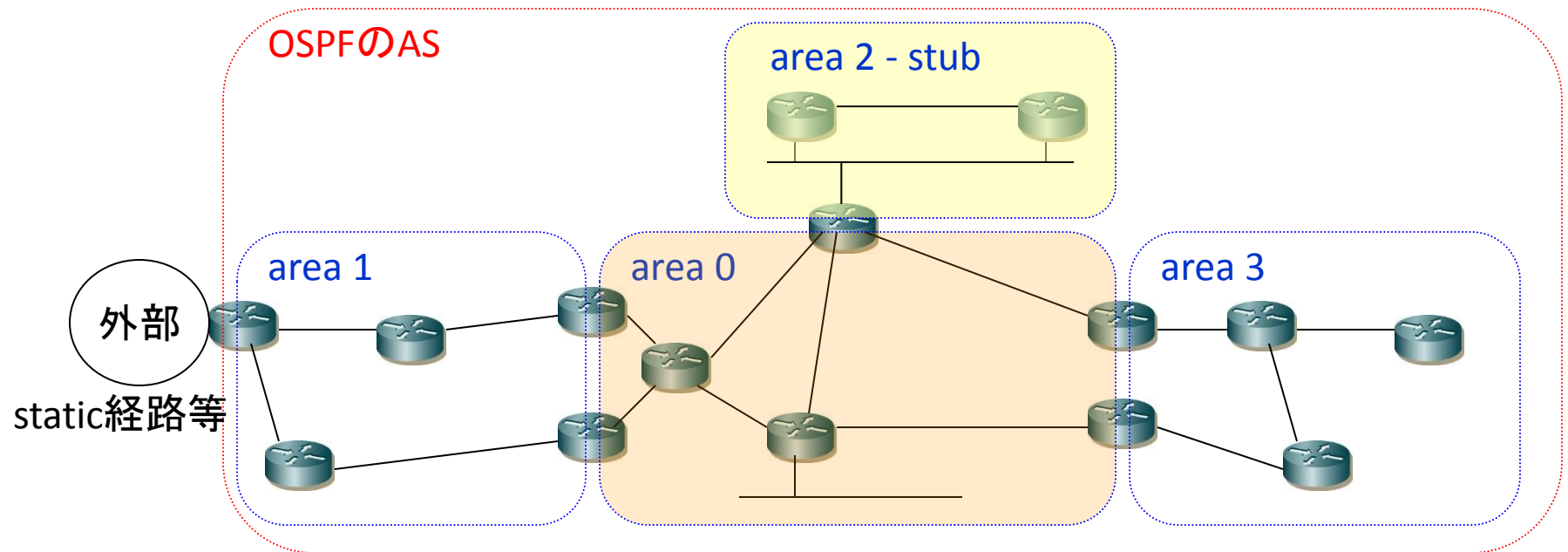
- エリアIDが0
- 各エリアの経路情報を交換できる特別なエリア
- OSPFのマルチエリア構成は、バックボーンエリアを中心としたスター型



# stubエリア

- 外部経路が増大した構成での対策を考えた
  - 外部経路は基本的に全エリアに広報される
  - これを軽減する仕組みがstubエリア
- 外部経路を伝播しない代わりに、default経路をエリア境界ルータが広報する
  - ASBRが無いエリアに適用できる
  - エリア内の全てのルータでstubエリアと設定する

# area 2をstubエリアにしている例



- area 2にはエリア間経路(inter area経路)とdefault経路のみが広報される
- この場合、area 3もstubエリアにすることができる

# LSA

LSA(link state advertisement)の  
パケットフォーマットを解説する

# リンクステート広告 (LSA)

- link state advertisement
- 各ルータが広告する情報のこと
- LSAの集合がLSDBになる
- 目的に応じた幾つかのタイプがある

# LSAの種類(基本)

## LSタイプ

### 1. ルータLSA

ルータに接続するリンク、ネットワーク情報を運ぶ  
リンク種別に応じてp2p, transit, stub, virtualの4種類

### 2. ネットワークLSA

ネットワークに接続するルータ情報を運ぶ

### 3. サマリLSA(ネットワーク)

エリア外のネットワークへの経路を運ぶ

### 4. サマリLSA(ASBR)

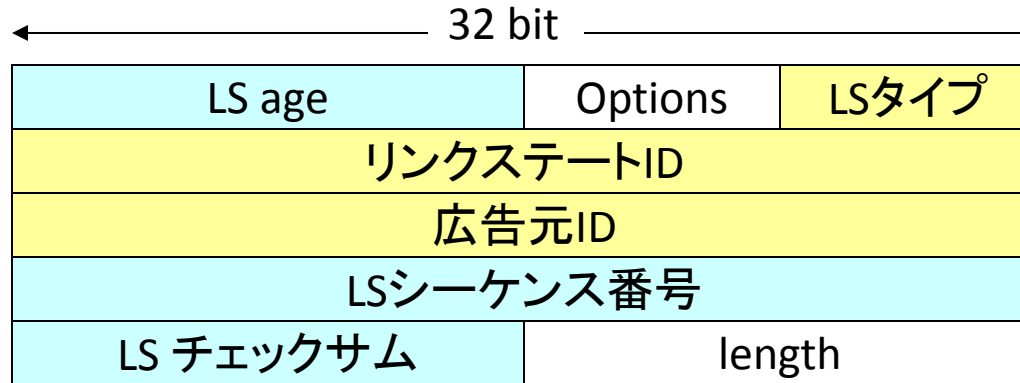
エリア外のASBRへの経路を運ぶ

### 5. AS-external-LSA

外部経路を運ぶ

Type 1とType 2のメトリックタイプが存在する

# LSA header

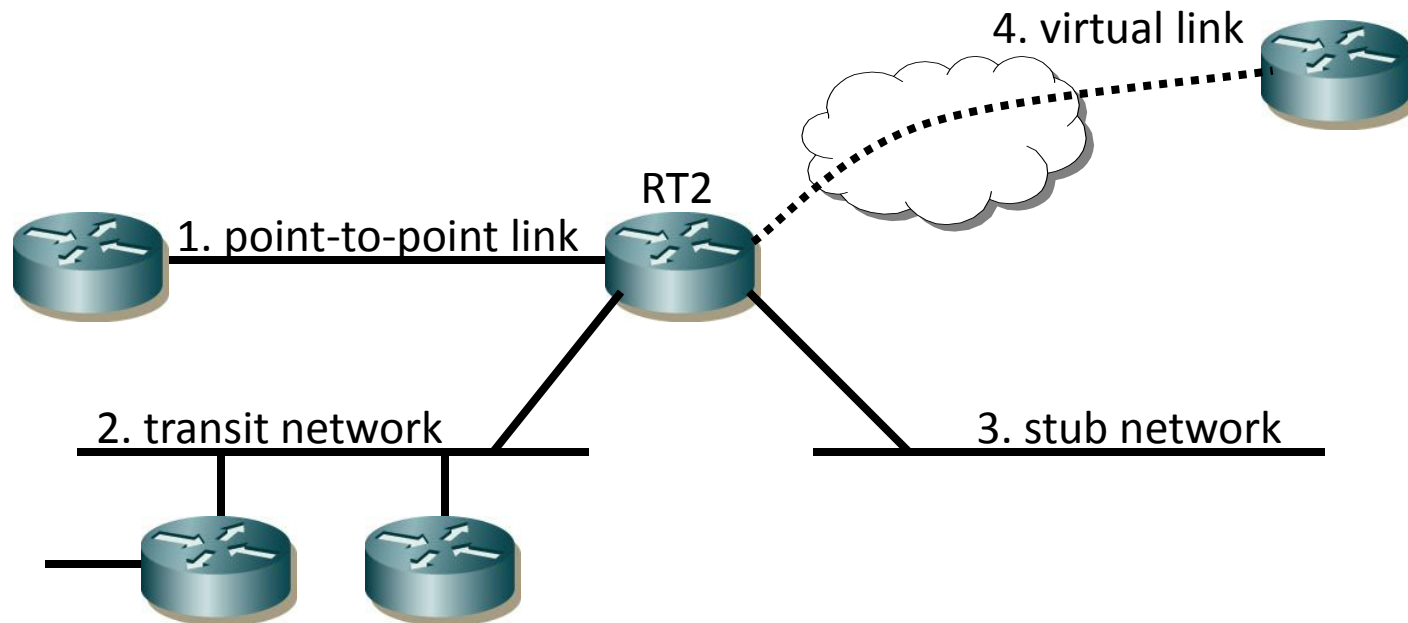


- 20-octetの固定長
- LSAを個別に識別できる
  - LSタイプ, リンクステートID, 広告元ID
- 新しいLSAを識別できる
  - LS age, LSシーケンス番号, LSチェックサム

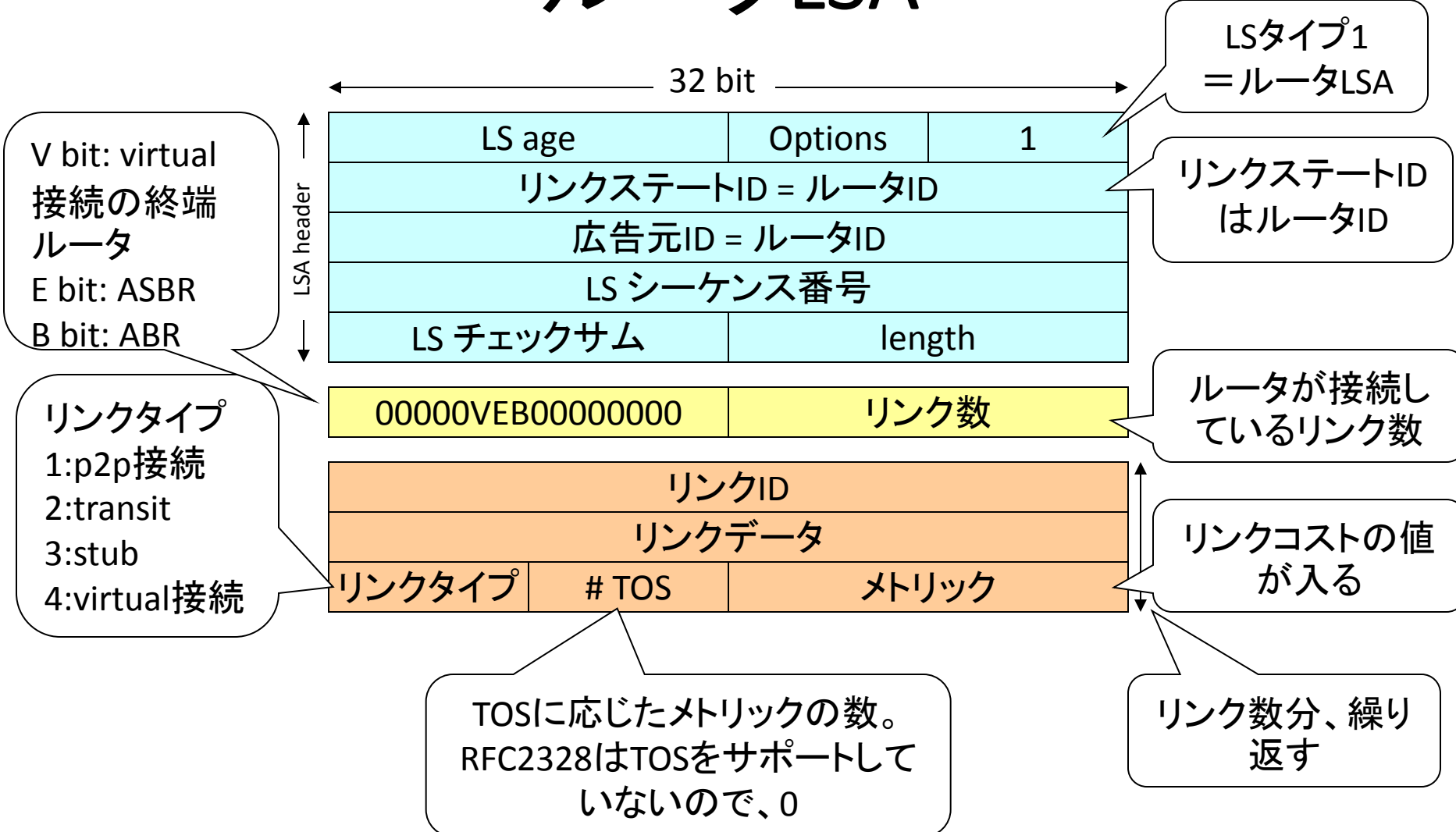


# LSタイプ1 - ルータLSA

- 全ルータが一つずつ広告する自己紹介
- 4種類のリンクタイプ



# ルータLSA



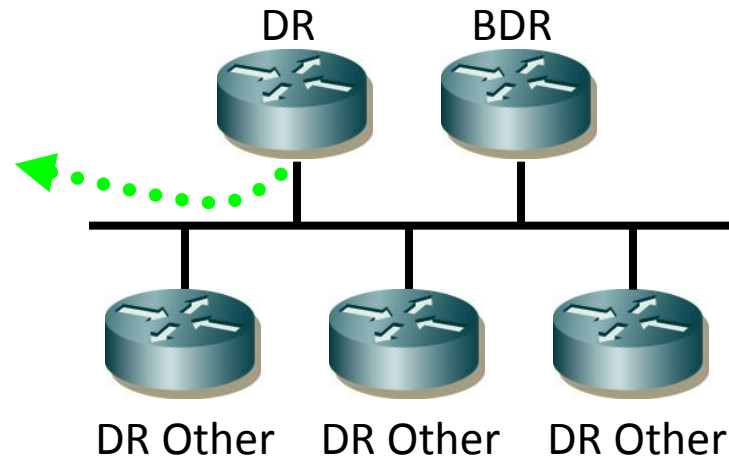
# ルータLSAで運ぶ情報

- stubネットワークのみが経路情報を運び、その他はトポロジ情報を運ぶ

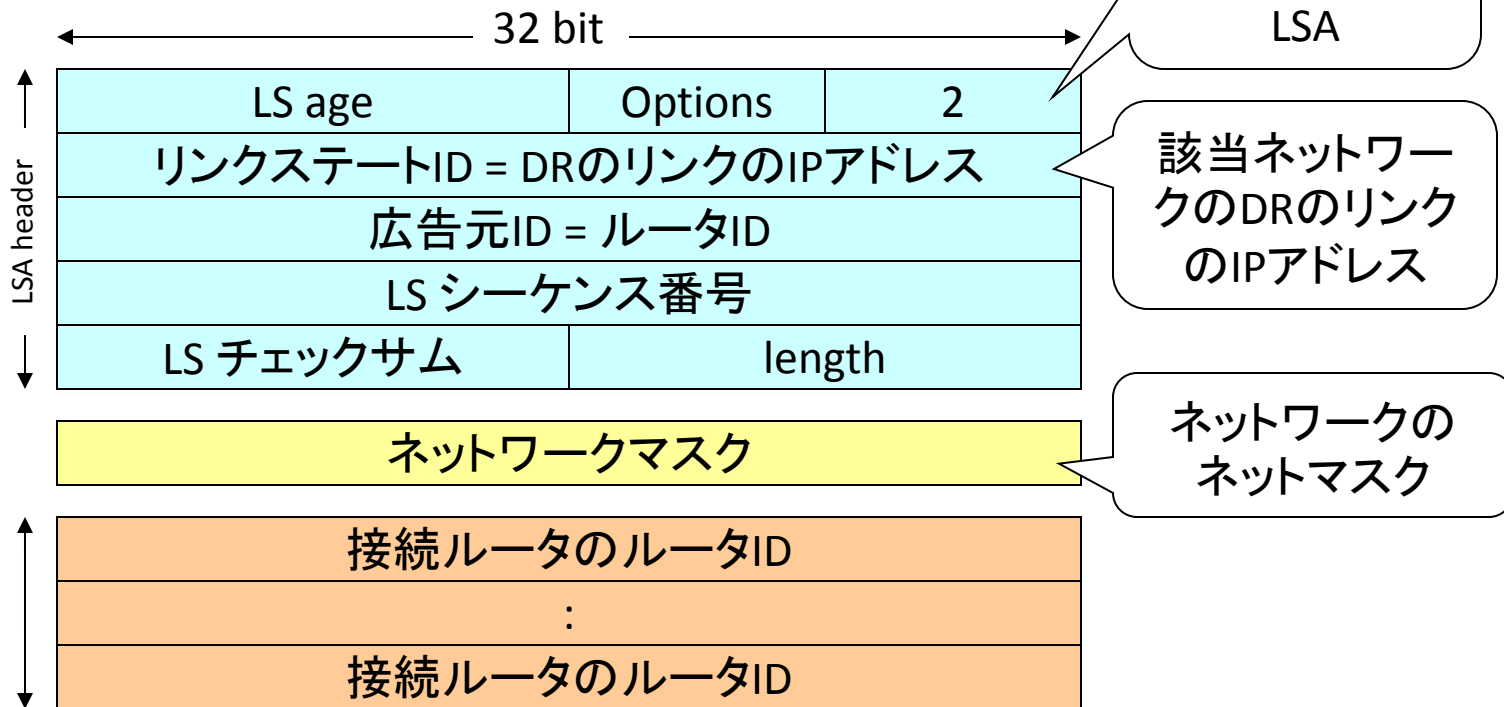
リンクタイプ	リンクID	リンクデータ	データ種類
1 p2p	ネイバのルータID	MIB-II ifindex値 (あれば、リンクのIPアドレス)	トポロジ
2 transit	DRのリンクのIPアドレス	リンクのIPアドレス	トポロジ
3 stub	リンクのIPネットワーク	ネットワークマスク	経路
4 virtual	ネイバのルータID	リンクのIPアドレス	トポロジ

# LSタイプ2 - ネットワークLSA

- transitネットワークに接続するルータのリスト
- ネットワークにつき1台の代表ルータ(DR)のみが広告する



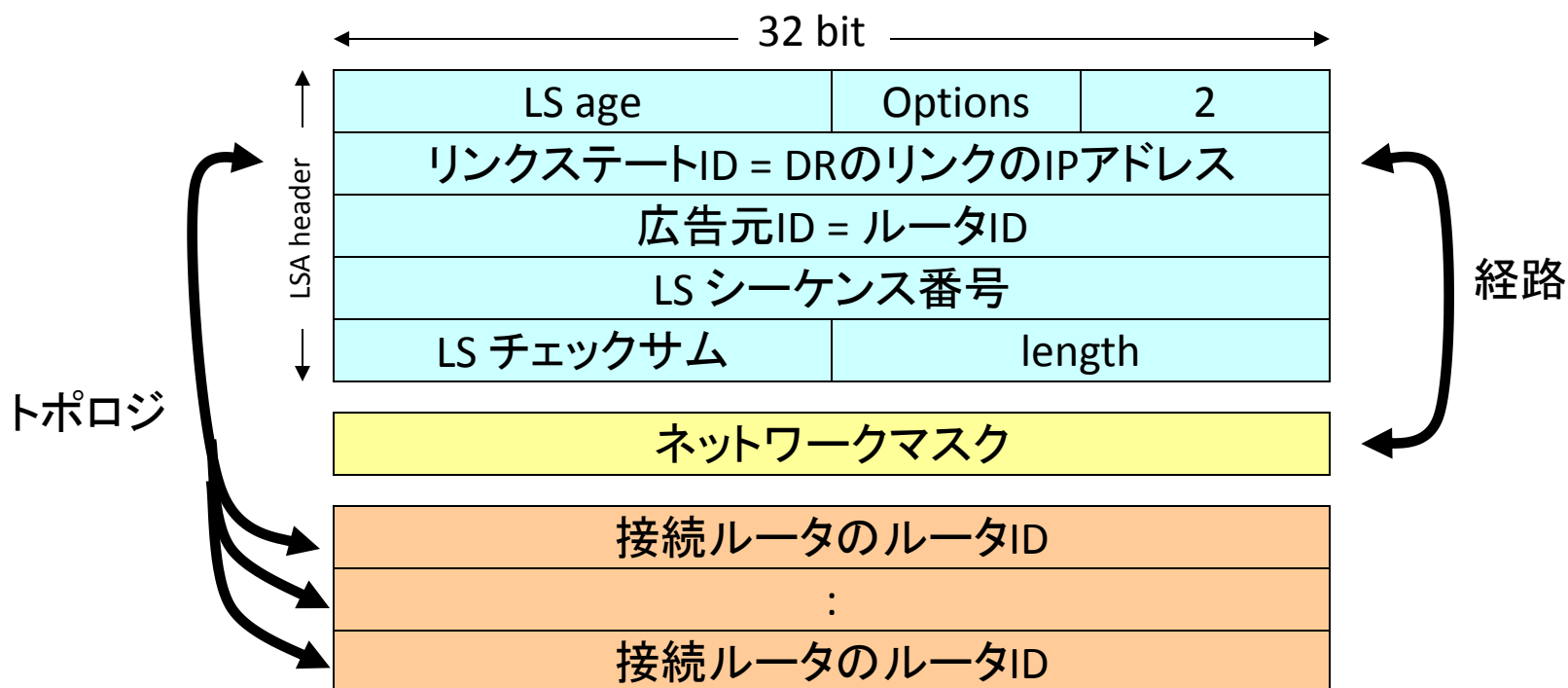
# ネットワークLSA



接続しているルータ数分、  
繰り返す

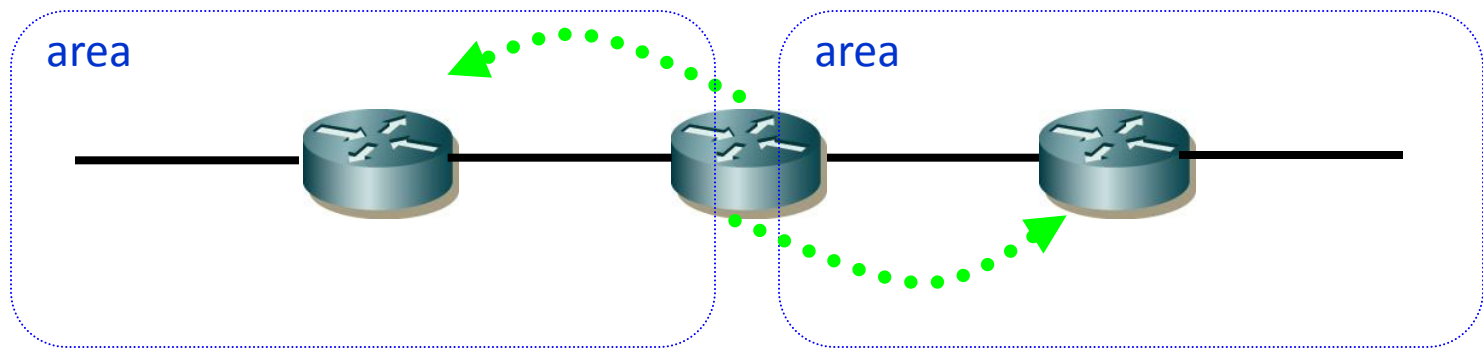
# ネットワークLSAで運ぶ情報

- トポロジと経路を同時に運ぶ

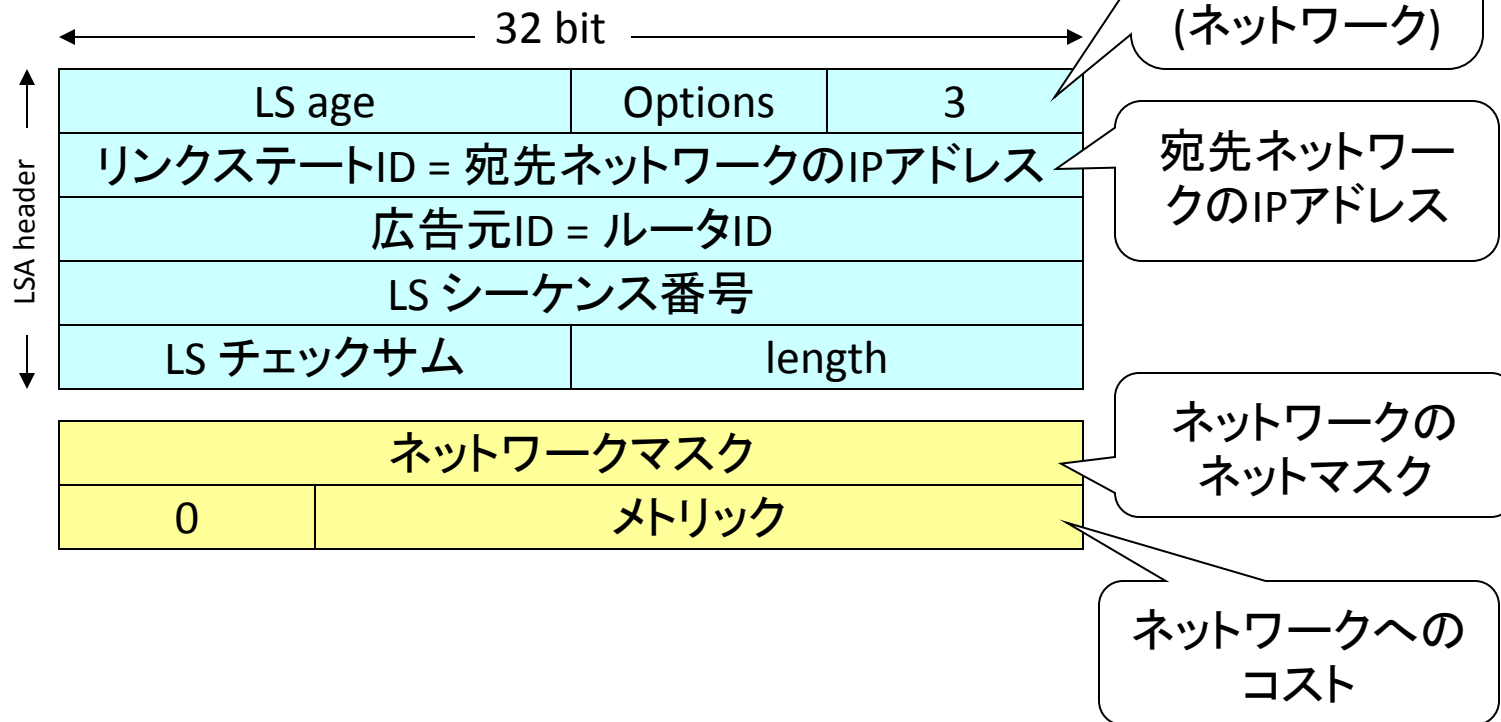


# LSタイプ3 - サマリLSA

- エリア外のネットワークへの経路情報を運ぶ
- エリア境界で、エリア境界ルーターが生成する



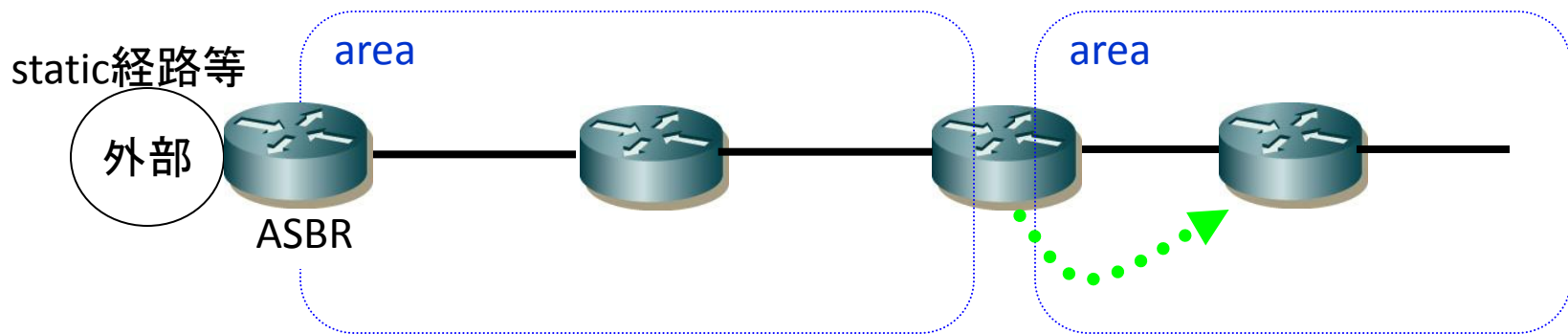
# LSタイプ3 サマリLSA



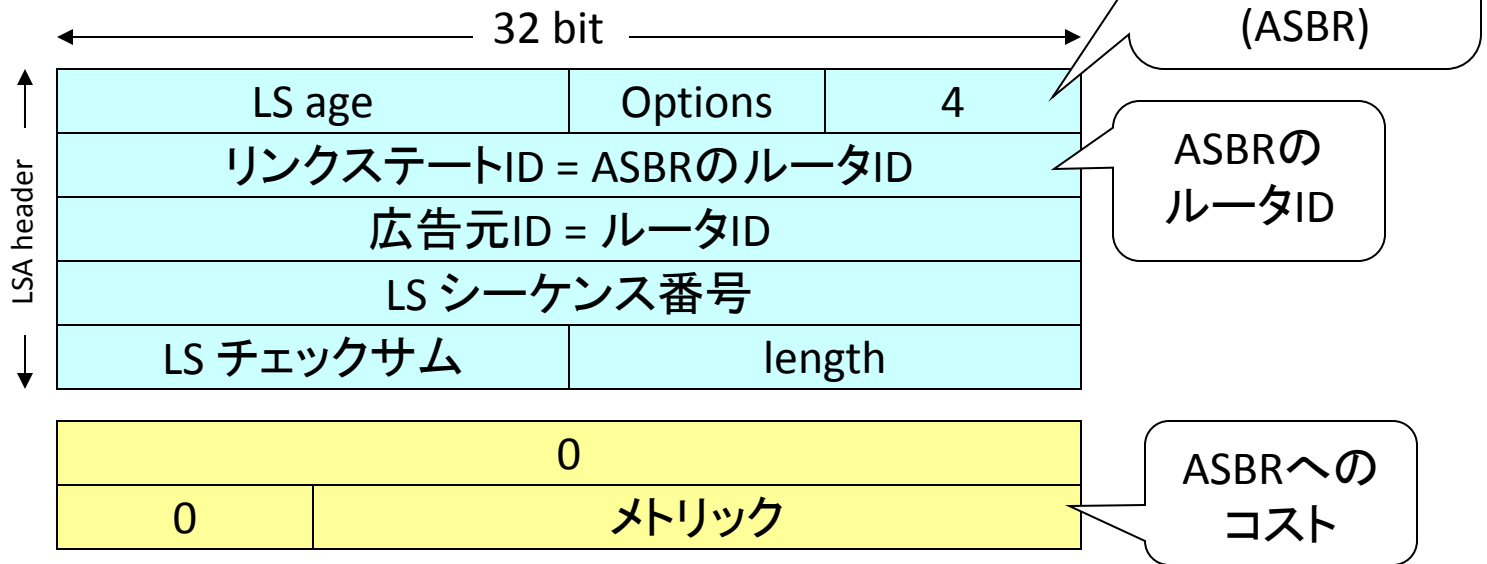


# LSタイプ4 - サマリLSA

- エリア外のASBRへの経路情報を運ぶ
- エリア境界で、エリア境界ルータが生成する
- LSタイプ3とほぼ一緒
  - リンクステートIDがルータID、ネットマスクが0

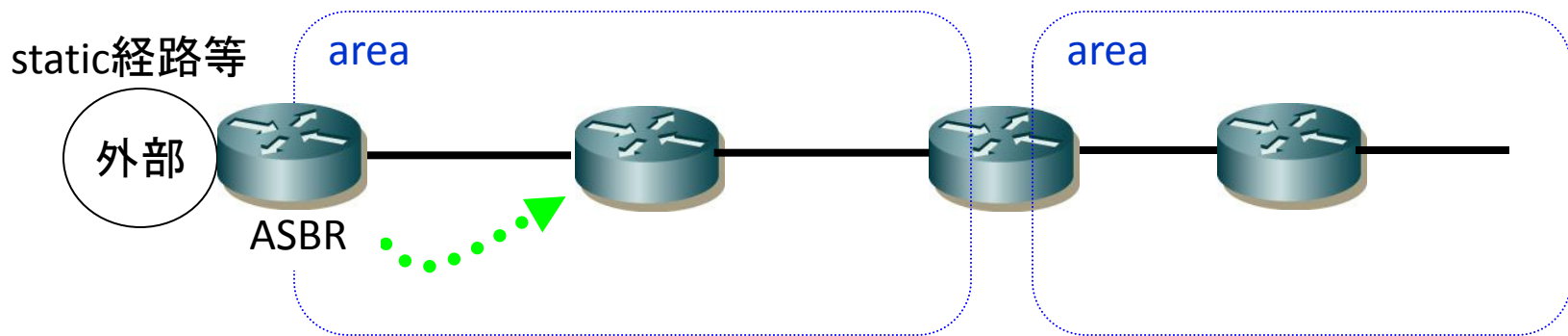


# LSタイプ4 サマリLSA

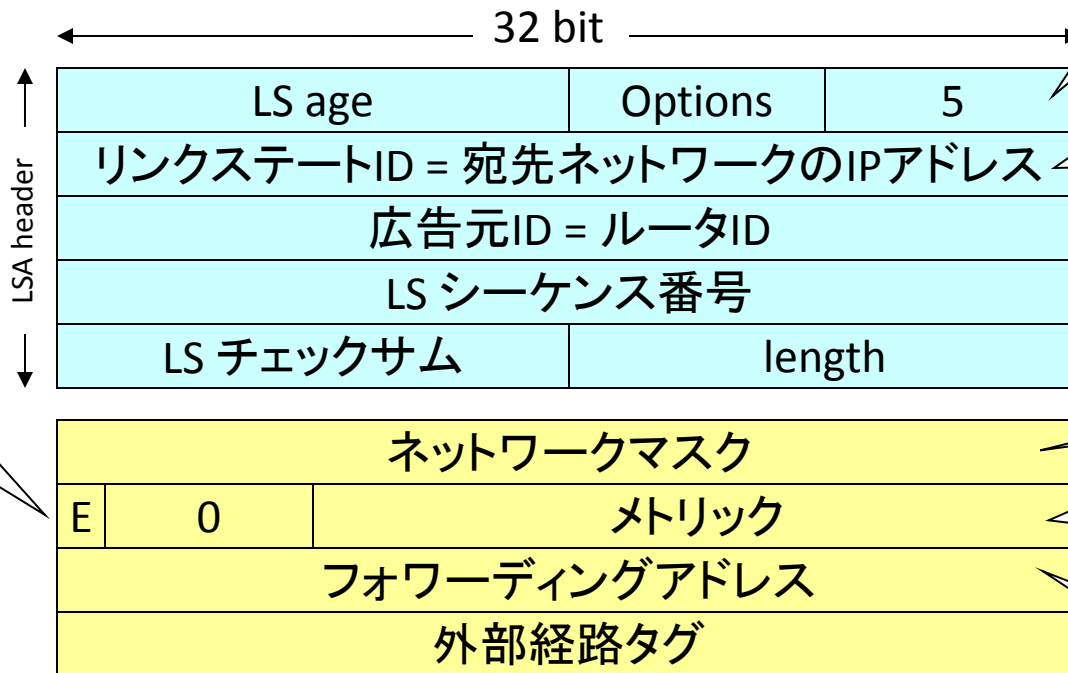


# LSタイプ5 - AS-external-LSA

- staticや他のプロトコルで学習した経路情報を外部経路としてOSPF内で運ぶ
- AS境界ルータ(ASBR)が生成する
- Type1またはType2のメトリック



# AS-external-LSA



LSタイプ5 = AS-external LSA

宛先ネットワークのIPアドレス

ネットワークのネットマスク

E bitに応じたメトリック

トラヒックの転送先。0.0.0.0だとASBR向けに経路が向く

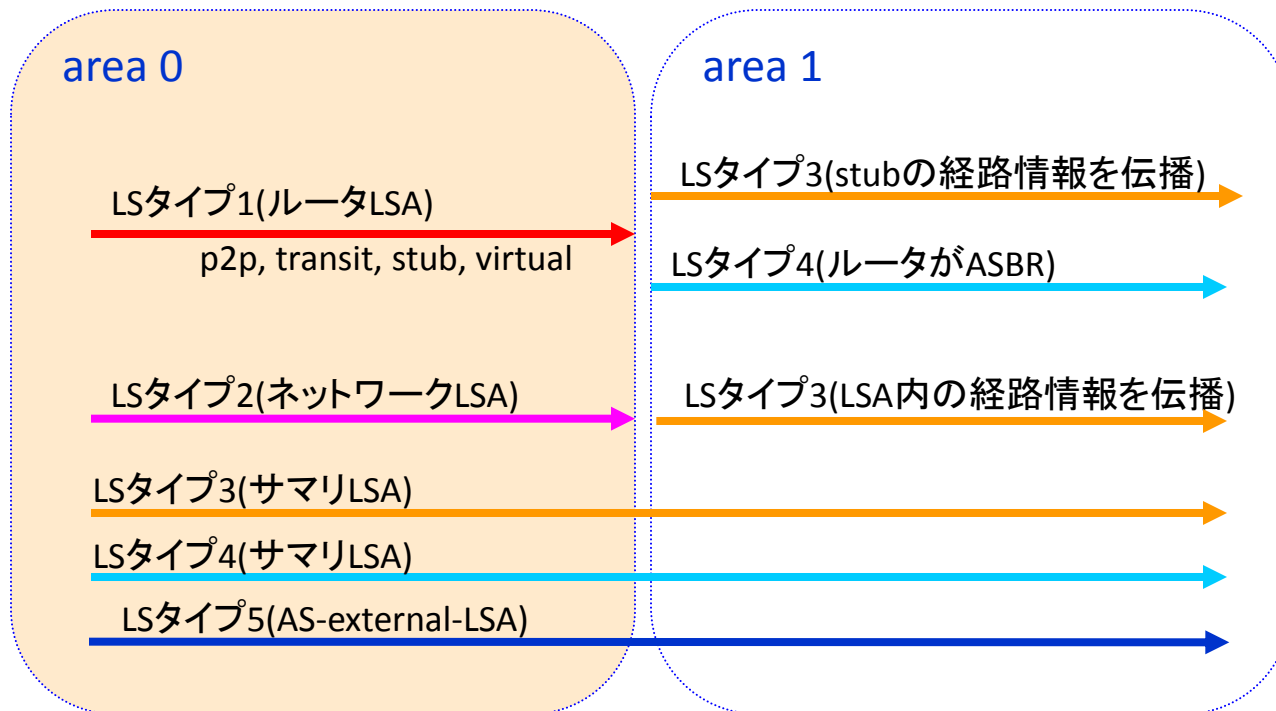
メトリックタイプ  
0:Type1  
1:Type2

# LSAとエリア

エリア間のLSAの伝播を整理する

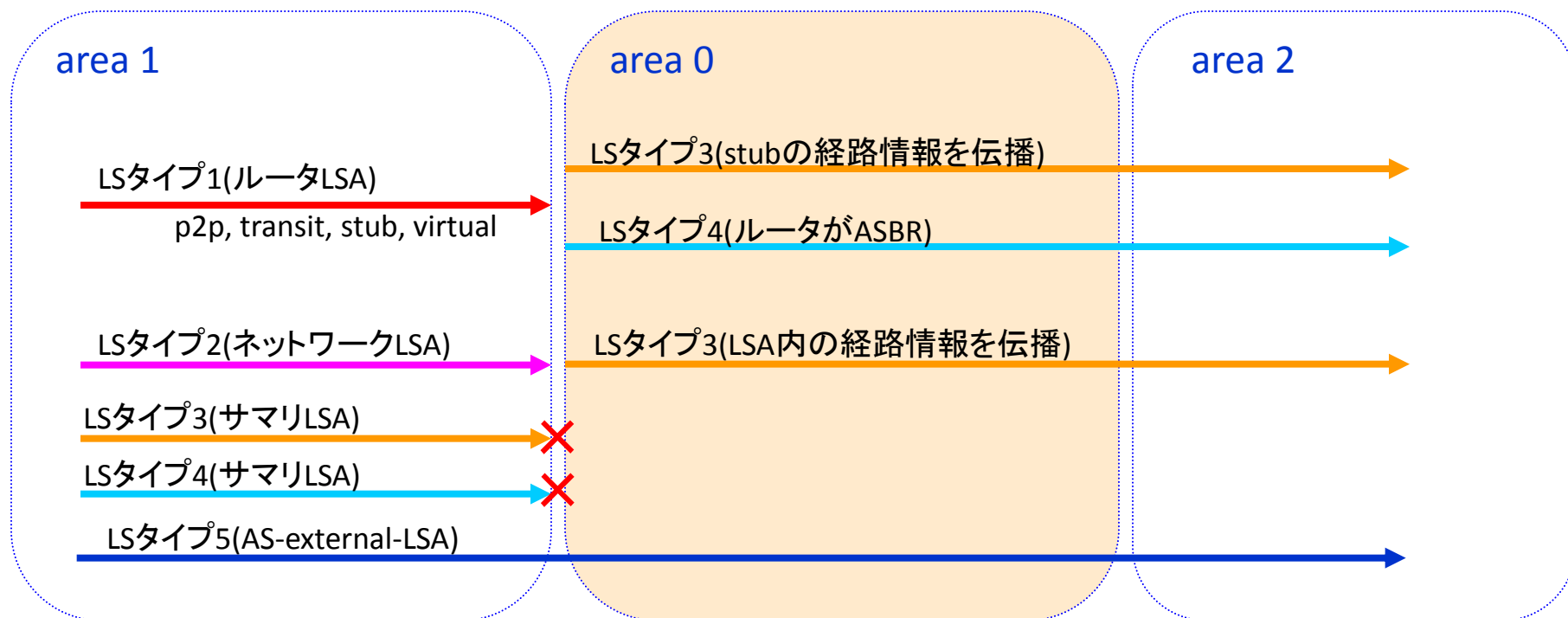
# バックボーンから他エリアへのLSA

- ルータLSA,ネットワークLSAの経路部分が伝播する
- その他のLSAはそのまま伝播する



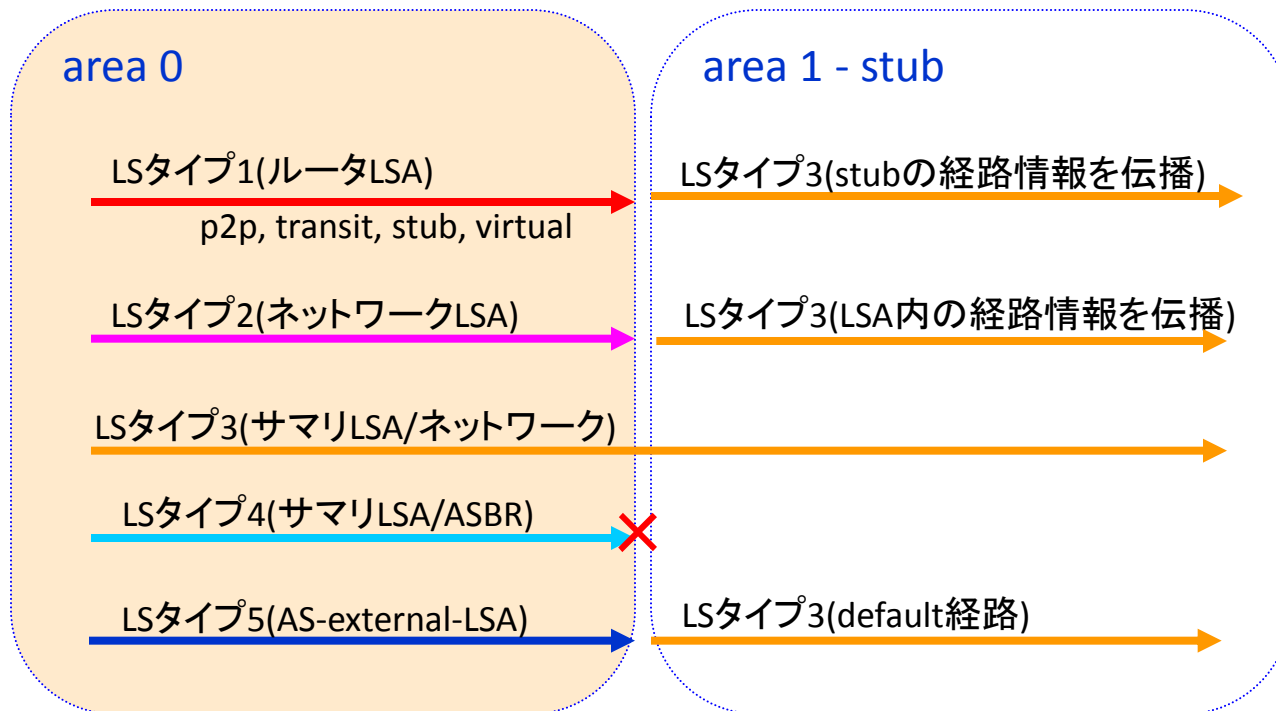
# エリア間でのLSA

- バックボーンエリアのみがサマリLSAを他エリアに中継できる



# stubエリアへのLSA

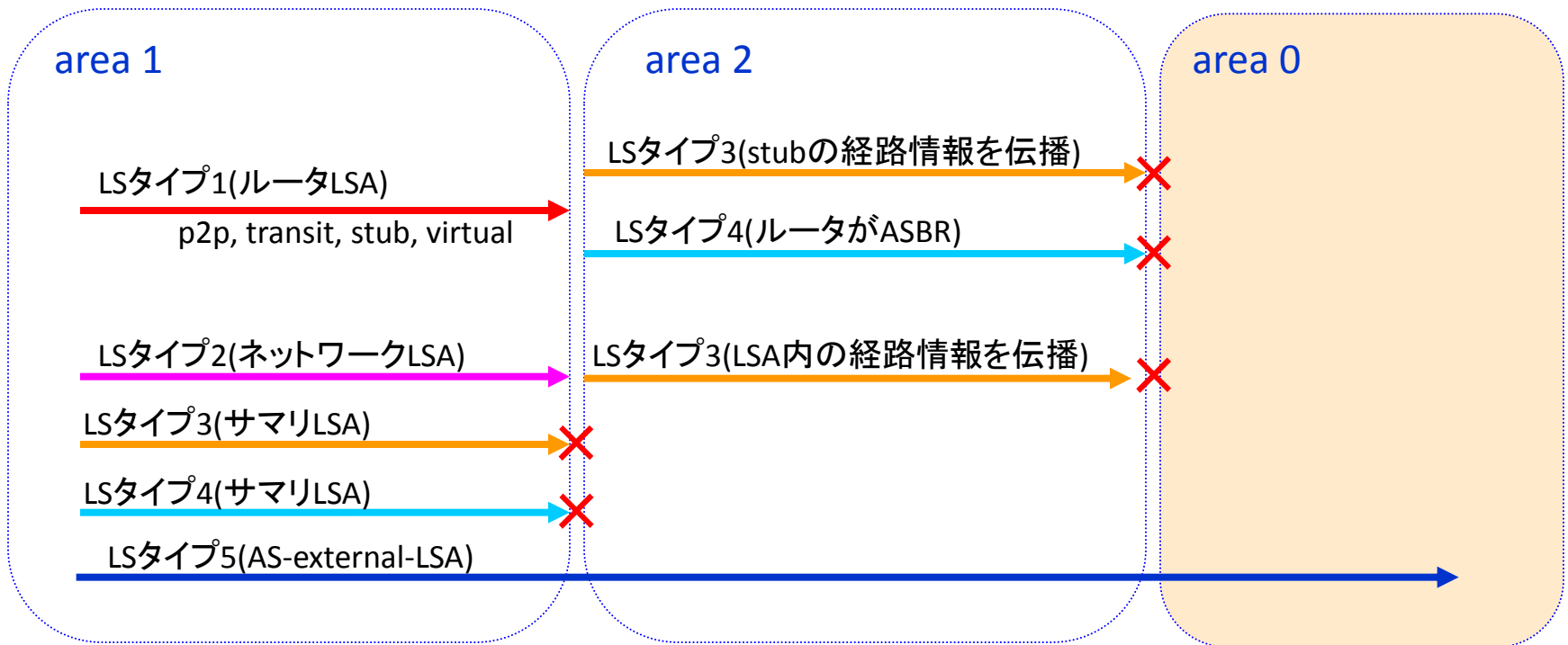
- 外部経路が伝播せず、default経路が広告される
- ASBRへの経路も必要ないので、伝播しない





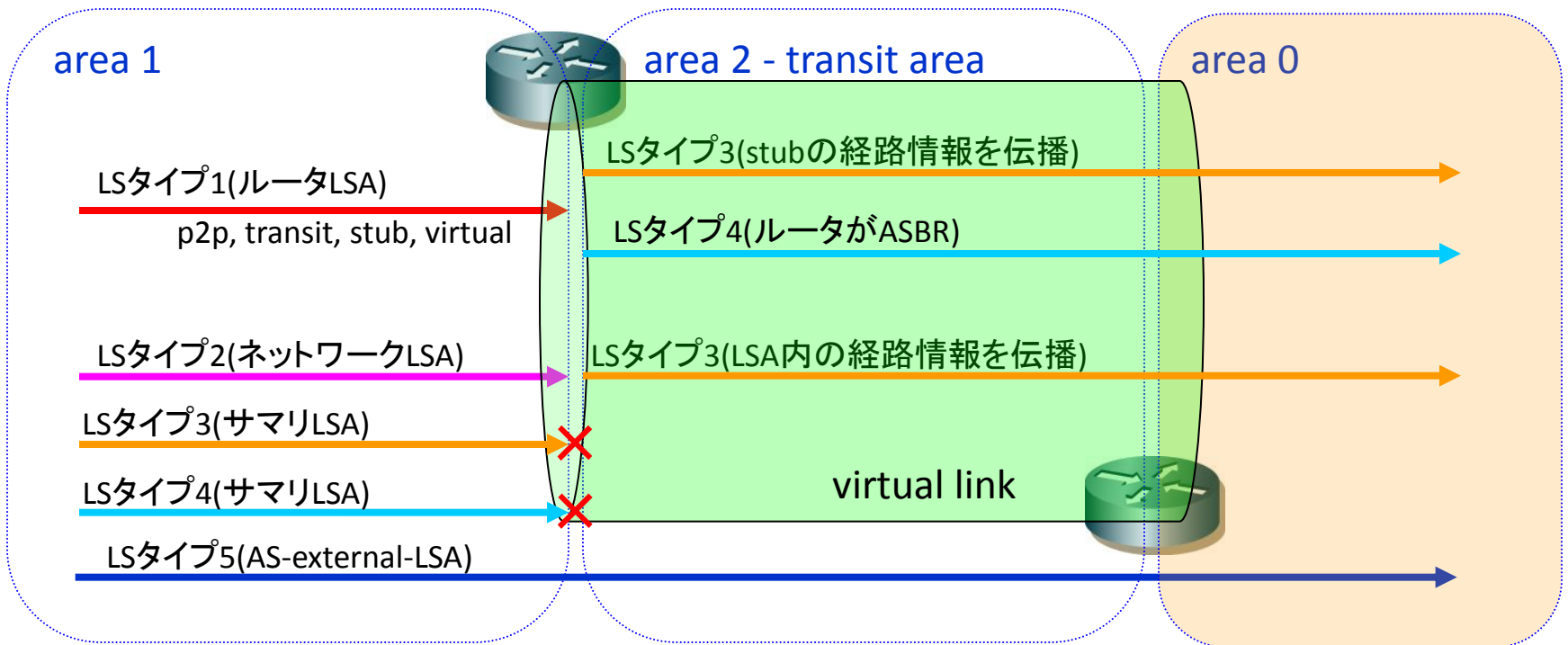
# エリア構成を誤った場合

- area 1がバックボーンエリアに接していない
- ほとんどの経路がバックボーンに届かない
- 外部経路も転送先アドレスが到達できないと考えられるので、ほぼ全ての経路が利用できない



# virtual linkの利用

- area 0が張り出している様にLSAのやり取りを行う
- 構成が複雑になるので、緊急時以外お勧めしない



# OSPF経路の優先順序

1. エリア内経路(intra area経路)
  - ルータLSA(LSタイプ1)
  - ネットワークLSA(LSタイプ2)
2. エリア間経路(inter area経路)
  - サマリLSA(LSタイプ3, 4)
3. 外部経路タイプ1
  - AS-external-LSA(LSタイプ5) メトリックタイプ1
4. 外部経路タイプ2
  - AS-external-LSA(LSタイプ5) メトリックタイプ2

# OSPFv3

- OSPF for IPv6のこと
  - 詳しくは[RFC2740]
  - IPv6に対応するために、変更が加えられた
- トポロジ情報と経路情報の分離
  - ルータLSA、ネットワークLSAから経路を削除
    - ルータLSAのstubネットワーク
    - ネットワークLSAのネットマスク
  - 代わりにIntra-Area-Prefix-LSAを用意
- LSAにFlooding Scopeの要素が追加

# OSPFv3

- LSAを分かりやすく改名
  - サマリLSA(LSタイプ3)→Inter-Area-Prefix-LSA
  - サマリLSA(LSタイプ4)→Inter-Area-Router-LSA
- リンクの識別手法の変更
  - OSPFv2 – 3種類
    - リンクタイプ
    - リンクID、リンクデータ
  - OSPFv3 – 4種類
    - リンクタイプ、
    - インタフェースID、ネイバインタフェースID、ネイバルータID

# 他のプロトコルとの比較

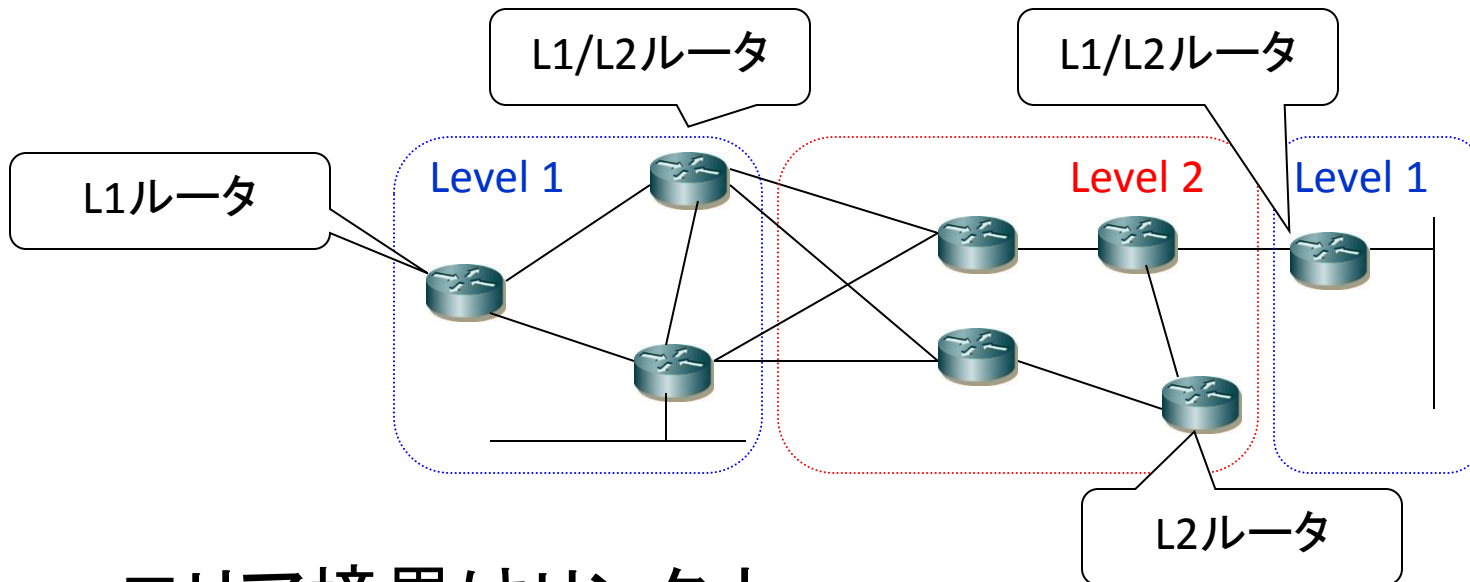
- OSPFとIS-IS
  - 1980年台の後半に開発が始まる
  - IS-ISのdraftが原型となり、OSPFが開発される
  - その後、各ベンダが実装
  - 現在もIETFのそれぞれのworking groupで議論が続く
  - どちらもSPFアルゴリズムを利用してリンクステートデータベースから経路情報を計算する

# IS-ISとOSPFv2

- IS-ISでIPの経路制御を出来るように拡張したものが、Integrated IS-IS。または Dual IS-IS

	Integrated IS-IS	OSPFv2
プロトコルパケットの転送	CLNS	IPv4
扱う経路情報	CLNS and/or IPv4	IPv4
階層化	Level1(エリア内) Level2(エリア間)	中継を担うarea0と その他のエリア
エリア境界	リンク	ルータ

# IS-IS



- エリア境界はリンク上
- Level2がエリア間の通信を担う
- Level1からエリア外への通信は近隣のL1/L2ルータに頼る
  - L2の経路をL1内に伝播させることもできる[RFC2966]



# IPv4/IPv6とIGP

- 一つのIGPで両方処理する
  - IPv4/IPv6が完全に同じネットワーク構成であれば問題なく運用できる
  - IS-ISで実現可能。将来はOSPFv3も行けるかも
- それぞれ異なるIGPを採用
  - 多少経路計算が冗長になる
  - IPv4 - OSPF (or IS-IS)
  - IPv6 - OSPFv3 (or IS-IS)

**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の基本アイデア

- 準備
  - 経路交換したいBGPルータとTCPでネイバを構築
- 通知
  - ベスト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ーを適用し、パス情報を元にベスト経路を計算

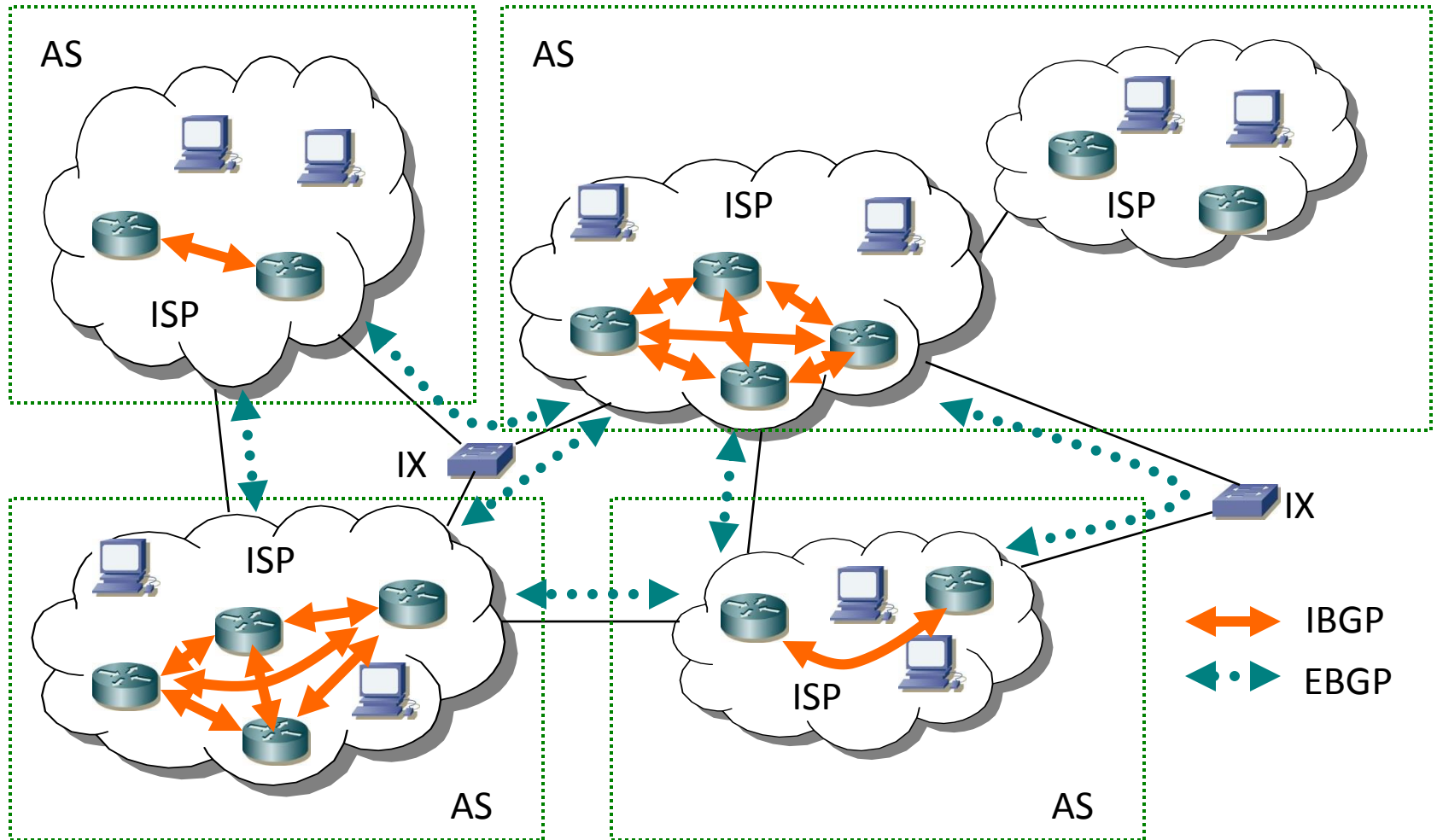
# BGP RFCs

- 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

# BGP用語

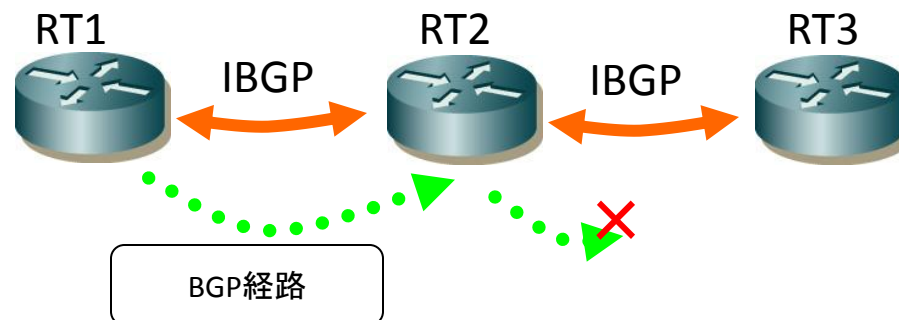
- BGP ID
  - ルータを識別する32bitの数値
  - インタフェースのIPアドレスから選ばれる
  - 実運用では変更が発生しないようにloopbackインタフェースに付与したIPアドレスを利用する
- NLRI
  - Network Layer Reachability Information
  - ネットワーク層到達可能性情報
  - prefixで示される宛先のこと

# BGPの世界



# IBGP(Internal BGP)

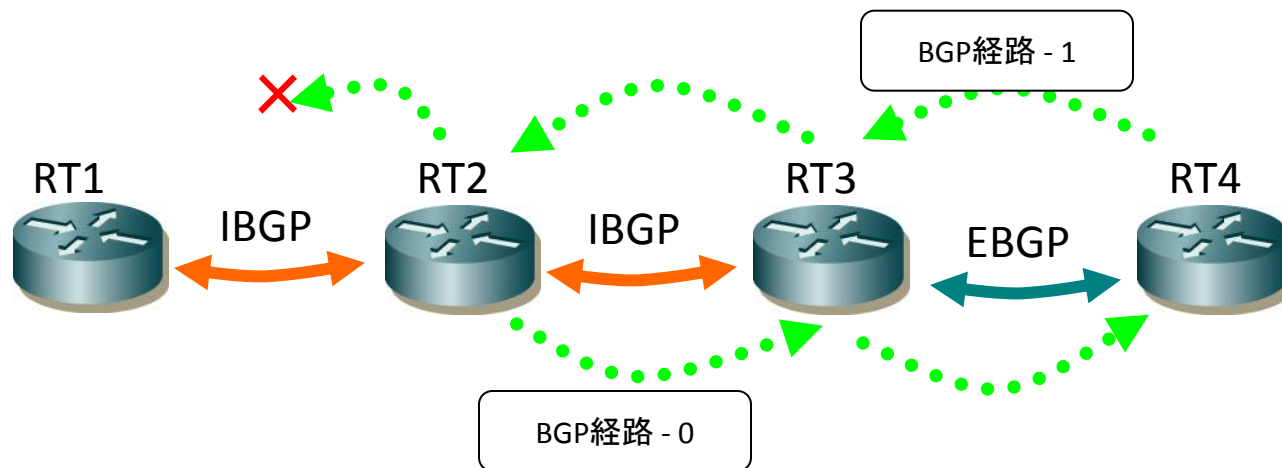
- 同じAS内でのBGP接続
- IBGPで受信した経路は他のIBGPルータに広報されない
  - 全ての経路を伝えるには、AS内の全BGPルータがfull-meshでIBGPを張る必要がある



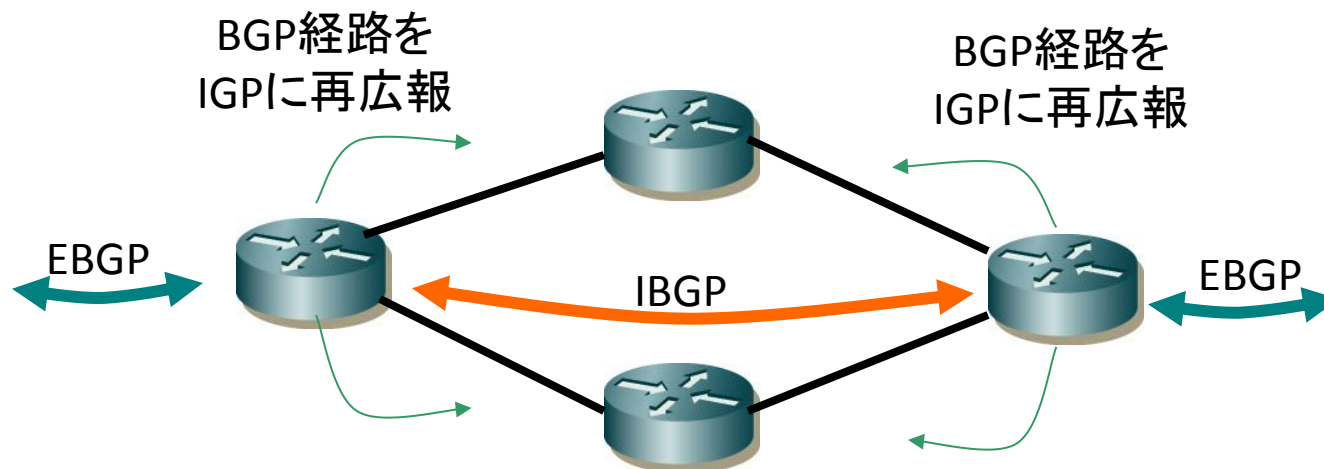


# EBGP(External BGP)

- 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

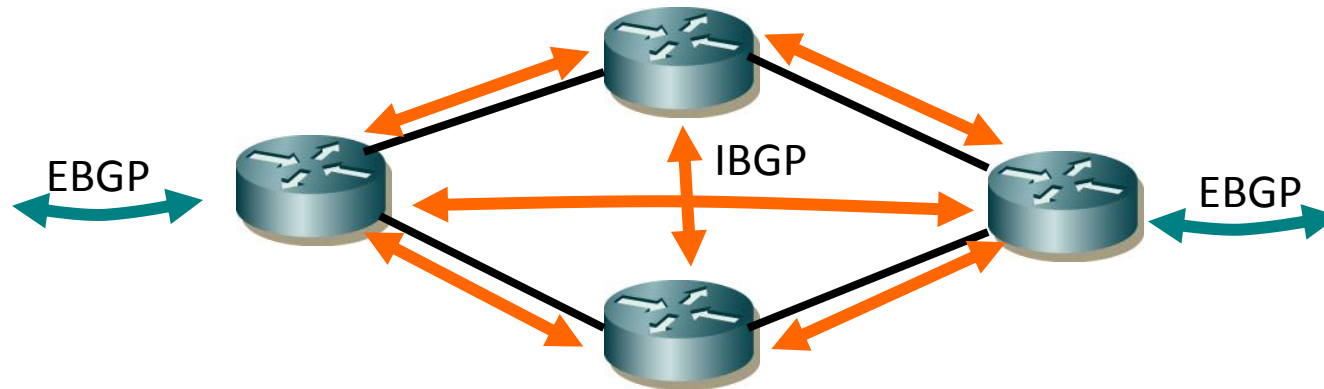


# BGPのいにしえのモデル



- 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
- 経路数が増大すると破綻

# 経路数の増大に対応したBGPモデル



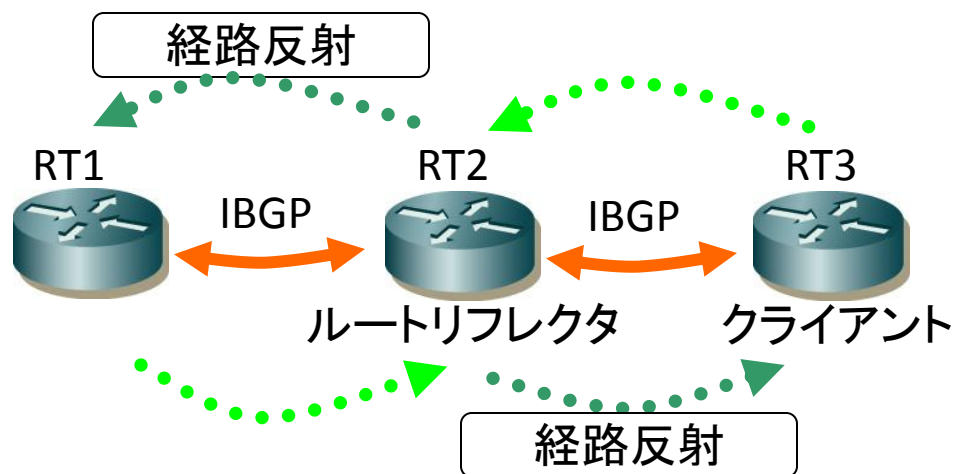
- 主要なルータは全てBGPルータ
  - IGPはトポロジと最低限の経路を運び、BGPでその他の全ての経路を運ぶ
- ・・・IBGP接続の増大

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

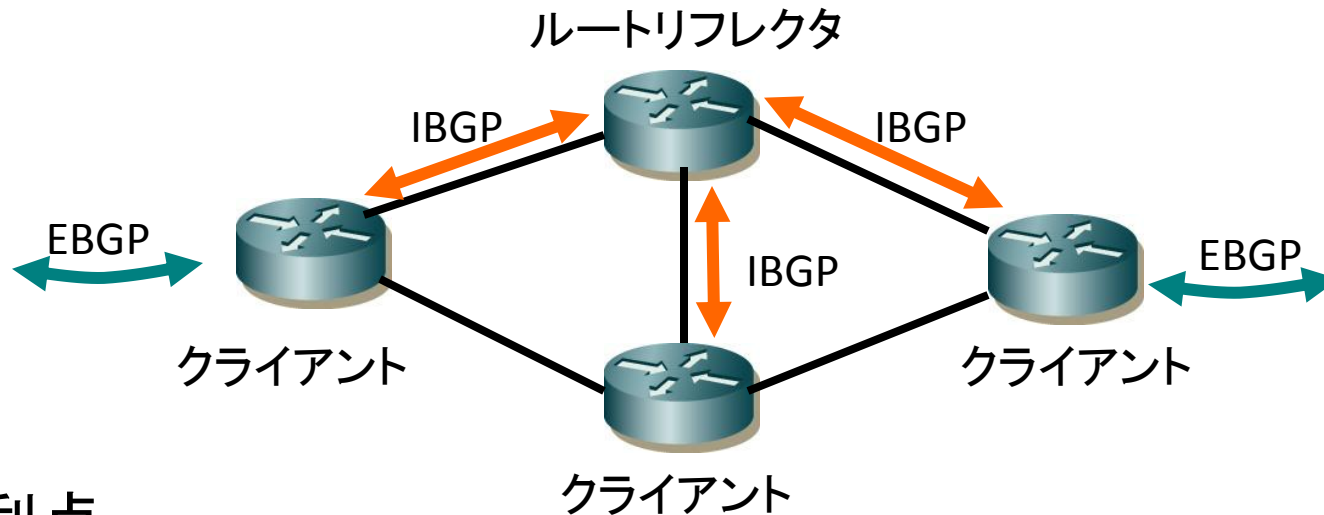
- AS内にBGPルータが増える毎にIBGP接続が増大していく
  - 20台目のBGPルータが接続すると19接続追加
  - ルーターリソースの問題、設定負荷の問題
- 解決策の模索
  - [RFC4456] ルートリフレクタ
  - [RFC3065] コンフェデレーション
  - 気にせずリソースを強大にする
  - ルーターを減らす

# 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

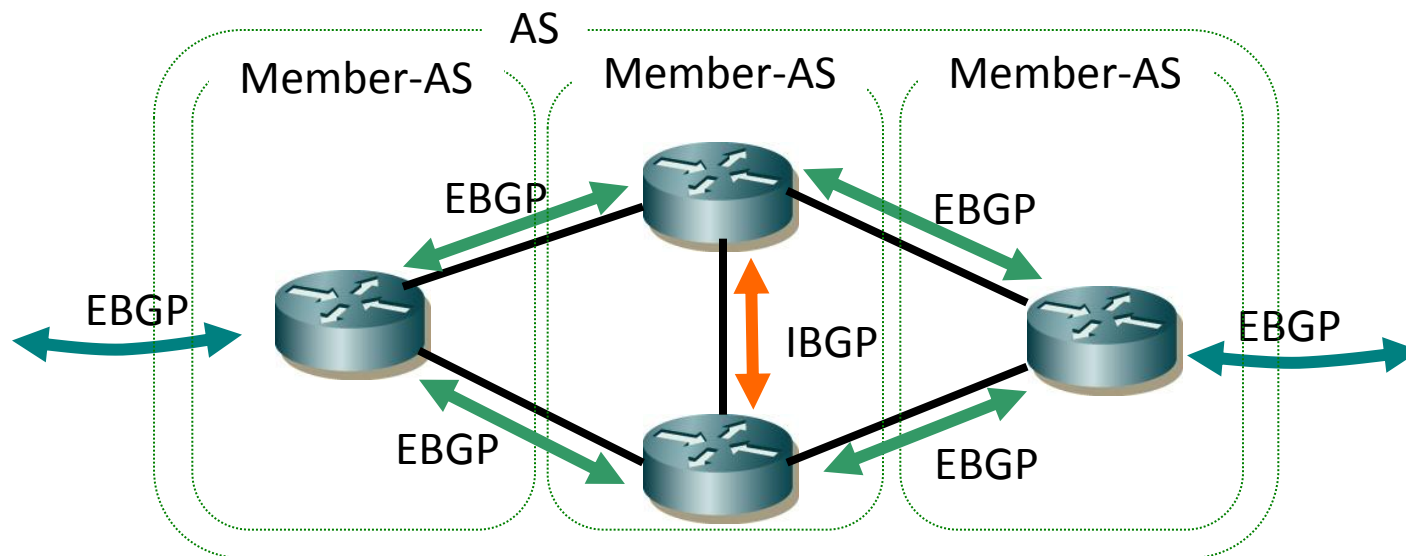
- IBGP接続数が削減できる
- 比較的容易に導入できる

- 欠点

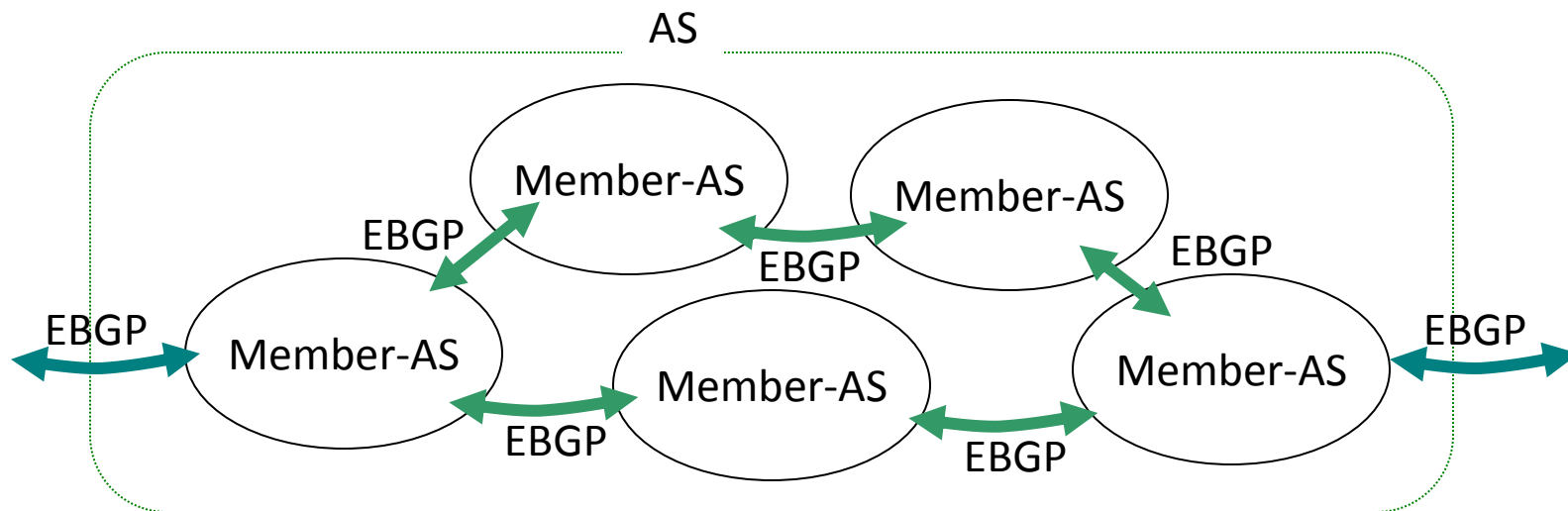
- 経路削除時に、UPDATEが増える可能性がある
- 経路情報が隠蔽されるため最適ではない経路を選ぶ可能性がある
  - リフレクタの階層はできるだけ物理トポロジに合わせるべし！

# 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



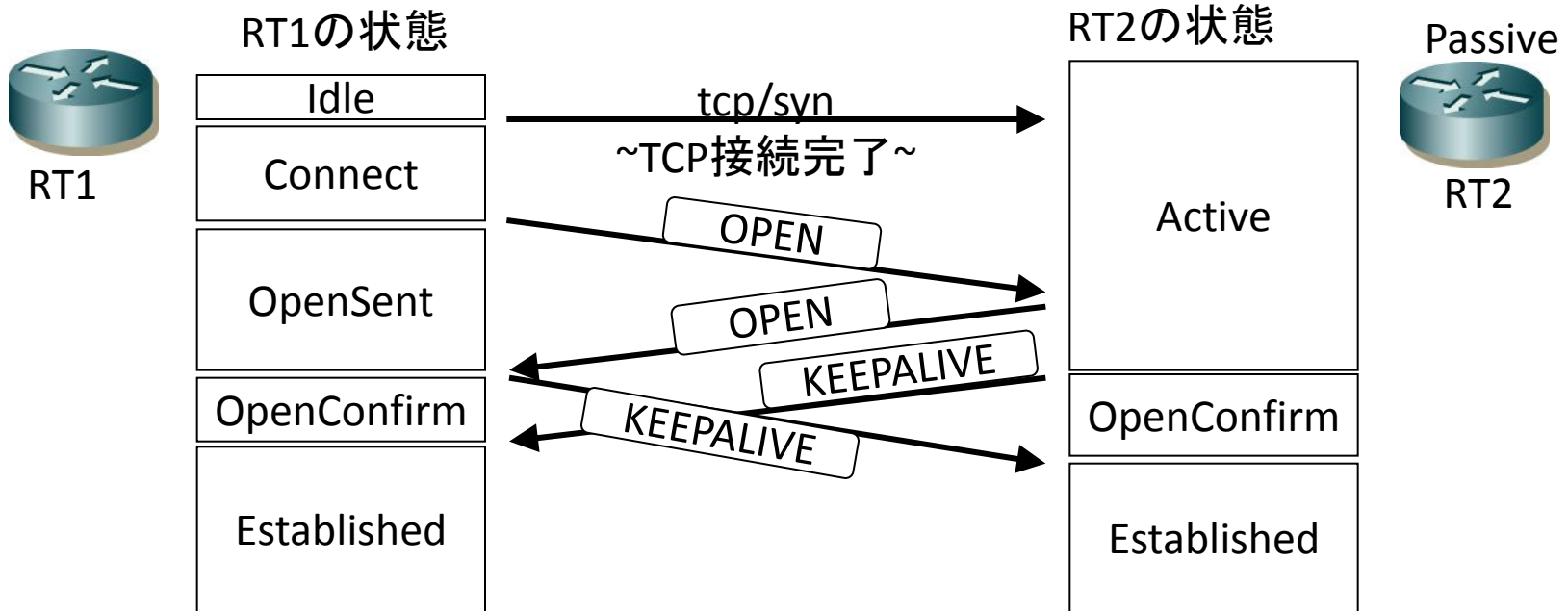
- 利点
  - IBGP接続数が削減できる
  - 管理区分を分けられる
- 欠点
  - 経路削除時にUPDATEが増える可能性がある
  - 経路情報が隠蔽されるため最適ではない経路を選ぶかもしれない



# BGPパッケージ

BGPのプロトコルパッケージの  
フォーマットを解説する

# BGP接続の確立



Idle – 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

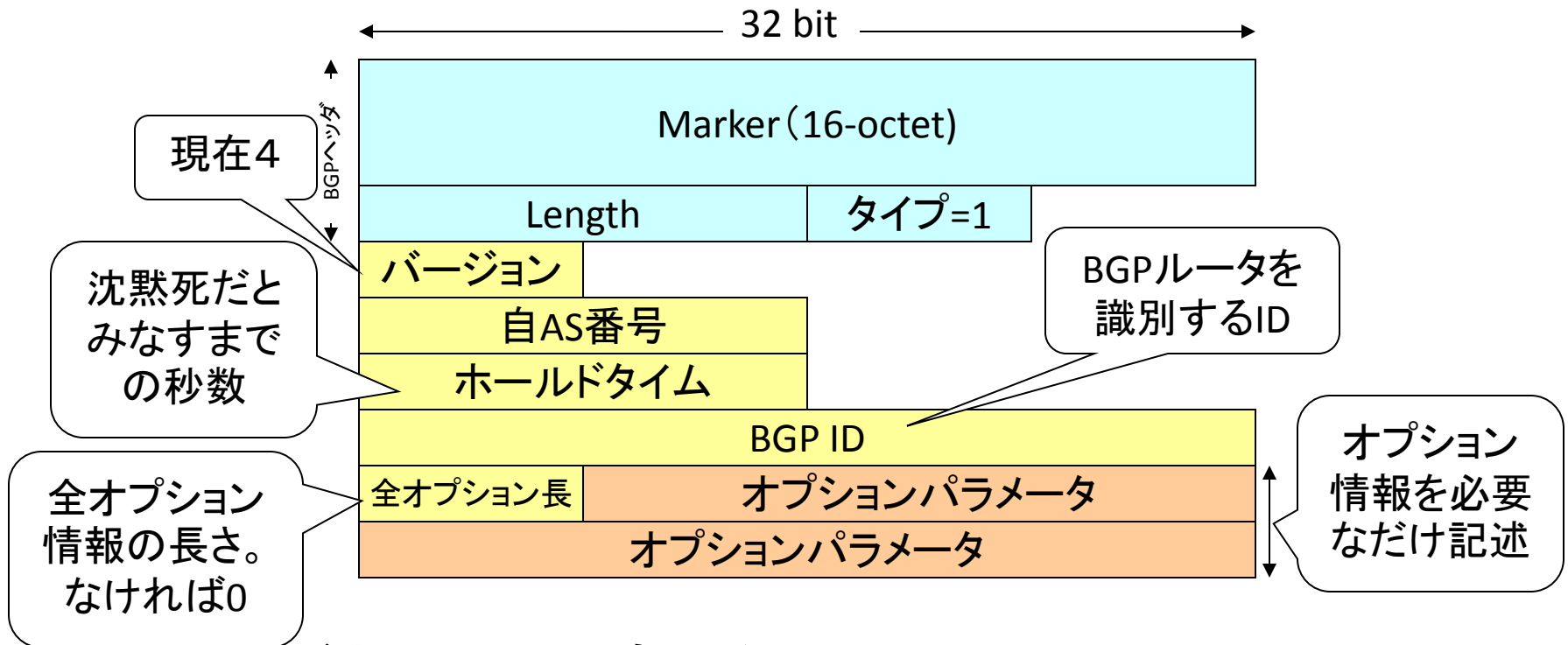


- Marker(マーカ)
  - 16-octetの全bitが1
  - 過去との互換性のため
- Length
  - 2-octetのメッセージ長
  - 19～4096
- タイプ(1-octet)
  1. OPEN
  2. UPDATE
  3. NOTIFICATION
  4. KEEPALIVE
  5. ROUTE\_REFRESH

# タイプ1 OPENメッセージ

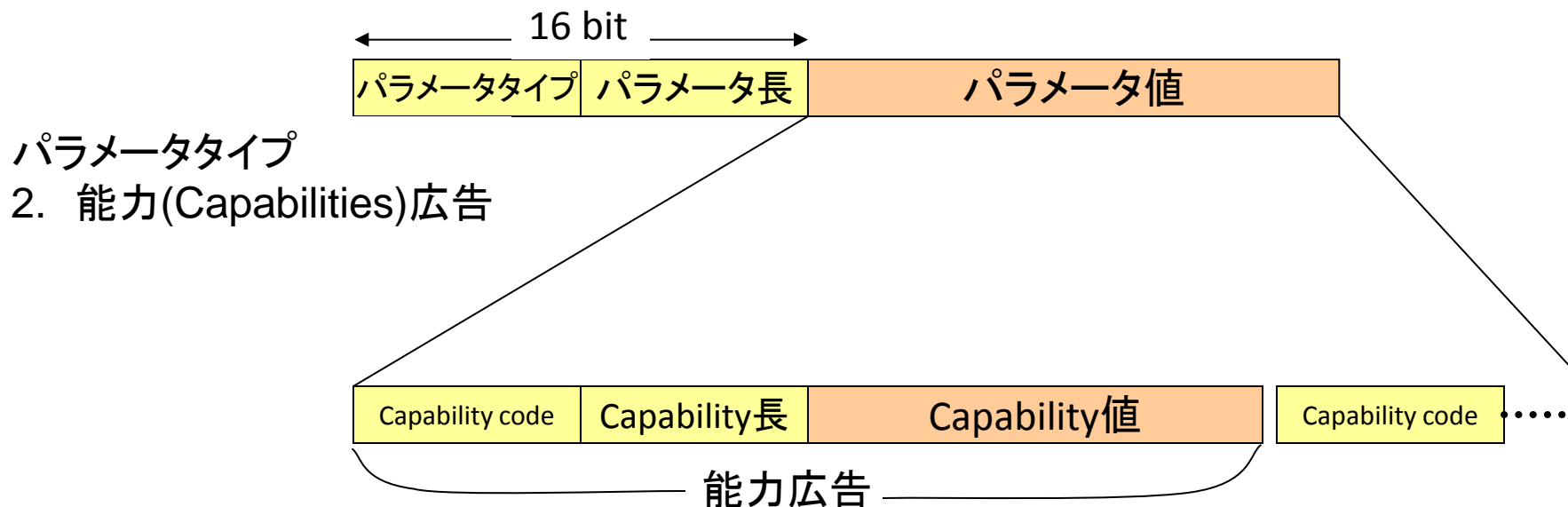
- TCP接続が確立後、最初にやりとりされる
- パラメタの交換
  - バージョン、AS番号やBGP ID、ホールドタイム
  - オプションパラメータで各種機能を通知しあう
- タイプ4 KEEPALIVEで接続確立

# 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

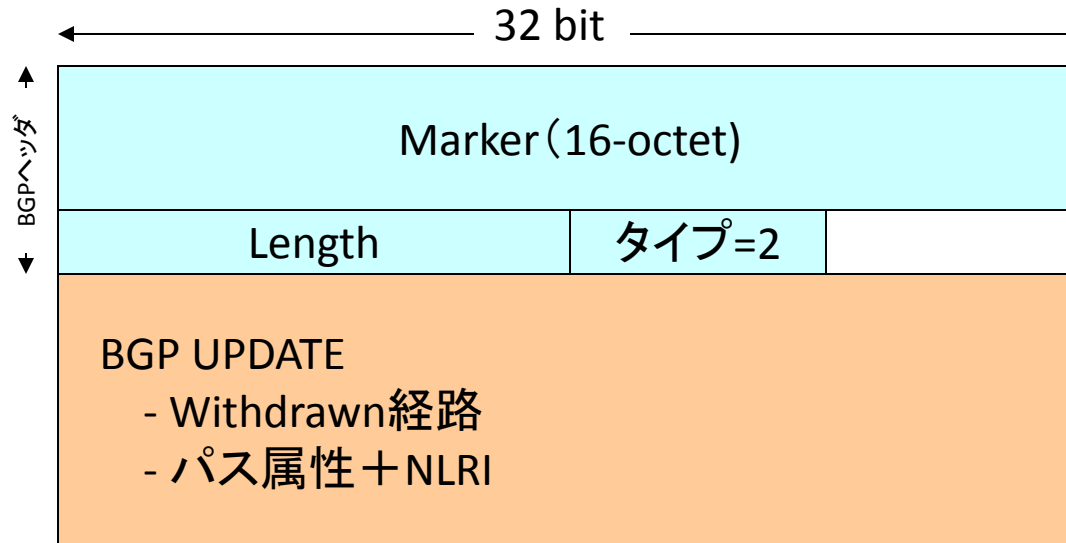
- 
- |   |                         |                      |
|---|-------------------------|----------------------|
| 1 | Multiprotocol Extension | サポートする<AFI, SAFI>の広告 |
|---|-------------------------|----------------------|
- 
- |   |               |                        |
|---|---------------|------------------------|
| 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



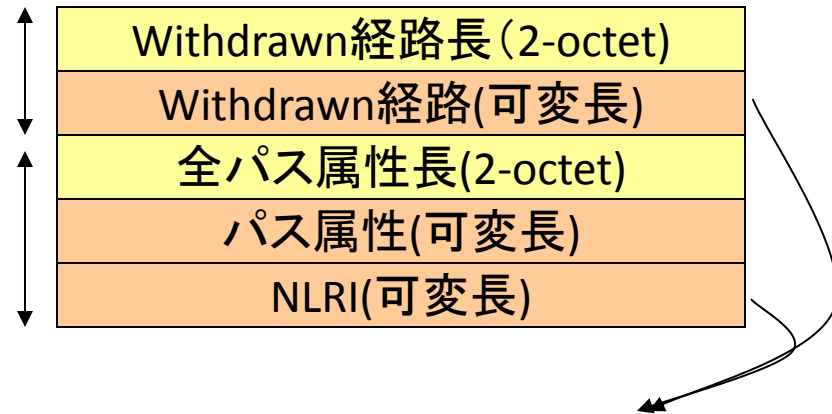
# タイプ2 UPDATEメッセージ



- 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

長さ(1-octet)	プレフィックス(可変長)
-------------	--------------

– 例: 10.0.0.0/8

8(1-octet)	10(1-octet)
------------	-------------

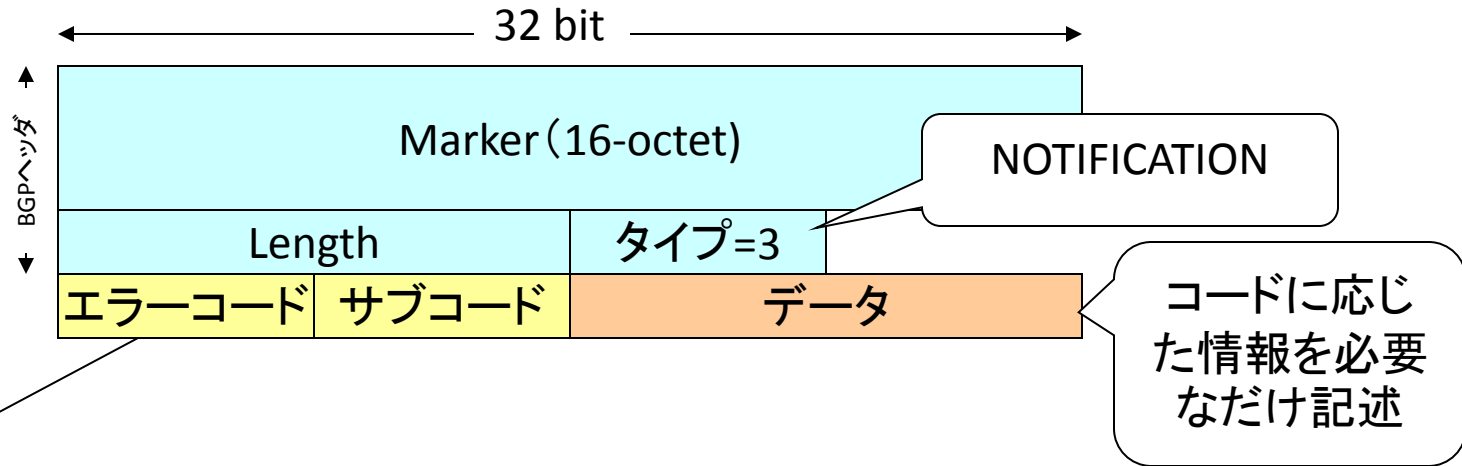
– 例: 10.0.0.127/25

25(1-octet)	10.0.0.127(4-octet)
-------------	---------------------

# 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

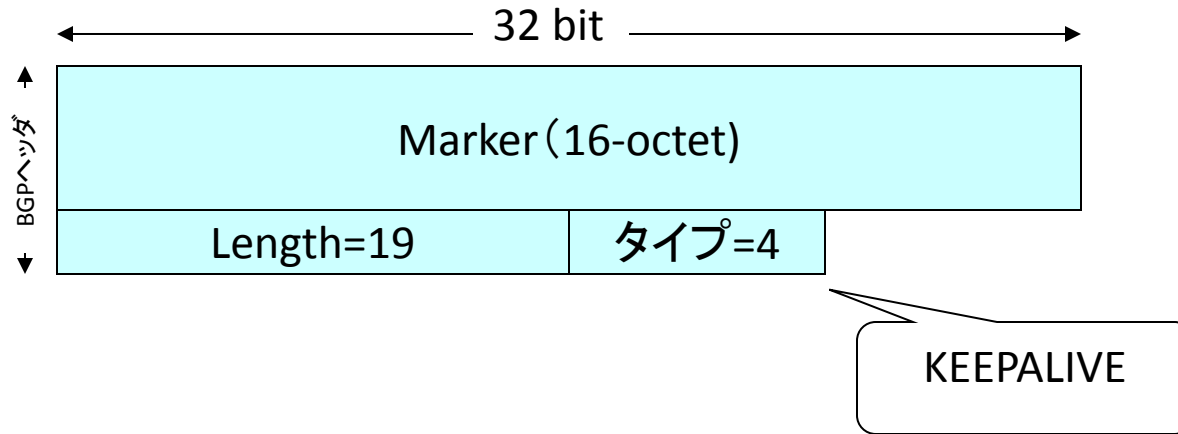


1. メッセージヘッダエラー
2. OPENメッセージエラー
3. UPDATEメッセージエラー
4. HoldTime超過
5. 状態遷移エラー
6. Cease

# 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

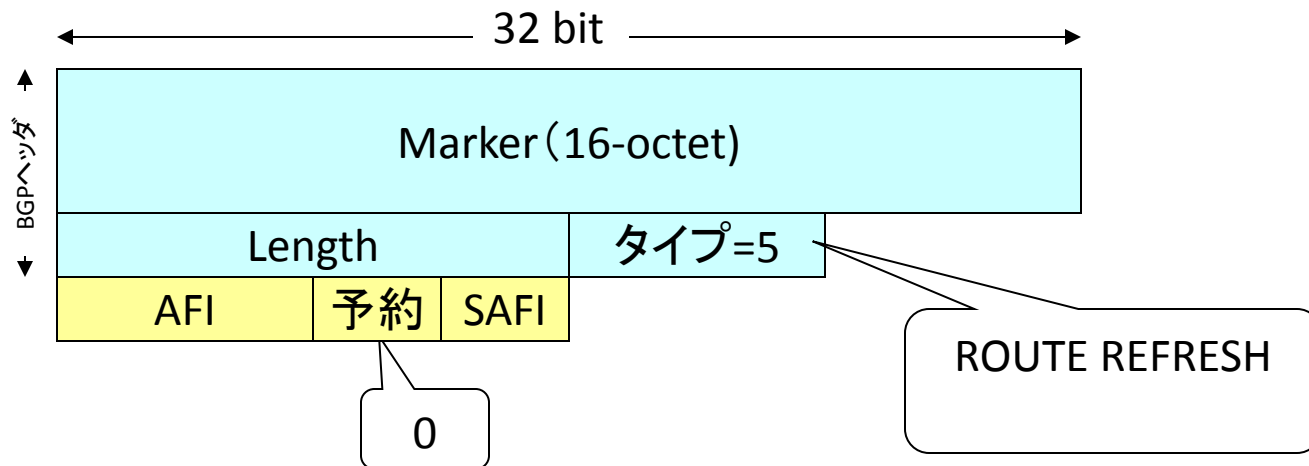


- KEEPALIVEであること以外、何も運ばない
- 最小のBGPメッセージ

# 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



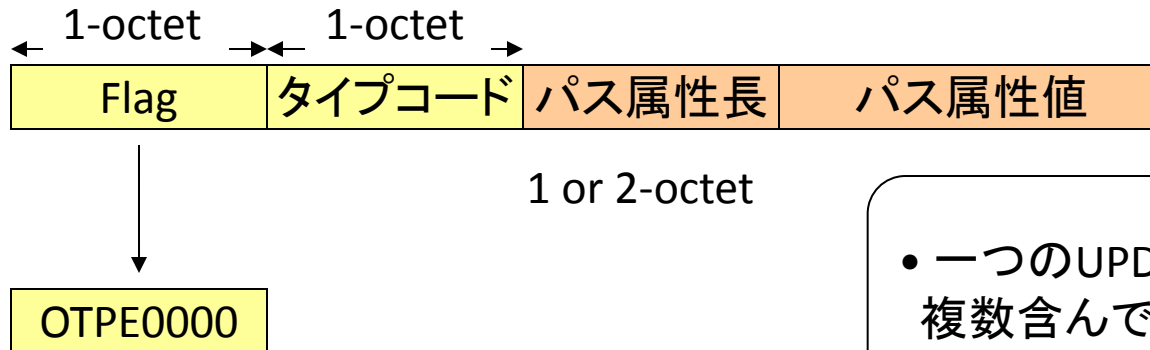
- AFI = Address Family Identifier
  - IPv4やIPv6など
- SAFI = Subsequent Address Family Identifier
  - UnicastやMulticastなど



# 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(パス属性の種類別)

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

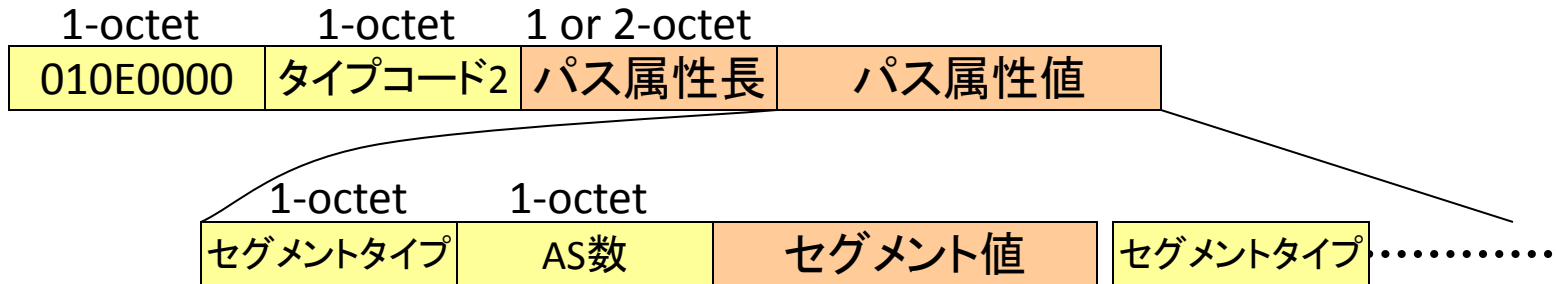
1 – EGP                      …… EGP[RFC904]から生成

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

- 新しい情報は先頭(左)に付加される

# AS\_PATH属性の処理

- 経路を転送する場合

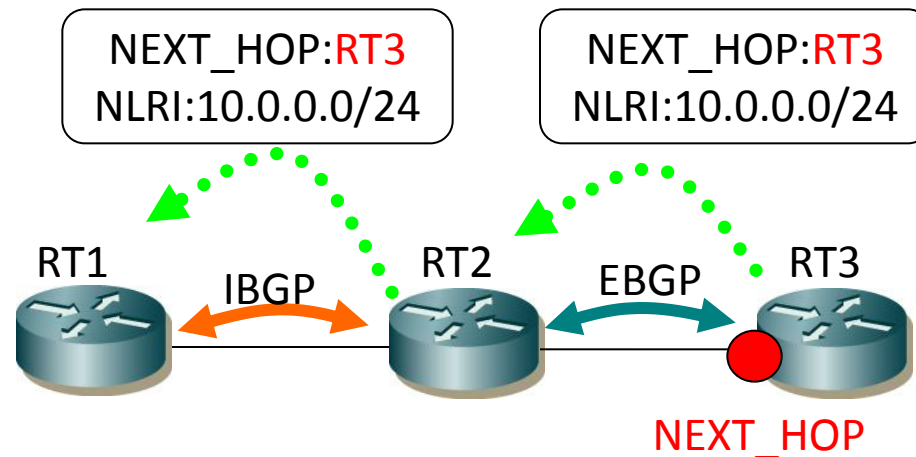
広報先	
IBGP	変更しない
EBGP	自AS番号をAS_SEQUENCEタイプでAS_PATH属性の先頭に付加する

- 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で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

- (0xFFFFFFFF01) **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
- (0xFFFFFFFF02) **NO\_ADVERTISE**
  - 他BGPルータに広報しない
- (0xFFFFFFFF03) **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
- (0xFFFFFFFF04) **NOPEER** [RFC3765]
  - 対等ピアには広報しない
  - まだ実装は無さそう

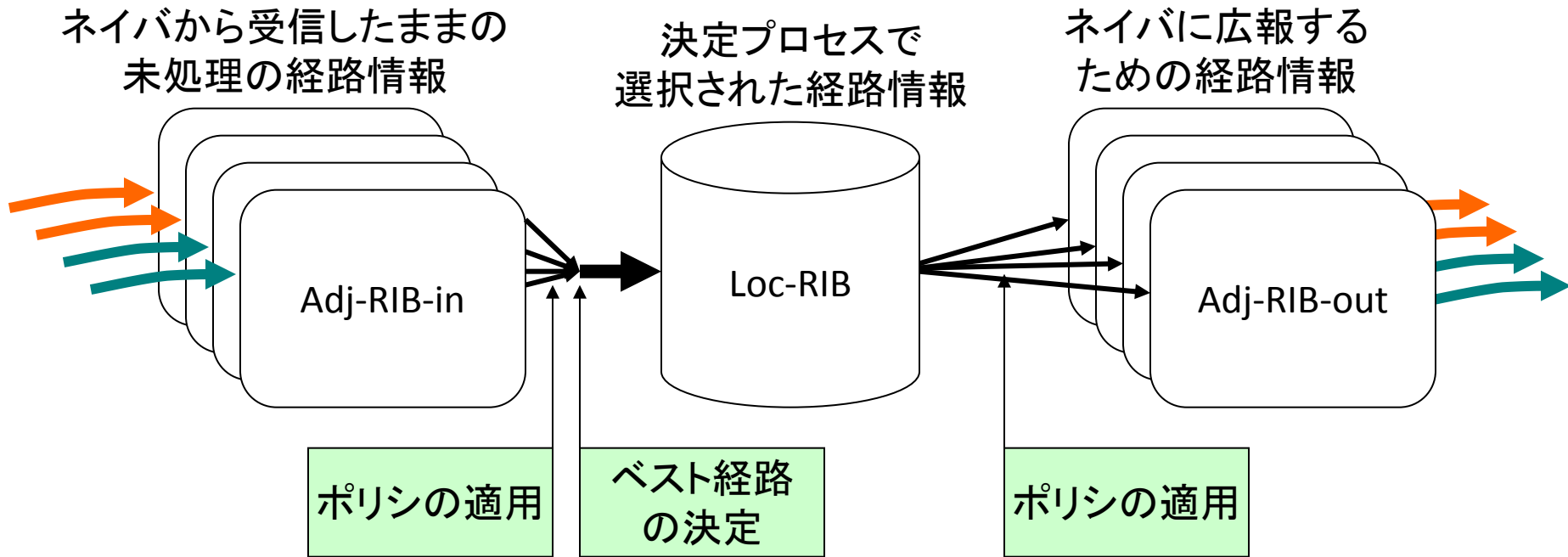
# EBGP & IBGPとパス属性

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

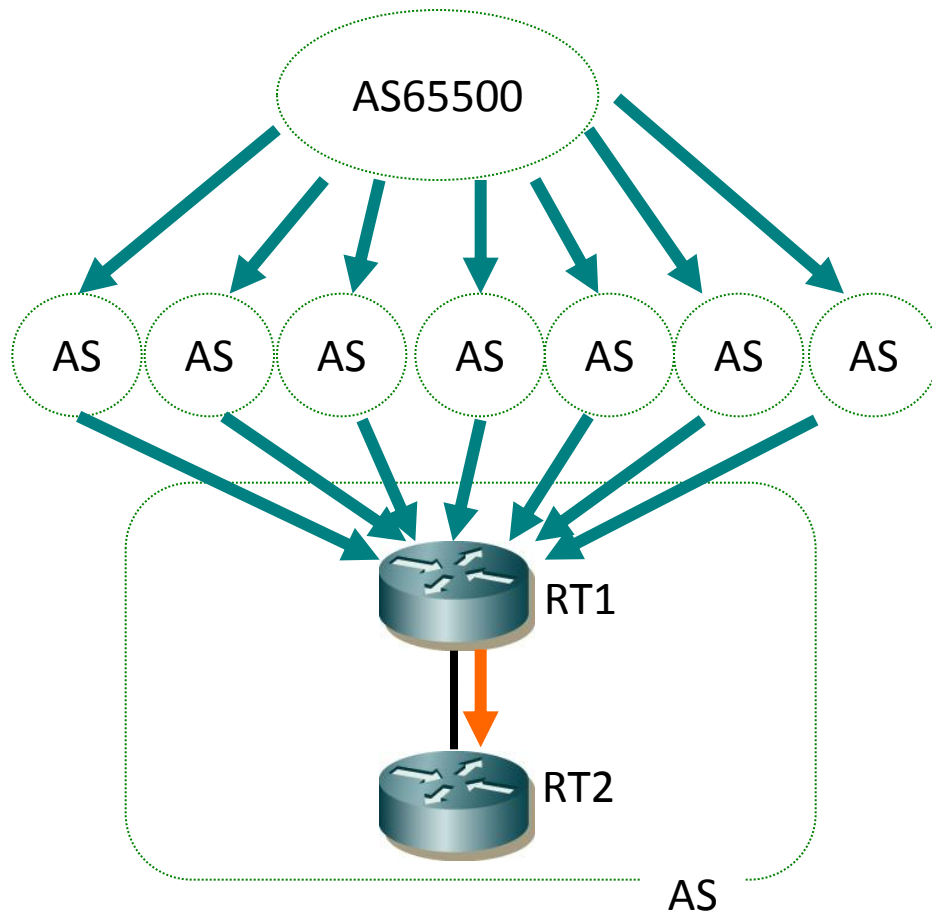
# BGPの経路処理



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



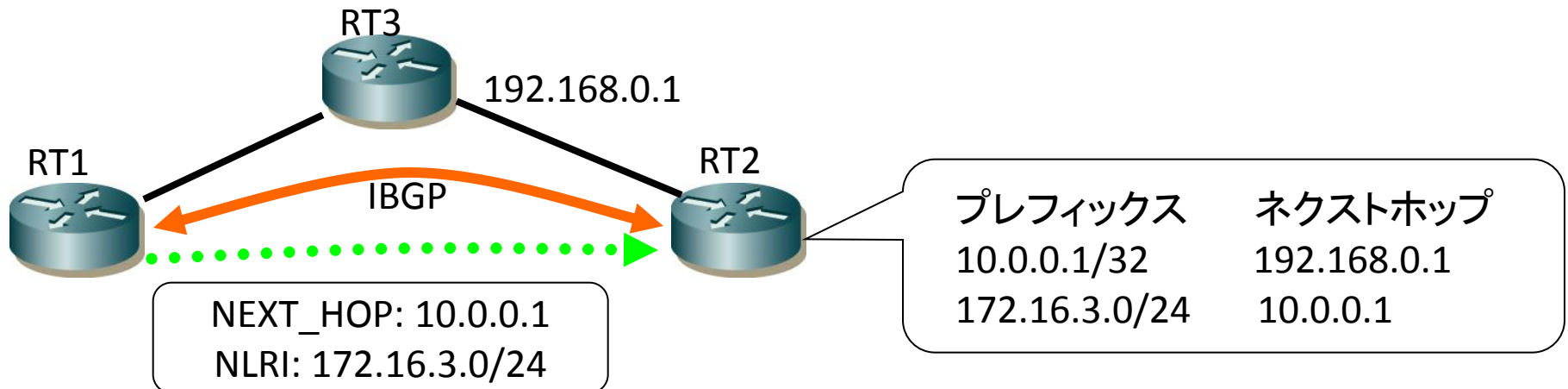
# ベスト経路のみを広報



- 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

# 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、ポリシーに依存)
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

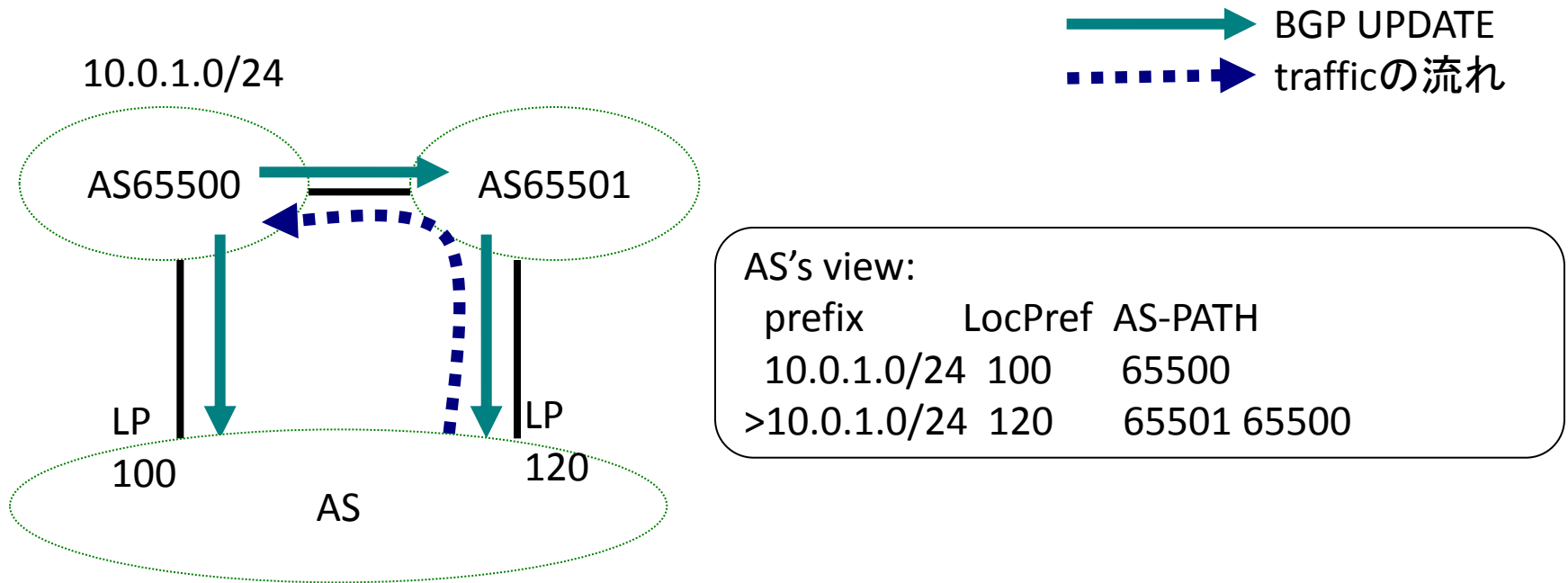
# 属性値の評価

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

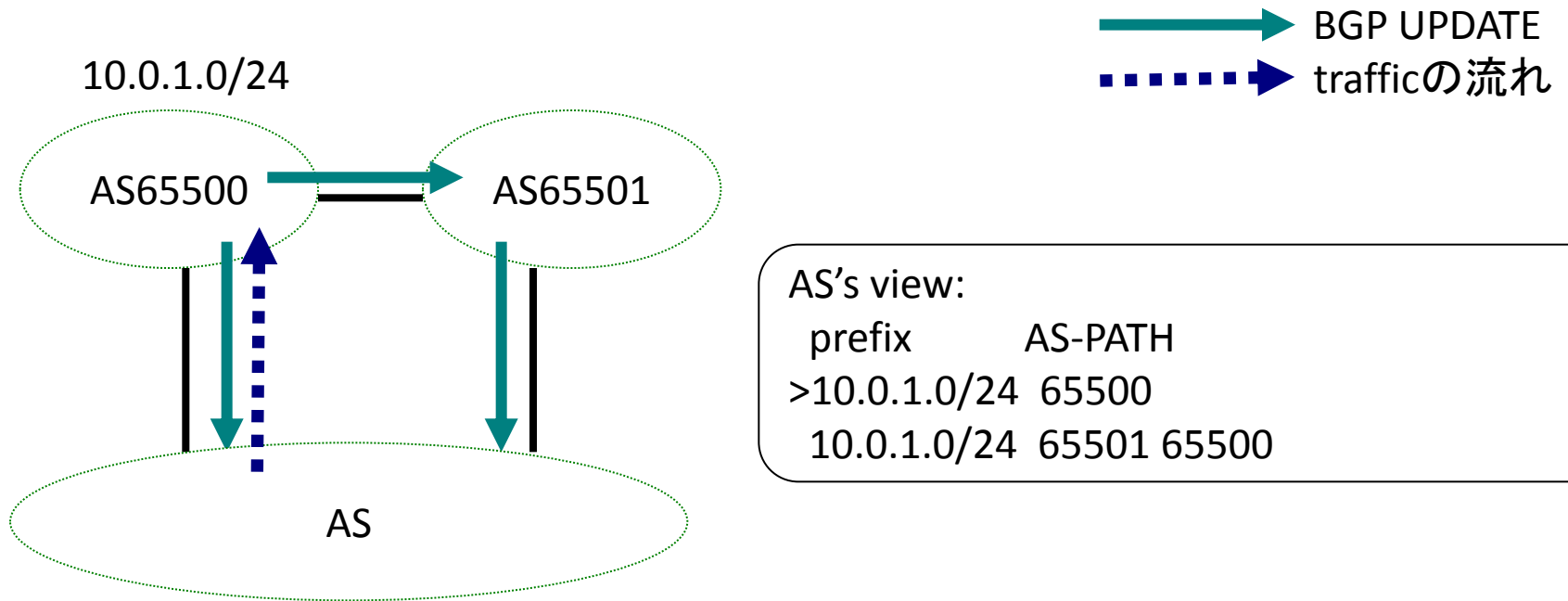
- Local Preference
  - 受信時に設定する
- AS Path
  - 相手ASから広報される
- MED
  - 相手ASから設定されて広報される、もしくは受信時に上書き設定する
- NEXT\_HOP Cost
  - AS内部のトポロジに依存する

# Local Preference



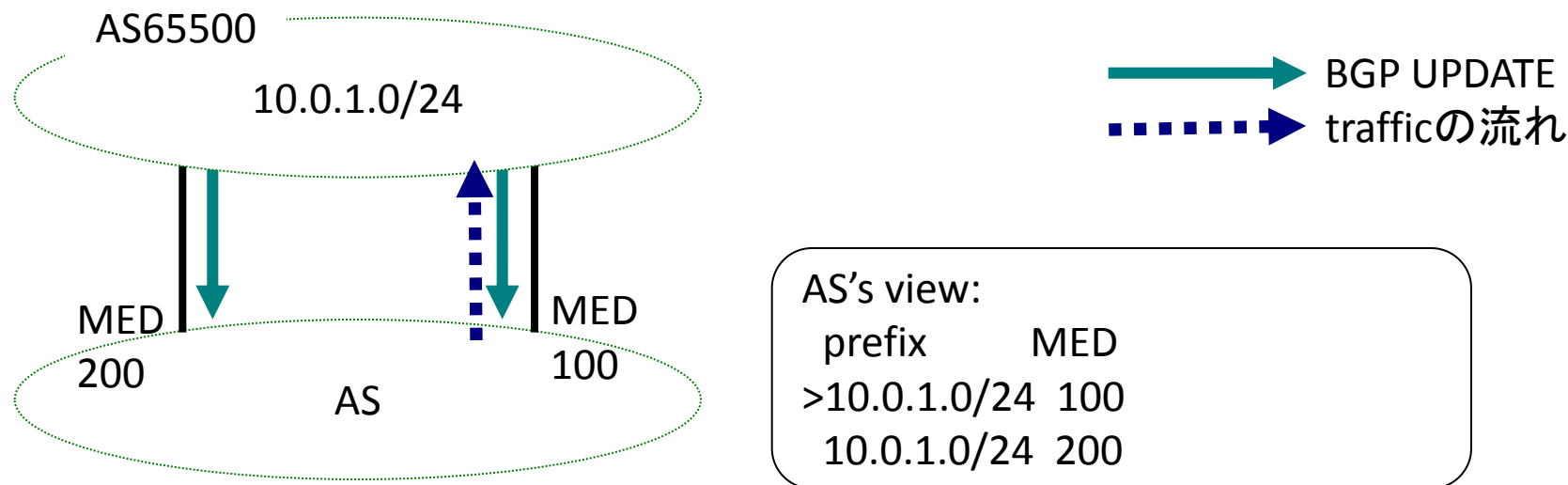
- Local Preferenceの大きな値が優先
- あるAS経由の経路を優先したい場合に有効

# AS Path



- AS Path長が短い経路が優先

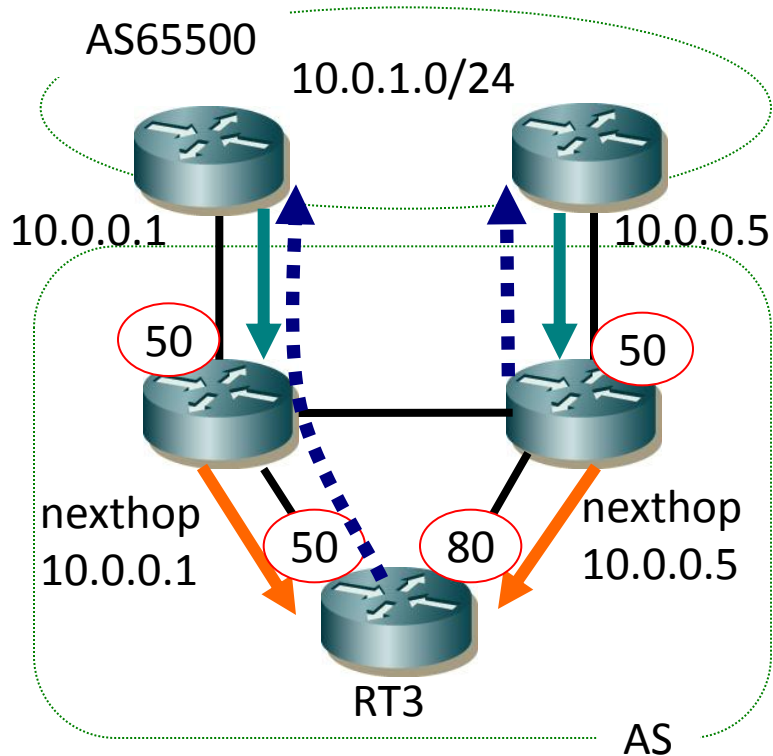
# MED(MULTI\_EXIT\_DISC)



- 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



→ BGP UPDATE  
→ trafficの流れ

- 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]

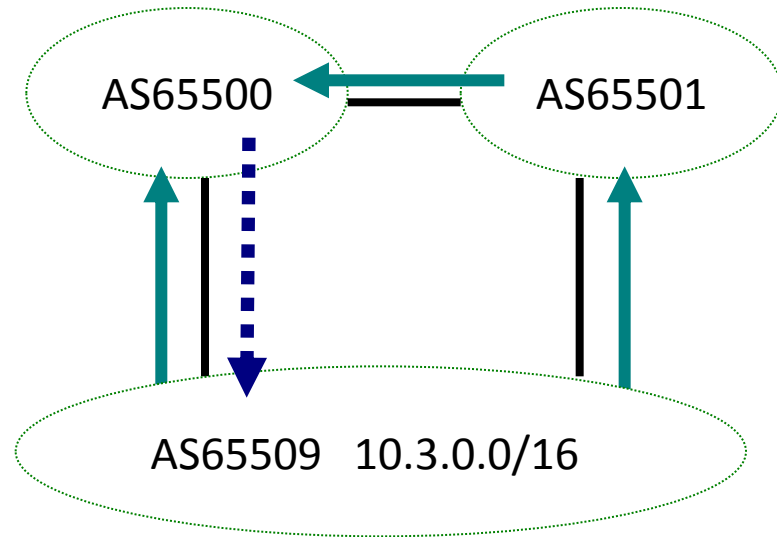
# 他ASへの広報で重要な属性値

- 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
- 相手とのポリシーのすり合わせが重要

# AS Path (広報時)

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

→ BGP UPDATE  
→ trafficの流れ

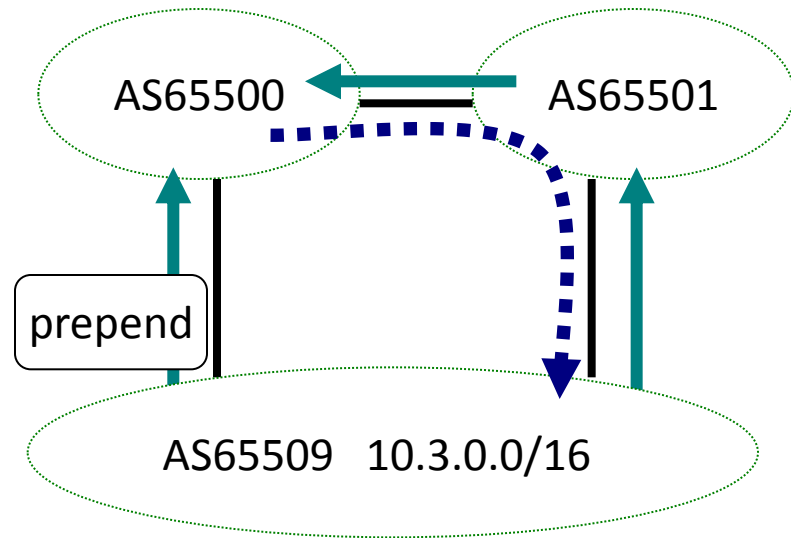


- AS Path長が短い経路が優先

# AS Path prepend

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

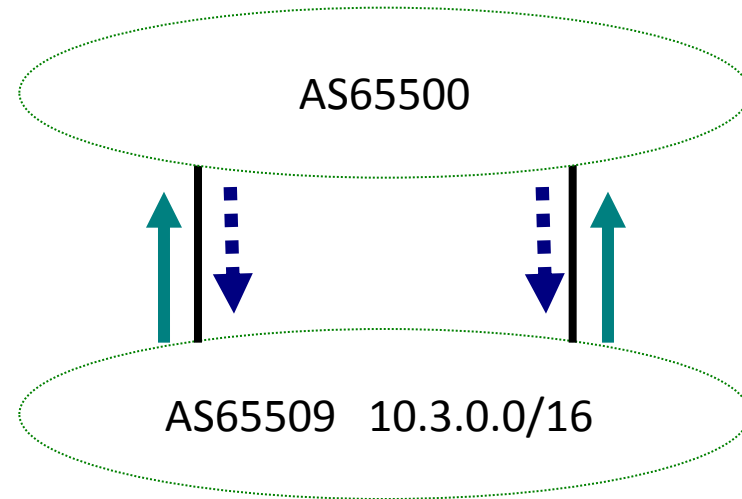
→ BGP UPDATE  
- - - - - trafficの流れ



- 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

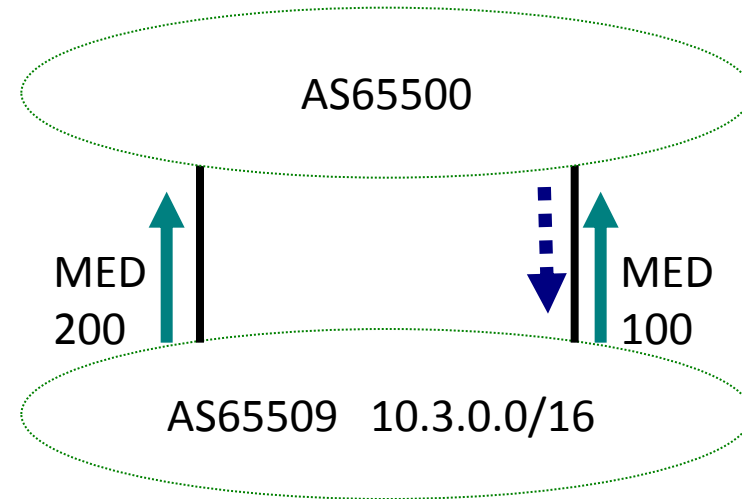


—————▶ BGP UPDATE  
- - - - -▶ trafficの流れ

- AS65500で特別な制御を行っていないならば、closest exitになるはず
  - トラフィックの分散は相手ASの構成に依存する

# MED (広報時)

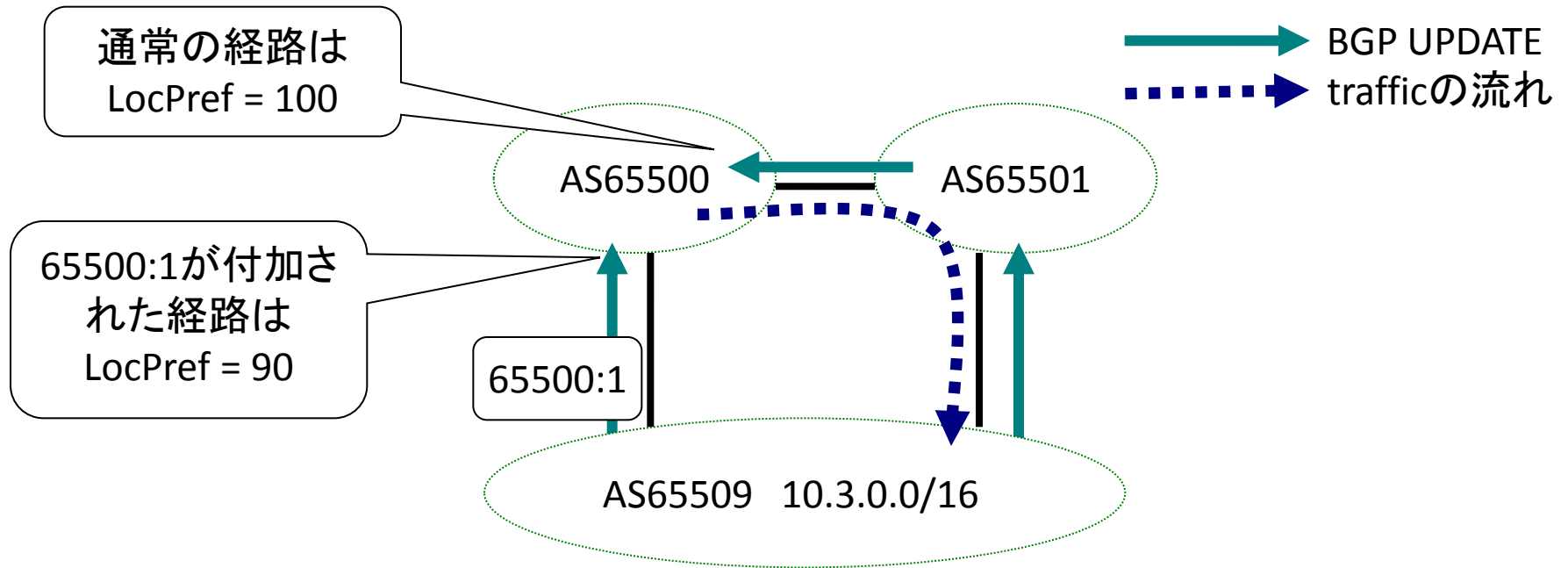
prefix	MED	AS-PATH
>10.3.0.0/16	100	65509
10.3.0.0/16	200	65509



—————▶ 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のポリシー依存

# Community利用例



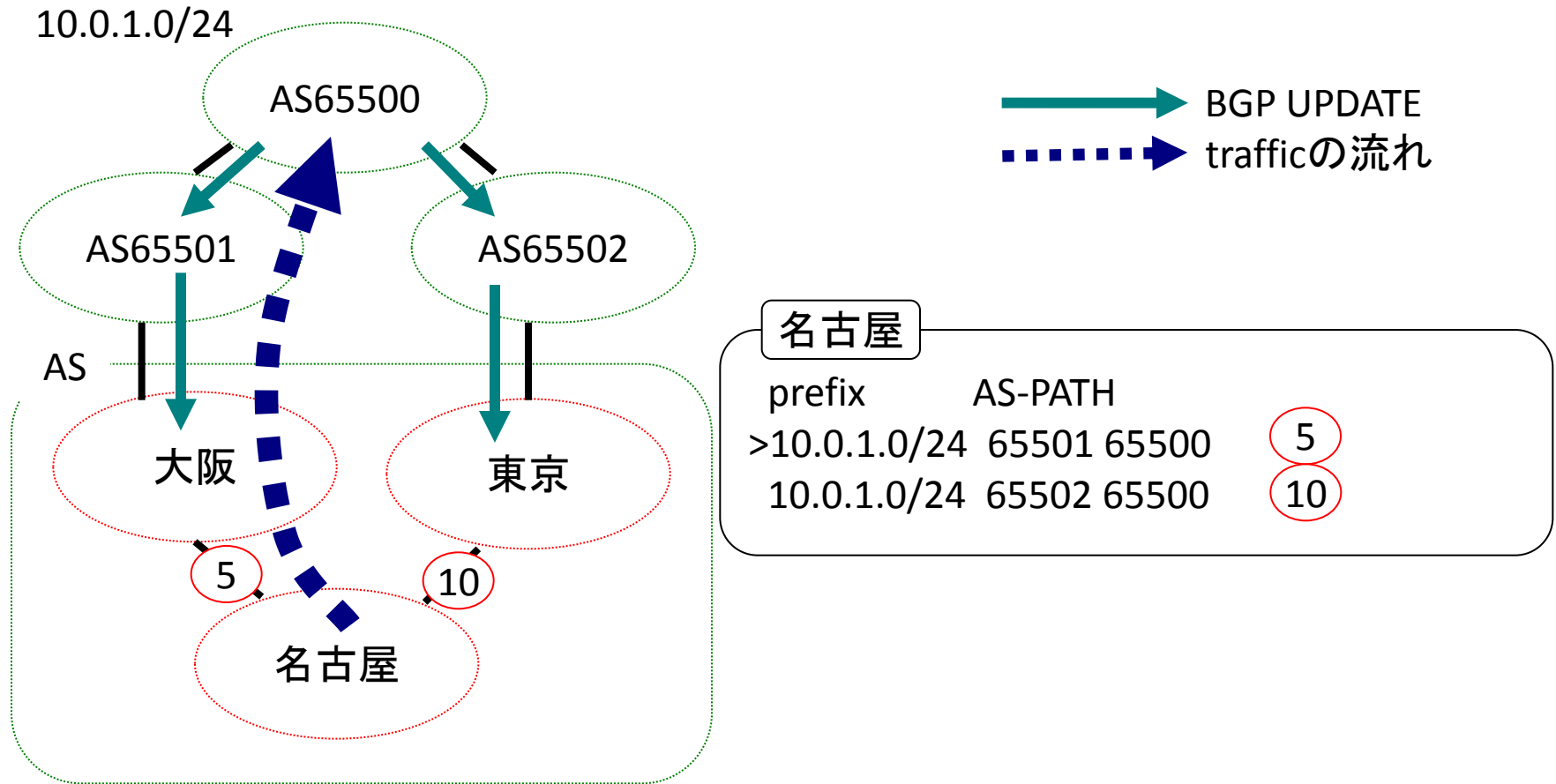
- AS65500がCommunity制御を実装していれば利用できる
- 経路にCommunity情報を付加して、その制御を利用する
- Communityを受け付けるかどうかはASのポリシー依存

# BGPのパス選択

OSPFとBGPの関わりなどを  
解説する



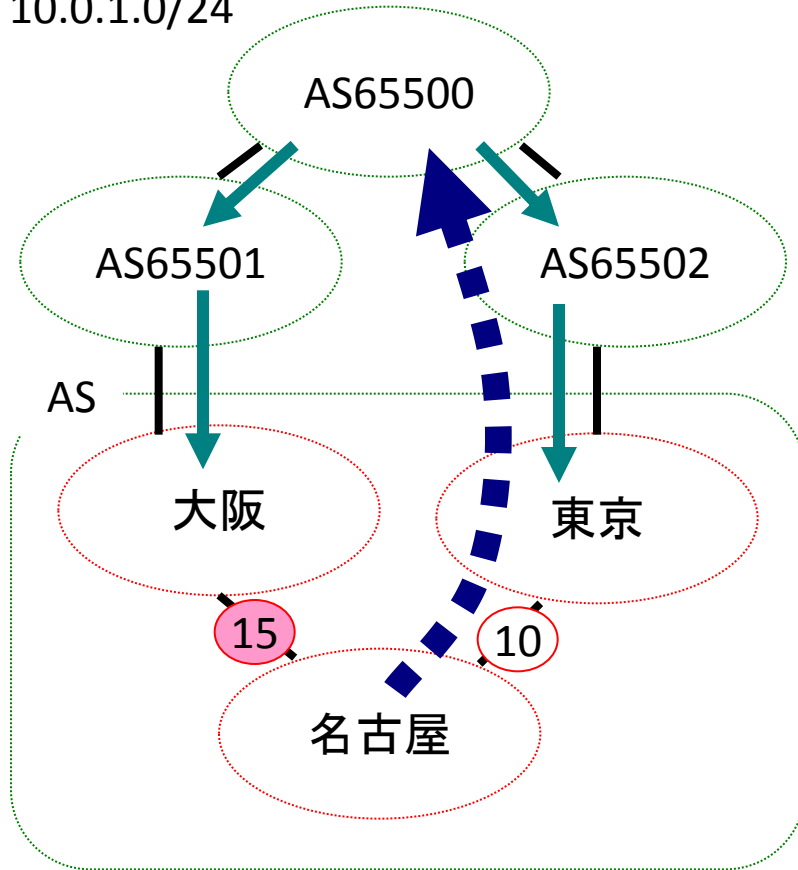
# closest exitとBGP



- 名古屋では、65501(大阪)経路を選択中

# OSPFのコスト変更

10.0.1.0/24



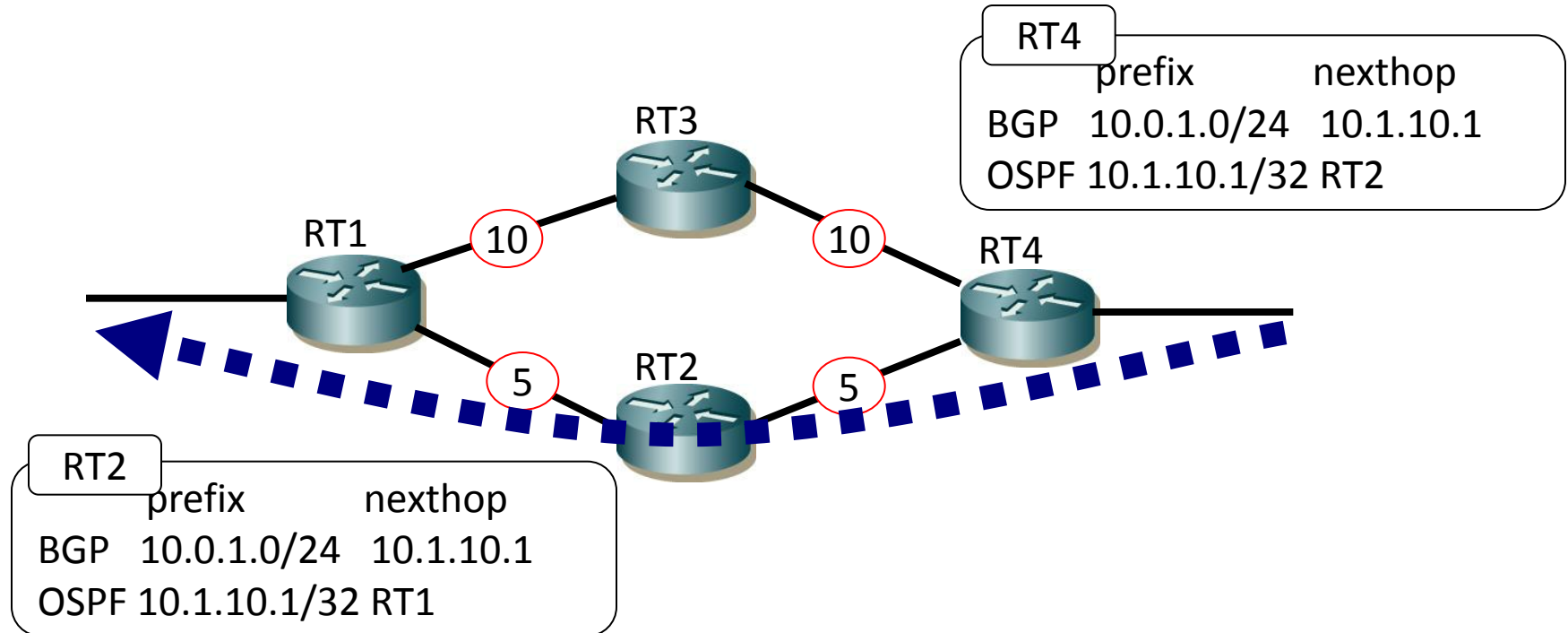
—————> BGP UPDATE  
- - - - -> trafficの流れ

名古屋

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

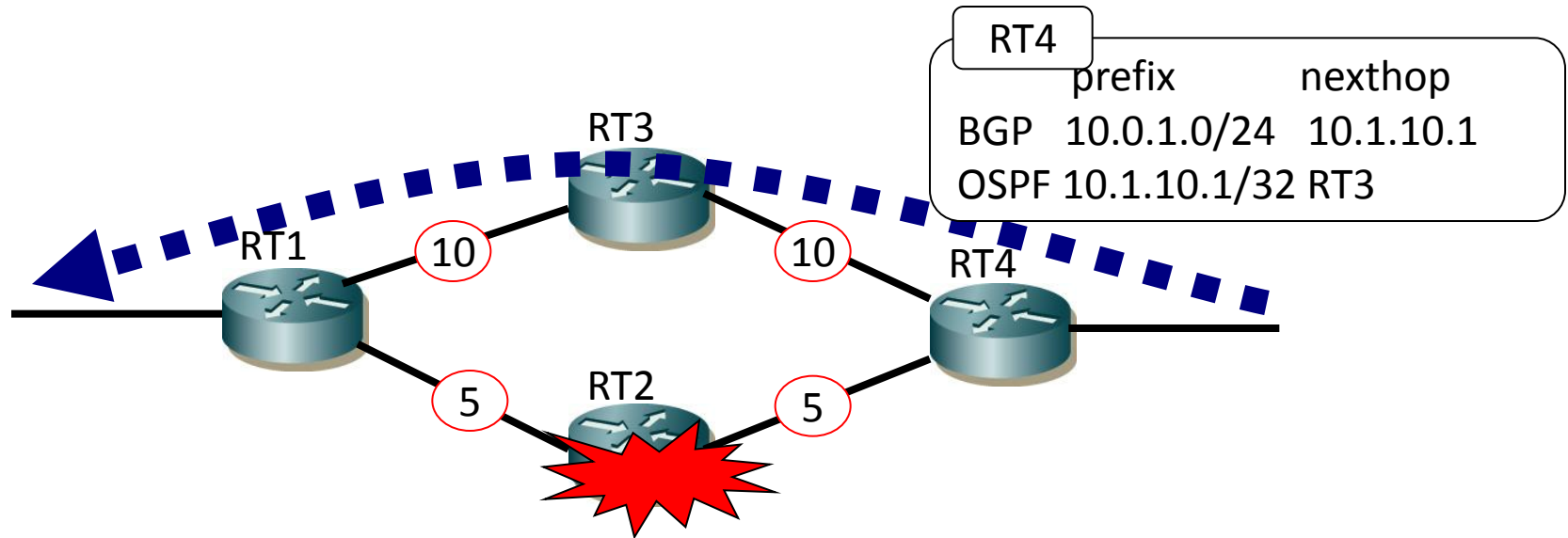
- 名古屋からは65502(東京経由)に更新

# OSPFコストとBGP



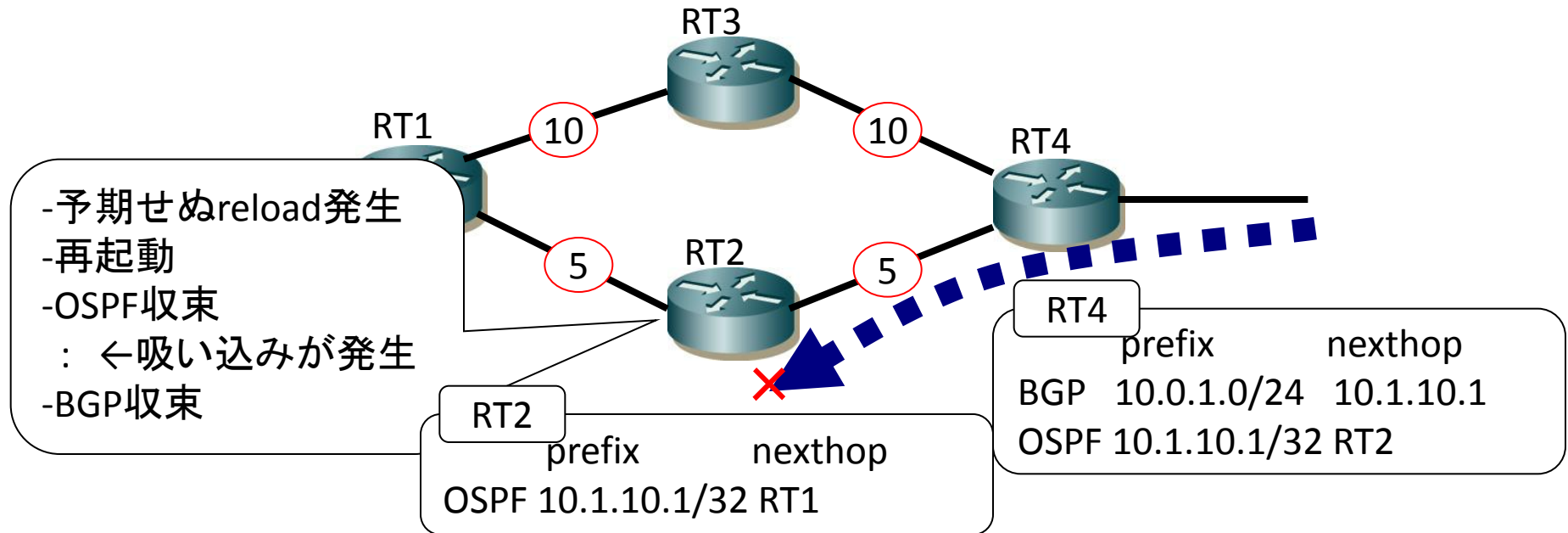
- BGPネクストホップへのOSPFコストが一番小さな経路が選択される

# RT2が突然reload



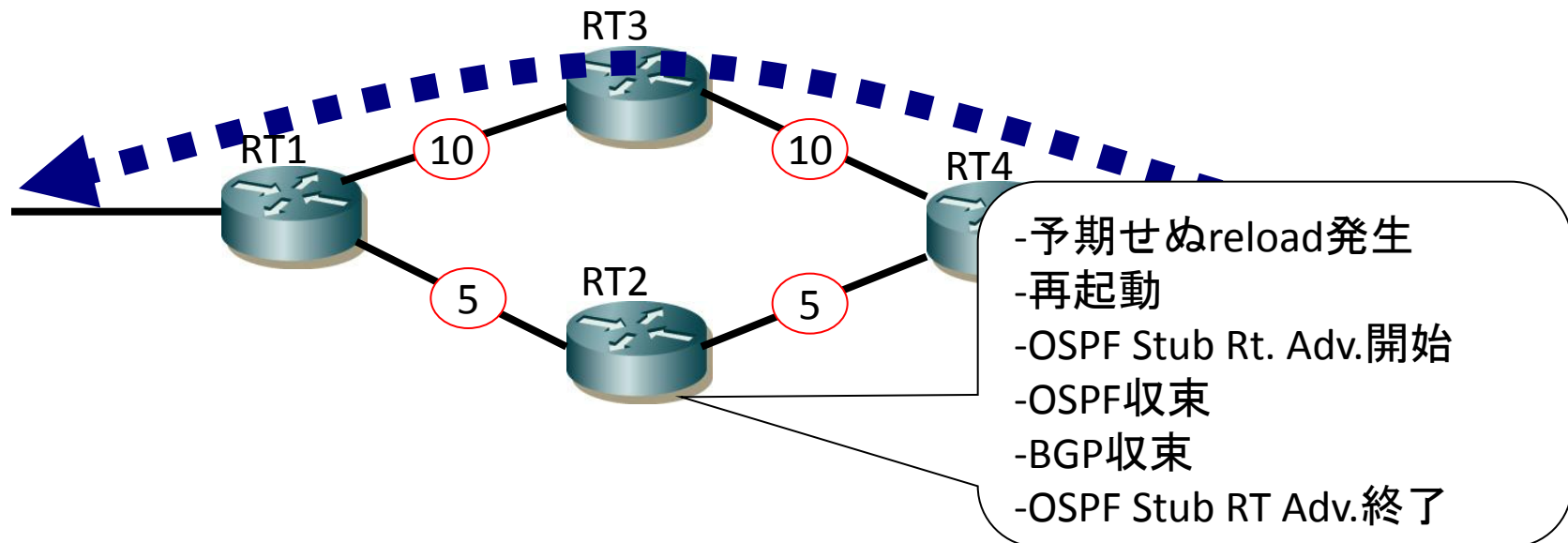
- RT2が再起動・・・
- 他のルータが障害を検出し、OSPF再計算
- トラフィックはRT3を迂回している

# OSPFとBGPの収束時間が違う



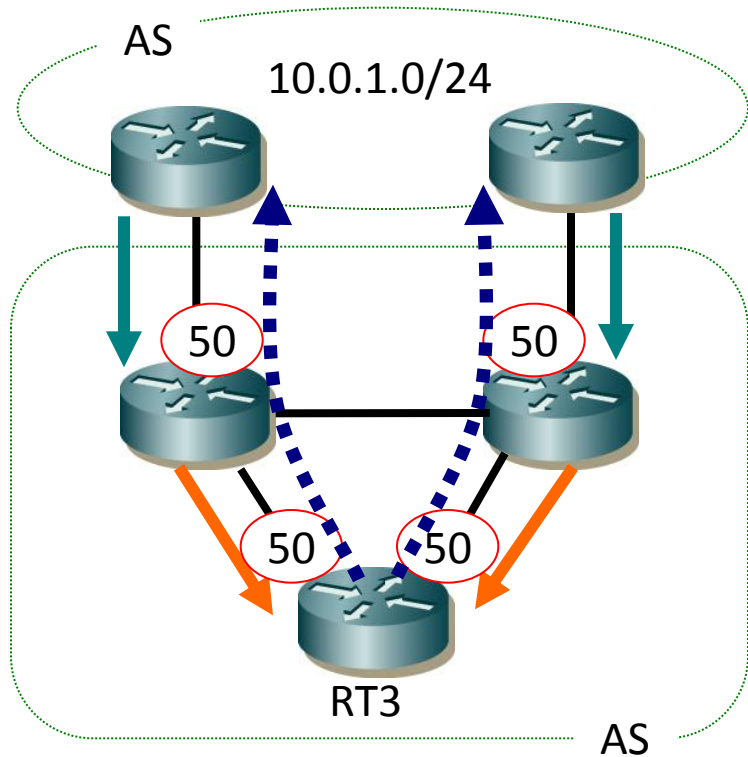
- OSPFは収束したので、RT4ではRT2側を選択
- RT2はまだBGP経路を受信しきっていない
- その間、RT2がトラフィックを破棄してしまう

# OSPF StubRouterAdvertisement



- ルータを経由するトラフィックを迂回させる機能
- OSPF起動後に実施して、BGP収束までトラフィ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

おわり



# 参考資料

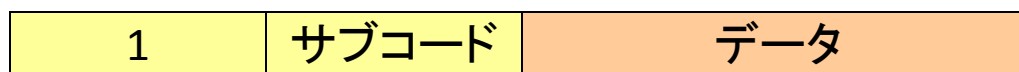
- BGP NOTIFICATIONメッセージコード
- BGP パス属性コードタイプ

# 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

# 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	データに含まれる値
1.	Markerの値が不正	
2.	Lengthの値が不正	不正だと判断したLengthの値
3.	解釈できないタイプ	解釈できなかったタイプの値

# 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

2	サブコード	データ
---	-------	-----

サブコード	エラー内容	データに含まれる値
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パラメータがある	
5.	[Deprecated]	
6.	ホールドタイム値が受け入れられない	
7.	未サポートのCapability	サポートしていない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

3	サブコード	データ
---	-------	-----

サブコード	エラー内容	データに含まれる値
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(TLV)
3.	あるべき周知必須属性が無かった	無かった周知必須属性のタイプコード
4.	フラグが不正	エラーを検出した属性(TLV)
5.	パス属性長が不正	エラーを検出した属性(TLV)
6.	ORIGIN属性値が未規定の値	エラーを検出した属性(TLV)
7.	[Deprecated]	
8.	NEXT_HOP属性値の書式が不正	エラーを検出した属性(TLV)
9.	オプション属性値でエラー	エラーを検出した属性(TLV)
10.	NLRIの書式が不正	
11.	AS_PATH属性の書式が不正	

# コード4 HoldTimer超過

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

4	サブコード	データ
---	-------	-----



# コード5 状態遷移エラー

- 予期せぬイベントが発生

5	サブコード	データ
---	-------	-----

# コード6 Cease

- その他のエラーを検出

6	サブコード	データ
---	-------	-----

サブコード	エラー内容	データに含まれる値
1.	最大受信経路数に到達	<AFI(2), SAFI(1), prefix上限値(4)>
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	

# BGPパス属性値コードタイプ

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	経路の宛先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	経路の優先度
6	ATOMIC_AGGREGATE	周知任意	経路が途中で集約された情報
7	AGGREGATOR	オプション通知	経路集約を行ったルータ
8	COMMUNITIES	オプション通知	処理を行うための情報

# BGPパス属性値コードタイプ 続き

---

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

---