

# ISPバックボーンネットワーク における経路制御設計 ～ 理論編 ～

---

Internet Week 2004 チュートリアル

前村 昌紀

フランステレコム日本研究所

maem@francetelecom.co.jp



1

## 目次

---

- ルーティング基礎事項の整理(15分)
  - ルーティングとは, RIPのおさらい, クラスレス
- OSPF – Open Shortest Path First(75分)
  - プロセス確立, LSDB, LSA,
  - 方路選択と耐障害性, IS-ISとの比較
- BGP – Border Gateway Protocol(75分)
  - ASと階層的経路制御, パケットタイプ,
  - パス属性値, ポリシルーティング
  - iBGPシステム的设计, スケーラビリティなど



2

# ルーティング基礎事項の整理

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～

# セクション目次

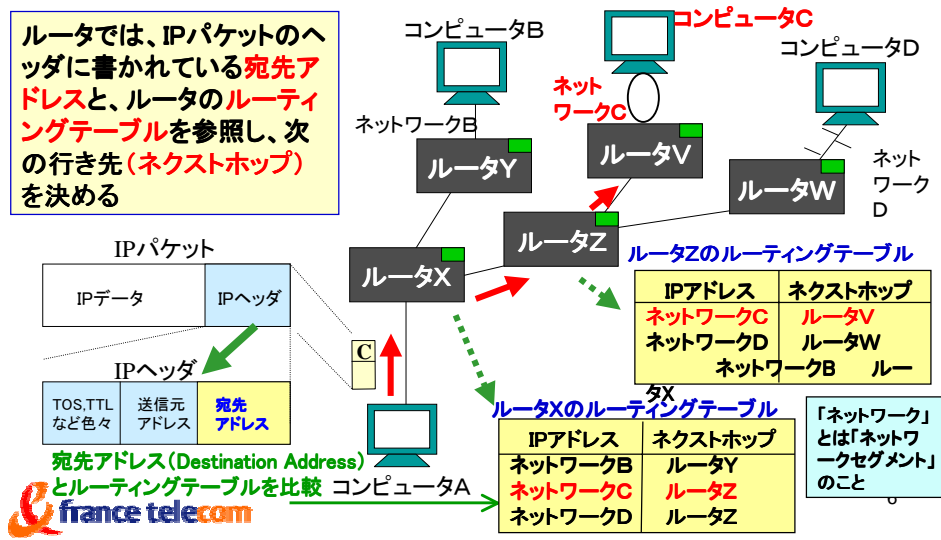
- ルーティング基礎事項の整理
  - ルーティングとは何か, 原型としてのRIP
  - クラスフルとクラスレス
  - IGP,EGP,ASとアルゴリズム

# ルーティングとは何か 原型としてのRIP

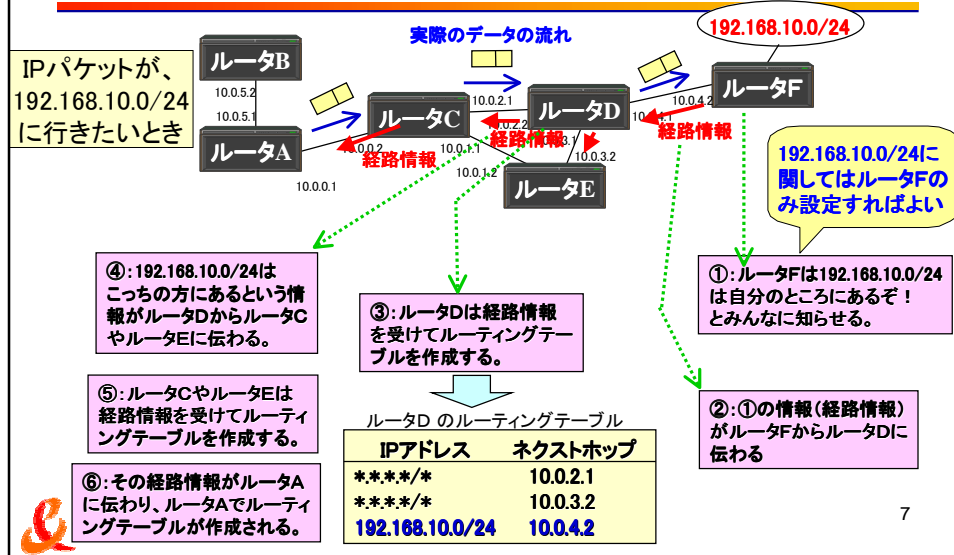
ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
ルーティング基礎事項の整理

## ルータとルーティング

ルータでは、IPパケットのヘッダに書かれている宛先アドレスと、ルータのルーティングテーブルを参照し、次の行き先(ネクストホップ)を決める



# ダイナミックルーティング



## RIP – Routing Information Protocol , リップ

- 原始的で簡素なルーティングプロトコル
  - 古くからUNIXシステム上で実装されている(routed)。
  - UDP 520番ポートを利用(セッションを確立しない)
  - 自分のもっているルーティングテーブルを接続しているネットワークに**30秒ごと**にブロードキャストする
  - **ネットマスクの情報を運ばない**
  - 隣接したルータから受け取った情報(ネットワークアドレス)に自分の知っている情報を付加し送信する
  - これが全ルータの間で繰り返し行われることでルータは接続されたすべてのネットワークとそこへの道筋を知る。



## RIPのメリットとデメリット

---

- メリット
  - 処理の負荷が小さい(軽い)・実装が簡単
  - **どんなルータでも利用可能, ホストも対応可能**
- デメリット
  - 30秒に一度全経路情報を伝える
    - 経路数が多くなると無駄が多い
  - ネットマスクの情報を運ばない
    - クラスフルなルーティングプロトコル
  - 収束に時間がかかる
  - 最大のホップ数は15までしか対応できない
  - ホップ数での比較なので、回線の帯域に応じて適切な経路を選ぶことが難しい

## RIP2 – RIPの改良版

---

- ネットマスク長情報を運ぶように改良
  - クラスレスプロトコルとなった
- しかし、RIP1と同様、30秒に一度全経路情報を伝えようとする
- 実装が少ない – RIP最大のメリットが消失

結局あまり普及せず今にいたる

## クラスフルとクラスレス

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
ルーティング基礎事項の整理

## Class A, B, C とは － クラスフルアドレス －

- IPv4の初期設計において、上位3ビットによってIPアドレスのどこまでがネットワークを示し、どこまでがネットワーク中のホストを示すか、自動的に分かるようにした。

4bits	Octet#1	Class	ネットワークアドレス	ホスト数
0???	0—127	A	第一オクテットだけ	16.8M
10??	128—191	B	第二オクテットまで	65534
110?	192—223	C	第三オクテットまで	254

## クラスフルなルーティングと ルーティングプロトコル

- プロトコルにネットマスクを扱う機能がない
- Class A, B, Cに従ってネットワークアドレスの認識を行う
- その中を更に分割したものをサブネットと言う
  - プロトコルを通じてネットマスクを伝達できない
  - サブネッティングは、自身のインターフェースに定義したものを参照して解釈する
    - クラスフルネットワークの中は統一したサブネットのサイズにしな  
いと扱えない
- 自身が属するネットワークアドレス以外は、サブネットとして認識できない
  - サブネットが2方向以上に散らばっていると経路制御不能



## クラスレスルーティング

- プロトコルがネットマスクの情報も扱う
  - ネットワークを示すもの==プリフィクス(Prefix)
  - プリフィクスの長さは一般的にビット数で表される
    - Class Cの 202.216.40.0 – 202.216.40/24  
(202.216.40.0/24)
- つまりクラスレスだと、
  - 連続するclass Cアドレスを任意の大きさとひとかたまりで扱える
  - Class Aのサブネットも全く同様に扱える
  - Class Cより小さいアドレスブロックも全く同様に、任意の大きさと扱える
    - これがいわゆるVLSM(Variable Length Subnet Mask)





# IGP, EGP, AS アルゴリズム

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
ルーティング基礎事項の整理

# IGP, EGP, AS

- AS – Autonomous System = 自律システム
  - (基本的には)ISPを1つの単位としてASを定める
  - ASの中外で別々の独立したルーティングを行う
- IGP – Interior Gateway Protocol
  - 内部(ゲートウェイ)プロトコル
  - 個別のネットワークセグメントまでに至る詳細なルーティングを実施する
  - RIP, OSPF, IS-IS, IGRP, EIGRP,,,
- EGP – Exterior Gateway Protocol
  - 外部(ゲートウェイ)プロトコル
  - ASを単位としてAS間の経路制御を行う
  - 実質的にBGPだけ。原型にEGP, 他にIDRP

## アルゴリズムの種類

---

- それぞれのプロトコルのアルゴリズムの特徴を表現したもの
- ディスタンスベクター – distance vector
  - RIPなど。「距離」と「方向」を扱う、という意味
- パスベクター – path vector
  - BGP。「パス属性」と「方向」を扱う、という意味
- リンクステート – link state
  - 管理領域内の全てのネットワークセグメント(リンク)の状態情報を収集して管理する方式

## OSPF – Open Shortest Path First

---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～

# セクション目次

- OSPF – Open Shortest Path First
  - OSPFをスライド7枚でとりあえず説明する
  - OSPFのプロセスを確立する
  - リンクステート広告(その1)
  - リンクステートデータベース
  - リンクステート広告(その2)
  - OSPFの方路選択と耐障害性
  - 基本的な適用技術
  - IS-ISとの比較

# 目次を別の観点から整理する

## • OSPFパケットの種類

- Hello
- Database Description
- Link-State Request
- Link-State Update
- Link-State Acknowledgment



機能, 運用

経路情報の生成

データベース構築

## • LSAの種類

- Router-LSA(Type1)
- Network-LSA(Type2)
- Summary-LSA(Type3, Type4)
- AS-external-LSA(Type5)
- NSSA External LSAs(Type7)

## OSPFをスライド7枚で とりあえず説明する

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

## OSPFのメリット： リンクステート型の精密な動作

- リンクステート型プロトコル
  - 全ルータが全リンク状態をDBで管理して、そこからルーティングテーブルを生成する
- リンクステート型ならではの精密なネットワーク管理
  - バックアップパス, イコールコストマルチパスの形成, 障害発生時の素早い収束

## リンクステート – Link State

---

- リンク(ネットワークセグメント)の状態情報
  - 死活, IPアドレス, マスク, ネットワークタイプ, 接続ルータ, メトリック
  - 「リンクステートアドバタイズメント(LSA)」で伝達
- 管理領域内の全てのルータは、管理領域全域のリンクステートを全て収集し、データベース化して管理する
  - リンクステートデータベース(LSDB)
  - RIPのような原始的なダイナミックルーティングとは全く異なる
- リンクの状態が変化したときにその変更を他のルータに伝える
  - リンクの死活や接続関係(トポロジ)の変化
  - 実は、変化のないときでも定期的に30分に1回リフレッシュを実施

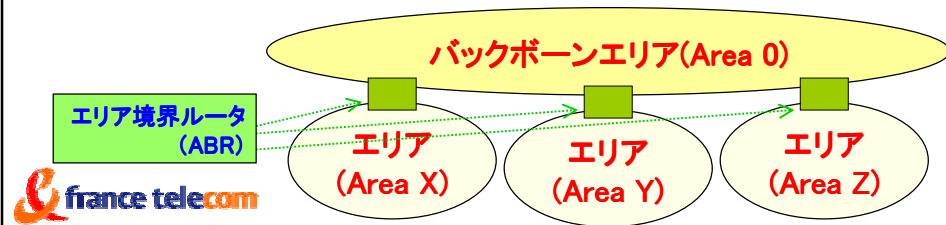
## OSPFのネットワーク設計

---

- 管理するネットワークをエリアに分割して、エリア外の管理を簡素化
  - クラスレス対応, エリアごとの経路集約が可能
- DR(指名ルータ), BDR(バックアップ指名ルータ)を利用した、マルチプルアクセスネットワークにおけるプロトコル負荷の軽減
  - ネットワーク設計上の考慮が必要

# エリア

- エリア
  - OSPF管理領域をいくつかのエリアに分割する
  - エリアIDは32ビットIPv4アドレスを割り当てることが可能
  - バックボーンエリア(area0)に他のエリアがぶら下がる形
  - LSDBはエリア内の全ルータで共通,エリア外のトポロジは管理せず、サマリのみを管理
- エリア境界ルータ(ABR: Area Border Router)
  - バックボーンエリア(エリア0)と他のエリアをつなぐルータ



# OSPFその他の特徴

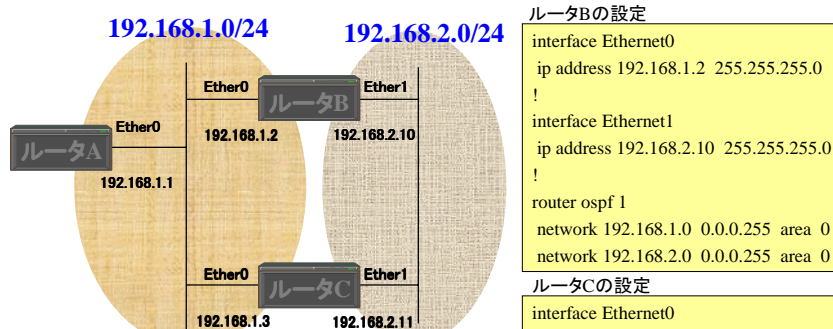
- 精密なルーティングを目指して、
  - OSIのIS-ISに見習って開発されたルーティングプロトコル
  - 原型としてSPFプロトコル
- IPマルチキャストを利用
- IPの上に直接実装されたプロトコル。
  - プロトコル番号89番
- DR(指名ルータ), BDR(バックアップ指名ルータ)を利用したマルチプルアクセスネットワークにおけるやり取りの簡素化

## OSPF設定の一例(ciscoの場合)

- router ospf <process ID>
  - 自分のASと同じ番号にすることが多い
    - 一つのAS内で一つしかOSPF processを走らせない場合
    - process IDは1~65535の何番にしてもいい
- network 192.168.0.0 0.0.0.15 area 0
  - 0.0.0.15はワイルドカードマスク
    - アドレスのうち無視する部分をマスクする
  - 192.168.0.0~192.168.0.15の範囲にあるアドレスのインタフェースで
    - OSPFを area 0 で話す
    - そのインタフェースのネットワークをOSPFに広告する

上記2つが基本で、最低限のOSPFのconfig

## OSPF設定の一例(ciscoの場合)



ルータAの設定

```
interface Ethernet0
ip address 192.168.1.1 255.255.255.0
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
```

ルータBの設定

```
interface Ethernet0
ip address 192.168.1.2 255.255.255.0
!
interface Ethernet1
ip address 192.168.2.10 255.255.255.0
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
```

ルータCの設定

```
interface Ethernet0
ip address 192.168.1.3 255.255.255.0
!
interface Ethernet1
ip address 192.168.2.11 255.255.255.0
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
```

## OSPFプロセスを確立する

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

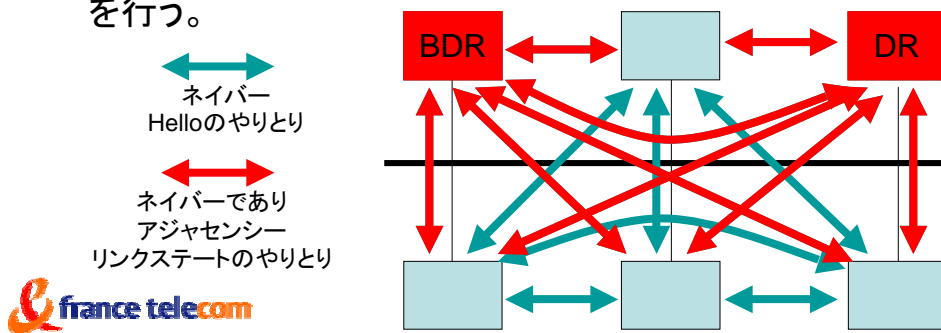
## パケット種別

	パケットタイプ	内容
1	Hello	ネイバー確立と維持
2	Database Description	アジャセンシー確立時のDB内容の伝達
3	Link-State Request	リンクステート情報の要求
4	Link-State Update	リンクステート情報の伝達・更新
5	Link-State Acknowledgement	Link-State Updateの受領確認



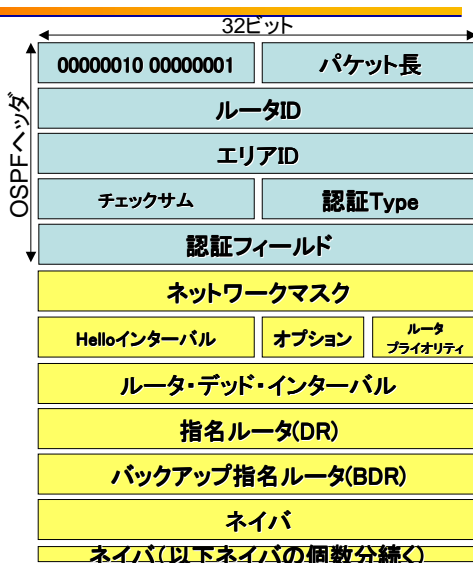
## アジャセンシー, DR, BDR

- アジャセンシー(adjacency:隣接) ==リンクステートのやり取りを行う関係
- マルチプルアクセスネットワークでは指名ルータ (designated router: DR), バックアップDR(BDR)が代表して、他のルータとのリンクステートのやり取りを行う。



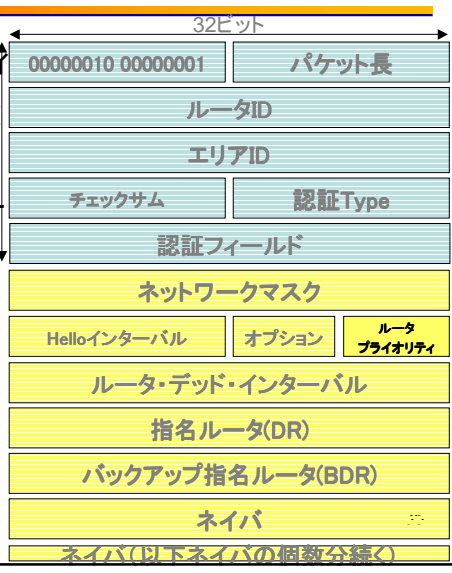
## HELLOパケット

- マルチキャストでネットワークセグメント上に発信される
  - Ackなどは返らない
- ルータIDで自分を名乗り、それまでに知っているDR,BDR, ネイバを示す
  - ネイバ同士は、相手からのHELLOパケットのネイバフィールドに自分のルータIDがあることで、ステータスを認識する
- DR, BDRはルータプライオリティの比較, なければルータIDの大きいほう。但し後発によって置き換わらない



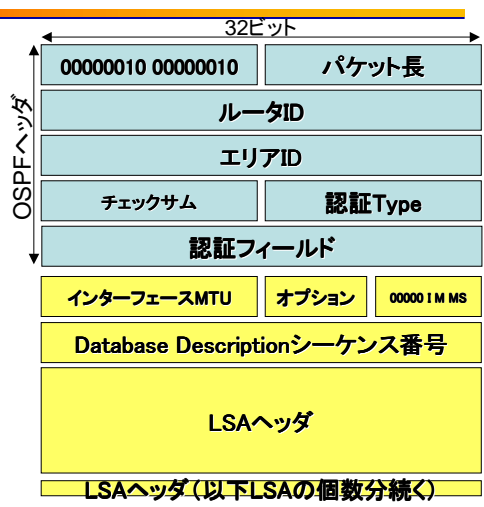
## ルータプライオリティ

- DR, BDRはルータプライオリティの比較, なければルータIDの大きいほう。但し後発によって置き換わらない
- Ip ospf priority をインターフェースで定義 (ciscoの場合)
- Priority 0 で、決してDR,BDRにならない
  - 高負荷なルータに設定すると有効



## Database Descriptionパケット

- アジャセンシー初期化の時に交換
- LSAヘッダを全て交換し、新たに送出, 受領するべきLSAを知る

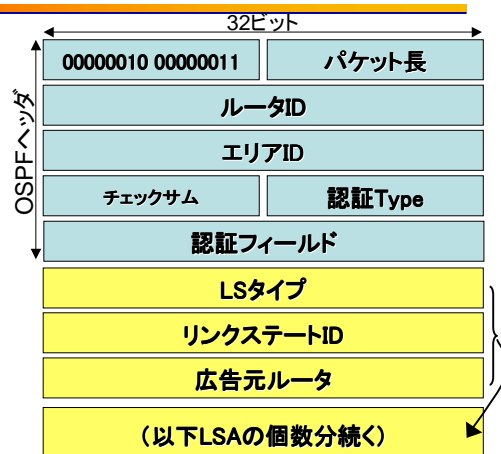


## アジャセンシー確立までの道

- HELLOパケットでアジャセンシーであることが決定
- Database Descriptionパケットでリンクステートデータベースの内容比較を行う
- 相手が持っていて自分が持っていない(もしくは古い)リンクステートに関して、リンクステート広告を要求(リンクステートリクエスト)
- 全リンクステートの交換完了
- 完了

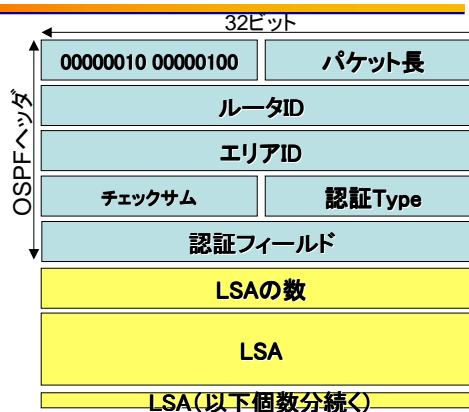
## Link-State Requestパケット

- 最新のLSAを送信するように要求する



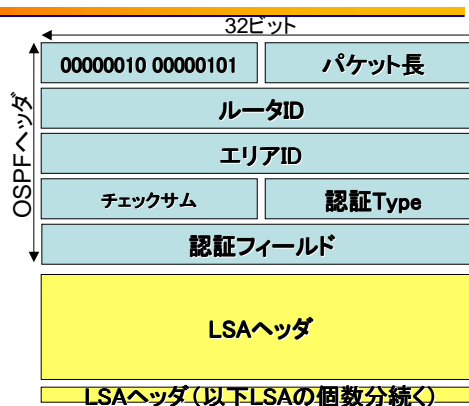
## Link-State Updateパケット

- LSA(Link-State広告)を複数送信することができる



## Link-State Acknowledgement パケット

- LSAヘッダのみを送って受領確認とする



# リンクステート広告(LSA) その1

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

# リンクステート広告(LSA)

- リンクステート広告(LSA: Link State Advertisement)
  - OSPFにおいて経路情報生成のために交換される本質的  
情報
  - Link-State Update(OSPFパケットタイプ#4)パケットの中に  
複数収容される

# LSAヘッダ

- LSAに関する各種情報が格納されている
- LSAの要約として、Database DescriptionやLink-State Acknowledgementで用いられる

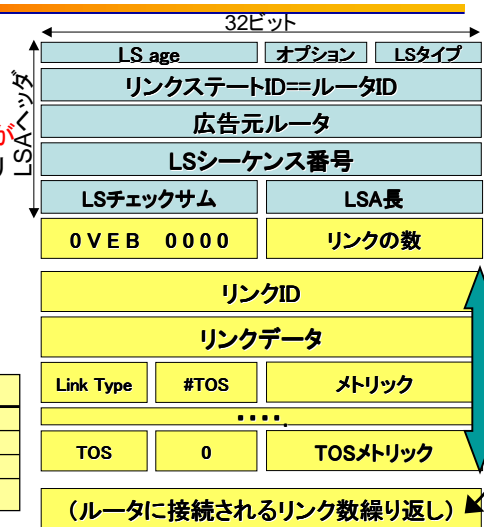


# ルータLSA (LSAタイプ1)

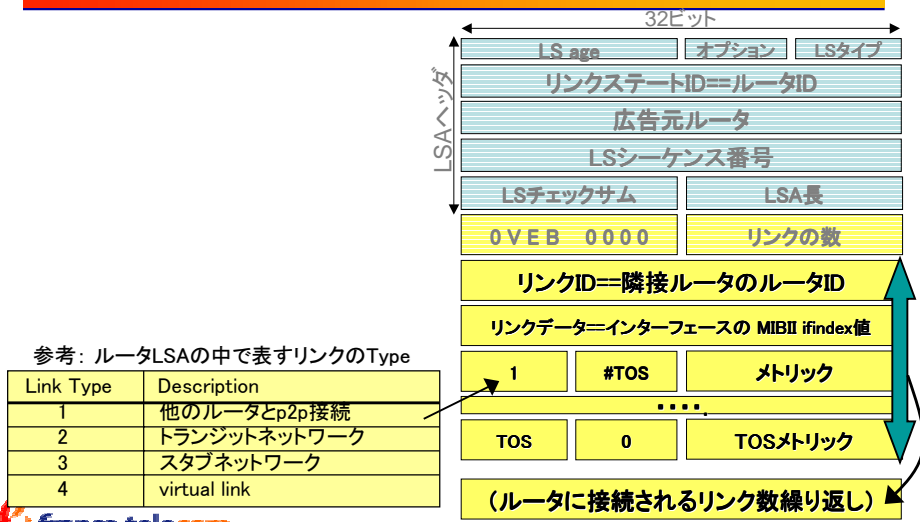
- 全てのルータで生成する
- ルータの接続情報
  - そのルータにどういふリンクがついているか、それぞれのリンクの種類\*とリンクの情報 (Link ID, Link DATA)とコストを情報としてもつ
- エリア内しか伝わらない

参考: ルータLSAの中で表すリンクのType

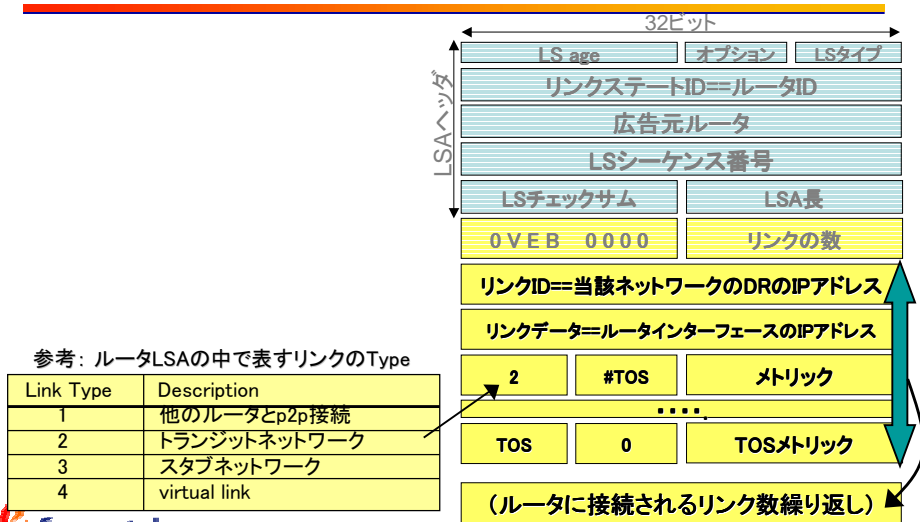
Link Type	Description
1	他のルータとp2p接続
2	トランジットネットワーク
3	スタブネットワーク
4	virtual link



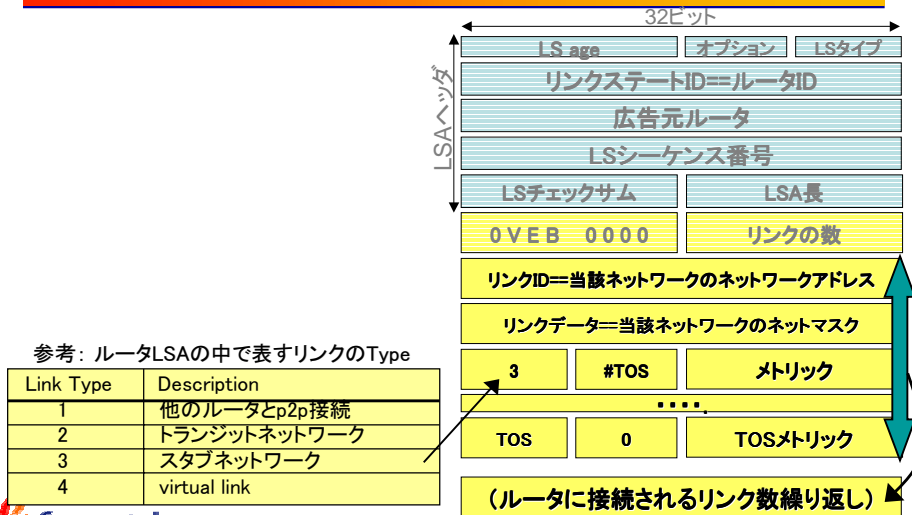
# ルータLSA(LSAタイプ1) リンクタイプ1 – p2p接続



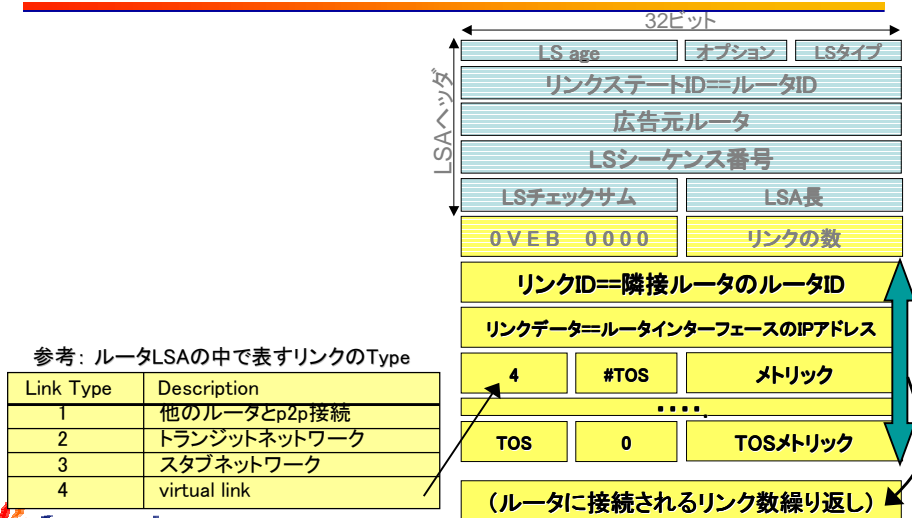
# ルータLSA(LSAタイプ1) リンクタイプ2 – トランジットネットワーク



## ルータLSA (LSAタイプ1) リンクタイプ3 – スタブネットワーク



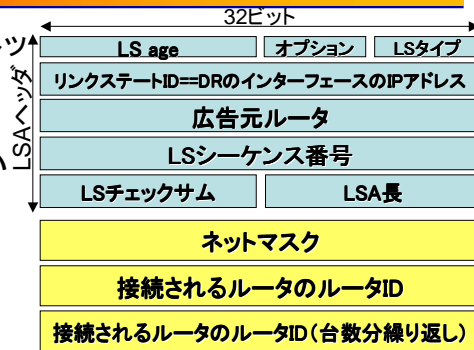
## ルータLSA (LSAタイプ1) リンクタイプ4 – バーチャルリンク





## ネットワークLSA(LSAタイプ2)

- ネットワークのDRがそのネットワークを代表して生成する。
- ネットマスクと接続されているルータのルータIDリスト



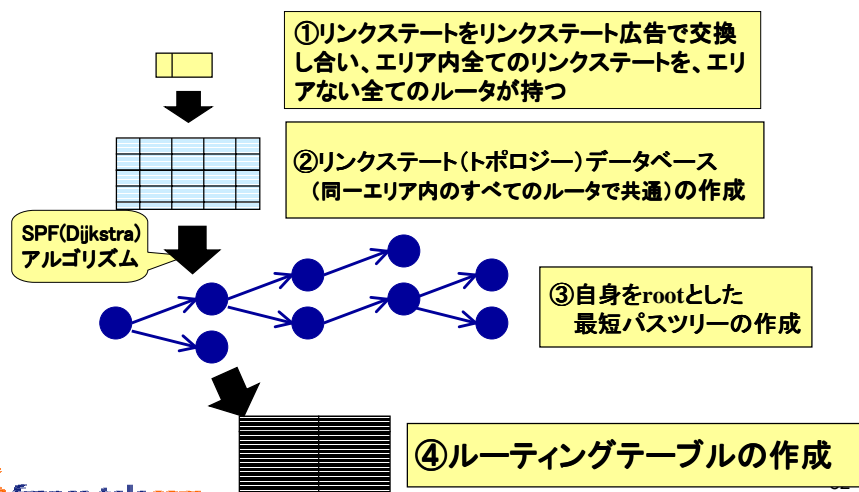
## すごく大雑把な捉え方

- ルータLSA
  - ルータにつながっているネットワークとコストの情報
- ネットワークLSA
  - ネットワークにつながっているルータのリスト
- この二つを全て連結するとトポロジが管理できる

# リンクステートデータベース

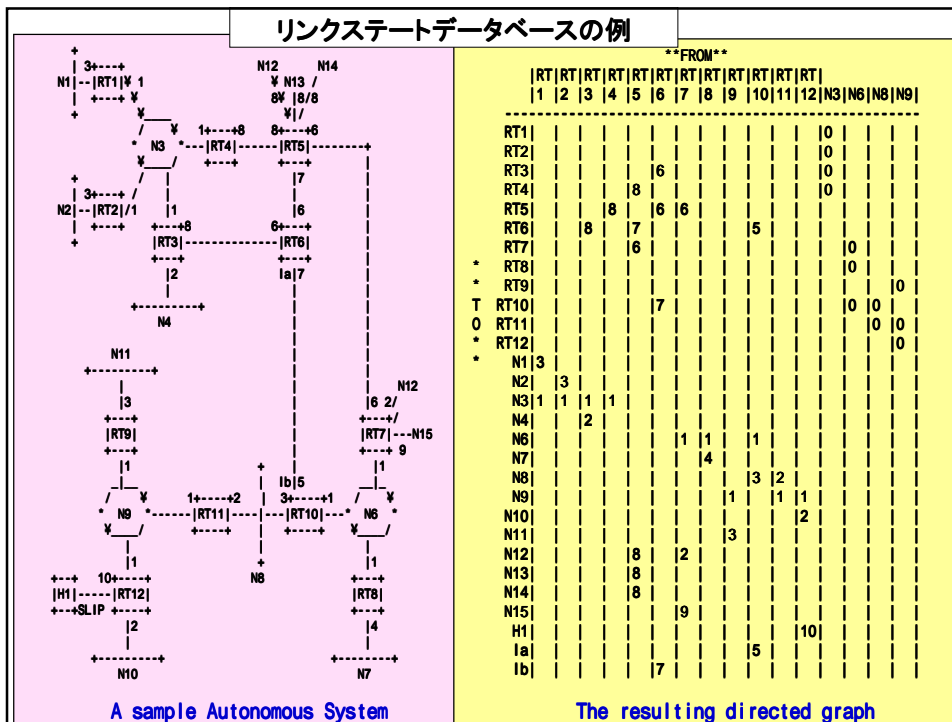
ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

## OSPFにおける ルーティングテーブルの生成



# リンクステートデータベース

- ルータとネットワークで構成される有向グラフ
  - ルータがネットワークにインタフェースを持っているときは、ルータとネットワークをつなぐ
  - 2つのルータが物理的にpoint-to-pointで結ばれているときは、ルータ同士をつなぐ
    - Numberedな/30を持つ場合、双方のルータにぶら下がるネットワークとして解釈される.
  - データベースの中はコストを値とする
    - ルータから見てネットワークに対してコスト値がつく
    - ネットワークからルータに向かうところは常にコスト0



# LSDBの内容

p2pはRT同士の辺となる

- FROMでNWがあるところは複数のルータがあるマルチアクセスネットワークとなる
- NWからRTIに向かうのは常に0

- RTからNやHIに向かうのはそのルータがそのネットワークにインタフェースを持つことを意味する
- 値はそのインタフェースでのコストを示す

- RTからIIに向かうのは、P2Pにおいて対抗のルータのインタフェースにアドレスが割り当てられていることを示す

\*\*FROM\*\*

	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	N3	N6	N8	N9
	1	2	3	4	5	6	7	8	9	10	11	12	N3	N6	N8	N9
RT1																10
RT2																10
RT3																10
RT4																10
RT5																10
RT6																10
RT7																10
RT8																10
RT9																10
RT10																10
RT11																10
RT12																10
N1																10
N2																10
N3																10
N4																10
N6																10
N7																10
N8																10
N9																10
N10																10
N11																10
N12																10
N13																10
N14																10
N15																10
HI																10
Ia																10
Ib																10

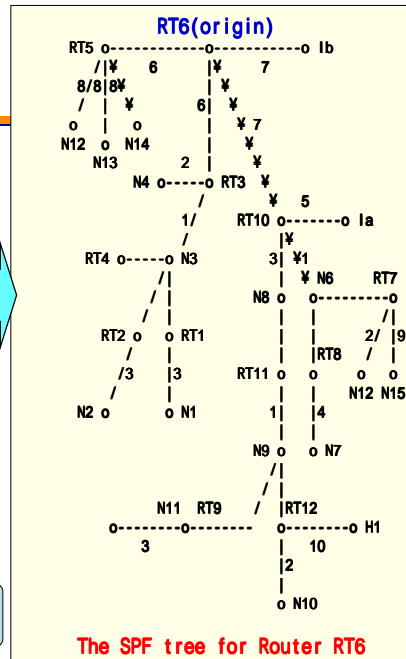


# 最短パスツリー

\*\*FROM\*\*

	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	N3	N6	N8	N9
	1	2	3	4	5	6	7	8	9	10	11	12	N3	N6	N8	N9
RT1																10
RT2																10
RT3																10
RT4																10
RT5																10
RT6																10
RT7																10
RT8																10
RT9																10
RT10																10
RT11																10
RT12																10
N1																10
N2																10
N3																10
N4																10
N6																10
N7																10
N8																10
N9																10
N10																10
N11																10
N12																10
N13																10
N14																10
N15																10
HI																10
Ia																10
Ib																10

SPF(Dijkstra) アルゴリズム



The SPF tree for Router RT6



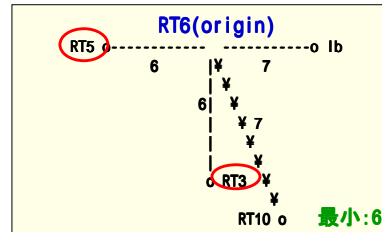
# SPF(Dijkstra)アルゴリズム (1)

すべての中で最小のものを確定していき、次はそこから次のノードまでを加えていく

		**FROM**																		
		RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT
		1	2	3	4	5	6	7	8	9	10	11	12	N3	N6	N8	N9			
RT1																			0	
RT2																			0	
RT3							6												0	
RT4						8	8												0	
RT5						8	6	6											0	
RT6			8		7				5										0	
RT7					6														0	
RT8																			0	
RT9																			0	
RT10								7										0	0	0
RT11																		0	0	0
RT12																		0	0	0
N13																				0
N2		3																		
N3		1	1	1																
N4			2																	
N6						1	1	1												
N7							4													
N8									3	2										
N9								1		1	1									
N10																				
N11										3										
N12						8		2												
N13						8														
N14						8														
N15								9												
H1											10									
Ia											5									
Ib												7								

データベースを見て、RT6から次のノードまでのツリーを作る

1回目



○ : 確定

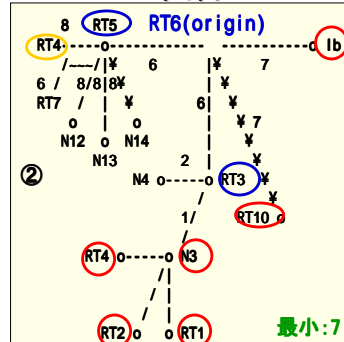
現在リーフにあるノードの中でRT6からのコストが最小である6のノードを確定する

# SPF(Dijkstra)アルゴリズム (2)

確定したところからDBを見て次のノードまで伸ばす (RT6などの既に確定しているノードは除く)

		**FROM**																		
		RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT	RT
		1	2	3	4	5	6	7	8	9	10	11	12	N3	N6	N8	N9			
RT1																			0	
RT2																			0	
RT3							6												0	
RT4						8	8												0	
RT5						8	6	6											0	
RT6			8		7				5										0	
RT7					6														0	
RT8																			0	
RT9																			0	
RT10								7										0	0	0
RT11																		0	0	0
RT12																		0	0	0
N13																				0
N2		3																		
N3		1	1	1																
N4			2																	
N6						1	1	1												
N7							4													
N8									3	2										
N9								1		1	1									
N10																				
N11										3										
N12						8		2												
N13						8														
N14						8														
N15								9												
H1											10									
Ia											5									
Ib												7								

2回目



○ : 旧確定 ○ : 新確定 ○ : 消去

現在リーフにあるノードの中でRT6からのコストが最小である7のノードを確定する

RT4はRT6→RT3→N3→RT4で確定した⑧で RT5→RT4のところは消去する

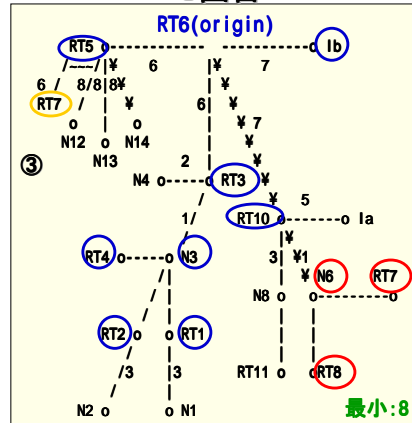
# SPF(Dijkstra)アルゴリズム (3)

確定したところからDBを見て次のノードまで伸ばす。これを繰り返す。

\*\*FROM\*\*

	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	N1	N2	N3	N4	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15	H1	Ia	Ib
RT1	0																												
RT2		0																											
RT3			0																										
RT4				0																									
RT5					0																								
RT6						0																							
RT7							0																						
RT8								0																					
RT9									0																				
RT10										0																			
RT11											0																		
RT12												0																	
N1	3												0																
N2		3												0															
N3	1		1	1											0														
N4				2												0													
N6							1	1									0												
N7								4										0											
N8									1										0										
N9										1										0									
N10											1										0								
N11												2										0							
N12													8										0						
N13														8										0					
N14															8										0				
N15																9													
H1																											10		
Ia																												5	
Ib																													7

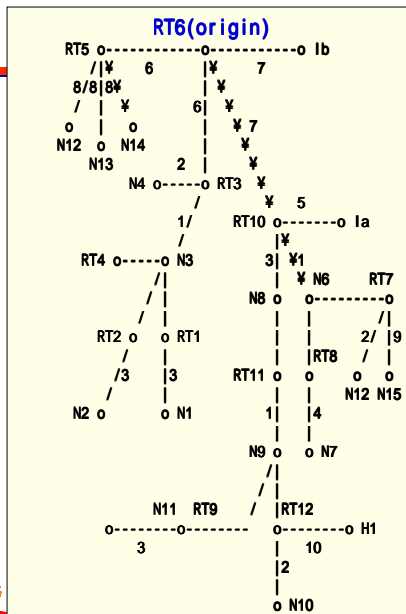
## 3回目



○:旧確定 ○:新確定 ○:消去

RT7はRT6→RT10→N6→RT7で確定したので RT5→RT7のところは消去する

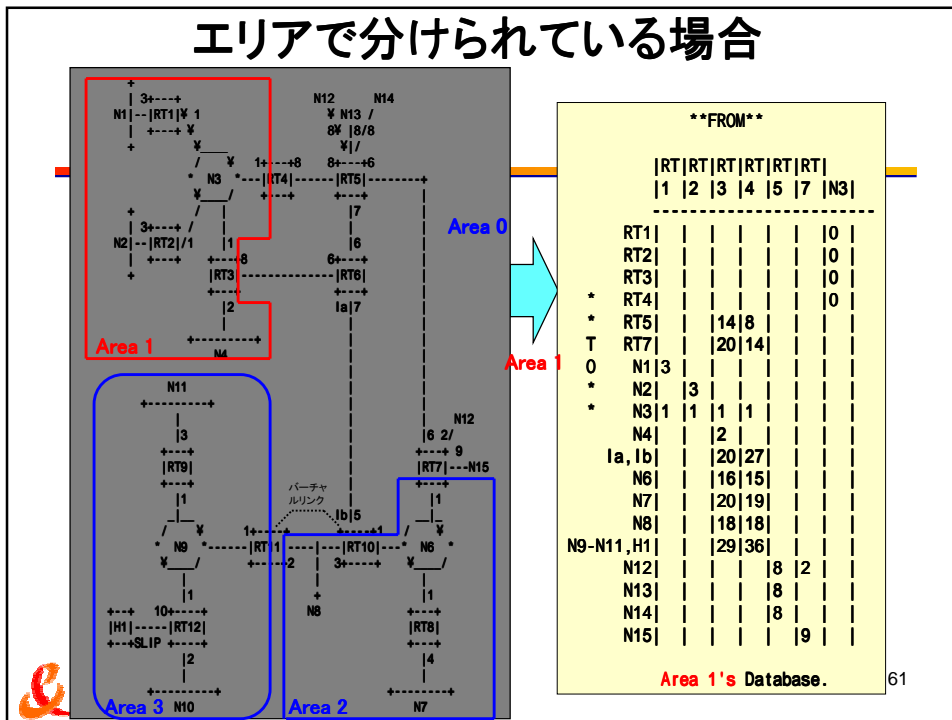
## ルーティング テーブルの作成



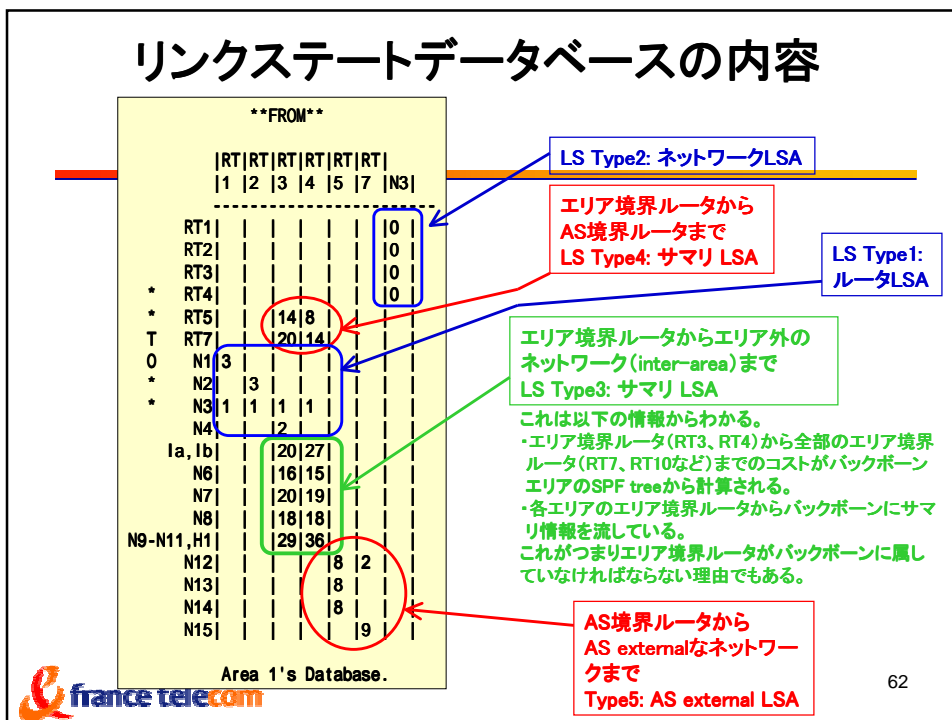
Destination	Next Hop	Distance
N1	RT3	10
N2	RT3	10
N3	RT3	7
N4	RT3	8
Ib	*	7
Ia	RT10	12
N6	RT10	8
N7	RT10	12
N8	RT10	10
N9	RT10	11
N10	RT10	13
N11	RT10	14
H1	RT10	21
<hr/>		
RT5	RT5	6
RT7	RT10	8

The portion of Router RT6's routing table listing local destinations.

## エリアで分けられている場合



## リンクステートデータベースの内容



# リンクステート広告(LSA) その2

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

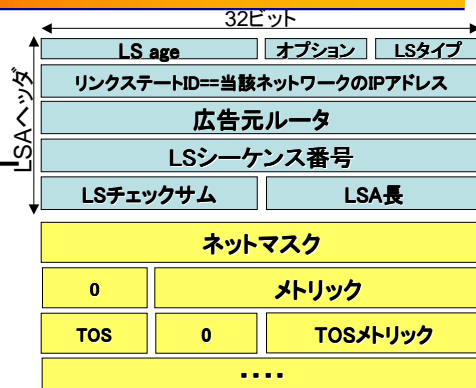
## AS境界とスタブエリア

- AS境界
  - OSPFの議論でAS境界と言う場合、「他のプロトコル (connected, static含む) が接するところ」という意味で使う。
- スタブ(stub)エリア
  - AS境界を一切含まないエリア
  - そのスタブエリアであることを明示的に定義することで、ABRでデフォルトルートを生成してエリア内のルーティングを実現し、エリア内のOSPF負荷を軽減することが出来る



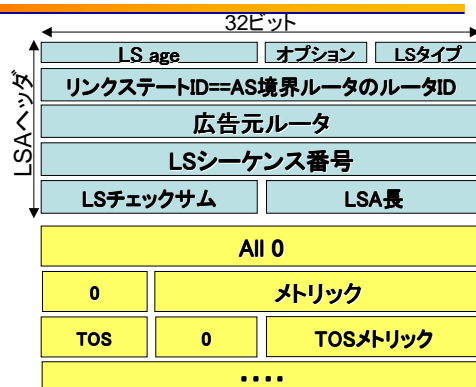
## サマリLSA (LSAタイプ3)

- エリア境界ルータによって生成される
- エリア外のネットワークに関する情報



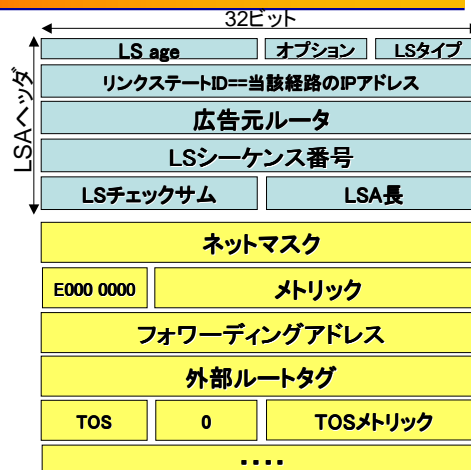
## サマリLSA (LSAタイプ4)

- エリア境界ルータによって生成される
- AS境界ルータの情報



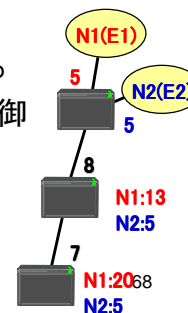
## AS-External LSA (LSAタイプ5)

- AS境界ルータによって生成される
- 他のプロトコルから redistribute(再分配)された経路
- Eビット: Type1(E=0), Type2(E=1)で、外部経路の取り扱いを変える



## 外部経路(External Routes)の取り扱い

- 扱い方(同一プリフィクスの選択方法)に2つのタイプ
  - Type1: redistributeされたメトリック+OSPFコスト
    - Redistribute時のメトリックを一定にすれば、最も近いASBRに経路制御される
  - Type2: redistributeされたコストのまま
    - Redistribute時のメトリックで優先制御を決定付ける
  - 同一プリフィクスに対して以下のように優先制御
    - IntraArea>InterArea>External 1>External 2



## NSSA – Not So Stubby Area

---

- RFC1587 "The OSPF NSSA Option"
- Not So Stubby – そんなにスタブっぽくない – 準スタブエリア
- Type 7 LSAを使う
- NSSAではType7 LSAを流すことができる
- NSSAのAS境界ルータでAS外部経路をType7 LSAとして redistributeする
  - Type 7 LSAsはNSSAのASBRでしか生成されず、NSSA内にしか流れない
  - 他のエリアに対しては、ABRでType 7 LSAsをType 5 LSAsに変更する。そのときサマライズやフィルターすることもできる。

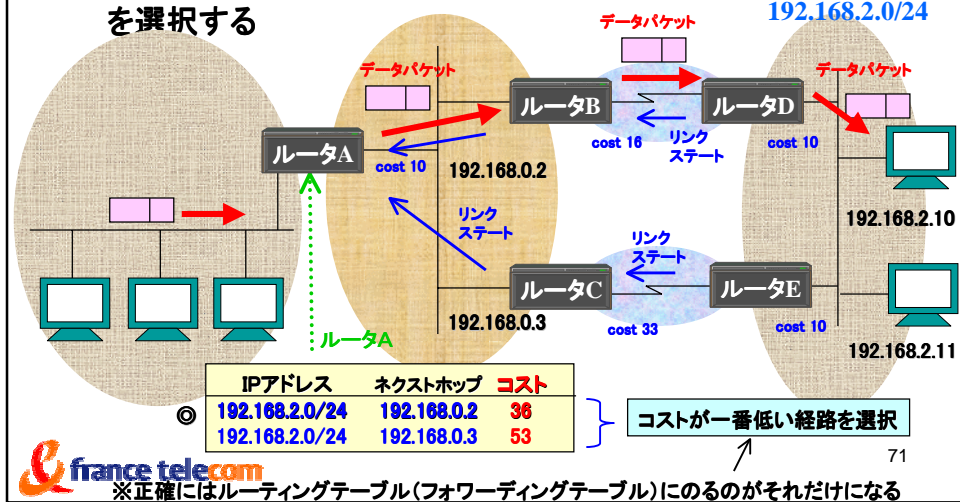
## OSPFの方路選択と 耐障害性

---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

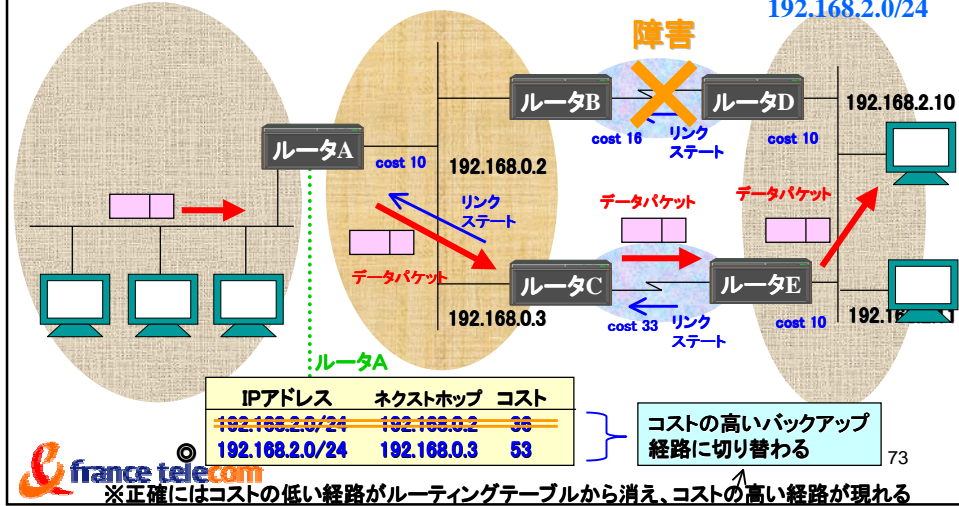
# コスト

- 同じネットワークが複数見える場合、コストが一番低い経路を選択する



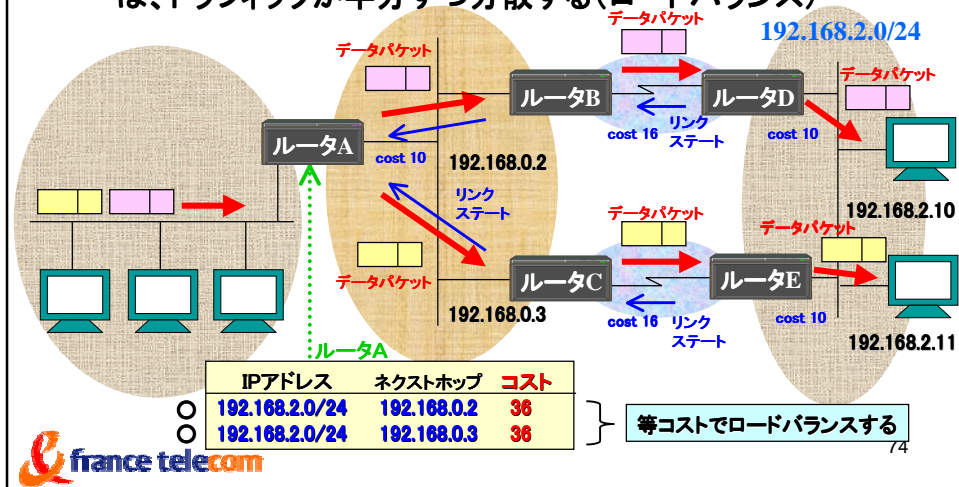
## 障害時にはバックアップ経路に切り替わる

- 障害時には、コストの高いバックアップ経路に切り替わる



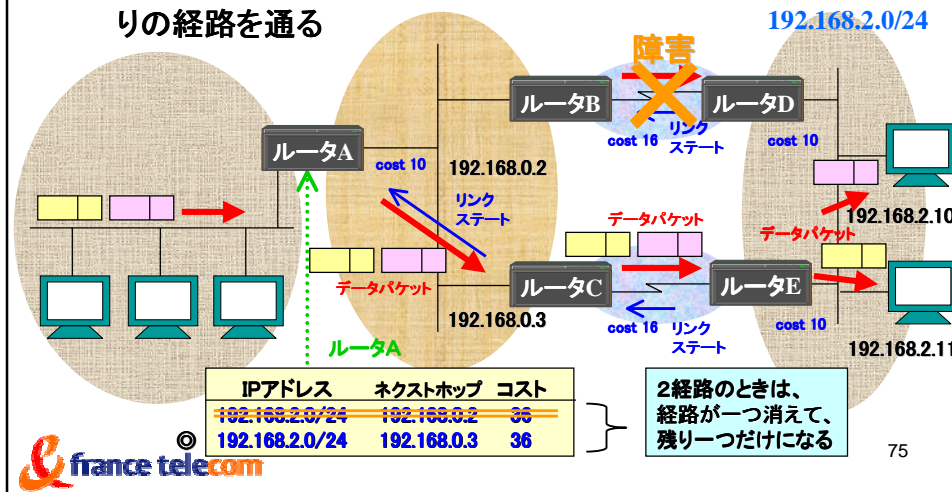
## ロードバランス – イコールコストマルチパス

- 同じネットワークが同じコストで見えるネットワークに対しては、トラフィックが半分ずつ分散する(ロードバランス)



## イコールコストマルチパスの 障害時の挙動

- 障害時には、障害した経路が消えて全てのトラフィックが残りの経路を通る



## イコールコストマルチパスの取り扱い

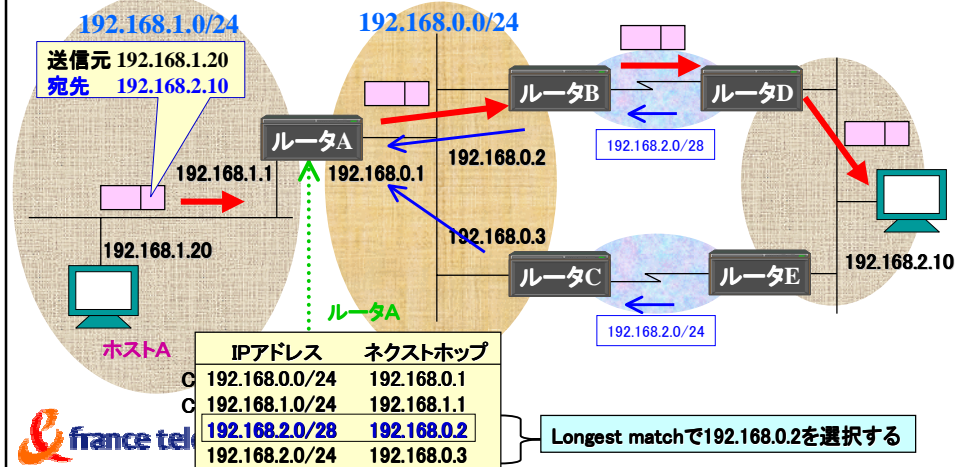
- OSPFでは同コストの複数方路を同時にルーティングテーブルに乗せられる機能を提供する
- パケットフォワーディングを実現するのはルータの実装
  - Ciscoでは最大6 (12.3では最大16まで許可)
    - maximum-paths 6 (router ospf セクション)

# 基本的な適用技術

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

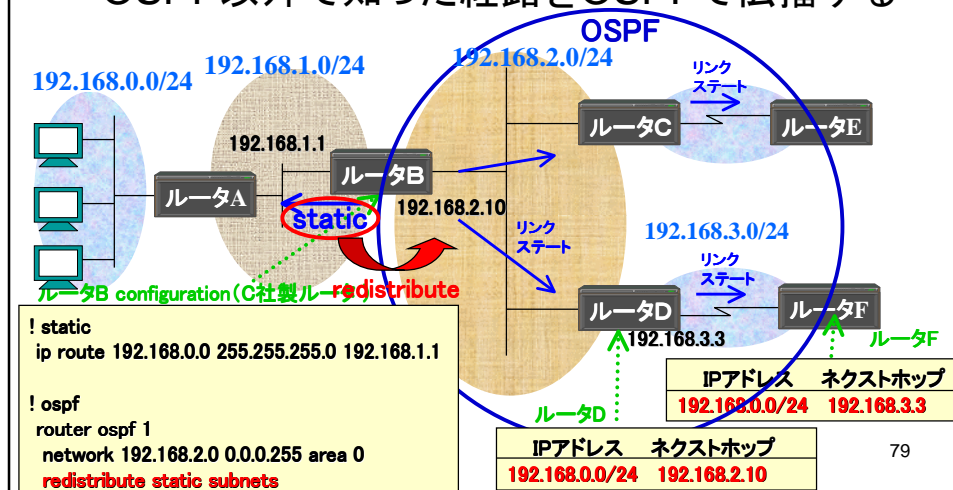
## 最長一致法アルゴリズム (longest-match)

- IPパケットの宛先アドレスを調べて、一致するネットワークアドレスが複数ある場合には、ビット列が長い方のネットワークアドレスを選択する



## Redistribution – 再分配

- OSPF以外で知った経路をOSPFで伝播する



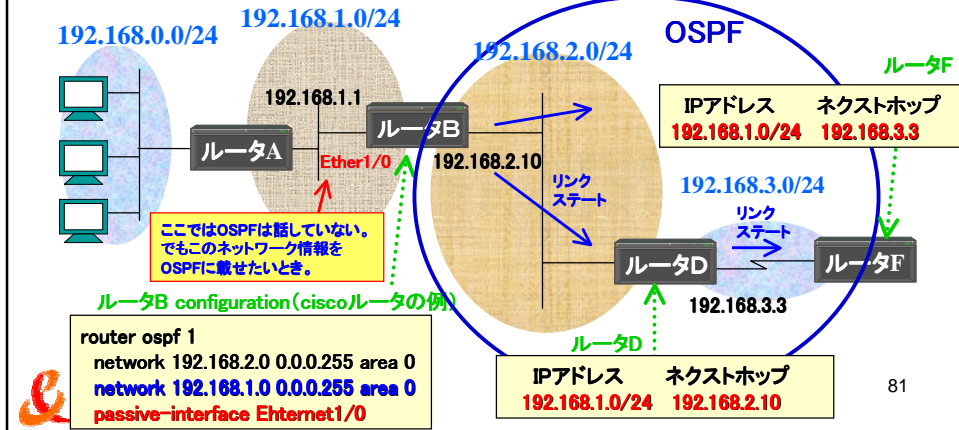
## Redistribution – 再分配

- 最も一般的な例 – スタティック経路をOSPFで認識させる
  - 特にデフォルトルート
    - Default-information originate (ciscoのコマンド)
- コネクテッド経路をredistributeする例もある
  - Passive-interfaceでも実現可能
- 他のルーティングプロトコルからのredistribution
- OSPF的には「外部経路(external routes)」となり、redistributeを設定するルータはAS境界ルータとなる



# Passive-Interface

- **passive-interface Ethernet1/0**
  - そのインタフェース上にOSPFのネイバはいないが、そのセグメントをOSPF上で認識させたい場合

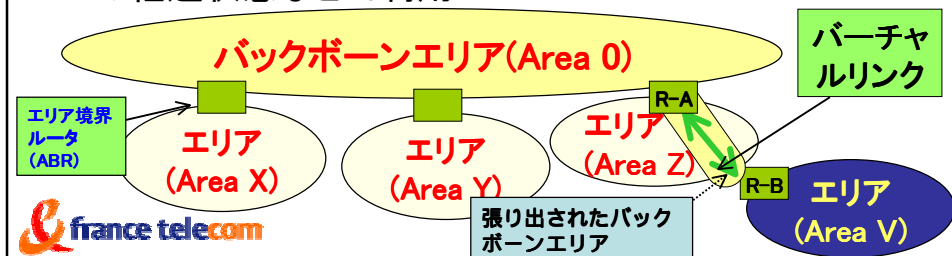


# エリア設計

- 基本 - 可能な限りシンプルに：最初からエリア分けを考えない。まずはエリア0のシングルエリアで考える。
- 信頼性を必要であれば、一つのエリアに複数のエリア境界ルータを設置する
- 一つのエリア境界ルータが所属するエリアはなるべく2つまでにすべき
  - つまりエリア0ともう一つのエリア、ということになる
- 経路の集約
  - エリア境界ルータにて経路の集約をする
  - エリアごとに経路を集約できるように、アドレス設計をする
    - area \*\* range <address> <mask> (エリア境界ルータ)
  - OSPFにredistributeされる経路も集約できるように、アドレス設計する
    - summary-address <address> <mask> (AS境界ルータ)

## バーチャルリンク

- バックボーンに対して物理的コネクションを持たないエリアを取り込む
- Area Z virtual-link [R-B loopback addr ]
- 設計が過度に複雑になるのを避けるため、定常状態での利用は避ける。障害時の緊急利用や移行作業の経過状態などで利用



## ルータID

- ループバックインターフェースを定義して、IPアドレスをOSPFのルータID, iBGPのネイバアドレスとして利用する
  - 物理インターフェースをルータIDやiBGPネイバアドレスとすると、障害でそのIPアドレスが利用不可能となったときにルータIDを変更する必要があり、不安定要素となる。
  - ループバックインターフェースは仮想インターフェースで、決してダウンしない:ルータIDとして常に利用可能
  - /32で十分
  - その名の通りルータの識別子として利用可能。

# IS-ISとの比較

Intermediate-System to Intermediate-System

---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
OSPF – Open Shortest Path First

## IS-IS

### OSIのリンクステート型プロトコル

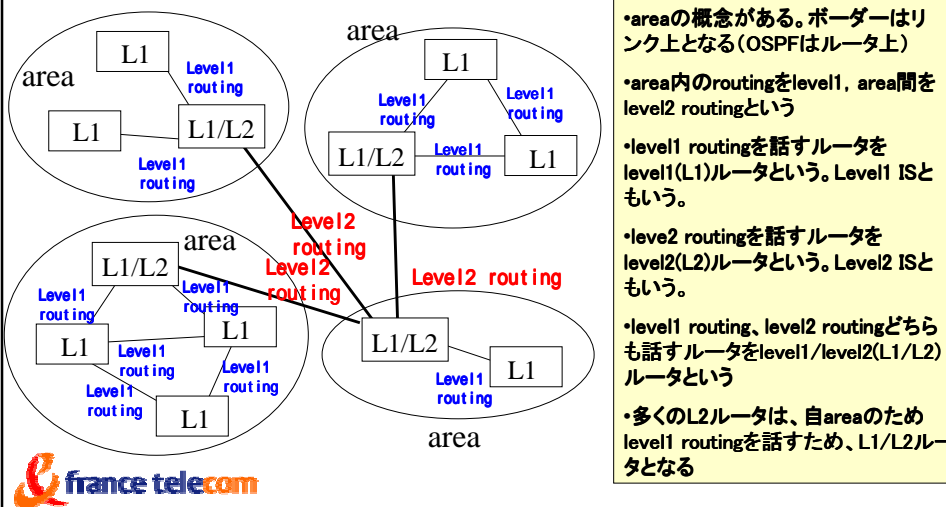
---

- OSPFと非常に類似したリンクステート型プロトコル
- もともとはOSI CLNS(connectionless network service)用のCLNP(CLN Protocol)の経路情報交換用で、それをIPも扱えるように拡張(Enhanced IS-IS, DUAL IS-IS)
- OSPFよりも前に実装されたため、IGPの規模対応を迫られた大手ISPを中心に利用。未だに多くの大手ISPが利用
  - 多くの大手ISPが利用するため安定性が高く、スケーラビリティも高そうだが、OSPFとアーキテクチャ的に大差ないと言われている。

# IS-ISとOSPF・プロトコルの比較

	IS-IS	OSPF
プロトコルスタック	OSI	TCP/IP
情報伝達に利用するプロトコル	CLNP (Connectionless Network Protocol)	IP(インターネットプロトコル)
ノードID	NET (Network Entry Title, NSAPアドレス)	ルータID(IPアドレス)
階層化, 区画化	レベル1(エリア内) レベル2(エリア間)	エリアによる区画化, 上位エリアはエリア0
リンクステートの交換	LSP(リンクステートPDU)	LSA(リンクステート広告)
コストの値	1リンク63(6ビット)まで, 総和1,023(10ビット)まで	65,535(16ビット)まで

# IS-ISネットワーク構成例



- areaの概念がある。ポーターはリンク上となる(OSPFはルータ上)
- area内のroutingをlevel1, area間をlevel2 routingという
- level1 routingを話すルータをlevel1(L1)ルータという。Level1 ISともいう。
- level2 routingを話すルータをlevel2(L2)ルータという。Level2 ISともいう。
- level1 routing、level2 routingどちらも話すルータをlevel1/level2(L1/L2)ルータという
- 多くのL2ルータは、自areaのためlevel1 routingを話すため、L1/L2ルータとなる

## Level1 / Level2 ルーティング

---

- SPFアルゴリズム
  - Level1とLevel2両方それぞれに関して独立に走る
- Level1 IS ルータにおいて
  - 他エリアへの通信に関しては、metric的に最も近いL1/L2ルータに向けてdefault routeを向けることによって通信が可能となる。
- 実際的には、Level2のみの起動の場合が多い

## BGP – Border Gateway Protocol

---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～

## セクション目次

---

- BGP – Border Gateway Protocol
  - ASと階層的経路制御
  - BGPの特徴
  - パケットタイプとセッション確立
  - パス属性値 – Path Attributes
  - 単純なBGPシステムの導入
  - iBGPシステムの構成
  - iBGPシステムのスケーラビリティ
  - 現在の経路制御におけるBGPとOSPFの関係
  - ポリシルーティング

## ASと階層的経路制御

---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

## ASとAS番号

---

- AS – Autonomous System
  - 自律システム
  - 別名: ルーティングドメイン
  - 単一管理・単一ルーティングポリシーで運用されるネットワーク領域 – 一般的には一つのISP
- AS番号 – ASN : Autonomous System Number
  - 16ビット(1~65535)の番号空間を持つ
    - AS5511==FT/*Opentransit Internet*, AS2914== *NTT/Verio*,
    - 64512~65534はプライベートAS, 65535はIANAリザーブ
    - 現在最大値は30000程度, 16,000個程度が観測される
  - The Internet とは、ASが相互接続された全体

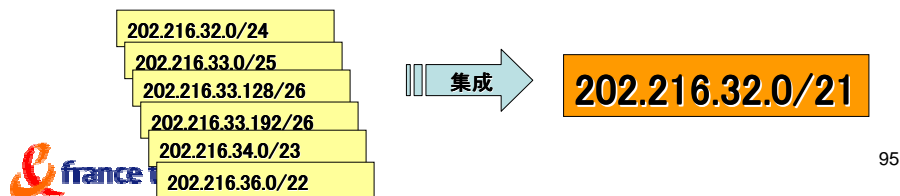
## The Internetにおける 階層的経路制御(1)

---

- CIDR – Classless Inter-Domain Routing
  - クラスレスにAS間ルーティングを実施する
  - 複数のClassC(=/24)アドレスも(あらゆるアドレスが)、任意の大きさでひとかたまりに扱える
  - AS内の小さなネットワークセグメント, ユーザネットワークをひとかたまりにして他のASに広告できる
    - 経路集積—aggregation

## The Internetにおける 階層的経路制御(2)

- 経路集成 – Aggregation
  - 複数の経路情報をひとかたまりにして、より大きなサイズの(より短いプレフィックスの)単一の経路情報にすること
  - 現在IPアドレスの割り振りはISP毎に行われているので、そこからユーザに割り当てるIPアドレスは割り振りブロックで集成することができる。



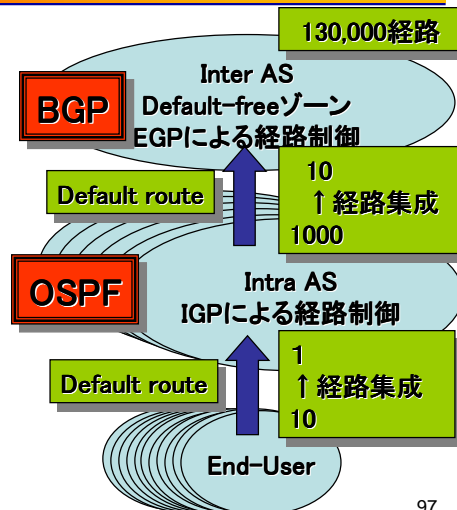
## The Internetにおける 階層的経路制御(3)

- 全インターネットを3つに階層化して、それぞれ独立して経路制御を扱う
  - InterAS
    - AS間, Default-Freeゾーン, EGPで制御
  - IntraAS
    - AS内, AS内の全経路, IGPで制御
  - End-User
    - ユーザサイト内。StaticやIGPで制御



## The Internetにおける 階層的経路制御(4)

- それぞれの境界で経路集成=情報量の縮退
- 上流の経路は全て default route で制御する
- 下流の詳細構成は気にせず、ひとかたまりの経路で制御する



## The Internetにおける 階層的経路制御(5)

- その内在的矛盾
  - CIDRは非階層的アドレス形態であったIPアドレスに階層構造を持ち込んだ
  - 階層構造を厳格に推し進めようとする...
    - 電話番号のように局番固定割り当てのような構造が望ましい
  - 末端に近くなるほどマルチホームがしにくい
    - 小さいアドレスブロックでマルチホームをするのは難しい
  - マルチホーム用の小さいブロックの割り当てが容認されつつある
    - 階層的経路制御の崩壊の兆し

# BGPの特徴


ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

## 基本事項の確認

- 現在のバージョンは4 – BGP4, RFC1771
- AS間経路制御に用いる
- ピアリング(peering) – 明示的に定義した隣接ルータとの間にTCP上でセッションを確立し、経路情報を交換する
- パスベクター型プロトコルと呼ばれ、プリフィクス単位の経路情報レコードにはパス属性と呼ばれる属性情報が添えられている。
- パス属性により経路の優先制御を行う。
- パス属性を調整することで経路制御ポリシーを実装することができる。


## BGPとOSPFの比較(1)

OSPF	BGP
IP上に直接乗るプロトコル Protocol number: 89	TCP上に乗るプロトコル Port number: 179
リンクステート型プロトコル リンクステート情報を伝播 状態変更毎にLSA, 連鎖伝播	パスベクター型プロトコル パス情報を伝播 状態変更毎にUPDATE, 連鎖伝播

 101

## BGPとOSPFの比較(2)

OSPF	BGP
基本的に、OSPFを起動した隣接ルータ全てと経路交換	明示的に定義した隣接ルータのみと経路交換
マルチキャストでセグメント上の全OSPFルータとやりとり	隣接ルータ毎にBGPセッションを確立(ピアリング)
あるネットワーク(ルータ)の状態変更は、全ルータのパスツリー再作成を引き起こす 30分でリフレッシュ--flooding	あるネットワークの状態変化は基本的にはそのプリフィクスだけの問題 リフレッシュなし

 102

## BGPとOSPFの比較(3)

OSPF	BGP
<p><b>トポロジの管理</b>に主眼を置く</p> <p>エリア内共通のLSDBを全ルータが作成し、LSDBから各ルータそれぞれがパスツリーを作成</p> <p>経路個別のポリシー付加は不可</p>	<p>プリフィクス(ネットワーク)の<b>パス属性</b>に着目</p> <p>受領したUPDATEは各AS, ルータのポリシーに基づいて処理, 以遠伝播する</p> <p>経路個別にポリシー付加が可能 →パス属性値としてプリフィクスに付加</p>
<p>精密で敏速な経路制御</p>	<p>ポリシーに基づいた経路制御</p>



## BGPとOSPFの比較(4)

OSPF	BGP
<p>IGP AS内の経路制御・トポロジ管理</p>	<p>EGP AS間の経路制御</p>
<p>AS内の各ルータはAS内の経路情報とネットワーク状態を交換し合っている</p>	<p>AS内の各ルータは他のASの経路情報を交換し合って同期を取っている</p>



# パケットタイプと セッション確立

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

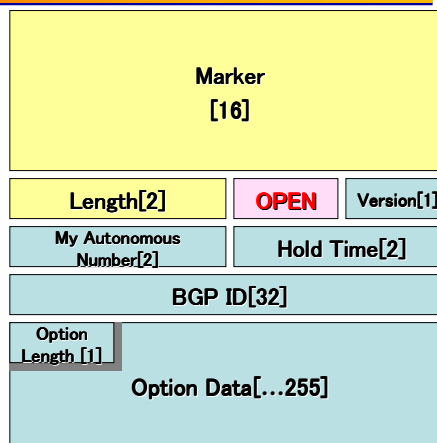
## メッセージヘッダ

- マーカー(Marker)
  - セキュリティ目的に利用
- 長さ(length)
- タイプ(Type)
  - メッセージタイプ
    - オープン(OPEN)
    - 更新(UPDATE)
    - 通知(NOTIFICATION)
    - キープアライブ(KEEPALIVE)



## OPENメッセージ

- BGPセッション開設に利用
- 以下のパラメータ提示してネゴシエーションを実施
  - バージョン
    - 現在はバージョン4
  - 自AS番号
  - ホールドタイム
    - キープアライブの間隔を指定
  - BGP ID
    - 自分が持つIPアドレスからひとつをIDとして利用する
  - オプションは現在のところ認証情報のみが定義されている
- 受諾にはKEEPALIVE, 拒否にはNOTIFICATIONを返す



107

## KEEPALIVEメッセージ

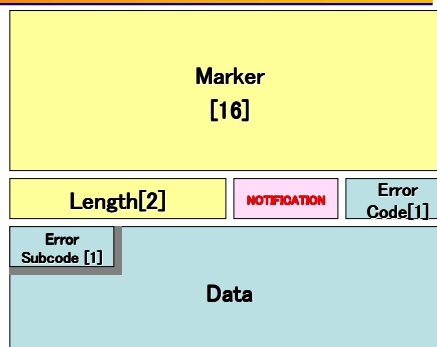
- セッションの正常性確認に利用
- UPDATEが一定期間以上発生しない場合に送信
  - Hold Time より頻繁に



108

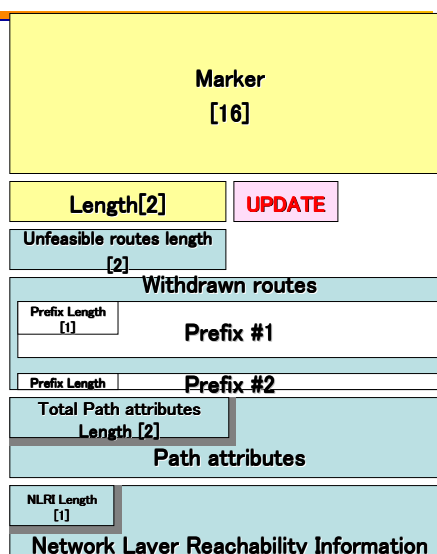
## NOTIFICATIONメッセージ

- エラー通知し、セッションを終了する
  - フォーマット不良
  - 無効値
  - 状態遷移エラー
  - Hold Time 満了
  - セッション中止



## UPDATEメッセージ

- 経路情報の交換に利用
- 取り消される(withdrawn) プリフィクスを複数、及びパス属性と到達可能プリフィクス(NLRI)の組一対を伝達可能
  - 経路情報を消すのは一気にできる
  - パス属性の変更や新しい経路情報は1プリフィクスずつUPDATEを送信



## BGPセッションのライフサイクル

---

- 両ルータでのコンフィグレーション
- OPEN
  - バージョン不一致などなど不具合があればNOTIFICATIONを発して切断
  - 不具合なければKEEPALIVEを返してセッション確立
- UPDATEパケットで経路情報を交換
- 新たな経路情報がなければ、KEEPALIVEでセッション維持
- セッション切断のためにはNOTIFICATION

## パス属性値 – Path Attributes

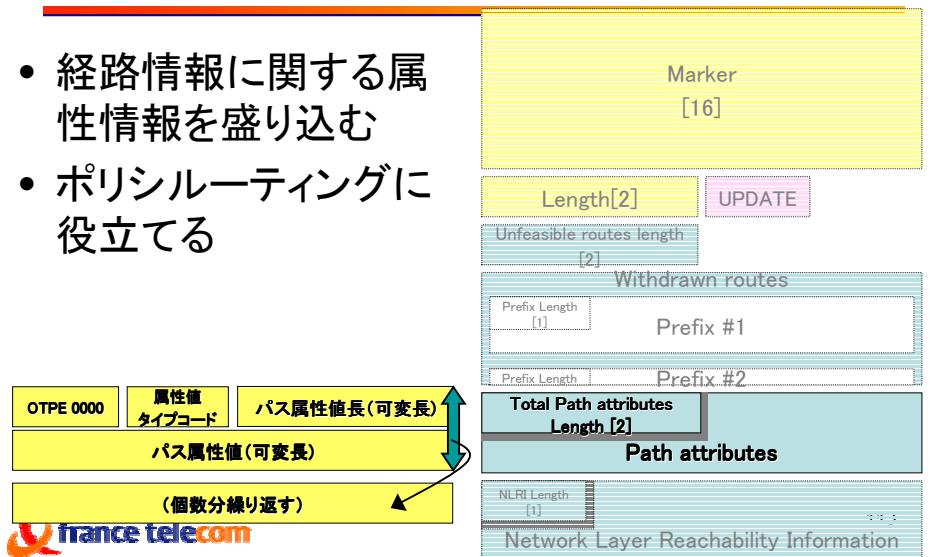
---

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol



# パス属性値

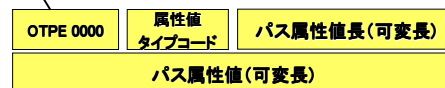
- 経路情報に関する属性情報を盛り込む
- ポリシルーティングに役立てる



# パス属性値の分類

- Bit0(O): Optional
  - 0=wellknown, 1=optional
- Bit1(T): Transitive
  - 0=non-transitive, 1=transitive
- Bit2(P): Partial
  - 0=complete, 1=partial
- Bit3(E): Extended Length
  - 0=パス属性値長は1オクテット
  - 1=パス属性値長は2オクテット

- プリフィクスに括りつけられた経路選択制御用の属性値群
- 必須, 任意, 透過性, 非透過性の4つに分類
  - 必須 - Well-known mandatory
    - 全てのBGPルータで解釈可能で、全ての経路レコードに必要
  - 任意 - Well-known discretionary
    - 全てのBGPルータで解釈可能で、必ずしもつけなくても良い
  - 透過性 - Optional transitive
    - 一部のBGPルータで解釈されない可能性があるが、解釈できなくても次に伝えるなければならない
  - 非透過性 - Optional non-transitive
    - 一部のBGPルータで解釈されない可能性があり、解釈できない場合次に伝える必要は必ずしもない



## パス属性値タイプ

値	名称・意味	分類	説明
1	ORIGIN	必須	必須なのでどんな経路にも付加される。(pp.115)
2	AS_PATH	必須	
3	NEXT_HOP	必須	
4	MULTI_EXIT_DISC	非透過	ポリシールーティングに大活躍(pp.116)
5	LOCAL_PREF	任意	
6	ATOMIC_AGGREGATE	任意	Aggregation関係
7	AGGREGATOR	透過	
8	COMMUNITY	透過	ポリシールーティングに大活躍(pp.116)
9	ORIGINATOR	非透過	BGPルートリフレクタに利用
10	CLUSTER_LIST	非透過	
14	MP_REACH_NLRI	非透過	BGP Multi Protocol Extension (BGP4+)の拡張に利用
15	MP_UNREACH_NLRI	非透過	
16	EXT_COMMUNITY	任意	MPLS-VPN用の拡張コミュニティ

OTPE 0000	属性値 タイプコード	パス属性値長(可変長)
	属性値(可変長)	

115

## Path Attributes(パス属性値) Well-known mandatory

- ORIGIN(type code=1)
  - 生成元のASでどういう形でBGP上に生成されたか
    - IGP, EGP, INCOMPLETE の3値
- AS\_PATH(type code=2)
  - 生成元ASまでの経過ASのリスト
- NEXT\_HOP(type code=3)
  - そのプリフィクスへの次のホップとなるIPアドレス

116

## Path Attributes(パス属性値) ポリシー制御のプレイヤーたち

---

- MULTI\_EXIT\_DISC(type code=4) – 非透過性
  - Multi Exit Discriminator
  - 複数相互接続点を持つ隣接ASに対してそれぞれの優先度を伝える
- LOCAL\_PREF(type code=5) – 任意
  - Local Preference
  - AS内で他ASから受け取った経路に関する優先度をつけるのに用いる
- COMMUNITY (type code=8) – 透過性
  - 任意の32ビットの情報を伝達する

## 単純なBGPシステムの導入

---

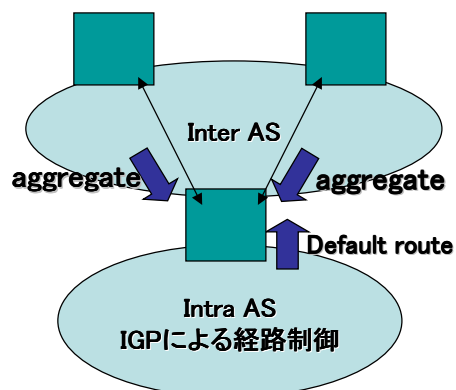
ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

## AS番号はどうやって 割り当てを受けるのか

- JPNICが割り当てを行う
  - <http://www.nic.ad.jp/ja/ip/asnumber.html>
- AS割り当ての条件 – RFC1930
  - 日本語訳  
<http://www.nic.ad.jp/ja/translation/rfc/1930.html>
  - マルチホーム接続 (IX接続を含む) となっていることが条件
    - シングルホームで済むのであれば、上流ASに含まれても問題なく、別のASを構成する必要がない

## 最も単純なBGPの導入

- IGPでデフォルトルートが指されるルータが単一のボーダルータ
- 2つ以上のASに接続



## BGP導入の実際

- 2つ以上の国内大手ISPを上流としてマルチホーム接続
- DIX-IE, NSPIXP3, JPIX, JPNAPなどのインターネットエクスチェンジに加入して、国内到達性を確保。別途国際ゲートウェイISP(あるいは国内大手ISP)に加入して海外到達性を確保
  - アドレスブロックは、JPNICなどから割り当てをうける

## BGPの 基本的な構成(1)

```
router bgp 50000 ← BGP起動
no synchronization
no auto-summary ← BGPグローバルコマンド
network 172.16.0.0 ← IGPで経路があればBGPで広告
network 192.0.1.0 ← 含まれるプリフィクスがIGPにあれば集成経路を広告
aggregate-address 223.224.0.0 255.255.0.0 summary-only
neighbor 210.171.224.110 remote-as 5511
neighbor 210.171.224.110 route-map AS551 ← 集成経路以外を抑制
neighbor 210.171.224.110 route-map jpix-out ← Peer確立
                                     ← Route-mapで
                                     ポリシを記述
```

## BGPの 基本的コンフィグレーション(2)

- Inbound方向のルートマップの例

```
route-map AS5511-in permit 10
match as-path 10
set local-preference 110
!
route-map AS5511-in permit 20
match as-path 20
set local-preference 100
!
```

シーケンス番号順  
に適用

それぞれのシーケ  
ンスで適合条件と  
アクションを定義

## BGPの 基本的コンフィグレーション(3)

- Outbound方向のルートマップの例

```
route-map jpix-out permit 10
match as-path 30
set metric 1000
!
```

# iBGPシステムの構成

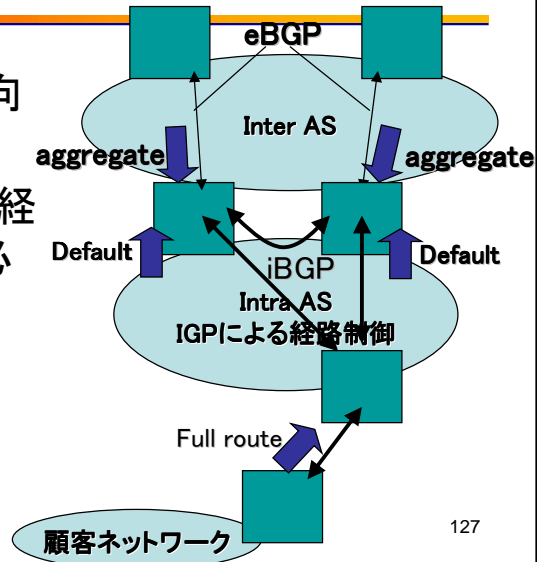
ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

# eBGPとiBGP

- eBGP – External BGP
  - 他のASとの間でセッションを張り経路情報の交換を行う
- iBGP – Internal BGP
  - 同じASの複数のBGPルータの間で、それぞれがeBGPを介して入手した(あるいは自AS内から生成した)外部経路を交換し、AS内の経路情報の同期を取る
  - 基本的には、iBGPで入手した経路情報はiBGPで遠く伝播しない
    - 全てのBGPルータとiBGPセッションを確立する必要がある(回避方法は後ほど)

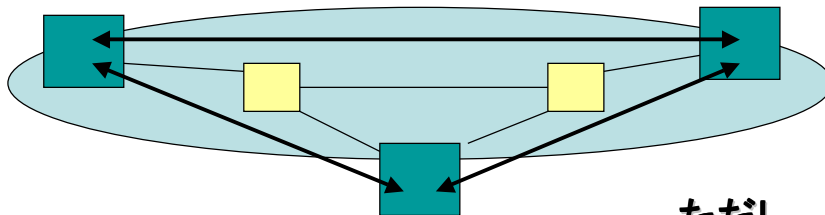
## 複数のボーダルータを置く

- 上流に2つ, 顧客向けに1つ
- ボーダルータ間の経路情報の同期が必要  
↓  
iBGPの確立



## iBGPの注意点

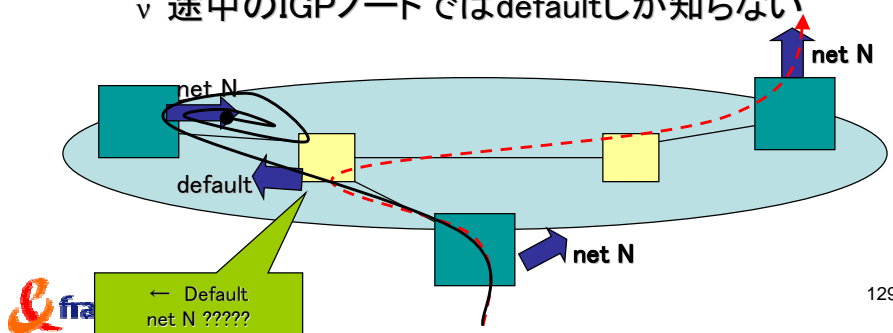
- eBGPは直接隣接を必要とするが、iBGPはAS内での同期が目的なので離れていても確立可能
- iBGPは全てのボーダルータとセッションを張る必要がある





## iBGP・仕様上の問題点

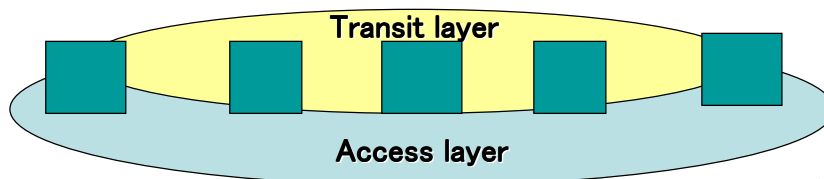
- Synchronization問題
  - トランジットしようとする経路はIGPで観測されていなければならない
  - ✓ Next-hopが別のボーダルータだった場合
  - ✓ 途中のIGPノードではdefaultしか知らない



129

## iBGPシステムの解

- No synchronization
  - IGP synchronizationの縛りを解くコマンド(cisco)
  - IGPで経路観測されない経路も利用可能
    - つまり、BGPルータ間に非BGPルータがあると矛盾が発生
- トランジット層の総BGPノード化
  - トランジット層とアクセス層の二層構造へ
  - BGPユーザが多い場合、「総トランジット層」に近づく



130

## iBGP問題のまとめ

---

- iBGPは隣接していなくても確立可能
- 仕様では、中間ノードが経路制御できないと問題があるので、IGPでBGP経路を知っている必要があった
- がしかし、それでは経路制御階層化の意味がないので、IGPとの同期を外すほうがよい
- IGP同期を外す結果、全てのBGPルータは隣接する必要がある
- BGPルータ(トランジット)層と非BGPルータ(アクセス)層の二層に階層化
- 総トランジット層へ

## iBGPシステムの基本(1) NEXT\_HOPをIGPで観測する

---

- iBGPで伝播される外部経路では、基本的にNEXT\_HOPの値は変わらない
  - eBGPの隣接ルータのIPアドレス
- BGP経路は、NEXT\_HOPがIGPでreachableでなければ有効とならない。そこで、
  - IXやプライベートピアリングのセグメントをIGPで認識させる
    - 例えばpassive-interfaceでOSPFプロセスに定義する
  - eBGPルータで、iBGPピアに対してnexthop-selfを設定して、自分のIPアドレスをNEXT\_HOPとして使う

## iBGPシステムの基本(2) loopbackをピア設定に利用する

- iBGPピアの設定では、loopbackアドレスを利用するのが「基本」
  - loopbackインターフェースはダウンしない
    - 隣接ルータと対面するインターフェースが落ちても迂回して到達することが可能
    - LoopbackインターフェースにもIGPを起動することを忘れずに
  - 全BGPルータで同じIPアドレスで対象ルータを認識することが可能

## iBGPの 基本的な構成

```
Interface Loopback 0
ip address 202.216.41.1 255.255.255.255
!
Interface FastEthernet 2/0
description NSPIX2 Segment
ip address 202.249.2.41 255.255.255.0
!
Router ospf 4000
network 202.216.41.1 0.0.0.0 area 0
network 202.249.2.0 0.0.0.255 area 0
passive-interface Loopback 0
passive-interface FastEthernet 2/0
!
Router bgp 4000
neighbor I BGP peer-group
neighbor I BGP remote-as 4000
neighbor I BGP update-source Loopback 0
neighbor 202.216.41.2 peer-group I BGP
neighbor 202.216.41.3 peer-group I BGP
neighbor 202.216.41.4 peer-group I BGP
```

Loopback 0 の設定 /32で構わない

FastE2/0 がIXセグメントだったとする

LoopbackとIXセグメントをOSPF上で定義、かつ非活性とする。これによって他のBGPルータでもそれぞれがIGP上で認識される

peer-groupを利用してみる。等質なコンフィグには非常に有効

Update-source で、ピアリングに利用するIPアドレスを定義する

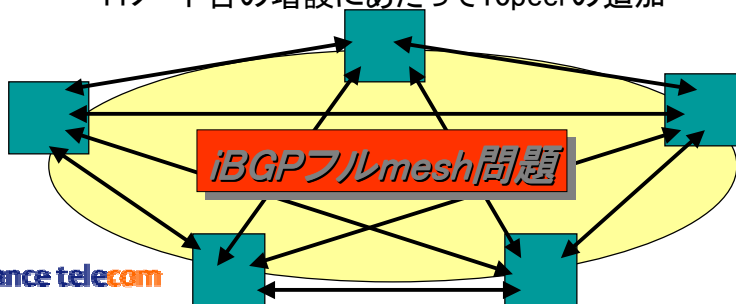
iBGPにloopbackアドレスを利用すると、BGPルータをIPアドレスで認識できるので運用上非常に便利

# iBGPシステムの スケーラビリティ

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

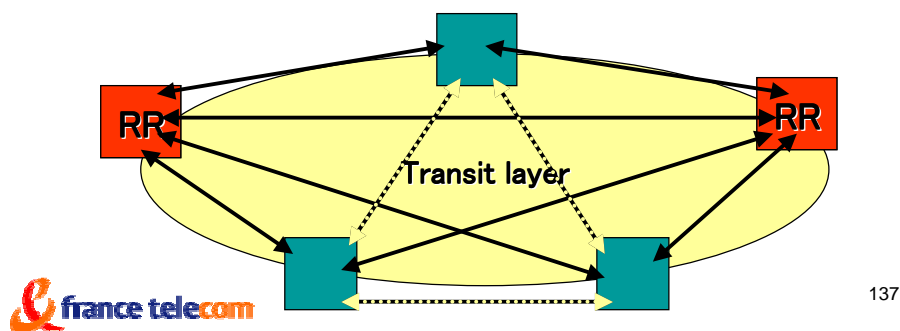
# iBGPシステムのスケーラビリティ

- iBGPで得た経路は他のiBGPpeerに再伝播しないため、全ノードをmesh状にpeerする
  - ボーダルータ5ノードで既に10peer
  - 10ノードでは?  ${}_{10}C_2 = 45$ 
    - 11ノード目の増設にあたって10peerの追加



## iBGPフルmesh問題解決策 iBGPルートリフレクタ(1)

- リフレクタとリフレクタクライアントの2階層化
- リフレクタからクライアントにはiBGPで得た経路を再分配する



## iBGPフルmesh問題解決策 iBGPルートリフレクタ(2)

- コンフィグレーション
  - リフレクタ側で以下のように設定
  - クライアント側では設定不要
    - 階層化可能

```
router bgp 5511
```

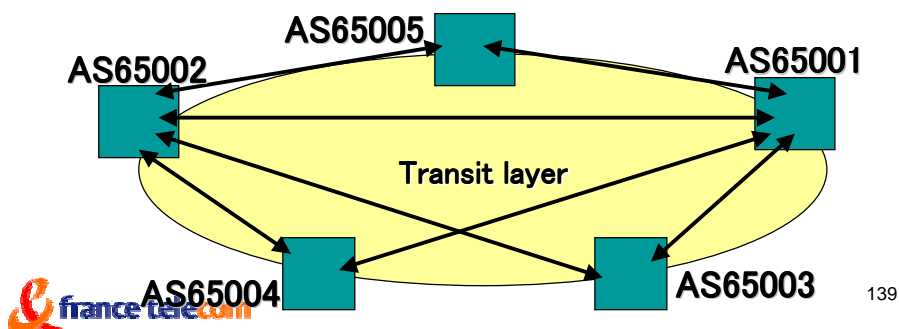
```
bgp cluster-id FOUR-BYTE-CLUSTER-ID
```

```
neighbor CLI.ENT.IPA.DDR remote-as 5511
```

```
neighbor CLI.ENT.IPA.DDR route-reflector-client
```

## iBGPフルmesh問題解決策 BGPコンフェデレーション(1)

- BGPコンフェデレーション(confederation)
  - ASの中を更に小さい単位でsubASに分け、その間をeBGPで結ぶ
  - フルmeshにはる必要はなくなる



## iBGPフルmesh問題解決策 BGPコンフェデレーション(2)

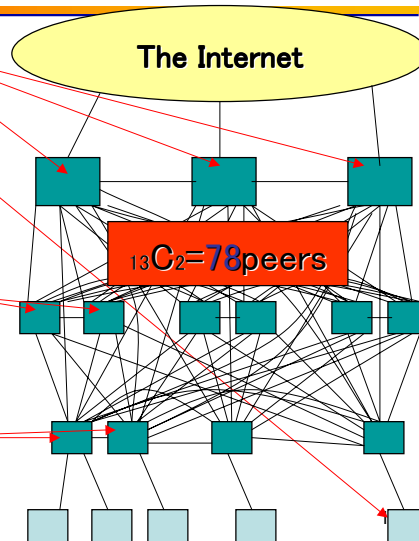
- コンフィグレーション
  - プライベートASを利用するのが普通
  - Confed内部となるAS番号をconfed peersで定義

```
router bgp 65000
  bgp confederation identifier 5511
  bgp confederation peers 65001 65002 65003 65004
  network .....
```

- 但しBGPコンフェデレーションはフルmesh回避よりも、管理領域分割や複数ネットワーク統合, IGP分割などに真価を発揮する

## AS内BGPスケーラビリティ問題の 実際

- 複数の対外接続
- 地域/POP毎にBGP接続加入者がいる
  - それぞれBGPノードが必要
- 冗長性確保が必要
  - POPにコアルータを2台
- BGP加入者増加
  - BGP加入者収容ルータの増加

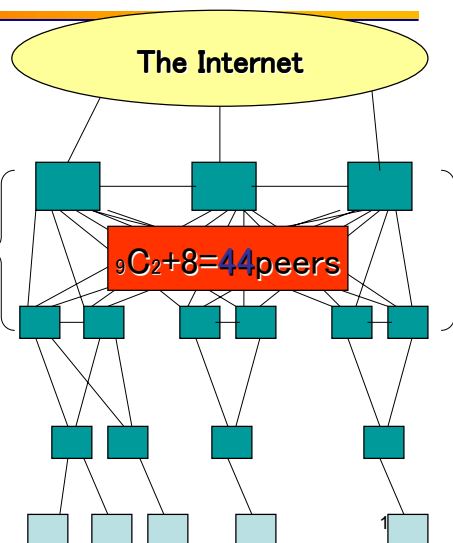


## AS内BGPスケーラビリティ問題の実際 —RRによる解法

- RRの導入

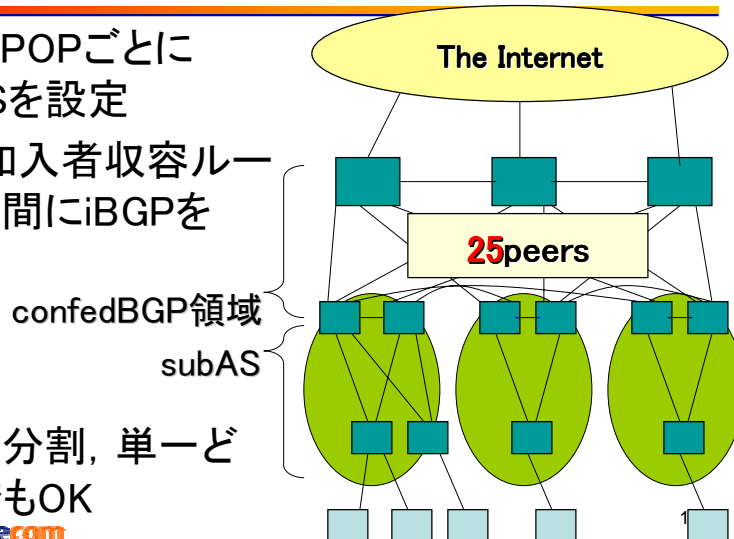
POPコアルータと対外接続ルータをフルメッシュ

加入者ルータがリフレクタクライアント



## AS内BGPスケーラビリティ問題の実際 —コンフェデレーションによる解法

- 地域・POPごとにsubASを設定
- BGP加入者収容ルータとの間にiBGPを設定
- IGPは分割, 単一どちらでもOK



## 現在の経路制御における BGPとOSPFの関係

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol



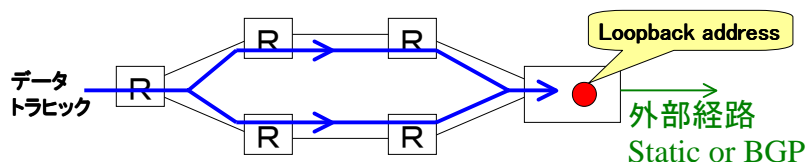


## 世界規模ISPにおける典型的なネットワーク構成

- 全ルータでBGPが起動される
  - そもそも末端ルータでもメモリフル実装
- BGPはルートリフレクタで階層化し、フルmeshを回避
- 加入者ルータ以外は二重化構成
- IGP(IS-ISが多い)によるロードバランシング実現
- Static経路はBGPにredistributeされる

## OSPFによるNEXT\_HOPへのロードバランシングの仕組み

- その経路へデータが行くためにはBGP next-hopであるredistributeしたルータのloopbackアドレスへ向かおうとする
- BGP next-hopへ向けてOSPFで作られたルーティングテーブルをrecursive lookupする
  - ロードバランスする



## BGPとOSPFの分担(1)

---

- OSPFはトポロジ管理に関しては精巧だが、外部経路を扱うことは不得手
  - 外部経路のフィルタリングも難しい
  - たとえスタティックルートでも、多くなると安定しない
- BGPはトポロジ管理はできないが、外部経路のコントロールは非常に得意
  - ポリシの付加やフィルタリングも容易

## BGPとOSPFの分担(2)

---

- いまやゲートウェイは複数・多数ある
  - IGPのデフォルトルートは外部対地の経路制御を行うことは非現実的
- iBGPがAS内のBGP経路の同期だけでなく、AS内での外部対地に対する経路制御を荷っている。

BGP: staticも含めた外部対地に対する経路制御

OSPF: 内部トポロジの管理

# ポリシルーティング

ISPバックボーンネットワークにおける経路制御設計～ 理論編 ～  
BGP – Border Gateway Protocol

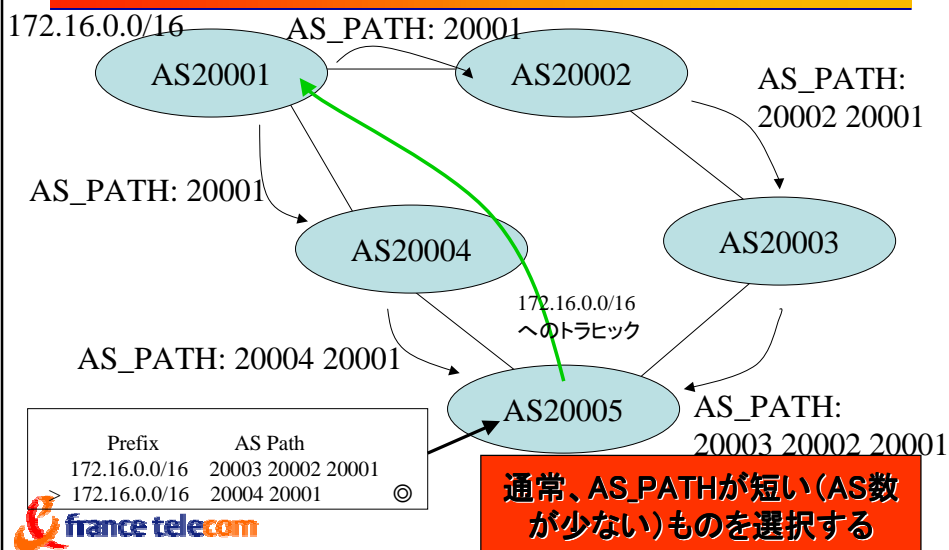
# ポリシルーティング

- BGPにおける経路情報の扱い
  - プリフィクス(NLRI)+パス属性
  - パス属性値の調整, パス属性値に基づく経路選択を行うことができる
- ルーティングポリシ
  - 複数peerを持つASとの間でどのようにトラヒックを交換するか
  - セキュリティのために経路をフィルタする
  - 複数のupstreamに対するトラヒックバランス

# ポリシールーティングを可能にする パス属性値

- AS\_PATH
  - 経過AS列, 短いほうが優先。
  - AS-path prependでAS列長の調整が可能
- LOCAL\_PREF – Local Preference
  - 設計者意図の優先順位付け
- MULTI\_EXIT\_DISC – Multi Exit Discriminator
  - 隣接する同一ASの複数peerの優先度
- COMMUNITY – Community Attribute
  - 32ビットの値を付加できる。プロトコル上、値に意味はないが、有効な利用法がカレントプラクティスに存在

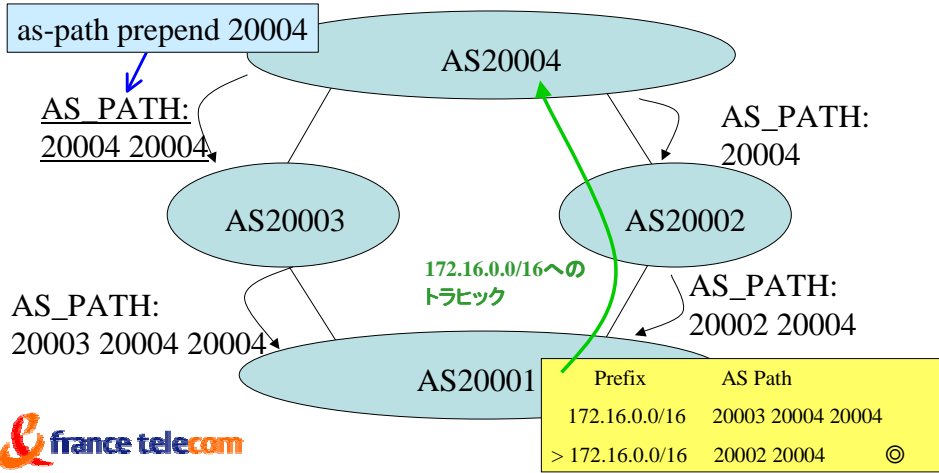
## AS\_PATH



# AS Path Prepend

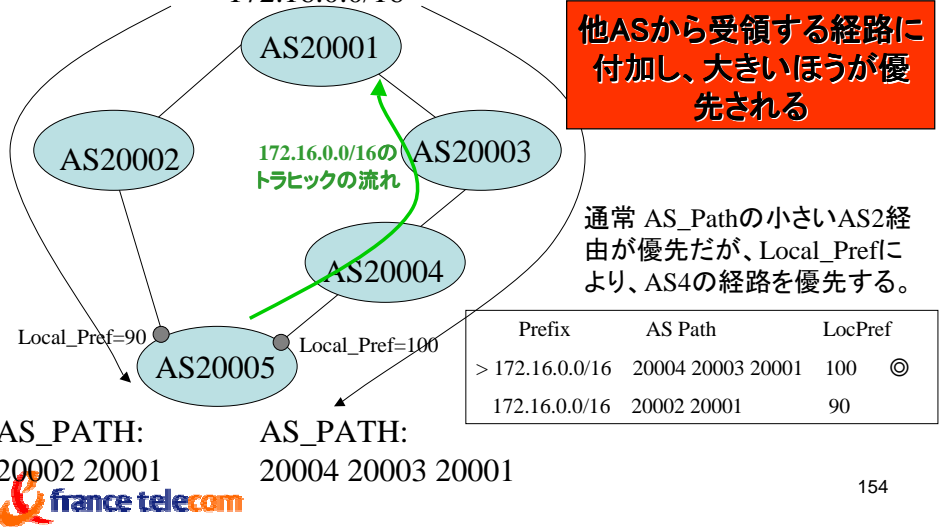
ASを余計につけて、AS\_PATH\_lengthを長く見せるテクニック

172.16.0.0/16

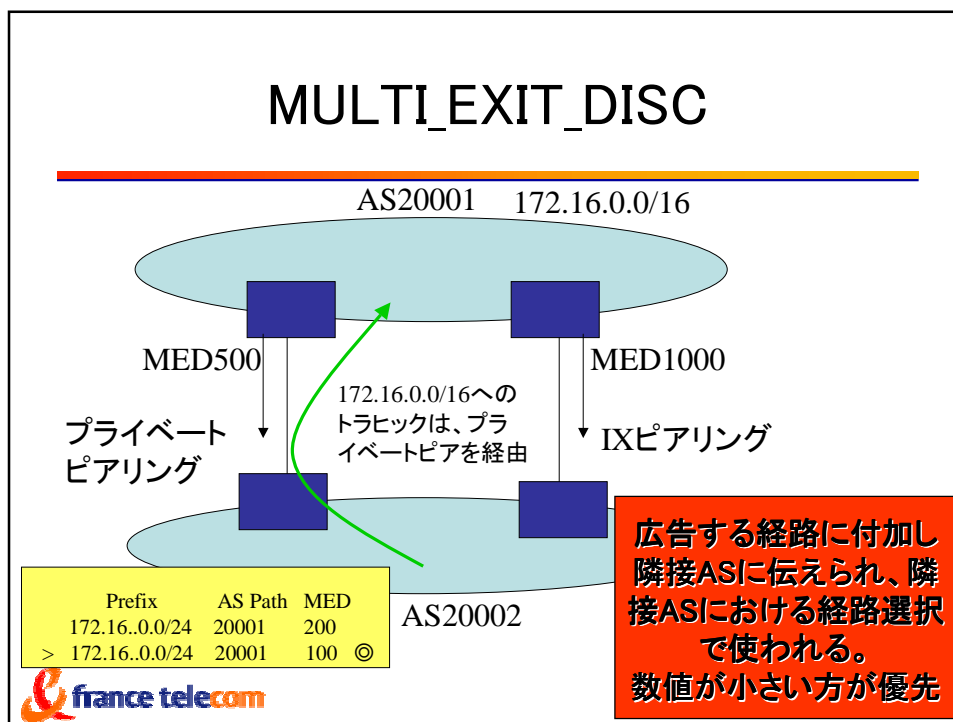


# LOCAL\_PREF

172.16.0.0/16



## MULTI\_EXIT\_DISC



## COMMUNITY(1)

- 32ビットの整数値, 透過性
- Well-known Community
  - No-export:
    - 自AS以外に広告しない
  - No-advertise:
    - 受領したルータ以降に広告しない
- Well-known ではないCommunity
  - 経路情報を受領したAS, ルータで解釈させ、何らかのポリシー付加を発生させる

## COMMUNITY(2)

- 一般的な利用法
  - New-format – 32ビットを16ビットずつに二分
    - 5511:1000
  - 上位 – ターゲットAS
  - 下位 – ターゲットASでの動作
- 例1: RFC1998 MCI(現SAVVIS)における実装例
  - 3561:70 そのプリフィクスにLocPref=70付与
  - 3561:80 そのプリフィクスにLocPref=80付与
  - . . . . .
  - そのASからの戻りトラヒックの制御に便利!

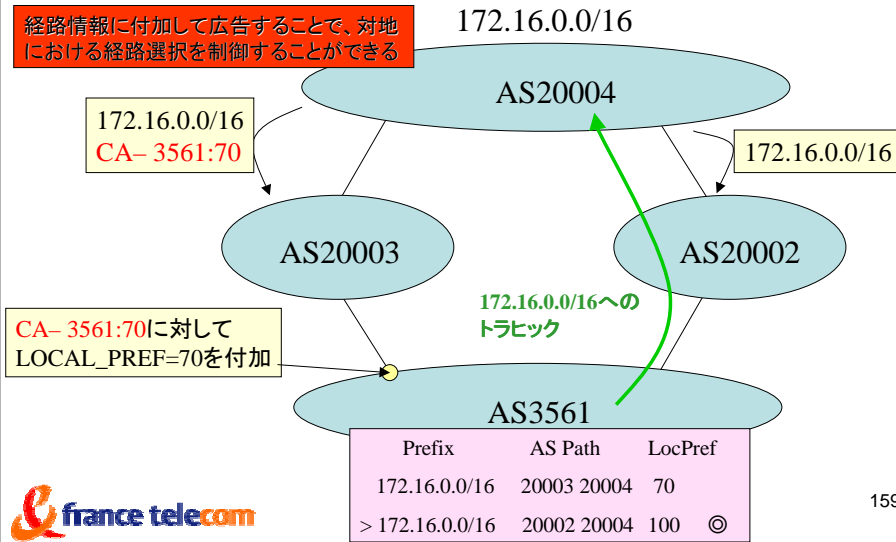
## COMMUNITY(3)

- AS5511 Opentransit Internetの場合
  - RIPEデータベースでコミュニティタグを公開

Community Definition	5511:540 North American Customers	Communities to control outbound routing announcements
5511:70 Local preference = 70	5511:541 Antilles	Community Definition
5511:80 Local preference = 80	5511:542 US & Canada East-Coast ( Chicago & D	5511:1000 Do not announce to US peers
5511:90 Local preference = 90	5511:543 US & Canada West-Coast	5511:1001 Prepend 5511 to US peers
Default = 100	5511:580 South American Customers	5511:1002 Prepend 5511 5511 to US peers
Community Definition	5511:582 Venezuela	5511:1500 Do not announce to AS1668 AOL/Time Warner
5511:5511 Community announced to customers or	5511:600 Asian Customers	5511:1501 Prepend 5511 to AS1668 AOL/Time Warner
5511:999 Routes received from customer	5511:601 Japan	5511:1502 Prepend 5511 5511 to AS1668 AOL/Time Warner
Communities to determine entry-point of customer	5511:602 Singapore	5511:1510 Do not announce to AS2914 NTT/Verio
Community Entry Point	5511:603 China	5511:1511 Prepend 5511 to AS2914 NTT/Verio
5511:500 European customers	Communities tagged on inbound routes:	5511:1512 Prepend 5511 5511 to AS2914 NTT/Verio
5511:501 Bagnolet	Community Definition	5511:1520 Do not announce to AS7018 AT&T/Worldnet
5511:502 Paris Pastourelle	5511:666 Route received from peering partner	5511:1521 Prepend 5511 to AS7018 AT&T/Worldnet
5511:503 Paris Bourse	5511:667 Route received from Sprintlink/AS1239	5511:1522 Prepend 5511 5511 to AS7018 AT&T/Worldnet
5511:504 Telehouse Paris	5511:700 Route received in US from peering	5511:1530 Do not announce to AS6461 Abovenet/MFN
5511:505 Aubervilliers	5511:710 Route received in Europe from peering	5511:1531 Prepend 5511 to AS6461 Abovenet/MFN
5511:506 Bercenay	5511:720 Route received in Asia from peering	5511:1532 Prepend 5511 5511 to AS6461 Abovenet/MFN
5511:506 Bercenay	5511:721 Route received in Asia from peering/res	5511:1540 Do not announce to AS6453 Teleglobe/BCE
...	...	5511:1541 Prepend 5511 to AS6453 Teleglobe/BCE
...	...	5511:1542 Prepend 5511 5511 to AS6453 Teleglobe/BCE
...	...	...

## COMMUNITYの利用方法

経路情報に付加して広告することで、対地における経路選択を制御することができる



## BGPの最適経路の決定プロセス

- 同一 prefixes の経路情報が複数あるとき、パス属性値に拠って最適方路を決定
  - 以下、cisco の例
    - 1. Local Preference が大きい
    - 2. AS\_PATH が短い
    - 3. MED が小さい
    - 4. IGP 上で Next-hop が近い (cost/metric)
    - 5. BGP のルータID が小さい



---

ご静聴ありがとうございました。

ISPバックボーンネットワークにおける経路制御設計  
～ 理論編 ～

フランステレコム日本研究所  
前村 昌紀

[maem@francetelecom.co.jp](mailto:maem@francetelecom.co.jp)