

# 本資料について

本資料は下記著書を基にして作成されたものです。著書の内容の正確さは保障できないため、正確な知識を求める方は原本を参照してください。

書籍名： とこトンやさしい IPルーティング

著者： 久米原 栄

発行者： 稲葉 俊夫

発行所： ソフトバンク パブリッシング株式会社

出版日： 2004年9月7日 初版第1刷発行

# IPルーティング

渡邊研究室

030432047 葛谷章一

# 概要

- 1章 ルーティング
- 2章 RIP
- 3章 IGRP
- 4章 OSPF
- 5章 EIGRP

# 1章 ルーティング

## ■ ルーティングとは

送信元から目的の宛先までの最適パスを探すタスクのことです。

## ■ なぜルーティングが必要なのか

異なるネットワークで接続される装置同士の通信は直接通信ができないために、どこに送信すればよいか分からなくなるので、ルーティングが必要である。

# ルータ

- ルータとは

ルーティングを行うデバイス

- 機能

パケットを受信すると、宛先アドレスを確認して、保有しているルーティングテーブルと比較して、どこに送るかを決定する

# ルーティングテーブル

- ルーティングテーブルとは

パケットの宛先アドレスを確認して最適ルートを判断基準となるもの

- 所有情報

宛先ネットワーク

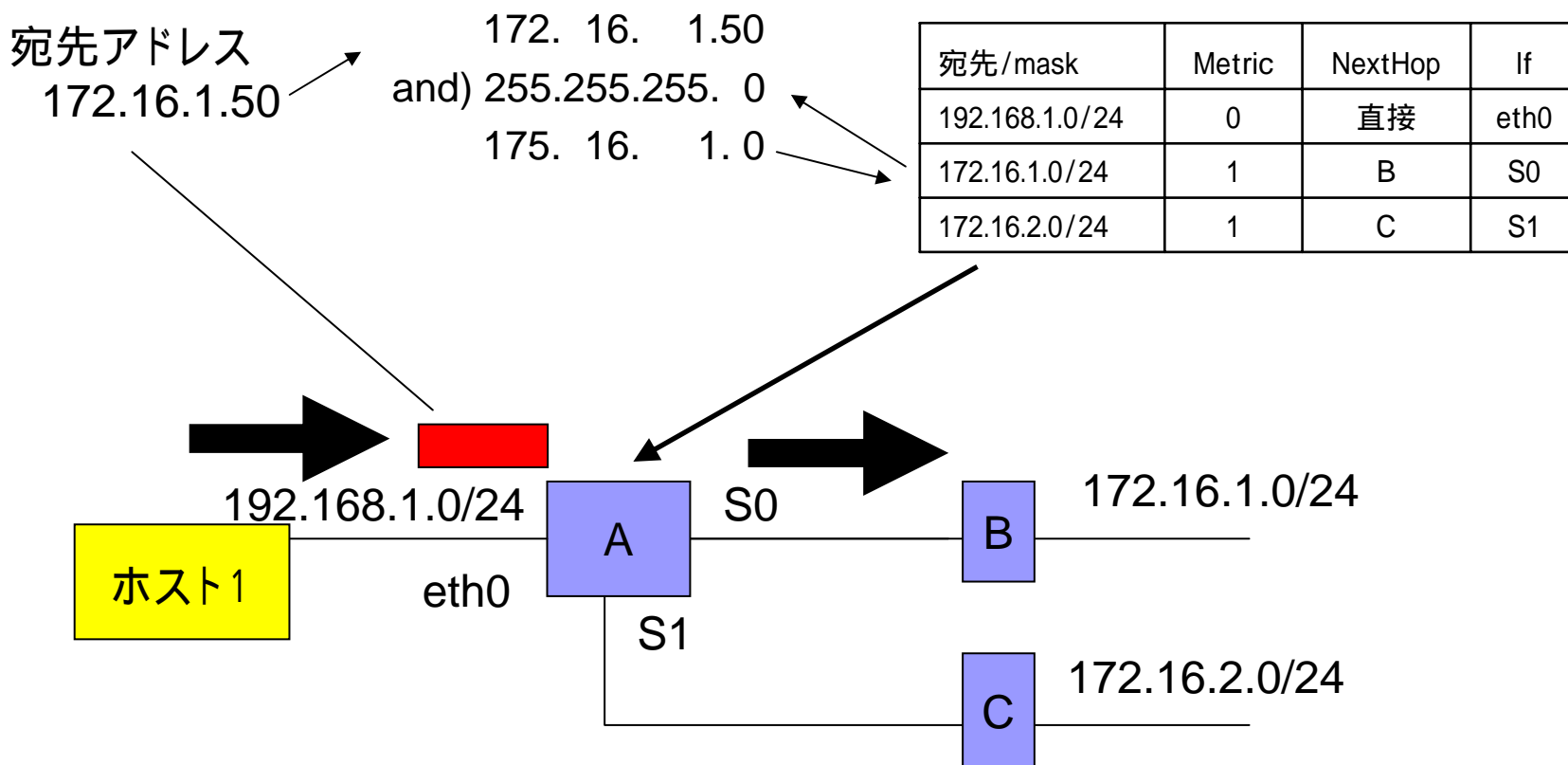
ネクストホップアドレス

メトリックなど

# ルーティング動作

- パケット宛先アドレスからサブネットワークマスクを利用してネットワークアドレスを割り出す。
- ルーティングテーブルの宛先ネットワークアドレスと処理対象パケットの宛先アドレスを比較する。
- ルートを見つけたらネクストホップルータにパケットを送信する

# ルーティング動作





# ルーティングテーブルの作り方

- **スタティックルーティング**  
ネットワーク管理者が手動でルートを設定してルーティング情報を固定する
- **ダイナミックルーティング**  
ルーティングプロトコル同士が自動的にルーティング情報を交換して作成する方法

# スタティックルーティング

## ■ 長所

帯域幅が狭い回線でもトラフィックが頻繁に発生しない。

ルータに負担がかからない

## ■ 短所

大規模ネットワークでは設定や管理が大変  
障害時に自動的にルートの変更が不可能

# ダイナミックルーティング

## ■ 長所

設定が簡単

障害に対して自動的に最適ルートを学習する

## ■ 短所

CPU、メモリにかかる負荷が大きい

ルーティングプロトコルに関する知識が必要

# ルーティングプロトコル

## ■ IGP

同一内AS内でルーティング情報を交換するプロトコル

例 RIP, IGRP, OSPFなど

## ■ EGP

異なるAS巻のルーティングを制御するプロトコル

例 EGP, BGPなど

ASは自律システムと言い、1つの統一的なポリシーで管理されているネットワーク同士を隔てるルータ

# ルーティングアルゴリズムの分類

## ■ ディスタンスベクタ型

宛先までにどのネクストホップを経由して、何ホップ数で行けるかが判断基準になる。

ルートの情報は隣接ルータから学習したもの

## ■ リンクステート型

ネットワークを小さく分割して管理する方式

各ルータは同じリンク情報のデータベースを持つ

# ディスタンスベクタ型の動作

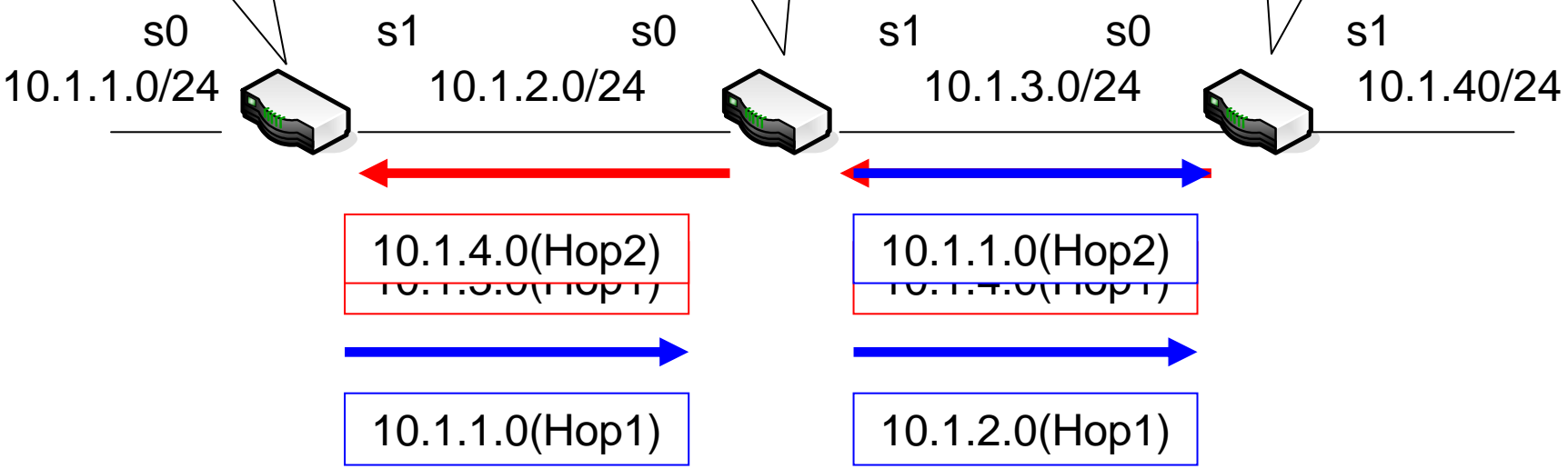
- ルーティングテーブルの交換を定期的に行う。
- ベルマンフォードアルゴリズムに従いルーティングテーブルを更新する。
  - ルーティングテーブル上にそのルートが存在しない場合は無条件で受け入れる。
  - ルーティングテーブル上にそのルートが存在する場合は、ネクストホップが違えばメトリックが優れた方を優先する。
  - ネクストホップが同じときは新しいものを採用する。

# ディスタンスベクタ型の動作

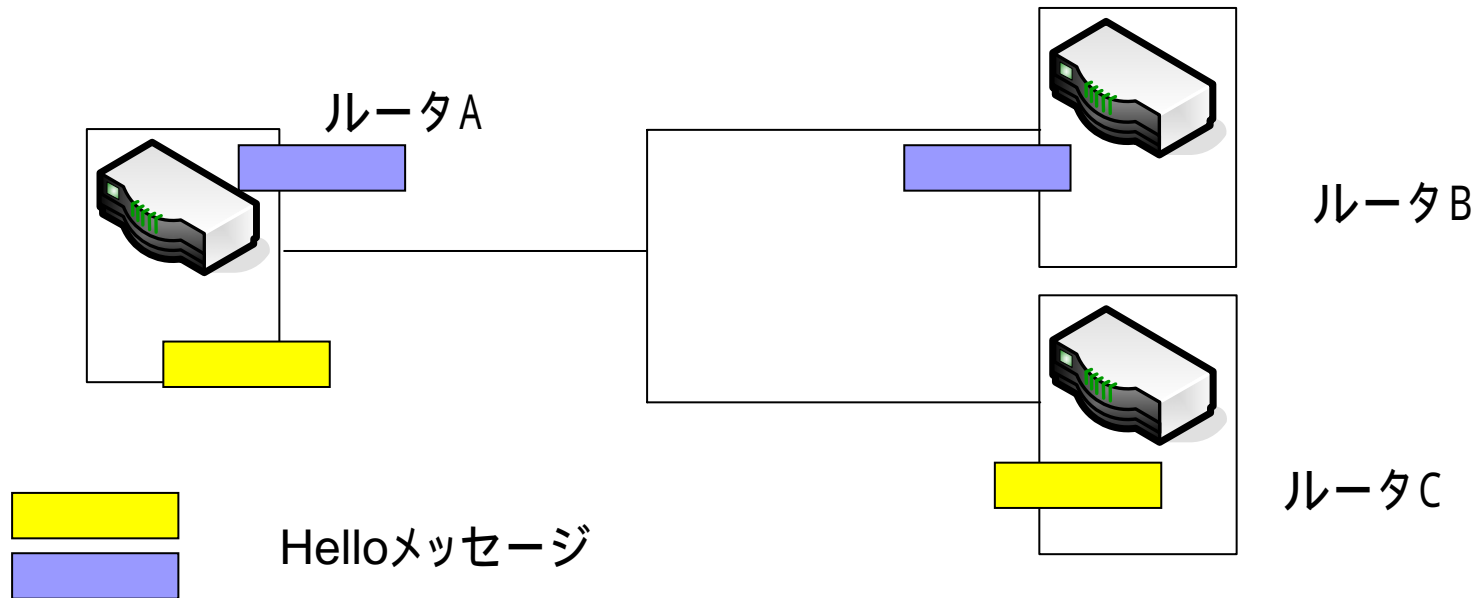
宛先	ホップ数	iff
10.1.1.0	00	s0
10.1.2.0	00	s1
10.1.3.0	1	s1
10.1.4.0	2	s1

宛先	ホップ数	iff
10.1.2.0	00	s0
10.1.3.0	00	s1
10.1.1.0	1	s0
10.1.4.0	1	s1

宛先	ホップ数	iff
10.1.3.0	00	s0
10.1.4.0	00	s1
10.1.2.0	11	s0
10.1.1.0	2	s0



# リンクステート型の動作



隣接ルータにHelloメッセージを送信して、隣接関係を確立する。ネットワーク上で何か変更があったときに差分アップデートを行う。



# 2章 RIP

## ■ RIPとは

最適ルートの選択基準であるメトリックには、ホップ数を使用して目的地まで最も少ないホップ数で最適ルートを探索して、ルーティングテーブルを作成するディスタンスベクタ型ルーティングプロトコル

# RIPの動作

- 各ルータは自分が保持するルート情報をネットワーク全体に定期的に送信する。
- 各ルータは受信した情報から、ルーティングテーブルを更新するかをベルマンフォードというアルゴリズムにより判断する。
- ルーティングテーブルを更新する時に、ホップ数を + 1 にする。

# RIPの問題点

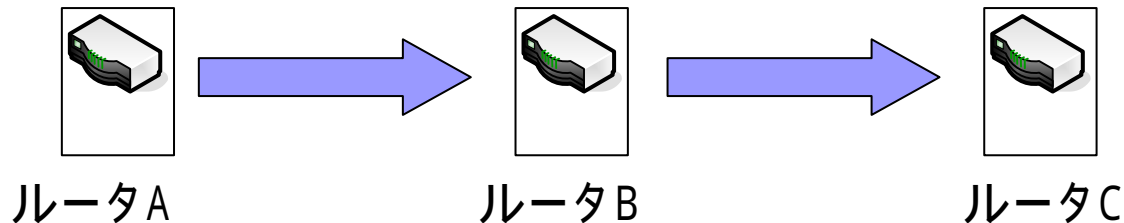
- 問題1

  - コンバージェンスが遅い

- 問題2

  - ルーティングループの発生

# 問題1の原因



- RIPの場合は30秒ごとに定期更新する
- 各ルータはばらばらのタイマで定期更新する
- ルータが3つの場合は最短で60秒かかる



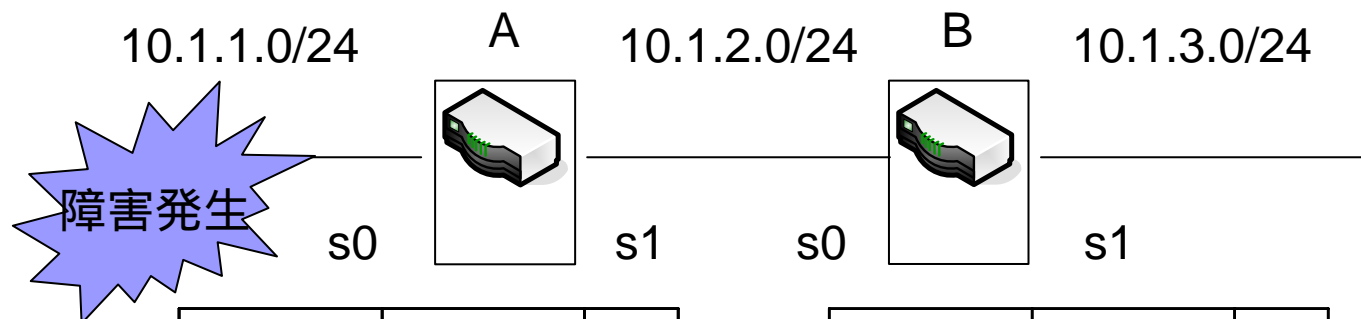
コンバージェンスが遅い

# 問題1 対策

## ■ トリガアップデートを使用する

ルート情報が変化した時に変化したルートに関するアップデートを定期更新の時間に達する前に素早く送信する処理

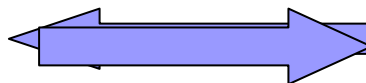
# 問題2の原因



宛先	ホップ数	if
10.1.1.0	down	s0
10.1.2.0	0	s1
10.1.3.0	1	s1

宛先	ホップ数	if
10.1.2.0	0	s0
10.1.3.0	0	s1
10.1.1.0	3	s0

10.1.1.0 Hop 3



ルートがダウンすると定期更新のたびにホップ数が  
順次増加するルーティンググループの発生

# 問題2の対策

## ■ スプリットホライズン

ルート情報を教えてくれた隣接ルータには、そのルート情報を教え返さない

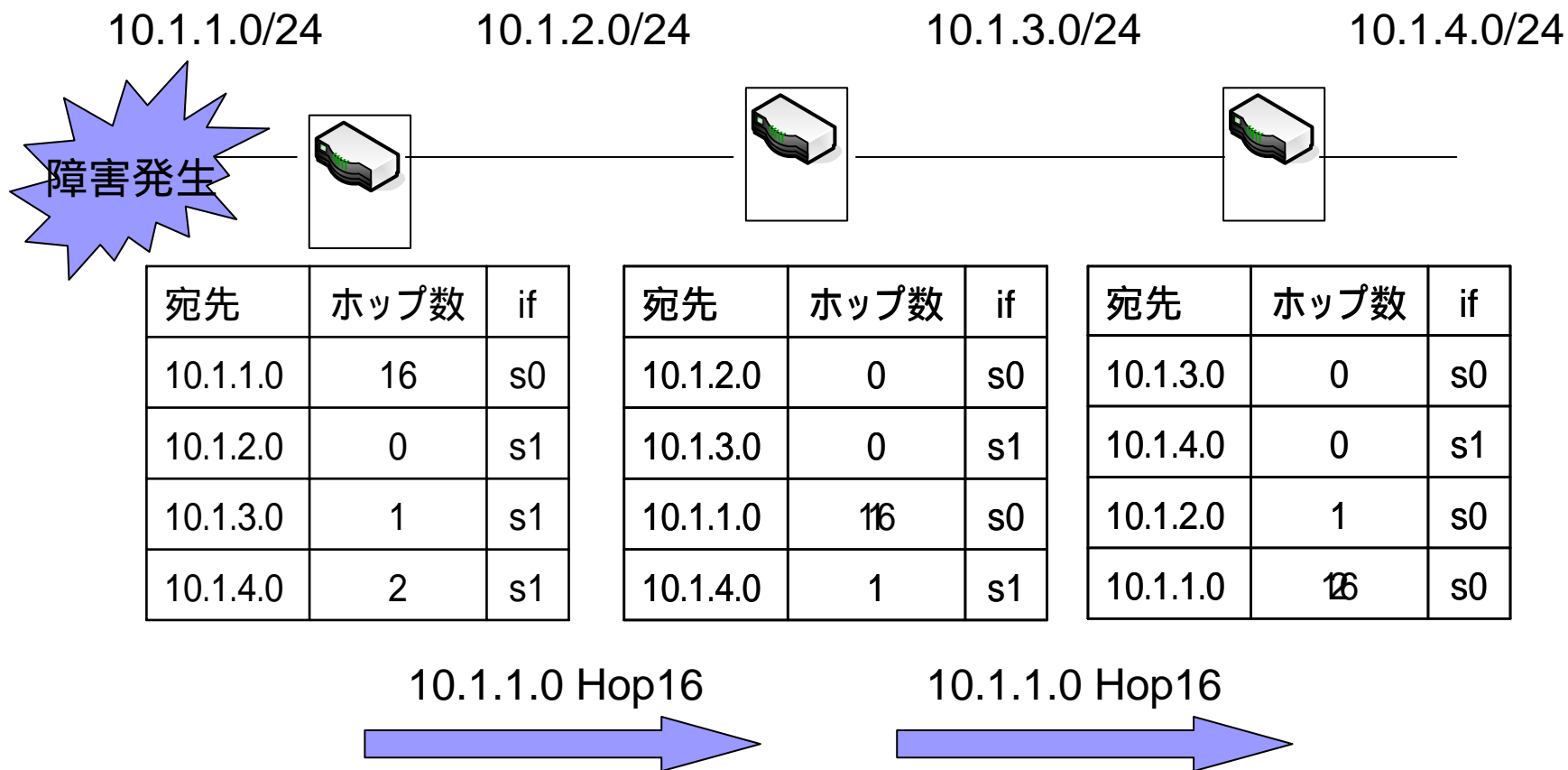
## ■ ルートポイズニング

ダウンしたルートを使用不可能にするために、メトリック値を16にする

## ■ ポイズンリバーブ

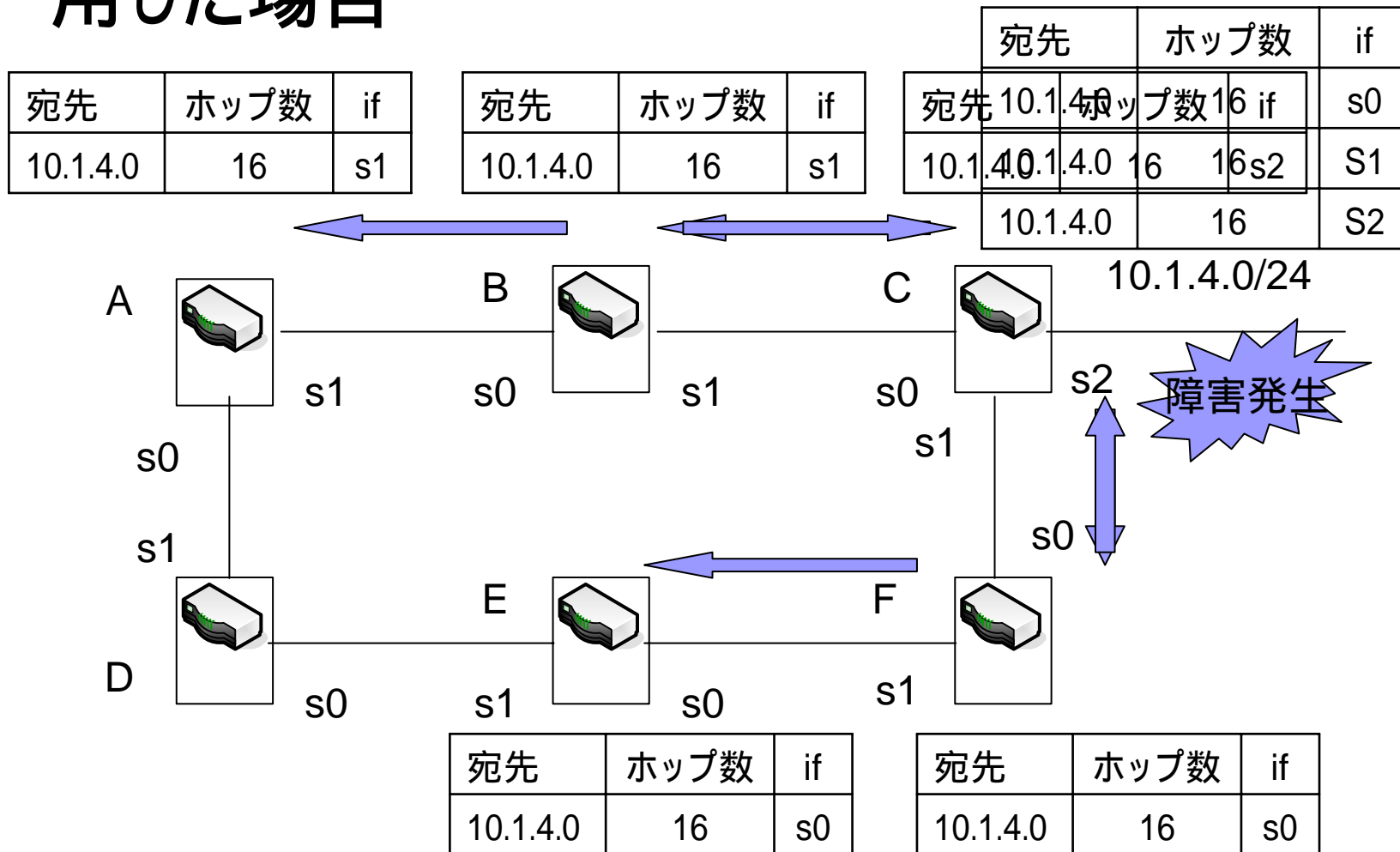
ルートがダウンした情報を受信したら、すべてのインターフェースからメトリック値を16にして送信する

# ルートポイズニングとスプリットホライズンを併用した場合





# ポイズンリバーブースとスプリットホライズンを併用した場合



# 3章 IGRP

## ■ IGRPとは

RIPの改良版で、複合メトリックという考え方を採用して、「帯域幅」と「遅延」を基準にして、最適ルートを選択するディスタンスベクタ型ルーティングプロトコル

また、メトリックとしてホップ数を採用しないが、ホップ数自体はカウントする。

# IGRPの動作

- RIPと同じで、各ルータが保有するルーティング情報を定期更新する
- メトリックは、ホップ数ではなくルート中の遅延の累積と最小帯域幅を利用して求める。

デフォルトメトリック =  $10000000 \div \text{帯域幅} + \text{遅延} \div 10$

# RIPとIGRPの比較

- RIPと同じディスタンスベクタ型ルーティング  
プロトコルなので、RIPと同じ問題を持っています。また、対策も同じです。
  - コンバージェンス時間が遅い
  - ルーティングループの発生の可能性あり
- ルータにAS番号の割り当てて、同じAS番号  
を持つルータ同士だけでルーティングテー  
ブルの情報を交換する。

# 4章 OSPF

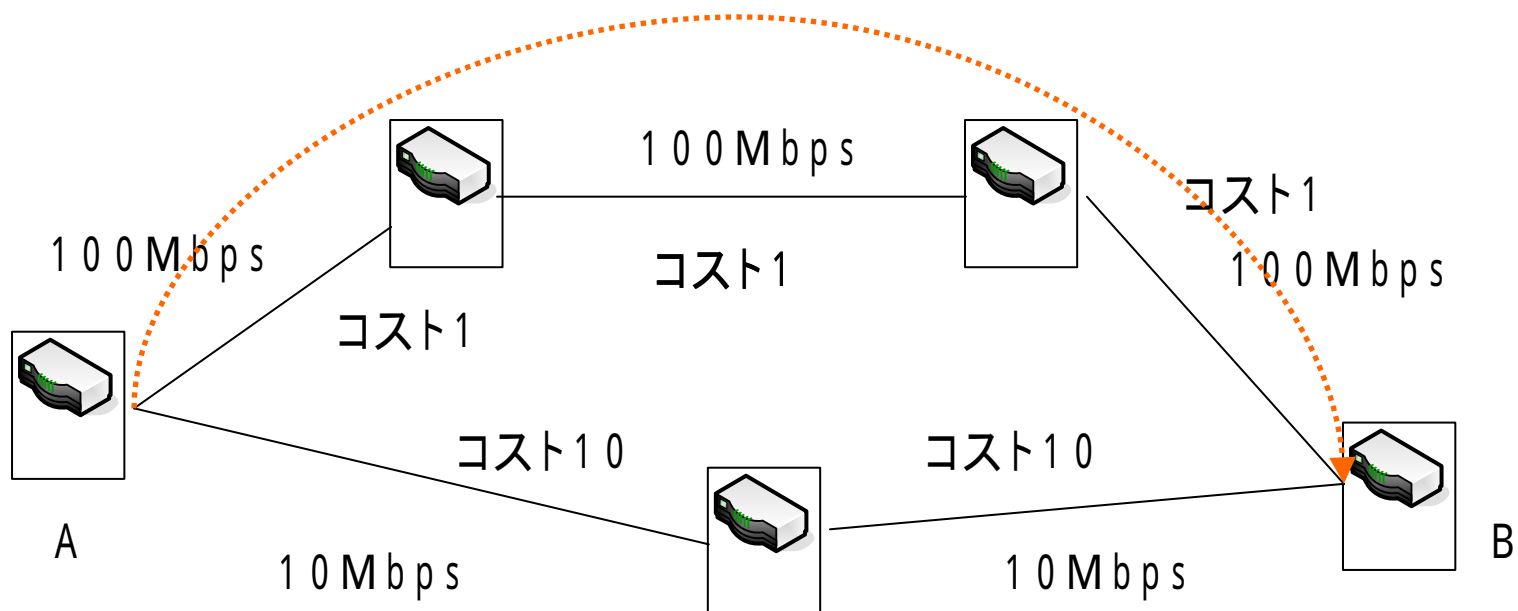
## ■ OSPFとは

RIPなどのディスタンスベクタ型のルーティングプロトコルの欠点を補い、大規模なネットワークに対応できるように考えられたリンクステート型のルーティングプロトコル

# OSPFの特徴

- ホップ数がRIPの場合は、15が限界でIGRPは最大255までが限界であったが、OSPFは制限がないので、大規模なネットワークに対応
- RIPの場合はメトリックとして、ホップ数を使用していたが、OSPFはコストを採用して最適ルートを選択
- ルーティングテーブル以外の情報を保持

# コストを利用した最適ルートの実立



- 大きな帯域幅の100 Mbpsをコスト1にし、10 Mbpsをコスト10にする
- 上の2つのルータを経由すると合計コスト3
- 下の1つのルータを経由すると合計コスト20

# ルーティングテーブル以外の情報を保持

- ルータ同士が共通のLSA (リンクステートデータベース) を持つことにより、隣接関係が確立する。
- しかし大規模ネットワークになると
  - ネットワーク上に流れるLSAの増大
  - リンクステートデータベースが大きい
  - ルーティングテーブルが大きい
  - ネットワーク内の変更による影響が大きい



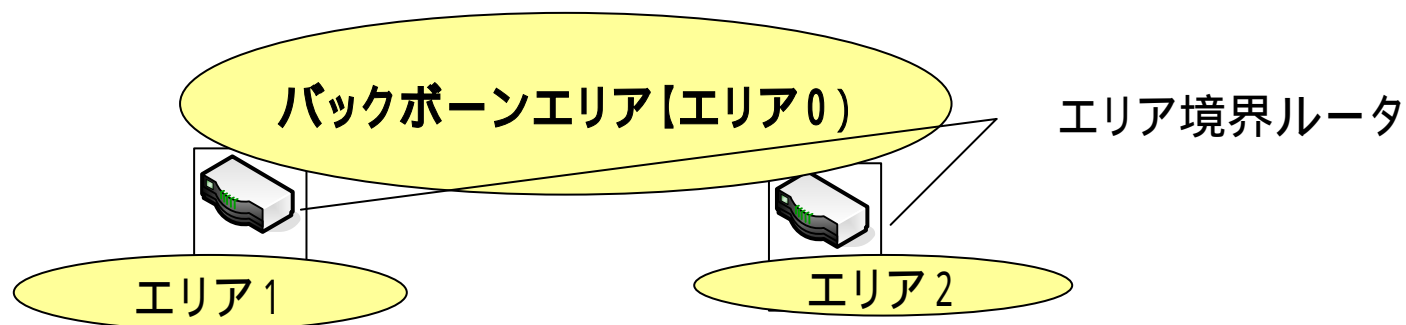
エリアの概念を導入



# エリアの概念

## ■ エリアとは

同一のリンクステートデータベースを持つルータの集合

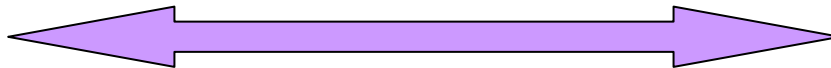
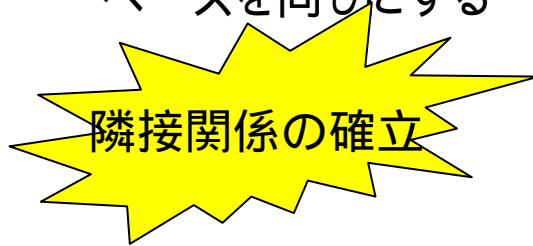


- ネットワークが分割し、エリア内だけのルーティングテーブルやLSAの交換だけで行うので、LSAの大きさが小さくなる。
- 他のエリア内にあるネットワークの変更があってもルーティングテーブルの再計算をする必要がない。

# OSPFの流れ



LSRから要求された情報を交換して、双方向状態の確立を示すデータベースを同じにする



定期的にHelloパケットを送信して、隣接関係を維持する

# 5章 EIGRP

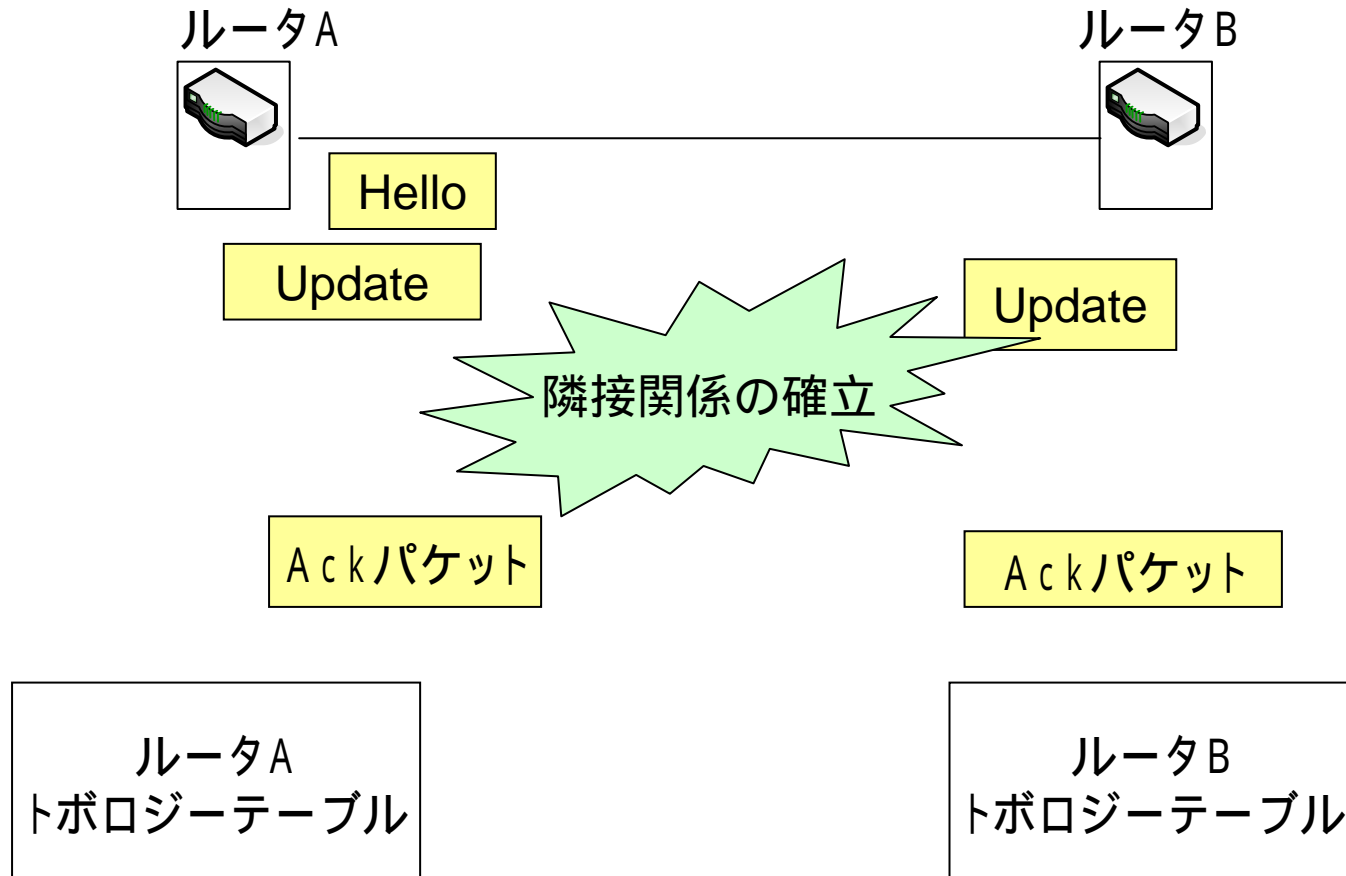
## ■ EIGRPとは

ディスタンスベクタ型を基本として、リンクステート型のいいところを取り入れたルーティングプロトコル

# EIGRPの特徴

- ルーティング情報として、ディスタンスベクタ型の考え方のルート情報を交換する
- リンクステート型のように、隣接ルータとの間に隣接関係を構築して、ルーティング情報を隣接関係のあるルータ同士で交換する
- ネットワークの変更時に変更部分である差分情報だけを送信する(リンクステート型の特徴)

# EIGRPの動作



# トポロジテーブル(1)

- トポロジテーブルとは  
隣接ルータからアップデートで教えてもらったすべてのルート情報を記載するテーブル
- トポロジテーブルに含まれる情報
  - サクサセ
  - フィージブルサクセサ
  - FD (フィージブルディスタンス)
  - AD (アドバタイズドディスタンス) など

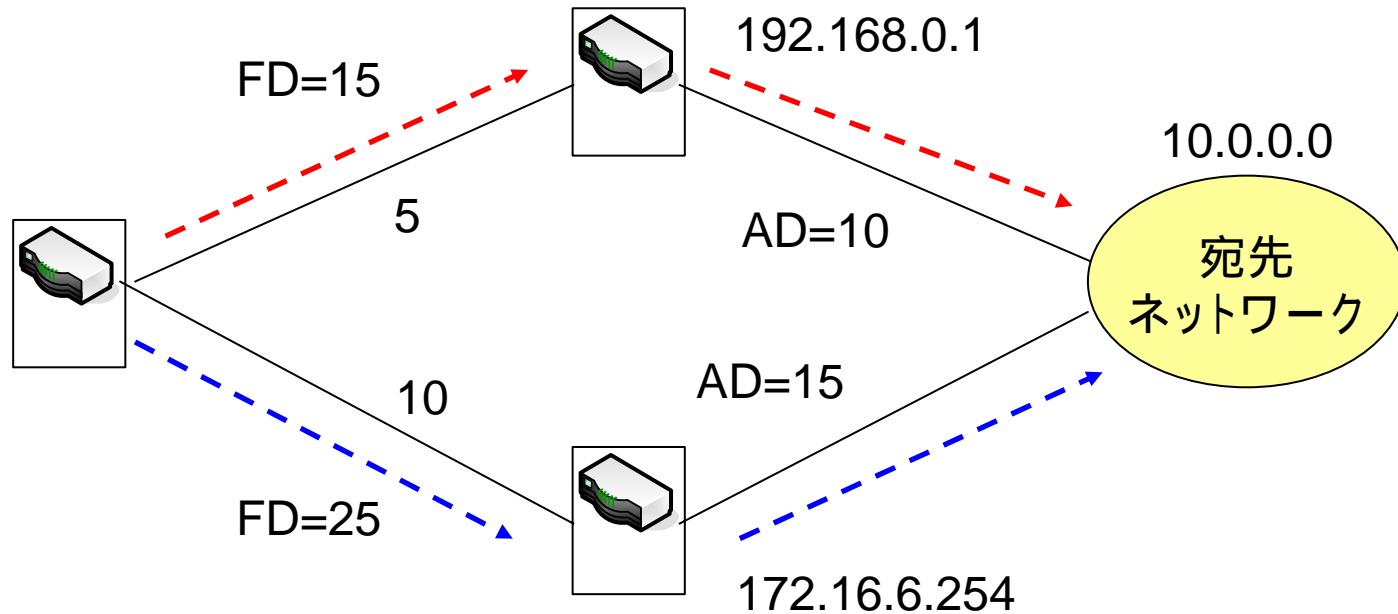
# トポロジテーブル(2)

## ■ トポロジテーブルの構成

隣接ルータが個々の宛先用にアドバダイズしてきたADと、当該ルータが個々の宛先に到達するために使用するFDで構成

## ■ ルーティングテーブルの作成方法

トポロジテーブルを用いて、最小FDのルートを最適ルートにして、ルーティングテーブルに書き込む



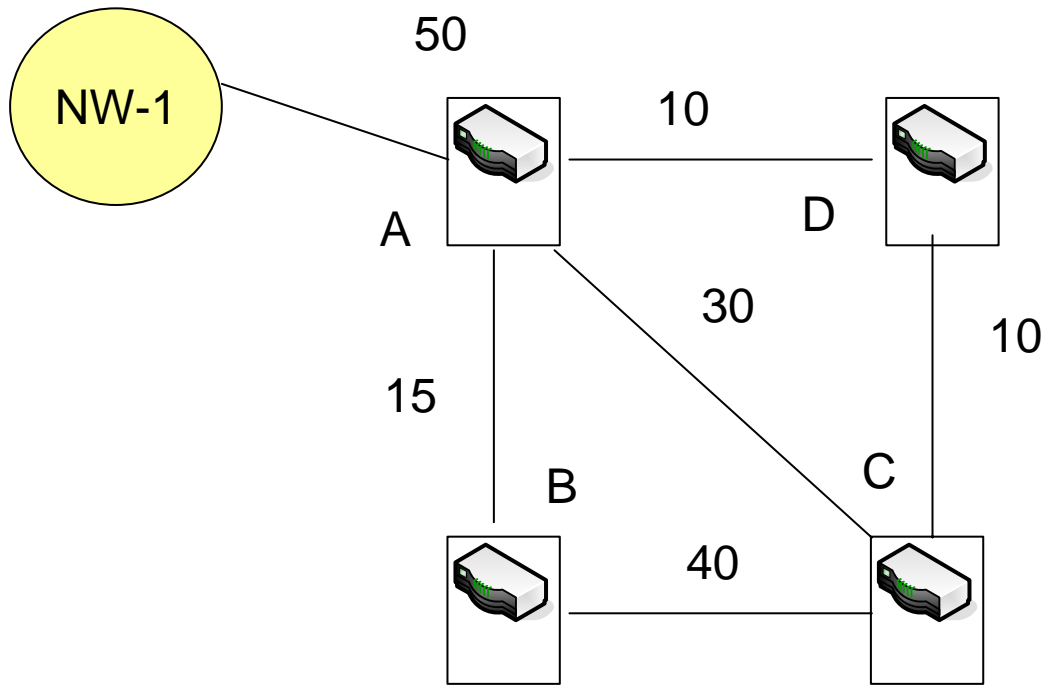
宛先ネットワーク	隣接ルータ	FD	AD
10.0.0.0	192.168.0.1	15	10
	172.16.6.254	25	15

トポロジテーブル

宛先ネットワーク	ネクストホップ	メトリック
10.0.0.0	192.168.0.1	15

ルーティングテーブルの作成





- 上の図のように複数のルートが存在する時に、隣接ルータから受信する情報のうちのどれをトポロジテーブルに保存するかを決定するために、DUAL (拡散アップデートアルゴリズム)を用いる。

# DUAL (拡散アップデートアルゴリズム)

## ■ DUAL動作

あるルータからその隣接ルータまでのメトリックを加算したFDを使用して、宛先へのパスを評価する。

最小のFDを持つルートを最適ルートにする。

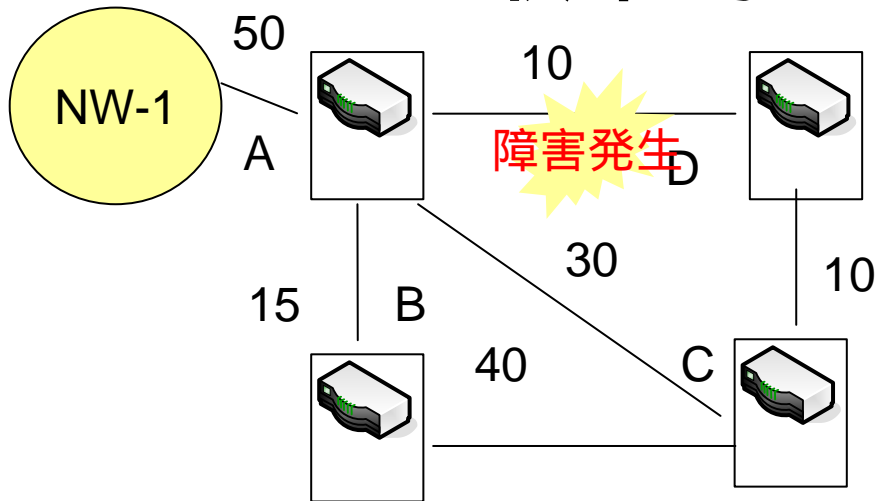
最適ルートのネクストホップをサクセサとする。

最適ルートがダウンした場合に備えて、バックアップルートを選択し、ネクストホップのフィージブルサクセサを選択しておく。

フィージブルサクセサになるための条件

最適ルートのFD > フィージブルサクセサのAD

# ネットワーク収束時のトポロジーテーブル



宛先ネットワーク	隣接ルータ	FD	AD	トポロジー
NW - 1	A	80	50	FS
	B	105	65	FS
	D	70	60	サクセサ

ルータCのトポロジーテーブル

宛先ネットワーク	隣接ルータ	FD	AD	トポロジー
NW - 1	A	65	50	サクセサ
	C	110	70	

ルータBのトポロジーテーブル

宛先ネットワーク	隣接ルータ	FD	AD	トポロジー
NW - 1	A	65	50	サクセサ

ルータDのトポロジーテーブル

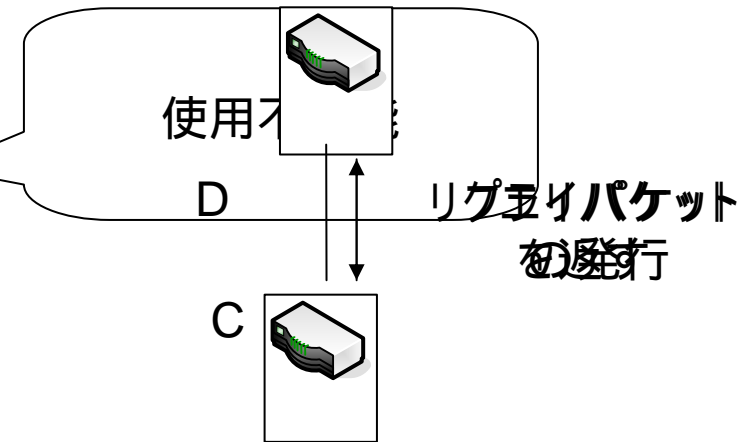
# リンク障害時のトポロジーテーブル

ルータDのトポロジーテーブル

宛先ネットワーク	隣接ルータ	FD	AD	トポロジー
NW-1	A	70	80	秒々サセ
	A	65	50	秒々サセ
	C			(Query)

ルータCのトポロジーテーブル

宛先ネットワーク	隣接ルータ	FD	AD	トポロジー
NW - 1	A	80	50	秒々サセ
	B	105	65	FS
	D	70	60	秒々サセ



最適ルートがダウンしたので残りからFDが最小のフェージブル秒々サセを秒々サセに昇格させる。

終わり

# RIPのタイマ

- ルータは隣接ルータから受信したルータ情報の信頼性を確保するために、そのルータ情報の有効期限を保持し、管理するために下記の4つのタイマを用います。
  - ・アップデートタイマ
  - ・無効タイマ
  - ・ホールドダウンタイマ
  - ・フラッシュタイマ

## ● アップデートタイマ

ルート更新情報の送信の間隔を計るために使用。ルータはアップデートを送信する際に、そのアップデートに含まれるルートのタイマを「0」にします。決まった値になるとルート更新情報を送信します。

## ● 無効タイマ

隣接ルータからルート情報を受信するたびに、以前受信したルート情報がアップデートに含まれているか確認して、確認できたらタイマを「0」にします。

確認できずにタイマが上限のアップデートタイマの6倍(180秒)に達するとホールドダウンタイマが起動します。

## ● ホールドダウンタイム

無効タイムが上限に達したときに起動し、ルートがダウンしたと疑い、そのルートが無効になったことをすべてのルータが認識できるようにします。

この間は、ダウンしたと思われるルートに関して、メトリックが劣るか同等のものは無視します。

メトリックが勝ったものを受信した場合は、新しいルートが発見されたものとして、ホールドダウンタイムを「0」にします。

## ● フラッシュタイム

無効タイムと同時に起動して240秒に達すると、ルートのルーティングテーブルを削除します。しかし、ダウンしたと疑われていたルートが復活したことを知らせるトリガアップデートを受け取ると、タイムを「0」にします。