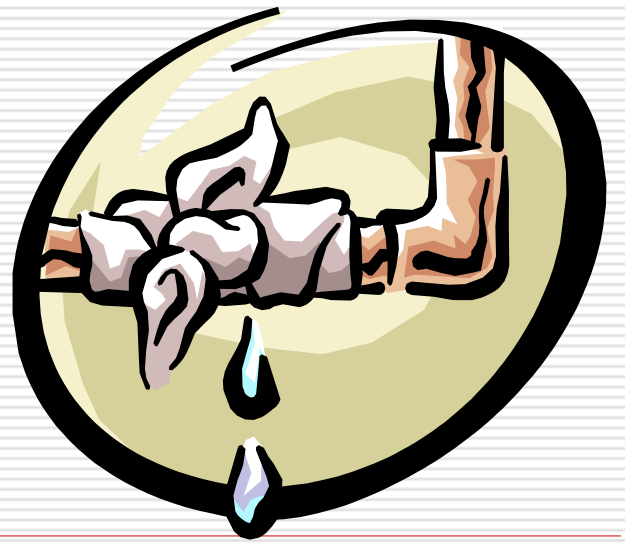
幸せなパケット転送　○○編

Yoshida Tomoya – yoshida@ocn.ad.jp
1E7D 79AD C610 B5F2 A94E 7FF4 F4AC A722 329C 3DE8

Matsuzaki Yoshinobu - maz@iij.ad.jp
8E 9C EC 04 87 6B B5 0E 1B 6D 46 3B CB F7 72 CE

パケットを届けよう！

- ペイロードが何だろうが気にしない
- sourceからdestinationへ
- Best Effort Network



ルータとパケット転送

- ルータは、IPパケットを隣接の機器に投げる
- 判断材料は
  - IPパケットの宛先IPアドレス
  - 自身が持っている経路情報
- 経路情報は、再帰的に計算する必要がある
- Equal Cost Multi Pathの事も考えなきゃね

RIBだけで頑張るには

- 毎回まじめに考えて、頑張って転送する
- flowの状態を持たないので、ECMPではpacket毎に分散させる場合が多い
- 多くはＣＰＵの性能に依存するので、おのずと限界が出てくる

On demand cacheの登場

- インターネットでは再帰経路は驚くほど多い
  - ＢＧＰのfull-routeは１４万ぐらい
  - これをパケットがくる度に計算すると負荷が無視できない
- ぢゃあ、一度パケット転送で学習したら、しばらくそれを使いまわせばいいぢゃん
- ECMPの分散はCacheの種類によって扱いが違う
  - destination IPアドレス or prefixのhash
  - source/destination IPアドレス＆portのhash

On demand cacheの泣き所

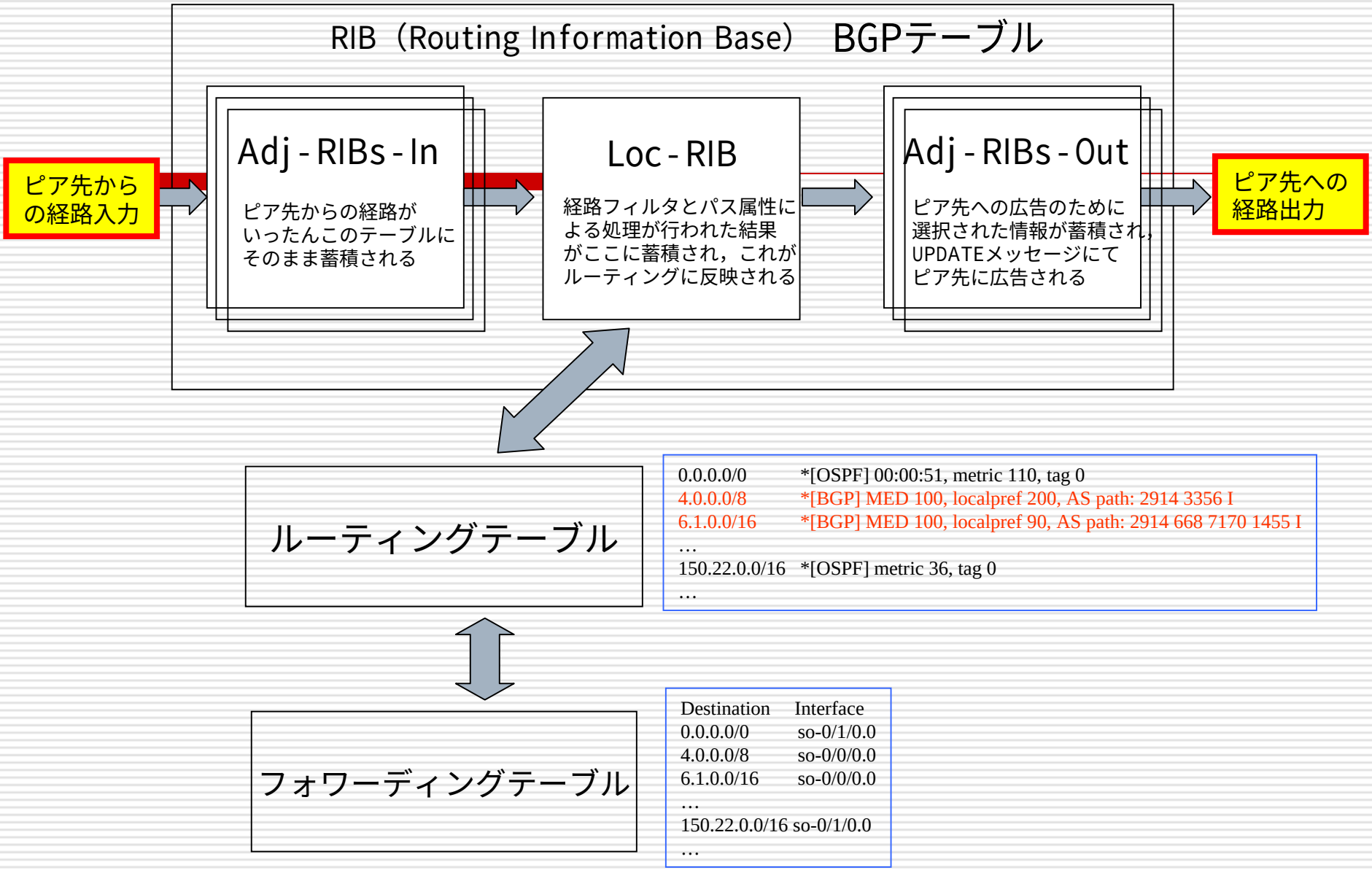
- 最初はまじめに考えないといけない
  - cacheに載ってないパケットがいっぱいくるとつらい
- 経路変化にどうやって追従しよう
  - 個別にCacheをagingする？
- Cacheの量が増えたらどうしよう
  - Tableサイズの制限？
  - あふれたらどうしよう？

FIBの登場

- もうメモリはケチらない
- 再帰計算の必要な経路とかを予め全部計算して、テーブルにして持っちゃえ
  - nexthop = fib(destination) の一発で解決
- ECMPは、nexthopの所で工夫
  - source、destinationのIPアドレス、portのhashを利用する場合が多い
  - per dest / per flow

FIBの泣き所

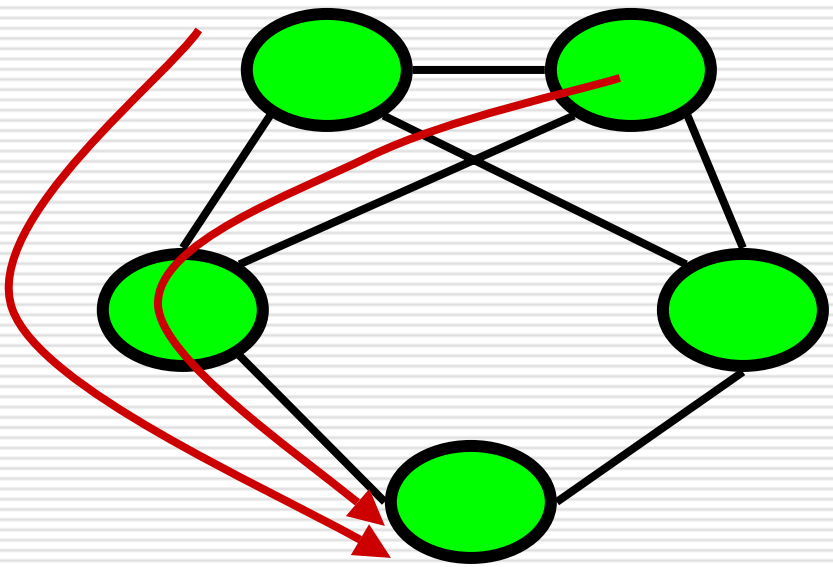
- テーブルの更新には工夫が必要
  - お願いだから、テーブル全体のreload時にfowardingが停止するのは止めて欲しい
- 箱の中でRIBとの一貫性が重要
  - RIBと分離されてるので、RIBと非同期でもFIBだけでforwardingはつづけれちゃう


RIB（Routing Information Base） BGPテーブル
ピア先から
の経路入力
Adj - RIBs - In
ピア先からの経路が
いったんこのテーブルに
そのまま蓄積される
Loc - RIB
経路フィルタとパス属性に
よる処理が行われた結果
がここに蓄積され，これが
ルーティングに反映される
Adj - RIBs - Out
ピア先への広告のために
選択された情報が蓄積され，
UPDATEメッセージにて
ピア先に広告される
ピア先への
経路出力
ルーティングテーブル
0.0.0.0/0 *[OSPF] 00:00:51, metric 110, tag 0
4.0.0.0/8 *[BGP] MED 100, localpref 200, AS path: 2914 3356 I
6.1.0.0/16 *[BGP] MED 100, localpref 90, AS path: 2914 668 7170 1455 I
…
150.22.0.0/16 *[OSPF] metric 36, tag 0
…
フォワーディングテーブル
Destination Interface
0.0.0.0/0 so-0/1/0.0
4.0.0.0/8 so-0/0/0.0
6.1.0.0/16 so-0/0/0.0
…
150.22.0.0/16 so-0/1/0.0
…

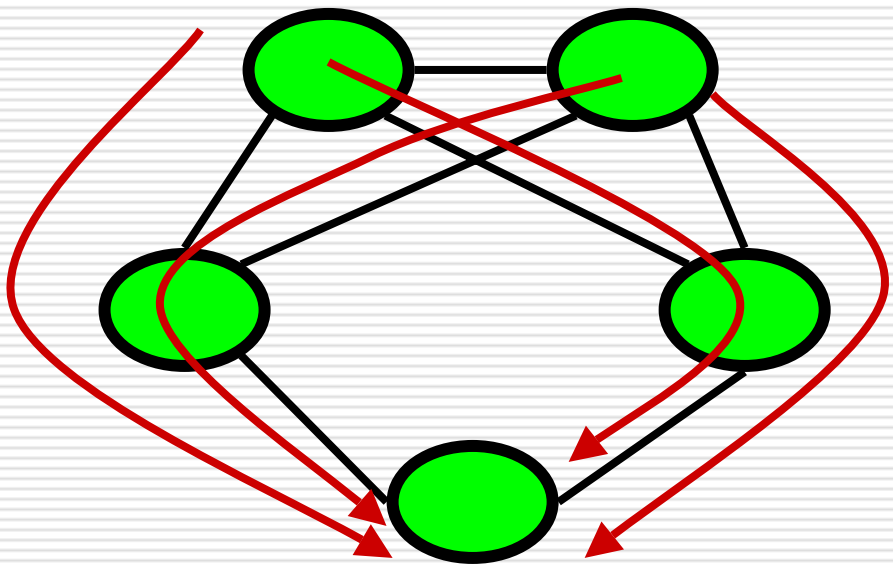
ECMP

- 経路数が少ない場合には、flow baseが幸せ

Per dest

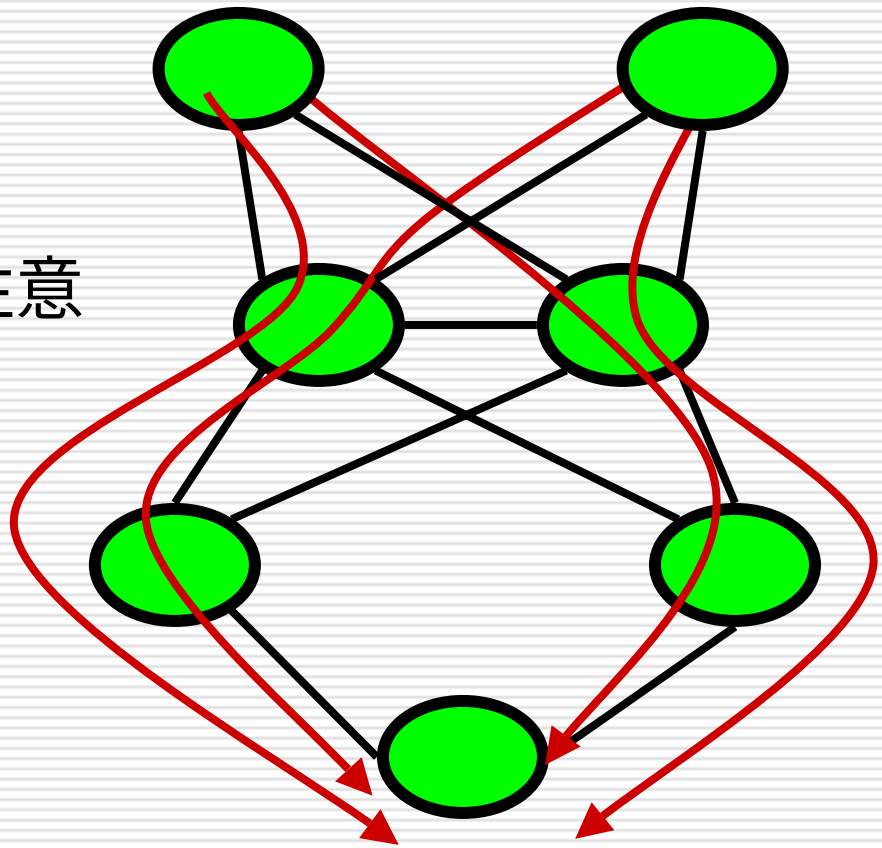


Per flow



ECMPの多段

- こんなことも・・・
- 各ルータのhashが
  重ならないように注意



分業の進むルータ内部

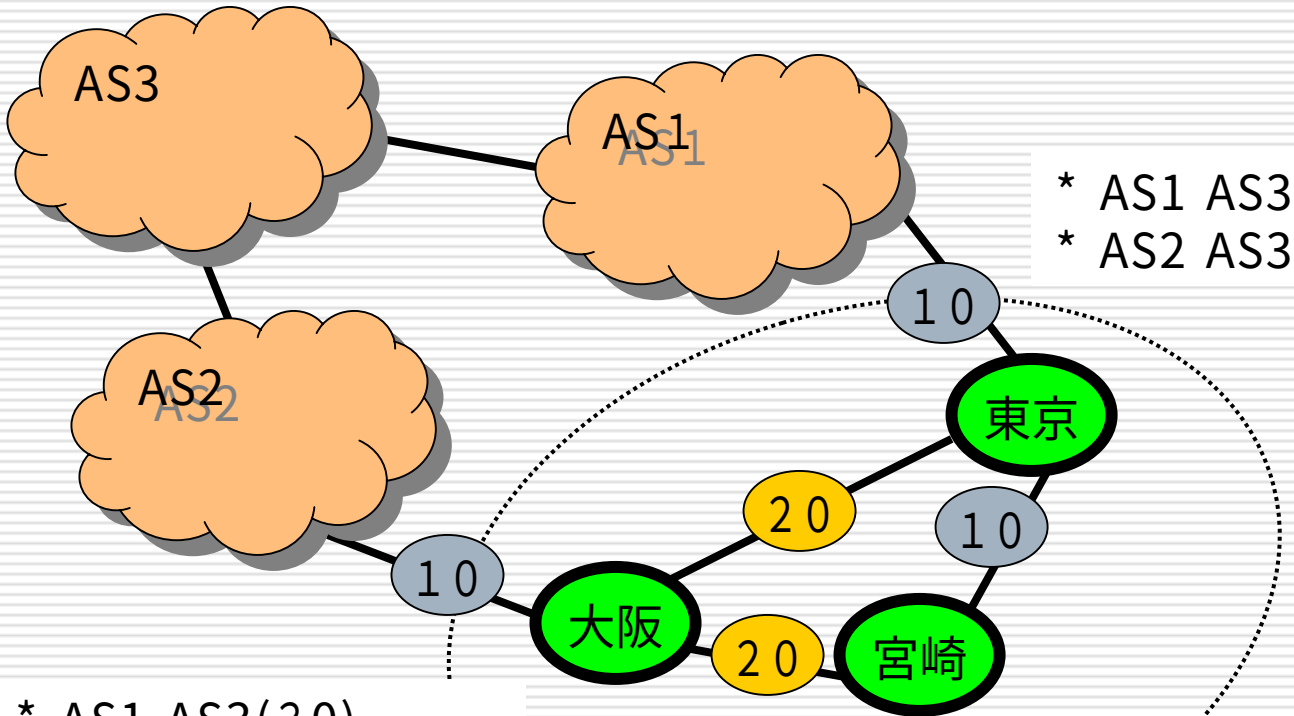
- ルーティングの制御
  - BGP、IS-IS、OSPF
  - RIB、FIBの構築
- パケット転送の制御
  - FIBに従って、パケットの転送

ルータはそれぞれ違うことを考えてる

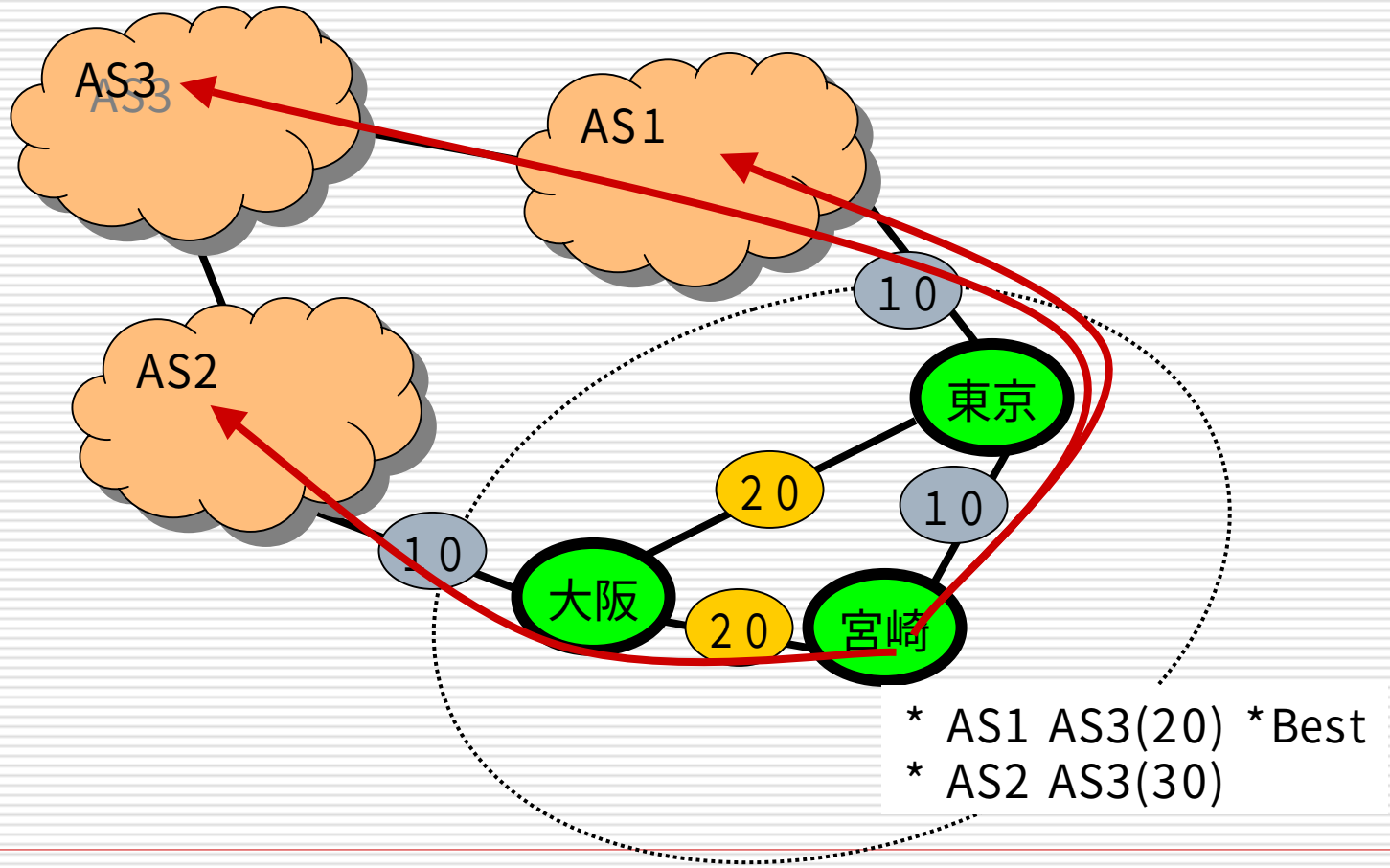
- 一般的に、ルータそれぞれで
  - OSPFのSPTは違う
  - BGPのBEST経路は違う
- 当然、ルータ毎にFIBは違う

- ルータ間で矛盾のないFIBを構築するためにルーティングプロトコルが頑張ってる

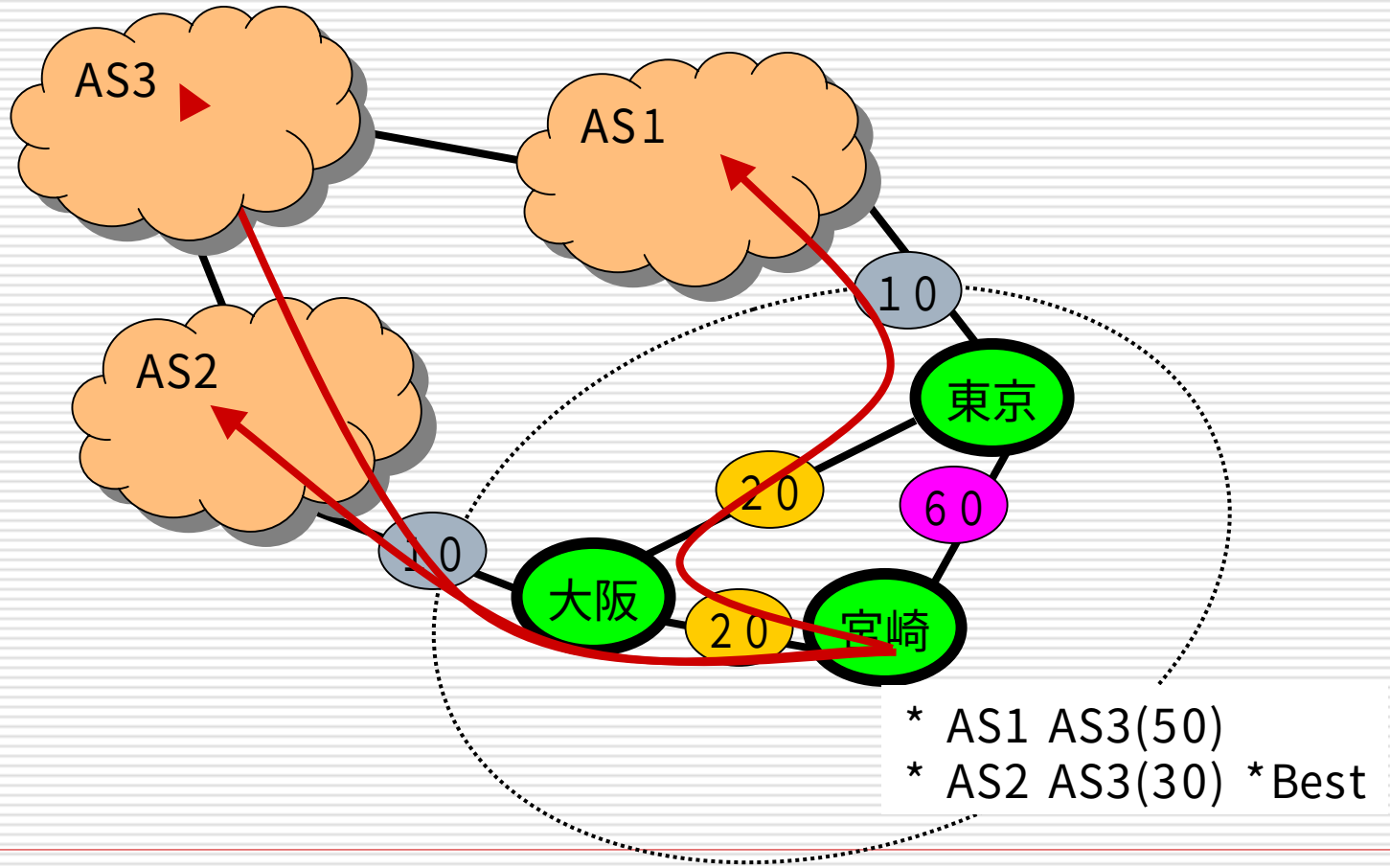
POP毎にBest経路は異なる



宮崎からのトラヒックに注目すると・・・



IGP Costの変更で、Best経路が変わる



矛盾のないFIBを維持しよう

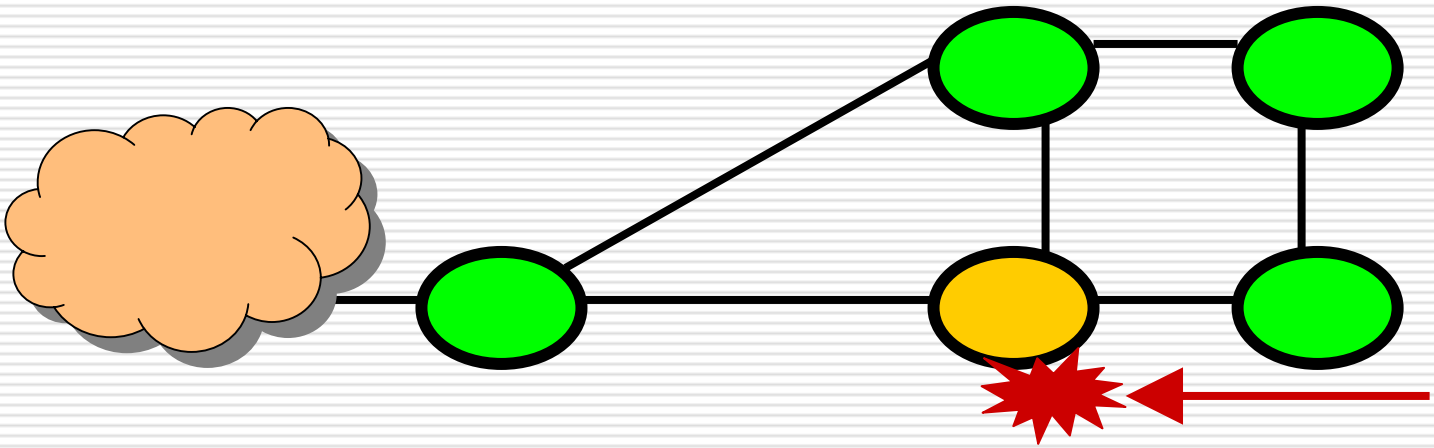
- 日々起こる経路の変動
  - ネットワーク構成の変更
  - 障害
- sourceからdestinationまで、矛盾さえなければパケットは届く！

例えば、AS内でのパケット転送

- バックボーンのルータはfull-route運用
- BGPのnexthopはIGPが運んでる
- 一般的にルータが起動する際、
  - BGPの収束時間 > IGPの収束時間

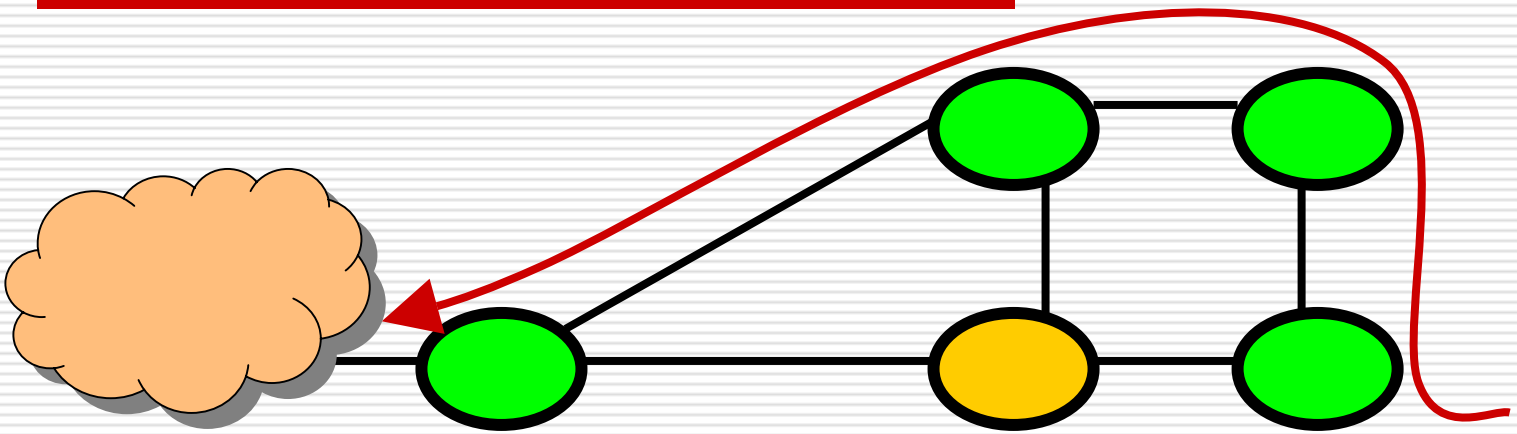
- 予期しない再起動では、問題が。。。

黄色のルータが再起動後、IGPとBGPの経路収束までに・・・



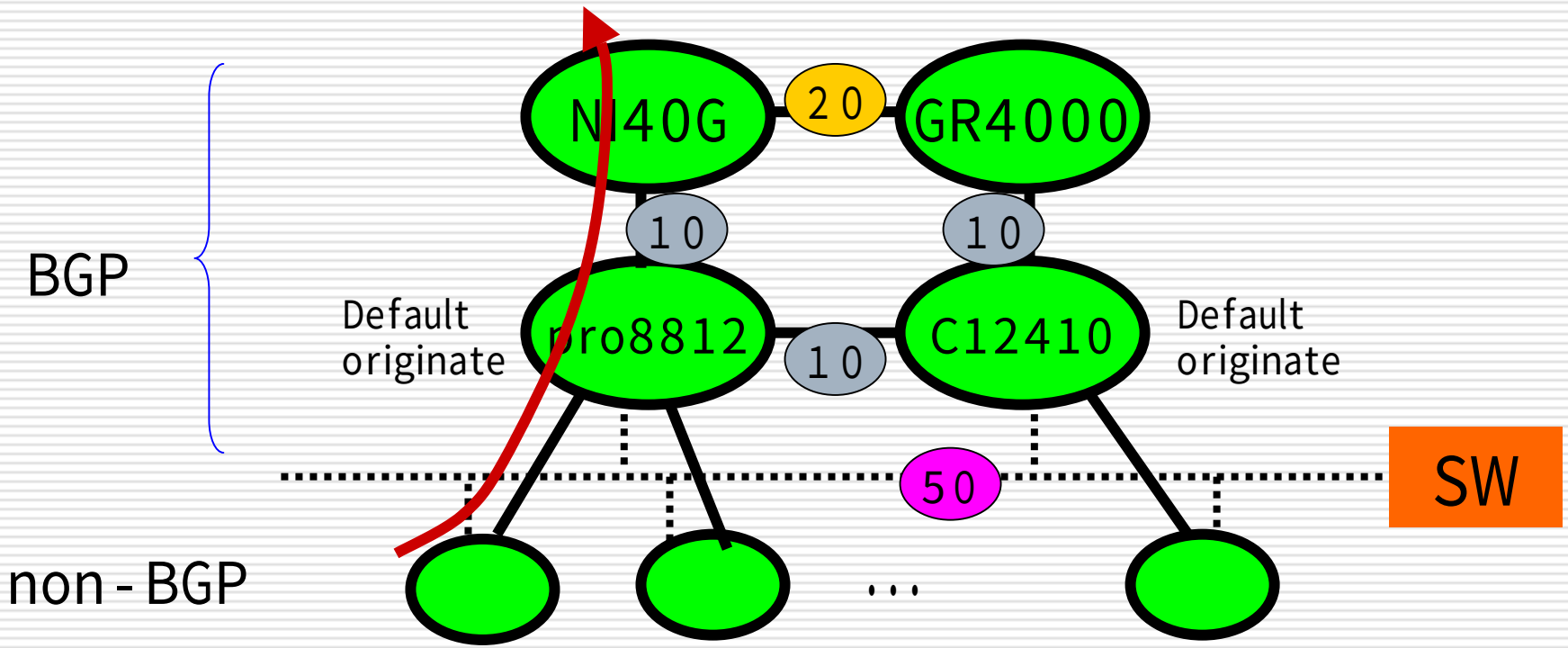
- トラヒックを経由するルータでBGPの経路収束が終わる前に、IGPではネクストホップがそのルータの先に見えてしまう

wait-for-bgp

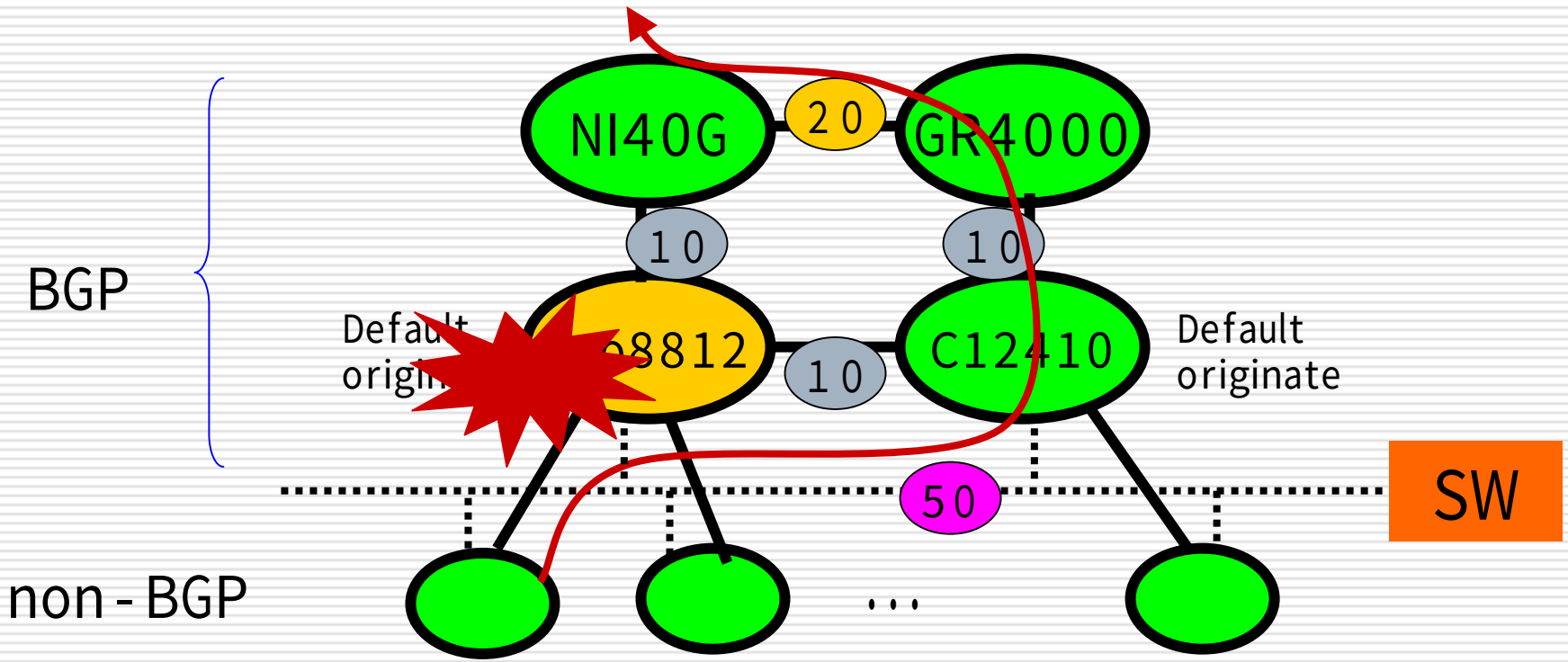


- 再起動したルータでBGPが収束するまで、IGP的に迂回しておくとばっちり
- RFC3137 – OSPF Stub Router Advertisement

具体例：networld+interop 2004
0）通常時

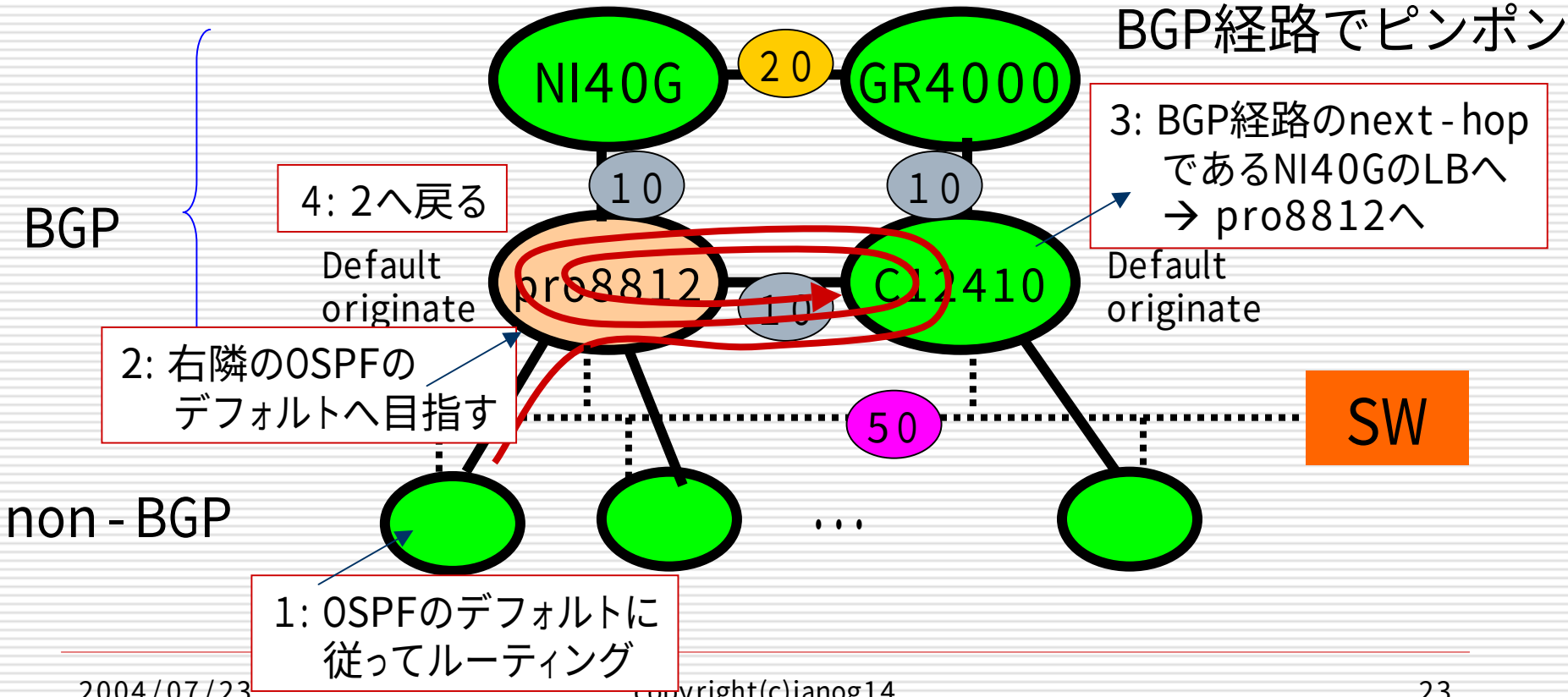


具体例：networld+interop 2004
1) リロード発生！



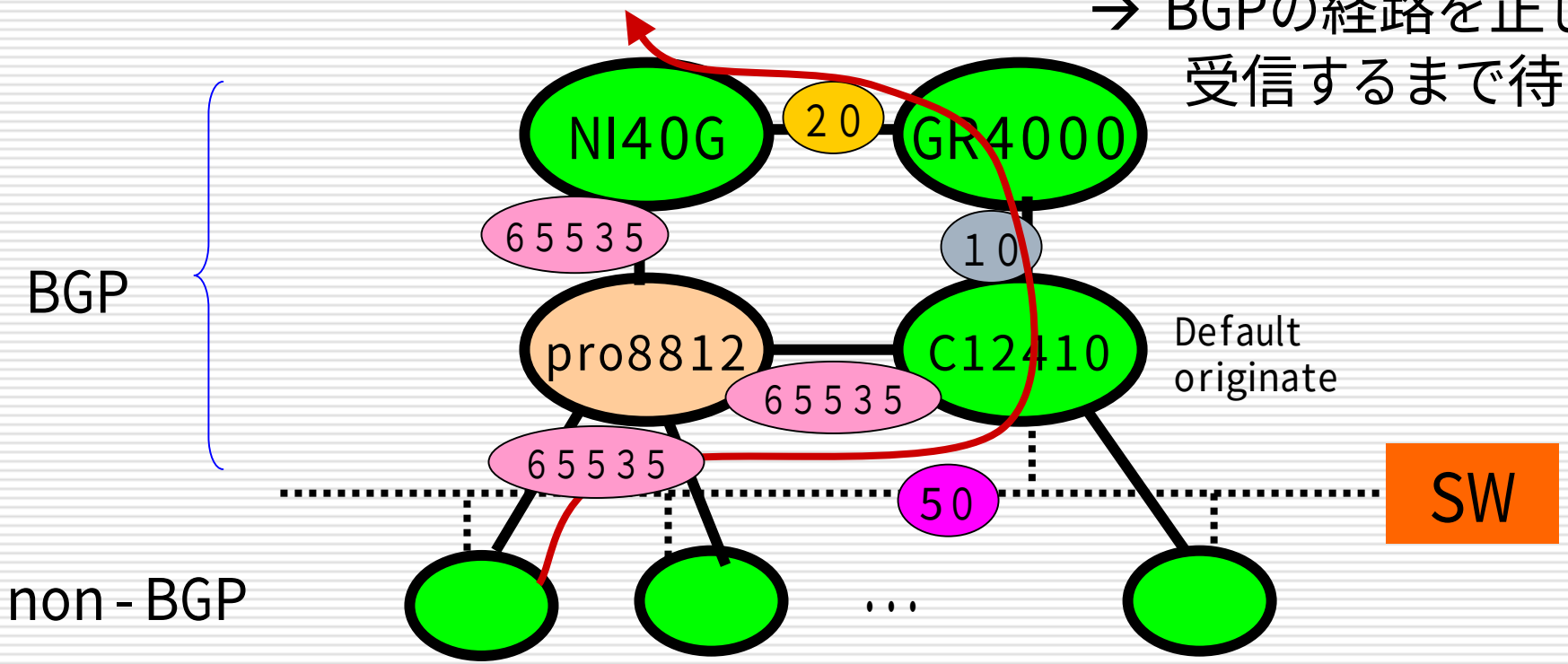
具体例 : networld+interop 2004
2) OSPF復活 + non-wfb

wait-for-bgp 未設定
→ OSPFのデフォルトと
BGP経路でピンポン

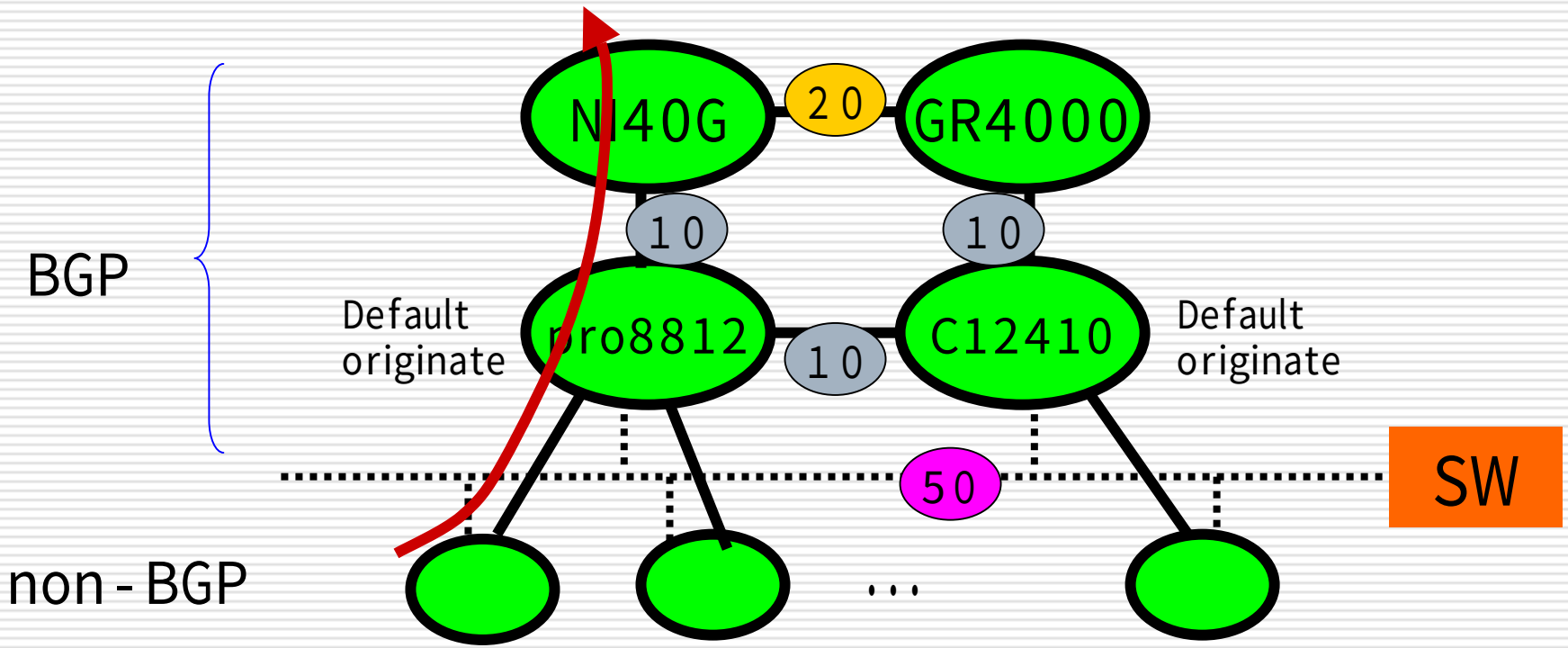


具体例：networld+interop 2004
2) OSPF復活 + wfb

wait-for-bgp 設定時
→ BGPの経路を正しく
受信するまで待つ



具体例：networld+interop 2004
3）BGPも正常時に復活



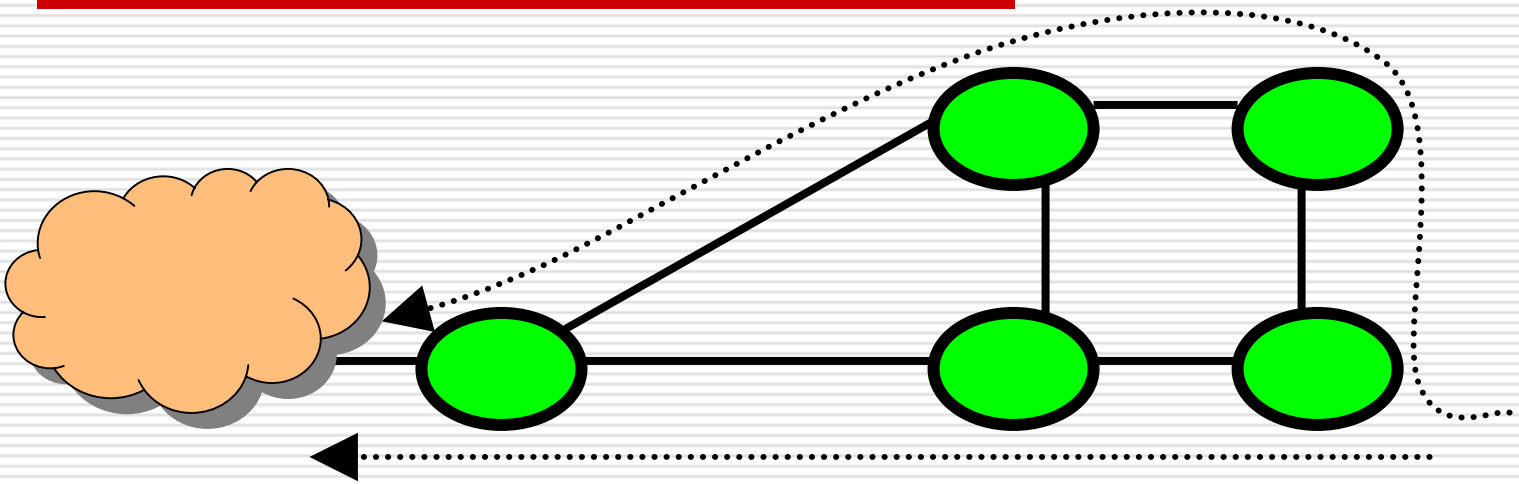
wait-for-bgp 設定例

- Cisco
  - router ospf 1
    max-metric router-lsa on-startup wait-for-bgp
- Procket
  - router ospf 1
    max-metric router-lsa on-startup wait-for bgp 100

OSPF Stub Router Advertisement

- 一定時間、router-LSAをmax値にする設定
  - Cisco
    - max-metric router-lsa on-startup <5-86400>
  - Juniper
    - edit protocols (ospf | ospf3) overload timeout 300

経路収束後、通常運用へ



- 経路収束後、迂回状態を自動的に解除
- みんな幸せ

stabilityへの挑戦

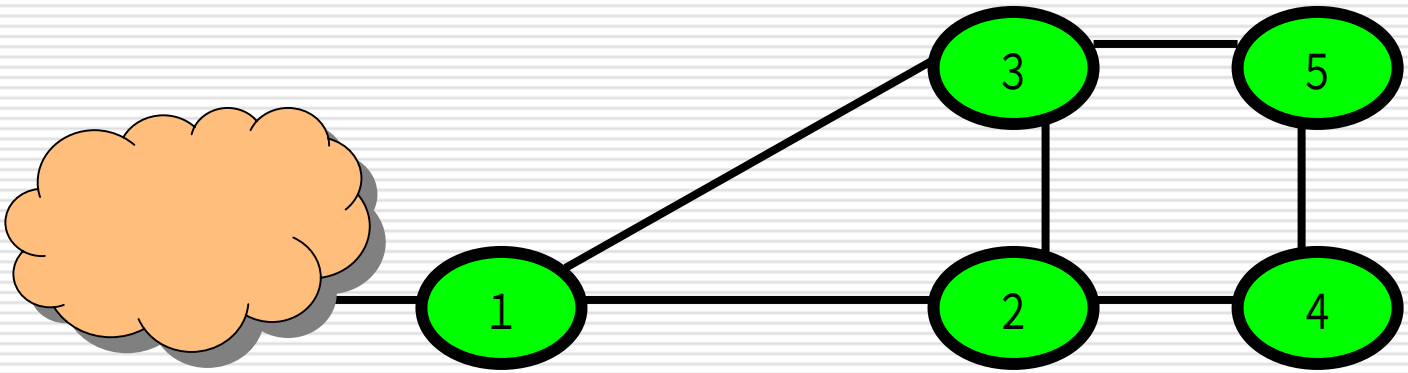
- Non Stop ほげほげ
- Graceful Restart
- headless forwarding
- Sub-second Convergence
- wait-for-bgp

- 耐障害性はどんどん高まる

幸せなパケット転送のために

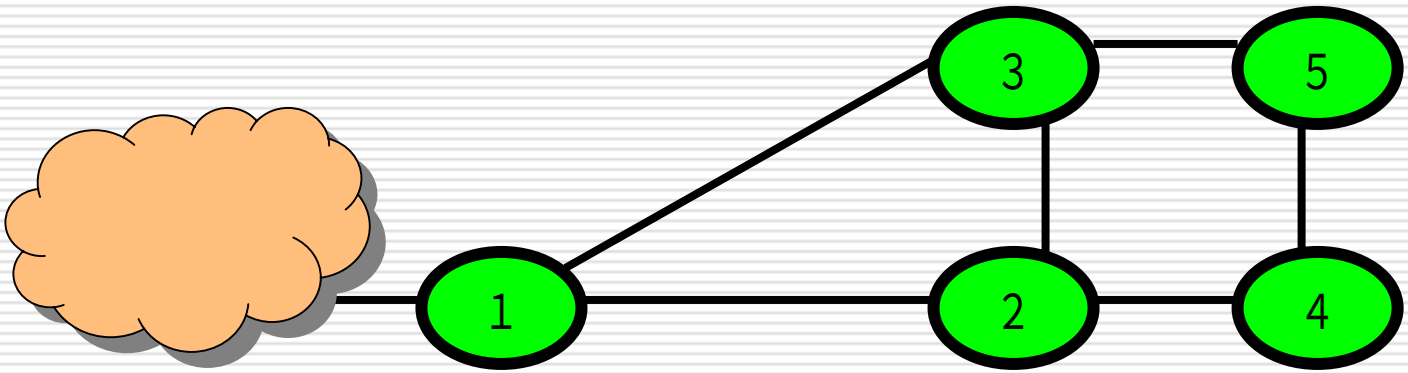
- 通常運用のことも少し考えてみよう
- BGPの更新から、FIBの更新に至る実網での挙動をよ〜く考えてみました

BGPの伝播って・・・



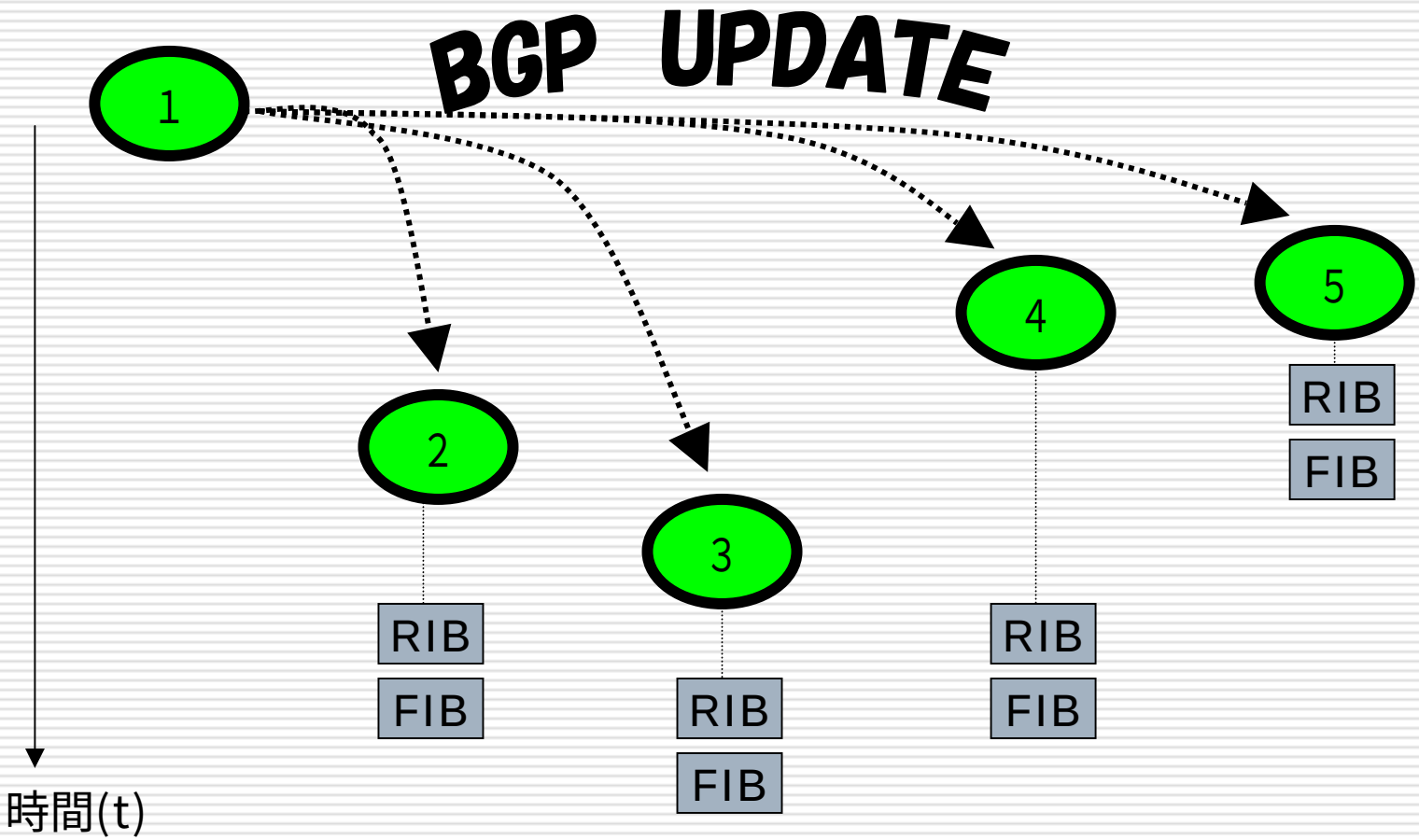
- 新規にネットワークと接続した際に、BGPの経路はどのように伝播してるでしょうか？

BGPの伝播ってランダムじゃないですか？



- 「1」のルータがどの順序でneighborにUPDATEを送ってるか、実はよく分からない
  - たいていランダム

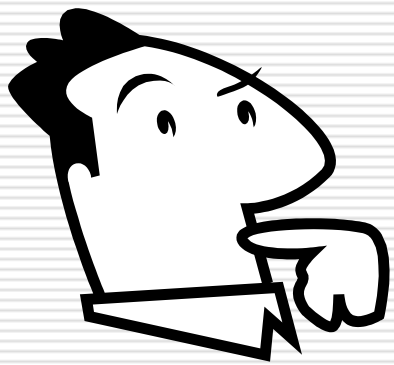
実装によっては、FIBの更新もばらばら



慌てるな！

- うまくいけば、ループは発生しません
- 発生しても、短い時間で回復します

- でも、もっとうまいやり方がありそうじゃない？



改善に向けて、その１

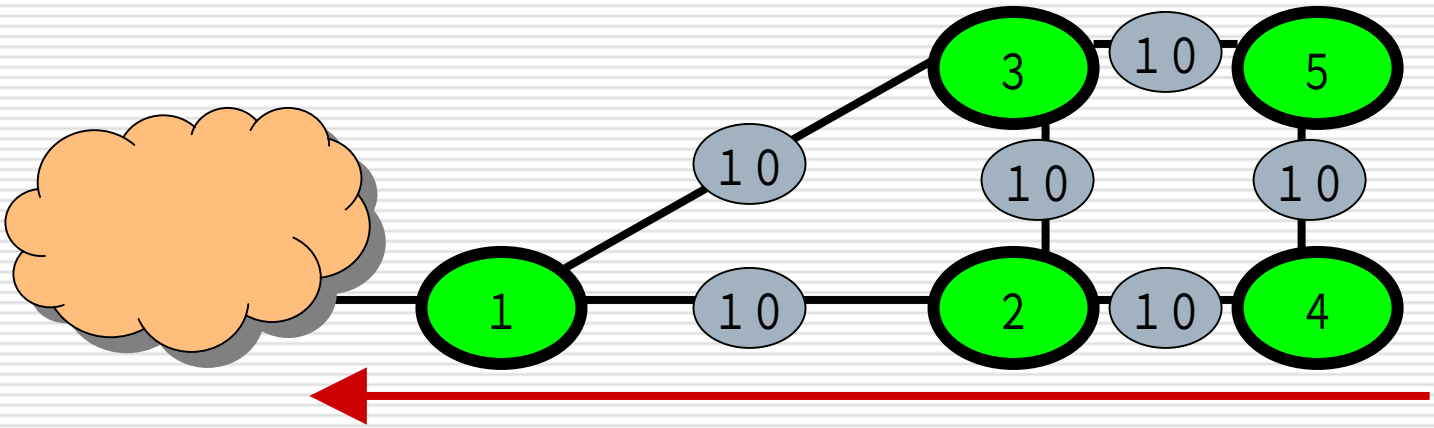
- BGP UPDATE到着後、直ぐにRIBとFIBの更新処理するのがお勧め
  - 連続したUPDATEはbackoffで対応
- それがだめなら、RIBとFIBの更新を近隣のルータで同期したい

- JuniperはUPDATEを受け取り次第更新、Ciscoは同様の機能を実装予定

改善に向けて、その２

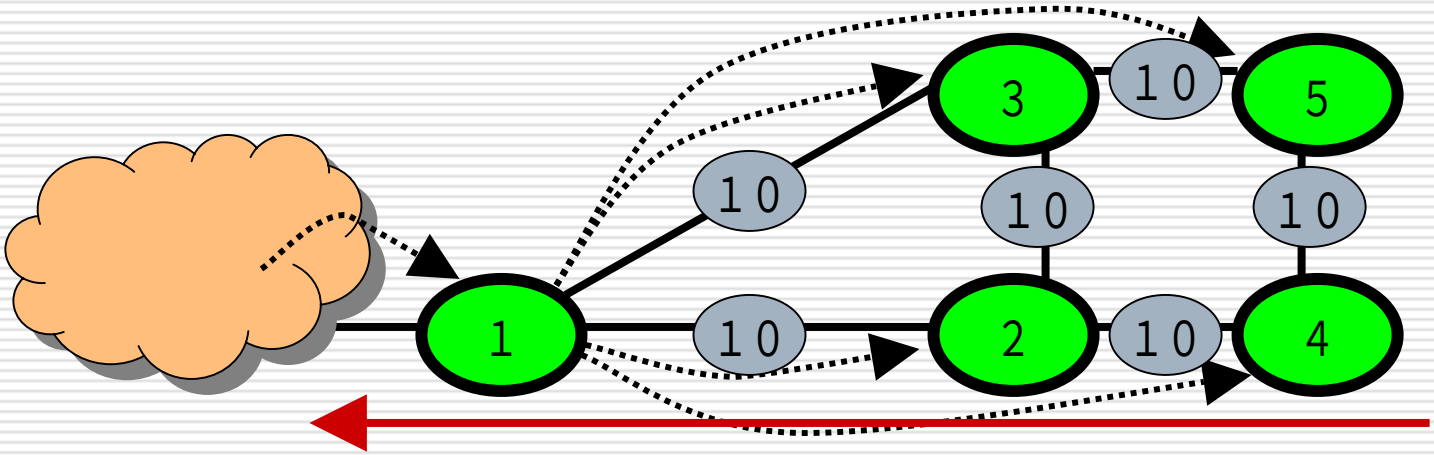
- BGP neighbor sorting
  - BGP UPDATEを、近隣のルータから順に送る
  - 最終的にトラヒックが流れるリンクを遡る様にUPDATEを送る
  - routing loopの発生を可能性を極力押さえる

BGP neighbor sorting



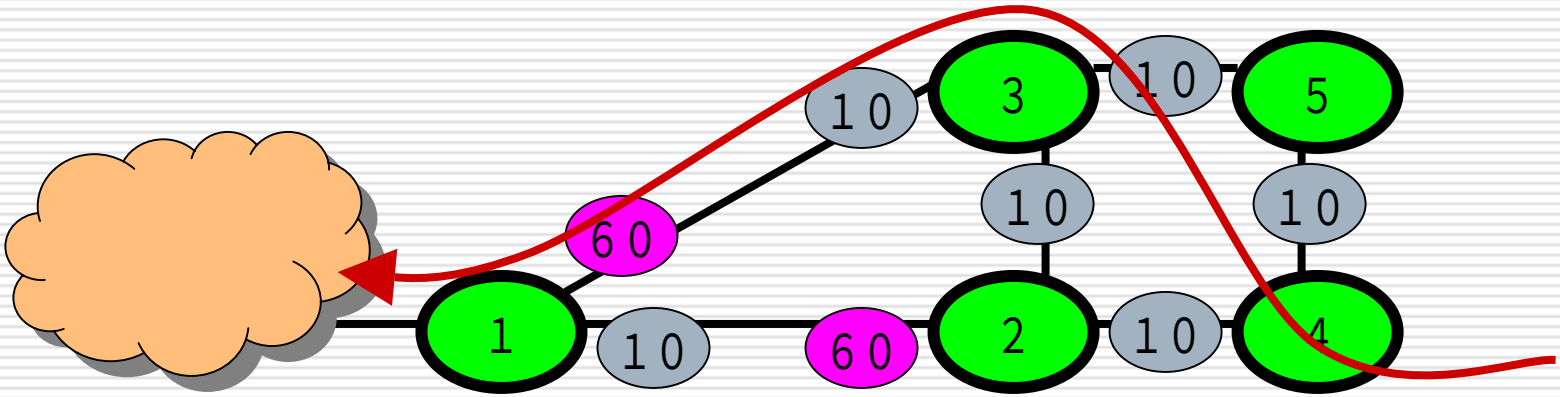
- 実装案としては、
  - 近隣 = neighbor addressへのIGP cost順

IGP Costでsortingした時の動き



- この構成ではトラヒックの流れに対して矛盾なくUPDATEが送られるので、うまく動く

ちょっと変わったネットワークだと



- IGPのCostがリンクで非対称なネットワークでは問題になるかも

BGP neighbor sorting 問題点

- まだ実装されてません(たぶん)
- IGPの構成によっては問題が起こるので、選択して有効にできるようにして欲しいです
- 途中で反応の遅いルータがあった場合に全体の収束時間が遅くなります
- UPDATEには概ねうまく動きますが、WITHDRAWNはどう処理するべきかなあ？
  - アイディア募集中

幸せなパケット転送　○○編

おしまい

質問、ご意見よろしく