クラウド接続もおまかせ、基礎からのネットワーク

# クラウド接続を意識したオンプレミスネットワークの基礎

Internet Week 2018

2018年11月27日

株式会社FORNEXT 篠宮 俊輔

# 自己紹介

- 名前: 篠宮 俊輔(しのみや しゅんすけ)
- 所属: 株式会社FORNEXT (東京都大田区)
  ネットワークエンジニア 兼 代表取締役
- ネットワークの技術面での支援を生業としています
  - ネットワーク設計、構築、運用(の技術的な部分)

# 内容

- 対外接続がインターネットだけのネットワークの、クラウド接続を題材として、オンプレミスネットワークの基礎を解説します
- 結構な割合をクラウドネットワークとの間で使用するダイナミックルーティングプロトコルであるBGPの話が占めます
- BGPを用いるネットワークに慣れる機会になればと思います

# オンプレミスネットワーク-クラウドネットワーク間を接続する理由(1)

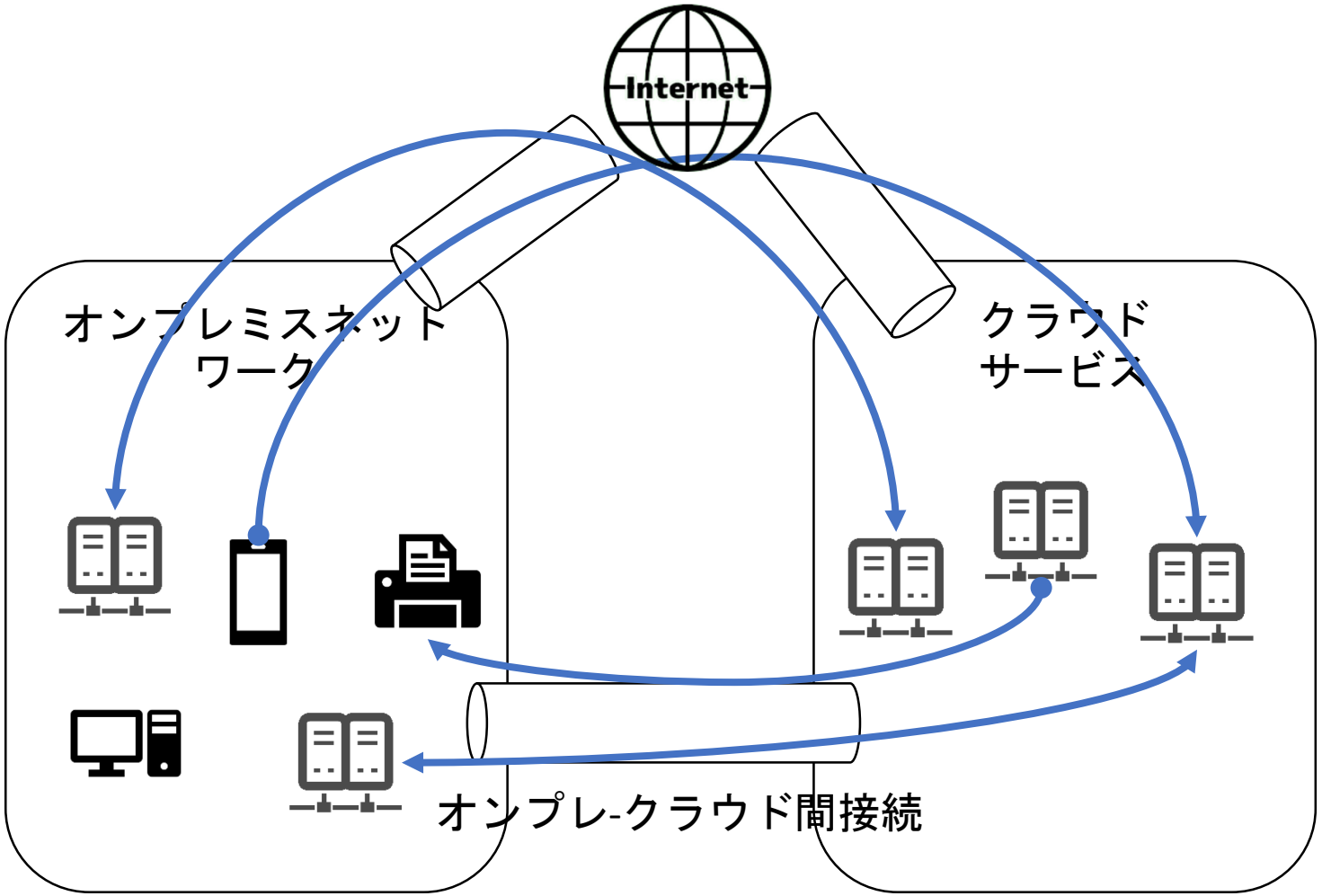
- インターネット経由のグローバルIPでの通信で足りるなら、不要
  - インターネット越しにアクセスするサービスとして実装できるなら

- 内側のネットワーク同士をつなげようと思ったら必要
  - 業務システムがクラウド上のサーバに接続
  - 端末から業務システムへの接続
  - ファイルサーバやDBサーバがクラウドにある。
  - インターネット越しに平文の通信はしたくない
  - サービスが、インターネットに露出させるようには出来ていない、向いていない

- 品質、速度の要件で
  - インターネット経由だと遅い環境で専用線で、より良い品質の接続が欲しい

- 内側のネットワーク同士をつなぐメリット
  - (対向が同じようにセキュアであれば)アクセス制御(セキュリティ)の考慮点が減る
  - プライベートIPの振ってある機器でも、NATしないので双方向のアクセスが出来る

# オンプレミスネットワーク-クラウドネットワーク間を接続する理由(2)



# 言葉の表記

- オンプレミスシステムのネットワーク → オンプレ、と表します
  - オンプレミス(on-premise)は、「自社運用の」という意味
- クラウドシステムの各利用者専用のネットワーク → クラウドと表します

# クラウドプラットフォームのメンテナンス

- 多数の契約者で使う物なので、脆弱性対応などの導入が積極的で、そのためクラウドIaaSサービスのメンテナンスの頻度も高い
- メンテナンス実施日時のお伺いなんてないし
- なので、冗長構成、冗長接続は必要
- メンテナンスが当たり前の様に行われる事を前提とする結果、対メンテナンス性、対故障性の高いシステムできあがるという思想も

# オンプレ-クラウド間接続に求めること ≒ 本チュートリアルで題材とすること

- オンプレ-クラウド間で到達すべきネットワーク間の到達性を得たい
- クラウド側の接続機器のメンテナンス時にも到達性を維持したい
  - メインをイーサネット接続、バックアップをIPsec VPN接続、など2本
- オンプレ-クラウド間の複数の接続を使い分けたい
- クラウド側だけではなく、オンプレ側もメンテナンス、故障への耐性を上げたい

# オンプレ-クラウド間接続と<br>オンプレ複数拠点構成の違い

- 基本的に同じであるが、クラウドの場合は、接続方法、構成はクラウド側がサポートしている方法に限られる<br>どんな接続方法でもつなげられるのではない
  - オンプレ内部-クラウド内部間を結びたければイーサネットかIPsec site-to-site VPNで
  - オンプレ、クラウド間の経路情報の伝達はBGPを使用する

- オンプレであれば、接続の機器を自由に選択して構成可能
  - 2拠点間をL2で結んで同一セグメント
    - サーバセグメントを多拠点に延長とか
  - L2VPNはったり
  - 拠点間接続のルーティングにBGPではなく、OSPF、RIP、スタティック経路使ったり

# 何にしてもルーティングが必要
# なぜBGPなのか

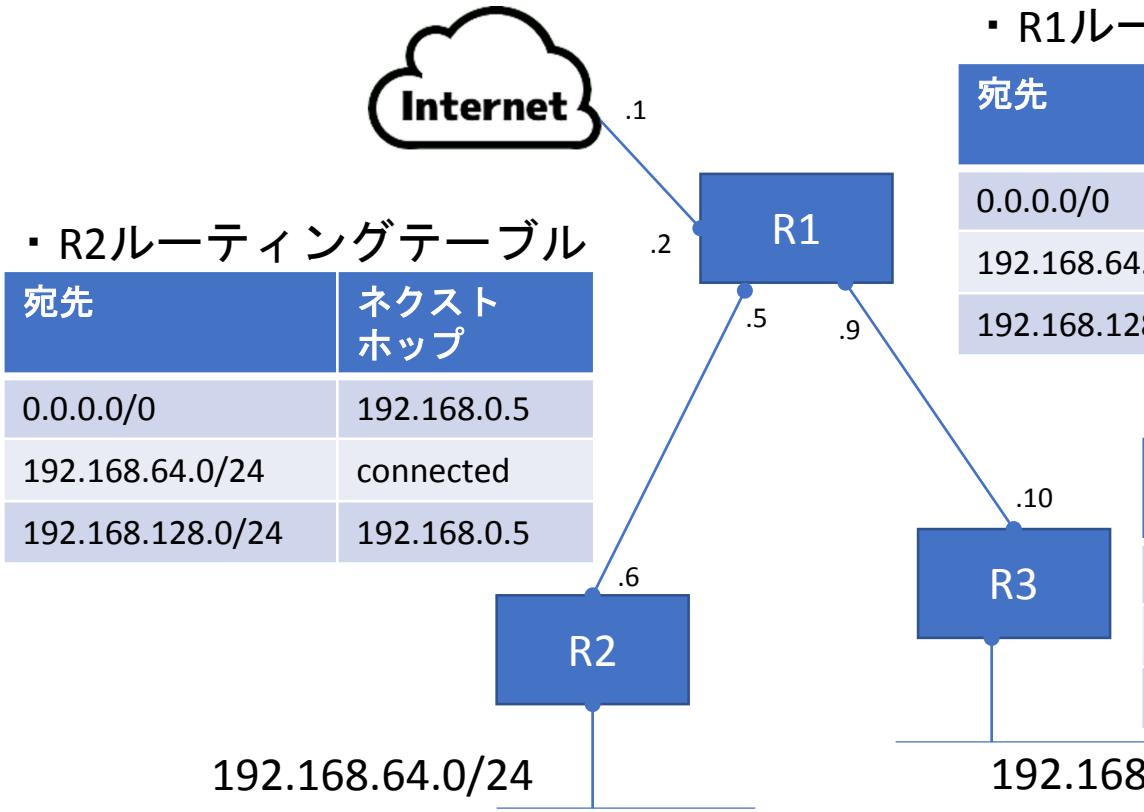
- BGPは経路制御が柔軟で、異なる運用主体間でのルーティングに向いている。
  - ASパス(の長さ)やLocal Preferenceでベストパスの選定
  - ASパス、プレフィックス、属性などによる経路フィルタ、属性値変更。

- OSPFだとそうはいかない
  - 運用主体が異なるネットワーク間の接続は難しい、手間。
  - マルチエリア構成の場合、バックボーンエリア(エリア0)を介してつなげないとならない
    - どちらのネットワークのバックボーンエリアに自ネットワークのエリアはつながる?
    - 他のネットワークのバックボーンエリア同士がつながったらネットワーク構成が漏れる
    - エリア内経路(の元であるLSA)はフィルタ出来ない

# 知識編
# ルーティングの基本

# ルーティングの基本1

- ネットワークを構成する個々のL3機器が宛先アドレスとネクストホップの対応の情報群(ルーティングテーブル)を使ってパケットを転送する
- このルーティングテーブルを(手動で)設定するのがスタティックルーティング
  ルーティングプロトコルを用いて設定するのがダイナミックルーティング




・R1ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.1 |
| 192.168.64.0/24 | 192.168.0.6 |
| 192.168.128.0/24 | 192.168.0.10 |

・R2ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.5 |
| 192.168.64.0/24 | connected |
| 192.168.128.0/24 | 192.168.0.5 |

・R3ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.9 |
| 192.168.64.0/24 | 192.168.0.9 |
| 192.168.128.0/24 | connected |

# ルーティングの基本2(1)
# 最長一致(longest match)

・R1ルーティングテーブル

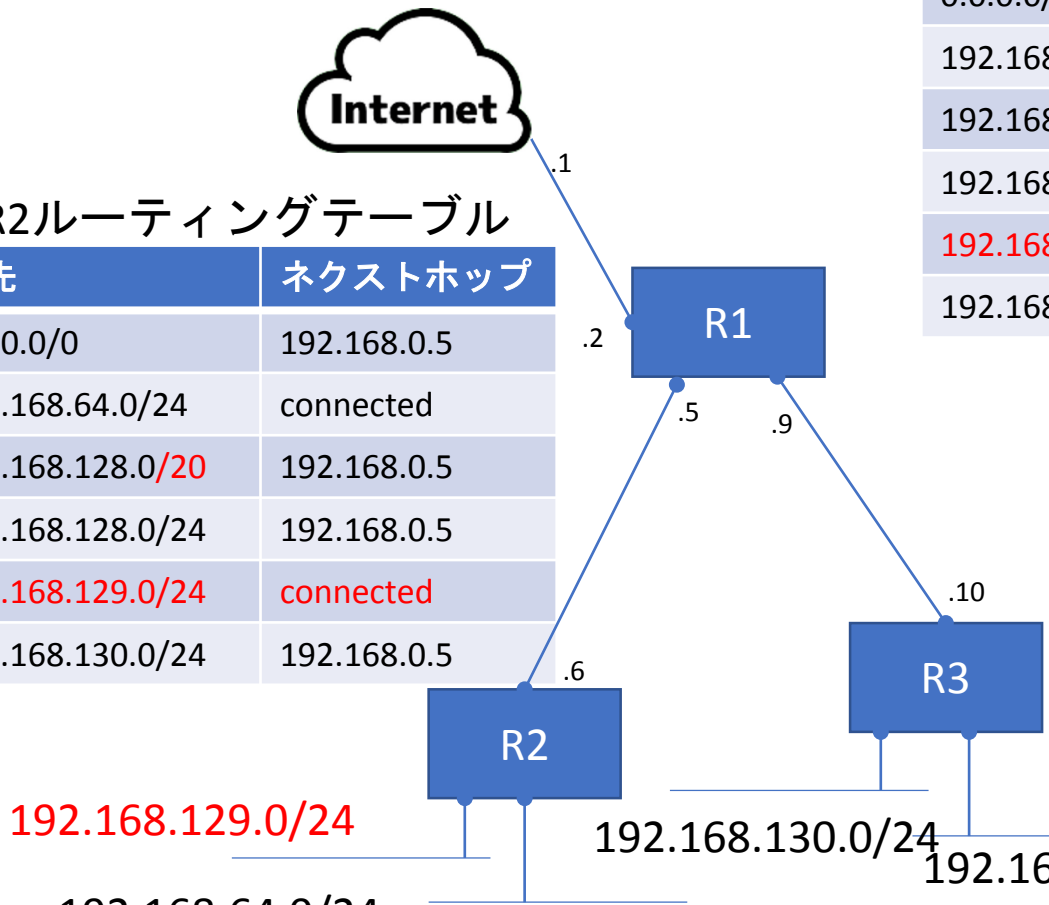
| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.1 |
| 192.168.64.0/24 | 192.168.0.6 |
| 192.168.128.0/20 | 192.168.0.10 |
| 192.168.128.0/24 | 192.168.0.10 |
| 192.168.129.0/24 | 192.168.0.6 |
| 192.168.130.0/24 | 192.168.0.10 |

・R2ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.5 |
| 192.168.64.0/24 | connected |
| 192.168.128.0/20 | 192.168.0.5 |
| 192.168.128.0/24 | 192.168.0.5 |
| 192.168.129.0/24 | connected |
| 192.168.130.0/24 | 192.168.0.5 |

・R3ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.9 |
| 192.168.64.0/24 | 192.168.0.9 |
| 192.168.128.0/20 | discard |
| 192.168.128.0/24 | connected |
| 192.168.129.0/24 | 192.168.0.9 |
| 192.168.130.0/24 | connected |



# ルーティングの基本2(2)
# 最長一致(longest match)

- プレフィックス部分が一致するエントリの中で、最も長いプレフィックス長のエントリを使ってルーティング

・192.168.129.10(11000000.10101000.10000001.00001010)宛てパケットに一致するプレフィックスのビット数

| 宛先 | 2進数表記 | 一致ビット数 | ネクストホップ |
|---|---|---|---|
| 0.0.0.0/0 | 00000000.00000000.00000000.00000000/0 | 0ビット | 192.168.0.9 |
| 192.168.64.0/24 | 11000000.10101000.01000000.00000000/24 | 一致しない | 192.168.0.9 |
| 192.168.128.0/20 | 11000000.10101000.10000000.00000000/20 | 20ビット | discard |
| 192.168.128.0/24 | 11000000.10101000.10000000.00000000/24 | 一致しない | connected |
| 192.168.129.0/24 | 11000000.10101000.10000001.00000000/24 | 24ビット | 192.168.0.9 |
| 192.168.130.0/24 | 11000000.10101000.10000010.00000000/24 | 一致しない | connected |

# ルーティングの基本3
# 集約経路

・R1ルーティングテーブル

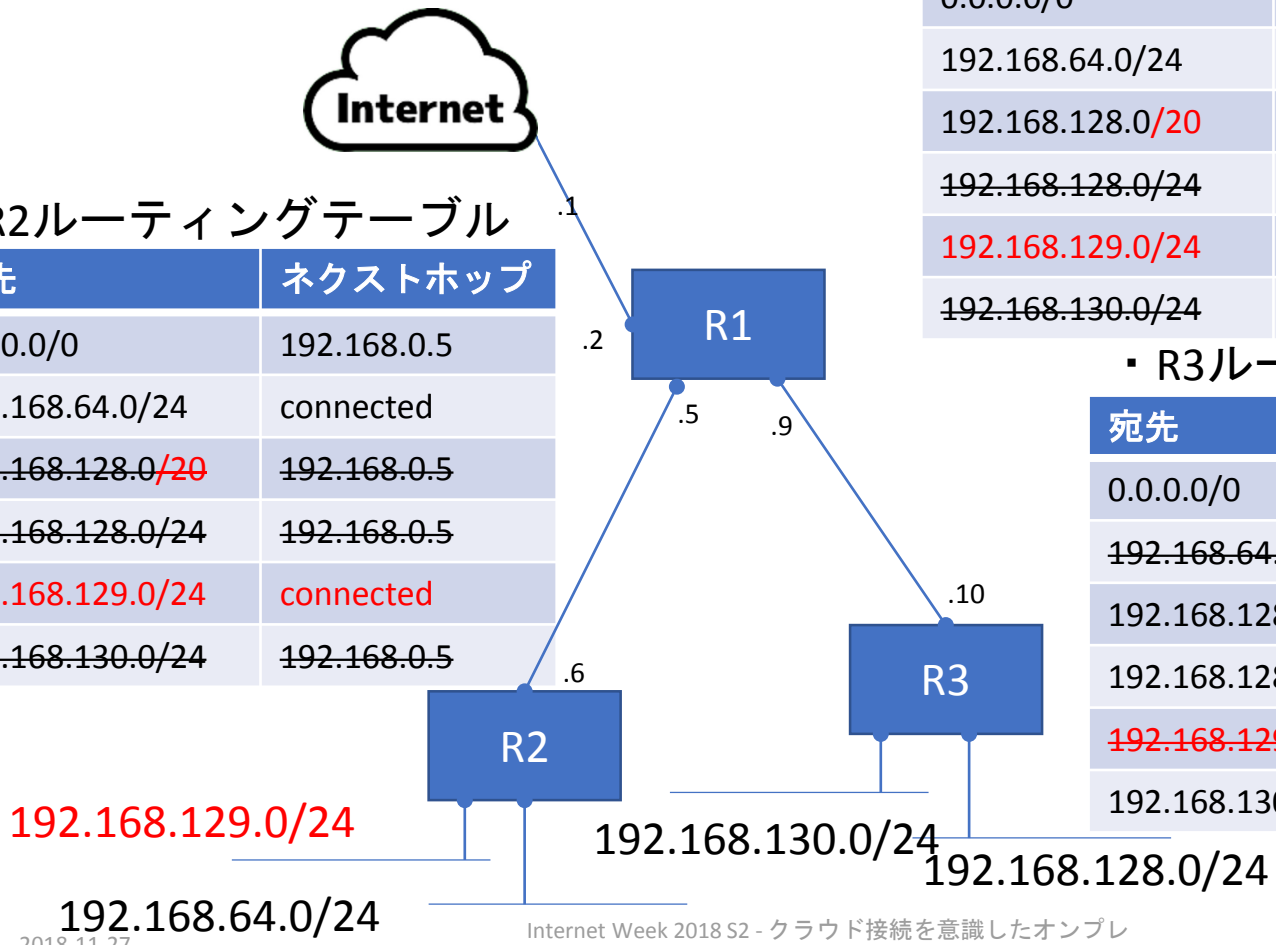
| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.1 |
| 192.168.64.0/24 | 192.168.0.6 |
| 192.168.128.0/20 | 192.168.0.10 |
| ~~192.168.128.0/24~~ | ~~192.168.0.10~~ |
| 192.168.129.0/24 | 192.168.0.6 |
| ~~192.168.130.0/24~~ | ~~192.168.0.10~~ |

・R2ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.5 |
| 192.168.64.0/24 | connected |
| ~~192.168.128.0/20~~ | ~~192.168.0.5~~ |
| ~~192.168.128.0/24~~ | ~~192.168.0.5~~ |
| 192.168.129.0/24 | connected |
| ~~192.168.130.0/24~~ | ~~192.168.0.5~~ |

・R3ルーティングテーブル

| 宛先 | ネクストホップ |
|---|---|
| 0.0.0.0/0 | 192.168.0.9 |
| ~~192.168.64.0/24~~ | ~~192.168.0.9~~ |
| 192.168.128.0/20 | discard |
| 192.168.128.0/24 | connected |
| ~~192.168.129.0/24~~ | ~~192.168.0.9~~ |
| 192.168.130.0/24 | connected |



# ダイナミックルーティング

- ルーティングプロトコルを使って、ルーティングテーブルを作成、維持(ネットワークの変化に対応)する方法
  - そのプロトコルにRIP、OSPF、IS-IS、BGPなど

- ルータ間で情報の交換を行い、ルータが何らかの基準、アルゴリズムに従って自ルータのルーティングテーブルを作成する
  - 基準は、インタフェースのコストとか、通過するルータ、AS数とか、指定した優先度など
  - 宛先のネットワークに到着できる(ループしない)ネットワークを維持する

- 複数のルータ間を複数のリンクで結び、ダイナミックルーティングを用いる事で、リンクやルータのダウン箇所を迂回できる

# BGP – Border Gateway Protocol

- インターネットを構成するネットワーク間で使われるルーティングプロトコル
  - 柔軟なので、単なるIPだけではなく様々なルーティングで使用される
    - IP-VPN(RD+プレフィックス)
    - L2スイッチング(MACアドレス)
- 関係するルータと経路情報を送受(交換)し、宛先プレフィックスへの経路(ベストパス)を決定、ルーティングテーブルを作成
  - このベストパスを決定するアルゴリズムがベストパスセレクションアルゴリズム
  - 送信を、「広告」「広報」などと表現することも
- ディスタンスベクタ(距離、方向)型
- BGPでの「距離」はASパス(長)
  - しかし、必ずしもASパス長だけでベストパスは決まるのではない

# ASとAS番号

- AS(Autonomous System)は、インターネットが、「ネットワークの相互接続体」とすると、そのインターネットを構成するネットワークのこと
- ASの識別子が、AS番号
  - JPNICがAS2515など。
  - 32ビット幅の数値。16ビット.16ビットの表現もあった
- AS番号は、IPアドレスと同じようにインターネット全体で一意になるように管理されているリソースの一つ。
- IPアドレスのプライベートIPアドレスやULAの様に、インターネットに自ネットワークの中だけで使えるプライベートAS番号が用意されている。
  - 64512〜65534
  - 4200000000-4294967294
- IPアドレスと同様に、ドキュメント用のAS番号もあります
- グローバルAS番号を勝手に使ってはいけないです

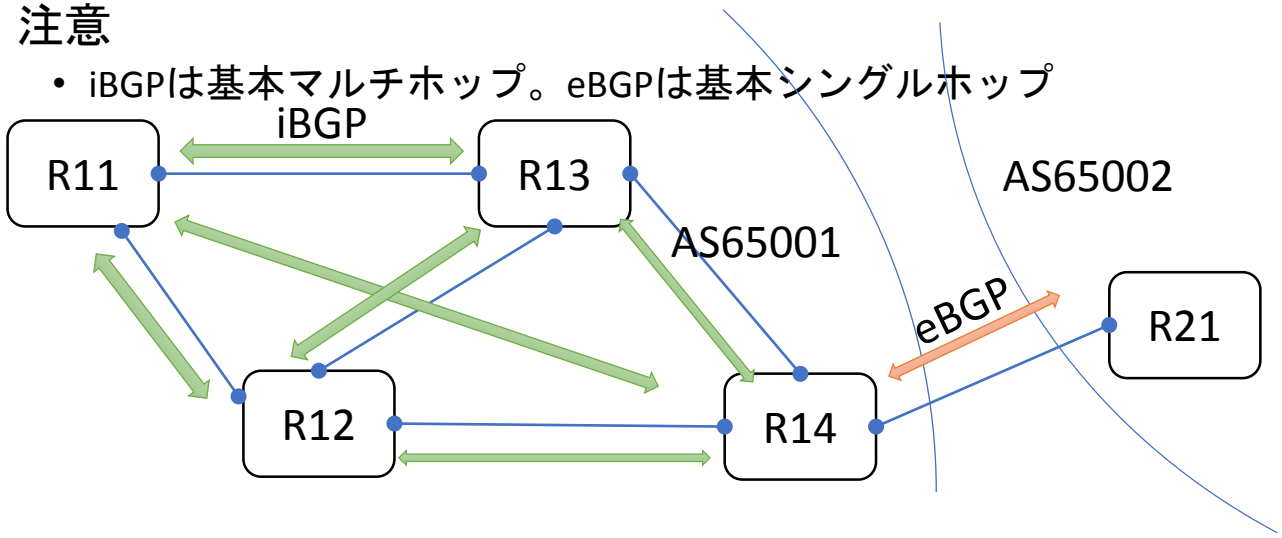
# BGPを用いてのクラウドとの接続

- オンプレ–クラウド間の二者間接続のため、BGPについての少ない知識、理解でも接続できます
  - インターネットのように、対向のASの先に、更にASがあるなどではない
  - 経路数も、そう多くない
  - 接続するクラウドが複数、オンプレの拠点が複数ある、程度

- BGPを使って行うのは、次
  - そもそもの、到達性の確保
  - クラウドとの間で複数の接続があるなら、それぞれにトラヒックを流し分けたい
  - 接続する機器が故障した、接続に用いるリンクが落ちた場合にその間を迂回して到達性を維持したい

# BGPの基本1: BGPセッション

- ルータ間でBGPセッションを確立し経路情報を交換
  - このBGPセッションは一度張ったら張りっぱなし
  - 相手のBGPルータをネイバと呼ぶ
  - ネイバとの接続があるかは、メッセージ受信(UPDATE、KEEPALIVE)で確認。HOLDTIME時間分メッセージを受信しなかったら断する
- BGPセッションのL4(トランスポート)は、TCP
  - IPリーチャブルであれば、BGPセッションは張れる
  - 正常に動作しないネットワーク構成でもBGPセッションは張れるので注意
    - iBGPは基本マルチホップ。eBGPは基本シングルホップ



# BGPの基本2: 交換する経路情報

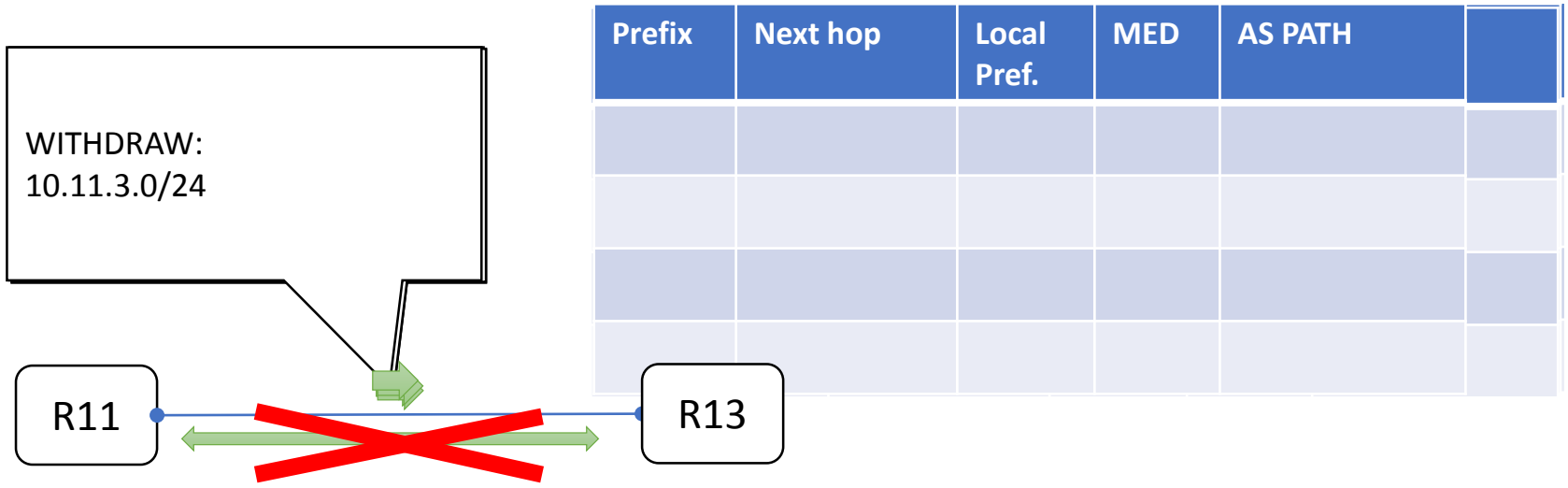
- ASパス、ネクストホップ、Local Preferenceなどの属性とNLRI(IPv4やIPv6ではプレフィックス)群のセットを経路情報としてBGPルータ間で交換

| 属性名 | 内容 |
|---|---|
| AS_PATH | 経路情報が経由してきたASのAS番号のリスト。eBGPでの送信時には、一番左に自身のAS番号を追加する。 |
| NEXT_HOP | NLRI(プレフィックス)で示されるネットワークに到達するためのネクストホップ |
| LOCAL_PREF | Local Preference。経路の優先度。大きいほど高優先。eBGPでは伝わらない。 |
| MULTI_EXIT_DISC | MED。経路へのメトリック。小さいほど近い。eBGPでも伝わる。基本的には、同じ隣接AS間の経路の比較だけに使われる。しかし変更可能(always-compare-med) |
| COMMUNITY | 経路につけるタグ。NO_EXPORTなどいくつかのWell-knownもあり |

他にも、ORIGIN属性など、多数属性はあります

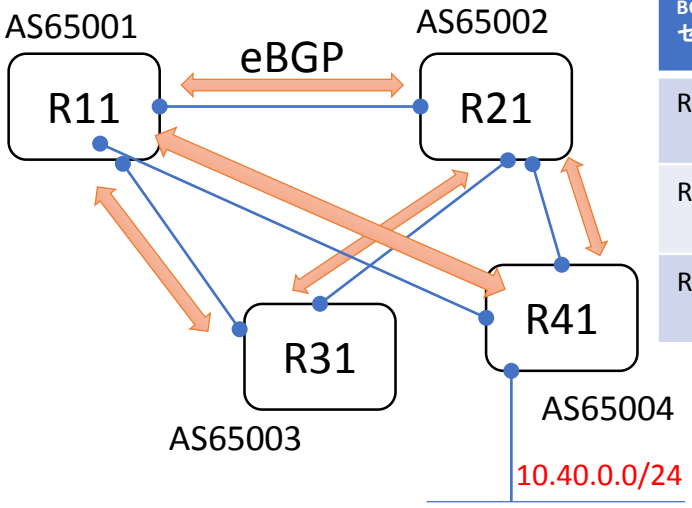
# BGPの基本2: 経路情報の広告

- UPDATEメッセージで経路情報を更新する、取り消す
  - 取り消さなければいつまでも保持
  - 既に受け取っているNLRIの経路情報が来たら、新しいもので上書き。
- BGPセッションが落ちたら、そのセッションで交換していた経路情報は忘れる



# BGPの基本3: ベストパスセレクション

- 保持している経路情報から、各プレフィックス毎にベストパスを一つ選ぶ
  - BGPネイバから受け取った経路情報や、自ルータで作りBGPに載せた経路から一つ選ぶ
  - このアルゴリズムが、ベストパスセレクションアルゴリズム
  - BGPの実装により細部は異なる部分もある。実装固有のパラメータなど




R11の持つ10.40.0.0/24の経路情報

| BGP セッション | Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|---|
| R21 | 10.40.0.0/24 | 172.16.0.21 (R21) | 100 | 65002 65004 | |
| R41 | 10.40.0.0/24 | 172.17.0.41 (R41) | 100 | 65004 | |
| R31 | 10.40.0.0/24 | 172.18.0.31 (R31) | 100 | 65003 65002 65004 | |

# ベストパスセレクションアルゴリズム
# Cisco IOSの例

- 1. 最も高い WEIGHT を持つパスが優先されます。
  注: WEIGHT は シスコ独自のパラメータです。 設定されているルータに対してローカルに割り当てられます。
- 2. 最も高い LOCAL_PREF を持つパスが優先されます。
- 3. network または aggregate BGP サブコマンドによって、あるいは IGP からの再配布を通じて、ローカルで発信されたパスが優先されます。network コマンドや redistribute コマンドによるローカル パスの方が、aggregate-address コマンドによるローカル集約よりも優先されます。
- 4. 最短の AS_PATH を持つパスが優先されます。
- 5. 最小のオリジン タイプを持つパスが優先されます。
- 6. 最小の Multi-Exit Discriminator（MED）を持つパスが優先されます。
- 7. iBGP パスよりも eBGP パスの方が優先されます。
  最適パスが選択される場合は、ステップ 9 に移動してください（マルチパス）。
- 8. BGP ネクストホップへの最小の IGP メトリックを持つパスが優先されます。
  最適パスがすでに選択されていても、続けてください。
- 9. マルチパスが BGP マルチパス用にルーティング テーブルでインストールされる必要があるかどうか判断します。
  最適パスがまだ選択されていない場合、続けてください。
- 10. 両方のパスが外部のときは、先に受信したパス（最も古いパス）が優先されます。
  この手順によってルートフラップが最小限に抑えられます。
- 11. 最小のルータ ID を持つ BGP ルータから送られたルートが優先されます。
- 12. 発信元 ID またはルータ ID が複数のパスで同じ場合は、最小のクラスタ リスト長を持つパスが優先されます。
- 13. 最小の隣接ルータ アドレスから送られたパスが優先されます。

13ルールありました

サポート / 技術サポート / IP / IP ルーティング / トラブルシューティング テクニカルノーツ / BGP で最適パスを選択するアルゴリズム
https://www.cisco.com/c/ja_jp/support/docs/ip/border-gateway-protocol-bgp/13753-25.html
から一部を抜粋

# ベストパスセレクションアルゴリズム抜粋

- 保持する経路に対して、1ルールずつ比較。
  タイブレーク(引き分け)したら、次のルールで比較
  これを、保持する経路情報が変わる度に適用してベストパスを決定

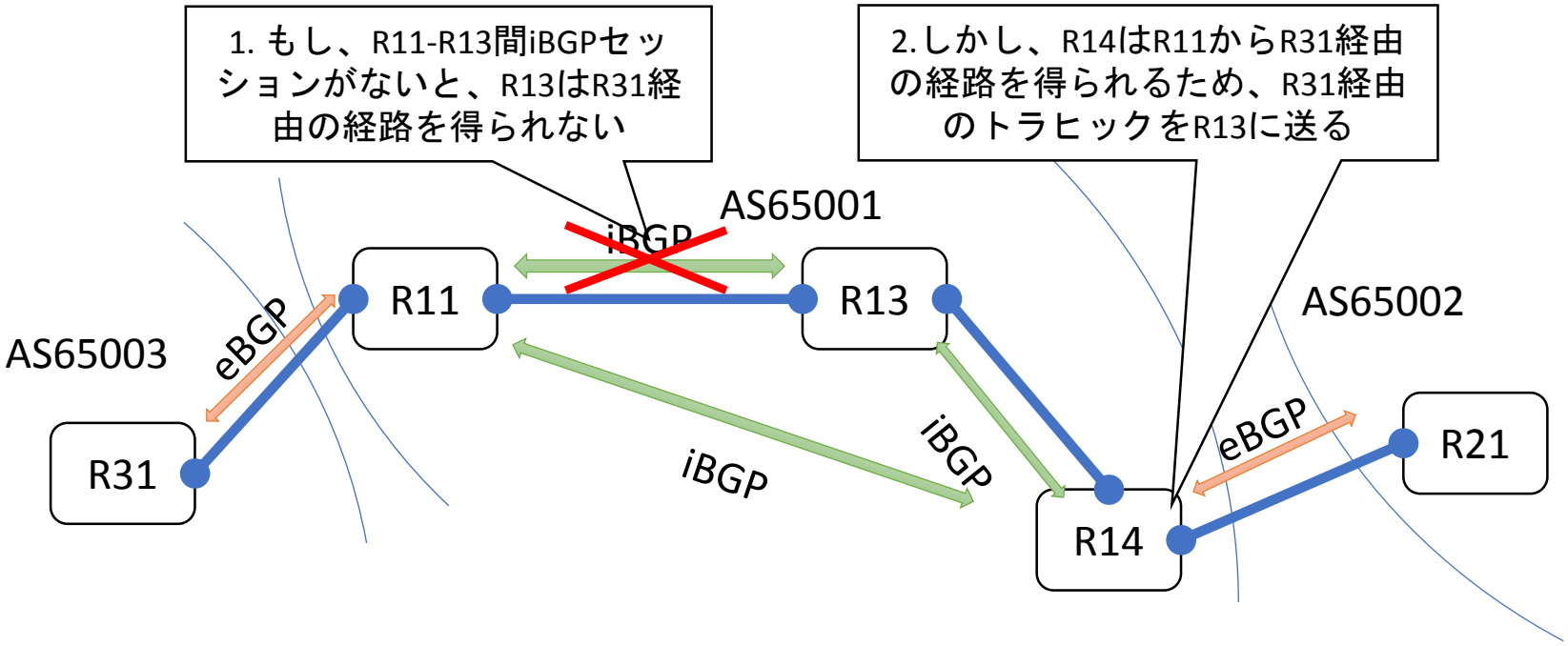
- 2. Local Preferenceが最大
- 4. ASパス長が最短
- 6. MEDの値が最小
- 7. iBGPよりeBGPを優先
- 8. ネクストホップまでのIGPメトリックが最小
- 10. 昔からあるパス

# iBGPとeBGP

| | eBGP | iBGP |
|---|---|---|
| セッションを張る2台のルータのAS | 異なるAS番号 | 同じAS番号 |
| 経路ループ排除 | 他のASに経路を渡す際、AS_PATHに自AS番号をくわえる。<br>他から受け取った経路のAS_PATHが含まれていたらその経路は無視<br><br>例: 自AS番号が65003だとする<br>65001 65002 65003 65000 | iBGPで受けた経路は、iBGPでは渡さない |
| 経路送信時に書き換える属性 | NEXT_HOP(BGPを張っているアドレスに)<br>AS_PATH(自AS番号を追加)<br>LOCAL_PREF(伝わらない) | next-hop selfの設定があれば、送信経路のNEXT_HOPをBGPを張っているアドレスに。 |
| セッションを張るアドレス | AS間のリンクのアドレス | ループバックアドレス。ただし、必ずそうで無ければならない訳ではない。 |

# iBGPのトポロジ

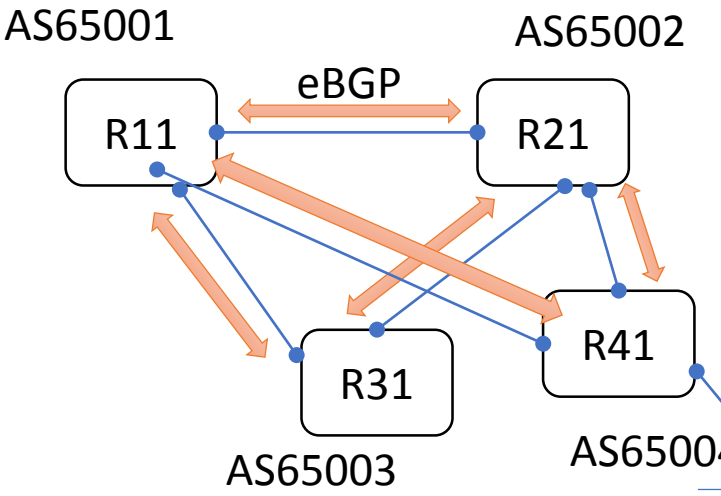
- 基本、フルメッシュ構成。またはRR構成
  - フルメッシュにしないと、ASを構成するルータで経路の一貫性が取れない
    - 一時的にiBGP セッションが落ちる場合も同様



# BGPでのトラヒック制御

# トラヒック制御、経路制御とは?

- 先ほどのベストパスセレクションの例は、ベストパスは、ネットワークトポロジに基づく属性(ASパス)で決まっていた。
  - 経由するASの数が少ないのがベスト、というポリシ。
- しかし、ASパスだけでは決めたくない場合がある
事情の例:
  - ASパスが最短の経路のリンクが細い
  - ASパスは長いが、金銭的コストが安いのでそこに流したい
  - トランジットのお客さまなので、その接続に流してお金をいただきたい
- 属性情報の書き換え、トポロジの変更など駆使して経路≒トラヒックを制御する




R11の持つ10.40.0.0/24の経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.40.0.0/24 | 172.16.0.21 (R21) | 100 | 65002 65004 | |
| 10.40.0.0/24 | 172.17.0.41 (R41) | 100 | 65004 | |
| 10.40.0.0/24 | 172.18.0.31 (R31) | 100 | 65003 65002 65004 | |

# トラヒック制御と経路制御

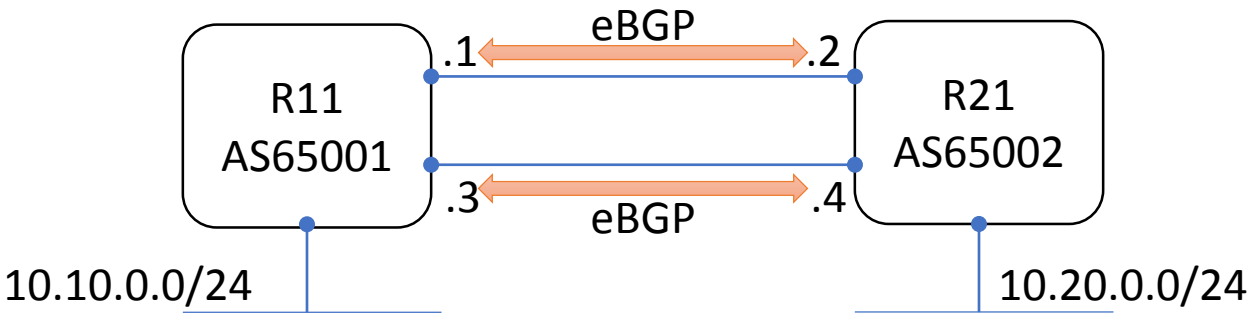
- トラヒックが経由するリンク(ネクストホップ)を制御する
  - 自分が送り出すトラヒックの制御(出の制御)、相手が送り出すトラヒックの制御(入の制御)の2方向がある。
- 出のトラヒックの制御は比較的簡単
  - 自分の都合の良い経路がベストになるよう属性情報を書き換える
  - 自身の接続のうち、どこから出すか?を制御できる
- 入りのトラヒックの制御は難しかったり出来なかったりする
  - 相手(自分では直接制御できない)に、自分の都合の良い経路をベストパスにもらわなければならない

- このトラヒック制御を、経路の属性情報の書き換えにより行う
  - Local Preferenceを大きな値に設定する(受信)
  - MEDを大きな値に設定する(送信、受信)
  - ASパスを付け足して、ASパス長を長くする(送信、受信)

- BGPではない手段として、more specific route(詳細経路)を用いる方法も

# 経路制御の例1(1)

- 同じプレフィックス宛てに複数の接続がある
  - AS65001の立場で考えます。
  - AS65001は、事情により1本目にトラヒックを流したい出、入共に。




R11の持つ10.20.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.20.0.0/24 | 172.16.0.2 | 100 | 65002 | |
| 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |

R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | |

# ベストパスセレクションアルゴリズム抜粋(再掲)

- 保持する経路に対して、次で比較。
  タイブレーク(引き分け)したら、次の基準で比較
  これを、保持する経路情報が変わる度に適用してベストパスを決定

この辺が使えそう

- 2. Local Preferenceが最大
- 4. ASパス長が最短
- 6. MEDの値が最小
- 6. iBGPよりeBGPを優先
- 8. ネクストホップまでのIGPメトリックが最小

# 経路制御の例1(2)
# AS65001の出の経路制御

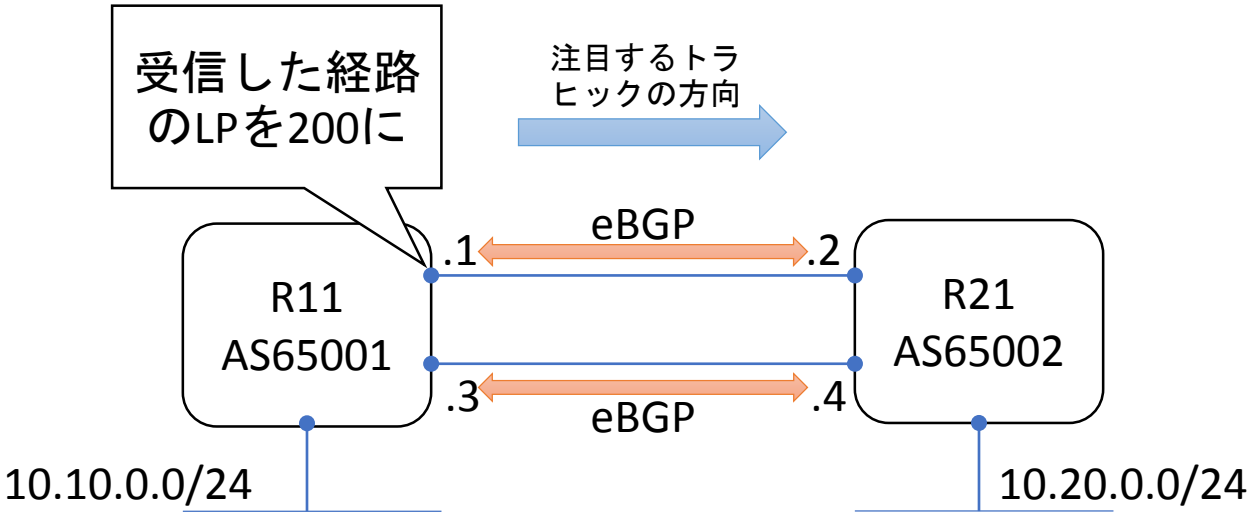
- 1本目を優先するために複数の方法がある。

R11の持つ10.20.0.0/24への経路情報

| ベストパスの決まり手 | 調整内容 | Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|---|---|
| Local Pref. | 1本目で受信した経路はLocal Pref.を 200に設定。2本目はそのまま(デフォルトの100) | * 10.20.0.0/24 | 172.16.0.2 | 200 | 65002 | |
| | | 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |
| ASパス(長) | 2本目で受信した経路にAS65002を1個プリペンド | * 10.20.0.0/24 | 172.16.0.2 | 100 | 65002 | |
| | | 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 65002 | |
| MED | 1本目で受信した経路のMEDは100に。2本目は200に。 | * 10.20.0.0/24 | 172.16.0.2 | 100 | 65002 | 100 |
| | | 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | 200 |

# 経路制御の例1(3)

- AS65001の出の制御のため、Local Preferenceで1本目を優先




R11の持つ10.20.0.0/24への経路情報

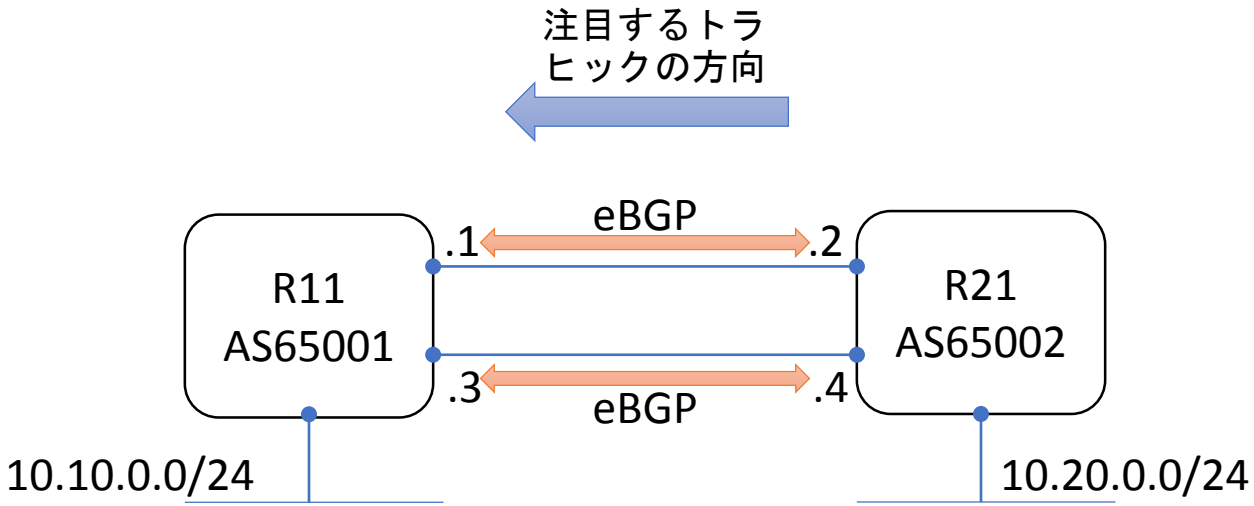
| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| *10.20.0.0/24 | 172.16.0.2 | 200 | 65002 | |
| 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |

R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | |

# 経路制御の例1(4)

- AS65001の入りの制御のため、AS65001で出来る何かをする。




R11の持つ10.20.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| *10.20.0.0/24 | 172.16.0.2 | 200 | 65002 | |
| 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |

R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | |

# 経路制御の例1(5)
# AS65001の入の経路制御

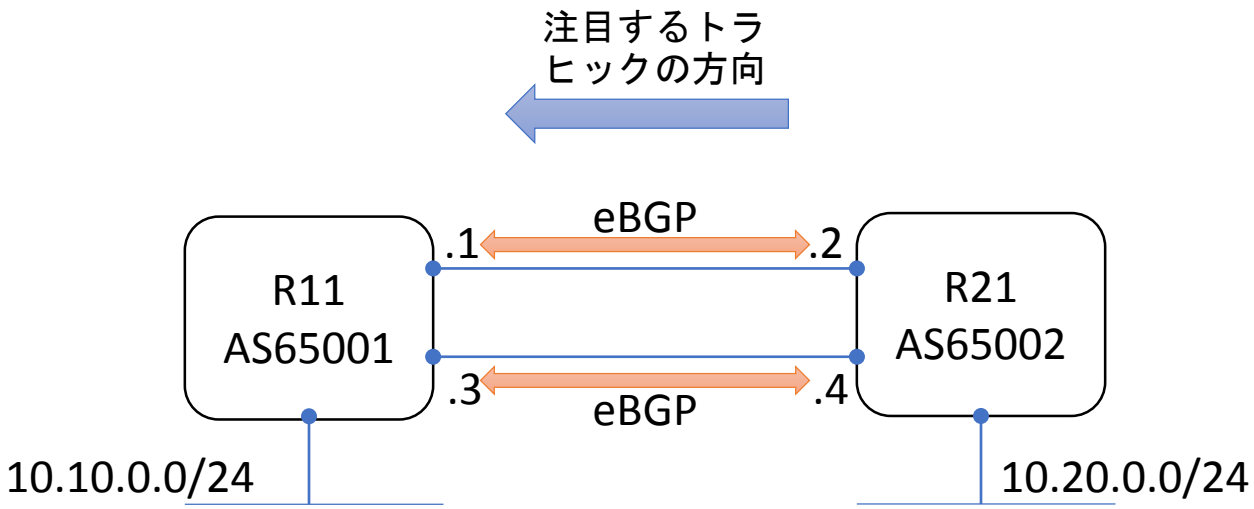
- AS65002に1本目を優先して貰うために複数の方法がある。

R21の持つ10.10.0.0/24への経路情報

| ベストパスの決まり手 | 調整内容 | Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|---|---|
| ASパス(長) | 2本目で広告する経路に自AS番号を1個プリペンド | * 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| | | 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 65001 | |
| MED | 1本目で広告する経路のMEDは100に。2本目は200に。 | * 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | 100 |
| | | 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | 200 |

# 経路制御の例1(6)

- 2本目での広告経路にASパスプリペンド




R11の持つ10.20.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| *10.20.0.0/24 | 172.16.0.2 | 200 | 65002 | |
| 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |

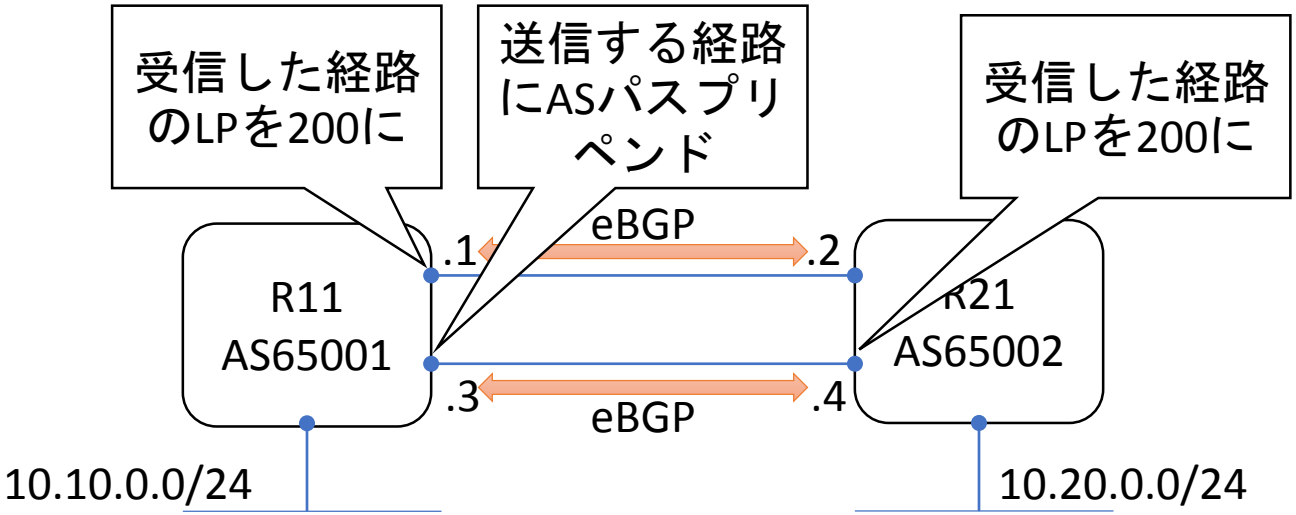
R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| * 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 65001 | |

# 経路制御の例1(6)

これでめでたし?

AS65002がLocal Pref.を設定していたら?




R11の持つ10.20.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| *10.20.0.0/24 | 172.16.0.2 | 200 | 65002 | |
| 10.20.0.0/24 | 172.17.0.4 | 100 | 65002 | |

R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| 10.10.0.0/24 | 172.16.0.1 | 100 | 65001 | |
| *10.10.0.0/24 | 172.17.0.3 | 200 | 65001 65001 | |

# 経路制御の方法の善し悪し

- Local Preferenceでの優先は、強いので相手から送られてきた属性情報(AS_PATH、MED)が評価されない
  - しかも、相手には伝わらない
  - しかし、確実に経路を優先できる
- ASパスプリペンドやMEDを調整しての広告であれば、相手に伝わる(こっちから入ってこないでね)
  - 数値次第で自身の思いを強くできる
    - 相手がプリペンドしてきた以上に他方にプリペンドする
    - 相手がMEDの値につけてきた差以上にMEDを加算する
- 経路の送信側、受信側であらかじめ取り決めておいたCOMMUNITY属性の値で制御することもある

# LP、ASパス長、MEDの使い分け

- 強さ(ベストパスセレクションでの評価順位)は、Local Preference > ASパス長 > MEDの順
- A、Bの二つの経路があった場合のポリシ実装の例

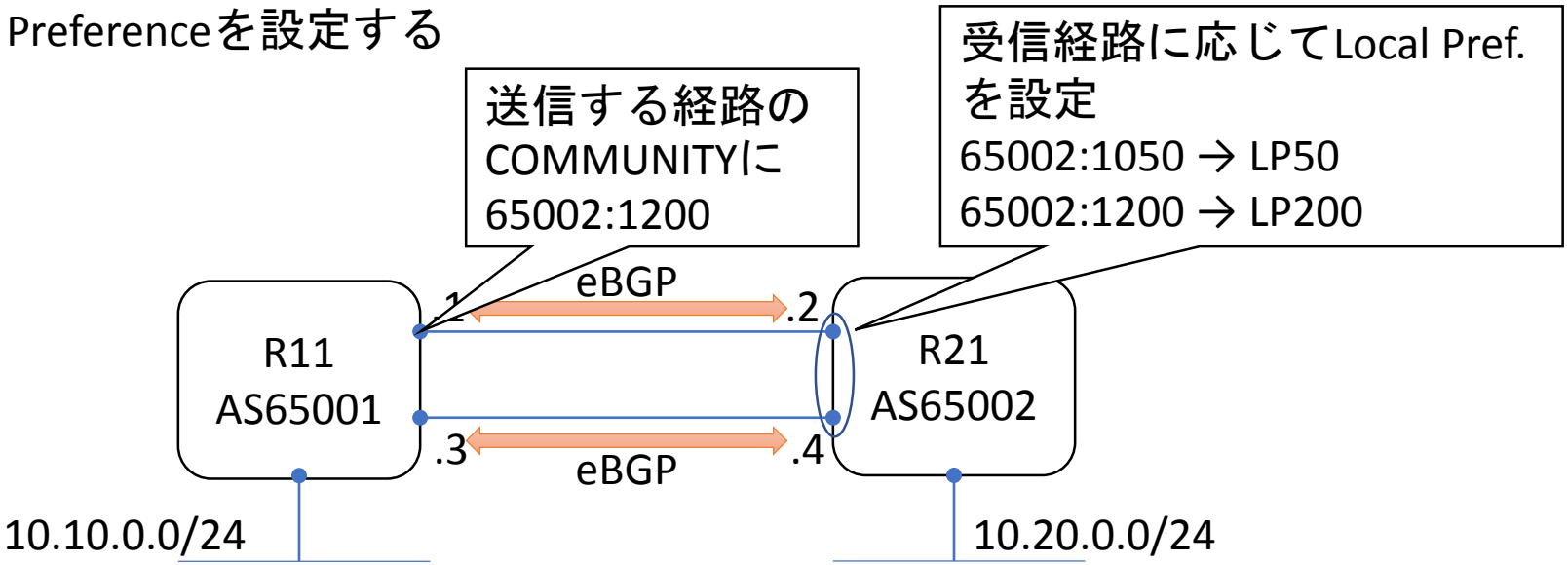
| 実現するポリシ | 属性の操作、設定 |
|---|---|
| Aが存在する以上、AはBより優先する | AのLocal Pref.をBのより大きくする。 |
| AとB、ASパス長が同じであれば、Aを優先 | AとBのLocal Pref.を同じに。<br>AのMEDをBより小さく設定する。 |
| AとB、ASパス長が同じ場合は、相手のMEDを尊重する | AとBのLocal Pref.を同じに。<br>相手が設定したMEDは上書きしない。<br>または、相手が設定したMEDの値の大小関係を維持したまま書き換え |

# COMMUNITY属性を使った制御(1)

- COMMUNITYは、Well-knownでNO_EXPORTとかNO_ADVERTISEなどある。
  - もともと、2バイトASを意識した2バイト:2バイト(RFC1997)
  - Well-knownの上位2バイトは、65535(上位65535は予約)
  - 4バイトAS対応のため、RFC4360、RFC8092など提案あり
- 複数個のCOMMUNITY属性を持つことが出来る
  - 例1: NO_EXPORT
  - 例2: 65001:0
  - 例3: 65001:0 65534:100 NO_ADVERTISE

# COMMUNITY属性を使った制御(2)

- 例: 経路の送信側がつけてきたCOMMUNITYに応じて、当該経路のLocal Preferenceを設定する




R21の持つ10.10.0.0/24への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED | COMMUNITY |
|---|---|---|---|---|---|
| 10.10.0.0/24 | 172.16.0.1 | 200 | 65001 | | 65002:1200 |
| *10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | | |

# COMMUNITY属性を使った制御(3)

- 自身で定義して使用する場合は、意味を決めてまとめておく
  例

| 値 | 意味 |
|---|---|
| 65001:0 | クラウドに広告する経路 |
| 65001:1001 | オンプレ拠点1の経路 |
| 6500:1002 | オンプレ拠点2の経路 |
| 65001:2000 | クラウドとの東日本の接続で得た経路 |
| 65001:2100 | クラウドとの西日本の接続で得た経路 |

- 組み合わせて、効率よくフィルタリング
  - クラウドに広告し、かつ、オンプレ拠点1の経路
    → COMMUNITY 65001:0と6501:1001が付いている経路
  - オンプレ環境1では、東日本での接続で得た経路を優先
    →オンプレ環境1のルータは、65001:2000の付いた経路のLocal Preferenceを上げる

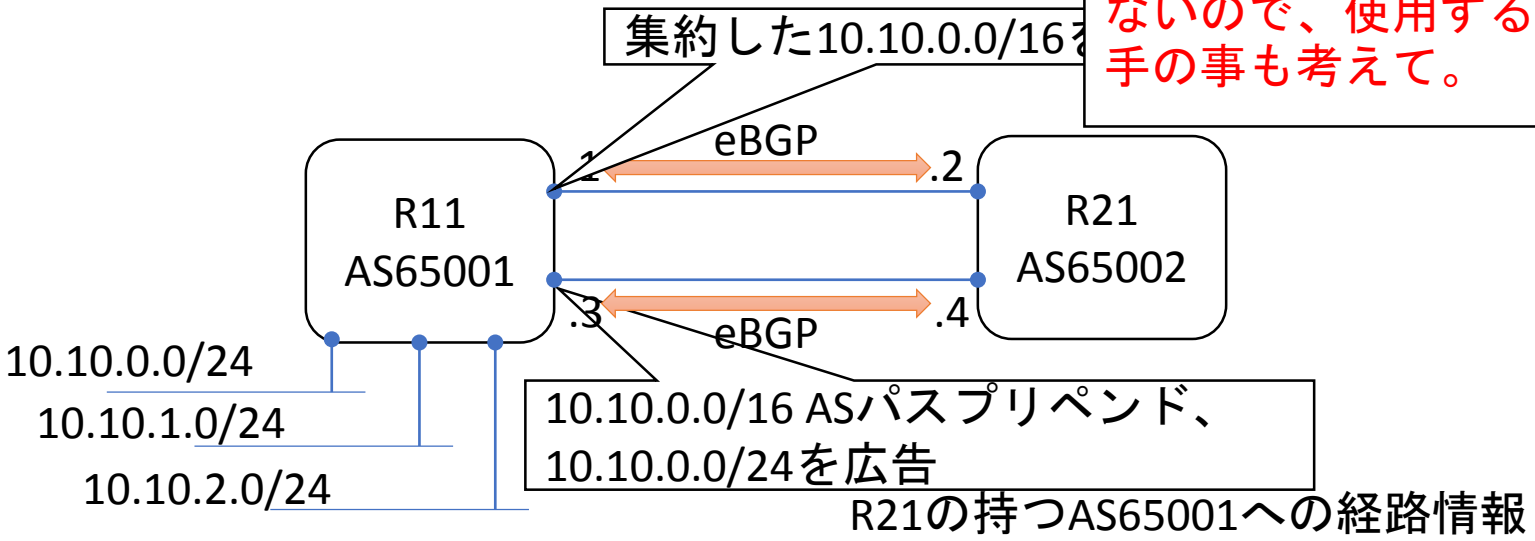
# more specific routeでのルーティング制御

- 最終手段、かなり強力
- 詳細経路とも呼ぶ
- インターネットでは、あまりやらない方が良いし、やる場合は十分理解しておこなう
  - IRS26 「longer経路を一番安全にwithdrawする方法選手権」参照
- 自ネットワークの拠点間接続や、クラウドくらいなら良いか
  - 伝わっていかなければならないネットワークが小さく、反映が早いため
    - この反映にかかる時間は、MRAIによるものが多くを占めるのでは?
  - 経路数が増えても、経路数が増えるのは自分に関係あるネットワークだけである
- BGPで制御ではないとも言える
  - IPでのルーティングの原則「最長一致」での制御

# 経路制御の例2

- AS65002から10.10.0.0/24宛てだけは、2本
  い。2本目のリンクが落ちている場合は、
  に。

ベストパスセレクションではなく、経路の存在で決まる。このようにされると、AS65002は10.10.0.0/24宛てを1本目のリンクに通すすべはないので、使用する際には相手の事も考えて。




R21の持つAS65001への経路情報

| Prefix | Next hop | Local Pref. | AS PATH | MED |
|---|---|---|---|---|
| * 10.10.0.0/16 | 172.16.0.1 | 100 | 65001 | |
| 10.10.0.0/16 | 172.16.0.3 | 100 | 65001 65001 | |
| * 10.10.0.0/24 | 172.17.0.3 | 100 | 65001 | |

# 実践編
# 既存ネットワークのへのBGP接続機器の導入

# ネットワークへBGPの導入

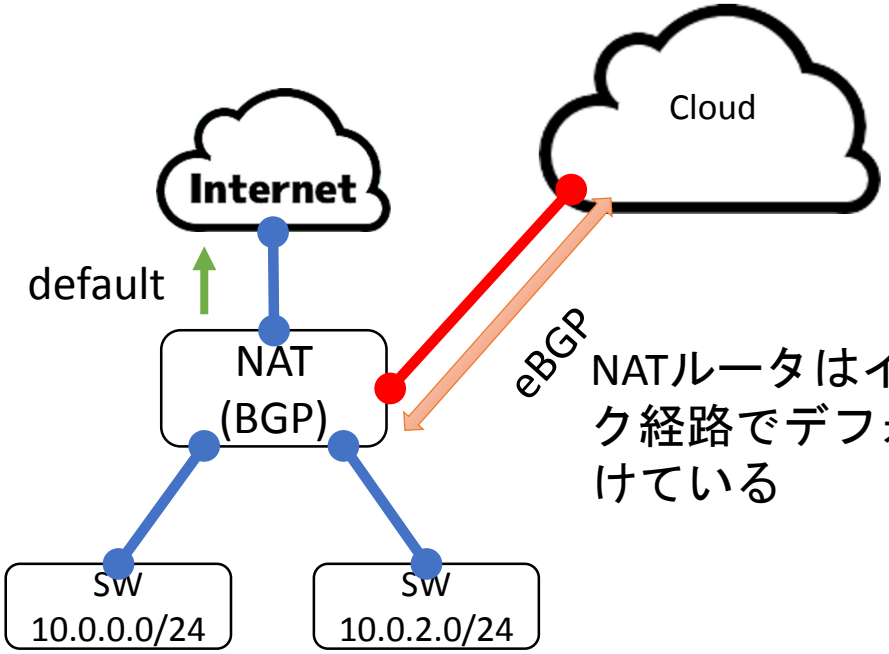
- 構成パターン1
  - ネットワークのL3機器が1台で、それでBGPを有効にする
  - 他との接続にeBGPを用いるだけ

- 構成パターン2
  - ネットワークのL3機器が複数で、それら全てでBGPを有効にする
  - 全機器がBGPでの経路を持つので実は罠が少ない

- 構成パターン3
  - ネットワークのL3機器が複数で、一部の機器にBGPを有効にする
  - IGP onlyの機器からBGP機器までIGPでルーティング
  - BGP経路情報のIGPへの再配布や、BGP機器をデフォルトルートで到達する場所に置く
  - BGP機器からネットワークまではIGPでルーティング
  - すこし複雑。しかし、これが一番多いか?

# 題材とするネットワーク1

- NATルータを介してインターネットに接続するネットワーク
- L3機器はNATルータ1台だけ。
- NATルータはBGPに対応していた。
- そこで、NATルータとCloud間で接続を設け、BGP接続する形とします




NATルータはインターネットにスタティック経路でデフォルトルート(0.0.0.0/0)を向けている

# 題材とするネットワーク1
# 行う事概要

- 決めるべき事を決める、または得る
  - クラウド側のネットワークのアドレスブロック
  - クラウド側のAS番号
  - 自身の側のAS番号
  - 通信できるネットワークとその実現方法の決定
    - オンプレ、クラウドのどのネットワーク間で通信が出来るようにするか?
    - 経路での制御と、フィルタでの制限またはその組み合わせで制御
  - 交換する経路の決定
- 接続設定

# クラウド側のアドレスブロックの決定

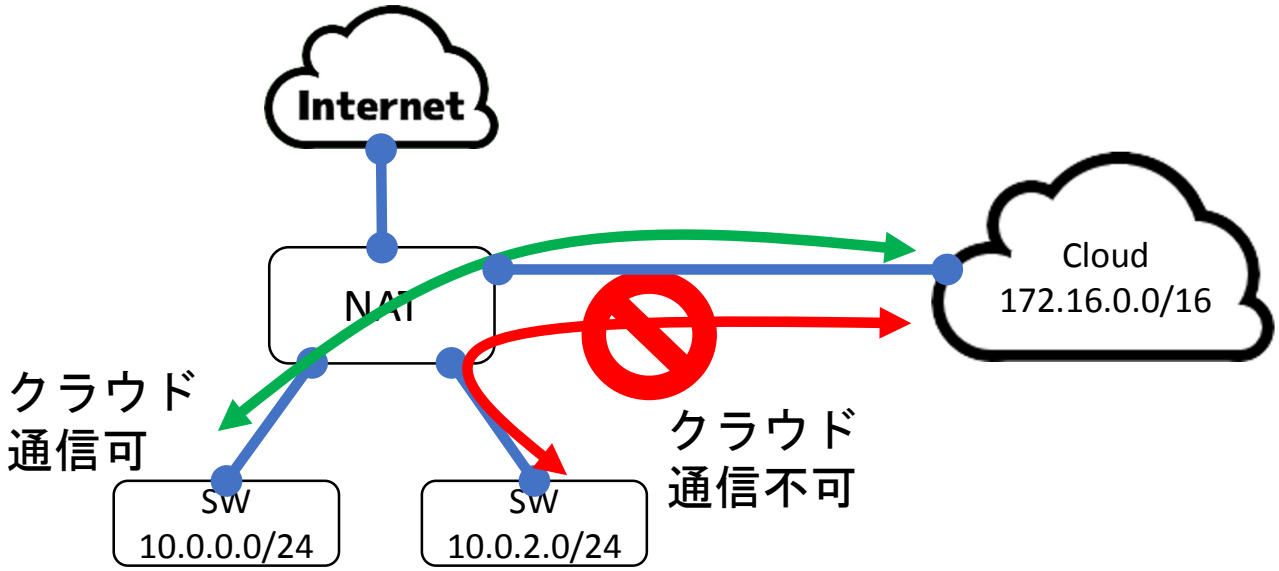
- 基本として、オンプレ側とクラウド側でかぶらないように。
- 実はかぶってしまっていたら、詳細経路で問題なくルーティング出来る場合も
- どうしようもなければリナンバリング

# 使用するAS番号の決定

- (おそらく)eBGPなので、その場合はクラウド側とオンプレ側で異なるAS番号を使用する
- グローバルAS番号を割り当てられていない組織であれば、プライベートAS番号を用いる
  - 64512～65534(2バイト)
- オンプレ側が1拠点しかないのであれば、考えることは少ない
- オンプレ側が複数拠点となるのであれば、オンプレ側複数拠点を同一ASにするか、異なるASにするか考える
  - オンプレ1－クラウド－オンプレ2、というルーティングを行うのであれば、オンプレ1とオンプレ2のAS番号は別に。
  - 同じだと経路ループと判定されてしまう
- グローバルAS番号を割り当てられているネットワークでも、そのグローバルAS番号を使わなければならないわけではない

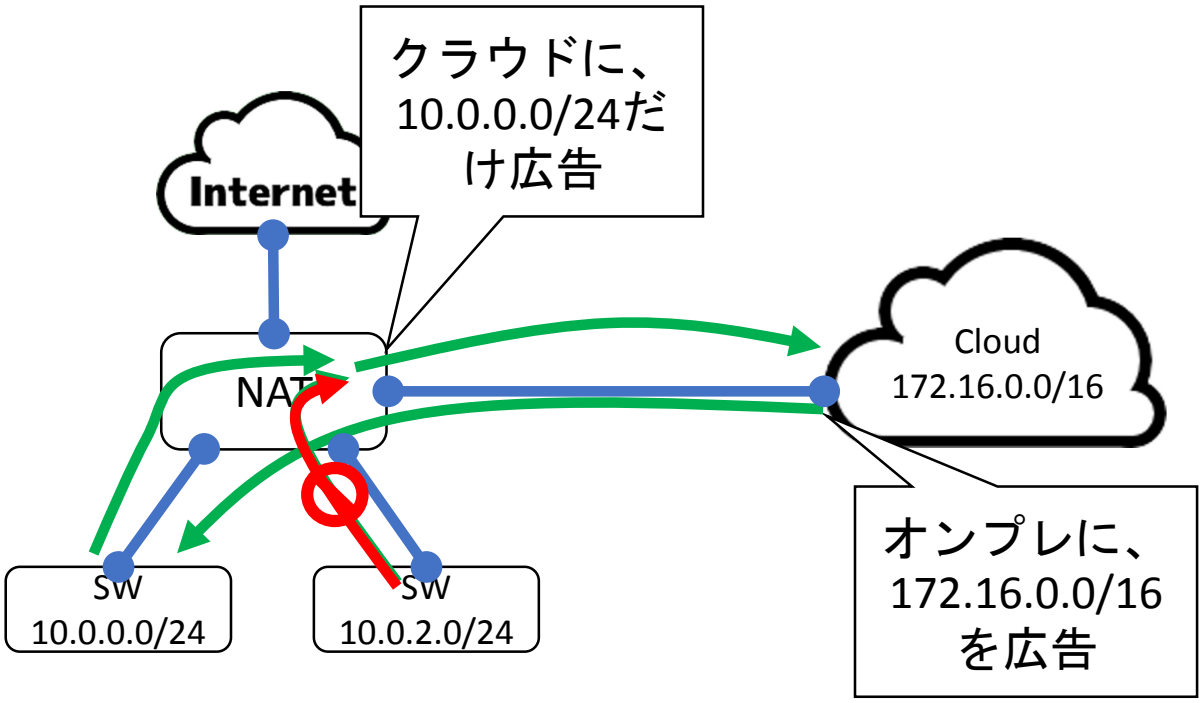
# 通信可能なネットワークの決定

- 次としたい、とする
  - オンプレ側10.0.0.0/24とクラウド172.16.0.0/16は通信可能
  - オンプレ側のその他のネットワークはクラウドと通信不可能
  - 「通信可能」「通信不可能」とあえてふわっとした表現をしています



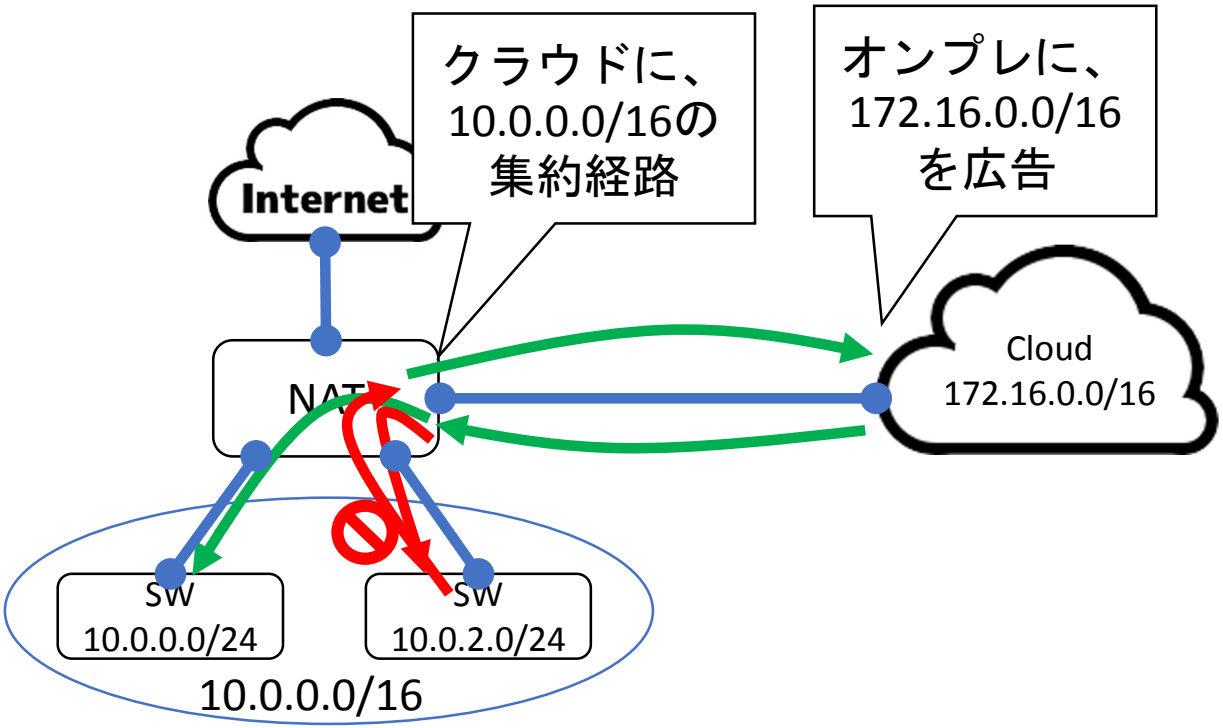
# ネットワークのアクセス制御の実装(1)

- BGPの話をしてきたので、まずはクラウドに伝える経路(プレフィックス)での制御
- しかし、この経路広告では10.0.2.0/24から172.16.0.0/16にパケットを送ることが出来る。不十分であれば、NATルータやクラウドの機能でパケットフィルタリング



# ネットワークのアクセス制御の実装(2)

- どうせパケットフィルタ使うなら、全ネットワーク間ルーティング可能にする方法も
- NATルータの入のパケットフィルタでの制限では、クラウドから10.0.0.0/16へのパケットはオンプレ-クラウド間リンクには流れるので帯域の浪費、課金に注意



# 広告する経路の決定

- オンプレからクラウドに広告する経路は、通信可能なネットワークの要件から決まることもある。
- しかし、オンプレ側にたくさんのネットワーク(プレフィクス)がある場合、クラウド側で受け入れ可能なプレフィックス数を越える可能性がある。
  - その場合は、経路集約せざるを得ない

10.0.0.0/24 ～ 10.0.7.0/24の8個のネットワークのうち、
10.0.2.0/24ではないプレフィックスを広告する場合の集約の例

| 集約前 7プレフィックス | 集約後 3プレフィックス |
|---|---|
| 10.0.0.0/24、10.0.1.0/24 | 10.0.0.0/23 |
| 10.0.3.0/24、 | 10.0.3.0/24 |
| 10.0.4.0/24、10.0.5.0/24、10.0.6.0/24、10.0.7.0/24 | 10.0.4.0/22 |

または、10.0.0.0/21に集約して10.0.2.0/24宛てはパケットフィルタ

# 設定:BGPへの経路の載せ方(1)

- Cisco IOSでは、ルーティングテーブル(show ip route)とは別に、BGPテーブル(show ip bgp)があり、BGPで広告する経路はBGPテーブルに載っていなければならない。
- JUNOSでは、BGPテーブルには別れていないので別の考え方となる。(ネイバとの間の出のフィルタでの、他のプロトコルからの出の許可)

| 方法 | コンフィグ例(Cisco IOS) | 備考、注意点 |
|---|---|---|
| BGPではないプロトコルにより作成されたプレフィックスを個別にBGP経路にする | router bgp 65001<br> address-familyipv4<br>  network 10.0.0.0 mask 255.255.255.0<br>  network 10.0.2.0 mask 255.255.255.0<br>  network 10.0.3.0 mask 255.255.255.0<br> exit-address-family<br>! | ・載せるプレフィックスが多くなると、コンフィグの行数がたくさんに |
| オンプレのネットワークを内包する経路をスタティックつくってBGP経路にする | ip route 10.0.0.0 255.255.0.0 null 0<br>router bgp 65001<br> address-family ipv4<br>  network 10.0.0.0 mask 255.255.0.0<br> exit-address-family<br>! | ・shorterの経路をBGPに載せる方法としては、シンプルではある。<br>・当該機器が10.0.0.0/16に含まれるネットワークへの到達性がなくならない構成であることが前提。 |

# 設定:BGPへの経路の載せ方(2)

| 方法 | コンフィグ例(Cisco IOS) | 備考、注意点 |
|---|---|---|
| IGP→BGPに経路再配布 | ip prefix-list pfx_ospf-to-bgp seq 5 permit 10.0.0.0/16 le 24<br>route-map rmap_ospf-to-bgp permit 10<br>match ip address prefix-list pfx_ospf-to-bgp<br>!<br>router bgp 65001<br>address-family ipv4<br>redistribute ospf 1 route-map rmap_ospf-to-bgp<br>exit-address-family<br>! | 多くのプレフィックスをBGPに載せたりするのには便利。 |
| 経路集約してBGPにする | IGP→BGPに経路再配布に加え、次。<br>router bgp 65001<br>address-family ipv4<br>aggregate-address 10.0.0.0 255.255.0.0 summary-only<br>exit-address-family<br>! | 内包されるプレフィックスがBGPテーブルにある場合だけ、集約経路が生成される。集約経路生成の条件付けも可能。 |

10.0.0.0/16に内包されるプレフィックス(10.0.100.0/24など)がBGPテーブルにあると、10.0.0.0/16が作成される。
summary-onlyにより、集約元となった経路は他へ広告されなくなる。

# 設定:入、出の経路のフィルタリング(1)

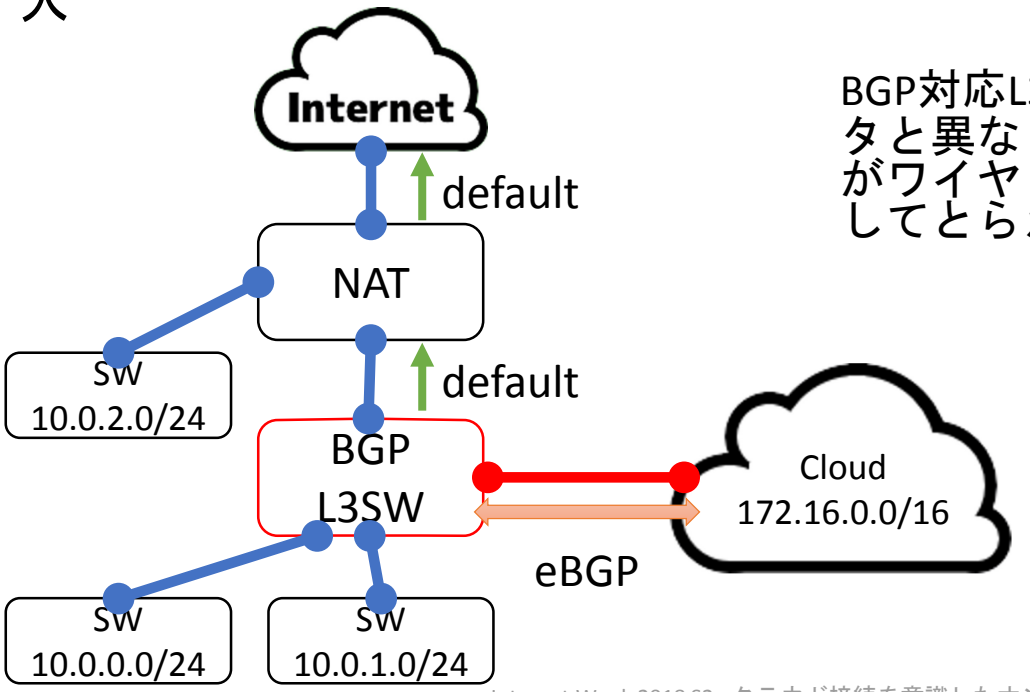
- フィルタリングと呼んでいるが、本資料では通す、落とすだけではなく属性の書き換えも含んだ広い意味で使っています。
- Cisco IOSだとroute-map、Juniper JUNOSだとpolicy-statement
- iBGPではフィルタリングは基本不要
- eBGPでは、フィルタを入れるべき
  - 他者との間の接続なので、相手に迷惑をかけないように配慮
  - 不要な経路を外に出さないよう、出のフィルタ
  - 意図しない経路を外から受けないよう、入のフィルタ

# 設定:入、出の経路のフィルタリング(2)
# フィルタリング条件

| フィルタリング条件 | 条件の例 |
| --- | --- |
| プレフィックス | プレフィックスそのもの(10.0.0.0/24)<br>プレフィックスに内包する(10.0.0.0/16 le 24)<br>など。 |
| ASパス | 正規表現で記述できるのが多い。<br>・_65002$<br>ASパスの一番右のASが65002=オリジンASが65002<br>・^65003_<br>ASパスの一番左のASが65003=この経路をくれたASが65003<br>・^$<br>ASパスが空 = 自ASで作られた経路<br>・^(65004_)+(65005_)+$<br>オリジンが65005で、その間に65004がいくつか入る。 |
| COMMUNITY | 正規表現で記述できるのもおおい。<br>値 Xがある |
| 学習元プロトコル | connected、static、OSPF |
| Local Pref.、MEDなど | |

# 題材とするネットワーク2

- 先のネットワーク構成では不十分であった
- NATルータのスループットが低く、オンプレ-クラウド間接続の必要な性能が出ない。10.0.0.0/24ネットワーク – クラウド間はGbE以上スループットを得たいなど
- そこで、新たにネットワークにもう一台のL3機器(BGP対応L3SW)を導入



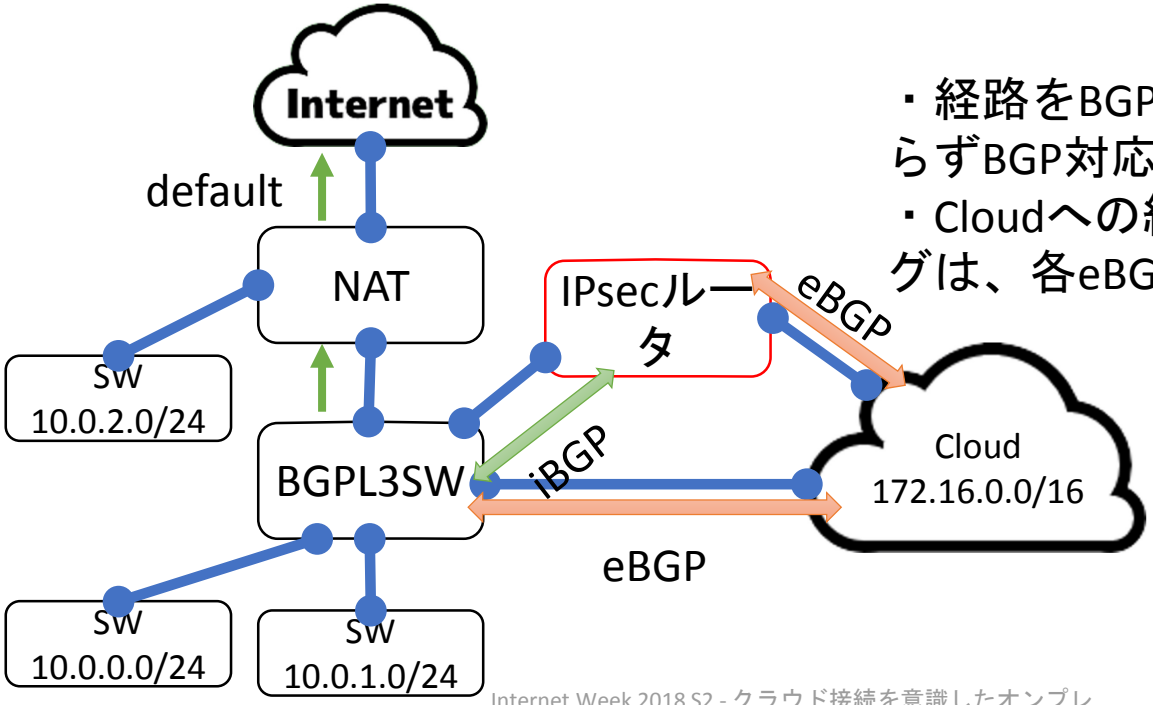

BGP対応L3SWは、NATルータと異なり、スループットがワイヤレート出る機器としてとらえてください。

# ネットワーク機器のBGPサポート

- L3機器でも、BGPをサポートしないものもある
  - 機器、ソフトウェア(ライセンス)が対応している必要あり。
    - Cisco Catalystだと、IP Services。Cisco ISRルータだとAdvanced Securityなど。
    - Juniper EXだと、EFLとAFLを追加。Juniper SRXだと基本(JSB)でサポート。
    - YAMAHA、NEC UNIVERGE IXシリーズはサポート
  - スイッチだとサービストラヒックにIPsecの適用は出来ない、と言って良いかと
  - Zebra(Quagga、FRRなど)、OpenBGPdなどPC UNIXで動くBGPデーモン

# 題材とするネットワーク3

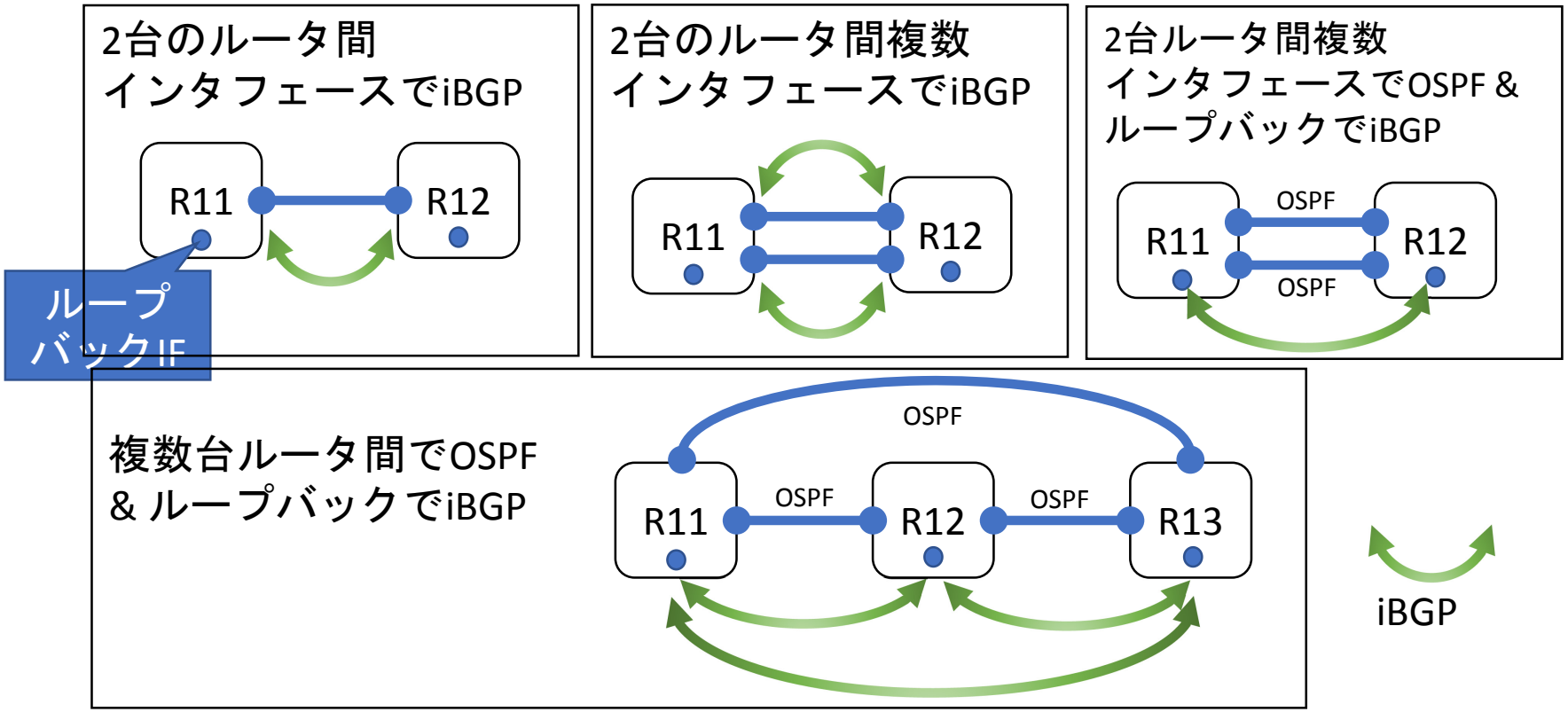
- 先のネットワーク構成では不十分であった
- クラウドとの接続は、IPsec VPN接続のバックアップも設けたい
  - BGP対応L3SWは、レイヤ3SWなので、IPsecを使えなかった
- なので、IPsec対応のルータの導入。BGP対応機器間をiBGPで接続




・経路をBGPに載せるのは、変わらずBGP対応L3SWで
・Cloudへの経路のフィルタリングは、各eBGP接続(2カ所)で。

# iBGPの使用

- iBGPセッションは、2台であれば、必ずしもループバックインタフェースで張る必要はない
- 複数台、複数リンクであれば、BGPルータ間の到達性をOSPFなどIGPで確保

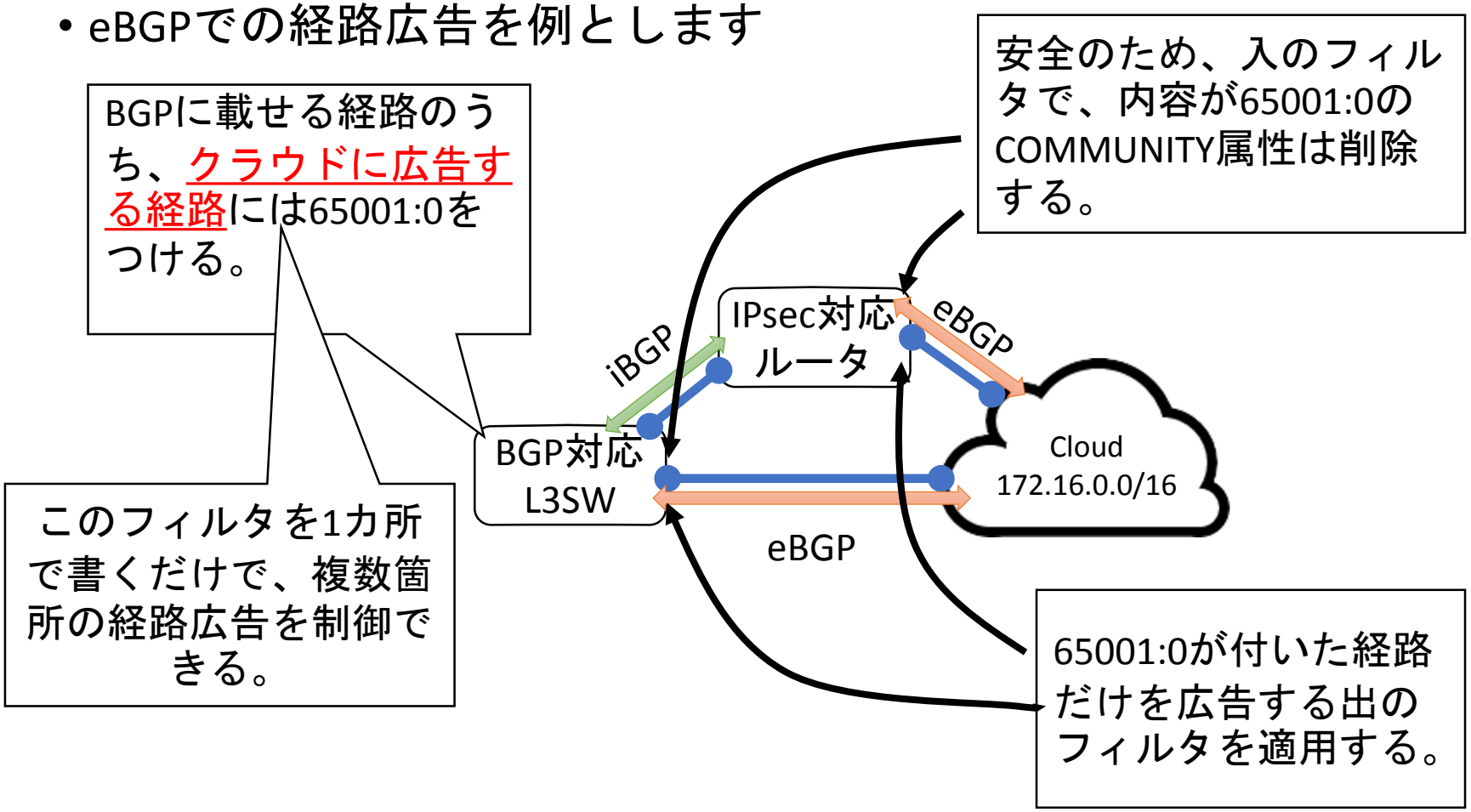


# 広告経路管理の一元化(1)

- クラウドにつながる接続が複数になった。
- 広告経路のフィルタを各クラウド接続の箇所(複数)で行う必要が
  - 各接続箇所でフィルタの設定を維持するのは、手間。忘れがち。
- そこで、COMMUNITY
  - COMMUNITY属性でクラウドに渡す経路はタグつけ
  - クラウド接続の部分のフィルタは書き換えなくて良い
- 受け取った経路にもCOMMUNITYつけておくとうれしいことがあるかも
- 使用する範囲のCOMMUNITYの値が付いた経路が外から入ってこないように注意。付いていたら、そのCOMMUNITY属性を削除。

# 広告経路管理の一元化(2)

- eBGPでの経路広告を例とします

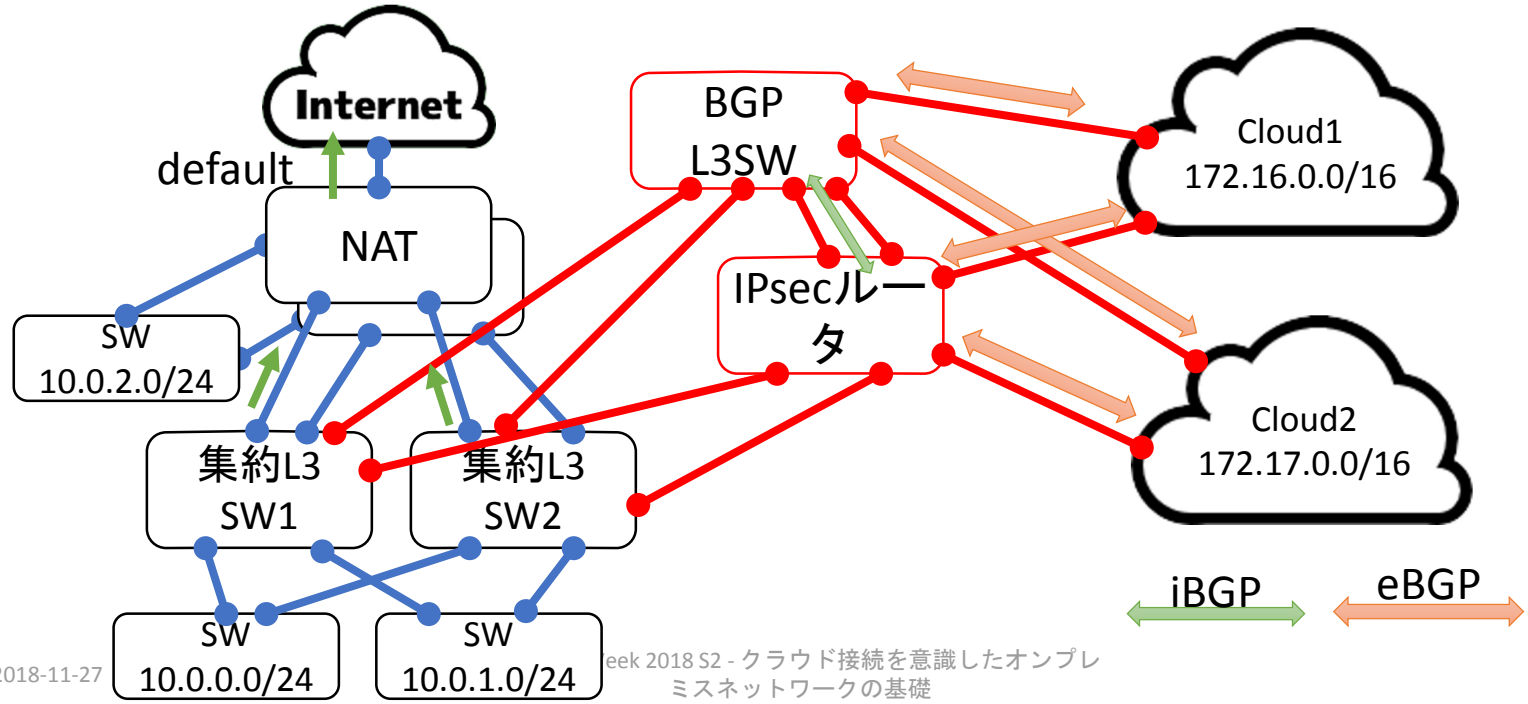


# ルータのパケット転送実装方式の注意点

- キャッシュして高速転送する実装だと、キャッシュが溢れるとスループットが低下
- ファイアウォールのように、最大セッション数などがスペックシートに記載されているような機器では無くても、似たような制限が
- インターネットからのアクセスを許していたりする＆グローバルIPアドレスがたくさんあると、スキャンによりキャッシュが大量に消費されてつらい

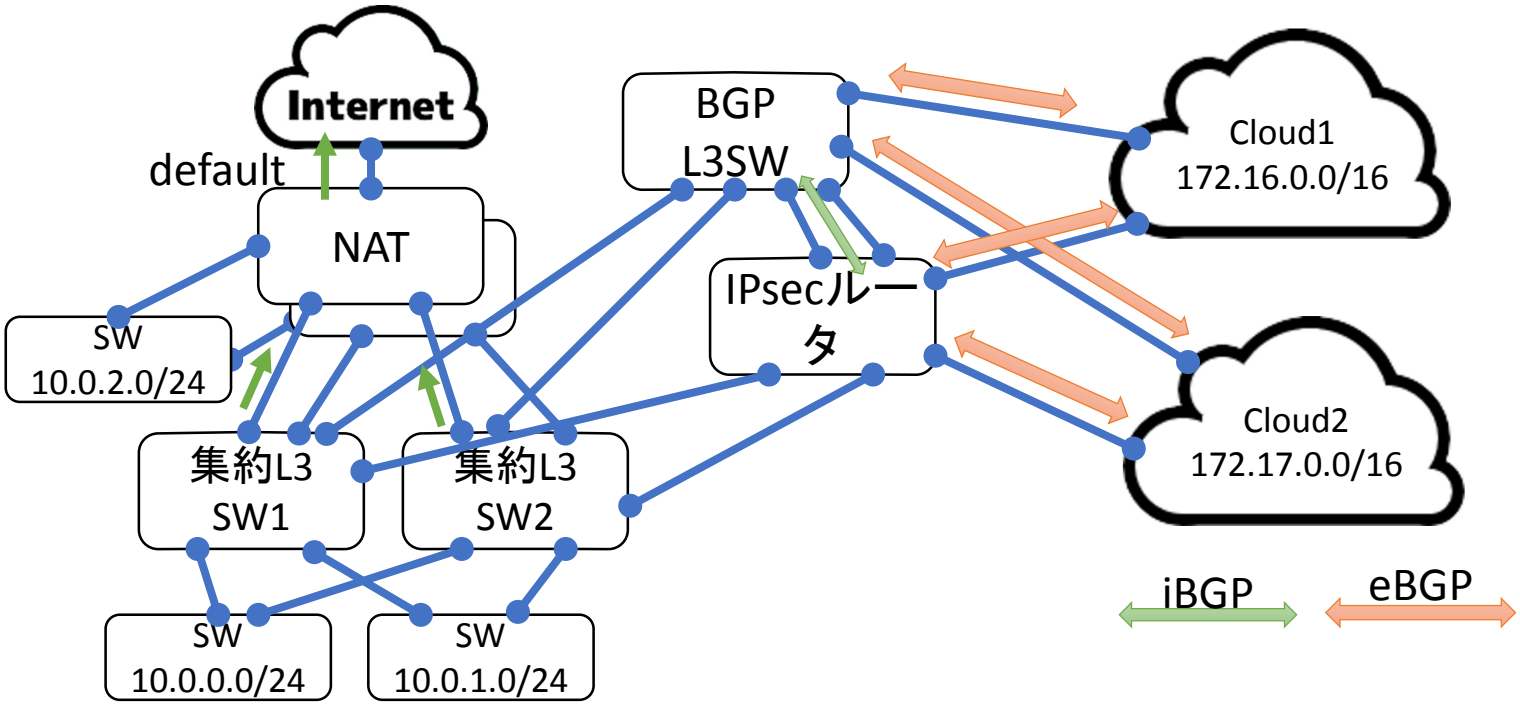
# 題材とするネットワーク4(1)

- 冗長化のされているネットワーク
- OSPFでルーティングテーブルを作っている。もともとのL3スイッチ(集約L3スイッチ)はBGP非対応であった
- 追加で、BGP接続の機器を導入してクラウドと接続
- (既存)オンプレL3機器からクラウド接続の機器へのルーティングは、BGP経路のOSPFへの再配布で対応



# 題材とするネットワーク4(2)

- 今までの例では、クラウドに到達性の必要なネットワークのトラヒックは、BGPをしゃべる機器に何もせずに到達していた
- この例では、到達しない
  - → BGP対応ルータからOSPFへの経路再配布で集約L3スイッチへクラウドへ経路を伝える



# BGP経路のIGPへの経路再配布例 Cisco IOS

- 今までの方向(IGP→BGP)とは逆
- きちんと、再配布する経路のフィルタをしましょう

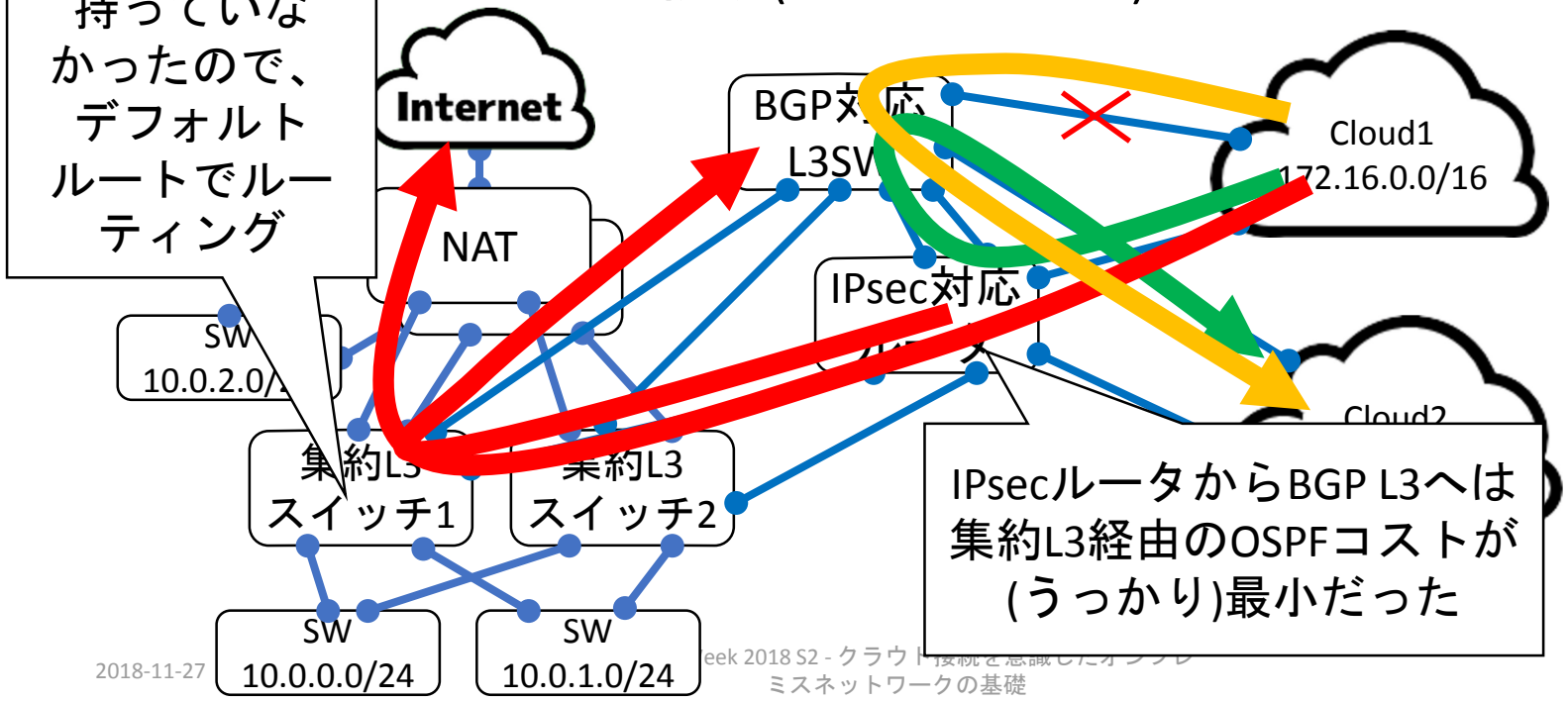
```
ip prefix-list pfx_bgp-to-ospf seq 5 permit 172.16.0.0/16

route-map rmap_bgp-to-ospf permit 10
 match ip address prefix-list pfx_bgp-to-ospf
!

router ospf 1
 redistribute bgp 65001 subnets route-map rmap_bgp-to-ospf
!
```

# 罠1:iBGPルータ間にBGP非対応機器

- 集約L3はBGPに依らないルーティングをする(OSPF経路だけでしかルーティングしない)ので、L3機器間のルーティングの一貫性が崩れる。
  → クラウド2のルーティングを考える
  のメインの接続(BGP L3SW経由)が切れていた

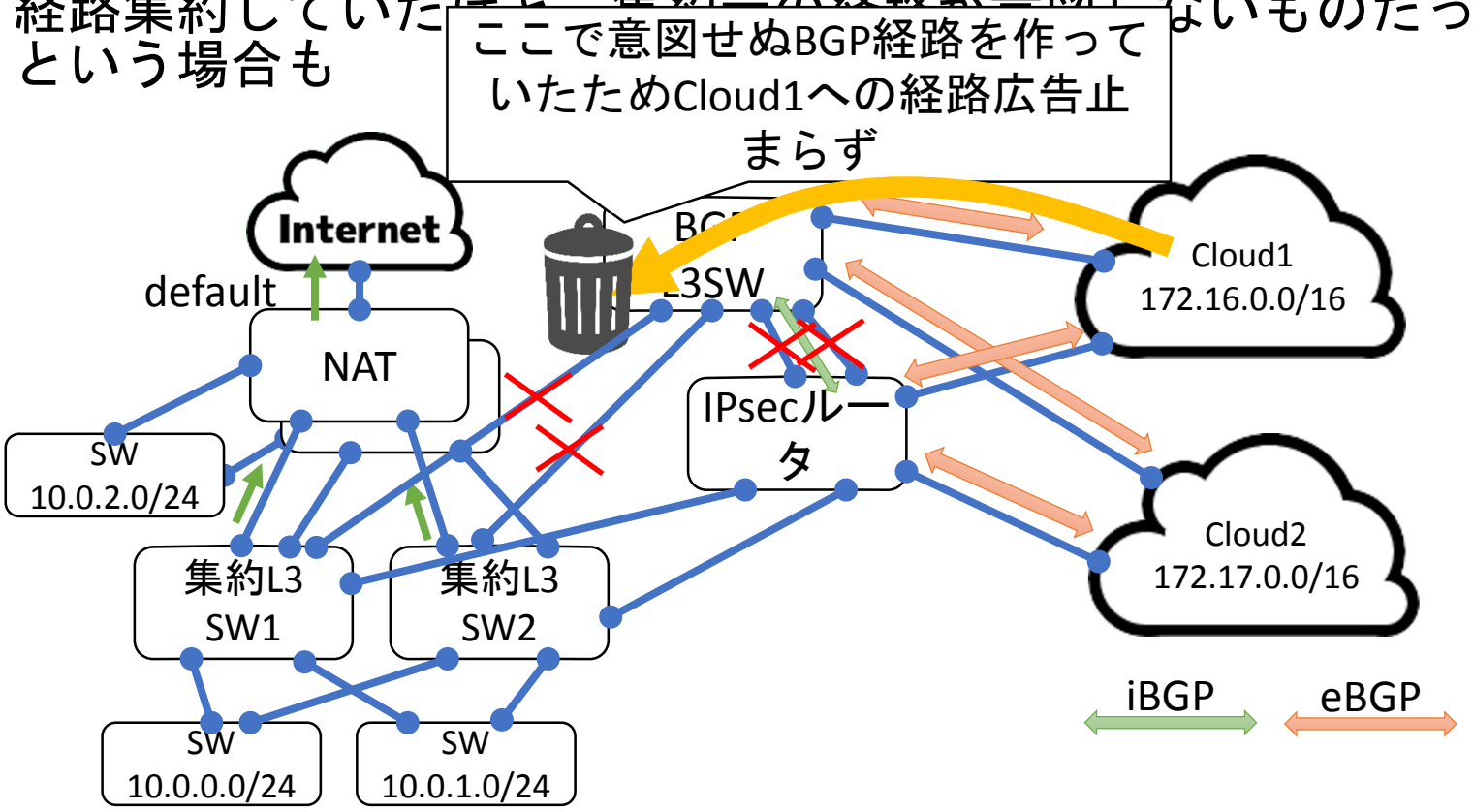


# 罠2:到達できないのに経路広告(1)

- 対外接続ルータがネットワークのコアから切り離されてしまったのに、スタティックに経路作って広告していたのでブラックホールに
- 経路集約していたけど、集約元の経路が意図しないものだったという場合も

# 罠2:到達できないのに経路広告(2)

- 対外接続ルータがネットワークのコアから切り離されてしまったのに、スタティックに経路作って広告していたのでブラックホールに
- 経路集約していたはず、集約元の経路が意図しないものだったという場合も



# 罠2:到達できないのに経路広告(3)
# 対策

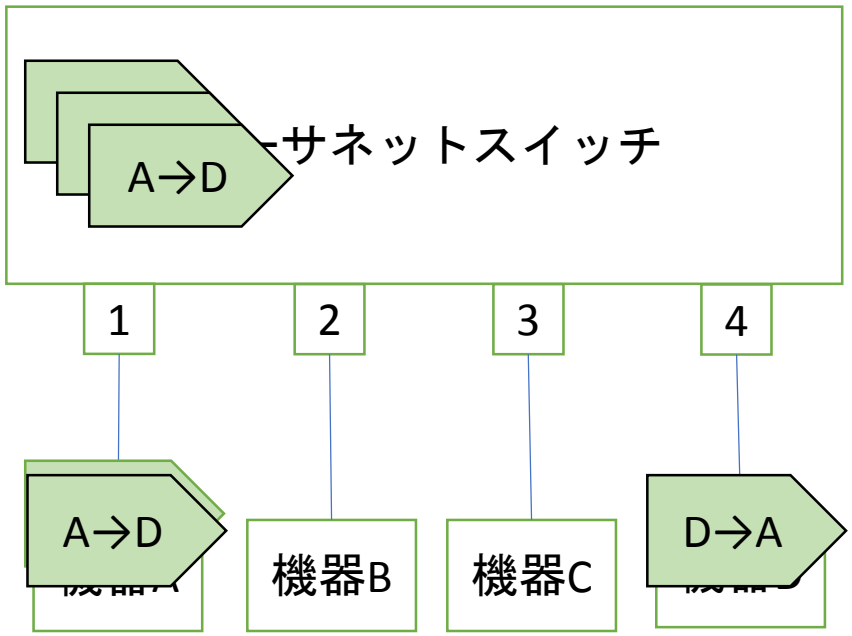
- 広告経路はコアで作る
  - エッジは、広告するだけにして、経路は作らない
- 集約経路の生成は、集約元経路をきちんとフィルタで制限する

# 罠3:BGP経路をIGPに再配布したのをBGPに再配布

- BGP経路を他のルーティングプロトコルに再配布すると、属性情報(AS_PATHも)が失われる
- それを更にBGP経路に再配布すると、ASパスが空の経路情報に
- プロトコル間の再配布は再び元のプロトコルに戻ってこないように注意
- 再配布する経路は十分注意し、フィルタリングも行う

# 罠4:L2内の非対称トラヒックによる、フラッディング(1)

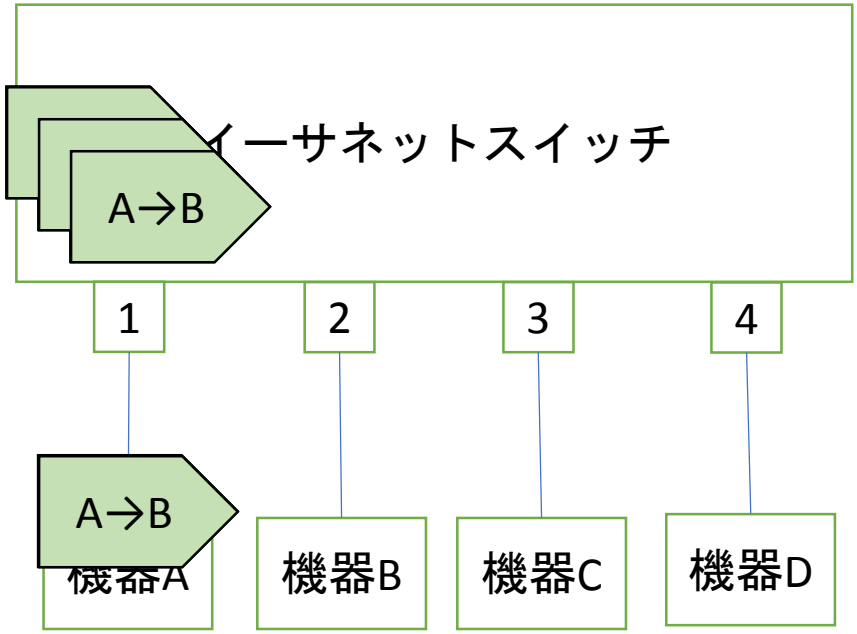
- イーサネットスイッチングの肝はMACアドレス学習であるが、MACアドレスを観測できない(学習できない)とフラッディングとなる




| インタフェース | 接続している機器のMACアドレス |
|---|---|
| 1 | A |
| 2 | |
| 3 | |
| 4 | D |

# 罠4:L2内の非対称トラヒックによる、フラッディング(2)

- イーサネットスイッチングの肝はMACアドレス学習であるが、MACアドレスを観測できない(学習できない)とフラッディングとなる

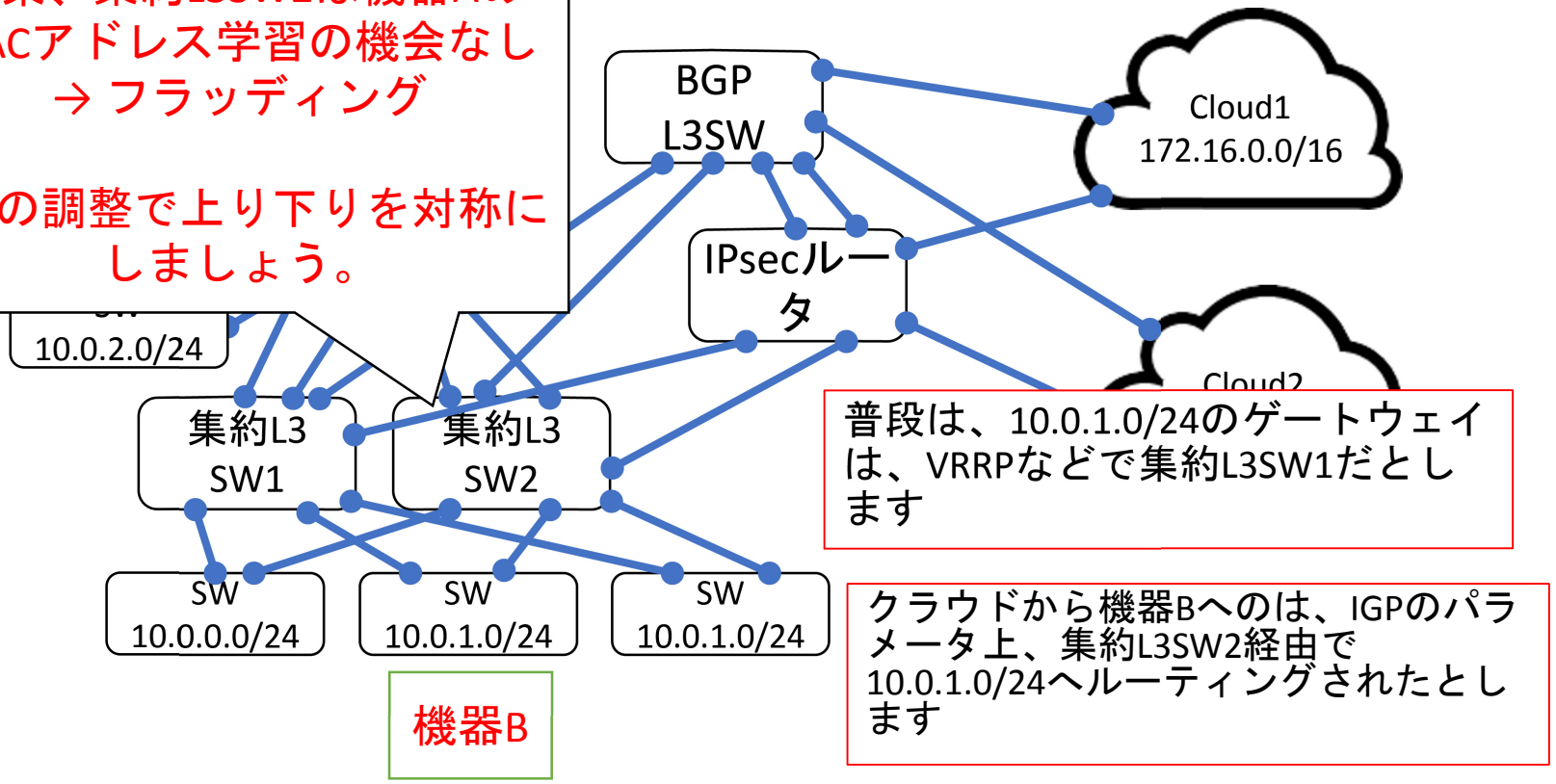



| インタフェース | 接続している機器のMACアドレス |
|---|---|
| 1 | A |
| 2 | |
| 3 | |
| 4 | D |

# 罠4:L2内の非対称トラヒックによる、フラッディング(3)

結果、集約L3SW2は機器AのMACアドレス学習の機会なし
→フラッディング

IGPの調整で上り下りを対称にしましょう。




普段は、10.0.1.0/24のゲートウェイは、VRRPなどで集約L3SW1だとします

クラウドから機器Bへのは、IGPのパラメータ上、集約L3SW2経由で10.0.1.0/24へルーティングされたとします

# おわりに