COMMUNITY アトリビュート

ここまで解説してきた WEIGHT、LOCAL_PREFERENCE、MED、AS_PATH アトリビュートはベストパス決定で利用します。ですが、COMMUNITY アトリビュートはベストパスの決定とは直接関係しません。
COMMUNITY アトリビュートを利用すると、

「特定の条件に基づいてルート情報をグループ化する」

ことができます。
グループ化したルート情報の識別情報(タグ)が COMMUNITY アトリビュートです。COMMUNITY アトリビュートは 32 ビットの数値です。しかし、32 ビットの数値をそのまま使うと分かりにくいため、COMMUNITY アトリビュートは次のように 16 ビットの AS 番号と 16 ビットの識別子を組み合わせて表記されます。

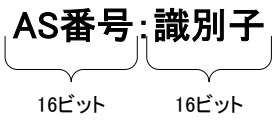

図 1 COMMUNITY のフォーマット

※ Cisco ルータでこの形式で COMMUNITY アトリビュートを扱うためには、ip bgp-community new-format コマンドが必要です。
※ 1 つのルートに複数の COMMUNITY を付加することも可能です。

COMMUNITY アトリビュートによってグループ化したルート情報をどのように扱うかは自由です。フィルタしたり LOCAL_PREFERENCE や MED などのアトリビュートを再設定したりします。

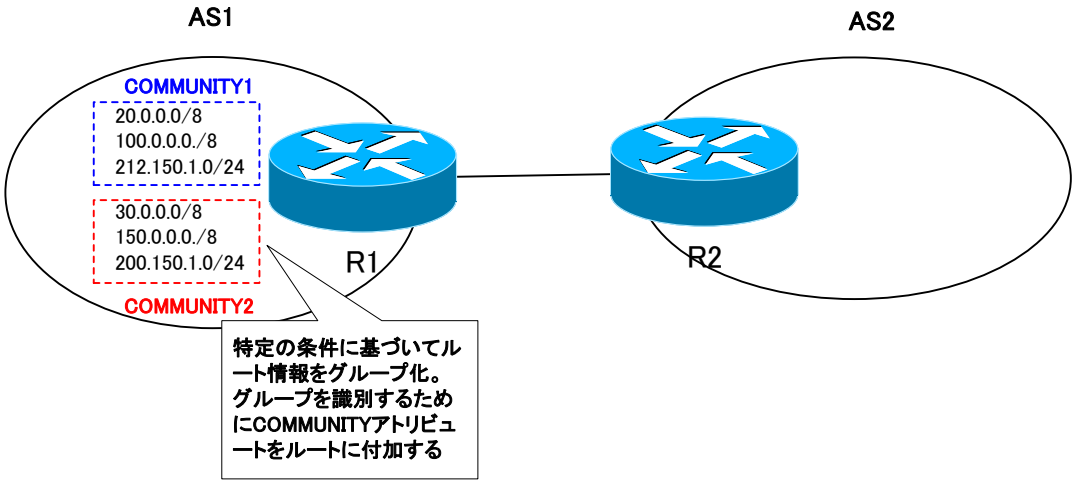

図 2 COMMUNITY の利用

COMMUNITY アトリビュートの特徴として、アトリビュートが伝達されていく範囲も注目すべき点です。LOCAL_PREFERENCE は自 AS 内、MED はネイバーAS 内というようにアトリビュートが伝達される範囲が限られています。しかし、COMMUNITY アトリビュートは、AS を越えてルート情報に付加されて伝達されていきます。そのため、離れた AS 間でのルート情報の制御に COMMUNITY アトリビュートを利用することが可能です。

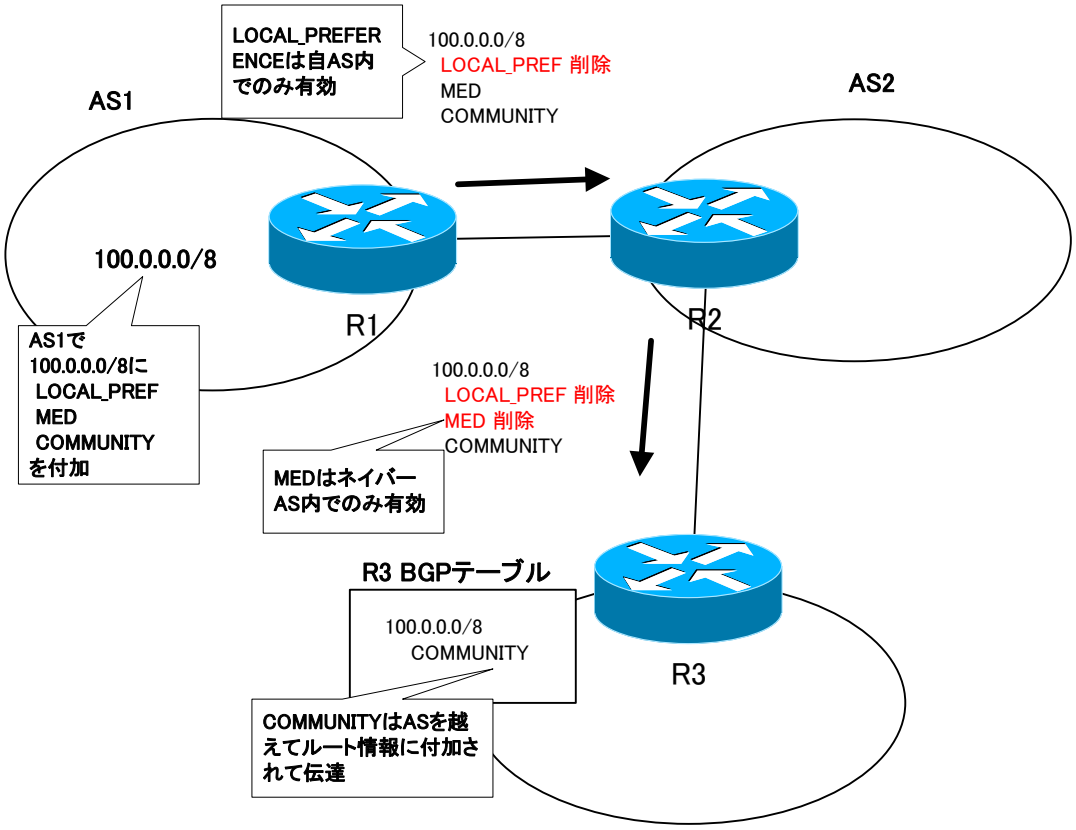

図 3 COMMUNITY の有効範囲

このような COMMUNITY アトリビュートの特徴をまとめると、次のようになります。

- 特定の条件に基づいてルート情報をグループ化する
- グループ化したルート情報の識別情報である
- AS を越えて伝達される
- 離れた AS 間でのルート情報の制御が可能になる

Well-Known COMMUNITY

COMMUNITY の値には、予約されている Well-known COMMUNITY があります。Well-known COMMUNITY をルートに付加すると、その種類に応じて**自動的にルートのフィルタ**が可能です。Well-known COMMUNITY で、**どこまで BGP ルートが伝わるようにするか**を簡単に制御することができます。

Well-known COMMUNITY の種類とその動作は次の通りです。

表 1 Well-known COMMUNITY

Well known COMMUNITY	動作	値(16 進数)
no_export	EBGP ネイバーにルートを送信しない	0xFFFFFF01
no_advertise	いかなる BGP ネイバーにもルートを送信しない	0xFFFFFF02
local_as	EBGP/IEBGP ネイバーにルートを送信しない	0xFFFFFF03

Well-known COMMUNITY とその動作を図で表すと次のようになります。

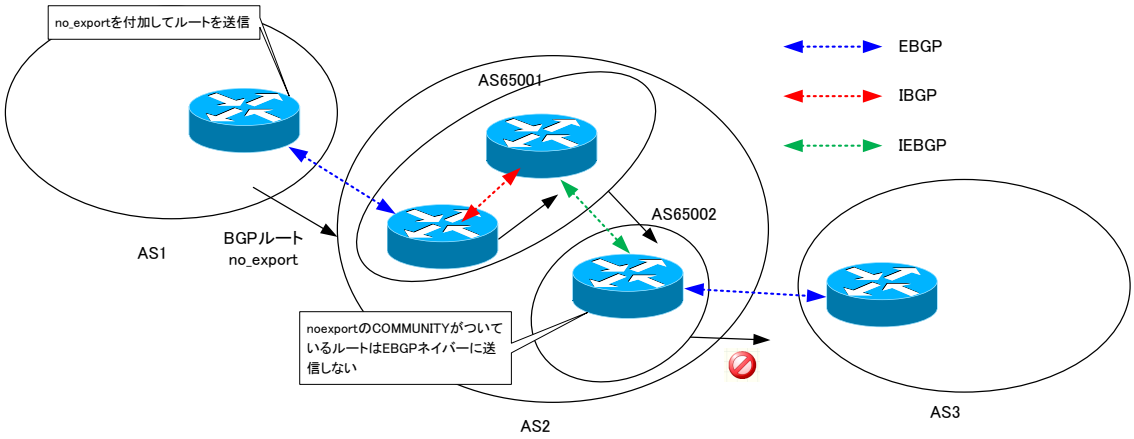

図 4 Well-known COMMUNITY no_export

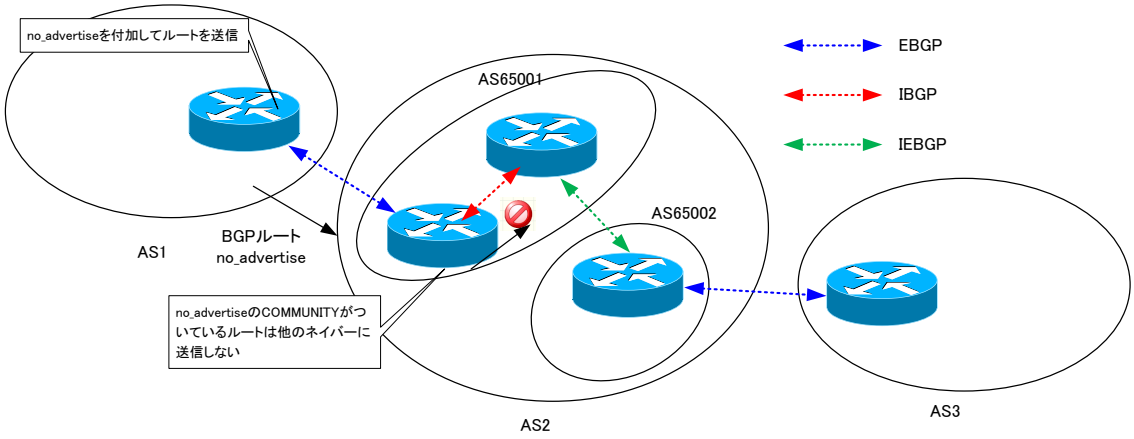

図 5 Well-known COMMUNITY no_advertise

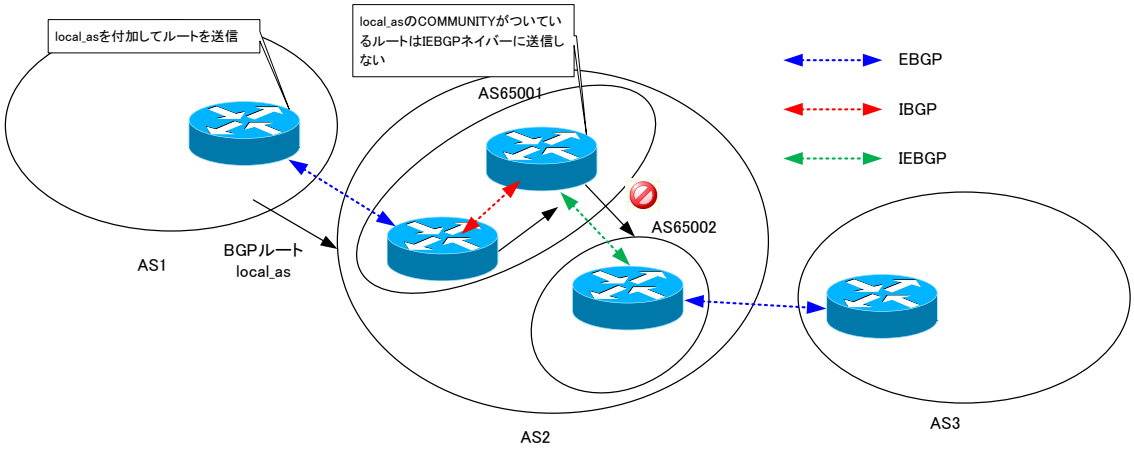

図 6 Well-known COMMUNITY local_as

COMMUNITY を付加するルータと実際にルートをフィルタするルータが違うルータであることに注意してください。ここが通常のルートフィルタが異なる点です。Well-known COMMUNITY を付加することで、他のルータにおけるルートの送信に影響を及ぼすことができます。

COMMUNITY アトリビュートを利用するための手順

Well-known COMMUNITY 以外にも任意のプライベート COMMUNITY でルート制御を行うことができます。プライベート COMMUNITY では、離れた AS 間でフィルタだけでなくアトリビュートの再設定などのさまざまな制御を行うことができます。そのためには、次のような手順で行います。

1. 制御を行いたい AS 間で協議
   - AS が異なるということは管理している組織も異なります。単体の AS だけですべて完結するというわけにはいきせん。まず、AS 間でどのようなルート情報に対してどのような制御を行うかを協議します。
   - 制御したいルート情報に対してどのような値の COMMUNITY アトリビュートを設定するかを決定します。
2. ルートの送信元 AS で COMMUNITY アトリビュートを付加
   - COMMUNITY アトリビュートの付加はルートマップで行います。ルートマップの match の条件に一致したルート情報に対して、set community コマンドで任意の COMMUNITY 値を付加することができます。あらかじめ協議した条件に基づいて、ルート情報に COMMUNITY を付加します。
   - Cisco ルータでは､デフォルトでネイバーに対して COMMUNITY アトリビュートをアドバタイズしません。ネイバーに対して COMMUNITY アトリビュートをアドバタイズするためには、neighbor send-community コマンドが必要です。
3. ルートの送信先 AS で COMMUNITY アトリビュートを参照してルート情報を制御
   - COMMUNITY アトリビュートの参照もルートマップで行います。ルートマップの match 条件で match community コマンドにより、ルート情報に付加されている COMMUNITY アトリビュートを参照できます。
   - ルートマップで COMMUNITY アトリビュートを参照して、あらかじめ協議していたルート情報の制御を行います。
   - ルート情報の制御として、フィルタを行うことが一般的です。
   - Well-known COMMUNITY では明示的な設定は不要です。前述のように Well-known COMMUNITY の種類に応じて自動的にルートフィルタが行われます。

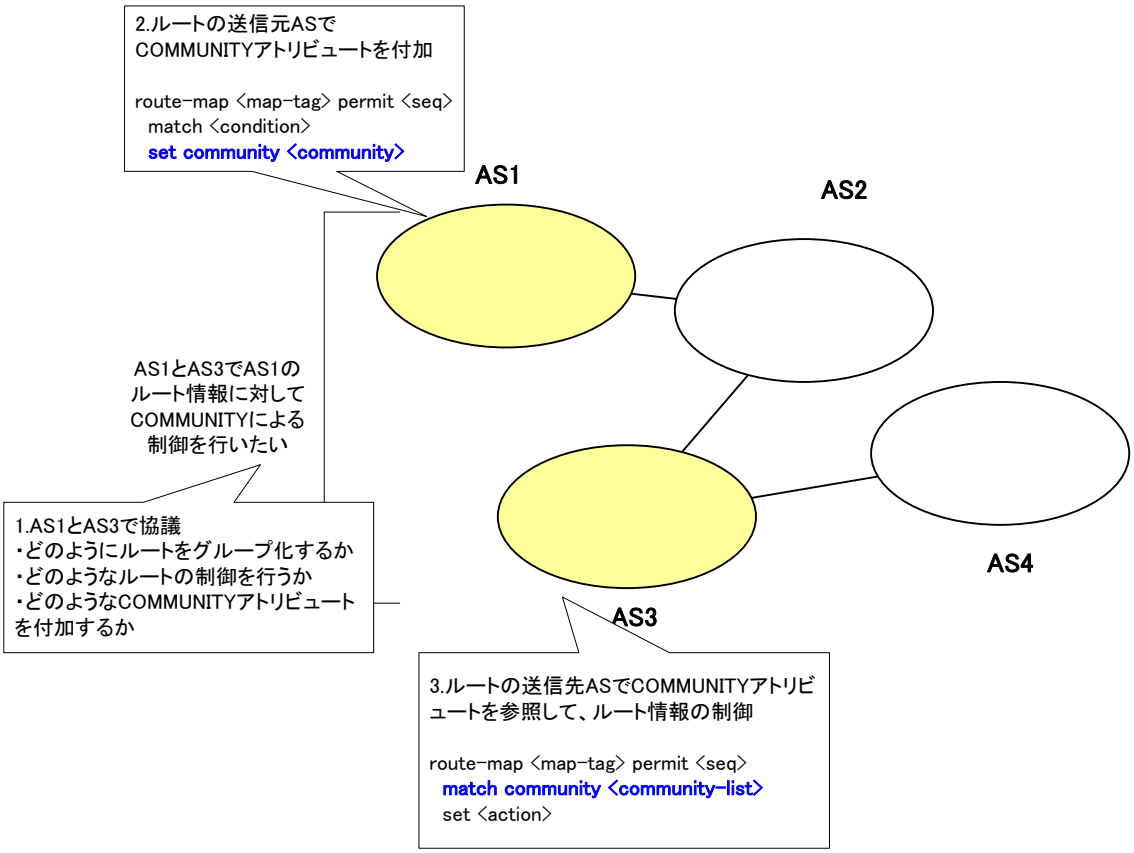

図 7 COMMUNITY アトリビュートの利用

COMMUNITY の設定コマンド

COMMUNITY を利用する場合の設定コマンドを紹介します。

- set community <community> {additive}
  ルートマップ内のコマンドです。match 条件に一致したルートに対して指定した COMMUNITY を付加します。すでに何らかの COMMUNITY が付加されている場合、上書きされます。additive のオプションがついていれば、既存の COMMUNITY に追加で新しい COMMUNITY を付加します。
  set community none とすると、現在ルートに付加されている COMMUNITY アトリビュートを削除します。
- set comm-list <community-list-num> delete
  コミュニティリストで指定された COMMUNITY 値を削除します。
- match community <community-list-num>
  ルートマップ内のコマンドです。COMMUNITY に基づいたアクションを行うルータでルートに付加されている COMMUNITY を参照するために使います。実際には、コミュニティリストを作成し、そのコミュニティリストによって COMMUNITY を参照します。
- ip community-list <num> permit <community>
  グローバルコンフィグレーションモードのコマンドです。参照したい COMMUNITY を指定するためのリストです。
- neighbor <ip-address> send-community
  BGP の設定モード内のコマンドです。Cisco ルータのデフォルトの動作では、BGP ルートに付加されている COMMUNITY をすべて削除してしまいます。COMMUNITY を削除せずに、ネイバーに送信するためには、このコマンドが必要です。
  COMMUNITY を利用する場合は、関連するルータすべてに設定が必要です。設定が抜けているルータが含まれると意図したとおりに COMMUNITY による制御ができませんので、注意してください。ピアグループのテンプレートに入れると設定漏れがなくなります。
- ip bgp community new-format
  グローバルコンフィグレーションモードのコマンドです。COMMUNITY の値を 32 ビットの数値の代わりに「AS 番号:識別子」の形式で扱えるようになります。
- show ip bgp <address>
  ルートに付加されている COMMUNITY を確認することができます。
- show ip bgp community [<community>|<community-list-num>]
  show ip bgp community では、何らかの COMMUNITY が付加されているルートを一

覧で表示することができます。特定の COMMUNITY やコミュニティリスト番号を指定すると、指定した COMMUNITY が付加されているルートを一覧表示します。

COMMUNITY の確認 ログ

```
R1#show ip bgp community
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.1.0/24     172.16.1.11              0             0 1 i
*> 100.1.2.0/24     172.16.1.11              0             0 1 i
*> 100.1.3.0/24     172.16.1.11              0             0 1 i
R1#show ip bgp 100.1.1.0
BGP routing table entry for 100.1.1.0/24, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to any
peer)
Flag: 0x880
  Not advertised to any peer
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-advertise
R1#show ip bgp 100.1.2.0
BGP routing table entry for 100.1.2.0/24, version 6
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised outside
local AS)
Flag: 0x880
  Advertised to non peer-group peers:
  3.3.3.3
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

```
      Community: local-AS
R1#show ip bgp 100.1.3.0
BGP routing table entry for 100.1.3.0/24, version 5
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to EBGP
peer)
Flag: 0x880
  Advertised to non peer-group peers:
  3.3.3.3
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-export
```

ケーススタディ 6: COMMUNITY

次のネットワーク構成で、Well-known COMMUNITY の動作をみていきましょう。

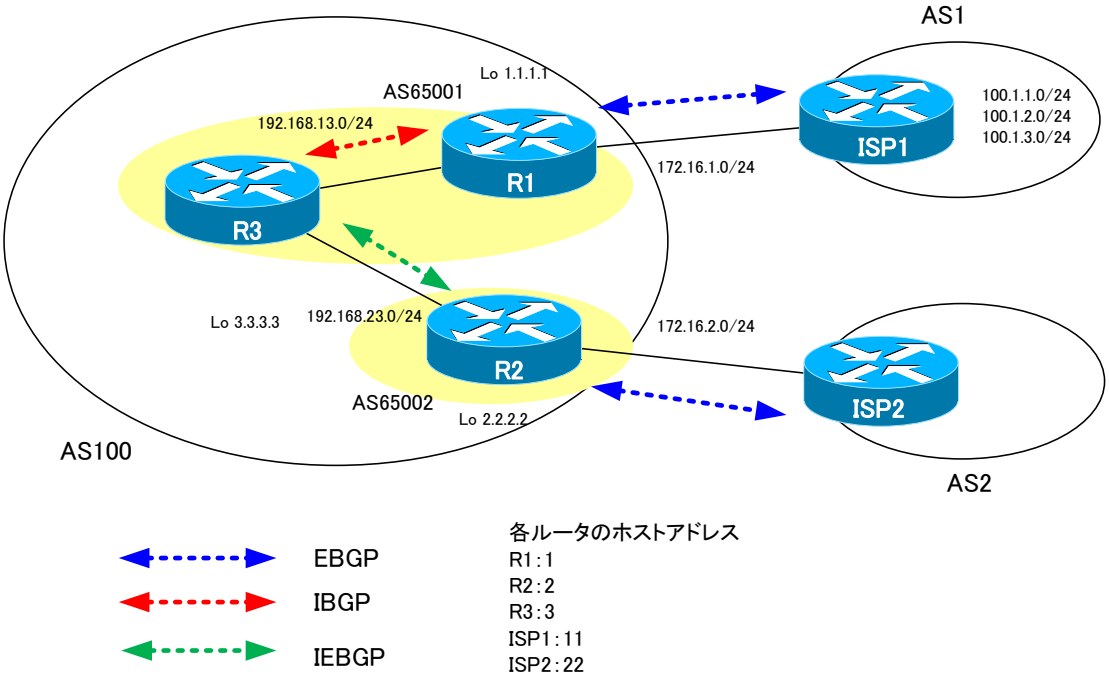

図 8 Well-known COMMUNITY 確認用のネットワーク構成

ここでは単純にするために、AS1 の ISP1 が生成する 100.1.1.0/24、100.1.2.0/24、100.1.3.0/24 の 3 つのルートだけを考えます。

Well-known COMMUNITY

ISP1 が生成するルートについて、伝わる範囲を制限します。

- 100.1.1.0/24 は R1 まで
- 100.1.2.0/24 は R3 まで
- 100.1.3.0/24 は R2 まで

これを実現するために、ISP1 は次のように Well-known COMMUNITY を付加します。

- 100.1.1.0/24 に no_advertise
- 100.1.2.0/24 に local_as
- 100.1.3.0/24 に no_export

100.1.1.0/24 は R1 から先のいかなるネイバーにも送信しないように no_advertise の Well-known COMMUNITY を付加します。100.1.2.0/24 にはコンフェデレーション構成のサブ AS から外部に出て行かないように local_as の Well-known COMMUNITY を付加します。100.1.3.0/24 には、AS100 から外に出て行かないように no_export の Well-known COMMUNITY を付加します。

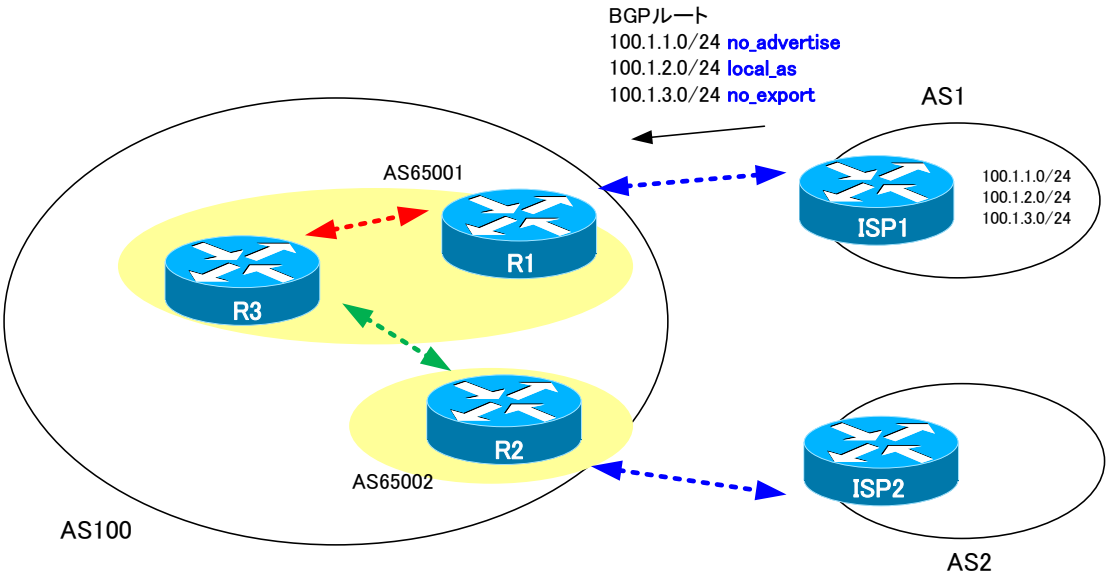

図 9 ISP1 で Well-known COMMUNITY を付加してルートを送信

ISP1 では、次のように設定して各ルートに Well-known COMMUNITY を付加します。

ISP1 Well-known COMMUNITY の付加

```
ISP1(config)#access-list 10 permit 100.1.1.0
ISP1(config)#access-list 20 permit 100.1.2.0
ISP1(config)#access-list 30 permit 100.1.3.0
ISP1(config)#route-map COMMUNITY permit 10
ISP1(config-route-map)#match ip address 10
ISP1(config-route-map)#set community no-advertise
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY permit 20
ISP1(config-route-map)#match ip address 20
ISP1(config-route-map)#set community local-AS
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY permit 30
ISP1(config-route-map)#match ip address 30
ISP1(config-route-map)#set community no-export
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY permit 1000
ISP1(config-route-map)#exit
ISP1(config)#router bgp 1
ISP1(config-router)#neighbor 172.16.1.1 route-map COMMUNITY out
ISP1(config-router)#end
ISP1#clear ip bgp 172.16.1.1 out
```

R1 で ISP1 によって付加された COMMUNITY を確認します。

R1 show ip bgp community

```
R1#show ip bgp community

R1#
```

ISP1 で付加されたはずの COMMUNITY を R1 では確認できません。これは、Cisco ルータではデフォルトでルートを送信するときに COMMUNITY を削除するからです。COMMUNITY を削除せずにルートを送信するためには、neighbor send-community の設定が必要です。ISP1 で neighbor send-community の設定を追加します。

ISP1 neighbor send-community

```
ISP1(config)#router bgp 1
ISP1(config-router)#neighbor 172.16.1.1 send-community
```

ISP1 でルートを再送信して R1 で再度 COMMUNITY を確認すると、次のようになります。

R1 show ip bgp community

```
R1#show ip bgp community
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.1.0/24     172.16.1.11              0             0 1 i
*> 100.1.2.0/24     172.16.1.11              0             0 1 i
*> 100.1.3.0/24     172.16.1.11              0             0 1 i
R1#show ip bgp 100.1.1.0
BGP routing table entry for 100.1.1.0/24, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to any
peer)
Flag: 0x880
  Not advertised to any peer
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-advertise
```

```
R1#show ip bgp 100.1.2.0
BGP routing table entry for 100.1.2.0/24, version 6
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised outside
local AS)
Flag: 0x880
  Advertised to non peer-group peers:
  3.3.3.3
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: local-AS
R1#show ip bgp 100.1.3.0
BGP routing table entry for 100.1.3.0/24, version 5
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to EBGP
peer)
Flag: 0x880
  Advertised to non peer-group peers:
  3.3.3.3
  1
    172.16.1.11 from 172.16.1.11 (111.1.1.11)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-export
```

ISP1 で neighbor send-community の設定をすれば、COMMUNITY が削除されないことがわかります。**COMMUNITY を利用した制御を行うためには、必ず neighbor send-community の設定が必要**なので忘れないように気をつけてください。ISP1 だけでなく COMMUNITY による制御を行うすべてのネイバーに対して設定してください。

※ 特に明記していませんが、R1、R2、R3 で neighbor send-community はすでに設定しています。

次に Well-known COMMUNITY によるルートの制御について確認しましょう。R1 から R3 へ送信しているルートをみると、次の通りです。

R1 から R3 へ送信しているルート

```
R1#show ip bgp neighbors 3.3.3.3 advertised-routes
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.2.0/24     172.16.1.11              0             0 1 i
*> 100.1.3.0/24     172.16.1.11              0             0 1 i
```

R1 から R3 へは no_advertise が付加されている 100.1.1.0/24 を送信していないことがわかります。

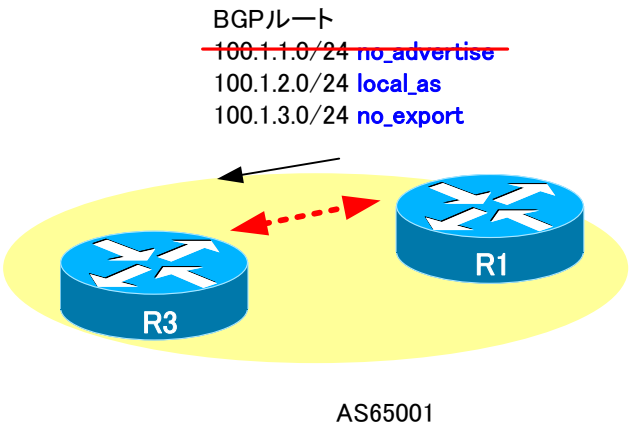

図 10 R1 から R3 へのルート送信

続いて、R3 から R2 へのルートの送信を確認します。

R3 から R2 へ送信しているルート

```
R3#show ip bgp
BGP table version is 7, local router ID is 100.100.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i100.1.2.0/24     1.1.1.1                  0    100      0 1 i
*>i100.1.3.0/24     1.1.1.1                  0    100      0 1 i
R3#show ip bgp neighbors 2.2.2.2 advertised-routes
BGP table version is 7, local router ID is 100.100.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i100.1.3.0/24     1.1.1.1                  0    100      0 1 i
```

R3 からみると R2 は IEBGP ネイバーです。そのため、local_as が付加されている 100.1.2.0/24 のルートを送信していないことがわかります。

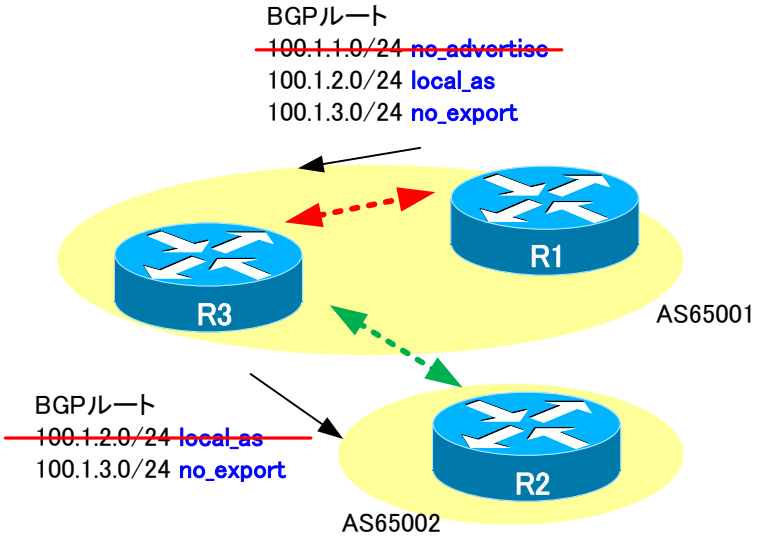

図 11 R3 から R2 へのルート送信

また、R2 から ISP2 へのルート送信は次の通りです。

R2 から ISP2 へ送信しているルート

```
R2#show ip bgp
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.3.0/24     1.1.1.1                  0     100      0 (65001) 1 i
R2#show ip bgp neighbors 172.16.2.22 advertised-routes

Total number of prefixes 0
```

R2 からみると ISP2 は EBGP ネイバーです。そのため、no_export の COMMUNITY が付加されている 100.1.3.0/24 のルートを送信しません。

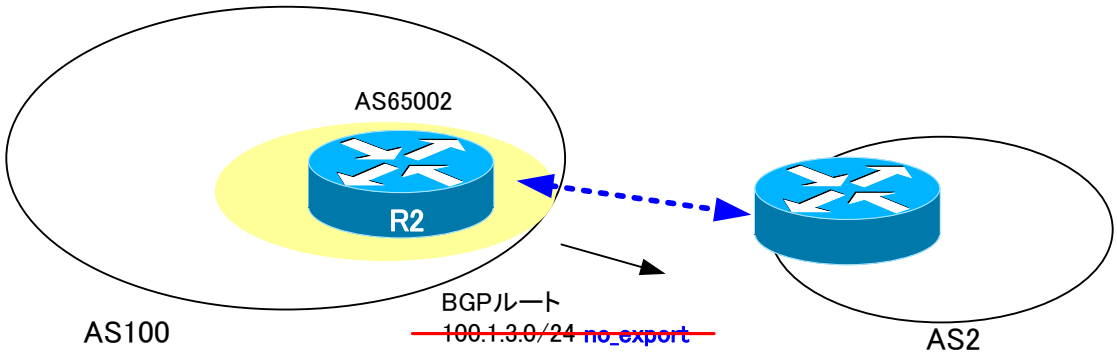

図 12 R2 から ISP2 へのルート送信

このように Well-known COMMUNITY を付加すると、その COMMUNITY に応じて自動的にフィルタがかかり、ルートが伝わる範囲を限定することができます。

プライベート COMMUNITY によるルート制御

Well-known COMMUNITY では自動的なルートのフィルタが可能ですが、その影響範囲が限定的です。COMMUNITY を付加する AS と実際にフィルタする AS の間に他の AS が含まれている場合、Well-known COMMUNITY ではフィルタできません。また、Well-known COMMUNITY ではフィルタ以外の動作もできません。

離れた AS 間でルートフィルタや他のアトリビュートのセットなどの制御を行うためにプライベート COMMUNITY を利用します。次のネットワーク構成で実際に設定してみます。

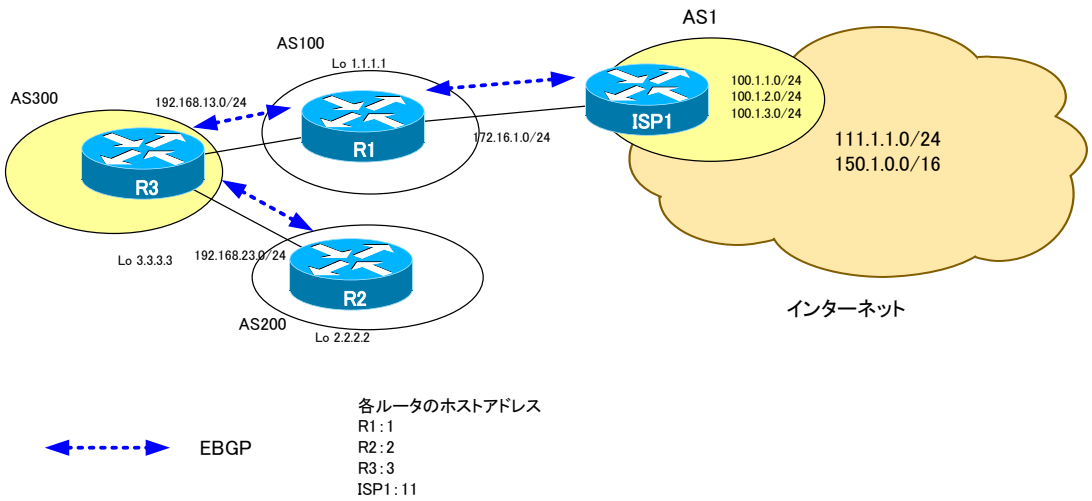

図 13 COMMUNITY によるルート制御の構成図

※ R1-R3、R3-R2 の EBGP ネイバーはそれぞれループバックインタフェースの IP アドレスで確立しています。

この図の AS1 と AS300 で次のようにルートを制御します。

- 111.1.1.0/24 と 150.1.0.0/16 は AS200 へ送信しない
- 100.1.1.0/24 には MED 10 を付加して AS200 へ送信する
- 100.1.2.0/24 には AS300 を追加でプリペンドして AS200 へ送信する
- 100.1.3.0/24 には追加の COMMUNITY 10:3 を付加して AS200 へ送信する

ルートを制御するために ISP1 で COMMUNITY を付加します。付加する COMMUNITY は自由に決められます。今回は下記の表ように各ルートに COMMUNITY を付加します。

表 2 ISP1 で付加する COMMUNITY

ルート	付加する COMMUNITY
111.1.1.0/24	1:1000
150.1.0.0/16	1:1000
100.1.1.0/24	1:1
100.1.2.0/24	1:2
100.1.3.0/24	1:3

COMMUNITY によってルートをグループ化するので、同じ制御を行いたいルートには同じ COMMUNITY を付加します。今回は、111.1.1.0/24 と 150.1.0.0/16 は同じ制御を行うため同じ値の COMMUNITY を付加します。それ以外の 3 つのルートはそれぞれ異なる制御なので、個別に COMMUNITY を付加します。
ISP1 での COMMUNITY 付加の設定は次の通りです。

ISP1 COMMUNITY 付加

```
ISP1(config)#ip bgp-community new-format
ISP1(config)#access-list 1 permit 111.11.1.0
ISP1(config)#access-list 1 permit 150.1.0.0
ISP1(config)#access-list 10 permit 100.1.1.0
ISP1(config)#access-list 20 permit 100.1.2.0
ISP1(config)#access-list 30 permit 100.1.3.0
ISP1(config)#route-map COMMUNITY2 permit 10
ISP1(config-route-map)#match ip address 1
ISP1(config-route-map)#set community 1:1000
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY2 permit 20
ISP1(config-route-map)#match ip address 10
ISP1(config-route-map)#set community 1:1
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY2 permit 30
ISP1(config-route-map)#match ip address 20
ISP1(config-route-map)#set community 1:2
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY2 permit 40
```

```
ISP1(config-route-map)#match ip address 30
ISP1(config-route-map)#set community 1:3
ISP1(config-route-map)#exit
ISP1(config)#route-map COMMUNITY permit 1000
ISP1(config-route-map)#exit
ISP1(config)#router bgp 1
ISP1(config-router)#neighbor 172.16.1.1 route-map COMMUNITY2 out
ISP1(config-router)#end
ISP1#clear ip bgp 172.16.1.1 out
```

※ COMMUNITY を「AS 番号:識別子」のフォーマットで扱うために ip bgp-community new-format を設定しています。他のルータも同様です。

今回の設定では、各ルータのネイバーの設定に neighbor send-community をすでに入れています。実際に COMMUNITY による制御を行う R3 において、ISP1 で付加した COMMUNITY を確認します。

R3 COMMUNITY の確認

```
R3#sh ip bgp community 1:1000
～省略～

   Network          Next Hop            Metric LocPrf Weight Path
*> 111.1.1.0/24     1.1.1.1                                0 100 1 i
*> 150.1.0.0        1.1.1.1                                0 100 1 i
R3#sh ip bgp community 1:1
～省略～

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.1.0/24     1.1.1.1                                0 100 1 i
R3#sh ip bgp community 1:2
～省略～

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 100.1.2.0/24      1.1.1.1                              0 100 1 i
R3#sh ip bgp community 1:3
～省略～

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.3.0/24      1.1.1.1                              0 100 1 i
```

R3 で、付加されている COMMUNITY を参照してルート制御を行います。そのための設定は次のようになります。

R3 COMMUNITY によるルート制御の設定

```
R3(config)#ip community-list 1 permit 1:1000
R3(config)#ip community-list 2 permit 1:1
R3(config)#ip community-list 3 permit 1:2
R3(config)#ip community-list 4 permit 1:3
R3(config)#route-map COMMUNITY_ACTION deny 10
R3(config-route-map)#match community 1
R3(config-route-map)#exit
R3(config)#route-map COMMUNITY_ACTION permit 20
R3(config-route-map)#match community 2
R3(config-route-map)#set metric 10
R3(config-route-map)#exit
R3(config)#route-map COMMUNITY_ACTION permit 30
R3(config-route-map)#match community 3
R3(config-route-map)#set as-path prepend 300
R3(config-route-map)#exit
R3(config)#route-map COMMUNITY_ACTION permit 40
R3(config-route-map)#match community 4
R3(config-route-map)#set community 10:3 additive
R3(config-route-map)#exit
R3(config)#route-map COMMUNITY_ACTION permit 1000
R3(config-route-map)#exit
R3(config)#router bgp 300
R3(config-router)#neighbor 2.2.2.2 route-map COMMUNITY_ACTION out
```

```
R3(config-router)#end
R3#clear ip bgp * out
```

この設定について、もう少し詳しく解説します。まず、付加されている COMMUNITY を参照するためのコミュニティリストを作成しています。

コミュニティリストの作成

```
R3(config)#ip community-list 1 permit 1:1000
R3(config)#ip community-list 2 permit 1:1
R3(config)#ip community-list 3 permit 1:2
R3(config)#ip community-list 4 permit 1:3
```

コミュニティリストをルートマップの match 条件で指定することで、ルートに付加されている COMMUNITY の値を参照した制御が可能です。

ルートマップ「COMMUNITY_ACTION」で最初に決めた実際のルート制御を設定しています。まず、シーケンス 10 ではルートのフィルタです。1:1000(コミュニティリスト 1)が付加されているルートは deny でフィルタします。

```
R3(config)#route-map COMMUNITY_ACTION deny 10
R3(config-route-map)#match community 1
```

シーケンス 20 では 1:1(コミュニティリスト 2)が付加されているルートに MED 10 を付加して送信しています。

```
R3(config)#route-map COMMUNITY_ACTION permit 20
R3(config-route-map)#match community 2
R3(config-route-map)#set metric 10
```

シーケンス 30 は 1:2(コミュニティリスト 3)が付加されているルートに対して追加の AS_PATH プリペンドを行っています。

```
R3(config)#route-map COMMUNITY_ACTION permit 30
R3(config-route-map)#match community 3
R3(config-route-map)#set as-path prepend 300
```

シーケンス 40 は 1:3(コミュニティリスト 4)が付加されているルートに対して追加の COMMUNITY 10:3 を付加しています。additive のオプションがないと、COMMUNITY 値が書き換わってしまうので注意してください。

```
R3(config)#route-map COMMUNITY_ACTION permit 40
R3(config-route-map)#match community 4
R3(config-route-map)#set community 10:3 additive
```

また、シーケンス 1000 はそのほかのルートをすべて permit するためのものです。

```
R3(config)#route-map COMMUNITY_ACTION permit 1000
```

今回考えている構成では、扱っているルートが全部で 6 つしかなく、これまでのシーケンスで処理されているのでシーケンス 1000 は不要です。ですが、実際の環境では、その他のたくさんのルートがあることがほとんどです。それらを送信するためにシーケンス 1000 が必要です。

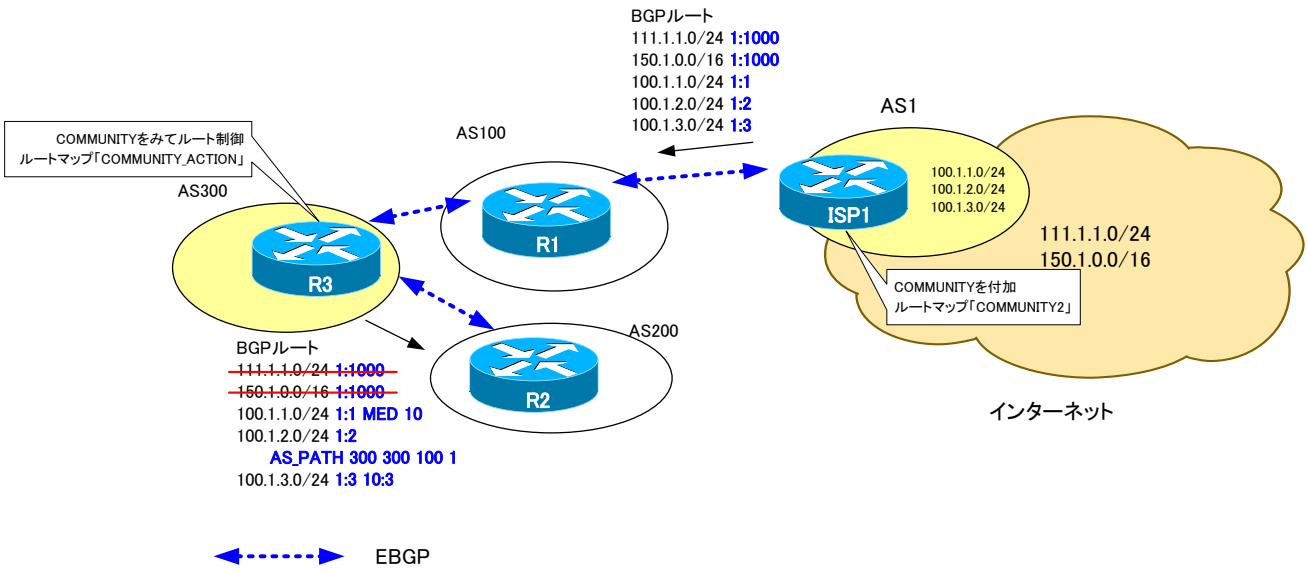

図 14 COMMUNITY 付加と制御の様子

R2 で実際の BGP テーブルを確認しましょう。

R2 BGP テーブル

```
R2#show ip bgp
BGP table version is 18, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.1.1.0/24     3.3.3.3                 10             0 300 100 1 i
*> 100.1.2.0/24     3.3.3.3                                0 300 300 100 1 i
*> 100.1.3.0/24     3.3.3.3                                0 300 100 1 i
R2#show ip bgp 100.1.1.0
BGP routing table entry for 100.1.1.0/24, version 16
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  300 100 1
    3.3.3.3 (metric 2) from 3.3.3.3 (100.100.1.3)
      Origin IGP, metric 10, localpref 100, valid, external, best
```

```
      Community: 1:1
R2#show ip bgp 100.1.2.0
BGP routing table entry for 100.1.2.0/24, version 18
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  300 300 100 1
    3.3.3.3 (metric 2) from 3.3.3.3 (100.100.1.3)
      Origin IGP, localpref 100, valid, external, best
      Community: 1:2
R2#show ip bgp 100.1.3.0
BGP routing table entry for 100.1.3.0/24, version 17
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  300 100 1
    3.3.3.3 (metric 2) from 3.3.3.3 (100.100.1.3)
      Origin IGP, localpref 100, valid, external, best
      Community: 1:3 10:3
```

R2 の BGP テーブルをみると、意図したとおりにルート制御ができていることがわかります。このように COMMUNITY を利用すれば、**離れた AS 間で何らかの条件でルートをグループ化してさまざまな制御を行う**ことが可能になります。