



BGP 用のセグメントルーティング

従来の BGP ベースのデータセンター (DC) ファブリックでは、パケットは自律システムの各ノードにホップバイホップで転送されます。トラフィックは、外部 BGP (eBGP) マルチパス ECMP に沿ってのみ送信されます。トラフィックエンジニアリングを行うことはできません。

MPLS ベースの DC ファブリックでは、ノード間の eBGP セッションは、BGP ラベル付きユニキャスト (BGP-LU) ネットワーク層到達可能性情報 (NLRI) を交換します。MPLS ベースの DC ファブリックを使用すると、ファブリック内の任意のリーフ (トップオブラックまたは境界ルータ) が単一のラベルを使用して他のリーフと通信できるため、従来の BGP ベースの DC ファブリックよりもパケット転送パフォーマンスが高くなり、カプセル化のオーバーヘッドが少なくなります。ただし、各ラベル値はホップごとに異なる可能性があるため、MPLS ベースの DC ファブリックはトラブルシューティングが難しく、構成が複雑です。

BGP は、セグメントルーティングプレフィックス SID インデックスを伝送するように拡張されました。BGP-LU は、各ノードが他のリーフノードの BGP プレフィックス SID を学習するのに役立ち、送信元と宛先の間で ECMP を使用できます。BGP のセグメントルーティングによって、ファブリックの構成、操作、およびトラブルシューティングが簡素化されます。BGP のセグメントルーティングでは、BGP プレフィックス SID を使用してデータセンターでトラフィックステアリング機能を有効にできます。

- [BGP プレフィックスセグメント識別子の設定 \(1 ページ\)](#)
- [セグメントルーティング出力ピアエンジニアリングの設定 \(3 ページ\)](#)
- [BGP リンクステートの設定 \(4 ページ\)](#)
- [例: SR-EPE および BGP-LS の設定 \(5 ページ\)](#)

BGP プレフィックスセグメント識別子の設定

BGP プレフィックスに関連付けられたセグメントは、BGP プレフィックス SID と呼ばれます。BGP プレフィックス SID は、セグメントルーティングまたは BGP ドメイン内でグローバルです。これは、BGP によって計算された ECMP 対応のベストパス上のパケットに関連するプレフィックスに転送する命令を識別します。BGP プレフィックス SID は、ラベルのセグメントルーティンググローバルブロック (SRGB) の範囲から手動で設定されます。

各 BGP スピーカーは、**segment-routing global-block** コマンドを使用して SRGB で設定する必要があります。SRGB の詳細については、「[セグメントルーティング グローバルブロックについて](#)」の項を参照してください。



(注) 範囲から割り当てられた値はドメイン全体で重要な意味を持つため、ドメイン内のすべてのルータに同じ値の範囲を設定することをお勧めします。

BGP プレフィックス SID を割り当てるには、最初に **set label-index index** 属性を使用してルーティング ポリシーを作成し、次にそのインデックスをノードに関連付けます。



(注) **set label-index** 属性を持つルーティング ポリシーは、ネットワーク構成または再配布構成に追加できます。他のルーティング ポリシー言語 (RPL) の構成も可能です。ルーティング ポリシーの詳細については、『[Implementing Routing Policy](#)』の章を参照してください。

例

次の例に、SRGB を設定し、\$SID パラメータと **set label-index** 属性を使用して BGP ルート ポリシーを作成し、プレフィックス SID インデックスをノードに関連付ける方法を示します。

```
RP/0/RP0/CPU0:router(config)# segment-routing global-block 16000 23999

RP/0/RP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RP0/CPU0:router(config-rpl)# end policy

RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RP0/CPU0:router(config-bgp-af)# commit
RP/0/RP0/CPU0:router(config-bgp-af)# end

RP/0/RP0/CPU0:router# show bgp 1.1.1.3/32
BGP routing table entry for 1.1.1.3/32
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          74         74
  Local Label: 16003
Last Modified: Sep 29 19:52:18.155 for 00:07:22
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.2
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.2
  3
  99.3.21.3 from 99.3.21.3 (1.1.1.3)
  Received Label 3
  Origin IGP, metric 0, localpref 100, valid, external, best, group-best
```

```
Received Path ID 0, Local Path ID 1, version 74
Origin-AS validity: not-found
Label Index: 3
```

セグメントルーティング出力ピアエンジニアリングの設定

セグメントルーティング出力ピアエンジニアリング (EPE) はコントローラを使用して、セグメントルーティングドメイン内の入力プロバイダーエッジまたはコンテンツソース (ノード) に、特定の出口プロバイダーエッジ (ノード) および特定の外部インターフェイスを使用して宛先に到達するよう指示します。BGP ピア SID は、ソースルーティングされたドメイン間パスを表すために使用されます。

コントローラは、BGP-LS EPE ルートを介して、BGP ピア SID と出力境界ルータの外部トポロジを学習します。コントローラは、BGP ラベル付きユニキャスト (BGP-LU) を使用して出口ノードとピアノードを経由して宛先にトラフィックを誘導するように入力ノードをプログラミングできます。

EPE 機能は、EPE 出力境界ルータおよび EPE コントローラでのみ必要です。

このタスクでは、EPE 出口ノードでセグメントルーティング EPE を設定する方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	router bgp as-number 例 : <pre>RP/0/RP0/CPU0:router(config)# router bgp 1</pre>	BGP AS 番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティングプロセスを設定できます。
ステップ 2	neighbor ip-address 例 : <pre>RP/0/RP0/CPU0:router(config-bgp)# neighbor 192.168.1.3</pre>	BGP ルーティングのためにルータをネイバー コンフィギュレーションモードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 3	remote-as as-number 例 : <pre>RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 3</pre>	ネイバーを作成し、リモート自律システム番号を割り当てます。

	コマンドまたはアクション	目的
ステップ 4	egress-engineering 例 : RP/0/RP0/CPU0:router (config-bgp-nbr) # egress-engineering	eBGP ピア用に EPE を使用して出力ノードを設定します。

BGP リンク ステートの設定

BGP リンクステート (LS) は、BGP を介して内部ゲートウェイ プロトコル (IGP) リンクステートデータベースを伝えるために定義されたアドレスファミリー識別子 (AFI) およびサブアドレスファミリー識別子 (SAFI) です。BGPLS は、ネットワーク トポロジ情報を トポロジサーバおよびアプリケーション層トラフィック最適化 (ALTO) サーバに提供します。BGP LS では、集約、情報の非表示、および抽象化に対するポリシー ベースの制御が可能です。BGP LS は、IS-IS および OSPFv2 をサポートしています。



(注) IGP は、リモートピアからの BGP LS データを使用しません。BGP は、ルータの他のコンポーネントに受信した BGP LS データをダウンロードしません。

セグメント ルーティングの場合、次の属性が BGP LS に追加されています。

- ノード : セグメント ルーティング機能 (SRGB 範囲を含む) およびアルゴリズム
- リンク : 隣接関係 SID と LAN 隣接関係 SID
- プレフィックス : プレフィックス SID およびセグメント ルーティング マッピング サーバ (SRMS) のプレフィックス範囲

次の例は、リンクステート情報を BGP ネイバーと交換する方法を示しています。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.0.0.2
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family link-state link-state
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# exit
```

IGP の拡張機能

所定の BGP ノードは、複数の独立したルーティング ドメインに接続できます。BGP への IGP リンクステート配布が OSPF プロトコルと ISIS プロトコルの両方に追加され、そのノードは、これらの複数のドメインにまたがるまたはドメインを含むパスを構築するアプリケーションに同様の方法でこの情報を渡すことができます。

BGP を使用して ISIS リンクステートデータを配布するには、ルータ コンフィギュレーション モードで **distribute bgp-ls** コマンドを使用します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router isis isp
RP/0/RP0/CPU0:router(config-isis)# distribute bgp-ls instance-id 32 level 2 throttle 5
```

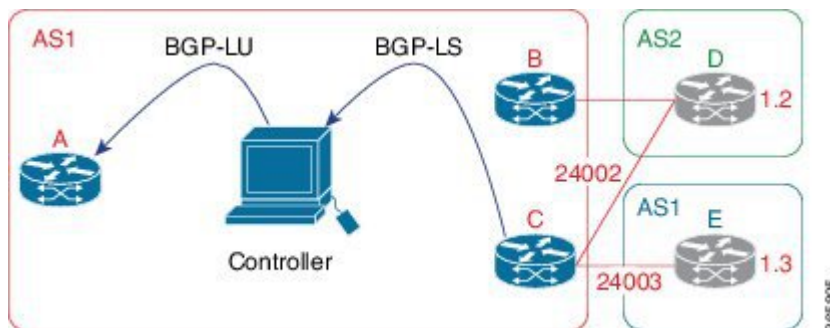
BGP を使用して OSPFv2 および OSPFv3 リンクステートデータを配布するには、ルータ コンフィギュレーション モードで **distribute bgp-ls** コマンドを使用します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router ospf 100
RP/0/RP0/CPU0:router(config-ospf)# distribute bgp-ls instance-id 32 throttle 10
```

例：SR-EPE および BGP-LS の設定

次の図では、入口ノード A および出口ノード B および C を備えた自律システム AS1 でセグメントルーティングが有効になっています。この例で、出口ノード C に EPE を設定します。

図 1: トポロジ



手順

ステップ 1 eBGP ピア D および E 用に EPE を使用してノード C を設定します。

例：

```
RP/0/RP0/CPU0:router_C(config)# router bgp 1
RP/0/RP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.3
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 3
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to E
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_in in
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_out out
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# exit
RP/0/RP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.2
```

```
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 2
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to D
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_in in
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# route-policy bgp_out out
RP/0/RP0/CPU0:router_C(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# exit
```

ステップ2 BGP-LSを使用してピアノードSIDをコントローラにアドバタイズするようにノードCを設定します。

例：

```
RP/0/RP0/CPU0:router_C(config-bgp)# neighbor 172.29.50.71
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# description to EPE_controller
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# address-family link-state link-state
RP/0/RP0/CPU0:router_C(config-bgp-nbr)# exit
RP/0/RP0/CPU0:router_C(config-bgp)# exit
```

ステップ3 設定をコミットします。

例：

```
RP/0/RP0/CPU0:router_C(config)# commit
```

ステップ4 設定を確認します。

例：

```
RP/0/RP0/CPU0:router_C# show bgp egress-engineering

Egress Engineering Peer Set: 192.168.1.2/32 (10b87210)
  Nexthop: 192.168.1.2
  Version: 2, rn_version: 2
  Flags: 0x00000002
  Local ASN: 1
  Remote ASN: 2
  Local RID: 1.1.1.3
  Remote RID: 1.1.1.4
  First Hop: 192.168.1.2
  NHID: 3
  Label: 24002, Refcount: 3
  rpc_set: 10b9d408

Egress Engineering Peer Set: 192.168.1.3/32 (10be61d4)
  Nexthop: 192.168.1.3
  Version: 3, rn_version: 3
  Flags: 0x00000002
  Local ASN: 1
  Remote ASN: 3
  Local RID: 1.1.1.3
  Remote RID: 1.1.1.5
  First Hop: 192.168.1.3
  NHID: 4
  Label: 24003, Refcount: 3
  rpc_set: 10be6250
```

出力は、ノードCが各eBGPピアに対してピアSIDを割り当てたことを示しています。

例：

```
RP/0/RP0/CPU0:router_C# show mpls forwarding labels 24002 24003
Local   Outgoing   Prefix      Outgoing    Next Hop    Bytes
Label   Label      or ID       Interface   Interface   Switched
-----
24002   Unlabelled No ID       Te0/0/0/1   192.168.1.2 0
24003   Unlabelled No ID       Te0/0/0/2   192.168.1.3 0
```

出力は、ノード C が転送情報ベース（FIB）にピア ノード SID をインストールしたことを示しています。

