



Cisco IOS XRv 9000 ルータの特徴的な機能

この章では、Cisco IOS XRv 9000 ルータの特徴的な機能について説明します。

- [BGP の最適なルート リフレクタ \(1 ページ\)](#)
- [マルチホップパスの BFD \(8 ページ\)](#)
- [CVAC : ブートストラップ構成のサポート \(10 ページ\)](#)
- [データプレーンの管理 \(13 ページ\)](#)
- [Early Fast Discard \(18 ページ\)](#)
- [bgp bestpath igp-metric ignore コマンドのサポート \(19 ページ\)](#)

BGP の最適なルート リフレクタ

表 1: 機能の履歴 (表)

機能名	リリース情報	機能説明
BGP ORR 6PE	リリース 7.3.1	<p>この機能が導入されます。特定の ORR テーブルのベストパスとして選択可能なパスがない場合は、デフォルトテーブルのベストパスを ORR グループのベストパスとして割り当てることができます。この機能は、IPv4 ネクストホップおよびデフォルトのフォールバックパスを使用した IPv6 ラベルユニキャストを有効にします。</p> <p>新しいキーワードがこのリリースで追加されました。</p> <ul style="list-style-type: none">• fallback-default-bestpath

BGP-ORR (最適なルート リフレクタ) により、仮想ルート リフレクタ (vRR) は、ルート リフレクタの (RR) クライアントの観点からベストパスを計算できます。

BGP ORR は次の方法でベストパスを計算します。

1. RRクライアントまたはRRクラスタ（RRクライアントのセット）のコンテキストで、SPFを複数回実行します。
2. それぞれの SPF 実行結果を、別個のデータベースに保存します。
3. これらのデータベースを使用して BGP のベストパス判断を処理し、これにより BGP がクライアントの観点から最適なベストパスを使用し、通知できるようにします。

自律システムでは、BGP のルートリフレクタは焦点として機能し、RR が計算したベストパスとルートをそのピア（RRクライアント）にアドバタイズします。RRによってアドバタイズされたベストパスはRRの観点から計算されることになるので、RRの配置は導入に関する重要な検討事項になります。

ネットワーク機能の仮想化（NFV）が主要な技術となっていることから、サービスプロバイダー（SP）は複数のサーバを使用するクラウドで仮想RR機能をホストしています。vRRはコントロールプレーンデバイス上で実行でき、トポロジまたはSPデータセンター内のどこにでも配置できます。Cisco IOS XRv 9000 ルータは、SPデータセンター内のNFVプラットフォーム上のvRRとして実装できます。vRRを利用することにより、SPはRR導入のメモリとCPU使用率を大幅に拡張できます。RRを最適な配置から移動するには、RRクライアントの観点から最適なパスを計算するORR機能をvRRが実装する必要があります。

BGP ORRには次のような利点があります。

- RRクライアントの観点からベストパスを計算します。
- vRRをトポロジまたはSPデータセンター内のどこにでも配置できます。
- SPはRR導入のメモリとCPU使用率を拡張できます。



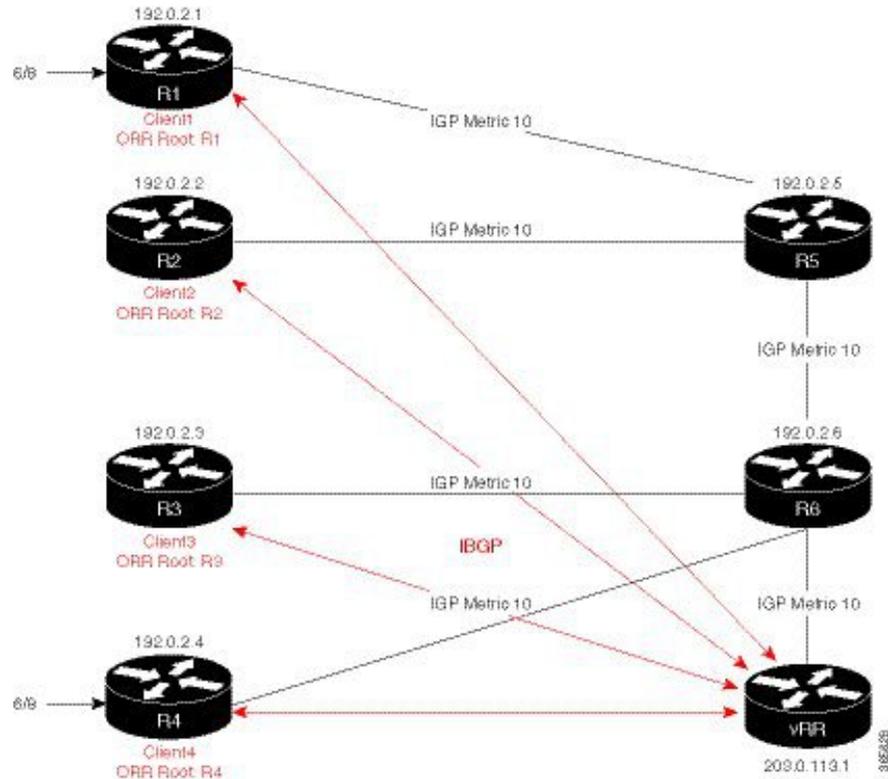
- (注) ORR機能を有効にすると、BGPとRIBのメモリフットプリントが増加します。ネットワーク内に設定されているvRRの数が増えると、ORRはBGPのコンバージェンスに悪影響を及ぼします。

使用例

次のような、BGPルートリフレクタのトポロジを検討します。

- ルータ R1、R2、R3、R4、R5、R6 がルートリフレクタクライアントである。
- ルータ R1 および R4 が vRR に 6/8 プレフィックスをアドバタイズする。

図 1: BGP-ORR トポロジ



vRR は、R1 および R4 からプレフィックス 6/8 を受信します。ネットワークに BGP ORR が設定されていない場合、vRR は RR クライアント R2、R3、R5、R6 の最も近い出力点として R4 を選択し、R4 から学習した 6/8 プレフィックスをこれらの RR クライアント (R2、R3、R5、R6) にリフレクトします。トポロジから、R2 のベストパスが R4 ではなく R1 であるのは明らかです。これは、vRR が RR の観点からベストパスを計算するためです。

BGP ORR がネットワークに設定されると、vRR は R2 の観点からネットワークの最短出力点を計算し、R2 に最も近い出力点は R1 であると判別します。その結果、vRR は R1 から学習した 6/8 プレフィックスを R2 にリフレクトします。

BGP ORR を設定します。

次の手順に従って、IPv4 ユニキャストおよび 6PE シナリオの BGP ORR を有効にします。

1. ルータ BGP モードで ORR をグローバルに定義します。
2. アドレスファミリモードで ORR グループを有効にします。
3. ネイバーを ORR クライアントとして指定します。

```
router bgp 100
  optimal-route-reflection ipv4 foo 10.1.1.1 10.1.1.2 10.1.1.3
  optimal-route-reflection ipv6 bar abcd::1 abcd::2 abcd::3
  address-family ipv4 unicast
    optimal-route-reflection apply foo
  address-family ipv6 unicast
```

```

optimal-route-reflection apply foo
allocate-label {all | route-policy <>}
neighbor 2.2.2.2
remote-as 100
address-family ipv4 unicast
  optimal-route-reflection foo
address-family ipv6 label-unicast
  optimal-route-reflection foo

```

ORR グループのベストパスがない場合は、デフォルトテーブルのベストパスの選択を有効にします。

```

router bgp 65000
  bgp router-id 10.1.1.1
  address-family ipv4 unicast
    optimal-route-reflection fallback-default-bestpath

```

確認

R2 が最適な出力を受信したかどうかを確認するには、EXEC モードで **show bgp <prefix>** コマンドを (R2 から) 実行します。上記の例では、R1 および R4 は 6/8 プレフィックスをアドバタイズします。次のように **show bgp 6.0.0.0/8** コマンドを実行します。

```

R2# show bgp 6.0.0.0/8
Tue Apr  5 20:21:58.509 UTC
BGP routing table entry for 6.0.0.0/8
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          8         8
Last Modified: Apr  5 20:00:44.022 for 00:21:14
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    192.0.2.1 (metric 20) from 203.0.113.1 (192.0.2.1)
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best
    Received Path ID 0, Local Path ID 1, version 8
    Originator: 192.0.2.1, Cluster list: 203.0.113.1

```

上記の例では、出力の状態として、R2 のベストパスが、IP アドレスが 192.0.2.1 でパスのメトリックが 20 である R1 経由であることが表示されています。

ORR によって計算された R2 のベストパスを判別するには、**show bgp** コマンドを vRR から実行します。R2 には他のピアとは異なるベストパスがある (または異なるポリシーが設定されている) ため、R2 は独自のアップデートグループを持ちます。

```

VRR#show bgp 6.0.0.0/8
Thu Apr 28 13:36:42.744 UTC
BGP routing table entry for 6.0.0.0/8
Versions:
  Process bRIB/RIB SendTblVer
  Speaker 13 13
Last Modified: Apr 28 13:36:26.909 for 00:00:15
Paths: (2 available, best #2)
Advertised to update-groups (with more than one peer):
  0.2
Path #1: Received by speaker 0
ORR bestpath for update-groups (with more than one peer):
0.1

```

```

Local, (Received from a RR-client)
192.0.2.1 (metric 30) from 192.0.2.1 (192.0.2.1)
Origin incomplete, metric 0, localpref 100, valid, internal, add-path
Received Path ID 0, Local Path ID 2, version 13
Path #2: Received by speaker 0
Advertised to update-groups (with more than one peer):
0.2
ORR addpath for update-groups (with more than one peer):
0.1
Local, (Received from a RR-client)
192.0.2.4 (metric 20) from 192.0.2.4 (192.0.2.4)
Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best
Received Path ID 0, Local Path ID 1, version 13

```



- (注) パス #1 は、アップデートグループ 0.1 にアドバタイズされます。R2 はアップデートグループ 0.1 に含まれます。

show bgp コマンドをアップデートグループ 0.1 に対して実行し、R2 がアップデートグループ 0.1 に含まれているかどうか確認します。

```

VRR# show bgp update-group 0.1
Thu Apr 28 13:38:18.517 UTC

Update group for IPv4 Unicast, index 0.1:
Attributes:
Neighbor sessions are IPv4
Internal
Common admin
First neighbor AS: 65000
Send communities
Send GSHUT community if originated
Send extended communities
Route Reflector Client
ORR root (configured): g1; Index: 0
4-byte AS capable
Non-labeled address-family capable
Send AIGP
Send multicast attributes
Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 5, replicated: 5
All neighbors are assigned to sub-group(s)
Neighbors in sub-group: 0.2, Filter-Groups num:1
Neighbors in filter-group: 0.2 (RT num: 0)
192.0.2.2

```

次の確認点として、g1 ポリシーを設定した結果 vRR に作成されたテーブルのコンテンツを確認します。R2 の観点からは、R1 に到達するためのコストは 20 で、R4 に到達するためのコストは 30 です。したがって、R2 に最も近い最適な出力は R1 経由になります。

```

VRR#show orrspf database g1
Thu Apr 28 13:39:20.333 UTC

ORR policy: g1, IPv4, RIB tableid: 0xe0000011
Configured root: primary: 192.0.2.2, secondary: NULL, tertiary: NULL

```

```
Actual Root: 192.0.2.2, Root node: 2000.0100.1002.0000
```

```
Prefix Cost
203.0.113.1 30
192.0.2.1 20
192.0.2.2 0
192.0.2.3 30
192.0.2.4 30
192.0.2.5 10
192.0.2.6 20
```

```
Number of mapping entries: 8
```

ORR GPE の検証

```
show bgp ipv6 labeled-unicast 1111::1/128
Tue Mar 2 10:25:00.748 PST
BGP routing table entry for 1111::1/128
Versions:
Process bRIB/RIB SendTblVer
Speaker 4 4
Last Modified: Mar 2 10:18:53.000 for 00:06:08
Paths: (3 available, best #3)
Advertised IPv6 Labeled-unicast paths to update-groups (with more than one peer):
0.2
Path #1: Received by speaker 0
ORR bestpath for update-groups (with more than one peer):
0.1
Local, (Received from a RR-client)
192.168.0.3 (metric 75) from 192.168.0.3 (192.168.0.3)
Received Label 24007
Origin incomplete, metric 0, localpref 100, valid, internal, add-path, labeled-unicast
Received Path ID 0, Local Path ID 2, version 4
Path #2: Received by speaker 0
Not advertised to any peer
Local, (Received from a RR-client)
192.168.0.4 (metric 190) from 192.168.0.4 (192.168.0.4)
Received Label 24007
Origin incomplete, metric 0, localpref 100, valid, internal, labeled-unicast
Received Path ID 0, Local Path ID 0, version 0
Path #3: Received by speaker 0
Advertised IPv6 Labeled-unicast paths to update-groups (with more than one peer):
0.2
Local, (Received from a RR-client)
192.168.0.5 (metric 65) from 192.168.0.5 (192.168.0.5)
Received Label 24007
Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best,
labeled-unicast
Received Path ID 0, Local Path ID 1, version 3
```

show bgp ipv6 labeled-unicast update-group

```
Tue Mar 2 10:25:51.308 PST

Update group for IPv6 Labeled-unicast, index 0.1:
Attributes:
Neighbor sessions are IPv4
Internal
Common admin
First neighbor AS: 1
Send communities
Send GSHUT community if originated
Send extended communities
Route Reflector Client
```

```

ORR root (configured): orr-grp-1; Index: 0
4-byte AS capable
Send AIGP
Send multicast attributes
Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 1, replicated: 2
All neighbors are assigned to sub-group(s)
Neighbors in sub-group: 0.2, Filter-Groups num:1
Neighbors in filter-group: 0.2(RT num: 0)
192.168.0.2 192.168.0.4

Update group for IPv6 Labeled-unicast, index 0.2:
Attributes:
Neighbor sessions are IPv4
Internal
Common admin
First neighbor AS: 1
Send communities
Send GSHUT community if originated
Send extended communities
Route Reflector Client
4-byte AS capable
Send AIGP
Send multicast attributes
Minimum advertisement interval: 0 secs
Update group desynchronized: 0
Sub-groups merged: 0
Number of refresh subgroups: 0
Messages formatted: 1, replicated: 4
All neighbors are assigned to sub-group(s)
Neighbors in sub-group: 0.1, Filter-Groups num:1
Neighbors in filter-group: 0.1(RT num: 0)
192.168.0.3 192.168.0.5

```

```

show bgp ipv6 unicast orr-group all
Tue Mar 2 10:26:41.072 PST
Name Tableid Nbrcnt Index Root
orr-grp-1 0xe0000019 2 0 192.168.0.3

```

次の show コマンドは、BGP スピーカーのグローバル ORR ポリシーグループテーブルを表示します。

```

Router# show bgp orr-group global all
Wed Apr 8 16:46:29.929 PDT
Name Policy-afi Global Tableid AFI-count Root
orr-grp-3 IPv4 Yes 0xe0000014 1 1.1.2.1
orr-grp-2 Ipv6 Yes 0xe0800013 0 1::1
orr-grp-1 IPv4 Yes 0xe0000012 2 192.168.0.3

```

次の show コマンドは、特定の ORR 名のグローバル ORR グループエントリの詳細を表示します。

```

Router# show bgp orr-group global orr-grp-1
Wed Apr 8 16:46:51.596 PDT
  ORR Name : orr-grp-1
    policy afi : IPv4
  global Defined : Yes
    tableid : 0xe0000012

```

```

      aficnt : 2
IPv4 unicast used : Yes
IPv6 unicast used : Yes
      root : 192.168.0.3

```

次の show コマンドは、BPM ORR ポリシーグループテーブルを表示します。

```

Router# show bgp orr-group bpm all
Wed Apr  8 16:49:44.223 PDT
Name          Policy-afi Global      AFI-cnt   Nbr-af-cnt   Root
orr-grp-3     IPv4         Yes         1          0             1.1.2.1
orr-grp-2     IPv6         Yes         0          0             1::1
orr-grp-1     IPv4         Yes         2          4             192.168.0.3

```

次の show コマンドは、特定の ORR 名の BPM ORR グループエントリの詳細を表示します。

```

Router# show bgp orr-group bpm orr-grp-1
Wed Apr  8 16:50:02.437 PDT
  ORR Name : orr-grp-1
    v4 policy : Yes
  global Defined : Yes
    AFI count : 2
  total nbr af cnt : 4
IPv4 unicast used : Yes
IPv6 unicast used : Yes
  IPv4 nbr af cnt : 2
  IPv6 nbr af cnt : 2
    root : 192.168.0.3

```

マルチホップパスの BFD

表 2: 機能の履歴 (表)

機能名	リリース情報	機能説明
iBGP および eBGP の BFDv4 および BFDv6 マルチホップ	リリース 7.3.1	この機能では、複数ホップ（最大 255 ホップ）離れた宛先に対する 1 秒未満の転送障害検出が可能になります。この機能は、BFD シングルホップをサポートするすべてのメディアタイプでサポートされます。

BFD マルチホップ (BFD-MH) は、同じサブネット上にない 2 つのアドレス間の BFD セッションです。BFD-MH の例には、PE および CE ループバック アドレス間の BFD セッションや、数 TTL ホップ離れたルータ間の BFD セッションがあります。外部および内部 BGP アプリケーションは BFD マルチホップをサポートします。BFD マルチホップは、複数のネットワークホップにまたがる場合もある任意のパス上の BFD をサポートします。

マルチホップパスの BFD 機能は、BFD シングルホップをサポートするすべてのメディアタイプでサポートされます。

BFD マルチホップセッションの設定

BFD マルチホップセッションは、クライアントによって指定された送信元アドレスと宛先アドレスの一意的ペア間で設定されます。IP 接続された2つのエンドポイント間でセッションを設定できます。グローバルルーティングテーブルと VRF テーブルの両方にある IPv4 アドレスがサポートされます。

BFD IPv6 マルチホップセッションの設定

BFD を BGP とともに使用すると、BFD セッションタイプ（シングルホップまたはマルチホップ）が BGP 設定に基づいて設定されます。eBGP-multihop キーワードを設定すると、BFD セッションもマルチホップモードで実行されます。それ以外の場合、セッションはシングルホップモードで実行されます。

特定のホップ数を超えるネイバーから送信された BFD パケットをドロップするには、**bfd multihop ttl-drop-threshold** コマンドを使用します。

- eBGP ネイバーの BFD IPv6 マルチホップの設定
- iBGP ネイバーの BFD IPv6 マルチホップの設定
- BGP ネイバーで BFD を有効化

eBGP ネイバーの BFD IPv6 マルチホップの設定

```
Router# configure
Router(config)# bfd multipath include location 0/7/CPU0
Router(config)# router bgp 65001
Router(config-bgp)# neighbor 21:1:1:1:1:1:2 ebgp-multihop 255
Router(config-bgp)# neighbor 21:1:1:1:1:1:2 bfd fast-detect
```

iBGP ネイバーの BFD IPv6 マルチホップの設定

```
Router# configure
Router(config)# bfd multipath include location 0/7/CPU0
Router(config)# router bgp 65001
Router(config-bgp)# neighbor 21:1:1:1:1:1:2
```

BGP ネイバーで BFD を有効化

```
Router# configure
Router(config)# router bgp 120
Router(config-bgp)# bfd minimum-interval 6500
Router(config-bgp)# bfd multiplier 7
Router(config-bgp)# neighbor 172.168.40.24
Router(config-bgp-nbr)# remote-as 2002
Router(config-bgp-nbr)# bfd fast-detect
```

実行コンフィギュレーション

次に、eBGP ネイバーの BFD IPv6 マルチホップの実行コンフィギュレーションを示します。

```
Router# show running-configuration
bfd multipath include location 0/7/CPU0
router bgp 65001
neighbor 21:1:1:1:1:1:2 ebgp-multihop 255
```

CVAC : ブートストラップ構成のサポート

```
neighbor 21:1:1:1:1:1:2 bfd fast-detect
```

次に、iBGP ネイバーの BFD IPv6 マルチホップの実行コンフィギュレーションを示します。

```
Router# show running-configuration

bfd multipath include location 0/7/CPU0
router bgp 65001
neighbor 21:1:1:1:1:1:2
```

次に、iBGP ネイバーの BFD IPv6 マルチホップの実行コンフィギュレーションを示します。

```
Router# show running-configuration

router bgp 120
  bfd minimum-interval 6500
  bfd multiplier 7
  neighbor 172.168.40.24
  remote-as 2002
  bfd fast-detect
```

確認

```
Router# show bfd session
Tue Apr 7 06:16:36.982 UTC
```

Src Addr	Dest Addr	VRF Name	H/W	NPU	Local	det	time(int*mult)	State
10.1.1.1	192.0.2.1	default	No	n/a	n/a	150ms	(50ms*3)	UP
10.1.1.2	192.0.2.2	default	No	n/a	n/a	150ms	(50ms*3)	UP
10.1.1.3	192.0.2.3	default	No	n/a	n/a	150ms	(50ms*3)	UP
10.1.1.4	192.0.2.4	default	No	n/a	n/a	150ms	(50ms*3)	UP

```
Router# show bfd ipv6 session
```

```
Tue Apr 7 06:16:45.012 UTC
```

Src Addr	Dest Addr	VRF Name	Local	det	time(int*mult)	State	Echo	Async
2001:DB8::1	2001:DB8:0:ABCD::1	default	0s (0s*0)	150ms	(50ms*3)	UP		
2001:DB8::2	2001:DB8:0:ABCD::2	default	0s (0s*0)	150ms	(50ms*3)	UP		
2001:DB8::3	2001:DB8:0:ABCD::3	default	0s (0s*0)	150ms	(50ms*3)	UP		
2001:DB8::4	2001:DB8:0:ABCD::4	default	0s (0s*0)	150ms	(50ms*3)	UP		

CVAC : ブートストラップ構成のサポート

Cisco Virtual Appliance Configuration (CVAC) は、複数の Cisco 仮想ルータでサポートされるアウトオブバンド構成機能です。CVACは、ハイパーバイザによって提供されるCD-ROM、ディスクイメージ、またはUSBドライブで、仮想ルータ環境に入っている構成を受け取ります。この構成は、起動時に検出されて適用されます。

これにより、ユーザは初期導入の際に新しい仮想ルータをスタートアップ（ブートストラップ）と組み合わせることができるようになり、通常は手動で実施する必要のある基本的な多数の要件（管理IPアドレスなど）の構成が非常に簡単に短時間で行えるようになります。



- (注) CVAC は、既存の構成（新しいシステムで入力を求められる初期ユーザ名およびパスワードを含む）がない場合に機能します。

Cisco IOS XRv 9000 ルータは、ネイティブの KVM、Openstack Config Drive、および Virsh で CVAC を完全にサポートします。VMware ESXi では、CVAC がサポートされません。

ブートストラップ構成 ISO の作成

Cisco IOS XRv 9000 ルータは、単一 CD-ROM ドライブ上のプレーン テキスト設定ファイルをサポートします。

- **iosxr_config.txt** : 標準的な XR 設定を提供します。

このテキスト ファイルは、CVAC が自動的に適用されるのに必要な設定 CLI の簡単な一覧を提供します。この操作は、手動で **copy iosxr_config.txt running-config** コマンドを実行するのと機能的には同じです。

1 つ以上の設定ファイルがある場合は、次のコマンドを使用して、Cisco IOS XRv 9000 ルータに挿入するのに適した ISO イメージを作成できます。

```
mkisofs -output bootstrap.iso -l -V config-1 --relaxed-filenames --iso-level 2
iosxr_config.txt
```

次に、Ubuntu で **mkisofs** コマンドを実行した場合のサンプル出力を示します。

```
Warning: creating filesystem that does not conform to ISO-9660.
I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
```

CVAC と KVM

CVAC が **config** ファイルを処理するようにするには、**Qemu** コマンドラインに付加的なドライブを（最後のドライブとして）追加します。

```
-drive file=./bootstrap.iso,if=virtio,media=cdrom,index=3
```

設定ファイルが適切に提供され、CVAC が正常に実行された場合、以下の **syslog** メッセージが表示されます。

```
RP/0/0/CPU0:Dec 14 09:10:22.719 : config[1]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'CVAC'
Use 'show configuration commit changes 1000000001' to view the changes.
```

```
RP/0/0/CPU0:Dec 14 09:10:23.619 : cvac[2]: %MGBL-CVAC-5-CONFIG_DONE :
```

ファイル /disk0:/iosxr_config.txt から設定が適用されました。

config ファイルからのいずれかの設定が拒否された場合、以下の syslog メッセージが追加で表示されます。

```
RP/0/0/CPU0:Dec 14 09:10:23.619 : cvac[2]: %MGBL-CVAC-3-CONFIG_ERROR : Errors were
encountered while applying configs from file /etc/sysconfig/iosxr_config.txt. Please
inspect 'show configuration failed' for details
```

失敗しなかった設定はコミットされます。disk0:/cvac.log ファイルにさらに詳しいデバッグ情報が含まれています。

CVAC と Virsh

1. Cisco IOS XRv 9000 ルータに挿入するのに適した ISO イメージを作成し、「ブートストラップ構成 ISO の作成」のセクションで説明されている手順に従います。
2. ルータのイメージと共にダウンロードされる Virsh.xml ファイルには、以下に示す Bootstrap セクションがあります。

```
<!-- BootstrapSection -->
<!-- Example Bootstrap CLI ISO -->
<!-- <disk type='file' device='cdrom' -->
<!-- <driver name='qemu' type='raw' -->
<!-- <source file='<ISO with file iosxr_config.txt' -->
<!-- <target dev='vdc' bus='virtio' -->
<!-- <readonly -->
<!-- <alias name='bootstrap_CLI' -->
<!-- </disk -->
```

3. Bootstrap セクションをアンコメントし、source file の参照先として、インスタンスを起動するマシン上のブートストラップ ISO ファイルの絶対パスを指定します。次に例を示します。

```
<!--BootstrapSection -->
<disk type='file' device='cdrom'>
<driver name='qemu' type='raw'>
<source file='/production/bootstrap.iso'/>
<target dev='vdc' bus='virtio'>
<readonly/>
<alias name='bootstrap_CLI'>
</disk>
```

CVAC と OpenStack Config-Drive

Config-Drive は、OpenStack でオーケストレーションされる VM を初期設定でブートストラップするメカニズムです。OpenStack Config-Drive については、[ここ](#)をクリックしてください。

Config-Drive のサポートが望ましい場合は、以下のコマンドラインを使用して XR 初期設定のプレーンテキスト ファイルを渡すことができます。

```
config-drive true user-data /iosxr_config.txt file /iosxr_config.txt=/iosxr_config.txt
```

ファイル `iosxr_config.txt` は、XR コマンドが含まれている生のテキストファイルです。前のセクションで言及されていた、KVM コマンドラインまたは Virsh で使用される ISO ファイルではありません。

起動時の CVAC

構成無しで新規インストールした VM インスタンスの場合、CVAC の動作は簡単なものになります。それは、パーサーを通じて受け入れられた構成がすべてコミットされるというものです。CVAC は、最後に適用した設定ファイルの署名 (CRC) を保持します。

それ以降のリブート時には、システムに渡される CVAC 設定ファイルの CRC と、最後に適用した CVAC 構成の CRC が照合チェックされます。変更がなければ、何も実行されません。つまり、最初の CVAC による構成の適用後、それ以降の構成の変更は、渡される CVAC 設定ファイルに変更がない場合、システム設定が変更されない (または元に戻されない) ことを意味します。

変更があると、新しい構成が既存の構成を上書きするように適用されます。これにより、最初の CVAC 構成、それ以降の構成の変更、および構成済みのシステムに対する CVAC 構成の追加の変更が可能になります。

すでにコミットされた構成上にパーサーによって読み込まれるだけの構成を、CVAC は合理化しようとしません。パーサーを通過するコマンドはコミットされ、前述したようにエラーがログに記録されます。

データ プレーンの管理

Cisco IOS XRv 9000 ルータのデータ プレーンは、ルータの起動後に自動的に開始されます。コントロールプレーンがデータプレーンとの通信を確立しない場合や通信できなくなった場合、コントロールプレーンは自動的にデータ プレーンを再起動します。

データプレーンは、メンテナンスとトラブルシューティングのために、管理コンソールから手動で開始、シャットダウン、リロードできます。この表では、データプレーンの管理に必要なコマンドを示します。

表 3: データプレーンの管理コマンド

タスク	使用するコマンド
データプレーンのステータスを確認する。	<pre>show sdr sysadmin-vm:0_RP0# show sdr Thu May 7 18:38:38.996 UTC sdr default-sdr location 0/RP0/VM1 sdr-id 2 IP Address of VM 192.0.0.4 MAC address of VM E2:3A:DD:0A:8C:06 VM State RUNNING start-time 2015-05-07T17:54:39.457822+00:00 Last Reload Reason FIRST_BOOT Reboot Count 1 location 0/RP0/VM2 sdr-id 2 IP Address of VM 192.0.0.6 MAC address of VM E2:3A:DD:0A:8C:06 VM State RUNNING start-time 2015-05-07T18:22:44.136498+00:00 Last Reload Reason FIRST_BOOT Reboot Count 1</pre> <p>データプレーンの場所をメモしておく必要があります。この例では、データプレーンの状態は場所 0/RP0/VM2 の下に表示されています。</p>
データプレーンを開始する	<pre>sdr default-sdr location <data-plane-location> start</pre> <p>例</p> <pre>sysadmin-vm:0_RP0# sdr default-sdr location 0/RP0/VM2 start Mon May 4 17:16:37.867 UTC Start ? [no,yes] yes result start sdr default-sdr location 0/RP0 request acknowledged</pre>
データプレーンをシャットダウンする	<pre>sdr default-sdr location <data-plane-location> shut</pre> <p>例</p> <pre>sysadmin-vm:0_RP0# sdr default-sdr location 0/RP0/VM2 shut Mon May 4 17:12:32.397 UTC Shut ? [no,yes] yes result shutdown sdr default-sdr location 0/RP0 request acknowledged</pre>

タスク	使用するコマンド
データプレーンをリロードする	<pre>sdr default-sdr location <data-plane-location> reload</pre> <p>例</p> <pre>sysadmin-vm:0_RP0# sdr default-sdr location 0/RP0/VM2 reload Mon May 4 17:21:17.390 UTC Reload ? [no,yes] yes result graceful reload sdr default-sdr location 0/RP0 request acknowledged</pre>



(注) 通常の運用では、ユーザはデータプレーンを手動で開始および停止してはいけません。

データプレーンのデバッグ

Cisco IOS XRv 9000 ルータには、データプレーンのステータスと統計をチェックするコマンドのセットが用意されています。これらのコマンドは、次のとおりです。



(注) 次に示すデータプレーンのデバッグコマンドは、一時的にトラフィックの転送を中断することがあります。

- `show controller dpa statistics`
- `show controller dpa fib ipv4|ipv6 [<prefix> | summary]`

- **show controller dpa version** : データプレーンのバージョンを表示します。

次に例を示します。

```
RP/0/RP0/CPU0:ios#show controller dpa version
Fri May 29 19:28:16.520 UTC
Image built on 13:29:13 May 29 2015 in workspace /workspace1/shope/ttl_commit
DPA started May 29 18:11:23, up 0 days, 01:16
```

- **show controller dpa logging** : データプレーンログを表示します。デフォルトでは、エラーや重要なイベントのログ情報のみが使用可能です。

次に例を示します。

```
RP/0/RP0/CPU0:SS_Node1#show controllers dpa logging
Mon Jun 29 19:47:33.245 UTC
Jun 29 01:43:32.820: Log File Started
Jun 29 01:43:32.820: DPA_INFO: DPA beginning initialization
Jun 29 01:43:32.823: DPA_INFO: Dataplane Agent enabled
Jun 29 01:43:32.823: DPA_INFO: Image built on 15:02:53 Jun 25 2015
Jun 29 01:43:32.823: DPA_INFO: Table WRED_STR of size 8388480 is being initialized
```

```

Jun 29 01:43:32.824: DPA_INFO: Table STATIC_POLICER_STR of size 8192 is being
initialized with data
Jun 29 01:43:32.824: DPA_INFO: Table HASH_DYN_BUCKET_STR of size 3355264 is being
initialized
Jun 29 01:43:32.825: DPA_INFO: Table HASH_BUCKET_STR of size 33552832 is being
initialized
Jun 29 01:43:32.829: DPA_INFO: Table DYN_FREE_BLOCK_STR of size 16777216 is being
initialized
Jun 29 01:43:32.832: DPA_INFO: Table INDEX_Q_STR of size 8192 is being initialized

Jun 29 01:43:32.832: DPA_INFO: Table HASH_HOST_DYN_BUCKET_STR of size 1677504 is
being initialized

```

- **show controller dpa statistics global** : データプレーンの統計を表示します。この統計には、ドロップパケット数、コントロールプレーンから入ってきたパケット、およびコントロールプレーンにパントされたパケットが含まれます。

次に例を示します。

```

RP/0/RP0/CPU0:ios#show control dpa statistics global
Fri May 29 19:27:35.497 UTC
Index  Punt                                     Count
-----
1575  ARP                                           28
1677  IFIB                                         1341
1701  IPv4 FIB                                     5

Index  Inject                                     Count
-----
267   IPv4 from fabric                           31
268   IPv4 from fabric multicast                 1290
270   IPv4 from fabric next-hop                 55
275   Inject to fabric                          1376
276   Inject to port                            11

Index  Drop                                     Count
-----
63    Egress uIDB in down state                 1
113   IPv6 disabled in uIDB                    52

```

- **show controller dpa fib ipv4|ipv6 [<prefix> | summary]** : データプレーンの FIB エントリを表示します。<prefix> および summary キーワードはオプションです。

次に例を示します。

```

RP/0/RP0/CPU0:ios#show controller dpa fib ipv4
Fri May 29 19:54:40.110 UTC

VRF id: 0
Default prefix 0.0.0.0/0 -> leaf:46423
total number of prefix:35
total_node_allocated:29 leaf_inserts:50 leaf_deletes 15 leaf_replaces: 2

Prefix          leaf_index
224.0.0.0/4     46436(0xb564)
224.0.0.0/24    46434(0xb562)
255.255.255.255/32 46429(0xb55d)
0.0.0.0/32     46433(0xb561)
1.1.1.1/32     46513(0xb5b1)
2.2.2.2/32     46510(0xb5ae)

```

summary キーワードを使用すると、このコマンドは各 **vrf** テーブルのプレフィックスと運用統計を表示します。次に例を示します。

```
RP/0/RP0/CPU0:R1-PE1#show control dpa fib ipv4 summary
```

```
VRF id: 0
Default prefix 0.0.0.0/0 -> leaf:46421
total number of prefix: 27859
allocated nodes:          1089
leaf_inserts:             27922
leaf_deletes              63
leaf_replaces:           173
```

```
VRF id: 1
Default prefix 0.0.0.0/0 -> leaf:46444
total number of prefix: 430
allocated nodes:          26
leaf_inserts:             430
leaf_deletes              0
leaf_replaces:           213
```

<**prefix**> キーワードを使用すると、このコマンドは、すべてのプレフィックスと、そのプレフィックスに一致する **vrf** テーブルのリストを表示します。次に例を示します。

```
RP/0/RP0/CPU0:ios#show controller dpa fib ipv4 5.11.23.131/32
```

```
VRF id: 0
Prefix          leaf_index
5.11.23.131/32  1164818(0x11c612)
```

```
VRF id: 1
```

```
VRF id: 2
```

```
VRF id: 3
```

- **show controller dpa tm queue <num>** : トラフィック マネージャ キューの内部データを表示します。このデータには、DRR ウェイト、Q-limit、瞬間的なパケットとバイト数が含まれます。

次に例を示します。

```
RP/0/RP0/CPU0:ios#show controller dpa tm queue 1
```

```
Fri May 29 19:31:25.556 UTC
Queue 1
Parent Subport:    0
Weight:           10
Q-Limit:          625000
Packets:          0
Bytes:            0
```

- **show controller dpa tm subport <num>** : トラフィック マネージャ サブポートの内部データを表示します。このデータには、DRR ウェイト、シェーピングされたレート、キュー設定、および瞬間的なパケットが含まれます。

次に例を示します。

```
RP/0/RP0/CPU0:ios#show controller dpa tm subport 3
```

```
Fri May 29 19:44:12.993 UTC
```

```

Subport 3
  Parent vPort:      3
  Weight:            10200
  Rate:              776726
  Being Deleted:    no
  Configured:       yes
  Queue 24 pkts:    0 bytes:    0
  Queue 25 pkts:    6 bytes:    8376
  Queue 26 pkts:    0 bytes:    0
  Queue 27 pkts:    0 bytes:    0
  Queue 28 pkts:    0 bytes:    0
  Queue 29 pkts:    0 bytes:    0
  Queue 30 pkts:    0 bytes:    0
  Queue 31 pkts:    0 bytes:    0
  Priority Queues:   1
  Best effort Queues: 7

```

- **show controller dpa tm vport <num>** : トラフィック マネージャ仮想ポートの内部データを表示します。このデータには、そのポートのレートとフロー制御が現在アクティブかどうかが含まれます。

次に例を示します。

```

RP/0/RP0/CPU0:ios#show controller dpa tm vport 0
Fri May 29 19:32:39.447 UTC
vPort 0
  Parent port:      0
  Rate:             95
  Flow control:     0

```

Early Fast Discard

Early Fast Discard (EFD) は、着信トラフィックがルータのキャパシティを超過した状況进行处理するための輻輳保護機能です。通常の輻輳制御はトラフィック マネージャ (TM) および添付された QoS ポリシーによって処理されますが、極端な場合には EFD 機能が有効化されます。EFD 機能は、優先順位の高いトラフィック (キープアライブ、制御、BFD など) にフィルタを適用し、それ以外のトラフィックをデータパス処理の非常に早い段階で破棄することで、制御パケットの重要なフローを維持します。

EFD 機能を有効化するには、設定モードで **early-fast-discard** コマンドを使用します。廃棄するトラフィックは、IP プレシデンス値、MPLS exp 値、VLAN cos 値を設定することで定義します。

Early Fast Discard の設定

使用例

Early Fast Discard はデータプレーン上に設定され、破棄するトラフィックを次の条件に基づいて定義することにより、着信トラフィック フローを管理します。

- IP precedence=4

- MPLS exp=3
- VLAN cos=5

デフォルト値は 6 および ge (greater than or equal to : 「以上」の略) です。

設定 (Configuration)

```
Router# configure
Router(config)# hw-module early-fast-discard
Router(config-early-fast-discard)# ip-prec 4 ip-op [lt | ge]
Router(config-early-fast-discard)# mpls-exp 3 mpls-op [lt | ge]
Router(config-early-fast-discard)# vlan-cos 5 vlan-op [lt | ge]
Router(config-early-fast-discard)# exit
```



(注) **no hw-module early-fast-discard** コマンドを使用して、EFD を非アクティブ化します。

実行コンフィギュレーション

```
RP/0/RP0/CPU0:ios#show run hw-module early-fast-discard
Thu Jul 16 15:51:34.672 UTC
hw-module early-fast-discard
 ip-prec 4 ip-op ge
 mpls-exp 3 mpls-op ge
 vlan-cos 5 vlan-op lt
!
```

関連項目

- [Early Fast Discard \(18 ページ\)](#)

bgp bestpath igp-metric ignore コマンドのサポート

bgp bestpath igp-metric ignore コマンドを使用すると、システムは内部ゲートウェイプロトコル (IGP) メトリックを無視して、ベストパスを選択できます。

デフォルトでは、BGP は常に IGP メトリックが最も低いパスを優先します。一方には IGP メトリックがあり、もう一方には IGP メトリックがない 2 つのパスがある場合に **bgp bestpath igp-metric ignore** コマンドを実行すると、どちらのパスも IGP メトリックがないかのように、BGP はベストパスの計算を実行します。

次に、ベストパスの選択を実行するときに内部ゲートウェイプロトコル (IGP) メトリックを無視するようにソフトウェアを設定する例を示します。この例では、ルータ BGP VRF コンフィギュレーション モードでコマンドが設定されています。

bgp bestpath igp-metric ignore コマンドのサポート

```
RP/0/0/CPU0:router#configure  
RP/0/0/CPU0:router(config)#router bgp 50000  
RP/0/0/CPU0:router(config-bgp)#vrf 1  
RP/0/0/CPU0:router(config-bgp-vrf)#bgp bestpath igp-metric ignore
```