パブリッククラウド用 L2 拡張の設定

この章では、企業とクラウドプロバイダーが LISP を使用して Cisco Catalyst 8000V インスタンスを含むパブリッククラウドの L2 拡張を設定できるようにする方法について説明します。コマンドライン インターフェイスを使用して、パブリック クラウド ネットワークとエンタープライズ ネットワーク間のレイヤ 2 ドメインを拡張します。

LISP レイヤ 2 拡張を設定する前に理解しておく必要がある用語と概念の一部を次に示します。

- **Locator/ID Separation Protocol（LISP）**：LISP は、単一 IP アドレスではなく 2 つの名前空間を使用するネットワークアーキテクチャおよびプロトコルです。
  - エンドポイント識別子（EID）：エンドホストに割り当てられます。
  - ルーティングロケータ（RLOC）：グローバル ルーティング システムを構成するデバイス（主にルータ）に割り当てられます。
- **LISP 対応仮想化ルータ**：ルーティング機能と LISP 機能（ホストモビリティを含む）をサポートする仮想マシンまたはアプライアンス。
- **エンドポイント ID（EID）**：EID は、パケットの最初の（最も内側の）LISP ヘッダーに含まれる送信元および宛先アドレスフィールドで使用される IPv4 または IPv6 アドレスです。
- **ルーティングロケータ（RLOC）**：LISP ノード間のフローをカプセル化および転送するために使用される IPv4 または IPv6 アドレス。RLOC は、EID-to-RLOC マッピングルックアップの出力です。
- **出力トンネルルータ（ETR）**：ETR はトンネルエンドポイントであるデバイスで、LISP 機能のあるコアネットワークの部分（インターネットなど）にサイトを接続し、サイトの EID-to-RLOC マッピングを公開し、Map-Request メッセージに応答し、サイトのエンドシステムに LISP でカプセル化されたユーザーデータをカプセル化解除して配信します。運用中、ETR は設定済みのすべての Map Server に定期的に Map-Register メッセージを送信します。送信される Map-Register メッセージには、ETR のサイトに接続されている EID 番号付きネットワークのすべての EID-to-RLOC エントリが含まれます。
- **入力トンネルルータ（ITR）**：ITR はトンネルの開始点となるデバイスです。ITR は、LISP 機能のあるサイトに向かうすべてのトラフィックの EID-to-RLOC マッピングを検索しま

す。ITR が EID 宛てのパケットを受信すると、まずマッピングキャッシュの EID を調べます。ITR が一致を見つけると、LISP ヘッダー内でパケットをカプセル化し、RLOC の 1 つを送信元 IP アドレスとし、マッピング キャッシュ エントリからの RLOC の 1 つを IP 接続先とします。ITR はその後、パケットを通常どおりルーティングします。

- **xTR**：入力トンネルルータ（ITR）機能と出力トンネルルータ（ETR）機能の両方を実行するデバイスの総称。
- **PxTR**：IP ネットワークと LISP ネットワーク間の相互接続ポイント。このピアリングポイントで ITR と ETR の役割を果たします。
- **マップサーバー（MS）**：MS は、LISP サイト ETR がその EID プレフィクスを登録する LISP インフラストラクチャ デバイスです。MS は、クライアント出力トンネルルータ（ETR）からの登録要求を承認し、正常に登録されたそれらの ETR の EID プレフィックスを集約し、Border Gateway Protocol（BGP）を用いて集約されたプレフィックスを 代替論理トポロジ（ALT）にアドバタイズすることで、分散 LISP マッピングデータベースの一部を実行します。

  小規模なプライベート マッピング システム展開では、すべての ETR がそれぞれの MS に登録されるように設定した状態で、MS はスタンドアローンとして設定できます（または複数の MS があってもよい）。複数の場合、すべての MS はプライベート マッピング システム展開内のマッピングシステムの完全な情報を有します。

  より大規模なマッピングシステム展開またはパブリック マッピング システム展開では、MS は、Generic Routing Encapsulation（GRE）トンネルと BGP セッションの部分メッシュを用いて、他のマップサーバーシステムに対して設定されます。
- **マップリゾルバ（MR）**：MR は LISP インフラストラクチャ デバイスです。ITR は、EID-to-RLOC マッピングを解決する際に、LISP Map-Request クエリを MR に送信します。MR は要求を受信し、適切なマップサーバーを選択します。

LISP と用語の詳細な概要については、「Locator ID Separation Protocol Overview」を参照してください。

LISP レイヤ 2 拡張の設定

Cisco Catalyst 8000V は、パブリッククラウド、プライベートクラウド、およびハイブリッドクラウドに展開できます。企業がハイブリッドクラウドに移行する場合、サーバーに対して一切変更を加えずに、サーバーをクラウドに移行する必要があります。企業は、同じサーバー IP

アドレス、サブネットマスク、およびデフォルトゲートウェイ設定を使用することを望むかもしれません。クラウド内で独自の IP アドレス方式を使用し、クラウドプロバイダーのインフラストラクチャのアドレス方式によって制限されないことを望む可能性があります。

この要件を満たすために、シスコは Amazon Web Services（AWS）上で動作する LISP レイヤ 2 拡張を Cisco Catalyst 8000V に提供します。この場合、Cisco Catalyst 8000V インスタンスはエンタープライズ データセンターとパブリッククラウド間のブリッジとして機能します。LISP レイヤ 2 拡張を設定すると、プライベートデータセンター内のレイヤ 2 ネットワークをパブリッククラウドに拡張して、お客様のサイトとパブリッククラウド間でのホスト到達可能性を実現できるようになります。また、データセンターとパブリッククラウド間のアプリケーションワークロードの移行を有効にすることもできます。

利点

- データ移行が容易になり、ネットワークのワークロード IP アドレスやファイアウォールルールが最適化されます。これにより、ブロードキャストドメインを拡張せずにサブネットの連続性を確保できます。
- プロバイダーサイトで VM を仮想的に追加し、VM がプロバイダーサイトで実行されている間に、クラウドバーストを活用して、仮想的に VM をエンタープライズサーバーに挿入できるようにします。
- 部分的な障害回復と障害回避のためのバックアップサービスを提供します。

LISP レイヤ 2 拡張の設定の前提条件

各 Cisco Catalyst 8000V ルータに 1 つの外部 IP アドレスを設定する必要があります。この場合、IPsec トンネルは 2 つの Cisco Catalyst 8000V インスタンスの IP アドレス間に構築され、IPsec トンネルにはプライベートアドレスがあります。

LISP レイヤ 2 拡張の設定の制約事項

- AWS ECS サブネットでは、企業 VRF 数と VM アドレス数が制限されます。
- IPv6 アドレス形式は、Cisco Catalyst 8000V Amazon マシンイメージ（AMI）ではサポートされていません。

LISP レイヤ 2 拡張の設定

L2 拡張機能を設定するには、まず AWS に Cisco Catalyst 8000V インスタンスを展開し、インスタンスを xTR として設定する必要があります。その後、展開を完了するためにマッピングシステムを設定する必要があります。

LISP サイトは、アップストリーム プロバイダーへの 2 系統の接続を持つ、ITR と ETR の両方として設定された（xTR とも呼ばれる）Cisco Catalyst 8000V インスタンスを使用します。次に LISP サイトは、ネットワークコアのマップリゾルバ/マップサーバー（MR/MS）として設定されたスタンドアロンデバイスに登録されます。マッピングシステムは、移行済みのパブリック IP に送信されるパケットの LISP カプセル化およびカプセル化解除を実行します。AWS からのトラフィックについては、必要に応じて（接続先へのルートがルーティングテーブルで見つからない場合は常に）、Cisco Catalyst 8000V インスタンスがエンタープライズ データセンターの PxTR を介してルーティングします。

LISP マップサーバーおよびマップリゾルバをマッピングサービスに使用する際、LISP xETR 機能を設定して有効化するには、次の手順を実行します。

AWS での Cisco Catalyst 8000V インスタンスの作成

ステップ 1 Amazon Web Services にログインします。左側のナビゲーションウィンドウで、[VPC] をクリックします。

ステップ 2 [Start VPC Wizard] をクリックし、左側のペインから [VPC with Single Public Subnet] を選択します。

ステップ 3 [Select] をクリックします。

ステップ 4 仮想プライベートクラウドにサブネットを作成します。次のプロパティを使用します。

a) Default Subnet：10.0.0.0/24（パブリック IP にマッピングされる）。

b) Additional subnets：0.0.1.0/24 および 1.0.0.2.0/24。これらはプライベート IP アドレスであり、Cisco Catalyst 8000V インスタンスから見て内部である可能性があります。

ステップ 5 [Create VPC] を選択します。

ステップ 6 [Security] > [Network ACLs] を選択します。

ステップ 7 [Create Security Group] をクリックして、Cisco Catalyst 8000V インスタンスのセキュリティグループを作成します。次のプロパティを設定します。

a) Name：SSH アクセス

b) TCP Port 22 traffic：インバウンド許可

c) SSH access to C8000V for management：有効

ステップ 8 追加のセキュリティグループを作成するには、ステップ 6 を実行します。

ステップ 9 Cisco Catalyst 8000V の製品ページに移動し、[Continue] をクリックします。

ステップ 10 [Launch with E2 Console] をクリックして、地理的地域に応じた Cisco Catalyst 8000V を起動します。

ステップ 11 適切なインスタンスタイプを選択します。サポートされているインスタンスタイプについては、表 2-1 および 2-2 を参照してください。

中規模インスタンスタイプ（m1.medium）の最小メモリ要件は 10Mbps です。大規模インスタンスタイプ（m1.large）の場合は 50Mbps です。

ECU は Elastic Compute Unit の略です。ECU は、CPU 容量を測定する Amazon 独自の方法です。

すべての EC2 インスタンスはハイパースレッド化されています。

ステップ 12 作成した VPC で Cisco Catalyst 8000V インスタンスを起動します。次のプロパティを使用します。

a) [Shutdown] 動作を [Stop] に設定します。
b) [Tenancy] を [Shared] に設定します。共有ハードウェアインスタンスを実行するには、[Shared] オプションを選択します。

ステップ 13 インスタンスをセキュリティグループ（SSH-ACCESS）に関連付けます。セキュリティルールを使用すると、Cisco Catalyst 8000V インスタンスのトラフィックを制御するファイアウォールルールを設定できます。

ステップ 14 秘密キーを Cisco Catalyst 8000V インスタンスに関連付けます。キーペアは、秘密キーと公開キーで構成されます。Cisco Catalyst 8000V インスタンスを認証して接続するには、秘密キーを指定する必要があります。公開キーは AWS に保存されます。必要に応じて、新しいキーペアを作成できます。

ステップ 15 [Launch Instance] をクリックします。

ステップ 16 Cisco Catalyst 8000V インスタンスが AWS に展開されているかどうかを確認します。

展開に成功すると、ステータスが *2/2/ checks passed* に変わります。

サブネットの設定

ステップ 1 Cisco Catalyst 8000V インスタンスを選択します。

ステップ 2 [Actions] > [Networking] > [Manage IP Addresses] の順に選択します。

ステップ 3 エンタープライズ ホスト アドレスを指定します。この IP アドレスは、eth1 のセカンダリアドレスです。

ステップ 4 [Yes, Update] をクリックします。

AWS 上の Cisco Catalyst 8000V とエンタープライズシステム上の Cisco Catalyst 8000V 間におけるトンネルの設定

エンタープライズデータセンター内に展開された Cisco Catalyst 8000V インスタンスとパブリッククラウド内に展開された Cisco Catalyst 8000V インスタンス間の通信は、両者の間に確立された IP セキュリティ（IPsec）トンネルによって保護されます。LISP カプセル化トラフィックは、パブリッククラウドと企業間のデータ発信元認証、完全性保護、アンチリプライ保護、および機密性を実現する IPsec トンネルで保護されます。

ステップ 1 AWS で Cisco Catalyst 8000V インスタンスを設定します。

```
interface Loopback1
 ip address 33.33.33.33 255.255.255.255
!
interface Tunnel2
 ip address 30.0.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 173.39.145.79
 tunnel protection ipsec profile p2p_pf1
```

```
!
interface GigabitEthernet2
 ip address 10.10.10.140 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

ステップ 2 企業サイトで 2 番目の Cisco Catalyst 8000V インスタンスを設定します。

```
interface Loopback1
 ip address 11.11.11.11 255.255.255.255

interface Tunnel2
 ip address 30.0.0.1 255.255.255.0
 tunnel source GigabitEthernet2
 tunnel mode ipsec ipv4
 tunnel destination 52.14.116.161
 tunnel protection ipsec profile p2p_pf1
!
!
interface GigabitEthernet3
 ip address 10.10.10.2 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

AWS で実行されているインスタンスでの LISP xTR の設定

AWS で実行されている Cisco Catalyst 8000V インスタンスで LISP xTR を設定するには、「Configuring LISP (Location ID Separation Protocol)」のセクションの設定手順に従います。

例：

```
router lisp
 locator-set aws
  33.33.33.33 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  itr
  etr map-server 11.11.11.11 key cisco
  etr
  use-petr 11.11.11.11
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set aws
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
```

```
   exit-service-ipv4
  !
  exit-instance-id
 !
 exit-router-lisp
!
router ospf 11
 network 30.0.0.2 0.0.0.0 area 11
 network 33.33.33.33 0.0.0.0 area 11
!

router lisp
 locator-set dmz
  11.11.11.11 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  etr map-server 11.11.11.11 key cisco
  etr
  proxy-etr
  proxy-itr 11.11.11.11
  map-server
  map-resolver
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set dmz
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  exit-instance-id
 !
 site DATA_CENTER
  authentication-key cisco
  eid-record 10.10.10.0/24 accept-more-specifics
  exit-site
 !
 exit-router-lisp
!
router ospf 11
 network 11.11.11.11 0.0.0.0 area 11
 network 30.0.0.1 0.0.0.0 area 11
!

!
!
```

AWS 上の Cisco Catalyst 8000V とエンタープライズシステム上の Cisco Catalyst 8000V 間における LISP レイヤ 2 トラフィックの確認

LISP レイヤ 2 トラフィックを確認するには、次の手順を実行します。

例：

```
Router#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 2, no-route 0, inactive 0

10.0.1.1/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
10.0.1.20/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
Router#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 4 entries

0.0.0.0/0, uptime: 00:09:49, expires: never, via static-send-map-request
  Negative cache entry, action: send-map-request
10.0.1.0/24, uptime: 00:09:49, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.4/30, uptime: 00:00:55, expires: 00:00:57, via map-reply, forward-native
  Encapsulating to proxy ETR
10.0.1.100/32, uptime: 00:01:34, expires: 23:58:26, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
11.11.11.11  00:01:34  up           1/100       -
Router#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set aws
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 2
  Last dynamic-EID discovered: 10.0.1.20, 00:01:37 ago
    10.0.1.1, GigabitEthernet2, uptime: 00:09:23
      last activity: 00:00:42, discovered by: Packet Reception
    10.0.1.20, GigabitEthernet2, uptime: 00:01:37
      last activity: 00:00:40, discovered by: Packet Reception

Router-DC#show ip lisp
Router-DC#show ip lisp data
Router-DC#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 1, no-route 0, inactive 0

10.0.1.100/32, dynamic-eid subnet1, inherited from default locator-set dc
  Locator  Pri/Wgt  Source     State
11.11.11.11    1/100  cfg-addr   site-self, reachable
```

```
Router-DC#show ip lisp
Router-DC#show ip lisp map
Router-DC#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 2 entries

10.0.1.0/24, uptime: 1d08h, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.20/32, uptime: 00:00:35, expires: 23:59:24, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
33.33.33.33  00:00:35  up           1/100

Router-DC#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set dc
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 1
  Last dynamic-EID discovered: 10.0.1.100, 1d08h ago
    10.0.1.100, GigabitEthernet2, uptime: 1d08h
      last activity: 00:00:47, discovered by: Packet Reception

Router-DC#show lisp site
LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport

Site Name      Last      Up     Who Last             Inst     EID Prefix
               Register         Registered           ID
dc             never     no     --                            10.0.1.0/24
               00:08:41  yes#   33.33.33.33                   10.0.1.1/32
               00:01:00  yes#   33.33.33.33                   10.0.1.20/32
               1d08h     yes#   11.11.11.11                   10.0.1.100/32
Router-DC#show ip cef 10.0.1.20
10.0.1.20/32
  nexthop 33.33.33.33 LISP0
Router-DC#
```

PMD マルチキューのサポート

Cisco IOS XE 17.7.1 以降では、AWS で実行される Cisco Catalyst 8000V インスタンスで PMD マルチキュー機能がサポートされます。現在、Cisco Catalyst 8000V で割り当てられる PMD RX キューと PMD TX キューはインターフェイスごとに 1 つだけです。この機能を使用すると、Cisco Catalyst 8000V で 4 つの PMD RX キューと 8 つの PMD TX キューが割り当てられます。これにより、パケット処理率が増加してパフォーマンスが向上します。

Cisco IOS XE 17.9.1 以降では、Cisco Catalyst 8000V による PMD TX キューの割り当てが 12 個に増加しています。

（注） IPsec トンネルの IP アドレスペアは PMD TXQ にハッシュされます。したがって、アドレスが競合してパフォーマンスが低下することがあります。この問題を回避するには、**show platform hardware qfp active datapath infrastructure sw-nic** コマンドを使用して、パフォーマンスが最適になるようにトラフィックが 8 つのキューすべてに均等に分散しているかどうかを確認します。

次に、**show platform hardware qfp active datapath infrastructure sw-nic** コマンドのサンプルのコマンド出力を示します。

```
Router# show platform hardware qfp act datapath infrastructure sw-nic
pmd b19811c0 device Gi1
  RX: pkts 418  bytes 37655 return 0 badlen 0
    pkts/burst 1  cycl/pkt 0  ext_cycl/pkt 0
    Total ring read 91995516, empty 91995113
  TX: pkts 355  bytes 57833
      pri-0: pkts 60  bytes 5590
             pkts/send 1
      pri-1: pkts 32  bytes 2616
             pkts/send 1
      pri-2: pkts 6  bytes 303
             pkts/send 1
      pri-3: pkts 38  bytes 6932
             pkts/send 1
      pri-4: pkts 176  bytes 39279
             pkts/send 1
      pri-5: pkts 25  bytes 1962
             pkts/send 1
      pri-6: pkts 8  bytes 459
             pkts/send 1
      pri-7: pkts 10  bytes 692
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 3160
    send 343  sendnow 0
    forced 343  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 0
    TX Queue 1: full 0  current index 0  hiwater 0
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 0
    TX Queue 4: full 0  current index 0  hiwater 0
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
pmd b1717380 device Gi2
  RX: pkts 289216546  bytes 102405925473 return 0 badlen 0
    pkts/burst 7  cycl/pkt 326  ext_cycl/pkt 381
    Total ring read 141222555, empty 103047391
  TX: pkts 757922  bytes 260498122
      pri-0: pkts 94302  bytes 32428428
             pkts/send 1
      pri-1: pkts 95525  bytes 32791822
             pkts/send 1
      pri-2: pkts 93002  bytes 31950500
             pkts/send 1
      pri-3: pkts 96799  bytes 33381108
             pkts/send 1
      pri-4: pkts 90823  bytes 31179044
             pkts/send 1
```

```
      pri-5: pkts 97436  bytes 33455916
             pkts/send 1
      pri-6: pkts 93243  bytes 32113540
             pkts/send 1
      pri-7: pkts 96792  bytes 33197764
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 760
    send 685135  sendnow 3
    forced 685117  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 31
    TX Queue 1: full 0  current index 0  hiwater 31
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 0
    TX Queue 4: full 0  current index 1  hiwater 31
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
pmd b14ad540 device Gi3
  RX: pkts 758108  bytes 302121148 return 0 badlen 0
    pkts/burst 1  cycl/pkt 572  ext_cycl/pkt 811
    Total ring read 78867251, empty 78155478
  TX: pkts 756904  bytes 301747138
      pri-0: pkts 9  bytes 540
             pkts/send 1
      pri-1: pkts 200064  bytes 80223776
             pkts/send 1
      pri-3: pkts 244086  bytes 97204792
             pkts/send 1
      pri-4: pkts 3  bytes 822
             pkts/send 1
      pri-5: pkts 250502  bytes 99404344
             pkts/send 1
      pri-7: pkts 62240  bytes 24912864
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 737
    send 705364  sendnow 3
    forced 705355  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 0
    TX Queue 1: full 0  current index 0  hiwater 31
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 31
    TX Queue 4: full 0  current index 0  hiwater 0
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。